Advanced Modelling in Finance
using Excel and VBA

Mary Jackson
and
Mike Staunton

JOHN WILEY & SONS, LTD
Chichester • New York • Weinheim • Brisbane • Singapore • Toronto
Advanced Modelling in Finance using Excel and VBA
Wiley Finance Series

Operational Risk: Measurement and Modelling
Jack King

Advance Credit Risk Analysis: Financial Approaches and Mathematical Models to Assess, Price and Manage Credit Risk
Didier Cossin and Hugues Pirotte

Dictionary of Financial Engineering
John F. Marshall

Pricing Financial Derivatives: The Finite Difference Method
Domingo A Tavella and Curt Randall

Interest Rate Modelling
Jessica James and Nick Webber

Handbook of Hybrid Instruments: Convertible Bonds, Preferred Shares, Lyons, ELKS, DECS and Other Mandatory Convertible Notes
Izzy Nelken (ed)

Options on Foreign Exchange, Revised Edition
David F DeRosa

The Handbook of Equity Derivatives, Revised Edition
Jack Francis, William Toy and J Gregg Whittaker

Volatility and Correlation in the Pricing of Equity, FX and Interest-Rate Options
Riccardo Rebonato

Risk Management and Analysis vol. 1: Measuring and Modelling Financial Risk
Carol Alexander (ed)

Risk Management and Analysis vol. 2: New Markets and Products
Carol Alexander (ed)

Implementing Value at Risk
Philip Best

Credit Derivatives: A Guide to Instruments and Applications
Janet Tavakoli

Implementing Derivatives Models
Les Clewlow and Chris Strickland

Interest-Rate Option Models: Understanding, Analysing and Using Models for Exotic Interest-Rate Options (second edition)
Riccardo Rebonato
Advanced Modelling in Finance
using Excel and VBA

Mary Jackson
and
Mike Staunton
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.13</td>
<td>Matrix algebra and related functions</td>
<td>33</td>
</tr>
<tr>
<td>2.13.1</td>
<td>Introduction to matrices</td>
<td>33</td>
</tr>
<tr>
<td>2.13.2</td>
<td>Transposing a matrix</td>
<td>33</td>
</tr>
<tr>
<td>2.13.3</td>
<td>Adding matrices</td>
<td>34</td>
</tr>
<tr>
<td>2.13.4</td>
<td>Multiplying matrices</td>
<td>34</td>
</tr>
<tr>
<td>2.13.5</td>
<td>Matrix inversion</td>
<td>35</td>
</tr>
<tr>
<td>2.13.6</td>
<td>Solving systems of simultaneous linear equations</td>
<td>36</td>
</tr>
<tr>
<td>2.13.7</td>
<td>Summary of Excel’s matrix functions</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>37</td>
</tr>
</tbody>
</table>

### 3 Introduction to VBA

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Advantages of mastering VBA</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Object-oriented aspects of VBA</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Starting to write VBA macros</td>
<td>42</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Some simple examples of VBA subroutines</td>
<td>42</td>
</tr>
<tr>
<td>3.3.2</td>
<td>MsgBox for interaction</td>
<td>43</td>
</tr>
<tr>
<td>3.3.3</td>
<td>The writing environment</td>
<td>44</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Entering code and executing macros</td>
<td>44</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Recording keystrokes and editing code</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>Elements of programming</td>
<td>47</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Variables and data types</td>
<td>48</td>
</tr>
<tr>
<td>3.4.2</td>
<td>VBA array variables</td>
<td>48</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Control structures</td>
<td>50</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Control of repeating procedures</td>
<td>51</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Using Excel functions and VBA functions in code</td>
<td>52</td>
</tr>
<tr>
<td>3.4.6</td>
<td>General points on programming</td>
<td>53</td>
</tr>
<tr>
<td>3.5</td>
<td>Communicating between macros and the spreadsheet</td>
<td>53</td>
</tr>
<tr>
<td>3.6</td>
<td>Subroutine examples</td>
<td>56</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Charts</td>
<td>56</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Normal probability plot</td>
<td>59</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Generating the efficient frontier with Solver</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>65</td>
</tr>
<tr>
<td>Appendix 3A</td>
<td>The Visual Basic Editor</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Stepping through a macro and using other debug tools</td>
<td>68</td>
</tr>
<tr>
<td>Appendix 3B</td>
<td>Recording keystrokes in ‘relative references’ mode</td>
<td>69</td>
</tr>
</tbody>
</table>

### 4 Writing VBA user-defined functions

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>A simple sales commission function</td>
<td>73</td>
</tr>
<tr>
<td>4.2</td>
<td>Creating Commission(Sales) in the spreadsheet</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>Two functions with multiple inputs for valuing options</td>
<td>75</td>
</tr>
<tr>
<td>4.4</td>
<td>Manipulating arrays in VBA</td>
<td>78</td>
</tr>
<tr>
<td>4.5</td>
<td>Expected value and variance functions with array inputs</td>
<td>79</td>
</tr>
<tr>
<td>4.6</td>
<td>Portfolio variance function with array inputs</td>
<td>81</td>
</tr>
<tr>
<td>4.7</td>
<td>Functions with array output</td>
<td>84</td>
</tr>
<tr>
<td>4.8</td>
<td>Using Excel and VBA functions in user-defined functions</td>
<td>85</td>
</tr>
</tbody>
</table>
## Contents

- **Part One**
  - 4.8.1 Using VBA functions in user-defined functions 85
  - 4.8.2 Add-ins 86
  - 4.9 Pros and cons of developing VBA functions 86
     - Summary 87
     - Appendix 4A Functions illustrating array handling 88
     - Appendix 4B Binomial tree option valuation functions 89
     - Exercises on writing functions 94
     - Solution notes for exercises on functions 95

- **Part Two** Equities 99

  - 5 Introduction to equities 101

  - 6 Portfolio optimisation 103
     - 6.1 Portfolio mean and variance 103
     - 6.2 Risk–return representation of portfolios 105
     - 6.3 Using Solver to find efficient points 106
     - 6.4 Generating the efficient frontier (Huang and Litzenberger’s approach) 109
     - 6.5 Constrained frontier portfolios 111
     - 6.6 Combining risk-free and risky assets 113
     - 6.7 Problem One–combining a risk-free asset with a risky asset 114
     - 6.8 Problem Two–combining two risky assets 115
     - 6.9 Problem Three–combining a risk-free asset with a risky portfolio 117
     - 6.10 User-defined functions in Module1 119
     - 6.11 Functions for the three generic portfolio problems in Module1 120
     - 6.12 Macros in ModuleM 121
        - Summary 123
        - References 123

  - 7 Asset pricing 125
     - 7.1 The single-index model 125
     - 7.2 Estimating beta coefficients 126
     - 7.3 The capital asset pricing model 129
     - 7.4 Variance–covariance matrices 130
     - 7.5 Value-at-Risk 131
     - 7.6 Horizon wealth 134
     - 7.7 Moments of related distributions such as normal and lognormal 136
     - 7.8 User-defined functions in Module1 136
        - Summary 138
        - References 138

  - 8 Performance measurement and attribution 139
     - 8.1 Conventional performance measurement 140
     - 8.2 Active–passive management 141
     - 8.3 Introduction to style analysis 144
### Part Three Options on Equities

#### 9 Introduction to options on equities
- 9.1 The genesis of the Black–Scholes formula
- 9.2 The Black–Scholes formula
- 9.3 Hedge portfolios
- 9.4 Risk-neutral valuation
- 9.5 A simple one-step binomial tree with risk-neutral valuation
- 9.6 Put–call parity
- 9.7 Dividends
- 9.8 American features
- 9.9 Numerical methods
- 9.10 Volatility and non-normal share returns

#### 10 Binomial trees
- 10.1 Introduction to binomial trees
- 10.2 A simplified binomial tree
- 10.3 The Jarrow and Rudd binomial tree
- 10.4 The Cox, Ross and Rubinstein tree
- 10.5 Binomial approximations and Black–Scholes formula
- 10.6 Convergence of CRR binomial trees
- 10.7 The Leisen and Reimer tree
- 10.8 Comparison of CRR and LR trees
- 10.9 American options and the CRR American tree
- 10.10 User-defined functions in Module0 and Module1

#### 11 The Black–Scholes formula
- 11.1 The Black–Scholes formula
- 11.2 Black–Scholes formula in the spreadsheet
- 11.3 Options on currencies and commodities
- 11.4 Calculating the option’s ‘greek’ parameters
- 11.5 Hedge portfolios
- 11.6 Formal derivation of the Black–Scholes formula
11.7 User-defined functions in Module1
  Summary
  References

12 Other numerical methods for European options
12.1 Introduction to Monte Carlo simulation
12.2 Simulation with antithetic variables
12.3 Simulation with quasi-random sampling
12.4 Comparing simulation methods
12.5 Calculating greeks in Monte Carlo simulation
12.6 Numerical integration
12.7 User-defined functions in Module1
  Summary
  References

13 Non-normal distributions and implied volatility
13.1 Black–Scholes using alternative distributional assumptions
13.2 Implied volatility
13.3 Adapting for skewness and kurtosis
13.4 The volatility smile
13.5 User-defined functions in Module1
  Summary
  References

Part Four Options on Bonds

14 Introduction to valuing options on bonds
14.1 The term structure of interest rates
14.2 Cash flows for coupon bonds and yield to maturity
14.3 Binomial trees
14.4 Black’s bond option valuation formula
14.5 Duration and convexity
14.6 Notation
  Summary
  References

15 Interest rate models
15.1 Vasicek’s term structure model
15.2 Valuing European options on zero-coupon bonds, Vasicek’s model
15.3 Valuing European options on coupon bonds, Vasicek’s model
15.4 CIR term structure model
15.5 Valuing European options on zero-coupon bonds, CIR model
15.6 Valuing European options on coupon bonds, CIR model
15.7 User-defined functions in Module1
  Summary
  References
x  Contents

16  Matching the term structure  243
    16.1  Trees with lognormally distributed interest rates  243
    16.2  Trees with normal interest rates  246
    16.3  The Black, Derman and Toy tree  247
    16.4  Valuing bond options using BDT trees  248
    16.5  User-defined functions in Module1  250
          Summary  252
          References  252

Appendix  Other VBA functions  253
          Forecasting  253
          ARIMA modelling  254
          Splines  256
          Eigenvalues and eigenvectors  257
          References  258

Index  259
Preface

When asked why they tackled Mount Everest, climbers typically reply “Because it was there”. Our motivation for writing Advanced Modelling in Finance is for exactly the opposite reason. There were then, and still are now, almost no books that give due prominence to and explanation of the use of VBA functions within Excel. There is an almost similar lack of books that capture the true vibrant spirit of numerical methods in finance.

It is no longer true that spreadsheets such as Excel are inadequate tools in highly technical and numerically demanding areas such as the valuation of financial derivatives. With efficient code and VBA functions, calculations that were once the preserve of dedicated packages and languages can now be done on a modern PC in Excel within seconds, if not fractions of a second. By employing Excel and VBA, our purpose is to try to bring clarity to an area that was previously covered with black boxes.

What started as an attempt to push back the boundaries of Excel through macros turned into a full-scale expedition into the VBA language within Excel and then developed from equities, through options and finally to cover bonds. Along the way we learned scores of new Excel skills and a much greater understanding of the numerical methods implemented across finance.

The genesis of the book came from material developed for the ‘Computer-Based Financial Modelling’ elective on the MBA degree at London Business School. The part on equities formed the basis for an executive course on ‘Equity Portfolio Management’ run annually by the International Centre for Money and Banking in Geneva. The parts on options and bonds comprise a course in ‘Numerical Methods’ on the MSc in Mathematical Trading and Finance at City University Business School. The book is within the reach of both students at the postgraduate level and those in the latter undergraduate years.

There are no prerequisites for readers apart from a willingness to adopt a pro-active stance when using the book—namely by taking advantage of the inherent ‘what-if’ quality of the spreadsheets and by looking at and using the code forming the VBA user-defined functions. Since we assume for the most part that asset returns are lognormal and therefore use binomial trees as a central numerical method, our explanations can be based on familiar results from probability and statistics. Comprehension is helped by the use of a common notation throughout, and transparency by the availability of complete solutions in both Excel and VBA forms.
Our main debt is to the individuals from the academic and practitioner communities in finance who first developed the theory and then the numerical methods that form the material for this book. In the words of Sir Isaac Newton “If I have seen further it is by standing on the shoulders of giants”.

We would also like to thank our colleagues at both London Business School and City University Business School, in particular Elroy Dimson, John Hatgioannides, Paul Marsh and Kiriakos Vlahos.

We would like to thank Sam Whittaker at Wiley for her enthusiasm, encouragement and much needed patience, invaluable qualities for an editor.

Last but not least, we are grateful for the patience of family and friends who have occasionally chivvied us about the book’s somewhat lengthy gestation period.
We hope that our text, Advanced Modelling in Finance, is conclusive proof that a wide range of models can now be successfully implemented using spreadsheets. The models range across the complete spectrum of finance including equities, equity options and bond options spanning developments from the early fifties to the late nineties. The models are implemented in Excel spreadsheets, complemented with functions written using the VBA language within Excel. The resulting user-defined functions provide a portable library of programs with more than sufficient speed and accuracy.

Advanced Modelling in Finance should be viewed as a complement (or dare we say, an antidote) to traditional textbooks in the area. It contains relatively few derivations, allowing us to cover a broader range of models and methods, with particular emphasis on more recent advances.

The major theoretical developments in finance such as portfolio theory in the 1950s, the capital asset pricing model in the 1960s and the Black–Scholes formula in the 1970s brought with them analytic solutions that are now straightforward to calculate. The subsequent decades have seen a growing body of developments in numerical methods. With an intelligent choice of parameters, binomial trees have assumed a central role in the more numerically-intensive calculations now required to value equity and bond options. The centre of gravity in finance now concerns the search for more efficient ways of performing such calculations rather than the theories from yesteryear.

The breadth of the coverage across finance and the sophistication needed for some of the more advanced models are testament to the ability of Excel, the built-in functions contained in Excel and the real programming environment that VBA provides. This allows us to highlight the commonality of assumptions (lognormality), mathematical problems (expectation) and numerical methods (binomial trees) throughout finance as a whole. Without exception, we have tried to ensure a consistent and simple notation throughout the book to reinforce this commonality and to improve clarity of exposition.

Our objective in writing a book that covers the broad range of subjects in finance has proved to be both a challenge and an opportunity. The opportunity has provided us with the chance to overview finance as a whole and, in so doing, to make important connections and bring out commonalities in asset price assumptions, mathematical problems, numerical methods and Excel solutions. In the following sections we summarise a few of these unifying insights that apply to equities, options and bonds with regard to finance, mathematical topics, numerical methods and Excel features. This is followed by a more detailed summary of the main topics covered in each chapter of the book.

1.1 FINANCE INSIGHTS

The genesis of modern finance as a subject separate from economics started with Markowitz’s development of portfolio theory in 1952. Markowitz used utility theory to model the preferences of individual investors and to develop a mean–variance approach
to examining the trade-off between return (as measured by an asset’s mean return) and risk (measured by an asset’s variance of return). This subsequently led to the development by Sharpe, Lintner and Treynor of the capital asset pricing model (CAPM), an equilibrium model describing expected returns on equities. The CAPM introduced beta as a measure of diversifiable risk, arguing that the creation of portfolios served to minimise the specific risk element of total risk (variance).

The next great theoretical development was the equity option pricing formula of Black and Scholes, which rested on the ability to create a (riskless) hedge portfolio. Contemporaneously, Merton extended the Black–Scholes formula to allow for continuous dividends and thus also options on commodities and currencies. The derivation of the original formula required the solving of the diffusion (or heat) equation familiar from physics, but was subsequently encompassed by the broader risk-neutral approach to the valuation of derivatives.

1.2 ASSET PRICE ASSUMPTIONS

Although portfolio theory was derived through individual preferences, it could also have been obtained by making assumptions about the distribution of asset price returns. The standard assumption is that equity returns follow a lognormal distribution—equivalently we can say that equity log returns follow a normal distribution. More recently, practitioners have examined the effect of departures from strict normality (as measured by skewness and kurtosis) and have also proposed different distributions (for example, the reciprocal gamma distribution).

Although bonds have characteristics that are different from equities, the starting point for bond option valuation is the short interest rate. This is frequently assumed to follow the lognormal or normal distribution. The result is that familiar results grounded in these probability distributions can be applied throughout finance.

1.3 MATHEMATICAL AND STATISTICAL PROBLEMS

Within the equities part, the mathematical problems concern optimisation. The optimisation can also include additional constraints, exemplified by Sharpe’s development of returns-based style analysis. Beta is estimated as the slope coefficient in a linear regression.

Options are valued in the risk-neutral framework as statistical expectations. The normal distribution of log equity prices can be approximated by an equivalent discrete binomial distribution. This binomial distribution provides the framework for calculating the expected option value.

1.4 NUMERICAL METHODS

In the context of portfolio optimisation, the optimisation involves portfolio variance, and the numerical method needed for optimisation is quadratic programming. Style analysis also uses quadratic programming, the quantity to be minimised being the error variance. Although not usually thought of as optimisation, linear regression chooses slope coefficients to minimise residual error. Here optimisation is of a different kind, regression analysis, which provides analytical formulas to calculate the beta coefficients.

Turning to option valuation, the binomial tree provides the structure within which the risk-neutral expectation can be calculated. We highlight the importance of parameter
choice by examining the convergence properties of three different binomial trees. Such trees also allow the valuation of American options, where the option can be exercised at any date prior to maturity.

With European options, techniques such as Monte Carlo simulation and numerical integration are also used. Numerical search methods, in particular the Newton–Raphson approach, ensure that volatilities implied by option prices in the market can be estimated.

1.5 EXCEL SOLUTIONS

The spreadsheets demonstrate how Excel can be used as a prototype for building models. Within the individual spreadsheets, all the formulas in the cells can easily be examined and we have endeavoured to incorporate all intermediate calculations in cells of their own. The spreadsheets also allow the hallmark ability to ‘what-if’ by changing parameter values in cells.

The implementation of all the models and methods occurs twice: once in the spreadsheets and once in the VBA functions. This dual approach serves as an important check on the accuracy of the numerical calculations.

Some of the VBA procedures are macros, normally seen by others as the main purpose of VBA in Excel. However, the majority of the procedures we implement are user-defined functions. We demonstrate how easily these functions can be written in VBA and how they can incorporate Excel functions, including the powerful matrix functions.

The Goal Seek and Solver commands within Excel are used in the optimisation tasks. We show how these commands can be automated using VBA user-defined functions and macros. Another under-used aspect of Excel involves the application of array functions (invoked by the Ctrl+Shift+Enter keystroke combination) and we implement these in user-defined functions. To improve efficiency, our binomial trees in user-defined functions use one-dimensional arrays (vectors) rather than two-dimensional arrays (matrices).

1.6 TOPICS COVERED

There are four parts in the book, the first part illustrating the advanced modelling features in Excel followed by three parts with applications in finance. The three parts on applications cover equities, options on equities and options on bonds.

Chapter 2 emphasises the advanced Excel functions and techniques that we use in the remainder of the book. We pay particular attention to the array functions within Excel and provide a short section detailing the mathematics underlying matrix manipulation.

Chapter 3 introduces the VBA programming environment and illustrates a step-by-step approach to the writing of VBA subroutines (macros). The examples chosen demonstrate how macros can be used to automate and repeat tasks in Excel.

Chapter 4 moves on to VBA user-defined functions, which have a crucial role throughout the applications in finance. We emphasise how to deal with both scalar and array variables—as input variables to VBA functions, their use in calculations and finally as output variables. Again, we use a step-by-step approach for a number of examples. In particular, we write user-defined functions to value both European options (the Black–Scholes formula) and American options (binomial trees).

Chapter 5 introduces the first application part, that dealing with equities.

Chapter 6 covers portfolio optimisation, using both Solver and analytic solutions. As will become the norm in the remaining chapters, Solver is used both in the spreadsheet
and automated in a VBA macro. By using the array functions in Excel and VBA, we detail how the points on the efficient frontier can be generated. The development of portfolio theory is divided into three generic problems, which recur in subsequent chapters.

Chapter 7 looks at (equity) asset pricing, starting with the single-index model and the capital asset pricing model (CAPM) and concluding with Value-at-Risk (VaR). This introduces the assumption that asset log returns follow a normal distribution, another recurrent theme.

Chapter 8 covers performance measurement, again ranging from single-parameter measures used in the very earliest days to multi-index models (such as style analysis) that represent current best practice. We show, for the first time in a textbook, how confidence intervals can be determined for the asset weights from style analysis.

Chapter 9 introduces the second application part, that dealing with options on equities. Building on the normal distribution assumed for equity log returns, we detail the creation of the hedge portfolio that is the key insight behind the Black–Scholes option valuation formula. The subsequent interpretation of the option value as the discounted expected value of the option payoff in a risk-neutral world is also introduced.

Chapter 10 looks at binomial trees, which can be viewed as a discrete approximation to the continuous normal distribution assumed for log equity prices. In practice, binomial trees form the backbone of numerical methods for option valuation since they can cope with early exercise and hence the valuation of American options. We illustrate three different parameter choices for binomial trees, including the little-known Leisen and Reimer tree that has vastly superior convergence and accuracy properties compared to the standard parameter choices. We use a nine-step tree in our spreadsheet examples, but the user-defined functions can cope with any number of steps.

Chapter 11 returns to the Black–Scholes formula and shows both its adaptability (allowing options on assets such as currencies and commodities to be valued) and its dependence on the asset price assumptions.

Chapter 12 covers two alternative ways of calculating the statistical expectation that lies behind the Black–Scholes formula for European options. These are Monte Carlo simulation and numerical integration. Although these perform less well for the simple options we consider, each of these methods has a valuable role in the valuation of more complicated options.

Chapter 13 moves away from the assumption of strict normality of asset log returns and shows how such deviation (typically through differing skewness and kurtosis parameters) leads to the so-called volatility smile seen in the market prices of options. Efficient methods for finding the implied volatility inherent in European option prices are described.

Chapter 14 introduces the third application part, that dealing with options on bonds. While bond prices have characteristics that are different from equity prices, there is a lot of commonality in the mathematical problems and numerical methods used to value options. We define the term structure based on a series of zero-coupon bond prices, and show how the short-term interest rate can be modelled in a binomial tree as a means of valuing zero-coupon bond cash flows.

Chapter 15 covers two models for interest rates, those of Vasicek and Cox, and Ingersoll and Ross. We detail analytic solutions for zero-coupon bond prices and options on zero-coupon bonds together with an iterative approach to the valuation of options on coupon bonds.

Chapter 16 shows how the short rate can be modelled in a binomial tree in order to match a given term structure of zero-coupon bond prices. We build the popular
Black–Derman–Toy interest rate tree (both in the spreadsheet and in user-defined functions) and show how it can be used to value both European and American options on zero-coupon bonds.

The final Appendix is a Pandora’s box of other user-defined functions, that are less relevant to the chosen applications in finance. Nevertheless they constitute a useful toolbox, including as they do functions for ARIMA modelling, splines, eigenvalues and other calculation procedures.

1.7 RELATED EXCEL WORKBOOKS

Part I which concentrates on Excel functions and procedures and understanding VBA has three related workbooks, AMFEXCEL, VBSUB and VBFNS which accompany Chapters 2, 3 and 4 respectively.

Part II on equities has three related workbooks, EQUITY1, EQUITY2 and EQUITY3 which accompany Chapters 6, 7 and 8 respectively.

Part III on options on equities has four files, OPTION1, OPTION2, OPTION3 and OPTION4 which accompany Chapters 10, 11, 12 and 13 respectively.

Part IV on bonds has two related workbooks, BOND1 and BOND2 which accompany Chapters 14, 15 and 16 as indicated in the text.

The Appendix has one workbook, OTHERFNS.

1.8 COMMENTS AND SUGGESTIONS

Having spent so much time developing the material and writing this book, we would very much appreciate any comments, suggestions and, dare we say, possible corrections and improvements. Please email mstaunton@london.edu or find your way to www.london.edu/ifa/services/services.html or www.business.city.ac.uk/irmi/mstaunton.html.
Part One

Advanced Modelling in Excel
The purpose of this chapter is to review certain Excel functions and procedures used in the text. These include mathematical, statistical and lookup functions from Excel’s extensive range of functions, as well as much-used procedures such as setting up Data Tables and displaying results in XY charts. Also included are methods of summarising data sets, conducting regression analyses, and accessing Excel’s Goal Seek and Solver. The objective is to clarify and ensure that this material causes the reader no difficulty. The advanced Excel user may wish to skim the content or use the chapter for further reference as and when required. To make the various topics more entertaining and more interactive, a workbook AMFEXCEL.xls includes the examples discussed in the text and allows the reader to check his or her proficiency.

2.1 ACCESSING FUNCTIONS IN EXCEL

Excel provides many worksheet functions, which are essentially calculation routines that have been coded up. They are useful for simplifying calculations performed in the spreadsheet, and also for combining into VBA macros and user-defined functions (topics covered in Chapters 3 and 4).

The Paste Function button (labelled $f_x$) on the standard toolbar gives access to them. (It was previously known as the function wizard.) As Figure 2.1 shows, functions are grouped into different categories: mathematical, statistical, logical, lookup and reference, etc.

![Paste Function dialog box showing the COMBIN function in the Math category](image)
Here the Math & Trig function COMBIN has been selected, which produces a brief description of the function’s inputs and outputs. For a fuller description, press the Help button (labelled ?).

On clicking OK, the Formula palette appears providing slots for entering the appropriate inputs, as in Figure 2.2. The required inputs can be keyed into the slots (as here) or ‘selected’ by referencing cells in the spreadsheet (by clicking the buttons to collapse the Formula palette). Note that the palette can be dragged away from its standard position. Clicking the OK button on the palette or the tick on the Edit line enters the formula in the spreadsheet.

As well as the Formula palette with inputs for function COMBIN, Figure 2.2 shows the construction of the cell formula on the Edit line, with the Paste Function button depressed (in action). Notice also the Paste Name button (labelled =ab) which facilitates pasting of named cells into the formula. (Attaching names to ranges and referencing cell ranges by names is reviewed in section 2.10.)

As well as all Excel functions, the Paste Function button also provides access to the user-defined category of functions which are described in Chapter 4.

Having discussed how to access the functions, in the following sections we describe some specific mathematical and statistical functions.

### 2.2 MATHEMATICAl FUNCTIONS

Within the Math & Trig category, we make use of the EXP(x), LN(x), SQRT(x), RAND(), FACT(x) and COMBIN(number, number_chosen) functions.

- EXP(x) returns values of the exponential function, exp(x) or e^x. For example:
  - EXP(1) returns value of e (2.7183 when formatted to four decimal places)
  - EXP(2) returns value of e^2 (7.3891 to four decimal places)
  - EXP(-1) returns value of 1/e or e^{-1} (0.36788 to five decimal places)
In finance calculations, cash flows occurring at different time periods are converted into future (or present) values by applying compounding (or discounting) factors. With continuous compounding at rate $r$, the compounding factor for one year is $\exp(r)$, and the equivalent annual interest rate $r_a$, if compounding were done on an annual basis, is given by the expression:

$$r_a = \exp(r) - 1$$

Continuous compounding and the use of the EXP function is illustrated further in section 2.7.1 on Data Tables.

$\ln(x)$ returns the natural logarithm of value $x$. Note that $x$ must be positive, otherwise the function returns #NUM! for numeric overflow. For example:

- $\ln(0.36788)$ returns value $-1$
- $\ln(2.7183)$ returns value $1$
- $\ln(7.3891)$ returns value $2$
- $\ln(-4)$ returns value #NUM!

In finance, we frequently work with (natural) log returns, applying the $\ln$ function to transform the returns data into log returns.

$\sqrt{x}$ returns the square root of value $x$. Clearly, $x$ must be positive, otherwise the function returns #NUM! for numeric overflow.

$\text{RAND}()$ generates a uniformly distributed random number greater than or equal to zero and less than one. It changes each time the spreadsheet recalculates. We can use RAND() to introduce probabilistic variability into Monte Carlo simulation of option values.

$\text{FACT}(n)$ returns the factorial of the number, which equals $1 \times 2 \times 3 \times \ldots \times n$.

For example:

- $\text{FACT}(6)$ returns the value 720
- $\text{COMBIN}(n, r)$ returns the number of combinations (subsets of size $r$) that can be made up from a ‘number’ of items. The subsets can be in any internal order. For example, if a share moves either ‘up’ or ‘down’ at four discrete times, the number of sequences with three ups (and one down) is:

$$\text{COMBIN}(4, 1) = 4 \text{ or equally } \text{COMBIN}(4, 3) = 4$$

that is the four sequences ‘up-up-up-down’, ‘up-up-down-up’, ‘up-down-up-up’ and ‘down-up-up-up’. In statistical parlance, $\text{COMBIN}(n, r)$ is the number of combinations of three items selected from four and is usually denoted as $\binom{4}{3}$ (or in general, $\binom{n}{r}$).

Excel has functions to transpose matrices, to multiply matrices and to invert square matrices. The relevant functions are:

- $\text{TRANSPOSE}(\text{array})$ which returns the transpose of an array
- $\text{MMULT}(\text{array1}, \text{array2})$ which returns the matrix product of two arrays
- $\text{MINVERSE}(\text{array})$ which returns the matrix inverse of an array

These fall in the same Math category. Since some readers may need an introduction to matrices before examining the functions, this material has been placed at the end of the chapter (see section 2.13).
2.3 STATISTICAL FUNCTIONS

Excel has several individual functions for quickly summarising the features of a data set (an ‘array’ in Excel terminology). These include AVERAGE(array) which returns the mean, STDEV(array) for the standard deviation, MAX(array) and MIN(array) which we assume are familiar to the reader.

To obtain the distribution of a moderate sized data set, there are some useful functions that deserve to be better known. For example, the QUARTILE function produces the individual quartile values on the basis of the percentiles of the data set and the FREQUENCY function returns the whole frequency distribution of the data set after grouping.

Excel also provides functions for a range of different theoretical probability distributions, in particular those for the normal distribution: NORMSDIST and NORMSINV for the standard normal with zero mean and standard deviation one; NORMDIST and NORMINV for any normal distribution.

Other useful functions in the statistical category are those for two variables, which provide many individual quantities used in regression and correlation analysis. For example:

- INTERCEPT(known_y’s, known_x’s)
- SLOPE(known_y’s, known_x’s)
- RSQ(known_y’s, known_x’s)
- STEYX(known_y’s, known_x’s)
- CORREL(array1, array2)
- COVAR(array1, array2)

There is also a little known array function, LINEST(known_y’s, known_x’s), which returns the essential regression statistics in array form. Most of these functions are examined in more detail in section 2.11 on regression. Their performance is compared and contrasted with the regression output from the Data Analysis Regression procedure.

In the next section, we explain how to use the FREQUENCY, QUARTILE and various normal functions via examples in the Frequency and SNorm sheets of the AMFEXCEL workbook.

2.3.1 Using the Frequency Function

FREQUENCY(data_array, bins_array) counts how often values in a data set occur within specified intervals (or ‘bins’), and then returns these frequencies in a vertical array. The bins_array is the set of intervals into which the values are grouped. Since the function returns output in the form of an array, it is necessary to mark out a range of cells in the spreadsheet to receive the output before entering the function.

We explain how to use FREQUENCY with an example set out in the Frequency sheet of the AMFEXCEL workbook. As shown in Figure 2.3, monthly returns and log returns (using the LN function) in columns D10:D71 and E10:E70 have been summarised in rows 4 to 7. Suppose the aim is to get the frequency distribution of the log returns (E10:E71), i.e. the so-called ‘data_array’. The objective might be to check that these returns are approximately normally distributed. First, we have to decide on intervals (or bins) for grouping the data. Inspection of the maximum and minimum log returns suggests about 10 to 12 intervals in the range –0.16 to +0.20. The ‘interval’ values, which have
been entered in range G5:G14, act as upper limits when the log returns are grouped into the so-called ‘bins’.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Returns for months 1 - 62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Summary Statistics:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Returns</td>
<td>Ln Returns</td>
<td>Frequency Distribution:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mean</td>
<td>1.78%</td>
<td>0.014</td>
<td>interval</td>
<td>freq</td>
<td>%freq</td>
<td>%cum freq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>St Dev</td>
<td>8.09%</td>
<td>0.080</td>
<td>-0.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Max</td>
<td>21.23%</td>
<td>0.193</td>
<td>-0.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Min</td>
<td>-14.21%</td>
<td>-0.153</td>
<td>-0.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9</th>
<th>Month</th>
<th>Returns</th>
<th>Ln Returns</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Feb-92</td>
<td>1</td>
<td>7.06%</td>
<td>0.0682</td>
</tr>
<tr>
<td>11</td>
<td>Mar-92</td>
<td>2</td>
<td>-11.54%</td>
<td>-0.1226</td>
</tr>
<tr>
<td>12</td>
<td>Apr-92</td>
<td>3</td>
<td>7.77%</td>
<td>0.0748</td>
</tr>
<tr>
<td>13</td>
<td>May-92</td>
<td>4</td>
<td>10.66%</td>
<td>0.1013</td>
</tr>
<tr>
<td>14</td>
<td>Jun-92</td>
<td>5</td>
<td>-11.72%</td>
<td>-0.1247</td>
</tr>
<tr>
<td>15</td>
<td>Jul-92</td>
<td>6</td>
<td>-8.26%</td>
<td>-0.0862</td>
</tr>
<tr>
<td>16</td>
<td>Aug-92</td>
<td>7</td>
<td>-2.89%</td>
<td>-0.0293</td>
</tr>
<tr>
<td>17</td>
<td>Sep-92</td>
<td>8</td>
<td>9.93%</td>
<td>0.0947</td>
</tr>
<tr>
<td>18</td>
<td>Oct-92</td>
<td>9</td>
<td>12.65%</td>
<td>0.1191</td>
</tr>
</tbody>
</table>

![Table of Summary Statistics]

Figure 2.3 Layout for calculating the frequency distribution of log returns data

To enter the FREQUENCY function correctly, select the range H5:H15. Then start by typing = and clicking on the Paste Function button (labelled f) to complete the function syntax:

```
=SUM(E10:E71,G5:G14)
```

After adding the last bracket ‘)’, with the cursor on Excel’s Edit line, enter the function by holding down the Ctrl then the Shift then the Enter keys. (You need to use three fingers, otherwise it will not work. If this fails, keep the output range of cells ‘selected’, press the Edit key (F2), edit the formula if necessary, then press Ctrl+Shift+Enter once again.)

You should now see the function enclosed in curly brackets {} in the cells, and the frequencies array in cells G5:G15. The results are in Figure 2.4. Use the SUM function in cell H17 to check that the frequencies sum to 62.

Interpreting the results, we can see that there were no log returns below -0.16, six values in the range -0.16 to -0.12 and no values above 0.20. (The bottom cell in the FREQUENCY array, G15, contains any values above the bins’ upper limit, 0.20.)

Since the FREQUENCY function has array output, individual cells cannot be changed. If a different number of intervals is required, the current array must be deleted and the function entered again.

It helps to convert the frequencies into percentage frequencies (relative to the size of the data set of 62 values) and then to calculate cumulated percentage frequencies as shown in columns I and J in Figure 2.4. The percentage frequency and cumulative percentage frequency formulas can be examined in the Frequency sheet.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns for months 1 - 62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary Statistics:</td>
<td>Returns</td>
<td>Ln Returns</td>
<td>Frequency Distribution:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>1.78%</td>
<td>0.0145</td>
<td>interval</td>
<td>freq</td>
<td>%freq</td>
<td>%cum freq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>St Dev</td>
<td>8.09%</td>
<td>0.0802</td>
<td>0.16</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>21.23%</td>
<td>0.1925</td>
<td>-0.12</td>
<td>6</td>
<td>10%</td>
<td>10%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>-14.21%</td>
<td>-0.1533</td>
<td>-0.08</td>
<td>4</td>
<td>6%</td>
<td>16%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.04</td>
<td>7</td>
<td>11%</td>
<td>27%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Month</td>
<td>Returns</td>
<td>Ln Returns</td>
<td>0.00</td>
<td>10</td>
<td>16%</td>
<td>44%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb-92</td>
<td>1</td>
<td>7.06%</td>
<td>0.0682</td>
<td>0.04</td>
<td>5</td>
<td>8%</td>
<td>52%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mar-92</td>
<td>2</td>
<td>-11.54%</td>
<td>-0.1226</td>
<td>0.08</td>
<td>16</td>
<td>26%</td>
<td>77%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apr-92</td>
<td>3</td>
<td>7.77%</td>
<td>0.0748</td>
<td>0.12</td>
<td>9</td>
<td>15%</td>
<td>92%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May-92</td>
<td>4</td>
<td>10.66%</td>
<td>0.1013</td>
<td>0.16</td>
<td>4</td>
<td>6%</td>
<td>98%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jun-92</td>
<td>5</td>
<td>-11.72%</td>
<td>-0.1247</td>
<td>0.20</td>
<td>1</td>
<td>2%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul-92</td>
<td>6</td>
<td>-8.26%</td>
<td>-0.0862</td>
<td>0.00</td>
<td>0</td>
<td>0%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aug-92</td>
<td>7</td>
<td>-2.89%</td>
<td>-0.0293</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep-92</td>
<td>8</td>
<td>9.93%</td>
<td>0.0947</td>
<td>Total</td>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.4** Frequency distribution of log returns with % frequency and cumulative distributions

The best way to display the percentage cumulative frequencies is an XY chart with data points connected by a smooth line with no markers. To produce a chart like that in Figure 2.5, select ranges G5:G14 and J5:J14 as the source data. Note that, to select non-contiguous ranges, select the first range, then hold down the Ctrl key whilst selecting the second and subsequent ranges.

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Distribution:</td>
<td>interval</td>
<td>freq</td>
<td>%freq</td>
<td>%cum freq</td>
<td>theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.16</td>
<td>0</td>
<td>0%</td>
<td>0%</td>
<td>1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.12</td>
<td>6</td>
<td>10%</td>
<td>10%</td>
<td>5%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-0.08</td>
<td>4</td>
<td>10%</td>
<td>16%</td>
<td>12%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.04</td>
<td>7</td>
<td>11%</td>
<td>27%</td>
<td>25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.00</td>
<td>10</td>
<td>16%</td>
<td>44%</td>
<td>45%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.04</td>
<td>5</td>
<td>8%</td>
<td>52%</td>
<td>62%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.08</td>
<td>16</td>
<td>26%</td>
<td>77%</td>
<td>79%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.12</td>
<td>9</td>
<td>15%</td>
<td>92%</td>
<td>91%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.16</td>
<td>4</td>
<td>6%</td>
<td>98%</td>
<td>97%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.20</td>
<td>1</td>
<td>2%</td>
<td>100%</td>
<td>99%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.24</td>
<td>0</td>
<td>0%</td>
<td>100%</td>
<td>45%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2.5** Chart of cumulative % frequencies (actual and strictly normal data)

For normally distributed log returns, the cumulative distribution should be sigmoid in shape (as indicated by the dashed line). The actual log returns data shows some departure from normality, possibly due to skewness.

### 2.3.2 Using the Quartile Function

QUARTILE(array, quart) returns the quartile of a data set. The second input ‘quart’ is an integer that determines which quartile is returned: if 0, the minimum value of the array; if 1, the first quartile (i.e. the 25th percentile of the array); if 2, the median value (50th percentile); if 3, the third quartile (75th percentile); if 4, the maximum value.