Financial Modeling

Simon Benninga

fourth edition

FINANCIAL MODELING

FINANCIAL MODELING

Simon Benninga

With a section on Visual Basic for Applications
by Benjamin Czaczkes

FOURTH EDITION

The MIT Press
Cambridge, Massachusetts
London, England

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu.

Print copies of this book include a card affixed to the inside back cover with a unique access code. Access codes are required to download Excel worksheets and solutions to end-of-chapter exercises. If you purchased an e-book, you may obtain a unique access code by emailing journals-access@mit.edu or calling 617-253-2889 or 800-207-8354 (toll-free in the U.S. and Canada).

This book was set in Times Roman by Toppan Best-set Premedia Limited. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Benninga, Simon.
Financial modeling / Simon Benninga.—Fourth edition.
pages cm
Includes bibliographical references and index.
ISBN 978-0-262-02728-1 (hardcover : alk. paper)
1. Finance—Mathematical models. 2. Microsoft Visual Basic for applications. I. Title.
HG173.B46 2014
332.01'5118—dc23
2013032409

10 9 8 7 6 5 4 3 2 1

To the memory of our parents:

Helen Benninga (1913–2008)
Groningen, Netherlands – Jerusalem, Israel

Noach Benninga (1909–1994)
Eenrum, Netherlands – Asheville, North Carolina

Esther Czaczkes (1931–2012)
Jerusalem, Israel – Jerusalem, Israel

Alfred Czaczkes (1923–1997)
Vienna, Austria – Jerusalem, Israel

Contents

Preface

The three previous editions of *Financial Modeling* have received a gratifyingly positive response from readers. The combination of a "cookbook," mixing explanation and implementation using Excel, has fulfilled a need in both the academic and the practitioner markets from readers who realize that the implementation of the finance basics typically studied in an introductory finance course requires another, more heavily computational and implementational approach. Excel, the most widely used computational tool in finance, is a natural vehicle for deepening our understanding of the materials.

In this fourth edition of *Financial Modeling*, I have added a section (Chapters 24–30) on Monte Carlo methods. The intention is to add a focus on the simulation of financial models. I have become convinced that a statistical understanding of modeling ("What is the mean and sigma of the portfolio return?") understates the impact of the uncertainty. Only by simulating the models and the return processes can we get a good feel for the dimensions of the uncertainty.

With the added section on Monte Carlo, *Financial Modeling* now consists of seven sections. Each of the first five sections of the book relates to a specific area of finance. These sections are independent of each other, though the reader should realize that they all assume some familiarity with the finance area—*Financial Modeling* is not an introductory text. Section I (Chapters 1–7) deals with corporate finance topics; Section II (Chapters 8–14) with portfolio models; Section III (Chapters 15–19) with option models; and Section IV (Chapters 20–23) with bond-related topics. Section V, as discussed above, introduces the reader to Monte Carlo methods in finance.

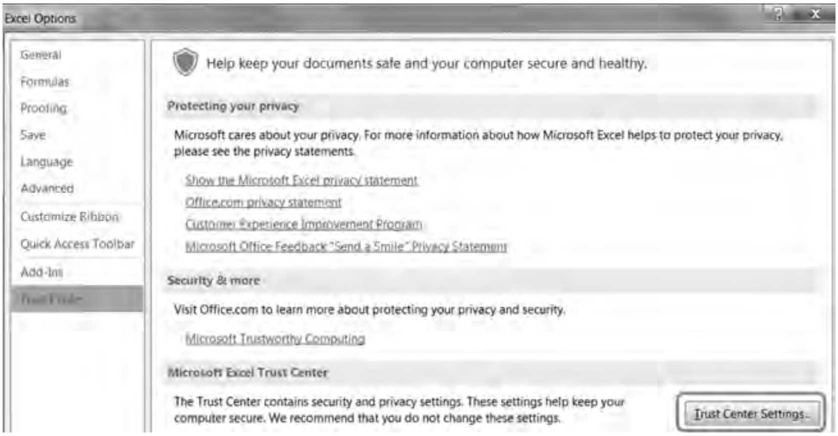
The last two sections of *Financial Modeling* are technical in nature. Section VI (Chapters 31–35) relates to various Excel topics which are used throughout the book. Chapters in Section VI can be read and accessed as necessary. Section VII (Chapters 36–39) deals with Excel's programming language, Visual Basic for Applications (VBA). VBA is used throughout *Financial Modeling* to create functions and routines which make life easier, but it is never intrusive—in principle the reader can understand the materials in all of the other chapters of *Financial Modeling* without needing the VBA chapters.

New Materials and Updates

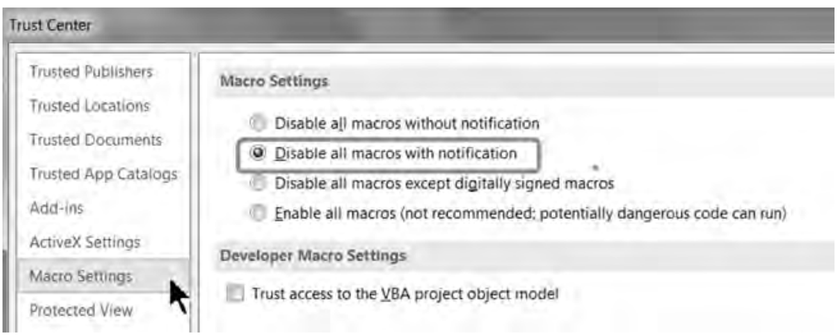
This edition of *Financial Modeling* contains much new and updated material. We have already mentioned the new section on Monte Carlo methods. Also new are two chapters on valuation (Chapters 2 and 4) and a chapter on term structure modeling (Chapter 22). Much of the material has been tweaked and improved. For example, the discussion of Excel financial functions now includes a discussion of XIRR and XNPV, including a fix for the bugs in these functions.

Getformula

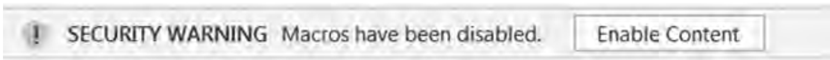
The Excel files with this edition include a function called **Getformula** that enables the user to track cell contents. **Getformula** is discussed in Chapter 0 and also on a file on the disk that is included with *Financial Modeling*. To allow **Getformula** to work, go to **File|Options|Trust Center**:



In the **Trust Center** settings, I recommend the following setting:



If you have done this, then when opening an Excel notebook for the first time, you will be confronted by the following warning:



For notebooks that come with this book, you can safely click **Enable Content**, which enables the formulas on the notebook.

Excel Versions

In the examples throughout the book I have used Excel 2013. To the best of my knowledge, all of the spreadsheets work in Excel versions 2003, 2007, 2010, and 2011 (for Mac), although some minor and obvious adaptations by the reader may be called for.

Files for the Fourth Edition

Purchasers of *Financial Modeling* get access to all the Excel files for the chapters and exercises.

Using *Financial Modeling* in a University Course

Financial Modeling has become the book of choice in many advanced finance classes that stress the combination of modeling/Excel skills and a deeper understanding of the underlying financial models. The *Financial Modeling*–based courses are often a third- or fourth-year undergraduate or second-year MBA course. The courses are very different and include much instructor-specific input, but they seem to have a few general features in common:

- A typical course starts with two or three classes which stress the Excel skills needed for financial modeling. Often these courses are held in a computer lab. Though almost all business school students know Excel, they often do not know the finesses of data tables (Chapter 31), some of the basic financial functions (Chapters 1 and 33), and array functions (Chapter 34).
- Most one-semester courses then cover at most one of the *Financial Modeling* sections. If we assume that in a typical university course, covering one chapter per week is an upper limit (and many chapters will require two weeks), then a typical course might concentrate on either corporate finance (Chapters 1–7),

portfolio models (Chapters 8–14), or options (Chapters 15–19). At a stretch, the instructor could perhaps throw in the shorter bond section (Chapters 20–23).

• I suggest that after the initial classes in a computer lab, the instructor move to a regular classroom. This enables the classroom emphasis to be on discussions of theory and implementation, with student homework concentrating on actual spreadsheets.

A major problem with a computer-based course is how to structure the final examination. Two solutions seem to work well. One alternative is to have students (whether alone or in teams) submit a final project; examples might be a corporate valuation if the course is based on Section I of the book, an event study for Section II, an option-based project for Section III, or the computation of a bond-expected return if the emphasis is on Section IV. A second alternative is to have students submit, by e-mail, a spreadsheet-based examination with severe time limits. One instructor using this book sends his class the final exam (a compendium of spreadsheet problems) at 9 in the morning and requires an e-mail with a spreadsheet answer by noon.

Acknowledgments

I thank a number of people who have made materially significant comments to this edition:

Meni Abudy, Zvika Afik, Javierma Bedoya, Lisa Bergé, Elizabeth Caulk, Sharon Garyn-Tal, Victor Lampe, Jongdoo Lee, Erez Levy, Warren Miller, Tal Mofkadi, Roger Myerson, Siddhartha Sarkar, Maxim Sharov, Permjit Singh, Sondre Aarseth Skjerven, Alexander Suhov, Kien-Quoc Van Pham, Chao Wang, Tim Wuu.

Finally, I would like to thank: my editor John Covell of MIT Press, Ellen Faran, the Director of MIT Press, and Nancy Benjamin and her editorial team at Books By Design. They have all been unfailingly helpful and patient.

Disclaimer

The materials in this book are intended for instructional and educational purposes only, to illustrate situations similar to those encountered in the real world. They may not apply directly to real-world situations. The author and MIT Press disclaim any responsibility for the consequences of implementation.

From the Preface to the Third Edition

The two previous editions of *Financial Modeling* have received a gratifyingly positive response from readers. The combination of a “cookbook,” mixing explanation and implementation using Excel has fulfilled a need in both the academic and the practitioner markets from readers who realize that the implementation of the finance basics typically studied in an introductory finance course requires another, more heavily computational and implementational, approach. Excel, the most widely used computational tool in finance, is a natural vehicle for deepening our understanding of the materials.

Acknowledgments

I want to start by thanking a group of wonderful editors: John Covell, Nancy Lombardi, Elizabeth Murry, Ellen Pope, and Peter Reinhart. My next thanks go to a dedicated group of colleagues who read the typescripts for *Financial Modeling*: Michael Chau, Jaksa Cvitanic, Arindam Bandopadhyaya, Richard Harris, Aurele Houngbedji, Iordanis Karagiannidis, Yvan Lengwiler, Nejat Seyhun, Gökçe Soydemir, David Y. Suk.

Many of the changes in this edition of *Financial Modeling* are due to the comments of readers, who have been assiduous in offering suggestions and improvements in the book. I follow a tradition started with the first two editions of *Financial Modeling* by acknowledging those readers whose comments have been incorporated into this edition:

Meni Abudy, Zvika Afik, Gordon Alexander, Apostol Bakalov, Naomi Belfer, David Biere, Vitaliy Bilyk, Oded Braverman, Roeland Brinkers, Craig Brody, Salvio Cardozo, Sharad Chaudhary, Israel Dac, Jeremy Darhansoff, Toon de Bakker, Govindvyas Dharwada, Davey Disatnik, Kevin P. Dowd, Brice Dupoyet, Cederik Engel, Orit Eshel, Yaara Geyra, Rana P. Ghosh, Bjarne Jensen, Marek Jochec, Milton Joseph, Erez Kamer, Saggi Katz, Emir Kiamilev, Brennan Lansing, Paul Ledin, Paul Legerer, Quinn Lewis, David Martin, Tom McCurdy, Tsahi Melamed, Tal Mofkadi, Geoffrey Morrisett, Sandip Mukherji, Max Nokhrin, Michael Oczkowski, David Pedersen, Mikael Petitjean, Georgio Questo, Alex Riahi, Arad Rostampour, Joseph Rubin, Andres Rubio, Ofir Shatz, Natalia Simakina, Ashutosh Singh, Permjit Singh, Gerald Strever, Shavkat Sultanbekov, Ilya Talman, Mel Tukman, Daniel Vainder, Guy Vishnia, Torben Voetmann, Chao Wang, James Ward, Roberto Wessels, Geva Yaniv, Richard Yeh, and Werner Zitzman.

Finally, I want to thank my very patient wife, Terry, who has maintained her own and my equilibrium through two books and a business school deanship in the past five years.

From the Preface to the Second Edition

The purpose of this book remains to provide a "cookbook" for implementing common financial models in Excel. This edition has been expanded by six additional chapters, covering financial calculations, cost of capital, value at risk (VaR), real options, early exercise boundaries, and term-structure modeling. There is also an additional technical chapter containing a potpourri of Excel hints.

I am indebted to a number of people (in addition to those mentioned in the previous preface) for help and suggestions: Andrew A. Adamovich, Alejandro Sanchez Arevalo, Yoni Aziz, Thierry Berger-Helmchen, Roman Weissman Bermann, Michael Giacomo Bertolino, John Bollinger, Enrico Camerini, Manuel Carrera, Roy Carson, John Carson, Lydia Cassorla, Philippe Charlier, Michael J. Clarke, Alvaro Cobo, Beni Daniel, Ismail Dawood, Ian Dickson, Moacyr Dutra, Hector Tassinari Eldridge, Shlomy Elias, Peng Eng, Jon Fantell, Erik Ferning, Raz Gilad, Nir Gluzman, Michael Gofman, Doron Greenberg, Phil Hamilton, Morten Helbak, Hitoshi Hibino, Foo Siat Hong, Marek Jochec, Russell W. Judson, Tiffani Kaliko, Boris Karasik, Rick Labs, Allen Lee, Paul Legerer, Guoli Li, Moti Marcus, Gershon Mensher, Tal Mofkadi, Stephen O'Neil, Steven Ong, Oren Ossad, Jackie Rosner, Steve Rubin, Dvir Sabah, Ori Salinger, Meir Shahar, Roger Shelor, David Siu, Maja Sliwinski, Bob Taggart, Maurry Tamarkin, Mun Hon Tham, Efrat Tolkowsky, Mel Tukman, Sandra van Balen, Michael Verhofen, Lia Wang, Roberto Wessels, Ethan Weyand, Ubbo Wiersema, Weiqin Xie, Ke Yang, Ken Yook, George Yuan, Khurshid Zaynutdinov, Ehud Ziegelman, and Eric Zivot. I also want to thank my editors, who again have been a great help: Nancy Lombardi, Peter Reinhart, Victoria Richardson, and Terry Vaughn.

From the Preface to the First Edition

Like its predecessor *Numerical Techniques in Finance,* the aim of this book is to present some important financial models and to show how they can be solved numerically and/or simulated using Excel. In this sense this is a finance "cookbook;" like any cookbook, it gives recipes with a list of ingredients and instructions for making and baking. As any cook knows, a recipe is just a starting point; having followed the recipe a number of times, you can think of your own variations and make the results suit your tastes and needs.

Financial Modeling covers standard financial models in the areas of corporate finance, financial statement simulation, portfolio problems, options, portfolio insurance, duration, and immunization. The aim in each case has been to explain clearly and concisely the implementation of the models using Excel. Very little theory is offered except where necessary to understand the numerical implementations.

While Excel is often not the tool to use for high-level, industrial-strength calculations (portfolios are an example), it is an excellent tool for understanding the computational intricacies involved in financial modeling. It is often the case that the fullest understanding of the models comes by calculating them, and Excel is one of the most accessible and powerful tools available for this purpose.

Along the way a lot of students, colleagues, and friends (these are nonexclusive categories) have helped me with advice and comments. In particular I would like to thank Olivier Blechner, Miryam Brand, Elizabeth Caulk, John Caulk, Benjamin Czaczkes, John Ferrari, John P. Flagler, Dan Fylstra, Kunihiko Higashi, Julia Hynes, Don Keim, Anthony Kim, Ken Kunimoto, Rick Labs, Adrian Lawson, Philippe Nore, Isidro Sanchez Alvarez, Nir Sharabi, Edwin Strayer, Robert Taggart, Mark Thaler, Terry Vaughn, and Xiaoge Zhou.

Finally, my thanks go to a wonderful set of editors: Nancy Lombardi, Peter Reinhart, Victoria Richardson, and Terry Vaughn.

0 Before All Else

0.1 Data Tables

Financial Modeling makes extensive use of data tables. I advise readers of the book to first make sure that they understand data tables (read Chapter 31, sections 1–5). Data tables are absolutely critical in the sensitivity analysis that is part of most financial models. They are a little bit complicated, but an invaluable addition to the modeling arsenal of the financial modeler.

In the remainder of this short chapter, I discuss **Getformula**.

0.2 What Is Getformula?

The Excel notebooks in *Financial Modeling*, fourth edition, contain a function called **Getformula** that aids in annotating your spreadsheets. In the example below, cell C5 shows the formula contained in cell B5; the formula in question computes the annual repayment of a loan of 165,000 for 7 years at 8%. Cell C5 contains the function =**Getformula(B5)**.

	A	B	C
2	Principal	165,000	
3	Interest	8%	
4	Term	7	<-- years
5	Annual payment	31,691.95	<-- =PMT(B3,B4,-B2)

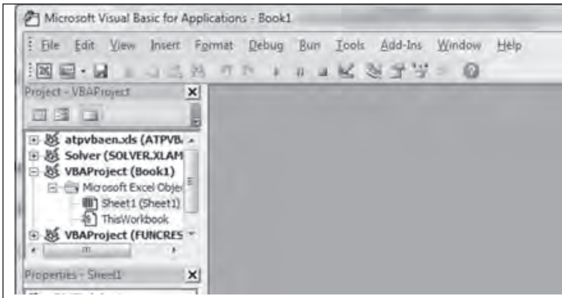
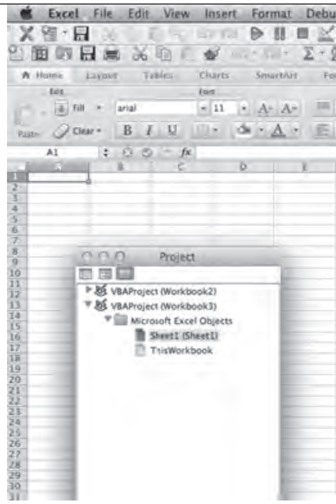
In this short chapter, we describe how to add this formula to your Excel notebook. Mac users: This works only in Excel 2011.

0.3 How to Put Getformula into Your Excel Notebook

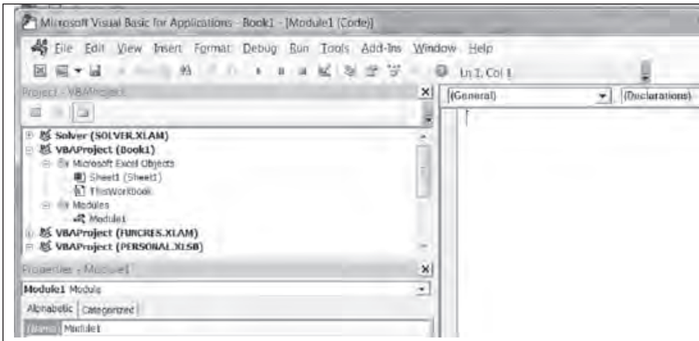
1. Open the Excel workbook in which you want the formula to work.
2. Open the VBA editor:

- On Windows computers: Press [Alt] + F11.
- On Mac (Excel 2011): Choose **Tools|Macro|Visual Basic Editor**

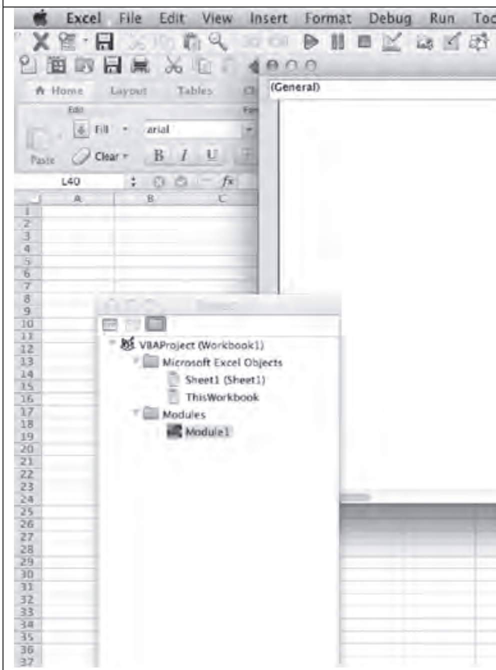
3. This will open the VBA editor.

	
Windows screen	Mac screen

4. Select **Insert|Module** at the top of the screen.



← Windows screen



← Mac screen

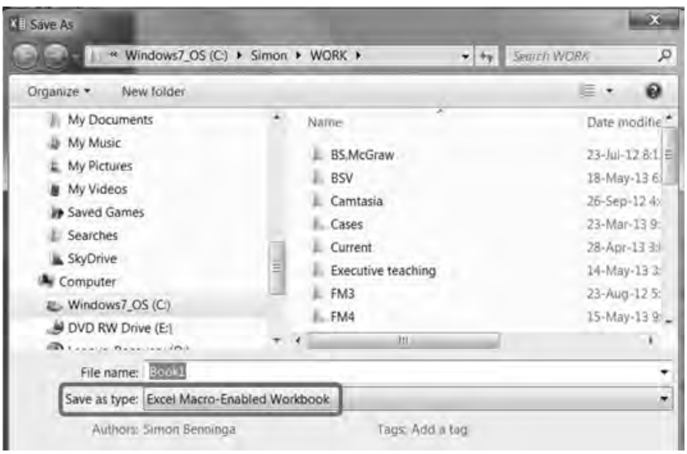
5. Now insert the following text into the Module window (where it says **General**). Just copy/paste the text below.

```
‘8/5/2006 Thanks to Maja Sliwinski and
‘Beni Czaczkes
Function getformula(r As Range) As String
     Application.Volatile
     If r.HasArray Then
     getformula = “<-- “ & _
       “ {“ & r.FormulaArray & “}”
     Else
     getformula = “<-- “ & _
       “ “ & r.FormulaArray
     End If
End Function
```

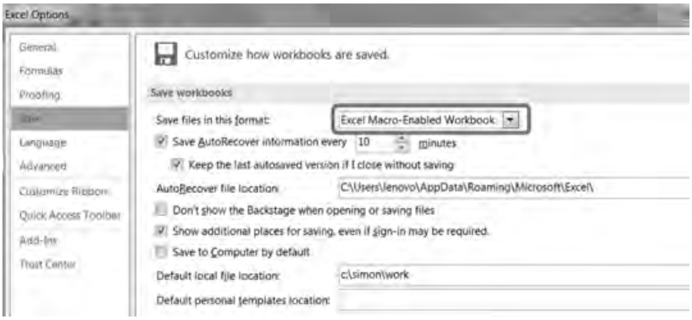
In Windows, close the VBA window (no need to save). On the Mac, just continue to work on the spreadsheet. The formula is now part of the spreadsheet and will be saved along with it.

0.4 Saving the Excel Workbook: Windows

To save the notebook with the **Getformula** macro in VBA, you will have to save it as a **Macro-enabled workbook**.

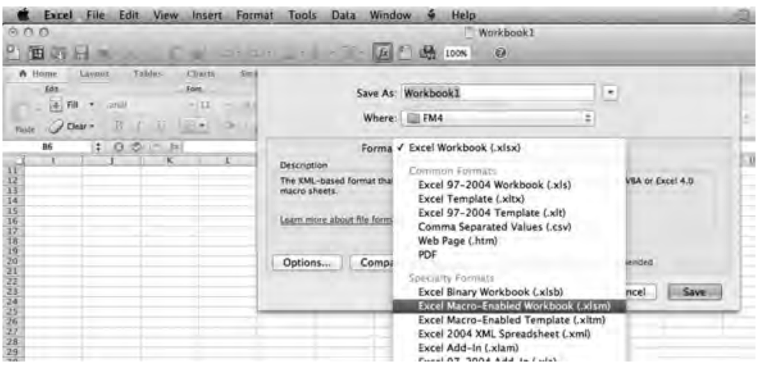


Macro-enabled workbooks have the extension .xlsm, whereas regular Excel workbooks have the extension .xlsx. Your users will never know the difference. We have changed our Excel settings (**File|Options|Save**) to make the Macro-enabled workbook our default:



0.5 Saving the Excel Workbook: Mac

The Mac screen for saving as a Macro-enabled workbook looks like this:



0.6 Do You Have to Put Getformula into Each Excel Workbook?

The short answer is "yes." You could create an add-in to Excel (see Chapter 39) that contains **Getformula**, but this will make it more difficult for you to share your workbooks. We prefer to put **Getformula** in each new spreadsheet we create.

0.7 A Shortcut to Use Getformula

Once you have put **Getformula** into your Excel workbook, you will have to use it! Ninety percent of our uses of this function point to the cell to the left of the formula itself:

	A	B	C	D	E
1		QUESTION 2			
2	Interest rate	11%			
3					
4	Year	Asset1	Asset2	Asset3	
5	1	1,000	0	0	
6	2	1,000	0	0	
7	3	1,000	1,700	0	
8	4	1,000	1,700	0	The cell below contains
9	5	1,000	1,700	3,000	Getformula(D13)
10	6	1,000	0	4,000	
11	7	1,000	0	5,000	
12					
13	Value	4,712	3,372	6,327	<-- =NPV($B$2,D5:D11)
14					
15					

We've put a short macro into our **Personal workbook** that automates this procedure. The remainder of this section describes how to automate the **Getformula** procedure.

Automating the Procedure

We want to automate this procedure of putting **Getformula** into a cell:

- Turn it into a macro.
- Attach a key sequence (in our case, [Ctrl] + t) to the macro.
- Make the macro and key sequence available in your Excel spreadsheets.

We will save the macro to our **Personal.xlsb** file. This file activates each time you start Excel. It's yours only—other readers of your spreadsheets won't see it. Below we describe the steps, for both Windows and the Mac.

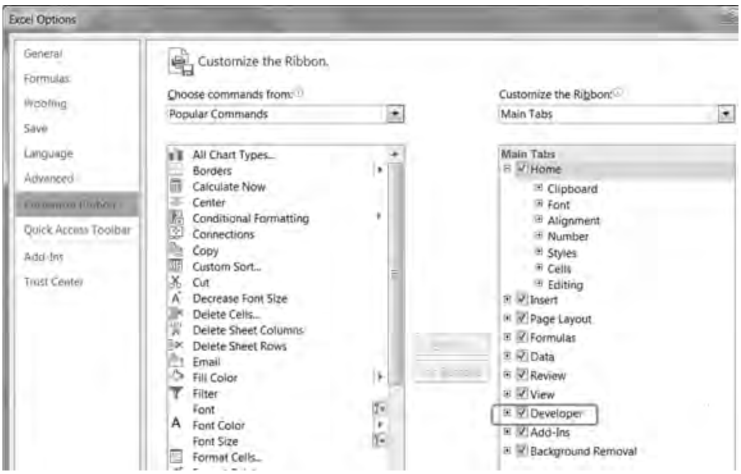
0.8 Recording Getformula: The Windows Case

Here are the steps to recording the macro in Windows:

- Activate the **Developer** tab on the menu bar.
- Use **Record Macro** to save a macro as a personal notebook.

Activate the Developer Tab

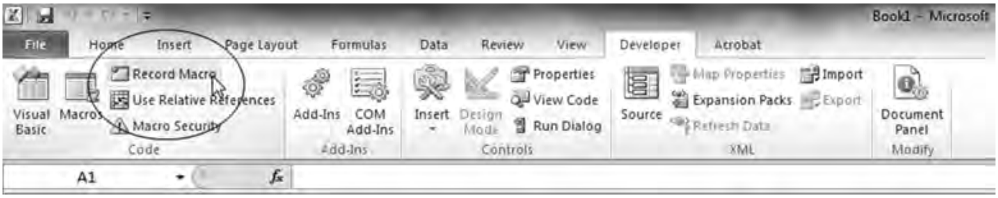
Go to **File|Options|Customize Ribbon** and activate the **Developer** tab as shown below:



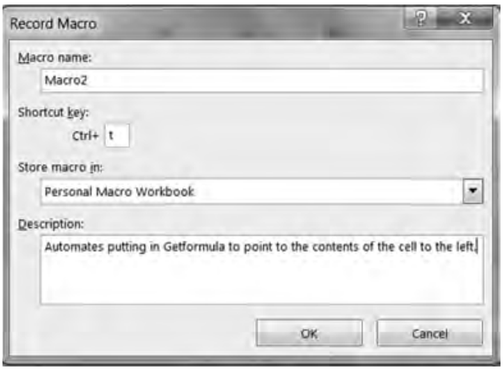
Use Record Macro

The Developer tab allows you to record a macro and save it as part of the Personal.xlsb notebook. We will illustrate with the copy as picture feature.

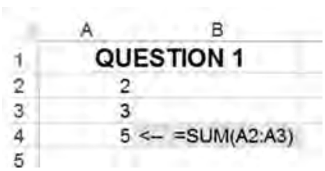
1. Open a blank Excel notebook and click on the **Developer** tab and then on **Record Macro**:



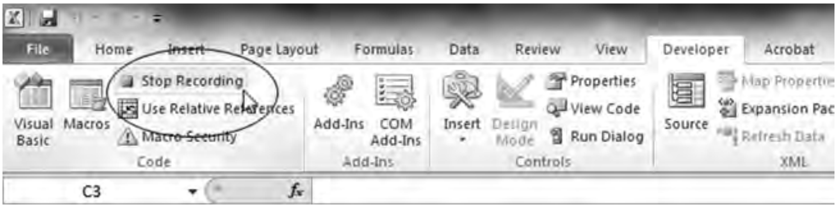
Excel will ask for details of the recording. Here's what I wrote. We will save this as a **Personal Macro Workbook** and then use the shortcut [Ctrl] + t:



2. Now go to your spreadsheet and use **Getformula**, pointing to the cell to the left of where you want **Getformula** to appear. In the spreadsheet below, we have typed =Getformula(A3) into cell B4:



3. Go back to the **Developer** tab and stop the recording:



4. Close down Excel. Excel will ask you if you want to save the Personal workbook. The answer is, of course, positive:

This creates the following file ("simon benninga" is of course my user name on my computer—you will substitute your user name).

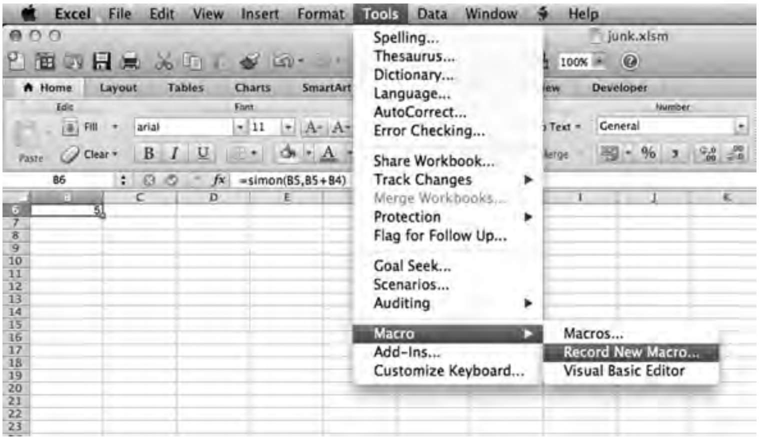
C:\Users\simon benninga\AppData\Roaming\Microsoft\Excel\XLSTART\
PERSONAL.XLSB

Using the Macro

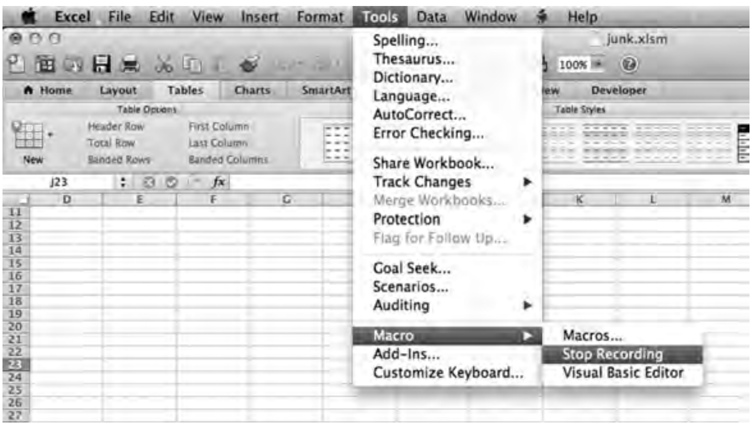
From now on, whenever you open a file on *your computer*, you can use [Ctrl] + t to copy a region as a picture. Cool!

0.9 Recording Getformula: The Mac Case

To record a macro in Excel, use **Tools|Macro|Record New Macro**:



To stop the recording:



As in the Windows case, you will be prompted to name the macro and assign it a control key sequence. When you save the spreadsheet, you will be asked if you want to save the Personal Workbook. Done!

I CORPORATE FINANCE AND VALUATION

The seven chapters that open *Financial Modeling* cover basic problems and techniques in corporate finance. Chapter 1 is an introduction to basic financial calculations using Excel. Chapter 2 is a short overview of various valuation methods as applied to corporations. The cost of capital, discussed in Chapter 3, is the rate at which corporate cash flows are discounted to arrive at enterprise value. Calculating this rate is not trivial and involves a combination of theoretical models and numerical computation, both discussed in the chapter.

Chapters 4 and 5 discuss two primary valuation methods. Chapter 4 shows how to derive the free cash flows required for valuation from the consolidated statement of cash flows. Chapter 5 shows how to build pro forma models, which simulate the corporate income statement and balance sheets. Pro forma models are at the heart of many corporate finance applications, including business plans, credit analyses, and valuations. The models require a mixture of finance, accounting, and Excel. Chapter 6 develops a pro forma model to value Caterpillar Corporation. The example we develop is typical of an exercise which accompanies many merger and acquisition valuations.

Section I closes with a discussion of the financial analysis of leasing in Chapter 7.

1 Basic Financial Calculations

1.1 Overview

This chapter aims to give you some finance basics and their Excel implementation. If you have had a good introductory course in finance, this chapter is likely to be at best a refresher.[1]

This chapter covers:

- Net present value (NPV)
- Internal rate of return (IRR)
- Payment schedules and loan tables
- Future value
- Pension and accumulation problems
- Continuously compounded interest
- Time-dated cash flows (Excel functions **XNPV** and **XIRR**)

Almost all financial problems are centered on finding the *value today* of a series of *cash receipts over time*. The cash receipts (or cash flows, as we will call them) may be certain or uncertain. The *present value* of a cash flow CF_t anticipated to be received at time t is $\frac{CF_t}{(1+r)^t}$. The numerator of this expression is usually understood to be the *expected time t cash flow*, and the discount rate r in the denominator is adjusted for the riskiness of this expected cash flow—the higher the risk, the higher the discount rate.

The basic concept in present value calculations is the concept of *opportunity cost*. Opportunity cost is the return which would be required of an investment to make it a viable alternative to other, similar investments. In the financial literature there are many synonyms for opportunity cost, among them: discount rate, cost of capital, and interest rate. When applied to risky cash flows, we will sometimes call the opportunity cost the risk-adjusted discount rate (RADR) or the weighted average cost of capital (WACC). It goes without saying that this discount rate should be risk-adjusted, and much of the standard finance literature discusses how to do this. As illustrated below, when we calculate the net present value, we use the investment's opportunity cost as a discount rate. When we calculate the internal rate of return, we compare the calculated return to the investment's opportunity cost to judge its value.

1. In my book *Principles of Finance with Excel* (Oxford University Press, 2nd edition, 2008) I have discussed many basic Excel/finance topics at greater length.

1.2 Present Value and Net Present Value

Both of these concepts are related to the value *today* of a set of future anticipated cash flows. As an example, suppose we are valuing an investment which promises $100 per year at the end of this and the next 4 years. We suppose that these cash flows are risk free: There is no doubt that this series of 5 payments of $100 each will actually be paid. If a bank pays an annual interest rate of 10% on a 5-year deposit, then this 10% is the investment's opportunity cost, the alternative benchmark return to which we want to compare the investment. We can calculate the value of the investment by discounting its cash flows using this opportunity cost as a discount rate:

	A	B	C	D
1	**COMPUTING THE PRESENT VALUE**			
2	Discount rate	10%		
3				
4	**Year**	**Cash flow**	**Present value**	
5	1	100	90.9091	<-- =B5/(1+$B$2)^A5
6	2	100	82.6446	<-- =B6/(1+$B$2)^A6
7	3	100	75.1315	<-- =B7/(1+$B$2)^A7
8	4	100	68.3013	<-- =B8/(1+$B$2)^A8
9	5	100	62.0921	<-- =B9/(1+$B$2)^A9
10				
11	Net present value			
12	Summing cells C5:C9	379.08	<-- =SUM(C5:C9)	
13	Using Excel's NPV function	379.08	<-- =NPV(B2,B5:B9)	
14	Using Excel's PV function	379.08	<-- =PV(B2,5,-100)	

The **present value**, 379.08, is the *value today* of the investment. In a competitive market, the present value should correspond to the market price of the cash flows. The spreadsheet illustrates three ways of obtaining this value:

- Summing the individual present values in cells C5:C9. To simplify the copying, note the use of "^" to represent the power and the use of both the relative and absolute references; for example: =B5/(1+$B$2)^A5 in cell C5.
- Using the Excel **NPV** function. As we show on the next page, Excel's **NPV** function is unfortunately misnamed—it actually computes the present value and not the net present value.

• Using the Excel **PV** function. This function computes the present value of a series of constant payments. **PV(B2,5,-100)** is the present value of 5 payments of 100 each at the discount rate in cell B2. The **PV** function returns a negative value for positive cash flows; to prevent this unfortunate occurrence, we have made the cash flows negative.[2]

The Difference Between Excel's PV and NPV Functions

The above spreadsheet may leave the misimpression that **PV** and **NPV** perform exactly the same computation. But this is not true—whereas **NPV** can handle any series of cash flows, **PV** can handle only constant cash flows:

	A	B	C	D
1	**COMPUTING THE PRESENT VALUE** **In this example the cash flows are not equal** **Either discount each cash flow separately or use Excel's NPV function** **Excel's PV doesn't work for this case**			
2	Discount rate	10%		
3				
4	**Year**	**Cash flow**	**Present value**	**Present value of each cash flow**
5	1	100	90.9091	<-- =B5/(1+$B$2)^A5
6	2	200	165.2893	<-- =B6/(1+$B$2)^A6
7	3	300	225.3944	<-- =B7/(1+$B$2)^A7
8	4	400	273.2054	<-- =B8/(1+$B$2)^A8
9	5	500	310.4607	<-- =B9/(1+$B$2)^A9
10				
11	Net present value			
12	Summing cells C5:C9	1065.26	<-- =SUM(C5:C9)	
13	Using Excel's NPV function	1065.26	<-- =NPV(B2,B5:B9)	

Excel's NPV Function Is Misnamed!

In standard finance terminology, the *present value* of a series of cash flows is the value today of the future cash flows:

$$Present\ value = \sum_{t=1}^{N} \frac{CF_t}{(1+r)^t}$$

2. This strange property—returning negative values for positive cash flows—is shared by a number of otherwise impeccable Excel functions such as PMT and PV. The somewhat convoluted logic which led Microsoft to write these functions this way is not worth explaining.

The *net present value* is the present value minus the cost of acquiring the asset (the cash flow at time zero):

$$Net\ present\ value = \sum_{t=0}^{N} \frac{CF_t}{(1+r)^t} = \underbrace{CF_0}_{\substack{\uparrow \\ \text{In many cases} \\ CF_0\ <\ 0\text{, meaning} \\ \text{that it represents the} \\ \text{price paid for the asset.}}} + \underbrace{\sum_{t=1}^{N} \frac{CF_t}{(1+r)^t}}_{\substack{\uparrow \\ \text{This is the present} \\ \text{value, given by Excel's} \\ \text{NPV function.}}}$$

Excel's language about discounted cash flows differs somewhat from the standard finance nomenclature. To calculate the finance *net present value* of a series of cash flows using Excel, we have to calculate the *present value* of the future cash flows (using the Excel **NPV** function), taking into account the time-zero cash flow (this is often the cost of the asset in question).

The Net Present Value, NPV

Suppose that the above investment is sold for $400. Clearly it would not be worth its purchase price, since—given the alternative return (discount rate) of 10%—the investment is worth only $379.08. The *net present value* (NPV) is the applicable concept here. Denoting by r the discount rate applicable to the investment, the NPV is calculated as follows:

$$NPV = CF_0 + \sum_{t=1}^{N} \frac{CF_t}{(1+r)^t}$$

where CF_t is the investment's cash flow at time t and CF_0 is today's cash flow.

Suppose, for example, that the series of 5 cash flows of $100 is sold for $250. Then, as shown below, the NPV = 129.08.

	A	B	C	D
1	**COMPUTING THE NET PRESENT VALUE**			
2	Discount rate	10%		
3				
4	**Year**	**Cash flow**	**Present value**	
5	0	-250	-250.00	<-- =B5/(1+$B$2)^A5
6	1	100	90.91	<-- =B6/(1+$B$2)^A6
7	2	100	82.64	<-- =B7/(1+$B$2)^A7
8	3	100	75.13	<-- =B8/(1+$B$2)^A8
9	4	100	68.30	<-- =B9/(1+$B$2)^A9
10	5	100	62.09	<-- =B10/(1+$B$2)^A10
11				
12	Net present value			
13	Summing cells C5:C10	129.08	<-- =SUM(C5:C10)	
14	Using Excel's NPV function	129.08	<-- =B5+NPV(B2,B6:B10)	

The NPV represents the *wealth increment* of the purchaser of the cash flows. If you buy the series of 5 cash flows of 100 for 250, then you have gained 129.08 in wealth today. In a competitive market the NPV of a series of cash flows ought to be zero: Since the present value should correspond to the market price of the cash flows, the NPV should be zero. In other words, the market price of our 5 cash flows of 100 ought—in a competitive market, assuming that 10% is the correct risk-adjusted discount rate—be 379.08.

The Present Value of an Annuity—Some Useful Formulas[3]

An *annuity* is a security which pays a constant sum in each period in the future. Annuities may have a finite or infinite series of payments. If the annuity is finite, and the appropriate discount rate is r, then the value today of the annuity is its present value:

$$PV\ of\ finite\ annuity = \frac{C}{1+r} + \frac{C}{(1+r)^2} + \ldots + \frac{C}{(1+r)^n} = C\left(\frac{1-\frac{1}{(1+r)^n}}{r}\right)$$

If the annuity promises an infinite series of constant future payments, then this formula reduces to:

$$PV\ of\ infinite\ annuity = \frac{C}{1+r} + \frac{C}{(1+r)^2} + \ldots = \frac{C}{r}$$

Both of these formulas can be computed with Excel. Below we compute the value of a finite annuity in three ways: using the formula (cell B6), using Excel's **PV** function (cell B7), and using Excel's **NPV** function:

3. All the formulas in this subsection depend on some well-known but oft-forgotten high school algebra.

	A	B	C
1	**COMPUTING THE VALUE OF A FINITE ANNUITY**		
2	Periodic payment, C	1,000	
3	Number of future periods paid, n	5	
4	Discount rate, r	12%	
5	Present value of annuity		
6	Using formula	3,604.78	<-- =B2*(1-1/(1+B4)^B3)/B4
7	Using Excel's **PV** function	3,604.78	<-- =PV(B4,B3,-B2)
8			
9	**Period**	**Annuity payment**	
10	1	1,000.00	<-- =B2
11	2	1,000.00	
12	3	1,000.00	
13	4	1,000.00	
14	5	1,000.00	
15			
16	Present value using Excel's **NPV** function	3,604.78	<-- =NPV(B4,B10:B14)

Computing the value of an infinite annuity is even simpler:

	A	B	C
1	**COMPUTING THE VALUE OF AN INFINITE ANNUITY**		
2	Periodic payment, C	1,000	
3	Discount rate, r	12%	
4	Present value of annuity	8,333.33	<-- =B2/B3

The Value of a Growing Annuity

A *growing annuity* pays out a sum C, which grows at a periodic growth rate g. If the annuity is finite, its value today is given by:

$$PV\ of\ finite\ growing\ annuity = \frac{C}{1+r} + \frac{C(1+g)}{(1+r)^2} + \frac{C(1+g)^2}{(1+r)^3} + \ldots + \frac{C(1+g)^{n-1}}{(1+r)^n} = \frac{C\left(1-\left(\frac{1+g}{1+r}\right)^n\right)}{r-g}$$

Taking this formula and letting $n \to \infty$, we can compute the value of *infinite growing annuity*:

$$PV\ of\ infinite\ growing\ annuity = \frac{C}{1+r} + \frac{C(1+g)}{(1+r)^2} + \frac{C(1+g)^2}{(1+r)^3} + \dots$$
$$= \frac{C}{r-g}, \text{ provided} \left|\frac{1+g}{1+r}\right| < 1$$

These formulas can easily be implemented in Excel. Below we compute the value of a finite growing annuity using the formula above and using Excel's **NPV** function:

	A	B	C
1	**COMPUTING THE VALUE OF A FINITE GROWING ANNUITY**		
2	First payment, C	1,000	
3	Growth rate of payments, g	6%	
4	Number of future periods paid, n	5	
5	Discount rate, r	12%	
6	Present value of annuity		
7	Using formula	4,010.91	<-- =B2*(1-((1+B3)/(1+B5))^B4)/(B5-B3
8			
9	**Period**	**Annuity payment**	
10	1	1,000.00	<-- =B2
11	2	1,060.00	<-- =$B$2*(1+$B$3)^(A11-1)
12	3	1,123.60	<-- =$B$2*(1+$B$3)^(A12-1)
13	4	1,191.02	<-- =$B$2*(1+$B$3)^(A13-1)
14	5	1,262.48	<-- =$B$2*(1+$B$3)^(A14-1)
15			
16	Present value using Excel's **NPV** function	4,010.91	<-- =NPV(B5,B10:B14)

When the growing annuity has an infinite life:

	A	B	C
1	**COMPUTING THE VALUE OF AN INFINITE GROWING ANNUITY**		
2	Periodic payment, C	1,000	<-- Starting at date 1
3	Growth rate of payments, g	6%	
4	Discount rate, r	12%	
5	Present value of annuity	16,666.67	<-- =B2/(B4-B3)

The Gordon Formula

The Gordon formula values a stock by discounting its future anticipated dividends at the cost of equity r_E. Letting P_0 be the current stock price, Div_0 the current dividend, and g the growth rate of future dividends, then

$$P_0 = \sum_{t=1}^{\infty} \frac{Div_0 (1+g)^t}{(1+r_E)^t} = \frac{Div_0 (1+g)}{r_E - g}$$

Using the formula for an infinite growing annuity, we can write this as

$$P_0 = \frac{Div_0 (1+g)}{r_E - g} \text{ provided } |g| < |r_E|$$

Inverting this formula shows that

$$r_E = \frac{Div_0 (1+g)}{P_0} + g$$

The Gordon formula is used in *Financial Modeling*'s Chapters 2, 4, 5, and 6 to model the firm's terminal value and in Chapter 3 to model the firm's cost of equity r_E .

1.3 The Internal Rate of Return (IRR) and Loan Tables

The *internal rate of return* (IRR) is defined as the compound rate of return r which makes the NPV equal to zero:

$$CF_0 + \sum_{t=1}^{N} \frac{CF_t}{(1+r)^t} = 0$$

To illustrate, consider the example given in rows 2–10 below: A project costing 800 in year zero returns a variable series of cash flows at the end of years 1–5. The IRR of the project (cell B10) is 22.16%:

	A	B	C
1	**INTERNAL RATE OF RETURN**		
2	**Year**	**Cash flow**	
3	0	-800	
4	1	200	
5	2	250	
6	3	300	
7	4	350	
8	5	400	
9			
10	Internal rate of return	22.16%	<-- =IRR(B3:B8)

Note that the Excel **IRR** function includes as arguments *all* of the cash flows of the investment, including the first—in this case negative—cash flow of –800.

Determining the IRR by Trial and Error

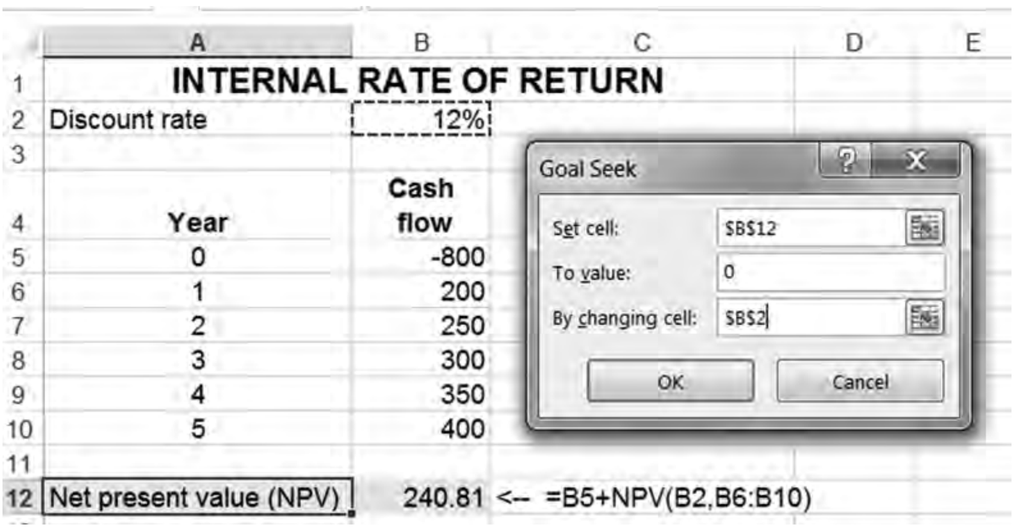
There is no simple formula to compute the IRR. Excel's **IRR** function uses trial and error, which can be simulated by using trial and error in a spreadsheet as illustrated below:

	A	B	C
1	**INTERNAL RATE OF RETURN**		
2	Discount rate	12%	
3			
4	**Year**	**Cash flow**	
5	0	-800	
6	1	200	
7	2	250	
8	3	300	
9	4	350	
10	5	400	
11			
12	Net present value (NPV)	240.81	<-- =B5+NPV(B2,B6:B10)

By playing with the discount rate or by using Excel's **Goal Seek** (found under **Data|What-if analysis**, see Chapter 31), we can determine that at 22.16% the NPV in cell B12 is zero:

	A	B	C
1	**INTERNAL RATE OF RETURN**		
2	Discount rate	22.16%	
3			
4	**Year**	**Cash flow**	
5	0	-800	
6	1	200	
7	2	250	
8	3	300	
9	4	350	
10	5	400	
11			
12	Net present value (NPV)	0.00	<-- =B5+NPV(B2,B6:B10)

Here's the way the **Goal Seek** screen looked before we got the correct answer:



	A	B	C
1	**INTERNAL RATE OF RETURN**		
2	Discount rate	12%	
3			
4	**Year**	**Cash flow**	
5	0	-800	
6	1	200	
7	2	250	
8	3	300	
9	4	350	
10	5	400	
11			
12	Net present value (NPV)	240.81	<-- =B5+NPV(B2,B6:B10)

Loan Tables and the Internal Rate of Return

The IRR is the compound *rate of return paid by the investment*. To understand this fully, it helps to make a *loan table*, which shows the division of the investment's cash flows between investment income and the return of the investment principal:

	A	B	C	D	E	F
1	**INTERNAL RATE OF RETURN**					
2	**Year**	**Cash flow**				
3	0	-800				
4	1	200				
5	2	250				
6	3	300				
7	4	350				
8	5	400				
9						
10	Internal rate of return	22.16%	<-- =IRR(B3:B8)			
11						
12	**USING THE IRR IN A LOAN TABLE**					
13	=-B3		=$B$10*B15	**Division of cash flow between investment income and return of principal**		
14	**Year**	**Investment at beginning of year**	**Cash flow at end of year**	**Income**	**Return of principal**	
15	1	800.00	200.00	177.28	22.72	<-- =C15-D15
16	2	777.28	250.00	172.25	77.75	
17	3	699.53	300.00	155.02	144.98	
18	4	554.55	350.00	122.89	227.11	
19	5	327.44	400.00	72.56	327.44	
20	6	0.00				
21	=B15-E15					
22			The remaining investment principal in the year after the last cash flow is zero, indicating that all the principal has been repaid.			
23						
24						
25						
26						

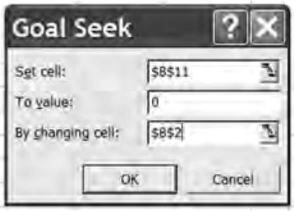
The *loan table* divides each of the cash flows of the asset into an income component and a return-of-principal component. The income component at the end of each year is IRR times the principal balance at the beginning of that year. Notice that the principal at the beginning of the last year (327.44 in the example) exactly equals the return of principal at the end of that year.

We can use the loan table to find the internal rate of return. Consider an investment costing 1,000 today that pays off the cash flows indicated below at the end of years 1, 2, … 5. At a rate of 15% (cell B2), the principal at the beginning of year 6 is negative, indicating that too little has been paid out in income. Thus the IRR must be larger than 15%:

	A	B	C	D	E	F
1	**USING A LOAN TABLE TO FIND THE IRR**					
2	IRR?	15.00%				
3						
4				**Division of cash flow between investment income and return of principal**		
5	**Year**	**Principal at beginning of year**	**Cash flow at end of year**	**Income**	**Principal**	
6	1	1,000.00	300	150.00	150.00	<-- =C6-D6
7	2	850.00	200	127.50	72.50	
8	3	777.50	150	116.63	33.38	
9	4	744.13	600	111.62	488.38	
10	5	255.74	900	38.36	861.64	
11	6	-605.89				
12			=$B$2*B6			
13	=B6-E6					

If the interest rate in cell B3 is indeed the IRR, then cell B11 should be 0. We can use Excel's **Goal Seek** (found under **Data|What-if analysis**) to calculate the IRR:

	A	B	C	D	E	F
1	**USING A LOAN TABLE TO FIND THE IRR**					
2	IRR?	15.00%				
3						
4				**Division of cash flow between investment income and return of principal**		
5	**Year**	**Principal at beginning of year**	**Cash flow at end of year**	**Income**	**Principal**	
6	1	1,000.00	300	150.00	150.00	<-- =C6-D6
7	2	850.00	200	127.50	72.50	
8	3	777.50	150	116.63	33.38	
9	4	744.13	600	111.62	488.38	
10	5	255.74	900	38.36	861.64	
11	6	-605.89				
12			=$B$2*B6			
13	=B6-E6					



As shown below, the IRR is 24.44%:

	A	B	C	D	E	F
1	**USING A LOAN TABLE TO FIND THE IRR**					
2	IRR?	24.44%				
3						
4				**Division of cash flow between investment income and return of principal**		
5	**Year**	**Principal at beginning of year**	**Cash flow at end of year**	**Income**	**Principal**	
6	1	1,000.00	300	244.36	55.64	<-- =C6-D6
7	2	944.36	200	230.76	-30.76	
8	3	975.13	150	238.28	-88.28	
9	4	1,063.41	600	259.86	340.14	
10	5	723.26	900	176.74	723.26	
11	6	0.00				
12			=$B$2*B6			
13	=B6-E6					

The loan table is an effective illustration that the IRR is the interest rate that pays off an investment over its term. Of course, we could have simplified life by just using the **IRR** function:

	A	B	C	D
15	**Direct calculation of IRR**			
16	**Year**	**Cash flow**		
17	0	-1,000		
18	1	300		
19	2	200		
20	3	150		
21	4	600		
22	5	900		
23				
24	IRR	24.44%	<-- =IRR(B17:B22)	

Excel's Rate Function

Excel's **Rate** function computes the IRR of a series of constant future payments. In the example below, we pay $1,000 today for an annual payment of $100 for the next 30 years. **Rate** shows that the IRR is 9.307%:

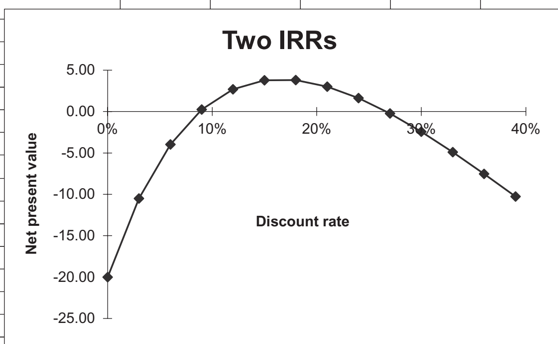
	A	B	C
1	**USING EXCEL'S RATE FUNCTION TO COMPUTE THE IRR**		
2	Initial investment	1,000	
3	Periodic cash flow	100	
4	Number of payments	30	
5	IRR	9.307%	<-- =RATE(B4,B3,-B2)

Note: **Rate** works much like **PMT** and **PV** discussed elsewhere in this chapter; it requires a sign change between the initial investment and the periodic cash flow (note that we have used –B2 in cell B5). It also has switches to allow for payments which start today and payments which start one period from now (not shown in the above example).

1.4 Multiple Internal Rates of Return

Sometimes a series of cash flows has more than one IRR. In the next example we can tell that the cash flows in cells B6:B11 have two IRRs, since the NPV graph crosses the x-axis twice:

	A	B	C	D	E	F	G	H	I
1	**MULTIPLE INTERNAL RATES OF RETURN**								
2	Discount rate	6%							
3	NPV	-3.99	<-- =NPV(B2,B7:B11)+B6				**DATA TABLE**		
4							**Discount rate**	**NPV**	
5	**Year**	**Cash flow**						-3.99	Table header, <-- =B3
6	0	-145					0%	-20.00	
7	1	100					3%	-10.51	
8	2	100					6%	-3.99	
9	3	100					9%	0.24	
10	4	100					12%	2.69	
11	5	-275					15%	3.77	
12							18%	3.80	
13							21%	3.02	
14							24%	1.62	
15							27%	-0.24	
16							30%	-2.44	
17							33%	-4.90	
18							36%	-7.53	
19							39%	-10.27	
20									
21							**Note**: For a discussion of how to create data tables in Excel, see Chapter 31.		
22									
23									
24									
25									
26									
27									
28									
29	**Identifying the two IRRs**								
30	First IRR	8.78%	<-- =IRR(B6:B11,0)						
31	Second IRR	26.65%	<-- =IRR(B6:B11,0.3)						



Excel's **IRR** function allows us to add an extra argument which will help us find both IRRs. Instead of writing **=IRR(B6:B11)**, we write **=IRR(B6:B11,guess)**. The argument **guess** is a starting point for the algorithm which Excel uses to find the IRR; by adjusting the **guess**, we can identify both the IRRs. Cells B30 and B31 give an illustration.

There are two things to note about this procedure:

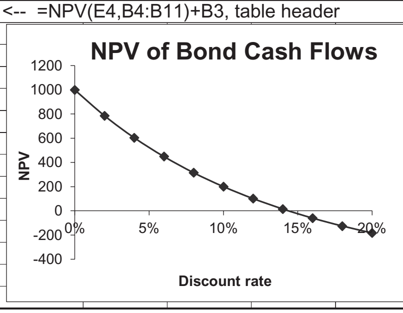
• The argument **guess** merely has to be close to the IRR; it is not unique. For example, by setting the guesses equal to 0.1 and 0.5, we will still get the same IRRs:

	A	B	C	D
29	**Identifying the two IRRs**			
30	First IRR	8.78%	<-- =IRR(B6:B11,0.1)	
31	Second IRR	26.65%	<-- =IRR(B6:B11,0.5)	

• In order to identify the number and the approximate value of the IRRs, it helps greatly to graph (as we did above) the NPV of the investment as a function of various discount rates. The internal rates of return are then the points where the graph crosses the x-axis, and the approximate location of these points should be used as the guesses in the IRR function.[4]

From a purely technical point of view, a set of cash flows can have multiple IRRs only if it has at least two changes of sign. Many typical cash flows have only one change of sign. Consider, for example, the cash flows from purchasing a bond having a 10% coupon, a face value of $1,000, and 8 more years to maturity. If the current market price of the bond is $800, then the stream of cash flows changes signs only once (from negative in year 0 to positive in years 1–8). Thus there is only one IRR:

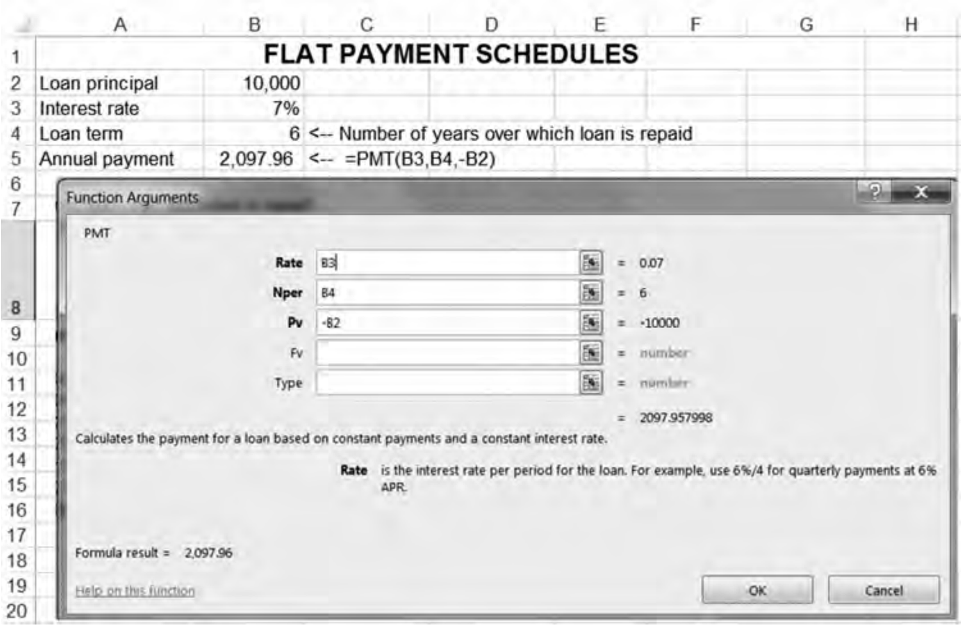
	A	B	C	D	E	F	G	H	I	J	K
1	**BOND CASH FLOWS: NPV CROSSES X-AXIS ONLY ONCE, SO THERE IS ONLY ONE IRR**										
2	**Year**	**Cash flow**			**Data table:** Effect of						
3	0	-800			discount rate on NPV						
4	1	100				1,000.00	<-- =NPV(E4,B4:B11)+B3, table header				
5	2	100			0%	1,000.00					
6	3	100			2%	786.04					
7	4	100			4%	603.96					
8	5	100			6%	448.39					
9	6	100			8%	314.93					
10	7	100			10%	200.00					
11	8	1,100			12%	100.65					
12					14%	14.45					
13	IRR	14.36%	<-- =IRR(B3:B11)		16%	-60.62					
14					18%	-126.21					
15					20%	-183.72					
16											
17											



4. If you don't put in a guess (as we did in the previous section), Excel defaults to a guess of 0.1. Thus, in the current example, IRR(B34:B39) will return 8.78%.

1.5 Flat Payment Schedules

Another common problem is to compute a "flat" repayment for a loan. For example: You take a loan for $10,000 at an interest rate of 7% per year. The bank wants you to make a series of payments which will pay off the loan and the interest over 10 years. We can use Excel's **PMT** function to determine how much each annual payment should be:



Notice that we have put "PV"—Excel's nomenclature for the initial loan principal—with a minus sign. As discussed above, if we do not do this Excel returns a negative payment (a minor irritant). You can confirm that the answer of 2,097.96 is correct by creating a loan table:

	A	B	C	D	E	F	G
1	**FLAT PAYMENT SCHEDULES**						
2	Loan principal	10,000					
3	Interest rate	7%					
4	Loan term	6	<-- Number of years over which loan is repaid				
5	Annual payment	2,097.96	<-- =PMT(B3,B4,-B2)				
6							
7					**Split payment into:**		=$B$3*C9
8		**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest**	**Return of principal**	
9		1	10,000.00	2,097.96	700.00	1,397.96	
10		2	8,602.04	2,097.96	602.14	1,495.82	=D9-E9
11		3	7,106.23	2,097.96	497.44	1,600.52	
12	=C9-F9	4	5,505.70	2,097.96	385.40	1,712.56	
13		5	3,793.15	2,097.96	265.52	1,832.44	
14		6	1,960.71	2,097.96	137.25	1,960.71	
15		7	0.00				

The zero in cell C15 indicates that the loan is fully repaid over its term of 6 years. You can easily confirm that the present value of the payments over the 6 years is the initial principal of 10,000.

1.6 Future Values and Applications

We start with a triviality. Suppose you deposit 1,000 in an account today, leaving it there for 10 years. Suppose the account draws annual interest of 10%. How much will you have at the end of 10 years? The answer, as shown in the following spreadsheet, is 2,593.74:

	A	B	C	D	E
1	**SIMPLE FUTURE VALUE**				
2	Interest	10%			
3					
4	**Year**	**Account balance, beginning of year**	**Interest earned during year**	**Total in account, end year**	
5	1	1,000.00	100.00	1,100.00	<-- =C5+B5
6	2	1,100.00	110.00	1,210.00	<-- =C6+B6
7	3	1,210.00	121.00	1,331.00	
8	4	1,331.00	133.10	1,464.10	
9	5	1,464.10	146.41	1,610.51	
10	6	1,610.51	161.05	1,771.56	
11	7	1,771.56	177.16	1,948.72	
12	8	1,948.72	194.87	2,143.59	
13	9	2,143.59	214.36	2,357.95	
14	10	2,357.95	235.79	2,593.74	
15	11	2,593.74			
16					
17	A simpler way		2,593.74	<-- =B5*(1+B2)^10	

=$B$2*B5

=D5

As cell C17 shows, you don't need all these complicated calculations: The *future value* of 1,000 in 10 years at 10% per year is given by:

$$FV = 1{,}000*(1+10\%)^{10} = 2{,}593.74$$

Now consider the following, slightly more complicated, problem: Again, you intend to open a savings account. Your initial deposit of 1,000 today will be followed by a similar deposit at the beginning of years 1, 2, … , 9. If the account earns 10% per year, how much will you have in the account at the start of year 10?

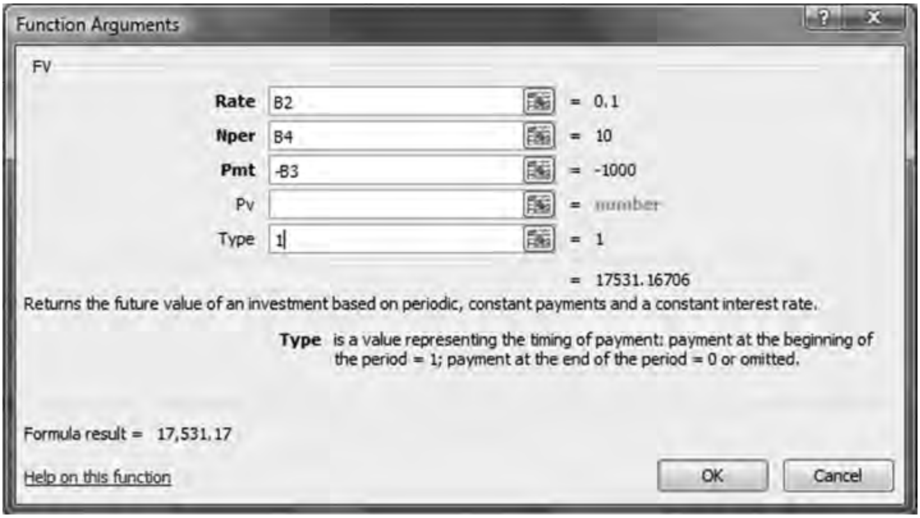
This problem is easily modeled in Excel:

	A	B	C	D	E	F
1	**FUTURE VALUE WITH ANNUAL DEPOSITS**					
2	Interest	10%				
3	Annual deposit	1,000	<-- Made today and at beginning of each of next 9 years			
4	Number of deposits	10				
5						
6	**Year**	**Account balance, beginning of year**	**Deposit at beginning of year**	**Interest earned during year**	**Total in account, end year**	
7	1	0.00	1,000	100.00	1,100.00	<-- =D7+C7+B7
8	2	1,100.00	1,000	210.00	2,310.00	<-- =D8+C8+B8
9	3	2,310.00	1,000	331.00	3,641.00	
10	4	3,641.00	1,000	464.10	5,105.10	=$B$2*(B7+C7)
11	5	5,105.10	1,000	610.51	6,715.61	
12	6	6,715.61	1,000	771.56	8,487.17	
13	7	8,487.17	1,000	948.72	10,435.89	
14	8	10,435.89	1,000	1,143.59	12,579.48	
15	9	12,579.48	1,000	1,357.95	14,937.42	
16	10	14,937.42	1,000	1,593.74	17,531.17	
17						
18	Future value	17,531.17	<-- =FV(B2,B4,-B3,,1)		=E7	

Thus the answer is that we will have 17,531.17 in the account at the end of year 10. This same answer can be represented as a formula that sums the future values of each deposit:

$$\begin{aligned} \textit{Total at beginning of year } 10 &= 1{,}000*(1+10\%)^{10} + 1{,}000*(1+10\%)^{9} \\ &\quad + \ldots + 1{,}000*(1+10\%)^{1} \\ &= \sum_{t=1}^{10} 1{,}000*(1+10\%)^{t} \end{aligned}$$

An Excel function: Note from cell B18 that Excel has a function **FV** which gives this sum. The dialog box brought up by **FV** is the following:



We note three things about this function:

- For positive deposits **FV** returns a negative number. This is an irritating property of this function, which it shares with **PV** and **PMT**. To avoid negative numbers, we have put the **Pmt** in as –1,000.
- The line **Pv** in the dialog box refers to a situation wherein the account has some initial value other than 0 when the series of deposits is made. In the above example, this has been left blank, which indicates that the initial account value is zero.
- As noted in the picture, "Type" (either 1 or 0) refers to whether the deposit is made at the beginning or the end of each period (in our example the former is the case).

1.7 A Pension Problem—Complicating the Future Value Problem

A typical exercise is the following: You are currently 55 years old and intend to retire at age 60. To make your retirement easier, you intend to start a retirement account:

- At the beginning of each of years 1, 2, 3, 4 (that is, starting today and at the beginning of each of the next four years), you intend to make a deposit into the retirement account. You think that the account will earn 8% per year.

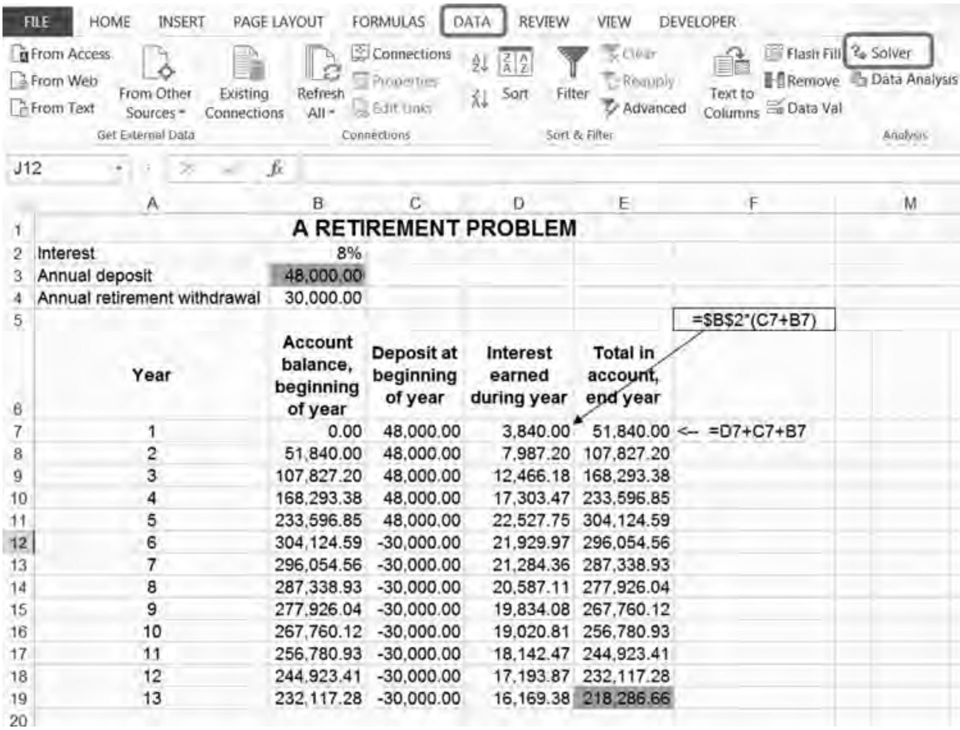
• After retirement at age 60, you anticipate living 8 more years.[5] At the beginning of each of these years you want to withdraw $30,000 from your retirement account. Your account balances will continue to earn 8%.

How much should you deposit annually in the account? The following spreadsheet fragment below shows how easily you can go wrong in this kind of problem—in this case, you've calculated that in order to provide $30,000 per year for 8 years, you need to contribute $240,000/5 = $48,000 in each of the first 5 years. As the spreadsheet shows, you'll end up with a lot of money at the end of 8 years! (The reason—you've ignored the powerful effects of compound interest. If you set the interest rate in the spreadsheet equal to 0%, you'll see that you're right.)

	A	B	C	D	E	F
1	**A RETIREMENT PROBLEM**					
2	Interest	8%				
3	Annual deposit	48,000.00				
4	Annual retirement withdrawal	30,000.00				
5						=$B$2*(C7+B7)
6	**Year**	**Account balance, beginning of year**	**Deposit at beginning of year**	**Interest earned during year**	**Total in account, end year**	
7	1	0.00	48,000.00	3,840.00	51,840.00	<-- =D7+C7+B7
8	2	51,840.00	48,000.00	7,987.20	107,827.20	
9	3	107,827.20	48,000.00	12,466.18	168,293.38	
10	4	168,293.38	48,000.00	17,303.47	233,596.85	
11	5	233,596.85	48,000.00	22,527.75	304,124.59	
12	6	304,124.59	-30,000.00	21,929.97	296,054.56	
13	7	296,054.56	-30,000.00	21,284.36	287,338.93	
14	8	287,338.93	-30,000.00	20,587.11	277,926.04	
15	9	277,926.04	-30,000.00	19,834.08	267,760.12	
16	10	267,760.12	-30,000.00	19,020.81	256,780.93	
17	11	256,780.93	-30,000.00	18,142.47	244,923.41	
18	12	244,923.41	-30,000.00	17,193.87	232,117.28	
19	13	232,117.28	-30,000.00	16,169.38	218,286.66	
20						
21	**Note**: This problem has 5 deposits and 8 annual withdrawals, all made at the beginning of the year. The beginning of year 13 is the last year of the retirement plan; if the annual deposit is correctly computed, the balance at the beginning of year 13 after the withdrawal should be zero.					

5. Of course you're going to live much longer! And I wish you good health! The dimensions of this problem have been chosen to make it fit nicely on a page.

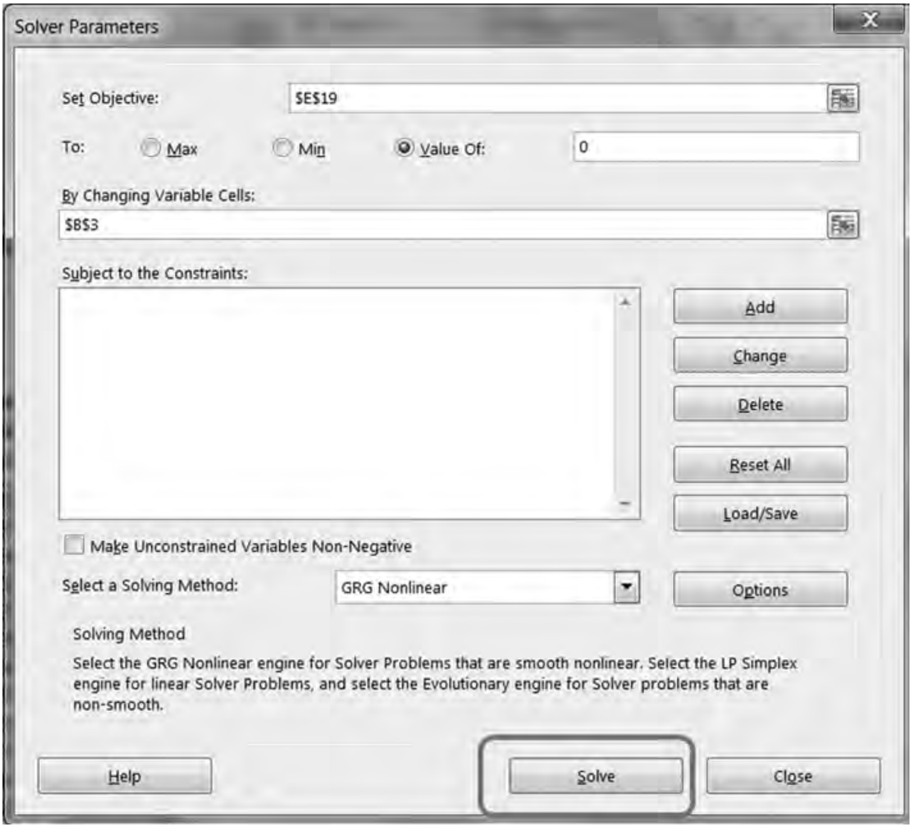
There are several ways to solve this problem. The first involves Excel's **Solver**. This can be found on the **Data** menu.[6]



	A	B	C	D	E
1		A RETIREMENT PROBLEM			
2	Interest	8%			
3	Annual deposit	48,000.00			
4	Annual retirement withdrawal	30,000.00			
5					
6	Year	Account balance, beginning of year	Deposit at beginning of year	Interest earned during year	Total in account, end year
7	1	0.00	48,000.00	3,840.00	51,840.00
8	2	51,840.00	48,000.00	7,987.20	107,827.20
9	3	107,827.20	48,000.00	12,466.18	168,293.38
10	4	168,293.38	48,000.00	17,303.47	233,596.85
11	5	233,596.85	48,000.00	22,527.75	304,124.59
12	6	304,124.59	-30,000.00	21,929.97	296,054.56
13	7	296,054.56	-30,000.00	21,284.36	287,338.93
14	8	287,338.93	-30,000.00	20,587.11	277,926.04
15	9	277,926.04	-30,000.00	19,834.08	267,760.12
16	10	267,760.12	-30,000.00	19,020.81	256,780.93
17	11	256,780.93	-30,000.00	18,142.47	244,923.41
18	12	244,923.41	-30,000.00	17,193.87	232,117.28
19	13	232,117.28	-30,000.00	16,169.38	218,286.66
20					

6. If the Solver does not appear on the Tools menu, then you have to load it. Go to **File|Options|Add-ins** and click **Solver Add-In** on the list of programs. Note that you could also use the **Goal Seek** tool to solve this problem. For simple problems such as this one, there is not much difference between the **Solver** and **Goal Seek**; the one (not inconsiderable) advantage of the **Solver** is that it remembers its previous arguments, so that if you bring it up again on the same spreadsheet, you can see what you did in the previous iteration. In later chapters we will illustrate problems that cannot be solved by **Goal Seek** and in which the use of the **Solver** is a necessity.

Clicking on **Solver** opens a dialog box. Below we've filled it in:



If we now click on the **Solve** box, we get the answer:

	A	B	C	D	E	F
1	A RETIREMENT PROBLEM					
2	Interest	8%				
3	Annual deposit	29,386.55				
4	Annual retirement withdrawal	30,000.00				
5						=$B$2*(C7+B7)
6	Year	Account balance, beginning of year	Deposit at beginning of year	Interest earned during year	Total in account, end year	
7	1	0.00	29,386.55	2,350.92	31,737.48	<-- =D7+C7+B7
8	2	31,737.48	29,386.55	4,889.92	66,013.95	
9	3	66,013.95	29,386.55	7,632.04	103,032.54	
10	4	103,032.54	29,386.55	10,593.53	143,012.62	
11	5	143,012.62	29,386.55	13,791.93	186,191.10	
12	6	186,191.10	-30,000.00	12,495.29	168,686.39	
13	7	168,686.39	-30,000.00	11,094.91	149,781.30	
14	8	149,781.30	-30,000.00	9,582.50	129,363.81	
15	9	129,363.81	-30,000.00	7,949.10	107,312.91	
16	10	107,312.91	-30,000.00	6,185.03	83,497.94	
17	11	83,497.94	-30,000.00	4,279.84	57,777.78	
18	12	57,777.78	-30,000.00	2,222.22	30,000.00	
19	13	30,000.00	-30,000.00	0.00	0.00	

Solving the Retirement Problem Using Financial Formulas

We can solve this problem in a more intelligent fashion if we understand the discounting process. The present value of the whole series of payments, discounted at 8%, must be zero:

$$\sum_{t=0}^{4} \frac{\textit{Initial deposit}}{(1.08)^t} - \sum_{t=5}^{12} \frac{30{,}000}{(1.08)^t} = 0$$

$$\Rightarrow \textit{Initial deposit} = \sum_{t=5}^{12} \frac{30{,}000}{(1.08)^t} \Bigg/ \sum_{t=0}^{4} \frac{1}{(1.08)^t}$$

Rewrite $\sum_{t=5}^{12} \frac{30{,}000}{(1.08)^t} = \frac{1}{(1.08)^4} \sum_{t=1}^{8} \frac{30{,}000}{(1.08)^t}$. We can now use Excel's **PV** and **PMT** functions to solve the problem:

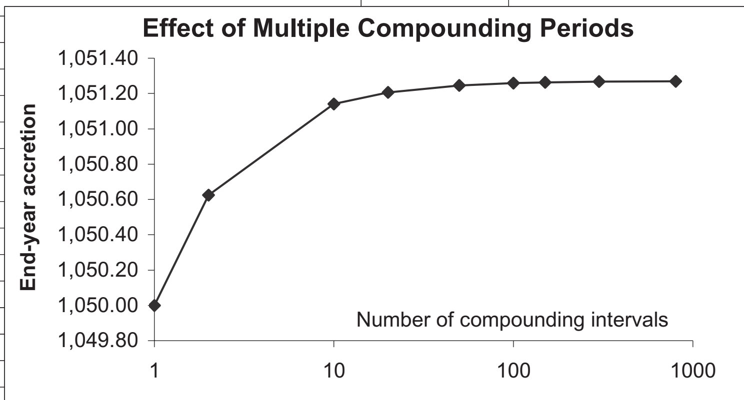
	A	B	C
1	A RETIREMENT PROBLEM		
2	Interest	8%	
3	Annual retirement withdrawal	30,000.00	
4	Years of withdrawal	8	
5	Years of deposit	5	
6	Present value of withdrawals	117,331.98	<-- =-PV(B2,B4,B3)/(1+B2)^B5
7	Annual deposit	29,386.55	<-- =PMT(B2,B5,-B6)

1.8 Continuous Compounding

Suppose you deposit \$1,000 in a bank account which pays 5% per year. This means that at the end of the year you will have \$1,000*(1.05) = \$1,050. Now suppose that the bank interprets "5% per year" to mean that it pays you 2.5% interest twice a year. Thus after 6 months you'll have \$1,025, and after 1 year you will have $\$1{,}000*\left(1+\frac{0.05}{2}\right)^2 = \$1{,}050.625$. By this logic, if you get paid interest n times per year, your accretion at the end of the year will be $\$1{,}000*\left(1+\frac{0.05}{n}\right)^n$. As n increases this amount gets larger, converging (rather quickly, as you will soon see) to $e^{0.05}$, which in Excel is written as the function **Exp**. When n is infinite, we refer to this as *continuous compounding*. (By typing **Exp(1)** in a spreadsheet cell, you can see that e = 2.7182818285. …)

As you can see in the next display, \$1,000 continuously compounded for 1 year at 5% grows to $\$1{,}000*e^{0.05} = \$1{,}051.271$ at the end of the year. Continuously compounded for t years, it will grow to $\$1{,}000*e^{0.05*t}$, where t need not be a whole number (for example, when $t = 4.25$ then the accumulation factor $e^{0.05*4.25}$ measures the growth of the initial investment at 5% annually, continuously compounded for 4 years and 3 months).

	A	B	C
1	**MULTIPLE COMPOUNDING PERIODS**		
2	Initial deposit	1,000	
3	Interest rate	5%	
4	Number of compounding periods per year	2	
5	Interest per compounding period	2.500%	<-- =B3/B4
6	Accretion in one year	1,050.625	<-- =B2*(1+B5)^B4
7	Continuous compounding with **Exp**	1,051.271	<-- =B2*EXP(B3)
8			



	A	B	C
24	**Compounding periods per year**	**End-year accretion**	
25	1	1,050.000	<-- =$B$2*(1+$B$3/A25)^A25
26	2	1,050.625	<-- =$B$2*(1+$B$3/A26)^A26
27	10	1,051.140	
28	20	1,051.206	
29	50	1,051.245	
30	100	1,051.258	
31	150	1,051.262	
32	300	1,051.267	
33	800	1,051.269	

The conclusion: More compounding periods increase the future value, though there is a clear asymptotic value; as we will see below, for accretion over t years, this value is e^{rt}.

Back to Finance—Continuous Discounting

If the accretion factor for continuous compounding at interest r over t years is e^{rt}, then the discount factor for the same period is e^{-rt}. Thus a cash flow C_t occurring in year t and discounted at continuously compounded rate r will be worth $C_t e^{-rt}$ today. This is illustrated below:

	A	B	C	D
1	**CONTINUOUS DISCOUNTING**			
2	Interest	8%		
3				
4	**Year**	**Cash flow**	**Continously discounted PV**	
5	1	100	92.312	<-- =B5*EXP(-$B$2*A5)
6	2	200	170.429	<-- =B6*EXP(-$B$2*A6)
7	3	300	235.988	
8	4	400	290.460	
9	5	500	335.160	
10				
11	Present value		1,124.348	<-- =SUM(C5:C9)

Calculating the Continuously Compounded Return from Price Data

Suppose at time 0 you had $1,000 in the bank and suppose that 1 year later you had $1,200. What was your percentage return? Although the answer may appear obvious, it actually depends on the compounding method. If the bank paid interest only once a year, then the return would be 20%:

$$\frac{1,200}{1,000} - 1 = 20\%$$

However, if the bank paid interest twice a year, you would need to solve the following equation to calculate the return:

$$1,000 * \left(1 + \frac{r}{2}\right)^2 = 1,200 \Rightarrow \frac{r}{2} = \left(\frac{1,200}{1,000}\right)^{1/2} - 1 = 9.5445\%$$

The annual percentage return when interest is paid twice a year is therefore 2*9.5445% = 19.089%.

In general, if there are n compounding periods per year, you have to solve $\frac{r}{n} = \left(\frac{1,200}{1,000}\right)^{1/n} - 1$ and then multiply the result appropriately. If n is very large, this converges to $r = \ln\left(\frac{1,200}{1,000}\right) = 18.2322\%$:

	A	B	C
1	CALCULATING RETURNS FROM PRICES		
2	Initial deposit	1,000	
3	End-of-year value	1,200	
4	Number of compounding periods	2	
5	Implied annual interest rate	19.09%	<-- =((B3/B2)^(1/B4)-1)*B4
6			
7	Continuous return	18.23%	<-- =LN(B3/B2)
8			
9	Implied annual interest rate with n compounding periods		
10	Number of compounding periods	Rate	
11		19.09%	<-- =B5, data table header
12	1	20.00%	
13	2	19.09%	
14	4	18.65%	
15	8	18.44%	
16	20	18.32%	
17	1,000	18.23%	

Why Use Continuous Compounding?

All of this may seem somewhat esoteric. However, continuous compounding/discounting is often used in financial calculations. In this book, it is used to calculate portfolio returns (Chapters 8–13) and in practically all of the options calculations (Chapters 15–19).

There's another reason to use continuous compounding—its ease of calculation. Suppose, for example, that your $1,000 grew to $1,500 in 1 year and 9 months. What's the annualized rate of return? The easiest—and most consistent—way to do this is to calculate the continuously compounded annual return. Since 1 year and 9 months equals 1.75 years, this return is:

$$1{,}000 * \exp[r * 1.75] = 1{,}500 \Rightarrow r = \frac{1}{1.75} \ln\left[\frac{1{,}500}{1{,}000}\right] = 23.1694\%$$

1.9 Discounting Using Dated Cash Flows

Most of the computations in this chapter consider cash flows which occur at fixed periodic intervals. Typically we look at cash flows which occur on dates 0, 1, … , *n*, where the period indicates an annual, semi-annual, or other fixed interval. Two Excel functions, **XIRR** and **XNPV**, allow us to do computations on cash flows which occur on specific dates that need not be at even intervals.[7]

In the following example we compute the IRR of an investment of $1,000 made on 1 January 2014 with payments on specific dates:

	A	B	C
1	**USING XIRR TO COMPUTE THE ANNUALIZED INTERNAL RATE OF RETURN**		
2	**Date**	**Cash flow**	
3	01-Jan-14	-1,000	
4	03-Mar-14	150	
5	04-Jul-14	100	
6	12-Oct-14	50	
7	25-Dec-14	1,000	
8			
9	IRR	37.19%	<-- =XIRR(B3:B7,A3:A7)

The function **XIRR** outputs an annualized return. It works by computing the daily IRR and annualizing it, $XIRR = (1 + daily\ IRR)^{365} - 1$.

XNPV computes the net present value of a series of cash flows occurring on specific dates:

7. If you do not see these functions, add them in by going to **Tools|Add-ins** on the toolbar and checking **Analysis ToolPak**.

	A	B	C
1	**USING XNPV TO COMPUTE THE NET PRESENT VALUE**		
2	Annual discount rate	12%	
3			
4	**Date**	**Cash flow**	
5	01-Jan-14	-1,000	
6	03-Mar-15	100	
7	04-Jul-15	195	
8	12-Oct-16	350	
9	25-Dec-17	800	
10			
11	Net present value	16.80	<-- =XNPV(B2,B5:B9,A5:A9)
12			
13	Note that **XNPV** has a different syntax from **NPV**! **XNPV** requires all the cash flows, including the initial cash flow, whereas **NPV** assumes that the first cash flow occurs one period hence.		

Fixing Bugs in XNPV and XIRR

Both **XNPV** and **XIRR** have bugs, which Microsoft has not fixed in several versions of Excel. The file with this chapter includes functions that fix these bugs, called **NXNPV** and **NXIRR**.[8]

- **XNPV** doesn't work with zero or negative interest rates.
- **XIRR** does not identify multiple internal rates of return.

The **XNPV** relates to the failure of this function to correctly deal with zero or negative discount rates.

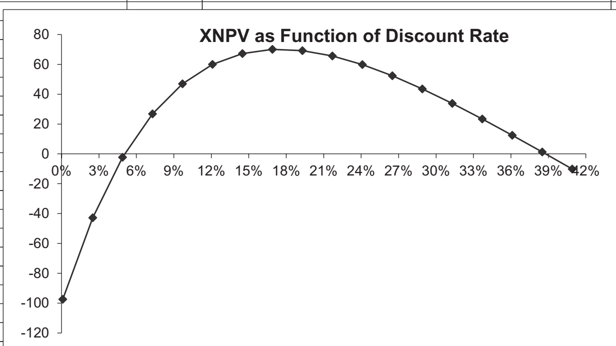
8. These bug fixes were developed by my colleague Benjamin Czaczkes.

	A	B	C
1	**PROBLEM WITH XNPV** XNPV does not work with zero or negative discount rates		
2	Discount rate	-3.00%	
3	Net present value	#NUM!	<-- =XNPV(B2,B7:B13,A7:A13)
4		-194.87	<-- =nXNPV(B2,B7:B13,A7:A13)
5			
6	**Date**	**Cash flow**	
7	30-Jun-14	-500	
8	14-Feb-15	100	
9	14-Feb-16	300	
10	14-Feb-17	400	
11	14-Feb-18	600	
12	14-Feb-19	800	
13	14-Feb-20	-1,800	

The **NXNPV** function fixes this problem.

The bug in **XIRR** is that the **Guess** switch on **XIRR** doesn't work. Consider the following problem:

	A	B	C	D	E	F
1	**PROBLEMS WITH XIRR**					
2	Discount rate	22.00%				
3	Net present value	64.96186	<-- =XNPV(B2,B9:B15,A9:A15)			
4	IRR	#NUM!	<-- =XIRR(B9:B15,A9:A15) , no Guess			
5		#NUM!	<-- =XIRR(B9:B15,A9:A15,35%), Guess = 35%			
6		#NUM!	<-- =XIRR(B9:B15,A9:A15,5%) , Guess = 5%			
7						
8	**Date**	**Cash flow**		**Data table: XNPV as function of discount rate**		
9	30-Jun-14	-500				
10	14-Feb-15	100		Rate	64.962	<-- =B3, data table header
11	14-Feb-16	300		0.1%	-97.366	
12	14-Feb-17	400		2.5%	-42.753	
13	14-Feb-18	600		4.9%	-2.310	
14	14-Feb-19	800		7.3%	26.837	
15	14-Feb-20	-1,800		9.7%	46.983	
16				12.1%	59.961	
17				14.5%	67.240	
18				16.9%	70.000	
19				19.3%	69.191	
20				21.7%	65.578	
21				24.1%	59.780	
22				26.5%	52.296	
23				28.9%	43.528	
24				31.3%	33.803	
25				33.7%	23.384	
26				36.1%	12.484	
27				38.5%	1.272	
28				40.9%	-10.113	
29						
30						
31						
32						
33						
34						



From the data table, it is evident that there are two internal rates of return (around 5% and around 39%). But the **XIRR** function does not identify either (see cells B4:B6).

The function **NXIRR** fixes this bug:

	A	B	C
1	**NXIRR FIXES THE XIRR BUG**		
2	Discount rate	-3.00%	
3	IRR	5.06%	<-- =nXIRR(B8:B14,A8:A14) , no Guess
4		38.77%	<-- =nXIRR(B8:B14,A8:A14,35%), Guess = 35%
5		5.06%	<-- =nXIRR(B8:B14,A8:A14,5%) , Guess = 5%
6			
7	**Date**	**Cash flow**	
8	30-Jun-14	-500	
9	14-Feb-15	100	
10	14-Feb-16	300	
11	14-Feb-17	400	
12	14-Feb-18	600	
13	14-Feb-19	800	
14	14-Feb-20	-1,800	

Exercises

1. You are offered an asset costing $600 that has cash flows of $100 at the end of each of the next 10 years.

   a. If the appropriate discount rate for the asset is 8%, should you purchase it?

   b. What is the IRR of the asset?

2. You just took a $10,000, 5-year loan. Payments at the end of each year are flat (equal in every year) at an interest rate of 15%. Calculate the appropriate loan table, showing the breakdown in each year between principal and interest.

3. You are offered an investment with the following conditions:

   - The cost of the investment is 1,000.
   - The investment pays out a sum X at the end of the first year; this payout grows at the rate of 10% per year for 11 years.

   If your discount rate is 15%, calculate the smallest X which would entice you to purchase the asset. For example, as you can see in the following display, $X = \$100$ is too small—the NPV is negative:

	A	B	C
1	Discount rate	15%	
2	Initial payment	129.2852	
3	NPV	-226.52	<-- =B6+NPV(B1,B7:B17)
4			
5	**Year**	**Cash flow**	
6	0	-1,000.00	
7	1	100.00	<-- 100
8	2	110.00	<-- =B7*1.1
9	3	121.00	<-- =B8*1.1
10	4	133.10	
11	5	146.41	
12	6	161.05	
13	7	177.16	
14	8	194.87	
15	9	214.36	
16	10	235.79	
17	11	259.37	

4. The following cash-flow pattern has two IRRs. Use Excel to draw a graph of the **NPV** of these cash flows as a function of the discount rate. Then use the **IRR** function to identify the two IRRs. Would you invest in this project if the opportunity cost were 20%?

	A	B
4	**Year**	**Cash flow**
5	0	-500
6	1	600
7	2	300
8	3	300
9	4	200
10	5	-1,000

5. In this exercise we solve iteratively for the internal rate of return. Consider an investment which costs 800 and has cash flows of 300, 200, 150, 122, 133 in years 1–5. Setting up the loan table below shows that 10% is greater than the IRR (since the return of principal at the end of year 5 is less than the principal at the beginning of the year):

	A	B	C	D	E	F	G	H
1	**IRR?**	10.00%						
2				**LOAN TABLE**			Division of payment between:	
3	**Year**	**Cash flow**		**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest**	**Principal**
4	0	-800		1	800.00	300.00	80.00	220.00
5	1	300		2	580.00	200.00	58.00	142.00
6	2	200		3	438.00	150.00	43.80	106.20
7	3	150		4	331.80	122.00	33.18	88.82
8	4	122		5	242.98	133.00	24.30	108.70
9	5	133		6	134.28	<-- Should be zero for IRR		

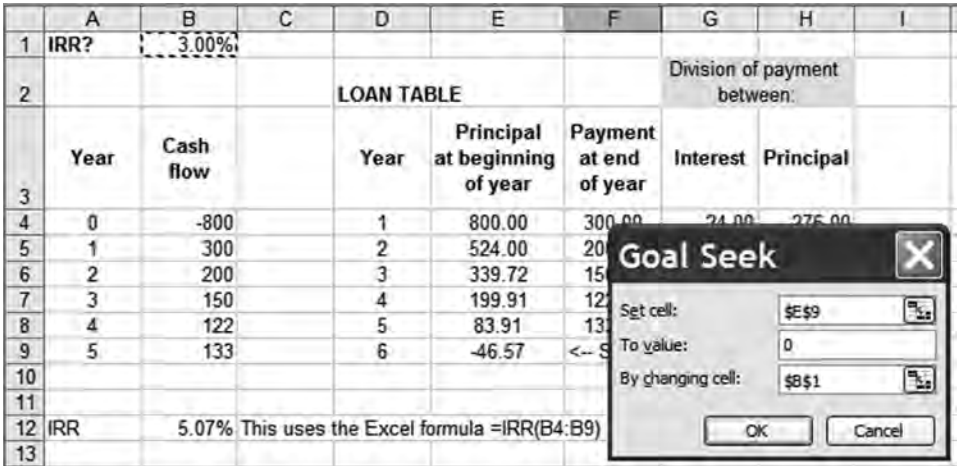
Setting the IRR? cell equal to 3% shows that 3% is less than the IRR, since the return of principal at the end of year 5 is greater than the principal at the beginning of year 5.

By changing the IRR? cell, find the internal rate of return of the investment.

	A	B	C	D	E	F	G	H
1	**IRR?**	3.00%						
2				**LOAN TABLE**			Division of payment between:	
3	**Year**	**Cash flow**		**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest**	**Principal**
4	0	-800		1	800.00	300.00	24.00	276.00
5	1	300		2	524.00	200.00	15.72	184.28
6	2	200		3	339.72	150.00	10.19	139.81
7	3	150		4	199.91	122.00	6.00	116.00
8	4	122		5	83.91	133.00	2.52	130.48
9	5	133		6	-46.57	<-- Should be zero for IRR		

6. An alternative definition of the IRR is the rate which makes the principal at the beginning of year 6 equal to zero.[9] This is shown in the printout above, in which cell E9 gives the principal at the beginning of year 6. Using the **Goal Seek** function of Excel, find this rate (below we illustrate how the screen should look).

9. In general, of course, the IRR is the rate of return that makes the principal in the year *following* the last payment equal to zero.



	A	B	C	D	E	F	G	H	I
1	IRR?	3.00%							
2				LOAN TABLE			Division of payment between:		
3	Year	Cash flow		Year	Principal at beginning of year	Payment at end of year	Interest	Principal	
4	0	-800		1	800.00	300.00	24.00	276.00	
5	1	300		2	524.00	20			
6	2	200		3	339.72	15			
7	3	150		4	199.91	12			
8	4	122		5	83.91	13			
9	5	133		6	-46.57	<-- S			
10									
11									
12	IRR	5.07%	This uses the Excel formula =IRR(B4:B9)						
13									

(Of course you should check your calculations by using the Excel **IRR** function.)

7. Calculate the flat annual payment required to pay off a 13%, 5-year loan of $100,000.

8. You have just taken a car loan of $15,000. The loan is for 48 months at an annual interest rate of 15% (which the bank translates to a monthly rate of 15%/12 = 1.25%). The 48 payments (to be made at the end of each of the next 48 months) are all equal.

   a. Calculate the monthly payment on the loan.
   b. In a loan table calculate, for each month: the principal remaining on the loan at the beginning of the month and the split of that month's payment between interest and repayment of principal.
   c. Show that the principal at the beginning of each month is the present value of the remaining loan payments at the loan interest rate (use either **NPV** or the **PV** functions).

9. You are considering buying a car from a local auto dealer. The dealer offers you one of two payment options:

   - You can pay $30,000 cash.
   - The "deferred payment plan": You can pay the dealer $5,000 cash today and a payment of $1,050 at the end of each of the next 30 months.

   As an alternative to the dealer financing, you have approached a local bank, which is willing to give you a car loan of $25,000 at the rate of 1.25% per month.

a. Assuming that 1.25% is the opportunity cost, calculate the present value of all the payments on the dealer's deferred payment plan.

b. What is the effective interest rate being charged by the dealer? Do this calculation by preparing a spreadsheet like this (only part of the spreadsheet is shown—you have to do this calculation for all 30 months):

	D	E	F	G	H
2	**Month**	**Cash payment**	**Payment under deferred payment plan**	**Difference**	
3	0	30,000	5,000	25,000	<-- =E3-F3
4	1	0	1,050	-1,050	<-- =E4-F4
5	2	0	1,050	-1,050	
6	3	0	1,050	-1,050	
7	4	0	1,050	-1,050	
8	5	0	1,050	-1,050	
9	6	0	1,050	-1,050	
10	7	0	1,050	-1,050	
11	8	0	1,050	-1,050	

Now calculate the IRR of the difference column; this is the monthly *effective interest rate* on the deferred payment plan.

10. You are considering a savings plan which calls for a deposit of $15,000 at the end of each of the next 5 years. If the plan offers an interest rate of 10%, how much will you accumulate at the end of year 5? Do this calculation by completing the following spreadsheet. This spreadsheet does the calculation twice—once using the **FV** function and once using a simple table which shows the accumulation at the beginning of each year.

	A	B	C	D
1	Annual payment	15,000		
2	Interest rate	10%		
3	Number of years	5		
4	Total value	$91,576.50	<-- =FV(B2,B3,-B1,,0)	
5				
6	**Year**	**Accumulation at beginning of year**	**Payment at end of year**	**Annual interest**
7	1	0	15,000	0.00
8	2	15,000	15,000	1,500.00
9	3	31,500		
10	4			
11	5			
12	6			

11. Redo the previous calculation, this time assuming that you make 5 deposits at the *beginning* of this year and the following 4 years. How much will you accumulate by the end of year 5?

12. A mutual fund has been advertising that, had you deposited $250 per month in the fund for the last 10 years, you would now have accumulated $85,000. Assuming that these deposits were made at the beginning of each month for a period of 120 months, calculate the effective annual return fund investors got.

    Hint: Set up the following spreadsheet and then use **Goal Seek**.

	A	B	C
1	Monthly payment	250	
2	Number of months	120	
3			
4	Effective monthly return?		
5	Accumulation		<-- =FV(B4,B2,-B1,,1)

The effective annual return can then be calculated in one of two ways:

- $(1 + \textit{monthly return})^{12} - 1$: This is the compound annual return, which is preferable, since it makes allowance for the reinvestment of each month's earnings.
- 12**monthly return*: This method is often used by banks.

13. You have just turned 35, and you intend to start saving for your retirement. Once you retire in 30 years (when you turn 65), you would like to have an income of $100,000 per year for the next 20 years. Calculate how much you would have to save between now and age 65 in order to finance your retirement income. Make the following assumptions:

    - All savings draw compound interest of 10% per year.
    - You make the first payment today and the last payment on the day you turn 64 (30 payments).
    - You make the first withdrawal when you turn 65 and the last withdrawal when you turn 84 (20 payments).

14. You currently have $25,000 in the bank, in a savings account that draws 5% interest. Your business needs $25,000, and you are considering two options: (a) Use the money in your savings account. (b) Borrow the money from the bank at 6%, leaving the money in the savings account.

Your financial analyst suggests that solution (b) above is better. His logic: The sum of the interest paid on the 6% loan is lower than the interest earned at the same time on the $25,000 deposit. His calculations are illustrated below. Show that this logic is wrong. (If you think about it, it couldn't be preferable to take a 6% loan when you are getting 5% interest from the bank. However, the explanation for this may not be trivial.)

	A	B	C	D	E	F
1	**EXERCISE 14, financial analyst's calculations**					
2	Interest earned	5%				
3	Interest paid	6%				
4	Initial deposit	25,000				
5					=PMT($B$3,2,-$B$4)	
6	**THE 6% LOAN**					
7	**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest paid**	**Repayment of principal**	
8	1	25,000.00	13,635.92	1,500.00	12,135.92	<-- =C8-D8
9	2	12,864.08	13,635.92	771.84	12,864.08	
10		**Total interest paid**		2,271.84		
11						
12	**Savings Account**					
13	**Year**	**In savings account at beginning of year**	**End-year interest earned**	**In account at end of year**		
14	1	25,000.00	1,250.00	26,250.00		
15	2	26,250.00	1,312.50	27,562.50		
16		**Interest earned**	2,562.50			

15. Use **XIRR** to compute the internal rate of return of the following investment:

	A	B
1	**Date**	**Cash flow**
2	30-Jun-07	-899
3	14-Feb-08	70
4	14-Feb-09	70
5	14-Feb-10	70
6	14-Feb-11	70
7	14-Feb-12	70
8	14-Feb-13	1,070

16. Use **XNPV** to value the following investment. Assume that the annual discount rate is 15%.

	A	B
4	**Date**	**Cash flow**
5	30-Jun-07	-500
6	14-Feb-08	100
7	14-Feb-09	300
8	14-Feb-10	400
9	14-Feb-11	600
10	14-Feb-12	800
11	14-Feb-13	-1,800

17. Identify the two internal rates of return of the investment in exercise 16.

2 Corporate Valuation Overview

2.1 Overview

Chapters 3, 4, 5, and 6 discuss various aspects of corporate valuation, one of the trickiest topics in finance. This chapter aims to provide a short overview of this complicated topic. In succeeding chapters we turn to the components of valuation: The computation of the cost of capital (Chapter 3), the direct valuation of corporate free cash flows from the firm's consolidated statement of cash flows (Chapter 4), and the projection of firm pro forma financial statements (Chapters 5 and 6).

What Is Corporate Valuation About?

When we discuss the valuation of a company, we may be referring to any of the following:

• Enterprise value: Valuing the company's productive activities.

• Equity: Valuing the shares of a company, whether for the purpose of buying or selling a single share or valuing all of the equity for purposes of a corporate acquisition.

• Debt: Valuing the company's debt. When debt is risky, its value depends on the value of the company that has issued the debt.

• Other: We may want to value other securities related to the company—for example, the firm's warrants or options, employee stock options, etc.

In Chapters 2–6 of *Financial Modeling*, we discuss the first two of these topics, leaving the valuation of debt and other securities for later chapters.[1]

2.2 Four Methods to Compute Enterprise Value (EV)

The key concept in corporate valuation is *enterprise value*. The enterprise value (EV) of the firm is the value of the firm's core business activities and forms the basis of most corporate valuation models. We distinguish between four approaches to computing the enterprise value:

1. Valuation of risky bonds is discussed in Chapter 23. The valuation of derivative securities is discussed in Chapters 15–19, 29, and 30.

- The accounting approach to EV moves items on the balance sheet so that all operating items are on the left-hand side of the balance sheet and all financial items are on the right-hand side. Although most academics sneer at this approach, it is often a useful starting point for thinking about the enterprise value.
- The efficient markets approach to EV revalues—to the extent possible—items on the accounting EV balance sheet at market values. An obvious revaluation is to replace the firm's book value of equity with the market value of the equity. To the extent that we know the market value of other firm liabilities—debt, pension obligations, etc.—this market value will also replace the book values.
- The discounted cash flow (DCF) approach values the EV as the present value of the firm's future anticipated free cash flows (FCFs) discounted at the weighted average cost of capital (WACC). The FCFs can best be thought of as the cash flows produced by the firm's productive assets—its working capital, fixed assets, goodwill, etc.
- In this book we use two implementations of the DCF approach. These approaches differ in their derivation of the firm's free cash flows.
  - In Chapter 4 we base our projections of future anticipated FCFs on an analysis of the firm's consolidated statement of cash flows.
  - In Chapters 5 and 6 we base our projections of future anticipated FCFs on a pro forma model for the firm's financial statements.

2.3 Using Accounting Book Values to Value a Company: The Firm's Accounting Enterprise Value

While we would rarely use accounting numbers to value a company, the balance sheet of a company is a useful starting framework for the valuation process. In this section we show how accounting statements can help us define the concept of *enterprise value* (EV). As a starting point, consider the balance sheet for XYZ Corp.:

	A	B	C	D	E
1	**XYZ CORP BALANCE SHEET**				
2	**Assets**			**Liabilities and equity**	
3	Short-term assets			Short-term liabilities	
4	Cash	1,000		Accounts payable	1,500
5	Marketable securities	1,500		Taxes payable	200
6	Inventories	1,500		Current portion of long-term debt	1,000
7	Accounts receivable	3,000		Short-term debt	500
8					
9	Fixed assets			Long-term debt	1,500
10	Land	150		Pension liabilities	800
11	Plant, property and equipment at cost	2,500			
12	Minus accumulated depreciation	-700		Preferred stock	200
13	Net fixed assets			Minority interest	100
14					
15				Equity	
16	Goodwill	1,000		Stock at par	1,000
17				Accumulated retained earnings	3,500
18				Stock repurchases	-350
19	**Total assets**	**9,950**		**Total liabilities and equity**	**9,950**

We rewrite this balance sheet:

- We separate the operational versus financial items in short-term assets and short-term liabilities.
- We move the operational current assets to the left side of the balance sheet.
- We move all the debt (short-term debt, current portion of long-term debt, and long-term) into one debt item.

	A	B	C	D	E	F
1	**XYZ BALANCE SHEET** **Operational current liabilities moved to left side** **All financial liabilities in one account on right side**					
2	**Assets**			**Liabilities and equity**		
3	Liquid assets (cash + marketable securities)	2,500		Financial debt		
4				Current portion of long-term debt	1,000	
5	Current assets, operational			Short-term debt	500	
6	Inventories	1,500		Long-term debt	1,500	
7	Accounts receivable	3,000		Total financial debt	3,000	
8	Minus, current liabilities, operational					
9	Accounts payable	-1,500				
10	Taxes payable	-200		Pension liabilities	800	
11	Net working capital	2,800	<-- =SUM(B6:B10)			
12				Preferred stock	200	
13	Fixed assets	1,950		Minority interest	100	
14						
15	Goodwill	1,000		Equity	4,150	
16						
17	**Left-hand side of rewritten balance sheet**	**8,250**	<-- =B11+B13+B15+SUM(B3:B4)	**Right-hand side of rewritten balance sheet**	**8,250**	<-- =E7+E10+SUM(E12:E15)

In the next step we subtract liquid assets (cash and marketable securities) from financial debts, to get the firm's net financial debt. When we finish this step, we have all of the firm's productive assets on the left side of the balance sheet and all of its financing on the right side. The left-hand side of the resulting balance sheet is the firm's *enterprise value*, defined as the value of the firm's operational assets: These are the assets that provide the cash flows for the firm's actual business activities:

	A	B	C	D	E	F
1	**XYZ ENTERPRISE VALUE BALANCE SHEET**					
2	**Assets**			**Liabilities and equity**		
3	Net working capital	2,800		Total financial debt	3,000	
4				Minus liquid assets	-2,500	
5	Fixed assets	1,950		Net debt	500	
6						
7	Goodwill	1,000		Pension liabilities	800	
8						
9				Preferred stock	200	
10				Minority interest	100	
11						
12				Equity	4,150	
13						
14	**Enterprise value**	**5,750**	<-- =B3+B5+B7	**Enterprise value**	**5,750**	<-- =E5+E7+SUM(E9:E12)

Caterpillar Corporation[2]

To show the rewriting of the balance sheet in practice, here is the 31 December 2011 balance sheet for Caterpillar Corp. (CAT):

CATERPILLAR CORP., BALANCE SHEET 31 December 2011				
Current assets			Current liabilities	
Cash and cash equivalents	3,057,000		Accounts payable	16,946,000
Short-term investments			Short-term debt	9,648,000
Net receivables	19,533,000		Other current liabilities	1,967,000
Inventory	14,544,000		Total current liabilities	28,561,000
Other current assets	994,000			
Total current assets	38,128,000		Long-term debt	24,944,000
			Other liabilities	14,539,000
Long-term investments	13,211,000			
Property, plant, and equipment (net)	14,395,000		Minority interest	46,000
Goodwill	7,080,000		Total liabilities	68,090,000
Intangible assets	4,368,000			
Other assets	2,107,000		Stocks, options, warrants	473,000
Deferred long-term asset charges	2,157,000		Common stock	4,273,000
			Retained earnings	25,219,000
			Treasury stock	-10,281,000
			Other stockholder equity	-6,328,000
			Total equity	13,356,000
Total assets	**81,446,000**		**Total equity and liabilities**	**81,446,000**

To get to the enterprise value balance sheet for Caterpillar, we move financial items from the left side of the balance sheet to the right, and we move operating current liabilities from the right side of the balance sheet to the left. Notice that we netted out liquid assets (cash and marketable securities) from the financial debts of the company. The assumption is that these assets are not needed for the core business activities of Caterpillar.

The book value of Caterpillar's enterprise value is $59,476,000:

2. We revisit CAT in Chapter 3, where we compute its weighted average cost of capital, and in Chapter 4, where we build a pro forma for the company.

	A	B	C	D	E	F
1	**CATERPILLAR CORP., 2011 ENTERPRISE VALUE BALANCE SHEET** **Book values**					
2	Net working capital	16,158,000	<-- =19533000+14544000+994000-16946000-1967000	Net financial debt	31,535,000	<-- =9648000+24944000-3057000
3	Long-term investments	13,211,000		Other liabilities	14,539,000	
4	Property, plant, and equipment	14,395,000				
5	Goodwill	7,080,000		Minority interest	46,000	
6	Intangible assets	4,368,000				
7	Other assets	2,107,000		Equity	13,356,000	
8	Deferred long-term asset charges	2,157,000				
9	**Enterprise value**	**59,476,000**	<-- =SUM(B2:B8)	**Enterprise value**	**59,476,000**	<-- =SUM(E2:E7)

2.4 The Efficient Markets Approach to Corporate Valuation

The Caterpillar example above assumes that the book value is a correct valuation of the company. But a simple calculation shows how problematic this is: At the end of 2011 Caterpillar had 624.72 million shares outstanding, and the market price per share was $90.60. This suggests that the Caterpillar's enterprise value is $102.720 billion—a far cry from the book value of the enterprise value of $59.476 billion.

	A	B	C
1	**CATERPILLAR VALUATION OF EQUITY AND FINANCIAL LIABILITIES: EFFICIENT MARKETS APPROACH** Most figures in thousand $		
2	Number of shares outstanding	624,722.72	<-- thousand shares
3	Price per share	90.60	<-- 30dec2011
4	Equity value ("Market Cap")	56,599,878	<-- =B2*B3, thousand $
5			
6	Cash and cash equivalents	3,057,000	
7	Short-term debt and current portion of long-term debt	9,648,000	
8	Long-term debt	24,944,000	
9	Net debt	31,535,000	<-- =SUM(B7:B8)-B6
10			
11	Other liabilities	14,539,000	
12			
13	Minority interest	46,000	
14	Preferred stock	0	
15			
16	**Enterprise value: Equity + Net debt + Minority Interest + Preferred**	102,719,878	<-- =SUM(B4,B9,B11,B13)

The efficient markets approach to the valuation of Caterpillar's equity and financial liabilities assumes that the market value of a company's shares or debt is simply the market value at the time of valuation. This approach is better than the accounting approach of the previous section and much simpler than the DCF valuations illustrated in succeeding sections and in Chapters 5 and 6. Moreover, it has the power of logic and much academic research behind it. If markets work—in the sense that there are many participants trading the corporate securities, that there is a lot of information about the company in question, and that the valuator has no special information—why not accept the market price as the true value of the company?[3]

Applying the efficient markets approach to the Caterpillar Enterprise Value Balance sheet gives 102,719,878 for the right-hand side of the enterprise value balance sheet. This means, of course, that we have to revalue the left-hand side of the balance sheet. One approach to bringing this enterprise value balance sheet into balance is to assume that the net working capital's book value is a reasonable approximation to its market value. We can then recompute the market value of the firm's long-term assets to bring the balance sheet into balance.

	A	B	C	D	E	F
1	**CATERPILLAR CORP., 2011 ENTERPRISE VALUE BALANCE SHEET** **Right-hand side revalued at market values** **Left-hand side brought into balance with right-hand side by adjusting long-term assets**					
2	Net working capital	16,158,000	<-- =19533000+14544000+994000-16946000-1967000	Net financial debt	31,535,000	<-- =9648000+24944000-3057000
3	Long-term investments	86,561,878	<-- =E10-B2	Other liabilities	14,539,000	
4	Property, plant, and equipment					
5	Goodwill			Minority interest	46,000	
6	Intangible assets					
7	Other assets			Equity	56,599,878	<-- Market cap
8	Deferred long-term asset charges					
9						
10	**Enterprise value**	**102,719,878**	<-- =SUM(B2:B8)	**Enterprise value**	**102,719,878**	<-- =SUM(E2:E7)
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

3. We have to be careful: Market prices change over time, often precipitously. The efficient markets hypothesis only says that it is impossible to predict market prices beyond the information compounded into current market information.

Note that we could also apply the market valuation to other components of the right-hand side of the balance sheet—we could try to revalue the firm's financial obligations, its other liabilities, and minority interest. This is usually not done, unless there is a convincing case that the book values for these liabilities differ materially from their market value.

2.5 Enterprise Value (EV) as the Present Value of the Free Cash Flows: DCF "Top Down" Valuation

In the previous section we valued the EV by using the market value of the right-hand side of the firm's enterprise value balance sheet—using the market value of its equity and perhaps the market valuation of other elements of the firm's financing. In this section we concentrate on the left-hand side of the enterprise value balance sheet.

The discounted cash flow (DCF) method focuses on two central concepts:

- The firm's free cash flows (FCFs) are defined as the cash created by the firm's operating activities.
- The firm's *weighted average cost of capital* (WACC) is the risk-adjusted discount rate appropriate to the risk of the FCFs (Chapter 3).
- The firm's *enterprise value* (EV) is the present value of the future FCFs discounted at the WACC: $EV = \sum_{t=1}^{\infty} \frac{FCF_t}{(1+WACC)^t}$. The idea is to value a company by considering the present value of the FCFs, where FCF is defined as the cash flow to the firm from its assets (the word *assets* is used broadly, and can be fixed assets, intellectual and trademark assets, and net working capital). The next subsection discusses this concept in more detail.
- In these chapters we often make two additional assumptions: We assume a limited number of predictive periods for the FCFs and we assume that cash flows occur throughout the year. This leads to

$$EV = \sum_{t=1}^{N} \frac{FCF_t}{(1+WACC)^{t-0.5}} + \frac{Terminal\ value}{(1+WACC)^{N-0.5}}$$

$$= \underbrace{\left[\sum_{t=1}^{N} \frac{FCF_t}{(1+WACC)^{t}} + \frac{Terminal\ value}{(1+WACC)^{N}}\right]}_{\substack{\uparrow \\ \text{Computed with Excel's NPV function}}}(1+WACC)^{0.5}$$

The assumption that, for valuation purposes, cash flows occur approximately in mid-year is meant to capture the fact that most corporate cash flows occur throughout the year and that it is therefore a mistake to value them as if they occur at year-end. As can be seen from the above formula, the mid-year assumption is easy to compute in Excel: We simply take the Excel **NPV** and multiply it by $(1 + \text{WACC})^{0.5}$.

Defining the Free Cash Flow (FCF)

The free cash flow (FCF) is a measure of how much cash is produced by the firm's operations. There are two accepted definitions of the FCF (both of which, of course, ultimately boil down to the same thing).

Free Cash Flow Based on the Income Statement	
Profit after taxes	This is the basic measure of the profitability of the business, but it is an accounting measure that includes financing flows (such as interest), as well as non-cash expenses such as depreciation. Profit after taxes does not account for either changes in the firm's working capital or purchases of new fixed assets, both of which can be important cash drains on the firm.
+ Depreciation and other non-cash expenses	Depreciation is a non-cash expense that in the FCF calculation is added back to the profit after tax. To the extent that there are other non-cash expenses in the firm's income statement, these are also added back.
– Increase in operating current assets	When the firm's sales increase, more investment is needed in inventories, accounts receivable, etc. This increase in current assets is not an expense for tax purposes (and is therefore ignored in the profit after taxes), but it is a cash drain on the company. When adjusting the FCF for current assets, we take care to not include financial current assets that are not directly related to sales such as cash and marketable securities.
+ Increase in operating current liabilities	An increase in the sales often causes an increase in financing related to sales (such as accounts payable or taxes payable). This increase in current liabilities—when related to sales—provides cash to the firm. Since it is directly related to sales, we include this cash in the free cash flow calculations. When adjusting the FCF for current liabilities, we take care to not include financial current liabilities that are not directly related to sales: changes in short-term debt and the current portion of long-term debt being the most prominent examples.

– Increase in fixed assets at cost (CAPEX)	An increase in fixed assets (the long-term productive assets of the company) is a use of cash, which reduces the firm's free cash flow.
+ After-tax interest payments (net)	FCF is an attempt to measure the cash produced by the business activity of the firm. To neutralize the effect of interest payments on the firm's profits, we: • Add back the after-tax cost of interest on debt (*after-tax* since interest payments are tax-deductible), • Subtract out the after-tax interest payments on cash and marketable securities.

An alternative, equivalent, definition is based on the firm's earnings before income and taxes:

Free Cash Flow Based on the EBIT (earnings before interest and taxes)

EBIT	
+ Depreciation and other non-cash expenses	
– Increase in Operating Current Assets	The sum $-\Delta CA + \Delta CL$ is the change in the firm's net working capital ΔNWC
+ Increase in Operating Current Liabilities	
– Increase in fixed assets at cost (CAPEX)	

How Can We Predict Future FCFs?

The most critical aspect in valuing a company is to project its anticipated future free cash flows. In this book we explore two ways of deriving these cash flows. Both methods are primarily based on accounting data. Since accounting data is historical data, we need to add some judgment about how this data will develop in the future.

- One method is to base our estimates of future free cash flows on the consolidated statement of cash flows (CSCF) of the firm (Chapter 4).
- The second method is to estimate a set of pro forma financial statements for the firm and to derive the free cash flows from these statements (Chapter 5).

Both methods are briefly reviewed in succeeding sections.

2.6 Free Cash Flows Based on Consolidated Statement of Cash Flows (CSCF)

The consolidated statement of cash flows is part of every financial statement. It is the accountant's explanation of how much cash was generated by the business, and how this cash was generated. The consolidated statement of cash flows (CSCF) is composed of three sections: Operating cash flows, investment cash flows, and financing cash flows. When we use the CSCF to determine FCFs, we use the following general procedure (explanations are given in subsequent subsections and again in Chapter 4):

- We accept the operating cash flows as reported by the firm.
- We carefully examine the investment cash flows, leaving for the FCF the investment cash flows related to productive activities and eliminating those investment cash flows relating to investment by the firm in financial assets.
- We do not count toward the FCF any of the financial cash flows.
- In all cases we are careful about whether specific items are one time or recurring, eliminating from consideration the one-time items.
- We adjust the totals of the adjusted CSCF numbers by adding back net interest paid.

The subsections below contain further explanations. A worked-out example is given in the next section.

CSCF, Section 1: Operating Cash Flows

The operating cash flows adjust the firm's net income for non-cash deductions to the income and for changes in the firm's operating net working capital. Because modern accounting statements include many non-cash items, translating the firm's accounts to a cash basis necessitates many adjustments. A classic adjustment is to add back depreciation to the firm's income: Since depreciation is a non-cash charge on the firm's income, it must be added back when making the adjustment for cash. But depreciation is just the tip of the non-cash iceberg:

- When firms issue stock options to employees, the value of these options is deducted from the firm's income. The logic behind this—that in giving its employees options, the firm has given them something of value which must be accounted for in its income statement—is impeccable. But the actual charge for options is a non-cash deduction, and in the consolidated statement of cash flows, it is added back.

- A firm's income statements must reflect the decrease in goodwill (so-called "impairment"). This impairment—the loss in economic value of an intangible asset that was purchased by the firm—is an economic loss to the firm's shareholders. But it is not a cash flow loss, and in the consolidated statement of cash flows it is added back.
- The list can include a large number of additional items, as we shall see in Chapter 4.

For purposes of computing the firm's free cash flows, we can usually leave all the items in the operating cash flow section of the CSCF.

SCF, Section 2: Investment Cash Flows

The second section of the CSCF includes all investments made by the firm. These investments include both investments in securities and investments in operating assets of the firm.

- Investment in securities can refer to the sale or the purchase of securities held by the firm. Investment in securities is not part of the firm's free cash flows, which are intended to measure solely cash flows related to the firm's core business activities.
- Investments in fixed assets are usually related to the firm's FCFs.

For purposes of computing the firm's free cash flows, we need to distinguish between financial investment cash flows (not part of the FCF) and investment in assets used to produce the firm's business income (part of the FCF).

CSCF, Section 3: Financing Cash Flows

The last section of the CSCF deals with changes in the firm's financing. For FCF purposes, we can ignore this section.

2.7 ABC Corp., Consolidated Statement of Cash Flows (CSCF)

Five years of the consolidated statement of cash flows of ABC Corp. are given below:

	A	B	C	D	E	F	G
1	**ABC CORPORATION** **Consolidated Statement of Cash Flows, 2008-2012**						
2		**2008**	**2009**	**2010**	**2011**	**2012**	
3	**Operating Activities:**						
4	Net earnings	479,355	495,597	534,268	505,856	520,273	
5	Adjustments to reconcile net earnings to net cash provided by operating activities						
6	Add back depreciation and amortization	41,583	47,647	46,438	45,839	46,622	
7	Changes in operating assets and liabilities:						
8	Subtract increase in accounts receivable	9,387	25,951	-12,724	1,685	-2,153	
9	Subtract increase in inventories	-37,630	-22,780	-16,247	-15,780	-5,517	
10	Subtract increase in prepaid expenses and other assets	-52,191	13,573	16,255	14,703	-2,975	
11	Add increase in accounts payable, accrued expenses, pensions, and other liabilities	29,612	51,172	6,757	40,541	60,255	
12	**Net cash provided by operating activities**	**470,116**	**611,160**	**574,747**	**592,844**	**616,505**	<-- =SUM(F4:F11)
13							
14	**Investing Activities:**						
15	Short-term investments, net	-5,000	-55,000	50,000	-10,000	20,000	
16	Purchases of property, plant, and equipment	-48,944	-70,326	-89,947	-37,044	-88,426	
17	Proceeds from dispositions of property, plant and equipment	197	6,956	22,942	6,179	28,693	
18	**Net cash used in investing activities**	**-53,747**	**-118,370**	**-17,005**	**-40,865**	**-39,733**	<-- =SUM(F15:F17)
19							
20	**Financing Activities:**						
21	Repayment of debt	0	0	-300,000	0	-7,095	
22	Proceeds from revolving credit facility borrowings	1,242,431	0	0	0	250,000	
23	Proceeds from the issuance of stock	48,286	114,276	69,375	68,214	37,855	
24	Dividends paid	-332,986	-344,128	-361,208	-367,499	-378,325	
25	Stock repurchased	-150,095	-200,031	-200,038	-200,003	-597,738	
26	**Net cash used in financing activities**	**807,636**	**-429,883**	**-791,871**	**-499,288**	**-695,303**	<-- =SUM(F21:F25)
27							
28	**Changes in cash balances**	**1,224,005**	**62,907**	**-234,129**	**52,691**	**-118,531**	<-- =F12+F18+F26
29							
30	**Supplemental disclosure of cash flow information**						
31	Cash paid during the period for						
32	Income taxes	255,043	175,972	314,735	283,618	305,094	
33	Interest	83,553	83,551	70,351	57,151	57,910	
34							
35	Income tax rate	34.73%	26.20%	37.07%	35.92%	36.96%	<-- =F32/(F4+F32)

To turn these CSCF into free cash flows:

- We keep all the items under operating activities.
- In the section for Investing Activities, we delete items that are not related to operations. For example, we would delete “short-term investments, net” under Investing Activities—these represent the purchase and sale of financial assets.
- We completely ignore the cash flows under Financing Activities
- We add back after-tax net interest to the sum of the remaining items to neutralize the subtraction of interest from the net income.

	A	B	C	D	E	F	G
1	**ABC CORPORATION** **CSCF rewritten to Free Cash Flow (FCF)**						
2		**2008**	**2009**	**2010**	**2011**	**2012**	
3	**Operating Activities:**						
4	Net earnings	479,355	495,597	534,268	505,856	520,273	
5	Adjustments to reconcile net earnings to net cash provided by operating activities						
6	Add back depreciation and amortization	41,583	47,647	46,438	45,839	46,622	
7	Changes in operating assets and liabilities:						
8	Subtract increase in accounts receivable	9,387	25,951	-12,724	1,685	-2,153	
9	Subtract increase in inventories	-37,630	-22,780	-16,247	-15,780	-5,517	
10	Subtract increase in prepaid expenses and other assets	-52,191	13,573	16,255	14,703	-2,975	
11	Add increase in accounts payable, accrued expenses, pensions and other liabilities	29,612	51,172	6,757	40,541	60,255	
12	**Net cash provided by operating activities**	**470,116**	**611,160**	**574,747**	**592,844**	**616,505**	<-- =SUM(F4:F11)
13							
14	**Investing Activities:**						
15	Short-term investments, net						
16	Purchases of property, plant and equipment	-48,944	-70,326	-89,947	-37,044	-88,426	
17	Proceeds from dispositions of property, plant and equipment	197	6,956	22,942	6,179	28,693	
18	**Net cash used in investing activities**	**-53,747**	**-118,370**	**-67,005**	**-30,865**	**-59,733**	<-- =SUM(F15:F17)
19							
20	**Financing Activities:**						
21	Repayment of debt						
22	Proceeds from revolving credit facility borrowings						
23	Proceeds from the issuance of stock						
24	Dividends paid						
25	Stock repurchased						
26	**Net cash used in financing activities**						<--
27							
28	**Free cash flow before interest adjustment**	**416,369**	**492,790**	**507,742**	**561,979**	**556,772**	<-- =F12+F18+F26
29	**Add back after-tax net interest**	**54,537**	**61,658**	**44,271**	**36,620**	**36,504**	<-- =(1-F37)*F35
30	**Free cash flow (FCF)**	**470,906**	**554,448**	**552,013**	**598,599**	**593,276**	<-- =F28+F29
31							
32	**Supplemental disclosure of cash flow information**						
33	Cash paid during the period for						
34	Income taxes	255,043	175,972	314,735	283,618	305,094	
35	Interest	83,553	83,551	70,351	57,151	57,910	
36							
37	Income tax rate	34.73%	26.20%	37.07%	35.92%	36.96%	<-- =F34/(F4+F34)

In Chapter 4 we discuss how these historical free cash flows can be used as the basis for cash flow projections. One valuation of a company might look like this:

	A	B	C	D	E	F	G	H
1	**ABC CORP. VALUATION**							
2	Free cash flow (FCF) year ending 31 Dec. 2012	593,276	<-- 593275.77278229					
3	Growth rate of FCF, years 1-5	8.00%	<-- Optimistic about short-term growth					
4	Long-term FCF growth rate	5.00%	<-- More pessimistic about long-term growth					
5	Weighted average cost of capital, WACC	10.70%						
6								
7	**Year**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	
8	FCF		640,738	691,997	747,357	807,145	871,717	<-- =F8*(1+$B$3)
9	Terminal value						16,057,940	<-- =G8*(1+B4)/(B5-B4)
10	Total		640,738	691,997	747,357	807,145	16,929,657	<-- =G8+G9
11								
12	Enterprise value	13,063,055	<-- =NPV(B5,C10:G10)*(1+B5)^0.5					
13	Add back initial cash and marketable securities	73,697	<-- From current balance sheet					
14	Subtract out 2012 financial liabilities	1,379,106	<-- From current balance sheet					
15	Equity value	11,757,646	<-- =B12+B13-B14					
16	Per share (1 million shares outstanding)	11.76	<-- =B15/1000000					

2.8 Free Cash Flows Based on Pro Forma Financial Statements

Another way to project free cash flows is to build a set of *predictive* financial statements based on our understanding of the company and its financial statements. We discuss the construction of such a model in Chapter 5 and give a fully worked-out example for Caterpillar in Chapter 6. A typical model might look like the following:

	A	B	C	D	E	F	G
1	**PRO FORMA FINANCIAL MODEL**						
2	Sales growth	10%					
3	Current assets/Sales	15%					
4	Current liabilities/Sales	8%					
5	Net fixed assets/Sales	77%					
6	Costs of goods sold/Sales	50%					
7	Depreciation rate	10%					
8	Interest rate on debt	10.00%					
9	Interest paid on cash and marketable securities	8.00%					
10	Tax rate	40%					
11	Dividend payout ratio	40%					
12							
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
14	**Income statement**						
15	Sales	1,000	1,100	1,210	1,331	1,464	1,611
16	Costs of goods sold	(500)	(550)	(605)	(666)	(732)	(805)
17	Interest payments on debt	(32)	(32)	(32)	(32)	(32)	(32)
18	Interest earned on cash and marketable securities	6	9	14	20	26	33
19	Depreciation	(100)	(117)	(137)	(161)	(189)	(220)
20	Profit before tax	374	410	450	492	538	587
21	Taxes	(150)	(164)	(180)	(197)	(215)	(235)
22	Profit after tax	225	246	270	295	323	352
23	Dividends	(90)	(98)	(108)	(118)	(129)	(141)
24	Retained earnings	135	148	162	177	194	211
25							
26	**Balance sheet**						
27	Cash and marketable securities	80	144	213	289	371	459
28	Current assets	150	165	182	200	220	242
29	Fixed assets						
30	At cost	1,070	1,264	1,486	1,740	2,031	2,364
31	Depreciation	(300)	(417)	(554)	(715)	(904)	(1,124)
32	Net fixed assets	770	847	932	1,025	1,127	1,240
33	**Total assets**	1,000	1,156	1,326	1,513	1,718	1,941
34							
35	Current liabilities	80	88	97	106	117	129
36	Debt	320	320	320	320	320	320
37	Stock	450	450	450	450	450	450
38	Accumulated retained earnings	150	298	460	637	830	1,042
39	**Total liabilities and equity**	1,000	1,156	1,326	1,513	1,718	1,941

Using the definition of free cash flows from section 2.5:

	A	B	C	D	E	F	G
41	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
42	**Free cash flow calculation**						
43	Profit after tax		246	270	295	323	352
44	Add back depreciation		117	137	161	189	220
45	Subtract increase in current assets		(15)	(17)	(18)	(20)	(22)
46	Add back increase in current liabilities		8	9	10	11	12
47	Subtract increase in fixed assets at cost		(194)	(222)	(254)	(291)	(333)
48	Add back after-tax interest on debt		19	19	19	19	19
49	Subtract after-tax interest on cash and mkt. securities		(5)	(9)	(12)	(16)	(20)
50	**Free cash flow**		176	188	201	214	228

We can now use these free cash flows to compute the enterprise value of the firm (row 62 below) and the value of its shares (cell B67):

	A	B	C	D	E	F	G	H
53	**Valuing the firm**							
54	Weighted average cost of capital	20%						
55	Long-term free cash flow growth rate	5%						
56								
57	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
58	FCF		176	188	201	214	228	
59	Terminal value						1,598	<-- =G58*(1+B55)/(B54-B55)
60	Total		176	188	201	214	1,826	
61								
62	Enterprise value, present value of row 60	1,348	<-- =NPV(B54,C60:G60)*(1+B54)^0.5					
63	Add in initial (year 0) cash and mkt. securities	80	<-- =B27					
64	Asset value in year 0	1,428	<-- =B63+B62					
65	Subtract out value of firm's debt today	(320)	<-- =-B36					
66	Equity value	1,108	<-- =B64+B65					
67	Share value (100 shares)	11.08	<-- =B66/100					

2.9 Summary

In this chapter we have introduced four enterprise-value valuation methods:

- The book value approach values the firm's enterprise value using its balance sheet numbers, appropriately rearranged.
- The efficient markets approach substitutes, where possible, market values for financial assets and liabilities instead of their book values, and then makes appropriate adjustments to the valuation of the firm's real assets.

• One approach to discounting the firm's free cash flows (FCFs) bases the estimates of future FCFs on the firm's consolidated statement of cash flows. These FCFs are then discounted at the appropriate weighted average cost of capital (WACC, Chapter 3).

• A second approach to discounting the firm's free cash flows constructs the FCFs from a model of the firm's projected future accounting statements (pro forma accounting statements). As in the previous bullet, these FCFs are discounted at the WACC.

Exercises

1. Three years of balance sheets for Cisco are given on the disk with this book. Restate these balance sheets so that the accounting enterprise value is on the left side.

2. Below are some year-end numbers for Cisco's equity. Restate the enterprise value in market terms.

	A	B	C	D	E
1	**CISCO EQUITY DATA**				
2		27-Jul-12	29-Jul-11	30-Jul-10	
3	Shares outstanding				
4	Price per share	15.69	15.97	23.07	
5	Shares outstanding (milions)	5,370	5,529	5,732	
6	**Market cap (millions)**	**84,255**	**88,298**	**132,237**	<-- =D4*D5

3. Examine Cisco's consolidated statement of cash flows (on the disk that accompanies this book) and transform this into a free cash flow.

4. Use the template for the ABC Corp. valuation in section 2.7 to value Cisco stock. Assume that the weighted average cost of capital for Cisco is 12.6%, the growth rate for years 1–5 is 4%, and that the long-term growth rate is 0%. (Details and template on the disk that accompanies this book.)

3 Calculating the Weighted Average Cost of Capital (WACC)

3.1 Overview

In this chapter we discuss the calculation of the firm's weighted average cost of capital (WACC). The WACC has two important uses in finance:

- When used as the discount rate for a firm's anticipated free cash flows (FCFs), the WACC gives the enterprise value of the firm. FCF is discussed in Chapter 2 and again at length in Chapters 4, 5, and 6—at this point it suffices to say that the FCF is the cash flow generated by the firm's core business activities. These chapters also show how to apply the WACC to the valuation of firms.
- The WACC is also the appropriate risk-adjusted discount rate for firm projects whose riskiness is similar to the average riskiness of the firm's cash flows. When used in this context, the WACC is often referred to as the firm's "hurdle rate."

The WACC is a weighted average of the firm's cost of equity r_E and its cost of debt r_D, with the weights created by the market values of the firm's equity (E) and debt (D):

$$WACC = \frac{E}{E+D} r_E + \frac{D}{E+D} r_D (1 - T_C)$$

where

E = *market value of the firm's equity*

D = *market value of the firm's debt*

T_C = *firm's corporate tax rate*

r_E = *firm's cost of equity*

r_D = *firm's cost of debt*

This chapter discusses the computation of the five components of the WACC—the market value of the firm's equity and debt E and D, the firm's tax rate T_C, the firm's cost of debt r_D, and the cost of equity r_E. We finish by showing detailed examples of how to compute the firm's WACC. The reader should be warned that the application of the models discussed requires a good deal of judgment—computing the WACC is equal parts science and art!

The main technical problem is the computation of the firm's cost of equity r_E. We consider two models for calculating the cost of equity r_E, the discount rate applied to equity cash flows.

- The Gordon model calculates the cost of equity based on the anticipated cash flows paid to the shareholders of the firm. In the implementation of the Gordon model, dividends growing at a constant future rate are most commonly used as the anticipated shareholder cash flows. We explore two variations of the model: multiple future growth rates and total equity cash flows.
- The capital asset pricing model (CAPM) calculates the cost of equity based on the correlation between the firm's equity returns and the returns of a large, diversified, market portfolio. Variations on this model include the tax framework in which the model is defined.

The other problematic component of the cost of capital is the cost of debt r_D, the anticipated future cost of the firm's borrowing. This book contains three models to calculate the cost of debt; two of these models are discussed in this chapter and a third method is discussed separately in Chapter 28.

- The cost of debt r_D is most commonly computed by using the firm's current net interest payments divided by its average net debt (net debt: debt minus cash and marketable securities).
- An alternative method is to compute r_D by imputing the firm's cost of debt from a rating-adjusted yield curve.
- Finally, we can compute the expected return on the firm's bonds as a proxy for its cost of debt; we discuss this method separately in Chapter 28.

A terminological note: "Cost of capital" is a synonym for the "appropriate discount rate" to be applied to a series of cash flows. In finance "appropriate" is most often a synonym for "risk-adjusted." Hence another name for the cost of capital is the "risk-adjusted discount rate" (RADR).

The Remainder of This Chapter

In the following sections we discuss the various components of the WACC with examples of how they might be computed:

- Section 3.2: Computing the equity value, *E*.
- Section 3.3: Computing the value of the firm's debt, *D*.

- Section 3.4: Computing the firm's corporate tax rate T_C.
- Section 3.5: Computing the firm's cost of debt, r_D.
- Sections 3.6–3.9: Computing the firm's cost of equity, r_E. We show how to use both the Gordon dividend model and the capital asset pricing model (CAPM) to compute r_E. Each model has a number of twists and variations, discussed in these sections.
- Sections 3.10–3.11: Computing the expected return on the market $E(r_M)$ and the risk-free rate r_f in the CAPM.
- Sections 3.12–3.15: Three worked-out cases of the computation of the WACC. We present a unified template that will help you make sense of the WACC computation.
- Section 3.16: Discussion of problems with using the dividend model and the CAPM, including the case of the determination of the WACC for non-marketed firms.

3.2 Computing the Value of the Firm's Equity, *E*

Of all the computations related to the WACC, computing the value of the firm's equity is the easiest: As long as the company is publicly listed, take *E* to be the product of the number of shares outstanding times the current value per share.

As an example, consider El Paso Pipeline Partners (EPB), a New York Stock Exchange company that owns gas pipelines and gas storage facilities. On 29 June 2012, EPB has 205.7 million shares outstanding, each trading at $33.80. The equity value of the company is $6.953 billion.[1]

	A	B	C
1	**COMPUTING THE VALUE OF EQUITY, E, FOR EL PASO PIPELINE PARTNERS (EPB)**		
2	Shares outstanding	205.70	<-- Million
3	Share price, 29 June 2012	33.80	
4	Equity value ("market cap")	6,953	<-- =B3*B2, million $

1. Most market traders refer to this number as the "market capitalization" or "market cap."

3.3 Computing the Value of the Firm's Debt, *D*

We compute the value of the firm's debt by the market value of its financial debt minus the market value of its excess liquid assets. A common approximation for this number is to take the balance sheet value of the firm's debt minus the value of the firm's cash balances and minus the value of its marketable securities. Here's an example for Kroger:

	A	B	C	D
1	**KROGER, COMPUTING NET DEBT (thousand $)**			
2		2011	2010	
3	Cash	825,000	188,000	
4	Marketable securities	0	0	
5				
6	Short-term and current portion of long-term debt	588,000	1,315,000	
7	Long-term debt	7,304,000	6,850,000	
8				
9	Net debt	7,067,000	7,977,000	<-- =SUM(C6:C7)-SUM(C3:C4)

For purposes of computing the weighted average cost of capital, our definition of debt excludes other debt-like items such as pension liabilities and deferred taxes. Though we consider these items as debts, it is hard to attach a cost to them; we prefer to approximate the WACC by using only financial obligations net of liquid assets.

It is not uncommon for a company to have negative net debt—this occurs when the company has more cash and marketable securities than debt. When this occurs, we set *D* in the WACC computation to be a negative number. Both Intel and Whole Foods Markets are examples:

	A	B	C	D
1	**INTEL HAS NEGATIVE NET DEBT (million $)**			
2		**2010**	**2011**	
3	Cash	5,498	5,065	
4	Marketable securities	16,387	9,772	
5				
6	Short-term debt and current portion of long-term debt	38	247	
7	Long-term debt	2,077	7,084	
8	**Net debt**	**-19,770**	**-7,506**	<-- =SUM(C6:C7)-SUM(C3:C4)
9				
10	**WHOLE FOODS HAS NEGATIVE NET DEBT (thousand $)**			
11		**2010**	**2011**	
12	Cash	218,798	303,960	
13	Marketable securities	329,738	442,320	
14				
15	Short-term debt and current portion of long-term debt	410	466	
16	Long-term debt	508,288	17,439	
17	**Net debt**	**-39,838**	**-728,375**	<-- =SUM(C15:C16)-SUM(C12:C13)

3.4 Computing the Firm's Tax Rate, T_C

In the WACC formula, T_C should measure the firm's *marginal* tax rate, but it is common to measure it by computing the firm's *reported* tax rate. Usually this should cause no problems, as the following example shows:

	A	B	C	D	E
1	**WHOLE FOODS MARKET TAX RATE**				
2		**2009**	**2010**	**2011**	
3	Income before tax	250,942	411,781	551,712	
4	Income tax expense	104,138	165,948	209,100	
5	**Tax rate, T_C**	**41.50%**	**40.30%**	**37.90%**	<-- =D4/D3

The tax rate for Whole Foods is reasonably stable at 38% to 41%. In our WACC computation we would most likely use the current tax rate or the average over the past several years.

Sometimes, however, this doesn't work, as the following example shows:

	A	B	C	D	E
1	**MERCK TAX RATE**				
2		**2009**	**2010**	**2011**	
3	Income before taxes	15,290,000	1,653,000	7,334,000	
4	Income tax expense	2,268,000	671,000	942,000	
5	**Tax rate, T_C**	**14.83%**	**40.59%**	**12.84%**	<-- =J4/J3

Companies like Merck are very good at placing their income in comfortable tax venues, and it appears that a reasonable estimate for the tax rate is somewhere between 13% and 15%. In 2010, a year of low income for Merck, these tax-planning strategies evidently did not work. Assuming that Merck's future profitability is indicated by the two good years 2009 and 2011, we would most likely assume that Merck's company's future tax rate T_C is in the range of the 2009 and 2011 tax rate.

3.5 Computing the Firm's Cost of Debt, r_D

We now turn to calculating the cost of debt r_D. In principle, r_D is the marginal cost to the firm (before corporate taxes) of borrowing an additional dollar. There are at least three ways of calculating the firm's cost of debt. We will state them briefly below and then go on to illustrate the application of two of the methods that although they may not be theoretically perfect are often used in practice:

- As a practical matter, the cost of debt can often be approximated by taking the *average cost* of the firm's existing debt. The problem with this method is that it runs the danger of confusing the *past costs* with the *future anticipated* cost of debt that we actually want to measure.
- We can use the yield of similar-risk, newly issued corporate securities. If a company is rated A and has mostly medium-term debt, then we can use the average yield on medium-term, A-rated debt as the firm's cost of debt. Note that this method is somewhat problematic because the yield on a bond is its *promised return*, whereas the cost of debt is the *expected return* on a firm's

debt. Since there is usually a risk of default, the promised return is generally higher than the expected return. Nevertheless, despite the problematics, this method is often a good compromise.

- We can use a model that estimates the cost of debt from data about the firm's bond prices, the estimated probabilities of default, and the estimated payoffs to bondholders in case of default. This method requires a lot of work and is mathematically non-trivial; we postpone its discussion until Chapter 28. For cost of capital calculations it would be used in practice only if the firm we are analyzing has significant amounts of risky debt.

The first two methods above are relatively easy to apply, and in many cases the problems or errors which are encountered in these methods are not critical.[2] As a matter of theory, however, both of these methods fail to make proper risk-adjustments for the cost of the firm's debt. The third method, which involves computing the expected return on a firm's debt, is more in line with standard financial theory, but it is also more difficult to apply. It may not, therefore, be worth the effort.

In the remainder of this section, we apply the first two of these methods to calculate the cost of debt for U.S. Steel and Merck.

Method 1: U.S. Steel's Average Cost of Debt

For U.S. Steel we compute the average cost of debt:

$$r_D = \frac{\textit{Current year's net interest paid}}{\textit{Average net debt over this and previous year}}$$

There are several aspects of our calculations worth noting:

2. It bears repeating that calculating the cost of capital requires a large number of assumptions and does not necessarily give a precise answer. Cost of capital estimation is not a *science*, it is an *art*. Users of cost of capital estimates should always do a sensitivity analysis around the numbers calculated. Given the data on the company you are analyzing, some sloppiness in the cost of capital calculations (with its accompanying savings in time) may be expedient.

	A	B	C	D	E
2	**UNITED STATES STEEL, COST OF DEBT**				
3		**2009**	**2010**	**2011**	
4	Cash	1,218,000	578,000	408,000	
5	Short-term investments	0	0	0	
6					
7	Short-term debt and current portion of long-term debt	19,000	216,000	400,000	
8	Long-term debt	3,828,000	3,517,000	3,345,000	
9					
10	Net debt	2,629,000	3,155,000	3,337,000	<-- =SUM(D6:D7)-SUM(D3:D4)
11	Interest	190,000	195,000	159,000	
12	**Implied cost of debt, r_D**		**6.74%**	**4.90%**	<-- =D10/AVERAGE(C9:D9)

- When calculating the average cost of debt r_D from the financial statements, it is important to include all financial debt, without distinguishing between short-term and long-term items.
- We treat liquid assets such as cash and cash equivalents as *negative debt* and subtract them from the firm's debt. The idea here is that the firm could use its cash to pay off part of its debt, so that the effective debt financing of the firm is its financial debt minus cash. However, the implementation of this particular piece of theory is largely a judgment call—we may not want to attribute all cash to the possibility of paying off debt, and we may want to compute the firm's cost of borrowing as opposed to the interest it earns on cash.

Were we to use the average cost of debt for U.S. Steel as a prediction of its future cost of debt r_D, we would most likely use the current cost $r_D = 4.90\%$ in the WACC computation. This is because we believe that historical costs of debt have little predictive power for future costs.

Cash Raises the Cost of Debt: The Case of Merck

When a firm has cash balances that earn less interest than the cost of borrowing, the average cost of debt, based on net interest and net debt is higher than the cost of borrowing. To see this, assume that the interest rate on cash is less by ε than the interest rate paid on debt:

$$\begin{aligned}\begin{matrix}\text{Average}\\ \text{cost of debt}\end{matrix} &= \frac{\text{Interest paid} - \text{Interest earned}}{\text{Debt} - \text{Cash}}\\ &= \frac{Debt * i_{Debt} - Cash * i_{Cash}}{Debt - Cash} = \frac{Debt * i_{Debt} - Cash * (i_{Debt} - \varepsilon)}{Debt - Cash}\\ &= \frac{(Debt - Cash) * i_{Debt} + \varepsilon * Cash}{Debt - Cash}\\ &= i_{Debt} + \frac{Cash}{Debt - Cash} * \varepsilon > i_{Debt}\end{aligned}$$

Here's a somewhat dramatic example for Merck:

	A	B	C	D	E
1	**MERCK, COST OF DEBT r_D**				
2		**2009**	**2010**	**2011**	
3	Cash	9,311,000	10,900,000	13,531,000	
4	Short-term investments	293,000	1,301,000	1,441,000	
5	Total liquid assets	9,604,000	12,201,000	14,972,000	<-- =D4+D3
6					
7	Short-term debt and current portion of long-term debt	1,379,000	2,400,000	1,990,000	
8	Long-term debt	16,095,000	15,482,000	15,525,000	
9	Total financial debt	17,474,000	17,882,000	17,515,000	<-- =D7+D8
10					
11	Net debt	7,870,000	5,681,000	2,543,000	<-- =D9-D5
12	Interest income	210,000	83,000	199,000	
13	Interest expense	460,000	715,000	749,000	
14	Net interest	250,000	632,000	550,000	<-- =D13-D12
15	Implied cost of debt, r_D		9.33%	13.38%	<-- =D14/AVERAGE(C11:D11)
16					
17	Interest rate earned		0.76%	1.46%	<-- =D12/AVERAGE(SUM(D3:D4),SUM(C3:C4))
18	Interest rate paid		4.04%	4.23%	<-- =D13/AVERAGE(SUM(D7:D8),SUM(C7:C8))

In 2011 Merck's cost of borrowing was 4.23%, and the company earned a respectable 1.46% on its massive reserves of cash and short-term investments. We might naively expect that this means that the average net cost of Merck's debt is between these two numbers. Instead the computations show that r_D = 13.38%!

$$r_D = \frac{\text{Average net interest paid}_{2010-11}}{\text{Average net debt}_{2010-11}} = \frac{550,000}{4,112,000} = 13.38\%$$

This estimate of r_D reflects the costs of holding large reserves of liquid assets which provide such low financial returns. From a purely financial point of view, Merck would have benefited its shareholders by using the liquid assets to repurchase its debt or by paying them out as either dividends or share repurchases.[3]

What number would we choose to represent the *marginal cost of borrowing* in the WACC equation? In large measure this depends on how we view the financial policy of Merck: If we view the firm as proceeding to build up cash reserves, while at the same time maintaining large amounts of financial debt, then 13.38% might be a reasonable measure of the cost of debt. If, on the other hand, we view the marginal debt financing of Merck as being with debt, without a parallel buildup of cash, then a number around 4% would better represent r_D.[4]

Method 2: r_D as the Rating-Adjusted Yield for Merck

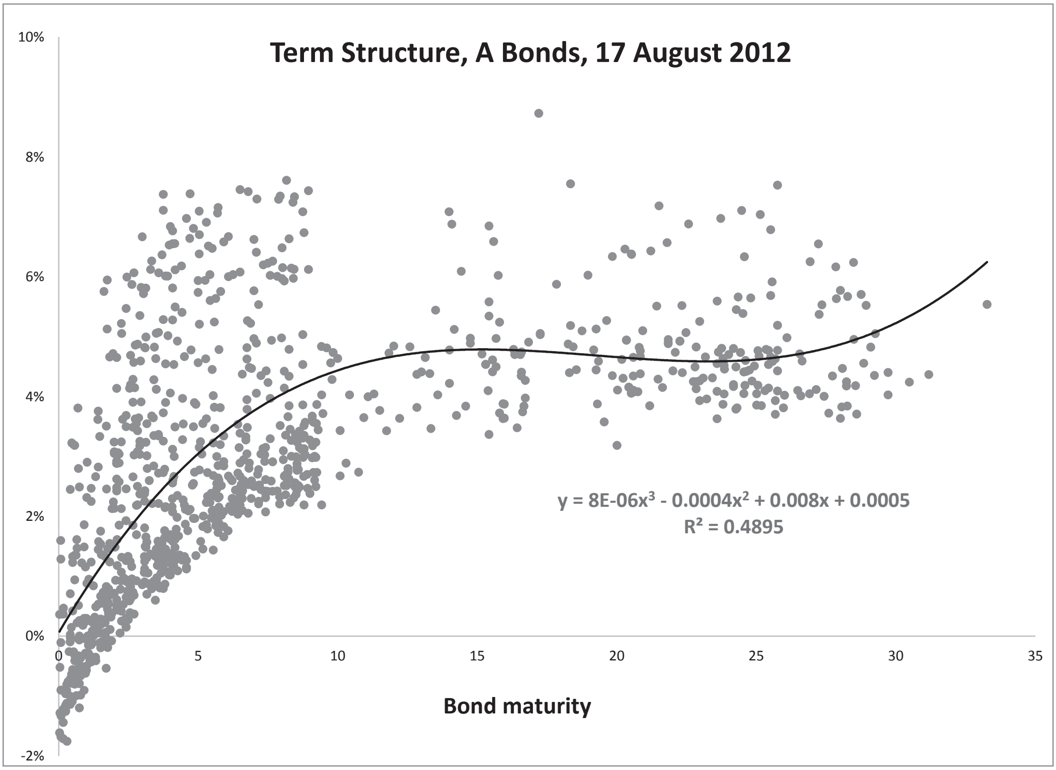
There is another way to measure the cost of debt for Merck's borrowing: We can impute the marginal cost of Merck's debt from a yield curve for the appropriate debt. Merck is rated A+ by Fitch, BBB+ by Standard and Poor's, and BAA2 by Moody's. From Yahoo we gather data for more than 1,000 A-Fitch-rated bonds; in the screen shot below much of this data is hidden:

	A	B	C	D	E	F
1	**FITCH-RATED A BONDS, Friday 17 August 2012**					
2		Price	Coupon	Maturity	Time to maturity	Yield to maturity
3	CITIGROUP INC	103.88	5.63%	27-Aug-12	0.0274	0.36%
4	LINCOLN NATL CORP IND	105.42	5.65%	27-Aug-12	0.0274	-1.62%
5	GOLDMAN SACHS GROUP INC	104.68	5.70%	1-Sep-12	0.0411	-0.52%
6	WELLS FARGO & CO NEW	104.85	5.13%	1-Sep-12	0.0411	-1.28%
7	BANK OF AMERICA CORPORATION	102.91	5.38%	11-Sep-12	0.0685	1.60%
8	BANK OF AMERICA CORPORATION	102.81	4.88%	15-Sep-12	0.0795	1.29%
996	GOLDMAN SACHS GRP INC MTN BE	103.30	5.75%	15-Jul-41	28.9288	5.52%
997	HEWLETT PACKARD CO	118.45	6.00%	15-Sep-41	29.0986	4.83%
998	VERIZON COMMUNICATIONS INC	106.63	4.75%	1-Nov-41	29.2274	4.35%
999	AMGEN INC	101.52	5.15%	15-Nov-41	29.2658	5.05%
1000	ANHEUSER BUSCH COS INC	135.00	6.50%	1-May-42	29.7233	4.40%
1001	CATERPILLAR INC DEL	151.00	6.95%	1-May-42	29.7233	4.03%
1002	ANHEUSER BUSCH COS INC	138.85	6.50%	1-Feb-43	30.4795	4.24%
1003	BOEING CO	142.90	6.88%	15-Oct-43	31.1808	4.37%
1004	BELLSOUTH TELECOM	104.75	5.85%	15-Nov-45	33.2685	5.54%
1005	BELLSOUTH TELECOM	123.96	7.00%	1-Dec-95	83.3452	5.64%
1006	CITIGROUP INC	107.55	6.88%	15-Feb-98	85.5562	6.39%
1007	CUMMINS INC	99.25	5.65%	1-Mar-98	85.5945	5.69%

3. This statement ignores the *option value* of the liquid assets: their ability to provide Merck with financial flexibility.

4. In the case of Merck, as we shall see in section 3.14, none of this makes much difference, since Merck is virtually an all-equity company.

Graphing these data shows either that these data cover a wide range of true credit risks or that the market is far from efficient:



The polynomial regression line describes about 50% of the variability of the yields as a function of the time to maturity. Using this regression equation and assuming a 7-year average maturity for Merck's debt, the cost of borrowing for Merck is 3.96%.

	A	B	C
1	COMPUTING MERCK'S r_D FROM THE A-YIELD CURVE		
2	Average time to maturity (years)	7	
3	Yield	3.96%	<-- =0.000008*B2^3 - 0.0004*B2^2 + 0.008*B2 + 0.0005

3.6 Two Approaches to Computing the Firm's Cost of Equity, r_E

The equation for the weighted average cost of capital is $WACC = E/(E + D)*r_E + E/(E + D)*r_D*(1 - T_C)$. Thus far in this chapter we have discussed the estimation of four of the five parameters of the WACC equation: E, D, T_C, r_D. We now come to the most problematic of the computations related to the WACC parameters—the computation of the cost of equity r_E. There are two approaches to r_E that can readily be computed:

• The Gordon dividend model computes r_E based on current dividend Div_0, current stock price P_0, and the anticipated growth of future dividends g:

$$r_E = \frac{Div_0(1+g)}{P_0} + g$$

• The capital asset pricing model (CAPM) computes r_E based on the risk-free rate r_f, the expected return on the market $E(r_M)$, and a firm-specific risk measure β:

$$r_E = r_f + \beta\left[E(r_M) - r_f\right]$$

where

r_f = *the market risk-free rate of interest*

$E(r_M)$ = *the expected return on the market portfolio*

β = *a firm-specific risk measure* = $\dfrac{Cov(r_{stock}, r_M)}{Var(r_M)}$

Each model has its variations and problems, which are discussed (*ad nauseam*?) in the next two sections.

3.7 Implementing the Gordon Model for r_E

The Gordon dividend model derives the cost of equity from the following deceptively simple statement:

> The value of a share is the present value of the future anticipated dividend stream from the share, where the future anticipated dividends are discounted at the appropriate risk-adjusted cost of equity r_E.[5]

5. This model is named after M. J. Gordon, who first published this formula in a paper entitled "Dividends, Earnings and Stock Prices," *Review of Economics and Statistics*.

The simplest application of the Gordon model is the case where the anticipated future growth rate of dividends is constant. Suppose that the current stock price is P_0, the current dividend is Div_0, and the anticipated growth rate of future dividends is g. The Gordon model states that the stock price equals the discounted (at the appropriate cost of equity r_E) future dividends:

$$P_0 = \frac{Div_0(1+g)}{1+r_E} + \frac{Div_1 * (1+g)^2}{(1+r_E)^2} + \frac{Div_1 * (1+g)^3}{(1+r_E)^3} + \frac{Div_1 * (1+g)^4}{(1+r_E)^4} \cdots$$
$$= \sum_{t=1}^{\infty} \frac{Div_0 * (1+g)^t}{(1+r_E)^t}$$

Provided that $|g| < r_E$, the expression $\sum_{t=1}^{\infty} \frac{Div_0 * (1+g)^t}{(1+r_E)^t}$ can be reduced to $\frac{Div_0(1+g)}{r_E - g}$ (we will spare you this derivation, which is based on a formula for geometric series usually studied in high school). Thus—given a constant anticipated dividend growth rate, we derive the Gordon model cost of equity:

$$P_0 = \frac{Div_0(1+g)}{r_E - g}, \quad \text{provided} \quad |g| < r_E$$

Solving the above equation for r_E gives the Gordon formula for the cost of equity:

$$r_E = \frac{Div_0(1+g)}{P_0} + g, \quad \text{provided} \quad |g| < r_E$$

Note the proviso at the end of this formula: In order for the infinite sum on the first line of the formula to have a finite solution, the growth rates of the dividends must be less than the discount rate. In our discussion of the Gordon model with supernormal growth rates (see below) we return to the case where this is not true.

To apply this formula, consider a firm whose current dividend is Div_0 = \$3 per share, whose share price is P_0 = \$50. Suppose the dividend is anticipated to grow by 12% per year. Then the firm's cost of equity r_E is 17.6%:

	A	B	C
1	**THE GORDON MODEL COST OF EQUITY**		
2	Current share price, P_0	60	
3	Current dividend, Div_0	3	
4	Anticipated dividend growth rate, g	12%	
5	Gordon model cost of equity, r_E	17.60%	<-- =B3*(1+B4)/B2+B4

Using the Gordon Model to Compute the Cost of Equity for Merck

We apply the Gordon model to Merck, whose 10-year dividend history is given below (note that some of the data has been hidden):

	A	B	C	D	E	F
1	**MERCK DIVIDEND HISTORY**					
2	Date	Dividend per share		Dividend growth		
3	4-Sep-02	0.36		Whole period		
4	4-Dec-02	0.36		Quarterly growth	0.39%	<-- =(B43/B3)^(1/40)-1
5	5-Mar-03	0.36		Annual growth	1.55%	<-- =(1+E4)^4-1
6	4-Jun-03	0.36				
7	20-Aug-03	2.88		Last 5 years		
8	3-Sep-03	0.37		Quarterly growth	0.50%	<-- =(B43/B23)^(1/20)-1
9	3-Dec-03	0.37		Annual growth	2.02%	<-- =(1+E8)^4-1
10	3-Mar-04	0.37				
11	2-Jun-04	0.37				
12	1-Sep-04	0.38				
40	13-Sep-11	0.38				
41	13-Dec-11	0.42				
42	13-Mar-12	0.42				
43	13-Jun-12	0.42				

The annualized growth rate of Merck's historical dividends may be either 1.55% or 2.02%, depending on the period taken. For purposes of computing the cost of equity r_E, the question is which of these rates better predicts future anticipated dividend growth rates.[6] In the spreadsheet below, we allow for both possibilities. The calculations use Merck's stock price at the end of June 2012, P_0 = \$41.75:

6. Or perhaps neither does! Perhaps we are better off using another story altogether to predict *future anticipated* dividend growth? We could use a pro forma model (discussed in Chapter 5) to predict the firm's anticipated dividend payout.

	A	B	C
1	**COMPUTING MERCK'S r_E WITH THE GORDON MODEL**		
2	Merck stock price P_0, 29 June 2012	10-Feb-00	
3	Current dividend		
4	Quarterly	0.42	
5	Annualized dividend, Div_0	1.68	<-- =4*B4
6	Dividend growth rate, g		
7	Last 5 years	1.55%	
8	Last 10 years	2.02%	
9			
10	Gordon model cost of equity, r_E		
11	Using last 5 years' growth	5.64%	<-- =$B$5/$B$2*(1+B7)+B7
12	Using last 10 years' growth	6.13%	<-- =$B$5/$B$2*(1+B8)+B8

For all practical purposes, given the margin of error in our estimates of the future, these numbers are identical—remember that we are trying to predict future dividend growth based on past dividend payouts.

Adjusting the Gordon Model to Account for All Cash Flows to Equity

As illustrated above, the Gordon model is computed on a per-share basis and for dividends only. However, for purposes of valuing the firm's equity, the Gordon model should be extended to include all cash flows to equity. In addition to dividends, cash flows to equity include at least two additional components:

- Share repurchases now account for around 50% of the total cash disbursed by American corporations to their shareholders.[7]
- The issuance of stock by the firm is an important *negative cash flow to equity*. In many firms the most important instance of stock issuance is the exercise by employees of their stock options.

In order to account for these additional cash flows to equity, we have to rewrite the Gordon model in terms of total equity value. The basic valuation model of Gordon now becomes:

$$Market\ value\ of\ equity = \sum_{t=1}^{\infty} \frac{Cash\ flow\ to\ equity_0 * (1+g)^t}{(1+r_E)^t}$$

7. See "Corporate Payout Policy" by Harry DeAngelo, Linda DeAngelo, and Douglas J. Skinner, *Foundations and Trends in Finance*, 2008. Also available at www.ssrn.com.

where

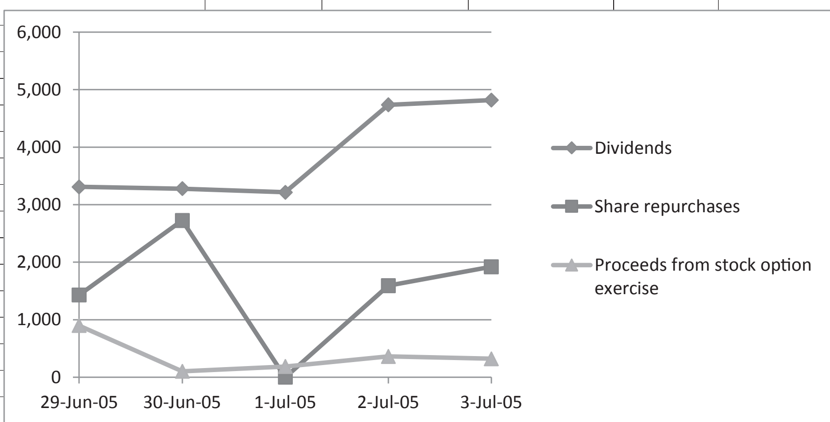
$$g = \textit{Anticipated growth rate of cash flow equity}$$

This gives the formula for the cost of equity r_E as:

$$r_E = \frac{\textit{Cash flow to equity}_0 (1+g)}{\textit{Market value of equity}} + g, \text{ if } |g| < r_E.$$

As an example we consider the data below for Merck:

	A	B	C	D	E	F
1	**GORDON MODEL FOR MERCK'S EQUITY PAYOUTS**					
2		**Dividends**	**Share repurchases**	**Proceeds from stock option exercise**	**Total equity payout**	
3	29-Jun-05	3,307	1,430	899	3,838	<-- =B3+C3-D3
4	30-Jun-05	3,279	2,725	102	5,901	
5	1-Jul-05	3,215	0	186	3,029	
6	2-Jul-05	4,734	1,593	363	5,964	
7	3-Jul-05	4,818	1,921	321	6,418	
8						
9	Growth	13.71%	<-- =(E7/E3)^(1/4)-1			
10						
11	Computing the Gordon model cost of equity r_E based on total equity payouts					
12	Shares outstanding (million)	3,041				
13	Price per share	41.75				
14	Market value of equity	126,955	<-- =B12*B13, $ million			
15						
16	Gordon model cost of equity, r_E	19.46%	<-- =E7*(1+B9)/B14+B9			
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						



If we assume that Merck's historic growth rate of cash flow to equity, 13.71%, will persist in the indefinite future, then its cost of equity is r_E = 19.46%.[8] This seems rather high. In the next subsection we present another variation of the Gordon model that may provide an answer.

"Supernormal Growth" and the Gordon Model

A basic condition of the Gordon formula $r_E = \frac{Div_0(1+g)}{P_0} + g$ is the condition $|g| < r_E$.[9] In finance examples, violations of $|g| < r_E$ usually occur for very fast-growing firms, in which—at least for short periods of time—we anticipate very high growth rates, so that $g > r_E$. If such "supernormal" growth were the case in the long run, the original dividend discount formula shows that P_0 would have an infinite value, since, when $g > r_E$, the expression $\sum_{t=1}^{\infty} \frac{Div_0 * (1+g)^t}{(1+r_E)^t} = \infty$. Thus a period of very high dividend growth rates (where $g > r_E$) must be followed by a period in which the long-term growth rate of dividends is less than the cost of equity, $g < r_E$.

Suppose that the firm is anticipated to pay high-growth dividends during periods 1, … , m, and that for subsequent periods the growth rate of dividends will be lower. We can write the discounted value of these anticipated future dividends as:

$$\begin{aligned} \textit{Share value today} &= \textit{Present value of dividends} \\ &= \underbrace{\sum_{t=1}^{m} \frac{Div_0 * (1+g_1)^t}{(1+r_E)^t}}_{\substack{\uparrow \\ \text{PV of } m \text{ years} \\ \text{of high-growth } g_1 \\ \text{dividends}}} + \underbrace{\sum_{t=m+1}^{\infty} \frac{Div_5 * (1+g_2)^{t-m}}{(1+r_E)^t}}_{\substack{\uparrow \\ \text{PV of remaining} \\ \text{normal-growth } g_2 \\ \text{dividends}}} \end{aligned}$$

The problem is usually to determine the cost of equity r_E from predicted growth rates. In the example below we use a VBA function **TwoStageGordon** to compute r_E that equalizes the values of both sides of the above equation.[10] The assumption is that a firm's share price is currently $P_0 = 30$, that its current dividend is $Div_0 = 3$, and that after 5 years of 35% growth in dividends, the dividend growth rate will slow to 8%. As you can see below, $r_E = 32.76\%$.

8. Most firms only report stock repurchases and employee stock exercises annually. Thus the only data available for these numbers is annual data, whereas dividends are reported quarterly and stock price data—used to compute the CAPM beta discussed in section 3.9—are available on a daily basis.

9. In this section we interpret Div_0 in the Gordon formula to denote either dividends per share or total equity payouts.

10. The construction of **TwoStageGordon** is discussed at the end of this section.

	A	B	C
1	**THE GORDON MODEL WITH TWO GROWTH RATES Using the TwoStageGordon function**		
2	Current dividend, Div_0	3.00	
3	Growth rate g_1, years 1-m ("supernormal")	35%	
4	Growth rate g_2, years 6 - ∞	8%	
5	Number of supernormal growth years, m	5	
6	Share price	30.00	
7	Cost of equity	32.76%	<-- =twostagegordon(B6,B2,B3,5,B4)

Implementing the Two-Stage Gordon Model for Merck

Below we use the two-stage model to compute the cost of equity for Merck. We assume that the 5-year growth rate of total equity payouts of 13.71% will hold only for the next 2 years, and that afterward the growth rate of equity payouts will be 5%. The cost of equity is $r_E = 11.20\%$. This is high, compared to the r_E computed for the Gordon model using only dividends (next section).

	A	B	C	D	E	F
1	**MERCK'S COST OF EQUITY USING THE TWO-STAGE GORDON MODEL**					
2		**Dividends**	**Share repurchases**	**Proceeds from stock option exercise**	**Total equity payout**	
3	29-Jun-05	3,307	1,430	899	3,838	<-- =B3+C3-D3
4	30-Jun-05	3,279	2,725	102	5,901	
5	1-Jul-05	3,215	0	186	3,029	
6	2-Jul-05	4,734	1,593	363	5,964	
7	3-Jul-05	4,818	1,921	321	6,418	
8						
9	Growth	13.71%	<-- =(E7/E3)^(1/4)-1			
10						
11	Computing the Gordon model cost of equity r_E based on total equity payouts and the 2-stage Gordon model					
12	Shares outstanding (million)	3,041				
13	Price per share	41.75				
14	Market value of equity	126,955	<-- =B12*B13, $ million			
15						
16	High growth rate, g_{high}					
17	Number of high-growth years, m	2				
18	Normal growth rate, g_{normal}	5.00%	<-- Author guess			
19						
20	Cost of equity, r_E using the function **twostagegordon**	11.20%	<-- =twostagegordon(B14,E7,B9,B17,B18)			

Technical Note

The function **TwoStageGordon** on the file with this chapter computes the cost of equity r_E for a two-stage Gordon model. The function computes the discount rate r_E, which equates the current share price to the present value of future equity cash flows:

```
Function TwoStageGordon(P0, Div0, Highgrowth, _
Highgrowthyrs, Normalgrowth)
    high = 1
    low = 0

    Do While (high - low) > 0.00001
    Estimate = (high + low) / 2
    factor = (1 + Highgrowth) / _
    (1 + Estimate)
    Term1 = Div0 * factor * (1 - factor ^ _
    Highgrowthyrs) / (1 - factor)
    Term2 = Div0 * factor ^ Highgrowthyrs * _
    (1 + Normalgrowth) / (Estimate - _
    Normalgrowth)
    If (Term1 + Term2) > P0 Then
        low = (high + low) / 2
        Else:  high = (high + low) / 2
    End If
    Loop
    TwoStageGordon = (high + low) / 2
End Function
```

3.8 The CAPM: Computing the Beta, β

The capital asset pricing model (CAPM) is the only viable alternative to the Gordon model for calculating the cost of capital. It is also the most widely used cost of equity model, the reasons being both its theoretical elegance and its implementational simplicity. The CAPM derives the firm's cost of capital from its covariance with the market return.[11] The classic CAPM formula for the firm's cost of equity is:

$$r_E = r_f + \beta\left[E(r_M) - r_f\right]$$

where

r_f = *the market risk-free rate of interest*

$E(r_M)$ = *the expected return on the market portfolio*

β = *a firm-specific risk measure* = $\dfrac{Cov(r_{stock}, r_M)}{Var(r_M)}$

In the remainder of this section we focus on measuring the firm's β; the next section shows how to apply the CAPM to find the firm's cost of equity r_E.

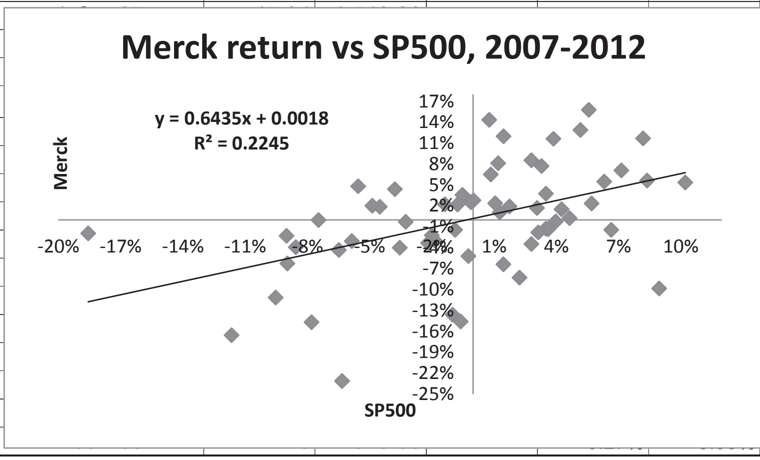
Beta Is the Regression Coefficient of the Firm's Stock Returns on the Market Returns

In the following spreadsheet, we show the 5 years of monthly prices and returns for Merck and for the S&P 500, which we take to proxy for the stock market as a whole. In cells B2:B4 we regress the Merck returns on those of the S&P 500:

$$\begin{aligned} r_{Merck,t} &= \alpha_{Merck} + \beta_{Merck} r_{SP,t} \\ &= -0.0018 + 0.6435 r_{SP,t}, \quad R^2 = 0.2245 \end{aligned}$$

11. The CAPM is discussed in detail in Chapters 8–11. At this point we outline the application of the model to finding the cost of capital without entering into the theory.

	A	B	C	D	E	F	G
1	**COMPUTING THE BETA FOR MERCK** **monthly returns for Merck and SP500, 2007-2012**						
2	Alpha	0.0018	<-- =INTERCEPT(E11:E70,F11:F70)				
3	Beta	0.6435	<-- =SLOPE(E11:E70,F11:F70)				
4	R-squared	0.2245	<-- =RSQ(E11:E70,F11:F70)				
5	t-statistic for alpha	0.2059	<-- =tintercept(E11:E70,F11:F70)				
6	t-statistic for beta	4.0979	<-- =tslope(E11:E70,F11:F70)				
7							
8		Prices			Returns		
9	**Date**	**Merck**	**SP500**		**Merck**	**SP500**	
10	1-Jun-07	39.90	1,503.35				
11	2-Jul-07	39.78	1,455.27		-0.30%	-3.25%	<-- =LN(C11/C10)
12	1-Aug-07	40.20	1,473.99		1.05%	1.28%	<-- =LN(C12/C11)
13	4-Sep-07	41.73	1,526.75		3.74%	3.52%	<-- =LN(C13/C12)
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							



Here's what we can learn from this regression:

• Merck's beta, β_{Merck}, shows the sensitivity of its stock return to the market return. It is calculated by the following formula:

$$\beta_{Merck} = \frac{Covariance(SP500\ returns, Merck\ returns)}{Variance(SP500\ returns)}$$

We can compute β either by using the formula above directly (cell B5 above) or by using the Excel **Slope** function (cell B4). Over the period covered, a 1% increase or decrease in the monthly returns of the S&P 500 was accompanied by a 0.6435% increase or decrease in Merck's returns. The statistic **TSlope** (cell B6) shows that the β_{Merck} is highly significant (see below for how this function was constructed).[12]

• Intel's alpha, α_{Merck}, shows that irrespective of changes in the S&P 500, the monthly return on Intel over the period was α_{Merck} = 0.18%. On an annual basis, this is 12*0.18 = 2.18%; this seems to indicate that, in the jargon of financial markets, Merck had positive performance over the period. Note, however, the **TIntercept** (cell B5): This function (its construction in Excel is discussed below) shows that the negative intercept is not significantly different from zero.

• The R^2 of the regression shows that 22.45% of the variation in Merck's returns is accounted for by variability in the S&P 500. An R^2 of 22% may seem low, but in the CAPM literature this is not uncommon. It says that roughly 22% of the variation in Merck's returns is explicable by the variation in the S&P 500 return. The rest of the variability in the Merck returns can be diversified away by including Merck shares in a diversified portfolio of shares. The average R^2 for stocks is approximately 30% to 40%, meaning that market factors account for approximately this percentage of a stock's variability, with factors idiosyncratic to the stock accounting for the rest. Merck's R^2, as you can see, is a bit on the low side—meaning that it has more idiosyncratic risk than an average stock.

The spreadsheet that accompanies this chapter shows three ways of doing the regression: One way is to use the functions **Intercept**, **Slope**, **Rsq**. A second method involves using the Excel functions **Covar** and **VarP**. A third way involves Excel's **Trendline** function. Having graphed the returns of Intel and the SP500 on an **XY Scatter** plot, we then do the following:

12. For the precise meaning of a *t*-statistic, you should refer to a good statistics text. For our purposes—a *t*-statistic over 1.96 indicates that with 95% probability the variable under discussion (when using **TIntercept** this is the intercept, or when using **TSlope** this is the slope) is significantly different from zero. Thus the *t*-statistic for the intercept of –0.2059 indicates that the intercept is not significantly different from zero, whereas the *t*-statistic for the slope of 4.0979 indicates that the slope is significantly different from zero.

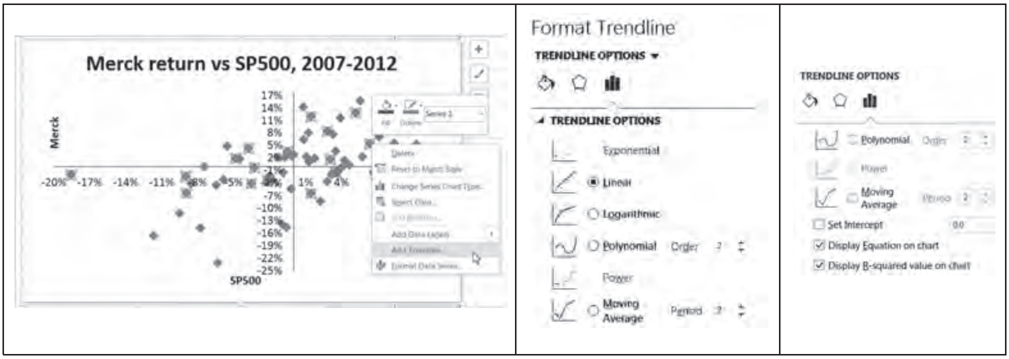


Figure 3.1
The sequence of commands for producing regression results from the **XY Scatter Plot** in Excel. Having marked the points, we right-click to select **Add Trendline** (left panel). We choose the linear regression (middle panel) and indicate that the regression equation and the R^2 should be printed on the graph.

The Home-Made Functions TIntercept and TSlope

The spreadsheet above uses two functions to compute the *t*-statistics for the intercept and slope. These functions are built on the **Linest** function discussed in Chapter 33. Applying **Linest** to the return data, we get:

	I	J	K	L
9		Cells J13:K17created with the formula {=LINEST(E11:E70,F11:F70,,1)}		
10				
11				
12		**Slope**	**Intercept**	
13	Slope -->	0.6435	0.0018	<-- Intercept
14	Standard error of slope -->	0.1570	0.0088	<-- Standard error of intercept
15	R-squared -->	0.2245	0.0682	<-- Standard error of y values
16	F statistic -->	16.7925	58.0000	<-- Degrees of freedom
17	SS_{xy} -->	0.0781	0.2696	<-- SSE = Residual sum of squares

By using the Excel function **Index** we define a VBA function **TIntercept** which divides the value of the intercept term produced by **Linest** (first row, second column of **Linest** output) by the standard error of the intercept (second row, second column). Here's the function for the *t*-statistic for the intercept:

```
Function tintercept(yarray, xarray)
    tintercept = Application.Index(Application. _
    LinEst(yarray, xarray, , 1), 1, 2) / _
    Application.Index(Application.LinEst(yarray, _
    xarray, , 1), 2, 2)
End Function
```

Similarly we can define a function **TSlope** that gives the *t*-statistic for the slope:

```
Function tslope(yarray, xarray)
    tslope = Application.Index(Application. _
    LinEst(yarray, xarray, , 1), 1, 1) / _
    Application.Index(Application.LinEst(yarray, _
    xarray, , 1), 2, 1)
End Function
```

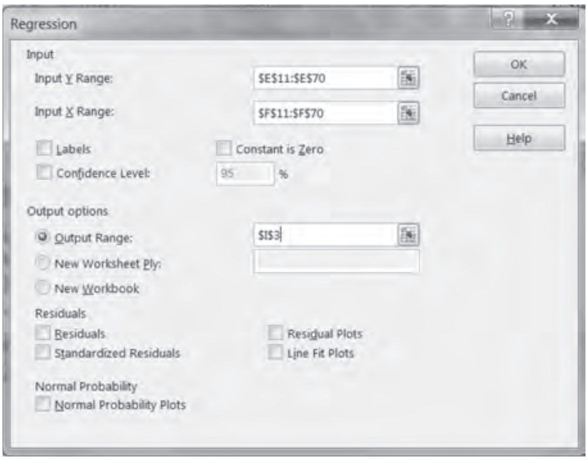
Both of these functions are embedded in the spreadsheet for this chapter.

Using Excel's Data Analysis Add-In

There's a fourth way to produce the regression output: By clicking on **Tools|Data Analysis|Regression**, we can use a sophisticated Excel routine that computes more statistics, including the *t*-statistics. The output produced by this routine is illustrated below:

	I	J	K	L	M	N	O	P	Q
3	SUMMARY OUTPUT								
4									
5	*Regression Statistics*								
6	Multiple R	0.473836264							
7	R Square	0.224520805							
8	Adjusted R Square	0.211150474							
9	Standard Error	0.068184238							
10	Observations	60							
11									
12	ANOVA								
13		*df*	*SS*	*MS*	*F*	*Significance F*			
14	Regression	1	0.078069683	0.078069683	16.79246425	0.000131232			
15	Residual	58	0.26964724	0.00464909					
16	Total	59	0.347716923						
17									
18		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
19	Intercept	0.001813136	0.008806331	0.205890007	0.837597818	-0.015814651	0.019440922	-0.015814651	0.019440922
20	X Variable 1	0.643502885	0.157033851	4.097860936	0.000131232	0.329165479	0.957840291	0.329165479	0.957840291

We used **Data|Data Analysis|Regression** to produce this output. The settings are shown below:



While **Data|Data Analysis|Regression** produces a lot of data, it has one major drawback: The output is not automatically updated when the underlying data changes. For this reason we prefer to use the other methods illustrated.

3.9 Using the Security Market Line (SML) to Calculate Merck's Cost of Equity, r_E

In the capital asset pricing model, the security market line (SML) is used to calculate the risk-adjusted cost of capital. In this section we consider two SML formulations. The difference between these two methods has to do with the way taxes are incorporated into the cost of capital equation.

Method 1: The Classic SML

The classic CAPM formula uses a security market line (SML) equation that ignores taxes:

$$\text{Cost of equity}, r_E = r_f + \beta\left[E(r_M) - r_f\right]$$

Here r_f is the risk-free rate of return in the economy and $E(R_M)$ is the expected rate of return on the market. The choice of values for the SML parameters is often problematic. A common approach is to choose:

- r_f equal to the risk-free interest rate in the economy (for example, the yield on Treasury bills). We leave the question of whether to use the short-term or long-term rate open until section 3.11. For the moment, for illustrative purposes, we use $r_f = 2\%$.
- $E(r_M)$ equal to the historic average of the market return, defined as the average return of a broad-based market portfolio. There is an alternative approach based on market multiples; both of these are discussed below. For the current section, we use $E(r_M) = 8\%$.

The following spreadsheet illustrates the classic CAPM cost of equity computation for Merck's cost of equity:

	A	B	C
1	**COMPUTING THE COST OF EQUITY FOR MERCK** **Classic CAPM: $r_E = r_f + \beta$*$[E(r_M) - r_f]$**		
2	Merck beta, β	0.6435	
3	Risk-free rate, r_f	2.00%	
4	Expected market return, $E(r_M)$	8.00%	
5	Merck cost of equity, r_E	5.86%	<-- =B3+B2*(B4-B3)

Method 2: The Tax-Adjusted SML

The classic CAPM approach makes no allowance for taxation. Benninga-Sarig (1997) show that the SML has to be adjusted for the marginal corporate tax rate in the economy.[13] Denoting the corporate tax rate by T_C, the tax-adjusted SML is:

$$\text{Cost of equity} = r_f(1-T_C) + \beta\left[E(r_M) - r_f(1-T_C)\right]$$

This formula can be applied by substituting $r_f(1 - T_C)$ for r_f in the classic CAPM. Note that the Tax-adjusted cost of equity has a lower intercept and a higher slope than the classic CAPM:

- The intercept is $r_f(1 - T_C)$ instead of r_f. This intercept is lower than the r_f intercept of the classic CAPM.
- The slope is $E(r_M) - r_f(1 - T_C)$ instead of $E(r_M) - r_f$. This slope can be written as the classic CAPM slope plus $T_C r_f$: $E(r_M) - r_f(1 - T_C) = [E(r_M) - r_f] + T_C r_f$

Another way to write the tax-adjusted cost of equity is:

$$\begin{aligned}\text{Tax-adjusted cost of equity} &= r_f(1-T_C) + \beta\left[E(r_M) - r_f(1-T_C)\right] \\ &= \underbrace{r_f + \beta\left[E(r_M) - r_f\right]}_{\text{Classic CAPM } r_E} + T_C r_f[\beta - 1]\end{aligned}$$

This rewriting makes clear that the difference between the classic r_E and the tax-adjusted r_E is a function of the corporate tax rate T_C, the risk-free rate r_f, and the equity beta β.[14] For Merck, the tax-adjusted approach gives a somewhat higher cost of equity:

13. The logic of the Benninga-Sarig approach is outlined in our book *Corporate Finance: A Valuation Approach* (McGraw-Hill, 1997). A more formal derivation of the model is given in "Risk, Returns and Values in the Presence of Differential Taxation," co-authored with Oded Sarig. *Journal of Banking and Finance*, 2003.

14. As this book is written, short-term, risk-free rates are close to zero, so that the difference between the tax-adjusted and classic CAPM is minimal. Presumably this will change at some point in the future.

	A	B	C
1	**COMPUTING THE COST OF EQUITY FOR MERCK** **Tax-adjusted CAPM: $r_E = r_f*(1-T_C) + \beta*[E(r_M) - r_f*(1-T_C)]$**		
2	Merck beta, β	0.6435	
3	Merck tax rate, T_C	12.84%	<-- ='Merck tax rate'!D5
4	Risk-free rate, r_f	2.00%	
5	Expected market return, $E(r_M)$	8.45%	
6	Merck tax-adjusted cost of equity, r_E	6.06%	<-- =B4*(1-B3)+B2*(B5-B4*(1-B3))

Although the tax-adjusted CAPM is more consistent with an economy with taxation, we confess that—given the uncertainties surrounding cost of capital computations—the difference between the classic CAPM and the tax-adjusted CAPM may not be worth the trouble.

3.10 Three Approaches to Computing the Expected Return on the Market, $E(r_M)$

Two critical questions remain in the computation of the cost of equity r_E using the CAPM:

- What is the expected return on the market, $E(r_M)$? Should it be computed from historical data? (And if so, how long should the data series be?) Or perhaps it can be computed from current market data without resort to history?
- What is the risk-free rate, r_f? Should it be a short-term or a long-term rate?

In this section we deal with the first question, leaving the computation of r_f for section 3.11. There are three major approaches to computing $E(r_M)$:

- The historical return on a major market index
- The historical market risk premium on the market index
- The Gordon model

All three approaches are illustrated in this section, and their effect on computing Merck's cost of equity is illustrated at the end of this section.

$E(r_M)$ as the Historical Average Return on a Market Portfolio

A simple approach to computing $E(r_M)$ is to take it as the average of the historical returns of a major market index. In the computation below we illustrate this approach by using Vanguard's 500 Index Fund as a proxy for the market.[15] The annualized return on this fund since 1987 is 8.27%. We can take this as a reliable proxy for the historical annual average return from holding the S&P 500:

	A	B	C	D
1	**MEASURING E(r$_M$) USING HISTORICAL DATA** **Derived from prices for the Vanguard 500 Index Fund (symbol: VFINX)** **These prices include dividends; April 1987 - June 2012**			
2	Average monthly return	0.69%	<-- =AVERAGE(C10:C311)	
3	Monthly standard deviation	4.58%	<-- =STDEV(C10:C311)	
4				
5	Annualized return	8.27%	<-- =12*B2	
6	Annualized standard deviation	15.87%	<-- =SQRT(12)*B3	
7				
8	**Date**	**Price**	**Return**	
9	1-Apr-87	15.66		
10	1-May-87	15.82	1.02%	<-- =LN(B10/B9)
11	1-Jun-87	16.62	4.93%	<-- =LN(B11/B10)
12	1-Jul-87	17.44	4.82%	<-- =LN(B12/B11)
13	3-Aug-87	18.11	3.77%	<-- =LN(B13/B12)
293	1-Dec-10	113.11	6.46%	
294	3-Jan-11	115.77	2.32%	
295	1-Feb-11	119.73	3.36%	
296	1-Mar-11	119.76	0.03%	
297	1-Apr-11	123.29	2.90%	
298	2-May-11	121.88	-1.15%	
299	1-Jun-11	119.84	-1.69%	
300	1-Jul-11	117.39	-2.07%	
301	1-Aug-11	110.99	-5.61%	
302	1-Sep-11	103.16	-7.32%	
303	3-Oct-11	114.42	10.36%	
304	1-Nov-11	114.15	-0.24%	
305	1-Dec-11	115.32	1.02%	
306	3-Jan-12	120.47	4.37%	
307	1-Feb-12	125.66	4.22%	
308	1-Mar-12	129.78	3.23%	
309	2-Apr-12	128.95	-0.64%	
310	1-May-12	121.19	-6.21%	
311	1-Jun-12	125.55	3.53%	

15. The Vanguard fund's prices incorporate dividends on the S&P 500. Many index figures (such as those available for ^GSPC on Yahoo) do not incorporate dividends.

Computing the Market Risk Premium $E(r_M) - r_f$ Directly

We can also compute the market risk premium directly. This requires a bit more work: In the spreadsheet below we show the monthly returns on the S&P 500 and monthly interest paid on U.S. Treasury bills. The average annualized risk premium on the S&P 500 is 4.40%.

	A	B	C	D	E	F
1	**MEASURING THE MARKET RISK PREMIUM $E(r_M) - r_f$ USING HISTORICAL DATA** **Vanguard 500 Index Fund (symbol: VFINX) minus Treasury Bills** **April 1987 - June 2012** **All measurements relate to monthly returns on SP500, r_{Mt}, and the Treasury bill rate r_{ft}**					
2	Average monthly risk premium	0.37%	<-- =AVERAGE(E10:E311)			Methodological note: I have used the St. Louis FRED data for 3-month Treasury Bills; this data is annualized, and I have divided it by 12 to get the monthly returns. Since the data can be taken as an ex-ante return, the April 1987 rate is attributed to May 1987. I've used 3-month instead of 1-month, because there are lots of data problems with the latter.
3	Monthly standard deviation	4.58%	<-- =STDEV(E10:E311)			
4						
5	Annualized risk premium	4.40%	<-- =12*B2			
6	Annualized standard deviation	15.85%	<-- =SQRT(12)*B3			
7						
8	**Date**	**Price**	**Return**	**Treasury bill rate**	**Market risk premium**	
9	1-Apr-87	15.66				
10	1-May-87	15.82	1.02%	0.48%	0.53%	<-- =C10-D10
11	1-Jun-87	16.62	4.93%	0.49%	4.45%	<-- =C11-D11
12	1-Jul-87	17.44	4.82%	0.49%	4.33%	
13	3-Aug-87	18.11	3.77%	0.49%	3.28%	
295	1-Feb-11	119.73	3.36%	0.01%	3.35%	
296	1-Mar-11	119.76	0.03%	0.01%	0.01%	
297	1-Apr-11	123.29	2.90%	0.01%	2.90%	
298	2-May-11	121.88	-1.15%	0.01%	-1.16%	
299	1-Jun-11	119.84	-1.69%	0.00%	-1.69%	
300	1-Jul-11	117.39	-2.07%	0.00%	-2.07%	
301	1-Aug-11	110.99	-5.61%	0.00%	-5.61%	
302	1-Sep-11	103.16	-7.32%	0.00%	-7.32%	
303	3-Oct-11	114.42	10.36%	0.00%	10.36%	
304	1-Nov-11	114.15	-0.24%	0.00%	-0.24%	
305	1-Dec-11	115.32	1.02%	0.00%	1.02%	
306	3-Jan-12	120.47	4.37%	0.00%	4.37%	
307	1-Feb-12	125.66	4.22%	0.00%	4.22%	
308	1-Mar-12	129.78	3.23%	0.01%	3.22%	
309	2-Apr-12	128.95	-0.64%	0.01%	-0.65%	
310	1-May-12	121.19	-6.21%	0.01%	-6.21%	
311	1-Jun-12	125.55	3.53%	0.01%	3.53%	

Applying the risk premium directly to the computation of Merck's cost of equity gives a cost of equity r_E close to 5% (note that we still haven't settled the question of r_f):

	A	B	C
1	**COMPUTING THE COST OF EQUITY FOR MERCK USING THE MARKET RISK PREMIUM $E(r_M) - r_f$**		
2	Merck beta, β	0.6435	<-- =Page96!B2
3	$E(r_M)$ derived from SP price/earnings	4.40%	<-- =Page100!B5
4	Merck tax rate, T_C	12.84%	<-- =Page98!B3
5	Risk free rate, r_f	2.00%	<-- Still to be discussed
6	Intel cost of equity, $r_{E,Intel}$		
7	Classic CAPM	4.83%	<-- =B5+B2*B3
8	Tax-adjusted CAPM	4.74%	<-- =B5*(1-B4)+B2*(B3+B4*B5)
9			
10	**Note**: The tax-adjusted model in cell B8 uses the equivalence: $E(r_M) - r_f(1-T_C) = E(r_M) - r_f + T_C{*}r_f$ For the low levels of taxes and low r_f in this example, there is virtually no difference between the two approaches.		

Calculating the Expected Return on the Market Using the Gordon Model

Setting $E(r_M) = 4.40\%$ approximates the historic market return in the United States for 1987–2012. Historic averages are appropriate if we think that the future anticipated rates of return will correspond to the historic average. On the other hand, we may want to take current market data to calculate directly the future anticipated market yield.

We can do this computation by using the Gordon model. Recall from section 3.6 that the model says that the cost of equity r_E is given by:

$$r_E = \frac{Div_0(1+g)}{P_0} + g$$

This formula also applies to the market portfolio, so that we can write: $r_M = \frac{Div_0(1+g)}{P_0} + g$, interpreting Div_0, P_0, and g to be the current dividend, price, and growth rate of the market portfolio. Assume that the firm pays out a constant proportion a of its earnings as dividends; then, indicating by EPS_0 the current earnings per share, $Div_0 = a{*}EPS_0$. Interpreting g to be the earnings growth of the firm, we can write:

$$E(r_M) = \frac{a * EPS_0(1+g)}{P_0} + g = \frac{a*(1+g)}{P_0 / EPS_0} + g$$

The term on the right-hand side of this equation, P_0/EPS_0, is the price earnings ratio of the market. We can use this formula to compute $E(r_M)$, and thus tie the cost of equity to currently observable market parameters. Here is an implementation:

	A	B	C
1	**COMPUTING E(r_M) USING MARKET MULTIPLE**		
2	Market price/earnings multiple, June 2012	15.20	
3	Equity cash flow payout ratio	50.00%	<-- Approx. U.S.: Dividends + repurchases
4	Anticipated growth of market equity cash flow	5.00%	<-- Analyst's estimate
5	Expected market return, E(r_M)	8.45%	<-- =B3*(1+B4)/B2+B4

In the remainder of this chapter, we will use this estimate for $E(r_M)$.

3.11 What's the Risk-Free Rate r_f in the CAPM?

Opinions about this seem to differ widely. Some authors suggest using a short-term rate, while others use a middle- or long-term rate. The author of this book has been guilty of making both suggestions, though at the current writing he believes that you should use the short-term rate.[16] For the examples in this chapter, here is some data from Yahoo/Finance on 29 June 2012:

US Treasury Bonds Rates

Maturity	Yield	Yesterday	Last Week	Last Month
3 Month	0.06	0.06	0.06	0.05
6 Month	0.14	0.14	0.13	0.12
2 Year	0.30	0.30	0.30	0.26
3 Year	0.39	0.39	0.41	0.35
5 Year	0.71	0.69	0.75	0.69
10 Year	1.64	1.58	1.68	1.62
30 Year	2.75	2.68	2.76	2.71

3.12 Computing the WACC, Three Cases

In the succeeding sections we compute the WACC for three companies: Merck, Whole Foods, and Caterpillar. In each of these examples we use $E(r_M) = 8.45\%$

16. Reason: Since the CAPM should apply to all risky assets, it should also apply to bonds (even default-free bonds like Treasuries, which are risky since they suffer significant price fluctuations). This suggests that the β of a risky asset impounds its risk, including the holding period risk, and that the proper risk-free rate r_f is a short-term rate. But in 1997 in my book *Principles of Corporate Finance: A Valuation Approach* (with Oded Sarig), we advocated using a mid- to long-term Treasury rate for r_f. As Emerson said: "A foolish consistency is the hobgoblin of little minds, adored by little statesmen and philosophers and divines" (Emerson forgot academics …).

(computed in section 3.10 from the S&P 500 price/earnings multiple) and the 3-month Treasury bill rate, $r_f = 0.06\%$ (as discussed in section 3.11).

These three examples, for Merck, Whole Foods, and Caterpillar, illustrate different situations and also how much ad hocery there is in the computations of the WACC.[17]

3.13 Computing the WACC for Merck (MRK)

We have discussed Merck's cost of equity r_E and cost of debt r_D in previous sections. Our template summarizes these computations:

	A	B	C
1	**COMPUTING THE WACC FOR MERCK**		
2	Shares outstanding	3.04	<-- Billions
3	Share price, 29 June 2012	41.75	
4	Equity value, E	126.92	<-- =B2*B3
5	Net debt, D	2.59	<-- Billions
6	Tax rate, T_C	12.84%	<-- =Page98!B3
7	Cost of debt, r_D	4.23%	<-- 0.0423
8	Expected market return, $E(r_M)$	8.45%	<-- =Page102!B5
9	Risk-free rate, r_f	2.00%	
10	Equity beta, β	0.6435	<-- ='Pages91,93'!B3
11			
12	**WACC based on Gordon per-share dividends**		
13	Current dividend/share	1.68	<-- =4*'Page84, bottom'!B43
14	Growth rate	2.02%	<-- ='Page84, bottom'!E9
15	Cost of equity, r_E	6.13%	<-- =B13*(1+B14)/B3+B14
16	WACC	6.08%	<-- =B15*$B$4/($B$4+$B$5)+B7*(1-$B$6)*$B$5/($B$4+$B$5)
17			
18	**WACC based on Gordon equity payouts**		
19	Current equity payout	6,418	<-- =Page86!E7
20	Growth rate	13.71%	<-- =Page86!B9
21	Cost of equity, r_E	11.20%	<-- ='Page88, bottom'!B20
22	WACC	13.90%	<-- =B21*$B$4/($B$4+$B$5)+B13*(1-$B$6)*$B$5/($B$4+$B$5)
23			
24	**WACC based on classic CAPM**		
25	Cost of equity, r_E	6.15%	<-- =B9+B10*(B8-B9)
26	WACC	6.03%	<-- =B25*$B$4/($B$4+$B$5)+B17*(1-$B$6)*$B$5/($B$4+$B$5)
27			
28	**WACC based on tax-adjusted CAPM**		
29	Cost of equity, r_E	6.06%	<-- =B9*(1-B6)+B10*(B8-B9*(1-B6))
30	WACC	6.14%	<-- =B29*$B$4/($B$4+$B$5)+B21*(1-$B$6)*$B$5/($B$4+$B$5)
31			
32	**Estimated WACC?**	6.08%	<-- =AVERAGE(B16,B26,B30)

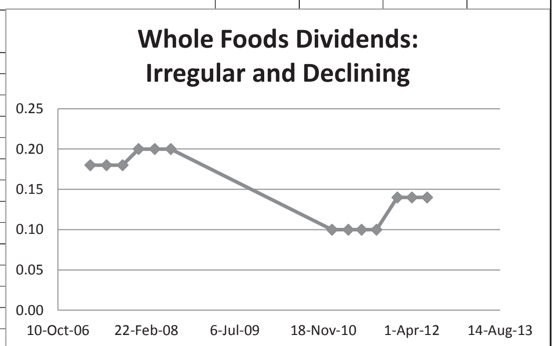
17. Wiktionary (http://en.wiktionary.org) defines "ad hocery" as "improvised reasoning." When used in combination with financial theory, we prefer to think of it as the golden mean between reality and theory.

The estimated WACC for Merck (cell B32) includes a judgment call: We have averaged three of the estimates that are closest together, throwing out the one estimate that appears to us to be extreme.

3.14 Computing the WACC for Whole Foods (WFM)

In a previous section we pointed out that Whole Foods (WFM) was an example of negative net debt; the company has liquid assets in excess of its debt. The company's dividends have declined over the past 5 years and exhibit a break between July 2008 and January 2011:

	A	B	C	D	E	F	G
1	**COMPUTING WHOLE FOODS (WFM) r_E WITH THE GORDON MODEL**						
2	WFM stock price P_0, 29 June 2012	95.32					
3	Current dividend						
4	Quarterly	0.14					
5	Annualized dividend, Div_0	0.56	<-- =4*B4				
6	Dividend growth rate, g since April 2007	-4.70%	<-- =(B26/B14)^(1/5.22)-1				
7	Dividend growth rate, g since January 2011	14.41%					
8	Gordon model cost of equity, r_E						
9	Based on dividends since April 2007	-4.14%	<-- =$B$5*(1+B6)/$B$2+B6				
10	Based on dividends since January 2011	15.08%	<-- =$B$5*(1+B7)/$B$2+B7				
11							
12	**Whole Foods Dividend History**						
13	Date	Dividend per share					
14	11-Apr-07	0.18					
15	11-Jul-07	0.18					
16	10-Oct-07	0.18					
17	9-Jan-08	0.20					
18	9-Apr-08	0.20					
19	9-Jul-08	0.20					
20	6-Jan-11	0.10					
21	8-Apr-11	0.10					
22	22-Jun-11	0.10					
23	15-Sep-11	0.10					
24	11-Jan-12	0.14					
25	3-Apr-12	0.14					
26	27-Jun-12	0.14					
27							
28	Term of dividends	5.22	<-- =(A26-A14)/365				



In the WACC template at the end of this subsection we use the average of the two dividend growth rates in B9 and B10.

Whole Foods total equity payouts show that in the past 3 years the company has absorbed equity from the capital markets. We do not think there is any growth rate for the total equity payouts that can be used in the WACC computation. In our WACC template for WFM, we ignore this method of computing the cost of equity.

	A	B	C	D	E	F
1	**GORDON MODEL FOR WHOLE FOOD'S EQUITY PAYOUTS**					
2		**Dividends**	**Common stock issuance**	**Proceeds from stock option exercise**	**Total equity payout**	
3	2007	96,742	54,383		42,359	<-- =B3-C3
4	2008	109,072	18,019		91,053	
5	2009	0	4,286		-4,286	
6	2010	0	46,962		-46,962	
7	2011	52,620	296,719		-244,099	

	A	B	C
1	**COMPUTING THE WACC FOR WHOLE FOODS MARKET (WFM)**		
2	Shares outstanding	183.56	<-- Millions
3	Share price, 29 June 2012	95.32	
4	Equity value, E	17,497	<-- =B2*B3, Millions
5	Net debt, D	-728	<-- Millions
6	Tax rate, T_C	37.90%	<-- ='Page75, bottom'!D5
7	Cost of debt, r_D	4.72%	<-- From financial statements, interest on term loan
8	Expected market return, $E(r_M)$	8.45%	<-- =Page102!B5
9	Risk-free rate, r_f	0.06%	
10	Equity beta, β	0.51	<-- From Yahoo
11			
12	**WACC based on Gordon per-share dividends**		
13	Current dividend/share	0.56	<-- =Page104!B5
14	Growth rate	5.47%	<-- =AVERAGE(Page104!B9:B10)
15	Cost of equity, r_E	6.09%	<-- =B13*(1+B14)/B3+B14
16	WACC	6.23%	<-- =B15*$B$4/($B$4+$B$5)+B7*(1-$B$6)*$B$5/($B$4+$B$5)
17			
18	**WACC based on Gordon equity payouts**		
19	Current equity payout		
20	Growth rate	Not applicable	
21	Cost of equity, r_E		
22	WACC		
23			
24	**WACC based on classic CAPM**		
25	Cost of equity, r_E	4.34%	<-- =B9+B10*(B8-B9)
26	WACC	4.53%	<-- =B25*$B$4/($B$4+$B$5)+B17*(1-$B$6)*$B$5/($B$4+$B$5)
27			
28	**WACC based on tax-adjusted CAPM**		
29	Cost of equity, r_E	4.33%	<-- =B9*(1-B6)+B10*(B8-B9*(1-B6))
30	WACC	4.52%	<-- =B29*$B$4/($B$4+$B$5)+B21*(1-$B$6)*$B$5/($B$4+$B$5)
31			
32	**Estimated WACC?**	5.09%	<-- =AVERAGE(B16,B26,B30)

3.15 Computing the WACC for Caterpillar (CAT)

Caterpillar's cost of debt is very low:

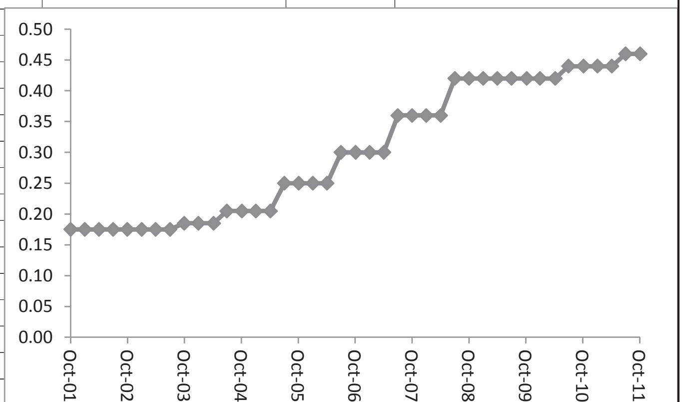
	A	B	C	D	E
1	**CATERPILLAR DEBT and Cost of debt r_D Numbers in thousands**				
2		**31-Dec-09**	**31-Dec-10**	**31-Dec-11**	
3	Cash	4,867,000	3,592,000	3,057,000	
4					
5	Short-term debt	9,648,000	7,981,000	9,784,000	
6	Long-term debt	24,944,000	20,437,000	21,847,000	
7					
8	Net debt	29,725,000	24,826,000	28,574,000	
9					
10	Interest expense	389,000	343,000	396,000	
11	Cost of debt, r_D?		1.26%	1.48%	<-- =D10/AVERAGE(C8:D8)

Caterpillar's tax rate is comfortably around 25% in years where it has significant income:

	A	B	C	D	E
1	**CATERPILLAR TAX RATE**				
2		31-Dec-09	31-Dec-10	31-Dec-11	
3	Income before tax	569,000	3,750,000	6,725,000	
4	Income tax expense	-270,000	968,000	1,720,000	
5	Implied tax rate	-47.45%	25.81%	25.58%	<-- =D4/D3

Below we show 10 years of dividend history for CAT. In our model we will take the last 5 years of annual dividend growth to compute CAT's r_E using the Gordon dividend model:

	A	B	C	D	E	F
1	**CATERPILLAR DIVIDEND GROWTH AND GORDON DIVIDEND r_E**					
2	Date	Dividend per share		Dividend growth		
3	18-Oct-01	0.18		Whole period		
4	17-Jan-02	0.18		Quarterly growth	2.45%	<-- =(B43/B3)^(1/40)-1
5	18-Apr-02	0.18		Annual growth	10.15%	<-- =(1+E4)^4-1
6	18-Jul-02	0.18				
7	17-Oct-02	0.18		Last 5 years		
8	16-Jan-03	0.18		Quarterly growth	2.16%	<-- =(B43/B23)^(1/20)-1
9	16-Apr-03	0.18		Annual growth	8.92%	<-- =(1+E8)^4-1
10	17-Jul-03	0.18				
11	16-Oct-03	0.19		Cost of equity		
12	15-Jan-04	0.19		Current share price, P_0	90.60	
13	22-Apr-04	0.19		Current dividend, Div_0	0.70	
14	16-Jul-04	0.21		Dividend growth, g	8.92%	<-- =E9
15	21-Oct-04	0.21		Gordon cost of equity, r_E	9.77%	<-- =E13*(1+E14)/E12+E14
16	18-Jan-05	0.21				
17	21-Apr-05	0.21				
18	20-Jul-05	0.25				
19	20-Oct-05	0.25				
20	18-Jan-06	0.25				
21	20-Apr-06	0.25				
22	18-Jul-06	0.30				
23	19-Oct-06	0.30				
24	18-Jan-07	0.30				
25	19-Apr-07	0.30				
26	18-Jul-07	0.36				
27	18-Oct-07	0.36				
28	17-Jan-08	0.36				
29	17-Apr-08	0.36				
30	17-Jul-08	0.42				
31	16-Oct-08	0.42				
32	15-Jan-09	0.42				
33	16-Apr-09	0.42				



Caterpillar's equity payouts are very variable. We use the growth over the last 2 years as the basis for the payout method of computing r_E:

	A	B	C	D	E	F
1	**CATERPILLAR'S EQUITY PAYOUTS**					
2		**Dividends**	**Shares issued**	**Treasury shares purchased**	**Total equity payout**	
3	31-Dec-07	845	-328	2,405	2,922	<-- =B3+C3+D3
4	31-Dec-08	953	-135	1,800	2,618	
5	31-Dec-09	1,029	-89		940	
6	31-Dec-10	1,084	-296		788	
7	31-Dec-11	1,159	-123		1,036	
8						
9	Growth, 5 years	-22.84%	<-- =(E7/E3)^(1/4)-1			
10	Growth, last 3 years	4.98%	<-- =(E7/E5)^(1/2)-1			

Using these values, we get the following template for CAT's WACC:

	A	B	C
1	**COMPUTING THE WACC FOR CATERPILLAR (CAT)**		
2	Shares outstanding	624.72	<-- million
3	Share price, 29 June 2012	90.60	
4	Equity value, E	56.60	<-- =B2*B3/1000
5	Net debt, D	28.57	<-- Billions
6	Tax rate, T_C	25.58%	<-- ='Page106, bottom'!D5
7	Cost of debt, r_D	1.48%	<-- ='Page106, top'!D11
8	Expected market return, $E(r_M)$	8.45%	<-- =Page102!B5
9	Risk-free rate, r_f	0.06%	
10	Equity beta, β	1.98	<-- Yahoo
11			
12	**WACC based on Gordon per-share dividends**		
13	Current dividend/share	0.70	<-- =4*Page107!B3
14	Growth rate	9.77%	<-- =Page107!E15
15	Cost of equity, r_E	10.61%	<-- =B13*(1+B14)/B3+B14
16	WACC	7.42%	<-- =B15*$B$4/($B$4+$B$5)+B7*(1-$B$6)*$B$5/($B$4+$B$5)
17			
18	**WACC based on Gordon equity payouts**		
19	Current equity payout	1.036	<-- ='Page108, top'!E7/1000
20	Growth rate	4.98%	<-- ='Page108, top'!B10
21	Cost of equity, r_E	6.90%	<-- =B19*(1+B20)/B4+B20
22	WACC	4.96%	<-- =B21*$B$4/($B$4+$B$5)+B7*(1-$B$6)*$B$5/($B$4+$B$5)
23			
24	**WACC based on classic CAPM**		
25	Cost of equity, r_E	16.68%	<-- =B9+B10*(B8-B9)
26	WACC	11.08%	<-- =B25*$B$4/($B$4+$B$5)+B17*(1-$B$6)*$B$5/($B$4+$B$5)
27			
28	**WACC based on tax-adjusted CAPM**		
29	Cost of equity, r_E	16.70%	<-- =B9*(1-B6)+B10*(B8-B9*(1-B6))
30	WACC	12.82%	<-- =B29*$B$4/($B$4+$B$5)+B21*(1-$B$6)*$B$5/($B$4+$B$5)
31			
32	**Estimated WACC?**	11.95%	<-- =AVERAGE(B26,B30)

Our WACC estimates divide into two groups: the two Gordon-based models produce significantly lower estimates for the r_E and WACC than the two CAPM-based methods. In this case we take the average of the two latter computations (our general preference is often for CAPM over the dividend models).

3.16 When Don't the Models Work?

All models have problems and nothing is perfect.[18] In this section we discuss some of the potential problems with the Gordon model and with the capital asset pricing model.

Problems with the Gordon Model

Obviously the Gordon model doesn't work if a firm doesn't pay dividends and appears to have no intention—in the immediate future—of paying dividends.[19] But even for dividend-paying firms, it may be difficult to apply the model. Particularly problematic, in many cases, is the extraction of the future dividend payout rate from past dividends.

Consider, for example, the dividend history of Ford Motor Company in the years 1989–1998:

	A	B	C
1	**FORD MOTOR CO. DIVIDEND HISTORY 1989-1998**		
2	**Year**	**Dividend**	
3	1989	3.00	
4	1990	3.00	
5	1991	1.95	
6	1992	1.60	
7	1993	1.60	
8	1994	1.33	
9	1995	1.23	
10	1996	1.46	
11	1997	1.64	
12	1998	22.81	
13	Growth rate, 1989-1997	-7.27%	<-- =(B11/B3)^(1/8)-1
14	Growth rate, 1989-1998	25.28%	<-- =(B12/B3)^(1/9)-1

18. "Happiness is the maximum agreement of reality and desire."—Stalin.

19. Firms cannot intend *never* to pay dividends, because such an intention would rationally mean that the value of the shares is zero.

The problem here is easily identifiable: Ford, whose dividends were in steady decline until 1997, paid a cash dividend of $21.09 in 1998, this in addition to its regular quarterly dividends (which summed to $1.72 in 1998). If we use past history to predict the future, any inclusion of the extraordinary cash dividend will cause us to overestimate the future dividend growth. Excluding the $21.09 dividend, however, also does not reflect the actual situation.

It appears that the 10-year history of Ford's dividends is not, perhaps, the best guide to its future dividend payout. There are several solutions to those wishing to use the Gordon model:

• If we exclude the extraordinary dividend of $21.09 in 1998, then the dividend growth over the 4 years ending in 1998 is a respectable 6.64%. If Ford's anticipated future dividend growth is estimated to be this rate, then—given its end-1998 stock price of $58.69—the Gordon model cost of equity is 9.77%:

	A	B	C
17	**FORD'S DIVIDENDS EXCLUDING THE 1998 $21.09 DIVIDEND**		
18	**Year**	**Dividend**	
19	1989	3.00	
20	1990	3.00	
21	1991	1.95	
22	1992	1.60	
23	1993	1.60	
24	1994	1.33	
25	1995	1.23	
26	1996	1.46	
27	1997	1.64	
28	1998	1.72	
29	Growth rate, 1994-1998	6.64%	<-- =(B28/B24)^(1/4)-1
30	Ford's stock price, end-1998	58.69	
31	Gordon cost of equity	9.77%	<-- =B28*(1+B29)/B30+B29

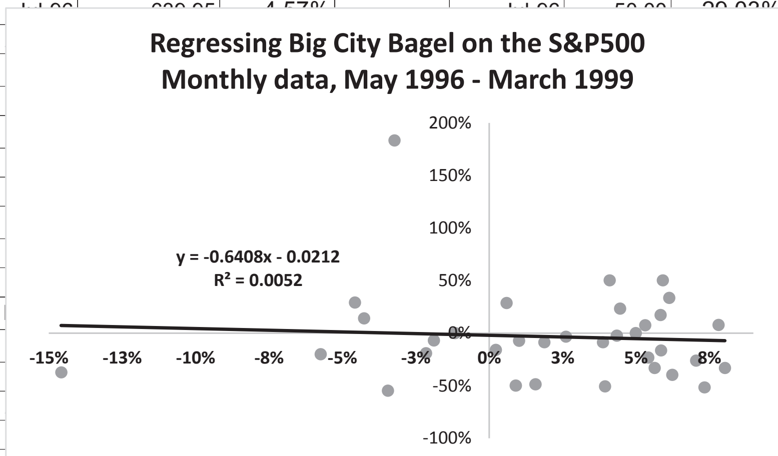
• A better alternative might be to use Ford's total payouts to equity, as illustrated in this chapter. This method does not mean, however, that we can get away from judgment calls (witness our extensive use of the two-stage Gordon model).

• A last alternative to finding Ford's cost of capital is to predict its future dividends by doing a full-blown financial model for the company. Such models—illustrated in the succeeding two chapters—are often used by analysts. Though they are complicated and time consuming to build, they take into account all of the firm's productive and financial activities. Potentially they are, therefore, a more accurate predictor of the dividend.

Problems with the CAPM

In the spreadsheet fragment below you will find the return of the S&P 500 and Big City Bagels. From the regression, you can see that Big City's β is –0.6408.

	A	B	C	D	E	F	G
1	**COMPUTING BIG CITY BAGEL'S BETA**						
2	**S&P 500 Index**				**Big City Bagels (BIGC)**		
3	**Date**	**Closing price**			**Date**	**Closing price**	
4	May-96	669.12			May-96	46.25	
5	Jun-96	670.63	0.23%		Jun-96	38.75	-16.22%
6	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21	Oct-97	914.62	-3.45%		Oct-97	10.16	-54.86%



Big City Bagel's stock is clearly risky—the annualized standard deviation of its returns is 135% as compared to about 17% for the S&P 500 over the same period. However, the β of Big City Bagels is –0.0542, which indicates that Big City has—in a portfolio context—*negative risk*. Were this true, it would mean that adding Big City to a portfolio would lower the portfolio variance enough to justify a below-risk-free return for Big City. While this might be true for some stocks, it is hard to believe that—in the long run—the β of Big City is indeed negative.[20]

20. A more plausible explanation is that—for the period covered—Big City's return has *nothing whatsoever* to do with the market return.

The R^2 of the regression between Big City's returns and the S&P 500 is essentially zero, meaning that the S&P 500 simply doesn't explain any of the variation in Big City returns. For statistics mavens: The *t*-statistics of the intercept and the slope indicate that neither differs significantly from zero. In short: The regression of Big City Bagels' historic returns on the S&P 500 indicates no connection between the two whatsoever.

What are we to make of this situation? How should we calculate the cost of capital for Big City? There are several alternatives:

- We could assume that the Big City β is –0.0542. The company's tax rate in March 1999 was essentially zero, so that the classical CAPM and the tax-adjusted version coincide:

	A	B	C
1	**COMPUTING THE COST OF EQUITY r_E FOR BIG CITY BAGELS** **March 1999**		
2	Big City's beta	-0.0542	
3			
4	Risk-free rate, r_f	4.29%	
5	Expected market return, $E(r_M)$	9.08%	
6	Cost of equity, r_E	4.03%	<-- =B4+B2*(B5-B4)

- We could assume that the β of Big City is in fact 0; given the standard deviation of the β estimate for Big City, the β is not statistically different from zero, so that this assumption makes sense. This means that all of Big City's risk is diversifiable and that the correct cost of equity for Big City is the riskless rate of interest.
- We could assume that the covariance (or lack thereof) between Big City and the S&P 500 is not indicative of their future correlation. This would eventually lead us to conclude that Big City's risk is comparable to that of similar companies. A small study of the βs of snack food companies during the same period shows their βs to be well over 1: New World Coffee has a β of 1.15, Pepsico has a β of 1.42, Starbucks has a β of 1.84. Thus we might conclude that the β of Big City (in the sense of its *future* correlation with the market) would be somewhere between 1.15 and 1.84. This would, of course, give a radically different cost of equity for Big City:

	A	B	C
1	**COMPUTING THE COST OF EQUITY r_E FOR BIG CITY BAGELS** **Assumes that forward-looking beta = 1.3**		
2	Big City's beta	1.3000	
3			
4	Risk-free rate, r_f	4.29%	
5	Expected market return, $E(r_M)$	9.08%	
6	Cost of equity, r_E	10.52%	<-- =B4+B2*(B5-B4)

(For what it's worth: This author would follow the latter case …)

3.17 Summary

In this chapter we have illustrated in detail the application of two models for calculating the cost of equity: The Gordon dividend model and the CAPM. We have also considered three of the four practicable models for calculating the cost of debt. Because the application of these models includes for many judgment calls, our advice is to do the following:

- Always use several models to calculate the cost of capital.
- If you have time, try to calculate the cost of capital not only for the firm you are analyzing but also for other firms in the same industry.
- From your analysis try to pick out a *consensus* estimate of the cost of capital. Don't hesitate to exclude numbers (such as Big City's negative cost of equity) that strike you as unreasonable.

In sum, the calculation of the cost of capital is not just a mechanistic exercise!

Exercises

1. ABC Corp. has a stock price $P_0 = 50$. The firm has just paid a dividend of \$3 per share, and intelligent shareholders think that this dividend will grow by a rate of 5% per year. Use the Gordon dividend model to calculate the cost of equity of ABC.

2. Unheardof, Inc., has just paid a dividend of \$5 per share. This dividend is anticipated to increase at a rate of 15% per year. If the cost of equity for Unheardof is 25%, what should be the market value of a share of the company?

3. Dismal.Com is a producer of depressing Internet products. The company is currently not paying dividends, but its chief financial officer thinks that starting in 3 years it can pay a dividend of $15 per share, and that this dividend will grow by 20% per year. Assuming that the cost of equity of Dismal.Com is 35%, value a share based on the discounted dividends.

4. Consider the following dividend and price data for Chrysler:

	A	B	C	D	E
1	**CHRYSLER CORPORATION (C)**				
2	**Year**	**Year-end stock price**	**Dividend per share**	**Growth rate**	
3	1986		0.40		
4	1987		0.50	25.00%	<-- =C4/C3-1
5	1988		0.50	0.00%	<-- =C5/C4-1
6	1989		0.60	20.00%	<-- =C6/C5-1
7	1990		0.60	0.00%	<-- =C7/C6-1
8	1991		0.30	-50.00%	<-- =C8/C7-1
9	1992		0.30	0.00%	
10	1993		0.33	10.00%	
11	1994		0.45	36.36%	
12	1995		1.00	122.22%	
13	1996	35.00	1.40	40.00%	

Use the Gordon model to calculate Chrysler's cost of equity at end-1996 on the basis of dividends only.

5. The current stock price of TransContinentalAirways is $65 per share. TCA currently pays an annual per-share dividend of $3. Over the past 5 years this dividend has grown annually at a rate of 23%. A respected analyst assumes that the current growth rate of dividends will hold up for the next 5 years, after which dividend growth will slow to 5% annually. Use the **twostagegordon** function to compute the cost of equity.[21]

6. ABC Corp. has just paid a dividend of $3 per share. You—an experienced analyst—feel quite sure that the growth rate of the company's dividends over the next 10 years will be 15% per year. After 10 years you think that the company's dividend growth rate will slow to the industry average, which is about 5% per year. If the cost of equity for ABC is 12%, what is the value today of one share of the company?

7. Consider a company which has β_{equity} = 1.5 and β_{debt} = 0.4. Suppose that the risk-free rate of interest is 6%, the expected return on the market $E(r_M)$= 15%, and that the corporate tax rate is 40%. If the company has 40% equity and 60% debt in its capital structure, calculate its weighted average cost of capital using both the classic CAPM and the tax-adjusted CAPM.

---

21. To do this problem you will have to copy the formula from the chapter spreadsheet to your answer spreadsheet. See Chapter 0 for details.

8. On the spreadsheet with this chapter you will find the following monthly data for Cisco's stock price and the S&P 500 index. Compute the equation $r_{CSCO,t} = \alpha_{CSCO} + \beta_{CSCO} r_{SP,t}$ and include the R^2 and t-statistics for the equation and its coefficients.[22]

	A	B	C
1	**CISCO (CSCO) AND S&P 500 PRICES July 2002 - June 2007**		
2	**Date**	**S&P 500**	**CSCO**
3	3-Jul-02	911.62	13.19
4	1-Aug-02	916.07	13.82
5	3-Sep-02	815.28	10.48
6	1-Oct-02	885.76	11.18
7	1-Nov-02	936.31	14.92
8	2-Dec-02	879.82	13.10
9	2-Jan-03	855.70	13.37
10	3-Feb-03	841.15	13.98
11	3-Mar-03	848.18	12.98
12	1-Apr-03	916.92	15.00
13	1-May-03	963.59	16.41
14	2-Jun-03	974.50	16.79
15	1-Jul-03	990.31	19.49
16	1-Aug-03	1008.01	19.14
17	2-Sep-03	995.97	19.59
18	1-Oct-03	1050.71	20.93
19	3-Nov-03	1058.20	22.70
20	1-Dec-03	1111.92	24.23
21	2-Jan-04	1131.13	25.71
22	2-Feb-04	1144.94	23.16

9. You are considering buying the bonds of a very risky company. A bond with a \$100 face value, a 1-year maturity, and a coupon rate of 22% is selling for \$95. You consider the probability that the company will actually survive to pay off the bond 80%. With 20% probability, you think that the company will default, in which case you think that you will be able to recover \$40.

a. What is the expected return on the bond?

b. If the company has cost of equity $r_E = 25\%$, tax rate $T_C = 35\%$, and 40% of its capital structure is equity, what is its weighted average cost of capital (WACC)?

10. It is 1 January 1997. Normal America, Inc. (NA) has paid a year-end dividend in each of the last 10 years, as shown by the table below:

22. To do this problem you will have to copy the functions **tintercept** and **tslope** from the chapter file to your answer spreadsheet. See previous footnote.

	A	B	C	D	E	F
1	**NORMAL AMERICA, INC.**					
2	**Year**	**Dec. 31 stock price**	**Dec. 15 dividend per share**			**S&P 500 return**
3	1986	33.00				
4	1987	30.69	2.50		1987	4.7%
5	1988	35.38	2.50		1988	16.2%
6	1989	42.25	3.00		1989	31.4%
7	1990	34.38	3.00		1990	-3.3%
8	1991	36.25	1.60		1991	30.2%
9	1992	32.25	1.40		1992	7.4%
10	1993	43.00	0.80		1993	9.9%
11	1994	42.13	0.80		1994	1.2%
12	1995	52.88	1.10		1995	37.4%
13	1996	55.75	1.60		1996	22.9%

a. Calculate NA's β with respect to the S&P 500.

b. Suppose that the Treasury bill rate is 5.5% and that the expected return on the market is $E(r_M) = 13\%$. If the corporate tax rate $T_C = 35\%$, calculate NA's cost of equity using both the classic CAPM and tax-adjusted model.

c. Assume that NA's cost of debt is 8%. If the company is financed by 1/3 equity and 2/3 debt, what is its weighted average cost of capital using each of the two CAPM models?

11. At the end of June 2007, the price/earnings ratio of the S&P 500 was 17.5. Assume that the index proxies for the market, that it has a 50% dividend payout ratio, and that dividends are expected to grow at 7%. Compute $E(r_M)$.

12. The template for exercise 12 gives the prices of the Vanguard Index 500 Fund (symbol: VFINX). This fund's prices replicate the S&P 500 with dividends reinvested. Use these data to estimate the expected return on the S&P 500 in two variations: All the data, the last 2 years. (This exercise shows the problematics of using historical market data to estimate the expected returns.)

13. Suppose that the S&P 500 price/earnings ratio is 17.5, the dividend payout ratio of the S&P is 50%, and that you estimate a future growth of dividends of 7%. What is $E(r_M)$?

	A	B	C
1	**COMPUTING E(r_M) FROM MARKET P/E**		
2	Current S&P 500 P/E	17.5	
3	Dividend payout ratio	50%	
4	Growth rate of dividends	7%	
5	E(r_M)		

14. The template for exercise 14 gives the 10-year history of Intel's quarterly dividends. Compute Intel's cost of equity r_E using the Gordon dividend model. Compare the cost of equity computed on the basis of 10 years of growth with that computed on the last 5 years of growth.

4 Valuation Based on the Consolidated Statement of Cash Flows

4.1 Overview

Chapter 2 of *Financial Modeling* defines four approaches to corporate valuation. All are based on computing the firm's enterprise value (EV), defined as the present value of the firm's future free cash flows (FCFs).

- The accounting approach to EV moves items on the balance sheet so that all operating items are on the left-hand side of the balance sheet and all financial items are on the right-hand side.
- The efficient markets approach to EV revalues—to the extent possible—items on the accounting EV balance sheet at market values. An obvious revaluation is to replace the firm's book value of equity with the market value of the equity.
- The discounted cash flow (DCF) approach values the EV as the present value of the firm's future anticipated free cash flows (FCFs) discounted at the weighted average cost of capital (WACC). The FCFs are the cash flows produced by the firm's productive assets—its working capital, fixed assets, goodwill, etc. In this book we use two implementations of the DCF approach. These approaches differ in their derivation of the firm's free cash flows:

  In this chapter we base our projections of future anticipated FCFs on an analysis of the firm's consolidated statement of cash flows. This method is easy to implement and (for a valuation method, all of which take a lot of time) relatively simple.

  In Chapters 5 and 6 we base our projections of future anticipated FCFs on a pro forma model for the firm's financial statements. Pro forma statements are powerful tools that can be used for business plans as well as valuations, but they are difficult and time consuming to implement.

Although both Chapter 4 (this chapter) and Chapter 5 differ in their method for deriving the free cash flows to be discounted, both chapters boil down to the following template:

	A	B	C	D	E	F	G	H
1	**BASIC CASH FLOW VALUATION TEMPLATE**							
2	Current free cash flow (FCF)	1,000						
3	Growth rate of FCF, years 1-5	8.00%						
4	Long-term FCF growth rate	5.00%						
5	Weighted average cost of capital (WACC)	11.00%						
6								
7	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
8	Future FCFs		1,080	1,166	1,260	1,360	1,469	<-- =F8*(1+$B$3)
9	Terminal value						25,713	<-- =G8*(1+B4)/(B5-B4)
10	Total		1,080	1,166	1,260	1,360	27,183	<-- =G8+G9
11								
12	Enterprise value	20,933	<-- =NPV(B5,C10:G10)*(1+B5)^0.5					
13	Add back initial cash	2,000	<-- From current balance sheet					
14	Subtract out debt	10,000	<-- From current balance sheet					
15	Equity value	12,933	<-- =B12+B13-B14					
16	Per share (1,000 shares)	12.93	<-- =B15/1000					

The difference between the two DCF approaches is in the derivation of the future FCFs. In this chapter we examine the firm's consolidated statement of cash flows (CSCFs) and use it as a basis for estimating the future FCFs. We then discuss issues related to estimating the short-term growth rate (8% above), the long-term growth rate (5%), assuming that you have learned from Chapter 3 how to compute the weighted average cost of capital (WACC) (11% above).

We focus on a number of important technical issues:

- Adjustments that need to be made in the passage from the consolidated statement of cash flows (CSCFs) to the free cash flow (FCF). These adjustments involve:
  - Financing adjustments
  - Corrections for the vagaries of accounting rules
  - Eliminating non-forward-looking items
- Dates that don't match. Quite often the dates are not evenly spaced. We may be, for example, projecting from annual statements that end on 31 December, but the current valuation date may be September. How do we make our valuation appropriate to this? The answer is to use **XNPV**, as we shall see.
- Estimating the return on assets versus the return on equity. **XIRR** can provide us with the answer.

Finally, we discuss the methodology for making reality fit our template (or is it vice versa? Sometimes it's hard to tell!).

4.2 Free Cash Flow (FCF): Measuring the Cash Produced by the Business

The *free cash flow* (FCF) is defined as the cash produced by a business without taking into account the way the business is financed—is the best measure of the cash produced by a business. We discussed the definition of the FCF in Chapter 2 and here only recall the definition:

<table>
<tr><th colspan="3">Defining the Free Cash Flow</th></tr>
<tr><td>Profit after taxes</td><td colspan="2">Accounting measure of firm profitability. It is not a cash flow.</td></tr>
<tr><td>+ Depreciation</td><td colspan="2">This noncash expense is added back to the profit after tax.</td></tr>
<tr><td>– Increase in operating current assets</td><td>Increase in sales-related current assets is not an expense for tax purposes (and is therefore ignored in the profit after taxes), but it is a cash drain on the company.</td><td rowspan="2">For purposes of FCF, our definitions of current assets and current liabilities excludes financing items such as cash and debt.</td></tr>
<tr><td>+ Increase in operating current liabilities</td><td>Increase in sale-related current liabilities provides cash to the firm.</td></tr>
<tr><td>– Increase in fixed assets at cost</td><td colspan="2">An increase in fixed assets (the long-term productive assets of the company) is a use of cash, which reduces the firm’s free cash flow.</td></tr>
<tr><td>+ After-tax net interest payments</td><td colspan="2">FCF measures the cash produced by the business activity of the firm. Add back after-tax net interest payments to neutralize the interest component of the profit after taxes.</td></tr>
</table>

In this chapter we base our computations of the free cash flow (FCF) on the firm’s consolidated statement of cash flows (CSCFs).

Computing the Terminal Value

The enterprise value (EV) is defined as the present value of all future FCFs, $EV = \sum_{t=1}^{\infty} \frac{FCF_t}{(1+WACC)^t}$. Our valuation model assumes that a FCF growth rate for the short term (years 1–5) and another growth rate for the long term (year

6 and subsequent). Denoting the short-term growth rate by *STg* and the long-term growth rate by *LTg*, the enterprise value of the firm can be written as:

$$EV = \underbrace{\sum_{t=1}^{5} \frac{FCF_0(1+STg)^t}{(1+WACC)^t}}_{\substack{\text{The present value}\\ \text{of cash flows, t=1,...,5,}\\ \text{growing at the}\\ \text{short-term growth rate}}} + \underbrace{\frac{1}{(1+WACC)^5}}_{\substack{\text{Discounting the}\\ \text{terminal value to}\\ \text{time 0}}} \underbrace{\sum_{t=1}^{\infty} \frac{FCF_5(1+LTg)^t}{(1+WACC)^t}}_{\substack{\text{Terminal value:}\\ \text{The present value in year 5}\\ \text{of cash flows, t>5,}\\ \text{growing at long-term}\\ \text{growth rate}}}$$

Using a standard technique, we can show that:

$$\begin{aligned} \textit{Terminal value} &= \sum_{t=1}^{\infty} \frac{FCF_5 * (1+LTg)^t}{(1+WACC)^t} \\ &= \begin{cases} \dfrac{FCF_5 * (1+LTg)}{WACC - LTg} & \text{if } WACC > LTg \\ \textit{undefined} & \textit{otherwise} \end{cases} \end{aligned}$$

Mid-Year Discounting

Rewriting the enterprise value (EV) to include the terminal value gives:

$$EV = \sum_{t=1}^{5} \frac{FCF_t}{(1+WACC)^t} + \frac{1}{(1+WACC)^5} \frac{FCF_5(1+LTg)}{(WACC - LTg)}$$

This formulation assumes that all cash flows occur at year-end. In fact most corporate cash flows occur throughout the year; if we approximate this fact by assuming that on average the year-*t* cash flow occurs in the middle of the year, we can rewrite the EV equation as

$$EV = \sum_{t=1}^{5} \frac{FCF_t}{(1+WACC)^{t-0.5}} + \frac{1}{(1+WACC)^{4.5}} \frac{FCF_5(1+LTg)}{(WACC - LTg)}$$

A little bit of algebra shows how this can be accommodated by Excel's **NPV** function:

$$EV = \underbrace{\left[\sum_{t=1}^{5} \frac{FCF_t}{(1+WACC)^t} + \frac{1}{(1+WACC)^5} \frac{FCF_5(1+LTg)}{(WACC - LTg)} \right]}_{\text{Can be computed by Excel's } \textbf{NPV} \text{ function}} * (1+WACC)^{0.5}$$

It is this version of EV that is used throughout this section of *Financial Modeling*.

4.3 A Simple Example

The consolidated statement of cash flows for ABC Corp. for the past 5 years is given below:

	A	B	C	D	E	F	G
1	**ABC CORPORATION Consolidated Statement of Cash Flows, 2008-2012**						
2		**2008**	**2009**	**2010**	**2011**	**2012**	
3	**Operating Activities:**						
4	Net earnings	479,355	495,597	534,268	505,856	520,273	
5	Adjustments to reconcile net earnings to net cash provided by operating activities						
6	Add back depreciation and amortization	41,583	47,647	46,438	45,839	46,622	
7	Changes in operating assets and liabilities:						
8	Subtract increase in accounts receivable	9,387	25,951	-12,724	1,685	-2,153	
9	Subtract increase in inventories	-37,630	-22,780	-16,247	-15,780	-5,517	
10	Subtract increase in prepaid expenses and other assets	-52,191	13,573	16,255	14,703	-2,975	
11	Add increase in accounts payable, accrued expenses, pensions, and other liabilities	29,612	51,172	6,757	40,541	60,255	
12	**Net cash provided by operating activities**	**470,116**	**611,160**	**574,747**	**592,844**	**616,505**	<-- =SUM(F4:F11)
13							
14	**Investing Activities:**						
15	Short-term investments, net	-5,000	-55,000	50,000	-10,000	20,000	
16	Purchases of property, plant and equipment	-48,944	-70,326	-89,947	-37,044	-88,426	
17	Proceeds from dispositions of property, plant and equipment	197	6,956	22,942	6,179	28,693	
18	**Net cash used in investing activities**	**-53,747**	**-118,370**	**-17,005**	**-40,865**	**-39,733**	<-- =SUM(F15:F17)
19							
20	**Financing Activities:**						
21	Repayment of debt	0	0	-300,000	0	-7,095	
22	Proceeds from revolving credit facility borrowings	1,242,431	0	0	0	250,000	
23	Proceeds from the issuance of stock	48,286	114,276	69,375	68,214	37,855	
24	Dividends paid	-332,986	-344,128	-361,208	-367,499	-378,325	
25	Stock repurchased	-150,095	-200,031	-200,038	-200,003	-597,738	
26	**Net cash used in financing activities**	**807,636**	**-429,883**	**-791,871**	**-499,288**	**-695,303**	<-- =SUM(F21:F25)
27							
28	**Changes in cash balances**	**1,224,005**	**62,907**	**-234,129**	**52,691**	**-118,531**	<-- =F12+F18+F26
29							
30	**Supplemental disclosure of cash flow information**						
31	Cash paid during the period for						
32	Income taxes	255,043	175,972	314,735	283,618	305,094	
33	Interest	83,553	83,551	70,351	57,151	57,910	
34							
35	Income tax rate	34.73%	26.20%	37.07%	35.92%	36.96%	<-- =F32/(F4+F32)

In order to turn this consolidated statement of cash flows (CSCFs) into free cash flows (FCFs), we eliminate all the financial items. We also adjust the figures by adding back after-tax net interest. Since we want to use the CSCF to produce forward-looking FCFs, we may also want to eliminate operating or investment items that are not expected to recur.[1] In most cases these adjustments amount to the following:

- Accept all operating activities in the CSCF.
- Dispense with all financing activities.
- Carefully examine the investing activities, eliminating those that are financial but keeping the operational items.
- Add back after-tax interest.

In the current example:

	A	B	C	D	E	F	G
1	**ABC CORPORATION** **Consolidated Statement of Cash Flows, 2008-2012**						
2		**2008**	**2009**	**2010**	**2011**	**2012**	
3	**Operating Activities:**						
4	Net earnings	479,355	495,597	534,268	505,856	520,273	
5	Adjustments to reconcile net earnings to net cash provided by operating activities						
6	Add back depreciation and amortization	41,583	47,647	46,438	45,839	46,622	
7	Changes in operating assets and liabilities:						
8	Subtract increase in accounts receivable	9,387	25,951	-12,724	1,685	-2,153	
9	Subtract increase in inventories	-37,630	-22,780	-16,247	-15,780	-5,517	
10	Subtract increase in prepaid expenses and other assets	-52,191	13,573	16,255	14,703	-2,975	
11	Add increase in accounts payable, accrued expenses, pensions, and other liabilities	29,612	51,172	6,757	40,541	60,255	
12	**Net cash provided by operating activities**	**470,116**	**611,160**	**574,747**	**592,844**	**616,505**	<-- =SUM(F4:F11)
13							
14	**Investing Activities:**						
15	Short-term investments, net						
16	Purchases of property, plant and equipment	-48,944	-70,326	-89,947	-37,044	-88,426	
17	Proceeds from dispositions of property, plant and equipment	197	6,956	22,942	6,179	28,693	
18	**Net cash used in investing activities**	**-53,747**	**-118,370**	**-67,005**	**-30,865**	**-59,733**	<-- =SUM(F15:F17)
19							
20	**Financing Activities:**						
21	Repayment of debt						
22	Proceeds from revolving credit facility borrowings						
23	Proceeds from the issuance of stock						
24	Dividends paid						
25	Stock repurchased						
26	**Net cash used in financing activities**						
27							
28	**Free cash flow before interest adjustment**	**416,369**	**492,790**	**507,742**	**561,979**	**556,772**	<-- =F12+F18+F26
29	**Add back after-tax net interest**	**54,537**	**61,658**	**44,271**	**36,620**	**36,504**	<-- =(1-F37)*F35
30	**Free cash flow (FCF)**	**470,906**	**554,448**	**552,013**	**598,599**	**593,276**	<-- =F28+F29
31							
32	**Supplemental disclosure of cash flow information**						
33	Cash paid during the period for						
34	Income taxes	255,043	175,972	314,735	283,618	305,094	
35	Interest	83,553	83,551	70,351	57,151	57,910	
36							
37	Income tax rate	34.73%	26.20%	37.07%	35.92%	36.96%	<-- =F34/(F4+F34)

1. The current example has no such items.

The Enterprise Value and the Value per Share

At this point we have estimates for the firm's historical FCFs. In order to apply the methodology, we need to estimate three parameters:

- The short-term growth rate of ABC cash flows.
- The long-term growth rate of ABC cash flows.
- The weighted average cost of capital (WACC).

Some analysis, combined with seat-of-the-pants intuition, leads to the following parameter choices and valuation. The **Data Table** does sensitivity analysis on the long-term growth rate and the WACC.

	A	B	C	D	E	F	G	H
1	**ABC CORP. VALUATION**							
2	Free cash flow (FCF) year ending 31 Dec. 2012	593,276	<-- ='ABC, CSCF to FCF'!F30					
3	Growth rate of FCF, years 1-5	8.00%	<-- Optimistic about short-term growth					
4	Long-term FCF growth rate	5.00%	<-- More pessimistic about long-term growth					
5	Weighted average cost of capital (WACC)	10.70%						
6								
7	**Year**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	
8	FCF		640,738	691,997	747,357	807,145	871,717	<-- =F8*(1+$B$3)
9	Terminal value						30,510,086	<-- =G8*(1+B4)/(B3-B4)
10	Total		640,738	691,997	747,357	807,145	31,381,803	<-- =G8+G9
11								
12	Enterprise value	22,209,831	<-- =NPV(B5,C10:G10)*(1+B5)^0.5					
13	Add back initial cash and marketable securities	73,697	<-- From current balance sheet					
14	Subtract out 2012 financial liabilities	1,379,106	<-- From current balance sheet					
15	Equity value	20,904,422	<-- =B12+B13-B14					
16	Per share (1 million shares outstanding)	20.90	<-- =B15/1000000					
17								
18	**Data table: Share value vs LT growth and WACC**							
19	data table header,		Long-term growth ↓					
20	=IF(B5>B4,B16,"nmf") -->	20.90	0%	3%	6%	9%	12%	
21		6%	10.31	15.74	nmf	nmf	nmf	
22	WACC →	8%	9.48	14.48	34.45	nmf	nmf	
23		10%	8.74	13.33	31.73	-60.24	nmf	
24		12%	8.06	12.30	29.26	-55.55	nmf	
25		14%	7.44	11.35	27.02	-51.29	-12.14	
26		16%	6.87	10.49	24.98	-47.44	-11.23	
27		18%	6.36	9.71	23.12	-43.93	-10.40	

The data table header (our nomenclature for what the **Data Table** actually computes) includes an **If** statement. The reason is that, as shown above, the terminal value formula is valid only if the long-term growth rate is less than the WACC.

In the following sections we deal with a number of issues that arise in this valuation process.

4.4 Merck: Reverse Engineering the Market Value

The method discussed in this chapter can often be used to extract market expectations of growth. As an example, consider Merck. Manipulation of the CSCF produces a 2011 FCF of $10,346 million. Using the WACC of 5.66% computed in Chapter 3 and some arbitrary values for the short- and long-term FCF growth rates gives us the valuation template for Merck below.[2]

	A	B	C	D	E	F	G	H
1	**MERCK CASH FLOW VALUATION TEMPLATE**							
2	Current free cash flow (FCF)	10,346	<-- ='Merck Free Cash Flow'!D51					
3	Growth rate of FCF, years 1-5	3.00%						
4	Long-term FCF growth rate	3.00%	<-- critical question					
5	Weighted average cost of capital (WACC)	5.66%	<-- Chapter 2					
6								
7	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
8	FCF		10,656	10,976	11,305	11,645	11,994	<-- =F8*(1+$B$3)
9	Terminal value						464,423	<-- =G8*(1+B4)/(B5-B4)
10	Total (sum of FCF+terminal)		10,656	10,976	11,305	11,645	476,417	<-- =G8+G9
11								
12	Enterprise value	411,797	<-- =NPV(B5,C10:G10)*(1+B5)^0.5					
13	Add back initial cash	14,972	<-- =13531+1441					
14	Subtract out debt	17,515	<-- From current balance sheet					
15	Equity value	409,254	<-- =B12+B13-B14					
16	Number of shares outstanding	3,040,838,643	<-- Merck's financial statements					
17	Share value from our model	134.59	<-- =B15/B16*1000000					
18	Actual stock price (30 Dec 2011)	37.70						

The WACC for Merck was discussed in Chapter 3, where we saw that plausible estimates are in the range of 5% to 7%; the WACC = 5.66% was determined in Chapter 3 as an average of these plausible estimates. If the short-term growth rate is 6% and the long-term growth is 3%, then our model produces a share value of $154.43, which is very much larger than Merck's stock price of $37.70. We use a **Data Table** to find what current market price represents in terms of long-term growth and WACC:

2. We have skipped many of the details, which are available on the Excel file for this chapter.

	A	B	C	D	E	F	G
20	**Data table: Model share price as function of LT growth and WACC, short-term FCF growth rate = 6%**						
21			Long-term growth rate ↓				
22	=IF(B5>B4,B17,"nmf") -->	154.43	-5%	-4%	-3%	-2%	0%
23		5.0%	51.83	56.09	61.42	68.28	90.21
24		5.2%	50.77	54.83	59.89	66.35	86.71
25		5.4%	49.76	53.63	58.43	64.52	83.48
26	WACC -->	5.6%	48.78	52.48	57.03	62.79	80.47
27		5.8%	47.84	51.37	55.70	61.15	77.67
28		6.0%	46.93	50.31	54.43	59.59	75.06
29		6.2%	46.06	49.29	53.22	58.11	72.62
30		6.4%	**45.22**	48.31	52.05	56.70	70.33
31		6.6%	**44.40**	47.36	50.94	55.35	68.18
32		6.8%	**43.62**	46.45	49.87	54.07	66.15
33		7.0%	**42.86**	45.58	48.85	52.84	64.25
34		7.2%	**42.12**	**44.74**	47.86	51.67	62.44
35		7.4%	**41.41**	**43.92**	46.91	50.54	60.74

We have used Excel's **Conditional Formatting** to highlight Merck share valuations that are +/– 20% of the current market price. It is now clear that the market imputes a negative long-term growth rate to Merck's free cash flows. We can also do some sensitivity analysis on the short-term growth rate (perhaps it doesn't make sense that Merck's long-term growth is so low and that its short-term growth is 6%). Changing the short-term growth to 3% produces the following **Data Table**:

	A	B	C	D	E	F	G
20	**Data table: Model share price as function of LT growth and WACC, short-term FCF growth rate = 3%**						
21			Long-term growth rate ↓				
22	=IF(B5>B4,B17,"nmf") -->	134.59	-5%	-4%	-3%	-2%	0%
23		5.0%	45.71	49.40	54.02	59.96	78.96
24		5.2%	**44.79**	48.31	52.69	58.28	75.93
25		5.4%	**43.91**	47.26	51.42	56.70	73.12
26	WACC -->	5.6%	**43.06**	46.26	50.21	55.20	70.51
27		5.8%	**42.24**	45.30	49.05	53.77	68.09
28		6.0%	**41.45**	**44.38**	47.95	52.42	65.82
29		6.2%	**40.69**	**43.49**	46.90	51.13	63.70
30		6.4%	**39.96**	**42.64**	45.89	49.91	61.72
31		6.6%	**39.25**	**41.82**	**44.92**	48.74	59.85
32		6.8%	**38.57**	**41.03**	**43.99**	47.62	58.10
33		7.0%	**37.91**	**40.27**	**43.10**	46.56	56.44
34		7.2%	**37.27**	**39.54**	**42.24**	45.54	54.88
35		7.4%	**36.66**	**38.83**	**41.42**	**44.56**	53.40

The conclusion remains the same: Market expectations of Merck's long-term growth prospects appear to be negative.

4.5 Summary

This chapter has illustrated a relatively corporate simple valuation technique. Starting with free cash flows derived from the consolidated statement of cash flows, we build a simple valuation template that is a function of only four parameters: the current FCF, the short-term FCF growth rate, the long-term FCF growth rate, and the weighted average cost of capital (WACC). Our technique allows us to focus on the main valuation parameters and also allows us to reverse engineer the growth and WACC expectations built into the current market price.

Exercise

The files that accompany this book contains information for Kellogg. Use this information and the template of section 4.1 to value Kellogg.

5 Pro Forma Financial Statement Modeling

5.1 Overview

The usefulness of financial statement projections for corporate financial management is undisputed. Such projections, termed *pro forma financial statements*, are the bread and butter for much corporate financial analysis. In this and the next chapter we will focus on the use of pro formas for valuing the firm and its component securities, but pro formas also form the basis for many credit analyses; by examining pro forma financial statements we can predict how much financing a firm will need in future years. We can play the usual "what if" games of simulation models, and we can use pro formas to ask what strains on the firm may be caused by changes in financial and sales parameters.

In this chapter we present a variety of financial models. All the models are sales driven, in that they assume that many of the balance sheet and income statement items are directly or indirectly related to sales. The mathematical structure of solving the models involves finding the solution to a set of simultaneous linear equations predicting both the balance sheets and the income statements for the coming years. However, the user of a spreadsheet need never worry about the solution of the model; the fact that spreadsheets can solve—by iteration—the financial relations of the model means that we only have to worry about correctly stating the relevant accounting relations in our Excel model.

5.2 How Financial Models Work: Theory and an Initial Example

Almost all financial statement models are *sales driven*; this term means that as many as possible of the most important financial statement variables are assumed to be functions of the sales level of the firm. For example, accounts receivable are often taken as a direct percentage of the sales of the firm. A slightly more complicated example might postulate that the fixed assets (or some other account) are a step function of the level of sales:

$$Fixed\ assets = \begin{cases} a & if\ sales < A \\ b & if\ A \le sales < B \\ etc. & \end{cases}$$

To solve a financial planning model, we must distinguish between those financial statement items that are functional relationships of sales and perhaps of other financial statement items and those items that involve policy decisions. The asset side of the balance sheet is usually assumed to be dependent only on functional relationships. The current liabilities may also be taken to involve functional relationships only, leaving the mix between long-term debt and equity as a policy decision.

A simple example is the following. We wish to predict the financial statements for a firm whose current balance sheet and income statement are as follows:

	A	B
13	**Year**	**0**
14	**Income statement**	
15	Sales	1,000
16	Costs of goods sold	(500)
17	Interest payments on debt	(32)
18	Interest earned on cash and marketable securities	6
19	Depreciation	(100)
20	Profit before tax	374
21	Taxes	(150)
22	Profit after tax	225
23	Dividends	(90)
24	Retained earnings	135
25		
26	**Balance sheet**	
27	Cash and marketable securities	80
28	Current assets	150
29	Fixed assets	
30	At cost	1,070
31	Depreciation	(300)
32	Net fixed assets	770
33	**Total assets**	1,000
34		
35	Current liabilities	80
36	Debt	320
37	Stock	450
38	Accumulated retained earnings	150
39	**Total liabilities and equity**	1,000

The current (year 0) level of sales is 1,000. The firm expects its sales to grow at a rate of 10% per year, and it anticipates the following financial statement relations:

Current assets:	Assumed to be 15% of end-of-year sales
Current liabilities:	Assumed to be 8% of end-of-year sales
Net fixed assets:	77% of end-of-year sales
Depreciation:	10% of the average value of assets on the books during the year.
Fixed assets at cost:	Sum of net fixed assets plus accumulated depreciation.
Debt:	The firm neither repays any existing debt nor borrows any more money over the 5-year horizon of the pro formas.
Cash and marketable securities:	This is the balance sheet *plug* (see explanation below). Average balances of cash and marketable securities are assumed to earn 8% interest.

The "Plug"

Perhaps the most important financial policy variable in the financial statement modeling is the "plug": This relates to the decision as to which balance sheet item will "close" the model:

- How do we guarantee that assets and liabilities are equal (this is "closure" in the accounting sense)?
- How does the firm finance its incremental investments (this is "financial closure")?

In general the plug in a pro forma model will be one of three financial balance sheet items: (i) Cash and marketable securities, (ii) debt, or (iii) stock.[1] As an example, consider the balance sheet of our first pro forma model:

1. As noted in Chapter 3, cash can often be considered negative debt and vice versa. We return to this point in section 5.5.

Assets	Liabilities and Equity
Cash and marketable securities	Current liabilities
Current assets	Debt
Fixed assets	Equity
Fixed assets at cost	Stock (net funds directly provided by shareholders)
– Accumulated depreciation	Accumulated retained earnings (profits not paid out)
Net fixed assets	
Total assets	**Total liabilities and equity**

In the current example we assume that cash and marketable securities will be the plug. This assumption has two meanings:

1. The *mechanical* meaning of the plug: Formally, we define

Cash and marketable securities = *Total liabilities and equity*
– Current assets – Net fixed assets

By using this definition, we guarantee that assets and liabilities will always be equal.

2. The *financial* meaning of the plug: By defining the plug to be cash and marketable securities, we are also making a statement about how the firm finances itself. In our model below, for example, the firm sells no additional stock, does not pay back any of its existing debt, and does not raise any more debt. This definition means that all incremental financing (if needed) for the firm will come from the cash and marketable securities account; it also means that if the firm has additional cash, it will go into this account.

Projecting Next Year's Balance Sheet and Income Statement

Above we have given the financial statement for year 0. We now project the financial statement for year 1:

	A	B	C	D
1	**SETTING UP THE FINANCIAL STATEMENT MODEL**			
2	Sales growth	10%		
3	Current assets/Sales	15%		
4	Current liabilities/Sales	8%		
5	Net fixed assets/Sales	77%		
6	Costs of goods sold/Sales	50%		
7	Depreciation rate	10%		
8	Interest rate on debt	10.00%		
9	Interest paid on cash and marketable securities	8.00%		
10	Tax rate	40%		
11	Dividend payout ratio	40%		
12				
13	**Year**	**0**	**1**	
14	**Income statement**			
15	Sales	1,000	1,100	<-- =B15*(1+$B$2)
16	Costs of goods sold	(500)	(550)	<-- =-C15*$B$6
17	Interest payments on debt	(32)	(32)	<-- =-$B$8*(B36+C36)/2
18	Interest earned on cash and marketable securities	6	9	<-- =$B$9*(B27+C27)/2
19	Depreciation	(100)	(117)	<-- =-$B$7*(C30+B30)/2
20	Profit before tax	374	410	<-- =SUM(C15:C19)
21	Taxes	(150)	(164)	<-- =-C20*$B$10
22	Profit after tax	225	246	<-- =C21+C20
23	Dividends	(90)	(98)	<-- =-$B$11*C22
24	Retained earnings	135	148	<-- =C23+C22
25				
26	**Balance sheet**			
27	Cash and marketable securities	80	144	<-- =C39-C28-C32
28	Current assets	150	165	<-- =C15*$B$3
29	Fixed assets			
30	At cost	1,070	1,264	<-- =C32-C31
31	Depreciation	(300)	(417)	<-- =B31+C19
32	Net fixed assets	770	847	<-- =C15*$B$5
33	**Total assets**	1,000	1,156	<-- =C32+C28+C27
34				
35	Current liabilities	80	88	<-- =C15*$B$4
36	Debt	320	320	<-- =B36
37	Stock	450	450	<-- =B37
38	Accumulated retained earnings	150	298	<-- =B38+C24
39	**Total liabilities and equity**	1,000	1,156	<-- =SUM(C35:C38)

The formulas are mostly obvious. (The dollar signs—indicating that when the formulas are copied the cell references to the model parameters should not change—are very important! If you fail to put them in the model will not copy correctly when you project years 2 and beyond.) In each year (model parameters are in boldface):

Income Statement Equations

- Sales = Initial sales ∗ (1 + **Sales growth**)^{year}
- Costs of goods sold = Sales ∗ **Costs of goods sold/Sales**

The assumption is that the only expenses related to sales are costs of goods sold. Most companies also book an expense item called Selling, General, and Administrative expenses (SG&A). The change you would have to make to accommodate this item are obvious (see an exercise at the end of this chapter).

- Interest payments on debt = **Interest rate on debt** ∗ Average debt over the year

This formula allows us to accommodate changes in the model for repayment of debt, and rollover of debt at different interest rates. Note that in the current version of the model, debt stays constant, but in other versions of the model discussed below debt will vary over time.

- Interest earned on cash and marketable securities = **Interest rate on cash** ∗ Average cash and marketable securities over the year
- Depreciation = **Depreciation rate** ∗ Average fixed assets at cost over the year

This calculation assumes that all new fixed assets are purchased during the year. We also assume that there is no disposal of fixed assets.

- Profit before taxes = Sales – Costs of goods sold – Interest payments on debt + Interest earned on cash and marketable securities – Depreciation
- Taxes = **Tax rate** ∗ Profit before taxes
- Profit after taxes = Profit before taxes – Taxes
- Dividends = **Dividend payout ratio** ∗ Profit after taxes.

The firm is assumed to pay out a fixed percentage of its profits as dividends. An alternative would be to assume that the firm has a target for its dividends per share.

- Retained earnings = Profit after taxes – Dividends

Balance Sheet Equations

- Cash and marketable securities = Total liabilities and equity – Current assets – Net fixed assets

As explained above, this means that cash and marketable securities are the balance sheet plug.

- Current assets = **Current assets/Sales** ∗ Sales
- Net fixed assets = **Net fixed assets/Sales** ∗ Sales[2]
- Accumulated depreciation = Previous year's accumulated depreciation + **Depreciation rate** ∗ Average fixed assets at cost over the year
- Fixed assets at cost = Net fixed assets + Accumulated depreciation

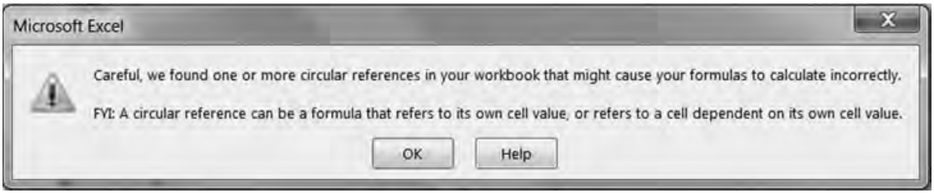
Note that this model does not distinguish between plant property and equipment (PP&E) and other fixed assets such as land.

- Current liabilities = **Current liabilities/Sales** ∗ Sales
- Debt is assumed to be unchanged. An alternative model, which we will explore later, assumes that debt is the balance sheet plug.
- Stock doesn't change. (Shareholders provide no additional direct financing: the company is assumed to issue no new stock or repurchase any stock.)
- Accumulated retained earnings = Previous year's accumulated retained earnings + Current year's additions to retained earnings

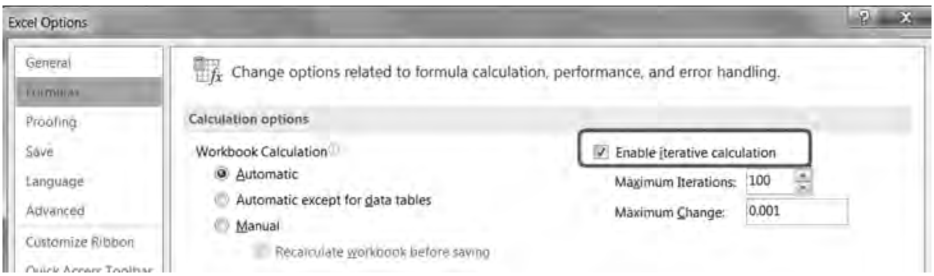
2. This is not the only way to model fixed assets. An alternative method assumes that *net* fixed assets are constant; see section 5.6 for an implementation.

Circular References in Excel

Financial statement models in Excel almost always involve cells that are mutually dependent. As a result the solution of the model depends on the ability of Excel to solve circular references. If you open a spreadsheet that involves iteration and if your spreadsheet is not set up for circular references you will see the following Excel error message:



To make sure your spreadsheet recalculates, you have to go to the **File|Options|Formulas** box and click **Enable iterative calculation**:



Extending the Model to Years 2 and Beyond

Now that you have the model set up, you can extend it by copying the columns:

	A	B	C	D	E	F	G
1	**FIRST FINANCIAL MODEL**						
2	Sales growth	10%					
3	Current assets/Sales	15%					
4	Current liabilities/Sales	8%					
5	Net fixed assets/Sales	77%					
6	Costs of goods sold/Sales	50%					
7	Depreciation rate	10%					
8	Interest rate on debt	10.00%					
9	Interest paid on cash and marketable securities	8.00%					
10	Tax rate	40%					
11	Dividend payout ratio	40%					
12							
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
14	**Income statement**						
15	Sales	1,000	1,100	1,210	1,331	1,464	1,611
16	Costs of goods sold	(500)	(550)	(605)	(666)	(732)	(805)
17	Interest payments on debt	(32)	(32)	(32)	(32)	(32)	(32)
18	Interest earned on cash and marketable securities	6	9	14	20	26	33
19	Depreciation	(100)	(117)	(137)	(161)	(189)	(220)
20	Profit before tax	374	410	450	492	538	587
21	Taxes	(150)	(164)	(180)	(197)	(215)	(235)
22	Profit after tax	225	246	270	295	323	352
23	Dividends	(90)	(98)	(108)	(118)	(129)	(141)
24	Retained earnings	135	148	162	177	194	211
25							
26	**Balance sheet**						
27	Cash and marketable securities	80	144	213	289	371	459
28	Current assets	150	165	182	200	220	242
29	Fixed assets						
30	At cost	1,070	1,264	1,486	1,740	2,031	2,364
31	Depreciation	(300)	(417)	(554)	(715)	(904)	(1,124)
32	Net fixed assets	770	847	932	1,025	1,127	1,240
33	**Total assets**	1,000	1,156	1,326	1,513	1,718	1,941
34							
35	Current liabilities	80	88	97	106	117	129
36	Debt	320	320	320	320	320	320
37	Stock	450	450	450	450	450	450
38	Accumulated retained earnings	150	298	460	637	830	1,042
39	**Total liabilities and equity**	1,000	1,156	1,326	1,513	1,718	1,941

Note that the most common mistake to make in the transition between the two-columned financial model and this one is the failure to mark the model parameters with dollar signs. If you commit this error, you will get zeros in places where there should be numbers.

5.3 Free Cash Flow (FCF): Measuring the Cash Produced by the Business

Now that we have the model, we can use it to make financial predictions. The most important calculation for valuation purposes is the *free cash flow* (FCF). FCF—the cash produced by a business without taking into account the way the business is financed—is the best measure of the cash produced by a business. An extended discussion of the FCF was included in Chapter 2. For reference, we briefly repeat the definition given in that chapter:

<table>
<tr><th colspan="3">Defining the Free Cash Flow</th></tr>
<tr><td>Profit after taxes</td><td colspan="2">Accounting measure of firm profitability. It is not a cash flow.</td></tr>
<tr><td>+ Depreciation</td><td colspan="2">This noncash expense is added back to the profit after tax.</td></tr>
<tr><td>− Increase in operating current assets</td><td>Increase in sales-related current assets is not an expense for tax purposes (and is therefore ignored in the profit after taxes), but it is a cash drain on the company.</td><td rowspan="2">For purposes of FCF, our definitions of current assets and current liabilities excludes financing items such as cash and debt.</td></tr>
<tr><td>+ Increase in operating current liabilities</td><td>Increase in sale-related current liabilities provides cash to the firm.</td></tr>
<tr><td>− Increase in fixed assets at cost</td><td colspan="2">An increase in fixed assets (the long-term productive assets of the company) is a use of cash, which reduces the firm's free cash flow.</td></tr>
<tr><td>+ After-tax net interest payments</td><td colspan="2">FCF measures the cash produced by the business activity of the firm. Add back after-tax net interest payments to neutralize the interest component of the profit after taxes.</td></tr>
</table>

Here is the calculation for our firm:

	A	B	C	D	E	F	G
40							
41	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
42	**Free cash flow calculation**						
43	Profit after tax		246	270	295	323	352
44	Add back depreciation		117	137	161	189	220
45	Subtract increase in current assets		(15)	(17)	(18)	(20)	(22)
46	Add back increase in current liabilities		8	9	10	11	12
47	Subtract increase in fixed assets at cost		(194)	(222)	(254)	(291)	(333)
48	Add back after-tax interest on debt		19	19	19	19	19
49	Subtract after-tax interest on cash and mkt. securities		(5)	(9)	(12)	(16)	(20)
50	**Free cash flow**		176	188	201	214	228

Reconciling the Cash Balances

The free cash flow calculation is different from the "consolidated statement of cash flows" that is a part of every accounting statement. The purpose of the consolidated statement of cash flows is to explain the increase in the cash accounts in the balance sheet as a function of the cash flows from the firm's operating, investing, and financing activities. In the pro forma example of this section we treat the cash and marketable securities as the balance sheet plug; however, it can also be derived from a standard accounting statement of cash flows:

	A	B	C	D	E	F	G	H
53	**CONSOLIDATED STATEMENT OF CASH FLOWS: RECONCILING THE CASH BALANCES**							
54	**Cash flow from operating activities**							
55	Profit after tax		246	270	295	323	352	<-- =G22
56	Add back depreciation		117	137	161	189	220	<-- =-G19
57	Adjust for changes in net working capital:							
58	Subtract increase in current assets		(15)	(17)	(18)	(20)	(22)	<-- =-(G28-F28)
59	Add back increase in current liabilities		8	9	10	11	12	<-- =G35-F35
60	Net cash from operating activities		356	400	448	502	562	<-- =SUM(G55:G59)
61								
62	**Cash flow from investing activities**							
63	Aquisitions of fixed assets--capital expenditures		(194)	(222)	(254)	(291)	(333)	<-- =-(G30-F30)
64	Purchases of investment securities		0	0	0	0	0	<-- Not in our model
65	Proceeds from sales of investment securities		0	0	0	0	0	<-- Not in our model
66	Net cash used in investing activities		(194)	(222)	(254)	(291)	(333)	<-- =SUM(G63:G65)
67								
68	**Cash flow from financing activities**							
69	Net proceeds from borrowing activities		0	0	0	0	0	<-- =G36-F36
70	Net proceeds from stock issues, repurchases		0	0	0	0	0	<-- =G37-F37
71	Dividends paid		(98)	(108)	(118)	(129)	(141)	<-- =G23
72	Net cash from financing activities		(98)	(108)	(118)	(129)	(141)	<-- =SUM(G69:G71)
73								
74	Net increase in cash and cash equivalents		64	70	76	82	88	<-- =G72+G66+G60
75	Check: changes in cash and mkt. securities		64	70	76	82	88	<-- =G27-F27

Line 75 checks that the changes in the cash accounts derived through the consolidated statement of cash flows match those derived in the financial model (which uses cash as a plug). As you can see, the model works, in the sense that changes in cash balances from the consolidated statement of cash flows in fact match those in the projected balance sheets of the pro forma model.

5.4 Using the Free Cash Flow (FCF) to Value the Firm and Its Equity

The *enterprise value* of the firm is the present value of the firm's future anticipated free cash flows. We can use the pro forma FCF projections and a cost of capital to determine the enterprise value of the firm. Suppose we have determined that the firm's weighted average cost of capital (WACC) is 20% (recall that the calculation of the WACC was discussed in Chapter 2). Then the *enterprise value* of the firm is the discounted value of the firm's projected FCFs plus its terminal value:

$$Enterprise\ value = \sum_{t=1}^{\infty} \frac{FCF_1}{(1+WACC)^t}$$

Most financial analysts consider it presumptuous to project an infinite number of free cash flows; therefore the projected cash flow stream is often cut off at some arbitrary date and a *terminal value* is substituted for the cash flows beyond this date:

$$Enterprise\ value = \frac{FCF_1}{(1+WACC)^1} + \frac{FCF_2}{(1+WACC)^2} + \ldots + \frac{FCF_5}{(1+WACC)^5} + \frac{Year\text{-}5\ Terminal\ Value}{(1+WACC)^5}$$

In this formula, the *Year-5 Terminal Value* is a proxy for the present value of all FCFs from year 6 onward. Instead of projecting the FCFs from year 6 onward, we use the most common terminal value model, assuming that year-5 free cash flows grow at a long-term growth rate *LTg*:

$$Terminal\ Value\ at\ end\ of\ year\ 5 = \sum_{t=1}^{\infty} \frac{FCF_{t+5}}{(1+WACC)^t} = \sum_{t=1}^{\infty} \frac{FCF_5 * (1+LTg)^t}{(1+WACC)^t}$$
$$= \frac{FCF_5 * (1+LTg)}{WACC - LTg}$$
$$\text{provided } |LTg| < WACC$$

This model (based on the formula for the present value of a growing annuity, see section 1.1) assumes that the year-5 cash flow will continue to grow at a constant long-term growth rate. Note that the formula is senseless if the long-term growth rate *LTg* is greater than or equal to the weighted average cost of capital; if this were the case, then the terminal value would be infinite (clearly impossible).

Here's an example which uses our projections:

	A	B	C	D	E	F	G	H	I	J
53	**Valuing the firm**									
54	Weighted average cost of capital	20%								
55	Long-term free cash flow growth rate	5%	<-- real growth 2% + inflation 3%?							
56										
57	**Year**	**0**	**1**	**2**	**3**	**4**	**5**			
58	FCF		176	188	201	214	228			
59	Terminal value						1,598	<-- =G58*(1+B55)/(B54-B55)		
60	Total		176	188	201	214	1,826			
61										
62	Enterprise value, present value of row 60	1,231	<-- =NPV(B54,C60:G60)							
63	Add in initial (year 0) cash and mkt. securities	80	<-- =B27							
64	Asset value in year 0	1,311	<-- =B63+B62							
65	Subtract out value of firm's debt today	(320)	<-- =-B36							
66	Equity value	991	<-- =B64+B65							

Note that the long-term FCF growth rate in cell B55 is different from the sales growth in cell B2 in the spreadsheet on page 135. The sales growth is the anticipated growth over the years 1–5; the long-term growth rate is probably better estimated by making a more realistic estimate of the growth of the firm's market sector. For firms operating in a mature market, we often estimate the long-term FCF growth as the sum of real growth plus anticipated inflation.

5.5 Some Notes on the Valuation Procedure

In this section we cover some issues related to the valuation procedure outlined in section 5.4.

Terminal Value

In determining the terminal value we used a version of the growing annuity model described in Chapter 1. We have assumed that—after the year-5 projection horizon—the cash flows will grow at a long-term rate of growth of 5%. This gives the terminal value as:

$$Terminal\ Value\ at\ end\ of\ year\ 5 = \frac{FCF_5 * (1 + Long\text{-}term\ FCF\ growth)}{WACC - Long\text{-}term\ FCF\ growth}$$

As noted in the previous section, this formula is only valid if the long-term FCF growth is less than the WACC.

There are other ways of calculating the terminal value. All of the following are common variations that can be implemented in the framework of our model (see end-of-chapter exercises):

- Terminal value = Year-5 book value of debt + Equity

  This calculation assumes that the book value correctly predicts the market value.

- Terminal value = (Enterprise market/book multiple) ∗ (Year-5 book value of debt + Equity)
- Terminal value = P/E ratio ∗ Year-5 profits + Year-5 book value of debt
- Terminal value = EBITDA ratio ∗ Year-5 anticipated EBITDA

  (EBITDA = Earnings before interest, taxes, depreciation, and amortization.)

The Treatment of Cash and Marketable Securities in the Valuation

We have added the initial cash balances back to the present value of the projected FCFs to get the enterprise value. This procedure assumes the following:

- Year-0 balances of cash and marketable securities are not needed to produce the FCFs in subsequent years.
- Year-0 balances of cash and marketable securities are "surpluses" which could be drawn down or paid out by shareholders without affecting the future economic performance of the firm.

A wholly equivalent assumption sometimes made by investment bankers and equity analysts is to assume that initial cash balances are *negative debt*. If you made this assumption, you would value the equity in the following way:

	A	B	C	D
68	**Cash and marketable securities as negative debt**			
69	NPV of row 60 = enterprise value	1,295	<-- =B62	
70	Net year 0 debt: debt minus cash	(240)	<-- =-B36+B27	
71	Equity value	1,055	<-- =B69+B70	

Mid-Year Discounting

While the NPV formula assumes that all cash flows occur at the end of the year, it is more logical to assume that they occur smoothly throughout the year. For discounting purposes, we should therefore discount cash flows as if, on average, they occur in the middle of the year. This means that the enterprise value is more logically calculated as:

$$Enterprise\ value = \frac{FCF_1}{(1+WACC)^{0.5}} + \frac{FCF_2}{(1+WACC)^{1.5}} + \ldots + \frac{FCF_5}{(1+WACC)^{4.5}} + \frac{Year\text{-}5\ Terminal\ Value}{(1+WACC)^{4..5}}$$

$$= \underbrace{\left[\frac{FCF_1}{(1+WACC)^1} + \frac{FCF_2}{(1+WACC)^2} + \ldots + \frac{FCF_5}{(1+WACC)^5} + \frac{Year\text{-}5\ Terminal\ Value}{(1+WACC)^5} \right]}_{\text{This can be calculated using Excel's NPV function}} * (1+WACC)^{0.5}$$

Incorporating this mid-year discounting into our value calculations gives:

	A	B	C	D	E	F	G	H	I	J
74	**Valuing the firm—using mid-year discounting**									
75	Weighted average cost of capital	20%								
76	Long-term free cash flow growth rate	5%								
77										
78	**Year**	**0**	**1**	**2**	**3**	**4**	**5**			
79	FCF		176	188	201	214	228			
80	Terminal value						1,598	<-- =G79*(1+B76)/(B75-B76)		
81	Total		176	188	201	214	1,826			
82										
83	Enterprise value, NPV of row 81	1,348	<-- =NPV(B75,C81:G81)*(1+B75)^0.5							
84	Add in initial (year 0) cash and mkt. securities	80	<-- =B27							
85	Asset value in year 0	1,428	<-- =B84+B83							
86	Subtract out value of firm's debt today	(320)	<-- =B65							
87	Equity value	1,108	<-- =B85+B86							

5.6 Alternative Modeling of Fixed Assets

The models in this chapter assume that the net fixed assets (NFA) are a function of sales. In effect this means that we assume that the depreciation of the fixed assets has actual economic meaning, so that the productive capacity of these assets is determined by their after-depreciation value. At the ethereal level of the pro forma models of this chapter, this appears to us to be an acceptable assumption.

There are, however, two alternative models that the financial modeler may want to consider. The first of these assumes that depreciation has no economic meaning. In this case the gross fixed assets are a function of sales. The second alternative model is to assume that the existing fixed asset base, if properly maintained, can accommodate reasonable levels of future sales.[3] Both alternative models are easily accommodated in the pro forma framework already set out; in the remainder of this section we give illustrations.

Gross Fixed Assets Are a Function of Sales

Suppose depreciation has no economic meaning, so that the fixed assets at cost represent the future productive capacity of assets.[4] This requires only a small adjustment to our previous model:

3. A third alternative is to extensively model future fixed asset expenditures. While this may add the whiff of "reality" to the financial model, it often just adds many layers of financial and modeling confusion.

4. Depreciation will of course continue to have a cash flow meaning, reducing taxable income.

	A	B	C	D	E	F	G	H
1	**FIXED ASSETS AT COST IS A FUNCTION OF SALES**							
2	Sales growth	10%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Fixed assets at cost/Sales	107%						
6	Costs of goods sold/Sales	50%						
7	Depreciation rate	10%						
8	Interest rate on debt	10.00%						
9	Interest paid on cash and marketable securities	8.00%						
10	Tax rate	40%						
11	Dividend payout ratio	40%						
26	**Balance sheet**							
27	Cash and marketable securities	80	229	398	589	805	1,049	
28	Current assets	150	165	182	200	220	242	
29	Fixed assets							
30	At cost	1,070	1,177	1,295	1,424	1,567	1,723	<-- =$B$5*G15
31	Depreciation	(300)	(412)	(536)	(672)	(821)	(986)	<-- =F31+G19
32	Net fixed assets	770	765	759	752	745	737	<-- =G30+G31
33	**Total assets**	1,000	1,158	1,338	1,541	1,770	2,028	

Net Fixed Assets Are Constant

In certain cases it may be appropriate to assume that the current fixed assets, if properly maintained, can accommodate reasonable levels of future sales. An example might be the case of a supermarket—if depreciation is taken to be the economic expression of the maintenance and asset replacement required to service the existing customer base, then this means that the net fixed assets would be constant over time. This assumption has been made in a number of Harvard cases.[5]

This variation in our basic model is easily made, as shown below:

	A	B	C	D	E	F	G	H
1	**NET FIXED ASSETS CONSTANT** **Depreciation reinvested into fixed assets**							
2	Sales growth	10%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Net fixed assets	Constant						
6	Costs of goods sold/Sales	50%						
7	Depreciation rate	10%						
8	Interest rate on debt	10.00%						
9	Interest paid on cash and marketable securities	8.00%						
10	Tax rate	40%						
11	Dividend payout ratio	40%						
26	**Balance sheet**							
27	Cash and marketable securities	80	223	386	570	777	1,010	
28	Current assets	150	165	182	200	220	242	
29	Fixed assets							
30	At cost	1,070	1,183	1,307	1,445	1,597	1,765	<-- =G32-G31
31	Depreciation	(300)	(413)	(537)	(675)	(827)	(995)	<-- =F31+G19
32	Net fixed assets	770	770	770	770	770	770	<-- =F32
33	**Total assets**	1,000	1,158	1,337	1,539	1,767	2,022	

5. See William Fruhan, *Financial Strategy: Studies in the Creation, Transfer, and Destruction of Shareholder Value* (Irwin, 1979), p. 161.

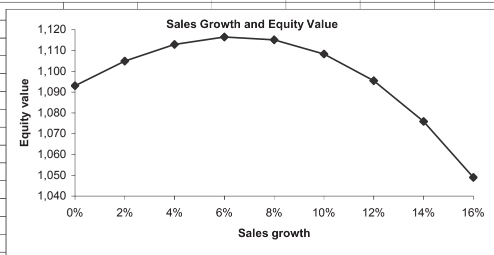
This model implies that depreciation equals capital expenditure. This can be seen in the free cash flows:

	A	B	C	D	E	F	G	H
41	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
42	**Free cash flow calculation**							
43	Profit after tax		251	284	320	361	406	
44	Add back depreciation		113	124	138	152	168	<-- =-G19
45	Subtract increase in current assets		(15)	(17)	(18)	(20)	(22)	
46	Add back increase in current liabilities		8	9	10	11	12	
47	Subtract increase in fixed assets at cost		(113)	(124)	(138)	(152)	(168)	<-- =-(G30-F30)
48	Add back after-tax interest on debt		19	19	19	19	19	
49	Subtract after-tax interest on cash and mkt. securities		(7)	(15)	(23)	(32)	(43)	
50	**Free cash flow**		255	281	308	339	372	

5.7 Sensitivity Analysis

As in any Excel model, we can perform extensive sensitivity analysis on our valuation. Taking the example in section 5.3 as our base case, we can ask, for example, what is the effect of the sales growth rate on the equity value of the firm:

	A	B	C	D	E	F	G	H	I	J
90	**Data table: The effect of sales growth (cell B2) on equity valuation**									
91		Growth	1,108	<-- =B87 , data table header						
92		0%	1,093							
93		2%	1,105							
94		4%	1,113							
95		6%	1,117							
96		8%	1,115							
97		10%	1,108							
98		12%	1,095							
99		14%	1,076							
100		16%	1,049							
101										
102										
103										
104										
105										
106										



Cells B91:C100 contain a data table (see Chapter 31 if you are unsure of how to construct these tables). While initial increases in sales growth increase the value of the firm, very high sales growth actually decreases firm value. We leave it to you to check that this is due to the high fixed assets to sales ratio.

Another variation is to calculate the effect on equity valuation of both the long-term FCF growth and the WACC. Here, however, you have to be careful: Examining the terminal value equation $Terminal\ Value = \dfrac{FCF_5 * (1 + long\text{-}term\ FCF\ growth)}{WACC - long\text{-}term\ FCF\ growth}$ will show you that this calculation only makes sense if the WACC is greater than the growth rate.[6] To overcome this problem we define the data table cell B107 (this is the calculation on which the data table does its sensitivity analysis) in the following way:

	A	B	C	D	E	F	G	H	I	J	K
107			**WACC ↓**								
108	=IF(B75<=B76,"nmf",B87) →	1,108.37	**10%**	**12%**	**14%**	**16%**	**18%**	**20%**	**22%**	**24%**	**26%**
109		**0%**	2,038.12	1,660.04	1,390.52	1,188.82	1,032.29	907.35	805.37	720.58	649.00
110	**Growth rate of sales -->**	**2%**	2,447.00	1,915.96	1,562.34	1,310.08	1,121.12	974.36	857.11	761.31	681.59
111		**4%**	3,128.45	2,299.84	1,802.89	1,471.75	1,235.34	1,058.12	920.35	810.19	720.10
112		**6%**	4,491.36	2,939.65	2,163.72	1,698.09	1,387.62	1,165.81	999.40	869.93	766.32
113		**8%**	8,580.08	4,219.26	2,765.09	2,037.61	1,600.82	1,309.39	1,101.03	944.61	822.81
114		**10%**	nmf	8,058.09	3,967.84	2,603.47	1,920.62	1,510.41	1,236.55	1,040.62	893.42
115		**12%**	nmf	nmf	7,576.07	3,735.18	2,453.61	1,811.94	1,426.27	1,168.64	984.20
116		**14%**	nmf	nmf	nmf	7,130.34	3,519.60	2,314.48	1,710.85	1,347.86	1,105.25
117		**16%**	nmf	nmf	nmf	nmf	6,717.58	3,319.58	2,185.15	1,616.70	1,274.71

5.8 Debt as a Plug

In the model shown above, cash and marketable securities were the plug and debt was a constant. However, for some values of the model parameters, you can get *negative* cash and marketable securities. Consider the following example—this is still the same model as above, but—as indicated on the spreadsheet itself—with some different parameter values:

6. If the growth rate > WACC, then $Terminal\ Value = \sum_{t=1}^{\infty} \dfrac{FCF_5 * (1 + long\text{-}term\ FCF\ growth)^t}{(1 + WACC)^t} = \infty$. Thus the WACC puts an effective bound on the long-term growth rate.

	A	B	C	D	E	F	G
1	**NEGATIVE CASH BALANCES: ILLUSTRATION**						
2	Sales growth	20%	<-- Increased from 10%				
3	Current assets/Sales	20%	<-- Increased from 15%				
4	Current liabilities/Sales	8%					
5	Net fixed assets/Sales	80%	<-- Increased from 77%				
6	Costs of goods sold/Sales	50%					
7	Depreciation rate	10%					
8	Interest rate on debt	10.00%					
9	Interest paid on cash and marketable securities	8.00%					
10	Tax rate	40%					
11	Dividend payout ratio	50%	<-- Increased from 40%				
12							
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
14	**Income statement**						
15	Sales	1,000	1,200	1,440	1,728	2,074	2,488
16	Costs of goods sold	(500)	(600)	(720)	(864)	(1,037)	(1,244)
17	Interest payments on debt	(40)	(40)	(40)	(40)	(40)	(40)
18	Interest earned on cash and marketable securities	6	4	(0)	(6)	(13)	(21)
19	Depreciation	(100)	(124)	(156)	(194)	(242)	(299)
20	Profit before tax	366	440	524	624	742	884
21	Taxes	(147)	(176)	(210)	(249)	(297)	(354)
22	Profit after tax	220	264	314	374	445	530
23	Dividends	(110)	(132)	(157)	(187)	(223)	(265)
24	Retained earnings	110	132	157	187	223	265
25							
26	**Balance sheet**						
27	Cash and marketable securities	80	28	(36)	(113)	(209)	(325)
28	Current assets	200	240	288	346	415	498
29	Fixed assets						
30	At cost	1,100	1,384	1,732	2,157	2,675	3,306
31	Depreciation	(300)	(424)	(580)	(774)	(1,016)	(1,315)
32	Net fixed assets	800	960	1,152	1,382	1,659	1,991
33	**Total assets**	1,080	1,228	1,404	1,615	1,865	2,163
34							
35	Current liabilities	80	96	115	138	166	199
36	Debt	400	400	400	400	400	400
37	Stock	450	450	450	450	450	450
38	Accumulated retained earnings	150	282	439	626	849	1,114
39	**Total liabilities and equity**	1,080	1,228	1,404	1,615	1,865	2,163

Given these changes the cash and marketable securities account (row 27 in the above example) turns negative by year 2, a result which is obviously illogical. However, the economic meaning of these negative numbers is clear: Given the increased sales growth, increased current asset and fixed asset requirements, and increased dividend payouts, the firm needs more financing.[7]

What we want is a model which recognizes that:

- Cash cannot be less than zero.
- When the firm needs additional financing, it *borrows* money.

7. If you examine the model as it now stands you will see that it implicitly assumes that this extra financing comes at the cost of the cash and marketable securities. If we consider this account a kind of checking account with interest, then the model implicitly assumes that the firm can finance overdrafts from this account at the same rate of interest as it is being paid on the account.

Here is the model:

	A	B	C	D	E	F	G	H
1	**NO NEGATIVE CASH BALANCES**							
2	Sales growth	20%	<-- Increased from 10%					
3	Current assets/Sales	20%	<-- Increased from 15%					
4	Current liabilities/Sales	8%						
5	Net fixed assets/Sales	80%	<-- Increased from 77%					
6	Costs of goods sold/Sales	50%						
7	Depreciation rate	10%						
8	Interest rate on debt	10.00%						
9	Interest paid on cash and marketable securities	8.00%						
10	Tax rate	40%						
11	Dividend payout ratio	50%	<-- Increased from 40%					
12								
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
14	**Income statement**							
15	Sales	1,000	1,200	1,440	1,728	2,074	2,488	
16	Costs of goods sold	(500)	(600)	(720)	(864)	(1,037)	(1,244)	
17	Interest payments on debt	(40)	(40)	(42)	(47)	(56)	(67)	
18	Interest earned on cash and marketable securities	6	4	1	-	-	-	
19	Depreciation	(100)	(124)	(156)	(194)	(242)	(299)	
20	Profit before tax	366	440	524	622	739	878	
21	Taxes	(147)	(176)	(209)	(249)	(296)	(351)	
22	Profit after tax	220	264	314	373	443	527	
23	Dividends	(110)	(132)	(157)	(187)	(222)	(263)	
24	Retained earnings	110	132	157	187	222	263	
25								
26	**Balance sheet**							
27	Cash and marketable securities	80	28	0	0	0	0	<-- =G39-G28-G32
28	Current assets	200	240	288	346	415	498	
29	Fixed assets							
30	At cost	1,100	1,384	1,732	2,157	2,675	3,306	
31	Depreciation	(300)	(424)	(580)	(774)	(1,016)	(1,315)	
32	Net fixed assets	800	960	1,152	1,382	1,659	1,991	
33	**Total assets**	1,080	1,228	1,440	1,728	2,074	2,488	
34								
35	Current liabilities	80	96	115	138	166	199	
36	Debt	400	400	436	514	610	728	<-- =MAX(G28+G32-G35-G37-G38,F36)
37	Stock	450	450	450	450	450	450	
38	Accumulated retained earnings	150	282	439	626	847	1,111	
39	**Total liabilities and equity**	1,080	1,228	1,440	1,728	2,074	2,488	

The equations for cash (row 27) and debt (row 36) are indicated for the year-5 entries. What they do, in accounting terms, is the following:

- Cash and marketable securities remains the plug in the model.
- The debt on the balance sheet conforms to the following test:
  - Current assets + Net fixed assets > Current liabilities + *Last year's debt* + Stock + Accumulated retained earnings?

  In this case even if cash and marketable securities = 0, we need to increase debt balances in order to finance the firm's productive activities.
  - Current assets + Net fixed assets < Current liabilities + *Last year's debt* + Stock + Accumulated retained earnings?

If this relation holds, then there is no need to increase debt and, in fact, the firm has to have positive cash and marketable securities as a balancing item, and the fact that we have made cash the plug will take care of this.

- In Excel programming terms, this formula becomes (for the year 5, but each previous year has the same type of equation): Max(G28+G32-G35-G37-G38,F36).

As shown in the Exercises for this chapter, the model can easily accommodate a situation in which there are minimum cash balances.

5.9 Incorporating a Target Debt/Equity Ratio into a Pro Forma

Another change we might want to make in our model relates to the plug. Suppose that the firm has a target ratio of debt to equity: In each of the years 1–5 it wants the ratio of debt/equity on the balance sheet to conform to a certain ratio. This situation is illustrated in the following example:

	A	B	C	D	E	F	G	H
1	**TARGET DEBT/EQUITY RATIO** **Cash is fixed, ratio of debt/equity changes in each year**							
2	Sales growth	10%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Net fixed assets/Sales	77%						
6	Costs of goods sold/Sales	50%						
7	Depreciation rate	10%						
8	Interest rate on debt	10.00%						
9	Interest paid on cash & marketable securities	8.00%						
10	Tax rate	40%						
11	Dividend payout ratio	60%						
12								
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
14	**Income statement**							
15	Sales	1,000	1,100	1,210	1,331	1,464	1,611	
16	Costs of goods sold	(500)	(550)	(605)	(666)	(732)	(805)	
17	Interest payments on debt	(32)	(30)	(29)	(28)	(29)	(32)	
18	Interest earned on cash & marketable securities	6	6	6	6	6	6	
19	Depreciation	(100)	(117)	(137)	(161)	(189)	(220)	
20	Profit before tax	374	409	445	483	521	560	
21	Taxes	(150)	(164)	(178)	(193)	(208)	(224)	
22	Profit after tax	225	246	267	290	313	336	
23	Dividends	(135)	(147)	(160)	(174)	(188)	(202)	
24	Retained earnings	90	98	107	116	125	134	
25								
26	**Balance sheet**							
27	Cash and marketable securities	80	80	80	80	80	80	
28	Current assets	150	165	182	200	220	242	
29	Fixed assets							
30	At cost	1,070	1,264	1,486	1,740	2,031	2,364	
31	Depreciation	(300)	(417)	(554)	(715)	(904)	(1,124)	
32	Net fixed assets	770	847	932	1,025	1,127	1,240	
33	**Total assets**	1,000	1,092	1,193	1,305	1,427	1,562	
34								
35	Current liabilities	80	88	97	106	117	129	
36	Debt	320	287	284	276	302	331	<-- =G41*(G37+G38)
37	Stock	450	469	457	451	412	372	<-- =G33-G35-G36-G38
38	Accumulated retained earnings	150	248	355	471	596	730	
39	**Total liabilities and equity**	1,000	1,092	1,193	1,305	1,427	1,562	
40								
41	**Target debt/equity ratio**	0.53	0.40	0.35	0.30	0.30	0.30	

Initial (year 0) debt/equity ratio: =B36/(B37+B38)

Row 41 of the spreadsheet shows the target debt/equity ratio in each of years 1–5. The firm wants to lower its current debt/equity ratio of 53% to 30% over the next 2 years. The relevant changes to the equations of our initial model are the following:

- Debt = **Target debt/Equity ratio** ∗ (Stock + Retained earnings)
- Stock = Total assets – Current liabilities – Debt – Accumulated retained earnings

Note that the firm will issue new debt years 4 and 5; in year 1 the stock account grows (indicating that new equity is issued), whereas in subsequent years stock decreases (indicating a repurchase of equity).

5.10 Project Finance: Debt Repayment Schedules

Here is another use for pro forma modeling: In a typical case of so-called "project finance," the firm borrows money in order to finance a project. The borrowing often comes with strings attached:

- The firm is not allowed to pay any dividends until the debt is paid off.
- The firm is not allowed to issue any new equity.
- The firm must pay back the debt over a specified period.

The following simplified example uses a variation of the version of our basic model with cash balances. A new firm or project is set up; in year 0:

- The firm has assets of 2,200, which are financed with 200 of current liabilities, 1,100 of equity, and 1,000 of debt.
- The debt must be paid off in equal installments of principal over the next 5 years. Until the debt is paid off, the firm is not allowed to pay dividends (if there is extra cash, this will go into a cash and marketable securities account).

	A	B	C	D	E	F	G	H
1	**PROJECT FINANCE** **No dividends, debt repayment schedule fixed, net fixed assets constant**							
2	Sales growth	15%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Costs of goods sold/Sales	45%						
6	Depreciation rate	10%						
7	Interest rate on debt	10.00%						
8	Interest paid on cash and marketable securities	8.00%						
9	Tax rate	40%						
10	Dividend payout ratio	0%	<-- No dividends until all the debt is paid off					
11								
12	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
13	**Income statement**							
14	Sales		1,150	1,323	1,521	1,749	2,011	
15	Costs of goods sold		(518)	(595)	(684)	(787)	(905)	
16	Interest payments on debt		(90)	(70)	(50)	(30)	(10)	
17	Interest earned on cash and marketable securities		1	3	9	21	40	
18	Depreciation		(211)	(233)	(257)	(284)	(314)	
19	Profit before tax		333	428	539	669	822	
20	Taxes		(133)	(171)	(216)	(268)	(329)	
21	Profit after tax		200	257	323	401	493	
22	Dividends		0	0	0	0	0	
23	Retained earnings		200	257	323	401	493	
24								
25	**Balance sheet**							
26	Cash and marketable securities	0	19	64	173	359	633	<-- =G38-G27-G31
27	Current assets	200	173	198	228	262	302	
28	Fixed assets							
29	At cost	2,000	2,211	2,443	2,700	2,985	3,299	
30	Depreciation	0	(211)	(443)	(700)	(985)	(1,299)	<-- =F30-$B$6*(G29+F29)/2
31	Net fixed assets	2,000	2,000	2,000	2,000	2,000	2,000	<-- NFA don't change
32	**Total assets**	2,200	2,192	2,262	2,401	2,621	2,935	
33								
34	Current liabilities	100	92	106	122	140	161	
35	Debt	1,000	800	600	400	200	0	<-- =F35-$B$35/5
36	Stock	1,100	1,100	1,100	1,100	1,100	1,100	
37	Accumulated retained earnings	0	200	456	780	1,181	1,674	
38	**Total liabilities and equity**	2,200	2,192	2,262	2,401	2,621	2,935	

The debt repayment terms are incorporated into the model by simply specifying the debt balances at the end of each year. Since the firm is assumed to issue no new equity (in accordance with the covenants on the lending), it follows that the model's plug cannot be on the liabilities side of the balance sheet. In our model the plug is the cash and marketable securities account.

The model incorporates one other assumption often made about fixed assets: It assumes that the *net* fixed assets stay constant over the life of the project. Essentially this means that the depreciation accurately reflects the capital maintenance of the fixed assets. As you can see from looking at rows 29–31 above, this means that the fixed assets at cost grow each year by the increase in asset depreciation. It also means that there is no net cash flow from depreciation:

	A	B	C	D	E	F	G	H
41	**FREE CASH FLOW CALCULATION**							
42	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
43	Profit after tax		200	257	323	401	493	
44	Add back depreciation		211	233	257	284	314	
45	Subtract increase in current assets		28	(26)	(30)	(34)	(39)	
46	Add back increase in current liabilities		(8)	14	16	18	21	
47	Subtract increase in fixed assets at cost		(211)	(233)	(257)	(284)	(314)	
48	Add back after-tax interest on debt		54	42	30	18	6	
49	Subtract after-tax interest on cash and mkt. securities		(0)	(2)	(6)	(13)	(24)	
50	**Free cash flow**		273	285	334	391	457	

Cash flow generated by depreciation equals capital expenditures.

In this example, the firm has no problem in making its debt principal repayments. As credit analysts, we might be interested in how the firm's ability to meet its payments is affected by the various parameter values. In the following example we have increased the ratio of COGS/sales. With the new parameter values, the firm can no longer meet its debt repayments in years 1–3. This fact can be seen in the pro forma: In years 1–4 the balances of cash and marketable securities are negative, indicating that—in order to make the repayment of the loan principal—the firm had to borrow money.[8]

8. From the point of view of corporate finance, positive balances of cash are like *negative balances* of debt. Thus, when the cash is negative, it is equivalent to the firm having borrowed money.

	A	B	C	D	E	F	G	H
1	**PROJECT FINANCE** **With these parameters the project cannot pay off its debt**							
2	Sales growth	15%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Costs of goods sold/Sales	55%						
6	Depreciation rate	10%						
7	Interest rate on debt	10.00%						
8	Interest paid on cash and marketable securities	8.00%						
9	Tax rate	40%						
10	Dividend payout ratio	0%	<-- No dividends until all the debt is paid off					
11								
12	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
13	**Income statement**							
14	Sales		1,150	1,323	1,521	1,749	2,011	
15	Costs of goods sold		(633)	(727)	(836)	(962)	(1,106)	
16	Interest payments on debt		(90)	(70)	(50)	(30)	(10)	
17	Interest earned on cash and marketable securities		(2)	(6)	(7)	(4)	4	
18	Depreciation		(211)	(233)	(257)	(284)	(314)	
19	Profit before tax		215	287	370	469	585	
20	Taxes		(86)	(115)	(148)	(187)	(234)	
21	Profit after tax		129	172	222	281	351	
22	Dividends		0	0	0	0	0	
23	Retained earnings		129	172	222	281	351	
24								
25	**Balance sheet**							
26	Cash and marketable securities	0	(52)	(92)	(83)	(18)	114	<-- =G38-G27-G31, the plug
27	Current assets	200	173	198	228	262	302	
28	Fixed assets							
29	At cost	2,000	2,211	2,443	2,700	2,985	3,299	
30	Depreciation	0	(211)	(443)	(700)	(985)	(1,299)	
31	Net fixed assets	2,000	2,000	2,000	2,000	2,000	2,000	<-- NFA don't change
32	**Total assets**	2,200	2,121	2,107	2,145	2,244	2,416	
33								
34	Current liabilities	100	92	106	122	140	161	
35	Debt	1,000	800	600	400	200	0	
36	Stock	1,100	1,100	1,100	1,100	1,100	1,100	
37	Accumulated retained earnings	0	129	301	523	804	1,155	
38	**Total liabilities and equity**	2,200	2,121	2,107	2,145	2,244	2,416	

5.11 Calculating the Return on Equity

We can use the pro forma models illustrated in this chapter to compute the anticipated return on equity. Look at the previous example: Equity owners in the project have to pay 1,100 in year 0. During years 1–4 they get no payoffs, but in year 5 they own the company. Suppose that the book value of the assets accurately reflects the market value. Then at the end of year 5 the equity in the firm is worth stock + accumulated retained earnings = 2,255. The return on the equity investment (ROE) is calculated as follows:

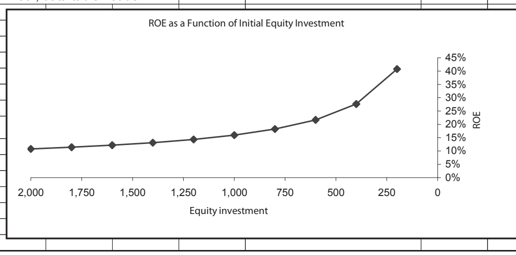
	A	B	C	D	E	F	G	H
56	**RETURN ON EQUITY (ROE)**							
57	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
58	Equity cash flow	-1,100	-	-	-	-	2,255	<-- =G22+G36+G37
59	RETURN ON EQUITY (ROE)	15.44%	<-- =IRR(B58:G58)					

Note that this equity return increases as the equity investment decreases.[9] Consider the case where the firm initially borrows 1,500 and the equity owners invest 600:

	A	B	C	D	E	F	G	H
56	RETURN ON EQUITY (ROE)							
57	Year	0	1	2	3	4	5	
58	Equity cash flow	-600	-	-	-	-	1,602	<-- =G22+G36+G37
59	RETURN ON EQUITY (ROE)	21.70%	<-- =IRR(B58:G58)					

As the following data table and graph show, the less the initial equity investment, the greater the equity return:

	A	B	C	D	E	F	G	H	I	J
61										
62	Data table: ROE as a function of initial		21.70%	<-- =B59 , data table header						
63	equity investment	2,000	10.80%							
64		1,800	11.43%							
65		1,600	12.19%							
66		1,400	13.14%							
67		1,200	14.36%							
68		1,000	15.98%							
69		800	18.26%							
70		600	21.70%							
71		400	27.61%							
72		200	40.76%							
73										
74										
75										
76										
77										



The ROE in Our First Full Model

The model in sections 5.3–5.5 has annual dividends. If we use the mid-year discounting explained in section 5.5 to value the firm, we can compute the return on equity (ROE) of an investor who purchases the firm at date 0 at its imputed equity valuation, gets 5 years of dividends, and sells it for the imputed terminal value of the equity:

9. Interesting but not surprising: As the equity investment goes down, the project becomes more leveraged and hence more risky for the equity investors. The increased return should compensate the equity holders for this extra risk. The really interesting question (not answered here) is whether the increased return is in fact a compensation for the riskiness.

	A	B	C	D	E	F	G	H
1	**COMPUTING THE ROE IN THE FIRST FINANCIAL MODEL**							
2	Sales growth	10%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Net fixed assets/Sales	77%						
6	Costs of goods sold/Sales	50%						
7	Depreciation rate	10%						
8	Interest rate on debt	10.00%						
9	Interest paid on cash and marketable securities	8.00%						
10	Tax rate	40%						
11	Dividend payout ratio	40%						
12								
13	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
14	**Income statement**							
15	Sales	1,000	1,100	1,210	1,331	1,464	1,611	
51								
52								
53	**Valuing the firm (mid-year discounting)**							
54	Weighted average cost of capital	20%						
55	Long-term free cash flow growth rate	5%						
56								
57	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
58	FCF		176	188	201	214	228	
59	Terminal value						1,598	<-- =G58*(1+B55)/(B54-B55)
60	Total		176	188	201	214	1,826	
61								
62	Enterprise value, NPV of row 60	1,348	<-- =NPV(B54,C60:G60)*(1+B54)^0.5					
63	Add in initial (year 0) cash and mkt. securities	80	<-- =B27					
64	Asset value, year 0	1,428	<-- =B63+B62					
65	Subtract out value of firm's debt today	(320)	<-- =-B36					
66	Equity value	1,108	<-- =B64+B65					
67								
68								
69	**RETURN ON EQUITY (ROE)**							
70	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
71	Projected dividends	(1,108)	98	108	118	129	141	
72	Anticipated equity value, year 5						1,737	<-- Terminal value + year 5 cash - year 5 debt
73	Equity cash flow	(1,108)	98	108	118	129	1,878	<-- =SUM(G71:G72)
74	RETURN ON EQUITY (ROE)	18.29%	<-- =IRR(B73:G73)					

5.12 Tax Loss Carryforwards

Firms can apply accumulated losses to reduce their current tax liabilities. In this section we show how to model such tax loss carryforwards in our pro forma model. We make a number of changes in our basic model, highlighted below:

- We assume that in addition to costs of goods sold that are a proportion of sales, the firm also has an annual fixed cost component of sales (in the Excel clip below, this number is 440 in cell B6).
- We assume that in year 0 the firm has an initial accumulated tax loss of 100 (cell B11).

	A	B	C	D	E	F	G	H
1	**MODELING TAX LOSS CARRYFORWARDS**							
2	Sales growth	11%						
3	Current assets/Sales	15%						
4	Current liabilities/Sales	8%						
5	Net fixed assets/Sales	45%						
6	Fixed costs	440						
7	Costs of goods sold/Sales	55%						
8	Depreciation rate	10%						
9	Interest rate on debt	10.00%						
10	Interest paid on cash and marketable securities	8.00%						
11	Initial tax-loss carryforward	-100						
12	Tax rate	40%						
13	Dividend payout ratio	40%						
14								
15	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
16	**Income statement**							
17	Sales	1,000	1,110	1,232	1,368	1,518	1,685	
18	Fixed costs		(440)	(440)	(440)	(440)	(440)	<-- =-$B$6
19	Costs of goods sold		(611)	(678)	(752)	(835)	(927)	
20	Interest payments on debt		3	9	15	21	27	
21	Interest earned on cash and marketable securities		1	(8)	(14)	(16)	(17)	<-- =$B$10*AVERAGE(F34:G34)
22	Depreciation		(82)	(96)	(112)	(130)	(152)	
23	Profit before tax		(18)	20	65	118	177	
24	Taxes		-	-	-	(34)	(71)	<-- =-MAX($B$12*G30,0)
25	Profit after tax		(18)	20	65	84	106	
26	Dividends		-	8	26	34	42	<-- =IF(G25<0,0,$B$13*G25)
27	Retained earnings		(18)	27	91	118	149	
28								
29	Tax loss carryforward from previous years		(100)	(118)	(99)	(34)	-	<-- =MIN(F29+F23,0)
30	Taxable profit	0	0	0	0	84	177	<-- =MAX(G23+G29,0)
31	Effective tax rate		0%	0%	0%	29%	40%	<-- =-G24/G23

In each year we model the accumulated tax loss (row 29). If the previous year exhibited a loss (for example, year 1, with a loss of 18), then the accumulated loss increased by that amount (cell D29). If the previous year exhibits a gain (year 2), then the accumulated tax loss carryforward gets closer to zero (cell D30). At some point (year 5 in our example), tax losses may be completely used up.

In any year, the taxable profits are given in row 30. If the accumulated tax loss carryforward is greater than that year's profit, then there is no tax. Otherwise only the difference in income is taxed (consider year 4, for example). These differences are reflected in the effective tax rate in row 31.

The effective tax rate is used in computing the free cash flows, as shown below:

	A	B	C	D	E	F	G	H
48	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	
49	**Free cash flow calculation**							
50	Profit after tax		(18)	20	65	84	106	
51	Add back depreciation		82	96	112	130	152	
52	Subtract increase in current assets		(17)	(18)	(20)	(23)	(25)	
53	Add back increase in current liabilities		9	10	11	12	13	
54	Subtract increase in fixed assets at cost		(131)	(151)	(173)	(198)	(227)	
55	Add back after-tax interest on debt		(3)	(9)	(15)	(21)	(27)	<-- =-(1-$C$31)*G20
56	Subtract after-tax interest on cash and mkt. securities		(1)	8	14	16	17	<-- =-(1-$C$31)*G21
57	**Free cash flow**		(79)	(45)	(7)	1	9	

5.13 Summary

Pro forma modeling is one of the basic skills of corporate financial analysis, a devious combination of finance, the implementation of accounting rules, and spreadsheet skills. In order to be useful, financial models must match the situation at hand, but they must also be simple enough so that the user can easily understand *why* the results happen (be they valuations, credit-worthiness, or simply commonsense predictions of how a firm or project might look several years down the road).

Exercises

1. Here's a basic exercise that will help you understand what's going on in the modeling of financial statements. Replicate the model in section 5.1. That is, enter the correct formulas for the cells and see that you get the same results as the book. (This turns out to be more of an exercise in accounting than in finance. If you're like many financial modelers, you'll see that there are some aspects of accounting that you've forgotten!)

2. The model of section 5.1 includes costs of goods sold but not selling, general, and administrative (SG&A) expenses. Suppose that the firm has $200 of these expenses each year, irrespective of the level of sales.

   a. Change the model to accommodate this new assumption. Show the resulting profit and loss statements, balance sheets, the free cash flows, and the valuation.

   b. Do a data table in which you show the sensitivity of the equity value to the level of SG&A. Let SG&A vary from $0 per year to $600 per year.

3. Suppose that in the model of section 5.1 the fixed assets *at cost* for years 1–5 are 100% of sales (in the current model, it is *net* fixed assets which are a function of sales). Change the model accordingly. Show the resulting profit and loss statements, balance sheets, and free cash flows for years 1–5. (Assume that in year 0, the fixed assets accounts are as shown in section 5.2. Note that since year 0 is given—it is the current situation of the firm, whereas years 1–5 are the predictions for the future—there is no need for the year 0 ratios to conform to the predicted ratios for years 1–5.)

4. Back to the basic model of section 5.1. Suppose that the fixed assets at cost follow the following step function:

$$Fixed\ Assets\ at\ Cost = \begin{cases} 100\% * Sales & \text{if } Sales \le 1{,}200 \\ 1{,}200 + 90\% * (Sales - 1{,}200) & 1{,}200 < Sales \le 1{,}400 \\ 1{,}380 + 80\% * (Sales - 1{,}400) & Sales > 1{,}400 \end{cases}$$

   Incorporate this function into the model.

5. Consider the model in section 5.6 (where debt is the plug).

   a. Suppose that the firm has 1,000 shares and that it decides to pay, in year 1, a dividend per share of 15 cents. In addition, suppose that it wants this dividend per share to grow in subsequent years by 12% per year. Incorporate these changes into the pro forma model.

b. Do a sensitivity analysis in which you show the effect on the debt/equity ratio of the annual growth rate of dividends. Vary this rate from 0% to 18%, in steps of 2%. For this exercise, define debt as net debt (i.e., debt – cash and marketable securities). (Note that since the WACC = 20%, the growth rate must be less than 20%.)

6. In the model of section 5.6, assume that the firm needs to have minimum cash balances of 25 at the end of each year. Introduce this constraint into the model.

7. In the valuation exercise of section 5.4, the terminal value is calculated using a Gordon dividend model on the cash flows. Replace this terminal value by the year 5 book value of debt + equity. This means that you are essentially assuming that the book value correctly predicts the market value.

8. Repeat the above exercise, but this time replace the terminal value by an EBITDA (earnings before interest, taxes, depreciation, and amortization) ratio times year-5 anticipated EBITDA. Show a graph of the *equity value* of the firm as a function of the assumed year 5 EBITDA ratio, varying this ratio from 6–14.

9. In the project finance pro forma of section 5.8 it is assumed that the firm pays off its initial debt of 1,000 in equal installments of principal over 5 years. Change this assumption and assume instead that the firm pays off its debt in equal payments of interest and principal over 5 years.

   Hint: This means that you have to use the **PMT** function to find the annual payments, then set up a loan table (as in Chapter 1) to split the annual payments into an interest and repayment of principal. Alternatively you can use the functions **PPMT** and **IPMT** discussed in Chapter 34.

10. This problem introduces the concept of "sustainable dividends": The firm whose financials are illustrated below wishes to maintain cash balances of 80 over the next 5 years. It also desires to issue neither additional stock nor make any changes in its current level of debt. This means that dividends are the plug in the balance sheet. Model this situation (note that for some parameter levels you may get "negative dividends," indicating that there is no sustainable level of dividends).

	A	B	C	D	E	F	G
1	**SUSTAINABLE DIVIDENDS--Template**						
2	Sales growth	10%					
3	Current assets/Sales	15%					
4	Current liabilities/Sales	8%					
5	Net fixed assets/Sales	77%					
6	Costs of goods sold/Sales	50%					
7	Depreciation rate	10%					
8	Interest rate on debt	10.00%					
9	Interest paid on cash & marketable securities	8.00%					
10	Tax rate	40%					
11							
12	**Year**	**0**	**1**	**2**	**3**	**4**	**5**
13	**Income statement**						
14	Sales	1,000					
15	Costs of goods sold	(500)					
16	Interest payments on debt	(32)					
17	Interest earned on cash & marketable securities	6					
18	Depreciation	(100)					
19	Profit before tax	374					
20	Taxes	(150)					
21	Profit after tax	225					
22	Dividends	(90)					
23	Retained earnings	135					
24							
25	**Balance sheet**						
26	Cash and marketable securities	80	80	80	80	80	80
27	Current assets	150					
28	Fixed assets						
29	At cost	1,070					
30	Depreciation	(300)					
31	Net fixed assets	770					
32	**Total assets**	1,000					
33							
34	Current liabilities	80					
35	Debt	320	320	320	320	320	320
36	Stock	450	450	450	450	450	450
37	Accumulated retained earnings	150					
38	**Total liabilities and equity**	1,000					

6 Building a Pro Forma Model: The Case of Caterpillar

6.1 Overview

In this chapter we implement the pro forma modeling techniques discussed in Chapter 5 to build a financial model for Caterpillar, Inc. Caterpillar, as most readers of this book will surely know, is a world leader in the manufacture and sales of large-scale earth-moving equipment.

Our energies in this chapter will be to understand Caterpillar's financial statements for the years 2007–2011 in order to fit them into the format illustrated in Chapter 5. This is not a trivial task, and it requires a combination of understanding of CAT's business, modeling skills, and a small dose of financial deviousness (otherwise we would never get to the end of the exercise).

Warning: This Chapter May Be Deleterious to Your Mental Health[1]

The material in this chapter is *irritating* and *complicated* but *not difficult*. Why irritating? Because pro forma financial models require a lot of assumptions and analysis. Everything is related to almost everything else. And because they require you to recall some basic accounting concepts. This is all irritating!

However, the case in this chapter illustrates one of the most important corporate finance applications: The valuation of a company in the framework of its accounting and financial parameters. Such valuations are the core of most business plans, corporate financial planning models, and (intelligent) analyst valuations.

1. It certainly affected the author's mental stability …

6.2 Caterpillar's Financial Statements, 2007–2011

Caterpillar's financial statements for the five years 2007–2011 are given below:

	A	B	C	D	E	F
1	**CATERPILLAR PROFIT AND LOSS, 2007-2011**					
2	Sales and revenues	**2007**	**2008**	**2009**	**2010**	**2011**
3	Sales of Machinery and Power Systems	41,962	48,044	29,540	39,867	57,392
4	Revenues of Financial Products	2,996	3,280	2,856	2,721	2,746
5	Total sales and revenues	44,958	51,324	32,396	42,588	60,138
6						
7	Operating costs:					
8	Cost of goods sold	32,626	38,415	23,886	30,367	43,578
9	Selling, general and administrative expenses	3,821	4,399	3,645	4,248	5,203
10	Research and development expenses	1,404	1,728	1,421	1,905	2,297
11	Interest expense of Financial Products	1,132	1,153	1,045	914	826
12	Deferred income taxes	1,054	1,181	1,822	1,191	1,081
13	Total operating costs	40,037	46,876	31,819	38,625	52,985
14						
15	Operating profit	4,921	4,448	577	3,963	7,153
16						
17	Interest expense excluding financial products	288	274	389	343	396
18	Other income (expense)	357	327	381	130	-32
19						
20	Consolidated profit before taxes	4,990	4,501	569	3,750	6,725
21						
22	Provision (benefit) for income taxes	1,485	953	-270	968	1,720
23	Profit of consolidated companies	3,505	3,548	839	2,782	5,005
24						
25	Equity in profit (loss) of unconsolidated affiliated companies	73	37	-12	-24	-24
26						
27	Profit of consolidated and affiliated companies	3,578	3,585	827	2,758	4,981
28	Less: Profit (loss) attributable to noncontrolling interests	37	28	-68	58	53
29						
30	**Profit**	**3,541**	**3,557**	**895**	**2,700**	**4,928**

	A	B	C	D	E	F
1	**CATERPILLAR BALANCE SHEETS, 2007-2011**					
2	**Assets**	**2007**	**2008**	**2009**	**2010**	**2011**
3	Current assets					
4	Cash and short-term investments	1,122	2,736	4,867	3,592	3,057
5	Receivables--trade and other	8,249	9,397	5,611	8,494	10,285
6	Receivables--finance	7,503	8,731	8,301	8,298	7,668
7	Deferred and refundable income taxes	816	1,223	1,216	931	1,580
8	Prepaid expenses and other current assets	583	765	862	908	994
9	Inventories	7,204	8,781	6,360	9,587	14,544
10	Total current assets	25,477	31,633	27,217	31,810	38,128
11						
12	Property, plant and equipment--net	9,997	12,524	12,386	12,539	14,395
13	Long-term receivables--trade and other	685	1,479	971	793	1,130
14	Long-term receivables--finance	13,462	14,264	12,279	11,264	11,948
15	Investments in unconsolidated affiliated companies	598	94	105	164	133
16	Noncurrent deferred and refundable income taxes	1,553	3,311	2,714	2,493	2,157
17	Intangible assets	475	511	465	805	4,368
18	Goodwill	1,963	2,261	2,269	2,614	7,080
19	Other assets	1,922	1,705	1,632	1,538	2,107
20	**Total assets**	**56,132**	**67,782**	**60,038**	**64,020**	**81,446**
21						
22	**Liabilities**	**2007**	**2008**	**2009**	**2010**	**2011**
23	Current liabilities					
24	Short-term borrowings					
25	Machine and power systems	187	1,632	433	204	93
26	Financial products	5,281	5,577	3,650	3,852	3,895
27	Accounts payable	4,723	4,827	2,993	5,856	8,161
28	Accrued expenses	3,178	4,121	2,641	2,880	3,386
29	Accrued wages, salaries and employee benefits	1,126	1,242	797	1,670	2,410
30	Customer advances	1,442	1,898	1,217	1,831	2,691
31	Dividends payable	225	253	262	281	298
32	Other current liabilities	951	1,027	1,281	1,521	1,967
33	Long-term debt due within one year					
34	Machinery and power systems	180	456	302	495	558
35	Financial products	4,952	5,036	5,399	3,430	5,102
36	Total current liabilities	22,245	26,069	18,975	22,020	28,561
37						
38	Long-term debt due after one year					
39	Machinery and power systems	3,639	5,736	5,652	4,505	8,415
40	Financial products	14,190	17,098	16,195	15,932	16,529
41	Liability for postemployment benefits	5,059	9,975	7,420	7,584	10,956
42	Other liabilities	2,003	2,190	2,496	2,654	3,583
43	Total liabilities	47,136	61,068	50,738	52,695	68,044
44						
45	Redeemable noncontrolling interest	0	524	477	461	473
46	Stockholders' equity					
47	Common stock of $1.00 par					
48	Common stock (814,894,624 shares)	2,744	3,057	3,439	3,888	4,273
49	Treasury stock at cost	-9,451	-11,217	-10,646	-10,397	-10,281
50	Profit employed in the business	17,398	19,826	19,711	21,384	25,219
51	Accumulated and other comprehensive income (loss)	-1,808	-5,579	-3,764	-4,051	-6,328
52	Noncontrolling interests	113	103	83	40	46
53	Total shareholders equity	8,996	6,190	8,823	10,864	12,929
54	**Total liabilities and equity**	**56,132**	**67,782**	**60,038**	**64,020**	**81,446**
55						
56						
57	Treasury stock, shares	190,908,490	213,267,983	190,171,905	176,071,910	167,361,280
58	Net shares outstanding	623,986,134	601,626,641	624,722,719	638,822,714	647,533,344

	A	B	C	D	E	F
1	**CATERPILLAR, CONSOLIDATED STATEMENT OF CASH FLOWS, 2007-2011**					
2	**Cash flow from operating activities**	**2007**	**2008**	**2009**	**2010**	**2011**
3	Profit of consolidated and affiliated companies	3,578	3,585	827	2,758	4,981
4	Adjustments for non-cash items:					
5	Depreciation and amortization	1,797	1,980	2,336	2,296	2,527
6	Other	162	355	137	469	457
7	Changes in assets and liabilities, net of acquisitions and divestitures:					
8	Receivables—trade and other	899	-545	4,014	-2,320	-1,345
9	Inventories	-745	-833	2,501	-2,667	-2,927
10	Accounts payable	387	-4	-2,034	2,570	1,555
11	Accrued expenses	231	660	-505	117	308
12	Accrued wages, salaries and employee benefits	576	286	-534	847	619
13	Deferred income taxes	66	-470	-646	604	173
14	Other assets, net	1,004	-217	235	358	-91
15	Other liabilities, net			12	-23	753
16	**Net cash provided by (used for) operating activities**	**7,955**	**4,797**	**6,343**	**5,009**	**7,010**
17						
18	**Cash flow from investing activities**					
19	Capital expenditures—excluding equipment leased to others	-1,700	-2,445	-1,348	-1,575	-2,515
20	Expenditures for equipment leased to others	-1,340	-1,566	-968	-1,011	-1,409
21	Proceeds from disposals of leased assets and property, plant and equipment	408	982	1,242	1,469	1,354
22	Additions to finance receivables	-13,946	-14,031	-7,107	-8,498	-10,001
23	Collections of finance receivables	10,985	9,717	9,288	8,987	8,874
24	Proceeds from sale of finance receivables	866	949	100	16	207
25	Investments and acquisitions (net of cash acquired)	-229	-117	-19	-1,126	-8,184
26	Proceeds from sale of businesses and investments (net of c	290	0	0	0	376
27	Proceeds from sale of available-for-sale securities	282	357	291	228	247
28	Investments in available-for-sale securities	-485	-339	-349	-217	-336
29	Other, net	461	197	-128	132	-40
30	**Net cash provided by (used for) investing activities**	**-4,408**	**-6,296**	**1,002**	**-1,595**	**-11,427**
31						
32	**Cash flow from financing activities**					
33	Dividends paid	-845	-953	-1,029	-1,084	-1,159
34	Distribution to noncontrolling interests	-20	-10	-10	0	-3
35	Common stock issued, including treasury shares reissued	328	135	89	296	123
36	Payment for stock repurchase derivative contracts	-56	-38			
37	Treasury shares purchased	-2,405	-1,800			
38	Excess tax benefit from stock-based compensation	155	56	21	153	189
39	Acquisitions of noncontrolling interests	0	0	-6	-132	-8
40	Proceeds from debt issued (original maturities greater than three months):					
41	- Machinery and Power Systems	224	1,673	458	216	4,587
42	- Financial Products	10,815	16,257	11,833	8,108	10,873
43	Payments on debt (original maturities greater than three months):					
44	- Machinery and Power Systems	-598	-296	-918	-1,298	-2,269
45	- Financial Products	-10,290	-14,143	-11,769	-11,163	-8,324
46	Short-term borrowings, net (original maturities three months or less)	-297	2,074	-3,884	291	-43
47	**Net cash provided by (used for) financing activities**	**-2,989**	**2,955**	**-5,215**	**-4,613**	**3,966**
48						
49	Interest expense excluding financial products	288	274	389	343	396
50	Tax rate	29.76%	21.17%	-47.45%	25.81%	25.58%

Rewriting the Balance Sheet

To more easily understand the balance sheet, we rewrite the balance sheets to combine the short- and long-term financial debt items. We also write all the operating (i.e., non-financial) current assets and operating current liabilities as one item.

	A	B	C	D	E	F
1	**CATERPILLAR BALANCE SHEETS, 2007-2011** **Rewritten: Operating Current Assets, Operating Current Liabilities, Financial Debts together**					
2	**Assets**	**2007**	**2008**	**2009**	**2010**	**2011**
3	Current assets					
4	Cash and short-term investments	1,122	2,736	4,867	3,592	3,057
5	Operating current assets	24,355	28,897	22,350	28,218	35,071
6						
7	Property, plant and equipment--net	9,997	12,524	12,386	12,539	14,395
8	Long-term receivables	14,147	15,743	13,250	12,057	13,078
9	Investments in unconsolidated affiliated companies	598	94	105	164	133
10	Noncurrent deferred and refundable income taxes	1,553	3,311	2,714	2,493	2,157
11	Intangible assets	475	511	465	805	4,368
12	Goodwill	1,963	2,261	2,269	2,614	7,080
13	Other assets	1,922	1,705	1,632	1,538	2,107
14	**Total assets**	**56,132**	**67,782**	**60,038**	**64,020**	**81,446**
15						
16	**Liabilities**	**2007**	**2008**	**2009**	**2010**	**2011**
17	Operating current liabilities	11,645	13,368	9,191	14,039	18,913
18						
19	Financial debt	28,429	35,535	31,631	28,418	34,592
20						
21	Liability for postemployment benefits	5,059	9,975	7,420	7,584	10,956
22	Other liabilities	2,003	2,190	2,496	2,654	3,583
23	Total liabilities	47,136	61,068	50,738	52,695	68,044
24						
25	Redeemable noncontrolling interest	0	524	477	461	473
26	Stockholders' equity					
27	Common stock (814,894,624 shares)	2,744	3,057	3,439	3,888	4,273
28	Treasury stock at cost	-9,451	-11,217	-10,646	-10,397	-10,281
29	Profit employed in the business	17,398	19,826	19,711	21,384	25,219
30	Accumulated and other comprehensive income (loss)	-1,808	-5,579	-3,764	-4,051	-6,328
31	Noncontrolling interests	113	103	83	40	46
32	Total shareholders equity	8,996	6,190	8,823	10,864	12,929
33	**Total liabilities and equity**	**56,132**	**67,782**	**60,038**	**64,020**	**81,446**

6.3 Analyzing the Financial Statements

We now build a set of financial statements with the technology of Chapter 5 but in roughly the same format as Caterpillar's historical statements. Our model is forward looking but based on CAT's historical financial statements. Our choices of model parameters is based on our analysis of the historicals and on a number of judgment calls.

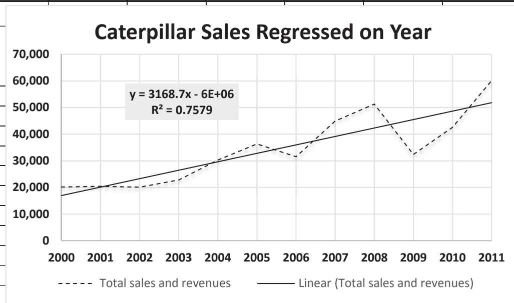
Sales Projections

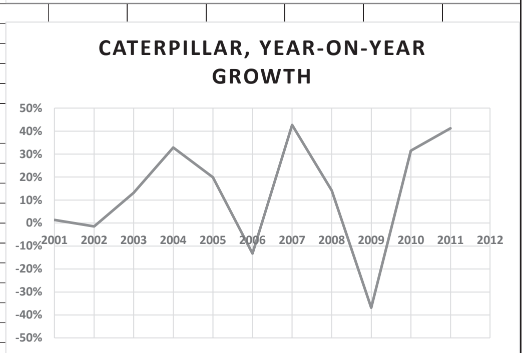
The sales projection is one of the most critical elements in our model. Should it be based on a historical analysis of Caterpillar sales? Or should it be based on expert opinions about the potential growth of the heavy-equipment industry? Should we differentiate between short-term (next 5 years) growth and long-term growth?

None of these questions has a satisfactory answer. We illustrate several methods and suggest that you use the pro forma model to experiment with the various answers to see whether they have a value impact. In the Excel below we show:

- Year-on-year sales growth
- Cumulative average growth rates (CAGR)
- Regression of sales on the year

	A	B	C	D	E
1	ANALYSIS OF CATERPILLAR SALES				
2	Year	Sales of machinery	Revenues of financial products	Total sales and revenues	Year-on-year growth
3	2000	18,913	1,262	20,175	
4	2001	19,027	1,423	20,450	1.36%
5	2002	18,648	1,504	20,152	-1.46%
6	2003	21,048	1,759	22,807	13.17%
7	2004	28,336	1,970	30,306	32.88%
8	2005	34,006	2,333	36,339	19.91%
9	2006	28,869	2,648	31,517	-13.27%
10	2007	41,962	2,996	44,958	42.65%
11	2008	48,044	3,280	51,324	14.16%
12	2009	29,540	2,856	32,396	-36.88%
13	2010	39,867	2,721	42,588	31.46%
14	2011	57,392	2,746	60,138	41.21%
15					
16	Growth measures				
17	CAGR	10.44%	<-- =(D14/D3)^(1/11)-1		
18	Slope	3,169	<-- =SLOPE(D3:D14,A3:A14)		
19	Slope/2011 sales	5.27%	<-- =B18/D14		
20					
21	Intercept	-6,320,495	<-- =INTERCEPT(D3:D14,A3:A14)		
22	R-squared	0.7579	<-- =RSQ(D3:D14,A3:A14)		
23					
24	CAGR is very affected by end points!				
25	2001-2011	11.39%	<-- =(D14/D4)^(1/10)-1		
26	2002-2011	12.92%	<-- =(D14/D5)^(1/9)-1		
27	2003-2011	12.88%	<-- =(D14/D6)^(1/8)-1		
28	2006-2011	13.79%	<-- =(D14/D9)^(1/5)-1		
29					
30	2001-2010	8.49%	<-- =(D13/D4)^(1/9)-1		
31	2002-2010	9.80%	<-- =(D13/D5)^(1/8)-1		
32	2003-2010	9.33%	<-- =(D13/D6)^(1/7)-1		
33	2005-2010	3.22%	<-- =(D13/D8)^(1/5)-1		





We focus on two numbers in the above Excel:

- The compound annual growth rate (CAGR) in cell B17 is $CAGR = \left(\frac{Sales_{2011}}{Sales_{2000}}\right)^{\left(\frac{1}{11}\right)} - 1 = 10.44\%$. CAGR is often used but because of its dependence on the endpoints, it is a problematic number. We find it a suspect measure of long-term growth, unless verified by other time spans. As the second chart above shows, the year-on-year growth of CAT exhibits a lot of variability. In the Excel above we have played with these end points, and find that the CAGR can vary from 3.22% to 13.79%.
- Regressing the annual sales on the year is not dependent solely on the starting and ending sales, and thus often provides a better estimation of future sales growth:

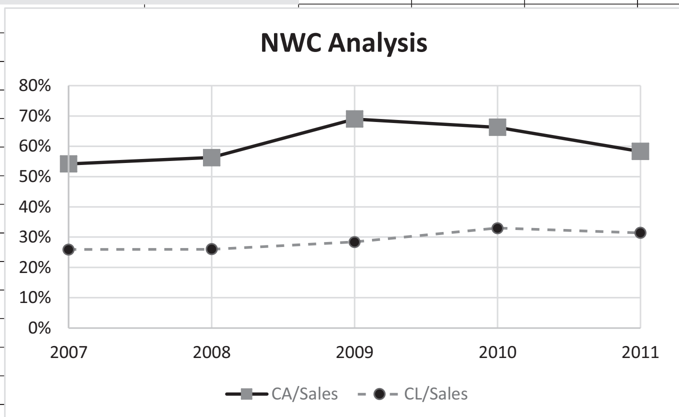
$$Sales_t = a + b * Year_t$$
$$= -6{,}320{,}495 + 3{,}169 * Year,\ R^2 = 75.79\%$$

This regression line suggests that sales will grow at a rate of about 3,169 annually. Impounding this estimate into the current sales (cell B14 above) gives an annual growth rate of about 5.27%. On the whole we find this a more satisfactory estimate of sales growth.[2]

Current Assets and Current Liabilities

The ratio of current assets to sales and current liabilities to sales shows a lot of stability over the 5 years. As model parameter values we have chosen the average of the 5 years.

	A	B	C	D	E	F
1	CATERPILLAR ANALYSIS OF RATIOS					
2		**2007**	**2008**	**2009**	**2010**	**2011**
3	CA/Sales	54.17%	56.30%	68.99%	66.26%	58.32%
4	CL/Sales	25.90%	26.05%	28.37%	32.96%	31.45%
5	Model values					
6	CA/Sales	58.32%	<-- =F3			
7	CL/Sales	31.45%	<-- =F4			
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						

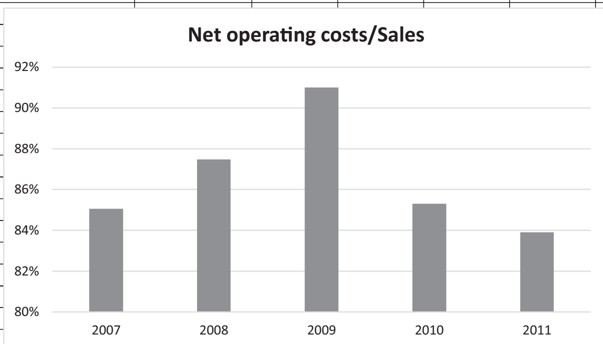


2. Our baseline prejudice is that for a company in an established, old-line business, growth rate should be approximately the economy's real growth rate plus the inflation rate. The 5.27% number falls in the framework of this prejudice.

Operating Costs

Like most industrial companies Caterpillar's operating costs include depreciation. Following the model in Chapter 5, we want to separate depreciation from other operating costs, so we refer to the consolidated statement of cash flows to compute the operating costs without depreciation. Operating costs have varied considerably over the past 5 years:

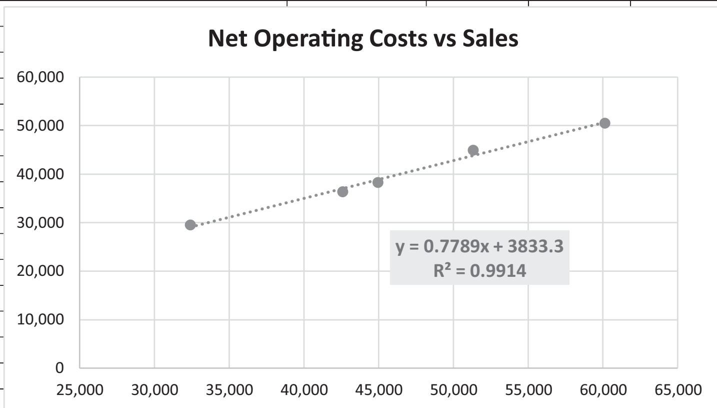
	A	B	C	D	E	F	G
1	**CATERPILLAR OPERATING COSTS**						
2		**2007**	**2008**	**2009**	**2010**	**2011**	
3	Operating costs from P&L	40,037	46,876	31,819	38,625	52,985	<-- ='Caterpillar P&L'!F13
4	Depreciation	1,797	1,980	2,336	2,296	2,527	<-- ='Caterpillar CSCF'!F5
5	Operating costs net of depreciation	38,240	44,896	29,483	36,329	50,458	<-- =F3-F4
6	Sales	44,958	51,324	32,396	42,588	60,138	<-- ='Caterpillar P&L'!F5
7	Net operating costs/Sales	85.1%	87.5%	91.0%	85.3%	83.9%	<-- =F5/F6



We note that the operating cost ratio is an increasing function of sales (shown below). This indicates a large fixed cost component. Modeling this and running a regression, we chose to model the operating costs as a linear function of sales:

$$Costs = 3{,}833 + 0.7789 * Sales$$

	A	B	C	D	E
25	Regressing net operating costs on sales				
26	Intercept	3,833	<-- =INTERCEPT(B5:F5,B6:F6)		
27	Slope	77.89%	<-- =SLOPE(B5:F5,B6:F6)		
28	R-squared	99.14%	<-- =RSQ(B5:F5,B6:F6)		



We use this regression in our pro forma model for Caterpillar.

Fixed Assets and Sales

We need to determine a model for the fixed assets of Caterpillar. This involves two decisions:

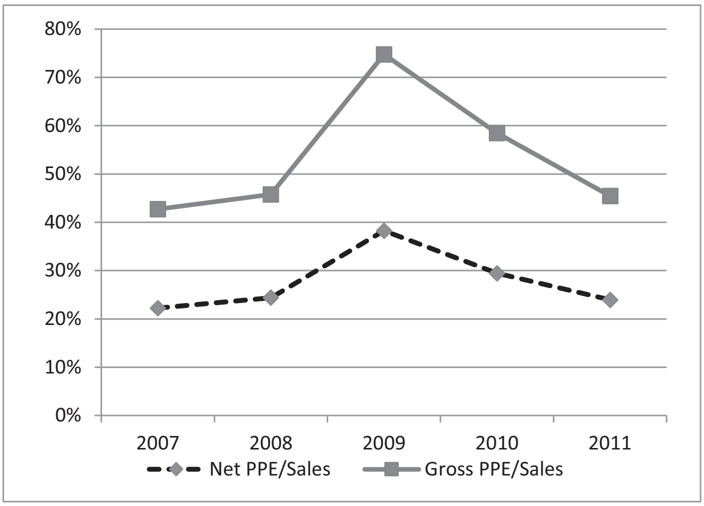
- Are fixed assets best modeled by assuming that the critical model ratio is Net FA/Sales or Gross FA/Sales? We have a slight predilection toward the former, believing it to make more economic sense.[3]
- What is the applicable average depreciation rate?

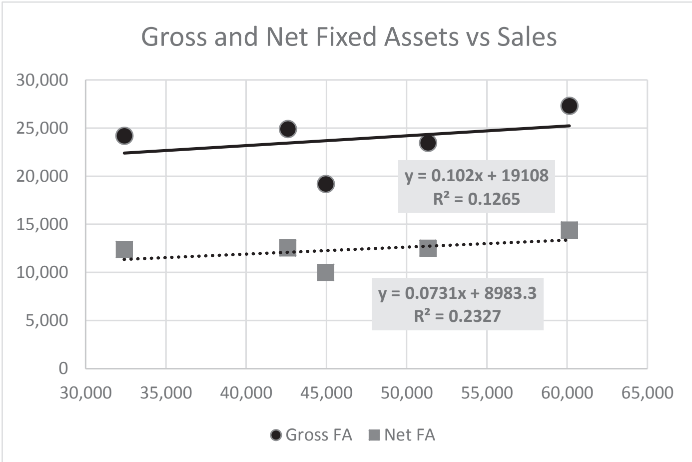
In this subsection we deal with the first question. The depreciation rate is discussed in the next section.

3. If depreciation has any economic meaning, then producing more sales should require more net fixed assets.

	A	B	C	D	E	F
1	**CATERPILLAR—ANALYSIS OF FIXED ASSETS**					
2		**2007**	**2008**	**2009**	**2010**	**2011**
3	Land	189	575	639	682	753
4	Buildings and land improvements	3,625	4,647	4,914	5,174	5,857
5	Machinery, equipment and other	9,756	12,173	12,917	13,414	14,435
6	Equipment leased to others	4,556	4,561	4,717	4,444	4,285
7	Construction-in-process	1,082	1,531	1,034	1,192	1,996
8						
9	Total property, plant and equipment, at cost	19,208	23,487	24,221	24,906	27,326
10	Less: Accumulated depreciation	-9,211	-10,963	-11,835	-12,367	-12,931
11	Property, plant and equipment—net	9,997	12,524	12,386	12,539	14,395
12						
13	Sales	44,958	51,324	32,396	42,588	60,138
14						
15	Net PPE/Sales	22.24%	24.40%	38.23%	29.44%	23.94%
16	Gross PPE/Sales	42.72%	45.76%	74.77%	58.48%	45.44%
17	Model value: Net PPE/Sales	25.00%	<-- =AVERAGE(B15,C15,E15,F15)			

There appears to be a slightly better connection between the Net PPE and Sales than the Gross PPE, and this is accordingly the value we use in our model.





Depreciation

Note 8 of the Caterpillar 2011 annual report gives the approximate lives of different fixed asset classes:

(Millions of dollars)	Useful Lives (Years)	2011	2010	2009
Land	—	**753**	682	639
Buildings and land improvements	20-45	**5,857**	5,174	4,914
Machinery, equipment and other	3-10	**14,435**	13,414	12,917
Equipment leased to others	1-10	**4,285**	4,444	4,717
Construction-in-process	—	**1,996**	1,192	1,034
Total property, plant and equipment, at cost		**27,326**	24,906	24,221
Less: Accumulated depreciation		**(12,931)**	(12,367)	(11,835)
Property, plant and equipment—net		**14,395**	12,539	12,386

In row 28 below we use **SumProduct** to compute the weighted average depreciation rate for CAT. In B29 we average these numbers to compute the model depreciation rate:

	A	B	C	D	E	F	G
19	**Depreciation analysis**						
20	%Buildings and land improvements	20.21%	21.73%	21.79%	22.46%	23.83%	<-- =F4/SUM(F$4:F$6)
21	%Machinery, equipment and other	54.39%	56.93%	57.29%	58.24%	58.73%	<-- =F5/SUM(F$4:F$6)
22	%Equipment leased to others	25.40%	21.33%	20.92%	19.29%	17.44%	<-- =F6/SUM(F$4:F$6)
23							
24	**Depreciation rates**						
25	Buildings and land improvements	3.1%	<-- =1/32.5				
26	Machinery, equipment and other	15.4%	<-- =1/6.5				
27	Equipment leased to others	18.2%	<-- =1/5.5				
28	Average depreciation rate	13.61%	13.31%	13.29%	13.16%	12.94%	<--
29	Model depreciation rate	13.26%	<-- =AVERAGE(B28:F28)				=SUMPRODUCT($B$25:
30							$B$27,F20:F22)

Dividends

Caterpillar total dividends have grown at a compound growth rate of 8.22% per year. We use this number in our model. Our previous objection to CAGR doesn't hold here, since various starting and ending points yield pretty much the same estimates. We prefer to use total dividends as opposed to dividends per share.

	A	B	C	D	E	F
1	**CATERPILLAR DIVIDENDS**					
2	Other liabilities	**2007**	**2008**	**2009**	**2010**	**2011**
3	Dividends	845	953	1,029	1,084	1,159
4	Year-on-year growth		12.78%	7.97%	5.34%	6.92%
5	Dividend CAGR	8.22%	<-- =(F3/B3)^(1/4)-1			
6						
7	Dividends per share	1.38	1.62	1.68	1.74	1.82
8	Dividend CAGR	7.16%	<-- =(F7/B7)^(1/4)-1			

Other Liabilities and Pensions

We are frankly confused by these two items. Why are other liabilities and pension liabilities growing at a rate much faster than sales? We have no good answer, and in our model assume that the growth of these important liabilities will slow down slightly in the future:

	A	B	C	D	E	F
1	**OTHER LIABILITIES AND PENSIONS**					
2		**2007**	**2008**	**2009**	**2010**	**2011**
3	**Other liabilities**	2,003	2,190	2,496	2,654	3,583
4	Year-on-year growth		9.34%	13.97%	6.33%	35.00%
5	Whole period	15.65%	<-- =(F3/B3)^(1/4)-1			
6	Excluding 2011	9.83%	<-- =(E3/B3)^(1/3)-1			
7						
8	**Pension liabilities**	5,059	9,975	7,420	7,584	10,956
9	Year-on-year growth		97.17%	-25.61%	2.21%	44.46%
10	Whole period	21.31%	<-- =(F8/B8)^(1/4)-1			
11	Excluding 2011	14.45%	<-- =(E8/B8)^(1/3)-1			
12	Model value	17.88%	<-- =AVERAGE(B10:B11)			

Caterpillar's Tax Rate

Except for the problematic year 2009, CAT's tax rate averages around 26%. This is the value we use in our model:

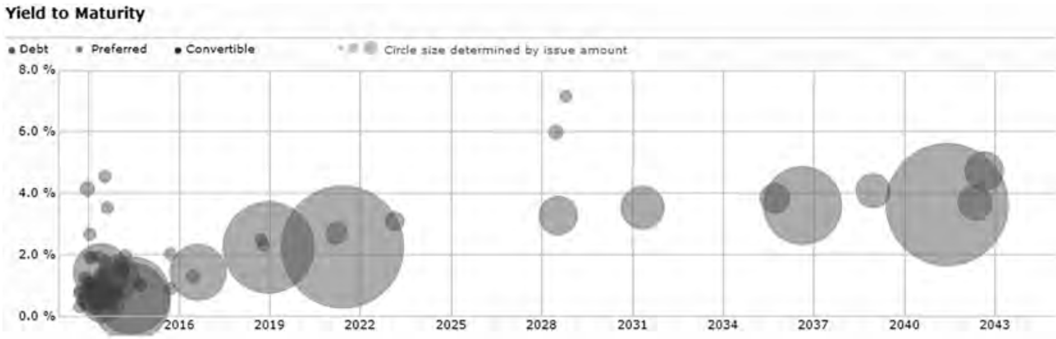
	A	B	C	D	E	F
1	**CATERPILLAR ANALYSIS OF TAX RATE**					
2		**2007**	**2008**	**2009**	**2010**	**2011**
3	Profits before taxes	4,990	4,501	569	3,750	6,725
4	Taxes	1,485	953	-270	968	1,720
5	Tax rate	29.76%	21.17%	-47.45%	25.81%	25.58%
6	Model value	25.58%	<-- =AVERAGEIF(B5:F5,">0")			

Long-Term Receivables

CAT uses long-term receivables as a marketing tool to provide financing to the purchases of its equipment, so that it makes sense that this item grows at the same rate as sales.

Cost of Debt

Caterpillar is a highly rated borrower in financial markets. A chart from Morningstar (http://quicktake.morningstar.com/stocknet/bonds.aspx?symbol=cat) shows the term structure of CAT's debt. We will assume that the company's average cost of debt is 4%.



Interest Earned on Cash and Short-Term Securities

At the time this chapter is written, short-term interest rates are very low. We will assume that CAT earns 1% on its short-term cash investments.

What's the Plug?

The plug should be one of the following financial items: cash and marketable securities, debt, or treasury stock. After some analysis, we use the cash/sales and the debt/assets ratios below. This leaves treasury stock as the plug in our model.

	A	B	C	D	E	F	G
1	**CATERPILLAR: VARIOUS PLUGS vs SALES AND OTHER**						
2		**2007**	**2008**	**2009**	**2010**	**2011**	
3	Sales	44,958	51,324	32,396	42,588	60,138	<-- ='Page 162'!F5
4							
5	Cash	1,122	2,736	4,867	3,592	3,057	<-- ='Page 163'!F4
6	Cash/Sales	2.50%	5.33%	15.02%	8.43%	5.08%	<-- =F5/F3
7	Debt	28,429	35,535	31,631	28,418	34,592	<-- ='Page 165'!F19
8	Debt/Sales	63.23%	69.24%	97.64%	66.73%	57.52%	<-- =F7/F3
9	Debt/Total assets	50.65%	52.43%	52.68%	44.39%	42.47%	<-- =F7/'Page 163'!F20
10							
11	Model values						
12	Cash/Sales	5.08%	<-- =F6				
13	Debt/Assets	42.47%	<-- =F9				

6.4 A Model for Caterpillar

Using the values computed for Caterpillar, we arrive at the following pro forma model for the company:

	A	B	C	D	E	F	G	H
1	**CATERPILLAR PRO FORMA MODEL**							
2	Sales growth	5.27%	<-- ='Page 167'!B19					
3	Current assets/Sales	58.32%	<-- ='Page 168'!B6					
4	Current liabilities/Sales	31.45%	<-- ='Page 168'!B7					
5	Net fixed assets/Sales	27.65%	<-- =AVERAGE('Page 171-173'!B15:F15)					
6	Cost of goods sold/Sales	83.90%	<-- ='Page 169-170'!F7					
7	Long-term receivables, growth	5.27%	<-- Assume this grows at the same rate as sales					
8	Depreciation rate	13.26%	<-- ='Page 171-173'!B29					
9	Pension liabilities, growth	17.88%	<-- ='Page 174'!B12					
10	Other liabilities, growth	9.83%	<-- ='Page 174'!B6					
11	Cash/Sales	5.08%	<-- ='Page 175'!B12					
12	Debt/Assets	42.47%	<-- ='Page 175'!B13					
13	Interest rate on debt	1.41%	<-- ='Ratios (5)'!F18					
14	Interest paid on cash & mkt. sec.	0.00%	<-- Author guesstimate					
15	Tax rate	25.58%	<-- ='Page 174, middle'!B6					
16	Dividend growth	8.22%	<-- ='Page 173'!B6					
17								
18	**Year**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	
19	**Income statement**							
20	Sales	60,138	63,307	66,642	70,154	73,850	77,742	<-- =F20*(1+$B$2)
21	Costs of goods sold	-50,458	-53,117	-55,915	-58,862	-61,963	-65,228	<-- =-$B$6*G20
22	Interest payments on debt		-521	-543	-567	-592	-618	<-- =-$B$13*G50
23	Interest earned on cash & marketable securities		0	0	0	0	0	<-- =$B$14*AVERAGE(F32:G32)
24	Depreciation	-2,527	-3,711	-4,524	-5,301	-6,196	-7,225	<-- =G38-F38
25	Profit before tax	7,153	5,958	5,659	5,424	5,100	4,671	<-- =SUM(G20:G24)
26	Taxes	-1,720	-1,524	-1,448	-1,387	-1,305	-1,195	<-- =-$B$15*G25
27	Profit after tax	5,433	4,434	4,212	4,036	3,795	3,476	<-- =G25+G26
28	Dividends	1,159	1,254	1,357	1,469	1,590	1,720	<-- =F28*(1+$B$16)
29	Retained earnings		5,688	5,569	5,505	5,385	5,196	<-- =G27+G28
30								
31	**Balance sheet**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	<-- 2016
32	Cash and marketable securities	3,057	3,218	3,388	3,566	3,754	3,952	<-- =$B$11*G20
33	Current assets	35,071	36,919	38,864	40,912	43,068	45,337	<-- =$B$3*G20
34	Fixed assets							<--
35	Land	753	753	753	753	753	753	<-- =F35
36	Construction in progress	1,996	1,996	1,996	1,996	1,996	1,996	<-- =F36
37	Depreciable assets at cost	24,577	31,397	36,844	43,117	50,335	58,635	<-- =G39-G38-G36-G35
38	Accumulated depreciation	-12,931	-16,642	-21,167	-26,468	-32,664	-39,889	<-- =F38-$B$8*AVERAGE(F37:G37)
39	Net PPE	14,395	17,504	18,427	19,398	20,420	21,496	<-- =$B$5*G20
40								<--
41	Long-term receivables	13,078	13,767	14,493	15,256	16,060	16,906	<-- =F41*(1+$B$7)
42	Investments in unconsolidated affiliated companies	133	133	133	133	133	133	<-- =F42
43	Noncurrent deferred and refundable income taxes	2,157	2,157	2,157	2,157	2,157	2,157	<-- =F43
44	Intangibles and other	13,555	13,555	13,555	13,555	13,555	13,555	<-- =F44
45	**Total assets**	81,446	87,253	91,016	94,977	99,147	103,536	<-- =G32+G33+G39+SUM(G41:G44)
46								<--
47	Current liabilities	18,913	19,910	20,959	22,063	23,225	24,449	<-- =$B$4*G20
48	Pension liabilities	10,956	12,915	15,224	17,946	21,155	24,937	<-- =F48*(1+$B$9)
49	Other liabilities	3,583	3,935	4,322	4,748	5,214	5,727	<-- =F49*(1+$B$10)
50	Debt	34,592	37,059	38,657	40,339	42,110	43,974	<-- =$B$12*G45
51	Redeemable noncontrolling interest	473	473	473	473	473	473	<-- =F51
52	Stock	2,744	2,744	2,744	2,744	2,744	2,744	<-- =F52
53	Treasury stock	-9,451	-12,868	-20,018	-27,496	-35,320	-43,511	<-- =G45-SUM(G47:G52)-G54
54	Accumulated retained earnings	17,398	23,086	28,655	34,161	39,545	44,742	<-- =F54+G29
55	**Total liabilities and equity**	79,208	87,253	91,016	94,977	99,147	103,536	<-- =SUM(G47:G54)

This model gives the following free cash flow projections:

	A	B	C	D	E	F	G	H
58	**Year**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	
59	**Free Cash Flow**							
60	PAT		4,434	4,212	4,036	3,795	3,476	<-- =G27
61	Add back depreciation		3,711	4,524	5,301	6,196	7,225	<-- =-G24
62	Minus NWC							
63	Minus increase in CA		-1,848	-1,945	-2,048	-2,156	-2,269	<-- =F33-G33
64	Add back increase in CL		997	1,049	1,104	1,163	1,224	<-- =G47-F47
65	Subtract CAPEX		-6,820	-5,447	-6,272	-7,218	-8,301	<-- =-(SUM(G35:G37)-SUM(F35:F37))
66	Add back net interest after taxes		388	404	422	440	460	<-- =-(1-$B$15)*(G22-G23)
67	Free Cash Flow		861	2,797	2,544	2,220	1,814	<-- =SUM(G60:G66)

6.5 Using the Model to Value Caterpillar

In Chapter 3 we discussed the cost of capital for Caterpillar, and came to the conclusion that WACC = 10.84%. Using this number in our model and assuming that the long-term FCF growth is 9% produces the following FCF valuation:

	A	B	C	D	E	F	G	H
70	WACC	10.84%						
71	Long-term FCF growth	9.00%						
72								
73	**Year**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	
74	FCF		861	2,797	2,544	2,220	1,814	<-- =G67
75	Terminal value						107,487	<-- =G74*(1+B71)/(B70-B71)
76	Total		861	2,797	2,544	2,220	109,301	<-- =G74+G75
77								
78	Enterprise value: PV of FCFs and terminal value	75,515	<-- =NPV(B70,C76:G76)*(1+B70)^0.5					
79	Add back initial cash	3,057	<-- =B32					
80	Asset value	78,572	<-- =B78+B79					
81	Subtract year 0 debt	-49,604	<-- =-SUM(B48:B51)					
82	Imputed equity value	28,968	<-- =B80+B81					
83	Divide by # shares outstanding	647,533,344	<-- ='Page 163'!F58					
84	Share value	44.74	<-- =B82/B83*1000000					
85	Current market value per share	90.60						

According to these model values, Caterpillar is currently overvalued.

A **Data Table** on the long-term growth rate and the weighted average cost of capital produces the following table. Cells with a valuation greater than the current market value of 90.60 are highlighted, and where the WACC < long-term growth, the cells have been blanked, since this does not accord with the valuation model (see section 5.4). Note that for some combinations, the model predicts that the value of the equity is negative—meaning that Caterpillar's year 0 debt outweighs the value of its equity.

	A	B	C	D	E	F	G
87			LT growth rate ↓				
88		44.74	**6.0%**	**7.0%**	**8.0%**	**9.0%**	**10.0%**
89	WACC-->	**6.0%**					
90		**6.5%**	389.00				
91		**7.0%**	160.45				
92		**7.5%**	84.25	374.31			
93	=IF(B70>B71,B84,"")	**8.0%**	46.13	153.15			
94		**8.5%**	23.24	79.40	360.22		
95		**9.0%**	7.97	42.51	146.14		
96		**9.5%**	-2.94	20.37	74.75	346.69	
97		**10.0%**	-11.14	5.59	39.04	139.41	
98		**10.5%**	-17.52	-4.97	17.61	70.29	333.71

6.6 Summary

On one hand, the pro forma financial model is an extraordinarily labor-intensive method of valuing the firm. On the other hand, in modeling Caterpillar, we have discovered much about the way the firm operates and about how much we understand (and don't …) about its financial statements.

The pro forma modeling technique cannot be undertaken lightly, but if an intensive investigation into the firm's workings is needed, this is the way to go.

7 Financial Analysis of Leasing

7.1 Overview

A lease is a contractual arrangement by which the owner of an asset (the *lessor*) rents the assets to a *lessee*. In this chapter we analyze long-term leases, in which the asset spends most of its useful life with the lessee. In economic terms, the leases we consider in this chapter are considered by the lessees as alternatives to purchasing an asset. The analysis of this chapter fits many long-term equipment leases, but not short-term leasing (car rentals, for example). Financial theory regards such leases as being essentially debt contracts: For the lessee, the lease is an alternative to purchasing the asset with debt, and the lessor understands that it is essentially providing financing for the lessee.

In the example that follows we consider a company that is faced with the choice of either purchasing or leasing a piece of equipment. We assume that the operating inflows and outflows from the equipment are not affected by its ownership—irrespective of how the asset is held (whether owned or leased), the owner/lessee will have the same sales and must bear the responsibility for maintaining the equipment. In the words of Statement 13 of the Financial Accounting Standards Board (FASB 13), the lease we are considering is one that "transfers substantially all of the benefits and risks incident to the ownership of property" to the lessee.

The analysis in this chapter concentrates exclusively on the *cash flows* from the lease. It is assumed that the lessor pays taxes on the income from the lease rentals and gets a tax shield on the depreciation of the asset, and that the lessee can claim the rent as an expense. The analysis assumes that the tax authorities treat the lessor as the owner of the asset and the lessee as the user.[1]

7.2 A Simple but Misleading Example

The essence of our analysis can be understood from the following simple example: A company has decided to acquire the use of a machine costing $600,000. If purchased, the machine will be depreciated on a straight-line basis to a residual value of zero. The machine's estimated life is 6 years, and the company's tax rate T_C is 40%.

1. From an economic point of view this assumption is not innocuous: If the lessee has true economic ownership of the asset, why doesn't it get to take the depreciation? But the facts are otherwise.

The company's alternative to purchasing the machine is to lease it for 6 years. A lessor has offered to lease the machine to the company for $140,000 annually, with the first payment to be made today and with five additional payments to be made at the start of each of the next 5 years.

One way of analyzing this problem (a misleading way, as it turns out) is to compare the present values of the cash flows to the company of leasing and of buying the asset. The company feels that the lease payment and the tax shield from depreciation are riskless. Suppose, furthermore, that the risk-free rate is 12%. On the basis of the following calculation, the company should lease the asset.[2]

$$NPV(leasing) = \sum_{t=0}^{5} \frac{(1-T_c)*Lease\ rental}{(1+12\%)^t} = \sum_{t=0}^{5} \frac{(1-T_c)*140{,}000}{(1+12\%)^t} = 386{,}801$$

$$NPV(buying) = Asset\ cost - PV\,(tax\ shields\ on\ depreciation)$$

$$= 600{,}000 - \sum_{t=1}^{6} \frac{0.40*100{,}000}{(1+12\%)^t} = 435{,}544$$

In a spreadsheet:

	A	B	C
1	**HOW NOT TO ANALYZE A LEASE**		
2	Asset cost	600,000	
3	Interest rate	12%	
4	Lease rental payment	140,000	
5	Annual depreciation	100,000	
6	Tax rate	40%	
7			
8	NPV (leasing)	386,801	<-- =-PV(B3,5,B4*(1-B6))+B4*(1-B6)
9	NPV (buying)	435,544	<-- =B2+PV(B3,6,B6*B5)

This analysis suggests that leasing the asset is preferable to buying it. However, it is misleading because it ignores the fact that leasing is very much like buying the asset with a loan. The financial risks are thus different when we compare a lease (implicitly a purchase with loan financing) against a straightforward purchase without loan financing. If the company is willing to lease the asset, then perhaps it should also be willing to borrow money to buy

2. At this point we assume that the residual value of the asset at the end of its life is zero. In section 7.5 we drop this assumption.

the assets. This borrowing will change the cash-flow patterns and could also produce tax benefits. Hence, our decision about the leasing decision could change if we were to take the loan potential into account.

In the following section we present a method of analyzing leases that deals with this problem by imagining what kind of loan would produce cash flows (and hence financial risks) equivalent to those produced by the lease. This method of lease analysis is called the *equivalent-loan method.*

7.3 Leasing and Firm Financing—The Equivalent-Loan Method

The idea behind the equivalent-loan method is to devise a hypothetical loan that is somehow equivalent to the lease.[3] It then becomes easy to see whether the lease or the purchase of an asset is preferable.

The easiest way to understand the equivalent-loan method is with an example. We return to the previous example:

	A	B	C	D	E	F	G	H	I
1	**EQUIVALENT-LOAN METHOD—THE LESSEE POINT OF VIEW**								
2	Asset cost	600,000							
3	Interest rate	12%							
4	Lease rental payment	140,000							
5	Annual depreciation	100,000							
6	Tax rate	40%							
7									
8	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	
9	**After-tax cash flows from leasing**								
10	After-tax lease rental	-84,000	-84,000	-84,000	-84,000	-84,000	-84,000		<-- =-$B$4*(1-$B$6)
11									
12	**After-tax cash flows from buying the asset**								
13	Asset cost	-600,000							
14	Depreciation tax shield		40,000	40,000	40,000	40,000	40,000	40,000	<-- =$B$5*$B$6
15	Net cash from buying	-600,000	40,000	40,000	40,000	40,000	40,000	40,000	<-- =G13+G14
16									
17	**Differential cash flow: Lease saves lessee**								
18	Lease minus buy	516,000	-124,000	-124,000	-124,000	-124,000	-124,000	-40,000	<-- =G10-G15
19									
20	IRR of differential cash flow	8.30%	<-- =IRR(B18:H18,0)						
21	Decision??	Buy	<-- =IF(B20<(1-B6)*B3,"Lease","Buy")						

Rows 2–6 give the various parameters of the problem. The spreadsheet then compares two *after-tax* cash flows, that of the lease and that of the buy; we write outflows with a minus sign and inflows (such as the tax shield from the depreciation) with a plus sign.

3. This method is due to Myers, Dill, and Bautista (1976). A somewhat more accessible explanation can be found in Levy and Sarnat (1979).

- The cash flow from leasing the asset is *(1 – tax rate)*lease payment* in each of years 0–5.
- The cash flow from buying the asset is the asset cost in year 0 (an outflow, hence negative) and the tax shield on the asset's depreciation, *tax rate*depreciation*, in years 1–6 (an inflow, hence written here with a positive sign).

The *differential cash flow* between the lease and the buy decision. This line shows that leasing the asset, instead of buying it, results in the following cash flows to the lessor:

- A cash inflow of $516,000 in year 0. This inflow is the *cash saved at time 0* by the lease: Purchasing the asset costs $600,000, whereas leasing the asset costs only $84,000 on an after-tax basis. Thus the lease initially saves the lessee $516,000.
- A cash outflow of $124,000 in years 1–5 and an outflow of $40,000 in year 6. This outflow corresponds to the *after-tax cost of the lease versus the buy* in these years. This cost has two components: the after-tax lease payment ($84,000) and the fact that when the lessee leases, it does not get the tax shield on the asset's depreciation ($40,000).

Thus leasing instead of purchasing the asset is like getting a loan of $516,000 with after-tax repayments of $124,000 in years 1–5 and an after-tax repayment of $40,000 in year 6. The lease, in other words, can be viewed as an alternative method of financing the asset. *In order to compare the lease to the buy, we should compare the cost of this financing with the cost of alternative financing*. The internal rate of return of the differential cash flows—8.30%—gives the after-tax cost of the financing implicit in the lease; this is larger than the after-tax cost of firm borrowing, since in this case (where the firm's tax rate is 40% and its borrowing cost is 12%), this cost is 7.20%. Our conclusion: Buying is preferable to leasing.

Why Do We Decide Against the Lease?

Not everyone is fully convinced by the preceding argument. We therefore present an alternative argument in this subsection. We show that if the firm can borrow at 12%, it can borrow *more money* with the *same schedule of after-tax repayments* as that which resulted from the lease versus the buy. This hypothetical loan is shown in the following table:

	A	B	C	D	E	F	G	H
24	=D26+B26-B27			Split of loan repayment between:			=B26-B27	
25	**Year**	**Principal at beg. year**	**Loan payment, end year**	**Interest**	**Repayment of principal**	**After-tax loan repayment**	**Lease minus buy cash flows, years 1-6**	
26	1	532,070	149,539	63,848	85,691	124,000	124,000	<-- {=-TRANSPOSE (C18:H18)}
27	2	446,379	145,426	53,565	91,861	124,000	124,000	
28	3	354,518	141,017	42,542	98,475	124,000	124,000	
29	4	256,044	136,290	30,725	105,565	124,000	124,000	
30	5	150,479	131,223	18,057	113,166	124,000	124,000	
31	6	37,313	41,791	4,478	37,313	40,000	40,000	
32								
33	=NPV((1-$B$6)*$B$3,G26:$G$31)							
34			=$B$3*B26					

The table (a version of the loan tables discussed in Chapter 1) shows the principal of a hypothetical bank loan bearing a 12% interest rate. At the beginning of year 0 (that is, at the time when the firm either purchases or leases the asset), for example, the firm borrows $532,070 from the bank. At the end of the year, the firm repays $149,539 to the bank, of which $63,848 is interest (since $63,848 = 12%*$532,070) and the remainder, $85,691, is repayment of principal. The net, after-tax, repayment in year 1—assuming full tax deductibility of the interest payment—is (1 – 40%)*$63,848+$85,691 = $124,000, which is, of course, the same after-tax differential cash flow calculated in our original spreadsheet.

Payments in subsequent years are calculated similarly to the illustration in the preceding paragraph. At the beginning of year 6, there is still $37,313 of principal outstanding; this is fully paid off at the end of the year with an after-tax payment of $40,000.

The point of this example? If the firm is considering leasing the asset in order to get the financing of $516,000, which the lease gives, it should instead borrow $532,070 from the bank at 12%; it can repay this larger loan with the same after-tax cash flows as are implicit in the lease. The bottom line: Purchasing is still preferable to leasing the asset.

The alternative loan table shown above was constructed in the following way:

The principal at the beginning of each of years 1–6 is the present value of the lease versus buy outflows, discounted at (1 − 38%)*12%. Thus, for example:

$$532{,}070=\sum_{t=1}^{5}\frac{124{,}000}{(1+(1-0.40)\cdot 0.12)^t}+\frac{40{,}000}{(1+(1-0.40)\cdot 0.12)^6}$$

$$446{,}379=\sum_{t=1}^{4}\frac{124{,}000}{(1+(1-0.40)\cdot 0.12)^t}+\frac{40{,}000}{(1+(1-0.40)\cdot 0.12)^5}$$

$$\vdots$$

$$37{,}313=\frac{40{,}000}{(1+(1-0.40)\cdot 0.12)}$$

Once the principal at the start of each year is known, it is an easy matter to construct the rest of the columns.

$$Interest = 12\% * Principal\ at\ the\ beginning\ of\ the\ year$$

$$Total\ payment = Interest\ in\ year\ t + Repayment\ of\ principal\ in\ year\ t$$

$$After\text{-}tax\ payment,\ year\ t = (1 - Tax\ rate) * Interest + Repayment\ of\ principal$$

7.4 The Lessor's Problem: Calculating the Highest Acceptable Lease Rental

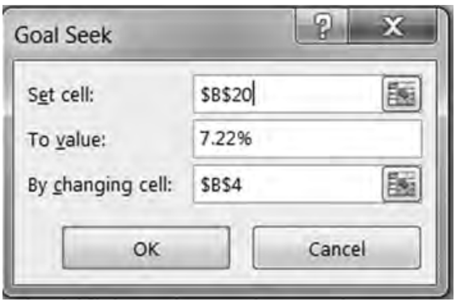
The lessor's problem is the opposite of that of the lessee:

- The lessee has to decide whether—given a rental rate on the leased asset—it is preferable to buy the asset or lease it.
- The lessor has to decide what *minimum rental rate* justifies the purchase of the asset in order to lease it out.

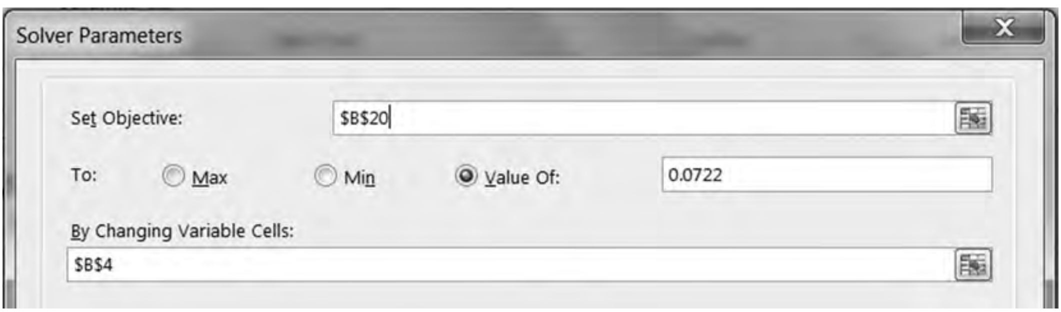
One way of solving the lessor's problem is to turn the above analysis around. We use the Excel **Goal Seek** (**Data|What-if analysis|Goal Seek**) to get $134,826 as the lessor's minimum acceptable rental:

	A	B	C	D	E	F	G	H
1	**THE LESSOR'S PROBLEM Calculating the lowest acceptable lease rate**							
2	Asset cost	600,000						
3	Interest rate	12%						
4	Lowest acceptable lease payment	134,822	<-- Computed either with **Goal Seek** or **Solver**					
5	Annual depreciation	100,000						
6	Tax rate	40%						
7								
8	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
9	**Lessor after-tax cash flows from leasing**							
10	After-tax lease rental	80,893	80,893	80,893	80,893	80,893	80,893	
11								
12	**Lessor after-tax cash flows from buying the asset**							
13	Asset cost	-600,000						
14	Depreciation tax shield		40,000	40,000	40,000	40,000	40,000	40,000
15	Net cash from buying	-600,000	40,000	40,000	40,000	40,000	40,000	40,000
16								
17	**Lessor cash flows**							
18	Lease + buy	-519,107	120,893	120,893	120,893	120,893	120,893	40,000
19								
20	IRR of differential cash flow	7.22%	<-- =IRR(B18:H18)					

Here's what the **Goal Seek** settings look like:



If you're using **Data|Solver** to do this problem, it would look like:



Mini-Case: When Is Leasing Profitable for Both the Lessor and Lessee?

The symmetry between the lessee's problem and the lessor's problem suggests that if the lessee wants to lease, it will not be profitable for the lessor to purchase the asset in order to lease it out.

In some cases, however, it may be that the differences in tax rates between the lessee and the lessor make it profitable for both to enter into a leasing arrangement. Here is an example: Greenville Electric Corp. is a public utility which pays no taxes. Its credit rating is of the highest order, since all of Greenville's debts are guaranteed by the city of Greenville. Greenville Electric has decided that it requires a new turbine. The turbine costs $10,000,000 and will be depreciated to zero salvage value over 5 years. Greenville Electric borrows at 6%.

Greenville can either lease or buy the plant. The lease offer it has is $1,800,000 per year for 6 years (starting today); the lessee is a captive leasing subsidiary of United Turbine Corp., the manufacturer of the turbine. United Turbine Leasing also borrows at 6% and has a tax rate of 40%.

As can be seen below, the lease is profitable to both lessor and lessee. Greenville Electric gets financing at a cost of 3.19%, compared to its borrowing cost of 6%; United Turbine gets an after-tax return of 4.3%, compared to its after-tax borrowing cost of 3.6%. Both Greenville Electric and United Turbine profit.[4]

4. Who loses? The government, of course! What makes the lease profitable is the utilization of otherwise unused depreciation tax shields.

	A	B	C	D	E	F	G	H
1	**GREENVILLE ELECTRIC CORP.**							
2	Turbine cost	10,000,000						
3	Greenville's borrowing rate	6.00%						
4	Lease payment	1,800,000						
5								
6	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
7	**Lessee after-tax lease costs**							
8	After-tax lease rental	-1,800,000	-1,800,000	-1,800,000	-1,800,000	-1,800,000	-1,800,000	
9								
10	**Lessee after-tax purchase costs**							
11	Asset cost	-10,000,000						
12	Depreciation tax shield (Greenville Electric's tax rate = 0)		0	0	0	0	0	0
13	Net cash from buying	-10,000,000	0	0	0	0	0	0
14								
15	**Cash saved by leasing**							
16	Lease - purchase cash flows	8,200,000	-1,800,000	-1,800,000	-1,800,000	-1,800,000	-1,800,000	0
17								
18	IRR of differential cash flow	3.19%	<-- =IRR(B16:H16,0)					
19	Greenville's after-tax borrowing cost	6.00%	<-- =B3					
20								
21	**UNITED TURBINE LEASING CORPORATION**							
22	Turbine cost	10,000,000						
23	Lease payment	1,800,000						
24	Depreciation (straight line, 5 years)	2,000,000						
25	United Turbine's borrowing rate	6.00%						
26	United Turbine's corporate tax rate	40%						
27								
28	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
29	**Lessor cash flows**							
30	Equipment cost	-10,000,000						
31	Lease payment, after tax	1,080,000	1,080,000	1,080,000	1,080,000	1,080,000	1,080,000	
32	Depreciation tax shield		800,000	800,000	800,000	800,000	800,000	800,000
33	Total lessor cash flow	-8,920,000	1,880,000	1,880,000	1,880,000	1,880,000	1,880,000	800,000
34								
35	IRR of lessor cash flows	4.30%	<-- =IRR(B33:H33)					
36	United Turbine's after-tax borrowing cost	3.60%	<-- =B25*(1-B26)					

7.5 Asset Residual Value and Other Considerations

In the above example we have ignored the residual value of the asset—its anticipated market value at the end of the lease term. In a mechanical sense, it is easy to include the residual value in the calculations (but you have to be careful with this—see the warning after the numerical example below). Suppose, for example, you think that the asset will have a market value of $100,000 in year 7; assuming that this value is fully taxed (after all, we've depreciated the asset to zero value over the first 6 years), the after-tax residual value will be (1 – *Tax rate*)*$100,000 = $60,000.

	A	B	C	D	E	F	G	H	I
1	**RESIDUAL VALUES IN LEASE ANALYSIS**								
2	Asset cost	600,000							
3	Interest rate	12%							
4	Lease rental payment	140,000							
5	Annual depreciation	100,000							
6	Tax rate	40%							
7	Residual value	100,000	<-- Anticipated to be realized in year 7; fully taxed						
8									
9	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
10	**After-tax cash flows from leasing**								
11	After-tax lease rental	-84,000	-84,000	-84,000	-84,000	-84,000	-84,000		
12									
13	**After-tax cash flows from buying the asset**								
14	Asset cost	-600,000							
15	Depreciation tax shield		40,000	40,000	40,000	40,000	40,000	40,000	
16	After-tax residual								60,000
17	Net cash from buying	-600,000	40,000	40,000	40,000	40,000	40,000	40,000	60,000
18									
19	**Differential cash flow**								
20	Lease minus buy	516,000	-124,000	-124,000	-124,000	-124,000	-124,000	-40,000	-60,000
21									
22	IRR of differential cash flow	10.49%	<-- =IRR(B20:I20,0)						
23	Decision??	Buy	<-- =IF(B22<(1-B6)*B3,"Lease","Buy")						

Not surprisingly, the possibility of realizing an extra cash flow from asset ownership makes the lease even less attractive than before (you can see this difference by noting that the return rate in cell B22, the IRR of the differential cash flows, has increased from 8.30% in our original example to 10.49%).

Be a bit careful here, however; the spreadsheet treats the residual value as if it has the same certainty of realization as the depreciation tax shields and the lease rentals. This can be far from the truth! There is no good practical solution to this problem; an ad hoc way of dealing with it might be to reduce the $100,000 by a factor which expresses the uncertainty about its realization. The finance technical jargon for this is "certainty equivalent factor," and you can find it referenced in any basic finance text.[5] The last spreadsheet snapshot in this chapter (below) assumes that you've decided that the certainty-equivalence factor for the residual value is 0.7:

5. For further references on certainty-equivalents, see for example Brealey-Myers-Allen (2011, Chapter 9). However, note that neither this work nor the present text (nor anyone else) can tell you precisely how to calculate the certainty-equivalence factor. It depends on your attitudes toward risk.

	A	B	C	D	E	F	G	H	I	J
1	**RESIDUAL VALUES IN LEASE ANALYSIS** **Estimated residual value multiplied by certainty-equivalence factor which represents uncertainty about realizing residual**									
2	Asset cost	600,000								
3	Interest rate	12%								
4	Lease rental payment	140,000								
5	Annual depreciation	100,000								
6	Tax rate	40%								
7	Residual value	100,000	<-- Anticipated to be realized in year 7; fully taxed							
8	Certainty-equivalence factor for residual	0.70								
9										
10	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
11	**After-tax cash flows from leasing**									
12	After-tax lease rental	-84,000	-84,000	-84,000	-84,000	-84,000	-84,000			
13										
14	**After-tax cash flows from buying the asset**									
15	Asset cost	-600,000								
16	Depreciation tax shield		40,000	40,000	40,000	40,000	40,000	40,000		
17	After-tax residual								42,000	<-- =(1-B6)*B7*B8
18	Net cash from buying	-600,000	40,000	40,000	40,000	40,000	40,000	40,000	42,000	
19										
20	**Differential cash flow**									
21	Lease minus buy	516,000	-124,000	-124,000	-124,000	-124,000	-124,000	-40,000	-42,000	
22										
23	IRR of differential cash flow	9.88%	<-- =IRR(B21:I21,0)							
24	Decision??	Buy	<-- =IF(B23<(1-B6)*B3,"Lease","Buy")							

7.6 Leveraged Leasing

Up to this point we analyzed the lease versus purchase decision from both the points of view of the lessee (the long-term user of the asset) and the lessor (the asset's owner, who rents it out to the lessee). In this chapter we analyze leveraged leasing: In a leveraged lease the lessor finances the purchase of the asset to be leased with debt. From the point of view of the lessee, there is no difference in the analysis of a leveraged or a non-leveraged lease. From the lessor's point of view, however, the cash flows of a leveraged lease present some interesting problems.

At least six parties are typically involved in a leveraged lease: the lessee, the equity partners in the lease, the lenders to the equity partners, an owner trustee, an indenture trustee, and the manufacturer of the asset. In most cases, a seventh party is also involved: a lease packager (a broker or leasing company). Figure 7.1 on page 190 illustrates the arrangements among the six parties of a typical leveraged lease.

The two major problems related to the analysis of leveraged leases are these:

- *The straightforward financial analysis of the lease from the point of view of the lessor*. This concerns the calculation of the cash flows obtained by the lessor, and a computation of these cash flows' net present value (NPV) or internal rate of return (IRR).
- *The accounting analysis of the lease*. Accountants use a method called the *multiple phases method* (MPM) to calculate a rate of return on leveraged

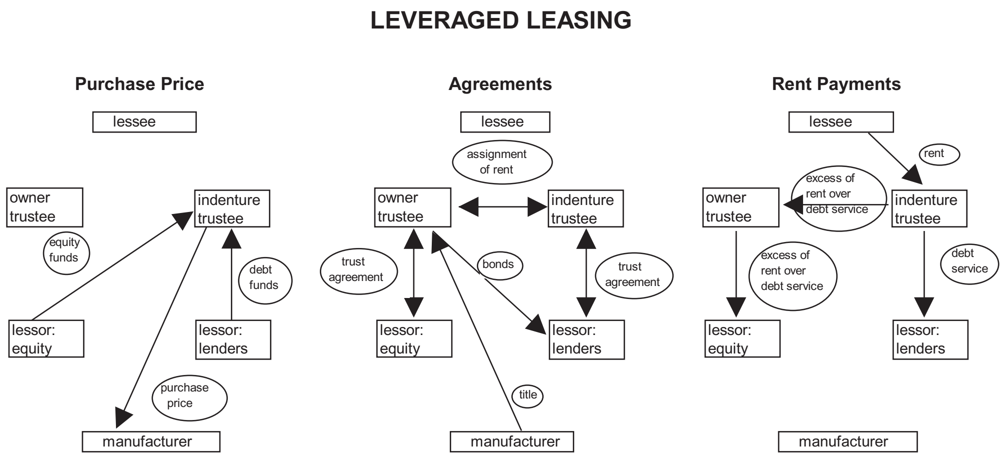


Figure 7.1

leases. The MPM rate of return is different from the internal rate of return (IRR). In an ordinary financial context this should be of no concern, since the efficient-markets hypothesis tells us that only cash flows matter. However, in a less than efficient world, people tend to get very concerned about how things look on their financial statements. Since the accounting rate of return on the lease is difficult to compute, we will use Excel to calculate it; then we will analyze the results.

7.7 A Leveraged Lease Example

We can explore these issues by considering an example, roughly based on an example given in Appendix E of FASB 13, the accounting profession's magnum opus on accounting for leases.

A leasing company is considering the purchase of an asset whose cost is $1,000,000. The asset will be purchased with $200,000 of the company's equity and with $800,000 of debt. The interest on the debt is 10%, so that the annual payment of interest and principal over the 15-year term of the debt is $105,179.[6]

6. Using Excel: =PMT(10%,15,-800000) gives 105,179.

The company will lease the asset out for $110,000 per year, payable at the end of each year. The lease term is 15 years. The asset will be depreciated over a period of 8 years, using standard IRS depreciation schedule for assets with a 7-year life.[7] The depreciation schedule for such assets is:

Year	Depreciation
1	14.28%
2	24.49%
3	17.49%
4	12.5%
5	8.92%
6	8.92%
7	8.92%
8	4.48%

Because the asset will be fully depreciated at the time it is sold (year 16), the whole anticipated residual value ($300,000) will be taxable. Since the company's tax rate is 40%, this means that the after-tax cash flow from the residual is (1 – 40%)∗300,000 = $180,000.

These facts are summarized in the spreadsheet below, which also derives the lessor's cash flows:

7. The depreciation schedule we use is referred to as the modified cost recovery system (MACRS) depreciation. More information can be obtained from an introductory finance text or from many websites (one example: www.real-estate-owner.com/depreciation-chart.html).

	A	B	C	D	E	F	G	H	I	J
1	**BASIC LEVERAGED LEASE EXAMPLE**									
2	Cost of asset	1,000,000								
3	Lease term	15								
4	Residual value	300,000	<-- Realized year 16							
5	Equity	200,000								
6	Debt	800,000	<-- 15-year term loan, equal payments of interest and principal							
7	Interest	10%								
8	Annual debt payment	105,179	<-- =PMT(B7,B3,-B6)				=(1-tax)*C14+tax*D14-(1-tax)*G14-H14			
9	Annual rent received	110,000								
10	Tax rate	40%								
11										
12	**Year**	**Equity Invested**	**Rental or salvage**	**Depreciation**	**Principal at start of year**	**Loan payment**	**Interest**	**Repayment of principal**	**Cash flow to equity**	
13	0	-200,000							-200,000	
14	1		110,000	142,800	800,000	105,179	80,000	25,179	49,941	
15	2		110,000	244,900	774,821	105,179	77,482	27,697	89,774	
16	3		110,000	174,900	747,124	105,179	74,712	30,467	60,666	
17	4		110,000	125,000	716,657	105,179	71,666	33,513	39,487	
18	5		110,000	89,200	683,144	105,179	68,314	36,865	23,827	
19	6		110,000	89,200	646,280	105,179	64,628	40,551	22,352	
20	7		110,000	89,200	605,728	105,179	60,573	44,606	20,730	
21	8		110,000	44,800	561,122	105,179	56,112	49,067	1,186	
22	9		110,000		512,056	105,179	51,206	53,973	-18,697	
23	10		110,000		458,082	105,179	45,808	59,371	-20,856	
24	11		110,000		398,711	105,179	39,871	65,308	-23,231	
25	12		110,000		333,403	105,179	33,340	71,839	-25,843	
26	13		110,000		261,565	105,179	26,156	79,023	-28,716	
27	14		110,000		182,542	105,179	18,254	86,925	-31,877	
28	15		110,000		95,617	105,179	9,562	95,617	-35,354	
29	16		300,000						180,000	
30										
31						IRR of cash flows			12.46%	<-- =IRR(I13:I29)

The last column gives the cash flow to the equity owners of the asset. A typical year's cash flow for the equity owner is calculated as follows:

$$Cash\ flow(t) = (1 - Tax) * Lease\ payment + Tax * Depreciation \\ - (1 - Tax) * Interest(t) - Principal\ repayment(t)$$

The cash flows of the typical long-lived leveraged lease are usually positive at the beginning of the lease term, and then decline over time, turning positive again at the end, when the residual value is received. There are three reasons for this phenomenon:

- The cash flow which stems from depreciation typically ends or falls off rapidly before the end of the lease term. The more accelerated the depreciation method, the larger will be the depreciation allowances (and hence the larger the depreciation tax shields) at the beginning of the asset's life.
- In the later years of the lease, the portion of the annual debt payments devoted to interest (tax-deductible) falls, while the portion of the annual debt payments that constitutes a repayment of principal (not tax deductible) rises.
- Finally, of course, we anticipated a large cash flow from the realization of the asset's residual value at the end of the lease term.

7.8 Summary

This chapter has looked at the lease-purchase decision. We have examined the decision to lease as a purely financing decision, assuming (i) that all operational factors between leasing and buying are equivalent, and (ii) that the essential firm decision to acquire the use of the asset has already been made. On the basis of these assumptions, the lease-purchase decision can be made using the equivalent-loan method.

A leveraged lease is an arrangement whereby the lessor—the owner of the asset—finances his investment with a combination of debt and equity. In this chapter we analyzed the equity income of the lessor in a leveraged lease. The economic analysis of the lease cash flows shows that at some point in the lease life, the equity owner has negative equity value.

Exercises

1. Your company is considering either purchasing or leasing an asset which costs $1,000,000. The asset, if purchased, will be depreciated on a straight-line basis over 6 years to a zero residual value. A leasing company is willing to lease the asset for $300,000 per year; the first payment on the lease is due at the time the lease is undertaken (i.e., year 0), and the remaining 5 payments are due at the beginning of years 1–5. Your company has a tax rate $T_C = 40\%$ and can borrow at 10% from its bank.

   a. Should your company lease or purchase the asset?

   b. What is the maximum lease payment it will agree to pay?

2. ABC Corp. is considering leasing an asset from XYZ Corp. Here are the relevant facts:

Asset cost	$1,000,000	
Depreciation schedule	year 1:	20%
	year 2:	32%
	year 3:	19.20%
	year 4:	11.52%
	year 5:	11.52%
	year 6:	5.76%
Lease term	6 years	
Lease payment	$200,000 per year, at the beginning of years 0, 1, … , 5.	
Asset residual value	zero	
Tax rates	ABC: $T_C = 0\%$ (ABC has tax-loss carryforwards which prevent it from utilizing any additional tax shields) XYZ: $T_C = 40\%$	

ABC's interest costs are 10% and XYZ's interest costs are 7%. Show that it will be advantageous both for ABC to lease the asset and for XYZ to purchase the asset in order to lease it out to ABC.

3. Continuing with the above example: Find the *maximum rental* which ABC will pay and the *minimum rental* which XYZ will accept.

4. Perform a sensitivity analysis (using **Data|Table**) on the certainty-equivalence factor in Section 7.5, showing how the IRR of the differential cash flows varies with the CE factor.

5. Hemp Airlines (HA, "we fly high") is about to buy 5 CFA3000 commuter jets. Each airplane costs $50 million. A friendly bank has put together a consortium to finance the deal. The consortium includes a 20% equity investment and an 80% debt component. The debt has an interest rate of 8% annually, and is a term loan over 10 years. At the end of each of the next 10 years, HA will pay a lease payment of $35 million. At the end of the 10-year lease term, Hemp has the option to buy the aircraft for $10 million each; it is anticipated that it will exercise this option in which the planes are priced at their anticipated fair market value. The airplanes will be depreciated on a straight-line basis over 5 years to zero salvage value.

   If the equity partner in the lease has a tax rate of 35%, what is its expected compound rate of return?

II PORTFOLIO MODELS

Modern portfolio theory, which has its origins in the work of Harry Markowitz, John Lintner, Jan Mossin, and William Sharpe, represents one of the great advances in finance. Chapters 8–14 implement some of the ideas of these researchers and show you how to compute the standard portfolio problems in finance. In these chapters we make intensive use of Excel's matrix functions, array functions, and data tables (see Chapters 32, 34, and 31).

Chapter 8 reviews the basic mechanics of portfolio calculations. Starting with price data, we calculate asset and portfolio returns. We start with a simple two-asset problem and then generalize to multiple asset portfolio calculations. Chapter 9 discusses both the theory and the mechanics of the calculation of efficient portfolios when there are no restrictions on short sales. Using Excel's matrix functions we can calculate two efficient portfolios, which can then be used to plot the whole efficient frontier.

The remaining chapters of this section discuss computational and implementation issues:

- Chapter 10 shows how to use return data to calculate the variance-covariance matrix. Excel's matrix-handling capabilities make it easy to do this calculation.
- Chapter 11 discusses the computation of beta, and we replicate a simple test of the capital asset pricing model (CAPM). We use some market data to derive the security market line (SML). We then relate the results to Roll's criticism of these tests. Excel makes it easy to do the regression analysis required for these tests. (Regressions are discussed in Chapter 33.)
- The preceding chapters have assumed that portfolio optimizers could sell securities short. In Chapter 12 we show how to use Excel's **Solver** to compute efficient portfolios when short sales are not allowed. We also show how to integrate other portfolio constraints into the optimization problem.
- Chapter 13 discusses the Black-Litterman model. This widely used model takes as its starting point the optimality of the benchmark portfolio and uses this assumption to derive the market's expected returns. The optimizer can then adjust the asset allocation to account for his own opinions.
- Chapter 14 shows how to do an event study, which is an attempt to determine whether a particular event in the capital market or in the life of a company affected a company's stock market performance. The event-study methodology aims to separate company-specific events from market and/or industry specific events and has often been used as evidence for or against market efficiency.

8 Portfolio Models—Introduction

8.1 Overview

In this chapter we review the basic mechanics of portfolio calculations. We start with a simple example of two assets, showing how to derive the return distributions from historical price data. We then discuss the general case of N assets; for this case it becomes convenient to use matrix notation and exploit Excel's matrix handling capabilities.

It is useful before going on to review some basic notation: Each asset i (assets may be stocks, bonds, real estate, or whatever, although our numerical examples will be largely confined to stocks) is characterized by several statistics: $E(r_i)$, the expected return on asset i; $Var(r_i)$, the variance of asset i's return; and $Cov(r_i, r_j)$, the covariance of asset i's and asset j's returns. Occasionally we will use μ_i to denote the expected return on asset i. In addition, it will often be convenient to write $Cov(r_i, r_j)$ as σ_{ij} and $Var(r_i)$ as σ_{ii} (instead of σ_i^2, as usual). Since the covariance of an asset's returns with itself, $\sigma_{ii} = Cov(r_i, r_i)$, is in fact the variance of the asset's returns, this notation is not only economical but also logical.

8.2 Computing Returns for Apple (AAPL) and Google (GOOG)

In this section we compute the return statistics for two stocks: Apple (stock symbol AAPL) and Google (GOOG). Here is the price and return data. The returns include the dividends; see Appendix 8.1 for more details.

	A	B	C	D	E	F	G
1	**PRICES AND RETURNS FOR APPLE AND GOOGLE June 2007 - June 2012**						
2	Monthly mean	2.61%	-0.24%	<-- =AVERAGE(F11:F71)			
3	Monthly variance	0.0125	0.0102	<-- =VAR.S(F11:F71)			
4	Monthly standard deviation	11.17%	10.09%	<-- =STDEV.S(F11:F71)			
5							
6	Annual mean	31.31%	-2.91%	<-- =12*C2			
7	Annual variance	0.1497	0.1221	<-- =12*C3			
8	Annual standard deviation	38.70%	34.94%	<-- =SQRT(12)*C4			
9							
10	**Date**	**AAPL**	**Google**		**AAPL**	**Google**	
11	1-Jun-07	122.04	580.11				
12	2-Jul-07	131.76	645.90		0.0766	0.1074	<-- =LN(C12/C11)
13	1-Aug-07	138.48	599.39		0.0497	-0.0747	<-- =LN(C13/C12)
66	3-Jan-12	456.48	522.70		0.1197	0.0246	<-- =LN(C66/C65)
67	1-Feb-12	542.44	497.91		0.1725	-0.0486	<-- =LN(C67/C66)
68	1-Mar-12	599.55	471.38		0.1001	-0.0548	
69	2-Apr-12	583.98	458.16		-0.0263	-0.0284	
70	1-May-12	577.73	449.45		-0.0108	-0.0192	
71	1-Jun-12	584.00	501.50		0.0108	0.1096	

These data give the closing price at the end of each month for each stock. We define the return as

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

At the top of the Excel sheet we calculate the return statistics for each stock. The *monthly return* is the percentage return that would be earned by an investor who bought the stock at the end of a particular month $t - 1$ and sold it at the end of the following month.

Note that we use the continuously compounded return on the stock, $r_t = \ln(P_t/P_{t-1})$. An alternative would have been to use the discrete return, $P_t/P_{t-1} - 1$. Appendix 2 at the end of this chapter discusses the reasons for our choice of the continuously compounded return.

We now make a heroic assumption: We assume that the return data for the 60 months represent the distribution of the returns for the coming month. We thus assume that the past gives us some information about the way returns will behave in the future. This assumption allows us to assume that the average of the historic data represents the *expected monthly return* from each stock. It also allows us to assume that we may learn from the historic data what is the variance of the future returns. Using the **Average, Var.s**, and **Stdev.s** functions in Excel, we calculate the statistics for the return distribution:

	A	B	C	D	E	F
2	Monthly mean	2.61%	-0.24%	<-- =AVERAGE(F11:F71)		
3	Monthly variance	0.0125	0.0102	<-- =VAR.S(F11:F71)		
4	Monthly standard deviation	11.17%	10.09%	<-- =STDEV.S(F11:F71)		
5						
6	Annual mean	31.31%	-2.91%	<-- =12*C2		
7	Annual variance	0.1497	0.1221	<-- =12*C3		
8	Annual standard deviation	38.70%	34.94%	<-- =SQRT(12)*C4		

Note: Sample Versus Population Statistics and Excel

We interrupt this discussion of portfolio computations with a short note on the statistical computations. In statistics it is common to distinguish between sample statistics and population statistics. If we examine the whole range of possibilities for a given random variable, then we are dealing with the *population*. If we are dealing with a set of outcomes for the random variable, then we are dealing with a *sample*. In the case of portfolio return statistics we will almost always be dealing with a *sample* rather than the whole population.

The table below gives the definitions of the population and sample statistics and the Excel functions that compute them.

	Population	**Sample**
Mean, average, μ	$\text{Average} = \frac{1}{N}\sum_{i=1}^{N} r_i$	$\text{Average} = \frac{1}{N}\sum_{i=1}^{N} r_i$
Variance	$\text{Var.p} = \frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2$	$\text{Var.s} = \frac{1}{N-1}\sum_{i=1}^{N}(r_i - \bar{r})^2$
Standard deviation	$\text{Stdev.p} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2}$	$\text{Stdev.s} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(r_i - \bar{r})^2}$
Covariance	$\text{Covariance.p} = \frac{1}{N}\sum_{t=1}^{N}(r_{it} - \bar{r}_i)(r_{jt} - \bar{r}_j)$	$\text{Covariance.s} = \frac{1}{N-1}\sum_{t=1}^{N}(r_{it} - \bar{r}_i)(r_{jt} - \bar{r}_j)$

Correlation

$$\text{Correl}(i, j) = \frac{\text{Covar.p}(i, j)}{\text{Stdev.p}(i) * \text{Stdev.p}(j)} = \frac{\text{Covar.s}(i, j)}{\text{Stdev.s}(i) * \text{Stdev.s}(j)}$$

	Population	**Sample**
Alternative Excel from previous versions, still work	**Var, VarP, Stdev, StdevP** **VarS, StdevS** **Covar** computes the population covariance. New versions of Excel distinguish between population and sample covariance.	
Regression slope	Excel **Slope**, equivalent to $\text{slope}(i\_\text{data}, M\_\text{data}) = \dfrac{\text{Covar.p}(i, M)}{\text{Var.p}(i, M)} = \dfrac{\text{Covar.S}(i, M)}{\text{Var.S}(i, M)}$	

Sample or Population? Does It Matter?

All of this discussion about population versus sample statistics may not matter much. We like the point of view stated by Press et al. in their splendid book *Numerical Recipes*: "There is a long story about why the denominator is $N - 1$ instead of N. If you have never heard that story, you may consult any good statistics text. We might also comment that if the difference between N and $N - 1$ ever matters to you, then you are up to no good anyway—e.g., trying to substantiate a questionable hypothesis with marginal data."[1]

Back to Our Portfolio Example

Next we want to calculate the *covariance* of the returns.

	A	B	C	D	E	F	G	H
1	**COMPUTING COVARIANCE AND CORRELATION**							
2		Stock returns			Return minus average			
3	**Date**	**AAPL**	**Google**		**AAPL**	**Google**	**Product**	
4	2-Jul-07	7.66%	10.74%	=B4-$B$65 -->	0.0505	0.1099	0.0056	<-- =E4*F4
5	1-Aug-07	4.97%	-7.47%	=B5-$B$65 -->	0.0237	-0.0723	-0.0017	<-- =E5*F5
59	1-Feb-12	17.25%	-4.86%		0.1464	-0.0462	-0.0068	
60	1-Mar-12	10.01%	-5.48%		0.0740	-0.0523	-0.0039	
61	2-Apr-12	-2.63%	-2.84%		-0.0524	-0.0260	0.0014	
62	1-May-12	-1.08%	-1.92%		-0.0369	-0.0168	0.0006	
63	1-Jun-12	1.08%	10.96%		-0.0153	0.1120	-0.0017	
64								
65	Average	2.61%	-0.24%					
66	Variance	0.0125	0.0102	<-- =VAR.S(C4:C63)		**Covariance computation**		
67	Standard deviation	0.1117	0.1009	<-- =STDEV.S(C4:C63)			0.001956	<-- =AVERAGE(G4:G63)
68		0.1117	0.1009	<-- =SQRT(C66)			0.001956	<-- =COVARIANCE.P(B4:B63,C4:C63)
69							0.001956	<-- =COVAR(B4:B63,C4:C63)
70		0.0123	0.0100	<-- =VAR.P(C4:C63)			0.001989	<-- =AVERAGE(G4:G63)*60/59
71		0.1108	0.1000	<-- =STDEV.P(C4:C63)			0.001989	<-- =COVARIANCE.S(B4:B63,C4:C63)
72		0.1108	0.1000	<-- =SQRT(C70)				
73						**Correlation computation**		
74							0.1765	<-- =CORREL(B4:B63,C4:C63)
75							0.1765	<-- =G68/(B71*C71)
76							0.1765	<-- =G71/(B67*C67)

1. William H. Press et al., *Numerical Recipes* (Cambridge University Press, 1986), p. 456.

The column **Product** contains the multiple of the deviation from the mean in each month, i.e., the terms $(r_{WMT,t} - E(r_{WMT}))(r_{TGM,t} - E(r_{TGT}))$, for $t = 1, \ldots, 12$. The population covariance is **Average(product)** = 0.0020. While it is worthwhile calculating the covariance this way at least once, there is a shorter way which is also illustrated above: The Excel functions **Covariance**.**P(AAPL,GOOG)** and **Covariance.S(AAPL,GOOG)** calculate the population and the sample covariance directly.[2] There is no necessity to find the difference between the returns and the means. Simply use **Covariance.P** or **Covariance.S** directly on the columns, as illustrated above.

The covariance is a hard number to interpret, since its size depends on the units in which we measure the returns. (If we were to write the returns in percentages—i.e., 4 instead of 0.04—then the covariance would be 20, which is 10,000 times the number we just calculated.) We can also calculate the *correlation coefficient* ρ_{AB}, which is defined as

$$\rho_{AAPL,GOOG} = \frac{Cov(r_{AAPL}, r_{GOOG})}{\sigma_{AAPL}\sigma_{GOOG}}$$

The correlation coefficient is unit-free; calculating it for our example gives $\rho_{AAPL,GOOG} = 0.1765$. As illustrated above, the correlation coefficient can be calculated directly in Excel using the function **Correl(AAPL,GOOG)**.

The correlation coefficient measures the degree of linear relation between the returns of Stock A and Stock B. The following facts can be proven about the correlation coefficient:

- The correlation coefficient is always between +1 and –1: $-1 \le \rho_{AB} \le 1$.
- If the correlation coefficient is +1, then the returns on the two assets are linearly related with a positive slope; i.e., if $\rho_{AB} = 1$, then

$$r_{At} = c + dr_{Bt}, \quad \text{where } d > 0$$

- If the correlation coefficient is –1, then the returns on the two assets are linearly related with a negative slope; i.e., if $\rho_{AB} = -1$, then

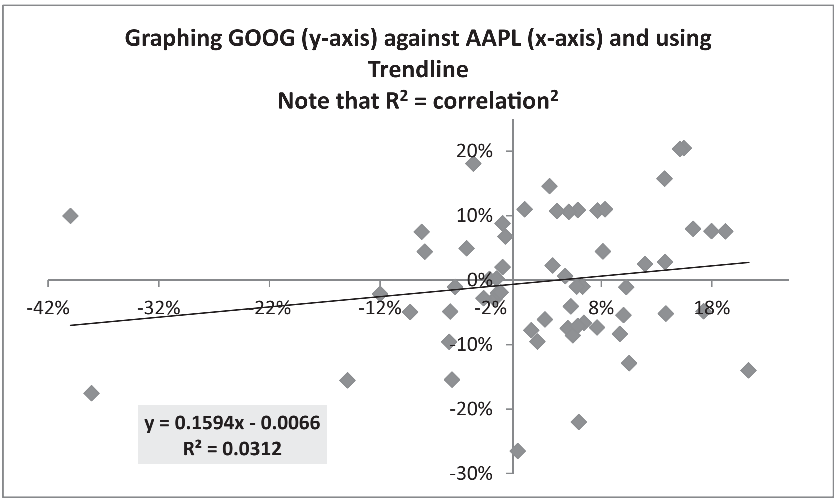
$$r_{At} = c + dr_{Bt}, \quad \text{where } d < 0$$

2. The function **Covar** was used in pre-Excel 2013 and calculates the population covariance; it still works in current versions. As noted elsewhere, previous Excel versions had functions **VarP**, **StdevP**, **VarS**, **StdevS** (though still working, these are now replaced by **Var.P**, **Stdev.P**, **Var.S**, **Stdev.S**). Note how, in our example, the difference between the sample and population covariance is very small.

• If the return distributions are independent, then the correlation coefficient will be zero. (The opposite is not true: If the correlation coefficient is zero, this does not necessarily mean that the returns are independent. See the Exercises for an example.)

A Different View of the Correlation Coefficient

Another way to look at the correlation coefficient is to graph the Apple and Google returns on the same axes and then use the Excel **Trendline** facility to regress the returns of GOOG on those of AAPL. (The use of Excel's **Trendline** function—used to calculate the regression equation—is explained in Chapter 33.) You can confirm from the previous calculations that the regression R^2 is the correlation squared:



8.3 Calculating Portfolio Means and Variances

In this section we show how to do the basic calculations for a portfolio's mean and variance. Suppose we form a portfolio invested equally in AAPL and GOOG. What will be the mean and the variance of this portfolio? It is worth doing the brute force calculations at least once in Excel:

	A	B	C	D	E
1	**CALCULATING THE MEAN AND STANDARD DEVIATION OF A PORTFOLIO**				
2	Proportion of AAPL	0.5			
3	Proportion of GOOG	0.5	<-- =1-B2		
4					
5		**AAPL return**	**GOOG return**	**Portfolio return**	
6	2-Jul-07	7.66%	10.74%	9.20%	<-- =$B$2*B6+$B$3*C6
7	1-Aug-07	4.97%	-7.47%	-1.25%	<-- =$B$2*B7+$B$3*C7
8	4-Sep-07	10.28%	-1.13%	4.57%	<-- =$B$2*B8+$B$3*C8
9	1-Oct-07	21.33%	-14.03%	3.65%	
10	1-Nov-07	-4.15%	4.91%	0.38%	
11	3-Dec-07	8.35%	10.97%	9.66%	
12	2-Jan-08	-38.07%	-17.58%	-27.83%	
13	1-Feb-08	-7.95%	4.37%	-1.79%	
14	3-Mar-08	13.79%	2.81%	8.30%	
15	1-Apr-08	19.24%	7.55%	13.40%	
16	1-May-08	8.17%	4.44%	6.30%	
17	2-Jun-08	-11.98%	-2.15%	-7.06%	

Computing the statistics for this example:

	G	H	I	J
5	**Asset returns**	**AAPL**	**GOOG**	
6	Mean return	2.61%	-0.24%	<-- =AVERAGE(C6:C65)
7	Variance	0.0125	0.0102	<-- =VAR.S(C6:C65)
8	Standard deviation	11.17%	10.09%	<-- =STDEV.S(C6:C65)
9	Covariance	0.0020	<-- =COVARIANCE.S(B6:B65,C6:C65)	
10				
11	Portfolio mean return			
12		1.18%	<-- =AVERAGE(D6:D65)	
13		1.18%	<-- =B2*H6+B3*I6	
14	Portfolio return variance			
15		0.0067	<-- =VAR.S(D6:D65)	
16		0.0067	<-- =B2^2*H7+B3^2*I7+2*B2*B3*H9	
17	Portfolio return standard deviation			
18		0.0816	<-- =SQRT(H16)	
19		0.0816	<-- =STDEV.S(D6:D65)	

The mean portfolio return is exactly the average of the mean returns of the two assets:

$$Expected\ portfolio\ return = E(r_p) = 0.5E(r_{AAPL}) + 0.5E(r_{GOOG})$$

In general the mean return of the portfolio is the *weighted average return* of the component stocks. If we denote by x the proportion invested in AAPL and $1 - x$ the proportion invested in stock GOOG, then the expected portfolio return is given by:

$$E(r_p) = xE(r_{AAPL}) + (1-x)E(r_{GOOG})$$

However, the portfolio's variance is *not* the average of the two variances of the stocks! The formula for the variance is

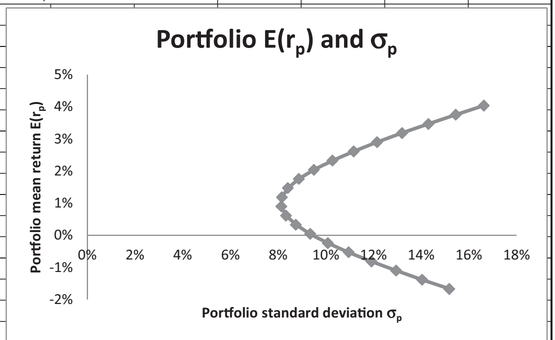
$$Var(r_p) = x^2 Var(r_{AAPL}) + (1-x)^2 Var(r_{GOOG}) + 2x(1-x)Cov(r_{AAPL}, r_{GOOG})$$

Another way of writing this is

$$\sigma_p^2 = x^2\sigma_{AAPL}^2 + (1-x)^2\sigma_{GOOG}^2 + 2x(1-x)\rho_{AAPL,GOOG}\sigma_{AAPL}\sigma_{GOOG}$$

A frequently performed exercise is to plot the means and standard deviations for various portfolio proportions x. To do this we build a table using Excel's **Data|What-If|DataTable** command (see Chapter 31); cells B16 and C16 contain the data table's header, which refers to cells B11 and B10, respectively.

	A	B	C	D	E	F	G	H	I	J
1	**CALCULATING THE MEAN AND STANDARD DEVIATION OF A PORTFOLIO**									
2	Asset returns	**AAPL**	**GOOG**							
3	Mean return	2.61%	-0.24%							
4	Variance	0.0125	0.0102							
5	Standard deviation	11.17%	10.09%							
6	Covariance	0.0020								
7										
8	Proportion of AAPL	0.5								
9										
10	Portfolio mean return	1.18%	<-- =B8*B3+(1-B8)*C3							
11	Portfolio return variance	0.0067	<-- =B8^2*B4+(1-B8)^2*C4+2*B8*(1-B8)*B6							
12	Portfolio return standard deviation	8.16%	<-- =SQRT(B11)							
13										
14	**Data table—varying the proportion of AAPL**									
15		Portfolio standard deviation	Portfolio mean return							
16	Proportion of AAPL	8.16%	1.18%	<-- =B10, table header						
17	-0.5	15.18%	-1.67%							
18	-0.4	14.04%	-1.38%							
19	-0.3	12.95%	-1.10%							
20	-0.2	11.91%	-0.81%							
21	-0.1	10.95%	-0.53%							
22	0	10.09%	-0.24%							
23	0.1	9.34%	0.04%							
24	0.2	8.75%	0.33%							
25	0.3	8.33%	0.61%							
26	0.4	8.13%	0.90%							
27	0.5	8.16%	1.18%							
28	0.6	8.41%	1.47%							
29	0.7	8.87%	1.75%							
30	0.8	9.50%	2.04%							
31	0.9	10.28%	2.32%							
32	1	11.17%	2.61%							
33	1.1	12.15%	2.89%							
34	1.2	13.20%	3.18%							
35	1.3	14.30%	3.46%							
36	1.4	15.45%	3.75%							
37	1.5	16.62%	4.04%							



8.4 Portfolio Mean and Variance—Case of *N* Assets

In the previous sections we discussed the computation of a portfolio's mean, variance, and standard deviation for the case wherein the portfolio is composed of only two assets. In this section we extend this discussion to portfolios of more than two assets. For this case, matrix notation greatly simplifies the writing of the portfolio problem.[3] In the general case of N assets, suppose that the proportion of asset i in the portfolio is denoted by x_i. We require that $\sum_i x_i = 1$, but we place no restrictions on the signs of the x_i; if $x_i > 0$, this

3. Chapter 32 gives an introduction to matrices sufficient to deal with all the problems encountered in this book. The Excel matrix functions **MMult** and **MInverse** used in portfolio problems are discussed in this chapter.

indicates a purchase of asset i and if $x_i < 0$, this indicates a short sale.[4] We usually write the portfolio composition x and the vector of means $E(r)$ as column vectors (we make no pretense at consistency—when convenient we will write these as row vectors):

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} \quad E(r) = \begin{bmatrix} E(r_1) \\ E(r_2) \\ E(r_3) \\ \vdots \\ E(r_N) \end{bmatrix}$$

We may then write x^T and $E(r)^T$ as the transpose of these two vectors:

$$x^T = [x_1, x_2, \ldots, x_N], \quad E(r)^T = [E(r_1), E(r_2), \ldots, E(r_N)]$$

The expected return of the portfolio whose proportions are given by X is the weighted average of the expected returns of the individual assets:

$$E(r_x) = \sum_{i=1}^{N} x_i E(r_i)$$

which, in matrix notation, can be written as:

$$E(r_p) = \sum_{i=1}^{N} x_i E(r_i) = x^T E(r) = E(r)^T x$$

The portfolio's variance is given by

$$Var(r_X) = \sum_{i=1}^{N} (x_i)^2 Var(r_i) + 2\sum_{i=1}^{N} \sum_{j=i+1}^{N} x_i x_j Cov(r_i, r_j)$$

This looks bad, but it is really a straightforward extension of the expression for the variance of a portfolio of two assets which we had before: Each asset's variance appears once, multiplied by the square of the asset's proportion in the portfolio; the covariance of each pair of assets appears once, multiplied by twice the product of the individual assets' proportions. Another way of writing the variance is to use the notation

$$Var(r_i) = \sigma_{ii}, \quad Cov(r_i, r_j) = \sigma_{ij}$$

4. In Chapter 12 we discuss portfolio optimization when there are short-sale restrictions. In the meantime we assume that short selling is unrestricted.

In this case the variance of portfolio x is:

$$Var(r_X) = \sum_i \sum_j x_i x_j \sigma_{ij}$$

The most economical representation of the portfolio variance uses matrix notation. It is also the easiest representation to implement for large portfolios in Excel. In this representation we call the matrix that has σ_{ij} in the ith row and the jth column the *variance-covariance matrix*:

$$S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \cdots & \sigma_{2N} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \cdots & \sigma_{3N} \\ \vdots & & & & \\ \sigma_{N1} & \sigma_{N2} & \sigma_{N3} & \cdots & \sigma_{NN} \end{bmatrix}$$

Then the portfolio variance is given by $Var(r_p) = x^T S x$. In Excel formulas, this is written as the array function **mmult(mmult(transpose(x),S),x)**.[5]

Computing the Covariance Between Two Portfolios

If we have two portfolios represented as row vectors, $x = [x_1, x_2, \ldots, x_N]$ and $y = [y_1, y_2, \ldots, y_N]$, then the covariance of the two portfolios is given by $Cov(x,y) = xSy^T = ySx^T$. In Excel formulas this is the array function **MMult (MMult(x,S),Transpose(y))**.

Portfolio Calculations Using Matrices—An Example

We implement the above formulas in a numerical example. Suppose that there are four risky assets which have the following expected returns and variance-covariance matrix:

	A	B	C	D	E	F
1	**A FOUR-ASSET PORTFOLIO PROBLEM**					
2	**Variance-covariance, S**					**Mean returns E(r)**
3	0.10	0.01	0.03	0.05		6%
4	0.01	0.30	0.06	-0.04		8%
5	0.03	0.06	0.40	0.02		10%
6	0.05	-0.04	0.20	0.50		15%

5. Array functions are discussed in Chapter 34. The many examples below illustrate their use for portfolio optimization.

We consider two portfolios of risky assets:

	A	B	C	D	E
8	Portfolio x	0.20	0.30	0.40	0.10
9	Portfolio y	0.20	0.10	0.10	0.60

We calculate the means, variances, and covariance of the two portfolios. We use the Excel array function **MMult** for the matrix multiplications and the array function **Transpose** to make a row vector into a column vector.[6]

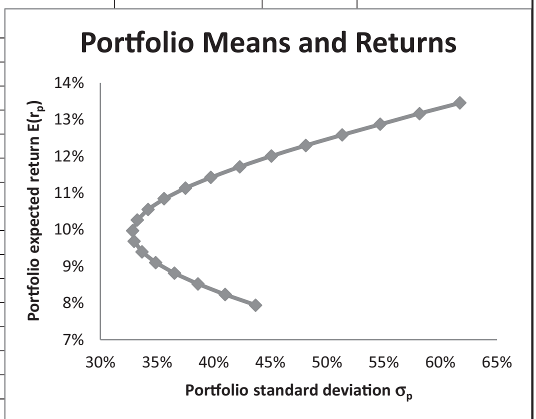
	A	B	C	D	E	F	G
1	**A FOUR-ASSET PORTFOLIO PROBLEM**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Portfolio x	0.20	0.30	0.40	0.10		
9	Portfolio y	0.20	0.10	0.10	0.60		
10							
11	**Portfolio x and y statistics: Mean, variance, covariance, correlation**						
12	Mean, $E(r_x)$	9.10%		Mean, $E(r_y)$	12.00%	<-- {=MMULT(B9:E9,F3:F6)}	
13	Variance σ_x^2	0.1216		Variance σ_y^2	0.2034	<-- {=MMULT(MMULT(B9:E9,A3:D6),TRANSPOSE(B9:E9))}	
14	Covariance(x,y)	0.0714					
15	Correlation ρ_{xy}	0.4540	<-- =B14/SQRT(B13*E13)				

We can now calculate the standard deviation and return of combinations of portfolios *x* and *y*. Note that once we have calculated the means, variances, and the covariance of the returns of the two portfolios, the calculation of the mean and the variance of any portfolio is the same as for the two-asset case.[7]

6. Remember from Chapter 34 that **MMult** and **Transpose** are array functions and must be entered by pressing [Ctrl] + [Shift] + [Enter] simultaneously.

7. This sentence is critical. As we shall see in the next chapter, all portfolio problems of *N* assets ultimately boil down to the two-asset case.

	A	B	C	D	E	F	G
1	**A FOUR-ASSET PORTFOLIO PROBLEM**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Portfolio x	0.20	0.30	0.40	0.10		
9	Portfolio y	0.20	0.10	0.10	0.60		
10							
11	**Portfolio x and y statistics: Mean, variance, covariance, correlation**						
12	Mean, $E(r_x)$	9.10%		Mean, $E(r_y)$	12.00%	<-- {=MMULT(B9:E9,F3:F6)}	
13	Variance σ_x^2	0.1216		Variance σ_y^2	0.2034	<--	
14	Covariance(x,y)	0.0714				{=MMULT(MMULT(B9:E9,A3:D6),TRANSPOSE(B9:E9))}	
15	Correlation ρ_{xy}	0.4540	<-- =B14/SQRT(B13*E13)				
16							
17	**Calculating returns of combinations of Portfolio x and Portfolio y**						
18	Proportion of x	0.3					
19	Mean portfolio return, $E(r_p)$	11.13%	<-- =B18*B12+(1-B18)*E12				
20	Portfolio variance σ_p^2	0.1406	<-- =B18^2*B13+(1-B18)^2*E13+2*B18*(1-B18)*B14				
21	Portfolio standard deviation σ_p	37.50%	<-- =SQRT(B20)				
22							
23	**Table of returns (using Data Table)**						
24	**Proportion of x**	**Standard deviation**	**Mean**				
25		37.50%	11.13%				
26	-0.5	61.72%	13.45%				
27	-0.4	58.15%	13.16%				
28	-0.3	54.68%	12.87%				
29	-0.2	51.33%	12.58%				
30	-0.1	48.13%	12.29%				
31	0.0	45.10%	12.00%				
32	0.1	42.29%	11.71%				
33	0.2	39.74%	11.42%				
34	0.3	37.50%	11.13%				
35	0.4	35.63%	10.84%				
36	0.5	34.20%	10.55%				
37	0.6	33.26%	10.26%				
38	0.7	32.84%	9.97%				
39	0.8	32.99%	9.68%				
40	0.9	33.67%	9.39%				
41	1.0	34.87%	9.10%				



8.5 Envelope Portfolios

An *envelope portfolio* is the portfolio of risky assets that gives the lowest variance of return of all portfolios having the same expected return. An *efficient portfolio* is the portfolio that gives the highest expected return of all portfolios having the same variance. Mathematically, we may define an envelope portfolio as follows: For a given return $\mu = E(r_p)$, an efficient portfolio $p = [x_1, x_2, \dots, x_N]$ is one that solves

$$\min \sum_i \sum_j x_i x_j \sigma_{ij} = Var(r_p)$$

subject to

$$\sum_i x_i r_i = \mu = E(r_p)$$

$$\sum_i x_i = 1$$

The *envelope* is the set of all envelope portfolios, and the *efficient frontier* is the set of all efficient portfolios.[8] As shown by Black (1972), the envelope is the set of all convex combinations of any two envelope portfolios.[9] This means that if $x = [x_1, x_2, \dots, x_N]$ and $y = [y_1, y_2, \dots, y_N]$ are envelope portfolios and if a is a constant, then the portfolio Z defined by

$$z = ax + (1-a)y = \begin{bmatrix} ax_1 + (1-a)y_1 \\ ax_2 + (1-a)y_2 \\ \vdots \\ ax_N + (1-a)y_N \end{bmatrix}$$

is also an envelope portfolio. Thus, we can compute the whole envelope frontier if we can find any two envelope portfolios.

8. Chapter 9 discusses the difference between envelope and efficient portfolios. In a word: The efficient frontier is a subset of the envelope containing only the optimal portfolios.

9. We discuss Black's theorem more extensively in the next chapter.

By this theorem, once we have found two efficient portfolios x and y, we know that any other efficient portfolio is a convex combination of x and y. If we denote the mean and variance of x and y by $\{E(r_x), \sigma_x^2\}$ and $\{E(r_y), \sigma_y^2\}$, and if $z = ax + (1 - a)y$, then

$$E(r_z) = aE(r_x) + (1-a)E(r_y)$$

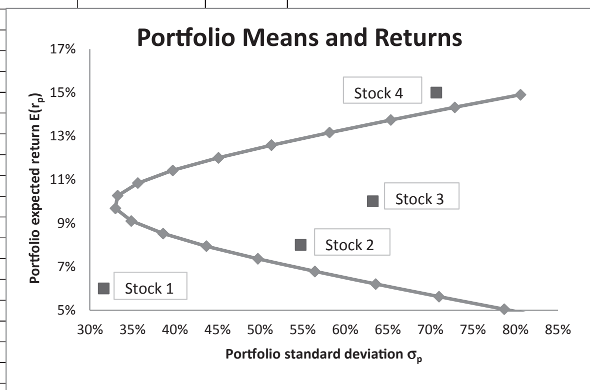
$$\begin{aligned}\sigma_z^2 &= a^2\sigma_x^2 + (1-a)^2\sigma_y^2 + 2a(1-a)Cov(x, y) \\ &= a^2\sigma_x^2 + (1-a)^2\sigma_y^2 + 2a(1-a)x^T S\, y\end{aligned}$$

Further details of the calculation of efficient portfolios are discussed in Chapter 9.

Portfolios *x* and *y* Are Not on the Envelope

To show that the concepts of envelope and efficient portfolio is non-trivial, we show that the two portfolios whose combinations are graphed in the previous example are not either envelope or efficient. This is easy to see if we extend the data table to include numbers for the individual stocks:

	A	B	C	D	E	F	G
1	**A FOUR-ASSET PORTFOLIO PROBLEM**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Portfolio x	0.20	0.30	0.40	0.10		
9	Portfolio y	0.20	0.10	0.10	0.60		
10							
11	**Portfolio x and y statistics: Mean, variance, covariance, correlation**						
12	Mean, $E(r_x)$	9.10%		Mean, $E(r_y)$	12.00%	<-- {=MMULT(B9:E9,F3:F6)}	
13	Variance σ_x^2	0.1216		Variance σ_y^2	0.2034	<--	
14	Covariance(x,y)	0.0714				{=MMULT(MMULT(B9:E9,A3:D6),TRANSPOSE(B9:E9))}	
15	Correlation ρ_{xy}	0.4540	<-- =B14/SQRT(B13*E13)				
16							
17	**Calculating returns of combinations of Portfolio x and Portfolio y**						
18	Proportion of x	0.3					
19	Mean portfolio return, $E(r_p)$	11.13%	<-- =B18*B12+(1-B18)*E12				
20	Portfolio variance σ_p^2	0.1406	<-- =B18^2*B13+(1-B18)^2*E13+2*B18*(1-B18)*B14				
21	Portfolio standard deviation σ_p	37.50%	<-- =SQRT(B20)				
22							
23	**Table of returns (using Data Table)**						
24	Proportion of x	Standard deviation	Mean				
25		37.50%	11.13%				
26	-1.0	80.60%	14.90%				
27	-0.8	72.88%	14.32%				
28	-0.6	65.38%	13.74%				
29	-0.4	58.15%	13.16%				
30	-0.2	51.33%	12.58%				
31	0.0	45.10%	12.00%				
32	0.2	39.74%	11.42%				
33	0.4	35.63%	10.84%				
34	0.6	33.26%	10.26%				
35	0.8	32.99%	9.68%				
36	1.0	34.87%	9.10%				
37	1.2	38.60%	8.52%				
38	1.4	43.69%	7.94%				
39	1.6	49.74%	7.36%				
40	1.8	56.44%	6.78%				
41	2.0	63.58%	6.20%				
42	2.2	71.02%	5.62%				
43	2.4	78.69%	5.04%				
44	2.6	86.53%	4.46%				
45	2.7	90.49%	4.17%				
46	Stock1	31.62%		6.00%			
47	Stock2	54.77%		8.00%			
48	Stock3	63.25%		10.00%			
49	Stock4	70.71%		15.00%			



Were the two portfolios on the envelope, then all of the individual stocks would fall on or inside the curved line in the graph. In our case, two of the stock returns (stock 1 and stock 4) fall outside the frontier created by combinations of portfolios *x* and *y*. Thus *x* and *y* cannot be efficient portfolios. In Chapter 9 you will learn to compute efficient and envelope portfolios, and as you will see there, this requires considerably more computation.

8.6 Summary

In this chapter we have reviewed the basic concepts and mathematics of portfolios. In succeeding chapters we shall describe how to compute the variance-covariance matrix from asset returns and how to calculate efficient portfolios.

Exercises

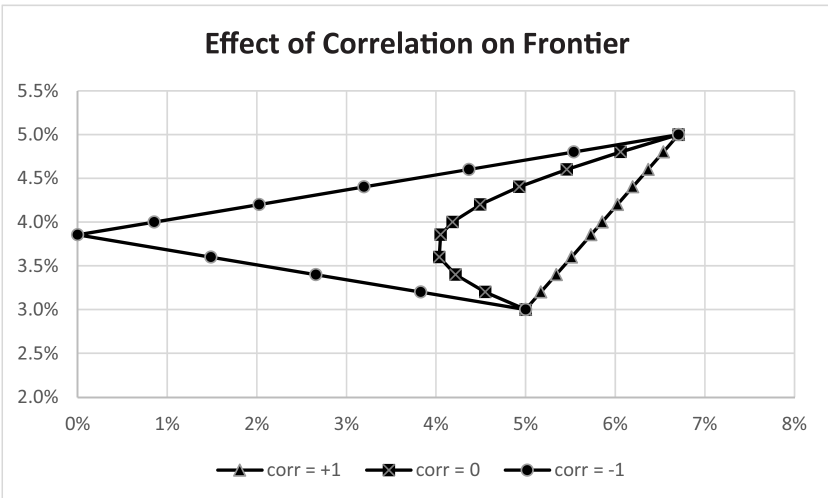
1. The exercise disk for this chapter contains monthly data for stock prices of Kellogg and IBM. Compute the return statistics and graph a frontier of combinations of the two stocks.
2. Consider the two stocks below. Graph the frontier of combinations of the two stocks. Show the effect on the frontier of varying the correlation from −1 to +1.

	A	B	C	D	E	F
1	**TWO STOCKS** **Varying the correlation coefficient**					
2		**Stock A**	**Stock B**			
3	Mean	3.00%	8.00%			
4	Sigma	15.00%	22.00%			
5	Correlation	0.3000				

3. The disk for these exercises gives 5 years of monthly prices for two Vanguard funds—the Vanguard Index 500 fund (symbol VFINX) and the Vanguard High-Yield Corporate Bond fund (VWEHX). The first of these funds tracks the Standard and Poor's 500, and VWEHX is a junk-bond fund. Compute the monthly returns and the frontier of combinations of these two funds.
4. Consider the two random variables *X* and *Y* whose values are given below. Note that *X* and *Y* are perfectly correlated, though perhaps not linearly correlated. Compute their correlation coefficient.

	A	B
1	**X**	**Y**
2	-5	25
3	-4	16
4	-3	9
5	-2	4
6	-1	1
7	0	0
8	1	1
9	2	4
10	3	9
11	4	16
12	5	25

5. Assets A and B have the means and variances indicated in the exercise file. Graph three cases, $\rho_{AB} = -1, 0, +1$ on one set of axes, producing the following chart:



6. Three assets have the following means and variance-covariance matrix:

	A	B	C	D	E	F
1		**Variance-covariance matrix**				**Means**
2		0.30	0.02	-0.05		10%
3		0.02	0.40	0.06		12%
4		-0.05	0.06	0.60		14%
5						
6		**Portfolio1**	**Portfolio2**			
7	Asset1	30%	50%			
8	Asset2	20%	40%			
9	Asset3	50%	10%			

a. Calculate the statistics—mean, variance, standard deviation, covariance, correlation—for the portfolios.

b. Create a chart of the mean and standard deviation of combinations of the portfolios.

c. Add the individual asset returns to the chart—are the two portfolios on the efficient frontier?

7. Consider the data below and find the portfolio weights so that the expected return of the portfolio is 14%. What is the corresponding portfolio standard deviation?

	A	B	C
1		**Mean return**	**Standard deviation of return**
2	Stock 1	12%	35%
3	Stock 2	18%	50%
4	Covariance(r_1,r_2)	0.08350	

8. In the previous problem find two portfolios whose standard deviation is 45%. (There is an analytical solution to this problem, but it can also be solved by **Solver**.)

Appendix 8.1: Adjusting for Dividends

When downloading data from Yahoo or other sources, the "adjusted price" includes an adjustment for dividends. In this appendix we discuss two ways of making this adjustment.[10] The first, and simplest, method of adjusting for dividends is to add them to the annual change in price. In the following

10. It might be argued that since the free sources available on the web make these adjustments automatically, the details in this appendix are superfluous. Nevertheless we think they offer some interesting insights. (If you disagree, turn the page!)

example, if you purchased GM stock at the 1986 year-end price of $33 per share and held it for 1 year, you would, at the end of the year, have made 0.57%.

$$\textit{Discretely compounded return}, 1987 = \frac{30.69 + 2.50}{33.00} - 1 = 0.568\%$$

The continuously compounded return is calculated by:

$$\textit{Continuously compounded return}, 1987 = \ln\left[\frac{30.69 + 2.50}{33.00}\right] = 0.567\%$$

(The choice between discrete and continuous compounding is discussed in Appendix 8.2.)

	A	B	C	D	E	F
1	**GENERAL MOTORS (GM) STOCK ADJUSTING FOR DIVIDENDS**					
2	**Year**	**Share price at end year**	**Dividend per share**	**Discretely compounded return**	**Continuously compounded return**	
3	1986	33.00				=(B4+C4)/B3-1
4	1987	30.69	2.50	0.57%	0.57%	
5	1988	41.75	2.50	44.20%	36.60%	
6	1989	42.25	3.00	8.38%	8.05%	<-- =LN((C4+B4)/B3)
7	1990	34.38	3.00	-11.54%	-12.26%	
8	1991	28.88	1.60	-11.35%	-12.04%	
9	1992	32.25	1.40	16.54%	15.30%	
10	1993	54.88	0.80	72.64%	54.60%	
11	1994	42.13	0.80	-21.78%	-24.56%	
12	1995	52.88	1.10	28.13%	24.79%	
13	1996	55.75	1.60	8.46%	8.12%	
14						
15	**Arithmetic annual return**			13.43%	9.92%	<-- =AVERAGE(E4:E13)
16	**Standard deviation of returns**			27.15%	22.84%	<-- =STDEVP(E4:E13)

Dividend Reinvestment

Another way of calculating returns is to assume that the dividends are reinvested in the stock:

	G	H	I	J	K	L	M	N
1	**GM: REINVESTING THE DIVIDENDS IN SHARES**							
2	**Year**	**Effective shares held at beginning of year**	**Share price at end year**	**Dividend per share**	**Total dividends received**	**Number of shares at end of year**	**Value of shares at end of year**	
3	1986		33.00				33.000	=H5+K5/I5
4	1987	1.00	30.69	2.500	2.500	1.081	33.188	
5	1988	1.08	41.75	2.500	2.704	1.146	47.855	<-- =L5*I5
6	1989	1.15	42.25	3.000	3.439	1.228	51.867	
7	1990	1.23	34.38	3.000	3.683	1.335	45.882	
8	1991	1.33	28.88	1.600	2.136	1.409	40.677	
9	1992	1.41	32.25	1.400	1.972	1.470	47.403	
10	1993	1.47	54.88	0.800	1.176	1.491	81.835	
11	1994	1.49	42.13	0.800	1.193	1.520	64.014	
12	1995	1.52	52.88	1.100	1.672	1.551	82.021	
13	1996	1.55	55.75	1.600	2.482	1.596	88.963	
14								
15		**Annualized continous return**					9.92%	<-- =LN(M13/M3)/10
16		**Compound geometric return**					10.43%	<-- =(M13/M3)^(1/10)-1
17								
18				=H5*J5				

Consider first 1987: Since we purchased the share at the end of 1986, we own 1 share at the end of 1987. If the 1987 dividend is turned into shares at the end-1987 price, we can use it to buy 0.081 additional shares:

$$\textit{New shares purchased at end}\text{-}1987 = \frac{2.50}{30.69} = 0.081$$

Thus we start 1988 with 1.081 shares. Since the 1988 dividend per share is $2.50, the total dividend received on the shares is 1.081*$2.50 = $2.704. Reinvesting these dividends in shares gives

$$\textit{New shares purchased at end}\text{-}1988 = \frac{2.704}{41.75} = 0.065$$

Thus, at the end of 1988, the holder of GM shares will have accumulated 1 + 0.081 + 0.065 = 1.146 shares.

As the spreadsheet fragment above shows, this reinvestment of dividends will produce a holding of 1.596 shares at the end of 1996, worth $88.963.

We can calculate the return on this investment in one of two ways:

$$\textit{Continuously compounded return} = \ln\left[\frac{\textit{end}-1996\ \textit{value}}{\textit{beginning investment}}\right]\Big/10$$

$$= \ln\left[\frac{88.963}{33.00}\right]\Big/10 = 9.92\%$$

Note that this continuously compounded return (the method preferred in this book) is the same as that calculated in the first spreadsheet fragment in this appendix from the annual returns (cell E15).

An alternative is to calculate the geometric return:

$$\textit{Compound geometric return} = \left[\frac{\textit{end} - 1996 \textit{ value}}{\textit{initial investment}}\right]^{1/10} - 1$$

$$= \left[\frac{88.963}{33.00}\right]^{1/10} - 1 = 10.43\%$$

Appendix 8.2: Continuously Compounded Versus Geometric Returns

Using the continuously compounded return assumes that $P_t = P_{t-1}e^{r_t}$, where r_t is the rate of return during the period $(t - 1, t)$. Suppose that $r_1, r_2, \ldots, r_{12}$ are the returns for 12 periods (a period could be a month or it could be a year), then the price of the stock at the end of the 12 periods will be:

$$P_{12} = P_0 e^{r_1 + r_2 + \ldots + r_{12}}$$

This representation of prices and returns allows us to assume that the *average periodic return* is $r = (r_1 + r_2 + \ldots + r_{12})/12$. Since we wish to assume that the return data for the 12 periods represent the distribution of the returns for the coming period, it follows that the continuously compounded return is the appropriate return measure, and not the discretely compounded return $r_t = (P_{At} - P_{A,t-1})/P_{A,t-1}$.

How Different Are Continuously Compounded and Discretely Compounded Returns?

The continuously compounded return will always be smaller than the discretely compounded return, but the difference is usually not large. The following table shows the differences for the example in section 8.2:

	A	B	C	D	E	F	G	H
1	**APPLE AND GOOGLE COMPARING CONTINUOUS AND DISCRETE RETURNS**							
2								
3	**Date**	**AAPL price**	**Continuous return**	**Discrete return**		**Google price**	**Continuous return**	**Discrete return**
4	1-Jun-07	122.04				580.11		
5	2-Jul-07	131.76	7.66%	7.96%	<-- =B5/B4-1	645.90	10.74%	11.34%
6	1-Aug-07	138.48	4.97%	5.10%		599.39	-7.47%	-7.20%
59	3-Jan-12	456.48	11.97%	12.71%		522.70	2.46%	2.49%
60	1-Feb-12	542.44	17.25%	18.83%		497.91	-4.86%	-4.74%
61	1-Mar-12	599.55	10.01%	10.53%		471.38	-5.48%	-5.33%
62	2-Apr-12	583.98	-2.63%	-2.60%		458.16	-2.84%	-2.80%
63	1-May-12	577.73	-1.08%	-1.07%		449.45	-1.92%	-1.90%
64	1-Jun-12	584.00	1.08%	1.09%		501.50	10.96%	11.58%
65								
66	Monthly average		2.61%	3.24%	=AVERAGE(G5:G64) -->		-0.24%	0.26%
67	Monthly standard deviation		11.17%	10.69%	=STDEV.S(G5:G64) -->		10.09%	10.07%
68								
69	Annual average		31.31%		=12*G66 -->		-2.91%	
70	Annual variance		134.04%		=12*G67 -->		121.05%	
71	Annual standard deviation		115.78%		=SQRT(G70) -->		110.02%	

Calculating Annual Returns and Variances from Periodic Returns

Suppose we calculate a series of continuously compounded *monthly* rates of return $r_1, r_2, \ldots, r_n$ and we wish to then calculate the mean and the variance of the *annual* rate of return. Clearly the mean annual return is given by:

$$Mean\ annual\ return = 12\left[\frac{1}{n}\sum_{t=1}^{n} r_t\right]$$

To calculate the variance of the annual rate of return, we assume that the monthly rates of return are independent identically distributed random variables. If we use the continuously compounded returns, it then follows that $Var(r) = 12\left[\frac{1}{n}\sum_{t=1}^{n} Var(r_t)\right] = 12\sigma_{monthly}^2$, and that the standard deviation of the annual rate of return is given by $\sigma = \sqrt{12}\sigma_{monthly}$.[11]

11. Note that this is not true for discretely computed returns. Thus the computation of the variance and the standard deviation are simpler when using continuous compounding.

9 Calculating Efficient Portfolios

9.1 Overview

This chapter covers the theory and calculations necessary for both versions of the classical capital asset pricing model (CAPM)—both that which is based on a risk-free asset (also known as the Sharp-Lintner-Mossin model) and Black's (1972) zero-beta CAPM (which does not require the assumption of a risk-free asset). You will find that using a spreadsheet enables you to do the necessary calculations easily.

The structure of the chapter is as follows: We begin with some preliminary definitions and notation. We then state the major results (proofs are given in the appendix to the chapter). In succeeding sections we implement these results, showing you:

- How to calculate efficient portfolios.
- How to calculate the efficient frontier.

This chapter includes more theoretical material than most chapters in this book: Section 9.2 contains the propositions on portfolios which underlie the calculations of both efficient portfolios and the security market line (SML) in Chapter 11. If you find the theoretical material in section 9.2 difficult, skip it at first and try to follow the illustrative calculations in section 9.3. This chapter assumes that the variance-covariance matrix is given; we delay a discussion of various methods of computing the variance-covariance matrix until Chapter 10.

9.2 Some Preliminary Definitions and Notation

Throughout this chapter we use the following notation: There are N risky assets, each of which has expected return $E(r_i)$. The matrix $E(r)$ is the column vector of expected returns of these assets:

$$E(r) = \begin{bmatrix} E(r_1) \\ E(r_2) \\ \vdots \\ E(r_N) \end{bmatrix}$$

and S is the $N \times N$ variance-covariance matrix:

$$S = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \cdots & \sigma_{N1} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{N2} \\ \vdots & & & \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_{NN} \end{bmatrix}$$

A *portfolio of risky assets* (when our intention is clear, we shall just use the word *portfolio*) is a column vector x whose coordinates sum to 1:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad \sum_{i=1}^{N} x_i = 1$$

Each coordinate x_i represents the proportion of the portfolio invested in risky asset i.

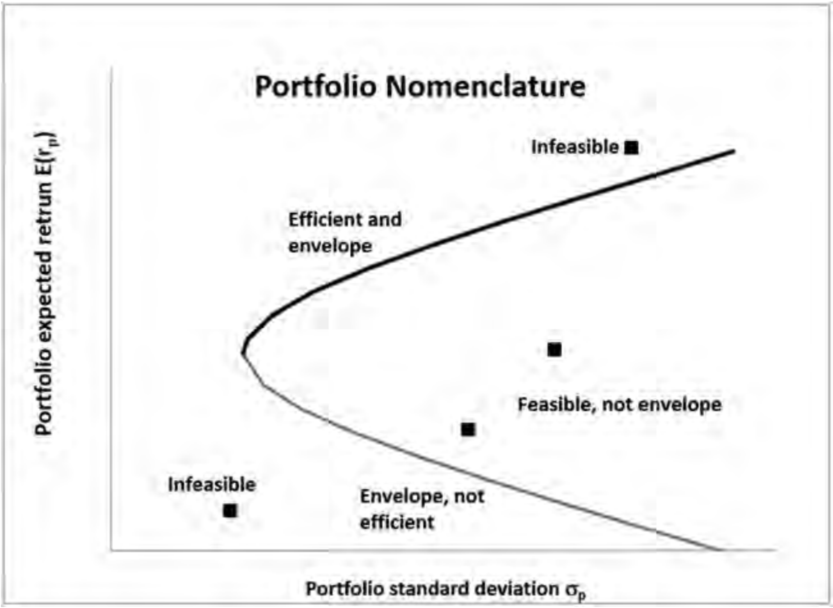
The *expected portfolio return* $E(r_x)$ of a portfolio x is given by the product of x and R:

$$E(r_x) = x^T \cdot R \equiv \sum_{i=1}^{N} x_i E(r_i).$$

The *variance of portfolio* x's *return*, $\sigma_x^2 \equiv \sigma_{xx}$, is given by the product $x^T Sx = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j \sigma_{ij}$.

The *covariance between the return of two portfolios x and y*, $Cov(r_x, ry)$, is defined by the product $\sigma_{xy} = x^T Sy = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i y_j \sigma_{ij}$. Note that $\sigma_{xy} = \sigma_{yx}$.

The following graph illustrates four concepts. A *feasible* portfolio is any portfolio whose proportions sum to 1. The *feasible set* is the set of portfolio means and standard deviations generated by the feasible portfolios; this feasible set is the area inside and to the right of the curved line. A feasible portfolio is on the *envelope* of the feasible set if for a given mean return it has minimum variance. Finally, a portfolio x is *an efficient portfolio* if it maximizes the return given the portfolio variance (or standard deviation). That is: x is efficient if there is no other portfolio y such that $E(R_y) > E(R_x)$ and $\sigma_y < \sigma_x$. The set of all efficient portfolios is called the *efficient frontier*; this frontier is the heavier line in the graph.



9.3 Five Propositions on Efficient Portfolios and the CAPM

In the appendix to this chapter we prove the following results, which are basic to the calculations of the CAPM. All of these propositions are used in deriving the efficient frontier and the security market line; numerical illustrations are given in the next section and in succeeding chapters.

PROPOSITION 1 Let c be a constant. We use the notation $E(r) - c$ to denote the following column vector:

$$E(r) - c = \begin{bmatrix} E(r_1) - c \\ E(r_2) - c \\ \vdots \\ E(r_N) - c \end{bmatrix}$$

Let the vector z solve the system of simultaneous linear equations $E(r) - c = Sz$. Then this solution produces a portfolio x on the envelope of the feasible set in the following manner:

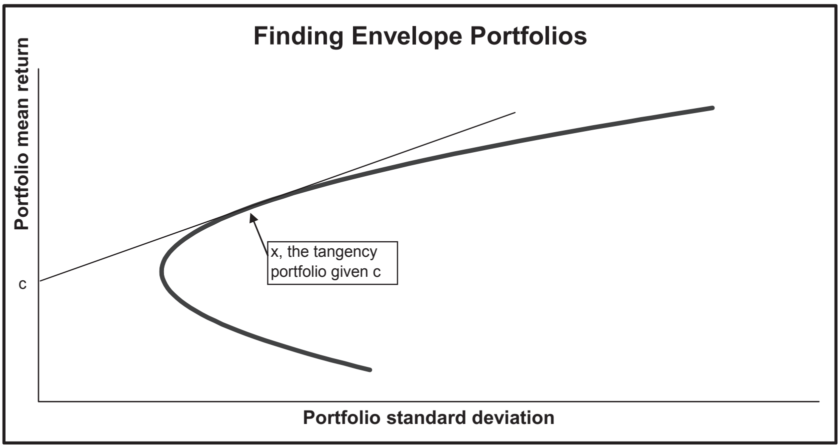
$$z = S^{-1}\{E(r) - c\}$$
$$x = \{x_1, \ldots, x_N\},$$

where

$$x_i = \frac{z_i}{\sum_{j=1}^{N} z_j}$$

Furthermore, all envelope portfolios are of this form.

Intuition A formal proof of the proposition is given in the appendix to this chapter, but the intuition is simple and geometric. Suppose we pick a constant c and we try to find an efficient portfolio x for which there is a tangency between c and the feasible set:



Proposition 1 gives a procedure for finding x; furthermore, the proposition states that all envelope portfolios (in particular, all efficient portfolios) are the result of the procedure outlined in the proposition. That is, if x is any envelope portfolio, then there exists a constant c and a vector z such that $Sz = E(r) - c$ and $x = z \Big/ \sum_i z_i$.

PROPOSITION 2 By a theorem first proved by Black (1972), any two envelope portfolios are enough to establish the whole envelope. Given any two envelope portfolios $x = \{x_1, \ldots, x_N\}$ and $y = \{y_1, \ldots, y_N\}$, all envelope portfolios are convex combinations of x and y. This means that given any constant a, the portfolio

$$ax+(1-a)y=\begin{bmatrix} ax_1+(1-a)y_1 \\ ax_2+(1-a)y_2 \\ \vdots \\ ax_N+(1-a)y_N \end{bmatrix}$$

is an envelope portfolio.

PROPOSITION 3 If y is any envelope portfolio, then for any other portfolio (envelope or not) x, we have the relationship:

$$E(r_x)=c+\beta_x\left[E(r_y)-c\right]$$

where

$$\beta_x=\frac{Cov(x,y)}{\sigma_y^2}$$

Furthermore, c is the expected return of all portfolios z whose covariance with y is zero:

$$c=E(r_z),\text{ where } Cov(y,z)=0$$

Notes If y is on the envelope, the regression of any and all portfolios x on y gives a linear relationship. In this version of the CAPM (usually known as "Black's zero-beta CAPM," in honor of Fischer Black, whose 1972 paper proved this result) the Sharpe-Lintner-Mossin security market line (SML) is replaced with an SML in which the role of the risk-free asset is played by a portfolio with a zero beta with respect to the particular envelope portfolio y. Note that this result is true for any envelope portfolio y.

The converse of Proposition 3 is also true:

PROPOSITION 4 Suppose that there exists a portfolio y such that for any portfolio x the following relation holds:

$$E(r_x)=c+\beta_x\left[E(r_y)-c\right]$$

where

$$\beta_x=\frac{Cov(x,y)}{\sigma_y^2}$$

Then the portfolio y is an envelope portfolio.

Propositions 3 and 4 show that *an SML relation holds if and only if we regress all portfolio returns on an envelope portfolio with an* $R^2 = 100\%$. As Roll (1977, 1978) has forcefully pointed out, these propositions show that it is not enough to run a test of the CAPM by showing that the SML holds.[1] The only real test of the CAPM is *whether the true market portfolio is mean-variance efficient*. We shall return to this topic in Chapter 10.

THE MARKET PORTFOLIO The *market portfolio M* is a portfolio composed of *all the risky assets in the economy*, with each asset taken in proportion to its value. To make this more specific: Suppose that there are N risky assets and that the market value of asset i is V_i. Then the market portfolio has the following weights:

$$Proportion\ of\ asset\ i\ in\ M = \frac{V_i}{\sum_{h=1}^{N} V_h}$$

If the market portfolio M is efficient (this is a big "if" as we shall see in Chapters 11 and 13, Proposition 3 is also true for the market portfolio. That is, the SML holds with $E(r_z)$ substituted for c:

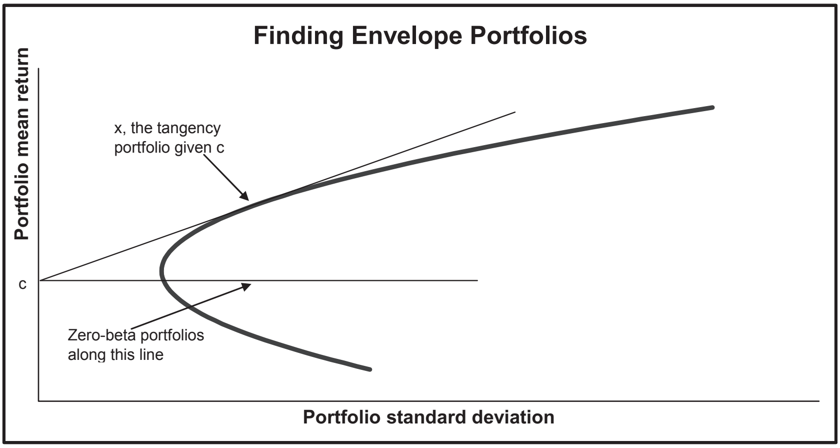
$$E(r_x) = E(r_z) + \beta_x [E(r_M) - E(r_z)]$$

where

$$\beta_x = \frac{Cov(x, M)}{\sigma_M^2} \quad \text{and} \quad Cov(z, M) = 0$$

This version of the SML has received the most empirical attention of all of the CAPM results. In Chapter 11 we show how to calculate β and how to calculate the SML; we go on to examine Roll's criticism of these empirical tests. From the following graph, it is easy to see how to locate a zero-beta portfolio on the envelope of the feasible set:

1. Roll's 1977 paper is more often cited and more comprehensive, but his 1978 paper is much easier to read and intuitive. If you're interested in this literature, start there.



When there is a risk-free asset, Proposition 3 specializes to the security market line of the classic capital asset pricing model:

PROPOSITION 5 If there exists a risk-free asset with return r_f, then there exists an envelope portfolio M such that:

$$E(r_x) = r_f + \beta_x [E(r_M) - r_f]$$

where

$$\beta_x = \frac{Cov(x, M)}{\sigma_M^2}$$

As shown in the classic papers by Sharpe (1964), Lintner (1965), and Mossin (1966), if all investors choose their portfolios only on the basis of portfolio mean and standard deviation, then the portfolio x of Proposition 5 is the market portfolio M.

In the remainder of this chapter, we explore the meaning of these propositions using numerical examples worked out on Excel.

9.4 Calculating the Efficient Frontier: An Example

In this section we calculate the efficient frontier using Excel. We consider a world with four risky assets having the following expected returns and variance-covariance matrix:

	A	B	C	D	E	F	G	H
1	**CALCULATING THE FRONTIER**							
2	**Variance-covariance, S**					**Mean returns E(r)**	**E(r) minus constant**	
3	0.10	0.01	0.03	0.05		6%	2.00%	<-- =F3-$B$8
4	0.01	0.30	0.06	-0.04		8%	4.00%	
5	0.03	0.06	0.40	0.02		10%	6.00%	
6	0.05	-0.04	0.02	0.50		15%	11.00%	
7								
8	Constant, c	4.00%						

Each cell of the column vector labeled **E(r) minus constant** contains the mean return of the given asset minus the value of the constant c (in this case c = 4%). We use this column in finding the second envelope portfolio below.

We separate our calculations into two parts: In the next subsection we calculate two portfolios on the envelope of the feasible set. In the subsequent subsection we calculate the efficient frontier.

Calculating Two Envelope Portfolios

By Proposition 2, we have to find two efficient portfolios in order to identify the whole efficient frontier. By Proposition 1 each envelope portfolio solves the system $R - c = Sz$ for z. To identify two efficient portfolios, we use two different values for c. For each value of c, we solve for z and then set $x_i = z_i \Big/ \sum_h z_h$ to find an efficient portfolio.

The c's we solve for are arbitrary (see section 9.6), but to make life easy, we first solve this system for $c = 0$. This gives the following results:

	A	B	C	D	E	F	G	H
10	**Computing an envelope portfolio with constant = 0**							
11	z					Envelope portfolio x		
12	0.3861	<-- {=MMULT(MINVERSE(A3:D6),F3:F6)}				0.3553	<-- =A12/SUM($A$12:$A$15)	
13	0.2567					0.2362		
14	0.1688					0.1553		
15	0.2752					0.2532		
16					Sum	1.0000	<-- =SUM(F12:F15)	

The formulas in the cells are:

- For z we use the array function **MMult(MInverse(A3:D6),F3:F6)**. The range A3:D6 contains the variance-covariance matrix and the cells F3:F6 contain the expected returns of the assets.
- For x: Each cell contains the associated value of z divided by the sum of all the z's. Thus, for example, cell F12 contains the formula =**A12/SUM($A$12:$A$15)**.

To find the second envelope portfolio we now solve this system for $c = 0.04$ (cell B8).

	A	B	C	D	E	F	G	H
18	**Computing an envelope portfolio with constant = 4.00%**							
19	**z**					**Envelope portfolio y**		
20	0.0404	<-- {=MMULT(MINVERSE(A3:D6),G3:G6)}				0.0782	<-- =A20/SUM($A$20:$A$23)	
21	0.1386					0.2684		
22	0.1151					0.2227		
23	0.2224					0.4307		
24					Sum	1.0000	<-- =SUM(F20:F23)	

The portfolio y in cells F20:F23 is, by the results of Proposition 1, an envelope portfolio. This vector z associated with y is calculated in a manner similar to that of the first vector, except that the array function in the cells is **MMult(MInverse(A3:D6),G3:G6)**, where G3:G6 contains the vector of expected returns minus the constant 0.04.

To complete the basic calculations, we compute the means, standard deviations, and covariance of returns for portfolios x and y:

	A	B	C	D	E	F	G	H	I	J	K
26	E(x)	9.37%			E(y)	11.30%	<-- {=MMULT(TRANSPOSE(F20:F23),F3:F6)}				
27	Var(x)	0.0862			Var(y)	0.1414	<-- {=MMULT(MMULT(TRANSPOSE(F20:F23),A3:D6),F20:F23)}				
28	Sigma(x)	29.37%			Sigma(y)	37.60%	<-- =SQRT(F27)				
29											
30	Cov(x,y)	0.1040	<-- {=MMULT(MMULT(TRANSPOSE(F12:F15),A3:D6),F20:F23)}								
31	Corr(x,y)	0.9419	<-- =B30/(B28*F28)								

The transpose vectors of x and of y are inserted using the array function **Transpose** (see Chapter 34 for a discussion of array functions). This now enables us to calculate the mean, variance, and covariance as follows:

E(x) uses the array formula **MMult(transpose_x,means)**. Note that we could have also used the function **SumProduct(x,means)**.

Var(x) uses the array formula **MMult(MMult(transpose_x, var_cov),x)**.

Sigma(x) uses the formula **Sqrt(var_x)**.

Cov(x,y) uses the array formula **MMult(MMult(transpose_x,var_cov),y)**.

Corr(x,y) uses the formula **cov(x,y)/(sigma_x*sigma_y)**.

The following spreadsheet illustrates everything that has been done in this subsection:

	A	B	C	D	E	F	G	H
1	**CALCULATING THE FRONTIER**							
2	**Variance-covariance, S**					**Mean returns E(r)**	**E(r) minus constant**	
3	0.10	0.01	0.03	0.05		6%	2.00%	<-- =F3-$B$8
4	0.01	0.30	0.06	-0.04		8%	4.00%	
5	0.03	0.06	0.40	0.02		10%	6.00%	
6	0.05	-0.04	0.02	0.50		15%	11.00%	
7								
8	Constant, c	4.00%						
9								
10	**Computing an envelope portfolio with constant = 0**							
11	**z**					**Envelope portfolio x**		
12	0.3861	<-- {=MMULT(MINVERSE(A3:D6),F3:F6)}				0.3553	<-- =A12/SUM($A$12:$A$15)	
13	0.2567					0.2362		
14	0.1688					0.1553		
15	0.2752					0.2532		
16					Sum	1.0000	<-- =SUM(F12:F15)	
17								
18	**Computing an envelope portfolio with constant = 4.00%**							
19	**z**					**Envelope portfolio y**		
20	0.0404	<-- {=MMULT(MINVERSE(A3:D6),G3:G6)}				0.0782	<-- =A20/SUM($A$20:$A$23)	
21	0.1386					0.2684		
22	0.1151					0.2227		
23	0.2224					0.4307		
24					Sum	1.0000	<-- =SUM(F20:F23)	
25								
26	E(x)	9.37%			E(y)	11.30%	<-- {=MMULT(TRANSPOSE(F2	
27	Var(x)	0.0862			Var(y)	0.1414	<-- {=MMULT(MMULT(TRANSP	
28	Sigma(x)	29.37%			Sigma(y)	37.60%	<-- =SQRT(F27)	
29								
30	Cov(x,y)	0.1040	<-- {=MMULT(MMULT(TRANSPOSE(F12:F15),A3:D6),F20:F23)}					
31	Corr(x,y)	0.9419	<-- =B30/(B28*F28)					

Calculating the Envelope

By Proposition 2 of section 9.3, convex combinations of the two portfolios calculated in the previous subsection allow us to calculate the whole envelope of the feasible set (which, of course, includes the efficient frontier). Suppose we let p be a portfolio that has proportion a invested in portfolio x and proportion (1 − *a*) invested in *y*. Then—as discussed in Chapter 8—the mean and standard deviation of *p*'s return are:

$$E(r_p) = aE(r_x) + (1-a)E(r_y)$$

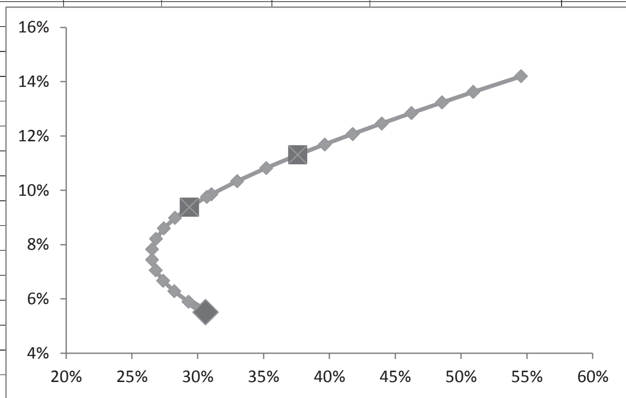
$$\sigma_p = \sqrt{a^2\sigma_x^2 + (1-a)^2\sigma_y^2 + 2a(1-a)Cov(x, y)}$$

Here's a sample calculation for our two portfolios:

	A	B	C	D	E	F	G
34	**A single portfolio calculation**						
35	Proportion of	0.3					
36	$E(r_p)$	10.72%	<-- =B35*B26+(1-B35)*F26				
37	σ_p^2	0.1207	<-- =B35^2*B27+(1-B35)^2*F27+2*B35*(1-B35)*B30				
38	σ_p	34.75%	<-- =SQRT(B37)				

We can turn this calculation into a data table (see Chapter 31) to get the following table:

	A	B	C	D	E	F	G	H	I
34	**A single portfolio calculation**								
35	Proportion of x	0.3							
36	$E(r_p)$	10.72%	<-- =B35*B26+(1-B35)*F26						
37	σ_p^2	0.1207	<-- =B35^2*B27+(1-B35)^2*F27+2*B35*(1-B35)*B30						
38	σ_p	34.75%	<-- =SQRT(B37)						
39									
40	**Data table: We vary the proportion of x to produce a graph of the frontier**								
41	**Proportion of x**	**Sigma**	**Return**						
42		34.75%	10.72%	<-- =B36, data table header					
43	-1.5	54.56%	14.20%						
44	-1.2	50.93%	13.62%						
45	-1.0	48.56%	13.23%						
46	-0.8	46.24%	12.85%						
47	-0.6	43.97%	12.46%						
48	-0.4	41.77%	12.08%						
49	-0.2	39.64%	11.69%						
50	0.0	37.60%	11.30%						
51	0.3	35.20%	10.82%						
52	0.5	33.00%	10.34%						
53	0.8	31.04%	9.86%						
54	0.8	30.68%	9.76%						
55	1.0	29.37%	9.37%						
56	1.2	28.27%	8.99%						
57	1.4	27.42%	8.60%						
58	1.6	26.83%	8.21%						
59	1.8	26.53%	7.83%						
60	2.0	26.52%	7.44%						
61	2.2	26.80%	7.06%						
62	2.4	27.37%	6.67%						
63	2.6	28.21%	6.28%						
64	2.8	29.30%	5.90%						
65	3.0	30.60%	5.51%						



The two portfolios, *x* and *y*, whose convex combinations compose the envelope are marked. Also marked are other portfolios, some of which contain short positions of either *x* or *y*. Note that the convex combinations all lie on the envelope, but may not necessarily be efficient. An example is the last point in the data table—300% in *x* and –200% in portfolio *y*. Thus, while every efficient portfolio is a convex combination of any two efficient portfolios, it is *not true* that every convex combination of any two efficient portfolios is efficient.

9.5 Finding Efficient Portfolios in One Step

The examples in section 9.4 find efficient portfolios by writing out most of the components of the portfolio separately on the spreadsheet. However, for some uses we will want to calculate the efficient portfolio in one step. Here's an example:

	A	B	C	D	E	F	G
1	**FINDING ENVELOPE PORTFOLIOS IN ONE STEP**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Constant	4%					
9							
10	**Envelope portfolio**						
11	0.0782						
12	0.2684	<-- {=MMULT(MINVERSE(A3:D6),F3:F6-					
13	0.2227	B8)/SUM(MMULT(MINVERSE(A3:D6),F3:F6-B8))}					
14	0.4307						
15							
16	Portfolio mean	11.30%	<-- =SUMPRODUCT(A11:A14,F3:F6)				
17	Portfolio sigma	37.60%	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(A11:A14),A3:D6),A11:A14))}				

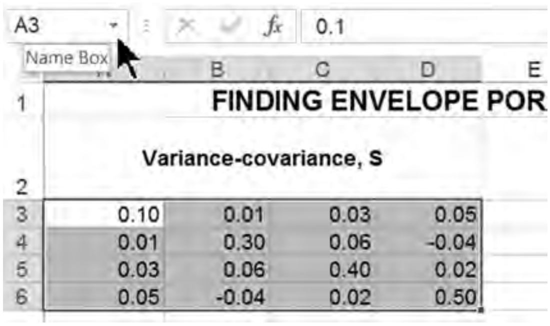
This approach requires a number of Excel tricks, most of which relate to the correct use of array functions. The end result is that we can write the Proposition 1 expression for an envelope portfolio, $x = \frac{S^{-1}\{E(r)-c\}}{Sum\left[S^{-1}\{E(r)-c\}\right]}$, in one cell:

- In cells A11:A14 we have used the array formula F3:F6-B8 to indicate the expected returns minus the constant in cell B8.
- In these same cells we have used **SUM(MMult(MInverse(A3:D6),F3:F6-B8))** to give the denominator of the expression

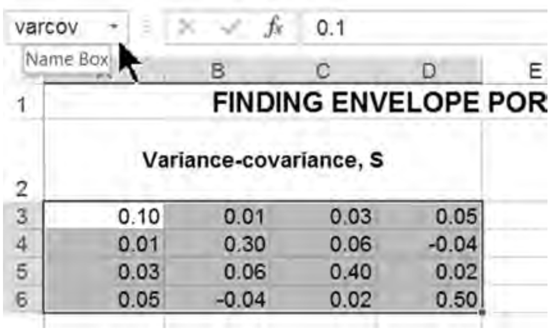
$$x = \frac{S^{-1}\{E(r)-c\}}{Sum\left[S^{-1}\{E(r)-c\}\right]}.$$

Using Cell Names for Clarity

We can make the whole process even clearer by using cell names. To define a cell name, simply mark the cell or a range of cells and go to the **Name Box**, as shown below:



You can now go into the box and type a name:



The name can now be used, as illustrated below:

	A	B	C	D	E	F	G
1	**FINDING ENVELOPE PORTFOLIOS IN ONE STEP**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Constant	4%					
9							
10	**Envelope portfolio**						
11	0.0782						
12	0.2684	<-- {=MMULT(MINVERSE(varcov),means-B8)/SUM(MMULT(MINVERSE(varcov),means-B8))}					
13	0.2227						
14	0.4307						
15							
16	Portfolio mean	11.30%	<-- =SUMPRODUCT(portx,means)				
17	Portfolio sigma	37.60%	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(portx),varcov),portx))}				

9.6 Three Notes on the Optimization Procedure

In this section we note three additional facts about the optimization procedure of Proposition 1, which leads to the computation of envelope portfolios.

Note 1: All Roads Lead to Rome: Envelope Is Determined by Any Two *c*'s

By Proposition 2, the envelope is determined by any two of its portfolios. This means that for the determination of the envelope it is irrelevant which two portfolios we use. To drive home this point, the spreadsheet below computes three envelope portfolios:

- The envelope portfolio x is computed with a constant $c = 0\%$.
- The envelope portfolio y is computed with a constant $c = 4\%$.
- A third envelope portfolio z is computed with a constant $c = 6\%$ (cells D11:D14). As shown in rows 20–26, portfolio z is composed of a convex combination of x and y. This is true for any x, y, and z.

This little exercise shows that the constants c which determine the envelope are completely arbitrary. Any two constants will determine the same envelope.

	A	B	C	D	E	F	G
1	**CALCULATING THE ENVELOPE All constants c lead to the same envelope**						
2	**Variance-covariance, S**					**Mean returns E(r)**	
3	0.10	0.01	0.03	0.05		6%	
4	0.01	0.30	0.06	-0.04		8%	
5	0.03	0.06	0.40	0.02		10%	
6	0.05	-0.04	0.02	0.50		15%	
7							
8	Constant	0%	4%	6%			
9							
10		**Portfolio x**	**Portfolio y**	**Portfolio z**			
11		0.3553	0.0782	-0.5724	<-- {=MMULT(MINVERSE(varcov),means-D8)/SUM(MMULT(MINVERSE(varcov),means-D8))}		
12		0.2362	0.2684	0.3439			
13		0.1553	0.2227	0.3811			
14		0.2532	0.4307	0.8474			
15							
16	Portfolio mean	9.37%	11.30%	15.84%	<-- =SUMPRODUCT(portfolioz,means)		
17	Portfolio sigma	29.37%	37.60%	65.20%	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(portfolioz),varcov),portfolioz))}		
18							
19							
20	**Show: portfolio z is a linear proportion of portfolio x and portfolio y**						
21	Proportion	-2.34822	<-- =(D11-C11)/(B11-C11)				
22	Check						
23	z1	-0.5724	<-- =$B$21*B11+(1-$B$21)*C11				
24	z2	0.3439	<-- =$B$21*B12+(1-$B$21)*C12				
25	z3	0.3811	<-- =$B$21*B13+(1-$B$21)*C13				
26	z4	0.8474	<-- =$B$21*B14+(1-$B$21)*C14				

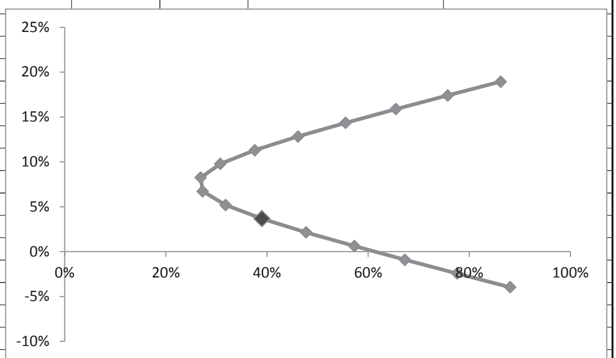
Note 2: Some Values of *c* Locate Non-Efficient Envelope Portfolios

The optimization procedure of Proposition 1 locates a portfolio *x*, which has proportions:

$$x = \frac{S^{-1}\{E(r) - c\}}{Sum\left[S^{-1}\{E(r) - c\}\right]}$$

Although always on the envelope, this portfolio is not necessarily efficient, as is shown in the example below, where a constant $c = 0.11$ leads to an inefficient portfolio.

	A	B	C	D	E	F	G	H
8	Constant	11%	4%					
9								
10		**Portfolio x**	**Portfolio y**					
11		1.1728	0.0782					
12		0.1413	0.2684					
13		-0.0437	0.2227					
14		-0.2704	0.4307					
15								
16	Portfolio mean	3.67%	11.30%					
17	Portfolio sigma	39.01%	37.60%					
18	Cov(x,y)	-0.00631						
19								
20	Single portfolio calculation							
21	Proportion of x	0.6						
22	Mean	6.73%	<-- =B21*B16+(1-B21)*C16					
23	Sigma	27.27%	<-- =SQRT(B21^2*B17^2+(1-B21)^2*C17^2+2*B21*(1-B21)*B18)					
24								
25								
26	**Data table to determine the frontier**							
27	**Proportion of x**	**Sigma**	**Mean**					
28		27.27%	6.73%					
29	-1.0	86.20%	18.93%					
30	-0.8	75.74%	17.41%					
31	-0.6	65.49%	15.88%					
32	-0.4	55.55%	14.36%					
33	-0.2	46.12%	12.83%					
34	0.0	37.60%	11.30%					
35	0.2	30.75%	9.78%					
36	0.4	26.87%	8.25%					
37	0.6	27.27%	6.73%					
38	0.8	31.78%	5.20%					
39	1.0	39.01%	3.67%					
40	1.2	47.73%	2.15%					
41	1.4	57.27%	0.62%					
42	1.6	67.27%	-0.90%					
43	1.8	77.57%	-2.43%					
44	2.0	88.05%	-3.96%					



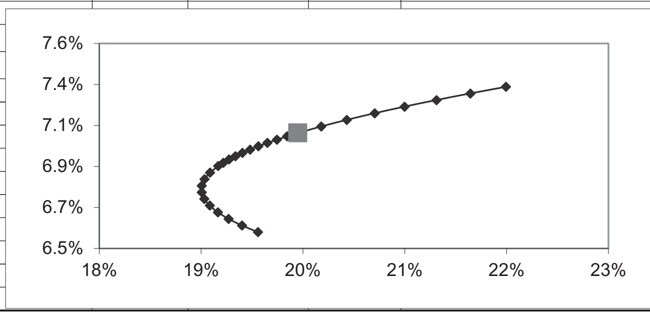
Note 3: The Portfolio Associated with $c = r_f$ Is Optimal

We've said all this before in our discussion of Proposition 1, but it's worth repeating.[2] If we set c to be equal to the risk-free rate of interest, and if the resulting optimizing portfolio $x = \frac{S^{-1}\{E(r)-c\}}{Sum\left[S^{-1}\{E(r)-c\}\right]}$ is efficient, then this portfolio is the optimal investment portfolio for an investor whose preferences are defined solely in terms of the mean and standard deviation of portfolio

2. And it forms the basis of our discussion of the Black-Litterman model in Chapter 13.

returns. In the example below, we assume that $r_f = 4\%$. Locating the optimizing portfolio $x = \dfrac{S^{-1}\{E(r)-c\}}{Sum\left[S^{-1}\{E(r)-c\}\right]}$ on the envelope shows that it is efficient. Therefore, the *optimal investment portfolio* for this case is given by x.

	A	B	C	D	E	F	G
1	**IF c = r_f AND THE OPTIMIZING PORTFOLIO IS EFFICIENT, THEN THE ENVELOPE PORTFOLIO IS OPTIMAL** **The portfolio *x* determined by the constant c = 4% is optimal**						
2	**Variance-covariance matrix**					**Expected returns**	
3	0.40	0.03	0.02	0.00		0.06	
4	0.03	0.20	0.00	-0.06		0.05	
5	0.02	0.00	0.30	0.03		0.07	
6	0.00	-0.06	0.03	0.10		0.08	
7							
8	Constant	0.04					
9							
10	**Computing an envelope portfolio with constant = 0.04**						
11	**z**					**Envelope portfolio x**	
12	0.0330	<-- {=MMULT(MINVERSE(A3:D6),F3:F6-B8)}				0.0423	<-- {=A12:A15/SUM(A12:A15)}
13	0.1959					0.2514	
14	0.0468					0.0601	
15	0.5035					0.6462	
16					Sum	1.0000	<-- =SUM(F12:F15)
17							
18							
19	$E(r_x)$	0.0710					
20	σ_x	0.1995					
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							



9.7 Finding the Market Portfolio: The Capital Market Line (CML)

Suppose a risk-free asset exists, and suppose that this asset has expected return r_f. Let M be the efficient portfolio which is the solution to the system of equations:

$$E(r) - r_f = Sz$$

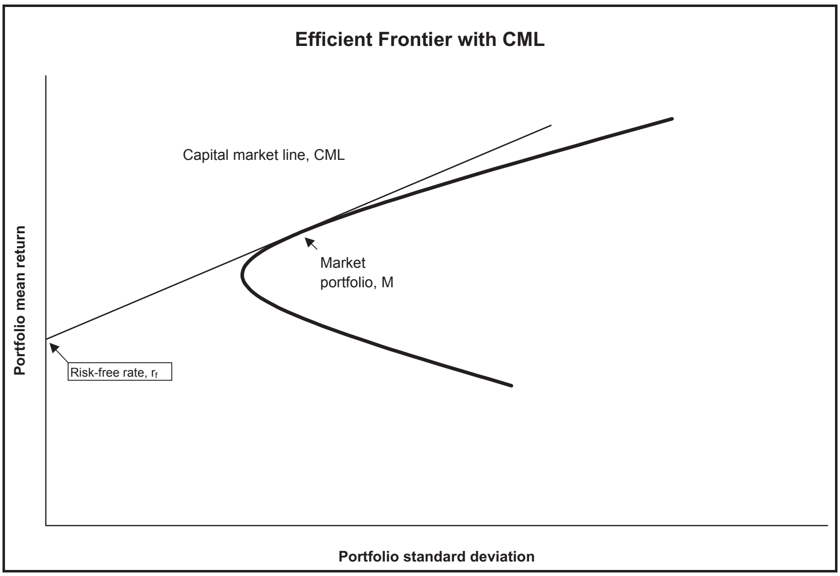
$$M_i = \frac{z_i}{\sum_{i=1}^{N} z_i}$$

Now consider a convex combination of the portfolio M and the risk-free asset r_f; for example, suppose that the weight of the risk-free asset in such a portfolio is a. It follows from the standard equations for portfolio return and σ that:

$$E(r_p) = ar_f + (1-a)E(r_M)$$

$$\sigma_p = \sqrt{a^2\sigma_{r_f}^2 + (1-a)^2\sigma_M^2 + 2a(1-a)Cov(r_f, r_y)} = (1-a)\sigma_M$$

The locus of all such combinations for $a \geq 0$ is known as the *capital market line* (CML). It is graphed below along with the efficient frontier:



The portfolio M is called the *market portfolio* for several reasons:

• Suppose investors agree about the statistical portfolio information (i.e., the vector of expected returns $E(r)$ and the variance-covariance matrix S). Suppose furthermore that investors are interested only in maximizing expected portfolio return given portfolio standard deviation σ. Then it follows that *all optimal portfolios will lie on the CML.*

- In the case above, it further follows that *the portfolio M is the only portfolio of risky assets included in any optimal portfolio*. *M* must therefore include *all the risky assets, with each asset weighted in proportion to its market value*. That is:

$$\textit{weight of risky asset i in portfolio } M = \frac{V_i}{\sum_{i=1}^{N} V_i}$$

where V_i *is the market value of asset i*

It is not difficult to find *M* when we know r_f: We merely have to solve for the efficient portfolio given that the constant $c = r_f$. When r_f changes, we get a different "market" portfolio—this is just the efficient portfolio given a constant of r_f. For example, in our numerical example, suppose that the risk-free rate is $r_f = 5\%$. Then solving the system $E(r) - r_f = Sz$ gives:

	A	B	C	D	E	F	G	H
1	**WHEN c = r$_f$, THE ENVELOPE PORTFOLIO IS THE MARKET PORTFOLIO M**							
2	**Variance-covariance matrix**					**Expected returns E(r)**		
3	0.40	0.03	0.02	0.00		0.06		
4	0.03	0.20	0.00	-0.06		0.05		
5	0.02	0.00	0.30	0.03		0.07		
6	0.00	-0.06	0.03	0.10		0.08		
7								
8	Constant	0.05						
9								
10	**Envelope portfolio is market portfolio M**							
11	0.0314	<-- {=MMULT(MINVERSE(A3:D6),F3:F6-B8)/SUM(MMULT(MINVERSE(A3:D6),F3:F6-B8))}						
12	0.2059							
13	0.0597							
14	0.7031							
15								
16	Portfolio expected return, E(r_M)	7.26%	<-- =SUMPRODUCT(A11:A14,F3:F6)					
17	Portfolio standard devation, σ_M	21.21%	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(A11:A14),A3:D6),A11:A14))}					

9.8 Testing the SML—Implementing Propositions 3–5

To illustrate Propositions 3–5 consider the following data for four risky assets:

	A	B	C	D	E	F
1	ILLUSTRATING PROPOSITIONS 3-5					
2	**Dates**	**Asset 1**	**Asset 2**	**Asset 3**	**Asset 4**	
3	1	-6.63%	-2.49%	-4.27%	11.72%	
4	2	8.53%	2.44%	-3.15%	-8.33%	
5	3	1.79%	4.46%	1.92%	19.18%	
6	4	7.25%	17.90%	-6.53%	-7.41%	
7	5	0.75%	-8.22%	-1.76%	-1.44%	
8	6	-1.57%	0.83%	12.88%	-5.92%	
9	7	-2.10%	5.14%	13.41%	-0.46%	
10						
11	Mean	1.15%	2.87%	1.79%	1.05%	<-- =AVERAGE(E3:E9)

The asset returns on seven dates are given in rows 3–7, and the average return is given in row 11.

We use some sophisticated array functions to compute the variance-covariance matrix:

	A	B	C	D	E	F	G
13	**Variance-covariance matrix**						
14		Asset 1	Asset 2	Asset 3	Asset 4		
15	Asset 1	0.0024	0.0019	-0.0015	-0.0024		
16	Asset 2	0.0019	0.0056	-0.0007	-0.0016	cells B15:E18 contain the formula	
17	Asset 3	-0.0015	-0.0007	0.0057	-0.0005	{=MMULT(TRANSPOSE(B3:E9-B11:E11),B3:E9-	
18	Asset 4	-0.0024	-0.0016	-0.0005	0.0094	B11:E11)/7}	
19							
20	**Finding an efficient portfolio w**						
21	Constant	0.50%					
22							
23	Asset 1	0.3129					
24	Asset 2	0.2464	<--	Cells B23:B26 contain the formula			
25	Asset 3	0.2690		{=MMULT(MINVERSE(B15:E18),TRANSPOSE(B11:E11)-			
26	Asset 4	0.1717		B21)/SUM(MMULT(MINVERSE(B15:E18),TRANSPOSE(B11:E11)-B21))}			

The efficient portfolio given the constant $c = 0.5\%$ is given in cells B23:B26; we compute this portfolio using the method of Proposition 1.[3] We call this portfolio w. The returns of portfolio w on dates 1–7 are given in column G below:

3. Following the discussion in section 9.6, a careful reader will recall that Proposition 1 only guarantees that this portfolio is on the envelope. But it is, in fact, efficient.

	A	B	C	D	E	F	G	H
2	**Dates**	**Asset 1**	**Asset 2**	**Asset 3**	**Asset 4**		**Efficient portfolio w**	
3	1	-6.63%	-2.49%	-4.27%	11.72%		-1.82%	<-- {=MMULT(B3:E9,B23:B26)}
4	2	8.53%	2.44%	-3.15%	-8.33%		0.99%	
5	3	1.79%	4.46%	1.92%	19.18%		5.47%	
6	4	7.25%	17.90%	-6.53%	-7.41%		3.65%	
7	5	0.75%	-8.22%	-1.76%	-1.44%		-2.51%	
8	6	-1.57%	0.83%	12.88%	-5.92%		2.16%	
9	7	-2.10%	5.14%	13.41%	-0.46%		4.14%	
10								
11	Mean	1.15%	2.87%	1.79%	1.05%	<-- =AVERAGE(E3:E9)	1.73%	

We illustrate Propositions 3–5 in two steps:

- Step 1: We regress the returns of each asset on the returns of the efficient portfolio: For $i = 1, \ldots, 4$ we run the regression $r_{it} = \alpha_i + \beta_i r_{wt} + \varepsilon_{it}$. This regression is often called the *first pass regression*. The results are given below.

	A	B	C	D	E	F	G
29	**Implementing propositions 3-5—finding the SML**						
30	Step 1: Regress each asset's returns on those of the efficient portfolio w						
31		**Asset 1**	**Asset 2**	**Asset 3**	**Asset 4**		
32	Alpha	0.0024	-0.0047	-0.0002	0.0028	<-- =INTERCEPT(E3:E9,$G$3:$G$9)	
33	Beta	0.5284	1.9301	1.0490	0.4478	<-- =SLOPE(E3:E9,$G$3:$G$9)	
34	R-squared	0.0897	0.5241	0.1505	0.0167	<-- =RSQ(E3:E9,$G$3:$G$9)	

- Step 2: We now regress the betas of the assets on their mean returns. Running this regression, $\bar{r}_i = \gamma_0 + \gamma_1 \beta_i + \varepsilon_i$, gives:

	A	B	C	D	E
36	Step 2: Regress the asset mean returns on their betas				
37	Intercept	0.005	<-- =INTERCEPT(B11:E11,B33:E33)		
38	Slope	0.0123	<-- =SLOPE(B11:E11,B33:E33)		
39	R-squared	1.0000	<-- =RSQ(B11:E11,B33:E33)		

To check the results of Propositions 3–5, we run a test:

	A	B	C	D	E
41	**Check Propositions 3 & 4: Step 2 coefficients should be: Intercept = c, Slope = E(r_w) - c**				
42	Intercept = c ?	yes	<-- =IF(B36=B20,"yes","no")		
43	Slope = E(r_w) - c ?	yes	<-- =IF(B38=G11-B21,"yes","no")		

The “perfect” regression results (note the $R^2 = 1$ in cell B39) are the results promised us by Propositions 3–5:

- The second-pass regression intercept is equal to c and the slope is equal to $E(r_w) - c$.
- If there is a riskless asset with return $c = r_f$, then Proposition 5 promises that in the second-pass regression $\bar{r}_i = \gamma_0 + \gamma_1 \beta_i + \varepsilon_i$, $\gamma_0 = r_f$ and $\gamma_1 = E(r_w) - r_f$.
- If there is no riskless asset, then Proposition 3 states that in the second-pass regression $\gamma_0 = E(r_z)$ and $\gamma_1 = E(r_w) - E(r_z)$, where z is a portfolio whose covariance with w is zero.
- Finally, if we run a two-stage regression of the type described on *any portfolio w* and get a “perfect regression,” then Proposition 4 guarantees that w is in fact efficient.

To drive home the point that this technique always works, we show you all the calculations using a different value for c (cell B21, highlighted below). As proved in Propositions 3–5, the result is still a perfect regression of the means on the betas:

	A	B	C	D	E	F	G	H
1	**ILLUSTRATING PROPOSITIONS 3-5** **This time the constant is 2% (cell B21)**							
2	**Dates**	**Asset 1**	**Asset 2**	**Asset 3**	**Asset 4**		Efficient portfolio w	
3	1	-6.63%	-2.49%	-4.27%	11.72%		-2.95%	<-- {=MMULT(B3:E9,B23:B26)}
4	2	8.53%	2.44%	-3.15%	-8.33%		3.64%	
5	3	1.79%	4.46%	1.92%	19.18%		5.16%	
6	4	7.25%	17.90%	-6.53%	-7.41%		-2.40%	
7	5	0.75%	-8.22%	-1.76%	-1.44%		2.24%	
8	6	-1.57%	0.83%	12.88%	-5.92%		0.01%	
9	7	-2.10%	5.14%	13.41%	-0.46%		-0.26%	
10								
11	Mean	1.15%	2.87%	1.79%	1.05%	<-- =AVERAGE(E3:E9)	0.78%	
12								
13	**Variance-covariance matrix**							
14		Asset 1	Asset 2	Asset 3	Asset 4			
15	Asset 1	0.0024	0.0019	-0.0015	-0.0024	<-- {=MMULT(TRANSPOSE(B3:E9-B11:E11),B3:E9-B11:E11)/7}		
16	Asset 2	0.0019	0.0056	-0.0007	-0.0016			
17	Asset 3	-0.0015	-0.0007	0.0057	-0.0005			
18	Asset 4	-0.0024	-0.0016	-0.0005	0.0094			
19								
20	**Finding an efficient portfolio w**							
21	Constant	2.00%						
22								
23	Asset 1	0.8234	<-- {=MMULT(MINVERSE(B15:E18),TRANSPOSE(B11:E11)-B21)/SUM(MMULT(MINVERSE(B15:E18),TRANSPOSE(B11:E11)-B21))}					
24	Asset 2	-0.2869						
25	Asset 3	0.2278						
26	Asset 4	0.2357						
27								
28								
29	**Implementing propositions 3-5--finding the SML**							
30	Step 1: Regress each asset's returns on those of the efficient portfolio w							
31		**Asset 1**	**Asset 2**	**Asset 3**	**Asset 4**			
32	Alpha	0.0061	0.0342	0.0165	0.0044	<-- =INTERCEPT(E3:E9,$G$3:$G$9)		
33	Beta	0.6968	-0.7075	0.1752	0.7776	<-- =SLOPE(E3:E9,$G$3:$G$9)		
34	R-squared	0.1570	0.0709	0.0042	0.0506	<-- =RSQ(E3:E9,$G$3:$G$9)		
35								
36	Step 2: Regress the asset mean returns on their betas							
37	Intercept	0.02	<-- =INTERCEPT(B11:E11,B33:E33)					
38	Slope	-0.0122	<-- =SLOPE(B11:E11,B33:E33)					
39	R-squared	1.0000	<-- =RSQ(B11:E11,B33:E33)					
40								
41	**Check Propositions 3 & 4: Step 2 coefficients should be:** **Intercept = c, Slope = E(r_w) - c**							
42	Intercept = c ?	yes	<-- =IF(B36=B20,"yes","no")					
43	Slope = E(r_w) - c ?	yes	<-- =IF(B38=G11-B21,"yes","no")					

9.9 Summary

In this chapter we have presented theorems relating to efficient portfolios and then showed how to implement these theorems to find the efficient frontier. Two basic propositions allow us to derive portfolios on the envelope of the feasible set of portfolios and the envelope itself. Three further propositions relate the expected returns of any asset or portfolio to the expected returns on any efficient portfolio. Under certain circumstances, this allows us to derive the security market line (SML) and the capital market line (CML) of the classic capital asset pricing model (CAPM).

In subsequent chapters we discuss the implementation of the CAPM. We show how to compute the variance-covariance matrix (Chapter 10), how to test the SML (Chapter 11), how to optimize in the presence of short-sale constraints (Chapter 12), and how to derive useful portfolio optimization routines from our knowledge of efficient set mathematics (Chapter 13, which discusses the Black-Litterman model).

Exercises

1. Consider the data below for six furniture companies.

	A	B	C	D	E	F	G	H	I
2	**Variance-covariance matrix**	**La-Z-Boy**	**Kimball**	**Flexsteel**	**Leggett**	**Miller**	**Shaw**		**Means**
3	**La-Z-Boy**	0.1152	0.0398	0.1792	0.0492	0.0568	0.0989		29.24%
4	**Kimball**	0.0398	0.0649	0.0447	0.0062	0.0349	0.0269		20.68%
5	**Flexsteel**	0.1792	0.0447	0.3334	0.0775	0.0886	0.1487		25.02%
6	**Leggett**	0.0492	0.0062	0.0775	0.1033	0.0191	0.0597		31.64%
7	**Miller**	0.0568	0.0349	0.0886	0.0191	0.0594	0.0243		15.34%
8	**Shaw**	0.0989	0.0269	0.1487	0.0597	0.0243	0.1653		43.87%

a. Given this matrix, and assuming that the risk-free rate is 0%, calculate the efficient portfolio of these six firms.

b. Repeat, assuming that the risk-free rate is 10%.

c. Use these two portfolios to generate an efficient frontier for the six furniture companies. Plot this frontier.

d. Is there an efficient portfolio with only positive proportions of all the assets?

2. A sufficient condition to produce positively weighted efficient portfolios is that the variance-covariance matrix be diagonal: That is, that $\sigma_{ij} = 0$, for $i \neq j$. By continuity, positively weighted portfolios will result if the off-diagonal elements of the variance-covariance matrix are sufficiently small compared to the diagonal. Consider a transformation of the above matrix in which:

$$\sigma_{ij} = \begin{cases} \varepsilon\sigma_{ij}^{original} & \text{if } i \neq j \\ \sigma_{ii}^{original} & \end{cases}$$

When $\varepsilon = 1$, this transformation will give the original variance-covariance matrix and when $\varepsilon = 0$, the transformation will give a fully diagonal matrix.

For $r = 10\%$ find the maximum ε for which all portfolio weights are positive.

3. In the example below, use Excel to find an envelope portfolio whose β with respect to the efficient portfolio y is zero. *Hint*: Notice that because the covariance is linear, so is β: Suppose that $z = \lambda x + (1 - \lambda)y$ is a convex combination of x and y, and that we are trying to find the β_z. Then

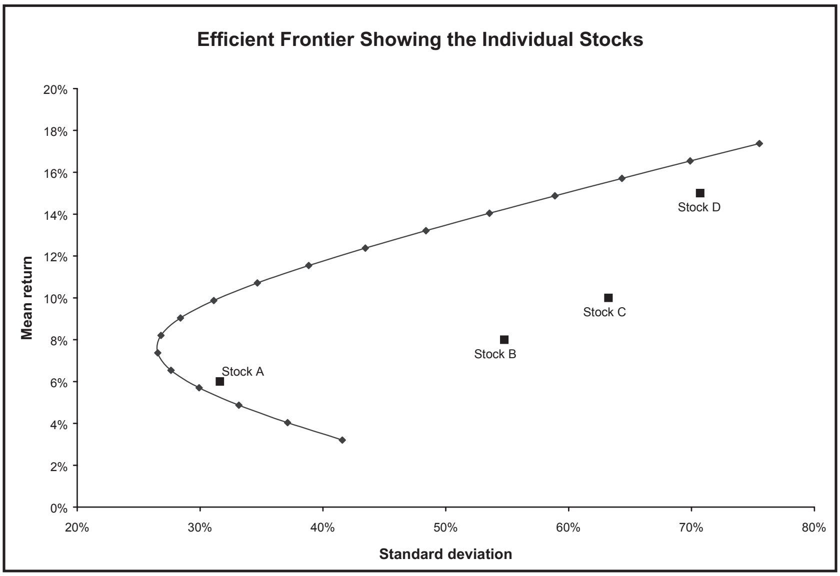
$$\beta_z = \frac{Cov(z, y)}{\sigma_y^2} = \frac{Cov(\lambda x + (1-\lambda)y, y)}{\sigma_y^2}$$
$$= \frac{\lambda Cov(x, y)}{\sigma_y^2} + \frac{(1-\lambda)Cov(y, y)}{\sigma_y^2} = \lambda\beta_x + (1-\lambda)$$

	A	B	C	D	E	F
1	**Variance-covariance matrix**					**Mean returns**
2	0.400	0.030	0.020	0.000		0.06
3	0.030	0.200	0.001	-0.060		0.05
4	0.020	0.001	0.300	0.030		0.07
5	0.000	-0.060	0.030	0.100		0.08

4. Calculate the envelope set for the four assets below and show that the individual assets all lie within this envelope set.

	A	B	C	D	E	F
1	**A FOUR-ASSET PORTFOLIO PROBLEM**					
2	Variance-covariance					Mean returns
3	0.10	0.01	0.03	0.05		6%
4	0.01	0.30	0.06	-0.04		8%
5	0.03	0.06	0.40	0.02		10%
6	0.05	-0.04	0.02	0.50		15%

You should get a graph which looks something like the following:



Mathematical Appendix

In this appendix we collect the various proofs of statements made in the chapter. As in the chapter, we assume that we are examining data for N risky assets. It is important to note that all the definitions of "feasibility" and "optimality" are made relative to this set. Thus the phrase "efficient" really means "efficient relative to the set of the N assets being examined."

PROPOSITION 0 The set of all feasible portfolios of risky assets is convex.

Proof A portfolio x is feasible if and only if the proportions of the portfolio add up to 1; i.e., $\sum_{i=1}^{N} x_i = 1$, where N is the number of risky assets. Suppose that x and y are feasible portfolios and suppose that λ is some number between 0 and 1. Then it is clear that $z = \lambda x + (1 - \lambda)y$ is also feasible.

PROPOSITION 1 Let c be a constant and denote by R the vector of mean returns. A portfolio x is on the envelope relative to the sample set of N assets if and only if it is the normalized solution of the system:

$$R - c = Sz$$

$$x_i = \frac{z_i}{\sum_h z_h}$$

Proof A portfolio x is on the envelope of the feasible set of portfolios if and only if it lies on the tangency of a line connecting some point c on the y-axis to the feasible set. Such a portfolio must either maximize or minimize the ratio $\frac{x(R-c)}{\sigma^2(x)}$, where $x(R - c)$ is the vector product which gives the portfolio's expected excess return over c, and $\sigma^2(x)$ is the portfolio's variance. Let this ratio's value, when maximized (or minimized), be λ. Then our portfolio must satisfy

$$\frac{x(R-c)}{\sigma^2(x)} = \lambda$$

$$\Rightarrow x(R-c) = \sigma^2(x)\lambda = xSx^T\lambda$$

Let h be a particular asset and differentiate this last expression with respect to x_h. This gives: $\bar{R}_h - c = Sx^T\lambda$. Writing $z_h = \lambda x_h$, we see that a portfolio is efficient if and only if it solves the system $R - c = Sz$. Normalizing z so that its coordinates add to 1 gives the desired result.

PROPOSITION 2 The convex combination of any two envelope portfolios is on the envelope of the feasible set.

Proof Let x and y be portfolios on the envelope. By the above theorem, it follows that there exist two vectors, z_x and z_y, and two constants c_x and c_y, such that:

- x is the normalized-to-unity vector of z_x; i.e., $x_i = \frac{z_{xi}}{\sum_h z_{xh}}$, and y is the normalized-to-unity vector of z_y.
- $R - c_x = Sz_x$ and $R - c_y = Sz_y$

Furthermore, since z maximizes the ratio $\frac{z(R-c)}{\sigma^2(z)}$, it follows that any normalization of z also maximizes this ratio. With no loss in generality, therefore, we can assume that z sums to 1.

It follows that for any real number a the portfolio $az_x + (1 - a)z_y$ solves the system $R - (ac_x + (1 - a)c_y) = Sz$. This proves our claim.

PROPOSITION 3 Let y be any envelope portfolio of the set of N assets. Then for any other portfolio x (including, possibly, a portfolio composed of a single asset) there exists a constant c such that the following relation holds between the expected return on x and the expected return on portfolio y:

$$E(r_x) = c + \beta_x[E(r_y) - c]$$

where

$$\beta_x = \frac{Cov(x, y)}{\sigma_y^2}$$

Furthermore, $c = E(r_z)$, where z is any portfolio for which $Cov(z,y) = 0$.

Proof Let y be a particular envelope portfolio and let x be any other portfolio. We assume that both portfolios x and y are column vectors. Note that

$$\beta_x = \frac{Cov(x, y)}{\sigma_y^2} = \frac{x^T Sy}{y^T Sy}$$

Now since y is on the envelope, we know that there exist a vector w and a constant c which solves the system $Sw = R - c$ and that $y = w \Big/ \sum_i w_i = w/a$. Substituting this in the expression for β_x, we get:

$$\beta_x = \frac{Cov(x,y)}{\sigma_y^2} = \frac{x^T Sy}{y^T Sy} = \frac{x^T (R-c)/a}{y^T (R-c)/a} = \frac{x^T (R-c)}{y^T (R-c)}$$

Next note that since $\sum_i x_i = 1$, it follows that $x^T I(R - c) = E(r_x) - c$ and that $y^T I(R - c) = E(r_y) - c$. This shows that

$$\beta_x = \frac{E(r_x) - c}{E(r_y) - c}$$

which can be rewritten as:

$$E(r_x) = c + \beta_x[E(r_y) - c]$$

To finish the proof, let z be a portfolio which has zero covariance with y. Then the above logic shows that $c = E(r_z)$. This proves the claim.

PROPOSITION 4 If in addition to the N risky assets, there exists a risk-free asset with return r_f, then the standard security market line holds:

$$E(r_x) = r_f + \beta_x[E(r_M) - r_f],$$

where

$$\beta_x = \frac{Cov(x, M)}{\sigma_M^2}$$

Proof If there exists a risk-free security, then the tangent line from this security to the efficient frontier dominates all other feasible portfolios. Call the point of tangency on the efficient frontier *M*; then the result follows.

Note: It is important to repeat again that the terminology "Market portfolio" refers in this case to the "Market portfolio relative to the sample set of *N* assets."

PROPOSITION 5 Suppose that there exists a portfolio *y* such that for any portfolio *x* the following relation holds:

$$E(r_x) = c + \beta_x [E(r_y) - c]$$

where

$$\beta_x = \frac{Cov(x, y)}{\sigma_y^2}$$

Then the portfolio *y* is on the envelope.

Proof Substituting in for the definition of β_x it follows that for any portfolio *x* the following relation holds:

$$\frac{x^T S y}{\sigma_y^2} = \frac{x^T R - c}{y^T R - c}$$

Let *x* be the vector composed solely of the first risky asset: $x = \{1, 0, \dots, 0\}$. Then the above equation becomes:

$$S_1 y \frac{y^T R - c}{\sigma_y^2} = E(r_1) - c$$

which we write:

$$S_1 a y = E(r_1) - c$$

where S_1 is the first row of the variance-covariance matrix *S*. Note that $a = \dfrac{y^T R - c}{\sigma_y^2}$ is a constant whose value is independent of the vector *x*. If we let *x* be a vector composed solely of the *i*th risky asset, we get:

$$S_i a y = E(r_i) - c$$

This proves that the vector $z = ay$ solves the system $Sz = R - c$; by Proposition 1 this means that the normalization of *z* is on the envelope. But this normalization is simply the vector *y*.

10 Calculating the Variance-Covariance Matrix

10.1 Overview

In order to calculate efficient portfolios, we must be able to compute the variance-covariance matrix from return data for stocks. In this chapter we discuss this computation, showing how to do the calculations in Excel. The most obvious calculation is the *sample variance-covariance matrix*: This is the matrix computed directly from the historic returns. We illustrate several methods for calculating the sample variance-covariance matrix, including a direct calculation in the spreadsheet using the excess return matrix and an implementation of this method with VBA.

While the sample variance-covariance matrix may appear to be an obvious choice, a large literature recognizes that it may not be the best estimate of variances and covariances. Disappointment with the sample variance-covariance matrix stems both from its often unrealistic parameters and from its inability to predict. These issues are discussed briefly in sections 10.5 and 10.6. As an alternative to the sample matrix, sections 10.7–10.10 discuss so-called “shrinkage” methods for improving the estimate of the variance-covariance matrix.[1]

Before starting this chapter, you may want to peruse Chapter 34 which discusses *array functions*. These are Excel functions whose arguments are vectors and matrices; their implementation is slightly different from standard Excel functions. This chapter makes heavy use of the array functions **Transpose( )** and **MMult( )** as well as some other “home-grown” array functions.

10.2 Computing the Sample Variance-Covariance Matrix

Suppose we have return data for N assets over M periods. Writing the return of asset i in period t as r_{it}, we write the *mean return* of asset i as:

$$\overline{r_i} = \frac{1}{M}\sum_{t=1}^{M} r_{it}, \; i = 1, \ldots, N$$

Then the covariance of the return of asset i and asset j is calculated as:

$$\sigma_{ij} = Cov(i, j) = \frac{1}{M-1}\sum_{t=1}^{M} (r_{it} - \overline{r_i}) \cdot (r_{jt} - \overline{r_j}), \; i, j = 1, \ldots, N$$

1. We return to the issue of prediction in Chapter 13, which discusses the Black-Litterman model of portfolio optimization.

The matrix of these covariances (which includes, of course, the variances when $i = j$) is the *sample variance-covariance matrix*. Our problem is to calculate these covariances efficiently. Define the *excess return matrix* to be:

$$A = matrix\ of\ excess\ returns = \begin{bmatrix} r_{11}-\overline{r}_1 & \cdots & r_{N1}-\overline{r}_N \\ r_{12}-\overline{r}_1 & \cdots & r_{N2}-\overline{r}_N \\ \vdots & & \vdots \\ r_{1M}-\overline{r}_1 & \cdots & r_{NM}-\overline{r}_N \end{bmatrix}$$

Columns of matrix A subtract the mean asset return from the individual asset returns. The transpose of this matrix is:

$$A^T = \begin{bmatrix} r_{11}-\overline{r}_1 & r_{12}-\overline{r}_1 & \cdots & \cdots & r_{1M}-\overline{r}_1 \\ \vdots & \vdots & & & \vdots \\ r_{N1}-\overline{r}_N & r_{N2}-r_N & \cdots & \cdots & r_{NM}-\overline{r}_N \end{bmatrix}$$

Multiplying A^T times A and dividing through by $M - 1$ gives the sample variance-covariance matrix:

$$S = [\sigma_{ij}] = \frac{A^T \cdot A}{M-1}$$

To consider the computational aspects, we use $M = 60$ months of return data for $N = 10$ stocks. The spreadsheet below shows the price data (adjusted for dividends) and the computed returns:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**FIVE YEARS OF PRICES FOR 10 STOCKS AND THE SP500**											
2		**McDonalds**	**US Steel**	**Arcelor-Mittal**	**Microsoft**	**Apple**	**Kellogg**	**General Electric**	**Bank of America**	**Pfizer**	**Exxon**	**S&P500**
3	**Date**	**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	**^GSPC**
4	1-Feb-07	37.57	84.74	44.84	25.53	84.61	43.45	28.97	44.68	19.57	64.20	1406.82
5	1-Mar-07	38.74	94.76	46.63	25.26	92.91	44.83	29.34	44.85	19.80	67.58	1420.86
6	2-Apr-07	41.52	97.03	47.10	27.14	99.80	46.12	30.59	44.74	20.74	71.10	1482.37
57	1-Jul-11	85.26	39.80	30.54	27.02	390.48	54.85	17.57	9.68	18.65	78.37	1292.28
58	1-Aug-11	89.74	30.01	21.74	26.40	384.83	53.84	16.00	8.15	18.60	73.18	1218.89
59	1-Sep-11	87.16	21.94	15.74	24.70	381.32	52.72	15.07	6.11	17.32	71.81	1131.42
60	3-Oct-11	92.16	25.27	20.51	26.43	404.78	53.73	16.55	6.82	18.87	77.20	1253.30
61	1-Nov-11	95.52	27.26	18.89	25.58	382.20	49.16	15.76	5.44	19.86	80.00	1246.96
62	1-Dec-11	100.33	26.42	18.19	25.96	405.00	50.57	17.91	5.56	21.42	84.30	1257.60
63	3-Jan-12	99.05	30.14	20.52	29.53	456.48	49.52	18.71	7.13	21.18	83.28	1312.41
64	1-Feb-12	99.99	31.01	23.30	30.77	493.17	50.21	19.13	8.18	21.14	84.88	1351.95

Using the Excel function **Ln(P_t/P_{t-1})**, we compute the monthly returns:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**FIVE YEARS OF MONTHLY RETURNS FOR 10 STOCKS AND THE SP500**											
2	**Date**	**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	**^GSPC**
3	1-Mar-07	3.07%	11.18%	3.91%	-1.06%	9.36%	3.13%	1.27%	0.38%	1.17%	5.13%	0.99%
4	2-Apr-07	6.93%	2.37%	1.00%	7.18%	7.15%	2.84%	4.17%	-0.25%	4.64%	5.08%	4.24%
5	1-May-07	4.59%	11.02%	12.16%	2.80%	19.42%	2.55%	1.91%	0.73%	4.89%	5.09%	3.20%
6	1-Jun-07	0.41%	-3.98%	3.93%	-4.06%	0.70%	-4.14%	2.60%	-3.66%	-7.23%	0.85%	-1.80%
7	2-Jul-07	-5.85%	-10.11%	-2.23%	-1.66%	7.66%	0.04%	1.24%	-3.06%	-8.39%	1.49%	-3.25%
8	1-Aug-07	2.83%	-3.72%	8.65%	-0.53%	4.97%	6.42%	0.28%	6.66%	6.70%	1.09%	1.28%
54	1-Jun-11	3.35%	-0.15%	3.86%	3.86%	-3.56%	-2.97%	-3.24%	-6.90%	-4.07%	-2.53%	-1.84%
55	1-Jul-11	2.53%	-14.09%	-10.97%	5.24%	15.12%	0.82%	-5.16%	-12.05%	-6.79%	-1.97%	-2.17%
56	1-Aug-11	5.12%	-28.23%	-33.99%	-2.32%	-1.46%	-1.86%	-9.36%	-17.20%	-0.27%	-6.85%	-5.85%
57	1-Sep-11	-2.92%	-31.32%	-32.29%	-6.66%	-0.92%	-2.10%	-5.99%	-28.81%	-7.13%	-1.89%	-7.45%
58	3-Oct-11	5.58%	14.13%	26.47%	6.77%	5.97%	1.90%	9.37%	10.99%	8.57%	7.24%	10.23%
59	1-Nov-11	3.58%	7.58%	-8.23%	-3.27%	-5.74%	-8.89%	-4.89%	-22.61%	5.11%	3.56%	-0.51%
60	1-Dec-11	4.91%	-3.13%	-3.78%	1.47%	5.79%	2.83%	12.79%	2.18%	7.56%	5.24%	0.85%
61	3-Jan-12	-1.28%	13.17%	12.05%	12.88%	11.97%	-2.10%	4.37%	24.87%	-1.13%	-1.22%	4.27%
62	1-Feb-12	0.94%	2.85%	12.71%	4.11%	7.73%	1.38%	2.22%	13.74%	-0.19%	1.90%	2.97%
63												
64	Mean	1.63%	-1.68%	-1.09%	0.31%	2.94%	0.24%	-0.69%	-2.83%	0.13%	0.47%	-0.07%

Below we compute the excess returns and the variance-covariance matrix:

	A	B	C	D	E	F	G	H	I	J	K	L
66	Variance-Covariance Matrix											
67		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	
68	**MCD**	0.0020	0.0037	0.0028	0.0015	0.0017	0.0007	0.0020	0.0031	0.0015	0.0011	
69	**X**	0.0037	0.0380	0.0284	0.0076	0.0111	0.0031	0.0127	0.0176	0.0043	0.0043	
70	**MT**	0.0028	0.0284	0.0267	0.0065	0.0097	0.0031	0.0102	0.0133	0.0038	0.0039	
71	**MSFT**	0.0015	0.0076	0.0065	0.0063	0.0049	0.0010	0.0046	0.0079	0.0018	0.0014	
72	**AAPL**	0.0017	0.0111	0.0097	0.0049	0.0126	0.0016	0.0049	0.0049	0.0007	0.0020	
73	**K**	0.0007	0.0031	0.0031	0.0010	0.0016	0.0026	0.0028	0.0046	0.0011	0.0003	
74	**GE**	0.0020	0.0127	0.0102	0.0046	0.0049	0.0028	0.0122	0.0163	0.0041	0.0022	
75	**BAC**	0.0031	0.0176	0.0133	0.0079	0.0049	0.0046	0.0163	0.0393	0.0080	0.0017	
76	**PFE**	0.0015	0.0043	0.0038	0.0018	0.0007	0.0011	0.0041	0.0080	0.0041	0.0011	
77	**XOM**	0.0011	0.0043	0.0039	0.0014	0.0020	0.0003	0.0022	0.0017	0.0011	0.0026	
78			<-- {=MMULT(TRANSPOSE(B83:K142),B83:K142)/59}									
79												
80												
81	Excess returns: r_{ij}-r_i											
82		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	
83	1-Mar-07	0.0144	0.1285	0.0501	-0.0137	0.0642	0.0289	0.0196	0.0321	0.0104	0.0467	<-- =K3-K$64
84	2-Apr-07	0.0530	0.0404	0.0209	0.0687	0.0422	0.0260	0.0486	0.0258	0.0451	0.0461	<-- =K4-K$64
85	1-May-07	0.0296	0.1269	0.1325	0.0249	0.1648	0.0231	0.0260	0.0356	0.0476	0.0462	<-- =K5-K$64
86	1-Jun-07	-0.0122	-0.0231	0.0502	-0.0437	-0.0224	-0.0438	0.0329	-0.0083	-0.0736	0.0039	<-- =K6-K$64
87	2-Jul-07	-0.0748	-0.0843	-0.0114	-0.0197	0.0473	-0.0020	0.0193	-0.0023	-0.0852	0.0102	
88	1-Aug-07	0.0119	-0.0205	0.0974	-0.0084	0.0204	0.0618	0.0097	0.0949	0.0657	0.0063	
89	4-Sep-07	0.0845	0.1313	0.1795	0.0217	0.0734	0.0168	0.0768	0.0327	-0.0180	0.0720	
90	1-Oct-07	0.0762	0.0351	0.0310	0.2197	0.1839	-0.0615	0.0009	-0.0121	0.0058	-0.0108	
91	1-Nov-07	-0.0120	-0.0806	-0.0644	-0.0911	-0.0709	0.0268	-0.0652	-0.0171	-0.0238	-0.0321	
92	3-Dec-07	-0.0088	0.2299	0.0576	0.0547	0.0541	-0.0327	-0.0170	-0.0690	-0.0458	0.0449	
93	2-Jan-08	-0.1111	-0.1537	-0.1419	-0.0913	-0.4101	-0.0948	-0.0403	0.0960	0.0260	-0.0937	
94	1-Feb-08	0.0003	0.0810	0.1464	-0.1801	-0.1088	0.0629	-0.0489	-0.0770	-0.0345	0.0148	
95	3-Mar-08	0.0140	0.1737	0.0892	0.0392	0.1085	0.0332	0.1172	-0.0022	-0.0637	-0.0331	
96	1-Apr-08	0.0497	0.2102	0.0963	0.0019	0.1631	-0.0292	-0.1169	0.0186	-0.0416	0.0910	
97	1-May-08	-0.0207	0.1332	0.1197	-0.0066	0.0523	0.0163	-0.0553	-0.0705	-0.0233	-0.0476	
98	2-Jun-08	-0.0635	0.0843	0.0120	-0.0319	-0.1492	-0.0786	-0.1228	-0.3064	-0.1040	-0.0118	

A VBA Function to Compute the Variance-Covariance Matrix

To automate this procedure, we write a VBA function that computes the variance-covariance matrix using the Excel function **Covariance.S**. When Excel functions with periods, such as **Covariance.S**, are used in VBA, the period becomes an underscore: **Covariance_S**:

```
'My thanks to Amir Kirsh
'Revised 2012 by Benjamin Czaczkes and _
Simon Benninga
Function VarCovar(rng As Range) As Variant
  Dim i As Integer
  Dim j As Integer
  Dim numcols As Integer
  numcols = rng.Columns.Count
  numrows = rng.Rows.Count
  Dim matrix() As Double
  ReDim matrix(numcols - 1, numcols - 1)
  For i = 1 To numcols
    For j = 1 To numcols
        matrix(i - 1, j - 1) = _
      Application.WorksheetFunction.
Covariance_S(rng.Columns(i), _
    rng.Columns(j))
    Next j
  Next i
  VarCovar = matrix
End Function
```

The VBA computes **Covariance_S** for every entry of the variance-covariance matrix.[2] Here's the result:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**PORTFOLIO ANALYSIS FOR MONTHLY DATA**											
2		Variance-Covariance Matrix										
3		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	
4	**MCD**	0.0020	0.0037	0.0028	0.0015	0.0017	0.0007	0.0020	0.0031	0.0015	0.0011	<-- {=varcovar('Page 253'!B3:K62)}
5	**X**	0.0037	0.0380	0.0284	0.0076	0.0111	0.0031	0.0127	0.0176	0.0043	0.0043	
6	**MT**	0.0028	0.0284	0.0267	0.0065	0.0097	0.0031	0.0102	0.0133	0.0038	0.0039	
7	**MSFT**	0.0015	0.0076	0.0065	0.0063	0.0049	0.0010	0.0046	0.0079	0.0018	0.0014	
8	**AAPL**	0.0017	0.0111	0.0097	0.0049	0.0126	0.0016	0.0049	0.0049	0.0007	0.0020	
9	**K**	0.0007	0.0031	0.0031	0.0010	0.0016	0.0026	0.0028	0.0046	0.0011	0.0003	
10	**GE**	0.0020	0.0127	0.0102	0.0046	0.0049	0.0028	0.0122	0.0163	0.0041	0.0022	
11	**BAC**	0.0031	0.0176	0.0133	0.0079	0.0049	0.0046	0.0163	0.0393	0.0080	0.0017	
12	**PFE**	0.0015	0.0043	0.0038	0.0018	0.0007	0.0011	0.0041	0.0080	0.0041	0.0011	
13	**XOM**	0.0011	0.0043	0.0039	0.0014	0.0020	0.0003	0.0022	0.0017	0.0011	0.0026	

Should We Divide by *M* – 1 or by *M*

In the above calculations, we use the sample covariance (in Excel **Covariance.S** and in VBA **Covariance_S**) to divide by $M - 1$ instead of M in order to get the unbiased estimate of the variances and covariances. We don't think this matters very much, but for reference to a higher authority, we suggest our discussion of M versus $M - 1$ in section 8.2.

Starting with Excel 2010, Microsoft has cleared up considerable confusion that once existed in Excel about whether to divide by M or $M - 1$. The new versions of Excel have standardized the nomenclature and computations for these functions:

Excel 2010 and later	Other (older) versions of this function (still work)	Comments	When used in VBA
Covariance.S		Sample covariance, divides by $M–1$	**Application. WorksheetFunction. Covariance_S**
Covariance.P	**Covar**	Population covariance, divides by M	**Application. WorksheetFunction. Covariance_P**

2. Since the covariance matrix is symmetric, we've actually done too many computations. But given the speed of our computers, who cares?

Excel 2010 and later	Other (older) versions of this function (still work)	Comments	When used in VBA
Var.S	**VarS**	Sample variance	**Application. WorksheetFunction.Var_S** **Application. WorksheetFunction.VarS**
Var.P	**VarP**	Population variance	**Application. WorksheetFunction. Var_P** **Application. WorksheetFunction.VarP**

Confused? Don't worry! As the discussion in Chapter 8 indicates, perhaps it doesn't matter very much.

10.3 The Correlation Matrix

Using the Excel function **Correl** we can compute the correlation matrix of the returns:

```
Function CorrMatrix(rng As Range) As Variant
  Dim i As Integer
  Dim j As Integer
  Dim numcols As Integer
  numcols = rng.Columns.Count
  numrows = rng.Rows.Count
  Dim matrix() As Double
  ReDim matrix(numcols - 1, numcols - 1)
```

```
  For i = 1 To numcols
   For j = 1 To numcols
   matrix(i - 1, j - 1) = _
   Application.WorksheetFunction.Correl(rng. _
   Columns(i), rng.Columns(j))
   Next j
  Next i
  CorrMatrix = matrix
End Function
```

	A	B	C	D	E	F	G	H	I	J	K
1	**CORRELATION MATRIX**										
2		**McDonalds**	**US Steel**	**Arcelor-Mittal**	**Microsoft**	**Apple**	**Kellogg**	**General Electric**	**Bank of America**	**Pfizer**	**Exxon**
3		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
4	**MCD**	1.0000	0.4199	0.3859	0.4238	0.3379	0.2920	0.4064	0.3506	0.5411	0.4741
5	**X**	0.4199	1.0000	0.8898	0.4898	0.5062	0.3078	0.5904	0.4556	0.3491	0.4361
6	**MT**	0.3859	0.8898	1.0000	0.5044	0.5277	0.3692	0.5659	0.4103	0.3602	0.4620
7	**MSFT**	0.4238	0.4898	0.5044	1.0000	0.5497	0.2416	0.5312	0.5050	0.3542	0.3581
8	**AAPL**	0.3379	0.5062	0.5277	0.5497	1.0000	0.2827	0.3964	0.2205	0.0945	0.3425
9	**K**	0.2920	0.3078	0.3692	0.2416	0.2827	1.0000	0.4846	0.4559	0.3487	0.1234
10	**GE**	0.4064	0.5904	0.5659	0.5312	0.3964	0.4846	1.0000	0.7461	0.5842	0.3926
11	**BAC**	0.3506	0.4556	0.4103	0.5050	0.2205	0.4559	0.7461	1.0000	0.6328	0.1723
12	**PFE**	0.5411	0.3491	0.3602	0.3542	0.0945	0.3487	0.5842	0.6328	1.0000	0.3435
13	**XOM**	0.4741	0.4361	0.4620	0.3581	0.3425	0.1234	0.3926	0.1723	0.3435	1.0000
14				<-- {=CorrMatrix('Page 253'!B3:K62)}							

Here's another version of the correlation matrix, this time only the upper half:

```
'Triangular correlation matrix
Function CorrMatrixTriangular(rng As Range) _
As Variant
  Dim i As Integer
  Dim j As Integer
  Dim numcols As Integer
  numcols = rng.Columns.Count
  numrows = rng.Rows.Count
  Dim matrix() As Variant
  ReDim matrix(numcols - 1, numcols - 1)
```

```
    For i = 1 To numcols
      For j = 1 To numcols
      If i <= j Then
        matrix(i - 1, j - 1) = _
      Application.WorksheetFunction.Correl(rng. _
      Columns(i), rng.Columns(j))
      Else
      matrix(i - 1, j - 1) = ""
      End If
      Next j
      Next i
    CorrMatrixTriangular = matrix
  End Function
```

	A	B	C	D	E	F	G	H	I	J	K
16		McDonalds	US Steel	Arcelor-Mittal	Microsoft	Apple	Kellogg	General Electric	Bank of America	Pfizer	Exxon
17		MCD	X	MT	MSFT	AAPL	K	GE	BAC	PFE	XOM
18	MCD	1.0000	0.4199	0.3859	0.4238	0.3379	0.2920	0.4064	0.3506	0.5411	0.4741
19	X		1.0000	0.8898	0.4898	0.5062	0.3078	0.5904	0.4556	0.3491	0.4361
20	MT			1.0000	0.5044	0.5277	0.3692	0.5659	0.4103	0.3602	0.4620
21	MSFT				1.0000	0.5497	0.2416	0.5312	0.5050	0.3542	0.3581
22	AAPL					1.0000	0.2827	0.3964	0.2205	0.0945	0.3425
23	K						1.0000	0.4846	0.4559	0.3487	0.1234
24	GE							1.0000	0.7461	0.5842	0.3926
25	BAC								1.0000	0.6328	0.1723
26	PFE									1.0000	0.3435
27	XOM										1.0000
28				<-- {=CorrMatrixTriangular('Page 253'!B3:K62)}							

Here are some statistics for the correlations. The average correlation for our sample (0.4226) is a bit high (usually a sample of stocks will give average correlation of 0.2–0.3). The largest correlations ($\rho_{\text{Arcelor,US Steel}} = 0.8898$, $\rho_{\text{GE,BankAmerica}} = 0.7461$) look quite high, though perhaps there are economic explanations.[3]

3. Arcelor and U.S. Steel are, of course, both steel companies. GE has one of the largest financing operations in the world; perhaps this explains the high correlation between the returns of Bank of America and GE? Or perhaps it's just a fluke of the data?

	A	B	C	D	E	F	G	H	I	J
30	Some correlation statistics									
31	Average	0.4226	<-- =AVERAGEIF(B18:K27,"<1")							
32	Largest	0.8898	<-- =LARGE(B18:K27,11)			Smallest	0.0945	<-- =SMALL($B$18:$K$27,1)		
33	Next largest	0.7461	<-- =LARGE(B18:K27,12)			Next small	0.1234	<-- =SMALL($B$18:$K$27,2)		
34	etc.	0.6328	<-- =LARGE(B18:K27,13)			etc.	0.1723	<-- =SMALL($B$18:$K$27,3)		
35	etc.	0.5904	<-- =LARGE(B18:K27,14)			etc.	0.2416	<-- =SMALL($B$18:$K$27,5)		
36	etc.	0.5842	<-- =LARGE(B18:K27,15)			etc.	0.2416	<-- =SMALL($B$18:$K$27,5)		

10.4 Computing the Global Minimum Variance Portfolio (GMVP)

The two most prominent uses of the variance-covariance matrix are to find the global minimum variance portfolio (GMVP) and to find efficient portfolios. Both uses illustrate the problematics of working with sample data and provide us with the introduction needed for sections 10.7–10.10, which discuss alternatives to the sample variance-covariance matrix. In this section we discuss the GMPV.

Suppose there are N assets having a variance-covariance matrix S. The GMVP is the portfolio $x = \{x_1, x_2, \ldots, x_N\}$ which has the lowest variance from among all feasible portfolios. The minimum variance portfolio is defined by

$$x_{GMVP} = \{x_{GMVP,1}, x_{GMVP,2}, \ldots, x_{GMVP,N}\} = \frac{1_{row} \cdot S^{-1}}{1_{row} \cdot S^{-1} \cdot 1_{row}^{T}},$$

$$\text{where } 1_{row} = \underbrace{\{1, 1, \ldots, 1\}}_{\substack{\uparrow \\ \text{N-dimensional} \\ \text{row vector of 1s}}} = \frac{1_{row} \cdot S^{-1}}{Sum(numerator)}$$

$$x_{GMVP} = \begin{Bmatrix} x_{GMVP,1} \\ x_{GMVP,2} \\ \vdots \\ x_{GMVP,N} \end{Bmatrix} = \frac{S^{-1} 1_{column}}{1_{column}^{T} \cdot S^{-1} \cdot 1_{column}}, \text{ where } 1_{column} = \underbrace{\begin{Bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{Bmatrix}}_{\substack{\uparrow \\ \text{N-dimensional} \\ \text{column vector of 1s}}}$$

$$= \frac{S^{-1} 1_{column}}{Sum(numerator)}$$

This formula is due to Merton.[4]

The particular fascination of the minimum variance portfolio is that it is the only portfolio on the efficient frontier whose computation does not require the asset expected returns. The mean μ_{GMVP} and the variance σ^2_{GMVP} of the minimum variance portfolio are given by:

$$\mu_{GMVP} = x_{GMVP} \cdot E(r), \sigma^2_{GMVP} = x_{GMVP} \cdot S \cdot x^T_{GMVP}$$

Here's an implementation of these formulas for our particular example. We use two VBA functions for the unit column and row vectors:

```
'I thank Priyush Singh and Ayal Itzkovitz
Function UnitrowVector(numcols As Integer) _
As Variant
  Dim i As Integer
  Dim vector() As Integer
  ReDim vector(0, numcols - 1)
  For i = 1 To numcols
    vector(0, i - 1) = 1
  Next i
  UnitrowVector = vector
End Function

Function UnitColVector(numrows As Integer) _
As Variant
  Dim i As Integer
  Dim vector() As Integer
  ReDim vector(numrows - 1, 0)
  For i = 1 To numrows
    vector(i - 1, 0) = 1
  Next i
  UnitColVector = vector
End Function
```

4. Robert C. Merton, "An Analytical Derivation of the Efficient Portfolio Frontier," *Journal of Financial and Quantitative Analysis* (1973).

Applying this to our 10-asset example:

	A	B	C	D	E	F	G	H	I	J	K
1		**COMPUTING THE GLOBAL MINIMUM VARIANCE PORTFOLIO**									
2		Variance-Covariance Matrix									
3		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
4	**MCD**	0.0020	0.0037	0.0028	0.0015	0.0017	0.0007	0.0020	0.0031	0.0015	0.0011
5	**X**	0.0037	0.0380	0.0284	0.0076	0.0111	0.0031	0.0127	0.0176	0.0043	0.0043
6	**MT**	0.0028	0.0284	0.0267	0.0065	0.0097	0.0031	0.0102	0.0133	0.0038	0.0039
7	**MSFT**	0.0015	0.0076	0.0065	0.0063	0.0049	0.0010	0.0046	0.0079	0.0018	0.0014
8	**AAPL**	0.0017	0.0111	0.0097	0.0049	0.0126	0.0016	0.0049	0.0049	0.0007	0.0020
9	**K**	0.0007	0.0031	0.0031	0.0010	0.0016	0.0026	0.0028	0.0046	0.0011	0.0003
10	**GE**	0.0020	0.0127	0.0102	0.0046	0.0049	0.0028	0.0122	0.0163	0.0041	0.0022
11	**BAC**	0.0031	0.0176	0.0133	0.0079	0.0049	0.0046	0.0163	0.0393	0.0080	0.0017
12	**PFE**	0.0015	0.0043	0.0038	0.0018	0.0007	0.0011	0.0041	0.0080	0.0041	0.0011
13	**XOM**	0.0011	0.0043	0.0039	0.0014	0.0020	0.0003	0.0022	0.0017	0.0011	0.0026
14											
15	GMVP as row	0.0326	0.2117	0.1754	0.0705	0.0873	0.0340	0.1166	0.1891	0.0493	0.0335
16			<-- {=MMULT(unitrowvector(10),B4:K13)/SUM(MMULT(unitrowvector(10),B4:K13))}								
17											
18	GMVP as column	0.0326									
19		0.2117	<-- {=MMULT(B4:K13,unitcolvector(10))/SUM(MMULT(B4:K13,unitcolvector(10)))}								
20		0.1754									
21		0.0705									
22		0.0873									
23		0.0340									
24		0.1166									
25		0.1891									
26		0.0493									
27		0.0335									
28											
29	**GMVP statistics**										
30	Mean	-0.80%	T(B15:K15,'Page 253'!B64:K64)								
31	Variance	0.0130	<-- {=MMULT(MMULT(B15:K15,B4:K13),B18:B27)}								
32	Sigma	11.40%	<-- =SQRT(B31)								

10.5 Four Alternatives to the Sample Variance-Covariance Matrix

In succeeding sections we illustrate four alternatives to the sample variance-covariance matrix:

- The **single-index model** assumes that the only sources of variance risk are the market variance and the betas of the assets.
- The **constant correlation model** assumes the correlation between all asset returns is constant, so that $\sigma_{ij} = \rho\ \sigma_i\ \sigma_j$.

- **Shrinkage methods** assume that the variance-covariance matrix is a convex combination of the sample variance-covariance and a matrix with variances on the diagonal and zeros elsewhere.
- **Option methods** use options to derive the standard deviations of returns for the assets. We combine this in section 10.9 with the constant correlation method to compute a variance-covariance matrix.

The first three models arise out of a distrust of return data for generating the covariances of the data in the future. The fourth method—using option data—goes further and assumes that even the sample variances are an inaccurate prediction of future variance.

10.6 Alternatives to the Sample Variance-Covariance: The Single-Index Model (SIM)

The single-index model (SIM) began as an attempt to simplify some of the computational complexities of calculating the variance-covariance matrix.[5] The basic assumption of the SIM is that the returns of each asset can be linearly regressed on a market index x:

$$\tilde{r}_i = \alpha_i + \beta_i \tilde{r}_x + \tilde{\varepsilon}_i$$

where the correlation between ε_i and ε_j is zero. Given this assumption, it is easy to establish the following two facts:

- $E(\tilde{r}_i) = \alpha_i + \beta_i E(\tilde{r}_x)$
- $\sigma_{ij} = \begin{cases} \beta_i \beta_j \sigma_x^2 & \text{when } i \neq j \\ \sigma_i^2 & \text{when } i = j \end{cases}$

5. W. M. Sharpe, "A Simplified Model for Portfolio Analysis," *Management Science* (1963).

Essentially the SIM involves changes in the estimates of the covariances, but not the sample variance. We can automate the procedure for computing the SIM by writing some VBA code:

```
Function sim(assetdata As Range, marketdata As Range) _
As Variant
    Dim i As Integer
    Dim j As Integer
    Dim numcols As Integer
    numcols = assetdata.Columns.Count
    Dim matrix() As Double
    ReDim matrix(numcols - 1, numcols - 1)

    For i = 1 To numcols
    For j = 1 To numcols
        If i = j Then
        matrix(i - 1, j - 1) = Application. _
        WorksheetFunction.Var_S(assetdata.Columns(i))
        Else
        matrix(i - 1, j - 1) = _
      Application.WorksheetFunction.Slope(assetdata. _
      Columns(i), marketdata) * _
      Application.WorksheetFunction.Slope(assetdata. _
      Columns(j), marketdata) * _
      Application.WorksheetFunction.Var_S(marketdata)
    End If
    Next j
    Next i
    sim = matrix
End Function
```

The two arguments of this function are the asset returns and the market returns. Applying this code in our example:

	A	B	C	D	E	F	G	H	I	J	K	L
1	**COMPUTING THE SINGLE-INDEX VARIANCE-COVARIANCE MATRIX**											
2		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	
3	**MCD**	0.0020	0.0036	0.0031	0.0013	0.0016	0.0006	0.0021	0.0032	0.0009	0.0006	
4	**X**	0.0036	0.0380	0.0198	0.0085	0.0105	0.0038	0.0137	0.0204	0.0059	0.0042	
5	**MT**	0.0031	0.0198	0.0267	0.0073	0.0090	0.0033	0.0117	0.0175	0.0051	0.0036	
6	**MSFT**	0.0013	0.0085	0.0073	0.0063	0.0038	0.0014	0.0050	0.0075	0.0022	0.0015	
7	**AAPL**	0.0016	0.0105	0.0090	0.0038	0.0126	0.0017	0.0062	0.0092	0.0027	0.0019	
8	**K**	0.0006	0.0038	0.0033	0.0014	0.0017	0.0026	0.0022	0.0034	0.0010	0.0007	
9	**GE**	0.0021	0.0137	0.0117	0.0050	0.0062	0.0022	0.0122	0.0121	0.0035	0.0025	
10	**BAC**	0.0032	0.0204	0.0175	0.0075	0.0092	0.0034	0.0121	0.0393	0.0052	0.0037	
11	**PFE**	0.0009	0.0059	0.0051	0.0022	0.0027	0.0010	0.0035	0.0052	0.0041	0.0011	
12	**XOM**	0.0006	0.0042	0.0036	0.0015	0.0019	0.0007	0.0025	0.0037	0.0011	0.0026	
13				<-- {=sim(B16:K75,L16:L75)}								
14												
15	**Date**	**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**	**^GSPC**
16	1-Mar-07	3.07%	11.18%	3.91%	-1.06%	9.36%	3.13%	1.27%	0.38%	1.17%	5.13%	0.99%
17	2-Apr-07	6.93%	2.37%	1.00%	7.18%	7.15%	2.84%	4.17%	-0.25%	4.64%	5.08%	4.24%
18	1-May-07	4.59%	11.02%	12.16%	2.80%	19.42%	2.55%	1.91%	0.73%	4.89%	5.09%	3.20%
19	1-Jun-07	0.41%	-3.98%	3.93%	-4.06%	0.70%	-4.14%	2.60%	-3.66%	-7.23%	0.85%	-1.80%
20	2-Jul-07	-5.85%	-10.11%	-2.23%	-1.66%	7.66%	0.04%	1.24%	-3.06%	-8.39%	1.49%	-3.25%
21	1-Aug-07	2.83%	-3.72%	8.65%	-0.53%	4.97%	6.42%	0.28%	6.66%	6.70%	1.09%	1.28%
67	1-Jun-11	3.35%	-0.15%	3.86%	3.86%	-3.56%	-2.97%	-3.24%	-6.90%	-4.07%	-2.53%	-1.84%
68	1-Jul-11	2.53%	-14.09%	-10.97%	5.24%	15.12%	0.82%	-5.16%	-12.05%	-6.79%	-1.97%	-2.17%
69	1-Aug-11	5.12%	-28.23%	-33.99%	-2.32%	-1.46%	-1.86%	-9.36%	-17.20%	-0.27%	-6.85%	-5.85%
70	1-Sep-11	-2.92%	-31.32%	-32.29%	-6.66%	-0.92%	-2.10%	-5.99%	-28.81%	-7.13%	-1.89%	-7.45%
71	3-Oct-11	5.58%	14.13%	26.47%	6.77%	5.97%	1.90%	9.37%	10.99%	8.57%	7.24%	10.23%
72	1-Nov-11	3.58%	7.58%	-8.23%	-3.27%	-5.74%	-8.89%	-4.89%	-22.61%	5.11%	3.56%	-0.51%
73	1-Dec-11	4.91%	-3.13%	-3.78%	1.47%	5.79%	2.83%	12.79%	2.18%	7.56%	5.24%	0.85%
74	3-Jan-12	-1.28%	13.17%	12.05%	12.88%	11.97%	-2.10%	4.37%	24.87%	-1.13%	-1.22%	4.27%
75	1-Feb-12	0.94%	2.85%	12.71%	4.11%	7.73%	1.38%	2.22%	13.74%	-0.19%	1.90%	2.97%

10.7 Alternatives to the Sample Variance-Covariance: Constant Correlation

The constant correlation model of Elton and Gruber (1973) computes the variance-covariance matrix by assuming that the variances of the asset returns are the sample returns, but that the covariances are all related by the same correlation coefficient, which is generally taken to be the average correlation coefficient of the assets in question. Since $Cov(r_i,r_j) = \sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$, this means that in the constant correlation model:

$$\sigma_{ij} = \begin{cases} \sigma_{ii} = \sigma_i^2 & \text{when } i = j \\ \sigma_{ij} = \rho\sigma_i\sigma_j & \text{when } i \neq j \end{cases}$$

Using our data for the 10 stocks, we can implement the constant correlation model. We first compute the correlations of all the stocks:

	A	B	C	D	E	F	G	H	I	J	K
1	ESTIMATING THE CONSTANT CORRELATION VARIANCE-COVARIANCE MATRIX										
2	Correlation	0.20									
3											
4		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
5	**MCD**	0.0020	0.0018	0.0015	0.0007	0.0010	0.0005	0.0010	0.0018	0.0006	0.0005
6	**X**	0.0018	0.0380	0.0064	0.0031	0.0044	0.0020	0.0043	0.0077	0.0025	0.0020
7	**MT**	0.0015	0.0064	0.0267	0.0026	0.0037	0.0017	0.0036	0.0065	0.0021	0.0017
8	**MSFT**	0.0007	0.0031	0.0026	0.0063	0.0018	0.0008	0.0017	0.0031	0.0010	0.0008
9	**AAPL**	0.0010	0.0044	0.0037	0.0018	0.0126	0.0012	0.0025	0.0044	0.0014	0.0011
10	**K**	0.0005	0.0020	0.0017	0.0008	0.0012	0.0026	0.0011	0.0020	0.0007	0.0005
11	**GE**	0.0010	0.0043	0.0036	0.0017	0.0025	0.0011	0.0122	0.0044	0.0014	0.0011
12	**BAC**	0.0018	0.0077	0.0065	0.0031	0.0044	0.0020	0.0044	0.0393	0.0025	0.0020
13	**PFE**	0.0006	0.0025	0.0021	0.0010	0.0014	0.0007	0.0014	0.0025	0.0041	0.0007
14	**XOM**	0.0005	0.0020	0.0017	0.0008	0.0011	0.0005	0.0011	0.0020	0.0007	0.0026
15		<-- {=constantcorr('Page 253'!B3:K62,'Page 265'!B2)}									

We've written a VBA function to compute this matrix from the return data:

```
Function constantcorr(data As Range, corr As Double) _
As Variant
    Dim i As Integer
    Dim j As Integer
    Dim numcols As Integer
    numcols = data.Columns.Count
    numrows = data.Rows.Count
    Dim matrix() As Double
    ReDim matrix(numcols - 1, numcols - 1)
    If Abs(corr) >= 1 Then GoTo Out
    For i = 1 To numcols
    For j = 1 To numcols
        If i = j Then
        matrix(i - 1, j - 1) = Application. _
        WorksheetFunction.Var_S(data.Columns(i))
        Else
```

```
        matrix(i - 1, j - 1) = corr * jjunk(data, i) * _
        jjunk(data, j)
      End If
    Next j
    Next i
Out:
    If Abs(corr) >= 1 Then constantcorr = VarCovar(data) _
    Else constantcorr = matrix
End Function
```

10.8 Alternatives to the Sample Variance-Covariance: Shrinkage Methods

A third class of methods of estimating the variance-covariance matrix has recently achieved popularity. So-called *shrinkage methods* assume that the variance-covariance matrix is a convex combination of the sample covariance matrix and some other matrix:

$$\textit{Shrinkage variance-covariance matrix} = \lambda * \textit{Sample var-cov} + (1-\lambda) * \textit{Other matrix}$$

In the example below, the "other" matrix is a diagonal matrix of only variances, with zeros elsewhere. The shrinkage estimator $\lambda = 0.3$ (cell B20).

	A	B	C	D	E	F	G	H
1	**ESTIMATING THE VARIANCE-COVARIANCE MATRIX USING THE SHRINKAGE APPROACH** **Gives weight 0.30 (the shrinkage factor) to sample var-cov and weight 0.70 to a diagonal matrix of only variances**							
2	**Return data**							
3	**Date**	**GE**	**MSFT**	**JNJ**	**K**	**BA**	**IBM**	
4	03-Jan-94	56.44%	-1.50%	6.01%	-9.79%	58.73%	21.51%	
5	03-Jan-95	18.23%	33.21%	41.56%	7.46%	-0.24%	6.04%	
6	02-Jan-96	56.93%	44.28%	57.71%	37.76%	65.55%	27.33%	
7	02-Jan-97	42.87%	79.12%	22.94%	-5.09%	54.34%	41.08%	
8	02-Jan-98	47.11%	38.04%	17.62%	32.04%	37.11%	2.63%	
9	04-Jan-99	34.55%	85.25%	26.62%	-10.74%	15.05%	-2.11%	
10	03-Jan-00	28.15%	11.20%	3.41%	-48.93%	43.53%	23.76%	
11	02-Jan-01	4.61%	-47.19%	10.69%	11.67%	28.29%	21.76%	
12	02-Jan-02	-19.74%	4.27%	23.11%	19.90%	-15.09%	4.55%	
13	02-Jan-03	-44.78%	-29.47%	-5.67%	10.88%	-23.23%	15.54%	
14	02-Jan-04	35.90%	18.01%	-1.27%	15.49%	39.82%	31.80%	
15								
16	**Average**	23.66%	21.38%	18.43%	5.51%	27.63%	17.63%	<-- =AVERAGE(G4:G14)
17	**Standard deviation**	32.17%	40.71%	18.97%	23.86%	29.93%	13.56%	<-- =STDEV(G4:G14)
18	**Variance**	0.1035	0.1657	0.0360	0.0570	0.0896	0.0184	<-- =VAR(G4:G14)
19								
20	**Shrinkage factor λ**	0.3	<-- This is the weight put on the sample var-cov					
21								
22	**Shrinkage matrix** **Uses the array formula {=B20*B34:G39+(1-B20)*B44:G49} to compute the shrinkage covariance matrix**							
23		**GE**	**MSFT**	**JNJ**	**K**	**BA**	**IBM**	
24	**GE**	0.1035	0.0228	0.0066	-0.0013	0.0257	0.0037	
25	**MSFT**	0.0228	0.1657	0.0124	-0.0016	0.0114	-0.0007	
26	**JNJ**	0.0066	0.0124	0.0360	0.0054	0.0030	-0.0012	
27	**K**	-0.0013	-0.0016	0.0054	0.0570	-0.0023	-0.0014	
28	**BA**	0.0257	0.0114	0.0030	-0.0023	0.0896	0.0074	
29	**IBM**	0.0037	-0.0007	-0.0012	-0.0014	0.0074	0.0184	
30								
31								
32	**Uses the array formula {=MMULT(TRANSPOSE(B4:G14-B16:G16),B4:G14-B16:G16)/10} to compute the constant sample covariance matrix. In the shrinkage var-cov, this matrix is given weight lambda.**							
33		**GE**	**MSFT**	**JNJ**	**K**	**BA**	**IBM**	
34	**GE**	0.1035	0.0758	0.0222	-0.0043	0.0857	0.0123	
35	**MSFT**	0.0758	0.1657	0.0412	-0.0052	0.0379	-0.0022	
36	**JNJ**	0.0222	0.0412	0.0360	0.0181	0.0101	-0.0039	
37	**K**	-0.0043	-0.0052	0.0181	0.0570	-0.0076	-0.0046	
38	**BA**	0.0857	0.0379	0.0101	-0.0076	0.0896	0.0248	
39	**IBM**	0.0123	-0.0022	-0.0039	-0.0046	0.0248	0.0184	
40								
41								
42	**Uses the array formula {=MMULT(TRANSPOSE(B4:G14-B16:G16),B4:G14-B16:G16)/10*IF(A44:A49=B43:G43,1,0)} to compute a matrix with only variances on diagonal and zeros elsewhere. In the shrinkage var-cov this matrix is given weight 1-lambda.**							
43		**GE**	**MSFT**	**JNJ**	**K**	**BA**	**IBM**	
44	**GE**	0.1035	0.0000	0.0000	0.0000	0.0000	0.0000	
45	**MSFT**	0.0000	0.1657	0.0000	0.0000	0.0000	0.0000	
46	**JNJ**	0.0000	0.0000	0.0360	0.0000	0.0000	0.0000	
47	**K**	0.0000	0.0000	0.0000	0.0570	0.0000	0.0000	
48	**BA**	0.0000	0.0000	0.0000	0.0000	0.0896	0.0000	
49	**IBM**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0184	

There is little theory about choosing the proper shrinkage estimator.[6] Our suggestion is to choose a shrinkage operator λ so that the GMVP is wholly positive (see next section for details).

10.9 Using Option Information to Compute the Variance Matrix[7]

Another way to compute the variance matrix is to use the information from the options market. We use the implied volatility for each of the stocks from their at-the-money call options and then compute the variance matrix using constant correlation:

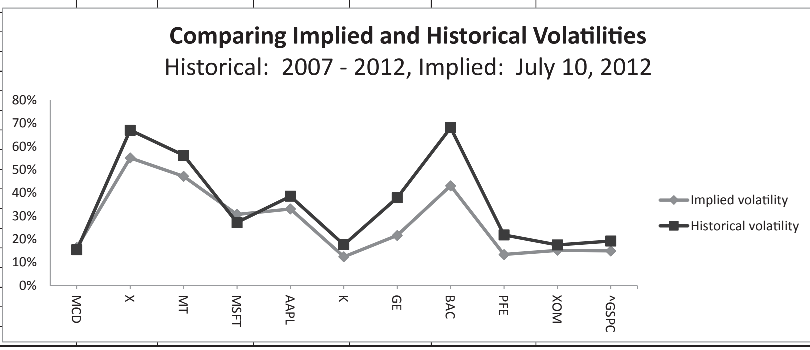
$$\sigma_{ij} = \begin{cases} \sigma^2_{i,implied} & \text{if } i = j \\ \rho\sigma_{i,implied}\sigma_{j,implied} & \text{if } i \neq j \end{cases}$$

Here's an example for our 10-stock case. We use data from the options markets and the function **CallVolatility** discussed in Chapter 17 to compute the implied volatility for each of the 10 stocks and the S&P 500. We use our data set of five years of returns to compute the historical volatility:

6. Three papers by Olivier Ledoit and Michael Wolf may offer some guidance: "Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection," *Journal of Empirical Finance*, 2003. "A Well-Conditioned Estimator for Large-Dimensional Covariance Matrices," *Journal of Multivariate Analysis*, 2004. "Honey, I Shrunk the Sample Covariance Matrix," *Journal of Portfolio Management*, 2004.

7. This section uses some information from the chapters on options.

	A	B	C	D	E	F	G
1	COMPARING IMPLIED AND HISTORICAL VOLATILITY						
2	Current date	10-Jul-12					
3	Expire date	17-Aug-12			=callvolatility(B7,C7,($B$3-$B$2)/365,$B$4,D7)		
4	Interest	1.00%					
5							
6		**Stock price**	**Strike**	**Price**	**Implied volatility**	**Historical volatility**	
7	**MCD**	90.25	90.00	2.09	16.50%	15.42%	<-- {=TRANSPOSE('Betas etc'!B11:L11)}
8	**X**	20.28	20.00	1.58	55.05%	67.00%	
9	**MT**	14.67	15.00	0.75	47.12%	56.18%	
10	**MSFT**	29.74	30.00	1.07	30.74%	27.21%	
11	**AAPL**	608.21	610.00	25.30	33.04%	38.53%	
12	**K**	48.98	50.00	0.40	12.44%	17.67%	
13	**GE**	19.62	20.00	0.39	21.63%	37.91%	
14	**BAC**	7.48	8.00	0.22	43.04%	68.10%	
15	**PFE**	22.44	23.00	0.18	13.38%	21.87%	
16	**XOM**	83.11	85.00	0.90	15.23%	17.55%	
17	**^GSPC**	1,341.47	1,350.00	22.50	14.96%	19.21%	



We can now use the implied volatilities as the basis for a constant correlation variance-covariance matrix:

	A	B	C	D	E	F	G	H	I	J	K
1	**CONSTANT CORRELATION MATRIX WITH IMPLIED VOLATILITIES**										
2	Correlation	0.20									
3											
4		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
5	**MCD**	0.0272	0.0182	0.0156	0.0101	0.0109	0.0041	0.0071	0.0142	0.0044	0.0050
6	**X**	0.0182	0.3031	0.0519	0.0338	0.0364	0.0137	0.0238	0.0474	0.0147	0.0168
7	**MT**	0.0156	0.0519	0.2221	0.0290	0.0311	0.0117	0.0204	0.0406	0.0126	0.0143
8	**MSFT**	0.0101	0.0338	0.0290	0.0945	0.0203	0.0076	0.0133	0.0265	0.0082	0.0094
9	**AAPL**	0.0109	0.0364	0.0311	0.0203	0.1091	0.0082	0.0143	0.0284	0.0088	0.0101
10	**K**	0.0041	0.0137	0.0117	0.0076	0.0082	0.0155	0.0054	0.0107	0.0033	0.0038
11	**GE**	0.0071	0.0238	0.0204	0.0133	0.0143	0.0054	0.0468	0.0186	0.0058	0.0066
12	**BAC**	0.0142	0.0474	0.0406	0.0265	0.0284	0.0107	0.0186	0.1852	0.0115	0.0131
13	**PFE**	0.0044	0.0147	0.0126	0.0082	0.0088	0.0033	0.0058	0.0115	0.0179	0.0041
14	**XOM**	0.0050	0.0168	0.0143	0.0094	0.0101	0.0038	0.0066	0.0131	0.0041	0.0232
15		<-- {=ImpliedVol(B5:K14,E22:E31,B2)}									
16											
17	Current date	10-Jul-12									
18	Expire date	17-Aug-12									
19	Interest	1.00%									
20											
21		**Stock price**	**Strike**	**Price**	**Implied volatility**						
22	**MCD**	90.25	90.00	2.09	16.50%	<-- =callvolatility(B22,C22,($B$18-$B$17)/365,$B$19,D22)					
23	**X**	20.28	20.00	1.58	55.05%						
24	**MT**	14.67	15.00	0.75	47.12%						
25	**MSFT**	29.74	30.00	1.07	30.74%						
26	**AAPL**	608.21	610.00	25.30	33.04%						
27	**K**	48.98	50.00	0.40	12.44%						
28	**GE**	19.62	20.00	0.39	21.63%						
29	**BAC**	7.48	8.00	0.22	43.04%						
30	**PFE**	22.44	23.00	0.18	13.38%						
31	**XOM**	83.11	85.00	0.90	15.23%						

The programming of the **ImpliedVolVarCov** is similar to previous VBAs in this chapter:

```
Function ImpliedVolVarCov(varcovarmatrix As _
  Range, volatilities As Range, corr As Double)
  As Variant
  Dim i As Integer
  Dim j As Integer
  Dim numcols As Integer
  numcols = varcovarmatrix.Columns.Count
  numrows = numcols
  Dim matrix() As Double
  ReDim matrix(numcols - 1, numcols - 1)
  If Abs(corr) >= 1 Then GoTo Out
```

```
  For i = 1 To numcols
  For j = 1 To numcols
    If i = j Then
    matrix(i - 1, j - 1) = volatilities(i) ^ 2
    Else
    matrix(i - 1, j - 1) = corr * _
      volatilities(i) * volatilities(j)
    End If
  Next j
  Next i
Out:
  If Abs(corr) >= 1 Then ImpliedVolVarCov = _
    "ERR" Else ImpliedVolVarCov = matrix
End Function
```

10.10 Which Method to Compute the Variance-Covariance Matrix?

This chapter gives five alternatives to computing the variance-covariance matrix:

- The sample variance-covariance
- The single-index model
- The constant correlation approach
- Shrinkage methods
- Implied volatility-based variance-covariance matrices

How do we compare these alternatives? Which one should we choose? We could compare the technical outcomes of using each of these methods—for example, show the alternative values of the GMVP using different methods—but this largely misses the point.

The choice of how to compute the variance-covariance matrix is largely a question of how you view capital markets. If you strongly believe that the past predicts the future, then perhaps your choice should be to use the sample varcov matrix. This author prefers to get away from history … our preference is to use an option-based volatility model with a changing correlation: In "normal" times we would use a "normal" correlation of between 0.2 and 0.3; in times of crisis, we would use a much higher correlation, say, $\rho = 0.5 - 0.6$.

10.11 Summary

In this chapter we considered how to compute the variance-covariance matrix which is central to all portfolio optimization. Starting with the standard sample variance-covariance matrix, we also showed how to compute several alternatives that have appeared in the literature as perhaps improving portfolio computations.

Exercises

1. Below you will find annual return data for six furniture companies for the years 1982–1992. Use these data to calculate the variance-covariance matrix of the returns.

	A	B	C	D	E	F	G	H
1	**DATA FOR 6 FURNITURE COMPANIES**							
2		**La-Z-Boy**	**Kimball**	**Flexsteel**	**Leggett & Platt**	**Herman Miller**	**Shaw Industries**	
3	**1982**	36.67%	0.20%	41.54%	21.92%	26.13%	22.50%	
4	**1983**	122.82%	61.43%	195.09%	62.27%	73.38%	117.89%	
5	**1984**	14.44%	63.51%	-38.38%	-1.27%	45.15%	7.80%	
6	**1985**	21.39%	28.42%	1.30%	81.17%	24.27%	38.14%	
7	**1986**	45.36%	-7.44%	21.89%	19.83%	10.73%	54.48%	
8	**1987**	20.19%	48.27%	9.11%	-10.21%	-11.92%	26.82%	
9	**1988**	-8.94%	-11.28%	12.65%	13.77%	7.06%	-6.24%	
10	**1989**	27.02%	12.85%	12.08%	32.55%	-7.55%	123.03%	
11	**1990**	-11.64%	2.42%	-17.13%	-6.48%	1.31%	15.48%	
12	**1991**	20.29%	6.90%	3.62%	50.12%	-5.54%	19.92%	
13	**1992**	34.08%	22.21%	33.46%	84.40%	5.71%	62.76%	
14								
15	**Beta**	0.80	0.95	0.65	0.85	0.85	1.40	
16	**Mean returns**	29.24%	20.68%	25.02%	31.64%	15.34%	43.87%	<-- =AVERAGE(G3:G13)

The remaining exercises refer to the data in the tab **Price data** on the exercise spreadsheet that accompanies this book. This tab gives three years of price data for six stocks and the S&P 500 as a surrogate for the market.

2. Compute the returns of the data and the statistics for each of the assets (mean return, variance and standard deviation of return, beta).

3. Compute the sample variance-covariance matrix and the correlation matrix for the six stocks.

4. Use the function **SIM** defined in the chapter to compute the single-index variance-covariance matrix.

5. Compute the global minimum variance portfolio (GMVP) using the sample variance-covariance matrix.

6. Compute the GMVP using the constant correlation covariance matrix.

11 Estimating Betas and the Security Market Line

11.1 Overview

The capital asset pricing model (CAPM) is one of the two most influential innovations in financial theory in the latter half of the twentieth century.[1] By integrating the portfolio decision with utility theory and the statistical behavior of asset prices, the formulators of the CAPM defined the paradigm which is now generally used for the analysis of stock prices.

What does the CAPM actually say? What are its empirical implications? Roughly speaking, we can differentiate between two kinds of implications of the CAPM. First, the capital market line (CML) defines the *individual optimal portfolios* for an investor interested in the mean and variance of her optimal portfolio. Second, given agreement between investors on the statistical properties of asset returns and on the importance of mean-variance optimization, the security market line (SML) defines the *risk-return* relation for *each individual asset*.

It is useful to differentiate between the case wherein a risk-free asset exists and the case wherein there is no risk-free asset.[2]

Case 1: A Risk-Free Asset Exists

Suppose a risk-free asset exists and has return r_f. We can differentiate between the individual optimization of investors and the general equilibrium implications of the CAPM:

• *Individual optimization:* Assuming that investors optimize based on the expected return and standard deviation of their portfolio returns (in the jargon of finance—they have "mean-variance" preferences), the CAPM states that each individual investor's optimal portfolio falls on the line $E(r_p) = r_f + \sigma_p\left[E(r_x) - r_f\right]$, where portfolio x is a portfolio which maximizes $\frac{E(r_y) - r_f}{\sigma_y}$ for all feasible portfolios y. Proposition 1 of Chapter 9 shows that x can be computed by $x = \{x_1, x_2, \ldots, x_N\} = \frac{S^{-1}\left[E(r) - r_f\right]}{\sum S^{-1}\left[E(r) - r_f\right]}$, where

1. The other remarkable innovation is option pricing theory, which is discussed in Chapters 15–19. These two innovations together have accounted for a number of Nobel Prizes in Economics: Harry Markowitz (1990), William Sharpe (1990), Myron Scholes (1997), and Robert Merton (1997). But for their untimely demise, others associated with these theories—Jan Mossin (1936–1987), and Fischer Black (1938–1995)—would doubtless also have received the Nobel.

2. The existence (or non-existence) of a risk-free asset is closely related to the investment horizon. Assets which are risk-free over a short term may not be riskless over a longer term.

S is the variance-covariance matrix of risky asset returns and $E(r) = \{E(r_1), E(r_2), \ldots, E(r_N)\}$ is the vector of expected asset returns.

• *General equilibrium*: If all investors agree about the statistical assumptions of the model—the variance-covariance matrix S and the vector of expected asset returns $E(r)$—and if a risk-free asset exists, then individual asset returns are defined by the security market line (SML):

$$E(r_i) = r_f + \frac{Cov(r_i, r_M)}{\sigma_M^2}\left[E(r_M) - r_f\right]$$

where M denotes the market portfolio—the value-weighted portfolio of all risky assets. The expression $\frac{Cov(r_i, r_M)}{\sigma_M^2}$ is generally termed the asset's *beta*:

$$\beta_i = \frac{Cov(r_i, r_M)}{\sigma_M^2}.$$

Case 2: No Risk-Free Asset Exists

If there is no risk-free asset, then the implications of the CAPM both for individual optimization and for general equilibrium are defined by Black's (1972) zero-beta model (Proposition 3 of Chapter 9):

• *Individual optimization*: In the absence of a risk-free asset, individual optimal portfolios will fall along the *efficient frontier*. As shown in Proposition 2 of Chapter 9, this frontier is the upward-sloping portion of the mean-sigma combinations created by the convex combination of any two optimizing portfolios $x = \frac{S^{-1}[E(r) - c_1]}{\sum S^{-1}[E(r) - c_1]}$ and $y = \frac{S^{-1}[E(r) - c_2]}{\sum S^{-1}[E(r) - c_2]}$, where c_1 and c_2 are two arbitrary constants.

• *General equilibrium*: In the absence of a risk-free asset, if all investors agree about the statistical assumptions of the model—the variance-covariance matrix S and the vector of expected asset returns $E(r)$—then individual asset returns are defined by the security market line (SML):

$$E(r_i) = E(r_z) + \frac{Cov(r_i, r_y)}{\sigma_y^2}\left[E(r_y) - E(r_z)\right]$$

where y is any efficient portfolio and z is a portfolio which has zero covariance with y (the so-called "zero beta portfolio").

The case of no risk-free asset is obviously weaker than the case of a risk-free asset. If there is a risk-free asset, the general equilibrium version of the CAPM says that all portfolios are situated on a single, agreed-upon line. If there is no risk-free asset, then all optimal portfolios are on the same frontier; but in this case asset betas can differ, since there are many portfolios *y* which fulfill the equation $E(r_i) = E(r_z) + \frac{Cov(r_i, r_y)}{\sigma_y^2}[E(r_y) - E(r_z)]$.

The CAPM as a Prescriptive and a Descriptive Tool

As you can see from the discussion above, the CAPM is both *prescriptive* and *descriptive*.

As a prescriptive tool, the CAPM tells a mean-variance investor how to choose his optimal portfolio. By finding a portfolio of the form $\frac{S^{-1}[E(r) - c_1]}{\sum S^{-1}[E(r) - c_1]}$, the investor can identify an optimal portfolio from the data set.

As a descriptive tool, the CAPM gives conditions under which we can generalize about the structure of expected returns in the market. Whether or not a risk-free asset exists, these conditions assume that investors agree on the statistical structure of asset returns—the variance-covariance matrix and the expected returns. In this case all the returns are expected to lie on a security market line (SML) of the form $E(r_i) = r_f + \frac{Cov(r_i, r_M)}{\sigma_M^2}[E(r_M) - r_f]$ (if there is a risk-free asset) or of the form $E(r_i) = E(r_z) + \frac{Cov(r_i, r_y)}{\sigma_y^2}[E(r_y) - E(r_z)]$ if there is no risk-free asset.

This Chapter

In this chapter we look at some typical capital market data and replicate a simple test of the descriptive part of the CAPM. This means that we have to calculate the betas for a set of assets, and we then have to determine the equation of the security market line (SML). The test in this chapter is the simplest possible test of the CAPM. There is an enormous literature in which the possible statistical and methodological pitfalls of CAPM tests are discussed. Good places to begin are textbooks by Elton, Gruber, Brown, and Goetzmann (2009), and Bodie, Kane, and Marcus (2010).[3]

11.2 Testing the SML

Typical tests of the security market line (SML) start with return data on a set of risky assets. The steps in the test are as follows:

• Determine a candidate for the market portfolio *M*. In our example we will use the Standard & Poor's 500 Index (S&P 500) as a candidate for *M*. This is a critical step: In principle, the "true" market portfolio should—as pointed out in Chapter 9—contain all the market's risky assets in proportion to their value. It is clearly impossible to calculate this theoretical market portfolio, and we must therefore make do with a surrogate. As you will see in the next two sections, the propositions of Chapter 9 can shed much light on how the choice of the market surrogate affects the *r*-squared of our regression test of the CAPM.

• For each of the assets in question, determine the asset beta, $\beta_i = \dfrac{Cov(r_i, r_M)}{\sigma_M^2}$.

This is often called the *first-pass regression*.

• Regress the mean returns of the assets on their respective betas (the *second-pass regression*):

$$\bar{r}_i = \gamma_0 + \gamma_1 \beta_i$$

If the CAPM in its descriptive format holds, then the second-pass regression should be the security market line.[4]

We illustrate the tests of the CAPM with a simple numerical example that uses data for the 30 stocks in the Dow-Jones Industrials. We start with the prices of the S&P 500 (symbol ^GSPC) and the stocks in the DJ30 (some of the rows and columns are not shown):

3. For further references, see the Selected References at the end of this book. Our personal expositional favorite is a paper by Roll, "Ambiguity When Performance Is Measured by the Securities Market Line," *Journal of Finance* (1978).

4. This is a direct consequence of Propositions 3 and 4 of Chapter 9.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**PRICE DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500** July 2001 - July 2006													
2	**Date**	**S&P500 Index ^GSPC**	**Alcoa AA**	**American International Group AIG**	**American Express AXP**	**Boeing BA**	**Citigroup C**	**Caterpillar CAT**	**DuPont DD**	**Disney DIS**	**General Electric GE**	**General Motors GM**	**Home Depot HD**	**Honeywell HON**
3	03-Jul-01	1211.23	35.37	81.32	33.73	53.64	40.97	24.73	36.29	25.02	38.33	49.72	48.26	32.75
4	01-Aug-01	1133.58	34.50	76.43	30.46	47.06	37.49	22.45	35.01	24.14	36.04	43.14	44.06	33.27
5	04-Sep-01	1040.94	28.06	76.24	24.30	30.79	33.15	20.11	32.07	17.68	32.93	33.81	36.79	23.57
6	01-Oct-01	1059.78	29.34	76.82	24.68	29.97	37.26	20.23	34.18	17.65	32.23	32.56	36.66	26.38
7	01-Nov-01	1139.45	35.09	80.54	27.60	32.43	39.34	21.45	38.21	19.43	34.08	39.63	44.78	29.77
8	03-Dec-01	1148.08	32.32	77.64	29.93	35.83	41.46	23.63	36.63	19.88	35.63	38.75	48.97	30.38
9	02-Jan-02	1130.20	32.59	72.51	30.13	37.83	39.06	22.91	38.06	20.21	33.03	40.77	48.09	30.19
10	01-Feb-02	1106.73	34.31	72.38	30.64	42.64	37.29	25.29	40.68	22.07	34.39	42.67	48.00	34.43
51	01-Jul-05	1234.18	27.48	59.76	47.68	65.02	42.62	53.28	41.22	25.37	33.53	34.89	42.97	38.43
52	01-Aug-05	1220.33	26.38	58.92	47.89	66.26	42.89	54.85	38.55	24.93	32.66	32.86	39.92	37.66
53	01-Sep-05	1228.81	24.05	61.67	49.79	67.18	44.60	58.07	38.16	23.88	32.94	29.42	37.76	36.89
54	03-Oct-05	1207.01	23.92	64.49	49.41	63.91	44.86	52.22	40.62	24.11	33.17	26.33	40.63	33.65
55	01-Nov-05	1249.48	27.16	66.82	51.04	67.67	48.04	57.37	42.02	24.67	34.94	21.44	41.46	36.15
56	01-Dec-05	1248.29	29.30	67.91	51.08	69.71	48.02	57.36	41.77	23.97	34.53	19.01	40.17	36.85
57	03-Jan-06	1280.08	31.21	65.15	52.19	67.79	46.09	67.69	38.48	25.31	32.27	23.56	40.24	38.01

We first transform these price data to returns:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**RETURN DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500** July 2001 - July 2006													
2	**Date**	**S&P 500 Index ^GSPC**	**Alcoa AA**	**American International Group AIG**	**American Express AXP**	**Boeing BA**	**Citigroup C**	**Caterpillar CAT**	**DuPont DD**	**Disney DIS**	**General Electric GE**	**General Motors GM**	**Home Depot HD**	**Honeywell HON**
3														
4	Average return	0.07%	-0.09%	[illegible]	[illegible]	0.67%	0.30%	1.79%	0.18%	0.29%	-0.23%	-0.87%	-0.52%	0.29%
5	Beta	1.00	1.90	[illegible]	[illegible]	[illegible]	1.30	1.39	1.00	1.28	0.84	1.41	1.55	1.66
6	Alpha	0	-0.23%	[illegible]	[illegible]	[illegible]	[illegible]	1.69%	0.11%	0.20%	-0.30%	-0.97%	-0.63%	0.17%
7	R-squared	1	0.6085	[illegible]	[illegible]	[illegible]	[illegible]	0.5158	0.4362	0.3845	0.3221	0.2607	0.5288	0.5473
8														
9	01-Aug-01	-6.63%	-2.49%	-6.20%	-10.20%	-13.09%	-8.88%	-9.67%	-3.59%	-3.58%	-6.16%	-14.20%	-9.11%	1.58%
10	04-Sep-01	-8.53%	-20.66%	[illegible]	[illegible]	[illegible]	-12.30%	-11.01%	-8.77%	-31.14%	-9.02%	-24.37%	-18.03%	-34.47%
11	01-Oct-01	1.79%	4.46%	[illegible]	[illegible]	[illegible]	11.69%	0.59%	6.37%	-0.17%	-2.15%	-3.77%	-0.35%	11.26%
12	01-Nov-01	7.25%	17.90%	4.73%	11.18%	7.89%	5.43%	5.86%	11.15%	9.61%	5.58%	19.65%	20.01%	12.09%
13	03-Dec-01	0.75%	-8.22%	-3.67%	8.10%	9.97%	5.25%	9.68%	-4.22%	2.29%	4.45%	-2.25%	8.94%	2.03%
14	02-Jan-02	-1.57%	0.83%	-6.84%	0.67%	5.43%	-5.96%	-3.09%	3.83%	1.65%	-7.58%	5.08%	-1.81%	-0.63%
15	01-Feb-02	-2.10%	5.14%	-0.18%	1.68%	11.97%	-4.64%	9.88%	6.66%	8.80%	4.03%	4.55%	-0.19%	13.14%
16	01-Mar-02	3.61%	0.47%	-2.52%	11.66%	4.85%	9.02%	2.38%	0.66%	0.36%	-2.89%	13.20%	-2.72%	0.41%

=AVERAGE(C9:C68)

=SLOPE(C9:C68,$B$9:$B$68)

=INTERCEPT(C9:C68,$B$9:$B$68)

=RSQ(C9:C68,$B$9:$B$68)

The First-Pass Regression

Row 4 gives each asset's average monthly return over the 60-month period (to annualize these returns, we would multiply by 12). Rows 5–7 report the results of the *first-pass regression*. For each asset i we report the regression $r_{it} = \alpha_i + \beta_i r_{SP,t}$. We use the Excel function **Slope** to compute the β of each asset, and the functions **Intercept** and **Rsq** to compute the α and R^2 for each regression.

As a check, we also compute the α, β, and R^2 for the S&P 500 index (column B). Not surprisingly, $\alpha_{SP} = 0$, $\beta_{SP} = 1$, $R^2 = 1$.

The Second-Pass Regression

The SML postulates that the mean return of each security should be linearly related to its beta. Assuming that the historic data provide an accurate description of the distribution of future returns, we postulate that $E(R_i) = \alpha + \beta_i \Pi + \varepsilon_i$, where the definitions of α and Π depend on whether we are in Case 1 or Case 2 of section 11.1:

$$\alpha = \begin{cases} r_f & \text{Case 1: there exists a risk-free asset} \\ E(r_z) & \text{Case 2: no risk-free asset. } z \text{ has zero correlation with efficient portfolio } y \end{cases}$$

$$\Pi = \begin{cases} E(r_M) - r_f & \text{Case 1} \\ E(r_y) - E(r_z) & \text{Case 2} \end{cases}$$

In the second step of our test of the CAPM, we examine this hypothesis by regressing the mean returns on the β's.

	A	B	C	D	E	F	G	H
1	**THE SECOND-PASS REGRESSION**							
2	**Stock**	**Average monthly return**	**Beta**	**Alpha**				
3	Alcoa AA	-0.09%	1.9028	-0.0023		**Second-pass regression, regressing monthly returns on Beta**		
4	American International Group AIG	-0.54%	0.9936	-0.0061		Intercept	0.0036	<-- =INTERCEPT(B3:B32,C3:C32)
5	American Express AXP	0.72%	1.3784	0.0062		Slope	-0.0020	<-- =SLOPE(B3:B32,C3:C32)
6	Boeing BA	0.67%	1.1515	0.0058		R-squared	0.0238	<-- =RSQ(B3:B32,C3:C32)
7	Citigroup C	0.30%	1.2952	0.0021				
8	Caterpillar CAT	1.79%	1.3903	0.0169		t-statistic, intercept	1.2381	<-- =tintercept(B3:B32,C3:C32)
9	DuPont DD	0.18%	1.0009	0.0011		t-statistic, slope	-0.8254	<-- =tslope(B3:B32,C3:C32)
10	Disney DIS	0.29%	1.2805	0.0020				
11	General Electric GE	-0.23%	0.8420	-0.0030				
12	General Motors GM	-0.87%	1.4060	-0.0097				
13	Home Depot HD	-0.52%	1.5528	-0.0063				
14	Honeywell HON	0.29%	1.6640	0.0017				
15	Hewlett Packard HPQ	0.61%	1.9594	0.0046				
16	IBM	-0.47%	1.5764	-0.0058				
17	Intel INTC	-0.73%	2.2648	-0.0089				
18	Johnson & Johnson JNJ	0.34%	0.2471	0.0032				
19	JP Morgan JPM	0.18%	1.7917	0.0005				
20	Coca Cola KO	0.12%	0.3590	0.0009				
21	McDonalds MCD	0.35%	1.2646	0.0025				
22	3M MMM	0.64%	0.6504	0.0059				
23	AltriaMO	1.30%	0.6633	0.0125				
24	Merck MRK	-0.63%	0.6099	-0.0068				
25	Microsoft MSFT	-0.35%	1.1219	-0.0043				
26	Pfizer PFE	-0.74%	0.5572	-0.0078				
27	Proctor Gamble PG	0.94%	0.1687	0.0093				
28	AT&T T	-0.41%	1.1275	-0.0050				
29	United Technologies UTX	1.03%	1.0659	0.0095				
30	Verizon VZ	-0.49%	1.0231	-0.0057				
31	Walmart WMT	-0.25%	0.6000	-0.0030				
32	Exxon Mobil XOM	0.88%	0.6455	0.0083				

The results (cells F4:G6) are very disappointing. Our test yields the following SML:

$$E(r_i) = \underbrace{0.0036}_{\substack{\uparrow \\ \gamma_0}} - \underbrace{0.0020}_{\substack{\uparrow \\ \gamma_1}} \beta_i, \ R^2 = 0.0238$$

There is nothing about these numbers which inspires confidence:

- γ_0 should correspond to the risk-free rate over the period. In section 11.9 we discuss this rate, which changed wildly over the 60 months surveyed. At this point it is enough to point out that the average monthly risk-free interest rate was 0.18% (or 0.0018, exactly half of γ_0).
- γ_1 should correspond to $E(r_M) - r_f$. The average monthly return of the S&P 500 over the period was –0.10% and the average monthly risk-free interest rate was 0.18%, so that γ_1 should be approximated by –0.28% (or 0.0028).
- Both the *t*-statistics for the *i* (cell G8) and the slope (cell G9) indicate that they are not statistically different from zero.[5]

Our test of the SML has failed. The CAPM may have prescriptive validity, but it does not describe our data.

Why Are the Results So Bad?

The experiment we did—checking the CAPM by plotting the security market line—does not appear to have worked out very well. There does not appear to be much evidence in favor of the SML: Neither the R^2 of the regression nor the *t*-statistics give much evidence that there is a relation between expected return and portfolio β.

There are a number of reasons why these disappointing results may hold:

- One reason is that perhaps the CAPM itself does not hold. This could be true for a variety of reasons:
  - ◊ Perhaps in the market short sales of assets are restricted. Our derivation of the CAPM (see Chapter 9 on efficient portfolios) assumes that there are no short-sale restrictions. Clearly this is an unrealistic assumption. The computation of efficient portfolios when short sales are restricted is

5. The functions **TIntercept** and **TSlope** were created by the author. They are attached to the spreadsheet for this chapter and are discussed in Chapter 3.

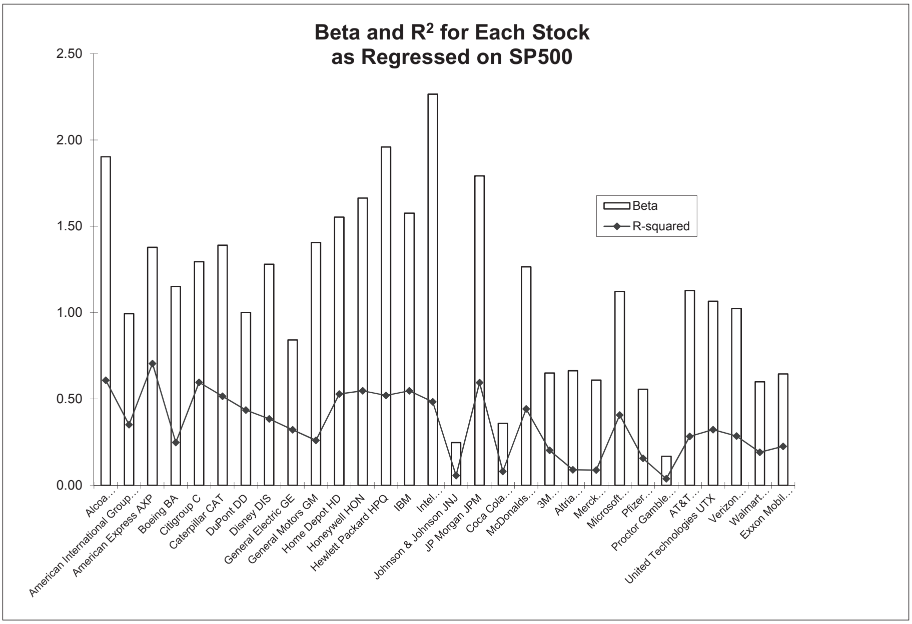
considered in Chapter 12. In this case, however, there are no simple relations (such as those proved in Chapter 9) between the returns of assets and their betas. In particular, if short sales are restricted, there is no reason to expect the SML to hold.

◊ Perhaps individuals do not have homogeneous probability assessments, or perhaps they do not have the same expectations of asset returns, variances, and covariances.

- Perhaps the CAPM holds only for portfolios and not for single assets.
- Perhaps our set of assets isn't large enough: After all, the CAPM talks about *all risky assets*, whereas we have chosen—for illustrative purposes—to do our test on a very small subset of these assets. The literature on CAPM testing records tests in which the set of risky assets has been expanded to include bonds, real estate, and even non-diversifiable assets such as human capital.
- Perhaps the "market portfolio" isn't efficient. This possibility is suggested by the mathematics of Chapter 9 on efficient portfolios, and it is this suggestion which we further explore in the next section.
- Perhaps the CAPM holds only if the market returns are positive (in the period surveyed they were, on average, negative).

11.3 Did We Learn Something?

The results of our exercise in section 11.1 are quite disappointing. Did we learn anything positive from this exercise? Absolutely. For example, the regression model does a pretty good job of describing individual asset returns in relation to the S&P 500:



On average the S&P 500 describes about 35% of the variability of the DJ30 stocks, which have an average beta of 1.12. If we exclude the seven stocks with the lowest R^2, the S&P describes almost 43% of the variation in the stocks' returns:

	A	B	C	D	E	F	G	H	I
1	**OUR SML EXERCISE: WHAT DID WE LEARN?**								
2	Average alpha	0.06%	<-- =AVERAGE('Page 277 bottom'!C6:AF6)						
3	Average beta	1.12	<-- =AVERAGE('Page 277 bottom'!C5:AF5)						
4	Average r-squared	0.3510	<-- =AVERAGE('Page 277 bottom'!C7:AF7)						
5									
6	**Average R^2 for best regressions**								
7	Cutoff for R^2	0.2	<-- Below we count all R^2 which are greater than this number						
8		9.8258	<-- =SUMIF('Page 277 bottom'!C7:AF7,">"&TEXT(B7,"0.00"))						
9		23	<-- =COUNTIF('Page 277 bottom'!C7:AF7,">"&TEXT(B7,"0.00"))						
10	Average R^2	0.4272	<-- =B8/B9						
11									
12	**T-statistics for intercept and slope**								
13		Alcoa AA	American International Group AIG	American Express AXP	Boeing BA	Citigroup C	Caterpillar CAT	DuPont DD	Disney DIS
14	t-stat for intercept	0.3144	0.6324	-1.0525	-0.1584	-0.2013	-1.6120	-0.0192	-0.0371
15	t-stat for slope	9.4942	5.6112	11.7783	4.3815	9.2729	7.8607	6.6993	6.0199
16									
17	Average absolute t-stat for intercept	0.3998	<-- {=AVERAGE(ABS(B14:AE14))}						
18	Average t-stat for slope	5.7866	<-- =AVERAGE(B15:AE15)						

Cell B10 above computes the average R^2 for those regressions which had an $R^2 > 0.2$. This is 23 of the Dow-Jones 30. So—on average the first-pass regressions are very significant. The average R^2 of 35% that we got for our first-pass regressions of the basic SML is actually a respectable number in finance. Students—influenced by over-enthusiastic statistics instructors and an overly linear view of the world—often feel that the R^2 of any convincing regression should be at least 90%. Finance does not appear to be a highly linear profession: A good rule of thumb is that any financial regression that gives an R^2 greater than 80% is possibly misspecified and misleading.[6]

Another way to look at the significance of our results is to compute the *t*-statistics for the intercept and slope of the first-pass regressions (rows 14–15 above). While the intercepts are not significantly different from zero (since their *t*-statistic is less than 2), the slopes are very significant.

An Excel Note: Computing the Absolute Value of an Array of Numbers

In the computations above we use a neat Excel trick related to array functions (see Chapter 34). By using **Abs** as an array function (that is, by entering the function using [Ctrl]+[Shift]+[Enter]), we can compute the average of the absolute values of a vector of numbers. A simple example is shown below:

	A	B	C	D	E	F
1	**USING ABS FUNCTION IN ARRAY** **The Excel "Abs" function computes the absolute value** **If we use it as an array function, it can be applied to a range of numbers**					
2						
3	Numbers	1	-2	-3	-6	8
4						
5	Average number	-0.4000	<-- =AVERAGE(B3:F3)			
6	Average absolute number	4.0000	<-- {=AVERAGE(ABS(B3:F3))}			
7	The above, but not as array function	1.0000	<-- =AVERAGE(ABS(B3:F3))			

Notice cell B7: Using the same function as a regular function does not produce the correct answer.

6. An exception to this useful rule relates to diversified portfolios—here the R^2 increases dramatically.

11.4 The Non-Efficiency of the "Market Portfolio"

When we calculated the SML in section 11.1, we regressed the mean return of each asset on the returns of the market portfolio. The propositions of Chapter 9 on efficient portfolios suggest that our failure to find adequate results may stem from the fact that the S&P 500 portfolio is not efficient relative to the set of the six assets which we have chosen. Proposition 3 of Chapter 9 states that if we had chosen to regress our asset returns on a portfolio that is efficient with respect to the asset set itself, we would get an *r*-squared of 100%. Proposition 4 of Chapter 9 shows that if we get an *r*-squared of 100% then the portfolio on which we regress the asset returns is necessarily efficient with respect to the set of assets. In this section we give a numerical illustration of these propositions.

In the spreadsheet below we create a "mysterious portfolio" in column B. This portfolio (its construction is described in the next subsection) is efficient with respect to the Dow-Jones 30. As you can see in cells A10:B12, when we perform the second-pass regression—regressing the individual average returns of the assets on their betas computed with respect to the mysterious portfolio—the results are perfect. The resulting regression has an intercept of 0.0030 and a slope of 0.0425. Most important—it has an R^2 of 100%.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**RETURN DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500** **July 2001 - July 2006**													
2	**Date**	**Mysterious portfolio**	**Alcoa AA**	**American International Group AIG**	**American Express AXP**	**Boeing BA**	**Citigroup C**	**Caterpillar CAT**	**DuPont DD**	**Disney DIS**	**General Electric GE**	**General Motors GM**	**Home Depot HD**	**Honeywell HON**
3														
4	Average return	4.55%	-0.09%	-0.54%	0.72%	0.67%	0.30%	1.79%	0.18%	0.29%	-0.23%	-0.87%	-0.52%	0.29%
5	Beta		-0.09	-0.20	0.10	0.09	0.00	0.35	-0.03	0.00	-0.13	-0.28	-0.19	0.00
6	Alpha		0.33%	0.36%	0.27%	0.27%	0.30%	0.19%	0.31%	0.30%	0.34%	0.38%	0.36%	0.30%
7	R-squared		0.0025	0.0242	0.0064	0.0024	0.0000	0.0579	0.0006	0.0000	0.0126	0.0176	0.0143	0.0000
8														
9	**SML--regressing the average returns on the betas**													
10	Intercept	0.0030	<-- =INTERCEPT(C4:AF4,C5:AF5)											
11	Slope	0.0425	<-- =SLOPE(C4:AF4,C5:AF5)											
12	R-squared	1.0000	<-- =RSQ(C4:AF4,C5:AF5)											
13														
14														
15	01-Aug-01	-1.01%	-2.49%	-6.20%	-10.20%	-13.09%	-8.88%	-9.67%	-3.59%	-3.58%	-6.16%	-14.20%	-9.11%	1.58%
16	04-Sep-01	0.40%	-20.66%	-0.25%	-22.59%	-42.42%	-12.30%	-11.01%	-8.77%	-31.14%	-9.02%	-24.37%	-18.03%	-34.47%
17	01-Oct-01	4.71%	4.46%	0.76%	1.55%	-2.70%	11.69%	0.59%	6.37%	-0.17%	-2.15%	-3.77%	-0.35%	11.26%
18	01-Nov-01	-1.33%	17.90%	4.73%	11.18%	7.89%	5.43%	5.86%	11.15%	9.61%	5.58%	19.65%	20.01%	12.09%
19	03-Dec-01	8.11%	-8.22%	-3.67%	8.10%	9.97%	5.25%	9.68%	-4.22%	2.29%	4.45%	-2.25%	8.94%	2.03%

The Mysterious Portfolio Is Efficient

The propositions of Chapter 9 leave us with only one conclusion: The "mysterious portfolio" must be efficient with respect to the DJ30. And so it is. In the spreadsheet below we show the construction of this efficient portfolio, which follows the propositions of Chapter 9.

- We first construct the variance-covariance matrix S using the function **Varcovar** defined in Chapter 10.
- We then compute the efficient portfolio by solving $\dfrac{S^{-1}[E(r)-c]}{\sum S^{-1}[E(r)-c]}$. In the spreadsheet below we use $c = 0.0030$, which then turns out to be the intercept of the second-pass regression.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**RETURN DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500** **July 2001 - July 2006**													
2	**Date**	**SP 500 Index ^SPX**	**Alcoa AA**	**American International Group AIG**	**American Express AXP**	**Boeing BA**	**Citigroup C**	**Caterpillar CAT**	**DuPont DD**	**Disney DIS**	**General Electric GE**	**General Motors GM**	**Home Depot HD**	**Honeywell HON**
3														
4	Average return	0.07%	-0.09%	[illegible]	[illegible]72%	0.67%	0.30%	1.79%	0.18%	0.29%	-0.23%	-0.87%	-0.52%	0.29%
5	Beta	1.00	1.90	[illegible]	[illegible]	[illegible]	1.30	1.39	1.00	1.28	0.84	1.41	1.55	1.66
6	Alpha	0	-0.23%	-0.61%	0.62%	0.58%	0.21%	1.69%	0.11%	0.20%	-0.30%	-0.97%	-0.63%	0.17%
7	R-squared	1	0.6085	[illegible]	[illegible]	[illegible]	0.5972	0.5158	0.4362	0.3845	0.3221	0.2607	0.5288	0.5473
8														
9	01-Aug-01	-6.63%	-2.49%	-6.20%	-10.20%	-13.09%	-8.88%	-9.67%	-3.59%	-3.58%	-6.16%	-14.20%	-9.11%	1.58%
10	04-Sep-01	-8.53%	-20.66%	[illegible]	[illegible]	[illegible]42%	-12.30%	-11.01%	-8.77%	-31.14%	-9.02%	-24.37%	-18.03%	-34.47%
11	01-Oct-01	1.79%	4.46%	[illegible]	[illegible]	[illegible]70%	11.69%	0.59%	6.37%	-0.17%	-2.15%	-3.77%	-0.35%	11.26%
12	01-Nov-01	7.25%	17.90%	4.73%	11.18%	7.89%	5.43%	5.86%	11.15%	9.61%	5.58%	19.65%	20.01%	12.09%
13	03-Dec-01	0.75%	-8.22%	-3.67%	8.10%	9.97%	5.25%	9.68%	-4.22%	2.29%	4.45%	-2.25%	8.94%	2.03%
14	02-Jan-02	-1.57%	0.83%	-6.84%	0.67%	5.43%	-5.96%	-3.09%	3.83%	1.65%	-7.58%	5.08%	-1.81%	-0.63%
15	01-Feb-02	-2.10%	5.14%	-0.18%	1.68%	11.97%	-4.64%	9.88%	6.66%	8.80%	4.03%	4.55%	-0.19%	13.14%
67	01-Jun-06	0.01%	2.00%	-2.92%	-2.13%	-1.62%	-2.15%	2.08%	-2.21%	-1.65%	-3.14%	10.09%	-5.91%	-2.16%
68	03-Jul-06	-0.37%	3.61%	-0.14%	-2.08%	-2.37%	1.71%	-2.58%	-2.58%	-0.57%	1.03%	-1.05%	-1.18%	-3.38%
69														
70														
71	**The cells below compute the variance-covariance matrix for the DJ30 by using the formula {=varcovar(C9:AF68)}**													
72		AA	AIG	AXP	BA	C	CAT	DD	DIS	GE	GM	HD	HON	HPQ
73	AA	0.0093	0.0032	0.0038	0.0045	0.0033	0.0046	0.0040	0.0042	0.0024	0.0046	0.0046	0.0061	0.0060
74	AIG	0.0032	0.0044	0.0020	0.0010	0.0023	0.0022	0.0022	0.0015	0.0015	0.0019	0.0021	0.0027	0.0024
75	AXP	0.0038	0.0020	0.0042	0.0030	0.0031	0.0029	0.0020	0.0037	0.0019	0.0033	0.0033	0.0042	0.0050
76	BA	0.0045	0.0010	0.0030	0.0083	0.0014	0.0029	0.0021	0.0037	0.0020	0.0043	0.0026	0.0053	0.0042
99	UTX	0.0038	0.0012	0.0031	0.0050	0.0022	0.0026	0.0021	0.0038	0.0018	0.0030	0.0023	0.0049	0.0048
100	VZ	0.0028	0.0018	0.0019	0.0007	0.0023	0.0022	0.0018	0.0014	0.0013	0.0006	0.0016	0.0024	0.0030
101	WMT	0.0018	0.0014	0.0011	0.0003	0.0015	0.0011	0.0014	0.0009	0.0009	0.0012	0.0024	0.0008	0.0008
102	XOM	0.0024	0.0010	0.0010	0.0017	0.0009	0.0021	0.0012	0.0004	0.0007	0.0020	0.0007	0.0020	0.0012
103														
104	**Finding an efficient portfolio**													
105	Constant	0.30%												
106														
107	AA	5.45%	<-- {=MMULT(MINVERSE(varcov),TRANSPOSE(C4:AF4)-B105)/SUM(MMULT(MINVERSE(varcov),TRANSPOSE(C4:AF4)-B105))}											
108	AIG	-11.77%												
109	AXP	-5.78%												
110	BA	-13.94%												
111	C	-36.60%												
112	CAT	76.26%												
113	DD	-22.55%												
114	DIS	-17.05%												
115	GE	-8.80%												
116	GM	-37.73%												
117	HD	-37.21%												
118	HON	-17.40%												
119	HPQ	39.79%												
120	IBM	-26.38%												
121	INTC	-18.62%												
122	JNJ	65.08%												
123	JPM	53.61%												
124	KO	-12.95%												
125	MCD	-12.18%												
126	MMM	-2.07%												
127	MP	42.15%												
128	MRK	8.29%												
129	MSFT	3.61%												
130	PFE	-61.19%												
131	PG	54.67%												
132	T	-8.38%												
133	UTX	44.05%												
134	VZ	-36.55%												
135	WMT	64.78%												
136	XOM	29.40%												
137	Sum	100.00%												

=AVERAGE(C9:C68)

=SLOPE(C9:C68,$B$9:$B$68)

=INTERCEPT(C9:C68,$B$9:$B$68)

=RSQ(C9:C68,$B$9:$B$68)

The "mysterious portfolio" is not unique. Following we show results using another constant *c,* which gives another version of the SML.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	RETURN DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500 July 2001 - July 2006													
2	Date	Mysterious portfolio	Alcoa AA	American International Group AIG	American Express AXP	Boeing BA	Citigroup C	Caterpillar CAT	DuPont DD	Disney DIS	General Electric GE	General Motors GM	Home Depot HD	Honeywell HON
3														
4	Average return	8.88%	-0.09%	-0.54%	0.72%	0.67%	0.30%	1.79%	0.18%	0.29%	-0.23%	-0.87%	-0.52%	0.29%
5	Beta		-0.07	-0.12	0.03	0.02	-0.02	0.15	-0.04	-0.02	-0.09	-0.16	-0.12	-0.03
6	Alpha		0.54%	0.56%	0.49%	0.49%	0.51%	0.42%	0.52%	0.51%	0.54%	0.58%	0.56%	0.51%
7	R-squared		0.0061	0.0399	0.0019	0.0005	0.0015	0.0467	0.0045	0.0010	0.0256	0.0259	0.0237	0.0009
8														
9	**SML--regressing the average returns on the betas**													
10	Intercept	0.0050	<-- =INTERCEPT(C4:AF4,C5:AF5)											
11	Slope	0.0838	<-- =SLOPE(C4:AF4,C5:AF5)											
12	R-squared	1.0000	<-- =RSQ(C4:AF4,C5:AF5)											

Note also that even though the R^2 of the second pass regression is 100% (since the "mysterious portfolio" is efficient), the R^2's of the individual first-pass regressions are far from notable.

11.5 So What's the Real Market Portfolio? How Can We Test the CAPM?

A little reflection will reveal that although the "mysterious" portfolio of the previous section may be efficient with respect to the 30 stocks of the Dow-Jones, it could not be the *true market portfolio*, even if the DJ30 stocks represented the whole universe of risky securities. This is because many of the stocks appear in the "mysterious portfolio" with negative weights. Surely a minimal characteristic of the market portfolio must be that all shares appear in it with *positive proportions*.

Roll (1977, 1978) suggests that the only test of the CAPM is to answer the question: *Is the true market portfolio mean-variance efficient?* If the answer to this question is "yes," then it follows from Proposition 3 of Chapter 9 that a linear relation holds between the mean of each portfolio and its β. In our example, we can shed some light on this question by building a table of the asset proportions of portfolios on the efficient frontier.

In the table below we give some evidence that all efficient portfolios for the DJ30 contain significant short positions. Using the wonders of Excel's **Data Table**, we compute the largest short and long positions for a series of efficient portfolios, each defined by its own constant *c*. All of these portfolios contain large short positions (and, as you can see, also large long positions):

	A	B	C	D	E	F	G	H	I
105	**An efficient portfolio**			**Data table: computing the largest short and long position for a given constant c**					
106	Constant	0.30%			**Largest short**	**Largest long**			
107				**Constant c**			<-- Data table hidden: =B141		
108	AA	5.5%		0.00%	-32.64%	52.33%			
109	AIG	-11.8%		0.05%	-35.51%	53.58%			
110	AXP	-5.8%		0.10%	-38.87%	55.05%			
111	BA	-13.9%		0.15%	-42.86%	56.79%			
112	C	-36.6%		0.20%	-47.69%	59.70%			
113	CAT	76.3%		0.25%	-53.65%	67.01%			
114	DD	-22.6%		0.30%	-61.19%	76.26%			
115	DIS	-17.0%		0.35%	-71.01%	88.32%			
116	GE	-8.8%		0.40%	-84.36%	104.71%			
117	GM	-37.7%		0.45%	-103.56%	128.28%			
118	HD	-37.2%		0.50%	-133.51%	165.05%			
119	HON	-17.4%		0.55%	-186.77%	230.42%			
120	HPQ	39.8%		0.60%	-307.86%	379.08%			
121	IBM	-26.4%		0.65%	-853.66%	1049.09%			
122	INTC	-18.6%		0.70%	-1398.93%	1140.50%			
123	JNJ	65.1%		0.75%	-422.59%	345.18%			
124	JPM	53.6%		0.80%	-249.90%	204.50%			
125	KO	-13.0%							
126	MCD	-12.2%							
127	MMM	-2.1%							
128	MP	42.1%							
129	MRK	8.3%							
130	MSFT	3.6%							
131	PFE	-61.2%							
132	PG	54.7%							
133	T	-8.4%							
134	UTX	44.1%							
135	VZ	-36.6%							
136	WMT	64.8%							
137	XOM	29.4%							
138	Sum	100.0%							
139									
140	Largest short	-61.2%	<-- =MIN(B108:B137)						
141	Largest long	76.3%	<-- =MAX(B108:B137)						

Our depressing conclusion: If the data for the DJ30 and the S&P 500 are representative, the CAPM as a descriptive theory of capital markets appears not to work.[7]

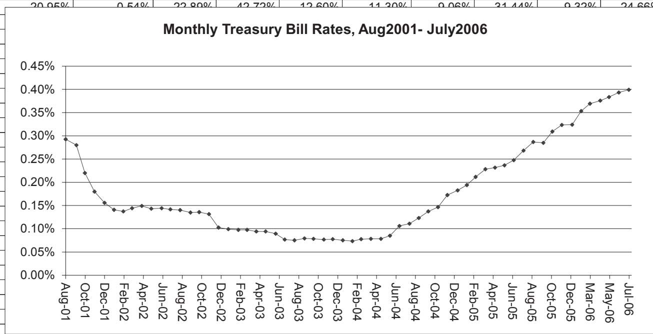
11.6 Using Excess Returns

Perhaps we should have conducted our experiment on the CAPM in terms of excess returns—the difference between the stocks' monthly returns and the risk-free rates? In this section we perform this variation on the experiment and show that it does little to improve our analysis.

7. All is not lost! In Chapter 13 we examine the Black-Litterman model, which is a more positivist approach to portfolio choice.

Below we show the same Dow-Jones data, with an additional column appended for Treasury bill returns; these varied wildly over the period:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	EXCESS RETURN DATA FOR THE DOW-JONES INDUSTRIAL STOCKS AND THE STANDARD AND POORS 500 Monthly returns minus monthly Treasury bill return July 2001 - July 2006														
2	Date	Treasury bill return risk-free rate	S&P 500 Index ^GSPC	Alcoa AA	American International Group AIG	American Express AXP	Boeing BA	Citigroup C	Caterpillar CAT	DuPont DD	Disney DIS	General Electric GE	General Motors GM	Home Depot HD	Honeywell HON
3															
4	Average return		-0.22%	-0.38%	-0.83%	0.43%	0.37%	0.01%	1.50%	-0.11%	0.00%	-0.53%	-1.16%	-0.81%	0.00%
5	Beta		1.00	1.90	0.99	1.38	1.15	1.30	1.39	1.00	1.28	0.84	1.41	1.55	1.66
6	Alpha		0	0.04%	-0.61%	0.73%	0.63%	0.29%	1.81%	0.11%	0.28%	-0.34%	-0.86%	-0.47%	0.36%
7	R-squared		1	0.6085	0.3518	0.7052	0.2487	0.5972	0.5158	0.4362	0.3845	0.3221	0.2607	0.5288	0.5473
8															
9	01-Aug-01	0.29%	-6.92%	-2.78%	-6.49%	-10.49%	-13.38%	-9.17%	-9.97%	-3.88%	-3.87%	-6.45%	-14.49%	-9.40%	1.28%
10	04-Sep-01	0.28%	-8.82%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-18.33%	-34.76%
11	01-Oct-01	0.22%	1.50%											-0.65%	10.97%
12	01-Nov-01	0.18%	6.96%											19.72%	11.80%
13	03-Dec-01	0.16%	0.46%											8.65%	1.74%
14	02-Jan-02	0.14%	-1.86%											-2.11%	-0.92%
15	01-Feb-02	0.14%	-2.39%											-0.48%	12.85%
16	01-Mar-02	0.14%	3.32%											-3.02%	0.11%
17	01-Apr-02	0.15%	-6.63%											-5.00%	-4.55%
18	01-May-02	0.14%	-1.20%											-10.94%	6.84%
19	03-Jun-02	0.14%	-7.81%											-12.83%	-10.96%
20	01-Jul-02	0.14%	-8.52%											-17.65%	-8.81%
21	01-Aug-02	0.14%	0.19%											6.16%	-7.45%
22	03-Sep-02	0.14%	-11.95%											-23.41%	-32.66%
23	01-Oct-02	0.14%	8.00%											9.83%	9.71%
24	01-Nov-02	0.13%	5.26%											-9.27%	8.80%
25	02-Dec-02	0.10%	-6.52%											-9.51%	-8.34%
26	02-Jan-03	0.10%	-3.07%											-14.19%	1.50%
27	03-Feb-03	0.10%	-2.01%											11.20%	-6.01%
28	03-Mar-03	0.10%	0.54%											3.77%	-7.21%
29	01-Apr-03	0.09%	7.50%											14.09%	9.69%
30	01-May-03	0.09%	4.67%											14.14%	10.89%
31	02-Jun-03	0.09%	0.83%											1.81%	2.17%
32	01-Jul-03	0.08%	1.32%	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-6.29%	4.91%
33	01-Aug-03	0.08%	1.48%	3.09%	-7.78%	1.68%	12.36%	-3.60%	5.95%	2.31%	-6.94%	3.62%	10.41%	2.73%	2.84%
34	02-Sep-03	0.08%	-1.49%	-9.09%	-3.36%	-0.29%	-8.84%	4.56%	-4.54%	-11.47%	-1.96%	1.12%	-0.71%	-1.03%	-9.82%



Running the second-pass regression shows only minor changes from the results of section 11.2:

	A	B	C	D	E	F	G	H
1	THE SECOND-PASS REGRESSION FOR EXCESS RETURNS							
2	Stock	Average monthly excess return	Beta	Alpha				
3	Alcoa AA	-0.38%	1.9028	0.0004		Second-pass regression, regressing monthly returns on Beta		
4	American International Group AIG	-0.83%	0.9936	-0.0061		Intercept	0.0007	<-- =INTERCEPT(B3:B32,C3:C32)
5	American Express AXP	0.43%	1.3784	0.0073		Slope	-0.0020	<-- =SLOPE(B3:B32,C3:C32)
6	Boeing BA	0.37%	1.1515	0.0063		R-squared	0.0238	<-- =RSQ(B3:B32,C3:C32)
7	Citigroup C	0.01%	1.2952	0.0029				
8	Caterpillar CAT	1.50%	1.3903	0.0181		t-statistic, intercept	0.2439	<-- =tintercept(B3:B32,C3:C32)
9	DuPont DD	-0.11%	1.0009	0.0011		t-statistic, slope	-0.8254	<-- =tslope(B3:B32,C3:C32)
10	Disney DIS	0.00%	1.2805	0.0028				
11	General Electric GE	-0.53%	0.8420	-0.0034				
12	General Motors GM	-1.16%	1.4060	-0.0086				

11.7 Summary: Does the CAPM Have Any Uses?

Is the game lost? Do we have to give up on the CAPM? Not totally:

• First of all, it could be that the mean returns are approximately described by their regression on a market portfolio. In this alternative description of the CAPM, we claim (with some justification) that the β of an asset (which measures the dependence of the asset's returns on the market returns) is an important measure of the asset's risk.

• Second, the CAPM might be a good normative description of how to choose portfolios. As we showed in the appendix of Chapter 3, larger diversified portfolios are quite well described by their betas, so that the average beta of a well-diversified portfolio may be a reasonable description of the portfolio's risk.

Exercises

1. In a well-known paper, Roll (1978) discusses tests of the SML in a four-asset context:

Variance-covariance matrix					Returns
0.10	0.02	0.04	0.05		0.06
0.02	0.20	0.04	0.01		0.07
0.04	0.04	0.40	0.10		0.08
0.05	0.01	0.10	0.60		0.09

a. Derive two efficient portfolios in this 4-asset model and draw a graph of the efficient frontier.

b. Show that the following four portfolios are efficient by proving that each is a convex combination of the two portfolios you derived in part a above:

Security 1	0.59600	0.40700	-0.04400	-0.49600
Security 2	0.27621	0.31909	0.42140	0.52395
Security 3	0.07695	0.13992	0.29017	0.44076
Security 4	0.05083	0.13399	0.33242	0.53129

c. Suppose that the market portfolio is composed of equal proportions of each asset (i.e., the market portfolio has proportions (0.25,0.25,0.25,0.25)). Calculate the resulting SML. Is the portfolio (0.25,0.25,0.25,0.25) efficient?

d. Repeat this exercise, but substitute one of the four portfolios of part b above as the candidate for the market portfolio.

The remaining questions relate to a data set for 10 stocks. The data are given on the exercise file with this chapter.

	A	B	C	D	E	F	G	H	I	J	K	L
1	PRICE DATA: 10 STOCKS AND S&P 500, Jul2007 - Jul2012 S&P 500 represented by Vanguard's Index 500 fund (includes dividends)											
2		1	2	3	4	5	6	7	8	9	10	11
3		Apple	Google	Whole Foods	Seagate	Comcast	Merck	Johnson-Johnson	General Electric	Hewlett Packard	Goldman Sachs	S&P 500
4												
5	Date	AAPL	GOOG	WFM	STX	CMCSA	MRK	JNJ	GE	HPQ	GS	S&P 500
6	2-Jul-07	131.76	510.00	35.75	21.18	24.17	39.78	51.39	31.84	43.69	178.93	125.27
7	1-Aug-07	138.48	515.25	42.72	23.37	24.00	40.20	52.84	31.93	46.85	167.21	121.40
8	4-Sep-07	153.47	567.27	47.26	23.15	22.25	41.73	56.18	34.24	47.34	205.91	123.22
9	1-Oct-07	189.95	707.00	47.98	25.29	19.37	47.04	55.73	34.04	49.14	235.90	127.81
10	1-Nov-07	182.22	693.00	41.66	23.43	18.90	47.93	58.29	31.67	48.64	215.65	129.83
11	3-Dec-07	198.08	691.48	39.52	23.16	16.80	47.23	57.39	30.92	48.07	204.62	124.39
12	2-Jan-08	135.36	564.30	38.41	18.50	16.71	37.47	54.33	29.49	41.63	190.21	123.53
13	1-Feb-08	125.02	471.18	34.23	19.69	17.98	36.00	53.67	27.89	45.49	161.69	116.10
14	3-Mar-08	143.50	440.47	32.11	19.11	17.85	31.11	56.19	31.15	43.56	157.65	112.33

2. Fill in the template file below.

	A	B	C	D	E	F	G	H	I	J	K	L
1	**RETURN DATA: 10 STOCKS AND SP500**											
2		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
3		**Apple**	**Google**	**Whole Foods**	**Seagate**	**Comcast**	**Merck**	**Johnson-Johnson**	**General Electric**	**Hewlett Packard**	**Goldman Sachs**	**S&P 500**
4	**Monthly statistics**											
5	Mean											
6	Variance											
7	Sigma											
8												
9	**Annual statistics**											
10	Mean											
11	Variance											
12	Sigma											
13												
14	**Regressing individual assets on the S&P 500**											
15	Alpha											
16	Beta											
17	Rsq											
18	T-test, intercept											
19	T-test, slope											
20												
21												
22	**RETURN DATA**											
23	**Date**	**AAPL**	**GOOG**	**WFM**	**STX**	**CMCSA**	**MRK**	**JNJ**	**GE**	**HPQ**	**GS**	**SP500**
24	1-Aug-07											
25	4-Sep-07											
26	1-Oct-07											
27	1-Nov-07											
28	3-Dec-07											
29	2-Jan-08											
30	1-Feb-08											
31	3-Mar-08											
32	1-Apr-08											
33	1-May-08											
34	2-Jun-08											
35	1-Jul-08											

3. Perform the second-pass regression: Regress the monthly average returns on the betas of the assets. Does this confirm that the S&P 500 is efficient?

4. Compute the variance-covariance matrix for the 10 stocks. Using the monthly average returns and a monthly risk-free interest rate of 0.20%, compute an efficient portfolio. Here's the template:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**COMPUTING AN EFFICIENT PORTFOLIO OF THE 10 STOCKS**												
2													
3	**Variance-covariance matrix**												
4		**AAPL**	**GOOG**	**WFM**	**STX**	**CMCSA**	**MRK**	**JNJ**	**GE**	**HPQ**	**GS**		**Average returns**
5	**AAPL**												
6	**GOOG**												
7	**WFM**												
8	**STX**												
9	**CMCSA**												
10	**MRK**												
11	**JNJ**												
12	**GE**												
13	**HPQ**												
14	**GS**												
15													
16													
17	**Risk-free**	0.20%											
18													
19		**Efficient portfolio**											
20	**AAPL**												
21	**GOOG**												
22	**WFM**												
23	**STX**												
24	**CMCSA**												
25	**MRK**												
26	**JNJ**												
27	**GE**												
28	**HPQ**												
29	**GS**												

5. Using the efficient portfolio instead of the S&P 500:

   a. Compute the monthly returns on the efficient portfolio.

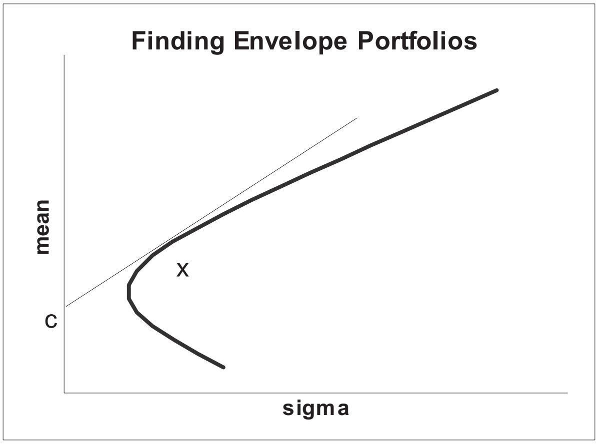
   b. Regress the average monthly returns of the stocks on their betas with respect to the efficient portfolio.

   c. Explain your results in light of Propositions 3 and 4 from Chapter 9.

12 Efficient Portfolios Without Short Sales

12.1 Overview

In Chapter 9 we discussed the problem of finding an efficient portfolio. As shown there, this problem can be written as finding a tangent portfolio on the envelope of the feasible set of portfolios:



The proof in the Appendix to Chapter 9 for solving for such an efficient portfolio involved finding the solution to the following problem:

$$\max \Theta = \frac{E(r_x) - c}{\sigma_p}$$

such that

$$\sum_{i=1}^{N} x_i = 1$$

where

$$E(r_x) = x^T \cdot R = \sum_{i=1}^{N} x_i E(r_i)$$

$$\sigma_p = \sqrt{x^T S x} = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j \sigma_{ij}}$$

Proposition 1 of Chapter 9 gives a methodology for solving this problem. Solutions to the maximization problem allow *negative* portfolio proportions;

when $x_i < 0$, this assumes that the *i*th security is sold short by the investor and that the proceeds from this short sale become immediately available to the investor.

Reality is, of course, considerably more complicated than this academic model of short sales. In particular, it is rare for all of the short-sale proceeds to become available to the investor at the time of investment, since brokerage houses typically escrow some or even all of the proceeds. It may also be that the investor is completely prohibited from making any short sales (indeed, most small investors seem to proceed on the assumption that short sales are impossible).[1]

In this chapter we investigate these problems. We show how to use Excel's **Solver** to find efficient portfolios of assets when we restrict short sales.[2]

12.2 A Numerical Example

We start with the problem of finding an optimal portfolio when no short sales are allowed. The problem we solve is similar to the maximization problem stated above, with the addition of the short-sales constraint $x_i > 0$ for the asset proportions:

$$\max \Theta = \frac{E(r_x) - c}{\sigma_p}$$

such that

$$\sum_{i=1}^{N} x_i = 1$$

$$x_i \geq 0, i = 1, \ldots, N$$

where

1. The actual procedures for implementing a short sale are not simple. A well-written academic survey is a recent paper by Gene D'Avolio, "The Market for Borrowing Stock," *Journal of Financial Economics* (2003). There's also a wonderful article, "Get Shorty," in the 1 December 2003 issue of *The New Yorker* magazine by James Surowiecki.

2. We do not go into the efficient set mathematics when short sales of assets are restricted. This involves the Kuhn-Tucker conditions, a discussion of which can be found in Edwin Elton, Martin Gruber, Stephen Brown, and W. N. Goetzmann, *Modern Portfolio Theory and Investment Analysis* (Wiley, 8th edition, 2009).

$$E(r_x) = x^T \cdot R = \sum_{i=1}^{N} x_i E(r_i)$$

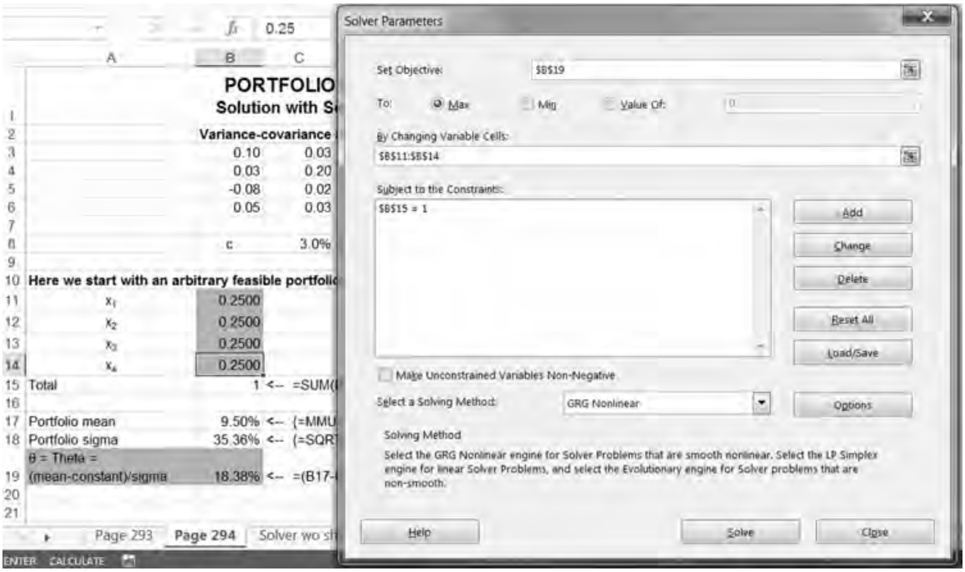
$$\sigma_p = \sqrt{x^T S x} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij}}$$

Solving an Unconstrained Portfolio Problem

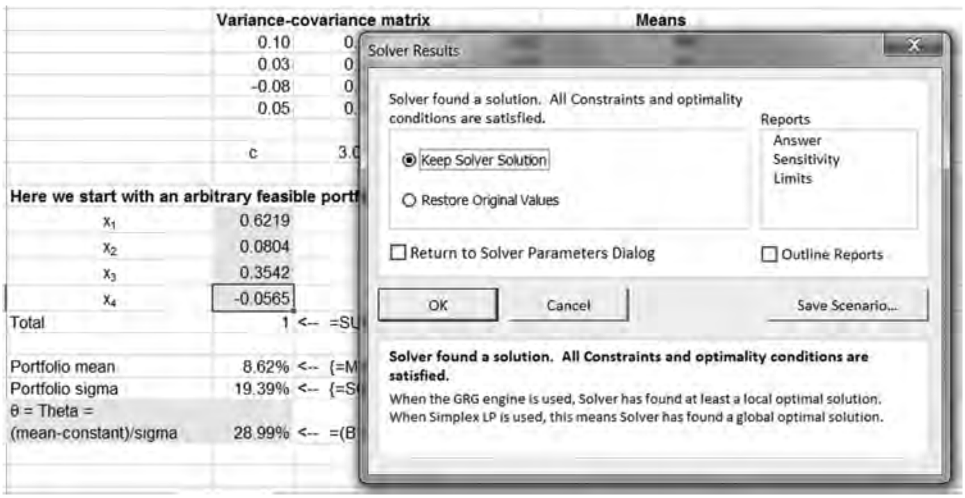
To set the scene, we consider the optimization problem below, which we solve without any short-sale constraints. The spreadsheet shows a four-asset variance-covariance matrix and associated expected returns. Given a constant $c = 8\%$, the optimal portfolio is given in cells B11:B14. Notice θ in cell B19: This is the *Sharpe ratio* of the portfolio, the ratio of its excess return over the constant c to its standard deviation: $\theta = \frac{E(r_x) - c}{\sigma_x}$. The optimal portfolio maximizes the Sharpe ratio θ.

	A	B	C	D	E	F	G	H
1	PORTFOLIO OPTIMIZATION ALLOWING SHORT SALES Follows Proposition 1, Chapter 9							
2		Variance-covariance matrix					Means	
3		0.10	0.03	-0.08	0.05		8%	
4		0.03	0.20	0.02	0.03		9%	
5		-0.08	0.02	0.30	0.20		10%	
6		0.05	0.03	0.20	0.90		11%	
7								
8		c	3.0%	<-- This is the constant				
9								
10	Optimal portfolio without short sale restrictions (Chapter 9, Proposition 1)							
11	x_1	0.6219	<-- {=MMULT(MINVERSE(B3:E6),G3:G6-C8)/SUM(MMULT(MINVERSE(B3:E6),G3:G6-C8))}					
12	x_2	0.0804						
13	x_3	0.3542						
14	x_4	-0.0565						
15	Total	1	<-- =SUM(B11:B14)					
16								
17	Portfolio mean	8.62%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}					
18	Portfolio sigma	19.39%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}					
19	θ = Theta = (mean-constant)/sigma	28.99%	<-- =(B17-C8)/B18					

There is another way to solve this unconstrained problem. Starting from an arbitrary portfolio (the spreadsheet below uses $x_1 = x_2 = x_3 = x_4 = 0.25$), we use **Solver** to find a solution:



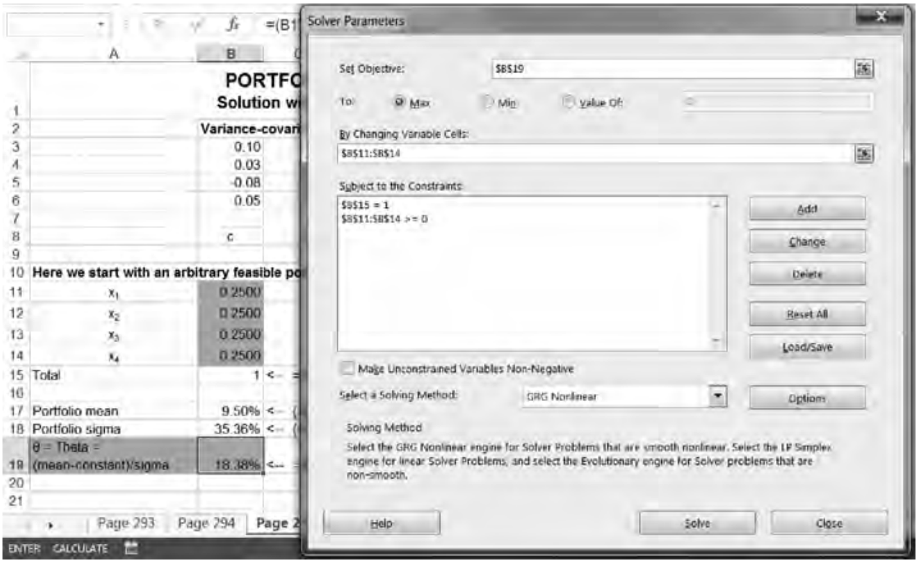
The **Solver** solution maximizes θ (cell B19) subject to the constraint that cell B15, which contains the sum of the portfolio positions, equals 1.[3] When we press **Solve** we get the solution we achieved before:



3. If **Tools|Solver** doesn't work, you may not have loaded the Solver add-in. To do so, go to **Tools|Add-ins** and click next to the **Solver Add-in**.

Solving a Constrained Portfolio Problem

The optimal solution above contains a short position in asset 4. To restrict the short selling, we add a no-short-sale constraint to **Solver**. Starting from an arbitrary solution, we bring up **Solver** as shown below:



Pressing **Solve** yields the following solution:

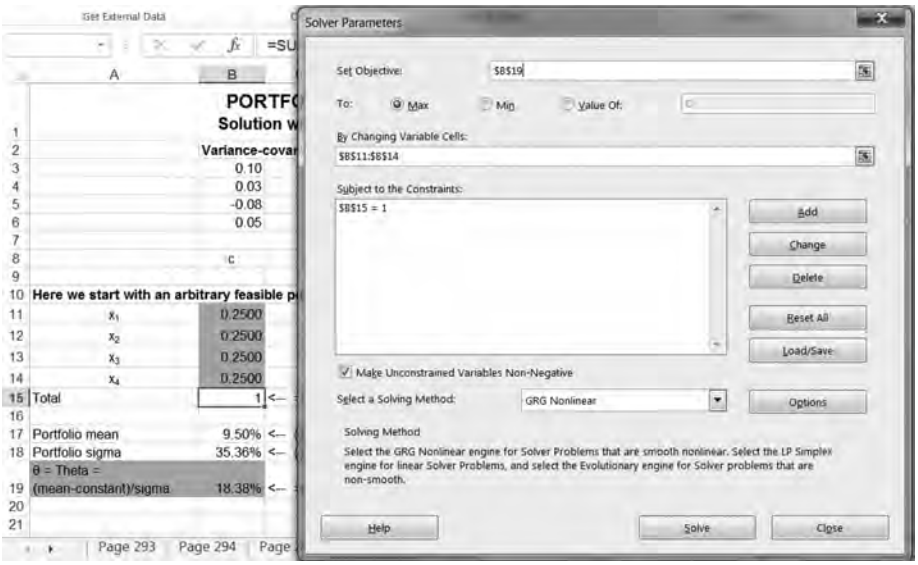
	A	B	C	D	E	F	G	H
1	**PORTFOLIO OPTIMIZATION WITHOUT SHORT SALES** **Solution with Solver, starting from an arbitrary feasible portfolio**							
2		**Variance-covariance matrix**					**Means**	
3		0.10	0.03	-0.08	0.05		8%	
4		0.03	0.20	0.02	0.03		9%	
5		-0.08	0.02	0.30	0.20		10%	
6		0.05	0.03	0.20	0.90		11%	
7								
8		c	3.0%	<-- This is the constant				
9								
10	**Here we start with an arbitrary feasible portfolio and use Solver**							
11	x_1	0.5856						
12	x_2	0.0965						
13	x_3	0.3179						
14	x_4	0.0000						
15	Total	1	<-- =SUM(B11:B14)					
16								
17	Portfolio mean	8.73%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}					
18	Portfolio sigma	20.32%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}					
19	θ = Theta = (mean-constant)/sigma	28.21%	<-- =(B17-C8)/B18					

The non-negativity constraints is added by clicking on the **Add** button in the **Solver** dialogue box. This brings up the following window (shown here filled in):

The second constraint (which constrains the portfolio proportions to sum to 1) is added in a similar fashion.

An Alternative Method

There's another way of doing this. **Solver** has an option: "Make Unconstrained Variables Non-Negative." Clicking this option gives the same result:



By changing the value of *c* in the spreadsheet, we can compute other portfolios; in the following example, we have set the constant $c = 8.5\%$:

	A	B	C	D	E	F	G	H
1	**PORTFOLIO OPTIMIZATION WITHOUT SHORT SALES** **Solution with Solver, starting from an arbitrary feasible portfolio**							
2		**Variance-covariance matrix**					**Means**	
3		0.10	0.03	-0.08	0.05		8%	
4		0.03	0.20	0.02	0.03		9%	
5		-0.08	0.02	0.30	0.20		10%	
6		0.05	0.03	0.20	0.90		11%	
7								
8		c	8.5%	<-- This is the constant				
9								
10	**Here we start with an arbitrary feasible portfolio and use Solver**							
11	x_1	0.0000						
12	x_2	0.2515						
13	x_3	0.4885						
14	x_4	0.2601						
15	Total	1	<-- =SUM(B11:B14)					
16								
17	Portfolio mean	10.01%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}					
18	Portfolio sigma	45.25%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}					
19	θ = Theta = (mean-constant)/sigma	3.33%	<-- =(B17-C8)/B18					

In both examples, the short-sale restriction is effective, with zero positions in some asset. However, not all values of *c* give portfolios in which the short-sale constraints are effective. For example, if the constant is 8%, we get:

	A	B	C	D	E	F	G	H
1	**PORTFOLIO OPTIMIZATION WITHOUT SHORT SALES** **Solution with Solver, starting from an arbitrary feasible portfolio**							
2		**Variance-covariance matrix**					**Means**	
3		0.10	0.03	-0.08	0.05		8%	
4		0.03	0.20	0.02	0.03		9%	
5		-0.08	0.02	0.30	0.20		10%	
6		0.05	0.03	0.20	0.90		11%	
7								
8		c	8.0%	<-- This is the constant				
9								
10	**Here we start with an arbitrary feasible portfolio and use Solver**							
11	x_1	0.2004						
12	x_2	0.2587						
13	x_3	0.4219						
14	x_4	0.1190						
15	Total	1	<-- =SUM(B11:B14)					
16								
17	Portfolio mean	9.46%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}					
18	Portfolio sigma	31.91%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}					
19	θ = Theta = (mean-constant)/sigma	4.57%	<-- =(B17-C8)/B18					

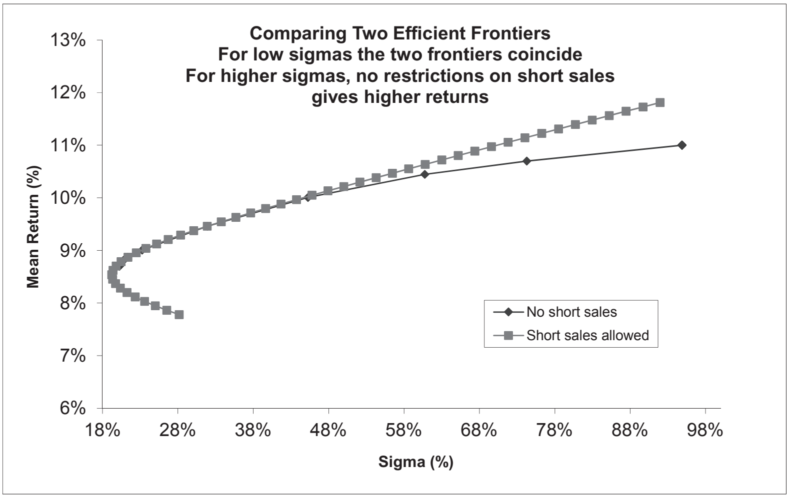
As we saw for the example where $c = 3\%$, as c gets lower, the short-sale constraint begins to be effective with respect to asset 4. For very high c's (the case below illustrates $c = 11\%$), only asset 4 is included in the maximizing portfolio:

	A	B	C	D	E	F	G	H
8		c	11.0%	<-- This is the constant				
9								
10	**Here we start with an arbitrary feasible portfolio and use Solver**							
11	x_1	0.0000						
12	x_2	0.0000						
13	x_3	0.0000						
14	x_4	1.0000						
15	Total	1	<-- =SUM(B11:B14)					
16								
17	Portfolio mean	11.00%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}					
18	Portfolio sigma	94.87%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}					
19	θ = Theta = (mean-constant)/sigma	0.00%	<-- =(B17-C8)/B18					

12.3 The Efficient Frontier with Short-Sale Restrictions

We want to graph the efficient frontier with short-sale restrictions. Recall that in the case of no short-sale restrictions discussed in Chapter 9, it was enough to find two efficient portfolios in order to determine the whole efficient frontier (this was proved in Proposition 2 of Chapter 9). When we impose short-sale restrictions, this statement is no longer true. In this case the determination of the efficient frontier requires the plotting of a large number of points. The only efficient (pardon the pun!) way of doing this is with a VBA program which repeatedly applies the **Solver** and puts the solutions in a table.

In section 12.3 we describe such a program. Once we have the program and the graph of the efficient frontier without short sales, we can compare this efficient frontier to the efficient frontier *with* short sales allowed:



The relation between these two graphs is not all that surprising:

• In general, the efficient frontier with short sales dominates the efficient frontier without short sales. This must clearly be so, since the short-sales restriction imposes an extra constraint on the maximization problem.

• For some cases, the two efficient frontiers coincide. One such point occurs, as we saw above, when $c = 8\%$.

Putting these two graphs on one set of axes shows that the effect of the short-sale restrictions is mainly for portfolios with higher returns and sigmas.

12.4 A VBA Program for the Efficient Frontier Without Short Sales

The output for the restricted short-sale case shown in section 12.3 was produced with the following VBA program:

```
Sub Solve()
       SolverOk SetCell:="$B$19", MaxMinVal:=1,
       ValueOf:="0", ByChange:="$B$11:$B$14"
       SolverSolve UserFinish:=True
End Sub
Sub Doit()
    Range("Results").ClearContents
    For counter = 1 To 40
        Range("constant") = -0.04 + counter * 0.005
        Solve
        Application.SendKeys ("{Enter}")
        Range("Results").Cells(counter, 1) _
        ActiveSheet.Range("constant")
        Range("Results").Cells(counter, 2) _
        ActiveSheet.Range("portfolio_sigma")
        Range("Results").Cells(counter, 3) _
        ActiveSheet.Range("portfolio_mean")
        Range("Results").Cells(counter, 4) _
        ActiveSheet.Range("x_1")
        Range("Results").Cells(counter, 5) _
        ActiveSheet.Range("x_2")
        Range("Results").Cells(counter, 6) _
        ActiveSheet.Range("x_3")
        Range("Results").Cells(counter, 7) _
        ActiveSheet.Range("x_4")
    Next counter
End Sub            ActiveSheet.Range("x_3")
        Range("Results").Cells(counter, 7) = _
            ActiveSheet.Range("x_4")
    Next counter
End Sub
```

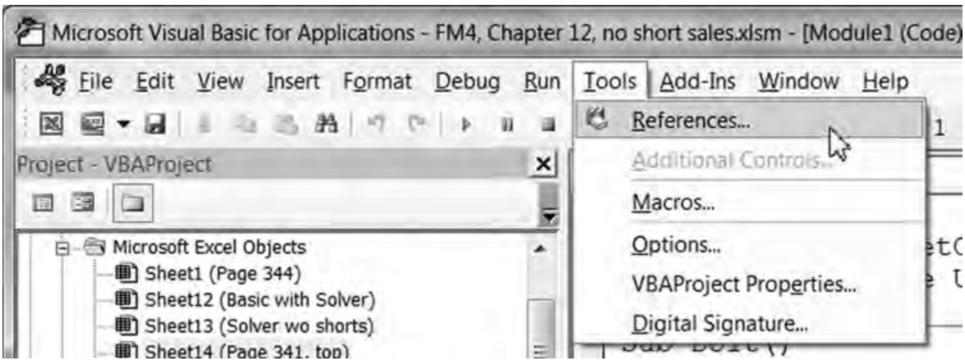
The program includes two subroutines: `Solve` calls the Excel Solver; and the subroutine `Doit` repeatedly calls the solver for different values of the range named `Constant` (this is cell C8 in the spreadsheet), putting the output in a range called "Results."

The final output looks like this:

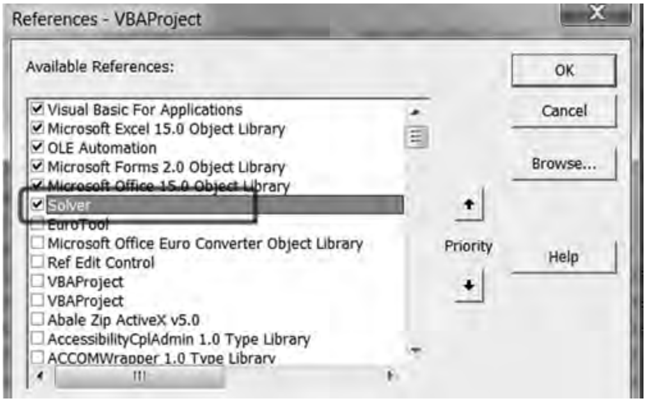
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	PORTFOLIO OPTIMIZATION WITHOUT SHORT SALES									RESULTS					
2		Variance-covariance matrix					Means			c	Sigma	Mean	x_1	x_2	x_3
3		0.10	0.03	-0.08	0.05		8%		Ctrl+A works the VBA program	-0.035	20.24%	8.70%	0.6049	0.0885	0.3066
4		0.03	0.20	0.02	0.03		9%		which calculates efficient	-0.03	20.25%	8.70%	0.6042	0.0887	0.3070
5		-0.08	0.02	0.30	0.20		10%		portfolios for no-short sales.	-0.025	20.25%	8.70%	0.6035	0.0890	0.3075
6		0.05	0.03	0.20	0.90		11%		This program iteratively	-0.02	20.25%	8.71%	0.6027	0.0893	0.3080
7									substitutes a constant ranging	-0.015	20.25%	8.71%	0.6017	0.0897	0.3086
8		c	16.0%	<-- This is the constant					from -3.5% 'till 16% (1/2%	-0.01	20.26%	8.71%	0.6007	0.0901	0.3092
9									jumps) and calculates the	-0.005	20.26%	8.71%	0.5994	0.0908	0.3098
10									optimal portfolio.	0	20.27%	8.71%	0.5982	0.0912	0.3106
11	x_1	0.0000	0							0.005	20.27%	8.71%	0.5968	0.0917	0.3115
12	x_2	0.0000	0							0.01	20.28%	8.72%	0.5950	0.0926	0.3123
13	x_3	0.0000	0							0.015	20.29%	8.72%	0.5932	0.0934	0.3134
14	x_4	1.0000	0							0.02	20.30%	8.72%	0.5910	0.0943	0.3147
15	Total	1.0000	<-- =SUM(B11:B14)							0.025	20.31%	8.73%	0.5885	0.0953	0.3161
16										0.03	20.32%	8.73%	0.5856	0.0965	0.3179
17	Portfolio mean	11.00%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}							0.035	20.34%	8.74%	0.5821	0.0980	0.3199
18	Portfolio sigma	94.87%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B3:E6,B11:B14)))}							0.04	20.37%	8.74%	0.5779	0.0998	0.3224
19	Theta	-5.27%	<-- =(B17-C8)/B18							0.045	20.41%	8.75%	0.5726	0.1019	0.3255
20										0.05	20.46%	8.76%	0.5659	0.1047	0.3294
21										0.055	20.54%	8.78%	0.5572	0.1083	0.3345
22										0.06	20.67%	8.80%	0.5452	0.1133	0.3415
23										0.065	20.90%	8.82%	0.5277	0.1205	0.3518
24										0.07	21.36%	8.87%	0.4992	0.1324	0.3684
25										0.075	23.27%	9.01%	0.4267	0.1630	0.3856

Adding a Reference to Solver in VBA

If the above routine does not work, you may need to add a reference to **Solver** in the VBA editor. Press [Alt] + F11 to get to the editor and then go to **Tools|References**:

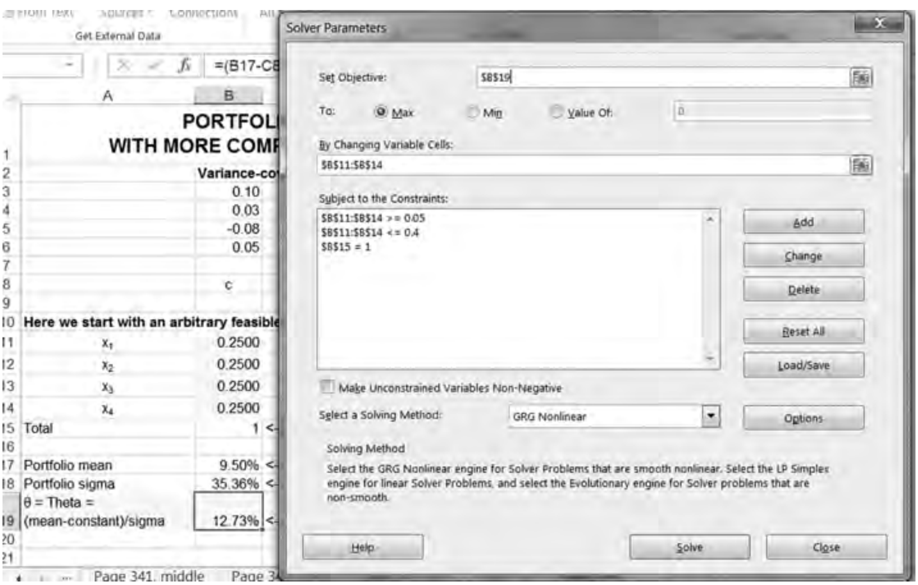


If this reference is missing, go to **Tools|References** on the VBA menu and make sure that **Solver** is checked:



12.5 Other Position Restrictions

It goes without saying that Excel and **Solver** can accommodate other position limits. Suppose, for example, that the investor wants at least 5% of her portfolio invested in any asset and no more than 40% of the portfolio invested in any single asset. This is easily set up in **Solver**:



This solves to give:

	A	B	C	D	E	F	G
1	**PORTFOLIO OPTIMIZATION WITH MORE COMPLICATED CONSTRAINTS**						
2		**Variance-covariance matrix**					**Means**
3		0.10	0.03	-0.08	0.05		8%
4		0.03	0.20	0.02	0.03		9%
5		-0.08	0.02	0.30	0.20		10%
6		0.05	0.03	0.20	0.90		11%
7							
8		c	5.0%	<-- This is the constant			
9							
10	**Here we start with an arbitrary feasible portfolio and use Solver**						
11	x_1	0.4000					
12	x_2	0.2270					
13	x_3	0.3230					
14	x_4	0.0500					
15	Total	1	<-- =SUM(B11:B14)				
16							
17	Portfolio mean	9.02%	<-- {=MMULT(TRANSPOSE(B11:B14),G3:G6)}				
18	Portfolio sigma	23.81%	<-- {=SQRT(MMULT(TRANSPOSE(B11:B14),MMULT(B				
19	θ = Theta = (mean-constant)/sigma	16.89%	<-- =(B17-C8)/B18				

12.6 Summary

No one would claim that Excel offers a quick way to solve for portfolio maximization, with or without short-sale constraints. However, it can be used to illustrate the principles involved, and the Excel **Solver** provides an easy-to-use and intuitive interface for setting up these problems.

Exercise

Given the data below:

a. Calculate the efficient frontier assuming no short sales are allowed.
b. Calculate the efficient frontier assuming that short sales are allowed.
c. Graph both frontiers on the same set of axes.

	A	B	C	D	E	F	G	H	I
3		A	B	C	D	E	F		Mean returns
4	A	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000		0.0100
5	B	0.0000	0.0400	0.0000	0.0000	0.0000	0.0000		0.0200
6	C	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000		0.0300
7	D	0.0000	0.0000	0.0000	0.1500	0.0000	0.0000		0.0400
8	E	0.0000	0.0000	0.0000	0.0000	0.2000	0.0000		0.0500
9	F	0.0000	0.0000	0.0000	0.0000	0.0000	0.3000		0.0550

13 The Black-Litterman Approach to Portfolio Optimization

13.1 Overview

Chapters 8–12 have set out the classic approach to portfolio optimization which was first explicated by Harry Markowitz in the 1950s and subsequently expanded by Sharpe (1964), Lintner (1965), and Mossin (1966). An enormous academic and practitioner literature (as well as several Nobel Prizes in Economics) testifies to the impact of this new point of view on asset valuation and portfolio choice. It is hardly an exaggeration to say that today no conversation about a stock's risk is complete without mentioning its beta, and discussions of portfolio performance regularly invoke the alpha (both of these topics are discussed in Chapter 11).

Markowitz, Sharpe, Lintner, and Mossin changed the paradigm of investment management. Well before Markowitz et al., individual investors knew that they should "diversify" and not "put all their eggs in one basket." But Markowitz and those who followed him gave statistical and implementational meaning to these clichés. Modern portfolio theory (MPT) changed the way intelligent investors discuss investment.

Nevertheless, MPT has disappointed. It is possible to come away from a standard textbook discussion of portfolio optimization with the impression that a fixed set of mechanical optimization rules, combined with a bit of knowledge about personal preferences, suffices to define an investor's optimal portfolio. Anyone who has tried to implement portfolio optimization using market data knows that the dream is often a nightmare. Implementations of portfolio theory produce wildly unrealistic portfolios, with huge short positions and correspondingly imaginary long positions. It might be thought that limiting short sales, as we have illustrated in Chapter 12, could solve some of these problems. However, short-sale limitations severely restrict the investible asset universe.

The main problem with the mechanical implementation of portfolio optimization is that historical asset return data produce bad predictions for future asset returns. The estimation of the covariances between asset returns and the estimation of expected returns—the underpinnings of portfolio theory—from historical data often produces unbelievable numbers.

In Chapter 10 we alluded to some of these problems in the context of estimating the variance-covariance matrix. There we showed that historical data may not be the best way to estimate this matrix—other methods, in particular the so-called "shrinkage" methods—may produce more reliable estimates of the covariances. In this chapter we take things a step further. We illustrate the problems of standard portfolio optimization by using a 10-asset portfolio problem. The MPT optimization for our data produces an insane "optimal" portfolio, with many huge long and short positions. The problems of portfolio optimization illustrated in our example are, unfortunately, not unusual. Using the data in a mechanical way to derive "optimal" portfolios simply doesn't work.[1]

In 1991 Fischer Black and Robert Litterman of Goldman Sachs published an approach which deals with many of the problematics of portfolio optimization.[2] Black and Litterman start with the assumption that an investor chooses his optimal portfolio from among a given group of assets. This group of assets—it might be the Standard & Poor's 500 Index, the Russell 2000, or a mix of international indices—defines the framework within which the investor chooses his portfolio. The investor's universe of assets defines a *benchmark* portfolio.

The Black-Litterman model takes as its starting point the assumption that, in the absence of additional information, the benchmark cannot be outperformed. This assumption is based on much research, which shows that it is very difficult to outperform a typical well-diversified benchmark.[3]

In effect the Black-Litterman model turns modern portfolio theory on its head—instead of inputting data and deriving an optimal portfolio, the BL

1. A paper by DeMiguel, Garlappi, and Uppal, "Optimal Versus Naive Diversification: How Inefficient Is the 1/N Portfolio Strategy?" in *The Review of Financial Studies* (2009), illustrates how badly mechanical optimization works. The authors examine optimal allocations among 10 sector portfolios. They find that a naive portfolio allocation rule of investing equal proportions in each portfolio—irrespective of market values—outperforms more sophisticated data-based optimizations.

2. Fischer Black and Robert Litterman, "Global Asset Allocation with Equities, Bonds, and Currencies," Goldman, Sachs & Co., Fixed Income Research (1991). Black and Litterman were apparently unaware of an earlier paper, which includes much of their research, by Sharpe, "Imputing Expected Security Returns from Portfolio Composition," *Journal of Financial and Quantitative Research* (1974). Sharpe's paper discusses the "reverse engineering" of the portfolio returns from portfolio composition.

3. "According to Lipper Analytical Services, over the 10 years ended in June 2000, more than 80% of 'general equity' mutual funds, meaning garden variety stock funds, underperformed the Standard and Poor's 500 Index—the major benchmark for stock mutual funds" (quoted in www.fool.com/Seminars/OLA/2001/Retire1_4C.htm). Or, from a well-known academic: "Professional investment managers, both in the U.S. and abroad, do not outperform their index benchmarks" Burton G. Malkiel, "Reflections on the Efficient Market Hypothesis: 30 Years Later," *The Financial Review* (2005).

approach assumes that a given portfolio is optimal and from this assumption derives the expected returns of the benchmark components. The implied vector of expected benchmark returns is the starting point of the BL model.

The BL implied asset returns can be interpreted as the market's information about the future returns of each asset in the benchmark portfolio. If our investor agrees with this market assessment, he's finished: He can then buy the benchmark, knowing that it is optimal. But what if he disagrees with one or more of the implied returns? BL shows how the investor's opinions can be incorporated into the optimization problem to produce a portfolio which is better for the investor.

In this chapter we start with an illustration of the problematics of MPT. We then go on to illustrate the Black-Litterman approach.

13.2 A Naive Problem

We start with a naive, though representative problem: The Super Duper Fund has set its benchmark portfolio to be a portfolio composed of 10 leading stocks. Joanna Roe, a new portfolio analyst for the Super Duper Fund, has decided to use portfolio theory to recommend optimal portfolio holdings based on this benchmark. The screen below gives 5 years of monthly price data on these stocks as of 1 July 2006 (note that some of the rows have been hidden):

	A	B	C	D	E	F	G	H	I	J	K	L
1	**PRICE AND MARKET CAP DATA FOR 10 COMPANIES**											
2		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM	
3	Market capitalization (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9	
4	Benchmark proportion	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%	<-- =K3/SUM($B$3:$K$3)
5												
6	Monthly price data (includes dividends)											
7	1-Jun-01	50.31	45.26	31.22	26.47	38.74	32.47	36.06	40.88	50.74	51.56	
8	2-Jul-01	49.72	48.26	35.64	22.83	35.61	33.82	35.37	36.29	53.98	50.55	
9	1-Aug-01	43.14	44.06	35.32	21.48	37.10	30.54	34.50	35.01	51.96	47.30	
10	4-Sep-01	33.81	36.79	30.66	14.92	37.79	24.37	28.06	32.07	53.16	44.71	
11	1-Oct-01	32.56	36.66	31.50	15.65	36.63	24.75	29.34	34.18	50.93	47.43	
12	1-Nov-01	39.63	44.78	35.37	20.45	36.92	27.68	35.09	38.21	54.07	52.33	
13	3-Dec-01	38.75	48.97	35.73	19.17	36.34	30.02	32.32	36.63	47.18	53.99	
14	2-Jan-02	40.77	48.09	36.99	20.64	39.71	30.22	32.59	38.06	47.48	50.70	
15	1-Feb-02	42.67	48.00	38.96	18.78	41.73	30.72	34.31	40.68	49.21	54.16	
16	1-Mar-02	48.69	46.71	38.30	16.81	42.21	34.52	34.47	40.95	46.46	52.81	
17	1-Apr-02	51.67	44.56	36.90	16.02	43.62	34.64	31.08	38.65	43.85	57.77	
61	1-Dec-05	19.01	40.17	33.12	28.48	73.08	51.23	29.30	41.77	31.10	76.60	
62	3-Jan-06	23.56	40.24	32.15	31.02	70.75	52.33	31.21	38.48	33.73	71.91	
63	1-Feb-06	20.11	41.83	32.53	32.64	70.32	53.76	29.19	39.91	34.09	73.20	
64	1-Mar-06	21.06	42.13	34.32	32.81	70.06	52.43	30.43	41.87	34.83	75.29	
65	3-Apr-06	22.66	39.77	36.09	32.38	72.34	53.81	33.63	43.74	34.03	84.98	
66	1-May-06	26.93	37.97	33.98	32.29	71.54	54.36	31.72	42.53	33.29	83.66	
67	1-Jun-06	29.79	35.79	32.30	31.68	73.43	53.22	32.36	41.60	36.43	80.77	

Row 3 gives the current total equity value of each of the benchmark stocks, and row 4 computes the benchmark proportions—the individual equity values divided by the total market capitalization of the benchmark.

Following the procedures described in Chapters 8 and 9, Joanna first transforms the price data to returns and then computes a variance-covariance matrix for these returns:

	A	B	C	D	E	F	G	H	I	J	K
1	**RETURN DATA FOR THE SUPER-DUPER BENCHMARK PORTFOLIO**										
2		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM
3	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9
4	Benchmark proportion	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%
5											
6	Mean return	-0.87%	-0.39%	0.06%	0.30%	1.07%	0.82%	-0.18%	0.03%	-0.55%	0.75%
7	Return sigma	10.78%	8.41%	6.23%	10.80%	8.71%	6.43%	9.54%	6.12%	8.06%	5.54%
8											
9	Date	GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM
10	02-Jul-01	-1.18%	6.42%	13.24%	-14.79%	-8.42%	4.07%	-1.93%	-11.91%	6.19%	-1.98%
11	01-Aug-01	-14.20%	-9.11%	-0.90%	-6.10%	4.10%	-10.20%	-2.49%	-3.59%	-3.81%	-6.65%
12	04-Sep-01	-24.37%	-18.03%	-14.15%	-36.44%	1.84%	-22.57%	-20.66%	-8.77%	2.28%	-5.63%
13	01-Oct-01	-3.77%	-0.35%	2.70%	4.78%	-3.12%	1.55%	4.46%	6.37%	-4.29%	5.91%
14	01-Nov-01	19.65%	20.01%	11.59%	26.75%	0.79%	11.19%	17.90%	11.15%	5.98%	9.83%
15	03-Dec-01	-2.25%	8.94%	1.01%	-6.46%	-1.58%	8.12%	-8.22%	-4.22%	-13.63%	3.12%
16	02-Jan-02	5.08%	-1.81%	3.47%	7.39%	8.87%	0.66%	0.83%	3.83%	0.63%	-6.29%
60	01-Sep-05	-11.06%	-5.56%	-3.48%	5.32%	5.28%	3.90%	-9.25%	-1.02%	-3.66%	3.05%
61	03-Oct-05	-11.10%	7.33%	-2.08%	-4.05%	1.81%	-0.78%	-0.54%	6.25%	3.63%	3.50%
62	01-Nov-05	-20.55%	2.02%	8.56%	5.66%	-3.06%	3.28%	12.70%	3.39%	5.40%	3.77%
63	01-Dec-05	-12.03%	-3.16%	6.39%	-3.32%	3.67%	0.08%	7.58%	-0.60%	7.86%	-1.26%
64	03-Jan-06	21.46%	0.17%	-2.97%	8.54%	-3.24%	2.12%	6.32%	-8.20%	8.12%	-6.32%
65	01-Feb-06	-15.83%	3.88%	1.18%	5.09%	-0.61%	2.70%	-6.69%	3.65%	1.06%	1.78%
66	01-Mar-06	4.62%	0.71%	5.36%	0.52%	-0.37%	-2.51%	4.16%	4.79%	2.15%	2.82%
67	03-Apr-06	7.32%	-5.76%	5.03%	-1.32%	3.20%	2.60%	10.00%	4.37%	-2.32%	12.11%
68	01-May-06	17.26%	-4.63%	-6.02%	-0.28%	-1.11%	1.02%	-5.85%	-2.81%	-2.20%	-1.57%
69	01-Jun-06	10.09%	-5.91%	-5.07%	-1.91%	2.61%	-2.12%	2.00%	-2.21%	9.01%	-3.52%

Naive Optimization

Using the return data, Joanna computes the sample variance-covariance matrix of excess returns as illustrated in Chapter 10. To implement the portfolio optimization, she needs data on the T-bill rate: The 1 July 2006 T-bill rate is 4.83% annually and 4.83%/12 = 0.40% monthly. Using the variance-covariance matrix, the T-bill rate, and the historical mean returns, she computes an "optimal" portfolio by solving the equation:

$$Optimal\ portfolio\{x_1, x_2, \ldots, x_{10}\} = \frac{S^{-1}\begin{bmatrix} \bar{r}_{GM} - r_f \\ \bar{r}_{HD} - r_f \\ \vdots \\ \bar{r}_{MMM} - r_f \end{bmatrix}}{[1, 1, \ldots, 1] * S^{-1} * \begin{bmatrix} \bar{r}_{GM} - r_f \\ \bar{r}_{HD} - r_f \\ \vdots \\ \bar{r}_{MMM} - r_f \end{bmatrix}}$$

$$= \frac{S^{-1}\begin{bmatrix} \bar{r}_{GM} - r_f \\ \bar{r}_{HD} - r_f \\ \vdots \\ \bar{r}_{MMM} - r_f \end{bmatrix}}{Sum\left[S^{-1} * \begin{bmatrix} \bar{r}_{GM} - r_f \\ \bar{r}_{HD} - r_f \\ \vdots \\ \bar{r}_{MMM} - r_f \end{bmatrix} \right]}$$

This portfolio is shown below (highlighted):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	SUPER DUPER BENCHMARK PORTFOLIO—NAIVE OPTIMIZATION												
2		General Motors GM	Home Depot HD	Internatio nal paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM		
3	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9		
4	Benchmark proportion	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%		
5													
6	**Variance-covariance matrix of excess returns**												
7		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM		Mean return
8	GM	0.0118	0.0031	0.0024	0.0042	0.0014	0.0033	0.0046	0.0018	0.0010	0.0014		-0.87%
9	HD	0.0031	0.0072	0.0019	0.0043	0.0022	0.0033	0.0046	0.0020	0.0002	0.0018		-0.39%
10	IP	0.0024	0.0019	0.0040	0.0031	0.0001	0.0024	0.0043	0.0021	0.0012	0.0016		0.06%
11	HPQ	0.0042	0.0043	0.0031	0.0119	0.0026	0.0049	0.0061	0.0033	0.0020	0.0022		0.30%
12	MO	0.0014	0.0022	0.0001	0.0026	0.0077	0.0016	0.0018	0.0009	0.0007	0.0008		1.07%
13	AXP	0.0033	0.0033	0.0024	0.0049	0.0016	0.0042	0.0038	0.0019	0.0011	0.0014		0.82%
14	AA	0.0046	0.0046	0.0043	0.0061	0.0018	0.0038	0.0093	0.0041	0.0018	0.0024		-0.18%
15	DD	0.0018	0.0020	0.0021	0.0033	0.0009	0.0019	0.0041	0.0038	0.0017	0.0019		0.03%
16	MRK	0.0010	0.0002	0.0012	0.0020	0.0007	0.0011	0.0018	0.0017	0.0066	0.0005		-0.55%
17	MMM	0.0014	0.0018	0.0016	0.0022	0.0008	0.0014	0.0024	0.0019	0.0005	0.0031		0.75%
18													
19	Current t-bill rate	0.40%	<-- =4.83%/12										
20													
21	"Optimal" portfolio												
22	GM	480.2%	<-- {=MMULT(MINVERSE(B8:K17),M8:M17-B19)/SUM(MMULT(MINVERSE(B8:K17),M8:M17-B19))}										
23	HD	981.8%											
24	IP	689.3%											
25	HPQ	221.3%											
26	MO	-263.7%											
27	AXP	-1763.7%											
28	AA	-324.5%											
29	DD	528.8%											
30	MRK	469.5%											
31	MMM	-918.9%											
32	Sum of proportions	1.00000	<-- =SUM(B22:B31)										

The "optimal" portfolio shown in cells B22:B31 is clearly not practically implementable: It contains too many large positions (both negative and positive). Note, for example, the –1763.7% short position in AXP and the 528.8% position in DD. Most mutual funds are prevented from taking short positions, and even funds which short sell will find it difficult to short sell 17.63 times the fund value in AXP or to invest 9.19 times the fund value in MMM. The enormous long positions which result from these short-sale positions (e.g., 9.82 times fund value invested long in HD) are similarly impracticable.

Why Does Naive Optimization Fail?

In some sense the strange portfolio "optimization" positions were predictable. The spreadsheet below highlights some disturbing features of the data which can partially explain the odd "optimized" portfolio.

- A number of the historical mean returns are negative. If we ignore the effects of correlations, a negative expected return should imply a short position in the stock.[4] There is, however, a deeper philosophical question about using past returns as proxies for future expected returns: Even though the past returns are negative, there is no reason to assume that this means that the future, expected returns from a stock should be negative. This is one of the problems when we use historical data to extract anticipations about the future.
- The correlations between asset returns are in some cases very large. Large correlations for a particular stock can lead us to prefer other stocks with smaller returns but more moderate correlations.

The spreadsheet below highlights stocks with negative historical returns and stocks whose correlations are greater than 0.5.

	A	B	C	D	E	F	G	H	I	J	K	L
1	NEGATIVE RETURNS AND HIGH CORRELATIONS											
2		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM	
3	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9	
4	Benchmark proportion	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%	
5												
6	Mean return	-0.87%	-0.39%	0.06%	0.30%	1.07%	0.82%	-0.18%	0.03%	-0.55%	0.75%	
7	Return sigma	10.78%	8.41%	6.23%	10.80%	8.71%	6.43%	9.54%	6.12%	8.06%	5.54%	
8												
9												
10		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM	
11	GM	1.0000	0.3320	0.3459	0.3534	0.1428	0.4676	0.4430	0.2737	0.1109	0.2277	<-- =CORREL($B$24:$B$83,K24:K83)
12	HD		1.0000	0.3512	0.4618	0.3012	0.6061	0.5618	0.3891	0.0260	0.3800	<-- =CORREL($C$24:$C$83,K24:K83)
13	IP			1.0000	0.4580	0.0159	0.5772	0.7181	0.5400	0.2362	0.4575	<-- =CORREL($D$24:$D$83,K24:K83)
14	HPQ				1.0000	0.2682	0.6965	0.5770	0.4924	0.2232	0.3666	
15	MO					1.0000	0.2839	0.2145	0.1647	0.0955	0.1645	
16	AXP						1.0000	0.6034	0.4855	0.2038	0.3798	
17	AA							1.0000	0.6863	0.2294	0.4525	
18	DD								1.0000	0.3287	0.5606	
19	MRK									1.0000	0.1079	
20	MMM										1.0000	

What About Changing the Variance-Covariance Matrix?

In Chapter 10 we discussed various methods for shrinking the variance-covariance matrix. "Shrinkage," you will recall, is a bit of jargon for taking a convex combination of the sample variance-covariance matrix with a diagonal matrix of only the variances. Shrinkage methods have been shown to be effective at improving the performance of the global minimum variance portfolio (GMVP).

4. In principle you might want to include a negative-return stock in your portfolio if it has a negative correlation with enough other stocks to lower the total portfolio variance. However, this rarely happens.

Will shrinkage help us solve the extreme positions of the naive portfolio optimization? We try this in the next spreadsheet, where cells B11:K20 contain a weighted combination of the sample covariance matrix (B41:K50) and an all-diagonal matrix of only the variances (B54:K63). The weight λ put on the sample covariance matrix is given in cell B7.

For $\lambda = 0.3$, the "optimal" portfolio indeed contains fewer extreme long and short positions. But it is clear that shrinkage can never solve the fundamental problem of the data—using negative historical returns as proxies for expected returns will always produce some negative portfolio positions in an optimizer. It is this problem which Black and Litterman solve and which we discuss in the next section.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**SUPER DUPER BENCHMARK PORTFOLIO—NAIVE OPTIMIZATION WITH SHRUNK VARIANCE-COVARIANCE MATRIX**												
2		General Motors GM	Home Depot HD	International paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM		
3	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9		
4	Benchmark proportion	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%		
5													
6	**Variance-covariance matrix of excess returns**												
7	Shrinkage factor, λ	0.3	<-- Weight on sample var-cov matrix										
8													
9	The matrix below is a weighted combination of the sample variance-covariance matrix and a pure diagonal matrix of only variances. {=B7*B39:K48+(1-B7)*B52:K61}												
10		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM		Mean return
11	GM	0.0118	0.0009	0.0007	0.0013	0.0004	0.0010	0.0014	0.0006	0.0003	0.0004		-0.87%
12	HD	0.0009	0.0072	0.0006	0.0013	0.0007	0.0010	0.0014	0.0006	0.0001	0.0005		-0.39%
13	IP	0.0007	0.0006	0.0040	0.0009	0.0000	0.0007	0.0013	0.0006	0.0004	0.0005		0.06%
14	HPQ	0.0013	0.0013	0.0009	0.0119	0.0008	0.0015	0.0018	0.0010	0.0006	0.0007		0.30%
15	MO	0.0004	0.0007	0.0000	0.0008	0.0077	0.0005	0.0005	0.0003	0.0002	0.0002		1.07%
16	AXP	0.0010	0.0010	0.0007	0.0015	0.0005	0.0042	0.0011	0.0006	0.0003	0.0004		0.82%
17	AA	0.0014	0.0014	0.0013	0.0018	0.0005	0.0011	0.0093	0.0012	0.0005	0.0007		-0.18%
18	DD	0.0006	0.0006	0.0006	0.0010	0.0003	0.0006	0.0012	0.0038	0.0005	0.0006		0.03%
19	MRK	0.0003	0.0001	0.0004	0.0006	0.0002	0.0003	0.0005	0.0005	0.0066	0.0001		-0.55%
20	MMM	0.0004	0.0005	0.0005	0.0007	0.0002	0.0004	0.0007	0.0006	0.0001	0.0031		0.75%
21													
22	Current T-bill rate	0.40%	<-- =4.83%/12										
23													
24	"Optimal" portfolio												
25	GM	84.3%	<-- {=MMULT(MINVERSE(B11:K20),M11:M20-B22)/SUM(MMULT(MINVERSE(B11:K20),M11:M20-B22))}										
26	HD	96.4%											
27	IP	50.2%											
28	HPQ	-3.6%											
29	MO	-76.1%											
30	AXP	-134.6%											
31	AA	32.5%											
32	DD	62.5%											
33	MRK	112.1%											
34	MMM	-123.8%											
35	Sum of proportions	1.0000	<-- =SUM(B25:B34)										

13.3 Black and Litterman's Solution to the Optimization Problem

The Black-Litterman (BL) approach provides an initial solution to the optimization problem above. The BL approach is composed of two parts:

Step 1: *What does the market think?* A vast amount of financial research shows that it is difficult to beat the returns of benchmark portfolios. The first step of the BL approach takes this research as a starting point. It assumes that the benchmark is optimal and derives the expected returns of each asset under this assumption. Another way of saying this is that in Step 1 we compute the expected returns of the assets which would make the investor choose the benchmark using the optimization techniques in Chapters 9–11.

Step 2: *Incorporating investor opinions.* In Step 1, BL shows how to compute the benchmark asset returns based on the assumption of optimality. Suppose the investor has divergent opinions from these market-based expected returns. Step 2 shows how to incorporate these opinions into the optimization procedure. Note that—because of the correlations between asset returns—an investor's opinion about any particular asset's returns will affect all the other expected returns. A critical part in Step 2 is to adjust *all* asset returns for an investor's opinion about any return.

An investor who follows the Black-Litterman procedure starts off by seeing what the market weights imply for the expected returns. He can then adjust these weights by adding his own opinions about any asset's expected returns. In the next two sections we discuss these two steps in detail.

13.4 BL Step 1: What Does the Market Think?

As shown in Chapter 9, an optimal portfolio must solve the equation:

$$\begin{bmatrix} \textit{Efficient} \\ \textit{portfolio} \\ \textit{proportions} \end{bmatrix} = \underbrace{\begin{bmatrix} \textit{Variance-} \\ \textit{covariance} \\ \textit{matrix} \end{bmatrix}^{-1} * \left\{ \begin{bmatrix} \textit{Expected} \\ \textit{portfolio} \\ \textit{returns} \end{bmatrix} - \begin{matrix} \textit{Risk-free} \\ \textit{rate} \end{matrix} \right\}}_{\uparrow \text{ Normalize to sum to 1}}$$

Solving the equation for the vector of expected portfolio returns, this means that an efficient portfolio must solve the following equation:

$$\begin{bmatrix} \textit{Expected} \\ \textit{portfolio} \\ \textit{returns} \end{bmatrix} = \begin{bmatrix} \textit{Variance-} \\ \textit{covariance} \\ \textit{matrix} \end{bmatrix} \begin{bmatrix} \textit{Efficient} \\ \textit{portfolio} \\ \textit{proportions} \end{bmatrix} * \begin{matrix} \textit{Normalizing} \\ \textit{factor} \end{matrix} + \begin{matrix} \textit{Risk-free} \\ \textit{rate} \end{matrix}$$

Joanna assumes that, in the absence of any additional knowledge or opinions about the market, the current market weights of the portfolio indicate the efficiency weights. She estimates that the expected benchmark return over the next month will be 1%, and uses this estimation to set the normalizing factor.[5] We illustrate below.

Solving the first part of the last equation (without the normalizing factor) gives:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**SUPER DUPER BENCHMARK PORTFOLIO—WHAT DOES THE MARKET THINK? No normalizing factor**												
2	Anticipated benchmark return	1.00%	<-- =12%/12										
3	Current T-bill rate	0.40%											
4													
5		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM		
6	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9		
7	Benchmark proportions	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%		
8													
9	**Variance-covariance matrix**												
10		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM		Without normalizing factor
11	GM	0.0118	0.0031	0.0024	0.0042	0.0014	0.0033	0.0046	0.0018	0.0010	0.0014		0.28%
12	HD	0.0031	0.0072	0.0019	0.0043	0.0022	0.0033	0.0046	0.0020	0.0002	0.0018		0.32%
13	IP	0.0024	0.0019	0.0040	0.0031	0.0001	0.0024	0.0043	0.0021	0.0012	0.0016		0.18%
14	HPQ	0.0042	0.0043	0.0031	0.0119	0.0026	0.0049	0.0061	0.0033	0.0020	0.0022		0.47%
15	MO	0.0014	0.0022	0.0001	0.0026	0.0077	0.0016	0.0018	0.0009	0.0007	0.0008		0.31%
16	AXP	0.0033	0.0033	0.0024	0.0049	0.0016	0.0042	0.0038	0.0019	0.0011	0.0014		0.28%
17	AA	0.0046	0.0046	0.0043	0.0061	0.0018	0.0038	0.0093	0.0041	0.0018	0.0024		0.38%
18	DD	0.0018	0.0020	0.0021	0.0033	0.0009	0.0019	0.0041	0.0038	0.0017	0.0019		0.22%
19	MRK	0.0010	0.0002	0.0012	0.0020	0.0007	0.0011	0.0018	0.0017	0.0066	0.0005		0.18%
20	MMM	0.0014	0.0018	0.0016	0.0022	0.0008	0.0014	0.0024	0.0019	0.0005	0.0031		0.16%
21													
22	Check: The expected return of the benchmark?	0.29%	<-- {=MMULT(B7:K7,M11:M20)}					Cells M11:M20 contain the array formula {=MMULT(B11:K20,TRANSPOSE(B7:K7)+B3)}					
23													
24													

5. One percent per month is equivalent to estimating an annual expected benchmark return of 12%.

Note that given the weights in row 7, the expected benchmark return is 0.29% per month as opposed to the 1% per month posited by Joanna (cell B2). To achieve this expected benchmark returns we multiply row 7 by a normalizing factor:

$$\begin{bmatrix} Benchmark \\ portfolio \\ returns \end{bmatrix} = \begin{bmatrix} Variance\text{-} \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} * \begin{matrix} Normalizing \\ factor \end{matrix} + \begin{matrix} Risk\text{-}free \\ rate \end{matrix}$$

$$= \begin{bmatrix} Variance\text{-} \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} * \underbrace{\left(\frac{Expected\ benchmark\ return - Risk\text{-}free\ rate}{\left(\begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix}^T \begin{bmatrix} Variance\text{-} \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} \right)} \right)}_{\substack{\uparrow \\ \text{The normalizing factor}}} + \begin{matrix} Risk\text{-}free \\ rate \end{matrix}$$

This procedure is illustrated below:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**SUPER DUPER BENCHMARK PORTFOLIO—WHAT DOES THE MARKET THINK?** **Normalizing factor computed in cell B4, based on anticipated benchmark return** **The expected returns make the benchmark optimal**												
2	Anticipated benchmark return	1.00%	<-- =12%/12										
3	Current T-bill rate	0.40%											
4	Normalizing factor	2.12	<-- {=(B2-B3)/MMULT(MMULT(B8:K8,B12:K21),TRANSPOSE(B8:K8))}										
5													
6		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM		
7	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9		
8	Benchmark proportions	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%		
9													
10	**Variance-covariance matrix**												
11		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM		With normalizing factor
12	GM	0.0118	0.0031	0.0024	0.0042	0.0014	0.0033	0.0046	0.0018	0.0010	0.0014		0.96%
13	HD	0.0031	0.0072	0.0019	0.0043	0.0022	0.0033	0.0046	0.0020	0.0002	0.0018		1.05%
14	IP	0.0024	0.0019	0.0040	0.0031	0.0001	0.0024	0.0043	0.0021	0.0012	0.0016		0.77%
15	HPQ	0.0042	0.0043	0.0031	0.0119	0.0026	0.0049	0.0061	0.0033	0.0020	0.0022		1.36%
16	MO	0.0014	0.0022	0.0001	0.0026	0.0077	0.0016	0.0018	0.0009	0.0007	0.0008		1.05%
17	AXP	0.0033	0.0033	0.0024	0.0049	0.0016	0.0042	0.0038	0.0019	0.0011	0.0014		0.97%
18	AA	0.0046	0.0046	0.0043	0.0061	0.0018	0.0038	0.0093	0.0041	0.0018	0.0024		1.17%
19	DD	0.0018	0.0020	0.0021	0.0033	0.0009	0.0019	0.0041	0.0038	0.0017	0.0019		0.84%
20	MRK	0.0010	0.0002	0.0012	0.0020	0.0007	0.0011	0.0018	0.0017	0.0066	0.0005		0.77%
21	MMM	0.0014	0.0018	0.0016	0.0022	0.0008	0.0014	0.0024	0.0019	0.0005	0.0031		0.73%
22													
23	Check: The expected return of the benchmark?	1.00%	<-- {=MMULT(B8:K8,M12:M21)}					Cells M12:M21 contain the array formula					
24								{=(MMULT(B12:K21,TRANSPOSE(B8:K8))*B4)+B3}					
25													
26													
27	Additional check: Optimal portfolio												
28	GM	2.71%	<-- {=MMULT(MINVERSE(B12:K21),M12:M21-B3)/SUM(MMULT(MINVERSE(B12:K21),M12:M21-B3))}										
29	HD	11.91%											
30	IP	2.56%											
31	HPQ	14.23%		Note that Chapter 9 optimization on the expected returns in cells M11:M20 and the variance-covariance matrix produces the market weights as the optimal portfolio.									
32	MO	24.69%											
33	AXP	10.57%											
34	AA	4.53%											
35	DD	6.17%											
36	MRK	12.80%											
37	MMM	9.81%											
38	Sum of proportions	100.0%	<-- =SUM(B28:B37)										

In the spreadsheet above, we have performed an additional check by deriving the optimal portfolio given the current T-bill rate of 0.40% and the expected returns in M12:M21. This should give us back the benchmark proportions in row 8—and it does!

13.5 BL Step 2: Introducing Opinions—What Does Joanna Think?

Having made two assumptions—(i) that the benchmark is efficient and (ii) that the expected benchmark return is 1% per month—Joanna has derived the expected returns for each of the benchmark components (cells M12:M21). We are now ready to introduce Joanna's opinions about asset returns. The rough idea is that if she disagrees with a market return, she can use the optimization procedure from Chapter 9 to derive a portfolio whose proportions differ from those of the benchmark.

We have to be careful, however: Because asset returns are correlated, any opinion Joanna has about one asset's returns will translate to an opinion about all other asset returns. To illustrate this point, suppose that Joanna thinks that the return on GM will be 1.1% over the next month instead of the market opinion of 0.96%. Then this translates to:

	A	B	C	D	E	F	G	H	I	J	K
1	ADJUSTING THE BENCHMARK FOR AN ANALYST'S OPINION In this example the only opinion is about GM										
2	Anticipated benchmark return	1.00%	<-- =12%/12								
3	Current T-bill rate	0.40%									
4	Normalizing factor	2.12	<-- {=(B2-B3)/MMULT(MMULT(B8:K8,B12:K21),TRANSPOSE(B8:K8))}								
5											
6		General Motors GM	Home Depot HD	International Paper IP	Hewlett-Packard HPQ	Altria MO	American Express AXP	Alcoa Aluminum AA	DuPont DD	Merck MRK	MMM
7	Market cap (billion $)	16.85	73.98	15.92	88.37	153.33	65.66	28.16	38.32	79.51	60.9
8	Benchmark proportions	2.71%	11.91%	2.56%	14.23%	24.69%	10.57%	4.53%	6.17%	12.80%	9.81%
9											
10	Variance-covariance matrix										
11		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM
12	GM	0.0118	0.0031	0.0024	0.0042	0.0014	0.0033	0.0046	0.0018	0.0010	0.0014
13	HD	0.0031	0.0072	0.0019	0.0043	0.0022	0.0033	0.0046	0.0020	0.0002	0.0018
14	IP	0.0024	0.0019	0.0040	0.0031	0.0001	0.0024	0.0043	0.0021	0.0012	0.0016
15	HPQ	0.0042	0.0043	0.0031	0.0119	0.0026	0.0049	0.0061	0.0033	0.0020	0.0022
16	MO	0.0014	0.0022	0.0001	0.0026	0.0077	0.0016	0.0018	0.0009	0.0007	0.0008
17	AXP	0.0033	0.0033	0.0024	0.0049	0.0016	0.0042	0.0038	0.0019	0.0011	0.0014
18	AA	0.0046	0.0046	0.0043	0.0061	0.0018	0.0038	0.0093	0.0041	0.0018	0.0024
19	DD	0.0018	0.0020	0.0021	0.0033	0.0009	0.0019	0.0041	0.0038	0.0017	0.0019
20	MRK	0.0010	0.0002	0.0012	0.0020	0.0007	0.0011	0.0018	0.0017	0.0066	0.0005
21	MMM	0.0014	0.0018	0.0016	0.0022	0.0008	0.0014	0.0024	0.0019	0.0005	0.0031
22											
23											
24	Expected benchmark returns, no opinions	Analyst opinion, delta, δ		Returns adjusted for opinions						Opinion-adjusted optimized portfolio	
25	0.96%	0.14%	GM	1.10%	<-- =A25+B12/$B$12*$B$25					7.85%	GM
26	1.05%	0.00%	HD	1.08%	<-- =A26+B13/$B$12*$B$25					11.28%	HD
27	0.77%	0.00%	IP	0.80%	<-- =A27+B14/$B$12*$B$25					2.43%	IP
28	1.36%	0.00%	HPQ	1.41%						13.48%	HPQ
29	1.05%	0.00%	MO	1.07%		Cells J25:J34 contain the array formula {=MMULT(MINVERSE(B12:K21),D25:D34-B3)/SUM(MMULT(MINVERSE(B12:K21),D25:D34-B3))}				23.39%	MO
30	0.97%	0.00%	AXP	1.00%						10.01%	AXP
31	1.17%	0.00%	AA	1.23%						4.30%	AA
32	0.84%	0.00%	DD	0.86%						5.84%	DD
33	0.77%	0.00%	MRK	0.78%						12.13%	MRK
34	0.73%	0.00%	MMM	0.75%						9.29%	MMM

The δ introduced in cell B25 above indicates Joanna's deviation from the Black-Litterman base case. In the example above, Joanna thinks that GM will differ from the market's return of 0.96% (cell A25) by $\delta_{GM} = 0.14\%$ (cell B25). What this example illustrates is that—because asset returns are correlated through the variance-covariance matrix—an opinion about one asset's returns (in this GM) also affects Joanna's anticipated returns for all other assets. Because of the covariance between asset returns, for example, this means that the HD return she expects is 1.08%:

$$r_{HD,opinion\ adjusted} = r_{HD,market} + \frac{Cov(r_{HD}, r_{GM})}{Var(r_{GM})}\delta_{GM} = 1.08\%$$

$$r_{IP,opinion\ adjusted} = r_{IP,market} + \frac{Cov(r_{IP}, r_{GM})}{Var(r_{GM})}\delta_{GM} = 0.80\%$$

and so on.

The newly optimized portfolio is given in cells J24:J33. Joanna's positive opinion about GM returns has, predictably, increased the proportion of GM in her portfolio. But her opinion about GM has also affected all the other portfolio weights:

	J	K	L
24	Opinion-adjusted optimized portfolio		Portfolio benchmark, no opinions
25	7.85%	GM	2.71%
26	11.28%	HD	11.91%
27	2.43%	IP	2.56%
28	13.48%	HPQ	14.23%
29	23.39%	MO	24.69%
30	10.01%	AXP	10.57%
31	4.30%	AA	4.53%
32	5.84%	DD	6.17%
33	12.13%	MRK	12.80%
34	9.29%	MMM	9.81%

The Black-Litterman Tracking Matrix

When Joanna has opinions about multiple stocks, the effect of these opinions on other stocks looks like a multivariate regression:

$$r_{GM,opinion\ adjusted} = r_{GM,market} + \frac{\sigma_{GM,GM}}{\sigma_{GM}^2}\delta_{GM} + \frac{\sigma_{HD,GM}}{\sigma_{GM}^2}\delta_{HD} + \frac{\sigma_{IP,GM}}{\sigma_{GM}^2}\delta_{IP} + \ldots + \frac{\sigma_{MMM,GM}}{\sigma_{GM}^2}\delta_{MMM}$$

$$r_{HD,opinion\ adjusted} = r_{HD,market} + \frac{\sigma_{GM,HD}}{\sigma_{HD}^2}\delta_{GM} + \frac{\sigma_{HD,HD}}{\sigma_{HD}^2}\delta_{HD} + \frac{\sigma_{IP,HD}}{\sigma_{HD}^2}\delta_{IP} + \ldots + \frac{\sigma_{MMM,HD}}{\sigma_{HD}^2}\delta_{MMM}$$

We define the Black-Litterman tracking matrix as:

$$BLtracking = \begin{bmatrix} \frac{\sigma_{GM,GM}}{\sigma_{GM}^2} & \frac{\sigma_{GM,HD}}{\sigma_{GM}^2} & \ldots & \frac{\sigma_{GM,MMM}}{\sigma_{GM}^2} \\ \frac{\sigma_{HD,GM}}{\sigma_{HD}^2} & \frac{\sigma_{HD,HD}}{\sigma_{HD}^2} & \ldots & \frac{\sigma_{HD,MMM}}{\sigma_{HD}^2} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\sigma_{MMM,GM}}{\sigma_{MMM}^2} & \frac{\sigma_{MMM,HP}}{\sigma_{MMM}^2} & \ldots & \frac{\sigma_{MMM,MMM}}{\sigma_{MMM}^2} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \frac{\sigma_{GM,HD}}{\sigma_{GM}^2} & \ldots & \frac{\sigma_{GM,MMM}}{\sigma_{GM}^2} \\ \frac{\sigma_{HD,GM}}{\sigma_{HD}^2} & 1 & \ldots & \frac{\sigma_{HD,MMM}}{\sigma_{HD}^2} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\sigma_{MMM,GM}}{\sigma_{MMM}^2} & \frac{\sigma_{MMM,HP}}{\sigma_{MMM}^2} & \ldots & 1 \end{bmatrix}$$

The opinion-adjusted returns are then:

$$\begin{bmatrix} r_{GM,\,market} \\ r_{HD,\,market} \\ \vdots \\ r_{MMM,\,market} \end{bmatrix} + \begin{bmatrix} 1 & \frac{\sigma_{GM,HD}}{\sigma_{GM}^2} & \cdots & \frac{\sigma_{GM,MMM}}{\sigma_{GM}^2} \\ \frac{\sigma_{HD,GM}}{\sigma_{HD}^2} & 1 & \cdots & \frac{\sigma_{HD,MMM}}{\sigma_{HD}^2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\sigma_{MMM,GM}}{\sigma_{MMM}^2} & \frac{\sigma_{MMM,HP}}{\sigma_{MMM}^2} & \cdots & 1 \end{bmatrix} * \begin{bmatrix} \delta_{GM} \\ \delta_{HD} \\ \vdots \\ \delta_{MMM} \end{bmatrix}$$

$$= \begin{bmatrix} r_{GM,\,opinion\text{-}adjusted} \\ r_{HD,\,market.opinion\text{-}adjusted} \\ \vdots \\ r_{MMM,\,opinion\text{-}adjusted} \end{bmatrix}$$

The VBA function **BLtracking** takes the variance-covariance matrix as its argument and is defined as:

```
  'BLtracking's argument is the variance-
  'covariance matrix
Function BLtracking(rng As Range) As Variant
  Dim i As Integer
  Dim j As Integer
  Dim numcols As Integer
  numcols = rng.Columns.Count
  Dim matrix() As Double
  ReDim matrix(numcols - 1, numcols - 1)
  For i = 1 To numcols
    For j = 1 To numcols
      matrix(i - 1, j - 1) = rng(i, j) _
        / rng(i, i)
  Next j
  Next i
  BLtracking = matrix
End Function
```

Here are the computations for our example:

	A	B	C	D	E	F	G	H	I	J	K
1	**BLACK-LITTERMAN TRACKING MATRIX Tracking factors = cov(i,j)/var(i)**										
2	**Tracking matrix**										
3		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM
4	GM	1.0000	0.2589	0.1999	0.3540	0.1153	0.2788	0.3920	0.1555	0.0829	0.1169
5	HD	0.4257	1.0000	0.2603	0.5934	0.3119	0.4635	0.6376	0.2834	0.0250	0.2501
6	IP	0.5985	0.4738	1.0000	0.7940	0.0222	0.5954	1.0994	0.5306	0.3056	0.4063
7	HPQ	0.3527	0.3594	0.2642	1.0000	0.2162	0.4145	0.5097	0.2791	0.1665	0.1878
8	MO	0.1769	0.2909	0.0114	0.3328	1.0000	0.2096	0.2351	0.1158	0.0884	0.1046
9	AXP	0.7843	0.7927	0.5595	1.1704	0.3845	1.0000	0.8956	0.4624	0.2556	0.3270
10	AA	0.5006	0.4950	0.4690	0.6533	0.1957	0.4066	1.0000	0.4404	0.1938	0.2625
11	DD	0.4820	0.5343	0.5496	0.8687	0.2342	0.5097	1.0694	1.0000	0.4327	0.5067
12	MRK	0.1483	0.0272	0.1826	0.2991	0.1032	0.1626	0.2715	0.2497	1.0000	0.0741
13	MMM	0.4437	0.5772	0.5151	0.7156	0.2588	0.4412	0.7802	0.6202	0.1571	1.0000
14		<-- {=bltracking(B18:K27)}									
15											
16	**Variance-covariance matrix**										
17		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM
18	GM	0.0118	0.0031	0.0024	0.0042	0.0014	0.0033	0.0046	0.0018	0.0010	0.0014
19	HD	0.0031	0.0072	0.0019	0.0043	0.0022	0.0033	0.0046	0.0020	0.0002	0.0018
20	IP	0.0024	0.0019	0.0040	0.0031	0.0001	0.0024	0.0043	0.0021	0.0012	0.0016
21	HPQ	0.0042	0.0043	0.0031	0.0119	0.0026	0.0049	0.0061	0.0033	0.0020	0.0022
22	MO	0.0014	0.0022	0.0001	0.0026	0.0077	0.0016	0.0018	0.0009	0.0007	0.0008
23	AXP	0.0033	0.0033	0.0024	0.0049	0.0016	0.0042	0.0038	0.0019	0.0011	0.0014
24	AA	0.0046	0.0046	0.0043	0.0061	0.0018	0.0038	0.0093	0.0041	0.0018	0.0024
25	DD	0.0018	0.0020	0.0021	0.0033	0.0009	0.0019	0.0041	0.0038	0.0017	0.0019
26	MRK	0.0010	0.0002	0.0012	0.0020	0.0007	0.0011	0.0018	0.0017	0.0066	0.0005
27	MMM	0.0014	0.0018	0.0016	0.0022	0.0008	0.0014	0.0024	0.0019	0.0005	0.0031

We can now use the tracking matrix to discuss the situation where there is more than one opinion.

Two or More Opinions

Suppose Joanna believes the GM monthly return will be 1.10% instead of its market return of 0.96% and the HD monthly return to be 1% instead of its equilibrium return of 1.05%. Joanna also believes that the expected returns on all other assets other than GM and HD are correct.

How can we reflect Joanna's opinions? We use the **BLtracking** matrix to compute the unique set of deltas that solves for the expected returns:

$$r_{HD,opinion\ adjusted} = r_{HD,market} + \frac{Cov(r_{HD}, r_{GM})}{Var(r_{GM})}\delta_{GM} = 1.08\%$$

$$r_{IP,opinion\ adjusted} = r_{IP,market} + \frac{Cov(r_{IP}, r_{GM})}{Var(r_{GM})}\delta_{GM} = 0.80\%$$

Here's the implementation in Excel:

	A	B	C	D	E	F	G	H	I	J	K	L
23	Expected benchmark returns, no opinions	Analyst opinion, delta		Joanna's opinions						Opinion-adjusted optimized portfolio		Portfolio benchmark, no opinions
24	0.96%	0.19%	GM	1.10%	<-- 0.011					10.35%	GM	2.71%
25	1.05%	-0.09%	HD	1.00%	<-- =1%					6.04%	HD	11.91%
26	0.77%	-0.01%	IP	0.77%	<-- =A26					1.28%	IP	2.56%
27	1.36%	0.00%	HPQ	1.36%	<-- =A27					14.09%	HPQ	14.23%
28	1.05%	0.01%	MO	1.05%	<-- =A28	Cells J24:J33 contain the array formula {=MMULT(MINVERSE(B11:K20),D24:D33-B3)/SUM(MMULT(MINVERSE(B11:K20),D24:D33-B3))}				25.07%	MO	24.69%
29	0.97%	-0.06%	AXP	0.97%	<-- =A29					9.65%	AXP	10.57%
30	1.17%	-0.04%	AA	1.17%						3.88%	AA	4.53%
31	0.84%	0.03%	DD	0.84%						7.16%	DD	6.17%
32	0.77%	-0.01%	MRK	0.77%						11.91%	MRK	12.80%
33	0.73%	0.01%	MMM	0.73%	<-- =A33					10.57%	MMM	9.81%
34		↑ {=MMULT(MINVERSE(tracking),D24:D33-A24:A33)}										
35												
36	**Tracking matrix**											
37		GM	HD	IP	HPQ	MO	AXP	AA	DD	MRK	MMM	
38	GM	1.0000	0.2589	0.1999	0.3540	0.1153	0.2788	0.3920	0.1555	0.0829	0.1169	
39	HD	0.4257	1.0000	0.2603	0.5934	0.3119	0.4635	0.6376	0.2834	0.0250	0.2501	
40	IP	0.5985	0.4738	1.0000	0.7940	0.0222	0.5954	1.0994	0.5306	0.3056	0.4063	
41	HPQ	0.3527	0.3594	0.2642	1.0000	0.2162	0.4145	0.5097	0.2791	0.1665	0.1878	
42	MO	0.1769	0.2909	0.0114	0.3328	1.0000	0.2096	0.2351	0.1158	0.0884	0.1046	
43	AXP	0.7843	0.7927	0.5595	1.1704	0.3845	1.0000	0.8956	0.4624	0.2556	0.3270	
44	AA	0.5006	0.4950	0.4690	0.6533	0.1957	0.4066	1.0000	0.4404	0.1938	0.2625	
45	DD	0.4820	0.5343	0.5496	0.8687	0.2342	0.5097	1.0694	1.0000	0.4327	0.5067	
46	MRK	0.1483	0.0272	0.1826	0.2991	0.1032	0.1626	0.2715	0.2497	1.0000	0.0741	
47	MMM	0.4437	0.5772	0.5151	0.7156	0.2588	0.4412	0.7802	0.6202	0.1571	1.0000	

Here are three comments:

- The δs in B24:B33 are computed so that the expected returns of GM and HD reflect Joanna's opinions and so that the expected returns of all other assets are unchanged from the initial computation.
- The formula for computing this δ-vector is given in row 34.
- The revised optimal portfolio—given the two opinions about GM and HD—is computed in J24:J33.

Two or More Opinions, a Different Interpretation

There is another interpretation to Joanna's opinions. Suppose she believes the GM monthly return will be 1.10% instead of its market return of 0.96% and the HD return to be 1% instead of its market return of 1.05%. Suppose further that she realizes that her two opinions reflect both on each other (i.e., that δ_{GM} influences the return on HD and vice versa) and on all the other returns.

With this alternative interpretation, we can easily solve for the new optimal portfolio:

	A	B	C	D	E	F	G	H	I	J	K	L
23	Expected benchmark returns, no opinions	Analyst opinion, delta		Monthly returns adjusted for opinions						Opinion-adjusted optimized portfolio		Portfolio benchmark, no opinions
24	0.96%	0.14%	GM	1.08%						6.19%	GM	2.71%
25	1.05%	-0.05%	HD	1.06%	<-- {=A24:A33+MMULT(tracking,B24:B33)}					6.71%	HD	11.91%
26	0.77%	0.00%	IP	0.83%						6.97%	IP	2.56%
27	1.36%	0.00%	HPQ	1.39%						9.10%	HPQ	14.23%
28	1.05%	0.00%	MO	1.06%		Cells J24:J33 contain the array				22.19%	MO	24.69%
29	0.97%	0.00%	AXP	1.04%		formula				18.88%	AXP	10.57%
30	1.17%	0.00%	AA	1.22%		{=MMULT(MINVERSE(B11:K20),D				0.19%	AA	4.53%
31	0.84%	0.00%	DD	0.88%		24:D33-				9.11%	DD	6.17%
32	0.77%	0.00%	MRK	0.79%		B3)/SUM(MMULT(MINVERSE(B11:				10.61%	MRK	12.80%
33	0.73%	0.00%	MMM	0.77%		K20),D24:D33-B3))}				10.05%	MMM	9.81%

We leave it to you to decide which of these conflicting versions of Joanna's opinions is correct. There is, of course, no "scientific" answer to this question.

Do You Believe Your Opinions?

Do we really believe our own opinions? Do we actually have confidence in what we believe? There is a whole theory of Bayesian adjustments to our beliefs, which is explained in Theil (1971).[6] An application to portfolio modeling can be found in Black and Litterman (1999) and other associated papers. This author finds these papers dauntingly complicated and difficult to implement. A simpler approach to the confidence question is to form a portfolio based on a convex combination of the market weights and the opinion-adjusted weights:

$$\begin{aligned} \textit{Portfolio proportions} &= (1-\gamma) * \textit{Market weights} \\ &\quad + \gamma * \textit{Opinion-adjusted weights} \end{aligned}$$

where γ is our degree of confidence in our opinions. An application to our last example is given below:

6. Henri Theil, *Principles of Econometrics* (Wiley, 1971).

	A	B	C	D	E	F	G	H	I	J	K	L
23	Expected benchmark returns, no opinions	Analyst opinion, delta		Returns adjusted for opinions						Opinion-adjusted optimized portfolio		Portfolio benchmark, no opinions
24	0.96%	0.19%	GM	1.10%	<-- 0.011					10.35%	GM	2.71%
25	1.05%	-0.09%	HD	1.00%	<-- =1%					6.04%	HD	11.91%
26	0.77%	-0.01%	IP	0.77%	<-- =A26					1.28%	IP	2.56%
27	1.36%	0.00%	HPQ	1.36%	<-- =A27					14.09%	HPQ	14.23%
28	1.05%	0.01%	MO	1.05%	<-- =A28	Cells J24:J33 contain the array formula {=MMULT(MINVERSE(B11:K20),D24:D33-B3)/SUM(MMULT(MINVERSE(B11:K20),D24:D33-B3))}				25.07%	MO	24.69%
29	0.97%	-0.01%	AXP	0.97%	<-- =A29					9.65%	AXP	10.57%
30	1.17%	-0.01%	AA	1.17%						3.88%	AA	4.53%
31	0.84%	0.01%	DD	0.84%						7.16%	DD	6.17%
32	0.77%	-0.01%	MRK	0.77%						11.91%	MRK	12.80%
33	0.73%	0.01%	MMM	0.73%	<-- =A33					10.57%	MMM	9.81%
34		↑ **{=MMULT(MINVERSE(B52:K61),D24:D33-A24:A33)}**										
35												
36	γ, opinion confidence	0.6	<-- Weight attached to analyst opinion									
37												
38	**Opinion and confidence-adjusted portfolio**											
39	GM	7.29%	<-- {=B36*J24:J33+(1-B36)*L24:L33}									
40	HD	8.39%										
41	IP	1.79%										
42	HPQ	14.15%										
43	MO	24.92%										
44	AXP	10.02%										
45	AA	4.14%										
46	DD	6.77%										
47	MRK	12.27%										
48	MMM	10.27%										

13.6 Using Black-Litterman for International Asset Allocation[7]

We end this chapter by implementing the BL model on data for five international indices. The spreadsheet below gives data on five major world stock market indices:

- The S&P 500, a value-weighted index of the 500 largest U.S. stocks.
- The MSCI World ex-US index: The Morgan Stanley Capital International (MSCI) World Ex-US Index comprises 21 developed countries based on GDP per capita.
- The Russell 2000 Index: The Russell 3000 Index, which is market cap weighted and captures about 98% of the investable U.S. marketplace. The Russell 2000 Index consists of the 2,000 smallest companies in the Russell 3000 Index.
- The MSCI Emerging Markets index: The Morgan Stanley Capital International (MSCI) Emerging Markets index consists of indices for 26 emerging economies.

7. I thank Steven Schoenfeld of Northern Trust for providing me with the data and some suggestions.

• The LB Global Aggregate index: This Lehman Brothers index covers the most liquid portion of the global investment-grade, fixed-rate bond market, including government, credit, and collateralized securities.

	A	B	C	D	E	F	G	H	I
1	**INDEX DATA, 2001-2005**								
2	**5 YEARS ENDING DEC05**								
3	**Correlation**	**S&P500**	**MSCI World ex-US**	**Russell 2000**	**MSCI Emerging**	**LB Global aggregate**		**Weight**	**Standard deviation**
4	**S&P500**	1.0000	0.8800	0.8400	0.8100	-0.1600		24%	14.90%
5	**MSCI World ex-US**	0.8800	1.0000	0.8300	0.8700	0.0700		26%	15.60%
6	**Russell 2000**	0.8400	0.8300	1.0000	0.8300	-0.1400		3%	19.20%
7	**MSCI Emerging**	0.8100	0.8700	0.8300	1.0000	-0.0500		3%	21.00%
8	**LB Global aggregate**	-0.1600	0.0700	-0.1400	-0.0500	1.0000		44%	5.80%
9								**100%**	

Column H gives the value weights on each index in a composite portfolio as of end of December 2005, and column I gives the standard deviation of each index component.

The Variance-Covariance Matrix

We use Excel array functions (Chapter 34) to compute the variance-covariance matrix for the five indices from the correlation matrix above

	A	B	C	D	E	F	G	H
12	Variance-covariance matrix: cells below contain formula {=I4:I8*TRANSPOSE(I4:I8)*B4:F8}							
13	**Variance-covariance matrix**	**S&P500**	**MSCI World ex-US**	**Russell 2000**	**MSCI Emerging**	**LB Global aggregate**		
14	**S&P500**	0.0222	0.0205	0.0240	0.0253	-0.0014		
15	**MSCI World ex-US**	0.0205	0.0243	0.0249	0.0285	0.0006		
16	**Russell 2000**	0.0240	0.0249	0.0369	0.0335	-0.0016		
17	**MSCI Emerging**	0.0253	0.0285	0.0335	0.0441	-0.0006		
18	**LB Global aggregate**	-0.0014	0.0006	-0.0016	-0.0006	0.0034		
19								
20	**Checks**							
21	First row of var-cov	0.0222	0.0205	0.0240	0.0253	-0.0014	<-- =I4*I8*F4	
22	Standard deviation of composite	8.72%	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(H4:H8),B14:F18),H4:H8))}					

The strange formula, **I4:I8*Transpose(I4:I8)*B4:F8**, in the cells B14:F19 is composed of two parts:

• **I4:I8*Transpose(I4:I8)** multiplies the column vector I4:I8 times its transpose. This is equivalent to multiplying

$$\begin{bmatrix} \sigma_{SP\,500} \\ \sigma_{MSCI\ World} \\ \sigma_{Russell\ 2000} \\ \sigma_{MSCI\ Emerging} \\ \sigma_{LB\ Global} \end{bmatrix} * \begin{bmatrix} \sigma_{SP\,500} & \sigma_{MSCI\ World} & \sigma_{Russell\ 2000} & \sigma_{MSCI\ Emerging} & \sigma_{LB\ Global} \end{bmatrix}$$

which—in the wonderful world of array functions—gives a matrix of the covariances:

$$\begin{bmatrix} \sigma^2_{SP\,500} & \sigma_{SP\,500}\sigma_{MSCI\,World} & \sigma_{SP\,500}\sigma_{Russell\,2000} & & \sigma_{SP\,500}\sigma_{LB\,Global} \\ \sigma_{MSCI\,World}\sigma_{SP\,500} & \sigma^2_{MSCI\,World} & \cdots & & \\ \vdots & & & & \\ \vdots & & & & \\ \sigma_{LB\,Global}\sigma_{SP\,500} & \sigma_{LB\,Global}\sigma_{MSCI\,World} & \cdots & \cdots & \sigma^2_{LB\,Global} \end{bmatrix}$$

• Multiplying the above by the matrix of correlations B4:F8 gives the variance-covariance matrix.

• Of course the whole array formula **I4:I8*Transpose(I4:I8)*B4:F8** is entered with [Ctrl] + [Shift] + [Enter].

In rows 21 and 22 we perform two checks on our computation: Row 21 contains brute-force computations of the first row of the variance-covariance matrix—just to make sure our array formula works as advertised. In cell B22 we compute the standard deviation of the five-index portfolio using their relative weights.

	A	B	C	D	E	F
25	Risk-free rate	5.00%				
26	Expected return on S&P 500	12.00%				
27						
28	**Black-Litterman implied returns**					
29	**S&P 500**	12.00%	<-- {=MMULT(B14:F18,H4:H8)*(B26-B25)/INDEX((MMULT(B14:F18,H4:H8)),1,1)+B25}			
30	**MSCI World ex-US**	12.97%				
31	**Russell 2000**	13.30%				
32	**MSCI Emerging**	14.45%				
33	**LB Global aggregate**	5.76%				

The Black-Litterman implied expected returns are based on three assumptions:

- The weighted portfolio of the five indices is mean-variance optimal.
- The anticipated risk-free rate is 5%.
- The expected return of the S&P 500 index is 12%.

Given these assumptions the expected returns on the five-index portfolio are given in cells B29:B33. Note the array formula given in these cells:

= MMult(B14:F18, H4:H8)*(B26 – B25)
/Index((MMult(B14:F18, H4:H8)), 1, 1) + B25

This formula uses the expected return on the S&P 500 to normalize the returns. It is equivalent to:

$$\begin{bmatrix} Benchmark \\ portfolio \\ returns \end{bmatrix} = \begin{bmatrix} Variance- \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} * \begin{matrix} Normalizing \\ factor \end{matrix} + \begin{matrix} Risk\text{-}free \\ rate \end{matrix}$$

$$= \begin{bmatrix} Variance- \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} * \underbrace{\left(\frac{SP500\ expected\ return - Risk\text{-}free\ rate}{\left(\begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix}^T \begin{bmatrix} Variance- \\ covariance \\ matrix \end{bmatrix} \begin{bmatrix} Benchmark \\ portfolio \\ proportions \end{bmatrix} \right)} \right)}_{\uparrow \ \text{The normalizing factor}} + \begin{matrix} Risk\text{-}free \\ rate \end{matrix}$$

What's the Upshot?

If we believe that the world portfolio is efficient (and in the absence of further information, there is little reason to believe otherwise), then the anticipated returns from each of its components are given by the Black-Litterman model, by the risk-free rate, and by an additional assumption on expected returns (in our case: the expected return of the S&P 500 index). The exercises to this chapter explore some other variations to the latter assumption.

13.7 Summary

Applying portfolio theory is not merely a matter of using historical market data to derive covariances and expected returns. Blindly applying sample data to derive optimal portfolios (as in section 13.1) usually leads to absurd results. The Black-Litterman approach gets around these absurdities by first assuming that—in the absence of analyst opinions and other information—the benchmark market weights and current risk-free interest rate correctly predict the future asset returns. The resulting asset returns can then be adjusted for opinions and confidence in opinions to derive an optimal portfolio.

Exercises

1. You have decided to create your own index of higher-beta components of the Dow-Jones 30 Industrials. Using Yahoo's stock screener, you come up with the data below.

   a. Compute the variance-covariance matrix of returns.

   b. Assuming that the risk-free rate is 5.25% annually (= 5.25%/12 = 0.44% monthly), and that the expected high-beta index annual return is 12% (= 1% monthly), compute the Black-Litterman monthly expected returns for each stock.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	HIGH-BETA INDEX FROM DOW-JONES 30 COMPONENTS														
2		3M Company MMM	Alcoa AA	American Express AXP	American International Group AIG	Caterpillar CAT	DuPont DD	Exxon XOM	Hewlett Packard HPQ	Home Depot HD	Honeywell HON	Intel INTC	IBM	McDonalds MCD	Merck MRK
3	Market cap ($B)	65.66	39.11	77.91	180.72	55.7	49.08	519.89	126.75	78.37	47.55	146.76	172.03	62.88	106.93
4	Beta	1.07	1.37	1.06	1.17	1.98	1.06	1.13	1.6	1.2	1.3	1.9	1.81	1.37	1.16
5															
6	**Stock prices**														
7	2-Jul-02	56.69	24.41	29.46	61.98	20.21	35.54	32.85	13.22	29.07	28.77	17.76	67.04	22.57	38.95
8	1-Aug-02	56.56	22.64	30.12	60.89	19.73	34.46	31.88	12.54	31.00	26.78	15.78	71.94	21.67	39.67
9	3-Sep-02	49.78	17.41	26.05	53.08	16.83	30.84	28.69	10.96	24.61	19.37	13.14	55.65	16.11	36.16
10	1-Oct-02	57.47	19.91	30.46	60.70	18.64	35.27	30.27	14.84	27.23	21.41	16.37	75.34	16.52	42.91
64	2-Apr-07	82.31	35.32	60.52	69.75	72.32	48.81	79.04	42.07	37.66	53.95	21.39	101.81	48.28	51.06
65	1-May-07	87.96	41.28	64.82	72.34	78.25	52.32	83.17	45.63	38.65	57.91	22.18	106.60	50.55	52.06
66	1-Jun-07	86.79	40.53	61.03	70.03	77.97	50.84	83.88	44.62	39.35	56.28	23.74	105.25	50.76	49.80
67	2-Jul-07	90.21	43.08	64.51	69.04	83.20	52.62	91.94	48.54	39.39	60.96	24.55	114.81	52.09	49.02

2. You are an analyst investing in the high-beta DJ30 portfolio from the previous problem. You believe that the monthly return of MMM will be 1%. What are your recommended optimal portfolio proportions?

3. Another analyst believes that HD will return only 0.5% per month over the next year. What are her recommended portfolio proportions?

14 Event Studies*

14.1 Overview

Event studies are some of the most powerful and widely used applications of the capital asset pricing model (CAPM) discussed in Chapters 8–11 of *Financial Modeling*. An event study is an attempt to determine whether a particular event in the capital market or in the life of a company affected a company's stock market performance. The event-study methodology aims to separate company-specific events from market- and/or industry-specific events, and has often been used as evidence for or against market efficiency.

An event study aims to determine if an event or announcement caused an abnormal movement in a company's stock price. The *abnormal returns* (AR) are calculated as the difference between a stock's actual return and its expected return, where the stock's expected return is typically measured using the market model, which relies only on a stock's market index to estimate its expected return.[1] Using the market model, we can measure the correlation between an individual stock's returns and its corresponding market returns. In some cases, we sum the abnormal returns to arrive at the *cumulative abnormal return* (CAR), which measures the total impact of an event through a particular time period, also called the event window.

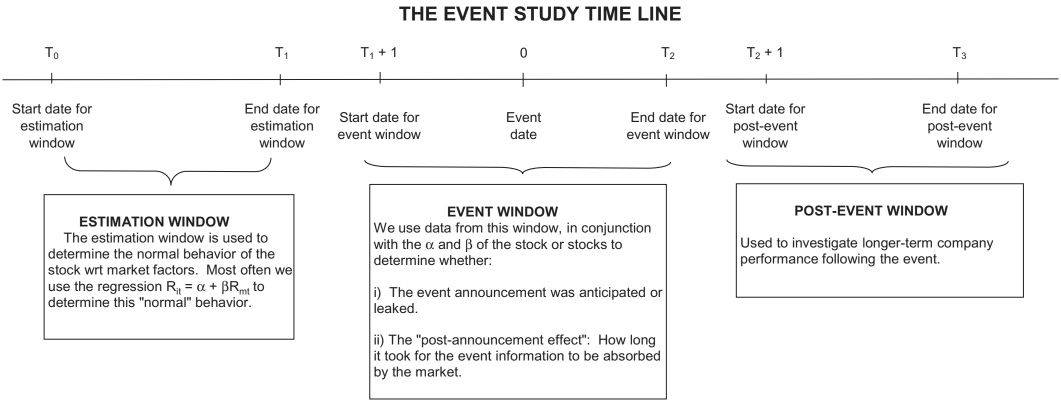
14.2 Outline of an Event Study

In this section we outline the methodology of an event study. In succeeding sections we apply the methodology to a number of different cases.

An event study is composed of three time frames: the *estimation window* (sometimes referred to as the control period), the e*vent window*, and the *post-event window*. The following chart illustrates these time frames.

* This chapter was co-authored with Dr. Torben Voetmann, Principal at the Brattle Group (Torben.Voetmann@brattle.com) and adjunct professor at the University of San Francisco.

1. Abnormal returns are also referred to as residual returns and both terms are used interchangeably throughout the chapter.



The time line illustrates the timing sequence of an event. The length of the estimation window is represented as T_0 to T_1. The event occurs at time 0 and the event window is represented as $T_1 + 1$ to T_2. The length of the post-event window is represented as $T_2 + 1$ to T_3.

An event is defined as a point in time when a company makes an announcement or when a significant market event occurs. For example, if we are studying the impact of mergers and acquisitions on the stock market, the announcement date is normally the point of interest. If we are examining how the market reacts to earnings restatements, the event window begins on the event date when a company announces its restatement. In practice, it is common to expand the event window to 2 trading days, the event date and the following trading day. This is done to capture the market movement if the event was announced immediately before the market closed or after market closing.

The *event window* often starts a few trading days before the actual event day. The length of the event window is centered on the announcement and is normally 3, 5, or 10. This enables us to investigate pre-event leakage of information.

The *estimation window* is also used to determine the normal behavior of a stock's return with respect to a market or industry index. The estimation of the stock's return in the estimation window requires us to define a model of "normal" behavior: Most often we use a regression model for this purpose.[2]

2. The regression model is similar to the first-pass regression discussed in Chapter 11. See further discussion below.

The usual length of the estimation window is 252 trading days (or 1 calendar year), but you may not always have this many days in your sample. If not, you need to determine whether the number of observations you do have is sufficient to produce robust results. As a guideline, you should have a minimum of 126 observations; if you have less than 126 observations in the estimation window it is possible that the parameters of the market model will not indicate the true stock price movements, and thus the relationship between the stock returns and the market returns. The estimation window that you select is supposedly a period that was free of any problems, i.e., a period that reflects the stock's normal price movements.[3]

The *post-event window* is most often used to investigate the performance of a company following announcements such as a major acquisition or an IPO. The *post-event window* allows us to measure the longer-term impact of the event. The post-event window can be as short as 1 month and as long as several years depending on the event.

Measuring the Stock's Behavior in the Estimation Window and the Event Window

As its name implies, the *estimation window* is used to estimate a model of the stock's returns under "normal" circumstances. The most common model used for this purpose is the market model, which is essentially a regression of the company's stock returns and the returns of the market index.[4] The market model for a stock i can be expressed as:

$$r_{it} = \alpha_i + \beta_i r_{Mt}$$

Here r_{it} and r_{Mt} represent the stock and the market return on day t. The coefficients α_i and β_i are estimated by running an ordinary least-square regression over the estimation window.

The most common criteria for selecting market and/or industry indices are whether the company is listed on NYSE/AMEX or Nasdaq and whether any restrictions are imposed by data availability. In general, the market index

3. Of course something will always be going on in such a long window—quarterly earnings announcements, dividend announcements, news about the companies being considered, etc. Our assumption is that these other events constitute at most "noise" and are not material for the event being studied.

4. Financial economists most often use the market model to estimate the expected return of a security, although they sometimes use the market-adjusted model or the two-factor market model. See an example of the two-factor model in section 14.3.

should be a broad-based value-weighted index or a float-weighted index. The industry index should be specific to the company being analyzed. For litigation purposes, it is common to construct the industry index instead of using alternative S&P 500 or MSCI indices. Most industry indices are available from Yahoo.[5]

Given the equation $r_{it} = \alpha_i + \beta_i r_{Mt}$ in the estimation window, we can now measure the impact of an event on the stock's return in the event window. For a particular day t in the event window, we define the stock's abnormal return as the difference between its actual return and its predicted return:

$$AR_{it} = \underbrace{r_{it}}_{\substack{\uparrow \\ \text{Actual stock} \\ \text{return in event} \\ \text{window day t}}} - \underbrace{(\alpha_i + \beta_i r_{Mt})}_{\substack{\uparrow \\ \text{Return predicted} \\ \text{by the stock's } \alpha, \\ \beta, \text{ and market return}}}$$

We interpret the abnormal return during the event window as a measure of the impact the event had on the market value of the security. This assumes that the event is exogenous with respect to the change in the security's market value.

The cumulative abnormal return is a measure of the total abnormal returns during the event window. CAR_t is the sum of all the abnormal returns from the beginning of the event window T_1 until a particular day t in the window:

$$CAR_t = \sum_{j=1}^{t} AR_{T_1+j}$$

Market-Adjusted and Two-Factor Models

As mentioned above, you can use several alternative models to calculate a security's expected return. The market-adjusted model is simplest in design and is often used to get a first impression of stock price movements. When using the market-adjusted model, you calculate the abnormal return by taking the difference between the actual return of the security and the actual return

5. Yahoo is probably not the best source for index data (though it is free!). A widely used source for industry data is Bloomberg. A wonderful free source of industry portfolio data is available from Fama-French at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

of the market index. Thus, there is no need to run ordinary least squares (OLS) regressions to estimate parameters. In fact, all you need is the returns at the time of the event. However, when testing the abnormal returns for statistical significance, you still need to gather returns for the estimation period.

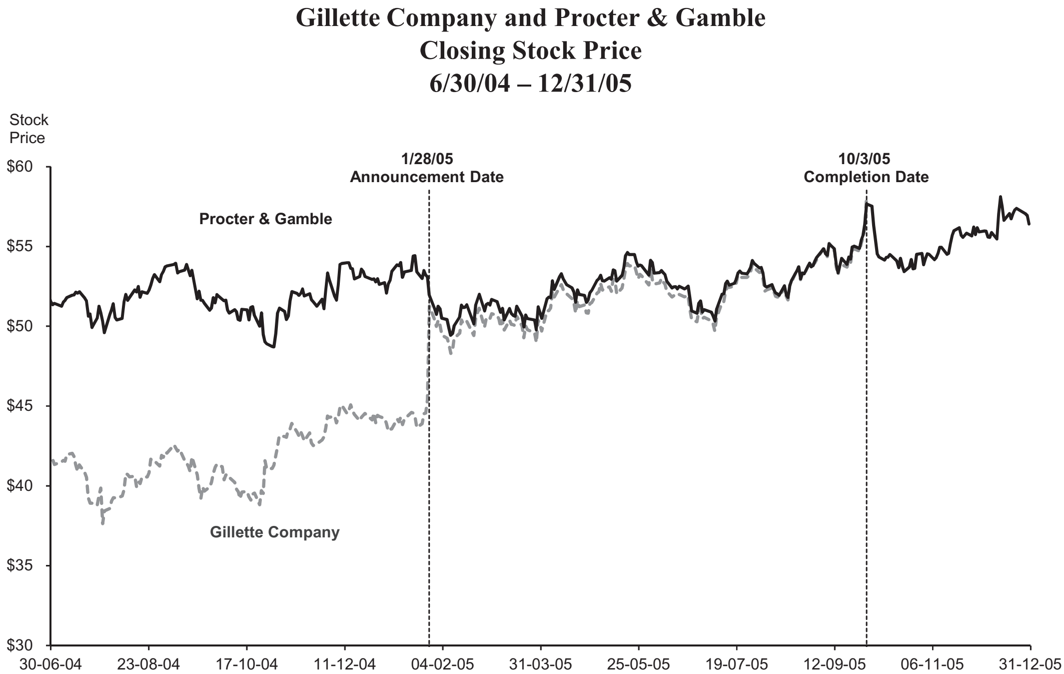
The two-factor model utilizes the returns from the market and the industry. You calculate a stock's expected return using parameters from a regression of the actual returns against the market and industry returns during the estimation period. The industry returns are included primarily to account for industry-specific information in addition to the market-specific information. To calculate the abnormal return you subtract from the actual return the portion that can be explained by the intercept, the market, and the industry. The two-factor model is illustrated in detail in section 14.3.

As Brown and Warner (1985) showed, the results in a large sample of events are not especially sensitive to your choice of estimation model.[6] However, if you are dealing with a small sample you should explore alternative models.

14.3 An Initial Event Study: Procter & Gamble Buys Gillette

On 28 January 2005, Procter & Gamble announced a bid for Gillette Company. As can be seen from the press release on page 341, the bid valued Gillette at a premium of 18% over its market price. As might be expected, the bid had a dramatic effect on Gillette's stock price:

6. Stephen Brown and Jerold Warner, "Using Daily Stock Returns: The Case of Event Studies," *Journal of Financial Economics* (1985).

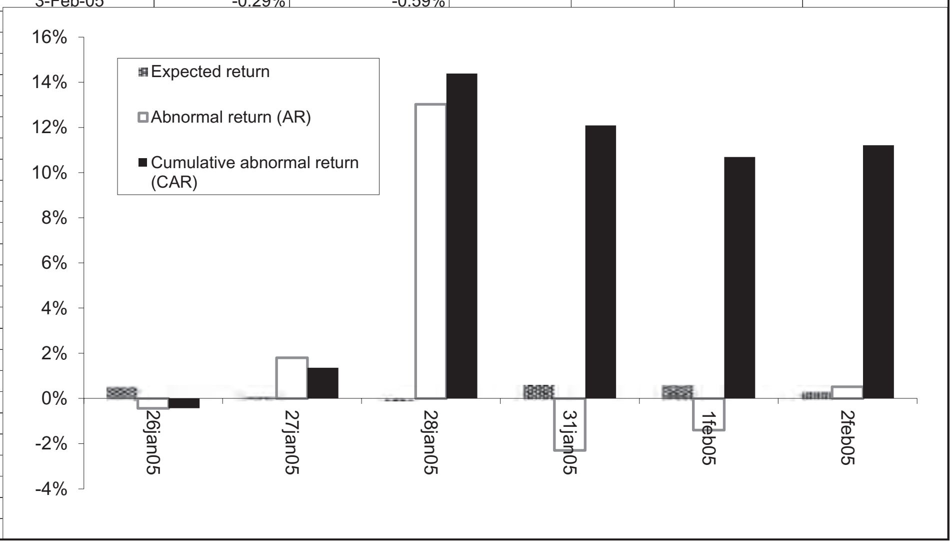


From the graph it appears that there might have also been a decrease in the price of Procter & Gamble.

The Estimation Window

We will attempt an event study to judge the impact of the takeover announcement on the returns of Gillette and Procter & Gamble. To do this, we first determine the estimation window as the 252 trading days preceding the 2 days before the announcement on 28 January 2005:

	A	B	C	D	E	F	G
1	**GILLETTE RETURNS: ESTIMATION WINDOW AND EVENT WINDOW**						
2	Intercept	0.0007	<-- =INTERCEPT($C$11:$C$262,$B$11:$B$262)				
3	Slope	0.6364	<-- =SLOPE($C$11:$C$262,$B$11:$B$262)				
4	R-squared	0.1315	<-- =RSQ($C$11:$C$262,$B$11:$B$262)				
5	Steyx	0.0113	<-- =STEYX($C$11:$C$262,$B$11:$B$262)				
6							
7	Days in estimation window	252	<-- =COUNT(A11:A262)				
8							
9				**EVENT WINDOW**			
10	**Date**	**NYSE**	**Gillette**	**Expected return**	**Abnormal return (AR)**	**Cumulative abnormal return (CAR)**	
11	27-Jan-04	-0.48%	-0.42%				
12	28-Jan-04	-1.26%	-1.27%				
13	29-Jan-04	0.00%	-0.94%				
14	30-Jan-04	-0.06%	-1.39%		Cell D263 contains formula =$B$2+$B$3*B263		
15	2-Feb-04	0.26%	-0.74%				
258	19-Jan-05	-0.78%	-0.09%				
259	20-Jan-05	-0.69%	-0.56%			Cell E263 contains formula =C263-D263	
260	21-Jan-05	-0.20%	-1.50%				
261	24-Jan-05	-0.18%	0.57%				
262	25-Jan-05	0.21%	1.44%				
263	26-Jan-05	0.68%	0.07%	0.50%	-0.44%	-0.44%	<-- =E263
264	27-Jan-05	0.04%	1.89%	0.09%	1.80%	1.36%	<-- =F263+E264
265	28-Jan-05	-0.24%	12.94%	-0.09%	13.03%	14.39%	<-- =F264+E265
266	31-Jan-05	0.82%	-1.71%	0.59%	-2.30%	12.09%	
267	1-Feb-05	0.80%	-0.83%	0.57%	-1.40%	10.69%	
268	2-Feb-05	0.32%	0.80%	0.27%	0.52%	11.21%	
269	3-Feb-05	-0.29%	-0.59%				



The regression results indicate the normal behavior of Gillette in the estimation window: $r_{Gillette,t} = 0.0007 + 0.6364r_{NYSE,t}$. The **Steyx** function measures the standard error of the regression-predicted y-values. Below we show how to use this value to measure the significance of the event's abnormal returns.

The Event Window

We define the event window as 2 days before and 3 days after the announcement. To measure the impact of the announcement effect in the event window we use the market model $r_{Gillette,t} = 0.0007 + 0.6346r_{NYSE,t}$. The formulas in the event window are in the spreadsheet above. As you can see, the announcement of the acquisition of Gillette by Procter & Gamble led to several large Gillette abnormal returns in the event window.

We can use **Steyx**, the standard error of the regression prediction to measure the significance of the abnormal returns. Only two of the abnormal returns—on the event date 28 January and the day following—are actually significant at the 5% level:

	A	B	C	D	E	F	G	H	I
1	GILLETTE RETURNS: THE SIGNIFICANCE OF THE ABNORMAL RETURNS								
2	Intercept	0.0007	<-- =INTERCEPT($C$11:$C$262,$B$11:$B$262)						
3	Slope	0.6364	<-- =SLOPE($C$11:$C$262,$B$11:$B$262)						
4	R-squared	0.1315	<-- =RSQ($C$11:$C$262,$B$11:$B$262)						
5	Steyx	0.0113	<-- =STEYX($C$11:$C$262,$B$11:$B$262)						
6									
7	Days in estimation window	252	<-- =COUNT(A11:A262)						
8									
9				EVENT WINDOW					
10	Date	NYSE	Gillette	Abnormal return (AR)	AR t-test	AR significant?			
11	27-Jan-04	-0.48%	-0.42%						
12	28-Jan-04	-1.26%	-1.27%						
13	29-Jan-04	0.00%	-0.94%						
14	30-Jan-04	-0.06%	-1.39%		Cell D263 contains formula =C263-($B$2+$B$3*B263)				
15	2-Feb-04	0.26%	-0.74%						
258	19-Jan-05	-0.78%	-0.09%						
259	20-Jan-05	-0.69%	-0.56%			Cell E263 contains formula =D263/$B$5			
260	21-Jan-05	-0.20%	-1.50%						
261	24-Jan-05	-0.18%	0.57%				Cell F263 contains formula =IF(ABS(E263)<1.96,"no","yes")		
262	25-Jan-05	0.21%	1.44%						
263	26-Jan-05	0.68%	0.07%	-0.44%	-0.39	no			
264	27-Jan-05	0.04%	1.89%	1.80%	1.59	no			
265	28-Jan-05	-0.24%	12.94%	13.03%	11.56	yes			
266	31-Jan-05	0.82%	-1.71%	-2.30%	-2.04	yes			
267	1-Feb-05	0.80%	-0.83%	-1.40%	-1.24	no			
268	2-Feb-05	0.32%	0.80%	0.52%	0.46	no			
269	3-Feb-05	-0.29%	-0.59%						

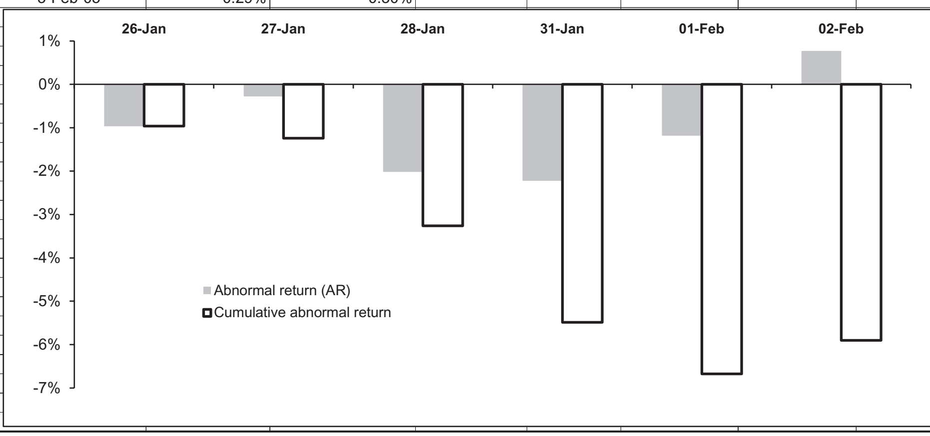
We calculate the test statistic by dividing the abnormal returns by the **Steyx** in cell B5. Assuming that the regression residuals are normally distributed, if the absolute value of the test statistic is larger than 1.96, then the abnormal return is significant at the 95% level (meaning that the chances that the abnormal return is random and insignificant are less than 5%). If the test statistic is larger than 2.58, its significance level is 1%. As can be seen from rows 263–268 above, at the 1% level, only the announcement itself has a significant abnormal return.[7]

What About Procter & Gamble?

Thus far we have concentrated on the event influence on the takeover target, Gillette. Applying the same methodology to Procter & Gamble's stock returns shows that the announcement had a negative impact on its stock returns. There may also have been some leakage of the information prior to the announcement on 28 January 2005:

7. One limitation of **Steyx** is that the variance is slightly understated. The true variance of the market model is the estimation variance from **Steyx** and the additional variance due to the sampling error in α_i and β_i. However, the sampling error approaches zero as the length of the estimation window increases. Since we suggest using 252 trading days in the estimation window, the effect of the sampling error is minimal and it is, therefore, often disregarded when calculating the variance of the abnormal returns.

	A	B	C	D	E	F	G	H
1	**PROCTER & GAMBLE RETURNS: ESTIMATION WINDOW AND EVENT WINDOW**							
2	Intercept	0.0004	<-- =INTERCEPT($C$11:$C$262,$B$11:$B$262)					
3	Slope	0.5877	<-- =SLOPE($C$11:$C$262,$B$11:$B$262)					
4	R-squared	0.1872	<-- =RSQ($C$11:$C$262,$B$11:$B$262)					
5	Steyx	0.0084	<-- =STEYX($C$11:$C$262,$B$11:$B$262)					
6								
7	Days in estimation window	252	<-- =COUNT(A11:A262)					
8								
9				**EVENT WINDOW**				
10	**Date**	**NYSE**	**Procter-Gamble**	**Abnormal return (AR)**	**AR t-test**	**AR significant?**	**Cumulative abnormal return**	
11	27-Jan-04	-0.48%	-0.65%					
12	28-Jan-04	-1.26%	-0.56%					
13	29-Jan-04	0.00%	2.41%					
14	30-Jan-04	-0.06%	0.08%		Cell D263 contains formula =C263-($B$2+$B$3*B263)			
15	2-Feb-04	0.26%	0.59%					
258	19-Jan-05	-0.78%	1.69%					
259	20-Jan-05	-0.69%	0.04%			Cell E263 contains formula =D263/$B$5		
260	21-Jan-05	-0.20%	-1.85%					
261	24-Jan-05	-0.18%	-0.79%				Cell F263 contains formula =IF(ABS(E263)<1.96,"no","yes")	
262	25-Jan-05	0.21%	0.94%					
263	26-Jan-05	0.68%	-0.52%	-0.96%	-1.14	no	-0.96%	<-- =D263
264	27-Jan-05	0.04%	-0.22%	-0.28%	-0.33	no	-1.25%	<-- =G263+D264
265	28-Jan-05	-0.24%	-2.12%	-2.02%	-2.39	yes	-3.26%	<-- =G264+D265
266	31-Jan-05	0.82%	-1.70%	-2.23%	-2.64	yes	-5.49%	
267	1-Feb-05	0.80%	-0.68%	-1.19%	-1.41	no	-6.67%	
268	2-Feb-05	0.32%	1.00%	0.77%	0.91	no	-5.90%	
269	3-Feb-05	-0.29%	-0.30%					



News Releases

P&G Signs Deal to Acquire The Gillette Company

Raises Long-Term Sales Growth Outlook

CINCINNATI, AND BOSTON, Jan. 28, 2005 /PRNewswire-FirstCall/ -- The Procter & Gamble Company (NYSE: PG) announced it has signed a deal to acquire 100% of The Gillette Company (NYSE: G). Gillette, founded in 1901 and headquartered in Boston, Mass., markets a number of category-leading consumer products such as Gillette(R) razors and blades including the Mach3(R) and Venus(R) brands, Duracell(R) CopperTop(R) batteries, Oral-B(R) manual and power toothbrushes, and Braun shavers and small appliances. The transaction is valued at approximately $57 billion (USD) making it the largest acquisition in P&G history.

Terms of the Deal

Under terms of the agreement, unanimously approved by the board of directors of both companies on January 27, P&G has agreed to issue 0.975 shares of its common stock for each share of Gillette common stock. Based on the closing share price of P&G and Gillette stock on January 27, 2005, this represents an 18% premium to Gillette shareholders.

P&G will acquire all of Gillette's business, including manufacturing, technical and other facilities. The transaction, which is subject to certain conditions including approval by Gillette's and P&G's shareholders and regulatory clearance, is expected to close in fall 2005.

Summarizing: What Happened on the Announcement Date?

On 28 January 2005, Procter & Gamble announced the purchase of Gillette. Each share of Gillette was purchased for 0.975 shares of Procter & Gamble. At the 5% significance level, the acquisition announcement had significant effects on the stock prices of Gillette and Procter & Gamble only on the announcement date and the day after. After an initial positive impact on Gillette (a 13.03% increase in the normally anticipated stock return on the event date 28 January and a further –2.30% on 31 January) and an initial negative impact on Procter & Gamble (–2.02% on 28 January and –2.23% on 31 January), there were no additional significant effects on the stock prices around the announcement. The cumulative effects are summarized below:

	A	B	C	D	E	F	G	H	I
1	**GILLETTE PURCHASED BY PROCTER & GAMBLE** **Measuring the synergies in event window**								
2		**Shares outstanding (thousands)**	**Share price, 25jan05**	**Market value, 25jan05 (billion $)**					
3	Gillette	1,000,000	44.53	44.53					
4	P&G	2,741,000	53.49	146.62					
5					Cell F12 contains formula				
6					=$D$3*SUM($B$12:B12)/1000				
7									
8							Cell G12 contains formula		
9							=$D$4*SUM($C$12:C12)/1000		
10	**Date**	**Abnormal returns (AR)**				**Cumulative abnormal valuations (billion $)**			
11		Gillette	P&G	Sum		Gillette	P&G	Sum	
12	26-Jan-05	-0.44%	-0.96%	-1.40%		-0.19	-1.41	-1.61	<-- =F12+G12
13	27-Jan-05	1.80%	-0.28%	1.52%		0.61	-1.83	-1.22	
14	28-Jan-05	13.03%	-2.02%	11.01%		6.41	-4.78	1.63	
15	31-Jan-05	-2.30%	-2.23%	-4.52%		5.38	-8.04	-2.66	
16	1-Feb-05	-1.40%	-1.19%	-2.59%		4.76	-9.78	-5.02	
17	2-Feb-05	0.52%	0.77%	1.29%		4.99	-8.66	-3.66	

The table above attempts to measure the short-term synergies of the announcement by multiplying the CAR for Gillette and P&G times their market value on the day before the event window. In the short period of time measured by this event window, the cumulative synergy appears to be negative, with the positive value creation for Gillette shareholders outweighed by the negative impact on P&G.[8]

14.4 A Fuller Event Study: Impact of Earnings Announcements on Stock Prices

In the previous section we used the event-study methodology to explore the impact of a merger announcement on the returns of both the takeover target (Gillette) and acquirer (Procter & Gamble). In this section we show how to aggregate the returns of an event in order to evaluate the market response to a particular type of event. We consider the effect of earning announcements on a set of stores in the grocery industry.

8. A study commissioned by the Massachusetts Secretary of the Commonwealth William F. Galvin after the merger offer suggests that the synergies of the merger, between $22 and $28 billion, were largely captured by Procter & Gamble. See the report and an article from *Business Week* on the disk that accompanies this book.

An Initial Example: Safeway's Positive Earnings Surprise on 20 July 2006

To set the stage, consider the earnings announcement made by Safeway on 20 July 2006. On this date Safeway announced earnings per share (EPS) of $0.42, a number which exceeded by 6 cents the market consensus estimate of $0.36.[9] On the same day, the S&P 500 declined by 0.85%, and Safeway stock rose by 8.39%. The spreadsheet below shows the example of Safeway's earnings announcement on 20 July 2006. We use the event-study methodology to gauge the market reaction to this earnings surprise:

	A	B	C
1	**THE MARKET REACTION TO A POSITIVE EARNINGS SURPRISE BY SAFEWAY, 20 July 2006**		
2	Announcement date	20-Jul-06	
3	Earnings per share	$0.42	
4	Consensus earnings estimate	$0.36	
5	Earnings surprise (forecast error)	$0.06	
6			
7	**How did the market interpret the earnings surprise?**		
8	Safeway	8.39%	
9	S&P 500	-0.85%	
10			
11	**Regressing Safeway returns on the S&P returns using the market model: Safeway = 0.0001 + 0.9289*SP**		
12	Intercept	0.0001	<-- =INTERCEPT(OFFSET('Stock
13	Slope	0.9289	Prices'!$A$2,138,8,252,1),OFFSET('Stock
14	Steyx	0.0118	Prices'!$A$2,138,2,252,1))
15			
16	**The residual return: Expected stock returns versus actual returns**		
17	Expected return	-0.78%	<-- =B13*B9+B12
18	Residual return	9.17%	<-- =B8-B17
19	t-statistic	7.75	<-- =B18/B14

In cells B12:B14 we have regressed Safeway daily returns on those of the S&P 500 for the 252 trading days preceding the announcement. The regression shows that $r_{Safeway} = 0.0001 + 0.9289 * r_{S\&P500}$ and that the standard error of the estimate is 0.0118.

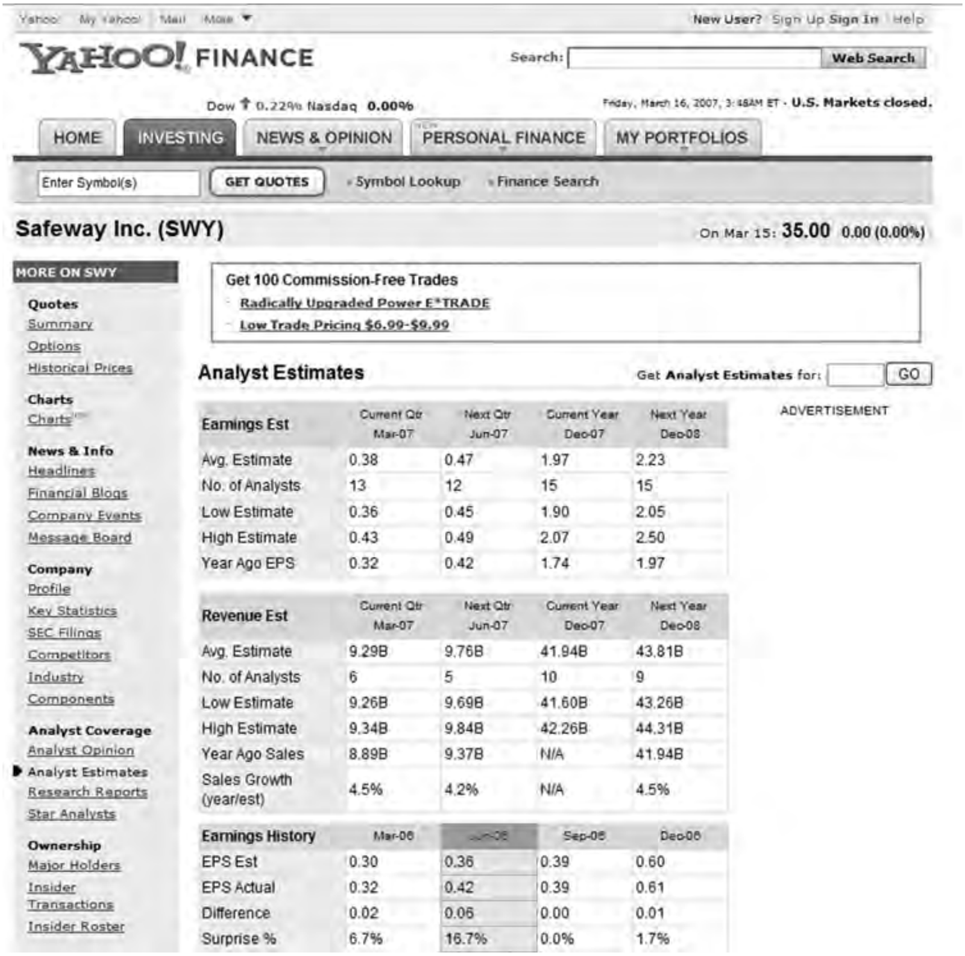
Given these data, we can see that on the day of the announcement the anticipated return of Safeway, given the negative 0.85% return of the S&P and absent the earnings surprise, should have been −0.78%. This means that the abnormal return, measuring the impact of the earnings announcement, was

9. Yahoo is the source of our earnings surprise data (see below).

9.17% (cell B18, above). The *t*-statistic for the return was 7.75, showing that it is highly significant.

The Earnings Surprise Numbers

Our earnings surprise numbers are drawn from Yahoo, as shown in the following screenshot. While Yahoo is an admirable source of data, it does not provide a database of historical analyst estimates and actual earnings numbers; such data are available from Bloomberg's Best Consensus Earnings Estimates and other commercial sources.



Yahoo! My Yahoo! Mail More New User? Sign Up Sign In Help

YAHOO! FINANCE Search: Web Search

Dow 0.22% Nasdaq 0.00% Friday, March 16, 2007, 3:48AM ET - U.S. Markets closed.

HOME INVESTING NEWS & OPINION PERSONAL FINANCE MY PORTFOLIOS

Enter Symbol(s) GET QUOTES Symbol Lookup Finance Search

Safeway Inc. (SWY) On Mar 15: 35.00 0.00 (0.00%)

MORE ON SWY

Quotes: Summary, Options, Historical Prices

Charts: Charts

News & Info: Headlines, Financial Blogs, Company Events, Message Board

Company: Profile, Key Statistics, SEC Filings, Competitors, Industry, Components

Analyst Coverage: Analyst Opinion, Analyst Estimates, Research Reports, Star Analysts

Ownership: Major Holders, Insider Transactions, Insider Roster

Get 100 Commission-Free Trades
- Radically Upgraded Power E*TRADE
- Low Trade Pricing $6.99-$9.99

Analyst Estimates Get Analyst Estimates for: GO

ADVERTISEMENT

Earnings Est	Current Qtr Mar-07	Next Qtr Jun-07	Current Year Dec-07	Next Year Dec-08
Avg. Estimate	0.38	0.47	1.97	2.23
No. of Analysts	13	12	15	15
Low Estimate	0.36	0.45	1.90	2.05
High Estimate	0.43	0.49	2.07	2.50
Year Ago EPS	0.32	0.42	1.74	1.97

Revenue Est	Current Qtr Mar-07	Next Qtr Jun-07	Current Year Dec-07	Next Year Dec-08
Avg. Estimate	9.29B	9.76B	41.94B	43.81B
No. of Analysts	6	5	10	9
Low Estimate	9.26B	9.69B	41.60B	43.26B
High Estimate	9.34B	9.84B	42.26B	44.31B
Year Ago Sales	8.89B	9.37B	N/A	41.94B
Sales Growth (year/est)	4.5%	4.2%	N/A	4.5%

Earnings History	Mar-06	Jun-06	Sep-06	Dec-06
EPS Est	0.30	0.36	0.39	0.60
EPS Actual	0.32	0.42	0.39	0.61
Difference	0.02	0.06	0.00	0.01
Surprise %	6.7%	16.7%	0.0%	1.7%

An Event Study: The Grocery Industry

We extend our Safeway study by considering 16 quarterly earnings announcements by four grocery companies for fiscal year 2006.[10]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	EARNINGS ANNOUNCEMENT IN 2006-07 FOR KROGER (KR), SUPERVALU (SVU), SAFEWAY (SWY), AND WHOLE FOODS (WFMI)																
2	**Calendar Date**	**Ticker**	**Consensus**	**Actual**	**Surprise: =D3-C3**		**Starting Point**	**Intercept**	**Slope**	**STEXY**		**Actual Return**	**S&P 500 Return**	**Expected Return: =H3+I3*M3**	**Abnormal Return: =L3-N3**	**t-stat**	
3	7-Mar-06	KR	0.36	0.39	0.03		44	0.0003	0.6662	0.0123		1.36%	-0.19%	-0.09%	1.46%	1.1808	<-- =O3/J3
4	20-Jun-06	KR	0.42	0.42	0.00		117	0.0004	0.6063	0.0113		5.05%	0.00%	0.04%	5.01%	**4.4355**	<-- =O4/J4
5	12-Sep-06	KR	0.29	0.29	0.00		175	0.0006	0.5628	0.0108		-5.67%	1.03%	0.64%	-6.31%	**-5.8201**	<-- =O5/J5
6	5-Dec-06	KR	0.28	0.30	0.02		234	0.0003	0.4483	0.0118		5.08%	0.40%	0.21%	4.87%	**4.1297**	
7	18-Apr-06	SVU	0.56	0.55	-0.01		73	-0.0008	0.7049	0.0123		-0.28%	1.69%	1.11%	-1.39%	-1.1313	
8	26-Jul-06	SVU	0.57	0.53	-0.04		142	-0.0003	0.5416	0.0122		-7.09%	-0.04%	-0.05%	-7.03%	**-5.7632**	
9	10-Oct-06	SVU	0.53	0.61	0.08		195	-0.0002	0.6014	0.0128		4.36%	0.20%	0.10%	4.26%	**3.3170**	
10	9-Jan-07	SVU	0.56	0.54	-0.02		256	0.0003	0.5238	0.0130		-1.70%	-0.05%	0.00%	-1.70%	-1.3061	
11	27-Apr-06	SWY	0.30	0.32	0.02		80	0.0001	1.0139	0.0130		2.88%	0.33%	0.34%	2.54%	**1.9473**	
12	20-Jul-06	SWY	0.36	0.42	0.06		138	0.0001	0.9289	0.0118		8.39%	-0.85%	-0.78%	9.17%	**7.7519**	
13	12-Oct-06	SWY	0.39	0.39	0.00		197	0.0003	0.7533	0.0134		-1.43%	0.95%	0.75%	-2.18%	-1.6265	
14	27-Feb-07	SWY	0.60	0.61	0.01		289	0.0012	0.7505	0.0131		-3.95%	-3.53%	-2.54%	-1.41%	-1.0814	
15	4-May-06	WFMI	0.35	0.36	0.01		85	0.0006	0.8345	0.0175		12.50%	0.32%	0.33%	12.17%	**6.9434**	
16	1-Aug-06	WFMI	0.34	0.35	0.01		146	-0.0007	1.2329	0.0164		-12.51%	-0.45%	-0.63%	-11.88%	**-7.2345**	
17	3-Nov-06	WFMI	0.29	0.29	0.00		213	-0.0014	1.3199	0.0195		-26.21%	-0.22%	-0.43%	-25.78%	**-13.2186**	
18	22-Feb-07	WFMI	0.40	0.38	-0.02		286	-0.0020	1.5321	0.0243		13.13%	-0.09%	-0.33%	13.46%	**5.5465**	
19																	
20														Positive surprise	2.65%	<-- =SUMIF($E$3:$E$18,">0",$O$3:$O$18)/COUNTIF(E3:E18,">0")	
21														Non-positive surprise	-3.24%	<-- =SUMIF($E$3:$E$18,"<=0",$O$3:$O$18)/COUNTIF(E3:E18,"<=0")	

Cell G18 contains formula: =COUNTIF('Stock Prices'!$A$3:$A$551,"<="&TEXT(A18,"0"))-252

Cell H18 contains formula: =INTERCEPT(OFFSET('Stock Prices'!$A$2,$G18,10,252,1),OFFSET('Stock Prices'!$A$2,$G18,2,252,1))

Cell I18 contains formula: =SLOPE(OFFSET('Stock Prices'!$A$2,$G18,10,252,1),OFFSET('Stock Prices'!$A$2,$G18,2,252,1))

Cell J18 contains formula: =STEYX(OFFSET('Stock Prices'!$A$2,$G18,10,252,1),OFFSET('Stock Prices'!$A$2,$G18,2,252,1))

For each announcement we have determined the intercept and slope of the market-model regression for the 252 days preceding the announcement.[11] Here's a specific example from the above spreadsheet:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
2	**Calendar Date**	**Ticker**	**Consensus**	**Actual**	**Surprise: =D3-C3**		**Starting Point**	**Intercept**	**Slope**	**STEXY**		**Actual Return**	**S&P 500 Return**	**Expected Return: =H3+I3*M3**	**Abnormal Return: =L3-N3**	**T-stat**
3	7-Mar-06	KR	0.36	0.39	0.03		44	0.0003	0.6662	0.0123		1.36%	-0.19%	-0.09%	1.46%	1.1808
18	22-Feb-07	WFMI	0.40	0.38	-0.02		286	-0.0020	1.5321	0.0243		13.13%	-0.09%	-0.33%	13.46%	**5.5465**

10. We included only Kroger, Supervalu, Safeway, and Whole Foods in the sample. This is obviously an incomplete sample, both in terms of the firms covered and the number of announcements. However, this extended example is meant to impart the flavor of a full-bodied event study.

11. The event window is defined in column G by the "Starting point," which uses **Countif** to locate a date 252 business days before the event date in the data base of stock returns. Notice that the "Starting point" is used in the **Intercept**, **Slope**, **Rsq** formulas in columns H, I, and J.

Row 3 tracks the market model of Kroger stock to the S&P 500 in the 252 trading days before the earnings announcement on 7 March 2006. The market model is $r_{Safeway} = 0.0001 + 0.9289{*}r_{S\&P500}$. Kroger's actual return on the announcement date, 1.36%, is 1.46% higher than the return which would have been predicted by its market model. However, dividing this 1.46% by the standard deviation of the abnormal returns (the regression residuals) (**Steyx** = 0.0123) gives a *t*-statistic of 1.1808, which is not significant at the 5% level.

Row 18 tracks the market model of Whole Foods stock to the S&P 500 in the year before the earnings announcement after market close on 21 February 2007. The market model is $r_{Safeway} = 0.0001 + 0.9289{*}r_{S\&P500}$, with **Steyx** = 0.0243. The abnormal return on the day of the announcement, 13.46%, is significant at the 1% level (hence the boldface in cell P18). To temper this interpretation, note that on the day of the earnings announcement, Whole Foods announced the merger with Wild Oats Markets; this makes it difficult to interpret the true market reaction to the earnings release. In general, we have to be careful interpreting abnormal returns when an event is announced with confounding information.

The Cumulative Abnormal Returns

We restate the data in slightly different form in the spreadsheet below—using the **Offset** function—and then compute the abnormal returns in an event window which goes from −10 to +10 days around the earnings announcements:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	**CUMULATIVE ABNORMAL RETURNS IN THE WINDOW -10 TO +10**																
2	Calendar Date	**7-Mar-06**	**20-Jun-06**	**12-Sep-06**	**5-Dec-06**	**18-Apr-06**	**26-Jul-06**	**10-Oct-06**	**9-Jan-07**	**27-Apr-06**	**20-Jul-06**	**12-Oct-06**	**27-Feb-07**	**4-May-06**	**1-Aug-06**	**3-Nov-06**	**22-Feb-07**
3	Ticker	KR	KR	KR	KR	SVU	SVU	SVU	SVU	SWY	SWY	SWY	SWY	WFMI	WFMI	WFMI	WFMI
4	Consensus	0.36	0.42	0.29	0.28	0.56	0.57	0.53	0.56	0.3	0.36	0.39	0.6	0.35	0.34	0.29	0.4
5	Actual	0.39	0.42	0.29	0.3	0.55	0.53	0.61	0.54	0.32	0.42	0.39	0.61	0.36	0.35	0.29	0.38
6	**Surprise**	0.03	0.00	0.00	0.02	-0.01	-0.04	0.08	-0.02	0.02	0.06	0.00	0.01	0.01	0.01	0.00	-0.02
7																	
8	Starting Point	44	117	175	234	73	142	195	256	80	138	197	289	85	146	213	286
9	Intercept	0.0003	0.0004	0.0006	0.0003	-0.0008	-0.0003	-0.0002	0.0003	0.0001	0.0001	0.0003	0.0012	0.0006	-0.0007	-0.0014	-0.0020
10	Slope	0.6662	0.6063	0.5628	0.4483	0.7049	0.5416	0.6014	0.5238	1.0139	0.9289	0.7533	0.7505	0.8345	1.2329	1.3199	1.5321
11	STEXY	0.0123	0.0113	0.0108	0.0118	0.0123	0.0122	0.0128	0.0130	0.0130	0.0118	0.0134	0.0131	0.0175	0.0164	0.0195	0.0243
12																	
13																	
14	**Day relative to event**	**ABNORMAL RETURN**															
15	-10	0.59%	-0.07%	0.72%	-0.46%	-0.68%	-1.81%	-1.04%	0.47%	0.32%	-0.01%	0.36%	0.46%	-0.52%	-3.87%	0.05%	0.26%
16	-9	-1.19%	0.89%	0.38%	-1.20%	-1.06%	0.33%	-0.43%	0.36%	0.65%	-0.96%	-0.94%	0.29%	-1.94%	0.93%	-0.06%	0.94%
17	-8	1.08%	0.58%	-0.32%	0.00%	0.25%	-0.35%	0.29%	-0.92%	-0.65%	0.21%	0.22%	0.60%	-1.20%	-0.33%	-0.98%	1.82%
18	-7	-0.67%	-1.34%	0.85%	-0.23%	-0.86%	1.30%	0.05%	0.66%	0.90%	-0.78%	-0.16%	-0.96%	-1.27%	-0.18%	0.53%	-0.87%
19	-6	0.98%	-0.35%	-0.59%	-0.67%	-0.55%	1.24%	-0.66%	-0.48%	-2.40%	0.20%	-3.61%	0.33%	-0.59%	-1.79%	0.10%	-0.11%
20	-5	-0.34%	1.41%	-0.84%	0.42%	0.29%	0.13%	2.44%	-1.05%	-0.66%	0.19%	-4.99%	0.68%	-1.35%	-0.97%	0.25%	0.35%
21	-4	-0.23%	0.11%	0.49%	0.62%	-0.71%	0.79%	-0.63%	0.71%	-0.57%	0.93%	0.88%	0.21%	0.09%	-1.88%	0.31%	0.35%
22	-3	-1.55%	-1.16%	-0.39%	-1.92%	-1.01%	-0.64%	-0.11%	-0.46%	-0.80%	0.92%	1.21%	-3.80%	0.41%	0.85%	-1.52%	1.22%
23	-2	0.93%	-0.59%	1.97%	0.98%	0.34%	0.49%	-0.05%	0.68%	0.61%	1.79%	2.36%	-1.12%	-0.28%	1.38%	0.03%	-1.01%
24	-1	0.03%	-0.26%	0.29%	2.63%	0.29%	-0.60%	2.95%	-0.79%	0.82%	-1.33%	0.30%	0.60%	1.08%	0.01%	-4.74%	-0.39%
25	0	1.46%	5.01%	-6.31%	4.87%	-1.39%	-7.03%	4.26%	-1.70%	2.54%	9.17%	-2.18%	-1.41%	12.17%	-11.88%	-25.78%	13.46%
26	1	1.43%	-0.63%	0.38%	-1.35%	-1.12%	-4.37%	0.58%	-1.57%	-0.12%	0.83%	-2.34%	1.01%	1.42%	-0.14%	-0.94%	-2.45%
27	2	-1.15%	-0.56%	0.93%	-0.15%	1.43%	-2.64%	0.99%	1.59%	1.33%	-1.44%	0.34%	0.37%	0.37%	-0.19%	2.02%	0.06%
28	3	1.08%	1.15%	-2.90%	-0.24%	-0.82%	0.87%	0.20%	-0.91%	-3.28%	-0.24%	0.35%	-1.44%	-1.78%	0.67%	3.11%	0.94%
29	4	0.03%	1.37%	0.27%	2.61%	0.36%	0.20%	-0.71%	-0.56%	-1.34%	-0.23%	1.26%	-0.06%	-0.40%	0.24%	0.14%	-1.13%
30	5	0.11%	-0.31%	-0.31%	1.19%	-0.99%	0.62%	-0.47%	-0.53%	0.58%	0.76%	-1.94%	-0.59%	0.04%	-4.00%	0.44%	-1.13%
31	6	-0.12%	2.70%	0.50%	-0.42%	-0.23%	-0.76%	0.83%	0.04%	1.02%	-1.13%	0.93%	-0.40%	-0.76%	-1.24%	0.84%	0.41%
32	7	-0.20%	0.35%	-0.96%	0.75%	0.63%	0.11%	0.27%	1.26%	-0.45%	-0.09%	0.13%	1.20%	1.25%	-0.02%	0.25%	1.97%
33	8	-0.18%	0.73%	1.90%	0.33%	0.46%	-0.41%	-0.02%	1.75%	-1.43%	0.26%	-2.28%	-0.12%	-0.46%	-1.25%	-0.34%	-0.63%
34	9	-1.34%	-0.52%	0.14%	-2.86%	0.23%	-1.15%	-0.23%	0.04%	-0.82%	1.04%	0.13%	-0.12%	-1.59%	7.14%	-0.99%	-1.50%
35	10	0.86%	0.08%	-0.14%	-1.19%	-0.88%	-1.27%	0.00%	0.48%	0.46%	-0.33%	0.92%	-0.12%	-0.85%	2.10%	-0.12%	-0.63%

We can compute the average abnormal return (AAR) and cumulative abnormal return (CAR) for each of the days in the event window. In the following table we perform this computation separately for positive versus non-positive earnings announcements:

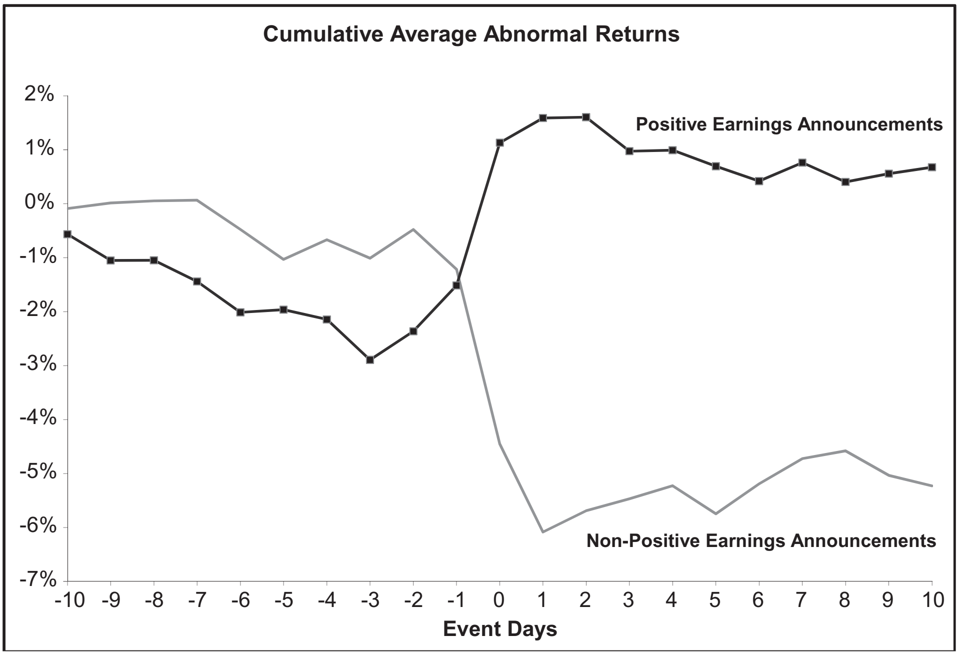
	S	T	U	V	W	X	Y	Z	AA	AB
3				Cell U11 contains formula						
4				{=SQRT(SUMPRODUCT(IF($B$6:$Q$6>0,$B$11:$Q$11),IF($						
5				B$6:$Q$6>0,$B$11:$Q$11))*(1/COUNTIF($B$6:$Q$6,">0")^2)						
6				)}						
7										
8										
9		**Unadjusted cross-sectional errors - Positive**				**Unadjusted cross-sectional errors - Non-Positive**				
10										
11			0.49%				0.54%			
12										
13		**Positive-Earnings Announcements**				**Non-Positive Earnings Announcements**				
14	**Day relative to event**	**AAR**	**T-stat**	**Cumulative Abnormal Returns**		**AAR**	**T-stat**	**Cumulative Abnormal Returns**		
15	-10	-0.57%	-1.1615	-0.57%		-0.09%	-0.1632	-0.09%		
16	-9	-0.48%	-0.9931	-1.05%		0.10%	0.1944	0.02%		
17	-8	0.00%	0.0026	-1.05%		0.04%	0.0670	0.05%		
18	-7	-0.39%	-0.8046	-1.44%		0.01%	0.0251	0.07%		
19	-6	-0.57%	-1.1777	-2.01%		-0.54%	-1.0058	-0.48%		
20	-5	0.05%	0.1043	-1.96%		-0.55%	-1.0278	-1.03%		
21	-4	-0.18%	-0.3753	-2.14%		0.37%	0.6774	-0.67%		
22	-3	-0.75%	-1.5410	-2.89%		-0.34%	-0.6376	-1.01%		
23	-2	0.53%	1.0900	-2.36%		0.54%	0.9917	-0.47%		
24	-1	0.85%	1.7437	-1.51%		-0.74%	-1.3685	-1.21%		
25	0	2.65%	**5.4406**	1.13%		-3.24%	**-6.0062**	-4.46%		
26	1	0.46%	0.9409	1.59%		-1.63%	**-3.0190**	-6.08%		
27	2	0.02%	0.0335	1.61%		0.40%	0.7356	-5.69%		
28	3	-0.63%	-1.2939	0.98%		0.22%	0.4121	-5.47%		
29	4	0.02%	0.0355	0.99%		0.24%	0.4434	-5.23%		
30	5	-0.30%	-0.6087	0.70%		-0.52%	-0.9620	-5.75%		
31	6	-0.28%	-0.5690	0.42%		0.55%	1.0269	-5.19%		
32	7	0.34%	0.6998	0.76%		0.47%	0.8643	-4.72%		
33	8	-0.36%	-0.7326	0.41%		0.14%	0.2680	-4.58%		
34	9	0.15%	0.3154	0.56%		-0.45%	-0.8382	-5.03%		
35	10	0.12%	0.2425	0.68%		-0.20%	-0.3614	-5.23%		
36										
37										
38	Cell T35 contains formula =SUMIF($B$6:$Q$6,">0",B35:Q35)/COUNTIF($B$6:$Q$6,">0")						Cell X35 contains formula =SUMIF($B$6:$Q$6,"<=0",B35:Q35)/COUNTIF($B$6:$Q$6,"<=0")			
39										
40		Cell U35 contains formula =T35/$U$11					Cell Y35 contains formula =X35/$Y$11			
41										
42										
43		Cell V35 contains formula =T35+V34					Cell Z35 contains formula =X35+Z34			

The test statistics for the positive and non-positive announcements have been computed by dividing the *average abnormal return* (AAR) for each day by the appropriate cross-sectional error for the specific type of return (cells U11 and Y11):

$$\text{cell U11: } \sqrt{\frac{\textit{Sumproduct of Steyx for positive announcements}}{(\textit{Number of positive announcements})^2}}$$

$$\text{cell Y11: } \sqrt{\frac{\textit{Sumproduct of Steyx for negative announcements}}{(\textit{Number of negative announcements})^2}}$$

Graphing the CARs gives:



On average, there appears to have been little leakage prior to the announcement date of neither the good news announcements nor the bad news. The market appears to have absorbed the information in the announcements rapidly—following the announcement date ("event day 0"), there appears to have been little additional response.

14.5 Using a Two-Factor Model of Returns for an Event Study

The model used in section 14.2 assumes an equilibrium model $r_{it} = \alpha_i + \beta_i r_{Mt}$. This so-called "one factor" model assumes that the returns on the stocks in question are driven only by one market index. In this section we illustrate a two-factor model. We assume that returns are a function of both a market and an industry factor: $r_{it} = \alpha_i + \beta_{i,Market} r_{Mt} + \beta_{i,Industry} r_{Industry,t}$. We then use this model to determine if a specific event influenced the returns and in which direction.

Our event: On 16 November 2006, Wendy's announced the purchase by tender of 22,418,000 shares at a price of $35.75 per share. This share purchase represented approximately 19% of the firm's equity. Wendy's stock closed on 16 November at $35.66.

WENDY'S

Wendy's Announces Final Results of its Modified "Dutch Auction" Tender Offer

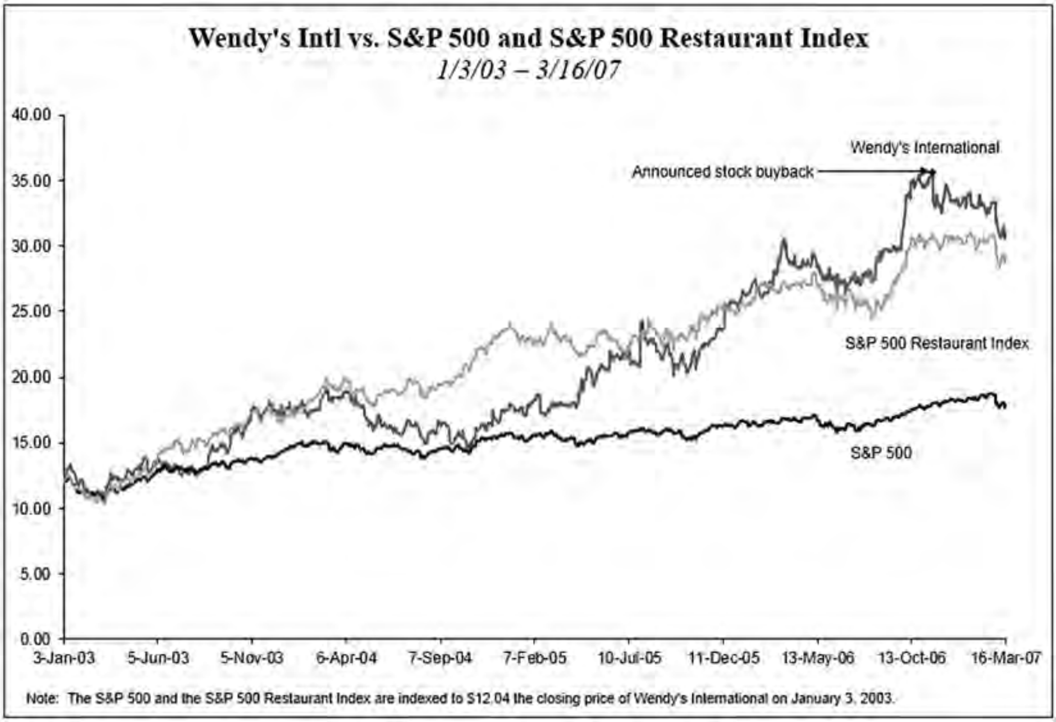
DUBLIN, Ohio (November 22, 2006) – Wendy's International, Inc. (NYSE:WEN) today announced the final results of its modified "Dutch Auction" tender offer, which expired at 5:00 p.m., Eastern Time, on November 16, 2006.

The Company has accepted for purchase 22,413,278 of its common shares at a purchase price of $35.75 per share, for a total cost of $801.3 million.

Shareholders who deposited common shares in the tender offer at or below the purchase price will have all of their tendered common shares purchased, subject to certain limited exceptions.

American Stock Transfer & Trust Company, the depositary for the tender offer, will promptly issue payment for the shares validly tendered and accepted for purchase under the tender offer.

The number of shares the Company accepted for purchase in the tender offer represents approximately 19% of its currently outstanding common shares.



Did the Repurchase Affect Wendy's Returns?

We start by regressing the daily returns on Wendy's on the S&P 500 and the S&P 500 Restaurant Index for the 252 days preceding the tender date of 16 November 2006. We use the array function **Linest** to do this computation.[12] The **Linest** box looks like:

12. The use of **Linest** to perform multiple regressions is discussed in Chapter 33. It is not the most user friendly of Excel functions.

	A	B	C	D
2		Industry	Market	Intercept
3	Slope -->	0.4157	0.5095	0.0012
4	Standard Error -->	0.0851	0.1410	0.0007
5	R^2 -->	0.3140	0.0103	#N/A
6	F statistic -->	56.9738	249	#N/A
7	SS_{xy} -->	0.0122	0.0266	#N/A

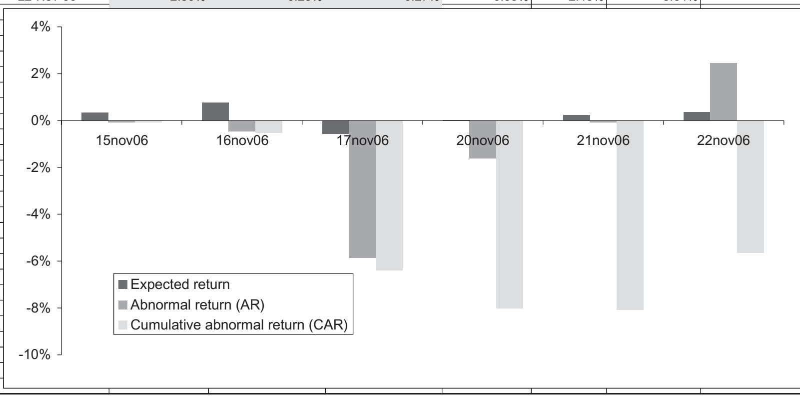
(Note that the #N/A above is produced by Excel and simply means that there is no entry for this column.)

From this box we can conclude that Wendy's return is sensitive to both the market and the industry.

$$r_{Wendys,t} = 0.0012 + \underbrace{0.5095}_{\substack{\uparrow \\ \text{Market reaction} \\ \text{coefficient.} \\ \text{Standard error:} \\ 0.1410}} * r_{Mt} + \underbrace{0.4157}_{\substack{\uparrow \\ \text{Industry reaction} \\ \text{coefficient.} \\ \text{Standard error:} \\ 0.0851}} * r_{Industry,t}$$

This **Linest** box is shown again (see following). Dividing the coefficients by their respective standard errors (row 9) shows that they are both significant at the 1% level. Note that cell C4 gives the standard error of the *y*-estimate; we use this in the analysis to determine the significance of the abnormal returns. A further analysis follows the spreadsheet.

	A	B	C	D	E	F	G	H	I	J
1	**WENDY'S RETURNS: ESTIMATION WINDOW AND EVENT WINDOW**									
2		Industry	Market	Intercept						
3	Slope -->	0.4157	0.5095	0.0012						
4	Standard Error -->	0.0851	0.1410	0.0007						
5	R^2 -->	0.3140	0.0103	#N/A		Cells B3:D7 contain the array formula {=LINEST(B15:B266,C15:D266,,TRUE)}				
6	F statistic -->	56.9738	249	#N/A						
7	SS_{xy} -->	0.0122	0.0266	#N/A						
8										
9	t-stat	4.8818	3.6142	1.8367	<-- =D3/D4	Cell C5 is the standard error of the y estimate, used in the t-test of the abnormal returns				
10										
11	Days in estimation window	252	<-- =COUNT(A15:A266)							
12										
13					**EVENT WINDOW**					
14	**Date**	**Wendy's Intl**	**S&P 500**	**S&P 500 Restaurant Index**	**Expected return**	**Abnormal return (AR)**	**Cumulative abnormal return (CAR)**		**T-test of AR**	
15	15-Nov-05	0.08%	-0.39%	-1.26%						
16	16-Nov-05	-0.37%	0.18%	-0.93%						
17	17-Nov-05	0.84%	0.94%	1.60%						
18	18-Nov-05	0.27%	0.44%	0.19%		Cell E267 contains formula =$D$3+$C$3*C267+$B$3*D267				
19	21-Nov-05	1.08%	0.53%	0.92%						
263	9-Nov-06	-0.89%	-0.53%	-0.35%			Cell F267 contains formula =B267-E267			
264	10-Nov-06	1.46%	0.19%	0.95%						
265	13-Nov-06	0.08%	0.25%	-0.05%						
266	14-Nov-06	0.73%	0.63%	0.03%						
267	15-Nov-06	0.25%	0.24%	0.21%	0.33%	-0.08%	-0.08%	<-- =F267	-0.0741	<-- =F267/$C$5
268	16-Nov-06	0.31%	0.23%	1.26%	0.76%	-0.45%	-0.53%	<-- =G267+F268	-0.4363	
269	17-Nov-06	-6.44%	0.10%	-1.78%	-0.57%	-5.87%	-6.40%	<-- =G268+F269	-5.6805	
270	20-Nov-06	-1.59%	-0.05%	-0.19%	0.02%	-1.61%	-8.01%		-1.5582	
271	21-Nov-06	0.15%	0.16%	0.07%	0.23%	-0.08%	-8.09%		-0.0784	
272	22-Nov-06	2.80%	0.23%	0.27%	0.35%	2.45%	-5.64%			



In rows 267–272 of the spreadsheet we use the two-factor model to analyze the abnormal returns (AR) and the cumulative abnormal returns (CAR) of the Wendy's announcement. While there is little AR or CAR on the days before the announcement, it is clear that the announcement on 16 November had a considerable impact on Wendy's returns on the day following (−5.87% abnormal return on 17 November) and on the next day (−1.61% AR on 20 November). Dividing the abnormal return by the standard error in C5 shows that only the AR on the event day is significant at the 5% level.

Furthermore, an analysis of the announcement broken down into the market and the industry factors shows that on both of the 2 days after the 16 November announcement date the effects of the market index on Wendy's returns were slight. On 17 November, however, there was a significant impact of the S&P 500 Restaurant Index on Wendy's which was lacking on 20 November.

To see this, we first discuss the day after the event, 17 November 2006:

	A	B	C	D	E	F	G	H
13					EVENT WINDOW			
14	Date	Wendy's Intl	S&P 500	S&P 500 Restaurant Index	Expected return	Abnormal return (AR)	Cumulative abnormal return (CAR)	
266	14-Nov-06	0.73%	0.63%	0.03%				
267	15-Nov-06	0.25%	0.24%	0.21%	0.33%	-0.08%	-0.08%	<-- =F267
268	16-Nov-06	0.31%	0.23%	1.26%	0.76%	-0.45%	-0.53%	<-- =G267+F268
269	17-Nov-06	-6.44%	0.10%	-1.78%	-0.57%	-5.87%	-6.40%	<-- =G268+F269
270	20-Nov-06	-1.59%	-0.05%	-0.19%	0.02%	-1.61%	-8.01%	
271	21-Nov-06	0.15%	0.16%	0.07%	0.23%	-0.08%	-8.09%	
272	22-Nov-06	2.80%	0.23%	0.27%	0.35%	2.45%	-5.64%	

On 17 November, the S&P 500 rose by 0.10% and the S&P 500 Restaurant Index fell by 1.78%. Given the regression $r_{Wendys,t} = 0.0012 + 0.5095{*}r_{Mt} + 0.4157{*}r_{Industry,t}$, the change in the S&P 500 would have affected Wendy's returns by approximately +0.05% (=0.5095*0.10%) and the change in the industry index would have affected Wendy's returns by approximately –0.74% (= 0.4157*–1.78%). But Wendy's decreased by –6.44% on the same day, well in excess of the impact of either of the two factors.

Here are the data for 20 November:

	A	B	C	D	E	F	G	H
13					EVENT WINDOW			
14	Date	Wendy's Intl	S&P 500	S&P 500 Restaurant Index	Expected return	Abnormal return (AR)	Cumulative abnormal return (CAR)	
266	14-Nov-06	0.73%	0.63%	0.03%				
267	15-Nov-06	0.25%	0.24%	0.21%	0.33%	-0.08%	-0.08%	<-- =F267
268	16-Nov-06	0.31%	0.23%	1.26%	0.76%	-0.45%	-0.53%	<-- =G267+F268
269	17-Nov-06	-6.44%	0.10%	-1.78%	-0.57%	-5.87%	-6.40%	<-- =G268+F269
270	20-Nov-06	-1.59%	-0.05%	-0.19%	0.02%	-1.61%	-8.01%	
271	21-Nov-06	0.15%	0.16%	0.07%	0.23%	-0.08%	-8.09%	
272	22-Nov-06	2.80%	0.23%	0.27%	0.35%	2.45%	-5.64%	

On 20 November, the S&P 500 fell by 0.05% and the S&P 500 Restaurant Index fell by 0.19%. Given the regression $r_{Wendys,t} = 0.0012 + 0.5095{*}r_{Mt} + 0.4157{*}r_{Industry,t}$, the change in the S&P 500 would have affected Wendy's returns by approximately –0.08% and the change in the Restaurant Industry

index would have affected Wendy's returns by approximately –0.03%. But Wendy's decreased by −1.59% on the same day, which is again well in excess of the impact of either of the two factors.

The impact of the announcement was felt even in the third day after the event, but we leave this analysis to the reader.

14.6 Using Excel's Offset Function to Locate a Regression in a Data Set

The analysis in section 14.2 requires us to do a regression of a specific stock's returns on the returns of the S&P 500, where the starting point of the regression is the 252 trading days before a specific date. The technique in section 14.2 uses a number of Excel functions:

• The functions **Intercept**, **Slope**, **Rsq** give the regression intercept, slope, and *r*-squared. These functions have been illustrated in Chapter 2 and in the previous portfolio chapters. The function **Steyx** gives the standard deviation of the regression residuals.

• The function **CountIf** counts the number of cells in a range which meet a specific condition. **CountIf** has the syntax **CountIf(data,condition)**. *However*, the **condition** must be a text condition (which means that in this example we will use the Excel function **Text** to translate a date to a text number—more later).

• The function **Offset** (see also Chapter 33) allows us to specify a cell or a block of cells in an array.

To illustrate the problem, consider the following data of returns for General Mills (GIS) and the S&P 500. We want to run a regression of the GIS returns on the S&P 500 returns for 10 dates before 29 January 1997:

	A	B	C	D	E	F	G
1	**USING OFFSET, COUNTIF, AND TEXT TO LOCATE A REGRESSION IN A DATA SET**						
2	**Date**	**General Mills GIS**	**Return**		**SP500**	**Return**	
3	3-Jan-97	57.96			748.03		
4	6-Jan-97	58.19	0.0040	<-- =LN(B4/B3)	747.65	-0.0005	<-- =LN(E4/E3)
5	7-Jan-97	59.33	0.0194		753.23	0.0074	
6	8-Jan-97	59.33	0.0000		748.41	-0.0064	
7	9-Jan-97	59.91	0.0097		754.85	0.0086	
8	10-Jan-97	59.91	0.0000		759.5	0.0061	
9	13-Jan-97	59.68	-0.0038		759.51	0.0000	
10	14-Jan-97	59.91	0.0038		768.86	0.0122	
11	15-Jan-97	59.56	-0.0059		767.2	-0.0022	
12	16-Jan-97	59.56	0.0000		769.75	0.0033	
13	17-Jan-97	59.56	0.0000		776.17	0.0083	
14	20-Jan-97	59.44	-0.0020		776.7	0.0007	
15	21-Jan-97	60.71	0.0211		782.72	0.0077	
16	22-Jan-97	61.4	0.0113		786.23	0.0045	
17	23-Jan-97	62.09	0.0112		777.56	-0.0111	
18	24-Jan-97	61.63	-0.0074		770.52	-0.0091	
19	27-Jan-97	61.29	-0.0055		765.02	-0.0072	
20	28-Jan-97	61.06	-0.0038		765.02	0.0000	
21	29-Jan-97	62.09	0.0167		772.5	0.0097	
22	30-Jan-97	62.21	0.0019		784.17	0.0150	
23	31-Jan-97	62.44	0.0037		786.16	0.0025	
24	3-Feb-97	62.09	-0.0056		786.73	0.0007	
25							
26	Starting date	29-Jan-97					
27	Rows from top of data to starting date	19	<-- =COUNTIF(A3:A24,"<="&TEXT(B26,"0"))				
28	**Regression**						
29	Intercept	0.0022	<-- =INTERCEPT(OFFSET(A3:F24,B27-11,2,10,1),OFFSET(A3:F24,B27-11,5,10,1))				
30	Slope	0.5198	<-- =SLOPE(OFFSET(A3:F24,B27-11,2,10,1),OFFSET(A3:F24,B27-11,5,10,1))				
31	R-squared	0.1413	<-- =RSQ(OFFSET(A3:F24,B27-11,2,10,1),OFFSET(A3:F24,B27-11,5,10,1))				
32							
33							
34	**Check**						
35	Intercept	0.0022	<-- =INTERCEPT(C11:C20,F11:F20)				
36	Slope	0.5198	<-- =SLOPE(C11:C20,F11:F20)				
37	R-squared	0.1413	<-- =RSQ(C11:C20,F11:F20)				

To run this regression, we first use **CountIf(data,condition)** to count the row number of the data on which the starting date falls. Since **condition** must be a text entry, we translate the date in cell B26 to a text by using **Text(b26,″0″)**. The Excel function **=CountIf(A3:A24,″<=″&Text(B26,″0″))** now counts the number of cells in the column A3:A24 which are less than or equal to the date in cell B26. The answer, as you can see in cell B27, is 19.

Next, we use **Offset(A3:F24,B27-11,2,10,1)** to locate the 10 rows of GIS returns before the 19th row indicated by the starting date. This is a tricky function!

$$\text{Offset}(\underbrace{A3:F24}_{\text{Reference array}},\underbrace{\underbrace{B27-11}_{\text{Starting row in reference array}},\ \underbrace{2}_{\text{2 columns over from top corner of reference array}}}_{\text{Fix starting corner in reference array from which data is considered}},\ \underbrace{\underbrace{10}_{\text{10 rows \textit{below} starting corner}},\ \underbrace{1}_{\text{Only 1 column of data}}}_{\text{Data considered}})$$

The functions **Intercept**, **Slope**, **Rsq** can now be used with **Offset (A3:F24,B27-11,2,10,1)** and **Offset(A3:F24,B27-11,5,10,1)**:

$$=\text{Intercept}(\underbrace{\text{Offset}(A3:F24, B27-11, 2, 10, 1)}_{\text{y-data}},$$

$$\underbrace{\text{Offset}(A3:F24, B27-11, 5, 10, 1)}_{\text{x-data}})$$

14.7 Summary

Event studies, used to determine the impact of a particular market effect on a specific stock or a generic market effect on a set of stocks, are some of the most widely used technologies in practical finance. While Excel may not be the optimal tool for performing an event study, we have used it in this chapter to illustrate both uses of the event study. We have shown that Excel can readily be used to perform either a one-factor or a two-factor event study. The Excel techniques employed are easily acquired by a sophisticated user.

III VALUATION OF OPTIONS

Chapters 15–19 deal with option pricing and applications. Chapter 15 is an introduction to options. After defining the option terminology, Chapter 15 discusses option payoffs and basic option arbitrage propositions. In Chapter 16 we discuss the binomial option pricing model and its implementation in Excel. After showing how these binomial models work, we use Visual Basic for Applications (VBA) to build binomial option pricing functions for both European and American options. One of the applications discussed is the pricing of employee stock options.

Chapter 17 discusses the Black-Scholes pricing formulas for European calls and puts. These formulas can be implemented either by direct calculation in the spreadsheet or by using VBA to build new spreadsheet functions. An extension of the Black-Scholes model to the pricing of dividend paying stocks (the so-called Merton model) is implemented. We show how to apply the option pricing models to the valuation of structured securities. Chapter 18 discusses the computation of "Greeks"—the derivatives of the option-pricing formula which show the sensitivities of the option valuation to its various parameters.

Chapter 19 discusses real options—the application of the option pricing models to real investments.

Once you have mastered the ideas in these chapters, we refer you to the Monte Carlo section of this book. Chapters 24–30 show how option pricing strategies can be simulated in Excel. These chapters also show how more complicated options—whose payments are path dependent—can be priced using Monte Carlo methods.

15 Introduction to Options

15.1 Overview

In this chapter we give a brief introduction to options. The chapter can, at best, serve as an introduction to the already informed. If you know nothing whatsoever about options, read an introduction to the topic in a basic finance text.[1] We start with the basic definitions and options terminology, go on to discuss graphs of option payoffs and "profit diagrams," and finally discuss some of the more important option arbitrage propositions (sometimes referred to as linear pricing restrictions). In subsequent chapters we discuss two methods of pricing options: The binomial option pricing model (Chapter 16) and the Black-Scholes option pricing model (Chapter 17).

15.2 Basic Option Definitions and Terminology

An *option on a stock* is a security that gives the holder the right to buy or to sell one share of the stock on or before a particular date for a predetermined price. Here is a brief glossary of terms and notation used in the field of options:

- *Call, C*: An option that gives the holder the right to buy a share of stock on or before a given date at a predetermined price.
- *Put, P*: An option that gives the holder the right to sell a share of stock on or before a given date at a predetermined price.
- *Exercise price, X*: The price at which the holder can buy or sell the underlying stock; sometimes also referred to as the *strike price*.
- *Expiration date, T*: The date on or before which the holder can buy or sell the underlying stock.
- *Stock price, S_t*: The price at which the underlying stock is selling at date t. The current stock price is denoted S_0.
- *Option price*: The price at which the option is sold or bought.

1. Good chapters can be found in the following books: John Hull, *Options, Futures and Other Derivatives* (Prentice Hall, 8th edition, 2011); Zvi Bodie, Alex Kane, and Alan J. Marcus, *Investments* (McGraw-Hill, 9th edition, 2011).

American versus European options: In the jargon of options markets, an American option is an option which can be exercised on or before the expiration date T, whereas a European option is one which can be exercised only on the expiration date T. This terminology is confusing for two reasons:

• The options sold on both European and American options exchanges are almost invariably American options.

• The simplest option pricing formulas (these include the famous Black-Scholes option pricing formula discussed in Chapter 17) are for *European options*. As we show in section 15.6, in many cases we can price American options as if they were European options.

We use C_t to denote the price of a European call on date t, and P_t to denote the European put price. If it is clear that the option price refers to today's price, we often drop the subscript, writing C or P instead of C_0 or P_0. When we need fuller notation, we write $C_t(S_t,X,T)$ for the price of a call on date t when the price of the underlying stock is S_t, the exercise price is X, and the expiration date is T. If we wish to specify that our option pricing formula relates to an American option, we use the superscript A: C_t^A, $C_t^A(S_t, X, T)$ or $P_t^A(S_t, X, T)$. When written without superscripts, the options pricing refers to European options.

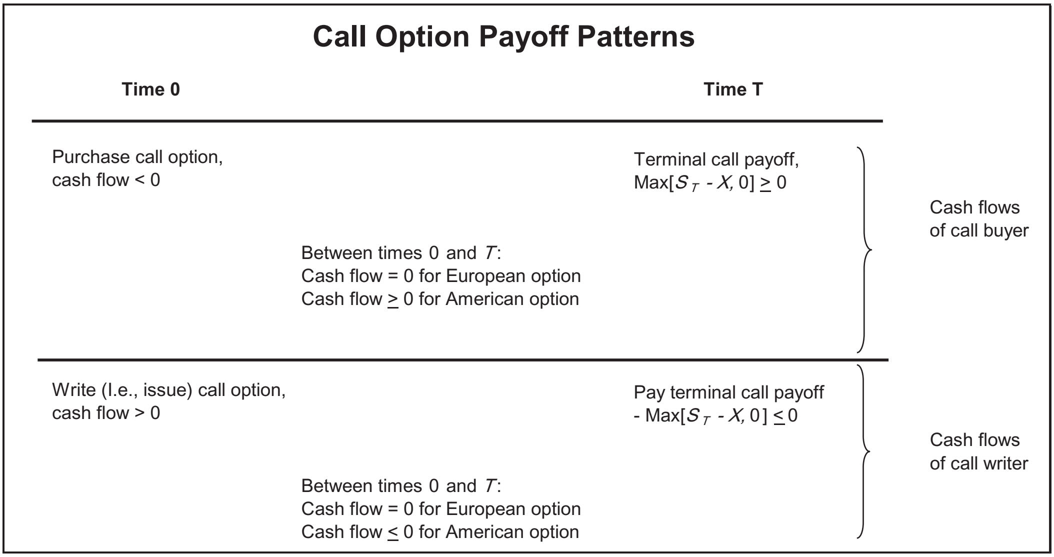
At-the-money, *in-the-money*, *out-of-the-money*: If the exercise price X of a call or a put is equal to the current price of the stock S_0, then the option is *at-the-money*. If a positive cash flow could be made by immediately exercising an American option (that is, $S_0 - X > 0$ for a call and $X - S_0 > 0$ for a put), then the option is *in-the-money*.[2]

Writing Options Versus Purchasing Options: Cash Flows

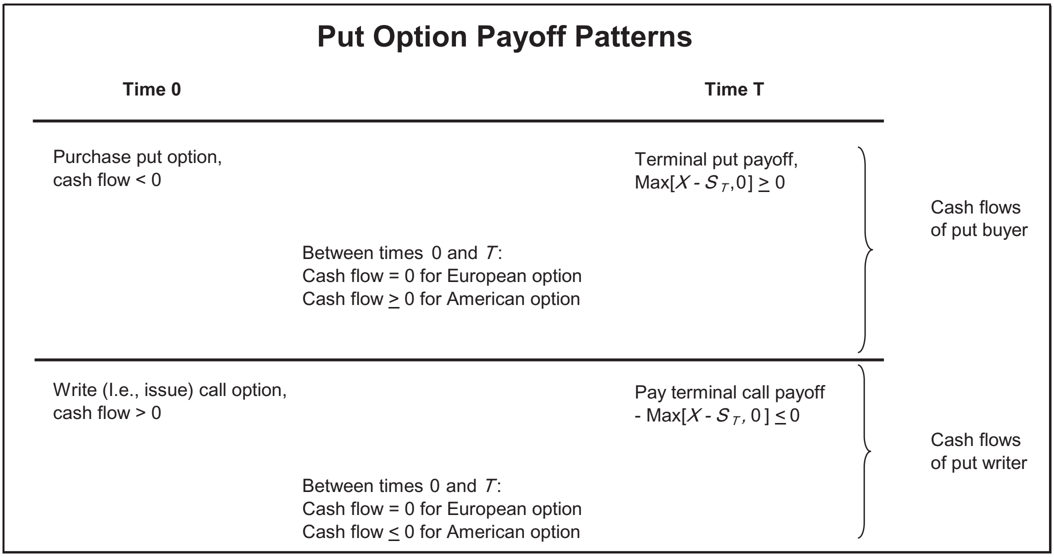
The purchaser of a call option acquires the right to buy a share of stock for a given price on or before date T and pays for this right at the time of purchase. The *writer* or seller of this call option is the seller of this right: The writer collects the option price today in return for obligating herself to deliver one share of stock in the future for the exercise price, if the purchaser of the call demands. In terms of cash flows, the purchaser of an option always has an

2. It is of course not logical, that you can ever make an immediate profit by buying an American option and immediately exercising it. Thus, for American calls, $C_0 > S_0 - X$, and for American puts, $P_0 > X - S_0$. *In-the-money* and *out-of-the-money* refers only to the relation between S_0 and X without taking into account the option price.

initial negative cash flow (the price of the option) and a future cash flow which is at worst zero (if it is not worthwhile exercising the option) and otherwise positive (if the option is exercised). The cash-flow position of the writer of the option is reversed: An initial positive cash flow is followed by a terminal cash flow which is at best zero.



A similar payoff pattern holds for the cash flows of the purchaser and writer of a put option on a stock:



15.3 Some Examples

Below we show the most actively traded options on 22 October 2012 (http://biz.yahoo.com/opt/stat1.html). Each option represents a trade on 100 units of the underlying (so that 36,371 Microsoft 29 November calls represent call options on 3.6 million shares of MSFT with an exercise price of $29). The "open interest" is the number of outstanding contracts at the end of the day. For MSFT, 36,371 calls were traded on 22 October, and at the end of the day 206,064 calls were outstanding.

MOST ACTIVE OPTIONS, 22 OCTOBER 2012										
						Option price				
Rank	Stock	Option exercise price	Call or put?	Option expiration	Closing stock price	Closing	Change from previous day	Volume	Open Interest	Short description of underlying
1	XLF	17	Call	17-Nov-12	16.11	0.06	0.01	433,998	2,358,344	Tracks index of financial stocks
2	SPY	143	Put	17-Nov-12	143.41	1.94	-0.23	89,617	1,803,974	Tracks SP500
3	QQQ	65	Put	17-Nov-12	66.02	0.81	-0.25	52,780	759,554	Tracks Nasdaq 100
4	IWM	78	Put	17-Nov-12	81.83	0.54	-0.09	43,168	448,734	Tracks Russell 2000
5	MSFT	29	Call	17-Nov-12	28	0.23	-0.20	36,371	206,064	Microsoft
6	HPQ	17	Put	18-May-13	14.71	3.23	-0.18	26,425	6,984	Hewlett-Packard
7	SLV	34	Call	17-Nov-12	31.39	0.15	0.01	26,250	121,974	Tracks silver price
8	UTX	72.5	Put	17-Nov-12	77.83	0.38	-0.01	24,515	22,674	United Technologies
9	FB	21	Call	17-Nov-12	19.32	0.78	0.10	24,390	183,944	Facebook
10	INTC	22	Call	17-Nov-12	21.46	0.20	0.01	23,339	190,434	Intel
11	ECA	26	Call	17-Nov-12	23.02	0.18	-0.23	20,252	219,634	Encana Corp.
12	GE	22	Call	17-Nov-12	21.7	0.31	-0.15	19,603	227,984	General Electric
13	BTU	16	Call	19-Jan-13	29.95	13.00	3.00	19,091	218,994	Peabody Energy
14	NLY	8	Put	17-Jan-15	15.94	0.66	0.05	16,771	177,934	Annaly Capital Management (a REIT)
15	CSCO	19	Call	17-Nov-12	18.19	0.36	0.00	16,654	569,894	Cisco
16	AET	45	Call	17-Nov-12	44.2	1.26	0.35	16,486	22,004	Aetna
17	EEM	41	Put	22-Dec-12	41.9	1.14	-0.22	15,454	444,464	Tracks MSCI emerging markets index
18	MS	19	Call	17-Nov-12	17.45	0.14	-0.05	14,654	289,574	Morgan Stanley
19	NXY	20	Put	22-Dec-12	24.14	0.77	0.27	14,645	71,564	Nexen (energy)
20	FXI	37	Call	17-Nov-12	37.67	1.16	0.25	13,948	1,127,324	Tracks China 25 index

A few facts stand out about this list:

- Seven of the options are on indices. The options enable investors/speculators to make bets on broad-based market movements.
- The list is approximately evenly divided between puts (8 out of 20) and calls. This is not always true—when investors are optimistic about future market movements, calls tend to dominate the most active, and vice versa.
- The options in the most-active list tend to be short-term options. Although longer-term options exist, these are traded less than the near-term.

15.4 Option Payoff and Profit Patterns

One of the attractions of options is that they allow their owners to change the payoff patterns of the underlying assets. In this section we consider:

- The basic payoff and profit patterns of a call and a put option and a share.
- The payoff patterns of various combinations of options and shares.

Stock Profit Patterns

We start with the **payoff pattern from a purchased stock**. Suppose you buy a share of General Pills stock in July at its then-current market price of \$40. If in September the price of the stock is \$70, you will have made a \$30 profit; if its price is \$30, you will have a loss (or a negative profit) of \$10.[3] Generalize this by writing the price of the stock in September as S_T and its price in July by S_0. Then we write the profit function from the stock as:

$$Profit\ from\ stock = S_T - S_0$$

Payoff from the Short Sale of a Stock

Suppose we had sold one share of GP stock short in July, when its market price was \$40. If in September the market price of GP was \$70, and if at that point we undid the short sale (i.e., we purchase a share at the market price in order to return the share to the lender of the original short), then our profit would be −\$30:

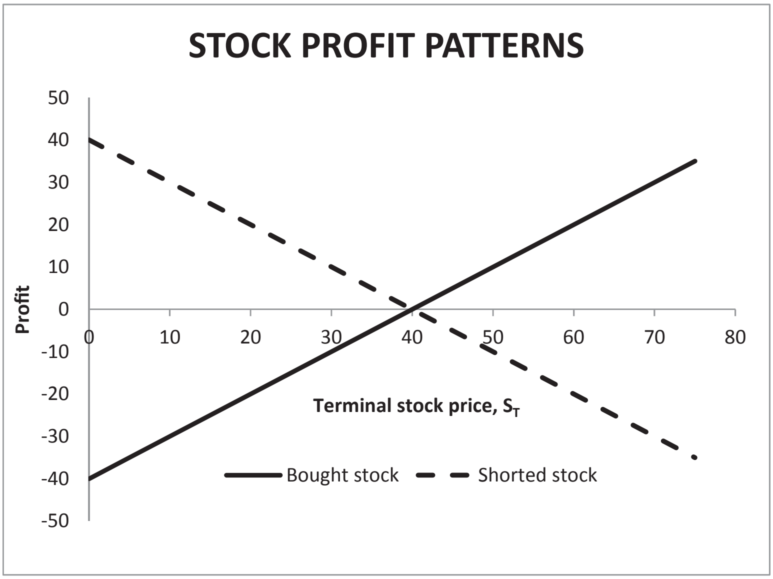
$$\begin{aligned} Profit\ from\ short\ sale\ of\ stock &= S_0 - S_T \\ &= -(profit\ from\ purchase\ of\ stock) \end{aligned}$$

Notice that the profit from the short sale is the *negative* of the profit from the purchase; this is always the case (also for options, considered below).

Graphing Stock Profit Patterns

The Excel graph below graphs the profit patterns from both a purchase and a short sale of the GP stock described above.

3. Our use of the word *profit* in this section constitutes a slight abuse of language and the standard finance concept of the word, since we are ignoring the interest costs associated with buying the asset. In the case at hand, this abuse of language is both traditional and harmless.



Call Option Profit Patterns

As in the case of a stock, we start with the **payoff pattern from a purchased call**. We go back to the General Pills (GP) options of the previous section. Suppose that in July you bought one GP September 40 call for \$4.[4] In September you will exercise the call only if the market price of GP is higher than \$40. If we write the initial (July) call price as C_0, we can write the profit function from the call in September as follows:

$$\begin{aligned} \textit{Call profit in September} &= \max(S_T - X, 0) - C_0 \\ &= \max(S_T - 40, 0) - 4 \\ &= \begin{cases} -4 & \text{if } S_T \le 40 \\ S_T - 44 & \text{if } S_T > 40 \end{cases} \end{aligned}$$

4. Because the exercise price of this call is equal to the current market price of the stock, it is called an *at-the-money* call. When the exercise price of the call is higher than the current market price, it is called an *out-of-the-money* call, and when the exercise price is lower than the current market price, the call is an *in-the-money* call.

Payoff Pattern from a Written Call

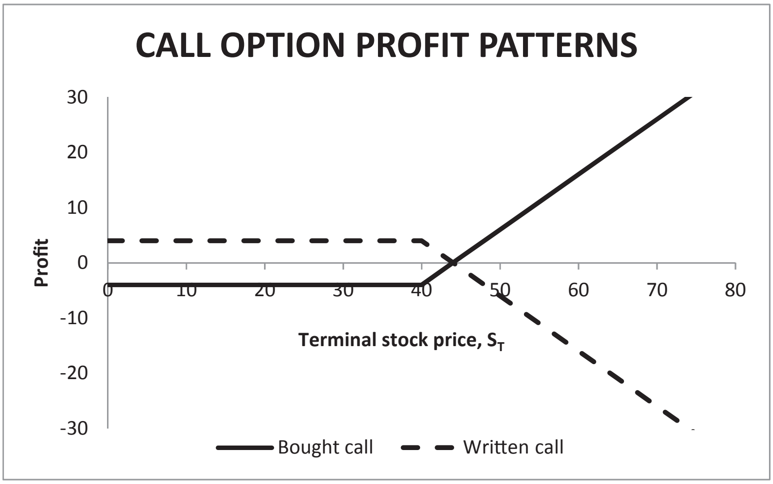
In options markets the purchaser of a call buys the call from a counterparty who issues the call. In the jargon of options, the issuer of the call is called the *call writer*. It is worthwhile to spend a few minutes considering the difference between the security bought by the call purchaser and the call writer:

- The call purchaser buys a security which *gives the right to buy a share of stock on or before date T for price X*. The cost of this privilege is the *call price* C_0, which is paid at the time of the call purchase. Thus the call purchaser has an initial negative cash flow (the purchase price C_0); on the other hand, his cash flow at date T is always non-negative: $\max(S_T - X, 0)$.
- The call writer gets C_0 at the date of the call purchase. In return for this price, the writer of the call *agrees to sell a share of the stock for price X on or before date T*. Notice that whereas the call purchaser has an option, the call writer has undertaken an obligation. Note that the cash flow pattern of the call writer is opposite to that of the call purchaser: The writer's initial cash flow is positive ($+C_0$), and her cash flow at date T is always non-positive: $-\max(S_T - X, 0)$.

The profit of a call writer is the opposite of that of the call purchaser. For the case of the GP options:

$$\begin{aligned} \textit{Call writer's profit in September} &= C_0 - \max(S_T - X, 0) \\ &= 4 - \max(S_T - 40, 0) \\ &= \begin{cases} +4 & \text{if } S_T \le 40 \\ 44 - S_T & \text{if } S_T > 40 \end{cases} \end{aligned}$$

Graphing the profit patterns of the bought and the written call gives:



Put Option Profit Patterns

Payoff Pattern from a Purchased Put

If in July you bought one GP September 40 put for \$2, then in September you will exercise the put only if the market price of GP is lower than \$40. If we write the initial (July) put price as P_0, we can write the profit function from the put in September as follows:

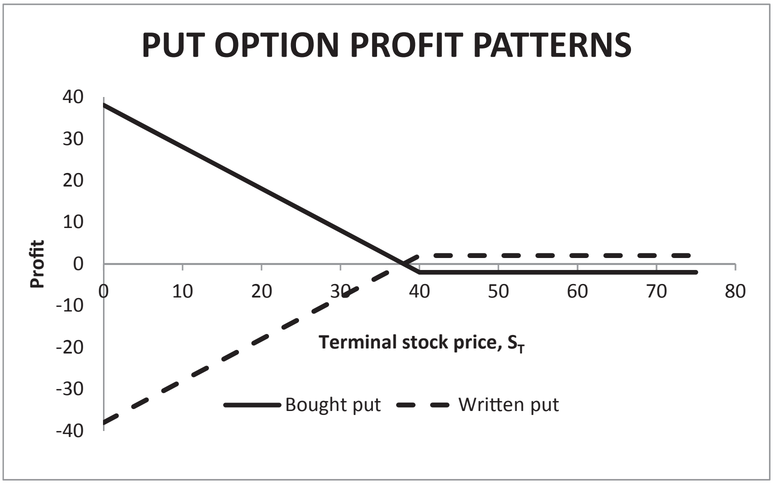
$$
\begin{aligned}
\textit{Put profit in September} &= \max(X - S_T, 0) - P_0 \\
&= \max(40 - S_T, 0) - 2 \\
&= \begin{cases} 38 - S_T & \text{if } S_T \le 40 \\ -2 & \text{if } S_T > 40 \end{cases}
\end{aligned}
$$

Payoff Pattern from a Written Put

The *put writer* obligates herself to purchase one share of GP stock on or before date T for the put exercise price of X. For putting herself in this invidious position, the writer of the put receives, at the time the put is written, the put price P_0. The payoff pattern from writing the GP September 40 put is therefore:

$$\begin{aligned} \textit{Put writer's profit in September} &= P_0 - \max(X - S_T, 0) \\ &= 2 - \max(40 - S_T, 0) \\ &= \begin{cases} -38 + S_T & \text{if } S_T \le 40 \\ 2 & \text{if } S_T > 40 \end{cases} \end{aligned}$$

Graphing the profit patterns of the bought and the written put gives:



15.5 Option Strategies: Payoffs from Portfolios of Options and Stocks

There is some interest in graphing the combined profit pattern from a portfolio of options and stocks. These patterns give an indication of how options can be used to *change the payoff patterns* of "standard" securities such as stocks and bonds. Here are a few examples.

The Protective Put

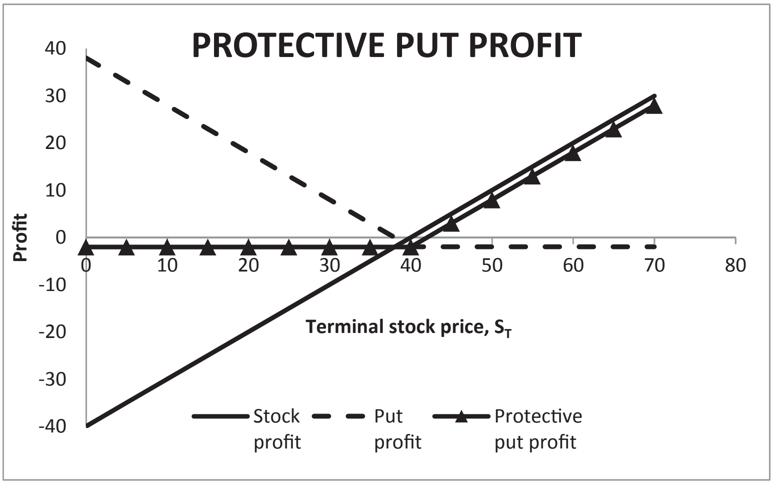
Consider the following combination:

- One share of stock, purchased for S_0
- One put, purchased for P with exercise price X

This option strategy is often called a "protective put" strategy or "portfolio insurance"; in Chapter 29 we return to this topic, exploring it in much further detail. The payoff pattern of the protective put is given by:

$$\begin{aligned} \textit{Stock profit} + \textit{Put profit} &= S_T - S_0 + \max(X - S_T, 0) - P_0 \\ &= \begin{cases} S_T - S_0 + X - S_T - P_0 & \text{if } S_T \le X \\ S_T - S_0 - P_0 & \text{if } S_T > X \end{cases} \\ &= \begin{cases} X - S_0 - P_0 & \text{if } S_T \le X \\ S_T - S_0 - P_0 & \text{if } S_T > X \end{cases} \end{aligned}$$

When applied to the GP example (that is, buying a share at \$40 and a put with X = \$40 for \$2) this gives the following graph:



This pattern looks very much like the payoff pattern from a call.[5]

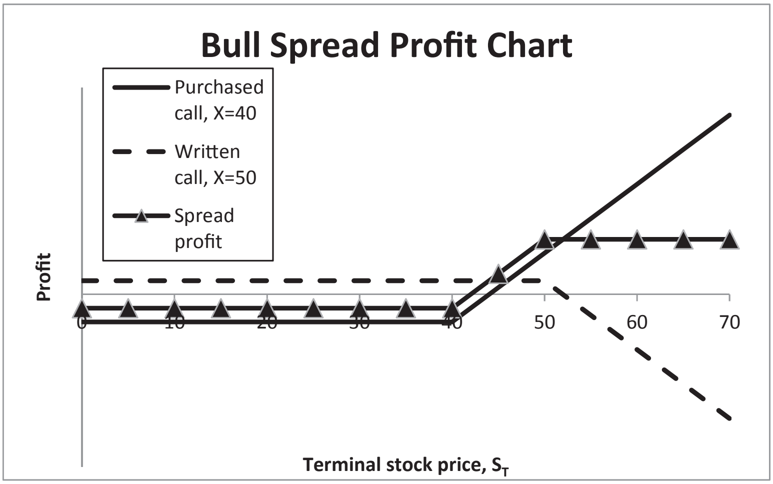
5. In section 15.5 we prove and illustrate the *put-call parity theorem*. It follows from this theorem that a call must be priced at a price C such that $C = P + S_0 - Xe^{-rT}$. Thus, when calls are correctly priced according to this theorem, the payoff from a put + stock combination is the same as that from a call + bond combination. We prove this theorem and give an illustration in the next section.

Spreads

Another combination involves buying and writing calls with different exercise prices. When the bought call has a low exercise price and the written call has a higher exercise price, the combination is called a *bull spread*. As an example, suppose you bought a call (for \$4) with an exercise price of \$40 and wrote a call (for \$2) with an exercise price of \$50. This bull spread gives a profit of

$$\max(S_T-40,0)-4-(\max(S_T-50,0)-2)$$
$$=\begin{cases}-4+2 & \text{if } S_T\le 40\\ S_T-40-4+2=S_T-42 & \text{if } 40\le S_T\le 50\\ S_T-40-4-(S_T-50-2)=8 & \text{if } S_T\ge 50\end{cases}$$

The Excel graph given below shows each of the two calls and the resulting spread profit:



15.6 Option Arbitrage Propositions

In succeeding chapters we price options given specific assumptions about the probability distribution of the underlying asset (usually the stock) on which the option is written. However, there is much that can be learned about the pricing of options without making these specific probability assumptions. In this section we consider a number of these *arbitrage restrictions* on option pricing. Our list is by no means exhaustive, and we have concentrated on those propositions which provide insight into the pricing of options or which will be used in later sections.

Throughout we assume that there is a single risk-free interest rate which prices bonds; we also assume that this risk-free rate is continuously compounded, so that the present value of a riskless security which pays off X at time T is given by $e^{-rT}X$.

PROPOSITION 1 Consider a call option written on a stock which pays no dividends before the option's expiration date T. Then the lower bound on a call option price is given by:

$$C_0 \geq Max(S_0 - Xe^{-rT}, 0)$$

Comment Before proving this proposition, it will be helpful to consider its meaning: Suppose that the riskless interest rate is 10%, and suppose we have an American call option with maturity $T = 1/2$ (i.e., the expiration date of the option is one-half year from today) with $X = 80$ written on a stock whose current stock price $S_0 = 83$. A naive approach to determining a lower bound on this option's price would be to state that it is worth at least \$3, since it could be exercised immediately with a profit of \$3. Proposition 1 shows that the option's value is *at least* $83 - e^{-0.10*0.5}\ 80 = 6.90$. Furthermore, a careful examination of the proof below will show that this fact *does not* depend on the option being an American option—it is also true for a European option.

	A	B	C
1	**PROPOSITION 1—HIGHER LOWER BOUNDS FOR CALL PRICES**		
2	Current stock price, S_0	83	
3	Option time to maturity, T	0.5	
4	Option exercise price, X	80	
5	Interest rate, r	10%	
6			
7	Naive minimum option price, Max(S_0-X,0)	3	<-- =MAX(B2-B4,0)
8	Proposition 1 lower bound on option price, Max(S_0 - Exp(-rT)X,0)	6.902	<-- =MAX(B2-EXP(-B5*B3)*B4,0)

Proof of Proposition 1 Standard arbitrage proofs are built on the consideration of the cash flows from a particular strategy. In this case the strategy is the following:

At time 0 (today):

- Buy one share of the stock
- Borrow the PV of the option exercise price X
- Write a call on the option.

At time T:

- Exercise the option if this is profitable.
- Repay the borrowed funds.

This strategy produces the following cash-flow table:

Today		At Time T	
Action	Cash Flow	$S_T < X$	$S_T \geq X$
Buy stock	$-S_0$	$+S_T$	$+S_T$
Borrow PV of X	$+Xe^{-rt}$	$-X$	$-X$
Write call	$+C_0$	0	$-(S_T - X)$
Total	$-S_0 + Xe^{-rT} + C$	$S_T - X < 0$	0

Note that at time T, the cash flow resulting from this option is either negative (if the call is not exercised) or zero (when $S_T \geq X$). Now a financial asset (in this case: the combination of purchasing a stock, borrowing X, and writing a call) which has only non-positive payoffs in the future must have a positive initial cash flow; this shows that:

$$C_0 - S_0 + Xe^{-rT} > 0 \quad \text{or} \quad C_0 > S_0 - Xe^{-rT}$$

To finish the proof, we note that in no case can the value of a call be less than zero. Thus we have that $C_0 \geq \max(S_0 - Xe^{-rT}, 0)$, which proves the proposition.

Proposition 1 has an immediate and very interesting consequence: In many cases the early-exercise feature of an American call option is worthless; this means that an American call option can be valued as if it were a European call. The precise conditions are the following:

PROPOSITION 2 Consider an American call option written on a stock which will not pay any dividends before the option's expiration date T. Then it is never optimal to exercise the option before its maturity.

Proof of Proposition 2 Suppose the holder of the option is considering exercising it early, at some date $t < T$. The only reason to consider such early exercise is that $S_t - X > 0$, where S_t is the price of the underlying stock at time t. However, by Proposition 1 the market value of the option at time t is at least $S_t - Xe^{-r(T-t)}$, where r is the risk-free rate of interest. Since $S_t - Xe^{-r(T-t)} > S_t - X$, it follows that the option's holder is better off selling the option in the market than exercising it.

Proposition 2 means that many American call options can be priced as if they were European calls. Note that this is not true for American puts, even if the underlying stock pays no dividends; in Chapter 16 we give some examples in the context of a binomial model.

PROPOSITION 3 (PUT BOUNDS) The lower bound on the value of a put option is:

$$P_0 \geq \max(0, Xe^{-rT} - S_0)$$

Proof of Proposition 3 The proof of this proposition has the same form as the proof of the previous theorem. We set up a table of strategies:

Today		At Time T	
Action	Cash Flow	$S_T < X$	$S_T \geq X$
Short stock	$+S_0$	$-S_T$	$-S_T$
Lend PV of X	$-Xe^{-rT}$	$+X$	$+X$
Write put	$+P_0$	$-(X - S_T)$	0
Total	$P_0 + S_0 - Xe^{rT}$	0	$X - S_T < 0$

Since the strategy has only negative or zero payoffs in the future, it must have a positive cash flow today, so that we can conclude that:

$$P_0 - Xe^{-rT} + S_0 \geq 0$$

Combined with the fact that in no case can a put value be negative, this proves the proposition.

PROPOSITION 4 (PUT-CALL PARITY) Let C_0 be the price of a European call with exercise price X written on a stock whose current price is S_0. Let P_0 be the price of a European put on the same stock with the same exercise price X. Suppose both put and call have exercise date T, and suppose that the continuously compounded interest rate is r. Then:

$$C_0 + Xe^{-rT} = P_0 + S_0$$

Proof of Proposition 4 The proof is similar in style to that of the two previous propositions. We consider a combination of the four assets (the put, the call, the stock, and a bond), and show that the pricing relation must hold:

Today		At Time T	
Action	Cash Flow	$S_T < X$	$S_T \geq X$
Buy call	$-C_0$	0	$+S_T - X$
Buy a bond with payoff X at time T	$-Xe^{-rT}$	X	X
Write a put	$+P_0$	$-(X - S_T)$	0
Short one share of the stock	$+S_0$	$-S_T$	$-S_T$
Total	$-C_0 - Xe^{-rT} + P_0 + S_0$	0	0

Since the strategy has future payoffs which are zero no matter what happens to the price of the stock, it follows that the initial cash flow of the strategy must also be zero.[6] This means that:

$$C_0 + Xe^{-rT} - P_0 - S_0 = 0$$

which proves the proposition.

6. This is a fundamental fact of finance: If a financial strategy has future payoffs which are identically zero, then its current cost must also be zero. Likewise, if a financial strategy has future payoffs which are non-negative, then its time-zero payoff must be negative (that is, it must cost something).

Put-call parity states that the stock price S_0, the price of a call C_0 with exercise price X and the price of a put P_0 with exercise price X, are simultaneously determined with the interest rate r. Following is an illustration which uses the call price C_0, the option exercise price X, the current stock price S_0, and the interest rate r to compute the price of a put with exercise price X and time to maturity T:

	A	B	C
1	**PUT-CALL PARITY**		
2	Current stock price, S_0	55	
3	Option time to maturity, T	0.5	
4	Option exercise price, X	60	
5	Interest rate, r	10%	
6	Call price, C_0	3	
7	Put price, P_0	5.0738	<-- =B6+B4*EXP(-B5*B3)-B2
8			
9	This spreadsheet uses put-call parity to derive the put price P_0 from the call price C_0, the interest rate r, the time to maturity T, and the exercise price X.		

PROPOSITION 5 (CALL OPTION PRICE CONVEXITY) Consider three European calls, all written on the same non-dividend paying stock and with the same expiration date T. We suppose that the exercise prices on the calls are X_1, X_2, and X_3, and denote the associated call prices by C_1, C_2, and C_3. We further assume that $X_2 = \frac{X_1 + X_3}{2}$. Then

$$C_2 < \frac{C_1 + C_3}{2}$$

It follows that the call option price is a convex function of the exercise price.

Proof of Proposition 6 To prove the proposition, we consider the following strategy of three calls with exercise prices $X_1 < X_2 < X_3$. We suppose one call each with exercise price X_1 and X_3 is purchased and that two calls with exercise price X_2 are written. Such a strategy is commonly called a "butterfly."[7]

At Time 0		At Time T			
Action	Cash Flow	$S_T < X_1$	$X_1 \le S_T < X_2$	$X_2 \le S_T < X_3$	$X_3 \le S_T$
Buy call with exercise price X_1	$-C_1$	0	$S_T - X_1$	$S_T - X_1$	$S_T - X_1$
Buy call with exercise price X_3	$-C_3$	0	0	0	$S_T - X_3$
Write two calls with exercise price X_2	$+2C_2$	0	0	$-2(S_T - X_2)$	$-2(S_T - X_2)$
Total	$2C_2 - C_1 - C_3$	0	$S_T - X_1 \ge 0$	$2X_2 - X_1 - S_T = X_3 - S_T > 0$	0

Since the payoffs in the future are all non-negative (with a positive probability of being positive, it follows that the initial cash flow from the position must be negative:

$$2C_2 - C_1 - C_3 < 0 \Rightarrow C_2 < \frac{C_1 + C_3}{2}$$

This proves the proposition. (Note that the assumption that $X_2 = \frac{X_1 + X_3}{2}$ is made for convenience and does not affect the generality of the argument.)

Without proof we state a similar proposition for puts:

PROPOSITION 6 (PUT PRICE CONVEXITY) Consider three European puts, all written on the same non-dividend paying stock and with the same expiration date T. We suppose that the exercise prices on the calls are X_1, X_2, and X_3, and denote the associated put prices by P_1, P_2, and P_3. We further assume that $X_2 = \frac{X_1 + X_3}{2}$. Then the put price is a convex function of the exercise price:

7. Any strategy of three options (calls or puts) with three exercise prices in which one option each of the extreme exercise price is either bought or written and in which two options with the middle exercise price are held in the opposite position is called a butterfly.

$$P_2 < \frac{P_1 + P_3}{2}$$

Without proof we state a similar proposition for a butterfly composed of puts.

PROPOSITION 7 (CALL OPTION BOUNDS WITH A KNOWN FUTURE DIVIDEND) Consider a call with exercise price X and maturity date T. Suppose that at some time $t < T$, the stock will, with certainty, pay a dividend D. Then the lower bound on the call option price is given by:

$$C_0 \geq \max(S_0 - De^{-rt} - Xe^{-rT}, 0)$$

Proof of Proposition 7 The proof involves only a minor modification of the proof of Proposition 1.

Today		Time t	At Time T	
Action	Cash Flow		$S_T < X$	$S_T \geq X$
Buy stock	$-S_0$	$+D$	$+S_T$	$+S_T$
Borrow the PV of the dividend D	$+De^{-rT}$	$-D$		
Borrow PV of X	$+Xe^{-rT}$		$-X$	$-X$
Write call	$+C_0$		0	$-(S_T - X)$
Total	$-S_0 + De^{-rt} + Xe^{-rT} + C_0$	0	$S_T - X < 0$	0

This proves the proposition.

15.7 Summary

This chapter summarizes the basic definitions and features of options. It is, however, by no means an adequate introduction to these complex securities for those with no preknowledge. To decipher the mysteries of options, we recommend the introductory chapters of a good option text.

Exercises

1. When you looked at the newspaper quotes for options on ABC stock, you saw that a February call option with $X = 37.5$ is priced at 6.375, whereas the April call option with the same exercise price is priced at 6. Can you devise an arbitrage out of these prices? Do you have an explanation for the newspaper quotes?

2. An American call option is written on a stock whose price today is $S = 50$. The exercise price of the call is $X = 45$.

   a. If the call price is 2, explain how you would use arbitrage to make an immediate profit.
   b. If the option is exercisable at time $T = 1$ year and if the interest rate is 10%, what is the minimum price of the option? Use Proposition 1.

3. A European call option is written on a stock whose current price $S = 80$. The exercise price $X = 80$, the interest rate $r = 8\%$, and the time to option exercise $T = 1$. The stock is assumed to pay a dividend of 3 at time $t = ½$. Use Proposition 7 to determine the minimum price of the call option.

4. A put with an exercise price of 50 has a price of 6 and a call on the same stock with an exercise price of 60 has a price of 10. Both put and call have the same expiration date. On the same set of axes, draw the profit diagram for:

   a. One put bought and one call bought.
   b. Two puts bought and one call bought.
   c. Three puts bought and one call bought.
   d. All three lines cross each other for the same value of S_T. Derive this value.

5. Consider the following two calls:

   - Both calls are written on shares of ABC Corp. whose current share price is $100. ABC does not pay any dividends.
   - Both calls have one year to maturity.
   - One call has $X_1 = 90$ and has price of 30; the second call has $X_2 = 100$ and has price of 20.
   - The riskless, continuously compounded interest rate is 10%.

   By designing a spread (i.e., buying one call and writing another) position, show that the difference between the two call prices is *too large* and that a riskless arbitrage exists.

6. A share of ABC Corp. sells for $95. A call on the share with exercise price $90 sells for $8.

   a. Graph the profit pattern from buying one share and one call on the share.
   b. Graph the profit pattern from buying one share and two calls.
   c. Consider the profit pattern from buying one share and calls. At which share price do all of the profit lines cross?

7. A European call with a maturity of 6 months and exercise price $X = 80$ is written on a stock whose current price is 85 is selling for $12.00; a European put written on the same stock with the same maturity and with the same exercise price is selling for $5.00. If the annual interest rate (continuously compounded) is 10%, construct an arbitrage from this situation.

8. Prove Proposition 6. Then solve the following problem: Three puts on shares of XYZ with the same expiration date are selling at the following prices:

   Exercise price 40: 6

   Exercise price 50: 4

   Exercise price 60: 1

   Show an arbitrage strategy which allows you to profit from these prices and prove that it works.

9. The current stock price of ABC Corp. is 50. Prices for six-month calls on ABC are given in the table below. Draw a profit diagram of the following strategy: Buy one 40 call, write two 50 calls, buy one 60 call, and write two 70 calls.

Call	Price
40	16.5
50	9.5
60	4.5
70	2

10. Consider the following option strategy, which consists only of calls:

Exercise Price	Bought/Written? Number?	Price per Call Option
20	1 written	45
30	2 bought	33
40	1 written	22
50	1 bought	18
60	2 written	17
70	1 bought	16

   a. Draw the profit diagram for this strategy.

   b. The prices given include one violation of an arbitrage condition. Identify this violation and explain.

11. A share of Formila Corp. is currently trading at \$38.50, and a 1-year call option on Formila with X = \$40 is trading at \$3. The risk-free interest rate is 4.5%.

   a. What should be the price of a 1-year put option on the stock with X = \$40? Why?

   b. If the price of a put is \$2, construct an arbitrage strategy.

   c. If the price of a put is \$4, construct an arbitrage strategy.

16 The Binomial Option Pricing Model

16.1 Overview

Next to the Black-Scholes model (discussed in Chapter 17), the binomial option pricing model is the most widely used option pricing model. It has many advantages: It is a simple model, which—in addition to giving many insights into option pricing—is easily programmed and adapted to numerous, and often quite complicated, option pricing problems. When extended to many periods, the binomial model becomes one of the most powerful ways of valuing securities like options whose payoffs are contingent on the market prices of other assets.

The binomial model depends on using state prices to compute the values of risky assets. When the state pricing principles underlying the model are understood, we gain deeper insight into the economics of contingent asset pricing. In this chapter, which illustrates the simple uses of the binomial model, we devote a considerable amount of space to deriving and using state prices. In Chapters 29 and 30 we return to the binomial model and use it in the Monte Carlo pricing of contingent securities.

16.2 Two-Date Binomial Pricing

To illustrate the use of the binomial model, we start with the following very simple example:

- There is one period and two dates, date 0 = today and date 1 = one year from now.
- There are two "fundamental" assets: A stock and a bond. There is also a derivative asset, a call option written on the stock.
- The stock price today is \$50 and at date 1 will either go up by 10% or go down by 3%.
- The one-period interest rate is 6%.
- The call option matures at date 1 and has exercise price X = \$50.

Here is a picture from a spreadsheet which incorporates this model. Notice that in cells B2, B3, and B6 we have used values for 1 plus the 10% up move, 1 plus the –3% down move, and 1 plus the 6% interest. We use capital letters U, D, and R to denote these values.[1]

1. Should there be a need to distinguish between 1.10—1 plus the 10% up move of the stock—and 10% (the up move itself), we will use U for the former and lowercase u for the latter.

	A	B	C	D	E	F	G	H	I	J
1	**BINOMIAL OPTION PRICING IN A ONE-PERIOD MODEL**									
2	Up, U	1.10								
3	Down, D	0.97								
4										
5	Initial stock price	50.00								
6	Interest rate, R	1.06								
7	Exercise price	50.00								
8										
9		**Stock price**					**Bond price**			
10				55.00	<-- =$B$11*B2				1.06	<-- =$G$11*$B$6
11		50.00						1.00		
12				48.50	<-- =$B$11*B3				1.06	<-- =$G$11*$B$6
13										
14		**Call option**								
15				5.00	<-- =MAX(D10-$B$7,0)					
16		???								
17				0.00	<-- =MAX(D12-$B$7,0)					

We wish to price the call option. We do this by showing that there is a *combination of the bonds and stocks which exactly replicates the call option's payoffs*. To show this, we use some basic linear algebra; suppose we find A shares of the stock and B bonds such that:

$$55A + 1.06B = 5$$
$$48.5A + 1.06B = 0$$

This system of equations solves to give

$$A = \frac{5}{55 - 48.5} = 0.7692$$
$$B = \frac{0 - 48.5A}{1.06} = -35.1959$$

Thus purchasing 0.77 of a share of the stock and borrowing \$35.20 at 6% for one period will give payoffs of \$5 if the stock price goes up and \$0 if the stock price goes down—the payoffs of the call option. It follows that the price of the option must be equal to the cost of replicating its payoffs; that is,

$$\textit{Call option price} = 0.7692 * \$50 - \$35.1959 - \$3.2656$$

This logic is called "pricing by arbitrage": If two assets or sets of assets (in our case—the call option and the portfolio of 0.77 of the stock and –\$35.20 of the bonds) have the same payoffs, they must have the same market price.

	A	B	C	D	E	F	G
19	**Solving the portfolio problem: A shares + B bonds combine to give option payoffs**						
20	A	0.7692	<-- =D15/(D10-D12)				
21	B	-35.1959	<-- =-D12*B20/B6				
22							
23	Call price	3.2656	<-- =B20*B5+B21				

Applying the same logic to a put gives the put price as 0.4354:

	A	B	C	D	E	F	G
1	**BINOMIAL PUT OPTION PRICING IN A ONE-PERIOD MODEL**						
2	Up, U	1.10					
3	Down, D	0.97					**Solving the put price**
4							55*A+1.06*B=0
5	Initial stock price	50.00					48.5*A+1.06*B=1.5
6	Interest rate, R	1.06					
7	Exercise price	50.00					A=-1.5/(55-48.5)
8		**Put option**					B=-55*A/1.06
9				0.00	<-- =MAX($B$7-B5*B2,0)		
10		???					
11				1.50	<-- =MAX($B$7-B5*B3,0)		
12							
13	**Solving the portfolio problem: A shares + B bonds combine to give option payoffs**						
14	A	-0.2308	<-- =-D11/(B5*(B2-B3))				
15	B	11.9739	<-- =-B5*B2*B14/B6				
16							
17	Put price	0.4354	<-- =B14*B5+B15				

In succeeding sections we show that this simple arbitrage argument can be extended to multiple periods. But in the meantime we confine ourselves in the next section to generalizing the logic.

16.3 State Prices

There is actually a simpler (and more general) way to solve this problem: Viewed from today, there are only two possibilities for next period: Either the stock price goes up or it goes down. Think about the market determining a price q_U for \$1 in the “up” state of the world and a price q_D for \$1 in the “down” state of the world. Then both the bond and the stock have to be priced using these *state prices*:

$$q_U * S * U + q_D * S * D = S \Rightarrow q_U U + q_D D = 1$$
$$q_U * R + q_D * R = 1$$

The state prices are thus an illustration of the linear pricing principle: If the stock price can move up in one period by a factor U and down by a factor D, and if 1 plus the one-period interest rate is R, then any other asset will be priced by discounting its payoff in the "up" state by q_U and by discounting its payoff in the "down" state by q_D.

The two equations above solve to give:

$$q_U = \frac{R - D}{R(U - D)}, \quad q_D = \frac{U - R}{R(U - D)}$$

In our case these state prices are given by:

	A	B	C
1	**DERIVING THE STATE PRICES**		
2	Up, U	1.10	
3	Down, D	0.97	
4	Interest rate, R	1.06	
5			
6	**State prices**		
7	q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))
8	q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))
9			
10	**Check: Confirm that state prices actually price the stock and the bond**		
11	Pricing the stock: 1 = q_U*U+q_D*D?	1	<-- =B7*B2+B8*B3
12	Pricing the bond: 1/R = q_U+q_D ?	1.06	<-- =1/(B7+B8)

In rows 11 and 12 we check that the state prices indeed give back the interest rate and the stock price.

We can now use the state prices to price the call and the put on the stock and also to establish that put-call parity holds. The call and the put options should be priced by:

$$C = q_U \max(S * U - X, 0) + q_D \max(S * D - X, 0)$$
$$P = q_U \max(X - S * U, 0) + q_D \max(X - S * D, 0)$$

or if priced by put call parity

$$P = C + PV(X) - S$$

In a spreadsheet:

	A	B	C
1	**BINOMIAL OPTION PRICING WITH STATE PRICES IN A ONE-PERIOD (TWO-DATE) MODEL**		
2	Up, U	1.10	
3	Down, D	0.97	
4	Interest rate, R	1.06	
5	Initial stock price, S	50.00	
6	Option exercise price, X	50.00	
7			
8	**State prices**		
9	q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))
10	q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))
11			
12	**Pricing the call and the put**		
13	Call price	3.2656	<-- =B9*MAX(B5*B2-B6,0)+B10*MAX(B5*B3-B6,0)
14	Put price	0.4354	<-- =B9*MAX(B6-B5*B2,0)+B10*MAX(B6-B5*B3,0)
15			
16	**Put-call parity**		
17	Stock + put	50.4354	<-- =B5+B14
18	Call + PV(X)	50.4354	<-- =B13+B6/B4
19			
20	**Note about PV(X) in put-call parity:** In the continuous-time framework (the standard Black-Scholes framework), PV(X) = X*Exp(-r*T). Because the framework here is discrete time, PV(X) is also discrete-time: PV(X)=X/(1+r)=X/R.		

The formulas we use (with $S = 50$, $X = 50$, $U = 1.10$, $D = 0.97$, $R = 1.06$) are:

$$C = q_U \max(S*U - X, 0) + q_D \max(S*D - X, 0)$$
$$= 0.6531*5 + 0.2903*0 = 3.2657$$

for the call, and

$$P = q_U \max(X - S*U, 0) + q_D \max(X - S*D, 0)$$
$$= 0.6531*\max(50-55, 0) + 0.2903*\max(50-48.5, 0) = 0.4354$$

for the put. As expected—the put-call parity theorem holds for this particular put and call (cells B17:B18):

$$P + S = 0.4354 + 50 = C + \frac{X}{R} = 3.27 + \frac{50}{1.06}$$

State Prices or Risk-Neutral Prices?

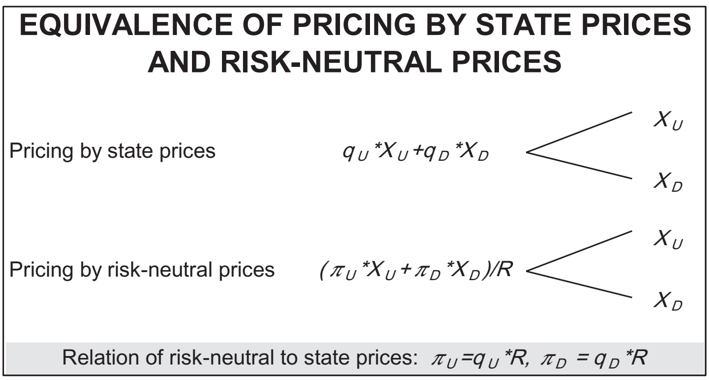
Multiplying the state prices by 1 plus the interest rate, R, gives the *risk-neutral prices*: $\pi_U = q_U R$, $\pi_D = q_D R$. The risk-neutral prices look like a probability distribution of the states, since they sum to 1:

$$\pi_U + \pi_D = q_U R + q_D R = \frac{R - D}{R(U - D)} R + \frac{U - R}{R(U - D)} R = 1$$

Furthermore, there is a fundamental equivalence of pricing by the risk-neutral prices and pricing by the state prices. Suppose an asset has state-dependent payoffs X_U in the "up" state and X_D in the "down" state of a 2-date model. Then the date 0 price of the asset using the state prices is $q_U X_U + q_D X_D$ and the date 0 price of the asset using the risk-neutral prices is the discounted expected asset payoff, where the expectation is computed using the risk-neutral prices as if they are the actual state probabilities:

$$\frac{\pi_U X_U + \pi_D X_D}{R} = \frac{\text{"}Expected\text{" } asset\ payoff\ using\ risk\text{-}neutral\ prices}{1 + r}$$
$$= q_U X_U + q_D X_D$$

Pricing with state prices or pricing with risk-neutral prices is, of course, the same. This author prefers to use state prices, but many researchers are more comfortable using the pseudo-probabilities of the risk-neutral prices and then discounting the "expected" payoffs.



To drive home the equivalence between state prices and risk-neutral prices, we close this subsection with a numerical example:

	A	B	C
1	**RISK-NEUTRAL PRICES OR STATE PRICES?**		
2	Up, U	1.10	
3	Down, D	0.97	
4	Interest rate, R	1.06	
5	Initial stock price, S	50.00	
6	Option exercise price, X	50.00	
7			
8	**State prices**		
9	q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))
10	q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))
11			
12	**Risk-neutral prices**		
13	$\pi_U = q_U$*R	0.6923	<-- =B9*$B$4
14	$\pi_D = q_D$*R	0.3077	<-- =B10*$B$4
15			
16	**Pricing the call and the put using state prices**		
17	Call price	3.2656	<-- =B9*MAX(B5*B2-B6,0)+B10*MAX(B5*B3-B6,0)
18	Put price	0.4354	<-- =B9*MAX(B6-B5*B2,0)+B10*MAX(B6-B5*B3,0)
19			
20	**Pricing the call and the put using risk-neutral prices**		
21	Call price	3.2656	<-- =(B13*MAX(B5*B2-B6,0)+B14*MAX(B5*B3-B6,0))/B4
22	Put price	0.4354	<-- =(B13*MAX(B6-B5*B2,0)+B14*MAX(B6-B5*B3,0))/B4

16.4 The Multi-Period Binomial Model

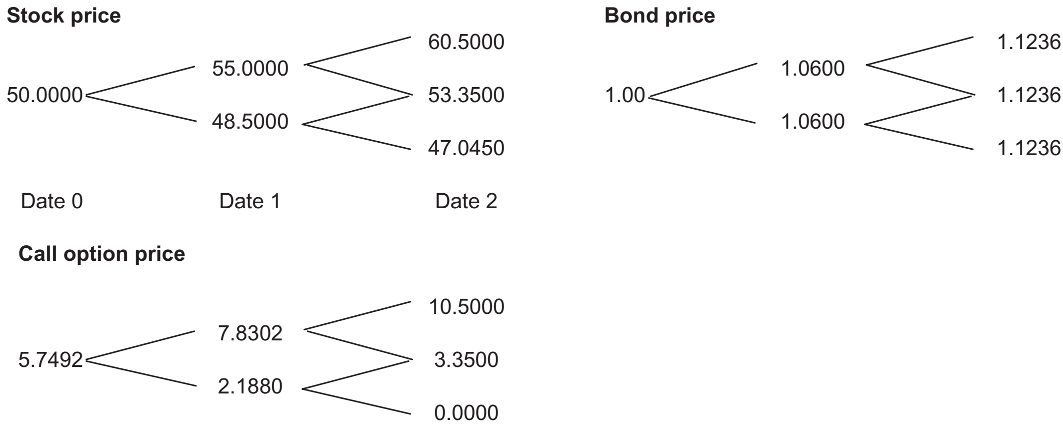
The binomial model can easily be extended to more than one period. Consider, for example, a two-period (three-date) binomial model that has the following characteristics:

- In each period the stock price goes up by 10% or down by −3% from what it was in the previous period. This means that $U = 1.10$, $D = 0.97$.
- In each period the interest rate is 6%, so that $R = 1.06$.

Because U, D, and R are the same in each period,

$$q_U = \frac{R - D}{R(U - D)} = 0.6531,\ q_D = \frac{U - R}{R(U - D)} = 0.2903$$

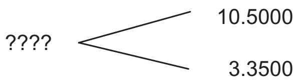
We can now use these state prices to price a call option written on the stock after two periods. As before, we assume that the stock price is \$50 initially and that the call exercise price is $X = 50$ after two periods. This gives the following picture:



How was the call option price of 5.7492 determined? To do this, we go backward, starting at period two:

At date two: At the end of two periods the stock price is either \$60.50 (corresponding to two "up" movements in the price), \$53.35 (one "up" and one "down" movement), or \$47.05 (two "down" movements in the price). Given the exercise price of $X = 50$, this means that the terminal option payoff in period 2 is \$10.50, \$3.35, or \$0.

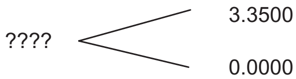
At date one: At date one, there are two possibilities: Either we have reached an "up" state, in which case the current stock price is \$55 and the option will pay off \$10.50 or \$3.35 in the next period:



We use the state prices of $q_u = 0.6531$, $q_d = 0.2903$ to price the option at this state:

$$\textit{Option price at "up" state, date } 1 = 0.6531 * 10.50 + 0.2903 * 3.35 = 7.8302$$

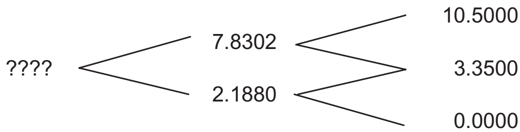
The alternative possibility is that we're in the "down" state of period one:



Using the same state prices (which, after all, depend only on the "up" and "down" movements of the stock price and the interest rate), we get:

$$\textit{Option price at "down" state, date 1} = 0.6531 * 3.35 + 0.2903 * 0 = 2.1880$$

At date 0: Going backward in this way, we've now filled in the following picture:



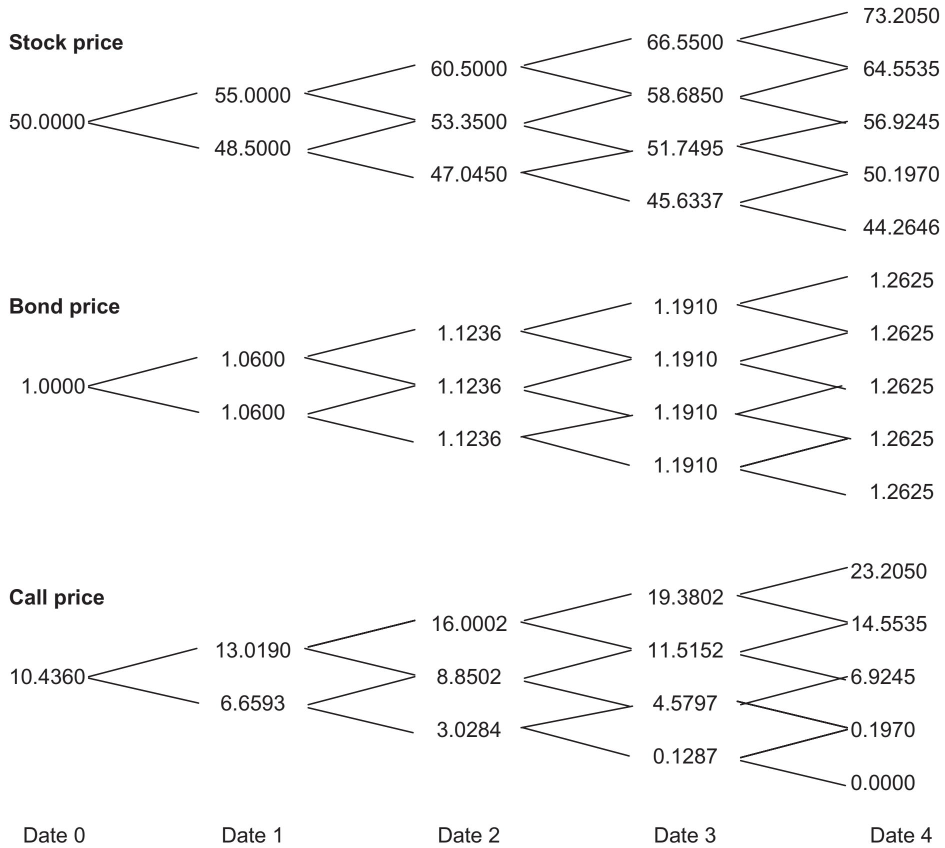
Thus at period 0 the buyer of an option owns a security which will be worth $7.83 if the underlying stock has an "up" movement in its return and which will be worth $2.19 if the stock has a "down" movement in its return. We can again use the state prices to value this option:

$$\textit{Option price at date 0} = 0.6531 * 7.830 + 0.2903 * 2.188 = 5.749$$

	A	B	C	D	E	F	G	H	I	J	K
1	**BINOMIAL OPTION PRICING WITH STATE PRICES IN A TWO-PERIOD (THREE-DATE) MODEL**										
2	Up, U	1.10									
3	Down, D	0.97		**State prices**							
4	Interest rate, R	1.06		q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))					
5	Initial stock price, S	50.00		q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))					
6	Option exercise price, X	50.00									
7											
8	**Stock price**						**Bond price**				
9					60.5000						1.1236
10			55.0000						1.0600		
11	50.0000				53.3500		1.00				1.1236
12			48.5000						1.0600		
13					47.0450						1.1236
14											
15											
16	**Call option price**										
17			=q_U*E18+q_D*E20								
18					10.5000	<-- =MAX(E9-$B$5,0)					
19			7.8302								
20	5.7492				3.3500	<-- =MAX(E11-$B$5,0)					
21			2.1880								
22	=q_U*C19+q_D*C21				0.0000	<-- =MAX(E13-$B$5,0)					
23			=q_U*E20+q_D*E22								
24											

Extending the Binomial Pricing Model to Many Periods

It is clear that the logic of the above example can be extended to many periods. Here's another Excel graphic showing a 5-date model using the same "up" and "down" parameters as before:



Do You Really Have to Price Everything Backward?

The answer is "no." There's no necessity to price the call price payoffs "backward" at each node back from the terminal date, *as long as the call is European*.[2] It is enough to price each of the terminal payoffs by the state prices, providing you count properly the number of paths to each terminal node. Here's an illustration, using the same example:

	A	B	C	D	E	F	G	H
1	**BINOMIAL OPTION PRICING WITH STATE PRICES IN A FOUR-PERIOD (FIVE-DATE) MODEL**							
2	Up, U	1.10						
3	Down, D	0.97		**State prices**				
4	Interest rate, R	1.06		q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))		
5	Initial stock price, S	50.00		q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))		
6	Option exercise price, X	50.00						
7								
8								
9	**Number of "up" steps at terminal date**	**Number of "down" steps at terminal date**	**Terminal stock price = S*U^(# up) *D^(# down)**	**Option payoff at terminal state**	**State price for terminal date = q_U^(# up) *q_D^(# down)**	**Number of paths to terminal state**	**Value =payoff*state price*#paths**	
10	4	0	73.2050	23.2050	0.1820	1	4.2224	
11	3	1	64.5535	14.5535	0.0809	4	4.7078	
12	2	2	56.9245	6.9245	0.0359	6	1.4933	
13	1	3	50.1970	0.1970	0.0160	4	0.0126	
14	0	4	44.2646	0.0000	0.0071	1	0.0000	
15						Call price	10.4360	<-- =SUM(G10:G14)
16						Put price	0.0407	<-- =G15+B6/B4^4-B5
17								
18	**Notes**							
19	There are 5 dates in this model (0, 1, ... , 5) but only 4 periods and thus only 4 possible "up" or "down" steps.							
20	The put price in cell G16 is computed using put-call parity: put = call + PV(X) - stock							

Here is an explanation for the table above: For each terminal option payoff, we consider:

2. When we discuss American options in section 16.5 we will see that backward pricing is critical.

How was this terminal payoff reached? How many "up" steps did the stock make and how many "down" steps did it make?		Example: The terminal payoff of 14.5535 arises when the stock price is 64.5535. This happens when the stock price goes up 3 times and down once.
What is the price per dollar of the payoff in the particular state?	$State\ price = q_U^{\#up\,steps} q_D^{\#down\,steps}$	Example: The value at time 0 of the terminal payoff considered above is $0.6531^3 * 0.2903^1 =$ 0.0809.
How many paths are there with the same terminal payoff?	The answer is given by the binomial coefficient $\binom{Number\ of\ periods}{Number\ of\ \text{“}up\text{”}\ steps}$	Example: There are $\binom{4}{3} = 4$ paths which give the terminal stock price of 64.5535. The Excel function **Combin(4,3)** gives this binomial coefficient.
What is the value at time 0 of a particular terminal payoff?	The answer is the product of the payoff times the price times the number of paths.	Example: 14.5535*0.0809*4 = 4.7078.
What is the value at time 0 of the option?	The sum of the values of each payoff.	Total value: 10.4360. This is the multi-period call option value in the five date (four-period) binomial model.

European puts can be priced either by using the logic above or—as in cell G16 above—by using put-call parity.

To recapitulate: The price of a European call option in a binomial model with *n* periods is given by:

$$Call\ price = \sum_{i=0}^{n} \binom{n}{i} q_U^i q_D^{n-i} \max(S * U^i D^{n-i} - X, 0)$$

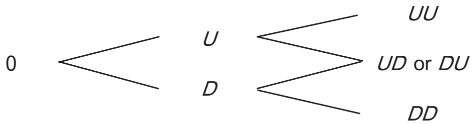
$$Put\ price = \begin{cases} \sum_{i=0}^{n} \binom{n}{i} q_U^i q_D^{n-i} \max(X - S * U^i D^{n-i}, 0) & \textit{Direct pricing} \\ Call\ price + \dfrac{X}{R^n} - S & \textit{Using put-call parity} \end{cases}$$

In section 16.7 we implement these formulas in VBA.

16.5 Pricing American Options Using the Binomial Pricing Model

We can use the binomial pricing model to calculate the prices of American options as well as European options. We reconsider the basic model above, in which $U_p = 1.10$, $Down = 0.97$, $R = 1.06$, $S = 50$, $X = 50$. We examine the three-date version of the model. Recall from Chapter 15 that an American call option on a non-dividend-paying stock has the same value as a European call option. It is thus more interesting to start with the pricing of an American put. The payoff patterns for the stock and the bond have been given above, and it remains only to consider the payoff patterns for a put option with $X = 50$. We reference the states of the world by using the following labels:

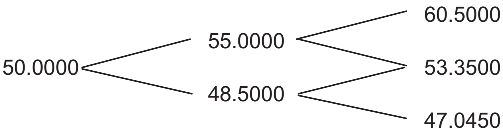
State labels



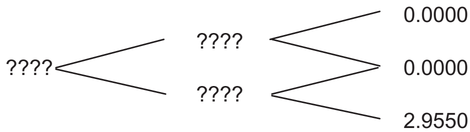
Put Payoffs at Date Three

Here are the values of the stock and the date three put payoffs:

Stock price



American put payoffs

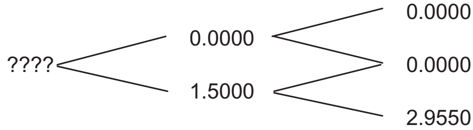


At date two, the holder of an American put can choose whether to hold the put or to exercise it. We now have the following value function:

$$\begin{matrix} \textit{Put value} \\ \textit{at date 2} \\ \textit{state } U \end{matrix} = \max \begin{cases} \textit{Put value if exercised} = \max(X - S_U, 0) \\ q_U * \textit{Put payoff in state } UU + q_D * \textit{Put payoff in state } UD \end{cases}$$

A similar function holds for the put value in state *d* at date two. The resulting tree now looks like:

American put payoffs



Here's the explanation:

- In state U, the put is valueless. When the stock price is \$55, it is not worthwhile to early-exercise the put, since $\max(X - S_U, 0) = \max(50 - 55, 0) = 0$. On the other hand, since the future put payoffs from state U are zero, state-dependent present value of these future payoffs (the second line in the previous formula) is also zero.
- In state D, on the other hand, the holder of the put gets $\max(50 - 48.5, 0) = 1.5$ if he exercises the put; however, if he holds the put without exercise, its market value is the state-dependent value of the future payoffs:

$$q_U * 0 + q_D * 2.9550 = 0.6531 * 0 + 0.2903 * 2.9550 = 0.8578$$

It is clearly preferable to exercise the put in this state rather than to hold onto it.

At date 0, a similar value function recurs:

$$\begin{matrix} \textit{Put value} \\ \textit{at date 0} \end{matrix} = \max \begin{cases} \textit{Put value if exercised} = \max(X - S_0, 0) \\ q_U * \textit{Put payoff in state } U + q_D * \textit{Put payoff in state } D \end{cases}$$

The spreadsheet is given below:

	A	B	C	D	E	F	G	H	I	J	K
1	**AMERICAN PUT PRICING IN A TWO-PERIOD MODEL**										
2	Up, U	1.10									
3	Down, D	0.97		**State prices**							
4	Interest rate, R	1.06		q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))					
5	Initial stock price, S	50.00		q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))					
6	Option exercise price, X	50.00									
7											
8											
9											
10	**Stock price**						**Bond price**				
11					60.5000						1.1236
12			55.0000						1.0600		
13	50.0000				53.3500		1.0000				1.1236
14			48.5000						1.0600		
15					47.0450						1.1236
16											
17											
18	**American put option**										
19						=MAX(MAX(X-S*U,0),q_U*put_payoff$_{UU}$+q_D*put_payoff$_{UD}$)					
20					0.0000						
21			0								
22	0.4354				0.0000						
23			1.5								
24					2.9550	=MAX(MAX(X-S*D,0),q_U*put_payoff$_{UD}$+q_D*put_payoff$_{DD}$)					
25											
26											
27		=MAX(MAX(X-S,0),q_U*put_value$_U$+q_D*put_value$_D$)									
28											
29											
30	**European put option**										
31					0.0000						
32			0								
33	0.2490				0.0000						
34			0.8578								
35					2.9550						

We can use the same logic to price an American call option, though—following Proposition 2 of Chapter 15, we know that the value of an American and a European call should coincide. And so they do:

	A	B	C	D	E	F	G	H	I	J	K
1	**AMERICAN CALL PRICING IN A TWO-PERIOD MODEL**										
2	Up, U	1.10									
3	Down, D	0.97		**State prices**							
4	Interest rate, R	1.06		q_U	0.6531	<-- =(B4-B3)/(B4*(B2-B3))					
5	Initial stock price, S	50.00		q_D	0.2903	<-- =(B2-B4)/(B4*(B2-B3))					
6	Option exercise price, X	50.00									
7											
8											
9											
10	**Stock price**						**Bond price**				
11					60.5000						1.1236
12			55.0000						1.0600		
13	50.0000				53.3500		1.0000				1.1236
14			48.5000						1.0600		
15					47.0450						1.1236
16											
17											
18	**American call option**										
19						=MAX(MAX(S*U-X,0),q_U*call_payoff$_{UU}$+q_D*call_payoff$_{UD}$)					
20					10.5000						
21			7.8302								
22	5.7492				3.3500						
23			2.1880								
24					0.0000	=MAX(MAX(S*D-X,0),q_U*call_payoff$_{UD}$+q_D*call_payoff$_{DD}$)					
25											
26											
27		=MAX(MAX(S-X,0),q_U*call_value$_U$+q_D*call_value$_D$)									
28											
29											
30	**European call option**										
31					10.5000						
32			7.8302								
33	5.7492				3.3500						
34			2.1880								
35					0.0000						

16.6 Programming the Binomial Option Pricing Model in VBA

The pricing procedure used in the above examples can easily be programmed using Excel's VBA programming language. In the binomial model the price can move *up* or *down* in any time period. If q_U is the state price associated with an up move and if q_D is the state price associated with a down move, then the binomial European option prices are given by:

$$\text{Binomial European call} = \sum_{i=0}^{n} \binom{n}{i} q_U^i q_D^{n-i} \max(S*U^i D^{n-i} - X, 0)$$

$$\text{Binomial European put} = \begin{cases} \sum_{i=0}^{n} \binom{n}{i} q_U^i q_D^{n-i} \max(X - S*U^i D^{n-i}, 0) \\ \text{or by put-call parity} \end{cases}$$

where U is an up move, R is a down move in the stock price, and $\binom{n}{i}$ is the binomial coefficient (the number of up moves in n total moves:

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!}$$

We use Excel's **Combin(n,i)** to give values for the binomial coefficients.

Here are two VBA functions which compute the value of binomial European calls and puts. The function **Binomial_eur_put** uses put-call parity to price the put:

```
Function Binomial_eur_call(Up, Down, Interest, _
Stock, Exercise, Periods)
    q_up = (Interest - Down) / _
    (Interest * (Up - Down))
    q_down = 1 / Interest - q_up
    Binomial_eur_call = 0
  For Index = 0 To Periods
    Binomial_eur_call = Binomial_eur_call _
    + Application.Combin(Periods, Index) _
    * q_up ^ Index * q_down ^ (Periods - Index) _
    * Application.Max(Stock * Up ^ Index * Down _
    ^ (Periods - Index) - Exercise, 0)
  Next Index
End Function

Function Binomial_eur_put(Up, Down, Interest, _
Stock, Exercise, Periods)
  Binomial_eur_put = Binomial_eur_call _
  (Up, Down, Interest, Stock, Exercise, _
  Periods) + Exercise / Interest ^ Periods - Stock
End Function
```

Implementing this in a spreadsheet, we get for the four-period example of section 16.4:

	A	B	C
1	VBA FUNCTIONS FOR CALLS AND PUTS		
2	Up, U	1.10	
3	Down, D	0.97	
4	Interest rate, R	1.06	
5	Initial stock price, S	50.00	
6	Option exercise price, X	50.00	
7	Number of periods, n	4	
8			
9	European call	10.4360	<-- =binomial_eur_call(B2,B3,B4,B5,B6,B7)
10	European put	0.0407	<-- =binomial_eur_put(B2,B3,B4,B5,B6,B7)
11			
12	**Checking put-call parity**		
13	Stock + put	50.0407	<-- =B5+B10
14	Call + PV(X)	50.0407	<-- =B9+B6/B4^B7

American Put Pricing

Proposition 2 in section 15.6 states that the price of an American call on a non-dividend-paying stock is the same as that of a European option. The pricing of an American put, however, can be different. The following VBA function below uses a binomial option pricing model like the one from section 16.5 to price American puts:

```
Function Binomial_amer_put(Up, Down, Interest, _
Stock, Exercise, Periods)
    q_up = (Interest - Down) / (Interest * _
    (Up - Down))
    q_down = 1 / Interest - q_up

    Dim OptionReturnEnd() As Double
    Dim OptionReturnMiddle() As Double
    ReDim OptionReturnEnd(Periods + 1)

    For State = 0 To Periods
    OptionReturnEnd(State) = Application.Max(Exercise _
    - Stock * Up ^ State * Down ^ (Periods - State), 0)
    Next State

    For Index = Periods - 1 To 0 Step -1
        ReDim OptionReturnMiddle(Index)
        For State = 0 To Index
           OptionReturnMiddle(State) = Application.Max _
           (Exercise - Stock * Up ^ State * Down ^ _
           (Index - State), _
           q_down * OptionReturnEnd(State) + q_up * _
           OptionReturnEnd(State + 1))
        Next State
        ReDim OptionReturnEnd(Index)
        For State = 0 To Index
            OptionReturnEnd(State) = _
            OptionReturnMiddle(State)
        Next State
    Next Index
    Binomial_amer_put = OptionReturnMiddle(0)
End Function
```

In this function we use two arrays, called **OptionReturnEnd** and **OptionReturnMiddle**. At each date t, these arrays store the option values for the date itself and the next date—$t + 1$.

Here's an implementation in a spreadsheet, using the two-period, three-date, example from section 16.5:

	A	B	C
1	**VBA FUNCTIONS FOR CALLS AND PUTS**		
2	Up, U	1.10	
3	Down, D	0.97	
4	Interest rate, R	1.06	
5	Initial stock price, S	50.00	
6	Option exercise price, X	50.00	
7	Number of periods, n	2	
8			
9	American put	0.4354	<-- =binomial_amer_put(B2,B3,B4,B5,B6,B7)
10	European put	0.2490	<-- =binomial_eur_put(B2,B3,B4,B5,B6,B7)
11	American call	5.7492	<-- =binomial_amer_call(B2,B3,B4,B5,B6,B7)
12	European call	5.7492	<-- =binomial_eur_call(B2,B3,B4,B5,B6,B7)

The values in cells B9 and B10 are for an American and a European put; these values correspond to those given in section 16.5. In cell B11 we use a function similar to the American put function to price American calls. Unsurprisingly—given Proposition 2 of Chapter 15—this function gives the same value as the binomial European call pricing function.

The VBA function works well for many more periods.[3] In the example below we calculate the value of an American put and call for options which expire at $T = 0.75$ of a year. The stochastic process which defines the stock returns has mean $\mu = 15\%$, and standard deviation $\sigma = 35\%$. The annual continuously compounded interest rate is $r = 6\%$, and each year is divided into 25 subperiods, so that a single period has length $\Delta t = 1/25 = 0.04$. Given these numbers, Up, Down, and R are defined by $Up = e^{\mu\Delta t + \sigma\sqrt{\Delta t}}$, $Down = e^{\mu\Delta t - \sigma\sqrt{\Delta t}}$, $R = e^{r\Delta t}$.

Here is the pricing of an American and a European call and put:

3. The discussion which follows is perhaps best read after Chapters 17 and 26.

	A	B	C
1	**VBA FUNCTIONS FOR CALLS AND PUTS** **n divisions per year, Δt = 1/n** **Up=exp(μ*Δt + σ*sqrt(Δt)), Down = exp(μ*Δt - σ*sqrt(Δt))**		
2	Mean return per year, μ	15%	
3	Standard deviation of annual return, σ	35%	
4	Annual interest rate, r	6%	
5			
6	Initial stock price, S	50.00	
7	Option exercise price, X	50.00	
8	Option exercise date (years)	0.75	
9	Number of divisions of 1 year	25	<-- each year divided into 25 subperiods
10	Δt, the length of one division	0.04	<-- =1/B9
11	Up move per Δt	1.078963	<-- =EXP(B2*B10+B3*SQRT(B10))
12	Down move per Δt	0.938005	<-- =EXP(B2*B10-B3*SQRT(B10))
13	Interest rate per Δt	1.002403	<-- =EXP(B4*B10)
14			
15	Number of periods until maturity, n	19	<-- =ROUND(B8*B9,0)
16			
17	American put	5.1311	<-- =binomial_amer_put(B11,B12,B13,B6,B7,B15)
18	European put	4.9213	<-- =binomial_eur_put(B11,B12,B13,B6,B7,B15)
19	American call	7.1501	<-- =binomial_amer_call(B11,B12,B13,B6,B7,B15)
20	European call	7.1501	<-- =binomial_eur_call(B11,B12,B13,B6,B7,B15)

Notice that we have compromised on the number of periods, by using Excel's **Round** function—since there are 25 divisions of one year and the option's maturity is T = 0.75, the actual number of periods to maturity is 25*0.75, which is not a round number.

This procedure works well even for a very large number of periods. In the example below, the option has a maturity $T = 0.5$, and the basic one-year period is divided into 400 subperiods. Excel easily computes the value of the American put and call, even though a considerable amount of computation is involved:

	A	B	C
1	**VBA FUNCTIONS FOR CALLS AND PUTS** **n divisions per year, Δt = 1/n** **Up=exp(μ*Δt + σ*sqrt(Δt)), Down = exp(μ*Δt - σ*sqrt(Δt))**		
2	Mean return per year, μ	15%	
3	Standard deviation of annual return, σ	35%	
4	Annual interest rate, r	6%	
5			
6	Initial stock price, S	50.00	
7	Option exercise price, X	50.00	
8	Option exercise date (years)	0.50	
9	Number of divisions of 1 year	400	<-- each year divided into 400 subperiods
10	Δt, the length of one division	0.0025	<-- =1/B9
11	Up move per Δt	1.018036	<-- =EXP(B2*B10+B3*SQRT(B10))
12	Down move per Δt	0.983021	<-- =EXP(B2*B10-B3*SQRT(B10))
13	Interest rate per Δt	1.00015	<-- =EXP(B4*B10)
14			
15	Number of periods until maturity, n	200	<-- =ROUND(B8*B9,0)
16			
17	American put	4.2882	<-- =binomial_amer_put(B11,B12,B13,B6,B7,B15)
18	European put	4.1471	<-- =binomial_eur_put(B11,B12,B13,B6,B7,B15)
19	American call	5.6248	<-- =binomial_amer_call(B11,B12,B13,B6,B7,B15)
20	European call	5.6248	<-- =binomial_eur_call(B11,B12,B13,B6,B7,B15)

16.7 Convergence of Binomial Pricing to the Black-Scholes Price

In this section we discuss the convergence of the binomial model to the Black-Scholes pricing formula. The discussion assumes some understanding of log-normality (discussed in Chapter 26) and the Black-Scholes option pricing formula discussed in Chapter 17. So you may want to skip this section and come back to it later.

Whenever we consider a finite approximation to the option pricing formulas, we have to use an approximation to the up and the down movements. One widespread translation of the interest rate r and the stock's volatility σ to an "up" or "down" move necessary for the binomial model is:

$$\Delta t = T / n \qquad R = e^{r\Delta t}$$

$$U = 1 + up = e^{\sigma\sqrt{\Delta t}} \qquad D = 1 + down = e^{-\sigma\sqrt{\Delta t}}$$

This approximation guarantees that as $\Delta t \to 0$ (i.e., as $n \to \infty$), the resulting distribution of the stock returns approaches the lognormal distribution.[4]

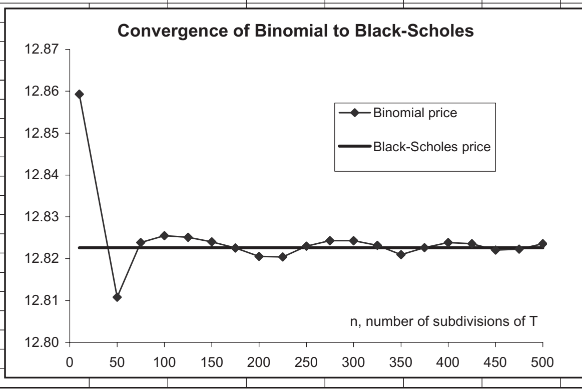
Here's an implementation of this methodology in a spreadsheet. The function **Binomial_Eur_call** is the same as that defined above; the function **BSCall** is the Black-Scholes formula and is defined and discussed in Chapter 17:

	A	B	C
1	**BLACK-SCHOLES AND BINOMIAL PRICING**		
2	S	60	Current stock price
3	X	50	Option exercise price
4	T	0.5000	Time to option exercise (in years)
5	r	8%	Annual interest rate
6	Sigma	30%	Riskiness of stock
7	n	20	Number of subdivisions of T
8			
9	Δt = T/n	0.0250	<-- =B4/B7
10	Up, U	1.0486	<-- =EXP(B6*SQRT(B9))
11	Down, D	0.9537	<-- =EXP(-B6*SQRT(B9))
12	Interest rate, R	1.0020	<-- =EXP(B5*B9)
13			
14	Binomial European call	12.8055	<-- =binomial_eur_call(B10,B11,B12,B2,B3,B7)
15	Black-Scholes call	12.8226	<-- =BSCall(B2,B3,B4,B5,B6)

The binomial model gives a good approximation to the Black-Scholes (cells B14:B15). As the n gets larger, this approximation gets better, though the convergence to the Black-Scholes price is not smooth:

4. An alternative approximation which converges to a lognormal price process is given in the next subsection. See also Omberg (1987), Hull (2006), and Benninga, Steinmetz, and Stroughair (1993).

	A	B	C	D	E	F	G	H	I	J	K
1	**BLACK-SCHOLES AND BINOMIAL PRICING: CONVERGENCE**										
2	S	60	Current stock price								
3	X	50	Option exercise price								
4	T	0.5000	Time to option exercise (in years)								
5	r	8%	Annual interest rate								
6	Sigma	30%	Riskiness of stock								
7	n	20	Number of subdivisions of T								
8											
9	Δt = T/n	0.0250	<-- =B4/B7								
10	Up, U	1.0486	<-- =EXP(B6*SQRT(B9))								
11	Down, D	0.9537	<-- =EXP(-B6*SQRT(B9))								
12	Interest rate, R	1.0020	<-- =EXP(B5*B9)								
13											
14	Binomial European call	12.8055	<-- =binomial_eur_call(B10,B11,B12,B2,B3,B7)								
15	Black-Scholes call	12.8226	<-- =BSCall(B2,B3,B4,B5,B6)								
16											
17	**Data table: Binomial price vs Black-Scholes**										
18	**n, number of subdivisions of T**	**Binomial price**	**Black-Scholes price**								
19		12.8055	12.8226	<-- Data table headers							
20	10	12.8593	12.8226								
21	50	12.8108	12.8226								
22	75	12.8238	12.8226								
23	100	12.8255	12.8226								
24	125	12.8251	12.8226								
25	150	12.8240	12.8226								
26	175	12.8226	12.8226								
27	200	12.8205	12.8226								
28	225	12.8204	12.8226								
29	250	12.8230	12.8226								
30	275	12.8243	12.8226								
31	300	12.8243	12.8226								
32	325	12.8232	12.8226								
33	350	12.8210	12.8226								
34	375	12.8226	12.8226								
35	400	12.8238	12.8226								
36	425	12.8236	12.8226								
37	450	12.8221	12.8226								
38	475	12.8223	12.8226								
39	500	12.8236	12.8226								



An Alternative Approximation to the Lognormal

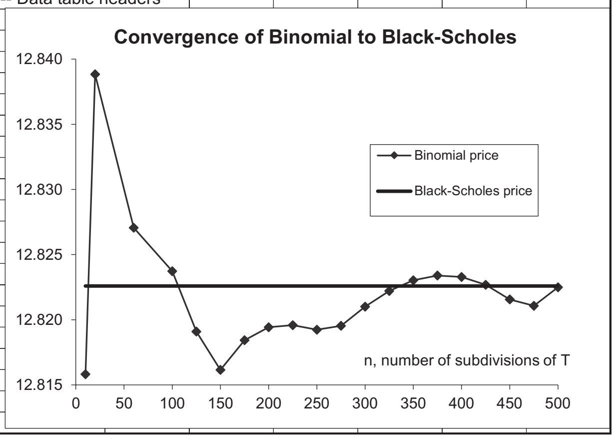
The approximation of the first part of this section is not the only approximation which works. If the stock price is lognormally distributed with mean μ and standard deviation σ, we can also use the following approximation:

$$\Delta t = T/n \qquad R = e^{r\Delta t}$$

$$U = 1 + up = e^{\mu\Delta t + \sigma\sqrt{\Delta t}} \qquad D = 1 + down = e^{\mu\Delta t - \sigma\sqrt{\Delta t}}$$

Implementing this in our spreadsheet gives:

	A	B	C	D	E	F	G	H	I	J
1	**BLACK-SCHOLES AND BINOMIAL PRICING: U=exp(μΔt + σ*sqrt(Δt)), D=exp(μΔt - σ*sqrt(Δt))**									
2	S	60	Current stock price							
3	X	50	Option exercise price							
4	T	0.5000	Time to option exercise (in years)							
5	r	8%	Annual interest rate							
6	Mean return, μ	12%								
7	Sigma, σ	30%	Riskiness of stock							
8	n	20	Number of subdivisions of T							
9										
10	Δt = T/n	0.0250	<-- =B4/B8							
11	Up, U	1.0517	<-- =EXP(B6*B10+B7*SQRT(B10))							
12	Down, D	0.9565	<-- =EXP(B6*B10-B7*SQRT(B10))							
13	Interest rate, R	1.0020	<-- =EXP(B5*B10)							
14										
15	Binomial European call	12.8388	<-- =binomial_eur_call(B11,B12,B13,B2,B3,B8)							
16	Black-Scholes call	12.8226	<-- =BSCall(B2,B3,B4,B5,B7)							
17										
18	**Data table: Binomial price vs Black-Scholes**									
19	**n, number of subdivisions of T**	**Binomial price**	**Black-Scholes price**							
20		12.8388	12.8226	<-- Data table headers						
21	10	12.8158	12.8226							
22	20	12.8388	12.8226							
23	60	12.8271	12.8226							
24	100	12.8237	12.8226							
25	125	12.8191	12.8226							
26	150	12.8162	12.8226							
27	175	12.8184	12.8226							
28	200	12.8194	12.8226							
29	225	12.8196	12.8226							
30	250	12.8192	12.8226							
31	275	12.8195	12.8226							
32	300	12.8210	12.8226							
33	325	12.8222	12.8226							
34	350	12.8230	12.8226							
35	375	12.8234	12.8226							
36	400	12.8233	12.8226							
37	425	12.8227	12.8226							
38	450	12.8216	12.8226							
39	475	12.8211	12.8226							
40	500	12.8225	12.8226							



The convergence of this parameterization to Black-Scholes is somewhat less smooth, though the ultimate result is the same.[5]

5. Note that both methods actually converge quite quickly—within several dozen steps the binomial is within 0.01 of the Black-Scholes price.

16.8 Using the Binomial Model to Price Employee Stock Options[6]

An employee stock option (ESO) is a call option given by a company to its employees as part of their remuneration package. Like all call options, the value of an ESO depends on the current price of the stock, the option's exercise price, and the time until exercise. However, ESOs typically have several special conditions:

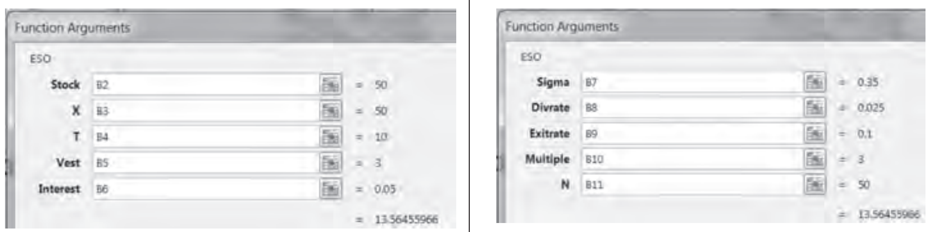
- The option has a vesting period. During this period, the employee is not allowed to exercise the option. An employee leaving the company before the vesting period forfeits his option. In the model of this section we assume that a typical employee of the company leaves at exit rate *e* per year.
- An employee leaving the company after the vesting period is forced to immediately exercise his option.
- For tax reasons, almost all ESOs have exercise prices equal to the stock price on the date of issue.

In the model below, adapted from a paper by Hull and White (2004), we assume that an employee will choose to exercise her option when the price of the stock is greater than some multiple *m* of the ESO's exercise price X. We first present the model's implementation and results and then discuss the VBA program, which gives us these results:

	A	B	C
1	**A BINOMIAL EMPLOYEE STOCK OPTION PRICING MODEL** **Based on Hull-White (2004)**		
2	S	50	Current stock price
3	X	50	Option exercise price
4	T	10.00	Time to option exercise (in years)
5	Vesting period (years)	3.00	
6	Interest	5.00%	Annual interest rate
7	Sigma	35%	Riskiness of stock
8	Stock dividend rate	2.50%	Annual dividend rate on stock
9	Exit rate, e	10.00%	
10	Option exercise multiple, m	3.00	
11	n	50	Number of subdivisions of one year
12			
13	Employee stock option value	13.56	<-- =ESO(B2,B3,B4,B5,B6,B7,B8,B9,B10,B11)
14	Black-Scholes call	19.18	<-- =BSCall(B2*EXP(-B8*B4),B3,B4,B6,B7)

6. This section has benefited from discussions with Torben Voetmann of Brattle Group and Zvi Wiener of Hebrew University, Jerusalem.

The **ESO** function in cell B13 depends on the 10 variables listed in cells B2:B11. The screen for this function looks like:



In the example above, the employee stock option is given when the stock price is \$50. The ESO has exercise price X = \$50. The option has a 10-year maturity and a 3-year vesting period. The interest rate is 5% annually, and the stock pays an annual dividend of 2.5% of its stock value. The rate at which employees leave the company is 10% per year. The model assumes that after the vesting period, the employee will choose to exercise his option if the stock price is three times or more the option's exercise price.[7] The binomial model with which the computations in cell B13 were done divides each year into 50 subdivisions.

Given these assumptions, the employee stock option is valued at \$13.56 (cell B13). A comparable Black-Scholes option on a dividend-paying stock would be valued at \$19.18.[8]

The ESO Valuation and FASB 123

The American Financial Accounting Standards Board (FASB) and the International Accounting Standards Board (IASB) agree that executive stock options should be priced using a model of the type explored above, and that the value of options awarded should be accounted for in a firm's net income. If, for example, a firm had issued 1 million options of the type which appear in the previous spreadsheet, we would value these options at \$13,564,600.

7. Research cited by Hull and White (2004) shows that the average stock-price-to-exercise-price ratio at which ESO owners exercise their options is between 2.2 and 2.8.

8. We're getting way ahead of ourselves here! The adaptation of Black-Scholes for dividend-paying stock is given in section 17.6.

The VBA Code for the ESO Model

Below we give the VBA code for this model. A short discussion follows the code.

```
Function ESO(Stock As Double, X As Double, T As _
Double, Vest As Double, Interest As Double, _
Sigma As Double, Divrate As Double, _
Exitrate As Double, Multiple As Double, _
n As Single)

    Dim Up As Double, Down As Double, _
    R As Double, Div As Double, _
    piUp As Double, piDown As Double, _
    Delta As Double, i As Integer, j As Integer

   ReDim Opt(T * n, T * n)
   ReDim S(T * n, T * n)
   Up = Exp(Sigma * Sqr(1 / n))
   Down = Exp(-Sigma * Sqr(1 / n))
   R = Exp(Interest / n)
   Div = Exp(-Divrate / n)
   'Risk-neutral Up and Down probabilities
   piUp = (R * Div - Down) / (Up - Down)
   piDown = (Up - R * Div) / (Up - Down)

   'Defining the stock price
   'j is the number of Up steps
    For i = 0 To T * n
        For j = 0 To i
        S(i, j) = Stock * Up ^ j _
        * Down ^ (i - j)
        Next j
    Next i
```

```
    'Option value on the last nodes of tree
    For i = 0 To T * n
       Opt(T * n, i) = _
       Application.Max(S(T * n, i) - X, 0)
    Next i

    'Early exercise when stock price > multiple
    ' * exercise after vesting
    For i = T * n - 1 To 0 Step -1
    For j = 0 To i
    If i > Vest * n And S(i, j) >= Multiple * X _
    Then Opt(i, j) = Application.Max(S(i, j) - X, 0)
    If i > Vest * n And S(i, j) < Multiple * X _
    Then Opt(i, j) = ((1 - Exitrate / n) * _
    (piUp * Opt(i + 1, j + 1) + piDown * _
    Opt(i + 1, j)) / R + Exitrate / n * _
    Application.Max(S(i, j) - X, 0))
    If i <= Vest * n Then Opt(i, j) = _
    (1 - Exitrate / n) * (piUp * _
    Opt(i + 1, j + 1) + piDown * Opt(i + 1, j)) / R

    Next j
    Next i

    ESO = Opt(0, 0)
End Function
```

Explaining the VBA code[9]

The VBA code has several parts. The first part defines the variables, adjusting the Up, Down, and 1 plus interest R for the n divisions of each year. Having made this adjustment, the code defines the risk-neutral probabilities π_{Up} and π_{Down}:

```
Up = Exp(Sigma * Sqr(1 / n))
Down = Exp(-Sigma * Sqr(1 / n))
R = Exp(Interest / n)
Div = Exp(-Divrate / n)
'Risk-neutral Up and Down probabilities
piUp = (R * Div - Down) / (Up - Down)
piDown = (Up - R * Div) / (Up - Down)
```

The stock price is defined as an array $S(i, j)$, where i defines the periods, $i = 0, 1, \ldots, T*n$, and j defines the number of Up steps at each period, $j = 0, 1, \ldots, i$. The next part of the code defines the stock price.

```
'Defining the stock price
  'j is the number of Up steps
   For i = 0 To T * n
       For j = 0 To i
       S(i, j) = Stock * Up ^ j _
       * Down ^ (i - j)
       Next j
   Next i
```

9. This subsection is tedious and can be skipped. But take a look at the next subsection, where we use **Data|Table** to do sensitivity analysis.

The option values are defined in the next piece of code, which is the heart of our employee stock option function. Option value is defined as an array Opt(i, j):

```
'Option value on the last nodes of tree
For i = 0 To T * n
   Opt(T * n, i) = _
   Application.Max(S(T * n, i) - X, 0)
Next i

'Early exercise when stock price > multiple
' * exercise after vesting
For i = T * n - 1 To 0 Step -1
For j = 0 To i
If i > Vest * n And S(i, j) >= Multiple * X _
Then Opt(i, j) = Application.Max(S(i, j) - X, 0)
If i > Vest * n And S(i, j) < Multiple * X _
Then Opt(i, j) = ((1 - Exitrate / n) * _
(piUp * Opt(i + 1, j + 1) + piDown * _
Opt(i + 1, j)) / R + Exitrate / n * _
Application.Max(S(i, j) - X, 0))
If i <= Vest * n Then Opt(i, j) = _
(1 - Exitrate / n) * (piUp * _
Opt(i + 1, j + 1) + piDown * Opt(i + 1, j)) / R

Next j
Next i
```

Here's what this piece of code says:

$$Opt(i,j)=\begin{cases}\max[S(T*n,j)-X,0] & \textit{Terminal nodes}\\ \max[S(i,j)-X,0] & \textit{After vesting}, S(i,j)\geq m*X\\ (1-Exitrate/n)*\dfrac{\pi_{Up}Opt(i+1,j+1)+\pi_{Down}Opt(i+1,j)}{R} + Exitrate/n*\max(S(i,j)-X,0) & \textit{After vesting}, S(i,j)<m*X\\ (1-Exitrate/n)*\dfrac{\pi_{Up}Opt(i+1,j+1)+\pi_{Down}Opt(i+1,j)}{R} & \textit{Before vesting}\end{cases}$$

At the terminal nodes, we simply exercise the option. Before the terminal nodes and after vesting, we check to see if the stock price is greater than the desired multiple m of the exercise price. If it is, we exercise the option. If $S(i, j) < m*X$, then the ESO's payoff depends on whether the employee exits the firm or not. With a probability $(1 - Exitrate / n)$ the employee does not exit the firm, in which case the option payoff is the discounted expected next period payoff:

$$(1-Exitrate/n)*\frac{\pi_{Up}Opt(i+1,j+1)+\pi_{Down}Opt(i+1,j)}{R}$$

On the other hand, if the employee exits the firm and the vesting period has passed, he will try to see if he can exercise the option, giving the expected payoff:

$$Exitrate/n*\max[S(i,j)-X,0]$$

Finally, before vesting, the ESO is simply worth the expected discounted (by the risk-neutral probabilities) payoff of the next period values:

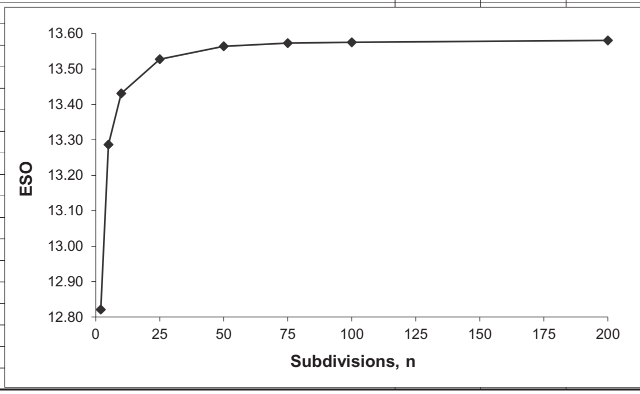
$$(1-Exitrate/n)*\frac{\pi_{Up}Opt(i+1,j+1)+\pi_{Down}Opt(i+1,j)}{R}$$

The final step in the code is to define the value of the function ESO: $ESO = Opt(0, 0)$.

Some Sensitivity Analysis

We can use data tables to perform sensitivity analysis on our **ESO** function.

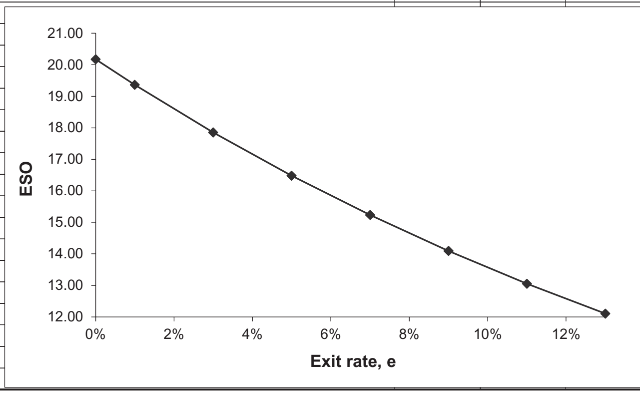
	A	B	C	D	E	F
1	**ESO FUNCTION SENSITIVITY TO NUMBER OF SUBDIVISIONS *n* OF ONE YEAR**					
2	S	50	Current stock price			
3	X	50	Option exercise price			
4	T	10.0000	Time to option exercise (in years)			
5	Vesting period (years)	3.00				
6	Interest	5.00%	Annual interest rate			
7	Sigma	35%	Riskiness of stock			
8	Stock dividend rate	2.50%	Annual dividend rate on stock			
9	Exit rate, e	10.00%				
10	Option exercise multiple, m	3.00				
11	n	25	Number of subdivisions of one year			
12						
13	Employee stock option value	13.5275	<-- =ESO(B2,B3,B4,B5,B6,B7,B8,B9,B10,B11)			
14	Black-Scholes call	19.1842	<-- =BSCall(B2*EXP(-B8*B4),B3,B4,B6,B7)			
15						
16	**Sensitivity of ESO value to number of subdivisions *n***					
17	n	13.5275	<-- =B13, data table header			
18	2	12.8213				
19	5	13.2870				
20	10	13.4312				
21	25	13.5275				
22	50	13.5646				
23	75	13.5733				
24	100	13.5753				
25	200	13.5810				
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						



The graph gives ample evidence that $n = 25$ or 50 and does well enough for valuing ESOs. Since larger values of n become time consuming, we recommend lower numbers.

In the graph below, we show the sensitivity of the ESO value to the employee exit rate e, the rate at which employees leave the firm each year:

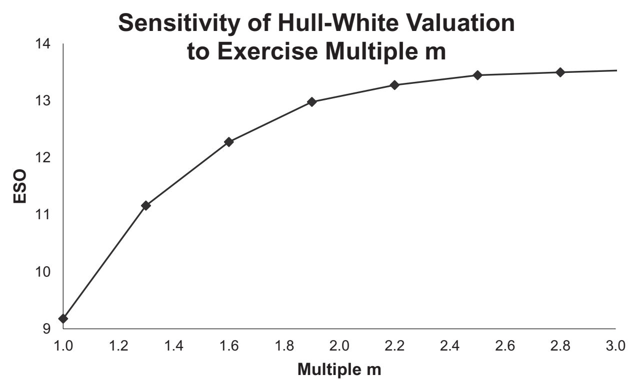
	A	B	C	D	E	F
1	**ESO FUNCTION SENSITIVITY TO EMPLOYEE EXIT RATE *e***					
2	S	50	Current stock price			
3	X	50	Option exercise price			
4	T	10.0000	Time to option exercise (in years)			
5	Vesting period (years)	3.00				
6	Interest	5.00%	Annual interest rate			
7	Sigma	35%	Riskiness of stock			
8	Stock dividend rate	2.50%	Annual dividend rate on stock			
9	Exit rate, e	10.00%				
10	Option exercise multiple, m	3.00				
11	n	50	Number of subdivisions of one year			
12						
13	Employee stock option value	13.5646	<-- =ESO(B2,B3,B4,B5,B6,B7,B8,B9,B10,B11)			
14	Black-Scholes call	19.1842	<-- =BSCall(B2*EXP(-B8*B4),B3,B4,B6,B7)			
15						
16	**Sensitivity of ESO value to exit rate e**					
17	Exit rate, e	13.5646	<-- =B13, data table header			
18	0%	20.1732				
19	1%	19.3621				
20	3%	17.8536				
21	5%	16.4828				
22	7%	15.2347				
23	9%	14.0963				
24	11%	13.0561				
25	13%	12.1039				
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						



The exit rate has a major effect on the value of the ESO: the higher the turnover of employees, the lower the value of the employee stock options. In terms of FASB 123 valuation, the exit rate *e* is an important valuation factor.

Finally we do a sensitivity of the ESO value on the exit multiple *m*. Recall that the Hull-White model assumes that an employee holding an ESO exercises her option when the stock price is a multiple *m* of the option exercise price *X*. Basically this locks the employee into a suboptimal strategy, since in general call options should be held to maturity (though note that in this case the option is written on a stock which pays a dividend, which may in some cases make early exercise optimal). In the example below, we clearly see the suboptimality of early ESO exercise: the higher the multiple *m*, the higher the value of the ESO.

	A	B	C	D	E	F
1	**ESO FUNCTION SENSITIVITY TO OPTION EXERCISE MULTIPLE *m***					
2	S	50	Current stock price			
3	X	50	Option exercise price			
4	T	10.0000	Time to option exercise (in years)			
5	Vesting period (years)	3.00				
6	Interest	5.00%	Annual interest rate			
7	Sigma	35%	Riskiness of stock			
8	Stock dividend rate	2.50%	Annual dividend rate on stock			
9	Exit rate, e	10.00%				
10	Option exercise multiple, m	3.00				
11	n	25	Number of subdivisions of one year			
12						
13	Employee stock option value	13.5275	<-- =ESO(B2,B3,B4,B5,B6,B7,B8,B9,B10,B11)			
14	Black-Scholes call	19.1842	<-- =BSCall(B2*EXP(-B8*B4),B3,B4,B6,B7)			
15						
16	**Sensitivity of ESO value to multiple m**					
17	m	13.5275	<-- =B13, data table header			
18	1.0	9.1758				
19	1.3	11.1610				
20	1.6	12.2760				
21	1.9	12.9793				
22	2.2	13.2735				
23	2.5	13.4467				
24	2.8	13.4985				
25	3.1	13.5423				
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						



Last But Not Least

The Hull-White model is a numerical approximation of the ESO option valuation, but it is not a closed-form formula. A recent paper by Cvitanić, Wiener, and Zapatero (2006) gives an analytical derivation of the value of employee stock options. The formula stretches over 16 pages of typescript and will not be given here. An Excel implementation of the formula exists and can be downloaded at http://pluto.mscc.huji.ac.il/~mswiener/research/ESO.htm.

16.9 Using the Binomial Model to Price Non-Standard Options: An Example

The binomial model can also be used to price non-standard options. Consider the following example: You hold an option to buy a share of a company. The option allows for early exercise, but the exercise price varies with the time at which you choose to exercise. For the case we consider, the option has the following conditions:

- There are *n* possible exercise dates *only* (i.e., the option is only exercisable on these dates).
- Exercise at date *t* precludes exercise at all dates $s > t$. However, if you don't exercise at date *s*, you may still exercise at date $t > s$.
- The exercise price at date *t* is X_t, meaning that the exercise price can vary with time.

We want to value this option using a binomial framework. To do so, we recognize that basically this is just an American option with three separate exercise prices.

Here's how we set this problem up in a spreadsheet, using the logic of the American option valuation described in section 16.5:

	A	B	C	D	E	F	G	H	I
1	**TIME-DEPENDENT EXERCISE PRICES**								
2	Initial stock price	100		**Exercise prices**					
3	Up	10%		Date 1	100				
4	Down	-5%		Date 2	105				
5	Interest rate	6%		Date 3	112				
6									
7	**State prices**								
8	q_u	0.6918	<-- =(B5-B4)/((1+B5)*(B3-B4))						
9	q_d	0.2516	<-- =(B3-B5)/((1+B5)*(B3-B4))						
10									
11	**Check**								
12	$1/(q_u+q_d)$	1.06	<-- =1/(B8+B9)						
13	q_u*(1+up)+q_d*(1+down)	1	<-- =B8*(1+B3)+B9*(1+B4)						
14									
15		**Stock price**						133.100	
16						121.000			
17				110.000				114.950	
18		100.000				104.500			
19				95.000				99.275	
20						90.250			
21								85.738	
22									=MAX(q_U*H25+q_D*H27,MAX(F16-E4,0))
23		**Date 0**		**Date 1**		**Date 2**		**Date 3**	
24									
25		**Value at each node**						21.100	<-- =MAX(H16-E6,0)
26						16.000			
27				11.583				2.950	<-- =MAX(H18-E6,0)
28		8.368				2.041			
29				1.412				0.000	
30						0.000			
31	=MAX(q_U*F28+q_D*F30,MAX(D19-E3,0))							0.000	
32									
33		**Early exercise?**							
34						yes			
35				no				=IF(q_U*H25+q_D*H27>= MAX(F16-E4,0),"no","yes")	
36						no			
37				no					
38						no			
39									

Most of this spreadsheet follows section 16.5. Cells B15:H21 describe the stock price over time, which follows a binomial process with the "Up" = 1.10 and "Down" = 0.95 (cells B3 and B4). Where things get interesting is in the valuation:

	A	B	C	D	E	F	G	H	I
22									=MAX(q_U*H25+q_D*H27,MAX(F16-E4,0))
23		**Date 0**		**Date 1**		**Date 2**		**Date 3**	
24									
25		**Value at each node**						21.100	<-- =MAX(H16-E6,0)
26						16.000			
27				11.583				2.950	<-- =MAX(H18-E6,0)
28		8.368				2.041			
29				1.412				0.000	
30						0.000			
31	=MAX(q_U*F28+q_D*F30,MAX(D19-E3,0))							0.000	
32									

As is usual for an American option, at each node of the tree, we consider whether the option is worth more whether exercised or whether held. But note that in the above picture, the exercise price varies with the date, so that the exercise price at date 3 is E5, that of date 2 is E4, and that of date 1 is E3.

As you can see in cell B28, the value of the American call option is 8.368.

16.10 Summary

The binomial model is intuitive and easy to implement. As a widespread alternative to Black-Scholes pricing, the model can easily be put into a spreadsheet and programmed in VBA. This chapter has explored both the basic uses of the binomial model and its implementation to price American and other non-standard options. A section on employee stock options has shown how to implement the Hull-White (2004) model for valuing these options. Throughout, we have laid special stress on the role of state prices in implementing the model.

Exercises

1. A stock selling for $25 today will, in 1 year, be worth either $35 or $20. If the interest rate is 8%, what is the value today of a one-year call option on the stock with exercise price $30? Use the simultaneous equation approach of section 16.2 to price the option.

2. In exercise 1, compute the state prices q_U and q_D, and use these prices to calculate the value today of a one-year put option on the stock with exercise price $30. Show that

put-call parity holds: That is, using your answer from this problem and the previous problem, show that:

$$Call\ price + \frac{X}{1+r} = Stock\ price\ today + Put\ price$$

3. In a binomial model a call option and a put option are both written on the same stock. The exercise price of the call option is 30 and the exercise price of the put option is 40. The call option's payoffs are 0 and 5 and the put option's payoffs are 20 and 5. The price of the call is 2.25 and the price of the put is 12.25.

   a. What is the riskless interest rate? Assume that the basic period is one year.

   b. What is the price of the stock today?

4. All reliable analysts agree that a share of ABC Corp., selling today for \$50, will be priced at either \$65 or \$45 1 year from now. They further agree that the probabilities of these events are 0.6 and 0.4, respectively. The market risk-free rate is 6%. What is the value of a call option on ABC whose exercise price is \$50 and which matures in 1 year?

5. A stock is currently selling for 60. The price of the stock at the end of the year is expected either to increase by 25% or to decrease by 20%. The riskless interest rate is 5%. Calculate the price of a European put on the stock with exercise price 55. Use the binomial option pricing model.

6. Fill in all the cells labeled ??? in the following spreadsheet: Why is there no additional pricing tree for an American call option?

	A	B	C	D	E	F	G	H	I	J	K
1	**THREE-DATE BINOMIAL OPTION PRICING**										
2	Up, U	1.35									
3	Down, D	0.95		state prices							
4				q_u							
5	Initial stock price	40		q_d							
6	Interest rate, R	1.25									
7	Exercise price	40									
8											
9	**Stock price**						**Bond price**				
10					???						???
11			???						???		
12		40			???		1				???
13			???						???		
14					???						???
15											
16	**European call option**						**European put option**				
17											
18					???						???
19			???						???		
20	???				???		???				???
21			???						???		
22					???						???
23											
24							**American put option**				
25											
26											???
27									???		
28							???				???
29									???		
30											???

7. Consider the following 2-period binomial model, in which the annual interest rate is 9% and in which the stock price goes up by 15% per period or down by 10%:

a. Price a European call on the stock with exercise price 60.

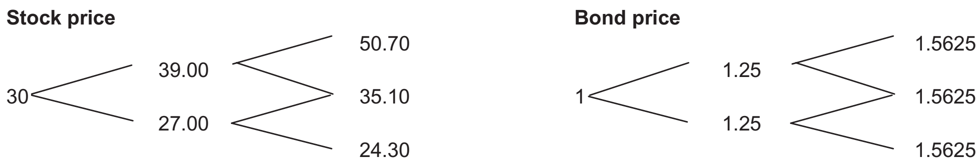
b. Price a European put on the stock with exercise price 60.

c. Price an American call on the stock with exercise price 60.

d. Price an American put on the stock with exercise price 60.

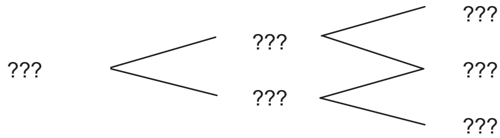
8. Consider the following three-date binomial model:

- In each period the stock price either goes up by 30% or decreases by 10%.
- The one-period interest rate is 25%



a. Consider a European call with $X = 30$ and $T = 2$. Fill in the blanks in the tree:

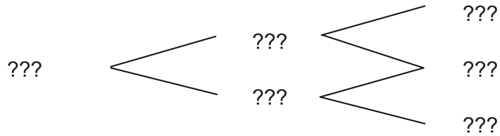
Call option price



b. Price a European put with $X = 30$ and $T = 2$.

c. Now consider an American put with $X = 30$ and $T = 2$. Fill in the blanks in the tree:

American put option price



9. A prominent securities firm recently introduced a new financial product. This product, called "The Best of Both Worlds" (BOBOW for short), costs $10. It matures in 5 years, at which point it repays the investor the $10 cost *plus* 120% of any positive return in the S&P 500 index. There are no payments before maturity.

For example: If the S&P 500 is currently at 1,500, and if it is at 1,800 in 5 years, a BOBOW owner will receive back $12.40 = $10*[1 + 1.2*(1800/1500-1)]. If the S&P is at or below 1,500 in 5 years, the BOBOW owner will receive back $10.

Suppose that the annual interest rate on a 5-year, continuously compounded, pure-discount bond is 6%. Suppose further that the S&P 500 is currently at 1,500 and that you believe that in 5 years it will be at either 2,500 or 1,200. Use the binomial option pricing model to show that BOBOWs are underpriced.

10. This problem is a continuation of the discussion of section 16.6. Show that as $n \to \infty$, the binomial European put price converges to the Black-Scholes put price. (Note that, as part of the chapter spreadsheet for this chapter, we have included a function called **BSPut**, which computes the Black-Scholes put price.)

11. Here's an advanced version of exercise 10. Consider an alternative parameterization of the binomial:

$$\Delta t = T / n \qquad R = e^{r\Delta t}$$

$$\text{Up} = e^{(r-\sigma^2/2)\Delta t+\sigma\sqrt{\Delta t}} \qquad q_U = \frac{R - \text{Down}}{R*(\text{Up} - \text{Down})}$$

$$\text{Down} = e^{(r-\sigma^2/2)\Delta t-\sigma\sqrt{\Delta t}} \qquad q_D = \frac{1}{R} - q_U$$

Construct binomial European call and put option pricing functions in VBA for this parameterization and show that they also converge to the Black-Scholes formula. (The message here is that the parameterization of the binomial Up and Down is not unique.)

12. A call option is written on a stock whose current price is $50. The option has maturity of 3 years, and during this time the annual stock price is expected to increase by 25% or to decrease by −10%. The annual interest rate is constant at 6%. The option is exercisable at date 1 at a price of $55, at date 2 for a price of $60, and at date 3 for a price of $65. What is its value today? Will you ever exercise the option early?

13. Reconsider the above problem. Show that if the date 1 exercise price is X, the date 2 exercise price is $X^*(1 + r)$, and the date 3 exercise price is $X^*(1 + r)^2$, you will not exercise the option early.[10]

14. An investment bank is offering a security linked to the price 2 years from today of Bisco stock, which is currently at $3 per share. Denote Bisco's stock price in two periods by S_2. The security being offered pays off $\max(S_2^3 - 40, 0)$. You estimate that in each of the next two periods, Bisco stock will either increase by 50% or decrease by 20%. The annual interest rate is 8%. Price the security.

10. It can also be shown that this property holds if the exercise prices grow more slowly than the interest rate. Thus for the problem considered in section 16.5, there will be early exercise of the American call only when the exercise prices grow at a rate faster than the interest rate.

17 The Black-Scholes Model

17.1 Overview

In a path-breaking paper published in 1973, Fischer Black and Myron Scholes proved a formula for pricing European call and put options on non-dividend-paying stocks. Their model is probably the most famous model of modern finance. The Black-Scholes formula is relatively easy to use, and it is often an adequate approximation to the price of more complicated options. In this chapter we make no pretense at a full-blown development of the model; this requires a knowledge of stochastic processes, and a not-inconsiderable mathematical investment. Instead, we shall describe the mechanics of the model and show how to implement it in Excel. We also illustrate several uses of the Black-Scholes formula in the valuation of structured assets.

17.2 The Black-Scholes Model

Consider a stock whose price is lognormally distributed.[1] The Black-Scholes model uses the following formula to price European calls on a stock:

$$C = SN(d_1) - Xe^{-rT}N(d_2)$$

where

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

Here C denotes the price of a call, S is the price of the underlying stock, X is the exercise price of the call, T is the call's time to exercise, r is the interest rate, and σ is the standard deviation of the logarithm of the stock's return. $N(\ )$ denotes a value of the standard normal distribution. It is assumed that the stock will pay no dividends before date T.

By the put-call parity theorem (see Chapter 15), a put with the same exercise date T and exercise price X written on the same stock will have price $P = C - S + Xe^{-rT}$. Substituting for C in this equation and doing some algebra gives the Black-Scholes European put pricing formula:

1. The lognormal distribution is discussed in Chapter 26 though for purposes of applying the Black-Scholes model, section 17.4 is sufficient.

$$P = Xe^{-rT} N(-d_2) - SN(-d_1)$$

In Chapter 16 we hinted at one form of the proof of the Black-Scholes formula. There it was shown numerically that the Black-Scholes formula coincides with the binomial option pricing model formula when (i) the length of a typical period → 0, (ii) the "Up" and the "Down" moves in the binomial model converge to a lognormal price process, and (iii) the term structure of interest rates is flat.

Implementing the Black-Scholes Formulas in a Spreadsheet

The Black-Scholes formulas for call and put pricing are easily implemented in a spreadsheet. The following example shows how to calculate the price of a call option written on a stock whose current price $S = 50$, when the exercise price $X = 45$, the annualized interest rate $r = 4\%$, and $\sigma = 30\%$. The option has $T = 0.75$ years to exercise. All three of the parameters T, r, and σ are assumed to be in annual terms.[2]

	A	B	C
1	**BLACK-SCHOLES OPTION PRICING FORMULA**		
2	S	50	Current stock price
3	X	45	Exercise price
4	r	4.00%	Risk-free rate of interest
5	T	0.75	Time to maturity of option (in years)
6	Sigma	30%	Stock volatility, σ
7			
8	d_1	0.6509	<-- (LN(S/X)+(r+0.5*sigma^2)*T)/(sigma*SQRT(T))
9	d_2	0.3911	<-- d_1-sigma*SQRT(T)
10			
11	N(d_1)	0.7424	<-- Uses formula NormSDist(d_1)
12	N(d_2)	0.6521	<-- Uses formula NormSDist(d_2)
13			
14	Call price	8.64	<-- S*N(d_1)-X*exp(-r*T)*N(d_2)
15	Put price	2.31	<-- call price - S + X*Exp(-r*T): by Put-Call parity
16		2.31	<-- X*exp(-r*T)*N(-d_2) - S*N(-d_1): direct formula

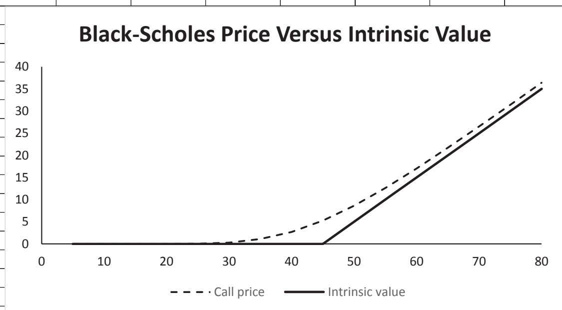
2. Section 26.7 in Chapter 26 discusses how to calculate the annualized σ of the lognormal process given non-annual data.

The spreadsheet calculates the put price twice: In cell B15 the put price is computed by using put-call parity, and in cell B16 it is calculated by using the direct Black-Scholes formula.

We can use this spreadsheet to do the usual sensitivity analysis. For example, the following **Data|Table** (see Chapter 31) gives—as the stock price *S* varies—the Black-Scholes value of the call compared to its intrinsic value [i.e., max(S – X, 0)].

	C	D	E	F
18		**Data table: Comparing the Black-Scholes to the intrinsic value**		
19		**Stock price**	**Call price**	**Intrinsic value**
20	Data table header: = B14		8.6434	5
21		5	0.0000	0
22		10	0.0000	0
23		15	0.0000	0
24		20	0.0029	0
25		25	0.0484	0
26		30	0.3101	0
27		35	1.1077	0
28		40	2.7319	0
29		45	5.2777	0
30		50	8.6434	5
31		55	12.6307	10
32		60	17.0378	15
33		65	21.7056	20
34		70	26.5256	25
35		75	31.4304	30
36		80	36.3811	35

Data table header: =Max(B2-B3,0). This is the option's *intrinsic value*.



17.3 Using VBA to Define a Black-Scholes Pricing Function

Although the spreadsheet implementation of the Black-Scholes formulas illustrated in the previous section is sufficient for some purposes, we are sometimes interested in having a closed form function which we can use directly in Excel. We can do so with Visual Basic for Applications. Below we define functions **dOne**, **dTwo**, and **BSCall**:

```
Function dOne(Stock, Exercise, Time, _
Interest, sigma)
   dOne = (Log(Stock / Exercise) + _
   Interest * Time) / (sigma * Sqr(Time)) _
      + 0.5 * sigma * Sqr(Time)
End Function

Function dTwo(Stock, Exercise, Time, _
Interest, sigma)
   dTwo = dOne(Stock, Exercise, Time, _
   Interest, sigma) - sigma * Sqr(Time)
End Function

Function BSCall(Stock, Exercise, Time, _
Interest, sigma)
   BSCall = Stock * Application.NormSDist _
   (dOne(Stock, Exercise, Time, Interest, _
   sigma)) - Exercise * Exp(-Time * Interest) * _
    Application.NormSDist(dTwo(Stock, Exercise, _
    Time, Interest, sigma))
End Function
```

Note the use of the Excel function **NormSDist**, which gives the standard normal distribution.[3]

3. We could have used the newer version of this function, **Norm.S.Dist(x,TRUE** or **FALSE)**. **TRUE** gives the cumulative distribution, whereas **FALSE** gives the probability density. It seems simpler to use the old version of this function, **NormSDist**, which gives the cumulative distribution.

Pricing Puts

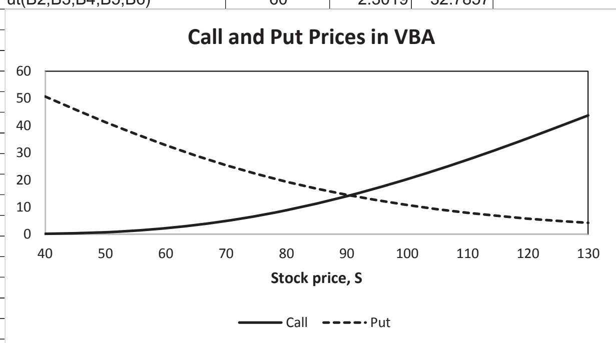
By the put-call parity theorem we know that a put is priced by the formula $P = C - S + Xe^{-rT}$. We can implement this in another VBA function, **BSPut**:

```
Function BSPut(Stock, Exercise, Time, _
Interest, sigma)
   BSPut = BSCall(Stock, Exercise, Time, _
   Interest, sigma) + Exercise * _
   Exp(-Interest * Time) - Stock
End Function
```

Using These Functions in an Excel Spreadsheet

Here's an example of these functions used in Excel. The graph was created by a data table (in presentations, we usually hide the first row of such a table; here we have shown it).

	A	B	C	D	E	F	G
1	BLACK-SCHOLES MODEL IN VBA						
2	S	100					=B9
3	X	100	=B8	Stock price	Call	Put	
4	T	1.00			20.3185	10.8022	<--This is the header of the Data Table
5	Interest	10.00%		40	0.1802	50.6639	
6	Sigma	40.00%		45	0.4104	45.8941	
7				50	0.8081	41.2918	
8	Call price	20.3185	<-- =BSCall(B2,B3,B4,B5,B6)	55	1.4241	36.9079	
9	Put price	10.8022	<-- =BSPut(B2,B3,B4,B5,B6)	60	2.3019	32.7857	
10							
11							
12	To the right is a data						
13	table that gives the						
14	call and put values for						
15	various stock						
16	prices.						
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							



17.4 Calculating the Volatility

The Black-Scholes formula depends on five parameters: the stock price S, the option exercise price X, the option's time to maturity T, the interest rate r, and the standard deviation of the returns of the stock underlying the option σ. Four of these five parameters are straightforward, but the fifth parameter, σ, is problematic. There are two common ways of computing sigma, σ:

- σ can be computed based on the *historical returns* of the stock.
- σ can be computed based on the *implied volatility* of the stock.

In the two subsections below, we illustrate both methods of computing σ and apply them to the pricing of options on the Standard & Poor's 500 (symbol SPY, called "spiders").

The Volatility of Historical Returns

We can use historical stock returns to compute the volatility. The method is the following:

- For a given time frame and frequency of returns, we compute the periodic volatility. The time frame commonly used varies wildly: Some practitioners use a short term of, say, 30 days, while others use a much longer (up to 1 year) time frame. Similarly, the frequency of returns can be daily, weekly, or sometimes monthly. Since options are mostly short term, a shorter time frame is more common.
- We annualize the periodic volatility by multiplying by the square root of the number of periods per year. Thus:

$$\sigma_{annual} = \begin{cases} \sqrt{12} * \sigma_{monthly} \\ \sqrt{52} * \sigma_{weekly} \\ \sqrt{250} * \sigma_{daily} \end{cases}$$

The number of days per year is an open question. Most practitioners use 250 or 252 for the number of transaction dates per year. However, instances of using 365 can also be found.

In the spreadsheet below we show one year's daily prices for SPDR S&P 500 (SPY). This is an exchange-traded fund (ETF) that tracks the Standard & Poor's 500. The historical prices of the SPY and the resulting historical volatility are computed below:

	A	B	C	D	E	F	G	H
1	**SPY HISTORICAL PRICES, DAILY DATA**							
2	**Date**	**Adj Close**	**Return**			**Return statistics, one year**		
3	10-Oct-11	117.07				Count	252	<-- =COUNT(C:C)
4	11-Oct-11	117.19	0.10%	<-- =LN(B4/B3)		Average daily return	0.08%	<-- =AVERAGE(C:C)
5	12-Oct-11	118.22	0.88%	<-- =LN(B5/B4)		Standard deviation of daily return	1.03%	<-- =STDEV.S(C:C)
6	13-Oct-11	117.98	-0.20%					
7	14-Oct-11	120	1.70%			Annualized mean return	0.99%	<-- =12*G4
8	17-Oct-11	117.71	-1.93%			Annualized sigma	16.34%	<-- =SQRT(252)*G5
9	18-Oct-11	120.01	1.94%					
10	19-Oct-11	118.59	-1.19%			**Return statistics, last half year**		
11	20-Oct-11	119.11	0.44%			Count	126	<-- =COUNT(C130:C255)
12	21-Oct-11	121.37	1.88%			Average daily return	0.05%	<-- =AVERAGE(C130:C255)
13	24-Oct-11	122.86	1.22%			Standard deviation of daily return	0.87%	<-- =STDEV.S(C130:C255)
14	25-Oct-11	120.47	-1.96%					
15	26-Oct-11	121.69	1.01%			Annualized mean return	0.59%	<-- =12*G12
16	27-Oct-11	125.93	3.42%			Annualized sigma	13.76%	<-- =SQRT(252)*G13
17	28-Oct-11	125.9	-0.02%					
18	31-Oct-11	122.87	-2.44%					
19	1-Nov-11	119.44	-2.83%					
20	2-Nov-11	121.39	1.62%					

The historical volatility based on a full year of data is 16.34%, whereas the volatility for the last 6 months is 13.76%.

The Implied Volatility

The implied volatility ignores history; instead it determines the option σ based on actual option prices. Whereas the historical volatility is a backward-looking volatility, the implied volatility is a forward-looking estimate.[4]

To estimate the implied volatility for the SPY calls expiring 19 January 2013, we solve the Black-Scholes formula for the sigma, which gives the current market price:

4. The nomenclature "forward-looking" versus "backward-looking" makes it sound as if the implied volatility is always better than the historical. This is, of course, not the intention.

	A	B	C
1	**IMPLIED VOLATILITY FOR THE JANUARY 2013 SPY OPTIONS**		
2	Current date	09-Oct-12	
3	Option expiration date	19-Jan-13	
4			
5	Current SPY price, S	144.2	
6	Option strike price, X	144	
7	Time to maturity, T	0.279452	<-- =(B3-B2)/365
8	Interest rate	0.08%	
9			
10	Actual call price	4.74	
11	Actual put price	4.91	
12			
13	Implied call volatility	15.22%	<-- =CallVolatility(B5,B6,B7,B8,B10)
14	Proof: Black-Scholes call price	4.74	<-- =BSCall(B5,B6,B7,B8,B13)
15			
16	Implied put volatility	16.54%	<-- =PutVolatility(B5,B6,B7,B8,B11)
17	Proof: Black-Scholes put price	4.91	<-- =BSPut(B5,B6,B7,B8,B16)

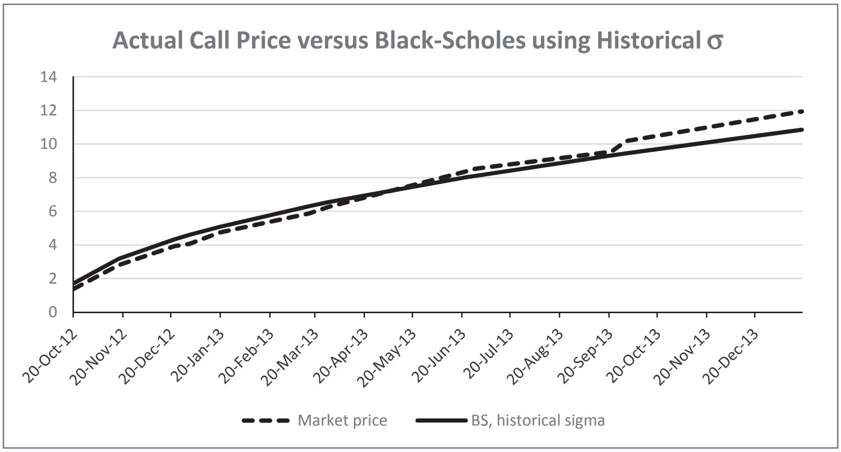
The implied volatility is 15.22% for the call and 16.54% for the put. As shown in cells B14 and B17, these volatilities, when inserted in the Black-Scholes formula, give back the current market price. We have used functions **CallVolatility** and **PutVolatility** described below.

Pricing At-the-Money SPY Options

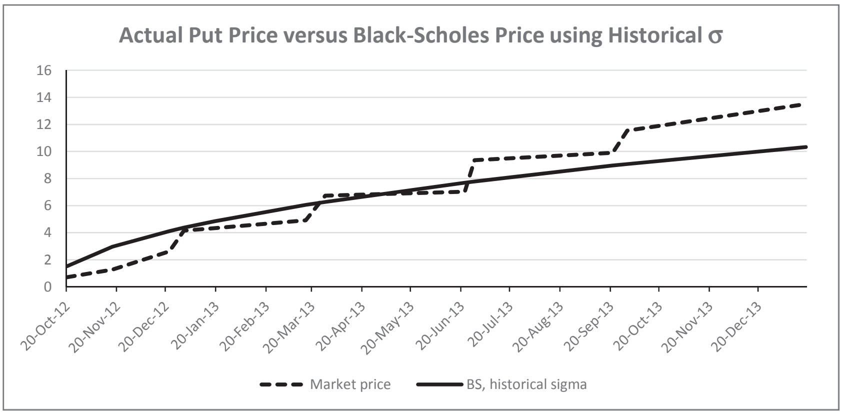
When we price the SPY at-the-money options for all maturities at the historical volatility, we see that the Black-Scholes model with the full-year's historical volatility as a proxy for σ does a reasonable job of pricing the calls:

	A	B	C	D	E	F	G	H
1	**PRICING THE SPY AT-THE-MONEY OPTIONS Historical and Implied volatility**							
2	Current date	09-Oct-12						
3	Current SPY price, S	144.2						
4	Exercise price, X	144						
5	Sigma, σ	16.34%	<-- =SPY!G8					
6								
7	**Expiration**	**Market price**	**Interest rate**	**Time to maturity, T**	**BS, historical sigma**		**Implied volatility**	
8	20-Oct-12	1.40	0.08%	0.0301	1.73	<-- =BSCall($B$3,$B$4,D8,C8,$B$5)	12.99%	<-- =CallVolatility($B$3,$B$4,D8,C8,B8)
9	17-Nov-12	2.80	0.08%	0.1068	3.18	<-- =BSCall($B$3,$B$4,D9,C9,$B$5)	14.33%	<-- =CallVolatility($B$3,$B$4,D9,C9,B9)
10	22-Dec-12	3.93	0.08%	0.2027	4.34		14.75%	
11	31-Dec-12	4.06	0.08%	0.2274	4.59		14.40%	
12	19-Jan-13	4.74	0.08%	0.2795	5.08		15.22%	
13	16-Mar-13	5.88	0.08%	0.4329	6.30		15.23%	
14	28-Mar-13	6.25	0.12%	0.4658	6.55		15.58%	
15	22-Jun-13	8.37	0.12%	0.7014	8.02		17.07%	
16	28-Jun-13	8.53	0.12%	0.7178	8.11		17.20%	
17	21-Sep-13	9.55	0.12%	0.9507	9.33		16.74%	
18	30-Sep-13	10.18	0.12%	0.9753	9.45		17.64%	
19	18-Jan-14	11.93	0.18%	1.2767	10.85		18.01%	

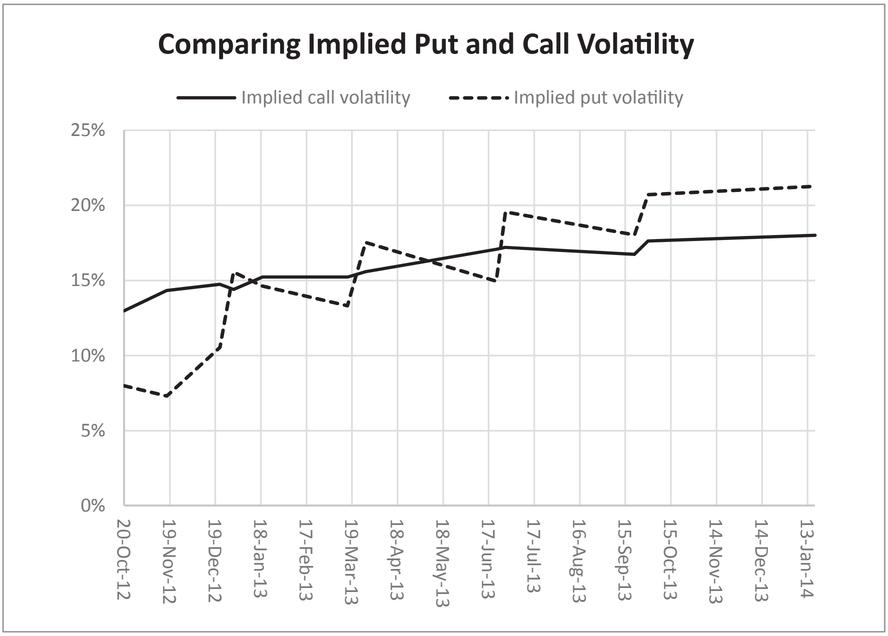
The historical volatility does a good job of pricing the call options:



SPY data for puts produce similar, though less good, results:



We also compute the implied volatilities of the at-the-money puts and calls:



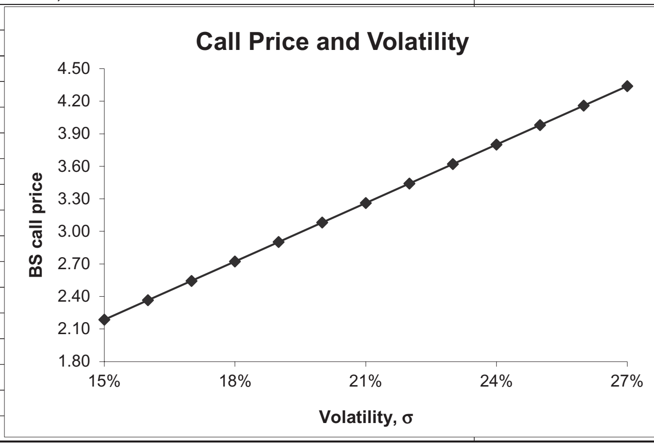
Comparing Historical with Implied Volatility

It is impossible to say which of these two methods is better for pricing options. On the one hand it is common to attribute to the historical returns some kind of validity as a predictor of the future anticipated returns. On the other hand the implied volatility gives a good indication of what the market is currently thinking. Our advice: Use them both and compare.

17.5 A VBA Function to Find the Implied Volatility

We design a Visual Basic for Applications (VBA) function which computes the implied volatility. To do this, we first note that the option price is a monotonic and increasing function of the sigma: Here's a **Data|Table** from our basic Black-Scholes spreadsheet:

	A	B	C	D
1	**BLACK-SCHOLES OPTION PRICE IS MONOTONIC IN SIGMA**			
2	S	45	Current stock price	
3	X	50	Exercise price	
4	T	1	Time to maturity of option (in years)	
5	r	8.00%	Risk-free rate of interest	
6	Sigma	30.00%	Stock volatility	
7				
8	Call price	4.88	<-- =BSCall(B2,B3,B4,B5,B6)	
9				
10	**Data table: Call price as function of volatility σ**			
11		4.8759	<-- =B8, table header	
12	15%	2.1858		
13	16%	2.3646		
14	17%	2.5437		
15	18%	2.7229		
16	19%	2.9023		
17	20%	3.0817		
18	21%	3.2612		
19	22%	3.4407		
20	23%	3.6202		
21	24%	3.7997		
22	25%	3.9792		
23	26%	4.1587		
24	27%	4.3381		
25				
26				
27				
28				



We use VBA to define a function **CallVolatility**, which finds the σ for a call option. The function is defined as **CallVolatility(Stock, Exercise, Time, Interest, Target)**, where the definitions are:

Stock is the stock price S

Exercise is the option's exercise price X

Time is the time to the option's maturity T

Interest is the interest rate r

Target is the call price C

The function finds σ for which the Black-Scholes formula = C.

```
Function CallVolatility(Stock, Exercise, Time, _
Interest, Target)
   High = 2
   Low = 0
   Do While (High - Low) > 0.0001
   If BSCall(Stock, Exercise, Time, Interest, _
   (High + Low) / 2) > Target Then
          High = (High + Low) / 2
          Else: Low = (High + Low) / 2
   End If
   Loop
   CallVolatility = (High + Low) / 2
End Function
```

The technique used by the function is very similar to the technique used in trial and error: We start with two estimates for the possible σ: A **High** estimate of 100% and a **Low** estimate of 0%. We now do the following:

- Plug the average of the **High** and the **Low** into the Black-Scholes formula. This gives us **CallOption(Stock, Exercise, Time, Interest, (High + Low) / 2)**. (Note the function **CallVolatility** assumes that the function **CallOption** is available to the spreadsheet.)
- If **CallOption(Stock, Exercise, Time, Interest, (High + Low) / 2 > Target**, then the current σ estimate of **(High + Low) / 2)** is too high and we replace **High** by **(High + Low) / 2**.
- If **CallOption(Stock, Exercise, Time, Interest, (High + Low) / 2 < Target**, then the current σ estimate of **(High + Low) / 2)** is too low and we replace **Low** by **(High + Low) / 2**.

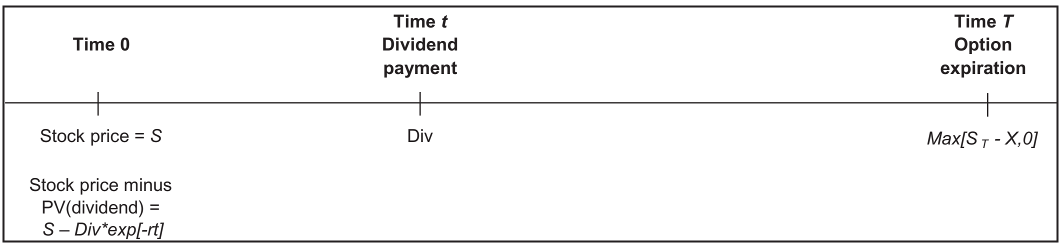
We repeat this procedure (often called the bisection method) until the difference **High-Low** is less than 0.0001 (or some other arbitrary constant).

17.6 Dividend Adjustments to the Black-Scholes

The Black-Scholes formula assumes that the option's underlying security pays no dividends prior to the exercise date *T*. In certain cases it is easy to make an adjustment to the model for dividends. This section looks at two such adjustments: We first look at option pricing when future dividends are known with certainty, and we then examine option pricing when the underlying security pays out a continuous dividend. The principle underlying both cases is the same: The options are priced on an adjusted underlying value which nets out the present value of dividends paid between the option purchase date and the option exercise date.

A Known Dividend to Be Paid Before the Option Expiration

It often happens that a stock's future dividend is known at the time the option is traded. This is most commonly the case when a dividend has already been announced, but it can also happen because many stocks pay quite regular and relatively inflexible dividends. In this case the option should be priced not on the current stock price *S* but on the stock price minus the present value of the dividend or dividends anticipated before the option expiration date *T*:



Here's an example: Coca-Cola (stock symbol KO) pays quarterly dividends in the middle of March, June, September, and November of each year. Dividends (see Figure 17.1) seem quite stable; on 28 July 2006, the last two dividends have been \$0.31 per share.

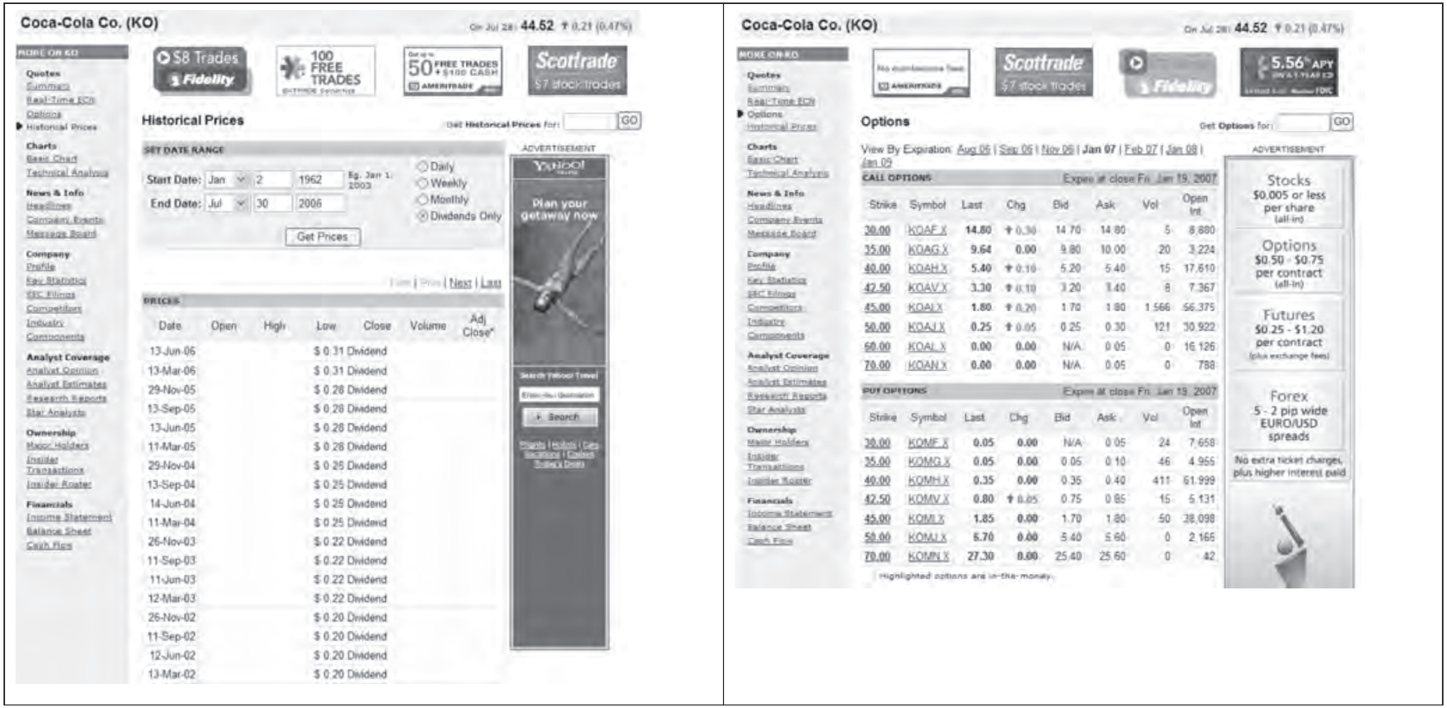


Figure 17.1
Yahoo data for Coca-Cola. Note the stability of the dividends. On 28 July 2006, the closing stock price is $44.52.

Computing the implied volatility for the January 2007 calls and puts on Coca-Cola shows that taking account of anticipated dividends makes a significant difference in the pricing. We can also deduce from the proximity of the prices which take dividends into account (cells B19:B20) versus the distance between the prices which do not take the dividends into account (cells B22:B23) that the former are correct.

	A	B	C	D	E
1	**PRICING THE COCA-COLA JAN07 CALLS AND PUTS**				
2	Current date	28-Jul-06			
3	Option expiration date	19-Jan-07			
4	Current stock price	44.52			
5	Interest rate	5.00%			
6					
7		Date	Anticipated dividend	Present value	
8	Mid September	13-Sep-06	0.31	0.31	<-- =C8*EXP(-$B$5*((B8-$B$2)/365))
9	End November	29-Nov-06	0.31	0.30	<-- =C9*EXP(-$B$5*((B9-$B$2)/365))
10					
11	Stock price net of PV(dividends)	43.91	-- =B4-SUM(D8:D9)		
12	Exercise price, X	45.00	<-- Approximately at the money		
13	Time to maturity, T	0.4795	<-- =(B3-B2)/365		
14	Interest rate, r	5.00%	Risk-free rate of interest		
15	Call price	1.80	<-- Call price on 28jul06		
16	Put price	1.85	<-- Put price on 28jul06		
17					
18	**Implied volatility**				
19	Call, S net of dividends	14.95%	<-- =CallVolatility(B11,B12,B13,B14,B15)		
20	Put, S net of dividends	15.15%	<-- =PutVolatility(B11,B12,B13,B14,B16)		
21					
22	Call, S with dividends	12.19%	<-- =CallVolatility(B4,B12,B13,B14,B15)		
23	Put, S with dividends	17.45%	<-- =PutVolatility(B4,B12,B13,B14,B16)		

Dividend Adjustments for Continuous Dividend Payouts—The Merton Model

In the above subsection we looked at the case of known future dividends. This subsection discusses a model due to Merton (1973) for the pricing options on a stock which pays continuous dividends. Continuous dividends may seem an odd assumption. But a basket of stocks such as the S&P 500 index or the Dow Jones 30 can best be approximated by the assumption of a continuous dividend payout, since there are many stocks and since the index components more or less pay out their dividends throughout the year.

Assuming the continuous dividend yield of k, Merton proved the following call option pricing formula:

$$C = Se^{-kT}N(d_1) - Xe^{-rT}N(d_2),$$

where

$$d_1 = \frac{\ln(S/X) + (r - k + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

This model is used below to price the exchange-traded fund which tracks the S&P 500 index:

	A	B	C
1	**MERTON'S DIVIDEND-ADJUSTED OPTION PRICING MODEL used here to price S&P 500 Spiders (symbol: SPY)**		
2	S	127.98	current stock price
3	X	127.00	exercise price
4	T	0.6329	<-- option expires 16-Mar-07, today's date 28-Jul-06
5	r	5.00%	risk-free rate of interest
6	k	1.70%	dividend yield
7	Sigma	14%	stock volatility
8			
9	d_1	0.3122	<-- =(LN(B2/B3)+(B5-B6+0.5*B7^2)*B4)/(B7*SQRT(B4))
10	d_2	0.2008	<-- =B9-B7*SQRT(B4)
11			
12	$N(d_1)$	0.6226	<--- Uses formula NormSDist(d_1)
13	$N(d_2)$	0.5796	<--- Uses formula NormSDist(d_2)
14			
15	Call price	7.51	<-- S*Exp(-k*T)*N(d_1)-X*exp(-r*T)*N(d_2)
16	Put price	3.94	<-- call price - S*Exp(-k*T) + X*Exp(-r*T): by Put-Call parity
17		3.94	<-- X*exp(-r*T)*N(-d_2)-S*Exp(-k*T)*N(-d_1): direct formula

The Merton model is often used to price currency options. Suppose we take an option on the euro. The option specifies a dollar exchange rate for euros (in the example below, the call option lets us buy 10,000 euros in 0.0575 years for $1.285 per euro). The asset underlying the option is a euro interest-bearing security with interest rate $r_€$.

	A	B	C	D
1	**PRICING AN OPTION TO BUY EUROS IN DOLLARS**			
2	S	1.276	Current exchange rate: U.S. dollar price of one euro	
3	X	1.285	Exercise price	
4	r_{US}	5.00%	U.S. interest rate	
5	$r_{€}$	5.50%	Euro interest rate	
6	T	0.0575	Time to maturity of option (in years)	
7	Sigma	4.70%	Euro volatility in dollars	
8	d_1	-0.6095	<--(LN(S/X)+(r_{US}-$r_{€}$+0.5*sigma^2)*T)/(sigma*SQRT(T))	
9	d_2	-0.6208	<-- d_1 - sigma*SQRT(T)	
10				
11	Number of euros per call contract	10,000		
12				
13	N(d_1)	0.2711	<--- Uses formula NormSDist(d_1)	
14	N(d_2)	0.2674	<--- Uses formula NormSDist(d_2)	
15				
16	Call price	23.69	<-- (S*Exp(-$r_{€}$*T)*N(d_1)-X*exp(-r_{US}*T)*N(d_2))*B11	
17	Put price	112.23	<-- (X*exp(-r_{US}*T)*N(-d_2)-S*Exp(-$r_{€}$*T)*N(-d_1))*B11: direct formula	

Intuition: The underlying asset of the currency option is a euro. The euro pays a dividend, which is the euro interest rate. Therefore the Merton model applies, with the underlying asset price being S*exp(-$r_{€}$*T), where $r_{€}$ is the interest rate on euros. Note also the change in d_1, where r_{US} -$r_{€}$ appears instead of r_{US} as in the regular Black-Scholes formula.

17.7 Using the Black-Scholes Formula to Price Structured Securities

A "structured security" is Wall Street parlance for securities which incorporate combinations of stocks, options, and bonds. In this section we give three examples of such securities and show how to price them using the Black-Scholes model.[5] In the process we also return to the discussion of Chapter 16, showing how the profit diagrams of option strategies can help us understand such securities.

A Simple Structured Security: Principal Protection Plus Participation in Market Upside Moves

A simple and popular structured offers guaranteed return of the investor's principal plus some participation in the upside moves of the market. Here's an example: Homeside Bank offers its customers the following "Principal-Protected, Upside Potential" security (PPUP).

- Initial investment in the security: $1,000.
- No interest paid on the security.
- In 5 years the PPUP pays back the $1,000 plus 50% of the increase in the S&P 500 index. Writing the index price today as S_0 and the index price in 5 years as S_T, the payoff on the PPUP can be written as:

5. Not all structureds can be priced using Black-Scholes; more complicated, path-dependent, securities often need to be priced using the Monte Carlo methods discussed in Chapters 24 and 25.

$$\$1{,}000\left[1+50\%*\max\left(\frac{S_T}{S_0}-1,0\right)\right]$$

To analyze the PPUP, we first rewrite the maturity payment as:

$$\$1{,}000\left[1+50\%*\max\left(\frac{S_T}{S_0}-1,0\right)\right]$$

$$=\underbrace{\$1{,}000}_{\substack{\uparrow\\\text{Payoff on}\\\text{zero coupon}\\\text{bond}}}+\$1{,}000*\frac{50\%}{S_0}*\underbrace{\max(S_T-S_0,0)}_{\substack{\uparrow\\\text{Payoff on at-the-money}\\\text{call option}}}$$

This shows that the PPUP payoff is composed of two parts:

- A \$1,000 return of principal. Since no interest is paid on this principal, its value today is the present value of the payment at the risk-free interest rate, \$1,000*e^{-rT}, where r is the interest rate and $T = 5$ is the maturity of the PPUP.
- $\$1{,}000*\frac{50\%}{S_0}$ times the value today of an at-the-money call on the S&P 500.

We can use the following spreadsheet to price this security:

	A	B	C
1	**ANALYZING A SIMPLE STRUCTURED PRODUCT** **\$1,000 Deposit with 50% Participation in S&P Increase over 5 Years**		
2	Initial S&P 500 price, S_0	950	<-- The price of the S&P 500 at PPUP issuance
3	Structured exercise price, X	950	
4	Risk-free interest rate for 5 years, r	5.00%	
5	Time to maturity, T	5	
6	Volatility of S&P 500, σ_{SP}	25%	
7	Participation rate	50%	<-- Percentage of increase in the S&P going to PPUP owner
8			
9	Strutured components, value today		
10	Bond paying \$1000 at maturity	778.80	<-- =EXP(-B4*B5)*1000
11	Participation rate /S_0*at-the-money call on S&P 500	162.52	<-- =1000*B7/B2*BSCall(B2,B3,B5,B4,B6)
12	Value of structured security today	941.32	<-- =SUM(B10:B11)

The value of the structured security (cell B12) is \$941.32. This valuation has two parts:

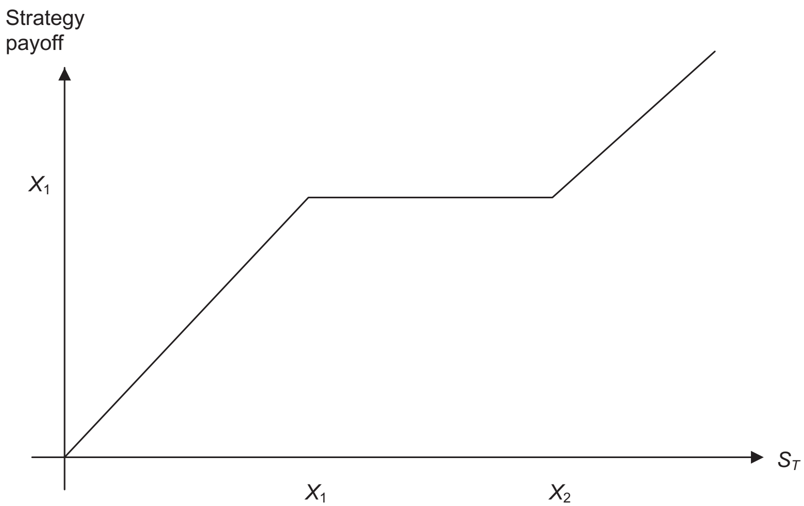
- The present value of the bond part of the PPUP is \$778.80 (cell B10).
- The value of $\$1{,}000*\frac{50\%}{950}$ at-the-money calls on the S&P is \$162.52.

Given the parameters in cells B2:B7, the PPUP is *overpriced*—it sells for \$1,000, whereas its market value ought to be \$941.32. Another way to think about the structured is to compute its implied volatility: What σ_{SP} (cell B6) will give the market valuation (cell B12) of the PPUP to equal the \$1,000 price being asked by the Homeside Bank? Either **Goal Seek** or **Solver** will solve this problem:

	A	B	C
1	**ANALYZING A SIMPLE STRUCTURED PRODUCT** **\$1,000 Deposit with 50% Participation in S&P Increase over 5 Years**		
2	Initial S&P 500 price, S_0	950	<-- The price of the S&P 500 at PPUP issuance
3	Structured exercise price, X	950	
4	Risk-free interest rate for 5 years, r	5.00%	
5	Time to maturity, T	5	
6	Volatility of S&P 500, σ_{SP}	42.00%	
7	Participation rate	50%	<-- Percentage of increase in the S&P going to PPUP owner
8			
9	Structured components, value today		
10	Bond paying \$1000 at maturity	778.80	<-- =EXP(-B4*B5)*1000
11	Participation rate /S_0*at-the-money call on S&P 500	221.20	<-- =1000*B7/B2*BSCall(B2,B3,B5,B4,B6)
12	Value of structured security today	1000.00	<-- =SUM(B10:B11)

More Complicated Structured Products

Suppose you want to create a security with the following payoff pattern:



The payoff pattern increases (dollar for dollar) as the terminal price of the underlying asset increases from $0 \le S_T \le X_1$. Between X_1 and X_2, the payoff pattern is flat, and for $X_2 \le S_T$, the payoff again increases, dollar-for-dollar with the price of the underlying. The algebraic formula for this payoff pattern is:

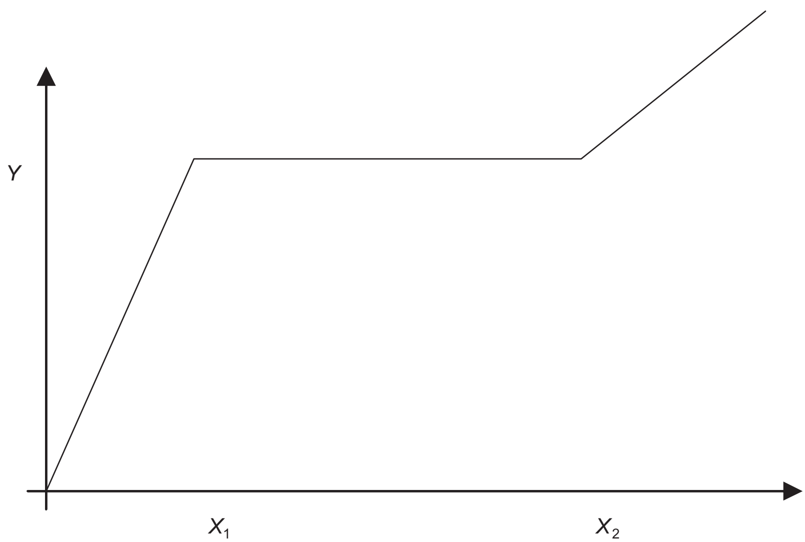
$$X_1 - \max(X_1 - S_T, 0) + \max(S_T - X_2, 0)$$

To prove that this formula creates the graph:

$$X_1 - \underbrace{\max(X_1 - S_T, 0)}_{\substack{\uparrow \\ \text{Payoff of written} \\ \text{put}}} + \underbrace{\max(S_T - X_2, 0)}_{\substack{\uparrow \\ \text{Payoff of bought} \\ \text{call}}}$$

$$= \begin{cases} X_1 - X_1 + S_T = S_T & S_T < X_1 \\ X_1 & X_1 \le S_T < X_2 \\ X_1 + S_T - X_2 & X_2 \le S_T \end{cases}$$

A slightly more complicated payoff pattern is the following:



The initial part of the payoff has slope Y/X_1, and the second increasing part of the payoff pattern has slope Y/X_2. This payoff pattern is created by the following formula:

$$Y - \underbrace{\frac{Y}{X_1}\max(X_1 - S_T, 0)}_{\substack{\uparrow \\ \text{Payoff of } \frac{Y}{X_1} \\ \text{written puts}}} + \underbrace{\frac{Y}{X_2}\max(S_T - X_2, 0)}_{\substack{\uparrow \\ \text{Payoff of } \frac{Y}{X_2} \\ \text{bought calls}}}$$

To prove that this is indeed the payoff:

$$Y - \underbrace{\frac{Y}{X_1}\max(X_1 - S_T, 0)}_{\substack{\uparrow \\ \text{Payoff of } \frac{Y}{X_1} \\ \text{written puts}}} + \underbrace{\frac{Y}{X_2}\max(S_T - X_2, 0)}_{\substack{\uparrow \\ \text{Payoff of } \frac{Y}{X_2} \\ \text{bought calls}}}$$

$$= \begin{cases} Y - \dfrac{Y}{X_1}(X_1 - S_T) = \dfrac{Y}{X_1}S_T & S_T < X_1 \\ Y & X_1 \le S_T < X_2 \\ Y + \dfrac{Y}{X_2}(S_T - X_2) = \dfrac{Y}{X_2}S_T & X_2 \le S_T \end{cases}$$

As an example of this kind of structured payoff security, Figure 17.2 on p. 447 shows the term sheet for a structured product issued by ABN-AMRO bank. The payment on this "Airbag" security depends on the value of the Stoxx50—an index of European stocks. Here are the details:

- Issuance date: 24 March 2003
- Terminal date: 24 March 2008
- Cost €1,020
- Payment at terminal date:

Payment at maturity

$$= \begin{cases} 1{,}000 * 1.33 * \left(\dfrac{Stoxx50_{Maturity}}{Stoxx50_{Initial}} \right) & \text{If } Stoxx50_{Maturity} < 1{,}618.50 \\ 1{,}000 & 1{,}618.50 < Stoxx50_{Maturity} < 2{,}158 \\ 1{,}000 * \left(\dfrac{Stoxx50_{Maturity}}{Stoxx50_{Initial}} \right) & \text{If } Stoxx50_{Maturity} > 2{,}158 \end{cases}$$

We recognize this security as one whose payoff has the form discussed earlier:

$$\underbrace{Y}_{\substack{\uparrow \\ \text{Bond payoff}}} - \underbrace{\frac{Y}{X_1}\max(X_1 - S_T, 0)}_{\substack{\uparrow \\ \frac{Y}{X_1} \text{ written puts with} \\ \text{exercise price } X_1}} + \underbrace{\frac{Y}{X_2}\max(S_T - X_2, 0)}_{\substack{\uparrow \\ \frac{Y}{X_2} \text{ purchased call} \\ \text{with exercise price } X_2}}$$

where

$$X_1 = 1{,}618.50$$
$$X_2 = 2{,}158$$
$$Y = 1{,}000$$

The spreadsheet on p. 448 shows the payoff. Cell B7 shows the payoff definition given by the Airbag issuer, and cell B8 shows the payoff in the option terms defined above. The data table in cells A13:B29 show that these two definitions are equivalent:

ABN·AMRO

AirBag on the Euro STOXX 50

17 March 2003

FINAL TERMS AND CONDITIONS

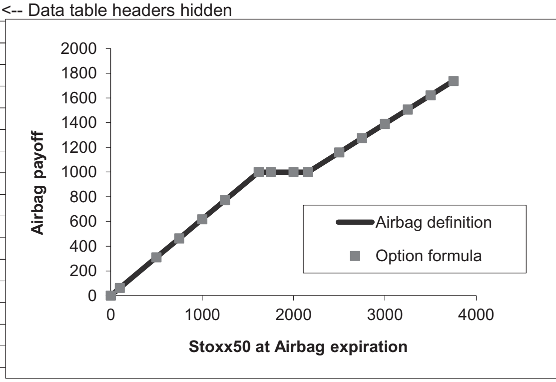
We are pleased to present for your consideration the transaction described below. We are willing to negotiate a transaction with you because we understand that you have sufficient knowledge, experience and professional advice to make your own evaluation of the merits and risks of a transaction of this type and you are not relying on ABN AMRO Bank N.V. nor any of the companies in the ABN AMRO group for information, advice or recommendations of any sort other than the factual terms of the transaction. This term sheet does not identify all the risks (direct or indirect) or other considerations which might be material to you when entering into the transaction. You should consult your own business, tax, legal and accounting advisors with respect to this proposed transaction and you should refrain from entering into a transaction with us unless you have fully understood the associated risks and have independently determined that the transaction is appropriate for you. Due to the proprietary nature of this proposal please understand that it is confidential.

SUMMARY	**Issuer & Lead Manager:**	ABN AMRO Bank N.V. (Senior Long Term Debt Rating: Moody's Aa3, S&P AA-)
	Issue:	AirBag on the Euro STOXX 50
	Underlying:	Euro STOXX 50 (Bloomberg: SX5E)
	Spot Reference (SX5E(t)):	2158.00
	Issue Price:	**EUR 1,020**
	Entitlement:	1
	Issue Size:	5,000 Certificates
	AirBag Start:	100% of the Spot Reference (2158.00)
	AirBag Stop:	75% of the Spot Reference (1618.50)
	Percentage drop without any loss:	25%
	SX5E(t1):	Official closing level of the Underlying on the Valuation Date
	Redemption:	1. If SX5E(t1) is less than or equal to the AirBag Stop: EUR 1,000 x 1.33 x $\left(\frac{SX5E(t1)}{SX5E(t)}\right)$ 2. If SX5E(t1) is greater than the AirBag Stop but less than or equal to the AirBag Start: EUR 1,000 x 100% 3. If SX5E(t1) is greater than the AirBag Start: EUR 1,000 x $\left(\frac{SX5E(t1)}{SX5E(t)}\right)$
	Form:	Global bearer (permanent)
	Clearing:	Euroclear Bank SA, Clearstream Banking SA
	ISIN Code:	XS0165647966
	Valoren Code:	1578781
	Common Code:	16564796
	Minimum Trading Size:	1 AirBag Certificate
	Quoted on:	Reuters page: ABNPB15, Bloomberg page: AAPB, Internet: www.abnamro-sp.com
	Listing:	None
	Applicable Law:	English
	Selling Restrictions:	No sales to US persons or into the US, standard Dutch and UK selling restrictions apply.
TIMETABLE	**Launch Date:**	17/03/03
	Pricing Date:	17/03/03
	Issue & Payment Date:	24/03/03
	Valuation & Expiration Date:	14/03/08
	Final Settlement Date:	21/03/08

This term sheet is for information purposes only and does not constitute an offer to sell or a solicitation to buy any security or other financial instrument. All prices are indicative and dependent upon market conditions and the terms are liable to change and completion in the final documentation.

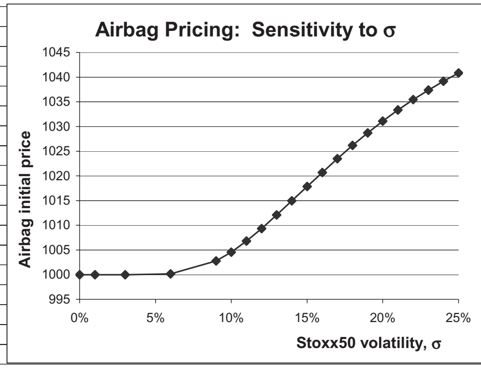
Figure 17.2
The ABN-AMRO term sheet for its Euro Stoxx50 Airbag security.

	A	B	C	D
1				**ABN-AMRO AIRBAG**
2	Y	1,000.00		
3	X_1	1,618.50		
4	X_2	2,158.00		
5	S_T	2,373.80		
6	Airbag payoff			
7	By Airbag definition	1,100.00	<-- =IF(B5<B3,B2*(B4/B3)*B5/B4,IF(B5>B4,B2*B5/B4,1000))	
8	Option formula	1,100.00	<-- =B2-B2/B3*MAX(B3-B5,0)+B2/B4*MAX(B5-B4,0)	
9				
10				
11	**Data table of payoffs**			
12	**S_T**	**Airbag definition**	**Option formula**	
13				<-- Data table headers hidden
14	0	0.00	0.00	
15	100	61.79	61.79	
16	500	308.93	308.93	
17	750	463.39	463.39	
18	1,000	617.86	617.86	
19	1,250	772.32	772.32	
20	1,618.5	1,000.00	1,000.00	
21	1,750	1,000.00	1,000.00	
22	2,000	1,000.00	1,000.00	
23	2,158	1,000.00	1,000.00	
24	2,500	1,158.48	1,158.48	
25	2,750	1,274.33	1,274.33	
26	3,000	1,390.18	1,390.18	
27	3,250	1,506.02	1,506.02	
28	3,500	1,621.87	1,621.87	
29	3,750	1,737.72	1,737.72	
30				



To see how the Airbag is priced, we use the Black-Scholes model and find the Stoxx50 volatility implied by the Airbag price:

	A	B	C
1	**PRICING THE ABN-AIRBAG Find the Implied Volatility**		
2	Stoxx50 price today, S_0	2,158.0	
3	X_1	1,618.50	
4	X_2	2,158.0	
5	Y	1,000.0	
6	Risk-free interest rate for 5 years, r	7.00%	
7	Time to maturity, T	5	
8	Volatility of the Stoxx50, sigma	15.75%	
9			
10	Airbag components, value today		
11	Bond paying X_1 at maturity	704.69	<-- =EXP(-B6*B7)*B5
12	Y/X_1 * written puts with exercise X_1	-4.69	<-- =-B5/B3*BSPut(B2,B3,B7,B6,B8)
13	Purchased call with exercise X_2	320.01	<-- =B5/B4*BSCall(B2,B4,B7,B6,B8)
14	Value of structured security today	1020.00	<-- =SUM(B11:B13)
15			
16			
17	**Table: Sensitivity of Airbag to Sigma**	1,020.00	<-- =B14, data table header
18	0%	1,000.00	
19	1%	1,000.00	
20	3%	1,000.00	
21	6%	1,000.16	
22	9%	1,002.76	
23	10%	1,004.57	
24	11%	1,006.80	
25	12%	1,009.34	
26	13%	1,012.09	
27	14%	1,014.95	
28	15%	1,017.84	
29	16%	1,020.70	
30	17%	1,023.49	
31	18%	1,026.16	
32	19%	1,028.70	
33	20%	1,031.11	
34	21%	1,033.35	
35	22%	1,035.45	
36	23%	1,037.39	
37	24%	1,039.19	
38	25%	1,040.84	



When the Stoxx50 σ is 15.75% (cell B8), the Airbag's price is €1,020 (cell B14). The table shows the sensitivity of this price to the σ. Notice that Airbag values are not very sensitive to σ: Doubling the sigma from 10% to 20% increases the Airbag value by about €17. This is because of the offsetting values of the short put and the long call in the Airbag.

We can do one more exercise on the Airbag. Using a two-dimensional **Data Table**, we examine the Airbag's price sensitivity to both time to maturity T and the Stoxx50 volatility σ:

	A	B	C	D	E	F	G	H
1	**ABN-AMRO AIRBAG SENSITIVITY TO TIME TO MATURITY AND SIGMA**							
2	Stoxx50 price today, S_0	2,158.0						
3	X_1	1,618.50						
4	X_2	2,158.0						
5	Y	1,000.0						
6	Risk-free interest rate for 5 years, r	7.00%						
7	Time to maturity, T	5						
8	Volatility of the Stoxx50, sigma	15.75%						
9								
10	Airbag components, value today							
11	Bond paying X_1 at maturity	704.69	<-- =EXP(-B6*B7)*B5					
12	Y/X_1 * written puts with exercise X_1	-4.69	<-- =-B5/B3*BSPut(B2,B3,B7,B6,B8)					
13	Purchased call with exercise X_2	320.01	<-- =B5/B4*BSCall(B2,B4,B7,B6,B8)					
14	Value of structured security today	1020.00	<-- =SUM(B11:B13)					
15								
16			**Time to maturity, T**					
17	Data table header: =B14	1020.00	**5**	**4**	**3**	**2**	**1**	**0.0001**
18		**5%**	1000.02	1000.07	1000.20	1000.59	1001.77	1000.20
19		**10%**	1004.57	1006.22	1008.40	1011.13	1013.78	1000.40
20		**15%**	1017.84	1021.09	1024.72	1028.28	1029.65	1000.59
21		**20%**	1031.11	1035.21	1039.61	1043.69	1044.54	1000.79
22	**Volatility of the Stoxx50, sigma -->**	**25%**	1040.84	1045.48	1050.44	1055.14	1056.54	1000.99
23		**30%**	1047.16	1052.22	1057.69	1063.09	1065.58	1001.19
24		**35%**	1050.86	1056.29	1062.26	1068.39	1072.19	1001.39
25		**40%**	1052.66	1058.44	1064.88	1071.75	1076.95	1001.59
26		**45%**	1053.10	1059.19	1066.10	1073.70	1080.28	1001.79
27		**50%**	1052.55	1058.94	1066.29	1074.59	1082.53	1001.99

The Airbag is a fairly stable security—its price varies no more than 10% for a wide variety of σ's and for nearly all times to maturity.

A Reverse Convertible: Analyzing the UBS "Goals"

The Swiss bank UBS has issued a series of stock-linked securities called "Goals." All of the Goals pay interest on the initial price; the final repayment depends on the market price of the underlying stock: If the stock price is high, the Goals investor gets back her initial investment, and if the stock price is low, the Goals investor is paid out in a package of shares whose value is less than her initial investment.

An example of such a security is the Cisco-linked Goals issued by UBS on 17 January 2001. The main details of this security are:

- The purchaser pays UBS $1,000 on 23 January 2001. In return, she gets 3 payments of \$97.50 $\left(= \frac{19.50\%}{2} * \$1{,}000 \right)$ on 23 July 2001, 23 January 2002, and 23 July 2002.

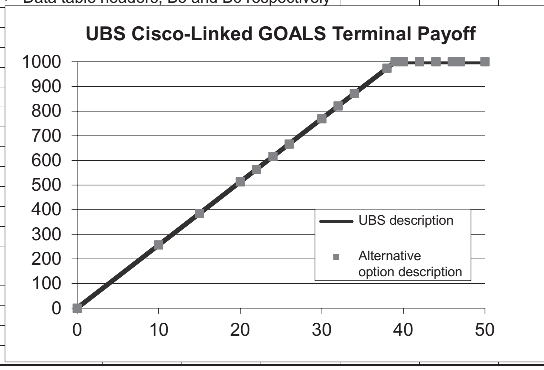
- On 23 July 2002, in addition to the \$97.50 payment
  - If the price of Cisco stock > \$39 per share, the purchaser of the security gets \$1,000.
  - If the price of Cisco stock < \$39 per share, the purchaser gets $\frac{1{,}000}{39} = 25.641$ shares of Cisco.
- The closing stock price of Cisco on 23 January 2001 was 42.625.
- The continuously compounded risk-free interest rate at the time of the Goals issuance was 5.2% annually (2.6% semiannually).

To analyze the Cisco-linked Goals, we start by noting that the cash flow can be written as:

Time	0	1	2	3
	-1,000	97.50	97.50	97.50 1,000-25.641*Max(39-S_T,0)

To prove this, we write a short spreadsheet which compares the Goals payoff per definition with its payoff as described above:

	A	B	C	D	E	F	G	H	I	J
1	**EQUIVALENCE OF 2 WAYS OF WRITING THE PAYOFF**									
2	Cisco price, 23 July 2002, S_T	32								
3	Payoff ratio	25.6410	<-- =1000/39							
4	Terminal payoff									
5	As described by UBS	820.51	<-- =IF(B2>39,1000,B2*B3)							
6	In option terms	820.51	<-- =1000-B3*MAX(39-B2,0)							
7										
8	**Data table: Comparing the payoff on Cisco-linked Goals**									
9	**Cisco stock price on 23 July 2002, S_T**	**UBS description**	**Alternative option description**							
10		820.51	820.51	<-- Data table headers, B5 and B6 respectively						
11	0	0.00	0.00							
12	10	256.41	256.41							
13	15	384.62	384.62							
14	20	512.82	512.82							
15	22	564.10	564.10							
16	24	615.38	615.38							
17	26	666.67	666.67							
18	30	769.23	769.23							
19	32	820.51	820.51							
20	34	871.79	871.79							
21	38	974.36	974.36							
22	39	1,000.00	1,000.00							
23	40	1,000.00	1,000.00							
24	42	1,000.00	1,000.00							
25	44	1,000.00	1,000.00							
26	46	1,000.00	1,000.00							
27	47	1,000.00	1,000.00							
28	50	1,000.00	1,000.00							



The equivalence of these two definitions means that the purchaser of the Cisco-linked Goals:

- Acquires a \$1,000 bond paying 9.75% interest semi-annually.
- Writes UBS 25.641 puts with exercise $X = 39$ and with time to maturity $T = 1.5$.

At the time the Goals were issued, the semi-annual interest rate was around 2.6%, far below 9.75%. Thus the bond component of the Goals was worth much more than \$1,000. On the other hand, the purchaser of the Goals was *giving* UBS 25.641 puts. The value of this "gift" should be accounted for in any analysis of the Goals.

We illustrate two ways to value the Goals. The first method assumes that the equilibrium NPV of any security should be zero. Doing this for the UBS security gives:

$$\underbrace{-1{,}000+\frac{97.50}{(1+r)^{0.5}}+\frac{97.50}{(1+r)^{1.0}}+\frac{1{,}097.50}{(1+r)^{1.5}}}_{\substack{\uparrow\\ \text{Value=\$205.11}\\ \text{at } r=2.6\%}}$$
$$-\,25.641 * \textit{Puts on Cisco}(X=39, T=1.5)=0$$

In the spreadsheet below, we use this logic to price the puts embedded in the UBS security and we compare this price to the Black-Scholes price.

$$\textit{Implicit put valuation}=\frac{1}{25.641}\left[\begin{array}{l}-1{,}000+\dfrac{97.50}{(1+2.6\%)^{0.5}}\\ +\dfrac{97.50}{(1+2.6\%)^{1.0}}+\dfrac{1{,}097.50}{(1+2.6\%)^{1.5}}\end{array}\right]$$
$$=\frac{\$205.11}{25.641}=\$8.00$$

UBS is implicitly paying the Goals purchaser \$8.00 per Cisco put. However, as shown below, the Black-Scholes price of such a put is \$11.71 if $\sigma=80\%$.[6] This makes the Goals a bad buy.

6. The Cisco-linked Goals was issued during the NASDAQ crash of the early 2000s. The implied volatility of Cisco stock during this period varied between 80% and 120%.

	A	B	C
1	**PRICING THE UBS GOALS IMPLICIT PUT**		
2	Annual risk-free rate	5.20%	
3	Coupon rate	19.50%	
4	Initial cost	1,000	
5	Conversion ratio: # of shares of Cisco received if share price is low	25.641	<-- =1000/39
6			
7	**Valuing the fixed payments at 5.20%**		
8	Fixed payments		
9	**Date**	**Cash flow**	
10	23-Jan-01	(1,000.00)	
11	23-Jul-01	97.50	<-- =$B$3*B4/2
12	23-Jan-02	97.50	
13	23-Jul-02	1,097.50	
14	**PV of Goals bond component**	**205.11**	<-- =XNPV(B2,B10:B13,A10:A13)
15			
16	Value of 25.641 puts embedded in Goals	205.11	<-- =B14
17	**Value per put**	**8.00**	<-- =B16/25.641
18	This is what UBS is *paying* the Goals purchaser for the embedded puts.		
19			
20	**Valuing the puts with Black-Scholes**		
21	S	42.625	Current stock price
22	X	39	Exercise price
23	r	5.20%	Risk-free interest rate
24	T	1.5	Time to maturity of option (in years)
25	Sigma	80%	Stock volatility
26	**Put price**	**11.71**	<-- =BSPut(B21,B22,B24,B23,B25)
27			
28	Is the Goals a good buy?	No	<-- =IF(B17>B26,"Yes","No")
29			
30	**Technical note**: For didactic clarity, the computations use 5.2% as the interest rate for valuing both the bond component of the Goals (rows 10-14) and for the option valuation. Given a 2.6% semi-annual discrete interest rate, it would be technically more correct to use an equivalent continuously compounded interest rate of **LN((1.026)^2)** in the option computations. The reader can confirm that the effect of this correction is negligible.		

There is another way to look at the Goals. Consider a Goals purchaser who buys the Goals and also buys 25.641 puts on Cisco with $T = 1.5$, $X = 39$. This "engineered" combination of a Goals + 25.641 puts creates a risk-free security:

$$
Payoff\ Goals + 25.641\ puts = \begin{cases} \$1{,}000 \underbrace{-25.641 * (39 - S_T)}_{\substack{\uparrow \\ \text{Payment on Goals} \\ \text{embedded puts}}} \\ \quad + \underbrace{25.641 * (39 - S_T)}_{\substack{\uparrow \\ \text{Payment on puts} \\ \text{purchased}}} = \$1{,}000 & S_T < 39 \\ \$1{,}000 & S_T \geq 39 \end{cases}
$$

In the spreadsheet below, we assume that the bought puts are priced using Black-Scholes and compare the rate of return on this "engineered" security to the risk-free rate.

	A	B	C
1	**CREATING A RISKLESS SECURITY WITH THE UBS GOALS AND 25.641 PUTS**		
2	**Initial cash flows**		
3	Buy UBS security	-1,000.00	
4	Buy 25.641 puts	-300.21	<-- =-25.641*BSPut(B16,B17,B19,B18,B20)
5			
6	**Cash flow of "engineered" security: GOALS + 25.641 bought puts**		
7	**Date**	**Cash flow**	
8	23-Jan-01	(1,300.21)	<-- =SUM(B3:B4)
9	23-Jul-01	97.50	
10	23-Jan-02	97.50	
11	23-Jul-02	1,097.50	
12			
13	IRR of above	-0.43%	<-- =XIRR(B8:B11,A8:A11)
14			
15	**Inputs for Black-Scholes formula in cell B4**		
16	S	42.625	Current stock price
17	X	39	Exercise price
18	r	5.20%	Risk-free rate of interest
19	T	1.5	Time to maturity of option (in years)
20	Sigma	80%	Stock volatility

Cell B13 uses the Excel function **XIRR** (see Chapter 33) to compute the annualized internal rate of return on the engineered security. Clearly this return is less than the alternative risk-free rate of return (5.2%), which can be earned in the market. This is an alternative confirmation of the fact that the Goals are a bad buy.

```
                THE SEC FILING FOR UBS AG $60,000,000 GOALS
                                 UBS AG
                              $60,000,000
                     19.5% GOALs DUE JULY 23, 2002
------------------------------------------------------------------

     Each note being offered has the following terms:

- - Issuer:           UBS AG
- - Issue:            $60,000,000 USD principal amount
                      of GOALs due July 23, 2002 linked
                      to shares in the common stock of
                      Cisco Systems, Inc.
- - Coupon:           19.5% per annum, payable
                      semi-annually in arrears on each
                      January 23 and July 23 which shall
                      be composed of (1) an interest
                      coupon representing a rate of 5.2%
                      per annum and (2) a coupon
                      representing an option premium of
                      14.3% per annum
- - Initial price of  $39.00 per share, subject to underlying stock
                      antidilution adjustments (strike price):
- - Key dates:        Trade: January 17, 2001
                      Settlement: January 23, 2001
                      Determination: July 18, 2002
                      Maturity: July 23, 2002

Proceeds at maturity are based on the closing price of Cisco Systems,
Inc. common stock three business days before maturity:

If the closing price of Cisco Systems, Inc. common stock is at or
above the initial price per share of $39.00, holders will receive a
cash payment equal to the principal amount of their GOALs.

If the closing price of Cisco Systems, Inc. is lower than the initial
price per share of $39.00, holders will receive 25.641 shares of Cisco
Systems, Inc. common stock for each $1,000 principal amount of their
GOALs (the stock redemption amount). Fractional shares will be paid in
cash. The number of shares received for each $1,000 invested will be
calculated by dividing the initial price per share of $39.00 into
$1,000. The stock redemption amount and the initial price per share of
$39.00 (strike price) may change due to stock splits or other
corporate actions.
```

Figure 17.3
From the SEC Filing for the UBS Cisco-linked Goals.

17.8 Bang for the Buck with Options

This section presents another application of the Black-Scholes formula. Suppose that you are convinced that a given stock will go up in a very short period of time. You want to buy calls on the stock that have a maximum "bang for the buck"—that is, you want the percentage profit on your option investment to be maximal. Using the Black-Scholes formula, it is easy to show that you should:

- Buy calls with the shortest possible maturity.
- Buy calls that are most highly out of the money (i.e., with the highest exercise price possible).

Here's a spreadsheet illustration:

	A	B	C
1	"BANG FOR THE BUCK" WITH OPTIONS		
2	S	25	Current stock price
3	X	25	Exercise price
4	r	6.00%	Risk-free rate of interest
5	T	0.5	Time to maturity of option (in years)
6	Sigma	30%	Stock volatility
7			
8	d_1	0.2475	<-- (LN(S/X)+(r+0.5*sigma^2)*T)/(sigma*SQRT(T))
9	d_2	0.0354	<-- d_1-sigma*SQRT(T)
10			
11	N(d_1)	0.5977	<-- Uses formula NormSDist(d_1)
12	N(d_2)	0.5141	<-- Uses formula NormSDist(d_2)
13			
14	Call price	2.47	<-- S*N(d_1)-X*exp(-r*T)*N(d_2)
15	Put price	1.73	<-- call price - S + X*Exp(-r*T): by put-call parity
16			
17	Call bang	6.0483	<-- =B11*B2/B14
18	Put bang	5.8070	<-- =NORMSDIST(-B8)*B2/B15

The "call bang" defined in cell B17 is simply the percentage change in the call price dividend by the percentage change in the stock price (in economics this is known as the "price elasticity"):

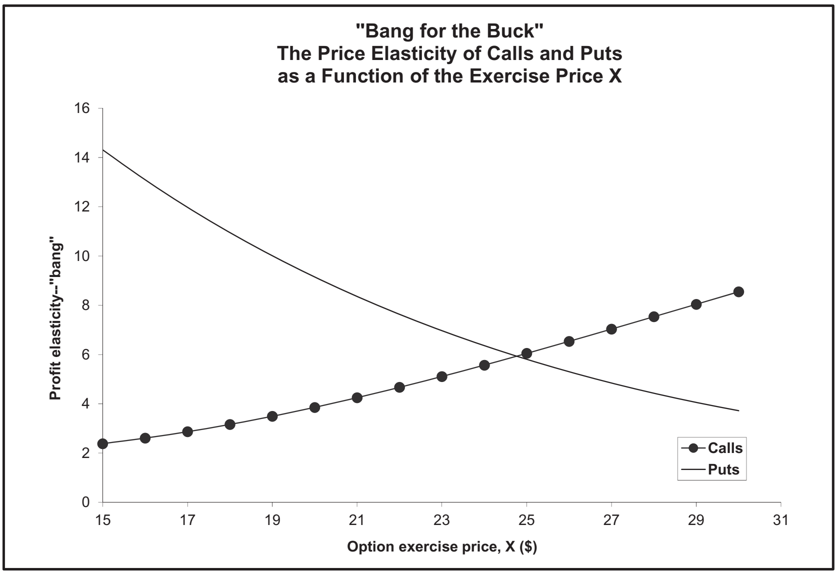
$$Call\ bang = \frac{\partial C / C}{\partial S / S} = \frac{\partial C}{\partial S}\frac{S}{C} = N(d_1)\frac{S}{C}$$

Similarly, for a put, the "bang for the buck" is defined by the formula below (of course the story behind the put "bang for the buck" is that you are convinced that the stock price will go down):

$$Put\ bang = \frac{\partial P / P}{\partial S / S} = \frac{\partial P}{\partial S}\frac{S}{P} = -N(-d_1)\frac{S}{P}$$

This is defined in cell B18. To make the numbers easier to understand, we have dropped the minus sign, making the "put bang" = $N(-d_1)\frac{S}{P}$.

The graph below shows the "bang for the buck" for both calls and puts:



If you play with the spreadsheet, you will see that the longer the time to maturity, the less the bang for the buck. (Another way of saying all of this is that the most risky options are the most out-of-the-money and the shortest-term options.)

	D	E	F	G	H	I
23		**Data table: Effect of S and T on "call bang"**				
24	Data table header: =B17					
25			**T--option time to exercise**			
26		6.0483	**0.25**	**0.5**	**0.75**	**1**
27		**15**	25.8856	14.1771	10.1696	8.1112
28		**16**	23.3305	12.9884	9.4123	7.5625
29		**17**	20.9954	11.9033	8.7218	7.0623
30		**18**	18.8590	10.9121	8.0914	6.6057
31		**19**	16.9052	10.0067	7.5154	6.1882
32		**20**	15.1222	9.1805	6.9891	5.8062
33		**21**	13.5007	8.4274	6.5082	5.4565
34		**22**	12.0334	7.7424	6.0691	5.1362
35		**23**	10.7137	7.1205	5.6682	4.8426
36		**24**	9.5347	6.5572	5.3025	4.5737
37		**25**	8.4893	6.0483	4.9691	4.3272
38		**26**	7.5694	5.5896	4.6655	4.1012
39		**27**	6.7664	5.1773	4.3892	3.8941
40		**28**	6.0706	4.8074	4.1379	3.7043
41		**29**	5.4720	4.4764	3.9094	3.5303
42		**30**	4.9598	4.1807	3.7019	3.3708

17.9 The Black (1976) Model for Bond Option Valuation[7]

Black (1976) suggested an adaptation of the Black-Scholes model which is often used for simple valuation of options on bonds or forwards. Letting *F* stand for the forward price of an asset, the Black-Scholes equation given in section 17.2 is replaced by

7. This section is advanced and can be skipped on first reading. A full discussion of the pricing of options on bonds is beyond the scope of the current edition of this book. However, this very useful and often-used adaptation of the Black model is simple enough to attach to this chapter.

$$C = e^{-rT}[FN(d_1) - XN(d_2)],$$

where

$$d_1 = \frac{\ln(F/X) + \sigma^2 T/2}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

The corresponding put price is given by:

$$P = e^{-rT}[XN(-d_2) - FN(-d_1)]$$

To use the Black (1976) model, consider the case of an option on a zero coupon bond, where the option maturity is $T = 0.5$. The option gives the holder the opportunity to buy the bond at time T for exercise price $X = 130$. Suppose that the risk-free interest rate is $r = 4\%$. If the forward price of the bond to the exercise date is $F = 133$ and the volatility of the forward price is $\sigma = 6\%$, then the pricing of the bond option using the Black (1976) model is given below.

	A	B	C
1	**USING THE BLACK (1976) MODEL TO PRICE A BOND OPTION**		
2	F	133.011	<-- Bond forward price
3	X	130.000	<-- Exercise price
4	r	4.00%	<--Risk-free rate of interest
5	T	0.5	
6	Sigma	6%	<-- Bond forward price volatility, σ
7			
8	d_1	0.5609	<-- =(LN(B2/B3)+B6^2*B5/2)/(B6*SQRT(B5))
9	d_2	0.5185	<-- =B8-SQRT(B5)*B6
10			
11	Call price	4.13	<-- =EXP(--B4*B5)*(B2*NORMSDIST(B8)-B3*NORMSDIST(B9))
12	Put price	1.02	<-- =EXP(-B4*B5)*(B3*NORMSDIST(-B9)-B2*NORMSDIST(-B8))

Thus a call on the bond is worth 4.13 and a put is worth 1.02.

Determining the Bond's Forward Price

The forward interest rate is the interest rate which can be locked in today for a loan in the future. In the example below, the current 7-year rate is 6% and the 4-year rate is 5%. By simultaneously creating a deposit in one maturity and a loan in the other maturity, we create a security which has zero cash flows everywhere except at years 4 and 7:

	A	B	C	D	E	F	G	H	I	J
1	**THE FORWARD INTEREST RATE**									
2	Bond maturity, W	7								
3	Option maturity, T	4								
4	Year W pure discount rate	6%								
5	Year T pure discount rate	5%								
6										
7	**Discretely-compounded interest rates**									
8		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
9	7-year deposit at 6.00%	100.00							-150.36	
10	4-year loan at 5.00%	-100.00				121.55				
11	**Sum of above: A 3-year deposit at year 4**	0.00				121.55			-150.36	
12										
13	Discretely compounded forward interest rate from year 4 to year 7	7.35%	<-- =(-I11/F11)^(1/(B2-B3))-1							
14										
15	**Continuously-compounded interest rates**									
16		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
17	7-year deposit at 6.00%	100.00							-152.20	
18	4-year loan at 5.00%	-100.00				122.14				
19	**Sum of above: A 3-year deposit at year 4**	0.00				122.14			-152.20	
20										
21	Continuously compounded forward interest rate from year 4 to year 7	7.33%	<-- =LN(-I19/F19)/(B2-B3)							

The spreadsheet above shows two forward rate computations. If the interest rates are discretely compounded, then the forward rate from year 4 to year 7 is given by:

$$\textit{Discretely compounded forward rate, year 4 to 7} = \left[\frac{(1+r_7)^7}{(1+r_4)^4}\right]^{(1/3)} - 1$$

$$= \left[\frac{(1+6\%)^7}{(1+5\%)^4}\right]^{(1/3)} - 1$$

$$= \left(\frac{1.5036}{1.2155}\right)^{(1/3)} - 1 = 7.35\%$$

If the rates are continuously compounded (as in the Black model and most option calculations):

$$\textit{Continously compounded forward rate, year 4 to 7} = \left(\frac{1}{3}\right)\ln\left[\frac{e^{r_7 * 7}}{e^{r_4 * 4}}\right]$$

$$= \left(\frac{1}{3}\right)\left(\frac{1.5220}{1.2214}\right) = 7.33\%$$

To apply the forward interest rate to the example in the previous subsection, assume that the bond in question has a maturity of 2 years and a face value at maturity of 147. Then if the 2-year interest rate is $r_2 = 6\%$ and the interest rate to the option's maturity is $r_{0.5} = 4\%$, the forward price of the bond is $F =$ 133.011, as shown below:

	A	B	C
1	DETERMINING THE FORWARD PRICE OF THE BOND		
2	Bond's maturity, N	2	
3	Option maturity, T	0.5	
4	Bond maturity value	147	
5			
6	Interest rate to N	6%	
7	Interest rate to T	4%	
8			
9	Bond forward price to T	133.011	<-- =B4*EXP(-B6*B2)*EXP(B7*B3)

17.10 Summary

The Black-Scholes formula for pricing options is one of the most powerful innovations in finance. The formula is widely used both to price options and as a conceptual framework for analyzing complex securities. In this chapter we have explored the implementation of the Black-Scholes formula. Using "plain vanilla" Excel allows us to price Black-Scholes options; using VBA we are able define both the Black-Scholes price and the implied volatility for options. Finally, we showed how to use Black-Scholes to price structured products—combinations of options, stocks, and bonds.

Exercises

1. Use the Black-Scholes model to price the following:
   - A call option on a stock whose current price is 50, with exercise price $X = 50$, $T = 0.5$, $r = 10\%$, $\sigma = 25\%$.
   - A put option with the same parameters.

2. Use the data from exercise 1 and **Data|Table** to produce graphs that show:

   - The sensitivity of the Black-Scholes call price to changes in the initial stock price S.
   - The sensitivity of the Black-Scholes put price to changes in σ.
   - The sensitivity of the Black-Scholes call price to changes in the time to maturity T.
   - The sensitivity of the Black-Scholes call price to changes in the interest rate r.
   - The sensitivity of the put price to changes in the exercise price X.

3. Produce a graph comparing a call's *intrinsic value* [defined as $\max(S - X, 0)$] and its Black-Scholes price. From this graph you should be able to deduce that it is never optimal to exercise early a call priced by the Black-Scholes.

4. Produce a graph comparing a put's intrinsic value [= $\max(X - S,0)$] and its Black-Scholes price. From this graph you should be able to deduce that it may be optimal to exercise early a put priced by the Black-Scholes formula.

5. The table below gives prices for American Airlines (AMR) options on 12 July 2007. The option with exercise price X = \$27.50 is assumed to be the at-the-money option.

   a. Compute the implied volatility of each option (use the functions **CallVolatility** and **PutVolatility** defined in the chapter).

   b. Graph these volatilities. Is there a volatility "smile"?

	A	B	C	D	E	F	G
1	**AMR OPTIONS**						
2	Stock price	27.82					
3	Current date	12-Jul-07					
4	Expiration date	16-Nov-07					
5	Time to maturity,	0.35	<-- =(B4-B3)/365				
6	Interest	5%					
7							
8	**Strike**	**Call option price**	**Implied volatility**		**Strike**	**Put option price**	**Implied volatility**
9	15.0	13.50			15.0	0.15	
10	17.5	10.40			17.5	0.25	
11	20.0	8.40			20.0	0.55	
12	22.5	7.20			22.5	1.06	
13	25.0	4.90			25.0	1.90	
14	27.5	3.30			27.5	2.95	
15	30.0	2.30			30.0	4.30	
16	32.5	1.65			32.5	6.10	
17	35.0	1.00			35.0	7.40	
18	37.5	0.70			37.5	9.60	
19	40.0	0.45			40.0	12.70	
20	45.0	0.25					
21	50.0	0.05					

6. Re-examine the $X = 17.50$ call for AMR in the previous exercise.

   a. Is the call correctly priced?

   b. What price would be necessary for this call in order for the implied volatility to be 60%?

7. Use the Excel **Solver** to find the stock price for which there is the maximum difference between the Black-Scholes call option price and the option's intrinsic value. Use the following values: $S = 45$, $X = 45$, $T = 1$, $\sigma = 40\%$, $r = 8\%$.

8. As shown in this chapter, Merton (1973) shows that for the case of an asset with price S paying a continuously compounded dividend yield k, this leads to the following call option pricing formula:

$$C = Se^{-kT}N(d_1) - Xe^{-rT}N(d_2),$$

where

$$d_1 = \frac{\ln(S/X) + (r - k + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

   a. Modify the **BSCall** and **BSPut** functions defined in this chapter to fit the Merton model.

   b. Use the function to price an at-the-money option on an index whose current price is $S = 1500$, when the option's maturity $T = 1$, the dividend yield is $k = 2.2\%$, its standard deviation $\sigma = 20\%$, and the interest rate $r = 7\%$.

9. On 12 July 2007 call and put options to purchase and sell 10,000 euros at $1.37 per euro are traded on the Philadelphia options exchange. The options' expiration date is 20 December 2007. If the dollar interest rate is 5%, the euro interest rate is 4.5% and the volatility of the euro is 6%, what should be the price of a call and a put?

10. Note that you can use the Black-Scholes formula to calculate the call option premium as a percentage of the exercise price in terms of *S/X*:

$$C = SN(d_1) - Xe^{-rT}N(d_2) \Rightarrow \frac{C}{X} = \frac{S}{X}N(d_1) - e^{-rT}N(d_2)$$

where

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

Implement this in a spreadsheet.

11. Note that you can also calculate the Black-Scholes put option premium as a percentage of the exercise price in terms of S/X:

$$P = -SN(-d_1) + Xe^{-rT}N(-d_2) \Rightarrow \frac{P}{X} = e^{-rT}N(-d_2) - \frac{S}{X}N(-d_1)$$

where

$$d_1 = \frac{\ln(S/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

Implement this in a spreadsheet. Find the ratio of *S/X* for which *C/X* and *P/X* cross when $T = 0.5$, $\sigma = 25\%$, $r = 10\%$. (You can use a graph or you can use Excel's Solver.) Note that this crossing point is affected by the interest rate and the option maturity, but not by σ.

12. Consider a structured security of the following type: The purchaser invests $1,000 and in three years gets back the initial investment plus 95% of the increase in a market index whose current price is 100. The interest rate is 6% per year, continuously compounded. Assuming the security is fairly priced, what is the implied volatility of the market index?

18 Option Greeks

18.1 Overview

In this chapter we discuss the sensitivities of the Black-Scholes formula to its various parameters. The "Greeks," as they are called (because of the Greek letters used to denote most of them), are the partial derivatives of the Black-Scholes formula with respect to its arguments. They can be thought of as giving a measure of the riskiness of an option:

• Delta, denoted by Δ, is the partial derivative of the option price with respect to the price of the underlying stock price: $\Delta_{Call} = \frac{\partial \text{Call}}{\partial S}, \Delta_{Put} = \frac{\partial \text{Put}}{\partial S}$. Delta, Δ, can be thought of as a measure of the variability of the option's price when the price of the underlying changes.

• Gamma, Γ, is the second derivative of the option's price with respect to the underlying stock. Gamma gives the convexity of the option price with respect to the stock price. For options priced by the Black-Scholes formula, the call and put have the same gamma: $\Gamma_{Call} = \frac{\partial^2 \text{Call}}{\partial S^2} = \Gamma_{Put} = \frac{\partial^2 \text{Put}}{\partial S^2}$.

• Vega is the sensitivity of the option price to the standard deviation of the underlying stock's return σ: For no obvious reason, the Greek letter kappa, κ, is sometimes used to denote vega. Given the Black-Scholes formula, calls and puts have the same vega: $\kappa = \frac{\partial \text{Call}}{\partial \sigma} = \frac{\partial \text{Put}}{\partial \sigma}$.

• Theta, θ, is change in the option's value as the time to maturity decreases. We generally expect that options become less valuable with the passage of time (though this turns out not to be always true). Writing T as the option's remaining time to maturity, we set theta equal to the negative of the derivative of the option price with respect to T: $\theta_{Call} = -\frac{\partial \text{Call}}{\partial T}, \theta_{Put} = -\frac{\partial \text{Put}}{\partial T}$.

• Rho, ρ, measures the interest rate sensitivity of an option: $\rho_{Call} = -\frac{\partial \text{Call}}{\partial r}, \rho_{Put} = -\frac{\partial \text{Put}}{\partial r}$.

In this chapter we show you how to measure an option's Greeks and how to use them in hedging. For generality we illustrate using the Merton model (Chapter 17, section 6), an extended version of the Black-Scholes formula which applies to both stocks paying a continuous dividend or to currencies.

18.2 Defining and Computing the Greeks

The "Greeks" are the sensitivities of an option price with respect to certain of its variables. In the table below we set out the Greeks for options defined on an underlying which pays a continuous dividend. As discussed in section 17.6, such options are priced using the Merton model. Of course, the standard Black-Scholes model is obtained from the Merton model by setting the dividend yield $k = 0$. Currency options can be priced by the Merton formula by setting S = current exchange rate, X = option exercise exchange rate, r = domestic interest rate, and k = foreign interest rate.

The Merton version of the Black-Scholes formula is given by:

$$C = Se^{-kT}N(d_1) - Xe^{-rT}N(d_2)$$
$$P = -Se^{-kT}N(-d_1) + Xe^{-rT}N(-d_2)$$

where

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + (r - k + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

The table on page 469 gives the Greeks for this formula. The appendix to this chapter gives the VBA implementations of these functions.

Black-Scholes Greeks

	Measures	Call	Put
Delta, written as either Δ or δ	Price sensitivity of option $\frac{\partial V}{\partial S}$	$\Delta_{Call} = e^{-kT}N(d_1)$	$\Delta_{Put} = e^{-kT}(N(d_1) - 1)$ $= -e^{kT}N(-d_1)$
Gamma, written as Γ	Second-order price sensitivity $\frac{\partial^2 V}{\partial S^2}$. The option's convexity with respect to underlying price.	$\frac{e^{-kT}N'(d_1)}{S\sigma\sqrt{T}} = \frac{e^{(d_1)^2/2-kT}}{S\sigma\sqrt{2T\pi}}$	
Vega, no Greek letter, though sometimes the Greek kappa κ is used	Sensitivity to volatility $\frac{\partial V}{\partial \sigma}$	$Se^{-kT}N'(d_1)\sqrt{T} = \frac{S\sqrt{T}\,e^{-(d_1)^2/2-kT}}{\sqrt{2\pi}}$	
Theta, written as θ	Time sensitivity $-\frac{\partial V}{\partial T}$	$-\frac{Se^{-kT}N'(d_1)\sigma}{2\sqrt{T}} + kSe^{-kT}N(d_1)$ $-rXe^{-rT}N(d_2)$	$-\frac{Se^{-kT}N'(d_1)\sigma}{2\sqrt{T}} - kSe^{-kT}N(-d_1)$ $+rXe^{-rT}N(-d_2)$
Rho, ρ	Interest rate sensitivity	$XTe^{-rT}N(d_2)$	$-XTe^{-rT}N(-d_2)$

Reminders: $d_1 = \frac{\ln(S/X) + \left(r - k + \frac{\sigma^2}{2}\right)T}{\sigma T}$, $d_2 = d_1 - \sigma\sqrt{T}$, $N'(x) = \frac{1}{\sqrt{2\pi}}e^{\left(-x^2/2\right)}$

The Greeks are implemented in the spreadsheet below, which shows both the brute-force calculation of each Greek as well as a VBA function implementation.

	A	B	C
1	**BLACK-SCHOLES GREEKS Uses Merton model for a continuously dividend-paying stock**		
2	S	100	Current stock price
3	X	90	Exercise price
4	T	0.5	Time to maturity of option (in years)
5	r	6.00%	Risk-free rate of interest
6	k	2.00%	Dividend yield
7	Sigma	35%	Stock volatility
8			
9	d_1	0.6303	<-- =(LN(B2/B3)+(B5-B6+0.5*B7^2)*B4)/(B7*SQRT(B4))
10	d_2	0.3828	<-- d_1-sigma*SQRT(T)
11			
12	N(d_1)	0.7357	<-- Uses formula NormSDist(d_1)
13	N(d_2)	0.6491	<-- Uses formula NormSDist(d_2)
14			
15	Call price	16.1531	<-- =B2*EXP(-B6*B4)*B12-B3*EXP(-B5*B4)*B13
16		16.1531	<-- =bsmertoncall(B2,B3,B4,B5,B6,B7)
17	Put price	4.4882	<-- =B3*EXP(-B5*B4)*NORMSDIST(-B10)-B2*EXP(-B6*B4)*NORMSDIST(-B9)
18		4.4882	<-- =bsmertonput(B2,B3,B4,B5,B6,B7)
19			
20	**Call Greeks, brute force**		
21	Delta	0.7284	<-- =EXP(-B6*B4)*NORMSDIST(B9)
22	Gamma	0.0131	<-- =EXP(-(B9^2)/2-B6*B4)/(B2*B7*SQRT(2*B4*PI()))
23	Vega	22.8976	<-- =B2*SQRT(B4)*EXP(-(B9^2)/2)*EXP(-B6*B4)/SQRT(2*PI())
24	Theta	-9.9587	<-- =-B2*EXP(-(B9^2)/2-B6*B4)*B7/SQRT(8*B4*PI())+B6*B2*EXP(-B6*B4)*B12-B5*B3*EXP(-B5*B4)*B13
25	Rho	28.3446	<-- =B3*B4*EXP(-B5*B4)*NORMSDIST(B10)
26			
27	**Call Greeks, VBA formulas**		
28	Delta	0.7284	<-- =deltacall(B2,B3,B4,B5,B6,B7)
29	Gamma	0.0131	<-- =optiongamma(B2,B3,B4,B5,B6,B7)
30	Vega	22.8976	<-- =vega(B2,B3,B4,B5,B6,B7)
31	Theta	-9.9587	<-- =Thetacall(B2,B3,B4,B5,B6,B7)
32	Rho	28.3446	<-- =rhocall(B2,B3,B4,B5,B6,B7)

Greek calculations for puts are given below:

	E	F	G
20			**Put Greeks, brute force**
21	Delta	-0.2616	<-- =-EXP(-B6*B4)*NORMSDIST(-B9)
22	Gamma	0.0131	<-- =EXP(-(B9^2)/2-B6*B4)/(B2*B7*SQRT(2*B4*PI()))
23	Vega	22.8976	<-- =B2*EXP(-(B9^2)/2-B6*B4)*SQRT(B4)/SQRT(2*PI())
24	Theta	-6.6984	<-- =-B2*EXP(-(B9^2)/2-B6*B4)*B7/SQRT(8*B4*PI())-B6*B2*EXP(-B6*B4)*(1-B12)+B5*B3*EXP(-B5*B4)*(1-B13)
25	Rho	-15.3255	<-- =-B3*B4*EXP(-B5*B4)*NORMSDIST(-B10)
26			
27			**Put Greeks, VBA formulas**
28	Delta	-0.2616	<-- =deltaput(B2,B3,B4,B5,B6,B7)
29	Gamma	0.0131	<-- =optiongamma(B2,B3,B4,B5,B6,B7)
30	Vega	22.8976	<-- =vega(B2,B3,B4,B5,B6,B7)
31	Theta	-6.6984	<-- =Thetaput(B2,B3,B4,B5,B6,B7)
32	Rho	-15.3255	<-- =rhoput(B2,B3,B4,B5,B6,B7)

Excel can be used to examine the sensitivities of the Greeks to various parameters. We give some examples below. Figures 18.1 and 18.2 show the deltas as functions of the stock price and as functions of the moneyness of the call option.

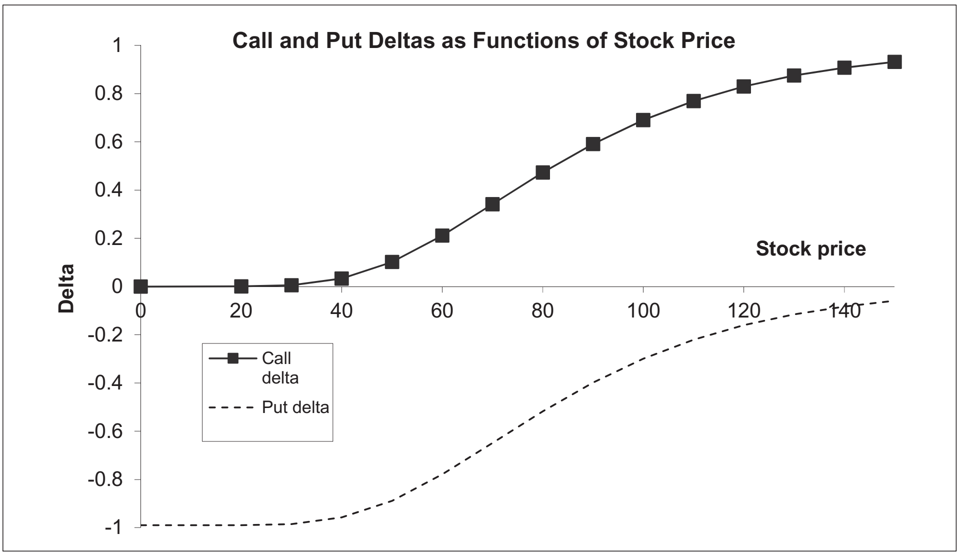


Figure 18.1
As a call or a put becomes more in-the-money, the delta tends toward +1 for a call and –1 for a put. Essentially, the call or put price moves in tandem with the underlying stock price. An extremely out-of-the-money put or call has a delta = 0.

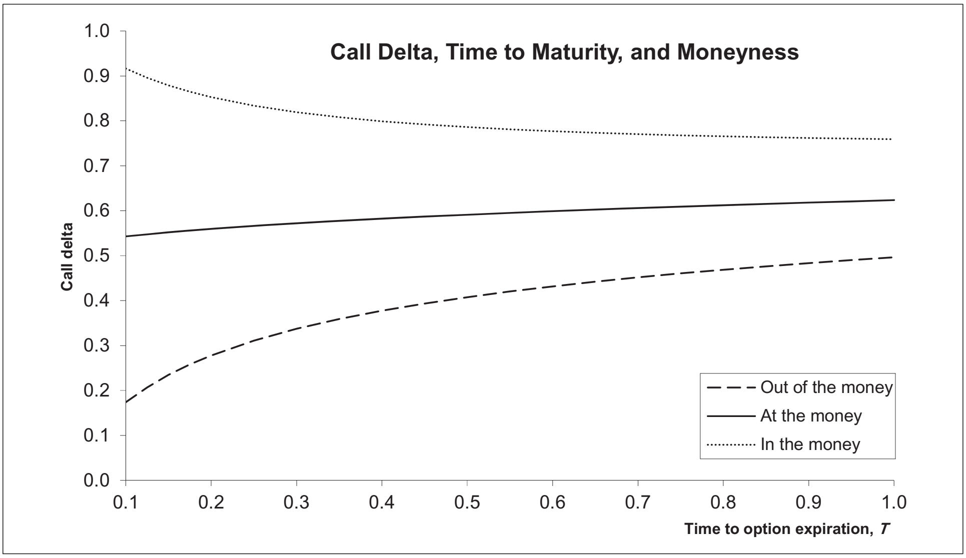


Figure 18.2
As the option's maturity T increases, the delta of an at-the-money and an out-of-the-money call increases, whereas the delta of an in-the-money call decreases.

Figures 18.3 and 18.4 show the theta of a call as a function of the stock price and the time to option expiration:

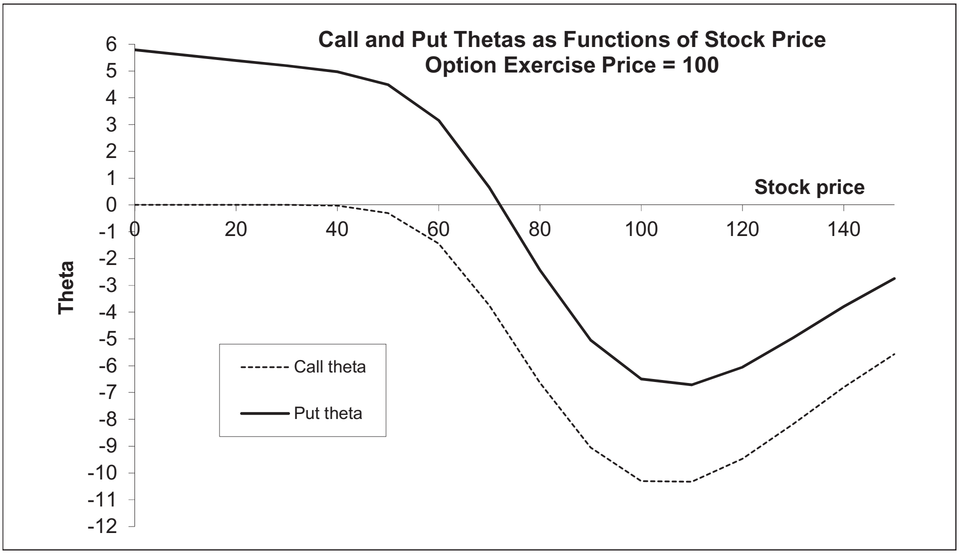


Figure 18.3
Very in-the-money puts can have a positive theta, meaning that as the time to maturity gets shorter, the put gains in value. Other than this case, options generally have a negative theta, meaning that they lose value as the time to maturity decreases.

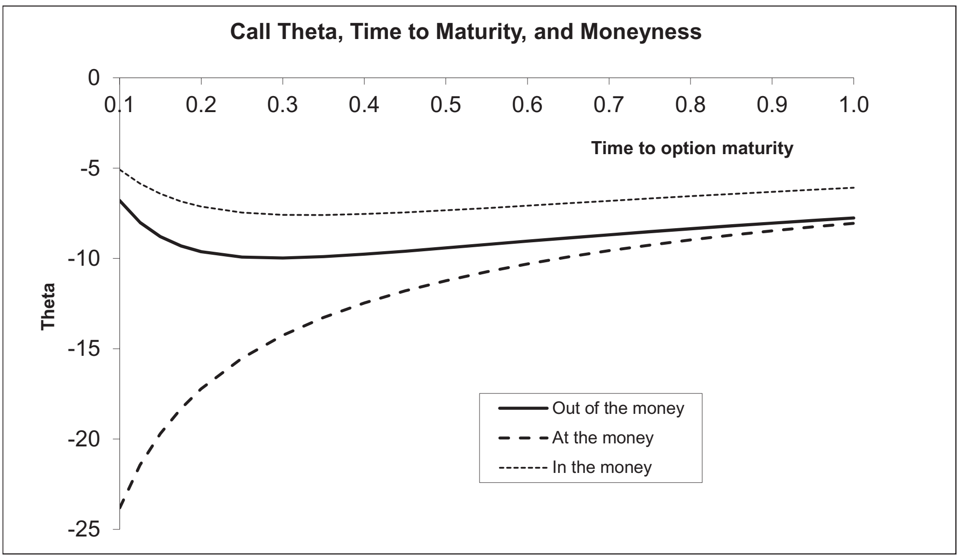


Figure 18.4
Calls always have negative theta (meaning that they lose value as the time to maturity decreases). However, the rate at which they lose value varies with the moneyness of the call.

18.3 Delta Hedging a Call[1]

Delta hedging is a fundamental technique in option pricing. The idea is to replicate an option by a portfolio of stocks and bonds, with the portfolio proportions determined by the Black-Scholes formula.

Suppose we decide to replicate an at-the-money European call option which has 12 weeks to run until expiration. The stock on which the option is written has $S_0 = \$40$ and exercise price $X = \$35$, the interest rate is $r = 4\%$, and the stock's volatility is $\sigma = 25\%$. The Black-Scholes price of this option is 5.44:

	A	B	C	D	E	F	G	H
1	**DELTA HEDGING A CALL**							
2	S, current stock price	40.00		Initial pricing of call using Black-Scholes formula				
3	X, exercise	35.00						
4	r, interest rate	2.00%						
5	k, dividend yield	0.00%						
6	T, expiration	0.2308	<-- =12/52					
7	Sigma	25%						
8								
9	BS value	5.44	<-- =bsmertoncall(B2,B3,B6,B4,B5,B7)					
10								
11				**Hedging portfolio**				
12	**Weeks until expiration**	**Time until expiration**	**Stock price**	**Stock = =C13*deltacall(C13,$B$3, B13,$B$4,0,$B$7)**	**Investment in stock**	**Bond**	**Porfolio value**	**Portfolio cash flow**
13	12	0.2308	40.000	35.48		-30.04	5.44	5.44
14	11	0.2115	38.042	30.19	-3.5498	-26.50	3.69	0.00
15	10	0.1923	38.884	33.17	2.3135	-28.82	4.35	0.00
16	9	0.1731	38.568	32.62	-0.2814	-28.55	4.07	0.00
17	8	0.1538	38.501	32.87	0.3044	-28.87	4.00	0.00
18	7	0.1346	37.768	30.87	-1.3759	-27.50	3.36	0.00
19	6	0.1154	39.383	36.54	4.3488	-31.86	4.67	0.00
20	5	0.0962	40.406	39.29	1.8031	-33.68	5.61	0.00
21	4	0.0769	39.626	38.34	-0.1859	-33.51	4.84	0.00
22	3	0.0577	39.216	38.20	0.2496	-33.77	4.43	0.00
23	2	0.0385	39.745	39.58	0.8646	-34.65	4.93	0.00
24	1	0.0192	41.522	41.52	0.1756	-34.83	6.69	0.00
25	0	0.0000	43.199				8.35	
26								
27	Hedged position payoff	8.35	<-- =G25					
28	Actual call payoff	8.20	<-- =MAX(C25-B3,0)					
29								
30								
31	**Formulas**							
32	Cell D14: =C14*deltacall(C14,$B$3,B14,$B$4,0,$B$7)							
33	Cell E14: =D14-D13*C14/C13							
34	Cell F14: =F13*EXP($B$4/52)-E14							
35	Cell G14: =D14+F14							
36	Cell H14: =(D13*C14/C13-D14)+F13*EXP($B$4*(B13-B14))-F14							

At initial date, the stock and bond positions are set using the Black-Scholes formula: Stock = SN(d_1), Bond = -X*exp(-rT)N(d_2).

At each subsequent date t, the stock position is adjusted to $S_t*\Delta_{call}$. The bond position is adjusted so that the net cash flow of the portfolio is zero.

At the final date, the stock and bond portfolios are liquidated.

1. This topic is discussed again in Chapter 29.

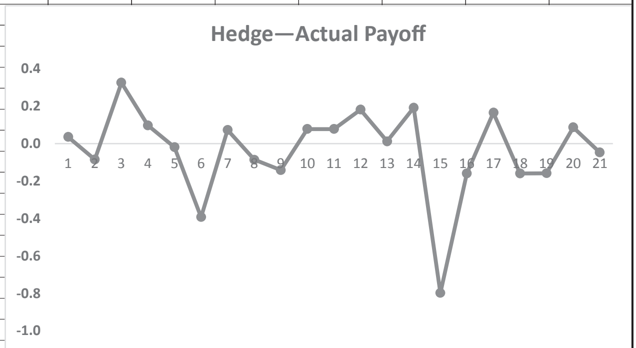
Note that we use the formula **BSMertoncall** but with the dividend yield $k = 0\%$, so that this is, in effect, a regular BS call option.

In the spreadsheet above, we create this option by replicating, on a week-to-week basis, the BS option pricing formula using delta hedging.

- At the beginning, 12 weeks before the option's expiration, we determine our stock/bond portfolio according to the formula $Call = SN(d_1) - Xe^{-rT}N(d_2)$, so that we have a dollar amount $SN(d_1)$ of shares in the portfolio and have borrowing of $Xe^{-rT}N(d_2)$. Having determined the portfolio holdings at the beginning of the 12-week period, we now determine our portfolio holdings for each of the successive weeks as follows:
- In each successive week we set the stock holdings in the portfolio according to the formula $SN(d_1)$, but we set the portfolio borrowing so that the *net cash flow of the portfolio* is zero. Note that $SN(d_1) = S\Delta_{Call}$, hence the name "delta hedging."
- At the end of the 12-week period, we liquidate the portfolio.

The delta hedge would be perfect if we rebalanced our portfolio continuously. However, here we have rebalanced only weekly. Had we a perfect hedge, the portfolio would have paid off $\max[S_{Terminal} - X, 0]$ (cell B27); the actual hedge payoff (cell B28) is slightly different. Using the technique of **Data Table** on a blank cell explained in Chapter 31, we replicate this simulation to check the deviation between the desired payoff and the hedge position payoff:

	K	L	M	N	O	P	Q	R	S	T	U	V
12	Simulation	Delta hedge payoff	Max(S_T-X,0)	Hedge—actual payoff								
13		5.4024	5.3669	0.0355	<-- =L13-M13, data table header							
14	1	3.4743	3.5591	-0.0848								
15	2	0.3252	0.0000	0.3252								
16	3	0.0971	0.0000	0.0971								
17	4	-0.0193	0.0000	-0.0193								
18	5	11.8112	12.2037	-0.3925								
19	6	3.5610	3.4877	0.0733								
20	7	10.6744	10.7604	-0.0860								
21	8	8.6602	8.8020	-0.1417								
22	9	10.3202	10.2422	0.0780								
23	10	4.3520	4.2742	0.0777								
24	11	2.3202	2.1388	0.1814								
25	12	0.0123	0.0000	0.0123								
26	13	6.4462	6.2541	0.1920								
27	14	2.7854	3.5831	-0.7978								
28	15	10.6339	10.7932	-0.1592								
29	16	3.1851	3.0200	0.1651								
30	17	1.8986	2.0582	-0.1596								
31	18	13.6581	13.8167	-0.1586								
32	19	7.9048	7.8179	0.0869								
33	20	7.5720	7.6190	-0.0471								



18.4 Hedging a Collar

A collar is an option strategy designed to protect the holder of a package of shares against possible price losses. The usual collar is a combination of a written call plus a purchased put, designed so that the net cost of the position is zero. Thus the collar provides costless protection to its holder. Here's an example: On 1 January 2008, a bank's client holds 5,000,000 shares of XYZ Corp. Each share is currently worth \$55. Because the stock is currently restricted, the client cannot sell the shares until 1 year from now. However, he is worried that the stock price will decline, and hence he desires to purchase a collar.

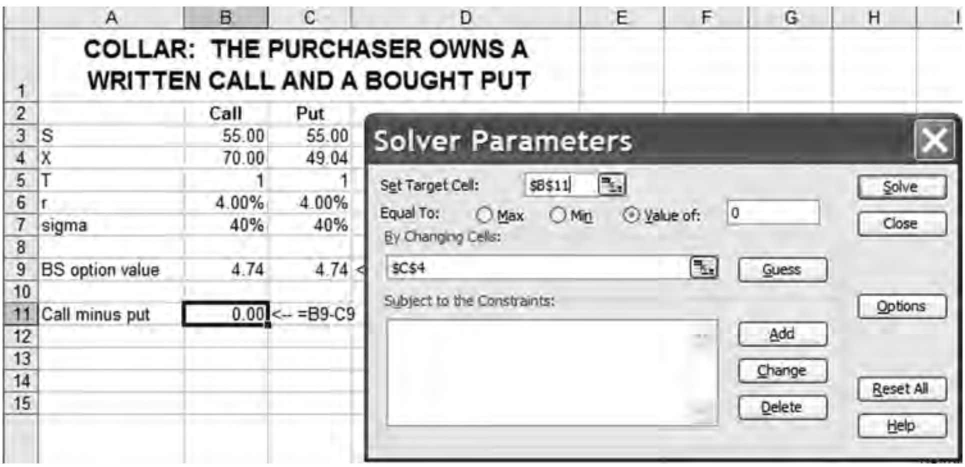
The client asks an investment bank to design the following package:

- He wants to buy a put on the shares with T = 1 year and exercise price x_{Put} = \$49.04
- He wants to write a call on the shares with T = 1 year and exercise price X_{Call} = \$70.00.

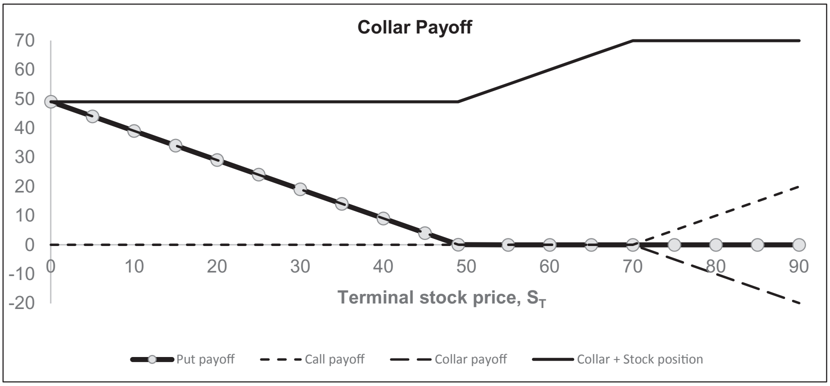
The exercise prices have been set so that the Black-Scholes value of the call and the put are equal:

	A	B	C	D
1	**COLLAR: THE PURCHASER OWNS A WRITTEN CALL AND A BOUGHT PUT**			
2		**Call**	**Put**	
3	S	55.00	55.00	
4	X	70.00	49.04	
5	T	1	1	
6	r, interest	4.00%	4.00%	
7	k, dividend yield	0.00%	0.00%	
8	Sigma	40%	40%	
9				
10	BS option value	4.74	4.74	<-- =bsmertonput(C3,C4,C5,C6,C7,C8)
11				
12	Call minus put	0.00	<-- =B10-C10	
13				
14				
15				=bsmertoncall(B3,B4,B5,B6,B7,B8)

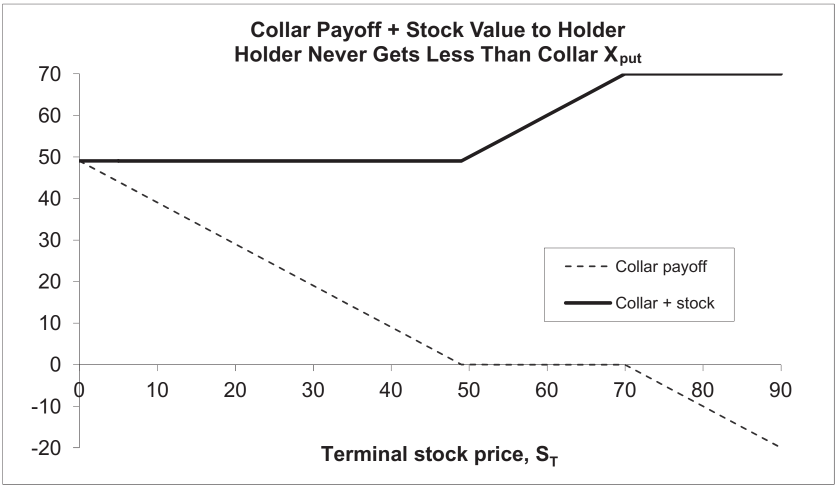
Given the $X_{Call} = 70$ for the call, the put exercise price was determined using **Solver**:



The point of the collar is to give the purchaser upside potential with limited downside risk. In the example above, for example, the terminal payoffs of the put, the call, and the collar are given by:



In addition to his collar, the client has a portfolio of the shares. The payoffs to the holder of the collar plus the shares are never less than $49. This is, of course, the protection that the client was seeking.



A Slightly Longer Story

Even though the Black-Scholes value of the collar is initially zero, actually the investment bank sold the collar to the client for $5. There are several reasons why the client might want to pay this:

- Perhaps there is low liquidity in the options (this is often the case in longer-term options), so that the bank is actually supplying a valuable liquidity service.
- It might be that the options do not actually exist—either because the particular long-term options in question are not marketed or perhaps because there are no options on the particular underlying stock (this is often the case for specific portfolios). In this case the bank is actually creating the options underlying the collar by creating an appropriate portfolio of stocks and bonds and by changing the portfolio proportions over time (see next subsection). The creation and constant monitoring of this portfolio is a service worth paying for.

Delta Hedging the Collar: The Bank's Problem

The client for the collar is short a call and long a put. The bank wants to make a similar investment, so that it will parallel the client's portfolio and have the money to pay off the client at the maturity of his collar. In terms of the

Black-Scholes formula this turns out to mean that the net position of the bank is a short stock financed by a bond investment:

$$\underbrace{-[SN(d_1[X_{\text{Call}}]) - X_{\text{Call}}e^{-rT}N(d_2[X_{\text{Call}}])]}_{\substack{\uparrow \\ \text{Short call}}}$$

$$\underbrace{-SN(-d_1[X_{\text{Put}}]) + X_{\text{Put}}e^{-rT}N(-d_2[X_{\text{Put}}])}_{\substack{\uparrow \\ \text{Long put}}}$$

$$= \underbrace{-S(N(d_1[X_{\text{Call}}]) + N(-d_1[X_{\text{Put}}]))}_{\substack{\uparrow \\ \text{Short stock position}}}$$

$$\underbrace{+e^{-rT}[X_{\text{Call}}N(d_2[X_{\text{Call}}]) + X_{\text{Put}}N(-d_2[X_{\text{Put}}])]}_{\substack{\uparrow \\ \text{Long bond position}}}$$

We rewrite this in terms of Greeks:

$$\underbrace{-S(N(d_1[X_{\text{Call}}]) + N(-d_1[X_{\text{Put}}]))}_{\substack{\uparrow \\ \text{Short stock position}}}$$

$$\underbrace{-e^{-rT}[X_{\text{Call}}N(d_2[X_{\text{Call}}]) + X_{\text{Put}}N(-d_2[X_{\text{Put}}])]}_{\substack{\uparrow \\ \text{Long bond position}}}$$

$$= S(-\Delta_{\text{Call}}(X_{\text{Call}}) - \Delta_{\text{Put}}(X_{\text{Put}})) + e^{-rT}\begin{bmatrix} X_{\text{Call}}N(d_2[X_{\text{Call}}]) \\ + X_{\text{Put}}N(-d_2[X_{\text{Put}}]) \end{bmatrix}$$

Here's a run of a simulated position over the course of a year. In this simulation the position is updated every $\Delta t = 0.05$; assuming 250 trading days in a year, this is approximately every 12 days.

	A	B	C	D	E	F	G
1	**DELTA HEDGING A COLLAR**						
2	S	55.00					
3	X_{call}	70.00					
4	X_{put}	49.04					
5	r	4.00%					
6	k, dividend yield	0.00					
7	Sigma	40%					
8				=(C10*B11/B10-C11)+D10*EXP($B$5*(A10-A11))			
9	**Time until expiration**	**Stock price**	**Stock =-B10*(deltacall(B10,$B$3,A10,$B$5,$B$6,$B$7)-deltaput(B10,$B$4,A10,$B$5,$B$6,$B$7))**	**Bond**	**Portfolio value**	**Portfolio cash flow**	
10	1.00	55.00	-36.28	36.28	0.00	0.00	
11	0.95	62.26	-42.31	37.59	-4.72	0.00	
12	0.90	60.06	-39.59	36.44	-3.15	0.00	
13	0.85	70.57	-50.85	40.85	-10.00	0.00	
14	0.80	68.88	-48.21	39.51	-8.70	0.00	
15	0.75	79.24	-61.94	46.06	-15.88	0.00	
16	0.70	79.96	-62.78	46.43	-16.34	0.00	
17	0.65	84.32	-69.21	49.54	-19.67	0.00	
18	0.60	76.84	-57.58	44.15	-13.44	0.00	
19	0.55	73.80	-52.43	41.36	-11.07	0.00	
20	0.50	89.93	-78.52	56.07	-22.45	0.00	
21	0.45	91.19	-81.01	57.57	-23.43	0.00	
22	0.40	84.63	-70.73	53.24	-17.49	0.00	
23	0.35	78.24	-59.24	47.20	-12.05	0.00	
24	0.30	88.04	-78.30	58.93	-19.37	0.00	
25	0.25	99.20	-96.32	67.14	-29.18	0.00	
26	0.20	95.41	-92.46	67.09	-25.37	0.00	
27	0.15	121.46	-121.44	70.98	-50.47	0.00	
28	0.10	136.73	-136.73	71.13	-65.60	0.00	
29	0.05	130.28	-130.28	71.28	-59.00	0.00	
30	0.00	139.32			-67.90		
31					↑		
32	**Check: Collar payoff to client at time 0**				=C29*B30/B29+D29*EXP($B$5		
33	Short call payoff	-69.32	<-- =-MAX(B30-B3,0)		*(A29-A30))		
34	Long put payoff	0.00	<-- =MAX(B4-B30,0)				
35	Total	-69.32	<-- =SUM(B33:B34)				
36							
37	**Payoff to bank from delta hedge**			Formula in cell F11 = =(C10*B11/B10-C11)-(D11-D10*EXP($B$5*(A10-A11)))			
38		-67.90	<-- =E30				
39							
40	**Terminal cash flow to bank**	1.42	<-- =-B35+B38				

Here's what happens in this spreadsheet:

- The initial (row 10) stock and bond positions are determined by the Black-Scholes formula. The stock position is $-S(\Delta_{Call}(X_{Call}) - \Delta_{Put}(x_{Put}))$ and the bond position is $e^{-rT}[X_{Call}N(d_2[X_{Call}])] + X_{Put}N(-d_2[X_{Put}])$. Not surprisingly, the net value of this portfolio is zero—this is the way we determined the collar X_{Call} and X_{Put}.
- In each of the subsequent rows, the stock position is determined by the Black-Scholes formula, and the bond position is determined so that the net cash flow of the position is zero:

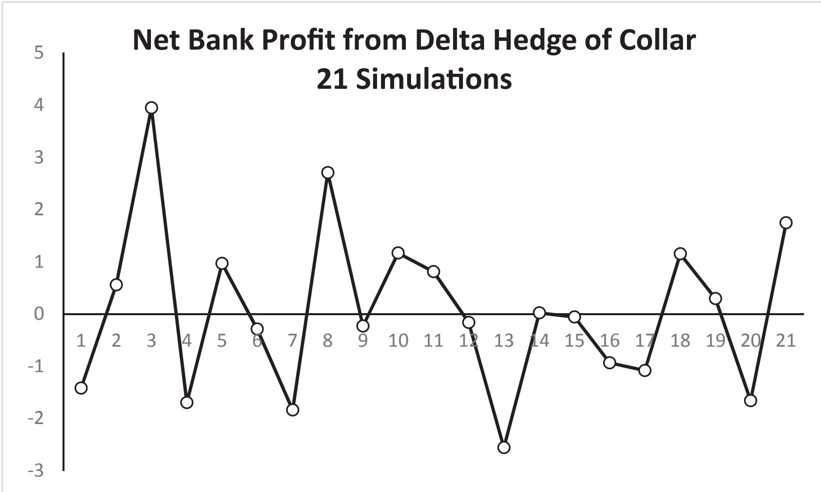
$$Bonds = \underbrace{Stock\_position_{t-1} * \frac{Stock\_price_t}{Stock\_price_{t-1}} - \underbrace{Stock\_position_t}_{\substack{\uparrow \\ \text{Determined by} \\ \text{Black-Scholes}}}}_{\substack{\uparrow \\ \text{Cash flow into stocks}}} + \underbrace{Bond\_position_{t-1} * \exp(r * \Delta t)}_{\substack{\uparrow \\ \text{Value today of } t-1 \text{ bond position}}}$$

- At the terminal date (row 30), the portfolio is liquidated:

$$Stock\_position_{Terminal} = Stock\_position_{Previous} * \frac{Stockprice_{Terminal}}{Stockprice_{Previous}} + Bond\_position_{Previous} * \exp(r * \Delta t)$$

- At the terminal date, the purchaser of the collar collects on his short position in the call and his long position in the put (cell B35). The bank collects its position value (cells E30 or B38). The bank's net cash flow on termination is the difference between these two (cell B40).

We can use **Data Table** on a blank cell to show many simulations. For details see the spreadsheet for this chapter. Below we show a graph that gives the bank's net profit in 21 simulations of the delta hedging strategy for the collar.



Making the Collar Gamma Neutral

As the option gets closer to expiration, the hedge position can become sensitive to small changes in the stock price, meaning that the gamma of the collar can grow enormously.

There are two solutions to moderate the gamma of the collar:

- We can increase our hedge frequency as we get closer to the collar expiration date.
- We can change our hedge strategy in order to temper the hedge gamma, as we get closer to the option maturity.

Below we explore both solutions.

Increasing the Hedge Frequency

The main problem of the hedge seems to be for the dates close to the end of the expiration period. Since the initial options have maturity $T = 1$, this means that we have to be very careful during the last two months of the hedging period. Delta hedging the position more often may work, though it is easy to come up with counter-examples:

	A	B	C	D	E	F
1	**MODERATING THE COLLAR GAMMA** **This example starts with T = 0.20 and hedges every Delta_t = 0.01**					
2	S	55.00				
3	X_{call}	70.00				
4	X_{put}	49.04				
5	r	4.00%				
6	k, dividend yield	2.00%				
7	Sigma	40%				
8						
9	**Time until expiration**	**Stock price**	**Stock =-B10*(deltacall(B10,$B$3,A10,$B$5,$B$6,$B$7)-deltaput(B10,$B$4,A10,$B$5,$B$6,$B$7))**	**Bond**	**Porfolio value**	**Portfolio cash flow**
10	0.20	55.00	-18.29	18.29	0.00	0.00
11	0.19	53.62	-18.43	18.90	0.47	0.00
12	0.18	54.12	-17.55	17.85	0.30	0.00
13	0.17	54.65	-16.60	16.74	0.14	0.00
14	0.16	50.15	-21.44	22.96	1.51	0.00
15	0.15	51.18	-19.49	20.58	1.08	0.00
16	0.14	50.05	-21.20	22.72	1.52	0.00
17	0.13	51.20	-18.76	19.80	1.04	0.00
18	0.12	50.71	-19.42	20.65	1.23	0.00
19	0.11	48.61	-24.11	26.15	2.04	0.00
20	0.10	50.83	-18.55	19.49	0.95	0.00
21	0.09	47.75	-26.65	28.73	2.08	0.00
22	0.08	44.94	-34.04	37.70	3.66	0.00
23	0.07	45.97	-32.46	35.36	2.90	0.00
24	0.06	41.06	-39.37	45.75	6.38	0.00
25	0.05	40.60	-39.75	46.59	6.84	0.00
26	0.04	38.35	-38.27	47.33	9.06	0.00
27	0.03	36.60	-36.58	47.40	10.82	0.00
28	0.02	36.28	-36.27	47.43	11.16	0.00
29	0.01	35.73	-35.72	47.45	11.73	0.00
30	0.00	36.36			11.11	<-- =C29*
31						
32	**Check: Collar payoff to client at time 0**					
33	Short call payoff	0.00	<-- =-MAX(B30-B3,0)			
34	Long put payoff	12.68	<-- =MAX(B4-B30,0)			
35	Total	12.68	<-- =SUM(B33:B34)			
36						
37	**Payoff to bank from delta hedge**					
38		11.11	<-- =E30			
39						
40	**Terminal cash flow to bank**	-1.56	<-- =-B35+B38			

Repeated simulation of this hedge shows that it works quite well.

Making the Hedge Gamma Neutral

Another strategy is to add another asset to the hedge position, in an effort to neutralize the gamma. In the example below we have added an out-of-the-money put to the position to neutralize the large call gamma:

	A	B	C	D	E
1	**COLLAR HEDGE, DELTA & GAMMA in this example we costlessly neutralize a large call gamma**				
2		**Call**	**Put**	**Another put**	
3	S	48.00	48.00	48.00	
4	X	70.00	49.04	35.00	
5	r	5.00%	5.00%	5.00%	
6	k, dividend yield	0.00%	0.00%	0.00%	
7	T	0.0200	0.0200	0.0200	
8	Sigma	40.00%	40.00%	40.00%	
9					
10	Option prices	0.00	1.66	0.00	
11					
12	Delta	0.0000	-0.6304	0.0000	<-- =deltaput(D3,D4,D7,D5,D6,D8)
13	Gamma	494,472,087	0	1,118,872	<-- =gamma(D3,D4,D7,D5,D6,D8)
14					
15					
16	**Bank position: short call with X = 70.00 + long put with X = 49.04 + put with X = 35.00**				
17	Call, X = 70.00	-1			
18	Put, X = 49.04	1			
19	Put, X = 35.00	441.938			
20					
21	Position delta	-0.6304	<-- {=SUMPRODUCT(TRANSPOSE(B17:B19),B12:D12)}		
22	Position gamma	0.1553	<-- {=SUMPRODUCT(TRANSPOSE(B17:B19),B13:D13)}		
23					
24	Position cost				
25	Without second put	1.6604	<-- =B17*B10+B18*C10		
26	With second put	1.6604	<-- =B17*B10+B18*C10+B19*D10		
27					
28	Traditional collar delta	-0.6304	<-- =-B12+C12		

This can be done at very little cost, since the put in question is almost costless (cell D10). Of course, it may not always be possible to costlessly neutralize the gamma. In this case we will have to make some compromises.

18.5 Summary

In this chapter we have explored the sensitivities of the option pricing formula to its various parameters. Using these Greeks, we have delved into the intricacies of delta hedging, a useful technique for replicating an option position with a combination of stocks and bonds. The interested reader should know that there is much more which can be said about this topic. Good starting places for further reading are Hull (2006) and Taleb (1997). An extensive collection of option pricing formulas including Greeks can be found in Haug (2006).

Exercises

1. Produce a graph similar to the second panel of Figure 18.2 for puts.
2. Figure 18.4 shows the call theta as a function of time to maturity. Produce a similar graph for puts.
3. Although θ is generally negative, there are cases (typically of high interest rates) where it can be positive:

   - An in-the-money put with a high interest rate
   - An in-the-money call on a currency which has a high interest rate (or—equivalently—an in-the-money call on a stock with a very high dividend payout rate).

   Find two examples.

Appendix: VBA for Greeks

The VBA for the Greeks used in this chapter:

Black-Scholes Functions

Throughout the chapter we use the Merton version of the Black-Scholes formula for pricing options with a continuous dividend yield (see Chapter 17 for details). The VBA connected with this model is given below.

```
Function dOne(stock, exercise, time, _
interest, divyield, sigma)
   dOne = (Log(stock / exercise) + _
   (interest - divyield) * time) / _
   (sigma * Sqr(time)) + 0.5 * sigma * _
   Sqr(time)
End Function
```

```
Function dTwo(stock, exercise, time, _
interest, divyield, sigma)
   dTwo = dOne(stock, exercise, time, _
   interest, divyield, sigma) - sigma * _
   Sqr(time)
End Function

Function BSMertonCall(stock, exercise, time, _
interest, divyield, sigma)
   BSMertonCall = stock * Exp(-divyield * _
   time) * Application.NormSDist _
   (dOne(stock, exercise, time, _
   interest, divyield, sigma)) - exercise * _
   Exp(-time * interest) * Application.NormSDist _
   (dTwo(stock, exercise, time, interest, _
   divyield, sigma))
End Function

'Put pricing function uses put-call parity theorem
Function BSMertonPut(stock, exercise, time, _
interest, divyield, sigma)
   BSMertonPut = BSMertonCall(stock, exercise, _
   time, interest, divyield, sigma) + _
   exercise * Exp(-interest * time) - _
   stock * Exp(-divyield * time)
End Function
```

Defining the Normal Distribution

The option pricing functions above use the old version of the Excel function **NormSDist**. In Excel 2010 and onward, there is another function **Norm.S.Dist(x,False/True)**. If the second parameter in this function is set to **False**, it computes the normal density; with the parameter set to **True**, it computes the normal distribution function.[2]

The VBA writing of these functions is **Application.Norm_S_Dist(x,0 or 1)**. Without regard for consistency, we have used both versions of this function in our VBA for Greeks.

Sometimes it is convenient to use a homemade function for the normal probability density defined below:

```
'The standard normal probability density,
'this is N'(x)
Function normaldf(x)
   normaldf = Exp(-x ^ 2 / 2) / _
   (Sqr(2 * Application.Pi()))
End Function
```

Defining the Greeks

Below we give the Greeks in VBA:

```
Function DeltaCall(stock, exercise, time, interest, _
   divyield, sigma)
   DeltaCall = Exp(-divyield * time) * _
   Application.NormSDist(dOne(stock, exercise, _
   time, interest, divyield, sigma))
End Function
```

2. In Excel instead of **False** or **True**, we can also use 0 or 1. When using these functions in VBA, 0 or 1 is mandatory.

```
Function DeltaPut(stock, exercise, time, interest, _
    divyield, sigma)
    DeltaPut = -Exp(-divyield * time) * _
    Application.NormSDist(-dOne(stock, exercise, _
    time, interest, divyield, sigma))
End Function
```

In VBA there is a native function called **Gamma** which has nothing to do with options. Hence we use **OptionGamma** to define the Greek that refers to the option's convexity with respect to the underlying price, $\partial^2 V / \partial S^2$.

```
Function OptionGamma(stock, exercise, time, _
interest, divyield, sigma)
    temp = dOne(stock, exercise, time, _
    interest, divyield, sigma)
    OptionGamma = Exp(-divyield * time) * _
    Application.Norm_S_Dist(temp, 0) / _
    (stock * sigma * Sqr(time))
End Function

Function Vega(stock, exercise, time, _
interest, divyield, sigma)
    Vega = stock * Sqr(time) * _
    normaldf(dOne(stock, exercise, _
    time, interest, divyield, sigma)) _
    * Exp(-divyield * time)
End Function
```

```
Function ThetaCall(stock, exercise, time, _
interest, divyield, sigma)
   ThetaCall = -stock * normaldf _
   (dOne(stock, exercise, time, _
   interest, divyield, sigma)) * _
   sigma * Exp(-divyield * time) / _
   (2 * Sqr(time)) + divyield * stock * _
   Application.NormSDist(dOne(stock, _
   exercise, time, interest, _
   divyield, sigma)) * Exp(-divyield * time) _
   - interest * exercise * Exp(-interest * _
   time) * Application.NormSDist _
   (dTwo(stock, exercise, time, _
   interest, divyield, sigma))
End Function

Function ThetaPut(stock, exercise, time, _
interest, divyield, sigma)
   ThetaPut = -stock * normaldf _
   (dOne(stock, exercise, _
   time, interest, divyield, sigma)) * _
   sigma * Exp(-divyield * time) / _
   (2 * Sqr(time)) - divyield * stock _
   * Application.NormSDist(-dOne(stock, _
   exercise, time, interest, divyield, _
   sigma)) * Exp(-divyield * time) _
   + interest * exercise * Exp _
   (-interest * time) * Application.NormSDist _
   (-dTwo(stock, exercise, time, _
   interest, divyield, sigma))
End Function
```

```
Function RhoCall(stock, exercise, time, _
interest, divyield, sigma)
   RhoCall = exercise * time * _
   Exp(-interest * time) * _
   Application.NormSDist(dTwo _
   (stock, exercise, time, interest, _
   divyield, sigma))
End Function

Function RhoPut(stock, exercise, time, _
interest, divyield, sigma)
   RhoPut = -exercise * time * _
   Exp(-interest * time) * _
   Application.NormSDist(-dTwo _
   (stock, exercise, time, interest, _
   divyield, sigma))
End Function
```

19 Real Options

19.1 Overview

The standard net present value (NPV) analysis of capital budgeting values a project by discounting its expected cash flows at a risk-adjusted cost of capital. This *discounted cash flow* (DCF) technique is by far the most widely used practice for evaluating capital projects, be they acquisitions of companies or the purchases of machines. However, standard NPV analysis does not take account of the *flexibility* inherent in the capital budgeting process: Part of the complexity of the capital budgeting process is that the firm can change its decision dynamically, depending on the circumstances.

Here are two examples:

1. A firm is considering replacing some of its machines with a new type of machine. Instead of replacing all the machines together, it can first replace one machine. Based on the performance of the first machine replaced, the firm can then decide whether to replace the rest of the machines. This "option to wait" (or perhaps the "option to expand") is not valued in the standard NPV process. It is essentially a call option.
2. A firm is considering investing in a project which will produce (uncertain) cash flows over time. One option—not valued in the standard NPV framework—is to *abandon* the project if its performance is not satisfactory. The *abandonment option* is, as we see below, a put option which is implicit in many projects. It is also sometimes called the "option to contract scale."

There are many other real options. In the leading book on the valuation of real options, Trigeorgis (1996) lists the following common real options:

- The option to defer or to wait with developing a natural resource or build a plant.
- The time-to-build option (staged investment): At each stage the investment can be re-evaluated and (possibly) abandoned or expanded.
- The option to alter operating scale (expand, contract, shut down, or restart).
- The option to abandon.
- The option to switch inputs or outputs.
- The growth option—an early investment in a project constitutes an option to "get into the market" at a later date.

The recognition of real options is an important extension of the NPV techniques. However, modeling and valuing real options is more difficult than modeling and valuing standard cash flows by the DCF method. Our examples below illustrate these difficulties. Often it is best to implement real options by recognizing that the DCF technique misjudges the value of a project because it ignores the project's real options. Our usual conclusion will be that real options add to the value of a project, and that the NPV thus underestimates the true value.

19.2 A Simple Example of the Option to Expand

In this section we give a simple example of the option to expand. Consider ABC Corp., which has six widget machines. ABC is considering replacing each of the old machines with a new machine that costs $1,000. The new machines have a 5-year life. The anticipated cash flows for the new machine are given below.[1]

	A	B	C	D	E	F	G
1	**THE OPTION TO EXPAND**						
2	Year	0	1	2	3	4	5
3	CF of single machine	-1000	220	300	400	200	150
4							
5	Discount rate for machine cash flows (risk-adjusted)	12%					
6	Riskless discount rate	6%					
7	Present value of machine's future cash flows	932.52	<-- =NPV(B5,C3:G3)				
8	NPV of single machine	-67.48	<-- =NPV(B5,C3:G3)+B3				

The financial analyst working on the replacement project has estimated a cost of capital for the project of 12%. Using these anticipated cash flows and the 12% cost of capital, the analyst has concluded that the replacement of a single old machine by a new machine is unprofitable, since the NPV is negative:

$$-1000+\underbrace{\frac{220}{1.12}+\frac{300}{(1.12)^2}+\frac{400}{(1.12)^3}+\frac{200}{(1.12)^4}+\frac{150}{(1.12)^5}}_{\substack{\uparrow \\ \text{The present value of the machine's future} \\ \text{cash flow is \$932.52}}}=-67.48$$

1. These cash flows are the incremental cash flow of replacing a single old machine by a new machine. The computations include taxes, incremental depreciation, and the sale of the old machine.

Now comes the (real options) twist. The line manager in charge of the widget line says, "I want to try one of the new machines for a year. At the end of the year, if the experiment is successful, I want to replace five other similar machines on the line with the new machines."

Does this change our previously negative conclusion about replacing a single machine? The answer is "yes." To see this, we now realize that what we have is a package:

- Replacing a single machine today. This has an NPV of −67.48.
- The *option* of replacing five more machines in one year. Suppose that the risk-free rate is 6%. Then we view each such option as a call option on an asset which has current value S equal to the present value of the machine's future cash flows. As can be seen in cell B7 above, this present value is $S = 932.52$. The exercise price of this option is $X = 1{,}000$. Of course these call options can be exercised only if we purchase the first machine now.[2]

Suppose we assume that the Black-Scholes option pricing model can price this option. In this case we have:

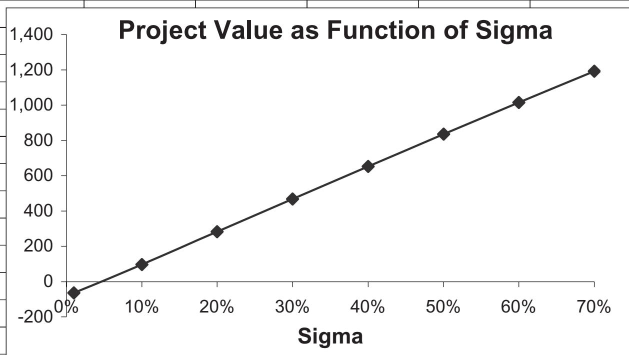
	A	B	C	D	E	F	G
1	**THE OPTION TO EXPAND**						
2	Year	0	1	2	3	4	5
3	CF of single machine	-1000	220	300	400	200	150
4							
5	Discount rate for machine cash flows (risk-adjusted)	12%					
6	Riskless discount rate	6%					
7	Present value of machine's future cash flows	932.52	<-- =NPV(B5,C3:G3)				
8	NPV of single machine	-67.48	<-- =NPV(B5,C3:G3)+B3				
9							
10	Number of machines bought next year	5					
11	Option value of single machine purchased in one more year	143.98	<-- =B24				
12	NPV of total project	652.39	<-- =B8+B10*B11				
13							
14	**Black-Scholes Option Pricing Formula**						
15	S	932.52	PV of machine CFs				
16	X	1000.00	Exercise price = Machine cost				
17	r	6.00%	Risk-free rate of interest				
18	T	1	Time to maturity of option (in years)				
19	Sigma	40%	<-- Volatility				
20	d_1	0.1753	<-- (LN(S/X)+(r+0.5*sigma^2)*T)/(sigma*SQRT(T))				
21	d_2	-0.2247	<-- d_1 - sigma*SQRT(T)				
22	$N(d_1)$	0.5696	<--- Uses formula NormSDist(d_1)				
23	$N(d_2)$	0.4111	<--- Uses formula NormSDist(d_2)				
24	Option value = BS call price	143.98	<-- S*N(d_1)-X*exp(-r*T)*N(d_2)				

2. What we're really doing is pricing the cost of learning!

As cell B12 shows, the value of the whole project is 652.39.

Our conclusion: Buying one machine today, and knowing that we have the option to purchase five more machines in one year is a worthwhile project. One critical element here is the volatility. The lower the volatility (i.e., the lower the uncertainty), the less worthwhile this project is:

	B	C	D	E	F	G	H	I
27	Data Table							
28	σ	652.39	<-- =B12 , data table header					
29	1%	-63.48						
30	10%	97.16						
31	20%	283.09						
32	30%	468.40						
33	40%	652.39						
34	50%	834.59						
35	60%	1,014.54						
36	70%	1,191.81						
37								
38								
39								
40								
41								
42								



This is not very surprising: The value of the project as a whole comes from our uncertainty about the actual cash flows 1 year from now. The less this uncertainty is (measured by σ), the less valuable the project will be.

Sidebar: Is Black-Scholes the Appropriate Valuation Tool for Real Options?

The answer is almost certainly no: Black-Scholes is not the appropriate tool. However, the Black-Scholes model is by far the most numerically tractable (i.e., easiest) model we have for valuing options of any kind. In valuing real options we often use the Black-Scholes model, realizing that at best it can give an approximation to the actual option value. Such is life.

Having said this, you should realize that the assumptions of the Black-Scholes option valuation model—continuous trading, constant interest rate, no exercise before final option maturity—are not really appropriate to the real options considered in this chapter. In many cases real options involve what, in a securities option context, would be considered dividend-paying securities and/or early exercise. Two examples:

- The staged-investment real option, when we have the opportunity to expand or contract the investment over time, is intrinsically an option with early exercise.
- When an option to abandon an investment exists, as long as the investment is still in place and not abandoned, it continues to pay "dividends," in the form of cash flows.

We can only hope that the Black-Scholes model gives an *approximation* to the option value intrinsic in the real options.

19.3 The Abandonment Option

Consider the following capital budgeting project:

	A	B	C	D	E
7	**Project cash flows**				
8					
9					150
10			100		
11					80
12	-50				
13					80
14			-50		
15					-60

As you can see, the initial cost of this project is 50. In one period the project will produce cash flows of either \$100 or –\$50; that is, under certain circumstances, it will lose money. Two periods hence the project again has chances of either losing money (in the worst case) or making money.

Valuing the Project

In order to value the project, we use the state prices from option pricing.[3] The state price q_u is the price today of \$1 to be paid in the succeeding period in

3. See discussion below on how to calculate these state prices.

the "up" state; and the price q_d is the price today of \$1 to be paid in the "down" state. The spreadsheet fragment below shows all the relevant details, leading to a project valuation of −\$29.38 (implying rejection of the project):

	A	B	C	D	E	F	G	H	I	J	K	L
1	**PRICING AN ABANDONMENT OPTION**											
2	**Market data**			**State prices**								
3	Expected market return	12%		q_U	0.3087	<-- =(1+B5-B9)/((1+B5)*(B8-B9))						
4	Sigma of market return	30%		q_D	0.6347	<-- =(B8-1-B5)/((1+B5)*(B8-B9))						
5	Risk-free rate	6%										
6												
7	**One-period "up" and "down" of market**											
8	Up	1.521962	<-- =EXP(B3+B4) , note that a valid alternative is 'Up' = EXP(B4)									
9	Down	0.83527	<-- =EXP(B3-B4) , note that a valid alternative is 'Down' = EXP(-B4)									
10												
11												
12	**Project cash flows**						**State-dependent present value factors**					
13												
14					150						0.0953	<-- =E3^2
15			100						0.3087			
16					80						0.1959	<-- =E3*E4
17	-50						1					
18					80						0.1959	<-- =E3*E4
19			-50						0.6347			
20					-60						0.4028	<-- =E4^2
21			=C15*I15									
22	**State-by-state present value**											
23					14.2981	<-- =E14*K14						
24			30.8740									
25					15.6755	<-- =E16*K16						
26	-50											
27					15.6755	<-- =E18*K18						
28			-31.7328									
29					-24.1673	<-- =E20*K20						
30												
31	**Net present value**		-29.38	<-- =SUM(A23:E29)								

The methodology is to calculate state-dependent present value factors (how this is done is discussed below) and to multiply these factors times the individual state-dependent cash flows. Each node of the tree is discounted by the relevant state price for the node; for example the cash flow of 80 which occurs at date 2 is discounted by $q_U q_D$. The NPV of the project is the sum of all the discounted cash flows plus the initial cost (cell B31).

The Abandonment Option Can Enhance Value

Now suppose that we can abandon the project at date 1 if its cash flow "threatens" to be −50; suppose, furthermore, that this abandonment means that all subsequent cash flows will also be zero. As the picture below shows, this *option to abandon the project* enhances its value:

	A	B	C	D	E	F	G	H	I	J	K	L
34	**Cash flows with abandonment**						**Present value with abandonment**					
35												
36					150						14.2981	<-- =E36*K14
37			100						30.8740			
38					80						15.6755	<-- =E38*K16
39	-50						-50					
40					0						0	
41			0						0			
42					0						0	
43												
44												
45							**Present value with abandonment**				10.85	<-- =SUM(G36:K42)

Thinking about this further, it is clear that it might even be worthwhile to *pay to abandon the project*. Here's what the project looks like when we pay $10 to abandon it in the troublesome state (this payment can be thought of as representing the cost of closing down a facility, etc.):

	A	B	C	D	E	F	G	H	I	J	K	L
34	**Cash flows with abandonment**						**Present value with abandonment**					
35												
36					150						14.2981	<-- =E36*K14
37			100						30.8740			
38					80						15.6755	<-- =E38*K16
39	-50						-50					
40					0						0	
41			-10						-6.3466			
42					0						0	
43												
44												
45							**Present value with abandonment**				4.50	<-- =SUM(G36:K42)

Abandonment When We Sell the Equipment

Another possibility is, of course, that "abandonment" means selling the equipment. In this case there might even be a positive cash flow from abandonment. As an example, suppose that we can sell the asset for $15:

	A	B	C	D	E	F	G	H	I	J	K	L
34	**Cash flows with abandonment**						**Present value with abandonment**					
35												
36					150						14.2981	<-- =E36*K14
37			100						30.8740			
38					80						15.6755	<-- =E38*K16
39	-50						-50					
40					0						0	
41			15						9.5198			
42					0						0	
43												
44												
45							**Present value with abandonment**				20.37	<-- =SUM(G36:K42)

Determining the State Prices

The method we have used above to determine the state prices was explained in greater detail in Chapter 16. We assume that in each period the market portfolio (by which we mean some large, diversified stock market portfolio such as the S&P 500) moves either "Up" or "Down"; the size of these moves is determined by the mean return μ of the market portfolio and by the standard deviation σ of the market portfolio's returns. Assuming that the returns on the market portfolio have mean $\mu = 12\%$ and standard deviation of returns $\sigma = 30\%$, we have—in the above examples—calculated

$$Up = \exp[\mu + \sigma] = 1.53, \quad Down = \exp[\mu - \sigma] = 0.84$$

Denote by q_U the price today for one dollar in the "Up" state in one period and denote by q_D the price today for one dollar in the "Down" state in one period. Then—as explained in Chapter 16—the state prices are calculated by solving the system of linear equations:

$$1 = q_U * Up + q_D * Down$$
$$\frac{1}{1+r} = q_U + q_D$$

The solution to this system of equations is:

$$q_U = \frac{R - Down}{R*(Up - Down)}, \quad q_D = \frac{Up - R}{R*(Up - Down)}$$

This method is illustrated in the above spreadsheet:

	A	B	C	D	E	F	G	H	I
2	**Market data**			**State prices**					
3	Expected market return	12%		q_U	0.3087	<-- =(1+B5-B9)/((1+B5)*(B8-B9))			
4	Sigma of market return	30%		q_D	0.6347	<-- =(B8-1-B5)/((1+B5)*(B8-B9))			
5	Risk-free rate	6%							
6									
7	**One-period "up" and "down" of market**								
8	Up	1.521962	<-- =EXP(B3+B4) , note that a valid alternative is 'Up' = EXP(B4)						
9	Down	0.83527	<-- =EXP(B3-B4) , note that a valid alternative is 'Down' = EXP(-B4)						

Alternative State Price Determinations

An alternative method of calculating the state prices is to try to match them to the project's cost of capital. Reconsider the project discussed above, and suppose that the actual probability of each state's occurrence is ½. Furthermore, suppose that the risk-free rate is 6%. Finally, assume that the project's discount rate—if it has no options whatsoever—is 22%. Then we can calculate the project's NPV without real options as 12.48:

	A	B	C	D	E	F	G	H
1	**MATCHING THE STATE PRICES TO THE COST OF CAPITAL**							
2	Project cost of capital	22%	<-- This is the discount rate for the project if it has **no options**					
3	Risk-free rate	6%						
4								
5	**Project cash flows**							
6					150			
7			100					
8					80			
9		-50						
10					80			
11			-50					
12					-60			
13								
14						=AVERAGE(C7:C1)		
15	**Project expected cash flows**: Assumes equal state probabilities							
16	Year	0	1	2				
17	Expected CF	-50	25	62.5		=AVERAGE(E6:E12)		
18	Project NPV	12.48	<-- =NPV(B2,C17:D17)+A9					
19								
20								
21	**State prices**							
22	q_U	0.4241	<-- =1/(1+B3)-B23					
23	q_D	0.5193	<-- Determined by Solver					

In cells B22 and B23 we look for state prices q_U and q_D, which have two properties:

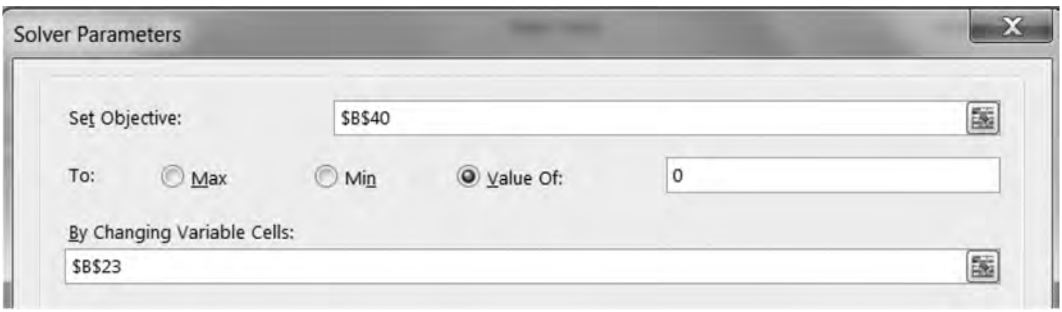
1. They are consistent with the risk-free interest rate. This means that $q_U + q_D = \frac{1}{R} = \frac{1}{1.06}$.

2. The state prices give the same NPV for the project as that calculated by the cost of capital.

The second requirement means that we have to use the Excel **Solver** to determine the state prices. Here's what the solution looks like (the discussion of how **Solver** was used follows the next spreadsheet):

	A	B	C	D	E	F	G
21	**State prices**						
22	q_U	0.4241	<-- =1/(1+B3)-B23				
23	q_D	0.5193	<-- Determined by Solver				
24							
25				=C7*B22			
26	**Project state-by-state discounting**						
27					26.9797	<-- =E6*B22^2	
28			42.4105				
29					17.6187	<-- =E8*B22*B23	
30		-50					
31					17.6187	<-- =E10*B22*B23	
32			-25.9646				
33					-16.1798	<-- =E12*B23^2	
34							
35			=C11*B23				
36							
37							
38	State-by-state NPV	12.48	<-- =SUM(A27:E33)				
39							
40	Target cell	(0.00)	<-- =B38-B18				

To determine the state prices, we use the **Solver** (**Data|Solver**):



One more note: You can also use Goal Seek (**Tools|Goal Seek**) to get the same result. However, Excel's Goal Seek does not remember its previous settings, meaning that each time you repeat this calculation you will have to reset the cell references. Here's what the Goal Seek dialog box looks like:

19.4 Valuing the Abandonment Option as a Series of Puts

The example above shows how and why the abandonment option can have value. It also illustrates another, more troublesome, feature of the abandonment option, namely, that it may be very difficult to value. While it is difficult enough to project expected cash flows, it is even more difficult to project state-by-state cash flows and state prices for a complex project.

A possible compromise in the valuation of an abandonment option is to value a project as a series of cash flows *plus* a series of Black-Scholes put options. Consider the following example: You are valuing a 4-year project with the expected cash flows given below and with a risk-adjusted discount rate of 12%. As you can see, the project has a negative NPV:

	A	B	C	D	E	F
1	**STANDARD DCF PROJECT VALUATION**					
2	**Project cash flows**					
3	Year	**0**	**1**	**2**	**3**	**4**
4	Cash flow	-750	100	200	300	400
5						
6	Risk-adjusted discount rate	12%	The project's cost of capital			
7	NPV without options	-33.53	<-- =B4+NPV(B6,C4:F4)			

Suppose that we can abandon the project at the end of any of the next 4 years, selling the equipment for 300. Although this abandonment option is an American option and not a Black-Scholes option, we value it as a series of Black-Scholes put options. In each case we suppose that we first get the year-end cash flow; we then value the abandonment option on the remaining project value.

• *End of year 1*: The asset's expected value at the end of year 1 will be the discounted value of its future expected cash flows: $702.44 = \frac{200}{1.12} + \frac{300}{(1.12)^2} + \frac{400}{(1.12)^3}$. The abandonment option means that we can get \$300 for the asset during the next 3 years. Suppose that the value has a volatility of 50%; then valuing this option as a Black-Scholes put with 1 year to maturity gives its value as 19.53. The following spreadsheet uses the VBA function **BSPut** defined in Chapter 17:

	A	B	C	D	E	F
1	**ABANDONMENT VALUE--DETAILS OF YEAR 1 CALCULATION**					
2	**Project cash flows**					
3	Year	**0**	**1**	**2**	**3**	**4**
4	Cash flow	-750	100	200	300	400
5						
6	Risk-adjusted discount rate	12%	The project's cost of capital			
7	NPV without options	-33.53	<-- =B4+NPV(B6,C4:F4)			
8						
9	**Valuing the year-1 abandonment put**					
10	Value of project, end year 1	702.44	<-- =NPV(B6,D4:F4)			
11	Abandonment value	300	Like strike price in put formula			
12	Time to option maturity (years)	3				
13	Risk-free rate	6%				
14	Sigma	50%				
15						
16	Put value	19.53	<-- =bsput(B10,B11,B12,B13,B14)			

• *End of year 2*: We have a put option with exercise price \$300 on an asset worth $586.73 = \frac{300}{1.12} + \frac{400}{(1.12)^2}$. Valuing the abandonment option as a Black-Scholes put with 2 years to exercise gives its value (when $\sigma = 50\%$) as 17.74.

• *End of year 3*: We have a put option with exercise price \$300 on an asset worth $357.14 = \frac{400}{1.12}$. The option has 1 more year remaining to its life and is worth 32.47.

- *End of year 4*: The asset is worthless in terms of future anticipated cash flows, but it can be abandoned for $300 (this is thus its scrap or salvage value). The abandonment option is worth $300.

In the spreadsheet below the asset has been valued as the sum of:

- The present value of the future expected cash flows. As we showed above, this is –$33.53.
- The present value (at the risk-free rate) of a series of Black-Scholes puts. This value is $299.10.

The total value of the project is –$33.53 + $299.10 = $265.57.

	A	B	C	D	E	F	G
1	**PRICING AN ABANDONMENT OPTION AS A SERIES OF PUTS**						
2	**Project cash flows**						
3	Year	**0**	**1**	**2**	**3**	**4**	
4	Cash flow	-750	100	200	300	400	
5							
6	Risk-adjusted discount rate	12%	The project's cost of capital				
7							
8	NPV without options	-33.53	<-- =NPV(B6,C4:F4)+B4				
9							
10	Sigma	50%					
11	Risk-free rate	6%					
12	Abandonment value	300	Project can be abandoned at end of any year for this amount				
13							
14	NPV of cash flows at RADR	-33.53	<-- =B8				
15	Value of abandonment option	299.10	<-- =NPV(B11,C20:F20)				
16	Adjusted present value	265.57	<-- =B15+B14				
17					Function in cell D19: =NPV($B$6,E4:$F$4)		
18							
19	End-year value of remaining cash flows		702.44	586.73	357.14	0.00	
20	Put option value		19.53	17.74	32.47	300.00	
21							
22					Function in cell D20: =bsput(D19,$B$12,$F$3-D3,$B$11,$B$10)		

19.5 Valuing a Biotechnology Project[4]

One of the interesting features of the biotech industry is the existence of highly valued firms that have no revenues. It is common understanding that the value

4. A version of this example originally appeared in Benninga and Tolkowsky (2002).

of those firms is in their future cash-flow opportunities. Therefore, understanding the translation of qualitative investment opportunities into quantitative valuation is of great importance when valuing those firms. In this section we use the real option method to value a biotechnology project and to illustrate the application of the real option approaches.

Consider the following story.[5] A firm is considering the initiation of research into a new drug. It knows that there are three stages to the drug's development:

- In the **discovery phase**, the firm does preliminary research about the viability of the idea. This research takes 1 year and costs $1,000 at the beginning of the year. With 50% probability the results will be positive enough to proceed to the next stage of research.
- If the discovery phase yields success, then the drug goes into the **clinical phase**, in which the drug is tested. This stage lasts 1 year and costs $2,000 at the beginning of the year, and with a probability of 30% yields enough positive results to proceed to the next stage.
- If the drug passes the clinical phase successfully, then it goes into the **market stage**, in which it is sold. This phase costs $15,000 per year (at the beginning of each year) and on average lasts 5 years. On average, a successful drug can be expected to start the marketing phase with income of $20,000. This income grows with annual mean 10% and standard deviation $\sigma = 100\%$.

The expected return on a project of this type is 25%. We assume that this is the cost of capital of the project in the case of a discounted cash flow (DCF) valuation.

5. We have made the story simple enough to fit an understandable spreadsheet. For a somewhat more complicated story in the same spirit, see Kellogg and Charnes (2000).

The Expected Value of the Project Using Traditional DCF Analysis

If we estimate the value of this project using traditional discounted cash flow analysis, we get a negative net present value for the project:

	A	B	C	D	E	F	G	H
1	**BIOTECH PROJECT EXPECTED CASH FLOWS**							
2	Discount rate	25%						
3	Growth	10%						
4								
5	**Year**	**Stage**	**Cost**	**Income**	**Net**	**Probability**	**Expected cash flow**	
6	0	Discovery	-1,000	0	-1,000	1	-1,000	<-- =F6*E6
7	1	Clinical	-2,000	0	-2,000	0.5	-1,000	<-- =F7*E7
8	2	Clinical	-2,000	0	-2,000	0.5	-1,000	
9	3	Marketing	-15,000	20,000	5,000	0.15	750	
10	4	Marketing	-15,000	22,000	7,000	0.15	1,050	
11	5	Marketing	-15,000	24,200	9,200	0.15	1,380	
12	6	Marketing	-15,000	26,620	11,620	0.15	1,743	
13	7	Marketing	-15,000	29,282	14,282	0.15	2,142	
14								
15	Project NPV	-268	<-- =G6+NPV(B2,G7:G13)					

Since the project's net present value is negative, the DCF approach indicates that it should not be undertaken.

Using a Real Options Approach

An alternative method for estimating the present value of proceeds is to plot the project's cash flows on a binomial tree. This is done below:

	A	B	C	D	E	F	G	H	I	J	K
1	**BIOTECH PROJECT, BINOMIAL TREE FOR THE CASH FLOWS**										
2	Marketing phase, Initial revenue	20,000					The expected return and variance of return				
3	Marketing, annual cost	15,000					is given by:				
4	Clinical annual cost	2,000					Expected	10%			
5	Initial, annual cost	1,000					σ	100%			
6											
7	Up	300%	<-- =EXP(H4+H5)								
8	Down	41%	<-- =EXP(H4-H5)								
9											
10	State prices										
11	q_u	0.2816									
12	q_d	0.6618	<-- =1/1.06-B11								
13											
14		**Net cash flows**								1,614,017	<-- =B2*B7^4-B3
15								527,253			
16							165,500			205,464	<-- =B2*B7^3*B8-B3
17						45,083		58,386			
18					5,000		9,428			14,836	<-- =B2*B7^2*B8^2-B3
19						-6,869		-5,068			
20							-11,694			-10,962	<-- =B2*B7*B8^3-B3
21				-2,000				-13,656			
22										-14,454	<-- =B2*B8^4-B3
23			-2,000								
24											
25		-1,000									
26											
27	**Time line**		**1**	**2**	**3**	**4**	**5**	**6**		**7**	
28											
29											
30		**State prices (to start of market phase)**								0.0001	<-- =B11^J27
31								0.0005			
32							0.0018			0.0013	<-- =B11^6*B12*COMBIN(4,3)
33						0.0063		0.0035			
34					0.0223		0.0083			0.0141	<-- =B11^5*B12^2*COMBIN(4,2)
35						0.0148		0.0083			
36							0.0098			0.0073	<-- =B11^4*B12^3*COMBIN(4,1)
37				0.079316				0.0065			
38										0.0043	<-- =B11^3*B12^4*COMBIN(4,0)
39			0.2816								
40											
41		1									
42											
43											
44	Binomial tree valuation	-268	<-- =SUMPRODUCT(B14:J25,B30:J41)								
45	Target										
46	DCF valuation	-268									
47	**DCF valuation - binomial tree value**	0	<-- Should be zero for correct state prices								

We use the Excel function **Sumproduct** to do this computation.

A Note About the State Prices

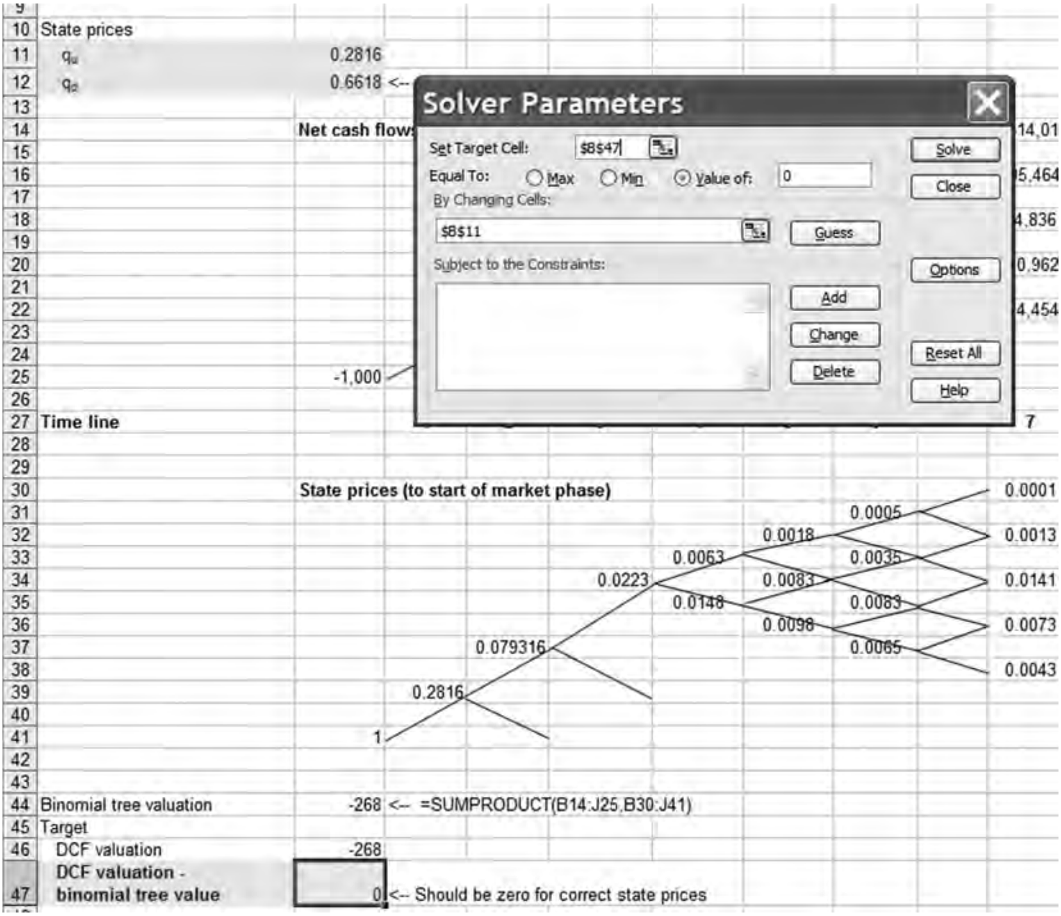
The net present value of the proceeds from the project is the product of the net cash flows and the appropriate state prices:

$$NPV = \sum_{t=0}^{7} \sum_{j=0}^{t} CF_{jt} * (q_U)^j * (q_D)^{t-j} * \begin{pmatrix} Number\ of \\ paths\ to\ node \end{pmatrix}$$

where CF_{jt} denotes the proceeds from the project at date t and state j, where j is the number of up moves. As explained in Chapter 16, in the standard binomial model, the state price for a node is $(q_U)^j * (q_D)^{n-j} \begin{pmatrix} t \\ j \end{pmatrix}$, where n is the time at which the node occurs, j is the number of Up steps needed to get to the node, and $\begin{pmatrix} t \\ j \end{pmatrix}$ is the number of paths to reach the node. The latter expression is computed in Excel by using the function **Combin(n,j)**. However, for the real options model above, the number of paths to each node is slightly different, since the beginning of the tree (the initial and clinical states) are accessible via only one path.

In the spreadsheet above, the prices q_U and q_D were computed (using **Solver**) so that the present value of the project on the binomial tree equals that of the DCF valuation and that the equilibrium condition $q_U + q_D = \frac{1}{1.06}$ holds.

Here's the **Solver** screen:



The Real Options Approach

The real options approach to R&D recognizes that at each stage in the project, the managers can choose whether to continue the project or not. They do this by comparing the value and costs of continuation. In option terminology, at each stage, the manager exercises her continuation option if the value from exercising the option exceeds the exercise price. In the spreadsheet below, we have eliminated obvious negative cash flows from the marketing stage, taking care to also eliminate the subsequent cash flows and to make an adjustment to the state prices (more on that below):

	A	B	C	D	E	F	G	H	I	J	K
1	**BIOTECH PROJECT, OPTION-ADJUSTED BINOMIAL TREE FOR THE CASH FLOWS**										
2	Marketing phase, Initial revenue	20,000					The expected return and variance of return				
3	Marketing, annual cost	15,000					is given by:				
4	Clinical annual cost	2,000					Expected	10%			
5	Initial, annual cost	1,000					σ	100%			
6											
7	Up	300%									
8	Down	41%									
9											
10	State prices										
11	q_u	0.2816									
12	q_d	0.6618									
13											
14		**Net cash flows**								1,614,017	<-- =B2*B7^4-B3
15								527,253			
16							165,500			205,464	<-- =B2*B7^3*B8-B3
17						45,083		58,386			
18					5,000		9,428			14,836	<-- =B2*B7^2*B8^2-B3
19											
20											
21				-2,000							
22											
23			-2,000								
24											
25		-1,000									
26											
27	**Time line**		**1**	**2**	**3**	**4**	**5**	**6**		**7**	
28											
29											
30		**State prices (to start of market phase)**								0.0001	<-- =B11^J27
31								0.0005			
32							0.0018			0.0010	<-- =B11^6*B12*(COMBIN(4,3)-1)
33						0.0063		0.0059			
34					0.0223		0.0167			0.0047	<-- =B11^5*B12*2*(COMBIN(4,1)-2)
35											
36											
37				0.079316							
38											
39			0.2816								
40											
41		1									
42											
43											
44	**Binomial tree valuation**	**229**	<-- =SUMPRODUCT(B14:J25,B30:J41)								

Another Note About State Prices

When we eliminate states in the real option approach, we must also adjust the number of paths to each node to account for the fact that some states are no longer reachable. This has been done in the above spreadsheet. For example the state in cell J32 (highlighted) is now reachable by one fewer path.

19.6 Summary

Recognizing that capital budgeting should include option aspects of projects is clear and obvious. Valuing these options is often difficult. In this chapter we have tried to emphasize the intuitions and—insofar as is possible—to give some implementation of the valuation.

Exercises

1. Your company is considering purchasing 10 machines, each of which has the following expected cash flows (the entry in B3 of −$550 is the cost of the machine):

	A	B	C	D	E	F
2	Year	0	1	2	3	4
3	CF of single machine	-550	100	200	300	400

You estimate the appropriate discount rate for the machines as 25%.

a. Would you recommend buying just one machine, if there are no options effects?

b. Your purchase manager recommends buying one machine today and then—after seeing how the machine operates—reconsidering the purchase of the other 9 machines in 6 months. Assuming that the cash flows from the machines have a standard deviation of 30% and that the risk-free rate is 10%, value this strategy.

2. Your company is considering the purchase of a new piece of equipment. The equipment costs $50,000 and your analysis indicates that the PV of the future cash flows from the equipment is $45,000. Thus the NPV of the equipment is −$5,000. This estimated NPV is based on some initial numbers provided by the manufacturer plus some creative thinking on the part of your financial analyst.

   The seller of the new piece of equipment is offering a course on how it works. The course costs $1,500. You estimate that the σ of the equipment's cash flows is 30%, the risk-free rate is 6%, and you will have another half year after the course to purchase the equipment at the price of $50,000. Is it worth taking the course?

3. Consider the project whose cash flows are given below:

	A	B	C	D	E	F	G	H
1	**Project cash flows**							
2								
3					169		**State prices**	
4			130				q_U	0.3000
5					91		q_D	0.5000
6	-100							
7					91			
8			70					
9					-90			

a. Using the state prices, value the project.

b. Suppose that at date 2 the project can be abandoned at no cost. What does this do to its value?

c. Suppose that at any time the project can be sold for 100. Show the tree of cash flows and value the project.

4. Suppose that the market portfolio has mean $\mu = 15\%$ and standard deviation $\sigma = 20\%$.

   a. If the risk-free rate of interest is 8% calculate the 1-period state prices for an "up" and a "down" state.

   b. Show the effect (in a data table) of the risk-free rate on the state prices.

   c. Show the effect of the σ on the state prices.

5. Consider the cash flows below:

	A	B	C	D	E
6	**Project cash flows**				
7					180
8			130		
9					90
10	-50				
11					60
12			-50		
13					-100

   a. If the cost of capital is 30% and the risk-free rate is 5%, find the state prices which match the project's NPV.

   b. If there exists an abandonment option so that we can change all negative cash flows to zero, value the project.

IV VALUING BONDS

Chapters 20–23 cover topics related to bonds and term structure. Chapters 20 and 21 concentrate on the classic duration and immunization formulations. In Chapter 20 we develop the basic Macaulay duration concept. Excel's **Duration( )** formula is somewhat cumbersome to use; we use VBA to build a new, easier-to-use formula. Chapter 21 discusses the use of duration to immunize bond portfolios. Chapter 22 shows how to model the term structure using a variety of methods; most of the chapter concentrates on the Nelson-Siegel term structure model. Chapter 23 uses a Markov process and much information about default probabilities and bond recovery ratios to model the expected rate of return on a risky corporate bond.

20 Duration

20.1 Overview

Duration is a measure of the sensitivity of the price of a bond to changes in the interest rate at which the bond is discounted. It is widely used as a risk measure for bonds—the higher a bond's duration, the more risky it is. In this chapter we consider a basic duration measure—Macaulay duration—which is defined for the case when the term structure is flat. In Chapter 21 we examine the uses of duration in immunization strategies.

Consider a bond with payments C_t, where $t = 1, \ldots, N$. Ordinarily, the first $N - 1$ payments will be interest payments, and C_N will be the sum of the repayment of principal and the last interest payment. If the term structure is flat and the discount rate for all of the payments is r, then the bond's market price today will be

$$P = \sum_{t=1}^{N} \frac{C_t}{(1+r)^t}$$

The Macaulay duration measure (throughout this chapter and the next, when we use the word *duration* we shall always refer to this measure) is meant to represent the time-weighted average maturity of the payments received from the bond. It is defined as:

$$D = \frac{1}{P} \sum_{t=1}^{N} \frac{tC_t}{(1+r)^t}$$

In section 20.3 we go deeper into the meaning of this formula. Before doing this, however, we show how to calculate the duration in Excel.

20.2 Two Examples

Consider two bonds. Bond A has just been issued. Its face value is \$1,000, it bears the current market interest rate of 7%, and it will mature in 10 years. Bond B was issued 5 years ago, when interest rates were higher. This bond has \$1,000 face value and bears a 13% coupon rate. When issued, this bond had a 15-year maturity, so its remaining maturity is 10 years. Since the current market rate of interest is 7%, bond B's market price is given by

$$\$1,421.41 = \sum_{t=1}^{10} \frac{\$130}{(1.07)^t} + \frac{\$1,00}{(1.07)^{10}}$$

It is worthwhile calculating the duration of each of the two bonds (just once!) the long way. We set up a table in Excel. Rows 17 through 24 show alternatives to the long computation of the duration.

	A	B	C	D	E	F	G
1	**BASIC DURATION CALCULATION**						
2	YTM	7%					
3							
4	**Year**	**$C_{t,A}$**	**$t*C_{t,A}$ / $Price_A*(1+YTM)^t$**		**$C_{t,B}$**	**$t*C_{t,B}$ / $Price_B*(1+YTM)^t$**	
5	1	70	0.0654		130	0.0855	<-- =$A5*E5/(E$16*(1+$B$2)^$A5)
6	2	70	0.1223		130	0.1598	<-- =$A6*E6/(E$16*(1+$B$2)^$A6)
7	3	70	0.1714		130	0.2240	
8	4	70	0.2136		130	0.2791	
9	5	70	0.2495		130	0.3260	
10	6	70	0.2799		130	0.3657	
11	7	70	0.3051		130	0.3987	
12	8	70	0.3259		130	0.4258	
13	9	70	0.3427		130	0.4477	
14	10	1,070	5.4393		1,130	4.0413	
15							
16	Bond price	1,000.00	<-- =NPV(B2,B5:B14)		1,421.41	<-- =NPV(B2,E5:E14)	
17	Duration	7.5152	<-- =SUM(C5:C14)		6.7535	<-- =SUM(F5:F14)	
18							
19	Using the Excel function **Duration** and the "home-made" function **Dduration**						
20	Bond A	7.5152	<-- =DURATION(DATE(1996,12,3),DATE(2006,12,3),7%,B2,1)				
21		7.5152	<-- =dduration(A14,7%,B2,1)				
22							
23	Bond B	6.7535	<-- =DURATION(DATE(1996,12,3),DATE(2006,12,3),13%,B2,1)				
24		6.7535	<-- =dduration(A14,13%,7%,1)				

As might be expected, the duration of bond A is longer than that of bond B, since the average payoff of bond A takes longer than that of bond B. To look at this another way, the net present value of bond A's first-year payoff ($70) represents 6.54% of the bond's price, whereas the net present value of bond B's first-year payoff ($130) is 8.55% of its price. The figures for the second-year payoffs are 6.11% and 7.99%, respectively. (For the second-year figures, you have to divide the appropriate line of the above spreadsheet by 2, since in the duration formula each payoff is weighted by the period in which it is received.)

Using the Excel Duration Formula

Excel has two duration formulas, **Duration( )** and **MDuration( )**. **MDuration —**somewhat inaccurately termed Macaulay duration by Excel—is defined as:

$$MDuration = \frac{Duration}{\left(1 + \frac{YTM}{Number\ of\ coupon\ payments\ per\ year}\right)}$$

Both formulas have the same syntax; for example, for **Duration( )** the syntax is:

Duration(settlement, maturity, coupon, yield, frequency, basis)

where

settlement is the settlement date (i.e., the purchase date) of the bond

maturity is the bond's maturity date

coupon is the bond's coupon

yield is the bond's yield to maturity

frequency is the number of coupon payments per year

basis is the "day count basis" (i.e., the number of days in a year). This is a code between 0 and 4:

0 or omitted US (NASD) 30/360

1 Actual/actual

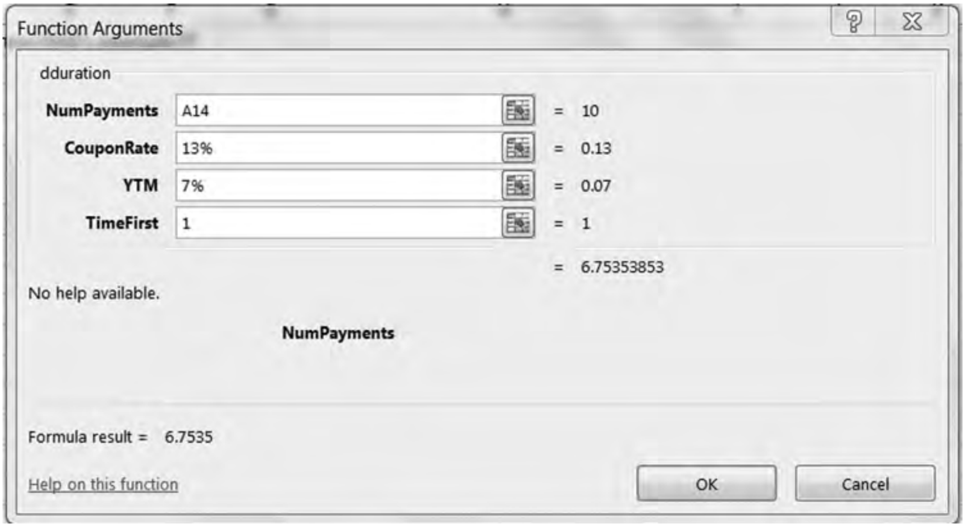
2 Actual/360

3 Actual/365

4 European 30/360

The **Duration** formula gives the standard Macaulay duration. The **MDuration** formula can be used in calculating the price elasticity of the bond (see section 20.3). These two duration formulas may require a bit of trickery to implement, because they demand a date serial number for both the settlement and the maturity. In the spreadsheet above, the Excel formula is implemented in cell C21 by assuming that bond A's settlement date (for our purposes: the current date) is 3 December 1996 and that the bond's maturity date is 3 December 2006. The choice of dates is arbitrary. The last parameter of the Excel duration formula, which gives the basis, is optional and could be omitted.

The insertion of serial date formats in the Excel **Duration** formula is often unhandy. Later in this chapter we use VBA to define a simpler duration formula which overcomes this problem and which also computes the duration of a bond when bond payments are unevenly spaced. This "homemade" duration formula is called **DDuration**. The programming aspects of this function are discussed in section 20.6. The previous spreadsheet illustrates this function in cells B21 and B24. The dialog box for **DDuration**'s computation of the duration of bond B is given below:



The parameter **TimeFirst** is the time from the bond purchase date until the first payment. For the examples of bond A and bond B, this parameter is 1.

20.3 What Does Duration Mean?

In this section we present three different meanings of duration. Each is interesting and important in its own right.

Duration as the Time-Weighted Average of the Bond's Payments

This was the original definition of Macaulay (1938). Rewrite the duration formula as follows:

$$D = \frac{1}{P}\sum_{t=1}^{N}\frac{tC_t}{(1+r)^t} = \sum_{t=1}^{N}\frac{C_t/P}{(1+r)^t} * t$$

The bracketed terms $\left[\frac{C_t/P}{(1+r)^t}\right]$ sum to 1. This follows from the definition of the bond price; each of these terms is the proportion of the bond's price represented by the payment at time t. In the duration formula, each of the terms $\left[\frac{C_t/P}{(1+r)^t}\right]$ is multiplied by its time of occurrence: Thus, *the duration is the time-weighted average of the bond's discounted payments as a proportion of the bond's price.*

Duration as the Bond's Price Elasticity with Respect to Its Discount Rate

This way of viewing duration explains why the duration measure can be used to measure the bond's price volatility; it also shows why duration is often used as a risk measure for bonds. To derive this interpretation, we take the derivative of the bond's price with respect to the current interest rate:

$$\frac{dP}{dr} = \sum_{t=1}^{N} \frac{-tC_t}{(1+r)^{t+1}}$$

A little algebra shows that:

$$\frac{dP}{dr} = \sum_{t=1}^{N} \frac{-tC_t}{(1+r)^{t+1}} = -\frac{DP}{1+r}$$

which transforms into two useful interpretations of duration:

- First, duration can be regarded as the *elasticity of the bond price with respect to the discount factor*, where by "discount factor" we mean 1 + r:

$$\frac{dP/P}{dr/(1+r)} = \frac{\%\ \textit{change in bond price}}{\%\ \textit{change in discount factor}} = -D$$

- Second, we can use duration to measure the *price volatility* of a bond, by rewriting the previous equation as:

$$\frac{dP}{P} = -D\frac{dr}{1+r}$$

To show this interpretation of duration in a spreadsheet, we go back to the examples of the previous section. Suppose that the market interest rate rises by 10%, from 7% to 7.7%. What will happen to the bond prices? The price of bond A will be

$$\$952.39 = \sum_{t=1}^{10} \frac{\$70}{(1.077)^t} + \frac{\$1{,}000}{(1.077)^{10}}$$

A similar calculation shows the price of bond B to be

$$\$1{,}360.50 = \sum_{t=1}^{10} \frac{\$130}{(1.077)^t} + \frac{\$1{,}000}{(1.077)^{10}}$$

As predicted by the price-volatility formula, the changes in the bond prices are approximated by $\Delta P \cong -DP\Delta r/(1+r)$. To see this, work out the numbers for each bond:

	A	B	C	D	E	F
1	**DURATION AS PRICE ELASTICITY** **The change in the bond price can be approximated by** **ΔP ≈ - Duration*Price*Δr/(1+r)**					
2	Discount rate	7%				
3						
4	**Bond A**			**Bond B**		
5	Coupon rate	7%		Coupon	13%	
6	Face value	1,000		Face value	1,000	
7	Maturity	10		Maturity	10	
8						
9	Price	1,000.00		Price	1,421.41	<-- =PV($B$2,E7,-E5*E6)+E6/(1+$B$2)^E7
10	Duration	7.5152		Duration	6.7535	<-- =DURATION(DATE(1996,1,1),DATE(2006,1,1),E5,B2,1)
11						
12	New discount rate	7.70%				
13	New price	952.39			1,360.50	<-- =PV($B$12,E7,-E5*E6)+E6/(1+$B$12)^E7
14						
15	**Change in price**					
16	Actual	47.61			60.92	<-- =E9-E13
17	Using duration as approximation DP ≈ - Duration *Price*Δr/(1+r)	49.17			62.80	<-- =-E10*E9*($B$2-$B$12)/(1+$B$2)
18						
19	Using **MDuration**	49.17			62.80	<-- =-($B$2-$B$12)*E9*MDURATION(DATE(1996,1,1),DATE(2006,1,1),E5,$B$2,1)

Note row 19 of the above spreadsheet: Instead of using the Excel **Duration** function and multiplying by $\frac{\Delta r}{(1+r)}$, we could have used the **MDuration** function and multiplied by Δr.

Babcock's Formula: Duration as the Convex Combination of Bond Yields

A third interpretation of duration is Babcock's (1985) formula, which shows that duration is a weighted average of two factors:

$$D = N\left(1 - \frac{y}{r}\right) + \frac{y}{r} PVIF(r, N) * (1 + r)$$

where

$$y = \frac{Bondcoupon}{Bondprice}, \text{ the current yield of the bond}$$

$$PVIF(r, N) = \sum_{i=1}^{N} \frac{1}{(1+r)^i}, \text{ the present value of an N-period annuity}$$

This formula gives two useful insights into the duration measure:

- Duration is a weighted average of the maturity of the bond and of (1 + r) times the PVIF associated with the bond. (Note that the PVIF is given by the Excel formula **PV(r,N,-1)**.)
- In many cases the current yield of the bond, y, is not greatly different from its yield to maturity r. In these cases, duration is not very different from (1 + r)*PVIF*.

Unlike the two previous interpretations, Babcock's formula holds only for the case of a bond with constant coupon payments and single repayment of principal at time N; that is, the formula does not extend to the case where the payments C_t differ over time.

Here's an implementation of Babcock's formula for bond B:

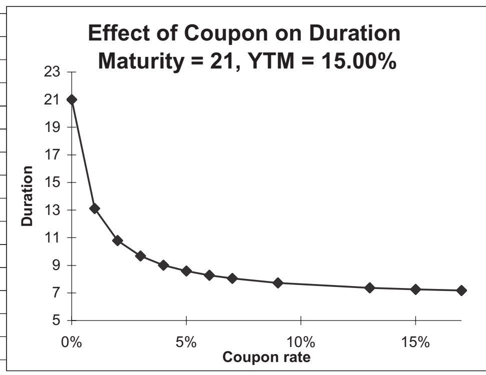
	A	B	C
1	**BABCOCK'S FORMULA FOR DURATION** **Duration is convex combination of current yield and present value factor:** **D = N*y/r + (1 - y/r) * PV(r,N,-1)*(1+r)**		
2	N, bond maturity	10	
3	r,	7%	
4	C, bond coupon	13%	
5	Face value	1,000	
6	Price	1,421.41	<-- =PV(B3,B2,-B4*B5)+B5/(1+B3)^B2
7	Current yield	9.15%	<-- =B4*B5/B6
8	PVIF(r,N)	7.0236	<-- =PV(B3,B2,-1)
9			
10	**Two duration formulas**		
11	Babcock's formula	6.7535	<-- =B2*(1-B7/B3)+B7/B3*B8*(1+B3)
12	Standard formula	6.7535	<-- =DURATION(DATE(1995,1,1),DATE(2005,1,1),B4,B3,1)

20.4 Duration Patterns

Intuitively we would expect that duration is a decreasing function of a bond's coupon and an increasing function of a bond's maturity. The first of these intuitions is correct, but the second is not.

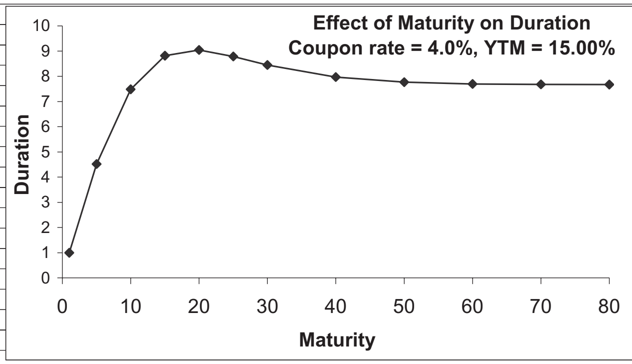
The spreadsheet below shows the effect of increasing the coupon on a bond's duration, which—as our intuition indicated—indeed declines as the coupon increases:

	A	B	C	D
1	**EFFECT OF COUPON ON DURATION**			
2	Current date	5/21/1996	<-- =DATE(1996,5,21)	
3	Maturity, in years	21		
4	Maturity date	5/21/2017	<-- =DATE(1996+B3,5,21)	
5	YTM	15%	Yield to maturity (i.e., discount rate)	
6	Coupon	4%		
7	Face value	1,000		
8				
9	Duration	9.0110	<-- =DURATION(B2,B4,B6,B5,1)	
10				
11	**Data table: Effect of coupon on duration**			
12			9.0110	<-- =B9 , data table header
13		0%	21.0000	
14		1%	13.1204	
15	Bond coupon -->	2%	10.7865	
16		3%	9.6677	
17		4%	9.0110	
18		5%	8.5792	
19		6%	8.2736	
20		7%	8.0459	
21		9%	7.7294	
22		13%	7.3707	
23		15%	7.2593	
24		17%	7.1729	
25				
26				
27				
28				
29				



It is not true, however, that the duration is always an increasing function of the bond maturity:

	A	B	C	D
1	**EFFECT OF MATURITY ON DURATION**			
2	Current date	5/21/1996	<-- =DATE(1996,5,21)	
3	Maturity, in years	21		
4	Maturity date	5/21/2017	<-- =DATE(1996+B3,5,21)	
5	YTM	15%	Yield to maturity (i.e., discount rate)	
6	Coupon	4%		
7	Face value	1,000		
8				
9	Duration	9.0110	<-- =DURATION(B2,B4,B6,B5,1)	
10				
11	**Data table: Effect of maturity on duration**			
12			9.0110	<-- =B9 , data table header
13		1	1.0000	
14		5	4.5163	
15	Bond maturity -->	10	7.4827	
16		15	8.8148	
17		20	9.0398	
18		25	8.7881	
19		30	8.4461	
20		40	7.9669	
21		50	7.7668	
22		60	7.6977	
23		70	7.6759	
24		80	7.6693	
25				
26				
27				
28				
29				
30				
31				



20.5 The Duration of a Bond with Uneven Payments

The duration formulas discussed above assume that bond payments are evenly spaced. This is almost invariably the case for bonds, *except for the first payment*. For example, consider a bond that pays interest on 1 May of each of the years 1997, 1998, … , 2010, with repayment of its face value on the last date. All the payments are spaced 1 year apart; however, if this bond is purchased on 1 September 1996, then the time to the first payment is 8 months (September to May), not 1 year. We shall refer to such a bond as a *bond with uneven payments*. In this section we discuss two aspects of this (extremely common) problem:

• The calculation of the duration of such a bond, when the YTM is known. We show that the duration has a very simple formula, related to the duration of a bond with even payments (i.e., the standard duration formula). In the process of the discussion we develop a simpler duration formula in Excel.

• The calculation of the YTM of a bond with uneven payments. This requires a bit of trickery, and ultimately leads us to another VBA function.

Duration of a Bond with Uneven Payments

Consider a bond with N payments, the first of which occurs at time $\alpha < 1$, and the rest of which are evenly spaced. In the derivation which follows, we show that the duration of such a bond is given by the sum of two terms:

• First term: The duration of a bond with N payments spaced at even intervals (i.e., the standard duration discussed above).

• Second term: $\alpha - 1$

It is relatively simple to show why this is so. Denote the payments on the bond by C_α, $C_{\alpha+1}$, $C_{\alpha+2}$, … , $C_{\alpha+N-1}$, where $0 < \alpha < 1$. The price of the bond is given by:

$$P = \sum_{t=1}^{N} \frac{C_{\alpha+t-1}}{(1+r)^{\alpha+t-1}} = (1+r)^{1-\alpha} \sum_{t=1}^{N} \frac{C_{\alpha+t-1}}{(1+r)^{t}}$$

The duration of this bond is given by:

$$D = \frac{1}{P} \sum_{t=1}^{N} \frac{(\alpha+t-1)C_{\alpha+t-1}}{(1+r)^{\alpha+t-1}}$$

Rewrite this last expression as follows:

$$\begin{aligned} D &= \frac{1}{P}(1+r)^{1-\alpha} \left\{ \sum_{t=1}^{N} \frac{tC_{t+\alpha-1}}{(1+r)^{t}} + \sum_{t=1}^{N} \frac{(\alpha-1)C_{t+\alpha-1}}{(1+r)^{t}} \right\} \\ &= \frac{1}{(1+r)^{1-\alpha} \sum_{t=1}^{N} \frac{C_{t+\alpha-1}}{(1+r)^{t}}} (1+r)^{1-\alpha} \left\{ \sum_{t=1}^{N} \frac{tC_{t+\alpha-1}}{(1+r)^{t}} + (\alpha-1) \sum_{t=1}^{N} \frac{C_{t+\alpha-1}}{(1+r)^{t}} \right\} \\ &= \frac{1}{\sum_{t=1}^{N} \frac{C_{t+\alpha-1}}{(1+r)^{t}}} \left\{ \sum_{t=1}^{N} \frac{tC_{t+\alpha-1}}{(1+r)^{t}} \right\} + \alpha - 1 \end{aligned}$$

Here is an example of the calculation of the duration of a bond with uneven periods. Recall that when there is α until the first payment, the duration formula is given by:

$$D=\sum_{t=1}^{N}\frac{1}{P}\frac{(\alpha+t-1)C_{\alpha+t-1}}{(1+r)^{\alpha+t-1}}$$

Here is an example of the calculation of the duration of a bond with uneven periods. Each of the cells D10:D14 calculates the value of a term of this formula:

	A	B	C	D
1	**DURATION OF BOND WITH UNEVEN PERIODS Brute Force Calculation and DDuration Function**			
2	Alpha	0.3	Time until first coupon payment (in years)	
3	N	5	Number of payments	
4	YTM	6%		
5	Coupon	100		
6	Face	1,000		
7	Bond price	1,217	<-- =NPV(B4,B10:B14)*(1+B4)^(1-B2)	
8				
9	**Period**	**Payment**	**t*C$_t$ /Price*(1+YTM)^t**	
10	0.3	100	0.0242	<-- =(B10*A10)/(1+$B$4)^A10/$B$7
11	1.3	100	0.0990	
12	2.3	100	0.1653	
13	3.3	100	0.2237	
14	4.3	1,100	3.0249	
15	Duration		3.5371	<-- =SUM(C10:C14)
16				
17	Newly defined VBA function		3.5371	<-- =dduration(B3,B5/B6,B4,B2)

As noted in section 20.2, the built-in Excel formula **Duration( )** is somewhat difficult to use, because of the insertion of the dates. We have already introduced a simpler duration formula using VBA; the syntax of this formula is **DDuration(numPayments, couponRate, YTM, timeFirst)**:

```
Function DDuration(NumPayments, CouponRate, _
YTM, TimeFirst)
   price = 1 / (1 + YTM) ^ NumPayments
   DDuration = NumPayments / (1 + YTM) ^ _
   NumPayments
   For Index = 1 To NumPayments
      price = CouponRate / (1 + YTM) ^ _
      Index + price
   Next Index
   For Index = 1 To NumPayments
      DDuration = CouponRate * Index / _
      (1 + YTM) ^ Index + DDuration
   Next Index
   DDuration = DDuration / price + _
   TimeFirst - 1
End Function
```

Our homemade formula **DDuration** requires only the number of payments on the bond, the coupon rate, and the time to the first payment α. The use of the formula is illustrated in the previous spreadsheet, in cell C17.

Calculating the YTM for Uneven Periods

As the discussion above shows, the calculation of duration requires us to know the bond's yield to maturity (YTM); this YTM is just the internal rate of return of the bond's payments and its initial price. Often the YTM is given, but when it is not, we cannot use Excel's **IRR** function but must instead use the **XIRR** function.

Consider a bond that currently costs $1,123 and which pays a coupon of $89 on each of the next 1 January. On 1 January 2001, the bond will pay $1,089, the sum of its annual coupon and its face value. The current date is 3

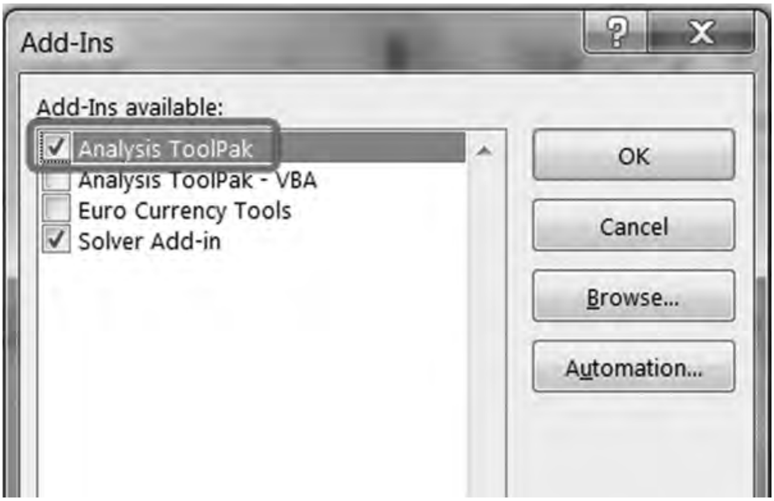
October 1996. The problem in finding the YTM of this bond is that while most of the bond payments are spaced 1 year apart, there is only 0.2466 of a year until the first coupon payment (0.2466 = (Date(1997,1,1)-Date(1996,10,3))/365). Thus we wish to use Excel to solve the following equation:

$$-1,123+\sum_{t=0}^{3}\frac{89}{(1+YTM)^{t+0.2466}}+\frac{1089}{(1+YTM)^{4.2466}}=0$$

To solve this problem, we can use the Excel function **XIRR**:

	A	B	C	D
1	**USING XIRR TO CALCULATE THE IRR WITH UNEVEN PAYMENTS**			
2	Current date	3-Oct-96		
3	Annual coupon	89	Paid January 1 for each of next 5 years	
4	Maturity date	1-Jan-01		
5	Face value	1,000		
6	Price of bond	1,123		
7				
8	Time to first payment	0.2466	<-- =(B12-B11)/365	
9				
10		**Date**	**Payment**	
11		3-Oct-96	-1,123	
12		1-Jan-97	89	
13		1-Jan-98	89	
14		1-Jan-99	89	
15		1-Jan-00	89	
16		1-Jan-01	1,089	
17				
18		YTM	7.300%	<-- =XIRR(C11:C16,B11:B16)

To use the **XIRR** function, you first have to make sure that the **Analysis ToolPak** is loaded into Excel. Go to **File|Options|Add-Ins|Manage**. This brings up the following menu, in which you have to make sure that **Analysis ToolPak** is checked:



You can now use **XIRR**, which returns the internal rate of return for a schedule of cash flows that is not necessarily periodic. To use this function you have to specify the list of cash flows and the list of dates. As in the case of the Excel function **IRR**, You can also provide a guess for the IRR, although this may be left out.[1]

Calculating the YTM for Uneven Payments Using a VBA Program

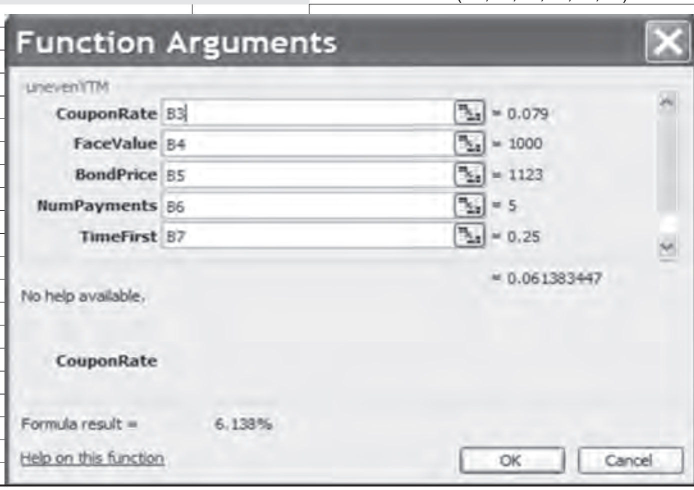
If you do not know the payment dates, you can use VBA to calculate the YTM for a series of uneven payments. The program below is composed of two functions. The first function, **annuityvalue**, calculates the value $\sum_{t=1}^{N} \frac{1}{(1+r)^t}$. The second function, **unevenYTM**, uses the simple bisection technique to calculate the YTM of a series of uneven payments, leaving you to choose the accuracy **epsilon** of the desired result:

1. There is also a function **XNPV** for finding the present value of a series of payments paid out at uneven dates. This function is discussed in Chapter 33.

```
Function Annuityvalue(interest, numPeriods)
   annuityvalue = 0
   For Index = 1 To numPeriods
      annuityvalue = annuityvalue + 1 / _
      (1 + interest) ^ Index
   Next Index
End Function

Function UnevenYTM(CouponRate, FaceValue, _
BondPrice, NumPayments, TimeFirst, epsilon)
   Dim YTM As Double
   high = 1
   low = 0
      While Abs(annuityvalue(YTM, _
      NumPayments) * CouponRate * _
         FaceValue + FaceValue / _
         (1 + YTM) ^ NumPayments - _
         BondPrice / (1 + YTM) ^ _
         (1 - TimeFirst)) >= epsilon
      YTM = (high + low) / 2
      If annuityvalue(YTM, NumPayments) * _
      CouponRate * _
         FaceValue + FaceValue / (1 + YTM) _
         ^ NumPayments - BondPrice / _
         (1 + YTM) ^ (1 - TimeFirst) > 0 _
         Then
            low = YTM
         Else
            high = YTM
      End If
      Wend
   UnevenYTM = (high + low) / 2
End Function
```

Here's an illustration of the use of this function:

	A	B	C
1	**ILLUSTRATION OF CALCULATION OF YTM OF UNEVEN PERIODS**		
2	This spreadsheet illustrates the **unevenYTM** VBA function: the syntax of this function is **unevenYTM**(CouponRate,FaceValue,BondPrice,NumPayments,TimeFirst,epsilon)		
3	Coupon rate	7.90%	
4	Face value	1,000.00	
5	Bond price	1,123.00	
6	Number of payments	5	
7	Time to first payment	0.25	
8	Epsilon	0.00001	<-- Controls the accuracy of the YTM calculation
9			
10	YTM	6.138%	<-- =unevenYTM(B3,B4,B5,B6,B7,B8)
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			



We can, of course, use the **DDuration** function in conjunction with **UnevenYTM** to compute the duration:

	A	B	C
1	**USING DDURATION AND UNEVENYTM TOGETHER**		
2	Coupon rate	7.90%	
3	Face value	1,000.00	
4	Bond price	1,123.00	
5	Number of payments	5	
6	Time to first payment	0.25	
7	Epsilon	0.00001	<-- Controls the accuracy of the YTM calculation
8			
9	YTM	6.138%	<-- =unevenYTM(B2,B3,B4,B5,B6,B7)
10			
11	Duration	3.5959	<-- =dduration(B5,B2,B9,B6)

20.6 Non-Flat Term Structures and Duration

In a general model of the term structure, payments at time t are discounted by rate r_t, so that the value of a bond is given by:

$$P = \sum_{t=1}^{N} \frac{C_t}{(1+r_t)^t}$$

The duration measure discussed in this chapter assumes either a flat term structure (i.e., $r_t = r$ for all t) or a term structure which shifts in a parallel fashion. When the term structure exhibits parallel shifts, we can write the bond price as:

$$P = \sum_{t=1}^{N} \frac{C_t}{(1+r_t+\Delta t)^t}$$

and then derive a measure of duration by taking the derivative with respect to Δt.

A general model of the term structure should explain how the discount rate r_t for time-t payments comes about, and how the rates at time t change. This is a difficult problem, discussed in Chapters 22 and 23.

Does this mean that the simple duration measure we present in this chapter is useless? Not necessarily. It may be that the Macaulay duration measure gives a good approximation for changes in bond value as a result of changes in the term structure, even for the case when the term structure itself is relatively complex and not flat.[2] In this section, we explore this possibility, using data from a file of term structures developed by Prof. J. Huston McCullogh, which is on the disk that accompanies this book.[3] The file contains monthly information on the term structure of interest rates in the United States for the period 12.1949 to 2.87 (i.e., December 1949–February 1987). A typical row of this file looks like:

	0 mo	1 mo	2 mo	3 mo	4 mo	5 mo	6 mo
12.1946	0.18	0.32	0.42	0.48	0.52	0.55	0.58

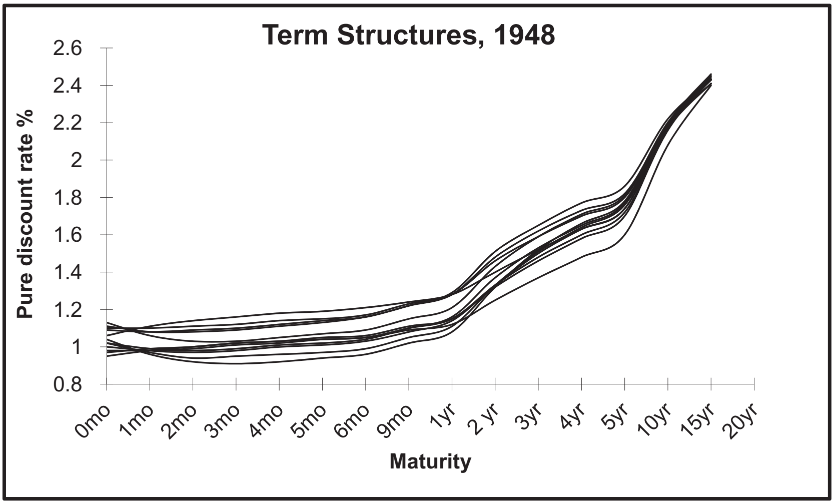
9 mo	1 yr	2 yr	3 yr	4 yr	5 yr	10 yr	15 yr	20 yr
0.65	0.72	0.95	1.15	1.3	1.41	1.82	2.16	2.32

This particular row gives the term structure of interest rates in December 1946. Interest rates are given in *annual percentage terms*; that is, 0.32 means 0.32% per year. Here are some pictures of term structures, taken from the file.[4] In the graphs below, each line represents the term structure in a particular month. In 1948 the term structures were very closely correlated, and all were upward sloping:

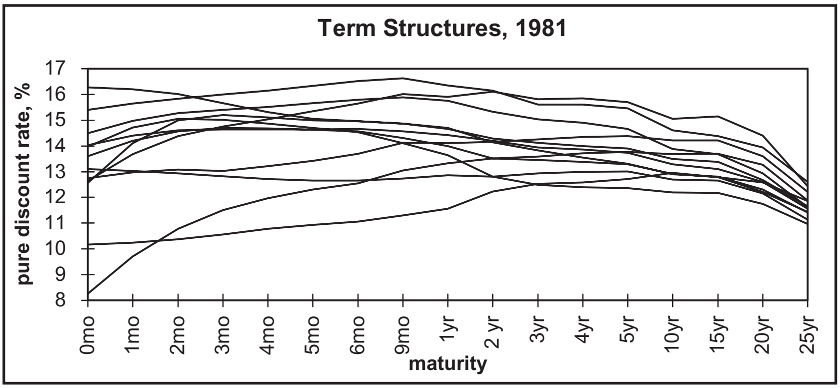
2. A paper by Gultekin and Rogalski (1984) seems to confirm this.

3. The data are from McCullogh (1990). For updated daily term structures on Treasuries, go to http://www.treasury.gov/resource-center.

4. The interest rates are pure discount rates, calculated so that the value of a bond with price P and with N payments, $C_1, C_2, \ldots, C_N$ is $P = \sum_{t=1}^{N} \frac{C_t}{(1+r_t)^t}$. The column marked "0mo" gives the *instantaneous interest rate*—the shortest-term interest rate in the market. You can think of this as the rate paid by a money market fund on a 1-day deposit.



Contrast this with the term structures in 1981: Here there were upward- and downward-sloping term structures, as well as term structures with “humps”:



Despite this great variety of term structure shapes, you will see in exercise 7 to this chapter that the Macaulay duration can give an adequate approximation to the change in bond value over short periods.

20.7 Summary

In this chapter we have summarized the basics of duration, a commonly used risk measure for bonds. The duration measure was originally developed by Macaulay (1938) to measure the time-weighted average of a bond's payments. It can also represent the bond's price elasticity to a change in its discount rate. This chapter has explored duration computation basics; in the next chapter we use duration to describe the immunization of a bond portfolio.

Exercises

1. In the spreadsheet below, create a **Data Table** in which the duration is computed as a function of the coupon rate (coupon = 0%, 1%, … , 11%). Comment on the relation between the coupon rate and the duration.

	A	B	C
1	**CHANGING THE COUPON RATE** **Effect on Duration**		
2	Current date	21-May-07	
3	Maturity, in years	21	
4	Maturity date	21-May-27	
5	YTM	15%	
6	Coupon	4%	
7	Face value	1,000	
8			
9	Duration	9.03982	<-- =DURATION(B2,B4,B6,B5,1)

2. What is the effect on a bond's duration of increasing the bond's maturity? As in the previous example, use a numerical example and plot the answer. Note that as $N \to \infty$, the bond becomes a consol (a bond that has no repayment of principal but an infinite stream of coupon payments). The duration of a consol is given by $(1 + YTM) / YTM$. Show that your numerical answers converge to this formula.

3. "Duration can be viewed as a proxy for the riskiness of a bond. All other things being equal, the riskier of two bonds should have lower duration." Check this claim with an example. What is its economic logic?

4. A pure discount bond with maturity N is a bond with *no payments* at times $t = 1, \ldots, N - 1$; at time $t = N$, a pure discount bond has a single terminal payment of both principal and interest. What is the duration of such a bond?

5. Replicate the two graphs in section 20.5.

6. On 23 January 1987, the market price of a West Jefferson Development Bond was $1,122.32. The bond pays $59 in interest on 1 March and 1 September of each of the years 1987–1993. On 1 September 1993, the bond is redeemed at its face value of $1,000. Calculate the yield to maturity of the bond and then calculate its duration.

7. Rewrite the formula **DDuration** in section 20.5, so that if the **timeToFirstPayment** α is not inserted, then α automatically defaults to 1.

21 Immunization Strategies

21.1 Overview

A bond portfolio's value in the future depends on the interest-rate structure prevailing up to and including the date at which the portfolio is liquidated. If a portfolio has the same payoff at some specific future date, no matter what interest-rate structure prevails, then it is said to be *immunized*. This chapter discusses immunization strategies, which are closely related to the conception of duration discussed in Chapter 20. Immunization strategies have been discussed for many concepts of duration, but this chapter is restricted to the simplest duration concept, that of Macaulay.

21.2 A Basic Simple Model of Immunization

Consider the following situation: A firm has a known future obligation, Q (a good example would be an insurance firm, which knows that it has to make a payment in the future). The discounted value of this obligation is

$$V_0 = \frac{Q}{(1+r)^N}$$

where r is the appropriate discount rate.

Suppose that this future obligation is hedged by a bond held by the firm. By this we mean that the firm currently holds a bond, whose value V_B is equal to the discounted value of the future obligation, V_0. If P_1, P_2, … P_M is the stream of anticipated payments made by the bond, then the bond's present value is given by

$$V_B = \sum_{t=1}^{M} \frac{P_t}{(1+r)^t}$$

Now suppose that the underlying interest rate, r, changes to $r + \Delta r$. Using a first-order linear approximation, we find that the new value of the future obligation is given by

$$V_0 + \Delta V_0 \approx V_0 + \frac{dV_0}{dr}\Delta r = V_0 + \Delta r\left[\frac{-NQ}{(1+r)^{N+1}}\right]$$

However, the new value of the bond is given by

$$V_B + \Delta V_B \approx V_B + \frac{dV_B}{dr}\Delta r = V_B + \Delta r \sum_{t=1}^{N} \frac{-tP_t}{(1+r)^{t+1}}$$

If these two expressions are equal, a change in r will not affect the hedging properties of the company's portfolio. Setting the expressions equal gives us the condition

$$V_B + \Delta r \sum_{t=1}^{N} \frac{-tP_t}{(1+r)^{t+1}} = V_0 + \Delta r \left[\frac{-NQ}{(1+r)^{N+1}} \right]$$

Recalling that

$$V_B = V_0 = \frac{Q}{(1+r)^N}$$

we can simplify this expression to get

$$\frac{1}{V_B} \sum_{t=1}^{M} \frac{tP_t}{(1+r)^t} = N$$

The last equation is worth restating as a formal proposition: Suppose that the term structure of interest rates is always flat (that is, the discount rate for a cash flow occurring at all future times is the same) or that the term structure moves up or down in parallel movements. Then a necessary and sufficient condition that the market value of an asset be equal under all changes of the discount rate r to the market value of a future obligation Q is that the duration of the asset equal the duration of the obligation. Here we understand the word "equal" to mean equal in the sense of a first-order approximation.

An obligation against which an asset of this type is held is said to be *immunized*.

The above statement has two critical limitations:

- The immunization discussed applies only to first-order approximations. When we get to a numerical example in the succeeding sections, we shall see that there is a big difference between first-order equality and "true" equality. In *Animal Farm*, George Orwell made the same observation about the barnyard: "All animals are equal, but some animals are more equal than others."
- We have assumed either that the term structure is flat or that the term structure moves up or down in parallel movements. At best, this might be considered to be a poor approximation of reality (recall the term structure graphs in

section 20.5). Alternative theories of the term structure lead to alternative definitions of duration and immunization (for alternatives, see Bierwag et al., 1981, 1983a, 1983b; Cox, Ingersoll, and Ross, 1985; Vasicek, 1977). In an empirical investigation of these alternatives, Gultekin and Rogalski (1984) found that the simple Macaulay duration we use in this chapter works at least as well as any of the alternatives.

21.3 A Numerical Example

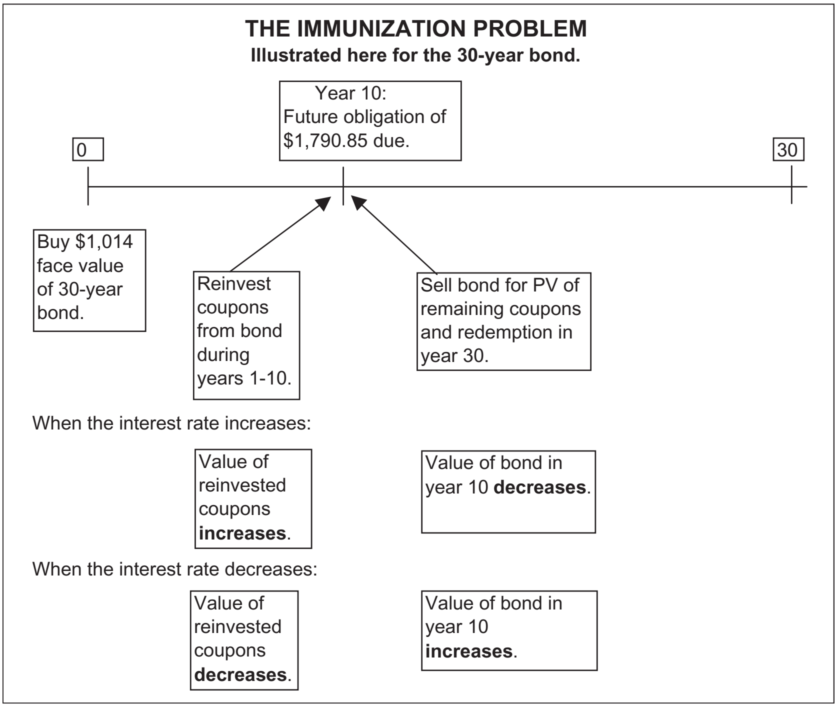
In this section we consider a basic numerical immunization example. Suppose you are trying to immunize a year-10 obligation whose present value is $1,000 (this means that, at the current interest rate of 6%, its future value is $\$1,000 \cdot (1.06)^{10} = \$1,790.85$. You intend to immunize the obligation by purchasing $1,000 worth of a bond or a combination of bonds.

You consider three bonds:

- Bond 1 has 10 years remaining until maturity, a coupon rate of 6.7%, and a face value of $1,000.
- Bond 2 has 15 years until maturity, a coupon rate of 6.988%, and a face value of $1,000.
- Bond 3 has 30 years until maturity, a coupon rate of 5.9%, and a face value of $1,000.

At the existing yield to maturity of 6%, the prices of the bonds differ. Bond 1, for example, is worth $\$1,051.52 = \sum_{t}^{10} \frac{67}{(1.06)^t} + \frac{1,000}{(1.06)^{10}}$; thus, in order to purchase $1,000 worth of this bond, you have to purchase $951 = $1,000/$1,051.52 of *face value* of the bond.

Bond 3, however, is currently worth $986.24, so that in order to buy $1,000 of market value of this bond, you will have to buy $1,013.96 of face value of the bond. If you intend to use this bond to finance a $1,790.85 obligation 10 years from now, here's a schematic of the problem you face:



As we will see below, the 30-year bond will exactly finance the future obligation of $1,790.85 only for the case in which the current market interest rate of 6% remains unchanged.

Here is a summary of price and duration information for the three bonds:

	A	B	C	D	E
1	**BASIC IMMUNIZATION EXAMPLE WITH 3 BONDS**				
2	Yield to maturity	6%			
3					
4		**Bond 1**	**Bond 2**	**Bond 3**	
5	Coupon rate	6.70%	6.988%	5.90%	
6	Maturity	10	15	30	
7	Face value	1,000	1,000	1,000	
8					
9	Bond price	$1,051.52	$1,095.96	$986.24	<-- =-PV($B$2,D6,D5*D7)+D7/(1+$B$2)^D6
10	Face value equal to $1,000 of market value	$ 951.00	$ 912.44	$ 1,013.96	<-- =D7/D9*D7
11					
12	Duration	7.6655	10.0000	14.6361	<-- =dduration(D6,D5,$B$2,1)

Note that to calculate the duration, we have used the "home-made" **DDuration** function defined in Chapter 20.

If the yield to maturity doesn't change, then you will be able to reinvest each coupon at 6%. Thus, bond 2, for example, will give a terminal wealth at the end of 10 years, of

$$\sum_{t=0}^{9} 69.88 \cdot (1.06)^t + \left[\sum_{t=1}^{5} \frac{69.88}{(1.06)^t} + \frac{1{,}000}{(1.06)^5} \right] = 921.07 + 1{,}041.62 = 1{,}962.69$$

The first term in this expression, $\sum_{t=0}^{9} 69.88 \cdot (1.06)^t$, is the sum of the reinvested coupons. The second and third terms, $\sum_{t=1}^{5} \frac{69.88}{(1.06)^t} + \frac{1{,}000}{(1.06)^5}$, represent the market value of the bond in year 10, when the bond has 5 more years until maturity. Since we will be buying only $912.44 of face value of this bond, we have, at the end of 10 years, 0.91244*$1,962.69 = $1,790.85. This is exactly the amount we wanted to have at this date. The results of this calculation for all three bonds, provided there is no change in the yield to maturity, are given in the following table:

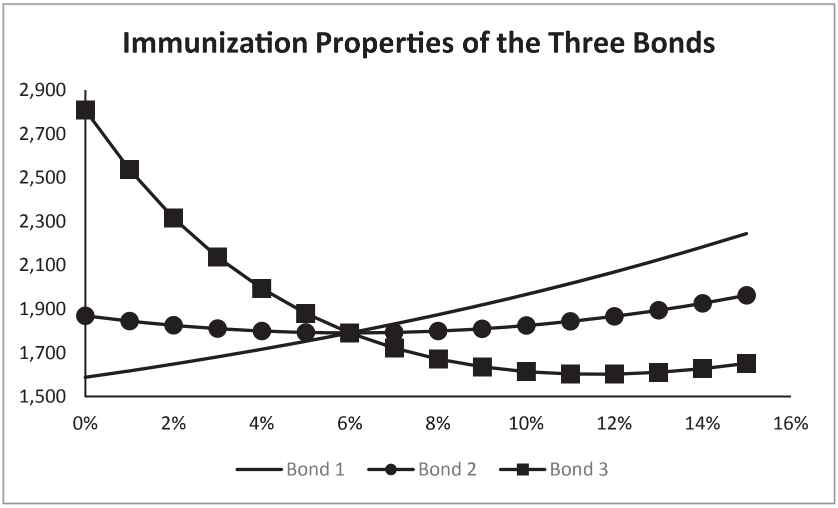
	A	B	C	D	E
14	New yield to maturity	6%			
15					
16		**Bond 1**	**Bond 2**	**Bond 3**	
17	Bond price	$1,000.00	$1,041.62	$988.53	<-- =-PV($B$14,D6-10,D5*D7)+D7/(1+$B$14)^(D6-10)
18	Reinvested coupons	$883.11	$921.07	$777.67	=-FV($B$14,10,D5*D7)
19	**Total**	$1,883.11	$1,962.69	$1,766.20	<-- =D17+D18
20					
21	Multiply by percent of face value bought	95.10%	91.24%	101.40%	<-- =D10/1000
22	**Product**	$ 1,790.85	$ 1,790.85	$ 1,790.85	<-- =D21*D19

The upshot of this table is that purchasing $1,000 of any of the three bonds will provide—10 years from now—funding for your future obligation of $1,790.85, *provided the market interest rate of 6% doesn't change*.

Now suppose that immediately after you purchase the bonds the yield to maturity changes to some new value and stays there. This will obviously affect the calculation we did above. For example, if the yield falls to 5%, the table will now look as follows:

	A	B	C	D	E
14	New yield to maturity	5%			
15					
16		**Bond 1**	**Bond 2**	**Bond 3**	
17	Bond price	$1,000.00	$1,086.07	$1,112.16	<-- =-PV($B$14,D6-10,D5*D7)+D7/(1+$B$14)^(D6-10)
18	Reinvested coupons	$842.72	$878.94	$742.10	=-FV($B$14,10,D5*D7)
19	**Total**	$1,842.72	$1,965.01	$1,854.26	<-- =D17+D18
20					
21	Multiply by percent of face value bought	95.10%	91.24%	101.40%	<-- =D10/1000
22	**Product**	$ 1,752.43	$ 1,792.97	$ 1,880.14	<-- =D21*D19

Thus, if the yield falls, bond 1 will no longer fund our obligation, whereas bond 3 will overfund it. Bond 2's ability to fund the obligation—not surprisingly, in view of the fact that its duration is exactly 10 years—hardly changes. We can repeat this calculation for any new yield to maturity. The results are shown in the figure below, which was produced by running a **Data|Table** (see Chapter 31):



Clearly, if you want an immunized strategy, you should buy bond 2!

21.4 Convexity: A Continuation of Our Immunization Experiment

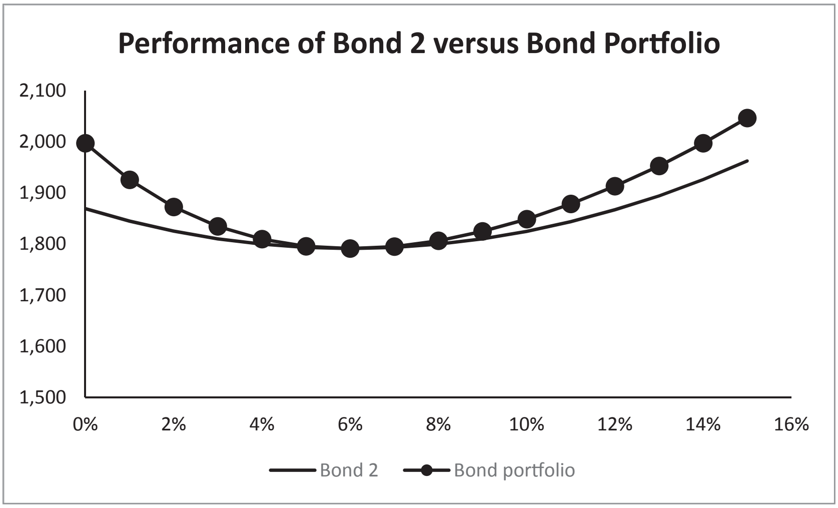
The duration of a portfolio is the weighted average duration of the assets in the portfolio. This means that there is another way to get a bond investment with a duration of 10: If we invest $665.09 in bond 1 and $344.91 in bond 3, the resulting portfolio also has a duration of 10. These weights are calculated as follows:

$$\lambda * Duration_{Bond1} + (1-\lambda) * Duration_{Bond3} = 7.6655\lambda + 14.6361(1-\lambda) = 10$$

Suppose we repeat our experiment with this portfolio of bonds. Starting in row 15 of the spreadsheet below, we repeat the experiment of the previous section (varying the YTM), but add in the portfolio of bond 1 and bond 3. The results below show that the future value in row 23 does not vary for the portfolio.

	A	B	C	D	E	F	G
1	**EXPERIMENTING WITH BOND PORTFOLIOS AND CONVEXITY**						
2	Yield to maturity (YTM)	6%					
3							
4		**Bond 1**	**Bond 2**	**Bond 3**			
5	Coupon rate	6.70%	6.988%	5.90%			
6	Maturity	10	15	30			
7	Face value	1,000	1,000	1,000			
8							
9	Bond price	$1,051.52	$1,095.96	$986.24	<-- =-PV($B$2,D6,D5*D7)+D7/(1+$B$2)^D6		
10	Face value equal to $1,000 of market value	$ 951.00	$ 912.44	$ 1,013.96	<-- =D7/D9*D7		
11							
12	Duration	7.6655	10.0000	14.6361	<-- =dduration(D6,D5,$B$2,1)		
13							
14							
15	New YTM	7%					
16							
17		**Bond 1**	**Bond 2**	**Bond 3**		**Bond 1 & 3 portfolio**	
18	Bond price	$1,000.00	$999.51	$883.47			
19	Reinvested coupons	$925.70	$965.49	$815.17			
20	**Total**	$1,925.70	$1,965.00	$1,698.64			
21							
22	Multiply by percent of face value bought	95.10%	91.24%	101.40%			
23	**Product**	$ 1,831.35	$ 1,792.95	$ 1,722.34		$ 1,794.84	<-- =B26*B23+(1-B26)*D23
24							
25	Portfolio of bonds 1 and 3						
26	Proportion of bond 1	0.6651	<-- =(10-D12)/(B12-D12)				
27	Proportion of bond 3	0.3349	<-- =1-B26				

Building a data table based on this experiment and graphing the results shows that the portfolio's performance is better than that of bond 2 by itself:



Convexity

Look again at the graph: Notice that, while for both bond 2 and the bond portfolio, the terminal value is somewhat convex in the yield to maturity, the terminal value of the portfolio is *more convex* than that of the single bond. Redington (1952), one of the influential propagators of the concept of duration and immunization, thought this convexity very desirable, and we can see why: No matter what the change in the yield to maturity, the portfolio of bonds provides *more overfunding* of the future obligation than the single bond. This is obviously a desirable property for an immunized portfolio, and it leads us to formulate the following rule:

In a comparison between two immunized portfolios, both of which are to fund a known future obligation, the portfolio whose terminal value is more convex with respect to changes in the yield to maturity is preferable.[1]

1. There is another interpretation of the convexity shown in this example: It shows the impossibility of parallel changes in the term structure! If such changes describe the uncertainty relating to the term structure, a bond position can be chosen which always benefits from changes in the term structure. This is an arbitrage, and therefore impossible. I thank Zvi Wiener for pointing this out to me.

21.5 Building a Better Mousetrap

Despite what was said in the preceding section, there is some interest in deriving the characteristics of a bond portfolio whose terminal value is as insensitive to changes in the yield as possible. One way of improving the performance (when so defined) of the bond portfolio is not only to match the first derivatives of the change in value (which, as we saw in section 20.3, leads to the duration concept), but also to match the second derivatives.

A direct extension of the analysis of section 20.3 leads us to the conclusion that matching the second derivatives requires:

$$N(N+1) = \frac{1}{V_B} \sum_{t=1}^{M} \frac{t(t+1)P_t}{(1+r)^t}$$

The following example illustrates the kind of improvement that can be made in a portfolio where the second derivatives are also matched. Consider 4 bonds, one of which, bond 2, is our old friend from the previous example, whose duration is exactly 10. The bonds are described in the following table:

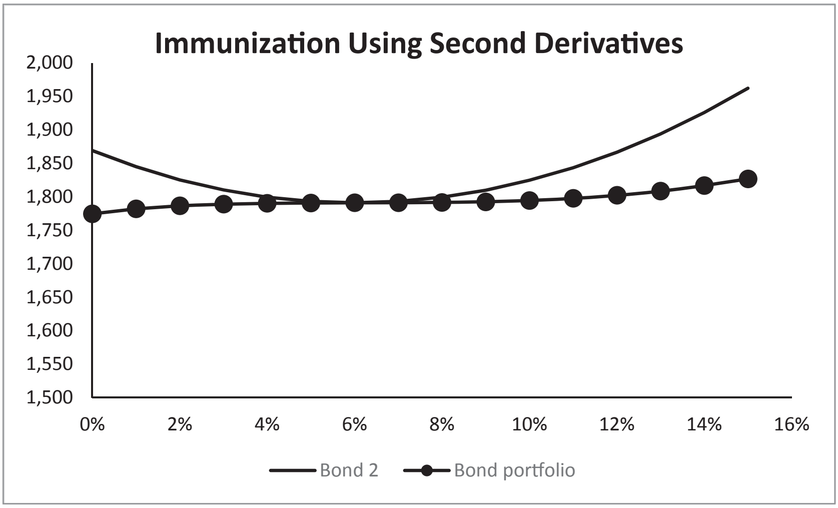
	A	B	C	D	E	F
1	BOND CONVEXITY					
2	Yield to maturity	6%				
3						
4		**Bond 1**	**Bond 2**	**Bond 3**	**Bond 4**	
5	Coupon rate	4.50%	6.988%	3.50%	11.00%	
6	Maturity	20	15	14	10	
7	Face value	1,000	1,000	1,000	1,000	
8						
9	Bond price	$827.95	$1,095.96	$767.63	$1,368.00	<-- =-PV($B$2,E6,E5*E7)+E7/(1+$B$2)^E6
10	Face value equal to $1,000 of market value	$ 1,207.80	$ 912.44	$ 1,302.72	$ 730.99	<-- =E7/E9*E7
11						
12	Duration	12.8964	10.0000	10.8484	7.0539	<-- =dduration(E6,E5,$B$2,1)
13	Second derivative of duration	229.0873	136.4996	148.7023	67.5980	<-- =secondDur(E6,E5,$B$2)/bondprice(E6,E5,$B$2)

Here `secondDur(numberPayments, couponRate, YTM)` is a VBA function we have defined to calculate the second derivative of the duration:

```
Function secondDur(numberPayments, couponRate,
YTM)

    For Index = 1 To numberPayments
        If Index < numberPayments Then
            secondDur = couponRate * Index * _
            (Index + 1) / (1 + YTM) ^ Index + _
                 secondDur
        Else
            secondDur = (couponRate + 1) * _
            Index * (Index + 1) _
                 / (1 + YTM) ^ Index + _
                 secondDur
        End If

        secondDur = secondDur
    Next Index

End Function
```

We need three bonds in order to calculate a portfolio of bonds whose duration and whose second duration derivative are exactly equal to those of the liability. The proportions of a portfolio which sets both the duration and its second derivative equal to those of the liability are bond 1 = –0.5619, bond 3 = 1.6415, bond 4 = –0.0797.[2] As the following figure shows, this portfolio provides a better hedge against the terminal value than even bond 2:

2. See next subsection for the details of this computation.



Computing the Bond Portfolio

We want to invest proportions x_1, x_3, x_4 in bonds 1, 3, and 4 so that:

- The portfolio is totally invested: $x_1 + x_3 + x_4 = 1$.
- The portfolio duration is matched to that of bond 2: $x_1D_1 + x_3D_3 + x_4D_4 = D_2$, where D_i is the duration of bond i.
- The second derivative of the portfolio duration is matched to that of bond 2: $x_1D_1^2 + x_3D_3^2 + x_4D_4^2 = D_2^2$, where D_i^2 is the duration derivative.

Writing this in matrix form, we get:

$$\begin{bmatrix} 1 & 1 & 1 \\ D_1 & D_3 & D_4 \\ D_1^2 & D_3^2 & D_4^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ D_2 \\ D_2^2 \end{bmatrix}$$

whose solution is given by:

$$\begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ D_1 & D_3 & D_4 \\ D_1^2 & D_3^2 & D_4^2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ D_2 \\ D_2^2 \end{bmatrix}$$

This can easily be set up in Excel:

	I	J	K	L	M	N
15	**Calculating the bond portfolio:**					
16					Vector of	
17	Matrix of coefficients				constants	
18	1	1	1		1	
19	12.8964	10.8484	7.0539		10.0000	
20	229.0873	148.7023	67.5980		110.0000	
21						
22	Solution					
23	-0.5619					
24	1.6415	<-- {=MMULT(MINVERSE(I18:K20),M18:M20)}				
25	-0.0797					
26						
27						
28	Explanation of the above: We want to invest proportions					
29	x_1, x_3, and x_4 in bonds 1, 3 and 4 respectively, in order					
30	that: a) The total investment is $1000; this means $x_1+x_2+x_4=1$					
31	b) Portfolio duration is matched to that of bond 2; this means					
32	that $x_1{*}D_1+x_3{*}D_3+x_4{*}D_4 = D_2$, where D_i is the duration					
33	of bond I.					
34	c) The weighted average duration derivatives are equal					
35	to that of bond 2.					
36						
37	These three conditions give us the matrix system in					
38	cells I18:K20 and the corresponding solution in					
39	cells I23:I25 .					

Given this solution, the last chart is produced by the following data table:

	A	B	C	D	E	F
26	**Data table: Sensitivity of Bond 2 and bond portfolio terminal values to interest rate**		**Bond 2**	**Bond portfolio**		
27					<-- =I23*B23+I24*D23+I25*E23 , data table header (hidden)	
28		0%	$ 1,868.87	$ 1,774.63		
29		1%	$ 1,844.71	$ 1,781.79		
30		2%	$ 1,825.14	$ 1,786.37		
31		3%	$ 1,810.05	$ 1,789.02		
32		4%	$ 1,799.35	$ 1,790.32		
33		5%	$ 1,792.97	$ 1,790.78		
34		6%	$ 1,790.85	$ 1,790.85		
35		7%	$ 1,792.95	$ 1,790.91		
36		8%	$ 1,799.26	$ 1,791.31		
37		9%	$ 1,809.76	$ 1,792.38		
38		10%	$ 1,824.46	$ 1,794.38		
39		11%	$ 1,843.37	$ 1,797.58		
40		12%	$ 1,866.53	$ 1,802.21		
41		13%	$ 1,893.98	$ 1,808.46		
42		14%	$ 1,925.77	$ 1,816.55		
43		15%	$ 1,961.98	$ 1,826.65		

21.6 Summary

The value of an immunized portfolio of bonds is insensitive to small changes in the underlying yield to maturity of the bonds. Immunization involves setting the bond portfolio's duration equal to the duration of the underlying liability against which the portfolio is held. This chapter shows how to effect the immunization of the portfolio. Needless to say, Excel is an excellent tool for immunization calculations.

Exercises

1. Prove that the duration of a portfolio is the weighted average duration of the portfolio assets.
2. Set up a spreadsheet that enables you to duplicate the calculations of section 21.5 of this chapter.
3. Using the example of section 21.5, find a combination of bonds 1 and 3 with a duration of 8. Then find a combination of bonds 1 and 2 with a duration of 8.
4. In exercise 3, which portfolio would you prefer to immunize an obligation with a duration of 8?
5. In exercise 3, recalculate the portfolio proportions assuming that you need a target duration of 12. Which portfolio would you prefer now?

22 Modeling the Term Structure*

22.1 Overview

This chapter discusses the problem of fitting an equation to the term structure of interest rate. The main problem looks like this: We are given a set of bond prices. Can we say something intelligent about the yields on these bonds, for example, in terms of time to maturity or in terms of yields versus duration?

The problem is surprisingly complicated. It is obvious that we need to take into account the riskiness of the bonds (the set of bonds should be of equal risk) and their time pattern of interest payments. There is an additional complicating factor: The most common interest rate measure for bonds is the yield to maturity (YTM), essentially the internal rate of return of the bond price and the future promised payments on the bond. For analytical purposes, however, it is more meaningful to attach a discount factor d_t to payments made in each time period. These discount factors define the so-called *pure discount yields* on the bonds.[1]

This chapter starts with the simplest term structure problem, where there is only one bond for each time period. We then go on to discuss more complicated situations. The functional form we fit to the term structure is the Nelson-Siegel model (1987). We also show a variation of this model due to Svensson (section 22.8).

22.2 Basic Example

In this section we introduce two methods of pricing bonds in the case where each bond maturity is associated with a single bond.[2] The two methods are

- Computing the yield to maturity of each bond. Each bond's yield to maturity (YTM) is the internal rate of return of the bond price and the future payments.

---

* This chapter was co-authored by Dr. Alexander Suhov of Tel Aviv University, a.y.suhov@gmail.com.

1. Those familiar with the Treasury bond strip market will recognize the factors d_t as the discount factors associated with Treasury strips. See www.treasurydirect.gov/instit/marketables/strips/strips.htm.

2. In the next section we discuss the case where there are multiple bonds for the same maturity with possibly inconsistent pricing.

• Computing a set of unique, time-dependent, discount factors for the bonds. Labeling the discount factor for time t by d_t, a bond's price is computed by $Price = \sum_{t=1}^{N} C_t d_t$, where C_t is the promised bond payment at time t.

To illustrate these two methods, consider 15 bonds, each of which pays an annual coupon until maturity and each of which has a face value of 100. The bonds have a maturity of 1, 2, …, 15. In the spreadsheet clip below we show the table of bonds, their coupon rates, and their YTMs:

	A	B	C	D	E	F
1	**INITIAL EXAMPLE**					
2	Bond	Price	Maturity	Annual coupon rate		YTM
3	1	96.60	1	2.0%		5.59%
4	2	93.71	2	2.5%		5.93%
5	3	91.56	3	3.0%		6.17%
6	4	90.24	4	3.5%		6.34%
7	5	89.74	5	4.0%		6.47%
8	6	90.04	6	4.5%		6.56%
9	7	91.09	7	5.0%		6.63%
10	8	92.82	8	5.5%		6.69%
11	9	95.19	9	6.0%		6.73%
12	10	98.14	10	6.5%		6.76%
13	11	101.60	11	7.0%		6.79%
14	12	105.54	12	7.5%		6.81%
15	13	109.90	13	8.0%		6.83%
16	14	114.64	14	8.5%		6.84%
17	15	119.73	15	9.0%		6.85%

The YTMs are computed by putting the bond payments into a triangular matrix, part of which is shown below.

	F	G	H	I	J	K	L	M	N
1		=IF(I$2<$C3,$D3*100,IF(I$2=$C3,(1+$D3)*100,0))							
2	YTM	Interest rate y_t	0	1	2	3	4	5	6
3	5.59%	1	-96.60	102.00	0.00	0.00	0.00	0.00	0.00
4	5.93%	2	-93.71	2.50	102.50	0.00	0.00	0.00	0.00
5	6.17%	3	-91.56	3.00	3.00	103.00	0.00	0.00	0.00
6	6.34%	4	-90.24	3.50	3.50	3.50	103.50	0.00	0.00
7	6.47%	5	-89.74	4.00	4.00	4.00	4.00	104.00	0.00
8	6.56%	6	-90.04	4.50	4.50	4.50	4.50	4.50	104.50
9	6.63%	7	-91.09	5.00	5.00	5.00	5.00	5.00	5.00
10	6.69%	8	-92.82	5.50	5.50	5.50	5.50	5.50	5.50
11	6.73%	9	-95.19	6.00	6.00	6.00	6.00	6.00	6.00
12	6.76%	10	-98.14	6.50	6.50	6.50	6.50	6.50	6.50
13	6.79%	11	-101.60	7.00	7.00	7.00	7.00	7.00	7.00
14	6.81%	12	-105.54	7.50	7.50	7.50	7.50	7.50	7.50
15	6.83%	13	-109.90	8.00	8.00	8.00	8.00	8.00	8.00
16	6.84%	14	-114.64	8.50	8.50	8.50	8.50	8.50	8.50
17	6.85%	15	-119.73	9.00	9.00	9.00	9.00	9.00	9.00
18									
19		<-- =IRR(H17:W17)							

There is another way to interpret the data in our basic example. Suppose that each time period t has its own discount factor d_t. Then the prices of the bonds could be written as the discounted prices of bond payments:

$$\text{Bond1: } 96.6 = 102 d_1$$
$$\text{Bond2: } 93.71 = 2.5 d_1 + 102.5 d_2$$
$$\text{Bond3: } 91.56 = 3 d_1 + 3 d_2 + 103 d_3$$

....

In general: $Price = \sum_{t=1}^{N} C_t d_t$

Using matrices, we can solve for the discount factors d_t:

$$\begin{bmatrix} C_{11} & 0 & 0 & 0 & 0 & \dots & 0 \\ C_{21} & C_{22} & 0 & 0 & 0 & \dots & 0 \\ \dots & & & & & \dots & \dots \\ C_{N1} & C_{N2} & & & & & C_{NN} \end{bmatrix} * \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} Price_1 \\ Price_2 \\ \vdots \\ Price_N \end{bmatrix}$$

Solving this system of equations gives

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} C_{11} & 0 & 0 & 0 & 0 & \dots & 0 \\ C_{21} & C_{22} & 0 & 0 & 0 & \dots & 0 \\ \dots & & & & & \dots & \dots \\ C_{N1} & C_{N2} & & & & & C_{NN} \end{bmatrix}^{-1} * \begin{bmatrix} Price_1 \\ Price_2 \\ \vdots \\ Price_N \end{bmatrix}$$

This computation is easily done in Excel:

	F	G	H	I	J	K	L	M	N
1		=IF(I$2<$C3,$D3*100,IF(I$2=$C3,(1+$D3)*100,0))							
2	Discount factors d_t	Interest rate y_t	0	1	2	3	4	5	6
3	0.9471	1	-96.60	102.00	0.00	0.00	0.00	0.00	0.00
4	0.8911	2	-93.71	2.50	102.50	0.00	0.00	0.00	0.00
5	0.8354	3	-91.56	3.00	3.00	103.00	0.00	0.00	0.00
6	0.7815	4	-90.24	3.50	3.50	3.50	103.50	0.00	0.00
7	0.7300	5	-89.74	4.00	4.00	4.00	4.00	104.00	0.00
8	0.6814	6	-90.04	4.50	4.50	4.50	4.50	4.50	104.50
9	0.6358	7	-91.09	5.00	5.00	5.00	5.00	5.00	5.00
10	0.5930	8	-92.82	5.50	5.50	5.50	5.50	5.50	5.50
11	0.5530	9	-95.19	6.00	6.00	6.00	6.00	6.00	6.00
12	0.5157	10	-98.14	6.50	6.50	6.50	6.50	6.50	6.50
13	0.4809	11	-101.60	7.00	7.00	7.00	7.00	7.00	7.00
14	0.4484	12	-105.54	7.50	7.50	7.50	7.50	7.50	7.50
15	0.4181	13	-109.90	8.00	8.00	8.00	8.00	8.00	8.00
16	0.3898	14	-114.64	8.50	8.50	8.50	8.50	8.50	8.50
17	0.3635	15	-119.73	9.00	9.00	9.00	9.00	9.00	9.00
18									
19	{=MMULT(MINVERSE(I3:W17),-H3:H17)}								

Pricing Advantages of the Discount Factors

The advantage of the discount factors is that they allow the accurate pricing of any other bond with the same time pattern of payments. Consider, for example, a 5-period bond with a coupon of 3%. The price of this bond, using the current term structure, is 85.5549:

	G	H	I	J	K	L	M
21	Time -->	0	1	2	3	4	5
22	New bond	85.5549	3	3	3	3	103
23		<-- {=MMULT(I22:W22,F3:F17)}					

From Discount Factors to a Term Structure

The discount factors determine a *zero-coupon term structure*: $d_t = \exp(-y_t * t)$, where y_t is the continuously compounded pure-discount rate for time t. Solving the discount factors for the zero-coupon interest rate gives

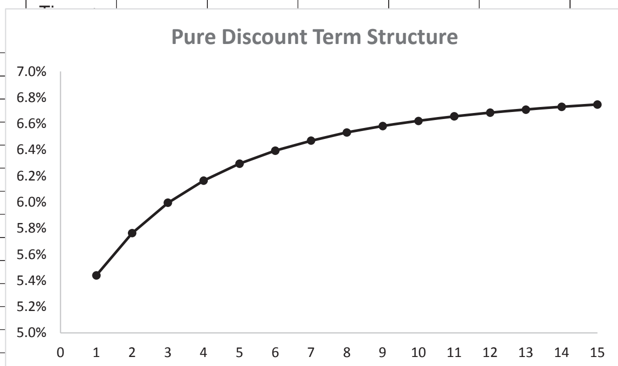
$$y_t = \ln(d_t)/t$$

This means that we can write our bond pricing equation in terms of discount rates instead of discount factors:

$$Price = \sum_{t=1}^{N} C_t d_t = \sum_{t=1}^{N} C_t e^{-y_t t}$$

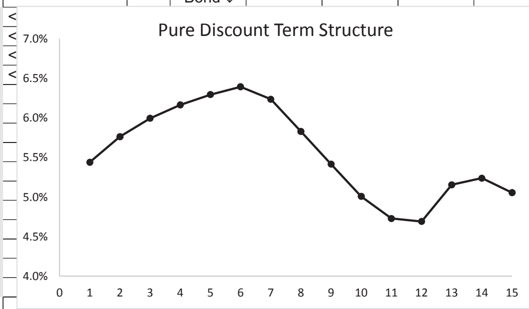
Applying this to our example:

	F	G	H	I	J	K	L	M	N	O	P
2	Discount factors d_t	Interest rate y_t									
3	0.9471	5.44%	<-- =-LN(F3)/C3								0.00
4	0.8911	5.76%	<-- =-LN(F4)/C4								0.00
5	0.8354	6.00%	<-- =-LN(F5)/C5								0.00
6	0.7815	6.16%	<-- =-LN(F6)/C6								0.00
7	0.7300	6.29%									4.00
8	0.6814	6.39%									4.50
9	0.6358	6.47%									5.00
10	0.5930	6.53%									5.50
11	0.5530	6.58%									6.00
12	0.5157	6.62%									6.50
13	0.4809	6.66%									7.00
14	0.4484	6.68%									7.50
15	0.4181	6.71%									8.00
16	0.3898	6.73%									8.50
17	0.3635	6.75%			15	-119.73	9.00	9.00	9.00	9.00	9.00
18											
19	{=MMULT(MINVERSE(L3:Z17),-K3:K17)}										



Different prices or coupons for bonds can produce quite intricate term structures. In the example below we compute the discount factors and interest rates for different sets of annual coupon rates:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2	Bond	Price	Maturity	Annual coupon rate		Discount factors	Interest rate			Time--> Bond ↓	0	1	2	3
3	1	96.60	1	2.0%		0.9471	5.44%	<						00
4	2	93.71	2	2.5%		0.8911	5.76%	<						00
5	3	91.56	3	3.0%		0.8354	6.00%	<						00
6	4	90.24	4	3.5%		0.7815	6.16%	<						50
7	5	89.74	5	4.0%		0.7300	6.29%							00
8	6	90.04	6	4.5%		0.6814	6.39%							50
9	7	91.09	7	4.8%		0.6463	6.24%							30
10	8	92.82	8	4.9%		0.6273	5.83%							90
11	9	95.19	9	5.0%		0.6142	5.42%							00
12	10	98.14	10	5.1%		0.6060	5.01%							10
13	11	101.60	11	5.3%		0.5944	4.73%							30
14	12	105.54	12	5.7%		0.5695	4.69%							70
15	13	109.90	13	6.5%		0.5117	5.15%							50
16	14	114.64	14	7.0%		0.4803	5.24%							00
17	15	119.73	15	7.3%		0.4684	5.06%							30
18														
19						{=MMULT(MINVERSE(L3:Z17),-K3:K17)}								



22.3 Several Bonds with the Same Maturity

In the previous section there was only one bond for each maturity date. In real data situations, there are often a number of bonds with similar maturities and possibly inconsistent pricing. In bond markets, which are typified by thin trading and often-misreported prices, this is a common occurrence. Suppose, as in the next example, we have several bonds with similar maturities. In the example below there are two bonds for 3-, 6-, and 9-year maturities, each with a slightly different YTM.

	A	B	C	D	E	F
1	**MULTIPLE BONDS WITH SAME MATURITY**					
2	Bond	Price	Maturity	Annual coupon rate		YTM
3	1	91.8967	1	2.0%		10.99%
4	2	83.2564	2	2.5%		12.47%
5	3	76.0000	3	3.0%		13.20%
6	4	76.2347	3	3.2%		13.32%
7	5	71.2110	4	3.5%		13.22%
8	6	67.9672	5	4.0%		13.14%
9	7	66.0000	6	4.5%		13.01%
10	8	66.1625	6	4.2%		12.56%
11	9	65.4881	7	5.0%		12.74%
12	10	65.7003	8	5.5%		12.53%
13	11	64.0000	9	5.8%		12.75%
14	12	66.6158	9	6.0%		12.35%
15	13	68.0989	10	6.5%		12.19%
16	14	70.0480	12	7.0%		11.79%
17	15	72.3857	15	7.5%		11.43%
18						
19					=IRR(H17:W17)	

The matrix of cash flows is no longer square, since there are now 15 bonds with 12 maturities. In section 22.2 we found the discount factors d_t by inverting the matrix of payments, but in this case this matrix is not invertible, since it is not square:

	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
2	YTM	Time--> Bond ↓	0	1	2	3	4	5	6	7	8	9	10	11	12
3	10.99%	1	-91.90	102.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	12.47%	2	-83.26	2.50	102.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	13.20%	3	-76.00	3.00	3.00	103.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	13.32%	4	-76.23	3.20	3.20	103.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	13.22%	5	-71.21	3.50	3.50	3.50	103.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	13.14%	6	-67.97	4.00	4.00	4.00	4.00	104.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	13.01%	7	-66.00	4.50	4.50	4.50	4.50	4.50	104.50	0.00	0.00	0.00	0.00	0.00	0.00
10	12.56%	8	-66.16	4.20	4.20	4.20	4.20	4.20	104.20	0.00	0.00	0.00	0.00	0.00	0.00
11	12.74%	9	-65.49	5.00	5.00	5.00	5.00	5.00	5.00	105.00	0.00	0.00	0.00	0.00	0.00
12	12.53%	10	-65.70	5.50	5.50	5.50	5.50	5.50	5.50	5.50	105.50	0.00	0.00	0.00	0.00
13	12.75%	11	-64.00	5.80	5.80	5.80	5.80	5.80	5.80	5.80	5.80	105.80	0.00	0.00	0.00
14	12.35%	12	-66.62	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00	106.00	0.00	0.00	0.00
15	12.19%	13	-68.10	6.50	6.50	6.50	6.50	6.50	6.50	6.50	6.50	6.50	106.50	0.00	0.00
16	12.06%	14	-70.05	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	107.00	0.00
17	11.95%	15	-72.39	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	7.50	107.50

We can find the discount factors by using a least squares approximation: For each maturity t, we find a factor d_t so that the square of the pricing errors for the set of bonds is minimized. The least squares approximation is usually employed to determine the approximate solution of overdetermined systems, that is, sets of equations in which the equations outnumber the unknowns. "Least squares" means that the overall solution minimizes the sum of the squares of the errors made in the results of every equation.

We want to solve for a vector of discount factors d that best approximates the pricing of the bonds.

$$[Cashflows] * \begin{bmatrix} Discount \\ factors \end{bmatrix} = \begin{bmatrix} Bond \\ Prices \end{bmatrix}$$

$$\begin{bmatrix} 102 & & & & & & & & \\ 2.5 & 102.5 & & & & & & & \\ 3 & 3 & 103 & & & & & & \\ \vdots & & & & & & & & \\ 5.8 & 5.8 & & & 5.8 & 105.8 & & & \\ 6 & 6 & & & 6 & 106 & & & \\ 6.5 & 6.5 & 6.5 & & & 6.6 & 106.5 & & \\ 7 & 7 & 7 & 7 & & & 7 & 107 & \\ 7.5 & 7.5 & 7.5 & 7.5 & & & 7.5 & 7.5 & 107.5 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \\ \vdots \\ \\ \\ \\ \\ d_{12} \end{bmatrix} = \begin{bmatrix} 91.8967 \\ 83.2564 \\ 76.0000 \\ \vdots \\ \\ \\ \\ \\ 72.3857 \end{bmatrix}$$

The formula for the least squares approximation is

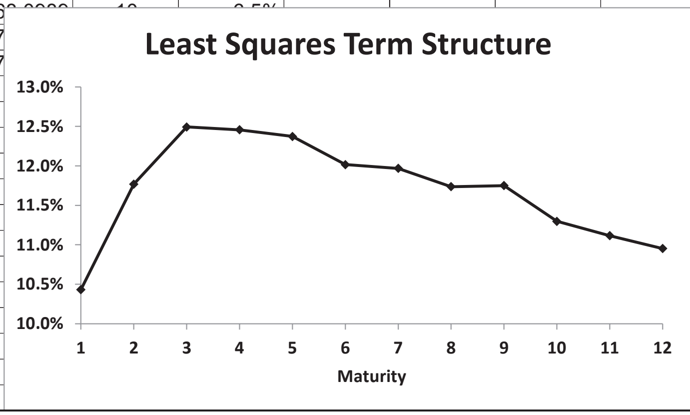
$$d = \left(Cashflows^T * Cashflows\right)^{-1} * \left(Cashflows^T * Prices\right)$$

In Excel this formula becomes

$$\mathbf{d = MMult}\begin{pmatrix} \mathbf{Minverse(MMult(Transpose(Cashflows), Cashflows),} \\ \mathbf{MMult(Transpose(Cashflows), Prices)} \end{pmatrix}$$

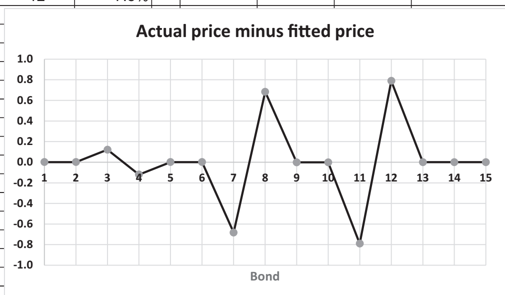
Implementing this on our example:

	A	B	C	D	E	F	G	H	I
1	**LEAST SQUARES SOLUTION TO TERM STRUCTURE**								
2						Least squares and term structure			
3	Bond #	Price	Maturity (years)	Annual coupon rate		Maturity	d(t)	y(t)	
4	1	91.8967	1	2.0%		1	0.9009	10.43%	<-- =-LN(G4)/F4
5	2	83.2564	2	2.5%		2	0.7903	11.77%	
6	3	76.0000	3	3.0%		3	0.6874	12.49%	
7	4	76.2347	3	3.2%		4	0.6076	12.46%	
8	5	71.2110	4	3.5%		5	0.5387	12.37%	
9	6	67.9672	5	4.0%		6	0.4863	12.01%	
10	7	66.0000	6	4.5%		7	0.4327	11.97%	
11	8	66.1625	6	4.2%		8	0.3911	11.74%	
12	9	65.4881	7	5.0%		9	0.3473	11.75%	
13	10	65.7003	8	5.5%		10	0.3231	11.30%	
14	11	64.0000	9	5.8%		11	0.2945	11.11%	
15	12	66.6158	9	6.0%		12	0.2687	10.95%	
16	13	6[illegible]	[illegible]	[illegible]					
17	14	7[illegible]							
18	15	7[illegible]							
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									



The fit of discount factors to bond prices will no longer (as in the previous section) be precise. Below we compare the actual bond prices to the fitted prices:

	A	B	C	D	E	F	G	H	I	J	K
1	**COMPARING FITTED TO ACTUAL PRICES**										
2						Least squares and term structure					
3	Bond #	Actual price	Maturity (years)	Annual coupon rate		Maturity	d(t)	y(t)		Fitted price	
4	1	91.8967	1	2.0%		1	0.9009	10.43%	<-- =-LN(G4)/F4	91.8959	<-- {=MMULT(O4:Z4,$G$4:$G$15)}
5	2	83.2564	2	2.5%		2	0.7903	11.77%		83.2556	<-- {=MMULT(O5:Z5,$G$4:$G$15)}
6	3	76.0000	3	3.0%		3	0.6874	12.49%		75.8791	<-- {=MMULT(O6:Z6,$G$4:$G$15)}
7	4	76.2347	3	3.2%		4	0.6076	12.46%		76.3548	
8	5	71.2110	4	3.5%		5	0.5387	12.37%		71.2104	
9	6	67.9672	5	4.0%		6	0.4863	12.01%		67.9665	
10	7	66.0000	6	4.5%		7	0.4327	11.97%		66.6826	
11	8	66.1625	6	4.2%		8	0.3911	11.74%		65.4793	
12	9	65.4881	7	5.0%		9	0.3473	11.75%		65.4895	
13	10	65.7003	8	5.5%		10	0.3231	11.30%		65.7017	
14	11	64.0000	9	5.8%		11	0.2945	11.11%		64.7904	
15	12	66.6158	9	6.0%		12	0.2687	10.95%		65.8269	
16	13	68.0989	10	6.5%						68.0989	
17	14	70.0480	11	7.0%						70.0480	
18	15	72.3857	12	7.5%						72.3857	



22.4 Fitting a Functional Form to the Term Structure

The term structure determined in the previous section may suit our purposes. We may, however, want to fit a functional form to the term structure. The advantage of a functional form is that it allows us to interpolate interest rates for time periods and coupons that are not in our data set. It also allows us to determine the sensitivity of the term structure to structural factors.

One popular fitted form for the term structure is the Nelson-Siegel (NS) term structure, which postulates that

$$y(t)=\alpha_1+\alpha_2\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right)\right)+\alpha_3\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right)-e^{-t/\beta}\right)$$

This model can also be written as:

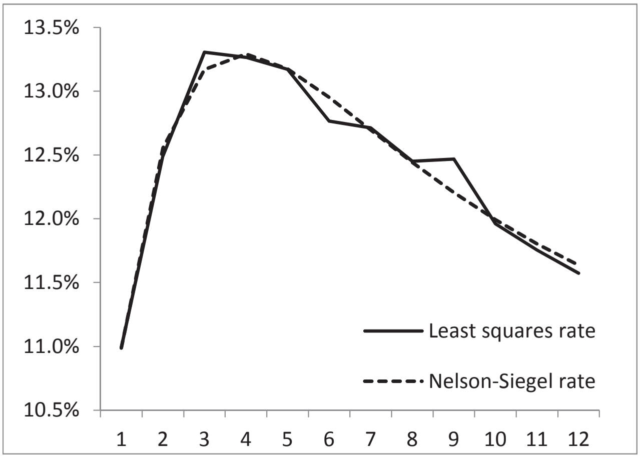
$$y(t)=\alpha_1+(\alpha_2+\alpha_3)\beta\left(\frac{1-e^{-t/\beta}}{t}\right)-\alpha_3 e^{-t/\beta}$$

In the next section we analyze the NS term structure model and discuss the meaning and the plausible values of its parameters (α_1, α_2, α_3, β). In this section we skip these analytics of the model and show how to fit NS term structure to the discount factors of the example of the previous section.[3] Our end result is given below (explanations to follow):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**FITTING NELSON-SIEGEL**												
2	**Bond #**	**Price**	**Maturity (years)**	**Annual coupon rate**	**Maturity**	**$d^{LS}(t)$**	**$y^{LS}(t)$**	**NS rate**	**Error**				
3	1	91.8967	1	2.0%	1	0.90094	11.00%	10.99%	5.0E-09		α1	0.09622	
4	2	83.2564	2	2.5%	2	0.79028	12.49%	12.56%	4.4E-07		α2	-0.0198	
5	3	76.0000	3	3.0%	3	0.68743	13.31%	13.17%	1.9E-06		α3	0.15235	
6	4	76.2347	3	3.2%	4	0.60759	13.27%	13.29%	7.4E-08		β	1.84868	
7	5	71.2110	4	3.5%	5	0.53867	13.17%	13.18%	1.6E-09		Error	1.4E-05	<-- =SUM(I3:I14)
8	6	67.9672	5	4.0%	6	0.48632	12.77%	12.95%	3.5E-06		α1+α2	0.0764	
9	7	66.0000	6	4.5%	7	0.4327	12.71%	12.70%	2.5E-08				
10	8	66.1625	6	4.2%	8	0.39109	12.45%	12.44%	7.5E-09				
11	9	65.4881	7	5.0%	9	0.34733	12.47%	12.21%	6.9E-06				
12	10	65.7003	8	5.5%	10	0.32313	11.96%	11.99%	1.1E-07				
13	11	64.0000	9	5.8%	11	0.29448	11.76%	11.80%	2.4E-07				
14	12	66.6158	9	6.0%	12	0.26871	11.57%	11.64%	4.1E-07	<-- =(G14-H14)^2			
15	13	68.0989	10	6.5%									
16	14	70.0480	11	7.0%									
17	15	72.3857	12	7.5%		=F14^(-1/G14)-1				=NSrate($L$3,$L$4,$L$5,$L$6,G14)			
18													
19					{=LS(C21:N35,B3:B17)}								

3. Charles R. Nelson and Andrew F. Siegel, "Parsimonious Modeling of Yield Curves," *Journal of Business*, 1987. Fitting the NS term structure to a set of bond data is an art! Subsequent sections of this chapter discuss alternative methods to the one described in this section.

The graph below shows the discount factors and the fitted NS term structure:



Computational Procedure

Here's how we computed the NS term structure above:

Step 1: We start by computing the discount factors d_t and the corresponding yields $y(t)$ for the data in a method similar to that discussed in sections 22.2 and 22.3. In the example above we label these $d^{LS}(t)$ and $y^{LS}(t)$.

Step 2: We now assume arbitrary but reasonable values for α_1, α_2, α_3, and β. In the next section we discuss what these values might be.

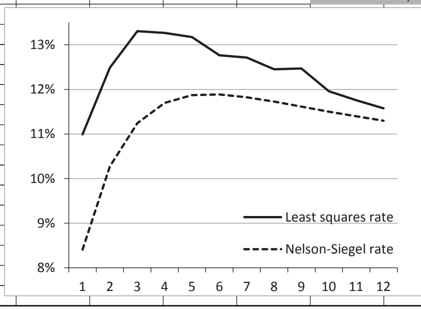
Step 3: Given the arbitrary starting values for α_1, α_2, α_3, and β, we have the NS discount zero coupon yields

$$y^{NS}(t) = \alpha_1 + \alpha_2\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right)\right) + \alpha_3\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right) - e^{-t/\beta}\right)$$

Step 4: We now optimize α_1, α_2, α_3, and β to minimize the sum of the squares between the NS and the LS: $\sum_{t=1}^{N}\left[y^{LS}(t) - y^{NS}(t)\right]^2$, where N is the number of bonds in our sample.

Here is an example: Suppose that in our example we initially set (α_1, α_2, α_3, β) = (0.1, –0.05, 0.13, 2). This gives the calculations below:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**FITTING NELSON-SIEGEL**												
2	**Bond #**	**Price**	**Maturity (years)**	**Annual coupon rate**	**Maturity**	**$d^{LS}(t)$**	**$y^{LS}(t)$**	**NS rate**	**Error**				
3	1	91.8967	1	2.0%	1	0.90094	11.00%	8.41%	6.7E-04		α1	0.1	
4	2	83.2564	2	2.5%	2	0.79028	12.49%	10.27%	4.9E-04		α2	-0.05	
5	3	76.0000	3	3.0%	3	0.68743	13.31%	11.24%	4.3E-04		α3	0.13	
6	4	76.2347	3	3.2%	4	0.60759	13.27%	11.70%	2.5E-04		β	2	
7	5	71.2110	4	3.5%	5	0.53867	13.17%	11.87%	1.7E-04		Error	0.00232	<-- =SUM(I3:I14)
8	6	67.9672	5	4.0%	6	0.48632	12.77%	11.89%	7.7E-05		α1+α2	0.05	<-- =L3+L4
9	7	66.0000	6	4.5%	7	0.4327	12.71%	11.82%	7.9E-05				
10	8	66.1625	6	4.2%	8	0.39109	12.45%	11.73%	5.3E-05				
11	9	65.4881	7	5.0%	9	0.34733	12.47%	11.61%	7.3E-05				
12	10	65.7003	8	5.5%	10	0.32313	11.96%	11.50%	2.1E-05				
13	11	64.0000	9	5.8%	11	0.29448	11.76%	11.40%	1.3E-05				
14	12	66.6158	9	6.0%	12	0.26871	11.57%	11.30%	7.6E-06	<-- =(G14-H14)^2			
15	13	68.0989	10	6.5%									
16	14	70.0480	11	7.0%			=F14^(-1/E14)-1						
17	15	72.3857	12	7.5%									
18					{=LS(C37:N51,B3:B17)}				=NSrate($L$3,$L$4,$L$5,L				
19									6,E14)				
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													

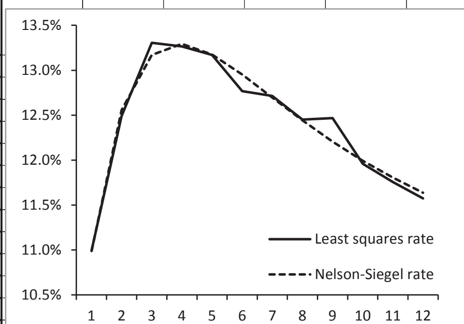


Here are some explanations:

- Columns A–D are the data.
- Column F gives the least squares fit of the discount factors for each maturity.
- Column G translates this LS fit to a pure-discount interest rate.
- Column H gives the Nelson-Siegel rate corresponding to our current factors (α_1, α_2, α_3, β) = (0.1, –0.05, 0.13, 2).
- Column I squares the difference between the LS rate and the NS rate. The summed squared difference is in cell L7.

We now use Excel's **Solver** to find (α_1, α_2, α_3, β) that minimizes this summed squared difference. Here's the dialog box:

The value in cell L6 is for β. As we will see in the next section, beta controls the "hump" in the term structure and must be positive. Here's the solution produced by **Solver**:



	K	L	M
3	α1	0.09622	
4	α2	-0.0198	
5	α3	0.15235	
6	β	1.84868	
7	Error	1.4E-05	<-- =SUM(I3:I14)
8	α1+α2	0.0764	<-- =L3+L4
14	14-H14)^2		

22.5 The Properties of the Nelson-Siegel Term Structure

Fitting the Nelson-Siegel (NS) term structure to data requires that we determine reasonable initial values for the parameters (α_1, α_2, α_3, β). In this section we examine the NS to explain what these values might be.

The NS zero coupon yield is

$$y(t) = \alpha_1 + \alpha_2\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right)\right) + \alpha_3\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right) - e^{-t/\beta}\right)$$

$$= \alpha_1 + (\alpha_2 + \alpha_3)\left(\beta\left(\frac{1-e^{-t/\beta}}{t}\right)\right) - \alpha_3 e^{-t/\beta}$$

NS Property 1: The Shortest-Term Rate $y(0)$

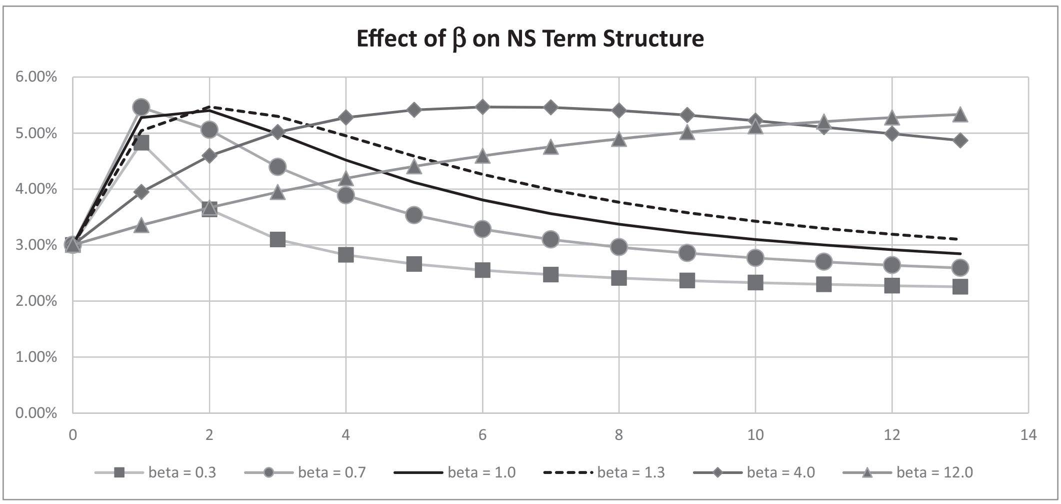
Setting $t = 0$ gives $y(0) = \alpha_1 + \alpha_2$. This is the NS shortest-term rate. It follows that for most term structures $\alpha_1 + \alpha_2 > 0$.

NS Property 2: The Longest-Term Rate $y(\infty)$

Setting $t = \infty$ gives $y(\infty) = \alpha_1$. This is the asymptotic long-term interest rate in the NS model.

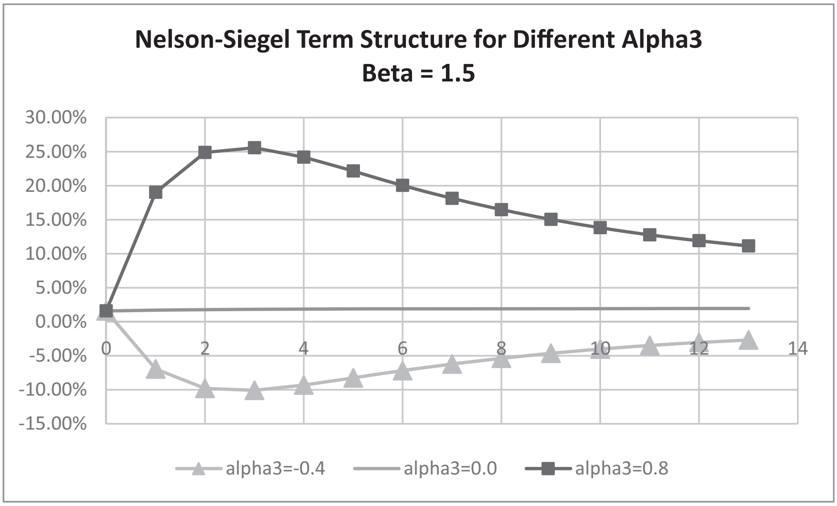
NS Property 3: β Controls the Location of the Term Structure Hump

The NS term structure is defined only if $\beta > 0$. If the term structure has a "hump," this can be controlled by the β. In the graph below, we illustrate this. Large betas (e.g., $\beta = 12$, below) produce upward-sloping term structures, whereas lower β's produce humpy term structures. Increasing the beta pushes the hump out. Roughly speaking, β is the location of the hump.



NS Property 4: α_3 Affects the Hump

The factor α_3 has no influence at either very short or very long yields and contributes to the term structure at medium-term yields only. Very roughly speaking, $\alpha_3 > 0$ produces a concave term structure, $\alpha_3 < 0$ produces a convex term structure, and $\alpha_3 = 0$ produces a flat term structure:



Nelson-Siegel: Summing Up

Fitting a term structure to the NS model involves setting initial values for (α_1, α_2, α_3, β). Plausible initial values can be set by "eyeballing" the empirical term structure and setting:

- $\alpha_1 + \alpha_2$ = approximate zero-term interest rate.
- α_1 = approximate long-term rate.
- β = approximate location of the hump.
- α_3 affects the concavity/convexity of the term structure.

22.6 Term Structure for Treasury Notes

In this section we fit the Nelson-Siegel (NS) model to the prices of Treasury notes. This is a much larger data set than the one discussed in the previous section, and it allows us to illustrate a different optimizing technique.

We start with the data set of prices for Treasury notes (i.e., bonds with maturities of less than 10 years). The date of the data is 31 March 1989.

	A	B	C	D	E	F	G	H	I	J	K
1	127 Treasury Notes on 31 March 1989										
2	Current date		31-Mar-89								
3											
4	Cusip #	Clean price	Maturity	Coupon rate	Accrued interest	Invoice price					
5	912827UU	99.8438	30-Apr-89	7.125	2.972	102.8158		The accrued interest formula in cell E5 is =COUPDAYBS($C$2,C5,2,1)/COUPDAYS($C$2,C5,2,1)/2*D5			
6	912827TP	99.7188	15-May-89	6.875	2.583	102.3016					
7	912827JQ	99.9688	15-May-89	9.250	3.475	103.4439					
8	912827QN	100.2500	15-May-89	11.750	4.414	104.6644					
9	912827UX	99.7813	31-May-89	8.000	2.659	102.4406					
10	912827UZ	99.5313	30-Jun-89	7.375	1.834	101.3648					
11	912827SK	100.0625	30-Jun-89	9.625	2.393	102.4555					
12	912827NK	101.3438	15-Jul-89	14.500	3.004	104.3479					
13	912827VC	99.4063	31-Jul-89	7.625	1.243	100.6490					
14	912827TX	98.9688	15-Aug-89	6.625	0.805	99.7740					
15	912827QW	101.5313	15-Aug-89	13.875	1.686	103.2177					
16	912827VF	99.3125	31-Aug-89	7.750	0.653	99.9654					
17	912827VH	99.4375	30-Sep-89	8.500	-	99.4375					
18	912827SU	99.8125	30-Sep-89	9.375	-	99.8125					
19	912827NS	101.0313	15-Oct-89	11.875	5.448	106.4794					
20	912827VL	98.9688	31-Oct-89	7.875	3.285	102.2536					
21	912827UE	98.0313	15-Nov-89	6.375	2.395	100.4263					

The "clean price" of column B above is the quoted price of the bond. In U.S. markets the actual price paid by the purchaser of the bond, the so-called invoice price, is the sum of the clean price plus the relative portion of the current coupon payment.[4] This latter term, the "accrued interest," is computed by:

$$\begin{matrix}Accrued\\ interest\end{matrix} = \frac{Current\ date - Last\ interest\ date}{Next\ interest\ date - Last\ interest\ date} * Periodic\ interest$$

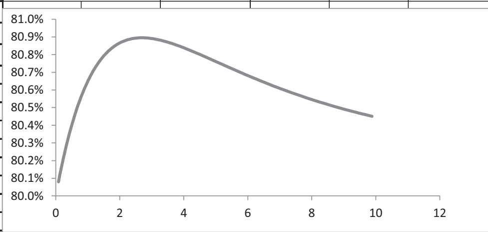
4. In most European bond markets, the quoted bond price is the actual price paid for the bond and there is no separate accrued interest calculation.

The Excel file computes the accrued interest using the formulas **Coupdaybs** and **Coupdays**.

Computing the NS Term Structure

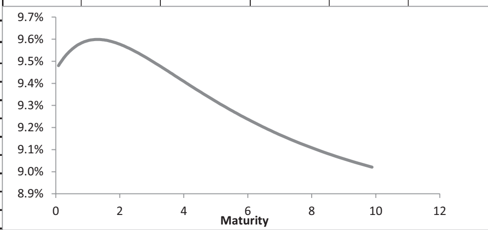
To compute the Nelson-Siegel term structure for these data, we define a VBA function **NSprice** that accepts (α_1, α_2, α_3, β) as input and computes the Nelson-Siegel price of a bond. We now optimize using **Solver** to minimize the sum of the absolute differences between the **NSprice** and the bond's invoice price. Here is the relevant optimization:

	A	B	C	D	E	F	G	H	I	J
1	COMPUTING THE NELSON-SIEGEL TERM STRUCTURE FOR TREASURY NOTES 127 Treasury Notes on 31 March 1989									
2	Settlement date		31-Mar-89				Nelson-Siegel parameters			
3							α_1	0.8000		
4							α_2	0.0000		
5							α_3	0.0300		
6							β	1.5000		
7										
8							Error	8,479.57	<-- {=SUM(ABS(F16:F142-H16:H142))}	
9										
10										
11										
12										
13										
14										
15	Cusip #	Clean price	Maturity	Coupon rate	Accrued interest	Invoice price	Term to maturity	NSprice	NSrate	
16	912827UU	99.8438	30-Apr-89	7.125	2.9720	102.8158	0.0822	96.9656	0.80079	<-- =NSrate($H$3,$H$4,$H$5,$H$6,G16)
17	912827TP	99.7188	15-May-89	6.875	2.5829	102.3016	0.1233	93.7089	0.80117	
18	912827JQ	99.9688	15-May-89	9.250	3.4751	103.4439	0.1233	94.7847	0.80117	
19	912827QN	100.2500	15-May-89	11.750	4.4144	104.6644	0.1233	95.9172	0.80117	
20	912827UX	99.7813	31-May-89	8.000	2.6593	102.4406	0.1671	90.9612	0.80155	



We ask **Solver** to minimize the error term in cell H8. The result is:

	A	B	C	D	E	F	G	H	I	J
1	**COMPUTING THE NELSON-SIEGEL TERM STRUCTURE FOR TREASURY NOTES 127 Treasury Notes on 31 March 1989**									
2	Settlement date		31-Mar-89				Nelson-Siegel parameters			
3							α_1	0.0861		
4							α_2	0.0085		
5							α_3	0.0169		
6							β	1.6258		
7										
8							Error	56.96	<-- {=SUM(ABS(F16:F142-H16:H142))}	
9										
10										
11										
12										
13										
14										
15	Cusip #	Clean price	Maturity	Coupon rate	Accrued interest	Invoice price	Term to maturity	NSprice	NSrate	
16	912827UU	99.8438	30-Apr-89	7.125	2.9720	102.8158	0.0822	102.7587	9.48%	<-- =NSrate($H$3,$H$4,$H$5,$H$6,G16)
17	912827TP	99.7188	15-May-89	6.875	2.5829	102.3016	0.1233	102.2344	9.49%	
18	912827JQ	99.9688	15-May-89	9.250	3.4751	103.4439	0.1233	103.4081	9.49%	
19	912827QN	100.2500	15-May-89	11.750	4.4144	104.6644	0.1233	104.6436	9.49%	
20	912827UX	99.7813	31-May-89	8.000	2.6593	102.4406	0.1671	102.3621	9.50%	



22.7 An Additional Computational Improvement

In this section we present another method of computing the Nelson-Siegel term structure. The method has the advantages of better convergence for short-term securities. In addition the method in the previous section optimizes four nonlinear parameters, whereas the method in this section optimizes three linear parameters and one nonlinear parameter; this minimizes the problem of non-uniqueness of the parameters.

Here are the steps in this method:

- We compute the discount factors using the NS method as illustrated in section 22.6.
- We use these discount factors to discount the bond's payments, but not the final return of face value.
- We compute a rate r_T for each bond's face value so that the model price of the bond (using the combination of the NS discount factors and r_T) equals the bond's invoice price.
- We now optimize Nelson-Siegel to minimize the differences between the r_T and the corresponding Nelson-Siegel rate.

Below we give the definitions:

$$P_1 = \sum_{n=1}^{N_1} C_1 * d\left(t_n^1\right) / Frequency + FaceValue_1 * e^{-r_1 t_{N_1}^1}$$

$$P_2 = \sum_{n=1}^{N_2} C_2 * d\left(t_n^2\right) / Frequency + FaceValue_2 * e^{-r_2 t_{N_2}^2}$$

$$\vdots$$

$$P_K = \sum_{n=1}^{N_K} C_K * d\left(t_n^K\right) / Frequency + FaceValue_K * e^{-r_K t_{N_K}^K}$$

Solving these equations for the rates r_T:

$$r_1 = -ln\left(\frac{\left(P_1 - \sum_{n=1}^{N_1} C_1 d\left(t_n^1\right) / Frequency\right)}{FaceValue_1}\right) \Big/ t_{N_1}^1$$

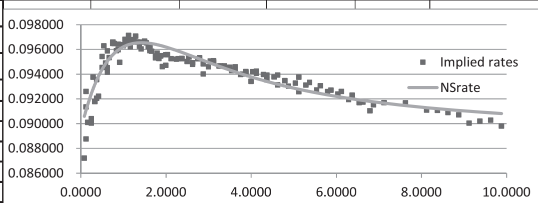
$$r_2 = -ln\left(\frac{\left(P_2 - \sum_{n=1}^{N_2} C_2 d\left(t_n^2\right) / Frequency\right)}{FaceValue_2}\right) \Big/ t_{N_2}^2$$

$$\vdots$$

$$r_K = -ln\left(\frac{\left(P_K - \sum_{n=1}^{N_K} C_K d\left(t_n^K\right) / Frequency\right)}{FaceValue_K}\right) \Big/ t_{N_K}^K$$

The implied rates could be also used for visualization of the term structure. In the following figure we minimize the difference between the implied rates and the corresponding NS rates:

	A	B	C	D	E	F	G	H	I	J	K
1	**COMPUTING THE NELSON-SIEGEL TERM STRUCTURE FOR TREASURY NOTES 127 Treasury Notes on 31 March 1989**										
2	Settlement date		31/Mar/89				Nelson-Siegel parameters				
3							α_1	0.0886			
4							α_2	0.0008			
5							α_3	0.0252			
6							β	0.8269			
7											
8							Error	0.0703	<-- {=SUM(ABS(F15:F141-G15:G141))}		
9											
10											
11											
12											
13						{=NSImpliedRate(B17,$C$2,C17,100,D17,2,$H$3,$H$4,$H$5,$H$6,1)}					
14	Cusip #	Maturity	Coupon rate	Price	Term to maturity	Implied rate	NSrate				
15	912827UU	30/Apr/89	7.125	102.81578	0.0822	0.087203	0.090596	<-- =NSrate($H$3,$H$4,$H$5,$H$6,E15)			
16	912827TP	15/May/89	6.875	102.30162	0.1233	0.088748	0.091106				
17	912827JQ	15/May/89	9.250	103.44389	0.1233	0.091340	0.091106				

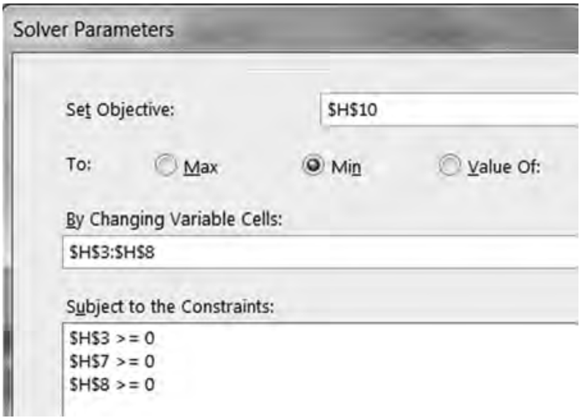


22.8 Nelson-Siegel-Svensson Model

The NS model utilizes four parameters to characterize the term structure. It was suggested by Svensson that the NS model can be significantly improved by introducing two additional parameters.[5] The Nelson-Siegel-Svensson model has the following algebraic form:

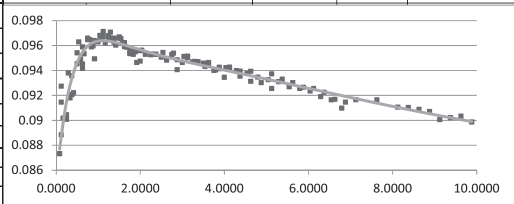
$$y(t) = \alpha_1 + (\alpha_2 + \alpha_3)\left(\beta_1 \frac{1 - e^{-t/\beta_1}}{t}\right) - \alpha_3 e^{-t/\beta_1} + \alpha_4 \left(\beta_2 \frac{1 - e^{-t/\beta_2}}{t} - e^{-t/\beta_2}\right)$$

We use the **Solver** parameters:



5. Lars E. O. Svensson, Estimating and Interpreting Forward Interest Rates: Sweden 1992–1994. IMF Working Paper 94/114, 1994.

Below we show the results:



	A	B	C	D	E	F	G	H	I	J	K
1	**COMPUTING THE NELSON-SIEGEL-SVENSSON TERM STRUCTURE FOR TREASURY NOTES** **127 Treasury Notes on 31 March 1989**										
2	Settlement date		31/Mar/89				Nelson-Siegel parameters				
3							α_1	0.0817			
4							α_2	0.0033			
5							α_3	0.0307			
6							α_4	0.02897			
7							β_1	0.4397			
8							β_2	2.528022			
9											
10							Error	0.0512	<-- {=SUM(ABS(F15:F141-G15:G141))}		
11											
12											
13						{=NSSImpliedRate(B17,$C$2,C17,100,D17,2,$H$3,$H$4,$H$5,$H$6,$H$7,$H$8,1)}					
14	Cusip #	Maturity	Coupon rate	Price	Term to maturity	Implied rate	NSSrate				
15	912827UU	30/Apr/89	7.125	102.81578	0.0822	0.087305812	0.087692799	<-- =NSSrate($H$3,$H$4,$H$5,$H$6,$H$7,$H$8,E15)			
16	912827TP	15/May/89	6.875	102.30162	0.1233	0.088826102	0.088826115				
17	912827JQ	15/May/89	9.250	103.44389	0.1233	0.091444851	0.088826115				

22.9 Summary

In this chapter we presented several simple and powerful mathematical techniques for term structure calculation. The Nelson-Siegel approximation was introduced in detail and its applications were discussed, along with (in section 22.8) the Nelson-Siegel-Svensson variation. We began with the simplest case, in which there is only one bond for each maturity, so that the term structure can be found uniquely. Next, we extended the method to a set in which several bonds can have the same maturity. Then, we discussed a real-life case of Treasury notes, which can have any maturity date. Finally, the Svensson extension of the NS methods was presented.

Appendix: VBA Functions Used in This Chapter

The function **NSrate** computes the Nelson-Siegel rate for time t based on specific values of the four NS parameters α_1, α_2, α_3, β:

```
Function NSrate(alpha1, alpha2, alpha3, beta, t)
   If t = 0 Then
      NSrate = alpha1 + alpha2
   Else
      NSrate = alpha1 + (alpha2 + alpha3) * _
      (beta / t) * (1 - Exp(-t / beta)) - _
      alpha3 * Exp(-t / beta)
   End If
End Function
```

The function **NSdiscount** computes the NS discount factors:

```
Function NSdiscount(alpha1, alpha2, alpha3, _
beta, t)
   If t = 0 Then
      NSdiscount = 1
   Else
      NSdiscount = Exp(-t * (alpha1 + _
      (alpha2 + alpha3) * (beta / t) * _
      (1 - Exp(-t / beta)) - alpha3 * _
      Exp(-t / beta)))
   End If
End Function
```

The function **NSprice** computes the price of a standard coupon bond using the Nelson-Siegel term structure:

```
Function NSprice(alpha1, alpha2, alpha3, _
beta, j, no_payments, rate, frequency)
    temp = 0
    rrate = rate / frequency

    For i = 0 To no_payments - 1
    temp = temp + rrate * 100 * _
    NSdiscount(alpha1, alpha2, alpha3, beta, _
    j + i / frequency)
    Next i
    NSprice = temp + 100 * NSdiscount(alpha1, _
    alpha2, alpha3, beta, j + (no_payments - 1) _
    / frequency)
End Function
```

The function **NSImpliedRate** uses Excel functions to compute the implied rates:

```
Private Function LS(XMatrix, YVector)
Dim X, Y, A, B As Variant
X = XMatrix
Y = YVector
n = UBound(X, 1)
l = UBound(Y)
If n <> l Then GoTo FEnd

With WorksheetFunction
   A = .MMult(.Transpose(X), X)
   LS = .MMult(.MInverse(A), _
   .MMult(.Transpose(X), Y))
End With
FEnd:
End Function
```

23 Calculating Default-Adjusted Expected Bond Returns

23.1 Overview

In this chapter we discuss the effects of default risk on the returns from holding bonds to maturity. The *expected return* on a bond that may possibly default is different from the bond's *promised return*. The latter is defined as the bond's *yield to maturity*, the internal rate of return calculated from the bond's current market price and its *promised* coupon payments and *promised eventual return* of principal in the future. The bond's expected return is less easily calculated: We need to take into account both the bond's probability of future default and the *recovery rate*, the percentage of its principal which holders can expect to recover in the case of default. To complicate matters still further, default can happen in stages, through the gradual degradation of the issuing company's creditworthiness.[1]

In this chapter we use a Markov model to solve for the expected return on a risky bond. Our adjustment procedure takes into account all three of the factors mentioned: the probability of default, the transition of the issuer from one state of creditworthiness to another, and the percentage recovery of face value when the bond defaults. In sections 23.2–23.4 we first use Excel to solve a relatively small-scale problem. We then use some publicly available statistics to program a fuller spreadsheet model. Finally, we show that this model can be used to derive bond betas, the CAPM's risk measure for securities (discussed previously in Chapters 8–11).

Some Preliminaries

Before proceeding, we define a number of terms:

• A bond is issued with a given amount of *principal* or *face value*. When the bond matures, the bondholder is promised the return of this principal. If the bond is issued *at par*, then it is sold for the principal amount.

• A bond bears an interest rate called the *coupon rate*. The periodic payment promised to the bondholders is the product of the coupon rate times the bond's face value.

1. Besides default risk, bonds are also subject to term structure risk: The prices of bonds may show significant variations over time as a result of changing term structure. This statement will be especially true for long-term bonds. In this chapter we abstract from term structure risk, confining ourselves only to a discussion of the effects of default risk on bond expected returns.

- At any given moment, a bond will be sold in the market for a *market price*. This price may differ from the bond's coupon rate.[2]
- The bond's *yield to maturity* (YTM) is the internal rate of return of the bond, assuming that it is held to maturity and that it does not default.

American corporate bonds are rated by various agencies on the basis of the bond issuer's ability to make repayment on the bonds. The classification scheme for two of the major rating agencies, Standard & Poor's (S&P) and Moody's, is given below:

Long-Term Senior Debt Ratings

Investment-Grade Ratings			Speculative-Grade Ratings		
S&P	Moody's	Interpretation	S&P	Moody's	Interpretation
AAA	Aaa	Highest quality	BB+ BB BB–	Ba1 Ba2 Ba3	Likely to fulfill obligations; ongoing uncertainty
AA+ AA AA–	Aa1 Aa2 Aa3	High quality	B+ B B–	B1 B2 B3	High-risk obligations
A+ A A–	A1 A2 A3	Strong payment capacity	CCC+ CCC CCC–	Caa	Current vulnerability to default
BBB+ BBB BBB–	Baa1 Baa2 Baa3	Adequate payment capacity	C D	Ca D	In bankruptcy or default, or other marked shortcomings

2. Just to complicate matters, in the United States the convention is to add to a bond's listed price the *prorated coupon* (the accrued interest) between the time of the last coupon payment and the purchase date. The sum of these two is termed the *invoice price* of the bond; the invoice price is the actual cost at any moment to a purchaser of buying the bond. In our discussion in this chapter we use the term *market price* to denote the invoice price. The computation of the accrued interest is illustrated in section 23.5.

When a bond defaults, its holders will typically receive some payoff, though less than the promised bond coupon rate and return of principal. We refer to the percentage of face value paid off in default as the *recovery percentage*.[3]

23.2 Calculating the Expected Return in a One-Period Framework

The bond's yield to maturity is *not* its expected return: It is clear that both a bond's rating and the anticipated payoff to bond holders in the case of bond default should affect its expected return. All other things being equal, we would expect that if two newly issued bonds have the same term to maturity, then the lower-rated bond (having the higher default probability) should have a higher coupon rate. Similarly, we would expect that an issued and traded bond whose rating has been lowered would experience a decrease in price. We might also expect that the lower the anticipated payoff in the case of default, the lower will be the bond's expected return.

As a simple illustration, we calculate the expected return of a one-year bond which can default at maturity. We use the following symbols:

F = face value of the bond

P = price of bond

Q = annual coupon rate of the bond

π = probability that the bond will *not* default at end of year

λ = fraction of bond's value bondholders collect upon default

The bond's expected end-of-year cash flow is $\pi \cdot (1 + Q) \cdot F + (1 - \pi) \cdot \lambda \cdot F$, and its *expected return* is given by

$$\textit{One-year bond expected return} = \frac{\textit{Expected year-end cash flow}}{\textit{Initial bond price}, P} - 1$$

$$= \frac{\pi \cdot (1+Q) \cdot F + (1-\pi) \cdot \lambda \cdot F}{P} - 1$$

3. The bond's recovery percentage is not, as you might think, the payoff to the bondholders in the final settlement of a bankruptcy. Instead it is usually computed as the price of the bond in the period immediately following a financial distress event.

This calculation is illustrated in the following spreadsheet:

	A	B	C
1	**EXPECTED RETURN ON A ONE-YEAR BOND WITH AN ADJUSTMENT FOR DEFAULT PROBABILITY**		
2	Face value, F	100	
3	Price, P	90	
4	Annual coupon rate, Q	8%	
5	Default probability	20%	
6	Recovery percentage	40%	
7			
8	Expected period 1 cash flow	94.4	<-- =B2*(1+B4)*(1-B5)+B2*B6*B5
9	Expected return	4.89%	<-- =B8/B3-1

23.3 Calculating the Bond Expected Return in a Multi-Period Framework

We now introduce multiple periods into the above problem. In this section we define a basic Markov model which uses a ratings transition matrix to compute a bond's expected return. The model is illustrated using a very simple set of ratings, much simpler than the complex rating system illustrated in section 23.1. Section 23.5 uses more realistic data.

We suppose that at any date there are four possible bond "ratings":

A, B, C — Bond ratings of solvent bonds in decreasing order of credit worthiness.

D — The bond is in default for the first time and pays off recovery rate λ of the face value.

E — The bond was in default in the previous period; it therefore pays off 0 in the current period and in any future periods.

The *transition probability* matrix Π is given by

$$\Pi = \begin{bmatrix} \pi_{AA} & \pi_{AB} & \pi_{AC} & \pi_{AD} & 0 \\ \pi_{BA} & \pi_{BB} & \pi_{BC} & \pi_{BD} & 0 \\ \pi_{CA} & \pi_{CB} & \pi_{CC} & \pi_{CD} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The probabilities in each row of the matrix Π indicate the probability that in *one period* the bond will go from a rating of i to a rating of j. In the numerical examples in this and the following two sections, we use the following Π:

	A	B	C	D	E	F
2		**A**	**B**	**C**	**D**	**E**
3	**A**	0.9700	0.0200	0.0100	0.0000	0.0000
4	**B**	0.0500	0.8000	0.1500	0.0000	0.0000
5	**C**	0.0100	0.0200	0.7500	0.2200	0.0000
6	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
7	**E**	0.0000	0.0000	0.0000	0.0000	1.0000

What does this matrix Π mean?

• If a bond is rated A in the current period, there is a probability of 0.97 that it will still be rated A in the next period. There is a probability 0.02 that it will be rated B in the next period and a probability of 0.01 that it will be rated C. It is impossible for the bond to be rated A today and D or E in the subsequent period.

• A bond that starts off with a rating of B can—in a subsequent period—be rated A (with a probability of 0.05), be rated B (with a probability of 0.8), or be rated C (probability 0.15). Bonds rated B in the current period do not default (rating D) in the next period. The transition probabilities from state C to states A, B, C, and D are 0.01, 0.02, 0.75, and 0.22, respectively.

• While it is possible to go from ratings A, B, or C to any of ratings A, B, C, or D, it is *not* possible to go from A, B, or C to E. This is so, since E denotes that default took place in the *previous period*.

• A bond that is currently in state D (i.e., first-time default) will necessarily be in E in the next period. Thus the fourth row of our matrix Π will always be [0 0 0 0 1].

• Once the rating is in E, it remains there permanently. This means that the fifth row of the matrix Π will also always be [0 0 0 0 1].

The Multi-Period Transition Matrix

The matrix Π defines the transition probabilities over one period. The two-period transition probabilities are given by the matrix product $\Pi*\Pi$. The spreadsheet below uses the array function **MMULT**.[4] It shows that the product $\Pi*\Pi$ is:

$$\textit{Two-period transition probability} = \Pi*\Pi$$
$$= \begin{bmatrix} 0.9420 & 0.0356 & 0.0202 & 0.0022 & 0.0000 \\ 0.0900 & 0.6440 & 0.2330 & 0.0330 & 0.0000 \\ 0.0182 & 0.0312 & 0.5656 & 0.1650 & 0.2200 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

Thus if a bond is rated "B" today, there is a probability of 9% that in two periods it will be rated "A," a probability of 64.4% that in two periods it will be rated B, a probability of 23.3% that in two periods it will be rated C, and a probability of 3.3% that in two periods it will default (and hence be rated D).

Here is the spreadsheet:

4. See discussion of matrix products and array functions in Chapter 34.

	A	B	C	D	E	F
1	**USING THE MMULT FUNCTION To compute multi-period transition matrices**					
2	**One-period transition matrix**					
3		**A**	**B**	**C**	**D**	**E**
4	**A**	0.9700	0.0200	0.0100	0.0000	0.0000
5	**B**	0.0500	0.8000	0.1500	0.0000	0.0000
6	**C**	0.0100	0.0200	0.7500	0.2200	0.0000
7	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
8	**E**	0.0000	0.0000	0.0000	0.0000	1.0000
9						
10	**Two-period transition matrix**					
11		**A**	**B**	**C**	**D**	**E**
12	**A**	0.9420	0.0356	0.0202	0.0022	0.0000
13	**B**	0.0900	0.6440	0.2330	0.0330	0.0000
14	**C**	0.0182	0.0312	0.5656	0.1650	0.2200
15	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
16	**E**	0.0000	0.0000	0.0000	0.0000	1.0000
17	**Cells B12:F16 contain the array formula =MMULT(B4:F8,B4:F8)**					
18						
19	**Three-period transition matrix**					
20		**A**	**B**	**C**	**D**	**E**
21	**A**	0.9157	0.0477	0.0299	0.0044	0.0022
22	**B**	0.1218	0.5217	0.2723	0.0513	0.0330
23	**C**	0.0249	0.0366	0.4291	0.1244	0.3850
24	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
25	**E**	0.0000	0.0000	0.0000	0.0000	1.0000
26	**Cells B21:F25 contain the array formula =MMULT(B4:F8,B12:F16)**					

In general, the year t transition matrix is given by the matrix power Π^t. Calculating these matrix powers by the procedure illustrated above is cumbersome, so we define a VBA function **Matrixpower** to compute powers of matrices:

```
Function Matrixpower(matrix, n)
  If n = 1 Then
     Matrixpower = matrix
     Else: Matrixpower = Application.MMult
      (Matrixpower(matrix, n - 1), matrix)
  End If
End Function
```

The use of this function is illustrated below. The function **Matrixpower** allows a one-step computation of the power of any transition matrix:

	A	B	C	D	E	F
1	**USING THE FUNCTION MATRIXPOWER To compute multi-period transition matrices**					
2	**One-period transition matrix**					
3		**A**	**B**	**C**	**D**	**E**
4	**A**	0.9700	0.0200	0.0100	0.0000	0.0000
5	**B**	0.0500	0.8000	0.1500	0.0000	0.0000
6	**C**	0.0100	0.0200	0.7500	0.2200	0.0000
7	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
8	**E**	0.0000	0.0000	0.0000	0.0000	1.0000
9						
10	**t**	**10**				
11						
12	**t-period transition matrix**					
13		**A**	**B**	**C**	**D**	**E**
14	**A**	0.7648	0.0799	0.0699	0.0148	0.0706
15	**B**	0.2123	0.1429	0.1747	0.0432	0.4269
16	**C**	0.0450	0.0250	0.0755	0.0208	0.8338
17	**D**	0.0000	0.0000	0.0000	0.0000	1.0000
18	**E**	0.0000	0.0000	0.0000	0.0000	1.0000
19	**Cells B14:F18 contain the array formula =matrixpower(B4:F8,B10)**					

From the above example it follows that if a bond started out with an A rating, there is a probability of 1.48% that the bond will be in default at the end of 10 periods and a 7.06% probability that it will have defaulted in a previous period (rating E).

The Bond Payoff Vector

Recall that Q denotes the bond's coupon rate and λ denotes the recovery percentage payoff of face value if the bond defaults. The payoff vector of the bond depends on whether the bond is currently in its last period N or whether $t < N$:

$$Payoff\,(t, t < N) = \begin{cases} Q \\ Q \\ Q \\ \lambda \\ 0 \end{cases} \quad Payoff\,(t, t = N) = \begin{cases} 1+Q \\ 1+Q \\ 1+Q \\ \lambda \\ 0 \end{cases}$$

The first three elements of each vector denote the payoff in non-defaulted states, the fourth element λ is the payoff if the rating is D, and the fifth element 0 is the payoff if the bond rating is E. (Recall that E is the rating for the period *after* the bond defaults—in our model the payoff in rating E is always zero.) The distinction between the two vectors depends, of course, on the repayment of principal in the terminal period.

Before we can define the expected payoffs, we need to define one further vector, which will denote the *initial state of the bond*. This current state vector is a vector with a 1 for the current rating of the bond and zeros elsewhere. Thus, for example, if the bond has rating A at date 0, then *initial* = [1 0 0 0 0]; if it has date 0 rating of B, then *initial* = [0 1 0 0 0].

We can now define the expected bond payoff in period *t*:

$$E[Payoff(t)] = Initial \cdot \Pi^t \cdot Payoff(t)$$

23.4 A Numerical Example

We continue using the numerical Π from the previous section to price a bond with the following characteristics:

- The bond is currently rated B.
- Its coupon rate $Q = 7\%$.
- The bond has 5 more years to maturity.
- The bond's current market price is 100% of its face value.
- The bond's recovery percentage $\lambda = 50\%$.

The following spreadsheet shows the facts listed above as well as the payoff vectors of the bond at dates before maturity (in cells F3:F7) and on the maturity date (cells I3:I7). The transition matrix is given in cells C10:G14, and the initial vector is given in C16:G16.

The expected bond payoffs are given in row 20. Before we explain how they were calculated, we note the important economic fact that—if the expected payoffs are as given—then the *bond's expected return* is calculated by the Excel **IRR** function. As cell B21 shows, this expected return is 4.61%. The actual formula in cell B21 is **IRR(B20:AN20)**. This allows the calculation of the IRR of bonds of maturity up to 40 years.

	A	B	C	D	E	F	G	H	I
1	**CALCULATING THE EXPECTED BOND RETURN**								
2	Bond price	100.00%				**Payoff (t<N)**			**Payoff (N)**
3	Coupon rate, Q	7%			Cells to right	7%		Cells to right	107%
4	Recovery rate, λ	50%			are called	7%		are called	107%
5	Bond term, N	5			"payoff1"	7%		"payoff2"	107%
6	Initial rating	B			in row 20	50%		in row 20	50%
7						0%			0%
8									
9			**A**	**B**	**C**	**D**	**E**		
10	**Transition matrix →**	**A**	0.9700	0.0200	0.0100	0.0000	0.0000		
11		**B**	0.0500	0.8000	0.1500	0.0000	0.0000		
12		**C**	0.0100	0.0200	0.7500	0.2200	0.0000		
13		**D**	0.0000	0.0000	0.0000	0.0000	1.0000		
14		**E**	0.0000	0.0000	0.0000	0.0000	1.0000		
15									
16	**Initial vector**		0	1	0	0	0		
17	Formula in cell C16: =IF(UPPER(B6)="A",1,0)								
18									
19	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
20	Expected payoffs	-1.0000	0.0700	0.0842	0.0897	0.0899	0.8802	0.0000	0.0000
21	Expected yield	4.61%	<-- =IRR(B20:AN20,0)						
22									
23									
24	IRR of expected payoffs		=IF(year>bondterm,0, IF(year=bondterm,MMULT(initial,MMULT(matrixpower(transition,year),payoff2)), MMULT(initial,MMULT(matrixpower(transition,year),payoff1))))						
25									
26									

Note the use of the **IF** statement in translating the bond's initial rating (cell B6) to the initial vector given in row 16. To avoid confusion, we write this as **IF(Upper(B6)="A",1,0)**, etc. This guarantees that even if the bond's rating is entered as a lowercase letter, the initial vector will come out correctly.

How to Calculate the Expected Bond Payoffs

As indicated in the previous section, the period t expected bond payoff is given by the following formula: $E[payoff(t)] = initial \cdot \Pi^t \cdot payoff(t)$. The formula in row 21 uses two **IF** statements to implement this formula:

= IF(year > bondterm, 0,
IF(year = bondterm, MMULT(initial, MMULT(matrixpower
(transition, year), payoff2)),
MMULT(initial, MMULT
(matrixpower(transition, year), payoff1))))

Here's what this means:

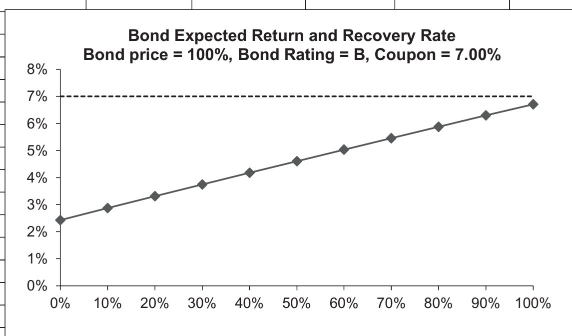
- First **IF**: If the current year is greater than the bond term *N* (in our example *N* = 5), then the payoff on the bond is 0.
- Second **IF**: If the current year is equal to the bond term *N*, then the expected payoff on the bond is **MMULT(initial,MMULT(matrixpower(transition, year),payoff2)).** Here **transition** is the name for the transition matrix in cells C10:G14 and **payoff2** is the name for the cells I3:I7.
- If the current year *n* is less than the bond term, then the expected payoff on the bond is **MMULT(initial,MMULT(matrixpower(transition,C18), payoff1)**, where **payoff1** is the name for the cells F3:F7.

Copying this formula gives the whole vector of expected bond payoffs.

23.5 Experimenting with the Example

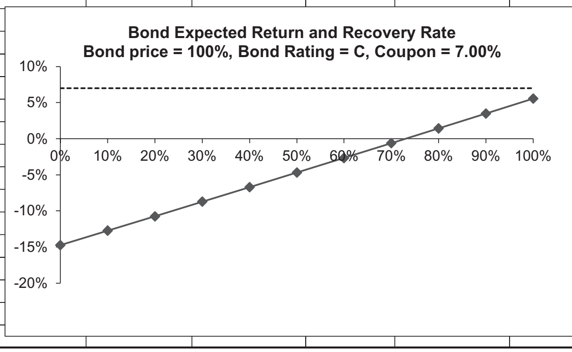
We can gain some insight into the relation between a bond's expected return, its coupon rate, and its yield to maturity (YTM) by constructing some data tables. In the data table below we compute the bond's expected return as a function of its recovery percentage λ:

	A	B	C
28	**Data table: Recovery percentage and expected yield**		
29	**Recovery percentage, λ**	4.61%	<-- =B21 , Tableheader
30	0%	2.43%	7%
31	10%	2.87%	7%
32	20%	3.31%	7%
33	30%	3.74%	7%
34	40%	4.18%	7%
35	50%	4.61%	7%
36	60%	5.03%	7%
37	70%	5.46%	7%
38	80%	5.88%	7%
39	90%	6.30%	7%
40	100%	6.71%	7%
41			
42	Note: The data table has a series with the		
43	coupon rate appended so that in the graph we		
44	can see the convergence of the bond expected		
45	return to the coupon rate (cells C30:C40)		



We conclude that a bond selling at par will, for every recovery percentage λ, have expected return less than the coupon rate. If the bond's initial rating is lower, then the expected return is less for every λ:

	A	B	C
28	**Data table: Recovery percentage and expected yield**		
29	**Recovery percentage, λ**	-4.69%	<-- =B21 , Tableheader
30	0%	-14.72%	7%
31	10%	-12.73%	7%
32	20%	-10.74%	7%
33	30%	-8.73%	7%
34	40%	-6.71%	7%
35	50%	-4.69%	7%
36	60%	-2.66%	7%
37	70%	-0.62%	7%
38	80%	1.43%	7%
39	90%	3.49%	7%
40	100%	5.55%	7%
41			
42	Note: The data table has a series with the		
43	coupon rate appended so that in the graph we		
44	can see the convergence of the bond expected		
45	return to the coupon rate (cells C30:C40)		

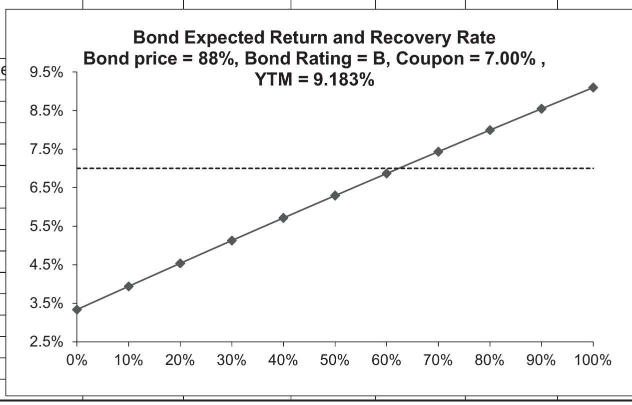


As shown in the example below, when the bond price is below par (meaning that the bond is sold at less than 100% of its face value), the bond's expected return can be both below and above its coupon rate.

	A	B	C	D	E	F	G	H	I	J
1	**CALCULATING THE EXPECTED BOND RETURN**									
2	Bond price	88.00%				**Payoff (t<N)**			**Payoff (N)**	
3	Coupon rate, Q	7%			Cells F6:F6	7%		Cells I3:I6	107%	
4	Recovery rate, λ	50%			are called	7%		are called	107%	
5	Bond term, N	8			"payoff1"	7%		"payoff2"	107%	
6	Initial rating	B			in row 19	50%		in row 19	50%	
7						0%			0%	
8										
9			**A**	**B**	**C**	**D**	**E**			
10	**Transition matrix →**	**A**	0.9700	0.0200	0.0100	0.0000	0.0000			
11		**B**	0.0500	0.8000	0.1500	0.0000	0.0000			
12		**C**	0.0100	0.0200	0.7500	0.2200	0.0000			
13		**D**	0.0000	0.0000	0.0000	0.0000	1.0000			
14		**E**	0.0000	0.0000	0.0000	0.0000	1.0000			
15										
16	**Initial vector**		0	1	0	0	0.0000			
17	Formula in cell C16: =IF(UPPER(B6)="A",1,0)									
18										
19	**Year**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
20	Expected payoffs	-0.8800	0.0700	0.0842	0.0897	0.0899	0.0867	0.0818	0.0761	0.6914
21	Expected yield	6.30%								
22										
23										
24										
25										
26										
27										
28	**Data table: Recovery percentage and expected yield**									
29	**Recovery percentage, λ**	6.30%	<-- Table he							
30	0%	3.33%	7%							
31	10%	3.94%	7%							
32	20%	4.54%	7%							
33	30%	5.13%	7%							
34	40%	5.72%	7%							
35	50%	6.30%	7%							
36	60%	6.87%	7%							
37	70%	7.43%	7%							
38	80%	8.00%	7%							
39	90%	8.55%	7%							
40	100%	9.10%	7%							
41										
42	Note: The data table has a series with the									
43	coupon rate appended so that in the graph we									
44	can see the convergence of the bond expected									

IRR of expected payoffs

=IF(year>bondterm,0,
IF(year=bondterm,MMULT(initial,MMULT(matrixpower(transition,year),payoff2)),
MMULT(initial,MMULT(matrixpower(transition,year),payoff1))))



23.6 Computing the Bond Expected Return for an Actual Bond

In this section we illustrate the computation of a bond expected return for an actual bond. Although the principles used are the same as those discussed above, we introduce three innovations:

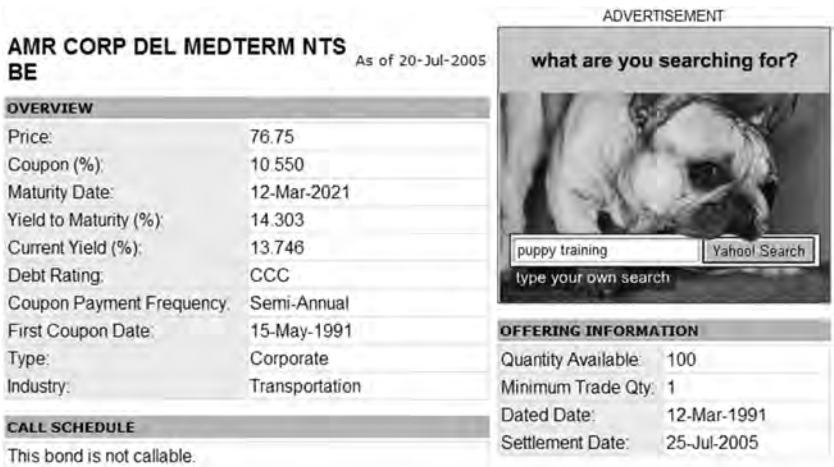
- We compute the bond's actual price using its quoted price and the accrued interest. The *accrued interest* is jargon for the unpaid part of the bond coupon since the last interest payment. In U.S. bond markets, the accrued interest is added to the quoted bond price to compute the amount actually paid for the bond. In most European bond markets, the quoted bond price is the actual price paid for the bond and there is no separate accrued interest calculation. The accrued interest is defined as:

$$Accrued\ interest = \frac{Current\ date - Last\ interest\ date}{Next\ interest\ date - Last\ interest\ date} * Periodic\ interest$$

- We use actual payment dates for the bond and use the **XIRR** function to compute the bond's expected yield.
- We use an actual transition matrix for bond ratings.

The bond we analyze is a CCC-rated bond issued by AMR (the parent company of American Airlines). Originally issued on 15 May 1991 and maturing on 12 March 2021, the AMR bond has a coupon of 10.55% payable semiannually on 15 May and 15 November.

When we looked up the bond on Yahoo on 20 July 2005, its price was 76.75% of par:



AMR CORP DEL MEDTERM NTS BE As of 20-Jul-2005

OVERVIEW	
Price:	76.75
Coupon (%):	10.550
Maturity Date:	12-Mar-2021
Yield to Maturity (%):	14.303
Current Yield (%):	13.746
Debt Rating:	CCC
Coupon Payment Frequency:	Semi-Annual
First Coupon Date:	15-May-1991
Type:	Corporate
Industry:	Transportation

CALL SCHEDULE
This bond is not callable.

OFFERING INFORMATION	
Quantity Available:	100
Minimum Trade Qty:	1
Dated Date:	12-Mar-1991
Settlement Date:	25-Jul-2005

To the quoted price of 76.75% we must add the bond's accrued interest:

	M	N
3	**Accrued interest calculation**	
4	Last payment date	15-Mar-05
5	Next payment date	15-Sep-05
6	Current date	20-Jul-05
7	Percentage of period	0.69
8	Accrued interest	0.0364

It follows that the actual price paid for the bond is 76.75% + 3.64% = 80.39%.

In the spreadsheet below we compute the AMR expected return:

	A	B	C	D	E	F	G	H	I	J	K
1	**CALCULATING THE EXPECTED BOND RETURN** **This version computes the expected bond return for AMR taking into account actual dates** **Uses annual transition matrix**										
2	Bond price	76.75%					**Payoff (t<N)**			**Payoff (N)**	
3	Coupon rate, Q	10.55%			Vector to	AAA	5.28%		Vector to	105.28%	
4	Actual price (includes accrued)	80.39%	<-- =B2+N8		right	AA	5.28%		right	105.28%	
5	Recovery rate, λ	50.00%			called	A	5.28%		called	105.28%	
6	Maturity date	12-Mar-21			"payoff1"	BBB	5.28%		"payoff2"	105.28%	
7	Current date	20-Jul-05				BB	5.28%			105.28%	
8	Initial rating	CCC	Coupon paid semiannually			B	5.28%			105.28%	
9	Bond YTM	14.81%				CCC	5.28%			105.28%	
10						Default	50.00%			50.00%	
11						E	0%			0%	
12											
13		**original**	**probability of migrating to rating by year end (%)**								
14	**Transition matrix →**	**rating**	AAA	AA	A	BBB	BB	B	CCC	Default	E
15		AAA	0.9366	0.0583	0.0040	0.0008	0.0003	0.0000	0.0000	0.0000	0.0000
16		AA	0.0066	0.9172	0.0694	0.0049	0.0006	0.0009	0.0002	0.0001	0.0000
17		A	0.0007	0.0225	0.9176	0.0519	0.0049	0.0020	0.0001	0.0004	0.0000
18		BBB	0.0003	0.0025	0.0483	0.8926	0.0444	0.0081	0.0016	0.0022	0.0000
19		BB	0.0003	0.0007	0.0044	0.0667	0.8331	0.0747	0.0105	0.0098	0.0000
20		B	0.0000	0.0010	0.0033	0.0046	0.0577	0.8419	0.0387	0.0530	0.0000
21		CCC	0.0016	0.0000	0.0031	0.0093	0.0200	0.1074	0.6396	0.2194	0.0000
22		Default	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
23		E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
24											
25											
26		AAA	AA	A	BBB	BB	B	CCC	Default	E	
27	Initial vector	0	0	0	0	0	0	1	0	0	
28											
29	**Computing the AMR bond expected return**										
30		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
31	**Date**	**20-Jul-05**	**15-Sep-05**	**15-Mar-06**	**15-Sep-06**	**15-Mar-07**	**15-Sep-07**	**15-Mar-08**	**15-Sep-08**	**15-Mar-09**	**15-Sep-09**
32	Expected payoffs	-0.8039	0.1509	0.1066	0.0781	0.0594	0.0471	0.0387	0.0329	0.0287	0.0257
33	Expected bond return	2.49%	<-- =XIRR(B32:AP32,B31:AP31)								
34											
35											
36											
37											
38											

Annualized IRR of expected payoffs computed using actual dates with **XIRR**.

=IF(E31>$B$6,0,
IF(E31=$B$6,MMULT(initial,MMULT(matrixpower(transition,E30),payoff2)),
MMULT(initial,MMULT(matrixpower(transition,E30),payoff1))))

Assuming a recovery rate of 50%, the bond has an expected return of 2.49%.

What Are the Recovery Rates?

The recovery rate is obviously a critical factor in computing the bond's expected return. There are considerable data on the recovery rates in bankruptcy from various industries. A table from an article by Edward Altman and Velore M. Kishore is given below; from this table we can see that the average recovery rate from a variety of industries was 41%.

Recovery Rates by Industry: Defaulted Bonds by Three-Digit SIC Code, 1971–1995

			Recovery Rate			
Industry	SIC Code	Number of Observations	Average	Weighted Observations	Median Average	Standard Deviation Weighted
Public utilities	490	56	70.47	65.48	79.07	19.46
Chemicals, petroleum, rubber and plastic products	280,290,300	35	62.73	80.39	71.88	27.10
Machinery, instruments, and related products	350,360,380	36	48.74	44.75	47.50	20.13
Services—business and personal	470,632,720,730	14	46.23	50.01	41.50	25.03
Food and kindred products	200	18	45.28	37.40	41.50	21.67
Wholesale and retail trade	500,510,520	12	44.00	48.90	37.32	22.14
Diversified manufacturing	390,998	20	42.29	29.49	33.88	24.98
Casino, hotel, and recreation	770,790	21	40.15	39.74	28.00	25.66
Building materials, metals, and fabricated products	320,330,340	68	38.76	29.64	37.75	22.86
Transportation and transportation equipment	370,410,420,450	52	38.42	41.12	37.13	27.98
Communication, broadcasting, movies, printing, publishing	270,480,780	65	37.08	39.34	34.50	20.79
Financial institutions	600,610,620,630, 670	66	35.69	35.44	32.15	25.72
Construction and real estate	150,650	35	35.27	28.58	24.00	28.69
General merchandise stores	530,540,560,570, 580,000	89	33.16	29.35	30.00	20.47
Mining and petroleum drilling	100,103	45	33.02	31.83	32.00	18.01
Textile and apparel products	220,230	31	31.66	33.72	31.13	15.24
Wood, paper, and leather products	240,250,260,310	11	29.77	24.30	18.25	24.38
Lodging, hospitals, and nursing facilities	700 through 890	22	26.49	19.61	16.00	22.65
Total		**696**	**41.00**	**39.11**	**36.25**	**25.56**

Source: Altman & Kishore, "Almost Everything You Wanted to Know about Recoveries on Defaulted Bonds," Table 3, *Financial Analysis Journal*, November/December 1996.

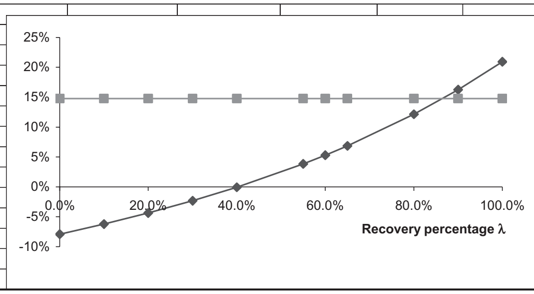
Using the Altman and Kishore numbers, the average recovery percentage for transportation companies is 38.42%, with a standard deviation of 27.98%. Taking one standard deviation on either side of the average, we can conclude that the recovery percentage for a transportation company is somewhere between (38.42% – 27.98%, 38.42% + 27.98%) = (~10%, ~66%).

In the spreadsheet below we have "backward engineered" a plausible set of recovery ratios, from 55% to 65%, for the AMR bond. These "guesstimates" for the AMR recovery is based on two assumptions:

- The AMR bond should not have an expected return significantly more than the riskless rate of return, which at the time of our calculations was around 4%.
- The AMR bond expected return should be significantly less than its YTM of ~15%. The YTM is based on promised payments, and we find it implausible that these should correspond to the expected returns.

This gives (highlighted area below) an expected bond yield of between 3.85% and 6.86% for the AMR bond:

	A	B	C
40	**Data table: Recovery percentage and expected AMR bond return**		
41	Recovery percentage, λ		YTM
42	0.0%	-7.93%	14.78%
43	10.0%	-6.22%	14.78%
44	20.0%	-4.36%	14.78%
45	30.0%	-2.32%	14.78%
46	40.0%	-0.05%	14.78%
47	55.0%	3.85%	14.78%
48	60.0%	5.31%	14.78%
49	65.0%	6.86%	14.78%
50	80.0%	12.15%	14.78%
51	90.0%	16.27%	14.78%
52	100.0%	20.93%	14.78%
53			
54	Cell B41 contains the data table header =B33		



23.7 Semiannual Transition Matrices

The analysis of the AMR bond in the previous section assumes that the annual transition probabilities are also valid for bonds paying semiannual coupons. We could refine this assumption by computing a semiannual transition matrix from the S&P data. Such a matrix would be the square root of the Π matrix. This is not a calculation which can be easily done in Excel. In the spreadsheet below we have used a computation from *Mathematica* to find the semiannual transition matrix.[5]

5. *Mathematica* is a high-powered computational program. See www.wolfram.com.

COMPUTING THE SQUARE ROOT OF THE TRANSITION MATRIX

The one-year transition matrix

original rating	probability of migrating to rating by year end (%) AAA	AA	A	BBB	BB	B	CCC	Default	E
AAA	0.9366	0.0583	0.0040	0.0008	0.0003	0.0000	0.0000	0.0000	0.0000
AA	0.0066	0.9172	0.0694	0.0049	0.0006	0.0009	0.0002	0.0001	0.0000
A	0.0007	0.0225	0.9176	0.0519	0.0049	0.0020	0.0001	0.0004	0.0000
BBB	0.0003	0.0025	0.0483	0.8926	0.0444	0.0081	0.0016	0.0022	0.0000
BB	0.0003	0.0007	0.0044	0.0667	0.8331	0.0747	0.0105	0.0098	0.0000
B	0.0000	0.0010	0.0033	0.0046	0.0577	0.8419	0.0387	0.0530	0.0000
CCC	0.0016	0.0000	0.0031	0.0093	0.0200	0.1074	0.6396	0.2194	0.0000
Default	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

The square root of the one-year transition matrix as computed by *Mathematica*
Note the negative entries

original rating	probability of migrating to rating by year end (%) AAA	AA	A	BBB	BB	B	CCC	Default	E
AAA	0.9677	0.0303	0.0015	0.0004	0.0001	0.0000	0.0000	0.0000	0.0000
AA	0.0034	0.9574	0.0362	0.0021	0.0002	0.0004	0.0001	0.0000	0.0000
A	0.0003	0.0117	0.9573	0.0272	0.0023	0.0010	0.0000	0.0003	-0.0001
BBB	0.0001	0.0012	0.0254	0.9439	0.0238	0.0038	0.0008	0.0017	-0.0007
BB	0.0002	0.0003	0.0018	0.0359	0.9115	0.0406	0.0056	0.0068	-0.0026
B	0.0000	0.0005	0.0017	0.0018	0.0314	0.9161	0.0225	0.0510	-0.0248
CCC	0.0009	-0.0001	0.0016	0.0050	0.0105	0.0624	0.7988	0.2706	-0.1495
Default	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

The semi-annual transition matrix. To eliminate the negative entries in the *Mathematica* matrix above:
a. We assume that a transition from AAA, ... , CCC --> E is impossible (i.e.: last column is zero except for last two entries).
b. We set all other negative entries to zero.
c. We set default probability so that each row sums to 1.

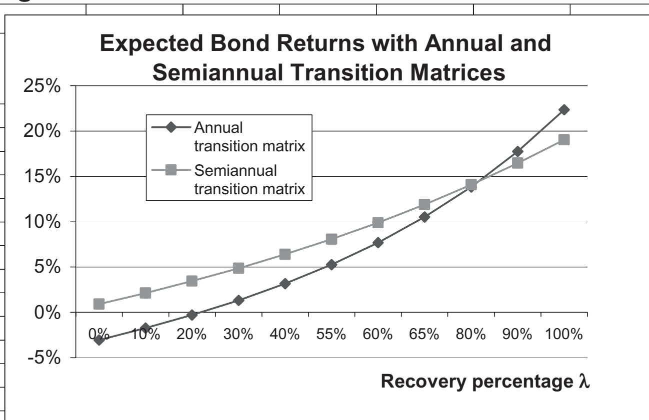
original rating	probability of migrating to rating by year end (%) AAA	AA	A	BBB	BB	B	CCC	Default	E
AAA	0.9677	0.0303	0.0015	0.0004	0.0001	0.0000	0.0000	0.0000	0.0000
AA	0.0034	0.9574	0.0362	0.0021	0.0002	0.0004	0.0001	0.0001	0.0000
A	0.0003	0.0117	0.9573	0.0272	0.0023	0.0010	0.0000	0.0001	0.0000
BBB	0.0001	0.0012	0.0254	0.9439	0.0238	0.0038	0.0008	0.0010	0.0000
BB	0.0002	0.0003	0.0018	0.0359	0.9115	0.0406	0.0056	0.0041	0.0000
B	0.0000	0.0005	0.0017	0.0018	0.0314	0.9161	0.0225	0.0261	0.0000
CCC	0.0009	0.0000	0.0016	0.0050	0.0105	0.0624	0.7988	0.1208	0.0000
Default	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

If we use the semiannual transition matrix to compute the expected bond returns we get:

	A	B	C	D	E	F	G	H	I	J	K
1	**CALCULATING THE EXPECTED BOND RETURN** **This version computes the expected bond return for AMR taking into account actual dates** **Uses semi-annual transition matrix**										
2	Bond price	76.75%					**Payoff (t<N)**			**Payoff (N)**	
3	Coupon rate, Q	10.55%			Vector to	AAA	5.28%		Vector to	105.28%	
4	Actual price (includes accrued)	80.39%	<-- =B2+N8		right	AA	5.28%		right	105.28%	
5	Recovery rate, λ	50.00%			called	A	5.28%		called	105.28%	
6	Maturity date	12-Mar-21			"payoff1"	BBB	5.28%		"payoff2"	105.28%	
7	Current date	20-Jul-05				BB	5.28%			105.28%	
8	Maturity (years)	15.65				B	5.28%			105.28%	
9	Number of semiannual payments	30.00	Coupon paid			CCC	5.28%			105.28%	
10	Initial rating	CCC	semiannually			Default	50.00%			50.00%	
11	Bond YTM	14.81%				E	0%			0%	
12											
13		**original**	**probability of migrating to rating by year end (%)**								
14	**Transition matrix →**	**rating**	AAA	AA	A	BBB	BB	B	CCC	Default	E
15		AAA	0.9677	0.0303	0.0015	0.0004	0.0001	0.0000	0.0000	0.0000	0.0000
16		AA	0.0034	0.9574	0.0362	0.0021	0.0002	0.0004	0.0001	0.0001	0.0000
17		A	0.0003	0.0117	0.9573	0.0272	0.0023	0.0010	0.0000	0.0001	0.0000
18		BBB	0.0001	0.0012	0.0254	0.9439	0.0238	0.0038	0.0008	0.0010	0.0000
19		BB	0.0002	0.0003	0.0018	0.0359	0.9115	0.0406	0.0056	0.0041	0.0000
20		B	0.0000	0.0005	0.0017	0.0018	0.0314	0.9161	0.0225	0.0261	0.0000
21		CCC	0.0009	-0.0001	0.0016	0.0050	0.0105	0.0624	0.7988	0.1208	0.0000
22		Default	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
23		E	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
24											
25											
26		AAA	AA	A	BBB	BB	B	CCC	Default	E	
27	Initial vector	0	0	0	0	0	0	1	0	0	
28											
29	**Period**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
30	**Date**	**20-Jul-05**	**15-Sep-05**	**15-Mar-06**	**15-Sep-06**	**15-Mar-07**	**15-Sep-07**	**15-Mar-08**	**15-Sep-08**	**15-Mar-09**	**15-Sep-09**
31	Expected payoffs	-0.8039	0.1068	0.0903	0.0771	0.0664	0.0578	0.0508	0.0451	0.0405	0.0367
32	Expected yield	8.07%	<-- =XIRR(B31:AP31,B30:AP30)								

The semiannual transition matrix gives expected bond returns which are, in general, higher than those given by the annual transition matrix:

COMPARING THE EXPECTED RETURNS WITH SEMIANNUAL VS. ANNUAL TRANSITION MATRICES Using actual dates and XIRR		
	Expected bond return	
Recovery percentage, λ	Annual transition matrix	Semiannual transition matrix
0%	-3.05%	0.93%
10%	-1.75%	2.14%
20%	-0.30%	3.45%
30%	1.32%	4.86%
40%	3.15%	6.40%
55%	5.25%	8.07%
60%	7.68%	9.90%
65%	10.51%	11.89%
80%	13.83%	14.07%
90%	17.73%	16.45%
100%	22.33%	19.04%



23.8 Computing Bond Beta

A vexatious problem in corporate finance is the computation of bond betas. The model presented in this chapter can be easily used to compute the beta of a bond. Recall from Chapter 2 that the capital asset pricing model's *security market line (SML)* is given by:

$$E(r_d) = r_f + \beta_d \left[E(r_m) - r_f \right]$$

where

$E(r_d)$ = *Expected return on debt*

r_f = *Return on riskless debt*

$E(r_m)$ = *Return on equity market portfolio*

If we know the expected return on debt, we can calculate the debt β, provided we know the risk-free rate r_f and the expected rate of return on the market $E(r_m)$. Suppose, for example, that the market risk premium $E(r_m) - r_f = 8.4\%$, and that $r_f = 7\%$. Then a bond having an expected return of 8% will have a β of 0.119:

	A	B	C
1	**CALCULATING A BOND'S BETA**		
2	Market risk premium, $E(r_m) - r_f$	8.40%	
3	r_f	7%	
4	Expected bond return	8.00%	
5	Implied bond beta	0.119	<-- =(B4-B3)/B2

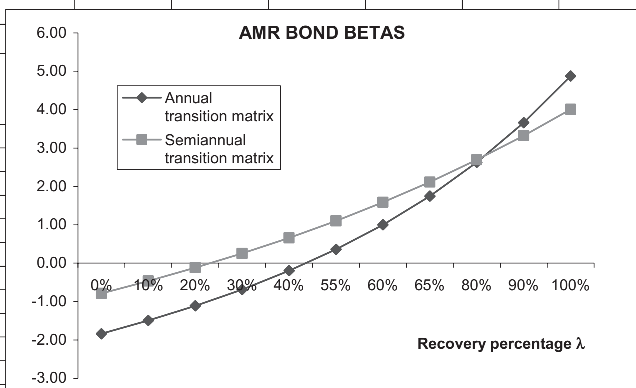
If we use the tax-adjusted version of the SML (see section 3.10), then the bond SML becomes $r_D = Cost\ of\ debt = r_f + \beta_{Debt}[E(r_M) - r_f(1 - T_C)]$. This gives the bond beta as:

	A	B	C
7	**Tax-adjusted SML: $r_D = r_f + \beta_{Debt}*[E(r_M) - r_f*(1-T_C)]$**		
8	Market risk premium, $E(r_m) - r_f$	8.40%	
9	r_f	7%	
10	Corporate tax rate, T_C	40%	
11	Expected bond return	8.00%	
12	Implied bond beta	0.089	<-- =(B11-B9)/(B8+B9*B10)

Using our data for AMR, we get, for the classical SML model:

COMPARING AMR BOND BETAS
SEMIANNUAL VS. ANNUAL TRANSITION MATRICES
Using actual dates and XIRR

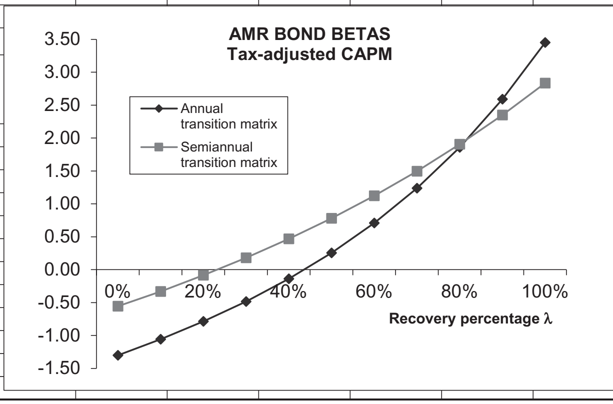
$E(r_M)$	8%	
r_f	3.90%	
	BOND BETA	
Recovery percentage, λ	**Annual transition matrix**	**Semiannual transition matrix**
0%	-1.84	-0.79
10%	-1.49	-0.47
20%	-1.11	-0.12
30%	-0.68	0.25
40%	-0.20	0.66
55%	0.36	1.10
60%	1.00	1.59
65%	1.75	2.11
80%	2.63	2.69
90%	3.66	3.32
100%	4.88	4.01



If we assume that the corporate tax rate is $T_C = 40\%$, then the tax-adjusted CAPM gives the following betas:

COMPARING AMR BOND BETAS
Using the tax-adjusted CAPM

$E(r_M)$	8%	
r_f	3.90%	
Corp. tax rate, T_C	40.00%	
	BOND BETA	
Recovery percentage, λ	**Annual transition matrix**	**Semiannual transition matrix**
0%	-1.30	-0.56
10%	-1.06	-0.33
20%	-0.79	-0.08
30%	-0.48	0.18
40%	-0.14	0.47
55%	0.25	0.78
60%	0.71	1.12
65%	1.24	1.50
80%	1.86	1.90
90%	2.59	2.35
100%	3.45	2.84



If these bond betas seem large, note that the AMR bond has a maturity comparable to these long-term Treasury bonds and has in addition considerable default risk.

Another fact which helps place the AMR bond beta into context is AMR's stock beta. According to Yahoo this beta is 3.617:

YAHOO! FINANCE

Finance Home - My Yahoo - Yahoo! - Help

Monday, July 25, 2005, 9:53AM ET - U.S. Markets close in 6 hours and 7 minutes. Dow +0.05% Nasdaq -0.04%

To track stocks & more, Register

Quotes & Info — Enter Symbol(s): e.g. YHOO, ^DJI — GO — Symbol Lookup | Finance Search

AMR Corp (AMR) — At 9:33AM ET: 13.61 ↓ 0.03 (0.22%)

MORE ON AMR

Quotes: Summary, Real-Time ECN, Options, Historical Prices
Charts: Basic Chart, Technical Analysis
News & Info: Headlines, Company Events, Message Board
Company: Profile, Key Statistics, SEC Filings, Competitors, Industry
Analyst Coverage

Key Statistics — Get Key Statistics for: GO

Data provided by Capital IQ, except where noted.

VALUATION MEASURES	
Market Cap (intraday):	2.20B
Enterprise Value (25-Jul-05)³:	13.40B
Trailing P/E (ttm, intraday):	N/A
Forward P/E (fye 31-Dec-06)¹:	N/A
PEG Ratio (5 yr expected):	N/A
Price/Sales (ttm):	0.12
Price/Book (mrq):	N/A
Enterprise Value/Revenue (ttm)³:	0.71
Enterprise Value/EBITDA (ttm)³:	12.133

FINANCIAL HIGHLIGHTS	
Fiscal Year	
Fiscal Year Ends:	31-Dec
Most Recent Quarter (mrq):	31-Mar-05
Profitability	

TRADING INFORMATION	
Stock Price History	
Beta:	3.617
52-Week Change:	54.47%
S&P500 52-Week Change:	7.66%
52-Week High (07-Jun-05):	14.95
52-Week Low (20-Oct-04):	6.34
50-Day Moving Average:	13.10
200-Day Moving Average:	10.71
Share Statistics	
Average Volume (3 month):	4,197,350
Average Volume (10 day):	4,896,860
Shares Outstanding:	161.39M
Float:	130.15M
% Held by Insiders:	1.40%
% Held by Institutions:	97.60%

23.9 Summary

In this chapter we have shown how to compute the expected return on a risky bond using a simple technique involving rating transitions. Computing a bond's expected returns puts the bond analysis on the same footing as the analysis of stocks. Expected returns—common in the analysis of stocks—are rarely computed for bonds, where the common analysis is in terms of yields to maturity. But the yield to maturity of a bond, essentially the bond's IRR based on its promised future payments, includes an ill-defined premium for the bond's default.

Having computed a bond's expected return, we can then compute its beta using the security market line (SML). Compared to the vast efforts to compute and calibrate stock betas, relatively little research energy has been expended on bond betas. The technique illustrated in this chapter, based on the transition matrix of the bond ratings, is relatively new. This technique still has to be refined and thoroughly tested by academic research. Several refinements to the rating-based technique for computing expected bond returns still need to be explored. These include:

- *Better transition matrices*. Transition matrices need to be refined, and perhaps made industry specific. (The problem with industry-specific data is that the number of observations drops dramatically. Nevertheless, there are examples of such data [for example, a Standard & Poor's 2004 study on real estate–backed loans cited in the Selected References).
- *Time-dependent transition matrices*. Our technique assumes that transition matrices are stationary—constant through time. Perhaps better techniques can be developed that allow for matrices to change with time. For example, we would expect that in difficult economic conditions, the ratings transition matrix would "shift to the right"—that the probabilities of a given rating getting worse over any period would increase.
- *More data on recovery ratios*.

Exercises

1. A newly issued bond with 1 year to maturity has a price of 100, which equals its face value. The coupon rate on the bond is 15%; the probability of default in 1 year is 35%; and the bond's payoff in default will be 65% of its face value.

   a. Calculate the bond's expected return.

   b. Create a data table showing the expected return as a function of the recovery percentage and the price of the bond.

2. Consider the case of five possible rating states, A, B, C, D, and E. A, B, and C are initial bond ratings, D symbolizes first-time default, and E indicates default in the previous period. Assume that the transition matrix Π is:

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.06 & 0.90 & 0.03 & 0.01 & 0 \\ 0.02 & 0.05 & 0.88 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

   A 10-year bond issued today at par with an A rating is assumed to bear a coupon rate of 7%.

   - If a bond is issued today at par with a B rating and with a recovery percentage of 50%, what should be its coupon rate so that its expected return will also be 7%?
   - If a bond is issued today at par with a C rating and with a recovery percentage of 50%, what should be its coupon rate so that its expected return will be 7%?

3. Using the transition matrix of the previous problem: A C-rated bond is selling at par on 18 July 2007. The bond's maturity is 17 July 2017, it has a coupon (paid annually on 17 July) of 11%, and it has a recovery percentage of $\lambda = 67\%$. What is the bond's expected return?

4. An underwriter issues a new 7-year B-rated bond with a coupon rate 9%. If the expected rate of return on the bond is 8%, what is the bond's implied recovery percentage λ? Assume the transition matrix given in section 23.5.

5. An underwriter issues a new 7-year C-rated bond at par. The anticipated recovery rate in default of the bond is expected to be 55%. What should be the coupon rate on the bond so that its expected return is 9%? Assume the transition matrix of exercise 2.

V MONTE CARLO METHODS

Section V of *Financial Modeling* shows how to simulate financial problems in Excel. Traditional finance theory concentrates on the solution of financial problems: What is the optimal portfolio? What is the price of an option? Simulation is no substitution for solution of problems, but it often gives new insights into the nature of the uncertainty underlying the problem. In some cases—for example the path-dependent options discussed in Chapter 30—there is no concise pricing solution, and simulation methods are the best way to arrive at an acceptable price.

Our aim in this section is to show the reader how to conceive of financial simulation and how to build simulations using Excel without additional add-ins. Chapter 24 is the basis of this discussion, showing how Excel's random-number generator can be used to generate various distributions. We include in this chapter two methods for producing correlated random numbers.

"Monte Carlo" refers to various simulation techniques that can be used to compute values of complicated functions that often have no analytical solution. Chapter 25 discusses the basics of Monte Carlo valuation. Its basic example shows how to compute the value of π using a simulation.

Most financial pricing models assume that asset returns are lognormally distributed. In Chapter 26 we discuss this assumption and show how it can be simulated. Chapter 27 extends the simulations to discuss outcomes of investing in portfolios, and Chapter 28 discusses value at risk (VaR) in the context of simulation models.

Finally Chapters 29 and 30 examine simulation methods as applied to options. In Chapter 29 we discuss the simulation of portfolio insurance, and in Chapter 30 we discuss Monte Carlo methods for pricing path-dependent options.

24 Generating and Using Random Numbers

24.1 Overview

In this chapter we discuss techniques for computing random numbers. We use random numbers extensively in Chapters 25–30 to simulate stock prices, investment strategies, and option strategies. In this chapter we show how to produce both uniformly distributed and normally distributed random numbers.

A random-number generator on a computer is a function that produces a seemingly unrelated set of numbers. The question of *what is* a random number is a philosophical one.[1] In this chapter we will ignore philosophy and concentrate on some simple random-number generators—primarily the Excel random-number generator **Rand( )** and the VBA random-number generator **Rnd**.[2] We will show how to use these generators to produce uniform random numbers and subsequently random numbers that are normally distributed. At the end of the chapter we use the Cholesky decomposition to produce correlated random numbers.

To imagine a set of uniformly distributed random numbers think of an urn filled with 1,000 little balls, numbered 000, 001, 002, … , 999. Suppose we perform the following experiment: Having shaken the urn to mix up the balls, we draw one ball out of the urn and record the ball's number. Next we put the ball back into the urn, shake the urn thoroughly so that the balls are mixed up again, and then draw out a new ball. The series of numbers produced by repeating this procedure many times should be *uniformly distributed* between 000 and 999.

A random-number generator on a computer is a function that imitates this procedure. The random-number generators considered in this chapter are sometimes termed *pseudo-random-number generators*, since they are actually deterministic functions whose values are indistinguishable from random numbers. All pseudo-random-number generators have cycles (i.e., they eventually start to repeat themselves). The trick is to find a random-number generator

1. Philosophical? Perhaps theological. Knuth (1981, p. 142) gives the following quote: "A random sequence is a vague notion embodying the idea of a sequence in which each term is unpredictable to the uninitiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence is to be put" (attributed to D. H. Lehmer, 1951).

2. In this book we usually write Excel functions in boldface without the parentheses. In this chapter we generally write **Rand( )** with the parentheses to emphasize that (1) the parentheses are necessary, and (2) that they are empty.

with a long cycle. The Excel **Rand( )** function has very long cycles and is a respectable random-number generator.

If you've never used a random-number generator, open an Excel spreadsheet and type **=Rand( )** in any cell. You will see a 15-digit number between 0.000000000000000 and 0.999999999999999. Every time you recalculate the spreadsheet (e.g., by pressing the **F9** key), the number changes. We leave the technical details of how **Rand( )** works for the exercises to this chapter, where we show you how to design your own random-number generator. Suffice it to say, however, that the series of numbers produced by the function should be (to use Lehmer's terminology from footnote 1) "unpredictable to the uninitiated."

In this chapter we shall deal with several kinds of random-number generators: We first examine the uniform random-number generators which come with Excel and VBA. Subsequently we generate normally distributed random numbers.[3] Finally we generate correlated random numbers using the Cholesky decomposition.

24.2 Rand( ) and Rnd: The Excel and VBA Random-Number Generators

Suppose you simply wanted to generate a list of random numbers. One way to do this would be to copy the Excel function **Rand( )** to a range of cells.

	A	B	C	D	E
1	**USING EXCEL'S RAND( ) FUNCTION**				
2	0.6230	0.9983	0.2132	0.3381	<-- =RAND()
3	0.3836	0.7527	0.9139	0.3635	
4	0.5948	0.7089	0.9563	0.1333	
5	0.4543	0.7327	0.1095	0.9702	
6	0.0250	0.1392	0.9793	0.5049	
7	0.5001	0.3219	0.1293	0.2255	
8	0.8931	0.4278	0.8038	0.2239	
9	0.5847	0.9270	0.6634	0.5449	
10	0.4985	0.2468	0.8391	0.5452	
11					
12	Each cell contains the function **Rand( )**. Each time you update the spreadsheet or press F9 the block of cells will produce a new set of random numbers.				

3. A common nomenclature speaks of "random deviates." Only in financial engineering can one find "normal deviates"!

In section 24.3 we will develop a crude test of how well **Rand( )** works.

Using VBA's Rnd Function

VBA contains its own function **Rnd**, which is equivalent to the Excel **Rand** function.[4] Here's a small VBA program which illustrates a basic use of the **Rnd** function:

```
Sub RandomList()
'Produces a simple list of random numbers
   For Index = 1 To 10
       Range("A4").Cells(Index, 1) = Rnd
   Next Index
End Sub
```

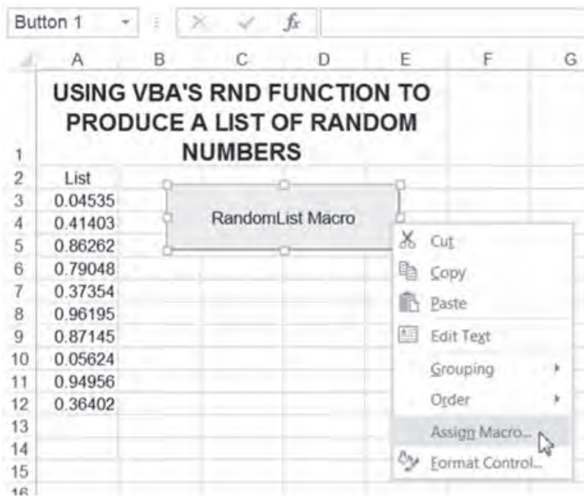
In the spreadsheet below, the VBA program has been assigned to a button, so that each time we click on the button, it runs a VBA program that produces 10 random numbers:

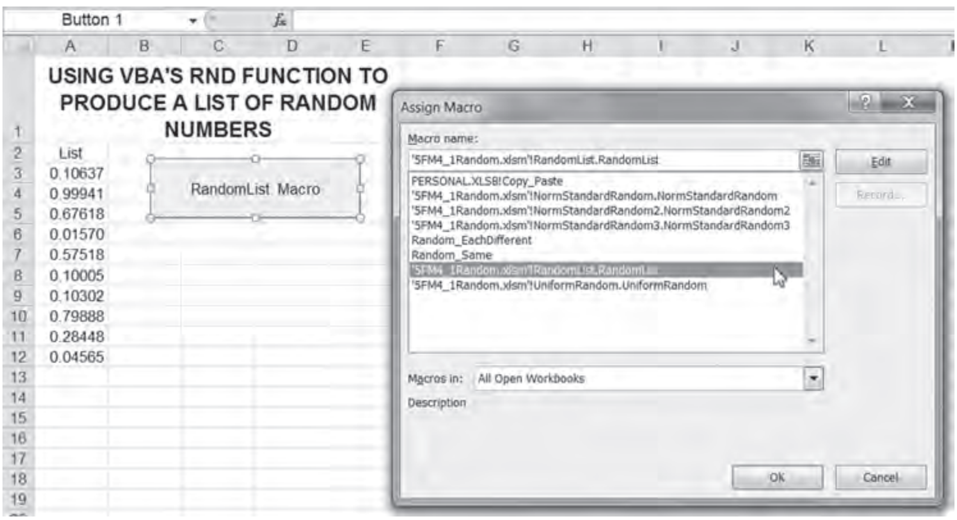
	A	B	C	D	E
1	USING VBA'S RND FUNCTION TO PRODUCE A LIST OF RANDOM NUMBERS				
2	List				
3	0.04535				
4	0.41403		RandomList Macro		
5	0.86262				
6	0.79048				
7	0.37354				
8	0.96195				
9	0.87145				
10	0.05624				
11	0.94956				
12	0.36402				

4. Confusing, no? Two different functions in the same computer package which do the same thing …

Assigning a Macro to a Button or to a Control Sequence

In the spreadsheet pictured on page 609, we have assigned the macro **RandomList** to the button marked "RandomList Macro." Any drawing shape in Excel can be assigned a VBA program. In this case we have created a rectangle; right-clicking on this rectangle, we have assigned it a macro:





24.3 Testing Random-Number Generators

Producing lists of random numbers is interesting, though a bit uninformative. Is the list of numbers thus produced really uniformly distributed? A simple test is to generate each number and determine whether it falls into the interval [0,0.1), [0.1,0.2), … , [0.9,1). The notation *[a,b)* denotes the *half-open* interval between *a* and *b*; a number *x* is in this interval if $a \le x < b$. If the list of numbers is really uniformly distributed, we would expect roughly an even number of the "random" numbers to be in each of the 10 intervals.

One way to test this is to generate a list of random numbers on the spreadsheet by copying **Rand( )** to many cells and then using the Excel array function **Frequency(data_array,bins_array)**.[5] This is illustrated in the following spreadsheet picture:

	A	B	C	D	E	F
1	**USING EXCEL'S FREQUENCY FUNCTION TO TEST THE DISTRIBUTION OF RAND()**					
2	**Random numbers**			**Bin**	**Frequency**	
3	0.8978	<-- =RAND()		0.1	0	
4	0.8354			0.2	1	
5	0.5188			0.3	1	
6	0.7317			0.4	0	
7	0.5067			0.5	0	<-- =FREQUENCY(A3:A12,D3:D12)
8	0.2418			0.6	4	
9	0.6406			0.7	1	
10	0.1228			0.8	1	
11	0.5611			0.9	2	
12	0.5543			1	0	
13						
14	Each cell in the range A3:A12 contains the formula **Rand()**. Pressing F9 will produce a new set of random numbers and frequencies.					

5. Array functions are explained in Chapter 34.

Writing **Frequency(A:A,D3:D12)** refers to all non-empty cells in column A:

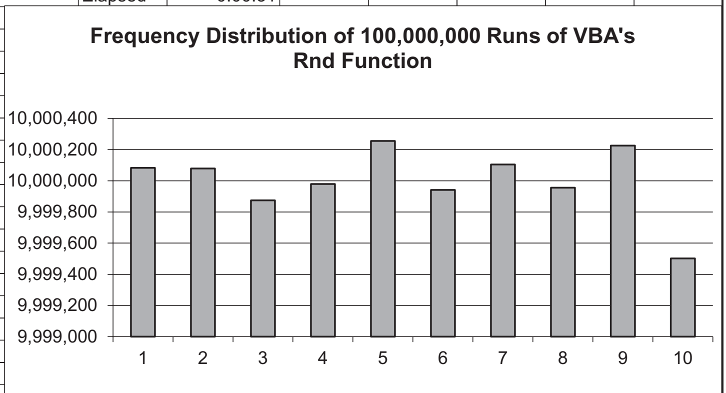
	A	B	C	D	E	F
1	**USING EXCEL'S FREQUENCY FUNCTION TO TEST THE DISTRIBUTION OF RAND()**					
2	0.724913	<-- =RAND()		**Bin**	**Frequency**	
3	0.624834			0.1	155	
4	0.093701			0.2	150	
5	0.019565			0.3	154	
6	0.135518			0.4	148	
7	0.248544			0.5	155	<-- =FREQUENCY(A:A,D3:D12)
8	0.195843			0.6	142	
9	0.532975			0.7	159	
10	0.231059			0.8	148	
11	0.495747			0.9	139	
12	0.016572			1	145	
13	0.526736					
14	0.211719			Total	1495	<-- =SUM(E3:E12)
15	0.214072					
16	0.287068					
17	0.877499					
18	0.058629					
19	0.154429					

This method is obviously not efficient (or even feasible) when we want to test the random-number generator for large numbers of random draws. The following program uses VBA to generate many random numbers and puts them into the bins in range A3:A12:

```
Sub UniformRandom()
'Puts random numbers into bins

   Range("E3") = Time
  'the number of random draws
   N = Range("B2").Value

   Dim distribution(10) As Long 'bins

   For k = 1 To N
      draw = Rnd
      distribution(Int(draw * 10) + 1) = _
         distribution(Int(draw * 10) + 1) + 1
   Next k

   For Index = 1 To 10
      Range("B5").Cells(Index, 1) = distribution(Index)
   Next Index

   Range("E4") = Time

End Sub
```

In the spreadsheet below we have generated 100 million random numbers in 25 seconds:

	A	B	C	D	E	F	G	H	I	J
1	UNIFORMLY-DISTRIBUTED RANDOM NUMBERS									
2	Runs	100,000,000								
3				Starttime	8:55:57		UniformRandom Macro			
4	Bin	Output		Stoptime	8:56:28					
5	1	10,000,082		Elapsed	0:00:31					
6	2	10,000,079								
7	3	9,999,875								
8	4	9,999,979								
9	5	10,000,256								
10	6	9,999,942								
11	7	10,000,105								
12	8	9,999,955								
13	9	10,000,225								
14	10	9,999,502								
15										
16										
17										
18										
19										
20										
21										
22										
23										



Here are some things to note about **UniformRandom**:

• The program has a "clock" to measure the amount of time it takes to run. At the start of the program, we use **Range("E3")=Time** to put the current time into cell E3. At the end of the program, **Range("E4")=Time** puts in the ending time. The cell **elapsed** contains the formula **=stoptime-starttime**. Note that in order for the cells to read correctly, you have to use the command **Format|Cells|Number|Time** on the relevant cells.

• The heart of the program uses the function `Int(draw * 10) + 1`. Multiplying the random draw by 10 produces a number whose first digit is 0, 1, … , or 9. The VBA function **Int** gives this integer. **Distribution** is a VBA array numbered 1 to 10, with **Distribution(1)** being the number of random numbers in [0,0.1), **Distribution(2)** the number of random numbers in [0.1,0.2), etc. Thus `Int(draw * 10) + 1` is the proper place in **Distribution** to which the current random draw belongs.

Using Randomize to Produce the Same List (or Not) of Random Numbers

Most random-number generators use the last generated "random" number to produce the next.[6] The first number used in a particular sequence is controlled by the "seed," which is typically taken from the computer's clock. VBA's **Rnd** is no exception, but it allows you to control the seed by using the command **Randomize**. The two small programs below illustrate two uses of this command.

• Using **Randomize** without any numeric argument resets the seed (meaning—it breaks the connection between the next random number and the current random number). This is illustrated in the macro **Random_EachDifferent**, though it is difficult to see the effect.

```
Sub Random_EachDifferent()
'Produces a list of random numbers
Randomize
'Initializes the VBA random number generator
   For Index = 1 To 10
       Range("A5").Cells(Index, 1) = Rnd()
   Next Index
End Sub
```

• Using **Randomize(seed)** uses a particular number as the seed.

• Using the sequence of commands **Rnd(negative number)** and **Randomize(seed)** guarantees the same sequence of random numbers. This is illustrated in the macro **Random_Same**.

6. There are more examples of this in the exercises at the end of the chapter.

```
Sub Random_Same()
'Produces the same list of random numbers
'which is always the same
Rnd (-4)
'Initializes the VBA random number generator
Randomize (Range("seed"))
   For Index = 1 To 10
      Range("B5").Cells(Index, 1) = Rnd()
   Next Index
End Sub
```

In the spreadsheet below, pushing the top button produces a random set of random numbers. Pushing the bottom button activates the macro **Random_ Same** and produces the same set of random numbers each time—provided the **Seed** (cell B2) isn't changed.

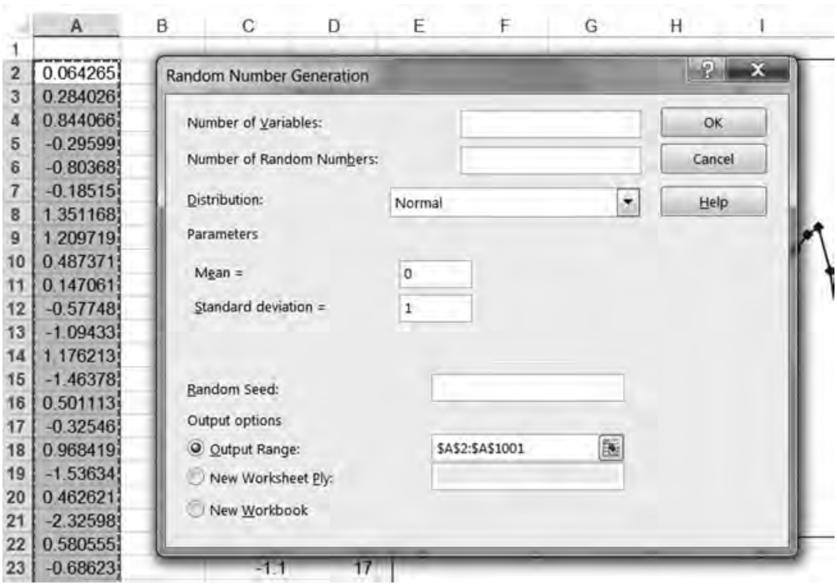
	A	B	C
1	**PRODUCING LISTS OF RANDOM NUMBERS**		
2	Seed	334	
3			
4	**Output: Each run different**	**Output: Each run same**	Run Random_EachDifferent
5	0.54165	0.29708	
6	0.50241	0.70653	
7	0.99067	0.65463	
8	0.85176	0.96848	
9	0.97838	0.48999	
10	0.40634	0.72373	Run Random_Same
11	0.88656	0.06518	
12	0.59110	0.60034	
13	0.72938	0.25382	
14	0.49635	0.70398	
15			
16	**Note**: to see the effect of the "Run Random_Same" button, erase the cells B5:B14. Changing the seed in cell B2 changes the output in column B.		

24.4 Generating Normally Distributed Random Numbers

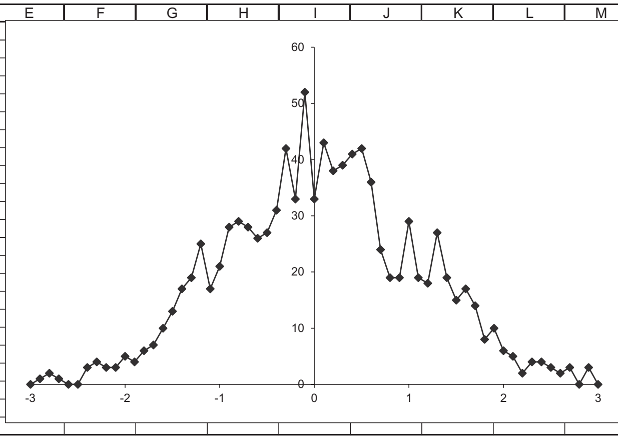
In the sections above we have generated numbers which are uniformly distributed. In this section we explore four ways to produce normally distributed random numbers using Excel.

Method 1: Normally Distributed Numbers Using Data|Data Analysis|Random Number Generation

One way to do this is to use the Excel command **Data|Data Analysis|Random Number Generation**. Here's how we get Excel to produce 1,000 random numbers which are normally distributed (with $\mu = 0$ and $\sigma = 1$) in column A of the spreadsheet:



	A	B	C	D
2	0.064265			
3	0.284026		Bin	Frequency
4	0.844066		-3	0
5	-0.29599		-2.9	1
6	-0.80368		-2.8	2
7	-0.18515		-2.7	1
8	1.351168		-2.6	0
9	1.209719		-2.5	0
10	0.487371		-2.4	3
11	0.147061		-2.3	4
12	-0.57748		-2.2	3
13	-1.09433		-2.1	3
14	1.176213		-2	5
15	-1.46378		-1.9	4
16	0.501113		-1.8	6
17	-0.32546		-1.7	7
18	0.968419		-1.6	10
19	-1.53634		-1.5	13
20	0.462621		-1.4	17
21	-2.32598		-1.3	19
22	0.580555		-1.2	25
23	-0.68623		-1.1	17
24	1.24697		-1	21



If we want to see whether the output is distributed normally we can have Excel do a frequency distribution (either by using the array function **Frequency** or by using **Data|Data Analysis|Histogram**). As the graph above shows, the output appears to be normally distributed.

Method 2: Normally Distributed Numbers Using Norm.S.Inv(Rand( ))

The Excel function **Norm.S.Inv(Rand( ))** can be used to generate normally distributed random numbers. To understand the use of this function, we start with an explanation of the function **Norm.S.Dist**. The Excel function **Norm.S.Dist** computes values of the standard normal distribution. In the spreadsheet below, for example, we use **Norm.S.Dist(0.5,1)** to compute N(0.5)—the probability that a random variable that is standard normally distributed will be < 0.5. We also use **Norm.S.Dist(1,1) – Norm.S.Dist(1,–1)** to compute the percentage of the standard normal distribution that is between –1 and +1:

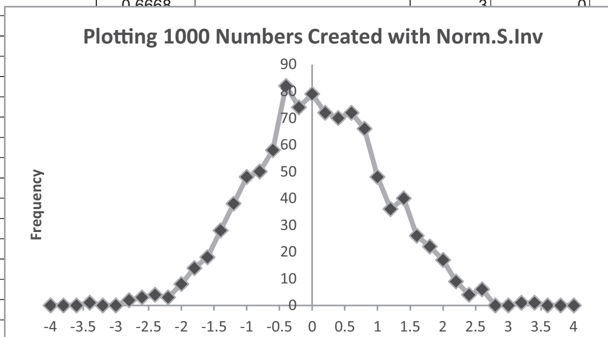
	A	B	C
1	**USING NORM.S.DIST**		
2	x	0.500	
3	Norm.S.Dist	0.6915	<-- =NORM.S.DIST(B2,1)
4			
5	x_1	1	
6	x_2	-1	
7	$N(x_2)-N(x_1)$	0.68269	<-- =NORM.S.DIST(B5,1)-NORM.S.DIST(B6,1)

Excel's **Norm.S.Inv( )** function inverts the function **Norm.S.Dist**. Given a number **x** between 0 and 1, **Norm.S.Inv(x)** produces number **y** such that **Norm.S.Dist(y,1) = x**. The function **Norm.S.Dist(Rand( ))** should produce a set of random numbers which is distributed standard normal:

	A	B	C
1	**NORMALLY DISTRIBUTED RANDOM NUMBERS USING NORM.S.INV( )**		
2	Any number between 0 and 1	0.6000	
3	Normal number	0.2533	<-- =NORM.S.INV(B2)
4	Check:	0.6000	<-- =NORM.S.DIST(B3,1)
5			
6	Random normal number	0.8281	<-- =NORM.S.DIST(RAND(),1)

In the spreadsheet below we produce 1,000 iterations of **Norm.S.Inv (Rand( ))** and graph the resulting frequencies. They look normally distributed:

	A	B	C	D	E	F
1	NORMALLY DISTRIBUTED RANDOM NUMBERS USING NORM.S.INV( )					
2	Random numbers produced with Rand()	NormSInv		Bin	Frequency	
3	0.4308	-0.1744	<-- =NORM.S.INV(A3)	-4	0	
4	0.4311	-0.1736	<-- =NORM.S.INV(A4)	-3.8	0	
5	0.6535	0.3949	<-- =NORM.S.INV(A5)	-3.6	0	<-- {=FREQUENCY(B:B,D3:D43)}
6	0.2190	-0.7755	<-- =NORM.S.INV(A6)	-3.4	1	
7	0.3964	-0.2626		-3.2	0	
8	0.7476	0.6668		-3	0	
9	0.8786					
10	0.9265					
11	0.0052					
12	0.2191					
13	0.9435					
14	0.2059					
15	0.2036					
16	0.6048					
17	0.5326					
18	0.2661					
19	0.1243					
20	0.2814					
21	0.8127					
22	0.5243					
23	0.4826					
24	0.5602					
25	0.9075	1.3256		0.4	70	



Method 3: Incorporating Norm.S.Inv( ) into VBA

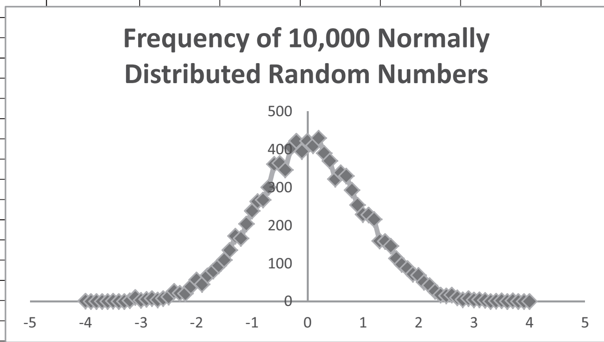
The VBA program **NormStandardRandom** uses **Norm.S.Inv** to produce random deviates. Here is the program and below is the output it produces. Note that in VBA we replace the periods in the function with underscores, writing **Norm_S_Inv**.

```
Sub NormStandardRandom()
'Produces a list of normally-distributed
'random numbers
'Randomize initializes the
'VBA random number generator
Randomize
Application.ScreenUpdating = False
Range("E2") = Time
  Range("A8").Range(Cells(1, 1), _
  Cells(64000, 1)).Clear
  N = Range("B2").Value

  For Index = 1 To N
     Range("A8").Cells(Index, 1) = _
     Application.WorksheetFunction.NormSInv(Rnd)
  Next Index
Range("E3") = Time
End Sub
```

The program **NormStandardRandom** includes two lines which measure the time taken for the whole simulation to run. The program is very slow, largely because of repeated calls on the spreadsheet function. As you can see below, 10,000 runs of the program take about 45 seconds on the author's Lenovo T420s. Here's a sample screen (the button operates the macro):

	A	B	C	D	E	F	G	H	I	J	K	L
1	**Normally Distributed Random Numbers** **using VBA and Excel's Norm.S.Inv() function**											
2	Runs	10,000		Starttime	19:43:33							
3				Stoptime	19:44:17							
4				Elapsed	0:00:44	<-- =E3-E2						
5												
6			**Frequency Distribution**									
7	**Output**		**Bin**	**Frequency**								
8	-0.07925		-4	1								
9	0.695962		-3.9	0								
10	0.314898		-3.8	0								
11	-0.62788		-3.7	0								
12	-0.5172		-3.6	0								
13	-0.69972		-3.5	1								
14	-0.11278		-3.4	0								
15	-1.00098		-3.3	0								
16	0.549194		-3.2	2								
17	-2.16114		-3.1	11								
18	-0.22532		-3	3								
19	0.565778		-2.9	6								
20	-2.01698		-2.8	8								
21	-0.2967		-2.7	4								
22	0.340505		-2.6	8								
23	-0.4739		-2.5	12								
24	0.382405		-2.4	27								
25	-0.15375		-2.3	22								
26	1.318265		-2.2	21								

NormStandardRandom
uses Norm.S.Inv()



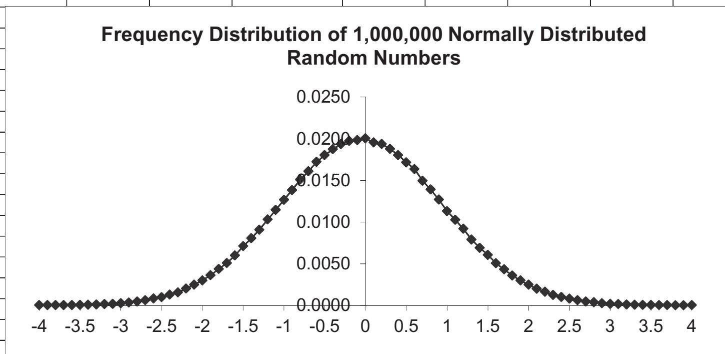
A Faster Version of Method 3

We can make method 3 much faster by storing all the data in VBA and only writing the final frequency distribution on the screen:

```
Sub NormStandardRandom2()
'Randomize Initializes the VBA
'random number generator
Randomize
Dim distribution(-40 To 40) As Double
Application.ScreenUpdating = False
Range("E2") = Time
  N = Range("B2").Value

  For Index = 1 To N
     X = Application.Norm_S_Inv(Rnd())

  If X < -4 Then
      distribution(-40) = distribution(-40) + 1
   ElseIf X > 4 Then
      distribution(40) = distribution(40) + 1
   Else: distribution(Int(X / 0.1)) = _
   distribution(Int(X / 0.1)) + 1
   End If

  Next Index

For Index = -40 To 40
   Range("B7").Cells(Index + 41, 1) = _
   distribution(Index) / (2 * N)
Next Index

Range("E3") = Time
End Sub
```

Here's the output for 100,000 iterations. Note the time in cell E4:

	A	B	C	D	E	F	G	H	I	J	K
1	**Normally Distributed Random Numbers** This sheet saves time by not recording the random numbers on the screen										
2	Runs	1,000,000		Starttime	19:47:53						
3				Stoptime	19:47:59		NormStandardRandom2				
4				Elapsed	0:00:06	<-- =E3-E2					
5											
6	**Bin**	**Output**									
7	-4	0.0000245									
8	-3.9	0.0000140									
9	-3.8	0.0000165									
10	-3.7	0.0000250									
11	-3.6	0.0000310									
12	-3.5	0.0000460									
13	-3.4	0.0000645									
14	-3.3	0.0001030									
15	-3.2	0.0001430									
16	-3.1	0.0001755									
17	-3	0.0002550									
18	-2.9	0.0003460									
19	-2.8	0.0004600									
20	-2.7	0.0006005									
21	-2.6	0.0008165									
22	-2.5	0.0009685									
23	-2.4	0.0012905									
24	-2.3	0.0015430									



There are a few things to note about this program:

- Most of the results of the normal distribution are between –4 and +4. When, in **NormStandardRandom2**, we classify the output into bins, we want these bins to be (–∞, –3.9], (–2.9, –2.8], … , (–3.9, ∞). To do this we first define an array `distribution(–40 To 40)`; this array has 81 indices. To classify a particular random number (say, *X*) into the bins of this array, we use the function:

```
If X < -4 Then
    distribution(-40) = distribution(-40) + 1
ElseIf X > 4 Then
    distribution(40) = distribution(40) + 1
Else: distribution(Int(X / 0.1)) = _
distribution(Int(X / 0.1)) + 1
End If
```

• **NormStandardRandom2** produces not a histogram (which is a *count* of how many times a number falls into a particular bin), but a *frequency distribution*. We do this by dividing by twice the number of runs (remember that each successful run produces two random numbers), 2*N*, before we output the data to the spreadsheet:

```
For Index = -40 To 40
    Range("output").Cells(Index + 41, 1) = _
        distribution(Index) / (2 * N)
Next Index
```

• Finally, note that the command `Application.ScreenUpdating = False` makes a big difference! This command prevents both the updating of the output in the cells and the Excel chart. Try running the program with and without this command to see the effect.

Method 4: The Box-Muller Method

The Box-Muller method for creating randomly distributed normal deviates is the fastest method of the four.[7] The eight lines which follow `Start` in the

7. See Box-Muller (1958) or Knuth (1981).

VBA program below define a routine which in each successful iteration creates two numbers which are drawn from a standard normal distribution. The routine creates two random numbers, $rand_1$ and $rand_2$, between −1 and +1. If the sum of the squares of these numbers is within the unit circle, then the two normal deviates are defined by

$$\{X_1, X_2\} = \left\{ rand_1 * \sqrt{\frac{-2\ln(S_1)}{S_1}}, rand_2 * \sqrt{\frac{-2\ln(S_1)}{S_1}} \right\}$$

where

$$S_1 = rand_1^2 + rand_2^2$$

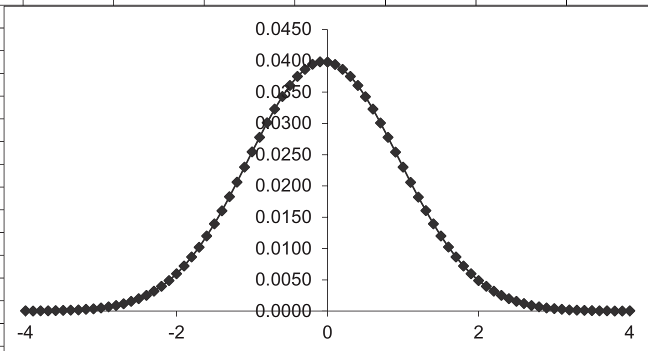
Here's the VBA program:

```
Sub NormStandardRandom3()
'Box-Muller for producing
'standard normal deviates

Dim distribution(-40 To 40) As Long

Range("E2") = Time
N = Range("B2").Value

Application.ScreenUpdating = False
For Index = 1 To N

start:
   Static rand1, rand2, S1, S2, X1, X2
   rand1 = 2 * Rnd - 1
   rand2 = 2 * Rnd - 1
   S1 = rand1 ^ 2 + rand2 ^ 2
   If S1 > 1 Then GoTo start
   S2 = Sqr(-2 * Log(S1) / S1)
   X1 = rand1 * S2
   X2 = rand2 * S2
```

```
    If X1 < -4 Then
       distribution(-40) = distribution(-40) + 1
    ElseIf X1 > 4 Then
       distribution(40) = distribution(40) + 1
    Else: distribution(Int(X1 / 0.1)) = _
    distribution(Int(X1 / 0.1)) + 1
    End If

    If X2 < -4 Then
       distribution(-40) = distribution(-40) + 1
    ElseIf X2 > 4 Then
       distribution(40) = distribution(40) + 1
    Else: distribution(Int(X2 / 0.1)) = _
    distribution(Int(X2 / 0.1)) + 1
    End If

Next Index

For Index = -40 To 40
    Range("B7").Cells(Index + 41, 1) = _
    distribution(Index) / (2 * N)
Next Index

Range("E3") = Time

End Sub
```

This routine is very fast; in the spreadsheet below we produce 10 million normal variates in 18 seconds:

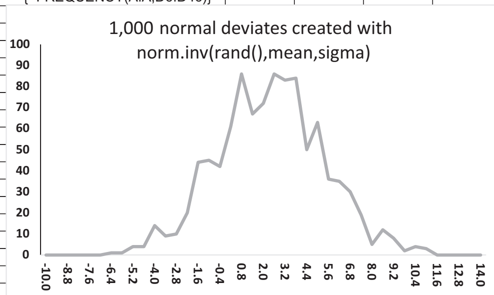
	A	B	C	D	E	F	G	H	I	J	K
1	**BOX-MULLER ROUTINE FOR STANDARD NORMAL**										
2	Runs	10,000,000		Starttime	10:54:28						
3				Stoptime	10:54:46			NormStandardRandom3			
4				Elapsed	0:00:18	<-- =E3-E2		Uses Box-Muller routine			
5											
6	**Bin**	**Output**									
7	-4	0.0000									
8	-3.9	0.0000									
9	-3.8	0.0000									
10	-3.7	0.0001									
11	-3.6	0.0001									
12	-3.5	0.0001									
13	-3.4	0.0001									
14	-3.3	0.0002									
15	-3.2	0.0003									
16	-3.1	0.0004									
17	-3	0.0005									
18	-2.9	0.0007									
19	-2.8	0.0009									
20	-2.7	0.0012									
21	-2.6	0.0015									
22	-2.5	0.0020									



24.5 Norm.Inv: Another Way to Generate Normal Deviates

The Excel function **Norm.Inv(rand( ),mean,sigma)** can also generate normal deviates. Whereas **Norm.S.Inv(rand( ))** generates only standard normal deviates, we can use **Norm.Inv** to change the mean and standard deviation of our deviates. In Chapter 26, we will sometimes use this function to generate normally distributed stock returns.

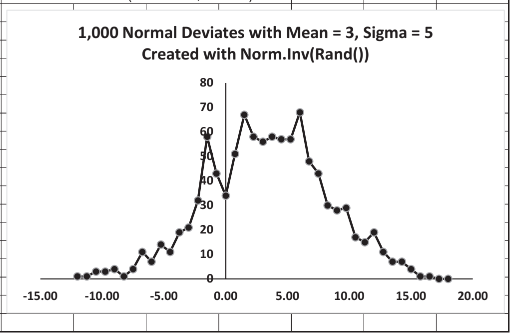
	A	B	C	D	E	F	G	H	I	J
1	NORMALLY DISTRIBUTED RANDOM NUMBERS USING NORM.INV(rand(),mean,sigma)									
2	Mean	2								
3	Sigma	3								
4										
5	Norm.Inv(rand(),mean,sigma)			Bin	Frequency					
6	-0.0771	<--		-10.0	0	<-- {=FREQUENCY(A:A,D6:D46)}				
7	-3.1876	=NORM.INV(RAND(),		-9.4	0					
8	0.7521	$B$2,$B$3)		-8.8	0					
9	2.0116			-8.2	0					
10	1.8121			-7.6	0					
11	2.5113			-7.0	0					
12	8.1300			-6.4	1					
13	2.6463			-5.8	1					
14	5.2369			-5.2	4					
15	4.8960			-4.6	4					
16	2.3377			-4.0	14					
17	4.4869			-3.4	9					
18	3.3379			-2.8	10					
19	0.4816			-2.2	20					
20	2.9400			-1.6	44					
21	-2.4114			-1.0	45					
22	3.3061			-0.4	42					
23	1.3751			0.2	61					



Norm.Inv as Alternative to Norm.S.Inv

Norm.S.Inv(Rand( )) inverts the standard normal distribution and can be used to create standard normal deviates. The Excel function **Norm.Inv(Rand( )),mean,sigma)** can be used to create normal deviates that are normally distributed with any normal distribution. The example below illustrates this:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	CREATING NORMAL DEVIATES WITH NORM.INV(RAND())												
2	Mean	3.00											
3	Sigma	5.00											
4					Bin	Frequency							
5	Norm.Inv				-12.00	1	<-- =FREQUENCY(A6:A1005,E5:E45)						
6	1.1512	<-- =NORM.INV(RAND(),$B$2,$B$3)			-11.25	1							
7	-1.7767	<-- =NORM.INV(RAND(),$B$2,$B$3)			-10.50	3							
8	8.8680	<-- =NORM.INV(RAND(),$B$2,$B$3)			-9.75	3							
9	6.9032				-9.00	4							
10	4.4525				-8.25	1							
11	5.2241				-7.50	4							
12	-1.9517				-6.75	11							
13	-2.7898				-6.00	7							
14	1.6515				-5.25	14							
15	-0.4800				-4.50	11							
16	3.5602				-3.75	19							
17	10.7653				-3.00	21							
18	1.6006				-2.25	32							
19	2.0960				-1.50	58							
20	8.3773				-0.75	43							
21	2.5934				0.00	34							
22	5.7685				0.75	51							
23	1.3261				1.50	67							



Getting Ahead of Ourselves: Do We Prefer Norm.S.Inv or Norm.Inv?

Ultimately our goal is to simulate stock returns; these (as we discuss in Chapter 26) are generally assumed to be normally distributed. If we are simulating the returns of a stock based on annual mean μ and standard deviation σ, then it is obviously more convenient to use **Norm.S.Inv(rand( ),μ,σ)**. As we discuss in Chapter 26, if (μ,σ) are the annual statistics of the stock return, and if we divide the year into n subperiods, then the periodic stock returns can be simulated with **Norm.Inv(Rand(), $\mu\Delta t$, $\sqrt{\Delta t}\sigma$)**, where $\Delta t = 1/n$.

On the other hand, much financial theory is phrased in terms of lognormal price processes. These are generally written as

$$r = \mu\Delta t + \sigma\sqrt{\Delta t}Z$$

where Z is a standard normal deviate. To conform to this writing, it is often more convenient (and theoretically equivalent) to simulate returns by:

$$r = \mu\Delta t + \sigma\sqrt{\Delta t}\ \textbf{Norm.S.Inv(Rand( ))}$$

In this book we interchangeably use both writings.

24.6 Generating Correlated Random Numbers

In this section we show how to produce correlated pseudo-random numbers. We start by showing how to generate two correlated normal and uniform random numbers and then go on to the multivariate case.

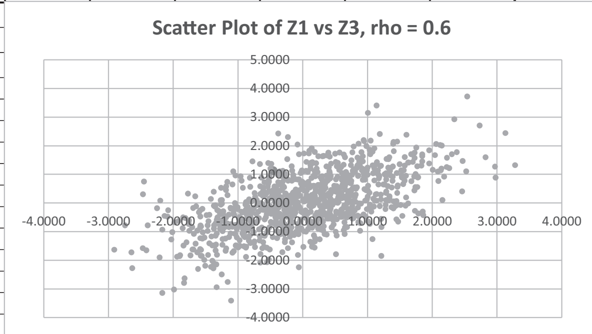
Example 1: Two Correlated Standard Normal Variables

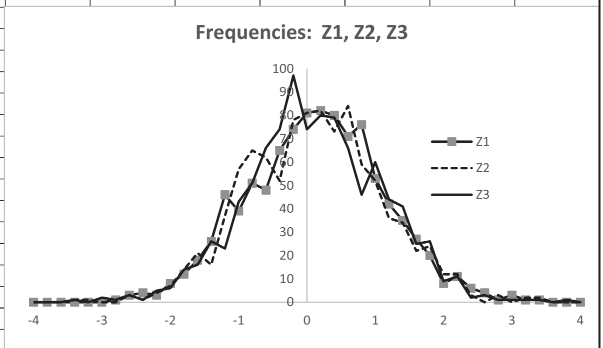
Suppose that Z_1 and Z_2 are standard normal deviates (created in the spreadsheet below with **Norm.S.Inv(Rand( ))**. Defining $z_3 = \rho z_1 + z_2\sqrt{1-\rho^2}$, we create a set of simulates that has the desired correlation with Z_1:

	A	B	C	D	E
1	**1,000 CORRELATED STANDARD NORMAL DEVIATES**				
2	ρ	0.6			
3					
4	Mean	0.0731	-0.0515	0.0026	<-- =AVERAGE(D12:D1011)
5	Sigma	0.9910	1.0322	1.0026	<-- =STDEV.S(D12:D1011)
6	Skewness	0.0572	0.1225	0.0376	<-- =SKEW(D12:D1011)
7	Kurtosis	-0.0962	-0.0599	-0.0923	<-- =KURT(D12:D1011)
8	Count	1000	1000	1000	<-- =COUNT(D12:D1011)
9	Corr(z_1,z_3)	0.5676	<-- =CORREL(B12:B1011,D12:D1011)		
10					
11		Z_1	Z_2	Z_3	
12	=NORM.S.INV(RAND()) -->	-0.9108	-0.2754	-0.7668	<-- =$B$2*B12+SQRT(1-$B$2^2)*C12
13	=NORM.S.INV(RAND()) -->	0.8994	0.2471	0.7373	<-- =$B$2*B13+SQRT(1-$B$2^2)*C13
14	=NORM.S.INV(RAND()) -->	0.3730	-0.7499	-0.3762	<-- =$B$2*B14+SQRT(1-$B$2^2)*C14
15		-0.7975	-0.3404	-0.7508	
16		0.4534	-0.1954	0.1157	
17		-0.5444	-0.4955	-0.7230	
18		1.1833	-1.1735	-0.2288	
19		-1.5330	0.7085	-0.3530	
20		-0.0806	-1.8064	-1.4934	

The frequencies and distributions are shown below:

	G	H	I	J	K
11					
12	Frequency distribution, Z_1, Z_2, Z_3				
13		Z_1	Z_2	Z_3	
14	-4	0	0	0	
15	-3.8	0	0	0	
16	-3.6	0	0	0	
17	-3.4	0	1	1	
18	-3.2	0	1	0	
19	-3	0	0	2	
20	-2.8	1	0	1	
21	-2.6	3	3	3	
22	-2.4	4	1	1	
23	-2.2	3	4	5	
24	-2	8	7	6	
25	-1.8	12	14	14	
26	-1.6	18	21	16	
27	-1.4	26	16	26	
28	-1.2	46	37	23	
29	-1	39	57	43	
30	-0.8	51	65	51	
31	-0.6	48	62	66	
32	-0.4	65	52	74	
33	-0.2	74	78	97	
34	0	81	81	74	
35	0.2	82	82	80	
36	0.4	80	73	79	
37	0.6	71	84	66	
38	0.8	76	59	46	
39	1	53	52	60	
40	1.2	42	36	44	
41	1.4	35	34	41	
42	1.6	27	22	25	
43	1.8	20	24	26	
44	2	8	12	9	





On the next page we give the scatter plots for a number of simulations with different correlation coefficients ρ.

Example 2: Two Correlated Uniform Variables

To create uniform variables that are correlated, we first produce correlated standard normal variables, and then use **Norm.S.Dist** to produce the uniform variables. Here's the procedure:

- Use **Norm.S.Inv(Rand( ))** to produce two random numbers that are normally distributed. Call these numbers z_1 and z_2.
- Let $z_3 = \rho z_1 + z_2\sqrt{1-\rho^2}$. As we showed above z_3 and z_1 are correlated with correlation ρ.

- Now define u_1 = **Norm.S.Dist(z₁,1)** and u_2 = **Norm.S.Dist(z₃,1)**. Then u_1 and u_2 are uniformly distributed and correlated with correlation ρ.

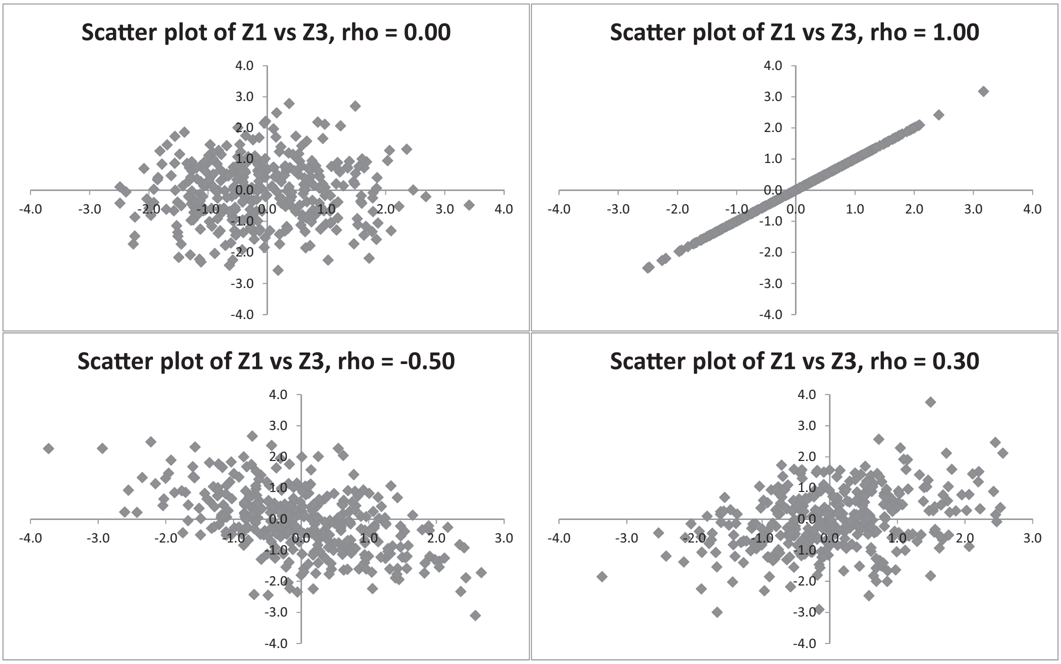
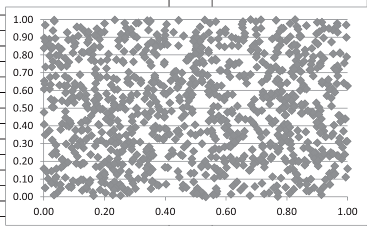


Figure 24.1
Scatter plots of random standard normal simulates with different correlations.

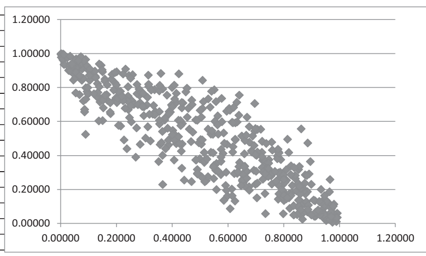
We show this routine twice. The first time we show it in an extensive spreadsheet below. Note the "diagonal aspect" of the scatter diagram of u_2 versus u_1 (this shows the correlation).

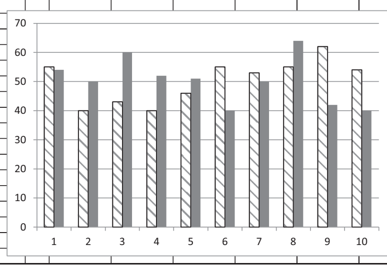
	A	B	C	D	E	F	G	H	I	J
1	**1,000 CORRELATED UNIFORM SIMULATED RANDOM NUMBERS Starting with Normal, then going to Uniform**									
2	Rho, ρ	0								
3	Mean	-0.0166	-0.0056	-0.0056	<-- =AVERAGE(D11:D10000)		Mean	0.4966	0.4975	<-- =AVERAGE(I11:I9998)
4	Sigma	1.0113	0.9758	0.9758	<-- =STDEVP(D11:D10000)		Sigma	0.2904	0.2839	<-- =STDEVP(I11:I9998)
5	Skewness	-0.0556	0.0681	0.0681	<-- =SKEW(D11:D10000)		Max	0.9999	0.9996	<-- =MAX(I11:I9998)
6	Kurtosis	0.2067	0.1383	0.1383	<-- =KURT(D11:D10000)		Min	0.0004	0.0005	<-- =MIN(I11:I9998)
7	Count	1,000	1,000	1,000	<-- =COUNT(D11:D10000)					
8	Corr(z_1,z_3)	0.0184	<-- =CORREL(B11:B10000,D11:D10000)				Corr(u_1,u_2)	0.0221	<-- =CORREL(H11:H1010,I11:I1010)	
9										
10		Z_1	Z_2	Z_3				U_1	U_2	
11	=NORMSINV(RAND())-->	-1.1794	0.4929	0.4929	<-- =rho*B11+SQRT(1-rho^2)*C11		=NORM.S.DIST(B11,1)-->	0.1191	0.6890	<-- =NORM.S.DIST(D11,1)
12		-0.6960	-0.7892	-0.7892				0.2432	0.2150	
13		0.7384	0.8904	0.8904				0.7699	0.8134	
14		-0.6938	-1.6462	-1.6462				0.2439	0.0499	
15		0.1559	-0.1719	-0.1719				0.5619	0.4318	
16		0.1059	0.9392	0.9392				0.5422	0.8262	
17		0.4912	-0.7760	-0.7760				0.6883	0.2189	
18		-0.9220	-0.7613	-0.7613				0.1783	0.2232	
19		-0.7405	0.4353	0.4353				0.2295	0.6683	
20		-0.2325	-0.9728	-0.9728				0.4081	0.1653	
21		-0.4111	2.5493	2.5493				0.3405	0.9946	
22		-0.9892	-1.9558	-1.9558				0.1613	0.0252	
23		1.0221	-0.0712	-0.0712				0.8466	0.4716	
24		-1.7364	0.8363	0.8363				0.0413	0.7985	
25		0.3628	-0.4396	-0.4396				0.6416	0.3301	
26		-0.7760	-0.3599	-0.3599				0.2189	0.3595	
27		0.4689	0.4987	0.4987				0.6804	0.6910	



We can make this process more efficient (and also more obscure) by combining functions as in the following spreadsheet. Column A cells are defined with **Rand**( ). Column B contains the formula **=NORM.S.DIST(Rho* NORM.S.INV(A7)+SQRT(1-Rho^2)*NORM.S.INV(RAND( )),1)**.

	A	B	C	D	E	F	G	H	I	J
1	**1,000 CORRELATED UNIFORM SIMULATED RANDOM NUMBERS More Efficient Procedure**									
2	Rho	-0.9								
3	Corr(u_1,u_2)	-0.89293								
4										
5										
6	U_1	U_2			Frequency Distribution					
7	0.81343	0.49546	<-- =NORM.S.DIST(Rho*NORM.S.INV(A7)+SQRT(1-Rho^2)*NORM.S.INV(RAND()),1)		Bins	U_1	U_2			
8	0.38688	0.46179			0.1	55	54			
9	0.53905	0.52559			0.2	40	50			
10	0.21526	0.57302			0.3	43	60			
11	0.35648	0.46330			0.4	40	52			
12	0.88493	0.18691			0.5	46	51			
13	0.82588	0.33920			0.6	55	40			
14	0.04539	0.84384			0.7	53	50			
15	0.13171	0.82252			0.8	55	64			
16	0.19458	0.82780			0.9	62	42			
17	0.82913	0.18255			1.0	54	40			
18	0.66841	0.30585								
19	0.06410	0.91164								
20	0.14516	0.93130								
21	0.28277	0.75389								
22	0.80075	0.13454								
23	0.78034	0.42222								
24	0.66430	0.30995								
25	0.60427	0.67723								
26	0.49552	0.47382								
27	0.43436	0.67985								
28	0.49882	0.38023								
29	0.38614	0.49012								
30	0.53584	0.68470								
31	0.06236	0.96573								
32	0.51508	0.35964								
33	0.96866	0.06450								
34	0.39951	0.80045								





24.7 What's Our Interest in Correlation? A Small Case[8]

Jacob has just retired at age 65 with savings of $1 million. He intends to invest 60% of this in a market index fund and the remaining 40% in a risk-free asset. He estimates the return on the risk-free asset as $r_f = 3\%$ annually, and he estimates that the market portfolio will have normally distributed returns with mean return $\mu = 11\%$ and $\sigma = 20\%$.

Jacob intends to withdraw $50,000 from his account at the beginning of this year. He thinks that this amount will, on average grow at 3% per year, with a standard deviation of 10%. Furthermore, he thinks that the growth rate of his annual spending will be correlated $\rho = 0.5$ with the stock market.[9]

Our question: If Jacob's life expectancy is age 90, how much will he leave to his beloved son, Simon? Here's a simulation that answers this question (some rows are hidden):

8. This section uses some materials from Chapter 26 and may be skipped on first reading.

9. His theory: When the market goes up, everyone spends more!

	A	B	C	D	E	F	G	H	I	J
1	**RETIREMENT PROBLEM Spending and Market are Correlated**									
2	Savings at 65	1,000,000								
3	Annual expenses									
4	Year 65	50,000								
5	Mean growth	3%								
6	Sigma of growth	10%								
7	Correlation with market	0.50					=$B$7*I17+(1-$B$7^2)^(0.5)*NORM.S.INV(RAND())			
8	Stock market									
9	Mean	11%								
10	Sigma	20%								
11	Risk-free	4%				=EXP($B$9+$B$10*I17)				
12	Invested in stock market	60%								
13	Left for heirs	906,029	<-- =B42							
14										
15	=EXP($B$5+$B$6*J18)		Spending		Returns				Two correlated standard normals	
16	Age	Savings at beg period	Growth over previous year	Spending	Stock market	Risk-free	Savings at end of period		Z_1	Z_2
17	65	1,000,000		50,000	1.0682	1.0408	1,004,359		-0.2203	-0.9133
18	66	1,004,359	0.9521	47,604	0.8125	1.0408	864,729		-1.5883	-0.7911
19	67	864,729	0.9953	47,381	1.0835	1.0408	871,664		-0.1488	-0.3470
20	68	871,664	1.0895	51,621	1.0418	1.0408	853,985		-0.3454	0.5573
21	69	853,985	1.1228	57,961	1.1690	1.0408	889,751		0.2309	0.8584
22	70	889,751	0.9806	56,836	0.7173	1.0408	705,228		-2.2114	-0.4961
23	71	705,228	1.1715	66,581	1.4352	1.0408	815,828		1.2564	1.2825
24	72	815,828	1.0569	70,371	1.2569	1.0408	872,528		0.5932	0.2537
25	73	872,528	0.9371	65,947	0.9205	1.0408	781,255		-0.9644	-0.9493
26	74	781,255	1.0533	69,463	1.1433	1.0408	784,598		0.1195	0.2194
27	75	784,598	0.9870	68,558	1.1017	1.0408	771,425		-0.0657	-0.4312
41	89	929,695	0.9389	65,230	1.0529	1.0408	906,029		-0.2921	-0.9299
42	90	906,029								

Some details of the simulation:

- Whatever savings are left at the end of a particular year constitute the initial savings for the next year.
- Z_1 and Z_2 are correlated standard normal variables. Note the formulas:

$$Z_1 = \textbf{Norm.S.Inv}(\textbf{Rand}(\ )),$$

$$Z_2 = \rho * Z_1 + \textbf{Sqrt}(\mathbf{1} - \rho^2) * \textbf{Norm.S.Inv}(\textbf{Rand}(\ ))$$

In this particular simulation, Jacob leaves $906,029 for his heirs. Using the technique of running a data table on a blank cell (see Chapter 31), we can run many simulations of this same problem:

	M	N	O
17	Simulation	906,029	<-- =B13, data table header
18	1	-356,743	
19	2	351,558	
20	3	1,886,149	
21	4	387,357	
22	5	1,683,477	
23	6	390,121	
24	7	281,914	
25	8	576,245	
26	9	7,151,210	
27	10	1,251,162	
28	11	845,362	
29	12	3,731,611	
30	13	-461,524	
31	14	4,928,934	
32	15	2,700,700	
33	16	368,321	
34	17	3,127,197	
35	18	3,417,390	
36	19	2,707,635	
37	20	421,627	
38			
39	Average	1,769,485	<-- =AVERAGE(N18:N37)
40	Sigma	1,959,905	<-- =STDEV(N18:N37)
41	Min	-461,524	<-- =MIN(N18:N37)
42	Max	7,151,210	<-- =MAX(N18:N37)
43	Negative	2	<-- =COUNTIF(N18:N37,"<0")

On average, Jacob's heirs should do pretty well. But in 2 out of the 20 simulations, his heirs will be left intestate. They might also use **Data Table** to check whether the correlation affects their inheritance.

	M	N	O	P	Q	R
15		=B13 data table header				
16				Correlation ↓		
17		-0.80	-0.4	0.00	0.4	0.80
18	1	-9,208,659	3,170,554	3,203,382	3,785,735	2,667,905
19	2	1,282,980	2,430,952	-353,824	2,204,197	2,726,465
20	3	7,963,118	-1,818,111	3,361,649	1,680,969	6,282,248
21	4	276,651	-2,947,014	14,337,774	1,784,993	6,879,205
22	5	-1,780,692	2,250,315	4,456,252	2,403,919	-913,402
23	6	3,067,479	-630,263	10,037,217	1,280,899	1,664,357
24	7	1,435,592	718,311	1,827,268	2,261,576	1,863,233
25	8	3,627,215	6,871,791	13,387,080	373,271	3,786,064
26	9	5,751,766	-1,122,584	1,320,789	498,684	619,038
27	10	4,014,473	8,050,127	2,878,871	1,748,100	5,051,896
28	11	439,089	4,053,251	2,173,128	7,381,844	13,445,269
29	12	2,562,847	-1,177,750	-1,562,287	5,884,251	2,545,434
30	13	5,725,661	-700,188	4,704,009	-1,681,796	3,438,369
31	14	-1,376,916	2,494,968	-3,210,949	2,090,863	1,769,926
32	15	2,557,209	802,532	3,262,467	5,176,845	646,516
33	16	-1,891,881	1,662,779	1,818,090	5,548,178	-260,280
34	17	3,120,033	5,377,046	2,878,251	4,211,705	9,040,059
35	18	-1,214,800	7,598,180	8,244,721	2,016,503	1,607,321
36	19	395,636	6,807,371	-272,614	4,129,666	4,852,653
37	20	14,077,754	3,165,395	12,316	3,868,468	1,012,387
38						
39	Average	2,041,228	2,352,883	3,625,180	2,832,444	3,436,233
40	Sigma	4,615,229	3,304,220	4,631,098	2,157,577	3,434,172
41	Min	-9,208,659	-2,947,014	-3,210,949	-1,681,796	-913,402
42	Max	14,077,754	8,050,127	14,337,774	7,381,844	13,445,269
43	Negative	5	6	4	1	2

As might be expected, in general the higher the negative correlation between the stock market and Jacob's expenses, the lower will be the inheritance. This makes sense: With negative correlation, when the stock market goes down, expenses go up (and vice versa).

24.8 Multiple Random Variables with Correlation: The Cholesky Decomposition

We can also create multiple correlated random simulates by using the Cholesky decomposition. A bit of background: A square matrix is called *positive definite* if, for any row vector x, the product $xSx^T > 0$. The variance-covariance matrix for security returns discussed in Chapters 8–12,

$$S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2N} \\ \vdots & & & \\ \sigma_{N1} & \sigma_{N2} & & \sigma_{NN} \end{bmatrix}$$

, is positive definite and symmetric (since $\sigma_{ij} = \sigma_{ji}$).

A square matrix is called *upper triangular* if it has non-zero entries only on the diagonal and elements above the diagonal. A square matrix is called *lower triangular* if (you can fill in the blanks).

The French mathematician André-Louis Cholesky (1875–1918) proved that any symmetric positive-definite matrix S can be written as the product of a lower-triangular matrix L and its transpose L^T. This is the *Cholesky decomposition*.

Example

In the example below cells B2:B5 contain a 4x4 variance-covariance matrix. Cells B8:D11 contain the Cholesky decomposition of this matrix—a lower-triangular matrix L. Cells B14:D17 multiply L times its transpose: As you can see, the result is to give back the original variance-covariance matrix.

	A	B	C	D	E
1	**Variance-covariance matrix, S**				
2	0.400	0.030	0.020	0.000	
3	0.030	0.200	0.000	-0.060	
4	0.020	0.000	0.300	0.030	
5	0.000	-0.060	0.030	0.100	
6					
7	**Cholesky decomposition, L**				
8	0.632	0.000	0.000	0.000	<-- {=cholesky(A2:D5)}
9	0.047	0.445	0.000	0.000	
10	0.032	-0.003	0.547	0.000	
11	0.000	-0.135	0.054	0.281	
12					
13	**Check: Multiply above matrix by its transpose**				
14	0.400	0.030	0.020	0.000	<-- {=MMULT(A8:D11,TRANSPOSE(A8:D11))}
15	0.030	0.200	0.000	-0.060	
16	0.020	0.000	0.300	0.030	
17	0.000	-0.060	0.030	0.100	

The function **Cholesky** can be found on the disk that accompanies *Financial Modeling*.[10]

Using the Cholesky Decomposition to Produce Correlated Normal Simulates

We start off with a varcov matrix $S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix}$. We want to run a simulation which in each iteration produces a vector $\begin{Bmatrix} x_{1t} \\ x_{2t} \\ x_{3t} \\ x_{4t} \end{Bmatrix}$ of four random numbers, so that these numbers have the following properties:

• The mean of each of the numbers is zero: $\frac{1}{n}\sum_{t=1}^{n} x_{it} \approx 0$. (Because of randomness, we can never demand exact equality to zero, only an approximation.)

• The variance of each series of numbers matches the varcov matrix:

$Var\{x_{11}, x_{12}, x_{13}, \ldots\} \approx \sigma_{11}$
$Var\{x_{21}, x_{22}, x_{23}, \ldots\} \approx \sigma_{22}$
…

• The covariance of any two of the series of numbers matches the covariance in the varcov matrix:

$Cov\{(x_{11}, x_{12}, x_{13}, \ldots), (x_{21}, x_{22}, x_{23}, \ldots)\} \approx \sigma_{21} = \sigma_{12}$
etc.

The way to do this is in two steps:

1. Produce a set of normally distributed numbers between 0 and 1.
2. Pre-multiply each vector of such numbers by the Cholesky decomposition *L*.

10. My thanks to Antoine Jacquier, who posted this on Wilmott.com and has allowed me to use it in this book.

In the illustration below we show one step of this simulation:

	A	B	C	D	E
1	BASIC MULTIVARIATE NORMAL SIMULATION				
2	Variance-covariance matrix				
3	0.40	0.03	0.02	0.01	
4	0.03	0.30	0.00	-0.06	
5	0.02	0.00	0.20	0.03	
6	0.01	-0.06	0.03	0.10	
7					
8	**Cholesky decomposition**				
9	0.6325	0.0000	0.0000	0.0000	<-- {=cholesky(varcov)}
10	0.0474	0.5457	0.0000	0.0000	
11	0.0316	-0.0027	0.4461	0.0000	
12	0.0158	-0.1113	0.0654	0.2882	
13					
14	**Generating four random normals**				
15	-1.3212	<-- =NORMSINV(RAND())			
16	-1.1972	<-- =NORMSINV(RAND())			
17	0.9280	<-- =NORMSINV(RAND())			
18	2.2751	<-- =NORMSINV(RAND())			
19					
20	**Generating multinomial normal output**				
21	-0.83560	<-- {=MMULT($A$9:$D$12,A15:A18)}			
22	-0.71594				
23	0.37547				
24	0.82886				

In cells A15:A18 we use **NormSInv** and **Rand** to produce four random numbers, each of which is normally distributed with mean zero and standard deviation 1.[11] In cells B21:B24 we pre-multiply this random vector by the Cholesky matrix in cells A9:D12. The claim is that the numbers in A21:A24 are normally distributed with mean zero and the variance-covariance structure given by the varcov matrix. We cannot, of course, prove this claim from a single simulation. In the following spreadsheet we replicate the above procedure 220 times:

11. For details of why this works, see Chapter 31.

	A	B	C	D	E	F
1	**BASIC MULTIVARIATE NORMAL SIMULATION**					
2	**Variance-covariance matrix**					
3	0.40	0.03	0.02	0.01		
4	0.03	0.30	0.00	-0.06		
5	0.02	0.00	0.30	0.03		
6	0.01	-0.06	0.03	0.10		
7						
8	**Cholesky decomposition**					
9	0.6325	0.0000	0.0000	0.0000	<-- {=cholesky(varcov)}	
10	0.0474	0.5457	0.0000	0.0000		
11	0.0316	-0.0027	0.5468	0.0000		
12	0.0158	-0.1113	0.0534	0.2907		
13						
14	**Generating four random normals**					
15	-1.1066	0.0427	0.5522	-0.1607	1.1747	-0.6570
16	1.3488	0.0973	-0.1789	-0.1380	-0.6901	-0.7721
17	0.6291	-0.6040	-0.1421	-0.7268	-1.6977	0.7685
18	0.6768	-2.2200	-0.7643	0.4689	-0.7651	-0.5160
19						
20	**Generating multinomial normal output**					
21	-0.69990	0.02701	0.34925	-0.10164	0.74294	-0.41553
22	0.68351	0.05513	-0.07144	-0.08293	-0.32083	-0.45250
23	0.30529	-0.32919	-0.05973	-0.40212	-0.88924	0.40158
24	0.06268	-0.68775	-0.20112	0.11033	-0.21766	-0.03338
25						
26	**Checking**					
27	Number of simulations	220	<-- =COUNT(15:15)			
28						
29	Mean1	0.05502	<-- =AVERAGE(21:21)			
30	Mean2	0.02662	<-- =AVERAGE(22:22)			
31	Mean3	0.07392	<-- =AVERAGE(23:23)			
32	Mean4	-0.02130	<-- =AVERAGE(24:24)			
33						
34		**From varcov matrix**	**From simulation**			
35	Var1	0.40	0.3658	<-- =VAR.P(21:21,21:21)		
36	Var2	0.30	0.2508	<-- =VAR.P(22:22,22:22)		
37	Var3	0.30	0.2997	<-- =VAR.P(23:23,23:23)		
38	Var4	0.10	0.0824	<-- =VAR.P(24:24,24:24)		
39						
40	Covar(1,2)	0.03	0.0318	<-- =COVARIANCE.P($21:$21,22:22)		
41	Covar(1,3)	0.02	0.0198	<-- =COVARIANCE.P($21:$21,23:23)		
42	Covar(1,4)	0.01	0.0158	<-- =COVARIANCE.P($21:$21,24:24)		
43						
44	Covar(2,3)	0.00	0.0095	<-- =COVARIANCE.P($22:$22,23:23)		
45	Covar(2,4)	-0.06	-0.0419	<-- =COVARIANCE.P($22:$22,24:24)		
46						
47	Covar(3,4)	0.03	0.0268	<-- =COVARIANCE.P(23:23,24:24)		

In rows 29–47 we check the mean, variance, and covariance of each of the variables, comparing them to zero (in the case of the means) and to the data in the varcov matrix. By pressing **F9**, you can repeat the simulations and convince yourself that we have indeed produced a set of multivariate normal simulations corresponding to the target variance-covariance structure.

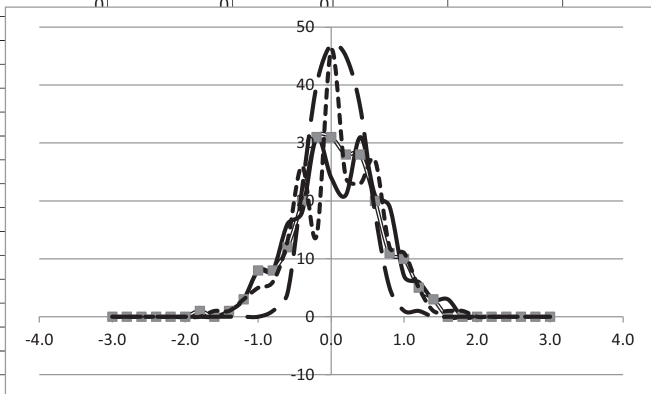
Here, for example, is the result of another press of **F9**:

	A	B	C	D	E	F
26	**Checking**					
27	Number of simulations	220	<-- =COUNT(15:15)			
28						
29	Mean1	0.05387	<-- =AVERAGE(21:21)			
30	Mean2	0.02040	<-- =AVERAGE(22:22)			
31	Mean3	-0.05087	<-- =AVERAGE(23:23)			
32	Mean4	-0.03520	<-- =AVERAGE(24:24)			
33						
34		**From varcov matrix**	**From simulation**			
35	Variable1	0.40	0.3767	<-- =VAR.P(21:21,21:21)		
36	Variable2	0.30	0.2587	<-- =VAR.P(22:22,22:22)		
37	Variable3	0.30	0.2655	<-- =VAR.P(23:23,23:23)		
38	Variable4	0.10	0.1131	<-- =VAR.P(24:24,24:24)		
39						
40	Covar(1,2)	0.03	0.0107	<-- =COVARIANCE.P($21:$21,22:22)		
41	Covar(1,3)	0.02	0.0016	<-- =COVARIANCE.P($21:$21,23:23)		
42	Covar(1,4)	0.01	-0.0106	<-- =COVARIANCE.P($21:$21,24:24)		
43						
44	Covar(2,3)	0.00	0.0029	<-- =COVARIANCE.P($22:$22,23:23)		
45	Covar(2,4)	-0.06	-0.0692	<-- =COVARIANCE.P($22:$22,24:24)		
46						
47	Covar(3,4)	0.03	0.0391	<-- =COVARIANCE.P(23:23,24:24)		

Do You Need Further Convincing?

Another way to convince yourself that we have done something sensible is to plot the frequency distribution of each of the variables, using the Excel **Frequency** function:

	A	B	C	D	E	F	G	H
50		Frequency distributions						
51		Variable1	Variable2	Variable3	Variable4			
52	-3.0	0	0	0	0	<-- {=FREQUENCY(24:24,$A$52:$A$82)}		
53	-2.8	0	0	0	0			
54	-2.6	0	0	0	0			
55	-2.4	0	0	0	0			
56	-2.2	0						
57	-2.0	0						
58	-1.8	0						
59	-1.6	0						
60	-1.4	1						
61	-1.2	3						
62	-1.0	8						
63	-0.8	8						
64	-0.6	16						
65	-0.4	18						
66	-0.2	31						
67	0.0	24						
68	0.2	21						
69	0.4	31						
70	0.6	21						
71	0.8	19						
72	1.0	7	10	11	1			
73	1.2	6	5	5	1			
74	1.4	3	3	1	0			
75	1.6	3	0	1	0			
76	1.8	0	0	1	0			
77	2.0	0	0	0	0			
78	2.2	0	0	0	0			
79	2.4	0	0	0	0			
80	2.6	0	0	0	0			
81	2.8	0	0	0	0			
82	3.0	0	0	0	0			



Note: Simulations are tricky! Press **F9** multiple times to get different frequencies and statistics (see Figure 24.2).

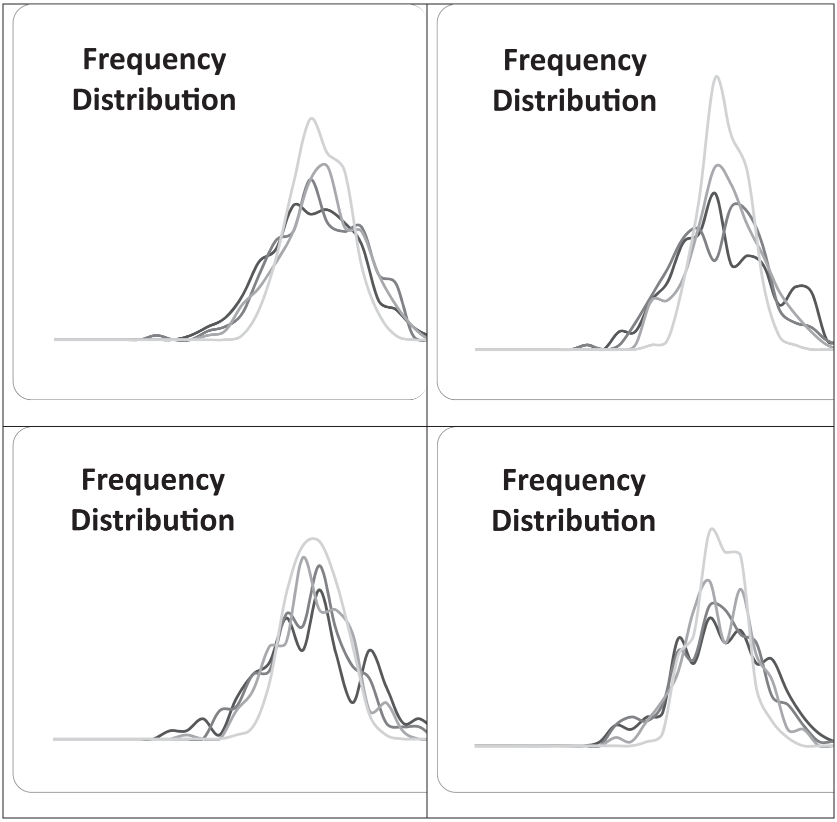


Figure 24.2
Four frequency distributions of random normals using the Cholesky decomposition. They all look "approximately" normal, but they don't correspond to the "textbook" shapes that we're used to seeing. That's life in the (simulated) real world!

24.9 Multivariate Normal with Non-Zero Means

Below we illustrate the case where the asset returns (given in F3:F6) are non-zero. To create multivariate simulations with the desired returns, we simply add the target means to the multivariate simulations in rows 21–24:

	A	B	C	D	E	F	G	H	I
1	**MULTIVARIATE SIMULATION WITH NON-ZERO MEANS**								
2	**Variance-covariance matrix**					**Mean returns**			
3	0.40	0.03	0.02	0.01		5%			
4	0.03	0.30	0.00	-0.06		6%			
5	0.02	0.00	0.20	0.03		7%			
6	0.01	-0.06	0.03	0.10		8%			
7									
8	**Cholesky decomposition**						Cells B21:B25 contain the formula =NORMSINV(RAND())		
9	0.6325	0.0000	0.0000	0.0000	<-- {=cholesky(varcov)}				
10	0.0474	0.5457	0.0000	0.0000					
11	0.0316	-0.0027	0.4461	0.0000			Cells B21:B25 contain the formula {=MMULT($A$9:$D$12,B15:B18)+$F$3:$F$6}		
12	0.0158	-0.1113	0.0654	0.2882					
13									
14	**Generating four standard random normals**								
15	0.5556	1.1031	0.3987	-0.3293	1.6139	0.2647	-0.2145	-0.5983	-0.2059
16	-0.7991	0.3031	0.2735	-1.1220	0.7048	1.2862	0.7759	0.0811	-0.5260
17	0.1769	0.4273	-1.5465	0.9186	-0.3287	-1.0359	0.0249	1.3598	0.2827
18	-0.0305	-0.7747	0.5205	-0.8122	-1.1149	-0.9038	-0.7232	-1.7548	-0.1096
19									
20	**Generating multinomial normal output with desired means**								
21	0.40140	0.74765	0.30216	-0.15825	1.07073	0.21740	-0.08569	-0.32840	-0.08025
22	-0.34970	0.27770	0.22813	-0.56783	0.52115	0.77440	0.47319	0.07586	-0.23677
23	0.16867	0.29466	-0.60802	0.47245	-0.02755	-0.38728	0.07221	0.65745	0.19103
24	0.18055	-0.13162	0.10467	0.02572	-0.31581	-0.38731	-0.21659	-0.35528	0.12222

To confirm that this works, we compute the statistics of the simulation:

	A	B	C	D	E
26	**Checking**				
27	Number of simulations	220	<-- =COUNT(15:15)		
28					
29					
30		**Target mean**	**Average of simulation**		
31	Mean1	0.05	0.0093	<-- =AVERAGE(21:21)	
32	Mean2	0.06	0.0512	<-- =AVERAGE(22:22)	
33	Mean3	0.07	0.0618	<-- =AVERAGE(23:23)	
34	Mean4	0.08	0.0884	<-- =AVERAGE(24:24)	
35					
36		**From varcov matrix**	**From simulation**		
37	Var1	0.40	0.3502	<-- =VARP(21:21,21:21)	
38	Var2	0.30	0.2641	<-- =VARP(22:22,22:22)	
39	Var3	0.20	0.2205	<-- =VARP(23:23,23:23)	
40	Var4	0.10	0.0897	<-- =VARP(24:24,24:24)	
41					
42	Covar(1,2)	0.03	0.0096	<-- =COVAR(21:21,22:22)	
43	Covar(1,3)	0.02	0.0076	<-- =COVAR(21:21,23:23)	
44	Covar(1,4)	0.01	-0.0002	<-- =COVAR(21:21,24:24)	
45					
46	Covar(2,3)	0.00	0.0151	<-- =COVAR(22:22,23:23)	
47	Covar(2,4)	-0.06	-0.0502	<-- =COVAR(22:22,24:24)	
48					
49	Covar(3,4)	0.03	0.0332	<-- =COVAR(23:23,24:24)	

The average (cells C31:C34) can be depressingly far from the desired targets! It takes *a lot* of simulated data to get close to the targets.[12]

12. In the sense of "hitting" targets—it is much more difficult to hit the means than to hit the variances. This sentence has an almost metaphysical meaning: When we compute asset means from historical returns—even if all the returns are drawn from a stationary distribution (i.e., with unchanging means, variances, and covariances), we are very unlikely in a small sample to get close to the actual mean of the distribution. On the other hand, we are much more likely to be close to the variances. The covariances are also very difficult to hit.

24.10 Multivariate Uniform Simulations

Once we have simulated correlated normal distributions, we can easily simulate uniform distributions by using **Norm.S.Dist**. In the example below, we first create (in rows 15–18) four series of correlated standard normal deviates.

	A	B	C	D	E	F	G	H
1	**BASIC MULTIVARIATE UNIFORM SIMULATION**							
2	**Variance-covariance matrix**							
3	1.00	0.03	0.02	0.01				
4	0.03	1.00	0.00	-0.06				
5	0.02	0.00	1.00	0.03				
6	0.01	-0.06	0.03	1.00				
7								
8	**Cholesky decomposition**							
9	1.0000	0.0000	0.0000	0.0000	<-- {=cholesky(varcov)}			
10	0.0300	0.9995	0.0000	0.0000				
11	0.0200	-0.0006	0.9998	0.0000				
12	0.0100	-0.0603	0.0298	0.9977				
13								
14	**Generating multinomial normal output**							
15	-0.95450	0.52393	-0.01498	0.45889	0.47207	-0.78706	0.81113	1.13469
16	0.08893	-2.06744	-0.36949	-0.33538	-1.35352	-0.12097	1.10413	1.46068
17	0.15682	1.89231	-1.66563	0.28293	-0.72093	-0.21203	1.04269	0.00869
18	-0.06897	1.13988	0.16480	-0.21954	-0.95462	1.18325	0.61655	-1.82526
19								
20	How many?	390	<-- =COUNT(15:15)					
21								
22	Mean		Variance	Theoretical	Skewness	Kurtosis		
23	-0.1087	<--	0.9225	1.0000	0.3156	0.0319	<-- =KURT(15:15)	
24	-0.0892	=AVERAGE(15:1	0.9923	1.0000	0.1090	0.2091	<-- =KURT(16:16)	
25	-0.0217	5)	1.0732	1.0000	0.0607	0.7864		
26	0.0015		0.9684	1.0000	-0.0014	-0.4970		

We now generate uniform variates by applying **Norm.S.Dist** to the normal variates as shown below:

	A	B	C	D	E	F	G	H
14	**Generating multinomial normal output**							
15	1.60155	0.35557	1.03640	0.00638	0.26480	-0.87386	0.07279	0.68435
16	-0.60257	1.26064	0.66585	-0.75586	-1.16279	-0.38719	0.75299	-1.76904
17	-0.41759	-2.21796	0.16184	0.02274	0.57614	-0.67249	-0.05777	0.35109
18	0.14609	-0.29084	-1.33622	0.69459	1.61119	0.83252	-0.05966	1.73705
19								
20	How many?	390	<-- =COUNT(15:15)					
21								
22	Mean		Variance	Theoretical	Skew	Kurt		
23	0.0239	<--	1.0822	1.0000	0.1665	-0.0117	<-- =KURT(15:15)	
24	-0.0782	=AVERAGE(15:1	0.9986	1.0000	-0.0257	-0.1524	<-- =KURT(16:16)	
25	-0.0075	5)	0.8513	1.0000	-0.1257	0.0361		
26	0.0195		0.9418	1.0000	-0.0408	0.1898		
27								
28	**Generating multinomial uniform output**				=NORM.S.DIST(D15,1)			
29	0.9454	0.6389	0.8500	0.5025	0.6044	0.1911	0.5290	0.7531
30	0.2734	0.8963	0.7472	0.2249	0.1225	0.3493	0.7743	0.0384
31	0.3381	0.0133	0.5643	0.5091	0.7177	0.2506	0.4770	0.6372
32	0.5581	0.3856	0.0907	0.7563	0.9464	0.7974	0.4762	0.9588

The statistics for the simulated uniform variates are given below.[13]

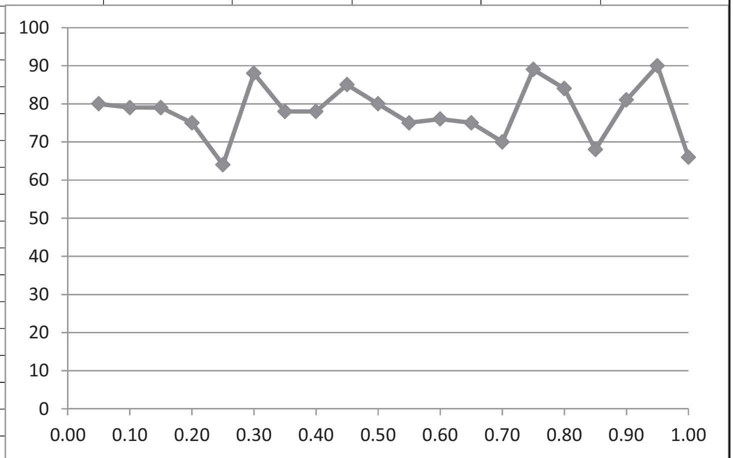
13. Recall that the theoretical variance of a (0,1) uniform variable is 1/12 = 0.0833.

	A	B	C	D	E
28	**Generating multinomial uniform output**				=NORM.S.DIST(D15,1)
29	0.0977	0.7258	0.2044	0.2876	0.8485
30	0.7253	0.5784	0.6019	0.8022	0.3373
31	0.0684	0.1728	0.1179	0.9484	0.2947
32	0.7104	0.3940	0.8212	0.8544	0.6659
33					
34		**Sample stats rows 29-32**	**Theoretical**		
35	Mean	0.4992	0.5000		
36	Variance	0.0830	0.0833		
37	Sigma	0.2880	0.2887		
38	Max	0.9998			
39	Min	0.0000			
40	Corr(1,2)	0.0650	0.0300		
41	Corr(1,3)	0.0908	0.0200		
42	Corr(1,4)	-0.0015	0.0100		
43	Corr(2,3)	-0.0593	0.0000		
44	Corr(2,4)	-0.0656	-0.0600		
45	Corr(3,4)	-0.0151	0.0300		
46					
47	Sample covariance matrix				
48	0.0766	0.0053	0.0073	-0.0001	
49	0.0053	0.0857	-0.0051	-0.0056	
50	0.0073	-0.0051	0.0848	-0.0013	
51	-0.0001	-0.0056	-0.0013	0.0842	

Note that without very large samples it is extremely difficult to match covariances. But a number of hits on **F9** might convince you that we have indeed matched the desired correlation structure.

Finally, a very simple test: what's the frequency distribution of rows 29–32?

	F	G	H	I	J	K	L	M
35	Frequency							
36	Bin							
37	0.05	80						
38	0.10	79						
39	0.15	79						
40	0.20	75						
41	0.25	64						
42	0.30	88						
43	0.35	78						
44	0.40	78						
45	0.45	85						
46	0.50	80						
47	0.55	75						
48	0.60	76						
49	0.65	75						
50	0.70	70						
51	0.75	89						
52	0.80	84						
53	0.85	68						
54	0.90	81						
55	0.95	90						
56	1.00	66						



24.11 Summary

Random numbers are widely used in financial engineering, especially in option pricing. This chapter has introduced you to the Excel and VBA random-number generators and has shown a number of techniques for producing normally distributed random numbers.

Exercises

1. Here is a random-number generator you can make yourself:

   - Start with some number, *Seed*.
   - Let $X_1 = Seed + \pi$. Let $X_2 = e^{5+\ln(X_1)}$.
   - The first random number is $Random = X_2 - Integer(X_2)$, where $Integer(X_2)$ is the integer part of X_2.
   - Repeat the process, letting $Seed = Random$.

   Run 1,000 of these random numbers and use **Frequency** to produce a frequency distribution in the intervals 0, 0.1, 0.2,…,1.

2. Write a VBA **Exercise1(seed)** that produces a random number based on a **Seed** and the rule of the previous exercise.

3. Define *AmodB* as the *remainder* when *A* is divided by *B*. For example 36mod25 = 11. Excel has this function; it is written **Mod(A,B)**. Now here is another random-number generator:

   - Let X_0 = **seed**.
   - Let $X_{n+1} = (7*X_n)\text{mod}10^8$.
   - Let $U_{n+1} = X_{n+1}/10^8$.

   The list of numbers U_1, U_2, … are the pseudo-random numbers generated by this random-number generator. (This is one of the many uniform random-number generators given in Abramowitz and Stegun, 1972).

4. Many states have daily lotteries, which are played as follows: Sometime during the day, you buy a lottery ticket, on which the seller inscribes a number you choose, between 000 and 999. That night there is a drawing on television in which a three-digit number is drawn. If the number on your ticket matches the number drawn, you win and collect $500. If you lose, you get nothing.

   a. Write an Excel function which produces a random number between 000 and 999. (Hint: Use **Rand( )** and **Int**( ).)
   b. Assume that you bet $1 every day of the year on the same number. Show your cumulative winnings throughout the year.

5. Tareq and Jamillah are playing with a single die for money. According to the rules of their game, Tareq pays Jamillah $0.50 at the start of every round, before they throw a die. They then throw the die; if it falls on an even number, Jamillah pays that amount in dollars to Tareq, and if it falls on an odd number, Tareq pays that amount to Jamillah.

   - Simulate Jamillah's cumulative winnings after 25 rounds of the game.
   - Simulate 50 games, and graph the results

6. Stock price simulation: A stock's price is lognormally distributed with mean $\mu = 15\%$. The current stock price is $S_0 = 35$. Following the template on the spreadsheet, create 60 *static* standard normal deviates using **Data|Data Analysis|Random Number Generation**. Use these random numbers to simulate the stock price path over 60 months. Create price paths for $\sigma = 15\%$, 30%, and 60% and graph these three paths on the same axes.

7. Stock price simulation: A stock's price is lognormally distributed with mean $\mu = 15\%$ and $\sigma = 50\%$. The current stock price is $S_0 = 35$. Following the template on the spreadsheet, create 60 *dynamic* standard normal deviates using **Norm.S.Inv(Rand( ))**. Use these random numbers to simulate the stock price path over 60 months and graph.

8. Marcus is 25 years old. He has a new job and intends to save $10,000 today and in each of the next 34 years (35 deposits altogether). He has decided on an investment policy in which he invests 30% of his assets in a risk-free bond with 3% continuously compounded annual interest and the remainder in the market portfolio that has lognormal returns $\mu = 12\%$ and $\sigma = 35\%$. Write a spreadsheet showing Marcus's accumulation by the time he is 60. A sample output is given below:

	A	B	C	D	E
1	**MARCUS'S INVESTMENT/SAVINGS DECISION**				
2	Annual deposit	10,000			
3	Risk-free rate	3%			
4	Market portfolio mean	12%			
5	Market portfolio sigma	35%			
6	Proportion in market portfolio	70%			
7	Accumulation at age 60	12,048,869	<-- =B45		
8					
9	**Age**	**Total investment at beginning of period**	**New investment**	**Total investment at end of period**	
10	25	0	10,000	13,065.44	<--
11	26	13,065.44	10,000	27,795.31	=(B10+C10)*($B$6*EXP($B$4+
12	27	27,795.31	10,000	52,906.64	$B$5*NORM.S.INV(RAND()))+(1-
13	28	52,906.64	10,000	58,590.40	$B$6)*EXP($B$3))
14	29	58,590.40	10,000	99,352.45	

9. Marcus decides that he needs at least \$2 million by the time he hits 60.

   - Run 100 simulations in order to determine the approximate probability of achieving this goal.
   - Compute the average and standard deviation of the terminal wealth.
   - Create a **Data Table** to determine the relation between the proportion invested in the risky asset and the probability of achieving the minimum at 60. Set the proportions in the risky asset to 0%, 10%, … , 100%.

10. Martha is playing a coin-toss game in which she tosses two coins. The probability of heads on the second coin is correlated with correlation $\rho = 0.6$ with the probability of heads on the first coin. In this particular game, Martha wins \$1 for each heads that she tosses.

   - Model one round of 2 coin tosses.
   - If she plays 10 rounds of 2 coin tosses each, how much will she win?
   - Use **Data Table** on a blank cell to model 25 cycles of this 10-round game.

25 An Introduction to Monte Carlo Methods

25.1 Overview

"Monte Carlo" (MC) methods refer to a variety of random simulations used to determine the values of parameters. In this introductory chapter to MC methods, we use MC to determine the value of π. In subsequent chapters we use MC to gain insight into investment and option strategies. The Monte Carlo method has its source in physics, where it is often used to determine model values for which there is no analytical solution.[1] One use of Monte Carlo in finance is similar: Monte Carlo methods use simulation to price assets whose prices are not readily determined by analytical means. In short: If there isn't a formula for computing the value of an asset, maybe we can determine its value with a simulation.

In Chapters 27–30 we also use Monte Carlo methods to give a feel for the uncertainty surrounding various investment and option strategies. When we do this, we are not necessarily interested in the pricing implications of uncertainty—we want to illustrate what kinds of results can occur from a given investment strategy in assets whose returns are uncertain.

In this chapter we give a layperson's introduction to Monte Carlo pricing.[2] We assume that you have read Chapter 24, which gives an introduction to random numbers.

25.2 Computing π Using Monte Carlo

All Monte Carlo methods involve random simulation. We illustrate how to use Monte Carlo to calculate the value of π, a number with which you are presumably familiar.

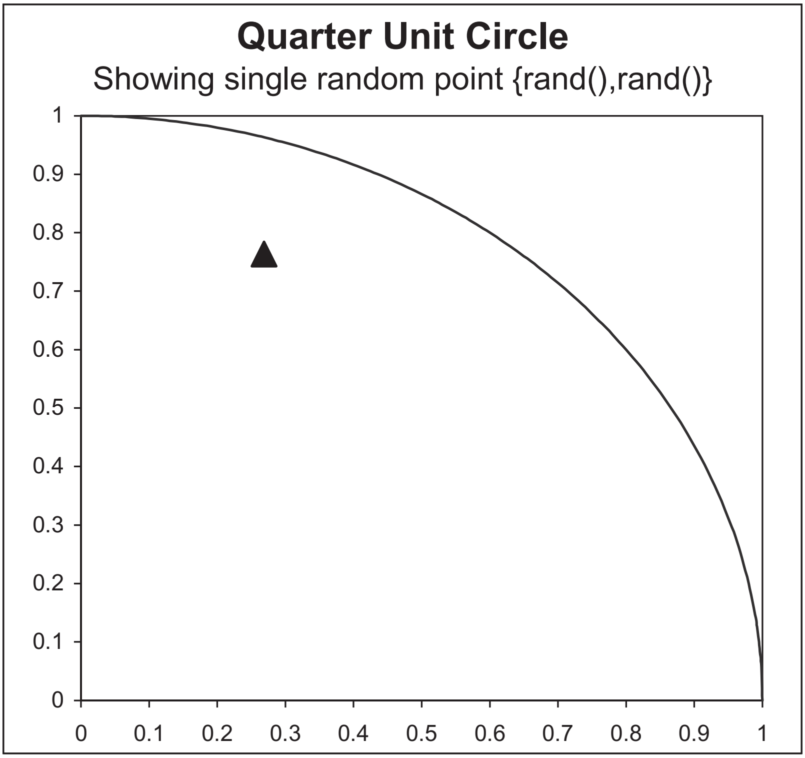
Here's our method: We know that the area of the unit circle (circle with radius 1) is π. It follows that the area of a quarter circle is $\pi/4$. We inscribe a quarter circle into a unit square, as illustrated below. We then proceed to "shoot" random points at the unit square. Each random point has an x component and a y component. We generate such points by using the Excel function **Rand**.

---

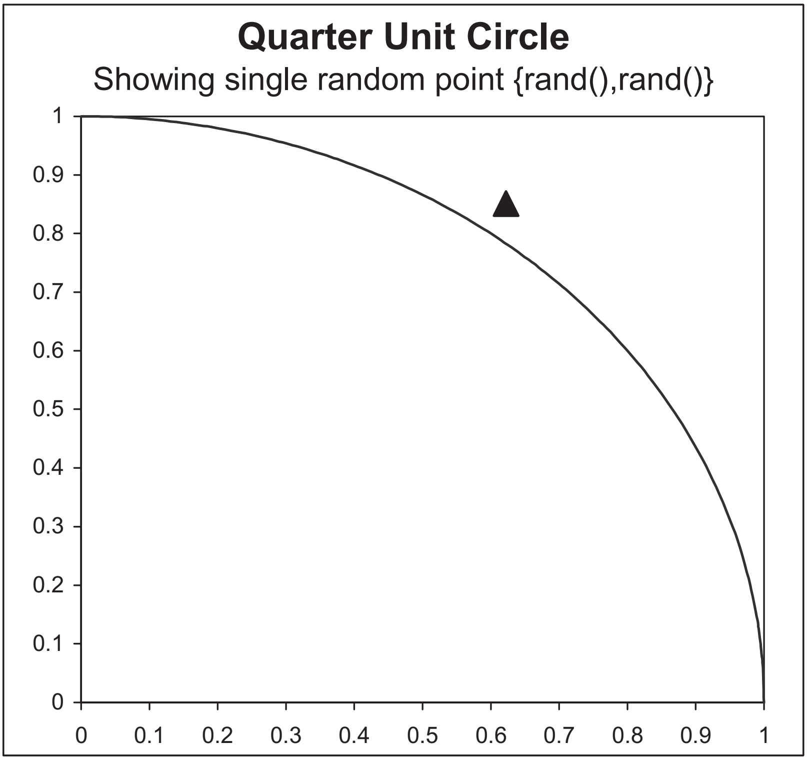
1. Two good websites with introductions to Monte Carlo methods are www.ornl.gov/~pk7/thesis/pkthnode19.html and www.puc-rio.br/marco.ind/monte-carlo.html.

2. An excellent nonintroductory text is Glasserman (2005).

The picture below shows the quarter circle inscribed in the unit square and a single random point, which happened to land inside the circle.



By pressing **F9** in the spreadsheet that accompanies this chapter, you can generate different values of the random point. In some cases, the point will be outside the unit circle:



We can easily do a calculation of the probability that the point will be inside the circle:

- The area of the whole unit circle is $\pi * r^2 = \pi$. Thus the area of the quarter unit circle is $\pi/4$.
- The random point—generated by **{Rand( ), Rand( )}**—is always inside the unit square, whose area is 1.
- Thus the probability of the random point being inside the unit circle is

$$\frac{Area\ unit\ circle}{Area\ unit\ square} = \frac{\pi / 4}{1} = \pi / 4$$

Monte Carlo Computation of π

If we count the relative number of points which fall inside the unit circle, we should approximate $\pi/4$. Thus

$$\begin{aligned} &\textit{Monte Carlo approximation of } \pi \\ &\quad = 4 * \textit{Relative number points inside unit circle} \\ &\quad = 4 * \frac{\textit{Number points inside unit circle}}{\textit{Total number of points}} \end{aligned}$$

In the spreadsheet below we generate a list of random numbers (columns B and C) and then use a Boolean function to test whether the numbers are in or out of the unit circle. In row 8 below, for example, the function **=(B8^2+C8^2<=1)** returns **TRUE** if the sum of the squares of B8 and C8 are less than or equal to 1 and 0 otherwise. In cell B2 we use **Count** to count the number of total points generated, and **CountIf** to determine the number of data points which fall inside the unit circle.[3]

3. All of the functions in this paragraph—Boolean functions, **Count**, and **CountIf**—are discussed in Chapter 33.

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	30	<-- =COUNT(A:A)		
3	Inside circle	22	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	2.933333333	<-- =B3/B2*4		
5					
6		Each cell in these columns contains the Excel function **=Rand()**			
7	**Experiment**	**Random1**	**Random2**	**In unit circle?**	
8	1	0.93377	0.14390	TRUE	<-- =(B8^2+C8^2<=1)
9	2	0.28866	0.68112	TRUE	
10	3	0.53592	0.60165	TRUE	
11	4	0.38665	0.27952	TRUE	
12	5	0.02396	0.62680	TRUE	
13	6	0.98093	0.02753	TRUE	
14	7	0.57770	0.34427	TRUE	
15	8	0.58922	0.26317	TRUE	
16	9	0.34752	0.77355	TRUE	
17	10	0.96858	0.08618	TRUE	
18	11	0.32394	0.66268	TRUE	
19	12	0.04574	0.69957	TRUE	
20	13	0.53216	0.99862	FALSE	
21	14	0.88861	0.60825	FALSE	
22	15	0.65561	0.39199	TRUE	
23	16	0.79978	0.21315	TRUE	
24	17	0.65605	0.99905	FALSE	
25	18	0.23636	0.95164	TRUE	
26	19	0.75541	0.17700	TRUE	
27	20	0.29197	0.01494	TRUE	
28	21	0.56781	0.40796	TRUE	
29	22	0.25891	0.04545	TRUE	
30	23	0.61585	0.88576	FALSE	
31	24	0.32830	0.43645	TRUE	
32	25	0.52969	0.47640	TRUE	
33	26	0.60362	0.85435	FALSE	
34	27	0.97880	0.20737	FALSE	
35	28	0.62883	0.84034	FALSE	
36	29	0.13787	0.66466	TRUE	
37	30	0.98497	0.36433	FALSE	

Each time we press **F9** to recalculate the spreadsheet, we get different values for the **=Rand**, and hence a different Monte Carlo value for π. Here are some examples:

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	30	<-- =COUNT(A:A)		
3	Inside circle	24	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	3.2	<-- =B3/B2*4		
5					
6		Each cell in these columns contains the Excel function **=Rand()**			
7	**Experiment**	**Random1**	**Random2**	**In unit circle?**	
8	1	0.19891	0.21584	TRUE	<-- =(B8^2+C8^2<=1)
9	2	0.60485	0.39918	TRUE	
10	3	0.54090	0.50887	TRUE	
11	4	0.01367	0.82935	TRUE	
12	5	0.29872	0.02460	TRUE	
13	6	0.98014	0.75186	FALSE	
14	7	0.01627	0.34278	TRUE	

Pressing **F9** again:

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS INITIAL EXPERIMENT**				
2	Number of data points	30	<-- =COUNT(A:A)		
3	Inside circle	25	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	3.333333333	<-- =B3/B2*4		
5					
6		Each cell in these columns contains the Excel function **=Rand()**			
7	**Experiment**	**Random1**	**Random2**	**In unit circle?**	
8	1	0.95610	0.35792	FALSE	<-- =(B8^2+C8^2<=1)
9	2	0.65874	0.62250	TRUE	
10	3	0.67164	0.23791	TRUE	
11	4	0.81134	0.46254	TRUE	
12	5	0.06633	0.01178	TRUE	
13	6	0.70038	0.91783	FALSE	
14	7	0.22334	0.19309	TRUE	

Now it's clear that the values we get for π are experimental, but if we did this for a lot of points we would get closer to the actual value of π. In the example below we run our experiment 65,000 times:

	A	B	C	D	E
1	**COMPUTING PI USING MONTE CARLO METHODS 65000 Iterations**				
2	Number of data points	65,000	<-- =COUNT(A:A)		
3	Inside circle	51,071	<-- =COUNTIF(D:D,TRUE)		
4	Pi?	3.142830769	<-- =B3/B2*4		
5					
6	**Experiment**	**Random1**	**Random2**	**In unit circle?**	
7	1	0.03174	0.88023	TRUE	<-- =(B7^2+C7^2<=1)
8	2	0.87070	0.97223	FALSE	
9	3	0.14515	0.65038	TRUE	
10	4	0.39535	0.31017	TRUE	

Pressing **F9** still gives much more accurate values for π. A few more presses of **F9** produced the following Monte Carlo values for π :

3.150544034, 3.144256741, 3.139556532,
3.149872576, 3.138213615, 3.132780906

Using more data points makes our MC value for π more accurate, though none of these values is that close to the actual value of π.[4]

25.3 Writing a VBA Program

This Monte Carlo business requires some VBA. Here's a program:

4. The actual value of π, correct to 50 digits, is 3.1415926535897932384626433832795028841971693993751. One of the end-of-chapter problems shows you a quick way of computing this value using several remarkable functions due to the Indian mathematician Srinivasa Ramanujan (1887–1920).

```
Sub MonteCarlo()
   n = Range("Number")
   Hits = 0
   For Index = 1 To n
      If Rnd ^ 2 + Rnd ^ 2 < 1 _
      Then Hits = Hits + 1
   Next Index
   Range("Estimate") = 4 * Hits / n
End Sub
```

The spreadsheet below shows this program and two other VBA programs. The program **MonteCarloTimer** computes both the **StartTime** and the **StopTime**, so that we can compute the elapsed time for the computations. Below, you can see that 10 million iterations of the program took 4 seconds on the author's Lenovo T420s laptop.

The program **MonteCarloTimeRecord** records each iteration of the program on the screen. This VBA routine allows you to see how the values of π in cell B3 develop. You can stop this or any VBA routine in midstream by pressing [Ctrl] + [Break]. This macro is incredibly time wasteful. It took us 103 seconds to run 5,000 iterations of the routine (compare this to running the 20 million iterations without the screen updating).

	A	B	C	D	E
1	**COMPUTING PI USING VBA**				
2	Number of data points	10,000,000	<-- This cell called "Number"		**Sub MonteCarlo()**
3	Pi?	3.1414508	<-- This cell called "Estimate"		n = Range("Number")
4					Hits = 0
5					For Index = 1 To n
6	StartTime	10:59:49	<-- This cell called "StartTime"		If Rnd ^ 2 + Rnd ^ 2 < 1 Then Hits = Hits + 1
7	StopTime	10:59:53	<-- This cell called "StopTime"		Next Index
8	Elapsed	0:00:04	<-- =Stoptime-StartTime		Range("Estimate") = 4 * Hits / n
9					**End Sub**
10					
11	**Note**				**Sub MonteCarloTime()**
12	[Ctrl]+a runs the macro "MonteCarlo"				'Includes timer
13	[Ctrl]+e runs the macro "MonteCarloTime" which also records the time				n = Range("Number")
14	[Ctrl]+f runs the macro "MonteCarloTimeRecord" which records the results as they are generated. For large number of points, this takes a *very long* time! Press **[Ctrl]+[Break]** to stop the macro.				Range("StartTime") = Time
15					
16					n = Range("Number")
17					Hits = 0
18					For Index = 1 To n
19					If Rnd ^ 2 + Rnd ^ 2 < 1 Then Hits = Hits + 1
20					Next Index
21					Range("Estimate") = 4 * Hits / n
22					Range("StopTime") = Time
23					**End Sub**
24					
25					**Sub MonteCarloTimeRecord()**
26					'Records everything (takes a long time)
27					n = Range("Number")
28					Range("StartTime") = Time
29					n = Range("Number")
30					Hits = 0
31					For Index = 1 To n
32					Range("Number") = Index
33					If Rnd ^ 2 + Rnd ^ 2 < 1 Then Hits = Hits + 1
34					Range("Estimate") = 4 * Hits / Index
35					Range("StopTime") = Time
36					Next Index
37					**End Sub**

25.4 Another Monte Carlo Problem: Investment and Retirement[5]

The problem: You are 65 years old and you have $1,000,000. You are trying to decide on a mix of investments: There is a riskless bond with an annual return of 6% and a risky stock portfolio with an expected log return of 12% and a standard deviation of return of 30%. Your limitations: You want to take $150,000 out of the account every year and have something left over at age 75.

To get a better handle on this situation, you plot out a spreadsheet:

5. The material in the remainder of the chapter anticipates some results of the next several chapters and can be skipped on first reading.

	A	B	C	D	E	F	G	H	I
1	**PLANNING YOUR RETIREMENT**								
2	Current wealth	1,000,000							
3	Risk-free rate	6%							
4	Parameters of risky investment								
5	Expected annual return	8%							
6	Standard deviation of return	20%							
7	Proportion invested in risky	70%							
8	Annual drawdown	150,000							
9									
10	**Year**	**Wealth at beginning of year**	**Invested in risky**	**Invested in bonds**	**Random number, normally distributed**	**1+return on risky investment**	**Wealth at end of year**	**Drawdown**	**Left at end of 10 years**
11	1	1,000,000	700,000	300,000	-0.6530	0.9507	984,011	150,000	
12	2	834,011	583,808	250,203	1.9864	1.6117	1,206,595	150,000	
13	3	1,056,595	739,616	316,978	-0.4627	0.9875	1,066,981	150,000	
14	4	916,981	641,887	275,094	0.3292	1.1570	1,034,774	150,000	
15	5	884,774	619,342	265,432	-0.1634	1.0485	931,203	150,000	
16	6	781,203	546,842	234,361	-0.0306	1.0767	837,625	150,000	
17	7	687,625	481,338	206,288	-1.7297	0.7665	587,982	150,000	
18	8	437,982	306,587	131,395	0.8048	1.2725	529,639	150,000	
19	9	379,639	265,748	113,892	0.5611	1.2119	443,005	150,000	
20	10	293,005	205,104	87,902	-0.0473	1.0731	313,431	150,000	163,431
21									
22		Normally distributed random numbers generated by =NORM.S.INV(RAND())					=C20*F20+D20*EXP($B$3)		
23									
24	Wealth at beginning of year =G19-H19			1+return on risky investment =EXP($B$5+$B$6*E20)					
25									
26									
27		Investment in risky asset =B20*$B$7							

In this spreadsheet column B shows the wealth at the beginning of every year. The wealth is divided between risky and riskless investments according to the proportions in cell B7. The riskless investment earns a continuously compounded return of 6% (meaning: \$100 invested in the riskless grows to $100{*}e^{6\%}$ at the end of the year). The risky part of the investment grows by a factor of $e^{\mu+\sigma * Z} = e^{8\%+20\% * Z}$, where Z is a random number which is normally distributed with mean 0 and standard deviation 1. As explained in Chapter 24, one way of generating these numbers is to use the Excel function **Norm.S.Inv(Rand( ))**. With each press of **F9**, this function recomputes and produces another normally distributed random number.

In the above simulation the investor has money left at the end of the 10-year period. But it is clear that not every simulation will leave the investor with spare cash at the end of year 10. A few presses of **F9** to recalculate the spreadsheet will produce something like this:

	A	B	C	D	E	F	G	H	I
1	**PLANNING YOUR RETIREMENT**								
2	Current wealth	1,000,000							
3	Riskfree rate	6%							
4	Parameters of risky investment								
5	Expected annual return	8%							
6	Standard deviation of return	20%							
7	Proportion invested in risky	70%							
8	Annual drawdown	150,000							
9									
10	**Year**	**Wealth at beginning of year**	**Invested in risky**	**Invested in bonds**	**Random number, normally distributed**	**1+return on risky investment**	**Wealth at end of year**	**Drawdown**	**Left at end of 10 years**
11	1	1,000,000	700,000	300,000	0.4850	1.1936	1,154,087	150,000	
12	2	1,004,087	702,861	301,226	1.2441	1.3893	1,296,359	150,000	
13	3	1,146,359	802,451	343,908	1.9765	1.6085	1,655,924	150,000	
14	4	1,505,924	1,054,147	451,777	-0.1250	1.0565	1,593,464	150,000	
15	5	1,443,464	1,010,425	433,039	-0.5280	0.9747	1,444,709	150,000	
16	6	1,294,709	906,296	388,413	-0.2036	1.0401	1,355,033	150,000	
17	7	1,205,033	843,523	361,510	-1.6497	0.7788	1,040,840	150,000	
18	8	890,840	623,588	267,252	0.3517	1.1622	1,008,528	150,000	
19	9	858,528	600,970	257,559	-0.7976	0.9236	828,516	150,000	
20	10	678,516	474,961	203,555	-0.7732	0.9281	656,939	150,000	506,939
21									
22		Normally-distributed random numbers generated by =NORM.S.INV(RAND())					=C20*F20+D20*EXP($B$3)		
23									
24	Wealth at beginning of year =G19-H19			1+return on risky investment =EXP($B$5+$B$6*E20)					
25									
26									
27		Investment in risky asset =B20*$B$7							

What interests us is *what percentage of the investment-consumption paths will end with a positive remainder*? We will use Monte Carlo techniques to answer this question. But before we do, we consider an economic question.

Should We Apply the 70% Rule Blindly?

In the above simulations we have split our investment between the risky and the riskless investments mechanically. We have also continued to draw down funds irrespective of whether there are funds in the account.

In the spreadsheet below we correct this. We assume that the investor defines a "safety cushion." The cushion in cell B8 below is 3, which is taken to mean that the annual drawdown is $150,000 if the investor's portfolio is worth at least 3*$150,000 at the end of the year. If this is not true, then the investor takes 1/3 of her portfolio value as a drawdown:

	A	B	C	D	E	F	G	H	I
1	**PLANNING YOUR RETIREMENT** **Using a safety cushion of 3** **Investor takes 150,000 if end-year wealth > 3*150,000, else takes end-year wealth/3**								
2	Current wealth	1,000,000							
3	Riskfree rate	6%							
4	Parameters of risky investment								
5	Expected annual return	8%							
6	Standard deviation of return	20%							
7	Proportion invested in risky	70%							
8	Safety cushion	3							
9	Annual drawdown	150,000							
10									
11	**Year**	**Wealth at beginning of year**	**Invested in risky**	**Invested in bonds**	**Random number, normally distributed**	**1+return on risky investment**	**Wealth at end of year**	**Drawdown**	**Left at end of 10 years**
12	1	1,000,000	700,000	300,000	-0.5471	0.9710	998,253	150,000	
13	2	848,253	593,777	254,476	0.4716	1.1904	977,072	150,000	
14	3	827,072	578,950	248,121	-0.8906	0.9065	788,306	150,000	
15	4	638,306	446,814	191,492	0.4212	1.1785	729,907	150,000	
16	5	579,907	405,935	173,972	0.9129	1.3003	712,560	150,000	
17	6	562,560	393,792	168,768	0.0895	1.1029	613,500	150,000	
18	7	463,500	324,450	139,050	-0.3590	1.0082	474,770	150,000	
19	8	324,770	227,339	97,431	-0.9809	0.8903	305,860	101,953	
20	9	203,906	142,734	61,172	-1.1469	0.8612	187,882	62,627	
21	10	125,255	87,678	37,576	-0.2989	1.0204	129,369	43,123	86,246
22									
23		Normally-distributed random numbers generated by =NORMSINV(RAND())					=C21*F21+D21*EXP($B$3)		
24									
25	Wealth at beginning of year =G20-H20			1+return on risky investment =EXP($B$5+$B$6*E21)					
26									
27							Drawdown calculated as =IF(G21>$B$8*$B$9,$B$9,G21/$B$8)		
28		Investment in risky asset =B21*$B$7							

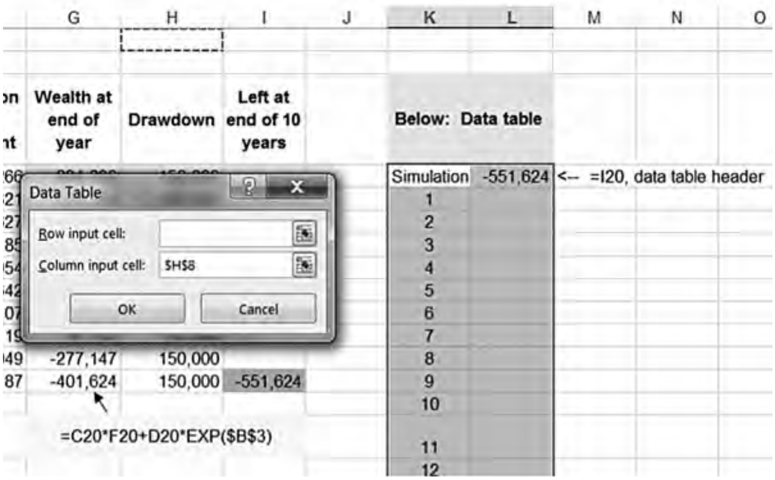
Many rules like this can be devised. The problem here—phrased as investment and payouts over retirement—is substantially the same as that faced by any endowment manager struggling with the problem of how to determine simultaneously the investment and the drawdown policy of the endowment. As far as we know there is no analytical solution to this problem, though as can be seen, it is not difficult to simulate the problem.

25.5 A Monte Carlo Simulation of the Investment Problem

In this section we run multiple simulations of the investment problem. The first set of simulations is run in the Excel spreadsheet itself, using the technique of "Data table on a blank cell" explained in section 31.7. The second set of simulations is run in VBA.

Data Table on a Blank Cell

We set up a **Data Table** on the investment simulation spreadsheet. Note that the data table header refers to the bequest in cell I20 and that the **Column input cell** in the Data Table dialog box refers to an empty cell. This is the technique explained in section 31.7:



There are 20 rows in the **Data Table**, though only 12 are shown. When we select **OK**, we run the simulation 20 times. Here is one run and some statistics. Pressing **F9** will run the simulation again:

	K	L	M	N	O	P	Q
5	Average	227,828	<-- =AVERAGE(L12:L30)				
6	Sigma	401,551	<-- =STDEV.S(L12:L31)				
7	Negative bequest	30%	<-- =COUNTIF(L12:L31,"<0")/COUNT(L12:L31)				
8							
9							
10	**Below: Data table**						
11	Simulation	220,554	<-- =I20, data table header				
12	1	-77,438					
13	2	372,005					
14	3	242,134					
15	4	-94,078					
16	5	754,722					
17	6	145,913					
18	7	-447,953					
19	8	211,791					
20	9	808,196					
21	10	724,259					
22	11	-352,192					
23	12	-136,035					
24	13	263,919					
25	14	106,654					
26	15	809,094					
27	16	378,955					
28	17	-193,356					
29	18	699,718					
30	19	112,415					
31	20	753,354					

Running the Simulation in VBA

We write a VBA function **SuccessfulRuns**, which simulates the above problem. Given an investment policy, the function **SuccessfulRuns** determines the percentage of investment/drawdown trajectories which will leave the retiree with positive wealth at the end of his investment horizon.

Here is some output from this function:

	A	B	C	D	E
1	HOW WELL DO WE DO? PERCENTAGE OF POSITIVE OUTCOMES				
2	Current wealth	1,000,000			
3	Riskfree rate	8%			
4	Parameters of risky investment				
5	Expected annual return	10%			
6	Standard deviation of return	40%			
7	Proportion invested in risky	40%			
8	Annual drawdown	100,000			
9	Years of investment	10			
10					
11	Runs	1,000			
12					
13	Successful runs	87.60%	<-- =successfulruns(B2,B8,B3,B5,B6,B7,B9,B11)		

The function result in cell B13 considers a retiree starting with \$1 million, making an investment decision between a risk-free asset with return 7% and a risky asset with stochastic returns $\mu = 10\%$ and $\sigma = 40\%$, and desiring to draw down \$100,000 per year. Simulating 1,000 returns, we determine that in 87.6% of the cases the investor will finish with positive wealth at the end of 10 years. Running the function again will, of course, produce different results.

The VBA Code for SuccessfulRuns

We almost forgot! Here it is:

```
Function SuccessfulRuns(Initial, _
Drawdown, Interest, Mean, Sigma, _
PercentRisky, Years, Runs)
   Dim PortfolioValue() As Double
   ReDim PortfolioValue(Years + 1)
   Dim Success As Integer

   Up = Exp(Mean + Sigma)
   Down = Exp(Mean - Sigma)

   PiUp = (Exp(Interest) - Down) / (Up - Down)
   PiDown = 1 - PiUp

   For Index = 1 To Runs
   For j = 1 To Years
   Randomize

   PortfolioValue(0) = Initial
   If Rnd > PiDown Then
      PortfolioValue(j) = _
      PortfolioValue(j - 1) * _
      PercentRisky * Up + PortfolioValue(j - 1) _
      * (1 - PercentRisky) * _
      Exp(Interest) - Drawdown

   Else
     PortfolioValue(j) = PortfolioValue(j - 1) * _
     PercentRisky * Down + PortfolioValue(j - 1) _
     * (1 - PercentRisky) * Exp(Interest) _
     - Drawdown
   End If
```

```
    Next j
        If PortfolioValue(Years) > 0 _
        Then Success = Success + 1
    Next Index

    SuccessfulRuns = Success / Runs

End Function
```

25.6 Summary

Monte Carlo methods are experimental techniques for determining the numerical value of a function or a procedure. In this chapter we have used the example of π to illustrate how Monte Carlo might be applied. As a valuation tool, MC methods are to be avoided when there is another, closed-form, way of determining the value. As we illustrate in exercises to this chapter, MC is not a good way to compute the value of π, given that many excellent formulas exist that approximate π with great accuracy. In cases where this is not true, however, you can use Monte Carlo to approximate the value.

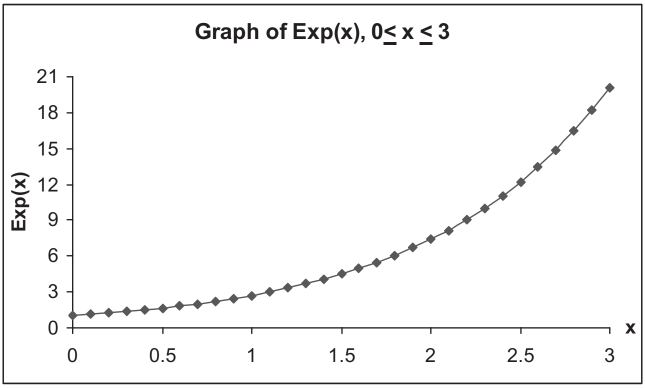
Monte Carlo methods can also be used to simulate investment problems in order to gain insight into the meaning of asset return uncertainty. In this chapter we illustrated this with some investment examples.

Exercises

1. In section 25.2 we designed a macro which calculates the value of π using Monte Carlo and which updates the screen every iteration. Modify the macro so that it updates the screen only every 1,000 iterations.
   Hints:
   - Use the VBA function **Mod**. From the VBA help menu, note that this function has syntax **a Mod b.** (A similar Excel function has syntax **Mod(a,b)**, but cannot be used in VBA.)
   - Use the VBA commands **Application.ScreenUpdating = True** and **Application.ScreenUpdating = False** to control the updating of the screen.

2. In the previous exercise, put a “switch” on the spreadsheet itself, which controls the updating of the macro (whether to update, yes or no, and how often to update).

3. Use Monte Carlo to calculate the integral of the function Exp(x) for $0 < x < 3$. Here’s the graph of this function:



4. One of the messages of this chapter is that while Monte Carlo is a clever method of calculation, it shouldn't be used when some better method exists. The MC valuation of π in section 25.2, for example, converges very slowly. It's a crummy way to compute π, since there are well-known methods for doing this. To see this, answer the following questions:

- Approximately how many runs does it require until you get 4-decimal accuracy for π using our MC simulation?
- Approximately how many runs does it require until you get 8-decimal accuracy for π using our MC simulation?
- Approximately how many runs does it require until you get 16-decimal accuracy for π using our MC simulation?

Note on exercises 5 and 6: We show you two alternative ways to compute the value of π. Both are substantially better than Monte Carlo!

5. The first method for computing π is:

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$$

Use this formula to approximate π.

6. The great Indian mathematical genius Ramanujan showed that

$$\frac{1}{\pi} = \frac{\sqrt{8}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!}{(n!)^4} \frac{(1103 + 26390n)}{396^{4n}}$$

The $n!$ indicates the factorial:

$$n! = n*(n-1)*(n-2)*\ldots*2*1$$
$$0! = 1$$

Excel's function **Fact** computes the factorial.

Use this series to construct a VBA function to value π, where n is the number of terms in the series. Show that two iterations give you over 15 digits of accuracy.

Some Comments on the Value of π

Exercise 6 is based on an article by D. H. Bailey, J. M. Borwein, and P. B. Borwein: "Ramanujan, Modular Equations, and Approximations to π Or How to Compute One Billion Digits of Pi." The article originally appeared in the *American Mathematical Monthly*, 1987, volume 96, number 3, pages 201–219. It can be downloaded at http://www.cecm.sfu.ca/organics/papers/borwein/index.html. Ramanujan's method in exercise 6 adds roughly 8 digits with each iteration. Ramanujan developed an even faster method, where only 13 iterations provide more than *1 billion* digits of π. (Excel's maximal accuracy is only 15 digits, but there is a nice Excel add-in which extends the precision to 32,767 digits: http://precisioncalc.com/).

Srinivasan Ramanujan (1887–1920) was one of the great mathematical geniuses of all time. Read a biography of this unique individual: *The Man Who Knew Infinity: A Life of the Genius Ramanujan* by Robert Kanigel. Published by Charles Scribner, 1991 (paperback edition—Washington Square Press, 1992).

Finally: So what's the value of π? *Mathematica*—a very sophisticated mathematical programming language (http://www.wolfram.com)—gives the following values for π. The values are computed using one of Ramanujan's formulas.

Number of Significant Digits	Value of π
25	3.141592653589793238462643
50	3.141592653589793238462643383279502884197169 3993751
75	3.141592653589793238462643383279502884197169 39937510582097494459230781640629
500	3.141592653589793238462643383279502884197169 39937510582097494459230781640628620899862 8 0348253421170679821480865132823066470938446 0955058223172535940812848111745028410270193 8521105559644622948954930381964428810975665 9334461284756482337867831652712019091456485 6692346034861045432664821339360726024914127 3724587006606315588174881520920962829254091 7153643678925903600113305305488204665213841 4695194151160943305727036575959195309218611 7381932611793105118548074462379962749567351 885752724891227938183011949 1

26 Simulating Stock Prices

26.1 Overview

The Monte Carlo approach to asset pricing is based on the simulation of asset prices. By simulating asset prices, we hope to answer questions relating to the investment outcomes such as:

- What is the probability that a given savings/spending pattern over time will enable a positive pension bequest?
- What is the likelihood that a discretely updated portfolio approach to option replication will successfully duplicate the option outcome?
- How can we price options whose terminal prices depend on the price path of an asset?

In order to simulate prices, we must have some assumptions about the distributional properties of stock prices. The standard finance assumption is that stock prices are lognormally distributed or, equivalently, that stock returns are normally distributed. In this chapter we give meaning to these two statements and show how they can be simulated in Excel. Subsequent chapters of Section V apply these concepts to the modeling of investments and options.

We have touched on this topic in our discussion of option pricing. In Chapter 16 we discussed the assumption that the development of a stock price over time follows a binomial model. We return to the binomial model in Chapter 30, where we use it to show Monte Carlo price-path-dependent options. In Chapter 17, we showed that the Black-Scholes model uses the assumption that stock prices are lognormally distributed. On a deeper level, we showed numerically in section 16.7 that the binomial model is in the limit equivalent to the assumption of lognormality. We did this by showing that the limiting price of a binomial option pricing model is the Black-Scholes price.

This chapter focuses on the lognormality of prices and shows how this assumption can be simulated. The structure of the chapter is as follows:

- We start with a discussion of what constitutes “reasonable” assumptions about stock prices.
- We then discuss why the lognormal distribution is a reasonable distribution for stock prices.
- Next, we show how to simulate lognormal price paths.
- Finally, we show you how to derive the parameters of the lognormal distribution—the mean and standard deviation of the stock returns—from historical stock price data.

26.2 What Do Stock Prices Look Like?

What are reasonable assumptions about the way stock prices behave over time? Clearly the price of a stock (or any other risky financial asset) is uncertain. What is its distribution? This is a perplexing question. One way to answer this question is to ask what are reasonable statistical properties of a stock price. Here are five reasonable properties:

1. The stock price is uncertain. Given the price today, we do not know the price tomorrow.

2. Changes in the stock's price are continuous. Over short periods of time, changes in a stock's price are very small, and the change goes to zero as the time span goes to zero.[1]

3. The stock price is never zero. This property means that we exclude the stocks of "dead" companies.

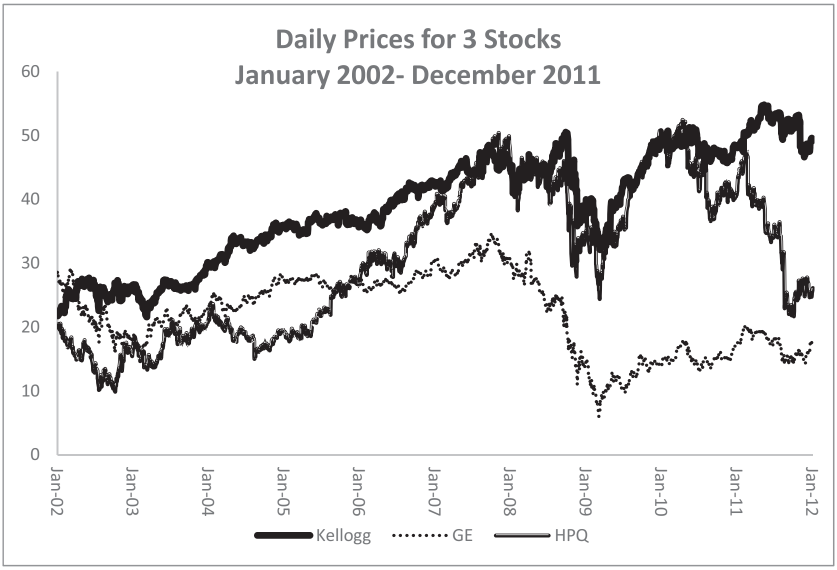
4. The average return from holding a stock tends to increase over time. Notice the word "tends": We do not *know* that holding a stock for a longer time will lead to a higher return; however, we *expect* that holding a risky asset over a longer term will lead to a higher *average* return.

5. The *uncertainty* associated with the return from holding a stock also tends to increase the longer the stock is held. Thus, given the stock's price today, the variance of the stock price tomorrow is small; however, the price variance in 1 month is larger, and the variance in 1 year is larger still.

Reasonable Stock Properties and Stock Price Paths

One way of viewing these five "reasonable properties" of stock prices is to think about *price paths*. A stock price path is a graph of a stock price over a period of time. Here, for example, are the price paths of several stocks:

1. If you have watched stock prices, you know that continuity is usually not a bad assumption. Sometimes, however, it can be disastrous (look at the way stock market prices behaved in October 1987, for a dramatic example of price *discontinuities*). It is possible to build a stock price model that assumes that prices are usually continuous but have occasional (and random) jumps. See Cox and Ross (1976), Merton (1976), and Jarrow and Rudd (1983).



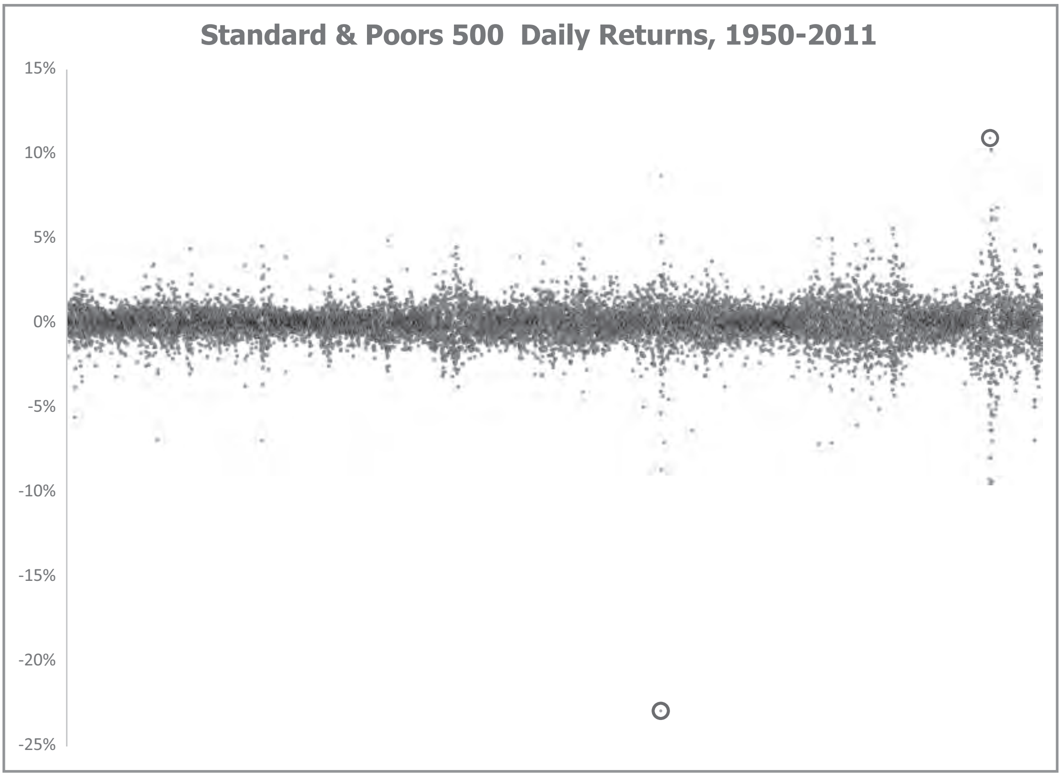
If we simulated stock price paths (something we will do using the lognormal model later in this chapter), how would we expect them to look? Our five properties imply that we would expect:

1. Wiggly lines.
2. Lines that are continuous (solid), with no jumps.
3. Lines that are always positive and never cross zero, no matter how low they get.
4. That at a given point in time, the average over all plausible lines is greater than the initial price of the stock. The farther out we go, the higher this average becomes.
5. That the standard deviation over all plausible lines is greater the farther out we go.

Here's another way of thinking about stock prices. Suppose we take the daily returns on the Standard & Poor's 500 index (we only show the start of the data):

	A	B	C	D
1	**SP500 DAILY PRICES 1950-2011**			
2	Date	Price	Return	
3	03-01-50	16.66		
4	04-01-50	16.85	1.13%	<-- =LN(B4/B3)
5	05-01-50	16.93	0.47%	
6	06-01-50	16.98	0.29%	
7	09-01-50	17.08	0.59%	
8	10-01-50	17.03	-0.29%	
9	11-01-50	17.09	0.35%	
10	12-01-50	16.76	-1.95%	
11	13-01-50	16.67	-0.54%	
12	16-01-50	16.72	0.30%	
13	17-01-50	16.86	0.83%	
14	18-01-50	16.85	-0.06%	
15	19-01-50	16.87	0.12%	

If we graph these returns over any given period, we get a mess of dots which is difficult to interpret:

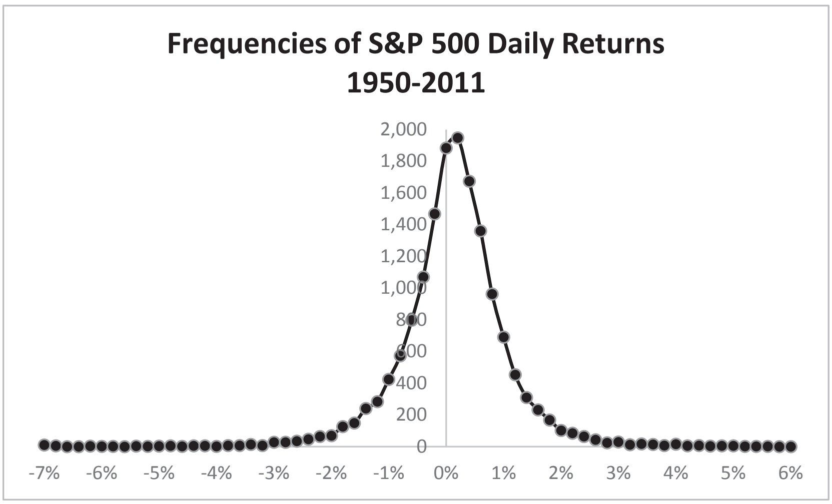


We have marked the largest daily increase and decline with circles: –22.9% on 19 October 1987 and +10.96% on 13 October 2008.

This smear of dots on a graph is difficult to interpret. Excel can help us make some sense of the data:

	A	C	D	E	F	G	H	I
1	**S&P 500 DAILY PRICES 1950-2011**							
2	Date	Return						
3	3-Jan-50				Number of daily returns	15,601		
4	4-Jan-50	1.13%	<-- =LN(B4/B3)		Number of days per year	251.629	<-- =G3/62	
5	5-Jan-50	0.47%	<-- =LN(B5/B4)					
6	6-Jan-50	0.29%	<-- =LN(B6/B5)		Max	10.96%	13-Oct-08	
7	9-Jan-50	0.59%			Min	-22.90%	19-Oct-87	
8	10-Jan-50	-0.29%						
9	11-Jan-50	0.35%			Number of returns between -1% and +1%	12,425	<-- =COUNTIFS(C:C,"<=1%",C:C,">=-1%")	
10	12-Jan-50	-1.95%			Average daily return	0.0278%	<-- =AVERAGE(C:C)	
11	13-Jan-50	-0.54%			Standard deviation of daily return	0.9822%	<-- =STDEV.S(C:C)	
12	16-Jan-50	0.30%						
13	17-Jan-50	0.83%			Average annual return	7.00%	<-- =G4*G10	
14	18-Jan-50	-0.06%			Standard deviation of annual return	15.58%	<-- =SQRT(G4)*G11	

Another way to look at the data is to plot the frequency of the returns, using Excel's **Frequency** function. The details are on the disk with the book. The plot looks approximately normally distributed, though perhaps with some left skewness:



An Excel Note

In the computations above:

- We have used **CountIfs** to count the number of daily returns between –1% and +1%.
- We have used the shortcut **C:C** to indicate all the data in column C.
- For the 62 years of data there are 251.629 average business days per year. This number is used to convert the average daily return and standard deviation of daily return to annual statistics:
  - Average annual return = (days per year) * (average daily return)
  - Standard deviation of annual return = **Sqrt(days per year)** * standard deviation of daily return
- To compute the date on which the max and min returns occur, we use the Excel functions **Match** and
  - **MATCH(G6,$C$3:$C$15604,0)** finds the row in the range C3:C15604 in which the value in G6 (the max return) occurs.
  - **INDEX($A$3:$A$15604,MATCH(G6,$C$3:$C$15604,0))** finds the value in row **MATCH(G6,$C$3:$C$15604,0)** in **A3:A15604**. This is the date of the occurrence.

Computing Returns and Their Distribution for a Continuous Return-Generating Process[2]

We can compute the stock return over a period by taking the natural logarithm of the price relatives, defined as $\ln\left(\frac{Price_t}{Price_{t-1}}\right)$. Furthermore, if $\{r_1, r_2, \ldots, r_M\}$ is a series of periodic returns, we can compute the mean, variance, and standard deviation of the returns as:

2. This subsection anticipates Section 26.7.

$$Periodic\ mean = \mu_{Periodic} = \underbrace{\frac{1}{M}\sum_{t=1}^{M} r_t}_{\substack{\uparrow \\ \text{Use Excel} \\ \textbf{Average} \\ \text{function}}},$$

$$Periodic\ variance = \sigma^2_{Periodic} = \underbrace{\frac{1}{M-1}\sum_{t=1}^{M}\left(r_t - \mu_{Periodic}\right)^2}_{\substack{\uparrow \\ \text{Use Excel} \\ \textbf{Stdev.s}\text{ function}}}$$

Assuming that there are n periods per year, we can compute the annualized returns by

$$Annualized\ mean = n * \mu_{Periodic}$$

$$Annualized\ variance = n * \sigma^2_{Periodic}$$

$$Annualized\ standard\ deviation = \sigma\sqrt{n}$$

26.3 Lognormal Price Distributions and Geometric Diffusions

In this section we get a bit more formal and describe what we mean by a lognormal price distribution. We then relate the lognormal price process to a geometric diffusion.

Suppose we denote by S_t the price at time t of a share of stock. The lognormal distribution assumes that *the natural logarithm of one-plus-the-return* from holding a share of stock between time t and time $t + \Delta t$ is normally distributed with mean μ and standard deviation σ. Denote the (uncertain) rate of return over an interval Δt by $\tilde{r}_{\Delta t}$. Then we can write $S_{t+\Delta t} = S_t \exp[\tilde{r}_{\Delta t}\Delta t]$. In the lognormal distribution, we assume that the rate of return $\tilde{r}_{\Delta t}$ over a short period Δt is normally distributed with mean $\mu\ \Delta t$ and variance $\sigma^2\ \Delta t$.

Another way of writing this relation is to write the stock price $S_{t+\Delta t}$ at time $t + \Delta t$ in the following way:

$$\frac{S_{t+\Delta t}}{S_t} = \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right]$$

where Z is a standard normal variable (mean = 0, standard deviation = 1).[3]

To see what this assumption means, suppose first that $\sigma = 0$. In this case we have

$$S_{t+\Delta t} = S_t \exp[\mu\Delta t]$$

which simply says that the stock price grows at an exponential rate with certainty. In this case the stock is like a riskless bond that bears interest rate μ, continuously compounded.

Now suppose that $\sigma > 0$. In this case, the lognormal assumption says that, although the tendency is for the stock price to increase, there is an uncertain element (normally distributed) that must be taken into account. The best way to think about this is in terms of a simulation. Suppose, for example, that we're trying to simulate a lognormal price process in which $\mu = 15\%$, $\sigma = 30\%$, and $\Delta t = 0.004$. Suppose the price at time 0 is $S_0 = 35$. To simulate the possible stock prices at time Δt we first have to pick (at random) a number Z from a standard normal distribution.[4] Suppose that this number is 0.1165. Then the stock price $S_{\Delta t}$ at time Δt will be

$$\begin{aligned} S_{\Delta t} &= S_0 * \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right] \\ &= 35 * \exp\left[0.15 * 0.004 + 0.3 * 0.1165 * \sqrt{0.004}\right] = 35.0985 \end{aligned}$$

Of course we could have drawn a different random number. If, for example, our random number Z had been –0.9102, then we would have

$$\begin{aligned} S_{\Delta t} &= S_0 * \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right] \\ &= 35 * \exp\left[0.15 * 0.004 - 0.3 * (-0.9102) * \sqrt{0.004}\right] = 34.4214 \end{aligned}$$

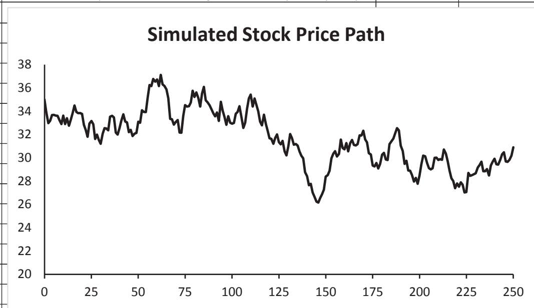
3. If you know about diffusion processes, then the lognormal price process is a *geometric diffusion*:

$\frac{dS}{S} = \left(\mu + \frac{\sigma^2}{2}\right)dt + \sigma dB$, where dB is a Wiener process ("white noise"): $dB = Z\sqrt{dt}$, where Z is a standard random variable.

4. See Chapter 24 for techniques (using both Excel and VBA) for generating random numbers.

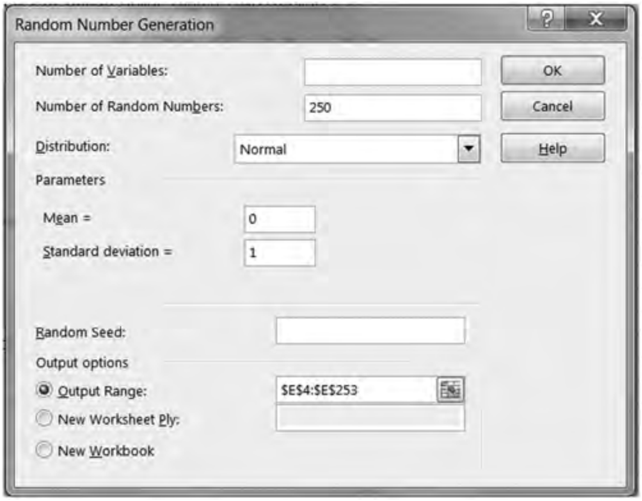
This process is illustrated in the spreadsheet picture below, where we generated a list of 250 numbers picked from a standard-normal distribution (the technical nomenclature is “standard-normal deviates”).[5] Each is an equally likely potential candidate to be Z. Having picked Z for a particular time interval Δt, the price $S_{t+\Delta t}$ follows.

	A	B	C	D	E	F	G	H	I
1	**CONCEPTUALIZING THE LOGNORMAL DISTRIBUTION**								
2				**Time**	**Normal deviates**	**Stock price**			
3	Mean	15%		0		35.0000	<-- =B6		
4	Sigma	30%		1	-1.697499	33.9110	<-- =F3*EXP(Mean*deltat+Sigma*SQRT(deltat)*E4)		
5	Δt	0.004		2	-1.503086	32.9774			
6	Initial stock price	35		3	0.2900038	33.1792			
7				4	0.7383846	33.6675			
8				5	-0.006387	33.6836			
9				6	-0.110343	33.6334			
10				7	-0.055375	33.6182			
11				8	-0.661842	33.2186			
12				9	-0.534442	32.9032			
13				10	1.1193083	33.6296			
14				11	-1.182809	32.9030			
15				12	0.7687686	33.4065			
16				13	-1.071367	32.7540			
17				14	0.813543	33.2834			
18				15	0.8621168	33.8527			
19				16	0.9543328	34.4919			
20				17	-0.872592	33.9459			



The spreadsheet uses **Data|Data Analysis|Random Number Generation** to generate a list of 250 standard-normal deviates. The command looks like this:

5. The number of business days in a year is approximately 250. Thus when we define $\Delta t = 1/250 = 0.004$, we are simulating the stock price on a daily basis over the course of a year.



To summarize: In order to simulate *the growth of the stock price*, when the price follows a lognormal price distribution:

- Multiply Δt (the elapsed time interval) by μ (the average rate of growth). This gives the certain portion of the return.
- Take a draw Z from a random variable which is standard normal, and then multiply this draw by $\sigma\sqrt{\Delta t}$. This gives the uncertain portion of the return. (The square root implies that the variance of the stock's return is linear in time. See below.)
- Add the two results and exponentiate. The daily return is $\exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right]$. If the price on date t is S_t, then the price on date t + 1 is $S_{t+1} = S_t \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right]$.

26.4 What Does the Lognormal Distribution Look Like?

We know that the normal distribution produces a "bell curve." What about the lognormal distribution? In the following experiment we simulate 1,000 random end-of-year stock prices. The experiment is a continuation of the experiment performed in the previous section; since we are simulating end-of-year prices, we set $\Delta t = 1$. To perform this experiment:

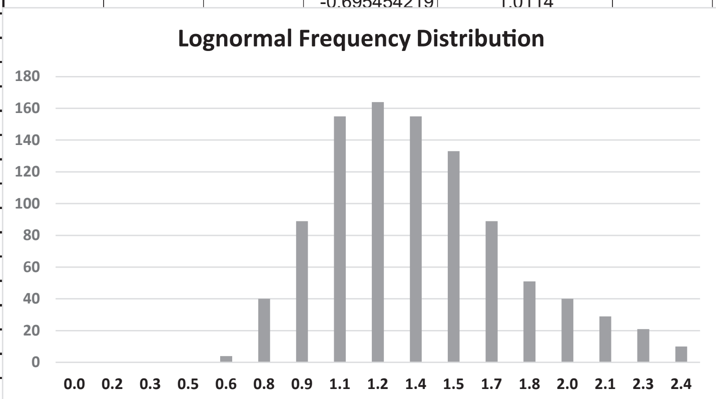
- We produce a list of 1,000 normal deviates.
- We use each normal deviate to produce an end-of-period stock price

$$S_1 = S_0 * \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\,\right] = S_0 * \exp[\mu + \sigma Z], \text{ since } \Delta t = 1$$

- We put the stock prices into bins and produce a histogram.

Here's what the spreadsheet for this experiment looks like:

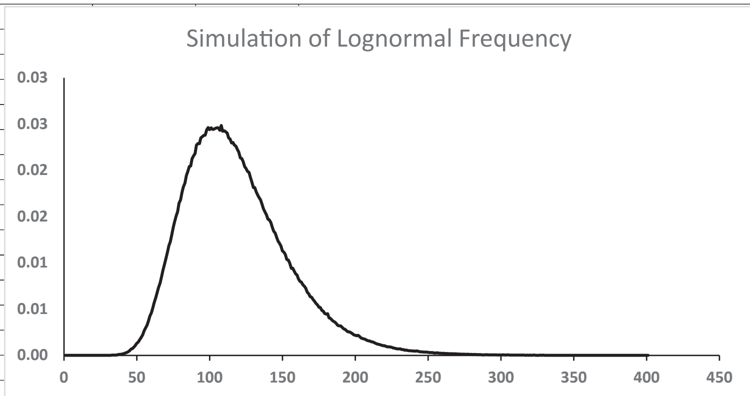
	A	B	C	D	E	F	G	H	I
1	**THE LOGNORMAL HISTOGRAM**								
2	Mean	22%		**1000 normally distributed numbers**	**Lognormal =exp(mu+sigma*Z)**		**Bins**	**Frequency**	
3	Sigma	30%							
4	Δt	1		-0.723596258	1.0029		0.00	0	
5				-0.447857929	1.0894		0.15	0	
6				-0.695454219	1.0114		0.30	0	
7							0.45	0	
8							0.60	4	
9							0.75	40	
10							0.90	89	
11							1.05	155	
12							1.20	164	
13							1.35	155	
14							1.50	133	
15							1.65	89	
16							1.80	51	
17							1.95	40	
18							2.10	29	
19							2.25	21	
20							2.40	10	
21							Table above created with the array formula {=FREQUENCY(E:E,G4:G20)}		
22									
23				-0.954453299	0.9358				
24				1.969010555	2.2495				



Having produced 1,000 lognormal price relatives $\exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\,\right]$, we can use the array function **Frequency( )** (this function is discussed in Chapter 33) to put them into bins.

When we do this simulation for a large number of points, the resulting density curve becomes smooth. Here, for example, is the frequency distribution of 1,000,000 trials with $\mu = 12\%$, $\sigma = 30\%$, and $\Delta t = 1$:

	A	B	C	D	E	F
1	**SIMULATION OF LOGNORMAL RETURNS**					
2	Runs	1,000,000				
3	Mean	12%				
4	Sigma	30%				
5						
6	Starttime	10:04:31 PM				
7	Stoptime	10:04:35 PM				
8	Elapsed	00:00:04				
9						
10						
11						
12						
13						
14						
15						
16						
17						



The VBA program that produced this output is shown below:

```
'Simulating the lognormal distribution
'Note that I take delta = 1!
Sub RandomNumberSimulation()
Application.ScreenUpdating = False
Range("starttime") = Time
N = Range("runs").Value
mean = Range("mean")
sigma = Range("sigma")
ReDim Frequency(0 To 1000) As Integer
```

```
For Index = 1 To N
start:
    Static rand1, rand2, S1, S2, X1, X2
    rand1 = 2 * Rnd - 1
    rand2 = 2 * Rnd - 1
    S1 = rand1 ^ 2 + rand2 ^ 2
    If S1 > 1 Then GoTo start
    S2 = Sqr(-2 * Log(S1) / S1)
    X1 = rand1 * S2
    X2 = rand2 * S2

    Return1 = Exp(mean + sigma * X1)
    Return2 = Exp(mean + sigma * X2)

    Frequency(Int(Return1 / 0.01)) = _
    Frequency(Int(Return1 / 0.01)) + 1
    Frequency(Int(Return2 / 0.01)) = _
    Frequency(Int(Return2 / 0.01)) + 1
Next Index

For Index = 0 To 400
    Range("simuloutput").Cells(Index + 1, 1) = _
    Frequency(Index) / N
Next Index

Range("stoptime") = Time
Range("elapsed") = Range("stoptime") -
Range("starttime")
Range("elapsed").NumberFormat = "hh:mm:ss"

End Sub
```

The routine which produces randomly distributed standard-normal deviates is contained in the eight lines following the word `start`; this routine is further explained in Chapter 31.

26.5 Simulating Lognormal Price Paths

We now return to the problem of simulating lognormal price paths that we started to discuss in section 26.3. We shall try to understand, through a simulation written in VBA, the meaning of the following sentences: "The price of a stock today is $25. The price of the stock is distributed lognormally, with an annual log mean return of 10% and an annual log standard deviation of 20%." We want to know how the price of the stock might behave on a daily basis throughout the next year. There are an infinite number of price paths for the stock. What we will do is simulate (randomly) one of these paths. If we want another price path, we can merely rerun the simulation.

There are about 250 business days in a year. Therefore the daily price movement of the stock between day t and day $t + 1$ can be simulation by setting $\Delta t = 1/250 = 0.004$, $\mu = 10\%$, and $\sigma = 20\%$. If the initial price of the stock $S_0 = \$25$, then the price after one day will be

$$S_{\Delta t} = S_0 * \exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right] = 25 * \exp\left[0.15\cdot 0.004 + 0.20\cdot Z\sqrt{0.004}\right]$$

and the price after 2 days will be

$$S_{0.008} = S_{0.004} * \exp\left[0.15\cdot 0.004 + 0.20\cdot Z\sqrt{0.004}\right]$$

and so on. At each step the random normal deviate Z is the uncertain factor in the price return. Because of this uncertainty, all paths produced will be different.

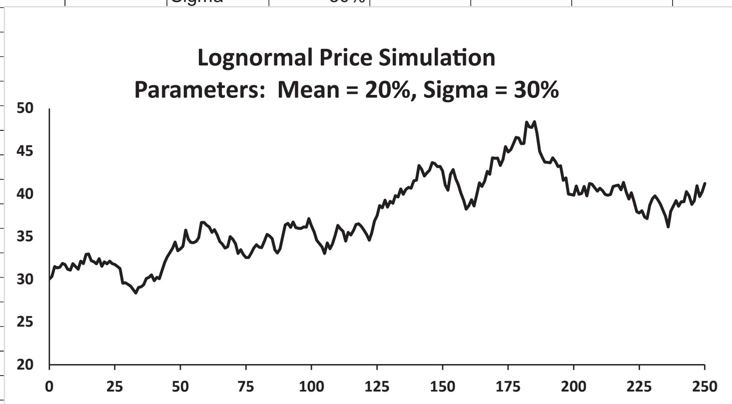
Here is a VBA program **PricePathSimulation** that reproduces a typical price path:

```
Sub PricePathSimulation()
Range("starttime") = Time

N = Range("runs").Value
mean = Range("mean")
sigma = Range("sigma")
delta_t = 1 / (2 * N)
```

```
ReDim price(0 To 2 * N) As Double

price(0) = Range("initial_price")

For Index = 1 To N
start:
    Static rand1, rand2, S1, S2, X1, X2
    rand1 = 2 * Rnd - 1
    rand2 = 2 * Rnd - 1
    S1 = rand1 ^ 2 + rand2 ^ 2
    If S1 > 1 Then GoTo start
    S2 = Sqr(-2 * Log(S1) / S1)
    X1 = rand1 * S2
    X2 = rand2 * S2

    price(2 * Index - 1) = price(2 * Index - 2)
    * Exp(mean * delta_t + _
        sigma * Sqr(delta_t) * X1)
    price(2 * Index) = price(2 * Index - 1) *
    Exp(mean * delta_t + _
        sigma * Sqr(delta_t) * X2)
Next Index

For Index = 0 To 2 * N
    Range("output").Cells(Index + 1, 1) = Index
    Range("output").Cells(Index + 1, 2) =
    price(Index)
Next Index

Range("stoptime") = Time
Range("elapsed") = Range("stoptime") -
Range("starttime")
Range("elapsed").NumberFormat = "hh:mm:ss"

End Sub
```

The output from this program looks like this on the spreadsheet:

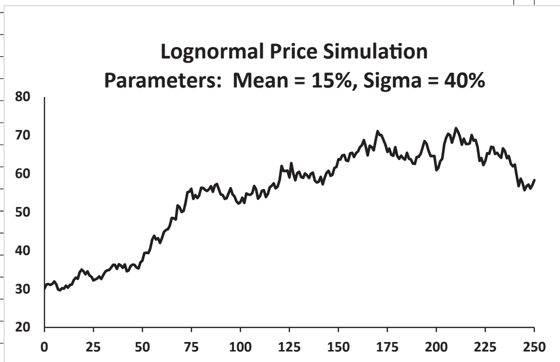
	A	B	C	D	E	F	G	H	I	J
1	**SIMULATING LOGNORMAL PRICE PATHS WITH VBA** Press [Ctrl]+R to operate macro									
2	**Day**	**Stock price**								
3	0	30.00			Runs	125				
4	1	30.33			Initial price	30				
5	2	31.49			Mean	20%				
6	3	31.32			Sigma	30%				
7	4	31.41								
8	5	31.87								
9	6	31.72								
10	7	31.16								
11	8	31.07								
12	9	31.84								
13	10	31.47								
14	11	31.18								
15	12	32.11								
16	13	31.84								
17	14	32.91								
18	15	32.95								
19	16	32.17								
20	17	32.05								
21	18	31.81								
22	19	32.40								
23	20	31.50								



Simulating Price Paths with Norm.Inv

In the spreadsheet below we do a similar simulation, using **Norm.Inv** to simulate the returns (see Chapter 31 for more details). $\mathbf{Norm.Inv}\left(\mathbf{Rand}(\ ), \mu\Delta t, \sigma\sqrt{\Delta t}\right)$ produces normally distributed returns with annual mean μ and annual standard deviation σ. $\mathbf{Exp}\left(\mathbf{Norm.Inv}\left(\mathbf{Rand}(\ ), \mu\Delta t, \sigma\sqrt{\Delta t}\right)\right)$ is the lognormally distributed price increase:

	A	B	C	D	E	F	G
1	**SIMULATING LOGNORMAL PRICE PATHS WITH NORM.INV** Uses Exp(Norm.Inv(rand(),Mean*Δt,Sqrt(Δt)*Sigma) to simulate returns Press F9 to produce new simulation						
2	**Day**	**Stock price**					
3	0	30.00			Initial price	30	
4	1	31.18	<-- =B3*EXP(NORM.INV(RAND(),Mean*$F$6,SQRT($F$6)*Sigma))		Mean	15%	
5	2	31.27	<-- =B4*EXP(NORM.INV(RAND(),Mean*$F$6,SQRT($F$6)*Sigma))		Sigma	40%	
6	3	31.09			Delta_t	0.004	<-- =1/250
7	4	31.29					
8	5	31.98					
9	6	31.25					
10	7	29.86					
11	8	29.65					
12	9	30.22					
13	10	30.05					
14	11	30.87					
15	12	30.34					
16	13	31.07					
17	14	31.14					
18	15	32.33					
19	16	32.97					
20	17	32.65					
21	18	34.34					
22	19	35.05					
23	20	34.53					
24	21	33.83					

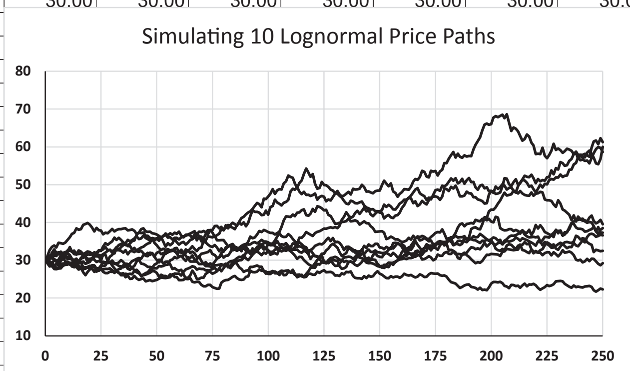


Ten Lognormal Price Paths

In the spreadsheet below we use a slightly different technique to simulate 10 lognormal price paths with the same statistical parameters. In each cell we use **Norm.S.Inv(Rand( ))** to draw a number from the standard normal distribution.

This normal deviate is then used to simulate the stock price, as illustrated below:

	A	B	C	D	E	F	G	H	I	J	K
1	**SIMULATING 10 LOGNORMAL PRICE PATHS**										
2	Initial price	30									
3	Mu	11%									
4	Sigma	30%									
5	Delta_t	0.004	<-- =1/250					=J9*EXP(Mu*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND()))			
6											
7		10 price paths									
8	Day	Path1	Path2	Path3	Path4	Path5	Path6	Path7	Path8	Path9	Path10
9	0	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
10	1	29.57								30.45	29.87
11	2	28.75								30.39	29.73
12	3	28.48								30.83	29.11
13	4	27.67								30.31	30.05
14	5	27.62								29.92	29.69
15	6	27.80								29.55	29.32
16	7	27.84								29.81	30.53
17	8	28.60								30.44	31.13
18	9	29.54								29.82	30.89
19	10	29.54								29.20	31.91
20	11	29.34								28.83	31.37
21	12	28.72								29.40	30.37
22	13	28.17								28.91	30.02
23	14	28.36								28.62	30.44
24	15	28.08								29.17	29.24
25	16	27.94	31.73	30.60	32.29	38.22	29.55	27.90	28.29	29.30	30.45

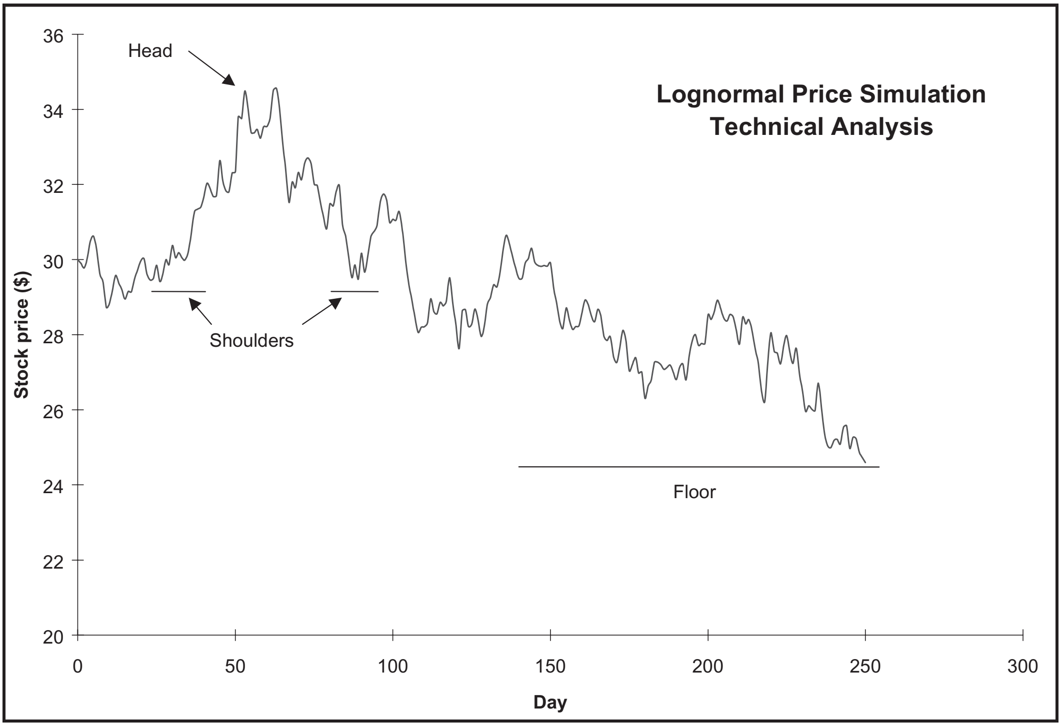


As you can see, on average the price of the asset increases over time, as does the variance of the returns. This accords with properties 4 and 5 of stock prices in section 26.3—we expect both the return on an asset and the uncertainty associated with this return to increase over time.

26.6 Technical Analysis

Security analysts are divided into “fundamentalists” and “technicians.” This division has nothing to do with their outlook on the Creator of the Universe, but rather with the way they regard stock prices. Fundamentalists believe that the value of a stock is ultimately determined by underlying economic variables. Thus, when a fundamentalist analyzes a company, she will look at its earnings, its debt/equity ratio, its markets, and so forth.

Technicians, in contrast, think that stock prices are determined by patterns. They believe that, by examining the pattern of past prices of a stock, they can predict (or at least make sensible statements about) the stock's future prices. A technician may tell you that "we're currently in a head-and-shoulders pattern," by which he means that a graph of the stock price looks like the figure below:



Other terms used by technicians include "floors" (there's one in the graph), "rebound levels," and "pennants."

The academic (some would say ivory tower) view of technical analysis is that it is worthless. A basic theory of financial theory says that markets efficiently incorporate the information known about the securities traded on them. There are several versions of this theory; one of them, the *weak efficient markets hypothesis*, says that at the very least all information about past prices is incorporated into the current price. The weak efficient markets hypothesis means that technical analysis cannot make predictions about futures prices, since technical analysis is based solely on past price information.[6]

Nevertheless, a lot of people believe in technical analysis (this in itself may give technical analysis some validity). The simulations we are running in this chapter will allow us to generate a myriad of patterns which, when analyzed, will yield "good" predictions of future prices. For example, in the figure above it appears that $24 is a floor for the stock price, since it never goes any lower. A perspicacious analyst can detect a clear head-and-shoulders pattern between days 40 and 100. There appears to be a ceiling of $35. Thus a technician might predict that the stock price will stay below $37 unless it rises above that level. (If you are going to be a technician, you have to learn to say these things with a straight face.)

26.7 Calculating the Parameters of the Lognormal Distribution from Stock Prices

The main purpose of this section is to show you how stock price data can be used to compute the annual mean return μ and the standard deviation of the annual return σ needed in the lognormal simulations (and—in Chapter 17—the σ needed as an input to the Black-Scholes formula). Before doing this, note that the mean and variance of the logarithm of the stock return over an interval Δt are

$$E\left[\ln\left(\frac{S_{t+\Delta t}}{S_t}\right)\right] = E\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right] = \mu\Delta t$$

$$\text{var}\left[\ln\left(\frac{S_{t+\Delta t}}{S_t}\right)\right] = \text{var}\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right] = \sigma^2\Delta t$$

This means that both the expected log return and the variance of the log return are linear in time.

Now suppose we want to estimate the lognormal μ and σ from data on historical prices. It follows that

$$\mu = \frac{\text{mean}\left[\ln\left(S_{t+\Delta t}/S_t\right)\right]}{\Delta t},\ \sigma^2 = \frac{\text{var}\left[\ln\left(S_{t+\Delta t}/S_t\right)\right]}{\Delta t}$$

6. For a discussion of this point, see Chapter 13 of Brealey, Myers, and Allen (2005); for a more advanced treatment, see Chapters 10–11 of Copeland, Weston, and Shastri (2003).

To make things specific, the following spreadsheet gives monthly prices for a particular stock. From these prices we calculate the log returns and the *annualized* mean and standard deviation. Note that we have used the function **Stdevp** to calculate σ; this assumes that the data represent the actual distribution.

	A	B	C	D
1	**CALCULATING THE ANNUAL MEAN AND SIGMA OF RETURNS FROM MONTHLY PRICE DATA FOR HALLIBURTON CORPORATION Oct. 2011—Oct. 2013**			
2	Monthly average		1.23%	<-- =AVERAGE(C10:C33)
3	Monthly standard deviation		6.74%	<-- =STDEV.S(C10:C33)
4				
5	Annual average, μ		14.76%	<-- =12*C2
6	Annual standard deviation, σ		23.36%	<-- =SQRT(12)*C3
7				
8	**Date**	**Closing price**	**Monthly return**	
9	3-Oct-11	36.54		
10	1-Nov-11	36.08	-1.27%	<-- =LN(B10/B9)
11	1-Dec-11	33.84	-6.41%	<-- =LN(B11/B10)
12	3-Jan-12	36.07	6.38%	<-- =LN(B12/B11)
13	1-Feb-12	35.88	-0.53%	<-- =LN(B13/B12)
14	1-Mar-12	32.63	-9.49%	<-- =LN(B14/B13)
15	2-Apr-12	33.64	3.05%	<-- =LN(B15/B14)
16	1-May-12	29.55	-12.96%	
17	1-Jun-12	27.99	-5.42%	
18	2-Jul-12	32.67	15.46%	
19	1-Aug-12	32.39	-0.86%	

Note that the annual average log return is 12 times the monthly average log return, whereas the annual standard deviation is $\sqrt{12}$ the monthly standard deviation. In general if the return data are generated for n periods per year, then

$$mean_{annual\ return} = n \cdot mean_{periodic\ return},\ \sigma_{annual\ return} = \sqrt{n} \cdot \sigma_{periodic\ return}$$

Of course, this is not the only way to calculate the parameters of the lognormal distribution. We should mention at least two other methods:

- We can use some other procedure to extrapolate the mean and standard deviation of *future* returns from the past history of returns. One example of this would be to use a moving average.

- We can use the Black-Scholes formula to find the *implied volatility*: The σ of the stock's log returns which fits the price of an option on the stock. This is illustrated in section 17.4.

26.8 Summary

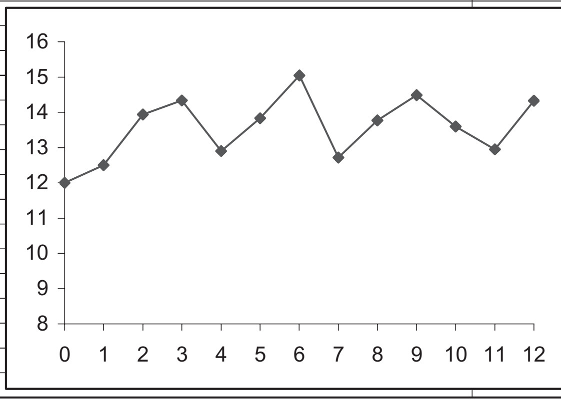
The lognormal distribution is one of the foundations of the Black-Scholes formula for option pricing discussed in the next chapter. In this chapter we have explored the meaning of lognormality for stock prices. We have shown how lognormality—the assumption that the returns on an asset are normally distributed—can be justified visually for the S&P 500 portfolio. We have also shown how to simulate price paths which are lognormally distributed. Finally, we have shown how to compute the mean and the standard deviation of a lognormal distribution from the historic returns of an asset.

Exercises

1. Use **Norm.S.Inv(Rand( ))** to produce a simulation of monthly stock prices, as illustrated below.[7]

7. The use of **Norm.S.Inv** for this purpose is discussed in section 24.4.

	A	B	C	D	E
1	**SIMULATING A LOGNORMAL PRICE PROCESS**				
2	Initial stock price	12			
3	Mean return, m	12%			
4	Return sigma, s	35%			
5	Delta t	8%	<-- =1/12		
6					
7	**Month**	**Random number**	**Stock price**		
8	0		12		
9	1	0.3021	12.4963	<-- =C8*EXP($B$3*$B$5+$B$4*SQRT($B$5)*B9)	
10	2	0.9810	13.9371		
11	3	0.1774	14.3318		
12	4	-1.1420	12.8982		
13	5	0.5932	13.8325		
14	6	0.7335	15.0463		
15	7	-1.7615	12.7198		
16	8	0.6826	13.7649		
17	9	0.4070	14.4869		
18	10	-0.7262	13.5974		
19	11	-0.5846	12.9464		
20	12	0.9055	14.3293		
21					
22					
23	Cell B20 contains formula =NORMSINV(RAND())				
24					
25					



2. Expand the previous exercise and use **Norm.S.Inv(Rand( ))** to produce a simulation of daily stock prices for 250 days (approximately 1 year of trading days).

3. Re-create the spreadsheet below. Play with the spreadsheet (each press of **F9** will recompute the numbers) to convince yourself that higher σ means a more volatile price path for the stock.

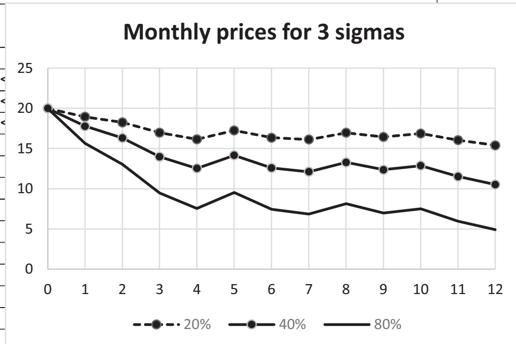
4. Write a VBA program which reproduces the lognormal frequency distribution for an arbitrary number of runs. That is, this program should

   - Produce N normal random deviates.
   - For each deviate produce a lognormal price relative $\exp\left[\mu\Delta t + \sigma Z\sqrt{\Delta t}\right]$.
   - Classify each price relative into a set of bins running from 0, 0.1, … , 3.
   - Put the frequencies on the spreadsheet and produce a frequency graph such as the one in section 26.5.

5. Run a few of the lognormal price path simulations. Examine the price pattern for trends. Find one or more of the following technical patterns:

   support area
   resistance area
   uptrend/downtrend
   head and shoulders

	A	B	C	D	E	F	G
1	**SIMULATING A LOGNORMAL PRICE PROCESS**						
2	Initial stock price	20					
3	Mean return, m	12%					
4	Delta t	0.0833	<-- =1/12				
5							
6			**Stock price with sigma =**				
7	**Month**	**Random number**	**20%**	**40%**	**80%**		
8	0		20.0000	20.0000	20.0000		
9	1	-1.10554	18.9519	17.7800	15.6492		
10	2	-0.82712	18.2497	16.3229	13.0580		
11	3	-1.43142	16.9710	13.9752	9.4766		
12	4	-1.02984	16.1521	12.5330	7.5459		
13	5	0.966426	17.2506	14.1534	9.5275		
14	6	-1.10249	16.3494	12.5869	7.4601		
15	7	-0.41797	16.1200	12.1144	6.8418		
16	8	0.707861	16.9612	13.2783	8.1378		
17	9	-0.70316	16.4501	12.3658	6.9876		
18	10	0.266112	16.8727	12.8798	7.5052		
19	11	-1.04668	16.0429	11.5283	5.9528		
20	12	-0.87832	15.4029	10.5211	4.9088		
21							



inverted head and shoulders

double top/bottom

rounded top/bottom

triangle (ascending, symmetrical, descending)

flag

6. The exercise file for this chapter contains daily price data for the S&P 500 index and for Abbott Laboratories for the 3 months April–June 2007. Use these data to compute the annual average, variance, and standard deviation of the logarithmic returns for the S&P and for Abbott. What is the correlation between the returns of the S&P 500 and Abbott?

7. The exercise file for this chapter gives daily returns from 1987–2012 for the Vanguard Index 500 fund (VFINX). This is a fund that tracks the S&P 500, but the returns include dividends (as opposed to ^GSPC, the index tracker).

   - Compute the overall daily return statistics: average and standard deviation.
   - Annualize these statistics, assuming that there are 250 days per year.
   - Compute the daily and annualized return statistics by year.

   Hint: Take a look at the functions **DAverage** and **DStdev** discussed in Chapter 33.

27 Monte Carlo Simulations for Investments

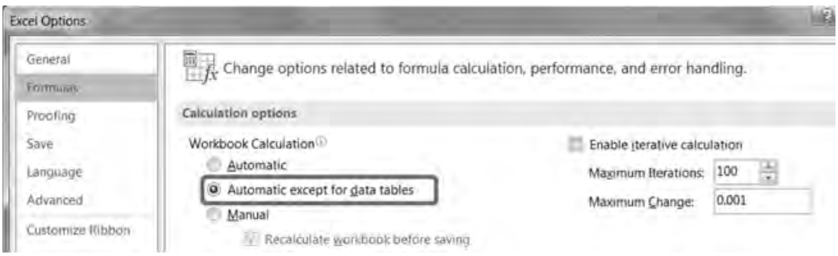
27.1 Overview

In this chapter we simulate the performance of portfolio investments of one or more stocks. We start with simulations of a single stock (a slight repetition of materials covered in the previous chapter). We then deal with correlation, first discussing the case of two correlated stock returns, and then generalizing to portfolios of multiple stocks, using the Cholesky decomposition (Chapter 24). We go on to discuss simulations of pension problems, where portfolio investment is used to finance future withdrawals. Finally we discuss the simulation of β, showing that low β stocks have higher α.

Throughout this chapter we use the technique of running a **Data Table** on a blank cell to do sensitivity analysis. This technique, which enables us to run multiple random simulations, is described in Chapter 31.

A Computational Note

The spreadsheets for this chapter are very computationally intensive. We have dealt with this problem by splitting them up into separate Excel notebooks dealing with the various issues. We also advise turning off the automatic computation of **Data Tables** (**File|Options|Calculation options|Automatic except for data tables**).



27.2 Simulating Price and Returns for a Single Stock

In this section we simulate portfolio investments. We start with an exercise we have already done in Chapter 26, namely, simulating stock returns over time. Suppose we start with one stock, and consider a possible future price path for this stock. In the spreadsheet below we simulate such a price path: At each point t, we generate a random return for the stock:

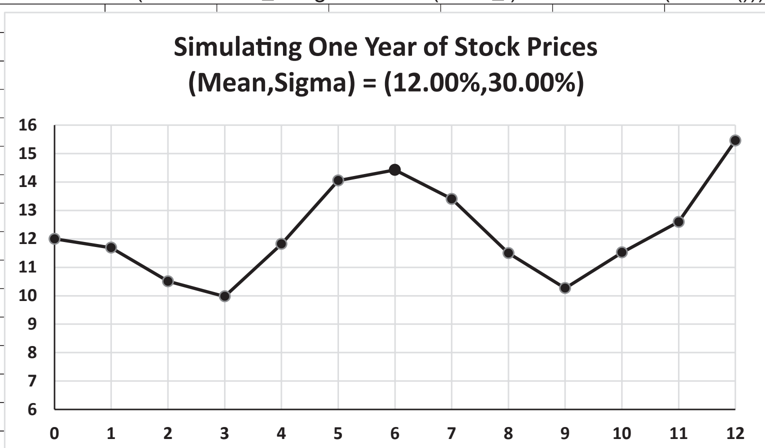
$$r_t = \mu\Delta t + \sigma\sqrt{\Delta t}Z$$

where Z is a standard normal deviate generated by **Norm.S.Inv(rand( ))**.[1] The stock price at time t is given by

$$S_t = S_{t-1}\exp\left[\mu\Delta t + \sigma\sqrt{\Delta t}Z\right]$$

Below is a simulation example for monthly stock prices:

	A	B	C	D	E	F	G	H	I
1	**SIMULATING MONTHLY STOCK PRICE**								
2	Mean	12%							
3	Sigma	30%							
4	Delta_t	0.0833	<-- =1/12						
5	Initial price	12.00							
6									
7	**Month**	**Price**							
8	0	12.00							
9	1	11.69	<-- =B8*EXP(Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND()))						
10	2	10.51	<-- =B9*EXP(Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND()))						
11	3	9.97							
12	4	11.82							
13	5	14.05							
14	6	14.42							
15	7	13.40							
16	8	11.50							
17	9	10.27							
18	10	11.52							
19	11	12.59							
20	12	15.46							
21									
22									
23									
24									
25									
26									

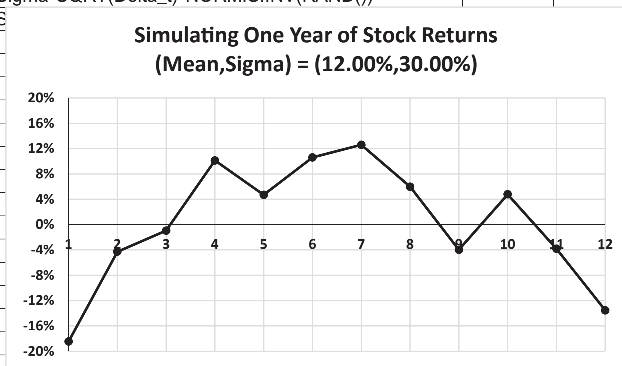


1. In Excel 2003 and 2007, replace this function with **NormSInv(Rand( ))**.

Working with Returns Instead of Prices

If we do this simulation with stock returns, we get the following:

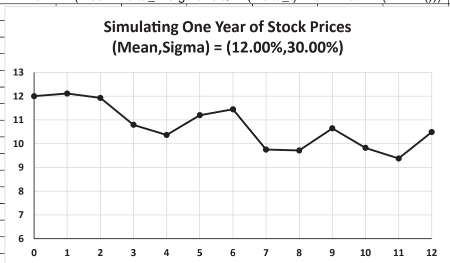
	A	B	C	D	E	F	G	H	I	J	K
1	SIMULATING MONTHLY STOCK RETURN										
2	Mean	12%									
3	Sigma	30%									
4	Delta_t	0.0833	<-- =1/12								
5	Initial price	12.00									
6											
7	**Month**	**Price**									
8	1	-18.47%	<-- =Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND())								
9	2	-4.28%	<-- =Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND())								
10	3	-0.97%	<-- =Mean*Delta_t+S								
11	4	10.12%									
12	5	4.70%									
13	6	10.59%									
14	7	12.59%									
15	8	5.97%									
16	9	-3.98%									
17	10	4.77%									
18	11	-3.81%									
19	12	-13.55%									
20											
21	Total return	3.69%	<-- =SUM(B8:B19)								
22											
23											
24											
25											



Some Return/Price Mathematics

The cumulative return for n months is $n(\mu\Delta t)+\sigma\sqrt{\Delta t}\,(Z_1+Z_2+\ldots+Z_n)$. This means that the expected price at the end of n months is $P_n = P_n\exp[\mu(n\Delta t) + \sigma^2(n\Delta t)/2]$. For 12 months (1 year), the expected price is $P_n = P_n\exp[\mu + \sigma^2/2]$. Of course you will never see exactly this expected price, but if you run the simulation many times, the average ending price will be approximately $P_n = P_n\exp[\mu + \sigma^2/2]$. We show this in the **Data Table** below:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	SIMULATING A SINGLE STOCK: PRICES												
2	Mean	12%									Mean price	13.7851	<-- =AVERAGE(L10:L209)
3	Sigma	30%									Max	26.3859	<-- =MAX(L10:L34)
4	Delta_t	0.0833	<-- =1/12								Min	5.5119	<-- =MIN(L10:L34)
5	Initial price	12.00									Expected ending	14.1527	<-- =B5*EXP(Mean+Sigma^2/2)
6													
7	Month	Price											
8	0	12.00									200 simulations of ending price		
9	1	12.11	<-- =B8*EXP(Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND()))									10.4890	<-- =B20, data table header
10	2	11.92	<-- =B9*EXP(Mean*Delta_t+Sigma*SQRT(Delta_t)*NORM.S.INV(RAND()))								1	15.8501	
11	3	10.79									2	9.6356	
12	4	10.37									3	14.8198	
13	5	11.20									4	9.1409	
14	6	11.45									5	11.1895	
15	7	9.75									6	11.5435	
16	8	9.71									7	14.1382	
17	9	10.65									8	14.5613	
18	10	9.82									9	13.7257	
19	11	9.38									10	16.7483	
20	12	10.49									11	26.3859	
21											12	16.8141	
22											13	18.5837	
23											14	8.6738	
24											15	5.5119	
25											16	16.5237	
26											17	8.9945	



In the above example the **Data Table** simulates the ending price (cell B20). The average ending price for 25 simulations is close to the expected ending price (cell L5).[2]

27.3 Portfolio of Two Stocks

We extend the exercise of the previous section to two stocks with correlation ρ. To simplify matters we will work only in returns, not in prices.

Theory

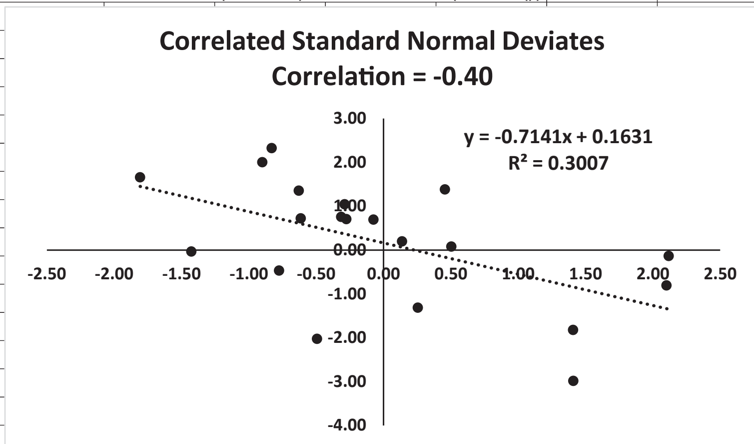
We recall from Chapter 24 that two standard normal deviates Z_1 and Z_2 are correlated with correlation ρ if

$$Z_2 = \rho Z_1 + \sqrt{1-\rho^2}\,Z_3$$

2. Were we to increase the number of simulations to 200, the average (cell L2) would be closer to the expected (cell L5). A statistician would refer to this as a "small sample problem." We prefer to note that 25 simulations of a series of annual returns is—in plain English—the simulation of 25 years of annual returns. This is a lot! Our interpretation: Finance is full of small sample problems, despite the large amount of data available. We return to this topic in section 27.7, where we simulate stock betas.

where Z_1 and Z_3 are both standard normal deviates (produced with the Excel function **Norm.S.Inv(Rand( ))**. We simulate this below:

	A	B	C	D	E	F	G	H	I
1	**SIMULATING TWO CORRELATED NORMAL DEVIATES**								
2	Correlation	-0.4							
3									
4	Z_1	Z_2							
5	-0.4937	-2.0283	<-- =$B$2*A5+SQRT(1-$B$2)*NORM.S.INV(RAND())						
6	-0.8308	2.3217							
7	-0.6305	1.3517							
8	-0.6152	0.7190							
9	-0.0759	0.6951							
10	0.4551	1.3796							
11	0.1376	0.1976							
12	-1.4275	-0.0342							
13	0.5033	0.0785							
14	-1.8080	1.6569							
15	-0.2900	1.0408							
16	-0.2760	0.7029							
17	2.1168	-0.1379							
18	-0.9016	2.0010							
19	-0.3151	0.7570							
20	0.2530	-1.3138							
21	1.4080	-2.9850							
22	-0.7758	-0.4706							
23	2.1012	-0.8090							
24	1.4077	-1.8207							



The simulation shows the regression of Z_2 on Z_1. The anticipated regression intercept should be 0, the anticipated regression slope should be the correlation ρ, and the R^2 should be ρ^2. But because we are simulating random numbers, this will never exactly happen. However, if we run this experiment many times (using **Data Table** on blank cell, section 31.7), we see that we get the approximate result:

	J	K	L	M	N	O
2	**Data table statistics**					
3		Correlation	Intercept	Slope	R-squared	
4	Average	-0.304	0.017	-0.397	0.128	<-- =AVERAGE(N12:N31)
5	Anticipated	-0.400	0.000	-0.400	0.160	<-- =B2^2
6	Max	0.091	0.518	0.102	0.450	<-- =MAX(N12:N31)
7	Min	-0.670	-0.397	-0.847	0.006	<-- =MIN(N12:N31)
8						
9	**Data table: 20 simulations**					
10	Simulation	Correlation	Intercept	Slope	R-squared	
11		-0.5773	0.2955	-0.6293	0.3333	<-- =RSQ(B5:B24,A5:A24), data table header
12	1	-0.2596	0.2801	-0.3636	0.0674	
13	2	-0.2471	-0.1470	-0.3136	0.0610	
14	3	-0.4407	0.0193	-0.4452	0.1942	
15	4	-0.5177	0.1862	-0.4032	0.2681	
16	5	-0.3139	0.2211	-0.7097	0.0986	
17	6	-0.1501	-0.2452	-0.1400	0.0225	
18	7	-0.3100	-0.2214	-0.3576	0.0961	
19	8	-0.2983	0.2574	-0.2799	0.0890	
20	9	-0.3279	0.5184	-0.3672	0.1075	
21	10	-0.3348	-0.2343	-0.7448	0.1121	
22	11	-0.6705	-0.3920	-0.8469	0.4496	
23	12	-0.4252	-0.1556	-0.5615	0.1808	
24	13	-0.3421	0.2272	-0.3944	0.1171	
25	14	-0.1578	0.0280	-0.2270	0.0249	
26	15	-0.2170	0.0656	-0.2197	0.0471	
27	16	0.0796	0.0408	0.1024	0.0063	
28	17	-0.5528	0.3799	-0.8223	0.3055	
29	18	-0.5310	0.0705	-0.6794	0.2820	
30	19	0.0912	-0.3967	0.0966	0.0083	
31	20	-0.1468	-0.1720	-0.2556	0.0215	

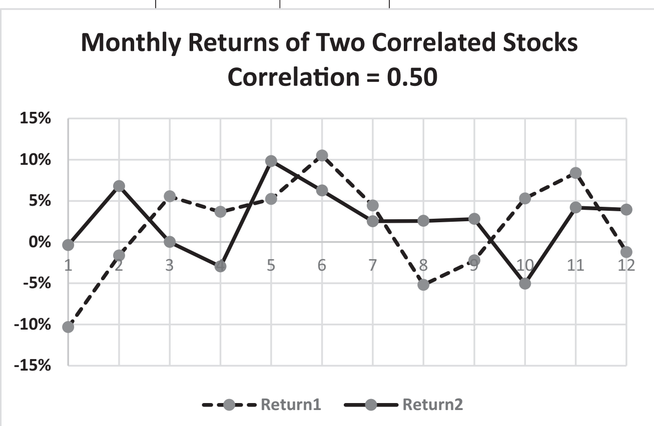
If we run many more iterations of this simulation, we get results closer to the anticipated:

	J	K	L	M	N	O
2	**Data table statistics, 200 simulations**					
3		Correlation	Intercept	Slope	R-squared	
4	Average	-0.322	0.034	-0.417	0.146	<-- =AVERAGE(N12:N211)
5	Anticipated	-0.400	0.000	-0.400	0.160	<-- =B2^2
6	Max	0.270	0.681	0.446	0.561	<-- =MAX(N12:N211)
7	Min	-0.749	-0.782	-1.076	0.000	<-- =MIN(N12:N211)

Simulating Correlated Stock Returns

Below we simulate two correlated normal standard deviates Z_1 and Z_2. We then use these deviates to compute the returns of the two stocks in months 1, 2, … , 12.

	A	B	C	D	E	F	G
1	**PORTFOLIO OF 2 STOCKS: RETURNS**						
2		**Stock1**	**Stock2**				
3	Mean	12%	15%				
4	Sigma	22%	30%		=corr*E10+SQRT(1-		
5	Correlation	0.5			corr^2)*NORM.S.INV(RAND())		
6	Delta_t	0.0833	<-- =1/12				
7							
8	Simulating portfolio returns						
9	Month	Return1	Return2		Z_1	Z_2	
10	1	-10.34%	-0.36%	<--	-1.3095	-0.1863	
11	2	-1.65%	6.79%	=mean2*Delta_t+sigma2*	-0.3064	0.6393	
12	3	5.54%	0.00%	SQRT(Delta_t)*F10	0.5239	-0.1443	
13	4	3.66%	-2.97%		0.3073	-0.4876	
14	5	5.21%	9.83%		0.4866	0.9904	
15	6	10.50%	6.25%		1.0965	0.5768	
16	7	4.44%	2.52%		0.3975	0.1464	
17	8	-5.20%	2.57%		-0.7160	0.1524	
18	9	-2.23%	2.80%		-0.3734	0.1792	
19	10	5.29%	-5.05%		0.4955	-0.7274	
20	11	8.38%	4.18%		0.8522	0.3386	
21	12	-1.22%	3.93%		-0.2560	0.3100	
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							



In a portfolio context:

	A	B	C	D
42	Portfolio Computations			
43	Initial wealth	1,000		
44	Proportion of stock1	25%		
45	Initial investment			
46	Stock1	250.00	<-- =B44*B43	
47	Stock2	750.00	<-- =(1-B44)*B43	
48	Simulated ending value	892.69	<-- =B46*EXP(SUM(B10:B21))+B47*EXP(SUM(C10:C21))	
49				
50	Expected return	14.25%	<-- =B44*mean1+(1-B44)*mean2	
51	Actual return	-11.35%	<-- =LN(B48/B43)	
52	Sigma of return	25.70%	<-- =SQRT(B44^2*sigma1^2+(1-B44)^2*sigma2^2+2*B44*(1-	
53			B44)*corr*sigma1*sigma2)	

27.4 Adding a Risk-Free Asset

We add a risk-free asset to the exercise of the previous section, simulating the performance of an investment in a portfolio of the two risky stocks and the risk free. Cells B13 and B14 below give the risky portfolio expected annual return and annual sigma. Cells B19 and B20 show the expected return and sigma for a portfolio invested 40% in the risky and 60% in the risk-free asset:

	A	B	C	D	E	F	G
1	**PORTFOLIO OF 2 STOCKS AND RISK-FREE**						
2		**Stock1**	**Stock2**				
3	Mean	12%	22%	<-- cell names: mean1, mean2			
4	Sigma	10%	15%	<-- cell names: sigma1, sigma1			
5	Correlation	0.2	<-- cell name: corr				
6	Delta_t	0.0833	<-- =1/12, cell name: delta_t				
7							
8	Risk free rate, r_f	3%	<-- cell name: rf				
9							
10	Risky portfolio						
11	Stock1	30%	<-- cell name: prop1				
12	Stock2	70%	<-- =1-prop1				
13	Expected annual return	19.00%	<-- =prop1*mean1+(1-prop1)*mean2				
14	Sigma annual return	11.48%	<-- =SQRT(prop1^2*sigma1^2+(1-				
15			prop1)^2*sigma2^2+2*prop1*(1-prop1)*corr*sigma1*sigma2)				
16	Investment						
17	In risky portfolio	40%	<-- cell name: prop				
18	In risk free	60%	<-- =1-prop				
19	Expected return	9.40%	<-- =prop*B13+(1-prop)*rf				
20	Sigma return	4.59%	<-- =prop*B14				
21							
22	Month	Portfolio return		Z_1	Z_2		
23	1	1.72%		-0.403	0.884169		
24	2	1.35%		0.335756	0.367992		
25	3	1.73%		-0.733	0.991814		
26	4	-1.41%		1.297981	-2.18324		
27	5	-0.12%		-0.51652	-0.59738		
28	6	0.30%		-1.49158	0.031419		
29	7	2.69%		0.787621	1.349174		
30	8	0.00%		-1.33656	-0.26242		
31	9	2.72%		0.954727	1.325199		
32	10	0.43%		-0.04545	-0.27502		
33	11	-0.37%		-0.14668	-0.9101		
34	12	1.96%		1.653972	0.501086		
35							
36	Annual return						
37	Simulated	11.01%	<-- =SUM(B23:B34)				
38	Expected	9.40%	<-- =prop*(prop1*mean1+(1-prop1)*mean2)+(1-prop)*rf				
39							
40	Sigma of annualized return						
41	Simulated	4.50%	<-- =SQRT(12)*STDEV.S(B23:B34)				
42	Expected	4.59%	<-- =B20				
43							
44	Formulas						
45	Cell B23	=prop*(prop1*(mean1*delta_t+sigma1*SQRT(delta_t)*D23)+(1-					
46		prop1)*(mean2*delta_t+sigma2*SQRT(delta_t)*E23))+(1-prop)*rf*delta_t					

27.5 Multiple Stock Portfolios

So far we've simulated the performance of a two-stock portfolio. When we turn to multiple stocks, we need to make use of the Cholesky decomposition (see section 24.7). We remind you how the Cholesky decomposition works: We want to create a set of normal deviates that has variance-covariance structure *S*:

$$S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2N} \\ & & & \\ \sigma_{N1} & \sigma_{N2} & & \sigma_{NN} \end{bmatrix}$$

As explained in Chapter 24, the steps in doing this are as follows:

1. Create a lower-triangular Cholesky decomposition of *S*. We denote this matrix by *L* and use the VBA function **Cholesky** on the disk with this book to compute the matrix.
2. Create a column vector of *N* standard normal deviates. We use the Excel function **Norm.S.Inv(Rand( ))**.
3. Multiply *L* times the column vector of standard normal deviates.
4. The result is a set of correlated standard normal variates.

Example: Standard Normal Deviates with Given Correlation Structure

Here's an example. We define a VBA function **CorrNormal** that takes as its argument the desired variance-covariance matrix. Below we use this function to simulate 3 years of monthly returns with means specified in cells B3:D3 and variance-covariance in the cells below the means.

	A	B	C	D	E	F	G	H	I	J	K
1	**NORMAL DEVIATES WITH DESIRED MEANS USING THE CHOLESKY DECOMPOSITION** **3 years of simulated monthly data**										
2	Monthly means					Month	3 years of simulated monthly returns				
3	4.00%	3.00%	2.00%	1.00%		1	39.65%	-14.76%	7.37%	-31.14%	<--
4						2	35.08%	24.91%	48.30%	5.50%	{=corrnormal($A$6:$D$9)+
5	Variance-covariance matrix					3	59.32%	-21.07%	55.70%	49.49%	$A$3:$D$3}
6	0.400	0.030	0.020	0.000		4	32.98%	21.50%	22.42%	18.62%	
7	0.030	0.300	0.000	-0.060		5	108.48%	21.66%	61.81%	-18.11%	
8	0.200	0.000	0.200	0.030		6	9.51%	-61.34%	-30.38%	-8.12%	
9	0.000	-0.060	0.030	0.100		7	27.94%	18.99%	-14.77%	-7.86%	
10						8	-111.90%	4.41%	-45.30%	28.64%	
11	Statistics					9	23.50%	-23.32%	30.42%	16.33%	
12	Count	36	<-- =COUNT(G:G)			10	-34.89%	-19.63%	10.32%	33.48%	
13						11	-6.82%	64.53%	-103.76%	-23.03%	
14	Data average vs theoretical mean					12	-85.11%	-19.16%	-84.99%	-3.20%	
15	0.0683	0.0089	0.0092	-0.0410	<-- =AVERAGE(J:J)	13	15.16%	-4.81%	-1.82%	-9.40%	
16	0.0400	0.0300	0.0200	0.0100	<-- Theoretical mean	14	-106.19%	14.36%	-84.03%	-65.22%	
17						15	82.83%	37.21%	31.91%	-14.62%	
18	Data variance vs theoretical variance					16	26.93%	-108.92%	18.73%	37.40%	
19	0.3293	0.2128	0.1769	0.0643	<-- =VAR.S(J:J)	17	45.52%	-22.53%	14.80%	-8.79%	
20	0.4000	0.3000	0.2000	0.1000	<-- Theoretical variance	18	-35.01%	78.25%	-51.82%	-42.48%	
21						19	45.79%	-34.86%	29.89%	22.20%	
22	Sample variance-covariance					20	56.55%	-6.03%	-35.33%	-29.66%	
23	minus the variance-covariance matrix					21	57.91%	-40.91%	11.52%	26.60%	
24	0.0799	0.0861	-0.1291	-0.0332	<-- {=A6:D9-varcovar(G:J)}	22	-9.86%	33.99%	0.02%	-43.59%	
25	0.0861	0.0931	0.0605	0.0045		23	-20.68%	-13.32%	0.28%	-10.90%	
26	0.0509	0.0605	0.0280	-0.0154		24	-54.97%	58.29%	5.19%	-42.82%	
27	-0.0332	0.0045	-0.0154	0.0375		25	66.82%	-5.66%	75.10%	-2.78%	
28						26	-107.69%	21.00%	-11.28%	-23.31%	

If we rerun this simulation for 500 months, the simulation more closely corresponds to the priors:

	A	B	C	D	E	F	G	H	I	J	K
1	**NORMAL DEVIATES WITH DESIRED MEANS USING THE CHOLESKY DECOMPOSITION** **500 months of simulated monthly data**										
2	Monthly means					Month	500 months of simulated monthly returns				
3	4.00%	3.00%	2.00%	1.00%		1	30.37%	54.92%	34.46%	-31.99%	<--
4						2	-39.31%	-55.10%	-19.66%	42.11%	{=corrnormal($A$6:$D$9)+
5	Variance-covariance matrix					3	-12.04%	-106.92%	38.44%	33.99%	$A$3:$D$3}
6	0.400	0.030	0.020	0.000		4	-26.48%	-45.08%	-23.79%	56.90%	
7	0.030	0.300	0.000	-0.060		5	-4.61%	14.74%	3.30%	23.00%	
8	0.200	0.000	0.200	0.030		6	73.97%	-30.09%	50.34%	-3.26%	
9	0.000	-0.060	0.030	0.100		7	-118.50%	-97.71%	-20.31%	7.28%	
10						8	-18.96%	-4.75%	-39.35%	-9.65%	
11	Statistics					9	-44.79%	195.97%	7.85%	13.42%	
12	Count	500	<-- =COUNT(G:G)			10	53.93%	-37.18%	12.84%	-4.62%	
13						11	-73.29%	-7.69%	-28.67%	-40.54%	
14	Data average vs theoretical mean					12	159.96%	-23.86%	103.38%	50.33%	
15	0.0261	-0.0077	0.0318	0.0138	<-- =AVERAGE(J:J)	13	111.39%	-27.33%	64.59%	1.55%	
16	0.0400	0.0300	0.0200	0.0100	<-- Theoretical mean	14	21.98%	29.09%	38.99%	28.19%	
17						15	-31.05%	14.37%	27.04%	22.06%	
18	Data variance vs theoretical variance					16	-30.86%	28.08%	-8.95%	54.89%	
19	0.3189	0.3375	0.1912	0.1034	<-- =VAR.S(J:J)	17	-43.76%	68.92%	-7.72%	-31.58%	
20	0.4000	0.3000	0.2000	0.1000	<-- Theoretical variance	18	-130.46%	60.87%	-79.26%	-58.12%	
21						19	-46.48%	75.66%	25.41%	26.95%	
22	Sample variance-covariance					20	59.00%	42.60%	-3.99%	-0.27%	
23	minus the variance-covariance matrix					21	-6.62%	101.89%	-36.76%	-25.29%	
24	0.0817	0.0379	-0.1485	-0.0129	<-- {=A6:D9-varcovar(G:J)}	22	23.67%	26.17%	-1.09%	-21.08%	
25	0.0379	-0.0368	0.0260	0.0169		23	-23.83%	-30.14%	42.18%	44.61%	
26	0.0315	0.0260	0.0092	-0.0057		24	-53.00%	24.61%	9.55%	15.60%	
27	-0.0129	0.0169	-0.0057	-0.0032		25	5.42%	-48.75%	7.17%	46.58%	

The VBA Function CorrNormal

The VBA function that creates the correlated normal deviates is a combination of two functions. The second of these functions **URandomlist** creates a column vector of standard normal deviates. These deviates are then multiplied times the Cholesky matrix.

```
Function CorrNormal(mat As Range) As Variant
CorrNormal = Application.Transpose(Application.
MMult(Cholesky(mat), _
    urandomlist(mat)))
End Function

Function urandomlist(mat As Range) As Variant
    Application.Volatile
    Dim vector() As Double
    numCols = mat.Columns.Count
    ReDim vector(numCols - 1, 1)
    For i = 1 To numCols
        vector(i - 1, 0) = Application.Norm_S_
    Inv(Rnd)
    Next i
    urandomlist = vector
End Function
```

27.6 Simulating Savings for Pensions

We return to a problem that we discussed in Chapter 1. There we discussed the case of a potential pensioner who wants to save for 5 years in order to enable eight subsequent withdrawals of 30,000. The question we discussed in section 1.6 is the calibration of the annual deposit so that the pension fund (with accumulated 8% annual interest) would be completely depleted after 8 years.

Here's the answer to this problem:

	A	B	C	D	E	F
1	**A RETIREMENT PROBLEM, Section 1.6**					
2	Interest	8%				
3	Annual deposit	29,386.55				
4	Annual retirement withdrawal	30,000.00				
5						=$B$2*(C7+B7)
6	**Year**	**Account balance, beginning of year**	**Deposit at beginning of year**	**Interest earned during year**	**Total in account, end year**	
7	1	0.00	29,386.55	2,350.92	31,737.48	<-- =D7+C7+B7
8	2	31,737.48	29,386.55	4,889.92	66,013.95	
9	3	66,013.95	29,386.55	7,632.04	103,032.54	
10	4	103,032.54	29,386.55	10,593.53	143,012.62	
11	5	143,012.62	29,386.55	13,791.93	186,191.10	
12	6	186,191.10	-30,000.00	12,495.29	168,686.39	
13	7	168,686.39	-30,000.00	11,094.91	149,781.30	
14	8	149,781.30	-30,000.00	9,582.50	129,363.81	
15	9	129,363.81	-30,000.00	7,949.10	107,312.91	
16	10	107,312.91	-30,000.00	6,185.03	83,497.94	
17	11	83,497.94	-30,000.00	4,279.84	57,777.78	
18	12	57,777.78	-30,000.00	2,222.22	30,000.00	
19	13	30,000.00	-30,000.00	0.00	0.00	
20						
21	**Note**: This problem has 5 deposits and 8 annual withdrawals, all made at the beginning of the year. The beginning of year 13 is the last year of the retirement plan; if the annual deposit is correctly computed, the balance at the beginning of year 13 after the withdrawal should be zero.					

In this section we discuss a Monte Carlo variation of this problem. As in Chapter 1, the future pensioner discovers that she has no pension savings and desires to make deposits to fund eight subsequent withdrawals. In this version, however, the savings are invested in a risky portfolio having mean return of 12% and return sigma of 18%. We are curious to see whether a specific level of annual deposits can fund the future planned withdrawals of 30,000 per year. To do this, we examine the bequest—the amount left in the pension plan at the end of year 13. Several things are clear:

- Except for the case where 100% of the savings are invested in the risk-free asset, there is no longer any certainty about the savings or the bequest.
- On average, the larger the proportion in the risky asset, the greater will be the average bequest.

Here's one simulation:

	A	B	C	D	E
1	**A RETIREMENT PROBLEM**				
2	Risk-free rate	8%	<-- Cell name: rf		
3	Risky asset				
4	Mean	12%	<-- Cell name: mean		
5	Sigma	18%	<-- Cell name: sigma		
6	Annual deposit	30,000	<-- Years 1-5		
7	Investment policy				
8	Risk-free	30%	<-- Cell name: prop		
9	Risky asset	70%	<-- =1-prop		
10	Annual retirement withdrawal	30,000	<-- Years 6-13		
11					
12	**Year**	**Account balance, beginning of year**	**Deposit at beginning of year**	**In account at end of year**	
13	1	0	30,000	31,649	<-- =(B13+C13)*(prop*EXP(rf)+(1-
14	2	31,649	30,000	69,767	prop)*EXP((mean+sigma*NORM.S.INV(RAND())
15	3	69,767	30,000	97,717	)))
16	4	97,717	30,000	139,754	
17	5	139,754	30,000	158,609	
18	6	158,609	-30,000	149,860	
19	7	149,860	-30,000	131,279	
20	8	131,279	-30,000	109,145	
21	9	109,145	-30,000	94,169	
22	10	94,169	-30,000	74,561	
23	11	74,561	-30,000	57,691	
24	12	57,691	-30,000	30,722	
25	13	30,722	-30,000	692	

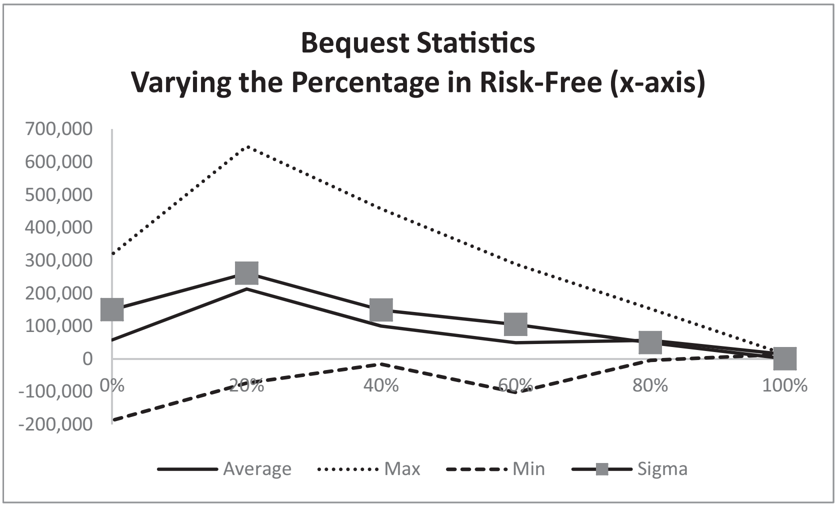
To get a feel for the uncertainty, we run our standard **Data Table** on a blank cell:

	G	H	I
3	Statistics for data table		
4	Average	130,469	<-- =AVERAGE(H15:H24)
5	Max	613,335	<-- =MAX(H15:H24)
6	Min	-36,264	<-- =MIN(H15:H24)
7	Sigma	193,818	<-- =STDEV.S(H15:H24)
8	% positive bequests	90%	<-- =COUNTIF(H15:H24,">0")/10
9			
10			
11			
12	Data table: Simulating the bequest		
13	**Simulation**	**Bequest**	
14		387,776	<-- =D25, data table header
15	1	613,335	
16	2	79,823	
17	3	41,689	
18	4	3,386	
19	5	67,000	
20	6	319,044	
21	7	94,766	
22	8	59,295	
23	9	-36,264	
24	10	62,616	

The average bequest for this specific series of 10 simulations is 130,469 but with considerable variation. Since the simulation is stochastic, each time you open the spreadsheet, you will see different numbers. In 90% of the cases we can count on full funding of the pension (in the sense that the bequest is positive). If we simulate the results varying the percentage invested in the risk-free asset, we get the following:

	G	H	I	J	K	L	M
2		Data table statistics					
3		Percentage in risk-free					
4		100%	80%	60%	40%	20%	0%
5	Average	14,147	31,980	82,341	93,310	126,400	160,835
6	Max	14,147	67,016	269,055	165,297	334,284	890,635
7	Min	14,147	-12,565	-64,989	-37,721	-48,516	-189,115
8	Sigma	0	25,632	94,586	72,617	121,558	308,885
9	% positive bequests	100%	90%	70%	90%	90%	70%
10							
11							
12	Data table: Bequest as function of % in risk-free (columns)						
13	Simulation	Percentage in risk-free					
14	-88,472	100%	80%	60%	40%	20%	0%
15	1	14,147	47,839	53,527	165,297	249,223	-56,263
16	2	14,147	17,554	130,848	132,197	15,894	216,808
17	3	14,147	-12,565	141,673	160,402	138,809	-189,115
18	4	14,147	51,438	-64,989	154,817	71,843	23,543
19	5	14,147	12,938	93,901	10,521	248,790	196,818
20	6	14,147	67,016	-8,049	14,001	334,284	403,491
21	7	14,147	47,580	92,645	98,430	-48,516	890,635
22	8	14,147	8,801	119,013	-37,721	116,709	172,262
23	9	14,147	56,973	-4,217	95,632	9,463	26,016
24	10	14,147	22,223	269,055	139,523	127,503	-75,851

The chart below summarizes these simulations:



27.7 Beta and Return

In this section we simulate a typical beta calculation. We assume that we know the "actual" beta, in the sense that we know the σ_M, σ_i, and the correlation ρ between stock i and the market. We then simulate this beta by creating returns drawn from an appropriate distribution. We use this example to illustrate how far the actual beta, derived from data, can be from the theoretical beta.

Recall from Chapter 11 that for an asset i, β_ι is defined as follows:

$$\beta_i = \frac{Cov(r_i, r_M)}{Var(r_M)}$$

In the simulations below, we use two equivalent expressions for β_ι. First, using the correlation ρ between i and M, we can write β_ι as:

$$\beta_i = \frac{Cov(r_i, r_M)}{Var(r_M)} = \frac{\rho\sigma_i\sigma_M}{\sigma_M^2} = \frac{\rho\sigma_i}{\sigma_M}$$

Second, if we have time-series data $\{r_{it}, r_{Mt}\}$ for the returns on the stock and the market, then we can estimate β_ι by running the regression:

$$r_{it} = \alpha_i + \beta_i r_{Mt}$$

If the data is correlated with correlation ρ, then we expect

$$\alpha_i = E(r_M) - \beta_i E(r_i),\ \beta_i = \frac{\rho\sigma_i}{\sigma_M},\ R^2 = \rho^2$$

In capital market calculations β_ι is typically calculated for monthly returns over a period of 3 to 5 years. Below replicate this procedure by simulating the 60 correlated returns of two assets. We will call the first asset "i" and the second asset "M." We start with some basic data for i and M:

	A	B	C	D	E
2		Mean	Sigma		
3	Stock i	6%	22%		
4	Market	10%	15%		
5	Correlation(i,M)	0.3000			
6	Beta i	0.4400	<-- =rho*sigma/sigma_market		

We now simulate our data:

	A	B	C	D	E	F
1	**SIMULATING BETA AND ALPHA**					
2		Mean	Sigma			
3	Stock i	6%	22%			
4	Market	10%	15%			
5	Correlation(i,M)	0.3000				
6						
7	Anticipated α_i	0.0160	<-- =mu_i-B8*mu_m			
8	Anticipated β_i	0.4400	<-- =rho*sigma_i/sigma_m			
9	Anticipated R^2	0.0900	<-- =rho^2			
10						
11						
12	Regress r_i on r_M					
13	Alpha	0.0086	<-- =INTERCEPT(E20:E79,F20:F79)			
14	Slope	0.4390	<-- =SLOPE(E20:E79,F20:F79)			
15	R-squared	0.0827	<-- =RSQ(E20:E79,F20:F79)			
16						
17	**Simulation**					
18	Month	Normal			Returns	
19		Z_1	Z_2		Stock	Market
20	1	0.6168	-1.4645		9.92%	3.66%
21	2	0.9019	1.1497		11.73%	14.98%
72	53	1.1386	-2.1888		13.23%	0.52%
73	54	-0.4403	1.0415		3.20%	14.51%
74	55	-1.5644	-1.4113		-3.94%	3.89%
75	56	1.2524	0.6107		13.95%	12.64%
76	57	1.3170	0.7016		14.36%	13.04%
77	58	-1.3743	-1.3212		-2.73%	4.28%
78	59	-2.2023	0.2114		-7.99%	10.92%
79	60	-0.1458	-1.1017		5.07%	5.23%
80						
81	Formulas					
82	Cell B20:=NORM.S.INV(RAND())					
83	Cell C20:=rho*B20+SQRT(1-rho^2)*NORM.S.INV(RAND())					
84	Cell E20:=mu_i+sigma_i*SQRT(1/12)*B20					
85	Cell F20:=mu_m+sigma_m*SQRT(1/12)*C20					

The formulas used in the spreadsheet:

- Z_1 = **Norm.S.Inv(Rand( ))**. As discussed in Chapter 24 this creates a standard normal deviate.
- $Z_2 = \rho * Z_1 + \text{norm.s.inv}(\text{rand}(\ )) * \sqrt{1-\rho^2}$. As discussed in Chapter 24 the standard normal deviate Z_2 has correlation ρ with Z_1.
- The column for the stock returns is created by $r_{stock} = \mu_{stock} + \sigma_{stock}\sqrt{1/12} * Z_1$.

• The column for the market returns is created by $r_{market} = \mu_{market} + \sigma_{market}\sqrt{1/12} * Z_2$.

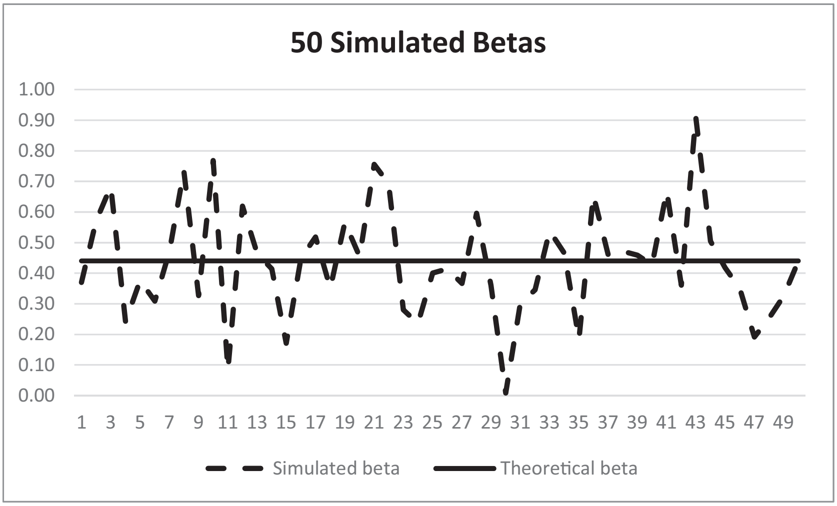
The result is that the market and stock returns are correlated with correlation ρ. We now run a standard first-pass regress of r_i on r_M. For the simulations discussed above, the results of the Monte Carlo simulation are not far from the theoretical results:

	A	B	C	D	E	F
7	Anticipated α_i	0.0160	<-- =mu_i-B8*mu_m			
8	Anticipated β_i	0.4400	<-- =rho*sigma_i/sigma_m			
9	Anticipated R^2	0.0900	<-- =rho^2			
10						
11						
12	Regress r_i on r_M					
13	Alpha	0.0086	<-- =INTERCEPT(E20:E79,F20:F79)			
14	Slope	0.4390	<-- =SLOPE(E20:E79,F20:F79)			
15	R-squared	0.0827	<-- =RSQ(E20:E79,F20:F79)			

Rerunning the simulation shows that not all results will be close to the desired. Below, for example, is another set of simulations in which the Monte Carlo β_i is quite distant from the anticipated β_i:

	A	B	C	D	E	F
7	Anticipated α_i	0.0160	<-- =mu_i-B8*mu_m			
8	Anticipated β_i	0.4400	<-- =rho*sigma_i/sigma_m			
9	Anticipated R^2	0.0900	<-- =rho^2			
10						
11						
12	Regress r_i on r_M					
13	Alpha	-0.0037	<-- =INTERCEPT(E20:E79,F20:F79)			
14	Slope	0.5942	<-- =SLOPE(E20:E79,F20:F79)			
15	R-squared	0.1499	<-- =RSQ(E20:E79,F20:F79)			

If we repeat our experiment 50 times, we see that the computed β_ι exhibits considerable variability:



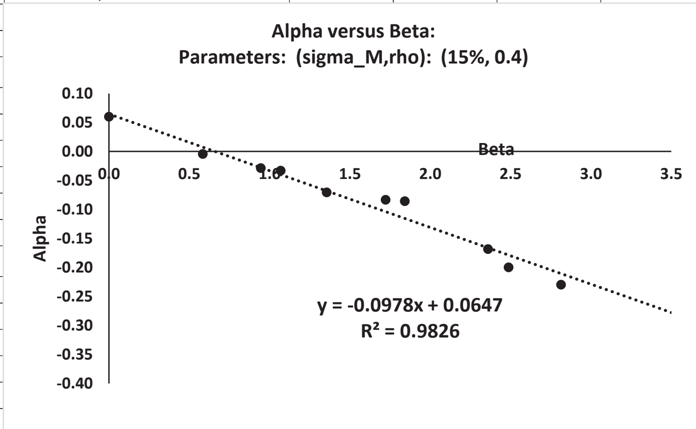
Are Beta and Alpha Related?

There is a growing literature showing that low beta stocks have high alphas and vice versa.[3] We show that this is true in our Monte Carlo simulation. In the spreadsheet below we run a **Data Table** showing beta and alpha as functions of σ_i.[4]

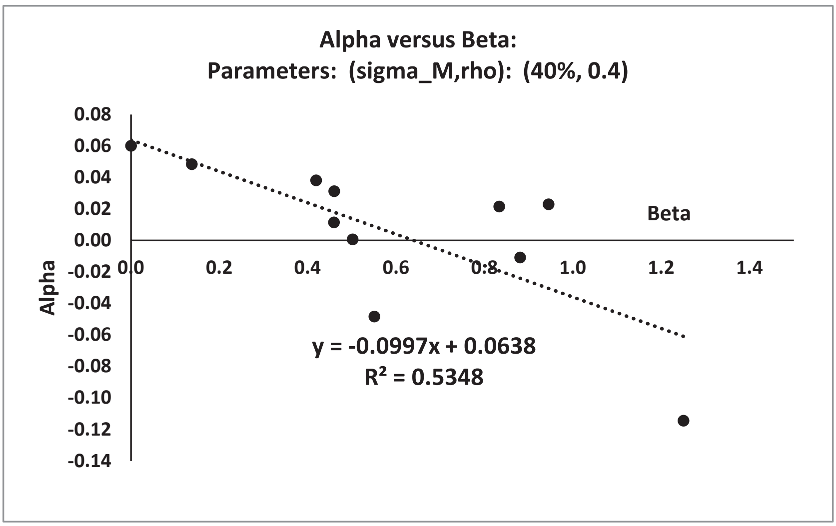
3. For references, see Frazzini-Pedersen (2011), Cremers-Petajisto-Zitzewitz (2010), Hong-Sraer (2012), and Nardin-Haugen (2012).

4. Of course, the higher the σ_I, the higher will be β_ι.

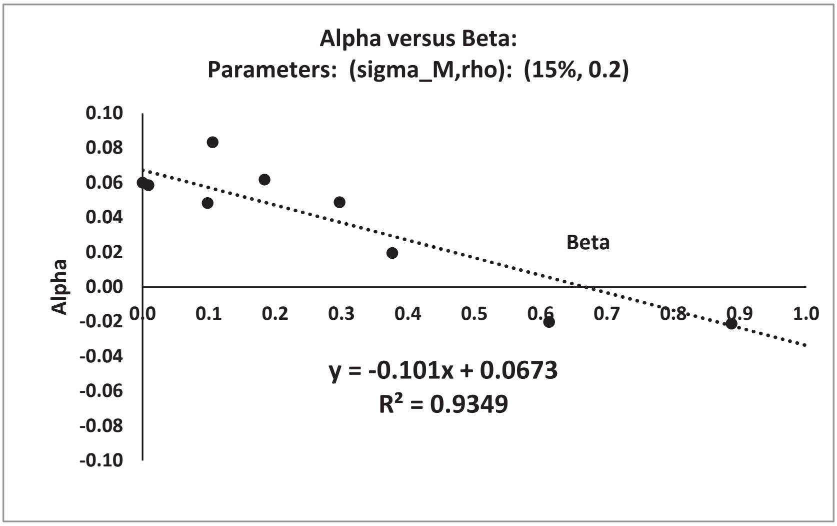
	H	I	J	K	L	M	N	O
4	Intercept	0.0647	<-- =INTERCEPT(J10:J20,I10:I20)					
5	Slope	-0.0978	<-- =SLOPE(J10:J20,I10:I20)					
6	Rsq	0.9826	<-- =RSQ(J10:J20,I10:I20)					
7								
8	Sigma_i	Beta_i	Alpha_i					
9		0.5706	0.0106	<-- =B13, data table header				
10	0%	0.000	0.060					
11	20%	0.584	-0.004					
12	30%	1.068	-0.033					
13	40%	0.944	-0.029					
14	50%	1.354	-0.070					
15	60%	1.841	-0.086					
16	70%	2.487	-0.200					
17	80%	2.813	-0.230					
18	90%	2.359	-0.168					
19	100%	1.722	-0.083					
20	110%	4.252	-0.345					
21								
22								
23								
24								
25								



Here are two more variations. In the first variation we have assumed a much higher σ for the market portfolio:



In the second variation, we change the correlation between the market portfolio and stock *i*:



27.8 Summary

Monte Carlo techniques give insight into investment problems that goes beyond the standard calculations of mean and standard deviation of returns. In this chapter we examine some common cases of asset management: the return on a single stock, on a portfolio of risky assets with a given correlations structure, a standard savings/pension problem with an uncertain investment component, and the computation of asset betas.

Exercises

1. Consider a portfolio of two stocks whose statistical parameters are given below.

   - Stock A: Annual mean return = 15%, annual standard deviation of return = 30%.
   - Stock B: $\mu = 8\%$, $\sigma = 15\%$.
   - Correlation(A,B) = $\rho = 0.3$

An investor with a buy-and-hold strategy buys a portfolio composed of 60% A and 40% B and holds it for 20 years. Simulate the annual returns on the portfolio. A suggested template is given below.

	A	B	C	D	E	F	G	H
1	**INVESTING IN A PORTFOLIO OF TWO STOCKS**							
2		StockA	StockB					
3	Mean	15%	8%					
4	Sigma	30%	15%					
5	Correlation	0.3						
6	Proportion of A	60%						
7								
8	Summary of portfolio returns							
9		Theoretical	Actual					
10	Mean							
11	Sigma							
12								
13		Simulated returns				Normal deviates		
14	Year	A	B					
15	1							
16	2							

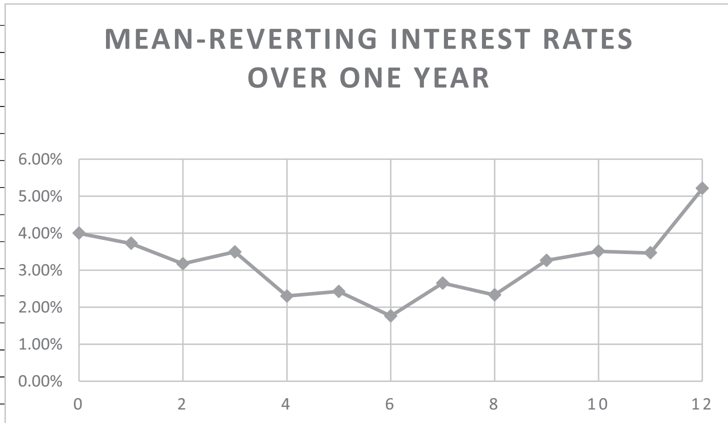
2. Reconsider the problem above. Assume that the risk-free rate is 4% and that the investor (still buy-and-hold) invests in a portfolio composed of 50% risk-free and 50% invested in the 60/40 portfolio of A and B. Compare the theoretical to the simulated returns.

3. The previous example assumes that the risk-free rate is constant. An alternative, perhaps more plausible, model might be to assume that the risk-free rate is mean reverting, with a long-run mean. Under this assumption, if the current rate is above the long-run mean, the next period rate will tend downward, and vice versa. One such model is the Ornstein-Uhlenbeck process:

$$r_t = r_{t-1} + \underbrace{\varphi(\mu - r_{t-1})\Delta t + \sigma\sqrt{\Delta t}Z}_{\substack{\uparrow \\ \text{Innovation in the rate}}}$$

Simulate this process over 12 months:

	A	B	C
1	**ORNSTEIN-UHLENBECK PROCESS FOR INTEREST RATES**		
2	Current rate	4%	
3	Mean, μ	3%	$r_t = r_{t-1} + \underbrace{\varphi(\mu - r_{t-1})\Delta t + \sigma\sqrt{\Delta t}\,Z}_{\text{Innovation in the rate}}$
4	"Pressure", φ	0.10	
5	Sigma, σ	2%	
6	Δt	0.0833	<-- =1/12
7			
8	Month	Rate	
9	0	4.00%	
10	1	3.72%	<-- =B9+$B$4*($B$3-B9)*$B$6+$B$5*SQRT($B$6)*NORM.S.INV(RAND())
11	2	3.18%	
12	3	3.50%	
13	4	2.31%	
14	5	2.43%	
15	6	1.76%	
16	7	2.65%	
17	8	2.33%	
18	9	3.27%	
19	10	3.51%	
20	11	3.47%	
21	12	5.21%	
22			
23			
24			
25			
26			



4. The disk that accompanies this book gives 5 years of monthly price data for five U.S. stocks.

   - Compute the monthly returns for the stocks.
   - Compute the stocks' average monthly returns and standard deviations.
   - Compute the variance-covariance matrix for the stock returns.
   - Compute the correlation matrix of returns for the stocks.
   - Compute the lower-Cholesky matrix for the variance-covariance structure.

5. Using the data from the previous example, simulate 36 months of stock returns assuming the same variance-covariance structure as the historical returns. Notice that it doesn't make sense to assume that the forward-looking expected monthly returns are the same as the historical returns. Instead, use the following values:

	A	B	C	D	E	F
2		Monthly mean and sigma				
3		JCP	AAPL	C	F	K
4	Historical mean	-1.08%	1.46%	-2.02%	2.02%	0.75%
5	Anticipated future mean	2.00%	1.50%	1.00%	2.00%	0.60%

28 Value at Risk (VaR)*

28.1 Overview

Value at risk (VaR) measures the worst expected loss under normal market conditions over a specific time interval at a given confidence level. As one of our references states: "VaR answers the question: how much can I lose with *x%* probability over a pre-set horizon?" (J. P. Morgan, *RiskMetrics—Technical Document*[1]). Another way of expressing this is that VaR is the lowest quantile of the potential losses that can occur within a given portfolio during a specified time period. The basic time period T and the confidence level (the quantile) q are the two major parameters that should be chosen in a way appropriate to the overall goal of risk measurement. The time horizon can differ from a few hours for an active trading desk to a year for a pension fund. When the primary goal is to satisfy external regulatory requirements, such as bank capital requirements, the quantile is typically very small (e.g., 1% of worst outcomes). However, for an internal risk management model used by a company to control the risk exposure, the typical number is around 5% (visit the Internet sites in Selected References for more details). A general introduction to VaR can be found in Linsmeier and Pearson (1996) and in Jorion (1997).

In the jargon of VaR, suppose that a portfolio manager has a daily VaR equal to $1 million at 1%. This statement means that there is only one chance in 100 that a daily loss bigger than $1 million occurs under normal market conditions.

28.2 A Really Simple Example

Suppose a manager has a portfolio that consists of a single asset. The return of the asset is normally distributed with mean return 20% and standard deviation 30%. The value of the portfolio today is $100 million. We want to answer various simple questions about the end-of-year distribution of portfolio value:

1. What is the distribution of the end-of-year portfolio value?

2. What is the probability of a loss of more than $20 million by year-end (i.e., what is the probability that the end-of-year value is less than $80 million)?

---

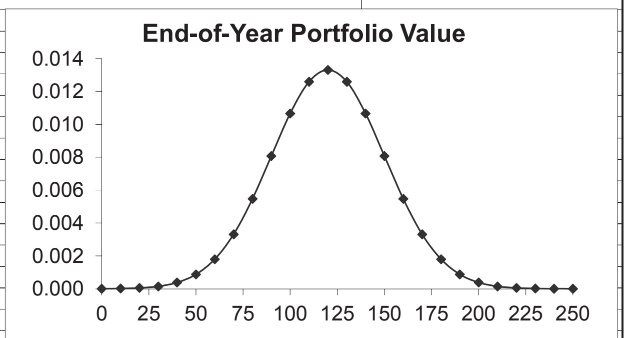
*This chapter is based on an article written with Zvi Wiener, "Value-at-Risk (VaR)," which first appeared in *Mathematica in Education and Research* 7 (1998).

1. This and other valuable documents produced by J. P. Morgan can be found on the disk that accompanies this book.

3. With 1% probability, what is the maximum loss at the end of the year? This is the VaR at 1%.

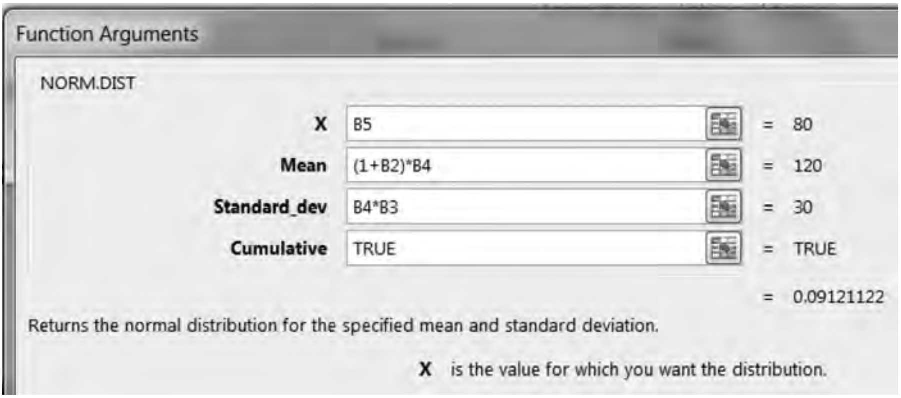
The probability that the end-of-year portfolio value is less than $80 million is about 9%:

	A	B	C	D
1	**PROBABILITY OF YEAR-END PORTFOLIO VALUE**			
2	Mean	20%		
3	Sigma	30%		
4	Initial investment	100		
5	Cutoff	80		
6	Probability that portfolio worth less than cutoff	9.12%	<-- =NORMDIST(B5,(1+B2)*B4,B4*B3,TRUE)	
7				
8	Data table header: =NORMDIST(B5,(1+B2)*B4,B4*B3,FALSE)			
9				
10	**Portfolio value**	0.0055		
11	0	0.0000		
12	10	0.0000		
13	20	0.0001		
14	30	0.0001		
15	40	0.0004		
16	50	0.0009		
17	60	0.0018		
18	70	0.0033		
19	80	0.0055		
20	90	0.0081		
21	100	0.0106		
22	110	0.0126		
23	120	0.0133		
24	130	0.0126		
25	140	0.0106		
26	150	0.0081		
27	160	0.0055		



Excel's **Norm.Dist** function can return both the cumulative distribution and the probability mass function.[2] Here's the way the screen looks when we apply the **Norm.Dist** function in cell B6:

2. In some versions of Excel this function appears without the dot as **NormDist**.

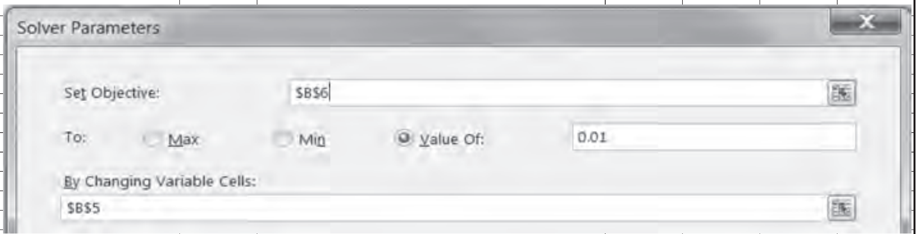


The spreadsheet uses two versions of **Norm.Dist**: First we use the function in cell B6 to determine the probability that the year-end value of the portfolio is less than 80. In this version of the function, we use the value TRUE for the last entry in **Norm.Dist**; when we write =NORMDIST(B5,(1+B2)*B4,B4*B 3,TRUE), **Norm.Dist** returns values of the cumulative normal distribution. In the data table we set this value to FALSE to plot the probability mass function of the year-end portfolio value.

28.3 Defining Quantiles in Excel

By using Excel's **Solver**, we can determine that with a probability of 1%, the end-of-year portfolio value will be less than 50.209. Recall that the value at risk is the worst expected loss under normal market conditions over a specific time interval at a given confidence level. Therefore, the value 50.210 means that the VaR of the portfolio at the 1% level is 100 − 50.210 = 49.790.

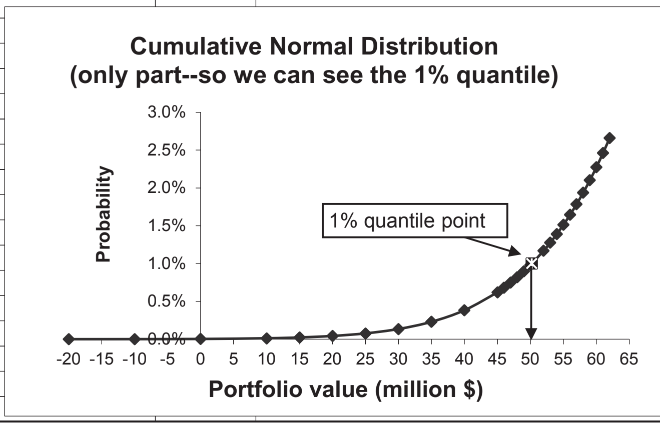
	A	B	C	D	E	F	G
1	**PROBABILITY OF END-YEAR PORTFOLIO VALUE**						
2	Mean	20%					
3	Sigma	30%					
4	Initial investment	100					
5	Cutoff	50.210					
6	Probability that portfolio worth less than cutoff	1.00%	<-- =NORMDIST(B5,(1+B2)*B4,B4*B3,TRUE)				



The cutoff is known as the quantile of the distribution. In Excel it can be determined by using **Solver**, as illustrated above. For two distributions we use—the normal and the lognormal distributions—Excel has built-in functions which find the quintile. These functions—**Norm.Inv**, **Norm.S.Inv**, and **Loginv**—find the inverse for the normal, standard normal, and lognormal distributions.

Here's an example for the numbers given above; this time we have written the function **=NORM.INV(0.01,(1+B3)*B5,B5*B4)** in cell B6. This function finds the cutoff point for which the normal distribution with mean = 120 and standard deviation = 30 has probability of 1%. You can see this point on the graph below, which shows part of the cumulative distribution:

	A	B	C
1	**CALCULATING THE QUANTILES**		
2	Mean	20%	
3	Sigma	30%	
4	Initial investment	100	
5	Cutoff	50.210	<-- =NORMINV(0.01,(1+B2)*B4,B4*B3)
6		1.00%	<-- =NORMDIST(B5,(1+B2)*B4,B4*B3,TRUE)
7			
8	VaR at 1.00% level	49.790	<-- =B5-B6
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			



The Lognormal Distribution

The lognormal distribution is a more reasonable distribution for many asset prices (which cannot become negative) than the normal distribution. Suppose that the return [return = Log(price relative)] on the portfolio is normally distributed with annual mean μ and annual standard deviation σ. Furthermore, suppose that the current value of the portfolio is given by V_0. Then it follows (see Hull, 2011) that the logarithm of the portfolio value at time T, V_T, is normally distributed.[3] This means:

$$\ln(V_T) \sim Normal\left[\ln(V_0) + \left(\mu - \frac{\sigma^2}{2}\right)T, \sigma\sqrt{T}\right]$$

3. John C. Hull, *Options, Futures, and Other Derivatives* (Prentice-Hall, 8th edition, 2011).

Suppose, for example, that $V_0 = 100$, $\mu = 10\%$, $\sigma = 30\%$. Thus the end-of-year log of the portfolio value is distributed normally:

$$\ln(V_1) \sim Normal\left[\ln(100) + \left(0.10 - \frac{0.3^2}{2}\right), 0.3\right] = Normal[4.666017, 0.3]$$

Thus a portfolio whose initial value is \$100 million and whose annual returns are lognormally distributed with parameters μ = 10% and σ = 30%, has an annual VaR equal to \$47.42 million at 1%:

	A	B	C
1	**QUANTILES FOR LOGNORMAL DISTRIBUTION**		
2	Initial value, V_0	100	
3	Mean, μ	10%	
4	Sigma, σ	30%	
5	Time period, T	1	<-- in years
6			
7	**Parameters of normal distribution of ln(V_T)**		
8	Mean	4.6602	<-- =LN(B2)+(B3-B4^2/2)*B5
9	Sigma	0.3000	<-- =B4*B5
10			
11	Cutoff	52.576	<-- =LOGINV(0.01,B8,B9)
12	VaR at 1% level	47.424	<-- =B2-B11

Most VaR calculations are not concerned with annual value at risk. The main regulatory and management concern is with loss of portfolio value over a much shorter time period (typically several days or perhaps weeks). It is clear that the distribution formula $\ln(V_T) \sim Normal\left[\ln(V_0) + \left(\mu - \frac{\sigma^2}{2}\right)T, \sigma\sqrt{T}\right]$ can be used to calculate the VaR over any horizon. Recall that T is measured in annual terms; if there are 250 business days in a year, then the daily VaR corresponds to T = 1/250 (for many fixed income instruments, one should use 1/360, 1/365, or 1/365.25, depending on the market convention):

28.4 A Three-Asset Problem: The Importance of the Variance-Covariance Matrix

As can be seen from the preceding examples, VaR is not—in principle, at least—a very complicated concept. In the implementation of VaR, however, there are two big practical problems (both problems are discussed in much greater detail in the material available on the RiskMetrics website, www.msci.com/resources/):

1. The first problem is the estimation of the parameters of asset return distributions. In "real world" applications of VaR, it is necessary to estimate means, variances, and correlations of returns. This is a not-inconsiderable problem! In this section we illustrate the importance of the correlations between asset returns. In the following section we give a highly simplified example of the estimation of return distributions from market data. For example, you can imagine that a long position in euros and a short position in U.S. dollars are less risky than a position in only one of the currencies, because of a high probability that profits of one position will be mainly offset by losses of another.

2. The second problem is the actual calculation of position sizes. A large financial institution may have thousands of loans outstanding. The database of these loans may not classify them by their riskiness, nor even by their term to maturity. Or—to give a second example—a bank may have offsetting positions in foreign currencies at different branches in different locations. A long position in euros in New York may be offset by a short position in euros in Geneva; the bank's risk—which we intend to measure by VaR—is based on the net position.

We start with the problem of correlations between asset returns. We continue the previous example, but assume that there are three risky assets. As before, the parameters of the distributions of the asset returns are known: all the means, μ_1, μ_2, μ_3, as well as the variance-covariance matrix of the returns:

$$S = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}$$

The matrix S is of course symmetric; σ_{ii} is the variance of the ith asset's return, and σ_{ij} is the covariance of the returns of assets i and j (if $i = j$, σ_{ij} is the variance of asset i's return).

Suppose that the total portfolio value today is $100 million, with $30 million invested in asset 1, $25 million in asset 2, and $45 million in asset 3. Then the return distribution of the portfolio is given by:

$$Mean\ return = x_1\mu_1 + x_2\mu_2 + x_3\mu_3$$

$$Variance\ of\ return = \{x_1, x_2, x_3\}.S.\{x_1, x_2, x_3\}^T$$

where $x = \{x_1, x_2, x_3\} = \{0.3, 0.25, 0.45\}$ is the vector of proportions invested in each of the three assets. Assuming that the returns are normally distributed (meaning that prices are lognormally distributed), we may calculate the VaR as in the following spreadsheet:

	A	B	C	D	E	F	G	H
1	**VaR FOR 3 ASSET PROBLEM**							
2		**Mean returns**		**Variance-covariance matrix**				**Portfolio proportions**
3	Asset 1	10%		0.10	0.04	0.03		0.30
4	Asset 2	12%		0.04	0.20	-0.04		0.25
5	Asset 3	13%		0.03	-0.04	0.60		0.45
6								
7	Initial investment	100						
8	Mean return	0.1185	<-- {=MMULT(TRANSPOSE(B3:B5),H3:H5)}					
9	Portfolio sigma	0.3848	<-- {=SQRT(MMULT(MMULT(TRANSPOSE(H3:H5),D3:F5),H3:H5))}					
10								
11	Mean investment value	111.8500						
12	Sigma of investment value	38.4838						
13								
14	Cutoff	22.3234	<-- =NORMINV(0.01,(1+B8)*B7,B9*B7)					
15	Cumulative PDF	0.01	<-- =NORMDIST(B14,B11,B12,TRUE)					
16	VaR at 1.00% level	77.6766	<-- =B7-B14					
17								
18		Note that the functions in cells B8 and B9 are array functions: You must press [Ctrl]+[Shift]+[Enter] after you write the function in the cell. The curly brackets {} are not written--they appear automatically.						

28.5 Simulating Data: Bootstrapping

Sometimes it helps to simulate data. In this section we give an example. We suppose that the current date is 10 February 1997, and we consider a firm which has an investment in two assets:

- It is long two units of an index fund. The fund's current market price is 293, so that the investment in the index fund is worth 2 * 293 = 586.
- It is short a foreign bond denominated in rubles. The bond is a zero-coupon bond (i.e., pays no interest), has face value of 100 rubles and maturity of 8 May 2000. If the current ruble interest rate is 5.30%, then the 10 February 1997 ruble value of the bond is

$$-100 * \exp[-5.30\% * (\text{May } 8, 2000 - \text{Feb. } 10, 1997)/365] = -84.2166$$

In dollars, the value of the bond is −84.2166*3.40 = −286.3365, so that the net portfolio value is 586 − 286.3365 = 299.66.

This is illustrated below:

	A	B	C	D	E	F	G	H	I
1	**BOOTSTRAPPING DATA—INITIAL POSITION**								
2	Units of Index held	2							
3	Bond maturity	8-May-00							
4									
5	**Date**	**Index value**	**Ruble interest rate**	**Ruble exchange rate**		**Total index value**	**Ruble bond value**	**Dollar bond value**	**Portfolio value**
6	10-02-97	293	5.30%	3.40		586.00	-84.2166	-286.336	299.66
7									
8						=B2*B6			
9								=G6*D6	=F6+H6
10									
11						=-100*EXP(-(B3-A6)/365*C6)			

Now suppose we have exchange rate and index data. We illustrate data for 40 days (the middle of the data has been hidden, but you will see that the rows go from 6 to 45):

	A	B	C	D	E	F
1	**EXCHANGE RATE AND INDEX DATA**					
2	Units of Index held	2				
3	Bond maturity	8-May-00				
4						
5	**Day**	**Index**	**Foreign interest rate**	**Exchange rate**		**Portfolio value**
6	02-01-97	462.71	5.28%	3.50		632.13
7	03-01-97	514.71	5.26%	3.47		738.41
8	04-01-97	456.5	5.23%	3.46		622.49
9	05-01-97	487.39	5.24%	3.45		685.17
10	06-01-97	470.42	5.25%	3.45		651.28
43	08-02-97	467.14	5.31%	3.44		644.75
44	09-02-97	562.06	5.32%	3.41		837.17
45	10-02-97	481.61	5.30%	3.40		676.88

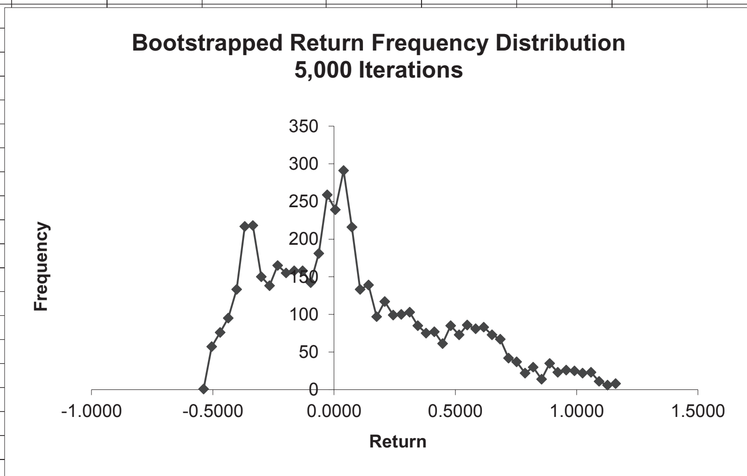
We want to use these data as a basis for generating "random" return data. We illustrate one technique for doing this which is called "bootstrapping": This refers to random reshufflings of the data. For each iteration, we reorder the series of index prices, interest rates, and exchange rates and calculate the return on the portfolio.[4]

	A	B	C	D	E	F	G	H
1	BOOTSTRAPPING RETURN DISTRIBUTIONS							
2	Units of Index held	2			Iterations	5,000	Start time	11:45:50
3	Bond maturity	8-May-00			Return	0.15	Elapsed	0:16:41
4	Number of data points	40						
5							=H46/H7-1	
6	Day	Index	Index rand	Foreign interest rate	Interest rand	Exchange rate	Exchange rand	Portfolio value
7	02-01-97	615.93	0.0029	5.31%	0.0148	3.40	0.0202	947.24
8	03-01-97	757.02	0.0447	5.24%	0.0179	3.41	0.0456	1,227.87
9	04-01-97	581.50	0.0452	5.32%	0.0377	3.44	0.0620	875.04
10	05-01-97	651.99	0.0742	5.28%	0.0383	3.42	0.0846	1,017.27
11	06-01-97	605.37	0.1027	5.28%	0.0634	3.50	0.1070	917.28
12	07-01-97	514.71	0.1455	5.28%	0.0640	3.43	0.1321	741.79
13	08-01-97	640.43	0.1574	5.28%	0.0652	3.48	0.1522	988.99
14	09-01-97	645.50	0.2020	5.25%	0.0789	3.43	0.1532	1,003.00
15	10-01-97	450.91	0.2049	5.34%	0.0884	3.46	0.1994	612.12
16	11-01-97	475.49	0.2075	5.26%	0.1111	3.46	0.2074	660.47
17	12-01-97	654.17	0.3184	5.36%	0.3611	3.42	0.2156	1,022.10
18	13-01-97	445.77	0.3308	5.31%	0.3662	3.37	0.2309	608.98
19	14-01-97	669.12	0.3799	5.28%	0.4016	3.44	0.2428	1,049.48
20	15-01-97	500.71	0.3878	5.31%	0.4112	3.44	0.2469	712.65
21	16-01-97	705.27	0.3951	5.35%	0.4387	3.46	0.2963	1,120.69
22	17-01-97	533.40	0.4201	5.28%	0.4603	3.46	0.3266	776.23
23	18-01-97	639.95	0.4465	5.32%	0.4751	3.42	0.3454	993.03
24	19-01-97	444.27	0.4551	5.30%	0.4763	3.39	0.4183	603.96
25	20-01-97	670.63	0.4654	5.25%	0.4797	3.45	0.5154	1,051.12
26	21-01-97	470.42	0.4655	5.25%	0.4952	3.42	0.5357	653.18
27	22-01-97	458.26	0.5114	5.26%	0.5059	3.45	0.5883	626.39
28	23-01-97	466.45	0.5386	5.27%	0.5217	3.47	0.6197	641.14
29	24-01-97	462.71	0.5456	5.27%	0.5596	3.48	0.6813	632.78
30	25-01-97	459.27	0.5682	5.24%	0.5798	3.42	0.7240	630.62
31	26-01-97	740.74	0.6100	5.23%	0.6026	3.41	0.7321	1,194.27
32	27-01-97	790.82	0.6245	5.28%	0.6068	3.41	0.7507	1,294.86
33	28-01-97	487.39	0.6405	5.34%	0.6384	3.44	0.7987	686.00
34	29-01-97	456.50	0.6795	5.29%	0.6583	3.52	0.8033	616.98
35	30-01-97	467.14	0.6922	5.24%	0.6601	3.40	0.8047	647.84
36	31-01-97	481.61	0.7349	5.27%	0.6699	3.41	0.8337	676.18
37	01-02-97	544.75	0.7357	5.26%	0.6836	3.42	0.8501	801.48
38	02-02-97	453.69	0.7423	5.24%	0.7388	3.68	0.8512	597.22
39	03-02-97	786.16	0.7436	5.27%	0.7867	3.49	0.8553	1,278.42
40	04-02-97	561.88	0.7944	5.29%	0.8107	3.41	0.8797	836.74
41	05-02-97	562.06	0.9345	5.32%	0.8328	3.45	0.8811	833.97
42	06-02-97	472.35	0.9353	5.23%	0.8759	3.42	0.9045	656.19
43	07-02-97	636.02	0.9406	5.35%	0.8899	3.42	0.9336	984.61
44	08-02-97	461.79	0.9630	5.30%	0.9238	3.49	0.9662	629.75
45	09-02-97	584.41	0.9688	5.26%	0.9403	3.47	0.9878	876.25
46	10-02-97	687.33	0.9713	5.25%	0.9585	3.41	0.9990	1,087.02

4. The bootstrapping technique is illustrated in the appendix to this chapter.

The distribution of the bootstrapped return data looks like this:

	J	K	L	M	N	O	P	Q	R	S
1	**Bootstrapped Return Frequency Distribution 5,000 Iterations**									
2	Max return	1.16080	<-- =MAX(J:J)		=K3					
3	Min return	-0.53821	<-- =MIN(J:J)							
4										
5						=($K$2-$K$3)/50+L7				
6	**Output**		**Bins**		**Cumulative percent**					
7	0.2859		-0.5382	1	0.0%					
8	-0.1820		-0.5042	57	1.1%					
9	0.0954		-0.4703	76	1.5%					
10	0.4311									
11	-0.0504									
12	0.1755									
13	0.0008									
14	-0.2790									
15	-0.3847									
16	-0.3961									
17	-0.1576									
18	0.0378									
19	0.5969									
20	0.2791									
21	-0.3907									
22	0.5339									
23	-0.0546									
24	-0.3956									
25	-0.4083									
26	0.0810									
27	0.2932									
28	0.1866									
29	-0.0121									
30	-0.3674		0.2433	99	2.0%					
31	-0.2599		0.2773	100	2.0%					
32	-0.4493		0.3113	103	2.1%					
33	0.1674		0.3453	85	1.7%					



The graph indicates the return distribution, which is far from normal. From columns L, M, and N, you can tell that the 1% VaR is about –50%, meaning, with a probability of 1%, the firm could lose 50% of its investment.

How Did We Produce the Bootstrapped Data?

Bootstrapping basically consists of reshuffling the data randomly, and then viewing each reshuffle as a point in a distribution. In the spreadsheet on page 732, the columns C, E, and G contain random numbers. The VBA program below contains three **For** loops which insert three columns of random numbers into the spreadsheet. On the spreadsheet for this chapter, this program can be run through the shortcut [Ctrl] + a.

Having inserted the random numbers, the spreadsheet then uses Excel's **Sort** function to sort the index prices (column B), the foreign interest rates (column D), and the exchange rate (column F). This produces random combinations of the three portfolio pricing factors, which give the resulting portfolio values in column H and the portfolio return in cell F3.

```
'My thanks to Marek Jochec for cleaning
'up this code!
Sub randomizeit()
   Range("starttime") = Time
   Range("J7:J15000").ClearContents
   Application.ScreenUpdating = False

   For Iteration = 1 To Range("iterations")
   For Row = 1 To 40
   Range("IndexRand").Cells(Row, 1) = Rnd
   Next Row

   For Row = 1 To 40
   Range("InterestRand").Cells(Row, 1) = Rnd
   Next Row

   For Row = 1 To 40
   Range("ExchangeRand").Cells(Row, 1) = Rnd
   Next Row
```

```
    Range("B7:C46").Sort Key1:=Range("C6"), _
    Order1:=xlAscending, Header:=xlNo
    Range("D7:E46").Sort Key1:=Range("E6"), _
    Order1:=xlAscending, Header:=xlNo
    Range("F7:G46").Sort Key1:=Range("G6"), _
    Order1:=xlAscending, Header:=xlNo

    Range("returndata").Cells(Iteration, 1) = _
    Range("meanreturn")
    Next Iteration

    Range("elapsed") = Time - Range("starttime")
End Sub
```

Having produced the bootstrapped data, we use the array function **Frequency** (see Chapter 34) to produce a distribution of the simulated data. Notice that this simulation takes a very long time! On the author's laptop 5,000 simulations took almost 17 minutes.

Monte Carlo Simulations

In this section we return to the three-asset problem discussed in section 28.5. Instead of doing statistical analysis, we run a simulated set of returns. In the spreadsheet below, we simulate (columns G–J) 30 days of daily returns on a portfolio of three assets. The annual mean returns of the assets are very pessimistic (A3:C3). For the portfolio (25%, 50%, 25%) the cumulative 30-day simulated return for the specific simulation is –5.40% (cells B19 and J34).

When we run 1,000 simulations for these 30 days in a data table, we find that in 16 cases (cell B24) the cumulative return is beyond −10%. This is the VaR at a 10% level:

	A	B	C	D	E	F	G	H	I	J
1	**VaR: SIMULATING PORTFOLIO PERFORMANCE**									
2	Annual means						30 days of simulated returns			
3	-20%	-11%	-13%				Daily asset returns			**Cumulative**
4							Asset1	Asset2	Asset3	**portfolio**
5	Daily means					1	0.60%	0.63%	-0.16%	0.34%
6	-0.08%	-0.04%	-0.05%	<-- =C3/250		2	0.53%	0.40%	-0.08%	0.57%
7						3	-0.73%	-0.60%	-0.05%	0.00%
8	Variance-covariance matrix					4	0.07%	0.35%	-0.08%	0.09%
9	7.170E-05	5.075E-05	-9.038E-06			5	-0.51%	-0.33%	-0.03%	-0.29%
10	5.075E-05	4.070E-05	-5.990E-06			6	-0.02%	-0.02%	0.02%	-0.38%
11	-9.038E-06	-5.990E-06	2.800E-06			7	-1.55%	-0.67%	0.23%	-1.12%
12						8	0.11%	0.42%	0.17%	-0.92%
13	Portfolio					9	0.96%	0.57%	-0.07%	-0.49%
14	25%	50%	25%			10	-1.21%	-1.03%	0.16%	-1.35%
15						11	-0.67%	-0.53%	-0.21%	-1.91%
16	Simulated portfolio returns					12	-1.38%	-1.39%	0.18%	-2.99%
17	Mean	-2.76%	<-- =AVERAGE(J:J)			13	-0.09%	-0.12%	-0.06%	-3.17%
18	Sigma	1.95%	<-- =STDEV.S(J:J)			14	-0.13%	-0.14%	-0.09%	-3.37%
19	Cumulative	-5.40%	<-- =J34			15	-0.33%	0.17%	0.00%	-3.45%
20						16	0.90%	0.65%	-0.17%	-3.02%
21						17	0.20%	0.57%	-0.11%	-2.80%
22	**Data table results**					18	-0.10%	0.16%	-0.07%	-2.84%
23	Min	-12.12%	<-- =MIN(B29:B1028)			19	-0.97%	-0.67%	0.08%	-3.48%
24	Risk	16	<-- =COUNTIF(B29:B1028,"<-10%")			20	-0.13%	-0.34%	-0.07%	-3.78%
25	Probability	1.60%	<-- =B24/1000			21	-0.79%	-0.47%	0.24%	-4.23%
26						22	-0.15%	-0.49%	-0.06%	-4.61%
27	**Data table: 1000 simulations**					23	-0.10%	0.00%	-0.25%	-4.78%
28		-5.40%	<-- =J34, data table header			24	-0.27%	-0.04%	0.16%	-4.91%
29	1	-5.07%				25	0.89%	0.55%	-0.24%	-4.55%
30	2	-0.64%				26	-0.21%	-0.45%	-0.10%	-4.94%
31	3	-3.15%				27	0.06%	0.06%	-0.15%	-5.01%
32	4	-1.04%				28	-0.03%	-0.01%	-0.08%	-5.12%
33	5	-5.66%				29	0.07%	0.12%	0.29%	-5.05%
34	6	-3.25%				30	-0.48%	-0.31%	0.01%	-5.40%
35	7	-6.55%								
36	8	-4.17%								

Appendix: How to Bootstrap: Making a Bingo Card in Excel

Bootstrapping refers to a technique of random shuffling of data to create more "data." This appendix gives a simple illustration of bootstrapping. It is based on the "birthday bingo" game created for Helen Benninga's 85th birthday. The game goes like this:

• Everyone gets a "Helen Bingo Card," which has five columns of five numbers each. The first column has five numbers from 1 to 17, the second column has five numbers between 18 and 34, and so on. So a typical card looks like this:

Helen's
85th Birthday
bingo game!!!

H	E	L	E	N
3	23	51	52	75
15	26	40	57	70
9	21	50	68	82
7	22	49	56	71
8	20	45	55	69

• We made up 85 questions with answers from 1 to 85. When a card with a question was drawn, someone had to give the correct answer, and then everyone who had the number on his or her card could cross it out. For example, if we asked, "How many grandchildren does Helen have?" and someone answered "13," then everyone with a 13 in the first column could cross it out.

• The first person with five numbers in a line (a column, a row, or a diagonal) won the prize. (Note that it didn't take any talent to win—all you had to do was hear the right answers.)

We wanted to use Excel to create the cards, but it wasn't initially clear how to go about this. Finally, the requisite trick, which is that we want to model the selection of balls from an urn without replacement, was discovered. (We will discuss this topic in greater detail later.)

The Trick

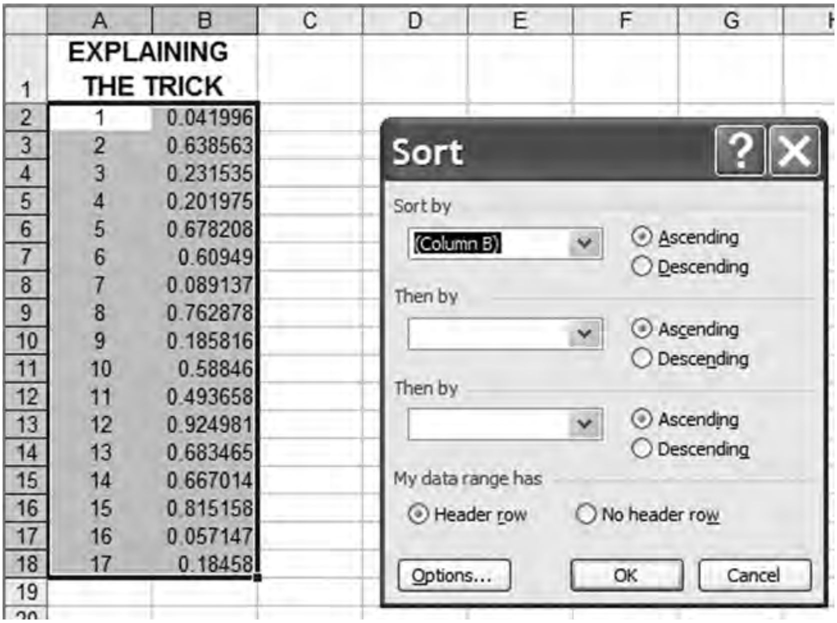
The trick is very simple. As an illustration, suppose we want to make a random draw of five numbers between 1 and 17 (these will be the five numbers that will appear in the first column of a particular Helen Bingo card). Here's how we go about this:

• First create a list of numbers from 1 to 17 and an adjoining column of random numbers. This will give something that looks like the following:

	A	B
1	EXPLAINING THE TRICK	
2	1	0.653152
3	2	0.425876
4	3	0.743173
5	4	0.911709
6	5	0.104356
7	6	0.09228
8	7	0.49608
9	8	0.210725
10	9	0.740506
11	10	0.724376
12	11	0.310175
13	12	0.437225
14	13	0.197224
15	14	0.145462
16	15	0.797405
17	16	0.52166
18	17	0.438188

The list of numbers was itself created in two stages: In the first stage **=Rand**( ) was entered into each of the cells B2:B18. In the second stage B2:B18 were copied and were then pasted special back into their locations using **Edit|Paste Special|Values**. This procedure gets rid of the formulas behind the numbers (else **Rand**( ) will change its values every time we hit [Enter]).

• Next, sort both columns using the second column as a sorting key. To do this, first mark off the relevant data, and then use the Excel command **Data|Sort**. This will bring up the following screen, in which I've chosen to sort the data by Column C.



	A	B
1	EXPLAINING THE TRICK	
2	1	0.041996
3	2	0.638563
4	3	0.231535
5	4	0.201975
6	5	0.678208
7	6	0.60949
8	7	0.089137
9	8	0.762878
10	9	0.185816
11	10	0.58846
12	11	0.493658
13	12	0.924981
14	13	0.683465
15	14	0.667014
16	15	0.815158
17	16	0.057147
18	17	0.18458

• In this case, the **Sort** command will give:

	A	B
1	EXPLAINING THE TRICK	
2	1	0.041996
3	16	0.057147
4	7	0.089137
5	17	0.18458
6	9	0.185816
7	4	0.201975
8	3	0.231535
9	11	0.493658
10	10	0.58846
11	6	0.60949
12	2	0.638563
13	14	0.667014
14	5	0.678208
15	13	0.683465
16	8	0.762878
17	15	0.815158
18	12	0.924981

- Finally, pick the first five numbers from the first column (in this example: 1, 16, 7, 17, 9). You could, of course, equally well pick the last five, the middle five, or any other five numbers from the column.

The Probabilistic Model

What we're doing here is just like picking random numbers out of an urn *without replacement*. This model, standard in all introductory probability books, imagines an urn filled with balls. Each ball has a different number—in our case, there are 17 balls with numbers between 1 and 17. The urn is shaken to mix up the balls, and then five balls are drawn out. Each ball, once drawn, is not placed back in the urn.

This is somewhat different from the standard random number generators, which pick random numbers with replacement (i.e., once the ball's number is recorded, it is placed back in the urn, so that it could possibly be drawn again).[5]

Writing a VBA Program

The next obvious step was to write a program in VBA to automate the procedure. The spreadsheet is given below.

5. Excel has a function **Randbetween(low,high)** which lets you create random integers between **low** and **high**. Thus, to create five numbers between 1 and 17, you just copy **=Randbetween(1,17)** into five adjacent cells. However, this is like drawing numbers from the urn with replacement, and hence can give you multiple draws of the same number—and this is a bingo no-no!

	B	C	D	E	F	G	H	I
1		Helen's						
2		85th Birthday					Ctrl + b runs the macro	
3		bingo game!!!						
4								
5		H	E	L	E	N		
6		2	20	47	63	73		
7		15	18	39	68	78		
8		5	31	43	58	80		
9		11	19	44	65	69		
10		17	26	51	64	76		
11								

The code which produced this spreadsheet is given below.[6]

```
Public Const NperR = 17
Public Const BingoRows = 5
Public Const BingoColumns = 5
Public Const BingoCard = "C6:G10"

Option Base 1
```

6. I thank Paul Legerer for vastly improving the code for this program from the previous edition of *Financial Modeling*. An astute reader will note that Paul's program internalizes the sorting of the random numbers in the VBA code, so that only the printing of the card is done in the spreadsheet.

```
Sub DoIt()
'loop 5 time (1 loop for each
'column on the bingo card)

For iii = 1 To BingoColumns
    Dim ArraySort(NperR, 2)
    For i = 1 To NperR

'first dimension of the array:
'random number between 0 and 1
    ArraySort(i, 1) = Rnd

'second dimension of the array:
'position in the array (1-17 in the first loop,
'18 to 34 in the second loop, etc…)
    ArraySort(i, 2) = i + (iii - 1) * NperR
    Next i

    For ii = 1 To NperR

'look for the minimum value in the array
'and keep also the value of the position
'(1 to 17)
    MinNum = ArraySort(ii, 1)
    MinIndex = ArraySort(ii, 2)
    RealIndex = ii
    For i = ii To NperR
        If ArraySort(i, 1) < MinNum Then
        MinNum = ArraySort(i, 1)
        MinIndex = ArraySort(i, 2)
        RealIndex = i
        End If

     Next i
```

```
'Replace the first number in the array by the
'minimum value and…
    TempNum = ArraySort(ii, 1)
    TempIndex = ArraySort(ii, 2)
    ArraySort(ii, 1) = MinNum
    ArraySort(ii, 2) = MinIndex
    ArraySort(RealIndex, 1) = TempNum
    ArraySort(RealIndex, 2) = TempIndex

'start again with the remaining numbers: once
'the last loop is completed, all numbers are
sorted
    Next ii

'write the first 5 numbers (number of rows on
the
'bingo card) of the results into the
spreadsheet
    With ActiveSheet.Range(BingoCard)
        For ii = 1 To BingoRows
             .Cells(ii, iii) = ArraySort(ii, 2)
        Next ii
        End With
    Next iii
End Sub
```

Another Way to Do the Bingo Cards[7]

There's another way to design the bingo cards, using the Excel **Rank** function:

7. I thank A. C. M. de Bakker for the suggestion in this subsection.

	A	B	C	D	E	F	G
1							
2		**H**	**E**	**L**	**E**	**N**	
3		1	31	36	63	83	<-- =(F$9-1)*17+RANK(F10,F$10:F$26)
4		3	27	38	58	84	<-- =(F$9-1)*17+RANK(F11,F$10:F$26)
5		11	32	44	55	78	
6		2	23	35	68	75	
7		6	19	45	61	76	
8							
9		1	2	3	4	5	
10		0.9562	0.2730	0.8788	0.2574	0.0735	<-- =RAND()
11		0.8333	0.5145	0.7366	0.5556	0.0702	<-- =RAND()
12		0.4827	0.2727	0.4318	0.8825	0.5355	<-- =RAND()
13		0.8475	0.6533	0.9783	0.0103	0.6092	
14		0.7706	0.9582	0.3832	0.3485	0.6019	
15		0.4284	0.6652	0.2587	0.6039	0.9998	
16		0.2066	0.3386	0.0672	0.0924	0.5632	
17		0.6735	0.1244	0.3091	0.8998	0.8532	
18		0.2589	0.6032	0.0847	0.4665	0.4054	
19		0.6858	0.9602	0.4834	0.6650	0.8638	
20		0.3187	0.6513	0.2867	0.8896	0.0098	
21		0.3953	0.1872	0.3402	0.1919	0.7563	
22		0.8029	0.3386	0.5020	0.2635	0.8696	
23		0.8241	0.5994	0.7240	0.0538	0.9613	
24		0.5427	0.9184	0.7793	0.4888	0.2180	
25		0.7136	0.4916	0.4389	0.9486	0.1391	
26		0.0453	0.8367	0.5422	0.2381	0.5272	

The entries in rows 10–26 are created with **Rand( )**. The entries of the bingo card are computed with the formula:

$$= \underbrace{(F\$9\text{-}1)*17}_{\substack{\uparrow \\ \text{Guarantees that all} \\ \text{numbers in column} \\ \text{F are more than 68} \\ \text{(and likewise for other} \\ \text{columns)}}} + \underbrace{RANK(F10, F\$10 : F\$26)}_{\substack{\uparrow \\ \text{What is the rank of F10 among F10 : F26?} \\ \text{When copied down one cell:} \\ \text{What is the rank of F11 among F10 : F26?} \\ \text{etc.}}}$$

The only disadvantage of this very clever implementation of bootstrapping is that the entries on the card change with each recomputation of the spreadsheet.

29 Simulating Options and Option Strategies

29.1 Overview

In this chapter we simulate options and option strategies. We start by showing how an option on a stock can be replicated by a dynamic portfolio of the stock and a bond. We go on to apply this approach to portfolio insurance (a combination of a put and stock) and to a butterfly. Our approach has its roots in the Black-Scholes formula. This formula can be interpreted as showing that an option is a portfolio of a position in the underlying asset and a position in the risk-free asset, where both positions are adjusted dynamically over time. Remarkably, the Black-Scholes formula shows that if the adjustment process is continuous, then the dynamic strategy is self-financing: It requires no additional cash flows after the establishment of the initial portfolio.

In a more realistic situation it is of course impossible to replicate the option strategy portfolio continuously. We have to compromise by making only periodic adjustments, and this forces us to consider first how to define the strategy over time, and second how well the compromise strategy performs in replicating the option strategy. We will show several approaches to this problem.

Background: Price Simulations and the Black-Scholes Formula

Recall from Chapter 26 that the way to simulate a stock price is to simulate $S_t = S_{t-1} * \exp\left[\mu\Delta t + \sigma\sqrt{\Delta t}Z\right]$, where Z is a draw from a standard normal distribution. In Excel this formula becomes:[1]

$$S_t = S_{t-1} * \exp\left[\mu\Delta t + \sigma\sqrt{\Delta t} * \text{Norm.S.Inv}(\text{Rand}(\ ))\right]$$

We will use this formula throughout the chapter to simulate stock prices.

Recall also that the Black-Scholes formula, explained in Chapter 17, states that the prices of a European call and put are given by:

$$Call = SN(d_1) - Xe^{-rT}N(d_2)$$
$$Put = -SN(-d_1) + Xe^{-rT}N(-d_2)$$
$$d_1 = \frac{\ln\left(S/X\right) + \left(r + \sigma^2/2\right)T}{\sigma\sqrt{T}},\ d_2 = d_1 - \sigma\sqrt{T}$$

1. The Excel function **Norm.S.Inv** is equivalent to the older version, **Normsinv**. Both functions are part of the newer Excel editions.

In this chapter our interpretation of this formula is that an option can be replicated by a long or short position in the underlying stock and a long or short position in the risk-free asset (jargon: "the bond"):

	Stock position	Bond position
$Call = SN(d_1)$ $-Xe^{-rT}N(d_2)$	$S_t N(d_1)$ Long position in stock	$-Xe^{-t*(T-t)}N(d_2)$ Short position in bond
$Put = -SN(-d_1)$ $+ Xe^{-rT}N(-d_2)$	$-S_t N(-d_1)$ Short position in stock	$Xe^{-t*(T-t)}N(-d_2)$ Long position in bond

It follows that if we replicate an option with a portfolio of stocks and bonds, this portfolio needs to be rebalanced often. A remarkable fact, proved by Black and Scholes in their pathbreaking 1973 paper, is that if the rebalancing is continuous, then the changes in the stock and bond positions exactly offset each other. This zero-investment property is also termed "self-financing" and is a hallmark of the Black-Scholes replicating portfolio.

While we cannot prove this fact, we can give some intuition. In the spreadsheet below, we look at two very close times. At time $t = 0$, we price a call option on a stock whose current price is $S_0 = 50$. The option has exercise price $X = 50$, and expires at $T = 0.5$. The replicating portfolio for the option is 28.9698 in the stock and –24.2745 in the bond, giving a call option value of 4.6952 (cell B19).

At time $\Delta t = 1/250$ (roughly 1 day later), a randomly generated stock price is 49.8708. The replicating portfolio is now 28.6400 in the stock and –24.0399 in the bond. To compute the investment required to reach this portfolio, we have to revalue the previous investments in the stock and the bond. The previous stock position has grown in value to $28.9698 * \frac{49.8708}{50} = 28.8949$ and the previous bond position now has value of $-24.2745 * e^{r*\Delta t} = -24.2784$ (1 day's interest has been added). This means that we must sell off some of the stock: (28.6400 – 28.8949) = 0.2548. Thus the cash flow from the change in the stock position is positive. In the bond position we want to change the debt from –24.2784 to –24.0399; this means that we require cash of 0.2385 to pay off some of the debt. The net cash flow from these changes is +0.2549 – 0.2386 = 0.0163. In a spreadsheet:

	A	B	C	D	E	F	G
1	**BLACK-SCHOLES AS A PORTFOLIO**						
2	Δt	0.0040	<-- =1/250				
3							
4		**Time = 0**	**Time = Δt**				
5	S	50.0000	49.8708				
6	X	50.0000	50.0000				
7	r	4.00%	4.00%				
8	T, time to option maturity	0.5000	0.4960	<-- =B8-B2			
9	Sigma	30%	30%				
10							
11	d_1	0.2003	0.1873		=E18-C18		
12	d_2	-0.0118	-0.0240				
13						Cash flow from position change	
14	$N(d_1)$	0.5794	0.5743	=B17*C5/B5	Value of previous position		=E17-C17
15	$N(d_2)$	0.4953	0.4904				
16							
17	Stock position, $S*N(d_1)$	28.9698	28.6401	<-- =C5*C14	28.8949	0.2548	<-- Δstock
18	Bond position, $-X*N(d_2)$	-24.2745	-24.0399	<-- =-C6*EXP(-C8*C7)*C15	-24.2784	-0.2385	<-- Δbond
19	Value of call option	4.6952	4.6002	<-- =SUM(C17:C18)		0.0163	
20							
21				=B18*EXP(B7*B2)			**=SUM(F17:F18) net cash flow from position change**
22							
23							
24							

The Excel notebook with this chapter gives a dynamic spreadsheet for the above example that changes values with each press of **F9**. Using this spreadsheet you can confirm:

- The position changes rarely completely net out.
- The smaller the Δt, the smaller the net cash flow will be from the position change. Black-Scholes proved that in the limit, this net cash flow is always zero, and that the strategy is thus completely self-financing.

29.2 Imperfect but Cashless Replication of a Call Option

To perfectly replicate a call option using the Black-Scholes formula will require continuous trading. Suppose we compromise a bit on strictly following the BS formula. In what follows we use the strategy of replication, but force it to be self-financing. We do this as follows:

- At time 0, we set the initial portfolio equal to that prescribed by the BS formula: $Stock = S_0 N(d_1)$, $Bond = -Xe^{-rT}N(d_2)$

• At times $t > 0$, we set the stock position $= S_t N(d_1)$, but we set the bond position equal to the cash flow from the change in the stock position:

$$Bond_{t+\Delta t} = \underbrace{Bond_t * e^{r\Delta t}}_{\substack{t+\Delta t \text{ value} \\ \text{of time } t \text{ bond} \\ \text{position}}} + \underbrace{S_t N(d_{1,t}) * \frac{S_{t+\Delta t}}{S_t}}_{\substack{t+\Delta t \text{ value} \\ \text{of time } t \\ \text{stock position}}} - \underbrace{S_{t+\Delta t} N(d_{1,t+\Delta t})}_{\substack{t+\Delta t \text{ desired} \\ \text{stock position}}}$$

$$= Bond_t * e^{r\Delta t} + S_{t+\Delta t}\{N(d_{1,t}) - N(d_{1,t+\Delta t})\}$$

Here's the way this looks. At $t = 0$ the replicating portfolio is exactly equivalent to the Black-Scholes (cells H13:I13), but at $t > 0$, the Black-Scholes value is different from the portfolio value (though quite close).

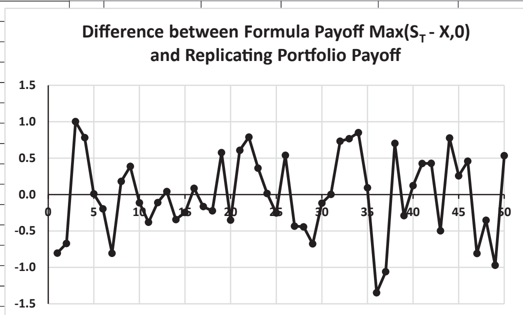
	A	B	C	D	E	F	G	H	I
1	**IMPERFECT BUT CASHLESS REPLICATION OF CALL OPTION WITH PORTFOLIO** Requires no investment over time but doesn't perfectly match Black-Scholes								
2	S_0, stock price	50							
3	X, exercise price	50							
4	T, strike date	0.5							
5	Mean stock return (mu)	12%							
6	Sigma of stock return	30%					=E13*EXP(interest*Delta_t)+B14*(NORM.S.DIST(done(B13,exercise,C13,interest,Sigma),1)-NORM.S.DIST(done(B14,exercise,C14,interest,Sigma),1))		
7	r, interest rate	4%							
8	Delta_t	0.0192	<-- =1/52						
9									
10	=B14*NORM.S.DIST(done(B14,exercise,C14,interest,Sigma),1)								
11				**Replicating portfolio**			**Comparing BS to portfolio value**		
12	**Week**	**Stock price**	**Time remaining**	**Stock**	**Bond**		**Portfolio**	**BS**	**Portfolio - BS**
13	0	50.0000	0.5000	28.9698	-24.2745		4.6952	4.6952	0.0000
14	1	51.0829	0.4808	31.5541	-26.2502		5.3039	5.2435	0.0605
15	2	47.5675	0.4615	22.7932	-19.6809		3.1123	3.2099	-0.0977
16	3	43.8959	0.4423	14.1025	-12.7647		1.3378	1.6521	-0.3143

Here's the result in terms of final payoffs:

	A	B	C	D	E	F	G	H	I
34	21	44.5525	0.0962	5.5524	-5.4820		0.0705	0.2463	-0.1759
35	22	45.1987	0.0769	5.7963	-5.6495		0.1468	0.2324	-0.0856
36	23	48.2980	0.0577	16.4192	-15.8793		0.5399	0.7571	-0.2172
37	24	45.4011	0.0385	2.5675	-3.0246		-0.4571	0.0630	-0.5201
38	25	48.4558	0.0192	11.5024	-11.7891		-0.2867	0.2752	-0.5619
39	26	50.0476	0.0000						
40									
41	Final payoff								
42	Perfectly replicated option	0.0476	<-- =MAX(B39-exercise,0)						
43	Investment portfolio payoff	0.0821	<-- =D38*B39/B38+EXP(interest*Delta_t)*E38						

We run this simulation 50 times and compare the payoffs on the perfect replication with those of the investment portfolio. We use the technique from Chapter 31, "**Data Table** on a blank cell," to run the simulation. Our conclusion: By and large, the imperfect replication strategy works ok.

	A	B	C	D	E	F	G	H	I
46	**Data table: running 50 replications**								
47	**Simulation**	**Perfect replication**	**Portfolio**	**Difference: Perfect - portfolio**					
48									
49	1	12.8857	13.6891	-0.8034		**Statistics for 50 simulations**			
50	2	20.2031	20.8742	-0.6711		Mean difference	-0.0024	<--	=AVERAGE(D49:D98)
51	3	4.9478	3.9451	1.0026		Sigma	0.5662	<--	=STDEV.S(D49:D98)
52	4	20.0870	19.3043	0.7827		Max difference	1.0026	<--	=MAX(D49:D98)
53	5	5.8534	5.8424	0.0109		Min difference	-1.3473	<--	=MIN(D49:D98)
54	6	6.3475	6.5438	-0.1962					
55	7	0.0000	0.8066	-0.8066					
56	8	0.0000	-0.1808	0.1808					
57	9	42.8039	42.4171	0.3868					
58	10	8.5893	8.7009	-0.1116					
59	11	0.0000	0.3803	-0.3803					
60	12	14.9803	15.0901	-0.1098					
61	13	0.0000	-0.0419	0.0419					
62	14	0.0000	0.3429	-0.3429					
63	15	0.0000	0.2457	-0.2457					
64	16	15.3211	15.2346	0.0865					
65	17	5.0699	5.2351	-0.1652					
66	18	0.0000	0.2243	-0.2243					
67	19	40.9947	40.4187	0.5760					
68	20	0.0000	0.3497	-0.3497					
69	21	7.7107	7.1019	0.6088					
70	22	0.0000	-0.7894	0.7894					
71	23	5.8724	5.5092	0.3632					



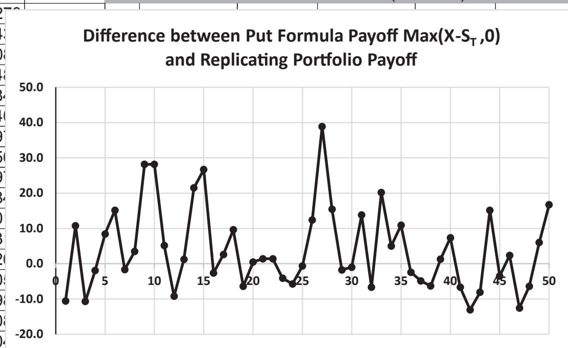
Without too much explanation, we can do the same for a put. Redoing the mathematics for a put:

$$Bond_{t+\Delta t} = \underbrace{Bond_t * e^{r\Delta t}}_{\substack{\uparrow \\ t+\Delta t \text{ value} \\ \text{of time } t \text{ bond} \\ \text{position}}} \underbrace{- S_t N(-d_{1,t}) * \frac{S_{t+\Delta t}}{S_t}}_{\substack{\uparrow \\ t+\Delta t \text{ value} \\ \text{of time } t \\ \text{stock position}}} + \underbrace{S_{t+\Delta t} N(-d_{1,t+\Delta t})}_{\substack{\uparrow \\ t+\Delta t \text{ desired} \\ \text{stock position}}}$$

$$= Bond_t * e^{r\Delta t} + S_{t+\Delta t}\{-N(-d_{1,t}) + N(-d_{1,t+\Delta t})\}$$

Putting this in a simulation and running the simulation 50 times:

	A	B	C	D	E	F	G	H	I
46	Data table: running 50 replications								
47	Simulation	Perfect replication	Portfolio	Difference: Perfect - portfolio					
48									
49	1	0.0000	10.5672	-10.5672		Statistics for 50 simulations			
50	2	11.3925	0.6324	10.7602		Mean difference		4.0640	<-- =AVERAGE(D49:D98)
51	3	0.0000	10.6771	-10.6771		Sigma		11.7531	<-- =STDEV.S(D49:D98)
52	4	0.0000	1.9068	-1.9068		Max difference		38.8730	<-- =MAX(D49:D98)
53	5	7.4425	-0.9831	8.4256		Min difference		-13.0531	<-- =MIN(D49:D98)
54	6	15.3637	0.2359	15.12					
55	7	0.0000	1.6436	-1.64					
56	8	3.7379	0.2299	3.50					
57	9	28.2190	0.0705	28.14					
58	10	27.3503	-0.8341	28.18					
59	11	5.1821	0.0421	5.14					
60	12	0.0000	9.1971	-9.19					
61	13	1.1650	-0.0918	1.25					
62	14	21.2614	-0.2358	21.49					
63	15	26.5251	-0.1598	26.68					
64	16	0.0000	2.6018	-2.60					
65	17	2.8785	0.2471	2.63					
66	18	9.2148	-0.4114	9.62					
67	19	0.0000	6.4053	-6.40					
68	20	1.4479	0.9499	0.49					
69	21	0.9231	-0.4849	1.40					
70	22	2.6633	1.2584	1.40					



29.3 Simulating Portfolio Insurance

Options can be used to guarantee minimum returns from stock investments. As we showed in our discussion of option strategies in Chapter 15, when you purchase a stock (or a portfolio of stocks) and simultaneously purchase a put on the stock (on the portfolio), you are assured that the strategy payoff will never be lower than the exercise price on the put:

$$Stock + Put = S_T + \max(X - S_T, 0)$$
$$= \begin{cases} S_T & \text{if } S_T > X \\ X & \text{if } S_T \le X \end{cases}$$

It is not always possible to find marketed puts in all portfolios; in this case the Black-Scholes option pricing formula can show us how to replicate a put by a dynamic strategy in which the investment in a risky asset (be it a single stock or a portfolio) and the investment in riskless bonds changes over time to mimic the returns of a put option. Such replication strategies are at the heart of the portfolio insurance strategies discussed here.

We start by considering the following simple example: You decide to invest in one share of General Pills stock, which currently costs $56. The stock pays no dividends. You hope for a large capital gain at the end of the year, but you worry that the stock's price may decline. To guard against a decline in the stock's price, you decide to purchase a European put on the stock. The put you purchase allows you to sell the stock at the end of 1 year for $50. The cost of the put, $2.38, is derived from the Black-Scholes model (see Chapter 17) using the following data: S_0 = $56, X = $50, σ = 30%, and r = 8%:

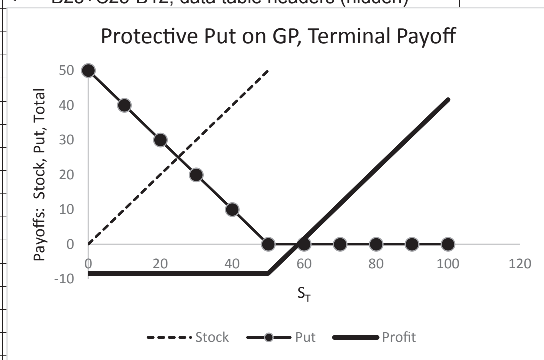
	A	B	C	D	E
2	S_0	56.00			
3	X	50.00			
4	T	1			
5	r	8.00%			
6	Sigma	30%			
7	Put price	2.38	<-- =bsput(B2,B3,B4,B5,B6)		

This *protective put* or *portfolio insurance* strategy guarantees that you will lose no more than $6 on your share of General Pills stock. If the stock's price at the end of the year is more than $50, you will simply let the put expire without exercising it. However, if the stock's price at the end of the year is less than $50, you will exercise the put and collect $50. It is as if you had purchased an insurance policy on the stock with a $6 deductible.

Of course this protection doesn't come for free: Instead of investing $56 in your single share of stock, you have invested $58.38. You could have deposited the additional $2.38 in the bank and earned interest of 8%*$2.38 = $0.19 in the course of the year; alternatively, you could have used the $2.38 to buy more shares.

To see how this strategy works, we perform sensitivity analysis of the strategy profit as a function of the terminal price of the stock S_T:

	A	B	C	D	E	F	G	H	I	J
9	**PORTFOLIO INSURANCE STRATEGY**									
10	Buy stock	56.00	<-- =B2							
11	Buy put	2.38	<-- =B7							
12	Total cost	58.38	<-- =B10+B11							
13										
14	Payoff at date T									
15	S_T	35.00								
16	Put payoff	15.00	<-- =MAX(B3-B15,0)							
17	Total payoff	50.00	<-- =SUM(B15:B16)							
18	Profit	-8.38	<-- =B17-B12							
19										
20	**Data table: Total strategy profit as function of ST**									
21										
22	**S_T**	**Stock**	**Put**	**Profit**						
23					<-- =B23+C23-B12, data table headers (hidden)					
24	0	0	50	-8.37698						
25	10	10	40	-8.37698						
26	20	20	30	-8.37698						
27	30	30	20	-8.37698						
28	40	40	10	-8.37698						
29	50	50	0	-8.37698						
30	60	60	0	1.623024						
31	70	70	0	11.62302						
32	80	80	0	21.62302						
33	90	90	0	31.62302						
34	100	100	0	41.62302						
35										
36										
37										
38										
39										



Portfolio Insurance When There Are No Traded Puts

In the example above, we have implemented a portfolio insurance strategy by purchasing a put whose underlying asset exactly corresponds to our share portfolio. But this technique may not always be possible:

- It could be that there is no traded put option on the shares we wish to insure.
- It could also be that we want to purchase portfolio insurance on a more complicated basket of assets, such as a portfolio of shares. Puts on portfolios do exist (for example, there are traded puts on the S&P 100 and S&P 500 portfolios), but there are no traded puts on most portfolios.

It is here that the Black-Scholes option pricing model comes to our aid. From this formula it follows that a put option on a stock (from here on, "stock" will be used to refer to a portfolio of stocks as well as a single stock) is simply a portfolio consisting of a short position in the stock and a long position in the risk-free asset, with both positions being adjusted continuously. For example, consider the Black-Scholes formula for a put with expiration date $T = 1$ and exercise price X . At time t, $0 \le t < 1$, the put has value

$$P_t = -S_t N(-d_1) + Xe^{-r(1-t)} N(-d_2)$$

$$d_1 = \frac{\ln\left(S_t / X\right) + \left(r + \sigma^2/2\right)(1-t)}{\sigma\sqrt{1-t}}, \; d_2 = d_1 - \sigma\sqrt{1-t}$$

where

$1-t$ is time remaining to maturity

S_t is the price of the stock at time t

Thus, buying a put is equivalent to investing $Xe^{-r(1-t)}N(-d_2)$ in a risk-free bond which matures at time 1 and investing $-S_tN(-d_1)$ in the stock. Since the investment in the stock is negative, this means that a put is equivalent to a short position in the stock and a long position in the risk-free asset.

The total investment required to buy one share of the stock plus a put on the stock is $S_t + P_t$. Writing this out and substituting in the Black-Scholes put formula gives

$$\begin{aligned} \textit{Total investment, protective put} &= S_t + P_t \\ &= S_t - S_t N(-d_1) + Xe^{-r(1-t)} N(-d_2) \\ &= S_t(1 - N(-d_1)) + Xe^{-r(1-t)} N(-d_2) \\ &= S_t N(d_1) + Xe^{-r(1-t)} N(-d_2) \end{aligned}$$

where the last equality uses the fact that for the standard normal distribution $N(x) + N(-x) = 1$. Another way of looking at this problem is to regard the total investment $S_t + P_t$ at time t as a portfolio of a stock and a bond; we can then ask, What is the proportion ω_t of this portfolio invested in the stock at time t? Rewriting the formula above in terms of portfolio proportions gives

$$\begin{aligned} \textit{Proportion invested in stock} &= \omega_t \\ &= \frac{S_t N(d_1)}{S_t N(d_1) + X e^{-r(1-t)} N(-d_2)} \\ \textit{Proportion invested in risk free asset} &= 1 - \omega_t \\ &= \frac{X e^{-r(1-t)} N(-d_2)}{S_t N(d_1) + X e^{-r(1-t)} N(-d_2)} \end{aligned}$$

To sum up: If you want to buy a specific portfolio of assets *and* an insurance policy guaranteeing that at $t = 1$ your total investment will not be worth less than X, then at each point in time, t, you should invest a proportion ω_t of your wealth in the specific portfolio you have chosen, and a proportion $1 - \omega_t$ in riskless, pure discount bonds that mature at $t = 1$. The Black-Scholes put pricing formula can be used to determine these proportions.

An Example

Suppose you decide to invest $1,000 in General Pills stock (currently selling at $56) and in protective puts on the shares with an exercise price of $50 and an expiration date 1 year from now. This ensures that your dollar value per share at the end of 1 year will be no less than $50. Suppose that there is no traded put on General Pills, so that you will have to create your own put by investing in the share and in riskless discount bonds. The riskless rate of interest is 8%, and the standard deviation of General Pills' log return is 30%.

We will construct a series of portfolios which implements this strategy on a week-by-week basis. Based on our discussion from the previous section, we know that this replication strategy will not be perfect. We simulate it to see how it works.

Week 0: At the beginning of this week, the initial investment in shares of General Pills should be:

$$\omega_0 = \frac{S_0 N(d_1)}{S_0 + P_0} = \frac{56 * 0.7865}{56 + 2.38} = 75.45\%$$

with the remaining proportion, $1 - \omega_0 = 24.55\%$, invested in riskless discount bonds maturing in 1 year. If traded European puts on GP existed and if these puts had an exercise price of 50 with an exercise date 1 year from now, these would be trading at \$2.38. Your strategy would consist of buying 17.13 shares of GP (cost = \$959.23) and 17.13 puts (cost = \$40.77). Buying \$754.40 worth of shares and \$245.60 worth of bonds exactly duplicates the initial investment in 17.13 shares and 17.13 puts. This equivalence is guaranteed by Black and Scholes. The calculations of the option price and the appropriate portfolio proportions are shown in the spreadsheet picture below:

	A	B	C
1	**Black-Scholes Option Pricing Formula Applied to General Pills Put**		
2	S	56	Stock price
3	X	50	Exercise price
4	T	1	Time remaining
5	r	8.00%	Risk-free rate of interest
6	Sigma	30%	Stock volatility
7	Put price	2.38	<-- call price - S + X*Exp(-r*T): by Put-Call parity
8			
9	**Calculating the portfolio insurance proportions**		
10	Omega	75.45%	<-- =B2*NORM.S.DIST(done(B2,B3,B4,B5,B6),1)/(B2+B7), proportion in shares
11	1-omega	24.55%	<-- =1-B10, proportion in bonds

You started at $t = 0$ with an initial investment of \$1,000; now suppose that by the beginning of the next week ($t = 1/52 = 0.0192$) the price of GP shares increased to \$60. Here is your updated portfolio at the beginning of week 1.

You can see that when the stock price increases, the portfolio proportion of the stock increases:

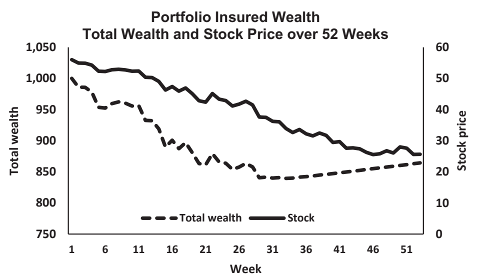
	A	B	C
1	**Updating the Portfolio Insurance Proportions**		
2	Previous stock price S_t	56.00	
3	Previous time to maturity	1.00	
4	Time interval, Δt	0.0192	<-- =1/52
5	New time to maturity	0.9808	<-- =B3-B4
6			
7	Previous portfolio		
8	Stock	754.50	
9	Bonds	245.50	
10	Current stock price $S_{t+\Delta t}$	60.00	Stock price
11			
12	Current portfolio before readjustment		
13	Stock	808.39	<-- =B8*B10/B2
14	Bonds	245.88	<-- =B9*EXP(B22*B4)
15	Total	1,054.27	<-- =SUM(B13:B14)
16			
17	**Calculating the portfolio insurance proportions**		
18	Proportion of stock Omega, ω	82.53%	<-- =B10*NORM.S.DIST(done(B10,B21,B5,B22,B23), 1)/(B10+bsput(B10,B21,B5,B22,B23))
19	1-omega	17.47%	<-- =1-B18

If, conversely the stock price at time Δt is 52, the proportion of the stock will decrease and the proportion of the risk-free will increase:

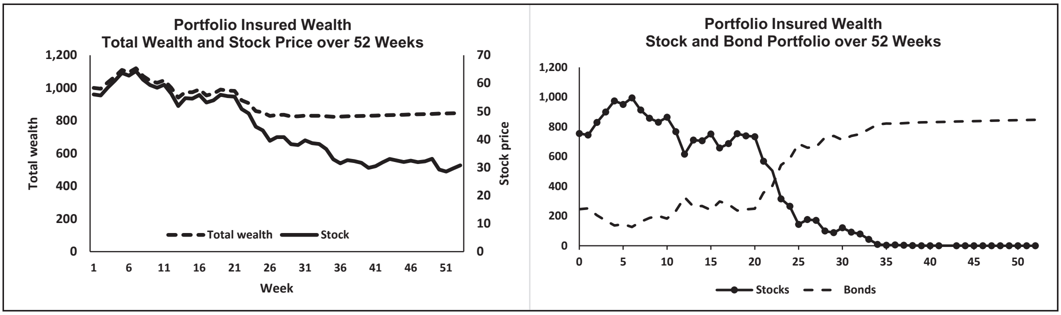
	A	B	C
1	**Updating the Portfolio Insurance Proportions**		
2	Previous stock price S_t	56.00	
3	Previous time to maturity	1.00	
4	Time interval, Δt	0.0192	<-- =1/52
5	New time to maturity	0.9808	<-- =B3-B4
6			
7	Previous portfolio		
8	Stock	754.50	
9	Bonds	245.50	
10	Current stock price $S_{t+\Delta t}$	52.00	Stock price
11			
12	Current portfolio before readjustment		
13	Stock	700.61	<-- =B8*B10/B2
14	Bonds	245.88	<-- =B9*EXP(B22*B4)
15	Total	946.49	<-- =SUM(B13:B14)
16			
17	**Calculating the portfolio insurance proportions**		
18	Proportion of stock Omega, ω	66.41%	<-- =B10*NORM.S.DIST(done(B10,B21,B5,B22,B23), 1)/(B10+bsput(B10,B21,B5,B22,B23))
19	1-omega	33.59%	<-- =1-B18

We now simulate this strategy over time:

	A	B	C	D	E	F	G
1	**PORTFOLIO INSURANCE STRATEGY**						
2	S_0	56.00					
3	X	50.00					
4	T	1				Formulas--in cell F12: =D12*E12	
5	r	8.00%				in cell G12: =E12*(1-D12)	
6	Sigma	30%					
7	Mean	12%					
8	Delta_t	0.0192	<-- =1/52				
9							
10		Beginning of week			Wealth		
11	Week	Stock	Put	Omega	Total wealth	Stocks	Bonds
12	0	56.00	2.38	0.7545	1,000.00	754.50	245.50
13	1	54.92	2.59	0.7324	985.87	722.01	263.86
14	2	54.87	2.58	0.7313	985.63	720.82	264.80
15	3	54.25	2.70	0.7175	977.88	701.66	276.22
16	4	52.34	3.19	0.6711	953.57	639.93	313.65
17	5	52.21	3.20	0.6671	952.40	635.33	317.07
18	6	52.76	3.02	0.6808	959.60	653.32	306.27
19	7	52.93	2.94	0.6850	962.26	659.15	303.10
20	8	52.73	2.97	0.6794	960.16	652.30	307.86
21	9	52.33	3.05	0.6684	955.70	638.76	316.94
22	10	52.34	3.02	0.6684	956.39	639.22	317.17
23	11	50.36	3.62	0.6104	932.64	569.25	363.39
24	12	50.26	3.63	0.6060	932.01	564.80	367.21
25	13	49.05	4.05	0.5661	919.00	520.28	398.73
26	14	46.24	5.24	0.4675	889.84	416.01	473.83



In the above simulation, the stock price decreased during the year, and the final proportions of the portfolio insurance strategy will be wholly in the risk-free asset. Below we present another possibility: The stock price increases over the year, and the portfolio proportions are increasingly in the stock, with the bond going to zero.



29.4 Some Properties of Portfolio Insurance

The preceding example illustrates some of the typical properties of portfolio insurance. Three important properties are the following:

PROPERTY 1 When the stock price is above the exercise price X, then the proportion ω invested in the risky asset is greater than 50%.

Proof The proof of this property requires a little manipulation of our formula for ω. Rewrite ω as:

$$\omega = \frac{SN(d_1)}{SN(d_1) + Xe^{-r(1-t)}N(-d_2)} = \frac{1}{1 + {Xe^{-r(1-t)}N(-d_2)} \big/ {SN(d_1)}}$$

We will show that when $S \geq X$, the denominator of ω is < 2, which will prove the proposition: First note that when $S \geq X$, $X/S \leq 1$. Next note that $e^{-r(1-t)} < 1$ for all $0 \leq t \leq 1$. Finally, examine the expression

$$\frac{N(-d_2)}{N(d_1)} = \frac{N\left(\sigma\sqrt{1-t} - d_1\right)}{N(d_1)}$$
$$= \frac{N\left(0.5\sigma\sqrt{1-t} - [\ln(S/X) + r(1-t)]/\sigma(1-t)\right)}{N\left(0.5\sigma\sqrt{1-t} + [\ln(S/X) + r(1-t)]/\sigma(1-t)\right)} < 1$$

This proves the property.

PROPERTY 2 When the stock's price increases the proportion ω invested in the stock increases and vice versa.

Proof To see this property, it is enough to see that when S increases, the value of the put decreases and $N(-d_1)$ decreases. Rewrite the original definition of ω as:

$$\omega = \frac{S[1 - N(-d_1)]}{S + P} = \frac{[1 - N(-d_1)]}{1 + P/S}$$

Thus, when S increases, the denominator of ω decreases and the numerator increases, which proves Property 2.

PROPERTY 3 As $t \to 1$, one of two things happens: If $S_t > X$, then $\omega_t \to 1$. If $S_t < X$, then $\omega_t \to 0$.

Proof To see this, note that when $S_t > X$ and $t \to 1$, $N(d_1) \to 1$ and $N(-d_1) \to 0$; thus for this case $\omega_t \to 1$. Conversely, when $S_t < X$ and $t \to 1$, $N(d_1) \to 0$ and $N(-d_1) \to 1$ and thus $\omega_t \to 0$. (Strictly speaking these statements are only true as "probability limits"—see Billingsley 1968. What about the case when, as $t \to 1$, $S_t/X \to 1$? In this case $\omega_t \to$ ½. However, the probability of this occurring is zero.)

29.5 Digression: Insuring Total Portfolio Returns

We digress slightly to consider an interesting portfolio insurance problem. So far we have considered only the problem of constructing artificial puts, one per share. A slightly different version of this problem involves constructing a portfolio of puts and shares that guarantees the *total* dollar returns on the *total* initial investment. A typical story goes like this:

You have \$1,000 to invest, and you want to guarantee that a year from now you will have at least \$1,000z. Here z is some number, generally between 0 and 1; for example, if $z = 0.93$, this means that you want your final wealth to be at least \$930.[2] You want to invest in a stock whose current price is S_0 and in a put on the stock with an exercise price X. You want the number of puts to be equal to the number of shares, so that each "package" of share + put costs you $S_0 + P(S_0, X)$. To implement the strategy, you must therefore buy α shares, where:

$$\alpha = \frac{1{,}000}{S_0 + P(S_0, X)}$$

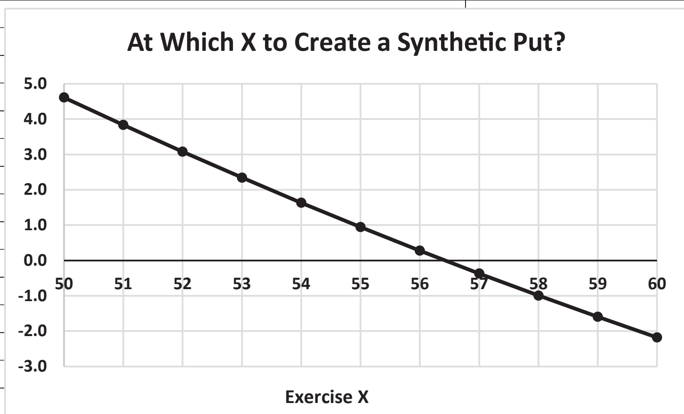
Since you have bought α shares and α puts with an exercise price of X, the minimum dollar return from your portfolio is αX. You want this to be equal to 1,000z, and therefore you solve to get $\alpha = 1{,}000z/X$. Thus you can guarantee your minimum return if

$$S_0 + P(S_0, X) = X / z$$

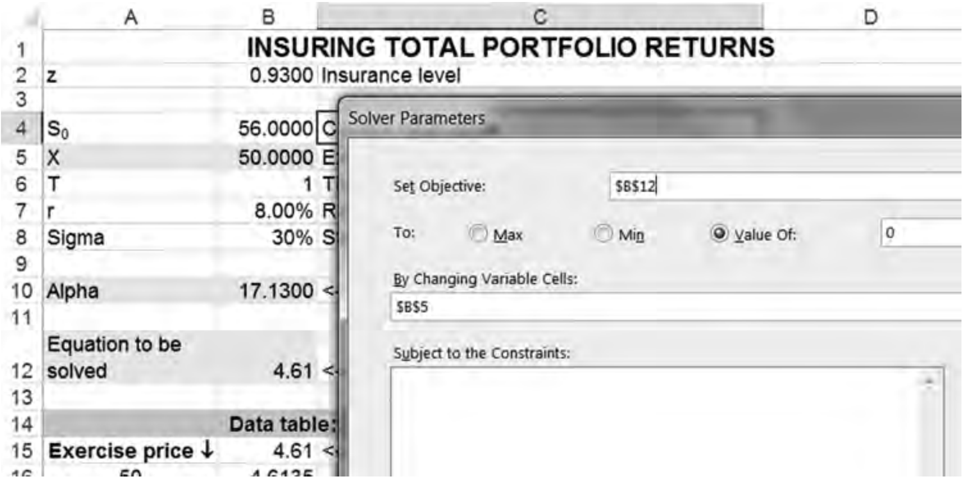
Here's a spreadsheet implementation of this equation. The data table shows the graph of $S_0 + P(S_0,X) - X/z$; where this graph crosses the x-axis is the solution for the put exercise price X when $S_0 = 56$, $\sigma = 30\%$, $r = 6\%$, $T = 1$, and $z = 93\%$.

2. As we show below, it is possible to insure (up to a point) even with $z > 1$.

	A	B	C	D
1	**INSURING TOTAL PORTFOLIO RETURNS**			
2	z	0.9300	Insurance level	
3				
4	S_0	56.0000	Current stock price	
5	X	56.4261	Exercise price	
6	T	1	Time to maturity of option (in years)	
7	r	8.00%	Risk-free rate of interest	
8	Sigma	30%	Stock volatility	
9				
10	Alpha	16.4817	<-- =1000/(B4+bsput(B4,B5,B6,B7,B8))	
11				
12	Equation to be solved	0.00	<-- =B4+bsput(B4,B5,B6,B7,B8)-B5/B2	
13				
14	**Data table: Sensitivity of B12 on X**			
15	**Exercise price ↓**	0.00	<-- =B12, data table header	
16	50	4.6135		
17	51	3.8358		
18	52	3.0800		
19	53	2.3463		
20	54	1.6348		
21	55	0.9453		
22	56	0.2778		
23	57	-0.3679		
24	58	-0.9922		
25	59	-1.5953		
26	60	-2.1776		
27				
28				
29				
30				
31				



As you can see, the X for which the equation in cell B12 equals zero is between 56 and 57. We can use **Solver** to find the exact value, $X = 56.4261$:



Solver gives the solution to the equation $S_0 + Put(S_0,X) - X/z = 0$. The solution is indicated in the next picture:

	A	B	C
1	**INSURING TOTAL PORTFOLIO RETURNS**		
2	z	0.9300	Insurance level
3			
4	S_0	56.0000	Current stock price
5	X	56.4261	Exercise price
6	T	1	Time to maturity of option (in years)
7	r	8.00%	Risk-free rate of interest
8	Sigma	30%	Stock volatility
9			
10	Alpha	16.4817	<-- =1000/(B4+bsput(B4,B5,B6,B7,B8))
11			
12	Equation to be solved	0.00	<-- =B4+bsput(B4,B5,B6,B7,B8)-B5/B2
13			
14	Check		
15	Cost of shares	922.98	<-- =B10*B4
16	Cost of puts	77.02	<-- =B10*bsput(B4,B5,B6,B7,B8)
17	Total cost	1,000.00	<-- =B15+B16
18			
19	Minimum portfolio return	930.00	<-- =B10*B5

The solution is to buy $\alpha = 16.4817$ puts and shares (the cost of which is \$1,000, as you can see in cell B19). The minimum return of this portfolio is $16.4817*X = \$930$ (cell B19).

Can You Insure for More Than Your Initial Investment?

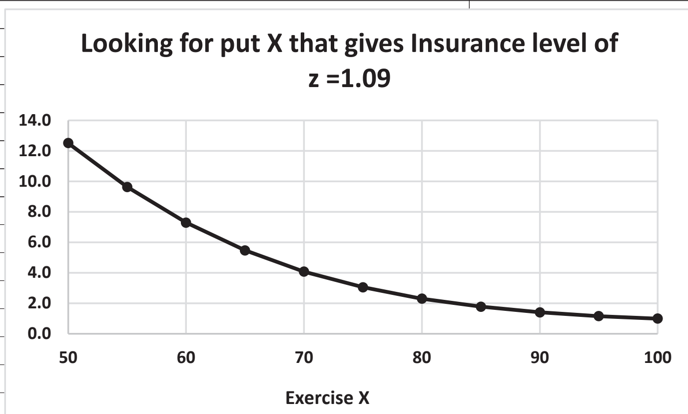
When we raise the insurance level, the implied exercise price of the put must go up. This means that as we buy more insurance, we spend relatively more of our \$1,000 on puts (insurance) and relatively less on the stocks (which have the upside potential).

Can we insure for more than our current level of investment? To put it another way, can we set $z > 1$? This means that we are picking an insurance level which guarantees that we end up with *more* than our initial investment. A little thought and some calculations reveal that we can indeed choose $z > 1$ as long as $z \leq 1 + r$. That is: We cannot guarantee ourselves a return greater than the riskless interest rate! To see this, we offer two examples. In the first example below, we solve for $z = 1.08 = 1 + r$. This has a solution (note that the value of cell B12 is zero):

	A	B	C
1	**INSURING TOTAL PORTFOLIO RETURNS**		
2	z	1.0800	Insurance level
3			
4	S_0	56.0000	Current stock price
5	X	104.8368	Exercise price
6	T	1	Time to maturity of option (in years)
7	r	8.00%	Risk-free rate of interest
8	Sigma	30%	Stock volatility
9			
10	Alpha	10.3017	<-- =1000/(B4+bsput(B4,B5,B6,B7,B8))
11			
12	Equation to be solved	0.00	<-- =B4+bsput(B4,B5,B6,B7,B8)-B5/B2
13			
14	Check		
15	Cost of shares	576.90	
16	Cost of puts	423.10	
17	Total cost	1,000.00	
18			
19	Minimum portfolio return	1,080.00	<-- =B10*B5

When $z > 1.08$, however, there is no solution. Returning to the first chart of this section, we can see that there is no solution that insures for 9% ($z = 1.09$) that is > the interest rate of 8%:

	A	B	C	D
1	**INSURING TOTAL PORTFOLIO RETURNS**			
2	z	1.0900	Insurance level	
3				
4	S_0	56.0000	Current stock price	
5	X	56.4261	Exercise price	
6	T	1	Time to maturity of option (in years)	
7	r	8.00%	Risk-free rate of interest	
8	Sigma	30%	Stock volatility	
9				
10	Alpha	16.4817	<-- =1000/(B4+bsput(B4,B5,B6,B7,B8))	
11				
12	Equation to be solved	8.91	<-- =B4+bsput(B4,B5,B6,B7,B8)-B5/B2	
13				
14	**Data table: Sensitivity of B12 on X**			
15	**Exercise price ↓**	8.91	<-- =B12, data table header	
16	50	12.5054		
17	55	9.6264		
18	60	7.2927		
19	65	5.4649		
20	70	4.0756		
21	75	3.0466		
22	80	2.3018		
23	85	1.7738		
24	90	1.4072		
25	95	1.1581		
26	100	0.9932		
27				
28				
29				
30				



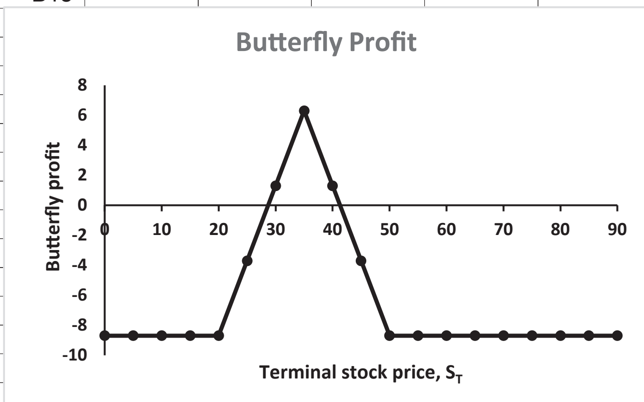
29.6 Simulating a Butterfly

For our last exercise in this chapter, we simulate a butterfly strategy. Recall from Chapter 15 that a butterfly consists of three options. In this section we simulate a butterfly over 1 month (22 days), rebalancing the position daily. Our butterfly consists of three calls written on a stock with current price $S_0 = 35$. We assume that the stock return is lognormal with $\sigma = 35\%$ and that $r = 2\%$. The calls:

- Call 1: $X = 20$, position: 1 call purchased
- Call 2: $X = 35$, position: 2 calls written
- Call 3: $X = 50$, position: 1 call purchased

The payoff/profit pattern of the butterfly is described below:

	A	B	C	D	E	F	G	H
1	**BUTTERFLY**							
2	S_0	35						
3	T	0.087302	<-- =22/252					
4	Sigma	80%						
5	r	2%						
6								
7		**X**	**Position**	**Cost**				
8	Call1	20	1	15.05	<-- =bscall($B$2,B8,$B$3,$B$5,$B$4)			
9	Call2	35	-2	3.32	<-- =bscall($B$2,B9,$B$3,$B$5,$B$4)			
10	Call3	50	1	0.29	<-- =bscall($B$2,B10,$B$3,$B$5,$B$4)			
11	Initial cost	8.70	<-- =SUMPRODUCT(C8:C10,D8:D10)					
12								
13	**One profit example (used in data table)**							
14	S_T	47						
15	Profit1	11.95	<-- =C8*(MAX($B$14-B8,0)-D8)					
16	Profit2	-17.36	<-- =C9*(MAX($B$14-B9,0)-D9)					
17	Profit3	-0.29	<-- =C10*(MAX($B$14-B10,0)-D10)					
18	Total	-5.70	<-- =SUM(B15:B17)					
19								
20	**Data table: profit as function of S_T**							
21	**S_T**	**Payoff**						
22		-5.70	<-- =B18					
23	0	-8.70						
24	5	-8.70						
25	10	-8.70						
26	15	-8.70						
27	20	-8.70						
28	25	-3.70						
29	30	1.30						
30	35	6.30						
31	40	1.30						
32	45	-3.70						
33	50	-8.70						
34	55	-8.70						
35	60	-8.70						
36	65	-8.70						
37	70	-8.70						
38	75	-8.70						



We recall the Black-Scholes formula and denoting the time remaining to option maturity by t:

$$Call(X) = SN(d_1, t) - Xe^{-rT} N(d_2, t)$$
$$d_1 = \frac{\ln\left(S/X\right) + \left(r + \sigma^2/2\right)t}{\sigma\sqrt{t}}, d_2 = d_1 - \sigma\sqrt{t}$$

In a table for the butterfly:

N_{Low} calls	$Call(X_{Low}, t) = N_{Low}\left[SN(d_1(X_{Low}, t)) - X_{Low}e^{-rT} N(d_2(X_{Low}, t))\right]$ $d_1(X_{Low}, t) = \frac{\ln\left(S/X_{Low}\right) + \left(r + \sigma^2/2\right)t}{\sigma\sqrt{t}}, d_2(X_{Low}, t) = d_1(X_{Low}, t) - \sigma\sqrt{t}$
N_{Mid} calls	$Call(X_{Mid}, t) = N_{Mid}\left[SN(d_1(X_{Mid}, t)) - X_{Mid}e^{-rT} N(d_2(X_{Mid}, t))\right]$ $d_1(X_{Mid}, t) = \frac{\ln\left(S/X_{Mid}\right) + \left(r + \sigma^2/2\right)t}{\sigma\sqrt{t}}, d_2(X_{Mid}, t) = d_1(X_{Mid}, t) - \sigma\sqrt{t}$
N_{High} calls	$Call(X_{High}, t) = N_{High}\left[SN(d_1(X_{High}, t)) - X_{High}e^{-rT} N(d_2(X_{High}, t))\right]$ $d_1(X_{High}, t) = \frac{\ln\left(S/X_{High}\right) + \left(r + \sigma^2/2\right)t}{\sigma\sqrt{t}}, d_2(X_{High}, t) = d_1(X_{High}, t) - \sigma\sqrt{t}$

Adding these together we get

$$\begin{aligned} Butterfly(t) = S_t\{&N_{Low} * N(d_1(X_{Low}, t)) + N_{Mid} * N(d_1(X_{Mid}, t)) \\ &+ N_{High} * N(d_1(X_{High}, t))\} + -e^{-rT}\{N_{Low} * X_{Low} N(d_2(X_{Low}, t)) \\ &+ N_{Mid} * X_{Mid} N(d_2(X_{Mid}, t)) + N_{High} * X_{High} N(d_2(X_{High}, t))\} \end{aligned}$$

Newly defined VBA functions:

$$\begin{aligned} Butterfly(t) = S_t &* \text{butterflyNd1}\left(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_t, X, t, \sigma, r}\right) \\ &- e^{-rT}\text{butterflyNd2}\left(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_t, X, t, \sigma, r}\right) \end{aligned}$$

The VBA function **ButterflyNd1** is defined below (the function **ButterflyNd2** is similar):

```
Function butterflyNd1(XLow, XMid, XHigh, _
NumberLow, NumberMid, NumberHigh, Stock, _
Time, Interest, sigma)
butterflyNd1 = Stock * _
     (NumberLow * Application.Norm_S_Dist _
     (dOne(Stock, XLow, Time, Interest, sigma), 1) _
    + NumberMid * Application.Norm_S_Dist _
    (dOne(Stock, XMid, Time, Interest, sigma), 1) _
    + NumberHigh * Application.Norm_S_Dist _
    (dOne(Stock, XHigh, Time, Interest, sigma), 1))
End Function
```

Self-Financing Butterfly Portfolio

We attempt to replicate this position dynamically, maintaining the self-financing requirement (i.e., no cash flows after time $t = 0$).

At time t = 0, we define the investment in the stock and the bond as above.

$$Stock(0) = S_0 \begin{Bmatrix} N_{Low} * N(d_1(X_{Low}, t=0)) + N_{Mid} * N(d_1(X_{Mid}, t=0)) \\ + N_{High} * N(d_1(X_{High}, t=0)) \end{Bmatrix}$$

Afterward, at time $t + \Delta t$, we define the stock investment as above and adjust the bond investment to net out the stock:

$$Stock(t+\Delta t) = S_{t+\Delta t} \begin{Bmatrix} N_{Low} * N(d_1(X_{Low}, t+\Delta t)) + N_{Mid} * (d_1(X_{Mid}, t+\Delta t)) \\ + N_{High} * N(d_1(X_{High}, t+\Delta t)) \end{Bmatrix}$$

Cash Flow from the Stock Position

The cash flow from the stock position is the value of the time t position minus the value of the time $t + \Delta t$ position.

$$Stock\ cash\ flow(t+\Delta t) = Stock(t) * \frac{S_{t+\Delta t}}{S_t} - Stock(t+\Delta t) =$$

$$\frac{S_{t+\Delta t}}{S_t}\{N_{Low} * N(d_1(X_{Low}, S_t, t)) + N_{Mid} * N(d_1(X_{Mid}, S_t, t))$$

$$+ N_{High} * N(d_1(X_{High}, S_t, t))\} -$$

$$S_{t+\Delta t}\left\{\begin{matrix} N_{Low} * N(d_1(X_{Low}, S_{t+\Delta t}, t+\Delta t)) + N_{Mid} * (d_1(X_{Mid}, S_{t+\Delta t}, t+\Delta t)) \\ +N_{High} * (d_1(X_{High}, S_{t+\Delta t}, t+\Delta t)) \end{matrix}\right\}$$

In Excel:

$$\frac{S_{t+\Delta t}}{S_t} * \mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_t, X, t, r, \sigma})$$
$$- \mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_{t+\Delta t}, X, t+\Delta t, r, \sigma})$$

Zero-Investment Cash Flow from the Bond Position

In order to define a self-financing strategy, we let the bond position at each time t "absorb" the change in the stock position. At time $t = 0$, the bond position is

$$Bond(0) = -e^{-rT}\left\{\begin{matrix} N_{Low} * X_{Low} N(d_2(X_{Low})) + N_{Mid} * X_{Mid} N(d_2(X_{Mid})) \\ +N_{High} * X_{High} N(d_2(X_{High})) \end{matrix}\right\}$$

At time $t + \Delta t$, the bond position is the previous position minus the cash flow on the stock position:

$$Bond(t+\Delta) = Bond(t) * e^{r\Delta t}$$
$$-\left\{\begin{matrix} \frac{S_{t+\Delta t}}{S_t} * \mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_t, X, t, r\sigma}) \\ -\mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_{t+\Delta t}, X, t+\Delta t, r, \sigma}) \end{matrix}\right\}$$

The cash flow on the bond position is

$$\text{Bond cash flow}(t+\Delta t) = Bond(t) * e^{r\Delta t} - Bond(t+\Delta t)$$
$$= Bond(t) * e^{r\Delta t} - Bond(t) * e^{r\Delta t}$$
$$+\left\{\begin{matrix} \frac{S_{t+\Delta t}}{S_t} * \mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_t, X, t, r\sigma}) \\ -\mathrm{butterflyNd1}(\mathrm{X_{Low}, X_{Mid}, X_{High}, N_{Low}, N_{Mid}, N_{High}, S_{t+\Delta t}, X, t+\Delta t, r, \sigma}) \end{matrix}\right\}$$

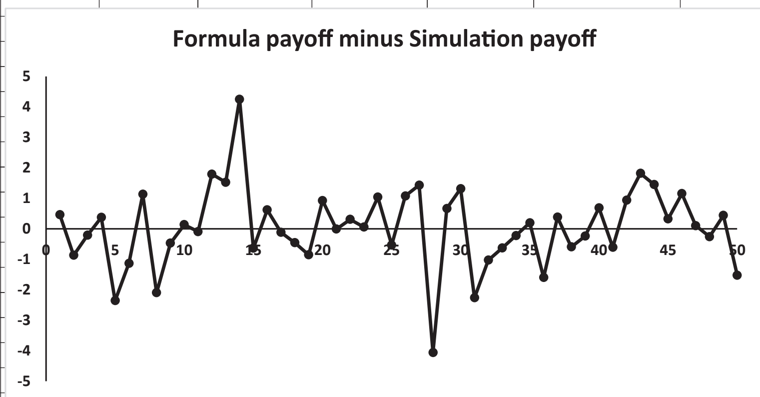
Running the Simulation

We run the simulation:

	A	B	C	D	E	F	G	H
1	**SIMULATING A BUTTERFLY**							
2	S_0	35			Butterfly	X	Number	
3	T	0.0873	<-- =22/252		Xlow	20	-1	
4	Sigma	80%			Xmid	35	2	
5	r	2%			Xhigh	50	-1	
6	Delta_t	0.0040	<-- =1/252					
7								
8	Butterfly simulation							
9	Time to maturity	Stock	Stock position	Bond position				
10	0.0873	35.0000	0.8045	-9.5036				
11	0.0833	34.9221	0.6791	-9.3806				
12	0.0794	34.6785	0.0585	-8.7656	<--			
13	0.0754	39.2902	12.4606	-21.1606	=D10*EXP($B$5*Delta_t)+B11/B10*butterflyN			
14	0.0714	35.4297	2.5165	-12.4425	d1($F$3,$F$4,$F$5,$G$3,$G$4,$G$5,B10,A1			
15	0.0675	35.7399	3.6778	-13.5827	0,$B$5,$B$4)-			
16	0.0635	37.6961	10.1079	-19.8126	butterflyNd1($F$3,$F$4,$F$5,$G$3,$G$4,G			
17	0.0595	39.5127	15.6702	-24.8894	5,B11,A11,$B$5,$B$4)			
18	0.0556	39.3373	16.0412	-25.3320				
19	0.0516	40.0574	18.7535	-27.7526				
20	0.0476	37.8111	12.9252	-22.9781				
21	0.0437	39.6546	19.8128	-29.2374				
22	0.0397	38.0253	15.3452	-25.5861				
23	0.0357	39.1652	20.6678	-30.4508				
24	0.0317	37.0963	12.9733	-23.8505				
25	0.0278	37.8587	17.7407	-28.3531				
26	0.0238	37.6234	17.7742	-28.4991				
27	0.0198	36.2854	10.6435	-22.0028				
28	0.0159	39.9963	32.6490	-42.9215				
29	0.0119	38.3893	28.0061	-39.5939				
30	0.0079	39.5149	36.2603	-47.0300				
31	0.0040	38.1897	35.1787	-47.1683				
32	0.0000	37.5170	34.5591	-47.1720				
33								
34	Comparison of formula payoff to simulation payoff							
35	Formula	-12.4830	<-- =G3*MAX(B32-F3,0)+G4*MAX(B32-F4,0)+G5*MAX(B32-F5,0)					
36	Simulation	-12.6129	<-- =SUM(C32:D32)					

Repeating this simulation 50 times and comparing the formula payoff (B35) to the simulation payoff (B36):

	A	B	C	D	E	F	G	H	I
34	Comparison of formula payoff to simulation payoff								
35	Formula	-3.0153	<-- =G3*MAX(B32-F3,0)+G4*MAX(B32-F4,0)+G5*MAX(B32-F5,0)						
36	Simulation	-2.2116	<-- =SUM(C32:D32)						
37									
38	Data table: 50 runs of the simulation								
39									
40	Run	-0.8037	<-- =B35-B36, data table header						
41	1	0.4614							
42	2	-0.8721			Statistics for 50 runs: Formula payoff-simulation				
43	3	-0.2146			Average	0.0329	<-- =AVERAGE(B41:B90)		
44	4	0.3714			Sigma	1.3119	<-- =STDEV.S(B41:B90)		
45	5	-2.3605			Max	4.2486	<-- =MAX(B41:B90)		
46	6	-1.1354			Min	-4.0611	<-- =MIN(B41:B90)		
47	7	1.1293							
48	8	-2.0989							
49	9	-0.4745							
50	10	0.1328							
51	11	-0.0947							
52	12	1.7906							
53	13	1.5235							
54	14	4.2486							
55	15	-0.6404							
56	16	0.6187							
57	17	-0.1200							
58	18	-0.4576							
59	19	-0.8565							
60	20	0.9230							
61	21	-0.0117							
62	22	0.3053							
63	23	0.0551							
64	24	1.0346							



29.7 Summary

This chapter concentrated on simulating option replication strategies. All of the replication strategies are based on the idea that the Black-Scholes formula gives the dynamic investment over time in the risky and riskless asset. When implementing this idea, we had to adjust this dynamic formula so that the net investment over time is zero (the so-called self-financing or zero-investment strategies). The replication strategies then become somewhat less than perfect, so that the terminal payoff of these strategies matches the formula payoff only approximately. For more complicated strategies (we illustrated for a butterfly), the mismatch between the formula and the strategy payoff can be significant.

Exercises

1. You are a portfolio manager, and you want to invest in an asset having $\sigma = 40\%$. You want to create a put on the investment so that at the end of the year you have losses no greater than 5%. Since there is no put on this specific asset, you plan to create a synthetic put by engaging in a dynamic investment strategy—purchasing a portfolio composed of dynamically changing proportions of the risky asset and riskless bonds. If the interest rate is 6%, how much should your initial investment be in the portfolio and in the riskless bond?

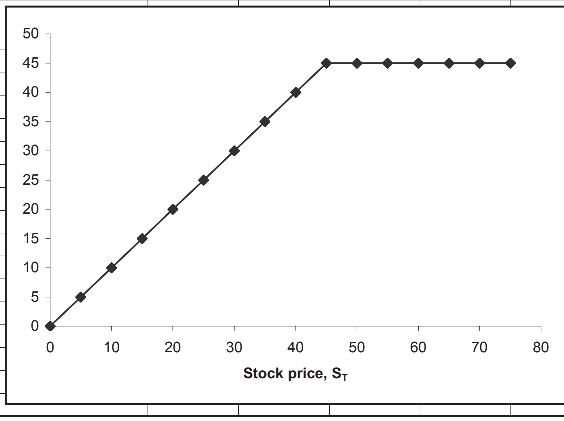
2. Simulate the above strategy, assuming weekly rebalancing of the portfolio.

3. Go back to the numerical example of section 29.5. Write a VBA function which solves for the implied asset value V_a. (Hint: Use the bisection method discussed to compute the implied volatility in section 17.5.) Then use this function to create a graph showing the trade-off between the implied asset value and the asset volatility.

4. You have been offered the chance to purchase stock in a firm. The seller wants $55 per share, but offers to repurchase the stock at the end of one half year for $50 per share. If the σ of the share's log returns is 80% determine the true value per share. Assume that the interest rate is 10%.

5. A covered call is a long stock and short call. The pattern of payoffs is given below:

	A	B	C	D	E	F	G	H
1	**COVERED CALL PROFIT PATTERN**							
2	Future stock price, S_T	50						
3	Call exercise price, X	45						
4	Call price	6						
5								
6	Payoffs at T							
7	Long stock, $+S_T$	50	<-- =B2					
8	Short call payoff, -Max[S_T-X,0]	-5	<-- =-MAX(B2-B3,0)					
9	Total payoff	45	<-- =B7+B8					
10								
11								
12	**Data table of payoffs**							
13	Future stock price, S_T	45						
14	0	0						
15	5	5						
16	10	10						
17	15	15						
18	20	20						
19	25	25						
20	30	30						
21	35	35						
22	40	40						
23	45	45						
24	50	45						
25	55	45						
26	60	45						
27	65	45						
28	70	45						
29	75	45						
30								



In this problem, you are asked to simulate the payoffs of a covered call over 52 weeks, with weekly updating of the positions. Start by deriving the formula for the covered call: Add together the Black-Scholes price and the stock price:

$$\underbrace{S_0}_{\substack{\uparrow \\ \text{Long stock}}} \underbrace{-S_0 N(d_1) + Xe^{-rT} N(d_2)}_{\substack{\uparrow \\ \text{Short call}}} = \underbrace{S_0\,(1 - N(d_1))}_{\substack{\uparrow \\ \text{Long position} \\ \text{in stock}}} + \underbrace{Xe^{-rT} N(d_2)}_{\substack{\uparrow \\ \text{Long position} \\ \text{in bond}}}$$

Thus we see that a covered call is a long position in the stock and a long position in the bond. Now implement the following spreadsheet to test the effectiveness of a simulated covered call strategy.

6. Section 29.2 discusses the cashless replication of a call. Use the same logic to program in a spreadsheet the replication of a put.

30 Using Monte Carlo Methods for Option Pricing

30.1 Overview

This chapter continues the discussion of the previous chapter and shows how to implement Monte Carlo methods for pricing options. The main objective is to show how to price Asian options and barrier options. Both of these options are *path dependent*: Their payoffs depend not just on the terminal price of an underlying asset but also on the intermediate asset prices before the terminal price. An Asian option has a payoff which depends on the average price of the underlying asset for some period before option maturity, and a barrier option's payoff depends on the underlying price reaching a particular level at some point before maturity. In sections 30.6–30.9 we will make these general statements more explicit.

What Does Risk Neutrality Mean?

Consider a situation in which there are two basic securities, a riskless bond and a stock. The interest rate on the bond is called the risk-free rate and is usually denoted by r. In the context of this situation of two basic assets, "risk-neutral pricing" can have two meanings. In both meanings of risk-neutrality derivative (i.e., non-basic) securities are priced as the discounted value (at the risk-free rate) of their expected payoffs. The two meanings of risk neutrality differ in the procedure used to arrive at the payoffs of the risky basic security (the stock).

• In the first meaning of risk neutrality, we change the underlying distribution of the stock's returns so that the expected stock price is one plus the risk-free rate. To price an option we apply the basic risk-neutral pricing principle of discounting the expected option payoffs by the risk-free rate. In this use of risk neutrality, we transform the stock's returns but compute the expected payoffs using the actual return probabilities. We illustrate this procedure in section 30.2.

• The second meaning of risk neutrality uses the binomial model (see Chapter 16). We transform the state prices into risk-neutral probabilities. In this use of risk neutrality, we do not transform the stock's payoffs but instead replace the actual state probabilities by the equivalent risk-neutral probabilities. We then price derivative securities by discounting their expected payoffs at the risk-free rate.

The first method of pricing is ideal for pricing options whose payoffs depend only on the terminal price of the stock. We call such options *path independent*. The second method is more general and can in principle be used to price any option, even if the payoffs are path dependent. Examples of such options explored in this chapter are Asian and barrier options. Both these options are path-dependent options—options whose price depends not only on the terminal price of the asset, but also on the path of the prices by which the terminal price was reached. In general a path-dependent option does not have an analytic price solution. Monte Carlo provides us with a handy numerical tool for pricing such options. Monte Carlo pricing of path-dependent options depends on a simulation of the price path of the underlying asset.

The Structure of This Chapter

In section 30.2 we show how the first risk-neutrality pricing principle (transforming the stock's returns) can be used to Monte Carlo price a standard call and a put. We then continue with the second meaning of risk neutrality and show how it can be used to price barriers and Asian options.

To make this chapter more self-contained we include a brief review of state prices and risk neutrality (section 30.3). We then show, in section 30.4, how to price a plain-vanilla option with a Monte Carlo algorithm. Since plain-vanilla options—jargon by which we simply mean European calls and puts on a stock whose price process is lognormal—are accurately priced using the Black-Scholes formula, this exercise allows us to check our pricing method against a known result and also allows us to develop the proper intuitions about the Monte Carlo pricing of more complicated options.

30.2 Pricing a Plain-Vanilla Call Using Monte Carlo Methods

In this section we explore the pricing of standard European calls and puts using Monte Carlo. We employ the first meaning of risk neutrality discussed in section 30.1, transforming the returns on the stock so that the stock's expected return is the risk-free rate and then discounting the expected option payoffs at the risk-free rate.

To use this complicated model to price European calls and puts with Monte Carlo methods might be viewed as an immense waste of time—the Black-Scholes formula (Chapter 17) gives a wonderful pricing solution for European calls and puts. However, like the example of estimating the value of

π discussed in Chapter 25, the exercise of pricing a plain-vanilla call using Monte Carlo methods gives us considerable insight into the application of Monte Carlo methods.

The Procedure

Consider a call on a stock with the parameters given below. This call can be priced using the Black-Scholes formula. We will show how our first risk-neutral procedure of transforming the returns can be used to price this call using Monte Carlo.

	A	B	C
1	**BLACK-SCHOLES CALL PRICE**		
2	S_0	50	
3	X	44	
4	T	1	
5	r	5%	
6	Sigma	15%	
7			
8	Black-Scholes call price	8.5417	<-- =BSCall(B2,B3,B4,B5,B6)

We now simulate the option price as follows:

- Step 1: We simulate a set of stock prices at time T. The stock prices have the property that they are lognormally distributed with mean r and standard deviation σ. To do this we assume that the stock prices have mean $r - \sigma^2/2$ and standard deviation σ.
- Step 2: We compute the terminal payoffs of our option for these stock prices: $\max(S_T - X,0)$.
- Step 3: We compute the average discounted payoff: $\exp[-r*T]*Average[\max(S_T - X,0)]$. This should be the same as the Black-Scholes price (and approximately it is).

Why Do We Assume That the Stock's Mean Return Is $r - \sigma^2/2$?

Suppose that the stock returns are normal with mean μ and standard deviation σ. As discussed in Chapter 26, this means that the expected stock price is $S_0\exp[(\mu + \sigma^2/2)t]$. In our first meaning of risk neutrality, we want to transform the return distribution so that the expected price has mean r. We do this by replacing μ by $r - \sigma^2/2$. This guarantees that the expected future price of the stock grows at rate r, which is the basic condition for the risk-neutral pricing:

$$S_0 \exp\left[\left(\mu + \sigma^2/2\right)t\right] = S_0 \exp\left[\left(\left(r - \sigma^2/2\right) + \sigma^2/2\right)t\right] = S_0 \exp[rt]$$

Step 1: Generating the Price Data

Having cleared up this theoretical issue, we now return to our computations. We first generate 1,000 future stock prices. Each price is generated using:

$$S_T = S_0 * \exp\left[\left(r - \frac{\sigma^2}{2}\right) * T + \sigma * \sqrt{T} * Z\right]$$

Here Z is the standard normal deviate produced, as illustrated in Chapter 26, by the Excel function **Norm.S.Inv(Rand( ))**. Here's some sample data:

	A	B	C	D	E	F	G	H	I
1	**SIMULATING RISK-NEUTRAL STOCK PRICES** **The terminal stock prices S_T in column D have standard deviation σ and expected return r**								
2				**Simulated time T stock prices**				**Stock prices**	
3	S_0	50		50.4594	<-- =$B$3*EXP(($B$6-		Count	1,000	<-- =COUNT(D:D)
4	X	44		57.3000	$B$7^2/2)*$B$5+$B$7*SQRT(		Max	80.5565	<-- =MAX(D:D)
5	T	1		43.1088	$B$5)*NORM.S.INV(RAND()))		Min	32.7373	<-- =MIN(D:D)
6	r	5%		62.2230			Average	52.6528	<-- =AVERAGE(D:D)
7	Sigma	15%		70.9889			Sigma	7.9982	<-- =STDEV.S(D:D)
8				54.0741					
9	Black-Scholes call price	8.5417	<--	55.4966			**Data statistics**		
10			=BSCall(B3,B4,	56.7003			Return	5.17%	<-- =LN(H6/B3)
11			B5,B6,B7)	60.0961			Sigma of return	15.19%	<-- =H7/H6
12				41.8308					
13				45.5635			**Theoretical statistics**		
14				44.9597			Return	5.13%	<-- =EXP(B6*B5)-1
15				59.0905			Sigma of return	15.00%	<-- =B7
16				50.5293					

Steps 2 and 3

We now use the sample data to compute the payoffs of our option. Below we compute the call payoff for each simulated stock price (column E), and in cell B10 we compute the discounted value (at the risk-free rate) of the average of these payoffs $\exp[-r*T]*Average[\max(S_T - X,0)]$. As predicted, this discounted average closely matches the Black-Scholes price.

	A	B	C	D	E	F
1	**PRICING A CALL WITH SIMULATED RISK-NEUTRAL PRICES** 1000 simulations (columns D, E)					
2				**Simulated time T stock prices**	**Simulated call payoff**	
3	S_0	50		47.3920	3.3920	<-- =MAX(D3-$B$4,0)
4	X	44		57.7402	13.7402	<-- =MAX(D4-$B$4,0)
5	T	1		46.5386	2.5386	<-- =MAX(D5-$B$4,0)
6	r	5%		62.5030	18.5030	
7	Sigma	15%		60.6115	16.6115	
8				59.3324	15.3324	
9	**Call option valuation**			63.4439	19.4439	
10	Discounted average payoff	8.6400	<-- =EXP(-B6*B5)*AVERAGE(E:E)	58.7748	14.7748	
11	Black-Scholes call price	8.5417	<-- =BSCall(B3,B4,B5,B6,B7)	71.8958	27.8958	
12				47.4755	3.4755	

The same procedure works for puts:

	A	B	C	D	E	F
1	**PRICING A PUT WITH SIMULATED RISK-NEUTRAL PRICES** 1000 simulations (columns D, E)					
2				**Simulated time T stock prices**	**Simulated call payoff**	
3	S_0	50		59.8926	0.0000	<-- =MAX($B$4-D3,0)
4	X	44		52.6776	0.0000	<-- =MAX($B$4-D4,0)
5	T	1		62.4782	0.0000	<-- =MAX($B$4-D5,0)
6	r	5%		67.1018	0.0000	
7	Sigma	15%		62.4515	0.0000	
8				53.0296	0.0000	
9	**Put option valuation**			46.7010	0.0000	
10	Discounted average payoff	0.3801	<-- =EXP(-B6*B5)*AVERAGE(E:E)	43.4355	0.5645	
11	Black-Scholes call price	0.3958	<-- =bsput(B3,B4,B5,B6,B7)	73.7133	0.0000	
12				53.5770	0.0000	

Digital Option

To show the power of this method, suppose we wish to price a security that pays off the second digit of the terminal stock price. As an example: If the $S_T = 43.5323$, our "digital security" pays off 3; if $S_T = 50.5323$, the payoff is 0. (Why anyone would want to buy such a security is a separate question.)

The spreadsheet below shows that this security can easily be priced using the Monte Carlo method explained in this section. We simulate 1,000 stock prices and use the Excel function **Int(Mod(S_T,10)** to determine the second digit of the terminal stock price:

	A	B	C	D	E	F
1	**DERIVATIVE PAYS OFF SECOND DIGIT OF TERMINAL STOCK PRICE**					
2				**Simulated time T stock prices**	**Simulated call payoff**	
3	S_0	50		53.6894	3.0000	<-- =INT(MOD(D3,10))
4	X	44		49.2395	9.0000	<-- =INT(MOD(D4,10))
5	T	1		44.3635	4.0000	<-- =INT(MOD(D5,10))
6	r	5%		45.7508	5.0000	
7	Sigma	15%		44.5083	4.0000	
8				57.4873	7.0000	
9	**Digital option valuation**			48.8942	8.0000	
10	Discounted average payoff	4.2606	<-- =EXP(-B6*B5)*AVERAGE(E:E)	46.8333	6.0000	
11				46.5321	6.0000	

30.3 State Prices, Probabilities, and Risk Neutrality

In the remainder of this chapter we use the second meaning of risk neutrality discussed in section 30.1: In the framework of a binomial model, we compute risk-neutral state probabilities which are then used to compute the expected payoffs of derivative securities. Our main focus will be on the pricing of barrier and Asian options, but the principles are general and can be applied to most derivatives.

In this section we start with a brief recapitulation of the basic facts about state prices and risk-neutral pricing discussed in Chapter 16. Suppose we have binomial framework with stock price S which grows at U or D in every period. Suppose the interest rate is R.[1] Risk neutrality is a property of every set of state prices: Given state prices $\left\{q_U = \frac{R-D}{R(U-D)}, q_D = \frac{U-R}{R(U-D)}\right\}$, the risk-neutral probabilities are defined by $\{\pi_U = Rq_U, \pi_D = Rq_D\}$. The risk-neutral probabilities can be used to price assets by taking the *discounted expected*

1. Properly speaking, U, D, and R are *one plus* the growth and interest rates. For the sake of linguistic parsimony, we well use "up growth," "down growth," and "interest rate" even though we mean something slightly different.

value of the asset payoffs.[2] An asset with payoffs one period hence of $\{Payoff_U, Payoff_D\}$ in states U and D respectively has value today of:

$$Asset\ value\ today = q_U Payoff_U + q_D Payoff_D = \underbrace{\frac{\pi_U Payoff_U + \pi_D Payoff_D}{R}}_{\substack{\uparrow \\ \text{The risk-neutral expected payoff} \\ \text{(expectation computed} \\ \text{with the risk-neutral probabilities)} \\ \text{discounted at the risk-free interest} \\ \text{rate.}}}$$

The $\{Payoff_U, Payoff_D\}$ are usually functions of the underlying asset price. For an ordinary call, for example, $\{Payoff_U = \max(S^*U - X,0), Payoff_D = \max(S^*D - X,0)\}$.

We can extend the risk-neutral pricing scheme to multi-period frameworks. Consider a multi-period binomial setting where U and D do not change over time, and indicate the date n state payoffs by $Payoff_{n,j}$, $j = 0,\ldots,n$. This notation $Payoff_{n,j}$ indicates the date-n payoff of the asset in a state where there are j up moves on the binomial tree; for a call in a binomial framework $Payoff_{n,j} = Max(S^*U^jD^{n-j} - X,0)$. Then the value of this asset is given by

$$Asset\ value\ today = \sum_{j=0}^{n}\binom{n}{j} q_U^j q_D^{n-j} Payoff_{n,j} = \underbrace{\frac{1}{R^n}\sum_{j=0}^{n}\binom{n}{j}\pi_U^j \pi_D^{n-j} Payoff_{n,j}}_{\substack{\uparrow \\ \text{The risk-neutral expected} \\ \text{discounted value.}}}$$

This particular notation assumes that the tree is *recombining*. It assumes, in other words, that the date-n payoffs are *path independent*—the option payoff is a function only of the terminal stock price and does not depend on the path by which this price is reached (see Figure 30.1). We return to this topic later in this chapter.

2. As discussed in Chapter 16, the risk-neutral probabilities are not the actual probabilities of the state occurrences. They are in fact "pseudo probabilities" which derive from the state prices.

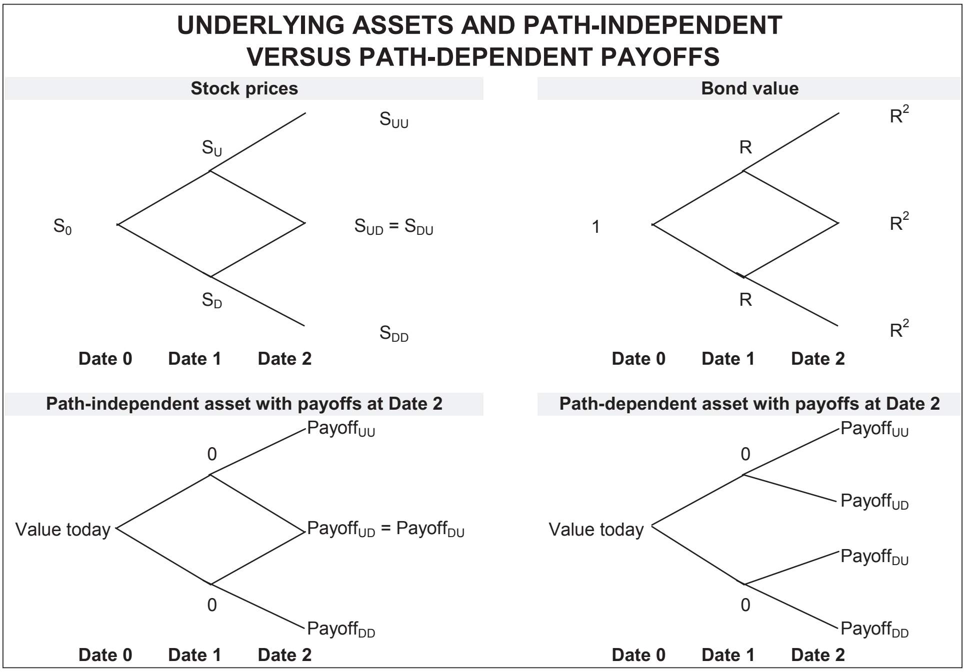


Figure 30.1
Recombining versus non-recombining binomial models.

30.4 Pricing a Call Using the Binomial Monte Carlo Model

We start with a ridiculously simple example. We use Monte Carlo to price a European call in a two-period setting in which the stock price goes either up or down each period. In the spreadsheet below we use Monte Carlo methods to price an at-the-money option on a stock whose price today is $S_0 = 50$. There are two periods, and in each period the stock price goes either Up = 1.4 or Down = 0.9; the interest rate is $R = 1.05$. Given Up and Down, the stock price tree looks like Figure 30.2.

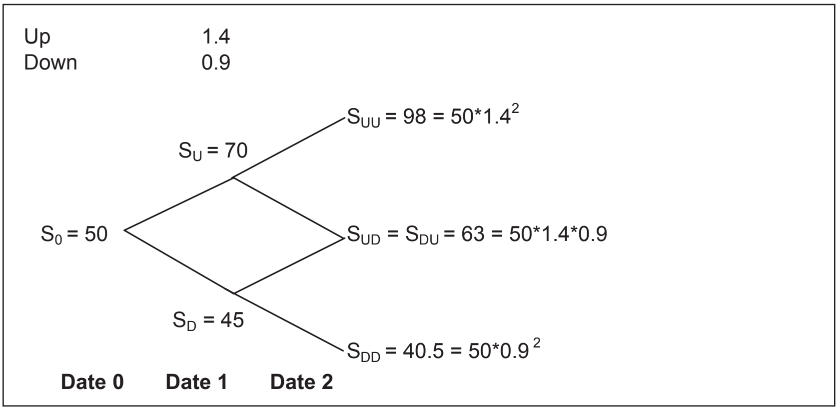


Figure 30.2
Stock price tree in a standard recombining binomial model.

The spreadsheet below shows two random price paths and their pricing:

	A	B	C	D
1	**SIMPLE SIMULATION: TWO PATHS IN A TWO-DATE MODEL**			
2	Initial stock price	50		
3	X	50		
4				
5	Up	1.4		
6	Down	0.9		
7	R	1.05		
8				
9	**State prices**			
10	q_u	0.2857	<-- =(B7-B6)/(B7*(B5-B6))	
11	q_d	0.6667	<-- =(B7-B6)/(B7*(B5-B6))	
12				
13	**Risk-neutral probabilities**			
14	π_u	0.3000	<-- =(B7-B6)/(B7*(B5-B6))	
15	π_d	0.7000	<-- =(B7-B6)/(B7*(B5-B6))	
16				
17	**Random paths and the Monte Carlo price**			
18	First period, up (1) or down (0)?	0	0	<-- =IF(RAND()>$B$15,1,0)
19	Second period, up (1) or down (0)?	0	1	<-- =IF(RAND()>$B$15,1,0)
20				
21	**Total ups**	0	1	<-- =SUM(C18:C19)
22	**Terminal stock price**	40.5	63	<-- =$B$2*up^C21*down^(2-C21)
23	**Option payoff**	0	13	<-- =MAX(C22-$B$3,0)
24				
25	**Average discounted payoff**	5.8957	<-- =AVERAGE(23:23)/R_^2	
26				
27	**Computing the actual option price with state prices**			
28	**Payoffs**			
29	top	48	<-- =MAX(B2*up^2-B3,0)	
30	middle	13	<-- =MAX(B2*up*down-B3,0)	
31	bottom	0	<-- =MAX(B2*down^2-B3,0)	
32	Actual option price	8.8707	<-- =q_up^2*B29+2*q_up*q_down*B30+q_down^2*B31	

The state prices and risk-neutral probabilities are computed in cells B10:B11 and B14:B15. Cells B18:B19 and C18:C19 show two random price paths. In each period we use a random number from Excel generated by the function **Rand**. If **Rand( ) >** π_D , then the stock price goes Up, and if **Rand() ≤** π_D then the stock price goes Down.

- In the first price path (cells B18:B19), the stock price goes down in both periods. The terminal stock price is 40.5 and the option's payoff is 0 (cell B23).
- In the second price path (cells C18:C19), the stock price goes up in the first period and down in the second period. The terminal stock price is 63 and the option's payoff is 13 (cell C23).

If these were the only two random price paths, the Monte Carlo option price would be the discounted average 5.8957 (cell B25). Notice that we have also computed the *actual call price* using the state prices; cell B32 shows this price to be 8.8707.

Risk-Neutral Probabilities in Monte Carlo

Notice the role of the risk-neutral probabilities in the Monte Carlo simulation: The price path is determined *not by the actual probabilities*, but by the risk-neutral probabilities π_U and π_D. There is no role for the actual probabilities in the Monte Carlo pricing.

Extending the Two-Period Model

You can't, of course, run a Monte Carlo simulation by using only two price paths. In the spreadsheet below, we've extended our price paths to all the columns in the spreadsheet. Excel's spreadsheet is 256 columns wide; this means that the computation in cell B25 is the average of the option value over 255 simulated price paths.

	A	B	C	D	E	F	G	H	I
1	**SIMPLE SIMULATION: A TWO-DATE MODEL** **as many price paths as there are columns** **pressing F9 runs the simulation and will change the value in cell B25** **This value should be compared to the actual option price in cell B32**								
2	Initial stock price	50							
3	X	50							
4									
5	Up	1.4							
6	Down	0.9							
7	R	1.05							
8									
9	**State prices**								
10	q_u	0.2857	<-- =(B7-B6)/(B7*(B5-B6))						
11	q_d	0.6667	<-- =(B7-B6)/(B7*(B5-B6))						
12									
13	**Risk-neutral probabilities**								
14	π_u	0.3000	<-- =(B7-B6)/(B7*(B5-B6))						
15	π_d	0.7000	<-- =(B7-B6)/(B7*(B5-B6))						
16									
17	**Random paths and the Monte Carlo price**								
18	First period, up (1) or down (0)?	0	0	1	0	0	0	1	1
19	Second period, up (1) or down (0)?	0	1	1	0	1	0	0	0
20									
21	**Total ups**	0	1	2	0	1	0	1	1
22	**Terminal stock price**	40.5	63	98	40.5	63	40.5	63	63
23	**Option payoff**	0	13	48	0	13	0	13	13
24									
25	**Average discounted payoff**	**8.7502**	<-- =AVERAGE(23:23)/R_^2						
26									
27	**Computing the actual option price with state prices**								
28	**Payoffs**								
29	top	48	<-- =MAX(B2*up^2-B3,0)						
30	middle	13	<-- =MAX(B2*up*down-B3,0)						
31	bottom	0	<-- =MAX(B2*down^2-B3,0)						
32	**Actual option price**	**8.8707**	<-- =q_up^2*B29+2*q_up*q_down*B30+q_down^2*B31						

The average discounted payoff (cell B25) is 8.7502. This value is random, meaning it will change each time we press **F9** to produce a new set of random paths. Monte Carlo methods imply that for even more paths we would converge to the actual option price of 8.8707. In the next section we show that the Monte Carlo method eventually produces convergence to this price.

30.5 Monte Carlo Plain-Vanilla Call Pricing Converges to Black-Scholes

Now that we understand the principles, we extend our logic. We write a VBA routine which prices a plain-vanilla call using Monte Carlo methods under conditions which converge to Black-Scholes pricing.

Our basic setup is as follows: We price a European call on a stock whose current price is S_0. The option's exercise price is X, and the time to maturity of the option is T. We assume that the stock price is lognormally distributed with mean μ and standard deviation σ.

To price the call using Monte Carlo:

- We divide the unit time interval into n divisions. This means that $\Delta t = 1/n$.
- For each Δt, we define $Up_{\Delta t} = \exp\left[\mu\Delta t + \sigma\sqrt{\Delta t}\right]$ and $Down_{\Delta t} = \exp\left[\mu\Delta t - \sigma\sqrt{\Delta t}\right]$. The interest rate on the interval Δt is $R_{\Delta t} = \exp[r\Delta t]$.
- This means that the state prices and risk-neutral probabilities are given by

$$q_u = \frac{R_{\Delta t} - Down_{\Delta t}}{R_{\Delta t}(Up_{\Delta t} - Down_{\Delta t})}, \quad q_d = \frac{Up_{\Delta t} - R_{\Delta t}}{R_{\Delta t}(Up_{\Delta t} - Down_{\Delta t})}$$

$$\pi_u = \frac{R_{\Delta t} - Down_{\Delta t}}{Up_{\Delta t} - Down_{\Delta t}}, \qquad \pi_d = \frac{Up_{\Delta t} - R_{\Delta t}}{Up_{\Delta t} - Down_{\Delta t}} = 1 - \pi_u$$

- Since the time to maturity of the option is T, the price path to T requires $m = T/\Delta t$ periods. A price path of length m is created by determining the Up or Down move of the stock as a function of a random number between 0 and 1 and the risk-neutral probability π_d. As discussed in the example in section 30.3, if the random number is greater than π_d, the stock makes an Up move; otherwise it makes a Down move.

A VBA Routine

The VBA routine below defines a function **VanillaCall**. This function requires as inputs the variables mentioned above. The variable **Runs** is the number of random price paths created; these paths are averaged to determine the Monte Carlo value of the call:

```
Function VanillaCall(S0, Exercise, Mean, sigma, _
Interest, Time, Divisions, Runs)
   deltat = 1 / Divisions
   interestdelta = Exp(Interest * deltat)

   up = Exp(Mean * deltat + _
   sigma * Sqr(deltat))
   down = Exp(Mean * deltat - _
   sigma * Sqr(deltat))

   pathlength = Int(Time / deltat)

'Risk-neutral probabilities
piup = (interestdelta - down) / _
(up - down)
pidown = 1 - piup

Temp = 0
For Index = 1 To Runs
   Upcounter = 0
   'Generate terminal price
   For j = 1 To pathlength
   If Rnd > pidown Then Upcounter = _
   Upcounter + 1
   Next j
   callvalue = Application.Max(S0 * _
   (up ^ Upcounter) * (down ^ (pathlength - _
   Upcounter)) - Exercise, 0) _
```

```
    / (interestdelta ^ pathlength)
    Temp = Temp + callvalue

Next Index
VanillaCall = Temp / Runs
End Function
```

The number of Up moves is stored in a counter called **Upcounter** and the value of the call for each **Run** is the discounted value of the call payoff for a particular terminal price $S_0 * Up^{Upcounter} Down^{pathlength-Upcounter}$, where `pathlength = Int(Time / deltat)` is the integer part of $T/\Delta t$:

```
callvalue = Application.Max(S0 * (up ^
Upcounter) * _
    (down ^ (pathlength - Upcounter)) _
    - Exercise, 0) / (interestdelta ^
    pathlength)
```

The Monte Carlo value of the call is given by: `VanillaCall = Temp / Runs`.

Understanding the Principles of the Monte Carlo Simulation

For future reference we state the principles of the Monte Carlo simulation. These principles hold not only for the plain-vanilla options of this section, but also for the Asian options treated later in this chapter:

- Price paths are generated by using the risk-neutral probabilities. In the program **VanillaCall**, for example, the price of the stock moves Up if the random number generator > π_D and moves Down if the random number generator ≤ π_D. Effectively this means that the risk-neutral probabilities $\{\pi_U = 1 - \pi_D, \pi_D\}$ of each price path are incorporated into the price path itself.
- The value of the option using Monte Carlo is determined by the discounted value of the simple average of all results over the price paths generated.

Implementing the MC Function VanillaCall in a Spreadsheet

The spreadsheet below shows the implementation of **VanillaCall**. The value in cell B14 is the option value as computed by the Black-Scholes formula; the function **BSCall** was defined in Chapter 19.

	A	B	C
1	**MONTE CARLO PRICING OF PLAIN-VANILLA CALLS**		
2	S_0 , current stock price	50	
3	X, exercise price	50	
4	r, interest rate	10%	
5	T, time	0.8	
6	μ, mean stock return	33%	
7	σ, sigma--standard deviation of stock return	30%	
8			
9	n, divisions of unit time	200	
10	Runs	3,000	
11			
12	VanillaCall	7.5861	<-- =vanillacall(B2,B3,B6,B7,B4,B5,B9,B10)
13			
14	BS call	7.2782	<-- =BSCall(B2,B3,B5,B4,B7)

The function divides the time to option expiration $T = 0.8$ (cell B5) into 100 divisions (cell B9), so that $\Delta t = 1/200$. Each time the function is called, it runs 3,000 price paths (cell B10). A particular call of the function shown above produced the value 7.5861 (cell B12), whereas the Black-Scholes call value—computed with the function **BSCall** defined in Chapter 19—is 7.2782 (cell B14).

How good is this MC routine? One way to test it is to run it many times. In the spreadsheet below, we've run 40 instances of the function **VanillaCall**.

	A	B	C	D	E
1	**RUNNING THE MONTE CARLO FUNCTION MANY TIMES**				
2	S_0 , current stock price	50			
3	X, exercise price	50			
4	r, interest rate	10%			
5	T, time	0.8			
6	μ, mean stock return	33%			
7	σ, sigma--standard deviation of stock return	30%			
8					
9	n, divisions	100			
10	Runs	3,000			
11					
12	VanillaCall	6.9059	<-- =vanillacall(B2,B3,B6,B7,B4,B5,B9,B10)		
13					
14	BS call	7.2782	<-- =BSCall(B2,B3,B5,B4,B7)		
15					
16		**Multiple runs of the function**			
17	7.4154	7.0526	7.2989	7.3993	<--
18	7.2086	7.1906	7.4140	7.2986	=vanillacall(B2,B3,B6,
19	7.1909	7.0336	7.2967	7.4546	B7,B4,B5,B9,B10)
20	7.5312	7.3078	7.5679	7.2017	
21	7.1911	7.0003	6.8358	7.1334	
22	7.2903	7.3150	7.3853	7.0925	
23	7.6109	7.3928	7.1644	7.4126	
24	7.6070	7.2575	7.1837	7.2717	
25	7.2296	7.5758	7.1568	7.4572	
26	7.3825	7.0926	7.1412	7.1313	
27					
28	7.2793	<-- =AVERAGE(A17:D26)			
29	0.1765	<-- =STDEV.S(A17:D26)			

The average value (cell A28) has a relatively low standard deviation (cell A29). The Monte Carlo routine works pretty well.

Improving the Efficiency of the MC Routine

Monte Carlo routines are inherently very wasteful—you have to run them many times to get a reasonable approximation to the true value. Thus there is a lot of mileage in making a particular routine more efficient. Continuing with our **VanillaCall** example, we show one example of such an efficiency gain.

Suppose that after j random numbers the random price is such that there is no chance that the call option will be in the money. Denote the number of Ups after j random coin tosses by $Upcounter(j)$. Then the call option cannot be in the money after n random numbers if $S_0 Up^{Upcounter(j)+(n-j)} Down^{j-Upcounter(j)} < X$. This formula assumes that all the remaining random numbers (there will be $n-j$ such numbers) will give an Up stock price movement.

For this case, we should stop choosing random numbers after *j* and let the call value be zero. The VBA routine below implements this logic:

```
Function BetterVanillaCall(S0, Exercise, Mean, _
sigma, Interest, Time, Divisions, Runs)
    deltat = Time / Divisions
    interestdelta = Exp(Interest * deltat)

up = Exp(Mean * deltat + sigma * Sqr(deltat))
down = Exp(Mean * deltat - sigma * Sqr(deltat))

pathlength = Int(Time / deltat)

'Risk-neutral probabilities
piup = (interestdelta - down)/(up - down)
pidown = 1 - piup

Temp = 0

For Index = 1 To Runs
    Upcounter = 0
    'Generate terminal price
    For j = 1 To pathlength
    If Rnd > pidown Then Upcounter = _
    Upcounter + 1
    If S0 * up ^ (Upcounter + pathlength - j) _
    * down ^ (j - Upcounter) < X _
    Then GoTo Compute
    Next j
Compute:
    callvalue = Application.Max(S0 * _
    (up ^ Upcounter) * (down ^ _
    (pathlength - Upcounter)) _
```

```
    - Exercise, 0) / (interestdelta _
    ^ pathlength)
    Temp = Temp + callvalue

Next Index
BetterVanillaCall = Temp / Runs
End Function
```

The highlighted portions of the code show the changes. The lines called **Compute** simply calculate the **Callvalue**.

The spreadsheet below shows the implementation:

	A	B	C
1	**MONTE CARLO PRICING OF PLAIN-VANILLA CALLS** **BetterVanillaCall: A somewhat more efficient function: If, after *j* random numbers which produce *k* Up moves, S_0*$Up^{(k+n-j)}$*$Down^{(j-k)}$<X, then we abort the random price path and let the call value = 0**		
2	S_0 , current stock price	50	
3	X, exercise price	45	
4	r, interest rate	6%	
5	T, time	0.8	
6	μ, mean stock return	12%	
7	σ, sigma--standard deviation of stock return	30%	
8			
9	n, divisions	100	
10	Runs	2,000	
11			
12	VanillaCall	9.0001	<-- =bettervanillacall(B2,B3,B6,B7,B4,B5,B9,B10)
13			
14	BS call	9.2931	<-- =BSCall(B2,B3,B5,B4,B7)

Where Do We Go from Here?

Now that we understand the Monte Carlo technology and its implementation in VBA, we can extend our examples in two directions. In the next section we discuss the pricing of Asian options—options in which the option's terminal payoff depends on the average price over the path. In section 30.6 we discuss the pricing of barrier options.

30.6 Pricing Asian Options

An Asian option is an option whose payoff depends in some way on the average price of the asset over a period of time prior to option expiration.[3] Asian options are sometimes called "average price options." There are two common kinds of Asian options:

- In the first kind of Asian option, the option's payoff is based on the difference between the average price of the underlying asset and the strike price: Max[Average underlying – Strike, 0]. The examples in Figures 30.3 and 30.4 of the oil contract traded on the NYMEX and the traded average price option (TAPO) traded on the London Metals Exchange are this kind of option.
- In the second kind of Asian option, the option's exercise price is the average of the underlying asset's price over a period preceding option maturity: Max[Terminal underlying – average underlying, 0]. Such *average strike* options are common in markets for electric energy. They assist hedgers whose primary risks are related to the average price of the underlying.

Asian options are particularly useful when the user sells the underlying during the period and is therefore exposed to the average price and when there is danger of price manipulation in the underlying. The Asian option mitigates the effect of manipulation, since it is based not on a single price, but on a sequence of prices.

3. See www.riskglossary.com/articles/asian_option.htm and www.global-derivatives.com/options/asian-options.php for some definitions and a discussion of the literature. A list of references is also given in the Selected References section of this book.

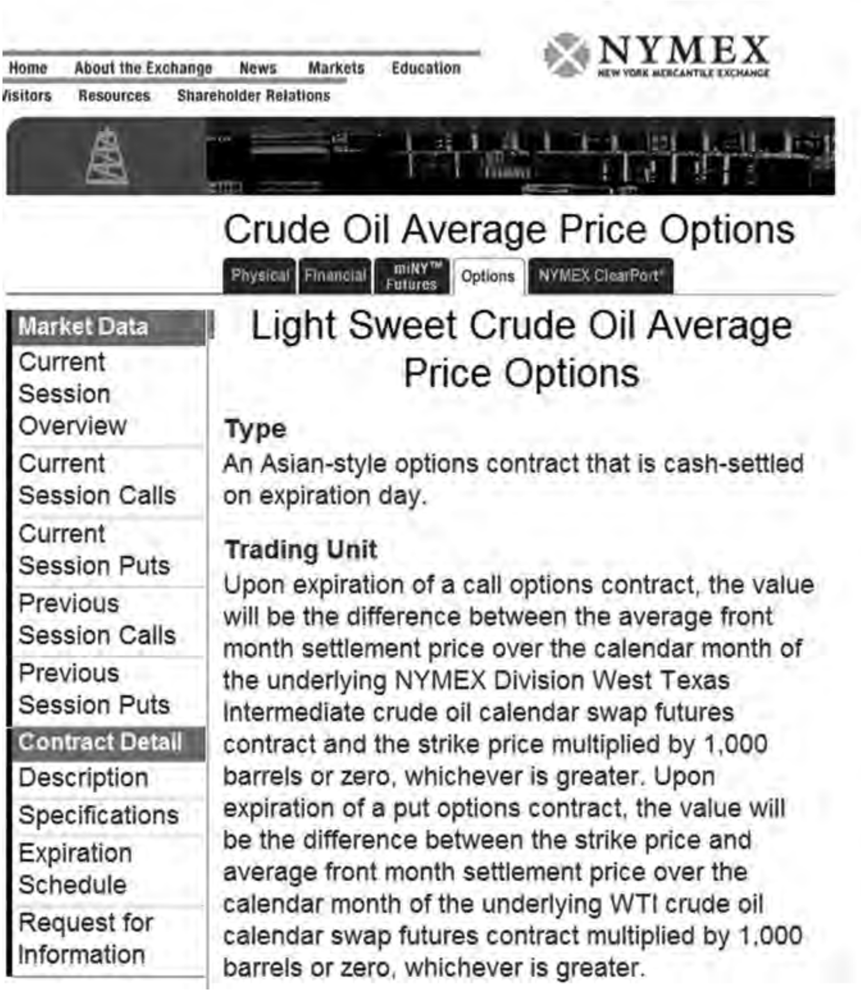
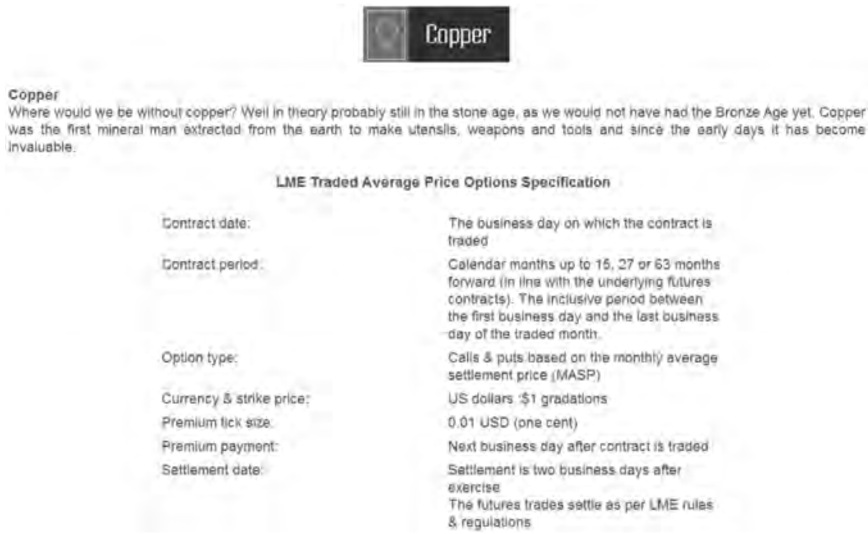


Figure 30.3
Average price crude oil options traded on NYMEX. www.nymex.com/AO_spec.aspx



Copper

Copper

Where would we be without copper? Well in theory probably still in the stone age, as we would not have had the Bronze Age yet. Copper was the first mineral man extracted from the earth to make utensils, weapons and tools and since the early days it has become invaluable.

LME Traded Average Price Options Specification

Contract date:	The business day on which the contract is traded
Contract period:	Calendar months up to 15, 27 or 63 months forward (in line with the underlying futures contracts). The inclusive period between the first business day and the last business day of the traded month.
Option type:	Calls & puts based on the monthly average settlement price (MASP)
Currency & strike price:	US dollars $1 gradations
Premium tick size:	0.01 USD (one cent)
Premium payment:	Next business day after contract is traded
Settlement date:	Settlement is two business days after exercise The futures trades settle as per LME rules & regulations

Figure 30.4
Asian options on copper traded on the London Metals Exchange (LME). www.basemetals.com/html/cuinfo.htm

An Initial Example of an Asian Option

We start by considering an Asian option on a stock whose price either increases by 40% or decreases by 20% each period. We look at five dates, starting with date 0:

	A	B	C	D	E	F	G	H	I	J
1	**ASIAN OPTION PICTURE**									
2	Initial stock price	30								
3	Up	1.4								
4	Down	0.8								
5	R, 1+interest rate	1.08								
6	Exercise price	50								
7										
8	**State prices**									
9	q_u	0.4321	<-- =(B5-B4)/(B5*(B3-B4))							
10	q_d	0.4938	<-- =(B3-B5)/(B5*(B3-B4))							
11										
12	**Risk-neutral probabilities**									
13	π_u	0.4667	<-- =B9*$B$5							
14	π_d	0.5333	<-- =B10*$B$5							
15										
16										
17									115.25	<-- =G18*B3
18	**Stock price**						82.32			
19					58.80				65.86	<-- =G18*B4
20			42.00				47.04			
21	30.00				33.60				37.63	<-- =G20*B4
22			24.00				26.88			
23					19.20				21.50	<-- =G22*B4
24							15.36			
25									12.29	<-- =G24*B4
26										
27										
28									1.3605	<-- =G29*$B$5
29		**Bond price**					1.2597			
30					1.1664				1.3605	<-- =G31*$B$5
31			1.0800				1.2597			
32	1.0000				1.1664				1.3605	
33			1.0800				1.2597			
34					1.1664				1.3605	
35							1.2597			
36									1.3605	

To compute the value of the option, we first compute each *price path*. There are 16 such paths. The spreadsheet below shows each path, the average stock price over the path, the option payoff, and the path's risk-neutral probability:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**PRICING AN ASIAN OPTION BY PRICING ALL THE PATHS**															
2	Initial stock price	30														
3	Up	1.40														
4	Down	0.80														
5	Interest	1.08														
6	Option exercise price	30														
7													Formula in cell O16: =$B$11^4			
8	State price, Up: q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))										Formula in cell O18: =$B$11^3*$B$12			
9	State price, Down, q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))													
10								Formula in cell M16 =AVERAGE(G16:K16)								
11	Risk-neutral prob., Up	0.4667	<-- =B8*B5													
12	Risk-neutral prob., Down	0.5333	<-- =B9*B5										Formula in cell N16 =MAX(M16-$B$6,0)			
13																
14									**STOCK PRICE**							
15	**Paths**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Period 0**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Average stock price**	**Option payoff**	**Path risk-neutral probability**	
16	All up (1 path)	up	up	up	up		30.00	42.00	58.80	82.32	115.25		65.67	35.67	0.0474	
17																
18	One down (4 paths)	down	up	up	up		30.00	24.00	33.60	47.04	65.86		40.10	10.10	0.0542	
19		up	down	up	up		30.00	42.00	33.60	47.04	65.86		43.70	13.70	0.0542	
20		up	up	down	up		30.00	42.00	58.80	47.04	65.86		48.74	18.74	0.0542	
21		up	up	up	down		30.00	42.00	58.80	82.32	65.86		55.80	25.80	0.0542	
22																
23	Two down (6 paths)	down	down	up	up		30.00	24.00	19.20	26.88	37.63		27.54	0.00	0.0619	
24		down	up	down	up		30.00	24.00	33.60	26.88	37.63		30.42	0.42	0.0619	
25		down	up	up	down		30.00	24.00	33.60	47.04	37.63		34.45	4.45	0.0619	
26		up	down	down	up		30.00	42.00	33.60	26.88	37.63		34.02	4.02	0.0619	
27		up	up	down	down		30.00	42.00	58.80	47.04	37.63		43.09	13.09	0.0619	
28		up	down	up	down		30.00	42.00	33.60	47.04	37.63		38.05	8.05	0.0619	
29																
30	Three down (4 paths)	up	down	down	down		30.00	42.00	33.60	26.88	21.50		30.80	0.80	0.0708	
31		down	up	down	down		30.00	24.00	33.60	26.88	21.50		27.20	0.00	0.0708	
32		down	down	up	down		30.00	24.00	19.20	26.88	21.50		24.32	0.00	0.0708	
33		down	down	down	up		30.00	24.00	19.20	15.36	21.50		22.01	0.00	0.0708	
34																
35	Four down (1 path)	down	down	down	down		30.00	24.00	19.20	15.36	12.29		20.17	0.00	0.0809	
36																
37													Option value	5.3756		
38																
39																
40									Formula in cell N37 =SUMPRODUCT(N16:N35,O16:O35)/B5^4							
41																

The paths are determined by the number and the sequence of the *Up* and the *Down* movements of the stock. For explanatory purposes we have highlighted two price paths:

• Along the path {up, down, up, up} the terminal stock price is 65.856, the average price is 43.699, the option payoff is 13.699, and the discounted expected value using the risk-neutral prices is 0.546:

	A	B	C
1	**PATH PRICE EXAMPLE: {Up, Down, Up, Up}**		
2	Initial stock price	30	
3	Up	1.40	
4	Down	0.80	
5	Interest	1.08	
6	Option exercise price	30	
7			
8	State price, Up: q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))
9	State price, Down, q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))
10			
11	Risk-neutral prob., Up	0.4667	<-- =B8*B5
12	Risk-neutral prob., Down	0.5333	<-- =B9*B5
13			
14	**Date**	**Price at beginning of period**	**Price movement: Up or Down**
15	0	30.000	
16	1	42.000	Up
17	2	33.600	Down
18	3	47.040	Up
19	4	65.856	Up
20	Average price along path	43.699	<-- =AVERAGE(B15:B19)
21	Option payoff at path end	13.699	<-- =MAX(B20-$B$6,0)
22	Path risk-neutral price	0.0542	<-- =B11^COUNTIF(C16:C19,"Up")*B12^COUNTIF(C16:C19,"Down")
23	Value of path: Payoff * risk-neutral price * discount factor	0.546	<-- =B21*B22/B5^4

The average stock price along this path is 43.699, so that the option pays off max[43.699 – 30,0] = 13.699.

• Along the path {up, up, down, up} the terminal stock price is the same as before: 65.856. However, the average price and thus the option payoff and value are different:

	A	B	C
1	**PATH PRICE EXAMPLE: {Up, Up, Down, Up}**		
2	Initial stock price	30	
3	Up	1.40	
4	Down	0.80	
5	Interest	1.08	
6	Option exercise price	30	
7			
8	State price, Up: q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))
9	State price, Down, q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))
10			
11	Risk-neutral prob., Up	0.4667	<-- =B8*B5
12	Risk-neutral prob., Down	0.5333	<-- =B9*B5
13			
14	**Date**	**Price at beginning of period**	**Price movement: Up or Down**
15	0	30.000	
16	1	42.000	Up
17	2	58.800	Up
18	3	47.040	Down
19	4	65.856	Up
20	Average price along path	48.739	<-- =AVERAGE(B15:B19)
21	Option payoff at path end	18.739	<-- =MAX(B20-$B$6,0)
22	Path risk-neutral price	0.0542	<-- =B11^COUNTIF(C16:C19,"Up")*B12^COUNTIF(C16:C19,"Down")
23	Value of path: Payoff * risk-neutral price * discount factor	0.747	<-- =B21*B22/B5^4

These two paths illustrate what we mean when we say that an Asian option price is *path dependent*: Two paths—both starting at the initial stock price of 30 and ending at 65.856—have different option payoffs because the stock price average along the path is different.

This example, which we've not yet concluded, also illustrates the difficulty of pricing Asian options: Each single path must be dealt with—16 separate paths. This distinguishes Asian options from the case of ordinary options ("plain vanilla," in the jargon of option pricers); for the particular example we're considering here, a plain-vanilla option requires dealing only with five ending prices.

To price the Asian option, we attach a risk-neutral probability to each price path:

	M	N	O	P
15	**Average stock price**	**Option payoff**	**Path risk-neutral probability**	
16	65.67	35.67	0.0474	<-- =B11^4
17				
18	40.10	10.10	0.0542	<-- =$B$11^3*$B$12
19	43.70	13.70	0.0542	
20	48.74	18.74	0.0542	
21	55.80	25.80	0.0542	
22				
23	27.54	0.00	0.0619	<-- =$B$11^2*$B$12^2
24	30.42	0.42	0.0619	
25	34.45	4.45	0.0619	
26	34.02	4.02	0.0619	
27	43.09	13.09	0.0619	
28	38.05	8.05	0.0619	
29				
30	30.80	0.80	0.0708	<-- =$B$11*$B$12^3
31	27.20	0.00	0.0708	
32	24.32	0.00	0.0708	
33	22.01	0.00	0.0708	
34				
35	20.17	0.00	0.0809	<-- =$B$12^4
36				
37	Option value	5.3756	<-- =SUMPRODUCT(N16:N35,O16:O35)/B5^4	

The option price is the discounted expected payoff value, where the expectations are computed with the risk-neutral probabilities:

$$\frac{\sum_{All\ paths} \pi_{Path} * Option\ payoff\ on\ path}{R^n} = 5.3756$$

Risk-Neutral Probabilities–Again

We repeat our earlier comments (page 789) about the role of risk-neutral probabilities: Each path is priced by its discounted risk-neutral probability, which is a function of the Up, Down, and *R*. The actual state probabilities are not relevant.

30.7 Pricing Asian Options with a VBA Program

Our spreadsheet example in the previous section illustrates both the principle of Monte Carlo pricing of options and the problematics of pricing the options directly in a spreadsheet. For four periods, we require the computation of $2^4 = 16$ paths. For a more general problem with n periods, there are 2^n paths to consider; this rapidly becomes too large for even a very powerful computer. In order to accurately price the options, you would need hundreds or thousands of simulations. Doing this in a spreadsheet directly would be cumbersome. The obvious answer is to write some VBA code to automate the process and which allows us to run an arbitrarily large number of simulations.

In this section we write VBA code to do a Monte Carlo simulation of Asian pricing options. We generate price paths by simulating a sequence of Up and Down movements of the underlying stock price; the probability of Up or Down depends on the risk-neutral probabilities—in this sense our Monte Carlo simulation for Asian options is similar to that illustrated in section 30.4 for plain-vanilla options. For each price path generated, we calculate the option payoff, and after generating a large number of price paths, we compute the option price by discounting and averaging these payoffs. The VBA function, **MCAsian**, which prices the Asian options, is given below:

```
Function MCAsian(initial, Exercise, Up, Down, _
   Interest, Periods, Runs)
   Dim PricePath() As Double
   ReDim PricePath(Periods + 1)

   'Risk-neutral probabilities
   piup = (Interest - Down) / (Up - Down)
   pidown = 1 - piup

   Temp = 0

   For Index = 1 To Runs
      'Generate path
      For i = 1 To Periods
         PricePath(0) = initial
```

```
            pathprob = 1
            If Rnd > pidown Then
              PricePath(i) = PricePath(i - 1) _
              * Up
              Else:
              PricePath(i) = PricePath(i - 1) * _
              Down
            End If
         Next i

        PriceAverage = Application.Sum _
        (PricePath) / (Periods + 1)
            callpayoff = Application.Max _
        (PriceAverage - Exercise, 0)
            Temp = Temp + callpayoff

    Next Index

    MCAsian = (Temp / Interest ^ Periods) / _
        Runs

End Function
```

Here's the implementation of the function in a spreadsheet:

	A	B	C
1	**PRICING AN ASIAN OPTION BY MONTE CARLO**		
2	Up	1.4	
3	Down	0.8	
4	Interest	1.08	
5	Initialprice	30	
6	Periods	20	
7	Exercise	30	
8	Runs	500	
9	Asian call value	9.7253	<-- =MCAsian(B5,B7,B2,B3,B4,B6,B8)

The function in cell B9 is our Monte Carlo valuation of the option—the simulated value. Any recalculation of the spreadsheet will cause the function to rerun and recompute the option value.

Below we show a block of replications of the function. Each of the cells A10:F17 contains **=MCAsian(Initialprice,exercise,Up,Down,Interest, Periods,Runs)**, so that we calculate 48 simulations of the option value.

	A	B	C	D	E	F	G
1	**PRICING AN ASIAN OPTION--VBA FUNCTION** **Prices an Asian option with 4 periods and 100 runs for each simulation**						
2	Up	1.4					
3	Down	0.8					
4	Interest	1.08					
5	Initialprice	30					
6	Periods	4					
7	Exercise	30					
8	Runs	100					
9							
10	5.5604	5.5127	5.3137	5.9568	4.9349	5.7926	<--
11	6.2343	4.7683	5.8404	5.5072	5.9239	5.0891	=MCAsian($B$5,$B$7,$B$2
12	5.3737	5.3555	4.2552	4.8458	5.3619	6.3329	,$B$3,$B$4,$B$6,$B$8)
13	4.3899	6.0211	5.4461	5.5281	5.7560	6.2756	
14	4.8685	5.3166	4.5529	4.8597	5.3485	5.8267	
15	6.0235	4.5321	4.4843	4.8879	4.9199	5.3249	
16	5.2095	4.7944	5.4976	4.4916	5.6223	5.3322	
17	5.2452	5.6684	5.4797	5.2801	5.9564	3.8410	
18							
19	Average of MC simulations	5.3071	<-- =AVERAGE(A10:F17)				
20	True value	5.3756	<-- From Section 30.6				
21							
22		3.8410	<-- =MIN(A10:F17)				
23		6.3329	<-- =MAX(A10:F17)				
24		0.5631	<-- =STDEV.S(A10:F17)				

We have deliberately priced the Asian option for which we know the true value—as we showed in section 30.6, the value of an Asian option in a four-period model with the parameters for Up, Down, and Interest illustrated above, is 5.3756. The average of our Monte Carlo simulations is 5.3071, with a standard deviation of 0.5631.

When we increase the number of runs (cell B8), we will usually decrease the standard deviation of our estimates (cell B24), which is equivalent to increasing the accuracy of the simulation. In the example below, we have run the 48 simulations 500 runs each:

	A	B	C	D	E	F
1	**PRICING AN ASIAN OPTION--VBA FUNCTION Prices an Asian option with 4 periods and 500 runs for each simulation**					
2	Up	1.4				
3	Down	0.8				
4	Interest	1.08				
5	Initialprice	30				
6	Periods	4				
7	Exercise	30				
8	Runs	500				
9						
10	5.4918	5.2237	5.4857	5.6620	5.0970	5.3702
11	6.0799	5.2819	5.8147	5.4304	5.5054	5.0958
12	5.5832	5.7454	5.0490	5.5856	5.0470	5.8236
13	5.3436	5.3475	5.5037	5.4204	5.2500	6.1101
14	4.9375	5.3760	5.3095	5.4251	5.4096	5.1571
15	5.5058	5.2473	5.1213	6.1103	5.2689	5.0927
16	5.6943	5.2531	5.0314	5.7414	4.8099	5.0938
17	5.0082	5.3070	5.3588	5.1692	5.7047	4.8963
18						
19	Average of MC simulations	5.3828	<-- =AVERAGE(A10:F17)			
20	True value	5.3756	<-- From Section 30.6			
21						
22		4.8099	<-- =MIN(A10:F17)			
23		6.1103	<-- =MAX(A10:F17)			
24		0.3088	<-- =STDEV.S(A10:F17)			

As you can see, the standard deviation is much reduced—about half of the standard deviation with 100 runs.[4]

Asian Options with More Periods

In the following spreadsheet we divide the unit time interval into n subperiods. We follow the procedure of section 30.4 for defining the returns, state prices, and risk-neutral probabilities over the subperiod Δt. Here is the resulting spreadsheet:

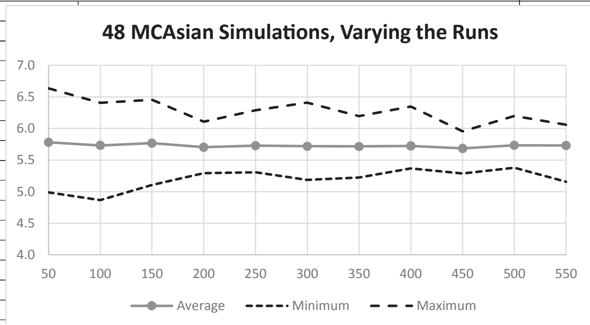
4. The bad news, of course, is that we have to increase the number of runs by a factor of 5 to reduce the standard deviation by one-half.

	A	B	C	D	E	F
1	**PRICING AN ASIAN OPTION--VBA FUNCTION** **Each time interval is divided into n subintervals. In this simulation the initial stock price = 50.00, the exercise price = 45.00, the time to maturity = 0.40, and the unit time interval is divided into 80 subintervals. The stock price process has mean return = 15.00% and standard deviation = 1.25%, and the interest rate = 8.00%.** **There are 100 runs in each Monte Carlo simulation**					
2	S_0, current stock price	50				
3	X, exercise price	45				
4	T, time to option exercise	0.4				
5	r, interest rate	8%				
6	μ, mean stock return	15%				
7	σ, standard deviation of stock return	22%				
8						
9	n, number of sub-intervals of T	80				
10	Delta t	0.0125	<-- =1/B9			
11						
12	Up over 1 sub-interval	1.0268	<-- =EXP(B6*B10+B7*SQRT(B10))			
13	Down over 1 sub-interval	0.9775	<-- =EXP(B6*B10-B7*SQRT(B10))			
14	Interest over 1 sub-interval	1.0010	<-- =EXP(B5*B10)			
15						
16	Runs	100				
17						
18	5.5148	5.1334	6.1348	5.8025	6.0843	5.8046
19	6.0585	6.1534	5.8453	6.0183	6.2981	5.3826
20	5.4753	5.9614	5.9462	5.3539	6.5194	5.6043
21	5.5941	6.0015	5.7392	6.1450	5.7562	5.8220
22	5.3215	6.0690	5.4965	5.0476	5.1488	5.7132
23	5.3827	6.0806	5.4600	6.0932	5.4499	5.6005
24	6.1500	5.7218	6.0046	6.1990	5.4302	5.2791
25	5.7279	6.2467	5.6817	5.6702	5.6883	5.2343
26						
27	Average of above	5.7510	<-- =AVERAGE(A18:F25)			
28	Minimum	5.0476	<-- =MIN(A18:F25)			
29	Maxmimum	6.5194	<-- =MAX(A18:F25)			
30	Standard deviation	0.3481	<-- =STDEV(A18:F25)			

The block of results in cells A18:F25 gives 48 results for running the function **MCAsian**. Below this block we give the statistics for these simulations.

The above simulations use 100 (cell B16) price paths per iteration of the function **MCAsian**; this is the number contained in cell B16. We can use **Data|Table** to see the effect of changing the number of runs:

	B	C	D	E	F	G	H	I
32	**Data table: Sensitivity of results on number of runs**							
33	**Runs**	**Average of 48 MCAsian**	**Minimum**	**Maximum**	**Standard deviation**			
34						<-- =B30 , data table header (hidden)		
35	50	5.7832	4.9871	6.6371	0.4255			
36	100	5.7343	4.8668	6.4071	0.3309			
37	150	5.7689	5.1077	6.4571	0.3145			
38	200	5.7052	5.2953	6.1088	0.2047			
39	250	5.7285	5.3057	6.2896	0.2198			
40	300	5.7208	5.1867	6.4113	0.2666			
41	350	5.7163	5.2251	6.1966	0.2093			
42	400	5.7248	5.3671	6.3500	0.1972			
43	450	5.6868	5.2898	5.9580	0.1707			
44	500	5.7353	5.3822	6.1974	0.1636			
45	550	5.7321	5.1561	6.0600	0.1808			
46								
47								
48								
49								
50								
51								



It is clear that increasing the number of runs narrows the bounds on the simulation.

30.8 Pricing Barrier Options with Monte Carlo[5]

A barrier option's payoff depends on whether the price reaches a specific level during the life of the option:

- A *knockin* barrier call option has payoff $\text{Max}(S_T - X, 0)$ only if at some time $t < T$, $S_t > \text{K}$. A knockin put has the same condition but pays off $\text{Max}(X - S_T, 0)$.
- A *knockout* barrier call or put option has these payoffs provided that at no time before T does the stock price reach the barrier.

Imposing a barrier makes it more difficult for an option to be in the money at expiration; thus barrier options have lower value than regular options.

A Simple Example of a Barrier Call Option

Below we show an extended example for a knockout barrier option which is similar to the example for an Asian option given in section 30.6:

5. Pricing barrier options with Monte Carlo isn't necessarily a good idea, but it's a good exercise. See Broadie, Glaserman, and Kou (1997) for a complete discussion.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**PRICING A KNOCKOUT BARRIER OPTION**															
2	Initial stock price	30														
3	Up	1.40														
4	Down	0.80														
5	Interest	1.08														
6	Option exercise price	30														
7	Barrier	50.00														
8																
9	q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))													
10	q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))													
11																
12	Risk-neutral probability, up	0.4667	<-- =B9*B5								Formula in cell M17: =MAX(G17:K17)<$B$7					
13	Risk-neutral probability, down	0.5333	<-- =B10*B5													
14																
15									**STOCK PRICE**							
16	**Paths**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Period 0**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Max(S_t)< Barrier?**	**Path risk-neutral probability**	**Knockout option payoff**	
17	All up (1 path)	up	up	up	up		30.00	42.00	58.80	82.32	115.25		FALSE	0.0474	0.00	
18																
19	One down (4 paths)	down	up	up	up		30.00	24.00	33.60	47.04	65.86		FALSE	0.0542	0.00	
20		up	down	up	up		30.00	42.00	33.60	47.04	65.86		FALSE	0.0542	0.00	
21		up	up	down	up		30.00	42.00	58.80	47.04	65.86		FALSE	0.0542	0.00	
22		up	up	up	down		30.00	42.00	58.80	82.32	65.86		FALSE	0.0542	0.00	
23																
24	Two down (6 paths)	down	down	up	up		30.00	24.00	19.20	26.88	37.63		TRUE	0.0619	7.63	
25		down	up	down	up		30.00	24.00	33.60	26.88	37.63		TRUE	0.0619	7.63	
26		down	up	up	down		30.00	24.00	33.60	47.04	37.63		TRUE	0.0619	7.63	
27		up	down	down	up		30.00	42.00	33.60	26.88	37.63		TRUE	0.0619	7.63	
28		up	up	down	down		30.00	42.00	58.80	47.04	37.63		FALSE	0.0619	0.00	
29		up	down	up	down		30.00	42.00	33.60	47.04	37.63		TRUE	0.0619	7.63	
30																
31	Three down (4 paths)	up	down	down	down		30.00	42.00	33.60	26.88	21.50		TRUE	0.0708	0.00	
32		down	up	down	down		30.00	24.00	33.60	26.88	21.50		TRUE	0.0708	0.00	
33		down	down	up	down		30.00	24.00	19.20	26.88	21.50		TRUE	0.0708	0.00	
34		down	down	down	up		30.00	24.00	19.20	15.36	21.50		TRUE	0.0708	0.00	
35																
36	Four down (1 path)	down	down	down	down		30.00	24.00	19.20	15.36	12.29		TRUE	0.0809	0.00	
37																
38												Knockout value		1.7375		
39																
40											Formula in cell N38: =SUMPRODUCT(O17:O36,N17:N36)/B5^4					
41																
42												Formula in cell O36: =M36*MAX(K36-$B$6,0)				

In this example, we model a five-date, four-period barrier call option. The barrier is 50 (cell B7). A knockout option pays off only if the price never goes through this barrier, and a knockin option pays off only if the stock price goes through the barrier. In an equation:

$$\begin{array}{l}\textit{Barrier knockin}\\ \textit{call payoff}\end{array} = \begin{cases} \max[S_T - X, 0] & \text{if } S_t > \textit{Barrier for } t < T \\ 0 & \textit{otherwise} \end{cases}$$

$$\begin{array}{l}\textit{Barrier knockout}\\ \textit{call payoff}\end{array} = \begin{cases} \max[S_T - X, 0] & \text{if } S_t < \textit{Barrier for } t < T \\ 0 & \textit{otherwise} \end{cases}$$

The knockout barrier call illustrated above pays off only when two things happen simultaneously:

- The stock price does not exceed the barrier. This happens for all the paths labeled "TRUE" in column M. To check this condition in cell M17 we use the Boolean function (=MAX(G17:K17)<$B$7).[6] This function evaluates to TRUE or FALSE, depending on whether the condition is met. Other cells in column M use a similar condition. When used in a formula as in the bullet below, the Boolean function evaluates to 1 if TRUE and to 0 if FALSE.
- The terminal stock price S_T is greater than the option exercise price of 30. In cell O18 we use the condition M17*MAX(K17-$B$6,0) to evaluate the option payoff.
    - If M17 = 0 (meaning that $S_t > 50$ somewhere along the path and the option was "knocked out"), then the option doesn't pay off.
    - If M17 = 1 (so that $S_t < 50$ throughout the path), then the option has a standard call payoff of $\max(S_T - X, 0)$.

As in all previous cases discussed in this chapter, the barrier call's value is the discounted expected payoff of the option, where the probabilities are the risk-neutral probabilities:

$$\textit{Option value} = \frac{\sum_{\substack{\textit{all states} \\ j}} \pi_j \textit{Payoff}_j}{R^4} = 1.7375$$

The Knockin Barrier Call

By changing the condition in column O we can price the knockin barrier call. This time we write (in cell O17, for example) the function =(1-M17)*MAX (K17-$B$6,0). The value in M17 tests whether the barrier has never been passed; if this is FALSE (i.e., has a value of zero), then the option is "knocked in" and the payoff is like that of a regular call. If M17 is TRUE, then the barrier has not been passed and the option does not pay off:

6. Boolean functions are discussed in Chapter 33.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**PRICING A KNOCKIN BARRIER OPTION**															
2	Initial stock price	30														
3	Up	1.40														
4	Down	0.80														
5	Interest	1.08														
6	Option exercise price	30														
7	Barrier	50.00														
8																
9	q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))													
10	q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))													
11																
12	Risk-neutral probability, up	0.4667	<-- =B9*B5								Formula in cell M17: =MAX(G17:K17)<$B$7					
13	Risk-neutral probability, down	0.5333	<-- =B10*B5													
14																
15									**STOCK PRICE**							
16	**Paths**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Period 0**	**Period 1**	**Period 2**	**Period 3**	**Period 4**		**Max(S_t)< Barrier?**	**Path risk-neutral probability**	**Option payoff**	
17	All up (1 path)	up	up	up	up		30.00	42.00	58.80	82.32	115.25		FALSE	0.0474	85.25	
18																
19	One down (4 paths)	down	up	up	up		30.00	24.00	33.60	47.04	65.86		FALSE	0.0542	35.86	
20		up	down	up	up		30.00	42.00	33.60	47.04	65.86		FALSE	0.0542	35.86	
21		up	up	down	up		30.00	42.00	58.80	47.04	65.86		FALSE	0.0542	35.86	
22		up	up	up	down		30.00	42.00	58.80	82.32	65.86		FALSE	0.0542	35.86	
23																
24	Two down (6 paths)	down	down	up	up		30.00	24.00	19.20	26.88	37.63		TRUE	0.0619	0.00	
25		down	up	down	up		30.00	24.00	33.60	26.88	37.63		TRUE	0.0619	0.00	
26		down	up	up	down		30.00	24.00	33.60	47.04	37.63		TRUE	0.0619	0.00	
27		up	down	down	up		30.00	42.00	33.60	26.88	37.63		TRUE	0.0619	0.00	
28		up	up	down	down		30.00	42.00	58.80	47.04	37.63		FALSE	0.0619	7.63	
29		up	down	up	down		30.00	42.00	33.60	47.04	37.63		TRUE	0.0619	0.00	
30																
31	Three down (4 paths)	up	down	down	down		30.00	42.00	33.60	26.88	21.50		TRUE	0.0708	0.00	
32		down	up	down	down		30.00	24.00	33.60	26.88	21.50		TRUE	0.0708	0.00	
33		down	down	up	down		30.00	24.00	19.20	26.88	21.50		TRUE	0.0708	0.00	
34		down	down	down	up		30.00	24.00	19.20	15.36	21.50		TRUE	0.0708	0.00	
35																
36	Four down (1 path)	down	down	down	down		30.00	24.00	19.20	15.36	12.29		TRUE	0.0809	0.00	
37																
38												Knockin value		9.0334		
39																
40											Formula in cell N38: =SUMPRODUCT(O17:O36,N17:N36)/B5^4					
41																
42												Formula in cell O36: =(1-M36)*MAX(K36-$B$6,0)				
43																

The spreadsheets for the knockout and knockin barriers illustrate another principle of pricing barrier options: The price of a knockin plus a knockout call equals the price of a plain-vanilla call:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	KNOCKIN + KNOCKOUT = PLAIN VANILLA																
2	Initial stock price	30															
3	Up	1.40															
4	Down	0.80															
5	Interest	1.08															
6	Option exercise price	30															
7	Barrier	50.00															
8																	
9	q_U	0.4321	<-- =(B5-B4)/(B5*(B3-B4))														
10	q_D	0.4938	<-- =(B3-B5)/(B5*(B3-B4))									Formula in cell M17: =MAX(G17:K17)<$B$7					
11																	
12	Risk-neutral probability, up	0.4667	<-- =B9*B5														
13	Risk-neutral probability, down	0.5333	<-- =B10*B5														
14																	
15							STOCK PRICE										
16	Paths	Period 1	Period 2	Period 3	Period 4		Period 0	Period 1	Period 2	Period 3	Period 4		$Max(S_t)<$ Barrier?	Path risk-neutral probability	Knockout payoff	Knockin payoff	Plain vanilla
17	All up (1 path)	up	up	up	up		30.00	42.00	58.80	82.32	115.25		FALSE	0.0474	0	85.2480	85.2480
18																	
19	One down (4 paths)	down	up	up	up		30.00	24.00	33.60	47.04	65.86		FALSE	0.0542	0	35.8560	35.8560
20		up	down	up	up		30.00	42.00	33.60	47.04	65.86		FALSE	0.0542	0	35.8560	35.8560
21		up	up	down	up		30.00	42.00	58.80	47.04	65.86		FALSE	0.0542	0	35.8560	35.8560
22		up	up	up	down		30.00	42.00	58.80	82.32	65.86		FALSE	0.0542	0	35.8560	35.8560
23																	
24	Two down (6 paths)	down	down	up	up		30.00	24.00	19.20	26.88	37.63		TRUE	0.0619	7.632	0.0000	7.6320
25		down	up	down	up		30.00	24.00	33.60	26.88	37.63		TRUE	0.0619	7.632	0.0000	7.6320
26		down	up	up	down		30.00	24.00	33.60	47.04	37.63		TRUE	0.0619	7.632	0.0000	7.6320
27		up	down	down	up		30.00	42.00	33.60	26.88	37.63		TRUE	0.0619	7.632	0.0000	7.6320
28		up	up	down	down		30.00	42.00	58.80	47.04	37.63		FALSE	0.0619	0	7.6320	7.6320
29		up	down	up	down		30.00	42.00	33.60	47.04	37.63		TRUE	0.0619	7.632	0.0000	7.6320
30																	
31	Three down (4 paths)	up	down	down	down		30.00	42.00	33.60	26.88	21.50		TRUE	0.0708	0	0.0000	0.0000
32		down	up	down	down		30.00	24.00	33.60	26.88	21.50		TRUE	0.0708	0	0.0000	0.0000
33		down	down	up	down		30.00	24.00	19.20	26.88	21.50		TRUE	0.0708	0	0.0000	0.0000
34		down	down	down	up		30.00	24.00	19.20	15.36	21.50		TRUE	0.0708	0	0.0000	0.0000
35																	
36	Four down (1 path)	down	down	down	down		30.00	24.00	19.20	15.36	12.29		TRUE	0.0809	0	0.0000	0.0000
37																	
38											Knockin	9.0334	<-- =SUMPRODUCT(N17:N36,P17:P36)/B5^4				
39											Knockout	1.7375	<-- =SUMPRODUCT(N17:N36,O17:O36)/B5^4				
40											Sum	10.7708	<-- =L38+L39				
41											Plain vanilla	10.7708	<-- =SUMPRODUCT(N17:N36,Q17:Q36)/B5^4				
42																	
43													Formula in cell P36: =(1-M36)*MAX(K36-$B$6,0)				
44																	
45														Formula in cell O36: =M36*MAX(K36-$B$6,0)			
46																	

30.9 Using VBA and Monte Carlo to Price a Barrier Option

We write two VBA functions to price knockin and knockout barrier options. Here is the function for knockout options:

```
Function MCBarrierIn(Initial, Exercise, Barrier, Up, _
Down, Interest, Periods, Runs)
   Dim PricePath() As Double
   ReDim PricePath(Periods + 1)

   'Risk-neutral probabilities
   piup = (Interest - Down) / (Up - Down)
   pidown = 1 - piup
```

```
    Temp = 0

    For Index = 1 To Runs
        'Generate path
        For i = 1 To Periods
            PricePath(0) = Initial
            pathprob = 1
            If Rnd > pidown Then
                PricePath(i) = PricePath(i - 1) * Up

                Else:
                PricePath(i) = PricePath(i - 1) * Down
            End If
         Next i

          If Application.Max(PricePath) > Barrier Then _
          Callpayoff = _
                Application.Max(PricePath(Periods) - _
                Exercise, 0) _
                Else Callpayoff = 0
           Temp = Temp + Callpayoff

   Next Index

   MCBarrierIn = (Temp / Interest ^ Periods) / Runs

End Function

Function MCBarrierOut(Initial, Exercise, _
Barrier, Up, Down, Interest, Periods, Runs)
    Dim PricePath() As Double
    ReDim PricePath(Periods + 1)

    'Risk-neutral probabilities
    piup = (Interest - Down) / (Up - Down)
    pidown = 1 - piup

    Temp = 0
```

```
    For Index = 1 To Runs
        'Generate path
        For i = 1 To Periods
            PricePath(0) = Initial
            pathprob = 1
            If Rnd > pidown Then
            PricePath(i) = PricePath(i - 1) * Up

            Else:
            PricePath(i) = PricePath(i - 1) * Down
            End If
        Next i

        If Application.Max(PricePath) < Barrier _
        Then Callpayoff = Application.Max _
        (PricePath(Periods) - Exercise, 0) _
        Else: Callpayoff = 0
        Temp = Temp + Callpayoff

    Next Index

    MCBarrierOut = (Temp / Interest ^ Periods) _
    / Runs

End Function
```

Since this function is very similar to the function **MCAsian** of section 30.6, we will not discuss it, except to point out that the operative part for the "knockin" option is contained in the following lines (note the use of Excel's **Max** function—in the form of **Application.Max**; VBA does not have its own maximum function):

```
If Application.Max(PricePath) < Barrier Then
Callpayoff = _
   Application.Max(PricePath(Periods) -
   Exercise, 0) _
   Else Callpayoff = 0
```

In the spreadsheet below we use this function and its associated function **MCBarrierOut** to price the options previously priced in our extensive example:

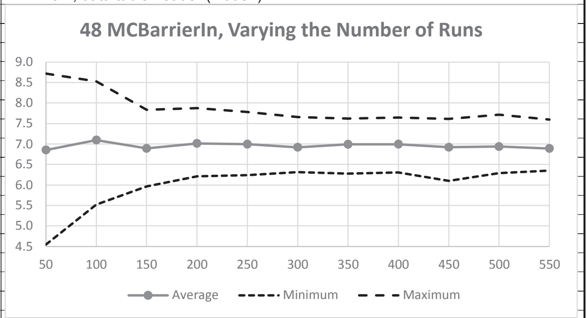
	A	B	C	D	E	F
1	**PRICING BARRIER OPTIONS BY MONTE CARLO**					
2	Up	1.4				
3	Down	0.8				
4	Interest	1.08				
5						
6	Initialprice	30				
7	Periods	4				
8	Exercise	30				
9	Barrier	50				
10						
11	Runs	100				
12						
13	Knockin option value	7.0810	<-- =mcbarrierin(B6,B8,B9,B2,B3,B4,B7,B11)			
14	Actual value	9.0334	<-- ='Initial knockin'!N38			
15						
16	Knockout option value	1.6829	<-- =mcbarrierout(B6,B8,B9,B2,B3,B4,B7,B11)			
17	Actual value	1.7375	<-- Determined from fully-worked out example			
18						
19	**48 iterations of MCBarrierIn**					
20	7.7944	7.2408	8.4210	9.2159	6.8481	13.1353
21	11.6143	8.3860	9.5916	9.1598	7.6430	11.3200
22	9.3154	6.7095	7.2673	9.2593	9.7948	8.9174
23	11.0269	10.1018	7.4959	7.5435	9.3069	9.3799
24	11.7265	8.4210	6.5369	10.6289	8.8613	8.6327
25	9.5016	9.6350	9.1037	7.9066	8.2262	8.9565
26	10.9358	7.4440	9.2550	12.1806	8.1056	8.9216
27	6.3771	7.4440	9.4318	9.3323	9.5831	9.1428
28						
29	Average of simulations	9.0162	<-- =AVERAGE(A20:F27)			
30	True value	9.0334	<-- =B14			
31						
32		6.3771	<-- =MIN(A20:F27)			
33		13.1353	<-- =MAX(A20:F27)			
34		1.5066	<-- =STDEV(A20:F27)			

Finally, we can also show the implementation of the functions **MCBarrierIn** and **MCBarrierOut** for the case where the unit period is divided into n subperiods:

	A	B	C	D	E	F
1	**PRICING BARRIER OPTIONS--VBA FUNCTION** **Each time interval is divided into n subintervals. In this simulation the initial stock price = 50.00, the exercise price = 45.00, the time to maturity = 0.40, and the unit time interval is divided into 80 subintervals. The stock price process has mean return = 15.00% and standard deviation = 1.25%, and the interest rate = 8.00%. There are 100 runs in each Monte Carlo simulation**					
2	S_0, current stock price	50				
3	X, exercise price	45				
4	Barrier	50				
5	T, time to option exercise	0.4				
6	r, interest rate	8%				
7	μ, mean stock return	15%				
8	σ, standard deviation of stock return	22%				
9						
10	n, number of sub-intervals of 1 period	80				
11	Delta t	0.0125	<-- =1/B10			
12						
13	Up over 1 sub-interval	1.0268	<-- =EXP(B7*B11+B8*SQRT(B11))			
14	Down over 1 sub-interval	0.9775	<-- =EXP(B7*B11-B8*SQRT(B11))			
15	Interest over 1 sub-interval	1.0010	<-- =EXP(B6*B11)			
16						
17	Runs	100				
18						
19	6.3585	6.9583	6.3533	7.7962	6.1659	7.0748
20	7.6459	6.6480	7.3670	7.0983	7.6145	7.5518
21	7.5930	6.0479	7.4641	6.9369	7.3640	6.8484
22	6.6697	6.8640	7.5060	5.5781	6.5915	6.8637
23	6.5892	6.0845	7.4208	6.6688	6.2547	7.0869
24	6.7855	6.8115	7.4927	8.0739	6.5437	6.8094
25	7.1895	6.7203	6.4660	7.2249	7.4684	6.1946
26	7.4440	7.3391	5.8421	7.1641	6.9199	6.3297
27						
28	Average of above	6.9142	<-- =AVERAGE(A19:F26)			
29	Minimum	5.5781	<-- =MIN(A19:F26)			
30	Maxmimum	8.0739	<-- =MAX(A19:F26)			
31	Standard deviation	0.5542	<-- =STDEV(A19:F26)			

As for the case discussed in section 30.6 for Asian options, as the number of runs (cell B17) gets larger, the approximations become better, although the improvement is not dramatic:

	B	C	D	E	F	G	H
34	**Runs**	**Average of 48 MCBarrierIn**	**Minimum**	**Maximum**	**Standard deviation**		
35						<-- =B31 , data table header (hidden)	
36	50	6.9429	4.9848	8.3475	0.8578		
37	100	6.8413	5.6721	8.6888	0.6402		
38	150	6.8778	6.0454	8.1278	0.5073		
39	200	6.9010	5.9834	8.0931	0.5616		
40	250	7.0201	5.6747	7.8031	0.4348		
41	300	6.8057	5.7971	7.4750	0.3576		
42	350	6.9462	6.2091	7.5758	0.3026		
43	400	6.9964	6.1380	7.6593	0.3181		
44	450	6.9201	6.0998	7.6141	0.3294		
45	500	6.9382	6.2906	7.7124	0.3502		
46	550	6.8899	6.3473	7.5942	0.2355		
47							
48							
49							
50							
51							
52							



Finally, we can show that the sum of the knockin plus knockout is approximately equal to the Black-Scholes call when *n*, the number of divisions of the unit time interval, is very large:

	A	B	C
1	**KNOCKIN + KNOCKOUT = CALL The almost-continuous case**		
2	S_0, current stock price	30	
3	X, exercise price	30	
4	Barrier	40	
5	T, time to option exercise	0.4	
6	r, interest rate	8%	
7	μ, mean stock return	15%	
8	σ, standard deviation of stock return	22%	
9			
10	n, number of subintervals of 1 period	200	
11	Delta t	0.0050	<-- =1/B10
12			
13	Up over 1 subinterval	1.0164	<-- =EXP(B7*B11+B8*SQRT(B11))
14	Down over 1 subinterval	0.9853	<-- =EXP(B7*B11-B8*SQRT(B11))
15	Interest over 1 subinterval	1.0004	<-- =EXP(B6*B11)
16			
17	Runs	700	
18			
19	Knockout barrier	1.653148	<-- =mcbarrierout(B2,B3,B4,B13,B14,B15,INT(B10*B5),B17)
20	Knockin barrier	0.504321	<-- =mcbarrierin(B2,B3,B4,B13,B14,B15,INT(B10*B5),B17)
21	Sum of knockout + knockin	2.157468	<-- =B19+B20
22	Black-Scholes call price	2.153173	<-- =BSCall(B2,B3,B5,B6,B8)

30.10 Summary

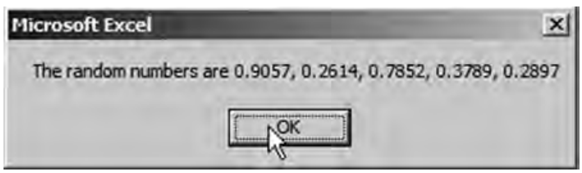
Monte Carlo methods—simulations of option pricing by tracing out many paths of the stock price—are at best a "second best" method of pricing. But in cases where no analytical formulas are available, Monte Carlo is easy to program in VBA and easy to see in Excel. In this chapter we have illustrated Monte Carlo methods for plain-vanilla options, Asian options, and barrier options. Other variations of path-dependent options and their Monte Carlo solutions are considered in the exercises.

Exercises

1. Create a VBA subroutine [call it **Exercise1( )**] which generates a random number and prints it on the screen in a message box that looks like this:

Note Use the VBA keyword **Rnd**.

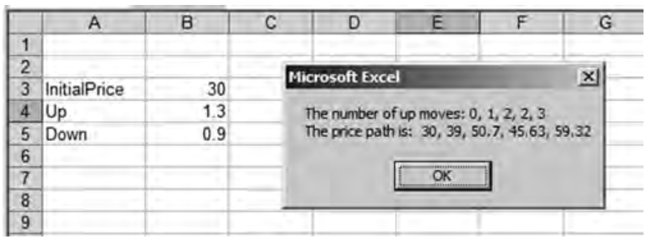
2. Create a VBA subroutine [call it **Exercise2( )**] which generates five random numbers and prints them out on the screen in a message box like the following:



Note Use **FormatNumber**(*Expression*,*NumDigitsAfterDecimal*) to print out only four digits.

3. Create a VBA subroutine [call it **Exercise3( )**] which generates five random digits of either 1 or 0 and prints them out on the screen in a message box like the following:

4. Suppose a stock price follows a binomial distribution. We want to create a *random price path* for the stock in a VBA macro. Here's the input with some sample output. Write a VBA for an appropriate subroutine.



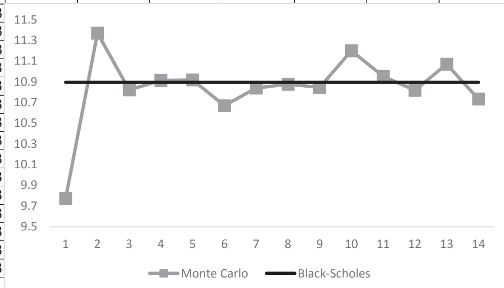
Notes

- In the VBA message box, use **Chr(13)** to start a new line.
- Use range names in the spreadsheet to transfer values from the spreadsheet to the VBA routine.

5. Repeat the last exercise. This time compute the average price of a path:

6. Use the function **VanillaCall** defined in the chapter to create a **Data Table** in which you can see the relation between the number of runs incorporated in the function and the Black-Scholes value of a call. Your result should look something like the following:

	A	B	C	D	E	F	G	H	I	J
1	**MONTE CARLO PRICING OF PLAIN-VANILLA CALLS**									
2	S_0 , current stock price	50								
3	X, exercise price	44								
4	r, interest rate	10%								
5	T, time	0.8								
6	μ, mean stock return	13%								
7	σ, sigma--standard deviation of stock return	30%								
8										
9	n, divisions of unit time	150	<-- Try playing around with this							
10	Runs	111	<-- This is what's altered in the data table							
11										
12	VanillaCall	13.0632	<-- =vanillacall(B2,B3,B6,B7,B4,B5,B9,B10)							
13										
14	BS call	10.8948	<-- =BSCall(B2,B3,B5,B4,B7)							
15										
16										
17	**Data table: the effect of runs on the MC Vanilla call value**									
18		Monte Carlo	Black-Scholes							
19	**Runs**	13.0632	10.8948							
20	100	9.7749	10.8948							
21	500	11.3694	10.8948							
22	1,000	10.8217	10.8948							
23	1,500	10.9121	10.8948							
24	2,000	10.9170	10.8948							
25	2,500	10.6694	10.8948							
26	3,000	10.8389	10.8948							
27	3,500	10.8755	10.8948							
28	4,000	10.8451	10.8948							
29	4,500	11.1993	10.8948							
30	5,000	10.9505	10.8948							
31	5,500	10.8181	10.8948							
32	6,000	11.0687	10.8948							
33	6,500	10.7333	10.8948							
34										



VI EXCEL TECHNIQUES

The five chapters of this section cover technical topics related to Excel. Chapter 31 discusses **Data Tables**, Excel's amazing sensitivity analysis tool. An important addition in this edition of *Financial Modeling* is a discussion of data tables on blank cells. This is a tool we use extensively in our discussions of simulations and Monte Carlo methods in Section V.

Chapter 32 discusses matrices, without going into much theory. In *Financial Modeling* we use matrices primarily in our discussion of portfolio optimization (Section II). Chapter 33 is a compendium of most of the Excel functions used in the book, and Chapter 34 is a discussion of array functions; these are functions that take rectangular arrays of cells as arguments and that are entered by pressing [Ctrl] + [Shift] + [Enter].

Finally, Chapter 35 covers a grab-bag of Excel hints that we couldn't place anywhere else in this book! The last section of this chapter shows how to put custom-made procedures into Excel's personal notebook. We use this particularly to automate the somewhat cumbersome Copy/Paste-as-Picture feature of Excel.

31 Data Tables

31.1 Overview

Data table commands are powerful commands that make it possible to do complex sensitivity analyses. Excel offers the opportunity to build a table in which only one variable is changed, or one in which two variables are changed. Excel data tables are array functions, and thus change dynamically when related spreadsheet cells are changed.

In this chapter you will learn how to build both one-dimensional and two-dimensional Excel data tables.

31.2 An Example

Consider a project which has an initial cost of $1,150, and seven subsequent cash flows. The cash flows in years 1–7 grow at rate g, so that the cash flow in year t is $CF_t = CF_{t-1}*(1 + g)$. Given a discount rate r, the net present value (NPV) of the project is

$$NPV = -1,150 + \frac{CF_1}{(1+r)^1} + \frac{CF_1(1+g)}{(1+r)^2} + \frac{CF_1(1+g)^2}{(1+r)^3} + \ldots + + \frac{CF_1(1+g)^6}{(1+r)^7}$$

The internal rate of return (IRR), i, is the rate at which the NPV equals zero:

$$0 = 1,150 + \frac{CF_1}{(1+i)^1} + \frac{CF_1(1+g)}{(1+i)^2} + \frac{CF_1(1+g)^2}{(1+i)^3} + \ldots + + \frac{CF_1(1+g)^6}{(1+i)^7}$$

These calculations are easily done in Excel. In the following example the initial cash flow is 234, the growth rate g = 10%, and the discount rate r = 15%:

	A	B	C	D	E	F	G	H	I
1	CF_1	234							
2	Growth rate	10%							
3	Discount rate	15%							
4									
5	Year	0	1	2	3	4	5	6	7
6	Cash flow	-1150.00	234.00	257.40	283.14	311.45	342.60	376.86	414.55
7									
8	NPV	101.46	<-- =+B6+NPV(B3,C6:I6)						
9	IRR	17.60%	<-- =IRR(B6:I6,0)						

Note the cell addresses for the growth rate, the discount rate, the NPV, and the IRR. They will be needed below.

31.3 Setting Up a One-Dimensional Data Table

We refer to a data table that does sensitivity analysis by varying one parameter as a *one-dimensional data table*. This section discusses such tables, and section 31.4 discusses two-dimensional data tables.

Suppose we want to know how the NPV and IRR are affected by a change in the growth rate. The command **Data Table** allows us to do this simply. The first step is to set up the table's structure. In the example below, we put the formulas for the NPV and IRR on the top row and we put the variable we wish to vary (in this case the growth rate) in the first column. At this point the table looks like this:

	F	G	H	I	J
10					
11		=B8			=B9
12					
13			NPV	IRR	
14			101.46	17.6%	
15		0			
16	Growth	5%			
17	rate	10%			
18		15%			
19					

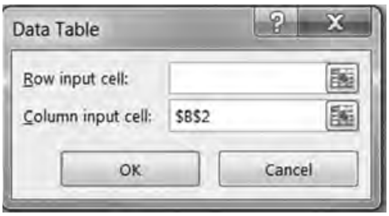
The actual table (as opposed to the labels for the columns and the rows) is outlined in the dark border. The numbers directly under the labels "NPV" and "IRR" refer to the corresponding formulas in the previous picture. Thus, if the cell B8 contains the calculation for the NPV, then the cell under the letters "NPV" contains the formula "=B8." Similarly, if the cell B9 contains the original calculation for the IRR, then the cell under "IRR" in the table contains the formula "=B9."

> We like to think of a data table spreadsheet as having two parts:
>
> • A basic example.
>
> • A table that does a sensitivity analysis on the basic example. In our example, the first row of the table contains references to calculations done in our basic example. While there are other ways to do data tables, this structure is both typical and easy to understand.

Now do the following:

• Highlight the table area (outlined in the dark border).

• Activate the command **Data|What-if Analysis|Data Table**. You will get a dialog box which asks you to indicate a **Row input cell** and/or a **Column input cell**.



In this case, the variable we wish to change is in the left-hand column of our table, so we leave the **Row input cell** blank and indicate cell B2 (this cell contains the growth rate in our basic example.) in the **Column input cell** box. Here's the result:

	F	G	H	I	J
10					
11		=B8			=B9
12					
13			NPV	IRR	
14			101.46	17.6%	
15		0	-176.46	9.71%	
16	Growth	5%	-47.82	13.67%	
17	rate	10%	101.46	17.60%	
18		15%	274.35	21.50%	
19					

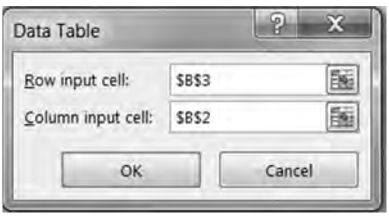
31.4 Building a Two-Dimensional Data Table

We can also use the **Data Table** command to vary *one* formula while changing *two* parameters. Suppose, for example, that we want to calculate the net present value (NPV) of the cash flows for different growth rates and different discount rates. We create a new table which looks like this:

	F	G	H	I	J
21	=B8		Discount rate ↓		
22		101.46	7%	10%	12%
23	Growth	0			
24	rate -->	5%			
25		10%			
26		15%			

The upper left-hand corner of the table contains the formula "=B8" as a reference to the basic example.

We now use the **Data Table** command again. This time we fill in both the **Row input cell** (indicating cell B3, the site of the discount rate in our basic example) and the **Column input cell** (indicating cell B2).

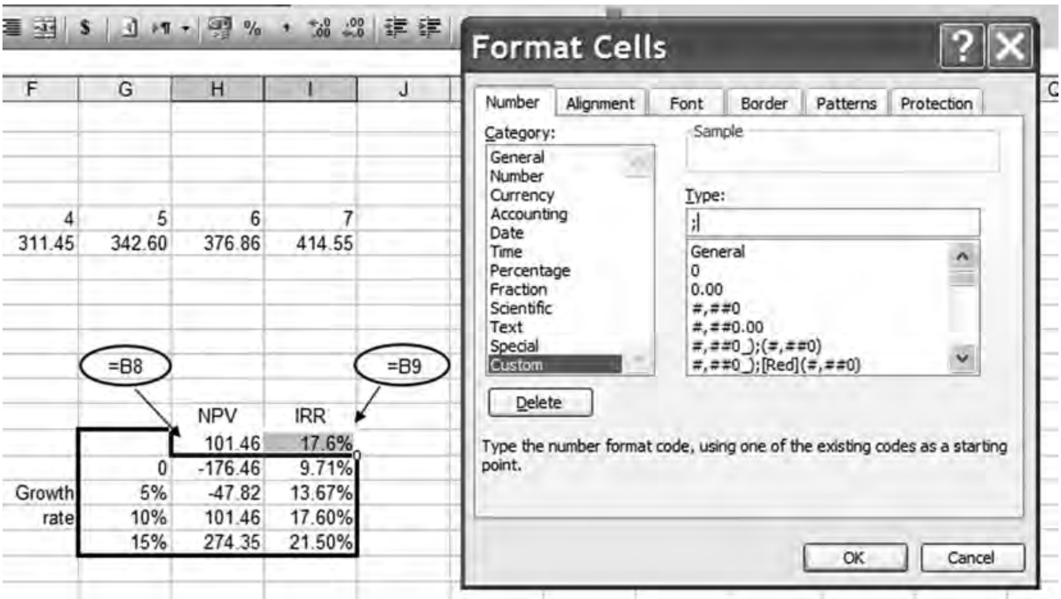


Here's the result:

	F	G	H	I	J
21	=B8		Discount rate ↓		
22		101.46	7%	10%	12%
23	Growth	0	111.09	-10.79	-82.08
24	rate -->	5%	297.62	150.74	65.13
25		10%	515.79	339.09	236.44
26		15%	770.34	558.25	435.41

31.5 An Aesthetic Note: Hiding the Formula Cells

Data tables tend to look a bit strange, because the formula being calculated shows up in the data table (in our examples: in the top row of the first data table, and in the left-hand top corner of the second data table). You can make your tables look nicer by *hiding* the formula cells. To do this, mark the offending cells and use the **Format Cells** command (or press the right mouse button and go to the **Number|Custom**). In the dialog box go to the box marked **Type** and insert a semicolon into the box. Here's the way this looks for the previous example:



The cell contents will now be hidden. This gives the following result.

	F	G	H	I	J
10					
11		=B8			=B9
12					
13			NPV	IRR	
14					
15		0	-176.46	9.71%	
16	Growth	5%	-47.82	13.67%	
17	rate	10%	101.46	17.60%	
18		15%	274.35	21.50%	
19					

31.6 Excel Data Tables Are Arrays

This means that Excel data tables are dynamically linked to your initial example. When you change a parameter in the original example, the corresponding column or row of the data table changes. For example, if we change the initial cash flow from 234 to 300, here's what will happen in the data table pictured above:

	A	B	C	D	E	F	G	H	I	J
1	CF_1	300								
2	Growth rate	10%								
3	Discount rate	15%								
4										
5	Year	0	1	2	3	4	5	6	7	
6	Cash flow	-1150.00	300.00	330.00	363.00	399.30	439.23	483.15	531.47	
7										
8	NPV	454.43	<-- =+B6+NPV(B3,C6:I6)							
9	IRR	26.01%	<-- =IRR(B6:I6,0)							
10										
11							=B8			=B9
12										
13								NPV	IRR	
14										
15							0	98.13	17.80%	
16						Growth	5%	263.06	21.92%	
17						rate	10%	454.43	26.01%	
18							15%	676.09	30.07%	
19										

31.7 Data Tables on Blank Cells (Advanced)

An exciting use of data tables is to run multiple iterations of random simulations. Here's an example: Suppose you use **Rand( )** to create 10 random numbers between 0 and 1. This Excel random number generator is discussed in detail in Chapter 24. The simulation we are describing might look like this:

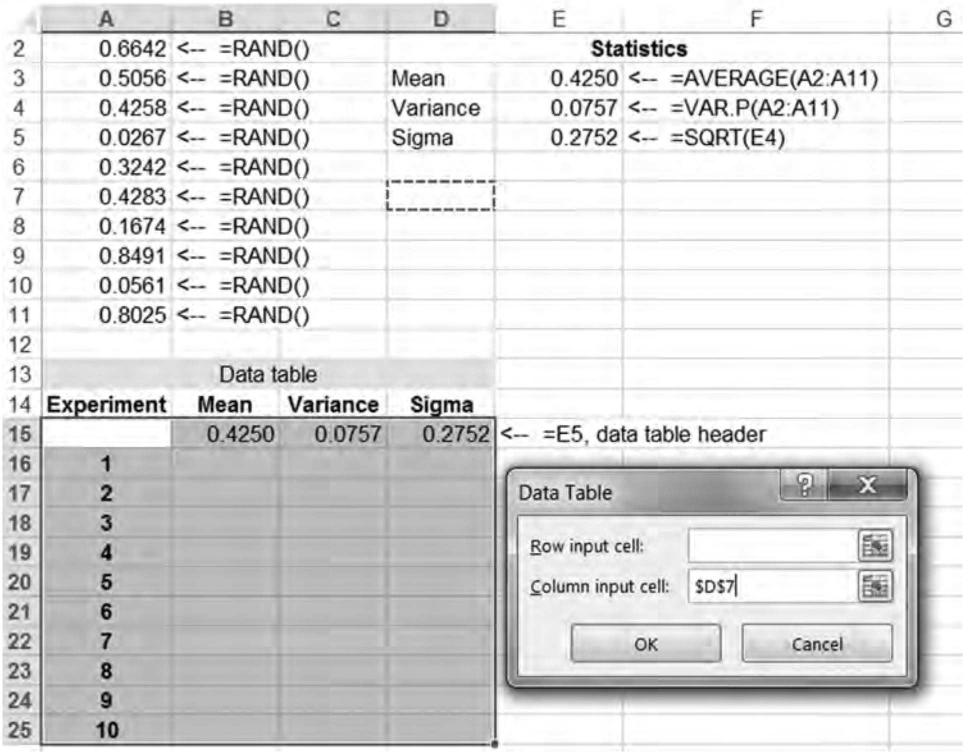
	A	B	C	D	E	F
1	**10 RANDOM NUMBERS**					
2	0.9428	<-- =RAND()		**Statistics**		
3	0.0698	<-- =RAND()		Mean	0.5679	<-- =AVERAGE(A2:A11)
4	0.6437	<-- =RAND()		Variance	0.0878	<-- =VAR.P(A2:A11)
5	0.2236	<-- =RAND()		Sigma	0.2963	<-- =SQRT(E4)
6	0.8855	<-- =RAND()				
7	0.3245	<-- =RAND()				
8	0.4922	<-- =RAND()				
9	0.9802	<-- =RAND()				
10	0.6795	<-- =RAND()				
11	0.4377	<-- =RAND()				

If you want to repeat this experiment 10 times, your spreadsheet might look like this:

	A	B	C	D	E	F	G	H	I	J	K
1	**10 RANDOM NUMBERS**										
2	0.1304	<-- =RAND()			**Statistics**						
3	0.0249	<-- =RAND()		Mean	0.5337	<-- =AVERAGE(A2:A11)					
4	0.5569	<-- =RAND()		Variance	0.1521	<-- =VAR.P(A2:A11)					
5	0.8470	<-- =RAND()		Sigma	0.3900	<-- =SQRT(E4)					
6	0.9996	<-- =RAND()									
7	0.9951	<-- =RAND()									
8	0.9524	<-- =RAND()									
9	0.0621	<-- =RAND()									
10	0.6378	<-- =RAND()									
11	0.1309	<-- =RAND()									
12											
13		10 experiments: each cell contains **Rand()**									
14		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
15		0.6073	0.9267	0.3659	0.9473	0.5307	0.7728	0.6772	0.8663	0.1242	0.9285
16		0.4076	0.3475	0.4079	0.0923	0.1159	0.3125	0.9913	0.9525	0.1252	0.5522
17		0.5411	0.4921	0.0958	0.7892	0.5212	0.4310	0.9040	0.0733	0.9615	0.9775
18		0.2448	0.7231	0.3462	0.9175	0.3743	0.2937	0.1035	0.3312	0.8416	0.5376
19		0.2678	0.7630	0.3316	0.0017	0.4115	0.6189	0.0608	0.3203	0.0959	0.4950
20		0.2402	0.7870	0.3109	0.3760	0.3872	0.1488	0.6636	0.3832	0.3897	0.6506
21		0.1288	0.2508	0.2282	0.6406	0.8679	0.9740	0.1927	0.0848	0.6565	0.4739
22		0.6039	0.9266	0.4174	0.5396	0.5827	0.3528	0.9261	0.2797	0.3899	0.8880
23		0.0390	0.2560	0.3029	0.5665	0.0725	0.4498	0.2902	0.5482	0.7215	0.6274
24		0.2482	0.4922	0.5431	0.9500	0.7878	0.5518	0.5362	0.0747	0.5163	0.8535
25											
26	**Statistics for the 10 experiments**										
27	Mean	0.3329	0.5965	0.3350	0.5821	0.4652	0.4906	0.5345	0.3914	0.4822	0.6984
28	Variance	0.0353	0.0617	0.0127	0.1049	0.0581	0.0545	0.1122	0.0882	0.0870	0.0337
29	Sigma	0.1880	0.2485	0.1126	0.3238	0.2410	0.2335	0.3350	0.2969	0.2949	0.1836

Using Data Table with Blank Cell Reference

There's a more efficient way of running this experiment, illustrated below. Note that the cell referenced in the **Data Table** dialog box is empty.



The result looks something like this ("something" because by the nature of this experiment, the results are random, so that each push of **F9** will produce a different set of numbers):

	A	B	C	D	E	F
1	**BETTER WAY OF RUNNING THE EXPERIMENT**					
2	0.5948	<-- =RAND()		**Statistics**		
3	0.0850	<-- =RAND()		Mean	0.5053	<-- =AVERAGE(A2:A11)
4	0.9396	<-- =RAND()		Variance	0.0717	<-- =VAR.P(A2:A11)
5	0.4272	<-- =RAND()		Sigma	0.2677	<-- =SQRT(E4)
6	0.1722	<-- =RAND()				
7	0.4833	<-- =RAND()				
8	0.6271	<-- =RAND()				
9	0.8981	<-- =RAND()				
10	0.2721	<-- =RAND()				
11	0.5537	<-- =RAND()				
12						
13	Data table					
14	**Experiment**	**Mean**	**Variance**	**Sigma**		
15		0.5053	0.0717	0.2677	<-- =E5, data table header	
16	**1**	0.5527	0.1225	0.3500		
17	**2**	0.4129	0.0624	0.2498		
18	**3**	0.5433	0.1072	0.3275		
19	**4**	0.4453	0.0862	0.2937		
20	**5**	0.3300	0.0420	0.2051		
21	**6**	0.5339	0.0719	0.2681		
22	**7**	0.6534	0.0601	0.2451		
23	**8**	0.2358	0.0157	0.1253		
24	**9**	0.4701	0.0884	0.2973		
25	**10**	0.6402	0.0469	0.2165		

A Slightly More Realistic Example (Also More Advanced)

Why use this technique? Go back to the pension problems discussed in section 27.6. Suppose you retire at age 75 with €1,000,000 in savings. You want to invest this 60% in the risky asset that has a mean annual return $\mu = 11\%$ and a standard deviation of return $\sigma = 30\%$. You plan to withdraw €100,000 at the end of this and each of the next 9 years.

The amount you'll have left at the end of 10 years depends on the random annual return on the risky asset. As explained in Chapter 26, this can be modeled by using **Exp(mu + sigma*Norm.S.Inv(Rand( )))**. We embed this function and the assumptions in the spreadsheet below. The amount left is cell B24. In the example shown below, this number is positive, but it can—depending on the parameters—also be negative.[1]

1. This negative number is, of course, problematic! Maybe you have other resources, maybe you should change your spending policy during the 10 years, or maybe you should die sooner (Heaven forfend!). We ignore all these issues.

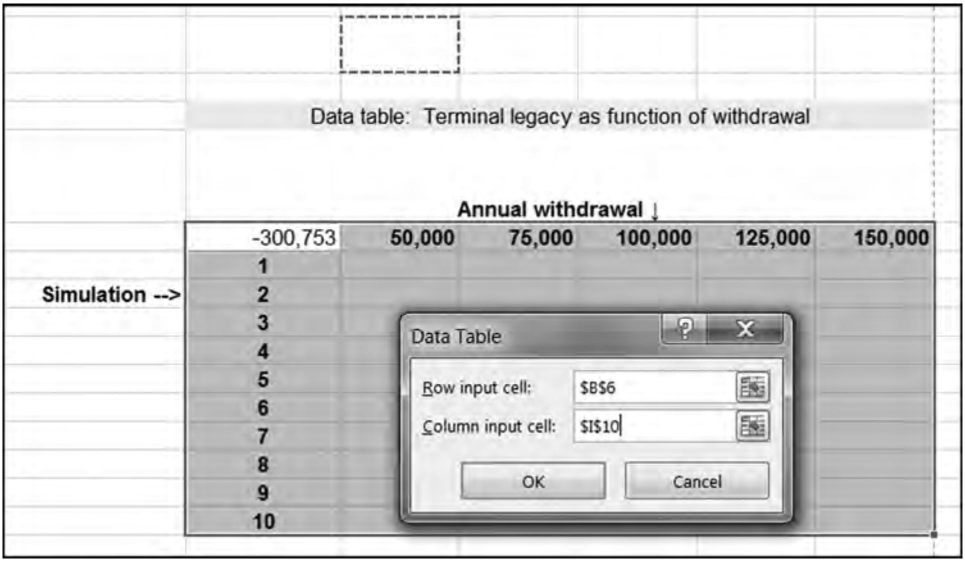
	A	B	C	D	E	F
1	**PENSION PROBLEM**					
2	Current savings	1,000,000				
3	Invested in					
4	Risky asset	60%				
5	Risk free asset	40%	<-- =1-B4			
6	Annual withdrawal	100,000				
7						
8	Return paramets					
9	Risk free rate	4%		=B14*$B$4*EXP(mu+sigma*NORM.S.INV(RAND())+$B$5*EXP(riskfree))		
10	Risky asset mean return μ	11%				
11	Risky asset sigma σ	30%				
12						
13	**Age**	**Savings at beginning of year**	**Savings at end of year**	**Withdrawal at end of year**	**Net after withdrawal**	
14	75	1,000,000	1,437,841	100,000	1,337,841	<-- =C14-D14
15	76	1,337,841	1,097,621	100,000	997,621	
16	77	997,621	713,551	100,000	613,551	
17	78	613,551	722,186	100,000	622,186	
18	79	622,186	543,920	100,000	443,920	
19	80	443,920	483,394	100,000	383,394	
20	81	383,394	651,125	100,000	551,125	
21	82	551,125	380,550	100,000	280,550	
22	83	280,550	339,858	100,000	239,858	
23	84	239,858	252,465	100,000	152,465	
24	85	152,465				

Now we use our new “data table on blank cell” technique to run 20 simulations to see what the final (cell B24) amount might be:

	A	B	C	D	E	F	G	H	I	J
1	**PENSION PROBLEM**									
2	Current savings	1,000,000								
3	Invested in									
4	Risky asset	60%								
5	Risk free asset	40%	<-- =1-B4							
6	Annual withdrawal	100,000								
7										
8	Return paramets									
9	Risk free rate	4%		=B14*$B$4*EXP(mu+sigma*NORM.S.INV(RAND())+$B$5*EXP(riskfree))						
10	Risky asset mean return μ	11%								
11	Risky asset sigma σ	30%								
12								Data table		
13	**Age**	**Savings at beginning of year**	**Savings at end of year**	**Withdrawal at end of year**	**Net after withdrawal**			**Simulation**	**Ending amount**	
14	75	1,000,000	1,045,933	100,000	945,933	<-- =C14-D14			-356,997	<-- =B24, data table header
15	76	945,933	742,643	100,000	642,643			**1**	-301,825	
16	77	642,643	477,596	100,000	377,596			**2**	2,830,285	
17	78	377,596	342,151	100,000	242,151			**3**	-585,010	
18	79	242,151	208,010	100,000	108,010			**4**	-117,106	
19	80	108,010	96,006	100,000	-3,994			**5**	-507,590	
20	81	-3,994	-4,144	100,000	-104,144			**6**	-135,525	
21	82	-104,144	-100,014	100,000	-200,014			**7**	-64,976	
22	83	-200,014	-241,630	100,000	-341,630			**8**	1,265,134	
23	84	-341,630	-256,997	100,000	-356,997			**9**	-345,502	
24	85	-356,997						**10**	-141,072	

Even Better

We can do a simulation on the annual amount withdrawn:

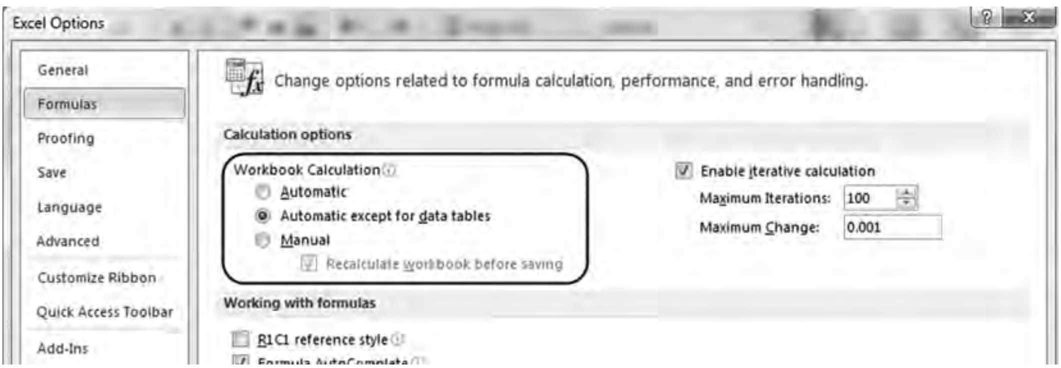


Here are some results, showing 10 simulations for five different annual withdrawal amounts. For fun and for a sanity check, we've also run some statistics in rows 26 and 27.

	G	H	I	J	K	L	M
12		Data table: Terminal legacy as funciton of withdrawal					
13		**Annual withdrawal ↓**					
14		-585,609	**50,000**	**75,000**	**100,000**	**125,000**	**150,000**
15		**1**	494,149	546,991	561,325	1,187,944	-1,709,883
16	**Simulation -->**	**2**	294,802	746,571	-419,406	-322,478	-634,332
17		**3**	167,369	773,073	-41,438	-425,747	638,336
18		**4**	5,224,365	-54,924	-608,211	-291,104	814,324
19		**5**	292,638	947,669	131,293	1,263,636	1,338,209
20		**6**	142,418	53,797	2,087,825	-122,044	-656,701
21		**7**	3,670,562	-95,539	-24,299	-551,433	-74,548
22		**8**	2,043,177	150,463	94,271	-362,274	1,483,009
23		**9**	246,264	1,285,907	105,789	-280,146	-837,601
24		**10**	608,727	166,481	-340,866	-1,718,429	426,265
25							
26		Average	1,318,447	452,049	154,628	-162,207	78,708
27		Sigma	1,778,101	474,664	755,472	856,126	1,035,210

31.8 Data Tables Can Stop Your Computer

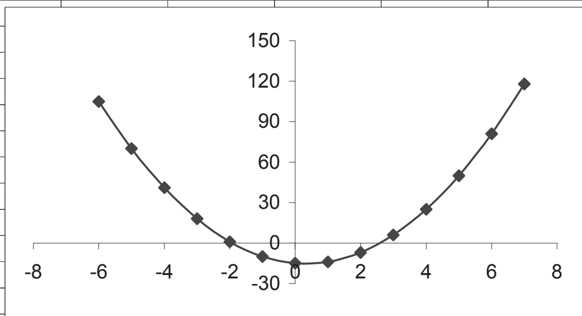
Data tables are amazing, but they can be an incredibly wasteful use of computer resources! A few juicy data tables can slow your spreadsheet to a crawl. One way to get around this is to set the recalculation to manual:



Exercises

1. a. Use **Data|Table** to graph the function $f(x) = 3x^2 - 2x - 15$, as illustrated below:

	A	B	C	D	E	F	G	H	I
1	USING DATA TABLE TO GRAPH A FUNCTION								
2	x	3							
3	f(x)	6	<-- =3*B2^2-2*B2-15						
4									
5									
6		x	6	<-- =B3, data table header					
7		-6	105						
8		-5	70						
9		-4	41						
10		-3	18						
11		-2	1						
12		-1	-10						
13		0	-15						
14		1	-14						
15		2	-7						
16		3	6						
17		4	25						
18		5	50						
19		6	81						
20		7	118						



b. Use **Solver** or **Goal|Seek** to find two values of x for which $f(x) = 0$.

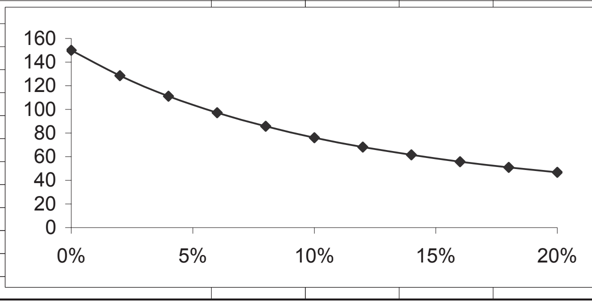
2. The Excel function **PV(rate, number_periods, payment)** calculates the present value of a constant payment. For example in the spreadsheet example below,

$$PV(15\%, 15, -10) = \sum_{t=1}^{15} \frac{10}{(1.15)^t} = 58.47$$

(Note that we have put the payment as a negative number, since otherwise Excel returns a negative value! This little irritation is discussed in Chapters 1 and 34.)

Use **Data Table** to graph the present value as a function of the discount rate, as illustrated below:

	A	B	C	D	E	F	G
1	**DATA TABLE AND PV**						
2	Rate	15%					
3	Number of periods	15					
4	Payment	-10	To get a positive PV, we let the payment be negative (see Chapters 1 & 34)				
5	Present value	$58.47	<-- =PV(B2,B3,B4)				
6							
7							
8	**Rate**	$58.47	<-- =B5, data table header				
9	0%	150.00					
10	2%	128.49					
11	4%	111.18					
12	6%	97.12					
13	8%	85.59					
14	10%	76.06					
15	12%	68.11					
16	14%	61.42					
17	16%	55.75					
18	18%	50.92					
19	20%	46.75					
20							
21							



3. The spreadsheet fragment below shows a net present value and internal rate of return calculation for a project:

	A	B	C	D	E	F	G	H
1	**NPV, DISCOUNT AND GROWTH RATES**							
2	Growth rate	10%						
3	Discount rate	15%						
4	Cost	500						
5	Year 1 cash flow	100						
6								
7	Year	0	1	2	3	4	5	
8	Cash flow	-500.00	100.00	110.00	121.00	133.10	146.41	
9								
10	NPV	(101.42)	<-- =NPV(B3,C8:G8)+B8					
11	IRR	6.60%	<-- =IRR(B8:G8)			Cell B15 contains the data table function =B10		
12								
13								
14			**Growth**					
15		($101.42)	0%	3%	6%	9%	12%	
16		0%	0.00	30.91	63.71	98.47	135.28	
17	**Discount rate**	3%	-42.03	-14.56	14.55	45.38	78.01	
18		6%	-78.76	-54.26	-28.30	-0.84	28.21	
19		9%	-111.03	-89.08	-65.85	-41.28	-15.33	
20		12%	-139.52	-119.78	-98.91	-76.86	-53.57	
21		15%	-164.78	-146.97	-128.15	-108.28	-87.32	
22		18%	-187.28	-171.15	-154.13	-136.16	-117.23	
23		21%	-207.40	-192.75	-177.30	-161.01	-143.84	
24		24%	-225.46	-212.11	-198.04	-183.22	-167.62	

Use **Data Table** to do a sensitivity analysis on the NPV of the project, varying the discount rates from 0%, 3%, 6%, …, 21% and varying the growth rates from 0%, 3%, …, 12%.

4. Using **Data Table** graph the function sin(x * y) for x = 0, 0.2, 0.4, … 1.8, 2 and y = 0, 0.2, 0.4, … 1.8, 2. Use the "Surface" graph option to make a three-dimensional graph of the function.

5. Boris and Tareq are tossing coins. For each toss, if the coin falls on heads, Tareq wins $1. If the coin falls on tails, Tareq pays Boris $1.

   - Simulate 10 rounds of this game, showing Tareq's cumulative winnings.
   - Use **Data Table** on a blank cell to simulate 25 games of 10 rounds each, showing Tareq's cumulative winnings.

6. Maria and Shavit are tossing coins. Their game works as follows:

   - On the first toss, if the coin falls on heads, Shavit pays Maria $1 (and vice versa).
   - On each successive toss:
     - If the coin falls on heads and Maria is ahead, Shavit pays her the square of her previous winnings.
     - If the coin falls on heads and Shavit is ahead, this cancels all of Maria's debt to Shavit.

   Simulate Maria's winnings after 10 tosses.

32 Matrices

32.1 Overview

The portfolio optimization chapters of *Financial Modeling* (Chapters 8–13) make extensive use of matrices to find efficient portfolios. This chapter contains enough information about matrices to make it possible for you to do the calculations required for portfolio mathematics.

A matrix is a rectangular array of numbers. All of the following are matrices:

	A	B	C	D	E	F	G	H	I
1	**MATRICES IN EXCEL**								
2	**Matrix A (a row vector)**				**Matrix B (square 3 x 3 matrix)**				**Matrix C (column vector)**
3	2	3	4		13	-8	-3		13
4					-8	10	-1		-8
5					-3	-1	11		-3
6									
7	**Matrix D (a 4 x 3 matrix)**								
8	13	-8	-3						
9	-8	10	-1						
10	-3	-1	11						
11	0	13	3						

A matrix with only one row is also called a *row vector*; a matrix with only one column is also called a *column vector*. A matrix with an equal number of rows and columns is called a *square matrix*.

A single letter is often used to denote a matrix or a vector. In this case we often write, for example, $B = [b_{ij}]$, where b_{ij} stands for the entry in row i and column j of the matrix. For a vector we might write $A = [a_i]$ or $C = [c_i]$. Thus for the examples given above:

$$a_3 = 4, \quad b_{22} = 10, \quad c_1 = 13, \quad d_{41} = 0$$

The matrix B above is *symmetric*, meaning that $b_{ij} = b_{ji}$. (The variance-covariance matrices used in the portfolio discussion of Chapters 8–13 are symmetric.)

32.2 Matrix Operations

In this section we briefly review the basic operations on a matrix: multiplying a matrix by a scalar, adding matrices, transposition of matrices, and matrix multiplication.

Multiplication by a Scalar

Multiplying a matrix by a scalar multiplies every entry in the matrix by the scalar. Thus, for example:

	A	B	C	D	E
1	MULTIPLYING A MATRIX BY A SCALAR				
2	Scalar	6			
3					
4	Matrix B	13	-8	-3	
5		-8	10	-1	
6		-3	-1	11	
7					
8	Scalar * Matrix B				
9		78	-48	-18	<-- =D4*$B$2
10		-48	60	-6	
11		-18	-6	66	

.

Matrix Addition

Matrices may be added together provided they have the same number of rows and columns. Adding two vectors or matrices is accomplished by adding their corresponding entries. Thus if $A = [a_{ij}]$ and $B = [b_{ij}]$, $A + B = [a_{ij} + b_{ij}]$:

	A	B	C	D	E	F	G	H	I
1	ADDITION OF MATRICES								
2	Matrix A			Matrix B			Sum of A + B		
3	1	3		2	3		3	6	<-- =B3+E3
4	3	0		23	5		26	5	
5	6	-9		8	6		14	-3	
6	5	11		-15	1		-10	12	
7	7	12		4	-1		11	11	

Matrix Transposition

Transposition is an operation by which the rows of a matrix are turned into columns and vice versa. Thus for the matrix *E*:

	A	B	C	D	E	F	G	H	I
1	**TRANSPOSITION OF MATRICES**								
2	**Matrix E**					**Transpose of E: E^T**			
3	1	2	3	4		1	0	16	<-- {=TRANSPOSE(A3:D5)}
4	0	3	77	-9		2	3	7	
5	16	7	7	2		3	77	7	
6						4	-9	2	
7									
8	Cells F3:H6 are generated with the array function Transpose(A3:D5). This function is inserted by marking off the target area, typing the formula, and then finishing by pressing [Ctrl]+[Shift]+[Enter] . See Chapter 34 for more details.								

The illustration above uses the array function **Transpose**. More details on the use of array functions are given in Chapter 34.

Multiplication of Matrices

You can multiply matrix *A* by matrix *B* to get product *AB* . However, you can only do this if the number of columns in *A* equals the number of rows in *B*. The resulting product *AB* is a matrix with the number of rows as *A* and the number of columns as in *B*.

Confused? A couple of examples will help. Suppose that *X* is a row vector and that *Y* is a column vector, both with *n* coordinates:

$$X = [x_1 \quad \cdots \quad x_n], Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Then the *product of X and the product of X and Y* are defined by

$$XY = [x_1 \quad \cdots \quad x_n] \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i.$$

Now suppose that *A* and *B* are two matrices, and that *A* has *n* columns and *p* rows and *B* has *n* rows and *m* columns:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \\ a_{p1} & a_{p2} & \cdots & a_{pn} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ b_{21} & \cdots & b_{2m} \\ \vdots & & \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}.$$

Then the product of A and B, written AB, is defined by the matrix:

$$AB = \begin{bmatrix} \sum_{h=1}^{n} a_{1h}b_{h1} & \sum_{h=1}^{n} a_{1h}b_{h2} & \dots & \sum_{h=1}^{n} a_{1h}b_{hm} \\ \vdots & & & \\ \vdots & & & \\ \sum_{h=1}^{n} a_{ph}b_{h1} & \dots & \dots & \sum_{h=1}^{n} a_{ph}b_{hm} \end{bmatrix}, \textit{with } ij^{th} \textit{ element} = \sum_{h=1}^{n} a_{ih}b_{hj}$$

Note that the *ij*th coordinate of AB is the product of the *i*th row of A times the *j*th column of B. For example, if

$$A = \begin{bmatrix} 2 & -6 \\ -9 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 6 & 9 & -12 \\ -5 & 2 & 4 \end{bmatrix} \quad \textit{then} \quad AB = \begin{bmatrix} 42 & 6 & -48 \\ -69 & -75 & 120 \end{bmatrix}.$$

The order of matrix multiplication is critical. Multiplication of matrices is not commutative; that is, $AB \neq BA$. As the above example shows, the fact that it is possible to multiply A times B does not always imply that the multiplication BA is even defined.

To multiply matrices in Excel, we use the array function **MMult**:

	A	B	C	D	E	F
1	**MULTIPLYING MATRICES**					
2	**Matrix A**			**Matrix B**		
3	2	-7		6	9	-12
4	0	3		-5	2	4
5						
6	**Product AB**					
7	47	4	-52	<-- {=MMULT(A3:B4,D3:F4)}		
8	-15	6	12			

To multiply two matrices together, the number of columns in the first matrix must equal the number of rows in the second. Thus we can multiply A times

B, but we cannot multiply *B* times *A*. If you try this in Excel, the function **MMult** will give you an error message:

	A	B	C	D	E	F
1	**MATRIX MULTIPLICATION: Number of columns of first matrix must equal number of rows of second matrix**					
2	**Matrix A**			**Matrix B**		
3	2	-7		6	9	-12
4	0	3		-5	2	4
5						
6	**Product BA**					
7	#VALUE!	#VALUE!	#VALUE!	<-- {=MMULT(D3:F4,A3:B4)}		
8	#VALUE!	#VALUE!	#VALUE!			

32.3 Matrix Inverses

A square matrix *I* is called the *identity matrix* if all its off-diagonal entries are 0 and all its diagonal entries are 1. Thus

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

It is easy to confirm that multiplying any matrix *A* by the identity matrix of the proper dimension leaves that *A* unchanged. Thus, if I_n is an $n \times n$ identity matrix and *A* is an $n \times m$ matrix, $IA = A$. Similarly, if I_m is an $m \times m$ identity matrix, $AI = A$.

Now suppose we are given a *square* matrix *A* of dimension *n*. The $n \times n$ matrix A^{-1} is called the *inverse* of *A* if $A^{-1}A = AA^{-1} = I$. The computation of an inverse matrix can be a lot of work; fortunately, however Excel has the array function **MInverse** which does the calculations for us. Here's an example:

	A	B	C	D	E	F	G	H	I	J
1	**MATRIX INVERSE** Use array function Minverse to compute the inverse of a square matrix									
2	Matrix A					Inverse of A				
3	1	-9	16	1		-0.0217	1.8913	0.5362	-1.1449	<-- {=MINVERSE(A3:D6)}
4	3	3	2	3		0.0000	-1.0000	-0.1667	0.6667	
5	2	4	0	-2		0.0652	-0.6739	-0.1087	0.4348	
6	5	7	3	4		-0.0217	-0.1087	-0.2971	0.1884	
7										
8	**Verifying the inverse**									
9	We multiply A*Inverse A: cells below contain array function {=MMULT(A3:D6,F3:I6)}									
10	1	1.07E-15	-2.22045E-16	-9.4369E-16						
11	0	1	-1.11022E-16	2.22045E-16						
12	6.94E-18	8.33E-17	1	5.55112E-16						
13	1.39E-17	1.17E-15	-4.44089E-16	1						

As illustrated above, you can use **MMult** to verify that the product of the above matrix and its inverse indeed give the identity matrix. Expressions like 1.07E-15 mean $1.07*10^{-15}$, and are thus essentially zero; you can use **Format|Cells|Number** to specify the number of decimal places and get rid of these ugly expressions:

	A	B	C	D
9	We multiply A*Inverse A: cells below contain array function {=MMULT(A3:D6,F3:I6)}			
10	1.0000	0.0000	0.0000	0.0000
11	0.0000	1.0000	0.0000	0.0000
12	0.0000	0.0000	1.0000	0.0000
13	0.0000	0.0000	0.0000	1.0000

A square matrix that has an inverse is called a *nonsingular matrix*. The conditions for a matrix to be nonsingular are the following: Consider a square matrix A of dimension n. It can be shown that $A = [a_{ij}]$ is nonsingular if and only if the only solution to the n equations

$$\sum_i a_{ij} x_i = 0, \quad j = 1, \ldots, n$$

is $x_i = 0$, $i = 1, \ldots, n$. Matrix inversion is a tricky business. If there exists a vector X whose components are almost zero and which solves the above system, then the matrix is *ill-conditioned*, and it may be very difficult to find an accurate inverse.

32.4 Solving Systems of Simultaneous Linear Equations

A system of *n* linear equations in *m* unknown is written as:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & y_1 \\ a_{21}x_1 & + & a_{22}x_1 & + & \cdots & + & a_{2n}x_1 & = & y_2 \\ \vdots & & & & & & & & \\ a_{n1}x_1 & + & a_{n2}x_1 & + & \cdots & + & a_{nn}x_1 & = & y_n \end{array}$$

Writing the matrix of coefficients as $A = [a_{ij}]$, the column vector of unknowns as $X = [x_j]$, and the column vector of constants as $Y = [y_j]$, we may write the above system in matrix notation as $AX = Y$.

Not every system of linear equations has a solution, and not every solution of such a system is unique. The system $AX = Y$ *always* has a unique solution, however, if the matrix A is square and nonsingular. In this case the solution is found by pre-multiplying both sides of the equation $AX = Y$ by the inverse of A:

since $AX = Y \Rightarrow A^{-1}AX = A^{-1}Y \Rightarrow X = A^{-1}Y$

Here is an example. Suppose we want to solve the 3 x 3 system of equations:

$$\begin{array}{ccccccc} 3x_1 & + & 4x_2 & + & 66x_3 & = & 16 \\ & & -33x_2 & + & x_3 & = & 77 \\ 42x_1 & + & 3x_2 & + & 2x_3 & = & 12 \end{array}$$

We set this up and solve it in Excel as follows:

	A	B	C	D	E	F	G	H
1	SOLVING SIMULTANEOUS EQUATIONS							
2	Matrix A of coefficients				Column vector y		Solution A^{-1} Y	
3	3	4	66		16		0.4343	
4	0	-33	1		77		-2.3223	<-- {=MMULT(MINVERSE(A3:C5),E3:E5)}
5	42	3	2		12		0.3634	
6								
7	Checking that the solution works							
8		16						
9		77	<-- {=MMULT(A3:C5,G3:G5)}					
10		12						

In cells B8:B10 we check that the solution indeed solves the system by multiplying the matrix A times the column vector G3:G5.

32.5 Some Homemade Matrix Functions

In the chapters on portfolio problems we use a number of matrix functions that are not included in Excel. The functions in this section need to be embedded into the spreadsheet where they are used—see the Technical Note at the beginning of this book or Chapter 36 of the VBA section.

Variance-Covariance Matrix

When dealing with stock returns, we often need to compute the variance-covariance matrix. The following function does the trick:

```
'My thanks to Amir Kirsch and Beni Czaczkes
Function VarCovar(rng As Range) As Variant
   Dim i As Integer
   Dim j As Integer
   Dim numcols As Integer
   numcols = rng.Columns.Count
   numrows = rng.Rows.Count
   Dim matrix() As Double
   ReDim matrix(numcols - 1, numcols - 1)
   For i = 1 To numcols
      For j = 1 To numcols
      matrix(i - 1, j - 1) = _
      Application.WorksheetFunction. _
      Covar(rng.Columns(i), rng.Columns(j)) _
      * numrows / (numrows - 1)
      Next j
   Next i
   VarCovar = matrix
End Function
```

Here is an application. The spreadsheet below shows five years of monthly returns for 10 stocks:

	A	B	C	D	E	F	G	H	I	J	K
1	**FIVE YEARS OF RETURNS FOR 10 STOCKS**										
2		**McDonalds**	**US Steel**	**Arcelor-Mittal**	**Microsoft**	**Apple**	**Kellogg**	**General Electric**	**Bank of America**	**Pfizer**	**Exxon**
3	**Date**	**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
4	1-Mar-07	3.07%	11.18%	3.91%	-1.06%	9.36%	3.13%	1.27%	0.38%	1.17%	5.13%
5	2-Apr-07	6.93%	2.37%	1.00%	7.18%	7.15%	2.84%	4.17%	-0.25%	4.64%	5.08%
6	1-May-07	4.59%	11.02%	12.16%	2.80%	19.42%	2.55%	1.91%	0.73%	4.89%	5.09%
7	1-Jun-07	0.41%	-3.98%	3.93%	-4.06%	0.70%	-4.14%	2.60%	-3.66%	-7.23%	0.85%
8	2-Jul-07	-5.85%	-10.11%	-2.23%	-1.66%	7.66%	0.04%	1.24%	-3.06%	-8.39%	1.49%
9	1-Aug-07	2.83%	-3.72%	8.65%	-0.53%	4.97%	6.42%	0.28%	6.66%	6.70%	1.09%
56	1-Jul-11	2.53%	-14.09%	-10.97%	5.24%	15.12%	0.82%	-5.16%	-12.05%	-6.79%	-1.97%
57	1-Aug-11	5.12%	-28.23%	-33.99%	-2.32%	-1.46%	-1.86%	-9.36%	-17.20%	-0.27%	-6.85%
58	1-Sep-11	-2.92%	-31.32%	-32.29%	-6.66%	-0.92%	-2.10%	-5.99%	-28.81%	-7.13%	-1.89%
59	3-Oct-11	5.58%	14.13%	26.47%	6.77%	5.97%	1.90%	9.37%	10.99%	8.57%	7.24%
60	1-Nov-11	3.58%	7.58%	-8.23%	-3.27%	-5.74%	-8.89%	-4.89%	-22.61%	5.11%	3.56%
61	1-Dec-11	4.91%	-3.13%	-3.78%	1.47%	5.79%	2.83%	12.79%	2.18%	7.56%	5.24%
62	3-Jan-12	-1.28%	13.17%	12.05%	12.88%	11.97%	-2.10%	4.37%	24.87%	-1.13%	-1.22%
63	1-Feb-12	0.94%	2.85%	12.71%	4.11%	7.73%	1.38%	2.22%	13.74%	-0.19%	1.90%

To compute the variance-covariance matrix:

	A	B	C	D	E	F	G	H	I	J	K
1	**VARIANCE-COVARIANCE MATRIX FOR 10 STOCKS**										
2		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
3	**MCD**	0.0020	0.0037	0.0028	0.0015	0.0017	0.0007	0.0020	0.0031	0.0015	0.0011
4	**X**	0.0037	0.0380	0.0284	0.0076	0.0111	0.0031	0.0127	0.0176	0.0043	0.0043
5	**MT**	0.0028	0.0284	0.0267	0.0065	0.0097	0.0031	0.0102	0.0133	0.0038	0.0039
6	**MSFT**	0.0015	0.0076	0.0065	0.0063	0.0049	0.0010	0.0046	0.0079	0.0018	0.0014
7	**AAPL**	0.0017	0.0111	0.0097	0.0049	0.0126	0.0016	0.0049	0.0049	0.0007	0.0020
8	**K**	0.0007	0.0031	0.0031	0.0010	0.0016	0.0026	0.0028	0.0046	0.0011	0.0003
9	**GE**	0.0020	0.0127	0.0102	0.0046	0.0049	0.0028	0.0122	0.0163	0.0041	0.0022
10	**BAC**	0.0031	0.0176	0.0133	0.0079	0.0049	0.0046	0.0163	0.0393	0.0080	0.0017
11	**PFE**	0.0015	0.0043	0.0038	0.0018	0.0007	0.0011	0.0041	0.0080	0.0041	0.0011
12	**XOM**	0.0011	0.0043	0.0039	0.0014	0.0020	0.0003	0.0022	0.0017	0.0011	0.0026
13											
14			Formula: {=VARCOVAR('Return data'!B4:K63)}								

Correlation Matrix

An obvious variation on the previous function computes the correlation matrix of the returns:

```
Function CorrMatrix(rng As Range) As Variant
     Dim i As Integer
     Dim j As Integer
     Dim numCols As Integer
     numCols = rng.Columns.Count
     numRows = rng.Rows.Count
     Dim matrix() As Double
     ReDim matrix(numCols - 1, numCols - 1)
     For i = 1 To numCols
          For j = 1 To numCols
          matrix(i - 1, j - 1) = _
          Application.WorksheetFunction.
          Correl(rng.Columns(i), rng.Columns(j))
          Next j
     Next i
     CorrMatrix = matrix
End Function
```

Applied to the above example:

	A	B	C	D	E	F	G	H	I	J	K
1	**CORRELATION MATRIX FOR 10 STOCKS**										
2		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
3	**MCD**	1.0000	0.4199	0.3859	0.4238	0.3379	0.2920	0.4064	0.3506	0.5411	0.4741
4	**X**	0.4199	1.0000	0.8898	0.4898	0.5062	0.3078	0.5904	0.4556	0.3491	0.4361
5	**MT**	0.3859	0.8898	1.0000	0.5044	0.5277	0.3692	0.5659	0.4103	0.3602	0.4620
6	**MSFT**	0.4238	0.4898	0.5044	1.0000	0.5497	0.2416	0.5312	0.5050	0.3542	0.3581
7	**AAPL**	0.3379	0.5062	0.5277	0.5497	1.0000	0.2827	0.3964	0.2205	0.0945	0.3425
8	**K**	0.2920	0.3078	0.3692	0.2416	0.2827	1.0000	0.4846	0.4559	0.3487	0.1234
9	**GE**	0.4064	0.5904	0.5659	0.5312	0.3964	0.4846	1.0000	0.7461	0.5842	0.3926
10	**BAC**	0.3506	0.4556	0.4103	0.5050	0.2205	0.4559	0.7461	1.0000	0.6328	0.1723
11	**PFE**	0.5411	0.3491	0.3602	0.3542	0.0945	0.3487	0.5842	0.6328	1.0000	0.3435
12	**XOM**	0.4741	0.4361	0.4620	0.3581	0.3425	0.1234	0.3926	0.1723	0.3435	1.0000
13											
14			Formula: {=CorrMatrix('Return data'!B4:K63)}								

Unit Rows and Unit Columns

Portfolio computations sometimes use unit rows or columns. For example global minimum variance portfolio (GMVP) of Chapter 10 has the following formula:

$$GMVP\ as\ row = \frac{unitrow(N) * S}{unitrow(N) * S * unitcolumn(N)}$$

$$GMVP\ as\ column = \frac{S * unitcolumn(N)}{unitrow(N) * S * unitcolumn(N)}$$

where S is the variance – covariance matrix:

$$S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \cdots & \sigma_{2N} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \cdots & \sigma_{3N} \\ \vdots & & & & \\ \sigma_{N1} & \sigma_{N2} & \sigma_{N3} & \cdots & \sigma_{NN} \end{bmatrix}$$

The denominator of the above formulas is simply the sum of the numerator, and we can thus rewrite the formulas as:

$$GMVP\ as\ row = \frac{unitrow(N) * S}{Sum(unitrow(N) * S)}$$

$$GMVP\ as\ column = \frac{S * unitcolumn(N)}{Sum(S * unitcolumn(N))}$$

The unit column or row can be computed in the following two functions:

```
'With thanks to Priyush Singh and Ayal Itzkovitz
Function UnitrowVector(numcols As Integer) _
As Variant
     Dim i As Integer
     Dim vector() As Integer
     ReDim vector(0, numcols - 1)
     For i = 1 To numcols
          vector(0, i - 1) = 1
     Next i
     UnitrowVector = vector
End Function

Function UnitColVector(numrows As Integer) As Variant
     Dim i As Integer
     Dim vector() As Integer
     ReDim vector(numrows - 1, 1)
     For i = 1 To numrows
          vector(i - 1, 0) = 1
     Next i
     UnitColVector = vector
End Function
```

Here's an example of these functions in a spreadsheet:

	A	B	C	D	E
1	**ROW AND COLUMN UNIT VECTORS**				
2	1	<-- {=UNITCOLVECTOR(5)}			
3	1				
4	1				
5	1				
6	1				
7					
8	1	1	1	1	<-- {=UNITROWVECTOR(4)}

A more interesting use of these functions is to use them directly in a formula. In the following example we compute the GMVP:

	A	B	C	D	E	F	G	H	I	J	K
1	**COMPUTING THE GLOBAL MINIMUM VARIANCE PORTFOLIO (GMVP)**										
2		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
3	**MCD**	0.0020	0.0037	0.0028	0.0015	0.0017	0.0007	0.0020	0.0031	0.0015	0.0011
4	**X**	0.0037	0.0380	0.0284	0.0076	0.0111	0.0031	0.0127	0.0176	0.0043	0.0043
5	**MT**	0.0028	0.0284	0.0267	0.0065	0.0097	0.0031	0.0102	0.0133	0.0038	0.0039
6	**MSFT**	0.0015	0.0076	0.0065	0.0063	0.0049	0.0010	0.0046	0.0079	0.0018	0.0014
7	**AAPL**	0.0017	0.0111	0.0097	0.0049	0.0126	0.0016	0.0049	0.0049	0.0007	0.0020
8	**K**	0.0007	0.0031	0.0031	0.0010	0.0016	0.0026	0.0028	0.0046	0.0011	0.0003
9	**GE**	0.0020	0.0127	0.0102	0.0046	0.0049	0.0028	0.0122	0.0163	0.0041	0.0022
10	**BAC**	0.0031	0.0176	0.0133	0.0079	0.0049	0.0046	0.0163	0.0393	0.0080	0.0017
11	**PFE**	0.0015	0.0043	0.0038	0.0018	0.0007	0.0011	0.0041	0.0080	0.0041	0.0011
12	**XOM**	0.0011	0.0043	0.0039	0.0014	0.0020	0.0003	0.0022	0.0017	0.0011	0.0026
13											
14	**GMVP**										
15	**MCD**	0.0326	<-- {=MMULT(B3:K12,UNITCOLVECTOR(10))/SUM(MMULT(B3:K12,UNITCOLVECTOR(10)))}								
16	**X**	0.2117									
17	**MT**	0.1754									
18	**MSFT**	0.0705									
19	**AAPL**	0.0873									
20	**K**	0.0340									
21	**GE**	0.1166									
22	**BAC**	0.1891									
23	**PFE**	0.0493									
24	**XOM**	0.0335									
25											
26		**MCD**	**X**	**MT**	**MSFT**	**AAPL**	**K**	**GE**	**BAC**	**PFE**	**XOM**
27		0.0326	0.2117	0.1754	0.0705	0.0873	0.0340	0.1166	0.1891	0.0493	0.0335
28		Formula: {=MMULT(UNITROWVECTOR(10),B3:K12)/SUM(MMULT(UNITROWVECTOR(10),B3:K12))}									

Exercises

1. Use Excel to perform the following matrix operations:

a. $$\begin{bmatrix} 2 & 12 & 6 \\ 4 & 8 & 7 \\ 1 & 0 & -9 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 2 \\ 8 & 0 & -23 \\ 1 & 7 & 3 \end{bmatrix}$$

b. $$\begin{bmatrix} 2 & -9 \\ 5 & 0 \\ 6 & -6 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ 2 & 3 & 2 \end{bmatrix}$$

c. $$\begin{bmatrix} 2 & 0 & 6 \\ 4 & 8 & 7 \\ 1 & 0 & -9 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 8 & 0 & -2 \\ 1 & 7 & 3 \end{bmatrix}$$

2. Find the inverses of the following matrices:

a. $\begin{bmatrix} 1 & 2 & 8 & 9 \\ 2 & 5 & 3 & 0 \\ 4 & 4 & 2 & 7 \\ 5 & -2 & 1 & 6 \end{bmatrix}$

b. $\begin{bmatrix} 3 & 2 & 1 \\ 6 & -1 & 3 \\ 7 & 4 & 3 \end{bmatrix}$

c. $\begin{bmatrix} 20 & 2 & 3 & -3 \\ 2 & 10 & 2 & -2 \\ 3 & 2 & 40 & 9 \\ -3 & -2 & 9 & 33 \end{bmatrix}$

3. Transpose the following matrices using the Excel array function **Transpose**:

a. $A = \begin{bmatrix} 3 & 2 & 1 \\ -15 & 4 & 1 \\ 6 & -9 & 1 \end{bmatrix}$

b. $B = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ -2 & 7 & -9 & 0 & 0 \\ 3 & -3 & 11 & 12 & 1 \end{bmatrix}$

4. Solve the following system of equations by using matrices:

$$\begin{aligned} 3x + 4y - 6z - 9w &= 15 \\ 2x - y + w &= 2 \\ y + z + w &= 3 \\ x + y - z &= 1 \end{aligned}$$

5. Solve the equations *AX*=*Y*, where:

$$A = \begin{bmatrix} 13 & -8 & -3 \\ -8 & 10 & -1 \\ -3 & -1 & 11 \end{bmatrix}, Y = \begin{bmatrix} 20 \\ -5 \\ 0 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

6. An ill-conditioned matrix is a matrix that "almost doesn't have" an inverse. A set of examples of such matrices are Hilbert matrices. An n-dimensional Hilbert matrix looks like:

$$H_n = \begin{bmatrix} 1 & 1/2 & \dots & 1/n \\ 1/2 & 1/3 & \dots & 1/(n+1) \\ \vdots & & & \\ 1/n & 1/(n+1) & & 1/(2n-1) \end{bmatrix}$$

a. Calculate the inverses of H_2, H_3, and H_8.

b. Consider the following system of equations:

$$H_n \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 + 1/2 + \cdots + 1/n \\ 1/2 + 1/3 + \cdots 1/(n+1) \\ \vdots \\ 1/n + 1/(n+1) + \cdots + 1/(2n-1) \end{bmatrix}$$

Find the answers to these problems by inspection.

c. Now solve $H_n{*}X = Y$ for $n = 2, 8, 14$. How do you explain the differences?

33 Excel Functions

33.1 Overview

Excel contains several hundred functions. This chapter surveys only those functions used in this book. The functions discussed are the following:

- Financial functions: **NPV**, **IRR**, **PV**, **PMT**, **XIRR**, and **XNPV**
- Date functions: **Now**, **Today**, **Date**, **Weekday**, **Month**, **Datedif**
- Statistical functions: **Average**, **Var**, **Varp**, **Stdev**, **Stdevp**, **Correl**, **Covar**
- Regression functions: **Slope**, **Intercept**, **Rsq**, **Linest**
- Conditional functions: **If**, **VLookup**, **HLookup**
- **Large**, **Rank**, **Percentile**, **Percentrank**
- **Count**, **CountA**, **CountIf**
- **Offset**

A separate chapter, Chapter 34, is devoted to the important topic of array functions.

33.2 Financial Functions

Excel has a large number of financial functions. The main functions used in this book are explored in this section.

NPV

The Excel definition of **NPV** differs somewhat from the standard finance definition. In the finance literature, the net present value of a sequence of cash flows $C_0, C_1, C_2, \ldots, C_n$ at a discount rate r refers to the expression

$$\sum_{t=0}^{n} \frac{C_t}{(1+r)^t} \text{ or } C_0 + \sum_{t=1}^{n} \frac{C_t}{(1+r)^t}$$

In many cases C_0 represents the cost of the asset purchased and is therefore negative.

The Excel definition of **NPV** always assumes that the first cash flow occurs after one period. The user who wants the standard finance expression must therefore calculate $\mathbf{NPV(r,\{C_1, \ldots , C_n\}) + C_0}$. Here is an example:

	A	B	C	D
1	**EXCEL'S NPV FUNCTION**			
2	Discount rate	10%		
3				
4	**Year**	**Cash flow**	**Present value**	
5	0	-100.00	-100.00	<-- =B5/(1+$B$2)^A5
6	1	35.00	31.82	
7	2	33.00	27.27	
8	3	34.00	25.54	
9	4	25.00	17.08	
10	5	16.00	9.93	
11				
12	Present value of future cash flows	111.65	<-- =SUM(C6:C10)	
13		111.65	<-- =NPV(B2,B6:B10)	
14	Net present value	11.65	<-- =B5+NPV(B2,B6:B10)	

The **NPV** function has a potential bug: It differentiates between blank cells and cells containing zeros. This can cause some confusion, as can be seen in the example below. The present value of the cash flows in B5:B7 is 65.75, which corresponds to $\frac{100}{1.15^3}$. But in the otherwise similar example of the cash flows in B11:B13, Excel's NPV function regards the first cash flow as being 100, and returns the answer $\frac{100}{1.15} = 86.96$. So—in using **NPV** you have to be explicit in putting in zeros for zero cash flows.

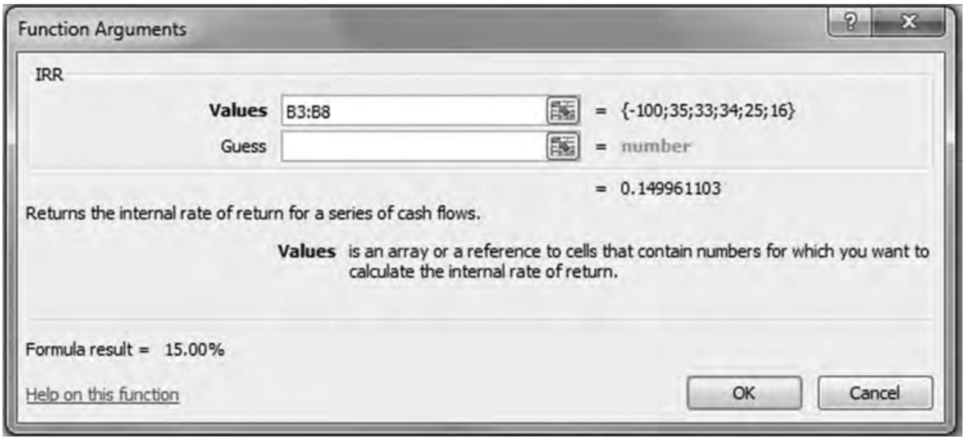
	A	B	C
1	**NPV IGNORES BLANK CELLS!**		
2	Discount rate	15%	
3			
4	**Year**	**Cash flow**	
5	1	0.00	
6	2	0.00	
7	3	100.00	
8	Present value	65.75	<-- =NPV(B2,B5:B7)
9			
10	**Year**	**Cash flow**	
11	1		
12	2		
13	3	100.00	
14	Present value	86.96	<-- =NPV(B2,B11:B13)

IRR

The internal rate of return (IRR) of a sequence of cash flows C_0, C_1, C_2, …, C_n is an interest rate r such that the net present value of the cash flows is zero:

$$\sum_{t=0}^{n} \frac{C_t}{(1+r)^t} = 0$$

The Excel syntax for the **IRR** function is **IRR(cash flows**, **guess)**. Here **cash flows** represents the whole sequence of cash flows, including the first cash flow C_0, and **guess** is an optional starting point for the algorithm which calculates the IRR.

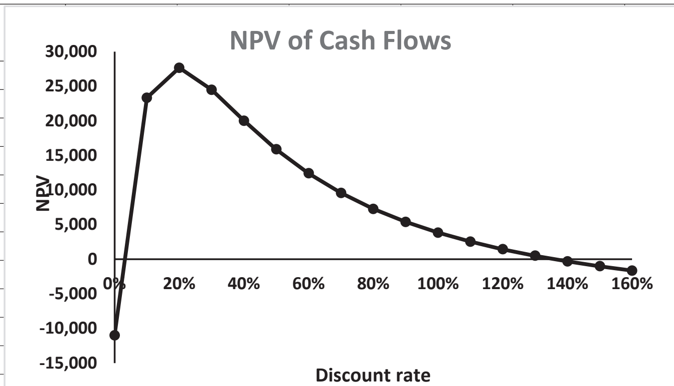


Guess is not required, and when there is only one IRR it is usually irrelevant:

	A	B	C
1	**EXCEL'S IRR FUNCTION**		
2	Year	Cash flow	
3	0	-100	
4	1	35	
5	2	33	
6	3	34	
7	4	25	
8	5	16	
9			
10	IRR	15.00%	<-- =IRR(B3:B8)
11		15.00%	<-- =IRR(B3:B8,5%), IRR Guess = 5%

If there are multiple IRRs, then choice of **guess** can make a difference. Consider, for example, the following cash flows:

	A	B	C	D	E	F	G	H	I
1	**MULTIPLE IRRs**								
2	**Year**	**Cash flows**							
3	0	-11,000							
4	1	15,000							
5	2	15,000							
6	3	15,000							
7	4	15,000							
8	5	15,000							
9	6	15,000							
10	7	15,000							
11	8	15,000							
12	9	15,000							
13	10	-135,000							
14									
15	IRR	1.86%	<-- =IRR(B3:B13,0)						
16		135.99%	<-- =IRR(B3:B13,2)						



The graph (created from a **Data Table** which is not shown) shows that there are two IRRs, since the NPV curve crosses the *x*-axis twice. To find both these IRRs, we have to change the **guess** (though the precise value of **guess** is still not critical). In the example below we have changed both guesses, but still get the same answer:

	A	B	C	D
15	IRR	1.86%	<-- =IRR(B3:B13,0)	
16		135.99%	<-- =IRR(B3:B13,2)	

Note A given set of cash flows typically has more than one IRR if there is more than one change of sign in the cash flows—in the above example, the initial cash flow is negative, and CF_1–CF_9 are positive (this accounts for one change of sign); but then CF_{10} is negative—making a second change of sign. If you suspect that a set of cash flows has more than one IRR, the first thing to do is to use Excel to make a graph of the NPVs, as we did above. The number of times that the NPV graph crosses the *x*-axis identifies the number of IRRs (and also their approximate values).

PV

This function calculates the present value of an annuity (a series of fixed periodic payments). For example:

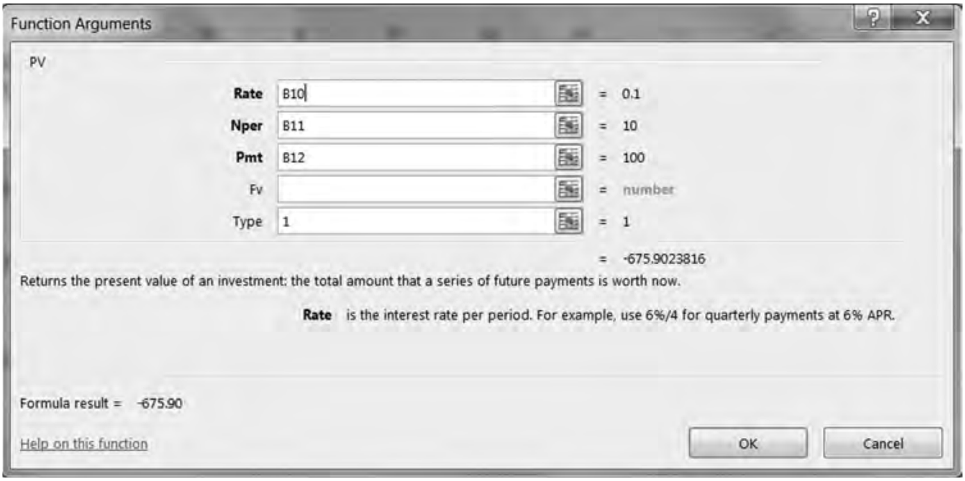
	A	B	C
1	**THE PV FUNCTION**		
2	Payments made at the end of the period		
3	Rate	10%	
4	Number of periods	10.00	
5	Periodic payment	100.00	
6	Present value	-614.46	<-- =PV(B3,B4,B5)

Thus $\$614.46 = \sum_{t=1}^{10} \frac{100}{(1.10)^t}$. Here are two things to note about the **PV** function:

- Writing **PV(B3,B4,B5)** assumes that payments are made at dates 1, 2, … , 10. If the payments are made at dates 0, 1, 2, … , 9, you should write:

	A	B	C
9	Payments made at the beginning of the period		
10	Rate	10%	
11	Number of periods	10.00	
12	Periodic payment	100.00	
13	Present value	-675.90	<-- =PV(B10,B11,B12,,1)

The formula **PV(B10,B11,B12,,1)** can also be generated from the dialog box:



- Irritatingly, the **PV** function (and the **PMT**, **IPMT**, and **PPMT** functions—see below) produces a negative answer when the payment or future value is positive (there is a logic here, but it's not worth explaining). The solution is obvious: Either write **-PV(B3,B4,B5)** or let the payment be negative by writing **PV(B3,B4,-B5)**.

PMT

This function calculates the payment necessary to pay off a loan with equal payments over a fixed number of periods. For example, the first calculation below shows that a loan of $1,000, to be paid off over 10 years at an interest rate of 8% will require equal annual payments of interest and principal of $149.03. The calculation performed is the solution of the following equation:

$$\sum_{t=1}^{n} \frac{X}{(1+r)^t} = \textit{Initial loan principal}$$

	A	B	C
1	**THE PMT FUNCTION**		
2	Payments made at the end of the period		
3	Rate	8%	
4	Number of periods	10.00	
5	Principal	1,000.00	
6	Payment	-149.03	<-- =PMT(B3,B4,B5)
7			
8	Payments made at the beginning of the period		
9	Rate	8%	
10	Number of periods	10.00	
11	Principal	1,000.00	
12	Payment	-137.99	<-- =PMT(B9,B10,B11,,1)

Loan tables can be calculated using the **PMT** function. These tables—explained in detail in Chapter 1—show the split between interest and principal of each payment. In each period, the payment on the loan (calculated with **PMT**) is split:

• We first calculate the interest owing for that period on the principal outstanding at the beginning of the period. In the table below, at the end of year 1, we owe $80 (= 8% * $1,000) of interest on the loan principal outstanding at the beginning of the year.

• The remainder of the payment (for year 1: $69.03) goes to reduce the principal outstanding.

	A	B	C	D	E	F
1	**LOAN TABLE**					
2	Interest	8%				
3	Number of periods	10			=$B$5	
4	Principal	1,000				
5	Annual payment	149.03	<-- =-PMT(B2,B3,B4)			
6						
7				Split payment into		
8	**Year**	**Principal at beginning of year**	**Payment**	**Interest**	**Repayment of principal**	
9	1	1,000.00	149.03	80.00	69.03	<-- =C9-D9
10	2	930.97	149.03	74.48	74.55	
11	3	856.42	149.03	68.51	80.52	
12	4	775.90	149.03	62.07	86.96	
13	5	688.95	149.03	55.12	93.91	
14	6	595.03	149.03	47.60	101.43	
15	7	493.60	149.03	39.49	109.54	
16	8	384.06	149.03	30.73	118.30	
17	9	265.76	149.03	21.26	127.77	
18	10	137.99	149.03	11.04	137.99	
19						
20	=B9-E9				=$B$2*B9	

Note that at the end of the 10 years the repayment of principal is exactly equal to the principal outstanding at the beginning of the year (i.e., the loan has been paid off).

The Functions IPMT and PPMT

As you have seen above, a loan table shows the split of a loan's flat payments (computed with **PMT**) between interest and principal. In the loan table of the previous subsection, we computed this split by first computing the flat payment per period (column C), then taking the interest on the principal at the beginning of the period (column D), and finally subtracting this interest from the period's total payment (column E).

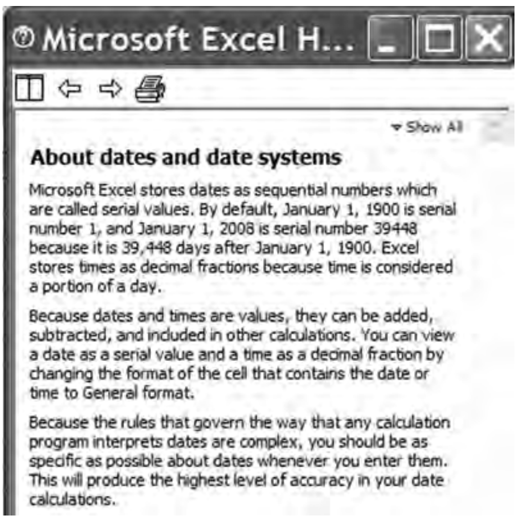
IPMT and **PPMT** perform this calculation without the necessity of relying on the total payment. Here's an example:

	A	B	C	D	E
1	**IPMT AND PPMT**				
2	Interest	8%			
3	Number of periods	10			
4	Principal	1,000			
5					
6	**Year**	**Principal payment at end year**		**Interest payment at end year**	
7	1	69.03	<-- =PPMT($B$2,A7,$B$3,-$B$4)	80.00	<-- =IPMT($B$2,A7,$B$3,-$B$4)
8	2	74.55	<-- =PPMT($B$2,A8,$B$3,-$B$4)	74.48	<-- =IPMT($B$2,A8,$B$3,-$B$4)
9	3	80.52		68.51	
10	4	86.96		62.07	
11	5	93.91		55.12	
12	6	101.43		47.60	
13	7	109.54		39.49	
14	8	118.30		30.73	
15	9	127.77		21.26	
16	10	137.99		11.04	

As you can see, the payments computed are the same as in the loan table of the previous subsection.

33.3 Dates and Date Functions

Read the quote from the Excel help below and you will know almost everything you need to know about entering dates into your spreadsheet.


Microsoft Excel H...

Show All

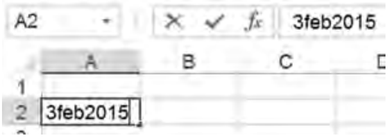
About dates and date systems

Microsoft Excel stores dates as sequential numbers which are called serial values. By default, January 1, 1900 is serial number 1, and January 1, 2008 is serial number 39448 because it is 39,448 days after January 1, 1900. Excel stores times as decimal fractions because time is considered a portion of a day.

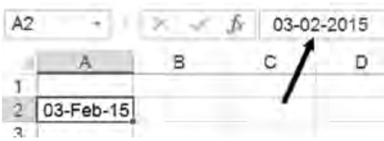
Because dates and times are values, they can be added, subtracted, and included in other calculations. You can view a date as a serial value and a time as a decimal fraction by changing the format of the cell that contains the date or time to General format.

Because the rules that govern the way that any calculation program interprets dates are complex, you should be as specific as possible about dates whenever you enter them. This will produce the highest level of accuracy in your date calculations.

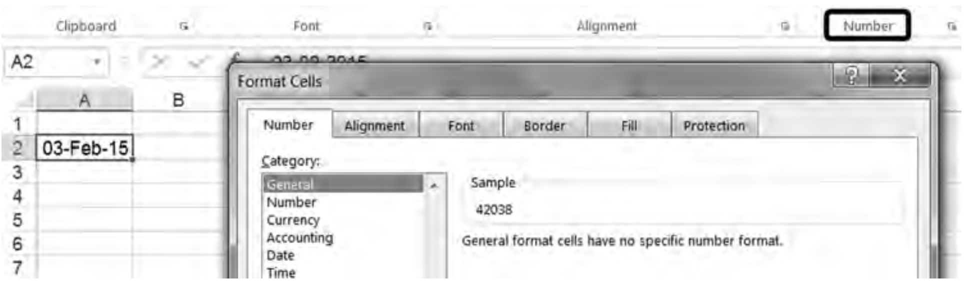
The basic fact you need to know is that Excel translates dates into a number: Here's an example: Suppose you decide to type a date into a cell:



When you hit [Enter], Excel decides that you've entered a date. Here's the way it appears:



Note that in the formula bar (indicated by the arrow above), Excel interprets the date entered as **03-02-2015**.[1] When you reformat the cell as **Format Cells|Number|General**, you see that Excel interprets this date as the number 42038, the number 1 being 1 January 1900.



Spreadsheet dates can be subtracted: In the spreadsheet below we've entered two dates and subtracted them to find the number of days between the dates:

1. The way this appears and is interpreted depends on the Regional Settings entered in the Windows Control Panel.

	A	B	C
1	Date1	15-Aug-40	
2	Date2	28-Sep-52	
3	Days between	4427	<-- =B2-B1

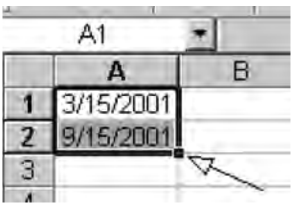
You can also add a number to a date to find another date. What, for example, was the date 165 days after 16 November 1947?

	B	C
5	16-Nov-47	
6	29-Apr-48	<-- =B5+165

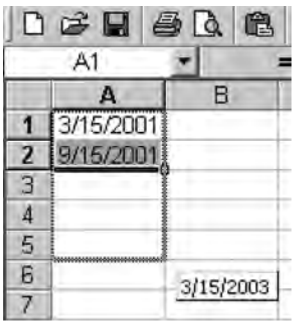
Stretching Out Dates

In the two cells below we've put in two dates and then "stretched" the cells out to add more dates with the same difference between them:

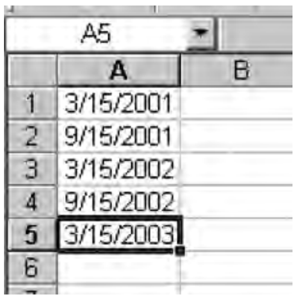
Write in two dates; mark both cells.



Grab the handle (arrow on previous drawing) and pull

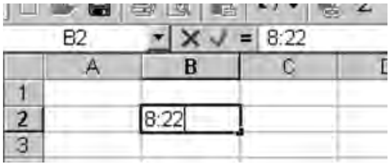


The result: More dates added with same spacing (in this case, 6 months):

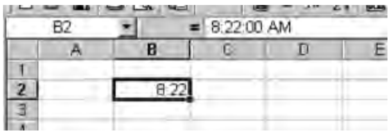


Times in a Spreadsheet

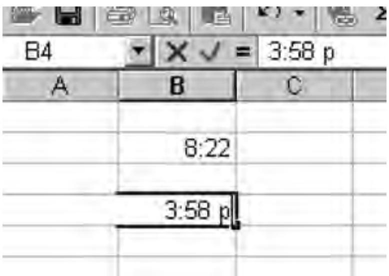
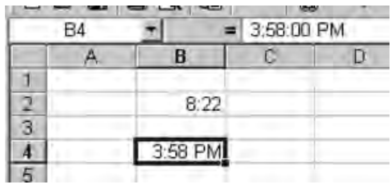
Hours, minutes, etc., can also be typed into a cell. In the cell below, we've typed in 8:22:

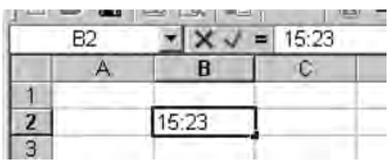
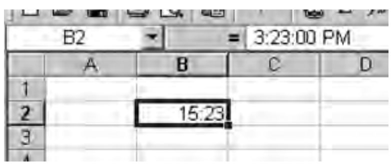


When we hit [Enter], Excel interprets this as 8:22 AM:



Excel recognizes 24-hour times and also recognizes the symbol **a** for AM and **p** for PM:

As entered	When you hit [Enter]
 Note that the **p** is separated from the time by a space. (Of course AM is represented by an **a**.)	

EXCEL RECOGNIZES 24-HOUR CLOCK	
As entered	When you hit [Enter]
	

You can subtract times just like you subtract dates; cell B5 below tells you that 7 hours and 32 minutes have elapsed between the two times (ignore the "AM" in B5):

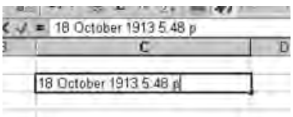
	B	C	D
3	3:48 PM		
4	8:16 AM		
5	7:32 AM	<-- =B3-B4	

When you reformat the cells above with **Format|Cells|Number|General**, you can see that times are represented in Excel as fractions of a day:

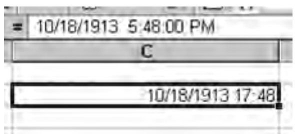
	B	C	D
3	0.658333		
4	0.344444		
5	0.313889	<-- =B3-B4	

If you type in a date and a time and reformat, you can also see this:

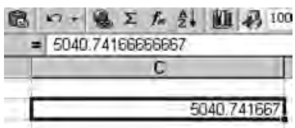
Here's what you typed:



Here's how it appears:



If you reformat this to **General**:



Time and Date Functions in Excel

Excel has a whole set of time and date functions. Here are several functions which we find useful:

- **Now** reads the computer clock and represents the date and the time. **Now** takes no arguments and is written with empty parentheses: **Now( )**.
- **Today** reads the computer's clock and prints the date. This function, like **Now**, is written with empty parentheses: **Today( )**.
- **Date(yyyy,mm,dd)** gives the date entered.
- **Weekday** gives the day of the week
- **Month, Weeknum, Day** give the month, week, day of a date

Here are some of these functions in a spreadsheet:

	A	B	C
1	**Serial representation**	**Date/Time Format**	
2	41620.41885	12-12-13 10:03	<-- =NOW()
3	41620	12-12-13	<-- =TODAY()
4	43924	Apr-3-2020	<-- =DATE(2020,4,3)
5			
6	**Different formatting of Now()**		
7		12-Dec-13	<-- =NOW()
8		12-12-13 10:03	<-- =NOW()
9		10:03 AM	<-- =NOW()
10			
11	**Using Weekday, Month, Weeknum, Day**		
12		5	<-- =WEEKDAY(NOW())
13		5	<-- =WEEKDAY("3apr1947")
14		4	<-- =MONTH(B4)
15		12	<-- =MONTH(NOW())
16		50	<-- =WEEKNUM(NOW())
17		12	<-- =DAY(NOW())

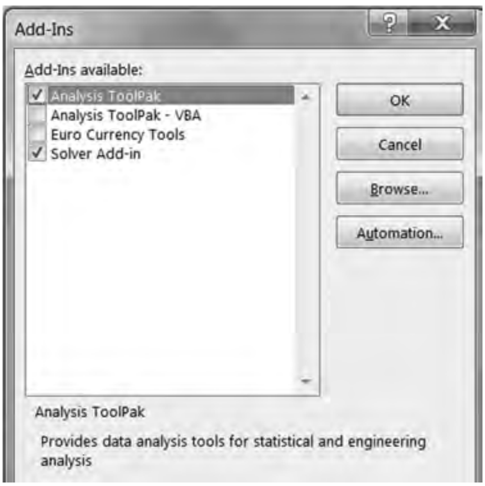
Calculating the Difference Between Two Dates: The Function Datedif

The Excel function **Datedif** computes the difference between two dates in various useful ways:

	A	B	C	D
1	DATEDIF COMPUTES THE DIFFERENCE BETWEEN TWO DATES			
2	Date1	03-Apr-47		
3	Date2	15-Jan-13		
4				
5		65	<-- =DATEDIF(B2,B3,"y")	Number of years between dates
6		789	<-- =DATEDIF(B2,B3,"m")	Number of months between dates
7		24029	<-- =DATEDIF(B2,B3,"d")	Number of days between dates
8		12	<-- =DATEDIF(B2,B3,"md")	Number of days in excess of full number of months
9		9	<-- =DATEDIF(B2,B3,"ym")	Number of months in excess of full number of years
10		287	<-- =DATEDIF(B2,B3,"yd")	Number of days in excess of full number of years

33.4 The Functions XIRR, XNPV

The functions **XIRR** and **XNPV** calculate the internal rate of return and the net present value for a series of cash flows received on specific dates. They are especially useful for calculating IRR and NPV when the dates are unevenly spaced.[2] If you do not have these functions, you will have to go to **File|Options|Add-Ins|Manage**. Choose **Add-Ins** and mark **Analysis ToolPak**:



2. Excel's **IRR** function assumes that the first cash flow occurs today, the next cash flow occurs one period hence, the following cash flow two periods hence, etc. Excel's **NPV** function assumes that the first cash flow occurs one period from now, the next cash flow in two periods, etc. We call this "even spacing of cash flows." When this is not the case, you'll need the **XIRR** and **XNPV** functions.

XIRR

Here's an example: You pay $600 on 16 February 2001 for an asset that repays $100 on 5 April, $100 on 15 July 2001, and then $100 on every 22 September from 2001 until 2009. The dates are not evenly spaced, so that you cannot use **IRR**. With **XIRR** (cell B16 below), you can compute the *annualized IRR* (the effective annual interest rate EAIR, as defined in Chapter 2).

	A	B	C
1	THE EXCEL XIRR FUNCTION		
2	Date	Payment	
3	16-Feb-01	-600	
4	05-Apr-01	100	
5	15-Jul-01	100	
6	22-Sep-01	100	
7	22-Sep-02	100	
8	22-Sep-03	100	
9	22-Sep-04	100	
10	22-Sep-05	100	
11	22-Sep-06	100	
12	22-Sep-07	100	
13	22-Sep-08	100	
14	22-Sep-09	100	
15			
16	XIRR	21.97%	<-- =XIRR(B3:B14,A3:A14)

The **XIRR** function works by discounting each cash flow at the daily rate. In our example the first cash flow of $100 occurs 48 days from now, the second in 149 days, … The **XIRR** transforms 21.97% to a daily rate and uses it to discount the cash flows:

$$-600+\frac{100}{(1.2197)^{48/365}}+\frac{100}{(1.2197)^{149/365}}+\ldots+\frac{100}{(1.2197)^{3140/365}}=0$$

	A	B	C	D	E
1	**HOW DOES XIRR WORK?** **XIRR computes the daily internal rate of return**				
2	**Date**	**Payment**	**Days from initial date**	**Present value**	
3	16-Feb-01	-600		-600.00	<-- =B3
4	05-Apr-01	100	48	97.42	<-- =B4/(1+$B$16)^(C4/365)
5	15-Jul-01	100	149	92.21	<-- =B5/(1+$B$16)^(C5/365)
6	22-Sep-01	100	218	88.81	<-- =B6/(1+$B$16)^(C6/365)
7	22-Sep-02	100	583	72.81	<-- =B7/(1+$B$16)^(C7/365)
8	22-Sep-03	100	948	59.70	<-- =B8/(1+$B$16)^(C8/365)
9	22-Sep-04	100	1,314	48.91	
10	22-Sep-05	100	1,679	40.10	
11	22-Sep-06	100	2,044	32.88	
12	22-Sep-07	100	2,409	26.96	
13	22-Sep-08	100	2,775	22.09	
14	22-Sep-09	100	3,140	18.11	
15					
16	XIRR	21.97%	<-- =XIRR(B3:B14,A3:A14)	0.00	<-- =SUM(D3:D14)
17					
18	Cell C4 contains the				
19	formula =A4-$A$3				

XNPV

The **XNPV** function computes the NPV for unevenly spaced cash flows. In the example below, we use the function to compute the NPV on the same example we used for **XIRR**.

	A	B	C
1	THE EXCEL XNPV FUNCTION		
2	Date	Payment	
3	16-Feb-01	-600	
4	05-Apr-01	100	
5	15-Jul-01	100	
6	22-Sep-01	100	
7	22-Sep-02	100	
8	22-Sep-03	100	
9	22-Sep-04	100	
10	22-Sep-05	100	
11	22-Sep-06	100	
12	22-Sep-07	100	
13	22-Sep-08	100	
14	22-Sep-09	100	
15			
16	Discount rate	15%	
17	XNPV	97.29	<-- =XNPV(B16,B3:B14,A3:A14)

Notice that **XNPV** requires you to indicate all the cash flows (starting with the initial cash flow), as opposed to **NPV**, which starts from the first cash flow.

Bugs in XIRR and XNPV

XIRR and **XNPV** address the problems of discounting when cash flows are time dated, but they have two bugs: **XNPV** does not work with zero discount rates, and **XIRR** does not solve multiple internal rates of return. Below we discuss these problems. We then define two new functions, **NXNPV** and **NXIRR**, that solve these problems. These two functions are included as additional functions in the spreadsheet that accompanies this chapter.[3]

Problem: XNPV Doesn't Work with Zero Discount Rates

XNPV works fine with positive discount rates but fails when discount rates are zero.[4] We illustrate this below.

3. The two additional functions **NXIRR** and **NXNPV** were developed by Benjamin Czaczkes. Read "Adding Getformula to Your Spreadsheet" on the disk that accompanies this book to see how you can copy these functions into your own spreadsheets.

4. Recall that when the discount rate is zero, the net present value is just the sum of all the cash flows.

	A	B	C	D	E	F
1	XNPV DOES NOT WORK WITH ZERO DISCOUNT RATES					
2	Discount rate	3%		Discount rate	0%	
3	XNPV	579.00	<-- =XNPV(B2,B6:B11,A6:A11)	XNPV	#NUM!	<-- =XNPV(E2,E6:E11,D6:D11)
4						
5	**Date**	**Cash flow**		**Date**	**Cash flow**	
6	13-Jan-12	-1,000		13-Jan-12	-1,000	
7	18-Aug-12	115		18-Aug-12	115	
8	20-Jan-13	121		20-Jan-13	121	
9	15-Jul-13	100		15-Jul-13	100	
10	01-Jan-14	333		01-Jan-14	333	
11	16-Jul-14	1,011		16-Jul-14	1,011	

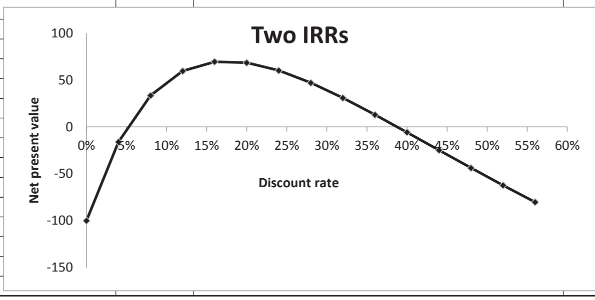
Our new function **NXNPV** solves this problem:

	A	B	C	D	E	F
1	NXNPV SOLVES THE PROBLEM					
2	Discount rate	3%		Discount rate	0%	
3	XNPV	579.00	<-- =nXNPV(B2,B6:B11,A6:A11)	XNPV	680	<-- =nXNPV(E2,E6:E11,D6:D11)
4						
5	**Date**	**Cash flow**		**Date**	**Cash flow**	
6	13-Jan-12	-1,000		13-Jan-12	-1,000	
7	18-Aug-12	115		18-Aug-12	115	
8	20-Jan-13	121		20-Jan-13	121	
9	15-Jul-13	100		15-Jul-13	100	
10	01-Jan-14	333		01-Jan-14	333	
11	16-Jul-14	1,011		16-Jul-14	1,011	

Problem: XIRR Doesn't Work with Two IRRs

The **Guess** switch on **XIRR** doesn't work. As shown below, this means that **XIRR** is unable to compute the IRR for cash flows having multiple rates of return:

	A	B	C	D	E	F	G
1	**PROBLEMS WITH XIRR**						
2	Discount rate	22%					
3	Net present value	65.09	<-- =nXNPV(B2,B9:B15,A9:A15), no Guess				
4	IRR	#NUM!	<-- =XIRR(B9:B15,A9:A15)				
5		#NUM!	<-- =XIRR(B9:B15,A9:A15,35%), Guess = 5%		**Data table: XNPV as function of discount rate**		
6		#NUM!	<-- =XIRR(B9:B15,A9:A15,5%), Guess = 35%		Discount	NPV	
7						65.09	<-- =B3, data table header
8	**Date**	**Cash flow**			0%	-100.00	
9	30-Jun-13	-500			4%	-15.97	
10	14-Feb-14	100			8%	33.62	
11	14-Feb-15	300			12%	59.64	
12	14-Feb-16	400			16%	69.54	
13	14-Feb-17	600			20%	68.52	
14	14-Feb-18	800			24%	60.19	
15	14-Feb-19	-1800			28%	47.09	
16					32%	30.97	
17					36%	13.10	
18					40%	-5.68	
19					44%	-24.75	
20					48%	-43.70	
21					52%	-62.25	
22					56%	-80.23	
23							
24							
25							
26							
27							
28							
29							
30							



From the data table, it is evident that there are two internal rates of return (around 5% and around 40%). But the **XIRR** function does not identify either (see cells B4:B6). The additional function **NXIRR** solves this problem:

	A	B	C
1	**NXIRR SOLVES THE PROBLEM**		
2	IRR	5.06%	<-- =nXIRR(B7:B13,A7:A13)
3		38.80%	<-- =nXIRR(B7:B13,A7:A13,35%), Guess = 35%
4		5.06%	<-- =nXIRR(B7:B13,A7:A13,5%), Guess = 5%
5			
6	**Date**	**Cash flow**	
7	30-Jun-13	-500	
8	14-Feb-14	100	
9	14-Feb-15	300	
10	14-Feb-16	400	
11	14-Feb-17	600	
12	14-Feb-18	800	
13	14-Feb-19	-1800	

33.5 Statistical Functions

Excel contains a number of statistical functions. We illustrate these functions using the following example. Note that with the introduction of Excel 2013, some of these functions have been renamed, though all of the functions still work; thus, for example, the population variance can be computed with the function **Var** (old versions of Excel) or **Var.p** (new versions).

	A	B	C	D	E	F	G	H
1	BASIC STATISTICAL FUNCTIONS							
2	Observation	X	Y					
3	1	35.30	10.98					
4	2	29.70	11.13					
5	3	30.80	12.51					
6	4	58.80	8.40					
7	5	61.40	9.27					
8	6	71.30	8.73					
9	7	74.40	6.36					
10	8	76.70	8.50					
11	9	70.70	7.82					
12	10	57.50	9.14					
13								
14								
15								
16								
17								
18								
19	Average	56.66	9.28	<-- =AVERAGE(C3:C12)		Correlation	-0.9049	<-- =CORREL(B2:B12,C2:C12)
20	Sample variance	334.15	3.23	<-- =VAR(C3:C12)		Covariance	40.4206	<-- =COVAR(B19:B29,C19:C29)
21		334.15	3.23	<-- =VAR.S(C3:C12)			44.4627	<-- =COVARIANCE.S(B19:B29,C19:C29)
22	Population variance	300.73	2.91	<-- =VARP(C3:C12)			40.4206	<-- =COVARIANCE.P(B19:B29,C19:C29)
23		300.73	2.91	<-- =VAR.P(C3:C12)				
24	Sample standard deviation	18.28	1.80	<-- =STDEV(C3:C12)		Regression intercept	14.3285	<-- =INTERCEPT(C3:C12,B3:B12)
25		18.28	1.80	<-- =SQRT(C21)		Regression slope	-0.0890	<-- =SLOPE(C3:C12,B3:B12)
26		18.28	1.80	<-- =STDEV.S(C3:C12)		Regression r-squared	0.8189	<-- =RSQ(C3:C12,B3:B12)
27	Population standard deviation	17.34	1.71	<-- =STDEVP(C3:C12)				
28		17.34	1.71	<-- =SQRT(C22)				
29		17.34	1.71	<-- =STDEV.P(C3:C12)				

The functions **Varp**, **Var.p**, **Stdevp**, and **Stdev.p** calculate the population variance and standard deviation, whereas the functions **Var, Stdev**, **Var.s**, **Stdev.s** compute the sample variance and standard deviation. The difference between these two functions is that **Varp** assumes that your data are the whole population and thus divides by the number of data points, whereas **Var** assumes that the data are samples from the distribution:

$$Varp(x_1, \ldots, x_N) = \frac{1}{N}\sum_{i=1}^{N}(x_i - Average(x_1, \ldots, x_N))^2$$

$$Stdevp(x_1, \ldots, x_N) = \sqrt{Varp(x_1, \ldots, x_N)}$$

$$Var(x_1, \ldots, x_N) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - Average(x_1, \ldots, x_N))^2$$

$$Stdev(x_1, \ldots, x_N) = \sqrt{Var(x_1, \ldots, x_N)}$$

Covar, Covariance.s, Covariance.p, and Correl

These functions—used extensively in the portfolio chapters (8–13)—are used to compute the covariance and correlation of two series of numbers. The functions **Covariance.s** and **Covariance.p** (computing the sample and the population covariance respectively) are new with Excel 2013. For the definitions, we refer you to section 8.2. Below is an example in which we compute the covariance and correlation for returns on McDonald's stock and Wendy's stock. Note the two computations for the correlation. In the first (cell E9) we use the Excel **Correl** function; in cell E10 we use the definition correlation = *Correlation (MCD, WEN) = Covar (MCD, WEN) / (*$\sigma_{MCD} * \sigma_{WEN}$*)*. The function **Covar** is the population covariance (that is, divides by 1/*M*, where *M* is the population size). Cells E11 and E12 give two more variations to the calculation of the correlation.

	A	B	C	D	E	F
1	**COMPUTING COVARIANCE AND CORRELATION FOR MCDONALD'S (MCD) AND WENDY'S (WEN)** Highlighted cells are new functions with Excel 2013					
2	**Date**	**MCD**	**WEN**		Covariance	
3	1-Aug-05	4.01%	-8.97%		0.00085	<-- =COVAR(B3:B26,C3:C26)
4	1-Sep-05	3.14%	-4.30%		0.00085	<-- =COVARIANCE.P(B3:B26,C3:C26)
5	3-Oct-05	-5.78%	3.42%		0.00085	<-- =COVARIANCE.P(B3:B26,C3:C26)
6	1-Nov-05	8.89%	8.71%		0.00089	<-- =COVARIANCE.S(B3:B26,C3:C26)
7	1-Dec-05	-0.36%	8.44%			
8	3-Jan-06	3.76%	6.47%		Correlation	
9	1-Feb-06	-0.29%	-1.52%		0.36204	<-- =CORREL(B3:B26,C3:C26)
10	1-Mar-06	-1.60%	6.97%		0.36204	<-- =COVAR(B3:B26,C3:C26)/(STDEVP(B3:B26)*STDEVP(C3:C26))
11	3-Apr-06	0.62%	-0.45%		0.36204	<-- =COVARIANCE.P(B3:B26,C3:C26)/(STDEV.P(B3:B26)*STDEV.P(C3:C26))
12	1-May-06	-4.14%	-2.20%		0.36204	<-- =COVARIANCE.S(B3:B26,C3:C26)/(STDEV.S(B3:B26)*STDEV.S(C3:C26))
13	1-Jun-06	1.29%	-3.35%			
14	3-Jul-06	5.20%	3.17%			
15	1-Aug-06	1.41%	6.29%			
16	1-Sep-06	8.61%	4.74%			
17	2-Oct-06	6.90%	9.76%			
18	1-Nov-06	2.53%	-5.79%			

Computing Statistics for a Database

Excel has a whole family of statistical functions that works on databases. These functions all start with the letter "D" (for "data," get it?). The list includes **DAverage**, **DCount**, **DMin**, **DVar** (the sample variance), **DVarP** (the population variance), **DStdev**, and **DStevP**. There are more, but we'll leave them to you to explore.

To see how these functions work, consider the following example: We have a set of monthly stock prices and returns for Apple. Suppose we want to calculate the average return of all the returns that are greater than 10%:

	A	B	C	D	E	F	G	H	I
1	**MANIPULATING APPLE STOCK RETURNS, PRICES USING DAVERAGE, DVAR, etc.**								
2						Criterion range (2 rows)			
3	Date	AAPL	Return			Date	AAPL	Return	
4	7-Sep-01	7.69						>10%	
5	1-Oct-01	8.70	12.34%	<-- =LN(B5/B4)					
6	1-Nov-01	10.56	19.38%			Average	15.38%	<-- =DAVERAGE(A3:C140,3,F3:H4)	
7	3-Dec-01	10.85	2.71%			Variance	0.0021	<-- =DVAR(A3:C140,3,F3:H4)	
8	2-Jan-02	12.25	12.14%			Sigma	4.55%	<-- =DSTDEV(A3:C140,3,F3:H4)	
9	1-Feb-02	10.75	-13.06%						
10	1-Mar-02	11.73	8.72%						
11	1-Apr-02	12.03	2.53%						

Looking at the above example, we see that there are three parts:

- The database is in cells A3:C140. It has headers **Date**, **AAPL**, **Return**.
- There is a two-row **Criterion range**: The top row matches the headers of the database, and the bottom row has ">10%" under **Return**.
- Now the functions: **DAverage(A3:C140,3,F3:H4)** computes the average for all returns that are greater than 10%. **DVar** with the arguments computes the population variance, and **DStdev** …

If you change the criterion range, you get a different answer:

	F	G	H	I
2	Criterion range (2 rows)			
3	Date	AAPL	Return	
4			<9%	
5				
6	Average	-1.71%	<-- =DAVERAGE(A3:C140,3,F3:H4)	
7	Variance	0.0078	<-- =DVAR(A3:C140,3,F3:H4)	
8	Sigma	8.83%	<-- =DSTDEV(A3:C140,3,F3:H4)	

Advanced Use of the "D" Functions

Suppose we want to know the statistics for Apple's returns for a specific year. Here's the trick:

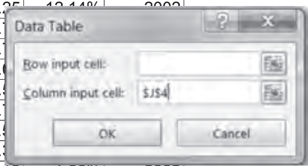
	F	G	H	I
2	Criterion range (2 rows)			
3	Date	AAPL stock price	Return	
4	>38353			<-- =">"&TEXT(G11,"0")
5				
6	Average	2.88%	<-- =DAVERAGE(A3:C140,3,F3:H4)	
7	Variance	0.0114	<-- =DVAR(A3:C140,3,F3:H4)	
8	Sigma	10.67%	<-- =DSTDEV(A3:C140,3,F3:H4)	
9				
10				
11	Date	01-Jan-05		

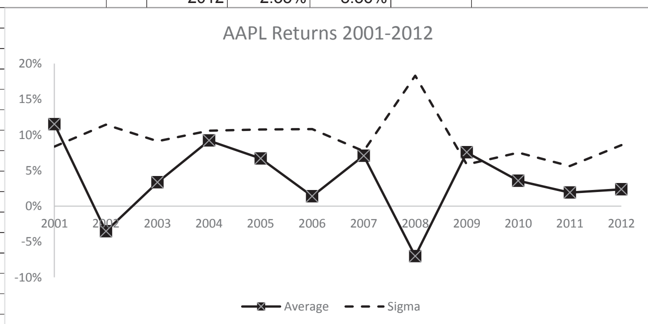
We've put the date in cell G11, and used the **Text** function to create an appropriate criterion under the **Date** header.

One more example and we leave the rest to you. We want to find Apple's return statistics for each of the years. We expand the database and the criterion range by adding in the year. A single example for 2001 shows the average return and sigma for that year to be 11.47% and 3.47%.[5] Running a data table with **Year** as the parameter to be varied gives the results below:

5. As a finance example, this is somewhat marred, since we have only 3 months in 2001.

	A	B	C	D	E	F	G	H	I	J	K
1	**FINDING APPLE'S YEAR-BY-YEAR RETURN STATISTICS**										
2							Criterion range below				
3	Date	AAPL stock price	Return	Year			Date	AAPL stock price	Return	Year	
4	7-Sep-01	7.69								2001	
5	1-Oct-01	8.70	12.34%	2001	<-- =YEAR(A5)						
6	1-Nov-01	10.56	19.38%	2001			Average	11.47%	<-- =DAVERAGE(A3:D140,3,G3:J4)		
7	3-Dec-01	10.85	2.71%	2001			Sigma	8.37%	<-- =DSTDEV(A3:D140,3,G3:J4)		
8	2-Jan-02	12.2	[illegible]	[illegible]							
9	1-Feb-02	10.					**Data base: vary the year in criterion range**				
10	1-Mar-02	11.						Average	Sigma		
11	1-Apr-02	12.								<-- =H7, data base header	
12	1-May-02	11.					2001	11.47%	8.37%		
13	3-Jun-02	8.					2002	-3.53%	11.42%		
14	1-Jul-02	7.					2003	3.33%	9.11%		
15	1-Aug-02	7.					2004	9.19%	10.56%		
16	3-Sep-02	7.19	-1.66%	2002			2005	6.69%	10.75%		
17	1-Oct-02	7.96	10.17%	2002			2006	1.38%	10.80%		
18	1-Nov-02	7.68	-3.58%	2002			2007	7.07%	7.69%		
19	2-Dec-02	7.10	-7.85%	2002			2008	-7.02%	18.29%		
20	2-Jan-03	7.12	0.28%	2003			2009	7.53%	5.85%		
21	3-Feb-03	7.44	4.40%	2003			2010	3.55%	7.51%		
22	3-Mar-03	7.01	-5.95%	2003			2011	1.90%	5.62%		
23	1-Apr-03	7.05	0.57%	2003			2012	2.35%	8.56%		
24	1-May-03	8.90	23.30%	2003							
25	2-Jun-03	9.45	6.00%	2003							
26	1-Jul-03	10.45	10.06%	2003							
27	1-Aug-03	11.21	7.02%	2003							
28	2-Sep-03	10.27	-8.76%	2003							
29	1-Oct-03	11.34	9.91%	2003							
30	3-Nov-03	10.36	-9.04%	2003							
31	1-Dec-03	10.59	2.20%	2003							
32	2-Jan-04	11.18	5.42%	2004							
33	2-Feb-04	11.85	5.82%	2004							
34	1-Mar-04	13.40	12.29%	2004							
35	1-Apr-04	12.78	-4.74%	2004							
36	3-May-04	13.91	8.47%	2004							
37	1-Jun-04	16.13	14.81%	2004							
38	1-Jul-04	16.03	-0.62%	2004							
39	2-Aug-04	17.09	6.40%	2004							
40	1-Sep-04	19.20	11.64%	2004							

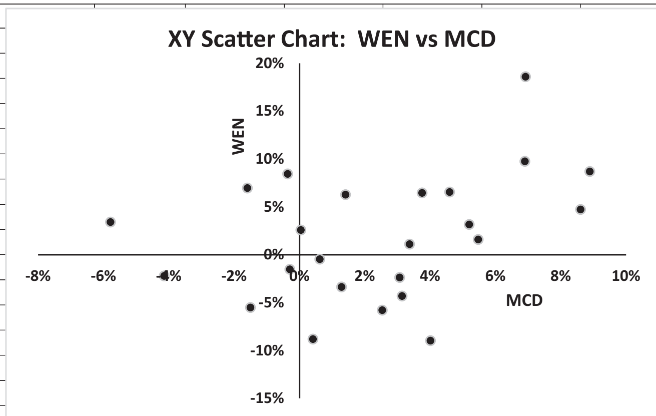




33.6 Regressions with Excel

There are several techniques to produce an ordinary least squares regression with Excel. We illustrate three techniques using the McDonald's and Wendy's data just discussed.

	A	B	C	D	E	F	G	H	I	J	K
1	**REGRESSING WEN ON MCD**										
2	**Date**	**MCD**	**WEN**								
3	1-Aug-05	4.01%	-8.97%								
4	1-Sep-05	3.14%	-4.30%								
5	3-Oct-05	-5.78%	3.42%								
6	1-Nov-05	8.89%	8.71%								
7	1-Dec-05	-0.36%	8.44%								
8	3-Jan-06	3.76%	6.47%								
9	1-Feb-06	-0.29%	-1.52%								
10	1-Mar-06	-1.60%	6.97%								
11	3-Apr-06	0.62%	-0.45%								
12	1-May-06	-4.14%	-2.20%								
13	1-Jun-06	1.29%	-3.35%								
14	3-Jul-06	5.20%	3.17%								
15	1-Aug-06	1.41%	6.29%								
16	1-Sep-06	8.61%	4.74%								
17	2-Oct-06	6.90%	9.76%								
18	1-Nov-06	2.53%	-5.79%								
19	1-Dec-06	5.47%	1.59%		**Regression using functions**						
20	3-Jan-07	0.05%	2.58%		Intercept	0.0038	<-- =INTERCEPT(C3:C26,B3:B26)				
21	1-Feb-07	-1.50%	-5.51%		Slope	0.6381	<-- =SLOPE(C3:C26,B3:B26)				
22	1-Mar-07	3.07%	-2.38%		R-squared	0.1311	<-- =RSQ(C3:C26,B3:B26)				
23	2-Apr-07	6.92%	18.61%								
24	1-May-07	4.59%	6.57%		**Note that**						
25	1-Jun-07	0.41%	-8.80%		Intercept	0.0038	<-- =AVERAGE(C3:C26)-F21*AVERAGE(B3:B26)				
26	2-Jul-07	3.37%	1.11%		Slope	0.6381	<-- =COVARIANCE.P(C3:C26,B3:B26)/VAR.P(B3:B26)				
27						0.6381	<-- =COVARIANCE.S(C3:C26,B3:B26)/VAR.S(B3:B26)				
28					R-squared	0.1311	<-- =CORREL(C3:C26,B3:B26)^2				



Using Excel Functions

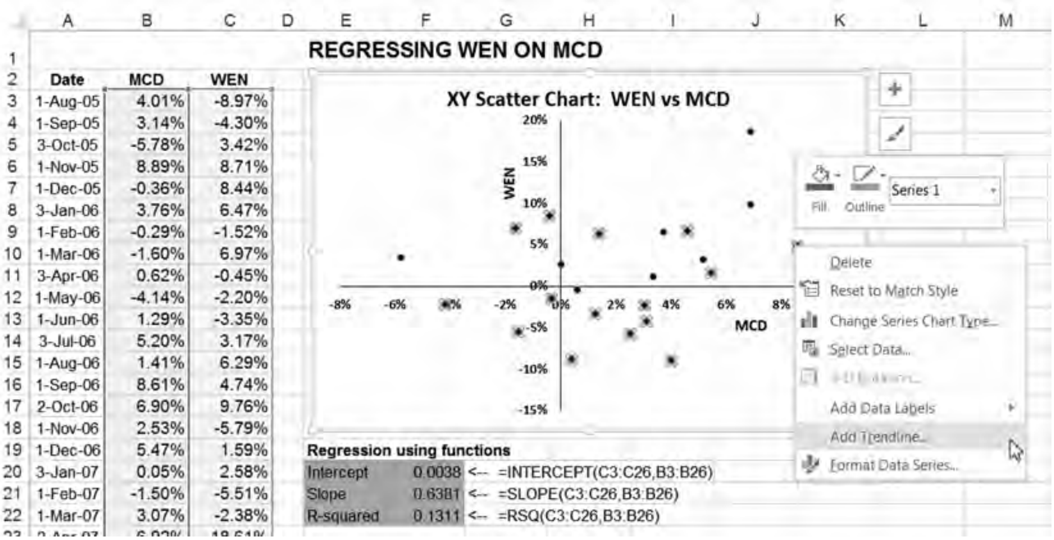
The first technique involves the functions **Slope**, **Intercept**, and **Rsq**; these functions give the parameters for a simple regression of the data in column B on column C. Using these numbers, the best linear explanation of the relation between the returns on WEN and MCD is:

$$WEN = 0.0038 + 0.6381\ MCD,\ R^2 = 13.11\%$$

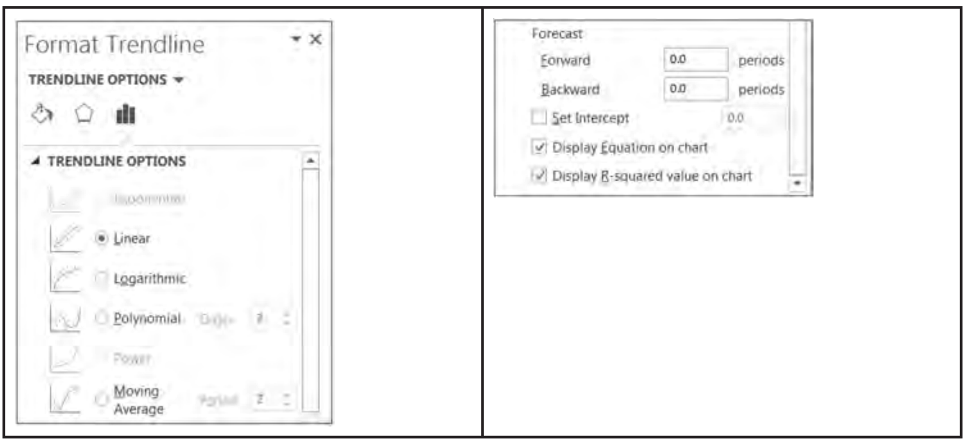
Using a Scatter Plot and Trendline

Another way that we can produce a simple regression is to graph the data in a scatter chart and then use the **Trendline** function to compute the regression.

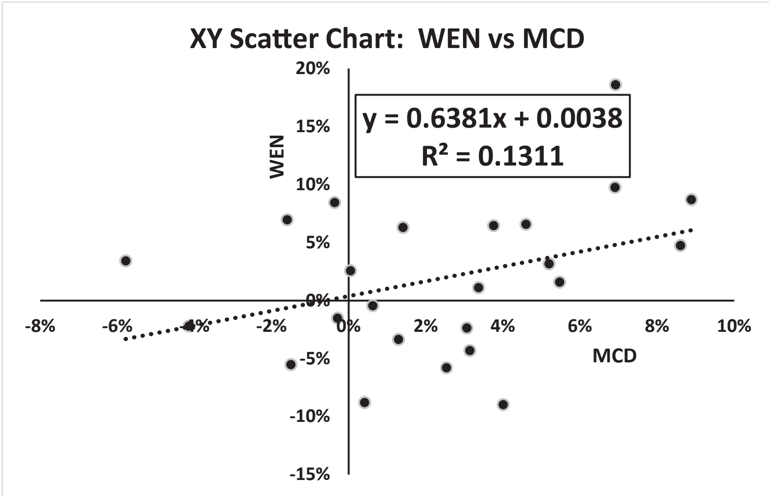
- First plot the data using an **XY Scatter Chart**.
- Click the data and then go to **Add Trendline**. The menu of regressions is given below the picture that follows this bullet.



Here's the menu. We've indicated that we want a linear regression and clicked on two boxes to display the equation and the R^2 in the chart.



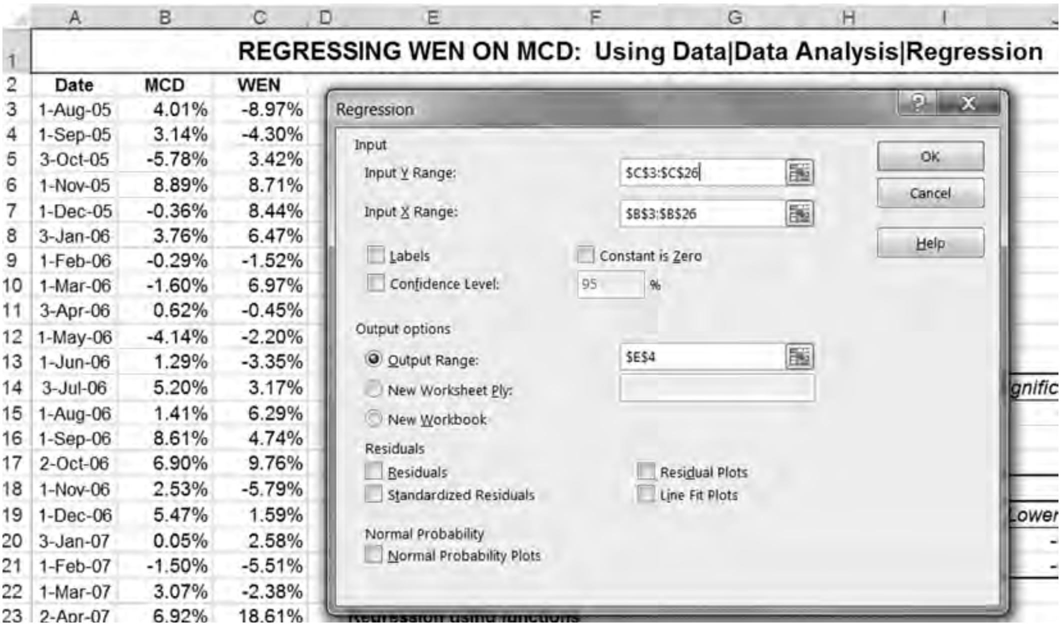
Here's the final result:



Using Data Analysis for Regressions

Using the same data as before, we press **Data|Data Analysis|Regression**.[6] In the picture below, we've asked that the output be put in cell E4:

6. This procedure is available only if you've installed the **Data Analysis** add-in. If you don't see **Data Analysis** on your Excel **Data** menu, go to **File|Options|Add-Ins**. At the bottom of the resulting screen, indicate **Manage Add-Ins**. On the next screen put a checkmark next to **Analysis ToolPak**.

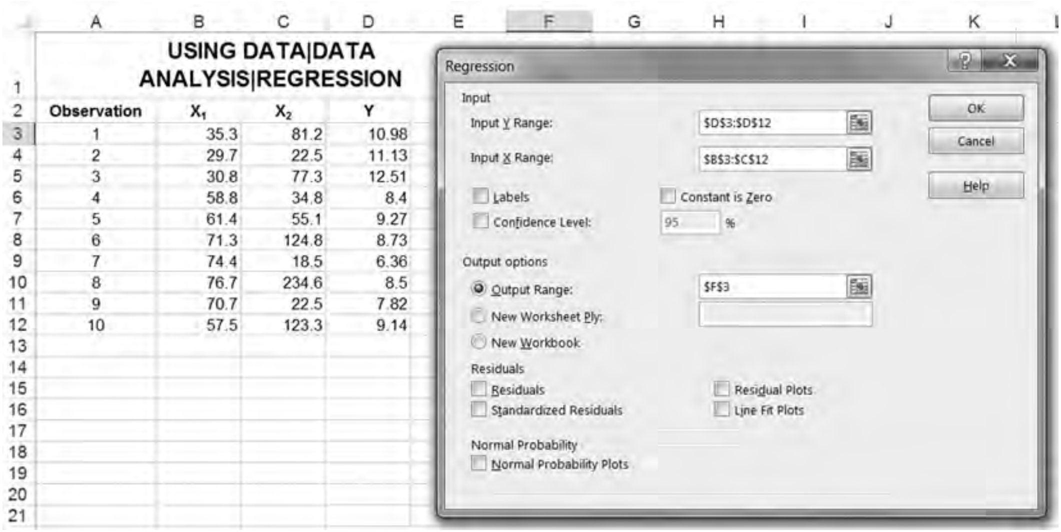


Here's the output. We have highlighted cells corresponding to the intercept, slope, and R^2 as computed by the relevant Excel functions:

	A	B	C	D	E	F	G	H	I	J	K
1	**REGRESSING WEN ON MCD: Using Data|Data Analysis|Regression**										
2	**Date**	**MCD**	**WEN**								
3	1-Aug-05	4.01%	-8.97%								
4	1-Sep-05	3.14%	-4.30%		SUMMARY OUTPUT						
5	3-Oct-05	-5.78%	3.42%								
6	1-Nov-05	8.89%	8.71%		*Regression Statistics*						
7	1-Dec-05	-0.36%	8.44%		Multiple R	0.3620					
8	3-Jan-06	3.76%	6.47%		R Square	0.1311					
9	1-Feb-06	-0.29%	-1.52%		Adjusted R Square	0.0916					
10	1-Mar-06	-1.60%	6.97%		Standard Error	0.0627					
11	3-Apr-06	0.62%	-0.45%		Observations	24					
12	1-May-06	-4.14%	-2.20%								
13	1-Jun-06	1.29%	-3.35%		ANOVA						
14	3-Jul-06	5.20%	3.17%			*df*	*SS*	*MS*	*F*	*Significance F*	
15	1-Aug-06	1.41%	6.29%		Regression	1	0.0131	0.0131	3.3186	0.0821	
16	1-Sep-06	8.61%	4.74%		Residual	22	0.0865	0.0039			
17	2-Oct-06	6.90%	9.76%		Total	23	0.0996				
18	1-Nov-06	2.53%	-5.79%								
19	1-Dec-06	5.47%	1.59%			*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
20	3-Jan-07	0.05%	2.58%		Intercept	0.0038	0.0152	0.2477	0.8067	-0.0278	0.0354
21	1-Feb-07	-1.50%	-5.51%		X Variable 1	0.6381	0.3503	1.8217	0.0821	-0.0883	1.3644
22	1-Mar-07	3.07%	-2.38%								
23	2-Apr-07	6.92%	18.61%								
24	1-May-07	4.59%	6.57%		Intecept	0.0038	<-- =INTERCEPT(C3:C26,B3:B26)				
25	1-Jun-07	0.41%	-8.80%		Slope	0.6381	<-- =SLOPE(C3:C26,B3:B26)				
26	2-Jul-07	3.37%	1.11%		R-squared	0.1311	<-- =RSQ(C3:C26,B3:B26)				

Using Data Analysis|Regression for Multiple Regressions

The same module can be used for multiple regressions. In the picture below, we've asked that the output be placed in cell F3:



Here's the output. Highlighted cells correspond to the most important values produced in the previous example.

	F	G	H	I	J	K	L	M	N
3	SUMMARY OUTPUT								
4									
5	*Regression Statistics*								
6	Multiple R	0.9589							
7	R Square	0.9196							
8	Adjusted R Square	0.8966							
9	Standard Error	0.5783							
10	Observations	10							
11									
12	ANOVA								
13		*df*	*SS*	*MS*	*F*	*Significance F*			
14	Regression	2	26.7674	13.3837	40.0228	0.0001			
15	Residual	7	2.3408	0.3344					
16	Total	9	29.1082						
17									
18		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
19	Intercept	14.1705	0.6271	22.5967	0.0000	12.6877	15.6534	12.6877	15.6534
20	X Variable 1	-0.0987	0.0110	-8.9405	0.0000	-0.1248	-0.0726	-0.1248	-0.0726
21	X Variable 2	0.0089	0.0030	2.9599	0.0211	0.0018	0.0159	0.0018	0.0159

Index

The discussion of the **Index** function belongs in the section on statistics only because we want to use it in our next subsection. We sometimes want to pick an individual value out of an array. In the example below, the range of cells A2:C4 contains a mixture of numbers and names. To pick out an individual item from this range, we use **=Index(A2:C4,row,column)**, where **row** and **column** are relative to the range itself. Thus "Howie" appears in row 2 and column 3 of the range A2:C4.

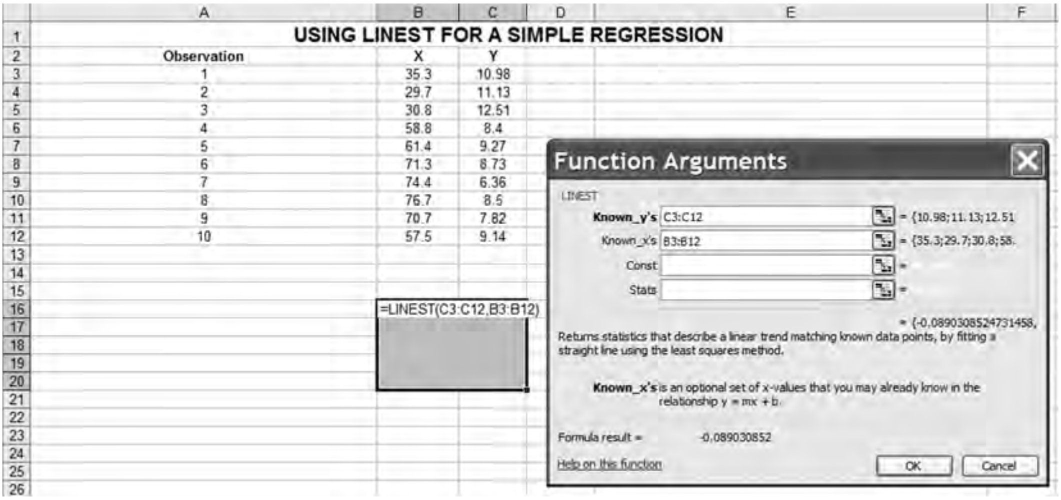
	A	B	C
1	**USING THE INDEX FUNCTION**		
2	a	b	3
3	Simon	6	Howie
4	q	7	Jack
5			
6	Howie	<-- =INDEX(A2:C4,2,3)	

In the next subsection we use the **Index** function to pick out a single item in the **Linest** array.

Using Linest

Excel has an array function **Linest** whose output consists of a number of regression statistics for an ordinary least squares regression.[7] Here is a picture of the spreadsheet and the **Linest** dialog box:

7. There is also an Excel function **Logest**, whose syntax is exactly the same as that of **Linest**. **Logest** calculates the parameters to fit an exponential curve.



Linest is an array function (see next chapter), which means that instead of [Enter], we press [Control] + [Shift] + [Enter] at the same time. With the data from the example above, we can use **Linest** to produce the following output:

	A	B	C	D
1	**USING LINEST FOR A SIMPLE REGRESSION**			
2	**Observation**	**X**	**Y**	
3	1	35.3	10.98	
4	2	29.7	11.13	
5	3	30.8	12.51	
6	4	58.8	8.4	
7	5	61.4	9.27	
8	6	71.3	8.73	
9	7	74.4	6.36	
10	8	76.7	8.5	
11	9	70.7	7.82	
12	10	57.5	9.14	
13				
14		**Linest output**		
15		**slope**	**intercept**	
16	Slope (also **=slope(C3:C12,B3:B12)**)-->	-0.0890	14.3285	<-- Intercept
17	Standard error of slope -->	0.0148	0.8770	<-- Standard error of intercept
18	R^2 (also **=Rsq(C3:C12,B3:B12)**) -->	0.8189	0.8117	<-- Standard error of y values (also **=Steyx(C3:C12,B3:B12)**)
19	F statistic -->	36.1825	8	<-- Degrees of freedom
20	SS_{xy} = Slope*(summed product of observations from means) -->	23.8377	5.2705	<-- SSE = Residual sum of squares
21				
22		Slope	-0.0890	<-- =INDEX(LINEST(C3:C12,B3:B12,,1),1,1)
23		Intercept	14.3285	<-- =INDEX(LINEST(C3:C12,B3:B12,,1),1,2)
24		R^2	0.8189	<-- =INDEX(LINEST(C3:C12,B3:B12,,1),3,1)
25		t-statistic	16.3376	<-- =C23/INDEX(LINEST(C3:C12,B3:B12,,1),2,2)
26				
27		Slope	-0.0890	<-- =INDEX(LINEST(C3:C12,B3:B12,,TRUE),1,1)
28	Standard error of slope		0.0148	<-- =INDEX(LINEST(C3:C12,B3:B12,,TRUE),2,1)
29		t-statistic	-6.0152	<-- =C27/C28

Linest produces a block of output without column headers or row labels which identify the output. Excel's Help provides a good explanation of the meaning of the output; in the above picture, we have added in the explanations.

Note the syntax of this function: **Linest(y-range,x-range,constant, statistics)**. The **y-range** is the range of dependent variables and the **x-range** is the range of the independent variables. If **constant** is omitted (as in the case above) or set to **True**, then the regression is calculated normally; if **constant** is set to **False**, then the intercept is forced to be zero. If **statistics** is set to **True** (as in the case above), then the range of statistics is calculated; otherwise only the slope and intercept are calculated.

Individual items of this output can be accessed by using the function **Index** discussed above. Suppose, for example, that we want to do a simple *t*-test on the slope; doing this requires us to divide the slope value by its standard error.

	A	B	C	D	E	F	G	H	I	J
19			**Linest output**							
20			**slope**	**intercept**						
21		Slope (also **=slope(D4:D13,C4:C13)**)-->	-0.0890	14.3285	<-- Intercept					
22		Standard error of slope -->	0.0148	0.8770	<-- Standard error of intercept					
23		R^2 (also **=Rsq(D4:D13,C4:C13)**) -->	0.8189	0.8117	<-- Standard error of y values (also **=Steyx(D4:D13,C4:C13)**)					
24		F statistic -->	36.1825	8	<-- Degrees of freedom					
25		SS_{xy} = Slope*(summed product of observations from means) -->	23.8377	5.2705	<-- SSE = Residual sum of squares					
26										
27		Slope	-0.0890	<-- =INDEX(LINEST(D4:D13,C4:C13,,1),1,1)						
28		Intercept	14.3285	<-- =INDEX(LINEST(D4:D13,C4:C13,,1),1,2)						
29		R^2	0.8189	<-- =INDEX(LINEST(D4:D13,C4:C13,,1),3,1)						
30										
31		Slope	-0.0890	<-- =INDEX(LINEST(D4:D13,C4:C13,,TRUE),1,1)						
32		S.e. of slope	0.0148	<-- =INDEX(LINEST(D4:D13,C4:C13,,TRUE),2,1)						
33		t-statistic	-6.0152	<-- =C31/C32						

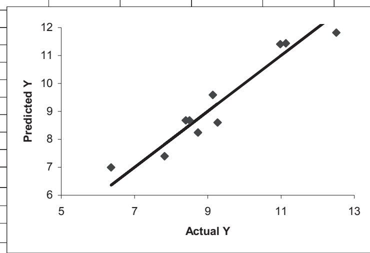
Multiple Regressions with Linest

Linest can also be used to do a multiple regression, as illustrated below:

	A	B	C	D	E	F
1	**USING LINEST TO DO A MULTIPLE REGRESSION**					
2		**Observation**	**X_1**	**X_2**	**Y**	
3		1	35.3	81.2	10.98	
4		2	29.7	22.5	11.13	
5		3	30.8	77.3	12.51	
6		4	58.8	34.8	8.4	
7		5	61.4	55.1	9.27	
8		6	71.3	124.8	8.73	
9		7	74.4	18.5	6.36	
10		8	76.7	234.6	8.5	
11		9	70.7	22.5	7.82	
12		10	57.5	123.3	9.14	
13						
14			**x_2 coeff.**	**x_1 coeff.**	**intercept**	
15		Slope -->	0.0089	-0.0987	14.1705	<-- Intercept
16		Standard error -->	0.0030	0.0110	0.6271	
17		R^2 -->	0.9196	0.5783	#N/A	
18		F statistic -->	40.0228	7.0000	#N/A	
19		SS_{xy} -->	26.7674	2.3408	#N/A	
20						
21						
22						
23			{=LINEST(E3:E12,C3:D12,,TRUE)}			
24						

The predicted versus the actual *Y*'s are shown below:

	A	B	C	D	E	F	G	H	I
1	**USING LINEST TO DO A MULTIPLE REGRESSION**								
2		Observation	X_1	X_2	Y			Predicted Y	
3		1	35.3	81.2	10.98			11.4071	<-- =$E$15+$C$15*D3+$D$15*C3
4		2	29.7	22.5	11.13			11.4394	
5		3	30.8	77.3	12.51			11.8166	
6		4	58.8	34.8	8.4			8.6770	
7		5	61.4	55.1	9.27			8.6004	
8		6	71.3	124.8	8.73			8.2413	
9		7	74.4	18.5	6.36			6.9932	
10		8	76.7	234.6	8.5			8.6817	
11		9	70.7	22.5	7.82			7.3937	
12		10	57.5	123.3	9.14			9.5897	
13									
14			x_2 coeff.	x_1 coeff.	intercept				
15		Slope -->	0.0089	-0.0987	14.1705	<-- Intercept			
16	Standard error -->		0.0030	0.0110	0.6271				
17		R^2 -->	0.9196	0.5783	#N/A				
18		F statistic -->	40.0228	7.0000	#N/A				
19		SS_{xy} -->	26.7674	2.3408	#N/A				
20			{=LINEST(E3:E12,C3:D12,,TRUE)}						The regression equation is Y = 14.1705 - 0.0987*x_1 + 0.0089*x_2 The chart shows the predicted versus the actual Y. If all the predictions were exact (i.e., R2 = 100%), then the predicted points would all fall on the 45-degree line (the dark line).



33.7 Conditional Functions

If, **VLookup**, and **HLookup** are three functions which allow you to put in conditional statements.

The syntax of Excel's **If** statement is **If(condition,output if condition is true, output if condition is false)**. In the example below, if the initial number is B3 < 3, then the desired output is 15. If B3 > 3, then the output is 0:

	A	B	C
1	**THE IF FUNCTION**		
2	Initial number	2	
3	If statement	15	<-- =IF(B2<=3,15,0)
4			
5	Initial number	2	
6	If statement	Less than or equal to 3	<-- =IF(B5<=3,"Less than or equal to 3","More than 3")

As you can see in row 6, you can make **If** print text also, by enclosing the desired text in double quotes. Since **VLookup**, and **HLookup** both have the same structure, we will concentrate on **VLookup** and leave you to figure out

HLookup for yourself. **VLookup** is a way to introduce a table search in your spreadsheet. Here is an example: Suppose the marginal tax rates on income are given by the table below (i.e., for income less than $8,000, the marginal tax rate is 0%; for income above $8,000, the marginal tax rate is 15%). Cell B9 illustrates how the function **VLookup** is used to look up the marginal tax rate.

	A	B	C
1	**VLOOKUP FUNCTION**		
2	**Income**	**Tax rate**	
3	0	0%	
4	8,000	15%	
5	14,000	25%	
6	25,000	38%	
7			
8	Income	15,000	
9	Tax rate	25%	<-- =VLOOKUP(B8,A3:B6,2)

The syntax of this function is **VLookup(lookup_value,table,column)**. The first column of the lookup table, A3:B6, must be arranged in ascending (increasing) order. The **lookup_value**, in this case the income of 15,000, is used to determine the applicable row of the **table**. The row is the first row whose value is < the **lookup_value**; in this case, this is the row which starts with 14,000. The **column** entry determines from which column of the applicable row the answer is taken; in this case the marginal tax rates are in column 2.

33.8 Large and Rank, Percentile, and PercentRank

Large(array, k) returns the *k*th largest number of the **array**, and **Rank(number, array)** returns the rank in **array** of **number**.

Here is an example of each function:

	A	B	C
1	**LARGE, RANK, PERCENTILE, PERCENTRANK**		
2	**Data**		
3	10.98		
4	11.13		
5	12.51		
6	8.40		
7	9.27		
8	8.73		
9	6.36		
10	8.50		
11	7.82		
12	9.14		
13			
14	Ranking, k	3	
15	K-th largest	10.98	<-- =LARGE(A3:A12,B14)
16			
17	Specific number	9.27	
18	Rank from top	4	<-- =RANK(B17,A3:A12)
19	Rank from bottom	7	<-- =RANK(B17,A3:A12,1)
20			
21	Percentile rank	0.8	
22	Percentile	11.01	<-- =PERCENTILE(A3:A12,B21)
23			
24	Specific number	9.27	
25	Percentile ranking	0.666	<-- =PERCENTRANK(A3:A12,B24)

Thus the third-largest number in the range A3:A12 is 10.98 and 9.27 is the fourth-largest number in the range A3:A12. If, as in cell B19, you specify an additional parameter in the function **Rank**, you will see that 9.27 is the seventh-ranking number from the bottom of the range A3:A12.

As illustrated, Excel has similar functions for percentiles: **Percentile** and **PercentRank**.

33.9 Count, CountA, CountIf, CountIfs, AverageIf, AverageIfs

As their names suggest, all six of these functions count:

- **Count**: Counts the number of numeric entries in a range of cells.
- **CountA**: Counts all the non-blank cells in a range.
- **CountIf**: Counts cells which fulfill a specific condition.

- **CountIfs**: Counts cells depending on multiple conditions.
- **AverageIf** and **AverageIfs**: Obvious.

Examples of **Count** and **CountA** are given below:

	A	B	C	D	E	F	G
1	**COUNT, COUNTA**						
2	**Count**: Count only numerical values	5	<-- =COUNT(E2:G4)		1	two	3
3	**CountA**: count all non-blank cells	8	<-- =COUNTA(E2:G4)		4		six
4					seven	8	9

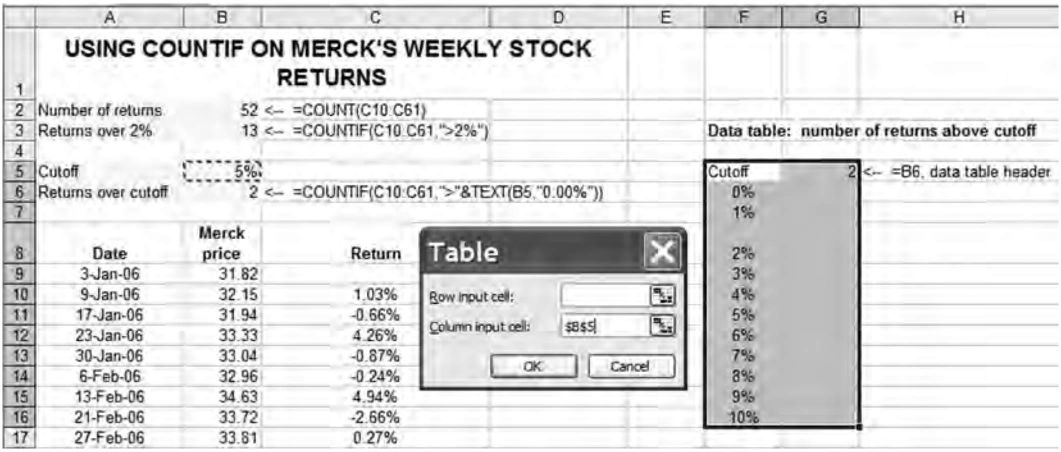
To use **CountIf** we have to specify a condition. The spreadsheet below gives a year of Merck's weekly stock returns (some rows have been hidden):

	A	B	C	D
1	**USING COUNTIF ON MERCK'S WEEKLY STOCK RETURNS**			
2	Number of returns	52	<-- =COUNT(C10:C61)	
3	Returns over 2%	13	<-- =COUNTIF(C10:C61,">2%")	
4				
5	Cutoff	5%		
6	Returns over cutoff	2	<-- =COUNTIF(C10:C61,">"&TEXT(B5,"0.00%"))	
7				
8	**Date**	**Merck price**	**Return**	
9	3-Jan-06	31.82		
10	9-Jan-06	32.15	1.03%	<-- =LN(B10/B9)
11	17-Jan-06	31.94	-0.66%	
12	23-Jan-06	33.33	4.26%	
13	30-Jan-06	33.04	-0.87%	
14	6-Feb-06	32.96	-0.24%	
15	13-Feb-06	34.63	4.94%	
16	21-Feb-06	33.72	-2.66%	
17	27-Feb-06	33.81	0.27%	
18	6-Mar-06	33.76	-0.15%	
19	13-Mar-06	34.61	2.49%	

In cell B3 we count all stock returns which are over 2%. In cells B5:B6 we illustrate a different technique. The cutoff in cell B5 is introduced into **CountIf** by means of the function **Text**.[8] Changing the entry in cell B5 allows us to

8. This function and other text functions are discussed in Chapter 35.

count the number of returns above a certain level. Here is this information in a **Data Table**:



	A	B	C	D	E	F	G	H
1	USING COUNTIF ON MERCK'S WEEKLY STOCK RETURNS							
2	Number of returns	52	<-- =COUNT(C10:C61)					
3	Returns over 2%	13	<-- =COUNTIF(C10:C61,">2%")			Data table: number of returns above cutoff		
4								
5	Cutoff	5%				Cutoff	2	<-- =B6, data table header
6	Returns over cutoff	2	<-- =COUNTIF(C10:C61,">"&TEXT(B5,"0.00%"))			0%		
7						1%		
8	Date	Merck price	Return			2%		
9	3-Jan-06	31.82				3%		
10	9-Jan-06	32.15	1.03%			4%		
11	17-Jan-06	31.94	-0.66%			5%		
12	23-Jan-06	33.33	4.26%			6%		
13	30-Jan-06	33.04	-0.87%			7%		
14	6-Feb-06	32.96	-0.24%			8%		
15	13-Feb-06	34.63	4.94%			9%		
16	21-Feb-06	33.72	-2.66%			10%		
17	27-Feb-06	33.81	0.27%					

Here is the resulting table:

	A	B	C	D	E	F	G	H
1	USING COUNTIF ON MERCK'S WEEKLY STOCK RETURNS							
2	Number of returns	52	<-- =COUNT(C10:C61)					
3	Returns over 2%	13	<-- =COUNTIF(C10:C61,">2%")			Data table: number of returns above cutoff		
4								
5	Cutoff	5%				Cutoff	2	<-- =B6, data table header
6	Returns over cutoff	2	<-- =COUNTIF(C10:C61,">"&TEXT(B5,"0.00%"))			0%	30	
7						1%	21	
8	Date	Merck price	Return			2%	13	
9	3-Jan-06	31.82				3%	8	
10	9-Jan-06	32.15	1.03%	<-- =LN(B10/B9)		4%	6	
11	17-Jan-06	31.94	-0.66%			5%	2	
12	23-Jan-06	33.33	4.26%			6%	1	
13	30-Jan-06	33.04	-0.87%			7%	1	
14	6-Feb-06	32.96	-0.24%			8%	1	
15	13-Feb-06	34.63	4.94%			9%	1	
16	21-Feb-06	33.72	-2.66%			10%	0	

Of the 52 weekly returns, 30 are over 0%, 21 over 1%,…
We can use the function **Countifs** to introduce multiple-criteria counts:

	A	B	C
1	**USING COUNTIFS FOR MULTIPLE-CRITERIA COUNTS**		
2	Lower cutoff	2%	
3	Upper cutoff	4%	
4	One cutoff	13	<-- =COUNTIF(C:C,">2%")
5	Multiple cutoffs	7	<-- =COUNTIFS(C:C,"<4%",C:C,">2%")
6		7	<-- =COUNTIFS(C:C,"<"&TEXT(B3,"0%"),C:C,">"&TEXT(B2,"0%"))
7			
8	**Date**	**Merck price**	**Return**
9	03-Jan-06	31.82	
10	09-Jan-06	32.15	1.03%
11	17-Jan-06	31.94	-0.66%
12	23-Jan-06	33.33	4.26%
13	30-Jan-06	33.04	-0.87%

33.10 Boolean Functions

When you include a question in parentheses, you are setting up a *Boolean function*:

	A	B	C
1	**BASIC BOOLEAN FUNCTIONS**		
2	x	22	
3	y	-15	
4			
5	Number	25	
6	Is number <= x?	FALSE	<-- =(B5<=B2)
7	Is number > y?	TRUE	<-- =(B5>B3)
8			
9	Multiplying	0	<-- =B6*B7

In cell B6 we have written **=(B5<=B2)**; this asks whether B5 is less than or equal to B2: If the answer is positive, Excel returns **False**, else it returns **True**. Multiplying **False*True** or **False*False** gives 0 (see cell B9), and multiplying **True*True** gives 1:

	A	B	C
1	**BASIC BOOLEAN FUNCTIONS**		
2	x	22	
3	y	-15	
4			
5	Number	20	
6	Is number <= x?	TRUE	<-- =(B5<=B2)
7	Is number > y?	TRUE	<-- =(B5>B3)
8			
9	Multiplying	1	<-- =B6*B7

Using Boolean Functions

Boolean functions can be useful in the most unexpected places. In the spreadsheet below, the first two columns contain the monthly returns for Marriott for a 2-year period. The problem we encounter is counting the number of returns which are between two arbitrary bounds, and taking the average of the returns which are between these bounds:

	A	B	C	D	E	F	G	H
1	**USING BOOLEAN FUNCTIONS**							
2	**Date**	**Marriott stock price**	**Return**					
3	7-Jan-05	31.20				How many data points?	24	<-- =COUNT(C4:C27)
4	1-Feb-05	31.65	1.43%	<-- =LN(B4/B3)		Maximal return	8.02%	<-- =MAX(C4:C27)
5	1-Mar-05	33.06	4.36%			Minimal return	-8.02%	<-- =MIN(C4:C27)
6	1-Apr-05	31.03	-6.34%					
7	2-May-05	33.40	7.36%					
8	1-Jun-05	33.78	1.13%			Upper bound	5%	
9	1-Jul-05	33.91	0.38%			Lower bound	-2%	
10	1-Aug-05	31.30	-8.01%					
11	1-Sep-05	31.25	-0.16%			How many < upper bound?	16	<-- =COUNTIF(C4:C27,"<"&G8)
12	3-Oct-05	29.58	-5.49%			How many > lower bound?	19	<-- =COUNTIF(C4:C27,">"&G9)
13	1-Nov-05	32.05	8.02%					
14	1-Dec-05	33.27	3.74%			How many between upper and lower bounds?	11	=SUMPRODUCT((C4:C27>G9)*(C4:C27<G8),(C4:C27>G9)*(C4:C27<G8))
15	3-Jan-06	33.11	-0.48%					
16	1-Feb-06	33.98	2.59%			Average of returns between the bounds	1.36%	=SUMPRODUCT((C4:C27>G9)*(C4:C27<G8),C4:C27)/SUMPRODUCT((C4:C27>G9)*(C4:C27<G8),(C4:C27>G9)*(C4:C27<G8))
17	1-Mar-06	34.14	0.47%					
18	3-Apr-06	36.36	6.30%					
19	1-May-06	35.99	-1.02%					
20	1-Jun-06	38.00	5.43%					
21	3-Jul-06	35.07	-8.02%					
22	1-Aug-06	37.61	6.99%					
23	1-Sep-06	38.59	2.57%					
24	2-Oct-06	41.71	7.77%					
25	1-Nov-06	45.09	7.79%					
26	1-Dec-06	47.72	5.67%					
27	3-Jan-07	45.10	-5.65%					

In cell G3 we use **Count** to determine the number of returns. Cells G11 and G12 use **CountIf** to determine how many returns are below the upper bound in cell G8 and how many are above the lower bound in cell G9. But how many of the returns are between the two bounds? You can't do it with **CountIf**, but we can do it (cell G14) with a trick involving Boolean functions:

$$=\text{SUMPRODUCT}(\underbrace{(\text{C4:C27} > \text{G9})*(\text{C4:C27} < \text{G8})}_{\substack{\uparrow \\ \text{Creates a vector of 1's where} \\ \text{the returns are} > \text{lower bound (G9)} \\ \text{and 0's where} \\ \text{the returns are} < \text{upper bound (G9)}}},$$

$$\underbrace{(\text{C4:C27} > \text{G9})*(\text{C4:C27} < \text{G8})}_{\substack{\uparrow \\ \text{Ditto}}})$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\substack{\uparrow \\ \text{Sumproduct multiplies the two vectors and sums the results.} \\ \text{This gives the number of data points between the two bounds.}}}$$

A similar trick is employed in cell G15 to find the average of the returns which are between the two bounds:

$$= \frac{\text{SUMPRODUCT}((\text{C4:C27} > \text{G9})*(\text{C4:C27} < \text{G8}), \text{C4:C27})}{\begin{array}{c}\text{SUMPRODUCT}((\text{C4:C27} > \text{G9})*(\text{C4:C27} < \text{G8}), \\ (\text{C4:C27} > \text{G9})*(\text{C4:C27} < \text{G8}))\end{array}}$$

$$= \frac{\begin{array}{l}\text{Numerator:}\quad \text{Multiplies vector of 1's and 0's times the returns,} \\ \quad\text{thus sums the returns which are between the bounds}\end{array}}{\text{Counts the returns which are between the bounds}}$$

This is all very tricky, but it is also very useful!

33.11 Offset

The function **Offset** allows us to specify a cell or a block of cells in an array. It cannot be used by itself—rather, it must be part of another Excel function. The example below shows a large array of numbers. We want to sum a four-row, five-column array of the larger array (these numbers are specified in cells B6 and B7); as specified in cells B3 and B4, we want this summed array to start to the right and below the third row and the second column of the large block of numbers:

	A	B	C	D	E	F	G	H
1	**USING OFFSET**							
2	**Starting corner**							
3	Rows down	3						
4	Columns over	2						
5	**Range to be summed**							
6	Number of rows	4						
7	Number of columns	5						
8	Sum	811	<-- =SUM(OFFSET(A11:H31,B3,B4,B6,B7))					
9	**Check**	811	<-- =SUM(C14:G17)					
10								
11	89	34	72	42	41	89	75	41
12	33	6	49	7	62	50	38	17
13	71	69	42	68	39	75	32	77
14	1	69	8	79	40	8	67	46
15	70	12	44	48	88	27	38	51
16	85	0	23	35	83	30	17	52
17	30	50	16	28	73	4	55	68
18	35	56	31	24	15	47	89	88
19	99	31	55	60	45	24	28	3
20	93	72	7	75	90	81	52	71
21	62	56	55	19	73	81	33	76
22	87	27	80	38	65	61	38	68
23	10	59	27	81	6	83	51	1
24	70	88	44	35	70	35	0	82
25	98	45	17	45	89	19	58	42
26	83	75	21	13	80	9	18	64
27	32	23	4	86	88	52	52	69
28	76	61	72	28	83	1	32	38
29	64	87	32	67	50	73	19	83
30	54	55	57	64	80	29	17	92
31	12	95	66	59	48	78	87	23

The function **OFFSET(A11:H31,B3,B4,B6,B7)** in cell B8 specifies a block of cells within the range A11:H31. This range starts *three rows below* (the value in cell B3) and *two columns to the right* (the value in cell B4) of the top left-hand cell of the range A11:H31. The range itself is four rows deep (the value in cell B6) and five columns wide (the value in cell B7).

The values in cells B6 and B7 always have to be positive, but the values in B3 and B4 can be negative as well as positive. In the example below the initial reference range is B22:H31, and **Offset** indicates a range which starts above this block of cells (since the value in B3 is negative):

	A	B	C	D	E	F	G	H	I
1	**USING OFFSET with negative value**								
2	**Starting corner**								
3	Rows down	-5							
4	Columns over	1							
5	**Range to be summed**								
6	Number of rows	4							
7	Number of columns	5							
8	Sum	899	<-- =SUM(OFFSET(B22:H31,B3,B4,B6,B7))						
9	**Check**	899	<-- =SUM(C17:G20)						
10									
11	89	34	72	42	41	89	75	41	
12	33	6	49	7	62	50	38	17	
13	71	69	42	68	39	75	32	77	
14	1	69	8	79	40	8	67	46	
15	70	12	44	48	88	27	38	51	
16	85	0	23	35	83	30	17	52	
17	30	50	16	28	73	4	55	68	
18	35	56	31	24	15	47	89	88	
19	99	31	55	60	45	24	28	3	
20	93	72	7	75	90	81	52	71	
21	62	56	55	19	73	81	33	76	
22	87	27	80	38	65	61	38	68	
23	10	59	27	81	6	83	51	1	
24	70	88	44	35	70	35	0	82	
25	98	45	17	45	89	19	58	42	
26	83	75	21	13	80	9	18	64	
27	32	23	4	86	88	52	52	69	
28	76	61	72	28	83	1	32	38	
29	64	87	32	67	50	73	19	83	
30	54	55	57	64	80	29	17	92	
31	12	95	66	59	48	78	87	23	
32									

For an innovative use of **Offset**, see Chapter 14, section 14.6.

34 Array Functions

34.1 Overview

An Excel array function or formula performs an operation on a rectangular block of cells. In the simplest cases, built-in Excel array functions such as **Transpose** or **MMult** take an array and transpose it or take two matrices and multiply them. Once you get the hang of array functions, you can design your own array formulas. In this chapter, for example, we show how to use array formulas to find the minimum or maximum of the off-diagonal elements of a matrix or to pick out the diagonal of a matrix—all useful tricks to know when doing portfolio calculations such as those discussed in Chapters 8–13.

The one critical thing to remember about array functions or formulas is that they are entered into a spreadsheet by pressing [Ctrl] + [Shift] + [Enter] keys simultaneously; this contrasts with the usual procedure whereby we enter a function or formula only by pressing [Enter].

34.2 Some Built-In Excel Array Functions

In this section we discuss some built-in Excel array functions: **Transpose**, **MMult**, **MInverse**, and **Frequency**. Other functions are discussed elsewhere in this book—for example, the function **Linest** is discussed in Chapter 32.

Transpose

Suppose we're trying to calculate the transpose of a 3 x 2 (3 rows, 2 columns) matrix that is in cells A2:B4 of the spreadsheet.

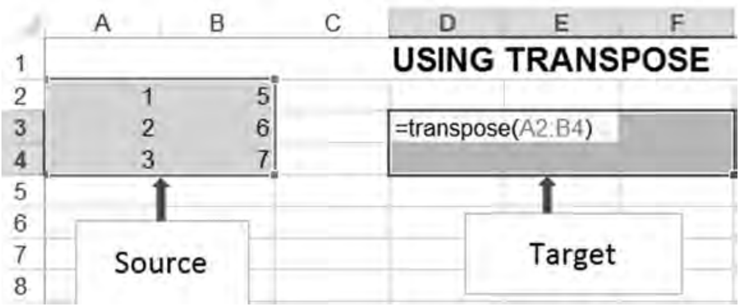
	A	B
2	1	5
3	2	6
4	3	7

Excel has a function called **Transpose( )**, but, like all array functions, its use requires care:

• **Mark the target**: Block off the cells D3:F4 into which you intend to put the transposed matrix.

• **Type the array function**: Now type **=Transpose(A2:B4)**. This will appear in the top left-hand corner of the blocked-off cells. Of course you can use the usual tricks to show Excel which cells you want (e.g., pointing or using named ranges).

At this point your spreadsheet looks like this:



• **[Ctrl] + [Shift] + [Enter]**: When you've finished typing the formula, *don't press [Enter]*! Instead, use [Ctrl] + [Shift] + [Enter]. This will put the array function into all of the blocked-off cells.

Here's what the final product will look like:

	A	B	C	D	E	F	G
1	**USING TRANSPOSE**						
2	1	5					
3	2	6		1	2	3	<-- {=TRANSPOSE(A2:B4)}
4	3	7		5	6	7	

Notice that the array function is surrounded by curly brackets { }. You don't type these in—Excel puts these in automatically.

Paste|Paste Special|Transpose

There is, of course, another way to transpose an array: You can copy the original array, and then use **Paste|Paste Special** to transpose the range, clicking on **Transpose**:

This will transpose the range, but it will not link the original with the target range—when you change something in the original range, the target range is unchanged. The neat thing about the array function **Transpose** is that it's a *dynamic function*, as are all the array functions and formulas: When you change one of the initial set of cells, the transposed array also changes.

MMult and MInverse—Multiplying and Inverting Matrices

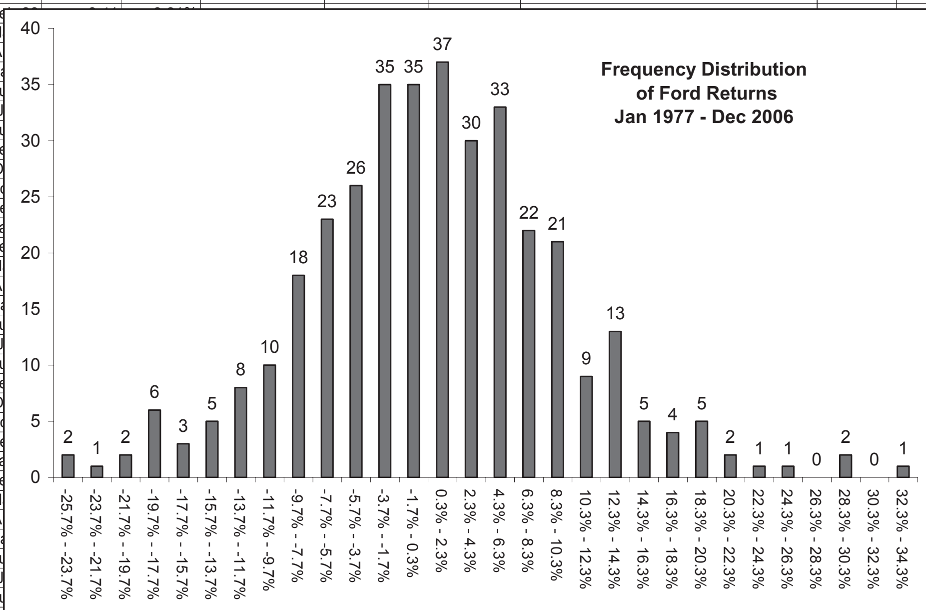
These two functions are used and explained in the portfolio chapters (8–13), so we only recapitulate briefly:

- **MMult(range1,range2)** multiplies the matrix in **range1** times that in **range2**. Of course this is only possible if the number of columns in **range1** equals the number of rows in **range2**.
- **MInverse(range)** calculates the inverse of the matrix in **range**. Note that **range** must be rectangular.

Frequency

The Excel array function **Frequency(data_array,bins_array)** calculates the frequency distribution of a data set. The spreadsheet below shows monthly return data for Ford stock over the period January 1977–December 2006. In column E we have put the bins, taking care that the first bin will be *below* the minimum monthly return over the period and that the last bin will be *above* the maximum monthly return. The range F8:F38 contains the array function **Frequency (C4:C363,E8:E38)**. From the output we can, for example, deduce that in the 20-year period there were two monthly returns between –25.71% and –23.71%, and 33 monthly returns between 4.29% and 6.29%.

	A	B	C	D	E	F	G	H	I
1	**THE FREQUENCY ARRAY FUNCTION**								
2	**Date**	**Ford stock price**	**Monthly return**						
3	3-Jan-77	0.50							
4	1-Feb-77	0.48	-4.08%	<-- =LN(B4/B3)	Minimum	-24.71%	<-- =MIN(C4:C363)		
5	1-Mar-77	0.45	-6.45%		Maximum	32.48%	<-- =MAX(C4:C363)		
6	1-Apr-77	0.47	4.35%						
7	2-May-77	0.46	-2.15%		**Bin**	**Frequency**			
8	1-Jun-77	0.50	8.34%		-25.71%	0	<-- {=FREQUENCY(C4:C363,E8:E38)}		
9	1-Jul-77	0.48	-4.08%		-23.71%	2			
10	1-Aug-77	0.47	-2.11%		-21.71%	1			
11	1-Sep-77	0.50	6.19%		-19.71%	2			
12	3-Oct-77	0.47	-6.19%		-17.71%	6			
13	1-Nov-77	0.48	2.11%		-15.71%	3			
14	1-Dec-77	0.50	4.08%		-13.71%	5			
15	3-Jan-78	0.47	-6.19%		-11.71%	8			
16	1-Feb-78	0.47	0.00%		-9.71%	10			
17	1-Mar-78	0.51	8.17%		-7.71%	18			
18	3-Apr-78	0.58	12.86%		-5.71%	23			
19	1-May-78	0.56	-3.51%		-3.71%	26			
20	1-Jun-78	0.53	-5.51%		-1.71%	35			
21	3-Jul-78	0.53	0.00%		0.29%	35			
22	1-Aug-78	0.51	-3.85%		2.29%	37			
23	1-Sep-78	0.53	3.85%		4.29%	30			
24	2-Oct-78	0.48	-9.91%		6.29%	33			
25	1-Nov-78	0.49	2.06%		8.29%	22			
26	1-Dec-78	0.50	2.02%		10.29%	21			
27	2-Jan-79	0.50	0.00%		12.29%	9			
28	1-Feb-79	0.50	0.00%		14.29%	13			
29	1-Mar-79	0.53	5.83%		16.29%	5			
30	2-Apr-79	0.55	3.70%		18.29%	4			
31	1-May-79	0.53	-3.70%		20.29%	5			
32	1-Jun-79	0.53	0.00%		22.29%	2			
33	2-Jul-79	0.53	0.00%		24.29%	1			
34	1-Aug-79	0.55	3.70%		26.29%	1			
35	4-Sep-79	0.56	1.80%		28.29%	0			
36	1-Oct-79	0.48	-15.42%		30.29%	2			
37	1-Nov-79	0.40	-18.23%		32.29%	0			
38	3-Dec-79	0.42	4.88%		34.29%	1			
39	2-Jan-80	0.45	6.90%						
40	1-Fe								
41	3-M								
42	1-A								
43	1-Ma								
44	2-Ju								
45	1-J								
46	1-Au								
47	2-Se								
48	1-O								
49	3-No								
50	1-De								
51	2-Ja								
52	2-Fe								
53	2-M								
54	1-A								
55	1-Ma								
56	1-Ju								
57	1-J								
58	3-Au								
59	1-Se								
60	1-O								
61	2-No								
62	1-De								
63	4-Ja								
64	1-Fe								
65	1-M								
66	1-A								
67	3-Ma								
68	1-Ju								
69	1-J								
70	2-Au								
71	1-Sep-82	0.39	-7.41%						



34.3 Homemade Array Functions

In our experience array functions often arise out of situations where you are called upon to do long, repetitive calculations. You then discover that the same calculation can be done in a single array function. In many cases it is not clear why a particular array technique should work. For example, in this chapter we will use the fact that A3+B6:B8 adds the contents of cell A3 to each of cells B6:B8. Why? Heaven only knows! We also use the (undocumented, as far as we are aware) trick that B3:B7∧A3:A7 raises cell B3 to the power A3, cell B4 to the power A4, ...

In this section we illustrate homemade array functions with two examples having to do with investment returns.

Computing the Compound Annual Return from 10 Years of Return Data

The table below gives the annual returns for the Harvard University endowment. You are asked to compute the compound annual return over the 10-year period. Assuming that the returns have been discretely computed (meaning that $r_t = \frac{Endowment\ value_t}{Endowment\ value_{t-1}}, t = 1, \ldots, 10$), you realize that the compounded annual return is $r = ((1+r_{2002})*(1+r_{2003})\ldots*(1+r_{2011}))^{1/10} - 1$. In cell B14 below, we do this calculation with a single array function:

	A	B	C
1	**HARVARD UNIVERSITY ENDOWMENT RETURNS** Years ending June 30		
2	**Year**	**Return**	
3	2002	-0.50%	
4	2003	12.50%	
5	2004	21.20%	
6	2005	19.20%	
7	2006	16.70%	
8	2007	23.00%	
9	2008	8.60%	
10	2009	-27.30%	
11	2010	11.40%	
12	2011	21.40%	
13			
14	Compound annual return	9.50%	<-- {=PRODUCT(1+B3:B12)^(1/10)-1}

The Excel function **Product** multiplies the entries in a range of cells. The entry in cell B14 adds 1 to each cell in B3:B12, multiplies the cells, takes the tenth root, and subtracts 1 from the result—all in one cell (entered, of course, with [Ctrl] + [Shift] + [Enter]).[1]

Computing the Compound Annual Continuous Return

Column B of the spreadsheet below gives the amounts which accumulated in a customer account of the Youngtalk Investment Fund. The annual continuously compounded return is computed by $r_t = \ln\left(\frac{Account_t}{Account_{t-1}}\right)$, and the average return over the period is $\frac{1}{10}\sum_{t=1}^{10} r_t$. In cell B15 below we do this calculation by averaging the annual returns, and in cell B16 we show an array function which does the whole calculation in a single cell. Pretty neat!

	A	B	C	D
1	**YOUNGTALK INVESTMENT FUND**			
2	**Year**	**Investment beginning of year**	**Continuous return for year**	
3	1996	100.00		
4	1997	121.51	19.48%	<-- =LN(B4/B3)
5	1998	132.22	8.45%	
6	1999	98.63	-29.31%	
7	2000	75.65	-26.53%	
8	2001	140.48	61.90%	
9	2002	221.40	45.49%	
10	2003	243.46	9.50%	
11	2004	280.11	14.02%	
12	2005	398.72	35.31%	
13	2006	543.58	30.99%	
14				
15	Compound annual return		16.93%	<-- =AVERAGE(C4:C13)
16	Same calculation with array function		16.93%	<-- {=AVERAGE(LN(B4:B13/B3:B12))}
17			16.93%	<-- =LN(B13/B3)/10 , even simpler!

1. There is nothing in Excel documentation that indicates why this marvelous feature should work. But it does …

A final note: Look at cell C17—if you know some continuous-time mathematics, you will know that **LN(B13/B3)/10** produces the same result. Simpler yet!

Computing Discount Factors with an Array Function

We are given a set of interest rates r_1, r_2, … , and we want to compute the formula $\sum_{t=1}^{n} \frac{1}{(1+r_t)^t}$. Excel's **NPV** function won't work for this, so we'll have to build our own function. The spreadsheet below shows two methods:

	A	B	C	D
1	**COMPUTING PRESENT VALUE FACTORS WITH AN ARRAY FUNCTION**			
2	Year	Interest rate		
3	1	6.23%		
4	2	4.00%		
5	3	4.20%		
6	4	4.65%		
7	5	4.80%		
8				
9	Present value	4.3746	<-- {=SUM(1/((1+B3:B7)^A3:A7))}	
10				
11	**Checking the formula with a recursive formula**			
12	Year	Interest rate	Sum of $1/(1+r_t)^t$	
13	1	6.23%	0.9414	<-- =1/(1+B13)^A13
14	2	4.00%	1.8659	<-- =C13+(1/(1+B14)^A14)
15	3	4.20%	2.7498	<-- =C14+(1/(1+B15)^A15)
16	4	4.65%	3.5836	
17	5	4.80%	4.3746	<-- =C16+(1/(1+B17)^A17)

In cell B9 we use an array function **{=SUM(1/((1+B3:B7)∧A3:A7))}**. Writing (1+B3:B7)∧A3:A7) adds 1 to each of cells B3:B7 and raises the result to the power in cells A3:A7. Applying **Sum** gives the result (of course, this being an array function, you have to enter it with [Ctrl] + [Shift] + [Enter]).

An alternative, given in rows 13–17, is to build the result recursively. This gives the same result, but with more work.

34.4 Array Formulas with Matrices

In this section we create some array functions that have to do with matrices.

Subtracting a Constant from a Matrix

In the portfolio computations of Chapters 8–13, we often have to subtract a constant from a matrix. This is easy to do with an array formula, entered with [Ctrl] + [Shift] + [Enter].

	A	B	C	D	E
1	**SUBTRACTING A CONSTANT FROM A MATRIX**				
2	**Matrix**				
3	1	6		Constant	3
4	2	6			
5	3	8			
6	4	9			
7	5	10			
8					
9	**Matrix minus constant**				
10	-2	3			
11	-1	3	<-- {=A3:B7-E3}		
12	0	5			
13	1	6			
14	2	7			

Creating a Matrix with Ones on the Diagonal and Zeros Elsewhere

This is a problem that comes up in Chapter 10: We want a matrix which has a diagonal of 1's, but has zero off-diagonal elements. The spreadsheet below shows three ways of doing this:

	A	B	C	D	E	F
1	**CREATING A MATRIX OF 1'S AND 0'S We want 1 on diagonal, and 0 elsewhere this comes up in Chapter 10**					
2	**Creating a diagonal matrix of 1's. The cells below contain the array formula =IF(B3:E3=A4:A7,1,0)}**					
3		**A**	**B**	**C**	**D**	
4	**A**	1	0	0	0	
5	**B**	0	1	0	0	
6	**C**	0	0	1	0	
7	**D**	0	0	0	1	
8						
9	**Creating a diagonal matrix of 1's. The cells below contain the array formula =IF(B$10=$A11,1,0)}**					
10		**A**	**B**	**C**	**D**	
11	**A**	1	0	0	0	
12	**B**	0	1	0	0	
13	**C**	0	0	1	0	
14	**D**	0	0	0	1	
15						
16	**Creating a diagonal matrix of 1's when there are no borders. The cells below contain the array formula =IF(ROW()-ROW($B$17)=COLUMN()-COLUMN($B$17),1,0)}**					
17		1	0	0	0	
18		0	1	0	0	
19		0	0	1	0	
20		0	0	0	1	

The first and second methods rely on the labeling of the rows and the columns. In the first example, the formula **=IF(B3:E3=A4:A7,1,0)** tests whether the row label equals the column heading; if this is true, we put a 1 in the cell, and otherwise we put in a 0. In the second example we use an **If** function with mixed absolute and relative references to create the same effect. In the third example, there are no column or row labels, and we rely on the functions **Column** and **Row** to test the equality of the relative row and column places in the matrix.

Finding the Maximum and Minimum Off-Diagonal Elements of a Matrix

We want to find the maximal and minimal elements of the off-diagonal elements of a matrix. Two ways to do this are illustrated below:

	A	B	C	D	E	F
1	**FINDING THE MAX,MIN OF OFF-DIAGONAL ELEMENTS OF A MATRIX the long way**					
2		**The source matrix**				
3		**A**	**B**	**C**	**D**	
4	**A**	10	2	3	4	
5	**B**	-3	20	4	-3	
6	**C**	1	5	60	6	
7	**D**	4	2	-10	25	
8						
9	**RANGE below contains array formula =IF(B3:E3=A4:A7,"",B4:E7)**					
10		**A**	**B**	**C**	**D**	
11	**A**		2	3	4	<-- {=IF(B3:E3=A4:A7,"",B4:E7)}
12	**B**	-3		4	-3	
13	**C**	1	5		6	
14	**D**	4	2	-10		
15						
16	Max of off-diagonals	6	<-- =MAX(B11:E14)			
17	Min of off-diagonals	-10	<-- =MIN(B11:E14)			
18						
19						
20	**Using only non-array formulas**					
21	**Range below contains the non-array formula =IF(B$3=$A4,"",B4)**					
22		**A**	**B**	**C**	**D**	
23	**A**		2	3	4	<-- =IF(E$3=$A4,"",E4)
24	**B**	-3		4	-3	
25	**C**	1	5		6	
26	**D**	4	2	-10		
27						
28	Max of off-diagonals	6	<-- =MAX(B23:E26)			
29	Min of off-diagonals	-10	<-- =MIN(B23:E26)			

In the example above we first use an array function to replace all the diagonal elements with a blank cell. We can then use **Max** and **Min** to determine the extreme off-diagonal elements. As shown in rows 20–29, we could also have used the non-array formula **=IF(B$3=$A4,"",B4)** in cell B11 and copied it to the rest of the matrix.

We can also find the maximum and minimum by incorporating the array formula directly into the **Max** and **Min**:

	A	B	C	D	E	F
1	**FINDING THE MAX,MIN OF OFF-DIAGONAL ELEMENTS OF A MATRIX in one step**					
2	**The source matrix**					
3		**A**	**B**	**C**	**D**	
4	**A**	10	2	3	4	
5	**B**	-3	20	4	-3	
6	**C**	1	5	60	6	
7	**D**	4	2	-10	25	
8						
9	Max of off-diagonals	6	<-- {=MAX(IF(B3:E3=A4:A7,"",B4:E7))}			
10	Min of off-diagonals	-10	<-- {=MIN(IF(B3:E3=A4:A7,"",B4:E7))}			

Replacing the Off-Diagonals Using VLookup

Now suppose that we want to replace the off-diagonal elements using a lookup table, as illustrated below:

	A	B	C	D	E	F
1	**REPLACING THE MAX,MIN OF OFF-DIAGONAL ELEMENTS OF A MATRIX long way, non-array formulas**					
2	**The source matrix**					
3		A	B	C	D	
4	A	10	2	3	4	
5	B	-3	20	4	-3	
6	C	1	5	60	6	
7	D	4	2	-10	25	
8						
9	**Lookup table for replacements**					
10	-10	i				
11	-6	ii				
12	-2	iii				
13	2	iv				
14	6	v				
15	10	vi				
16						
17	**Range below contains non-array formula =IF(B$3=$A4,B4,VLOOKUP(B4,$A$10:$B$15,2)), which has been copied to all the cells**					
18		A	B	C	D	
19	A	10	iv	iv	iv	<-- =IF(E$3=$A4,E4,VLOOKUP(E4,$A$10:$B$15,2))
20	B	ii	20	iv	ii	
21	C	iii	iv	60	v	
22	D	iv	iv	i	25	

Array formulas can simplify this procedure:

	A	B	C	D	E	F
1	**REPLACING THE MAX,MIN OF OFF-DIAGONAL ELEMENTS OF A MATRIX Using array formula**					
2		**The source matrix**				
3		**A**	**B**	**C**	**D**	
4	**A**	10	2	3	4	
5	**B**	-3	20	4	-3	
6	**C**	1	5	60	6	
7	**D**	4	2	-10	25	
8						
9	Lookup table for replacements					
10	-10	i				
11	-6	ii				
12	-2	iii				
13	2	iv				
14	6	v				
15	10	vi				
16						
17	**Range below contains array formula =IF(B3:E3=A4:A7,B4:E7,VLOOKUP(B4:E7,A10:B15,2))**					
18		**A**	**B**	**C**	**D**	
19	**A**	10	iv	iv	iv	<-- {=IF(B3:E3=A4:A7,B4:E7,VLOOKUP(B4:E7,A10:B15,2))}
20	**B**	ii	20	iv	ii	
21	**C**	iii	iv	60	v	
22	**D**	iv	iv	i	25	

Exercises

1. Use a homemade array function to multiply the vector {1,2,3,4,5} times the constant 3.
2. Use the array functions **Transpose** and **MMult** to multiply the row vector {1,2,3,4,5} times the column vector $\begin{Bmatrix} -8 \\ -9 \\ 7 \\ 6 \\ 5 \end{Bmatrix}$.
3. Below you will find the variance-covariance matrix of six stocks. Use an array function to create a matrix with only variances on the diagonal and with zeros elsewhere.

	A	B	C	D	E	F	G
1		**GE**	**MSFT**	**JNJ**	**K**	**BA**	**IBM**
2	**GE**	**0.1035**	0.0758	0.0222	-0.0043	0.0857	0.1414
3	**MSFT**	0.0758	**0.1657**	0.0412	-0.0052	0.0379	0.1400
4	**JNJ**	0.0222	0.0412	**0.0360**	0.0181	0.0101	0.0455
5	**K**	-0.0043	-0.0052	0.0181	**0.0570**	-0.0076	0.0122
6	**BA**	0.0857	0.0379	0.0101	-0.0076	**0.0896**	0.0856
7	**IBM**	0.1414	0.1400	0.0455	0.0122	0.0856	**0.2993**

4. For the problem above: Use an array function to create a matrix with zeros on the diagonal and the covariances off-diagonal.

5. The exercise Excel notebook gives data for three mutual funds. Compute the discrete annual returns for each fund and then use an array function to compute the compound annual return over the period. Recall that discretely compounded, the return in year t is $(Fund\ value_t\ /\ Fund\ value_{t-1}) - 1$. If the returns were continuously compounded, then the year-t return would be $\ln(Fund\ value_t\ /\ Fund\ value_{t-1})$.

35 Some Excel Hints

35.1 Overview

This chapter covers a grab bag of Excel hints dealing with problems and needs that we sometimes run into. The chapter makes no pretence at uniformity or extensiveness of coverage. Topics covered include:

- Fast fills and copy
- Graph titles that change when data changes
- Creating multi-line cells (useful for putting line breaks in cells and linked graph titles)
- Typing Greek symbols
- Typing sub- and superscripts (but not both)
- Naming cells
- Hiding cells
- Formula auditing
- Writing on multiple spreadsheets
- Using Excel's personal notebook to copy and paste and format quickly

35.2 Fast Copy: Filling in Data Next to Filled-In Column

Usually, we copy cells by dragging on the fill handle of the cell with the formula. There is sometimes an easier method. Consider the following situation:

	A	B	C
1	**AUTO FILL/COPY**		
2	1	2	
3	2	5	<-- =B2+3
4	3		
5	4		
6	5		
7	6		
8	7		
9	8		

Now double-click on “fill handle” (shown below with the cross). After double-clicking, the range B2:B9 will automatically fill with the formula in B3.

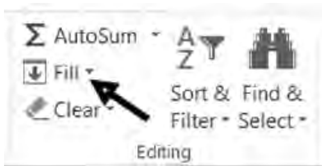
	A	B	C
1	AUTO FILL/COPY		
2	1	2	
3	2	5	<-- =B2+3
4	3		
5	4		
6	5		
7	6		
8	7		
9	8		

Here’s the result:

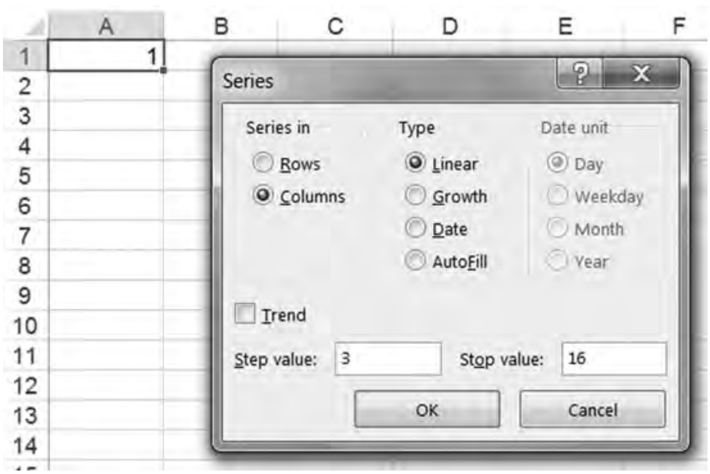
	A	B	C
1	AUTO FILL/COPY		
2	1	2	
3	2	5	<-- =B2+3
4	3	8	
5	4	11	
6	5	14	
7	6	17	
8	7	20	
9	8	23	
10			
11	Double-clicking on the "fill handle" of a cell will fill in the rest of the column provided there's a filled cell next to it.		

35.3 Filling Cells with a Series

Sometimes we want to fill a set of cells with a series. This can be accomplished by going to **Editing|Fill|Series** on the **Home** tab:



Here's an example. Starting with cell A1, we want to fill a column with cells increasing by 3 until we get to 16:



Clicking **OK** gives

	A
1	1
2	4
3	7
4	10
5	13
6	16

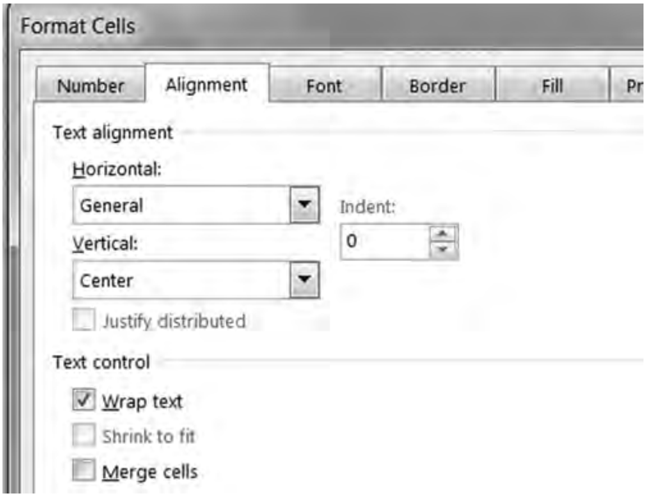
This interesting command has several other options which we leave for you to try.

35.4 Multi-Line Cells

It is sometimes useful to put a line break in a cell, thus creating a multi-line cell. Do this with [Alt] + [Enter] where you want a line break.

	A
1	**PUTTING LINE BREAKS IN CELLS**
2	This is a multi-line cell. The break was entered by inserting [Alt]+[Enter] at the first break point.

There are, of course other ways to make a cell multi-lined. The most obvious is to use the **Wrap text** box in the **Format Cells|Alignment** command:



Here's the cell after we word-wrap it. (Note that in the dialog box above we've also set the vertical alignment of the cell to **Center**.)

	A
1	The line in this cell runs over into neighboring cells, but by using **Home|Number|Alignment** we can word wrap the cell.

35.5 Multi-Line Cells with Text Formulas

Sometimes you want to put a line break in a cell which has text formulas in it. In the example below, the text formula in cell A4 combines the text in cells A1 and A2.

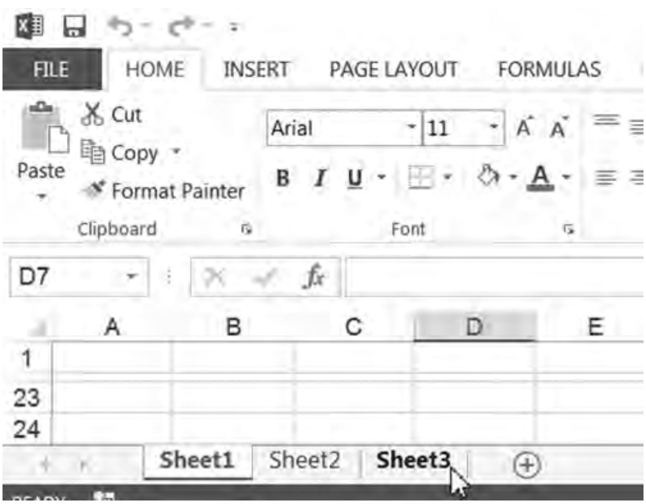
	A	B
1	Simon	
2	Jack	
3	SimonJack	<-- =A1&A2
4	SimonJack	<-- =A1&CHAR(10)&A2, not correctly formatted
5	Simon Jack	<-- =A1&CHAR(10)&A2, pushed the Wrap Text on the Home tab

We can put a line break into the text formula by doing two things:

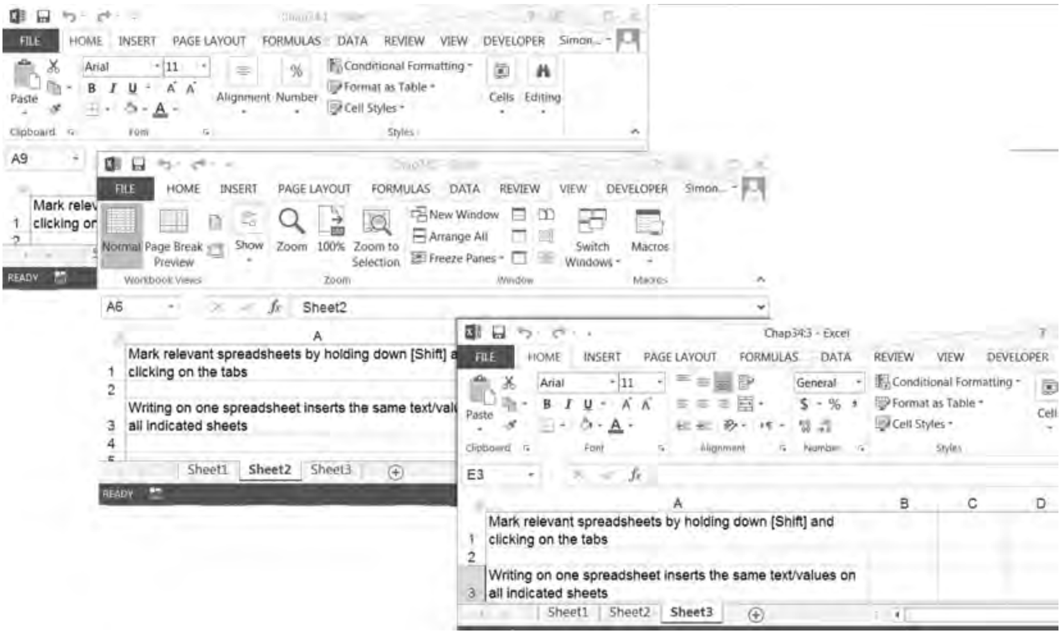
- Put **Char(10)** between A1 and A2—that is, write the formula =**A1&Char(10)&A2** into the cell. **Char(10)** is the code for a hard line return.
- Push **Word Wrap** on the **Home** tab. Now you will have a break between the contents of the two cells at the point you entered **Char(10)**.

35.6 Writing on Multiple Spreadsheets

This Excel trick enables you to write in multiple spreadsheets at the same time. First: Hold down [Shift] and indicate several spreadsheet tabs by clicking on them. In the example below we've clicked on the tabs labeled **Spreadsheet1**, **Spreadsheet2**, **Spreadsheet3**. Now—whatever you write on one sheet will be written on all three.



Now anything we write in one of the sheets is also written in the same cells of all the others, so that we can produce three identical spreadsheets:



35.7 Moving Multiple Sheets of an Excel Notebook

We write on multiple sheets by holding down [Shift] and marking the tabs of the relevant spreadsheets. A similar trick works to move multiple sheets of the same Excel notebook:

- Mark the multiple sheets by holding down [Shift] and clicking on the appropriate sheets.
- Now use **Edit|Move** or **copy sheet** to move or copy the sheets to another location on the same spreadsheet or to a different spreadsheet.

35.8 Text Functions in Excel

The **Text** function lets you change numbers to text. Here are some examples:

	A	B	C
1	**TEXT FUNCTIONS**		
2	Income	15,000	
3	Tax rate	35%	
4	Taxes owed	5,250	<-- =B3*B2
5			
6			
7	Tax rate as text	35.00%	<-- =TEXT(B3,"0.00%")
8		0.4	<-- =TEXT(B3,"0.0")
9			
10	Income as date	Jan. 24, 1941	<-- =TEXT(B2,"mmm. dd, yyyy")

Note that you can choose different ways of formatting the cell B3 in text form—in cell B7 we have formatted the tax rate as a percentage with two decimal points, whereas in cell B8 we have formatted the tax rate as one decimal, causing it to be rounded off.

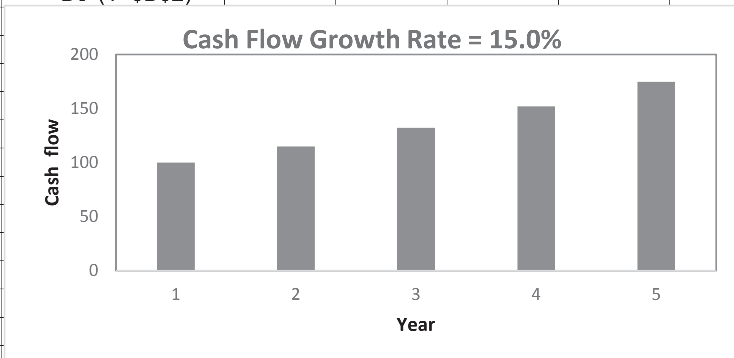
Note also the somewhat stupid example in cell B10: Since dates in Excel are just numbers that express the number of days from 1 January 1900, we can express the income of $15,000 in cell B2 as a date.

In the next section we use text functions to create chart titles that update themselves.

35.9 Chart Titles That Update

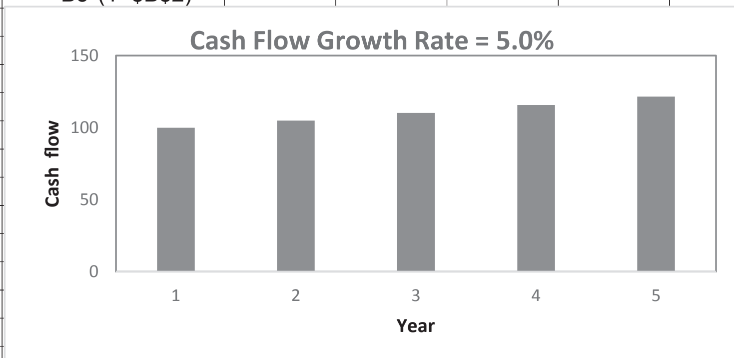
You want to have the chart title change when a parameter on the spreadsheet changes. For example, in the next spreadsheet, you want the chart title to indicate the growth rate.

	A	B	C	D	E	F	G	H	I
1	**CHART TITLES THAT UPDATE AUTOMATICALLY**								
2	Growth	15%							
3									
4	Year	Cash flow							
5	1	100.00							
6	2	115.00	<-- =B5*(1+$B$2)						
7	3	132.25							
8	4	152.09							
9	5	174.90							
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20			Chart title below contains the text function ="Cash Flow Growth Rate = "&TEXT(B2,"0.0%")						
21									
22			Cash Flow Growth Rate = 15.0%						



Once we have completed the necessary steps, changing the growth rate will change both the graph *and* its title:

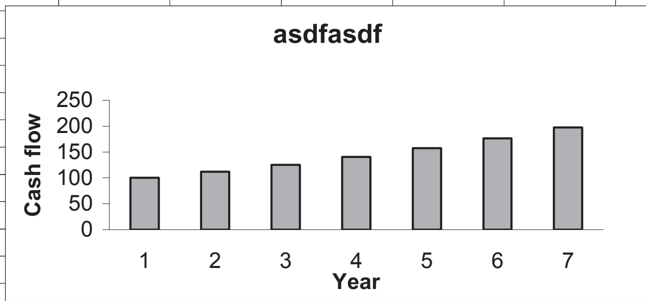
	A	B	C	D	E	F	G	H	I
1	CHART TITLES THAT UPDATE AUTOMATICALLY								
2	Growth	5%							
3									
4	Year	Cash flow							
5	1	100.00							
6	2	105.00	<-- =B5*(1+$B$2)						
7	3	110.25							
8	4	115.76							
9	5	121.55							
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									



To make graph titles update automatically, carry out the following steps:

- Create the graph you want in the format you want it. Give the graph a "proxy title." (It makes no difference what; you're going to eliminate it soon.) At this stage your graph might look like:

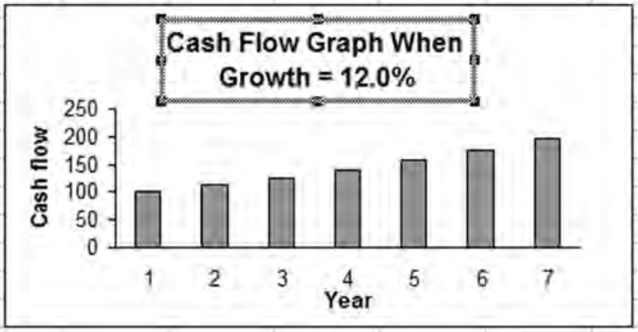
	A	B	C	D	E	F	G	H	I
1	**CHART TITLES THAT UPDATE AUTOMATICALLY**								
2	Growth	12%							
3									
4	Year	Cash flow							
5	1	100.00							
6	2	112.00	<-- =B5*(1+$B$2)						
7	3	125.44							
8	4	140.49							
9	5	157.35							
10	6	176.23							
11	7	197.38							
12									
13									
14									
15									
16									
17									
18									
19									
20				**Chart title below contains text function ="Cash Flow Graph When Growth = "&TEXT(B2,"0.0%")**					
21									
22				Cash Flow Graph When Growth = 12.0%					



- Create the title you want in a cell. In the example above, cell D22 contains the formula: =“Cash Flow Graph When Growth = “**&TEXT(B2,”0.0%**”)
- Click on the graph title to mark it, and then go to the formula bar and insert an equal sign to indicate a formula. Then **point** at cell D22 with the formula and click [Enter]. In the picture below, you see the chart title highlighted and in the formula bar “=Changing graph titles!$D$22” indicating the title of the graph.

Chart Title | =’chart titles, step2’!$D$22

	A	B	C	D	E	F	G	H	I
1			CHART TITLES THAT UPDATE AUTOMATICALLY						
2	Growth	12%							
3									
4	Year	Cash flow							
5	1	100.00							
6	2	112.00	<-- =B5*(1+$B$2)						
7	3	125.44							
8	4	140.49							
9	5	157.35							
10	6	176.23							
11	7	197.38							
12									
13									
14									
15									
16									
17									
18									
19									
20				Chart title below contains text function ="Cash Flow Graph When Growth = "&TEXT(B2,"0.0%")					
21									
22				Cash Flow Graph When Growth = 12.0%					

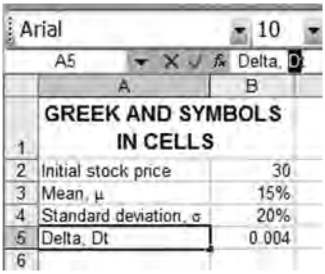


35.10 Putting Greek Symbols in Cells

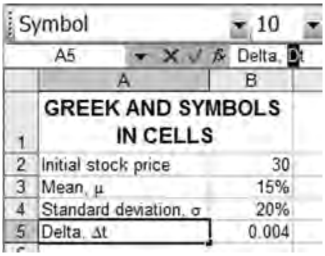
How do we type Greek letters in a spreadsheet?

	A	B
1	GREEK AND SYMBOLS IN CELLS	
2	Initial stock price	30
3	Mean, μ	15%
4	Standard deviation, σ	20%
5	Delta, Δt	0.004

This is fairly simple, if you know the Greek equivalents for the Greek letters (for example, μ and σ are lowercase “m” and “s,” respectively, Σ and Δ are uppercase “S” and “D”). For example, we first typed “Delta, Dt” into cell A5 and then marked the “D” in the formula bar:



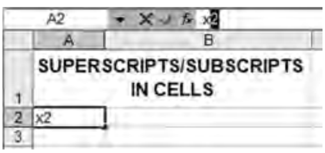
We then changed the font from Arial to Symbol:



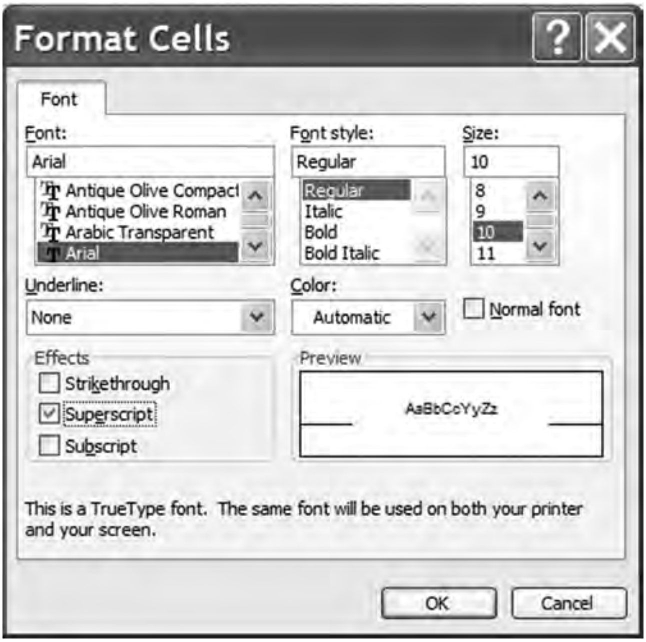
Pressing [Enter] produces the desired result.

35.11 Superscripts and Subscripts

It is not a problem to type subscripts or superscripts in Excel. Enter text into a cell, and then mark the letters you want to turn into a subscript or superscript:



Now go to **Format Cells** and check the **Superscript** box:



Here's the result:

	A	B
1	SUPERSCRIPTS/SUBSCRIPTS IN CELLS	
2	x^2	
3		
4	$x^{2}{}_{i}$	Cannot put superscript and subscript one above the other

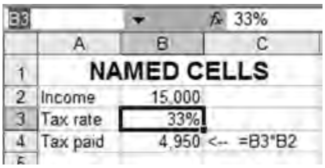
As you can see in cell A4 above, you cannot put a subscript and a superscript on the same letter. That is, you cannot create x_i^2.

35.12 Named Cells

It is sometimes useful to give a name to a cell. Here's an example:

	A	B	C
1	NAMED CELLS		
2	Income	15,000	
3	Tax rate	33%	
4	Tax paid	4,950	<-- =B3*B2

We want to refer to cell B3 by the name "tax." To do this, we mark the cell and then go to the name tab on the toolbar:



B3		f_x 33%	
	A	B	C
1	NAMED CELLS		
2	Income	15,000	
3	Tax rate	33%	
4	Tax paid	4,950	<-- =B3*B2

Typing in the word "tax" on the highlighted B3 allows us to reference B3 by this name anywhere in the Excel notebook:

	A	B	C
1	NAMED CELLS		
2	Income	15,000	
3	Tax rate	33%	
4	Tax paid	4,950	<-- =tax*B2

Sometimes Excel lets us use cell names without ever actually going through the procedure just described. In the next example, Excel lets us use the column headers as cell names:

	A	B	C	D
7	**Sales**	**Margin**	**Profit**	
8	1000	20%	200	<-- =Sales*Margin
9	5000	30%	1500	<-- =Sales*Margin

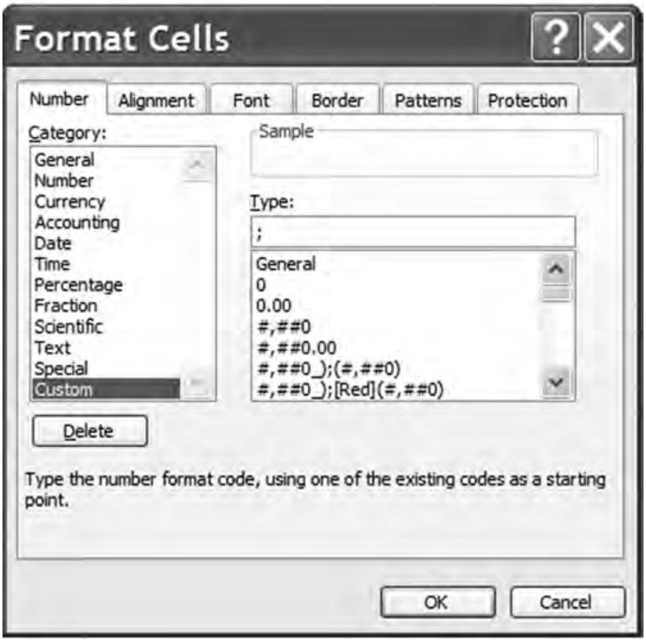
To manage the named cells, go to the **Name Manager** on the **Formulas** tab:

35.13 Hiding Cells (in Data Tables and Other Places)

In this text, we have often hidden the cell contents of data table headers. Here's a simple data table (this topic is discussed in detail in Chapter 31):

	A	B	C	D
1	**HIDING CELLS**			
2	Payment	100		
3	Number of payments	15		
4	Discount rate	15%		
5	Present value	$584.74	<-- =PV(B4,B3,-B2)	
6				
7			**PV of payments**	
8	**Data table**		584.74	<-- =B5 , data table header
9		0%	1,500.00	
10		3%	1,193.79	
11		6%	971.22	
12		9%	806.07	
13		12%	681.09	
14		15%	584.74	
15		18%	509.16	
16		21%	448.90	

The data table header in cell C8 is necessary for the table to work, but it is ugly and may be confusing if the table is copied into other documents. To hide the contents of C8, mark the cell and go to the **Format Cells** menu (or click on the right mouse button):



In the **Number|Custom|Type** box we have put in a semicolon. This preserves the cell contents but prevents them from being seen. Now when you copy the cells, this is the way they will appear:

	B	C	D
7		**PV of payments**	
8			<-- =B5 , data table header
9	0%	1,500.00	
10	3%	1,193.79	
11	6%	971.22	
12	9%	806.07	
13	12%	681.09	
14	15%	584.74	
15	18%	509.16	
16	21%	448.90	

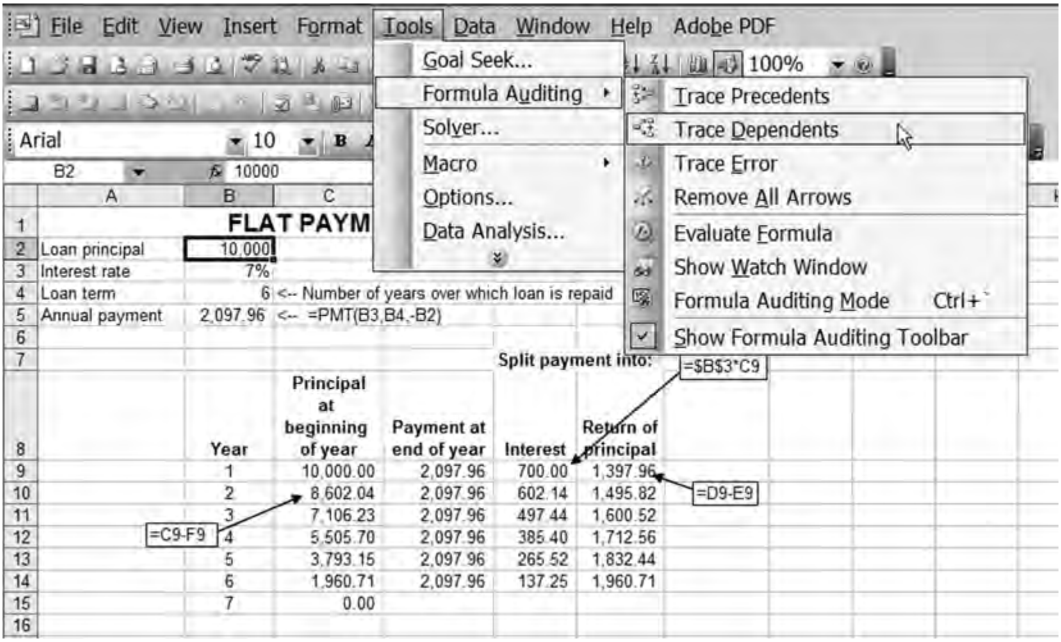
Note the comment in cell D8: We advise you always to *annotate* your spreadsheet, so that when you come back to it after a few weeks/months, you will know that cell C8 really does have something in it!

One final note: To hide a cell which contains a reference to another cell containing a formula, three semicolons (;;;) may be necessary. In the spreadsheet below, cell B4 contains the function **IF**. Cell B6 refers to this cell. To hide B6, we use three semicolons instead of one in the **Format Cells|Number|Custom|Type.** (If there's logic here, it escapes us!)

	A	B	C
1	**HIDING A CELL WHICH REFERS TO A FORMULA**		
2	a	33	
3	b	8	
4	c	bbb	<-- =IF(B2+B3<15,"aaa","bbb")
5			
6	Cell to be hidden -->		<-- =B4

35.14 Formula Auditing

Excel can tell you where you've used a cell in your formulas and which cells a particular formula depends on. Clicking on **Tools|Formula Auditing** brings up a menu which allows you to do this:



Here's the result:

	A	B	C	D	E	F	G
1	**FLAT PAYMENT SCHEDULES**						
2	Loan principal	10,000					
3	Interest rate	7%					
4	Loan term	6	<-- Number of years over which loan is repaid				
5	Annual payment	2,097.96	<-- =PMT(B3,B4,-B2)				
6							
7					**Split payment into:**		=$B$3*C9
8		**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest**	**Return of principal**	
9		1	10,000.00	2,097.96	700.00	1,397.96	
10		2	8,602.04	2,097.96	602.14	1,495.82	=D9-E9
11		3	7,106.23	2,097.96	497.44	1,600.52	
12	=C9-F9	4	5,505.70	2,097.96	385.40	1,712.56	
13		5	3,793.15	2,097.96	265.52	1,832.44	
14		6	1,960.71	2,097.96	137.25	1,960.71	
15		7	0.00				

In a similar way we can check to see which cells are precedents of a particular cell:

	A	B	C	D	E	F	G
1	**FLAT PAYMENT SCHEDULES**						
2	Loan principal	10,000					
3	Interest rate	7%					
4	Loan term	6	<-- Number of years over which loan is repaid				
5	Annual payment	2,097.96	<-- =PMT(B3,B4,-B2)				
6							
7					**Split payment into:**		=$B$3*C9
8		**Year**	**Principal at beginning of year**	**Payment at end of year**	**Interest**	**Return of principal**	
9		1	10,000.00	2,097.96	700.00	1,397.96	
10		2	8,602.04	2,097.96	602.14	1,495.82	=D9-E9
11		3	7,106.23	2,097.96	497.44	1,600.52	
12	=C9-F9	4	5,505.70	2,097.96	385.40	1,712.56	
13		5	3,793.15	2,097.96	265.52	1,832.44	
14		6	1,960.71	2,097.96	137.25	1,960.71	
15		7	0.00				

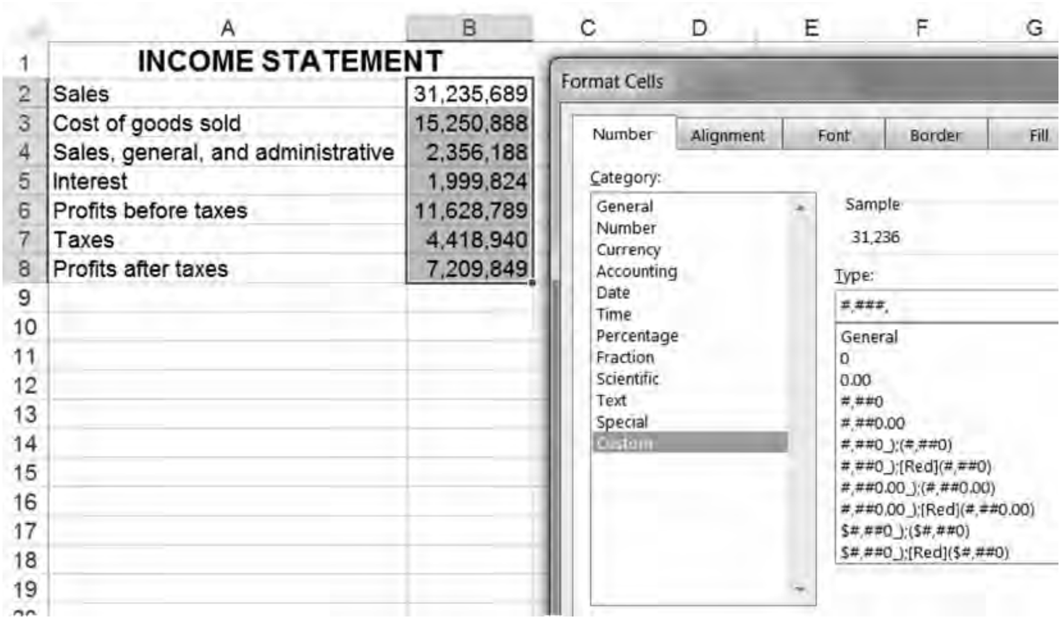
Formula auditing can help you implement a general rule of good spreadsheet writing: You should try to avoid cells which don't have either precedents or dependents.

35.15 Formatting Millions as Thousands

By using **Format Cells|Custom** you can change millions into thousands. To see where this is handy, consider the following income statement:

	A	B
1	**INCOME STATEMENT**	
2	Sales	31,235,689
3	Cost of goods sold	15,250,888
4	Sales, general, and administrative	2,356,188
5	Interest	1,999,824
6	Profits before taxes	11,628,789
7	Taxes	4,418,940
8	Profits after taxes	7,209,849

We want to make the income statement appear in thousands (in other words, instead of 31,235,689 we will see 31,236. Here's how this can be done:



In the **Type** box we have indicated **#,###,**. The comma at the end indicates that we want Excel to drop the last three digits in the number (and round the number), and the **#,###** indicates that we want the remaining numbers to appear with a comma. This is merely a formatting change—the actual numbers are not changed: In cell B10 in the output below, we've multiplied the Sales by 2; the result is 62,471,378.

Adding another comma to the **Type** box (that is, **#,###,,**) will drop another three digits.

	A	B	C
1	INCOME STATEMENT		
2	Sales	31,236	
3	Cost of goods sold	15,251	
4	Sales, general, and administrative	2,356	
5	Interest	2,000	
6	Profits before taxes	11,629	
7	Taxes	4,419	
8	Profits after taxes	7,210	
9			
10	The cells retain their values	62,471,378	<-- =B2*2

35.16 Excel's Personal Notebook: Automating Frequent Procedures

Excel's personal notebook allows you to save macros and procedures that only you can access. We give two examples of such procedures:

- We explain Excel's **Copy as Picture** feature and show how to attach this to a macro stored in your personal notebook. This greatly simplifies copying from Excel and pasting into Microsoft Word (all the copy/pastes in this book were made this way).
- We explain how to save number formatting in the personal notebook.

Using the Copy as Picture Feature of Excel[1]

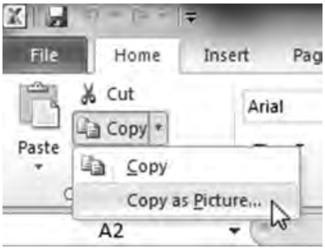
Excel 2010 and 2013 contain a very nice way to copy from Excel as a picture. This is useful for embedding pictures of Excel spreadsheets into Word without a link. Here's the way it works:

1. In Excel indicate the selection you want to copy:

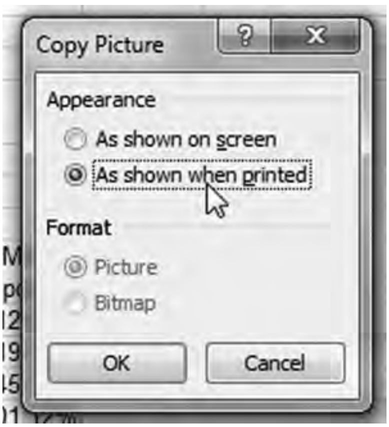
	A	B	C	D	E	F	G
1							
2	Variance-covariance matrix					Means	
3	0.2000	-0.0200	0.0250	-0.0080		3%	
4	-0.0200	0.3000	0.0600	0.0030		2%	
5	0.0250	0.0600	0.4000	0.0000		8%	
6	-0.0080	0.0030	0.0000	0.5000		4%	
7							
8	Constant,	-1%					
9							

1. This section applies to Excel 2010 and 2013, but not to previous versions.

2. On the **Home** tab go to **Copy|Copy as Picture:**



3. Indicate **As shown when printed**:



4. You can now go to Microsoft Word and Copy and Paste. The result: You get a picture without any Excel link.

	A	B	C	D	E	F
2	Variance-covariance matrix					Means
3	0.2000	-0.0200	0.0250	-0.0080		3%
4	-0.0200	0.3000	0.0600	0.0030		2%
5	0.0250	0.0600	0.4000	0.0000		8%
6	-0.0080	0.0030	0.0000	0.5000		4%

Automating the Procedure

We want to automate this procedure:

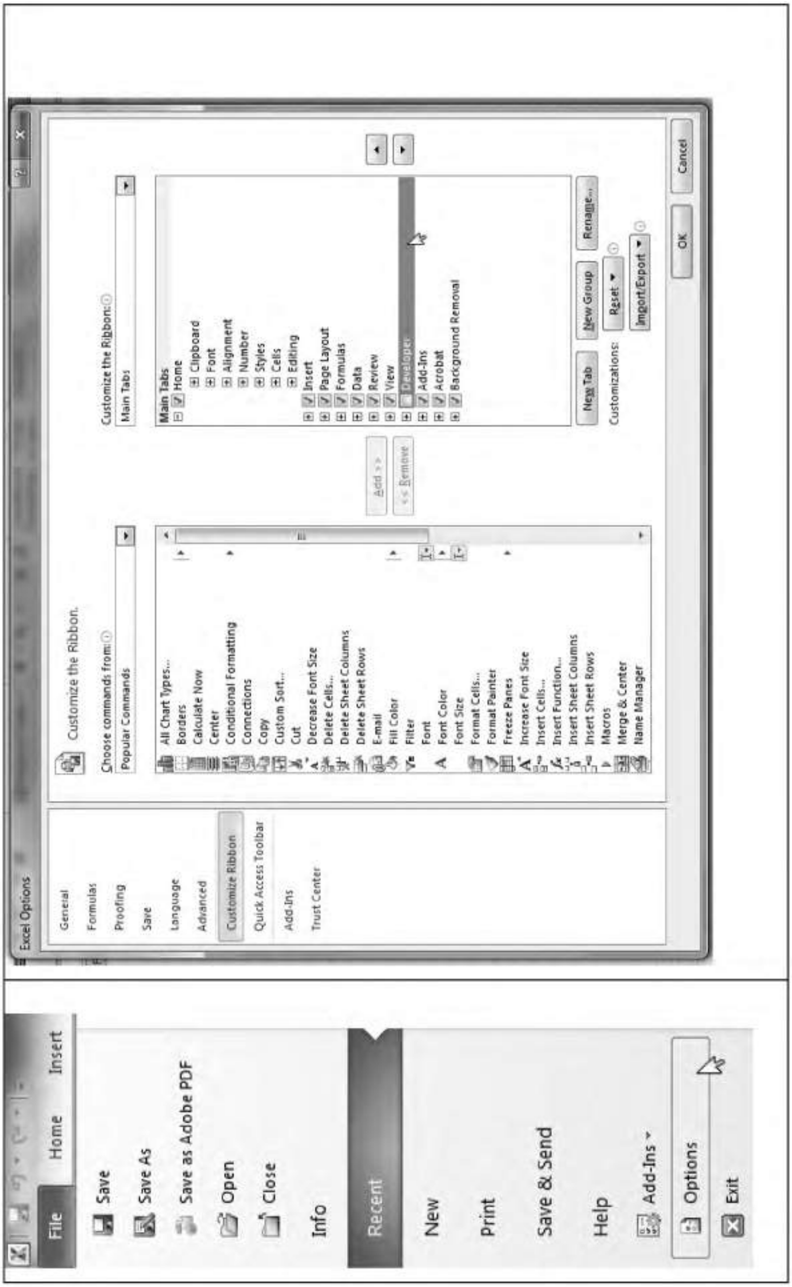
- Turn it into a macro.
- Attach a key sequence (in this case, [Ctrl] + q) to the macro.
- Make the macro and key sequence available in all your Excel spreadsheets.

To do this, you have to create a **Personal.xlsb** file. This file is hidden but activates each time you start Excel. It's yours only—other readers of your spreadsheets won't see it. Here are the steps:

- Activate the **Developer** tab on the menu bar.
- Use **Record Macro** to save a macro as a personal notebook.
- Edit the personal notebook with what you want.

Activate the Developer Tab

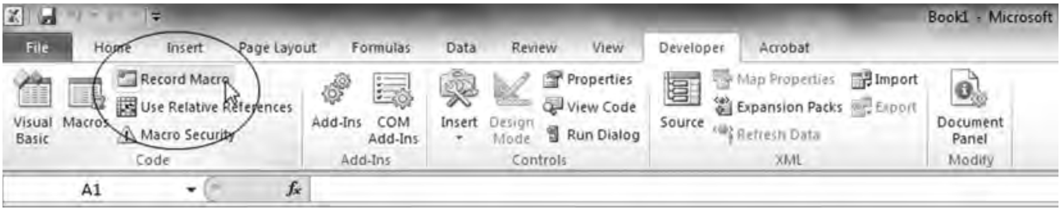
Go to **File|Options|Customize Ribbon** and activate the **Developer** tab as shown below:


File
Home
Insert
Save
Save As
Save as Adobe PDF
Open
Close
Info
Recent
New
Print
Save & Send
Help
Add-Ins
Options
Exit
Excel Options
General
Formulas
Proofing
Save
Language
Advanced
Customize Ribbon
Quick Access Toolbar
Add-Ins
Trust Center
Customize the Ribbon.
Choose commands from:
Popular Commands
All Chart Types...
Borders
Calculate Now
Center
Conditional Formatting
Connections
Copy
Custom Sort...
Cut
Decrease Font Size
Delete Cells...
Delete Sheet Columns
Delete Sheet Rows
E-mail
Fill Color
Filter
Font
Font Color
Font Size
Format Cells...
Format Painter
Freeze Panes
Increase Font Size
Insert Cells...
Insert Function...
Insert Sheet Columns
Insert Sheet Rows
Macros
Merge & Center
Name Manager
Add >>
<< Remove
Customize the Ribbon:
Main Tabs
Main Tabs
Home
Clipboard
Font
Alignment
Number
Styles
Cells
Editing
Insert
Page Layout
Formulas
Data
Review
View
Developer
Add-Ins
Acrobat
Background Removal
New Tab
New Group
Rename...
Customizations:
Reset
Import/Export
OK
Cancel

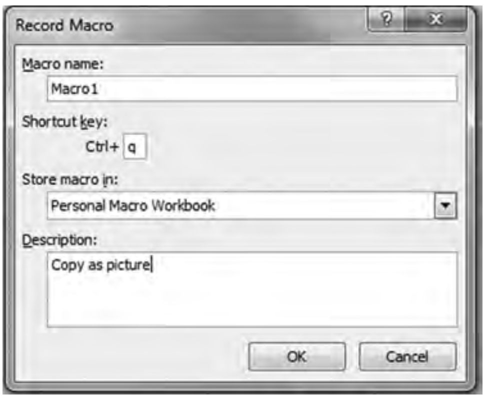
Use Record Macro

The **Developer** tab allows you to record a macro and save it as part of the **Personal.xlsb** notebook. We will illustrate with the copy as picture feature.

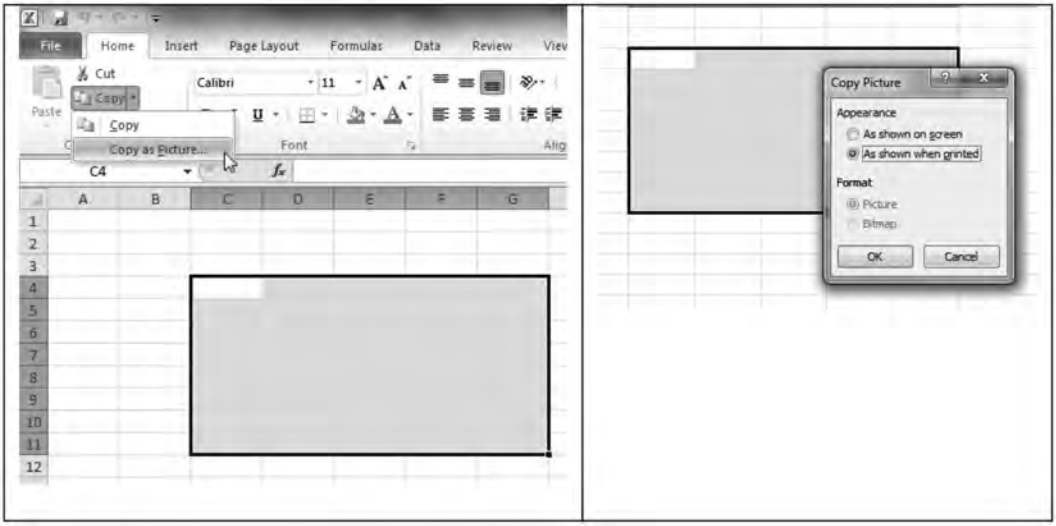
1. Open a blank Excel notebook and click on the **Developer** tab and then on **Record Macro**:



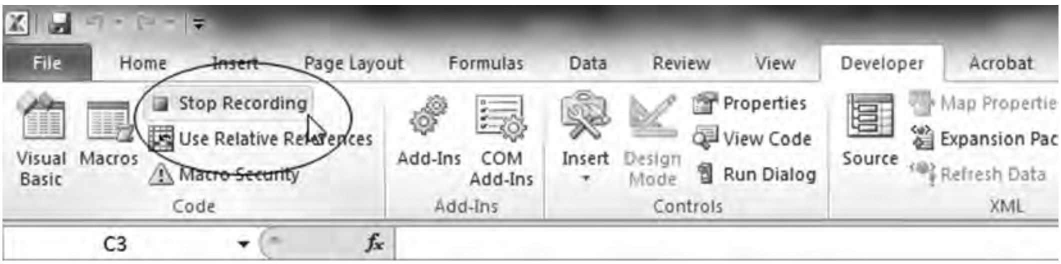
Excel will ask for details of the recording. Here's what I wrote. Note that I'm saving this as a **Personal Macro Workbook** and that I'm using the shortcut keystrokes [Ctrl] + q:



2. Now go to the **Home** tab, mark an area of the spreadsheet, and go through the whole **Copy as Picture** feature:



3. Go back to the **Developer** tab and stop the recording:



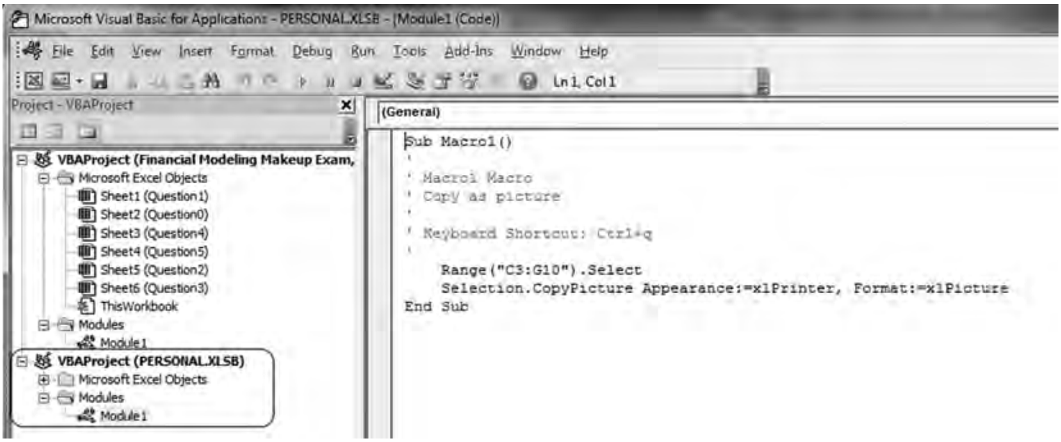
4. Close down Excel. Excel will ask you if you want to save the Personal notebook. The answer is, of course, positive:

This creates the following file (“simon benninga” is of course my user name on my computer—you will substitute your user name):

C:\Users\simonbenninga\AppData\Roaming\Microsoft\Excel\XLSTART\PERSONAL.XLSB

Edit the Personal Notebook

Now that you’ve saved the **Personal.xlsb** file, you may want to edit it. Open any Excel file, hit [Alt] + [F11] to go the VBA editor. Note that the Personal file is also there:



We have edited this by deleting the line **Range(“C3:G10”).Select**. The resulting macro is given below.

```
Range(“C3:G10”):
Sub Macro1()
‘
‘ Macro1 Macro
‘ Copy as picture
‘
‘ Keyboard Shortcut: Ctrl+q
    Selection.CopyPicture Appearance:=xlPrinter, _
    Format:=xlPicture
End Sub
```

Using the Macro

From now on, whenever you open a file on *your computer*, you can use [Ctrl] + q to copy a region as a picture. Note that this feature works only on your computer—it's in your personal notebook.

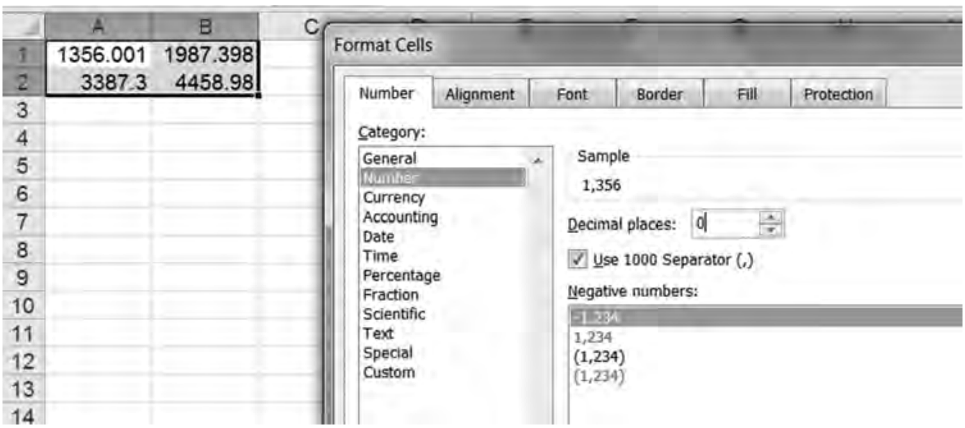
Quick Number Formatting

We often want to format numbers with comma separators and without decimals. For example:

	A	B
1	1356.001	1987.398
2	3387.3	4458.98

To make the formatting of these numbers uniform:

- Mark the numbers.
- Go to the appropriate formatting command and then press [Enter].



The result:

	A	B
1	1,356	1,987
2	3,387	4,459

By using the same routine as for Copy and Paste as Picture above, we can generate a macro in our personal notebook. We have assigned [Ctrl] + w to this macro:

```
Sub Commas()
' Commas Macro
' Keyboard Shortcut: Ctrl+w
    Selection.NumberFormat = "#,##0"
End Sub
```

VII VISUAL BASIC FOR APPLICATIONS (VBA)

Visual Basic for Applications (VBA) is the programming language attached to Excel. VBA is very functional and flexible. Because of its ready integration with Excel worksheets, VBA is widely used in the financial community. VBA incorporates many features that are part of standard programming languages, and it is not difficult to master if you have some programming experience.

You do not need to be proficient in VBA to understand Sections I–VI of *Financial Modeling*. These sections can be understood without anything more than the very rudimentary VBA principles incorporated in the preface to this book (or alternatively in the small file called "Adding Getformula to your Spreadsheet" that is part of the disk that comes with the book).

The four chapters of this section cover Visual Basic for Applications (VBA) topics for the reader interested in developing his or her own programs. Chapter 36 shows how to write functions that can be added in to Excel spreadsheets. *Financial Modeling* uses many of these "homemade" functions. Examples are the two-stage Gordon model (Chapter 3), Black-Scholes pricing of options (Chapter 17), and derivation of the Nelson-Siegel term structure (Chapter 22).

Chapter 37 discusses more advanced topics related to variables and arrays in VBA. We have used this topic in fixing the bugs in Excel's **XNPV** and **XIRR** functions (Chapter 1). Chapter 38 shows how to build subroutines in VBA. A subroutine is not a function, but rather an automation of some repetitive action. *Financial Modeling* uses subroutines in a number of places—for example, in computing the efficient frontier without short sales (Chapter 12).

Finally, Chapter 39 discusses objects and add-ins. Among other topics discussed in this chapter is the creation of user-defined add-ins in Excel.

36 User-Defined Functions with VBA

36.1 Overview

Chapters 36–39 discuss the uses of Excel's programming language, Visual Basic for Applications (VBA). VBA provides a complete programming language and environment fully integrated with Excel and all other Microsoft Office applications. In this chapter we introduce user-defined functions, which are used in various places in this book.

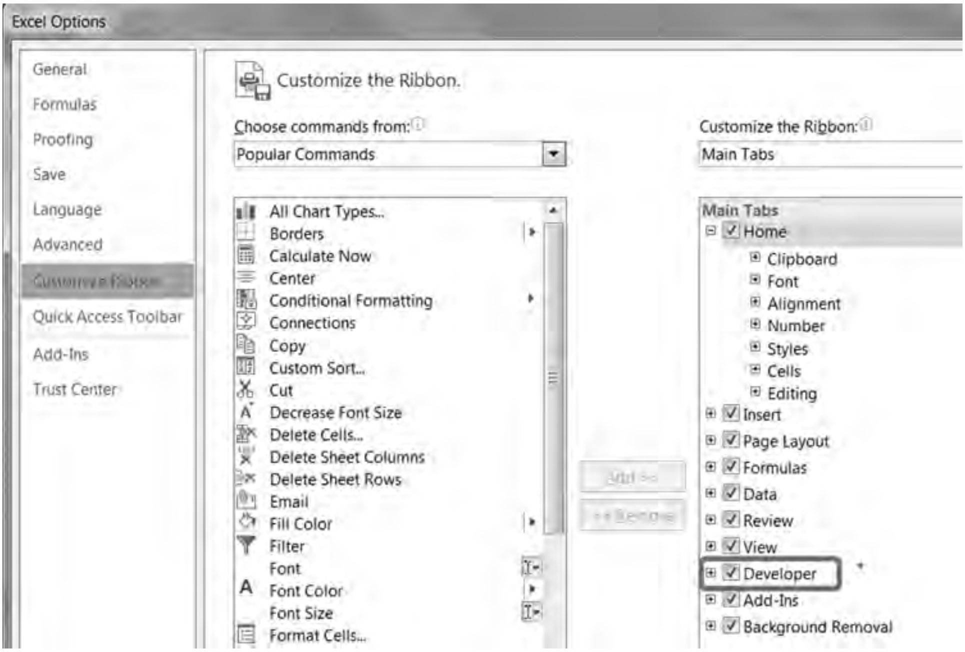
The examples and screen shots depict the Excel 2013 working environment but are fully compatible (unless otherwise noted) with all versions of Excel using Visual Basic for Applications (Version 5 and above).

36.2 Using the VBA Editor to Build a User-Defined Function

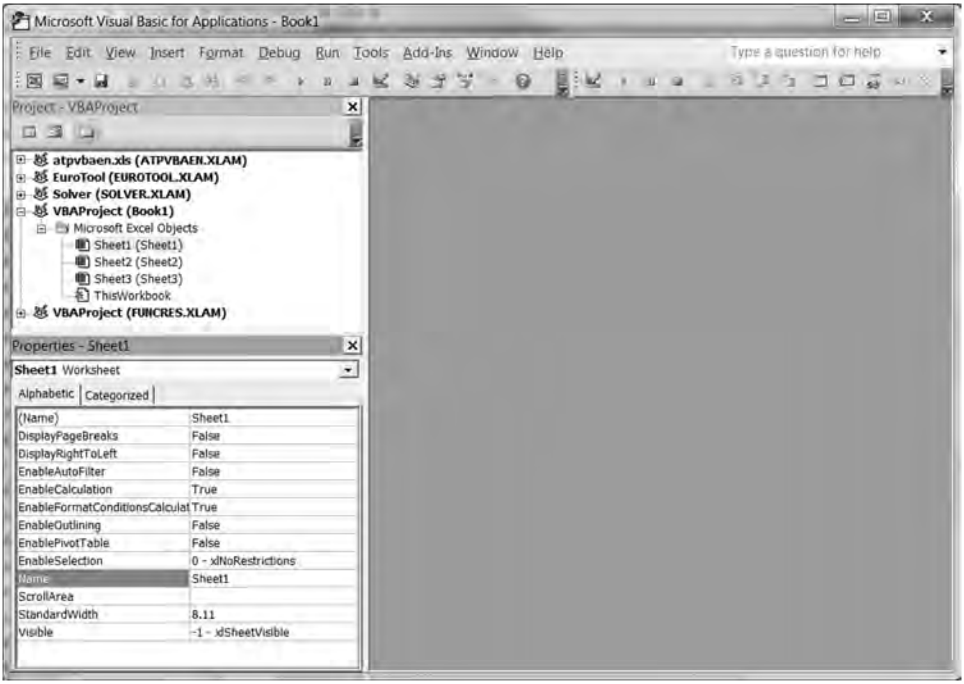
Throughout this book we have used VBA to define functions that are not included in Excel. One example is the function **Getformula** that is attached to all the spreadsheets in this book; another example is the function that computes the Black-Scholes option value (Chapter 17). In this section we show you how to build a user-defined function. A user function is a saved list of instructions for Excel that produces a value. Once defined, a user function can be used inside an Excel worksheet like any other function.[1]

1. User-defined functions are usually attached to a specific workbook and are only available if that workbook is currently open in Excel. If you want a macro to be available whenever you use Excel on a specific computer save it in the **Personal Macro Workbook**; see Chapter 35, section16. Another way of having access to a VBA function across worksheets is to put it in an add-in; see Chapter 39 for an introduction to add-ins in Excel.

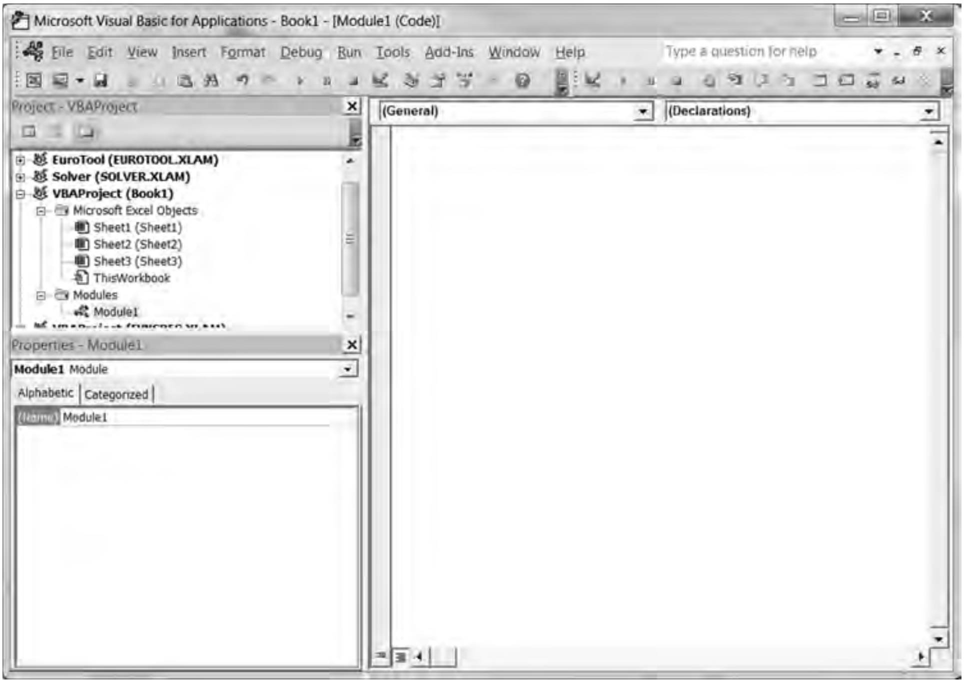
In this section we will write our first user-defined function. Before you can do this, you need to activate the VBA editor. You can do this either by using the keyboard shortcut [Alt] + F11 or from the Excel ribbon (**Developer Tab|Visual Basic Editor**). By default, Excel doesn't display the **Developer** tab on the Excel ribbon. To show the **Developer** tab, go to **File|Options|Customize the Ribbon** and indicate **Developer**:



The result in both cases is a new window like the following screen shot (your window may look slightly different, but it will be functionally equivalent).



A user-defined function needs to be written in a module. To open a new module, select **Insert|Module** from the menu in the VBA editor environment. This will open a new window, as illustrated in the next screen shot:



We are now ready to write our first function. This function (named "**plus**") will add together two numbers.

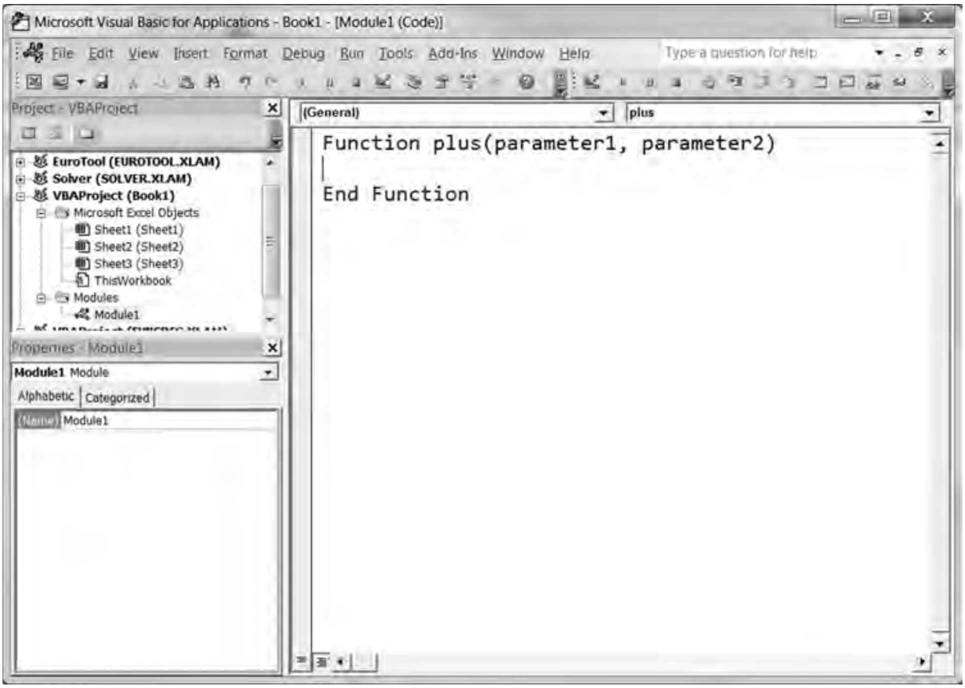
A user-defined function in Excel has three obligatory elements:

1. A header line with the name of the function and a list of parameters.
2. A closing line (usually inserted by VBA).
3. Some program lines between the header and the closing line.

Start writing the first line of the function:

```
function plus (parameter1,parameter2)
```

As soon as you end the line with a tap on the Enter key, VBA will do a cleanup job. The color of all the words that VBA recognizes as part of its programming language ("reserved words") will change. All reserved words will be capitalized. A space will be added after the comma separating the first parameter from the second parameter. The closing line for the function will be inserted, and the cursor will be in position between the header and the closing line ready for you to go on typing.



We are now ready to type our function line. This is the line that makes our function do something.[2] Our first function will take two variables and return their sum:

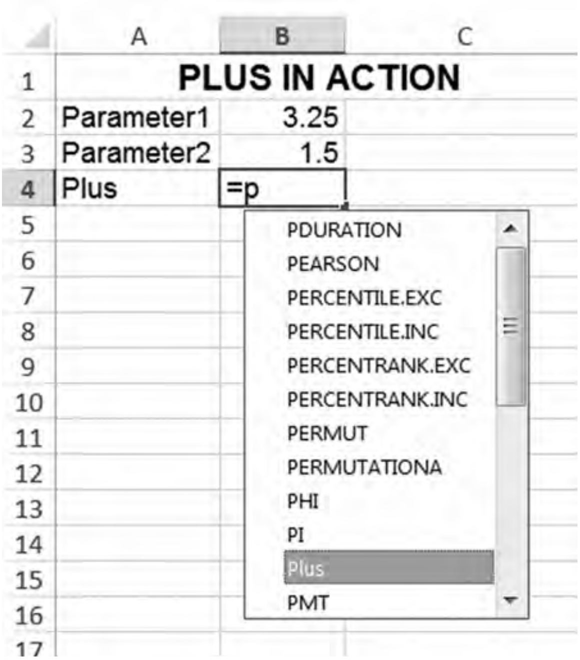
```
Function plus(parameter1, parameter2)
   plus = parameter1 + parameter2
End Function
```

You can now use this function in your spreadsheet:

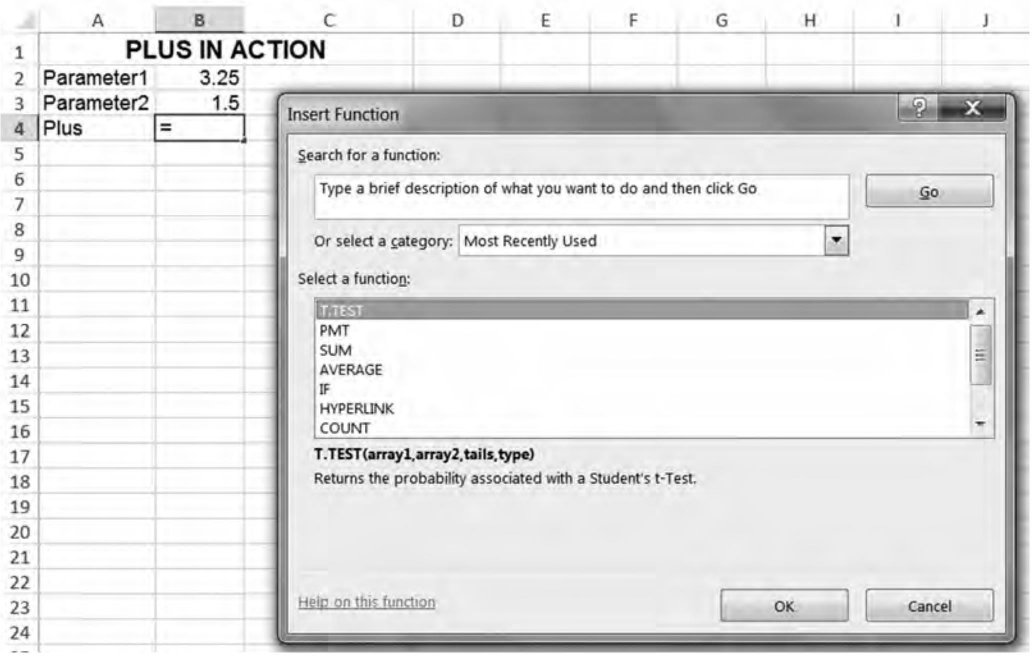
	A	B	C
1	**PLUS IN ACTION**		
2	Parameter1	3.25	
3	Parameter2	1.5	
4	Plus	4.75	<-- =plus(B2,B3)

2. The indentation of lines in VBA code, which we added manually, is not required by VBA but makes reading the code much easier.

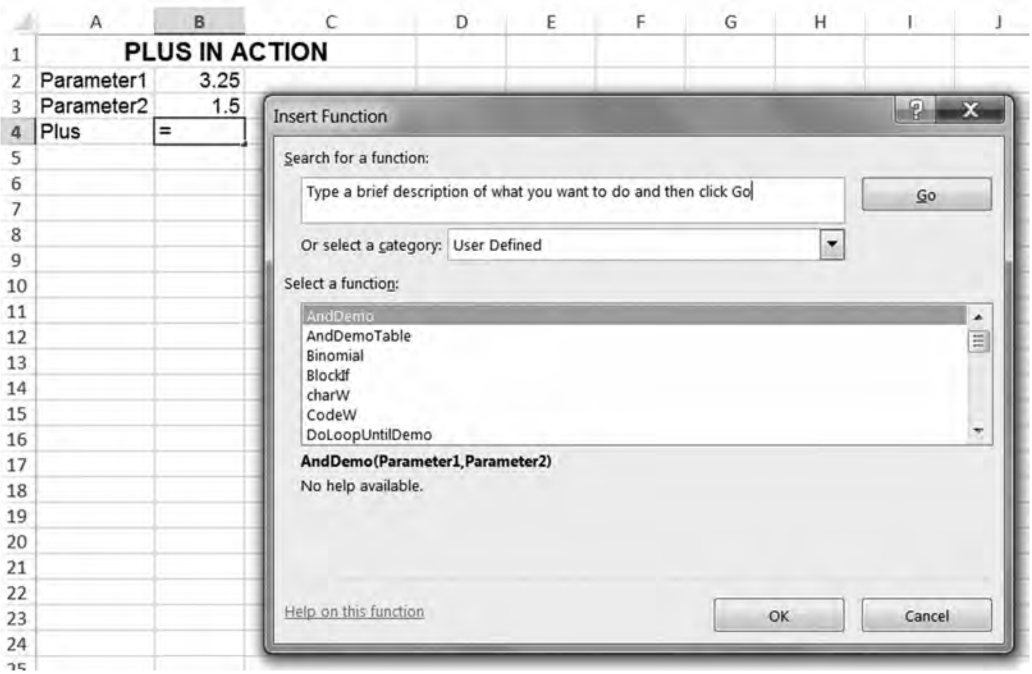
The fastest way to insert a function (assuming you know its name) is to start typing the name. When the suggested list of names narrows down, select the appropriate function name from the list:



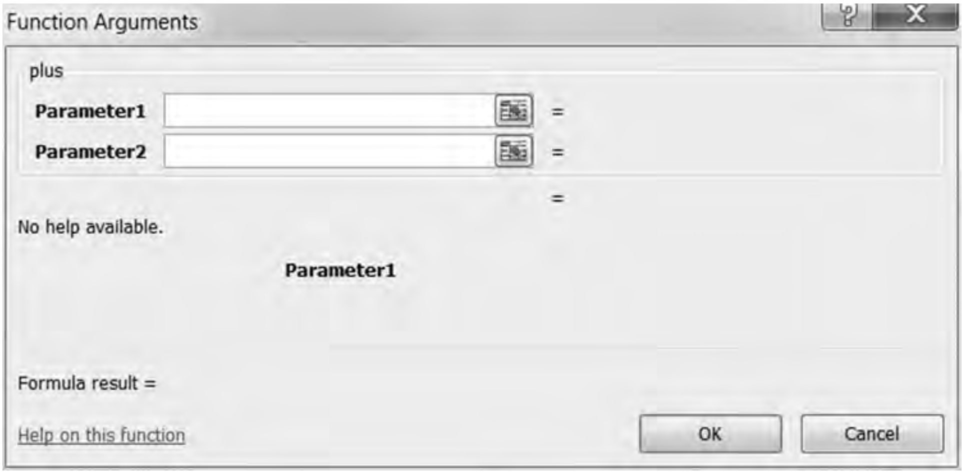
You can also use the function in the Excel Function Wizard. Clicking on this  icon on the toolbar will produce the following screen:



Selecting **User Defined** from the pull-down menu will present the following screen listing all user-defined functions; one of them should be the function we have just added, **plus**:



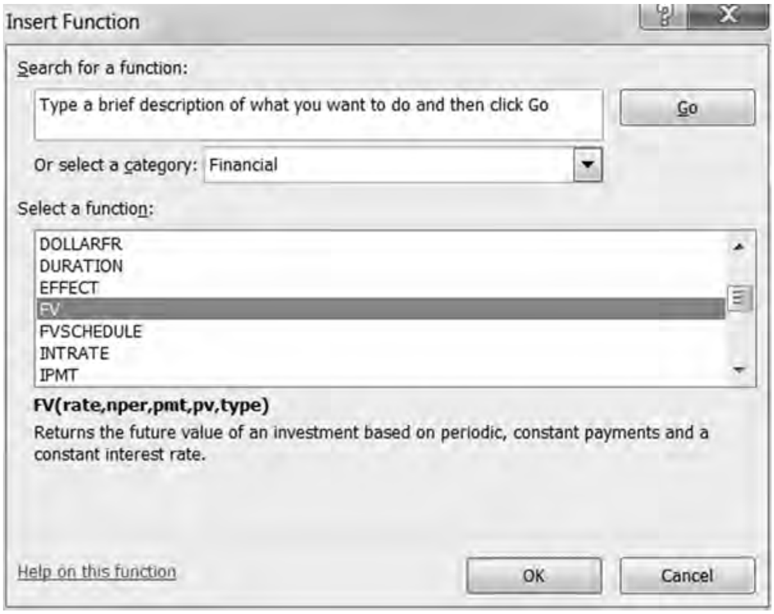
When you select **plus** and click OK, you will see that Excel treats this like any other function, bringing up a dialogue box that asks for the location or value of **parameter1** and **parameter2**:



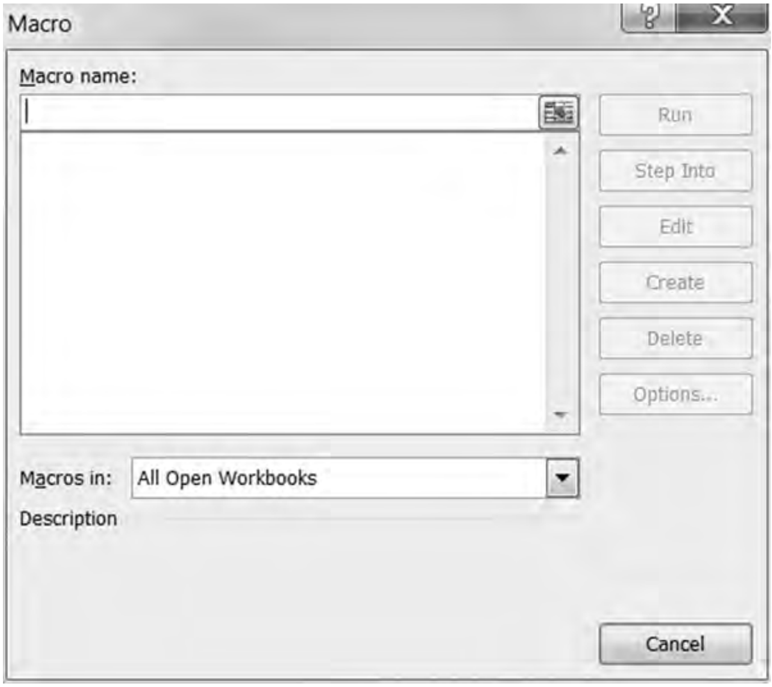
Notice that at this point there is no explanation or help for the function. The next section provides part of the remedy.

36.3 Providing Help for User-Defined Functions in the Function Wizard

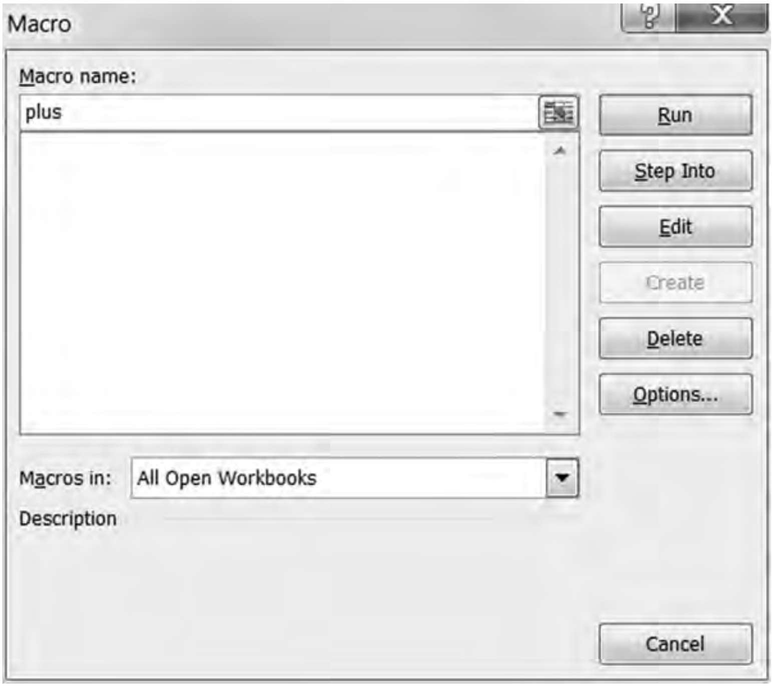
Excel's Function Wizard (shown below) provides a short help line (an explanation of what the function does). Here's how Excel explains its own functions in the Function Wizard:



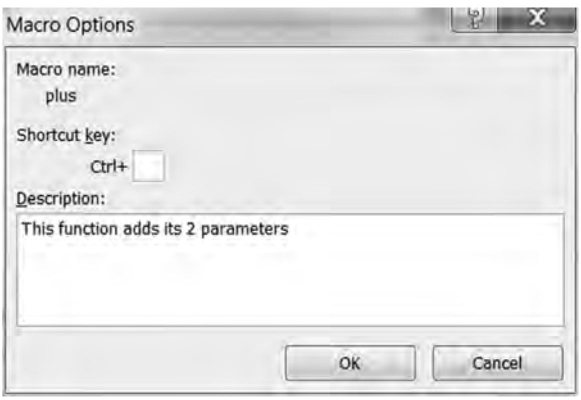
To attach a text description to our function, activate the macro selection box. You can do this either from the Excel ribbon (**Developer|Macros**) or by using the keyboard shortcut [Alt] + F8.



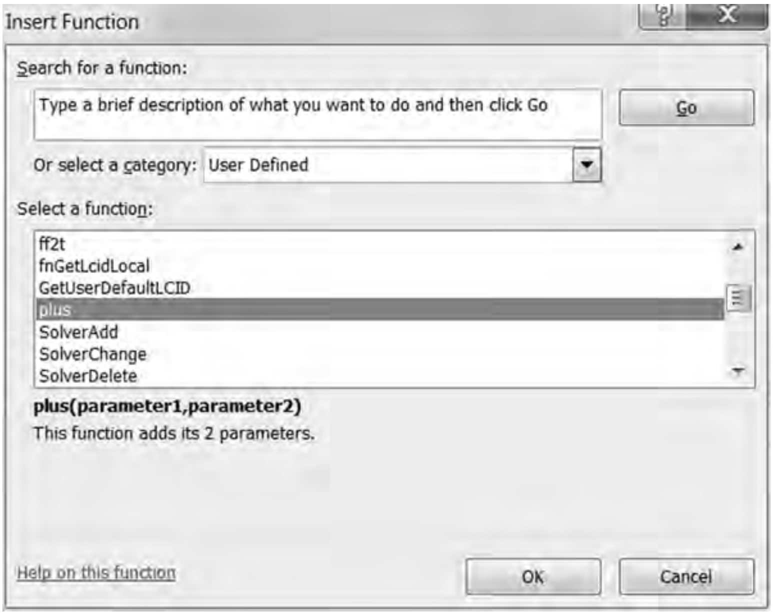
Click in the **Macro name** box, and type the name of the function (notice that you don't see the function name in the macro dialogue box above … you have to type it in):



Click on the **Options** button:



Type the description in the **Description** box. Click **OK**, and close the macro selection box. Our function now has a help line.

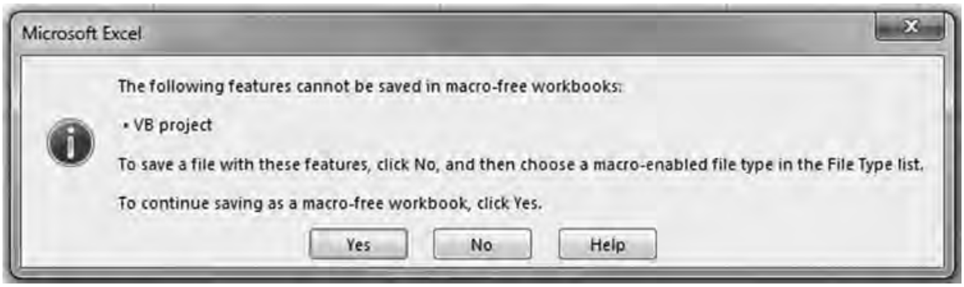


Excel functions have help lines attached to each of the parameters and a help file entry. We can supply the same for our function; sadly, the subject is beyond the scope of this introduction.

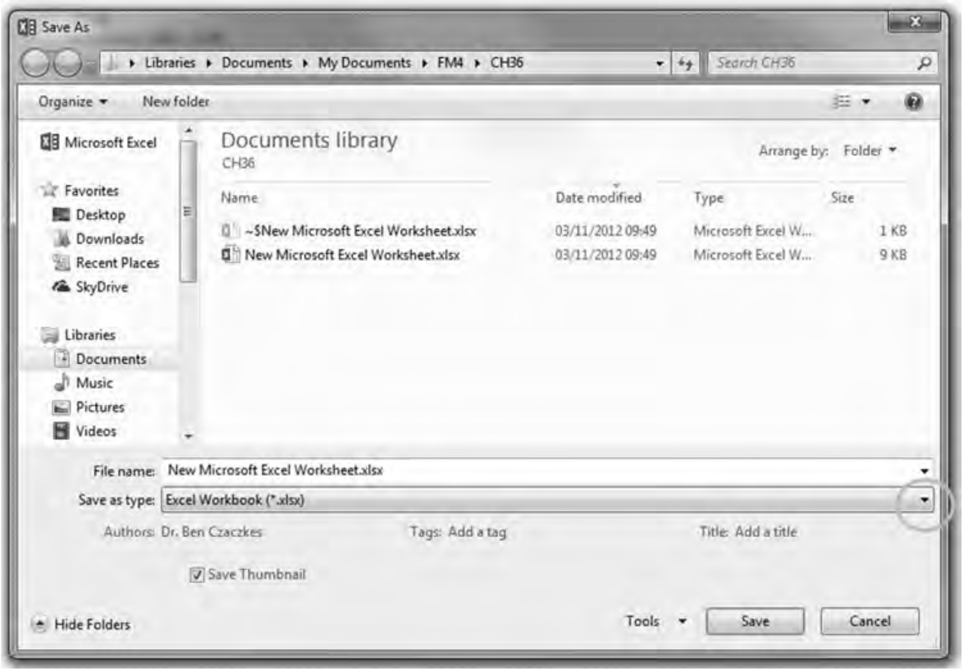
36.4 Saving Excel Workbook with VBA Content

At some point in the process, you need to save your work.[3] Starting with Excel 2007, an Excel workbook with VBA content has to be saved as a “macro-enabled file.” When you first try to save a workbook with VBA content, Excel will present you with the following message:

3. We suggest soon and often.



You should choose **No** and get the **Save As** dialog to enable you to choose a new file type.



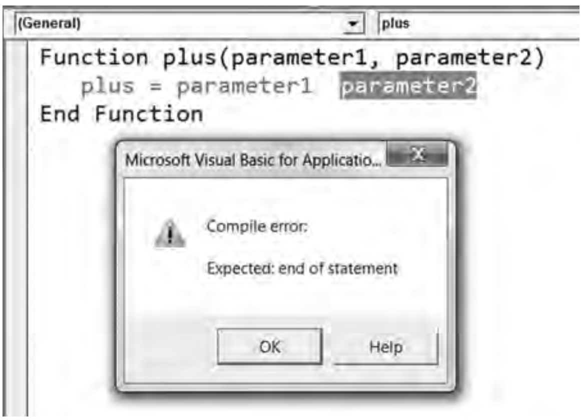
Now open the circled selection, select the second option, **xlsm**, and save the workbook. If you use VBA often, then you might consider changing the default Excel file type to **xlsm**.[4]

36.5 Fixing Mistakes in VBA

Once you start using VBA, you're sure to make mistakes. In this section we illustrate several typical mistakes and help you correct them. This list is not meant to be exhaustive—we have selected mistakes typically made by VBA beginners.

Mistake 1: Using the Wrong Syntax

Suppose that in writing **Plus** you forget the "+" between **parameter1** and **parameter2** (recall that the function is supposed to return **parameter1 + parameter2**). Once you hit the Enter key, you get the following error message:

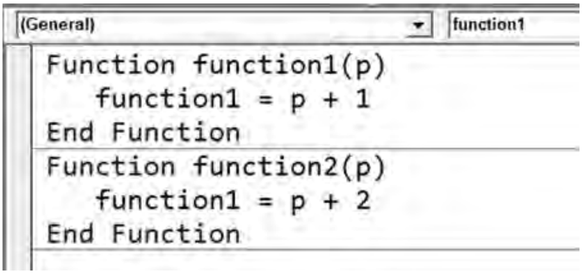


Clicking the OK button corrects this problem.

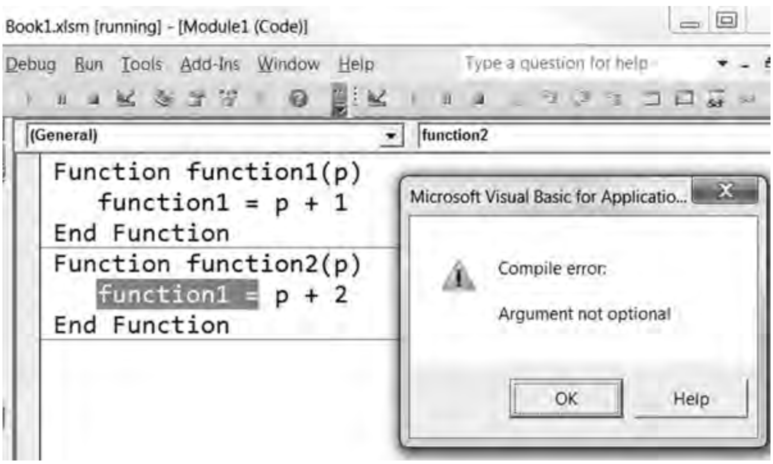
4. The command is **File|Options|Save|Save files in this format**.

Mistake 2: Right Syntax with a Typing Error

It's easy to make typing errors that will only be detected once you try to use the function. In the example below, we define two functions—**function1** and **function2**. Unfortunately, the program line for **function2** mistakenly calls the function "function1":

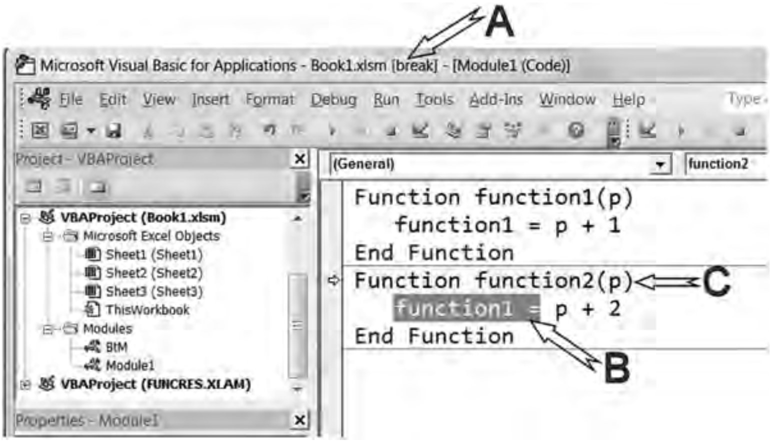


The VBA editor does not immediately recognize this mistake. The mistake will pop up when you try to use the function in a worksheet. Excel will notify you that you've made a mistake and take you to the VBA editor:



If you recognize your mistake, you can correct it. You can also try to go to the VBA help by clicking **Help** (in many cases this will lead to an incomprehensibly complicated explanation).

Suppose you recognize your mistake. You click **OK**, and get ready to correct the error by replacing the word "Function1" with "Function2." At this point your screen looks like this:



Notice:

A. The word **[break]** in the title bar.

B. The offending symbol is selected.

C. The function line is highlighted and pointed to by an arrow in the margin.

Because VBA found an error while trying to execute the function, it moved into a special execution mode called debug-break mode. For now all we need to do is get out of this special mode so we can get on with our work. We do this by clicking the ■ icon on the VBA toolbar. Now you can fix the function and use it.

We can (and should) have VBA check the module for errors before trying to use the functions in the module. From the VBA menu we select **Debug|Compile VBAproject**; this will find the first error in the module and point it out as before but without going into debug-break mode.

36.6 Conditional Execution: Using If Statements in VBA Functions

In this section we explore the **If** statements available to you in VBA. Not all things in life are linear, and sometimes decisions have to be made. **If** statements are one way of doing this in VBA

The One-Line If Statement

The one-line **If** statement is the simplest way to control the execution of a VBA function: One statement is executed if a condition is true and another is executed if a condition is not true. The complete condition and its statement should be on one line. Here's an example:

```
Function OneLineIf(Parameter)
        If Parameter > 5 Then OneLineIf = 1
        Else OneLineIf = 15
End Function
```

We can now use the function **OneLineIf** in Excel. When **Parameter** is > 5, **OneLineIf** returns 1 and when **Parameter** is < 5, **OneLineIf** returns 15.

	A	B	C
1	**ONELINEIF IN ACTION**		
2	Parameter		
3	12	1	<-- =OneLineIf(A3)
4	3	15	<-- =OneLineIf(A4)

The one-line **If** statement doesn't even need the **Else** part. The function below, **OneLineIf2**, returns 0 if the condition "Parameter > 5" is not fulfilled:

```
Function OneLineIf2(Parameter)
    If Parameter > 5 Then OneLineIf = 1
End Function
```

	A	B	C
6	ONELINEIF2 IN ACTION		
7	Parameter		
8	12	1	<-- =OneLineIf2(A8)
9	3	0	<-- =OneLineIf2(A9)

Good Programming Practice: Assign a Value to Your Function *First*

In the above functions, it would be good programming practice to first assign a value to the function before introducing the **If** statement. This way we know that **OneLineIf3** defaults to −16 if the condition on **Parameter** is not fulfilled.

```
Function OneLineIf3(Parameter)
   OneLineIf3 = -16
   If Parameter > 5 Then OneLineIf3 = 1
End Function
```

To see the difference this makes, look at the spreadsheet below:

	A	B	C
11	ONELINEIF3 IN ACTION		
12	Parameter		
13	12	1	<-- =OneLineIf3(A13)
14	3	-16	<-- =OneLineIf3(A14)

If … ElseIf Statements

If more than one statement is to be conditionally executed, the block **If… ElseIf** statement can be used. It uses the following syntax:

If Condition0 **Then**

  Statements

ElseIf Condition1 **Then**

  Statements

[… More ElseIfs …]

Else

Statements

End If

The **Else** and **ElseIf** clauses are both optional. You may have as many **ElseIf** clauses as you want following an If, but none can appear after an **Else** clause. **If** statements can be contained within one another.

Here's an example:

```
Function BlockIf(Parameter)
    If Parameter < 0 Then
        BlockIf = -1
    ElseIf Parameter = 0 Then
        BlockIf = 0
    Else
        BlockIf = 1
    End If
End Function
```

Here's how this function works in Excel:

	A	B	C
23	BLOCKIF IN ACTION		
24	Parameter		
25	-3	-1	<-- =BlockIf(A25)
26	0	0	<-- =BlockIf(A26)
27	13	1	<-- =BlockIf(A27)

Nested If Structures

As stated in the previous section, **If** statements can be used as part of the statements used in another **If** statement. A program structure that has some **If** statements inside others is called a *nested If* structure. Each **If** statement in the structure must be a complete **If** statement. Either the one-line or the block version can be used.

The following function demonstrates the use of the **NestedIf** structure:

```
Function NestedIf(P1, P2)
   If P1 > 10 Then
     If P2 > 5 Then NestedIf = 1 Else NestedIf
     = 2
   ElseIf P1 < -10 Then
     If P2 > 5 Then
       NestedIf = 3
     Else
       NestedIf = 4
     End If
   Else
     If P2 > 5 Then
       If P1 = P2 Then NestedIf = 5 Else
       NestedIf = 6
     Else
       NestedIf = 7
     End If
   End If
End Function
```

This is how it looks in Excel:

	A	B	C	D
30	NESTEDIF IN ACTION			
31	11	6	1	<-- =NestedIf(A31,B31)
32	22	3	2	<-- =NestedIf(A32,B32)
33	-22	6	3	<-- =NestedIf(A33,B33)
34	-57.3	4	4	<-- =NestedIf(A34,B34)
35	6	6	5	<-- =NestedIf(A35,B35)
36	-5	7	6	<-- =NestedIf(A36,B36)
37	4	3	7	<-- =NestedIf(A37,B37)

36.7 The Boolean and Comparison Operators

The expressions used as conditions in an **If** statement are also known as Boolean expressions. Boolean expressions can have one of two values: TRUE when the condition holds, and FALSE when the condition is violated. Usually Boolean expressions are constructed using the Comparison and/or Boolean operators. The following is a list of the most common Comparison operators.

Operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
=	Equal to
<>	Not equal to

The And Boolean Operator

The next function uses a Boolean operator to check whether two conditions hold at the same time.

```
Function AndDemo(parameter1, parameter2)
    If (parameter1 < 10) And (parameter2 > 15) _
    Then
        AndDemo = 3
    Else
        AndDemo = 12
    End If
End Function
```

Here are some illustrations:

	A	B	C	D
1	ANDDEMO IN ACTION			
2	parameter1	parameter2		
3	9	14	12	<-- =AndDemo(A3,B3)
4	9	16	3	<-- =AndDemo(A4,B4)
5	11	14	12	<-- =AndDemo(A5,B5)
6	11	16	12	<-- =AndDemo(A6,B6)

Notice what **AndDemo** does: It checks **both** conditions (parameter1 < 10) **and** (parameter2 > 15). If both conditions hold, then the combined conditions hold and the function returns a value of 3. Otherwise (i.e., if either one of the conditions is violated) it returns 12. (Note that both conditions are in parentheses.)

The following function and screen shot demonstrate all four possible combinations of two conditions and the resulting combined condition:

```
Function AndTable(parameter1, parameter2)
    AndDemoTable = parameter1 And parameter2
End Function
```

	A	B	C	D
1	ANDDEMO IN ACTION			
2	parameter1	parameter2		
3	9	14	12	<-- =AndDemo(A3,B3)
4	9	16	3	<-- =AndDemo(A4,B4)
5	11	14	12	<-- =AndDemo(A5,B5)
6	11	16	12	<-- =AndDemo(A6,B6)

The Or Boolean Operator

The function **OrDemo**, illustrated below, checks whether at least one of two conditions holds:

```
Function OrDemo(parameter1, parameter2)
    If (parameter1 < 10) Or (parameter2 > 15) _
    Then
        OrDemo = 3
    Else
        OrDemo = 12
    End If
End Function
```

	A	B	C	D
18	ORDEMO IN ACTION			
19	parameter1	parameter2		
20	9	14	3	<-- =OrDemo(A20,B20)
21	9	16	3	<-- =OrDemo(A21,B21)
22	11	14	12	<-- =OrDemo(A22,B22)
23	11	16	3	<-- =OrDemo(A23,B23)

Notice what **OrDemo** does: It checks whether **either** the first condition (Parameter1 < 10) **or** the second condition (Parameter2 > 15) **or both** conditions hold. Only if both conditions are violated will the function return a value of 12. Otherwise (i.e., if either one or both of the conditions hold) it returns 3. (Note that both conditions are in parentheses.)

The following function and the screen shot demonstrate all four possible combinations of two conditions and the resulting combined condition:

```
Function OrDemoTable(parameter1, parameter2)
    OrDemoTable = parameter1 Or parameter2
End Function
```

	A	B	C	D
1	ORTABLE IN ACTION			
2	parameter1	parameter2		
3	FALSE	FALSE	FALSE	<-- =ORDemoTable(A3,B3)
4	FALSE	TRUE	TRUE	<-- =ORDemoTable(A4,B4)
5	TRUE	FALSE	TRUE	<-- =ORDemoTable(A5,B5)
6	TRUE	TRUE	TRUE	<-- =ORDemoTable(A6,B6)

36.8 Loops

Looping structures are used when you need to do something repeatedly. As always there is more than one way to achieve the desired effect. In general there are two major looping constructs:

- *A top-checking loop*: The loop condition is checked before anything else gets done. The something to be done can be left undone if the condition is not fulfilled on entry to the loop.
- *A bottom-checking loop*: The loop condition is checked after the something to be done is done. The something to be done will always be done at least once.

VBA has the two major looping structures covered from all possible angles by the **Do** statement and its variations. All the following subsections will use a version of the factorial function for demonstration purposes. The function used is defined as:

$$f(0)=1 \quad f(1)=1 \quad f(2)=2*f(1)=2 \ldots f(n)=n*f(n-1)$$

The Do While Statement

The **Do While** statement is a member of the top-checking loops family. It makes VBA execute one or more statements **zero** or more times, while a condition is true. The following function demonstrates this behavior:

```
Function DoWhileDemo(N)
    If N < 2 Then
        DoWhileDemo = 1
    Else
        i = 1
        j = 1
        Do While i <= N
            j = j * i
            i = i + 1
        Loop
        DoWhileDemo = j
    End If
End Function
```

	A	B	C
1	DOWHILEDEMO IN ACTION		
2	5	120	<-- =DoWhileDemo(A2)
3	9	362880	<-- =DoWhileDemo(A3)
4	13	6227020800	<-- =DoWhileDemo(A4)

The Do … Loop While Statement

The **Do** … **Loop While** statement is a member of the bottom-checking loops family. It makes VBA execute one or more statements **one** or more times, while a condition is true. The following function demonstrates this behavior:

```
Function DoLoopWhileDemo(N)
    If N < 2 Then
        DoLoopWhileDemo = 1
    Else
        i = 1
        j = 1
        Do
            j = j * i
            i = i + 1
        Loop While i <= N
        DoLoopWhileDemo = j
    End If
End Function
```

	A	B	C
1	DOLOOPWHILEDEMO IN ACTION		
2	5	120	<-- =DoLoopWhileDemo(A2)
3	9	362880	<-- =DoLoopWhileDemo(A3)
4	13	6227020800	<-- =DoLoopWhileDemo(A4)

The Do Until Statement

The **Do Until** statement is a member of the top-checking loops family. It makes VBA execute one or more statements **zero** or more times, until a condition is met. The following function demonstrates this behavior:

```
Function DoUntilDemo(N)
    If N < 2 Then
        DoUntilDemo = 1
    Else
        i = 1
        j = 1
        Do Until i > N
            j = j * i
            i = i + 1
        Loop
        DoUntilDemo = j
    End If
End Function
```

	A	B	C
1	**DOUNTILDEMO IN ACTION**		
2	5	120	<-- =DoUntilDemo(A2)
3	9	362880	<-- =DoUntilDemo(A3)
4	13	6227020800	<-- =DoUntilDemo(A4)

The Do … Loop Until Statement

The **Do** … **Loop Until** statement is a member of the bottom-checking loops family. It makes VBA execute one or more statements **one** or more times, until a condition becomes true. The following function demonstrates this behavior:

```
Function DoLoopUntilDemo(N)
    If N < 2 Then
        DoLoopUntilDemo = 1
    Else
        i = 1
        j = 1
        Do
            j = j * i
            i = i + 1
        Loop Until i > N
        DoLoopUntilDemo = j
    End If
End Function
```

	A	B	C
1	DOLOOPUNTILDEMO IN ACTION		
2	5	120	<-- =DoLoopUntilDemo(A2)
3	9	362880	<-- =DoLoopUntilDemo(A3)
4	13	6227020800	<-- =DoLoopUntilDemo(A4)

The For Loop

One last (for now) variation on the loopy theme, the **For** loop, is used mainly for loops where the number of times the action is repeated is known in advance. The following functions demonstrate its use and variations:

```
Function ForDemo1(N)
    If N <= 1 Then
        ForDemo1 = 1
    Else
        j = 1
        For i = 1 To N Step 1
            j = j * i
        Next i
        ForDemo1 = j
    End If
End Function
```

	A	B	C
1	**FORDEMO1 IN ACTION**		
2	5	120	<-- =ForDemo1(A2)
3	9	362880	<-- =ForDemo1(A3)
4	13	6227020800	<-- =ForDemo1(A4)

The **Step** part of the statement can be dropped if (as is in our case) the increment is 1. For example:

```
For i = 1 To N
   j = j * i
Next i
```

If you want the loop to count down, the **Step** argument can be negative, as demonstrated in the next function:

```
Function ForDemo2(N)
    If N <= 1 Then
        ForDemo2 = 1
    Else
        j = 1
        For i = N To 1 Step -1
            j = j * i
        Next i
        ForDemo2 = j
    End If
End Function
```

	A	B	C
1	**FORDEMO2 IN ACTION**		
2	5	120	<-- =ForDemo2(A2)
3	9	362880	<-- =ForDemo2(A3)
4	13	6227020800	<-- =ForDemo2(A4)

The **For** loop can be exited early by using the **Exit For** statement as demonstrated in the next function (not the factorial function).

```
Function ExitForDemo(Parameter1, Parameter2)
    Sum = 0
    For i = 1 To Parameter1
        Sum = Sum + i
        If Sum > Parameter2 Then Exit For
    Next i
    ExitForDemo = Sum
End Function
```

	A	B	C	D
1	EXITFORDEMO IN ACTION			
2	Parameter1	Parameter2		
3	5	22	15	<-- =ExitForDemo(A3,B3)
4	6	22	21	<-- =ExitForDemo(A4,B4)
5	7	22	28	<-- =ExitForDemo(A5,B5)
6	8	22	28	<-- =ExitForDemo(A6,B6)

36.9 Using Excel Functions in VBA

VBA can use most of Excel's worksheet functions. We illustrate by showing how to define the binomial distribution (even though this, itself, is an Excel function). The probability distribution of a binomial random variable is defined as $Binom(p,n,x)=\binom{n}{x}p^x(1-p)^{n-x}$ where p is the probability of success; x is the number of successes, and n is the number of trials. $\binom{n}{x}=\frac{n!}{(n-x)!x!}$ is the binomial coefficient, which gives the number of ways of choosing x elements from among n elements. For example, suppose you want to form a two-person team from eight candidates and you want to know how many possible teams can be formed. The answer is given by $\binom{8}{2}=\frac{8!}{6!2!}=\frac{8\cdot7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1}{6\cdot5\cdot4\cdot3\cdot2\cdot1*2\cdot1}=28$. The Excel function **Combin**(8, 2) does this calculation.

We use this Excel function in the following VBA function:

```
Function Binomial(p, n, x)
    Binomial = Application.WorksheetFunction. _
    Combin(n, x) * p ^ x * (1 - p) ^ (n - x)
End Function
```

As usual, this can be applied inside a spreadsheet:

	A	B	C
1	**BINOMIAL IN ACTION**		
2	p	0.5	
3	n	10	
4	x	6	
5	Binomial	0.20507813	<-- =Binomial(B2,B3,B4)

Note that we used **Application.WorksheetFunction.Combin(n, x)** to compute $\binom{n}{x}$ in our function. As you might guess from its name (**Application.WorksheetFunction.Something**), this function is the Excel Worksheet function **Combin( )**. Most, but not all,[5] Excel worksheet functions can be used in VBA in exactly the same way. For a complete list see the Help file.

One more thing to notice is the underscore (_) preceded by a space at the end of line 2. If a line gets too long to deal with, it can be continued on the next line using this contraption (the second and third lines of **Binomial** are one line as far as VBA is concerned).[6]

Suppose we try to use our **Binomial** function to calculate **Binomial (0.5,10,15)**. This won't work:

	A	B	C
1	**BINOMIAL IN ACTION**		
2	p	0.5	
3	n	10	
4	x	15	
5	Binomial	#VALUE!	<-- =Binomial(B2,B3,B4)

5. When an equivalent function is available as a native VBA function, the corresponding Excel function is not available in VBA. For example, in VBA use **rnd( )** and not **Application.WorksheetFunction.Rand()** and **sqr( )** and not **Application.WorksheetFunction.Sqrt( )**.

6. What's too long? This is a matter of programming taste, but for our purposes any line over 70–80 characters is considered too long

The reason for the problem is that in the computation $\binom{n}{x}$ used in **Binomial**, we have to have $x < n$. In this case, VBA causes Excel to return the error message **#VALUE!**. The subject of Excel error values is somewhat obscure and is discussed in the Appendix to this chapter.

36.10 Using User-Defined Functions in User-Defined Functions

User-defined functions can be used in other user-defined functions, just like Excel functions. The next function is a replacement for the COMBIN worksheet function. COMBIN is defined as $c(n, x) = \frac{n!}{(n-x)!x!}$ where ! stands for the factorial function. (Recall that the factorial function *n!* is defined for any $n > 0$: 0! = 1, and for $n > 0$, $n! = n*(n-1)*(n-2) \ldots 1$.)

We will now write our VBA version of the two functions: the factorial function and the COMBIN function.

```
 1 Function HomeFactorial(n)
 2   If Int(n) <> n Then
 3      HomeFactorial = CVErr(xlErrValue)
 4   ElseIf n < 0 Then
 5      HomeFactorial = CVErr(xlErrNum)
 6   ElseIf n = 0 Then
 7      HomeFactorial = 1
 8   Else
 9      HomeFactorial = HomeFactorial(n - 1) * n
10   End If
11 End Function
```

Line 2 checks if the input is an integer by comparing the integer part of “n” to “n.” The function “Int” is a part of VBA. If we have erred, for example, by asking for **HomeFactorial(3.3)**, then line 3 of the program will cause Excel to return **#VALUE!**. Similarly, lines 4 and 5 check if we have improperly asked for **HomeFactorial** of a negative number; if this is the case, then line 5 causes Excel to return **#NUM!**. For a fuller explanation of the use of error values, see the Appendix to this chapter.

Line 9 introduces a new concept; the function uses itself to calculate the value it should return. This is called recursion. Here's an illustration of the function in action:

	A	B	C	D	E
1	**RECURSION IN ACTION**				
2	1	1	<-- 1	1	<-- =HomeFactorial(A2)
3	2	2	<-- =B2*A3	2	<-- =HomeFactorial(A3)
4	3	6	<-- =B3*A4	6	<-- =HomeFactorial(A4)
5	4	24	<-- =B4*A5	24	<-- =HomeFactorial(A5)
6	5	120	<-- =B5*A6	120	<-- =HomeFactorial(A6)

We can now use **HomeFactorial** to create our VBA version of **Combin** (which we will call **HomeCombin**):

```
Function HomeCombin(n, x)
    HomeCombin = HomeFactorial(n) / _
      (HomeFactorial(n - x) * HomeFactorial(x))
End Function
```

Finally, we can use **HomeCombin** to create a VBA version of the binomial function:

```
Function HomeBinom(p, n, x)
    If n < 0 Then
        HomeBinom = CVErr(xlErrValue)  'Make the function
                                       'return #VALUE!
    ElseIf x > n Or x < 0 Then
        HomeBinom = CVErr(xlErrNum)    'Make the function
                                       'return #NUM!
    Else
        HomeBinom = HomeCombin(n, x) _
                    * p ^ x * p ^ (n - x)
    End If
End Function
```

Putting Comments in VBA Code

As illustrated above, VBA will ignore anything on a line which follows an apostrophe (note that each new line of comments has to begin with an apostrophe.

Exercises

1. Write a VBA function for $f(x) = x^2 - 3$.

	A	B	C
1	**Exercise 1**		
2			
3	X		
4	1	-2	<-- =Exercise1(A4)
5	2	1	<-- =Exercise1(A5)
6	3	6	<-- =Exercise1(A6)

2. Write a VBA function for $f(x)=\sqrt{2x^2}+2x$. Note that there are two ways to do this: The first is to use the VBA function **Sqr**. The second is to use the VBA operator "$\wedge$". We suggest you try both.

	A	B	C
8	**Exercise 2**		
9			
10	X	Exercise2	
11	1	3.414213562	<-- =Exercise2(A10)
12	2	6.828427125	<-- =Exercise2(A10)
13	1	3.414213562	<-- =Exercise2a(A12)
14	2	6.828427125	<-- =Exercise2a(A13)

3. Suppose a share was priced at price P_0 at time 0, and suppose that at time 1 it will be priced P_1. Then the continuously compounded return is defined as $return=\ln\left(\frac{P_1}{P_0}\right)$. Implement this function in VBA. There are two ways to do this: You can use **Worksheetfunction.Ln** or the VBA function **Log**.

	A	B	C	D
18	**Exercise 3**			
19				
20	P0	P1		
21	100	110	0.09531018	<-- =Exercise3(A21,B21)
22	100	200	0.693147181	<-- =Exercise3(A22,B22)
23	100	110	0.09531018	<-- =Exercise3a(A23,B23)
24	100	200	0.693147181	<-- =Exercise3a(A24,B24)

4. A bank offers different yearly interest rates to its customers based on the size of the deposit in the following way:

- For deposits up to 1,000, the interest rate is 5.5%
- For deposits from 1,000 and up to 10,000, the interest rate is 6.3%
- For deposits from 10,000 and up to 100,000, the interest rate is 7.3%
- For all other deposits the interest rate is 7.8%

Implement the function **Interest(Deposit)** in VBA. Note that you can use the **BlockIf** structure.

	A	B	C
1	**Exercise 4**		
2	Deposit		
3	-1	#VALUE!	<-- =Interest(A3)
4	100	5.50%	<-- =Interest(A4)
5	1100	6.30%	<-- =Interest(A5)
6	9999.99	6.30%	<-- =Interest(A6)
7	10000	6.30%	<-- =Interest(A7)
8	10000.001	7.30%	<-- =Interest(A8)
9	100000.001	7.80%	<-- =Interest(A9)

5. Using the function in exercise 4, implement a function **NewDFV(Deposit, Years)**. The function will return the future value of a deposit with the bank assuming the deposit and accrued interest is reinvested for a given number of years. Thus, for example, **NewDFV(10000,10)** will return 10000*(1.063)∧10.

	A	B	C	D
1	**Exercise 5**			
2	Deposit	Years		
3	10000	10	18421.82	<-- =NewDFV(A3,B3)
4	10000.001	10	20230.06	<-- =NewDFV(A4,B4)

6. An investment company offers a bond linked to the FT100 index. On redemption the bond pays the face value plus the largest of A: the face value times the change in the index. Or B: 5% yearly interest compounded monthly. Thus, for example, 100 invested when the index was 110 and redeemed a year later when the index was 125 will pay A: 100 + 100*(125 − 110)/110 = 113.636 and not B: 100*(1 + 0.05/12) ∧ 12 = 105.116. Implement a VBA function **Bond(Deposit, Years, FT0, FT1).**

	A	B	C	D	E	F
64	**Exercise 6**					
65						
66	Deposit	Years	FT0	FT1		
67	100	1	110	125	113.636	<-- =Bond(A67,B67,C67,D67)
68	100	1	110	100	105.116	<-- =Bond(A68,B68,C68,D68)
69	100	12	110	125	1,261.394	<-- =Bond(A69,B69,C69,D69)
70	100	12	110	1387.53	1,261.394	<-- =Bond(A70,B70,C70,D70)
71	100	12	110	1387.535	1,261.395	<-- =Bond(A71,B71,C71,D71)

7. Implement a VBA function **ChooseBond(Deposit, Years, FT0, FT1).** The function will return the value 1 if the superior investment is the bank in exercise 5 or the value 2 if it is the company in exercise 6.

	A	B	C	D	E	F
76	**Exercise 7**					
77						
78	Deposit	Years	FT0	FT1		
79	100	1	110	125	2	<-- =ChooseBond(A79,B79,C79,D79)
80	100	1	110	110	1	
81	100	1	110	116.04	1	
82	100	1	110	116.05	2	
83	100000	1	110	125	2	<-- =ChooseBond(A83,B83,C83,D83)
84	100000	1	110	110	1	
85	100000	1	110	118.02	1	
86	100000	1	110	118.03	2	

8. A bank offers the following saving scheme: Invest a fixed amount on the first of each month for a set number of years. On the first of the month after your last installment, you get your money plus the accrued interest. The bank quotes a yearly interest rate but interest is calculated and compounded on a monthly basis. Eight different interest rates are offered depending on the monthly deposit and the number of years the program is to run.
The following table lists the interest rates offered.

	For sums <= 100 a month	For sums > 100 a month
For a period of 2 years	3.5%	3.9%
For a period of 3 years	3.7%	4.5%
For a period of 4 years	4.2%	5.1%
For a period of 5 years	4.6%	5.6%

Write a two-argument function **DFV(Deposit, Years)**, returning the future value of such an investment.

	A	B	C	D
1	**Exercise 8**			
2	Deposit	Years	DFV	
3	10	5	675.7458	<-- =DFV(A3,B3)
4	10	4	523.5107	<-- =DFV(A4,B4)
5	10	3	381.2934	<-- =DFV(A5,B5)
6	10	2	248.9488	<-- =DFV(A6,B6)
7	10	1	120	<-- =DFV(A7,B7)

9. Using the information provided in exercise 8 write a two-argument function **DEP(DFV, Years)** that will return the monthly contribution necessary to get a certain sum in the future (2, 3, 4, or 5 years). Note: This problem is more interesting; remember that the interest rate depends on the monthly contribution.

	A	B	C	D
1	**Exercise 9**			
2	DFV	Years	DEP	
3	-100	2	-4.01689	<-- =DEP(A3,B3)
4	100	2	4.01689	<-- =DEP(A4,B4)
5	1000	4	19.10181	<-- =DEP(A5,B5)
6	2499	2	99.96106	<-- =DEP(A6,B6)
7	2500	2	100.0011	<-- =DEP(A7,B7)

10. Fibonacci numbers are named after Leonardo Fibonacci (1170–1230), an outstanding European mathematician of the medieval period. Fibonacci numbers are defined as follows:

$F(0)=0$

$F(1)=1$

$F(2)=F(0)+F(1)=1$

$F(3)=F(1)+F(2)=2$

$F(4)=F(2)+F(3)=3$

…

In general $F(n)=F(n-2)+F(n-1)$.

Write a **recursive** VBA function that computes the *n*th number in the Fibonacci series.

	A	B	C
1	**Exercise 10**		
2	**n**	**Fibonacci**	
3	0	0	<-- =Fibonacci(A3)
4	1	1	<-- =Fibonacci(A4)
5	2	1	<-- =Fibonacci(A5)
6	3	2	<-- =Fibonacci(A6)
7	4	3	<-- =Fibonacci(A7)
8	5	5	<-- =Fibonacci(A8)
9	6	8	<-- =Fibonacci(A9)
10	7	13	<-- =Fibonacci(A10)

11. Write a VBA function that computes the *n*th number in the Fibonacci series; do not use recursion.

	A	B	C
1	**Exercise 11**		
2	**n**	**LoopFibonacci**	
3	0	0	<-- =LoopFibonacci(A3)
4	1	1	<-- =LoopFibonacci(A4)
5	2	1	<-- =LoopFibonacci(A5)
6	3	2	<-- =LoopFibonacci(A6)
7	4	3	<-- =LoopFibonacci(A7)
8	5	5	<-- =LoopFibonacci(A8)
9	6	8	<-- =LoopFibonacci(A9)
10	7	13	<-- =LoopFibonacci(A10)

Appendix: Cell Errors in Excel and VBA

Excel uses a special kind of value to report errors. The **CVErr( )** function is part of VBA. It converts a value, supplied by you, to the special kind of value used for errors in Excel. Excel has a number of error values that a function can return to signal that something went wrong. Here's an example: The function **NewMistake(x,y)** returns the result *x*/*y*. However, if *y* = 0, the function outputs the (cryptic) error message #DIV0!.

```
Function NewMistake(x, y)
    If y <> 0 Then NewMistake = x / y Else _
        NewMistake = CVErr(xlErrDiv0)
End Function
```

To Anticipate Future Confusion

All the VBA error values are written "xlErr … ." Because the typed alphabet letter "l" also looks like the number one, it would have been easier had Microsoft used capital letters "XLErr … ." But …

This is **NewMistake** in Excel:

	A	B	C	D
1	**NewMistake In Action**			
2				
3	X	Y	NewMistake	
4	1	2	0.5	<-- =NewMistake(A4,B4)
5	2	1	2	<-- =NewMistake(A5,B5)
6	0	1	0	<-- =NewMistake(A6,B6)
7	1	0	#DIV/0!	<-- =NewMistake(A7,B7)

Error values and their explanation are listed below.

Error Value	VBA Name	Possible causes
#NULL!	XlErrNull	The #NULL! error value occurs when you specify an intersection of two areas that do not intersect.
#DIV/0!	XlErrDiv0	The #DIV/0! error value occurs when a formula divides by 0 (zero).
#VALUE!	XlErrValue	The #VALUE! error value occurs when the wrong type of argument is used.
#REF!	XlErrRef	The #REF! error value occurs when a cell reference is not valid.
#NAME?	XlErrName	The #NAME? error value occurs when Microsoft Excel doesn't recognize text in a formula.
#NUM!	XlErrNum	The #NUM! error value occurs when a problem occurs with a number in a formula or function.
#N/A	XlErrNA	The #N/A error value occurs when a value is not available to a function or formula.

37 Variables and Arrays

37.1 Overview

In the first part of this chapter we introduce function variable definitions. The second part of the chapter introduces arrays. An array is a group of variables of the same type sharing the same name and referenced individually using an index. Vectors and matrices are good examples of one- and two-dimensional arrays. The relationship between arrays and worksheet ranges opens the discussion, followed by sections describing simple and dynamic arrays (whose size can be changed at run time). The chapter concludes with sections on the use of arrays as parameters and a short discussion of typed variables.

37.2 Defining Function Variables

Function variables are used to store values. Function variables can be either parameter or simple variables. Parameters are defined when the function is defined by listing them within parentheses after the function's name. Up until now we used simple variables as and when needed, relying on VBA to define the variable for us when it was first used. In most scenarios encountered in this book, this practice is good enough, and it has the advantage of being quick.

The first time we encountered both flavors of function variables was in the function **DoWhileDemo**:

```
Function DoWhileDemo(N)
    If N < 2 Then
        DoWhileDemo = 1
    Else
        i = 1 ' A Loop counter
        j = 1 ' An accumulator for the series
        Do While i <= N
            j = j * i
            i = i + 1
        Loop
        DoWhileDemo = j
    End If
End Function
```

The variable **N** is a parameter that gets its value from the application that activates the function (either Excel or another function). The variables **i, j** are simple variables. Function variables (aka internal or local variables) of both types are recognized only in the function in which they were defined (implicitly or explicitly) and are not recognized by Excel or by other VBA functions.

As this is a very short function, there really is no reason to define the variables explicitly and the addition of comments makes everything clear enough. Longer functions with more variables might benefit by defining the variables at the top of the function, as it makes for more maintainable, and clear programming. Simple variables are defined using the **Dim** statement as demonstrated by the following function:

```
Function NewDoWhileDemo(N)
Dim i ‘ a Loop counter
Dim j ‘ An accumulator for the series
    If N < 2 Then
        NewDoWhileDemo = 1
    Else
        i = 1
        j = 1
        Do While i <= N
            j = j * i
            i = i + 1
        Loop
        NewDoWhileDemo = j
    End If
End Function
```

The Option Explicit Statement

We can make VBA alert us if we use an undeclared variable by inserting the **Option Explicit** statement as the first line in the module. With this statement any use of an undeclared variable will result in an error and not the creation of a new variable. The **Option Explicit** statement holds for all the routines in the module.

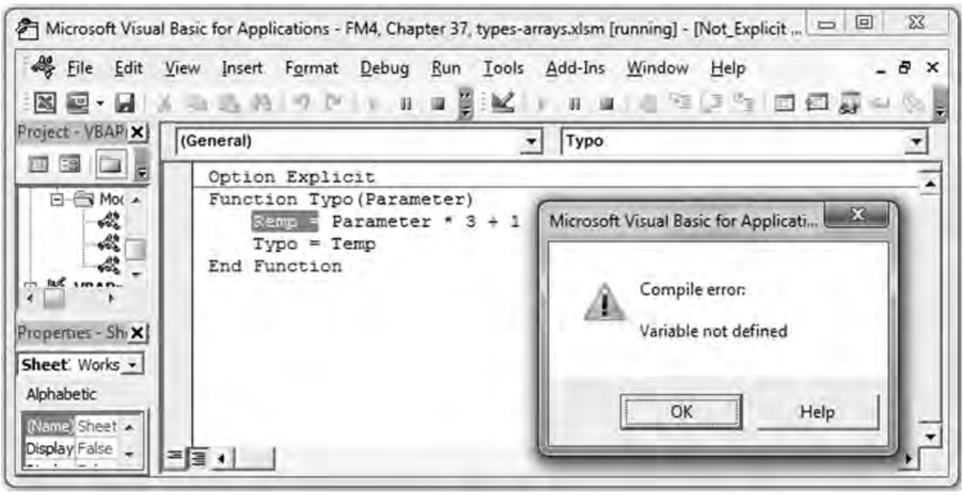
Forcing the definition of variables can help prevent errors from creeping into your functions. Here is an (slightly forced) example: The following function contains a typing error (“Temp” is spelled “Remp”):

```
Function Typo(Parameter)
    Remp = Parameter * 3 + 1
    Typo = Temp
End Function
```

Without the **Option Explicit** statement, Excel merrily displays the following result:

	A	B	C
1	TYPO IN ACTION		
2	5	0	<-- =Typo(A2)

However, inserting the **Option Explicit** statement before the VBA code and recalculating the worksheet results in the following “Run Time Error”:



Once we are alerted to the problem, we can click the OK button, stop VBA from running, and fix the problem by replacing "Remp" with "Temp." (Recall from Chapter 36 that after you fix the mistake in VBA, you have to press the ■ button on the VBA editor toolbar.)

37.3 Arrays and Excel Ranges

A VBA array is a group of variables of the same type sharing the same name and referenced individually using an index (or indices). VBA has its own version of arrays and we shall deal with this type of array in the following sections. For now let us demonstrate the **Variant**. If we want a function to accept an Excel range as a parameter we have to leave the parameter type-less, or declare the parameter as Variant (which amounts to the same thing). From inside the function the variable looks like an array. To demonstrate, we shall now write a small function, **SumRange**, that sums the value in the first four elements of its parameter.

```
Function SumRange(R)
   S = 0
   For i = 1 To 4
      S = R(i) + S
   Next i
   SumRange = S
End Function
```

	A	B
1	**SUMRANGE IN ACTION**	
2	1	<-- 1
3	2	<-- 2
4	3	<-- 3
5	4	<-- 4
6	10	<-- =SumRange(A2:A5)

	A	B	C	D
1	**SUMRANGE IN ACTION**			
2	1	2	3	4
3	10	<-- =SumRange(A2:D2)		

In both cases the variable *R* can be treated as an array, with the first element being **R(1)** and the last element **R(4)**. Each of the elements can be treated as a single variable, that is, **R(2)** is a variable and so is **R(i-3)** (assuming that i-3 has an integer value >= 1 and <= 4). Ranges treated as arrays always start with index 1.

What happens if the range passed to our function is rectangular? To demonstrate, we introduce a modified version of **SumRange**, inserting a second parameter that tells the function how many elements to sum.

```
Function SumRange1(R, N)
   S = 0
   For i = 1 To N
      S = R(i) + S
   Next i
   SumRange = S
End Function
```

	A	B	C	D
1	SUMRANGE1 IN ACTION			
2	3	4	5	
3	6	7	8	
4	9	10	11	
5	18	<-- =SumRange1(A2:C4,4)		
6	25	<-- =SumRange1(A2:C4,5)		
7	33	<-- =SumRange1(A2:C4,6)		

As we can see, VBA treats the rectangular array as a linear array composed of the rows of the original range. The second parameter of the function **SumRange1** indicates how many elements should be summed. Thus, for example, Sumrange1(A2:C4,5) sums the first row plus two cells in the second row.

A Payback Period Function

A slightly more complex use of ranges can be shown with a simple Payback Period function. Recall that the payback period in capital budgeting refers to the period of time required for the return on an investment to "repay" the sum of the original investment. For example, a $1,000 investment which pays cash flows of $500 per year has a 2-year payback period. To simplify matters, the function **PayBack** defined below gives a whole year solution. If the sum of the cash flows for 5 years is < 0 and for 6 years is > 0, then the function will return 6. We also assume that the first cash flow is the initial investment (negative) and that no other cash flows are negative.

```
Function PayBack(R, N)
   Temp = 0
   For i = 1 To N
       Temp = Temp + R(i)
       If Temp >= 0 Then Exit For
   Next i
   PayBack = i - 1
End Function
```

	A	B	C	D	E	F
1	**PAYBACK IN ACTION**					
2	Period	1	2	3	4	5
3	Cash-flow	-1500	400	600	600	300
4	PayBack	3	<-- =PayBack(B3:F3,5)			

There are a few problems with this function as currently defined. One is that the function returns a wrong answer if the investment does not pay back its initial outlay, as demonstrated by the next screen shot.

	A	B	C	D	E	F
1	**PAYBACK IN ACTION**					
2	Period	1	2	3	4	5
3	Cash-flow	-4000	400	600	600	300
4	PayBack	5	<-- =PayBack(B3:F3,5)			

The problem is solved by inserting a check before returning the payback period.

```
Function PayBack1(r, n)
    Temp = 0
    For i = 1 To n
        Temp = Temp + r(i)
        If Temp >= 0 Then Exit For
    Next i
    If Temp >= 0 Then
        PayBack1 = i - 1
    Else
        PayBack1 = "No Payback"
    End If
End Function
```

	A	B	C	D	E	F
1	PAYBACK1 IN ACTION					
2	Period	1	2	3	4	5
3	Cash-flow	-4000	400	600	600	300
4	PayBack	No Payback	<-- =PayBack1(B3:F3,5)			

37.4 Simple VBA Arrays

There are several ways to declare VBA arrays, all using the **Dim** statement. The simplest way to declare an array is simply to tell VBA the largest value the array index can take. Unless you indicate otherwise, VBA arrays always start with index 0. In the function below, **MyArray** has 6 elements numbered 0, 1, 2, … , 5.

```
Function ArrayDemo1()
    Dim MyArray(5)
    For i = 0 To 5
        MyArray(i) = i * i
    Next i
    S = ""
    For i = 0 To 5
        S = S & " # " & MyArray(i)
    Next i
    ArrayDemo1 = S
End Function
```

If you use **ArrayDemo1** in a spreadsheet, here is the result:

	A	B
1	**ARRAYDEMO1 IN ACTION**	
2	# 0 # 1 # 4 # 9 # 16 # 25	<-- =ArrayDemo1()

Notice:

• **MyArray** has six elements (variables), the first being **MyArray(0)** and the last **MyArray(5)**. All VBA arrays start from 0, unless you specify otherwise (see discussion of **Option Base** below).

• An array element is treated just like a variable. **MyArray(2)** is a variable and so is **MyArray(i-3)** (assuming that i-3 has an integer value >= 0 and <= 5).

• The use of the concatenation operator **&**. This operator concatenates (combines) its two operands to create a string. If an operand to the concatenation operator is not a string, it is converted to a string, and then the concatenation takes place.

If you try and access an array element that is not part of the array, VBA will return an error value, as demonstrated by the following function:

```
Function ArrayDemo2(N)
   Dim MyArray(5)
   Dim i As Integer
       For i = 0 To 5
        MyArray(i) = i * i
    Next i
   ArrayDemo2 = MyArray(N)
End Function
```

	A	B	C
1	**ARRAYDEMO2 IN ACTION**		
2	0	0	<-- =ArrayDemo2(A2)
3	1	1	<-- =ArrayDemo2(A3)
4	2	4	<-- =ArrayDemo2(A4)
5	3	9	<-- =ArrayDemo2(A5)
6	4	16	<-- =ArrayDemo2(A6)
7	5	25	<-- =ArrayDemo2(A7)
8	6	#VALUE!	<-- =ArrayDemo2(A8)

LBound and UBound

LBound and **UBound** are two internal VBA functions that are very useful when dealing with arrays**.** These functions return the minimum and maximum value that an array index can have. The following function demonstrates their use on a one-dimensional array:

```
Function ArrayDemo3(N)
   Dim MyArray(5)
   If N = "LB" Then
      ArrayDemo3 = LBound(MyArray)
   ElseIf N = "UB" Then
      ArrayDemo3 = UBound(MyArray)
   End If
End Function
```

	A	B	C
1	ARRAYDEMO3 IN ACTION		
2	LB	0	<-- =ArrayDemo3(A2)
3	UB	5	<-- =ArrayDemo3(A3)

Note that the array **MyArray** has six elements, the first being **MyArray(0)** as indicated by **LBound**, and the last being **MyArray(5)** as indicated by **UBound**.

When used on a multidimensional array, a second parameter should be supplied indicating the dimension in whose bounds we are interested, as the next function demonstrates.

```
Function ArrayDemo4(Dimension, Bound)
   Dim MyArray(2, 3, 4)
   If Bound = “LB” Then
      ArrayDemo4 = LBound(MyArray, Dimension)
   ElseIf Bound = “UB” Then
      ArrayDemo4 = UBound(MyArray, Dimension)
   End If
End Function
```

	A	B	C	D
1	ARRAYDEMO4 IN ACTION			
2	LB	1	0	<-- =ArrayDemo4(B2,A2)
3	UB	1	2	<-- =ArrayDemo4(B3,A3)
4	LB	2	0	<-- =ArrayDemo4(B4,A4)
5	UB	2	3	<-- =ArrayDemo4(B5,A5)
6	LB	3	0	<-- =ArrayDemo4(B6,A6)
7	UB	3	4	<-- =ArrayDemo4(B7,A7)

How to Get the Bound of an Excel Range in a Function

Sadly the internal functions **UBound** and **LBound** do not work for a range passed to a function. We can make use of the fact that the parameter is actually a range and use some of its properties to get the result we need. The following function demonstrates this:

```
Function RangeBound(R, What)
   If What = "C" Then
      RangeBound = R.Columns.Count
   ElseIf What = "R" Then
      RangeBound = R.Rows.Count
   End If
End Function
```

	A	B	C
1	RANGEBOUND IN ACTION		
2	C	2	<-- =rangebound(D1:E5,A2)
3	R	5	<-- =rangebound(D2:E6,A3)
4	c	0	<-- =rangebound(D3:E7,A4)
5	r	0	<-- =rangebound(D4:E8,A5)

Did you notice that the function does not work with lowercase characters? If we want to be case agnostic, as one usually does, we can use the VBA function **UCase** to convert "what" to uppercase.

```
Function RangeBound1(R, What)
   If UCase(What) = "C" Then
      RangeBound1 = R.Columns.Count
   ElseIf UCase(What) = "R" Then
      RangeBound1 = R.Rows.Count
   End If
End Function
```

	A	B	C
1	RANGEBOUND1 IN ACTION		
2	C	2	<-- =rangebound1(D1:E5,A2)
3	R	5	<-- =rangebound1(D2:E6,A3)
4	c	2	<-- =rangebound1(D3:E7,A4)
5	r	5	<-- =rangebound1(D4:E8,A5)
6	1	0	<-- =rangebound1(D5:E9,A6)

Fixing Excel's NPV Function

Recall from Chapter 1 that

> Excel's language about discounted cash flows differs somewhat from the standard finance nomenclature. Excel uses the letters NPV to denote the present value (**not** the net present value) of a series of cash flows.
>
> To calculate the finance net present value of a series of cash flows using Excel, we have to calculate the present value of the future cash flows (using the Excel NPV function) and subtract from this present value the time-zero cash flow. (This is often the cost of the asset.)

Let us try and write a function **nNPV** that addresses this shortcoming. In the process we shall learn a few things about Excel Ranges in VBA. In order to make the function simple, it will only work on a row of cash flows.

```
Function nNPV(Rate, R)
   nNPV = R(1) + Application.WorksheetFunction _
   .npv(Rate, R.Range("B1", R.End(xlToRight)))
End Function
```

R.Range(CellTopLeft,CellBottomRight) returns a range defined by its parameters. Note that the cell addresses are relative to **R** and not the worksheet.

R.End(Direction) returns one of the four possible last cells in the **R** according to **Direction**. Possible values for **Direction** are xlDown, xlToLeft, xlToRight, xlUp.

Assuming **R** is a row of cells, **R.Range("B1", R.End(xlToRight))** returns a range containing all the cells in **R** excluding the first one.

	A	B	C	D	E	F
1	NNPV IN ACTION					
2	Cash Flows ▶	-400	100	100	100	100
3	Rate ▶	10%				
4	Excel NPV ▶	-75.4667776	<-- =NPV($B$3,$B$2:$F$2)			
5	Excel C_0+NPV ▶	-83.01345537	<-- =$B$2+NPV($B$3,$C$2:$F$2)			
6	nNPV ▶	-83.01345537	<-- =nNPV($B$3,$B$2:$F$2)			

A New IRR Function

Another useful function we can write using our newly acquired tools is **nIRR**. Recall from Chapter 1 that the internal rate of return (IRR) is defined as the compound rate of return r that makes the NPV equal to zero:

$$CF_0 + \sum_{t=0}^{N} \frac{CF_t}{(1+r)^t} = 0$$

We now use a technique called successive refinement to calculate IRR:

1. If we were given an initial guess for r we use it and if not we use 50% to calculate NPV.
2. If the calculated NPV is zero (or sufficiently near) we return the current guess.
3. If the calculated NPV is negative we set our guess to $r = r + r/2$.
4. If the calculated NPV is positive we set our guess to $r = r - r/2$.
5. We shall now recalculate NPV.
6. Repeat steps 2–5.

We assume that the first cash flow is negative and that all others are positive.

Here is the function:

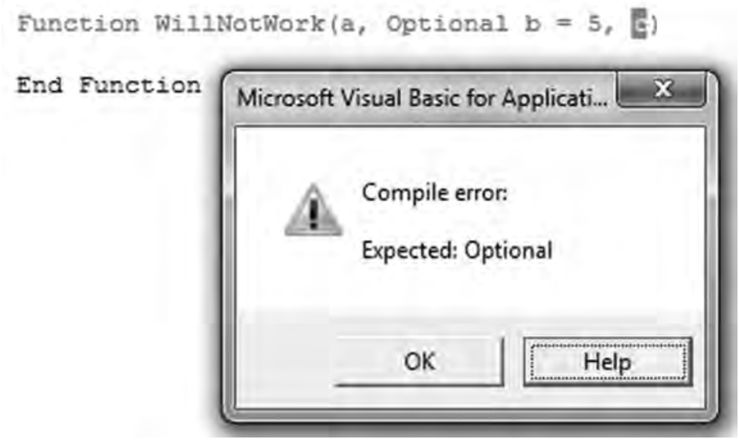
```
Function nIRR(R, Optional guess = 0.5)
   n = nNPV(guess, R)
   Do While Abs(npv) > 0.0001
      If n < 0 Then
         guess = guess - guess / 2
      Else
         guess = guess + guess / 2
      End If
      n = nNPV(guess, R)
   Loop
   nIRR = guess
End Function
```

Optional Parameters

Note that the use of **Optional guess = 0.5** to declare the last parameter as optional and give it a default value if the user did not supply one. Once a parameter is declared as optional, all the following parameters have to be declared optional as well. For example, this declaration is fine:

```
Function WillWork(a, Optional b = 5, Optional c = 4)
```

Whereas this will result in an error:



As noted, this function is very slow so it might take a few seconds to calculate its results.

	A	B	C	D	E	F
1	**NIRR IN ACTION**					
2	Cash Flows ▶	-375	100	100	100	100
3	Guess ▶	5%				
4	IRR ▶	2.63247%	<-- =IRR($B$2:$F$2,$B$3)			
5	nIRR ▶	2.63247%	<-- =nIRR($B$2:$F$2,$B$3)			
6	nIRR ▶	2.63248%	<-- =nIRR($B$2:$F$2)			
7	nNPV ▶	7.708E-11	<-- =nNPV($B$4,$B$2:$F$2)			
8	nNPV ▶	9.795E-06	<-- =nNPV($B$5,$B$2:$F$2)			
9	nNPV ▶	-9.55E-05	<-- =nNPV($B$6,$B$2:$F$2)			

The Option Base Statement

Excel arrays start at 1, whereas VBA arrays start at 0, unless otherwise defined. We can use a module option to make all not specifically declared array indices start at 1. We use **ArrayDemo3** to demonstrate. We open a new VBA module with first line “OptionBase1.” We rename our previous function to reflect this change.

```
Option Base 1
Function ArrayDemo3OptionBase1(N)
   Dim MyArray(5)
   If N = "LB" Then
      ArrayDemo3OptionBase1 = LBound(MyArray)
   ElseIf N = "UB" Then
      ArrayDemo3OptionBase1 = UBound(MyArray)
   End If
End Function
```

If we insert **Option Base 1** as the first line of the module to get (the only change is in cell B2 where we get 1 and not 0).

	A	B	C
1	ARRAYDEMO3OPTIONBASE1 IN ACTION		
2	LB	1	<-- =ArrayDemo3Optionbase1(A2)
3	UB	5	<-- =ArrayDemo3Optionbase1(A3)

The **Option Base 1** statement, like all option statements, should be inserted before all functions and subroutines in a module. Like all option statements, its effect is limited to all routines in the current module.

37.5 Multidimensional Arrays

Arrays can have more than one index. In a two-dimensional array the first index refers to the rows and the second to the columns. There is no formal limit to the number of indices you can declare in an array. The syntax for declaring a multidimensional array is demonstrated in the following functions:

```
Function Matrix1(R, C)
   Dim MyMat(2, 1)
   For i = 0 To 2
      For j = 0 To 1
         MyMat(i, j) = i * j
      Next j
   Next i
   If R >= 0 And R <= 2 And C >= 0 And C <= 1 _
Then
      Matrix1 = MyMat(R, C)
   End If
End Function
```

	A	B	C	D
1	**MATRIX1 IN ACTION**			
2	R	C	Matrix1(R,C)	
3	0	0	0	<-- =Matrix1(A3,B3)
4	1	0	0	<-- =Matrix1(A4,B4)
5	2	0	0	<-- =Matrix1(A5,B5)
6	0	1	0	<-- =Matrix1(A6,B6)
7	1	1	1	<-- =Matrix1(A7,B7)
8	2	1	2	<-- =Matrix1(A8,B8)
9	3	1	0	<-- =Matrix1(A9,B9)
10	1	3	0	<-- =Matrix1(A10,B10)

The following function demonstrates the use of **LBound** and **UBound** with multidimensional arrays:

```
Function Matrix2(R, C)
   Dim MyMat(1, 1)
   For i = LBound(MyMat, 1) To UBound(MyMat, 1)
      For j = LBound(MyMat, 2) To _
      UBound(MyMat, 2)
         MyMat(i, j) = i * j
      Next j
   Next i
   If R >= LBound(MyMat, 1) And _
      R <= UBound(MyMat, 1) And _
      C >= LBound(MyMat, 2) And _
      C <= UBound(MyMat, 2) Then
      Matrix2 = MyMat(R, C)
   End If
End Function
```

	A	B	C	D
1	**MATRIX2 IN ACTION**			
2	R	C	Matrix2(R,C)	
3	0	0	0	<-- =Matrix2(A3,B3)
4	1	0	0	<-- =Matrix2(A4,B4)
5	2	0	0	<-- =Matrix2(A5,B5)
6	0	1	0	<-- =Matrix2(A6,B6)
7	1	1	1	<-- =Matrix2(A7,B7)
8	1	2	0	<-- =Matrix2(A8,B8)

Note the use of the second argument to **LBound** and **UBound**. If used with only one argument, both functions return the largest index value the first dimension of the array can have; if the array has more than one dimension (as in this case), we can use a second argument to the function to specify the dimension we are interested in.

37.6 Dynamic Arrays and the ReDim Statement

Every so often it can be handy to have the size of an array set (and reset) when the program is running. Dynamic arrays are arrays that can have their size changed at run time. You declare dynamic arrays using the **Dim** statement but with nothing in the parentheses, as in:

```
Dim SomeName()
```

Before you can use the array you need to set its size using the **ReDim** statement, as in:

```
ReDim ArrayName(SomeIntegerExpression)
```

For example, you might type

```
ReDim Prices(12)
```

To set the size of the dynamic array **Prices** to 12 elements, a more typical case would involve the use of a variable for the size as in:

```
ReDim Prices(I)
```

This will set the size of Prices to the value of I.

The **ReDim** statement can also be used to change the size of a dynamic array (or indeed any VBA array). If you change the size of an array, all the data in the array are lost. Use **ReDim Preserve** to keep the old data, as in:

```
ReDim Preserve ArrayName(SomeIntegerExpression)
```

The following function calculates the present value of a series of future cash flows. To simplify the function, the interest rate is fixed at 5% per period. The function illustrates the use of a dynamic array (the variable **CF)** that derives its size from the size of the original input (the variable **n**):

```
Function DynPV(r As Range)
  ` n is number of periods
  ` cf() is dynamic array for cash
  ` flows
  Dim n
  Dim cf()
  Dim Temp
  Dim i
  `Below we distinguish if the data
  `is in a column or in a row
  If r.Columns.Count = 1 Then
    n = r.Rows.Count
  ElseIf r.Rows.Count = 1 Then
    n = r.Columns.Count
  Else
    Exit Function
  End If
  ` re-dimension the array
  ReDim cf(1 To n)
  For i = 1 To n
    cf(i) = r(i)
  Next i
  Temp = 0
  For i = 1 To n
  Temp = Temp + cf(i) / 1.05 ^ i
  Next i
  DynPV = Temp
End Function
```

Running the function produces the following:

	A	B	C
1	**DYNPV IN VERTICAL ACTION**		
2	Cash Flows		
3	100		
4	200		
5	300		
6	DynPV ▶	535.79527	<-- =DynPV(A3:A5)

	A	B	C	D
1	**DYNPV IN HORIZONTAL ACTION**			
2	Cash Flows ▶	100	200	300
3	DynPV ▶	535.79527	<-- =DynPV(B2:D2)	

Using the ReDim Preserve Statement

As stated previously the **Preserve** part of the **ReDim** statement prevents the loss of data from the re-dimensioned array. The use of **Preserve** imposes two major limitations on the use of **ReDim**.

- The inability to change the lower boundary of the index.
- The inability to change the number of dimensions.

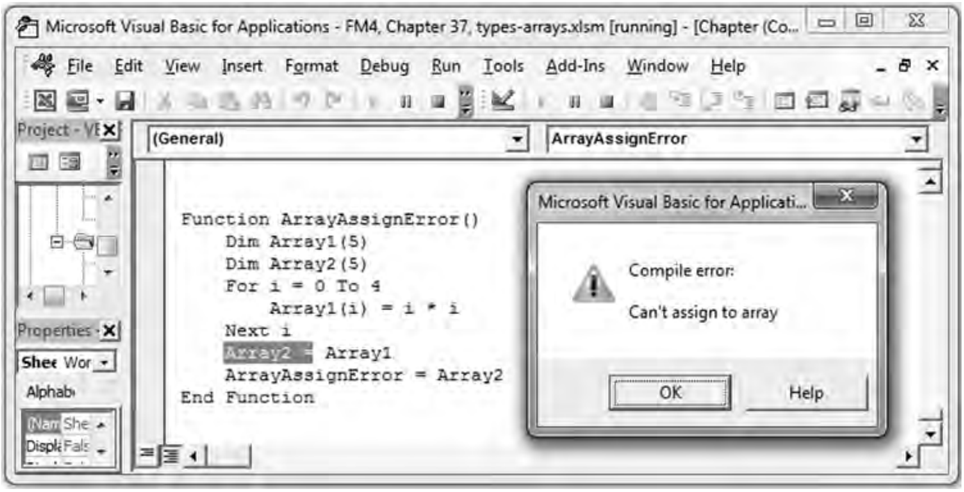
The main use of the **ReDim Preserve** is in interactive programs and as such it will be demonstrated in a later chapter dealing with user interaction.

37.7 Array Assignment

Here's an error that's easy to make: In the following example we want to tell VBA that **Array2** is equal to **Array1**:

```
Function ArrayAssignError()
    Dim Array1(5)
    Dim Array2(5)
    For i = 0 To 4
        Array1(i) = i * i
    Next i
    Array2 = Array1
    ArrayAssignError = Array2
End Function
```

VBA doesn't allow this, as you can see on the next screen shot.



Obviously one way to assign arrays is to assign each element separately using a **For** loop.

```
For I = 0 To 4: Array2(I) = Array1(I):
Next I
```

The : Operator

Note the use of the "**:**" operator to signal the end of a statement. This way we can put two or more short statements on the same line. Another, much shorter, way of assigning arrays is discussed in the next section.

37.8 Variants Containing an Array

A **Variant** type variable can contain an array. The procedure is somewhat more complicated than the declaration of a normal array, but the reward in terms of assignment is sometimes worth the inconvenience. The following function demonstrates the use of a **Variant** containing an array:

```
Function ArrayAssign(r, j)
   Dim Array1                         ‘This is a variant
   Dim Array2                         ‘This is a variant
   Dim n                              ‘number of elements
                                      ‘in R
   Array1 = Array()
   If r.Columns.Count = 1 Then        ‘data in column
      n = r.Rows.Count
   ElseIf r.Rows.Count = 1 Then       ‘data in row
      n = r.Columns.Count
   Else                               ‘invalid data
      Exit Function
   End If
   ReDim Array1(1 To n)
   For i = 1 To n
      Array1(i) = r(i)
   Next i
   ‘****************************Watch this spot
   Array2 = Array1                   ‘Watch this spot
   ‘****************************Watch this spot
   If j >= 1 And j <= n Then
      ArrayAssign = Array2(j)
   End If
End Function
```

The **Array()** function (on the fifth line) returns a **Variant** containing an array. The assignment on the same line makes **Array1** into an array (not initialized at the moment). The **ReDim** statement on line 15 makes **Array1** into an **n** element array. The reward for all our trouble is illustrated on line 18. Here is the function in a worksheet context:

	A	B	C	D	E
1	ARRAYASSIGN IN HORIZONTAL ACTION				
2	55	88	77	12	99
3	1	55	<-- =ArrayAssign($A$2:$E$2,A3)		
4	2	88	<-- =ArrayAssign($A$2:$E$2,A4)		
5	3	77	<-- =ArrayAssign($A$2:$E$2,A5)		
6	4	12	<-- =ArrayAssign($A$2:$E$2,A6)		
7	5	99	<-- =ArrayAssign($A$2:$E$2,A7)		
8	6	0	<-- =ArrayAssign($A$2:$E$2,A8)		

37.9 Arrays as Parameters to Functions

Arrays can be used as parameters to functions. The following set of functions presents an improved version of **DynPV** discussed in section 37.6. Notice how much easier it is to read the main function **NewDynPV**, when all the auxiliary tasks are relegated to separate functions.

A function **ComputePV(CF())** is used to compute the present value of a series of cash flows contained in an array of **Doubles**.

```
Function ComputePV(CF())
    Temp = 0
    For i = LBound(CF) To UBound(CF)
        Temp = Temp + CF(i) / 1.05 ^ i
    Next i
    ComputePV = Temp
End Function
```

Note the fact that in **ComputePV(CF())**, **CF()** has to be declared without index information. Consequently, we use **LBound** and **UBound** to get index information.

The function **GetN(R As Range)** returns the number of elements in R:

```
Function GetN(R As Range)
   If R.Columns.Count = 1 Then  'data in column
      GetN = R.Rows.Count
   ElseIf R.Rows.Count = 1 Then 'data in row
      GetN = R.Columns.Count
   Else
      GetN=0
   End If
End Function
```

Here is the main function:

```
Function NewDynPV(R As Range)
   Dim n As Integer        ' Number of periods
   Dim CF() As Double      ' Dynamic array for  cash flows
   n = GetN(R)
   If (n=0) Then
      NewDynPV = n
      Exit Function
   End If
   ReDim CF(1 To n)        ' re-dimension the array
   For i = 1 To n
      CF(i) = R(i)
   Next i
   NewDynPV = ComputePV(CF)
End Function
```

	A	B	C	D
1	**NEWDYNPV IN ACTION**			
2	**Cash Flows ▶**	100	200	300
3	**NewDynPV ▶**	535.79527	<-- =newDynPV(B2:D2)	

Better IRR and NPV Functions

We can now revisit **nIRR** and **nNPV** from section 37.4, and try and make them faster using internal arrays.

```
Function fNPV(Rate, cf)
    Temp = 0
    For i = LBound(cf, 2) + 1 To UBound(cf, 2)
        Temp = Temp + cf(1, i) / (1 + Rate) ^ _
          (i - 1)
    Next i
    fNPV = Temp + cf(1, LBound(cf, 2))
End Function

Function fIRR(R, Optional guess = 0.5)
   cf = R.Value
   n = fNPV(guess, cf)
   Do While Abs(npv) > 0.0001
      If n < 0 Then
         guess = guess - guess / 2
      Else
         guess = guess + guess / 2
      End If
      n = fNPV(guess, cf)
   Loop
   fIRR = guess
End Function
```

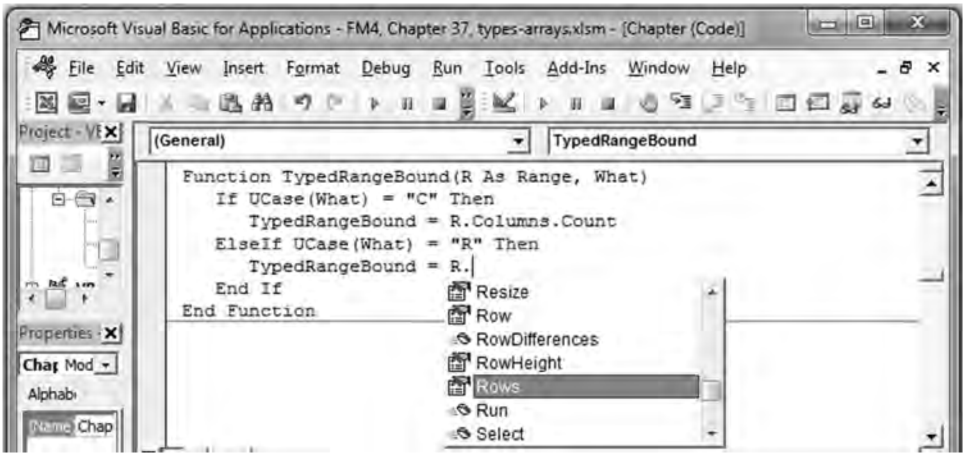
	A	B	C	D	E	F
1	**FIRR IN ACTION**					
2	Cash Flows ▶	-375	100	100	100	100
3	Guess ▶	5%				
4	IRR ▶	2.63247%	<-- =IRR($B$2:$F$2,$B$3)			
5	fIRR ▶	2.63247%	<-- =fIRR($B$2:$F$2,$B$3)			
6	fIRR ▶	2.63248%	<-- =fIRR($B$2:$F$2)			
7	nNPV ▶	7.70797E-11	<-- =nNPV($B$4,$B$2:$F$2)			
8	nNPV ▶	9.79546E-06	<-- =nNPV($B$5,$B$2:$F$2)			
9	nNPV ▶	-9.54928E-05	<-- =nNPV($B$6,$B$2:$F$2)			

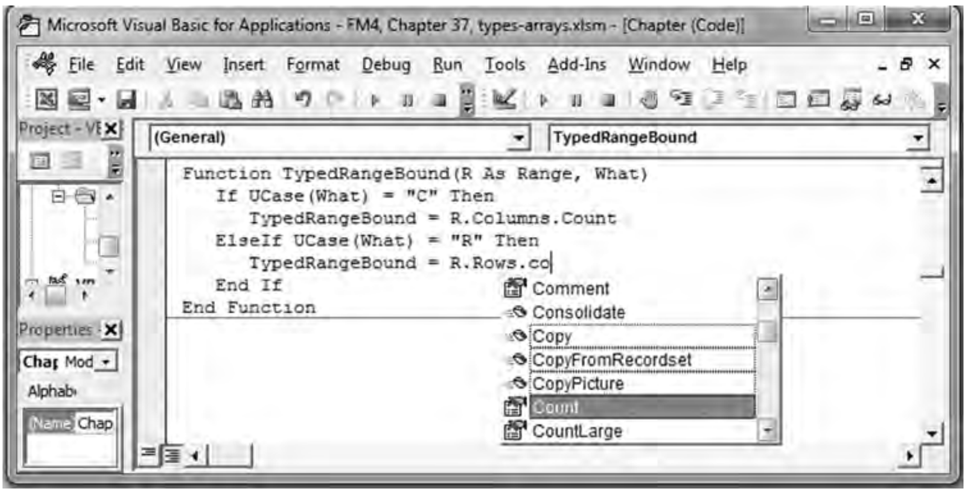
This works an order of magnitude faster but can be further improved.

37.10 Using Types

All values, variables, and functions in VBA are categorized into types, either by default or explicitly. By default all variables and functions in VBA are of the type **Variant**. **Variant** is a category of values (type) that includes all other categories. In most cases we can just ignore the type, but sometimes it can be very useful to give a variable a type other than **Variant**. Variable types allow VBA to give us information about the variable as we use it, and this will become apparent as we start using Excel objects in the following chapters. For now we will just explain the mechanics of defining a typed variable, and provide a short demonstration of the help offered by VBA when dealing with typed variables. A type is given to a variable when it is defined by following the variables name with the word **As** followed with a type name. For example, the statement “Dim x As Integer” defines a variable named x of the type Integer.

To demonstrate the usefulness of typed variables recall the function **RangeBound** from section 37.4. Here is a new version with the first parameter explicitly given the type Range. Although not necessary (it worked without it), it does make life easier. When you type in the function and a variable has a type, VBA can give you hints as you try to use properties. In our example, as soon as you type the period after R, VBA provides a list of possible Properties or Methods. If the selected property (Rows, in our case) has properties of its own, then a period following its name will produce a list for you to choose from.





37.11 Summary

VBA functions use variables to store information. Variables can hold all sorts of information. Declaring and using variables that can hold only a specific type of information (**Typed Variables**) can make your programming task easier and your programs more readable, and use less computer memory.

An array is a group of variables of the same type, sharing the same name and referenced individually using one or more indices. In VBA an array index is an integer. By default the index of the first element in an array is 0; this can be changed to 1 for all arrays used in a module by using the **Option Base 1** statement. The size and number of dimensions of an array are set at the time the array is declared and have to be known when the program is written. Dynamic arrays are arrays whose size (but not number of dimensions) can be set at run time.

Exercises

1. Write a function **NewPV(CF, r)** which calculates the present value of a given cash flow **CF** at interest rate *r* for 5 periods:

$$NewPV(CF,r)=\frac{CF}{(1+r)^1}+\frac{CF}{(1+r)^2}+\frac{CF}{(1+r)^3}+\frac{CF}{(1+r)^4}+\frac{CF}{(1+r)^5}$$

	A	B	C	D
1	**NEWPV IN ACTION**			
2	**CF**	**r**	**NewPV**	
3	100.0000	10%	379.0787	<-- =NewPV(A3,B3)
4	50.0000	10%	189.5393	<-- =NewPV(A4,B4)
5	100.0000	1%	485.3431	<-- =NewPV(A5,B5)
6	50.0000	1%	242.6716	<-- =NewPV(A6,B6)

2. Rewrite the function in exercise 2 as **BetterNewPV(CF, r, n),** so it could deal with *n* periods.

	A	B	C	D	E
1	**BETTERNEWPV IN ACTION**				
2	**CF**	**r**	**n**	**BetterNewPV**	
3	100.0000	5%	5	432.9477	<-- =BetterNewPV(A3,B3,C3)
4	50.0000	10%	5	189.5393	<-- =BetterNewPV(A4,B4,C4)
5	100.0000	1%	10	947.1305	<-- =BetterNewPV(A5,B5,C5)
6	50.0000	1%	10	473.5652	<-- =BetterNewPV(A6,B6,C6)

3. A bank offers different interest rates on loans. The rate is based on the size of the periodical repayment (**CF**) and the following table. Rewrite the function in exercise 2 as **BankPV(CF, r, n)** so that it reflects the present value of a loan in the bank.

For Periodical Repayments <=	The Interest Rate Is
100.00	*r*
500.00	*r* – 0.5%
1,000.00	*r* – 1.1%
5,000.00	*r* – 1.7%
1,000,000.00	*r* – 2.1%

	A	B	C	D	E
1	**BANKPV IN ACTION**				
2	**CF**	**r**	**n**	**BankPV**	
3	-1	5%	5	E	<-- =BankPV(A3,B3,C3)
4	100.00	5%	5	432.95	<-- =BankPV(A4,B4,C4)
5	100.01	5%	5	439.04	<-- =BankPV(A5,B5,C5)
6	1000.00	5%	5	4464.36	<-- =BankPV(A6,B6,C6)
7	1000.01	5%	5	4540.79	<-- =BankPV(A7,B7,C7)
8	5000.00	5%	5	22703.71	<-- =BankPV(A8,B8,C8)
9	5000.01	5%	5	22964.11	<-- =BankPV(A9,B9,C9)

4. A bank offers different interest rates on deposit accounts. The rate is based on the size of the periodical deposit (**CF**) and the following table. Write a future value function **BankFV(CF, r, n)**.

For Periodical Deposits	The Interest Rate Is
<=100.00	*r*
<=500.00	*r* + 0.5%
<=1,000.00	*r* + 1.1%
<=5,000.00	*r* + 1.7%
>5,000.00	*r* + 2.1%

	A	B	C	D	E
1	**BANKFV IN ACTION**				
2	**CF**	**r**	**n**	**BankFV**	
3	**-1**	5%	5	E	<-- =Bankfv(A3,B3,C3)
4	100.00	5%	5	580.19	<-- =Bankfv(A4,B4,C4)
5	100.01	5%	5	588.86	<-- =Bankfv(A5,B5,C5)
6	1,000.00	5%	5	5992.91	<-- =Bankfv(A6,B6,C6)
7	1,000.01	5%	5	6099.47	<-- =Bankfv(A7,B7,C7)
8	5,000.00	5%	5	30497.07	<-- =Bankfv(A8,B8,C8)
9	5,000.01	5%	5	30856.78	<-- =Bankfv(A9,B9,C9)

5. Another bank offers 1% increase in interest rate to savings accounts with a balance of more than 10,000.00. Write a future value function **Bank1FV(CF, r, n)** that reflects this policy.

	A	B	C	D	E
1	**BANK1FV IN ACTION**				
2	**CF**	**r**	**n**	**Bank1FV**	
3	**-1**	5%	5	E	<-- =Bank1FV(A3,B3,C3)
4	9999.00	5%	5	59620.97	<-- =Bank1FV(A4,B4,C4)
5	10000.00	5%	5	59626.94	<-- =Bank1FV(A5,B5,C5)
6	10001.00	5%	5	59759.16	<-- =Bank1FV(A6,B6,C6)
7				5.96	<-- =D5-D4
8				132.22	<-- =D6-D5

6. The bank in exercise 5 changed its bonus policy and now offers the interest rate increase based on the following table. Rewrite **Bank1FV(CF, r, n)** to reflect this change.

Balance	Interest Rate
<=1,000.00	$r + 0.2\%$
<=5,000.00	$r + 0.5\%$
<=10,000.00	$r + 1.0\%$
>10,000.00	$r + 1.3\%$

	A	B	C	D	E
1	**BANK2FV IN ACTION**				
2	**CF**	**r**	**n**	**Bank2FV**	
3	**-1**	5%	5	E	<-- =Bank2FV(A3,B3,C3)
4	9999.00	5%	5	60237.93	<-- =Bank2FV(A4,B4,C4)
5	10000.00	5%	5	60243.96	<-- =Bank2FV(A5,B5,C5)
6	10001.00	5%	5	60288.29	<-- =Bank2FV(A6,B6,C6)

7. Write a version of the present value function with two interest rates, one for positive cash flows and another for negative cash flows. The function should be written for use in a worksheet, and accept both column and row ranges as parameters. The function declaration line should be:

```
Function MyPV(CF As Variant, PositiveR As Double, _
  NegativeR As Double) As Double
```

	A	B	C	D	E	F	G
1	**MYPV IN ACTION**						
2	PositiveR	5%	100	100	100	272.3248	<-- =MyPV(C2:E2,$B$2,$B$3)
3	NegativeR	10%	-100	-100	-100	-248.6852	<-- =MyPV(C3:E3,$B$2,$B$3)
4			-100	100	100	86.17762	<-- =MyPV(C4:E4,$B$2,$B$3)
5			-63	<-- =MyPV(C2:C4,$B$2,$B$3)			

8. Write a future value version of the function in exercise 7.

9. A bank offers different interest rates on loans. The rate is based on the size of the periodical repayment (**CF**$_i$) and the following table. Write a present value function **BankPV(CF, r)** so that it reflects the present value of a loan in the bank. The function should be useable as a worksheet function. **CF** could be either a row range or a column range.

For Periodical Repayments <=	The Interest Rate Is
100.00	r
500.00	$r - 0.5\%$
1,000.00	$r - 1.1\%$
5,000.00	$r - 1.7\%$
1,000,000.00	$r - 2.1\%$

10. A bank offers different interest rates on deposit accounts. The rate is based on the size of the periodical deposit (**CF_i**) and the following table. Write a future value function **BankFV(CF, r)**. The function should be useable as a worksheet function. **CF** could be either a row range or a column range.

For Periodical Deposits	The Interest Rate Is
<=100.00	*r*
<=500.00	*r* + 0.5%
<=1,000.00	*r* + 1.1%
<=5,000.00	*r* + 1.7%
>5,000.00	*r* + 2.1%

11. Another bank offers 1% increase in interest rate to savings accounts with a balance of more than 10,000.00. Write a future value function **Bank1FV(CF, r)** that reflects this policy. The function should be useable as a worksheet function. **CF** could be either a row range or a column range.

38 Subroutines and User Interaction

38.1 Overview

A subroutine is a VBA user routine used to automate routine or repetitive operations in Excel. Subroutines are sometimes called macros. Modules and module variables are introduced as the last subject of this chapter.

38.2 Subroutines

A subroutine looks like a function, but the word “Sub” replaces the word **Function** in the definition. The parentheses following the subroutine name are blank (recall that the parentheses following the function name give the function’s parameters). Separating the first and last line are the statements that the subroutine executes. The following is a very simple subroutine that puts a message on the screen:

```
Sub SayHi()
    MsgBox "Hi", , "I say Hi"
End Sub
```

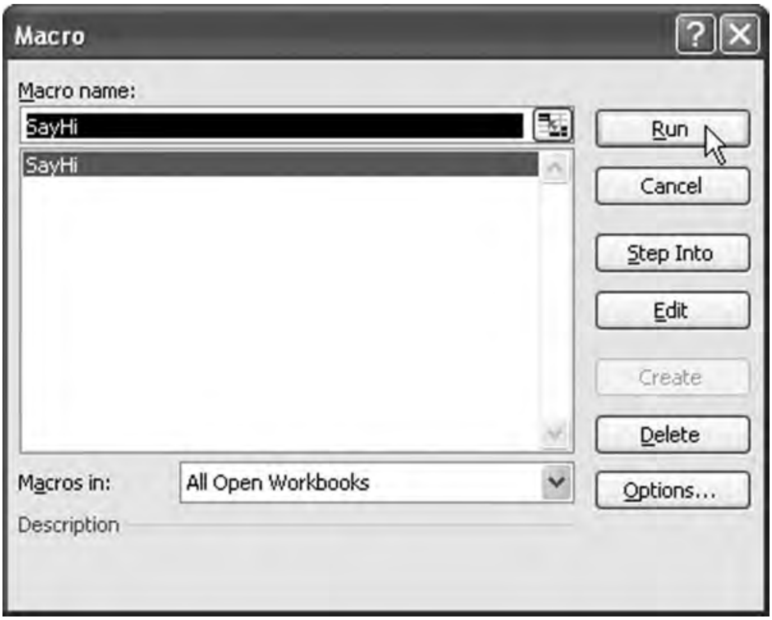
The above subroutine introduces a built-in VBA subroutine called **MsgBox**. It also introduces the way one subroutine is activated (called) from another. **MsgBox** is named as a command on a line followed by its list of arguments separated by commas. Notice the syntax:

```
MsgBox "Hi", , "I say Hi"
```

The commas separate the three arguments of every **MsgBox**:

- “Hi” is the message which will be displayed.
- The second argument is empty: Notice the space between the commas. This argument can be used to define buttons for the message box. This topic is discussed in section 38.3.
- The third argument is “I say Hi”—this is the message box title.

A subroutine can be activated (run) from an Excel worksheet in various ways. The simplest way of running a subroutine is from the **Macros** button on the **Developer** tab on the **Ribbon** or by using the keyboard shortcut [Alt] + F8. Either way, the macro selection box appears.[1] The box lists all available subroutines alphabetically. Find our subroutine, click on its name, and click the **Run** button.



And this is what you will see:

1. If you don't have the **Developer** tab, go to **File|Options|Customize the Ribbon**. In the right side of the resulting screen, mark the **Developer** box.

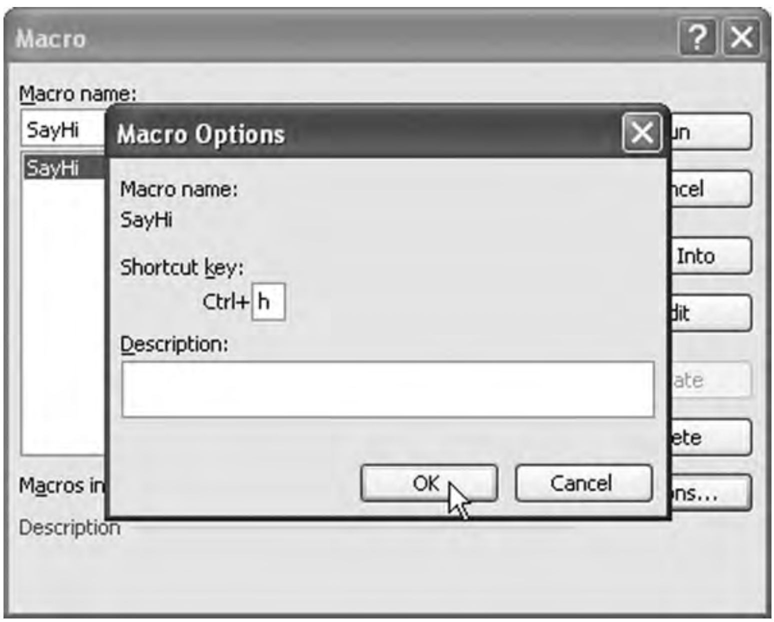
At this point Excel is locked up—you have to click the **OK** button before you can proceed.

Keyboard Shortcut for Subroutines

Using a keyboard shortcut is a faster way to make a subroutine run. To attach a shortcut to our subroutine:

- Select the **Options** button from the macro selection box.
- Type a character in the provided space and click **OK**.
- Close the macro selection box using the corner **X**.

You can now activate the subroutine using the shortcut ([Ctrl] + h, in our case).



Recording a Subroutine

One easy way to start writing subroutines is to record the sequence of actions you want in the subroutine, and then edit the resulting subroutine to produce the final results you need. Something we do a lot in this book is to insert the function **Getformula** in a cell to the right of the cell with the interesting formula. Let's record a subroutine that performs this action (see Figures 38.1–38.3).

1. Select the cell to the right of the cell with the formula we are interested in (B4, in our case)
2. Select the **Developer** tab on the ribbon.
3. In the code group click **Use Relative References**. This causes Excel to record relative cell addresses in the subroutine rather than its default, which is to record the actual cell addresses.
4. Click **Record Macro**.
5. At this point you have the option to name the subroutine as well as all sorts of other options. Since we are going to change most of these options in the near future, we ignore all options and finally click **OK** to start recording.
6. Type in the formula **=Getformula(A4)**.
7. Click **Stop Recording** (this is the same button that used to be the **Record Macro** button).

RECORDING A SUBROUTINE—SCREENS

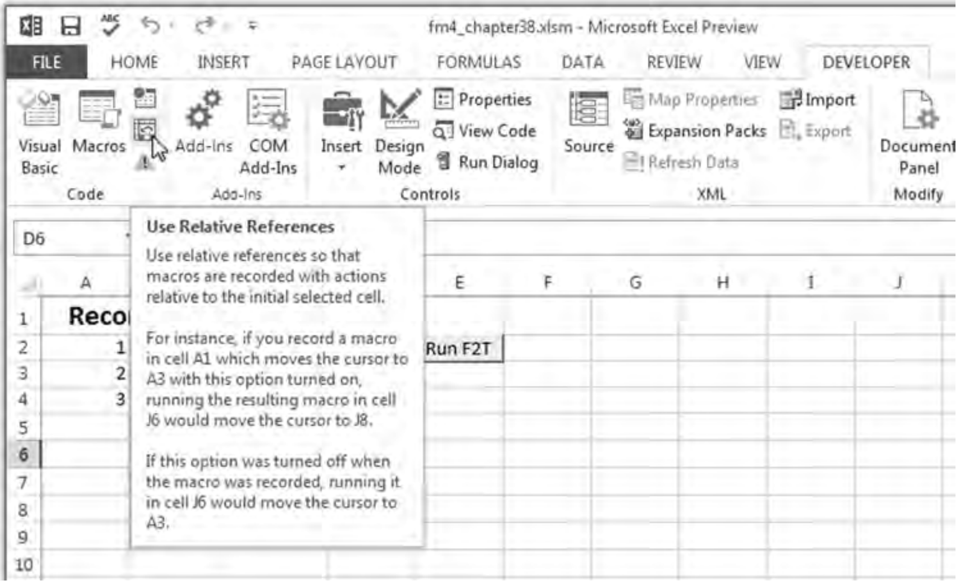


Figure 38.1
Indicating recording with relative references.

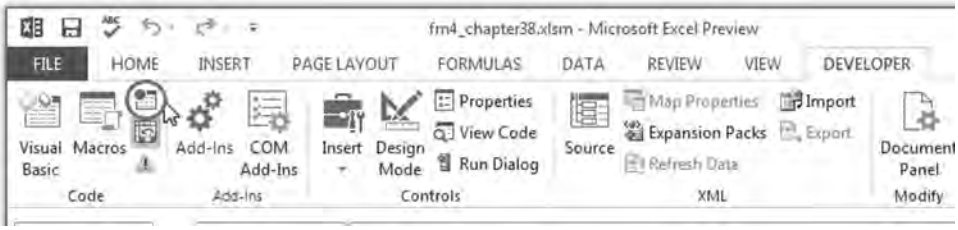


Figure 38.2
Start recording the subroutine.

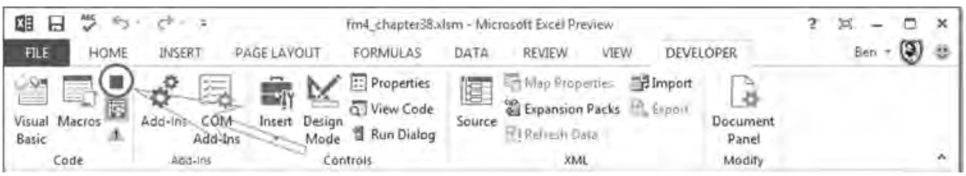


Figure 38.3
End subroutine recording. Button is in the same place as the **Record Macro** button.

If we now go to the VBA editor we can see that a new module has been added to the workbook. The module contains **Macro1**:

```
Sub Macro1()
    ActiveCell.FormulaR1C1 = _
    "=getformula(RC[-1])"
End Sub
```

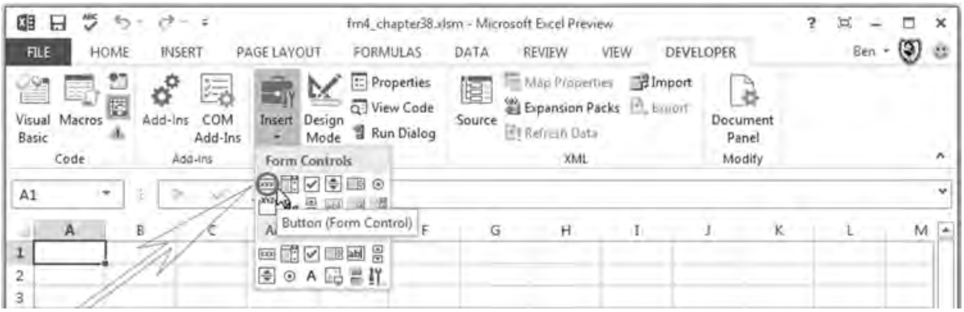
We can add some bells and whistles to this recorded subroutine. In the recorded subroutine below, we changed the name and put in a line to prevent the accidental overwriting of a non-blank cell.

```
Sub RecordGetformula()
' Puts in Getformula, points to cell
' to the left
   If IsEmpty(ActiveCell) Then
   ActiveCell.FormulaR1C1 = _
   "=getformula(RC[-1])"
   End If
End Sub
```

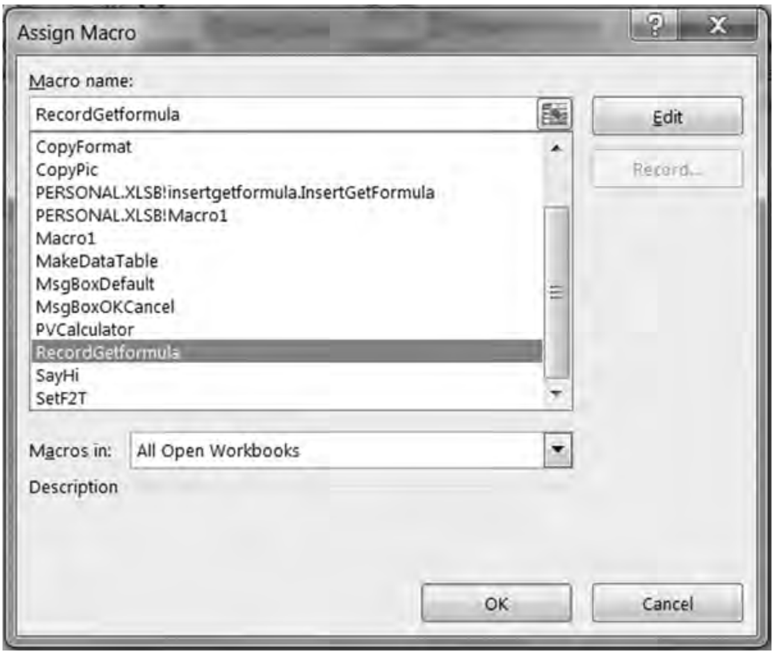
Running a Subroutine from a Button on the Worksheet

Instead of running the subroutine from the **Developer** tab or running it from a key combination, we can also insert a button on the worksheet to run a subroutine on that worksheet. To illustrate, we insert a button that runs the subroutine **RecordGetformula**:

1. Select the **Developer** tab on the ribbon.
2. In the **Controls** group, click **Insert**.
3. From the **Form Controls,** select **Button**.

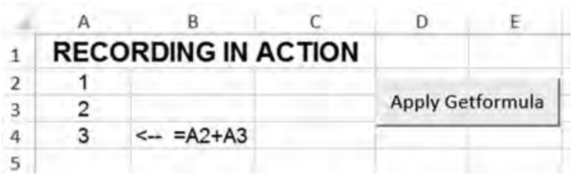


4. Drag the crosshair on the sheet to draw the button.
5. Once you release the mouse button, the **Assign Macro** dialogue will appear.



6. Select our subroutine and click **OK**.

7. The button should be selected (control handles all round) and you can now edit the text that appears on the button. If the button is not selected, right-clicking on it will open a local menu enabling you to change the text or the subroutine assigned to the button.



38.3 User Interaction

In this section we show how to use subroutines to elicit data from the user of the spreadsheet. We illustrate with the **MsgBox** command, which (as discussed above) displays a message on the screen and returns a value based on the button clicked. Some of the different options available with this function are demonstrated in the following subroutines:

```
Sub MsgBoxDefault()
    Dim Temp As Integer
    Temp = MsgBox("Default Message", , _
    "Default Title")
    MsgBox _
    "The value returned by MsgBox is: " _
    & Temp
End Sub
```

Note: The default configuration of **MsgBox** produces one **OK** button. The default title is “Microsoft Excel.” Clicking the **OK** button makes **MsgBox** return the value 1.

```
Sub MsgBoxOKCancel()
    Dim Temp As Integer
    Temp = MsgBox("Default Message", _
    vbOKCancel)
    MsgBox _
    "The value returned by MsgBox is: " _
    & Temp
End Sub
```

As previously noted, the second argument to **MsgBox** determines which buttons are displayed. This incarnation of the demo subroutine uses the constant **vbOKCancel** to produce the two buttons **OK** and **Cancel**. Note that if the **Cancel** button is clicked, **MsgBox** returns the value 2.

InputBox: Getting Data from the User

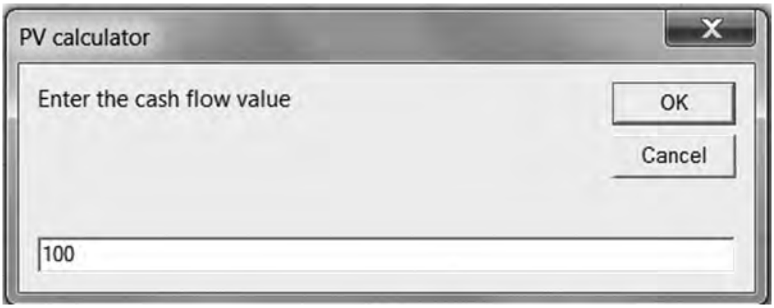
InputBox is an internal VBA function used to get textual information from the user into a variable in a subroutine. The workings of the function are demonstrated in the following present value calculator subroutine. The subroutine **PVCalculator** calculates $\sum_{t=1}^{10} \frac{CF}{(1.05)^t}$, where *CF* is a number inputted by the user:

```
Sub PVCalculator()
    Dim CF
    CF = InputBox("Enter the cash flow value", _
        "PV calculator", "100")
    MsgBox "The present value of 4" & CF & _
        "At 5% for 10 periods is: " & _
        Round(Application.PV(0.05, 10, -CF), _
        2), vbInformation, "PV calculator"
End Sub
```

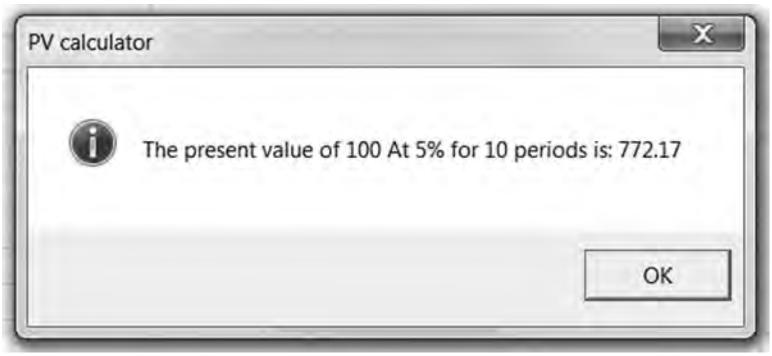
Note the syntax:

```
CF = InputBox("Enter the cash flow value", _
     "PV calculator", "100")
```

- "Enter … please," the first argument in **InputBox**, is the message to display.
- "PV calculator," the second argument, is the title for the box.
- "100," the third argument, is the default string to place in the box. If you do not replace this by some other value, this will also be the returned value from the function.
- Running the subroutine should result in the following:



At this point you can replace “100” by some other number (in this example, we’ve chosen to leave it). Clicking on the **OK** button results in the following box:

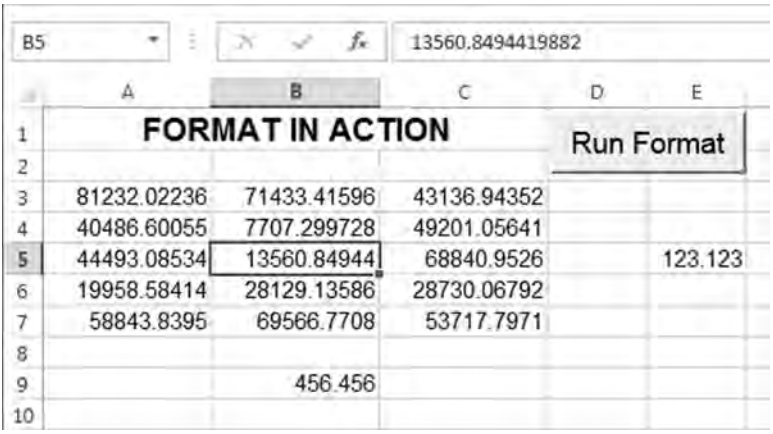


38.4 Using Subroutines to Change the Excel Workbook

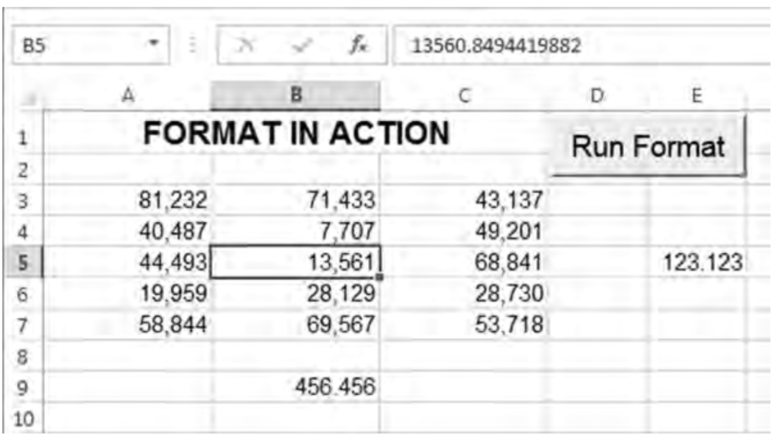
Subroutines can be used to make changes to a spreadsheet. Here’s a small example, very similar to the example presented at the end of Chapter 35. In this version of the subroutine we change the current region’s format to numbers with comma separators and without decimals.

```
Sub Format()
   ActiveCell.CurrentRegion.NumberFormat _
   = "#,##0"
End Sub
```

ActiveCell.CurrentRegion is the range around the active cell (B5 in the screen snaps), the same range that would be selected by pressing [Ctrl] + A in the worksheet (A3:C7 in the screen shots).

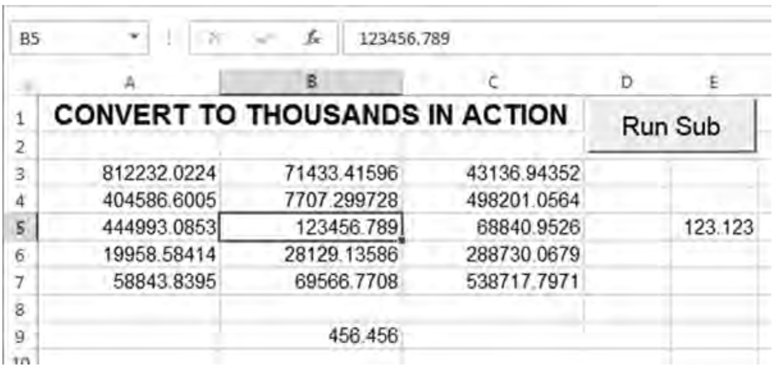


After:

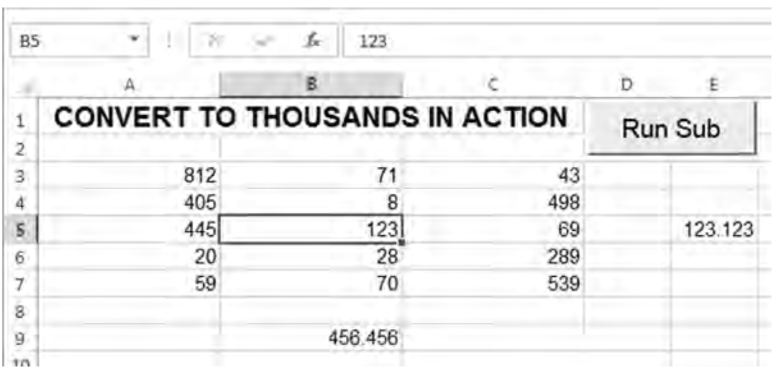


The next subroutine changes the actual data in **ActiveCell.CurrentRegion** to thousands by dividing each number in the range by 1,000 and rounding it to the nearest integer.

```
Sub ConvertToThousands()
    s = ActiveCell.CurrentRegion.Cells.Count
    For i = 1 To s
        ActiveCell.CurrentRegion(i).Value = _
        Round(ActiveCell.CurrentRegion(i). _
        Value / 1000, 0)
    Next i
End Sub
```



After:



38.5 Modules

VBA organizes user-defined functions and subroutines in units called modules. We can (and sometimes should) have more than one module in a VBA project (i.e., the part of the workbook that has our functions and subroutines). Modules have names: By default VBA uses the name "Module" followed by a number to indicate the module's name, but you might find it useful to give them a somewhat more descriptive name.

To rename a module (in the VBA Editor), select the module on the **Project Explorer** pane:

If the **Project Explorer** pane is not visible, select **Project Explorer** from the **View** menu.

Once a module is selected, the module's list of properties should appear in the **Properties Pane**. If the **Properties Pane** is not visible, select **Properties Window** from the **View** menu. Click on the module's name (it should be the only property available) and change it. A module name should start with an alphabetic character and consist of only alphabetic characters, digits, and the underscore character (_); no other characters should be used.

Once you tap the Enter key, the name is changed. Notice the change in the Project Explorer.

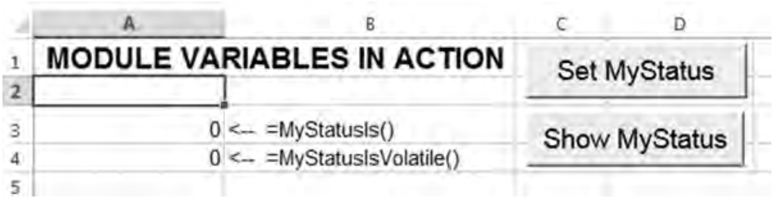
Modules must have unique names, and they cannot be named after subroutines and functions. If a module called Tom has a function called Tom in it, the function Tom will not be available to the workbook. One common practice is to start module names (and only module names) with M.

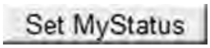
Module Variables

The **Dim** statement can be used before any routine in the module to define a module variable. Module variables are recognized anywhere in the module and keep their value until the workbook is closed. Module variables can be used to store information relevant to more than one routine without the need to pass the information via the parameters. Module variables are more commonly used in large modules with many interacting routines, so the following demonstration is, of necessity, somewhat trivial:

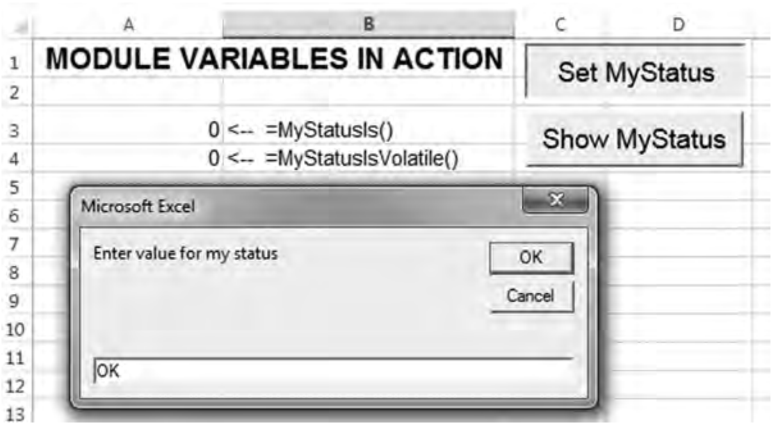
```
Dim MyStatus
Sub SetMyStatus()
   MyStatus = InputBox _
   ("Enter value for my status", , "OK")
    Calculate
End Sub
Function MyStatusIs()
    MyStatusIs = MyStatus
End Function
Sub ShowMyStatus()
    MsgBox "MyStatus is: " & MyStatus
End Sub
Function MyStatusIsVolatile()
    Application.Volatile
    MyStatusIsVolatile = MyStatus
End Function
```

When you first open the workbook, here is what you see:

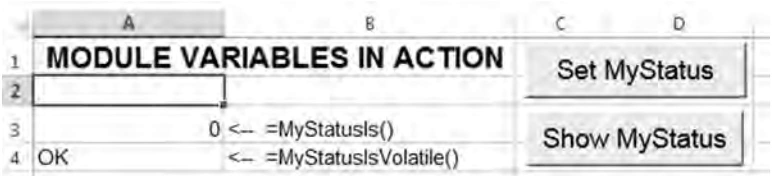


If you click on 

you get the **Input Box**:



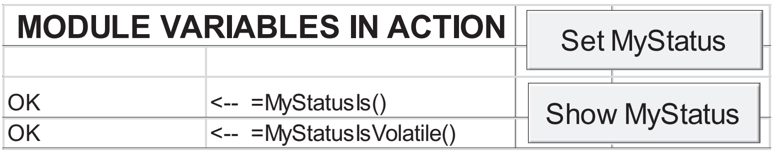
And a click on the OK button will produce the following:



We now know that the variable **MyStatus** has the value “OK.” So why is the function **MyStatusIs** returning a zero, or for that matter, why is **MyStatusIsVolatile** returning the (correct) value of “OK”?

Application.Volatile

The answer to the question above lies with the **Application.Volatile** statement in **MyStatusIsVolatile**. When **Application.Volatile** is used as the first statement in a function used in a worksheet, the function gets recalculated whenever something gets recalculated on the worksheet. **MyStatusIs** will only recalculate if its (nonexisting) parameter changes, in this case only if we edit the cell and press Enter. So if we if we select cell A3, press F2 (for Edit Cell), and press Enter we get



38.6 Summary

A subroutine is a VBA user routine used to automate routine or repetitive operations in Excel. VBA provides two important and very flexible functions for user interaction: **MsgBox** and **InputBox**. VBA groups subroutines and functions into units called modules; keeping related functions and subroutines grouped is useful when dealing with large projects. All of these topics, explored in this chapter, will help you with financial programming in Excel.

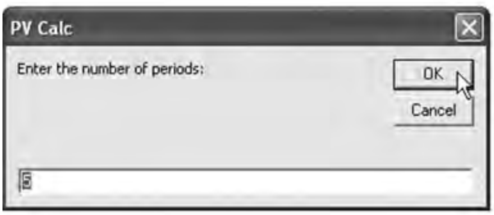
Exercises

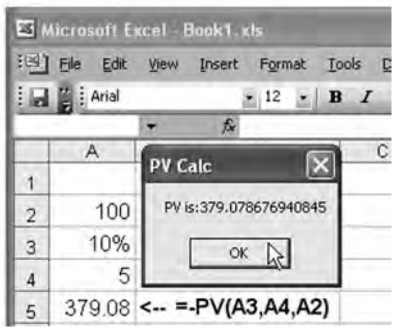
1. Write a subroutine that displays the following message box. The message box should be on top of all other windows, and prevent the user from doing anything in any application, until one of the buttons is clicked.

   Hint: You need to use some options of **MsgBox** that were not covered in the text, use the VBA help system.

2. Write a present value calculator subroutine similar to the one which appears in section 38.4. However—as illustrated below—your subroutine should ask the user for the cash flow value, the interest rate, and the number of periods. It should then display the result in a message box. Sensible default values should be supplied for all arguments. Do not use the Excel function **PV**; write your own present value function and use it. A reminder:

$$PV(CF, r, n) = \sum_{i=1}^{n} \frac{CF}{(1+r)^i}$$



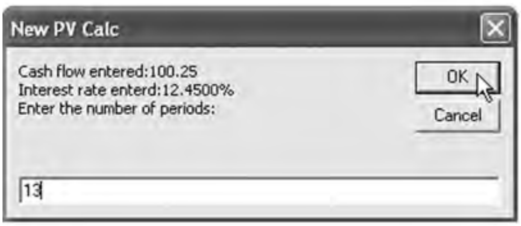


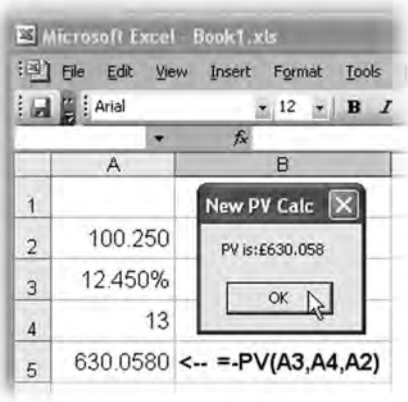
You can use the **PV** function provided by Excel, as we did, to verify the correctness of your subroutine.

3. Rewrite the subroutine in the previous exercise so that the user interface is as demonstrated in the following screen shots. Some of the functions needed to write the subroutine were not covered in the text. We used the following functions:

   - **Val**—A function used to convert a string of digits to a number.
   - **Left**—A function used to return the left part of a string.
   - **Right**—A function used to return the right part of a string.
   - **FormatPercent**—A function used to format a number.
   - **FormatCurrency**—A function used to format a number.

   More information about these functions is available from the VBA Help file. We recommend you use it.





Note: Your computer might display a different currency symbol.

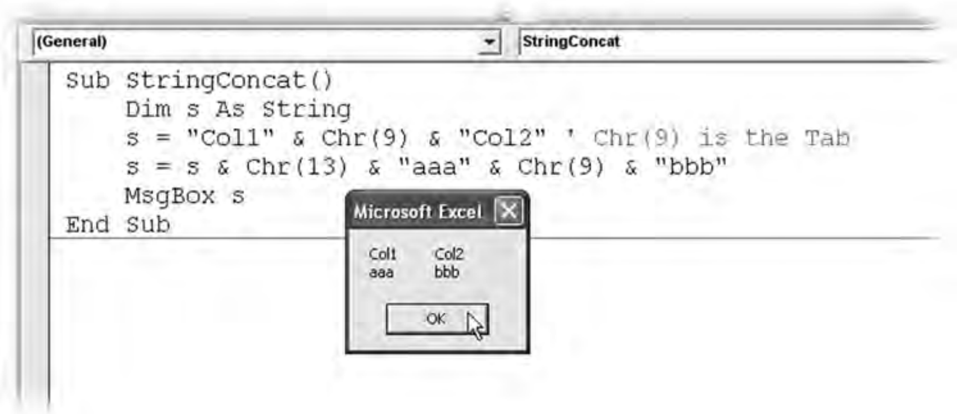
4. Rewrite the subroutine in the previous exercise so it deals properly with the **Cancel** button.

   - A simple version of the new subroutine will abort the subroutine if **Cancel** is clicked in any stage.
   - A more sophisticated version of the new subroutine will allow the user to reenter the data from scratch.
   - The most sophisticated version of the new subroutine will allow reentering the data using the old data as a default.

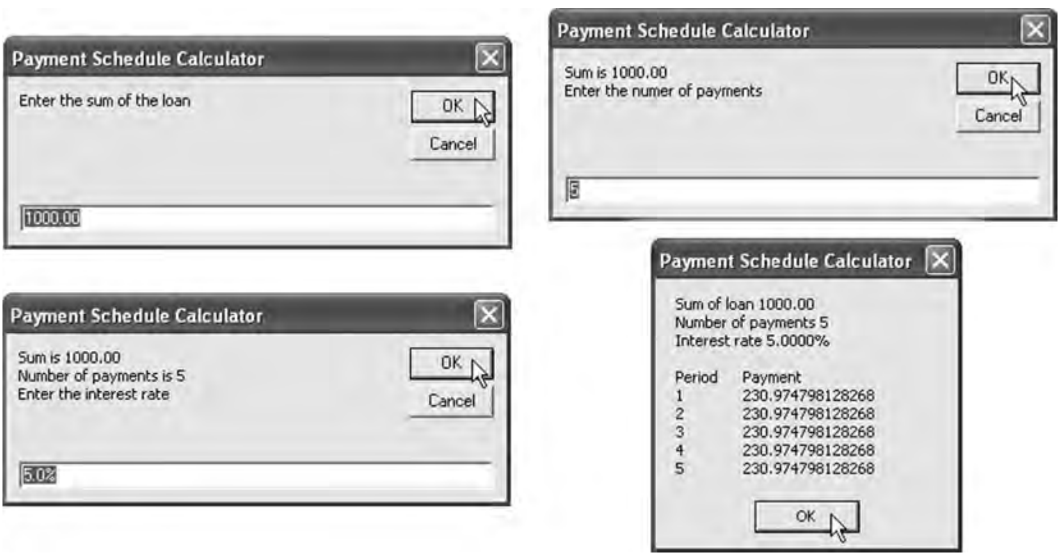
   Note: The last version is a slightly more complicated exercise using loops within loops.

5. Write a payment schedule calculator subroutine. The subroutine is to ask the user for the sum of the loan, the number of payments, and the interest rate. Assume payment at the end of the period. The output should look like the example below.
Hints:

- You may want to use the worksheet function **PMT**.
- The following subroutine and its output might be of interest:

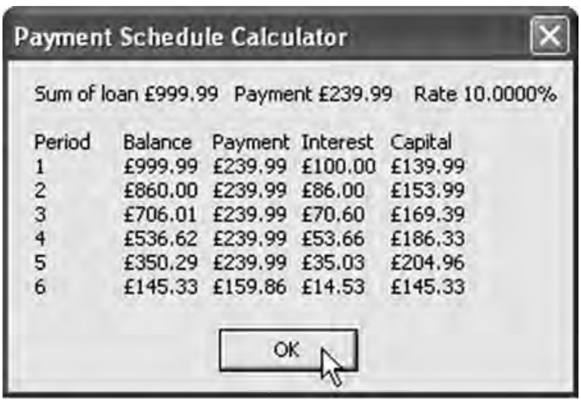


Here is an example of the requested subroutine in action:

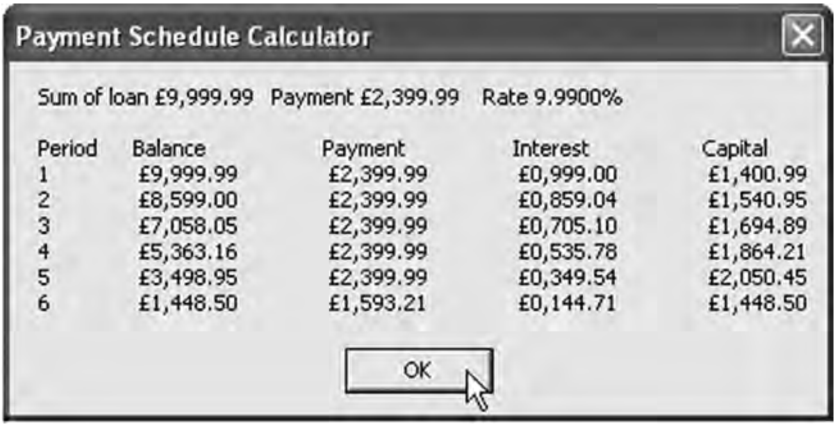


6. Rewrite the payment schedule calculator subroutine so it displays the payments broken down into interest and capital payments. The input boxes in the example were removed for compactness.

7. Write a payment schedule calculator subroutine. The subroutine is to ask the user for the sum of the loan, the payment, and the interest rate. Assume payment at the end of the period. The subroutine should display the payments broken down into interest and capital payments. Obviously, the last payment can be smaller (but not larger) than the payment supplied by the user. The output should look like the example below (input boxes removed for compactness):



8. A somewhat more complicated version of the subroutine in exercise 7 would produce the following, better-looking results. Write this version of the subroutine. *Note*: A quick look at the Help file for the **Format** function might be advantageous at this point.

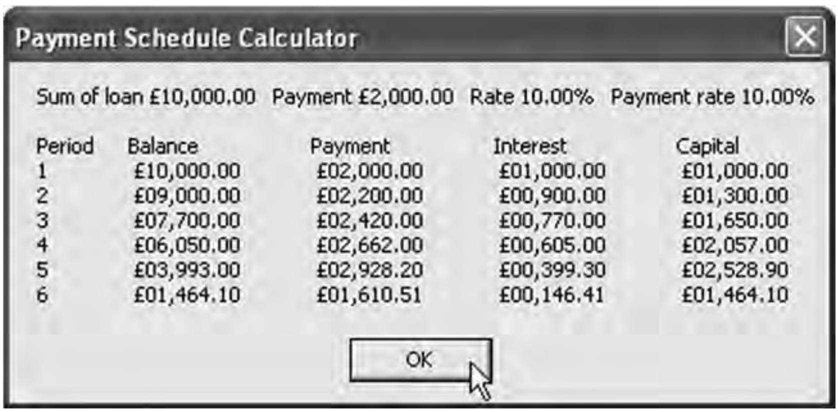

Payment Schedule Calculator

Sum of loan £9,999.99 Payment £2,399.99 Rate 9.9900%

Period	Balance	Payment	Interest	Capital
1	£9,999.99	£2,399.99	£0,999.00	£1,400.99
2	£8,599.00	£2,399.99	£0,859.04	£1,540.95
3	£7,058.05	£2,399.99	£0,705.10	£1,694.89
4	£5,363.16	£2,399.99	£0,535.78	£1,864.21
5	£3,498.95	£2,399.99	£0,349.54	£2,050.45
6	£1,448.50	£1,593.21	£0,144.71	£1,448.50

OK

9. A sliding payment schedule involves payment that changes by a fixed percentage over the life of the loan. Write a sliding payment version of the payment schedule calculator in exercise 8. In addition to all the inputs described above, the subroutine will get a payment rate of change (as percentage) from the user. This is what it should look like in action:


Payment Schedule Calculator

Sum of loan £10,000.00 Payment £2,000.00 Rate 10.00% Payment rate 10.00%

Period	Balance	Payment	Interest	Capital
1	£10,000.00	£02,000.00	£01,000.00	£01,000.00
2	£09,000.00	£02,200.00	£00,900.00	£01,300.00
3	£07,700.00	£02,420.00	£00,770.00	£01,650.00
4	£06,050.00	£02,662.00	£00,605.00	£02,057.00
5	£03,993.00	£02,928.20	£00,399.30	£02,528.90
6	£01,464.10	£01,610.51	£00,146.41	£01,464.10

OK

39 Objects and Add-Ins

39.1 Overview

This chapter deals with several more advanced subjects in VBA. Most of these subjects relate to the Excel Object Model. The bulk of the chapter describes some useful Excel objects and ways of dealing with them. Names, a way to make worksheets clearer and more readable, are presented in section 39.6. The chapter closes with a discussion of Excel Add-Ins, one easy way to make self-crafted functions automatically available across workbooks.

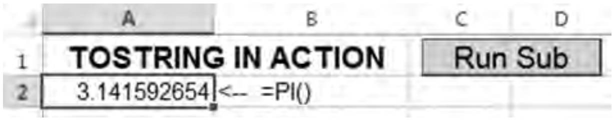
39.2 Introduction to Worksheet Objects

Objects are the basic building blocks of VBA. Although you may not be aware that you are using objects, most things you do in VBA require the manipulation of objects. We can think of an object as a sort of a container with variables, functions, and subroutines inside. All of Excel's components (workbooks, worksheets, ranges, etc.) are represented by an object in the VBA Object Hierarchy. The object's data are held in special variables called properties that can be accessed using the Dot (**.**) operator. One of the most important object types in VBA is the **Range Object**. A worksheet cell and a range of cells are all objects of the type **Range**. The following subsection introduces some predefined **Range Object** variables.

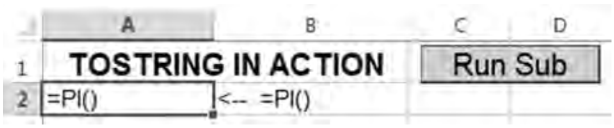
The Active Cell

VBA has many variables predefined for our use; one of the more useful is `ActiveCell`. `ActiveCell` is a predefined **Range Object** variable that represents the cell in the worksheet with the cursor box around it. The following function replaces the contents of the active cell with a string representation of the contents. We use the property `Formula`; this property holds the text in the cell as a string and can be changed.

```
Sub ToString()
   ActiveCell.Formula = "'" & _
   ActiveCell.Formula
End Sub
```



Before



After

The Selection

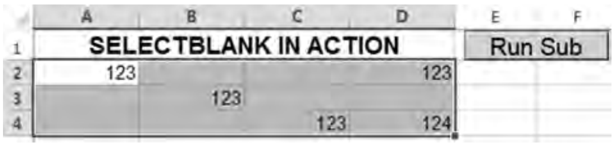
Another very useful predefined variable is the `Selection`. This variable represents the currently selected item in Excel. Unlike the `ActiveCell` variable, `Selection` is not limited to a range and can be any selection (range, chart, and many more). We suggest that you check the type using the `TypeName` function. Methods are functions contained within an object. Methods are used to manipulate the object. Like properties, methods can be accessed using the Dot **(.)** operator. The line between methods and properties is sometimes very fuzzy. The following subroutine demonstrates some methods of the **Range Object**, and use of the predefined variable `Selection`:

```
Sub SelectBlank()
   If UCase(TypeName(Selection)) <> "RANGE" _
   Then Exit Sub
   Selection.SpecialCells(xlCellTypeBlanks). _
   Select
End Sub
```

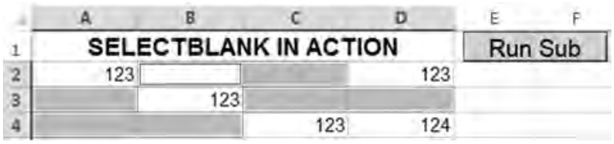
The first line checks to see if the current selection is a range and stops the subroutine if it is not; a message would have been appropriate under non-educational circumstances.

The first part of the second line `Selection.SpecialCells (xlCellTypeBlanks)` uses the `SpecialCells` method of the **Range Object** to return a `Range` containing all the blank cells in the current selection.

The `Select` method of the returned `Range` is activated to select it.



Before



After

39.3 The Range Object

In the previous section we encountered some predefined **Range Object** variables. This section demonstrates the use of ranges in VBA and presents more of the properties and methods of the **Range Object**.

A Range as a Parameter to a Function

In this subsection we build a function that accepts a `Range` as a parameter. Our new function, named `MeanReturn`, accepts a column range of asset prices as a parameter and computes and returns the mean return of the assets in the column. Recall that the return of an asset for period t is $r_t = \dfrac{Price_t - Price_{t-1}}{Price_{t-1}}$ and the mean return of an asset is $\overline{r} = \dfrac{1}{N}\sum_{t=1}^{N} r_t$. An auxiliary function `AssetReturn` is used to compute r_t.

```
Function MeanReturn(Rng)
    NumRows = Rng.Rows.Count
    Prices = Rng.Value
    T = 0
    For i = 2 To NumRows
        T = T + AssetReturn(Prices(i - 1, 1), _
        Prices(i, 1))
    Next i
    MeanReturn = T / (NumRows - 1)
End Function
```

Lines of note:

- `NumRows = Rng.Rows.Count`

In this line the `Dot` operator is used twice. `Rng` is our `Range` object. `Rows` is property of the `Range` object so `Rng.Rows` is an object of the `Collection` type that represents all the rows in our range. `Count` is a property of `Collection` type objects that stores the number of members in the collection, so `Rng.Rows.Count` is a variable that stores the number of rows in our range.

- `Prices = Rng.Value`

`Value` is a property of the `Range` object containing the values of all the cells in the range. `Value` is of the type `Variant`. If the range is more than one cell in size `Value` is a two-dimensional array. The first index of `Value` is the row index starting from 1, and the second index is the column index staring from 1.

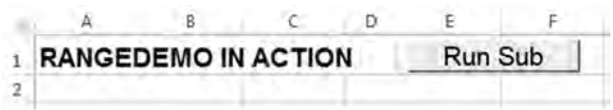
	A	B	C
1	MEANRETURN IN ACTION		
2	100		
3	110	10.00%	<-- =(A3-A2)/A2
4	121	10.00%	<-- =(A4-A3)/A3
5	145	19.83%	<-- =(A5-A4)/A4
6	174	20.00%	<-- =(A6-A5)/A5
7		14.96%	<-- =AVERAGE(B3:B6)
8		14.96%	<-- =MeanReturn(A2:A6)

The Range Property

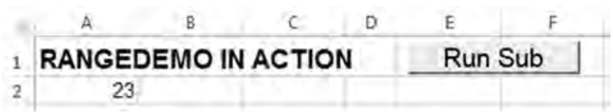
The `Range` property is one way to access a range on a worksheet. `Range` is a property of many Excel objects. When used on its own, as in the next subroutine, `Range` is a short way of writing `ActiveSheet.Range`.

```
Sub RangeDemo()
    Range("A2").Formula = 23
End Sub
```

As expected, the subroutine will set the formula in cell A2 of the active worksheet to 23.



Before



After

The next subroutine sets the formula of each cell in the range A2:C3 of the active worksheet to 23.

```
Sub RangeDemo1()
    Range("A2:C3").Formula = 23
End Sub
```

Another way of addressing a range of cells using the `Range` property is demonstrated by the next subroutine. The subroutine sets the formula of each cell in the range A2:C3 of the active worksheet to 23. The first argument to `Range` is the cell in the top left corner of the range, and the second is the cell in the bottom right corner of the range.

```
Sub RangeDemo2()
    Range("A2", "C3").Formula = 23
End Sub
```

`Range` is also a property of the `Range` object. The range returned by `Range` when used this way is relative to the `Range` object. The next subroutine sets the formula of the cell C3 of the active worksheet to 999.

```
Sub RangeDemo3()
    Range("B2").Range("B2").Formula = 999
End Sub
```

Note: `Range("B2")` returns the range (or cell) B2 of the active worksheet. `Range("B2").Range("B2")` returns the cell B2 of the range that has B2 as the top left corner. In worksheet terms, `Range("B2").Range("B2")` returns the cell C3.

The next subroutine sets the formula of each cell in the range C2:D3 of the active worksheet to 23. The subroutine uses the cell C2 as a starting point.

```
Sub RangeDemo4()
    Range("C2").Range("A1", "B2").Formula = 23
End Sub
```

Note: `Range("C2")` is the same as `Range("C2").Range("A1")` and refers to the cell C2 in the worksheet. `Range("C2").Range("B2")` refers to the cell D3 in the worksheet, B2 means one column to the left and one line down. And so `Range("C2").Range("A1", "B2")` is the same as `Range("C2", "D3")`.

39.4 The With Statement

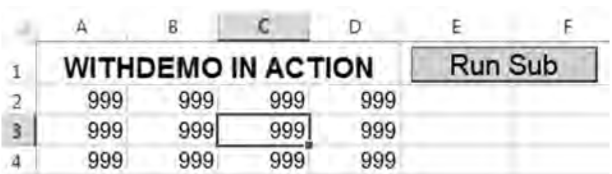
The `With` statement allows you to perform a series of statements on a specified object without restating the obvious (the object's name and its pedigree, which can be very long). If you have more than one property to change or more than one method to use for a single object, use the `With` statement. `With` statements make your procedures run faster, and help you avoid repetitive typing. The following, somewhat contrived, subroutine sets some properties of the font of the cell in the top left-hand corner of the current region of the active cell. The font is set to be Arial, bold, and 15 points in size.

```
Sub WithoutDemo()
    ActiveCell.CurrentRegion.Range("A1"). _
    Font.Bold = True
    ActiveCell.CurrentRegion.Range("A1"). _
    Font.Name = "Arial"
    ActiveCell.CurrentRegion.Range("A1"). _
    Font.size = 15
End Sub
```

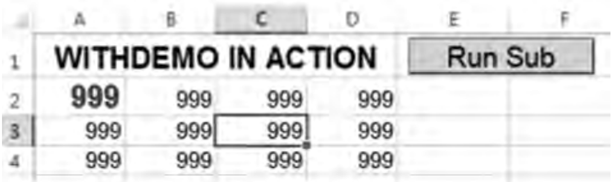
And here is the same subroutine using the `With` statement:

```
Sub WithDemo()
    With ActiveCell.CurrentRegion. _
    Range("A1").Font
        .Bold = True
        .Name = "Arial"
        .size = 15
    End With
End Sub
```

Notice the **Dot (.)** operator before the properties in the `With` statement. Recall from Chapter 38 that `ActiveCell.CurrentRegion` is the contiguous range of non-empty cells around the active cell (C3 in the screen shots), the same range that would be selected by pressing [Ctrl] + A in the worksheet (A1:D4 in the screen shots).



Before



After

39.5 Collections

A `Collection` is a set of items that can be referred to as a unit. Members can be added using the `Add` method and removed using the `Remove` method. Specific members can be referred to using an integer index. The number of members currently in a `Collection` is available via the `Count` method. Our use of `Collections` will be restricted to using the (quite numerous) arsenal of `Collections` that are part of the Excel Object Model, to name but a few, Range is a collection of cells, Worksheets is a collection of all the worksheets in a workbook, and Workbooks is a collection of all the open workbooks in Excel.

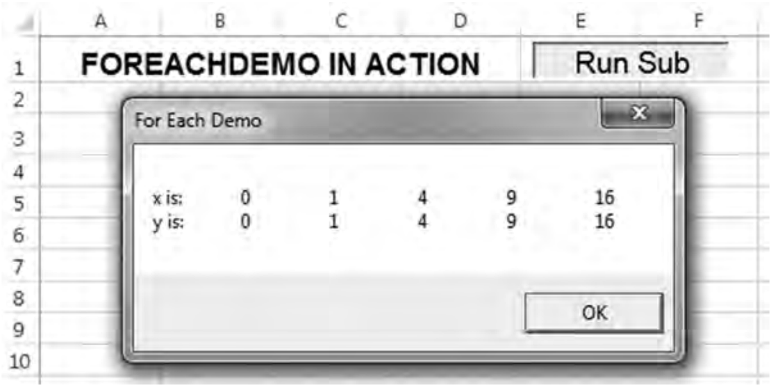
The For Each Statement in Use with Arrays and Collections

The `For Each` statement is a variation of the `For` loop. This statement comes in two distinct flavors. The first variation uses the statement to loop over a VBA array as demonstrated in the following subroutine:

```
Sub ForEachDemo()
    Dim A(4)
    For i = 0 To 4: A(i) = i * i: Next i
    x = "x is: "
    y = "y is: "
    For Each Element In A
        x = x & vbTab & Element
        Element = Element * 2
    Next Element
    For Each Element In A
        y = y & vbTab & Element
    Next Element
    MsgBox x & vbCrLf & y, , "For Each Demo"
End Sub
```

Points of note:

1. The current member of the array is available to the statements within the loop body through the loop variable (`Element` in the above function).
2. The loop variable (`Element` in the above example) has to be of the type `Variant` irrespective of the array type.
3. Changes to `Element` will **not** be reflected in the actual array. Notice that the changes to `Element` in line 8 are not reflected in *y*.
4. You don't need to know the number of dimensions or the range of indices to loop over the array.

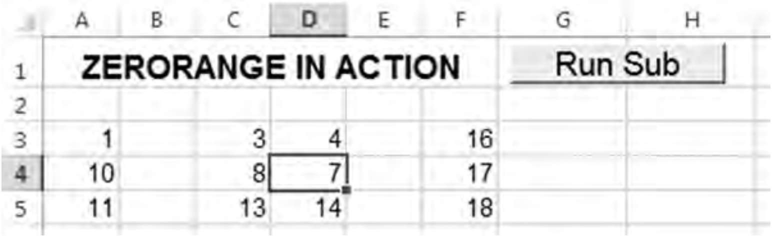


The For Each Statement in Use with Collections

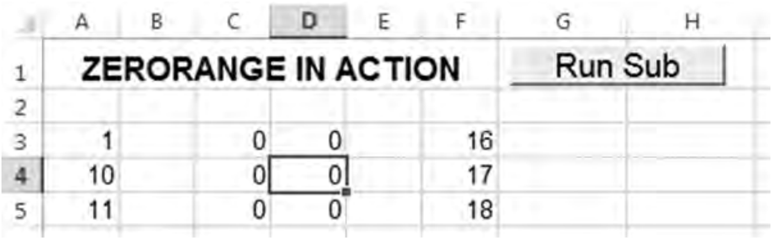
The second version of the `For Each` statement loops over `Collections`:

```
Sub ZeroRange()
    Set Rng = ActiveCell.CurrentRegion
    For Each Cell In Rng
        Cell.Formula = 0
    Next Cell
End Sub
```

Here is what happens when you run the subroutine:



Before



After

Points of note:

1. Ranges are collections and so the variable **`Rng`** is a collection of all the cells in the current region of the active cell (C3:D15 in our example).

2. **`Cell`** is a variable used to iterate over **all** the members of the collection.

3. **`Cell`** has to be one of the following types: `Variant`, `Object`, or the specific type of element the `Collection` is made of. (Recall that all variables are of type **Variant** unless specifically defined.)

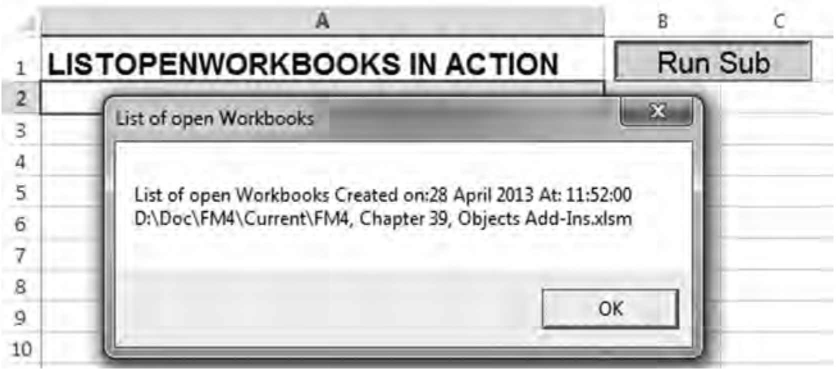
4. **`Cell`** refers to the actual member of the `Collection`, and changes to `Cell` will be reflected in the `Collection`.

5. A complete explanation of the use of the `Set` statement is beyond the scope of this book. For our purposes, just prefix the reserved word `Set` to all object assignments.

The Workbooks Collection and the Workbook Object

All the currently open workbooks are represented by a **Workbook** object in the **Workbooks Collection**. The following subroutine lists all open workbooks:

```
Sub ListOpenWorkbooks()
    Temp = "List of open Workbooks" & _
        " Created on:" & FormatDateTime(Date, _
        vbLongDate) _
        & " At: " & FormatDateTime(Time, _
        vbLongTime)
    For Each Element In Workbooks
        Temp = Temp & vbCrLf & Element.FullName
    Next Element
    MsgBox Temp, vbOKOnly, "List of open _
    Workbooks"
End Sub
```



Lines of note:

```
Temp = "List of open Workbooks" & _
       " Created on:" & _
       FormatDateTime(Date, vbLongDate) _
       & " At: " & _
       FormatDateTime(Time, vbLongTime)
```

- The `Date` function returns the current system date.
- The `Time` function returns the current system time.
- The `FormatDateTime` function formats `Date` and `Time` variables for display.

```
For Each Element In Workbooks
    Temp = Temp & vbCrLf & Element.FullName
Next Element
```

The `For` statement loops over the entire `Workbooks Collection`. On each iteration, `Element` is one of the `Workbook` objects in the `Collection`. `FullName` is a property of the `Workbook` object containing the full path name of the workbook.

The Worksheets Collection and the Worksheet Object

All the worksheets in a workbook are `Worksheet` objects in the `Worksheets Collection` that is a property of the `Workbook` object. We can use the `Worksheets Collection` without an object as a short form for `ActiveWorkbook.Worksheets`.

39.6 Names

In Excel you can use user-defined names to refer to a cell or a range of cells. Use easy-to-understand names, such as Products, to refer to hard-to-understand ranges, such as `Sales!C20:C30`. Using names can make formulas easy to read: Compare the formula `=sum('sheet12'!a10:a10)` to `=sum(lastYearSales)`. This section deals with the VBA side of names.

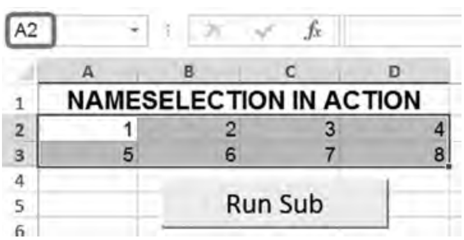
Naming a Range Using a Subroutine

The following subroutine gives the name "Jon" to the cells selected.

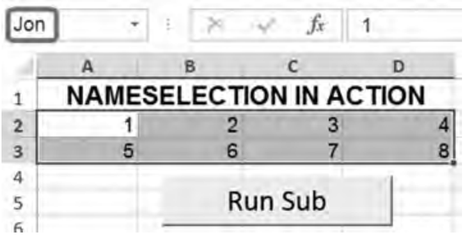
```
Sub NameSelection()
    Names.Add "Jon", "=" & _
    Selection.Address
End Sub
```

Select cells A2:B3 and run the subroutine. The next two screenshots show the Excel name box before and after we operate the NameSelection subroutine:

Before



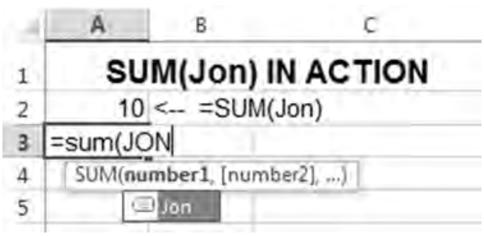
After



`Names` is a `Collection` of all the names in the active workbook. `Add` is a method of the `Names Collection` used to add members to the collection. We use only the first two parameters of the method. The first parameter `"Jon"` is the name to add to the `Names Collection`. The second parameter is a string containing the address, formula, or value to which the added name refers, preceded by =.

Looking for Defined Names

The name "Jon" has just now been defined, and we can use it in the workbook as demonstrated in the following screen shot. Notice that whatever capitalization you use, Excel reverts to the original "Jon."



The name "Jon" is not directly available in VBA as demonstrated by the following function:

```
Function SumJon()
SumJon = Application.WorksheetFunction. _
Sum(Jon)
End Function
```

	A	B	C
1	SUM(Jon) IN ACTION		
2	10	<-- =SUM(Jon)	
3	0	<-- =SumJon()	

Referring to a Named Range

To get to values in a named range we can use the built-in function `Application.Evaluate`, as demonstrated by the next function. Note that the function is designed to be used as an **Array Function** and the use of `Application.Volatile` to make sure the value gets updated whenever a change is made to the workbook.

```
Function JonAsArray()
    Application.Volatile
    JonAsArray = Application.Evaluate("Jon")
End Function
```

	A	B	C
1	JonASARRAY IN ACTION		
2	1	3	<-- {=JonAsArray()}
3	2	4	<-- {=JonAsArray()}

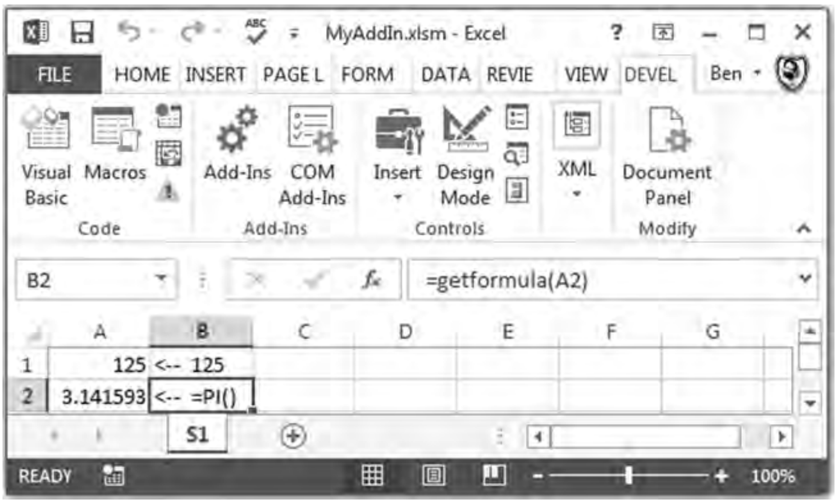
Referring to the actual range the name refers to is beyond the scope of this book.

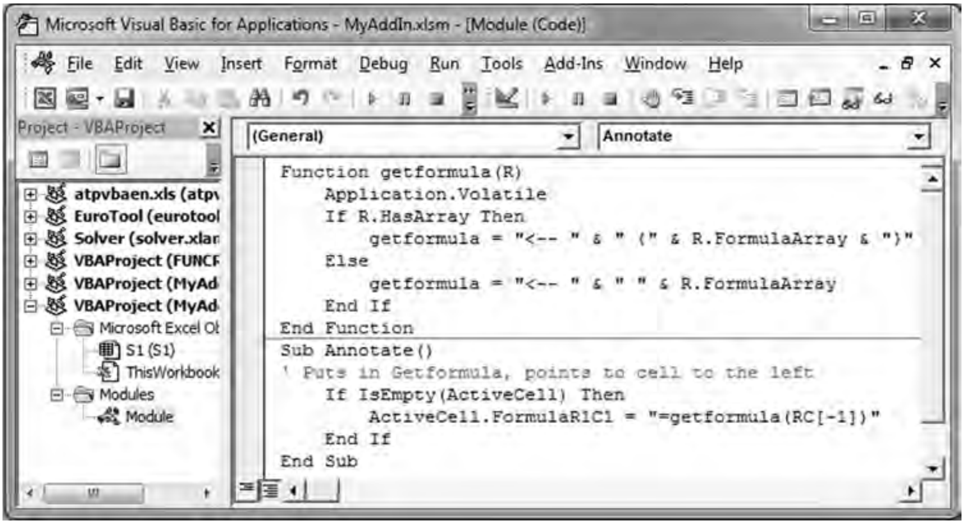
39.7 Add-Ins and Integration

An Excel Add-In is a file that Excel can load when it starts up. The file contains VBA code that adds additional functionality to Excel, usually in the form of new functions. Add-Ins provide an excellent way of increasing the power of Excel and they are the ideal vehicle for distributing your custom functions. This section shows you how to convert an Excel Workbook containing VBA functions to an Add-In, and how to load and use Add-Ins in Excel and VBA. The process is somewhat arcane and the steps below should be followed in the order in which they are presented.

Create and Debug Your Base Workbook

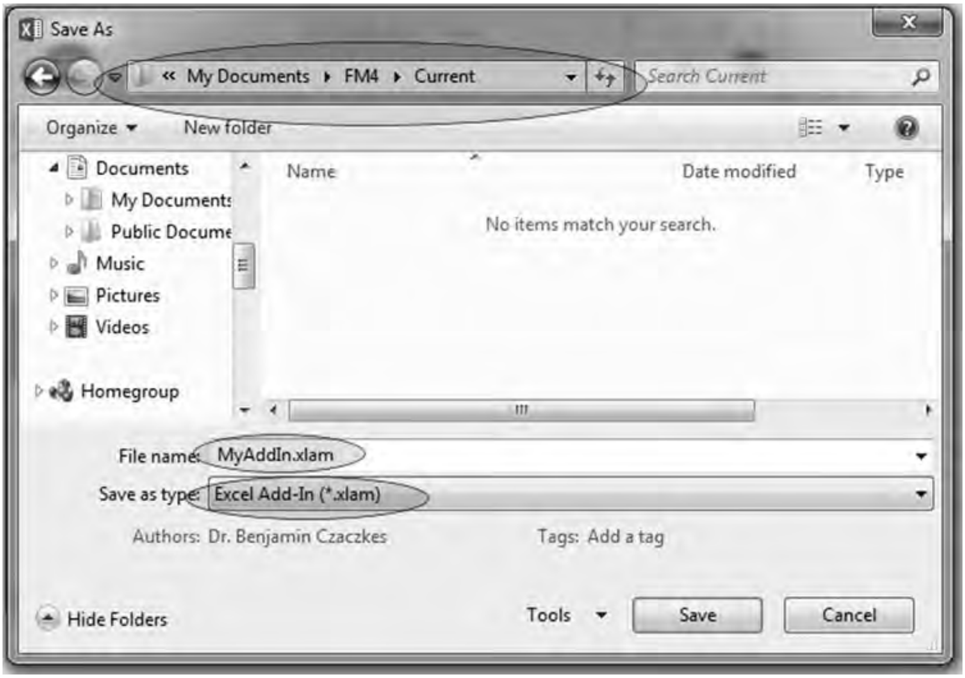
As editing an Add-In once it's created is very difficult, it is important that the original workbook on which the Add-In is based is kept intact and as a workbook. For this demonstration we have created a workbook containing one worksheet and a VBA project containing one module with one function and one subroutine.





Convert the Base Workbook to an Add-In

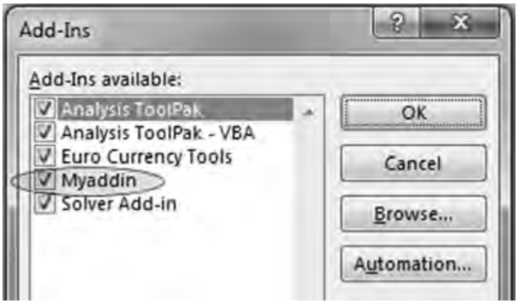
To make the Add-In, save the workbook as an Add-In. Select **Save As** from the Excel file menu and change **Save as type** to "Excel Add-In (*.xlam)." The **Save in** location will change to the Add-Ins directory on your computer. You may want to navigate to a different location (we tend to keep files together). Now click **Save**. You may want to use a new name for the Add-In (we did not).



Install and Use an Add-In from an Excel Worksheet

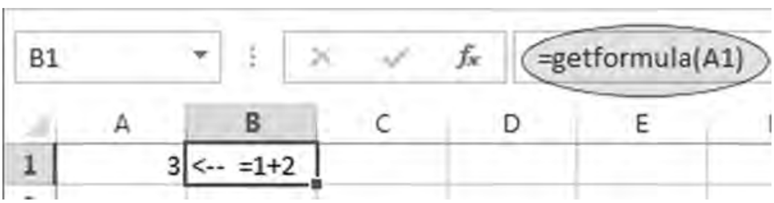
Installing an Add-In is done on a per computer basis (actually per computer user basis). So we do not get confused with the chapter we suggest you close Excel and reopen it with a brand new Excel workbook. Select **Developer|Add-Ins**. The following dialogue should be presented (your names may vary):

Click **Browse** and navigate to the location of your Add-In. Select it and click **OK**.



Notice that a new Add-In is available and activated. Click **OK** to close the Add-Ins dialogue. All the functions in our Add-in are now available to all workbooks in Excel. To verify insert a formula in a cell (A1), select the cell next to the cell with the formula (B1), and press [Alt] + F8.

You should get the **Macro** dialogue box. Notice that, sadly, annotate is not on the list, but if you type “annotate” in the **Macro name** box, then the **Run** button will become available and when pressed will produce the expected results.



39.8 Summary

This chapter discussed two separate topics. We started with a more extensive discussion of objects, which underlie the VBA programming concept. Objects allow you to be much more parsimonious in expressing your programming references. We finished the chapter with a discussion of how to build Add-Ins in Excel.

Exercises

1. Suppose you have a spreadsheet with a series of numbers and formulas:

	A	B	C
1	EX1	Run Sub	
2			
3	Price	Return	
4	1000.00		
5	1069.45	0.069451061	<-- =A5/A4-1
6	1158.53	0.083296895	<-- =A6/A5-1
7	1213.49	0.047440948	<-- =A7/A6-1
8	1269.56	0.046204665	<-- =A8/A7-1
9	1287.02	0.013745727	<-- =A9/A8-1
10	1316.34	0.022788538	<-- =A10/A9-1
11	1332.09	0.01195957	<-- =A11/A10-1

Suppose you want to turn this into:

	A	B	C
1	EX1	Run Sub	
2			
3	Price	Return	
4	1000.00		
5	1069.45	6.945106119	<-- =(A5/A4-1)*100
6	1158.53	8.329689467	<-- =(A6/A5-1)*100
7	1213.49	4.744094796	<-- =(A7/A6-1)*100
8	1269.56	4.620466489	<-- =(A8/A7-1)*100
9	1287.02	1.374572664	<-- =(A9/A8-1)*100
10	1316.34	2.278853849	<-- =(A10/A9-1)*100
11	1332.09	1.195956963	<-- =(A11/A10-1)*100

Write a subroutine that does this. Your subroutine should:

- Put in a set of parentheses and multiply the cell contents by 100.
- Move down one cell (see **ActiveCellDemo1**, section 39.1).
- Ask if you want to repeat the process (if "yes," it should do it; if "no," the subroutine should exit).

Note: The parentheses have to come after the "=." The **Right** function might be used for this operation.

You may want to refer to section 39.2 for more information on the **MsgBox** function and the values it returns.

2. Rewrite the subroutine in exercise 1 so that it deals correctly with the end of the series. One possible treatment is not to ask to repeat the process when the last cell in the series is dealt with.

   Hint: For this subroutine it might be useful to think of the last cell in the series as the cell that fulfills the criterion `Cell.Item(2,1).Formula=""` (see section 39.2).

3. Write a subroutine that multiplies all cells in the current region by 2.

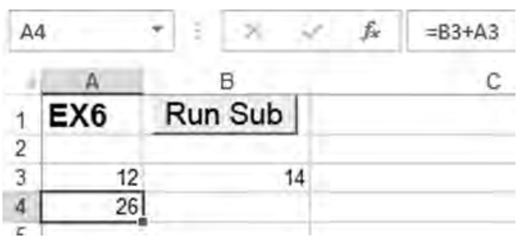
4. Rewrite the subroutine in exercise 3 so that its action is dependent on the cell's contents.

   - If the cell contents is a formula, it will be replaced by the same formula multiplied by 2.
   - If the cell contents is a number, it will be replaced by a number equal to the old number multiplied by 2.
   - On all other cells in the current region, nothing will be done.

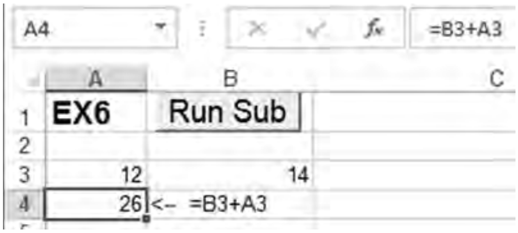
   Note: To make life easier, you may assume, for the purposes of this exercise, that a formula is anything beginning with "=" and a number is anything beginning with the characters "0" to "9."

5. Rewrite the subroutine in exercise 4 so that it uses another method (the correct one) to detect the existence of a formula in a cell. Look at the different properties of the **Range** object in the Help file.

6. The annotations (using **Getformula**) for worksheet formulas in this book were done with a subroutine. For example, running the subroutine on this worksheet



produces the following:



Write a subroutine to perform the annotation. If the cell immediately to the right of the active cell is not empty, the subroutine should overwrite it with **Getformula** only after receiving confirmation from the user.

7. The **Selection** object represents the current selection in the worksheet. **Selection** is usually, and for our purposes always, a **Range** object. Rewrite the subroutine in exercise 6 so that it works on a selected range.

   Note the following:

   If the selected range is a single cell, activate the subroutine in exercise 6.

   If the selected range is a column, activate a subroutine repeatedly for all cells in the column.

   If the selected range is more than one column, the subroutine should abort with an appropriate message.

8. Array functions are functions that return more than one value. For example, the **Transpose** worksheet function returns its argument turned by 90 degrees, as the following worksheet demonstrates:

	A	B	C	D	E	F
1	**TRANSPOSE IN ACTION**					
2	1	2	3	4	1	<-- {=TRANSPOSE(A2:D2)}
3					2	<-- {=TRANSPOSE(A2:D2)}
4					3	<-- {=TRANSPOSE(A2:D2)}
5					4	<-- {=TRANSPOSE(A2:D2)}

The curly brackets were not typed in but were added by Excel to indicate an array formula. The following subroutine created the preceding worksheet:

```
Sub TransposeMe()
      Range("E3:E6").FormulaArray = "=Transpose(A3:D3)"
End Sub
```

The next subroutine is a more complicated version that could deal with any size or place in the row range:

```
Sub TransposeMeToo()
    C = Selection.Columns.Count
    R = Selection.Rows.Count
    If C = 1 Then   'Its a Column
      MsgBox "I don't do Columns"
    ElseIf R = 1 Then 'Its a Row
      Selection.Cells(1, C + 1).Range("A1:A" & C). _
      FormulaArray = "=Transpose(" _
            & Selection.AddressLocal(False, False) & ")"
    Else 'What is it?
        MsgBox "What is it?"
    End If
End Sub
```

Rewrite **TransposeMeToo** so it could deal with column ranges as well as row ranges.

9. Rewrite **TransposeMeToo** of exercise 8 so it could deal with **all** ranges.

Selected References

Note: This bibliography is not intended to be extensive. We give the references mentioned in *Financial Modeling* and occasionally add books and articles that may help the reader expand his/her horizons to more advanced topics. Sometimes the same reference appears several times in different sections. Sometimes we give the references for a number of chapters together, and at other times we reference single chapters.

Chapters 1–6: Corporate Finance and Valuation

Benninga, S., and O. Sarig. 1997. *Corporate Finance: A Valuation Approach*. McGraw-Hill.

Benninga, S., and O. Sarig. 2003. Risk, Returns and Values in the Presence of Differential Taxation. *Journal of Banking and Finance*.

Brealey, R. A., S. C. Myers, and F. Allen. 2013. *Principles of Corporate Finance*, 11th ed. McGraw-Hill.

DeAngelo, H., L. DeAngelo, and D. J. Skinner. 2008. Corporate Payout Policy. *Foundations and Trends in Finance*.

Dittmar, A. K., and R. F. Dittmar. 2004. Stock Repurchase Waves: An Explanation of the Trends in Aggregate Corporate Payout Policy. Working paper, University of Michigan.

Fruhan, W. E. 1979. *Financial Strategy: Studies in the Creation, Transfer, and Destruction of Shareholder Value.* Irwin.

Gordon, M. J. 1959. Dividends, Earnings and Stock Prices. *Review of Economics and Statistics*.

McKinsey & Company, T. Koller, M. Goedhart, and D. Wessels. 2010. *Valuation: Measuring and Managing the Value of Companies,* 5th ed. Wiley.

Ross, S. A., R. W. Westerfield, and J. Jaffe. 2010. *Corporate Finance*, 9th ed. McGraw-Hill.

Chapter 7: Leasing

Abdel-Khalik, A. R. 1981. *Economic Effects on Lessees of FASB Statement No. 13, Accounting for Leases*. Financial Accounting Standards Board.

Copeland, T. E., and J. F. Weston. 1982. A Note on the Evaluation of Cancelable Operating Leases. *Financial Management*.

Financial Accounting Standards Board. 1976. *Statement No. 13: Accounting for Leases*. Stamford, Conn.

Franks, J. R., and S. D. Hodges. 1978. Valuation of Financial Lease Contracts: A Note. *Journal of Finance*.

Levy, H., and M. Sarnat. 1979. On Leasing, Borrowing and Financial Risk. *Financial Management*.

Lewellen, W. G., M. S. Long, and J. J. McConnell. 1979. Asset Leasing in Competitive Capital Markets. *Journal of Finance*.

McConnell, J. J., and J. S. Schallheim. 1983. Valuation of Asset Leasing Contracts. *Journal of Financial Economics*.

Myers, S. C., D. A. Dill, and A. J. Bautista. 1976. Valuation of Financial Lease Contracts. *Journal of Finance*.

Ofer, A. R. 1976. The Evaluation of the Lease versus Purchase Alternative. *Financial Management*.

Schallheim, J. S. 1994. *Lease or Buy? Principles for Sound Decision Making*. Harvard Business School.

Chapters 8–12: Portfolio Calculations

Bengtsson, C., and J. Holst. 2002. On Portfolio Selection: Improved Covariance Matrix Estimation for Swedish Asset Returns. Working paper, Lund University.

Black, F. 1972. Capital Market Equilibrium with Restricted Borrowing. *Journal of Business*.

Bodie, Z., A. Kane, and A. J. Marcus. 2011. *Investments*, 9th ed. McGraw-Hill/Irwin.

Burton, G. M. 2005. Reflections on the Efficient Market Hypothesis: 30 Years Later. *Financial Review*.

Chan, L. K. C., J. Karceski, and J. Lakonishok. 1999. On Portfolio Estimation: Forecasting Covariances and Choosing the Risk Model. *Review of Financial Studies*.

D'Avolio, G. M. 2003. The Market for Borrowing Stock. *Journal of Financial Economics*.

Disatnik, D., and S. Benninga. 2007. Shrinking the Covariance Matrix—Simpler Is Better. *Journal of Portfolio Management*.

Elton, E. J., and M. J. Gruber. 1973. Estimating the Dependence Structure of Share Prices. *Journal of Finance*.

Elton, E. J., M. J. Gruber, S. J. Brown, and W. N. Goetzmann. 2009. *Modern Portfolio Theory and Investment Analysis*, 8th ed. Wiley.

Elton, E. J., M. J. Gruber, and T. Ulrich. 1978. Are Betas Best? *Journal of Finance*.

Fama, E., and K. French. 1997. Industry Costs of Equity. *Journal of Financial Economics*.

Frost, P., and Savarino, J. 1986. An Empirical Bayes Approach to Efficient Portfolio Selection. *Journal of Financial and Quantitative Analysis*.

Green, R. C. 1986. Positively Weighted Portfolios on the Minimum-Variance Frontier. *Journal of Finance*.

Green, R. C., and B. Hollifield. 1992. When Will Mean-Variance Efficient Portfolios Be Well Diversified? *Journal of Finance*.

Jensen, M. C. (Ed.). 1972. *Studies in the Theory of Capital Markets*. Praeger.

Ledoit, O., and M. Wolf. 2003. Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection. *Journal of Empirical Finance*.

Ledoit, O., and M. Wolf. 2004. Honey, I Shrunk the Sample Covariance Matrix. *Journal of Portfolio Management*.

Lintner, J. 1965. The Valuation of Risky Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budget. *Review of Economics and Statistics*.

Markowitz, H. 1952. Portfolio Selection. *Journal of Finance*.

Merton, R. C. 1973. An Analytic Derivation of the Efficient Portfolio Frontier. *Journal of Financial and Quantitative Analysis*.

Mossin, J. 1966. Equilibrium in a Capital Market. *Econometrica*.

Nielsen, L. R. 1987. Positively Weighted Frontier Portfolios: A Note. *Journal of Finance*.

Roll, R. 1977. A Critique of the Asset Pricing Theory's Tests, Part I: On Past and Potential Testability of the Theory. *Journal of Financial Economics*.

Roll, R. 1978. Ambiguity When Performance Is Measured by the Securities Market Line. *Journal of Finance*.

Sharpe, W. F. 1963. A Simplified Model for Portfolio Analysis. *Management Science*.

Sharpe, W. F. 1964. Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *Journal of Finance*.

Surowiecki, J. 2003. Get Shorty. *The New Yorker*. http://newyorker.com/talk/content/?031201ta_talk_surowiecki.

Chapter 13: Black-Litterman

Best, M. J., and R. R. Grauer. 1985. Capital Asset Pricing Compatible with Observed Market Value Weights. *Journal of Finance*.

Best, M. J., and R. R. Grauer. 1991. On the Sensitivity of Mean-Variance-Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results. *Review of Financial Studies*.

Bevan, A., and K. Winkelmann. 1998. Using the Black-Litterman Global Asset Allocation Model: Three Years of Practical Experience. Goldman-Sachs.

Black, F., and R. Litterman. 1991. Global Asset Allocation with Equities, Bonds, and Currencies. Goldman-Sachs. (See the disk that accompanies this book.)

Chopra, V. K., and W. Ziemba. 1993. The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice. *Journal of Portfolio Management*.

DeMiguel, V., L. Garlappi, and R. Uppal. 2007. Optimal versus Naive Diversification: How Inefficient Is the 1/N Portfolio Strategy? *Review of Financial Studies*.

Goldman-Sachs. 1999. The Intuition Behind Black-Litterman Model Portfolios. Author.

Jagannathan, R., and T. Ma. 2002. Risk Reduction in Large Portfolios: A Role for Portfolio Weight Constraints. *Journal of Finance*.

Kandel, S., and R. F. Stambaugh. 1995. Portfolio Inefficiency and the Cross-Section of Expected Returns. *Journal of Finance*.

Litterman, R. 2003. *Modern Investment Management: An Equilibrium Approach*. Wiley.

Schwartz, T. 2000. How to Beat the S&P 500 with Portfolio Optimization. Mimeo.

Sharpe, W. F. 1974. Imputing Expected Portfolio Returns from Portfolio Composition. *Journal of Financial and Quantitative Analysis*.

Chapter 14: Event Studies

Ball, C., and W. N. Torous. 1988, Investigating Security Price Performance in the Presence of Event Date Uncertainty. *Journal of Financial Economics*.

Binder, J. J. 1985, On the Use of the Multivariate Regression Model in Event Studies. *Journal of Accounting Research*.

Boehmer, E., J. Musumeci, and A. B. Poulsen. 1991. Event-Study Methodology Under Conditions of Event-Induced Variance. *Journal of Financial Economics*.

Brown, S., and J. B. Warner. 1985. Using Daily Stock Returns: The Case of Event Studies. *Journal of Financial Economics*.

Campbell, J. Y., A. Lo, and C. McKinley. 1996. *The Econometrics of Financial Markets*. Princeton University Press.

Fama, E. 1976, *Foundations of Finance*. Chapters 3 and 4. Basic Books.

Fama, E., L. Fisher, M. Jensen, and R. Roll. 1969. The Adjustment of Stock Prices to New Information. *International Economic Review*.

MacKinlay, C. 1997. Event Studies in Economics and Finance. *Journal of Economic Literature*.

Salinger, M. 1992. Value Event Studies. *Review of Economics and Statistics*.

Thompson, R. 1995. Empirical Methods of Event Studies in Corporate Finance. Chapter 29. R. Jarrow et al. (Eds.), *Handbooks in OR & MS* (Vol. 9). Elsevier Science B.V.

Chapters 15–19: Options

Bhaghat, S., J. Brickley, and U. Loewenstein. 1987. The Pricing Effects of Interfirm Cash Tender Offers. *Journal of Finance*.

Billingsley, P. 1968. *Convergence of Probability Measures*. Wiley.

Black, F., and M. Scholes. 1973. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*.

Brennan, M. J., and E. S. Schwartz. 1976. The Pricing of Equity-Linked Life Insurance Policies with an Asset Value Guarantee. *Journal of Financial Economics*.

Brennan, M. J., and R. Solanki. 1981. Optimal Portfolio Insurance. *Journal of Financial and Quantitative Analysis*.

Copeland, T. E., and J. F. Weston Shastri. 2003. *Financial Theory and Corporate Policy*. Addison-Wesley.

Cox, J., and S. A. Ross. 1976. The Valuation of Options for Alternative Stochastic Processes. *Journal of Financial Economics*.

Cox, J., and M. Rubinstein. 1985. *Options Markets*. Prentice-Hall.

Cox, J., S. A. Ross, and M. Rubinstein. 1979. Option Pricing: A Simplified Approach. *Journal of Financial Economics*.

Cvitanić, J., Z. Wiener, and F. Zapatero. 2006. Analytic Pricing of Employee Stock Options. *Review of Financial Studies*.

Gatto, M. A., R. Geske, R. Litzenberger, and H. Sosin. 1980. Mutual Fund Insurance. *Journal of Financial Economics*.

Haug, E. G. 2006. *The Complete Guide to Option Pricing Formulas*, 2nd ed. McGraw-Hill.

Hull, J. 2000. *Options Futures, and Other Derivative Securities*, 8th ed. Prentice Hall.

Hull, J., and A. White. 2004. How to Value Employee Stock Options. *Financial Analysts Journal*.

Jacobs, B. 1983. The Portfolio Insurance Puzzle. *Pensions and Investment Age*.

Jacques, W. E. 1987. Portfolio Insurance or Job Insurance? *Financial Analysts Journal*.

Jarrow, R. A., and A. Rudd. 1983. *Option Pricing*. Irwin.

Knuth, D. E. 1981. *The Art of Computer Programming. Vol. 2: Seminumerical Algorithms*. Addison-Wesley.

Leland, H. E. 1980. Who Should Buy Portfolio Insurance? *Journal of Finance*.

Leland, H. E. 1985. Option Pricing and Replication with Transaction Costs. *Journal of Finance*.

Merton, R. C. 1973. Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science*.

Merton, R. C. 1976. Option Pricing When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*.

Omberg, E. 2012. A Note on the Convergence of Binomial-Pricing and Compound-Option Models. *Journal of Finance*.

Pozen, R. C. 1978. When to Purchase a Protective Put. *Financial Analysts Journal*.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 2007. *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge University Press.

Rubinstein, M. 1985. Alternative Paths to Portfolio Insurance. *Financial Analysts Journal*.

Rubinstein, M., and H. E. Leland. 1981. Replicating Options with Positions in Stock and Cash. *Financial Analysts Journal*.

Chapter 19: Real Options

Amram, M., and N. Kulatilaka. 1998. *Real Options: Managing Strategic Investment in an Uncertain World*. Harvard Business School Press.

Benninga, S., and E. Tolkowsky. 2004. Real Options—An Introduction and an Application to R&D Valuation. *Engineering Economist*.

Dixit, A. K., and R. S. Pindyck. 1995. The Options Approach to Capital Investment. *Harvard Business Review*.

Luehrman, T. A. 1998. Investment Opportunities as Real Options: Getting Started on the Numbers. *Harvard Business Review*.

Trigeorgis, L. 1993. Real Options and Interactions with Financial Flexibility. *Financial Management*.

Trigeorgis, L. 1996. *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. MIT Press.

Chapters 20–21: Duration and Immunization

Altman, E. 1989. Measuring Corporate Bond Mortality and Performance. *Journal of Finance*.

Altman, E., and V. M. Kishore. 1996. Almost Everything You Wanted to Know About Recoveries on Defaulted Bonds. *Financial Analysts Journal*.

Babcock, G. 1985. Duration as a Weighted Average of Two Factors. *Financial Analysts Journal*.

Bierwag, G. O. 1977. Immunization, Duration, and the Term Structure of Interest Rates. *Journal of Financial and Quantitative Analysis*.

Bierwag, G. O. 1978. Measures of Duration. *Economic Inquiry*.

Bierwag, G. O., G. G. Kaufman, and A. Toevs. 1983a. Duration: Its Development and Use in Bond Portfolio Management. *Financial Analysts Journal*.

Bierwag, G. O., G. G. Kaufman, and A. Toevs. 1983b. *Innovations in Bond Portfolio Management: Duration Analysis and Immunization*. JAI Press.

Bierwag, G. O., G. G. Kaufman, R. Schweitzer, and A. Toevs. 1981. The Art of Risk Management in Bond Portfolios. *Journal of Portfolio Management*.

Billingham, C. J. 1983. Strategies for Enhancing Bond Portfolio Returns. *Financial Analysts Journal*.

Chance, D. M. 1983. Floating Rate Notes and Immunization. *Journal of Financial and Quantitative Analysis*.

Chance, D. M. 1996. Duration Convexity, and Time as Components of Bond Returns. *Journal of Fixed Income*.

Chua, J. H. 1984. A Closed-Form Formula for Calculating Bond Duration. *Financial Analysts Journal*.

Cooper, I. A. 1977. Asset Values, Interest Rate Changes, and Duration. *Journal of Financial and Quantitative Analysis*.

Cox, J. C., J. E. Ingersoll, and S. A. Ross. 1979. Duration and Measurement of Basis Risk. *Journal of Business*.

Fisher, L., and R. L. Weil. 1971. Coping with the Risk of Market-Rate Fluctuations: Returns to Bondholders from Naive and Optimal Strategies. *Journal of Business*.

Gultekin, B., and R. J. Rogalski. 1984. Alternative Duration Specifications and the Measurement of Basis Risk: Empirical Tests. *Journal of Business*.

Hicks, J. 1939. *Value and Capital*. Clarendon Press.

Ingersoll, J. E., Jr., J. Skelton, and R. L. Weil. 1978. Duration Forty Years Later. *Journal of Financial and Quantitative Analysis*.

Leibowitz, M. L., and A. Weinberger. 1981. The Uses of Contingent Immunization. *Journal of Portfolio Management*.

Macaulay, F. R. 1938. *Some Theoretical Problems Suggested by Movements of Interest Rates, Bond Yields and Stock Prices in the United States Since 1856*. National Bureau of Economic Research.

McCullogh, J. H. 1990. U.S. Term Structure Data, 1946–1987. *Handbook of Monetary Economics* (Vol. 1). North-Holland.

Ott, R. A., Jr. 1986. The Duration of an Adjustable-Rate Mortgage and the Impact of the Index. *Journal of Finance*.

Redington, F. M. 1952. Review of the Principle of Life-Office Valuations. *Journal of the Institute of Actuaries*. Reprinted in G. A. Hawawini (Ed.), *Bond Duration and Immunization: Early Developments and Recent Contributions*. Garland, 1972.

Samuelson, P. A. 1945. The Effects of Interest Rate Increases on the Banking System. *American Economic Review*.

Smith, D. J. 1998. A Note on the Derivation of Closed-Form Formulas for Duration and Convexity Statistics on and between Coupon Dates. *Journal of Financial Engineering*.

Weil, R. L. 1973. Macaulay's Duration: An Appreciation. *Journal of Business*.

Chapter 22: Term Structure Modeling

Adams, K. J., and D. R. Van Deventer. 1994. Fitting Yield Curves and Forward Rate Curves with Maximum Smoothness. *Journal of Fixed Income*.

Annaert, J., G. P. Claes, M. J. K. Ceuster, and H. Zhang. 2013. Estimating the Spot Rate Curve Using the Nelson-Siegel Model: A Ridge Regression Approach. *International Review of Economics and Finance*.

Chambers, D., W Carleton, and D. Waldman. 1984. A New Approach to Estimation of Term Structure of Interest Rates. *Journal of Financial and Quantitative Studies*.

Coleman, T. S., L. Fisher, and R. Ibbotson. 1992. Estimating the Term Structure of Interest from Data That Include the Prices of Coupon Bonds. *Journal of Fixed Income*.

Coleman, T., L. Fisher, and R. Ibbotson. n.d. *U.S. Treasury Yield Curves 1926–1994*. Moody's Investor Service.

Diament, P. 1993. Semi-Empirical Smooth Fit to the Treasury Yield Curve. *Journal of Fixed Income*.

Fabozzi, F. J. 1996. *Bond Markets, Analysis and Strategies.* Prentice Hall.

Fisher, M., and D. Zervos. 1996. Yield Curve. In H. R. Varian (Ed.), *Computational Economics and Finance: Modeling and Analysis with Mathematica*. Springer.

Ho, T. S. Y. 1992. Key Rate Durations: Measures of Interest Rate Risk. *Journal of Fixed Income*.

Jarrow, R. A.. 1996. *Modeling Fixed Income Securities and Interest Rate Options*. McGraw-Hill.

Litterman, R., and J. Scheinkman. 1991. Common Factors Affecting Bond Returns. *Journal of Fixed Income*.

Mann, S. V., and P. Ramanlal. 1997. Relative Performance of Yield Curve Strategies. *Journal of Portfolio Management*.

McCulloch, J. F. 1971. Measuring the Term Structure of Interest Rates. *Journal of Business*.

McCulloch, J. F. 1975. The Tax-Adjusted Yield Curve. *Journal of Finance*.

Nelson, C., and A. Siegel. 1987. Parsimonious Modeling of Yield Curves. *Journal of Business*.

Shea, G. 1984. Pitfalls in Smoothing Interest Rate Term Structure Data: Equilibrium Models and Spline Approximations. *Journal of Financial and Quantitative Analysis*.

Shea, G. 1985. Interest Rate Term Structure Estimation with Exponential Splines: A Note. *Journal of Finance*.

Stigum, M., and F. L. Robinson. 1996. *Money Market & Bond Calculations*. Irwin.

Suits, D. B., A. Mason, and L. Chan. 1978. Spline Functions Fitted by Standard Regression Methods. *Review of Economics and Statistics*.

Sundaresan, S. 1997. *Fixed Income Markets and Their Derivatives*. South-Western College.

Taggart, R. A., Jr. 1996 *Quantitative Analysis for Investment Management*. Prentice-Hall.

Tuckman, B. 1996. *Fixed-Income Securities.* Wiley.

Vasicek, O., and G. Fong. 1982. Term Structure Estimation Using Exponential Splines. *Journal of Finance*.

Chapter 22: Term Structure

Diebold, F. X., and C. Li. 2006. Forecasting the Term Structure of Government Bond Yields. *Journal of Econometrics*.

Gürkaynak, R. S., B. Sack, and J. H. Wright. 2007. The U.S. Treasury Yield Curve: 1961 to the Present. *Journal of Monetary Economics*.

Nelson, C., and A. Siegel. 1987. Parsimonious Modeling of Yield Curves. *Journal of Business*.

Svensson, L. E. O. 1995. Estimating Forward Interest Rates with the Extended Nelson & Siegel Method. *Sveriges Riksbank Quarterly Review.*

Chapter 23: Expected Bond Returns

Altman, E. I., and A. C. Eberhart. 1994. Do Seniority Provisions Protect Bondholders' Investments? *Journal of Portfolio Management*.

Altman, E. I., and V. M. Kishore. 1996. Almost Everything You Wanted to Know About Recoveries on Defaulted Bonds. *Financial Analysts Journal*.

Altman, E. I., and S. Nammacher. 1984. The Default Rate Experience of High Yield Corporate Debt. *Financial Analysts Journal*.

Amihud, Y., and H. Mendelson. 1991. Liquidity, Maturity, and the Yields on U.S. Treasury Securities. *Journal of Finance*.

Antonov, A., and Y. Yanakieva. 2004. Transition Matrix Generation. International Conference on Computer Systems and Technologies.

Bakshi G., D. B. Madan, and F. X. Zhang. 2004. Understanding the Role of Recovery in Default Risk Models: Empirical Comparisons and Implied Recovery Rates. Unpublished working paper.

Cantor, R. 2001. Moody's Investors Service Response to the Consultative Paper Issued by The Basel Committee on Banking Supervision and Its Implications for the Rating Agency Industry. *Journal of Banking and Finance*.

Carty, L. V., and J. S. Fons. 2003. Measuring Changes in Corporate Credit Quality. Moody's Special Report, November.

Chacko, G. 2005. Liquidity Risk in the Corporate Bond Markets. Working paper, Harvard Business School.

Cochrane, J. H. 2001. *Asset Pricing*. Princeton University Press.

Crosbie P. J., and J. R. Bohn. 2002. *Modeling Default Risk*. KMV.

Driessen, J., and F. de Jong. 2005. Liquidity Risk Premia in Corporate Bond Markets. Discussion paper, University of Amsterdam.

Duffie, D. 1998. Defaultable Term Structure Models with Fractional Recovery of Par. Working paper, Stanford University.

Duffie, D., and K. J. Singleton. 1997. An Econometric Model of the Term Structure of Interest-Rate Swap Yields. *Journal of Finance*.

Duffie, D., and K. J. Singleton. 1999. Modeling Term Structures of Defaultable Bonds. *Review of Financial Studies*.

Dynkin L., A. Gould, J. Hyman, V. Konstantinovsky, and B. Phelps. 2006. *Quantitative Management of Bond Portfolios.* Princeton University Press.

Edwards, A. K., L. E. Harris, and M. S. Piwowar. 2004. *Corporate Bond Market Transparency and Transaction Costs*. UWFC.

Fama, E., and K. R. French. 1993. Common Risk Factors in the Returns of Stocks and Bonds. *Journal of Financial Economics*.

Finger, C. C., V. Finkelstein, G. Pan, J. P. Lardy, T. Ta, and J. Tierney. 2002. CreditGrades Technical Document, RiskMetrics Group, Inc.

Gebhardt W. R., S. Hvidkjaer, and B. Swaminathan. 2005. The Cross-Section of Expected Corporate Bond Returns: Betas or Characteristics? *Journal of Financial Economics*.

Hagan, P., and G. West. 2005. Methods for Constructing a Yield Curve. *Wilmott Magazine*.

Hickman, W. B. 1958. *Corporate Bond Quality and Investor Experience*. Princeton University Press.

Jafry, Y., and T. Schuermann. 2004. Metrics for Comparing Credit Migration Matrices. *Journal of Banking and Finance*.

Jarrow, R. A., D. Lando, and S. M. Turnbull. 1997. A Markov Model for the Term Structure of Credit Spreads. *Review of Financial Studies*.

Jarrow, R. A., and S. M. Turnbull. 1995, Pricing Derivatives on Financial Securities Subject to Credit Risk. *Journal of Finance*.

Lando, D. 1998. On Cox Processes and Credit Risky Securities. *Review of Derivatives Research*.

Lando, D. 2004. *Credit Risk Modeling Theory and Applications*. Princeton University Press.

Löffler, G. 2004. An Anatomy of Rating Through the Cycle. *Journal of Banking and Finance*.

Löffler, G. 2005. Avoiding the Rating Bounce: Why Rating Agencies Are Slow to React to New Information. *Journal of Economic Behavior and Organization*.

Longstaff, F. A. 2001. The Flight-to-Liquidity Premium in U.S. Treasury Bond Prices. *Journal of Business*.

Longstaff, F. A., S. Mithal, and E. Neis. 2004. Corporate Yield Spreads: Default Risk or Liquidity? New Evidence from the Credit-Default Swap Market. *Journal of Finance*.

Merton, R. C. 1974. On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *Journal of Finance*.

Moody's Special Report. 1992. *Corporate Bond Defaults and Default Rates*. Moody's Investors Service.

Nelson, C., and A. Siegel. 1987. Parsimonious Modeling of Yield Curves, *Journal of Business*.

Parnes, D. 2005. Homogeneous Markov Chain, Stochastic Economics, and Non-Homogeneous Models for Measuring Corporate Credit Risk. Working paper, University of South Florida.

Subramanian, K. V. 2001. Term Structure Estimation in Illiquid Markets. *Journal of Fixed Income*.

Yu, F. 2002. Modeling Expected Return on Defaultable Bonds. *Journal of Fixed Income*.

Chapters 24–30: Monte Carlo

Abramowitz, M., and I. A. Stegun. 1965. *Handbook of Mathematical Functions*. Dover.

Acworth, P., M. Broadie, and P. Glasserman. 1996. A Comparison of Some Monte Carlo and Quasi-Monte Carlo Techniques for Option Pricing. In Niederreiter et al. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods*. Springer.

Bailey, D. H., J. M. Borwein, and P. B. Borwein. 1987. Ramanujan, Modular Equations, and Approximations to π; or, How to Compute One Billion Digits of Pi. *American Mathematical Monthly*.

Baker, N. L., and R. A. Haugen. 2012. Low Risk Stocks Outperform Within All Observable Markets of the World. http://ssrn.com/abstract=2055431.

Barone-Adesi, G., and R. E. Whaley. 1987. Efficient Analytic Approximation of American Option Values. *Journal of Finance*.

Benninga, S., and M. Blume. 1985. On the Optimality of Portfolio Insurance. *Journal of Finance*.

Box, G. E. P., M. E. Muller. 1958. A Note on the Generation of Random Normal Deviates. *Annals of Mathematical Statistics*.

Boyle, P. P. 1977. Options: A Monte Carlo Approach. *Journal of Financial Economics*.

Boyle, P. P., M. Broadie, M. Glasserman. 1997. Monte Carlo Methods for Security Pricing. *Journal of Economic Dynamics and Control*.

Boyle, P. P., and S. H. Lau. 1994. Bumping Up Against the Barrier with the Binomial Method. *Journal of Derivatives*.

Boyle, P. P., and Y. K. Tse. 1990. An Algorithm for Computing Values of Options on the Maximum or Minimum of Several Assets. *Journal of Financial and Quantitative Analysis*.

Boyle, P. P., and T. Vorst. 1992. Option Replication in Discrete Time with Transaction Costs. *Journal of Finance*.

Broadie, M., and P. Glasserman. 1997. Pricing American-Style Securities Using Simulation. *Journal of Economic Dynamics and Control*.

Broadie, M., P. Glasserman, and G. Jain. 1997. Enhanced Monte Carlo Estimates for American Option Prices. *Journal of Derivatives*.

Cox, J., S. A. Ross, and M. Rubinstein. 1979. Option Pricing: A Simplified Approach. *Journal of Financial Economics*.

Cremers, M., A. Petajisto, and E. Zitzewitz. 2010. Should Benchmark Indices Have Alpha? Revisiting Performance Evaluation. http://ssrn.com/abstract=1108856.

Dupire, B. 1998. *Monte Carlo Methodologies and Applications for Pricing and Risk Management*. Risk Books.

Frazzini, A., and L. H. Pedersen. 2011. Betting Against Beta. http://ssrn.com/abstract=2049939.

Fu, M., S. B. Laprise, D. B. Madan, Y. Su, and R. Wu. 2001. Pricing American Options: A Comparison of Monte Carlo Simulation Approaches. *Journal of Computational Finance*.

Galanti, S., and A. Jung. 1997. Low-Discrepancy Sequences: Monte Carlo Simulation of Option Prices. *Journal of Derivatives*.

Gamba, A. 2002. Real Options Valuation: A Monte Carlo Approach. http://ssrn.com/abstract=302613.

Gentle, J. E. 1998. *Random Number Generation and Monte Carlo Methods*. Springer.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. Springer.

Grant, D., G. Vora, and D. E. Weeks. 1996a. Path-Dependent Options: Extending the Monte Carlo Simulation Approach. *Management Science*.

Grant, D., G. Vora, and D. E. Weeks. 1996b. Simulation and Early-Exercise of Option Problem, *Journal of Financial Engineering*.

Haugh, M. 2004. The Monte Carlo Framework, Examples from Finance and Generating Correlated Random Variables. www.columbia.edu/~mh2078/MCS04/MCS_framework_FEegs.pdf.

Haugh, M. B., and L. Kogan. 2008. Duality Theory and Approximate Dynamic Programming for Pricing American Options and Portfolio Optimization. In J. R. Birge and V. Linetsky (Eds.), *Handbooks in OR & MS* (Vol. 15). Elsevier.

Heynen, R. C., and H. M. Kat. 1994a. Crossing Barriers. *Risk*.

Heynen, R. C., and H. M. Kat. 1994b. Partial Barrier Options. *Journal of Financial Engineering*.

Hong, H G., and D. A. Sraer. 2012. Speculative Betas. Available at SSRN. http://ssrn.com/abstract=1967462 or http://dx.doi.org/10.2139/ssrn.1967462.

Kanigel, R. 1992. *The Man Who Knew Infinity: A Life of the Genius Ramanujan*. Scribner.

Knuth, D. E. 1981. *The Art of Computer Programming. Volume 2: Semi-Numerical Algorithms*. Addison-Wesley.

Lehmer, D. H. 1951. Mathematical Methods in Large-Scale Computing Units, Proceedings of the 2nd Symposium on Large-Scale Digital Calculating Machinery. *Annals of the Computational Laboratory of Harvard University*. Harvard University Press.

Longstaff, F. A., and E. S. Schwartz. 2001. Valuing American Options by Simulation: A Simple Least-Square Approach. *Review of Financial Studies*.

Raymar, S. B., and M. J. Zwecher. 1997. Monte Carlo Estimation of American Call Option on the Maximum of Several Stocks. *Journal of Derivatives*.

Rogers, L. C. G. 2002. *Monte Carlo Valuation of American Options*. Mathematical Finance.

Tavella, D. 2002. *Quantitative Methods in Derivatives Pricing—An Introduction to Computational Finance*. Wiley.

Tezuka, S. 1998. Financial Applications of Monte Carlo and Quasi-Monte Carlo Methods. In P. Hellekalek and G. Larcher (Eds.), *Random and Quasi-Random Point Sets*. Springer.

Tilley, J. A. 1998. Valuing American Options in a Path Simulation Model. In B. Dupire (Ed.), *Monte Carlo*. Risk Books. Originally published in 1993.

Tsitsiklis, J. N., and B. Van Roy. 2001. Regression Methods for Pricing Complex American-Style Options. *IEEE Transactions on Neural Networks*.

Willard, G. A. 1997. Calculating Prices and Sensitivities for Path-Independent Derivative Securities in Multifactor Models. *Journal of Derivatives*.

Chapter 28: Value at Risk

Beder T. 1996. VAR: Seductive But Dangerous. *Financial Analysts Journal*.

Jorion, P. 1997. *Value at Risk: the New Benchmark for Controlling Market Risk*. McGraw-Hill.

Linsmeier, T. J., and N. D. Pearson. 2000. Risk Measurement: An Introduction to Value at Risk. *Financial Analysts Journal*.

RiskMetrics. 1995. Introduction to RiskMetrics. This and other documents can be found at www.riskmetrics.com/techdoc.html.

Index