

Tips for advanced Spreadsheets

With MS-Excel 2003-2007

Francis Wellens

NAO Office

Training of 1 to 2 December 2010

TABLE OF CONTENTS

1	Introduction.	4
1.1	<i>The Spreadsheet Concept</i>	4
1.2	<i>Data and formulas entering</i>	4
1.2.1	Entering Data	4
1.2.2	Entering formulas	4
1.2.3	Entering conditional formulas	5
1.2.4	Using the file handle	5
1.3	<i>Absolute and relative references</i>	6
1.3.1	Relative References	6
1.3.2	Absolute References	6
2	Advanced worksheet functions	7
2.1	<i>Names</i>	7
2.1.1	Rules for defining names	7
2.1.2	Defining Names	8
2.1.3	Creating Names Automatically.	8
2.1.4	Managing Names .	8
2.2	<i>Using names</i>	8
2.2.1	Go To	8
2.2.2	Names in formulae	8
2.3	<i>Worksheet names</i>	8
2.4	<i>3-D formulae</i>	9
2.5	<i>Conditional formating</i>	9
2.6	<i>Conditional & logical functions</i>	9
2.6.1	If Statements	10
2.6.2	Logical Test	10
2.6.3	Value If True / False .	10
2.6.4	Nested If	11
2.7	<i>Statistical If Statements</i>	12
2.7.1	And, or, not	12
2.7.2	And	13
2.7.3	Or	13
2.7.4	Not	14
2.8	<i>Lookup functions</i>	14
2.8.1	Lookup.	14
2.8.2	Vector Lookup	14
2.8.3	Hlookup	16
2.8.4	Vlookup	17
2.8.5	Nested Lookups	17
2.9	<i>Other useful functions</i>	18
2.9.1	ISERROR	18
3	Goal seek and solver	18
3.1	<i>Goal seeking and solving</i>	18
3.1.1	Goal Seek.	18
3.1.2	Goal-Seeking with Charts	19
3.1.3	Solver	19
3.1.4	Solver Parameters	19
3.1.5	Build constraints into your Solver parameters:	19
3.1.6	Solver and Scenario Manager	19
3.1.7	Opening Scenario Manager	19

4	Using excel to manage lists	19
4.1	<i>Excel lists,list terminology</i>	19
4.1.1	Row And Column Content	19
4.1.2	Column Labels	19
4.1.3	List Size And Location	19
4.1.4	Miscellaneous	19
4.2	<i>Sorting data .</i>	19
4.2.1	Custom Sorting Options	19
4.2.2	Creating A Custom Sort Order	19
4.3	<i>Adding subtotals to a list</i>	19
5	Structuring of Worksheets	19

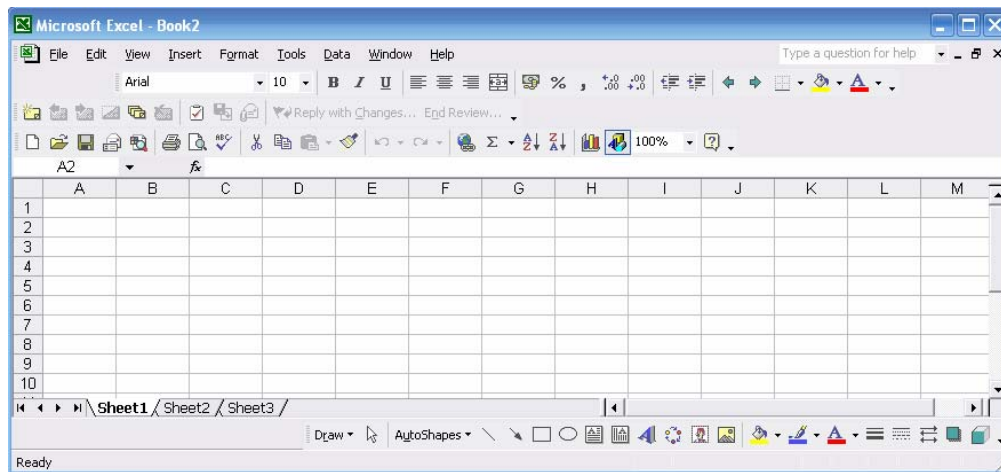
1 INTRODUCTION.

As introduction we will have a small reminder about the basics of spreadsheets in general and Excel in particular

1.1 THE SPREADSHEET CONCEPT

A computer spreadsheet is similar to a very large piece of paper, which is ruled into rows and columns. The intersection of a row and a column is called a cell and each cell has its own unique reference, similar to a map reference.

A spreadsheet can hold a variety of different types of data types, and is generally used when calculations need to be performed. The power of a computer spreadsheet is its ability to recalculate formulae whenever the data is changed. This saves a great deal of time and allows the user to create different results easily. Graphing data, printing, formatting data is all made much easier using a spreadsheet.



1.2 DATA AND FORMULAS ENTERING

1.2.1 Entering Data

Excel recognises text and numeric entries and initially displays them with different alignments – left for text and right for numbers. You can override these with other formats if required.

Until you confirm an entry, Excel remains in "Enter" mode, (see Status bar). Excel will return to the "Ready" mode, and the text will appear in the cell. When you press [ENTER] to confirm an entry, Excel may move the active cell down to the cell below. You can disable this setting or choose to move the active cell in a different direction using the Options dialog (Tools menu).

You may find that you have typed an entry into the wrong cell. Provided you have not confirmed the entry by pressing [ENTER] or clicking the green tick from the formula bar, you can abandon it.

1.2.2 Entering formulas

In a spreadsheet application, at a very basic level, values often need to be added, subtracted, multiplied and divided. To allow for the fact that individual values might change, spreadsheet formulae generally refer not to actual values, but to the cells where those values are being held. If values have been entered into A1 and A2, then


A1+A2 will return an answer which will automatically recalculate if the value of A1 should change. It is this automatic recalculation which makes spreadsheets invaluable.

Excel recognises formulae because they are preceded by an equals sign (=).

You enter formulae by typing them in the cell where you want the formula's result to appear. You can type the entire formula or use the pointer to indicate cells you want to incorporate

When you confirm entry of a formula, Excel will display the result on the worksheet, but the underlying calculation appears on the Formula bar.

Many methods are available for entering functions e.g.:

- = A2+A3+A4
- = Sum(A1..A4)
- using (autosum) Σ
- Utilising the "insert function" command 

1.2.3 Entering conditional formulas

=**SUMIF**(range,criteria,sum_range)

- **Range** is the range of cells you want to test.
- **Criteria** is the criteria in the form of a number, expression, or text that defines which cells will be added. For example, criteria can be expressed as 32, "32", ">32", "apples".
- **Sum range** are the actual cells to sum. The cells in sum range are summed only if their corresponding cells in range match the criteria. If sum range is omitted, the cells in range are summed.

	A	B	C	D	E	F
1	Order date	Customer	Product	Quantity	Price	Total
2	01/01/97	Viking Supplies	Widget	10	5	50
3	02/01/97	Bloggs & Co	Ratchet	23	7	161
4	02/01/97	Jones Brothers	Gimlet	200	3	600
5	03/01/97	Viking Supplies	Gimlet	201	3	603
6	01/01/97	Viking Supplies	Widget	296	5	1480
7	02/01/97	Bloggs & Co	Ratchet	371	7	2597
8	02/01/97	Jones Brothers	Gimlet	446	3	1338
9	03/01/97	Viking Supplies	Gimlet	521	3	1563
10	01/01/97	Viking Supplies	Widget	596	5	2980
11	02/01/97	Bloggs & Co	Ratchet	671	7	4697
12						
13		Viking Supplies Total				
14		Bloggs & Co No. of orders				

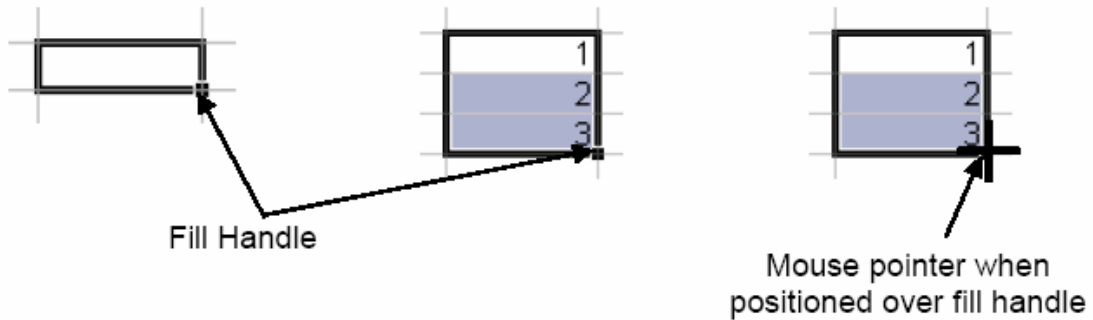
=SUMIF(B2:B11, "Viking Supplies", F2:F11)

=**COUNTIF**(range,criteria)

e.g. =COUNTIF(B2:B11, "Bloggs & Co.")

1.2.4 Using the fill handle

The fill handle is a very useful tool in Excel. It allows you to copy and clear data but also to fill in series of data (dates, weekdays etc.). Those aspects of the fill handle are dealt with later in this manual. The fill handle appears in the bottom right hand corner of the active cell or selection. When your mouse is over the fill handle, the white plus pointer changes to a black plus.



1.3 ABSOLUTE AND RELATIVE REFERENCES

1.3.1 Relative References

When you fill formulae, you tell Excel to base the formulae it creates on the one you have entered by starting from the cell that contains it.

In the example shown left, the formula being copied says $=D2 * E2$. However, once copied, if you click on any of the copies, Excel will have updated the references to keep the row numbers current ($D3 * E3$, $D4 * E4$ and so on). This is because Excel, by default, uses relative referencing.

fx =D2*E2		
D	E	F
Quantity	Price	Total
10	5	50
23	7	
200	3	
201	3	
296	5	
371	7	
446	3	
521	3	
596	5	
671	7	

When you enter a formula, you enter specific cell references. Behind the scenes, Excel ‘translates’ those references into positions relative to the result cell. So, in our example shown left, Excel would take the formula: $=D2 * E2$ and translate it as follows: $=[\text{two cells to the left}] * [\text{one cell to the left}]$. It therefore does not matter which row you copy the formula into, Excel will always use “[two cells to the left] multiplied by [one cell to the left]” to generate the result.

Most of the time, this is what you would want, but there are occasions where you need to stop Excel updating cell references when you copy formulae.

1.3.2 Absolute References

The example below shows a formula being created to work out the amount of discount each order would receive. The order totals are in column F and the discount rate is in B13. The initial formula has therefore been set up as: $=F2 * B13$

SUM X ✓ fx =F2*B13							
	A	B	C	D	E	F	H
1	Order Date	Customer	Product	Quantity	Price	Total	Discount Amount
2	01/01/2001	Viking Supplies	Widget	10	5	50	$=F2 * B13$
3	02/01/2001	Bloggs & Co	Ratchet	23	7	161	
4	03/01/2001	Jones Brothers	Gimlet	200	3	600	
5	04/01/2001	Viking Supplies	Gimlet	201	3	603	
6	05/01/2001	Viking Supplies	Widget	296	5	1480	
7	06/01/2001	Bloggs & Co	Ratchet	371	7	2597	
8	07/01/2001	Jones Brothers	Gimlet	446	3	1338	
9	08/01/2001	Viking Supplies	Gimlet	521	3	1563	
10	09/01/2001	Viking Supplies	Widget	596	5	2980	
11	10/01/2001	Bloggs & Co	Ratchet	671	7	4697	
12							
13	Discount Rate	25%					

The formula will generate a result for the first order. However, when copied, you will get zeros against the discount amounts for the other orders. This is due to the relative referencing that Excel applies to all formulae by default. Having copied the above formula, if you clicked on any formula in the Discount amount column below the first one, you would see that Excel has updated the references (=F3*B14, F4*B15).

This is where the problem lies – you want Excel to change the first reference as your formula needs to refer to the different order totals, but the discount rate should remain constant.

You need to make that reference *absolute*.

- Select the cell containing the relative reference you want to make absolute
- Select the reference
- Press F4 as often until you obtain the reference suiting your case

Pressing [F4] repeatedly over a reference allows you to toggle between making both the row and column absolute (\$B\$13), just the column absolute (\$B13), just the row (B\$13), or nothing absolute (B13).

2 ADVANCED WORKSHEET FUNCTIONS

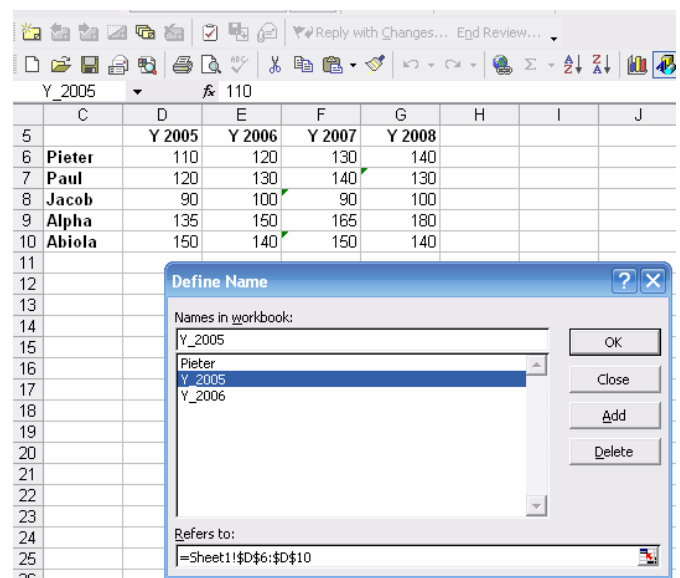
2.1 NAMES

When entering formulae or referring to any area on the spreadsheet, it is usual to refer to a "range". For example, B6 is a range reference; B6:B10 is also a range reference. A problem with this sort of reference is that it is not always easy to remember what cells to reference. It may be necessary to write down the range, or select it, which often means wasting time scrolling around the spreadsheet. Instead, Excel offers the chance to name ranges on the spreadsheet, and to use these names to select cells, refer to them in formulae or use them in Database, Chart or Macro commands.

2.1.1 Rules for defining names

Names are unique within a workbook and the names that you choose to use must adhere to certain rules.

- The first character of a name must be a letter or an underscore character. Remaining characters in the name can be letters, numbers, full stops, and underscore characters.
- Names cannot be the same as a cell reference, such as Z\$100 or R1C1.
- Spaces are not allowed. Underscore characters and full stops may be used as word separators - for example, First.Quarter or Sales_Tax.
- A name can contain up to 255 characters.
- Names can contain uppercase and lowercase letters. Microsoft Excel does not distinguish between uppercase and lowercase characters in names. For example, if you have created the name Sales and then create another name called SALES in



the same workbook, the second name will replace the first one.

2.1.2 Defining Names

There are a number of ways to set up names on a spreadsheet after selecting the field:

Insert → Name → Define. In the example, there is a range of sales figures that could be named “Y2005”;

Or directly typing the name at the left of the formula bar

2.1.3 Creating Names Automatically.

Menu Insert → Name → Create

In Office 2007 click the create from selection button on the in the defined names group on the formulas ribbon

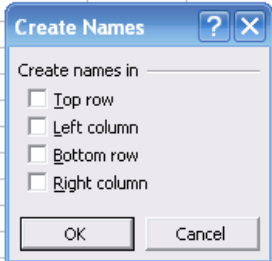
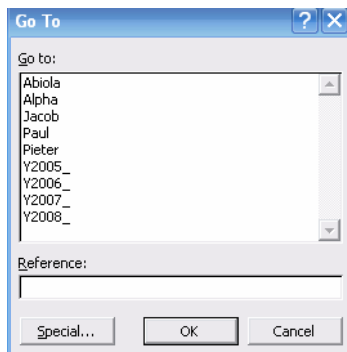
2.1.4 Managing Names .

In 2003, with the name definition,

In 2007, with the Name Manager

Click the name manager button on the in the defined names group on the formulas ribbon.

	B	C	D	E	F
4		Y2005	Y2006	Y2007	Y2008
5	Pieter	110	120	130	140
6	Paul	120	130	140	130
7	Jacob	90	100	90	100
8	Alpha	135	150	165	180
9	Abiola	150	140	150	140
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					

2.2 USING NAMES

2.2.1 Go To

2003 Menu Edit → Go To (or CTL+G or F5)

2007 HOME menu → FIND & SELECT in the EDITING GROUP → GO TO (or F5)

→selects all the cells of the selected name

To move to a cell that belongs to two ranges: Press [F5] and type the first range name in the Reference box, then

type a space and the second range name.

2.2.2 Names in formulae

Names can be used in any simple formula, as well as any of Excel's built-in functions. Instead of typing cell references or selecting cells, simply type the name or paste the name into the formula. Ex: Average = Sum(Paul)/count(Paul)

To paste a name into a formula:

2003: Menu Insert → Name → Paste.(F3)

2007: formulas ribbon → defined names → use in formula (F3)



2.3 WORKSHEET NAMES

Excel assigns the names “Sheet 1, Sheet 2” and so on to worksheets in a workbook but you should overwrite them with more meaningful names to describe the data they contain.

2.4 3-D FORMULAE

When you have data on multiple worksheets, you often want a summary page that will collect data from cells across all the sheets and calculate it in a certain way. You can do this with a 3-D formula. 3-D formulae are no different from the formulae already discussed in an earlier section of this manual, however, the way you refer to cells on different worksheets has not yet been explored.

When you refer to a cell that is in another worksheet, Excel uses the following syntax:

- **Single cell** : 'Worksheet name'![Cell reference]
- **Cells on adjacent worksheet**: '1st Worksheet name:last Worksheet name'![Cell reference]
- **Cells on non-adjacent worksheets**: '1st Worksheet name'![Cell reference], '2nd Worksheet name'![Cell reference]

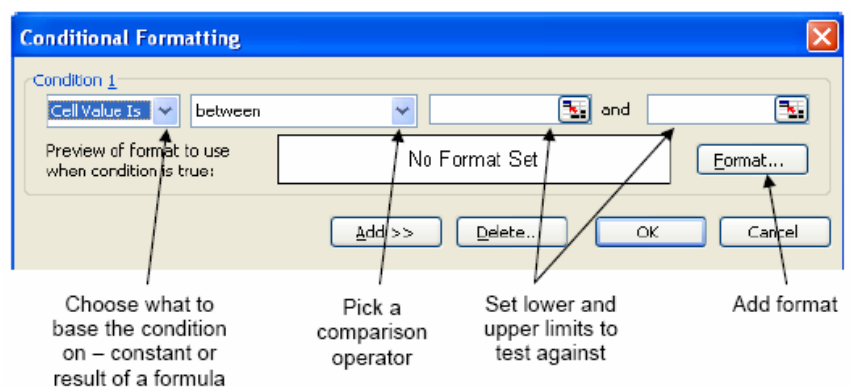
2.5 CONDITIONAL FORMATING

You can make Excel format cells in certain ways according to whether their data meets a condition. For example, in a table of numbers, you might want to display numbers over 100 in red, numbers below 50 in green and anything in-

between in blue. You can use conditional formats to achieve this.

- 2003 menu: Format→ conditional formatting

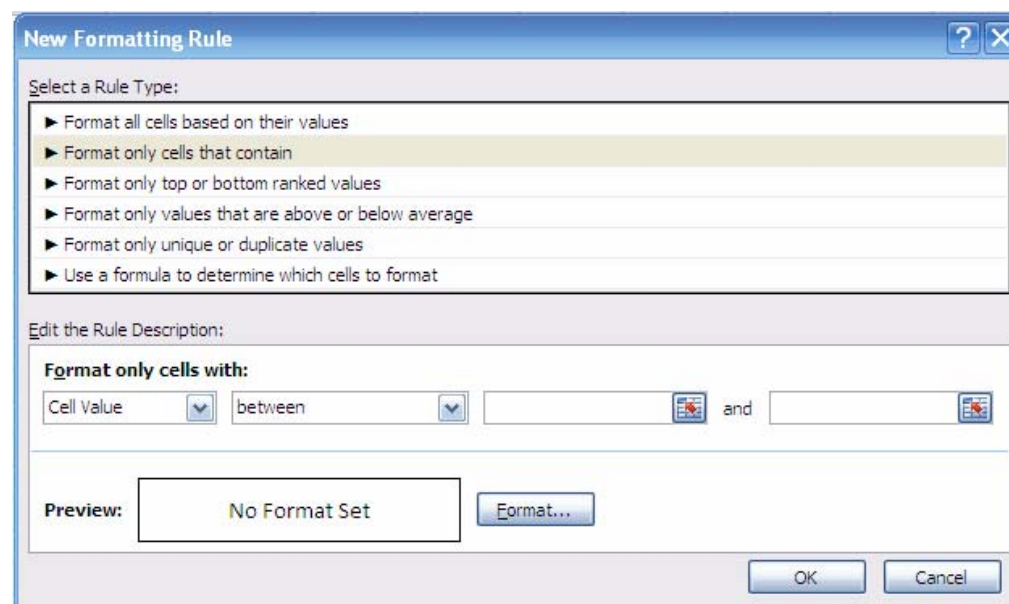
2007 home ribbon→ Styles→ Conditional formatting→ New rule



between in blue. You can use conditional formats to achieve this.

- 2003 menu: Format→ conditional formatting

2007 home ribbon→ Styles→ Conditional formatting→ New rule



2.6 CONDITIONAL & LOGICAL FUNCTIONS

Excel has a number of logical functions which allow you to set various "conditions" and have data respond to them. For example, you may only want a certain calculation

performed or piece of text displayed if certain conditions are met. The functions used to produce this type of analysis are found in the menu Function →Insert→ LOGICAL.

2.6.1 If Statements

The IF function is used to analyse data, test whether or not it meets certain conditions and then act upon its decision. The formula can be entered either by typing it or by pasting it from the Formula, Paste Function box. Typically, the IF statement is accompanied by three arguments enclosed in one set of parentheses; the condition to be met (logical_test); the action to be performed if that condition is true (value_if_true); the action to be performed if false (value_if_false). Each of these is separated by a comma, as shown;

=IF(logical_test , value_if_true , value_if_false)

2.6.2 Logical Test

2003: Menu: Insert→Function menu → LOGICAL category →IF function

2007: formulas ribbon →function library → logical

>	greater than
<	less than
> =	greater than or equal to
< =	less than or equal to
=	equal to
< >	not equal to

Therefore, a typical logical test might be B1 > B2, testing whether or not the value contained in cell B1 of the spreadsheet is greater than the value in cell B2. Names can also be included in the logical test, so if cells B1 and B2 were respectively named SALES and TARGET, the logical test would read SALES > TARGET. Another type of logical test could include text strings. If you want to check a cell to see if it contains text, that text string must be included in quotation marks. For example, cell C5 could be tested for the word YES as follows; C5="YES".

It should be noted that Excel's logic is, at times, brutally precise. In the above example, the logical test is that sales should be greater than target. If sales are equal to target, the IF statement will return the false value. To make the logical test more flexible, it would be advisable to use the operator > = to indicate "meeting or exceeding".

2.6.3 Value If True / False .

Provided that you remember that TRUE value always precedes FALSE value, these two values can be almost anything. If desired, a simple number could be returned, a calculation performed, or even a piece of text entered. Also, the type of data entered can vary depending on whether it is a true or false result. You may want a calculation if the logical test is true, but a message displayed if false. (Remember that text to be included in functions should be enclosed in quotes).

Taking the same logical test mentioned above, if the sales figure meets or exceeds the target, a BONUS is calculated (e.g. 2% of sales). If not, no bonus is calculated so a

value of zero is returned. The IF statement in column D of the example reads as follows:

=IF(B2>=C2,B2*2%,0)

	D2		=IF(B2>=C2,B2*2%,0)		
	A	B	C	D	E
1	Name	Sales	Target	Bonus	
2	Bill	580	640	0	
3	Karen	1200	1000	24	
4	Ted	710	700	14.2	
5					

value will remain the same and the false value will be the text string "NO BONUS";

=IF(B2>=C2,B2*2%,"NO BONUS")

A particularly common use of IF statements is to produce "ratings" or "comments" on figures in a spreadsheet. For this, both the true and false values are text strings. For example, if a sales figure exceeds a certain amount, a rating of "GOOD" is returned, otherwise the rating is "POOR";

=IF(B2>1000,"GOOD","POOR")

	D2		=IF(B2>1000,"Good","Poor")		
	A	B	C	D	E
1	Name	Sales	Target	Rating	
2	Bill	580	640	Poor	
3	Karen	1200	1000	Good	
4	Ted	710	700	Poor	
5					
6					

2.6.4 Nested If

When you need to have more than one condition and more than two possible outcomes, a NESTED IF is required. This is based on the same principle as a normal IF statement, but involves "nesting" a secondary formula inside the main one. The secondary IF forms the FALSE part of the main statement, as follows;

=IF(1st logic test , 1st true value , IF(2nd logic test , 2nd true value , false value))

Only if both logic tests are found to be false will the false value be returned. Notice that there are two sets of parentheses, as there are two separate IF statements. This process can be enlarged to include more conditions and more eventualities - up to seven IFs can be nested within the main statement. However, care must be taken to ensure that the correct number of parentheses are added. In the example, sales staff could now receive one of three possible ratings ;

=IF(B2>1000,"GOOD",IF(B2<600,"POOR","AVERAGE"))

To make the above IF statement more flexible, the logical tests could be amended to measure sales against cell references instead of figures. In the example, column E has been used to hold the upper and lower sales thresholds.

=IF(B2>=\$E\$2,"GOOD",IF(B2<=\$E\$3,"POOR","AVERAGE"))

(If the IF statement is to be copied later, this cell reference should be absolute).

	A	B	C	D	E	F	G
D2					=IF(B2>\$E\$2,"Good",IF(B2<\$E\$3,"Poor","Average"))		
1	Name	Sales	Target	Ratings	Thresholds		
2	Bill	580	640	Poor	1000		
3	Karen	1200	1000	Good	600		
4	Ted	710	700	Average			
5							
6							
7							

With 2003, max 7 nesting levels

With 2007, max 64 nesting levels.

2.7 STATISTICAL IF STATEMENTS

A very useful technique is to display text or perform calculations only if a cell is the maximum or minimum of a range. In this case the logical test will contain a nested statistical function (such as MAX or MIN). If, for example, a person's sales cell is the maximum in the sales column, a message stating "Top Performer" could appear next to his or her name. If the logical test is false, a blank message could appear by simply including an empty set of quotation marks. When typing the logical test, it should be understood that there are two types of cell referencing going on. The first is a reference to one person's figure, and is therefore relative. The second reference represents the RANGE of everyone's figures, and should therefore be absolute.

=IF(relative cell = MAX(absolute range) , "Top Performer" , "")

	A	B	C	D	E
B2					
					=IF(C2=MAX(\$C\$2:\$C\$4),"Top Performer", "")
1	Name	Message	Sales		
2	Bill		580		
3	Karen	Top Performer	1200		
4	Ted		710		
5					

In this example the IF statement for cell B2 will read;

=IF(C2=MAX(\$C\$2:\$C\$4),"Top Performer", "")

When this is filled down through cells B3 and B4, the first reference to the individual's sales figure changes, but the reference to all three sales figures (\$C\$2:\$C\$4) should remain constant. By doing this, you ensure that the IF statement is always checking to see if the individual's figure is the biggest out of the three.

A further possibility is to nest another IF statement to display a message if a value is the minimum of a range. Beware of syntax here - the formula could become quite unwieldy!

2.7.1 And, or, not

Rather than create large and unwieldy formulae involving multiple IF statements, the AND, OR and NOT functions can be used to group logical tests or "conditions" together. These three functions can be used on their own, but in that case they will only return the values "TRUE" or "FALSE". As these two values are not particularly meaningful on a spreadsheet, it is much more useful to combine the AND, OR and NOT functions within an IF statement. This way, you can ask for calculations to be performed or other text messages to appear as a result.

2.7.2 And

This function is a logical test to see if all conditions are true. If this is the case, the value "TRUE" is returned. If any of the arguments in the AND statement are found to be false, the whole statement produces the value "FALSE". This function is particularly useful as a check to make sure that all conditions you set are met.

Arguments are entered in the AND statement in parentheses, separated by commas, and there is a maximum of 30 arguments to one AND statement. The following example checks that two cells, B1 and B2, are both greater than 100.

```
=AND(B1>100,B2>100)
```

If either one of these two cells contains a value less than a hundred, the result of the AND statement is "FALSE". This can now be wrapped inside an IF function to produce a more meaningful result. You may want to add the two figures together if they are over 100, or display a message indicating that they are not high enough.

```
=IF(AND(B1>100,B2>100),B1+B2,"Figures not high enough")
```

	B3		=IF(AND(B1>100,B2>100),B1+B2,"Figures not high enough")					
	A	B	C	D	E	F	G	H
1		50						
2		30						
3		Figures not high enough						
4								

Another application of ANDs is to check that a number is between certain limits. The following example checks that a number is between 50 and 100. If it is, the value is entered. If not, a message is displayed;

```
=IF(AND(B1>50,B1<100),B1,"Number is out of range")
```

2.7.3 Or

This function is a logical test to see if one or more conditions are true. If this is the case, the value "TRUE" is returned. If just one of the arguments in the OR statement is found to be true, the whole statement produces the value "TRUE". Only when all arguments are false will the value "FALSE" be returned. This function is particularly useful as a check to make sure that at least one of the conditions you set is met.

```
=IF(OR(B1>100,B2>100),B1+B2,"Figures not high enough")
```

In the above formula, only one of the numbers in cells B1 and B2 has to be over 100 in order for them to be added together. The message only appears if neither figure is high enough.

	B3		=IF(OR(B1>100,B2>100),B1+B2,"Figures not high enough")					
	A	B	C	D	E	F	G	H
1		50						
2		30						
3		Figures not high enough						
4								

This function is a logical test to see if one or more conditions are true. If this is the case, the value "TRUE" is returned. If just one of the arguments in the OR statement is found to be true, the whole statement produces the value "TRUE". Only when all arguments are false will the value "FALSE" be returned. This function is particularly useful as a check to make sure that at least one of the conditions you set is met.

```
=IF(OR(B1>100,B2>100),B1+B2,"Figures not high enough")
```

In the above formula, only one of the numbers in cells B1 and B2 has to be over 100 in order for them to be added together. The message only appears if neither figure is high enough.

B3		=IF(OR(B1>100,B2>100),B1+B2,"Figures not high enough")						
	A	B	C	D	E	F	G	H
1		50						
2		30						
3		Figures not high enough						
4								

2.7.4 Not

NOT checks to see if the argument is false. If so, the value "TRUE" is returned. It is best to use NOT as a "provided this is not the case" function. In other words, so long as the argument is false, the overall statement is true. In the example, the cell contents of B1 is returned unless the number 13 is encountered. If B1 is found to contain 13, the message "Unlucky!" is displayed;

=IF(NOT(B1=13),B1,"Unlucky!")

The NOT function can only contain one argument. If it is necessary to check that more than one argument is false, the OR function should be used and the true and false values of the IF statement reversed. Suppose, for example, a check is done against the numbers 13 and 666;

=IF(OR(B1=13,B1=666),"Unlucky!",B1)

B3		=IF(OR(B1=13,B1=666),"Unlucky!",B1)				
	A	B	C	D	E	F
1		13				
2		99				
3		Unlucky!				
4						

2.8 LOOKUP FUNCTIONS

As already mentioned, Excel can produce varying results in a cell, depending on conditions set by you. For example, if numbers are above or below certain limits, different calculations will be performed and text messages displayed. The usual method for constructing this sort of analysis is using the IF function. However, as already demonstrated, this can become large and unwieldy when you want multiple conditions and many possible outcomes. To begin with, Excel can only nest seven IF clauses (64 for 2007) in a main IF statement, whereas you may want more than eight logical tests or "scenarios". To achieve this, Excel provides some LOOKUP functions. These functions allow you to create formulae which examine large amounts of data and find information which matches or approximates to certain conditions. They are simpler to construct than nested IFs and can produce many more varied results.

2.8.1 Lookup.

Before you actually start to use the various LOOKUP functions, it is worth learning the terms that you will come across, what they mean and the syntax of the function arguments.

2.8.2 Vector Lookup

A vector is a series of data that only occupies one row or column. LOOKUP will look through this row or column to find a specific value. When the value is found, a

corresponding "result" in the adjacent row or column is returned. For example, column D of a spreadsheet may contain figures, and the adjacent column E contain corresponding text. LOOKUP will search for the requested figure in column D and return the corresponding text from column E.

The syntax for LOOKUP is as follows;

=LOOKUP(lookup_value , lookup_vector , result_vector)

- The **lookup_value** represents the number or text entry to look for;
- the **lookup_vector** is the area in which to search for the lookup_value;
- the **result_vector** is the adjacent row or column where the corresponding value or text is to be found.

B8		fx =LOOKUP(B7,C2:C10,D2:D10)			
A	B	C	D	E	
		Salary	Car		
		1000	Nissan Sunny		
		1100	VW Golf		
		1200	Citroën C3		
		1250	BMW 101		
		1350	Audi A4		
Salary:	1350.0	1400	Citroën Picasso		
Car	Audi A4	1550	BMW 320		
		1600	Mercedes 190		
		2000	Audi A6		

It is essential that data in the lookup vector is placed in ascending order, i.e. numbers from lowest to highest, text from A to Z. If this is not done, the LOOKUP function may return the wrong result.

In the diagram, column C contains varying salaries, against which there is a company car in column D which corresponds to each salary.

For example, a 1100 salary gets a Golf, a 2000 salary gets a Audi A6. A LOOKUP formula can be used to return whatever car is appropriate to a salary figure that is entered. In this case, the lookup_value is the cell where the salary is entered (B7), the lookup_vector is the salary column (C2:C10), and the result_vector is the car column (D2:D10). Hence the formula;

=LOOKUP(B7, C2:C10, D2:D10)

Typing 1350 in cell B13 will set the lookup_value. LOOKUP will search through the lookup_vector to find the matching salary, and return the appropriate car from the result_vector, which in this case is Audi A4.

Alternatively, the formula could be simplified and cell references avoided by using Formula, Define Name to give appropriate range names. Call B7 Salary, C2:C10 Salaries and D2:D10 Cars. The LOOKUP formula could then be simplified to;

=LOOKUP(Salary,Salaries,Cars)

One of the advantages of the LOOKUP function is that if the exact lookup_value is not found, it will approximate to the nearest figure below the requested value. For instance, if a user enters a Salary of 1375, there is no figure in the Salaries range which matches this. However, the nearest salary below 1375 is 1350, so the corresponding car is returned, which is a Audi a4. This technique is very useful when the lookup_vector indicates grades or "bands". In this case, anyone in the salary "band" between 1600 and 2000 gets a Mercedes 190. Only when their salary meets or exceeds 2000 do they get an Audi A6.

To insert a lookup function:

- 2003: menu: Function → Lookup & Reference → Lookup →

- 2007: FORMULAS Ribbon →FUNCTION LIBRARY →LOOKUP AND REFERENCE→

→ dialog box with two versions of LOOKUP. the first is the "array" and the second the "vector".

The second of these forms, the "vector" LOOKUP is by far the most useful, and it is recommended that you only use this form.

2.8.3 Hlookup

The horizontal LOOKUP function (HLOOKUP) can be used not just on a "vector" (single column or row of data), but on an "array" (multiple rows and columns). HLOOKUP searches for a specified value horizontally along the top row of an array. When the value is found, HLOOKUP searches down to a specified row and enters the value of the cell. This is useful when data is arranged in a large tabular format, and it would be difficult for you to read across columns and then down to the appropriate cell. HLOOKUP will do this automatically.

The syntax for HLOOKUP is;

=HLOOKUP(lookup_value , table_array , row_index_number)

The lookup_value is, as before, a number, text string or cell reference which is the value to be found along the top row of the data; the table_array is the cell references (or range name) of the entire table of data; the row_index_number represents the row from which the result is required. This must be a number, e.g. 4 instructs HLOOKUP to extract a value from row 4 of the table_array.

It is important to remember that data in the array must be in ascending order. With a simple LOOKUP function, only one column or row of data, referred to as a vector, is required. HLOOKUP uses an array (i.e. more than one column or row of data). Therefore, as HLOOKUP searches horizontally (i.e. across the array), data in the first row must be in ascending order, i.e. numbers from lowest to highest, text from A to Z. As with LOOKUP, if this rule is ignored, HLOOKUP will return the wrong value.

As an example, a user may have a spreadsheet which displays various different rates of interest for a range of amounts over different time periods;

B19		=HLOOKUP(B17,E13:J18,B18)									
A	B	C	D	E	F	G	H	I	J		
				perio	1	2	3	4	5		
			Capital		20000	25000	30000	40000	60000		
			2	10	12.5	14.1	14.9	15.5	16.3		
			3	15	14	14.3	15	16	17.5		
			4	20	16.5	17	17	17.6	19		
Amount	28000		5	25	17.2	17.5	18	18.5	19.3		
Row	3		6	30	17.9	18	18.6	19	20.2		
Rate	14.3	Hlookup									
				Amount	28000						
Period	15			Period	20						
Column	4			Rate	17	Nested Hlookup					
Rate	15	Vlookup									

range of amounts over different time periods;

Whatever the amount a customer wants to borrow, he may pay up to five

different rates of interest depending on whether the loan is over 10, 15 or more years. The HLOOKUP function will find a specific amount, then move down the array to find the appropriate interest rate for the required time period.

Designate cell B17 as the cell to hold the amount, i.e. the lookup_value; cells E13:J18 are the table_array; the row_index_number will be 3 if a customer wants the loan over 20 years, 4 if he wants the loan over 25 years, and so on. Cell B51 holds this formula;
=HLOOKUP(B17,E13:J18,B18)

As with the LOOKUP function, the advantage of HLOOKUP is that it does not necessarily have to find the exact lookup_value. If, for example, you wanted to find out what interest rate is applicable to a 28000 loan, the figure 28000 can be entered in the lookup_value cell (A51) and the rate 14.30% appears. As before, Excel has looked for the value in the array closest to, but lower than, the lookup_value.

2.8.4 Vlookup

The VLOOKUP function works on the same principle as HLOOKUP, but instead of searching horizontally, VLOOKUP searches vertically. VLOOKUP searches for a specified value vertically down the first column of an array. When the value is found, VLOOKUP searches across to a specified column and enters the value of the cell. The syntax for the VLOOKUP function follows the same pattern as HLOOKUP, except that instead of specifying a row index number, you would specify a column index number to instruct VLOOKUP to move across to a specific column in the array where the required value is to be found.

=VLOOKUP(lookup_value, table_array, col_index_number)

In the case of VLOOKUP, data in the first column of the array should be in ascending order, as VLOOKUP searches down this column for the lookup_value.

2.8.5 Nested Lookups

One of the limitations of the horizontal and vertical LOOKUP functions is that for every lookup_value changed, the column or row index number stays constant. Using our example, the HLOOKUP will search for any amount, but always for the same time period. Conversely, the VLOOKUP will search for any time period, but always for the same amount. In both cases, if you want to alter the time period and the amount the formula must be edited to alter the column or row index number.

There is, however, a technique whereby one LOOKUP function is "nested" within another. This looks up one value, which will then be used in a second LOOKUP formula as a column or row index number. Using this technique allows you to, say, enter a time period and an amount and see the correct interest rate.

Because nested LOOKUPS have more than one lookup_value, more than one array is needed. This second array should consist of the column or row numbers to be used in the LOOKUP formula. The example shows our main interest rates spreadsheet, with an additional two columns of data;

LOOKUP(G21,E14:E18,D14:D18)

Notice there is no equals sign, because this formula is not being entered in a cell of its own. The formula will return a value between 2 and 6 which will be used as a row_index_number in a HLOOKUP formula. This HLOOKUP will look in the main interest rate table for an amount typed in by you, and will respond to the row_index_number returned from the nested LOOKUP formula. The cells G20 and G21 hold the amount and time period to be typed in by you, and the entire nested HLOOKUP, typed in G22, is as follows;

=HLOOKUP (G20,E13:J18,LOOKUP(G21,E14:E18,D14:D18))

2.9 OTHER USEFUL FUNCTIONS

2.9.1 ISERROR

Iserror(Value)

Value refers to any error value (#N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!)

Useful in combination with If function

=IF(ISERROR(AVERAGE(F14:J14)), "", AVERAGE(F14:J14))

3 GOAL SEEK AND SOLVER

3.1 GOAL SEEKING AND SOLVING

Excel has a number of ways of altering conditions on the spreadsheet and making formulae produce whatever result is requested. Excel can also forecast what conditions on the spreadsheet would be needed to optimise the result of a formula. For instance, there may be a profits figure that needs to be kept as high as possible, a costs figure that needs to be kept to a minimum, or a budget constraint that has to equal a certain figure exactly. Usually, these figures are formulae that depend on a great many other variables on the spreadsheet. Therefore, you would have to do an awful lot of trial-and-error analysis to obtain the desired result. Excel can, however, perform this analysis very quickly to obtain optimum results. The Goal Seek command can be used to make a formula achieve a certain value by altering just one variable. The Solver can be used for more painstaking analysis where many variables could be adjusted to reach a desired result. The Solver can be used to not only obtain a specific value, but to maximise or minimise the result of a formula (e.g. maximise profits or minimise costs).

3.1.1 Goal Seek.

The Goal Seek command is used to bring one formula to a specific value. It does this by changing one of the cells that is referenced by the formula. Goal Seek asks for a cell reference that contains a formula (the Set cell). It also asks for a value, which is the figure you want the cell to equal. Finally, Goal Seek asks for a cell to alter in order to take the Set cell to the required value.

In the example spreadsheet, cell B8 contains a formula that sums advertising and payroll costs. Cell B10 contains a profits formula based on the revenue figure, minus the total costs.

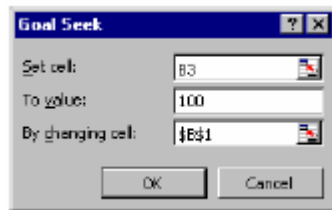
	E10	-B3-B8		
	A	B	C	D
2				
3	Revenue	5000		
4				
5	Advert costs	300		
6	Payroll	1150		
7				
8	Total costs	1450		
9				
10	Profits	3550		
11				

A user may want to see how a profit of 4000 can be achieved by altering payroll costs.

To launch the Goal seeker:

Click on the cell whose value you wish to set. (The Set cell must contain a formula).

- 2003: Menu Tools → Goal Seek
- 2007: Data Ribbon → Data Tools → What - If Analysis → Goal Seek.

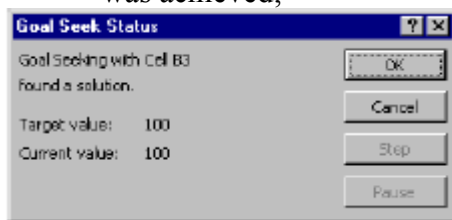


The Goal Seek command automatically suggests the active cell as the Set cell. This can be overtyped with a new cell reference or you may click on the appropriate cell on the spreadsheet.

Now enter the desired value this formula should reach. Click inside the TO VALUE box and type in the value you want your selected formula to equal.

Finally, click inside the BY CHANGING CELL box and either type or click on the cell whose value can be changed to achieve the desired result.

- Click the OK button and the spreadsheet will alter the cell to a value sufficient for the formula to reach your goal. Goal Seek also informs you that the goal was achieved;



- You now have the choice of accepting the revised spreadsheet, or returning to the previous values. Click OK to keep the changes, or Cancel to restore previous values.

Goal Seek can be used repeatedly in this way to see how revenue or other costs could be used to influence the final profits. Simply repeat the above process and alter the changing cell reference.

The changing cell must contain a value, not a formula. For example, if you tried to alter profits by changing total costs, this cell contains a formula and Goal Seek will not accept it as a changing cell. Only the advertising costs or the payroll cells can be used as changing cells.

Goal Seek will only accept one cell reference as the changing cell, but names are acceptable. For instance, if a user had named either cells B5 or B6 as "Advert_costs" or "Payroll" respectively, these names could be typed in the BY CHANGING CELL box. For goal-seeking with more than one changing cell, use the Solver

3.1.2 Goal-Seeking with Charts

3.1.3 Solver

For more complex trial-and-error analysis the Excel Solver should be used. Unlike Goal Seek, the Solver can alter a formula not just to produce a set value, but to

maximise or minimise the result. More than one changing cell can be specified, so as to increase the number of possibilities, and constraints can be built in to restrict the analysis to operate only under specific conditions.

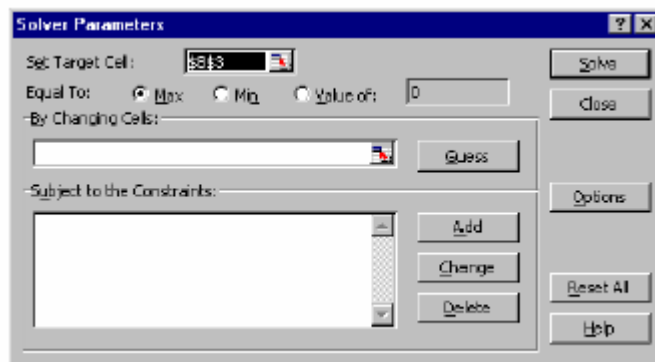
The basis for using the Solver is usually to alter many figures to produce the optimum result for a single formula. This could mean, for example, altering price figures to maximise profits. It could mean adjusting expenditure to minimise costs, etc. Whatever the case, the variable figures to be adjusted must have an influence, either directly or indirectly, on the overall result, that is to say the changing cells must affect the formula to be optimised. Up to 200 changing cells can be included in the solving process, and up to 100 constraints can be built in to limit the Solver's results.

3.1.4 Solver Parameters

The Solver needs quite a lot of information in order for it to be able to come up with a realistic solution. These are the Solver parameters.

To set up the Solver:

- 2003: Menu: Tools→Solver
- 2007: DATA ribbon→Analysis→ Solver



Like Goal Seek, the Set Cell is the cell containing the formula whose value is to be optimised. Unlike Goal Seek, however, the formula can be maximised or minimised as well as set to a specific value.

- Decide which cells the Solver should alter in order to produce the Set Cell result. You can either type or click on the appropriate cells, and [Ctrl] click if non-adjacent cell references are required.

When using a complex spreadsheet, or one that was created by someone else, there is an option to let the Solver guess the changing cells. Usually it will select the cells containing values that have an immediate effect on the Set Cell, so it may be a good idea to amend this.

3.1.5 Build constraints into your Solver parameters:

Constraints prevent the Solver from coming up with unrealistic solutions.

- In the Solver dialog, choose Add
- This dialog box asks you to choose a cell whose value will be kept within certain limits. It can be any cell or cells on the spreadsheet (simply type the reference or select the range).
- This cell can be subjected to an upper or lower limit, made to equal a specific value or forced to be a whole number. Drop down the arrow in the centre of

the Constraint box to see the list of choices:- To set an upper limit, click on the <= symbol; for a lower limit, >=; the = sign for a specific value and the int option for an integer (whole number).

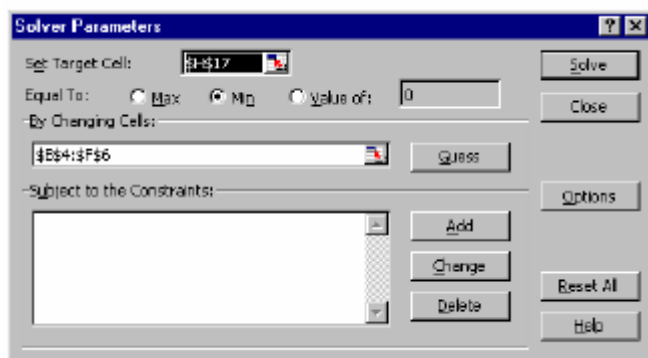
→ Once the OK button is chosen, the Solver Parameter dialog box re displays and the constraint appears in the window at the bottom. This constraint can be amended using the Change button, or removed using the Delete button.

When maximising or minimising a formula value, it is important to include constraints which set upper or lower limits on the changing values. For instance, when maximising profits by changing sales figures, the Solver could conceivably increase sales to infinity. If the sales figures are not limited by an upper constraint, the Solver will return an error message stating that the cell values do not converge. Similarly, minimising total costs could be achieved by making one of the contributing costs infinitely less than zero. A constraint should be included, therefore, to set a minimum level on these values.

The example here shows the number of cases for five London hospitals, split into three types: Elective, Emergency and Day cases. Below this are the respective costs of each type of case for each hospital, and finally the total costs (number multiplied by price) for each type in each hospital. All these figures are totalled in column H, to arrive at a final total costs figure in cell H17.

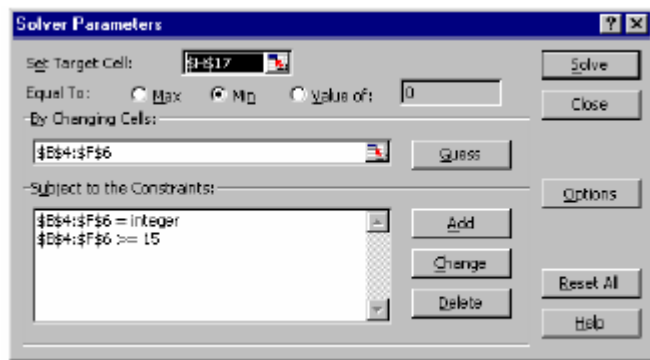
	A	B	C	D	E	F	G	H	I
2									
3		BARTS	CHARX	HAMSMTH	MOORFLDS	ST GEORGES			
4	Elective	4	5	5	4	7		25	
5	Emerg	3	2	4	5	3		17	
6	Day	9	8	7	10	6		40	
7		16	15	16	19	16		82	Total Cases
8									
9	Elective	11	9	13	2.4	6		41.4	
10	Emerg	8	8.2	10	5	4		35.2	
11	Day	3	6	14	3	2		28	
12		22	23.2	37	10.4	12		104.6	Total Price
13									
14	Elective	44	45	65	9.6	42		205.6	
15	Emerg	24	16.4	40	25	12		117.4	
16	Day	27	48	98	30	12		215	
17		95	109.4	203	64.6	66		538	Total Cost
18									

Call up the Tools, Solver command from the menu. The Set Cell in this case will be H17, the total costs cell. It can be assumed that the costs of cases cannot be changed, only the number in each hospital, therefore the changing cells will be B4:F6:



The problem is that, if Solve is chosen now, the number of cases could reduce to infinitely below zero and produce an error message. Fortunately, constraints can be built in to force each hospital to have a minimum number of cases, and for there to be a total number of cases overall. Choose the Add option to add a constraint, highlight

the cells B4:F6, drop down the arrow and click on \geq to set a minimum limit. Here, type whatever the minimum number of cases should be. To avoid the error message, simply type 1 and choose OK. Also, as hospital cases cannot be fractions, add another constraint to force these cells to be integers. Finally, a constraint could be added to set a total number of cases (cell H7). The Solver parameters should now appear as follows:



When Solve is chosen, the Solver carries out its analysis and finds a solution. This may be unsatisfactory, as it has calculated that the best way to minimise costs is to put the majority of cases in St George's as day patients. Further constraints could now be added to force the Solver to place minimum numbers of cases in the other hospitals, or set a maximum limit on St Georges' day cases.

3.1.6 Solver and Scenario Manager

The Solver uses sophisticated numeric analysis and iterative methods to perform trial-and-error calculations very quickly. The original values of the spreadsheet, therefore, have a profound effect on the result of a Solver solution. It may be the case that there is no concrete solution to a spreadsheet problem, and the Solver may produce a "best guess" within the specified constraints. Changing the original values and running the Solver again may produce a different result, and it may therefore be helpful to save the different "scenarios" that are produced. It may also be necessary to save scenarios where constraints have either been added, removed or amended. Using Excel's Scenario Manager can facilitate this, by allowing you to save each new Solver solution and compare it to previous ones. For most complex spreadsheet problems, the Solver and Scenario Manager are used together.

3.1.7 Opening Scenario Manager

Once Solver has found a solution, choose Save Scenario from the dialog box. The scenario can be named and either the new values kept or the original values restored.

Scenarios can be viewed, modified, reports can be provided.

4 USING EXCEL TO MANAGE LISTS

4.1 EXCEL LISTS, LIST TERMINOLOGY

4.1.1 Row And Column Content

Although Excel's primary function is as a Spreadsheet, it can also be used for a number of list operations. It is possible to store, and manipulate information

(customer records, staff records, or stock inventories for example) on an Excel worksheet, organise it in different ways, and "query" the list to extract information which meets specific, user-defined criteria. The list is effectively treated as a database.

In order to use Excel's database capacity, information must be laid out in rows and columns subject to certain constraints. There are some database terms with which the user should become familiar:

The information being stored must be divided up into categories. For example, information on staff might include Firstname, Lastname, and Department. In an Excel List, each category must be entered in a separate column. Do not mix text and numbers in a column – the data must belong to the same category of information and therefore should be the same data type. Do not use spaces in front of column entries, use alignment buttons instead if you need to move data away from the column edges.

4.1.2 Column Labels

This is the title at the top of each column, describing the category of information which it contains. Each label name must be unique and must be made up of text rather than numbers or calculations. The column names must appear directly above the list information - they may not be separated from the rest of the list by a blank row. Format your column labels to distinguish them from the list data.

4.1.3 List Size And Location

The List is the whole collection of information, all Fields, Fieldnames and Records and should be laid out as a regular block of data.

- A list can be up to 256 columns wide by 65536 rows (an entire worksheet).
- Do not place more than one list on a worksheet. If you want more than one list in a workbook, place each list on a separate sheet.
- Leave at least one blank row and one blank column between the list and other data on the worksheet.
- Place additional data diagonally below and to the right of your list. This ensures that data will not be affected when you filter the list.


4.1.4 Miscellaneous

- Excel does not distinguish between upper and lower case characters in a list, unless you use the Case-sensitive sort option.
- When you use formulae in lists, Excel uses the results of the formulae.

4.2 SORTING DATA .

Although not confined to database information, the sorting facility in Excel is particularly appropriate for changing the order in which records are listed. Remember to save the file containing the database information prior to sorting.

If you will need to restore the original record order, it is a good idea to include a column of record numbers before sorting the database. This can be achieved simply by adding a column with a suitable heading, and using the fill handle or the data series command to enter consecutive numbers adjacent to each record

- single-level sorting click in a cell within the column 
- multi-level sorting: go through the menu for data sorting

4.2.1 Custom Sorting Options

The ascending and descending sort orders rearrange your list by alphabetical, numerical, reverse alphabetical or reverse numerical order. For some types of data, such as months, this may not be the order that you need to use. You can use one of the custom sort orders provided with the Excel program to rearrange your data in chronological order by day of the week or by month.

Menu: Data → Sort → options allow for changes in the sorting process

4.2.2 Creating A Custom Sort Order

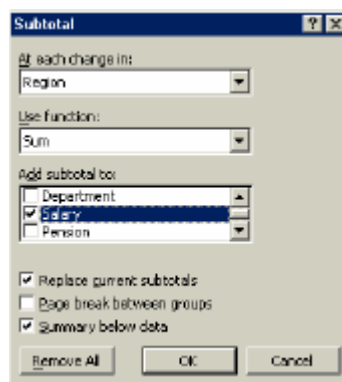
Custom sort orders can be created in

- 2003: menu Tools → options → custom lists
- 2007: home ribbon → editing group → custom sort

4.3 ADDING SUBTOTALS TO A LIST

Automatic subtotals are useful in summarising the data contained in a list. Subtotals are created by using an Excel summary function such as SUM(), COUNT() or AVERAGE(). To use the subtotals, your data must be organised in a properly designed list and sorted according to the column by which you want to summarise the data.

- Sort the list according to the column by which you want to summarise the data.
- 2003 Menu: Data → Subtotals
- 2007 Data ribbon → outline group → Subtotals



- Choose Data from the menu bar, then Subtotals.
- From the At Each Change In drop-down list, select the field by which you want to summarise the data.
- From the Use Function drop-down list, select the summary function you want to use to generate the subtotals.
- In the Add Subtotal To box, check the column to which you want the function to be applied.
- If desired, check the options for replacing the current subtotals (if any), inserting a page break for each summary group, and inserting the summary below each group.
- Click OK.

When you use the Data, Subtotals command, it adds its own Grand Total, so you should not use the SUM() function in your list. If you use the Data, Subtotals

command, the SUM() function will be inaccurate since it includes the subtotals in the calculation. (see working with lists).

5 STRUCTURING OF WORKSHEETS

Like all other computer files, a workbook should be structured. But moreover, like in data-bases, a basic data (information) should only figure one time in a workbook. In a worksheet, essentially three types of data structures can be distinguished:

- the basic data
- the calculations
- the presentations.

Ideally, each type should be presented on different and inter-linked spreadsheet.

5.1 THE BASIC DATA:

It contains all the data that can change regularly. Each information should have only one data field, the other times it appears are referring to that field.

5.2 THE CALCULATIONS

On the calculation worksheets (or process worksheets, contain all the possible calculations, here also it is better to proceed on several worksheets than only one).

5.3 THE PRESENTATIONS

The presentations are the data from the other worksheets presented in the different ways needed for the corresponding reports. If there is only one presentation, it is clear that the calculation and presentation worksheets are the same.

6 PASTING EXCEL DOCUMENTS IN WORD DOCUMENTS

Pasting Excel documents into Word documents can be done on several ways of which:

- Transforming it into a Word object (table, text etc) (paste)
- Transforming it into a picture (paste special)
- Keeping it as an Excel object (paste special)

The two last ones give normally the best effect because they remain untouched.

In both cases it can be pasted with link, this means that the word document refers to the original Excel documents and get an updated figure when you ask to update.