Advanced Financial Modelling in Excel

Danielle Stein Fairhurst
About this Publication

This booklet is an extract from “Using Excel for Business Analysis: a Fundamental Approach to Financial Modelling” by Danielle Stein Fairhurst and due for publication by Wiley Finance in April 2012. Chapters 1 & 2 have not been included as they are not directly referred to in the Advanced Financial Modelling Online Course.

A full copy of this book is available in hard copy or Kindle version will be available after the publishing date at www.amazon.com or http://au.wiley.com/WileyCDA/WileyTitle/productCd-111813284X,descCd-buy.html
# Read This First

<table>
<thead>
<tr>
<th>Who this course is for...</th>
<th>This course is designed for business professionals who need to set out a business case, analyse the performance of a business or unit, or perform complex manipulations of financial information and data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>What skills and knowledge you'll learn...</td>
<td>The course focuses on utilising advanced formulas in a financial modelling environment and shows how to build a model which minimises errors. Participants will build a financial model from scratch which they can take away with them for future reference.</td>
</tr>
<tr>
<td>What you'll need to know before beginning...</td>
<td>Participants should use Excel on a regular basis and be comfortable with using advanced functions and linking. This course covers the practical usage of Excel in an analytical context. It is designed for users who are comfortable with commonly used Excel tools and techniques. Whilst a brief overview of tools used in this course is provided, it is assumed that delegates are confident Excel users.</td>
</tr>
<tr>
<td>Software used...</td>
<td>Instructions are given using both Microsoft Excel 2003 and 2007/10. You will see this symbol in front of the Excel 2003 instructions, and this one for Excel 2007/10. Excel for Mac users are also catered for, and those instructions are prefaced by this symbol. Students can choose their preferred version, however, you will be encouraged to try out Excel 2010 if you have not done so before.</td>
</tr>
<tr>
<td>Passwords.....</td>
<td>You will be provided with several Excel models when attending this course. If any of the files are password protected, the password will be “plumsolutions”.</td>
</tr>
</tbody>
</table>
# Table of Contents

About this Publication .................................................................................................................. 2  

CHAPTER 3 Principles of Best Practice ..................................................................................... 11  

Best Practice in Financial Modelling ....................................................................................... 11  

Document your assumptions .................................................................................................... 11  

Linking, not hard-coding .......................................................................................................... 11  

Only enter data once ............................................................................................................... 12  

Bad Habits ............................................................................................................................... 12  

Use Consistent Formulas ........................................................................................................ 13  

Format & Label Clearly .......................................................................................................... 13  

CHAPTER 4 Financial Modelling Techniques .......................................................................... 15  

How long should a Formula be? ............................................................................................. 15  

Common Excel Error Values ................................................................................................. 17  

Avoid Error Displays in Formulas .......................................................................................... 19  

Circular References ............................................................................................................... 20  

How to Fix Circular References ............................................................................................ 20  

Circular References in Interest Calculations ......................................................................... 22  

Enabling Iterative Calculations .............................................................................................. 23  

CHAPTER 5 Using Excel in Financial Modelling ..................................................................... 25  

Excel Versions ........................................................................................................................ 25  

History of Excel Versions ...................................................................................................... 25  

Current Excel Usage .............................................................................................................. 25  

Which Version am I Using? .................................................................................................... 26  

The Upgrade ........................................................................................................................... 28  

Major changes between versions ........................................................................................... 28
Completely New in 2007 ................................................................. 29
Excel 2007/2010 Compatibility .................................................. 30
Upgrading from Excel 2007 to Excel 2010 ................................ 31
Major changes between versions ............................................. 31
Improvements to the charts between versions ......................... 34
New calculation and function features .................................... 35
Additional changes made to Excel 2010 ................................. 35
Filter Options ........................................................................... 36
Should you Upgrade? ................................................................. 37
Handy Excel Shortcuts ............................................................... 38
 Editing ...................................................................................... 38
 Navigating ................................................................................ 39
 Formatting ................................................................................ 39
 In Formulas ............................................................................... 40
 Auditing .................................................................................... 40
 Mac Excel Shortcuts ................................................................. 40
 Editing ...................................................................................... 41
 Navigating ................................................................................ 41
 Formatting ................................................................................ 42
 In Formulas ............................................................................... 42
 Auditing .................................................................................... 42
CHAPTER 6 Functions for Financial Modelling ......................... 44
Lookup Formulas ...................................................................... 44
VLOOKUP (Vertical Lookup) .................................................. 44
#N/A Errors in VLOOKUPs ....................................................... 46
Breaking a VLOOKUP ........................................................................................................... 47
Improving VLOOKUPS with MATCH ................................................................................ 47
HLOOKUP (Horizontal Lookup) ........................................................................................ 48
LOOKUP Function ............................................................................................................... 49
So, which LOOKUP Function should I use? ..................................................................... 51
Using INDEX and MATCH to create a More Robust Formula ........................................... 53
OFFSET Function .............................................................................................................. 55
Using an OFFSET to Model Cashflow ................................................................................ 57
Nesting OFFSET and COLUMN Formulas .......................................................................... 58
CHOOSE Function ............................................................................................................. 59
Using a Nested CHOOSE Formula with Dates ................................................................. 60
Financial Functions .......................................................................................................... 61
Project Evaluation Functions ............................................................................................ 61
NPV (Net Present Value) .................................................................................................. 61
What Cost of Capital should we use? .............................................................................. 61
What is wrong with the NPV Function? ............................................................................ 62
IRR (Internal Rate of Return) .......................................................................................... 63
Comparing IRR with the risk factor .................................................................................. 64
Using NPV and IRR to Make Decisions .......................................................................... 64
The Problem with IRR ....................................................................................................... 64
What difference does an X make? XNPV & XIRR ......................................................... 65
CHAPTER 7 Tools for Model Display ................................................................................ 68
Conditional Formatting .................................................................................................... 68
To apply Conditional Formatting ..................................................................................... 68
To Remove Conditional Formatting ................................................................................. 69
Data Bars .................................................................................................................................................. 70
Icon Sets and Colour Scales .................................................................................................................... 70
Enhancements to Conditional Formatting in Excel 2010 ......................................................................... 72
Sparklines .............................................................................................................................................. 74
Editing Sparklines .................................................................................................................................. 75
Changing Properties for group of sparklines ............................................................................................ 77

Form Controls ......................................................................................................................................... 78
Accessing Form Controls .......................................................................................................................... 78
Checkboxes .............................................................................................................................................. 78
Option Button ......................................................................................................................................... 82
Spin Buttons ........................................................................................................................................... 85
Combo Boxes ........................................................................................................................................... 86
Boolean Logic (Binary Code) in Form Controls ....................................................................................... 89
Form Controls vs. ActiveX Controls ......................................................................................................... 90

CHAPTER 8 Tools for Financial Modelling ................................................................................................. 91

Array Formulas ....................................................................................................................................... 91
Advantages of using Array Formulas ........................................................................................................ 91
Disadvantages of using Array Formulas .................................................................................................... 92
Simple Array Formula Example ................................................................................................................ 92
Uses of Array Formulas ............................................................................................................................. 93
Transposing Data using an Array ................................................................................................................ 94
Using a TRANSPOSE Array Function ....................................................................................................... 96

Pivot Tables ............................................................................................................................................... 97
The use of Pivot Tables in Financial Modelling ........................................................................................ 97
Common Uses of Pivot Tables ................................................................................................................ 98
Removing Redundant Assumptions and Source Data in a Model ........................................ 127
Formula Auditing .................................................................................................................. 128
Tracing Precedents ............................................................................................................... 128
Tracing Dependents ............................................................................................................. 130
Error Checking Tool ............................................................................................................ 131
Evaluating Formula Tool ...................................................................................................... 134
Viewing all the Formulas ...................................................................................................... 136
Direct Editing ......................................................................................................................... 137

CHAPTER 11 Scenarios & Sensitivity Analysis ................................................................. 140
Overview of Scenario Analysis Tools and Methods .............................................................. 140
Manual drop-downs ............................................................................................................. 140
In-cell drop-downs (using data validation) .......................................................................... 140
Object drop-down Boxes (using combo boxes) ..................................................................... 141
Scenario Manager .................................................................................................................. 141
Data Tables ............................................................................................................................ 142
Scenario Manager .................................................................................................................. 143
Using Data Tables for Sensitivity Analysis ........................................................................ 144
One-variable Data Table ........................................................................................................ 146
Two-variable Data Table ........................................................................................................ 147
Limitations of Data Tables .................................................................................................... 149
Advanced Conditional Formatting ......................................................................................... 150
Modifying Conditional Formatting ....................................................................................... 153
Comparing Scenario Methods .............................................................................................. 153
Creating the Working Model ............................................................................................... 153
Manual Sensitivity Analysis ................................................................................................. 155
Creating Built in Scenarios using a Data Validation Drop-down ........................................... 155
Creating Scenarios using a Combo Box Drop-down .................................................. 158
What’s the Difference between a Data Validation and Combo Box Drop-down? .......... 160
Creating Scenarios using a Two Variable Data Table .............................................. 161
CHAPTER 12 Presenting Model Output .................................................................. 164
Charting with Two Different Axes and Chart Types .............................................. 164
Waterfall Charts ......................................................................................... 168
Creating a Basic Waterfall Chart (with only positive values) ............................... 168
Creating a Waterfall Chart (with Positive and Negative Values) ......................... 171
Profitability Waterfall Chart ........................................................................... 174
Creating a Complex Waterfall Chart (with Positive and Negative Values Crossing the X Axis) .......................................................... 175
Additional Resources and Research ............................................................. 178
Where to From Here .................................................................................. 178
Web Links ................................................................................................. 178
White Papers & Other Documents ................................................................... 179
Other Books ............................................................................................. 179
CHAPTER 3 Principles of Best Practice
Best Practice in Financial Modelling

The principles of best practice outlined here are for the purpose of reducing errors, making a model easier to read, audit, update and use for its intended purpose. This list is by no means exhaustive, but outlines the most important principles in Financial Modelling. By following these key principles, your model is easier to navigate and check, and much more likely to be robust, accurate, reliable and error-free.

Document your assumptions

- Your model design, layout and structure can be perfect, but its validity is really reliant on the assumptions that go into it. i.e. Garbage in, Garbage out. The most beautifully structured model is the world is pretty useless if the assumptions that go into it are garbage!

- Assumptions documentation helps with validation & avoids misinterpretation. If there is any possible misunderstanding about why, how or what the assumptions are in the model, make sure it’s down in black and white on the assumptions page.

- List assumptions on a separate page, clearly labelled. For a smaller model, you might decide to mix source data and assumptions together, or they could be separated in a large model.

- The more detail the better! If your assumptions are extremely detailed, you could have a “key assumptions” section where you summarise the important assumptions but leave all the detail in case it is needed.

- Document source data and unique calculations as you go. Don’t leave it all to the end – and don’t rely on your memory! Months later you, or another modeller will have no idea why you calculated something a certain way. Make sure it’s well-documented.

Linking, not hard-coding

- Link as much as possible so that when the inputs change, the outputs also change (pretty much the whole point of Financial Modelling!)

- Use Named Ranges when linking to external files. See the section on “Linking to External Files” for more detail on this.
• No hard coding except for input variables unless referenced or source data is listed or it is otherwise *blindly obvious* where you got the number from.

• By linking, you can trace source data back through the links, so making your model auditable, traceable and easy to validate.

• Never use a value within a formula. The only exceptions are those things which are “standard” or commonly accepted values which will not change, such as:
  - 24 hours in a day
  - 7 days in a week
  - 60 minutes in an hour

• You may notice that many modellers put input variables in **blue font**. Whilst this is a bit old-fashioned, it’s still quite standard practice and it’s a kind of signal to your user or another modeller that this is a hard-coded input variable that can change (i.e. you are allowed to change this value!) The problem with blue font though is that you can’t see it when there is no value in the cell, so many modellers use a beige background with blue font. See the section on “Workbook Anatomy” for more detail on colour coding.

**Only enter data once**

• Never enter the same value twice; enter it in once as a source and always reference to that one cell.

• Never use a value within a formula eg. =IF(G$6>=$E7,136800,0) should have a link, not 136800 typed in.

• However, do not link source data to a cell which itself has been linked to the original source. Always go directly to the source data. The result of this malpractice has been aptly termed ‘Daisy Chaining’ by F1F9 / Financial Mechanics. It is an inefficient and error prone (yet tempting!) practice which is should be avoided when building financial models.

**Bad Habits**

There are a few bad habits financial modellers can get into:

• Hit “enter” or the green tick after editing a formula; don’t just click elsewhere. It’s a bad habit many people have, and one that can cause unintentional errors

• Highlight the range only, not the entire row or column – using the entire row or column is tempting but uses too much memory
Daisy chaining links throughout the entire model instead of linking to the source, as described above.

**Use Consistent Formulas**

- In a table or block of data, the same formula exists **all the way** across and down the entire block.
- Use Absolute / Mixed Referencing to achieve this. (See the section on “Cell Referencing” for a practical exercise on how to do this.)
- It saves time, and avoids error – and this practice is a keystone to good financial modelling technique. If you only pick up one modelling technique from this book – this is it!
- If you really need to change formulas so that it is different to the cell adjacent to it, you can indicate this with a double border to show that the formula has been broken. This will help with auditing. (See the section in “Model Design” on Colour Coding for an example of how this looks.)
- Avoid blank rows or columns in data blocks as this can cause problems with sorting, filters and pivot tables.
- If sheets are similar, use a template where possible and copy the same sheet. Then when you need to update it, group the sheets to make global changes.
- Try not to use macros to perform lengthy calculations or manipulations as this reduces transparency. Only in limited areas should macros be used, such as goal seek on the scenario generation page.

**Format & Label Clearly**

- Format cells appropriately. Use symbols for currency $€¥, percentages % etc and commas for thousands – this makes your model easier to read and avoid mistakes and misinterpretations.
- Label your data clearly. It sounds simple, but mixing units (e.g. mixing apples and oranges) is a common source of error in financial modelling, and good formatting and labelling will avoid this.
- Include a dedicated units column (eg, column C) and make sure the units are entered into that column, eg, $’000, MWh, litres, headcount etc.
• If you decide to round figures into thousands, show this **clearly** at the top of the row with a descriptive heading such as “Revenue $’000” to avoid confusion and misinterpretation.

• Column and row headings should have unit or currency headings and only contain one type of unit or currency.

• When building multi-sheet models, columns should be used consistently. For example, if building a model for the period 2013 – 2020, use column F on each worksheet for 2013, column G for 2014, and so on.

• It is often a good idea to reserve a column (eg, Column D) for constants that apply to all years, months or days. For example, if growth rate is 5%, have that in column D, and then link all calculations to column D.

In general these best practice guidelines outlined above are really just common sense. Whilst some of them may seem tedious and overly prescriptive to the beginner, most of these points are for the purpose of reducing error and bringing increased robustness and clarity to the model. These practices are probably second nature to the experienced financial modeller who will most likely already follow these guidelines by instinct.
CHAPTER 4 Financial Modelling Techniques

How long should a Formula be?

If you’ve been modelling for a while you’ll know that it’s pretty easy to end up with a really long formula. Excel will handle them—as long as they are constructed correctly—but they can be pretty tricky to follow—especially for other people. Even if you’ve written it yourself, you’ll struggle to follow it at a later date.

As with most concepts in financial modelling; a formula should be as simple as possible and as complex as necessary!

The way to make a formula shorter of course, is to break in down into several steps, so adding additional “working” cells. This can make your model cumbersome if there is too much of this. Therefore:

The good modeller needs to strike a balance between:

1. Making the formula too long and complex; and
2. Having a model with additional working rows and columns

If you do have a very long formula, one way to make it a bit easier to understand is to use Alt+Enter in the middle of the formula to force the next section of the formula onto another line. Note that this shortcut is Control+Option+Enter in Excel for Mac 2011.

For example:

![Figure 4.5 Very long formula broken in several lines](image-url)
Of course, a formula of this length is not recommended! One of the new features of Excel 2007/10 is the expandable formula bar, but it is generally accepted best practice that a formula should not exceed about half of the width of the formula bar. This does not include file paths if the formula is linking to an external file which is currently closed, and therefore the full file path is included in the formula as shown below. Keep formulas particularly short when linked to external files for this reason.

Fig 4.6 Formula linking to an closed external file

Below is an example of two different ways of achieving the same result. The revenue has been calculated by:
Row 3: Forecasting the number of customers
Row 4: Forecasting how many calls per day each will make
Row 5: Applying seasonality to these calls
Row 6: Multiply row 3 and row 5 to get the total number of calls per day
Row 7: Multiply row 6 by the number of days in the month to get the number of calls per month
Row 8: Multiply row 7 by the price to get the total revenue.

Fig 4.7 Revenue calculated two different ways
Alternatively, the whole calculation could be done in row 9 using the complex formula:

\[ \text{Assumptions!B9} \times (1 + \text{cust\_growth})^{(B2 - B8)} \times \text{B4 \times Assumptions!B6 \times Assumptions!B4 \times price} \]

You could argue that this is a more succinct, sparse way of modelling. However, it makes it very difficult to take apart this formula for checking or debugging. Calculating the revenue step by step as shown in the top example is a much better way of modelling.

Common Excel Error Values

Excel error values should not be confused with error checks. Error checks are formulas created by the modeller which have been designed to alert the user as soon as the model is not calculating correctly. Excel error values are errors displayed by Excel when the formula has not been built properly. For example, using the multiplication symbol “x” instead of “*” e.g. =I36xJ36 is an incorrect formula which will return the value #NAME? because Excel does not recognise it. Other common sources of error are parentheses (brackets) that do not match or missing sections for functions.

If you get an Excel error message, click OK on the error message’s dialogue box. Then either edit the formula on the Formula Bar to correct the formula, or click the Cancel button to clear it.

Here are some error values you might get:

<table>
<thead>
<tr>
<th>Error Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#######</td>
<td>The railroad tracks; not really an error. The result is too long to fit in the cell, so just make the column wider.</td>
</tr>
<tr>
<td>#DIV/0!</td>
<td>You’re trying to divide by zero. Check that the formula is dividing by is not empty, as this will translate to a zero value.</td>
</tr>
<tr>
<td>#NAME?</td>
<td>There’s a name in the formula Excel doesn’t recognise. You may have named a range, and then spelt it incorrectly, or typed “VLOOKKUP” instead of “VLOOKUP”. If you used a name you defined, check its spelling. You can avoid this error by selecting a name in the Name Box instead of typing it in. If you typed in a function, check its spelling or verify that such a function exists. If you are performing operations on text, make sure you have enclosed it in double quotation marks.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>#REF!</td>
<td>Your formula refers to a cell that no longer exists, due to a change in the worksheet. This happens if you deleted cells referred to in the formula or pasted moved cells on cells referred to in the formula. This is probably one of the more difficult errors to fix because you no longer have access to the cell that was deleted. You will need to re-enter the cell reference which is missing.</td>
</tr>
<tr>
<td>#VALUE!</td>
<td>The formula is using a different type of reference to what the function was expecting. The most common cause of this is if a user tries to enter text into a cell instead of a number, or the formula is referring to a label instead of the numeric value. For example: “TBA” or “Less than 10” instead of a numerical value. Check to see that some fields are not referring to cells containing text.</td>
</tr>
<tr>
<td>#N/A</td>
<td>This means that there is some data missing; the value the formula looking for is not available. A common reason for receiving this error is if a LOOKUP cannot find the criteria it is searching for in the source data. Make sure that the function can find the data you’ve asked it to look for.</td>
</tr>
<tr>
<td>Circular Reference</td>
<td>This means you are trying to link a formula to itself somehow. You will see Circular followed by a cell reference in the area below the worksheet. Circular references often cause models to stop calculating. See below for more information on Circular References.</td>
</tr>
</tbody>
</table>
Avoid Error Displays in Formulas

Sometimes a formula may return an error message. Usually, you will want to know when a formula error occurs. But now and then you may prefer to avoid the messages. For example, if you are calculating a percentage in a column and one of the entry fields contain a zero for example, you will get a #DIV/0! error. Or if using a VLOOKUP function, the criteria you have entered does not exist in the data, a #N/A error will appear. You can stop these errors from appearing by using an IF() function to check for an error.

=IF(ISERROR(VLOOKUP("payback",Payback!B1:C109,2,0)),0,VLOOKUP("payback",Payback!B1:C109,2,0))

You can adapt this technique to any operation. The original formula serves as the argument for the ISERROR() function, and it repeats as the last argument of the IF() function. Like this:

=IF(ISERROR(OriginalFormula),"",OriginalFormula)

**Hot Tip:** to make this quicker, you’ll find using the copy and paste shortcuts Control-C and Control-V (Command-C and Command-V in Excel for Mac 2011) will make building this formula much more efficient.

Note that many modellers do not agree with avoiding error displays, and recommend that you always show errors. The reason for this is that using error suppression can mask other, genuine errors, so use this function with caution.

**[Excel 2007/10 and Excel for Mac 2011]** If you are using the new version of Excel, you will be able to use the new =IFERROR formula. This is much more concise, as it does not require the entire formula to be repeated.

The formula shown above would instead look like this:

=IFERROR(VLOOKUP("payback",Payback!B1:C109,2,0),0)

**Only use the IFERROR formula if you are sure that your model will not be opened using Excel 2003!**
Circular References

If you have been using Excel for any length of time, you’ve probably come across a circular reference. If your formula is trying to refer to itself, you’ll end up with a circular reference. A common (and easily fixed) cause of this is when a sum range includes the sum itself. For example, the formula below in cell B11 is referring to cell B11, and this means that the formula cannot calculate properly.

![Formula creating a circular reference](image)

You cannot rely on a model which contains circular references – the calculation may not work, and you can’t be sure the other formulas are calculating properly.

How to Fix Circular References

You can have intentional circular formulas, (in which case you’ll have to enable iterative calculations – see below) but if your circular reference is unintentional, you’ll have to get rid of it as soon as possible. The older it is, the harder it is to track down and fix!

You’ll get a warning message as soon as the circular reference happens so the simplest way to fix this is to stop and simply undo the last thing you did. You really just need to find the offending cell and remove the link. Sounds simple, but sometimes it’s not that easy! The notification in the status bar in the bottom left hand corner may help find the cell causing the problem:
However, you need to be on the correct tab in order for this message to show. If you aren’t sure which tab the circular reference is on, you need hunt it down by going to each tab to see if it shows on the status bar. Alternatively, you can use the auditing tool to locate the circular reference.

**Locating a Circular Reference with the Auditing Tool**

**[Excel 2007/10]** On the Formulas tab, in the Formula Auditing group, click the arrow on the Error Checking button, select Circular References, and this will show the circular reference in the model. Click on the reference to go to the circular reference.

**Fig 4.15 Finding the Circular Reference Auditing Tool in the Ribbon in Excel 2010**

**[Excel 2003]** Show the Circular Reference toolbar by going to View – Toolbars – Circular Reference, or the toolbar will show up automatically as soon as a circular reference has been created. Click on the reference to go to the circular reference.

**Fig 4.16 The Circular Reference Auditing Tool in the Ribbon in Excel 2003**

**[Excel for Mac 2011]** Using the “Circular:” reference on the status bar, select the cell containing the circular reference. On the Formulas tab, in the Formula Auditing group, click
the “Trace Precedents” button.

![Circular Reference Auditing Tool in the Ribbon in Excel for Mac 2011]

You can move between cells in a circular reference by double-clicking the tracer arrows. Keep reviewing and correcting the circular reference until the status bar no longer shows the word "Circular References."

**Circular References in Interest Calculations**

There are a few instances where you might wish to keep a circular reference within the model. A common reason for this is when you are calculating interest payments on a profit and loss statement.

![Circular reference in interest calculations]

_Fig 4.18 Circular reference in interest calculations_
When modelling financial statements, this circular reference can be avoided in a few different ways:

1. Hard-code one of the inputs, such as the interest amount. This is not recommended, as this will impede the flow of numbers throughout the model!

2. Use a macro to update the numbers, keep the links flowing, but paste the number to avoid the circular reference. This will work, but needs some good testing to make sure it works properly. See the section on “Macros in Financial Modelling” for points to consider before including macros in a financial model.

3. Calculate the interest based on the closing balance of the debt amount in the previous period. This is most appropriate for statements modelled on a monthly basis, as the variance between debt balances from year to year can mean that the interest amount calculated using this method is less accurate. Although not perfect, this is still my preferred method for getting around the problem of circular references in interest calculations.

4. Lastly, another commonly used method in financial statements is to allow the circular reference by enabling iterative calculations. The problem with this method is that it might allow other circular reference to be enabled which means that other parts of the model might not calculate properly.

Enabling Iterative Calculations

If you decide to enable iterative calculations, you must determine how many times the formula should recalculate. When you turn on iterative calculations without changing any of the defaults, Excel stops calculating after 100 iterations or after all values in the circular reference change by less than 0.001 between iterations, whichever comes first. However, you can control the maximum number of iterations and the amount of acceptable change.

How to Enable Iterative Calculations:

[Excel 2007/10] Go to Excel Options by either clicking the Microsoft Office Button in 2007, or the File Tab in 2010, and then select Options (or Excel Options), and then click the Formulas category.

In the Calculation options section, select the Enable iterative calculation tick box.

Enable Iterative calculation
Maximum Iterations: 100
Maximum Change: 0.001
Fig 4.19 Enabling iterative calculations in Excel 2007/10

**[Excel 2003]** Go to Tools – Options and then select the Calculation tab. Tick the Iterations box.

**[Excel for Mac 2011]** Go to “Preferences” (Or hit Command+,). In the Calculation options section, select the “Limit iteration” tick box.

Fig 4.20 Enabling iterative calculations in Excel for Mac 2011
CHAPTER 5 Using Excel in Financial Modelling

Excel Versions

There are currently three different versions of Excel that modellers are likely to be using; Excel 2003, 2007 or 2010. Ordinarily, it would not be such a big deal but the differences between Excel 2003 were so great, it does have quite a big impact on the financial modeller, and therefore we need to consider, in detail, the implications of the changes between the different versions.

History of Excel Versions

Excel 2003 was released in November 2003, and there was not a huge change between this version and its predecessors, Excel 2002 (XP) and Excel 2000. However, in January 2007, Office 2007 was released which was wildly different in terms of interface from anything users had even seen. The 2010 version was released in June 2010 which had a few changes, but nothing like the change between 2003 and 2007.

Current Excel Usage

When Excel 2007 was first released four years ago, it caused division amongst analysts and financial modellers, with many vowing to “stay as far away from it as possible”. The new ribbon layout has been described as “chaotic”, “not intuitive” and one poll respondent says it “does my head in”! From a Financial Modelling perspective of course, model structure, layout and best practice is far more important than the fancy new tools introduced by Excel 2007, and many modellers have shunned the new version for the past few years in favour of its more primitive predecessor. It can certainly take some getting used to, but there are many new features which make it worthwhile, such as new functions, increase file size and additional charting options and tools.

These new features were added to even further with the introduction of Excel 2010 last year. Not surprisingly, there has been a much greater take-up of this “new-fangled” version in the past few years; in fact the latest poll showed that 75% of users have migrated to 2007 or 2010, compared with a mere 29% as captured by a similar Plum Solutions survey conducted only two years prior.

Key Poll Findings:
The majority of Excel users are either on Excel 2010 or 2007. Even though Excel 2003 has not been available for purchase from Microsoft for quite some time, a Plum Solutions survey which was conducted in early 2011 showed that 25% of Excel users were still on 2003 – most of them because their company simply had not upgraded yet and so they had option. Excel 2003 will eventually be phased out over the next few years, but until then, no matter which version of Excel you are using, financial modellers still need to be aware of the differences between the versions. This is particularly in terms of functionality, where if a function, tool, or colour is created in 2010 or 2007, and then opened in 2003 it can lose functionality. At times, it can simply mean a slight difference in interface, but often, as with the example of new functions not available in 2003, it can cause errors to be displayed.

Many users accept Microsoft Excel’s dominance in the market place and are keen to keep their skills up to date, by voluntarily upgrading to the latest version. Less than 10% have remained on Excel 2003 through their own choice, an indication perhaps that most people are not resistant to change where their computer skills are concerned. Of those who dislike the newer versions, under 25’s and over 54’s did so in equal measure, reflecting perhaps that older and younger age groups are more resistant to change. A massive 33% of those who have embraced the new Excel 2010 are in the 30-36 year old age bracket.

The Ribbon which was new to Office 2007 is the source of most complaints, with comments such as “illogical layout”, “why fix [the menu system] when it ain’t broke?” and “hated ribbons initially, but getting used to it.” However, a number of comments did welcome the ribbon - and one even trashed the old menu system.

**Which Version am I Using?**

If you have got a toolbars at the top of your screen which expand to drop-down menus, you are using what some call “Classic Excel”; Excel 2003 or earlier.

**Classic Excel:**
Fig 5.6 Classic Excel

If your screen looks like this, with a Microsoft button in the top left hand corner, and a “Ribbon” instead of drop-down menus, you are using Excel 2007.

Excel 2007:

Fig 5.7a Excel 2007

If you have the Ribbon, but a File Tab instead of the Microsoft Button, you are using Excel 2010.

Fig 5.7 Excel 2010
The Upgrade

The upgrade from Excel 2003 to Excel 2007, and migrating to the “Ribbon” layout was probably one of the most significant changes for users yet. There are many new functions, most of which are an improvement to 2003 but they certainly did take some getting used to! One of the big issues many financial modellers face is with dealing with multiple versions of Excel.

Major changes between versions

The biggest change is the new “Ribbon” format replacing the toolbars. It is now more task-orientated and things are easier to find so you’ll probably find many features you never knew existed! It will take some time to get up to speed, however, and you will probably lose some productivity at first.

- The office assistant paper clip is gone for good!
- The colours and charting are much more professional looking in 2007

Themes and styles have much more functionality.
- Live Preview - when you move the mouse over the various formatting selections, your document takes on the formatting attributes of the current selection — just as a preview. So you can flit from one choice to another and try different formatting without actually applying it until you’re ready.
- Conditional formatting is no longer limited to three conditions, and easier to use.
- It is possible to add comments to named ranges.
- The formula bar is resizable so you can write even longer formulas (not always a good thing!)
- As with 2003, you can edit named ranges – but the tool for this is now called the Name Manager. Except for the new name, it’s pretty much the same as it was before,
although you can now delete multiple names at once and assign comments to each name.

- The size of each worksheet has expanded from 65,000 to over 1 million rows!
- It now remembers up to fifty recently used files instead of nine.
- There are more options in the right-click menu.
- Sort and Filter are now together, so you can sort from the filter drop-down list.
- Sorting limits have increased from three to 64 levels and you can even sort by colour! (This feature is activated when at least one cell contains a fill or font colour)
- It has a new file format which facilitates integration with external data sources, and also offers reduced file sizes and improved data recovery. In Excel 2007, the default format for an Excel workbook is the XML-based file format (.xlsx). If the file contains macros, the format will need to be .xlsm.
- “On demand” Ribbons which only appear when required for less used tools like pivot tables, charts etc. Although many new users hate the Ribbon, the “on demand” feature is much better than the old Excel 2003 which clutter up your screen, are sometimes not loaded when you open Excel and remain floating after you don’t need them anymore.
- Pivot tables and charting do not use wizards anymore, and are completely different.

**Completely New in 2007**

- Many previously custom functions are now standard, including: IFERROR, EOMONTH, SUMIFS, COUNTIFS, AVERAGEIFS, XNPV, XIRR
- Data bars in conditional formatting will show dynamic bar relative to the size of the value in the cell.
- Remove duplicates tool (very useful!)

**Hot Tip:** Press the ALT key, and the ribbon shortcut keys will appear as shown below:

![Fig 5.9 Shortcut keys Excel 2007/10](image)

If you really hate the Ribbons in Excel 2007 but are stuck with it, you can also purchase add-in software which installs the Classic menu for Excel 2007 or 2010, which looks like this:
Excel 2007/2010 Compatibility

Until everyone has upgraded to the new version, compatibility will remain the biggest challenge for Excel 2007 users, especially if others need to open the model in 2003. There are a number of new functions in 2007/10 which are not available in 2003. If you are the only person using 2007/10 and everyone in your office or your clients are still on 2003, you will have to make sure you do not use the new functions otherwise you will have compatibility issues. Note that all the great graphics and colours still look fine when you open the model in 2003, as long as you do not use a formula not supported in 2003.

A compatibility checker tells you if your workbook contains features that previous versions of Excel will not support. But be careful – you will need to remember to save a document in Excel 2003 format to maintain compatibility with other users until the new 2007 file format becomes the standard. While Microsoft has released a converter to read 2007 files in earlier versions, do not rely on your colleagues to have it installed.

If you use new functions not available in 2003, the compatibility checker will keep advising you to save a copy in 2003 – which gets rather annoying after a while.
Upgrading from Excel 2007 to Excel 2010

The introduction of Office 2007 caused such an uproar, that upgrade from Excel 2007 to Excel 2010 was completed very quickly. Most of the complaints were because the 2007 version had completely changed the menu navigation. While Microsoft attempted to simplify the navigation screens to market the product to a general public, the impact was felt across all professions. The Office 2010 environment was quickly scaled back and made more instinctual to the users that operate the system on a daily basis.

Major changes between versions

The first major change for Office 2010 is the performance of the software. Included in the package, Microsoft offers Excel 2010 in the 64 Bit format. This substantially uses more memory to maximise performance calculating complex equations. Microsoft introduces the High Performance Computing (HPC) which solves complex calculations using multiple computers. This allows Excel to offload complex calculations to other computers on the network. It allows the computer to calculate other parts in a workbook in parallel increasing performance of the software.

The biggest change for Office 2010 is moving back to the classic format and navigation in the menu screens. The most noticeable change is the MS. Office button is no longer present in the 2010 version.

One of the most powerful changes that were added to Excel 2010 is the ability to customise the Ribbon and Toolbars. In Excel 2003, the toolbar is very customisable. Users were accustomed to moving their menu bar around, and only showing the tools that they used frequently. This customisation ability was drastically reduced with the Excel 2007 Ribbon, where there is very little freedom to change the menu. Excel 2010 however allows you to create your own workspace environment. You can click on Excel Options as shown and customise the ribbon to add menus, tabs, or functions any way you like.

Fig 5.11 Customising the Ribbon in Excel 2010

The changes between Excel 2007 and 2010 are quite difficult to detect because the home tabs are virtually identical. However, there are some significant improvements on the other tabs inside Excel.
Inside the Insert tab, the first noticeable different between 2007 and 2010 is the Screenshot button. This function allows you to capture a screen anywhere on your desktop and import into the office environment.

![Screenshot button in Excel 2010](image)

*Fig 5.12 The screenshot button in Excel 2010*

The next noticeable improvement is the new feature called Sparklines, which are great for inserting miniature charts into reports and dashboards. Sparklines are a small mini-chart that can be plotted to enhance the visualisation of the tabular data. This function allows you to insert a line, bar, or win/loss into a single cell. This will allow you to add enhanced visual data analysis functions into pivot tables and dashboards.

![Sparklines in Excel 2010](image)

*Fig 5.13 Sparklines in Excel 2010*

Another function that has been added to the Excel 2010 is the slicer. Slicer is used to filter data inside a pivot table. This function makes it quick and easy to sort and display the data inside the pivot table. For example: If you create a pivot table with student’s grades, subjects and total percentages. The slicer can sort out each individual students or subjects in the table.
The Excel 2010 has improved the tab under Pivot table by adding a **show values as** drop down menu that allows you to quickly change how to show the information in the table. You can also change the values inside the table by using the dropdown menu: **summarise values by**.

Excel 2007 had made substantial improvements to the **conditional formatting** function inside Excel. It is much easier to fill cells based on a value or rules inside it, and the visual displays such as data bars, colour scales and icon sets are very easy to use, and visually appealing. Excel 2010 has added a little bit of additional functionality such as data bars with solid fill, instead of a gradient. The data visualisation gurus were not very happy about a bar chart without solid comparison lines!
Improvements to the charts between versions

Microsoft made some significant changes to the chart features inside Excel 2010. This improved function of Excel to show graphs and charts.

- Excel 2010 has enhanced the chart capability by supporting **Macro Recoding** to record format changes to the charts. The Excel 2007 version did not produce Macro code when creating format changes with recording.
• The Excel 2007 version only allowed up to 32,000 points in a data series on the 2d charts. Excel 2010 fixes this by allowing as many points as the memory in the computer will allow.

• Chart formatting has been improved by double clicking any chart element and a format bar will appear or you can right click on the chart element to get a mini-format selector.

• Excel 2010 has brought back the chart pattern fill which was not available on the 2007 version.

New calculation and function features

• The conditional sum wizard used in Excel 2007 is now replaced with a Function Wizard. This wizard will include the SUMIF and SUMIFS functions.

• The lookup wizard used in Excel 2007 is now replaced with a Function Wizard. This wizard will include the SUMIF and SUMIFS functions.

• Some of the statistical functions have been improved for better accuracy, such as Poisson distribution (POISSON) and Normal Distribution (NORMDIST).

• Excel 2010 is able to handle an increased range of input variables into the functions. This allows Excel to return a broad range of results from the improved math functions.

Additional changes made to Excel 2010

The Microsoft Button encountered quite a bit of resistance from users, and in the 2010 version, Microsoft returned to using a File button. In addition to returning to the File button, under that ribbon the developers added extra tools into “Backstage”. In the backstage view, you are able to perform several additional functions including:

• A Recent tab that allows you to view all the docs you have saved and worked on. This keep track where each workbook is located and allows a quick access to it.

• The New tab allows you to create a new workbook with additional templates from Office.

• The Info tab allows an easier to navigate tasks including information on the file, protecting the workbook, assigns security permissions, and check for compatibility.

• The Print tab will allow easy navigation through the print options including fitting to page, portrait or landscape, paper size, and margins. In the print tab you are able to preview the workbook before sending to the print or file destination.
The **Save and Send Tab** allows you to send the document to a PDF conversion, or to the web or a SharePoint center for file sharing online. You can also save as a different file version or change the file type.

*Fig 5.19 Backstage view in Excel 2010*

**Filter Options**

When you run an auto filter on Excel 2010, you have a nice feature that allows you to **search** inside the auto filter. This can be a very useful function when sorting large sets of data.

When you have data on a table, the filter and sort will stay in the column heading even when you scroll down. This is very convenient because you can always tell what the column content is without having to scroll to the top.
Should you Upgrade?

According to the Plum Solutions survey, in Jan 2011, 25% of Excel users were still using Excel 2003 or earlier! If you plan to look for work soon and you wonder which version of Excel future employers will want to see on your resume, Excel 2010 is a must, but having familiarity with both versions will certainly improve your marketability.

If the new features such as new functions and increased capacity are not enough to tempt you, there are many reasons to upgrade – by remaining on “Classic Excel” you are really just delaying the inevitable. 2003 is out of date: The old Office platform was built on what are now antiquated technologies – the leap to Office 2007 represents a clean break in platform design.

The main reason many companies upgrade is because they are working with clients or external parties using the later versions, and it’s just easier if you are using the same version. They often also upgrade to obtain some of the new features, in particularly the improved file format which has much greater capacity.
Handy Excel Shortcuts

Working in Excel – particularly for the purpose of financial modelling – can be very time-consuming and increasing your speed and accuracy when modelling will increase productivity significantly. Excel users just starting out with using Excel, would be very comfortable with using the mouse for navigation and editing purposes. However, with growing familiarity, the mouse is not the fastest or most efficient way of modelling. Excel offers a lot of shortcuts to tackle this problem, and as you become a faster modeller, you will find these shortcuts very handy.

There are many advantages of using Excel shortcuts:

a) **Ease of use**: While intuitively it may seem like mouse is the more comfortable option, but it a lot more comfortable for your wrist, arm and shoulders to punch the keys on the keyboard than trying to align the cursor on the screen.
b) **Easier on the Body**: Given the natural placement of keyboard and mouse on any desktop, using the keyboard more often is less stressful on the body. Using the mouse constantly can cause stress on the wrists and shoulders. With the mouse most users typically end up using just the wrists and index finger which can cause fatigue and long term problems.
c) **Speed of execution**: Keystrokes are much faster than the mouse. To put this to test, try creating a new worksheet in Excel using the mouse by clicking on File (or Microsoft Button in Excel 2007), then New – Blank Workbook. Alternatively try the shortcut CTRL-N There is a distinct improvement in speed with keystrokes over the mouse click.
d) **Standard Shortcuts**: Within the Windows environment, you will find that similar functions have the same shortcuts so it is not like you are developing expertise in just Excel. The shortcuts in Excel are generally common across all other office suites and even other applications like browsers, notepad, paintbrush, etc.
e) **No Other Choice**: There are some functions where unless you use the shortcuts you will not get the desired results – without manual edits. For example, when creating an array formula, the shortcut Control-Shift-Enter is the only way to get your array formula to insert correctly.

Below is a small selection of some of the most common and particularly useful shortcuts that are invaluable in using Excel for the purpose of financial modelling.

**Editing**

CTRL + S  Save Workbook
CTRL + C Copy
CTRL + V Paste
CTRL + X Cut
CTRL + Z Undo
CTRL + Y Redo
CTRL + A Select All
CTRL + R* Copies the far left cell across the range
CTRL + D* Copies the top cell down the range
CTRL + B Bold
ALT + TAB Switch Program
ALT + F4 Close Program
CTRL + N New workbook
SHIFT + F11 New worksheet
ALT + W* Close worksheet
ALT + E + L* Delete a sheet
CTRL + TAB Switch workbooks
CTRL + page up Select next worksheet to the right
CTRL + page down Select next worksheet to the left

Navigating

CTRL + 9 Hide Row
SHIFT + CTRL + 9 Unhide Row
SHIFT + Spacebar Highlight row
CTRL + Spacebar Highlight column
CTRL + Minus sign Delete selected cells
Arrow keys Move to new cells
CTRL + Pg
Up/Down Switch worksheets
CTRL + Arrow keys Go to end of continuous range and select a cell
SHIFT + Arrow keys Select range
SHIFT + CTRL + Arrow Select continuous range
Home Move to beginning of line
CTRL + Home Move to cell “A1”
SHIFT + ENTER Move to cell above
TAB Move to cell to the right
SHIFT + TAB Move to cell to the left
ALT + Display a drop-down list

Formatting

CTRL + 1 Format Box
ALT + H + 0*  Increase decimal
ALT + H + 9*  Decrease decimal
SHIFT + CTRL + ~  General format
SHIFT + CTRL + !  Number format
SHIFT + CTRL + #  Date format
SHIFT + CTRL + $  Currency format
SHIFT + CTRL + %  Percentage format

**In Formulas**

F2  Edit formula, showing precedent cells
    Edit formula, showing precedent cells (same as
    F2)
CTRL + ’  Start new line in same cell
ALT + ENTER  Highlight within cells
SHIFT + Arrow  Change absolute referencing ("$")
F4  Cancel a cell entry
ESC  Start a formula
= (equals sign)  Sum selected cells
CTRL + ‘  Copy formula from above cell
CTRL + ~  Show formulas or values
F9  Recalculate all workbooks
SHIFT + CTRL + Enter  Enter array formula

**Auditing**

ALT + M + P*  Trace immediate precedents
ALT + M + D*  Trace immediate dependents
ALT + M + A + A*  Remove tracing arrows
CTRL + [  Highlight precedent cells
CTRL + ]  Highlight dependent cells
F5 + Enter  Go back to original cell
SHIFT + CTRL + {  Trace all precedents (indirect)
SHIFT + CTRL + }  Trace all dependents (indirect)

* Note that these are unavailable as preset keyboard shortcuts for Mac for Excel 2011. Create shortcuts for these functions by selecting Tools > Customize Keyboard.

**Mac Excel Shortcuts**
Almost all of the keyboard shortcuts that are available for Windows are available on the Mac as well. If you are new to the Mac, remember that in most cases, Command replaces Control, and Option replaces Alt.

A few of the keyboard shortcuts found in Excel for Windows are unavailable for Mac (noted in the list above by an *). These, however, may be created manually by selecting Tools > Customize Keyboard.

### Editing

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND + S</td>
<td>Save Workbook</td>
</tr>
<tr>
<td>COMMAND + C</td>
<td>Copy</td>
</tr>
<tr>
<td>COMMAND + V</td>
<td>Paste</td>
</tr>
<tr>
<td>COMMAND + X</td>
<td>Cut</td>
</tr>
<tr>
<td>COMMAND + Z</td>
<td>Undo</td>
</tr>
<tr>
<td>COMMAND + Y</td>
<td>Redo</td>
</tr>
<tr>
<td>COMMAND + A</td>
<td>Select All</td>
</tr>
<tr>
<td>COMMAND + B</td>
<td>Bold</td>
</tr>
<tr>
<td>COMMAND + TAB</td>
<td>Switch Program</td>
</tr>
<tr>
<td>COMMAND + Q</td>
<td>Close Program</td>
</tr>
<tr>
<td>COMMAND + N</td>
<td>New workbook</td>
</tr>
<tr>
<td>SHIFT + F11</td>
<td>New worksheet</td>
</tr>
<tr>
<td>COMMAND + TAB</td>
<td>Switch workbooks</td>
</tr>
<tr>
<td>COMMAND + page up</td>
<td>Select next worksheet to the right</td>
</tr>
<tr>
<td>COMMAND + page down</td>
<td>Select next worksheet to the left</td>
</tr>
</tbody>
</table>

### Navigating

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL + 9</td>
<td>Hide Row</td>
</tr>
<tr>
<td>SHIFT + CONTROL + 9</td>
<td>Unhide Row</td>
</tr>
<tr>
<td>SHIFT + Spacebar</td>
<td>Highlight row</td>
</tr>
<tr>
<td>CONTROL + Spacebar</td>
<td>Highlight column</td>
</tr>
<tr>
<td>CONTROL + Minus sign</td>
<td>Delete selected cells</td>
</tr>
<tr>
<td>Arrow keys</td>
<td>Move to new cells</td>
</tr>
<tr>
<td>COMMAND + Pg</td>
<td>Switch worksheets</td>
</tr>
<tr>
<td>Up/Down</td>
<td>Switch worksheets</td>
</tr>
<tr>
<td>COMMAND + Arrow keys</td>
<td>Go to end of continuous range and select a cell</td>
</tr>
<tr>
<td>SHIFT + Arrow keys</td>
<td>Select range</td>
</tr>
<tr>
<td>SHIFT + COMMAND + Arrow</td>
<td>Select continuous range</td>
</tr>
</tbody>
</table>
Home: Move to beginning of line
COMMAND + Home: Move to cell “A1”
SHIFT + ENTER: Move to cell above
TAB: Move to cell to the right
SHIFT + TAB: Move to cell to the left
CONTROL + : Display a drop-down list

**Formatting**

COMMAND + 1: Format Box
SHIFT + CONTROL + ~: General format
SHIFT + CONTROL + !: Number format
SHIFT + CONTROL + #: Date format
SHIFT + CONTROL + $: Currency format
SHIFT + CONTROL + %: Percentage format

**In Formulas**

CONTROL + U: Edit formula, showing precedent cells
CONTROL + OPTION + RETURN: Start new line in same cell
SHIFT + Arrow: Highlight within cells
COMMAND + T: Change absolute referencing ("$")
ESC: Cancel a cell entry
= (equals sign): Start a formula
CONTROL + SHIFT + T: Sum selected cells
COMMAND + ’: Copy formula from above cell
CONTROL + `: Show formulas or values
F9: Recalculate all workbooks (Note that the user will have to inactivate Expose in order for this shortcut to work)
SHIFT + CONTROL + RETURN: Enter array formula

**Auditing**

COMMAND + [: Highlight precedent cells
COMMAND + ]: Highlight dependent cells
SHIFT + COMMAND + {: Trace all precedents (indirect)
SHIFT + COMMAND + }: Trace all dependents (indirect)
CHAPTER 6 Functions for Financial Modelling

Lookup Formulas

Lookup functions are very commonly used in financial modelling and analysis – sometimes a little too commonly used in my opinion! They are useful to know, but sometimes another function would create a more robust solution.

VLOOKUP (Vertical Lookup)

VLOOKUP stands for “Vertical Lookup”. It can be used anytime that you have a list of data with the key field in the left-most column, and is by far the most commonly used form of LOOKUP formula.

Let’s say you have a shopping price list like this:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fruit Price List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Fruit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Apples</td>
<td>$0.80</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Oranges</td>
<td>$0.60</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Pears</td>
<td>$1.00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bananas</td>
<td>$2.50</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4 Oranges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 6.12 Example shopping price list

In cell B8, we’d like to find out how much it will cost to buy 4 oranges, using a VLOOKUP.

How to create a VLOOKUP:

1. Find and select the VLOOKUP function. Using the wizard is probably going to be the easiest for VLOOKUPs as there are four parameters it needs.
2. The first parameter is the criteria we are testing; i.e. the word “Oranges”, so enter “B8” in the first field.
3. The next parameter is the table which contains the data you want to reference. OK, this is where it gets tricky. The criteria you are looking for must always be in the far left hand column of the data table you are referencing in the table array. So, in this case, the data we are referencing will be the range B3:C6. This is what your wizard should look like so far:
4. Now, if you were going to copy this formula down the page, you would have to use absolute referencing for the Table Array, or use a named range. Let’s create a named range here.

5. OK, cancel out of the wizard and create a named range for the whole table. Highlight the range B3:C6 and type over the name box in the top left hand corner of your screen with the word “FruitList”, and press enter. We can now use this in the VLOOKUP. Now that we have used a named range, we won’t have to worry about absolute referencing in our VLOOKUP formula.

6. Go back into the VLOOKUP wizard again, re-reference the lookup value to B8, and either select the table array in the second field box, or simply type in “FruitList”. Make sure you have spelt it the same way as your named range! Your wizard so far should look like this:
7. The third field called Col_index_num tells Excel in which column the value is found. We need to tell the function which column in the table array we want it to return. In this case, we want it to tell us the price. So, counting from the far left hand side of the table array, we want it to return the second column. So we enter a number 2 in the col_index_num field.

8. The optional 4th parameter tells Excel if a “close” match is OK. If you want an exact match, enter zero, otherwise leave it blank. You may also enter TRUE or FALSE (but typing a zero or leaving it blank is quicker!) Click OK to complete the formula.

Note: There are very few instances where you want a close match. (see the section on Tiering Tables for an example of using a close match in VLOOKUPS) In most VLOOKUPs, you want an exact match, so enter a zero in the last field.

9. Your formula should now be =VLOOKUP(B8,FruitList,2,0)

10. Now, try changing the word “oranges” to “apples”. The price should change.

11. Lastly, we need to multiply the price by the number of items we are purchasing.
   Change your formula to: =VLOOKUP(B8,FruitList,2,0)*A8

#N/A Errors in VLOOKUPs

While this example worked out perfectly, when viewers use VLOOKUP, it usually means that they are matching up lists that came from different sources. When lists come from different sources, there can always be subtle differences that make the lists hard to match. This means you will get a #N/A error if the VLOOKUP function cannot find a match. Here are two examples of what can go wrong and how to correct them.
Example 1: One list has dashes, for example, and the other list does not. The first time that you try the VLOOKUP, you will get #N/A errors. To remove the dashes with a formula, use a find and replace. Replace “-“ with nothing (i.e. a blank). Your data will now match and the formula works. You can wrap that function in the VLOOKUP to get the description.

Example 2: One list has a trailing blank space after the entry. This one is subtle, but very common. Use =TRIM() to remove excess spaces. When you initially enter the formula, you find all of the answers are N/A errors. You know for sure that the values are in the list and everything looks OK with the formula.

One standard thing to check is to move to the cell with the lookup value. Press F2 to put the cell in Edit mode. Once in edit mode, you can see that the cursor is located one space away from the final letter. This indicates that there is a trailing space in the entry.

To solve the problem, use the TRIM function. =TRIM(D4) will remove leading spaces, trailing spaces, and will replace any internal double spaces with a single space. In this case, TRIM works perfectly to remove the trailing space. =VLOOKUP(TRIM(D4),$I$3:$J$351,2,FALSE) is the formula.

Breaking a VLOOKUP

If you have created a VLOOKUP in a model such as the one above, this should work well.... Until someone enters or deletes a column in your source data! With a formula such as =VLOOKUP(B8,FruitList,2,0)*A8, it specifically asks for the second column, so it will not work if someone inserts a column within the FruitList range. This is because your required column becomes the third column, but the VLOOKUP is asking for the second.

VLOOKUPS and HLOOKUPS are not a very robust formulas – you can see how easy they are to break! For this reason, try some of the alternatives below to make your LOOKUP more robust, use another formula, or else protect your model so that users cannot insert or delete rows or columns.

Improving VLOOKUPS with MATCH

The best way around this is to make the “2” in your VLOOKUP a formula instead of a hard-coded number. One way of doing this is to nest the VLOOKUP with a MATCH function. We will create the MATCH formula in a separate cell, and then replace the “4” with the MATCH formula.

1. Using the VLOOKUP example above, test the problem by inserting or deleting a column in our sheet. You will see that the value in your VLOOKUP is now incorrect.
2. Now go to another spare cell, say cell B17, and create a MATCH formula that will find the word “Price” in the range B2:C2 above and tell us its position in the range.
3. Your formula wizard should look like this:
4. By putting a “0” in the Match_type range, you are asking it to return you an exact match instead of a close match.
5. When you hit OK, your formula should be =MATCH(C2,$B$2:$C$2,0) and the result should be 2.
6. Go into the MATCH formula bar, highlight the entire formula (except for the = sign), and press Control-C (Command-C in Excel for Mac 2011). Hit enter. Now, go back to your original VLOOKUP, and replace the “2” in your VLOOKUP formula with this entire MATCH formula by highlighting the “2” within the formula then pressing Control-V (Command-V in Excel for Mac 2011). Hit enter.
7. Your formula should look like this:
   \[=VLOOKUP(B8,FruitList,MATCH(C2,$B$2:$C$2,0),0)*A8\]
8. Test that this works, by inserting or deleting a column.
9. Another way of doing this is to use the =COLUMN or =ROW function instead of MATCH. This will automatically return the range’s column or row reference. Using this, the formula instead would be:  
   \[=VLOOKUP(B8,FruitList,COLUMN(C3)-1,0)*A8\] (The -1 is required because the range is starting from column B.)

**HLOOKUP (Horizontal Lookup)**

HLOOKUP works in exactly the same way as the VLOOKUP, except that the data is arranged horizontally instead of vertically; for example:
The HLOOKUP is subject to exactly the same issues as the VLOOKUP and works in exactly the same way except for the orientation.

**LOOKUP Function**

The LOOKUP function is much simpler than either the VLOOKUP or the HLOOKUP, and it has the added advantage of being able to have the results column or row **either** to the left or the right of the criteria column or row – a huge advantage. However, the data MUST be sorted in alphabetical order or it won’t work. This does limit it usage significantly, and it is for this reason that it is far less popular than the VLOOKUP or HLOOKUP formulas.

It’s worth knowing how it works though, so here’s a brief overview:

1. Select LOOKUP from the function wizard
2. Choose the first option:

![Fig 5.19 LOOKUP function option](image)

3. Similar to an HLOOKUP or VLOOKUP, the criteria being tested goes in the first field.

4. The column (or row) containing the criteria goes in the second field, and then the column (or row) containing the result goes in the last field.

![Fig 5.20 LOOKUP function dialog box](image)

5. Copy the LOOKUP function down, and multiply the price by the total amount to complete the pricing table as shown:
Note that this works exactly the same whether the data is shown vertically or horizontally. Importantly, unlike a VLOOKUP or HLOOKUP, it does not matter whether the result data is to the left or right, above or below the lookup data.

However, you would only use a LOOKUP for numerical values where the numbers are sorted, and you actually want a close match, not an exact match. You would not use a LOOKUP function for text, as shown in the last fruit shopping list VLOOKUP example, because it will only return a close match and this is dangerous. This makes the LOOKUP function almost useless in most cases where a lookup is required. The only time a modeller would use a LOOKUP is when they specifically want a close match.

So, which LOOKUP Function should I use?

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| VLOOKUP  | • Data does not need to be sorted alphabetically, as long as you use the 0 or FALSE in the last field | • The column number is hard coded, so if a column is inserted, it can mess up the formula (you can replace this with a MATCH to fix this)  
• The column containing the results MUST be to the right hand side of the column containing the lookup criteria. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLOOKUP</td>
<td>• As above</td>
</tr>
</tbody>
</table>
| LOOKUP  | • Column/row number not hard-coded  
          • Does not matter where the columns or rows are in relation to each other | • Data MUST be sorted or it will give the wrong result. |
Using INDEX and MATCH to create a More Robust Formula

As described in the section on "Linking to External Files", robust formulas such as a nested formula using a combination of an INDEX and a MATCH, together with named ranges will make the link a lot less likely to break and cause problems in models.

Let’s say that you are referencing a table in another file. Your co-worker keeps inserting and deleting rows and columns. The VLOOKUP solution as described above may work, but with large tables, an INDEX and MATCH formula combination will be more efficient.

Below is a table which contains the source data, and we need to pick up the Staff Amenities costs for Geelong.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Campbelltown</td>
<td>Melbourne</td>
<td>Sydney</td>
<td>Parramatta</td>
<td>Geelong</td>
<td>Brisbane</td>
<td>Toowoomba</td>
<td>Perth</td>
</tr>
<tr>
<td>2</td>
<td>Staff Costs</td>
<td>$68,432</td>
<td>$682,118</td>
<td>$311,589</td>
<td>$13,907</td>
<td>$460,430</td>
<td>$552,516</td>
<td>$414,387</td>
</tr>
<tr>
<td>3</td>
<td>Superannuation</td>
<td>$51,159</td>
<td>$61,391</td>
<td>$46,043</td>
<td>$5,252</td>
<td>$41,439</td>
<td>$49,726</td>
<td>$37,285</td>
</tr>
<tr>
<td>4</td>
<td>Workers Comp</td>
<td>$5,684</td>
<td>$6,821</td>
<td>$5,136</td>
<td>$6,139</td>
<td>$4,604</td>
<td>$5,525</td>
<td>$4,144</td>
</tr>
<tr>
<td>5</td>
<td>Staff Amenities</td>
<td>$28,422</td>
<td>$23,106</td>
<td>$25,579</td>
<td>$30,695</td>
<td>$23,023</td>
<td>$27,626</td>
<td>$20,719</td>
</tr>
<tr>
<td>6</td>
<td>Consumables</td>
<td>$34,106</td>
<td>$40,927</td>
<td>$30,695</td>
<td>$36,834</td>
<td>$27,628</td>
<td>$33,151</td>
<td>$24,863</td>
</tr>
<tr>
<td>7</td>
<td>Recruitment</td>
<td>$17,053</td>
<td>$20,464</td>
<td>$15,348</td>
<td>$18,417</td>
<td>$13,813</td>
<td>$16,575</td>
<td>$12,432</td>
</tr>
<tr>
<td>8</td>
<td>Travel</td>
<td>$11,369</td>
<td>$13,642</td>
<td>$10,232</td>
<td>$12,278</td>
<td>$9,209</td>
<td>$11,050</td>
<td>$8,288</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>$716,224</td>
<td>$859,499</td>
<td>$644,602</td>
<td>$773,522</td>
<td>$580,142</td>
<td>$696,170</td>
<td>$522,128</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 6.22 Sample Data

You could simply reference the cell F5, using the formula =F5, but this will not help you if your file is closed, and someone else changes the above table. Let’s say your co-worker inserts a column while your model is closed. When you open your file again, it will still reference F5, but Geelong has moved to column G so the formula in your model will be wrong!

By using a combination of an INDEX and MATCH formula, we will be able to specify the exact co-ordinates of the required value, even if it’s position in the table changes.

1. In cell B12, insert the word “Geelong”, and in cell A13, insert the word “Staff Amenities”. These cells will become input variables that we will link our formula to.

2. Now let’s find the co-ordinates of the required data. Start by creating a MATCH formula in a spare cell that will find the word “Geelong” and tell us its position in the range of city names.

3. Your formula should be =MATCH(A13,A1:A11,0), and the result should be “6”. Note that you could type the word “Geelong” directly into the formula, but linking it is much better financial modelling best practice.

4. Now do the same thing for the expense types. In another spare cell, create a MATCH formula that will find the word “Staff Amenities” and tell us its position in the range of expense types.

5. Your formula should be =MATCH(A13,A1:A8,0) and the result should be “5”.

© Plum Solutions Pty Ltd 2012  Page 53  www.plumsolutions.com.au
6. Now create the INDEX formula that will return the value that you specify. Note that when you use the wizard, it will ask you which argument list you wish to use. Select the first one.

![INDEX function options](image1)

**Fig 6.23 INDEX function options**

7. Once you are in the formula wizard, ask it to return the first row and the first column, just to test it.

![INDEX function wizard](image2)

**Fig 6.24 INDEX function wizard**

8. Your formula should look like this: \( \text{INDEX}(A1:S8,5,6) \) and the result should be “$23,021”.

9. This formula will work fine, but it cannot handle someone inserting or deleting rows. In order to make it completely robust, we need to replace the “5” and “6” with the results of the MATCH formulas. We are going to turn this into a nested formula, just like we did in the “Nested IF Function” section.

10. Go back into the first MATCH formula bar, highlight the whole thing (except the “=” sign) and press Control C (Command V in Excel for Mac 2011).

11. Go into the INDEX formula and replace the column reference with this formula. Highlight the “6” and press Control V (Command V in Excel for Mac 2011).
12. Now do the same for the row reference. Go into the second MATCH formula bar, highlight the whole thing (except the “=” sign) and press Control C (Command C in Excel for Mac 2011).

13. Go into the INDEX formula and replace the row reference with this formula. Highlight the “5” and press Control V.

14. Don’t forget to add in some absolute referencing. Your formula should look like this:

   =INDEX($A$1:$I$8,MATCH(A13,$A$1:$A$8,0),MATCH(B12, $A$1:$I$1,0))

15. Test the robustness of your formula by inserting or deleting rows or columns. You can now cut this formula to another file, and it will work in the same way.

16. Remember that if you want to move this formula around, use the cut and paste function, not copy and paste. Cut and paste will move all formulas with it. If you cut this formula to another file, you should follow best practice by using named ranges in the formula, particularly where the INDEX function is referencing the data. This way if the data is added to, the named range will expand accordingly.

OFFSET Function

The OFFSET function, in my opinion, is less useful than the other functions we have covered so far as it can be tricky to build and is complex to audit. However, many financial modellers are fond of including it in their models and therefore you will be likely to come across it and need to know how to decipher it!

The OFFSET is used to return the address of a cell or a range of cell through the use of a reference cell, and is generally used in order to stagger series of values by a variable amount. For example, if you want to delay a project by a certain number of months, but want those months to be variable, the OFFSET function will move the value by the number of months that are specified in the model.

To give a very simple example, let’s say you wanted to pick up the value in cell B3, i.e. 54, which is located one column across and two columns down from cell A1.
So, using the OFFSET wizard the reference point given is cell A1, and the offset rows are 1 and offset columns are 2.

Don’t worry about the height and width for now (this is for when you want the result to be a range, rather than a single cell, such as a named range – see the section on “Dynamic Range Names” for an example) Your formula will be: `=OFFSET(A1,2,1)`

This is picking up the cell two rows to the right, and two columns below the reference cell. If you wanted to pick up a cell above or to the left of the cell, you’d used a negative offset number for the rows and columns (i.e. -2 instead of 2).
Using an OFFSET to Model Cashflow

So now that you know how to do an OFFSET function, let’s see how it’s used in a more practical context.

Let’s say that you have the following sales being made in each month, but the terms of payment are 1 month – hence the cash is received a month after the sale is made.

Let’s create a “Cash Receipts” line in row 5 using the OFFSET formula which will delay the cash receipts by the number in cell B1.

1. Start in cell C5, and create an OFFSET function which uses cell C4 as a reference point, and offsets the number of columns by the number of months in cell B1.
2. Leave rows blank, as you do not wish to move rows
3. Make the offset number of columns a negative number, because you wish to return the cell to the left, not the right
4. Your wizard should like this below, and the formula in cell C5 should be =OFFSET(C4,,,-$B$1)

Fig 6.28 Calculating a dynamic cashflow using the OFFSET function
5. Copy this formula across the row

**Nesting OFFSET and COLUMN Formulas**

This works when the terms are one month, however, if you change the number of months in cell B1 to 2, it will pick up the word “Sales” in cell A4! Similarly, if you change the cell B1 to 3, it returns a #REF! error value. We need to create a nested formula with a COLUMN function to make this model more robust.

A COLUMN function will return the column reference of the formula. E.g. COLUMN(B3) returns the value 2.

Similarly, a ROW function will return the row reference of the formula. E.g. ROW(B3) returns the value 3.

We need to add a formula in front of our OFFSET function to say that if the number of months in cell B1 is greater than the column position, it should return a zero.

1. In cell C5, replace the formula with =IF($B$1>=COLUMN(C3)-1,0,OFFSET(C4,,ABS(-$B$1)))
2. Don’t forget the absolute referencing, and copy it all the way across, including cell B5
3. Test to see if the formula is working correctly by changing the cell B1 to different values
4. If you are expecting no delay in cash receipts, you can enter a zero in cell B1, and the formula will still work

Your model should look like this:

<table>
<thead>
<tr>
<th></th>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terms: 2 month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td>Apr</td>
<td>May</td>
<td>Jun</td>
<td>Jul</td>
<td>Aug</td>
<td>Sep</td>
</tr>
<tr>
<td>4</td>
<td>Sales</td>
<td>$437</td>
<td>$966</td>
<td>$935</td>
<td>$906</td>
<td>$413</td>
<td>$276</td>
<td>$563</td>
<td>$358</td>
</tr>
<tr>
<td>5</td>
<td>Cash Receipts</td>
<td>$0</td>
<td>$0</td>
<td>$437</td>
<td>$966</td>
<td>$935</td>
<td>$906</td>
<td>$413</td>
<td>$275</td>
</tr>
</tbody>
</table>

*Fig 6.29 Completed dynamic cashflow using a nested OFFSET formula*
Alternatively, you may also choose to suppress the errors and text using an IF and ISNUMBER nested formula. In this case, it will return a zero if the result of the OFFSET is either an error, or text. Only a numerical value will be displayed in the output. The completed formula in cell B5 should look like this:

```
=IF(ISNUMBER(OFFSET(B4,-$B$1)),OFFSET(B4,-$B$1),0)
```

**Hot Tip:** Note that cell C1 contains the word “month” instead of “months”. If you insert the formula `=IF(B1=1,"month","months")` into cell C1, the syntax will always be correct no matter what value you insert in cell B1.

Note that it is not normally a good idea to use entire column or row references in formulas because they use a lot more memory, however by using them in this case, there is no limitation on how large the range can grow. If memory is an issue, you may prefer to limit the range to, say, 1,000 rows and columns or as many as you think your users may need. In this case you could put some colour markings so that the user knows when they have exceeded the range limits.

**CHOOSE Function**

The **CHOOSE** function returns a value from a list of values based on a given position.

For example, let’s say you want to pick up the second value in the following list of data:

```
Fig 6.31 List of days of the week
```

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunday</td>
</tr>
<tr>
<td>2</td>
<td>Monday</td>
</tr>
<tr>
<td>3</td>
<td>Tuesday</td>
</tr>
<tr>
<td>4</td>
<td>Wednesday</td>
</tr>
<tr>
<td>5</td>
<td>Thursday</td>
</tr>
<tr>
<td>6</td>
<td>Friday</td>
</tr>
<tr>
<td>7</td>
<td>Saturday</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

© Plum Solutions Pty Ltd 2012 Page 59  www.plumsolutions.com.au
1. Type out the list of days, starting with Sunday.

2. Select cell A9. Go into the function wizard and select CHOOSE

3. Index_num is the value that you want the formula to return. Enter “2” in this field

4. Enter the values from which you want the formula to choose. Note that you can hard-code these, or use a reference the way that we have done here.

5. Hit OK, and your formula should be: =CHOOSE(2,A1,A2,A3,A4,A5,A6,A7)

Using a Nested CHOOSE Formula with Dates

Let’s say you’d like to be able to enter in any date, and have a formula that tells you, in words, the day of the week it is.

1. Go to cell B9, and enter your date. This could be your birth date, or you could create a formula with today’s date by typing =TODAY(). Every time you open the model, it will show the current date. Note that =NOW() will show the date and the time.

2. The formula =WEEKDAY(B9) will return the day of the week as a number, with Sunday being the first day of the week. Therefore, Sunday = 1, Monday = 2 etc.

3. Go back to your previous formula in cell A9, and turn it into a nested formula by replacing the hard-coded 2 with the WEEKDAY formula.

4. Your formula should be: =CHOOSE(WEEKDAY(B9),A1,A2,A3,A4,A5,A6,A7)
Financial Functions

Financial functions are designed to save Excel users time when calculating long and complicated formula such as interest repayments, depreciation or NPVs. They can be created manually, and if you have studied business or finance at University you will have had to create these calculations the long way, or using a financial calculator, but the pre-defined function in Excel saves time and makes it much easier to calculate commonly used financial calculations.

Project Evaluation Functions

There are three financial functions which are commonly used to assess a business case – or any series of cash flows; NPV, IRR and Payback period. There are pre-defined functions in Excel for calculating NPV and IRR, but there is not one for payback period. For methods on how to calculate it, see the section on “Payback period”.

NPV (Net Present Value)

NPV is the value of the expected future cash flows from an investment, expressed in today’s dollars. The investor specifies a target rate of return (the cost of capital) for investing capital; it is an "opportunity cost" concept. The investor has the choice of investing in this project, or putting their funds elsewhere, so they determine the “hurdle rate” or the amount they want to get back from the project.

The general rule (and the one you would have studied in your University finance textbooks!) for considering the investment is if the NPV is greater than zero, the investment should be accepted; if the NPV is negative, it should be rejected. A positive NPV means the investor can expect to earn a rate of return greater than the required return rate for such an investment. However, from a financial modelling perspective, the decision making process is much more complicated than a “deal or no deal” situation. It really depends on the scenario and sensitivity analysis from the financial model to test whether or not the project should go ahead. Large companies often have policies regarding the standard cost of capital or how long the payback period can be before it is rejected.

What Cost of Capital should we use?

How much time is spent calculating the cost of capital really depends on how detailed your modelling is. Many models will simply use a nominated amount and document this as an assumption. This nominated cost of capital could be anything between say 6% and 15% (although sometimes higher!) and it can fluctuate depending on the perceived risk of the project. We may decide to use very high required rate of return for a risky project to compensate for the risk taken.
However, instead of simply nominating a cost of capital amount to use as we will do in the example below, you may decide to calculate the weighted average cost of capital, called the “WACC” for short. This calculation takes into account the mixture and rates of debt and equity in the company, and is therefore a much more accurate way of evaluating what the expected rate of return for a project. For more detail on how to calculate the WACC, see the section on “Weighted Average Cost of Capital”.

How to calculate the NPV:

With a series of cashflows as shown below, the NPV is calculated as follows:

1. Select NPV function. At the Rate prompt, enter the cost of capital, which we will assume is 12%
2. At the Value1 prompt, link the formula to the cells which contain the expected return (i.e. as we are calculating the five year NPV, it will be the profit or loss for 2013 to 2017) The formula should be \( \text{NPV}(B4,B2:F2) \)
3. The result of your formula should be $127,568

In this case, the NPV greater than zero, meaning that the return is greater than the required rate of return of 12% stated by the company. This means that theoretically, the project should be accepted. However, scenario and sensitivity analysis will help determine how sensitive the model is to changes in inputs and gain a better perspective on whether or not this project should be accepted.

What is wrong with the NPV Function?

Note that a key assumption of the NPV function is that the cash flows occur at the end of the period, whereas in reality it will probably occur unevenly throughout the year with a large portion of costs spent closer to the beginning of the period. In this example we have included the initial investment in the first year. If, however, we know that a large initial investment will be made prior to the start of the project, then this should be included in Year 0, and added to the NPV calculation like this:
=NPV(B4,B2:F2)+Y0_investment.

**IRR (Internal Rate of Return)**

IRR equates the present value of the cash inflows and the present value of the cash outflows. The decision rule for IRR is if the IRR is greater than or equal to the investor’s required rate of return, the investment should be accepted; otherwise it should be rejected.

**How to calculate the IRR:**

Using the same investment assumptions as in the previous example, the rate of return on the initial investment can be calculated as follows:

1. Select the IRR function
2. At the Values prompt, select the cells which specify the cells containing the requested information.
3. Leave the Guess prompt blank at this stage. See the section below on “The Problem with IRR” for more detail on why you need to include a guess. In this instance all the cashflows are positive, so we don’t need to enter a guess.
4. Click on the OK button and the yield (IRR) is displayed. The formula should be: =IRR(B2:F2).
4. The result of your formula should be 13.3% and look something like this:

![Fig 6.45 Using the IRR function](image-url)

In this case, the IRR is higher than the required rate of return of 12% stated by the company. This means that theoretically, the project should be accepted.
Comparing IRR with the risk factor

Just knowing about the IRR is of little use unless you can make decisions using it. To do that, you must compare the IRR with the risk factor.

There are two possible outcomes when you compare IRR with the risk.

1. The projected returns from the investment are greater than the risk. In other words, the returns from the investment are high enough to justify the risk of the investment. This is a positive recommendation to invest.

2. The projected returns from the investment are less than the risk. In other words, the returns from the investment are not high enough to justify the risk of the investment. This is a negative recommendation—cautioning against investment.

Like NPV, IRR uses all three criteria—returns, risk, and time—in its evaluation and for this reason, many managers find it an easy, accurate, and dependable tool to use.

Using NPV and IRR to Make Decisions

Just because the NPV is positive, and our IRR is greater than our required rate of return does not necessarily mean that we should go ahead with the project! Blindly accepting the output of a model is a dangerous business. As we know, a model is only as good as the assumptions that go into it and if we have included aggressively optimistic assumptions in our model, of course the NPV will look good—but this does not mean the project will do well!

All financial models should be subjected to, at minimum, a base case, best case and worst case scenario which is used to evaluate the sensitivity of outputs to changes in inputs. The NPV, IRR and Payback period which are calculated by the model should be used as a decision making factor and not the ultimate deciding factor of the fate of the project.

The Problem with IRR

There are a few problems with the IRR function that the modeller needs to be aware of.

Sometimes the IRR function in Excel can produce multiple results. Every time the cash flows changes sign; i.e. from negative to positive or from positive to negative, the formula will create another solution, and this is why it a good idea to insert a “guess” amount – especially if the sign of the cashflows is not consistent.

Take the following set of cashflows as an example:
If we use the ordinary IRR function without using the guess =IRR(B2:H2), we get a result of 13%. However, because there are several negative returns in this series of cashflows, there are multiple results. If you enter the following guesses; different IRRs will be returned:

- =IRR(B2:H2) gives a result of 13%
- =IRR(B2:H2,0) gives a result of 6%
- =IRR(B2:H2,30%) gives a result of 25%
- =IRR(B2:H2,400%) gives a result of 481%

Which one is correct? They all are! There are as many valid IRRs as there are changes in sign. Excel goes through an iterative process and comes up with the first solution it finds – which does not necessary give you result you want. This is why it’s important to use the guess. If the guess is omitted, it is assumed to be 10 percent.

Another issue with the IRR is that the way that the IRR function calculates in Excel, it assumes that cash that is generated during the investment period will be reinvested at the rate that has been calculated by the IRR. If the project is generating a lot of cash, the IRR calculation can overstate the financial benefits substantially, which is something that needs to be considered, and possibly manually manipulated in the calculations if necessary.

We don’t normally make any changes to the calculations to account for these problems, but it’s important to consider these issues when calculating the IRR.

**What difference does an X make? XNPV & XIRR**

The NPV formula assumes that the values entered into the formula are annual, in chronological order and occur at the end of the period

The XNPV formula offers more flexibility as you can specify exactly when the payments occur.

However, XNPV and XIRR are only available in Excel 2007 and later!

XNPV allows you to enter payments which occur at varying intervals, not necessarily in chronological order (although the first payment must still be shown first) and occur at any time, and is therefore a much more flexible and useful function. Also, because it does not
assume that the payments occur at the end of the period, the XNPV formula will be more accurate.

**How to calculate the NPV using the XNPV function:**

If you were to be more precise about the actual timings of the expected cashflow, you would get a more accurate NPV calculation using XNPV.

Let’s say that instead of saying that you expect a net outflow of $3.2m in 2013, we are a little more specific about when the outflows actually occur. Let’s you split that into $2m in June 2013, and the rest at the end of the year. We will also put a date in the middle of the year for subsequent years, which is more likely to reflect more accurately what will happen, instead of assuming that payments are made or received at the end of the period, as it does with the NPV calculation.

Below are the cashflows:

![Fig 6.47 XNPV function](image)

Using the XNPV formula, and assuming that the cost of capital is, say, 12%, the formula should be \( \text{XNPV}(B4,B2:G2,B1:G1) \), giving us a result of $190,828.

Similarly, the XIRR can be calculated using the formula \( \text{XIRR}(B2:G2,B1:G1) \) giving us a result of 13.8%.

Be careful when comparing the NPV and XNPV functions, as the NPV assumes the current period (the date we are in right now) is a full year prior to the first cash amount and therefore discounts the first amount back by a full 12% (or whatever the nominated discount rate is). If this is not the case, you need to exclude the first period from the function, and add it in separately, like this: \( \text{NPV}(B4,B2:F2)+\text{Y0\_investment} \).

The XNPV function, however, assumes that the current date is the date of the first cashflow – so in the example above, the current date is assumed to be 1st June 2013. Therefore the first cash amount is not discounted at all. IRR and XIRR have the same issue. This is quite misleading, as most people naturally assume that the functions, being slight variations on
the same formula, work in a similar way, but the way they treat the first cash amount is quite different!
CHAPTER 7 Tools for Model Display

Conditional Formatting

Conditional Formatting is a tool that allows you to apply formats to a cell or range of cells, and have that formatting change depending on the value of the cell or the value of a formula. For example, you can have a cell appear **bold** only when the value of the cell is greater than 100. When the value of the cell meets the format condition, the format you select is applied to the cell. If the value of the cell does not meet the format condition, the cell's default formatting is used.

This is handy for variance analysis on a P&L, for example. If you want to highlight any instance where a variance is greater than 10% higher than budget, the cell will turn red.

Remember that Conditional Formatting is the same as adding one or more formulas to each cell in which you use it, so applying Conditional Formatting to a large number of cells may cause performance degradations.⚠️ Use caution when applying to large ranges!

Conditional Formatting is greatly “improved” in Excel 2007/10. There are many more options (including the handy Data Bars!), and the limitation of three conditions has been lifted.

To apply Conditional Formatting

1. Select the desired cell/s in the spreadsheet. Let’s say, for example, you wanted to highlight in red font any cell value is between 1 and 10.

2. [Excel 07/10 and Excel for Mac 2011] On the Home tab, in the Styles group, select “Conditional Formatting”. Select Highlight Cells Rule and then Between. Type in 1 and 10, and define the desired font.
Add a new rule for each new condition – and you can have as many as you like (unlike Excel 2003, where you are limited to three rules.)

[Excel 2003] Select Conditional Formatting from under the Format menu bar. The following dialogue box will appear:

Fig 7.30 Conditional formatting in Excel 2003

Select “Highlight Cells Rule” and then “Between”. Type in 1 and 10, and define the desired font.

3. Click the Add button if you wish to add more conditions (but only three!)

To Remove Conditional Formatting

[Excel 07/10 and Excel for Mac 2011] On the Home tab, in the Styles group, select “Conditional Formatting”. Select “Manage Rules” and you can delete your chosen rules one by one. Alternatively, you can clear rules from selected cells or from the entire sheet by clicking on “Clear Rules” instead.

[Excel 2003] Go to the dialogue box as shown above, press the “Delete” button, and delete each rule.
Data Bars

Data bars are a new feature in Excel 2007 and therefore it is only possible to create them in 2007, 2010, and Excel for Mac 2011, but not Excel 2003. They are a dynamic bar chart that you can apply to any numerical data in Excel which show graphically the relative size of each value.

![Fig 7.31 Data bars in Excel 2010](image)

Go to a table of numeric data, and highlight the numbers.

4. **[Excel 07/10 and Excel for Mac 2011]** On the Home tab, in the Styles group, select “Conditional Formatting”, and then “Data Bars”.

5. Choose your desired bar.

6. Practice changing the underlying numbers, and see the bars change.

Icon Sets and Colour Scales

**[Excel 2007/10 and Excel for Mac 2011]** Icon Sets are a feature that was new to Excel 2007 which allows you to conditionally display a small icon which represents changes in data. For example, if revenues are down this month compared to last, you may choose to represent this with a red arrow pointing down, or if your cash balance is above a certain threshold, you could use a green checkmark to give the user a quick way to know everything is okay.

Your options for icons currently in Excel 2007/10 are the following. Note that currently there is no way to import any additional icons, but most likely in future releases this will be possible.
Another feature also new to Excel 2007 was Colour Scales. Looking at the image below, how would you be able to quickly assess what is good, bad and in between?

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invest 1</td>
<td>0.50</td>
<td>0.44</td>
<td>0.21</td>
<td>0.10</td>
<td>0.10</td>
<td>0.20</td>
<td>-0.27</td>
</tr>
<tr>
<td>Invest 2</td>
<td>0.60</td>
<td>0.33</td>
<td>0.36</td>
<td>0.25</td>
<td>0.08</td>
<td>0.30</td>
<td>0.44</td>
</tr>
<tr>
<td>Invest 3</td>
<td>0.20</td>
<td>0.45</td>
<td>0.28</td>
<td>0.27</td>
<td>0.32</td>
<td>0.39</td>
<td>0.44</td>
</tr>
<tr>
<td>Invest 4</td>
<td>0.08</td>
<td>0.17</td>
<td>0.21</td>
<td>0.01</td>
<td>0.28</td>
<td>0.31</td>
<td>0.05</td>
</tr>
<tr>
<td>Invest 5</td>
<td>0.07</td>
<td>0.08</td>
<td>0.11</td>
<td>0.31</td>
<td>0.35</td>
<td>0.01</td>
<td>0.26</td>
</tr>
<tr>
<td>Invest 5</td>
<td>0.02</td>
<td>-0.08</td>
<td>-0.13</td>
<td>-0.25</td>
<td>-0.26</td>
<td>-0.21</td>
<td>-0.27</td>
</tr>
</tbody>
</table>

By using Colour Scales, you can assign rules for what is good, bad and in between, and display the data with colour to represent these rules. You can see in the image below how data is more useful when the user can quickly spot areas of concern.

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invest 1</td>
<td>0.50</td>
<td>0.44</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>Invest 2</td>
<td>0.60</td>
<td>0.33</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>Invest 3</td>
<td>0.20</td>
<td>0.45</td>
<td>0.28</td>
<td>0.27</td>
</tr>
<tr>
<td>Invest 4</td>
<td>0.08</td>
<td>0.17</td>
<td>0.21</td>
<td>0.01</td>
</tr>
<tr>
<td>Invest 5</td>
<td>0.07</td>
<td>0.08</td>
<td>0.11</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Enhancements to Conditional Formatting in Excel 2010

[Excel 2010 and Excel for Mac 2011] You have access to more icon sets, including triangles, stars, and boxes. You can also more easily hide icons from view, as is evident from the example below.

![Fig 7.35 Hiding icons in Excel 2010](image)

Being able to hide the icons is a new feature applicable to Excel 2010 and Excel for Mac 2011 only.
Editing the Rule gives you a variety of options:

![Editing icon set rules in Excel 2010](image)

**Fig 7.36 Editing icon set rules in Excel 2010**

You can also easily mix and match data bars, colour scales and icons from different sets. In the example below the same range has both data bars and icons:

![Applying multiple types of conditional formatting on a single cell](image)

**Fig 7.37 Applying multiple types of conditional formatting on a single cell**

The automatic negative bars as shown above are a new feature applicable to Excel 2010 and Excel for Mac 2011 only.

In Excel 2010 and Excel for Mac 2011, you can also create conditional formatting with references to a different sheet in the workbook. This means that you no longer need to copy or link your data to the same sheet when using conditional formatting.
Sparklines

Just like conditional formatting and the icon sets, Excel also has another interesting micro-charting feature termed the Sparkline. These lines are not as descriptive as the regular graphs and charts, but are very effective in displaying a quick view about the trends in the data or metrics. This is a new feature introduced in Excel 2010, so it is an enhancement over the older versions. At the same time, since this is the first release, the design is not necessarily as polished as a modeller would prefer.

How to create a Sparkline

1. Select the data series for which the Sparkline needs to be created.

<table>
<thead>
<tr>
<th>Year</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
<td>$149,029</td>
</tr>
<tr>
<td>19</td>
<td>$80,000</td>
<td>$146,474</td>
<td>$156,814</td>
<td>$165,274</td>
<td>$155,114</td>
<td>$147,603</td>
<td>$159,488</td>
<td>$160,725</td>
<td>$212,261</td>
<td>$111,039</td>
</tr>
<tr>
<td>20</td>
<td>$96,029</td>
<td>$156,532</td>
<td>$166,518</td>
<td>$176,274</td>
<td>$165,114</td>
<td>$157,603</td>
<td>$169,488</td>
<td>$170,725</td>
<td>$212,261</td>
<td>$111,039</td>
</tr>
<tr>
<td>21</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
<td>$140,029</td>
</tr>
</tbody>
</table>

*Fig 7.38 Data to contain sparklines*

2. Under the Insert Tab in Sparklines ribbon, select Line. Note that Sparklines are found in the Charts ribbon under “Insert Sparklines” in Excel for Mac 2011.

*Fig 7.39 Choosing sparklines from the Ribbon*

3. In the Create Sparklines dialog, the data range is automatically selected, enter the cell or cell range in which you want the sparklines to be placed. Click Ok.

*Fig 7.40 Sparklines dialog box*
4. The sparklines are drawn in the selected cells.

![Completed report with sparklines](image)

**Fig 7.41 Completed report with sparklines**

**Hot Tip:** Use a combination of sparklines and icons to create more meaningful representations of data points.

**Editing Sparklines**

The attributes of the sparklines can be modified from options under the Design tab.

**Editing Data:** Select the sparklines and click on the Edit Data in Sparkline ribbon under the Design tab.

![Editing sparklines](image)

**Fig 7.42 Editing sparklines**

This will open the Edit Sparklines dialog where you can change the data range as well as the Sparkline location.

![Edit sparklines dialog box](image)

**Fig 7.43 Edit sparklines dialog box**
Editing Chart type: Select the sparklines and click on one of the three options in the ribbon under the Design tab. The three options are:

- Line: Simple Line Graph
- Column: Column Graph, similar to area plots
- Win/ Loss: Map the data in Win/ Loss format

![Fig 7.44 Changing sparkline type on the Ribbon](image)

Highlighting Important Points: Excel allows you to highlight important data points in the sparklines. You can choose from:

- High Point
- Low Point
- Negative Points
- First Point
- Last Point

To highlight important data points, select from a series of checkboxes in the Show ribbon, under the Design tab.

![Fig 7.45 Highlighting sparkline data points on the Ribbon](image)

Sparkline Styles: You can also edit the style of the sparklines by changing the line colours and marker colours.

![Fig 7.46 Editing sparkline formatting on the Ribbon](image)
Changing Properties for group of sparklines

While each Sparkline represents a unique series of data, you can edit the properties of a group of sparklines using the group and ungroup features. When the sparklines are grouped, you can edit the properties for the entire group, even if you select just a single line from the group.

Fig 7.47 Grouping sparklines together

To separate out the sparklines in the group, select the ungroup in the Group ribbon under the Design tab.

**Hot Tip:** If you want to create a series of sparklines for a series of data points, you can create it for the first list and just copy paste the Sparkline to auto-create it for the rest of the lists.
Form Controls

Form controls are objects such as drop-down boxes and option buttons that sit over the top of Excel sheets, like charts do. They can interact with formulas so that when the option is selected, it changes the formula and can drive the model, and can be especially useful for scenario selection, and assumptions in a financial model.

Building form controls into a financial model can be time consuming for the modeller to build, but they do make the model look very professional and the model interface becomes extremely easy to use.

Accessing Form Controls

[Excel 2003] If you are using Excel 2003, there is a “Forms” toolbar which can be placed on the menu by going to View – Toolbars – Forms

![Fig 7.59 The forms toolbar in Excel 2003](image)

[Excel 2007/10 and Excel for Mac 2011] If you are using Excel 2007 or 2010, you will need to have the Developer tab showing in the ribbon. If you cannot see it in the Ribbon, you’ll need to change your options so that you can see the Developer tab in the Ribbon. See the section on “Showing the Developer Tab in the Ribbon” for instructions on how to do this.

Checkboxes

Tick boxes (or checkboxes, as Microsoft calls them!) are very handy tools that can help the user choose their desired options from a list. By building checkboxes into a financial model, the model developer allows the user to customise the model by adjusting the combinations of inputs for the financial model assumptions.

How to create Checkboxes:
1. **[Excel 2007/10]** Go to the Developer tab on the Excel toolbar and click on “Insert”. If you cannot see the Developer tab, follow the instructions above on how to show the Developer tab in the Ribbon.

   ![Fig 7.60 Insert controls icon](image)

   **Fig 7.60 Insert controls icon**

**[Excel for Mac 2011]** Go to the Developer tab on the Excel toolbar. If you cannot see the Developer tab, follow the instructions above on how to show the Developer tab in the Ribbon.

![Fig 7.60.1 Form Controls in Excel for Mac](image)

**Fig 7.60.1 Form Controls in Excel for Mac**

**[Excel 2003]** Select Checkbox from the Forms Toolbar.

2. From the “Forms Controls” select the Checkbox as shown below.

![Fig 7.61 Inserting the check box](image)

**Fig 7.61 Inserting the check box**

3. Draw the checkbox and you should see the box with a name “Check Box 1”. Subsequent Check Box names will be “Check Box 2”, “Check Box 3” and so on. For ease of use and understanding you must place the checkbox next to the item in the list you want to check.

Note that you can change the checkbox name to anything you prefer. In this example the
item name itself has been used to keep things simple.

**Fig 7.62 Check box in the sheet**

4. You can now link the tick mark in this box to any function in your financial model. To do that you need to open the “Format Control” associated with the checkbox. This can be done in two ways.
   a. Right click on the checkbox and select “Format Control”

   **Fig 7.63 Assigning check box options**

   b. Alternatively, in Excel 2007/10, select the checkbox and click “Properties” on the ribbon under the Developer tab.
5. In the “Format Control” dialog you can select the default value of the checkbox (unchecked, checked or mixed) and the cell to which you want to link the checkbox status. (Here it is linked to C5). To enhance the look of the checkbox you can select 3D shading.

6. The cell to which the status is linked will have a Boolean value of TRUE or FALSE. See below for more detail on “Boolean Logic in Form Controls”.

7. You can add multiple checkboxes corresponding to the items in the list. Based on the Boolean status you can design your financial model to pick the respective items.
8. Practice adding multiple items to the list and associating checkboxes to each item.

Option Button

Sometimes the user may need to choose one item from a list which drives the rest of the calculations and analyses. The Option button (sometimes called a Radio button) offers you the ability to create this logic to restrict the number of options the user can choose in the financial model.

How to create option buttons:

2. From the “Forms Controls” select the “Option Button” as shown in the figure below.

[Excel for Mac 2011] Go to the Developer tab on the Excel toolbar. Select the “Radio Button” as shown in the figure below.
Fig 7.67.1 Radio Button in Excel for Mac

Excel 2003] Select Option Button from the Forms Toolbar

3. Draw the Options button in front of the item in the list. Like the checkboxes, the Option button default name starts with “Option Button 1” and increases with each new button. The name can be later customised.

Fig 7.68 Worksheet with option button

4. You can edit the properties of the button using “Format Control”. You can access the format control dialog in the same manner as for Checkbox.
   a. Right Click – Format Control
   b. Click Properties in the ribbon under the Developer tab

5. In the “Format Control” dialog you can select the default value of the checkbox (unchecked, or checked) and the cell to which you want to link the checkbox status. (Here it is linked to D5). To enhance the look of the option button you can select 3D shading.
6. The cell to which the status is linked will have a numerical value of the item in the list\(^1\). In the figure below the 4\(^{th}\) item was selected, hence $D5 = 4$.

7. You can add multiple option buttons corresponding to the items in the list. Based on the item number selected, you can design your financial model to carry out the necessary analysis.

8. Practice adding multiple items to the list and associating option buttons to each item. As you switch between each item in the list you can observe the number change in the linked cell.

---

\(^1\) By design, all the Option buttons will always refer to a single cell, unlike checkboxes. If you change the cell link for one option button, it will automatically update the rest.
Spin Buttons

Spin buttons are useful for controlling what the user can enter into the input fields in your financial models. By including a spin button, the user can increase or decrease the number in the increments that you specify. For example, if you set it to increments of 5, and the current value is 400, when the user hits the up arrow, it will change to 405, then 410, 415 etc., and the same with the down arrow.

How to create a Spin Button:

1. Go to the Developer tab on the Excel toolbar and click on “Insert”
2. From the “Forms Controls” select the “Spin Button” as shown below

![Fig 7.71 Inserting the spinner](Excel for Mac 2011) Go to the Developer tab on the Excel toolbar. Select the “Spin Button.”

![Excel 2003] Select Spinner from the Forms Toolbar

3. Draw the spin button next to the input box for the value you want to control. The figure below shows an example of the spin button.

![Fig 7.72 Completed spin button](Excel 2003)

4. You can edit the properties of the button using “Format Control”. You can access the format control dialog in the same manner as for Checkbox.
   a. Right Click – Format Control
   b. Click Properties in the ribbon under the Developer tab
5. On the Format Control dialog, you can specify the current or default value in the cell, the minimum value (here it is set to 0), maximum value (here it is 1,000) and the
incremental change (here it is 10). You must also specify the cell in which you want to change the values in the Cell Link.

![Format control dialog box](image)

*Fig 7.73 Format control dialog box*

6. Practice pressing the arrows up and down.

**Combo Boxes**

Creating a drop-down box is an important tool in many user-friendly models where the modeller wants the user to select from a pre-defined number of options, making the model easier to use, and also avoiding mistyping and potential errors. The advantage of the combo boxes is that you eliminate any ambiguity in the user inputs and still keep things simple for the user to provide the inputs.

![Completed combo box drop-down](image)

*Fig 7.74 Completed combo box drop-down*

**How to create a combo box:**

1. Go to the Developer tab on the Excel toolbar and click on “Insert”
2. From the “Forms Controls” select the “Combo Box” as shown in the figure below
Fig 7.75 Inserting the combo box

[Excel 2003] Forms Toolbar – “Combo Box”. Once you have found the combo box tool, the instructions from here are the same in all versions of Excel.

[Excel for Mac 2011] Note that combo boxes can only be used in conjunction with VBA in Excel for Mac 2011, and therefore the instructions in this section do not apply to Mac users. Please see the section on “Use of Macros in Financial Modelling” for more details on how to use VBA to build form controls in Excel for Mac 2011.

3. Draw the Combo Box next to the input box whose input you want to control. The figure below shows an example of the combo box button.

Fig 7.76 Drawing the combo box

4. To fix the inputs, create an input box with inputs such as shown below. Here we are using the range B8:C11.
5. You can now associate these inputs with your combo box using the “Format Control”. You can access the format control dialog in the same manner as for Checkbox.
   a. Right Click – Format Control
   b. Click Properties in the ribbon under the Developer tab

6. On the Format Control dialog, give the input range (here it is B9:B11) and the Cell Link (here it is $D$8). Select 3-D shading to enhance the combo box.

7. INDEX is a more graceful manner of representing this link. Use the formula =INDEX(B9:C11,D8,2) to create the necessary linkage.
8. To keep the form clean, typically modellers would change the font colour of cell D8 to white to that the text is invisible to the user but available for calculation.

9. Practice changing the combo box, and see the values change

The combo box is an alternative to using the Data Validation drop-down tool. See the section on “Bullet-Proofing your Model” for instructions on how to create a Data Validation in-cell drop-down box. The only advantage of the combo box is that the drop-down arrow is visible whether the cell is selected or not. This is a significant advantage if users are not very Excel savvy.

From the modeller’s perspective, however, a combo box takes much longer to build, requiring more cells for the process, and is therefore less efficient. For these reasons, most modellers prefer the Data Validation drop-down tool.

**Boolean Logic (Binary Code) in Form Controls**

Most form controls use Boolean logic, which converts everything to two values; a one or a zero. For example, if a Checkbox is ticked, it will return a value in the output cell of TRUE which equals 1, and if it is un-ticked, it will be FALSE, which equals zero. We can use the output cell to drive a formula.

In the example below, the output cell is B5. When the checkbox is ticked, it shows the value TRUE which is equal to one, so this has been by using the formula =B7*((B5*B2)+1)
Fig 7.74 Ticked check boxes drive calculation

If the checkbox is left un-ticked, the output cell would display FALSE, which gives the value zero, and according to the formula, the GST would then not be included in the total.

Fig 7.75 Un-ticked check boxes drive calculation

Similar formulas can be created to link to other form controls such as option buttons and combo boxes.

Form Controls vs. ActiveX Controls

If you are using Excel 2007 or 2010, you may have observed that the “Insert” button under the Developer tab has two set of options Form Control and ActiveX Controls. While both have the same set of functions lists, they have a slight difference. The ActiveX controls are tailor-made for Microsoft Office environment and use Microsoft ActiveX to enhance their functionality. If you are absolutely certain that your financial model would be used in the MS Office environment alone, then you could possibly add a few additional enhancements to your controls. However if your users would also be using other environments like Mac OS, the ActiveX controls can cause problems – as Apple MAC does not support Microsoft ActiveX.

If you want to keep your financial modelling product platform independent, stick to Form Controls. As far as core functionality is concerned, Form controls will offer you all the necessary flexibility required for a smooth user experience.
CHAPTER 8 Tools for Financial Modelling

Array Formulas

Array formulas are an advanced type of Excel calculation. Before we launch into what an array formula is, let me begin by saying that I don’t recommend you include array formulas in your model unless it performs a purpose that is not possible any other way! Array formulas can be difficult to understand, and hard to edit if you don’t know what they are because you cannot change part of an array. In general, array formulas are very difficult to edit and audit and basically add a whole new layer of complexity that makes your model far more complicated than it really needs to be. Most array formulas I have seen are usually built by consultants who want to ensure that the client is unable to edit the model themselves, so guaranteeing themselves work in the future!

An array is basically just a collection of data of the same type that can be treated as a single entity. An array formula treats the entire array as a single input to the formula. The reason that array formulas are difficult to understand is because they don’t follow the usual rules of an ordinary formula. “Trace precedents” and “Trace dependents” will work on arrays, but don’t help very much! Many people don’t know how to use them, and if you have never seen an array formula before, they can be pretty hard to work out. Therefore if you are building a model for someone else to use, using array formulas to build your model are not a great idea because if someone needs to make changes to the formula they may have trouble.

However, in order to seriously consider yourself an Excel power user, you should know how to build an array formula – or at least be able to recognise and edit one if you come across it – if only so that you can remove it from the model, and replace it with a far more auditable and user-friendly formula.

Array formulas are a very powerful tool, and there are many reasons why analysts use them. One of the main advantages is security – nobody can accidentally delete part of the array block when using an array formula. Also, because the data can be manipulated as a whole block and used in the formula as a single unit, it’s a lot harder to make a mistake when building the formula.

You can tell if someone has used an array formula because it includes curly brackets ({} in the formula. Data tables (see the section on “Scenarios” for how to build a data table) are a type of array formula – and probably one of the most useful sort of array formula – especially for the financial modeller.

Advantages of using Array Formulas

- They ensure consistency because all formulas in the table are exactly the same
- A model containing arrays will use less memory and be more efficient
- Because it’s not possible to change a single cell on its own within an array formula, it is unlikely that you or someone else will change your formula accidentally.
Because array formulas are difficult to understand, it means those with only basic Excel knowledge are less likely to change (and mess up!) your formulas.

Array formulas do make it possible to perform some calculations that would otherwise be impossible using ordinary formulas. This is really the only reason you should use an array formula!

Disadvantages of using Array Formulas

- Although array formulas use less memory, if you use too many large arrays in one model it can slow down your calculation.

- You cannot use column references (such as "A:A" or "D:D") in your array formulas. This is not best practice in financial modelling anyway, so no great loss!

- They are difficult to audit for many Excel users and require specialist skills (both an advantage and a disadvantage!)

Simple Array Formula Example

1. Create two simple blocks of data (arrays) as below:

   ![Fig 8.7 Sample data](image)

2. Highlight the block of cells C1:C5 and type/highlight =A1:A5*B1:B5 in the formula bar
Fig 8.8 Entering an array formula

3. Now, hold down the Control and Shift keys whilst hitting Enter at the same time (CTRL+SHIFT+ENTER). (COMMAND+SHIFT+ENTER) in Excel for Mac 2011

Fig 8.9 Completed array formula

4. The formula will appear in the whole array block. Note that the curly brackets have appeared around the formula.

5. Note that you now cannot make any changes to the array block.

Nb: Array formulas are sometimes referred to as “CSE formulas,” because you press CTRL+SHIFT+ENTER to enter them.

Uses of Array Formulas

Financial Modellers generally advocate the use of Array formulas on when it is not possible to do it any other way. Some examples of these situations are as follows:
1. Data tables are a form of array formula that is extremely useful for scenarios and sensitivity analysis in Financial Modelling. Instead of changing inputs one at a time, it shows hypothetically what an outcome would be simultaneously in a single table.

2. TRANSPOSE allows you to transpose a range of data either vertically or horizontally, and retain links.

3. Using a SUM and a VLOOKUP together as an array formula makes a very concise formula. For example, =SUM(VLOOKUP(value, list, {2,3,4},0)) will give you sum of values in columns 2, 3 and 4 instead of having to do =VLOOKUP(value, list,2,0)+ VLOOKUP(value, list,3,0)+ VLOOKUP(value, list,4,0). Note: you need to hold down Control, Shift and Enter for this to work.

**Transposing Data using an Array**

One of my favourite uses of an array formula is to create an =TRANSPOSE function which will transpose data, allowing a link to be maintained within the model.

Let’s say you have the following data in a table:

<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>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Reno</td>
<td>71</td>
<td>76</td>
<td>83</td>
<td>89</td>
<td>98</td>
<td>104</td>
<td>108</td>
<td>105</td>
<td>101</td>
<td>91</td>
<td>77</td>
</tr>
<tr>
<td>4</td>
<td>Carson</td>
<td>72</td>
<td>76</td>
<td>81</td>
<td>88</td>
<td>93</td>
<td>101</td>
<td>105</td>
<td>103</td>
<td>103</td>
<td>91</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>Tahoe</td>
<td>59</td>
<td>60</td>
<td>67</td>
<td>67</td>
<td>81</td>
<td>90</td>
<td>93</td>
<td>94</td>
<td>87</td>
<td>80</td>
<td>70</td>
</tr>
</tbody>
</table>

*Fig 8.10 Temperature data*

And you’d like to transpose the data so that Months and Cities are switched around the other way like this:
Before you start cutting and pasting, you could try the transpose values tool.

1. Highlight the original table, and copy it to the clipboard using Control C (Command C in Excel for Mac 2011)

2. Right-hand click in the cell that you want the data to appear, and select “Paste Special”

3. Tick the “Transpose” box as shown below, and press OK

![Fig 8.12 Paste special dialog box](image)
This will paste all of the transposed values as hard-coded data in a new table. Using this method creates two separate, independent tables of data.

**Using a TRANSPOSE Array Function**

However, you may wish to maintain the original table as the source data, and have the second table linked to the first. Instead of linking each cell individually (time consuming and prone to error) you could use a TRANSPOSE Array Function.

1. Highlight a range with exactly the correct rows and columns for the destination data (Tip: it is probably easiest to use the Paste – Transpose procedures above to determine exactly how big the range needs to be)

2. In the formula bar, type the formula `=TRANSPOSE()` referencing the original data; i.e. `=TRANSPOSE(A2:M5)`

3. Now, hold down the Control and Shift keys whilst hitting Enter at the same time (CTRL+SHIFT+ENTER). (COMMAND+SHIFT+ENTER) in Excel for Mac 2011

4. The formula will appear in the whole array block. Note that the curly brackets have appeared around the formula, so the formula will be `{=TRANSPOSE(A2:M5)}`

5. Note that changes you make in the original block of data will be reflected in the second block.
Pivot Tables

Excel pivot tables are very useful and powerful feature of Excel. They can be used to summarise, analyse, explore and present your data. A pivot table is a great reporting tool that sorts and sums independent of the original data layout in the sheet.

For example, you can take the sales data with columns like salesman, region and product-wise revenues and use pivot tables to quickly find out how products are performing in each region. Below is a snapshot of a pivot table that has been used to summarise the annual fees expected by each club member by department.

For how to create a pivot table from scratch, see the section below on “How to create a Pivot Table”.

The layout of pivot tables change significantly in the upgrade from Excel 2003 to 2007. Although the basic functionality remains the same (with the usual improvements in 2007), because the wizard layout has changed to a field list, if you were an avid pivot table user in Excel 2003, the new layout does take some getting used to.

The use of Pivot Tables in Financial Modelling

Useful as they are, pivot tables are not used very widely in Financial Modelling. This is because they are static – until refreshed, and this does not fit very well with the methodology of most financial models. As discussed in earlier sections, the whole point of financial modelling is that you are able
to change the inputs, and the outputs change! It’s far too easy for a user to change the inputs, and then forget to refresh the pivot table, so using incorrect outputs from the model.

It is for this reason that I recommend the use of an alternative live, dynamic formula, such as SUMIF or SUMIFS in place of a pivot table when building a classic financial model which depends on inputs being changed. See the section on “Aggregation Functions” for how to use the SUMIF and SUMIFS functions.

Whilst it is possible to change the settings so that the pivot table will automatically refresh whenever the file is opened, there is no guarantee that the user will open and close the file before using the data, and it cannot be relied upon.

It’s not a bad idea to change the settings to automatically refresh in any case, however. You can do this by right-hand clicking on a pivot table, then:

Excel 07/10 and Excel for Mac 2011] go to Pivot Table Options, and on the Data Tab, select “Refresh data when opening the file.”

[Excel 2003] go to Table Options, and under Data Options, click on “Refresh on open”

Common Uses of Pivot Tables

Pivot tables are more widely used in management reporting, than in pure financial models.

Here are some commonly used examples of pivot tables:

- Summarising data, such as finding the average sales for each region for each product from a product sales data table.
- Listing unique values in any column of a table (this can also be done with the “remove duplicates” functionality of Excel 2007/10) but prior to the 2007 release, pivot tables were a common way of obtaining a unique list.
· Filtering, sorting, drilling-down data in the reports without writing any formulas or macros.
· Transposing data; i.e. moving rows to columns or columns to rows.

Other things you should know about Pivot Tables

· You can apply any formatting to the pivot tables. Excel has some very good pivot table formats (and they are better in Excel 2007 and 2010 and Excel for Mac 2011).
· You can easily change the pivot table summary formulas. Right click on pivot table and select “summarise data by” option.
· You can also apply conditional formatting on pivot tables although you may want to be a bit careful as pivot tables scale in size depending on the data.
· Whenever the original data from which pivot tables are constructed, just right click on the pivot table and select “Refresh Data” option.

**Hot Tip:** If you want to drill down on a particular summary value, just double click on it. Excel will create a new sheet with the data corresponding to that pivot report value – very useful!

Macros in Financial Modelling

Many aspiring modellers think that they need to be proficient in macro-building in order to become a “serious” financial modeller. Whilst it’s not a bad idea to have a working knowledge of macros - and the language they are built in, Visual Basic for Applications (VBA) - it’s certainly not critical for a financial modeller to become super VBA programmer. As discussed in previous sections, the best sort of financial model uses the simplest tools, and introducing macros to a financial model brings a whole new level of complexity.

Introduction to Macros

Before launching further into the pros and cons of including macros in your financial model, let’s have a quick overview of what macros are and how to build them. VBA in Excel is a huge topic in itself and one I’m not planning on going into too deeply here. Once you start writing macros it’s quite easy to get hooked as you can do some truly amazing things in Excel which will bring the functionality of your work in Excel to a whole new level. If you do develop an interest in the subject, there are lots of books and training courses specifically on macros and VBA. Make sure that you buy a book or attend a course that is specifically on VBA for Excel and its use in finance as there is a lot of programming that is done in VBA that is not finance-related.
A macro is a collection of commands that are performed in a set order. A macro enables you to repeat operations that you would normally do by hand, but it is much faster, and when written correctly, much more reliable. Often, a macro will do in seconds what takes hours or days by hand. It can also perform tasks that are physically impossible manually. If you find you are performing the same commands or actions over and over again, in exactly the same sequence, you can create a macro to record all those actions for you. You can then assign the macro to a button and then run the macro using a single click and even assign the macro to a keyboard command.

**Macro Settings**

Before we begin creating macros, there are a few settings in Excel you’ll need to change:

1. **Showing the developer tab in the ribbon** is not strictly required to record and run macros, but it contains tools which makes it easier to run and edit macros, as well as containing some handy control tools and buttons.

2. **Changing the security settings** is required, and you **must** change this setting if you want to run macros.

The default installation version of Excel does not have these settings. You will only need to change these settings once, and you will not need to do it again unless you re-install your version of Excel, or move to another machine.

**Showing the Developer Tab in the Ribbon**

This is only relevant for Excel 2007/2010 and Excel for Mac 2011 users. While it is possible to record a macro by clicking on the launch button in the bottom left hand corner of your screen as shown below, the developer tab contains other functions which are useful when developing macros and other tools, so it’s best to install the developer tab.
If you do not see the Developer Tab, click on File – Options – Customize Ribbon – Check the “Developer” box. (Excel for Mac 2011 users go to Excel – Excel Preferences – Ribbon – Check the “Developer” box.) For more detailed instructions on how to do this, go to the section on “Showing the Developer Tab in the Ribbon” under Form Controls.

Setting Macro Security Levels

Macros perform various automated functions which run via executable code. While most macros are useful and harmless, some malicious macro viruses, can destroy data or otherwise damage your machine. For this reason, many people decide to disable macros altogether or turn security up so that macros can only be run from documents written by trusted publishers. However, turning up the security may disable legitimate macros, so changing the security level is a personal decision. You may also want to change this setting before you open documents from different sources.

You will need to decide on the level of security appropriate for the way that you work. If you only ever run macros that you have written yourself, then Low may be suitable. High will be difficult to work with, as it won’t allow you to run macros at all. If you are often running macros from a trusted location, you may choose to select the "Enable all macros" option.
To change your macro security settings:

**[Excel 2010]**

1. Click on the File menu in the top left hand corner of your screen.
2. Scroll down to “Options” at the very bottom and select it.
3. Select “Trust Centre” and click on “Trust Centre Settings”
4. Choose the desired level of security.
5. Close the model down, and then reopen it.

**[Excel for Mac 2011]**

1. Click on the “Excel” menu at top left hand corner of your screen.
2. Click “Preferences”
3. Click the “Security” button, and go to the “Macro Security” section
4. Choose the desired level of security.
5. Close the model down, and then reopen it.

**[Excel 2007]**

6. Click on the Microsoft Office button in the top left hand corner of your screen.
7. Under "Excel Options", go to the "Trust Centre" section.
8. Click on the "Trust Centre Settings" button, and go to the "Macro Settings" section.
9. Choose the desired level of security.
10. Close the model down, and then reopen it.

**[Excel 2003]**

1. Go to Tools – Options, and on the “Security” tab, press the “Macro Security” button.
2. Choose from low, medium or high security. You’ll need to choose either Medium or Low security in order for your macros to run.
   
   (a) High security - you won’t be able to run macros
   (b) Medium security – it will prompt you whether you want to run macros or not
   (c) Low security – it will allow all macros to run
Recording & Running a Simple Macro

The best way to get started with macros is to record one.

To Record a Macro:

These are the basic steps to create a simple macro by recording it.

[Excel 2007/10] In the “View” tab in the “Macros” section, click on the arrow under the “Macro” button, then “Record Macro”.

[Excel for Mac 2011] In the Developer ribbon in the “Visual Basic” section, click the “Record” button.

[Excel 2003] Go to Tools – Macro – Record New Macro

This will bring up the Record Macro dialog box.

Excel will come up with the suggested name Macro1. You can type over this. Macro names must be one word with no spaces. It is best to give the macro a name that relates to what it does, i.e. PrintCompanyData, rather than a non-descriptive name such as Macro1. Type a name for the macro into the Macro Name box. You can also allocate a shortcut key.

YOU ARE NOW RECORDING YOUR MACRO. EVERY COMMAND OR ACTION YOU PERFORM WILL BE ADDED TO THE MACRO!

1. Perform all the actions you wish to be included in the macro. E.g., print, save, change colour, insert or delete rows or columns will all be included in the macro if you perform these actions while the macro is recording.
2. When you wish to stop recording the macro:
   - [Excel 2007/10] Click on the arrow under the “Macro” button, then “Stop Recording”, or
   - [Excel for Mac 2011] In the Developer ribbon in the “Visual Basic” section, click the “Record” button again, which will stop the recording.
   - [Excel 2003] Press the Stop Recording button which has appeared, or go
This macro will save within the Excel you used when recording the macro. This macro is available whenever you have the file open.

To Run the Macro:

1. **[Excel 2007/10]** In the “View” tab in the “Macros” section, click the “Macro” button.
   **[Excel for Mac 2011]** In the Developer ribbon in the “Visual Basic” section, click the “Macro”.
   **[Excel 2003]** Go to Tools – Macro – Macros.
   This will bring up the Macro dialog box, containing a list of all the available macros.
2. Click in the list to select the macro you wish to run.
3. Click on the “Run” button. All the actions you performed in the steps above will repeat themselves.

Practical Exercise

OK, let’s try recording a macro that will be useful. Let’s say you want to change the formatting of a cell from 150000 to $150,000 – a very commonly performed task. (This macro will be useful, as it’s not possible change this formatting with a single mouse click or keystroke.)

1. Type 150000 into cell A1 on a blank worksheet.
2. Begin recording, as shown above. Call the macro “FormatCurrency”. Assign a shortcut key, e.g. Ctrl + m, and press OK.

![Fig 8.30 Naming the recorded macro](image-url)
3. Select cell A1, and change the formatting to $150,000. This can be done through the menu bar, or right-hand click, Format cells and change the currency to the desired format.

4. Stop recording.

5. Now we can test to see if the macro worked. Go back to cell A1 and change the formatting to a percentage, for example.

6. Then run the macro as shown above. The formatting of cell A1 will revert to the formatting we specified in the macro.

Now we can edit the macro and make changes within the visual basic code.

### How to edit a macro:

- Select the “Macros” icon from the “Code” tab on the ribbon (Visual Basic tab in Excel for Mac 2011). The Macro dialog box contains a list of all the available macros.
- Select the macro that you want to edit, and press “edit”. This will take you into Visual Basic for Applications coding editor, where you can manually edit a macro.
- You can change things such as sheet names or cell references, rather than re-recording the macro. You can even make the macro run more efficiently by removing the duplication of the .select command.
- Make changes to the code, and close down the editor.

Let’s edit our macro.

7. Go into the VBA editor as shown above. If you haven’t been into the editor before, it’s a bit scary but don’t worry. You can always close down the editor, and you’re back into your Excel model.

8. The screen should look something like this:

   ![Fig 8.31 Viewing the recorded macro in the VBA editor](image)
The text in green font is purely descriptive text. Any text preceded by the ‘’ sign will be ignored by the macro when running. You can delete or replace this with documentation of your own, describing what the macro does.

Take a look at the code. It contains two steps: It selects cell A1, and formats it as we have specified. This means that every time we run this macro, it will select cell A1 and format it. If we remove the reference to cell A1, it will change the formatting of whatever cell is selected. This is much more useful!

9. Remove the descriptive text, and the first line of code. Your code should now say:

```
Sub FormatCurrency()
    Selection.NumberFormat = "$#,##0"
End Sub
```

10. Close down the VBA editor and return to your model.

11. Type some random numbers anywhere in your model – even on a different page.

Select the cells, and run the macro. These numbers should change to your desired format.

Creating Macro Buttons

You can assign a macro to almost any object or shape on your Excel worksheet, including shapes, buttons, and even combo-box drop-downs. The most commonly used object (that users are used to seeing) is a rectangular-shaped button. There are two ways of creating these buttons, either as shapes or traditional form buttons.

How to create a shape:

1. **[Excel 2007/10]** On the “Insert” tab, in the “Illustrations” group, a “Drawing Shape” can be used as a macro button. The style can be changed to look like this:

2. **[Excel 2003]** Go to Insert – Picture – Autosshapes (Insert – Shape... in Excel for Mac 2011). This will bring up the AutoShapes toolbar and you can select a rectangle from the Basic Shapes. Change the colours by right-hand clicking on the outside edge of the shape and select “Format AutoShape”. The button will be flat, rather than three dimensional like the button below:

3. Right hand click to assign the macro to the button. Whenever you press this button, the assigned macro will automatically run.
Alternatively, the more traditional macro button such as this one can be used.

If you wish to create a traditional form macro button in [Excel 2007/10], it’s still possible, but it’s been very well-hidden! You need to have the developer tab showing before you can create these buttons. See above for how to make the Developer tab show in the Ribbon.

How to create a traditional form button:

1. [Excel 2007/10] On the “Developer” tab, in the “Controls” group, select the insert button and select the form button as shown.

   Fig 8.32 Inserting the button in Excel 2007/10


   Fig 8.33 Form Controls ribbon in Excel for Mac 2011

   [Excel 2003] The form button is located on the toolbar in Excel 2003. To display the forms toolbar, go to View – Toolbars – Forms. Select the button from the toolbar.

   Fig 8.34 Forms toolbar in Excel 2003

1. Now click and drag over the spreadsheet to create the shape
2. When you let go of the mouse button, the Assign Macro dialog box will appear. Select appropriate macro
3. To change the assigned macro, right click on the shape and click on “Assign Macro”
4. You should add the appropriate text and format the shape with colour, shadow, etc
Macro in Financial Modelling Case Studies

Here are a few of the most commonly used problems in financial modelling that are solved by macros.

There are hundreds of situations where macros can save time and increase accuracy. They might not necessarily be built and saved within financial models, but they could be used in the collection of the data that goes into model. Here is a small sample of the kinds of situations where macros come in handy for financial modeling and analysis:

- We are doing a salary comparison between different countries, and wanted to how much we’d need to pay in Malaysian Ringgit so that their salary is the same as someone working in Singapore. If they earn the equivalent of 100,000 MYR after tax, how much would we need to pay them before tax? The problem is that tax is calculated on pre-tax salary, so our formula gives us a circular reference. We can use a macro with a goal seek to repeatedly set the pre-tax salary to the amount entered by the user.

- Fifty identical budget templates have been created, and you discover a formula needs to be changed. Instead of making the change fifty times, record a macro – it will be much quicker, and far less prone to error.

- A non-profit organisation sets their pricing so that all costs are fully recovered to break even. The costs keep changing though and the user is a senior account manager who doesn’t know how to do a goal seek in Excel. The modeller creates a simple macro using a goal seek so that all the user needs to do is change the costs and press the button to find out how much the pricing needs to change to under the new costing.

- A dump of information containing several thousand rows is exported from a database into Excel every day. The data needs to be formatted and manipulated manually to be used in a daily takings report. Automating this with a macro can literally save hours of manual data manipulation.

- A reporting model is built using a pivot table, but when new data is entered, the user sometimes forgets to refresh the pivot table. The modeller builds a button to refresh a pivot table every time it’s pressed.

Dangers and Pitfalls of using Macros

Wonderful as they are, macros should always be used with caution. Quite often formulas or filtering will achieve the desired result, without the need for building a macro. Always try a formula or standard Excel solution first before considering the inclusion of a macro in your model.
- Unless ranges are defined properly using relative cell references, macros don’t move their references the way that formulas do. If any data has moved since the macro was written, the macro will refer to the wrong cell. A poorly written macro can copy and paste over data, and you won’t know what it has done!

- Visual Basic is a complex programming language, a skill that many people don’t have and are therefore not able to edit or follow what a macro is doing - especially if it hasn’t been documented properly.

- Macros are time consuming to write, and often a much quicker and more transparent solution can be built just as easily with a plain formula.

- Running macros requires security settings to be changed (as shown above) and if your model is used by other people they may not realise they need to change their settings causing your model not to work properly.

- In Excel 2007/2010 and Excel for Mac 2011, macro enable files need to be saved in .xlsm file format. If they are accidentally saved as an .xlsx macro-free workbook, the macros will be removed from the file, and they won’t run. If this happens, the macros have been deleted and you’ll need to either rewrite the code, or copy it back in from a backup copy!

- Unless you will be maintaining the model yourself, most Excel users in business have more than enough to do without also trying to learn how to program.

- If you’re the one running the macro each time the model needs to be updated, that’s fine, but what will happen after you’re gone? If the process stops working due to a macro problem, it can hold up monthly procedure significantly.

Basically, there are more things that can go wrong if a financial model contains macros. When it comes to building financial models, consider whether a VBA solution is absolutely necessary before including it in your model. Macros should not be used unless there is no other option!
CHAPTER 9 Common uses of tools in Financial Modelling

Escalation methods for Modelling

There are several different ways of applying a growth, indexation or escalation method over a period of time to a principal amount. We often need to be able to include escalation in our models for the purpose of forecasting sales growth year on year for example, or increasing costs by an inflation amount. Calculating a growth amount might seem straightforward initially, there are actually several different methods that can be used and when applying growth rates in a model, you need to be clear about which method you are using.

Whilst we can use functions to calculate future values based on set or varying growth rates, we’ll be firstly calculating them manually so that we understand the mechanics of the calculations, and then use the functions.

Using Absolute (Fixed) Growth Rate

Here we have compound growth over five years at a fixed interest rate.

The capital grows each year by 5%. At the simplest level our formula is $PV(1+R)$ where PV is Present Value and R is the rate. The reason we are adding 1 (effectively 100%) to the growth rate (5%) is because we want the capital sum returned along with the accrued interest. Multiplying an amount by (1+growth) is a very common tool in financial modelling.

![Fig 8.35 Fixed, compounding exercise](image)

Exercise (Option 1a) Fixed & Compounding:

1. Create the exercise as shown above. The rate is fixed in a single cell (B4), therefore we need an absolute reference, namely $B$4. So, after the first year (in 2011) our formula in cell C7 is $=B7*(1+$B$4)$.

2. Copy the formula in C7 to D7 through G7

This method grows the amount at a fixed rate, which compounds each year.
With most models we normally want to show the amount that is forecast in each year, but if you want a shortcut formula that will show, in a single cell, how much we’ll have at a fixed 5% growth after five years, we can use the FV function. Let’s verify the amount you calculated for the year 2015 by using this function.

3. Somewhere in a cell to the right of this table, enter Excel’s FV (Future Value) function. Either type =FV( into the cell and follow the prompts, or


Because there are no periodic payments being made, we need only include the yearly rate, the period in years and the PV (present value).

4. Prefix the function with a minus symbol to get a positive result. Your formula should be: 
\[-FV($B$4,5,0,B7), and the totals using both methods should be the same. Your sheet should look something like this:

![Completed fixed, compounding exercise](image)

**Fig 8.36 Completed fixed, compounding exercise**

**Exercise (Option 1b) Fixed & Non-compounding:**

Another variation on this escalation method is to still use a fixed growth rate, but not compound it. To do this, we will use an escalation index across the top. This may be more practical in some model designs, but bear in mind that it yields different results!
1. In row 10, create your escalation index row using the formula =$B$10+$B$1 in cell C9, and copying across.

2. Escalate in cell C14 by using the formula =$B$14*C10, and copying across.

This formula gives you a lower result in year 5 than Option 1a because it is not compounding on the previous year. It simply adds a flat $5,015 x 5\%$ each year, instead of adding the growth rate to the increased total.

**Using Relative (Varying) Growth Rates**

The next example returns the future value of an initial principal after applying a series of interest rates. While the rate was fixed in the example above, here it fluctuates from year to year. This is considered to be a superior model design, as it allows for fluctuation of growth rates. Even if the growth rate does not change from year to year, by building the model in this way, we are allowing for changes in the future. By using the above design with a single growth rate, we are restricted to only having a rate that does not change.

We’ll no longer use absolute references for the interest rate, but will allow both it and the value to change from year to year as we copy the formula across D22 through to G22. Cell C22’s formula is thus again $PV \times (1+R)$ as previously, but this time $R$ is not referenced absolutely =B22*(1+B19)
Exercise (Option 2a) Varying rates & Compounding:

1. Complete Cell C22 in the exercise workbook to achieve the result above, then copy the result to the right.

Let’s try a shortcut function to get directly to the results without going through the annual calculations.

[Excel 07/10 and Excel for Mac 2011] To the right of the results, enter Excel’s FVSCHEDULE function to verify the figure for 2013.

Use the function FVSCHEDULE in your exercise spreadsheet

[Excel 2003] Note that the FVSCHEDULE function is not available in Excel 2003.

Your sheet should look something like this:

![Fig 8.39 Completed relative, compounding exercise]

Exercise (Option 2b) Varying Rates & Non-compounding:

In this next example, we are using another escalation index across the top.

![Fig 8.40 Relative, non-compounding exercise]
1. In row 25, create your escalation index row using the formula \( =B25+B11 \) in cell C25, and copying across.

2. Escalate in cell C29 by using the formula \( =$B29*C25 \), and copying across.

This formula gives you a lower result in year 5 than Option 2a because it is not compounding on the previous year.

**Using Exponential Operations on an Absolute (Fixed) Growth Rate**

This exercise is similar to the first one, except that we will include the use of the caret symbol (^) to perform exponential (to the power of) operations. For instance, \( 3 \times 3 \times 3 = 27 \). So does \( 3^3 \) and, in Excel’s equivalent, \( 3^3 \) does so too.

**Exercise (Option 3):**

Below we have compound growth over a period at a fixed 5% p.a. interest rate.

![Fig 8.41 Complex escalation](image)

Our formula will allow for start and end-dates to the period, in the form of:

\[-PV\times(100\%+rate)^{(End \text{ year}-Start \text{ year})}\]

Excel will perform the exponential operation before the multiplication so the encompassing set of brackets (greyed above) is not essential. The Formula in C37 is therefore

\[=B22\times(1+B19)^{(C21-B21)}\]

Note that instead of using “End year-Start year”, you may also insert a “helper row” at the top to link to instead, which makes a slightly less complex formula. The next exercise below shows how this can be used.

1. Complete the formula for cell C22 and copy it across D22 through G22 to achieve the results in the diagram above.

2. The function \[-FV($B$19,G21-$B$21,0,B22)\] will verify the answer for the year 2015.
Practical Usage of Exponential Growth Rates

You will notice that we achieve the same result ($6,401) under option 1 as we did under option 3 – but with a far less complex formula. Using exponential growth rates are useful when you need to compound the growth rate, but want each cell to operate independently of each other.

If for example, you have start dates and end dates which need to be included in a formula as well as compounding escalation, you will need to use exponential growth rates. In the example below, create a formula that will evaluate the start date and end date, only inserting the cost in the relevant years and then compounding the inflation.

This will require the IF and AND functions to be used in combination to establish whether the column heading year falls within an employees start- and end dates.

For instance, in English terms we will be checking that IF(AND(2010 >= Employee Startdate,2010 <= Employee EndDate)) is true before writing inflation escalated costs into the appropriate cell of column F. Otherwise we will write a zero.

Use the term in years (Row 5) to exponentially escalate the costs (in Column C). For instance, if the year 2010 (F6) meets the IF test, F7 will equate to Cost*(1+inflation)^0, which leaves it, in this case, unchanged.

Your formula should be: =IF(AND(F$6<=$E7,F$6>=$D7),$C7*(1+inflation)^F$5,0)

And the sheet should look something like this:
Calculating Cumulative Totals

As always, we try to follow best practice when building financial models, and one of the important points of best practice is to ensure that formulas are as consistent as possible. Calculating cumulative totals are a common calculation in financial modelling, and this is one such situation where ensuring consistent formulas can be tricky.

In the example below, we need to calculate the cumulative number of customers in order to calculate the revenue.

Below is one solution to the problem. It will achieve a correct total, but it necessitates using different formulas in adjacent cells.

Sometimes this is the only solution (without creating an overly complicated formula), and if necessary, you may simply need to resign yourself to the fact that consistent formulas are not possible. There is, however, a simple solution:
Using the formula \( =\text{SUM}(\$B\$3:B3) \) will achieve the same result, with consistent formulas as shown below:

![Fig 8.50 Cumulative totals using consistent formula](image)

*Fig 8.50 Cumulative totals using consistent formula*
Weighted Average Cost of Capital (WACC)

Before investing in a project or new venture, we need to evaluate whether the expected returns justify the risks. Many financial models are built for the purpose of new project evaluation, and the most commonly used tools we use for evaluating the expected returns are NPV (net present value), IRR (internal rate of return) and payback period. How to calculate these measures is discussed in detail in the previous section on “Project Evaluation Functions”.

However, in order to calculate the NPV, we need to know our cost of capital, or our required rate of return for the project. You may also hear the cost of capital referred to as the discount rate or hurdle rate. The cost of capital is basically the “opportunity cost” of the funds invested, or in other words, the rate of return that the investor would expect to receive if they invested the money somewhere else rather than this particular project. In effect, by investing in the business the investor is willing to forego the returns from other avenues. Therefore, cost of capital is the minimum required return for a project - the returns need to be greater than the cost of capital for the project to be accepted.

The importance of WACC is that it becomes a baseline to determine suitable investments for success of the business. WACC is typically represented as a percent, so any investment decisions taken by the company must aim to deliver returns greater than the WACC to make them worthwhile. For example, if the WACC of a company is 15%, then most of the investments that the company makes must be with the goal of generating returns greater than 15%. WACC is not just calculated to the entire company or business. It can also be calculated for individual projects to determine if it is worthwhile for the company to pursue the project.

Company capital is sourced from many avenues and is targeted for various areas of business growth and sustenance. Typically, the two major sources of capital are debt and equity. Both these tools are distinctly different, have different costs, and hence need to be considered differently in our calculations. As a result, not all capital carries the same weight which is why we sometimes need to calculate the Weighted Average Cost of Capital (WACC) in order to work out how much our capital is worth. Sometimes when calculating the NPV of a project, a modeller may simply use a nominated cost of cost capital (we used a rate of 12% in the previous section on “Net Present Value (NPV)”), but to evaluate a project more accurately, we can calculate the WACC which will give us the exact cost of capital for that company. Note that the WACC will be completely different for each company, depending on their individual mix of equity and debt.

How to Calculate the WACC

The Weighted Average Cost of Capital or WACC is the average returns on capital weighted proportionally based on its category. Typically the company sources its capital from stocks (common and preferred), bonds as well as long term debts. Each category of capital is
invested in different avenues to help the business sustain and grow. WACC represents the average cost of capital with proportional contributions from the various sources.

WACC can be calculated using the given formula.

\[
WACC = \left( \frac{\text{Debt}}{\text{Debt} + \text{Equity}} \times \text{Cost of Debt} \right) \times (1 - \text{Tax Rate}) + \left( \frac{\text{Equity}}{\text{Debt} + \text{Equity}} \times \text{Cost of Equity} \right)
\]

Where,

- **Debt**: Total capital raised from debts.
- **Equity**: Total Capital raised from equity. Note that if the company has common and preferred shares, then the two need to be weighted and factored accordingly.
- **Cost of Debt**: This is the cost of raising the debt. If taken as bank loans, then it is the interest that needs to be paid to the banks.
- **Cost of Equity**: This is the return that the shareholder or investor expects from their holding in the company. Common ways of calculating this are the Capital Asset Pricing Model (CAPM) or Gordon’s Growth Model.
- **Tax Rate**: This is the prevalent corporate tax rates.

**Exercise: Calculating the WACC in Excel**

While the formula looks very complex, once the concepts are clear, WACC can be calculated relatively easily in Excel. Let’s do an exercise where we need to manually calculate the WACC.

Let’s say you have the following inputs:

- Debt is $6,023,000 for which you are paying 8.5%
- Equity is $4,421,000 for which you expect a return of 13%
- Tax is 30%
1. Set up your sheet as shown above.
2. In row 4, calculate the cost of debt and equity before tax.
3. Your formula in cell B4 should be \( =B2\times B3 \). Copy it across to cell C4.
4. In row 6, remove the tax from the debt amount.
5. Your formula in cell B6 should be \( =B4\times(1-B5) \). Copy it across to cell C6. Although there is no tax applicable to the equity amount, we’ll still copy the formula across to maintain consistency of formulas (hence following best practice).
6. Add together the value of debt + equity in cell D2, and cost after tax in cell D6.
7. Now calculate the WACC which will be the cost after tax of debt and equity as a proportion of the total.
8. Your formula in cell D7 should be \( =D6/D2 \), and your worksheet should look something like this:

Hence the weighted average cost of capital for this company is 9.78%. This means that when this company is evaluating new opportunities, the minimum required rate of return would be 9.78%, as this is what their capital is currently costing them. When calculating the NPV for a new project, this would be the input for the discount rate in the NPV function. Theoretically, the NPV at this discount rate must be greater than zero, and the IRR of a
series of cashflows needs to be higher than this amount in order for a project to be accepted.

**Building a Tiering Table**

Calculating an amount based on tiering tables is a pretty complex formula, and is commonly used in pricing models. There are two types of tiering table (also called volume breaks, block or stepped pricing). The simplest version calculates the entire amount at the tier. The more complex table, like the tax tiers, is progressive such that the first number of units at the first tier and the next amount at the next price.

For example, in a tax table, an income of $8,000 would be taxed at zero for the first $6,000 and then the next $2,000 is taxed at 15%, so you would only pay $2,000 x 15% = $300, not 15% on the whole amount.

**Flat Tiering Structure**

Let’s say for example, you have the below pricing structure. If the customer purchases five items it’s $15 each, but if the order is between six and 50 items, they are priced at $12 each, and so on.

We’d like to populate the table below to calculate how much a customer would be charged at different volumes.

![Fig 9.24 Flat tiered pricing calculation](image)

A horribly long and complicated IF statement will do the trick:

=IF(A3<F3,G3,IF(A3<F4,G4,IF(A3<F5,G5,IF(A3<F6,G6,G7))))

But there is a much simpler way, using a VLOOKUP or LOOKUP.
1. In cell B3, calculate the per unit price using a close-match VLOOKUP. Your formula should be: \( =\text{VLOOKUP}(A3,\$E\$3:G\$7,3) \) Note that the last field is omitted, which means it will return a close match not an exact match. You may also use the formula \( =\text{VLOOKUP}(A3,\$E\$3:G\$7,3,\text{TRUE}) \) which will give the same result.

2. Multiply the unit price by the number of units and add them up. Your total should be $7,564, and your sheet should look something like this:

![Completed flat tiered pricing calculation](image)

**Fig 9.25 Completed flat tiered pricing calculation**

**Hot tip:** Because we need a close match, not an exact match, we could also have use the LOOKUP function instead. This had the added advantage of not having to have the criteria to the left of the result – LOOKUPS are backwards compatible. You also do not have to worry about someone inserting or deleting columns in the source data range and messing up the formula. For more detail on how to use LOOKUPS and VLOOKUPS, see the section on “Lookup Functions”.

**Progressive Tiering Structure**

If the tiering calculation is not as straightforward as the one below, we may need to calculate under a progressive tiered structure. A good example of this is the way that the tax calculations work. This formula calculation is not for the faint-hearted, so unless you specifically need to calculate a progressive tiering table, feel free to skip this section!
The first person’s salary is $123,120, which means that they are in the $80k - $180k tax bracket. However, we can’t just do a flat tiering calculation on this because the first $6k is tax free, the next $40k is only at 15%, etc. The formula in column I takes care of the calculations below the threshold. Note that the tiers begin with one cent, eg. $6,000.01

You’ll need to split your calculation into several parts, as shown in the diagram above. Firstly work out the lowest tier, and then calculate the difference between the salary and the tier (in this case $80,000). Then the remainder should be calculated at the marginal tax rate.

**Hint:** this calculation will be easier if you calculate these in separate cells, and then combine them at the end – just like we have done with previous examples of nested functions, such as INDEX / MATCH or VLOOKUP/COLUMN.
1. Start by using a close match VLOOKUP to pick up column I (the amount payable up to the threshold). Your formula should be: \( \text{=VLOOKUP(C2,$F$2:$I$6,4)} \)

2. Next, we need to work out the amount over the total salary is above the lowest tier (i.e. in this example, $123,120 - $80,000 = $43,120) To do this we’ll do another close match VLOOKUP returning the first column and subtract this from the salary amount. In a separate cell, your formula should be \( \text{=C2-VLOOKUP(C2,$F$2:$F$6,1)} \). Your answer should be $43,120.

3. Next, we need to find out how much to multiply the $43,120 by (i.e. the highest tax tier). To do this, create a VLOOKUP returning the third column. Your formula should be: \( \text{VLOOKUP(C2,$F$2:$H$6,3)} \). Your answer should be 37%.

4. Now, put all three formulas together like this: \( \text{=VLOOKUP(C2,$F$2:$I$6,4)}+\text{(C2-VLOOKUP(C2,$F$2:$F$6,1))}*\text{VLOOKUP(C2,$F$2:$H$6,3)} \)

The good news is that if you’ve used cell referencing correctly, you can simply copy the formula down!

---

![Fig 9.27 Completed progressive tiered calculation](image-url)
CHAPTER 10 Model Review

Rebuilding an Inherited Model

If you are lucky, you will get to build your own model from scratch, but most often you will be handed someone else’s model to validate, modify, and make your own. One of the toughest tasks for any financial modeller is trying to use, understand, audit or validate an existing model. While it may seem like a simple enough task as the hard work is already done, the real challenge will be to understand how the model was designed. Excel is such a flexible tool and there are so many different ways to achieve the same end that trying to understand the way a model has been built is like trying to get inside another modeller’s mind! Give three different modellers a problem, and you’ll undoubtedly have three different model designs as each modeller will approach the model design in a different way and almost certainly use different formulas.

Obviously the best place to start when trying to understand a model is to talk to the original designer if that is at all possible. There is no one who understands the model better than the modeller who designed it in the first place. However, the original modeller is often unavailable and we therefore need to allow the model to speak for itself.

If financial modelling best practice has been used, understanding a financial model in Excel should be a fairly straightforward task, however, many inherited models have not been built by professional modellers, but rather cobbled together by different areas of the business to achieve a (usually short-term) outcome. These kinds of inherited models can be a real nightmare to unravel, and often it’s simply easier to start from scratch! If this is not practical, there are some tricks and techniques in Excel that can help you to take apart someone else’s model. Using these techniques you can make the task of rebuilding an inherited model more achievable. We will explore some of these techniques in this section.

Removing Redundant Assumptions and Source Data in a Model

A common problem modellers encounter when rebuilding an inherited model is redundant assumptions and sources in the data model. This can happen when the modeller has needed to use source information which is already in the spread sheet but was unaware that it already exists, and then creates another set of assumptions or source data. Often as a model evolves, parts of the model calculations get deleted, but the assumptions which fed into those calculation remained which were no longer required.

The easiest way to see if any formulas are linking to an input cell is to use the trace dependents formula auditing tool. This tool will allow you to trace forward and backwards throughout the model. One technique you can use to remove redundant cells is to colour code all of the input assumption and calculation cells. Then go to the model outputs, and
trace back the formulas. As each input assumption is validated, change the colour of the input cells back to their original colour. At the end of the process, if any cells are still coloured, they are redundant to the model and can be deleted.

If you think a section of the model is not being used, before deleting it, check it using the trace dependents formula auditing tool as described below. Be aware that there are a few tools and formulas such as INDIRECT and OFFSET which sometimes can be relying on a cell but which do not show up in the formula auditing tool. For this reason, make sure you save before deleting, and check the output of the model for #REF! or other errors before continuing. If the deletion has caused a problem, you can simply undo or close without saving to revert to the version of the file prior to deletion. It’s also best to delete the entire rows or columns instead of just clearing the cells when removing sections of a model in this way. Deleting the cell will show an error and flag the issue if it had been linking to a formula, whereas clearing cells will simply change the value to zero which will not necessarily return an error and this can cause undetected errors to remain in the model.

**Formula Auditing**

One of the most effective ways to start understanding a financial model (especially one that you did not build yourself!) is to look at the formulas and identify the predecessors and dependents. This will give you an idea of the logic used by the previous builder of the model. Usually the best place to start in this technique is the final output. Take a look at each output and trace back the predecessors until you eventually find out what calculations, source data and assumptions have been used by the original designer.

If you have a formula and you want to know which cells are used in the formula, the easiest way is to use **Formula Auditing**. You can access the Formula Auditing features in excel from the Formula Auditing ribbon under the Formulas tab as shown below.

![Fig 10.1 Formula auditing icons in the Ribbon](image)

**Tracing Precedents**

[Excel 2007/10 and Excel for Mac 2011] To trace the precedents, select the cell you want to audit and click on “Trace Precedents” in the Formula Auditing ribbon.
To trace precedents in Excel 2003, go to Tools - Formula Auditing - Trace precedents. You can always add Formula Auditing to your toolbar, so you can access it more easily.

The Trace Precedents feature will allow you to see all the other cells whose values were used in the formula used in the present cell. It is possible to understand the calculation just by observing the cell formula too, but the arrows make it easier to locate the cells which reduce the chances of errors in auditing the model.
Fig 10.4 Using trace precendents on a formula

**NOTE:** Double clicking on the blue arrows will take you to the preceding cells. If you double click again, it will bring you back to the original cell. Also note that if the preceding cells are on another page, it will show the dotted line. To go to that cell, simply double-click on the dotted line, and it will bring up a list of all the off-sheet references of that cell. Double-click on the reference to go to that cell.

**Tracing Dependents**

Just like precedents, you can also trace the dependents to a particular cell. These are all the cells that use the value in the current cell for the calculations.

[Excel 2007/10 and Excel for Mac 2011] Select the the “Trace Dependents” button in the Formula Auditing ribbon under the Formulas tab.

This is particularly useful if you are considering deleting a cell and don’t want to mess up the model!

[Excel 2003] Similarly, to trace precedents in Excel 2003, go to Tools - Formula Auditing - Trace Dependents. You can always add Formula Auditing to your toolbar, so you can access it more easily.

**Hot Tip:** Click on the “Trace Precedents” button again to trace the precedents of the precedent cells. This is invaluable in auditing complex formulas.

While arrows are great visual aids to understand the formula, sometimes they can become a distraction, especially, if there are too many of them. You can clear all the arrows by clicking
on Remove Arrows button. This is also important if you plan to print the spreadsheet. If you don’t remove the arrows before printing, they will show up on the printed document.

![Fig 10.6 Removing all arrows](image)

**Note:** There are some formulas upon which the Formula Auditor will not work very well, such as array formulas, data tables and the INDIRECT formula. These are visible using the “Dependency Auditor” tool which can be purchased separately from Microsoft.

**Hot Tip:** You can also jump to precedent cells by using these shortcuts:

**[Excel 2003/7/10]**
- CTRL + [ (open square bracket) to jump to and highlight precedent cells
- CTRL + ] (close square bracket) to jump to and highlight dependent cells
- CTRL + ’ (apostrophe usually located below the “, not to be confused with ‘) to edit cell (same as F2)

**[Excel for Mac 2011]**
- COMMAND + [ (open square bracket) to jump to and highlight precedent cells
- COMMAND + ] (close square bracket) to jump to and highlight dependent cells
- COMMAND + U to edit cell (same as F2)

**Error Checking Tool**

There are times when we unintentionally introduce errors in the financial model. This could be because we did not comprehend the logic completely or because we mistyped the formula. Excel provides an error checking function that helps identify the problem with the
function. This is extremely useful when you incorrect formula is also syntactically correct. Excel indicates the anomalies in the model with a green triangle in the top left corner of the cell.

![Fig 10.7 Error checking tools showing error in SUM formula](image)

To understand the nature of the error, select the cell and click on the drop down menu. There are many categories of such errors, but inconsistent formula is one of the most common errors, or omitting adjacent cells as shown above.

**[Excel 2003]** The error checking tool can be found in Excel 2003, by going to Tools > Formula Auditing > Error Checking. If you have the Formula Auditing toolbar displayed, you can click on the “Error Checking” Button directly too.

You can change the kind of background error checking you want to perform on your model by selecting the “Error Checking Options” as shown in Figure 10.7. It will display the screen shown below where you can control the kind of checks you want performed.
Fig 10.8 Editing error checking options

You can also access the Excel Options dialog from selecting File (or Microsoft Office Button in Excel 2007), then Options. In Excel for Mac 2011, access Excel Options by clicking Excel, then Preferences, then Error Checking.

You can review all the errors in the financial model using the Error Checking feature in Excel.

Fig 10.9 Error checking feature

The Error checking offers two options
• **Error Checking:** Used to check formulas in error.
• **Trace Error:** This is used to trace the source of the error in the final result. To use this feature, select the cell with the error (usually shown as #ERR?, #NAME?, etc) and click on Error Checking - Trace Error.

![Fig 10.10 Error checking options](image)

Excel will trace back the source of the error with a red arrow as shown below.

![Fig 10.11 Tracing the error source](image)

[Excel 2003]: In Excel 2003 the same can be done from Tools - Formula Auditing - Trace Errors

**Evaluating Formula Tool**

Sometimes it is difficult to decipher a complex nested formula just by looking at precedents and dependent relations. It is easier to actually evaluate it step by step to understand how it works. Excel facilitates this by the use of Evaluate Formula feature.

To evaluate a formula in any cell, select the cell and click on Evaluate Formula in the Formula Auditing ribbon under the Formula tab.
This will display an Evaluate Formula dialog that will show you the complete formula in the cell. Keep clicking on the “Evaluate” button repeatedly to see how Excel calculates the value step-by-step.

**Fig 10.12 Evaluate formula in the Ribbon**

![Evaluate Formula dialog box](image)

**Fig 10.13 Evaluate formula dialog box**

[Excel 2003]: In Excel 2003 the same can be done from Tools > Formula Auditing > Evaluate Formula

Once you have looked up the source of the parameter, you can Step Out to continue the evaluation.

Note: The Evaluate Formula auditing option is currently unavailable for Excel for Mac 2011.

**Hot Tip**: You can also evaluate a partial formula manually by selecting each expression and pressing F9. You must select an entire section of the formula that makes sense in its own right; for example, in the horribly long and complex formula shown below, we can select this portion of the formula: \text{SUM(B$2:B2)}
By then pressing F9, the formula will display the result of that expression, which in this case is 250.

If you then hit the enter key, it will paste that value into the formula (usually not what we want to do!) but if you hit escape, it will return to normal. This is a handy little trick that is sometimes easier and less complicated than using the Evaluate Formula tool.

For more detail on why you would want to break up a formula in this way, see the section on “How long should a Formula be?”

Note: The Evaluate Formula auditing option and the above shortcut are currently unavailable for Excel for Mac 2011.

Viewing all the Formulas

If you want to just look at all the formulas and not the evaluated value, you can use the Show Formulas option in Excel. This is also a very effective tool to understand the general logic in the model and can sometimes make anomalies more evident.
This can also be done using the CTRL + ` shortcut key. Note that ` which is usually located below the ~ key at the top left of the keyboard should not be confused with the CTRL + ‘ shortcut used to edit cells. ❌ No keyboard shortcut exists at this time for Excel for Mac 2011.

![Fig 10.16 Show formulas option in the Ribbon](image)

[Excel 2003]: In Excel 2003 the same can be done from Tools - Formula Auditing - Show Formulas

**Direct Editing**

By default, Excel allows users to edit the formulas directly in the cell, so that when you double-click on a cell it allows direct editing. However, while auditing any inherited spreadsheet, this may not be a good idea, as we could inadvertently mess up the formulas that are already in the cell. Excel allows users to disable direct editing, thereby protecting the cells to a degree, but still allowing users to view the formula. It is quite different from protection, which does not let you do anything at all on the cell, and it’s much easier to turn off and on than protection.

Using this tool you can double-click on the cell to trace back and ensure that you understand the assumptions made by the original modeller and take care that you are not making any inadvertent changes to the formulas in the cell. You can still make any changes necessary directly in the formula bar.

You can Disable Direct Editing from File > Options > Advanced > “Allow editing directly in cells” ❌ Note in Excel for Mac 2011 this option is found in Excel > Preferences > Edit and named “Double-Click allows editing directly in cells.”
Now when you double click on the cell, it will highlight all the cells used as parameters in the formula, but not let you edit the cell directly.

![Excel Options]

**Fig 10.17 Disabling direct editing in cells**

**Fig 10.18**

**[Excel 2003]**: To Disable Direct Editing in Excel 2003, Click Tools > Options > Edit. Deselect the “Edit Directly in Cell” option.
CHAPTER 11 Scenarios & Sensitivity Analysis

Overview of Scenario Analysis Tools and Methods

Many people will create a financial model, and save it as the base case. They then change all the numbers, and save it as a worst case. Then they change all the numbers again and save it as a best case. Whilst this method will work, it’s not a very efficient way of performing scenario analysis. If a subsequent change needs to be made to the model, it will need to be made several times, and there a high possibility that an error will be made between the different versions.

Let’s take a look at the technical methods available in Excel for creating scenarios and sensitivities, which will provide a much better way of building models. Essentially, there are three ways:

1. **Manual drop-downs**
   a. In-cell drop-downs (using data validation)
   b. Object drop-downs (using combo boxes)

2. **Scenario Manager**

3. **Data Tables**

Manual drop-downs

Creating scenario analysis using manual drop-down boxes means that you can only view one scenario at a time.

**In-cell drop-downs (using data validation)**

![Fig 11.1 Data validation drop-down box](image)

The value sits within the cell of data validation drop-down box. In this case, the modeller has limited the valid entries to this cell to one of three options; either “Best Case”, “Base Case” or “Worst Case”. This is the easiest and most commonly used type of drop-down box.
See the section on “Bullet Proofing your Model” for how to create a data validation drop-down box, and the next section on “Comparing Scenario Methods” for a practical exercise on how to use them in scenario analysis.

Object drop-down Boxes (using combo boxes)

There is very little difference from the user perspective with a combo box drop-down, but it’s built very differently. This type of box is an object which sits on top of the sheet, rather than within the cell itself. If you look very closely, you can see that the box is sitting across several cells instead of within a single cell. This type of box is a little more difficult to build, but is easier for the user, as the drop-down arrow appears all the time, rather than only when the cell is selected.

See the section on “Form Controls” for how to create a combo box drop-down, and the next section on “Comparing Scenario Methods” for a practical exercise on how to use them in scenario analysis.

Scenario Manager

The scenario manager is an Excel tool in which you can create multiple scenarios. Once it is set up, the user can select a scenario, and input cells will change automatically. Scenario manager is fairly limited and is not a particularly helpful in large and complex models. It is therefore not a very widely used tool for scenarios in Financial Modelling.
Data Tables

Data tables are not as commonly used as drop-down scenarios, mostly because of a lack of knowledge on how to use them. Because a data table uses array formulas, it is unlike most other formulas in that you cannot trace dependents, and it is therefore very difficult to follow unless you are already familiar with data tables.

However, Data tables are an extremely useful and powerful scenario and sensitivity analysis tool, as, unlike drop-down boxes, they show the results of multiple scenarios at a time.

See the next section on “Data Tables” for how to create a data table, and the next section on “Comparing Scenario Methods” for a practical exercise on how to use them in scenario analysis.
Scenario Manager

Scenario Manager is a tool which, in the later versions of Excel, is grouped together with Goal Seek and Data Tables in the What-if Analysis section of the Data tab. Being grouped with other tools which are so useful would lead the aspiring modeller to believe that Scenario Manager is also a critical tool to know, however, despite its useful-sounding name, and the good company it keeps, Scenario Manager is quite limited in its functionality and is not particularly helpful in large and complex financial models. It is therefore is not commonly used by expert financial modellers, however, for completeness, we will cover it very briefly!

Let’s take a very simple example of a person creating their personal budget for next year, but who does not know what their mortgage payments will be due to the changing rate of interest, or how much their rail card will cost.

![Fig 11.4 Scenario manager example](image)

You can define different scenarios and then switch between them to do ‘what-if’ analysis to see if you will end up in debt or be able to afford a holiday. Scenarios work best on complex spreadsheets where there is a large knock-on effect from changes in the variable data.

Scenarios are created and managed using the Scenario Manager.

1. Create and format the spreadsheet. You can leave the blue variable cells empty

2. **[Excel 2007/10 and Excel for Mac 2011]** On the Data tab, in the Data Tools section, click on the What-if Analysis icon, and select Scenario Manager from the drop-
Go to Tools - Scenarios

3. This will bring up the Scenario Manager dialog box
4. Click on the “Add” button, to create a new scenario
5. This will bring up the Add Scenario dialog box
6. Enter a name for the first scenario into the Scenario Name box, i.e. Best Case
7. Enter the cell references for the variable cells into the Changing Cells box. Use absolute references, separate each reference with a comma (if there is more than one), but don’t use spaces. You can actually hold down the [Ctrl] key and click on each cell in the spreadsheet to insert the references into the box. Click “OK”
8. This will bring up the Scenario Values dialog box.
9. Enter the variables values for this scenario (i.e. Base Case) eg. $15,000 for mortgage and $2,000 for travel. Click “OK”
10. This will take you back to the Scenario Manager dialog box.
11. Follow the previous steps again to create each scenario - i.e. Worst Case ($20,000 for mortgage and $3,000 for travel) and Best Case ($10,000 for mortgage and $1,500 for travel.
12. When you have created all the scenarios you can use the Scenario Manager to view each scenario.
13. Scenarios are sheet-specific, i.e. they only exist in the sheet you created them in.

As we can see, the Scenario Manager is quite a rudimentary tool, which simply changes hard-coded numbers. It’s not very easy to see, display or print the different options unless we go into the Scenario Manager tool. Using the Summary Tool creates a summary report of the scenarios created, but they are not dynamic nor interactive and therefore of little use as a modelling tool. The biggest downfall of Scenario Manager is that the user cannot see them on the sheet unless they actually go in to view the scenarios.

Using Data Tables for Sensitivity Analysis

Data tables are one of the more advanced and complex financial modelling tools. They can be used for scenarios and sensitivity analysis, but they are not as commonly used as drop-down scenarios often because of a lack of knowledge on how to use them. Because Data tables use array formulas, they are unlike most other formulas in that you cannot trace dependents, and they are very difficult to follow unless you are familiar with them.

Note that a modeller who does not know data tables will be unable to edit the table, or make any changes.
Let’s create an interest rate calculator upon which we can test the sensitivity of monthly repayments to changes in interest rates and loan terms.

5. Firstly, set up the model like this with the hard-coded input assumptions.

![Interest Rate Calculator](image1)

Fig 11.5 Loan calculation inputs

6. In cell B12, use a PMT formula to calculate the monthly repayments. See the section on “Loan Calculations” for more information on how to use this function.

7. Using the wizard, it should look like this:

![PMT Function Wizard](image2)

Fig 11.6 PMT function wizard

8. Your formula should be: \( \text{PMT}(B8/12,12*B10,B6) \)

The function returns a negative value because this is an expense. For our purposes, change it to a positive by preceding the function with the minus symbol.
One-variable Data Table

The Data Table presents a body of data derived from a function. The rows and columns of the table are drawn from one or two of the arguments or variables fed to the function. In this case the data comes from the results of the PMT function, using the 'Interest Rate' variable (cell B8).

9. We will decide what the column elements will be by entering them in cells E7 to E12. For this exercise use 6.00% to 8.50% in increments of half a percent as shown:

<table>
<thead>
<tr>
<th></th>
<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>1</td>
<td>INTEREST RATE CALCULATOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Monthly Repayment under different scenarios</td>
<td>$2,586</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Loan Amount</td>
<td>$500,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Interest Rate</td>
<td>7.50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Term of Loan (Years)</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Monthly Repayment</td>
<td>$2,586</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Fig 11.7 One-variable data table inputs](image)

10. Merge cells D7 to D12, and change the orientation under the alignment tab under Format Cells if you wish to have the Interest Rate title orientated vertically as shown above.

11. Enter in cell F6 the formula =B12 which is the cell containing the PMT function. The table, when created, will use the PMT function to populate the table according to the values in input column (this will become clearer once the table is populated).

12. Highlight cells E6:F12. You must highlight all the cells for it to work.

[Excel 07/10] and [Excel for Mac 2011] Select the What-If Analysis from the Data tab. Choose Data Table from the options which appear.

[Excel 2003] Select Table from the Data menu.

13. The Data Table dialog box will appear. Because we are only doing a one-variable data table, we only need to enter data for one of the Interest_Rate or Term variables, but which one depends on whether our input variable is arranged in a row or a column. Because it is in a column, we should use the Column input cell field. Link this field to the input field for the interest rate (cell B8).

14. Your dialog box should look like this:
15. Click OK, and your data table will populate. Note that the formula in the cells will have curly brackets around them, denoting that it is an array formula. \{=\text{TABLE}(B8)\} You will not be able to edit the cells.

16. Your sheet should look something like this:

Two-variable Data Table

17. Now, let’s change this to a two-variable data table. Clear cells F7:F12. You will need to highlight and clear them all at once, as you cannot change or delete part of a data table.

18. With a two-variable table, the output cell needs to be at the top-left of the table, at the intersection of the row and column variables. Therefore cut and paste cell F6 to cell E6.

19. In cells F6:I6, enter the number of years you wish to test in your data table. For this exercise enter the values 20, 25, 30 and 35 across the table. Change the formatting as necessary. We now have the makings of a table with the Term across the top row and the Interest Rate down the left-hand column as shown below:
20. Now highlight the Table area cells E6:I12. You must highlight all the cells for it to work.

[Excel 07/10 and Excel for Mac 2011] Select Data Table from the drop-down list under the What-if Analysis icon in the Data Tools section on the Data tab.

[Excel 2003] Select Table from the Data menu.

21. Your row input cell will be the entry field for the values shown in the row (the Term in Years), and the column input cell will the entry field for the values shown in the column (the Interest Rate). Your dialog box should look like this:

![Data Table dialog box](image)

22. Your table should now look like this:

![Completed two-variable data table](image)
23. You might wish to change the font in cell E6 to white; while it is required for the data table to work, it does not add any value visually (and is simply confusing).

**Hot Tip:** Data tables look great when you apply colour scales as shown below as you can visually see the incremental change to the data. For more information on how to use colour scales in [Excel 07/10 and Excel for Mac 2011], see the section on “Conditional Formatting”.

![Fig 11.13 Data table with colour scales](image)

**Limitations of Data Tables**

We can see from the example above that data tables are a great way to look at multiple scenarios or sensitivity analysis at a time. Instead of manually changing the interest rate, or the term of the loan, we can display at a glance the impact of these changes.

However, data tables have a couple of limitations which means that they are not appropriate in every scenario or sensitivity analysis situation.

1. The inputs and outputs need to be on the same page
2. You have a limitation of showing only two inputs, and one output at a time. This is not a restriction with other forms of scenario analysis.
3. Formula auditing (trace precedents and trace dependents) don’t work very well in data tables.

Data tables are extremely useful when, as in the example shown above, you want to see the incremental change of one or two inputs on a single output. For example, how much does my profit margin change by if I change my price from $450 to $460, $470, $480, $490 and $500? A data table would not be an appropriate solution if you the output of your financial model were a full set of financial statements for example! In this situation, a drop-down scenario would be most appropriate.
Advanced Conditional Formatting

A data table is a good opportunity to use conditional formatting with a formula. We can use advanced conditional formatting to enhance a two-variable Data Table, making it more visually interesting and interactive for the user.

Our final result will look similar to the below, using the two variables ‘Term of Loan’ and ‘Interest Rate’ to drive the conditional formatting. If you changed the interest rate in cell B8 from 7.5% to 7%, the row currently being highlighted will change from row 10 to row 9. Similarly, if the term of the loan was changed in cell B10 from 25 to 30, the column currently being highlighted will change from column G to column H. In this way, we can see at a glance which inputs have been selected currently in the model.

For a review of the basics, see the section on “Conditional Formatting”. As you know, Conditional Formatting is a tool that allows you to apply formats to a cell or range of cells, and have that formatting change depending on the value of the cell (or the value of a formula).

In this particular example we want the Interest Rate and Term to intersect at the appropriate repayment instalment. As the variables to the PMT function are changed, so the intersection point will move to spotlight the repayment amount.

One way we can accomplish this is to apply conditional formatting by means of a formula to each row within the table and, likewise, to each column. For instance, in the example below, we can stipulate that if cell E10 is equal to the Interest Rate (B8), then Excel must colour the row 10 within the table.

**Fig 11.14 Completed data table using advanced conditional formatting**
Fig 11.15 Highlighting selected interest scenario using conditional formatting

This would require ten formulas if we were to cover each of the six interest rates and each of the four Terms. However we can accomplish the exercise with two formulas only, by locking either the rows or columns on the one hand with absolute references, and allowing other elements of the cell address to ‘float’ across the row or up and down the column by using relative references on the other. The exercise below illustrates how this is done.

7. [Excel 07/10] Select the entire table (cells E6 to I12.) Choose ‘Conditional Formatting’ on the Styles section (Format for Excel for Mac 2011) of the Home tab. Select “New Rule” from the drop-down list, and then “Use a formula to determine which cells to format”

[Excel for Mac 2011] Select the entire table (cells E6 to I12.) Choose ‘Conditional Formatting’ on the Format section of the Home tab. Select “New Rule” from the drop-down list, set Style to “Classic” and then “Use a formula to determine which cells to format”

[Excel 2003] From the Format menu, select Conditional Formatting and choose the Option ‘Formula is’.

8. Choose to use a formula to determine which cells to format. Note that E below is an absolute reference to hold us in column E, but that the row number (presently 6) is free roam down within the column.
9. Format the cell as required using the Format button.

10. [Excel 07/10] Repeat the exercise to format the Terms. In this case the formula will reverse the relative and absolute references like this: E$6, so locking in row 6. The rules should look like this:

![Conditional formatting dialog box in Excel 2007/10](image)

[Excel 07/10] Use the Add>> button of the Conditional Formatting box to add the Terms formula.

[Excel 03] Use the Add>> button of the Conditional Formatting box to add the Terms formula.
Modifying Conditional Formatting

11. Highlight the relevant rules (in this case the entire Data Table)

[Excel 07/10 and Excel for Mac 2011] choose Manage Rules... from the Conditional Formatting menu

[Excel 2003] choose Conditional Formatting from the Format menu.

Comparing Scenario Methods

Let’s create a very simple property development feasibility model to demonstrate the difference between the methods of scenario analysis.

You are considering purchasing an empty block upon which you are considering developing a residential villa complex on the property.

The assumptions are:

1. The land will cost $4.3m to purchase
2. Council contribution fees will be $750,000
3. The properties that we develop will be 5,000 square metres in total
4. Building costs will be $1,500 per square metre
5. We will sell the properties for $3,000 per square metre
6. Sales commission to the estate agent will be 2.5% of the sale price

Creating the Working Model

1. Layout the model, and enter the assumptions as shown below:
These assumptions are all input variables, so format them as inputs with blue font and beige background.

2. Calculate the income from selling the properties in cell C11 by multiplying the sale price per square metre by the total number of square metres. Your formula should be: \(=D8*C8\).

3. Similarly, calculate the building costs in cell C14 by multiplying the building cost per square metre by the total number of square metres. Your formula should be: \(=B8*D8\).

4. Calculate the sales commission by multiplying the income by the sales commission. Your formula should be: \(=C11*E8\).

5. Add up the expenses in cell C17.

6. Calculate the Net Profit in cell C19 by deducting the expenses from the profit.

7. Calculate the Profit Margin in cell C20 by showing the profit as a percentage of the revenue. Your formula in cell C20 should be \(=C19/C11\).

8. Your model should look something like this so far:
We have now completed the workings of the model ready to add some scenarios. Note that this is a very simple one-page model which we will use to demonstrate how to build scenarios. Your model can be a lot more complex than this, containing many more pages and calculations, but the methodology for creating a scenario will be exactly the same.

**Manual Sensitivity Analysis**

The two input variables that we think are the most likely to change are the Building cost and the Sale price per square metre. Note that because of the way we have built this model with these assumptions hard coded and all the calculation linking to these inputs, we can quite easily change these inputs manually and this will affect the output of the model. We can see that if the Building costs were to decrease from $1,500 to $1,200 per square metre, this would increase the profit margin to 24%. Additionally, if the Sale price were to increase from $3,000 to $3,300 per square metre, this would increase the profit margin further to 31%.

You can see that we can also manually change the sales commission and the total number of square metres in the development, although we have determined that these are unlikely to change.

**Creating Built in Scenarios using a Data Validation Drop-down**

Tweaking these input variables manually is a form of sensitivity or what-if analysis, but it's difficult to control and not very auditable. It’s much better practice to build some pre-defined scenarios into the model, and allow the user to switch between the scenarios using a drop-down box.

Let’s build some scenarios using this method.
9. In cells A4 to A6, enter “Best Case”, “Base Case” and “Worst Case”.
10. Enter in hard-coded assumptions as shown below:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Best Case</td>
<td>$2,500</td>
<td>$8,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Base Case</td>
<td>$3,200</td>
<td>$8,500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Worst Case</td>
<td>$4,750</td>
<td>$7,700</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Fig 11.21 Model scenario inputs*

11. At the very top of the model in cell B1, create a data-validation drop-down box that the user can select from to choose the scenario they’d like to see. For instructions on how to do this, see the section on “Data Validation Drop-down Boxes”.
12. The data validation should be linked to the source cells, as shown below:

*Fig 11.22 Data validation drop-down list dialog box*
13. Your sheet should look like this:

![Figure 11.23 Model with completed data validation drop-down box](image)

14. Now we need to change the input assumptions in cells B8 and C8 from hard-coded inputs to formulas that will change depending on the scenario that has been selected from the drop-down box.

15. There are several different functions that we can use to achieve this; either IF Statement, a VLOOKUP or a SUMIF to name a few, will all return a similar result. If you choose a SUMIF, the formula in cell B8 should look like this:

\[ =\text{SUMIF}(\$A4:\$A6,\$B1,\$B4:\$B6) \]

16. If you’ve got the cell referencing correct, you can simply copy the formula across to cell C8 without changing the formula.

17. Change the cell and font colour of cells B8 and C8 to denote that they now contain formulas instead of hard-coded values.

18. Your completed sheet should now look something like this:
Fig 11.24 Completed data validation drop-down box model with scenario formulas

19. Practice changing the scenario drop-down in cell B1, and watch the numbers change! Under the best case, you’ll be making a 29% profit and worst case, you’ll make a loss of 5%.

Creating Scenarios using a Combo Box Drop-down

Now that we’ve got our scenarios working, let’s take a look at we could achieve the same result using a combo box drop-down instead. Note that Combo Boxes require VBA code as previously discussed in the section on “Form Controls”.

Leaving your scenario model we’ve just created intact, make a copy of the sheet by right-hand clicking on the sheet tab at the bottom. Select “Move or Copy” and then select “Create a copy” as shown below and press OK.

Fig 11.25 Sheet copy dialog box
1. Remove the data validation drop-down we have created in cell B1. The easiest way to do this is to copy a blank cell and paste it over cell B1. This will mess up your calculations – don’t worry about this, we’ll fix them later.

2. Now create a combo box drop-down in the same area. For instructions on how to do this, see the section on “Combo Boxes”.

3. Right-hand click on the combo box, and select Format Control. On the Control tab, under Input Range, select the words “Best Case”, “Base Case” and “Worst Case” in cells A4 to A6.

4. Also choose a cell link, which is where you want the output cell from the drop-down box to be. We often choose the cell behind the combo box, in this call cell B1, as this will be hidden behind the drop-down, and unlikely to then be accidentally deleted. You may need to move the combo box out of the way temporarily (select it the combo box whilst holding down the control key to do this).

5. Your format control dialog box should look like this:

![Fig 11.26 Combo box format control dialog box](image)

6. Press ok. Click away from the combo box, and then select it again. Practice changing the options from the drop-down, and you’ll notice the number in cell B1 changing.

![Fig 11.27 Model with combo box drop-down](image)
7. What we need to do now is to change the formula in cells B8 and C8 which are driving the calculations in the model. The SUMIF (or IF, VLOOKUP or whatever you used) function we created earlier is not working because cell B2 no longer contains name of the scenario. This now needs to be a formula that will select the first option if it contains a 1, the second option if it contains a 2, and so on.

8. Again, there are several functions we can use to achieve this, a CHOOSE or an INDEX to name a few. If you use a CHOOSE function, your formula in cell B8 will be: =CHOOSE($B$1,B4,B5,B6). If you use absolute referencing, you can copy this across to cell C8.

9. Practice changing the drop-down box and make sure it works properly. Your model should now work in exactly the same way as it did earlier, except that we are now using a combo box instead of a data validation drop-down.

10. Hold down the control key, and move the combo box so that it covers the cell link output number in cell B1.

11. Your sheet should now look something like this:

![Fig 11.28 Completed combo box model](image)

**What’s the Difference between a Data Validation and Combo Box Drop-down?**

From the user perspective, both the data validation and the combo box drop-down should be almost indistinguishable from each other. They both work in the same way – that the user selects from a drop-down list, and the model will change. However, there are a couple of key differences from the model developer’s perspective:

- The combo box takes longer to build, as it requires inputs and outputs
- As we can see above, the formulas we use to link to it are quite different
- The combo box can easily be embedded in a chart
We can assign macros to a combo box so that unbeknownst to the user a macro is launched as soon as an option is selected from the drop-down.

The source data which appears in the combo box must be orientated vertically, not horizontally. This is an important point when designing the model - if you plan to use a combo box for scenario analysis, make sure that the input assumptions are listed vertically. Sources for data validation drop-down boxes can be orientated either horizontally or vertically.

![Fig 11.29 Combo box drop-down with horizontally orientated source data](image)

As shown above, combo boxes will not work if the source data is oriented horizontally instead of vertically.

**Creating Scenarios using a Two Variable Data Table**

Let’s take a look to see we could achieve the same result on our property development feasibility model using a data table method of scenario analysis instead of a drop-down box.

Leaving your scenario model we’ve just created intact, make another copy of the sheet by right-hand clicking on the sheet tab at the bottom. Select “Move or Copy” and then select “Create a copy” and press OK.

Instead of creating base, best and worst case scenario from which the user needs to select the scenario that they want to display, we are instead going to create a single matrix where the user can see at a glance simultaneously the results of different inputs. We need to change the design of the model from the way that it was just a little bit in order to create this scenario analysis using a data table instead.

1. Change the assumptions in row 8 from formulas back into hard-coded input numbers.
2. Change the formatting of the cell so that we can see they are hard-coded, not formulas. The quickest way to do this is to copy the formatting from another cell by using the Format Painter.
3. Move the assumption titles to row 7, remove the best, base and worst case inputs and the scenario drop-downs in rows 1 to 6, and replace it with a data table layout as shown below:

![Data Table Example](image)

*Fig 11.30 Model layout for data validation scenario method*

4. To create the titles for the data table inputs, use Merge Cells, and then for the vertically orientated title, wrap text, and right-hand click. Select Format Cells and on the alignment tab change the orientation. To show the title on the right-hand side of the cell instead of the centre, choose Right text alignment instead of Centred under the Text Alignment area.

5. Now create a two data table by selecting the input assumption for sale price for row, and building cost for column. For detail on how to create a data table, see the previous section on “Data Tables”.

6. Link cell B2 to profit margin in cell C20, as this is the output we want to show on the data table.

7. Highlight the entire data table, and go to the data table tool under the Data tab, in the What-if Analysis section. In Excel 2003, go to Data – Table.

8. The data table dialog box will look like this:
9. When completed, the data table will look like this:

![Data Table](image)

*Fig 11.31 Creating a two-variable data table*

This gives us exactly the same results as the drop-down box options, but instead of having to flick through the drop-downs, the user can see at a glance simultaneously the results of the different scenarios.

In this situation, the data table is an appropriate option for displaying the sensitivity of our profit margin to changes in inputs because the inputs and outputs are all on one page, and we are only interested in a single output. We do lose some detail by using the data table method, as we cannot see, for example, the building cost amount under each different scenario, only the final outcome.

This feasibility model shows clearly the difference between three of the scenario methods. We have chosen not to use Scenario Manager, because even though it is perfectly possible on this model, it is not a particularly useful tool. In this example, any of the data validation drop-down, combo box drop-down or the data table scenario analysis methods are suitable, however it really depends on the size, layout, and required output of the model as to which method is best to use.
CHAPTER 12 Presenting Model Output

Charting with Two Different Axes and Chart Types

One of the best ways of analysing the output of your model is to plot two different variables on the same chart, so that we can understand trends, and look for correlations in the data. Sometimes the only way to spot a trend on an anomaly in the model is to look at it graphically. For example, how does our overall revenue compare to headcount? Surely as revenue increases so should headcount, and if it doesn’t we need to understand why. What if we compare regions and their number of units sold and profit? Surely the regions with the highest number of units sold should be the most profitable? This may not necessarily be the case. Let’s take a look at some data, and plot the number of units sold and the profits using two different chart types on two axes.

If we have the data as shown below, we can create a simple line chart. The easiest way to do this is to highlight the data in column A, B and D by holding down the control key as shown below.

[Excel 2007/10] Select the 2-D line from the insert tab on the ribbon.
[Excel for Mac 2011] Select the 2-d line from the Chart tab on the ribbon.
[Excel 2003] Select Insert chart from the chart menu, then finish to complete the chart wizard.

![Inserting a line chart](image.png)
Note that you’ll need to hold down the control key whilst selecting non-consecutive ranges.

The chart will look something like this:

![Chart](image)

Fig 12.55

The problem is that the chart looks quite nonsensical because we are plotting two different types of units; Number of Units Sold, and Profit $.

The first thing we need to do, is move the Units Sold to a new Y axis on the right hand side.

1. Select one of the series by clicking on it. Make sure the entire series has been selected.
2. Right-hand click and go to Format Data Series.
3. On the Series Options tab, click on Secondary Axis instead of Primary Axis.

![Format Data Series](image)

Fig 12.56

1. This will create a new axis on the right hand side.

This works in all Excel versions, but in [Excel 2007/10 and Excel for Mac 2011] you can also on the Layout tab (Format tab in Excel for Mac 2011), click on “Format Selection” as shown.
Insert the axis labels to make your chart clearer.

[Excel 2007/10] Choose Primary Vertical Axis Title and Secondary Vertical Axis Title from the Axis Titles on the Layout tab in the Ribbon as shown below.

This will look better as a combination of a column and line graph, so let’s change the Profit series to a column graph.

1. Select the Profit series.
2. Right-hand click and select Change Series Type.
3. Choose a column chart.

Your chart should look like this:
Fig 12.59 Completed chart on two axes and chart types

Insert a title by

[Excel 2007/10] selecting Chart Title from the layout tab.

[Excel 2003] Right-hand click, go to Chart Options, and enter the Title under Chart Title under the Title tab.

Hot Tip: These charts can be a little confusing if you try to put too much in, so keep it simple, by sticking to two series only. Make sure that the axis are formatted correctly if currency symbols are appropriate. Leave the legend in which will make it clearer which section of the chart refers to which series.
Waterfall Charts

Waterfall charts, also called “Bridge” or “Stepped” charts, have become popular in recent years, particularly when displaying the output of a financial model. A waterfall chart displays very effectively the incremental impact of each period of time, or unit. It is a type of bar chart where the value of the second bar generally begins where the first one finished. To illustrate, a waterfall chart might look something like this:

![Completed basic waterfall chart](image)

*Fig 12.68 Completed basic waterfall chart*

This kind of waterfall chart as show above is very simple, and will only take a few minutes to build, as the incremental cash amounts for each month are always positive, and there are no values below the X Axis. We’ll take a look first of all at creating this kind of very simple waterfall chart, and tackle the more complex charts next.

Creating a Basic Waterfall Chart (with only positive values)

Let’s say you have a forecast incoming cash for the next six months.

<table>
<thead>
<tr>
<th>Month</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>$250</td>
</tr>
<tr>
<td>Feb</td>
<td>$200</td>
</tr>
<tr>
<td>Mar</td>
<td>$1,000</td>
</tr>
<tr>
<td>Apr</td>
<td>$120</td>
</tr>
<tr>
<td>May</td>
<td>$650</td>
</tr>
<tr>
<td>Jun</td>
<td>$800</td>
</tr>
</tbody>
</table>
Now, you could forecast it as a cumulative bar chart:

![Cumulative Cash Forecast](image)

*Fig 12.69 Cumulative cash chart*

But we’d like to see the incremental contribution of each month individually which will add a new dimension to the analysis.

Contrary to popular belief, you don’t need fancy software to build a basic waterfall chart. They are really not that difficult to build in Excel, and are achieved by a little trick whereby we create a “floating” or dummy series in a stacked column chart.

1. Open up a blank worksheet and start by entering the dates, and your cash amount in columns A and C as shown, leaving column B blank:

![Basic waterfall chart workings layout](image)

*Fig 12.70 Basic waterfall chart workings layout*

2. In column B, we need to create a dummy series which will be used to push each cash series bar up to where the last one ended. So column B will contain the cumulative Balance from the previous period. You’ll see what I mean in a minute.

3. We need to make cell B3 a cumulative balance, so the formula in cell B3 should be =B2+C2. Copy B3 down to B7. Let’s label it the ‘Base’ column because it’s effectively where the bottom of each column starts. It should now look something like this:
4. We can now create a stacked bar chart from this data. The easiest way to do this is to highlight the whole range A2:C7, and create the chart.

5. **[Excel 07/10]** Select the stacked column from the column section on the Insert tab, (Charts tab in Excel for Mac 2011) and the chart will appear automatically.  
   **[Excel 2003]** Select Chart from the Insert menu. When the chart wizard appears, simply press Finish.

The chart should now look something like this: (although the colours and background might look a little different if you are using Excel 2003 or Excel for Mac).

6. Click on the series legend and delete it.

7. Add the title “Cash Forecast Waterfall”
   **[Excel 07/10 and Excel for Mac 2011]** Click on the graph, select Chart Title from the Layout tab
   **[Excel 2003]** Right hand click and go to Chart Options
8. Click on the bottom series, and change the series fill colour of the bars to white or no fill by right-hand clicking and selecting Format Data Series. You may also like to edit the colour of the chart series that is showing on the chart.

9. You might like to remove the background gridlines by clicking on them and deleting. If you are using Excel 2003, the background of the chart is probably grey, not white, so you can change the background colour by right-hand clicking on the chart, and selecting Format Plot Area.

10. If there is a border around the white series, you can remove this by going into Format Data Series again and selecting the No Border option.

11. Your waterfall chart should look something like this:

```
<table>
<thead>
<tr>
<th>Month</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>$250</td>
</tr>
<tr>
<td>Feb</td>
<td>$200</td>
</tr>
<tr>
<td>Mar</td>
<td>$1,000</td>
</tr>
<tr>
<td>Apr</td>
<td>($500)</td>
</tr>
<tr>
<td>May</td>
<td>$650</td>
</tr>
<tr>
<td>Jun</td>
<td>$800</td>
</tr>
</tbody>
</table>
```

Creating a Waterfall Chart (with Positive and Negative Values)

What if your cash forecast changed, however, and in April you are expecting a poor month, with a loss amount of $500?
We'll need to create separate columns for additional series in order to allow for negative values in our waterfall chart.

1. Change the layout of your sheet, so that it shows the negative and positive amounts in separate columns as shown below. Remove the dummy figures from column B. Even though the value in cell D5 is a negative amount, we will enter in it as positive for the chart.

2. Maintain a Balance in row E. Cell E2 would be (=E1+C2-D2). Copy this formula down to E7 to achieve the following results:
3. Now we’ll use the dummy column B, as we did before, to control the stepping stones of our chart. The formula in cell B2 should be =E1-D2. Copy it down the range.

4. Generate a chart, as you did in the exercise above, for this data, excluding column E.
5. Again, change the Base series to white font or no fill.
6. You may wish to show the Negative cash amount in a different colour. If you’d like to see all series the same colour, simply change the colours of both to the same colour.
7. Practice changing the input numbers from positive to negative, and check that the chart is working correctly.
8. The final chart should look something like this:

![Cash Forecast Waterfall Chart](image)

*Fig 12.79 Completed waterfall chart (with positive and negative values)*

**Profitability Waterfall Chart**

The above waterfall chart is also useful for displaying the profitability of the company by category, instead of by time series.

In the example below we have simply replaced the months with the revenue and expense categories, and changed the numbers. Using exactly the same simple template, we can display how much of the revenue is being spent in each expense category, and how much residual profit is left.
Creating a Complex Waterfall Chart (with Positive and Negative Values Crossing the X Axis)

You may have noticed that the charts we created above all remain as positive total values. They will not work if the columns are below the X Axis. For example, if the cash loss amount as shown in one of the last example was $2,000 instead of $500, this template would not work correctly:

![Incorrect waterfall chart](image)

In order to create a waterfall chart that will handle both positive and negative values, as well as a balance which crosses the X Axis, we'll need to create much more complex calculations to do this. If you are sure that you don’t need this additional complexity, then...
one of the previous examples will suffice. However, if you spend the extra time setting up
the template described below, you know that the chart will work, no matter what the
parameters of the inputs are. Be warned: these formulas are not for the faint-hearted! You
might prefer to simply type in the formulas as described instead of trying to follow the logic
of what each formula is trying to do as detailed below.

1. To do this, we need to treat the series which are positive, negative, above and below
the X axis (or zero) separately, so we need to plot five series:
   a. Positive values above zero
   b. Positive values below zero
   c. Negative values below zero
   d. Negative values above zero
   e. The floating or dummy series (labelled “blank” below)
2. Therefore you firstly need to set up the sheet to look like this:

   ![Fig 12.82 Layout for complex waterfall chart]

   3. **Positive values above zero:** to calculate this, we need to show the minimum of the
cumulative amount and the current month’s amount – but only if this value is
greater than zero. The formula in cell D2 should be:
   \[=\text{MAX}(0,\text{MIN}(\text{SUM}(B$2:B2),B2))\]
   Copy it down the range.

   4. **Positive values below zero:** here if the series goes below zero, we’ll need to know
the difference between the value entered, and the positive amount above zero
(calculated in the last step), displayed as a negative amount. The formula in cell E2
should be \[=\text{MAX}(0,B2-D2)\]. Copy it down the range.

   5. **Negative values below zero:** to calculate this, we need to show the maximum of the
cumulative amount and the current month’s amount – but only if this value is less
than zero. The formula in cell F2 should be: \[=\text{MIN}(0,\text{MAX}(\text{SUM}(B$2:B2),B2))\]

   6. **Negative values above zero:** if the negative values go above zero, we want to
calculate the difference between the value entered, and the negative amount above
zero (calculated in the last step). The formula in cell G2 should be: \[=\text{MAX}(0,F2-B2)\]
   Copy it down the range.

   7. **The floating or dummy series:** this calculation adds together both the highest and
lowest of the cumulative values so that it can be a starting point for the other series.
The formula in cell C3 should be:

\[
\text{MAX(0,MIN(SUM(B$2:B2),SUM(B$2:B3)))} + \text{MIN(0,MAX(SUM(B$2:B2),SUM(B$2:B3 \}))}
\]

Copy it down the range. Note that cell C2 should be left blank.

8. Now that all five series have been completed, create the stacked chart by highlighting the whole block except for the cash amounts in column B. Your sheet should look something like this:

![Diagram of Draft Complex Waterfall Chart](image)

**Fig 12.83 Draft complex waterfall chart**

9. Change the colour of the “Blank” series to white font or no fill.

10. Now you have four different series in different colours. We want the positive and negative amounts to be the same colour, so change the colour of two of series so that they are the same. You might prefer to have all series the same colour.

11. Delete the legend, and add a title.

12. Your waterfall chart should look something like this:

![Diagram of Completed Complex Waterfall Chart](image)

**Fig 12.84 Completed complex waterfall chart**
Additional Resources and Research

Congratulations! You have now completed this Financial Modelling Course and you have now come further along the road to becoming an expert Financial Modeller.

You will have to agree that we have covered a lot during the course. This course was designed to get you to the point where you can competently perform a variety of operations using the software.

The following is a little advice about what to do next:

Spend some time playing with what you have learnt. You should reinforce the skills that you have acquired and use some of the modelling techniques we have discussed.

You will be taking a copy of your model/s that we have built together home with you today. Make sure you refer to it in future when you are building your own model.

Where to From Here

We have tried to build up your skills and knowledge by having you work through specific tasks. The step by step approach that we have taken will serve as a perfect reference for you when you need to repeat a task.

However, the step by step approach we have taken today does not necessarily mean that you fully understand the operation or what is going on. We strongly urge you to try things out for yourself. Continue to use Excel, experiment with little-known features or features that you are unsure of, and your learning will be enhanced even further.

Web Links

Whilst www.plumsolutions.com.au/freestuff is a good place to start, there are a lot of other online resources available as well. Here are some of our favourite Excel & Financial Modelling websites – there is loads of free information here, so make sure you check them out. There are also a lot of free newsletters you can subscribe to with handy tips and tricks.

- http://chandoo.org/wp/ aims to make you "awesome in Excel" and his blog is incredibly popular, due to the fantastic info he sends out all for free - pretty much on a daily basis! Signing up to his newsletter is one of the best moves you can make if you want to continually improve your Excel skills.

- Financial Mechanics have some fantastic new videos at http://financialmechanics.tv you should definitely check out, as well as a newsletter and user forum and a new podcast series.

• [http://blog.corality.com](http://blog.corality.com) A great blog with tutorials on lots of Financial Modelling and Excel-related topics, especially the area of risk and model audit.

• [http://www.navigatorpf.com](http://www.navigatorpf.com) also has some excellent tutorials for download and a blog.

• [http://www.fimodo.com/](http://www.fimodo.com/) is a refreshingly impartial Financial Modelling blog with contributions from Financial Modelling experts around the world. No hidden sales or agendas!


• [http://www.financialmodelingguide.com](http://www.financialmodelingguide.com) and the related www.finance30.com have a huge student following and have a lot of content on general finance as well as Financial Modelling, although you sometimes have to wade through a lot of content to get to the good stuff.

• Join the LinkedIn “Financial Modelling in Excel” group at [www.LinkedIn.com](http://www.LinkedIn.com). If you aren’t a LinkedIn user yet, it is a great place to connect and stay connected with people in your industry – kind of like Facebook for Business!

Many of these guys you can follow on Twitter for a constant stream of updates and Excel tips and tricks.

### White Papers & Other Documents

Available for download online. You may need to register to access them.


• “Best Practice Modelling Standards, Spreadsheet Standards Review Board available at [http://www.ssrb.org](http://www.ssrb.org)

### Other Books

These books are all available at [www.bookdepository.com](http://www.bookdepository.com) or [www.amazon.com](http://www.amazon.com)


• **Building Financial Models with Microsoft Excel**: a guide for Business Professionals” – K. Scott Proctor, Wiley Finance, 2009


• “**Next Generation Excel**: Modeling in Excel for Analysts and MBAs” – Isaac Gottlieb, Wiley Finance, 2010

• “**Show me the Numbers – Designing Tables and Graphs to Enlighten**” – Stephen Few, Analytics Press 2004