

Excel Yourself 2014

By Neale Blackwood

www.a4accounting.com.au

a4@iinet.net.au



Contents

Introduction	3
March 2014 - Unhide a Single Column	4
Manual Method	4
Macro Solution.....	5
April 14 - Using Rank to Extract Entries	6
Syntaxes	8
June 14 - Creating Unique Codes	9
July 14 - Bullet Chart	11
September 14 - Handling Badly Laid Out Data	15

Introduction

I'll review some of my Excel Yourself articles that were first published in the INTHEBLACK magazine during 2014. Due to the one hour time restriction I will only be covering 5 out of the 11 articles.

I have chosen those articles that will benefit most from the webinar format and included some extra content in some of the articles.

The workings sheets for ten of the eleven articles have been included in the companion file for this session. The October article example has not been included as it related to a linked file and I don't include linked examples in webinars.

The companion file includes the code for the macro articles as well. The macro code can be viewed by pressing Alt + F11 and clicking on the module (left-hand side of VBA window) for the month involved.

The companion Excel sheets are arranged in date order starting with the February article on the left and working to the right to end with the December article. The green sheet tabs are the ones to be covered during the session.

The articles were all first published in the CPA Australia INTHEBLACK magazine during 2014.

The content of this manual has been re-written and adapted from the original articles.

March 2014 - Unhide a Single Column

The inspiration for this article came from one of my own files where I track cash flow. I have my cash flow categories listed in column A on the left and the weeks as columns, going across the sheet. I hide the completed weeks so that I can go back to them if necessary.

The problem arises when I want to check what was in the previous, hidden week. It means I have to unhide hundreds of columns just to see the last hidden column. I then have to re-hide all those previous columns.

In the example below I only want to unhide column J.

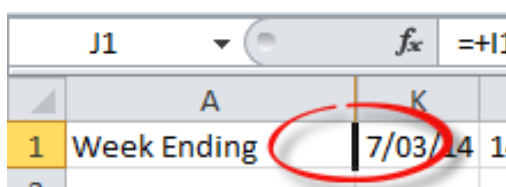
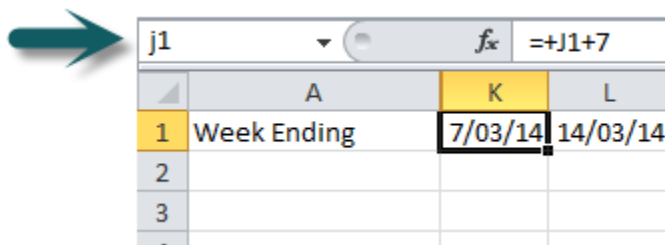
	A	K	L	M	N	O	P
1	Week Ending	7/03/14	14/03/14	21/03/14	28/03/14	4/04/14	11/04/14
2							

All I want to do is unhide the column to the left of the current week. That's when I thought that a macro would be an easy solution.

There is a way to unhide just the previous column, but it does take a few keystrokes.

Manual Method

1. Click the Name Box (above the column letters and to the left of the Formula Bar) and type the reference of a cell in the column you want to unhide, in our case J1.



When you type the cell reference in the Name Box and press Enter, Excel will select that cell, even if it isn't visible.

Excel shows a thick vertical line then a cell in a hidden column is selected.

2. If you have Windows XP you can use a keyboard shortcut to unhide the column.

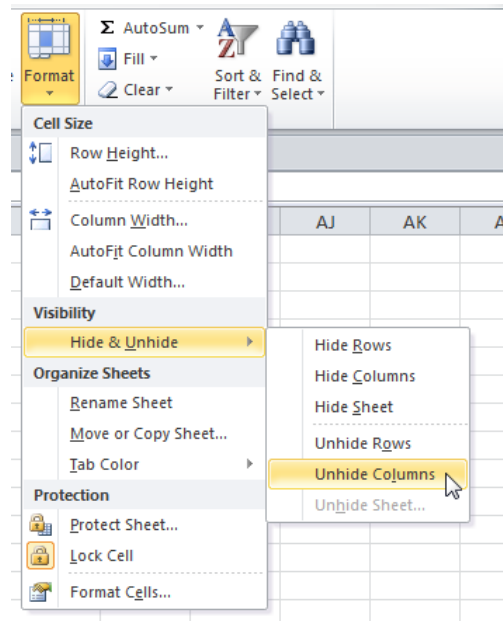
Ctrl + Shift +)

Unfortunately this shortcut doesn't work if you have Windows 7. (Hence, I created a macro).

- On the Home ribbon click the Format icon (far right), then click Hide & Unhide and then click Unhide Columns.

Done, the column is unhidden.

That took seven keyboard or mouse actions to perform. The macro will save quite a few keystrokes and time.



Macro Solution

The macro that will unhide the column to the left is only one line of code. To see the macro press Alt + F11 and double click the Mar_14 module on the left of screen.

```
Sub Unhide_Column_Left ()

    ActiveCell.Offset (0, -1).EntireColumn.Hidden = False

End Sub
```

This takes advantage of two useful VBA statements.

ActiveCell means whatever cell is active in the current workbook when the macro is run. This means you select a cell to work with before running the macro. This creates a flexible macro that the user can manipulate just by selecting different cells.

The Offset command is similar to the OFFSET function. It allows you to select another cell based on a starting cell and then moving a number of rows, columns or both from the starting cell to select another cell to work with.

In our example the numbers in the brackets refer to rows (0) and columns (-1). This means the starting cell (the active cell in this case) is the base; from there we select another cell based on the values in the brackets. 0 means we move zero rows from the base (active cell). The -1 means we move one column to the left. (Negative numbers move the rows up and the columns to the left. Positive numbers move the rows down and the columns to the right)

Setting the property .EntireColumn.Hidden to False makes the selected column visible.

The other example macros from the article are shown in the Mar_14 module demonstrating selecting surrounding cells.

April 14 - Using Rank to Extract Entries

If you need to extract and list the top 5 clients in terms of revenue you can't use the RANK function as it has issues if there are ties.

See the image on the right.

Rows 4 and 7 have a tie for second place in column C, which uses the RANK function. The RANK function duplicates ranks for ties, so that C4 and C7 both display 2 and there is no 3 in the column C.

	A	B	C	D
1	Client	Sales	Rank Equal	Rank 1 to 10
2	Client A	18,849	8	8
3	Client B	33,557	4	4
4	Client C	45,891	2	3
5	Client D	18,613	9	9
6	Client E	29,505	6	6
7	Client F	45,891	2	2
8	Client G	15,268	10	10
9	Client H	48,819	1	1
10	Client I	26,195	7	7
11	Client J	32,474	5	5

For our needs we need to have one value shown as second ranked and the other valued ranked third. It doesn't really matter which one is given which placing.

That enables us to then extract the top 5 entries using an INDEX and MATCH combination.

Note: the syntaxes for all the functions used are included at the end of this section.

We can adjust the RANK function result with a COUNTIF function to achieve ranked values with no duplicates. The formula in column D provides us with a sequential ranking. The formula in D2, which has been copied down is

=RANK(B2,\$B\$2:\$B\$11)+COUNTIF(B2:\$B\$11,B2)-1

The first part of the formula is the same as the RANK function used in column C which can result in ties.

We use the COUNTIF function to adjust the RANK result. The COUNTIF function counts how many times an entry appears in a range. In this case we have used a relative reference for the start of the range B2. The end of the range is a fixed reference \$B\$11. This means as the formula is copied down the range starts at the same row as it is used on and gets smaller as it is copied down.

Hence, at the top of the range it is using the whole range and it counts how many times the value in column B appears. In cell D4 it appears twice. Now the value must appear at least once in the range, so we deduct 1 from the result to find what number to adjust to the RANK result by.

We can test this works by making another cell equal 45,891, say H3. The relative reference at the start of the range is what makes this technique work.

If the value is not duplicated the COUNTIF returns 1 and we deduct 1, so the RANK result is not adjusted.

If a value is duplicated 3 times the first time it is encountered going down the column the COUNTIF will return 3 as the whole range is used and we

deduct 1 and add that to the RANK result. So the RANK result is increased by 2 (3 - 1).

	E	F	G	H
1		List Top 5	Client	Sales
2		1	Client H	48,819
3		2	Client F	45,891
4		3	Client C	45,891
5		4	Client B	33,557
6		5	Client J	32,474

The next time the duplicated value is encountered going down the column the COUNTIF will only return 2 (the range size has reduced due to the relative starting cell) from which we deduct 1, so 1 is added to the RANK result. The last time the duplicated value is encountered it is only there once so the RANK result is not adjusted.

With sequential values Column D can be used with an INDEX - MATCH combination to extract the top five values.

Note: in the current layout you can't use a VLOOKUP function to extract the results.

The formula for G2 is

=INDEX(A:A,MATCH(\$F2,\$D:\$D,0))

The INDEX function initially defines a range, column A in this case.

You then specify what row number to extract from the range.

The MATCH function will provide the row number to use in the INDEX.

The MATCH function returns a number representing the position of the value being searched for within the range. In this case it looks up the value from cell F2 in column D. It will return the row number of the cell that holds the value of 1.

The INDEX function uses the MATCH result to extract the corresponding row from column A.

The formula has been copied across to cell H2 and copied down. In this case it is extracting the value from column B.

=INDEX(B:B,MATCH(\$F2,\$D:\$D,0))

There is a third option in the INDEX function that allows you to extract from a specific column, but in our example we used a single column so the column reference, which is optional, is not required.

Syntaxes

The syntaxes for the four functions used are shown below.

RANK(Value,Range)

Value - the value to be ranked. Usually a relative reference (no \$ symbols), in our example it is cell B2.

Range - the range that holds all of the values to be ranked. This range tends to be a fixed reference. In our example \$B\$2:\$B\$11 is a fixed range. This enables the formula to be copied to the other cells that need to be ranked.

COUNTIF(Range, Value)

Range - the range containing all of the values. In our example B2:\$B\$11 - see discussion above about the type of range used.

Value - the value to count within the range. Usually a cell reference, B2 in our example.

INDEX(Range, Row_number, Column_number)

Range - can be a range; a single row or a single column. Our first example uses the single column A:A.

Row_number - the row number within the range of the cell to extract. In our case this is supplied by the MATCH function. Using a MATCH function adds flexibility to the INDEX function.

Column_number - (optional) the column number within the range of the cell to extract. In our example this is not required.

MATCH(Value,Range,Search_Type)

Value - the value to find within the range. Cell \$F2, holds the value 1. The \$ sign fixes the column reference so the formula can be copied across.

Range - the range containing all the values. On our case we are searching column \$D:\$D where we have created the sequential rank numbers. The reference is fixed so the formula can be copied across.

Search_Type - in our case we have used 0 which means an exact match is required. An approximate match is possible if the range is sorted. In our case the range isn't sorted, so we have to use an exact match.

June 14 - Creating Unique Codes

This has similarities to the April article in that we want to create unique codes. The solution also involves the COUNTIF function.

In creating unique codes it is useful to understand something about how Excel sorts alpha-numeric codes.

In the image on the right, column A entries have been sorted in column C.

How they appear is not usually the outcome you are expecting or requiring.

Excel looks at the numbers after the characters and it lists all the numbers starting with 1 first. Then all the numbers starting with 2 are shown next.

To solve this you need to identify how many numeric digits you will use and include leading zeroes in the numbers.

	A	B	C
1	Code		Code
2	ABC1		ABC1
3	ABC2		ABC10
4	ABC3		ABC11
5	ABC4		ABC12
6	ABC5		ABC13
7	ABC6		ABC14
8	ABC7		ABC2
9	ABC8		ABC3
10	ABC9		ABC4
11	ABC10		ABC5
12	ABC11		ABC6
13	ABC12		ABC7
14	ABC13		ABC8
15	ABC14		ABC9



	A
1	Code
2	ABC
3	ABC
4	ABC
5	DEF
6	DEF
7	DEF
8	DEF
9	GHI
10	GHI

In our case (on the left) we have a column of codes that contain duplicates.

We need to make these codes unique so that we can refer to specific rows within the list.

We also want to maintain the current order of the listing.

To achieve our result we are going to use the TEXT function and the COUNTIF function, plus a range reference technique similar to the one used in the April article.

The image on the right has the result we require.

The formula for the cell B2 is

`=A2&TEXT(COUNTIF(A2:A2,A2),"000")`

The & symbol joins text together. We will start with the code from column A and add a sequential number with 3 digits.

The TEXT function formats numbers and dates using the format specified within the quotation marks.

The COUNTIF function is counting how many times the code appears in column A but note the range reference used.

`COUNTIF(A2:A2,A2)`

	A	B
1	Code	Unique Code
2	ABC	ABC001
3	ABC	ABC002
4	ABC	ABC003
5	DEF	DEF001
6	DEF	DEF002
7	DEF	DEF003
8	DEF	DEF004
9	GHI	GHI001
10	GHI	GHI002

By using the range \$A\$2:A2, the start cell is fixed at \$A\$2 but the end cell is relative, so the range expands as it is copied down. This means the number of cells that are being counted increases. As the formula is copied down the column, it provides a sequential number based on the number of times the value appears in column A. That number is then formatted by the TEXT function (see below) into 001, 002 and 003 etc.

Syntax

TEXT(Value, Format)

Value - usually a number or a date to be formatted as text.

Format - the format to apply to the number or date. It must be enclosed in quotation marks. The formats are based on those on the Custom option in the Number tab of the Format Cells dialog.

Using "000" as the format forces leading zeroes on the sequential numbers provided by the COUNTIF function eg 001, 002, 003 etc. The leading zeroes mean you can sort the alpha numeric number correctly.

If you need to capture the values in the column B so that they don't change, you can use Paste Special Values - there are a number of ways to do that which I will demonstrate on screen.

July 14 - Bullet Chart

A bullet chart is a chart developed by Stephen Few. He is an author who specialises in data visualisation topics and I recommend his books to you if you are interested in developing better charts.

The chart is meant to replace gauge type charts which can be developed in Excel but they are complex and they take up a lot of space considering the limited amount of data they contain.

Unfortunately when I wrote the article I did break one of his rules for data visualisation and it was my fault entirely.

So in this demonstration I will keep to the specification that he developed. The rule I broke was using traffic light colours in the chart. His specification involves using shades of grey rather than red or green for the areas on the chart.

Colour Blindness

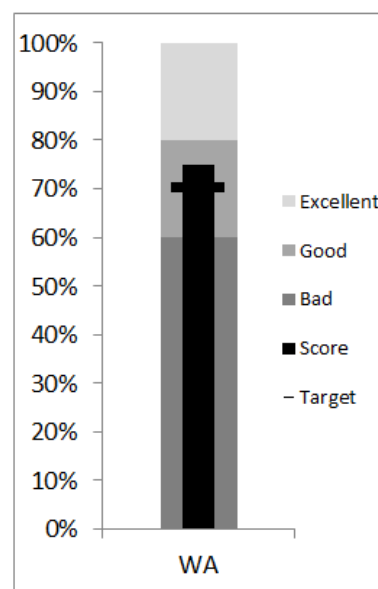
It is worth noting that about 1 in 12 men are colour blind to some extent and 1 in 100 women.

The technique I will use is based on one contained in the book Excel 2007 Dashboards and Reports for Dummies by Michael Alexander published by Wiley.

On the right is the chart we want to create.

The data is shown below.

	A	B
1	Feedback	WA
2	Score	75%
3	Target	70%
4	Bad	60%
5	Good	20%
6	Excellent	20%



The layout of the data enables us to create the bullet chart more easily, so let me explain it.

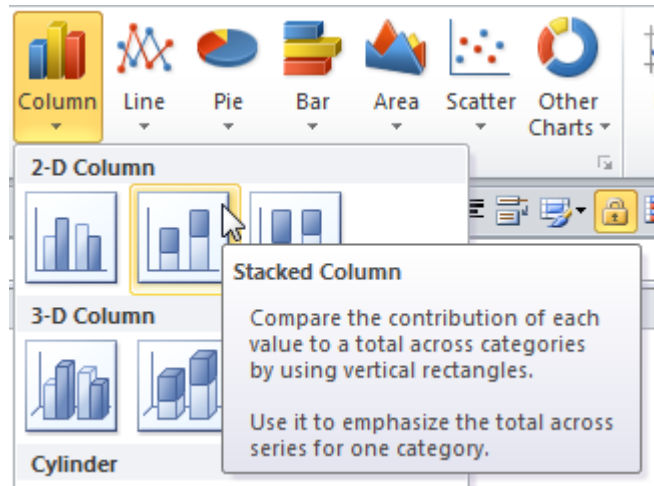
We have a score of 75% in cell B2. We want to plot the score against a target, 70% in cell B3. The three ratings are in the range B4:B6. The three ratings need to be treated as cumulative. Hence between 0% and 60% we consider the result to be Bad. Between 60% and 80% we consider this to be Good. A score between 80% and 100% is considered Excellent. The three rating values shown add up to 100%.

This layout works well with the following technique to create the bullet chart.

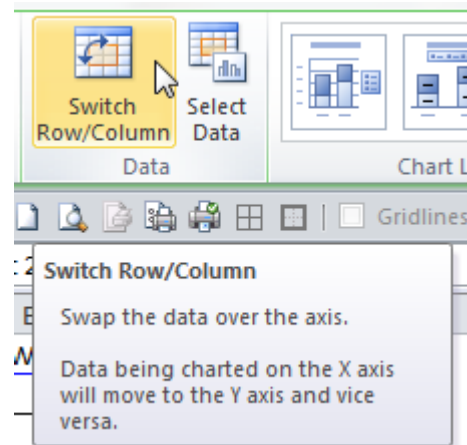
Here are a couple of chart tips before starting the instructions. To delete anything on a chart, select it and press the Delete key on your keyboard. To modify any part of a chart, right click it and select the Format option at the bottom of the menu.

To create a bullet chart follow these steps in sequence. Completely read each point before following the instructions for that point. (Note: All Excel format options use the U.S. spelling of color)

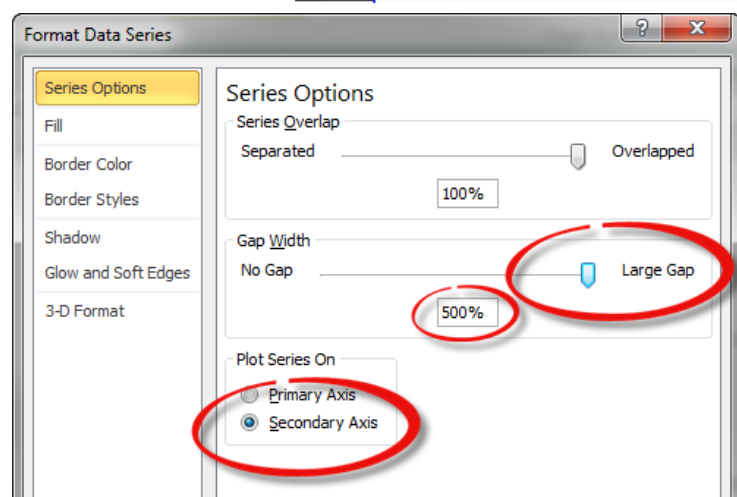
1. Select the range A1 to B6. Click the Insert ribbon tab, click the drop down for *Column* and select the *Stacked Column* option (second icon, top line), see image on the right.



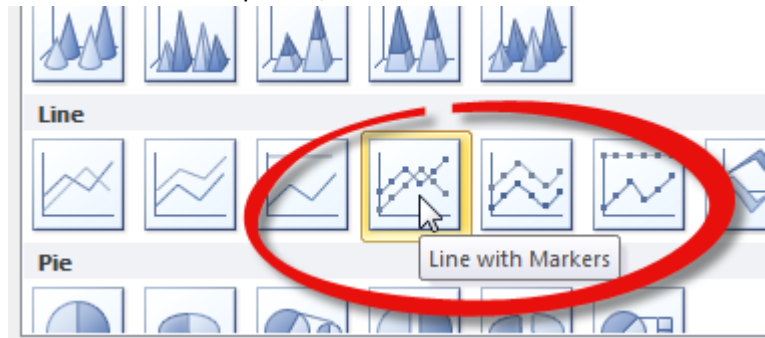
2. With the chart selected click the Design ribbon tab, click on the *Switch the Row/Column* icon, see image on right. This will create a stacked column chart of all the percentages.



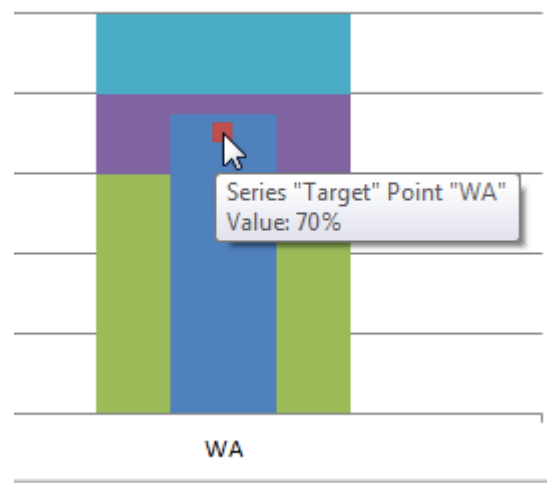
3. Right click the bottom segment in the column stack, which should be the score data series. Select *Format Data Series* and then in the *Series Options* select the *Secondary Axis* option and change the *Gap Width* to 500% as per image on right. Click *Close*.



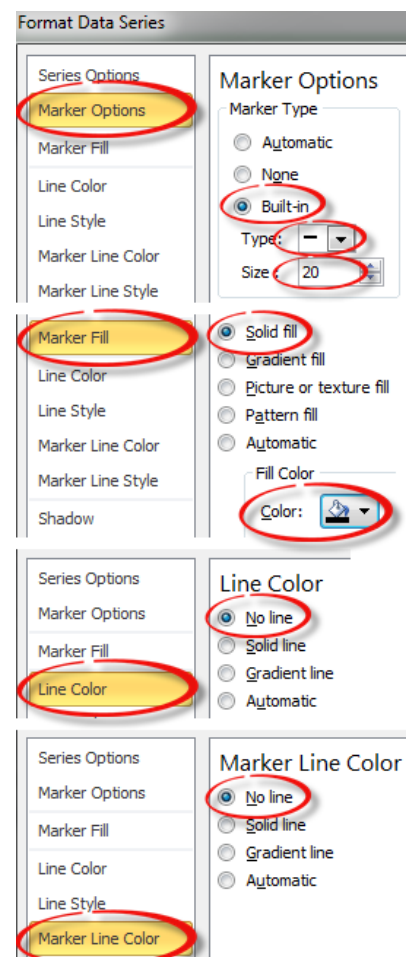
4. Delete the right-hand vertical Axis. This creates a separate, thinner column for the score.
5. Right click the Target data series (bottom segment of the stacked column) and select *Change Series Chart Type*. Select the *Line with Markers* icon. See image on right. Click *OK*.



This will insert a single point on the chart see image on right.

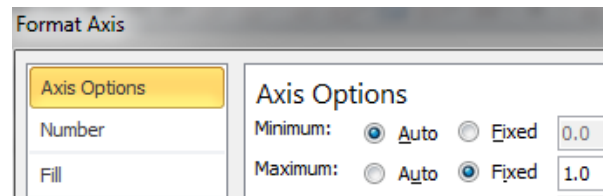


Right click that point and choose *Format Data Series*. Click the *Marker Options* and choose the *Built-in* option. From the *Type* drop-down select the widest horizontal line. Change the *Size* to 20. Click the *Marker Fill* option and choose *Solid fill* and select the black colour in the *Fill Color* section. Click the *Line Color* option and select *No line*. Click on the *Marker Line Color* option and select *No Line*. Image on right shows all four settings. Click *Close*.



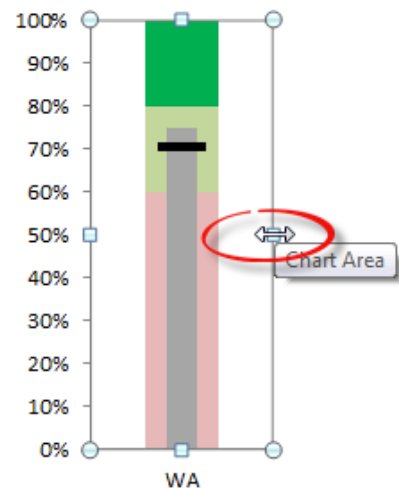
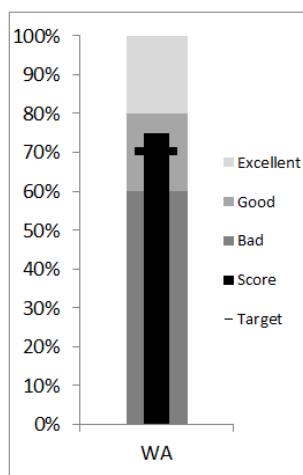
6. Change the colour for the score column to grey, change bad to dark grey, change good to medium grey and excellent to light grey. You can select each column segment and use the *Fill Color* option on the Home ribbon tab to make the colour changes.

7. Delete the chart gridlines, Change maximum value on the vertical axis by right clicking the vertical axis and selecting *Format Axis*. Change the *Maximum* option to *Fixed* and the value to 1.0, and then click *Close*. As per image on right



8. Nearly done, reduce the width of the plot area using the double-headed arrow icons around the edge of the plot area, see image on right.

Final chart.



Having created a single bullet chart you can now add extra bullet charts by adding columns to the right and amending the data series to incorporate the extra columns. The easy way to do that is to select the chart and note the coloured lines on the cells, see image below. Use your mouse to example the range to the right.

	A	B
1		WA
2	Score	75%
3	Target	70%
4	Bad	60%
5	Good	20%
6	Excellent	20%
7		

	A	B	C
1		WA	NSW
2	Score	75%	85%
3	Target	70%	75%
4	Bad	60%	70%
5	Good	20%	20%
6	Excellent	20%	10%
7			

September 14 - Handing Badly Laid Out Data

(Note: the sheet names used in this example are the same as those in the article)

A certain commonly used accounting package has a unique way of providing reports to Excel. The reports tend to be difficult to use in their downloaded form (see example below) and they need to be converted into a proper data layout so they can be used.

	B	C	D	E	F	G	H
1							
2				ABC PTY Ltd			
3							
4							
5							
6							
7				Card List [Detail]			
8							
9							
10							
11	John Smith						
12			Card ID: 12345				
13							
14			First Address: 25 Big St				
15			Second Address:				
16			City: Fremantle				
17			State: WA				
18			Postcode: 6160				
19			Phone #1: 08 1234 5678				
20			Phone #2:				
21			Email: john.smith@gmail.com				
22				Date of Birth: 21/05/1961			
23							
24							
25							
26							
27							

Even though this layout is hard to work with, it is consistent throughout the report and that means we can setup formulas to extract the data and convert it into a data layout.

This download report is in a sheet called Data.

We have a companion report that lists all the employees.

This makes the job easier. The employees are listed in the Staff sheet as shown on the right starting in row 11.

We will populate a Report sheet using three separate formulas.

All formulas use the INDEX function, mentioned in an earlier article.

The layout of the Report sheet is shown below. Note the sequence of the columns matches the sequence of the data rows in the downloaded report. Column C is there to make the final formula easier to create and will be explained later.

	A
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	Employee
11	John Smith
12	Sally Tan
13	Henry Ng
14	

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Card ID	Blank	First Address	Second Address	City	State	Postcode	Phone #1	Phone #2	Email	Date of Birth
2	John Smith	12345		25 Big St		Fremantle	WA	6160	08 1234 5678		john.smith@gmail.com	21/05/1961
3	Sally Tan	98765		19 Little Road		Perth	WA	6000	08 8765 4321		S_Tan@hotmail.com	22/02/1978
4	Henry Ng	3333333		3 Victoria Avenue		KALGOORLIE	WA	6430	08 9876 5432		henry.ng@gmail.com	1/08/1973

The first formula we need is in cell A2 in the Report sheet is.

=INDEX(Staff!A:A,ROW()+9)

This extracts the first name from the Staff sheet.

It assumes the names start in row 11 of the Staff sheet. The ROW()+9 part of the formula creates a sequential number starting at a 11.

ROW() returns the current row number. In cell A2 it returns 2. We add 9 to it to equal 11 - the row where the names start. As the formula is copied down, the number increments to 12, 13 etc.

Now we have the names we want to extract. We can find them in the Data sheet by using the MATCH function.

This technique works because the layout of the report is consistent as you go down the sheet - the name is always in column B and the very next row has the Card Id. Then there is a blank row which is mirrored in our Report sheet as column C which has the title Blank. This makes the formula easier to create and copy across.

The other pieces of the data are listed in sequence down the column one row after the other except the date of birth which will require a separate formula. The second formula is in cell B2 and it is

=IFERROR(INDEX(Data!\$E:\$E,MATCH(\$A2,Data!\$B:\$B,0)+COLUMN()-1),"")

We will cover the IFERROR a little later. Initially we will look at

INDEX(Data!\$E:\$E,MATCH(\$A2,Data!\$B:\$B,0)+COLUMN()-1)

This returns the data from column E of the Data sheet. We can figure out which row to extract by finding the name in column B and adding one to the result. That's what the MATCH function allows us to do.

The MATCH function finds row number of the entry in column B. Then we use the COLUMN function to return the current column number. In this case, in cell B2 the result from the COLUMN() function is 2. We subtract one from it because we want to add one to the MATCH result. As we copy this formula across the columns, the number added will increase to 2, 3 and 4 etc. The COLUMN() function will return a sequential value as we copy it across. This causes the row used in the INDEX to increase and make its way down the Data sheet to extract the data one row at a time.

The final entry is the date of birth which is in column G of the report so we need to modify the formula in L2 which is the same, except it extracts from column G.

=IFERROR(INDEX(Data!\$G:\$G,MATCH(\$A2,Data!\$B:\$B,0)+COLUMN()-1),"")

The IFERROR part handle of the formula handles the situation where there is a zero in column A of the Report sheet. The IFERROR function handles all of Excel's errors and you can specify what to display. In this case a blank cell "" is displayed if an error is encountered.

IFERROR was added in Excel 2007 and only works in versions from Excel 2007 onwards.