

# Sum Product

NEWSLETTER #127 - June 2023

[www.sumproduct.com](http://www.sumproduct.com) | [www.sumproduct.com/thought](http://www.sumproduct.com/thought)



*Halfway through the year already (well, we will be at the end of the month...)! The Power BI Updates appear to have been paused as Microsoft makes two special announcements, introducing **Fabric** and **Copilot in Power BI** to the world. But that's not all – a we have plenty of other content in this newsletter to keep you occupied too!*

For example, we have some good news for Yours Truly who has been recognised on something other than *Crimestoppers*. I don't expect to win in such illustrious company but it's fantastic to be a finalist for the Lifetime Achievement category in the Global Excel Awards. You can read more below.

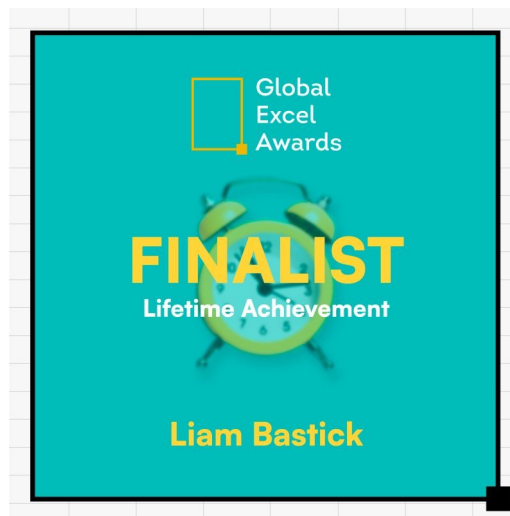
This month we consider single element slicing (a neat trick for dashboard creation in Excel). And then there are all the usual newsletter chums such as the Beat the Boredom Challenge, Charts & Dashboards, reticulating Visual Basics, Power Pivot Principles, Power Query Pointers, Excel Updates, more Keyboard Shortcuts and the A to Z of Excel functions brought to you by **LEN**.

As always, happy reading and remember: stay safe, stay happy, stay healthy.

*Liam Bastick*, Managing Director, SumProduct



## Global Excel Awards Lifetime Achievement Award Finalist



SumProduct is pleased to announce that **Liam Bastick** has been made a finalist for the Global Excel Awards Lifetime Achievement Award. Here is Liam's response to the news:

"I decided to see who the other nominees were for this great award - and they are no other than two of my (past) MVP peers and friends **Bill Jelen** and **Bob Umlas**. I am happy to settle for the Bronze medal in this category!

"Bill is no less than Mr Excel himself (that's his website) and I have learned a tremendous amount from him and his army of contributors over the years (so much so we used to name it "AskPablo.com"). He is a great guy and very knowledgeable on all things Excel.

"Bob is another legend of Excel. I cannot believe he has not been nominated in the "Most Obscure Uses of Excel" category (he could fill that on his own with no disrespect to the worthy nominees therein). Like Bill, he has written many books too.

"I feel a bit of a fraud being placed in the same section as these two greats. Therefore, I am not asking for anyone to vote for me; I am asking you to vote for the awards themselves and pick your favourites. There are some great (and well-known names) in all the categories including another buddy of mine, **Leila Gharani**, to name but one (her sense of humour is nearly as bad as mine!).

"All of the nominees are champions of Excel. Please "do your thing" and help to keep Excel on the map going forward. Don't vote for me - vote for Excel. And spread the word, let's have LOTS of votes for everybody! Good luck one and all."

Voting closes on 4<sup>th</sup> June 2023, and the winners will be announced on 19<sup>th</sup> June 2023. You can vote here: <https://globalexcelsummit.com/global-excel-awards>.



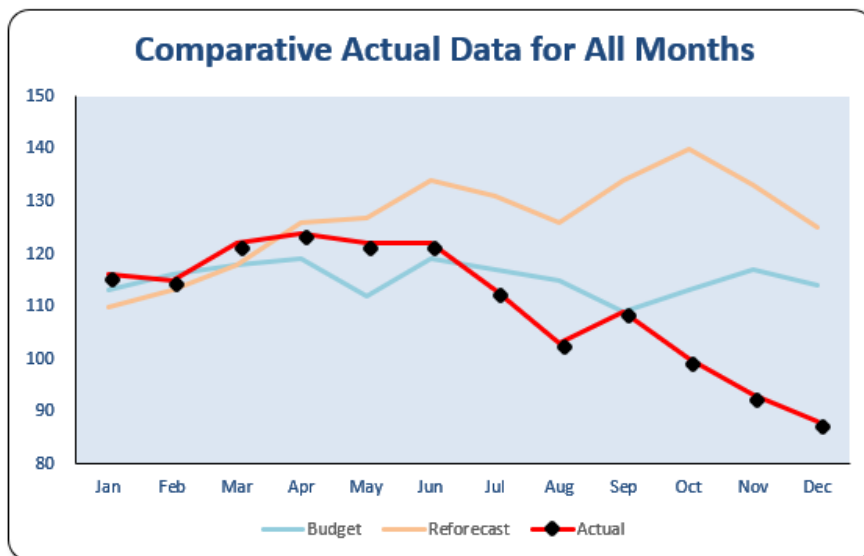
## Slicing One Element on a Chart Only

We tend to alternate the topics we select for this section of this newsletter, and this month we thought we would address a recent reader's question: how do you slice on just one element of a chart?

Consider a scenario where you create a chart, say, a line chart, but you only want one element (one of the lines) to be modified using a slicer. To illustrate, let's have some data:

Month	Budget	Reforecast	Actual
Jan	113	110	116
Feb	116	113	115
Mar	118	118	122
Apr	119	126	124
May	112	127	122
Jun	119	134	122
Jul	117	131	113
Aug	115	126	103
Sep	109	134	109
Oct	113	140	100
Nov	117	133	93
Dec	114	125	88

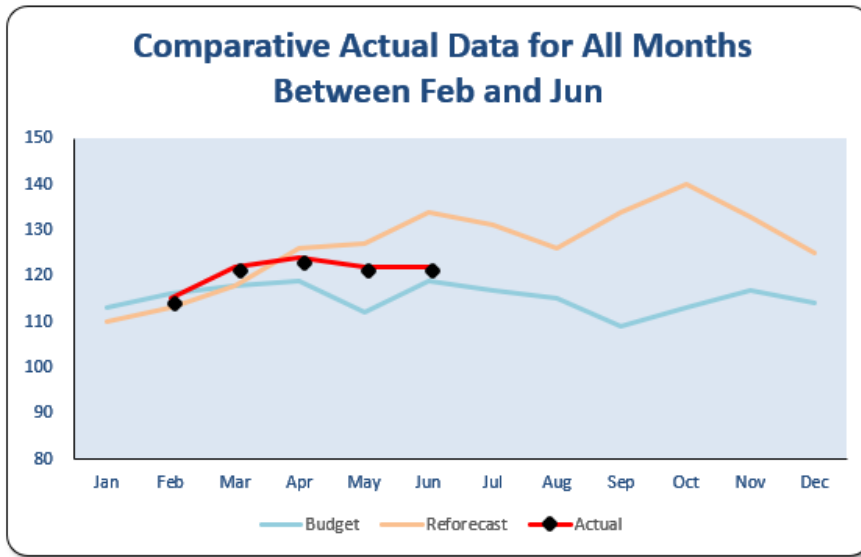
Imagine you were asked to put the following chart together with an associated slicer:



So far, so good, but the slicer should only affect the Actual line item (in red, with the black marker in our example). For example:

Month ☰ ⏏

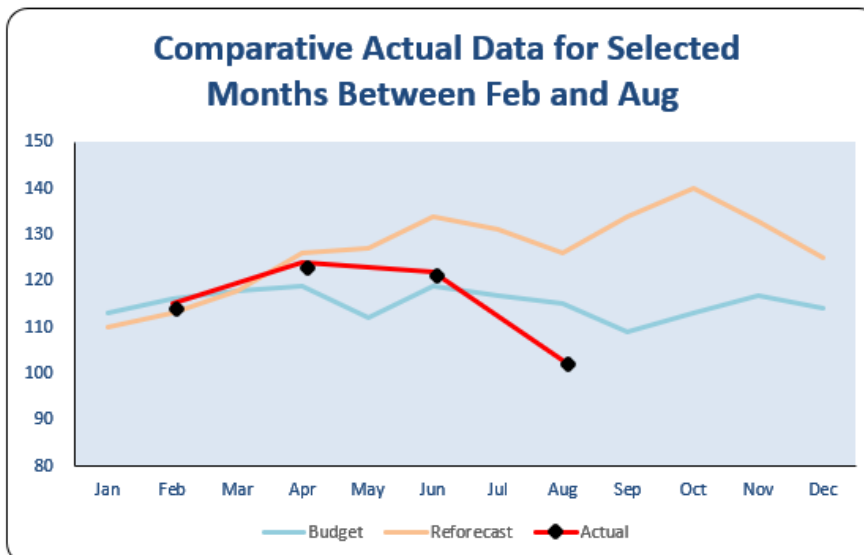
Jan	Feb	Mar
Apr	May	Jun
Jul	Aug	Sep
Oct	Nov	Dec



or:

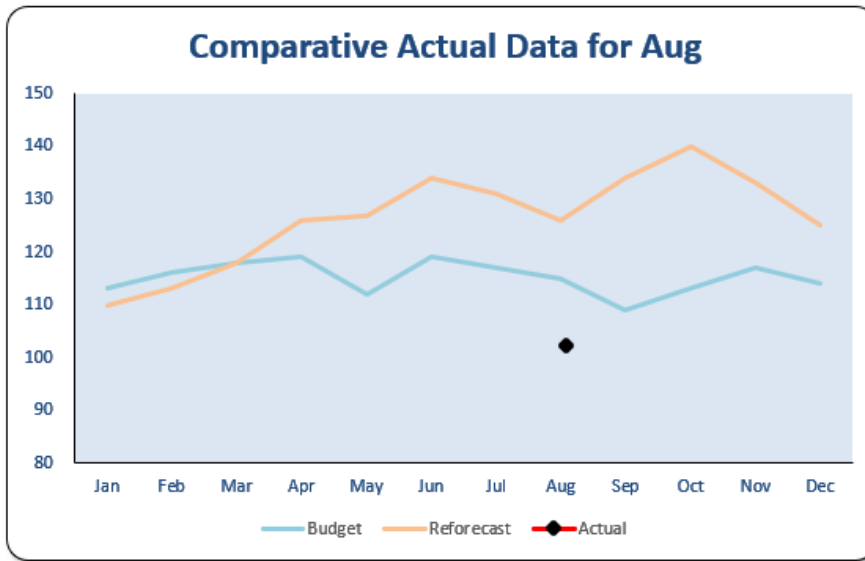
Month ☰ ⏏

Jan	Feb	Mar
Apr	May	Jun
Jul	Aug	Sep
Oct	Nov	Dec



or finally:

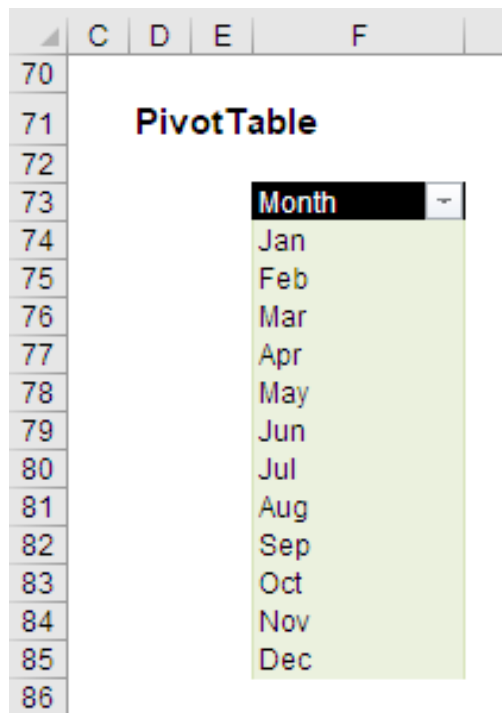
Month		
Jan	Feb	Mar
Apr	May	Jun
Jul	Aug	Sep
Oct	Nov	Dec



Do you see how the chart title changes automatically? The question is: how do you do it? First of all, our data has been placed in the following cells (shown just so you may follow the formulae):

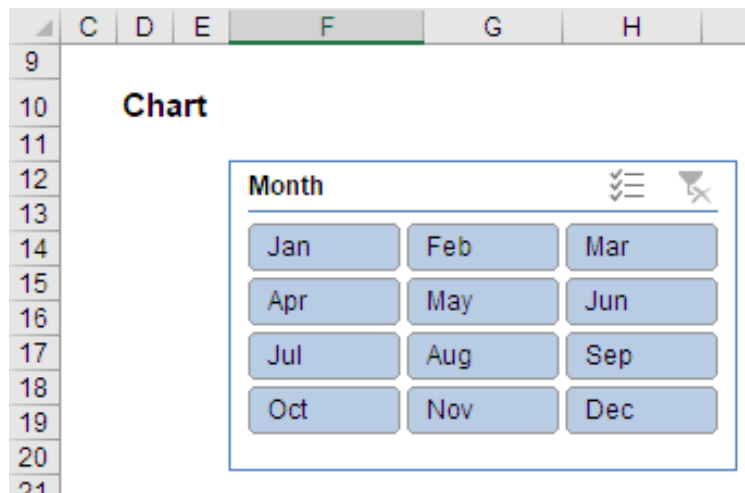
	C	D	E	F	G	H	I
49							
50		<b>Original Data</b>					
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							

Before I play with this, I am going to highlight this entire table (cells **F52:I64**) and create a PivotTable (**ALT + N + V**, or go to the 'Insert' tab on the Ribbon and click on 'PivotTables' in the 'Tables' grouping) showing the months only in rows as follows:

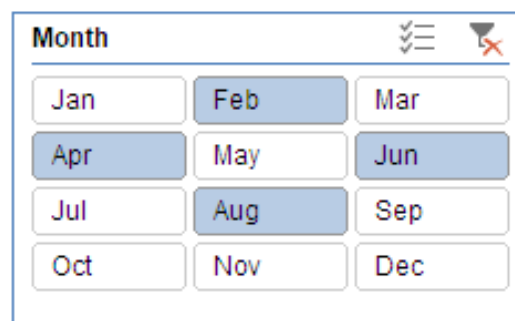


I have to have a PivotTable or a Table to create a Slicer. I tend to fall back on PivotTables (even though they are slightly more complex) as you can have one Slicer control multiple PivotTables which isn't possible with multiple Tables.

Next, I add a Slicer for the 'Months' field (ensure your cursor is situated in the PivotTable and then use the keyboard shortcut **ALT + JT + SF**, or else select 'Insert Slicer' from the 'Filter' grouping on the contextual 'PivotTable Tools' tab 'Analyze' on the Ribbon).



Since the Slicer has originated from the created PivotTable, changing the Slicer, *e.g.*



automatically manifests the same modifications in the source PivotTable, viz.



I can use this relationship to create a modified data table:

	C	D	E	F	G	H	I	J	K
91									
92				<b>Data Use in Chart</b>					
93									
94				<b>Month</b>	<b>Budget</b>	<b>Reforecast</b>	<b>Actual</b>	<b>Month Used</b>	<b>Cumulative</b>
95				Jan	113	110	#N/A	-	-
96				Feb	116	113	115	1	1
97				Mar	118	118	#N/A	-	-
98				Apr	119	126	124	1	2
99				May	112	127	#N/A	-	-
100				Jun	119	134	122	1	3
101				Jul	117	131	#N/A	-	-
102				Aug	115	126	103	1	4
103				Sep	109	134	#N/A	-	-
104				Oct	113	140	#N/A	-	-
105				Nov	117	133	#N/A	-	-
106				Dec	114	125	#N/A	-	-
107									

Columns F, G and H are based on the source data, but columns I:K have been calculated differently:

- Starting with column J (rather than column I), this column contains the formula  
**=COUNTIF(\$F\$74:\$F\$85,\$F95)**

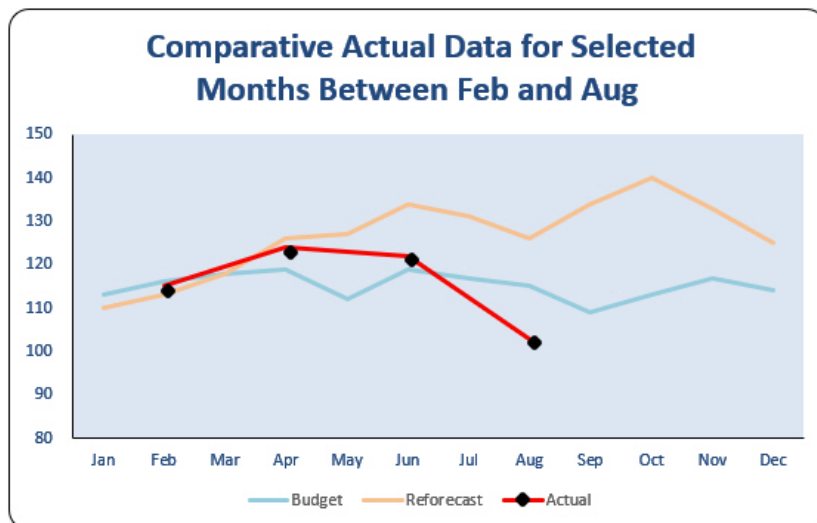
in row 95. The months only occur once in the PivotTable, so this **COUNTIF** function counts one [1] if the month is present in the PivotTable (*i.e.* the month was selected in the Slicer) and zero [0] otherwise

- Column I contains the formula  
**=IF(J95,I53,NA())**

which references the corresponding actual data for the month provided the value in column J is TRUE, *i.e.* any value other than zero. Since I have already explained the only other value is 1, this formula is including the actual data if the Slicer has selected the month and puts #N/A otherwise. Whilst *prima facie* errors are usually discouraged in spreadsheets, in this case #N/A causes the value not to display in the chart at all.

It should be noted you might have considered the **SUBTOTAL** or **AGGREGATE** functions instead, but as complex as they are, if the PivotTable were to be hidden, the wrong results might occur. Sometimes, simpler is better!

The cells **F94:I106** are all that are required to display the chart. Simply select this range and create a line chart (**ALT + N + N**), or else select a lie chart from the 'Charts' section of the 'Insert' tab of the Ribbon):



- Finally, column K is required to assist with the chart title, which I haven't yet explained. The formula here,

**=SUM(\$J\$95:\$J95)\*\$J95**

keeps a running total of all the months displayed (i.e. the first month selected is 1, the second is 2, etc.). This is useful as the number 1 relates to the earliest month and the maximum value relates to the last month selected. This latter number also represents how many months have been selected too.

In fact, the labelling requires several preliminary calculations, viz.

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
112															
113				<b>Calculations Required</b>											
114															
115				All Months Selected?				FALSE							=SUM(J95:J106)=ROWS(J95:J106)
116				Only One Month Selected?				FALSE							=SUM(J95:J106)=1
117				Earliest Month Selected				Feb							=INDEX(\$F\$95:\$F\$106,MATCH(1,\$J\$95:\$J\$106,0))
118				Earliest Month Number				2							=MATCH(J117,\$F\$95:\$F\$106,0)
119				Latest Month Selected				Aug							=INDEX(\$F\$95:\$F\$106,MATCH(MAX(\$K\$95:\$K\$106),\$K\$95:\$K\$106,0))
120				Latest Month Number				8							=MATCH(J119,\$F\$95:\$F\$106,0)
121				Months in Range				7							=J120-J118+1
122				Months Missing in Selected Range?				TRUE							=MAX(K95:K106)<>J121
123															

Let me go through them:

The formula in cell J115, 'All Months Selected?'

**=SUM(J95:J106)=ROWS(J95:J106)**

checks to see whether the total of all of the 1's from column I (above) equals the number of rows in the range. This can only happen if all months have a '1' allocated to them, i.e. all months have been selected.

The formula in cell J116,

**=SUM(J95:J106)=1**

checks that one and only one month has been selected.

Cells J117 and J119 contain similar formulae. For example, the formula in the former cell,

**=INDEX(\$F\$95:\$F\$106,MATCH(1,\$J\$95:\$J\$106,0))**

returns the name of the first month selected chronologically (the other formula identifies the last period selected). This uses the INDEX MATCH combination which we have explained previously.

Cells J118 and J120 use similar calculations too. The first formula

**=MATCH(J117,\$F\$95:\$F\$106,0)**

returns the corresponding month number (which is what the other formula does too). Therefore, the formula in cell J121

**=J120-J118+1**

determines how many months would be included if all months between the first and last months had been selected. This is because the final formula in cell J122,

**=MAX(K95:K106)<>J121**

checks whether the number of months selected equals the number of months in total between the first and last months previously selected. If a month or more is missing, the result will be TRUE instead of FALSE.

These interim calculations are used as follows to create the chart title all in one cell:

	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S		
125																	
126				Chart Title			Comparative Actual Data for Selected Months Between Feb and Aug										
127																	
128																	

Cell J126 contains the formula

**= "Comparative Actual Data for "&  
IF(J115,"All Months",IF(J116,J117,I  
F(J122,"Selected","All")&  
" Months Between "&J117&" and "&J119))**

This concatenated formula will display the details required, e.g. "Comparative Actual Data for All Months or "Selected Actual Data for Aug").

To get the formula from cell J126 into the chart itself, add a chart title, and while it is selected, click on the formula bar, type '=' and then click on cell J126. Don't type the formula =J126 or =Example!\$J\$126 in. In some versions of Excel, for this to work, the cell has to be selected on the sheet for the title to become formulaically dynamic.

Simple (sort of)!



## Beat the Boredom Challenge

With many of us currently "working from home" / quarantined, there are only so Zoom / Teams calls and virtual parties you can make before you reach your (data) limit. Perhaps they should measure data allowance in blood pressure millimetres of mercury (mmHg). To try and

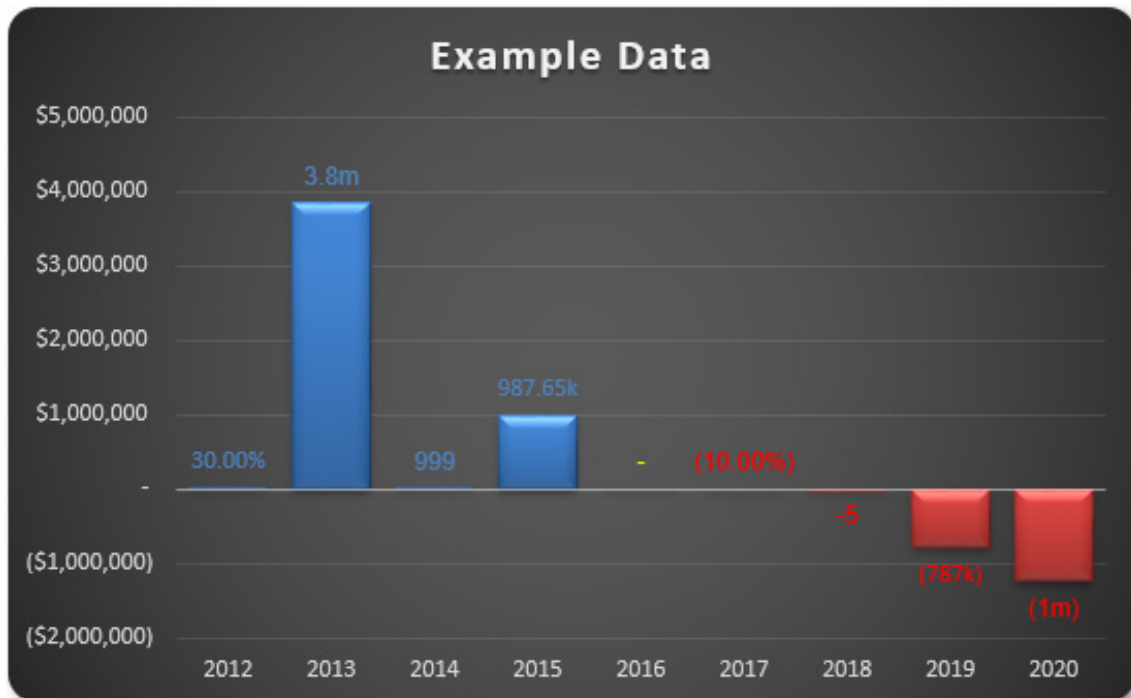
keep our readers engaged, we will continue to reproduce some of our popular **Final Friday Fix** challenges from yesteryear in this and upcoming newsletters. One suggested solution may be found later in this newsletter. Here's this month's...

Consider the following pre-formatted chart data (all of the following inputs are numbers typed in, not text):

### Chart Data

Year	Value
2012	30.00%
2013	3.8m
2014	999
2015	987.65k
2016	-
2017	(10.00%)
2018	-5
2019	(787k)
2020	(1m)

All you have to do is create the associated Excel chart with replicated label formatting:



This chart has to work after you have closed the file and reopened...

This month's challenge *might* just drive you mad!

Sounds easy? Try it. One solution *just might* be found later in this newsletter – but no reading ahead!



## Charts and Dashboards

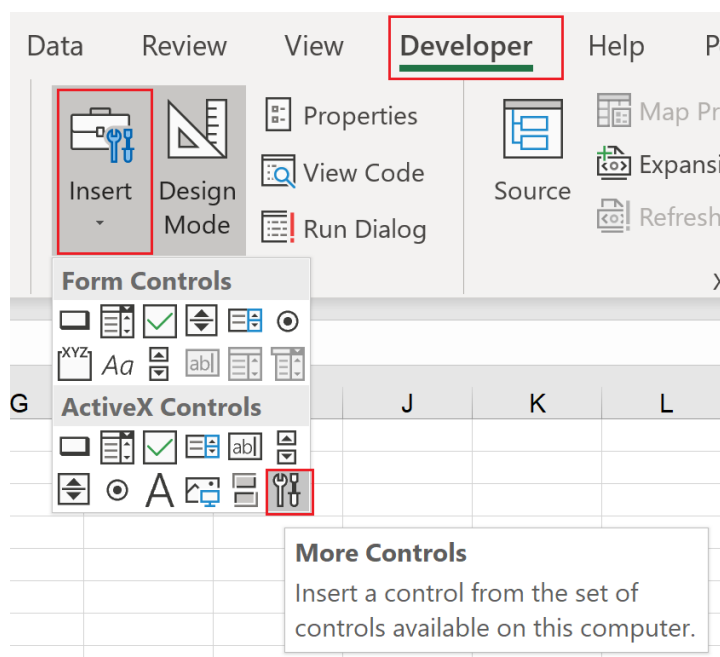
*It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we again look at ActiveX controls in Excel.*

In addition to the various Form Controls we have been through in recent months, ActiveX Controls may be included in Excel or other programs. An ActiveX Control is a component program object that can be re-used by many application programs within a computer or among computers in a network. Whilst Form Controls are built into Excel, ActiveX Controls are loaded separately. Generally, Form Controls are used based on their simple nature, while ActiveX Controls allow for more flexible design and should be used when the task is not able to be done with the more basic Form Controls.

When installing new ActiveX Controls, the setup program for the controls usually register each control on the computer, which makes it available

to use from Excel. If the control does not appear in the list, we must register it manually. It is also worth noting that, not all ActiveX Controls can be used directly on worksheets; some can be used only on Microsoft Visual Basic for Applications (VBA) UserForms. When working with these controls, Excel displays the message 'Cannot insert object' if we try to add them to a worksheet incorrectly.

From the Developer tab on the Ribbon, in the Controls group, click Insert, and then under 'ActiveX Controls', select a control, or click 'More Controls' to view all the available ActiveX controls. Then:



- click the worksheet location where you want the ActiveX control to appear
- to edit the control, make sure that you are in the Design mode by going to the Developer tab, and in the Controls group, turn on 'Design Mode'
- to specify the control properties, on the Developer tab, in the Controls group, click Properties (or right-click on the control then choose Properties). In the Properties dialog, for detailed information about each property, select the property, and then press **F1** to display a 'Visual Basic Help' topic, or type the property name in the 'Visual Basic Help' search box.

To register an ActiveX Control, first click the ActiveX Control that you want to register, making sure you do this in the Design Mode (**Developer -> Controls -> Design Mode**). Next, choose the 'More Controls' under 'ActiveX Control', similar to when selecting a control (above). Then, at the bottom of the 'More Controls' dialog, click 'Register Custom'.

In here, locate the folder containing the control file (with **.ocx** file name extension) or dynamically link the library file (with **.dll** file name extension) for the control you wish to register. Finally, select the file for the control, click 'Open'.

More next month...

## Visual Basics

We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month we get hooked on speed.



Sometimes a Macro is still INCREDIBLY slow to run and you're still not sure why. Obviously, it is "Reticulating Splines" (a programmer's in-joke no more than technobabble)! But if the code is extensive, with considerable number crunching for example, how do we speed it up?

Well there are a few ways that you can do this. Over the next few newsletters we'll look at a few tips on how you can do that.

### **No Calculations!**

You might be "Calculating Llama Expectoration Trajectories" which require a lot of numbers being copied back and forth over the spreadsheet. Each time a new value is entered in a spreadsheet, Excel recalculates all the cells that refer to it. The same thing happens in VBA.

Excel's calculation engine is still running in the background and every time VBA writes to the workbook. It is highly recommended that you turn OFF automatic calculations before running the steps in the Macro.

You could just turn it off in the Formula tab of the Ribbon but that's a manual action. We could easily automate this using the command:

```
Application.Calculation = xlCalculationManual
```

This will turn off automatic calculations. If you do need to Excel to recalculate cell values for a later step in the program you can just simply add the line:

```
Calculate
```

Top tip: This can be limited to even a specific Range or Worksheet as Calculate is also an object Method (e.g. `Range("A1").Calculate`). But remember, you'll need to turn it back on at the end!

For automatic calculation simply use:

```
Application.Calculation = xlCalculationAutomatic
```

If you prefer the mode 'Automatic (except for Data Tables)' use:

```
Application.Calculation = xlCalculationSemiautomatic
```

However, given that we do not know where our macro may end up or what the end users preferred settings are, the best practice way would be to store the Calculation property of the Application and revert it at the end.

```
Sub StopReticulatingSplines()  
    Dim CalcMode As Integer  
    CalcMode = Application.Calculation  
    Application.Calculation = xlCalculationManual  
    Call DoStuff  
    Application.Calculation = CalcMode  
End Sub
```

Next month, we'll review more optimisation tricks.

## Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we discuss how to create conditional columns with multiple criteria, er, again.

In this month's newsletter, we discuss the different methods to create conditional custom columns in Power Pivot, this time using the **OR** function and the **'|'** operator. Let's have a look at the following data set...again:

	A	B	C	D	E
1					
2		Category 1	Category 2	Category 3	Customer Name
3		Y	Y	0	Harry
4		Y	N	1	Joe
5		N	Y	1	Kat
6		N	N	1	Nicole
7		Y	Y	2	Monia
8		N	Y	0	Jeff
9		N	Y	0	Petra
10		N	N	1	Lenny
11		Y	Y	2	Gavin
12					

We have a Table with three categories and a column with customer names. We are going to assume that this Table has been retrieved by Power Query (otherwise known as Get & Transform) and loaded into our data model in Power Pivot.

For the purposes of this exercise, let's assume that we wish to create a conditional column that will return with the value "Included" if a customer is "Y" in Category 1 or "N" in Category 2.

We would use the following code to create our custom column:

```
=IF(OR([Category 1]="Y", [Category 2]="N"),"Included",BLANK())
```

As we can see our logic works and we get the desired result:

	Category 1	Category 2	Category 3	Customer Name	Included ...
1	Y	Y	0	Harry	Included
2	Y	N	1	Joe	Included
3	N	Y	1	Kat	
4	N	N	1	Nicole	Included
5	Y	Y	2	Monia	Included
6	N	Y	0	Jeff	
7	N	Y	0	Petra	
8	N	N	1	Lenny	Included
9	Y	N	2	Gavin	Included

What if we want a conditional column with three conditions?

- Category 1 = "N"
- Category 2 = "Y"
- Category 3 > 1

Keeping in mind that the **OR** function in Power Pivot only allows for two logical statements:

```
OR(Logical 1, Logical 2)
```

We will have to use a dreaded nested **IF** formula:

```
=IF(OR([Category 1]="N",OR([Category 2]="Y",[Category 3]>1)),"Included",BLANK())
```

We achieve the intended result:

	Category 1	Category 2	Category 3	Customer Name	Included ...	Included 3 Criteria
1	Y	Y	0	Harry	Included	Included
2	Y	N	1	Joe	Included	
3	N	Y	1	Kat		Included
4	N	N	1	Nicole	Included	Included
5	Y	Y	2	Monia	Included	Included
6	N	Y	0	Jeff		Included
7	N	Y	0	Petra		Included
8	N	N	1	Lenny	Included	Included
9	Y	N	2	Gavin	Included	Included

That's all well and good, but what if we have four or five criteria? We do have an alternative we can use the '||' operator.

The '||' operator serves as a substitute to the **OR** function so we can write the following formula:

**=IF([Category 1]="N" || [Category 2]="Y" || [Category 3]>1,"Included",BLANK())**

	Category 1	Category 2	Category 3	Customer Name	Included ...	Included 3 Criteria	Included 3 Criteria
1	Y	Y	0	Harry	Included	Included	Included
2	Y	N	1	Joe	Included		
3	N	Y	1	Kat		Included	Included
4	N	N	1	Nicole	Included	Included	Included
5	Y	Y	2	Monia	Included	Included	Included
6	N	Y	0	Jeff		Included	Included
7	N	Y	0	Petra		Included	Included
8	N	N	1	Lenny	Included	Included	Included
9	Y	N	2	Gavin	Included	Included	Included

The '||' operator allows us to combine several criteria into one logical test for the **IF** formula drastically simplifying the formula. That being said, the '||' operator is a great way to avoid creating nasty nested **IF** formulae when creating our custom columns in Power Pivot.

That's it for this month; more next time.

## Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from [www.sumproduct.com/blog](http://www.sumproduct.com/blog). If you wish to read more in the meantime,

simply check out our Blog section each Wednesday. This month, we consider ways to determine the end (and the start) of a particular date period in **M**.

This month, we will take a look at some **Date() M** functions that can be used to return a date / time that represents the end or beginning of a particular period containing that data, for example, the end of the current year. Since these functions come in pairs (end and start for each time period), we will look at them in pairs, and give an example for each

pair of functions. We're including these functions in this mini-series on dates mainly because they can be useful in calculations. These functions are also available under the 'Date' dropdown in the 'Add Column' tab from the Power Query Editor screen, but however they are called, it's important to know what value is returned in order to use them correctly.

### Date.StartOfMonth and Date.EndOfMonth

**Date.StartOfMonth**(datetime as nullable datetime) as nullable datetime

This returns a datetime value representing the start of the month that contains **datetime**. Similarly,

**Date.EndOfMonth**(datetime as nullable datetime) as nullable datetime

returns a datetime value for the end of the month that contains **datetime**.

We may look at the start and end of the month containing the final payment date:



Final Payment	End of Month of Final Payment	Start of Quarter of Final Payment	End of Quarter of Final Payment
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00
09/2016 00:00:00	01/10/2016 00:00:00	01/07/2016 00:00:00	01/10/2016 00:00:00

The M formulae used are:

= Date.StartOfQuarter([Final Payment])

= Date.EndOfQuarter([Final Payment])

We can then see the start and end of the financial quarter that the final payment takes place in.

**Date.StartOfWeek and Date.EndOfWeek**

**Date.StartOfWeek**(datetime as nullable datetime, optional firstday as nullable number) as nullable datetime

This returns a datetime value representing the start of the week containing datetime. Similarly,

**Date.StartOfWeek**(datetime as any, optional firstday as nullable number) as any

returns a datetime value for the end of the week containing datetime.

We have a little more flexibility here, as we may decide what day the week starts on using the value firstday, as we did in the last newsletter, with the function Date.DayOfWeek().

Valid values are: Day.Sunday, Day.Monday, Day.Tuesday, Day.Wednesday, Day.Thursday, Day.Friday, and Day.Saturday. If we don't specify a firstday then the default is Monday.

We will look at the start and end date of the week that final payment takes place by adding two columns.

Final Payment	End of Quarter of Final Payment	Start of Week of Final Payment	End of Week of Final Payment
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	19/09/2016 00:00:00	26/09/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	19/09/2016 00:00:00	26/09/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	19/09/2016 00:00:00	26/09/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/10/2016 00:00:00	01/01/2017 00:00:00	03/10/2016 00:00:00	10/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00
01/07/2016 00:00:00	01/10/2016 00:00:00	26/09/2016 00:00:00	03/10/2016 00:00:00

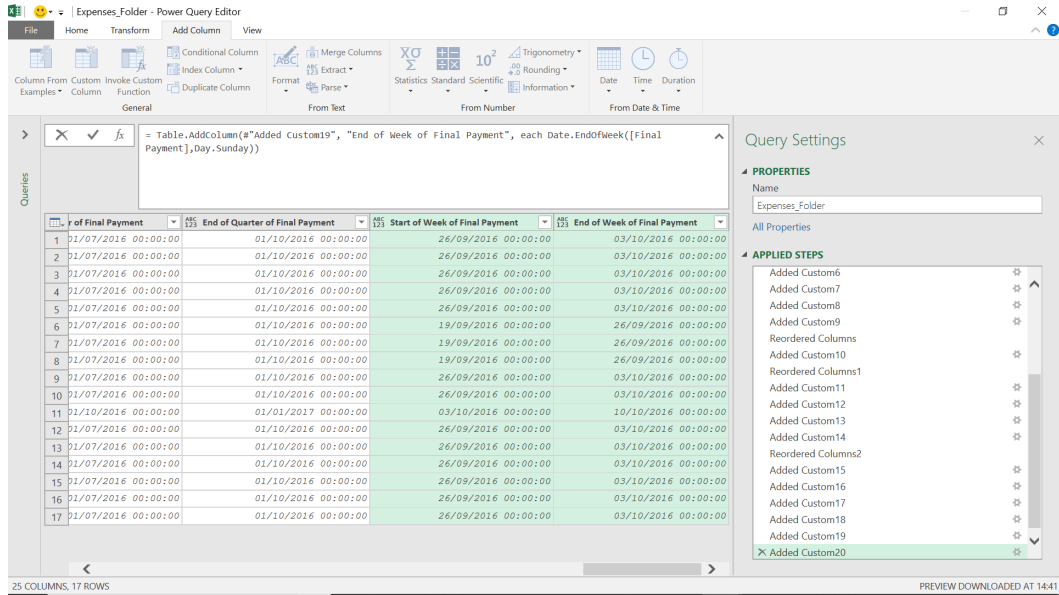


The M formulae used are:

**= Date.StartOfWeek([Final Payment])**

**= Date.EndOfWeek([Final Payment])**

We have taken the default start of the week, which is Monday. It's important to note that the end of week date also defaults to a Monday, which will affect any calculations. The end of week day will be the same as the start of week day, so if we change the end of week step to use Sunday as the start of the week,



the end of the week date is now a Sunday too.

**Date.StartOfYear and Date.EndOfYear**

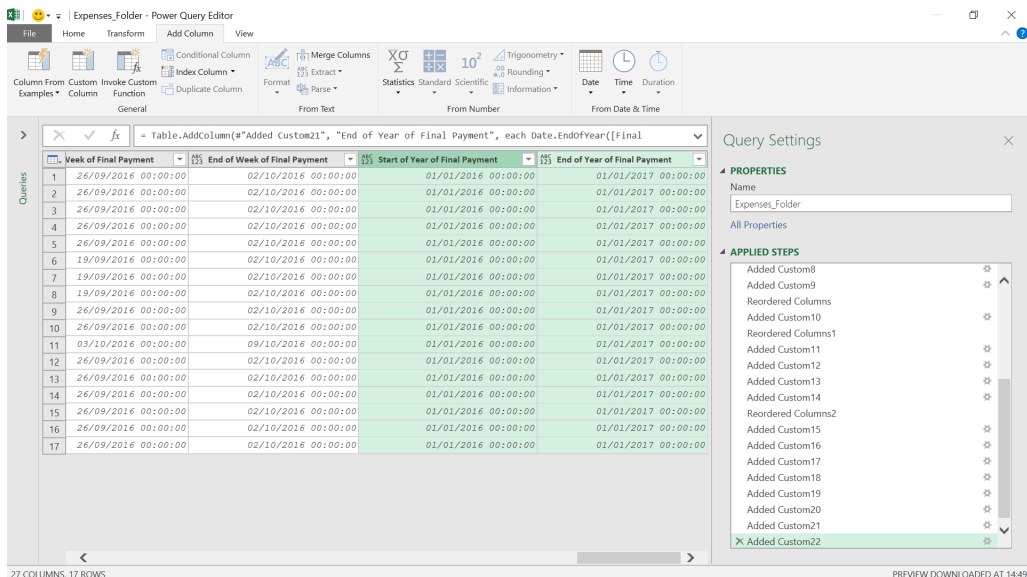
**Date.StartOfYear(datetime as nullable datetime) as nullable datetime**

This returns a datetime value representing the start of the year containing **datetime**. Similarly,

**Date.EndOfYear(datetime as nullable datetime) as nullable datetime**

returns a datetime value for the end of the year containing **datetime**.

We will look at the start and end date of the year that final payment takes place by adding two columns.





The M formulae I have used are:

= Date.StartOfYear([Final Payment])

= Date.EndOfYear([Final Payment])

It's useful to know that the end of the year here is actually the first day of the next year, which is important when constructing calculations.

### Date.StartOfDay and Date.EndOfDay

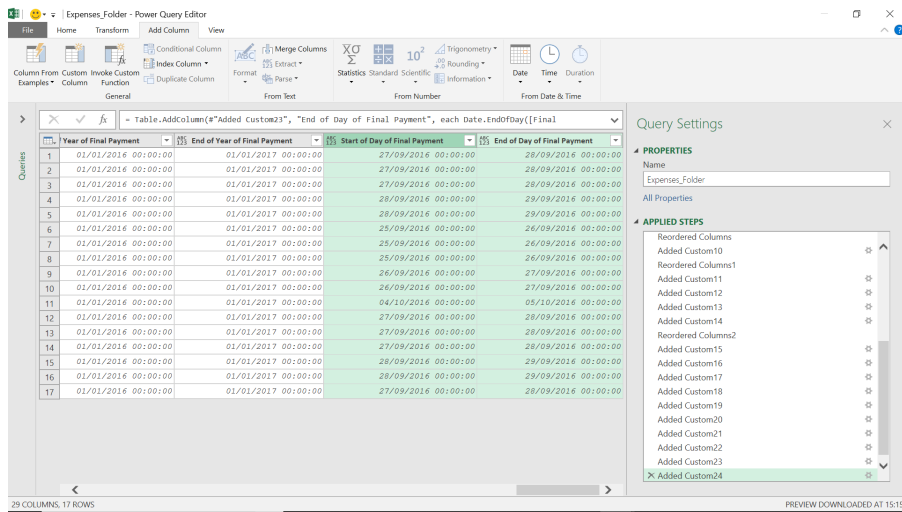
**Date.StartOfDay(datetime as nullable datetime) as nullable datetime**

This returns a datetime value for the start of the date of **datetime**. Similarly,

**Date.EndOfDay(datetime as nullable datetime) as nullable datetime**

returns a datetime value for the end of the date of **datetime**.

I will look at the start and end of the date that final payment takes place by adding two columns.



The M formulae used are:

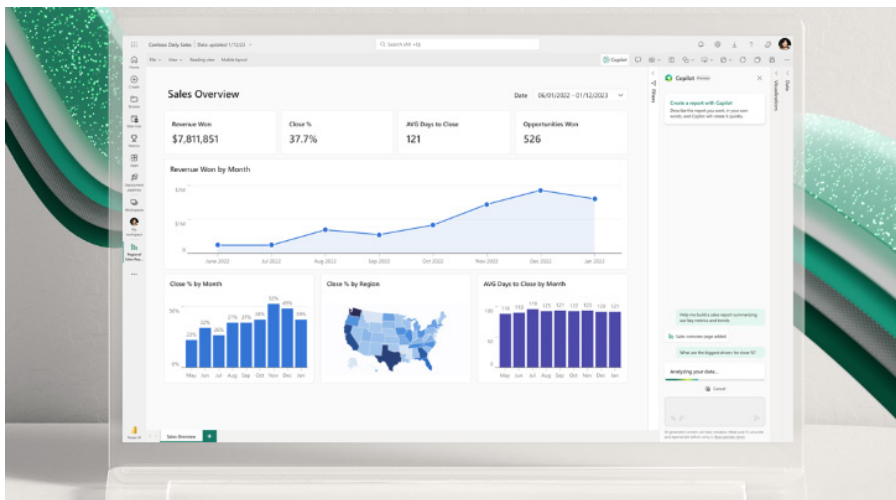
= Date.StartOfDay([Final Payment])

= Date.EndOfDay([Final Payment])

I can see that the end of day function actually gives me the start of the next day, which I would need to know if I included it in a calculation.

More next time.

## Power BI Updates: Special Announcements



The “normal” Power BI updates appear to have been “rested” this month; this appears to be due to the “special announcement” of introducing both **Fabric** and **Copilot** into the hallowed halls of Power BI.

Businesses have been collecting and storing massive amounts of data from applications, services, Internet of Things (IoT) sensors and other sources for many years. This volume of data amassed grows at staggering rates.

But harnessing value from this data has proved challenging for most. To counter these troubles, Microsoft has introduced Fabric and Copilot into the self-service business intelligence realms:

- **Fabric**, now in Preview, is an end-to-end, human-centred analytics product that brings together all of an organisation’s data and analytics in one place. It combines Power BI, Azure

Synapse and Azure Data Factory into one unified software as a service (SaaS) platform. Data engineers, data warehousing professionals, data scientists, data analysts and business users can seamlessly collaborate within Fabric to foster a well-functioning data culture across the organisation

- **Copilot in Power BI**, now in private Preview, is something we have detailed previously. It combines advanced generative AI with your data to help everyone uncover and share insights faster. The theory is that you merely describe the insights you need or ask a question about your data, and Copilot will analyse and pull the right data into a report converting data into actionable insights instantly.

Let’s take a look.

## Fabric

Generative AI and language model services, such as Azure OpenAI Service, are enabling customers to use and create everyday AI experiences that are reinventing how analysts spend their time. Powering organisation-specific AI experiences requires a constant supply of clean data from a well-managed and highly integrated analytics system. However, most organisations’ analytics systems struggle to integrate their specialised yet often disconnected services.

Microsoft Fabric is an end-to-end, unified analytics platform that brings together all the data and analytics tools that organizations need. Fabric

integrates technologies like Azure Data Factory, Azure Synapse Analytics and Power BI into a single unified product, empowering data and business professionals alike to unlock the potential of their data and lay the foundation for the era of AI.

Fabric is an end-to-end analytics product that addresses every aspect of an organisation’s analytics needs. There are five [5] areas that really Microsoft wants to emphasise. These are detailed below.

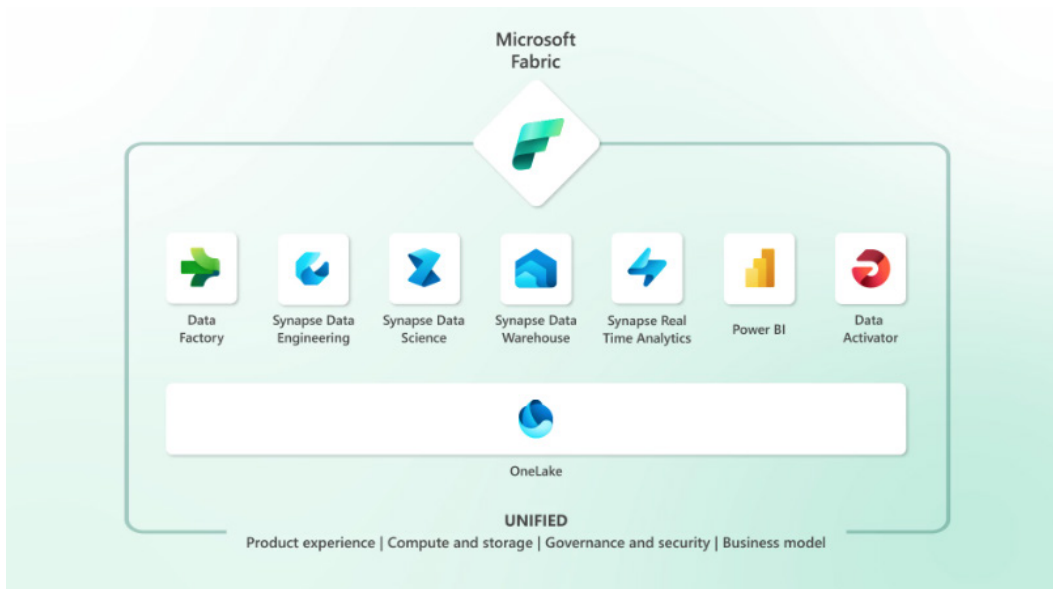
### 1. FABRIC IS A COMPLETE ANALYTICS PLATFORM

Every analytics project has multiple subsystems. Every subsystem needs a different array of capabilities, often requiring products from multiple vendors. Integrating these products can be a complex, fragile and expensive endeavour.

With Fabric, users may implement a single product with a unified experience and architecture that provides all the capabilities required for a developer to extract insights from data and present it to the

business user. By delivering the experience as software as a service (SaaS), everything is automatically integrated and optimized, and users can sign up within seconds and get real business value within minutes.

Fabric empowers every team in the analytics process with the role-specific experiences they need, so data engineers, data warehousing professionals, data scientists, data analysts and business users can get on with what they are paid to do.



Fabric comes with seven core workloads:

1. **Data Factory (Preview)** provides more than 150 connectors to cloud and on-premises data sources, drag-and-drop experiences for data transformation and the ability to orchestrate data pipelines
2. **Synapse Data Engineering (Preview)** enables better authoring experiences for Spark, instant start with live pools and the ability to collaborate

3. **Synapse Data Science (Preview)** provides an end-to-end workflow for data scientists to build sophisticated AI models, collaborate easily, and train, deploy and manage machine learning models
4. **Synapse Data Warehousing (Preview)** provides a converged lake house and data warehouse experience with industry-leading SQL performance on open data formats
5. **Synapse Real-Time Analytics (Preview)** enables developers to work with data streaming in from the Internet of Things (IoT) devices, telemetry, logs and more, and analyse large volumes of semi-structured data with high performance and low latency
6. **Power BI** in Fabric provides visualisation and AI-driven analytics that enable business analysts and business users to gain insights from data. The Power BI experience is also deeply integrated into Microsoft 365, providing relevant insights where business users already work
7. **Data Activator (coming soon)** provides real-time detection and monitoring of data and can trigger notifications and actions when it finds specified patterns in data all in a no-code experience.

You can try these experiences today by signing up for a Microsoft Fabric free trial.

## 2. FABRIC IS LAKE-CENTRIC AND OPEN

You don't need us to tell you that today's data lakes can be messy and complicated, making it hard for customers to create, integrate, manage and operate them. Once they are operational, multiple data products using different proprietary data formats on the same data lake can cause significant data duplication and concerns about vendor lock-in.

Fabric comes with a SaaS, multi-cloud data lake called OneLake that is built-in and automatically available to every Fabric tenant. All Fabric workloads are automatically wired into OneLake, just like all Microsoft 365 applications are wired into OneDrive. Data is organised in an intuitive data hub, and automatically indexed for discovery, sharing, governance and compliance.

OneLake serves developers, business analysts and business users alike, helping eliminate pervasive and chaotic data silos created by different developers provisioning and configuring their own isolated storage accounts. Instead, OneLake provides a single, unified storage system for all developers, where discovery and sharing of data are easy with policy and security settings enforced centrally. At the API layer, OneLake is built on and fully compatible with Azure Data Lake Storage Gen2 (ADLSg2), instantly tapping into ADLSg2's vast ecosystem of applications, tools and developers.

A key capability of OneLake is "Shortcuts". OneLake allows easy sharing of data between users and applications without having to move and duplicate information unnecessarily. Shortcuts allow OneLake to

virtualise data lake storage in ADLSg2, Amazon Simple Storage Service (Amazon S3), and Google Storage (coming soon), enabling developers to compose and analyse data across clouds.

Fabric is deeply committed to open data formats across all its workloads and tiers. It treats Delta on top of Parquet files as a native data format that is the default for all workloads. This deep commitment to a common open data format means that customers need to load the data into the lake only once and all the workloads can operate on the same data, without having to separately ingest it. It also means that OneLake supports structured data of any format and unstructured data, giving customers total flexibility.

By adopting OneLake as a store and Delta and Parquet as the common format for all workloads, Microsoft offers its customers a data stack that's unified at the most fundamental level. Customers do not need to maintain different copies of data for databases, data lakes, data warehousing, business intelligence or real-time analytics. Instead, a single copy of the data in OneLake can directly power all the workloads.

Managing data security (table, column and row levels) across different data engines can be a persistent nightmare for customers. Fabric will provide a universal security model that is managed in OneLake, and all engines enforce it uniformly as they process queries and jobs. This model is coming soon.

## 3. FABRIC IS POWERED BY AI

Microsoft has announced they are infusing Fabric with Azure OpenAI Service at every layer to help customers unlock the full potential of their data, enabling developers to leverage the power of generative AI against their data and assisting business users to find insights in their data. With Copilot in Microsoft Fabric in every data experience, users may use conversational language to create dataflows and data pipelines, generate code and entire functions, build machine learning models, or visualise results. Customers can even create their own conversational language experiences that combine Azure OpenAI Service models and their data and publish them as plug-ins.

Copilot (*see below*) in Microsoft Fabric builds on existing commitments to data security and privacy in the enterprise. Copilot inherits an organisation's security, compliance and privacy policies. Microsoft does not use organisations' tenant data to train the base language models that power Copilot.

Copilot in Microsoft Fabric will be coming soon. Stay tuned for the latest updates and public release date for Copilot in Microsoft Fabric.

## 4. FABRIC EMPOWERS EVERY BUSINESS USER

Customers aspire to drive a data culture where everyone in their organisation is making better decisions based on data. To help users foster this culture, Fabric integrates with the Microsoft 365 applications people use every day.

Power BI is a core part of Fabric and is already infused across Microsoft 365. Through Power BI's deep integrations with popular applications such as Excel, Microsoft Teams, PowerPoint and SharePoint, relevant data from OneLake is easily discoverable and accessible to users right from Microsoft 365 helping users drive more value from their data

With Fabric, you can turn your Microsoft 365 apps into hubs for uncovering

and applying insights. For example, users in Microsoft Excel can directly discover and analyse data in OneLake and generate a Power BI report with a click of a button. In Teams, users may infuse data into their everyday work with embedded channels, chat and meeting experiences. Business users can bring data into their presentations by embedding live Power BI reports directly in Microsoft PowerPoint. Power BI is also natively integrated with SharePoint, enabling sharing and dissemination of insights. With Microsoft Graph Data Connect (Preview), Microsoft 365 data is natively integrated into OneLake so analysts may unlock insights on their customer relationships, business processes, security and compliance, and people productivity.

## 5. FABRIC REDUCES COSTS THROUGH UNIFIED CAPACITIES

Today's analytics systems typically combine products from multiple vendors in a single project. This results in computing capacity provisioned in multiple systems like data integration, data engineering, data warehousing and business intelligence. When one of the systems is idle, its capacity cannot be used by another system causing significant wastage.

Purchasing and managing resources is massively simplified with Fabric. Users may purchase a single pool of computing that powers all Fabric workloads. With this all-inclusive approach, analysts can create solutions that leverage all workloads freely without any friction in their experience or commerce. The universal compute capacities significantly reduce costs, as any unused compute capacity in one workload can be utilised by any of the workloads.

## CURRENT MICROSOFT ANALYTICS SOLUTIONS

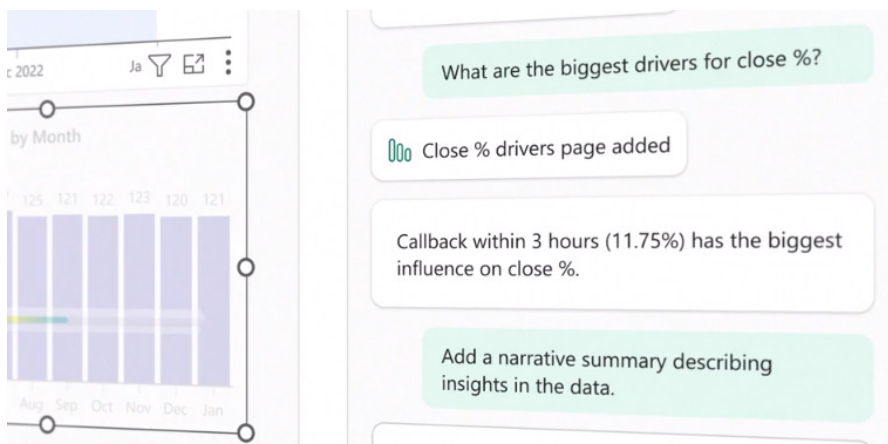
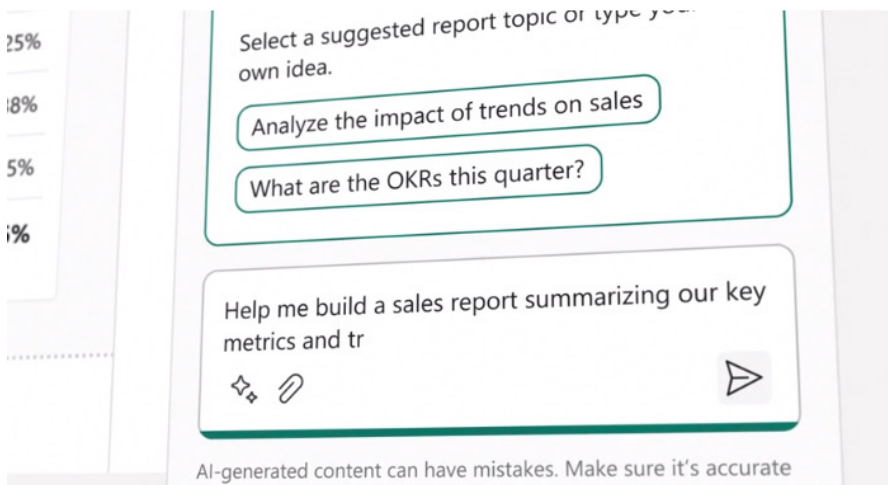
Existing Microsoft products such as Azure Synapse Analytics, Azure Data Factory and Azure Data Explorer will continue to provide a robust, enterprise-grade platform as a service (PaaS) solution for data analytics. Fabric represents an evolution of those offerings in the form of a simplified SaaS solution that can connect to existing PaaS offerings. Users will be able to upgrade from their current products into Fabric at their own pace.

Microsoft Fabric is currently in Preview. You may sign up for the free trial. Everyone who signs up gets a fixed Fabric trial capacity, which may be used for any feature or capability from integrating data to creating machine learning models. Existing Power BI Premium users can simply turn on Fabric through the Power BI administrator portal. After 1 July 2023, Fabric will be enabled for all Power BI tenants.

### Copilot in Power BI

With Copilot in Power BI, Microsoft is infusing the power of large language models into Power BI at every layer to help users get more done and create more value from their data. Using Copilot, you can simply describe the visuals and insights you're looking for and Copilot will then extrapolate. Users may create and tailor reports in seconds,

generate and edit DAX calculations, create narrative summaries, and ask questions about their data, all in conversational language. With the ability to easily tailor the tone, scope and style of narratives and add them seamlessly within reports, Power BI can also deliver data insights even more impactfully through easy-to-understand text summaries.



Microsoft has already released the quick measure suggestions for DAX capability that helps analysts quickly create the code they need. The rest of the Copilot in Power BI experiences are now in private Preview. Stay tuned to these Power BI Updates for the latest information.

Normal service should resume next month!



# New Features for Excel

This month, in Excel for web, the Chart Task Pane is now generally available, along with hyperlinks in Comments, and performance improvements related to conditional formatting – these performance

improvements are also available in Excel for Windows and Mac. For Insiders, the Quick Access Toolbar is now shown by default in Excel for Windows.

You can check out the whole list here:

## Excel for the web

- 'Chart Task' Pane
- Hyperlink in Comments
- Speed up filtering when cells contain unique or duplicate Conditional Formatting rules
- Reducing unwanted fragmenting of Conditional Formatting rules

## Excel for Windows

- Quick Access Toolbar (Insiders)
- Reducing unwanted fragmenting of Conditional Formatting rules (Current Channel)

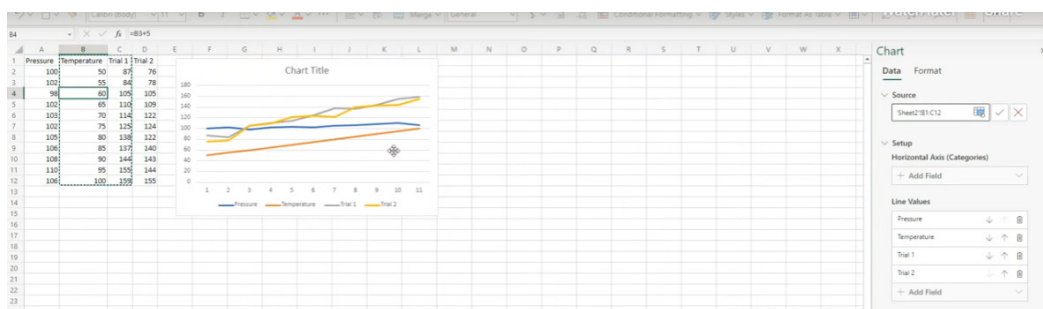
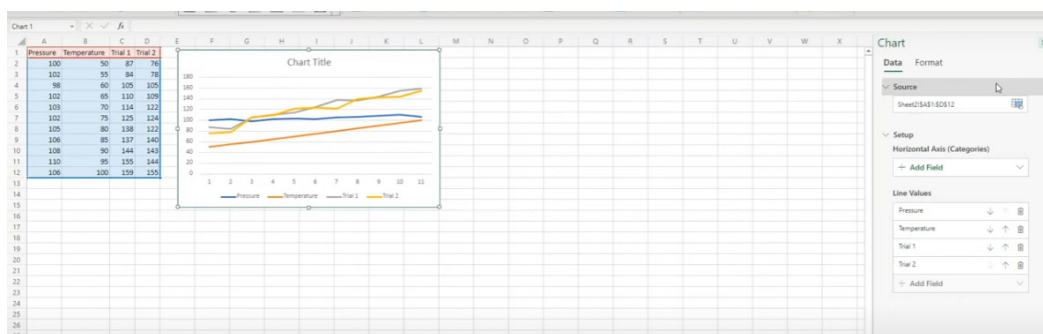
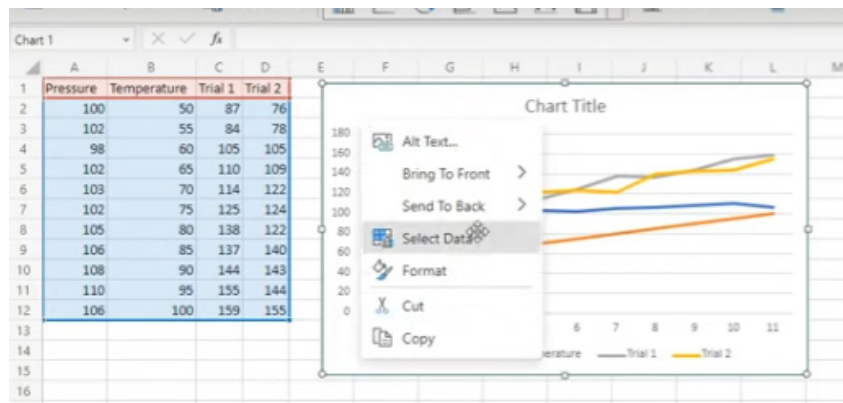
## Excel for Mac

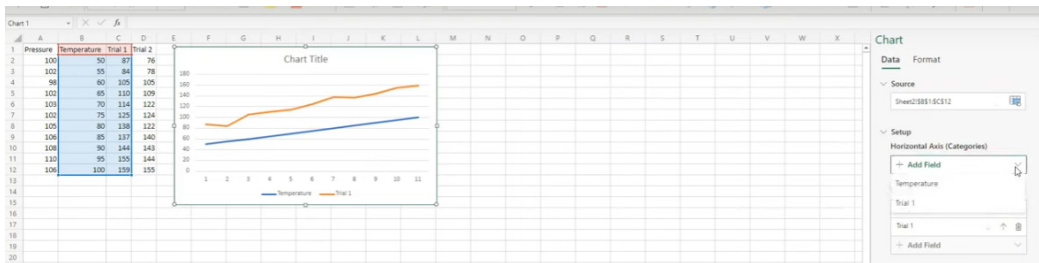
- Reducing unwanted fragmenting of Conditional Formatting rules.

Let's plough through. They are quite brief this month!

## 'Chart Task' Pane

In Excel for the Web, the 'Chart Task' pane now supports data editing such as modifying the chart's data range, removing a series or choosing which axis to plot your data on. You can find these options in the new Data tab alongside the existing Format tab. This feature is currently rolling out to Production.





### Hyperlink in Comments

Also in Excel for the Web, you can add hyperlinks in Comments to reference a source like a spreadsheet, document or webpage in conversation with others collaborating with you.

### Speed up filtering when cells contain unique or duplicate Conditional Formatting rules

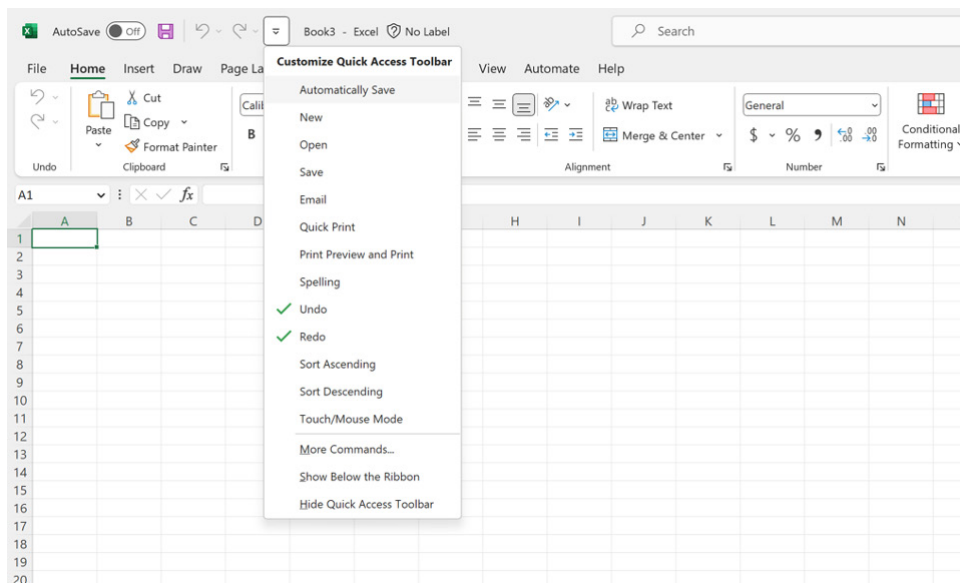
Excel for the Web has sped up filtering when cells contain unique or duplicate Conditional Formatting rules. This has been achieved by optimising the underlying comparison algorithm.

### Reducing unwanted fragmenting of Conditional Formatting rules

For Excel for the Web, the Current Channel of Excel for Windows and for Excel for Mac, when a workbook is opened this feature will merge those fragmented Conditional Formatting rules that are identical, within a contiguous range of cells, and with unchanged priority ordering. It excludes rules whose evaluation relies on a selection range like 'Above average' or 'Below average', Unique or Duplicate, Gradients, etc., and rules in PivotTables.

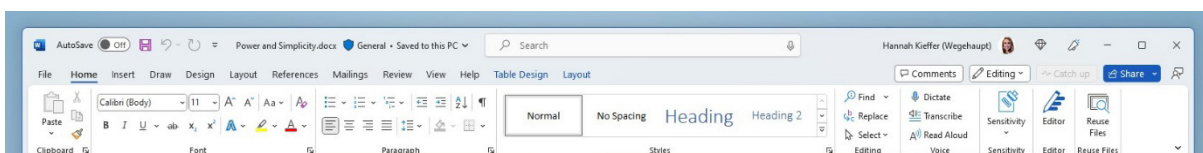
### Quick Access Toolbar

Finally, for the Insiders Channel of Excel for Windows, with this latest update, the Quick Access Toolbar (QAT) is now shown by default and includes the Undo, Redo and Save commands, and the AutoSave has found its rightful place back on the title bar. You can quickly customise the QAT to keep only the commands you need.

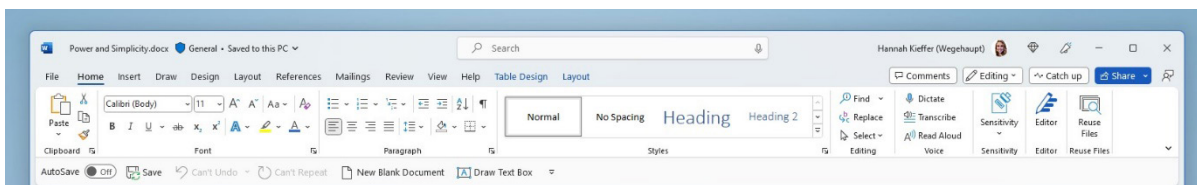


There are two different scenarios based upon your current experience:

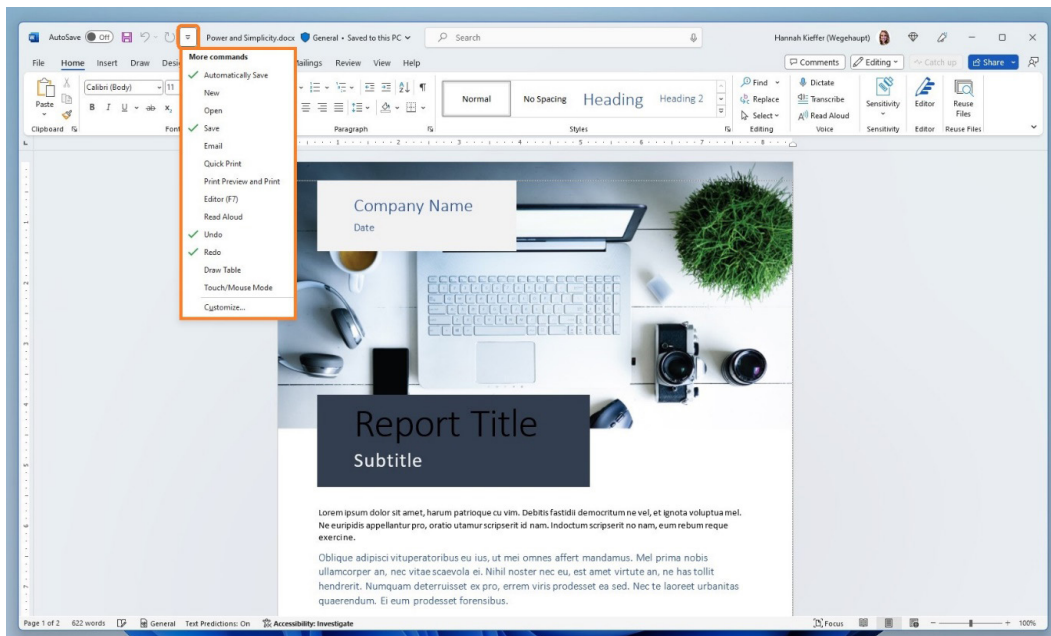
1. If your QAT is currently hidden, the QAT will be made visible above the Ribbon with the AutoSave, Save, Undo and Redo commands. You may customise this by right-clicking the QAT and selecting 'Customize'



- If your QAT is currently visible, the AutoSave, Save, Undo and Redo commands will be added to its left side. Custom commands and QAT location will be preserved. You can customise it by right-clicking the QAT and selecting Customise.



To add or remove commands, click the 'More commands' down arrow on the right side of the QAT, and then select 'Customize'.



This feature is available to Insiders running Version 2305 (Build 16403.20000) or later.

The updated version of the grid with all the new features is fast becoming too complicated to show clearly here. Nonetheless, you can find the interactive links at [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer).

### Excel Features Availability

Page 1 of 4

Feature	Insider		Production				Web
	Windows <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	Windows/CC <a href="#">Find the latest Excel version for this platform</a>	Windows/MEC <a href="#">Find the latest Excel version for this platform</a>	Windows/SA <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	
<a href="#">Block XLL Add-ins</a>	Version 2302 (Build 16130.20128) or later						
<a href="#">PivotTables: Manual Sort of Rows &amp; Columns</a>			Already Supported	Already Supported	Already Supported	Already Supported	February 2023
<a href="#">Automatic Recalculation Optimization</a>	Version 2208 (Build 15529.10000) or later	Version 16.64 (Build 22081401) or later					
<a href="#">Import Data from SQL Server Database</a>		Version 16.68 (Build 22110901) or later					
<a href="#">Import Data from Additional Sources</a>						Version 16.69 (Build 23010700) or later	
<a href="#">Power Query Editor</a>						Version 16.69 (Build 23010700) or later	
<a href="#">IMAGE function</a>			Version 2211 (Build 15831.20150) or later	Version 2211 (Build 15831.20252) or later			December 2022
<a href="#">Check Formulas with Value Preview Tooltips</a>	Version 2302 (Build 16116.20000) or later	Version 16.70 (Build 230116) or later					
<a href="#">Office Scripts</a>	Version 2212 (Build 15922.20000) or later	Version 16.68 (Build 22120101) or later					
<a href="#">Automate Tasks with Power Automate Tab</a>			Version 2301 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	
<a href="#">PivotTable Show Details to XLO</a>							January 2023
<a href="#">Excel Live in Teams</a>							December 2022
<a href="#">Formula Suggestions</a>							December 2022*
<a href="#">Formulas by Example</a>							December 2022*
<a href="#">Suggested Links</a>							December 2022
<a href="#">Add search bar in queries pane</a>							December 2022

\*Starting to roll out

Features Flyer: [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer)

CC: Current Channel, MEC: Monthly Enterprise Channel, SA: Semi Annual Enterprise Channel. All information is subject to change.



## Excel Features Availability

Page 2 of 4

Feature	Insider		Production				Web
	Windows <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	Windows/CC <a href="#">Find the latest Excel version for this platform</a>	Windows/MEC <a href="#">Find the latest Excel version for this platform</a>	Windows/SA <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	
Add keyboard shortcut to open PQ editor			Version 2211 (Build 15730.31883) or later				
Create nested PQ data types	Version 2211 (Build 15928.10000) or later						
Add Get Data from Dynamic Arrays	Version 2105 (Build 14014.20002) or later						
<a href="#">Data from picture</a>			Version 2210 (Build 15723) or later	Version 2210 (Build 15726.20262) or later			Version 16.38 or later
Chart Data Feils							November 2022
<a href="#">Show Changes</a>			Version 2209 (Build 15793.10000) or later				Version 16.66 (Build 22092500) or later
<a href="#">New Paste Options</a>	Version 2210 (Build 15726.20000) or later						
<a href="#">Quickly Find the Command you need</a>			Version 2206 (Build 15331.20010) or later				October 2022
<a href="#">New DAX Functions</a>	Version 2208 (Build 15504.10000) or later						
<a href="#">Navigation Pane</a>			Version 2209 (Build 15629.10000) or later				
<a href="#">Smooth Scrolling</a>			Version 2205 (Build 15225.20092) or later	Version 2208 (Build 15601.20230)	Version 2208 (Build 15601.20456) or later		Already Supported
<a href="#">Check Performance</a>							September 2022
<a href="#">Share Section of Excel Workbook</a>							September 2022
Dynamic Array Support in Charts	Version 2209 (Build 15617.10000) or later			Version 2210 (Build 15726.20262) or later			September 2022
Modern Comments			Version 2209 (Build 15427.20000) or later				
<a href="#">Manage Your Storage Accounts from Mac</a>		Version 16.64 (Build 22082100) or later					

Features Flyer: [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer)

CC: Current Channel; MEC: Monthly Enterprise Channel; SA: Semi-Annual Enterprise Channel  
All information is subject to change.

## Excel Features Availability

Page 3 of 4

Feature	Insider		Production				Web
	Windows <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	Windows/CC <a href="#">Find the latest Excel version for this platform</a>	Windows/MEC <a href="#">Find the latest Excel version for this platform</a>	Windows/SA <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	
<a href="#">New Excel functions</a>			Version 2208 (Build 15427.20194) or later	Version 2208 (Build 15601.20230) or later			August 2022
<a href="#">Power Query Group operations</a>							August 2022
<a href="#">Improvements to the connected Power BI experience</a>	Version 2208 (Build 15601.20028) or later						August 2022
Add and edit rich text formatting			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Sort by color or icon from auto filter menu			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy data connections			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy Shared Workbook feature			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Delete chart elements							August 2022
Multiline formula bar							August 2022
Search within PivotTable Field List			Already Supported	Already Supported	Already Supported	Already Supported	July 2022
<a href="#">Set automatic data conversions</a>	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20356) or later		Version 16.63 (Build 22070801) or later	
<a href="#">Resize Conditional Formatting dialog box</a>		Version 16.64 (Build 22070600) or later					
Sheet protection			Already Supported	Already Supported	Already Supported	Already Supported	June 2022

Features Flyer: [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer)

CC: Current Channel; MEC: Monthly Enterprise Channel; SA: Semi-Annual Enterprise Channel  
All information is subject to change.

## Excel Features Availability

Page 4 of 4

Feature	Insider		Production				Web
	Windows <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	Windows/CC <a href="#">Find the latest Excel version for this platform</a>	Windows/MEC <a href="#">Find the latest Excel version for this platform</a>	Windows/SA <a href="#">Find the latest Excel version for this platform</a>	Mac <a href="#">Find the latest Excel version for this platform</a>	
Semi-select for links creation			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Add "PivotTable Connections to Slicer settings" pane			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20298) or later	Version 2204 (Build 15128.20298) or later		Version 16.62 (22061100) or later	
<a href="#">Power Query refresh for selected data sources</a>			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Changing source file for workbook links			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
<a href="#">Improved Recommended PivotTable experience</a>	Version 2204 (Build 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20176) or later				
<a href="#">Improved Find dialog and Find All</a>						Version 16.60 (220410) or later	
<a href="#">LAMBDA Helper Functions</a>			Version 2202 (Build 14931.20120) or later	Version 2202 (Build 14931.20274) or later	Version 2208 (Build 15601.20456) or later	Version 16.56 (Build 211211) or later	Already Supported

Features Flyer: [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer)

CC: Current Channel; MEC: Monthly Enterprise Channel; SA: Semi-Annual Enterprise Channel  
All information is subject to change.

More next month, we're sure.

## The A to Z of Excel Functions: LEFT



The **LEFT** function returns the first character or characters in a text string, based upon the number of characters specified. It employs the following syntax to operate:

**LEFT(text, [number\_of\_characters])**

The **LEFT** function has the following arguments:

- **text**: this is required and represents the text string that contains the characters you want to extract
- **number\_of\_characters**: this argument is optional and specifies the number of characters you want **LEFT** to extract.

It should be further noted that:

- **number\_of\_characters** must be greater than or equal to zero
- if **number\_of\_characters** is greater than the length of text, **LEFT** returns all of **text**
- if **number\_of\_characters** is omitted, it is assumed to be one [1].

Please see our examples below:

	A	B	C
1	Data		
2	SumProduct		
3	Liam		
4	31 Dec 20		
5			
6			
7	Formula	Description	Result
8	<b>=LEFT(A2,3)</b>	First three characters in the string in cell A2	<b>Sum</b>
9	<b>=LEFT(A3)</b>	First character in the string in cell A3	<b>L</b>
10	<b>=LEFT(A4,2)</b>	First two characters in the string in cell A4. Note that this is a date and the underlying serial number is 44196	<b>44</b>

## The A to Z of Excel Functions: LEFTB



The **LEFTB** function returns the first character or characters in a text string, based on the number of bytes specified. It employs the following syntax to operate:

**LEFTB(text, [number\_of\_bytes])**

The **LEFTB** function has the following arguments:

- **text**: this is required and represents the text string that contains the characters you want to extract
- **number\_of\_bytes**: this argument is optional and specifies the number of characters you want **LEFTB** to extract, based upon bytes.

It should be further noted that:

- **number\_of\_bytes** must be greater than or equal to zero
- if **number\_of\_bytes** is omitted, it is assumed to be one [1]

this function may not be available in all languages. **LEFTB** counts two [2] bytes per character only when a DBCS language is set as the default language (the languages that support DBCS include Japanese, Chinese (Simplified), Chinese (Traditional), and Korean

otherwise, **LEFTB** behaves the same as **LEFT**, counting one [1] byte per character.

For example, =**LEFTB**("中国香港",2) is equal to "中". **LEFTB** returns the first character only, because each character is counted as two [2] bytes.

However, =**LEFT**("中国香港",2) is equal to "中国" as **LEFT** returns the first two [2] characters, because each character is counted as one [1]. **LEFT** returns the first two characters no matter what the default language setting is on your computer.

## The A to Z of Excel Functions: LEN



The **LEN** function returns the number of characters in a text string. It employs the following syntax to operate:

**LEN(text)**

The **LEN** function has the following argument:

- **text**: this is required and represents the text string whose length you wish to ascertain.

It should be further noted that:

- spaces count as characters in **text**.

Please see our examples below:

	A	B	C
1	Data		
2	SumProduct		
3		empty	
4	31 Dec 20		
5			
6			
7	Formula	Description	Result
8	=LEN(A2)	Length of text string in cell A2	10
9	=LEN(A3)	Length of text string in cell A3 (this is empty)	0
10	=LEN(A4)	Length of text string in cell A4 (date is a serial number, 44196)	5

## The A to Z of Excel Functions: LENB

	A	B
1	Text	Result
2	How	3
3	How to	6
4	How to Use LENB Function	24

The **LENB** function returns the number of bytes used to represent the characters in a text string. It employs the following syntax to operate:

**LENB(text)**

The **LENB** function has the following argument:

- **text**: this is required and represents the text string whose length you wish to ascertain.

It should be further noted that:

- spaces count as characters in **text**
- this function may not be available in all languages. **LENB** counts two [2] bytes per character only when a DBCS language is set as the default language (the languages that support DBCS include Japanese, Chinese (Simplified), Chinese (Traditional), and Korean)
- otherwise, **LENB** behaves the same as **LEN**, counting one [1] byte per character.

For example, **=LENB("中国香港")** is equal to 8 only when a DBCS language is set as the default language; otherwise, it will return 4. However, **=LEN("中国香港")** is equal to 4 no matter what the default language setting is on your computer.

More Excel Functions next month.

## Beat the Boredom Suggested Solution

Our challenge this month had several mini-challenges, the last min-challenge being potentially the largest obstacle of all.

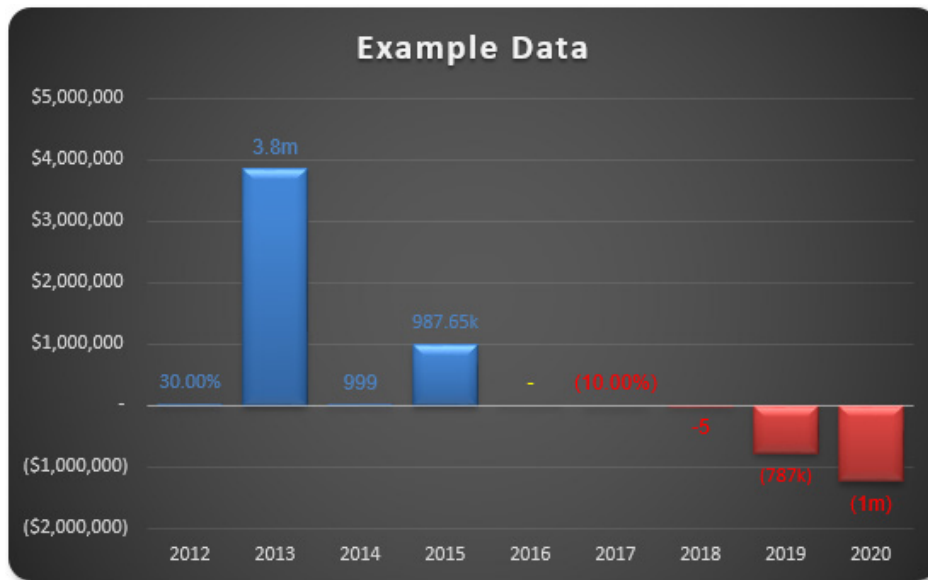
### *The Challenge*

You were asked to consider the following pre-formatted chart data (all of the following inputs are numbers typed in, not text):

### Chart Data

Year	Value
2012	30.00%
2013	3.8m
2014	999
2015	987.65k
2016	-
2017	(10.00%)
2018	-5
2019	(787k)
2020	(1m)

The first challenge was to get the formatting correct! And then, "all" you had to do was create the associated Excel chart with replicated label formatting:



But it was harder than that: this chart has to work after you have closed the file and reopened...

### Suggested Solution

As mentioned above, there were several mini-challenges:

1. Understand how you can create multiple number formats in Excel, never mind in a chart data label
2. Create the basic chart
3. Create the formatting in the data labels, realising custom number formatting will not work and conditional formatting may not be applied
4. Make the solution robust enough to cope with saving the file and re-opening.

I will address all four areas below.

### Mini-Challenge 1: Creating Multiple Number Formats in Excel

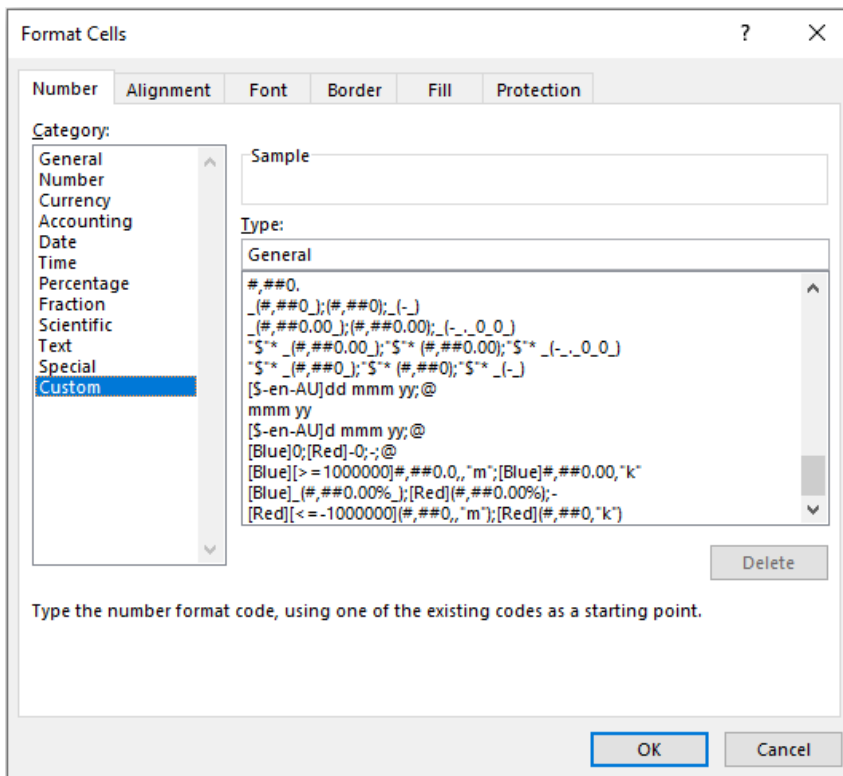
This is more a challenge of understanding how it works, rather than trying to solve the problem from scratch. Returning to our problem our data appears as follows:

#### Chart Data

Year	Value
2012	30.00%
2013	3.8m
2014	999
2015	987.65k
2016	-
2017	(10.00%)
2018	-5
2019	(787k)
2020	(1m)

Here, the correct custom formatting has been generated for numbers simply typed in, without using VBA code or Excel formulae.

Given that one of the primary purposes of a spreadsheet is to present numerical data, it is important how numerical data is presented. Cells may be individually formatted, using **CTRL + 1** or **ALT + O + E** in all versions of Excel:



Formatting only changes the appearance, not the underlying value, of a cell. For example, if cells **A1** and **B1** had the number '1.4' typed in but were formatted to zero decimal places, then if cell **C1 = A1 + B1**, you would truly have  $1 + 1 = 3$  (well,  $1.4 + 1.4 = 2.8$  anyway).

Excel has many built-in number formats that are fairly easy to understand, e.g. Currency, Date, Percentage. Selecting the 'Custom' category activates the 'Type' input box and allows between 200 and 250

custom number formats in a particular workbook, depending upon the language version of Excel that has been installed.

The 'Type' input box allows up to four aspects of formatting to be specified in a cell. These aspects are referred to as sections and are separated by a semi-colon (;). To ascertain what is contained in each section depends on the total number of sections used, viz.

No. of Sections	Section Details (assuming no conditions)
1 (min)	All numerical values
2	Non-negative Numbers; Negative Numbers
3	Positive Numbers; Negative Numbers; Zero Values
4 (max)	Positive Numbers; Negative Numbers; Zero Values; Text

Notice only four options are available: what to do if positive, negative, zero or text.

I have written a [previous article](#) that discussed how to construct the following:

Format	Cell Value	Appearance
[Blue]\$*_(#,##0.0_0_0);[Red]\$*_(#,##0.00);[Color 7]-_0_0_0;[Cyan]@*.'is text'	123456789 \$	123,456.8
	-123456789 \$	(123,456.79)
	0	-
This is text	This is text.....	is text

There is an alternative syntax to the above four arguments:

No. of Sections	Section Details (assuming conditions)
1 (min)	All numerical values
2	Numbers meeting criterion; All other values
3 (max)	Numbers meeting first criterion; Numbers meeting second criterion which do not meet the first criterion; All other values

In the same [past article](#), this was demonstrated as follows:

Format	[>=1000000]#,##0,,"M";[>=1000]#,##0,"K";0	
	Number	Appearance
	0.1	0
	1	1
	10	10
	100	100
	1,000	1K
	10,000	10K
	100,000	100K
	1,000,000	1M
	10,000,000	10M
	100,000,000	100M
	1,000,000,000	1,000M

The conditions are included in square brackets such that if the condition is true, the following formatting will be applied.

In this example, there are only three sections, so text will be formatted 'generally'. The first section, `[>=1000000]#,##0,,"M"`, will format all numbers greater than or equal to a million to the nearest million and add an "M" to the end of the number.

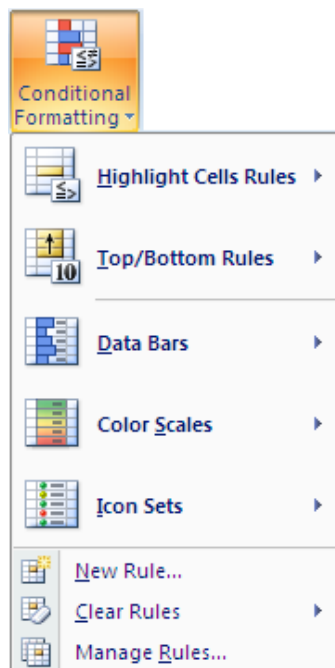
The second section will only be considered if the first condition is not true, so the order of the two 'conditional formats' needs to be thought through. Here, the second section, `[>=1000]#,##0,"K"`, will format all

numbers greater than or equal to a thousand (but necessarily less than a million) to the nearest thousand and add a "K" to the end of the number.

The third and final section, `0`, will format all other numbers (every value less than 1,000) to the nearest integer without thousands separator(s).

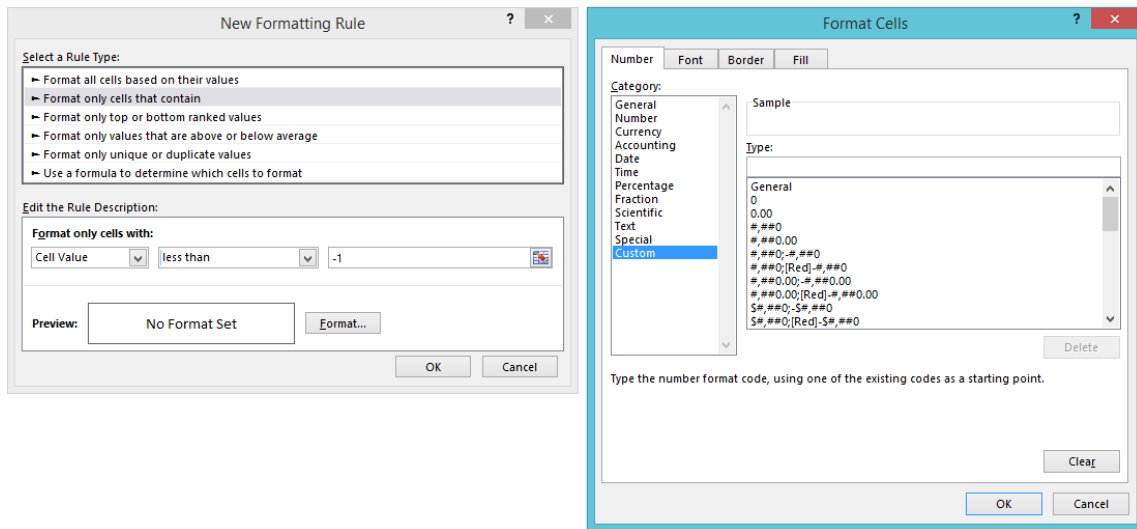
This example is *almost* what we require; but it only allows three scenarios. We have many more.

The trick here is to resort to [conditional formatting](#). Located in the Styles group of the Home tab, the conditional formatting feature allows you to consider multiple conditions:





For instance, inspecting 'Highlight Cells Rules' is akin to many of the "Cell Value Is" functionalities of its predecessor, e.g. Greater Than, Less Than, Between, Equal To. I can use this to exploit a loophole in the restrictive number of conditions custom number formatting appears to allow. For example:



In this above illustration, I have selected conditional formatting to occur if the value in the cell is less than -1. Pressing the 'Format...' button then allows the user to select how the number formatting might appear.

Returning to our data:

### Chart Data

Year	Value
2012	30.00%
2013	3.8m
2014	999
2015	987.65k
2016	-
2017	(10.00%)
2018	-5
2019	(787k)
2020	(1m)

Using both functionalities, the requirement is fairly straightforward. My suggested solution would be:

- Construct the underlying number formatting first. Personally, I would use custom number formatting so that all positive and negative numbers appeared as percentages, zero as a hyphen ("-") and text as displayed above
- Next, I would apply conditional formatting number formatting where the cell value is greater than one so that numbers greater than a million could be displayed to the nearest 0.1m, numbers less than a million but greater than or equal to 1,000 could be displayed to the nearest 0.00k and numbers lower than 1,000 (but necessarily greater than one) could be displayed as integers
- Finally, I would apply a second set of conditional formatting where the cell value is less than -1 as required.

Since many, many conditional formats may be applied to one cell in Excel, you can soon apply significantly more than four formats to any cell(s) in Excel.

### Mini-Challenge 2: Create the Chart

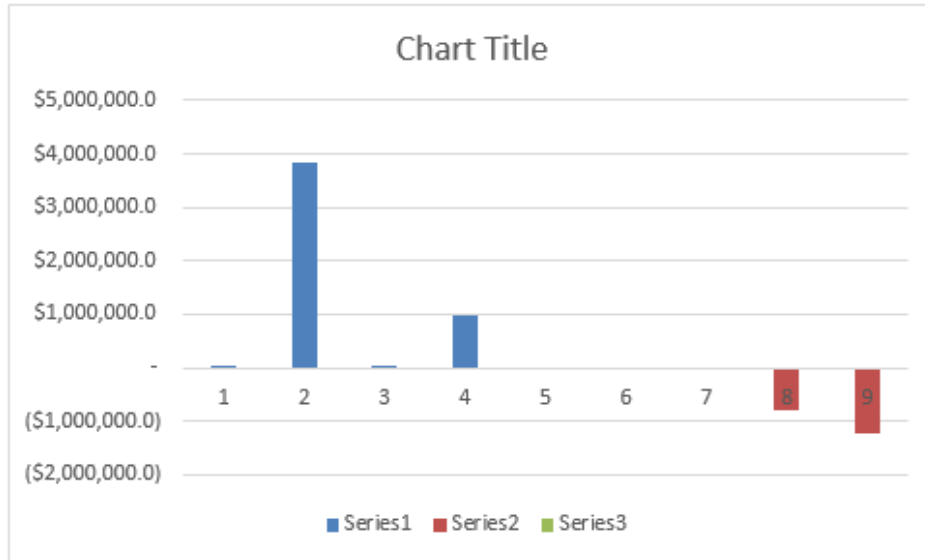
This isn't so hard, once you realise that if you want your data labels to be three different colours (blue for positives, red for negatives and yellow for zeros) and you want your columns to be different too (blue for positives, red for negatives, blank for zeros), you will need to separate your data into positives, negatives and zeros:

Chart Data		Positives	Negatives	Zeros
2012	30.00%	\$0.3	#N/A	#N/A
2013	3.8m	\$3,845,454.0	#N/A	#N/A
2014	999	\$999.0	#N/A	#N/A
2015	987.65k	\$987,654.0	#N/A	#N/A
2016	-	#N/A	#N/A	-
2017	(10.00%)	#N/A	(\$0.1)	#N/A
2018	-5	#N/A	(\$5.0)	#N/A
2019	(787k)	#N/A	(\$787,452.0)	#N/A
2020	(1m)	#N/A	(\$1,234,567.0)	#N/A

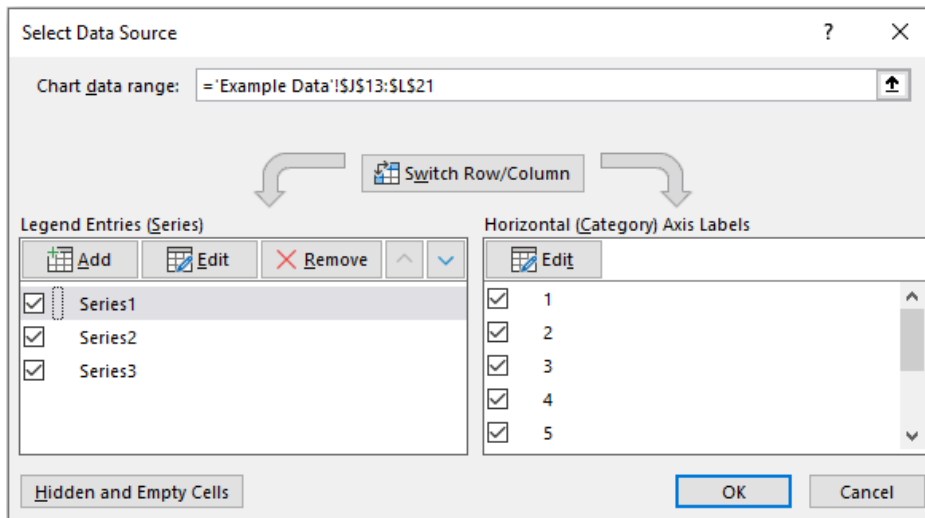
Here, I have added a “helper” table:

- Positives: **J13** contains the formula `=IF($G13>0,$G13,NA())`. This puts the corresponding number in this cell if it is positive, else #N/A is returned. #N/A is different to zero, in that charts (depending upon which chart type is selected) will usually ignore these values (zeros are “noticed”)
- Negatives: **K13** contains the formula `=IF($G13<0,$G13,NA())`, which returns only the negative values
- Zeros: **L13** contains the formula `=IF($G13=0,$G13,NA())`, which returns only the zero values.

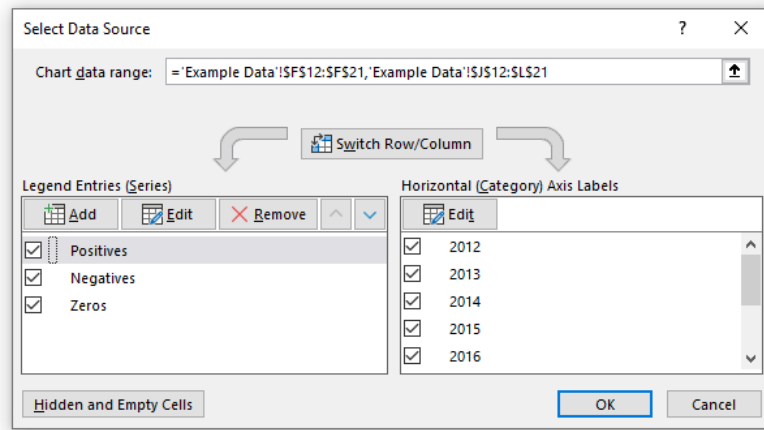
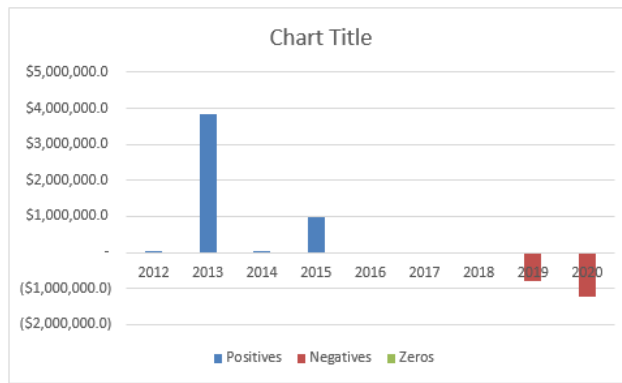
If you select cells **J13:L21** and insert a Stacked Column Chart (**ALT + N + C1**, then choose Stacked Column Chart), you will create a chart as follows:



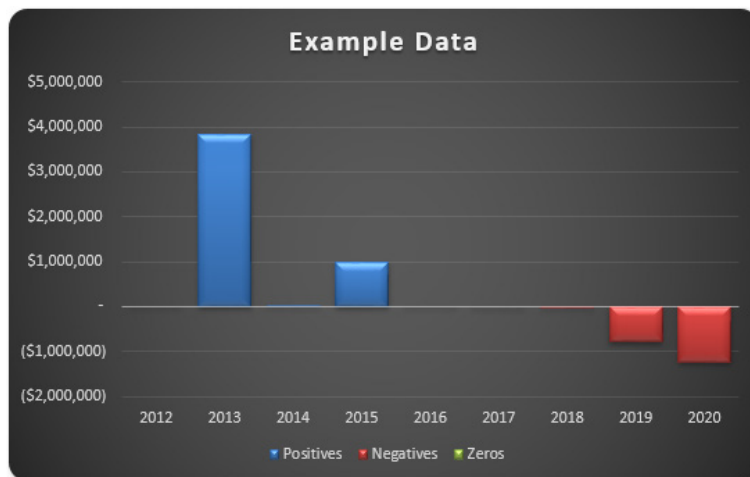
Right-clicking on the chart and selecting ‘Select Data...’ from the shortcut menu summons the ‘Select Data Source’ dialog, viz.



By editing the Legend Entries (Series) and the Horizontal (Category) Axis Labels, I can replace ‘Series1’, ‘Series2’ and ‘Series3’ with ‘Positives’, ‘Negatives’ and ‘Zeros’ respectively, as well as swap out the integer counters for the Year:

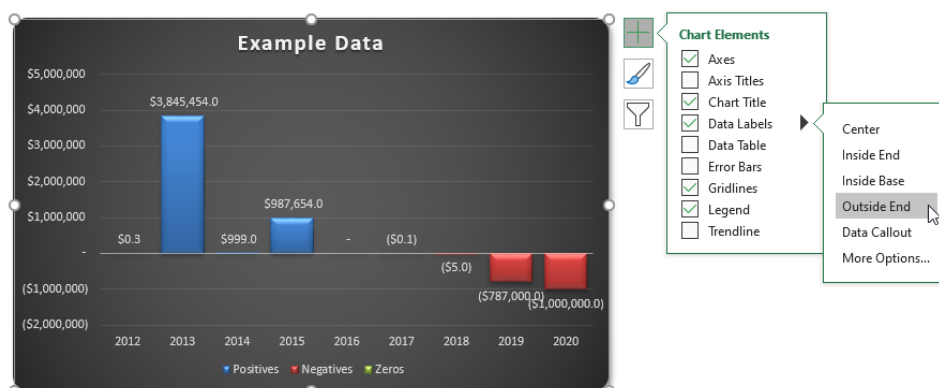


A little formatting here and there (e.g. reducing the columns' gap widths, changing columns, adding a title) works wonders:



It appears we are nearly there. All we need to do is add the data labels. What could be simpler? Well, understanding Schwarzschild's solution to Einstein's field equations on general relativity, for one.

Adding data labels is simple enough, by selecting the chart and adding chart elements (clicking on the big plus [+] button) viz

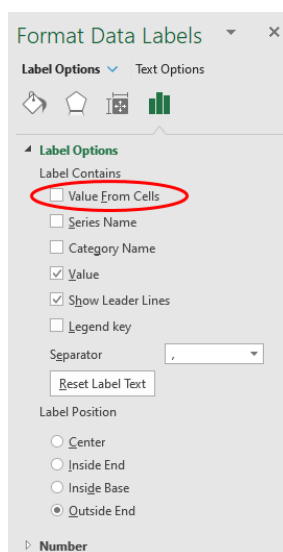


The data labels may be aligned outside end (above and below the non-negative and negative columns respectively), but what about formatting? Given we have three series, having blue for positive labels, red for negative labels and yellow for zeros is straightforward. However,

how do you mirror the number formatting for positives and negatives when each has more scenarios than custom number formatting can deal with? You cannot use conditional formatting with data labels, so our earlier trick is redundant here.

### Mini-Challenge 3: Creating Data Label Formatting

The data labels cannot be created using custom number formatting. I will have to get more creative. If you select the data labels and then format the selection, the 'Format Data Labels' pane will appear:



We need to change the label contents from Value to 'Value From Cells'. This means we can select a different cell – such as text – associated with these values. Then, it's easy.

Therefore, I create a second "helper" table:

N13    =IF(ISNA(J13),".",TEXT(J13,IF(J13>=10^6,"#,##0.0,\m",IF(J13>=10^3,"#,##0.00,\k",IF(J13<1,"0.00%","0")))))

Chart Data		Positives	Negatives	Zeros	Positives	Negatives	Zeros
2012	30.00%	\$0.3	#N/A	#N/A	30.00%	.	.
2013	3.8m	\$3,845,454.0	#N/A	#N/A	3.8m	.	.
2014	999	\$999.0	#N/A	#N/A	999	.	.
2015	987.65k	\$987,654.0	#N/A	#N/A	987.65k	.	.
2016	-	#N/A	#N/A	#N/A	.	.	.
2017	(10.00%)	#N/A	(\$0.1)	#N/A	.	(10.00%)	.
2018	-5	#N/A	(\$5.0)	#N/A	.	-5	.
2019	(787k)	#N/A	(\$787,000.0)	#N/A	.	(787k)	.
2020	(1m)	#N/A	(\$1,000,000.0)	#N/A	.	(1m)	.

These three added fields create text strings that mimic the number formatting using Excel's TEXT function. In its simplest form, TEXT has the following form:

**=TEXT(value to be formatted, "Format required")**

The formatting in inverted commas (speech marks) follows the same syntax as custom number formatting, except you cannot use speech marks for text (a backslash should be used instead). Therefore, the three formulae added are:

- The formula for positives (e.g. cell J13) is given by **=IF(ISNA(J13),".",TEXT(J13,IF(J13>=10^6,"#,##0.0,\m",IF(J13>=10^3,"#,##0.00,\k",IF(J13<1,"0.00%","0")))))**  
This formula checks to see whether the number is positive (IF(ISNA(J13), ...)) – if there is an #N/A error in cell J13 then the number is not positive and no further formatting is required. When this happens a period (.) is displayed. The character is subjective; I use it as it is difficult to see and text labels misbehave if they are allowed to be completely empty or just contain a space.  
The rest of the formula uses TEXT to format the positive number appropriately in millions, thousands, percentages or just as an integer
- The formula for negatives (e.g. cell K13) is given by **=IF(ISNA(K13),".",TEXT(-K13,IF(-K13>=10^6,"(#,##0,\m)",IF(-K13>=10^3,"(#,##0,\k)",IF(-K13<1,"(0.00%)","-0")))))**  
This formula checks to see whether the number is negative (IF(ISNA(K13), ...)). The rest of the formula uses TEXT to format the negative number appropriately in millions, thousands, percentages or just as an integer
- The formula for everything else – which should be zeros (e.g. cell L13) is given by **=IF(ISNA(L13),".",IF(\$G13=0,"-","NA()))**  
This is much simpler, but just allows for any possible text that has crept through.

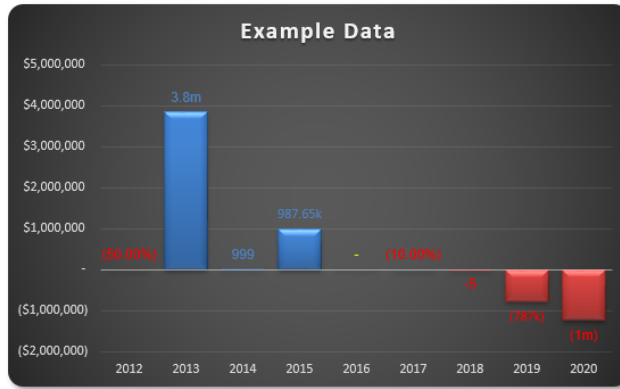
If the 'Value From Cells' selects these ranges rather than the 'Values', the chart will be generated just as we require (having placed the chart over my "helper" tables):

If the 'Value From Cells' selects these ranges rather than the 'Values', the chart will be generated just as we require (having placed the chart over my "helper" tables):

**Values**

**Chart Data**

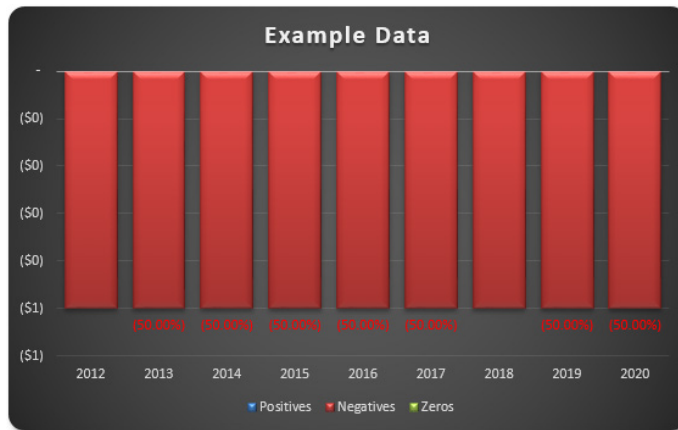
Year	Value
2012	(50.00%)
2013	3.8m
2014	999
2015	987.65k
2016	-
2017	(10.00%)
2018	-5
2019	(787k)
2020	(1m)



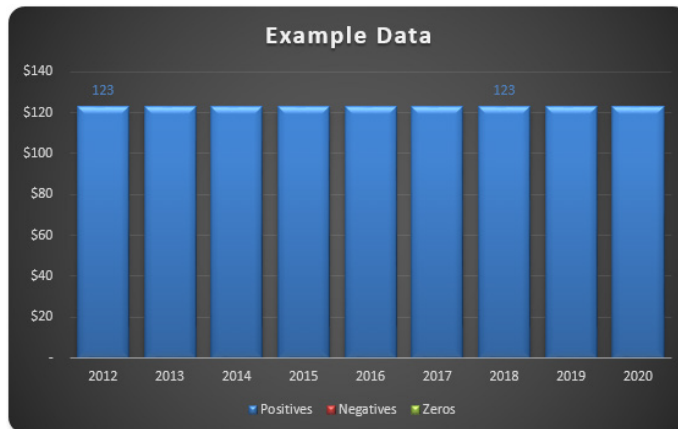
We're finished!

**Mini-Challenge 4: Make the Solution Robust**

No we're not. Try closing and saving the file, then re-open. If you make all of the input values (50.00%), you will probably see something similar to the following chart:



You may not be missing the same labels, but you will probably be missing *some*. Similarly, if I type '123' into all the inputs I may get:



Typically, the labels missing the first time will appear here, and *vice versa*. Not good!

This is *soooo* frustrating! This is a **bug** in Excel. It is still prevalent in Excel 2019 *etc.* and the latest versions of Microsoft 365. A quick consultation with your favourite search engine tends to lead you to ideas such as saving the chart as a chart template, and create a macro on opening that will re-apply the chart. Thank you very much; that's not for me.

It's not often on these challenges that we cheat quite like I will be doing this time. For this problem, there is a common free, third party add-in

that appears to come to the rescue, namely past Excel MVP Rob Bovey's [XY Chart Labeler \(sic\)](#).

If your IT administrator will allow you, downloading this add-in allows you to add 'XY chart labels' – and these seem to stick. It's easy to use – all you have to do is add the chart labels and reference the text values using the add-in rather than Excel's tools.

Until next month.

## Upcoming SumProduct Training Courses - COVID-19 update

Due to the COVID-19 pandemic that is currently spreading around the globe, we are suspending our in-person courses until further notice. However, to accommodate the new working-from-home dynamic, we are switching our public and in-house courses to an online delivery stream, presented via Microsoft Teams, with a live presenter running through the same course material, downloadable workbooks to complete the hands-on exercises during the training session, and a recording of the sessions for

your use within 1 month for you to refer back to in the event of technical difficulties. To assist with the pacing and flow of the course, we will also have a moderator who will help answer questions during the course.

If you're still not sure how this will work, please contact us at [training@sumproduct.com](mailto:training@sumproduct.com) and we'll be happy to walk you through the process.

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	7 - 9 Jun 2023	09:00-17:00 AEST	(-1 day) 23:00-17:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	14 Jun 2023	09:00-17:00 AEST	(-1 day) 23:00-17:00 GMT	1 Day
Online (Australia)	Financial Modelling	15 - 16 Jun 2023	09:00-17:00 AEST	(-1 day) 23:00-17:00 GMT	2 Days

## Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we look at **ALT** plus letters. Even less exciting, there is only one active letter since the retirement of Excel 2003 and earlier versions (not much to revise this month!):

Keystroke	What it does
ALT + F	Go to Backstage area

There are c.550 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file at [www.sumproduct.com/thought/keyboard-shortcuts](http://www.sumproduct.com/thought/keyboard-shortcuts). Also, check out our new daily **Excel Tip of the Day** feature on the [www.sumproduct.com](http://www.sumproduct.com) homepage.

## Our Services

We have undertaken a vast array of assignments over the years, including:

- **Business planning**
- **Building three-way integrated financial statement projections**
- **Independent expert reviews**
- **Key driver analysis**
- **Model reviews / audits for internal and external purposes**
- **M&A work**
- **Model scoping**
- **Power BI, Power Query & Power Pivot**
- **Project finance**
- **Real options analysis**
- **Refinancing / restructuring**
- **Strategic modelling**
- **Valuations**
- **Working capital management**

If you require modelling assistance of any kind, please do not hesitate to contact us at [contact@sumproduct.com](mailto:contact@sumproduct.com).

## Link to Others

These newsletters are not intended to be closely guarded secrets. Please feel free to forward this newsletter to anyone you think might be interested in converting to "the SumProduct way".

If you have received a forwarded newsletter and would like to receive future editions automatically, please subscribe by completing our newsletter registration process found at the foot of any [www.sumproduct.com](http://www.sumproduct.com) web page.

## Any Questions?

If you have any tips, comments or queries for future newsletters, we'd be delighted to hear from you. Please drop us a line at [newsletter@sumproduct.com](mailto:newsletter@sumproduct.com).

## Training

SumProduct offers a wide range of training courses, aimed at finance professionals and budding Excel experts. Courses include Excel Tricks & Tips, Financial Modelling 101, Introduction to Forecasting and M&A Modelling.

Check out our more popular courses in our training brochure:



Drop us a line at [training@sumproduct.com](mailto:training@sumproduct.com) for a copy of the brochure or download it directly from [www.sumproduct.com/training](http://www.sumproduct.com/training).