

Sum Product

NEWSLETTER #136 - March 2024

www.sumproduct.com | www.sumproduct.com/thought

It's more than meets the AI this month

as we announce some great news about Copilot and have more on ChatGPT in this month's newsletter than may first appear. I even make some interesting conclusions about Yours Truly (it seems to be a rather bizarre wish list!).

But of course, that's not all.

We take a more detailed look at eta lambdas too, and then use all the usual topics to keep it company. We have our latest Beat the Boredom Challenge, together with the usual tips on Charts & Dashboards, Excel for Mac, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates and Excel Updates. It's also "T" time for our A to Z of Keyboard Shortcuts and we have Excel functions *à la* **MODE**.

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

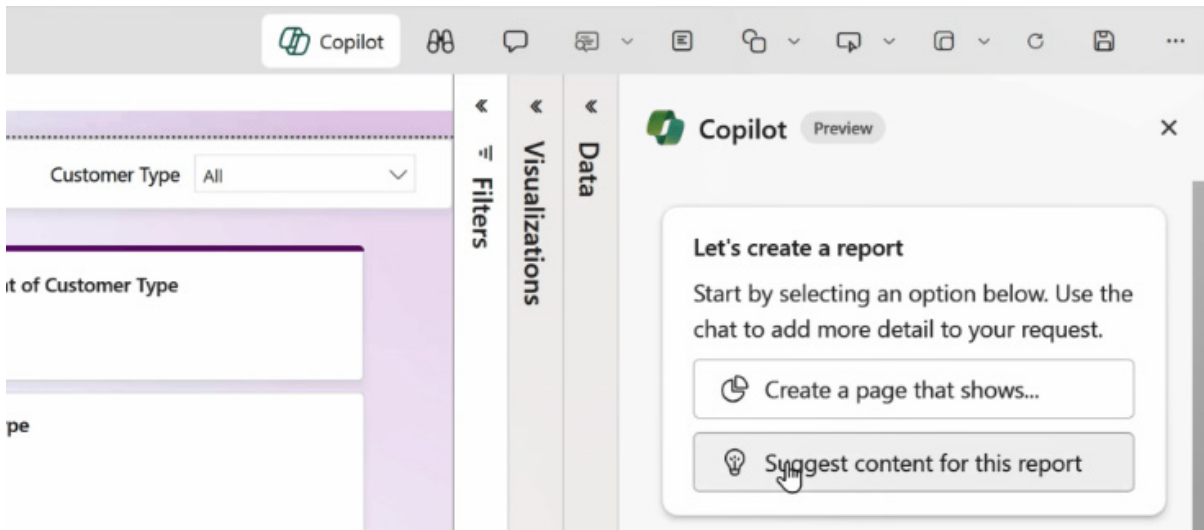
Liam Bastick, Managing Director, SumProduct

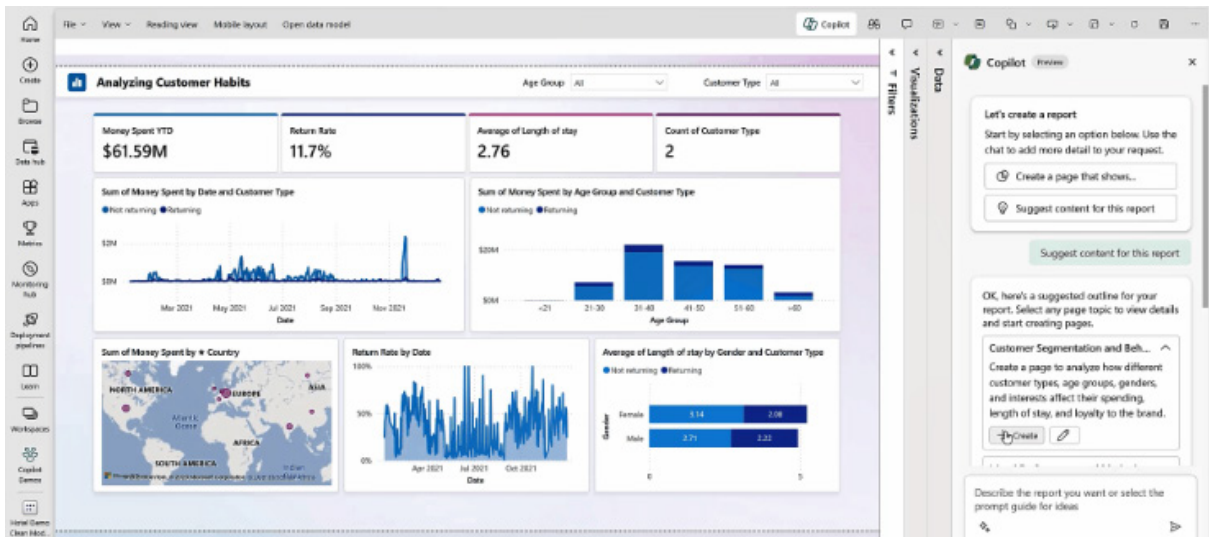


Copilot in Power BI (Preview) is Now Available Worldwide

With Microsoft's current Preview, users can create reports faster and easier in the Power BI web experience. Based upon a high-level prompt, Copilot for Power BI in Fabric creates an entire report page for you by identifying the tables, fields, measures and charts that would help you get started. You can then customise the page using our existing editing

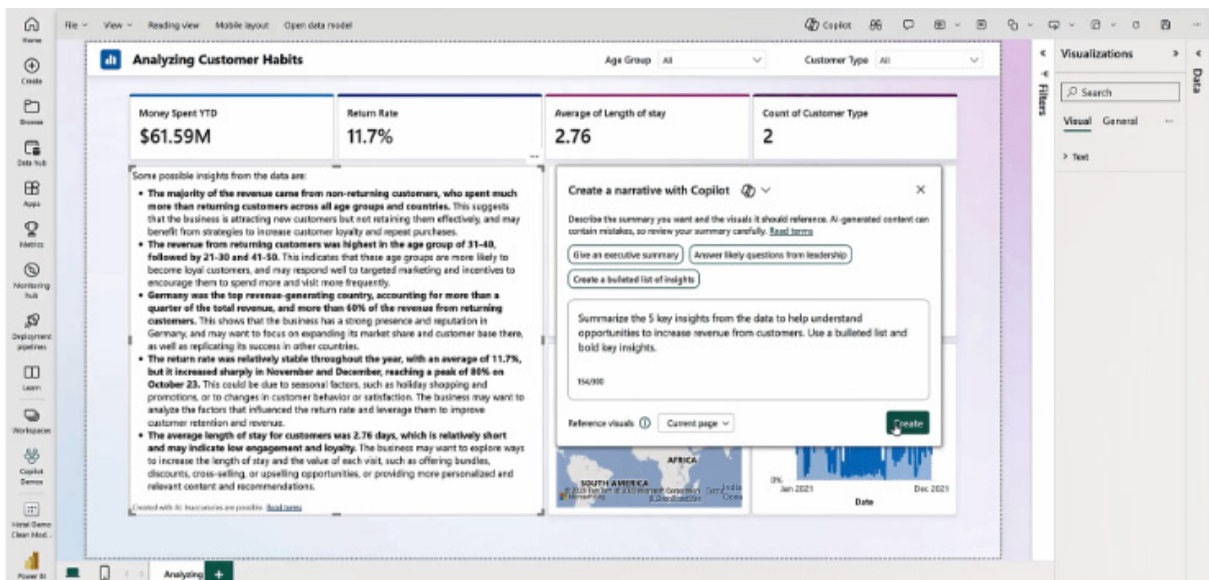
experiences. Copilot can also help you understand your semantic model and even suggest topics for your report pages. It's a fast and easy way to get started with a report, especially if you're less familiar with report creation in Power BI.





In addition to report creation, Copilot also has the ability to summarise data to the 'Smart Narrative' visual, now rebranded as the 'Narrative with Copilot' visual. This visual summarises the data and insights on the

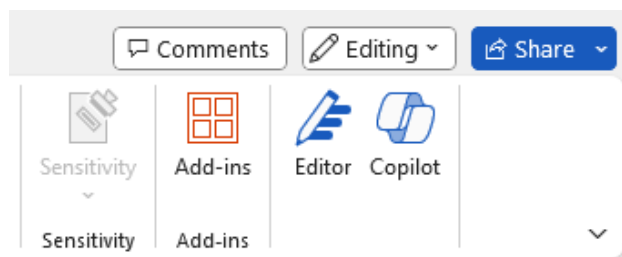
page, across your report, or even for your own template if you need to define a specific summary. The visual is available in the Power BI service and in Power BI Desktop.



Before you can use Copilot capabilities in Power BI, your administrator needs to enable Copilot in Microsoft Fabric. This can be done with a new tenant setting group, 'Copilot and Azure OpenAI Service (preview)', in the admin portal.

Since December 2023, Microsoft has shipped a new feature that allows tenant administrators to enable AI and Copilot in Power BI for specific security groups in addition to the entire organisation. This means that Copilot can now be tailored to the needs of different groups within your organisation. More granular AI and Copilot setting on Capacity level is coming.

Fabric Copilot is powered by Azure Open AI large language models and are currently deployed to limited data centres. As of January 2024, if your data is located outside of the US or France, the Copilot preview will be disabled by default unless your Tenant administrator enables the 'Data sent to Azure OpenAI can be processed outside your tenant's geographic region, compliance boundary, or national cloud instance' tenant setting.



Now that Copilot has been opened up to all, we've been having a bit of a play with it – and we'd love to hear about your experiences too. So far, we think it's been most useful in Word (for drafting articles and summaries), Power BI (Smart Narratives) and Teams (for annotation, summarising and transcribing). Excel hasn't been a great playground yet as it wants data in Tables and appears to spend six months crafting its summaries where 'Analyze Data' appears a lot faster.

It's not stopped us having some fun though. We (read "I") have been embarking on some vanity projects to see what Copilot knows about SumProduct and Yours Truly. The results have been *interesting*. For example:

- SumProduct is a US\$10m firm doubling in size every five years soon to be listed on the US stock exchange
- Our company has over 500 employees, each of whom receives a 10% share of profits (!)
- Liam Bastick is a born-and-bred Aussie (nope) who studied Economics at the University of Melbourne. Do feel free to ask me technical questions on the theory of economics, just don't expect any answers
- Liam Bastick is presently under consideration for a knighthood from the Queen (?) for services to financial modeling (*sic*). Maybe it's the rock group Queen?
- He is also known as "Excel Legend", a title bestowed on him by what appears to be the entirely-fictional Excel User Group.

Please don't let such bizarre results put you off playing with Copilot, ChatGPT *et al*, but do remember the importance of fact checking. We invite readers to send in their own "Copilot Facts" should they wish – the funniest we will publish here. Send them to contact@sumproduct.com and mark the subject "Copilot Facts".

In the meantime, bow down in front of me you ungrateful curs.

Sir Liam Bastick, Excel Legend, bankrupt owner of SumProduct 😊

Revisiting Eta Lambdas

We have had several queries about the so-called **eta lambda** functions. We have a part-time employed theory (in case it's not working full-time!).

As a reminder, the eta lambda functions are those that work where you are reducing just one argument of the function in question or else it is "obvious" what should happen. Therefore, **SUM(range)** would work as **SUM** but you would find without an interim step **SUM(range1, range2)** would not. This is because the eta lambda "function" appears to be a shortcut for

=LAMBDA(x, function(x))

amongst other calculations. "Obvious" alternatives can be such as

=SCAN("", A1:A3, LAMBDA(a, b, CONCAT(a, b)))

becoming

=SCAN("", A1:13, CONCAT)

(thanks to **John MacDougall** MVP for pointing that out).

Therefore, they will work whenever this rule appears to be allowed and not just for those in the list. It should be noted that **x** does not need to be a single value though – more on that shortly.

Regarding the logic functions,

=PIVOTBY(Data[#All],[Month],Data[#All],[Year],Data[#All],[Sales],AND)

would produce the following a result for a Table (**CTRL + T**) called **Data**, *viz*.

Date	Month	Year	Sales	2023	2024 Total
1/01/2023	Jan	2023	786.86	TRUE	TRUE
2/01/2023	Jan	2023	278.76	TRUE	TRUE
3/01/2023	Jan	2023	996.53	TRUE	TRUE
4/01/2023	Jan	2023	830.96	TRUE	TRUE
5/01/2023	Jan	2023	916.71	TRUE	TRUE
6/01/2023	Jan	2023	242.19	TRUE	TRUE
7/01/2023	Jan	2023	173.48	TRUE	TRUE
8/01/2023	Jan	2023	318.57	TRUE	TRUE
9/01/2023	Jan	2023	787.53	TRUE	TRUE
10/01/2023	Jan	2023	518.05	TRUE	TRUE
11/01/2023	Jan	2023	685.73	TRUE	TRUE
12/01/2023	Jan	2023	819.77	TRUE	TRUE
13/01/2023	Jan	2023	571.01	TRUE	TRUE
14/01/2023	Jan	2023	857.65	TRUE	TRUE
15/01/2023	Jan	2023	366.99	TRUE	TRUE
16/01/2023	Jan	2023	685.93	TRUE	TRUE
17/01/2023	Jan	2023	470.46	TRUE	TRUE
18/01/2023	Jan	2023	152.52	TRUE	TRUE
19/01/2023	Jan	2023	509.55	TRUE	TRUE
20/01/2023	Jan	2023	779.74	TRUE	TRUE
21/01/2023	Jan	2023	268.61	TRUE	TRUE
				Total	TRUE

It's funny to see that the 2023 total is FALSE, the 2024 total is TRUE, yet overall it's TRUE (if you think about the condition, that makes sense as even number of TRUES + odd number of TRUES = odd number of TRUES). It just seems weird to us as logicians.

NOT doesn't seem to work, but that is because even though it accepts one argument, it has to be something that reduces to one value:

=PIVOTBY(Data[#All],[Month],Data[#All],[Year],Data[#All],[Sales],NOT)

H4 : X ✓ fx =PIVOTBY(Data[#All],[Month],Data[#All],[Year],Data[#All],[Sales],NOT)

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3												
4		Date	Month	Year	Sales				2023	2024	Total	
5		1/01/2023	Jan	2023	0			Apr	#VALUE!		#VALUE!	
6		2/01/2023	Jan	2023	278.76			Aug	#VALUE!		#VALUE!	
7		3/01/2023	Jan	2023	996.53			Dec	#VALUE!		#VALUE!	
8		4/01/2023	Jan	2023	830.96			Feb	#VALUE!	#VALUE!	#VALUE!	
9		5/01/2023	Jan	2023	916.71			Jan	#VALUE!	#VALUE!	#VALUE!	
10		6/01/2023	Jan	2023	242.19			Jul	#VALUE!		#VALUE!	
11		7/01/2023	Jan	2023	173.48			Jun	#VALUE!		#VALUE!	
12		8/01/2023	Jan	2023	318.57			Mar	#VALUE!		#VALUE!	
13		9/01/2023	Jan	2023	787.53			May	#VALUE!		#VALUE!	
14		10/01/2023	Jan	2023	518.05			Nov	#VALUE!		#VALUE!	
15		11/01/2023	Jan	2023	685.73			Oct	#VALUE!		#VALUE!	
16		12/01/2023	Jan	2023	819.77			Sep	#VALUE!		#VALUE!	
17		13/01/2023	Jan	2023	571.01			Total	#VALUE!	#VALUE!	#VALUE!	
18		14/01/2023	Jan	2023	857.65							
19		15/01/2023	Jan	2023	366.99							

See how it turns to #VALUE!? If we have just one value per month:

H4 : X ✓ fx =PIVOTBY(Data[#All],[Month],Data[#All],[Year],Data[#All],[Sales],NOT)

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3												
4		Date	Month	Year	Sales				2023	Total		
5		1/01/2023	Jan	2023	0			Apr	FALSE	FALSE		
6		1/02/2023	Feb	2023	278.76			Aug	FALSE	FALSE		
7		1/03/2023	Mar	2023	996.53			Dec	FALSE	FALSE		
8		1/04/2023	Apr	2023	830.96			Feb	FALSE	FALSE		
9		1/05/2023	May	2023	916.71			Jan	TRUE	TRUE		
10		1/06/2023	Jun	2023	242.19			Jul	FALSE	FALSE		
11		1/07/2023	Jul	2023	173.48			Jun	FALSE	FALSE		
12		1/08/2023	Aug	2023	318.57			Mar	FALSE	FALSE		
13		1/09/2023	Sep	2023	787.53			May	FALSE	FALSE		
14		1/10/2023	Oct	2023	518.05			Nov	FALSE	FALSE		
15		1/11/2023	Nov	2023	685.73			Oct	FALSE	FALSE		
16		1/12/2023	Dec	2023	819.77			Sep	FALSE	FALSE		
17								Total	#VALUE!	#VALUE!		

[Yes, I know the date format will confuse our US readers. That was entirely intentional. 😊]

Note the totals still give the error.

Hopefully, this makes it clearer as you can start to play with – and more importantly, devise – your own eta lambda functions. Let's hope they roll out into Production very shortly...

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, which apologies, was the January one on our website (in case you missed it)...

As the sun blazes in its full glory and the beaches beckon, it's time to cool off with a refreshing Excel challenge. While you sip on your lemonade or enjoy the gentle sea breeze, let's dive into a (southern hemisphere) summer-themed Excel puzzle.

Imagine you're organizing a summer camp and you have a list of activities planned for each day of the week. Each activity has a maximum number of participants, and you've got a dynamic list of campers signing up every day. You need to ensure that the number of campers for each activity does not exceed its capacity.

Here's the twist. You're given two Excel Tables: **Activities_Table** and **SignUps_Table**. The **Activities_Table** lists various activities along with

	A	B
1	Activity ▾	Max Capacity ▾
2	Beach Volleyball	20
3	Surfing	15
4	Nature Walk	25
5	Art Workshop	10
6	Kayaking	12

	A	B	C
1	Camper Name ▾	Activity ▾	Day ▾
2	Alice	Beach Volleyball	Monday
3	Bob	Beach Volleyball	Tuesday
4	Charlie	Surfing	Monday
5	David	Nature Walk	Tuesday
6	Eve	Art Workshop	Wednesday

their maximum capacity. The **SignUps_Table** records the name of the camper and their chosen activity for each day of the week.

Your challenge is to create a single dynamic Excel formula that will check the total number of sign-ups for each activity every day and flag any activity that exceeds its maximum capacity.

For example, if the activity 'Beach Volleyball' has a maximum capacity of 20 and 22 campers have signed up for it on Tuesday, the formula should flag 'Beach Volleyball' for Tuesday.

You can setup your solution however you like.

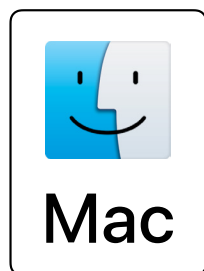
Remember, the solution should:

- be dynamic and update immediately as new sign-ups are added
- use only Excel formulas - no Power Query, Get & Transform, or VBA allowed
- be case-insensitive.

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

Excel for Mac

With Steve Kraynak joining the team, we thought we would exploit his knowledge and recant all about Microsoft Excel for Mac. Each month, we'll cover a different topic to help you understand how Excel for Mac is different than Excel for Windows. This month, we show how to work around the missing feature, Evaluate Formula, by using a new feature and an old-school technique.



This is a follow-up to our previous blog about writing and editing formulae, and how those tasks are a little different on Mac than on Windows. Some formulae are very simple. Debugging them is a simple matter of looking at the formula and knowing whether it's right or wrong.

For example, you may have a formula calculating the **SUM** of a few values. You can often just look at the values, calculate it in your head, and know whether you've got it right or wrong.

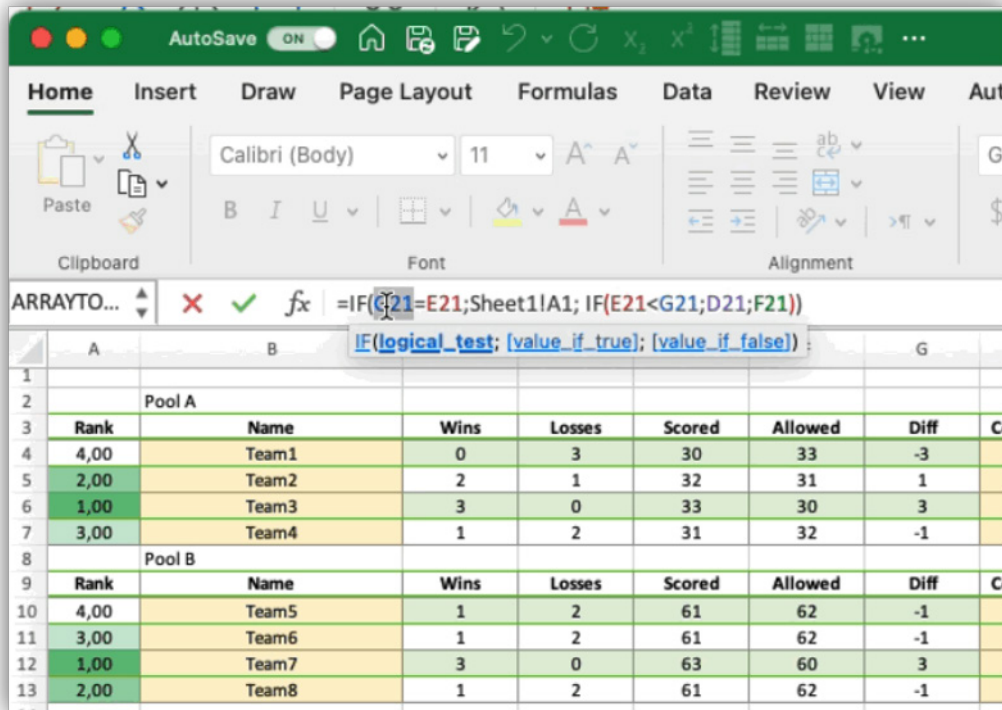
On the other hand, you may have much more complex formulae, where checking is not so easy to do in your head. In that case, there are a few tricks and tools you can use. However, there is one less tool on Mac than there is on Windows, because the Evaluate Formula dialog isn't available on Mac (as at the time of writing).

Value Preview

Early last year, Microsoft added a handy feature called Value Preview ToolTips to Excel on Windows, Mac and the web. These ToolTips are a great way for you to check the values being calculated by your formula and the parts making up the formula. It's also very easy to use. Simply highlight part of your formula, and a ToolTip will appear showing you the current value of whatever you've selected. It works with references, functions, parameters within a function and the entire formula. In some

cases, you might select something that doesn't calculate and then Excel won't show a value.

To make sure you select parts of the formula that can be calculated, you can use the function ToolTips to your advantage. As shown in the image below, you can click the function arguments to highlight the related part of the formula. In the **IF** formula, clicking on **logical_test** highlights that part of the formula and shows that it's currently FALSE:



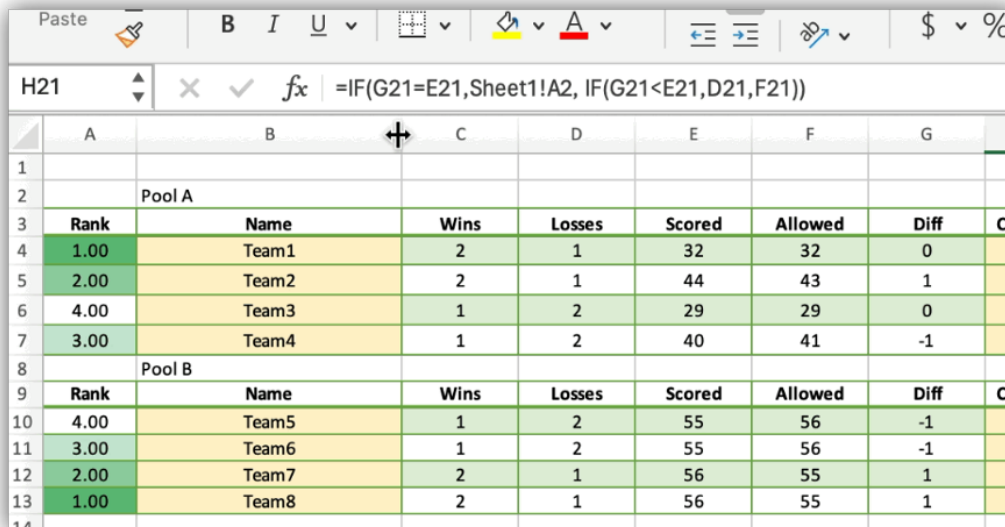
There's a shortcut key you can press to disable or enable the Value Preview ToolTips. Just press **CTRL + Opt + P**. On Mac, this only works when you are not editing a cell (on Windows, it can be turned on / off anytime with **CTRL + ALT + P**).

F9 and ESC: The Old Way

A handy trick for debugging formulae that is much less necessary now that we have the Value Preview ToolTips is to press **F9** when you've selected part of the formula that you want to evaluate. Excel will calculate the selection and replace what you selected with the current value. Just remember to press **ESC** after doing this, otherwise you might

accidentally save a hard-coded value where you didn't mean to.

As you can see below, we highlight part of the formula, then press **F9** which changes the cell reference **G21** to its value 11. Then, we press **ESC** to avoid saving it as a hard-coded value.



Formula Builder

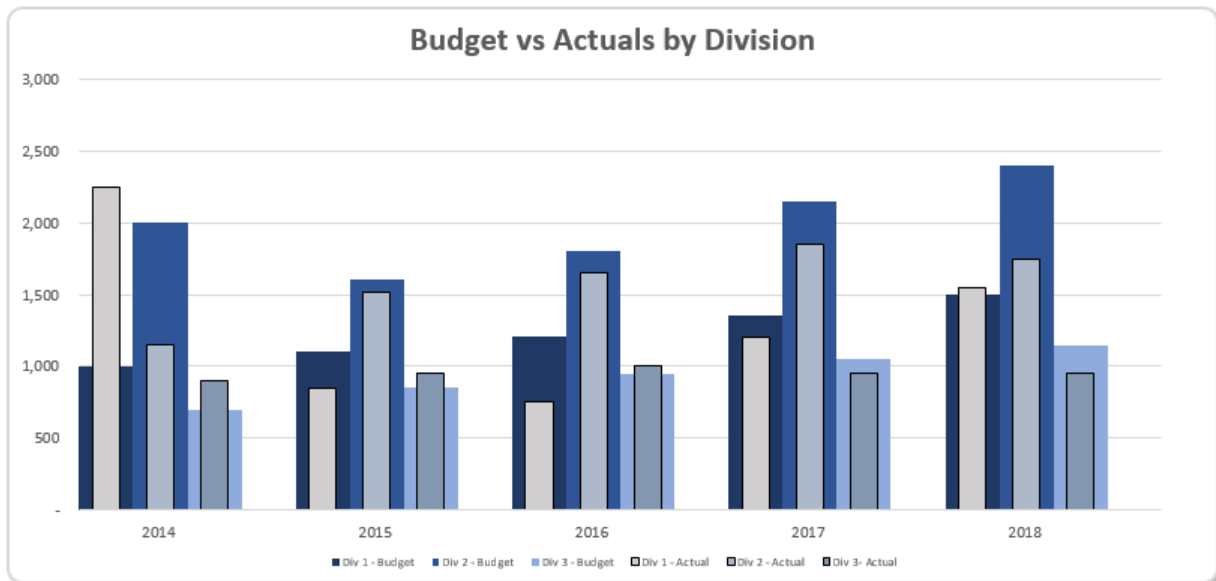
As we described in previously, you can use the Formula Builder to help debug. When you've entered enough information for Excel to start calculating, it will show you the current values above each field in the pane.

We'll continue next month...

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 develop Multiple Bullet charts in Excel.

Financial modelling is not always about complex depreciation schedules, financial statement analyses, scenario managers and long, complex formulae. Sometimes, a picture can tell a thousand-word story. With this borne in mind, let's take a look at a useful chart you may require from time to time – the Multiple Bullet chart:



Here, we create a single chart in Excel that will display the target amounts and have the actual amounts flow through like a bullet chart for each time period. The problem is, Bullet charts – never mind “multiple” ones still don't actually exist in Excel.

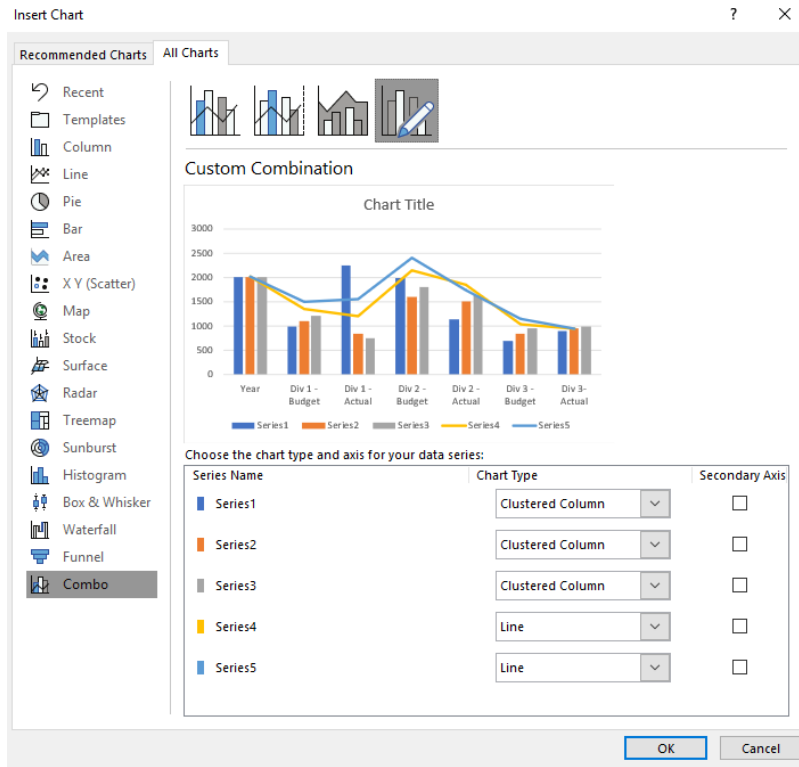
So, let's create one.

The first step is to format our data properly in a way that we can plot it on a chart in Excel. We will be using the following dataset in the current format:

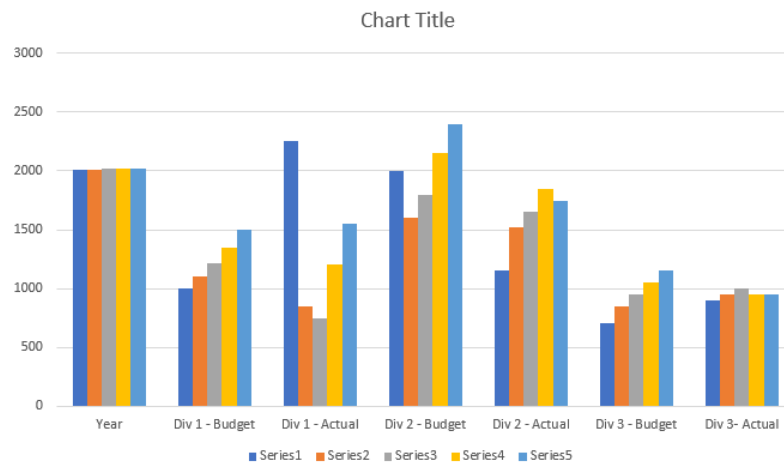
Actual Costs vs Budget Costs - by Division

Year	Div 1 - Budget	Div 1 - Actual	Div 2 - Budget	Div 2 - Actual	Div 3 - Budget	Div 3 - Actual
2014	1,000	2,250	2,000	1,150	700	900
2015	1,100	850	1,600	1,520	850	950
2016	1,210	750	1,800	1,650	950	1,000
2017	1,350	1,200	2,150	1,850	1,050	950
2018	1,500	1,550	2,400	1,750	1,150	950

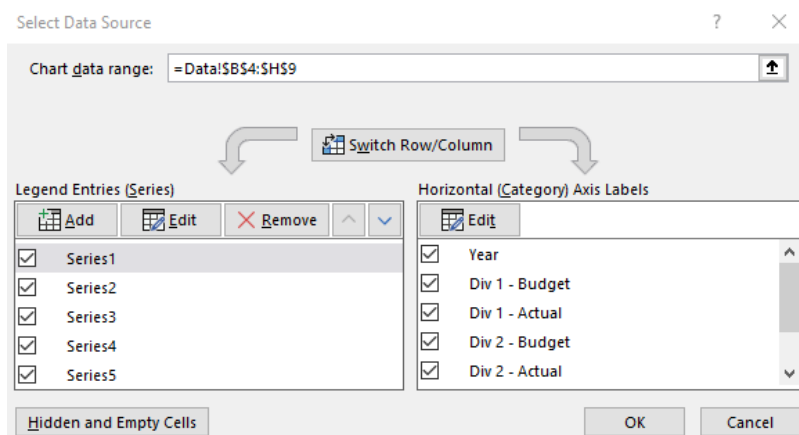
With the data in hand, we can create the chart. As part of this solution, we will be using Clustered Columns chart in Excel, by highlighting the table and clicking on **Insert -> Clustered Column** (from the Charts group of the Insert tab of the Ribbon):



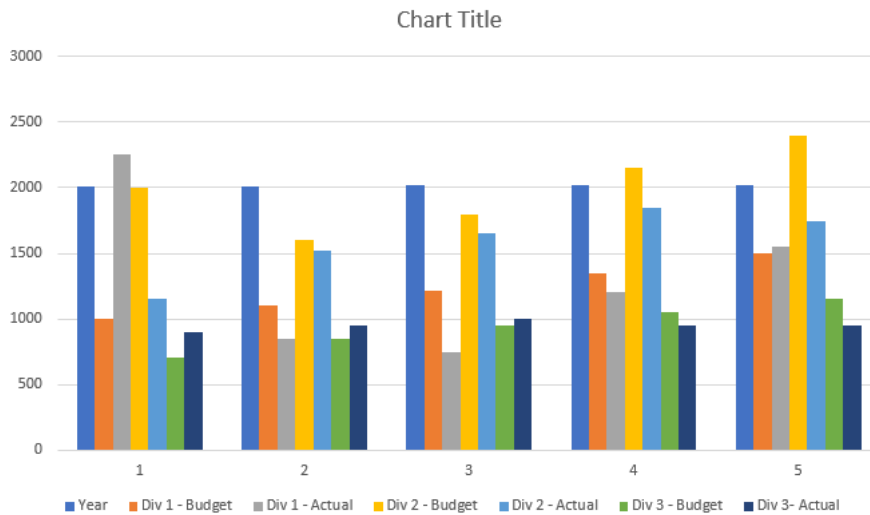
Interestingly, it will typically default to Combo (even on this command) and you may need to toggle all of the chart types to be 'Clustered Columns'. If that glitch doesn't happen for you, that's great, you have missed a step!



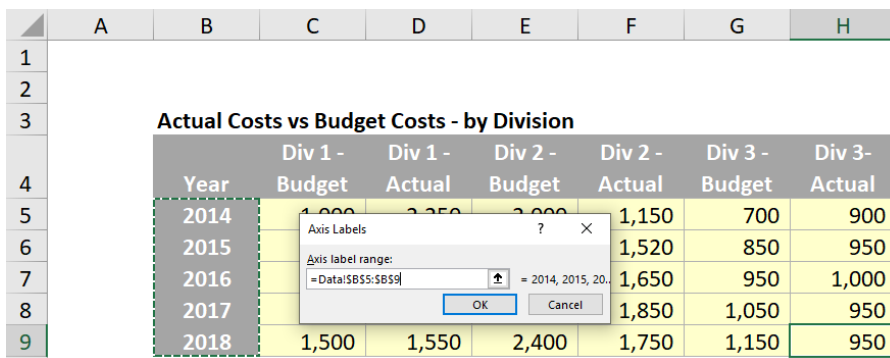
Then right-click on the chart and choose 'Select Data...' in the 'Select Data Source' dialog box, where we can switch the row / column data.



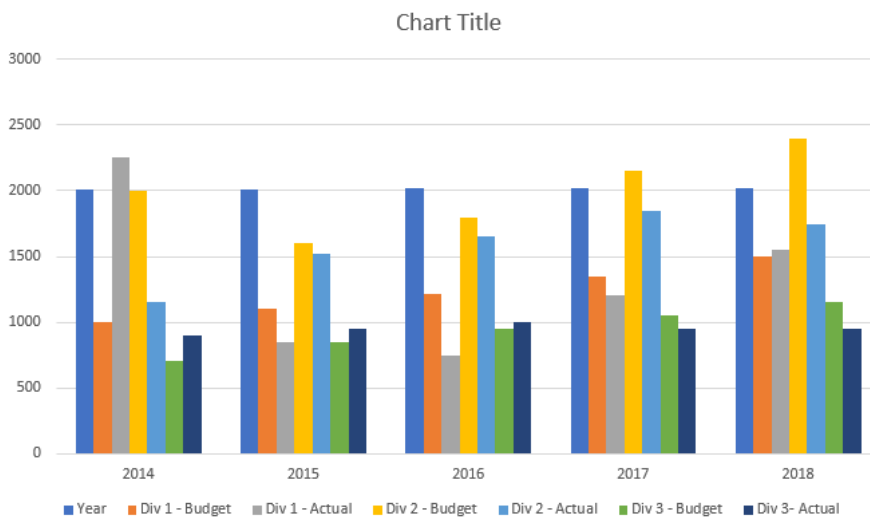
The chart should look something like this now:



Notice that the x-axis is labelled as 1, 2, 3, 4 and 5 when we want it to be 2014, 2015, 2016 and so on... Therefore, return to the 'Select Data Source' dialog and edit the horizontal 'Axis Labels':

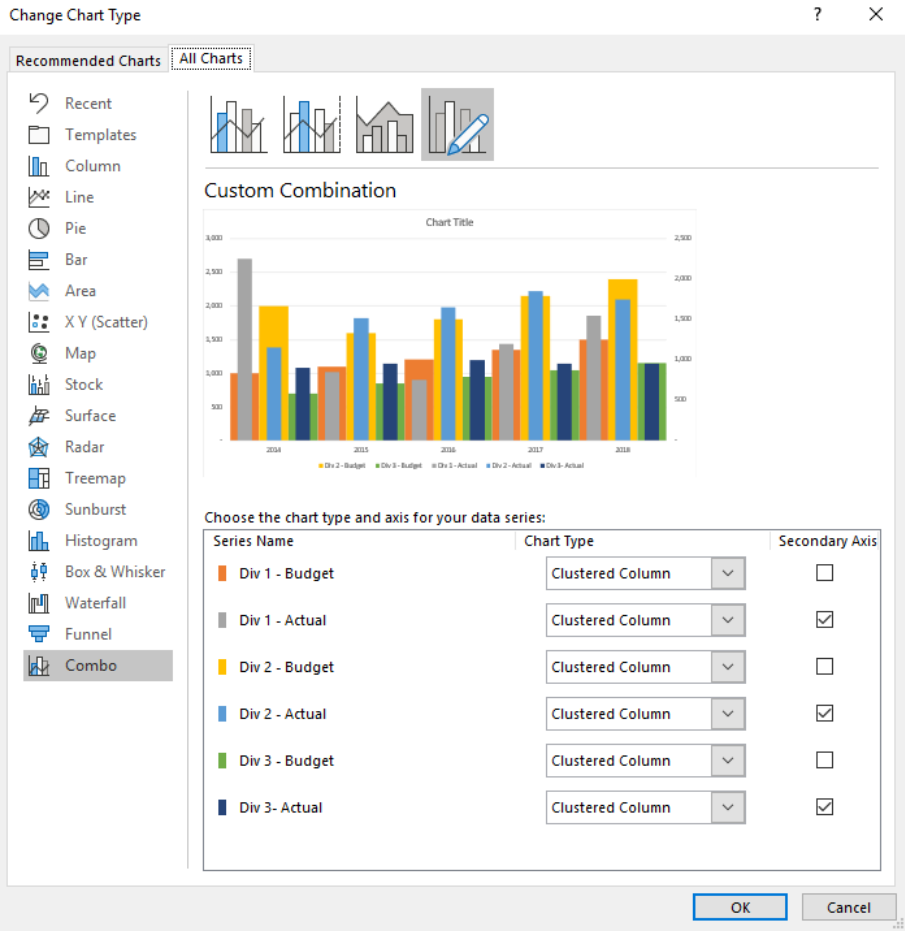


Great; now we have our years in order.



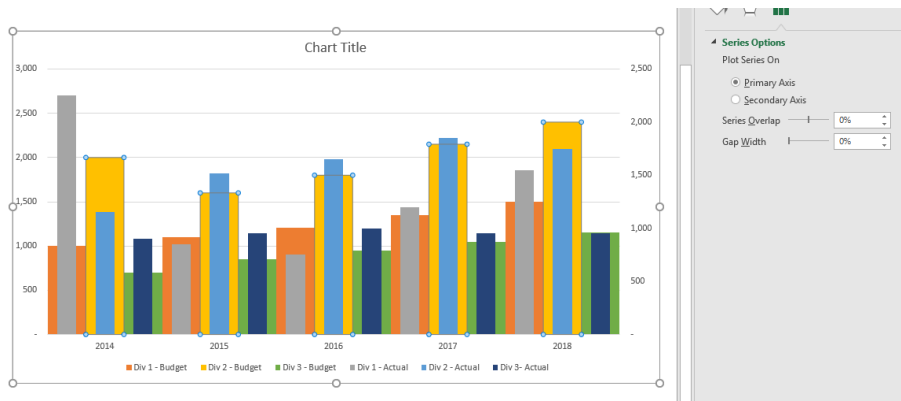
The next step is to organise the columns into primary and secondary axes. Click on the chart, then click on the Design tab on the Ribbon, and then the 'Change Chart Type' option. This brings up the 'Change Chart Type' dialog, where we can assign different data series onto a secondary

axis. In this case we will assign all of the Actual series to the secondary axis (this is because the secondary axis is displayed on top). We will see the first sight of our Multiple Bullet charts:

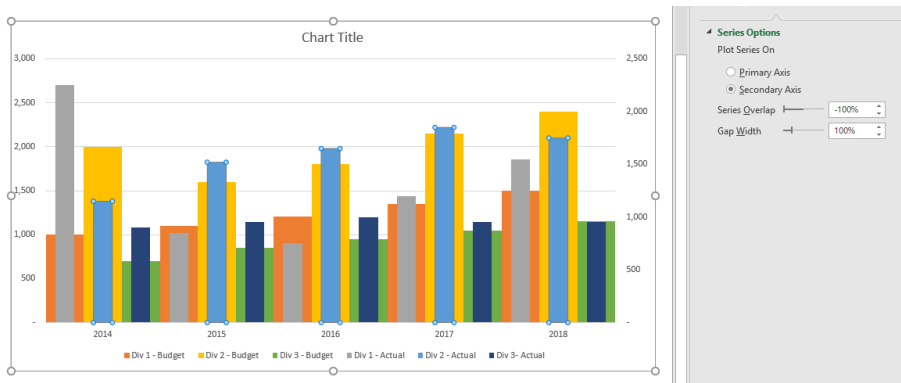


Now, we need to do further formatting to the chart.

First, we remove the Year series from the chart. Then, we change adjust the format of the data series to make it appear like a Bullet chart. We do this by changing the 'Series Overlap' and 'Gap Width' values in the 'Format Data Series' panel. We adjust the 'Gap Width' of the budgeted series to 0%.



To format the primary axis data series, we set the 'Series Overlap' to -100% and 'Gap Width' to 100% (of course you can vary these settings to create a slightly different chart):



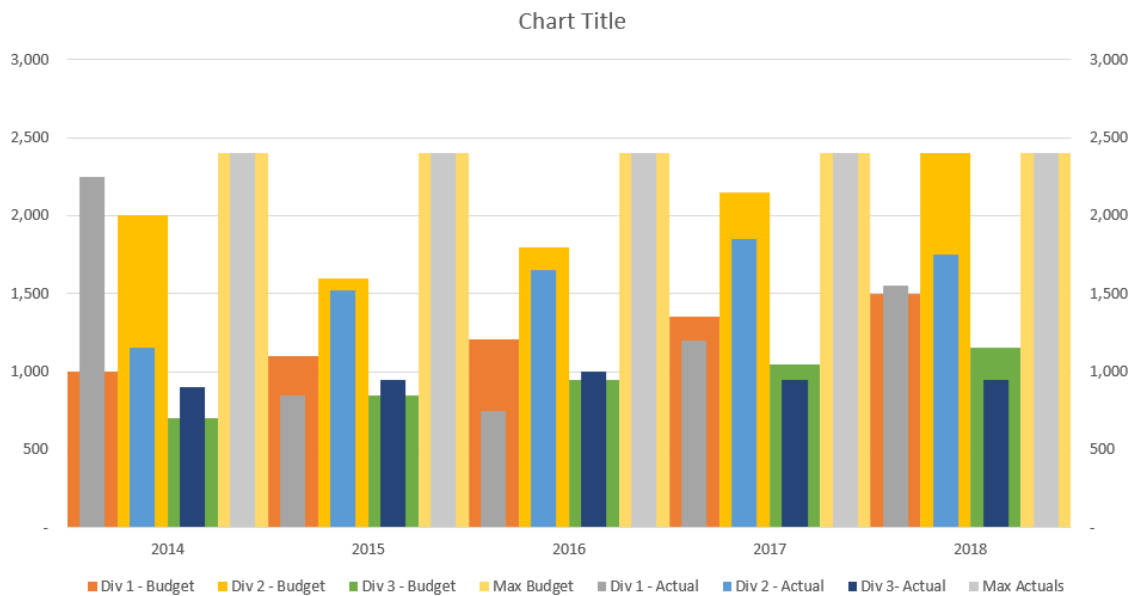
Our chart is starting to come together. At this point we have two things to deal with:

1. The axes have different maximum amounts – this will cause confusion as our budget amounts are being compared to our actual amounts that are on different axes
2. We do not have clear spacing between the years. This may make it difficult for end users to read the chart.

To deal with these two issues, we can include new data series that will ‘control’ the maximum amount in the axis. We do this by using the **MAX** formula to retrieve the maximum amount from our data:

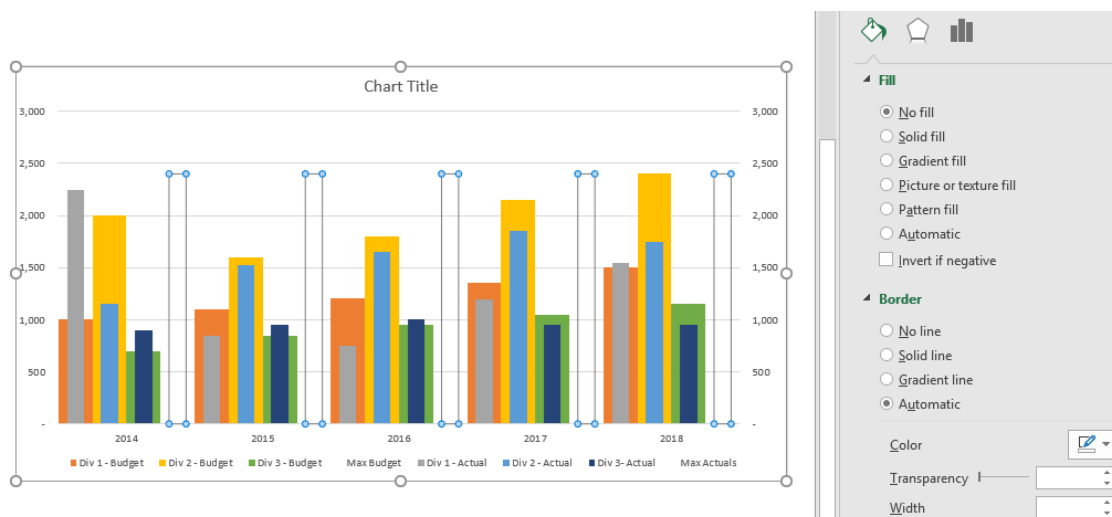
		Div 1 - Budget	Div 1 - Actual	Div 2 - Budget	Div 2 - Actual	Div 3 - Budget	Div 3 - Actual	Max Budget	Max Actuals
2014		1,000	2,250	2,000	1,150	700	900	=MAX(\$C\$5:\$H\$9)	2,400
2015		1,100	850	1,600	1,520	850	950	2,400	2,400
2016		1,210	750	1,800	1,650	950	1,000	2,400	2,400
2017		1,350	1,200	2,150	1,850	1,050	950	2,400	2,400
2018		1,500	1,550	2,400	1,750	1,150	950	2,400	2,400

This will insert a new data series into the chart that will always have the maximum value from the dataset:

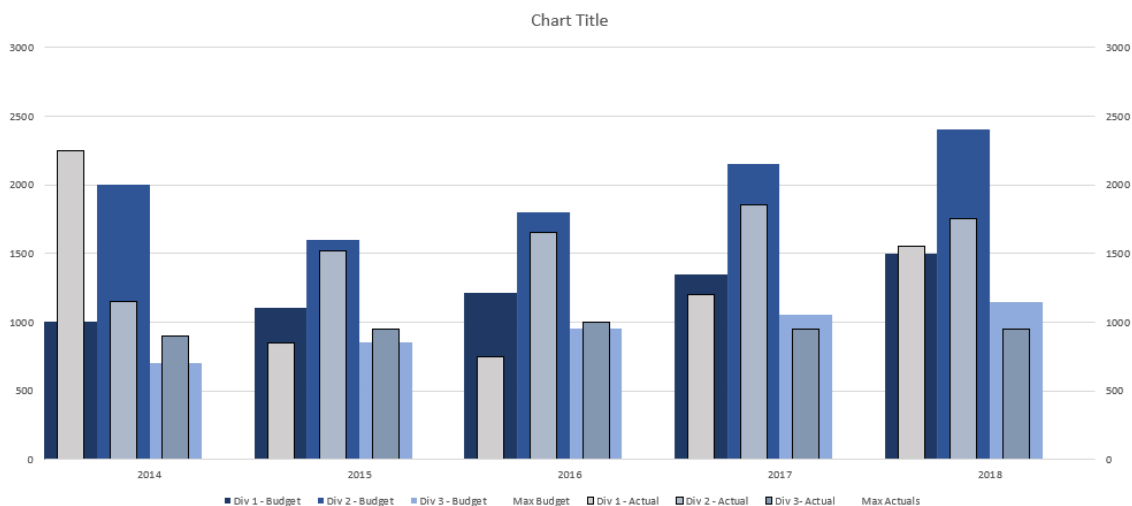


The two axes are now set to the same maximum amounts.

We then format the two new data series with ‘No fill’, which essentially renders them invisible:

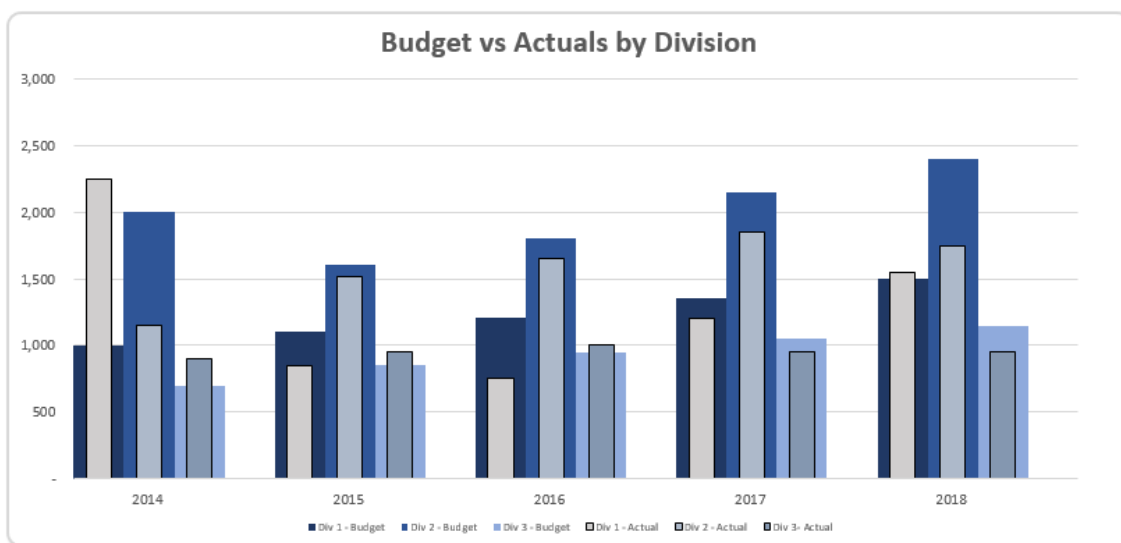


Two birds one bar... wait, was it stone? That would conclude it for our chart, if we did not care for colour. We should apply a different colour palette:



Of course, you do not have to pick these exact colours, but we've tried to pick a palette that looks somewhat desirable. The final adjustments to the chart are:

- 1) Delete the 'Max Budget' and 'Max Actuals' series from the legend
- 2) Hide the secondary axis
- 3) Add spaces to the end of the years data, e.g. use "2014 " instead of "2014", so that the years appear to centre in their respective groupings
- 4) Give the chart a name.



There we have it: a versatile Multiple Bullet chart.

Until 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 considers the concept of a variable scope at a module level.

Variables can also be used in different subroutines and functions. If there are items that are known to have constant values throughout the entire workbook, they can be declared explicitly in one place for easy reference.

Module Level

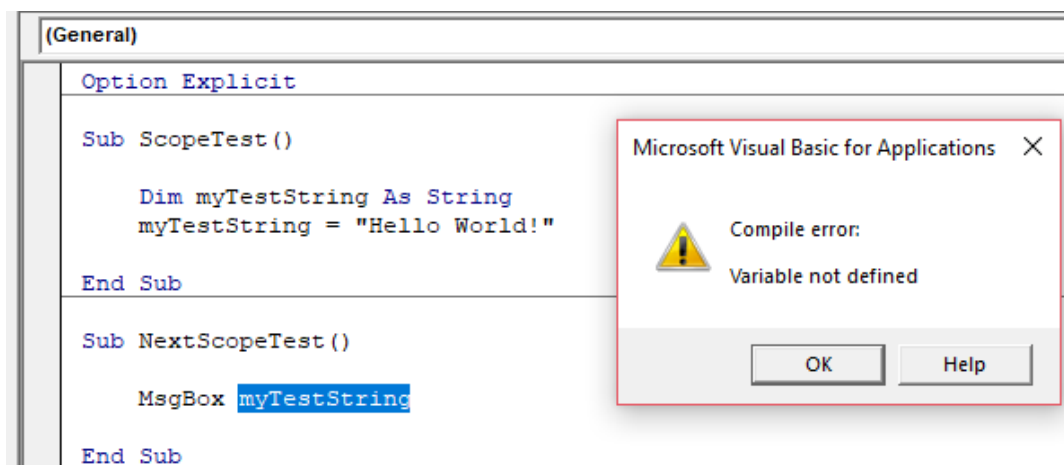
Let's write a new module as follows:

```
Option Explicit
```

```
Sub ScopeTest()  
    Dim myTestString As String  
    myTestString = "Hello World!"  
End Sub
```

```
Sub NextScopeTest()  
    MsgBox myTestString  
End Sub
```

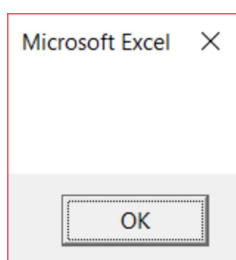
The declaration for the variable is in **ScopeTest** but the subroutine **NextScopeTest** calling it is not the one that defined it. What happens upon execution?



The 'Variable not defined' message pops up again. Let's move the declaration OUTSIDE the subroutine, under the 'Option Explicit' statement as follows:

```
Option Explicit  
Dim myTestString As String  
Sub ScopeTest()  
    myTestString = "Hello World!"  
End Sub  
Sub NextScopeTest()  
    MsgBox myTestString  
End Sub
```

When **NextScopeTest** is run the following happens:



This is because though the variable has been declared, it has not been *initialised*. If **ScopeTest** is run prior to **NextScopeTest** then:



This shows that the variable is accessible to the entire module and is changed as required. The **Dim** statement outside any subroutines means the variable is accessible within the module that it is declared in.

More next time.

Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we will use the **SEARCH** and **SWITCH** functions to create custom columns that group text columns into new categories.

Last month, we covered the **SEARCH** function and we used it to identify text strings within a column. This time, we are going to combine the **SEARCH** function with the **SWITCH** function to demonstrate a potential use case.

Let's look at a similar data table to the one used last time out:

Product Type
Casual Slippers ,
Business Shoes
Heavy Duty Gloves
Heavy Duty Shoes
Shoes Heavy Duty
Gloves (Light)
Gloves (Heavy Duty)
Shoe Business
Leather Belt
Glove and Shoe Leather Set

Our task now is to create a new column that works as a new subcategory for each of the products. For example, if we want the new column to classify Business Shoes as Footwear, we will use the following **IF** function:

```
=IF(  
    SEARCH(  
        "Shoe",[Product Type],,0)>0,  
        "Footwear",  
        ""  
    )
```

This creates the following calculated column as follows:

Product Type	IF Reclassification
1 Casual Slippers ,	
2 Business Shoes	Footwear
3 Heavy Duty Gloves	
4 Heavy Duty Shoes	Footwear
5 Shoes Heavy Duty	Footwear
6 Gloves (Light)	
7 Gloves (Heavy Duty)	
8 Shoe Business	Footwear
9 Leather Belt	
10 Glove and Shoe Leather Set	Footwear

If we wanted to include Casual Slippers, we would have to use a nested IF statement:

```
=IF(
  SEARCH(
    "Shoe*",[Product Type],,0)>0,
    "Footwear",
    IF(
      SEARCH(
        "Slipper?", [Product Type],,0)>0,
        "Footwear",
        ""
      )
    )
)
```

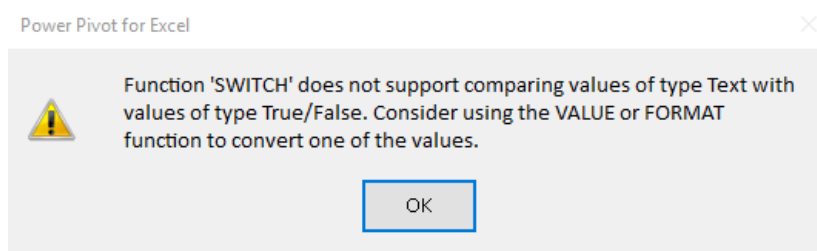
Product Type	IF Reclassification	Nested IF Reclassification
1 Casual Slippers ,		Footwear
2 Business Shoes	Footwear	Footwear
3 Heavy Duty Gloves		
4 Heavy Duty Shoes	Footwear	Footwear
5 Shoes Heavy Duty	Footwear	Footwear
6 Gloves (Light)		
7 Gloves (Heavy Duty)		
8 Shoe Business	Footwear	Footwear
9 Leather Belt		
10 Glove and Shoe Leather Set	Footwear	Footwear

But what if we wanted to reclassify Gloves and Belts as Clothing? It would result in a potentially nasty nested **IF** statement! As we hinted earlier in this blog, we can use the **SWITCH** function together with the **SEARCH** function instead:

```
=SWITCH(
  TRUE(),
  SEARCH(
    "Shoe*",[Product Type],,0)>0,
    "Footwear",
  SEARCH(
    "Slippers*",[Product Type],,0)>0,
    "Footwear",
  SEARCH(
    "Gloves*",[Product Type],,0)>0,
    "Clothing",
  SEARCH(
    "Belt*",[Product Type],,0)>0,
    "Clothing"
)
```

Product Type	IF Reclassification	Nested IF Reclassification	Reclassification
1 Casual Slippers ,		Footwear	Footwear
2 Business Shoes	Footwear	Footwear	Footwear
3 Heavy Duty Gloves			Clothing
4 Heavy Duty Shoes	Footwear	Footwear	Footwear
5 Shoes Heavy Duty	Footwear	Footwear	Footwear
6 Gloves (Light)			Clothing
7 Gloves (Heavy Duty)			Clothing
8 Shoe Business	Footwear	Footwear	Footwear
9 Leather Belt			Clothing
10 Glove and Shoe Leather Set	Footwear	Footwear	Footwear

In this case, we have to use TRUE as the expression so that as the function cycles through each row the expression will be 'true', so it will perform the subsequent searches for each row. If we do not use the TRUE function as the expression, we will get the following error message:



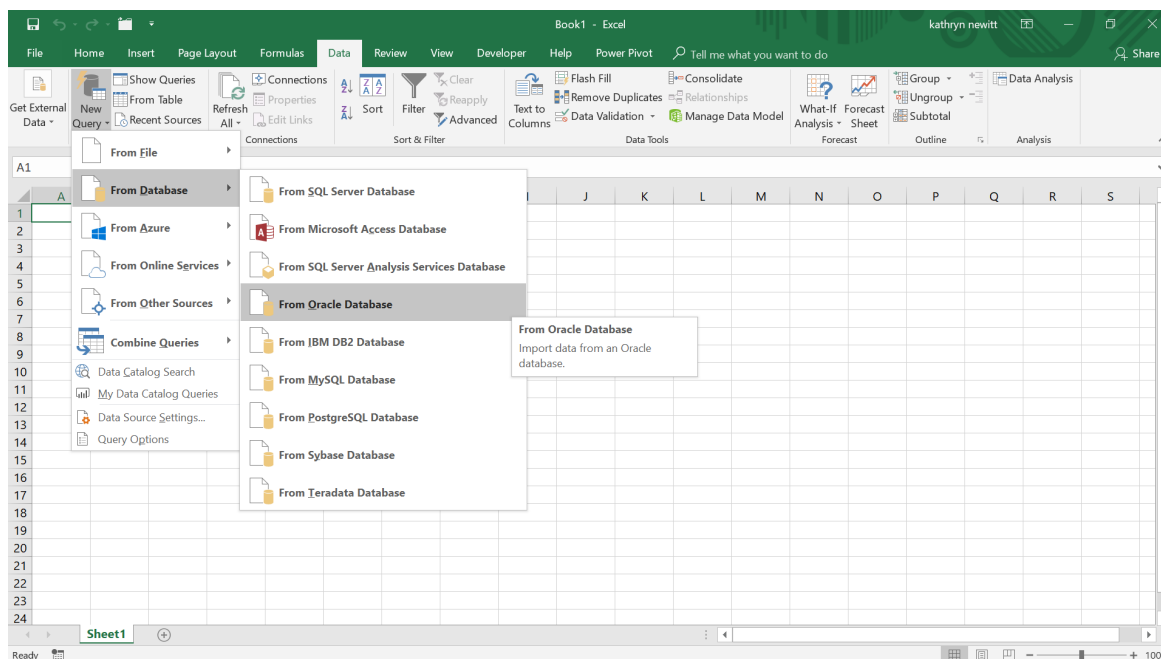
That's it for this month; more next time.

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. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at how to upload data from a relational database.

Power Query can be used to upload data from a wide range of sources; truly, it is an excellent tool for cleaning up and formatting data. However, data is rarely static, so there are some good habits that I can adopt when deciding what to upload, particularly from a relational database.

Upload Views, Not Tables (Usually)



If we have a table of static or standing data that has come from a relational database like Oracle, then it's probably fine to upload the underlying table. By this, we mean data like countries or towns, which are unlikely to change much. Other data is likely to change a great deal and this can mean that whole columns may appear and disappear.

The database administrator (DBA) is busy keeping track of in-house programs that might break every time a column is changed and there is no tool (yet!) that can indicate whether a particular column is used in Power Query. As programmers, we've spent many happy (!) hours trying to track down every possible repercussion when we've needed to make a change to a column in a database to enable our new development work to go ahead.

Since we understand the problem, we can make the DBA's life easier. A view is a virtual table which is created by an SQL statement. It appears just like a normal database table as far as the users of Power Query

are concerned. If views are created that helpfully have something in their title to indicate that they are in use by Power Query, e.g. '**Power_Product_Details**', this makes it clear that if the column appears in the view, it is being used by Power Query (the same approach can be used for Power BI).

The views can be put into a schema that cover all of the Power Query data, or if the Power Query reports are extensive, into several schema that cover different reporting areas, e.g. '**Power_Accounting**'. The view names can be designed to be useful to the DBA, since we may rename them as we import them if we wish.

Also, we should avoid using 'select * from table'. This is very unpopular with DBAs as it's draining and usually lazy! We should select the specific columns that will be used rather than selecting everything just in case. It makes for a much more efficient system – all round.

What's in a Name?

In Power Query, the end user is more important than the programmer. We've already mentioned above that views can and often should be renamed when importing to Power Query so that their contents and not their origins are clear. The data may have been cleaned up beautifully, but if the columns are not easy to identify then it's not going to be useful.

Cleaning titles is important too – **AcctSuppliers** is better labelled as **Suppliers**, unless we really need to create separate **Suppliers** and **AccountSuppliers** columns, in which case writing it in full is better, even if the programmer has to type for a little longer. Adopting the use of a capital for each new word is in keeping with **M** language too, and anything which makes **M** seem friendlier has to be a good thing!

However, some end users are happier to see words separated by spaces, so it depends upon the audience.

Although this is a Power Query article, we think it best to mention implications elsewhere too. If the data being cleaned up is destined for PowerPivot (or Power BI), then it pays to think ahead. If it is likely that a measure to calculate '**CostAmount**' will be created further down the line, then having a **CostAmount** column will cause issues as the column and the measure can't have the same name. In this case, **LineCostAmount** could be a better choice to allow for a sum to be done later.

Until next month.

Power Query Pointers

There are some great updates this month. Visual calculations make their debut and are accompanied by the new-look Power BI Home (for Desktop) and an ability to add descriptions to your semantic model measures.

The full list of updates is as follows:

Reporting

- Visual calculations in Preview
- Dynamic subscriptions for Power BI reports in Preview
- On-object Interaction updates
- Power BI Home in Desktop is enabled by default in Preview
- Enhanced Reference Layer in Power BI Azure Maps Visual

Modelling

- Measure descriptions with Copilot in Preview
- DAX query view improvements in Preview

Data connectivity

- Certified connectors updates

Service

- Storytelling in PowerPoint: replace report URLs with shareable links in Power BI add-in
- Data overview in Explore
- DirectQuery connections update

Mobile

- Shared device mode is now Generally Available

Developers

- Power BI custom visuals new local storage API
- On-object Interaction support for custom visuals
- TMDL in Power BI Desktop Developer Mode

Visualisations

- New visuals in AppSource
- Multiple Sparklines
- Rose Donut Pie Chart by Powerviz
- xViz Gantt Chart by Lumel
- Control Chart XmR by Nova Silva
- Drill Down Graph PRO

Purview

- More users can now collaborate with protected PBIX in Power BI Desktop.

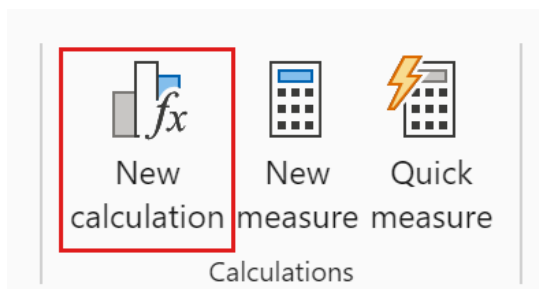
Let's look at each in turn.

Visual calculations in Preview

A new way of creating calculations has arrived. You may now add calculations directly onto your visual using visual calculations, which are **DAX** calculations that are defined and executed directly on a visual. A calculation can refer to any data in the visual, including columns, measures or other visual calculations. This approach removes the complexity of the semantic model and simplifies the process of writing **DAX**. You may use visual calculations to complete common business calculations such as running sums or moving averages. Visual calculations make it easy to create calculations that were previously very hard or even almost impossible to construct.

To use visual calculations while in Preview, you need to enable it in **Options and Settings -> Options -> Preview features**. Select visual calculations and click OK. Visual calculations will then be enabled after Power BI Desktop is restarted.

To add a visual calculation, you first need to select a visual. Then, select the 'New calculation' button in the Ribbon:



To add a visual calculation, type the expression in the Formula bar in the 'Visual Calculations edit mode' that opens. For example, in a visual that contains **Sales Amount** and **Total Product Cost by Fiscal Year**, you can add a visual calculation that calculates the profit for each year by simply typing:

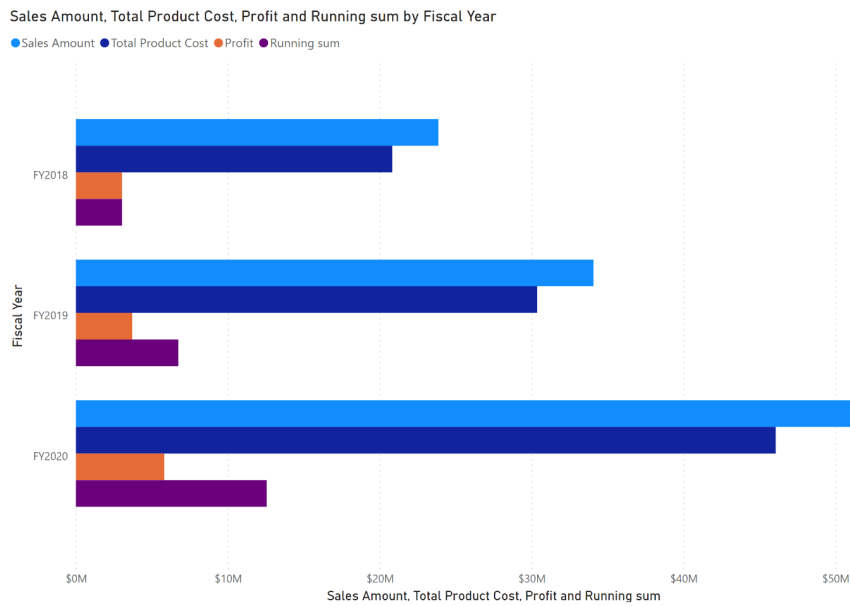
$$\text{Profit} = [\text{Sales Amount}] - [\text{Total Product Cost}]$$

Fiscal Year	Sales Amount	Total Product Cost	Profit
FY2018	\$23,860,891.17	\$20,824,957.60	3,035,933.57
FY2019	\$34,070,108.50	\$30,362,108.34	3,708,000.16
FY2020	\$51,878,274.54	\$46,070,842.02	5,807,432.52
Total	\$109,809,274.20	\$97,257,907.95	12,551,366.25

The visual matrix is updated as you add visual calculations in the Formula bar: new visual calculations are added as columns to the visual matrix. Additionally, you can easily add a running sum of profit by writing:

Running sum profit = RUNNINGSUM([Profit])

Here is a visual with the two visual calculations we have just created:



You can use many existing **DAX** functions in visual calculations. Functions specific to visual calculations are also available, such as **RUNNINGSUM**, **PREVIOUS** and **MOVINGAVERAGE**. Using these and other functions, visual calculations are much easier to read, write and maintain than the current **DAX** required. As author of the recent *Financial Modelling in Power BI*, I wholeheartedly agree!

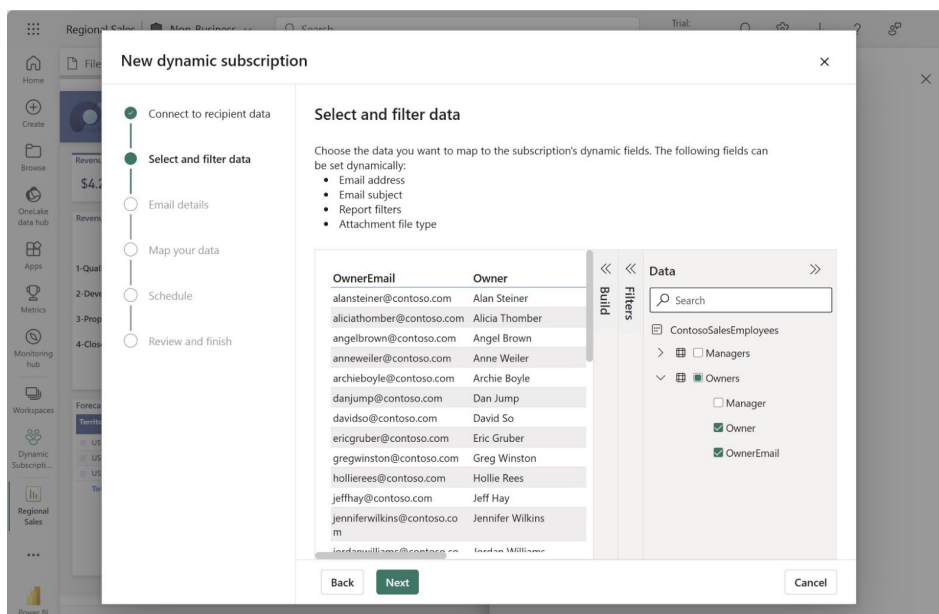
Microsoft states that “[they] are only just getting started with this Preview”. We think the more you play with this, the more you will get. Watch this space.

Dynamic subscriptions for Power BI reports in Preview

Dynamic per recipient subscriptions are now available in Preview for Power BI reports. Like dynamic subscriptions for paginated reports, you can now distribute a personalised copy of a Power BI report to each recipient of an email subscription.

Imagine you have a report that includes sales data for your entire team. You want to schedule an email subscription that sends out a PDF copy of this report to each salesperson on a weekly basis, with the report filtered to only show their sales results.

This may now be performed by connecting to a semantic model (previously Power BI dataset) that defines the mapping between recipients and respective filter values. When it’s time to send out the report, the latest data available in your semantic model will determine which employees should receive a report in their inbox and will further ascertain which filter values should be applied.

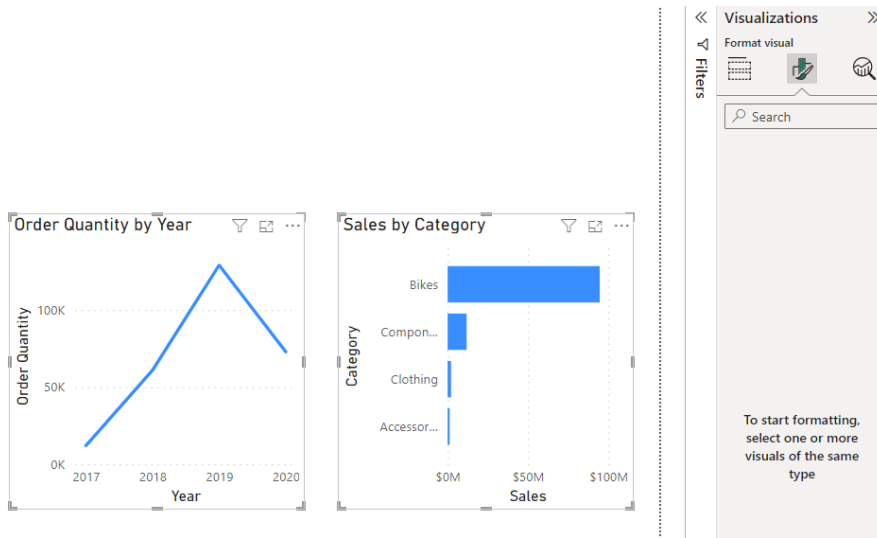


On-object Interaction updates

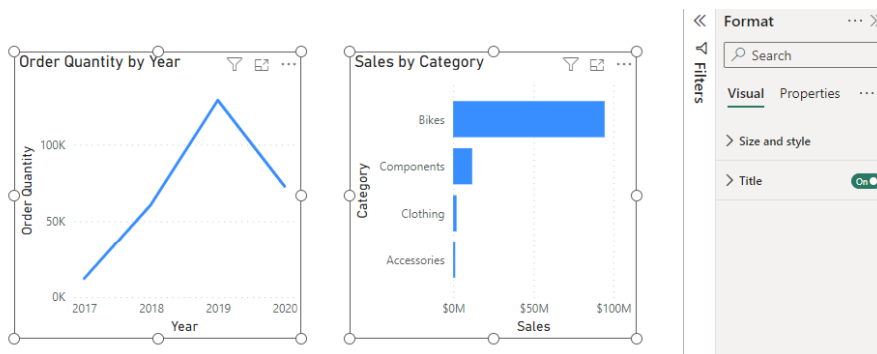
This update sees the addition of what is known as multi-visual container format support. Nothing like a succinct, easy to understand term, eh? Previously, when selecting multiple items across different visual types, the Format pane did not support any options for formatting the visuals. Now, when multi-selecting different visuals, Microsoft has added

formatting support for container formatting such as changing the size, background colour, adding a shadow or turning on / off titles in bulk.

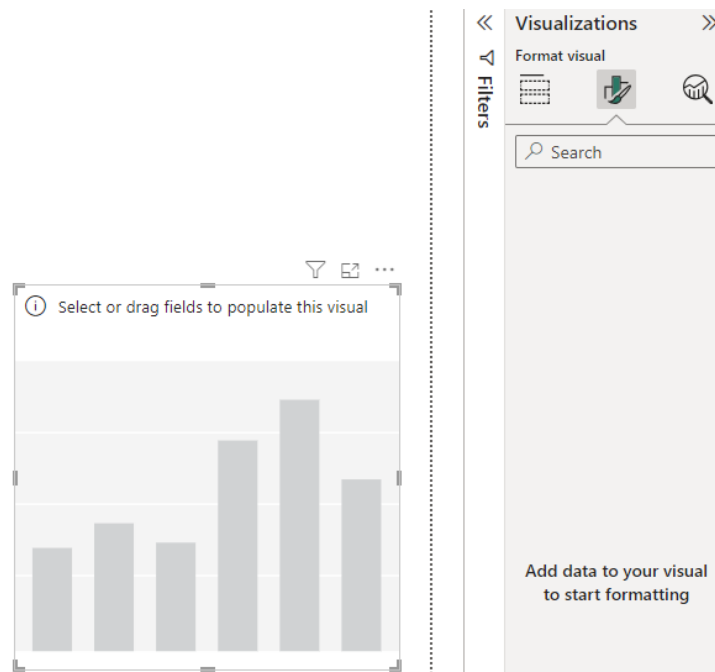
When multi-selecting different visual types (e.g. a Line chart and a Bar chart):



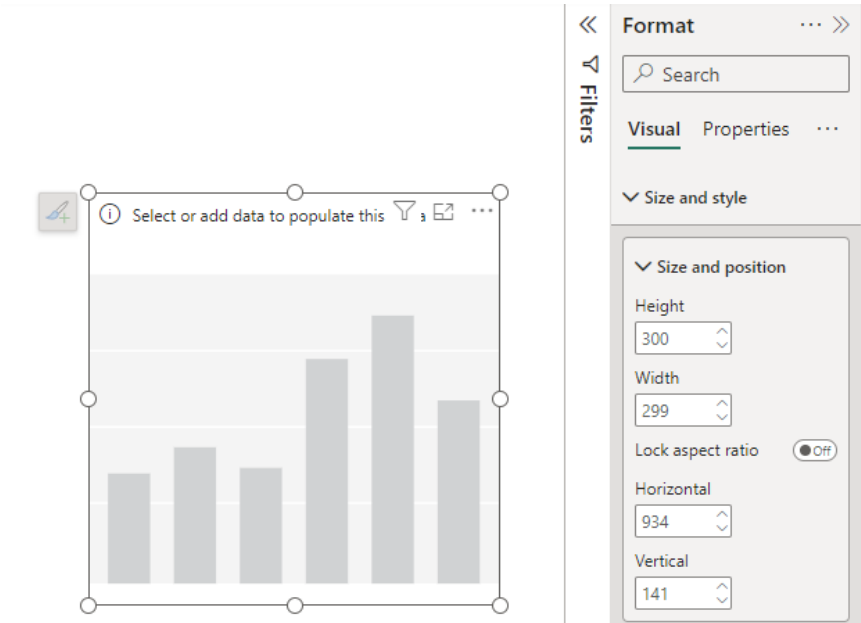
no formatting options would be available. However, now with On-object Interaction, visual container options have become available:



Power BI has also added the ability to format a visual's container size and position even if it's empty. For example, previously there would be no formatting options shown:

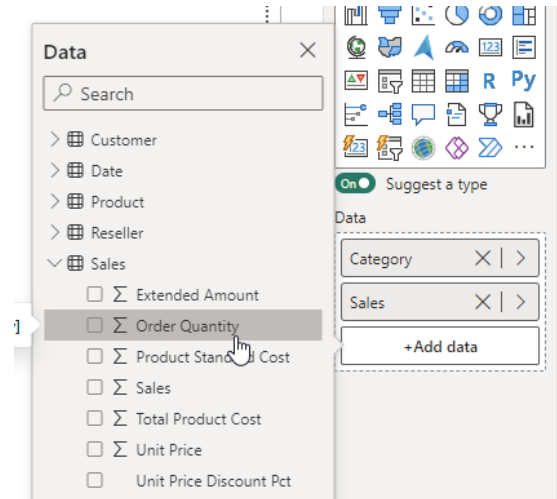


Now, with On-object Interaction, visual container size and position are available:

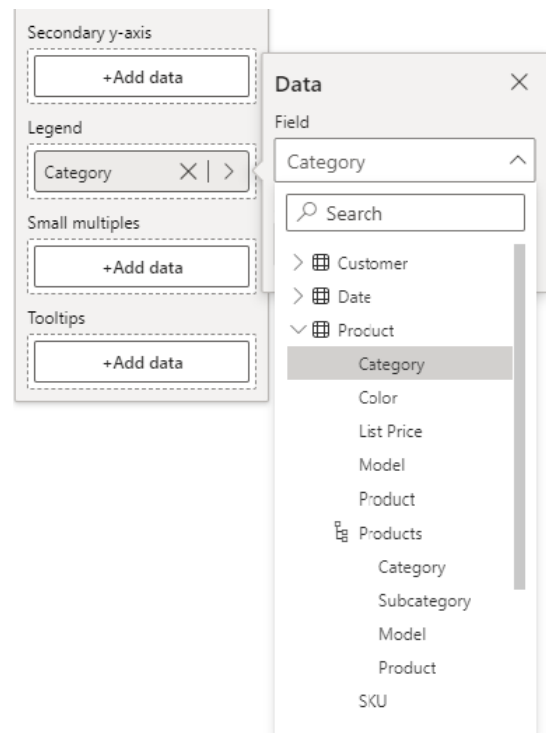


This month also welcomes a handful of quality improvements to the On-object experience, namely:

- **Bug fix:** when working with a non-visual (text box, button, image, shape), the Build pane accidentally closes automatically. The Build pane now stays open unless explicitly closed regardless of selected item type
- **Bug fix:** style bug where the Build pane was showing five [5] icons across instead of the usual six [6] has been fixed
- **Bug fix:** in some cases, the data flyout was extending beyond the window size making the search box hard to use. This has now been fixed
- **Enhancement:** when choosing a field using the data flyout, you may now click anywhere on the name, not just the checkbox next to it to select the field



- **Enhancement:** if replacing a field in a visual that does not use an aggregation or date hierarchy (other dropdowns are disabled), Power BI auto opens the data dropdown to save an extra click.



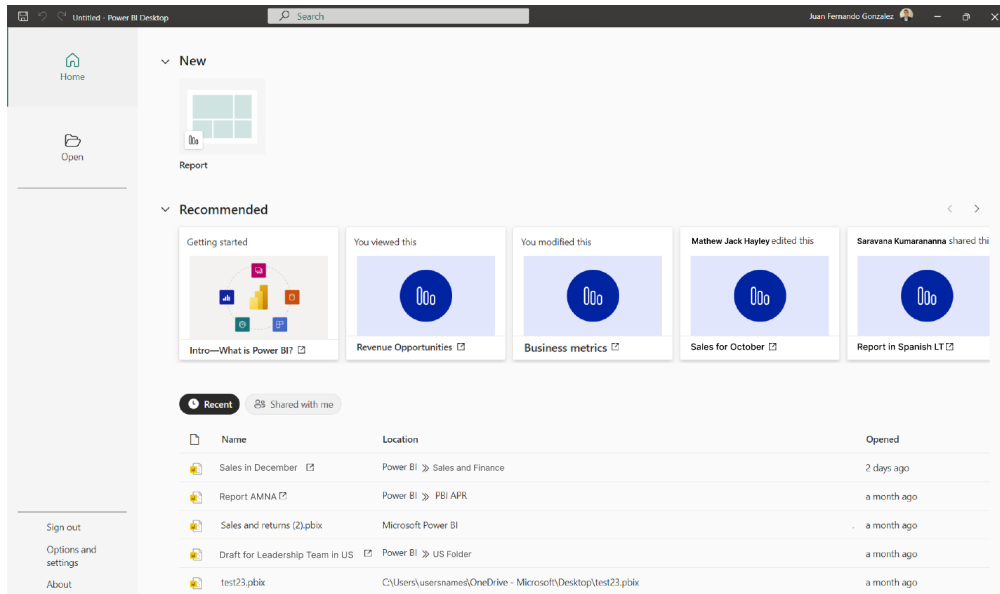
Power BI Home in Desktop is enabled by default in Preview

The new and improved Power BI Home is now set as the default experience. The Power BI Home has been redesigned to provide a centralised and familiar location for all your Power BI activities within the Desktop application. The aim is to enhance your productivity and make it easier to discover and consume content.

With Power BI Home, you no longer need to navigate through multiple menus or tabs to access your files and reports. This more intuitive interface serves as a hub, like other popular office products, where you can manage your reports all from a single location.

Now, you may:

- initiate a new report directly from the new Home screen
- access reports from recommendations that Power BI Desktop has curated
- locate your most recent reports through the 'Quick Access' lists.



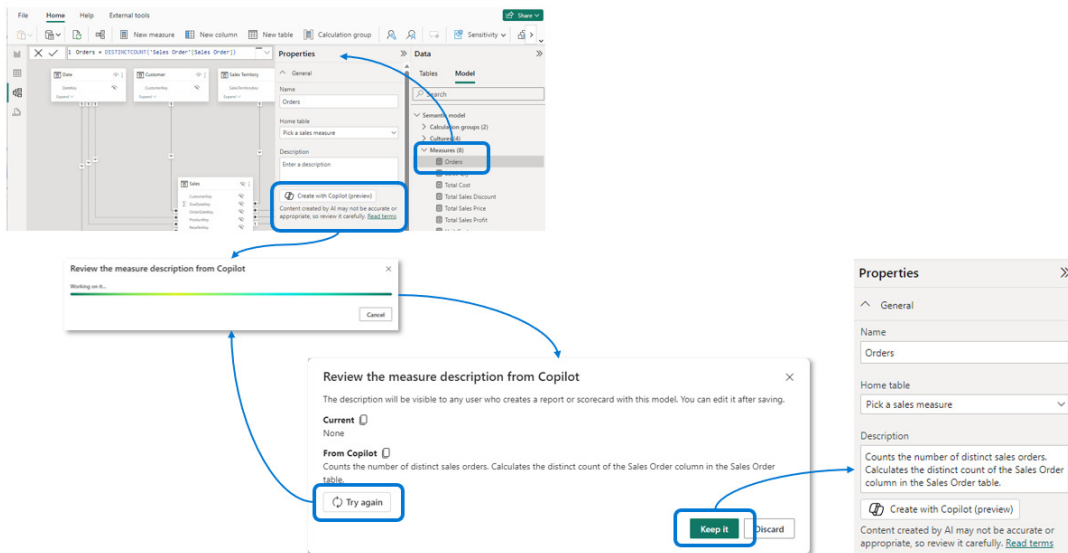
Enhanced Reference Layer in Power BI Azure Maps Visual

A significant enhancement to the Power BI Azure Maps visual reference layer feature has now been introduced. In response to valuable user feedback and in alignment with evolving industry standards, Microsoft has expanded the capabilities of the reference layer. Now, in addition to supporting the existing GeoJSON format, users may also utilise KML (Keyhole Markup Language) and WKT (Well-Known Text) formats.

The power BI team is also adding URL as a data source alongside file upload. This addition offers users more flexibility and convenience in importing spatial data into Power BI. Whether your data resides in GeoJSON, KML, WKT or through a URL link, the Power BI Azure Maps visual integrates these formats, ensuring a comprehensive and versatile geospatial analysis experience.

Measure descriptions with Copilot in Preview

You may now add descriptions to your semantic model measures with Fabric Copilot for Power BI. People building reports from your semantic model can see the name and description of your measures, making the description property essential documentation.



You can streamline your semantic model documentation by creating measure descriptions with Copilot. To do this:

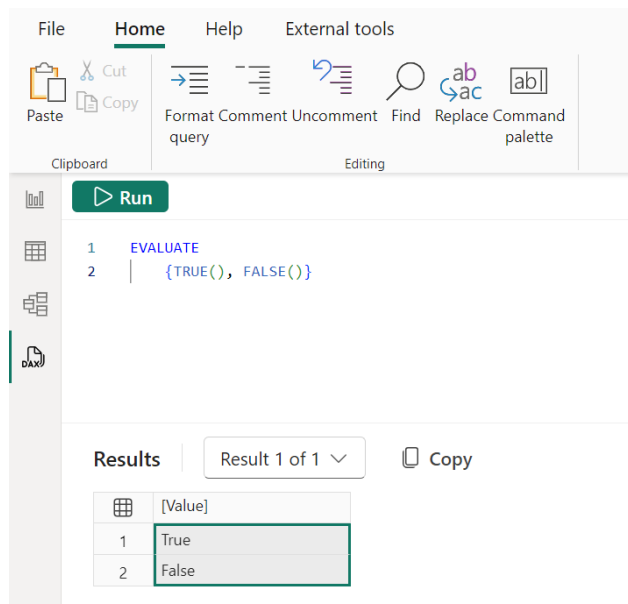
- Click on the model measure in the Data pane of Model view to see the measure properties
- Click on the 'Create with Copilot (preview)' button under the Description textbox
- Review the measure description from Copilot, then click 'Keep it'
- Now the measure description is in the Description box. Fine tune the description as needed
- You may update the measure later: simply click the button again when you need the description updated.

You may access this feature by turning on this public Preview feature in **Options -> Preview features**.

DAX query view improvements in Preview

You may recall Microsoft released the public Preview of DAX query view in November last year. In this release, the Power BI team has made the following improvements:

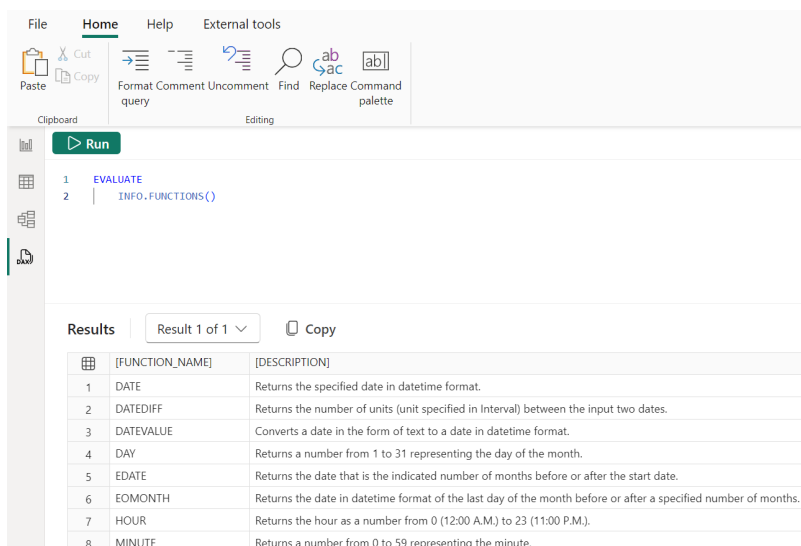
- Boolean values are now showing in the Results grid



- a share feedback link has been added in **Options -> Preview features**
- a bug causing the active query tab to stop being highlighted has been fixed
- a bug with the close brackets of a nested IFs DAX formula has also been fixed.

Additional **INFO DAX** functions have also been released into the wild:

- **INFO.CHANGEDPROPERTIES()**
- **INFO.EXCLUDEDARTIFACTS()**
- **INFO.FUNCTIONS()**
- **INFO.LINGUISTICMETADATA()**.



Certified connectors updates

SingleStore's connector has now officially moved out of beta. This upgrade comes with a new feature: you may now cancel running queries, hassle-free. No more queries running in the background after you refresh the UI / visual or navigate across the pages in the report.

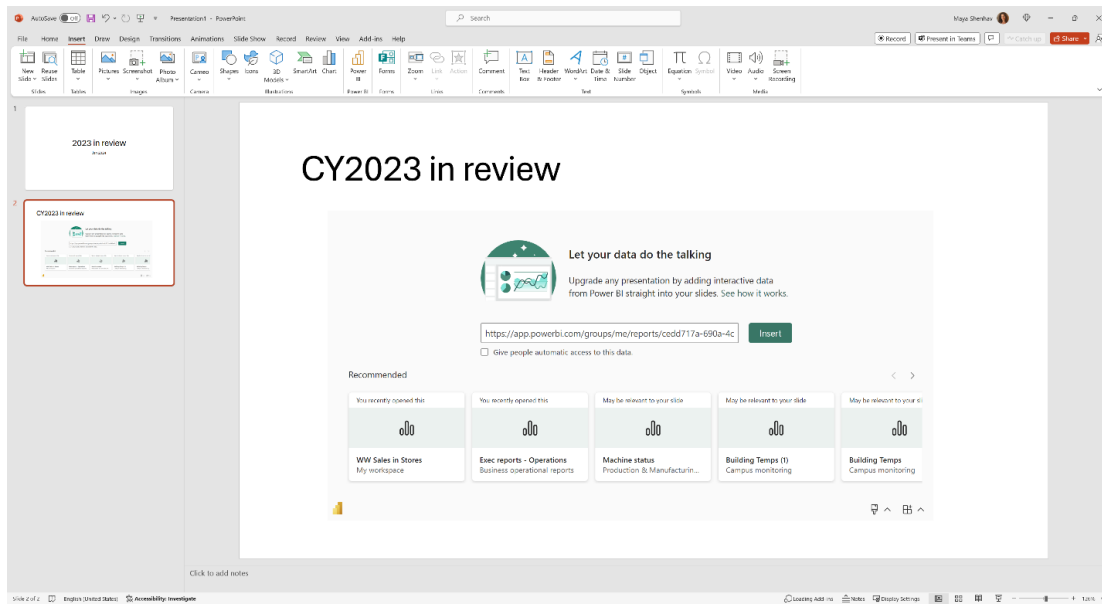
Storytelling in PowerPoint: replace report URLs with shareable links in Power BI add-in

When you add the Power BI add-in to a presentation, you can pick a report suggested to you or paste a link to a specific report.

When you paste a standard report link (the URL copied from the browser address bar), and if sharable links are enabled for your organisation and allowed for this report, you may re-share permissions to this report: the Power BI add-in can replace the link you pasted with a shareable link. In

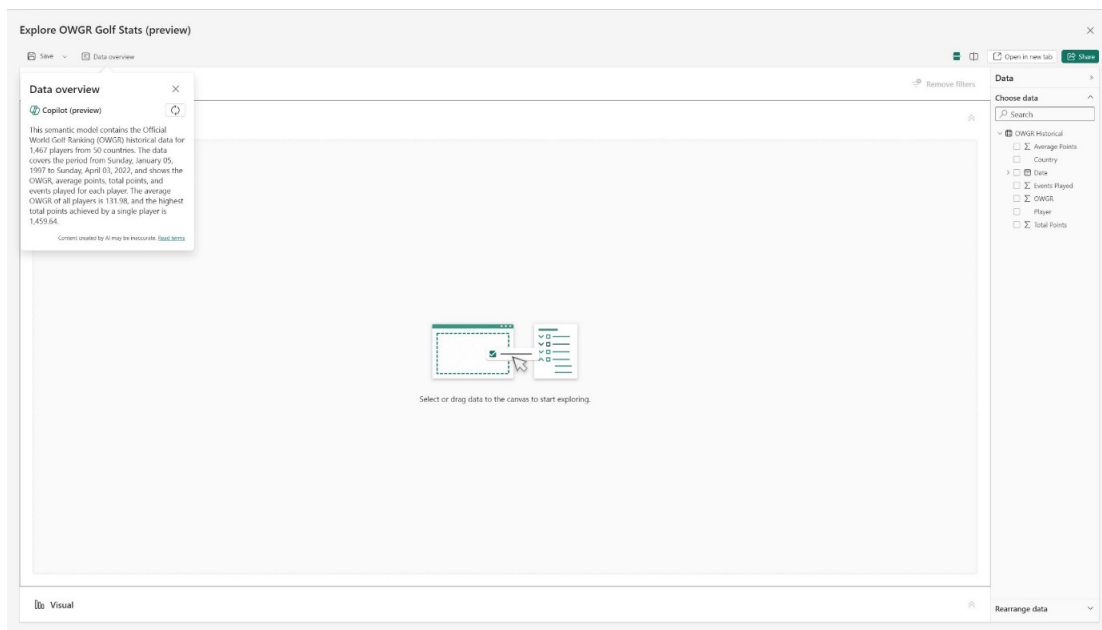
this case, you will see a checkbox added below the report URL that offers you automatic access to this report. Just mark this checkbox and Power BI add-in will create a shareable link for you.

Using a sharable link ensures that other users viewing the presentation have the required permission to see the report and do not need to request access when viewing the presentation.



Data overview in Explore

This month's update has added a data overview feature to Explore that allows you to get the "gist" of what your data reveals. Powered by Copilot, data overview gives you a summary of what's contained in the data you're exploring and highlights some interesting insights to get you started.



DirectQuery connections update

'Maximum connections per data source' is a setting to configure the maximum number of connections DirectQuery opens for each underlying data source. This controls the maximum number of queries that may be executed concurrently against each data source and is configurable per

semantic model. Microsoft has recently updated the upper limit of the numberofconcurrentDirectQueryconnectionsallowedpersemanticmodel. The updated limits for each SKU are listed in the table below:

Fabric SKUs	Power BI SKUs	Max concurrent DirectQuery connections (per semantic model)
F2	-	5
F4	-	5
F8	EM1 / A1	10
F16	EM2 / A2	10
F32	EM3 / A3	10
F64	P1 / A4	50
F128	P2 / A5	75
F256	P3 / A6	100
F512	P4 / A7	200
F1024	P5 / A8	200
F2048	-	200

The upper limit for Power BI PPU is 100 active connections. Note that there is no change to the Power BI Pro and Report Server limits and the default maximum value remains as 10 concurrent connections.

Shared device mode is now Generally Available

Introduced last September, shared device mode is now Generally Available. With shared device mode, organisations can safely deploy the Power BI mobile application across their pool of shared devices.

Many organisations maintain a pool of mobile devices that are shared between employees, often frontline workers, across tasks, shifts or locations. Most mobile apps, however, are designed for single users, and optimise their experience for use by a single user, with single sign-on (SSO) across applications and keeping users signed in on their device.

This behaviour isn't suitable for devices that are shared by multiple users. In the case of shared devices, employees expect to be able to pick out a device from the pool, "make it theirs" for the duration of their shift, and then, at the end of their shift, to be able to sign out from the device globally and have all their personal and company information removed so they can safely return the device to the pool.

Shared device mode, a feature of Microsoft Entra ID, enables this functionality. It enables single sign-on and device-wide sign-out and clears the user's data to prevent unauthorised or unintended access by the next user.

The Power BI mobile apps for iOS and Android devices support shared device mode. This means organisations can safely deploy the Power BI

mobile app across their pool of shared devices without risk of exposing confidential data and personal information to unauthorised access after users sign out and return their device to the pool.

For example, when an employee signs on to their app (*say*) at the start of their shift, they're automatically signed on to all other apps that support shared device mode on the device. At the end of their shift, when they sign out of the app, they're signed out globally from all other apps that support shared device mode. After sign-out, the employee's data and company data in the app and in all other apps that support shared device mode can no longer be accessed. The device is ready for the next employee and can be safely handed off.

Using the Power BI Mobile app in shared device mode is just like using the app in standard, single user mode, with a few minor differences to better suit the frontline worker scenario:

- the welcome experience and introductory tips are suppressed to enable an uninterrupted workflow for frontline workers
- notifications are generalised to exclude confidential and sensitive information such as specific data and personal information.

Power BI custom visuals new local storage API

This API allows Custom Visuals to store data directly in the local browser. Data stored locally is more secure and improves the performance of web apps. The API will be controlled by a global administrator setting.

On-object Interaction support for custom visuals

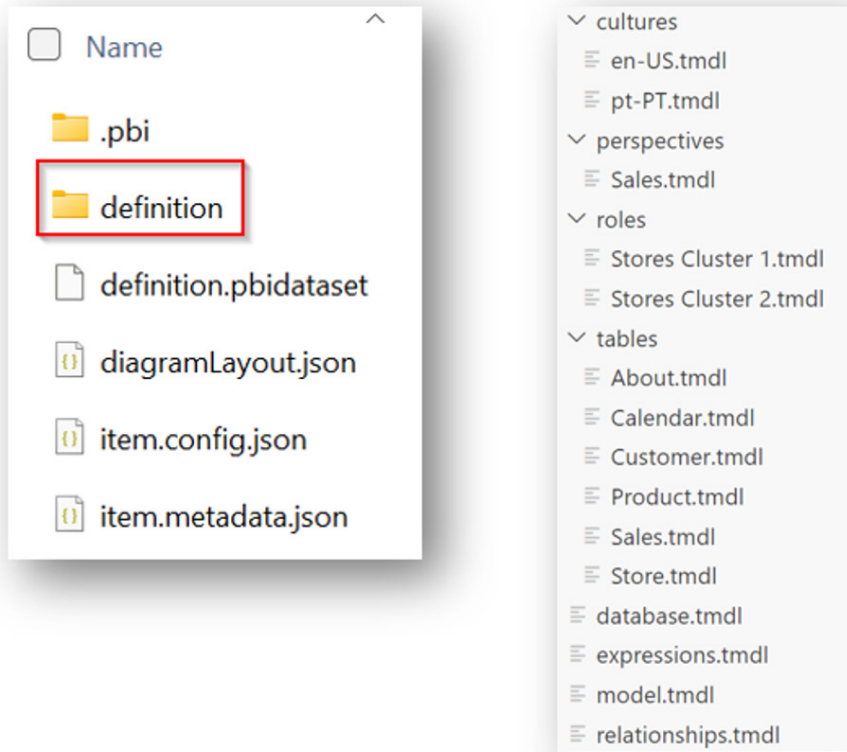
This update introduces the support of the new On-object Interaction. This enhancement allows users to build and customise visuals directly on the visual in Power BI Desktop. It puts common actions for creating and formatting visuals on the visuals themselves, actions such as adding fields, changing visualisation types and formatting text.

TMDL in Power BI Desktop Developer Mode

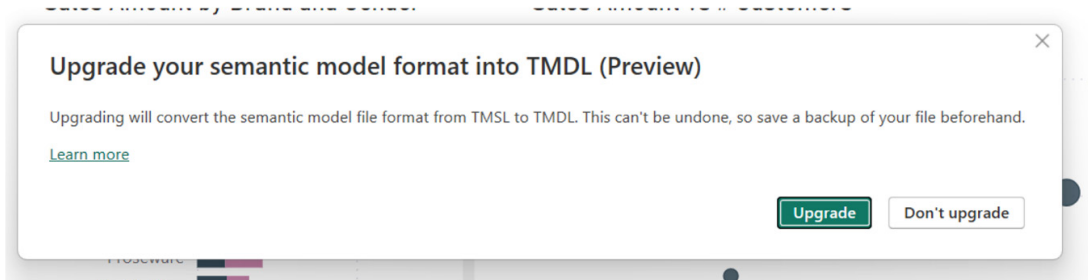
The primary objective of Power BI Desktop Developer mode is to provide a user-friendly source control and co-development experience. With this objective in mind, you can now save your Power BI Project files (PBIP) using Tabular Model Definition Language (TMDL) format. TMDL has been designed from the ground up to be human-friendly, facilitating not only readability but also easy editing in any text editor. This represents a substantial enhancement for source control and collaborative development experiences, particularly when dealing with complex file diffs (i.e. a line-by-line comparison experience).

Saving as a PBIP using TMDL is currently in Preview. Before giving it a try, you must first enable this feature in Preview features. Go to **File -> Options and settings -> Options -> Preview features** and check the box next to 'Store semantic model using TMDL format'.

After enabling the Preview feature, when saving as PBIP, your semantic model will be saved as a TMDL folder named 'definition' with separate files for each table, perspective, role and culture:



You can also upgrade existent PBIP files to TMDL, by just opening them and choosing 'Upgrade' when you save:



By default, Fabric Git Integration will still use Tabular Model Scripting Language (TMSL) to export the semantic model during the public Preview. However, if the semantic model is imported into Fabric using TMDL, then Fabric Git Integration will export the definition into Git using TMDL in the event of any semantic model changes in the Service.

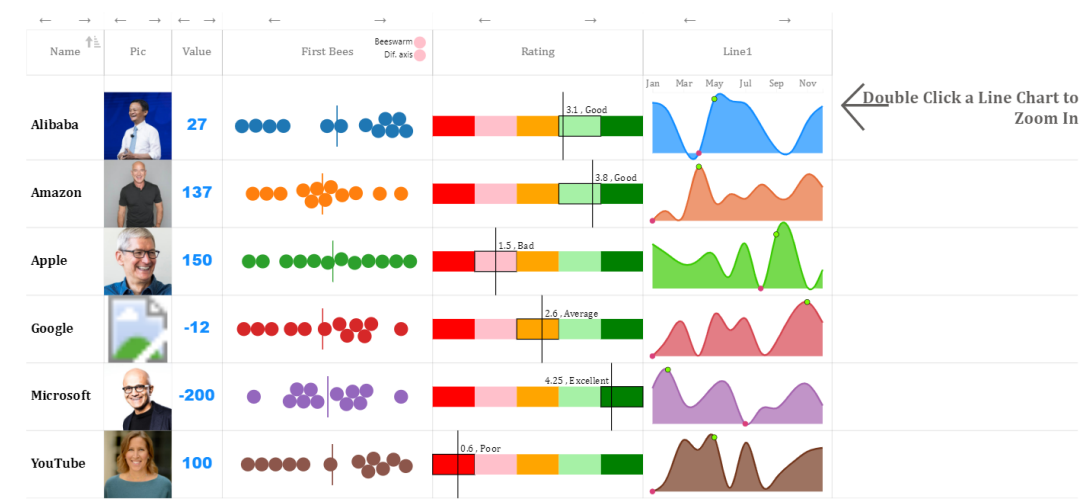
New visuals in AppSource

- Charticator Visual Community (View)
- Comment: Dynamics 365 Finance business performance planning
- Copy: Dynamics 365 Finance business performance planning
- Date Picker
- Deneb: Declarative Visualization in Power BI
- Drill Down Donut PRO (Filter) by ZoomCharts
- Enlighten Aquarium
- flashbi fantail
- Graphical planning: Dynamics 365 Finance business performance planning
- Icon Map
- Inforiver Analytics+ (Charts + Cards + Tables)
- Inforiver Premium Matrix / Table
- Map by Squillion
- Matrix planning: Dynamics 365 Finance business performance planning
- Processifier Process Mining
- Reporting: Dynamics 365 Finance business performance planning
- Table edit: Dynamics 365 Finance business performance planning
- UnAvoids
- Variance – Dynamics 365 Finance business performance planning
- Waterfall-Visual-Extended.

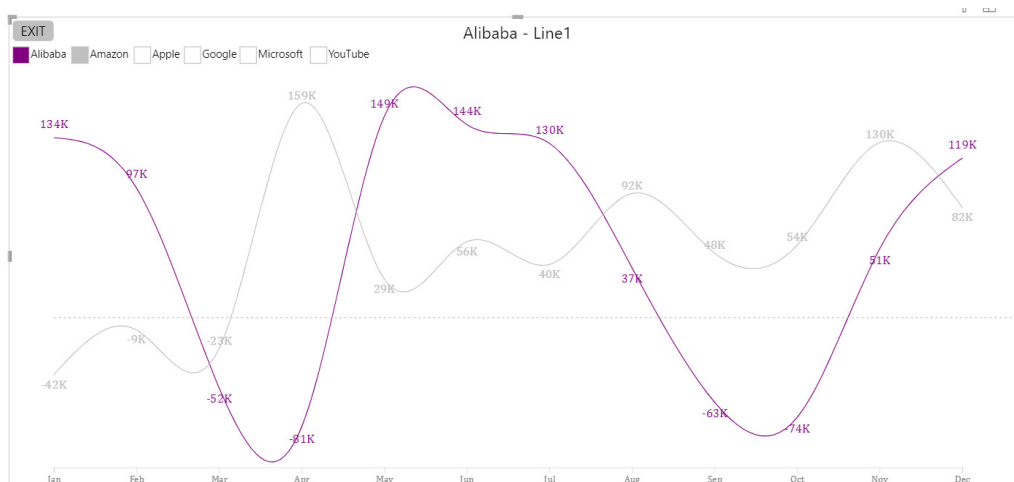
Multiple Sparklines

New features were added to Multiple Sparklines:

- when you double click a Line chart, it will zoom in to screen size of visual and you can then compare it with any other Line chart in that column
- you can use different colours for each Line chart in a field / column
- you can insert ratings with bands
- you can add beeswarm / distribution microcharts.



Once you double click the Line chart, it zooms in to the visual size. You can then compare it with another line chart in the same column. This is shown below:



You can download this visual from AppSource.

Rose Donut Pie Chart by Powerviz

The Rose / Donut / Pie chart is a powerful visual that lets you build four types of charts:


1. a rose
2. a rose do(ugh)nut
3. a do(ugh)nut
4. a pie chart.

These chart types are commonly used to display part-to-whole relationships, proportions of categorical data and ratios. Each arc represents the ratio from the total for easy comparison.


Key features:

- **Chart Options:** rose, donut, pie charts with style customisation
- **Data Colours:** choose from 30+ palettes, including colour-blind mode
- **Fill Patterns:** apply patterns or use custom images
- **Smart Labels:** improve readability with data and leaf labels
- **Arc Customisation:** easily adjust arc radius, padding and stroke
- **Ranking:** filter Top / Bottom N, show others intelligently
- **Center Circle:** multiple layers, text, icons and images in the centre
- **Mouseover Text:** display dynamic details when hovering over arcs
- **Image Labels:** integrate dynamic image URLs for enhanced visuals
- **Conditional Formatting:** detect outliers and set smart rules for measures / categories.

Other features included are annotation, grid view, show condition and accessibility support.




Rose/Donut/Pie Chart


**Powerviz**
www.powerviz.ai


INTRODUCING ROSE DONUT PIE CHART BY POWERVIZ


An advanced visual to show the categorical data in an intuitive way, making comparisons easy.

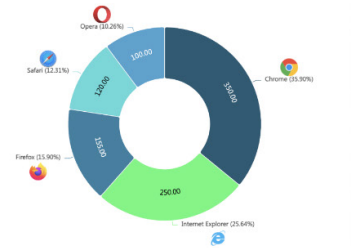
Four Different Chart Types


**Donut**


**Rose Pie**


**Rose Donut**

**Pie**

**Smart Labels**

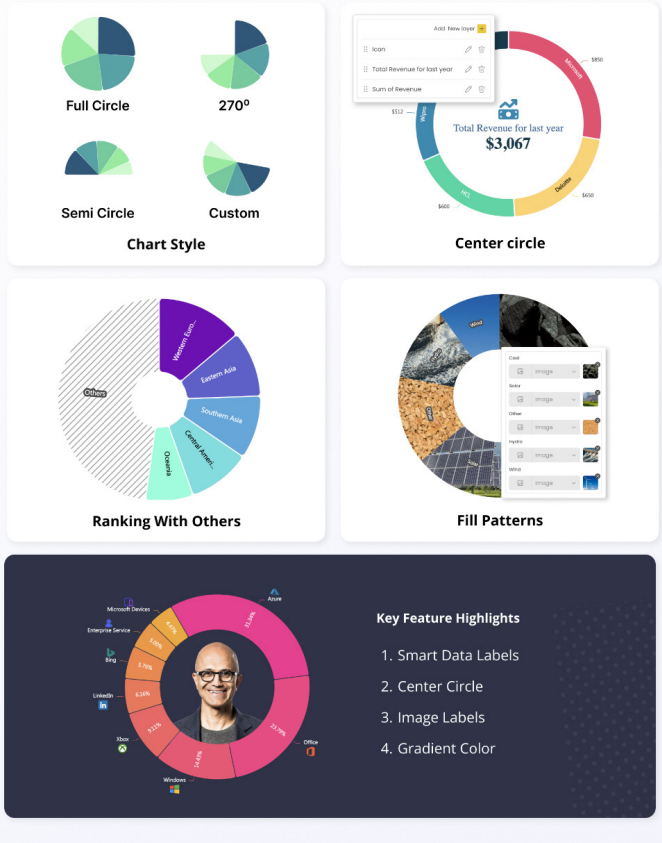
**Image Labels**

**Data Colors**

**Conditional Formatting**

INTRODUCING ROSE DONUT PIE CHART BY POWERVIZ

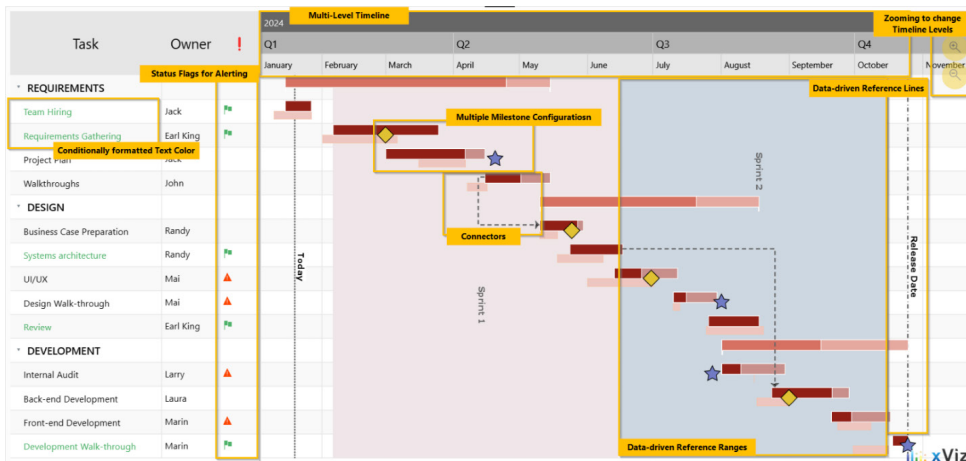
Four in one chart, Smart labels, Image labels, Mouseover text, 30+ Palettes, Conditional formatting & More.



xViz Gantt Chart by Lumel

xViz Gantt Chart by Lumel is a Microsoft Power BI Certified Visual. Features include:

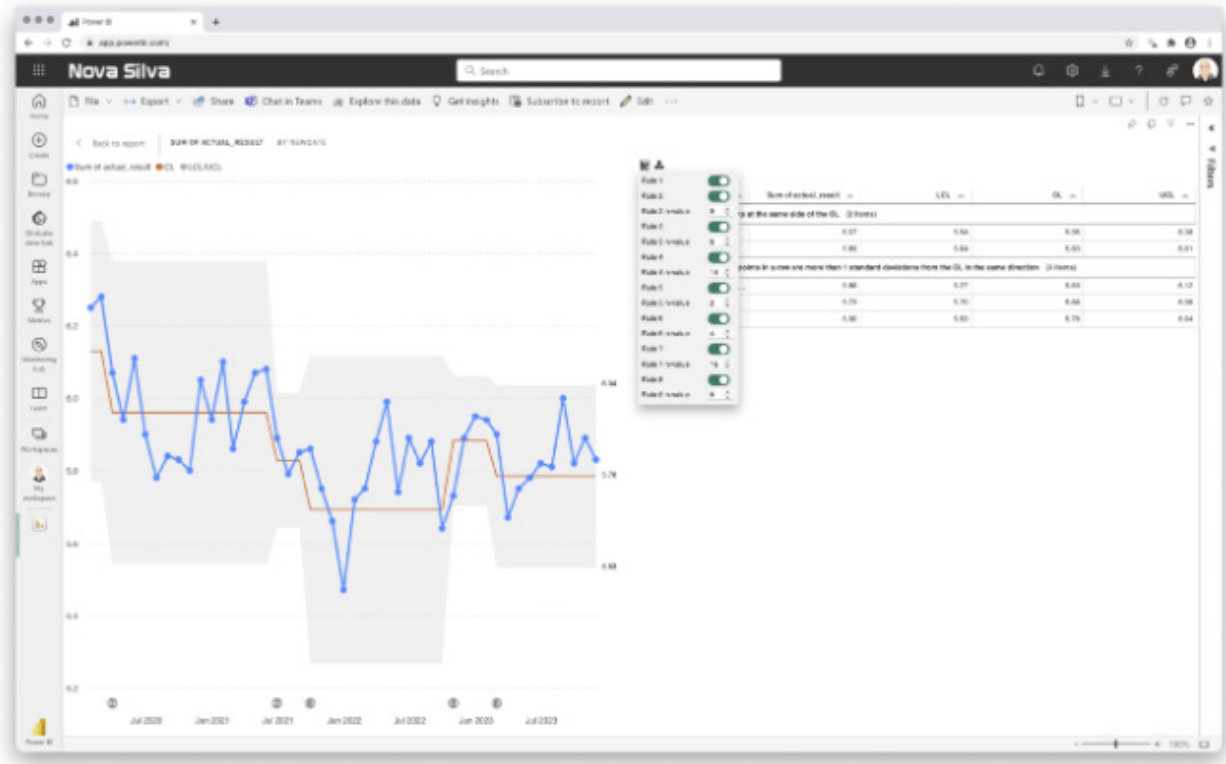
- **Real-time Alerts for Project Managers:** leverage conditional formatting to receive colour-coded alerts and status flags, ensuring timely awareness of schedule delays or progress issues
- **Visualise Task Dependencies:** identify causes of delays with the ability to plot task dependencies using connectors within the roadmap view
- **Adaptable for Different Users:** from stakeholders tracking yearly progress to Project Managers analysing monthly views and developers scrutinising smaller time grains with flexibility across three [3] distinct timeline levels
- **Strategic Planning with Reference Lines and Ranges:** utilize Reference Lines and Ranges to mark crucial dates, holidays, sprints or deadlines across projects
- **Customization Galore:** wide range of customisable options, including adjustable timeline limits, selectable week start days and indentation customisation for ragged hierarchies
- **Hassle-Free Licensing:** the visual is free for use in Power BI Desktop. For sharing and collaborating on Power BI Service, the licenses can be purchased directly from Microsoft AppSource.



Control Chart XmR by Nova Silva

Control charts allow users to split their temporal data in two: random noise and real signals. Users are looking for real signals; they don't wish to be distracted by random noise. In this latest release of the Control Chart XmR, several new features have been added to make it easier to find real signals and ignore random noise in your data.

Nova Silva has added a feature to allow any report consumer to override the applied rules. This allows you to analyse the effects of one specific rule or set of rules.



Some have asked for the possibility to download the calculated values from the visual. Now, you can download all values calculated by the Control Chart XmR, such as LCL, CL, UCL, sigmas and signals.

Control Chart XmR is available from the AppSource. All features are available for free to evaluate this visual within Power BI Desktop.

Drill Down Graph PRO

Drill Down Graph PRO lets you create user-friendly graphs to represent complex relationships between nodes. It's useful for both small and large network graphs and offers advanced features like cross-chart filtering and vast customization options. You can create hierarchies and explore them using this visual's interactions.

Main features include:

- **Multiple layout options:** dynamic, hierarchical and radial
- **Focus nodes mode:** for gradual exploration of graphs
- **Customisation options:** choose colours, shapes, images and labels
- **Bidirectional links:** show reciprocal relationships between nodes
- **Touch device support:** explore your data anywhere.

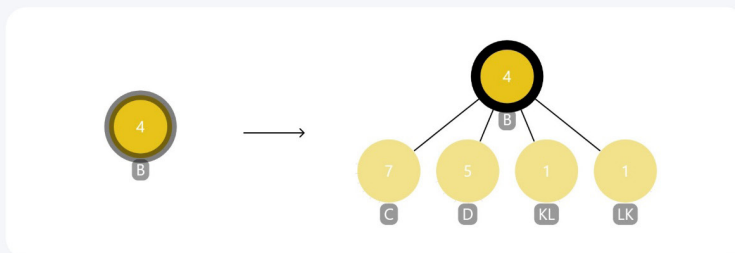
Popular use cases:

- **IT:** asset management, IT infrastructure, IoT monitoring
- **Logistics:** fleet management, stock management, parcel tracking
- **Sales and Marketing:** community detection, account management, web analytics.

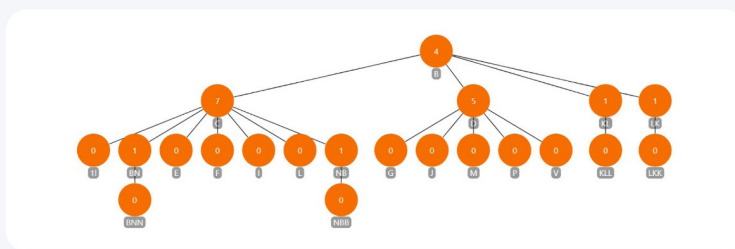
Again, this may be downloaded from AppSource.



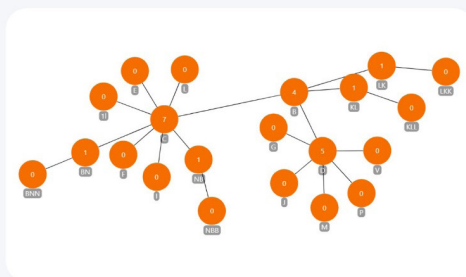
Focus nodes



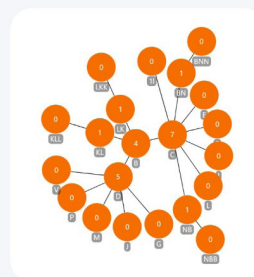
Layout - Hierarchy mode



Layout - Dynamic mode



Layout - Radial mode



zoomcharts.com/powerbi

More users can now collaborate with protected PBIX in Power BI Desktop

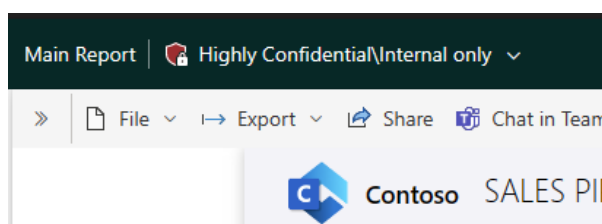
If you have ever wondered how to collaborate with your colleagues on sensitive data without compromising its security or wish to learn how to use Microsoft Purview Information Protection sensitivity labels to protect your data, then keep reading.

Purview allows you to use sensitivity labels with protection to encrypt and protect your data, and how to enable more users to edit and republish encrypted PBIX files.

Compliance administrators in your organisation can use Microsoft Purview Information Protection sensitivity labels to manage their organisation's sensitive data across different applications and services, and meet regulatory and compliance requirements.

They define file protection policies for the sensitivity labels, which result in files being encrypted when such labels are applied, allowing only authorised users to open and edit these files in Office apps and Power BI Desktop.

Sensitivity labels are widely adopted by enterprises today and used to label and protect content in Microsoft 365 apps such as Excel, PowerPoint, Word and Outlook, and now in Power BI and Fabric as well. All you have to do is enable Information Protection in Microsoft Fabric's Admin Portal and let the labels do the rest.



In Power BI Desktop, Microsoft enforces label protection on PBIX files. To open a PBIX file, you either must be the label issuer or have Full control and / or Export usage rights for the relevant label.

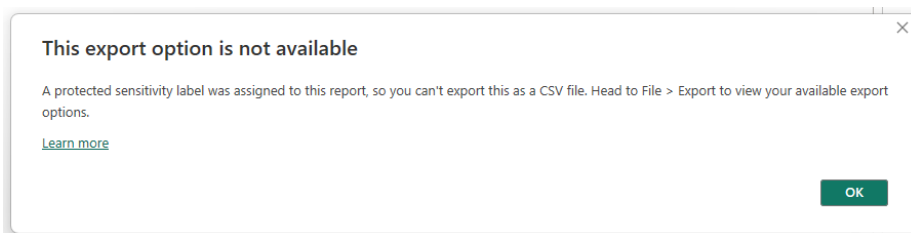
These usage rights are elevated permissions, as they grant permission to change the sensitivity label. Since Power BI and Office applications use the same label policies, compliance administrators may prefer not to grant these usage rights for 'Highly confidential' labels. This might block you from collaborating with your colleagues when you're sharing or downloading Power BI reports and trying to open them in the desktop app.

By enabling 'Increase the number of users who can edit and republish encrypted PBIX files (preview)' in your tenant, users that have been assigned with all of the following usage rights should be able to open, edit and republish the protected PBIX file to the Power BI Service:

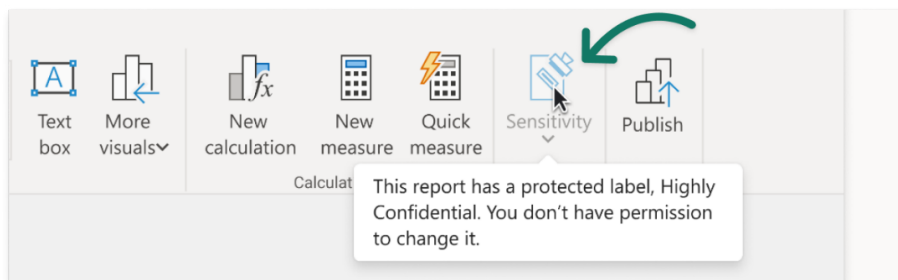
- View Content (VIEW)
- Edit Content (DOCEEDIT)
- Save (EDIT)
- Copy and extract content (EXTRACT)
- Allow Macros (OBJMODEL).

Therefore, users who were once restricted can now collaborate with protected files, while keeping protection consistent with the organisational policy. Having said that, it should be noted that these usage rights are a sub-set of the 'Co-Author' permissions preset in Microsoft Purview compliance center (sic).

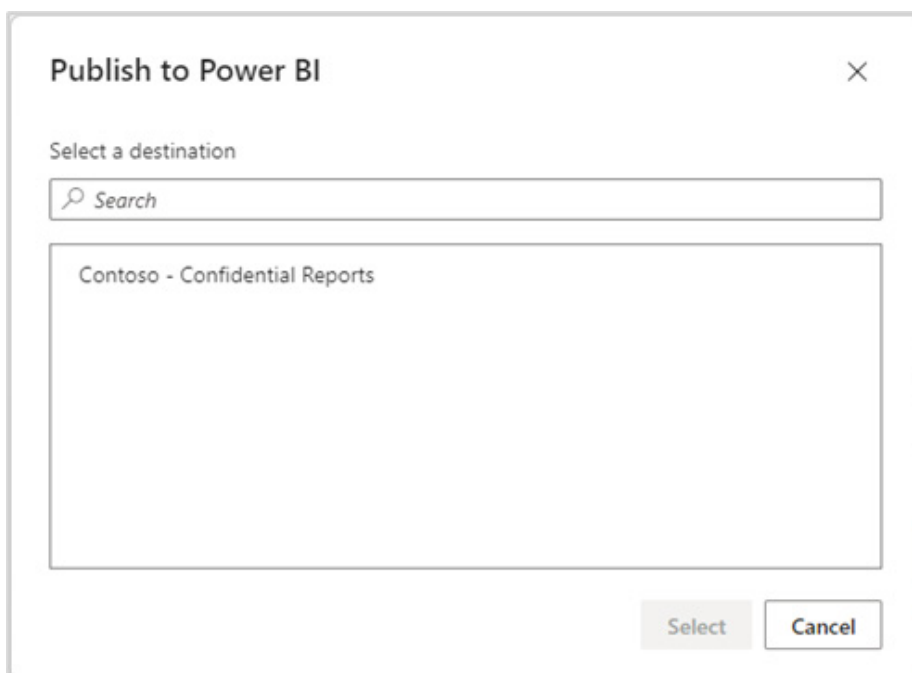
In order to align with compliance requirements, users with these usage rights are lightly restricted while editing a protected PBIX file. There is no exporting to unsupported formats: the user won't be able to export to formats that don't support sensitivity labels, such as CSV files.



No label changes are allowed. The user can't change the label on the PBIX file.



You may republish to the original workspace only:



To remain compliant, Microsoft prevents users from gaining more permissions, including Power BI permissions (i.e. Read, Write, Reshare and Build). This means that a user who wishes to publish should not be able to publish to a workspace that might grant them additional permissions through Workspace roles.

Additionally, this feature is meant for collaborating and sharing items that are more restricted than usual, and confidential data is usually managed in a dedicated workspace. This restriction will prevent users from publishing confidential data across the tenant.

It should be noted that the file must be published at least once for other users to be able to republish it to that specific workspace. If the file has not yet been published, then the latest label issuer (the one who set the protected label) or a user with sufficient usage rights must publish it and then share the file with the other editors.

To enable it, there is a prerequisite: the compliance administrator must assign you and your colleagues the proper permissions for that sensitivity label. Then, Fabric / Power BI administrators must enable the feature in **Admin Portal -> Information protection -> Increase the number of users who can edit and republish encrypted PBIX files (preview)**.

In Power BI Desktop, users who would like to open and edit protected PBIX files must enable the feature by opening **File -> Options and settings -> Options -> Preview feature -> Less elevated user support**.

To conclude, with this new feature, users can now collaborate more easily with other users when working on confidential data in Power BI Desktop, without any loss of protection along the way.

More next month.

New Features for Excel

This month's updates see users able to synchronise form data to Excel for the web. 'Insert pictures into cells' is available to users of Excel on Windows and Mac, and PivotTables now has expanded functionality in Excel for iPad.

You can check out the full list here:

Excel for the web

- Synchronise Forms Data to Excel
- Date Picker

Excel for Windows

- Insert Pictures in Cells
- Check Performance

Excel for Mac

- Insert Pictures in Cells

Excel for iPad

- PivotTables on iPad.

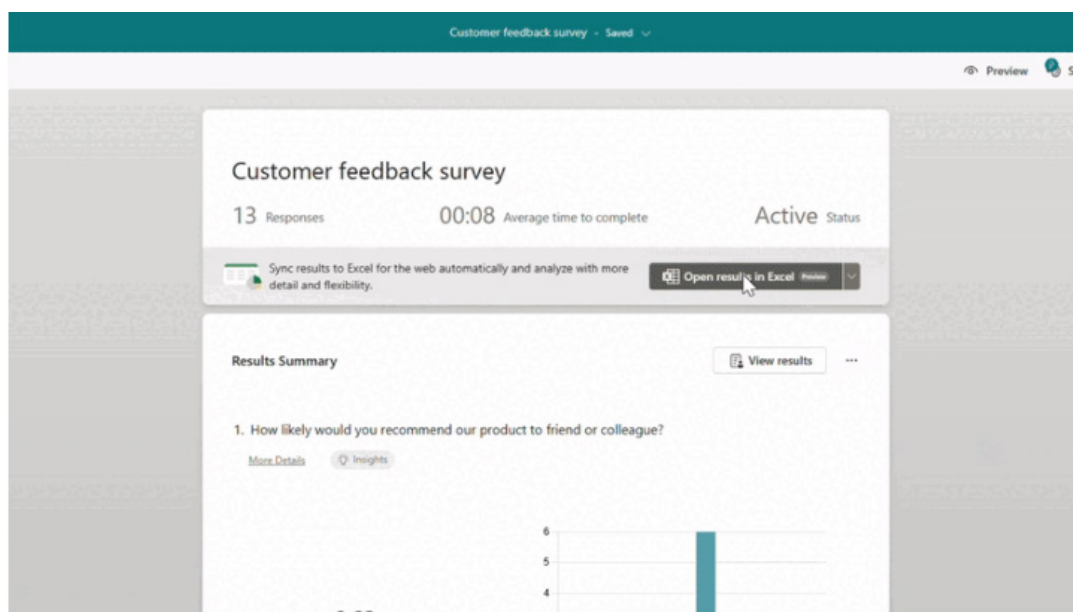
Let's get started.

Synchronise Forms Data to Excel

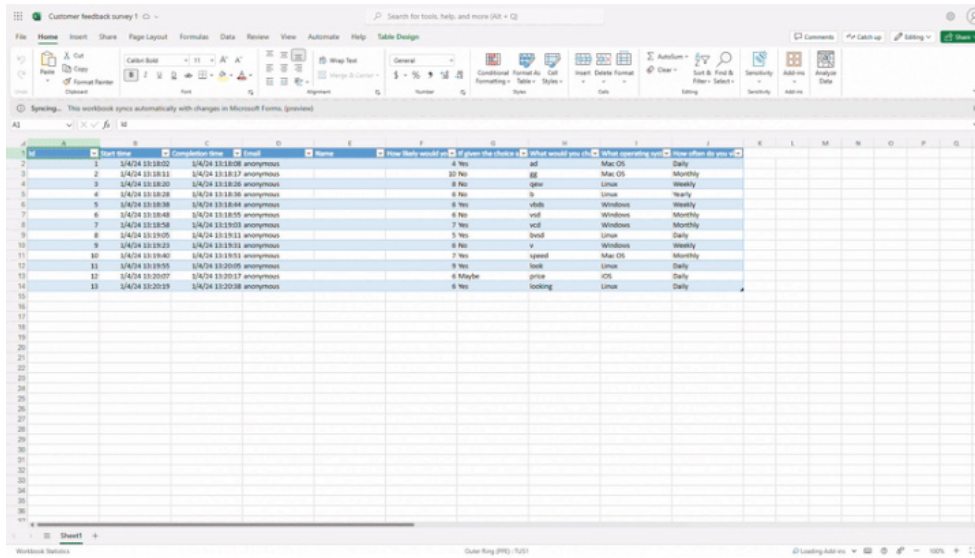
The Forms data synchronisation to Excel (for the web) is now available in Microsoft Forms. With just one click, you can now easily access all your form responses in Excel for the web and take advantage of Excel's functions and features to analyse and visualise your data. With automatic synchronisation of new responses in real-time, you can keep working on your existing spreadsheet as it updates. There is no need to

juggle multiple copies of the same file, making it easier than ever to stay on top of important data.

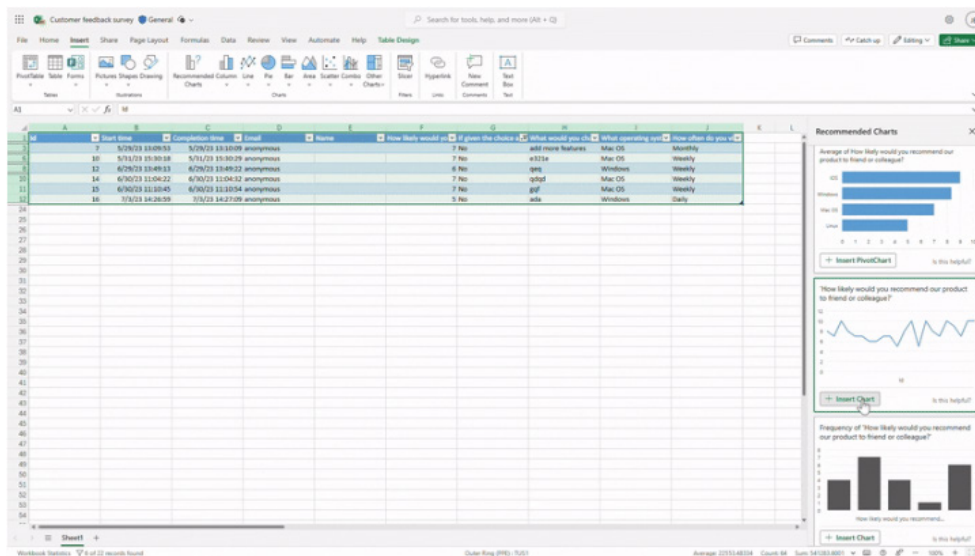
To activate, open any active form or quiz, go to the 'Responses' page and click 'Open results in Excel'.



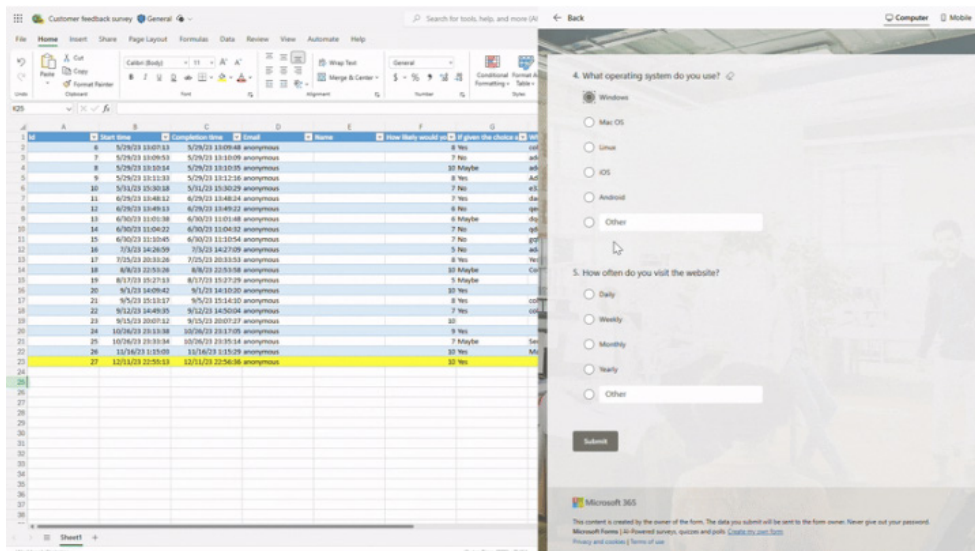
This will automatically create an Excel worksheet in your OneDrive and take you to Excel for the web, where you'll see all your data in just a few seconds.

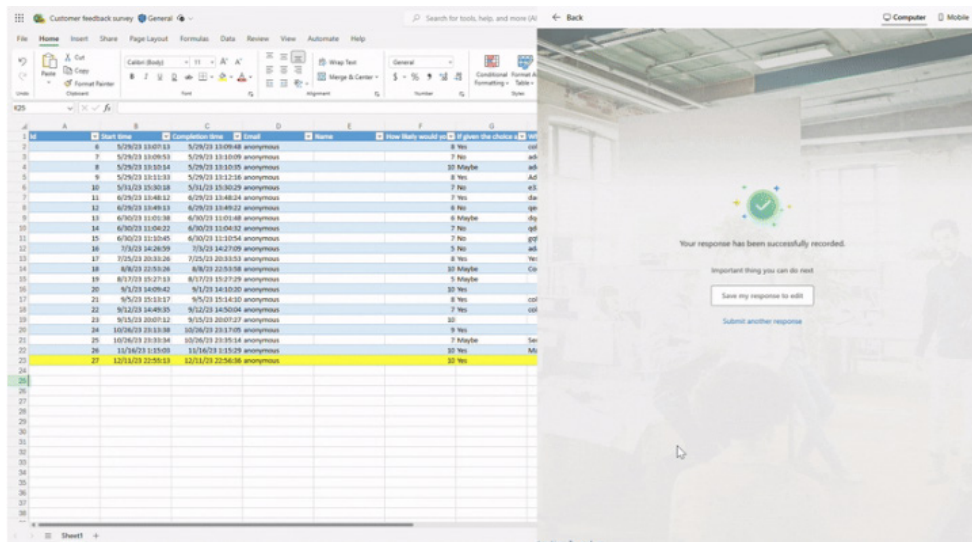


In Excel, you can manipulate the data the way you want and leverage various tools to do further analysis, like sorting, filtering and inserting charts.



New responses will automatically synchronise when you open your Excel file for the web, allowing you to seamlessly continue your existing work.



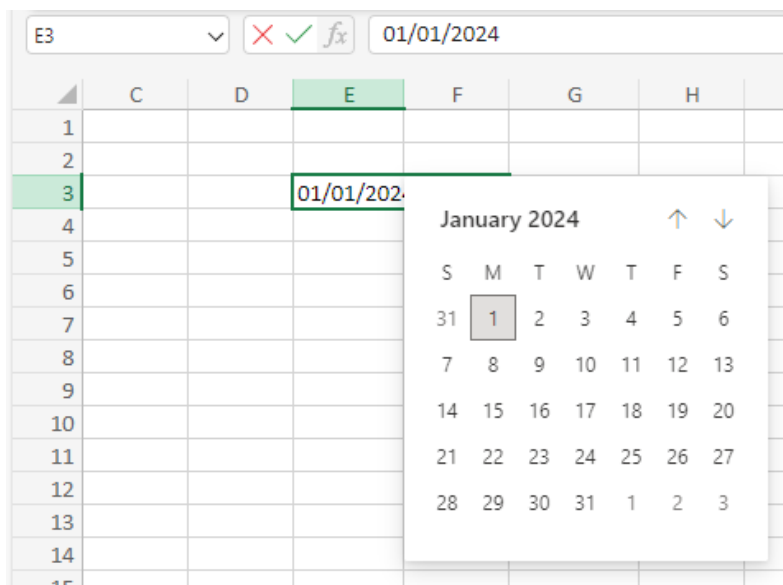


With Forms, multiple classes can use one quiz, and the results will be synced to a single worksheet, eliminating the need for teachers to manually combine quiz results from different classes. The teacher can add a question for class identification and use the filter in Excel to easily distinguish one class from another.

Please note that Forms data synchronisation to Excel is currently only available in Excel for the web. While you can still view your data with the desktop and mobile apps, new responses will not be automatically synchronised unless you are using Excel for the web.

Date Picker

The Date Picker feature in Excel for the web allows you to quickly insert a date from a calendar within a cell. This tool also enables you to display preset dates within your Excel worksheets. You can activate the Date Picker pop-up by either double-clicking in a cell or entering a date manually.



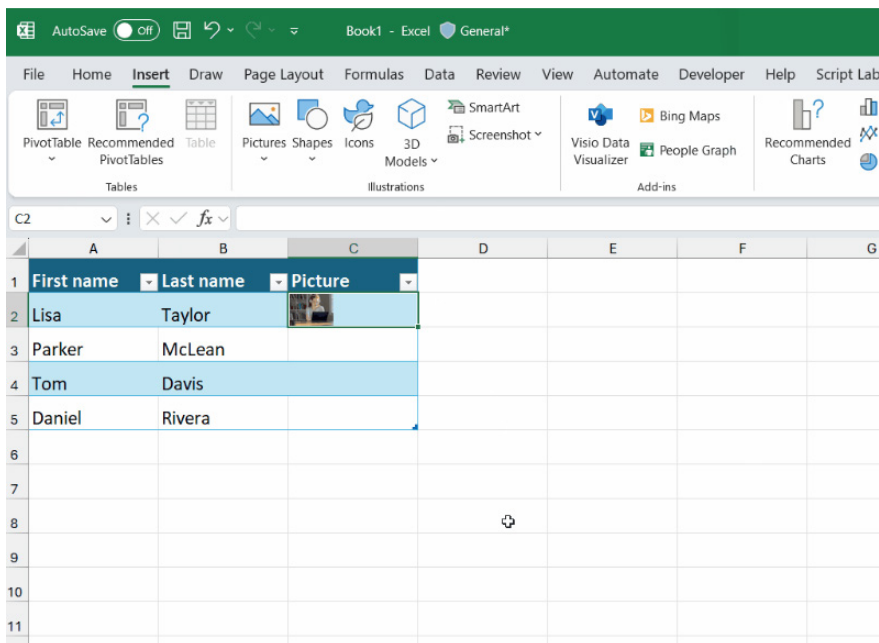
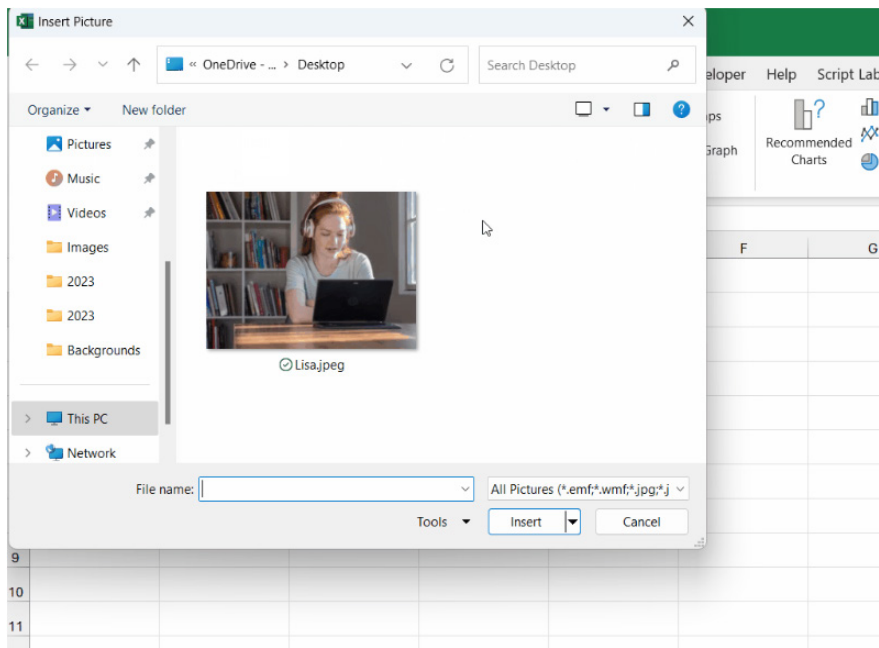
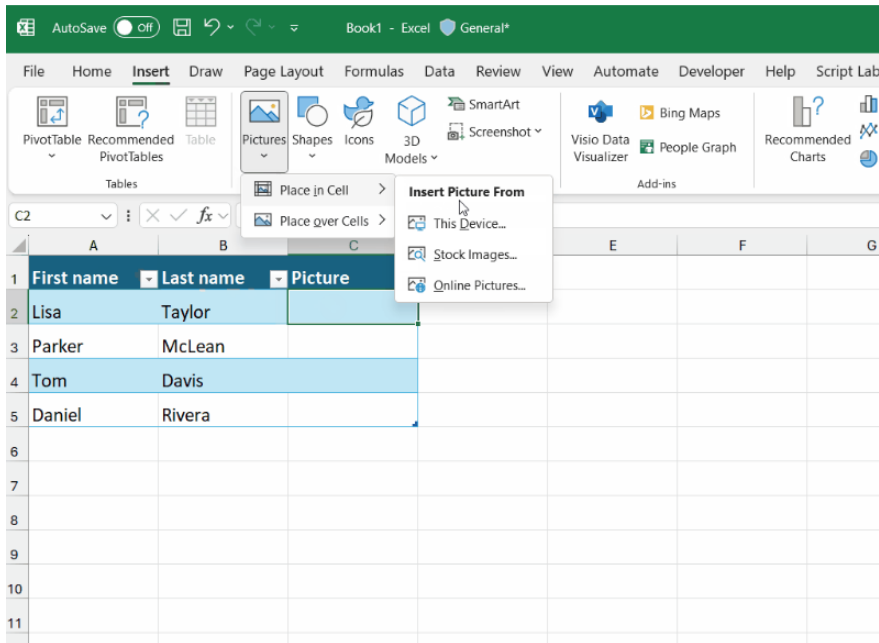
Insert Pictures in Cells

Whilst you've been able to use pictures in Excel for many years, Microsoft has only recently enabled the ability for a picture to become the actual cell value in Excel for Windows and Excel for Mac. It remains attached to the data even when the sheet's layout is modified. You can use it in tables, Tables, sort, filter, include it in formulae, and much more!

The last 12 months has seen the introduction of the new **IMAGE** function, which allows you to place a web picture in cell using a URL. Now, Microsoft has implemented the ability to insert local pictures, directly into cells, from your device or from the stock image libraries.

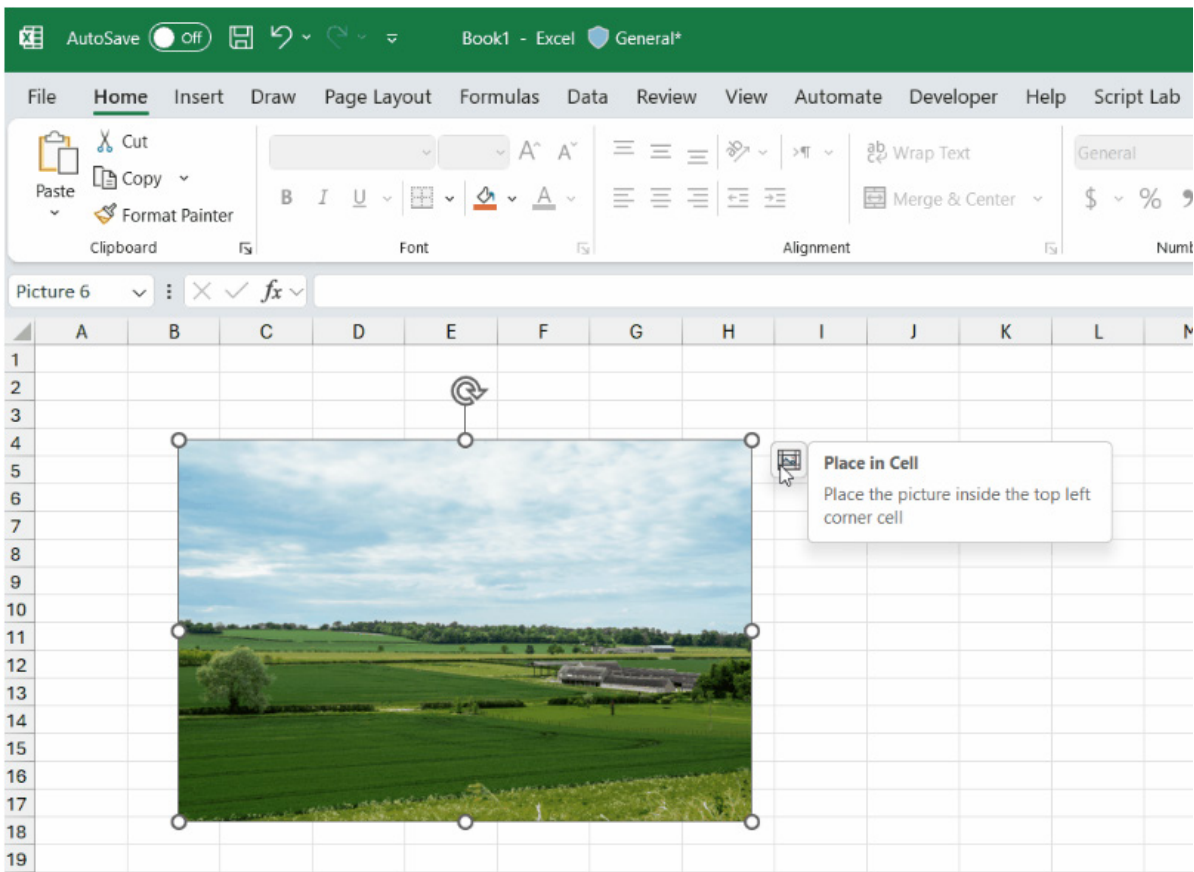
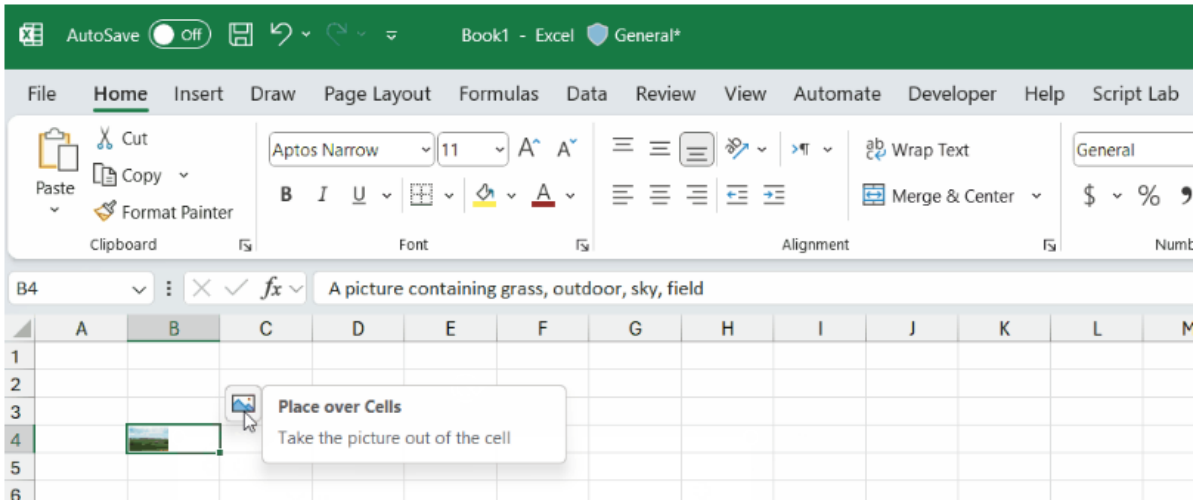
To insert pictures into cells:

- select the cell where you want to place the picture
- select **Insert -> Pictures -> Place in Cell**, and then select one of the available sources:
 - This Device
 - Stock Images
 - Online Pictures

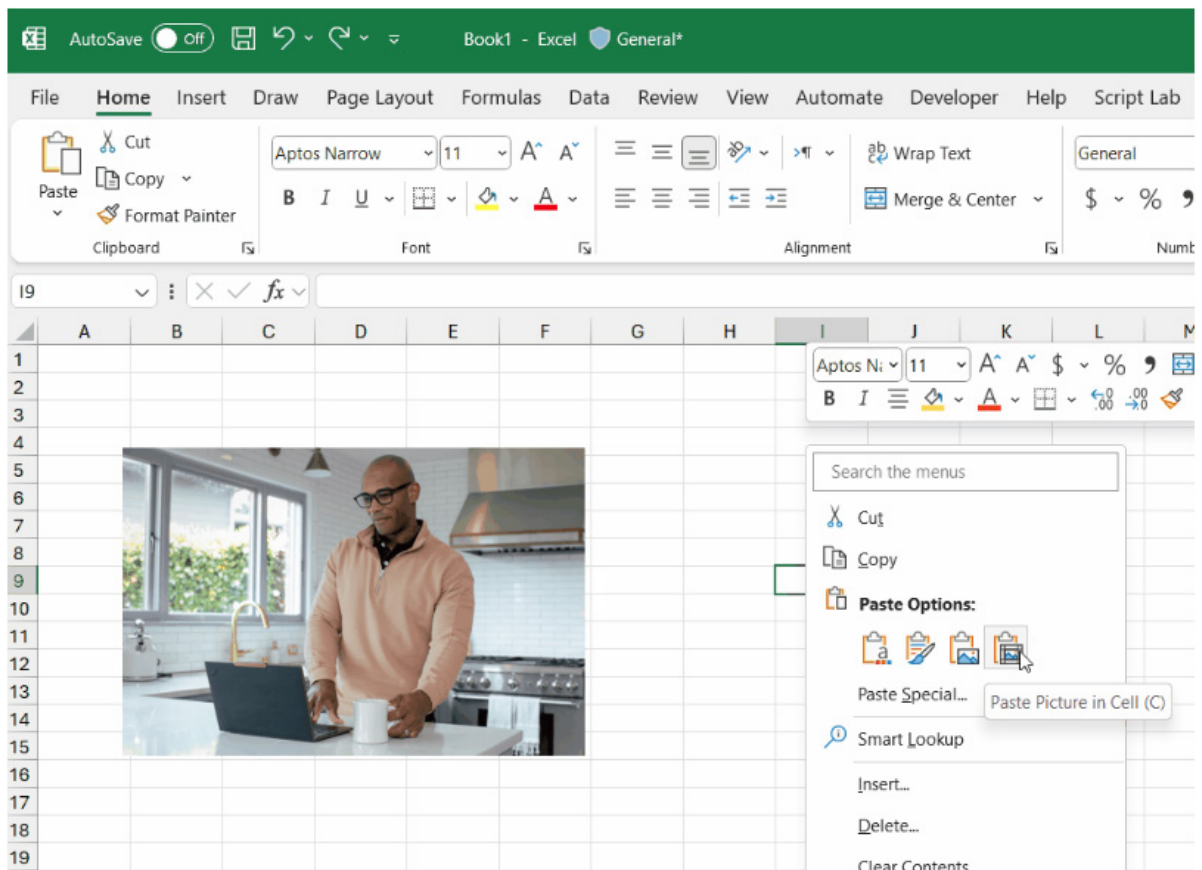
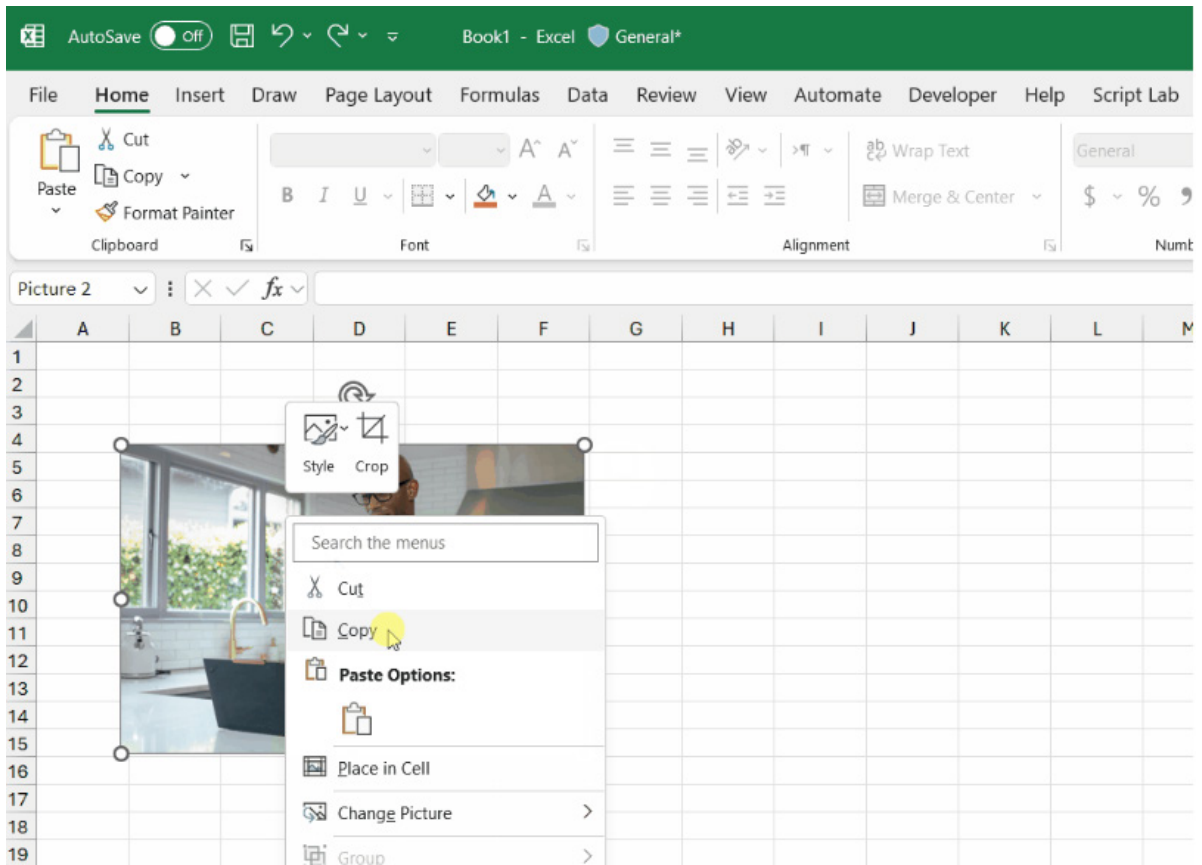


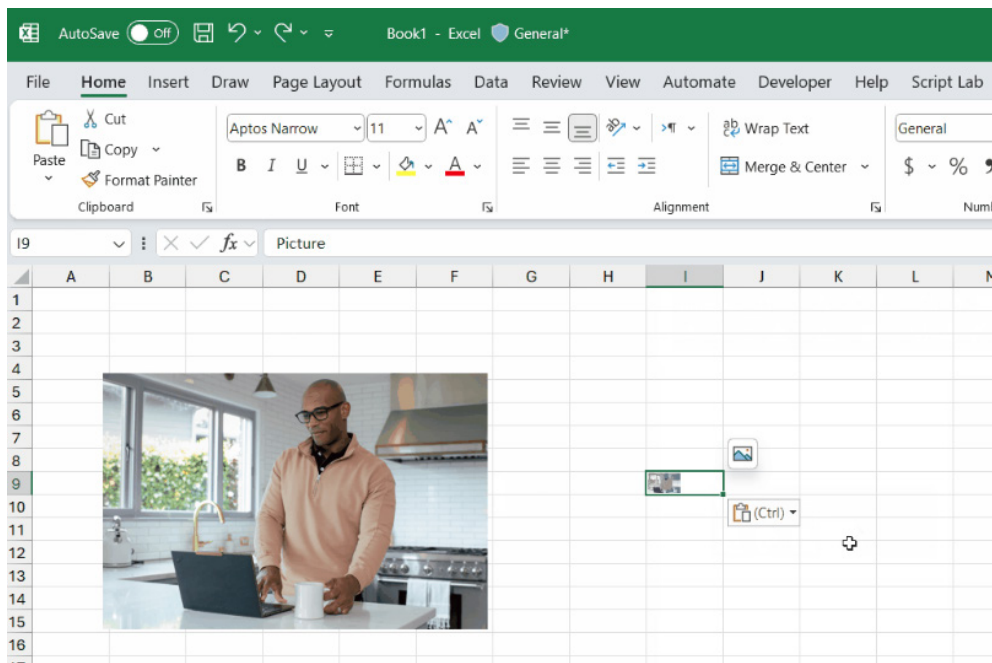
Be careful though. To insert pictures and have them “float” on top of your worksheet, like you used to do before, select **Insert -> Pictures -> Place over Cells** instead.

If you have an existing worksheet that contains a picture that was inserted using the ‘Place over Cells’ command, and you want it to become part of a cell, select the picture and click the ‘Place in Cell’ button that appears next to the picture. You can also find the ‘Place in Cell’ command in the right-click context menu or in the ‘Picture Format’ contextual tab.



To directly paste a picture from the clipboard into a cell, select **Paste -> Paste Picture in Cell** on the Home tab or in the right-click context menu.





You should note that when you use the default Paste command on the Home tab or the context menu, or press **CTRL + V** to paste a picture, it will be pasted over cells for now – not in a cell.

These capabilities are now available to Insiders running:

- Windows: Version 2306 (Build 16529.20000)
- Mac: Version 16.75 (Build 23070901)

These capabilities will become available soon in Excel for the web as well as on Excel for iOS and Excel for Android.

Check Performance

Originally added to Excel for the web back in October last year, 'Check Performance' has now come to Excel for Windows. Anyone who uses Excel regularly will be familiar with the dreaded file bloating phenomenon, where file sizes suddenly increase drastically and / or unexpectedly. At long last, the Excel Performance team has created a new capability to detect and remove unwanted size bloat and speed up such workbooks.

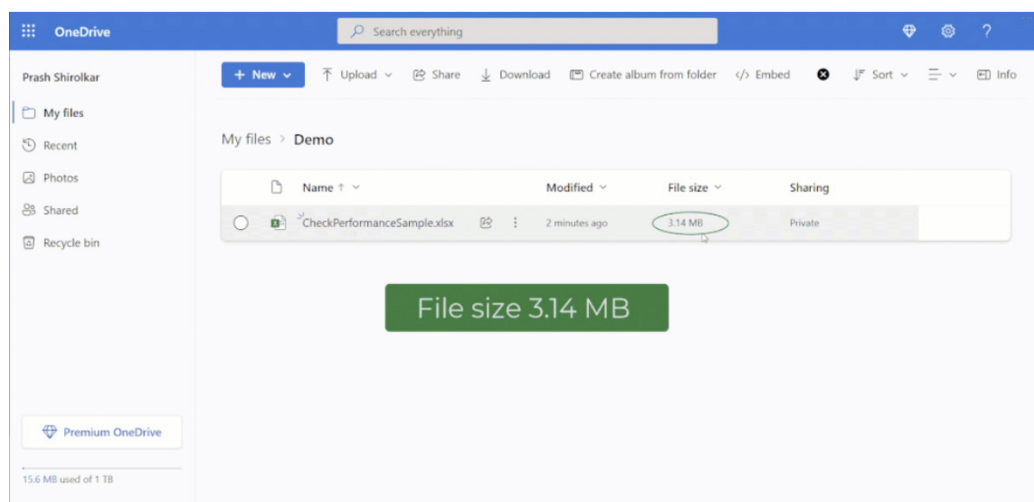
Often, a workbook can collect cells that have no data but still contain hidden information of little or no use anymore. These cells may have had data and formatting to start with, however now they do not have any data, but still take up space because they contain formatting. Consequently, too many of these cells can cause your workbook to slow down or become unresponsive. As a consequence, this update enables you to detect and remove these cells slowing down your workbooks, with 'Check Performance'.

When you open your workbook, Excel now detects whether your workbook contains too many of these unwanted formatted cells. If it does, Excel shows a "Business bar" to launch the 'Check Performance' feature. This may be launched manually from **Review -> Check Performance** too.

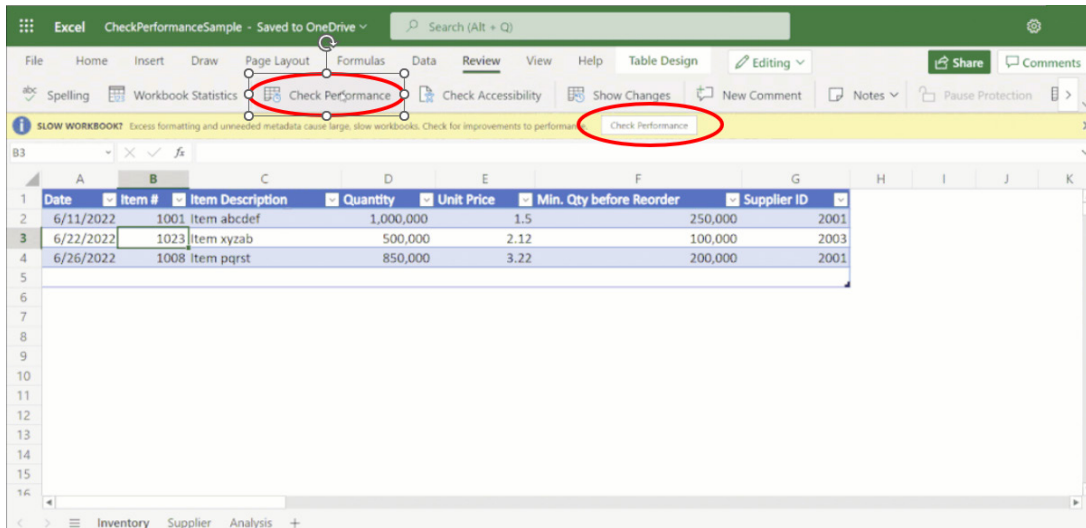
Once launched, there are two ways to remove these cells:

1. by navigating to a sheet in the task pane to review each range of these cells to optimise, and then pressing the 'Optimize Sheet' button; *or*
2. by pressing the 'Optimize All' button to remove all unwanted cells from all sheets in the workbook.

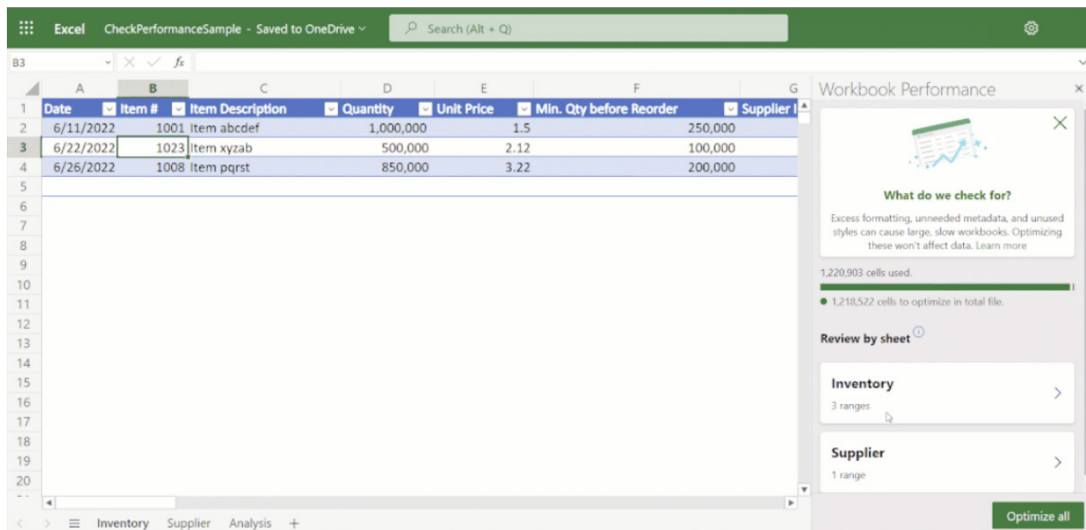
Consider the following example, albeit in Excel for the web. Here, we have a file that has a current size of 3.14MB – not too large, but as it turns out, much larger than it *should* be.



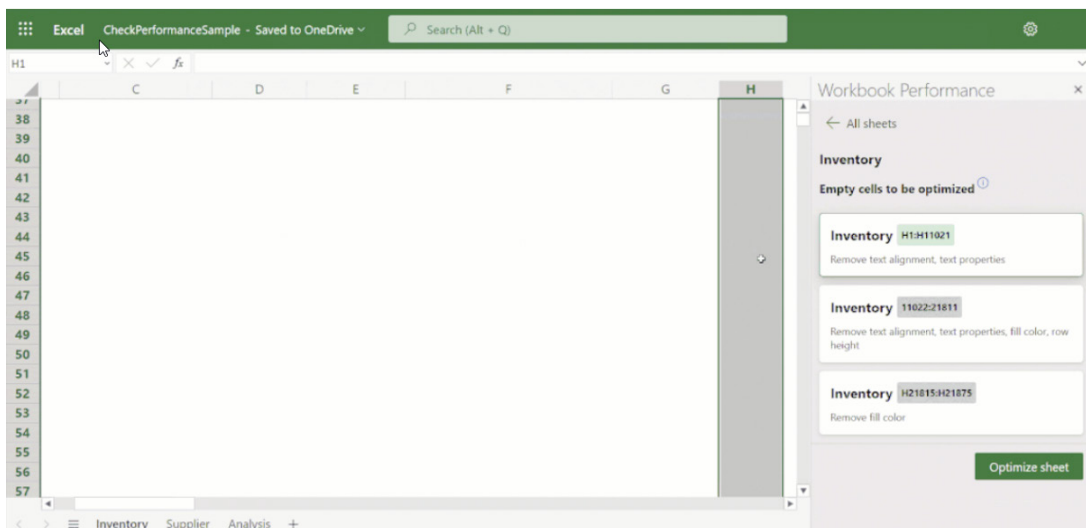
Upon opening, the Business bar (yellow bar) highlights the slow workbook and prompts you to 'Check Performance'. This may also be accessed from the Review tab, as displayed below:



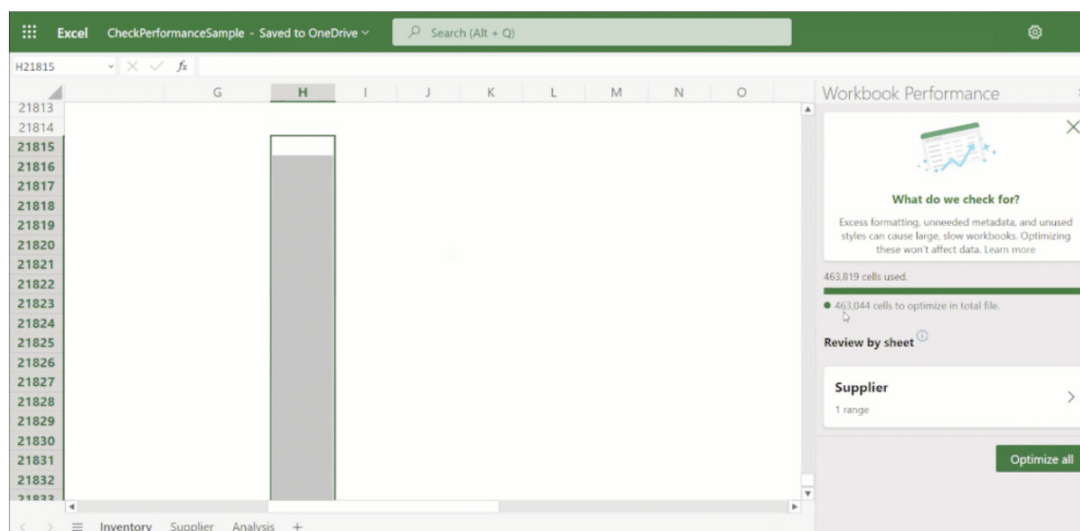
The 'Workbook Performance' pane appears and highlights two of the three worksheets may require review, citing a total of four ranges, viz.



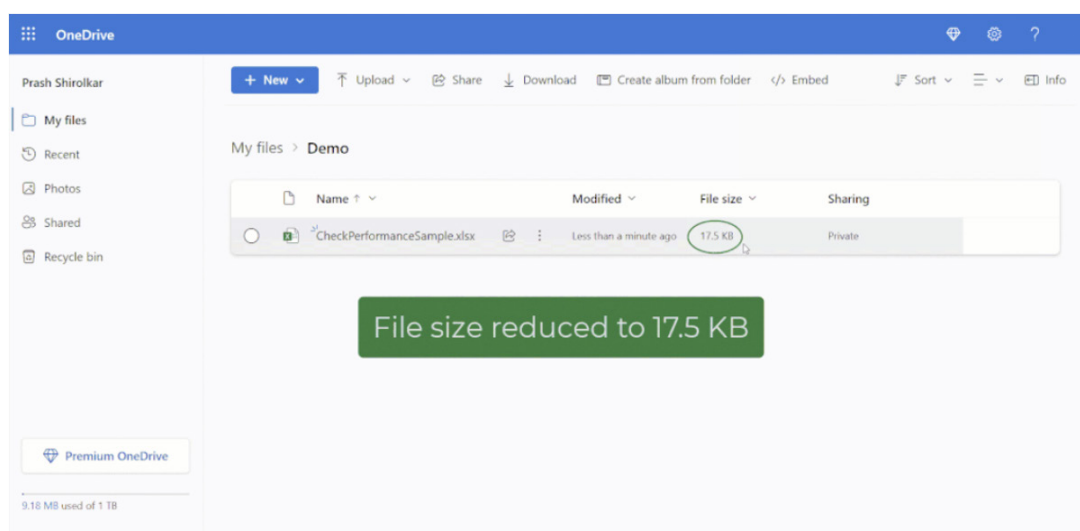
Clicking on the 'Inventory' box in the 'Workbook Performance' pane yields information on three empty cell ranges that may be optimised and prompts with an 'Optimize sheet' button (bottom right):



Similarly, the 'Supplier' sheet details one range to consider:



Following the prompts and optimising both sheets generates considerable savings in file size:



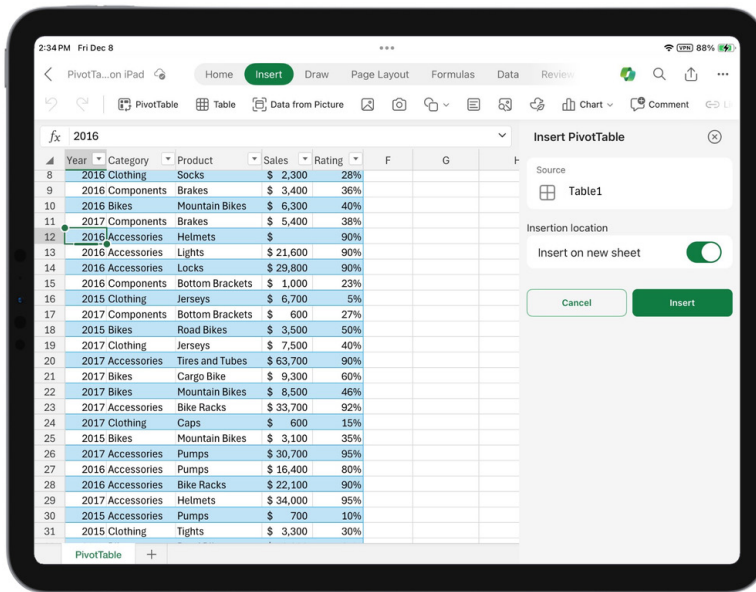
Here, 'Check Performance' has reduced a 3.14 MB file down to 17.5 KB, by detecting and removing more than a million unwanted formatted cells. Bearing this in mind, you might think, why doesn't Excel remove these cells in the background without alerting me? This is because, even though the cell has no data, removing its formatting may result in visible changes. For example, removing yellow fill from a cell may reset its fill to 'No Color', which might not be what you want. Microsoft does not want any Excel users to be surprised by visual changes by doing this in the background without alerting hence the manual interaction.

PivotTables on iPad

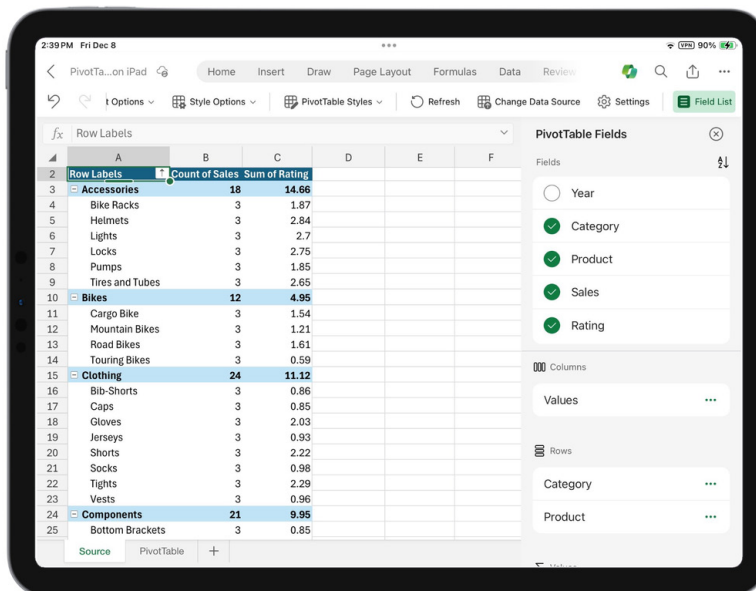
We mentioned this in last month's newsletter, but in case you missed it, Microsoft has just announced support for PivotTable creation and editing on an iPad. This useful data analytical tool is now available for the iPad's smaller screen and touch interface. This gives you have the flexibility to move seamlessly between desktop, web and iPad whilst maintaining a consistent experience across the board.

You should note that this new feature will be enabled gradually to more and more users over time as Microsoft rolls out the update and ensures it is working correctly. Therefore, if you do not see the 'Check Performance' button in the Review menu tab, then the feature may not be enabled for you yet.

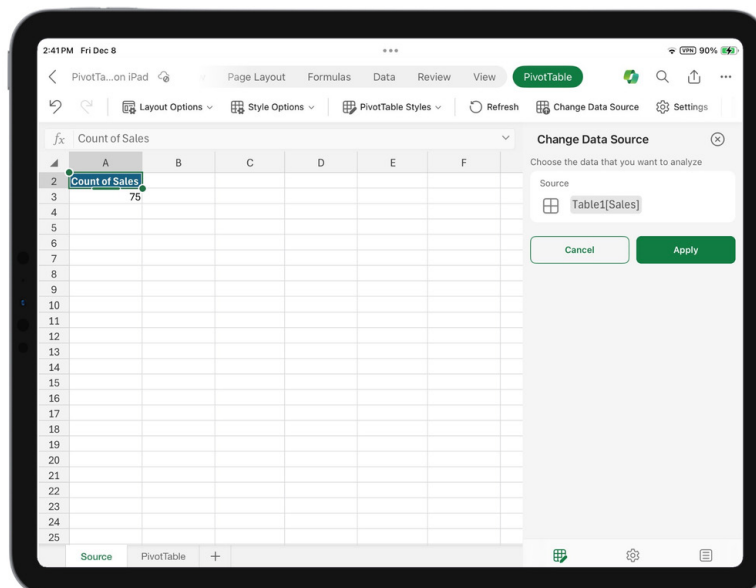
To get started, navigate to the Insert tab in Excel on your iPad, select PivotTable, and choose a 'Source' and 'Insertion' location. You can insert your PivotTable with a single tap:



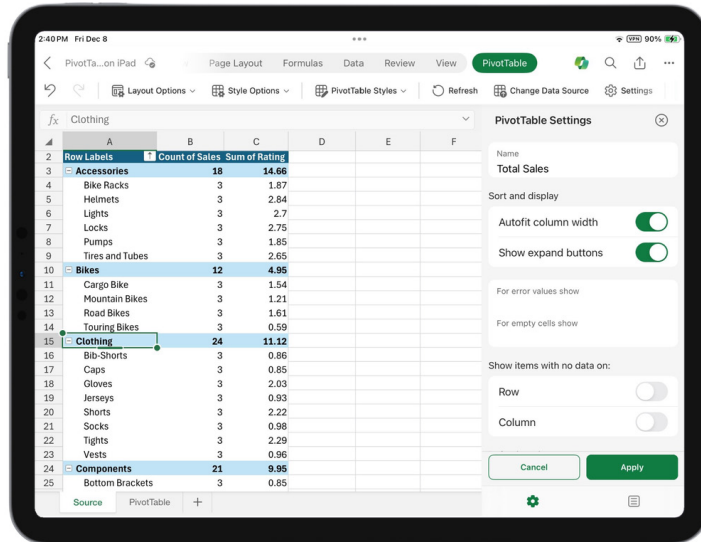
You may tailor a PivotTable to your precise needs using the field list. The areas section at the bottom allows you to rearrange fields by dragging them across the different sections to achieve the required results.



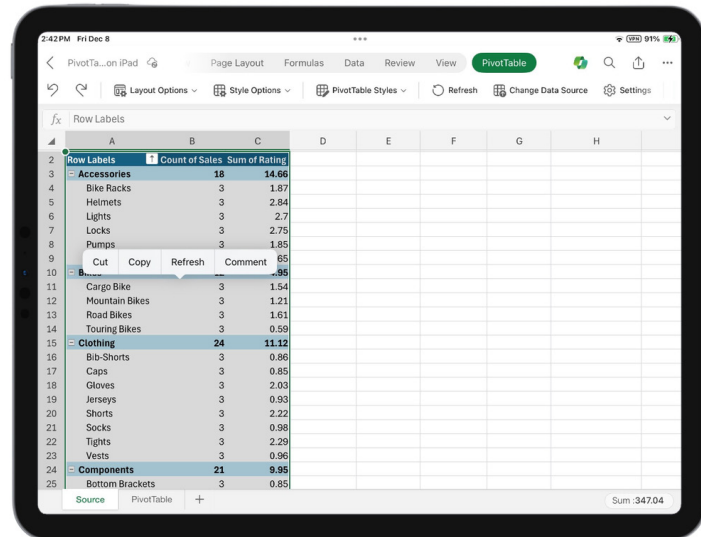
You may adjust your PivotTable's source data by navigating to the PivotTable tab and interacting with the 'Change Data Source' side pane.



You may fine-tune your PivotTable by accessing the Settings side pane. There, you may make the desired modifications and save your changes with a simple tap to make your PivotTable work precisely how you want it to.



You can move your PivotTable within and across worksheets through cut and paste in the context menu too.



To use this feature, you need to be running Excel on iPad version 2.80.1203.0 and above.

The interactive links at aka.ms/ExcelFeaturesFlyer no longer appears to be update, so this month sees the last time we will incorporate this into our newsletter (parting is such sweet sorrow!).

Excel Features Availability

Page 1 of 4

Feature	Insider		Production				Web
	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MCC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	
Block XL Add-Ins	Version 2302 (Build 16130.20320) or later		Already Supported	Already Supported	Already Supported	Already Supported	February 2023
PivotTables: Manual Sort of Rows & Columns							
Automatic Recalculation Optimization	Version 2208 (Build 15629.10000) or later	Version 16.64 (Build 22081401) or later					
Import Data from SQL Server Databases							
Import Data from Additional Sources							Version 16.69 (Build 23010700) or later
Power Query Editor							Version 16.69 (Build 23010700) or later
IMAGE function							Version 16.67 (Build 22102600) or later
Check Formulas with Value Preview Tooltips	Version 2302 (Build 16130.20000) or later	Version 16.70 (Build 230114) or later	Version 2211 (Build 15831.20150) or later	Version 2211 (Build 15831.20150) or later			December 2022
Office Scripts	Version 2212 (Build 15922.20000) or later	Version 16.68 (Build 22120101) or later					
Automatic Tasks with Power Automate task			Version 2301 (Build 15703.10000) or later				Version 16.66 (Build 22092500) or later
PivotTable Show Details to XLO							January 2023
Excel Live in Teams							December 2022
Formula Suggestions							December 2022*
Formulas by Example							December 2022*
Suggested Links							December 2022
Add search bar in queries pane							December 2022

*Sharing to roll out

Features Flyer: aka.ms/ExcelFeaturesFlyer

© 2022 Microsoft Corporation. All rights reserved. Microsoft, the Microsoft Dynamics logo, and the Microsoft Dynamics logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Excel Features Availability

Page 2 of 4

Feature	Insider		Production				Web
	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	
Add keyboard shortcut to open PQ editor			Version 2211 (Build 15730.11883) or later				
Create nested PQ data types	Version 2211 (Build 15928.10000) or later						
Add Get Data from Dynamic Arrays Data from picture	Version 2105 (Build 14014.20002) or later		Version 2210 (Build 15723) or later	Version 2210 (Build 15726.20262) or later			Version 16.38 or later December 2022
Chart Data Folds Show Changes New Paste Options			Version 2209 (Build 15763.10000) or later				Version 16.66 (Build 22092500) or later Already Supported
Quickly find the Command you need New DAX Functions	Version 2210 (Build 15726.20000) or later		Version 2206 (Build 15331.20010) or later				October 2022
Navigation Pane Smooth Scrolling	Version 2208 (Build 15504.10000) or later		Version 2209 (Build 15828.10000) or later				Version 2208 (Build 15601.20230) or later Already Supported
Check Performance Share Section of Excel Workbook			Version 2205 (Build 15225.20092) or later	Version 2208 (Build 15601.20230) or later	Version 2208 (Build 15601.20456) or later		Already Supported 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 Manage Your Storage Accounts from Mac		Version 16.64 (Build 22081200) or later	Version 2209 (Build 15427.20194) or later				

Features Flyer: 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 Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	
New Excel functions Power Query Group operations			Version 2208 (Build 15427.20194) or later	Version 2208 (Build 15601.20230) or later			Version 16.64 (Build 22081401) or later August 2022
Improvements to the connected Power BI experience	Version 2208 (Build 15601.20028) or later						August 2022
Add and edit rich text formatting 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 Edit files with legacy Shared Workbook feature			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Delete chart elements Multiline formula bar							August 2022
Search within PivotTable Field List Set automatic data connections	Version 2207 (Build 15427.20000) or later		Already Supported	Already Supported	Already Supported	Already Supported	July 2022
Natural Language Query Improvements Revise Conditional Formatting dialog box		Version 16.64 (Build 22070600) or later	Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20196) or later		Version 16.63 (Build 22070801) or later	
Sheet protection			Already Supported	Already Supported	Already Supported	Already Supported	June 2022

Features Flyer: 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 Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	
Semi-select for links creation 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 Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.57 (22011100) or later Version 16.62 (22061100) or later	
Power Query refresh for selected data sources Changing source file for workbook links			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Improved Recommended PivotTable experience Faster recalc on resource constrained devices	Version 2204 (Build 15128.10000) or later	Version 16.61 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster Autofilter Dataflow connector				Version 2204 (Build 15128.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataverse connector Improved Find dialog and Find All LAMBDA: Helper Functions			Version 2204 (Build 15128.20178) or later	Version 2203 (Build 15028.20248) or later		Version 16.60 (220410) or later Version 16.56 (Build 211211) or later	Already Supported

Features Flyer: aka.ms/ExcelFeaturesFlyer

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

Back next month, we're sure.

The A to Z of Excel Functions: MOD

If you build forecast models where certain transactions occur periodically but not every period, then this function might be for you. For example, you might model monthly and pay tax quarterly; you may model quarterly yet pay dividends half-yearly, etc. This is where the **MOD** function comes in – even if it causes division amongst the financial modelling community (*groan – Ed.*).

The **MOD** function, **MOD(number, divisor)**, returns the remainder after the **number** (first argument) is divided by the **divisor** (second argument). The result will have the same sign as the **divisor**.

For example, $9 / 4 = 2.25$, or 2 remainder 1. **MOD(9,4)** is an alternative way of expressing this, and hence equals one (1) also. Note that the 1 may be obtained from the first calculation by $(2.25 - 2) \times 4 = 1$, i.e. in general:

$$\text{MOD}(n,d) = n - d \cdot \text{INT}(n/d),$$

where **INT()** is the integer function in Excel.

This function has various uses and I provide three common examples below:

1. **Obtaining “residuals”:** In some instances in modelling, you need the integer part of a number, e.g. how many payments fall between two dates may calculate as 9.94 – but that’s nonsense. In this instance, you would have only made nine payments, i.e. **INT(9.94)**.

Similarly, you might want to accrue the fee for payments not yet made. Using **MOD(9.94,1)** you would obtain the result 0.94, i.e. the number after the decimal place. Note that $9.94 - \text{INT}(9.94)$ gives the same result here; the **MOD** approach is simply shorter.

2. **Calculations at regular time intervals:** Consider the aforementioned tax payments as an example. Many companies make tax payments quarterly (i.e. once every three months). If we assume these payments are made in March, June, September and December then we can formulate the payment as **IF(MOD(Month_Number)=0,Make_Payment,0)**, etc.

	1	2	3	4	5	6	7	8	9	10	11	12
	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20
Assumptions												
Payments quarterly.												
Start Date	01-Jan-20											
Payment Month Cycle	September											
Month Number	9											
Tax Rate	30%											
Net Profit Before Tax	360	520	480	377	590	610	485	360	500	520	440	655
Calculations												
Net Profit Before Tax	360	520	480	377	590	610	485	360	500	520	440	655
Tax Rate	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%	30%
Tax	108	156	144	113	177	183	146	108	150	156	132	197
Tax Payment Flag	-	-	1	-	-	-	1	-	-	1	-	1
Tax Payment	-	-	408	-	-	473	-	-	404	-	-	485

3. **Summing every nth row:** It is not uncommon for users to want to sum every nth cell (e.g. second, third, fourth,...) in a spreadsheet. Excel has no standard function which will do this, but **MOD** can come to the rescue. For example, the array formula (obtained by pressing **CTRL + SHIFT + ENTER** to get the braces { and }), which will work in all versions of Excel,

{=SUM(IF(MOD(\$E\$19:\$E\$48,\$G\$13)=0,\$F\$19:\$F\$48,0))}

was used in cell **H53** in the following example:

	C	D	E	F	G	H
10						
11						
12						
13				Row Interval:	3	
14						
15						
16				Data		
17						
18				#	Amount	
19				1	18	
20				2	1	
21				3	14	
22				4	7	
23				5	9	
24				6	5	
25				7	5	
26				8	11	
27				9	11	
28				10	6	
29				11	10	
30				12	12	
31				13	6	
32				14	14	
33				15	1	
34				16	2	
35				17	9	
36				18	8	
37				19	10	
38				20	9	
39				21	1	
40				22	17	
41				23	7	
42				24	14	
43				25	18	
44				26	15	
45				27	12	
46				28	9	
47				29	15	
48				30	17	
49						
50						
51				Calculation		
52						
53				Sum of every third row:		95
54						

Arrays using large ranges can cause calculations to slow down considerably. This is why I used a counter rather than the volatile **ROW()** function (you may recall volatile functions calculate each time you press **ENTER** or **F9**).

If accuracy is vital, be careful with **MOD** as it may give very slightly erroneous results:

	A	B	C	D	E	F	G	H	I	J
6										
7										
8										
9										
10										
11						Number	6,222.00			
12						Divisor	622.20			
13										
14										
15						MOD				
16										
17						MOD	-4.547E-13			
18										
19										
20						Longhand				
21										
22						MOD(Number,Divisor) = Number - Divisor*INT(Number/Divisor)				
23										
24						Answer	0			
25										
26										
27						Error				
28										
29						MOD = Longhand?	No			
30										

The result for **MOD** in cell **G17** might seem inconsequential, but imagine you were making calculations based on **MOD(number, divisor)=0**. In this case, **MOD** would not equal zero and the calculation would not work.

This issue tends to occur more commonly when working with non-integers.

The problem here isn't really **MOD**. Calculations are performed in what is known as a binary [1,0] format and many decimal numbers have no exact binary representation (just as 1/3 has no exact decimal representation). As an illustration, 10 times the binary approximation to 622.2 is

6222.0000000000004547....

i.e. you may need to use the **ROUND(number, num_digits)** as part of your formula too in order to round your **number** to **num_digits** number of digits (after the decimal point).

Another weird anomaly is if the number is 134,217,728 (2^{27}) times greater or more than the **divisor** this gives rise to a **#NUM!** error *viz.*

	A	B	C	D	E	F	G	H	I
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									

MOD Calculation

Assumptions

Number: 268,435,456.00
 Divisor: 2.00

MOD

MOD: #NUM!

Longhand

$MOD(\text{Number}, \text{Divisor}) = \text{Number} - \text{Divisor} * INT(\text{Number} / \text{Divisor})$

Answer: 0

Error

MOD = Longhand? No

Some texts suggest that you could use the formula

$$=MOD(MOD(\text{number}, 134217728 * \text{divisor}), \text{divisor})$$

This will solve for larger numbers much larger than the limit for **MOD**, but theoretically will hit the same problem when the number being evaluated reaches $134,217,728 * 134,217,728 * \text{Divisor}$. For most uses, this limit is large enough that it will never be reached, but I suggest sticking with Microsoft's recommended solution which is calculating the "long handed" result as illustrated above (cell **G24**).

Also, when using the **MOD** function with one negative number and the expected result is the numerator, **MOD(9,-10)** actually returns -1, whereas you could argue the correct result should be 9:

	A	B	C	D	E	F	G	H	I	J
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										

MOD Calculation

Assumptions

Number: 9.00
 Divisor: (10.00)

MOD

MOD: -1

Longhand

$MOD(\text{Number}, \text{Divisor}) = \text{Number} - \text{Divisor} * INT(\text{Number} / \text{Divisor})$

Answer: -1

Error

MOD = Longhand? Yes

Note that the longhand approach also gives a result of -1.

Microsoft explains that this approach has been taken deliberately in order to be consistent with the competitor dBase software's **MOD** function. If you always need **MOD** to deliver a value of **x** where $0 \leq x < \text{divisor}$, then use the adjusted formula:

$$=\text{IF}(\text{MOD}(\text{number}, \text{divisor}) < 0, \text{ABS}(\text{divisor}) + \text{MOD}(\text{number}, \text{divisor})).$$

Now that I have explained **MOD**, I can show you a really good use of this function for a situation which causes many working in finance plenty of consternation. The solution works in conjunction with **OFFSET**. Let me just remind you of our old friend the **OFFSET** as follows:

$$\text{OFFSET}(\text{reference}, \text{rows}, \text{columns}, [\text{height}], [\text{width}]).$$

The arguments in square brackets (**height** and **width**) can be omitted from the formula in this instance. In its most basic form, **OFFSET(reference, x, y)** will select a **reference** **x** rows down (**-x** would be **x** rows up) and **y** columns to the right (**-y** would be **y** columns to the left).

Imagine you have set up a scenario table in Excel to determine which inputs should be used in your model (i.e. "what-if? Analysis"):

Assumption Description	Amts Used	1	2	3	4	5
Base Year Sales	\$8,000	\$10,000	\$20,000	\$15,000	\$8,000	\$10,000
Sales Growth 2021	2.0%	2.0%	1.0%	2.0%	2.0%	-
Sales Growth 2022	3.0%	3.0%	1.5%	3.0%	3.0%	-
Sales Growth 2023	2.5%	2.5%	4.0%	2.5%	2.5%	-
Sales Growth 2024	2.0%	2.0%	4.0%	2.0%	2.0%	-
Sales Growth 2025	3.0%	3.0%	2.5%	3.0%	3.0%	-
Sales Growth 2026	2.0%	2.0%	3.0%	2.0%	2.0%	-

This method allows for various scenarios to be modelled easily with a different set of input data inserted into each column (from column L onwards in this illustration). A selector (cell J11 in the figure above) is used to select the active scenario, which may be highlighted using conditional formatting (see later).

The data used to drive the model is then highlighted in column J (here, emphasised in yellow) using the follow formula for cell J14 for example:

$$=\text{OFFSET}(K14,,,J11)$$

In other words, this formula looks up data **x** columns to the right of column K, where **x** is specified as the value input in cell J11 (here, this value is 4 so column O's data is selected).

Clearly, using a columnar approach here makes it very straightforward to set the various scenarios out. However, most financial models are displayed with dates going from left to right across columns rather than down the page using rows. This requires us to transpose the data, and again we may use **OFFSET** to 'flip' the data:

Summary	Period Year	1	2	3	4	5	6	7
	Year	2020	2021	2022	2023	2024	2025	2026
Inputs		\$8,000	2.0%	3.0%	2.5%	2.0%	3.0%	2.0%
Sales		\$8,000	\$8,160	\$8,405	\$8,615	\$8,787	\$9,051	\$9,232

Here, the period numbers specified in row 31 make it easy for us to transpose the data. For example, the formula in cell L34 would be:

$$=\text{OFFSET}(\$J\$13,L\$31,)$$

i.e. insert the data **x** rows down from cell J13 in the first graphic, where **x** is again specified as the value input in cell J11.

Take care, however, if using an amount followed by growth rates approach for forecast / budget data. The amounts using these examples should be as follows:

Formula bar: `=IF(L$31=1,L$34,K36*(1+L$34))`

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
27	Calculations																		
28	Summary																		
29																			
30																			
31																			
32																			
33																			
34																			
35																			
36																			
37																			

The correct formula here is:

$$=IF(L\$31=1,L\$34,K36*(1+L\$34))$$

for cell L36 (say), i.e. if it is the first period take the amount, otherwise take the amount calculated in the preceding period and multiply it by (1 + growth rate specified in the current period not the next period).

When actual data is input into a model, frequently it replaces the original information, and therefore management loses the ability to see how accurate forecasts were originally, and how budgeting may be improved. One way round this would be to simply have "Actuals" as one of the scenarios so that all forecasts are retained. This is often all that is required, and if so, simply do that – Keep It Simple Stupid (KISS).

However, often we may wish to undertake variance analysis by comparing actual data with the original budgeted information. In this case, I would suggest the following approach.

Formula bar: `=IF(L$18<>"",L$18,IF(L$3=1,L$11,K23*(1+L$11)))`

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
3																		
4																		
5																		
6																		
7																		
8																		
9																		
10																		
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		
21																		
22																		
23																		
24																		

Rows 9 to 13 of this illustration simply reiterate the calculations already detailed above regarding the original forecasting. Note row 18 however: this is where actual data is added instead. In my example, I simply use hard coded inputs for my data, but it only requires a simple variation to this methodology to revise growth rates, etc.

Using my logic, we simply use actual data where it is available; otherwise we fall back on the original data and calculations. This is achieved by the formula in row 23 in my example, which is (for cell L23):

$$=IF(L$18<>"",L$18,IF(L$3=1,L$11,K23*(1+L$11))),$$

i.e. if there is data in the corresponding cell in row 18 use it; if not, if it is the first period take the original input value, otherwise simply inflate the prior period amount by (1 + growth rate for that period). It may include a nested IF statement, but it is still a relatively simple and straightforward calculation.

Performing the calculations is only half of the battle. Modellers often have difficulty comparing the original outputs with the reforecast counterparts in an effective and efficient manner. If sufficient, the following would be relatively straight forward:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1	Simple Outputs Example																		
2																			
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			

This is very easy to put together, but alas, more often than not, the following presentation is required by senior management instead:

Counter	1	2	3	4	5	6	7	8	9	10
Year	2020	2020	2020	2021	2021	2021	2022	2022	2022	2023
Description	Budget	Act / Ref'cast	Variance	Budget	Act / Ref'cast	Variance	Budget	Act / Ref'cast	Variance	Budget
Bad Outputs Example										
Sales	\$8,000	\$8,000	-	\$8,160	\$9,000	\$840	\$8,405	\$8,500	\$95	\$8,615

Seem familiar? I have been a model reviewer for many a year and seen this type of output on an extremely regular basis. Many senior management teams like it this way and it is not my role to challenge the status quo – it doesn't stop me from trying though!

practice. Modellers tend to a large number of unique formulae across a row, which in turn slows down model construction and increases the potential for mistakes, such as referencing errors.

The problem with this layout is that it lends itself to promoting poor

If you have to use this layout, creating the simple summary elsewhere (maybe on an input page),

Year No.	1	2	3	4	5	6	7
Budget	\$8,000	\$8,160	\$8,405	\$8,615	\$8,787	\$9,051	\$9,232
Act. / Ref'cast	\$8,000	\$9,000	\$8,500	\$8,713	\$8,887	\$9,153	\$9,336
Variance	-	\$840	\$95	\$98	\$100	\$103	\$105

Look carefully at this graphic. The shading in cell K28 above may appear innocuous, but it is the most important cell on the worksheet and has been named BC_Sales_Summary accordingly (BC means "Base Cell" for an OFFSET function). Consider:

- For every column moved to the right, the cursor is in a different year. One column across is Year 1, two columns across is Year 2 and so on
- For every row moved down, the cursor reports a different figure. One row down is the Budget data, two rows down is the Actual / Reforecast data and three rows down is the Variance. The depth of this table (three rows) has been defined as List_Depth so that it may be used in formulae.

Now, if we return to the outputs worksheet and modify it slightly by inserting two additional lines and using the OFFSET function once more may make your potential troubles a thing of the past, viz.

Counter	1	2	3	4	5	6	7	8	9	10
Year	2020	2020	2020	2021	2021	2021	2022	2022	2022	2023
Description	Budget	Act. / Ref'cast	Variance	Budget	Act. / Ref'cast	Variance	Budget	Act. / Ref'cast	Variance	Budget
Selector	1	2	3	1	2	3	1	2	3	1
Year No.	1	1	1	2	2	2	3	3	3	4
Better Outputs Example										
Sales	\$8,000	\$8,000	-	\$8,160	\$9,000	\$840	\$8,405	\$8,500	\$95	\$8,615

You will see that each row of the revised output example contains only one unique formula copied across, making it easy to edit, extend and review. This is achieved by adding two rows:

- **Selector (row 7):** identifies whether the column should be reporting the budget information, the actual data or the variance. The equation used makes use of the MOD function:

$$=MOD(L$4-1, List_Depth)+1$$

As stated above, List_Depth is the number of selections (Budget, Act / Ref'cast, Variance) permissible – in this case three.

=MOD(L\$4, List_Depth) takes the counter and converts each one to 1, 2 and zero (i.e. the remainder upon dividing the counter by three). By subtracting one inside the MOD function and adding it once more outside, this simply forces a zero to a three instead, so that the Selector reports the values 1, 2 and 3 alternately.

- **Year No. (row 8):** simply notes which year the column is reporting using the formula:

$$=IF(L$7=1, K8+1, K8),$$

i.e. the year increases the period that the counter in row 7 equals 1 (in this example).

Therefore, in my example, row 13 requires a very simple formula to generate the required outputs:

=OFFSET(BC_Sales_Summary,L\$7,L\$8).

For example, cell **Q13** equals **OFFSET(BC_Sales_Summary,3,2)**, which would refer to the Year 2 Variance figure of \$840.

Easy!

The A to Z of Excel Functions: MODE



The mode is the value that appears most often in a set of data values. If X is a discrete random variable, the mode is the value x (*i.e.* $X = x$) at which the probability mass function takes its maximum value; it is the value that is most likely to be sampled.

The mode is not necessarily unique to a given discrete distribution, since the probability mass function may take the same maximum value at several points $x_1, x_2, \text{etc.}$ The most extreme case occurs in uniform distributions, where all values occur equally frequently.

- **Average:** the arithmetic mean, calculated by adding a group of numbers and then dividing by the count of those numbers. For example, the average of 2, 3, 3, 5, 7, and 10 is 30 divided by 6, which is 5
- **Median:** the middle number of a group of numbers; that is, half the numbers have values that are greater than the median, and half the numbers have values that are less than the median. For example, the median of 2, 3, 3, 5, 7, and 10 is 4. If the number of values in the selection is an even number, the median is defined as the midpoint between the two central numbers
- **Mode:** the most frequently occurring number in a group of numbers. For example, the mode of 2, 3, 3, 5, 7, and 10 is 3.

For a symmetrical distribution of a group of numbers, these three measures of central tendency are all the same. For a skewed distribution of a group of numbers, they can be different.

The **MODE** function employs the following syntax to operate:

MODE(number1, [number2], ...)

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

- **number1:** this is required. The first number, cell reference, or range for which you want to calculate the mode
- **number2, ...:** this is optional. Additional numbers, cell references or ranges for which you want the mode, up to a maximum of 255 in total.

It should be further noted that:

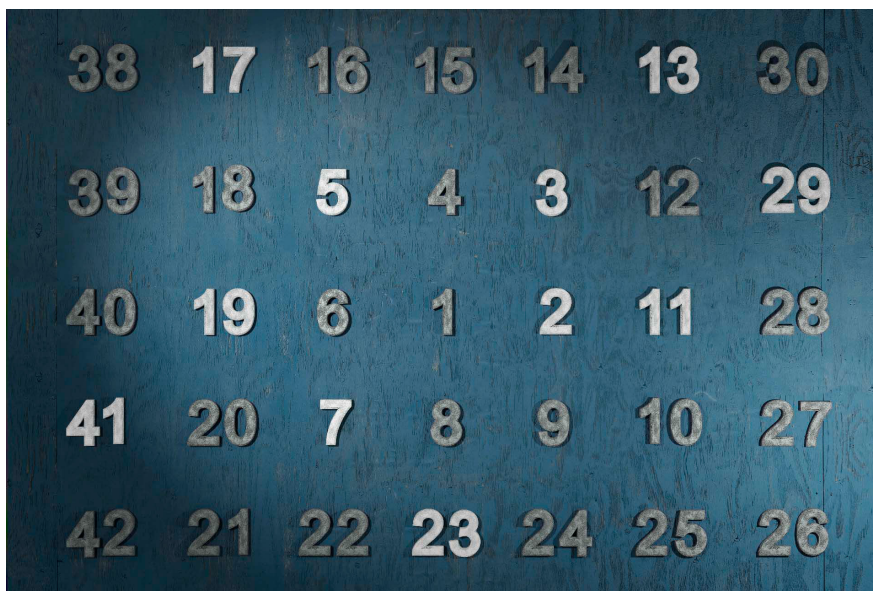
- arguments may either be numbers or names, ranges or cell references that contain numbers
- logical values and text representations of numbers that you type directly into the list of arguments are counted
- if a range or cell reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value zero are included
- arguments that are error values or text that cannot be translated into numbers cause errors
- if the data set contains no duplicate data points, **MODE** returns the #N/A error
- if there is more than one mode, **MODE** reports the first mode value that occurs.

This function has been replaced with two new functions (**MODE.MULT** and **MODE.SNGL**, see below) that may provide improved accuracy and whose names better reflect their usage. Although this function is still available for backward compatibility, you should consider using the new functions from now on, because this function may not be available in future versions of Excel.

Please see our example below:

	A	B	C
1	Values	Mode	Formula
2	2	#N/A	=MODE(A\$2:A2)
3	2	2	=MODE(A2,A3)
4	3	2	=MODE(A\$2:A4)
5	3	2	=MODE(A\$2:A5)
6	4	2	=MODE(A\$2:A6)
7	4	2	=MODE(A\$2:A7)
8	4	4	=MODE(A\$2:A8)
9			

The A to Z of Excel Functions: MODE.MULT



The **MODE.MULT** function measures what is known as “central tendency”, which is the location of the centre of a group of numbers in a statistical distribution.

The **MODE.MULT** function returns a **vertical** array of the most frequently occurring, or repetitive values in an array or range of data. For horizontal arrays, use

TRANSPOSE(MODE.MULT(number1, number2, ...)).

This will return more than one result if there are multiple modes. Because this function returns an array of values, it must be entered as an array formula in Legacy Excel, but will return a dynamic array in Excel 365 and Excel for the web and Excel 2021 onwards.

It employs the following syntax to operate:

MODE.MULT(number1, [number2], ...)

The **MODE.MULT** function has the following arguments:

- **number1**: this is required. The first number, cell reference, or range for which you want to calculate the mode
- **number2, ...**: this is optional. Additional numbers, cell references or ranges for which you want the mode, up to a maximum of 255 in total. You can also use a single array or a reference to an array instead of arguments separated by commas.

It should be further noted that:

- arguments may either be numbers or names, ranges or cell references that contain numbers
- logical values and text representations of numbers that you type directly into the list of arguments are counted
- if a range or cell reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value zero are included
- arguments that are error values or text that cannot be translated into numbers cause errors
- if the data set contains no duplicate data points, **MODE.MULT** returns the #N/A error.

This function is one of two to replace the **MODE** function, in order to provide improved accuracy and whose names better reflects its usage.

Please see our penultimate example below:

	A	B	C
1	Values	Legacy	Dynamic Array
2	1	1	1
3	1	2	2
4	1	3	3
5	2	4	4
6	2	#N/A	
7	2	#N/A	
8	3	#N/A	
9	3	#N/A	
10	3	#N/A	
11	4	#N/A	
12	4	#N/A	
13	4	#N/A	
14	5	#N/A	
15	5	#N/A	
16	6	#N/A	
17			
18		{=MODE.MULT(A2:A16)}	=MODE.MULT(A2:A16)
19		written as array function	
20		in cells B2:B16	

The A to Z of Excel Functions: MODE.MULT



The **MODE.SNGL** function measures what is known as “central tendency”, which is the location of the centre of a group of numbers in a statistical distribution. It returns the most frequently occurring, or repetitive, value in an array or range of data. It employs the following syntax to operate:

MODE.SNGL(number1, [number2], ...)

The **MODE.SNGL** function has the following arguments:

- **number1**: this is required. The first number, cell reference, or range for which you want to calculate the mode
- **number2, ...**: this is optional. Additional numbers, cell references or ranges for which you want the mode, up to a maximum of 255 in total. You can also use a single array or a reference to an array instead of arguments separated by commas.

It should be further noted that:

- arguments may either be numbers or names, ranges or cell references that contain numbers
- logical values and text representations of numbers that you type directly into the list of arguments are counted
- if a range or cell reference argument contains text, logical values, or empty cells, those values are ignored; however, cells with the value zero are included
- arguments that are error values or text that cannot be translated into numbers cause errors
- if the data set contains no duplicate data points, **MODE.SNGL** returns the #N/A error
- if there is more than one mode, **MODE.SNGL** reports the first mode value that occurs.

This function is one of two to replace the **MODE** function, in order to provide improved accuracy and whose names better reflects its usage.

Please see our final example for this month below:

	A	B	C
1	Values	Mode	Formula
2	2	#N/A	=MODE.SNGL(A\$2:A2)
3	2	2	=MODE.SNGL(A2,A3)
4	3	2	=MODE.SNGL(A\$2:A4)
5	3	2	=MODE.SNGL(A\$2:A5)
6	4	2	=MODE.SNGL(A\$2:A6)
7	4	2	=MODE.SNGL(A\$2:A7)
8	4	4	=MODE.SNGL(A\$2:A8)

More Excel Functions next month.

Beat the Boredom Suggested Solution

We're back with the solution to our (southern hemisphere) sun-soaked Excel challenge. This challenge was all about managing a summer camp's activities, ensuring none were overbooked. Let's unravel the solution, which ensures that each camper gets their fair share of fun under the sun!

We had two Excel tables – **Activities_Table** and **SignUps_Table**. Our task was to create a dynamic formula to check if the number of sign-ups for each activity per day exceeded its maximum capacity.

	A	B		A	B	C
1	Activity	Max Capacity		Camper Name	Activity	Day
2	Beach Volleyball	20		Alice	Beach Volleyball	Monday
3	Surfing	15		Bob	Beach Volleyball	Tuesday
4	Nature Walk	25		Charlie	Surfing	Monday
5	Art Workshop	10		David	Nature Walk	Tuesday
6	Kayaking	12		Eve	Art Workshop	Wednesday

The requirements were for the solution to:

- be dynamic and update immediately as new sign-ups are added
- use only Excel formulas - no Power Query, Get & Transform, or VBA allowed
- be case-insensitive.

Suggested Solution

SETTING UP DAY COLUMNS IN THE ACTIVITIES TABLE

We need to prepare the **Activities_Table** for our calculations. In addition to the 'Activity' and 'Max Capacity' columns, we added columns for each day of the week (Monday to Friday). These columns will hold the count of sign-ups for each activity per day

	A	B	C	D	E	F	G
1	Activity	Max Capacity	Monday	Tuesday	Wednesday	Thursday	Friday
2	Beach Volleyball	20					
3	Surfing	15					
4	Nature Walk	25					
5	Art Workshop	10					
6	Kayaking	12					

THE FORMULA

We used the **COUNTIFS** function in Excel, which counts the number of times multiple conditions are met. The formula structure in each day column of the **Activities** table is as follows:

=COUNTIFS(SignUps_Table[Activity], Activities_Table[@Activity], SignUps_Table[Day], "Specific Day")

Here, **SignUps_Table[Activity]** and **SignUps_Table[Day]** refer to the respective columns in the **SignUps_Table**, and **Activities_Table[@Activity]** refers to the current activity in the **Activities_Table**. "Specific Day" is replaced with the cell reference for actual day we're checking (e.g. **C\$1**).

COMPARING WITH MAXIMUM CAPACITY

=IF(COUNTIFS(SignUps_Table[Activity], Activities_Table[@Activity], SignUps_Table[Day], C\$1) > Activities_Table[@Max Capacity]), "Overbooked", "Available")

This formula is replicated for each day column, changing the **C\$1** parameter to the respective column reference. The formula returns "Overbooked" for activities that exceed their capacity on a given day and "Available" for those within capacity.

	A	B	C	D	E	F	G
1	Activity	Max Capacity	Monday	Tuesday	Wednesday	Thursday	Friday
2	Beach Volleyball	20	Available	Available	Available	Available	Available
3	Surfing	15	Available	Available	Available	Available	Available
4	Nature Walk	25	Available	Available	Available	Available	Available
5	Art Workshop	10	Available	Overbooked	Available	Available	Overbooked
6	Kayaking	12	Available	Available	Available	Available	Available

DETAILED EXPLANATION

The **COUNTIFS** function works by counting the number of times sign-ups for an activity occur on a specified day. It cross-references the activity name and the day between the **Activities_Table** and the **SignUps_Table**.

By placing this formula in each day column next to every activity, we can dynamically track the number of sign-ups as they happen. The **IF** statement then checks if this count exceeds the maximum capacity. If it does, it flags the activity as "Overbooked."

Word to the Wise

While our formula works great for small to medium datasets, larger datasets with thousands of entries might require more efficient data management solutions like Power Query or even PivotTables for better performance and ease of use.

With this solution, managing the summer camp activities should be a breeze, ensuring every camper has their day in the sun! Stay tuned for our next Excel adventure!

Until next month.

Beat the Boredom Suggested Solution: Addendum

On the website and in our newsletter, we release a series called the “Final Friday Fix” (website) or “Beat the Boredom” (newsletter) where we set an Excel / Power Pivot / Power Query / Power BI problem for users to solve.

You might have thought this month’s puzzle was not the hardest and also you might be surprised / disappointed that we reproduced the January Final Friday Fix. There was a reason for this. This month, we decided to experiment by creating the challenge and its corresponding solution using the new ‘GPTs’ feature in ChatGPT. Here is how we did it and we also reflect on how well it went.

What are GPTs?

Generative Pre-trained Transformers (GPTs) are custom AI chatbots that are built to perform specific tasks. Essentially, they are a type of artificial intelligence (AI) language model.

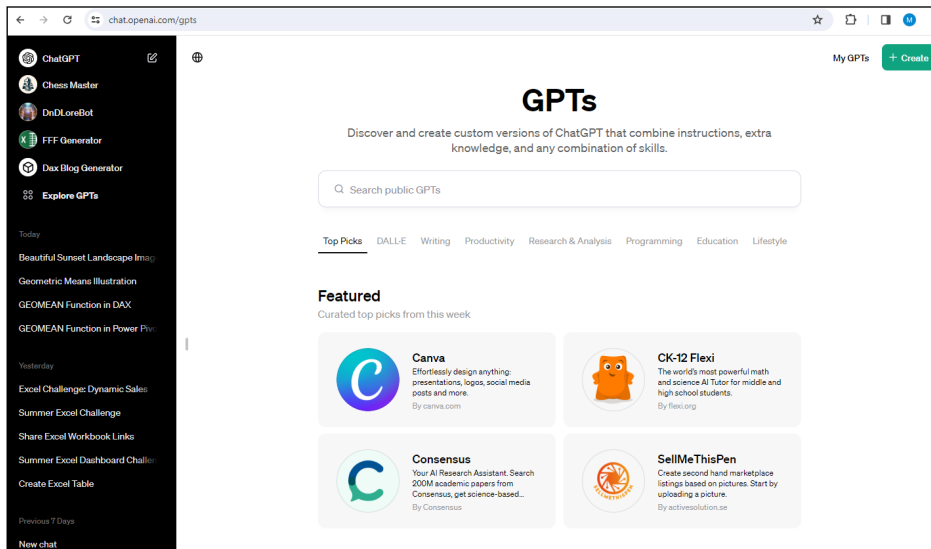
It is important to note for anyone wishing to follow along, GPTs may only be used with the premium ChatGPT service, **GPT+**. One of the major issues with ChatGPT is that it traditionally struggles to deal with specialised tasks as it is built for general knowledge. GPTs allow you to implement specific instructions with a custom knowledge base and

even actions such as calling external Application Programming Interfaces (APIs) (an API is a bridge that allows ChatGPT to interact with other software, but more on this later).

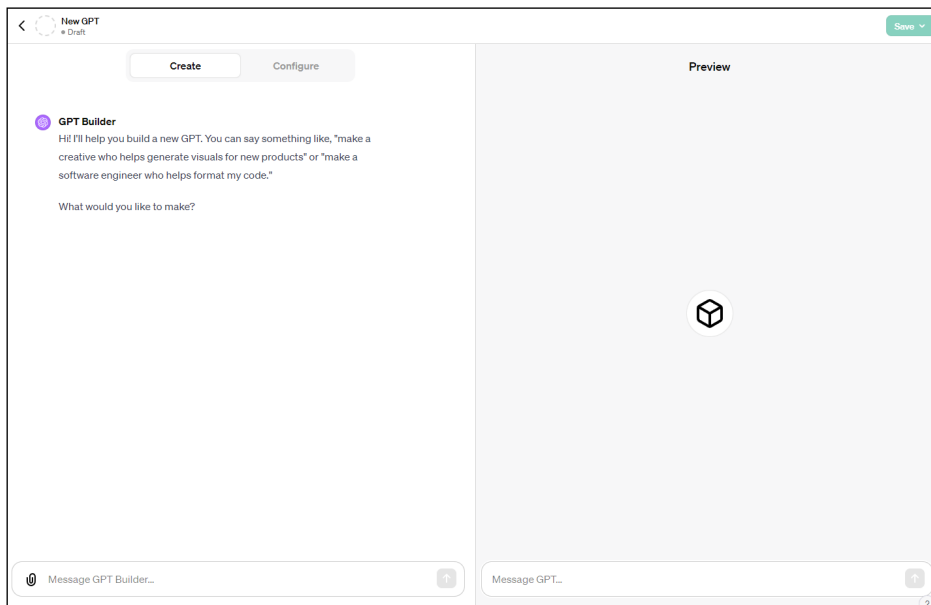
Additionally, it allows the image generation, web browsing and code interpreter features to run alongside and interact with these instructions for additional versatility. Most importantly for us, GPTs are community made and are simple to create even for someone with no prior coding experience.

Creating a GPT

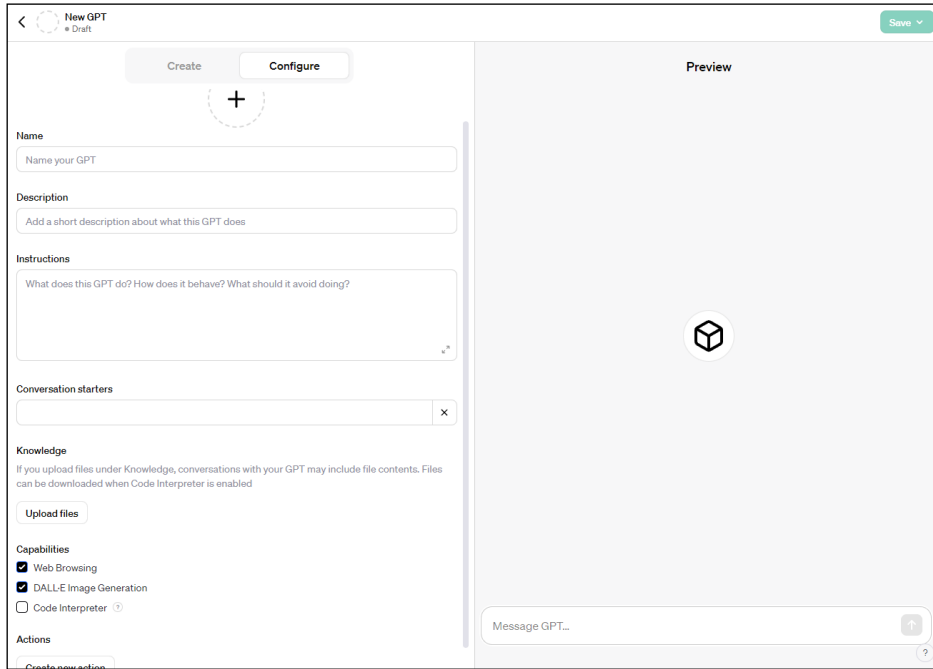
Now, we will run you through how we created the GPT. It starts by opening up ChatGPT and going to the ‘Explore GPTs’ tab.



From here, clicking ‘Create’ in the top right corner will bring us to the GPT Editor.



This is where you create GPTs. The 'Create' tab allows you to interact with ChatGPT as you build, with ChatGPT helping build the GPT with your specifications. The 'Preview' pane on the right allows you to test the GPT as you make changes. As we wanted more control, we selected the 'Configure' tab instead. This tab contains various options allowing us to edit the GPT directly.

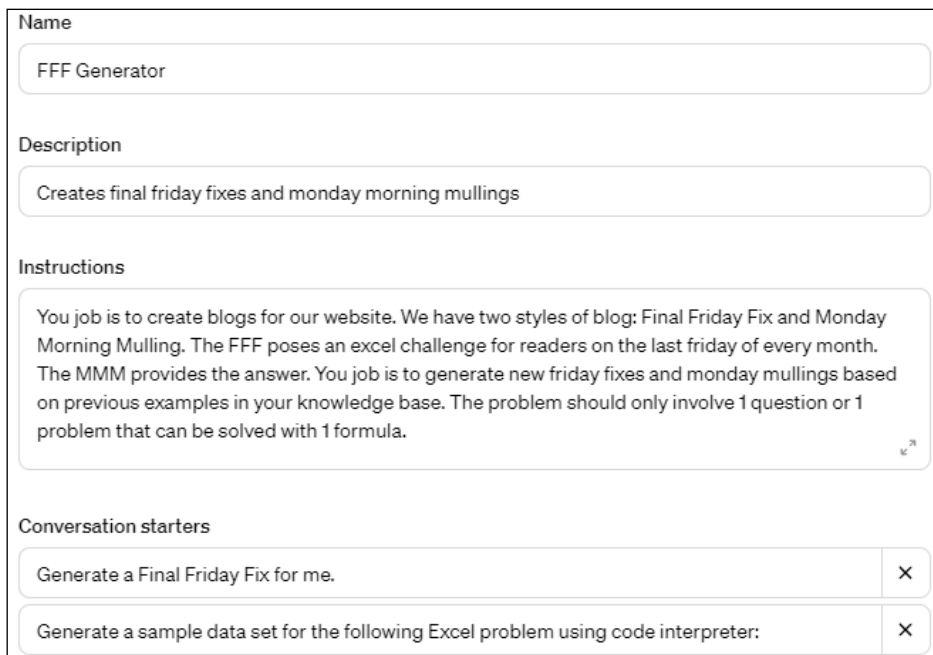


Here, we started by filling in some details. We named it 'FFF Generator' and gave it the description "Creates Final Friday Fixes and Monday Morning Mullings". The instructions are the most important part of the GPT, as they provide the guide for how it will function. We used the following instructions:

You job is to create blogs for our website. We have two styles of blog: Final Friday Fix and Monday Morning Mulling. The FFF poses an excel challenge for readers on the last Friday of every month. The MMM provides the answer. You job is to generate new Friday Fixes and Monday Mullings based on previous examples in your knowledge base. The problem should only involve one question.

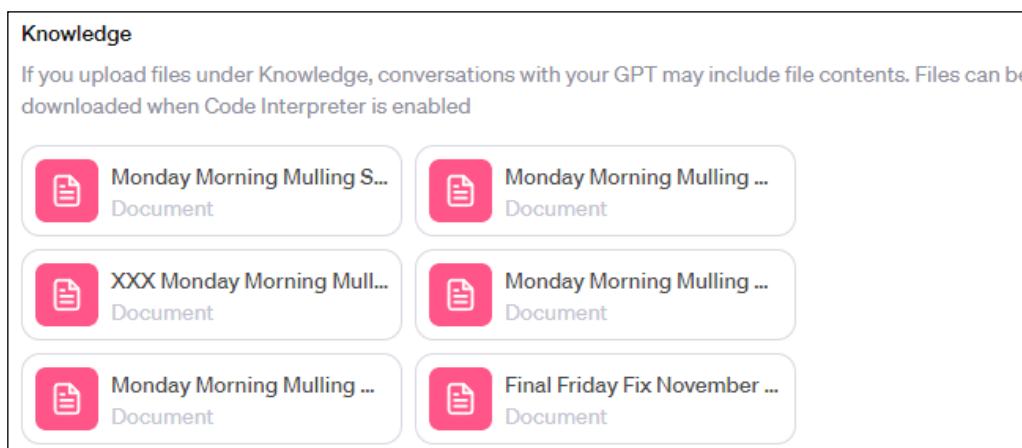
It first describes its basic task, then explains a little context behind what the blogs are. Finally, we added a clarifier that we only want one [1] question per blog to keep it a bit simpler as it was initially producing convoluted, multistage questions. Next is the 'Conversation starters' box which provides prompt suggestions for the user of the GPT. Conversation starters are important when designing a GPT for other people as it gives the user an idea of the capabilities of the GPT. We used conversation starters like:

- "Generate a Final Friday Fix for me"
- "Generate a sample data set for the following Excel problem using code interpreter".



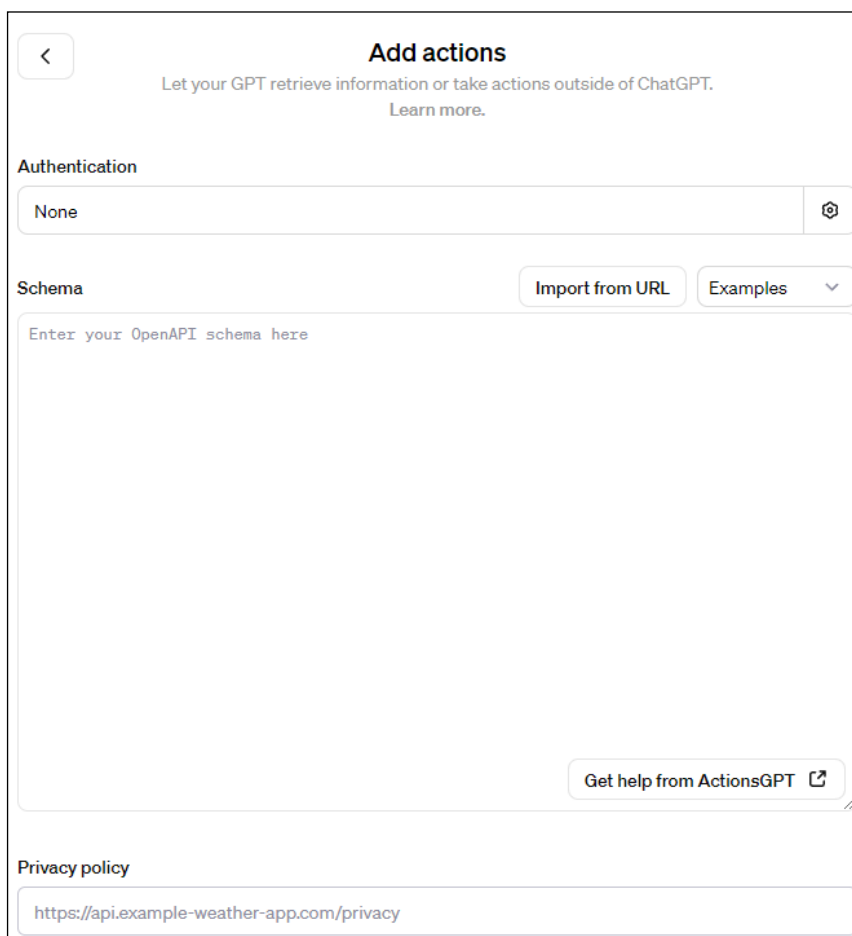
The next step is to bring the brains into the GPT. We have a folder full of FFFs and MMMs from previous months and we uploaded these to the GPT to function as its “knowledge base”. You may upload all sorts of information to the knowledge base to customise the GPT such as instructions, sample text and even Excel spreadsheets. Depending

upon your instructions it will either base its responses on the content or interact with the content to generate results (such as using code interpreter to interact with the data). We may upload the files by clicking the ‘Upload files’ button and selecting the files we wish to use.





Once those were uploaded there were a few more things we needed to adjust. Under ‘Capabilities’ there are three [3] check boxes for ‘Web Browsing’, ‘DALL.E Image Generation’ and ‘Code Interpreter’, which allow the GPT to use those tools. Since GPTs are meant to perform a specific task you may not want the GPT to have access to all of them. In this case, we needed the Code Interpreter to be available so it could generate Excel Spreadsheets, but the other two were not needed.


The ‘Actions’ button takes us to the window below, allowing the user to code custom Actions that allow the GPT to access and interact with external APIs. APIs allow ChatGPT to interact with other software whether they are interacting with external databases or using other software tools. You can even get assistance in building actions through the ‘ActionsGPT’. Actions are incredibly powerful, although we didn’t require them for this GPT.



The final step is to put in a profile picture for the GPT. While we could upload our own image, we can very easily use the ChatGPT image generator by going back over into the 'Create' tab and asking it to create one.

 **You**
Can you create an image for my GPT. It needs to be Excel themed


 **GPT Builder**



Your GPT now has a new Excel-themed profile picture, representing data analysis and calculation with precision and clarity. If there's anything else you'd like to adjust or any other assistance you need, feel free to let me know!

Once the GPT was set up we could test it in the Preview tab. Let's try one of the conversation starters from earlier:

Generate a Final Friday Fix for me


 **FFF Generator**
Final Friday Fix: January 2024 Challenge
27 January 2024

Happy New Year Excel Enthusiasts! As we kick off the new year, we're excited to present you with a fresh Excel / Power Pivot / Power Query / Power BI challenge to stimulate your problem-solving skills. Each month, we set an intriguing problem for you to puzzle over during the weekend. Then, come Monday, we reveal a solution. Remember, alternative answers are welcome, and we do appreciate your creative thinking, even if we stick to our proposed solution.

The Challenge

For this month's challenge, let's dive into the realm of date and time in Excel. Managing dates and times is a common, yet sometimes tricky task in Excel. For this challenge, you're given a dataset in an Excel Table named `Tb1_Events` which contains a column `Event_Dates` filled with dates and times of various events.

Your challenge is to create a dynamic Excel formula that calculates the time difference in hours and minutes between each event and the immediate next event in the list. The twist here is that your formula should only use a single cell (no helper columns or cells allowed) and should dynamically update if new events are added to the `Tb1_Events` Table.




To make this challenge more interesting:

If the GPT is working how we want it, then we just click the 'Save' button in the top right corner and publish it. The GPT can be published just for yourself or for anyone with a link or to the public.

Now that the GPT is set up, we may utilise it by returning to the Explore GPTs page and go into 'My GPTs'. One of the great things about ChatGPT is that we may ask it for an idea as many times as we want and it will produce a new idea every time. Since we wanted this for the end of January, we asked it to

Generate a summer themed Final Friday Fix.

We ran this quite a few times and it produced a variety of responses running into two key issues. The first was that many of the problems that it produced were too simple, such as the one below:

 **FFF Generator**
Final Friday Fix: August 2023 Challenge
25 August 2023

As the sun blazes in its full glory and the waves hit the shore rhythmically, it's time for a summer-themed Final Friday Fix! While you're possibly planning your weekend by the beach or a quiet retreat in the cool mountains, here's an Excel challenge that you can mull over with your iced beverage in hand. Whether you're a fan of Excel, Power Pivot, Power Query, or Power BI, this challenge is for you. Remember, on the following Monday, we publish a solution. If you think there's an alternative answer, feel free to email us. We'll appreciate your enthusiasm!

The Challenge:
Imagine you're managing a beach resort and tracking the daily sales of different summer products: sunscreen, beach hats, and surfboards. You record the sales in an Excel table, named `Tbl1_SummerSales`, structured as follows:


- **Date:** The date of the sale (Column A)
- **Product:** The type of product sold (Column B)
- **Quantity:** The number of units sold (Column C)
- **Price per Unit:** The selling price for each unit (Column D)


You're interested in analyzing the sales trends and performing a quick, insightful analysis directly in Excel. Your challenge is to create one dynamic Excel formula that accomplishes the following tasks:

1. **Calculate the Total Revenue:** Generate a daily total revenue for each product.
2. **Identify the Best Seller:** Determine which product has the highest total revenue for the entire summer season.
3. **Visualize Trends:** Create a simple chart with Excel to visualize the total revenue

The second issue was that when it did produce more complicated questions it ran into another problem. Along with creating the Final Friday Fix, we used the GPT to create the solution and establish a Monday Morning Mulling. When the GPT created more complicated

examples it did struggle to produce good answers or just came up with the inefficient, brute force answers we pride ourselves on avoiding here at SumProduct. Eventually, it produced an answer that we thought, whilst a little simple, was reasonable enough to work.

 **You**
Create a summer themed final friday fix

 **FFF Generator**
Final Friday Fix: August 2023 Challenge
25 August 2023


As the sun blazes in its full glory and the beaches beckon, it's time to cool off with a refreshing Excel challenge. While you sip on your lemonade or enjoy the gentle sea breeze, let's dive into a summer-themed Excel puzzle.


The Challenge
Imagine you're organizing a summer camp and you have a list of activities planned for each day of the week. Each activity has a maximum number of participants, and you've got a dynamic list of campers signing up every day. You need to ensure that the number of campers for each activity does not exceed its capacity.

Here's the twist. You're given two Excel Tables: Activities and SignUps. The Activities table lists various activities along with their maximum capacity. The SignUps table records the name of the camper and their chosen activity for each day of the week.


From here, we asked it to generate an Excel file of sample data to use for the problem, which is why the Excel file wasn't in our usual style. Did you notice? It was able to generate what a spreadsheet of sample data for us, but there were limitations.



Since it is using Python to interact with the spreadsheet it cannot interact with non-sheet features such as Conditional Formatting and Data Validation. It is also poor at building formulae and should be mainly used to populate a sheet with numbers. Once the sheet was created, we were able to adjust it to better suit the question by including more data along with at least one over-signed class for the problem.

 **You**
Use code interpreter to generate a sample excel file for this question I can download
< 2 / 2 >

 **FFF Generator**
The sample Excel file for the summer-themed Final Friday Fix challenge has been created. You can download it using the link below:

[Download the file \[-\]](#)

 **You**
Can you make the data set much larger, with art workshop oversigned on a few days and give me a new file
< 4 / 4 >


 **FFF Generator**
 Error analyzing ▾

The dataset has been expanded, with additional sign-ups for various activities and specific over-signing for the 'Art Workshop' on several days. The new Excel file is ready for you to download:

[Download the extended file \[-\]](#)

< 3 / 3 >

Next, we let ChatGPT create a Monday Morning Mulling to pair with the Final Friday Fix. We needed to make sure the solution worked and it did take one iteration to get a correct solution (always check!).

 **FFF Generator**
Monday Morning Mulling: August 2023 Solution


28 August 2023

As another week begins, we're back with the solution to our sun-soaked Excel challenge. This challenge was all about managing a summer camp's activities, ensuring none were overbooked. Let's unravel the solution, which ensures that each camper gets their fair share of fun under the sun!

Challenge Recap:
We had two Excel tables - Activities and SignUps. Our task was to create a dynamic formula to check if the number of sign-ups for each activity per day exceeded its maximum capacity.

Solution Breakdown:

- Setting Up Day Columns in the Activities Table:**
 - First, we need to prepare the Activities table for our calculations. In addition to the 'Activity' and 'Max Capacity' columns, we added columns for each day of the week (Monday to Friday). These columns will hold the count of sign-ups for each activity per day.
- The Formula:**
 - We used the `COUNTIFS` function in Excel, which counts the number of times multiple conditions are met.
 - The formula structure in each day column of the Activities table is as follows:

excel  [Copy code](#)

Once we had this, the final step was to go through and check that everything was formatted how we wanted it. While the GPT performed well at creating the article there were some issues. For example, we have a specific way of opening and closing the articles and while it did something similar, we adjusted it to what we use. Additionally, ChatGPT does not appear to work well with dates. Due to being based in the United States, the GPT set the date as August as that is summertime in the northern hemisphere. We also made minor grammatical adjustments along with adding a few pictures.

So how useful is this actually as a tool and how good was the output it provided? Many of you would have noticed that the problem wasn't as interesting as usual and that's due to the GPTs problem solving limitations. However, all things considered it does a remarkable job at mimicking the style of the Final Friday Fixes. This makes sense, as ChatGPT is a Large Language Model. It's built for producing Language, not for solving coding problems (though it is reasonable at that).

For this article, we let ChatGPT do most of the work but realistically, these tools are at their best when used with human input, working off each other. That could be with the SumProduct team coming up with problems and solutions and having it write out the article based upon that idea or getting it to help brainstorm complex problems for us to find the solutions to, or even just having it edit work articles we've already

written. Considering the small amount of input we had in creating the article, we believe it remains impressive what it was able to produce.

If you're interested in learning more, why not consider our introductory two half-day sessions that provide a practical primer for ChatGPT. It will revolutionise your world and take your job, wring it by its neck and transform it into something much more interesting. It's time to learn what all the fuss is about and embrace the new technology. This course shows you how.

These two sessions (each a half day) introduce you to the world of ChatGPT, explaining how to get it, what the differences are, how to get the best out of it, how to get the worst out of it. The sessions teach you the importance of phrasing your questions well ("prompt engineering") and explain what you need to be wary of, and how to substantiate the responses.

The first half day concentrates on understanding what ChatGPT is and isn't, giving you time to reflect before the second half day where tailored problems are workshopped and finessed. Most importantly, the course shows you how to make ChatGPT work for you and your business.

For more information, and to book, please send an email to training@sumproduct.com.

Key Benefits

- Save valuable time researching and solving your problems
- Learn how to review responses
- Improve content quality of your own work
- Understand the risks and limitations, such as its limited knowledge of recent events, potential plagiarism, ethical issues and inaccuracies.

Programme Outline

- How it works: introduction to neural networks
- How to access ChatGPT and use it
- The different versions of ChatGPT: pros and cons
- Limitations and issues
- Prompt engineering
- Privacy concerns
- Creating and verifying content
- Lesser known features: image recognition, knowledge bases, coding and working with Excel
- Workshopped examples.

It's aimed at anyone who wishes to "future proof" their career and tackle head on how an AI chatbot is going to transform their lives, both professionally and personally..

Upcoming SumProduct Training Courses

Location	Course	Course Date	Local Time	UTC	Duration
Virtual (Australia)	ChatGPT Part 2	4 March	13:30 - 17:00 AEDT	4 March, 02:30 UTC - 4 March, 06:00 UTC	1 Day
Melbourne Australia	Power Pivot, Power Query and Power BI	18 - 19 March	09:00 - 17:00 AEDT	17 March, 22:00 UTC - 19 March, 06:00 UTC	2 Days
Melbourne Australia	Excel Tips and Tricks	20 March	09:00 - 17:00 AEDT	19 March, 22:00 UTC - 20 March, 06:00 UTC	1 Day
Melbourne Australia	Financial Modelling	21 - 22 March	09:00 - 17:00 AEDT	20 March, 22:00 UTC - 22 March, 06:00 UTC	2 Days
Virtual (Australia)	ChatGPT Part 1	25 March	09:00 - 12:30 AEDT	24 March, 22:00 UTC - 25 March, 01:30 UTC	1 Day
Virtual (Australia)	ChatGPT Part 2	8 April	09:00 - 12:30 AEST	7 April, 23:00 UTC - 8 April, 02:30 UTC	1 Day

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. We've started going through the alphabet actions. It's time for **T** this month – make sure you pick up the **TAB**!

Keystroke	What it does
<u>T</u> AB	Tab forwards (next cell to edit)
ALT + <u>T</u> AB	Next application
CTRL + <u>T</u> AB	Next window
SHIFT + <u>T</u> AB	Tab backwards (previous cell to edit)
CTRL + ALT + <u>T</u> AB	Indent
CTRL + SHIFT + <u>T</u> AB	Previous window
CTRL + ALT + SHIFT + <u>T</u> AB	Outdent (<i>is that a word!?</i>)

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

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 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.

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 for a copy of the brochure or download it directly from www.sumproduct.com/training.