1 Building Blocks: Selected Excel Functions and Tools

This chapter provides examples of the use of a selection of Excel functions. It is not possible within the scope of this text to provide complete coverage of all Excel functions; rather the focus is on those that are generally important in financial modelling at the intermediate and advanced level. Readers may naturally refer to other texts on Excel or to the **Help** menu within Excel (**F1** short-cut) to learn more about the full range of functions.

CORE FUNCTIONS FOR FINANCIAL MODELLING

This section summarises the basic functions required for many financial modelling applications. While many of these are essentially self-explanatory and are likely to be well known to many readers, certain aspects of their use and features are worth highlighting.

Arithmetic Operations

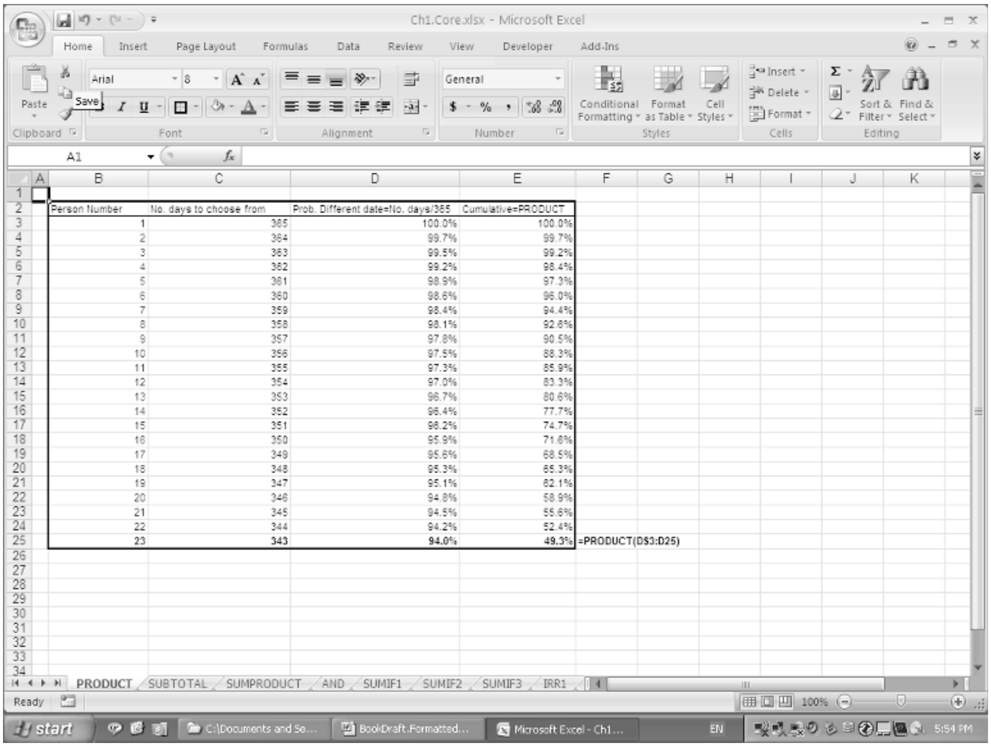
The basic functions for arithmetical operations (classified in Excel within either the **Math & Trig** or **Statistical** categories) include:

- **AVERAGE** calculates the average of a set of numbers.
- **COUNT** counts the number of cells that contain numbers (**COUNTA** counts the number of non-empty cells, and so includes the counting of text fields).
- **MIN** and **MAX** calculate the minimum and maximum of a set of values.
- **PRODUCT** multiplies its arguments.
- **SUBTOTAL** calculates the sum (or other values) of a range of cells, ignoring other **SUBTOTAL** functions, so avoiding potential double-counting of values.
- **SUM** adds up a set of numbers.
- **SUMPRODUCT** multiplies the corresponding elements of two ranges and forms their sum.

Example: PRODUCT

Where the values in a contiguous range of cells are to be multiplied, the **PRODUCT** function provides a smaller formula with easier updating than the alternative approach (in which individual cell references are multiplied).

The file Ch1.Core.xlsx (PRODUCT worksheet) (Figure 1.1) shows an example in which a range of cells containing probabilities is multiplied. It shows that there is a probability of just less than 50% that a group of 23 people have birthdays on different days to each other. That is, in a group of 23 people, it is more likely than not that at least two people share a birthday.



Person Number	No. days to choose from	Prob. Different date=No. days/365	Cumulative=PRODUCT	
1	365	100.0%	100.0%	
2	364	99.7%	99.7%	
3	363	99.5%	99.2%	
4	362	99.2%	98.4%	
5	361	98.9%	97.3%	
6	360	98.6%	96.0%	
7	359	98.4%	94.4%	
8	358	98.1%	92.6%	
9	357	97.8%	90.5%	
10	356	97.5%	88.3%	
11	355	97.3%	85.9%	
12	354	97.0%	83.3%	
13	353	96.7%	80.6%	
14	352	96.4%	77.7%	
15	351	96.2%	74.7%	
16	350	95.9%	71.6%	
17	349	95.6%	68.5%	
18	348	95.3%	65.3%	
19	347	95.1%	62.1%	
20	346	94.8%	58.9%	
21	345	94.5%	55.6%	
22	344	94.2%	52.4%	
23	**343**	**94.0%**	**49.3%**	=PRODUCT(D$3:D25)

Figure 1.1

Example: SUBTOTAL

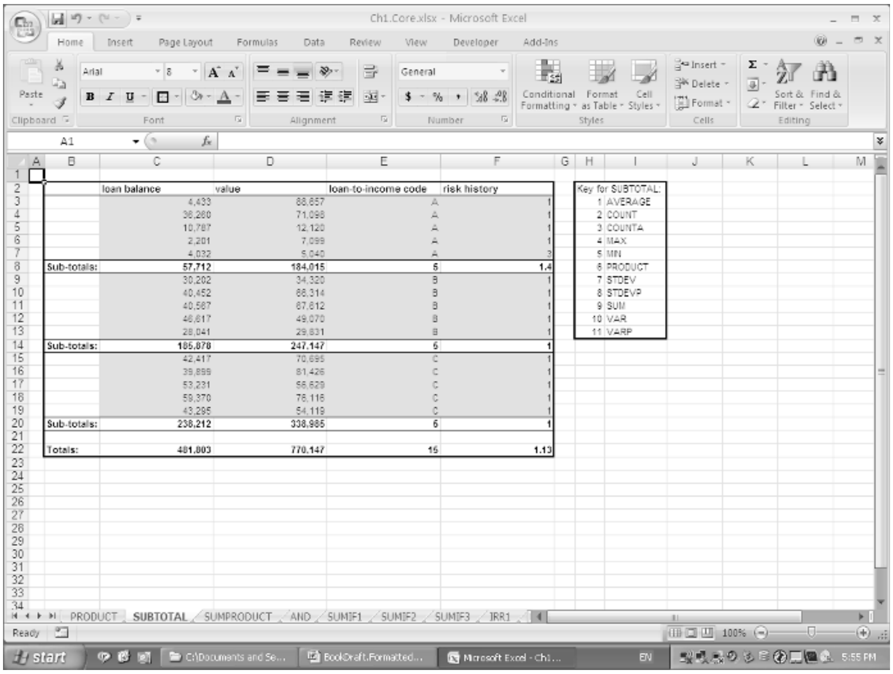
The **SUBTOTAL** function ignores other **SUBTOTAL** functions, and so avoids double-counting when applied to a range that contains this function (unlike the **SUM** function, which would lead either to double-counting or to a large set of cumbersome, inflexible and error-prone formulae).

The function has an argument that allows different calculations to be performed on the data set. For example, the sum of the range requires the use of the argument 9, whereas the average and count require the value of 1 and 2 respectively (see the **Help** menu for the full description).

Frequent uses of the function include:

- The creation of subtotals in a large list of data that is sorted into categories.
- In financial statement modelling, where a company's total assets may be calculated from the (subtotal) of its fixed and current assets, which may themselves each be calculated as the subtotal of a more detailed breakdown (such as equipment, working capital, etc.).
- The analysis of sets of filtered data (see later), where the function ignores any hidden rows that result from a list having been filtered (unlike **SUM**, **COUNT**).

The **SUBTOTAL** function can be entered either by direct insertion into a cell (by explicit typing or insertion from the **Math & Trig** category), or by use of **Data/Subtotal** (**Data/**



	loan balance	value	loan-to-income code	risk history
	4,433	88,657	A	1
	36,260	71,098	A	1
	10,787	12,120	A	1
	2,201	7,099	A	1
	4,032	5,040	A	3
Sub-totals:	57,712	184,015	5	1.4
	30,202	34,320	B	1
	40,452	66,314	B	1
	40,567	67,612	B	1
	46,617	49,070	B	1
	28,041	29,831	B	1
Sub-totals:	185,878	247,147	5	1
	42,417	70,695	C	1
	39,899	81,426	C	1
	53,231	56,629	C	1
	59,370	76,116	C	1
	43,295	54,119	C	1
Sub-totals:	238,212	338,985	6	1
Totals:	481,803	770,147	15	1.13

Key for SUBTOTAL:	
1	AVERAGE
2	COUNT
3	COUNTA
4	MAX
5	MIN
6	PRODUCT
7	STDEV
8	STDEVP
9	SUM
10	VAR
11	VARP

Ch1.Core.xlsx - Microsoft Excel
Home Insert Page Layout Formulas Data Review View Developer Add-Ins
A1
PRODUCT SUBTOTAL SUMPRODUCT AND SUMIF1 SUMIF2 SUMIF3 IRR1
Ready

Figure 1.2

Subtotals in Excel 2003) when applied to a list or table of data. In the latter case the data will usually have first been ordered or sorted in some way (perhaps through use of the **Data/Sort** menu), so that the inserted subtotals are at the relevant break-points in the list. This latter route will result in grouped data appearing.

The file Ch1.Core.xlsx (SUBTOTAL worksheet) (Figure 1.2) shows an example where the function was entered by direct insertion (the arguments for the different types are also shown for convenience).

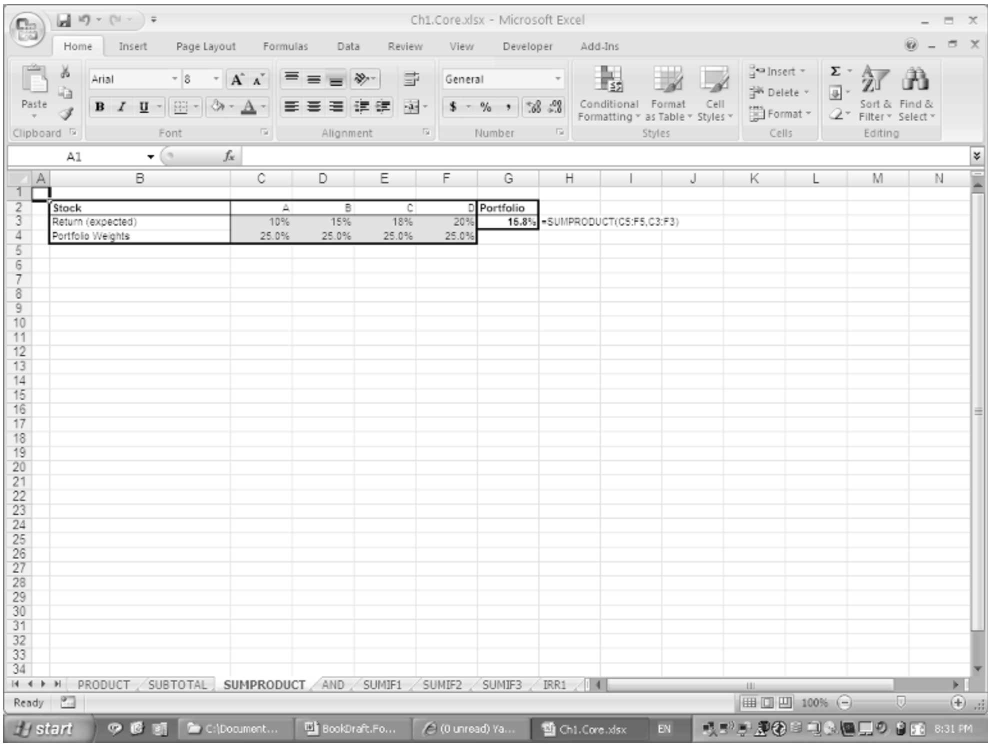
Example: SUMPRODUCT

The file Ch1.Core.xlsx (SUMPRODUCT worksheet) (Figure 1.3) shows an example of the **SUMPRODUCT** function in a simple portfolio analysis situation. It is used in order to calculate the weighted average (i.e. expected) return of a portfolio that consists of assets with given weights and expected returns.

Logical Operations

The basic logical functions include:

- **AND** checks if two conditions both hold, and returns **TRUE** or **FALSE** accordingly. Similarly **OR** and **NOT** functions exist. These can be useful to avoid writing embedded **IF** statements when checking multiple conditions.



Stock	A	B	C	D	Portfolio
Return (expected)	10%	15%	18%	20%	15.8%
Portfolio Weights	25.0%	25.0%	25.0%	25.0%	

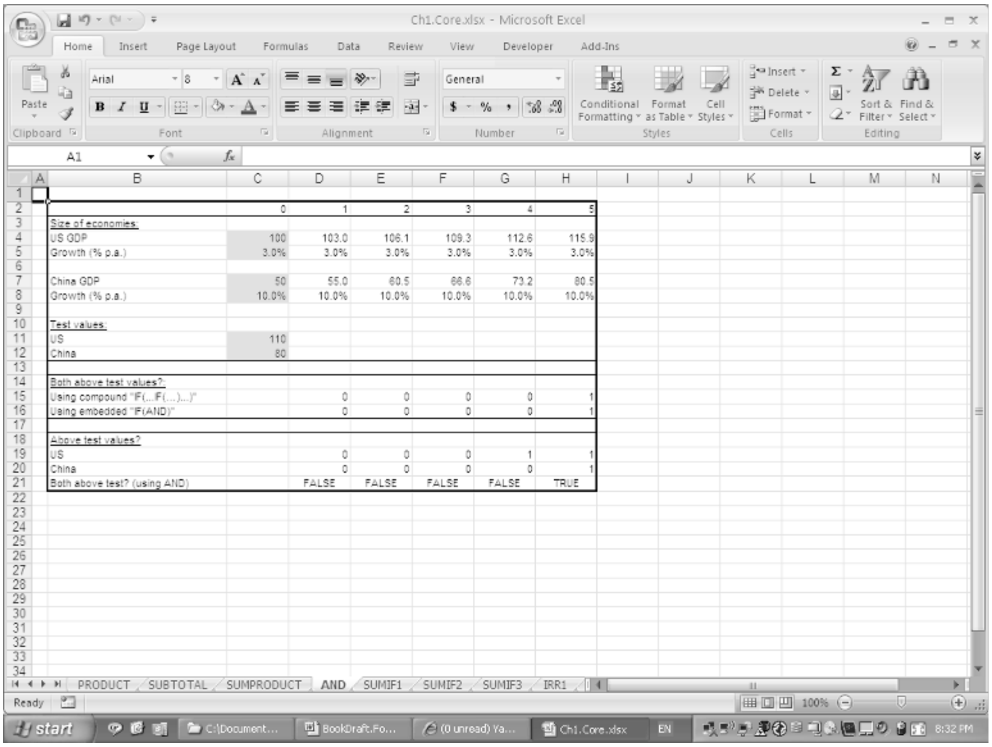
=SUMPRODUCT(C5:F5,C3:F3)

Figure 1.3

- **IF** checks whether a condition is true or not and returns a specified value in each case. Its use is implicit in a direct comparison expression such as =**F7**>**F6**, which would evaluate to either **TRUE** or **FALSE** (these are not text strings, but when used in any subsequent formulae, are interpreted by Excel as 1 or 0 respectively). Therefore =**50*(F7**>**F6)** would return either 50 or 0. Similarly, while one may write =**IF(F7**>**F6,1,0)**, this would not be the same as =**IF(F7**>**F6,"TRUE","FALSE")**, which returns text strings (and is therefore generally inconvenient when the results of such expressions are to be used in further numerical calculations).

Related functions include:

- **SUMIF** (classified in the **Math & Trig** category) adds the values of cells in a given range according to whether a criterion is met in another range. Excel 2007 also has a **SUMIFS** function in which a range is summed according to multiple criteria being met; an example is shown later in this chapter. In addition, in some cases the use of **Database** functions, **PivotTables**, or the **Conditional Sum Wizard** can provide more appropriate alternatives (see later).
- **COUNTIF** (classified in the **Statistical** category) counts the number of cells that meet a specified criterion. In Excel 2007, the **AVERAGEIF** function exists, as do **AVERAGEIFS** and **COUNTIFS** when multiple criteria are to be met.



	C	D	E	F	G	H
	0	1	2	3	4	5
Size of economies:						
US GDP	100	103.0	106.1	109.3	112.6	115.9
Growth (% p.a.)	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%
China GDP	50	55.0	60.5	66.6	73.2	80.5
Growth (% p.a.)	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%
Test values:						
US	110					
China	80					
Both above test values?:						
Using compound "IF(...IF(....)...)"		0	0	0	0	1
Using embedded "IF(AND)"		0	0	0	0	1
Above test values?						
US		0	0	0	1	1
China		0	0	0	0	1
Both above test? (using AND)		FALSE	FALSE	FALSE	FALSE	TRUE

Figure 1.4

Example: AND

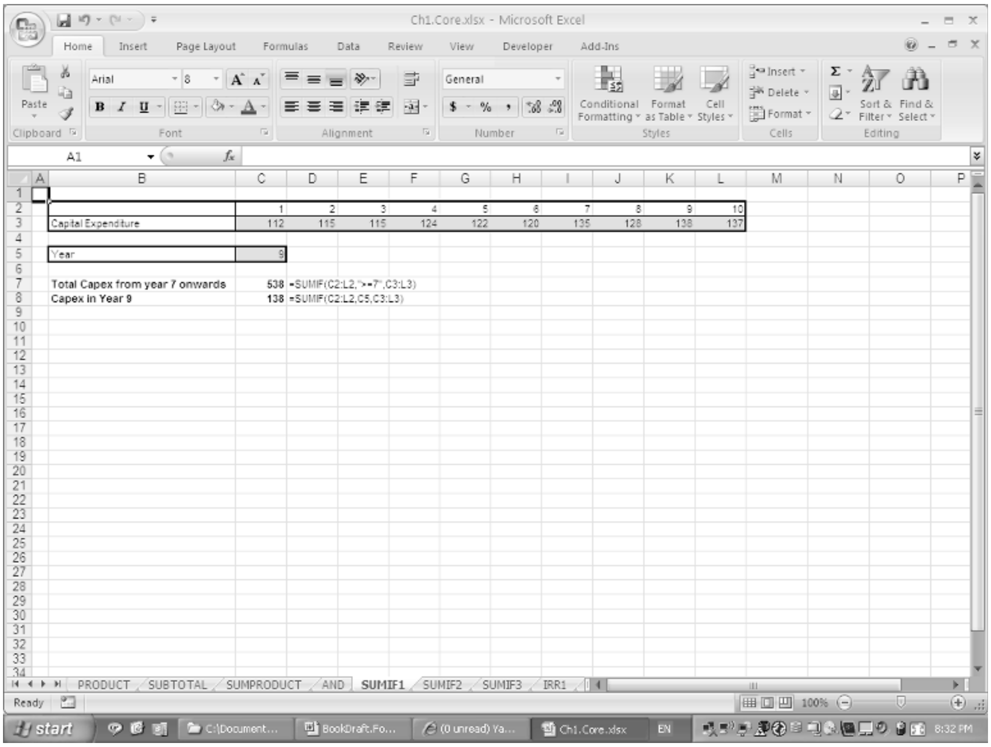
The file Ch1.Core.xlsx (AND worksheet) (Figure 1.4) shows the hypothesised development of the gross domestic product (GDP) of the US and Chinese economies (indexed so that the starting value of the US is 100), and demonstrates the use of **AND** to check whether two conditions hold simultaneously. Various possibilities are shown, including a compound **IF** statement, the **AND** function embedded within the **IF** statement, and the **AND** statement applied to the result of checking individually whether each of the conditions is met. Note that in the previous example, the **AND** function returns either **TRUE** or **FALSE**.

Example: SUMIF

The **SUMIF** function adds the values of the cells in a range according to whether a criterion is met in another range.

The file Ch1.Core.xlsx (SUMIF1 worksheet) (Figure 1.5) shows its use to calculate the total capital expenditure from Year 7 onwards in a 10-year forecast, as well as to lookup the capital expenditure in Year 9 (this could also be achieved with a **Lookup** function, described later).

The **SUMIF** function can be particularly useful in modelling applications where the values of some model inputs are themselves derived from data sets.



	1	2	3	4	5	6	7	8	9	10
Capital Expenditure	112	115	115	124	122	120	135	128	138	137
Year	9									
Total Capex from year 7 onwards	538	=SUMIF(C2:L2,">=7",C3:L3)								
Capex in Year 9	138	=SUMIF(C2:L2,C5,C3:L3)								

Figure 1.5

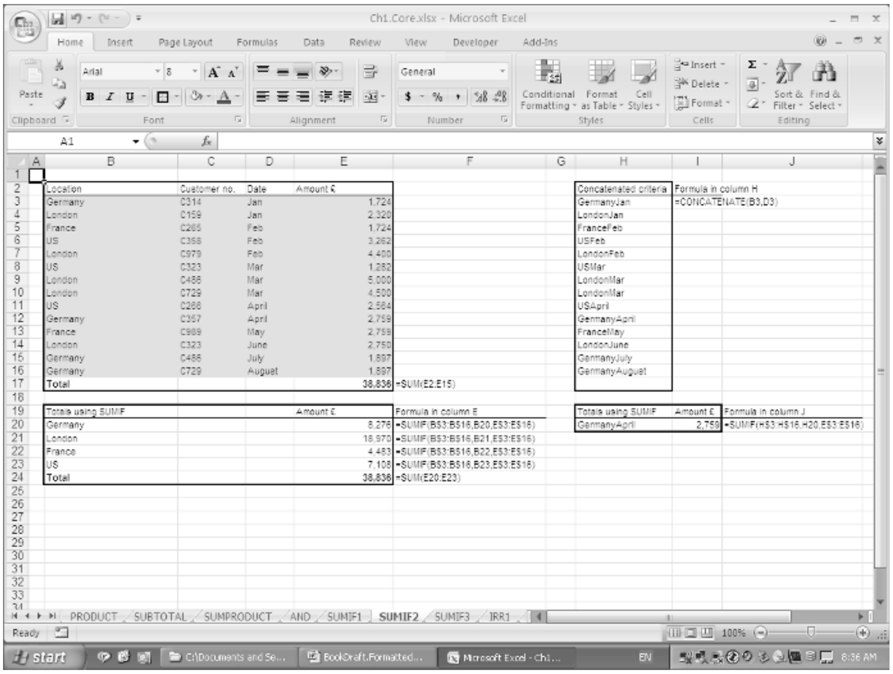
The file Ch1.Core.xlsx (SUMIF2 worksheet) (Figure 1.6) shows the use of the function to perform simple database queries. It also shows how the concatenation of multiple database fields (using **&** or the **CONCATENATE** function) can often be used to create a sum according to the multiple criteria being met. This approach can sometimes be easier and more flexible than the alternatives (which are discussed later, including the **SUMIFS** function, the **Conditional Sum Wizard**, **Database** functions (which would require setting up many criteria ranges with field headings for each), or **PivotTables** (where there would be no live-link to the data set).

The file Ch1.Core.xlsx (SUMIF3 worksheet) (Figure 1.7) shows how the function may be used to determine the number of unique records in a list.

Financial Calculations

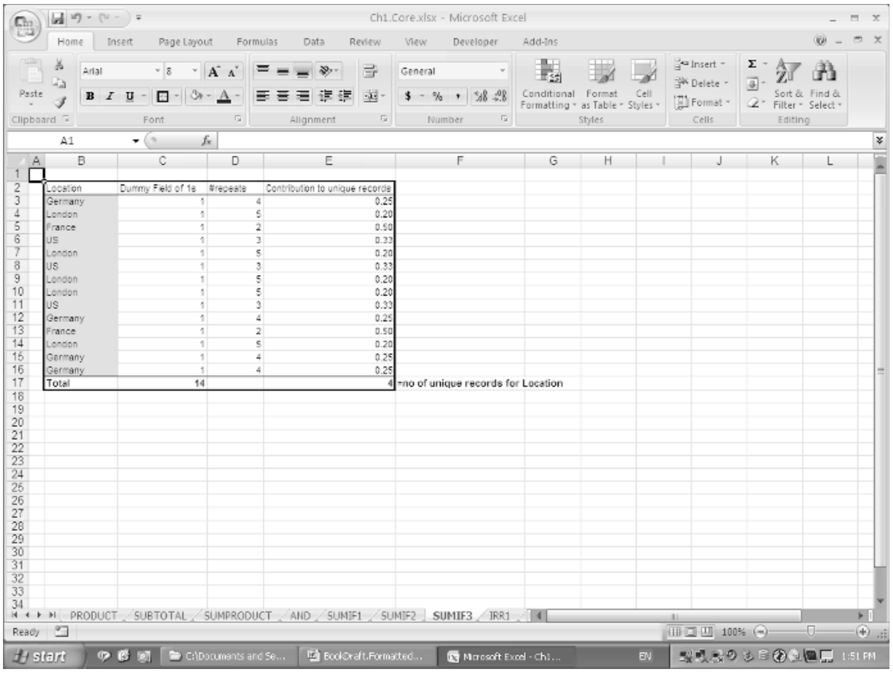
Certain functions are frequently used in financial calculations (and are classified in either the **Financial** or **Math & Trig** categories) including:

- **IRR** calculates the internal rate-of-return of equally spaced cash flows, and has a number of applications, such as in project evaluation and yield analysis. (The **XIRR** function can be used where the cash flows are not equally spaced, and the **YIELD** function for



	B	C	D	E	F	H	I	J
2	Location	Customer no.	Date	Amount £		Concatenated criteria	Formula in column H	
3	Germany	C314	Jan	1,724		GermanyJan	=CONCATENATE(B3,D3)	
4	London	C159	Jan	2,320		LondonJan		
5	France	C265	Feb	1,724		FranceFeb		
6	US	C358	Feb	3,262		USFeb		
7	London	C979	Feb	4,400		LondonFeb		
8	US	C323	Mar	1,282		USMar		
9	London	C486	Mar	5,000		LondonMar		
10	London	C729	Mar	4,500		LondonMar		
11	US	C266	April	2,564		USApril		
12	Germany	C357	April	2,759		GermanyApril		
13	France	C989	May	2,759		FranceMay		
14	London	C323	June	2,750		LondonJune		
15	Germany	C486	July	1,897		GermanyJuly		
16	Germany	C729	August	1,897		GermanyAugust		
17	Total			38,836	=SUM(E2:E15)			
18								
19	Totals using SUMIF			Amount £	Formula in column E	Totals using SUMIF	Amount £	Formula in column J
20	Germany			8,276	=SUMIF(B$3:B$16,B20,E$3:E$16)	GermanyApril	2,759	=SUMIF(H$3:H$16,H20,E$3:E$16)
21	London			18,970	=SUMIF(B$3:B$16,B21,E$3:E$16)			
22	France			4,483	=SUMIF(B$3:B$16,B22,E$3:E$16)			
23	US			7,108	=SUMIF(B$3:B$16,B23,E$3:E$16)			
24	Total			38,836	=SUM(E20:E23)			

Figure 1.6



	B	C	D	E	F
2	Location	Dummy Field of 1s	#repeats	Contribution to unique records	
3	Germany	1	4	0.25	
4	London	1	5	0.20	
5	France	1	2	0.50	
6	US	1	3	0.33	
7	London	1	5	0.20	
8	US	1	3	0.33	
9	London	1	5	0.20	
10	London	1	5	0.20	
11	US	1	3	0.33	
12	Germany	1	4	0.25	
13	France	1	2	0.50	
14	London	1	5	0.20	
15	Germany	1	4	0.25	
16	Germany	1	4	0.25	
17	Total	14		4	=no of unique records for Location

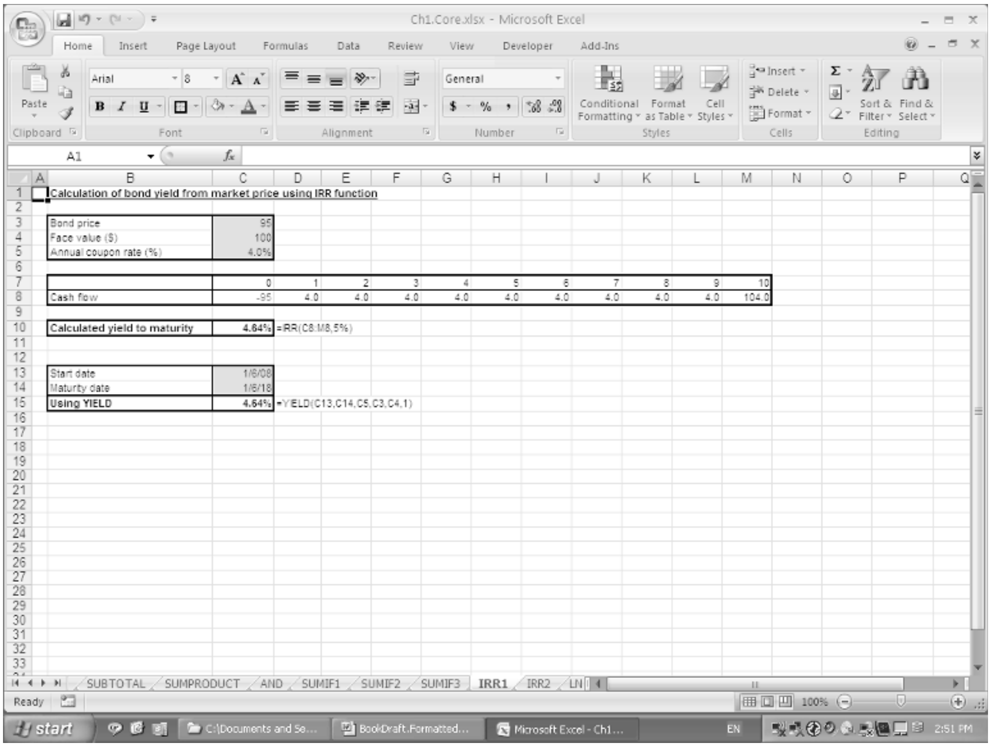
Figure 1.7

bond-related applications; these functions are included in Excel 2007, but are part of the **Analysis ToolPak** add-in in Excel 2003; see later.)

- **LN** calculates the natural logarithm of a number (and **EXP** the exponential).
- **NPV** calculates the net present value of equally spaced cash flows at a given discount rate. (The **XNPV** function can be used when cash flows are not equally spaced; this function is included in Excel 2007, but is part of the **Analysis ToolPak** add-in in Excel 2003; see later.)
- **PMT** calculates the constant level of repayment required on a loan (interest and principal) with a fixed interest rate (similarly **PPMT** calculates the principal repayment component only).

Example: IRR

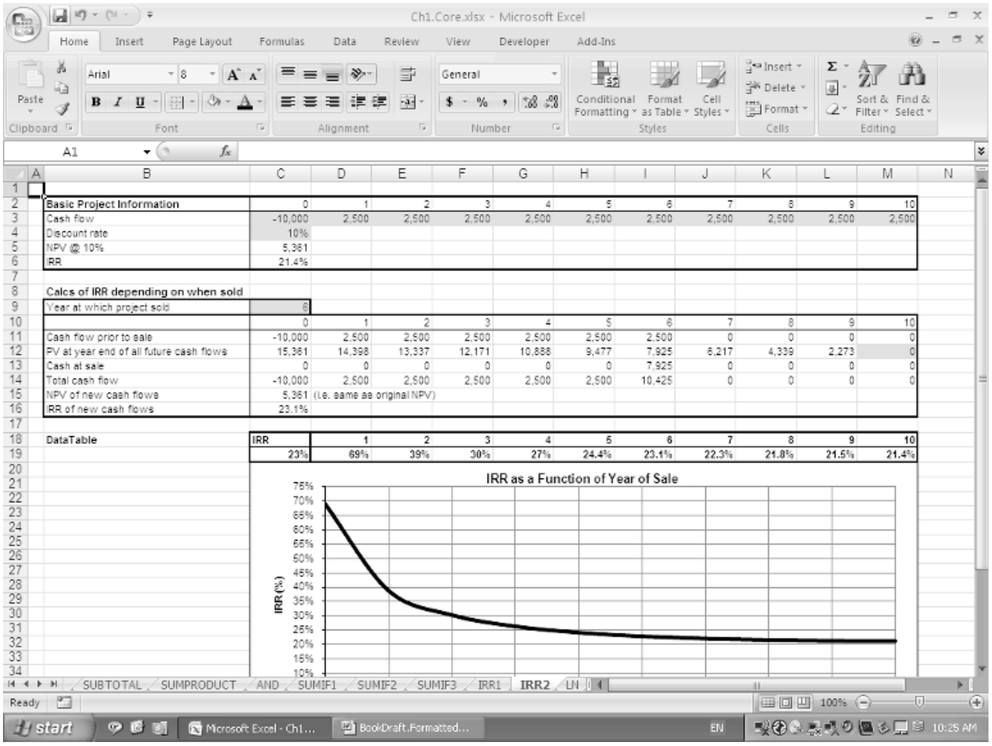
The file Ch1.Core.xlsx (IRR1 worksheet) (Figure 1.8) uses the **IRR** function to calculate the yield on a bond with an assumed current purchase price and a repayment schedule. The **YIELD** function could also be used for such a calculation, and this is also shown. Of course the **IRR** function can be used for any profile of periodic cash flows, whereas the **YIELD** function is only applicable in the specific application of bond yields (as the periodic cash flows are implicit from the face value, maturity and coupon frequency, and do not need to be explicitly calculated).



Calculation of bond yield from market price using IRR function

Bond price	95											
Face value ($)	100											
Annual coupon rate (%)	4.0%											
	0	1	2	3	4	5	6	7	8	9	10	
Cash flow	-95	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	104.0	
Calculated yield to maturity	**4.64%**	=IRR(C8:M8,5%)										
Start date	1/6/08											
Maturity date	1/6/18											
Using YIELD	**4.64%**	=YIELD(C13,C14,C5,C3,C4,1)										

Figure 1.8



	B	C	D	E	F	G	H	I	J	K	L	M
1												
2	**Basic Project Information**	0	1	2	3	4	5	6	7	8	9	10
3	Cash flow	-10,000	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500	2,500
4	Discount rate	10%										
5	NPV @ 10%	5,361										
6	IRR	21.4%										
7												
8	**Calcs of IRR depending on when sold**											
9	Year at which project sold	6										
10		0	1	2	3	4	5	6	7	8	9	10
11	Cash flow prior to sale	-10,000	2,500	2,500	2,500	2,500	2,500	2,500	0	0	0	0
12	PV at year end of all future cash flows	15,361	14,398	13,337	12,171	10,888	9,477	7,925	6,217	4,339	2,273	0
13	Cash at sale	0	0	0	0	0	0	7,925	0	0	0	0
14	Total cash flow	-10,000	2,500	2,500	2,500	2,500	2,500	10,425	0	0	0	0
15	NPV of new cash flows	5,361	(i.e. same as original NPV)									
16	IRR of new cash flows	23.1%										
17												
18	**DataTable**	IRR	1	2	3	4	5	6	7	8	9	10
19		23%	69%	39%	30%	27%	24.4%	23.1%	22.3%	21.8%	21.5%	21.4%

Figure 1.9

The internal rate-of-return is the discount rate that would result in the net present value of the cash flows being equal to zero. It is well-known that its use as a measure of project performance is inadequate. For example, when the cash flows of a project change sign over time, there will be several values possible values for the internal rate-of-return. More subtly, if a project may at any point be sold for its future net present value, then the internal rate-of-return of the project depends on the date at which the project is sold. This means that its use as a measure of performance or of project selection can be misleading.

The file Ch1.Core.xlsx (IRR2 worksheet) (Figure 1.9) demonstrates this (it also uses the **Data Table** functionality described in Chapter 2 to test the sensitivity of the internal rate-of-return to the date at which the project is sold).

Example: LN

The **LN** function is useful in many contexts, including the calculation of the growth rates of asset values and the calculation of half-lives in maintenance modelling. If the value of a process halves every T years, its value at the end of each year is equal to **EXP(LN(0.5)/T)** of the value at the beginning of the year. Similarly, if an asset's value grows continously in time, then its future value will grow exponentially. When calculating asset returns, the use of the **LN** function applied to the ratio of the ending to beginning values over a period of time will give the continuous time (constant compounding) growth rate:

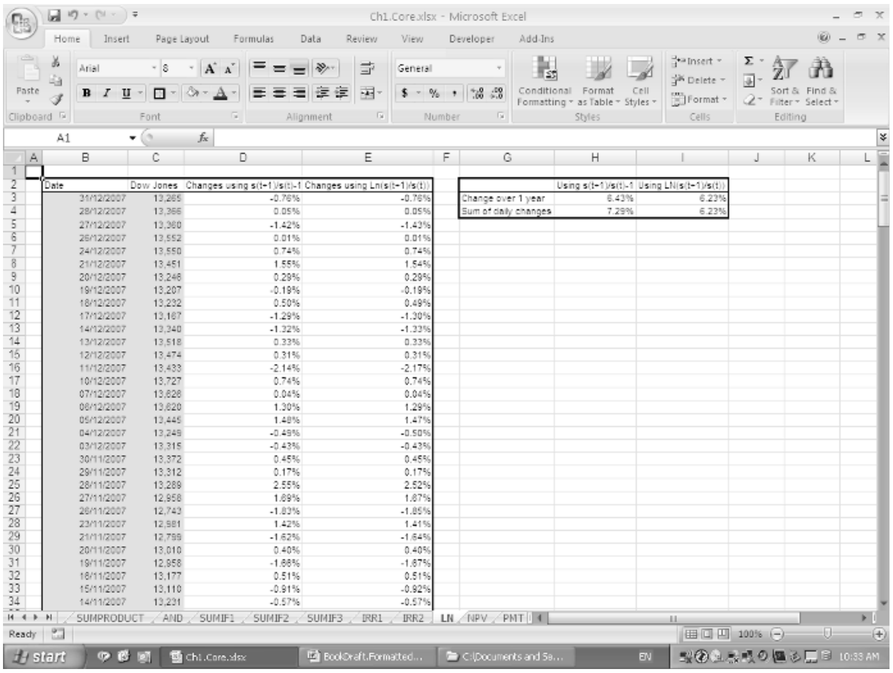
Growth rate = **LN**(Ending/Starting)

One of the properties of this method of calculating growth rates is that the periodic growth rates are additive, e.g. the **LN** of the ratio calculated from year end and beginning asset values is equal to the sum of the changes of the **LN** of the ratio on a daily basis. When considering very short periods (such as daily changes) it makes little difference to the individual measurements whether the **LN** function is applied to the ratios of values or whether the change is measured using the alternative formula:

Growth rate = Ending/Starting − 1

This latter formula is used in applications where the passage of time is considered to be a discrete process, rather than a continuous one. Note that the additive property of the changes is lost when this approach is used. However, the ending period asset value when using the two approaches in a forecasting sense would be the same as long as each methodology is used in a self-consistent way. That is, the calculation of an ending value when the input change is assumed to be measured by the first formula would require use of the **EXP** function, whereas for the second approach it would involve a multiplication of the starting value by 1 plus the growth rate.

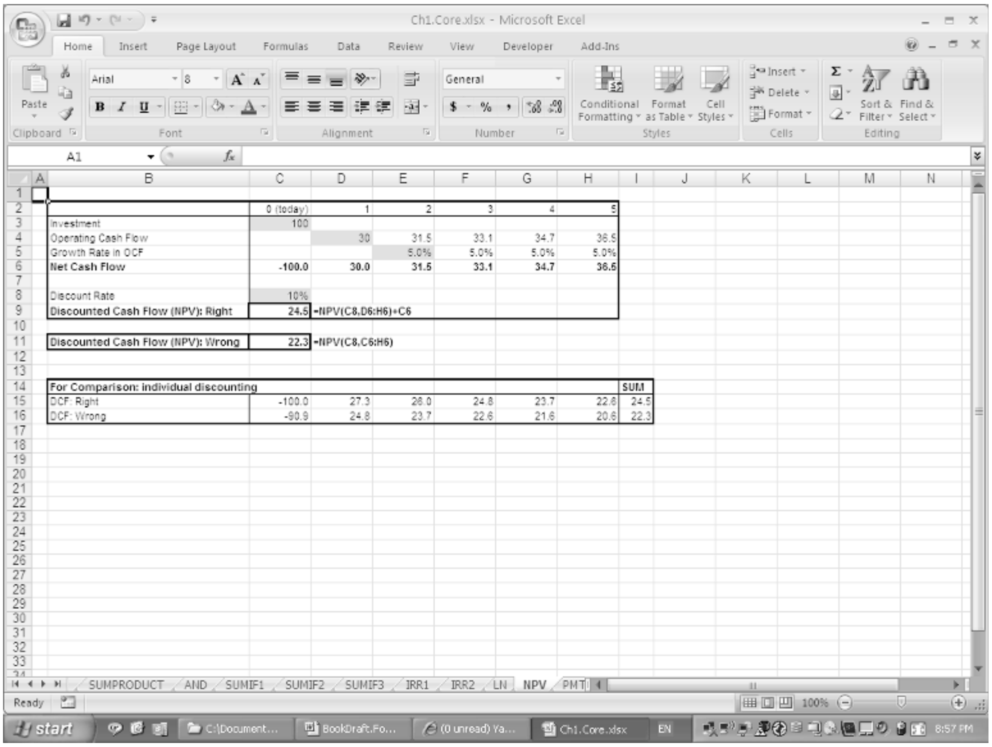
The file Ch1.Core.xlsx (LN worksheet) (Figure 1.10) shows the two calculation approaches using a set of daily data for the Dow Jones index in 2007.



Date	Dow Jones	Changes using s(t+1)/s(t)-1	Changes using Ln(s(t+1)/s(t))
31/12/2007	13,265	-0.76%	-0.76%
28/12/2007	13,366	0.05%	0.05%
27/12/2007	13,360	-1.42%	-1.43%
26/12/2007	13,552	0.01%	0.01%
24/12/2007	13,550	0.74%	0.74%
21/12/2007	13,451	1.55%	1.54%
20/12/2007	13,246	0.29%	0.29%
19/12/2007	13,207	-0.19%	-0.19%
18/12/2007	13,232	0.50%	0.49%
17/12/2007	13,167	-1.29%	-1.30%
14/12/2007	13,340	-1.32%	-1.33%
13/12/2007	13,518	0.33%	0.33%
12/12/2007	13,474	0.31%	0.31%
11/12/2007	13,433	-2.14%	-2.17%
10/12/2007	13,727	0.74%	0.74%
07/12/2007	13,626	0.04%	0.04%
06/12/2007	13,620	1.30%	1.29%
05/12/2007	13,445	1.48%	1.47%
04/12/2007	13,249	-0.49%	-0.50%
03/12/2007	13,315	-0.43%	-0.43%
30/11/2007	13,372	0.45%	0.45%
29/11/2007	13,312	0.17%	0.17%
28/11/2007	13,289	2.55%	2.52%
27/11/2007	12,958	1.69%	1.67%
26/11/2007	12,743	-1.83%	-1.85%
23/11/2007	12,981	1.42%	1.41%
21/11/2007	12,799	-1.62%	-1.64%
20/11/2007	13,010	0.40%	0.40%
19/11/2007	12,958	-1.66%	-1.67%
16/11/2007	13,177	0.51%	0.51%
15/11/2007	13,110	-0.91%	-0.92%
14/11/2007	13,231	-0.57%	-0.57%

	Using s(t+1)/s(t)-1	Using LN(s(t+1)/s(t))
Change over 1 year	6.43%	6.23%
Sum of daily changes	7.29%	6.23%

Figure 1.10



	B	C	D	E	F	G	H	I
2		0 (today)	1	2	3	4	5	
3	Investment	100						
4	Operating Cash Flow		30	31.5	33.1	34.7	36.5	
5	Growth Rate in OCF			5.0%	5.0%	5.0%	5.0%	
6	**Net Cash Flow**	**-100.0**	**30.0**	**31.5**	**33.1**	**34.7**	**36.5**	
7								
8	Discount Rate	10%						
9	**Discounted Cash Flow (NPV): Right**	**24.5**	**=NPV(C8,D6:H6)+C6**					
10								
11	**Discounted Cash Flow (NPV): Wrong**	**22.3**	**=NPV(C8,C6:H6)**					
12								
13								
14	**For Comparison: individual discounting**							**SUM**
15	DCF: Right	-100.0	27.3	26.0	24.8	23.7	22.6	24.5
16	DCF: Wrong	-90.9	24.8	23.7	22.6	21.6	20.6	22.3

Figure 1.11

Example: NPV

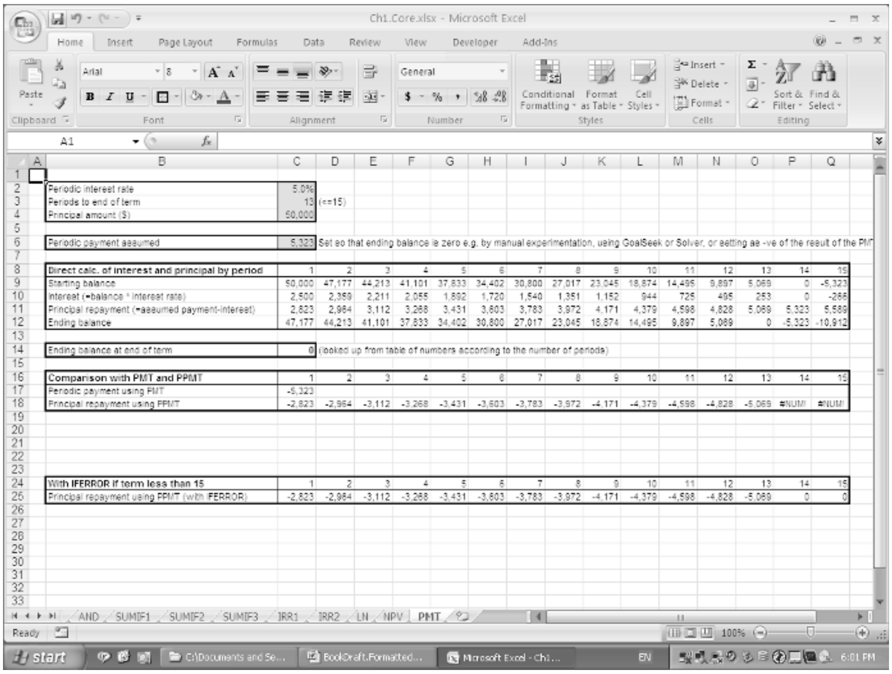
The use of the **NPV** function to calculate the discounted value of a set of cash flows is essentially straightforward providing one is familiar with the concept of discounting. The main area where mistakes are often made is to overlook that the function implicitly assumes that the value in the first cell of the range is discounted for one period, the second value for two periods, and so on.

The file Ch1.Core.xlsx (NPV worksheet) (Figure 1.11) shows an example in which the cash outflow associated with an investment that is being made immediately would not require discounting, and should be excluded as an argument of the **NPV** function (otherwise all cash flows will be discounted by one period too many).

Example: PMT and PPMT

The **PMT** and **PPMT** functions can be useful to calculate respectively the total and the principal component of the periodic repayments required on a loan (the interest payment being the difference, but which can more easily be calculated directly from the interest rate and loan balance).

The file Ch1.Core.xlsx (PMT worksheet) (Figure 1.12) shows the explicit calculations of the repayments (split between interest and principal), where the required total repayment level must be found by trial and error (or by using Excel's **GoalSeek** or **Solver** described



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
2	Periodic interest rate	5.0%														
3	Periods to end of term	13	(<=15)													
4	Principal amount ($)	50,000														
6	Periodic payment assumed	5,323	Set so that ending balance is zero e.g. by manual experimentation, using GoalSeek or Solver, or setting as -ve of the result of the PMT													
8	**Direct calc. of interest and principal by period**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	Starting balance	50,000	47,177	44,213	41,101	37,833	34,402	30,800	27,017	23,045	18,874	14,495	9,897	5,069	0	-5,323
10	Interest (=balance * interest rate)	2,500	2,359	2,211	2,055	1,892	1,720	1,540	1,351	1,152	944	725	495	253	0	-266
11	Principal repayment (=assumed payment-interest)	2,823	2,964	3,112	3,268	3,431	3,603	3,783	3,972	4,171	4,379	4,598	4,828	5,069	5,323	5,589
12	Ending balance	47,177	44,213	41,101	37,833	34,402	30,800	27,017	23,045	18,874	14,495	9,897	5,069	0	-5,323	-10,912
14	Ending balance at end of term	0	(looked up from table of numbers according to the number of periods)													
16	**Comparison with PMT and PPMT**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	Periodic payment using PMT	-5,323														
18	Principal repayment using PPMT	-2,823	-2,964	-3,112	-3,268	-3,431	-3,603	-3,783	-3,972	-4,171	-4,379	-4,598	-4,828	-5,069	#NUM!	#NUM!
24	**With IFERROR if term less than 15**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
25	Principal repayment using PPMT (with IFERROR)	-2,823	-2,964	-3,112	-3,268	-3,431	-3,603	-3,783	-3,972	-4,171	-4,379	-4,598	-4,828	-5,069	0	0

Figure 1.12

later). The results are compared with those that would result if the **PMT** and **PPMT** functions were used directly. (In the example shown, the term of the loan is set to be equal to 13, and hence a **#NUM** error is returned for periods beyond that. In this case, such an error has no real consequence for the situation at hand. Nevertheless, the example also shows how Excel 2007's **IFERROR** function (see later for more details) can be used to reset the error value to zero in such cases.)

When using these functions, frequent mistakes include having an interest rate that is inconsistent with the periods used (e.g. monthly/annual), or overlooking that the sign of the cash flows is negative (when repayments are required). Errors can be particularly hard to spot if the functions are used in embedded formulae in non-annual models.

DATABASE FUNCTIONS, FEATURES AND PIVOT TABLES

The analysis of data sets is important in a number of modelling applications, such as the calibration of model inputs, the assessment of important factors or drivers of behaviour in a situation, as well as being generally important as a stand-alone application. This section discusses a number of Excel's tools in this area, including:

- **Database** functions, which return calculations of the values in a database that meet certain criteria without explicitly extracting the relevant data points.
- **Filter** and **Advanced Filter** options, which present or extract a filtered data set whose elements meet specific criteria.

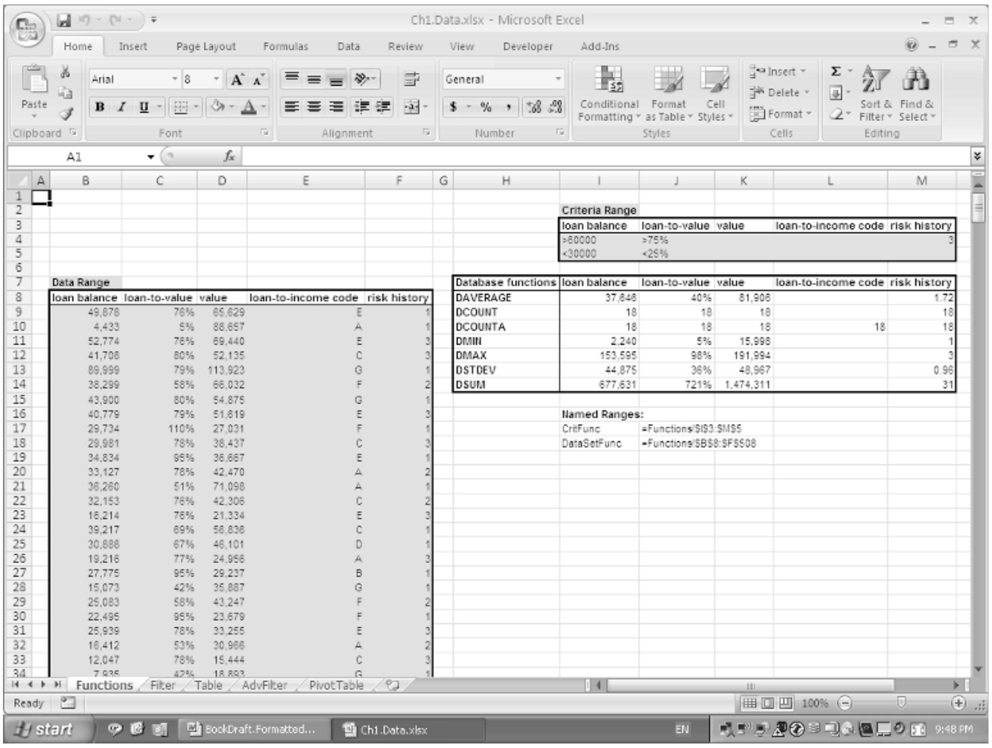
- **PivotTables**, which create summary reports by category and cross-tabulations.

Note that in some cases the **SUMIF** or **SUMIFS** functions (see earlier) or the **Conditional Sum Wizard** (see later) can provide appropriate alternatives to the use of **Database** functions. In addition, Excel has other tools to analyse data sets, including **Statistical** functions and simple regression analysis, some of which are discussed in the next section.

Example: Database Queries using DSUM and other Database Functions

The **Database** functions (such as **DAVERAGE, DCOUNT DCOUNTA DMIN, DMAX, DSUM)** calculate the relevant figure (average, count, etc.) of the numbers in a range that meet specified criteria. Statistical quantities, such as the standard deviation of the data points meeting the criteria, can also be calculated with **DSTDEV** (or **DSTDEVP**, when the sample is intended to represent the whole population) and functions such as **DVAR** and **DVARP** exist for calculating the variance.

When using **Database** functions, the field headings must be included in the definition of the database and criteria ranges. The criteria range can consist of multiple contiguous rows, where each row is equivalent to an **OR** condition (so that the presence of any blank row within the criteria range would select all records); multiple criteria within a row are equivalent to an **AND** condition, i.e. that all criteria within the row need to be met.



Criteria Range

loan balance	loan-to-value	value	loan-to-income code	risk history
>60000	>75%			3
<30000	<25%			

Database functions	loan balance	loan-to-value	value	loan-to-income code	risk history
DAVERAGE	37,846	40%	81,906		1.72
DCOUNT	18	18	18		18
DCOUNTA	18	18	18	18	18
DMIN	2,240	5%	15,998		1
DMAX	153,595	98%	191,994		3
DSTDEV	44,875	36%	48,967		0.96
DSUM	677,631	721%	1,474,311		31

Named Ranges:

CritFunc	=Functions!$I$3:$M$5
DataSetFunc	=Functions!$B$8:$F$508

Data Range

loan balance	loan-to-value	value	loan-to-income code	risk history
49,878	76%	65,629	E	1
4,433	5%	88,657	A	1
52,774	76%	69,440	E	3
41,708	80%	52,135	C	3
89,999	79%	113,923	G	1
38,299	58%	66,032	F	2
43,900	80%	54,875	G	1
40,779	79%	51,619	E	3
29,734	110%	27,031	F	1
29,981	78%	38,437	C	3
34,834	95%	36,667	E	1
33,127	78%	42,470	A	2
36,260	51%	71,098	A	1
32,153	76%	42,306	C	2
16,214	76%	21,334	E	3
39,217	69%	56,836	C	1
30,888	67%	46,101	D	1
19,216	77%	24,956	A	3
27,775	95%	29,237	B	1
15,073	42%	35,887	G	1
25,083	58%	43,247	F	2
22,495	95%	23,679	F	1
25,939	78%	33,255	E	3
16,412	53%	30,966	A	2
12,047	78%	15,444	C	3
7,935	42%	18,893	G	1

Figure 1.13

The file Ch1.Data.xlsx (Functions worksheet) (Figure 1.13) shows an example of the use of these functions (named ranges have been used for the database and criteria ranges; see Chapter 2 for more information on their use).

Example: Filtering

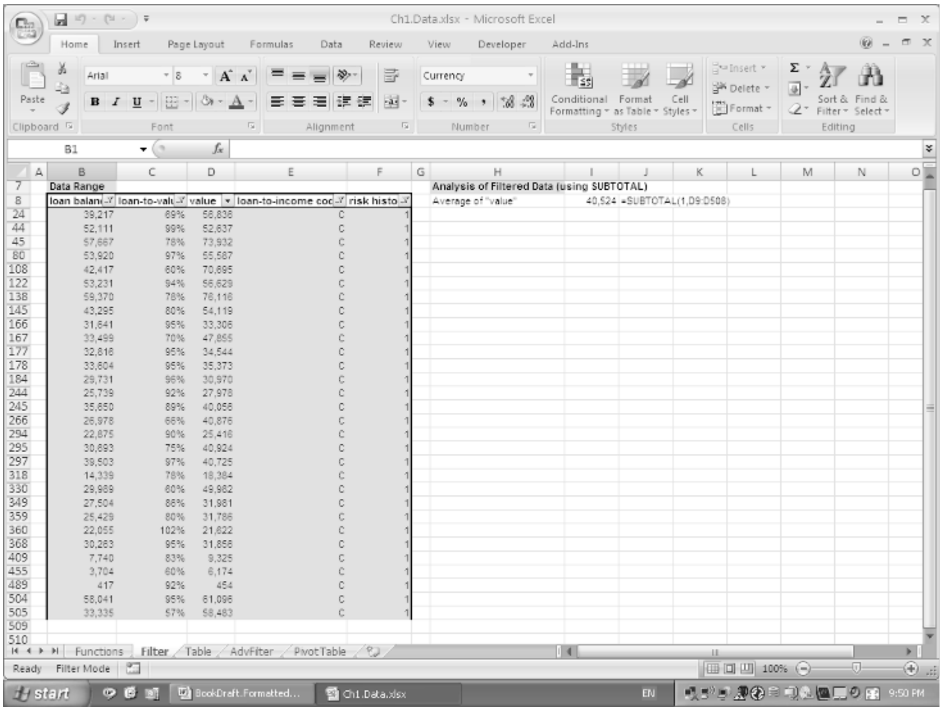
A range of data can be filtered by selecting the range including field headers (the shortcut **CTRL+SHIFT+DOWN** can be used), and applying the **Data/Filter** command (**Data/Filter/AutoFilter** in Excel 2003). A drop-down list of potential filter criteria for each field appears (in Excel 2007 these include the application of numerical or text filters); the drop-down arrows can be removed by repeating this procedure. Statistical analysis of a filtered list can be done with the **SUBTOTAL** function (see earlier). The use of **SUM**, **COUNT**, etc., will pick up those hidden points that do not meet the criteria.

The file Ch1.Data.xlsx (Filter worksheet) (Figure 1.14) shows an example.

Example: Tables in Excel 2007

In Excel 2007, a range can also be turned into a **Table** by selecting it (including field headers) and using **Home/Format as Table** (or **Insert/Table**); a **Table** can be converted back to a range by clicking on the **Table** and using **Table Tools/Convert to Range**.

The file Ch1.Data.xlsx (Table worksheet) (Figure 1.15) shows an example, once again using the **SUBTOTAL** function on the filtered data.



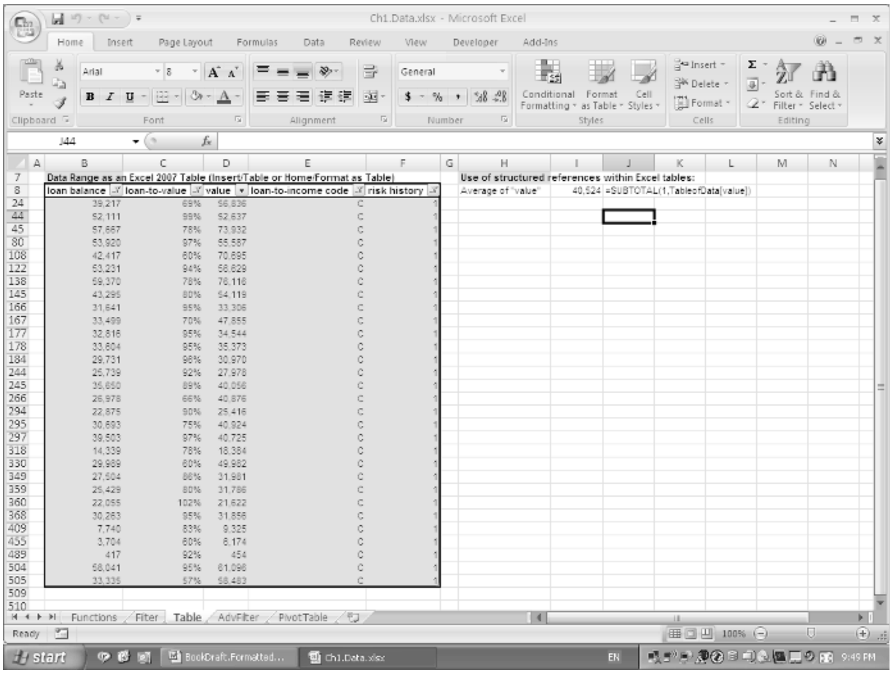
Data Range

	loan balan	loan-to-valu	value	loan-to-income coc	risk histo
24	39,217	69%	56,836	C	1
44	52,111	99%	52,637	C	1
45	57,667	78%	73,932	C	1
80	53,920	97%	55,587	C	1
108	42,417	60%	70,695	C	1
122	53,231	94%	56,629	C	1
138	59,370	78%	76,116	C	1
145	43,295	80%	54,119	C	1
166	31,641	95%	33,306	C	1
167	33,499	70%	47,855	C	1
177	32,816	95%	34,544	C	1
178	33,604	95%	35,373	C	1
184	29,731	96%	30,970	C	1
244	25,739	92%	27,978	C	1
245	35,650	89%	40,056	C	1
266	26,978	66%	40,876	C	1
294	22,875	90%	25,416	C	1
295	30,693	75%	40,924	C	1
297	39,503	97%	40,725	C	1
318	14,339	78%	18,384	C	1
330	29,989	60%	49,982	C	1
349	27,504	86%	31,981	C	1
359	25,429	80%	31,786	C	1
360	22,055	102%	21,622	C	1
368	30,263	95%	31,856	C	1
409	7,740	83%	9,325	C	1
455	3,704	60%	6,174	C	1
489	417	92%	454	C	1
504	58,041	95%	61,096	C	1
505	33,335	57%	58,483	C	1

Analysis of Filtered Data (using SUBTOTAL)

Average of "value"	40,524	=SUBTOTAL(1,D9:D508)

Figure 1.14



Data Range as an Excel 2007 Table (Insert/Table or Home/Format as Table)

	loan balance	loan-to-value	value	loan-to-income code	risk history
24	39,217	69%	56,836	C	1
44	52,111	99%	52,637	C	1
45	57,667	78%	73,932	C	1
80	53,920	97%	55,587	C	1
108	42,417	60%	70,695	C	1
122	53,231	94%	56,629	C	1
138	59,370	78%	76,116	C	1
145	43,295	80%	54,119	C	1
166	31,641	95%	33,306	C	1
167	33,499	70%	47,855	C	1
177	32,816	95%	34,544	C	1
178	33,604	95%	35,373	C	1
184	29,731	96%	30,970	C	1
244	25,739	92%	27,978	C	1
245	35,650	89%	40,056	C	1
266	26,978	66%	40,876	C	1
294	22,875	90%	25,416	C	1
295	30,693	75%	40,924	C	1
297	39,503	97%	40,725	C	1
318	14,339	78%	18,384	C	1
330	29,989	60%	49,982	C	1
349	27,504	86%	31,981	C	1
359	25,429	80%	31,786	C	1
360	22,055	102%	21,622	C	1
368	30,263	95%	31,856	C	1
409	7,740	83%	9,325	C	1
455	3,704	60%	6,174	C	1
489	417	92%	454	C	1
504	58,041	95%	61,096	C	1
505	33,335	57%	58,483	C	1

Use of structured references within Excel tables:

Average of "value" 40,524 =SUBTOTAL(1,TableofData[value])

Figure 1.15

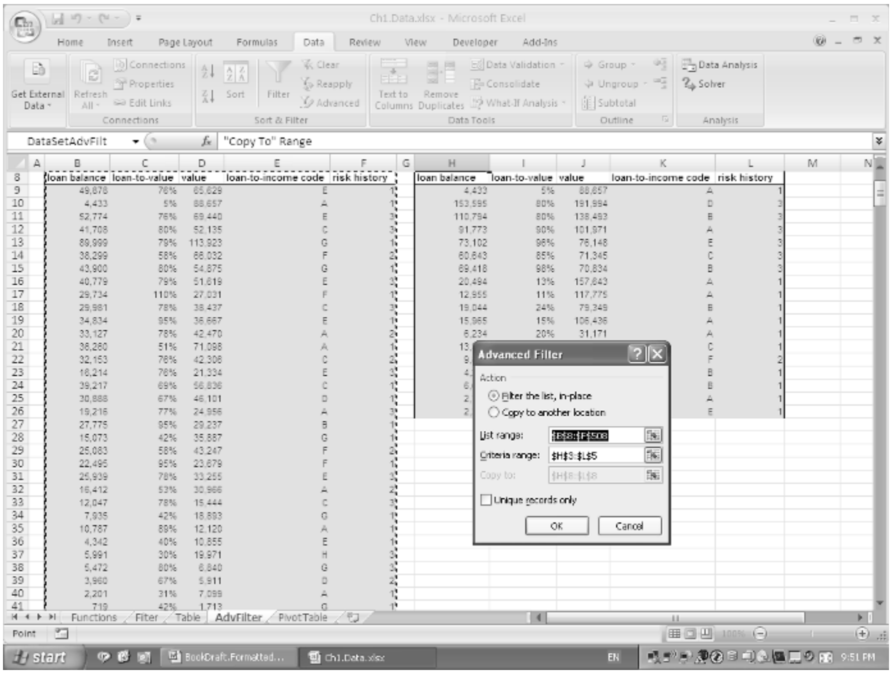
Example: Advanced Filtering

Potential disadvantages of the standard **Filter** approach include that the criteria used to filter the data set are embedded within the drop-down menus and are not explicit. Also, more complex criteria can be difficult or impossible to set up. The **Advanced Filter** (**Data/AdvancedFilter** or **Data/Filter/AdvancedFilter** in Excel 2003) can overcome these limitations, as there is an explicit criteria range, and it also allows those records that meet the criteria to be extracted to a separate area. Any named ranges that may have been set up can be accessed (for example, when completing the **List range** box) by pressing **F3**.

The file Ch1.Data.xlsx (AdvFilter worksheet) (Figure 1.16) shows an example.

Example: PivotTables

PivotTables can be used to produce cross-tabulation reports which summarise aspects of a database by category. The data range should be selected (including the field names) before using **Insert/PivotTables** (**Data/PivotTables** in Excel 2003), and then following the step-through menu ("**wizard**"), which is essentially self-explanatory. Row and column labels (including multiple labels to create subcategories) can be placed on the **PivotTable** by dragging from the field list, or right-clicking on the fields. The **PivotTable Tools** (displayed when clicking on the table) can be used to change aspects of the table, such as the **Field Settings** (which determine whether, for example, the sum or the average of the relevant entry is to be displayed), as well as using the **Refresh** button if the values in the data set change. Labels can be removed by clicking on them in the **Field List** or dragging them



DataSetAdvFilt "Copy To" Range

	loan balance	loan-to-value	value	loan-to-income code	risk history		loan balance	loan-to-value	value	loan-to-income code	risk history
9	49,878	76%	65,629	E	1		4,433	5%	88,657	A	1
10	4,433	5%	88,657	A	1		153,595	80%	191,994	D	3
11	52,774	76%	69,440	E	3		110,794	80%	138,493	B	3
12	41,708	80%	52,135	C	3		91,773	90%	101,971	A	3
13	89,999	79%	113,923	G	1		73,102	96%	76,148	E	3
14	38,299	58%	66,032	F	2		60,643	85%	71,345	C	3
15	43,900	80%	54,875	G	1		69,418	98%	70,834	B	3
16	40,779	79%	51,619	E	3		20,494	13%	157,643	A	1
17	29,734	110%	27,031	F	1		12,955	11%	117,775	A	1
18	29,981	78%	38,437	C	3		19,044	24%	79,349	B	1
19	34,834	95%	36,667	E	1		15,965	15%	106,436	A	1
20	33,127	78%	42,470	A	2		6,234	20%	31,171	A	1
21	36,260	51%	71,098	A	1		13,			C	1
22	32,153	76%	42,306	C	2		9,			F	2
23	16,214	76%	21,334	E	3		4,			B	1
24	39,217	69%	56,836	C	1		6,			B	1
25	30,888	67%	46,101	D	1		2,			A	1
26	19,216	77%	24,956	A	3		2,			E	1
27	27,775	95%	29,237	B	1						
28	15,073	42%	35,887	G	1						
29	25,083	58%	43,247	F	2						
30	22,495	95%	23,679	F	1						
31	25,939	78%	33,255	E	3						
32	16,412	53%	30,966	A	2						
33	12,047	78%	15,444	C	3						
34	7,935	42%	18,893	G	1						
35	10,787	89%	12,120	A	1						
36	4,342	40%	10,855	E	1						
37	5,991	30%	19,971	H	3						
38	5,472	80%	6,840	G	3						
39	3,960	67%	5,911	D	2						
40	2,201	31%	7,099	A	1						
41	719	42%	1,713	G	1						

Advanced Filter

Action

(●) Filter the list, in-place

( ) Copy to another location

List range: $B$8:$F$508

Criteria range: $H$3:$L$5

Copy to: $H$8:$L$8

☐ Unique records only

Figure 1.16

back from the labels area of the **Field List** window. A **PivotTable** can be deleted using the **PivotTable Tools** by choosing **Select/Entire PivotTable** and then clearing the contents. **PivotCharts** can also be produced and are essentially self-explanatory once one is familiar with **PivotTables**.

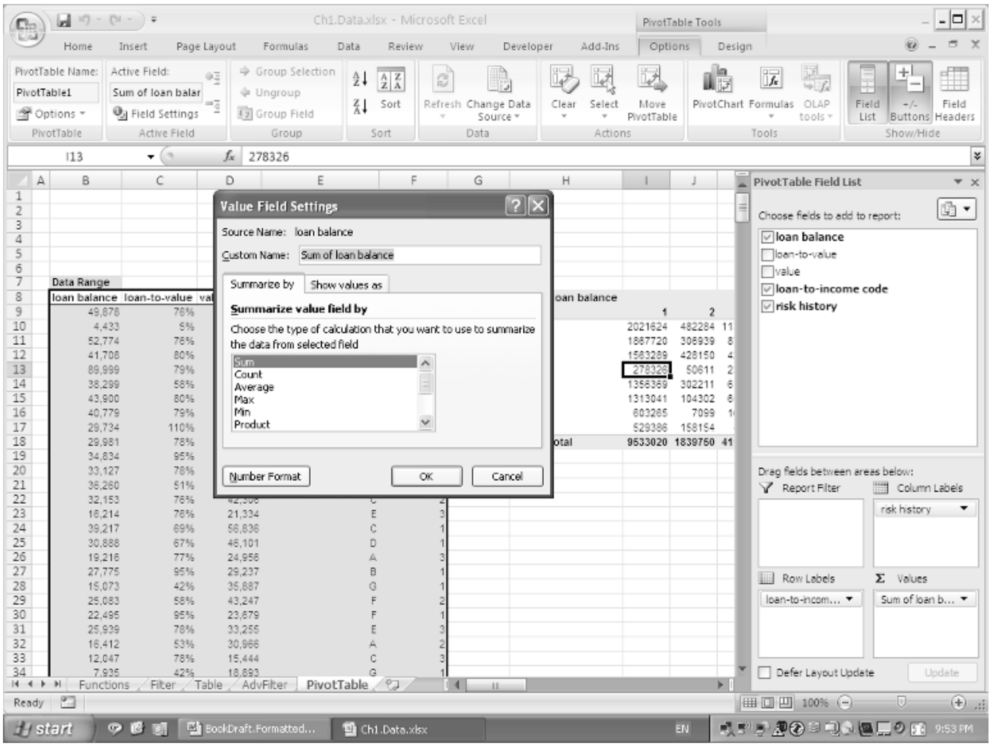
The file Ch1.Data.xlsx (PivotTable worksheet) (Figure 1.17) shows an example.

STATISTICAL FUNCTIONS

Statistical functions are often required to conduct analysis of historic data (for example, to calibrate model inputs) and to analyse the results of models (such as simulation models, where the output is typically a large data set). Basic arithmetic operations can be conducted with functions such as **AVERAGE**, **COUNT**, **MIN** and **MAX** described earlier.

Certain statistical functions relate to the variability found in data sets, such as:

- **CORREL** calculates the correlation coefficient between two data sets.
- **COVAR** calculates the covariance of two data sets (the extent to which the data sets co-vary, such as large values in one set occurring generally when large values in the other occur).
- **SLOPE** calculates the slope of the linear regression line of two data sets.
- **STDEV** calculates the standard deviation of a population based on a sample. Similarly, **VAR** calculates the variance (i.e. the square of the standard deviation). **STDEVP** and


I13 278326

Value Field Settings

Source Name: loan balance

Custom Name: Sum of loan balance

Summarize by | Show values as

Summarize value field by

Choose the type of calculation that you want to use to summarize the data from selected field

Sum
Count
Average
Max
Min
Product

Data Range

loan balance	loan-to-value	val		
49,878	76%			
4,433	5%			
52,774	76%			
41,708	80%			
89,999	79%			
38,299	58%			
43,900	80%			
40,779	79%			
29,734	110%			
29,981	78%			
34,834	95%			
33,127	78%			
36,260	51%			
32,153	76%			
16,214	76%	21,334	E	3
39,217	69%	56,836	C	1
30,888	67%	46,101	D	1
19,216	77%	24,956	A	3
27,775	95%	29,237	B	1
15,073	42%	35,887	G	1
25,083	58%	43,247	F	2
22,495	95%	23,679	F	1
25,939	78%	33,255	E	3
16,412	53%	30,966	A	2
12,047	78%	15,444	C	3
7,935	42%	18,893	G	1

	loan balance	
	1	2
	2021624	482284
	1867720	306939
	1563289	428150
	278326	50611
	1356369	302211
	1313041	104302
	603265	7099
	529386	158154
otal	9533020	1839760

PivotTable Field List

Choose fields to add to report:

☑ loan balance
☐ loan-to-value
☐ value
☑ loan-to-income code
☑ risk history

Drag fields between areas below:

Report Filter	Column Labels
	risk history

Row Labels	Values
loan-to-incom...	Sum of loan b...

☐ Defer Layout Update

Figure 1.17

VARP calculate the same figures assuming that the data provided is the whole population rather than a sample, hence requiring no correction for biases introduced by samples.

Functions that provide further statistical measures about a data set include:

- **KURT** calculates the coefficient of excess kurtosis (i.e. the figure is adjusted by subtracting 3, so that a Normal distribution would have a **KURT** of zero). The meaning of this is discussed in Chapter 4.
- **SKEW** calculates the coefficient of skewness; its meaning is also discussed in Chapter 4.

Functions that provide measures of the order of points within the data set include:

- **LARGE** and **SMALL** show the value of a point with a certain rank in the data set (e.g. the *k*th largest or smallest value).
- **RANK** calculates the rank (i.e. ordered position) of a data point within its data set.

When implementing simulation techniques (see Chapter 6), other frequently required functions include:

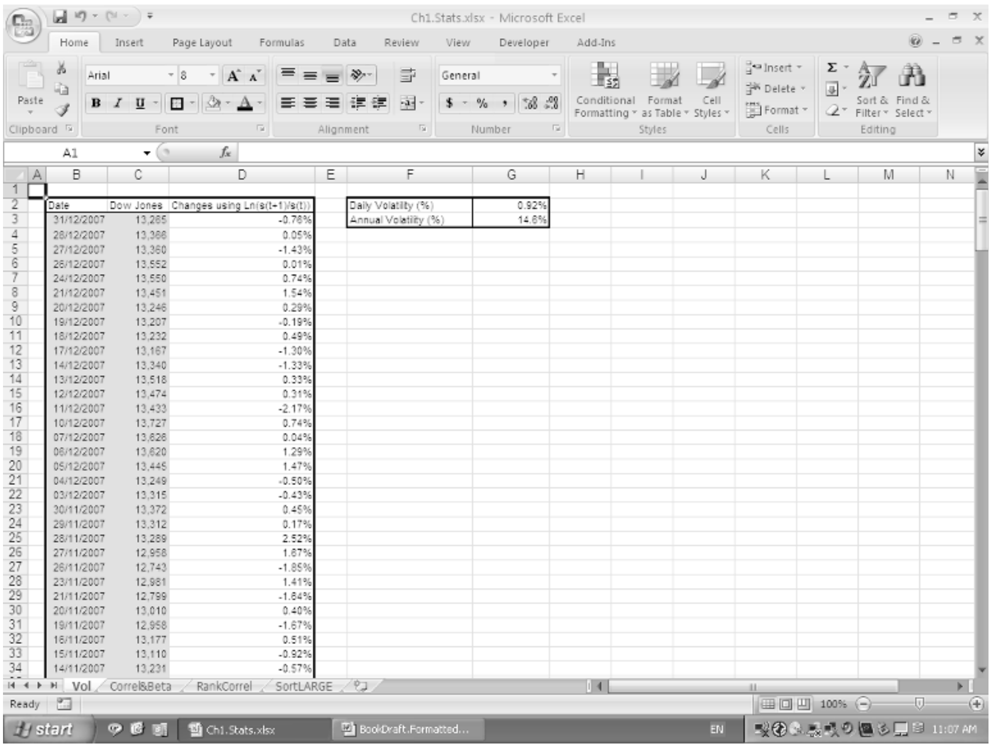
- **NORMSINV** calculates the inverse cumulative Normal distribution, i.e. requires a probability as an input and calculates the value from a standard Normal distribution that is associated with this (cumulative) probability. **NORMINV** can be used where the distribution is to have a specified mean and standard deviation (rather than the standard 0 and 1). Similarly, **LOGINV** calculates the inverse of a cumulative Lognormal distribution.

- **RAND** generates a random number uniformly distributed on [0, 1]. The combination **NORMSINV(RAND())** will therefore generate a sample from a standard Normal distribution.
- **FREQUENCY** is an array function (see later) that can be used to find the number of occurrences of data set that lie within a range, and is useful when analysing historic data and simulation results.

Example: Measuring Volatility using STDEV

The standard deviation of an asset's returns (or price changes) is a measure of its volatility.

The file Ch1.Stats.xlsx (Vol worksheet) (Figure 1.18) uses daily data for the Dow Jones index in 2007 to calculate the logarithm of the daily ratios of the index (using the **LN** function, as described earlier). The **STDEV** function is then applied to calculate the volatility of daily returns (and an annual rate is calculated by scaling by the square root of the number of trading days in the year).



Date	Dow Jones	Changes using Ln(s(t+1)/s(t))
31/12/2007	13,265	-0.76%
28/12/2007	13,366	0.05%
27/12/2007	13,360	-1.43%
26/12/2007	13,552	0.01%
24/12/2007	13,550	0.74%
21/12/2007	13,451	1.54%
20/12/2007	13,246	0.29%
19/12/2007	13,207	-0.19%
18/12/2007	13,232	0.49%
17/12/2007	13,167	-1.30%
14/12/2007	13,340	-1.33%
13/12/2007	13,518	0.33%
12/12/2007	13,474	0.31%
11/12/2007	13,433	-2.17%
10/12/2007	13,727	0.74%
07/12/2007	13,626	0.04%
06/12/2007	13,620	1.29%
05/12/2007	13,445	1.47%
04/12/2007	13,249	-0.50%
03/12/2007	13,315	-0.43%
30/11/2007	13,372	0.45%
29/11/2007	13,312	0.17%
28/11/2007	13,289	2.52%
27/11/2007	12,958	1.67%
26/11/2007	12,743	-1.85%
23/11/2007	12,981	1.41%
21/11/2007	12,799	-1.64%
20/11/2007	13,010	0.40%
19/11/2007	12,958	-1.67%
16/11/2007	13,177	0.51%
15/11/2007	13,110	-0.92%
14/11/2007	13,231	-0.57%

Daily Volatility (%)	0.92%
Annual Volatility (%)	14.6%

Figure 1.18

Example: Correlation, Covariance, and β using CORREL, COVAR, SLOPE

The measurement and use of the correlation coefficient between the returns of assets arises in many contexts, for example:

- The construction of an optimal portfolio of assets generally requires that the correlation coefficient between them be estimated.
- The estimation of the cost-of-capital for a project using the Capital Asset Pricing Model (see Chapter 3) involves the correlation coefficient either explicitly or implicitly.
- The use of correlated sampling is an important way to capture dependency relationships between variables in simulation models (see Chapter 4).

In the context of cost-of-capital calculations, the beta (β) of a project (or of a particular asset, such as a stock) can be expressed in several ways, including the covariance of the stock's returns with those of the market, or the correlation coefficient of the stock's returns with the market's, scaled by the ratio of the standard deviation of returns:

$$\beta_s = \frac{\mathrm{cov}(r_\mathrm{s}, r_\mathrm{m})}{\mathrm{cov}(r_\mathrm{m}, r_\mathrm{m})} = \frac{\rho_\mathrm{sm}\sigma_\mathrm{s}\sigma_\mathrm{m}}{\sigma_\mathrm{m}\sigma_\mathrm{m}} = \frac{\rho_\mathrm{sm}\sigma_\mathrm{s}}{\sigma_\mathrm{m}}$$

Here, r_s and r_m denote the return on the asset and market respectively, ρ_sm the correlation coefficient between the returns, and σ_s and σ_m the standard deviation of the returns.

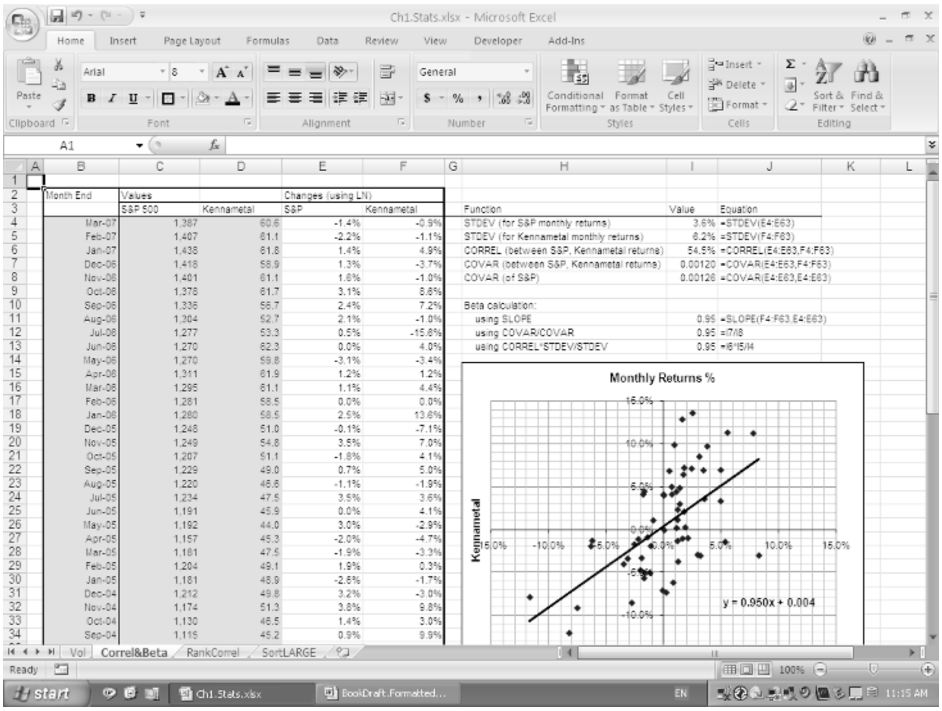
It is known from statistical theory that these expressions correspond to the slope of the regression line (where the stock's return is on the y-axis and the market's return is on the x-axis). The slope of such a line can be shown by creating a **Scatter** (or **XY) chart**, right-clicking on one of the data points to **Add Trendline**, and under **Options** selecting **Display Equation on chart**.

The file Ch1.Stats.xlsx (Correl&Beta worksheet) (Figure 1.19) shows an example. It contains the monthly (logarithmic) returns for a five-year period of the S&P500 and a NYSE-quoted company, Kennametal. The functions **CORREL**, **COVAR**, **STDEVP** and **SLOPE** are used to calculate the statistics on correlation and volatility, and the β is calculated by use of these methods, as well as being shown as the slope of the regression line.

Example: Rank Correlation using RANK and CORREL

The classical measure of correlation discussed above is known as the Pearson Product Moment (or linear) correlation. When using correlation in simulation techniques (and in some other applications), the creation of correlation between random variables usually requires a less stringent definition of correlation, and for this a rank correlation (Spearman Rank correlation) is often used. Rank correlations are based on calculating the correlation coefficient using a modified data set in which each data point is replaced by its position (or rank) within its own set (i.e. the smallest value is given a rank of 1, and so on). Variables that are 100% rank correlated will not necessarily form a linear relationship (i.e. will not in general be 100% linearly correlated), but rather form a monotonic set. In this sense the use of rank correlation allows for more flexibility in the creation and sampling of multivariate distributions.

The file Ch1.Stats.xlsx (RankCorrel worksheet) (Figure 1.20) shows the use of the **RANK** function to calculate a new data set derived from the position of each point within its own set



Month End	Values		Changes (using LN)	
	S&P 500	Kennametal	S&P	Kennametal
Mar-07	1,387	60.6	-1.4%	-0.9%
Feb-07	1,407	61.1	-2.2%	-1.1%
Jan-07	1,438	61.8	1.4%	4.9%
Dec-06	1,418	58.9	1.3%	-3.7%
Nov-06	1,401	61.1	1.6%	-1.0%
Oct-06	1,378	61.7	3.1%	8.6%
Sep-06	1,336	56.7	2.4%	7.2%
Aug-06	1,304	52.7	2.1%	-1.0%
Jul-06	1,277	53.3	0.5%	-15.6%
Jun-06	1,270	62.3	0.0%	4.0%
May-06	1,270	59.8	-3.1%	-3.4%
Apr-06	1,311	61.9	1.2%	1.2%
Mar-06	1,295	61.1	1.1%	4.4%
Feb-06	1,281	58.5	0.0%	0.0%
Jan-06	1,280	58.5	2.5%	13.6%
Dec-05	1,248	51.0	-0.1%	-7.1%
Nov-05	1,249	54.8	3.5%	7.0%
Oct-05	1,207	51.1	-1.8%	4.1%
Sep-05	1,229	49.0	0.7%	5.0%
Aug-05	1,220	46.6	-1.1%	-1.9%
Jul-05	1,234	47.5	3.5%	3.6%
Jun-05	1,191	45.9	0.0%	4.1%
May-05	1,192	44.0	3.0%	-2.9%
Apr-05	1,157	45.3	-2.0%	-4.7%
Mar-05	1,181	47.5	-1.9%	-3.3%
Feb-05	1,204	49.1	1.9%	0.3%
Jan-05	1,181	48.9	-2.6%	-1.7%
Dec-04	1,212	49.8	3.2%	-3.0%
Nov-04	1,174	51.3	3.8%	9.8%
Oct-04	1,130	46.5	1.4%	3.0%
Sep-04	1,115	45.2	0.9%	9.9%

Function	Value	Equation
STDEV (for S&P monthly returns)	3.6%	=STDEV(E4:E63)
STDEV (for Kennametal monthly returns)	6.2%	=STDEV(F4:F63)
CORREL (between S&P, Kennametal returns)	54.5%	=CORREL(E4:E63,F4:F63)
COVAR (between S&P, Kennametal returns)	0.00120	=COVAR(E4:E63,F4:F63)
COVAR (of S&P)	0.00126	=COVAR(E4:E63,E4:E63)
Beta calculation:		
using SLOPE	0.95	=SLOPE(F4:F63,E4:E63)
using COVAR/COVAR	0.95	=I7/I8
using CORREL*STDEV/STDEV	0.95	=I6*I5/I4

Figure 1.19

(when ordered in ascending order). The **CORREL** function is applied to both sets of data. There is a slight difference in the correlation coefficients calculated by the two methods. (Note that Excel also has a function **PEARSON**, which is equivalent to **CORREL**.)

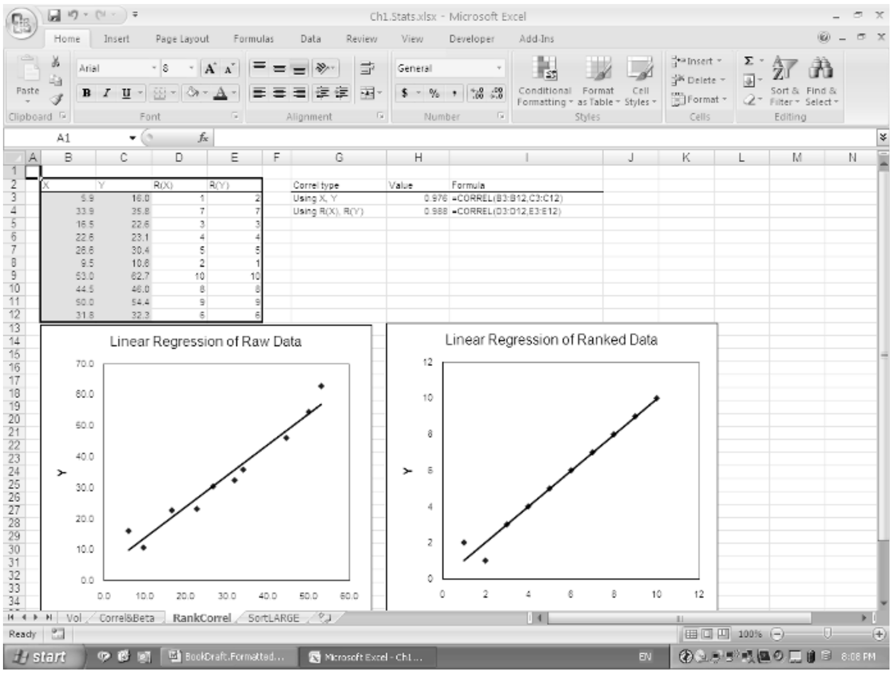
Example: Automatic Sorting of Data using LARGE

The **LARGE** function can be used to automate the sorting of a data set into ascending or descending order. An advantage over the use of **Data/Sort** is that the sorted list is lived-linked to the data set and so will update automatically if it changes (e.g. as new daily data comes in).

The file Ch1.Stats.xlsx (SortLARGE worksheet) (Figure 1.21) shows an example in which a data set is sorted by applying **LARGE**, where a set of integers from 1 upwards is used in order to place the points in descending order. As covered in the next section, to find the date on which any of these specific returns occurred, **Lookup** functions (such as the combination of **INDEX** with **MATCH** or **VLOOKUP** if the data set were re-ordered appropriately) can be used.

LOOKUP AND REFERENCE FUNCTIONS

Many Excel users have only a cursory awareness of **Lookup** and **Reference** functions. However a good knowledge of them is arguably one of the single most important capabilities required to construct intermediate and advanced models.



X	Y	R(X)	R(Y)
5.9	16.0	1	2
33.9	35.8	7	7
16.5	22.6	3	3
22.6	23.1	4	4
26.6	30.4	5	5
9.5	10.8	2	1
53.0	62.7	10	10
44.5	46.0	8	8
50.0	54.4	9	9
31.8	32.3	6	6

Correl type	Value	Formula
Using X, Y	0.976	=CORREL(B3:B12,C3:C12)
Using R(X), R(Y)	0.988	=CORREL(D3:D12,E3:E12)

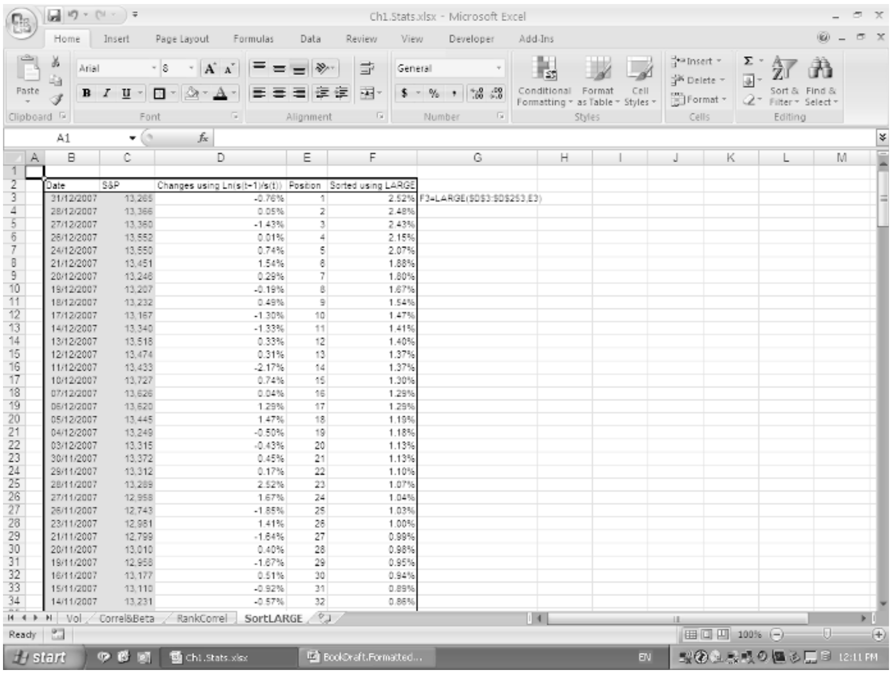
Ch1.Stats.xlsx - Microsoft Excel
Linear Regression of Raw Data
Linear Regression of Ranked Data
Y
Vol
Correl&Beta
RankCorrel
SortLARGE

Figure 1.20

Some frequently used examples are:

- **MATCH** finds the relative position of a specified value in an array.
- **HLOOKUP** (**VLOOKUP**) searches the top row (left column) of a table for a specified value and finds the column in that row (row in that column) that contains that value. It then provides the value that is at a specified row (column) within the table.
- **INDEX** looks up the value in a specified row and column of a matrix. The function also exists in a reference form, where it returns a reference to specified cells rather than to the value of a cell.
- **OFFSET** provides the value in a cell that is a specified number of rows and columns from a specified reference cell or range. It can also be used to return the values in a range of cells (rather than an individual cell) that is a specified number of rows and columns from a certain reference cell or range.
- **CHOOSE** uses one of a set of values according to some key, and can be used where the arguments are in a non-contiguous range. In some cases, the function can provide more flexibility than the other **Lookup** functions.
- **INDIRECT** returns the reference specified by a text string.

There are other functions that may on occasion be useful, such as **COLUMN** (or **ROW**) to find the column (row) number of the given reference, and **COLUMNS** or (**ROWS**) to find the number of columns (rows) of a reference range.



Date	S&P	Changes using Ln(s(t-1)/s(t))	Position	Sorted using LARGE
31/12/2007	13,265	-0.76%	1	2.52%
28/12/2007	13,366	0.05%	2	2.48%
27/12/2007	13,360	-1.43%	3	2.43%
26/12/2007	13,552	0.01%	4	2.15%
24/12/2007	13,550	0.74%	5	2.07%
21/12/2007	13,451	1.54%	6	1.88%
20/12/2007	13,246	0.29%	7	1.80%
19/12/2007	13,207	-0.19%	8	1.67%
18/12/2007	13,232	0.49%	9	1.54%
17/12/2007	13,167	-1.30%	10	1.47%
14/12/2007	13,340	-1.33%	11	1.41%
13/12/2007	13,518	0.33%	12	1.40%
12/12/2007	13,474	0.31%	13	1.37%
11/12/2007	13,433	-2.17%	14	1.37%
10/12/2007	13,727	0.74%	15	1.30%
07/12/2007	13,626	0.04%	16	1.29%
06/12/2007	13,620	1.29%	17	1.29%
05/12/2007	13,445	1.47%	18	1.19%
04/12/2007	13,249	-0.50%	19	1.18%
03/12/2007	13,315	-0.43%	20	1.13%
30/11/2007	13,372	0.45%	21	1.13%
29/11/2007	13,312	0.17%	22	1.10%
28/11/2007	13,289	2.52%	23	1.07%
27/11/2007	12,958	1.67%	24	1.04%
26/11/2007	12,743	-1.85%	25	1.03%
23/11/2007	12,981	1.41%	26	1.00%
21/11/2007	12,799	-1.64%	27	0.99%
20/11/2007	13,010	0.40%	28	0.98%
19/11/2007	12,958	-1.67%	29	0.95%
16/11/2007	13,177	0.51%	30	0.94%
15/11/2007	13,110	-0.92%	31	0.89%
14/11/2007	13,231	-0.57%	32	0.86%

F3=LARGE($D$3:$D$253,E3)

Figure 1.21

The following provides some simple examples of uses of these functions. In many cases, there is not a unique way to achieve a particular purpose, so that alternative formulations with different functions may exist.

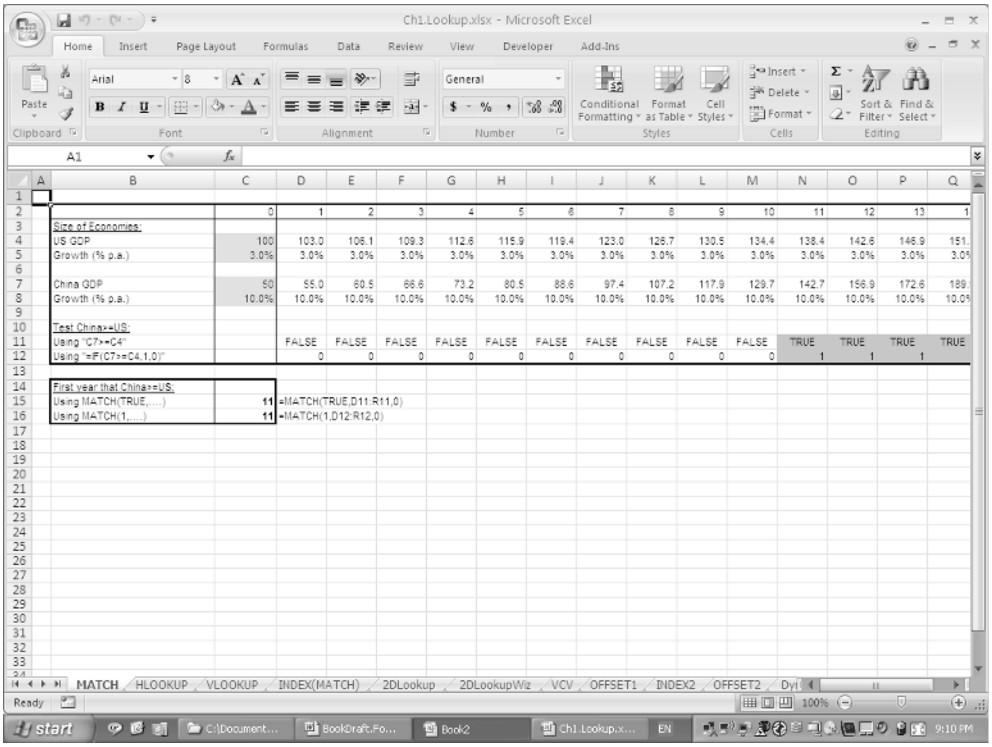
Example: Finding Equivalent Values using MATCH

The file Ch1.Lookup.xlsx (Match worksheet) (Figure 1.22) uses the earlier hypothesised development of the US and Chinese economies, and aims to find the first year in which the GDP of China will be larger than that of the US. (For ease of presentation the example uses conditional formatting as discussed in Chapter 2 to colour the cells when this is the case.)

The **MATCH** function is applied to the range containing the results of the GDP comparison, to find the relevant position in the range. (Depending on whether the logical test is set to return **TRUE/FALSE** or 1/0, the argument used in the **MATCH** function would need to be changed appropriately.) In this example, the GDP of China would be larger than that of the US for the first time in Year 11. Note that the **MATCH** function has an optional argument, which is the type of match required, with zero representing an exact match (see Excel **Help** or **F1** on the function for more details).

Example: Volume Discounts using HLOOKUP

The file Ch1.Lookup.xlsx (HLOOKUP worksheet) (Figure 1.23) uses the **HLOOKUP** function to find, for a specified quantity of a product purchased, the discount that applies according to some schedule. Note that the last parameter of the function is set to **TRUE**



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2			0	1	2	3	4	5	6	7	8	9	10	11	12	13	1
3		Size of Economies:															
4		US GDP	100	103.0	106.1	109.3	112.6	115.9	119.4	123.0	126.7	130.5	134.4	138.4	142.6	146.9	151.
5		Growth (% p.a.)	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0
6																	
7		China GDP	50	55.0	60.5	66.6	73.2	80.5	88.6	97.4	107.2	117.9	129.7	142.7	156.9	172.6	189.
8		Growth (% p.a.)	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0
9																	
10		Test China>=US:															
11		Using "C7>=C4"		FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
12		Using "=IF(C7>=C4,1,0)"		0	0	0	0	0	0	0	0	0	0	1	1	1	
13																	
14		First year that China>=US:															
15		Using MATCH(TRUE,.....)	**11**	=MATCH(TRUE,D11:R11,0)													
16		Using MATCH(1,.....)	**11**	=MATCH(1,D12:R12,0)													

Figure 1.22

(or 1), so that the closest match in ascending order is found (rather than an exact match); this is necessary here, as the volume discounts must apply to all intermediate points in the table, i.e. the discount that applies to a quantity of 7 should be the same as that which applies to any number from 5 to (just less than) 10.

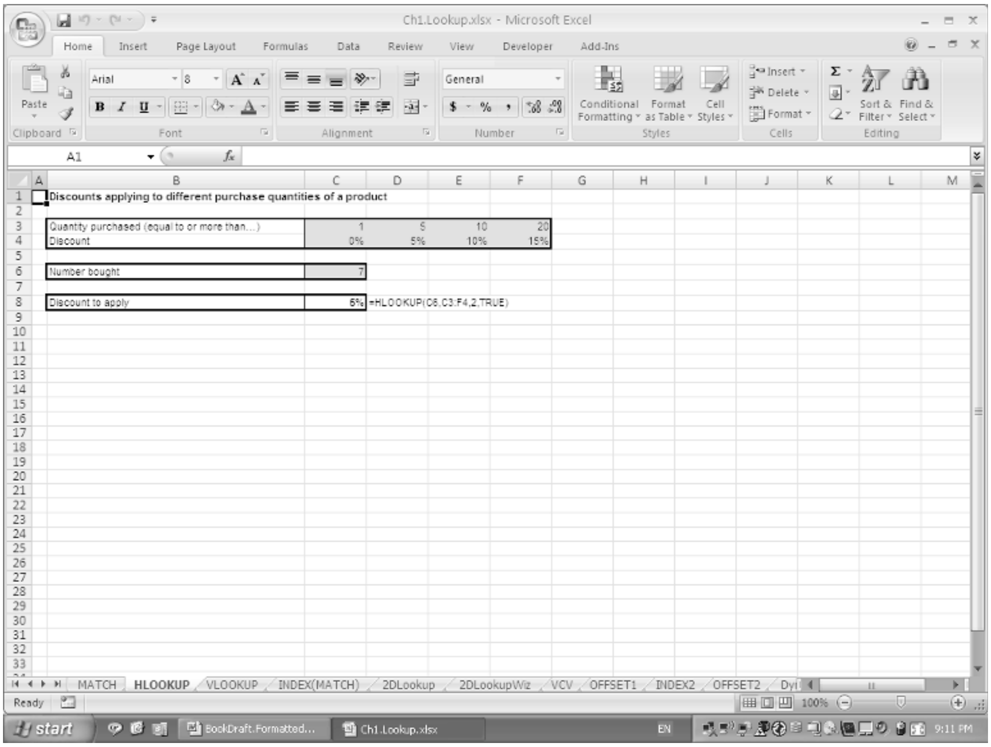
Example: Currency Conversion I–using VLOOKUP

The items in a multi-currency database can be converted into a common currency by looking up the appropriate exchange-rate for each item.

The file Ch1.Lookup.xlsx (VLOOKUP worksheet) (Figure 1.24) shows how the **VLOOKUP** can be used to find the appropriate row for each currency, and then to pick out the value of the exchange rate, which is then used as an input to a regular Excel formula to convert the local currency amount to the sterling equivalent. The application of this function assumes that the data is arranged vertically, and that the left column contains the currency names.

Example: Currency Conversion II–using INDEX(MATCH)

One of the constraints on using the **H-** and **VLOOKUP** functions is that the contiguous range in which the data is to be looked up must be arranged so that the top row (or left column) contains the items to be searched. On occasion this can be inconvenient, and a



	A	B	C	D	E	F
1		**Discounts applying to different purchase quantities of a product**				
2						
3		Quantity purchased (equal to or more than...)	1	5	10	20
4		Discount	0%	5%	10%	15%
5						
6		Number bought	7			
7						
8		Discount to apply	5%	=HLOOKUP(C6,C3:F4,2,TRUE)		

Figure 1.23

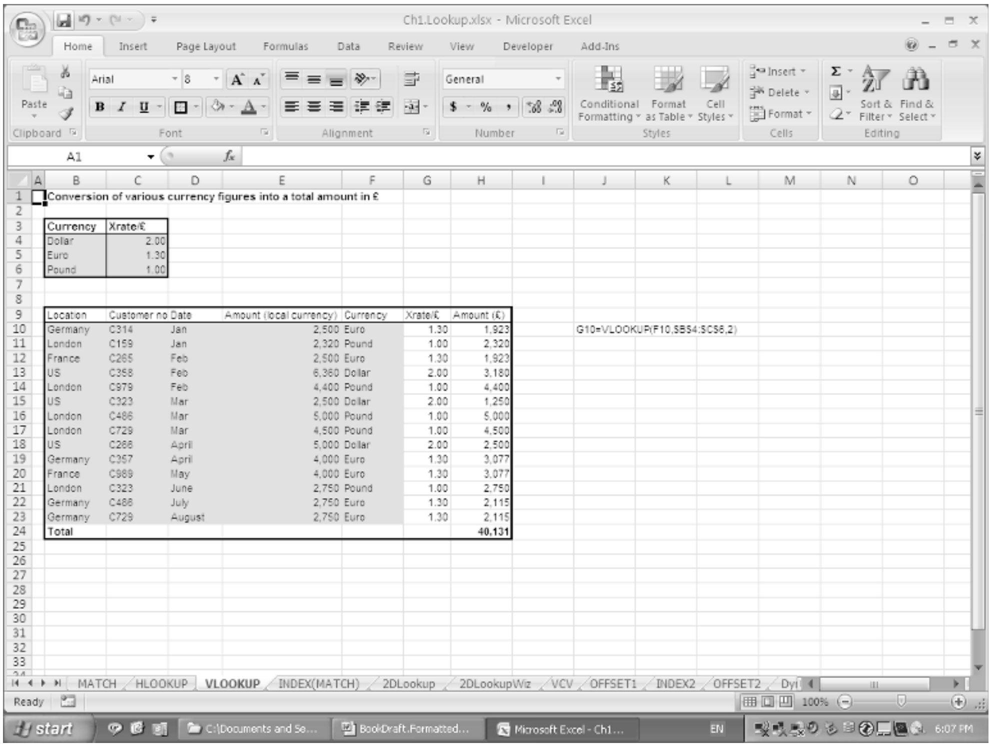
more flexible approach to the same problem can be implemented using a combination of the **INDEX** and **MATCH** functions.

The file Ch1.Lookup.xlsx (INDEX(MATCH) worksheet) (Figure 1.25) shows the currency conversion example used above, but as the lookup field is in the right-hand column, the **VLOOKUP** function cannot be used. Instead, the **MATCH** function is used to determine the correct row to lookup the value and the **INDEX** function used to lookup the exchange rate. (The example shows this procedure performed step-by-step and also using a compound formula. As described in Chapter 2, a robust way to build the compound formulae (where desired) is to create the step-by-step process and paste a copy of the **MATCH** function from within the **Formula Bar** in place of the reference to that cell within the **INDEX** formula.)

Example: Two-Dimensional Look-Up using INDEX(MATCH)

The **INDEX(MATCH)** combination can also be used to perform a two-dimensional lookup. For example, a volume discount may also be product-specific.

The file Ch1.Lookup.xlsx (2DLookup worksheet) (Figure 1.26) shows the use of the **MATCH** function to lookup the relevant row and column number, which are then used as arguments of the **INDEX** function to calculate the discount that applies according both to the product and volume purchased. In this case an exact **MATCH** is required when determining the row number (match type = 0), but a less-than-or-equal match is necessary for the column number (match type = 1).



Conversion of various currency figures into a total amount in £

Currency	Xrate/£
Dollar	2.00
Euro	1.30
Pound	1.00

Location	Customer no	Date	Amount (local currency)	Currency	Xrate/£	Amount (£)
Germany	C314	Jan	2,500	Euro	1.30	1,923
London	C159	Jan	2,320	Pound	1.00	2,320
France	C265	Feb	2,500	Euro	1.30	1,923
US	C358	Feb	6,360	Dollar	2.00	3,180
London	C979	Feb	4,400	Pound	1.00	4,400
US	C323	Mar	2,500	Dollar	2.00	1,250
London	C486	Mar	5,000	Pound	1.00	5,000
London	C729	Mar	4,500	Pound	1.00	4,500
US	C266	April	5,000	Dollar	2.00	2,500
Germany	C357	April	4,000	Euro	1.30	3,077
France	C989	May	4,000	Euro	1.30	3,077
London	C323	June	2,750	Pound	1.00	2,750
Germany	C486	July	2,750	Euro	1.30	2,115
Germany	C729	August	2,750	Euro	1.30	2,115
Total						40,131

`G10=VLOOKUP(F10,$B$4:$C$6,2)`

Ch1.Lookup.xlsx - Microsoft Excel
Home Insert Page Layout Formulas Data Review View Developer Add-Ins
A1
MATCH HLOOKUP VLOOKUP INDEX(MATCH) 2DLookup 2DLookupWiz VCV OFFSET1 INDEX2 OFFSET2
Ready

Figure 1.24

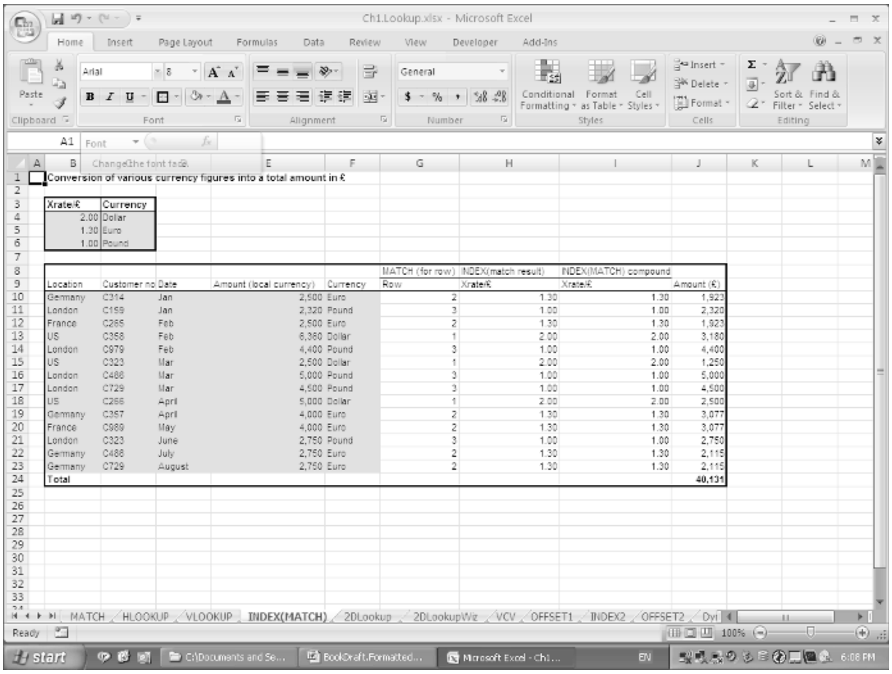
In the (2DLookupWiz worksheet) a similar result can be generated in an essentially automated way by the use of the **Lookup** wizard. This is an add-in that can be loaded under **Office/Excel Options/Add-ins** (or **Tools**/**Add-ins** in Excel 2003), with the menu then appearing under **Formulas/Solutions** (or **Tools** in Excel 2003). The use of the wizard is essentially self-explanatory once it is loaded, and is also similar to the use of the **Conditional Sum Wizard** that is discussed later in this chapter.

Example: Variance–Covariance Matrices using H- and VLOOKUP

When considering a portfolio of assets with given weights and assumed expected returns, volatilities (standard deviations) and correlations of returns, the portfolio's expected return can be calculated using the **SUMPRODUCT** function (as shown earlier). However, the calculation of the portfolio's standard deviation (as the square root of its variance) requires matrix multiplication involving the vector of portfolio weights (w) and the variance–covariance matrix (VCV):

$$\text{Variance of portfolio} = w\,VCV\,w^t$$

The implementation of such matrix multiplication is discussed later in this chapter (using the **MMULT** function). Here we assume that a correlation matrix has been calculated (either using the earlier method using **CORREL**, or–as shown later–using the **OFFSET** or other



Conversion of various currency figures into a total amount in €

Xrate/€	Currency
2.00	Dollar
1.30	Euro
1.00	Pound

					MATCH (for row)	INDEX(match result)	INDEX(MATCH) compound	
Location	Customer no	Date	Amount (local currency)	Currency	Row	Xrate/€	Xrate/€	Amount (€)
Germany	C314	Jan	2,500	Euro	2	1.30	1.30	1,923
London	C159	Jan	2,320	Pound	3	1.00	1.00	2,320
France	C265	Feb	2,500	Euro	2	1.30	1.30	1,923
US	C358	Feb	6,360	Dollar	1	2.00	2.00	3,180
London	C979	Feb	4,400	Pound	3	1.00	1.00	4,400
US	C323	Mar	2,500	Dollar	1	2.00	2.00	1,250
London	C486	Mar	5,000	Pound	3	1.00	1.00	5,000
London	C729	Mar	4,500	Pound	3	1.00	1.00	4,500
US	C266	April	5,000	Dollar	1	2.00	2.00	2,500
Germany	C357	April	4,000	Euro	2	1.30	1.30	3,077
France	C989	May	4,000	Euro	2	1.30	1.30	3,077
London	C323	June	2,750	Pound	3	1.00	1.00	2,750
Germany	C486	July	2,750	Euro	2	1.30	1.30	2,115
Germany	C729	August	2,750	Euro	2	1.30	1.30	2,115
Total								40,131

Figure 1.25

Lookup functions to create a formula that can be copied to all cells of the correlation matrix), and are concerned with the calculation of the *VCV* matrix.

The elements of the *VCV* matrix are given by:

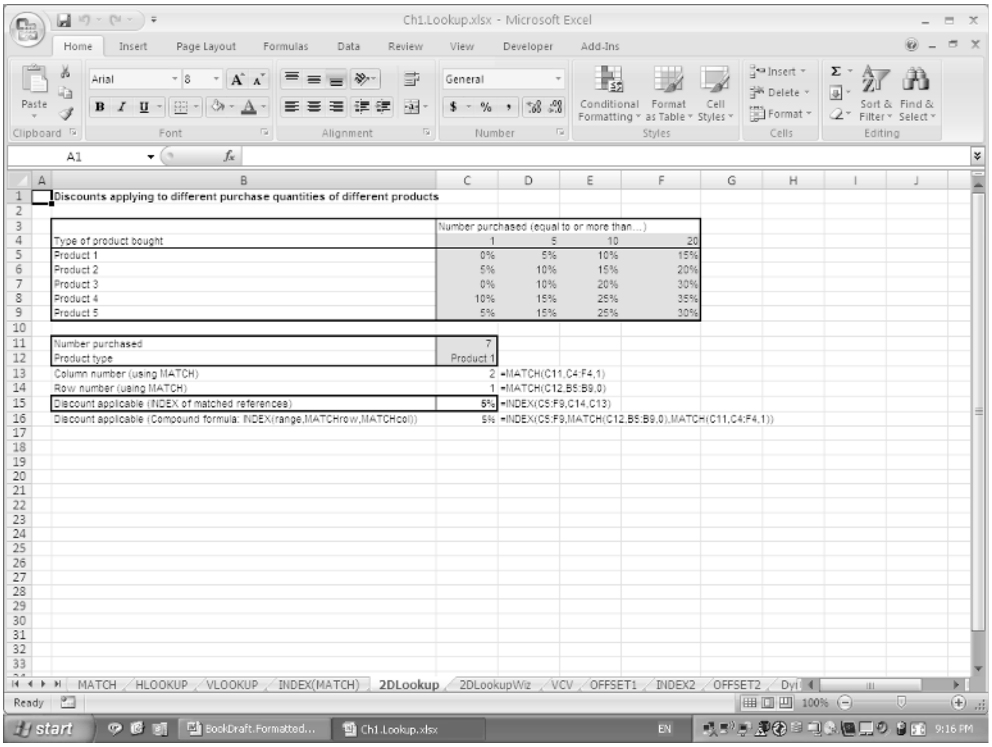
$$VCV_{ij} = \rho_{ij}\sigma_i\sigma_j$$

where ρ_{ij} represents the correlation coefficient between the return of assets i and j, and σ_i the standard deviation of asset i.

The file Ch1.Lookup.xlsx (VCV worksheet) (Figure 1.27) shows the use of **HLOOKUP** (**VLOOKUP** could be used if the standard deviation data were arranged in a column) to create a formula in one cell of the variance–covariance matrix that can be copied to all other cells in the matrix. Especially for large matrices such an approach would be much less time-consuming than one using standard cell references, as such formulae would need to be set up individually for each cell in the matrix in accordance with the required data sets that need to be referenced.

Example: Time Axis as a Variable using OFFSET or INDEX

Lookup functions can also be used to create situations in which the time-axis of a model is variable, at least in a discrete sense (for example, that each column represents a year, but precisely which year it represents is flexible or can be determined by the user). For example, if the time-line of part of a project is altered (e.g. due to a delay) then it would generally be very cumbersome to change the formula links in the model, especially where different



Discounts applying to different purchase quantities of different products

	Number purchased (equal to or more than…)			
Type of product bought	1	5	10	20
Product 1	0%	5%	10%	15%
Product 2	5%	10%	15%	20%
Product 3	0%	10%	20%	30%
Product 4	10%	15%	25%	35%
Product 5	5%	15%	25%	30%

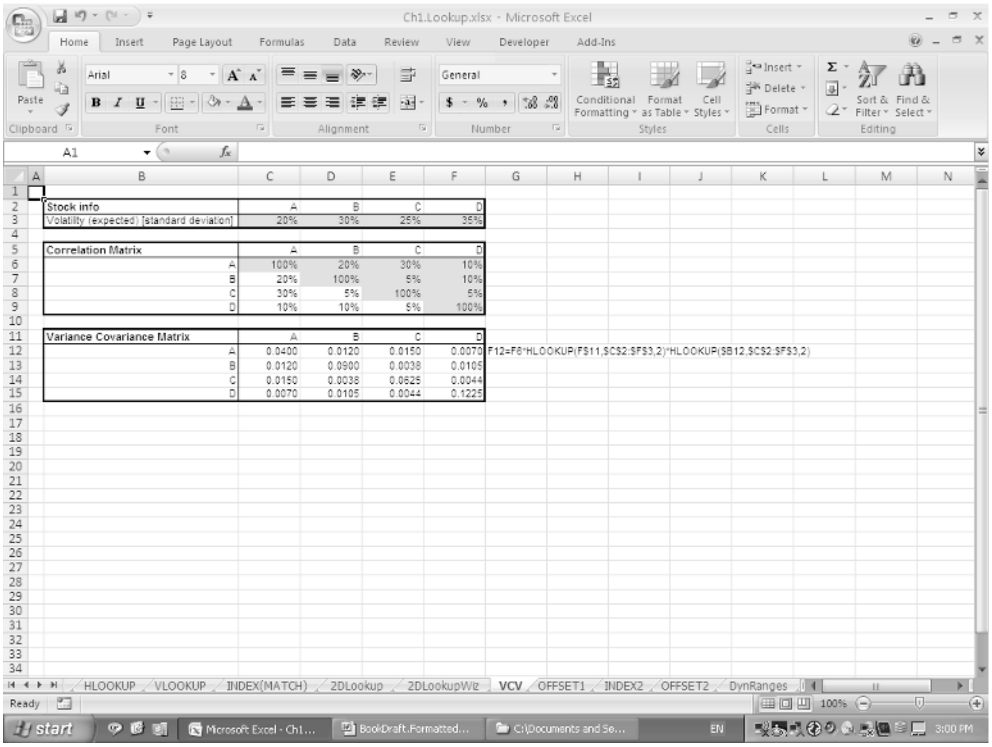
Number purchased	7	
Product type	Product 1	
Column number (using MATCH)	2	=MATCH(C11,C4:F4,1)
Row number (using MATCH)	1	=MATCH(C12,B5:B9,0)
Discount applicable (INDEX of matched references)	5%	=INDEX(C5:F9,C14,C13)
Discount applicable (Compound formula: INDEX(range,MATCHrow,MATCHcol))	5%	=INDEX(C5:F9,MATCH(C12,B5:B9,0),MATCH(C11,C4:F4,1))

Figure 1.26

variables or parts of a model may have their own separate time-lines. Example applications include:

- The modelling of revenues (or profit or cash flow) that may arise from a portfolio of projects (e.g. in drug development) when the expected launch date of each project may need to be updated as more information becomes available.
- The base case cost for a project where parts of the project could be delayed by different amounts.
- The calculation of a depreciation schedule for a book of assets purchased at different times, where the depreciation profile of each asset may be different.
- The cash repayment profile for a portfolio of bonds or other commitments where one initially set up such a portfolio (in terms of the timing of the repayment of individual elements) so that it is optimised in some way (e.g. the maximum repayment is held below some level or is matched as closely as possible with cash inflow profile of the business). In this case one would wish to create a model in which the repayment timing on each element of the portfolio can be varied.

The file Ch1.Lookup.xlsx (OFFSET1 worksheet) (Figure 1.28) shows the use of the **OFFSET** function to translate a generic schedule of bond repayments after the year of issue of each into a specific schedule once the issue date is decided. One can then experiment



Stock info	A	B	C	D
Volatility (expected) [standard deviation]	20%	30%	25%	35%

Correlation Matrix	A	B	C	D
A	100%	20%	30%	10%
B	20%	100%	5%	10%
C	30%	5%	100%	5%
D	10%	10%	5%	100%

Variance Covariance Matrix	A	B	C	D
A	0.0400	0.0120	0.0150	0.0070
B	0.0120	0.0900	0.0038	0.0105
C	0.0150	0.0038	0.0625	0.0044
D	0.0070	0.0105	0.0044	0.1225

F12=F6*HLOOKUP(F$11,$C$2:$F$3,2)*HLOOKUP($B12,$C$2:$F$3,2)

Figure 1.27

with the issue date (and the other input variables) to select a suitable repayment profile. Note that this could also be implemented using the **INDEX** function. The screen shot for the INDEX2 worksheet should be indentical to that for the OFFSET1 worksheet.

Example: Transposing References using OFFSET

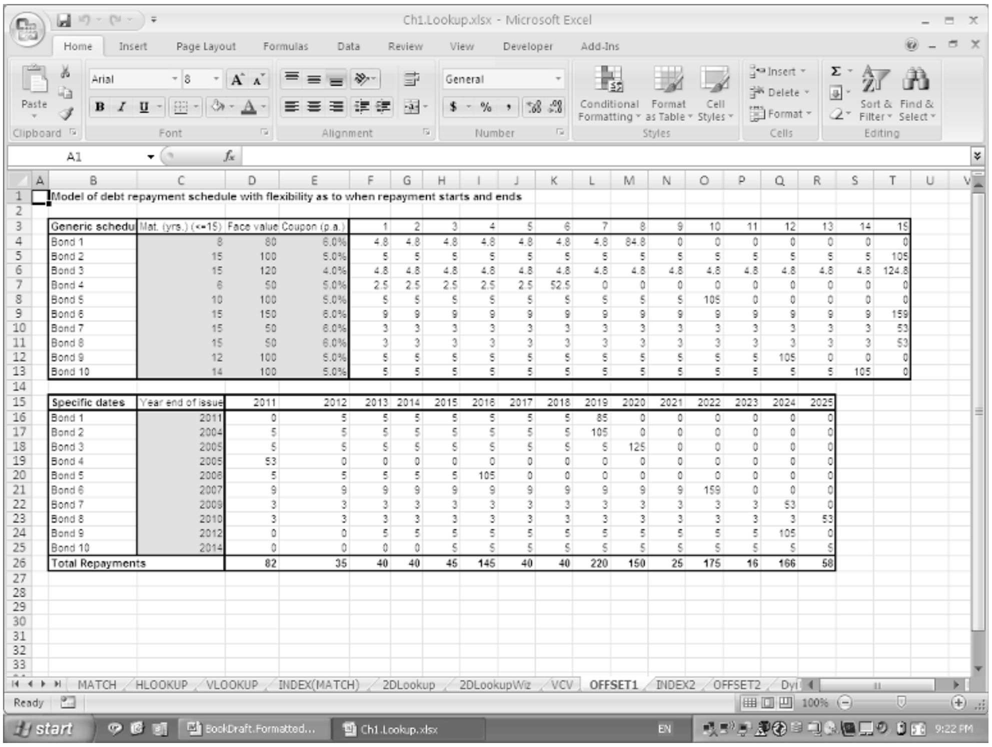
The file Ch1.Lookup.xlsx (OFFSET2 worksheet) (Figure 1.29) shows an example of the use of the **OFFSET** function to transpose data or formula references.

Example: Flexible Ranges using INDEX, OFFSET and INDIRECT

The **OFFSET**, **INDEX** and **INDIRECT** functions can each be used to create formulae that refer to ranges that are flexible (i.e. whose size varies and automatically adjusts).

The **OFFSET** function can be used to return the values in a range (rather than just the contents of an individual cell). The optional height and width arguments of the function can be used to specify the size of the output range (where these optional arguments are omitted, the size of the output range is assumed to be the same as that of the reference range, which in the examples above was only a single cell, but in general need not be).

The **INDEX** function exists in a reference form, in other words it returns a reference to specified cells rather than the value of a cell. When used in this form, the range referred to by the function will generally be directly used as a range argument in another function.



Model of debt repayment schedule with flexibility as to when repayment starts and ends

Generic schedu	Mat. (yrs.) (<=15)	Face value	Coupon (p.a.)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bond 1	8	80	6.0%	4.8	4.8	4.8	4.8	4.8	4.8	4.8	84.8	0	0	0	0	0	0	0
Bond 2	15	100	5.0%	5	5	5	5	5	5	5	5	5	5	5	5	5	5	105
Bond 3	15	120	4.0%	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	124.8
Bond 4	6	50	5.0%	2.5	2.5	2.5	2.5	2.5	52.5	0	0	0	0	0	0	0	0	0
Bond 5	10	100	5.0%	5	5	5	5	5	5	5	5	5	105	0	0	0	0	0
Bond 6	15	150	6.0%	9	9	9	9	9	9	9	9	9	9	9	9	9	9	159
Bond 7	15	50	6.0%	3	3	3	3	3	3	3	3	3	3	3	3	3	3	53
Bond 8	15	50	6.0%	3	3	3	3	3	3	3	3	3	3	3	3	3	3	53
Bond 9	12	100	5.0%	5	5	5	5	5	5	5	5	5	5	5	105	0	0	0
Bond 10	14	100	5.0%	5	5	5	5	5	5	5	5	5	5	5	5	5	105	0

Specific dates	Year end of issue	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
Bond 1	2011	0	5	5	5	5	5	5	5	85	0	0	0	0	0	0
Bond 2	2004	5	5	5	5	5	5	5	5	105	0	0	0	0	0	0
Bond 3	2005	5	5	5	5	5	5	5	5	5	125	0	0	0	0	0
Bond 4	2005	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bond 5	2006	5	5	5	5	5	105	0	0	0	0	0	0	0	0	0
Bond 6	2007	9	9	9	9	9	9	9	9	9	9	9	159	0	0	0
Bond 7	2009	3	3	3	3	3	3	3	3	3	3	3	3	3	53	0
Bond 8	2010	3	3	3	3	3	3	3	3	3	3	3	3	3	3	53
Bond 9	2012	0	0	5	5	5	5	5	5	5	5	5	5	5	105	0
Bond 10	2014	0	0	0	0	5	5	5	5	5	5	5	5	5	5	5
Total Repayments		82	35	40	40	45	145	40	40	220	150	25	175	16	166	58

Figure 1.28

The **INDIRECT** function has a cell reference as its argument and returns the value that is in that cell.

The file Ch1.Lookup.xlsx (DynRanges worksheet) (Figure 1.30) shows the use of each of these to create a formula that calculates the moving average of points in a data set, where the look-back period for the calculation can be varied by the user (i.e. the number of points used in the calculations is flexible).

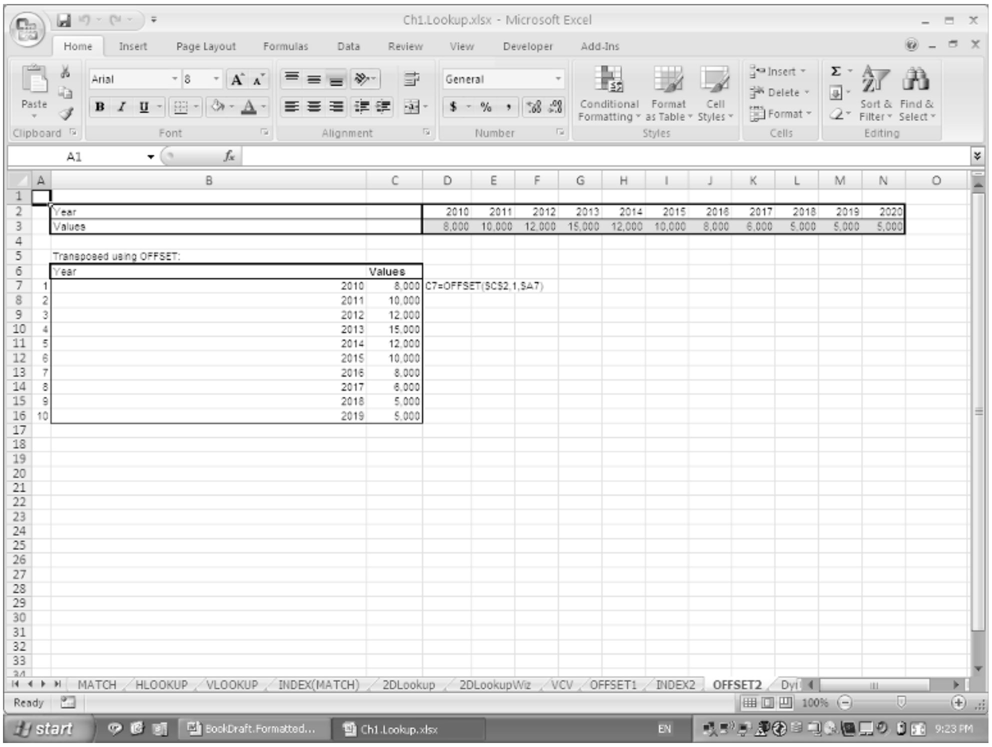
Example: Flexible Correlation Matrix using OFFSET

Another example of the use of **OFFSET** in a form where it returns an array (directly as an argument of a formula) is for the calculation of correlation matrices. For anything but a small number of assets, this would be a much quicker way of calculating the correlation matrix than using individual entries of the **CORREL** function, as the range arguments would need to be changed from cell to cell within the matrix.

The file Ch1.Lookup.xlsx (OFFSET3Correl worksheet) (Figure 1.31) shows an example.

TEXT FUNCTIONS

Excel's **TEXT** functions can be useful not only in their own right but also to allow manipulation of numerical fields by first turning them into text, operating on the text fields and



Year	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Values	8,000	10,000	12,000	15,000	12,000	10,000	8,000	6,000	5,000	5,000	5,000

Transposed using OFFSET:

	Year	Values
1	2010	8,000
2	2011	10,000
3	2012	12,000
4	2013	15,000
5	2014	12,000
6	2015	10,000
7	2016	8,000
8	2017	6,000
9	2018	5,000
10	2019	5,000

C7=OFFSET($C$2,1,$A7)

Figure 1.29

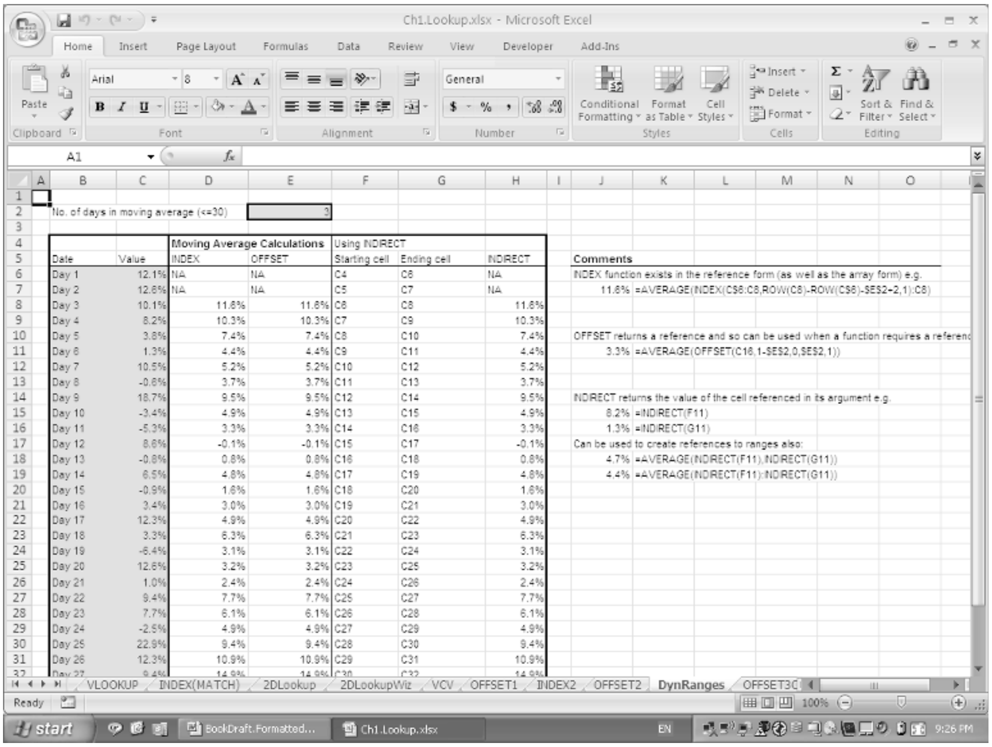
turning the text back into a numerical field. A selected list of some of these functions includes:

- **LEN** counts the length of a string.
- **MID** refers to text in the middle of a text string.
- **REPLACE** replaces a portion of text, starting at a specified place.
- **SEARCH** determines the starting position of a text fragment within a string; similarly, **FIND** finds the position of one text string within another.
- **TEXT** formats a numerical value and converts it to text.
- **VALUE** converts a text string that “looks like” a value to an actual value.

Example: Manipulating Data I using TEXT, MID and VALUE

Occasionally one may need to take a set of multi-digit numbers and split each one into individual digits. For example, a sequence of digits such as 100100001 may represent whether an individual borrower was up-to-date or not with repayments of a loan or mortgage, where every digit represents the position at the end of a given month. It may be required to analyse the aggregate monthly position of a portfolio of borrowers, requiring each individual digit to be present in a separate cell.

The file Ch1.Text.xlsx (SplitNo worksheet) (Figure 1.32) shows the use of the **TEXT** function to convert each number into text, from which each relevant digit can be extracted


No. of days in moving average (<=30) 3

Date	Value	Moving Average Calculations INDEX	OFFSET	Using INDIRECT Starting cell	Ending cell	INDIRECT
Day 1	12.1%	NA	NA	C4	C6	NA
Day 2	12.6%	NA	NA	C5	C7	NA
Day 3	10.1%	11.6%	11.6%	C6	C8	11.6%
Day 4	8.2%	10.3%	10.3%	C7	C9	10.3%
Day 5	3.8%	7.4%	7.4%	C8	C10	7.4%
Day 6	1.3%	4.4%	4.4%	C9	C11	4.4%
Day 7	10.5%	5.2%	5.2%	C10	C12	5.2%
Day 8	-0.6%	3.7%	3.7%	C11	C13	3.7%
Day 9	18.7%	9.5%	9.5%	C12	C14	9.5%
Day 10	-3.4%	4.9%	4.9%	C13	C15	4.9%
Day 11	-5.3%	3.3%	3.3%	C14	C16	3.3%
Day 12	8.6%	-0.1%	-0.1%	C15	C17	-0.1%
Day 13	-0.8%	0.8%	0.8%	C16	C18	0.8%
Day 14	6.5%	4.8%	4.8%	C17	C19	4.8%
Day 15	-0.9%	1.6%	1.6%	C18	C20	1.6%
Day 16	3.4%	3.0%	3.0%	C19	C21	3.0%
Day 17	12.3%	4.9%	4.9%	C20	C22	4.9%
Day 18	3.3%	6.3%	6.3%	C21	C23	6.3%
Day 19	-6.4%	3.1%	3.1%	C22	C24	3.1%
Day 20	12.6%	3.2%	3.2%	C23	C25	3.2%
Day 21	1.0%	2.4%	2.4%	C24	C26	2.4%
Day 22	9.4%	7.7%	7.7%	C25	C27	7.7%
Day 23	7.7%	6.1%	6.1%	C26	C28	6.1%
Day 24	-2.5%	4.9%	4.9%	C27	C29	4.9%
Day 25	22.9%	9.4%	9.4%	C28	C30	9.4%
Day 26	12.3%	10.9%	10.9%	C29	C31	10.9%
Day 27	9.4%	14.9%	14.9%	C30	C32	14.9%

Comments

INDEX function exists in the reference form (as well as the array form) e.g.

11.6% =AVERAGE(INDEX(C$6:C8,ROW(C8)-ROW(C$6)-$E$2+2,1):C8)

OFFSET returns a reference and so can be used when a function requires a referenc

3.3% =AVERAGE(OFFSET(C16,1-$E$2,0,$E$2,1))

INDIRECT returns the value of the cell referenced in its argument e.g.

8.2% =INDIRECT(F11)

1.3% =INDIRECT(G11)

Can be used to create references to ranges also:

4.7% =AVERAGE(INDIRECT(F11),INDIRECT(G11))

4.4% =AVERAGE(INDIRECT(F11):INDIRECT(G11))

Figure 1.30

as a text field (using the **MID** function), before converting this text field into a value (using the **VALUE** function).

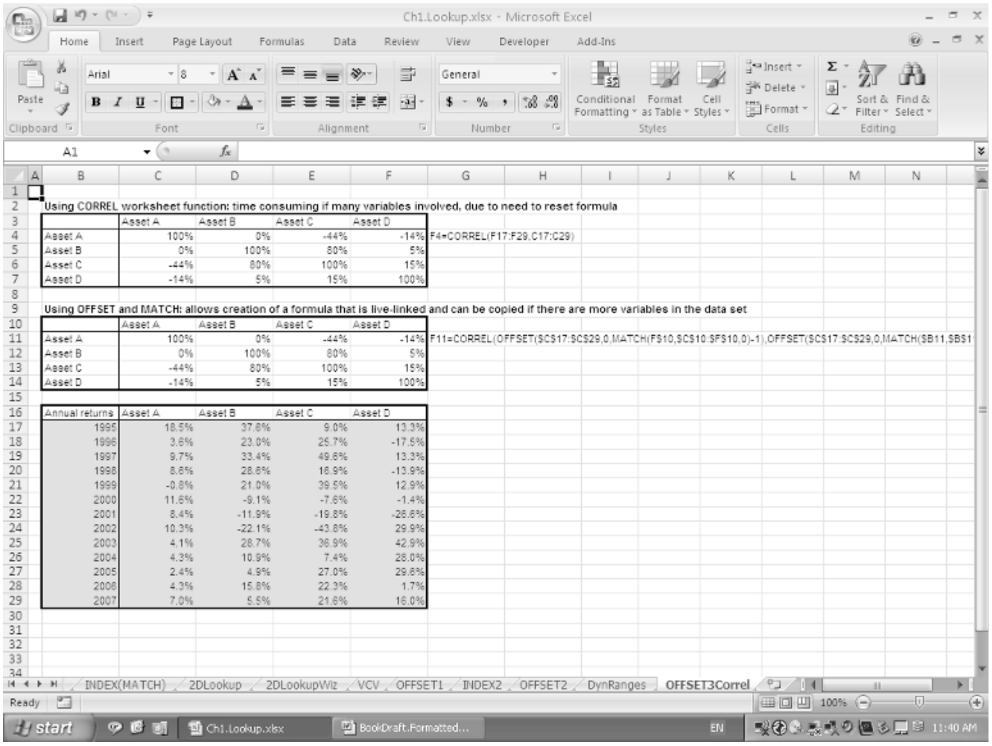
Example: Manipulating Data II using SEARCH, MID and VALUE

Sometimes one may be required to extract a value that is embedded within a text field.

The file Ch1.Text.xlsx (DataSet1 worksheet) (Figure 1.33) shows an example where it is desired to know the total time spent on a set of customer projects for various customers, where this data is contained within the corresponding text field. The key part of solving such a problem is to identify a commonality that defines the position of the relevant field; in this case it is that the field is directly after the first bracket. The **SEARCH** function is used to find the position of the bracket (**FIND** could also be used), following which the **MID** function extracts the text field that immediately follows the bracket. The **VALUE** function is used to convert this text field to data, allowing the total number of days to be calculated.

Example: Updating Text Labels and Graph Titles using TEXT

When producing charts whose titles or legends contain numbers (perhaps themselves calculated from a data set in the model), it can be useful if these fields update automatically if the numbers or data set change. This is especially true if several graphs of different cases



Using CORREL worksheet function: time consuming if many variables involved, due to need to reset formula

	Asset A	Asset B	Asset C	Asset D
Asset A	100%	0%	-44%	-14%
Asset B	0%	100%	80%	5%
Asset C	-44%	80%	100%	15%
Asset D	-14%	5%	15%	100%

F4=CORREL(F17:F29,C17:C29)

Using OFFSET and MATCH: allows creation of a formula that is live-linked and can be copied if there are more variables in the data set

	Asset A	Asset B	Asset C	Asset D
Asset A	100%	0%	-44%	-14%
Asset B	0%	100%	80%	5%
Asset C	-44%	80%	100%	15%
Asset D	-14%	5%	15%	100%

F11=CORREL(OFFSET($C$17:$C$29,0,MATCH(F$10,$C$10:$F$10,0)-1),OFFSET($C$17:$C$29,0,MATCH($B11,$B$1

Annual returns	Asset A	Asset B	Asset C	Asset D
1995	18.5%	37.6%	9.0%	13.3%
1996	3.6%	23.0%	25.7%	-17.5%
1997	9.7%	33.4%	49.6%	13.3%
1998	8.6%	28.6%	16.9%	-13.9%
1999	-0.8%	21.0%	39.5%	12.9%
2000	11.6%	-9.1%	-7.6%	-1.4%
2001	8.4%	-11.9%	-19.8%	-26.6%
2002	10.3%	-22.1%	-43.8%	29.9%
2003	4.1%	28.7%	36.9%	42.9%
2004	4.3%	10.9%	7.4%	28.0%
2005	2.4%	4.9%	27.0%	29.6%
2006	4.3%	15.8%	22.3%	1.7%
2007	7.0%	5.5%	21.6%	16.0%

Figure 1.31

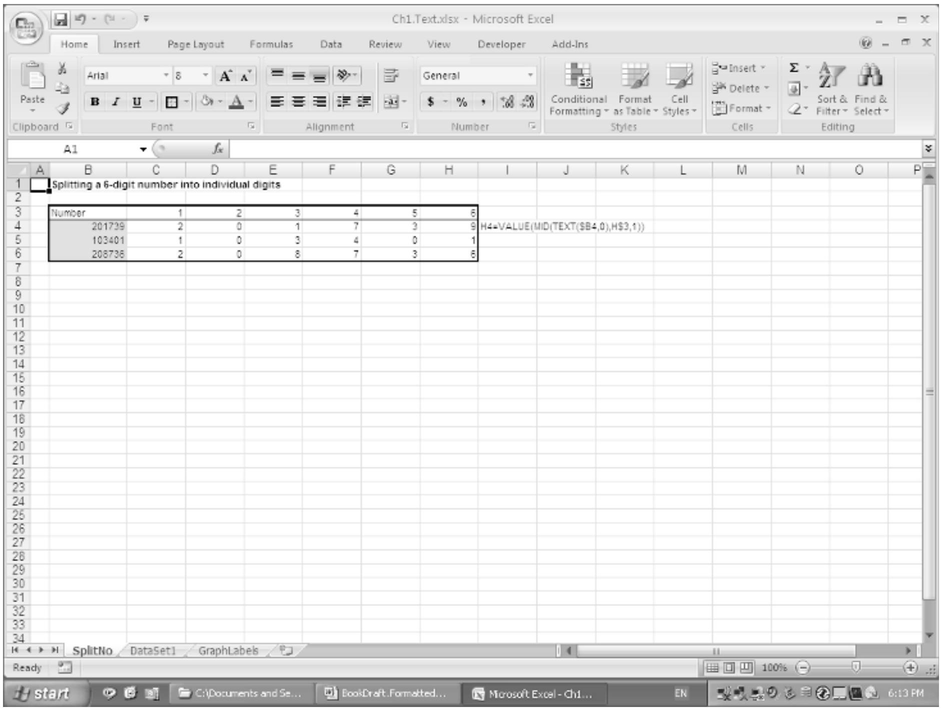
or sensitivities are required, as it will help to avoid confusion about which graph shows which scenario. This is a two-step process. The first step involves creating the desired title and labels as Excel formulae which update as the data changes (using the **TEXT** function). The second step involves linking these formulae to the chart labels and title.

The file Ch1.Text.xlsx (GraphLabels worksheet) (Figure 1.34) shows an example. The creation of the updating labels is achieved with the use of the **TEXT** function, such as:

="USA ("&TEXT(C5,"0.0%")&" p.a. growth)"

As a formula, the expression must of course begin with an equals (=) sign, with the first part of the expression being made into a text label by enclosing it in inverted commas, which is then joined to the rest of the formula using &. The **TEXT** function has a first argument which refers to the cell containing the number to be updated, and a second argument which defines the format to be used (another frequent format is to use **"0.0"** to show a numerical field with one decimal point, or **"0"** to show no decimal places, as used for the chart title). Some spaces may need to be added at appropriate places within the text fields, e.g. immediately after an opening inverted comma or before a closing one.

The linking of the series names from a chart to the updating cells can be done when editing the data source of the chart. For chart titles, one must edit the title field typing = within


Splitting a 6-digit number into individual digits

Number	1	2	3	4	5	6
201739	2	0	1	7	3	9
103401	1	0	3	4	0	1
208736	2	0	8	7	3	6

H4=VALUE(MID(TEXT($B4,0),H$3,1))

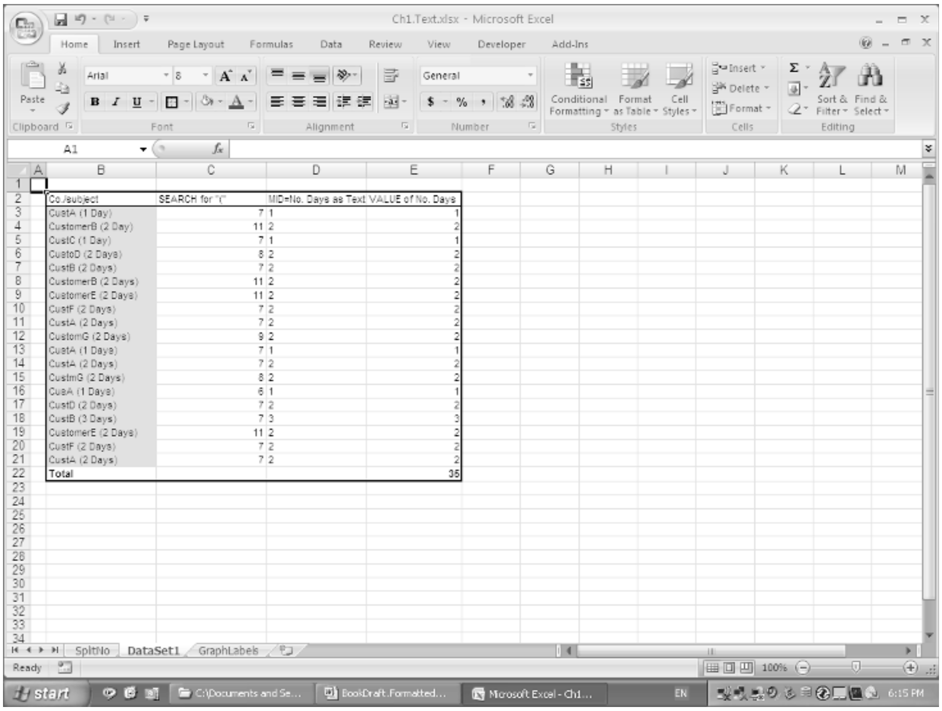
Figure 1.32

the Excel **Formula Bar**, point to the cell containing the required title, and press **RETURN**.

INFORMATION FUNCTIONS

Excel has a number of **Information** functions, many of which return **TRUE** (1 if used within a formula) or **FALSE** (0 if used within a formula) according to whether some condition holds. These can be used in many ways, including:

- Writing comments within numerical formulae. A formula can be multiplied by **ISTEXT** with a comment as its argument (e.g. =**105*ISTEXT("data from 2007")** will evaluate to 105). **ISNUMBER** could also be used in this context, where the corresponding formula is added rather than multiplied, as the presence of the text comment would result in this formula evaluating to zero.
- General error checking (including in VBA code). **ISERROR** can be used to check for any error value (#**N/A**, #**VALUE**!, #**REF**!, #**DIV**/0!, #**NUM**!, #**NAME**?, or #**NULL**!). The function **IFERROR** (in Excel 2007) returns the value of an expression or an alternative value when the base value returns an error. **ISERR** checks for any error value except **#N/A** (value not available), and **ISNA** checks specifically for the **#N/A** error.
- Providing other information about the content, position or format of a cell, using functions such as **CELL** and **TYPE**.



Co./subject	SEARCH for "("	MID=No. Days as Text	VALUE of No. Days
CustA (1 Day)	7	1	1
CustomerB (2 Day)	11	2	2
CustC (1 Day)	7	1	1
CustoD (2 Days)	8	2	2
CustB (2 Days)	7	2	2
CustomerB (2 Days)	11	2	2
CustomerE (2 Days)	11	2	2
CustF (2 Days)	7	2	2
CustA (2 Days)	7	2	2
CustomG (2 Days)	9	2	2
CustA (1 Days)	7	1	1
CustA (2 Days)	7	2	2
CustmG (2 Days)	8	2	2
CusA (1 Days)	6	1	1
CustD (2 Days)	7	2	2
CustB (3 Days)	7	3	3
CustomerE (2 Days)	11	2	2
CustF (2 Days)	7	2	2
CustA (2 Days)	7	2	2
Total			35

Figure 1.33

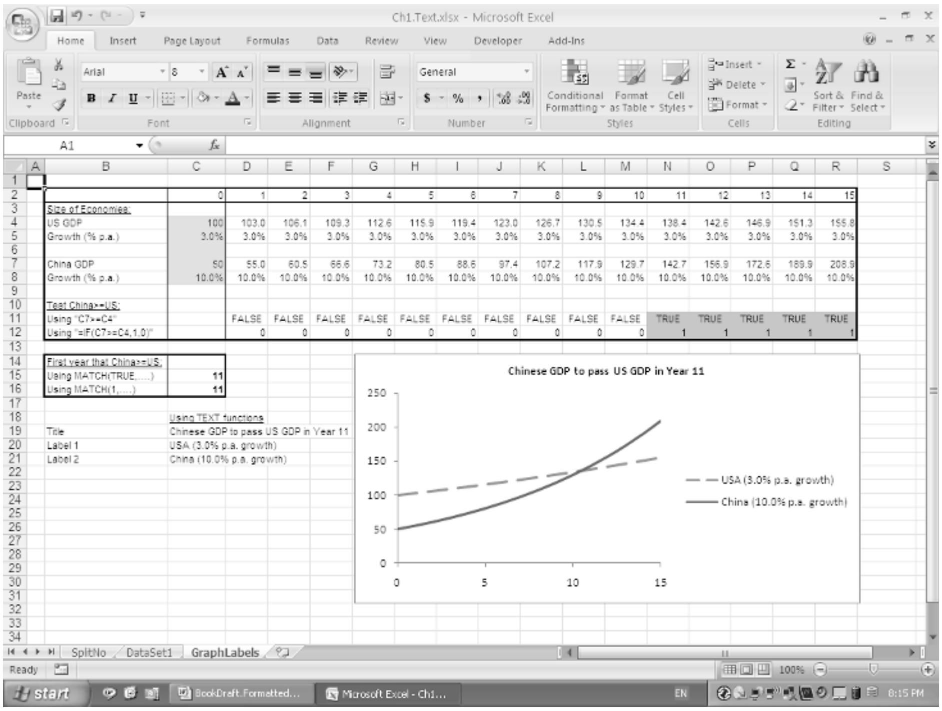
Example: Summing a Range using IFERROR

IFERROR is a **Logical** function in Excel 2007. It can be used to check for errors, returning a base value when the value of an expression is not an error, and an alternative value in the case of an error. An example is where one wishes to sum a range, but where some elements in the range result from a **Lookup** function, which in some cases may return an error, and where in such cases a value of zero is desired to be used for that value in the range. (In Excel 2003, similar functionality can be achieved by using **ISERROR** embedded within an **IF** statement, but this required a more complex and cumbersome format.)

The file Ch1.Info.xlxs (IFERROR worksheet) (Figure 1.35) shows an example where it is desired to form the sum of a subset of the values in the data set, where the subset is defined according to the labels (country names) that the user will define. The **INDEX(MATCH)** combination (see earlier) is used to search for the relevant values. In the first part of the example no error-check is made, whereas in the second part, **IFERROR** is used to perform an error-check (so that if no country label is defined in the cells of the range, a zero value is returned and the sum can still be formed).

Example: Updating Labels using CELL

The **CELL** function provides information about the position, content or format of a cell. Its arguments include the information type that is sought (such as the address, column or row



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Size of Economies:																
US GDP	100	103.0	106.1	109.3	112.6	115.9	119.4	123.0	126.7	130.5	134.4	138.4	142.6	146.9	151.3	155.8
Growth (% p.a.)	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%
China GDP	50	55.0	60.5	66.6	73.2	80.5	88.6	97.4	107.2	117.9	129.7	142.7	156.9	172.6	189.9	208.9
Growth (% p.a.)	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%
Test China>=US:																
Using "C7>=C4"		FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
Using "=IF(C7>=C4,1,0)"		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

First year that China>=US:	
Using MATCH(TRUE,....)	11
Using MATCH(1,....)	11

	Using TEXT functions
Title	Chinese GDP to pass US GDP in Year 11
Label 1	USA (3.0% p.a. growth)
Label 2	China (10.0% p.a. growth)

Figure 1.34

number of a cell or its values) and a reference that specifies the cell for which information is desired. It can be used in a variety of ways, such as:

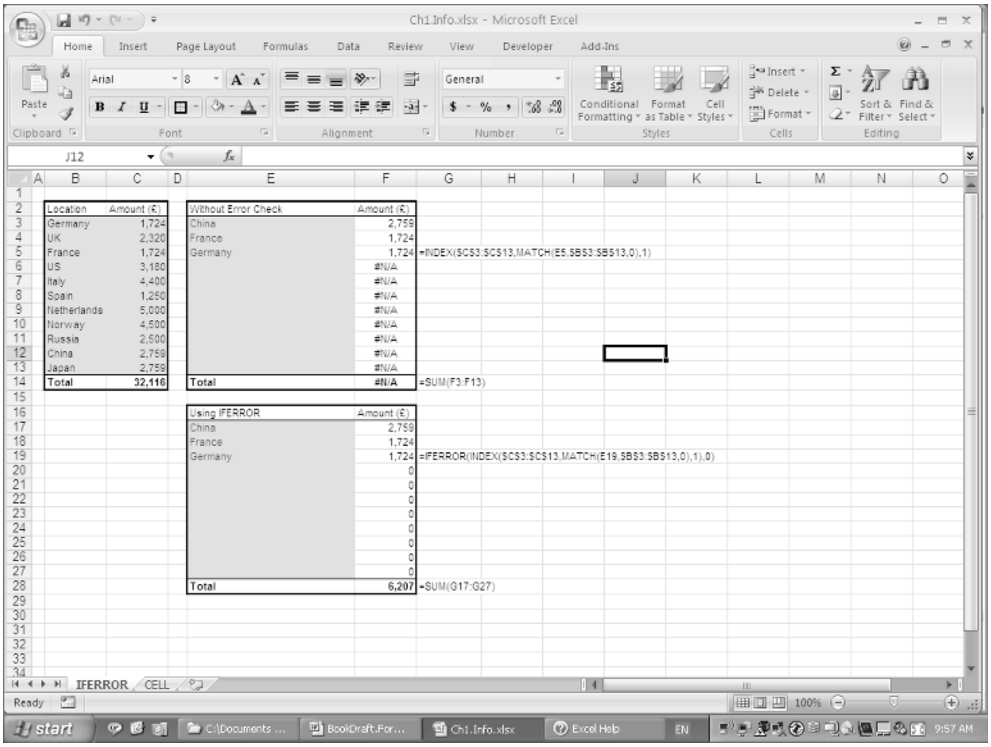
- To provide an updating label for a variable (so that, for example, its description refers to the data range that is being used in the formula for that variable).
- To show the cell in a workbook that was most recently changed, by using it in the form **CELL("address")**, i.e. without a reference argument.

The file Ch1.Info.xlxs (CELL worksheet) (Figure 1.36) shows an example.

ARRAY FUNCTIONS, FORMULAE AND MATRIX CALCULATIONS

The use of array formulae is in some circumstances both powerful and unavoidable. Examples include matrix multiplication and transposition, and formulae to count the frequency of the occurrence of a set of points within a set of values.

Array formulae return arrays as their output, and must be entered in Excel using **CTRL+SHIFT+ENTER** (rather than the usual **ENTER**). Some array functions extend over multiple cells, in which case the entire range must first be highlighted and the formula entered in the **Formula Bar** or by selecting the function using the **Formulas** menu before entering with **CTRL+SHIFT+ENTER**.



Location	Amount (€)
Germany	1,724
UK	2,320
France	1,724
US	3,180
Italy	4,400
Spain	1,250
Netherlands	5,000
Norway	4,500
Russia	2,500
China	2,759
Japan	2,759
Total	32,116

Without Error Check	Amount (€)	
China	2,759	
France	1,724	
Germany	1,724	=INDEX($C$3:$C$13,MATCH(E5,$B$3:$B$13,0),1)
	#N/A	
	#N/A	
	#N/A	
	#N/A	
	#N/A	
	#N/A	
	#N/A	
	#N/A	
Total	#N/A	=SUM(F3:F13)

Using IFERROR	Amount (£)	
China	2,759	
France	1,724	
Germany	1,724	=IFERROR(INDEX($C$3:$C$13,MATCH(E19,$B$3:$B$13,0),1),0)
	0	
	0	
	0	
	0	
	0	
	0	
	0	
	0	
Total	6,207	=SUM(G17:G27)

Figure 1.35

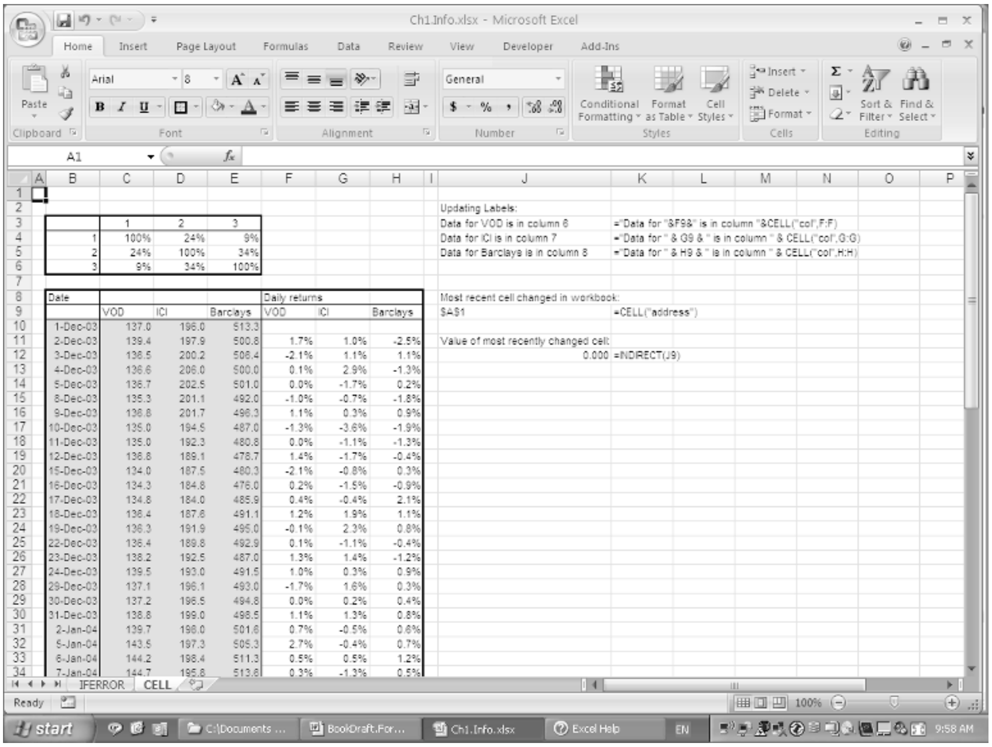
Despite their potential power, there are several disadvantages with using array functions and formulae. First, their use can slow down the calculation of a workbook. Second, a failure to enter the formula correctly (e.g. by using just **ENTER**) could result in a value appearing in the cell, but one that is incorrect, so that inadvertent errors may arise (in contrast to when a **#VALUE** error message appears). Third, many users are not familiar with them, which may result in the model being harder for others to understand or interpret, or where the user may accidentally edit a formula and return **ENTER**, leading to an error message or an incorrect value (if the ranges with array formula are not protected).

Example: SUMPRODUCT using SUM as an Array Formula

The file Ch1.Array.xlsx (SUMPasArray worksheet) (Figure 1.37) shows a basic example for illustrative purposes in which the **SUM** function is used to create an array function which produces a result that is the same as the **SUMPRODUCT** function. In this case the function is created by selecting the first range, using the times symbol, selecting the second range and entering using **CTRL+SHIFT+ENTER.**

Example: Histogram of Returns using FREQUENCY

FREQUENCY is a **Statistical** function that counts the number of times that points within a data set lie within each of a set of predefined ranges (or bins). It can be used to create



	1	2	3
1	100%	24%	9%
2	24%	100%	34%
3	9%	34%	100%

Date				Daily returns		
	VOD	ICI	Barclays	VOD	ICI	Barclays
1-Dec-03	137.0	196.0	513.3			
2-Dec-03	139.4	197.9	500.8	1.7%	1.0%	-2.5%
3-Dec-03	136.5	200.2	506.4	-2.1%	1.1%	1.1%
4-Dec-03	136.6	206.0	500.0	0.1%	2.9%	-1.3%
5-Dec-03	136.7	202.5	501.0	0.0%	-1.7%	0.2%
8-Dec-03	135.3	201.1	492.0	-1.0%	-0.7%	-1.8%
9-Dec-03	136.8	201.7	496.3	1.1%	0.3%	0.9%
10-Dec-03	135.0	194.5	487.0	-1.3%	-3.6%	-1.9%
11-Dec-03	135.0	192.3	480.8	0.0%	-1.1%	-1.3%
12-Dec-03	136.8	189.1	478.7	1.4%	-1.7%	-0.4%
15-Dec-03	134.0	187.5	480.3	-2.1%	-0.8%	0.3%
16-Dec-03	134.3	184.8	476.0	0.2%	-1.5%	-0.9%
17-Dec-03	134.8	184.0	485.9	0.4%	-0.4%	2.1%
18-Dec-03	136.4	187.6	491.1	1.2%	1.9%	1.1%
19-Dec-03	136.3	191.9	495.0	-0.1%	2.3%	0.8%
22-Dec-03	136.4	189.8	492.9	0.1%	-1.1%	-0.4%
23-Dec-03	138.2	192.5	487.0	1.3%	1.4%	-1.2%
24-Dec-03	139.5	193.0	491.5	1.0%	0.3%	0.9%
29-Dec-03	137.1	196.1	493.0	-1.7%	1.6%	0.3%
30-Dec-03	137.2	196.5	494.8	0.0%	0.2%	0.4%
31-Dec-03	138.8	199.0	498.5	1.1%	1.3%	0.8%
2-Jan-04	139.7	198.0	501.6	0.7%	-0.5%	0.6%
5-Jan-04	143.5	197.3	505.3	2.7%	-0.4%	0.7%
6-Jan-04	144.2	198.4	511.3	0.5%	0.5%	1.2%
7-Jan-04	144.7	195.8	513.6	0.3%	-1.3%	0.5%

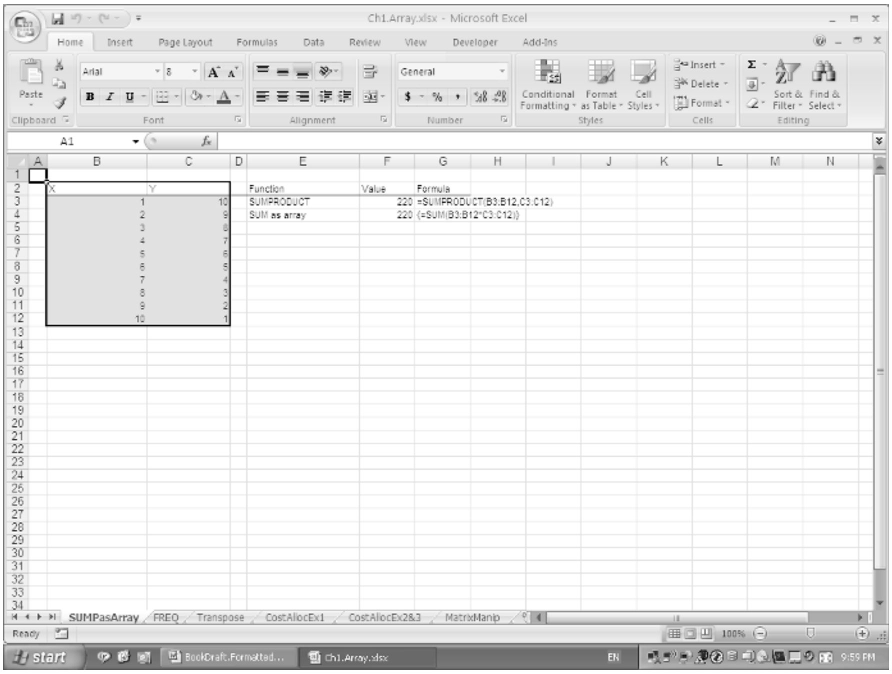
Updating Labels:	
Data for VOD is in column 6	="Data for "&F9&" is in column "&CELL("col",F:F)
Data for ICI is in column 7	="Data for " & G9 & " is in column " & CELL("col",G:G)
Data for Barclays is in column 8	="Data for " & H9 & " is in column " & CELL("col",H:H)
Most recent cell changed in workbook:	
$A$1	=CELL("address")
Value of most recently changed cell:	
0.000	=INDIRECT(J9)

Figure 1.36

the data to produce a histogram of the data set. In practice, the choice of the definition of the width of the bins will be important; if the bin size is too wide the histogram will not be sufficiently detailed, whereas for narrow bin widths the histogram will look fragmented and multi-moded.

When using the **FREQUENCY** function, the entire range that is to contain the function must be selected (this range must be one cell larger than the bin range because there could be data points larger than the upper value of the largest bin), and the function must be typed within the **Formula Bar** or by selecting the function using the **Formulas** menu (rather than by direct typing in the first cell of the range, for example). On pressing **CTRL+SHIFT+ ENTER** the function will show the number of data points in the set that lie between the lower bin value and the upper bin value. Once this function is entered the relative frequency of each bin can be calculated.

The file Ch1.Array.xlsx (FREQ worksheet) (Figure 1.38) shows an example where the average daily movement of the Dow Jones index for the period under consideration (2007) is 0.02% with a standard deviation of 0.92%. The bin size is set to 0.25%, starting at −3.00%. As mentioned above, the range in which the function is entered is one cell larger than the range containing the bins, to account for the data points that are larger than the maximum bin value.



X	Y		Function	Value	Formula
1	10		SUMPRODUCT	220	=SUMPRODUCT(B3:B12,C3:C12)
2	9		SUM as array	220	{=SUM(B3:B12*C3:C12)}
3	8				
4	7				
5	6				
6	5				
7	4				
8	3				
9	2				
10	1				

Figure 1.37

Example: Capex and Depreciation Schedules using TRANSPOSE

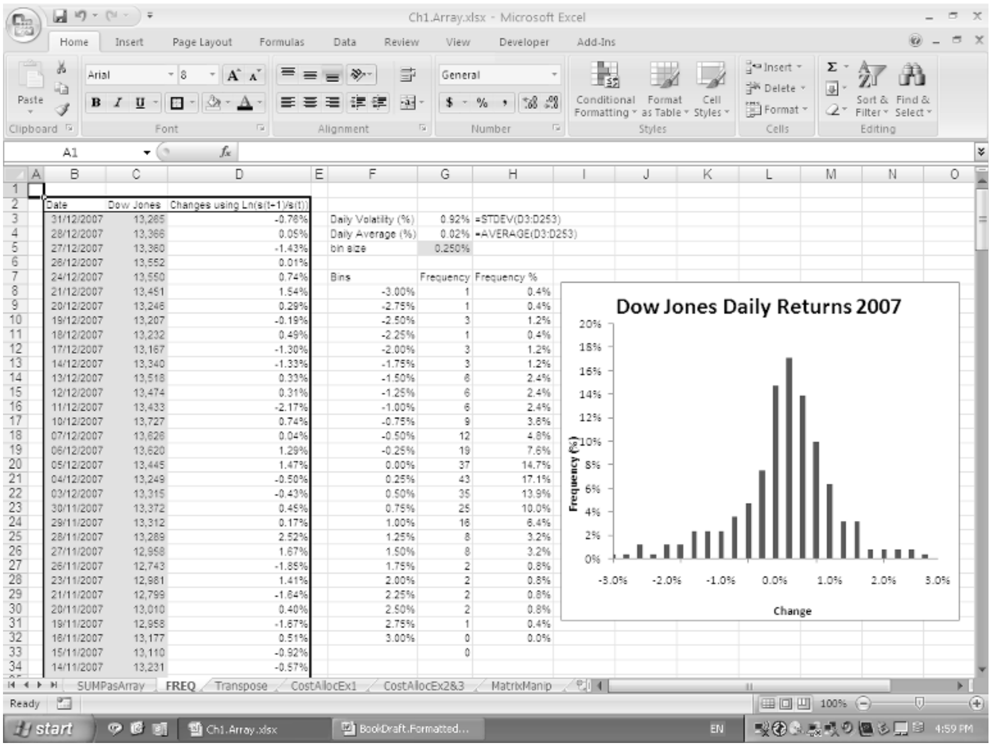
The **TRANSPOSE** function can be used to turn a row of entries into column format. For example, if a capital expenditure (capex) profile has been defined or determined over time, and a depreciation policy has been set, then one may wish to calculate the depreciation charge by year.

The file Ch1.Array.xlsx (Transpose worksheet) (Figure 1.39) shows an example. The year numbers and the capex amounts in the rows are transposed to columns (cells B12 through B21 and C12 through C21). A depreciation policy has then been applied, in this case using the **VDB** (**Financial**) function.

Example: Cost Allocation using TRANSPOSE

The file Ch1.Array.xlsx (CostAllocEx1 worksheet) (Figure 1.40) shows the use of array formulae to allocate the projected costs of a set of central overhead departments to business units, based on an allocation matrix and the **TRANSPOSE** formula. That is, each year the costs allocated to each business unit is the sum of the product of the range containing the indirect costs with the percentages from the allocation matrix. The layout of the allocation matrix requires that the percentages are transposed using **TRANSPOSE** in order for the **SUMPRODUCT** function to be used.

The file Ch1.Array.xlsx (CostAllocEx2&3 worksheet) (Figure 1.41) shows that if the allocation matrix had been initially set up in a way that was structurally transposed, then the use of **TRANSPOSE** and the array formula would not be necessary. The section of


Ch1.Array.xlsx - Microsoft Excel
Home Insert Page Layout Formulas Data Review View Developer Add-Ins
A1
SUMPasArray FREQ Transpose CostAllocEx1 CostAllocEx2&3 MatrixManip
Ready
Dow Jones Daily Returns 2007
Frequency (%)
Change
20% 18% 16% 14% 12% 10% 8% 6% 4% 2% 0%
-3.0% -2.0% -1.0% 0.0% 1.0% 2.0% 3.0%

	B	C	D	F	G	H
1						
2	Date	Dow Jones	Changes using Ln(s(t+1)/s(t))			
3	31/12/2007	13,265	-0.76%	Daily Volatility (%)	0.92%	=STDEV(D3:D253)
4	28/12/2007	13,366	0.05%	Daily Average (%)	0.02%	=AVERAGE(D3:D253)
5	27/12/2007	13,360	-1.43%	bin size	0.250%	
6	26/12/2007	13,552	0.01%			
7	24/12/2007	13,550	0.74%	Bins	Frequency	Frequency %
8	21/12/2007	13,451	1.54%	-3.00%	1	0.4%
9	20/12/2007	13,246	0.29%	-2.75%	1	0.4%
10	19/12/2007	13,207	-0.19%	-2.50%	3	1.2%
11	18/12/2007	13,232	0.49%	-2.25%	1	0.4%
12	17/12/2007	13,167	-1.30%	-2.00%	3	1.2%
13	14/12/2007	13,340	-1.33%	-1.75%	3	1.2%
14	13/12/2007	13,518	0.33%	-1.50%	6	2.4%
15	12/12/2007	13,474	0.31%	-1.25%	6	2.4%
16	11/12/2007	13,433	-2.17%	-1.00%	6	2.4%
17	10/12/2007	13,727	0.74%	-0.75%	9	3.6%
18	07/12/2007	13,626	0.04%	-0.50%	12	4.8%
19	06/12/2007	13,620	1.29%	-0.25%	19	7.6%
20	05/12/2007	13,445	1.47%	0.00%	37	14.7%
21	04/12/2007	13,249	-0.50%	0.25%	43	17.1%
22	03/12/2007	13,315	-0.43%	0.50%	35	13.9%
23	30/11/2007	13,372	0.45%	0.75%	25	10.0%
24	29/11/2007	13,312	0.17%	1.00%	16	6.4%
25	28/11/2007	13,289	2.52%	1.25%	8	3.2%
26	27/11/2007	12,958	1.67%	1.50%	8	3.2%
27	26/11/2007	12,743	-1.85%	1.75%	2	0.8%
28	23/11/2007	12,981	1.41%	2.00%	2	0.8%
29	21/11/2007	12,799	-1.64%	2.25%	2	0.8%
30	20/11/2007	13,010	0.40%	2.50%	2	0.8%
31	19/11/2007	12,958	-1.67%	2.75%	1	0.4%
32	16/11/2007	13,177	0.51%	3.00%	0	0.0%
33	15/11/2007	13,110	-0.92%		0	
34	14/11/2007	13,231	-0.57%			

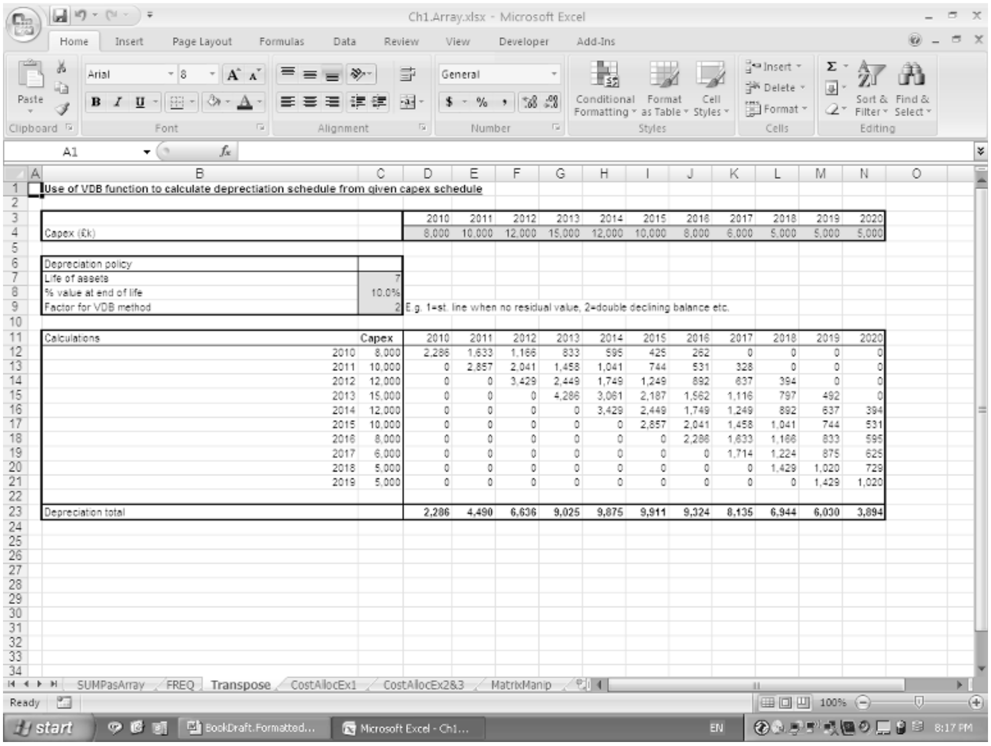
Figure 1.38

the worksheet titled Example 2 uses a formula that can be copied to all relevant cells within a row but needs to be edited separately when copying downwards. The section titled Example 3 uses the **INDEX/MATCH** combination (using the reference form of the **INDEX** function) to create a single formula that can be copied from one cell to the entire range.

Each of the three approaches has its advantages and disadvantages. The third avoids the use of array functions and presents the general formula that can be copied. On the other hand, the second example is generally the easiest to understand and may be preferable in situations where there are only a few business units or allocation objects (as in this example), so that the number of individual edits of the formulae is manageable. (Of course, in situations where the allocation matrix is predefined in the format of Example 1, the **TRANSPOSE** function could be applied as an array function to the whole matrix in order to turn it into the format of Examples 2 and 3.)

Example: Matrix Multiplication using MMULT and TRANSPOSE

Matrix multiplication can be achieved with the array function **MMULT**. Other matrix functions include **MDETERM** and **MINVERSE**, which calculate the determinant and inverse of a matrix respectively. As mentioned earlier, the calculation of the portfolio's variance



Use of VDB function to calculate deprectiation schedule from given capex schedule

		2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Capex (£k)		8,000	10,000	12,000	15,000	12,000	10,000	8,000	6,000	5,000	5,000	5,000

Depreciation policy	
Life of assets	7
% value at end of life	10.0%
Factor for VDB method	2

E.g. 1=st. line when no residual value, 2=double declining balance etc.

Calculations	Capex	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
2010	8,000	2,286	1,633	1,166	833	595	425	262	0	0	0	0
2011	10,000	0	2,857	2,041	1,458	1,041	744	531	328	0	0	0
2012	12,000	0	0	3,429	2,449	1,749	1,249	892	637	394	0	0
2013	15,000	0	0	0	4,286	3,061	2,187	1,562	1,116	797	492	0
2014	12,000	0	0	0	0	3,429	2,449	1,749	1,249	892	637	394
2015	10,000	0	0	0	0	0	2,857	2,041	1,458	1,041	744	531
2016	8,000	0	0	0	0	0	0	2,286	1,633	1,166	833	595
2017	6,000	0	0	0	0	0	0	0	1,714	1,224	875	625
2018	5,000	0	0	0	0	0	0	0	0	1,429	1,020	729
2019	5,000	0	0	0	0	0	0	0	0	0	1,429	1,020
Depreciation total		**2,286**	**4,490**	**6,636**	**9,025**	**9,875**	**9,911**	**9,324**	**8,135**	**6,944**	**6,030**	**3,894**

Figure 1.39

requires matrix multiplication involving the vector of portfolio weights (w) and the variance-covariance matrix (VCV):

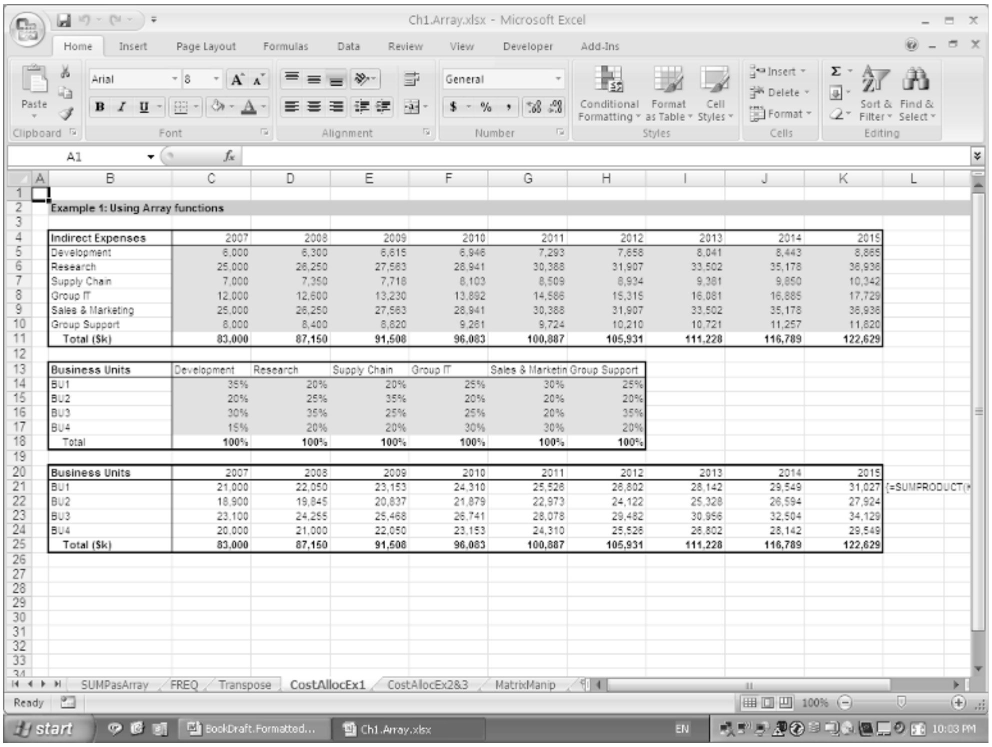
$$\text{Variance of portfolio} = w\text{VCV}w^t$$

The file Ch1.Array.xlsx (MatrixManip worksheet) (Figure 1.42) shows an example. The variance–covariance matrix is calculated using the **HLOOKUP** function from the volatility of each asset and the matrix of correlations (as described earlier in the section on **Lookup** functions). The variance of the portfolio is calculated using the **TRANSPOSE** function and the **MMULT** function, with the standard deviation being the square root of the variance.

GOALSEEK AND SOLVER

Example: Required Growth Rate using GoalSeek

GoalSeek (under **Data/What If Analysis** or under **Tools** in Excel 2003) can be used to find the value of a single input cell that will result in a single model output having a desired value. This can be used to calibrate a model or reset the input in some way.



Example 1: Using Array functions

Indirect Expenses	2007	2008	2009	2010	2011	2012	2013	2014	2015
Development	6,000	6,300	6,615	6,946	7,293	7,658	8,041	8,443	8,865
Research	25,000	26,250	27,563	28,941	30,388	31,907	33,502	35,178	36,936
Supply Chain	7,000	7,350	7,718	8,103	8,509	8,934	9,381	9,850	10,342
Group IT	12,000	12,600	13,230	13,892	14,586	15,315	16,081	16,885	17,729
Sales & Marketing	25,000	26,250	27,563	28,941	30,388	31,907	33,502	35,178	36,936
Group Support	8,000	8,400	8,820	9,261	9,724	10,210	10,721	11,257	11,820
Total ($k)	**83,000**	**87,150**	**91,508**	**96,083**	**100,887**	**105,931**	**111,228**	**116,789**	**122,629**

Business Units	Development	Research	Supply Chain	Group IT	Sales & Marketin	Group Support
BU1	35%	20%	20%	25%	30%	25%
BU2	20%	25%	35%	20%	20%	20%
BU3	30%	35%	25%	25%	20%	35%
BU4	15%	20%	20%	30%	30%	20%
Total	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

Business Units	2007	2008	2009	2010	2011	2012	2013	2014	2015	
BU1	21,000	22,050	23,153	24,310	25,526	26,802	28,142	29,549	31,027	{=SUMPRODUCT(
BU2	18,900	19,845	20,837	21,879	22,973	24,122	25,328	26,594	27,924	
BU3	23,100	24,255	25,468	26,741	28,078	29,482	30,956	32,504	34,129	
BU4	20,000	21,000	22,050	23,153	24,310	25,526	26,802	28,142	29,549	
Total ($k)	**83,000**	**87,150**	**91,508**	**96,083**	**100,887**	**105,931**	**111,228**	**116,789**	**122,629**	

Figure 1.40

When using **GoalSeek** (and **Solver**) it is often more transparent and flexible to set up an output calculation with the objective of setting this equal to zero. For example:

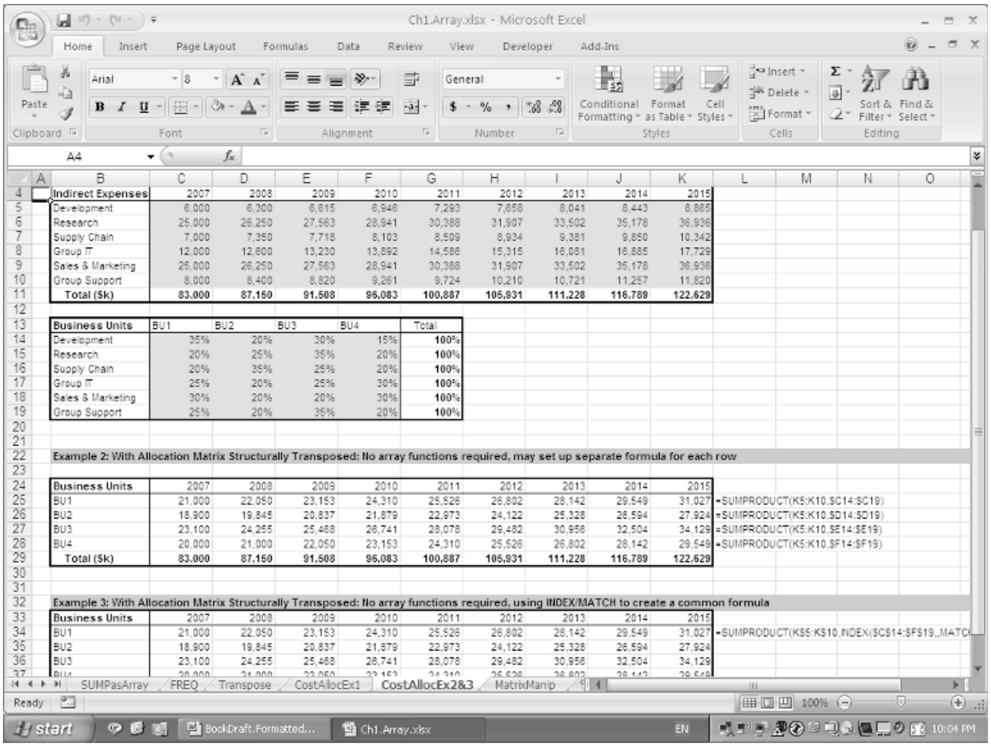
- Where it is desired to find the input value so that a calculated quantity is equal to a fixed number, then the difference between this calculation and the fixed number can be set as an output that should be zero.
- Where it is desired that two calculated quantities be equal, then the difference between them can be calculated as an output, with the aim making this equal to zero.

The file Ch1.GS&Solver.xlsx (GS1 worksheet) (Figure 1.43) of shows an example using the earlier US and Chinese GDP forecast, where it is desired to find the growth rate required so that Chinese GDP would exceed that of the US for the first time in Year 8. The **GoalSeek** menu is essentially self-explanatory, and results in a periodic growth rate of about 13.6%.

Example: Implied Volatility using GoalSeek

GoalSeek can also be used to find the implied volatility of a European option, i.e. the volatility required in the Black–Scholes (BS) formula so that the observed market price is equal to the option value according to the formula.

The file Ch1.GS&Solver.xlsx (GS2 worksheet) (Figure 1.44) calculates the BS value of a European option under assumptions about the key parameters and variables (the formula



Indirect Expenses	2007	2008	2009	2010	2011	2012	2013	2014	2015
Development	6,000	6,300	6,615	6,946	7,293	7,658	8,041	8,443	8,865
Research	25,000	26,250	27,563	28,941	30,388	31,907	33,502	35,178	36,936
Supply Chain	7,000	7,350	7,718	8,103	8,509	8,934	9,381	9,850	10,342
Group IT	12,000	12,600	13,230	13,892	14,586	15,315	16,081	16,885	17,729
Sales & Marketing	25,000	26,250	27,563	28,941	30,388	31,907	33,502	35,178	36,936
Group Support	8,000	8,400	8,820	9,261	9,724	10,210	10,721	11,257	11,820
Total ($k)	**83,000**	**87,150**	**91,508**	**96,083**	**100,887**	**105,931**	**111,228**	**116,789**	**122,629**

Business Units	BU1	BU2	BU3	BU4	Total
Development	35%	20%	30%	15%	**100%**
Research	20%	25%	35%	20%	**100%**
Supply Chain	20%	35%	25%	20%	**100%**
Group IT	25%	20%	25%	30%	**100%**
Sales & Marketing	30%	20%	20%	30%	**100%**
Group Support	25%	20%	35%	20%	**100%**

Example 2: With Allocation Matrix Structurally Transposed: No array functions required, may set up separate formula for each row

Business Units	2007	2008	2009	2010	2011	2012	2013	2014	2015	
BU1	21,000	22,050	23,153	24,310	25,526	26,802	28,142	29,549	31,027	=SUMPRODUCT(K5:K10,$C14:$C19)
BU2	18,900	19,845	20,837	21,879	22,973	24,122	25,328	26,594	27,924	=SUMPRODUCT(K5:K10,$D14:$D19)
BU3	23,100	24,255	25,468	26,741	28,078	29,482	30,956	32,504	34,129	=SUMPRODUCT(K5:K10,$E14:$E19)
BU4	20,000	21,000	22,050	23,153	24,310	25,526	26,802	28,142	29,549	=SUMPRODUCT(K5:K10,$F14:$F19)
Total ($k)	**83,000**	**87,150**	**91,508**	**96,083**	**100,887**	**105,931**	**111,228**	**116,789**	**122,629**	

Example 3: With Allocation Matrix Structurally Transposed: No array functions required, using INDEX/MATCH to create a common formula

Business Units	2007	2008	2009	2010	2011	2012	2013	2014	2015	
BU1	21,000	22,050	23,153	24,310	25,526	26,802	28,142	29,549	31,027	=SUMPRODUCT(K$5:K$10,INDEX($C$14:$F$19,,MATC
BU2	18,900	19,845	20,837	21,879	22,973	24,122	25,328	26,594	27,924	
BU3	23,100	24,255	25,468	26,741	28,078	29,482	30,956	32,504	34,129	

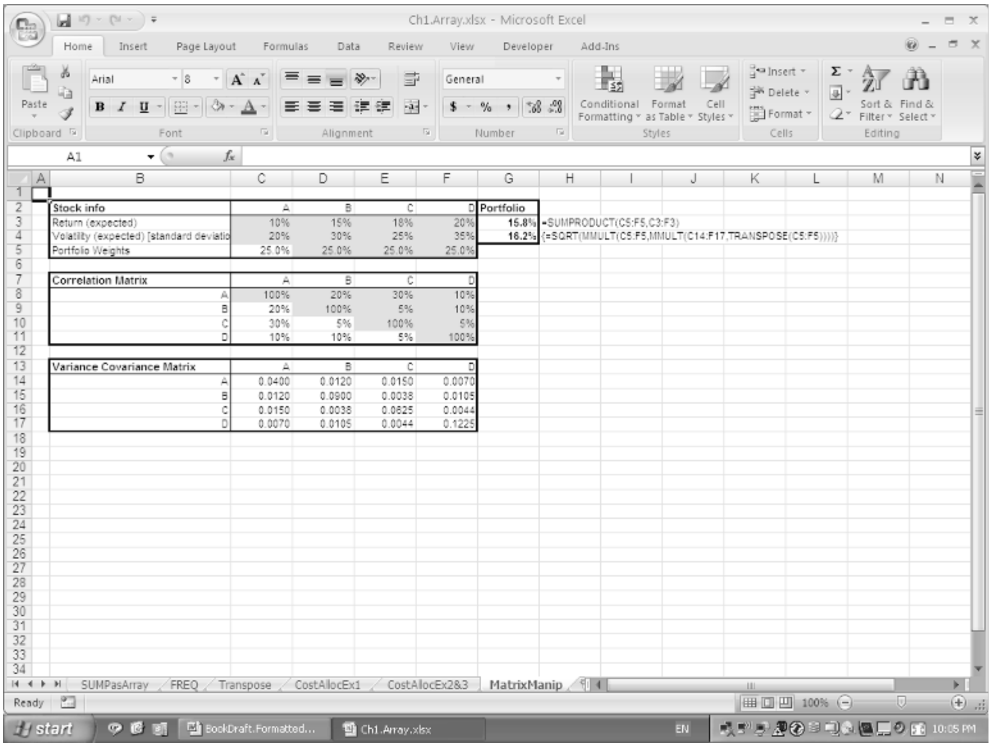
Figure 1.41

is described in more detail in Chapters 2 and 5). **GoalSeek** is used to find the volatility so that the calculated option value is equal to the observed market price of 5; this gives an implied volatility of around 22% p.a.

Example: Portfolio Optimisation using Solver

The **Solver** add-in provides more extensive capability than **GoalSeek,** in that multiple inputs may be varied in order to achieve some result for the output, and constraints may be imposed. **Solver** is found under the **Analysis** group on the **Data** tab (under **Tools** in Excel 2003). If **Solver** (or the **Analysis** group) is not displayed it can be installed (and uninstalled) using the **Manage** tool for add-ins under **Office/Excel Options/Add-ins** (or **Tools**/**Add-ins** in Excel 2003).

The file Ch1.GS&Solver.xlsx (Solver worksheet) (Figure 1.45) shows an example whose objective is to find the mix of assets that maximise the return on the portfolio for a given level of the risk. The Excel array formulae covered earlier are used to implement the required matrix multiplications. **Solver** has been run several times, using a range of target values for the standard deviation (in Chapter 6, a similar example is used where the process of running **Solver** several times is automated using VBA). Two separate runs of **Solver** were used, one where the weights for each asset were constrained to be non-negative and another where the weights were not constrained. The constraints for the weights were set up by creating a cell that calculates the minimum of the weights and constraining that value to



Stock info	A	B	C	D	Portfolio	
Return (expected)	10%	15%	18%	20%	15.8%	=SUMPRODUCT(C5:F5,C3:F3)
Volatility (expected) [standard deviatio	20%	30%	25%	35%	16.2%	{=SQRT(MMULT(C5:F5,MMULT(C14:F17,TRANSPOSE(C5:F5))))}
Portfolio Weights	25.0%	25.0%	25.0%	25.0%		

Correlation Matrix	A	B	C	D
A	100%	20%	30%	10%
B	20%	100%	5%	10%
C	30%	5%	100%	5%
D	10%	10%	5%	100%

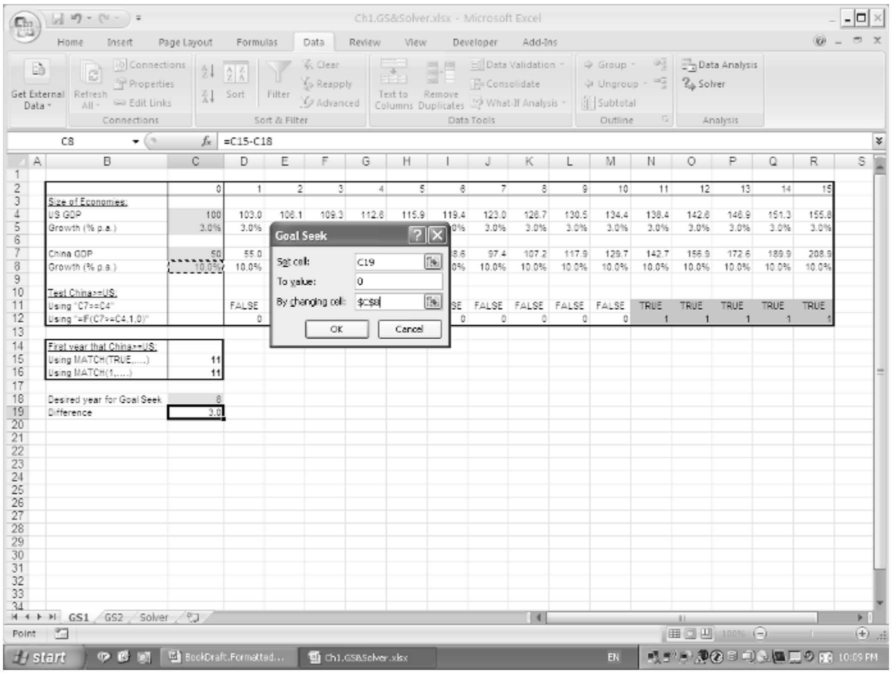
Variance Covariance Matrix	A	B	C	D
A	0.0400	0.0120	0.0150	0.0070
B	0.0120	0.0900	0.0038	0.0105
C	0.0150	0.0038	0.0625	0.0044
D	0.0070	0.0105	0.0044	0.1225

Figure 1.42

be non-zero. (An explicit individual constraint for each weight could also have been set up. The **Solver/Options** menu could have been invoked, as this allows the limiting of the solution search to non-negative values; in this case some further adaptation of the model or the constraint would be necessary to ensure that the weight of asset A was also considered and did not become negative.) The results show that an unconstrained situation allows for larger returns in some cases, corresponding to short-selling of the respective assets.

THE ANALYSIS TOOLPAK AND OTHER ADD-INS

The following gives an introduction to key aspects of some of the add-ins that are included with Excel 2007, including the **Analysis ToolPak** (which provides statistical tools, and which in Excel 2003 also provides some additional worksheet functions that are directly included in Excel 2007), the **Conditional Sum Wizard** (which helps to create formulae that adds values based on a condition or set of conditions) and **Solver** (which finds a combination of inputs that lead to the output having some value, with constraints). Other add-ins included with Excel 2007 that are not covered further here are the **Lookup Wizard** (an example was provided earlier in the section on **Lookup** functions), **Euro Currency Tools** (which allows conversions between legacy currencies in the Euro-area and related topics), and tools for VBA programmers (the **Analysis ToolPak-VBA** and the **Internet Assistant VBA**).



C8 =C15-C18

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
2		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	Size of Economies:																
4	US GDP	100	103.0	106.1	109.3	112.6	115.9	119.4	123.0	126.7	130.5	134.4	138.4	142.6	146.9	151.3	155.8
5	Growth (% p.a.)	3.0%	3.0%					0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%	3.0%
6																	
7	China GDP	50	55.0					8.6	97.4	107.2	117.9	129.7	142.7	156.9	172.6	189.9	208.9
8	Growth (% p.a.)	10.0%	10.0%					0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%	10.0%
9																	
10	Test China>=US:																
11	Using "C7>=C4"		FALSE					SE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
12	Using "=IF(C7>=C4,1,0)"		0					0	0	0	0	0	1	1	1	1	1
13																	
14	First year that China>=US:																
15	Using MATCH(TRUE,....)	11															
16	Using MATCH(1,....)	11															
17																	
18	Desired year for Goal Seek	8															
19	Difference	3.0															

Goal Seek

Set cell: C19

To value: 0

By changing cell: $C$8

OK Cancel

Figure 1.43

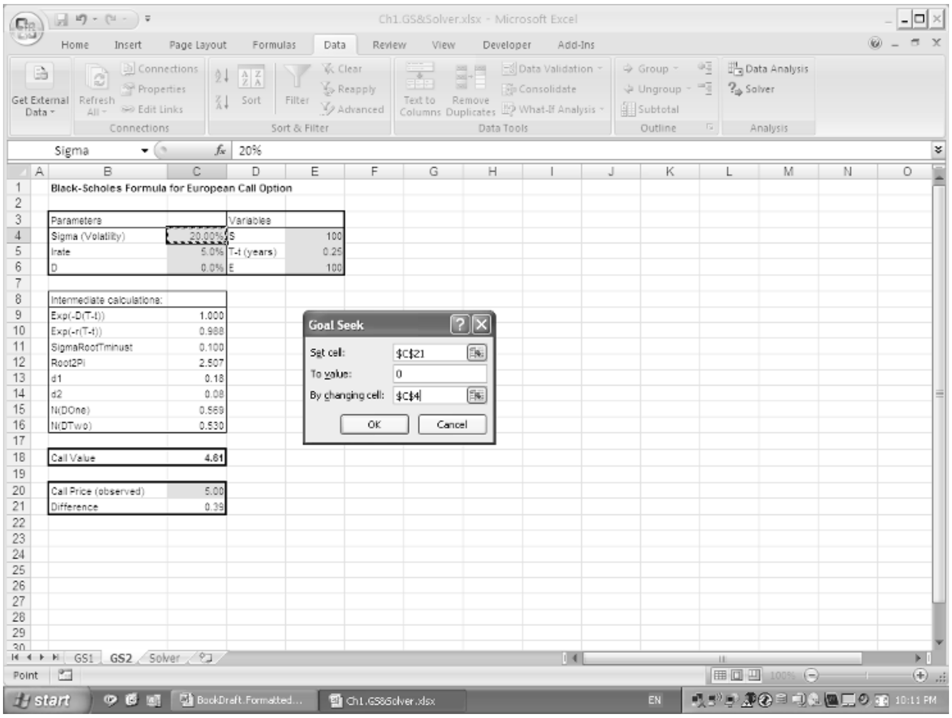
As mentioned earlier, add-ins can be installed (and uninstalled) using the **Manage** tool for add-ins under **Office/Excel Options/Add-ins** (or **Tools**/**Add-ins** in Excel 2003). Once loaded the add-ins may appear in several places in Excel (**Solver** and the **Analysis Tool-Pak** appear under **Data/Analysis**, the **Conditional Sum Wizard** appears under **Formulas/Solutions** and so on. In Excel 2003, they appear under **Tools**).

The Analysis ToolPak

The **Analysis ToolPak** provides statistical tools to enhance the capabilities of Excel. In Excel 2003, the add-in contains some functions that are directly included as worksheet functions in Excel 2007 (and are accessible without the add-in being loaded). Some of the most relevant for financial modelling include:

- **XNPV** calculates the net present value of non-equally spaced cash flows.
- **XIRR** calculates the internal rate-of-return of non-equally spaced cash flows.
- **YEARFRAC** calculates the fraction of the year corresponding to the number of days between two dates.
- **YIELD** calculates the yield on a bond or security that pays constant periodic interest.

The add-in is relatively straightforward to use providing one is familiar with the underlying statistical procedures. Most of these (e.g. analysis of variance, F-tests, t-tests, Fourier analysis, and so on) are beyond the scope of this text. For the purposes here, the most



Sigma | 20%

	A	B	C	D	E
1		Black-Scholes Formula for European Call Option			
2					
3		Parameters		Variables	
4		Sigma (Volatility)	20.00%	S	100
5		Irate	5.0%	T-t (years)	0.25
6		D	0.0%	E	100
7					
8		Intermediate calculations:			
9		Exp(-D(T-t))	1.000		
10		Exp(-r(T-t))	0.988		
11		SigmaRootTminust	0.100		
12		Root2Pi	2.507		
13		d1	0.18		
14		d2	0.08		
15		N(DOne)	0.569		
16		N(DTwo)	0.530		
17					
18		Call Value	4.61		
19					
20		Call Price (observed)	5.00		
21		Difference	0.39		

Goal Seek

Set cell: $C$21

To value: 0

By changing cell: $C$4

OK Cancel

Figure 1.44

directly relevant include the calculation of correlation coefficients (or covariances) and the calculation of multiple regression coefficients and related statistics. An example of the use of the **Correlation** option is provided below.

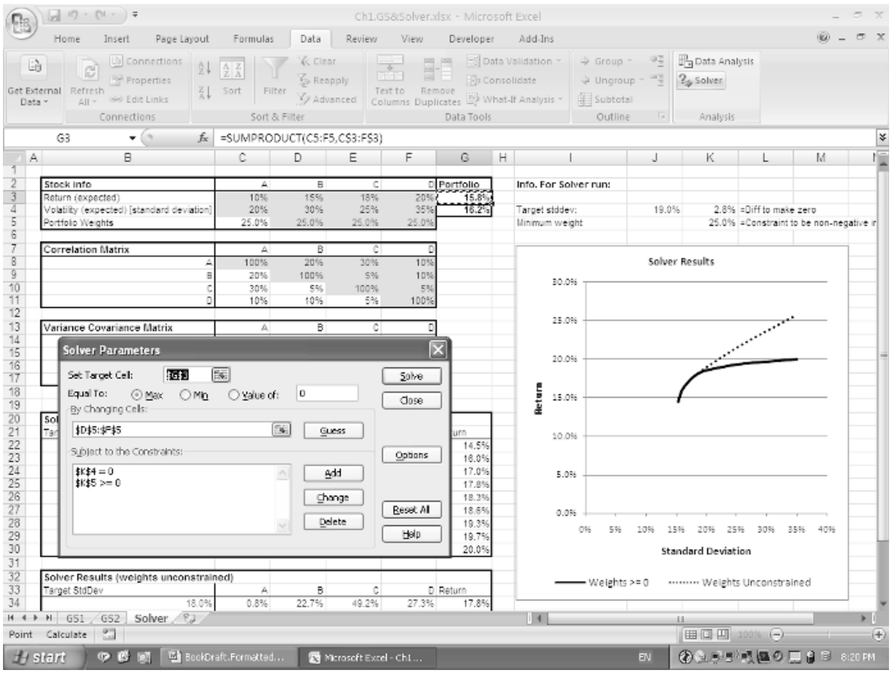
Example: Correlation Matrices Revisited

The **Correlation** tool within the **Analysis ToolPak** is especially useful to create correlation matrices for large sets of data, and provides an alternative to the earlier methods shown (which consisted of either manually setting up the formula for each element of the matrix or using **Lookup** functions to create a general formula that can be copied to all matrix elements.) One advantage of the use of the **Data Analysis** tool is to avoid the potential complexities of using the **Lookup** functions. On the other hand a disadvantage of this approach is that the correlation matrix is not live-linked and so needs to be updated if the data changes.

The file Ch1.OtherTools.xlsx (DataAnalysisCorr worksheet) (Figure 1.46) shows an example.

Example: Complex Conditional Sums using the Conditional Sum Wizard

The **Conditional Sum Wizard** can in some circumstances provide a more appropriate alternative to the use of **Database** functions, **PivotTables**, or the **SUMIF** function. (As mentioned earlier Excel 2007 also has a **SUMIFS** function in which a range is summed


G3 =SUMPRODUCT(C5:F5,C$3:F$3)

Stock Info	A	B	C	D	Portfolio
Return (expected)	10%	15%	18%	20%	15.8%
Volatility (expected) [standard deviation]	20%	30%	25%	35%	16.2%
Portfolio Weights	25.0%	25.0%	25.0%	25.0%	

Correlation Matrix	A	B	C	D
A	100%	20%	30%	10%
B	20%	100%	5%	10%
C	30%	5%	100%	5%
D	10%	10%	5%	100%

Variance Covariance Matrix A B C D

Info. For Solver run:

Target stddev:	19.0%	2.8%	=Diff to make zero
Minimum weight		25.0%	=Constraint to be non-negative i

Solver Parameters
Set Target Cell: $G$3
Equal To: [✓ Max] Min Value of: 0
By Changing Cells: $D$5:$F$5
Subject to the Constraints:
$K$4 = 0
$K$5 >= 0

Solver Results (weights unconstrained)					
Target StdDev	A	B	C	D	Return
18.0%	0.8%	22.7%	49.2%	27.3%	17.8%

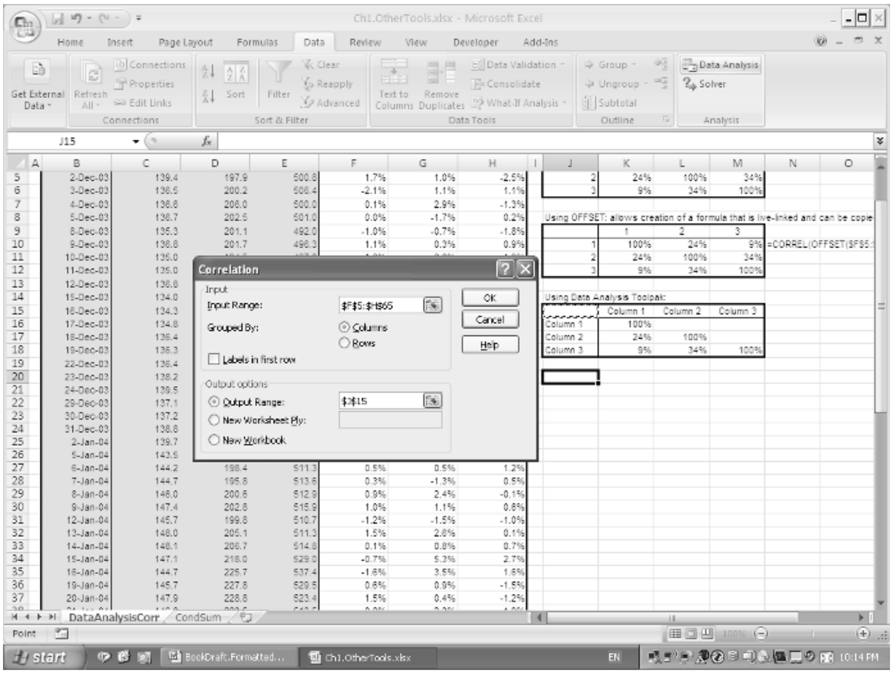
Figure 1.45

according to multiple criteria being met, and this can also be used in this context; an example is also shown.)

The wizard can be used to generate formulae with multiple embedded conditions in an automated way. The formula generated is an array function, therefore if it is edited or modified it will need to be entered using **CTRL+SHIFT+ENTER** (see earlier).

The step-by-step wizard is essentially self-explanatory. When on step three, the option **copy the formula and conditional values** can be chosen to ensure that the formula generated during step four refers to the value of particular conditions (which will be placed in separate individual cells); in this way, the formula is easier to adapt, copy and modify so that different conditional values can be tested, i.e. that multiple database queries can be set up. Note that if a different query is required (such as requiring three conditions to be satisfied instead of two), then it is generally easier to re-run the wizard than to try to adapt the formula directly. However, where the number of conditions to be satisfied is unchanged, then often a direct adaptation of the formula is easy to achieve (e.g. adapting a base formula which tests for two conditions so that two different conditions are satisfied).

The file Ch1.OtherTools.xlsx (CondSum worksheet) (Figure 1.47) shows an example of the result of applying the wizard in this way. The results are compared with the use of the **DSUM** formula shown earlier and with the use of the **SUMIFS** function. Note that the formula can be copied to different cells so that different criteria can be created, whereas the use of the **DSUM** function would require a new criteria range for each application of the function.



	B	C	D	E	F	G	H
5	2-Dec-03	139.4	197.9	500.8	1.7%	1.0%	-2.5%
6	3-Dec-03	138.5	200.2	506.4	-2.1%	1.1%	1.1%
7	4-Dec-03	138.8	206.0	500.0	0.1%	2.9%	-1.3%
8	5-Dec-03	138.7	202.5	501.0	0.0%	-1.7%	0.2%
9	8-Dec-03	135.3	201.1	492.0	-1.0%	-0.7%	-1.8%
10	9-Dec-03	138.8	201.7	496.3	1.1%	0.3%	0.9%
11	10-Dec-03	135.0					
12	11-Dec-03	135.0					
13	12-Dec-03	138.8					
14	15-Dec-03	134.0					
15	16-Dec-03	134.3					
16	17-Dec-03	134.8					
17	18-Dec-03	136.4					
18	19-Dec-03	136.3					
19	22-Dec-03	136.4					
20	23-Dec-03	138.2					
21	24-Dec-03	139.5					
22	29-Dec-03	137.1					
23	30-Dec-03	137.2					
24	31-Dec-03	138.8					
25	2-Jan-04	139.7					
26	5-Jan-04	143.5					
27	6-Jan-04	144.2	198.4	511.3	0.5%	0.5%	1.2%
28	7-Jan-04	144.7	195.8	513.6	0.3%	-1.3%	0.5%
29	8-Jan-04	146.0	200.6	512.9	0.9%	2.4%	-0.1%
30	9-Jan-04	147.4	202.8	515.9	1.0%	1.1%	0.6%
31	12-Jan-04	145.7	199.8	510.7	-1.2%	-1.5%	-1.0%
32	13-Jan-04	148.0	205.1	511.3	1.5%	2.6%	0.1%
33	14-Jan-04	148.1	206.7	514.8	0.1%	0.8%	0.7%
34	15-Jan-04	147.1	218.0	529.0	-0.7%	5.3%	2.7%
35	16-Jan-04	144.7	225.7	537.4	-1.6%	3.5%	1.6%
36	19-Jan-04	145.7	227.8	529.5	0.6%	0.9%	-1.5%
37	20-Jan-04	147.9	228.8	523.4	1.5%	0.4%	-1.2%

J	K	L	M
2	24%	100%	34%
3	9%	34%	100%

Using OFFSET: allows creation of a formula that is live-linked and can be copie

	1	2	3
1	100%	24%	9%
2	24%	100%	34%
3	9%	34%	100%

=CORREL(OFFSET($F$5:

Using Data Analysis Toolpak:

	Column 1	Column 2	Column 3
Column 1	100%		
Column 2	24%	100%	
Column 3	9%	34%	100%

Correlation

Input

Input Range: $F$5:$H$65

Grouped By: [✓ Columns] Rows

Labels in first row

Output options

[✓ Output Range:] $J$15

New Worksheet Ply:

New Workbook

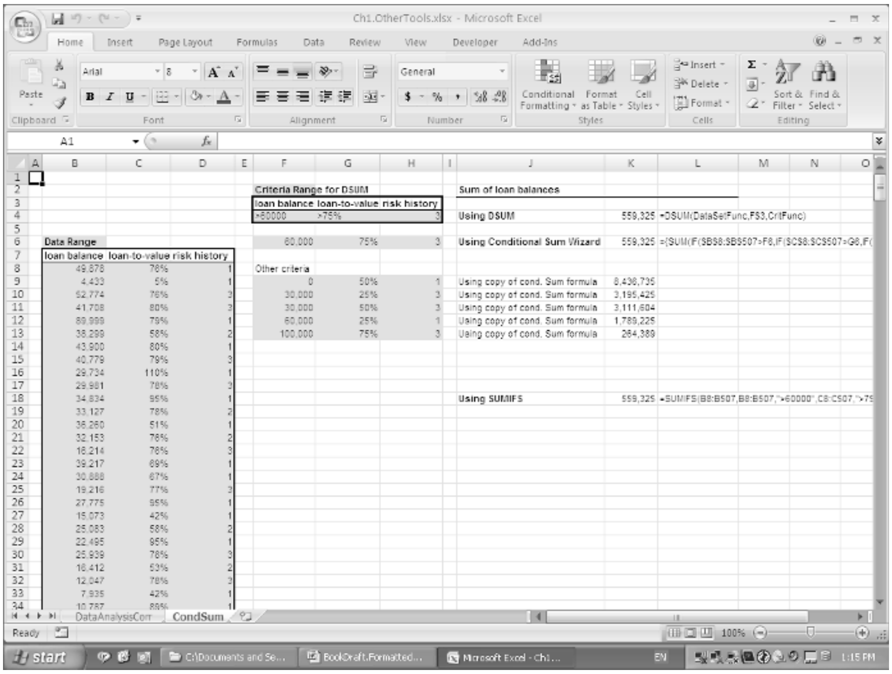
Figure 1.46

SELECTED EXCEL SHORT-CUTS

Many Excel operations can be performed quickly using keyboard short-cuts. On occasion, an excessive use of short-cuts may lead to a lack of robustness in the modelling process, with too much focus placed on the rapid building of the model, rather than on the modelling itself. It is perhaps to be recommended that the individual modeller should therefore develop a good familiarity with those short-cuts that will be most relevant, but not to be unduly concerned with knowledge of the full range of possible short-cuts.

Some core short-cuts and related tools that the author uses regularly include:

- **CTRL+C** to copy a cell, **CTRL+V** to paste, **CTRL+X** to cut, and **CTRL+Z** to undo.
- **CTRL+1** to display the **Format Cells** menu.
- **Format Painter** (on the **Home** tab) to format one cell as another. Double-clicking on the icon will ensure that it remains active so that it can be applied to multiple ranges in sequence (and clicking the icon again will deactivate it).
- **F4** to implement absolute cell references when editing a formula in the **Formula bar** (e.g. place a $ before a row reference); repeated pressing of the key will cycle through all possible row and column combinations.
- **CTRL+SHIFT+ARROW** to select the range from the current cell to end of range in the direction of the arrow (**CTRL+ARROW** to move to the end of current region in the direction of the arrow without selecting the intermediate cells).
- **CTRL+ENTER** to enter the same value in all cells of an already selected range.
- **F1** (help); **F2** (editing formulae); **F7** (check spelling); **Ctrl+F** (find).



	A	B	C	D	E	F	G	H	I	J	K	L
1												
2						Criteria Range for DSUM				Sum of loan balances		
3						loan balance	loan-to-value	risk history				
4						>60000	>75%	3		Using DSUM	559,325	=DSUM(DataSetFunc,F$3,CritFunc)
5												
6		Data Range				60,000	75%	3		Using Conditional Sum Wizard	559,325	={SUM(IF($B$8:$B$507>F6,IF($C$8:$C$507>G6,IF(
7		loan balance	loan-to-value	risk history								
8		49,878	78%	1		Other criteria						
9		4,433	5%	1		0	50%	1		Using copy of cond. Sum formula	8,438,735	
10		52,774	76%	3		30,000	25%	3		Using copy of cond. Sum formula	3,195,425	
11		41,708	80%	3		30,000	50%	3		Using copy of cond. Sum formula	3,111,604	
12		89,999	79%	1		60,000	25%	1		Using copy of cond. Sum formula	1,789,225	
13		38,298	58%	2		100,000	75%	3		Using copy of cond. Sum formula	264,389	
14		43,900	80%	1								
15		40,779	79%	3								
16		29,734	110%	1								
17		29,981	78%	3								
18		34,834	95%	1						Using SUMIFS	559,325	=SUMIFS(B8:B507,B8:B507,">60000",C8:C507,">79
19		33,127	78%	2								
20		36,260	51%	1								
21		32,153	76%	2								
22		16,214	76%	3								
23		39,217	69%	1								
24		30,888	67%	1								
25		19,216	77%	3								
26		27,775	95%	1								
27		15,073	42%	1								
28		25,083	58%	2								
29		22,495	95%	1								
30		25,939	78%	3								
31		16,412	53%	2								
32		12,047	78%	3								
33		7,935	42%	1								
34		10,787	88%	1								

Figure 1.47

- **F3** to paste a list of all named ranges (includes those whose scope is the workbook and the particular worksheet, but not those whose scope is only another worksheet; see Chapter 2 for more information).
- **F5** to go to a cell or range; particularly useful if there are named ranges in the worksheet, and also using the **Special** option to find formulae, constants, etc.; see Chapter 2 for more information.
- **Home** to move to column A; **CTRL+Home** to move to cell A1.
- Use of the **AutoSum** icon (on status toolbar) to check sums, averages, count, etc., and hence readily perform extra cross-checks to the model.
- In Excel 2007, the **KeyTips (ToolTips)** can be displayed by pressing **ALT**. This shows the letters that may be typed to directly access the menu. For example, **ALT** followed by **M** (for **Formulas**) and then **H** (**Show Formulas**) will display the formulae (equivalent to **CTRL+'**).