SumProduct

NEWSLETTER #117 - August 2022

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

Still here! We're into August now and not only is consulting, auditing and training continuing to keep us busy, we are delighted to announce we have been re-recognised for our efforts to the Excel and Power BI technical communities for our exploits, as two of our team remain Microsoft Most Valuable Professionals (MVPs).

So we'd better keep it up! This month, we detail the second of the new Excel Updates and wonder whether the Microsoft gang will continue to roll this out. If they do, we shall continue to report it. But as always, there is more!

We have (yet) another Beat the Boredom Challenge, as well as our usual articles on Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers and Power BI Updates. We continue with our imaginary Excel Functions (I imagine we will get past them eventually!), and we even have some of those **CTRL + SHIFT** keyboard shortcuts for your perusal.

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

Liam Bastick, Managing Director, SumProduct

MVP Renewal

SumProduct is pleased to announce **Tim Heng** and yours truly have been re-awarded Microsoft's Most Valuable Professional (MVP) award for Excel for 2022-23. It's a landmark for Tim too, as he racks up his fifth time in the Crazy Gang. Congratulations from us all here at SumProduct.

This award recognises exceptional technical community leaders from around the world who voluntarily share their high quality, real world expertise with others. Microsoft MVPs are a highly select group of experts representing technology's best and brightest who share a deep commitment to community and a willingness to help others.

Worldwide, there are over 100 million participants in technical communities; of these participants, there are c.4,000 active Microsoft MVPs. In Excel, we believe there are *c.*86 that have received this award.

Microsoft's MVP Award evaluates technical expertise and voluntary community contributions for the past year, considering the quality, quantity and level of impact of contributions. It's a difficult award to attain and just as difficult to retain.

At SumProduct, you can rely on our experience and willingness to help - simply drop us a line at contact@sumproduct.com should you need assistance.

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…*

Sometimes when modelling you need to identify the location of the nth occurrence of a character in a text string, perhaps to truncate the text or to manipulate it in some other fashion.

Character	l
Occurrence Number	3
Text	Hello Lesley
Result	10

This month's challenge is to write a formula in one cell that will identify the **n**th occurrence of a character in a text string. There are some requirements:

- the formula needs to be in just one cell (no "helper" cells)
- this is a formula challenge – no Power Query / Get & Transform or Text to Columns!
- the formula must work in **all** current versions of Excel (so no VBA, dynamic arrays, **LAMBDA**, **LET** or user defined functions)
- the model may be large or unstable, so no volatile functions are allowed
- the formula must be case sensitive. For example, in the illustration above the third occurrence of "l" in "Hello Lesley" is in position 10, *i.e.* "Hello Lesley" – the capital "L" is ignored.

If that all seems too easy, you may supplement the challenge by locating the **last** occurrence in the same text string too, subject to the same restrictions.

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

Charts and Dashboards

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

A Waterfall chart is quite different from the other charts, in that its purpose is to show the positive and negative movements from a starting value, that explain the difference in the ending value. These charts are being used more and more in the accounting and finance industry to explain variances in profitability, cash flow and account reconciliations. They are still relatively new to Excel first emanating in Office 2016. You can find out how to build one in earlier versions of Excel here.

For example, a business might want to know how the bank account has dropped, after having what was thought to be a great month of sales. You could build a chart that starts with the bank balance on the first of the month, showing:

- positive impacts on cash as a result of receipts from customers, cash sales, proceeds from the sale of assets, *etc.*
- negative effects on cash flow such as payments to suppliers, payrolls, loaning money to a third party, *etc.*

Hopefully by accounting for all these movements in and out of the bank, we will arrive at the bank balance at the end of the month.

Consider the following scenario. The management team is impressed that the company has exceeded its budgeted profit for the 2018/19 year *(say)* but would like a breakdown of the figures at a high level to start understanding how this occurred. The accounting department have put together a summary Profit & Loss Statement for 2018/19 as shown below:

	Budget	Actual
Total Income	**$ 1,316,581**	**$ 1,392,924**
less Expenses		
Payroll Costs	$ 913,413	$ 938,820
Vehicle Fleet	$ 21,668	$ 29,628
Operational Costs	$ 185,239	$ 140,448
Other Costs	$ 73,176	$ 93,228
Total Expenses	**$ 1,193,496**	**$ 1,202,124**
Net Profit / (Loss)	**$ 123,085**	**$ 190,800**

Management could work through this report, but a graphical representation of the figures would make the exercise easier. The first step is obviously to organise the data the way Excel needs it to construct a Waterfall chart. There is additional work involved in setting up the data table specifically the way Excel wants, but the result is worth the effort. There are only two columns required for the chart. The first column contains the categories or captions that will appear along the horizontal access, and the second column contains the figures.

The important element is the order of the rows. The starting value must be in the first row of the table, and the ending value must be in the last row. It is critical to ensure that the starting value plus or minus the figures running down the data table come to the ending value. If the figures don't flow correctly, the chart will be meaningless.

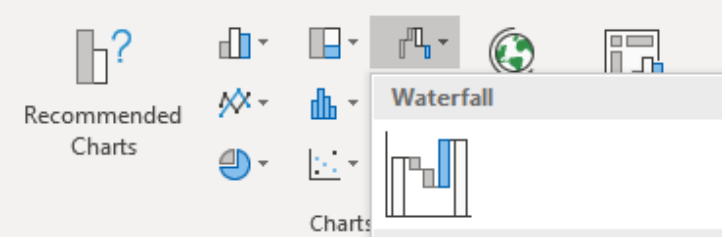
Management teams want to explain the movement from budgeted profit to actual profit, so budgeted profit is the starting value and actual profit is the ending value here. The numbers required in between are the movements in each of the income and expense categories. As such, our data table needs to be set up as follows:

Waterfall Chart Data for 2018/19			
Category	Budget	Actual	Chart Data
Budgeted Profit	$123,085		$123,085
Income	$1,316,581	$1,392,924	$76,343
Payroll Costs	$913,413	$938,820	($25,407)
Vehicle Fleet	$21,668	$29,628	($7,960)
Operational Costs	$185,239	$140,448	$44,791
Other Costs	$73,176	$93,228	($20,052)
Actual Profit		$190,800	$190,800

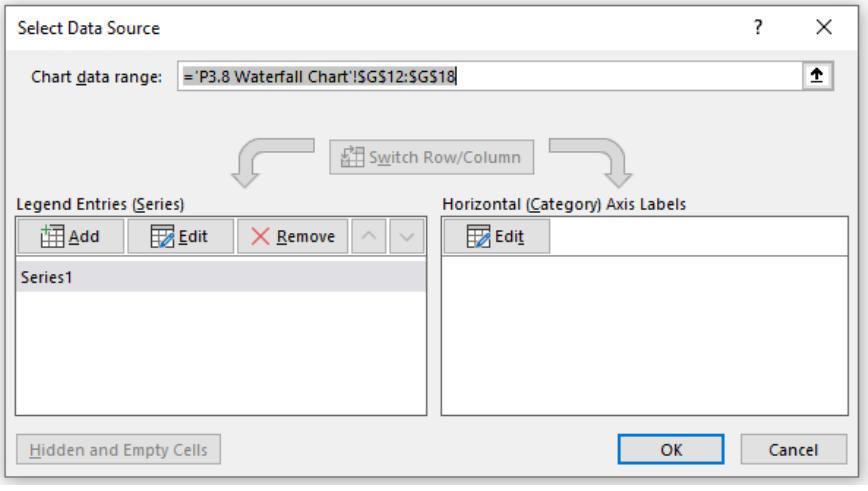
Note that the budgeted profit figure is in the top row of the table and the actual profit is at the bottom, and these values are simply transferred across to the 'Chart Data' column. Looking at income, the company actually earned more than it budgeted for, so the movement is a positive number, however it spent more on Payroll Costs and Vehicle Fleet costs, so the movement is negative. It is important when calculating the movements to determine whether the movement value is positive or negative. Also note that the sum of the figures in the right-hand column from the $123,085 to ($20,052) adds up to the final figure of $190,800.

With the data table ready, we may proceed to create the chart. However, we're going to do things a little differently this time. Up to this point, every time we've created a chart, the process has been to highlight the data and then pick the chart required. This time, we're going to choose the chart first and specify the data later.

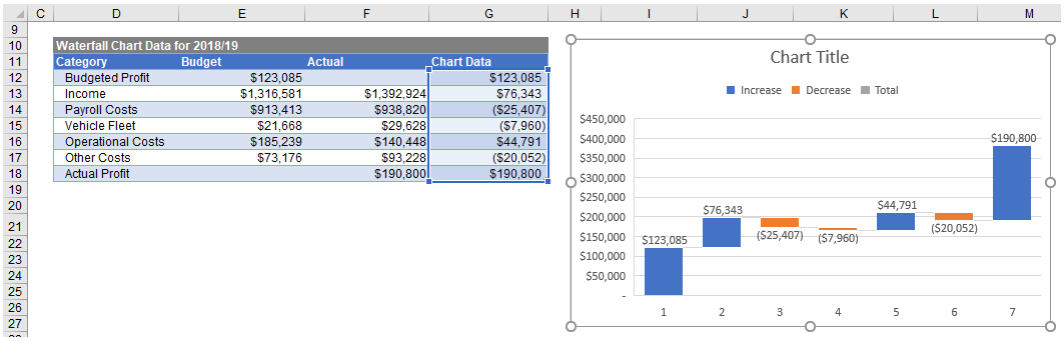
So, without selecting any data, let's go to the Insert tab on the Ribbon and create a Waterfall chart. There is only one Waterfall chart in Excel, and it is located under the third little icon along the top of the Charts section. Excel will return an empty chart box with a heading.



Excel should automatically switch to the Design tab of the Ribbon. If it has not, we may just click on the empty chart box and go to the Design tab. Then, click on the 'Select Data' button and the following dialog will appear:

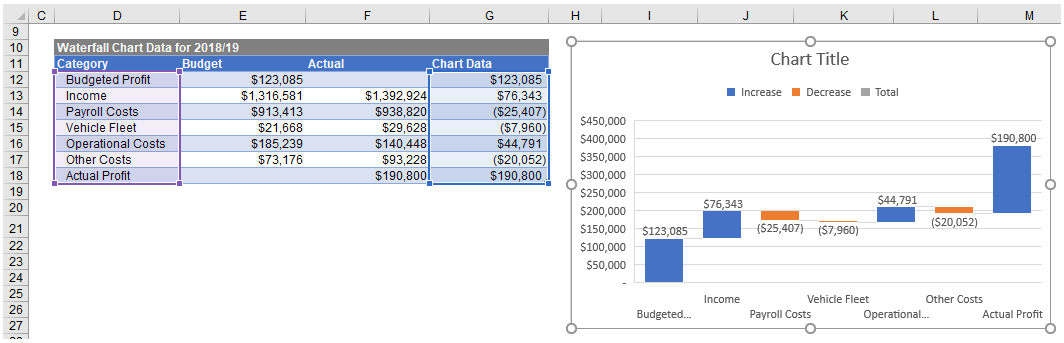


We're now going to provide Excel with the information it needs to create the chart. The first range it is asking for as I can see in the top of the window is the 'Chart data range'. Referring to the data table shown above, the data range is in cells **G12:G18**, so I highlight these cells and Excel applies the data automatically to the chart box.



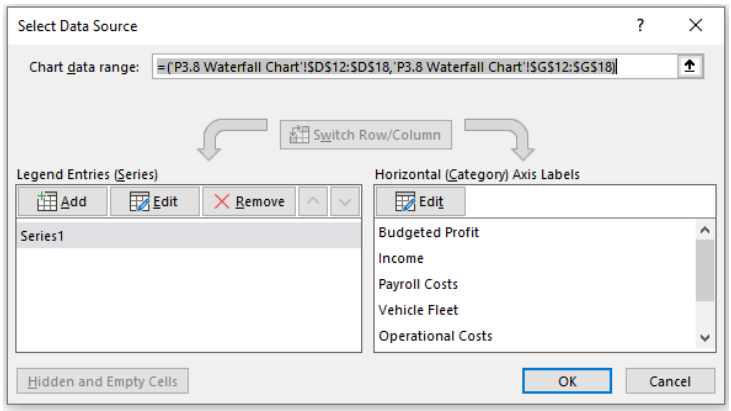
Waterfall Chart Data for 2018/19			
Category	Budget	Actual	Chart Data
Budgeted Profit	$123,085		$123,085
Income	$1,316,581	$1,392,924	$76,343
Payroll Costs	$913,413	$938,820	($25,407)
Vehicle Fleet	$21,668	$29,628	($7,960)
Operational Costs	$185,239	$140,448	$44,791
Other Costs	$73,176	$93,228	($20,052)
Actual Profit		$190,800	$190,800

Next, we need to specify the category labels to replace the numbers one to seven [1 to 7] on the horizontal axis of the chart. Our labels are in cells **D12:D18** so I click on the Edit button under the heading 'Horizontal (Category) Axis Labels' and highlight cells **D12:D18** then click OK, and again Excel will apply these to the chart.

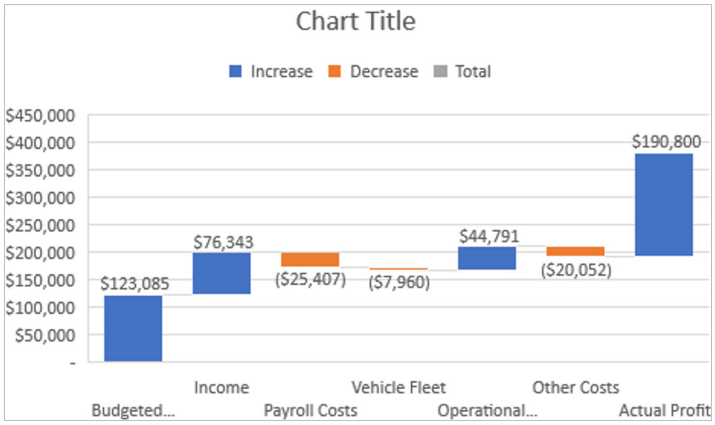


Waterfall Chart Data for 2018/19			
Category	Budget	Actual	Chart Data
Budgeted Profit	$123,085		$123,085
Income	$1,316,581	$1,392,924	$76,343
Payroll Costs	$913,413	$938,820	($25,407)
Vehicle Fleet	$21,668	$29,628	($7,960)
Operational Costs	$185,239	$140,448	$44,791
Other Costs	$73,176	$93,228	($20,052)
Actual Profit		$190,800	$190,800

The 'Select Data Source' window should look like this:

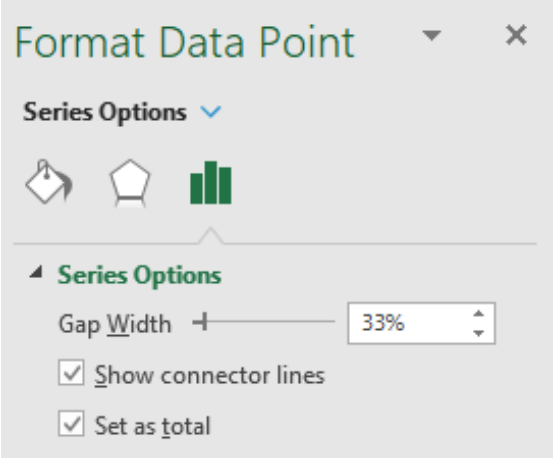


With the data applied to the chart, we have the initial Waterfall Chart as the one below:

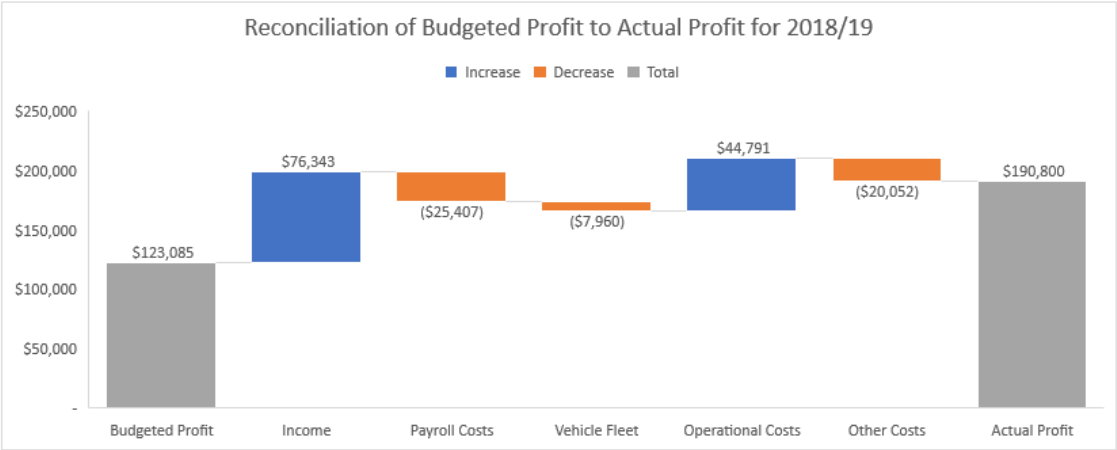


This chart doesn't look that nice at the moment: you could argue there are a few more steps to finalise the Waterfall chart.

The legend indicates that blue columns reflect increases, orange columns reflect decreases and grey columns indicate totals – but there are no grey columns. We need to tell Excel that the budgeted profit and the actual profit figures are the starting and ending values respectively and therefore set them as totals. To do this, click on the data series in the chart and then specifically click on the blue column relating to budgeted profit. Notice that all the other columns fade in colour and the budgeted profit remains in full colour. Right-click on the budgeted profit column and choose 'Format Data Point', under 'Series Options', tick the box 'Set as total'.



The budgeted profit column should now turn grey indicating it is a total. With the 'Format Data Point' panel still open, we now click on the actual profit column. This column should now be the only one in full colour. Again, we'll tick the 'Set as total' option and this column should also turn grey. You may also notice that this column has now dropped on the chart so that it touches the horizontal axis.



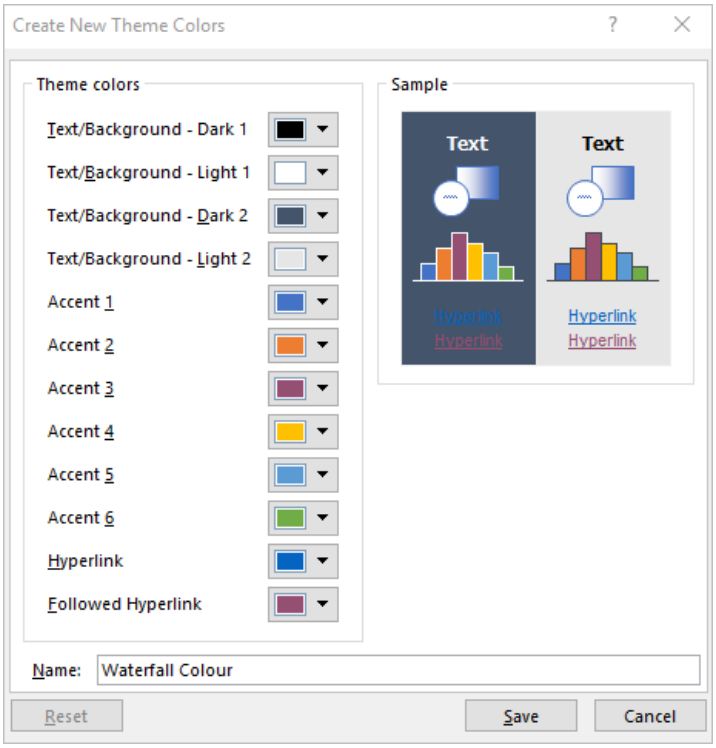
This chart portrays visually the same information as the summary Profit & Loss Statement prepared by accounting but enables the viewer to understand the information quicker and more simply. Basically, from the budgeted profit figure we started with, the company made more income than expected and saved on Operational Costs, however it did spend more on the other expense categories, though this unplanned additional expenditure could be the result of earning so much more income... but that analysis is for management to work their way through!

In case you were wondering, Excel has deliberately used blue and orange instead of the "more traditional" colours of green and red, purely because blue and orange overcome issues for people who may have colour vision deficiencies.

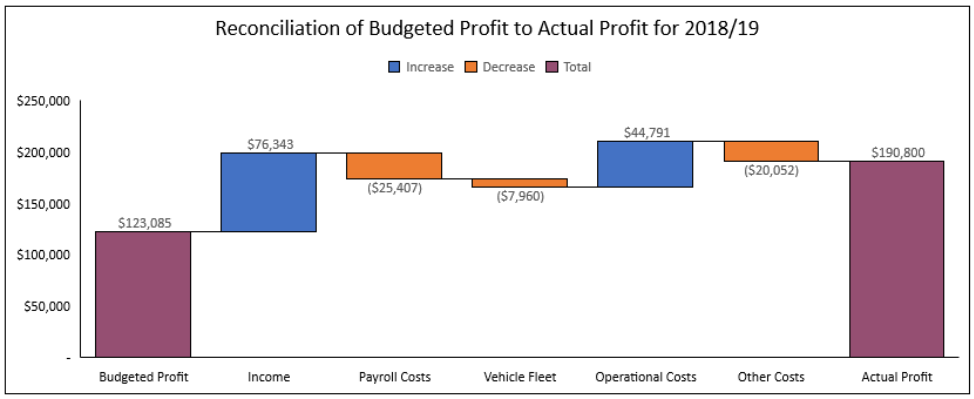
Finally, let's add some formatting to complete the chart.

The grey columns are a little boring, so let's change them to purple (hoping everyone may still distinguish the colours!). Sounds simple but this is actually more complex than it appears. If we specifically change the two profit figure columns to purple, this will not change the colour for the series, so the legend will not reflect the change made. As far as Excel is concerned, the colour for the totals is still grey. When we change the colour of two particular columns, it is not changing the setting for the series. In the Series Options setting, it is only showing one fill that can be applied to the series, so how does Excel know to colour increases in blue, decreases in orange and totals in grey?

The answer lies in the colour settings for Excel. Go to the 'Page Layout' tab on the Ribbon, click on the arrow next to Colors and choose 'Customize Colors...'. Here, Accent 1, Accent 2 and Accent 3 contain the settings for the three colours in the Waterfall Chart. If we wish to change the colours for the chart, we may modify them here. Let's imagine we're happy with blue and orange, but we want the totals in purple, so we change Accent 3 to purple, give this modified colour scheme a name at the bottom of the window like "Waterfall Colour" and click OK. The new colour scheme will be reflected on the chart (If it doesn't, just click once on the chart for the change to take effect).



You may also change the font colour of all chart element to black and get them bordered too. The finished product would then look as follows:



More next month…

Visual Basics

We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month, we consider For Each … Next loops.

Loops are a powerful tool in programming to repeat a sequence of instructions. **For** loops have been covered previously; they are useful for performing actions a fixed number of times. But what if you want to perform an action to every object in a set?

For Each … Next loops are a great way to cycle through sets, like an array or a range. Sometimes, we don't know the size of the set, such as in a table. Using **ListObject.ListRows.Count** is a way to find the number of rows in order to use the simple **For** loop but using **For Each … Next** more clearly illustrates that the instructions are happening to *every* row.

Our example this month depicts sales data, in a table called **Tbl_Sales**:

	A	B	C	D
1	Date	Rep	Sales	Reps Cumulative Sale
2	4/06/2017	Jill	$213	
3	5/06/2017	Jill	$840	
4	4/07/2017	Jill	$955	
5	7/07/2017	Jack	$128	
6	5/08/2017	Jill	$571	
7	7/08/2017	Jill	$742	
8	9/08/2017	Jill	$648	
9	23/08/2017	Jill	$723	
10	25/09/2017	Jill	$551	
11	26/09/2017	Jill	$958	
12	5/10/2017	Jack	$928	
13	12/10/2017	Jack	$602	

Let's calculate each sales rep's cumulative sales as time goes by, *i.e.* we shall calculate the total sales they each made to the date of that sale, *e.g.* Jill made $213 + $840 + $955 = $2,008 cumulative sales on the 4th of July (admittedly this can be achieved quite easily with a **SUMIF** formula with a moving range).

Here's the code:

```
Option Explicit

Sub ForEachLoopExample()

    Dim SalesTable As ListObject
    Set SalesTable = Range("Tbl_Sales").ListObject

    Dim TableRow As ListRow
    Dim CurrentRep As String
    Dim CurrentSale As Long

    Dim JackTotal As Long
    JackTotal = 0
    Dim JillTotal As Long
    JillTotal = 0

    For Each TableRow In SalesTable.ListRows

        CurrentRep = TableRow.Range(1, 2).Value
        CurrentSale = TableRow.Range(1, 3).Value

        Select Case CurrentRep

            Case "Jack"
                JackTotal = JackTotal + CurrentSale
                TableRow.Range(1, 4).Value = JackTotal

            Case "Jill"
                JillTotal = JillTotal + CurrentSale
                TableRow.Range(1, 4).Value = JillTotal

        End Select

    Next

End Sub
```

Notice that when using **Option Explicit** to force declaration of variables, the child of the parent object must also be declared, *e.g.* here, for a table, using **ListRows**.

	A	B	C	D
1	Date	Rep	Sales	Reps Cumulative Sale
2	4/06/2017	Jill	$213	$213
3	5/06/2017	Jill	$840	$1,053
4	4/07/2017	Jill	$955	$2,008
5	7/07/2017	Jack	$128	$128
6	5/08/2017	Jill	$571	$2,579
7	7/08/2017	Jill	$742	$3,321
8	9/08/2017	Jill	$648	$3,969
9	23/08/2017	Jill	$723	$4,692
10	25/09/2017	Jill	$551	$5,243
11	26/09/2017	Jill	$958	$6,201
12	5/10/2017	Jack	$928	$1,056
13	12/10/2017	Jack	$602	$1,658
14	4/11/2017	Jill	$809	$7,010
15	28/11/2017	Jack	$103	$1,761
16	11/01/2018	Jack	$161	$1,922

This is a pretty basic example of how each row can be examined in a table, but **For Each … Next** can also be used for arrays and cell ranges too.

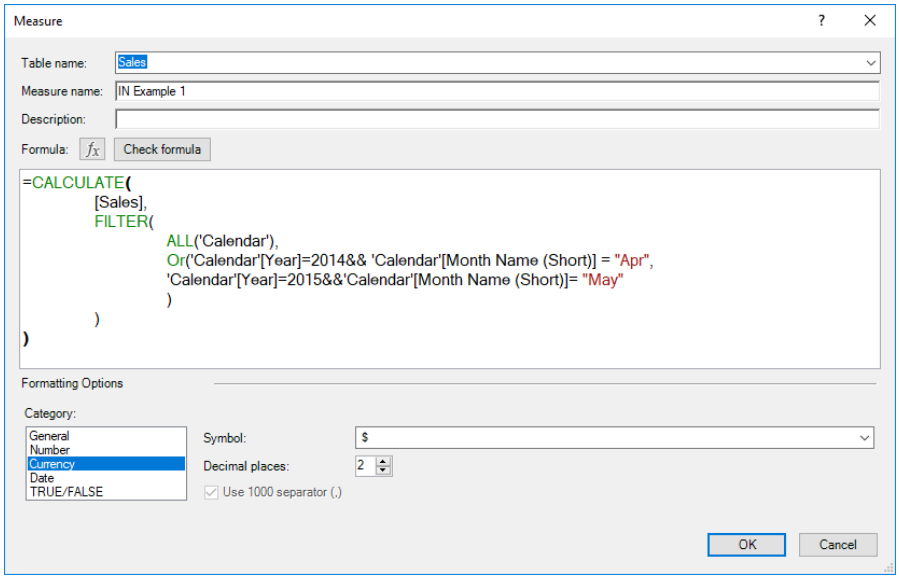
More next time.

Power Pivot Principles

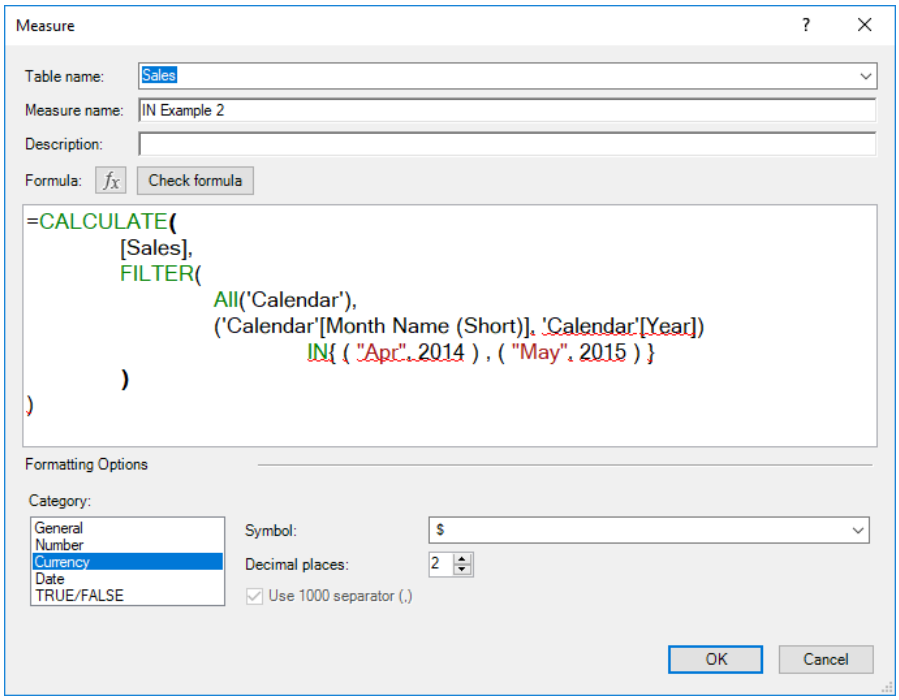
*We continue our series on the Excel COM add-in, Power Pivot. This month, we look a little closer at the **IN** function.*

Last month we looked at concatenating filters to combine multiple criteria into one **CALCULATE** function. In this newsletter, we will improve on that concept using the **IN** function.

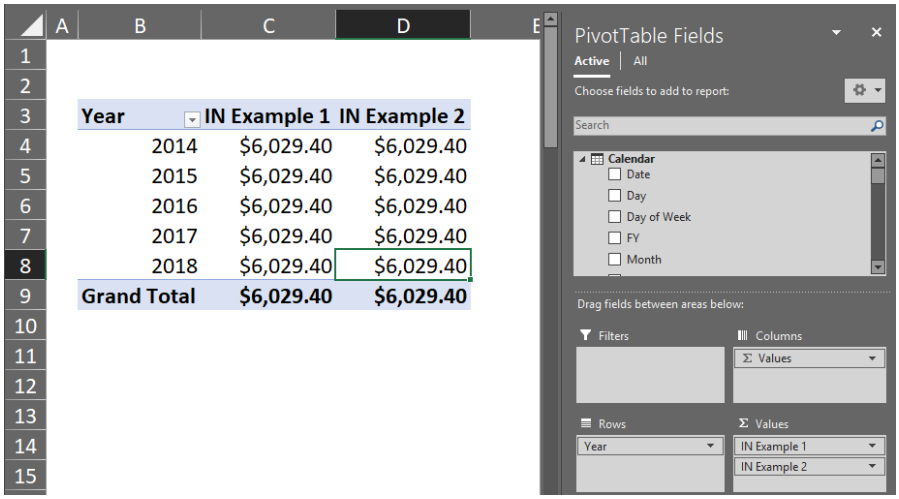
As a recap, we used the following measure to illustrate combining two filter criteria into one measure:



This measure achieves our desired result; however, it is a little complicated. Additionally, what will we do if we have a third criteria that we want to add to the filter? Let's see if we can do better using the **IN** function.



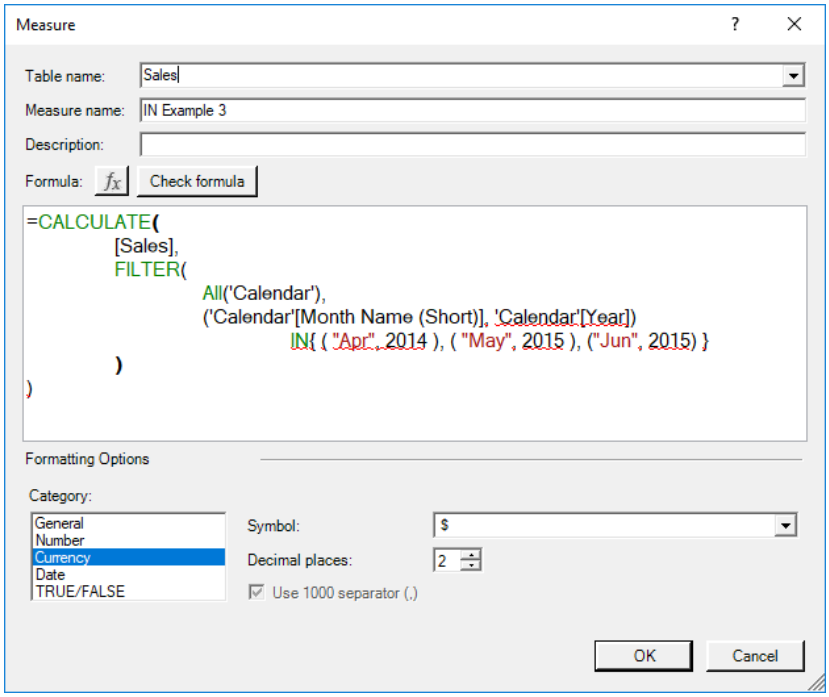
After using the **IN** function, we have achieved the same result in lesser steps and improved readability!



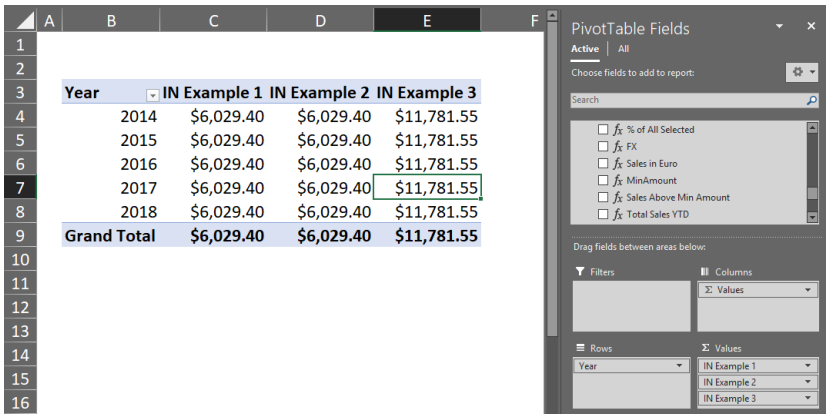
Year	IN Example 1	IN Example 2
2014	$6,029.40	$6,029.40
2015	$6,029.40	$6,029.40
2016	$6,029.40	$6,029.40
2017	$6,029.40	$6,029.40
2018	$6,029.40	$6,029.40
Grand Total	**$6,029.40**	**$6,029.40**

(Sorry about the fact all of our data has the same values per year – it is working, we assure you!)

Not only that, but we can have more than two criteria in the filter. Consider the following:



This results in:



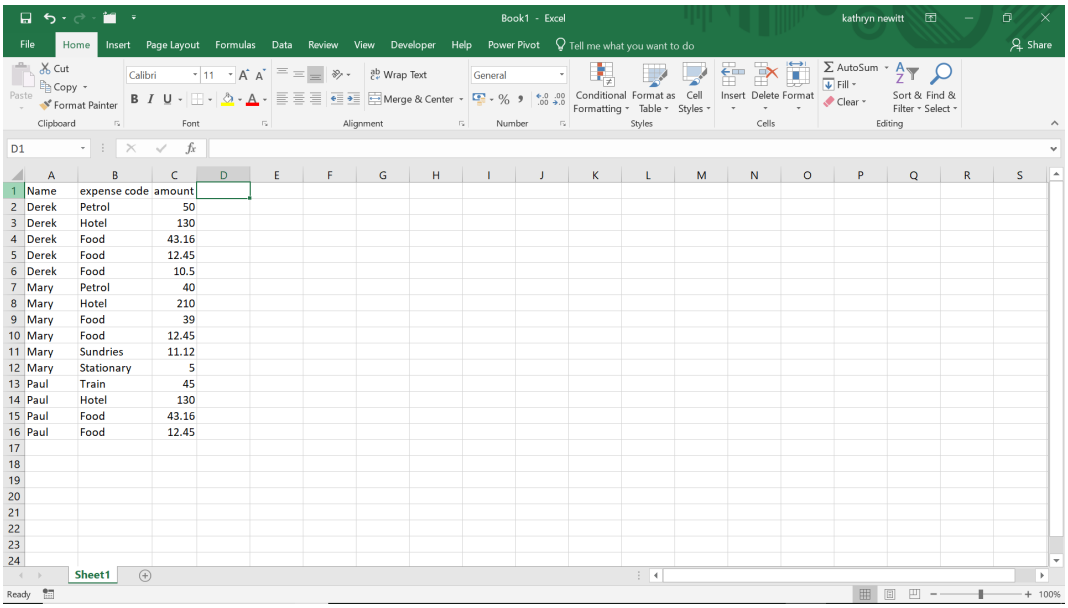
Year	IN Example 1	IN Example 2	IN Example 3
2014	$6,029.40	$6,029.40	$11,781.55
2015	$6,029.40	$6,029.40	$11,781.55
2016	$6,029.40	$6,029.40	$11,781.55
2017	$6,029.40	$6,029.40	$11,781.55
2018	$6,029.40	$6,029.40	$11,781.55
Grand Total	**$6,029.40**	**$6,029.40**	**$11,781.55**

More *Power Pivot Principles* next month.

Power Query Pointers

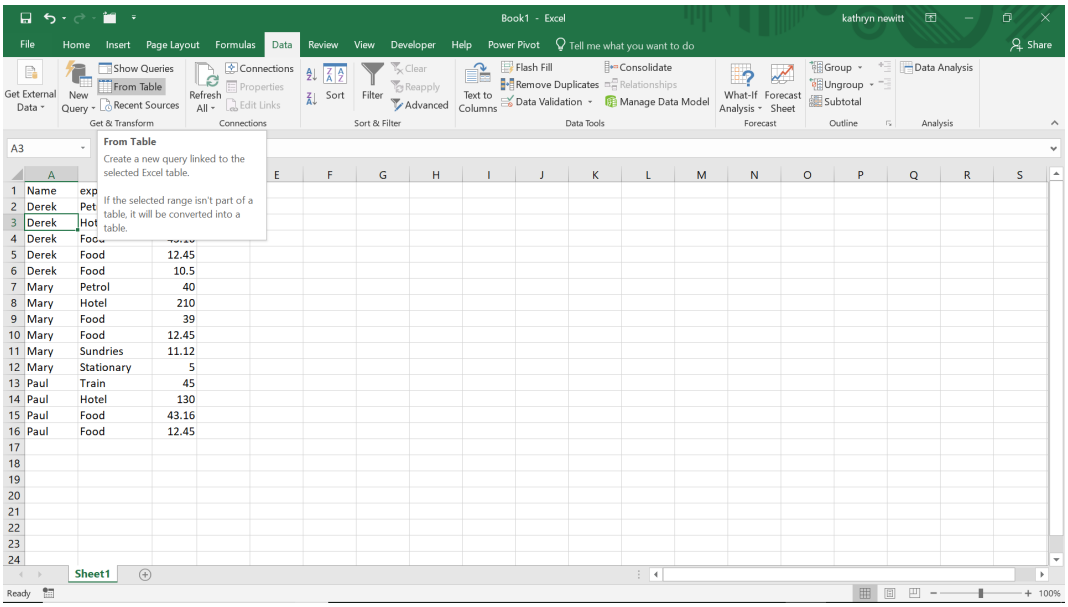
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. *If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at cross joins.*

This month, we have a list of expenses that our fictional salespeople have incurred in a month. We'd like to expand this to estimate what they might spend in the next 10 months.

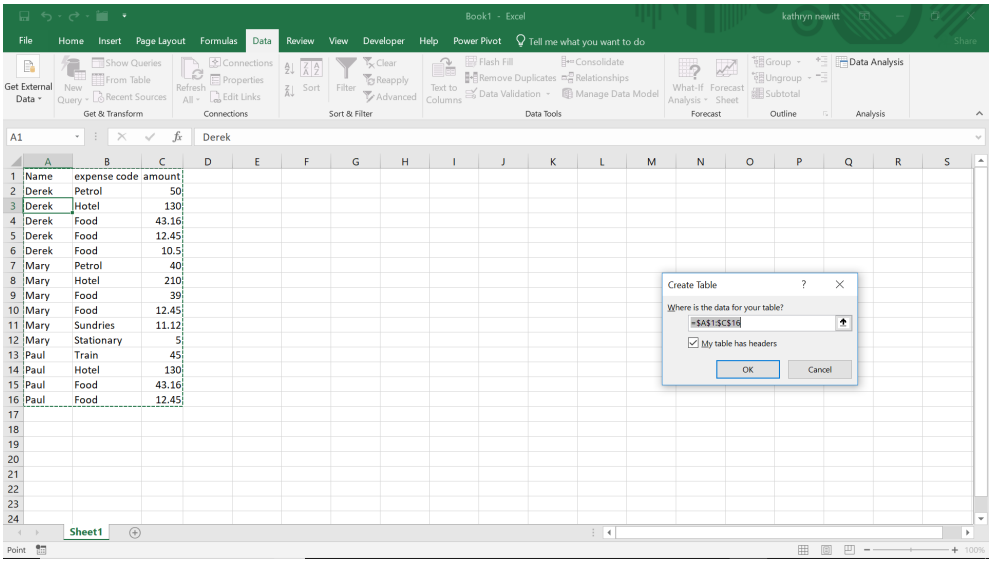


	A	B	C
1	Name	expense code	amount
2	Derek	Petrol	50
3	Derek	Hotel	130
4	Derek	Food	43.16
5	Derek	Food	12.45
6	Derek	Food	10.5
7	Mary	Petrol	40
8	Mary	Hotel	210
9	Mary	Food	39
10	Mary	Food	12.45
11	Mary	Sundries	11.12
12	Mary	Stationary	5
13	Paul	Train	45
14	Paul	Hotel	130
15	Paul	Food	43.16
16	Paul	Food	12.45

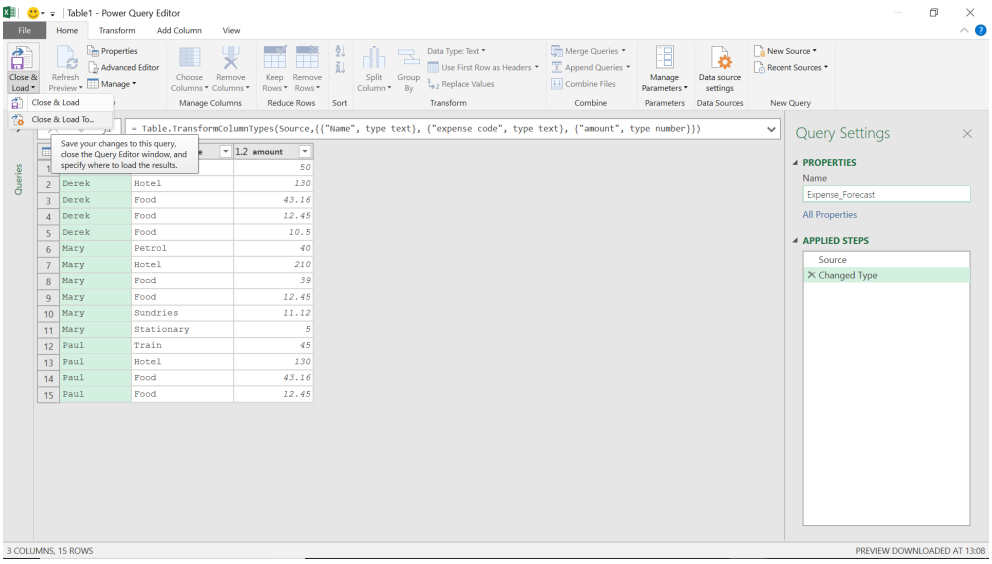
We'll start by creating a query from our expenses data, by choosing 'From Table' in the 'Get and Transform' section of the 'Data Tab':



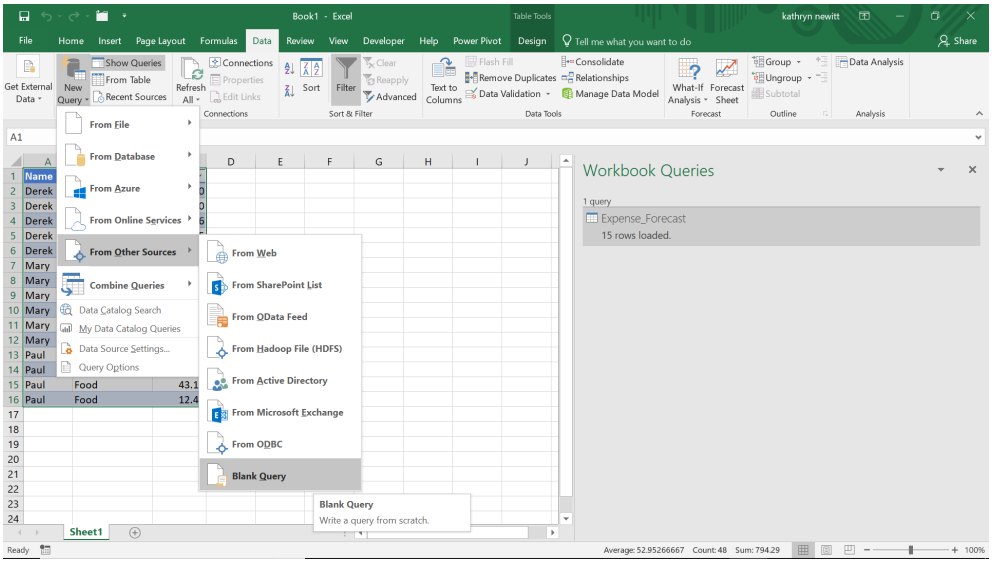
We may confirm where our data is, and that the first line contains the headers. It doesn't matter that the data is not currently in a table: Power Query will convert it to a table as part of the process.



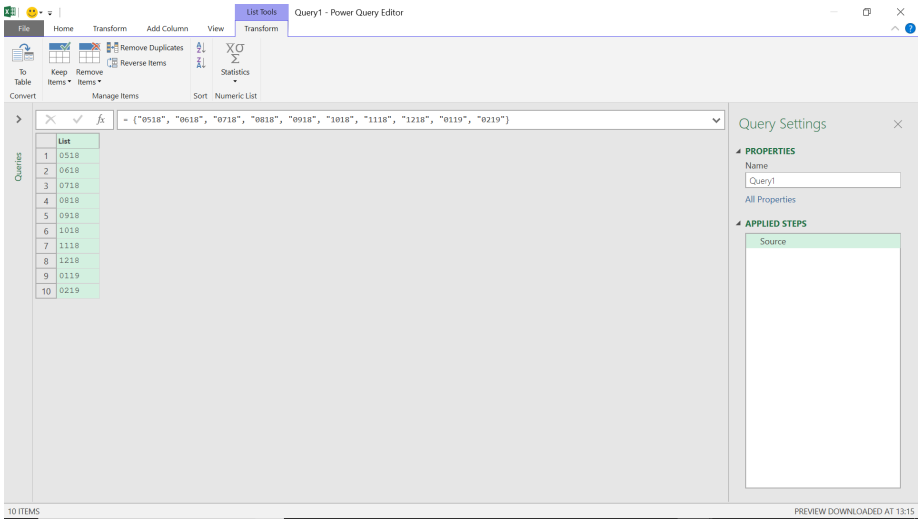
The query is shown below. Let's call it '**Expense_Forecast**' and save it as connection only.



Now, we wish to create a list of months – this is easy enough to do by creating a new blank query from the 'New Query' option in the 'Get and Transform' section of the 'Data' tab, *viz.*



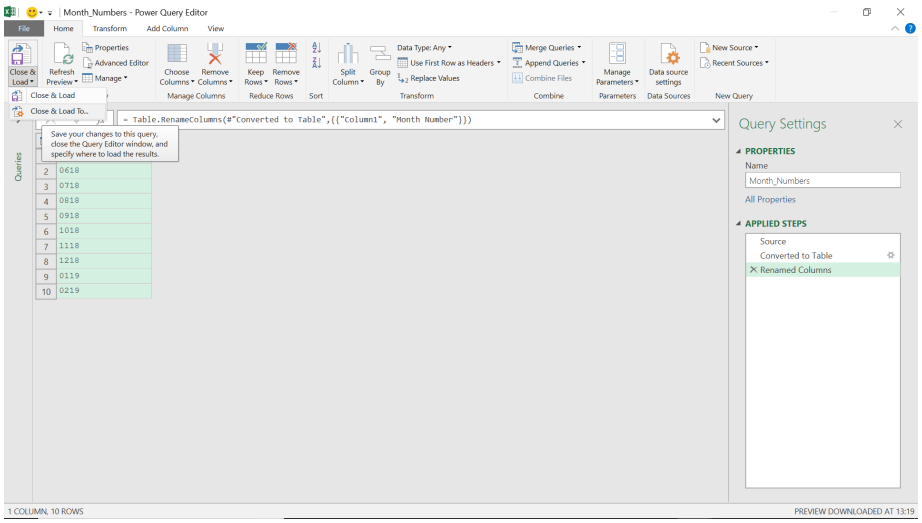
We may create a list using basic list functionality (more on this may be found at https://www.sumproduct.com/blog/article/power-query-blogs/power-query-birthday-lists).



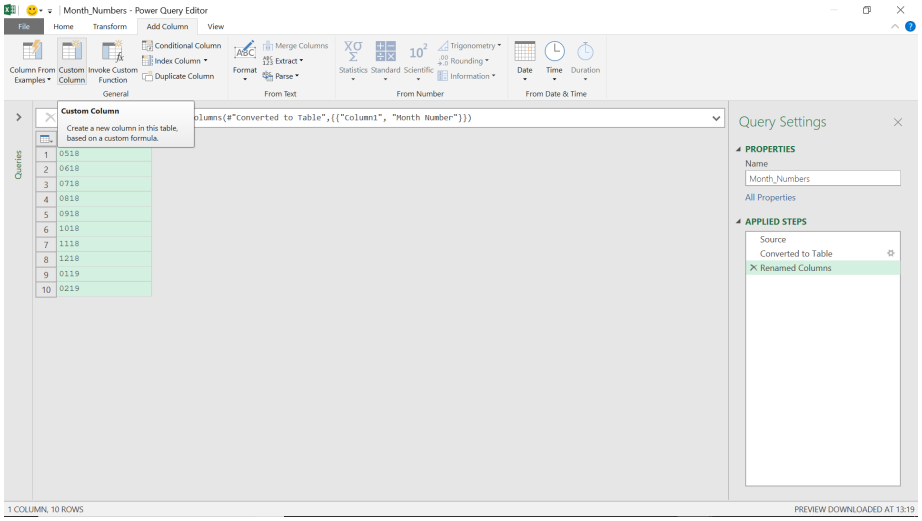
The formula used is

= {"0518", "0618", "0718", "0818", "0918", "1018", "1118", "1218", "0119", "0219"}

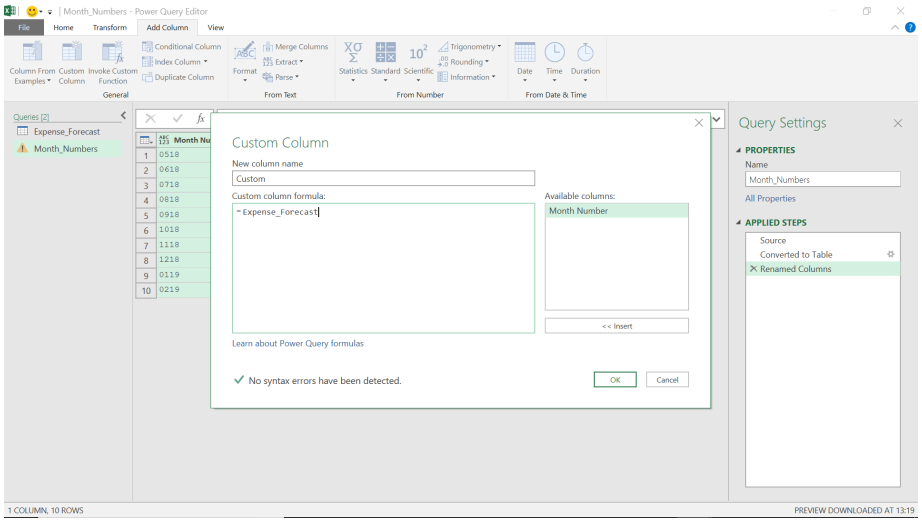
In order to link this list to our **'Expense_Forecast'** table, we want this to be a table too, so we'll convert it using the 'To Table' option in the 'Convert' section of the 'Transform Tab'. We'll rename the column and call the query '**Month_Numbers**'. You may save this query as connection only too.



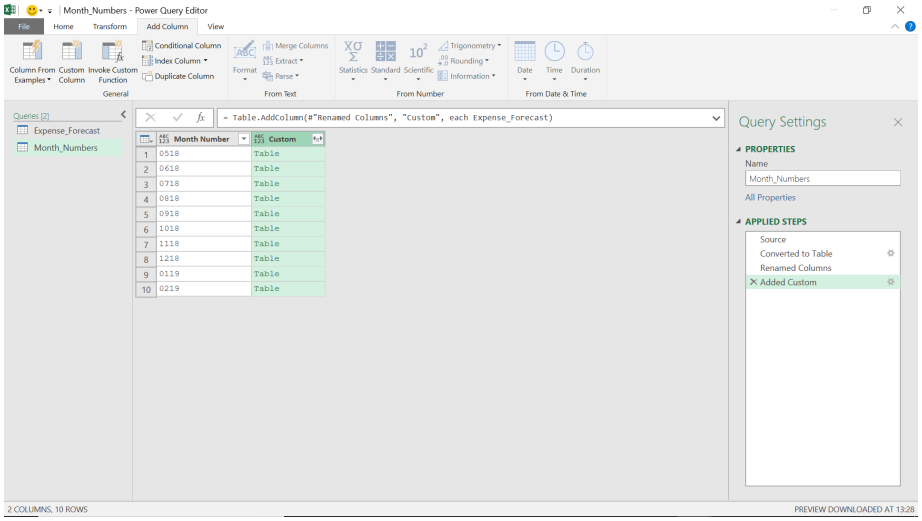
We now need to combine this information. Let's decide to do this by adding the expense information to the '**Week_Numbers**' query. We may do this by going to the 'Add Column' tab. Let's choose to add a 'Custom Column'.



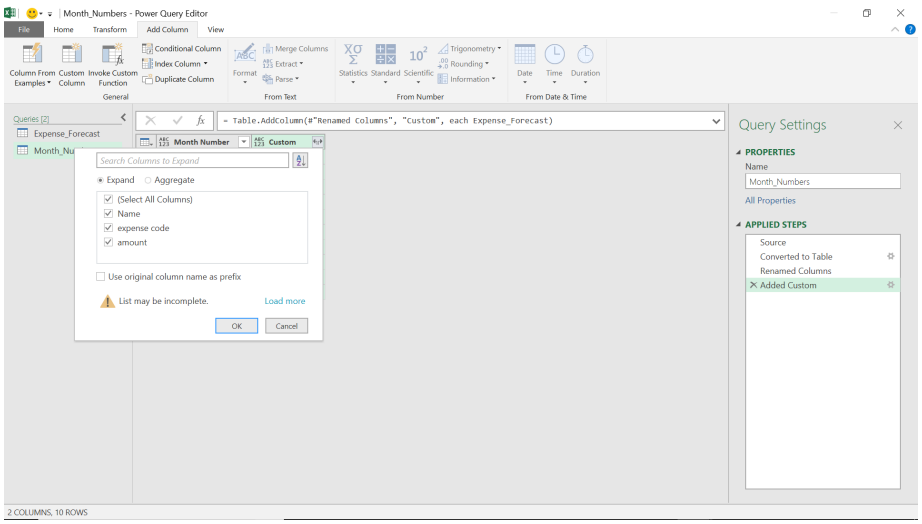
If we expand the Queries pane to the left of the screenshot, you may use the name of our other query in the column.



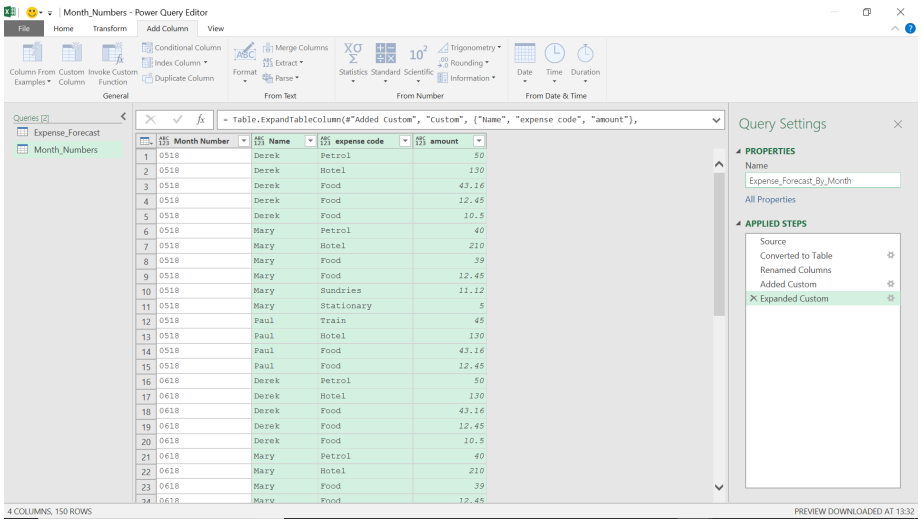
This will create a new column for the '**Expense_Forecast**' data.



We may now expand this column to show the data in the table – in this case, let's choose all the available data. We will untick the 'Use original column name as prefix' option (no, we don't know why it's the default either!).



We can then see the basic expense forecast for each employee and category for the next 10 months. We may then upload this data to Excel or PowerPivot and further investigate with the effects of inflation and predicted workloads.



More next month!

Improved Power BI Experience in Excel

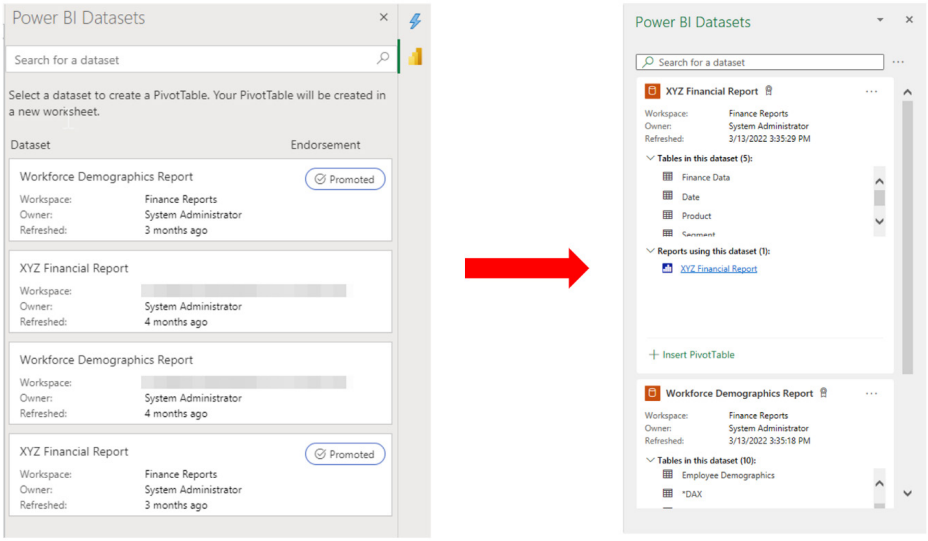
July saw several improvements to the Power BI experience in Excel that will help many easily search data artifacts (including Power BI datasets and Power BI datamarts) and uncover better data insights all inside a spreadsheet environment. These improvements are rolling out to users in phases and should have been completed before you read this!

With these new changes to the Power BI experience in Excel, you may:

- more easily search for data in a new user-friendly interface
- discover new data artifacts with a deep link to the Power BI Data hub where you can find recommended datasets and other trusted data in your organisation
- derive additional insights from your data with direct links to Power BI reports from inside Excel.

Power BI has become a common central data repository for many organisations and now you can access all that data right inside Excel.

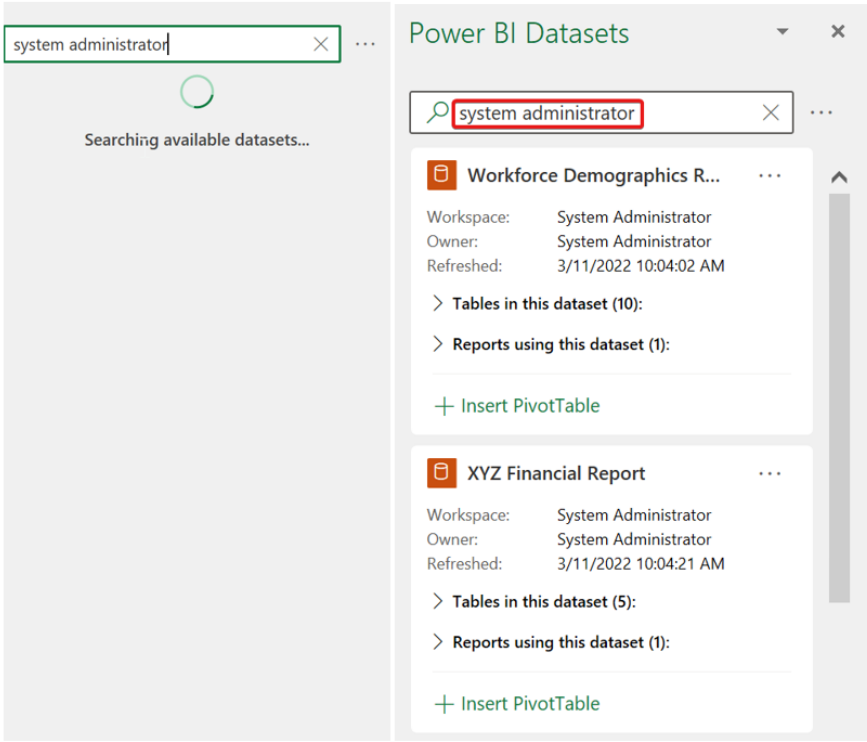
The new Power BI User Interface (UI) in Excel strives to be an improvement from the prior interface. Now it is easier for you to discover the right Power BI data for your analysis in Excel. Also, it now shows the names of the tables contained in a particular dataset so you may better understand your data.



You can easily search for data using any of the following parameters:

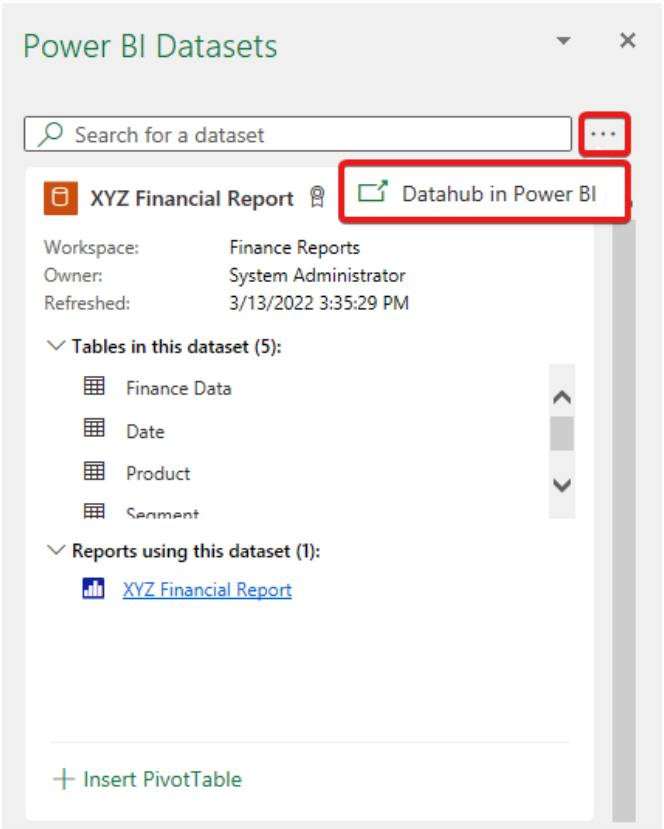
- Dataset name
- Dataset Owner
- Workspace Name.

Any search will return an exact (or near exact match) for the dataset(s) you are looking for. For example, if the dataset you are looking for is owned by system administrator and you search using that term, all datasets that you have access to that are owned by system administrator will be returned.

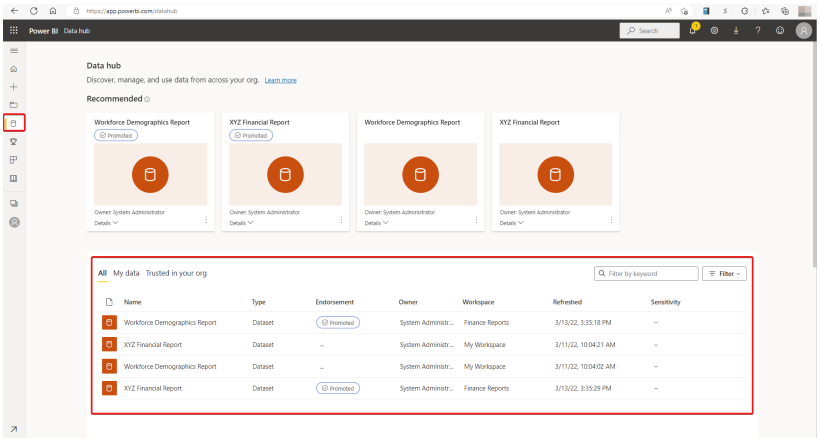


The 'Refreshed' field shows you the exact timestamp for the day and time when the data was last updated so you know you are working with the most up-to-date data.

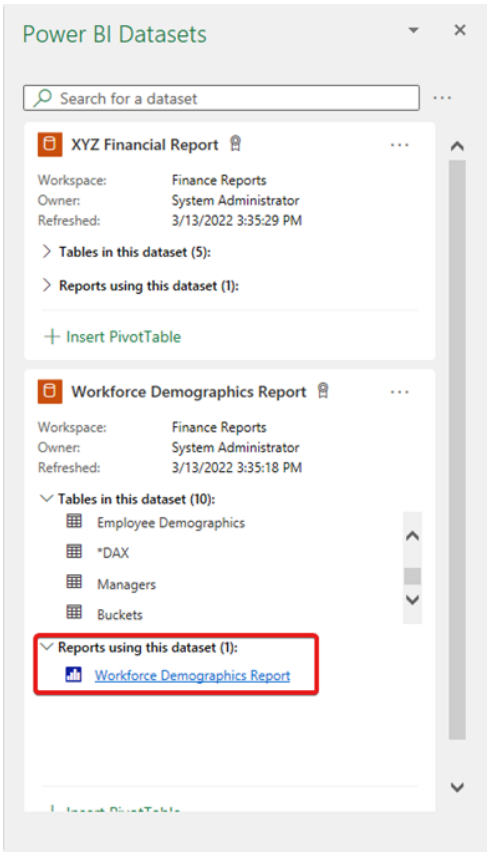
Furthermore, in the Power BI datasets pane in Excel, you can now navigate to the Data hub in the Power BI Service where you can view all Power BI datasets and other data artifacts you have access to. To navigate to the Power BI data hub from Excel, select the ellipsis (...) beside the search bar in the Power BI datasets pane and click 'Datahub in Power BI'. A new browser window will open that then takes you to the Power BI Service.



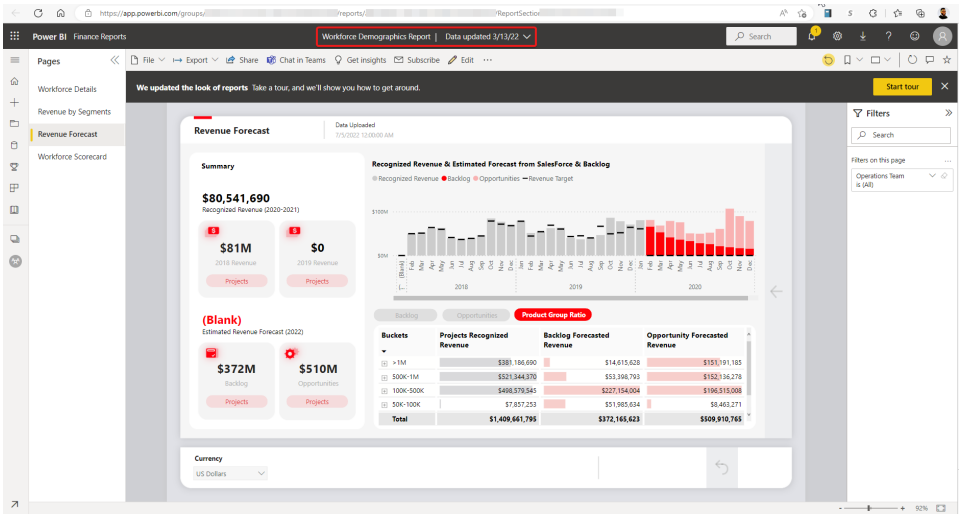
In the Power BI Data hub, you can discover recommended datasets based upon the activity of people you work closely with and find datasets trusted in your organisation. From any dataset in the Power BI Data hub, you can create new reports, download the PBIX file or create an 'Analyze in Excel' workbook in OneDrive to share with others.



Also, while you can connect directly to a Power BI dataset from Excel and build connected reports in the spreadsheet using PivotTables, you may also want to discover Power BI reports that use the same dataset either to get additional insights or validate your analysis in Excel. Under the 'Reports using this dataset' item in the pane, you can now see the Power BI reports that use a particular dataset.



Each of those reports are direct links to the Power BI Service and clicking on any of the reports will open a new browser tab where you may view the report.



Power BI Updates

The latest updates see Error Bars become Generally Available, combined with updates for Metric visuals and Datamarts, and improvements regarding query performance and the data hub. But, as always, there's more. The full list is as follows:

Reporting

- Error Bars now Generally Available
- Filled Map for Azure Maps visual
- Composite models on Power BI Datasets and Analysis Services now in Preview
- Bold / italics / underline for text inside the header ToolTip
- Metric visuals updates

Modelling

- New DAX function: **NETWORKDAYS**
- Support multi-role Row Level Security in composite model
- Query performance improvement

Data connectivity and preparation

- Connect to Datamarts (Preview)
- Display name support for the Dataverse connector
- BitSight Security Ratings (Connector Update)
- Databricks (Connector Update)
- Eduframe Reporting (New Connector)
- Funnel (Connector Update)

Service

- Data hub improvements: data preview and export

Mobile applications

- Introducing Data in Space
- Track metrics with multiple milestones and targets from your mobile app
- Find content that's relevant to you (Windows app)

Embedded Analytics

- Accessibility enhancements for embedded reports
- Copy visual as an image in embed for your organisational scenarios

Developers

- Drill API

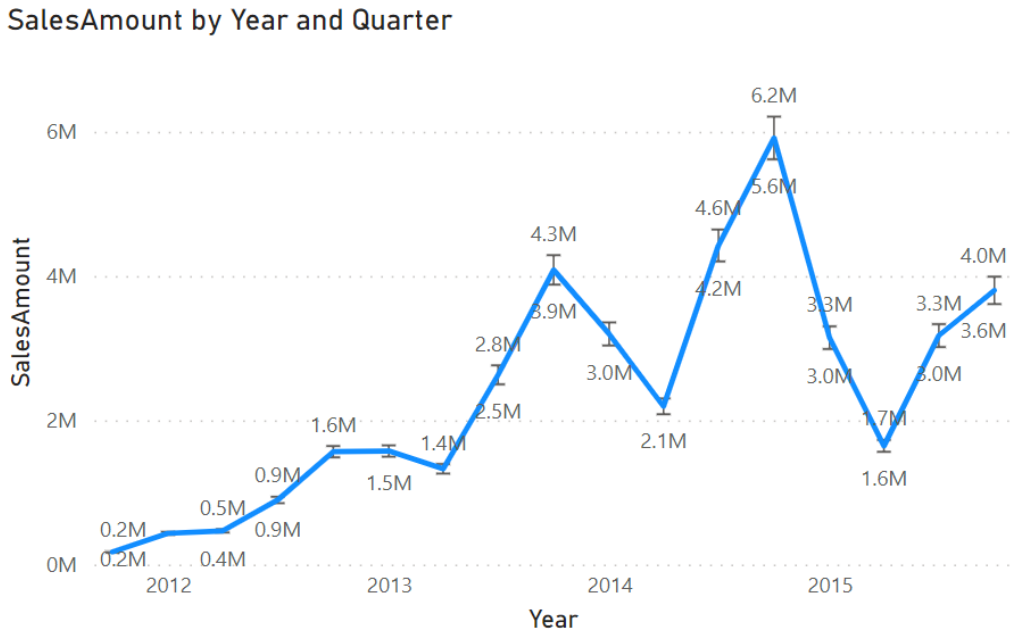
Visualisations

- New Power BI visuals' category list in AppSource
- New visuals in AppSource
- Drill Down Timeline PRO by ZoomCharts
- SMART KPI List by Nova Silva.

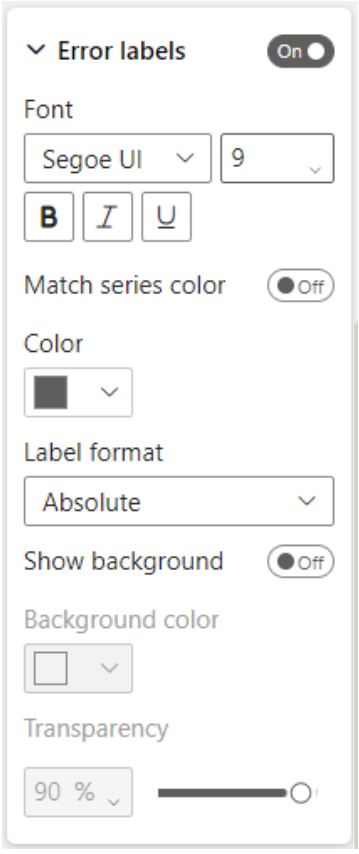
Let's now go through each in turn.

Error Bars now Generally Available

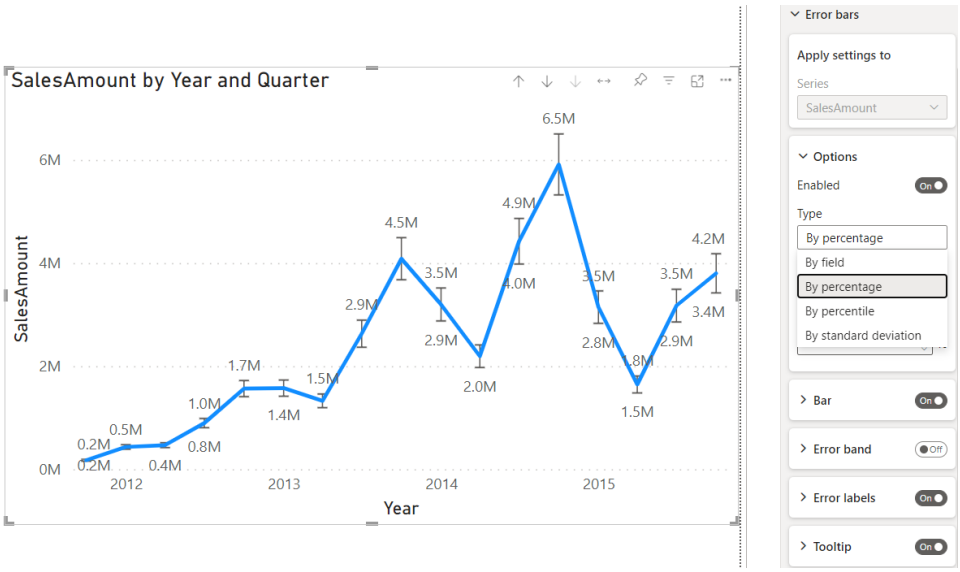
The error bars feature has finally become Generally Available in this update. Further, this month's edition also adds a variety of new capabilities. For a stary, you'll notice a new option to enable data labels for your error bars. Similar to data labels on your values, you can now get the actual values of the upper and lower bounds directly on the visual too.



You can enable these in the 'Error labels' card of the Formatting pane, where you may also customise font styles and colour, just as you may with regular data labels. There is also a 'match series color' option to allow you to colour the error labels with the colour of their associated data series.



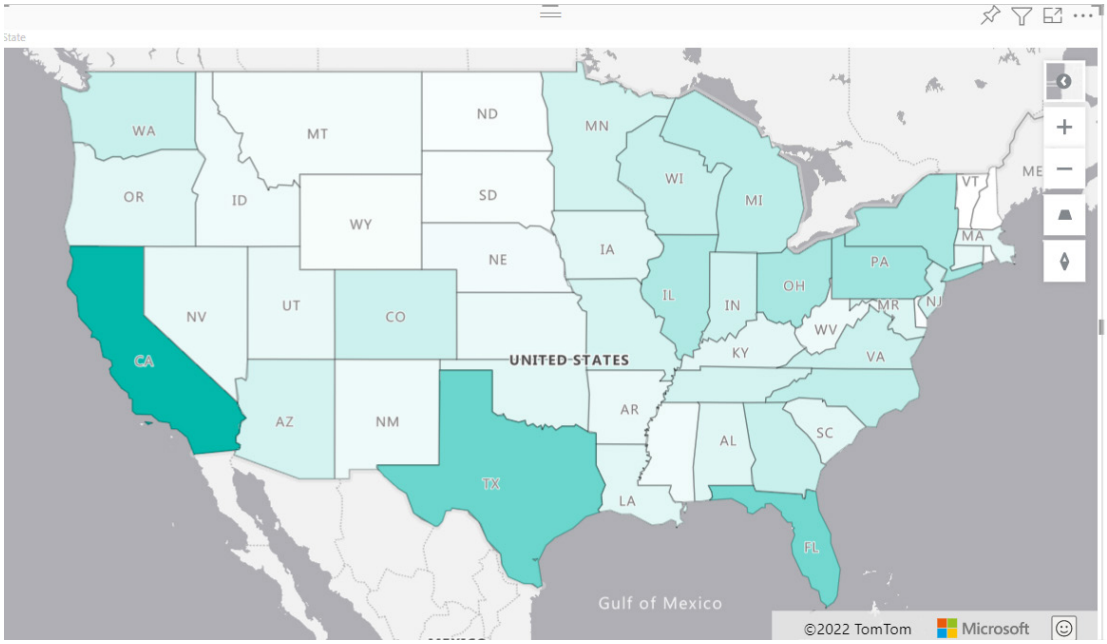
This update also adds new type options for the error bars. Now, on top of being able to create error bars based upon upper and lower bound options you set, you may also base upper and lower bounds on percentage, percentile and standard deviation options as well. Choosing Percentage will show you upper and lower bounds as calculated from the displayed value of your value field and choosing Percentile or 'Standard deviation' will show you bounds calculated from the aggregated data points at each **x**-axis value on your chart.



Finally, this update includes a new 'make symmetrical' option for your 'by field' error bars. 'Make symmetrical' allows you to choose just one relative measure for your error bars and will mirror that field in both directions. For cases where your upper and lower bounds are the same, this will help you require one less field to create your error bars.

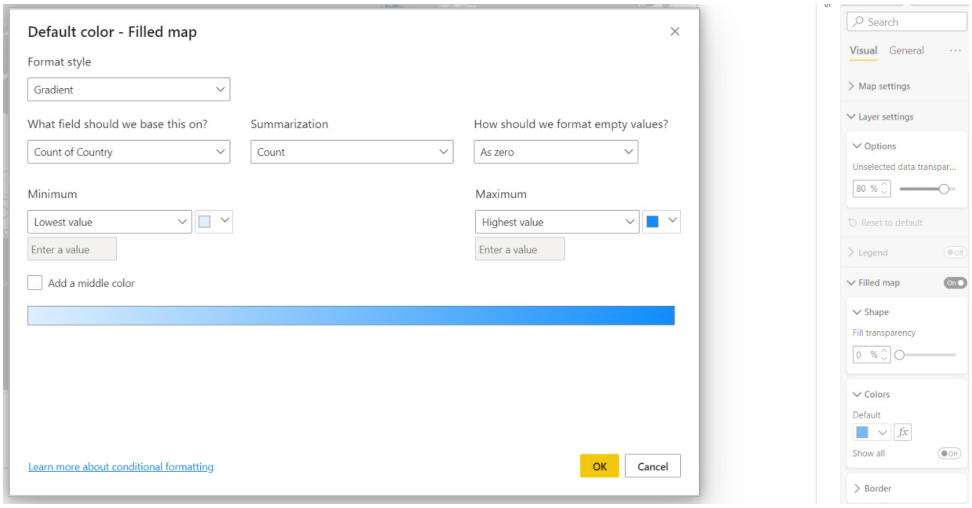
Filled Map for Azure Maps visual

The latest update introduces filled map layers to the Azure Maps visual. Filled maps use coloured-in shapes to differentiate data, helping to present variations or patterns across different geographical regions.



To begin using this feature, make sure that the Azure maps Preview feature is enabled in the Options menu. Then, make sure you have a set of locations to visualise. These may either be location names to drag into the location field well or latitude and longitudes to drag into their corresponding field wells: just make sure that the locations denote areas with boundaries. This will help the Azure maps visual separate the areas into distinct shapes. Then turn on the 'Filled map' option in the Formatting pane and you'll see those shapes drawn onto your map.

You may colour these shapes in two ways. If you want to differentiate the regions by a categorical field, you can drag that field into the Legend field well and set the colours of each category in the 'Filled map' card in the Formatting pane. You can also colour these shapes by a measure using conditional formatting instead. While the Legend field well is empty, click the **fx** button under 'Filled map colors' in the Formatting pane. Then, set the rules for your measure, press OK and you're ready to go.



Filled maps are an effective visualisation for numerical data being aggregated at the regional level, as well as for categorical data that varies by region.

Composite models on Power BI Datasets and Analysis Services now in Preview

This month sees support for Power BI datasets added that have Dynamic **M** Query Parameters defined. Now, you may create a composite model on such datasets to enrich or extend them. With Dynamic **M** Query Parameters, you can let report viewers use filters or slicers to set values for an **M** query parameter.

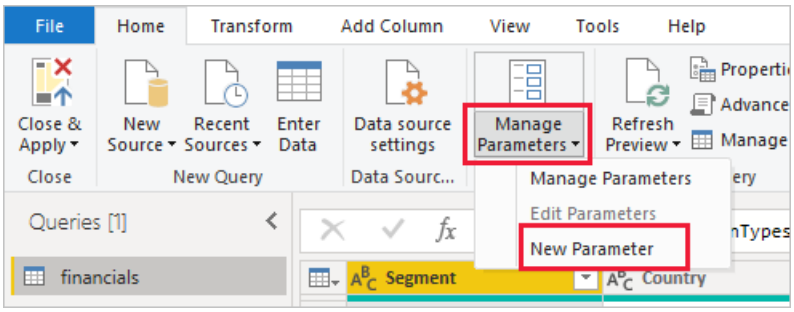
With Dynamic **M** Query Parameters, model developers can let report viewers use filters or slicers to set the value(s) for an **M** Query Parameter, which can be especially useful for query performance optimisations. With Dynamic **M** Query Parameters, model authors have more control over how filter selections get incorporated into DirectQuery source queries.

When builders understand the intended semantics of their filters, they often know how to write efficient queries against their data source, and can thus ensure filter selections get incorporated into source queries at the right point to achieve their intended results with improved performance.

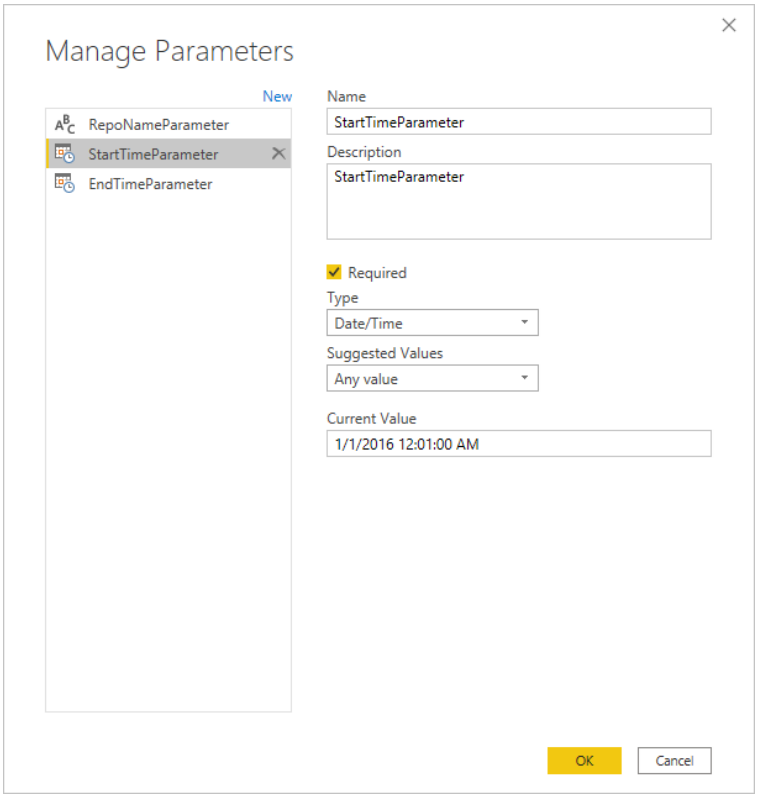
As a prerequisite for this feature, you must have a valid **M** Query Parameter created and referred in one or more Direct Query tables.

Example

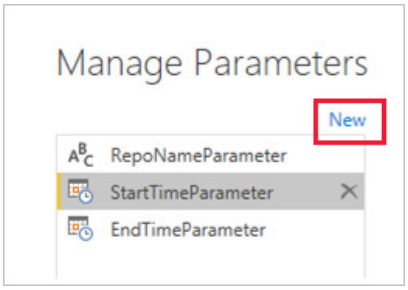
In Power BI Desktop, select **Home -> Transform data- > Transform data** to open the Power Query Editor. Then, select 'New Parameters' under the 'Manage Parameters' button in the Ribbon.



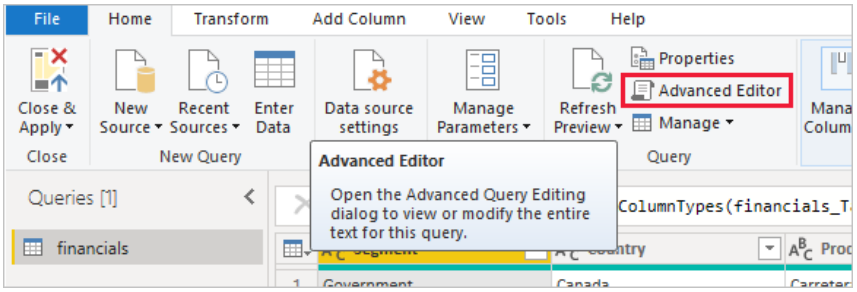
We'll then fill out the following information about the parameter:



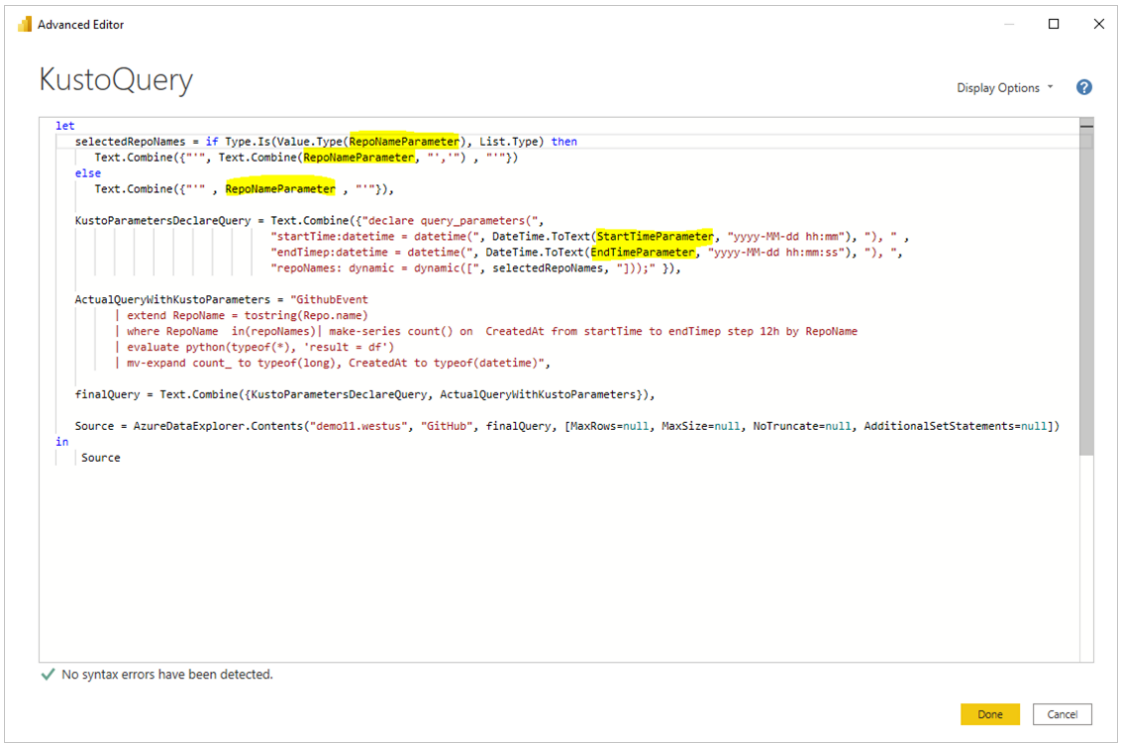
If you have more parameters to add, simply click 'New':



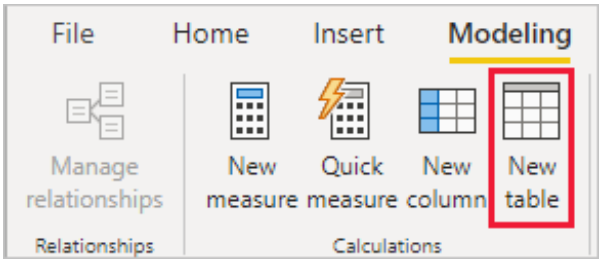
Once you've created your parameters, you can reference them in the **M** query. To modify the **M** query, open the Advanced Editor having selected the query that you want to modify:



Then, reference the parameters in the **M** query, highlighted in yellow *(below)*:

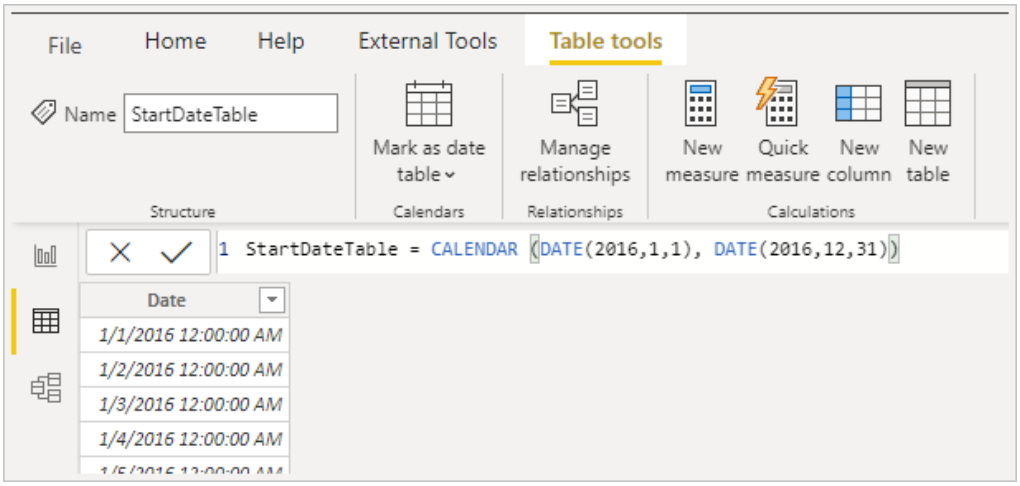


Now that you have created the parameters and referenced them in the **M** query, you will then need to create a table with a column that provides the possible values available for that parameter. This will allow it such that the parameters are dynamically set based on filter selection. In this example, we want our **StartTime** parameter and **EndTime** parameter to be dynamic. Since these parameters are requiring a Date/Time parameter, we want to generate date inputs that may be used to set the date for the parameter. To commence, we create a new table:



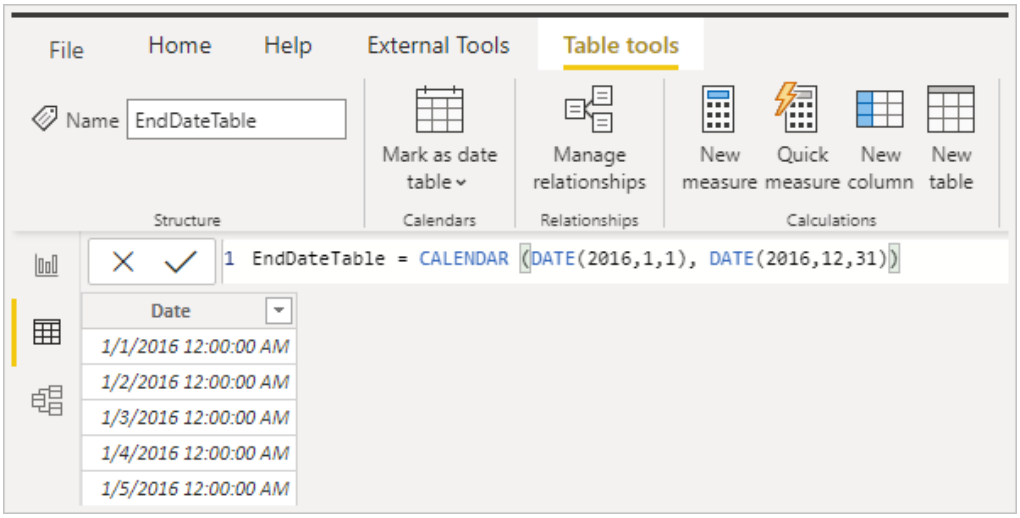
As an example, here's the first table created for the values for **StartTime** parameter:

StartDateTable = CALENDAR (DATE(2016,1,1), DATE(2016,12,31))



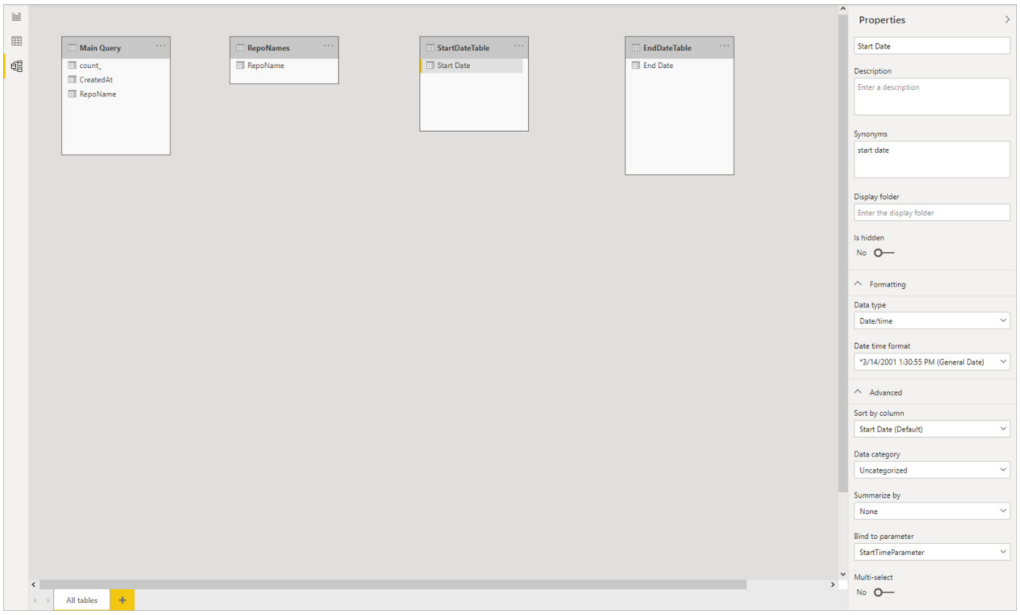
Following on, here's the second table created for the values for **EndTime** parameter:

EndDateTable = CALENDAR (DATE(2016,1,1), DATE(2016,12,31))

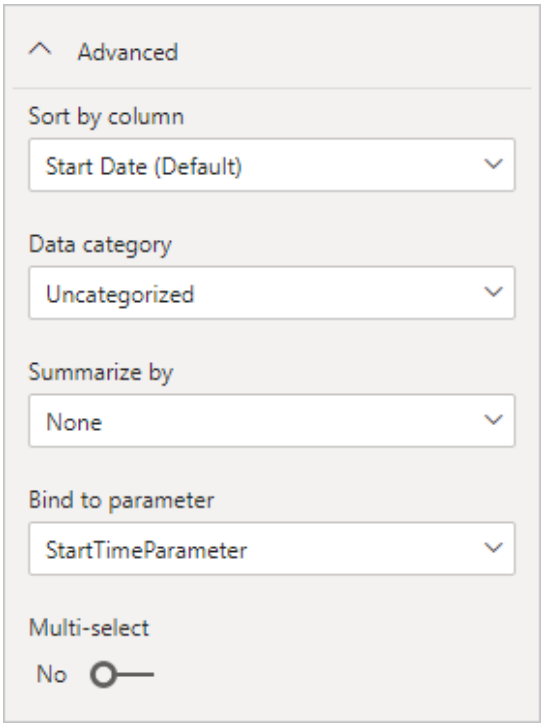


It's recommended that you use a different column name that is not in an actual table. Otherwise, if names are duplicated, the selected value will be applied as a filter to the actual query.

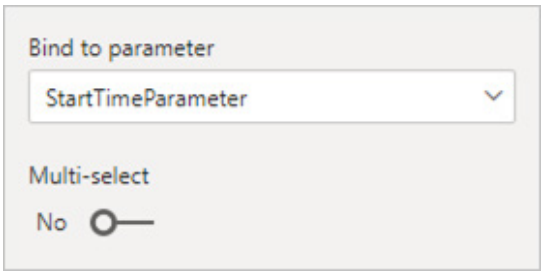
Now that the tables with the Date field have been created, we can bind each field to a parameter. Binding the field to a parameter essentially means that as the selected value for the field changes, the value will get passed to the parameter and update the query where the parameter is referenced. Therefore, to bind field, go to the Modeling *(sic)* tab, select the newly created field, and then go to the Advanced properties (noting that the column Data Type should match with the **M** parameter type):



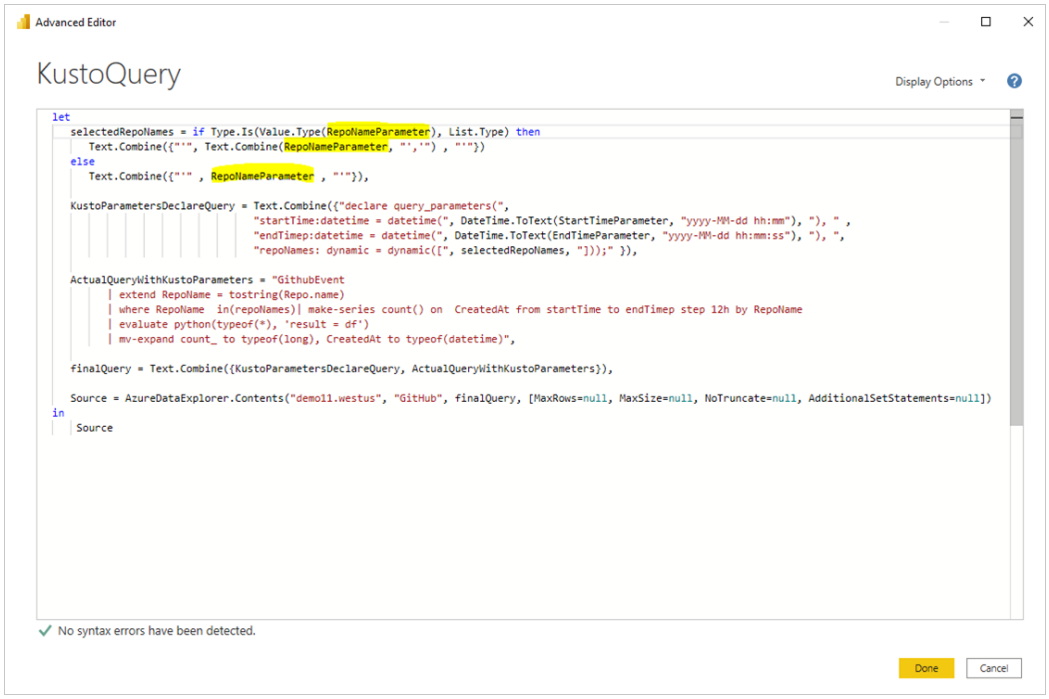
Select the dropdown under ‘Bind to parameter’ and select the parameter that you want to bind to the field:



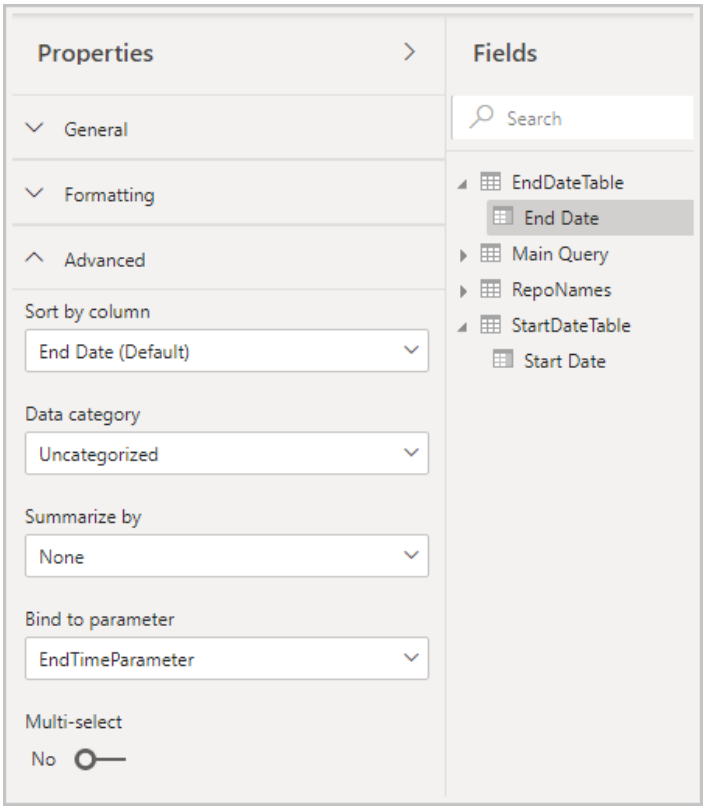
Since this example is for a single-select value (setting the parameter to a single value), you'll want to keep Multi-select set to No, which is the default:



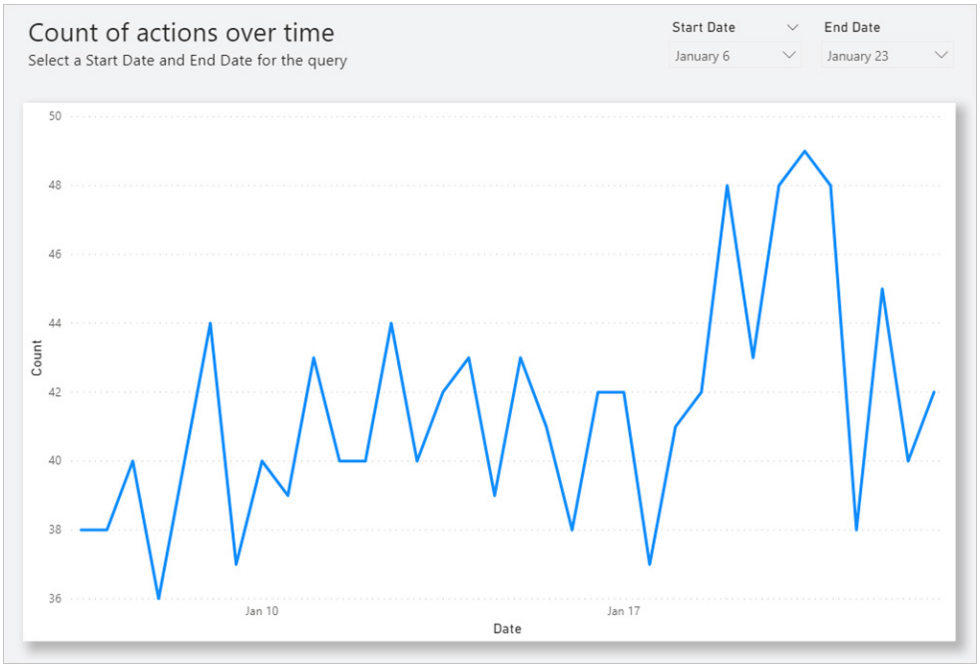
If your use cases require multi-selection (passing multi-values to a single parameter), you must toggle the switch to Yes and ensure that your **M** query is set up properly to accept multiple values in the **M** query. Here's an example for **RepoNameParameter**, which allows for multiple values:



You may repeat these steps if you have other fields to bind to other parameters:

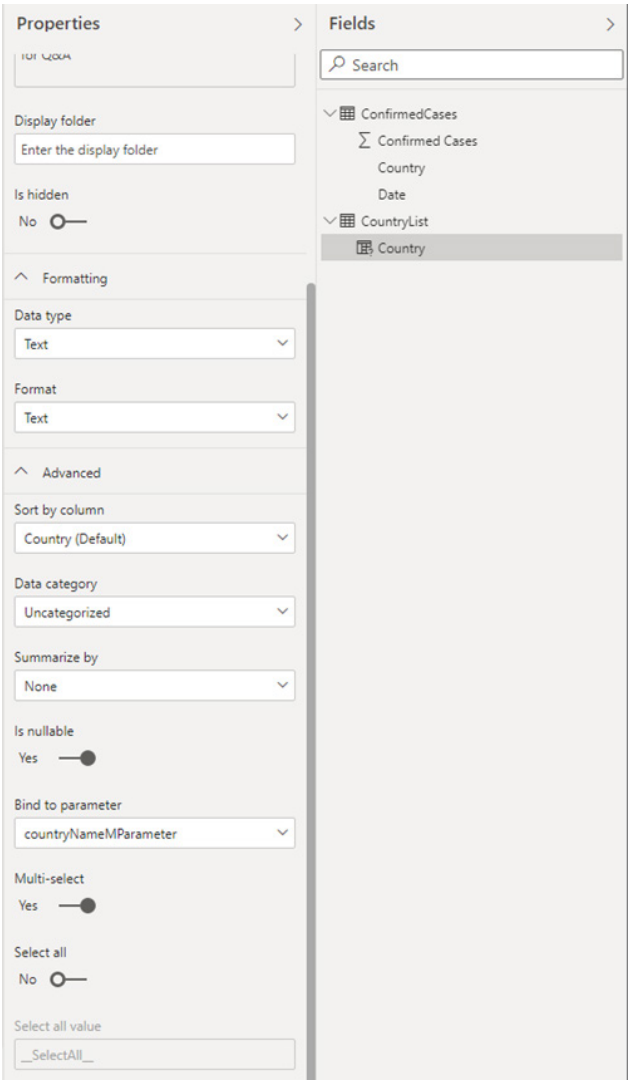


Finally, you can reference this field in a slicer or as a filter:

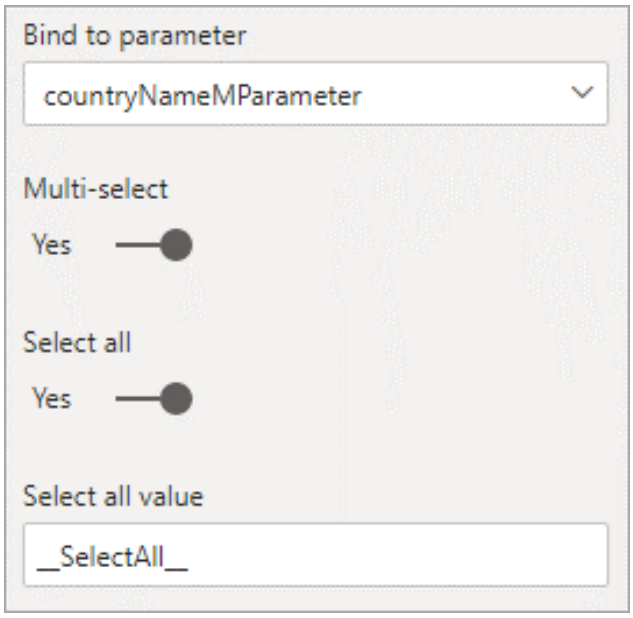


If the mapped column is set to No for Multi-select, you must use either a single select mode in the slicer or require single select in the Filter card.

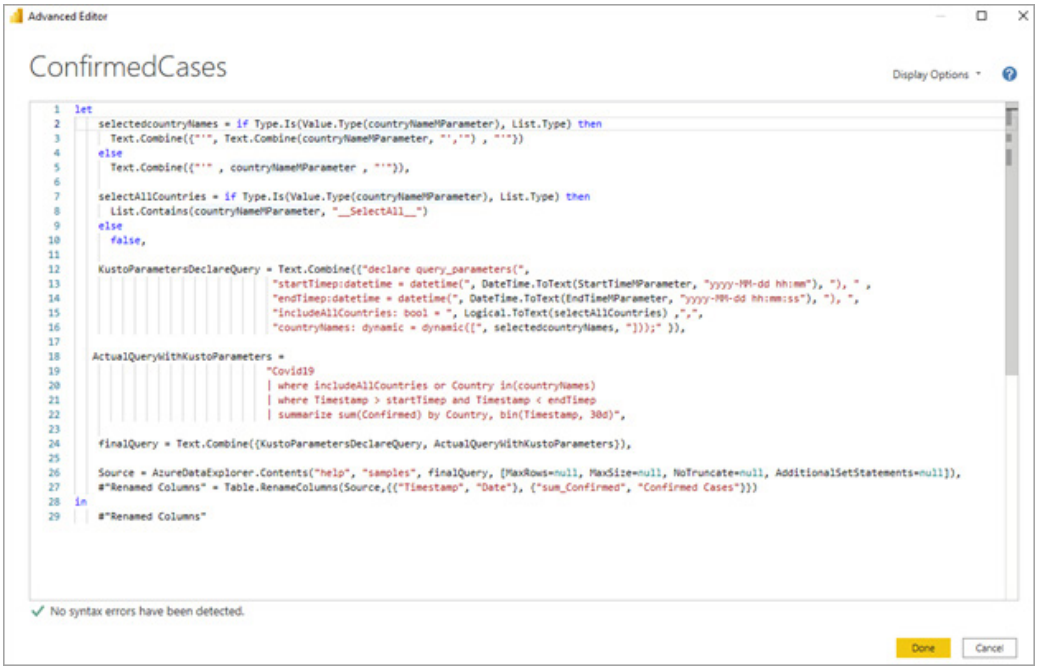
There are additional steps if you want end users to be able to use the 'Select all' option in the Slicer or Filter card. Consider the following scenario. Within the Model tab of Power BI Desktop, let's have a field called **Country** (list of countries) that is bound to an **M** parameter called **countryNameMParameter**:



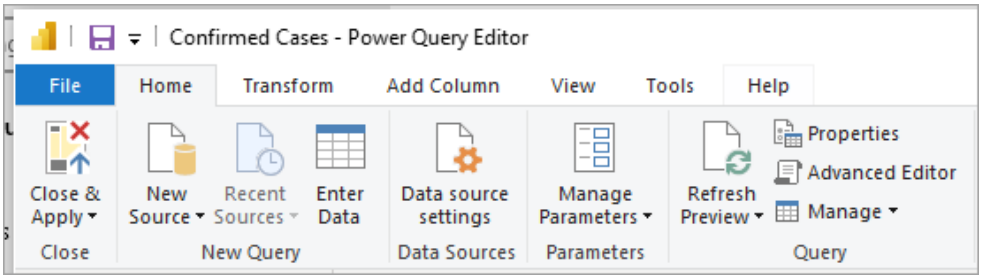
You'll also notice that this parameter is enabled for Multi-select but not enabled for 'Select all'. When we enable the 'Select all' toggle, we'll see an enabled input called 'Select all' value:



The 'Select all' value is used to refer to the 'Select all' option in the **M** query. This value is passed to parameter as a list that contains the value you defined for 'Select all'. Therefore, when you are defining this value or using the default value, you will need to make sure that this value is unique and does not exist in the field bound to the parameter. Once you have set the value or used the default value for 'Select all', you will then need to update the **M** query to account for this 'Select all' value.



To edit the **M** query, you will need first launch the Power Query Editor and then select 'Advanced Editor' in the Query section:



In the Advanced Editor, we need to add a Boolean expression that will evaluate to TRUE if the parameter is enabled for Multi-select and contains the 'Select all' value (else return FALSE). For our example that would look like this:

```
KustoParametersDeclareQuery = Text.Combine({"declare query_parameters(",
                        "startTimep:datetime = datetime(", DateTime.ToText(StartTimeMParameter, "yyyy-MM-dd hh:mm"), "), " ,
                        "endTimep:datetime = datetime(", DateTime.ToText(EndTimeMParameter, "yyyy-MM-dd hh:mm:ss"), "), ",
                        "includeAllCountries: bool = ", Logical.ToText(selectAllCountries) ,",",
                        "countryNames: dynamic = dynamic([", selectedcountryNames, "]));" }),

ActualQueryWithKustoParameters =
                        "Covid19
                        | where includeAllCountries or Country in(countryNames)
                        | where Timestamp > startTimep and Timestamp < endTimep
                        | summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",

finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),
```

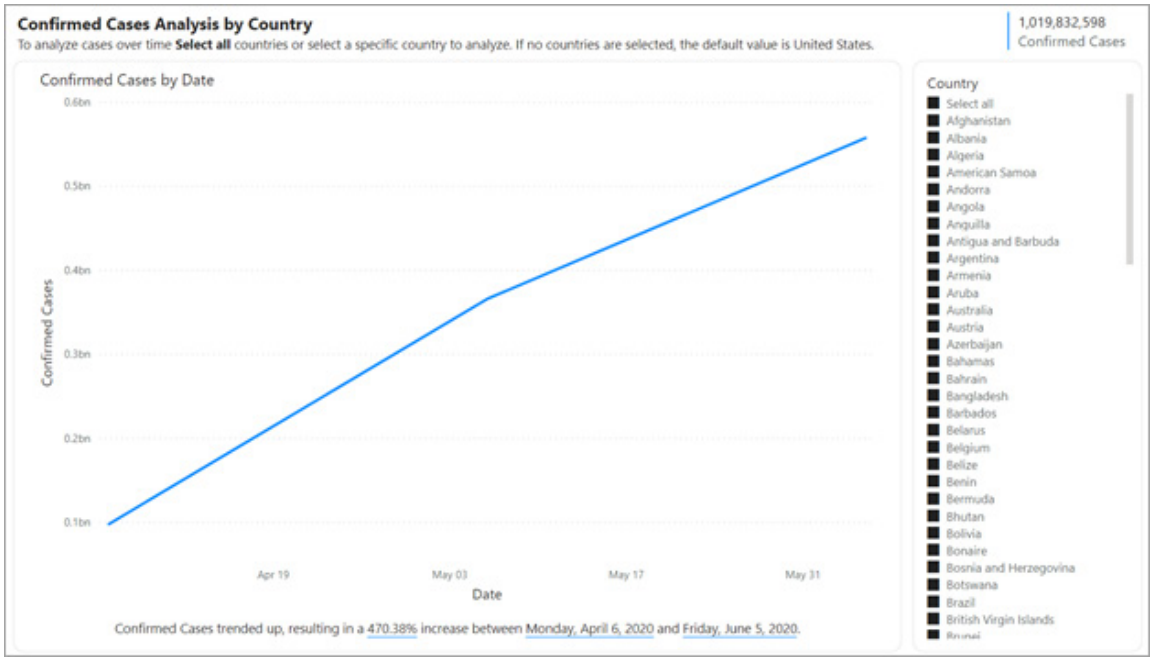
Next, we will need to incorporate the result of this 'Select all' Boolean expression into the source query. For our example, we have a Boolean query parameter in the source query called **includeAllCountries** that is set to the result of the Boolean expression from the previous step. We then use this parameter in a filter clause in the query, such that FALSE for the Boolean will filter to the selected country name(s) and a TRUE would effectively apply no filter:

```
selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
  List.Contains(countryNameMParameter, "__SelectAll__")
```

For reference here is the full query employed:

```
let
    selectedcountryNames = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
      Text.Combine({"'", Text.Combine(countryNameMParameter, "','") , "'"})
    else
      Text.Combine({"'" , countryNameMParameter , "'"}),

    selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
      List.Contains(countryNameMParameter, "__SelectAll__")
    else
      false,

    KustoParametersDeclareQuery = Text.Combine({"declare query_parameters(",
                              "startTimep:datetime = datetime(", DateTime.ToText(StartTimeMParameter,
"yyyy-MM-dd hh:mm"), "), " ,
                                "endTimep:datetime = datetime(", DateTime.ToText(EndTimeMParameter,
"yyyy-MM-dd hh:mm:ss"), "), ",
                                 "includeAllCountries: bool = ", Logical.ToText(selectAllCountries)
,",",
                               "countryNames: dynamic = dynamic([", selectedcountryNames, "]));" }),

   ActualQueryWithKustoParameters =
                               "Covid19
                               | where includeAllCountries or Country in(countryNames)
                               | where Timestamp > startTimep and Timestamp < endTimep
                               | summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",

    finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),

     Source = AzureDataExplorer.Contents("help", "samples", finalQuery, [MaxRows=null, MaxSize=null,
NoTruncate=null, AdditionalSetStatements=null]),
   #"Renamed Columns" = Table.RenameColumns(Source,{{"Timestamp", "Date"}, {"sum_Confirmed", "Confirmed
Cases"}})
in
    #"Renamed Columns"
```

Once you have updated your **M** query to account for this new 'Select all' value, you can now use the 'Select all' function in slicers or filters:



When you allow report readers to dynamically set the values for the **M** query parameters, they may be able to access additional data or trigger modifications to the source system using injection attacks, depending upon how the parameters are referenced in the **M** query and what values are passed to that parameter.

For example, let's say you have a parameterised query constructed similar to the following:

```
Products
| where Category == [Parameter inserted here] & HasReleased == 'True'
 | project ReleaseDate, Name, Category, Region```
```

You may have no issues with a friendly user who passes an appropriate value for the parameter, for example, *Games:*

```
| where Category == 'Games' & HasReleased == 'True'
```

However, an attacker may be able to pass a value that modifies the query to get access to more data, for example, *'Games'//:*

```
Products
| where Category == 'Games'// & HasReleased == 'True'
| project ReleaseDate, Name, Category, Region
```

In this example, the attacker can get access to information on games that have not been released yet by changing part of the query into a comment.

To mitigate the security risk, it's best to avoid string concatenation of **M** parameter values within the query. Instead, consume those parameter values in **M** operations that fold to the source query, so that the **M** engine and connector construct the final query. Alternatively, if available, make use of a parameter passing mechanism built-in to the source query language and connectors. For example, Azure Data Explorer has built-in query parameter capabilities that are designed to protect against injection attacks.

As examples:

- Example using **M** query's filtering operations:

  ```
  Table.SelectRows(Source, (r) =\&gt; r[Columns] = Parameter)
  ```

- Example declaring the parameter in the source query (or passing the parameter value as an input to a source query function):

  ```
  declare query\_parameters (Name of Parameter : Type of Parameter);
  ```

There are some considerations and limitations to consider when using dynamic **M** query parameters:

- a single parameter cannot be bound to multiple fields nor vice-versa
- aggregations are not supported with the feature
- Row Level Security (RLS) is not supported with the feature
- parameter names cannot be reserved words in DAX nor contain spaces. Appending 'Parameter to the end of the parameter name can help avoid this limitation

- if your parameter is of Date/Time data type, you will need to cast it within the **M** query as **DateTime.Date(YourDateParameter)**
- if using SQL sources, you may notice a confirmation dialog every time the parameter value changes. This is due to a security setting: 'Require user approval for new native database queries'. You can find and turn off this setting within the Security tab of the Options dialog in Power BI Desktop
- unsupported out-of-box parameter types are the following:
  - Any
  - Duration
  - True/False
  - Binary
- unsupported filters:
  - Relative time slicer or filter
  - Relative date
  - Hierarchy slicer
  - Multi-field include filter
  - Exclude filter / Not filters
  - Cross-highlighting
  - Drill-down filter
  - Cross drill filter
  - Top **N** filter
- Unsupported operations:
  - And
  - Contains
  - Less than
  - Greater than
  - Starts with
  - Does not start with
  - Is not
  - Does not contain
  - Is blank
  - Is not blank.

Bold / italics / underline for text inside the header ToolTip

Since introducing the bold / italics / underline formatting options for text, Microsoft has been making sure these options are truly supported everywhere. This update adds these formatting options to the header ToolTip text as well.

Metric visuals updates

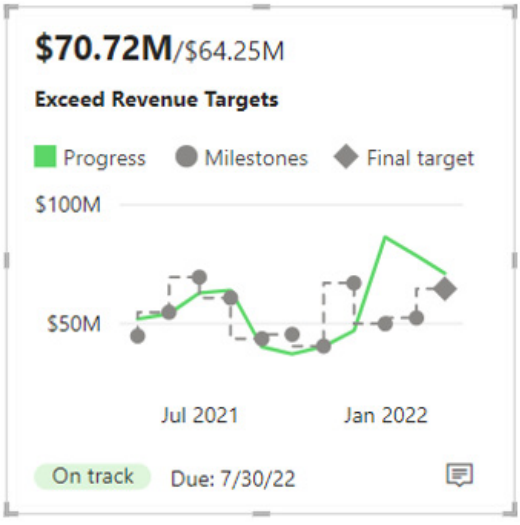
There are several updates to share this month to streamline creation and sharing of metrics. These are:

- Metric visuals
- Move and copy scorecard
- Follow metrics
- Share link to a metric.

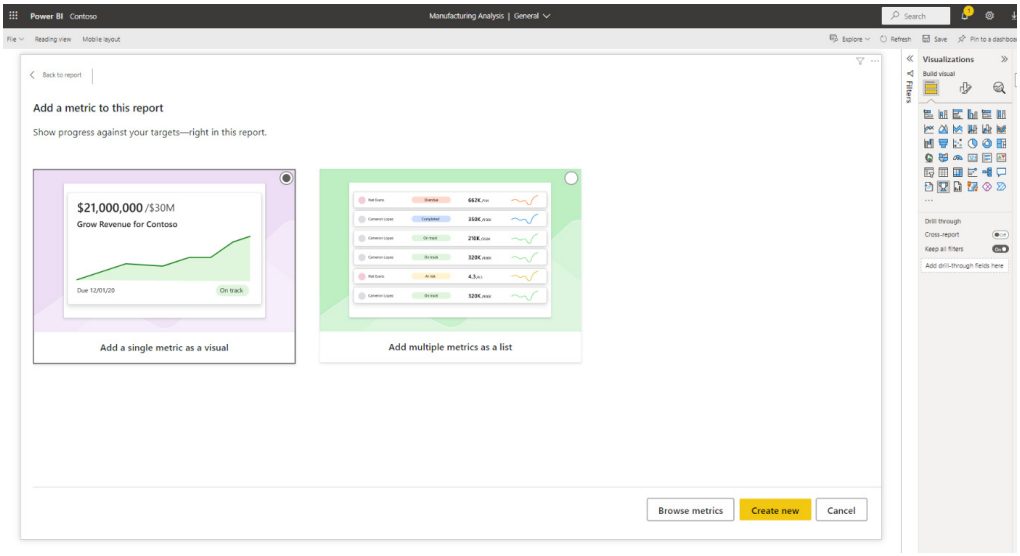
Let's plough through them.

METRIC VISUALS

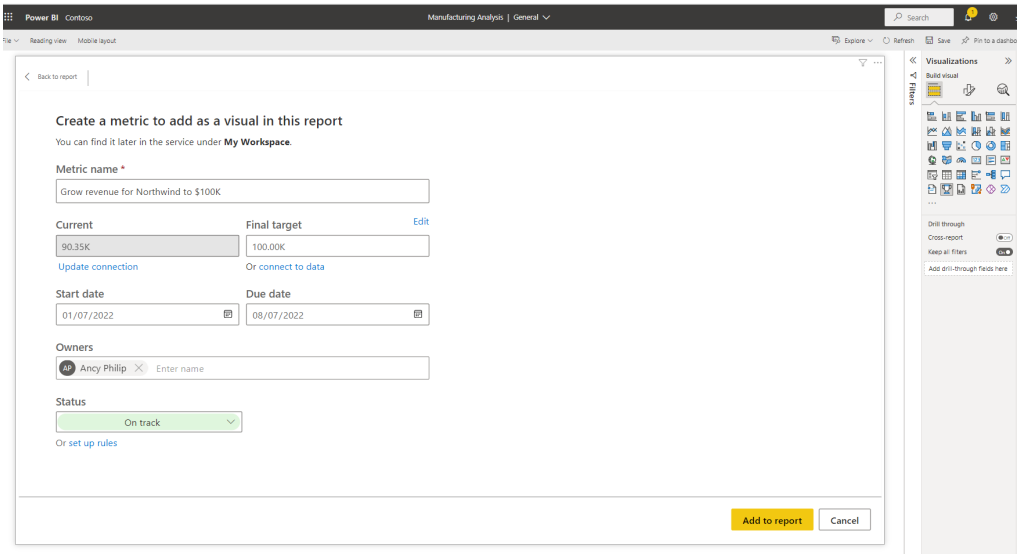
The Scorecard visual shipped last year has now been updated to include metric visuals. This provides a level of flexibility in integrating metrics into your Power BI reporting solutions. You can now include individual metrics instead of the entire scorecard and create report pages showcasing metrics alongside other visuals, in the context of the rest of the report data.



You may either create a new metric or add a metric from an existing scorecard as a visual in the report.



You can create a metric by either entering values or connecting to data in reports you have access to.



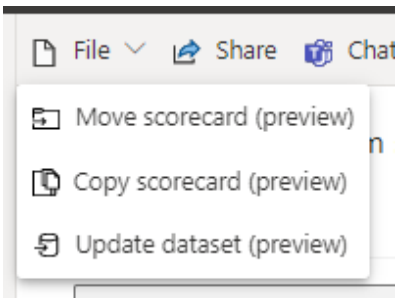
You may view notes, perform check-ins, set up rules on this metric, just like on a regular scorecard.

You can also format the metric visual to the look and feel of the rest of the report and it is configurable with options to turn the individual metric elements (such as targets, owner *etc.*) on / off.

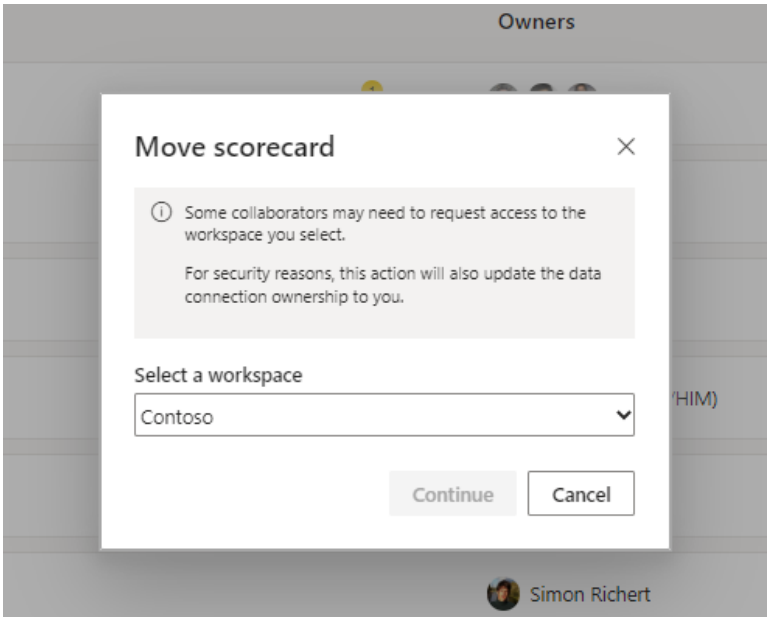
It should be noted that his will be available in Power BI Service by the time you read this and will be made available in Power BI Desktop later in August.

MOVE / COPY SCORECARD

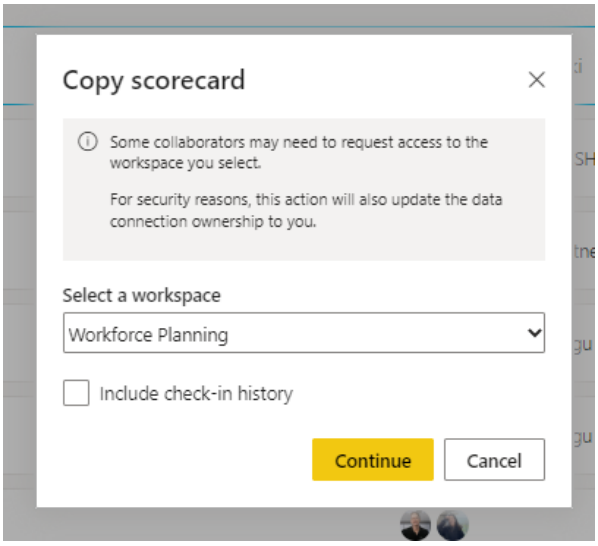
Many organisations have "template" scorecards that different departments or business groups might want to use as a starting point. Alternatively, you might want to develop a "test" scorecard in 'My workspace' to make sure the metrics are working correctly before moving it to the correct workspace when it's ready to share. You can now do all this using the new move or copy scorecard functionality.



You can move the entire scorecard with all the scorecard features and metadata including the check-ins, status rules as-is to a new workspace. You will be provided with a dropdown of a list of workspaces you have access to pick from.



Copy scorecard has the option to include / exclude check-in history. This makes it easy to reuse the same scorecard for a newer time period (for example, for fiscal year scorecards) so you don't have to do this all over again.

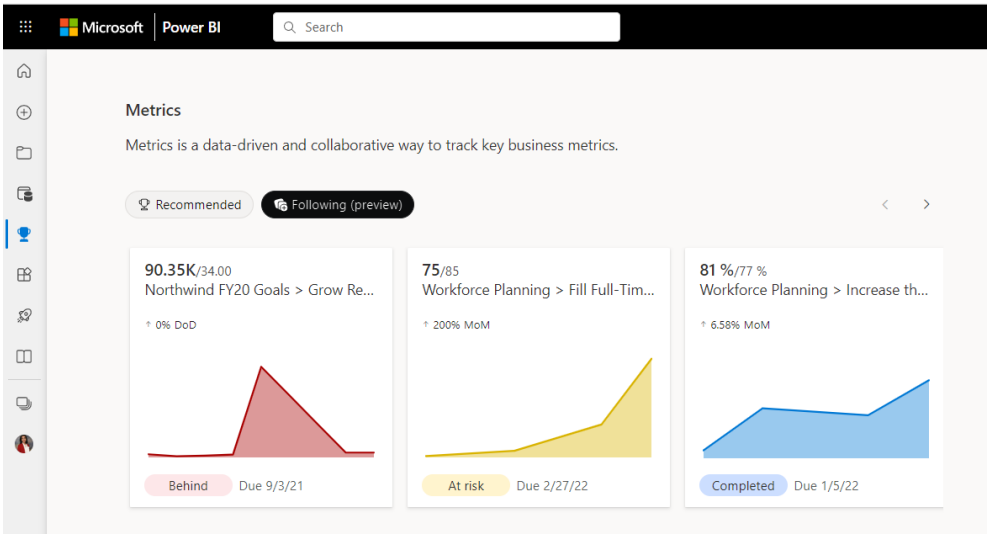


The copy scorecards capability is also useful in getting started quickly if you want to create a new scorecard by reusing the metric definition from an existing scorecard.

FOLLOW METRICS

You can now use the 'Follow' functionality to quickly access metrics you're interested in and stay up to date on the activity on these metrics. You can follow metrics as you're browsing through different scorecards and later access all your followed metrics in one place.

You can view all the metrics you are following in the Metrics hub under the 'Following' section. Clicking on a metric tile takes you to the scorecard containing the metric. This makes it easy to get a quick overview of the metrics you care about without having to navigate to different scorecards often.



Once you follow a metric, you will get Teams notifications right in the activity feed and as a banner (Toast card) whenever the metric definition is updated, a check-in is added or when an automated status rule changes the status of the metric. When you click the notification, the scorecard opens and the Details pane for the metric is shown. This Teams integration makes it easy to stay up to date about the metrics you follow without having to leave Teams to view the scorecards.

SHARE DIRECT LINK TO A METRIC

Often you might want to draw someone else's attention to a specific metric in a scorecard for them to look at it. Now, instead of having to @mention someone in a new check-in, you can reference a metric by simply sharing a direct link to it. When a metric is selected on a scorecard, the URL in the address bar is now updated with the metric ID. The link in the 'Share' scorecard option also has been updated to include the metric ID if a metric is selected. When the recipient opens the link, they now will be able to view the scorecard with the Details pane of that metric open.

New DAX function: NETWORKDAYS

There's a new DAX function added this month: **NETWORKDAYS**. This function returns the number of whole working days between two days. You should note that you may use any way of expressing a date in Power BI to specify the start and end dates, including the **dt"YYYY-MM-DD"** notation. Working days exclude weekends, which are customisable using the optional weekends parameter. By default, this function will use Saturday and Sunday as the weekend days. Any dates provided in an optional holidays parameter will also be excluded when calculating working days.

For example, the following will return a result of 20 working days:

```
WorkingDays := NETWORKDAYS(DATE(2022,10,1), dt"2022-10-30")
```

For reference, 1 October 2022 is a Saturday, and 30 October 2022 is a Sunday.

The following returns 21 working days, because it specifies the weekend to be Friday and Saturday:

```
WorkingDaysFriSat := NETWORKDAYS(DATE(2022,10,1), dt"2022-10-30", 7)
```

Finally, the following returns a result of 19 working days, because it specifies two working days in the timespan as holidays:

```
WorkingDaysFriSatHolidays :=
    VAR _holidays = {DATE(2022, 10, 3), DATE(2022, 10, 4)}
    RETURN NETWORKDAYS(DATE(2022, 10, 1), dt"2022-10-30", 7, _holidays)
```

Support multi-role Row Level Security in composite model

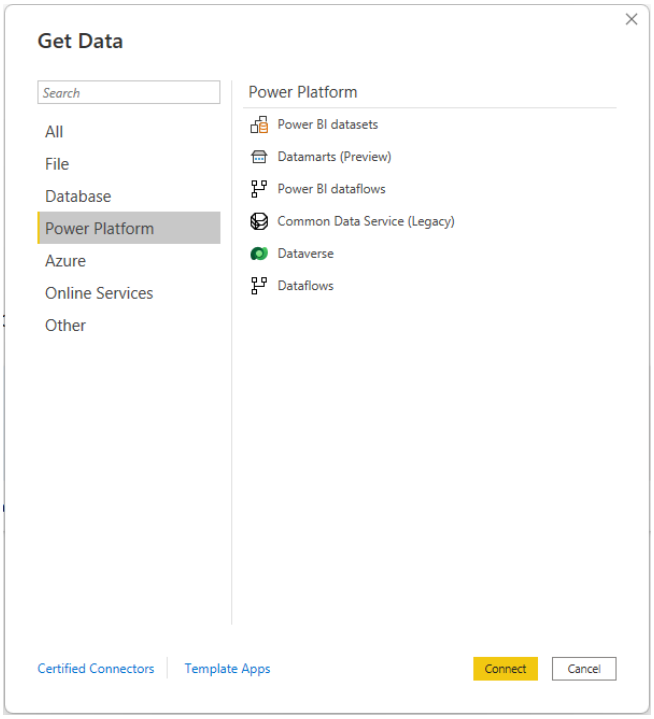
Power BI has also made a major improvement to composite models this month by enabling support for multi-role row level security (RLS). Model owners can now assign a single user to more than one RLS role in a composite model. Prior to this release, users who set up RLS in their composite models this way would likely hit a query failure.

Query performance improvement

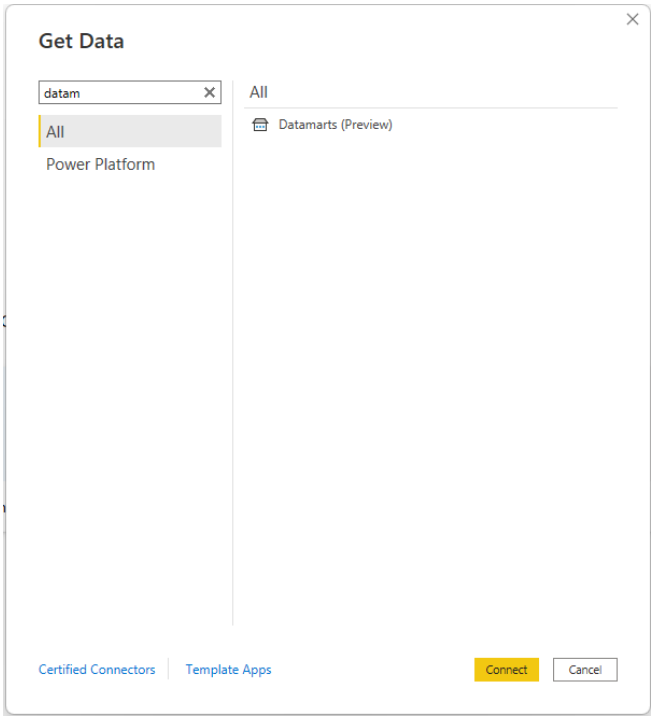
Microsoft has improved query performance to propagate RLS filters defined on import dimension tables to DirectQuery tables using Mashup connectors. Mashup connectors include Snowflake, RedShift, Databricks and others. Previously, when model authors defined RLS filters on an import dimension table and the dimension table filtered a DirectQuery fact table using a Mashup connector, the fact table SQL query didn't include a filter on the foreign key column restricted by the RLS filter when users added a column from the dimension table and a measure from the fact table to a report. This produced a large intermediate result set that could exceed the million-row threshold. This latest improvement ensures that RLS filters are propagated from the dimension table to the fact table in the basic visual query so that users don't hit that error or suffer from slow query performance.

Connect to Datamarts (Preview)

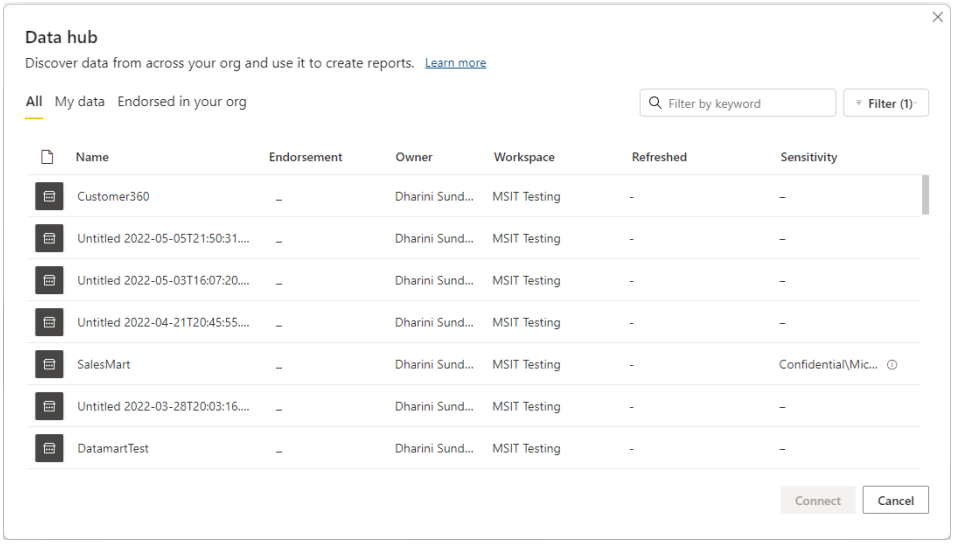
This month, you will notice that you can now better discover the entry point to connect to datamarts from the 'Get Data' experience (Power Platform section) within Power BI Desktop.



You can also search for datamarts within the 'Get data' experience.

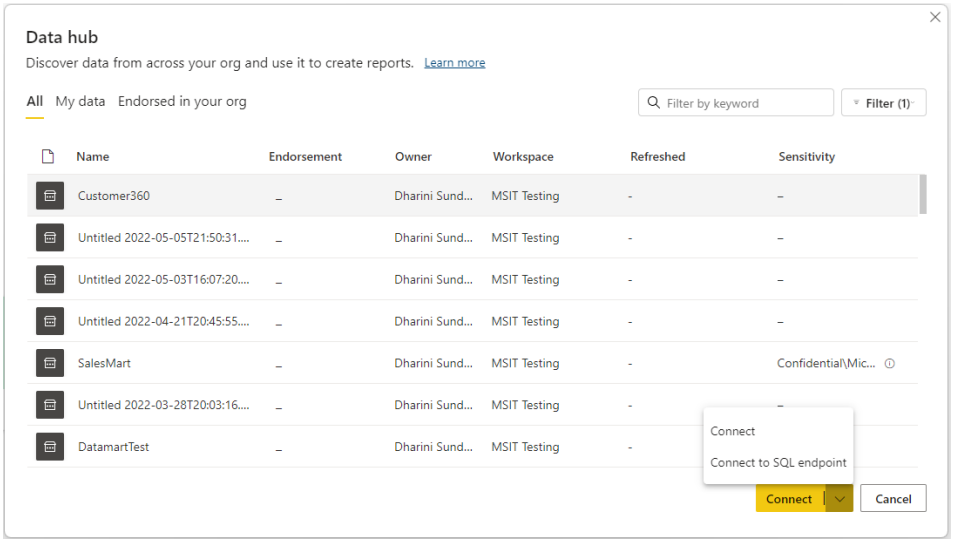


You will be able to see all datamarts that you have access to within the ‘Data hub’ experience.

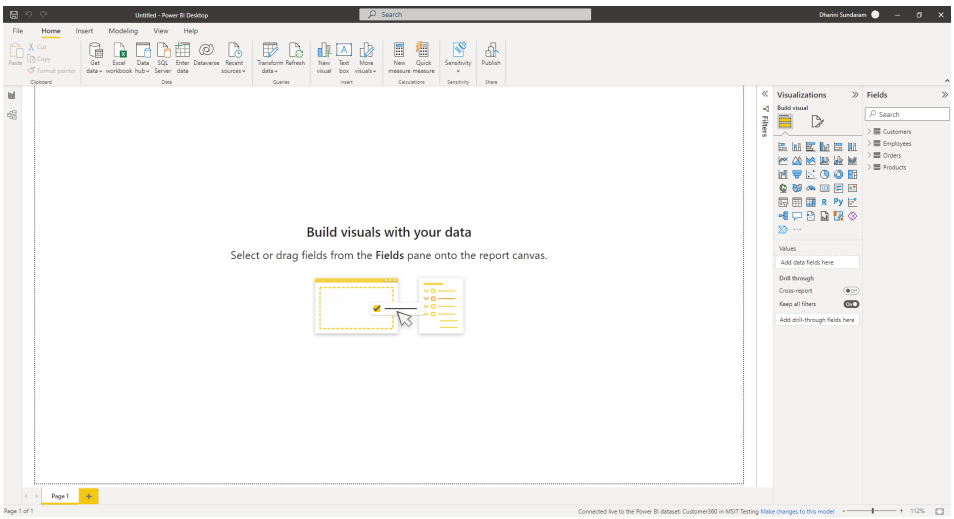


With this new update, clicking on a datamart provides you with two options:

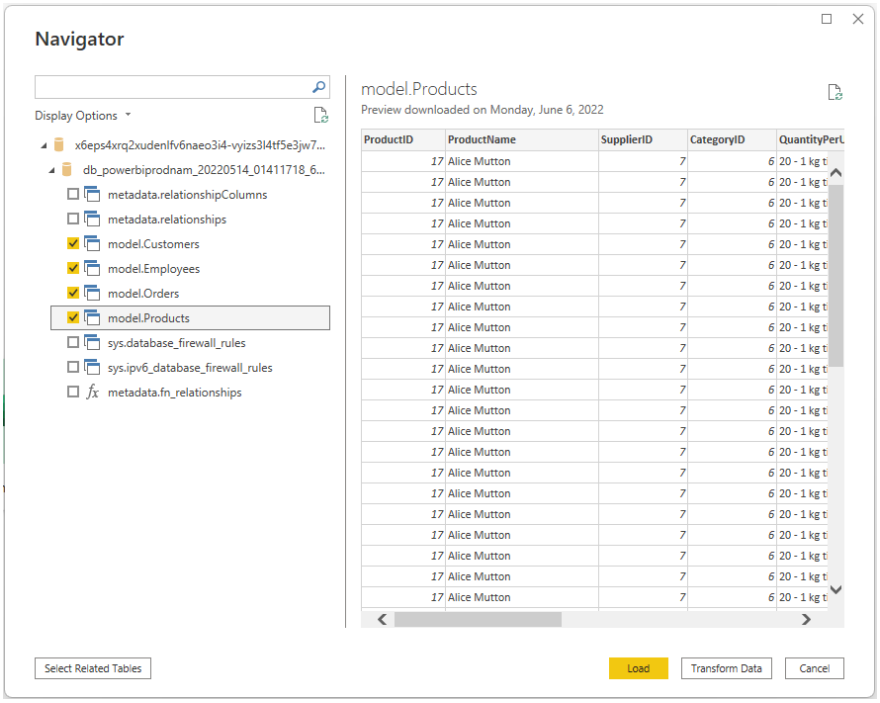
1. **Connect (Auto-generated dataset):** connect to the datamart’s underlying auto-generated dataset using live connect so that you can easily create reports
2. **Connect to SQL endpoint:** connect to the datamart’s SQL endpoint using direct query or import and build datasets or reports. This provides an easy discovery experience and reduces the friction associated with finding and copy-pasting the datamart’s SQL connection string from datamart settings or information page.



Clicking ‘Connect’ connects to the datamart’s auto-generated dataset using live connect and start building visuals to create your report.

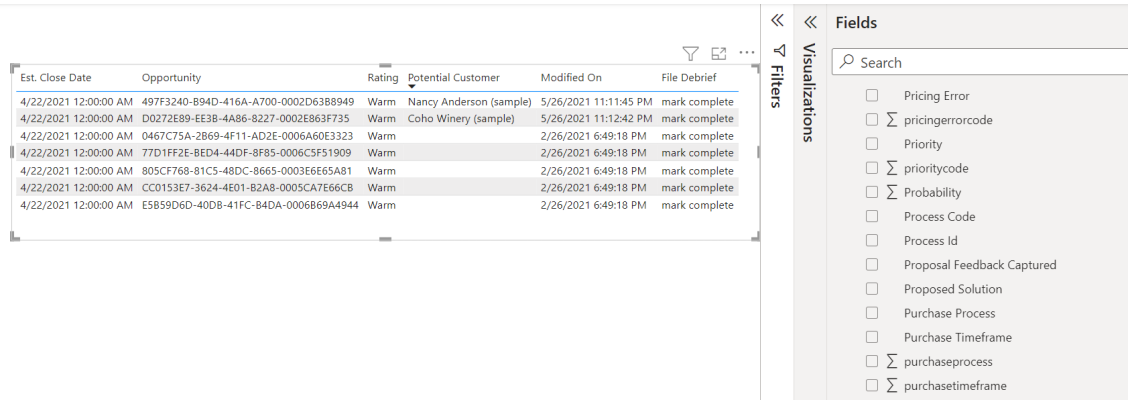


Clicking 'Connect to SQL endpoint' connects to the datamart's SQL endpoint provides the familiar Power Query Navigator experience to select the tables / views and load or transform the data. You can choose to import or direct query the data from the datamart.



Display name support for the Dataverse connector

Power BI now supports display names from Dataverse. Now, when creating reports using the Dataverse connector, in either Import or DQ mode, you'll see the display names automatically applied to all columns in the model. This should save Dataverse users a lot of time, as you won't need to manually name every column to match names end users are used to seeing in their applications, like Dynamics.



For reports saved in the Power BI Service, to make sure things don't change without warning for report consumers, display names will not automatically appear. For all existing reports, you'll need to open the report up in Power BI Desktop and refresh the model for display names to show up.

On the first schema refresh, the display names will override any existing renames you've made in the report, so if you want to retain those unique names, you'll need to manually reapply the names in the model. This will only happen when the display names are first applied, so future schema refreshes should not require any rework.

One important thing to note is that the display names are applied at the model level, so you will continue to see the original system names in the query editor. This also means that if you rename fields with display names in the Query Editor, that change will not be visible in the model (as the display names are applied on top of it). If you want to rename fields with display names, you'll need to make that change in the model.

Other query editor changes will still work as they have before. For example, you can change the data type of a field in the query editor, and that will still flow through to the model, regardless of if the column has a display name or not. If you do make significant changes to a field, such that it is not correlated to the original field in Dataverse anymore, such as by splitting one column into many, those changes will of course flow through the model, but the display name will not be applied (as the original column is no longer there).

Lastly, if display names change in Dataverse in the future, all your reports will continue to work. Refreshes will continue to run, and you won't need to change anything in your report. For the new display names to appear in the report, you'll just need to do a schema refresh in Power BI Desktop and republish.

Overall, this change should reduce the amount of manual work you have to do to make your reports based on Dataverse data easily readable and familiar for your report consumers.

BitSight Security Ratings (Connector Update)

BitSight is a security ratings provider, surfacing rich data on the cybersecurity performance of companies. This updated connector supports data tables from both BitSight for Security Performance Management (SPM) and BitSight for Third Party Risk Management. SPM enables Chief Information Security Officers to measure, monitor, manage and report on their cybersecurity program performance over time, and to facilitate a universal understanding of cyber risk across their organization.

BitSight for Third Party Risk Management (TPRM) enhances the process of measuring and improving an organization's third party cybersecurity performance over time. Through evidence-based monitoring and oversight, TPRM incorporates objective, ongoing information to better assess the vendor landscape and help prioritize when and where to take action. This connector enables organisations to pull the BitSight data into Microsoft Power BI for further analysis and dashboard creation. The dashboarding and analysis generated in Power BI can then be leveraged in executive reporting, tracking remediation progress, combining the BitSight data with other security data sources to gain a more complete view of their cybersecurity program performance over time and help bring universal understanding of cyber risk to stakeholders, as well as other use cases.

Databricks (Connector Update)

The Databricks connector has been updated to support localisation. The system proxy feature has also been made optional as a bug workaround.

Eduframe Reporting (New Connector)

Eduframe provides a fully flexible ecommerce storefront, course and student management, representing resource management for Continuing Education providers, integrating with Canvas LMS, invoicing software and CRM systems. The Power BI connector for Eduframe helps you quickly import Eduframe data to get an instant dashboard that gives you insights into revenue, number of orders, number of students and enrolments (over time), the number of courses and planned courses, opportunities (leads), *etc.*

Further, it enables business schools and continuing education departments at higher education institutions to attract and retain learners, automate manual work and provide the most engaging experiences for both learners and staff, all in one platform.

Drieam, the developer of Eduframe, is also the Canvas learning platform partner for educators worldwide. Eduframe is seamlessly integrated with Canvas.

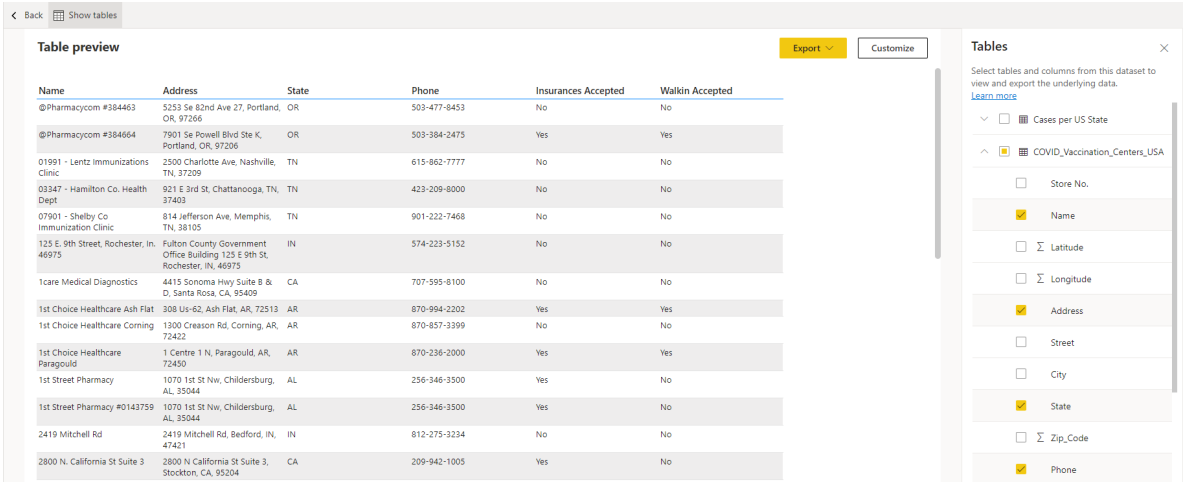
Funnel (Connector Update)

The Funnel connector has been updated. It now fixes a bug where changing the view name in Funnel would break the data source connection.

Data hub improvements: data preview and export

The dataset details page helps you explore, monitor and leverage datasets to gain insights. When you click on a dataset in the data hub or in a workspace, the details page for that dataset opens.

With this release you can see data preview and export data from a dataset in just a couple of clicks. To preview data from a dataset, you can select a table or columns from the Tables view on the right-side pane. Previews may not show all the data you've selected. To see more, you can export or customise the table.



Introducing Data in Space

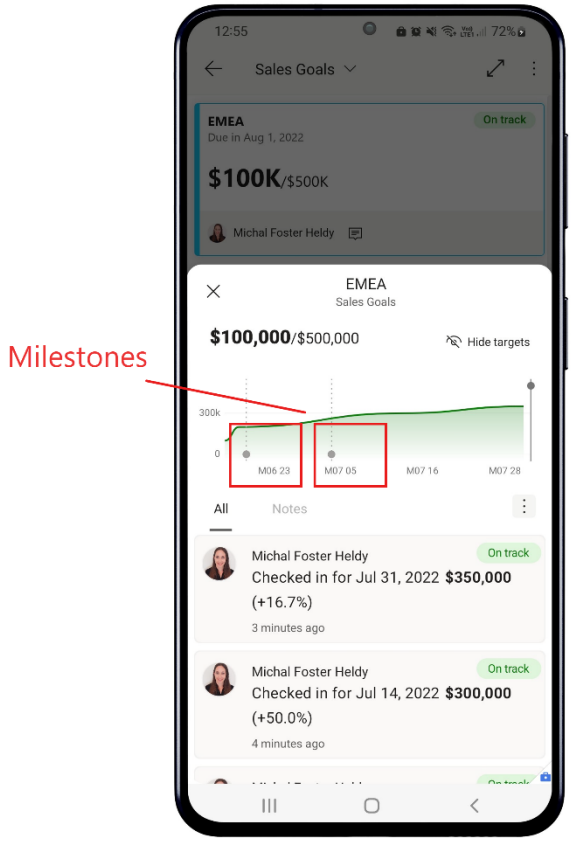
Data in Space is a new feature that uses augmented reality technology in the Power BI Mobile app to create persistent spatial anchors in the real world and attach Power BI content to those anchors. With Data in space, Power BI data can now become contextually integrated with the physical world it describes.

Data in space connects your business data to your real-world scenarios in facility management, manufacturing, retail and many more, and now employees can easily discover and use it for better, more informed decisions.

Sounds very fluffy! Maybe it sends your data into space..?

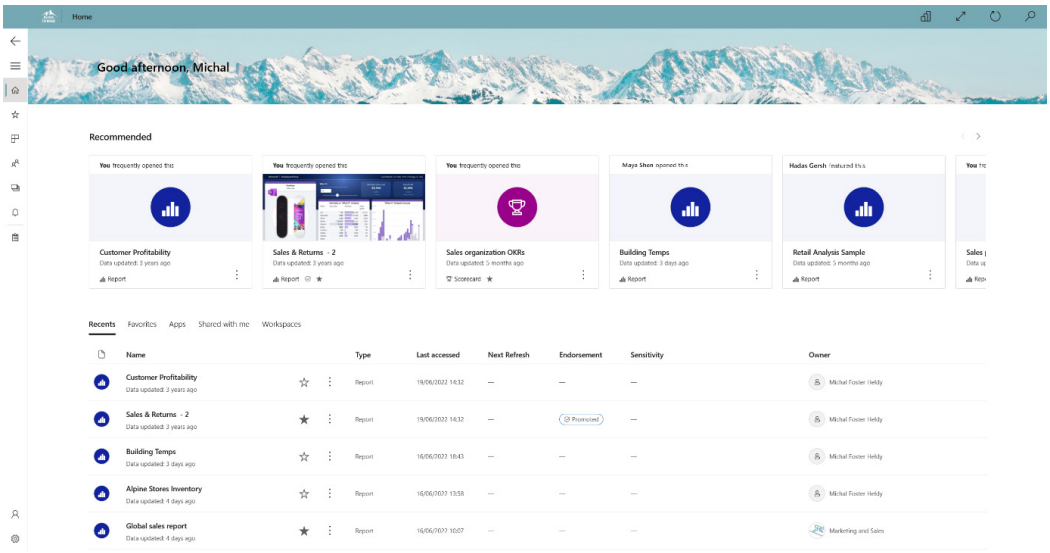
Track metrics with multiple milestones and targets from your mobile app

Now you can track and update metrics that feature multiple targets. Open the metric's Details pane to see the milestones as well as the final target visualised alongside your current progress, so that it's easier than ever to get a snapshot of how you're performing against your metrics.



Find content that's relevant to you (Windows app)

Big Brother is watching you. Now you may "enhance your Power BI Windows app experience and productivity by exploring content from your organisation that has been picked especially for you". It's there for you on the Home page in the new Recommended strip.

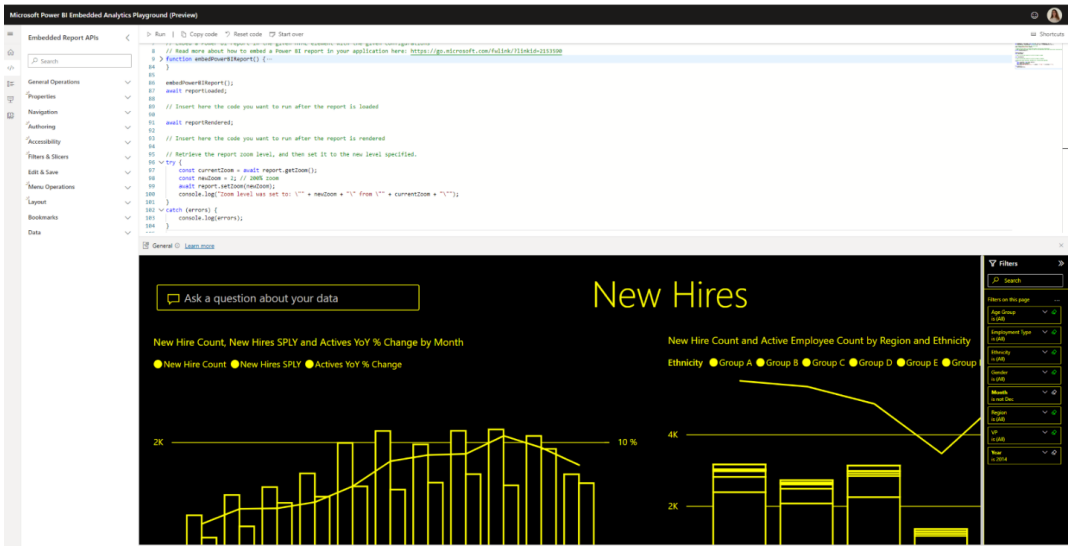


Accessibility enhancements for embedded reports

When embedding Power BI into your applications, it's important to consider the different types of users who will be interacting with your reports and allow users who may have visual or physical impairments to benefit from your reports.

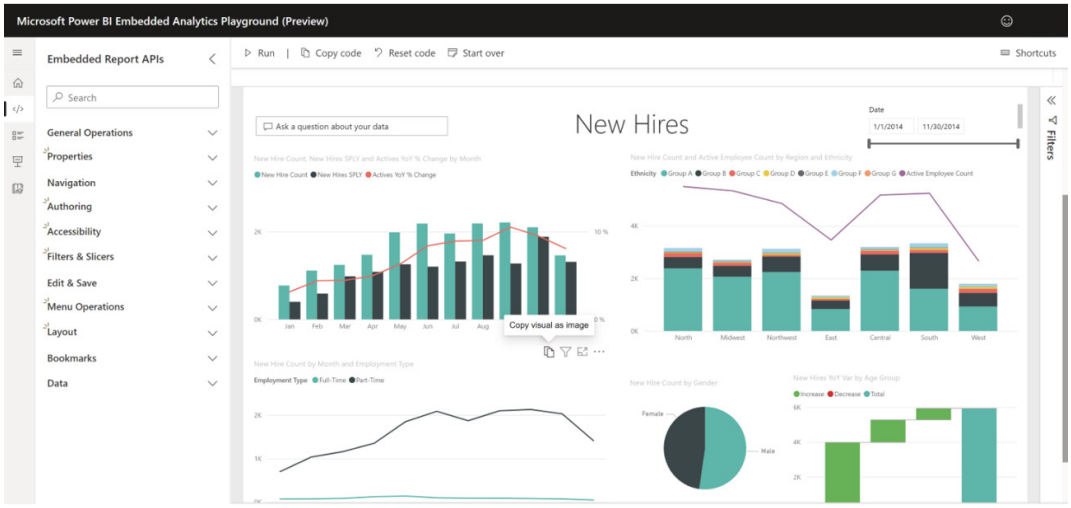
Now, different APIs have been provided that may help make your content more accessible. You can adjust zoom levels and add title attributes to the embedded component with just a few lines of code, and enable keyboard shortcuts and high contrast modes for embedded reports.

Shown below is an example of a report with high contrast mode set in the embedded configurations, and with the zoom level set to 200% after the report has been loaded using the getZoom and setZoom APIs.



Copy visual as an image in embed for your organisational scenarios

When viewing a Power BI report embedded into your organisation's portal, your users may want to copy a specific visual to share it elsewhere. This capability is available in the Power BI Service, and it is now also available for embeddead reports when embedding for your organisation.



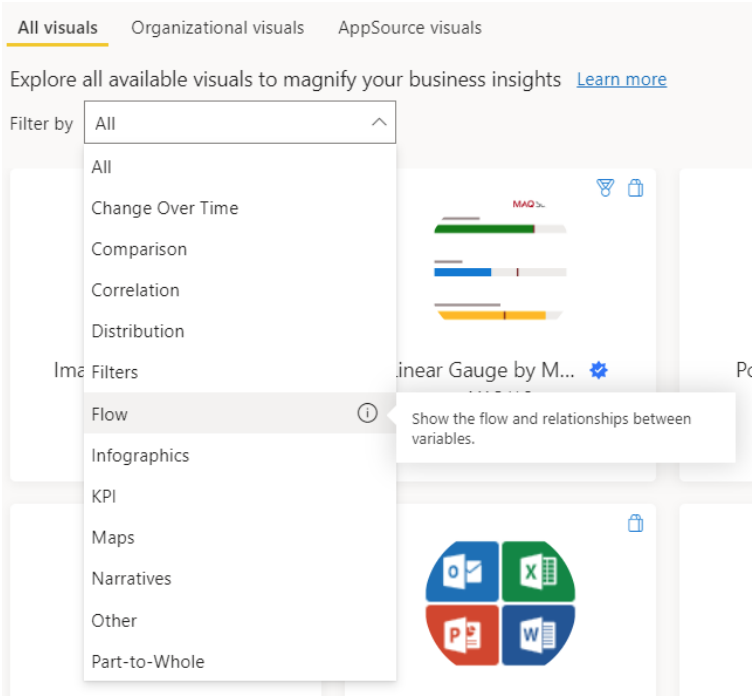
If you're not seeing the copy icon in your embedded reports, keep in mind that this is currently not supported in the embed for your customers scenario.

Drill API

The new API is available with 4.7 API release. It will enable the visual to show next level, expand to the next level, or drill up based on the paraments passed to the API.

New Power BI visuals' category list in AppSource

This update endeavours to make it easier and quicker for you to find the visuals you are looking for by updating the Power BI visuals AppSource categories. The revised category list has more, easier to understand options to help you find a suitable visual quickly and easily. Hovering over the icon will also provide you with brief description of the category.

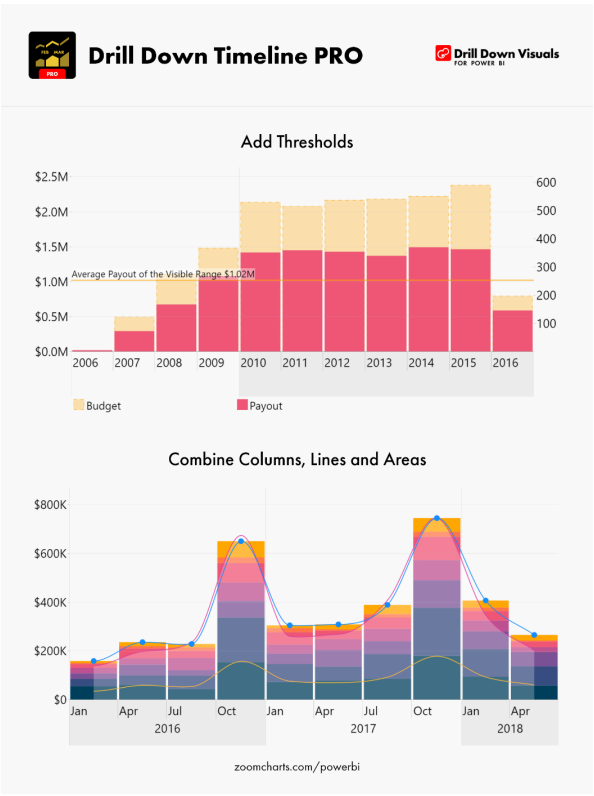


New visuals in AppSource

The Multi target KPI by Aleksei Kolokolov is new this month.

Drill Down Timeline PRO by ZoomCharts

Drill Down Timeline PRO lets you explore time-based data using DAX calculated measures. You may click directly on the chart to drill down to examine specific periods in detail. You may also combine multiple series and choose between multiple chart types (line, column, area).



Main features include:

- on-chart interactions: click on the chart to drill down to months, days or hours
- use navigation arrows to scroll to the next or previous period
- customise up to 25 series
- employ conditional formatting
- use series defaults and series value labels to customise multiple series simultaneously
- ToolTip customisation and ToolTip field support
- value aggregation when selecting multiple units
- add thresholds to show value milestones
- full customisation, *e.g.* styles, colours, gradients, opacities, backgrounds and fonts
- touch device friendly.

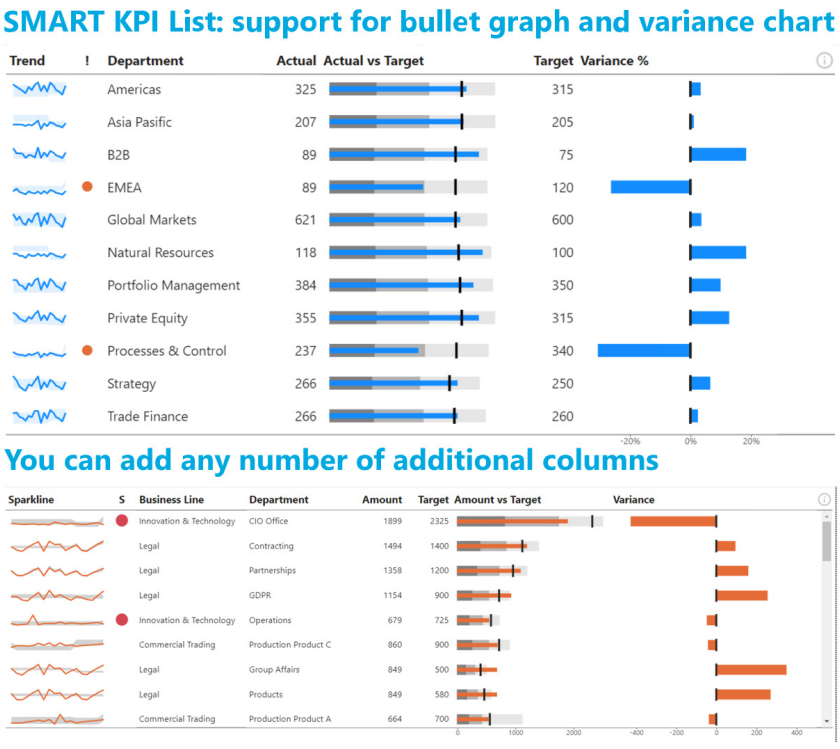
Examples where this may be used include:

- **Banking and Finance:** stock exchange indices, capital ratios and transaction volumes
- **Sales and Marketing:** web traffic, audience reach and sales revenue
- **Information Technologies:** network traffic, response times, systems log and error trends
- **Manufacturing:** quality metrics, uptime and downtime, production output and cost.

Available in AppSource.

SMART KPI List by Nova Silva

This visualisation has been enhanced this month.



Now you can extend the existing bar chart into a bullet graph by adding the qualitative ranges behind the bar. Previously, you had to choose between a bar chart and a variance chart. Now you can show either or both at the same time.

You may also add additional columns to the visual to include more context. Furthermore, all columns may be set to an exact width by dragging the edges.

You can try the SMART KPI List now on your own data by downloading it from the AppSource. All features are available for free to evaluate this visual within Power BI Desktop.

That's it for this month. See you in September!

New Features for Excel

Similar to the Power BI Updates, the Excel gang continues to summarise recent updates in everyone's favourite spreadsheeting software, providing details across Excel for the web, Windows, Mac and mobile.

This month's updates are as follows:

Excel for the web

- Sheet protection
- 'PivotTable Connections' in slicer settings pane
- Semi-select for links creation

Excel for Windows

- Data from Picture (Insider Beta)
- Automatic alternative text suggestions on charts and PivotCharts (Current Channel & Monthly Enterprise Channel)

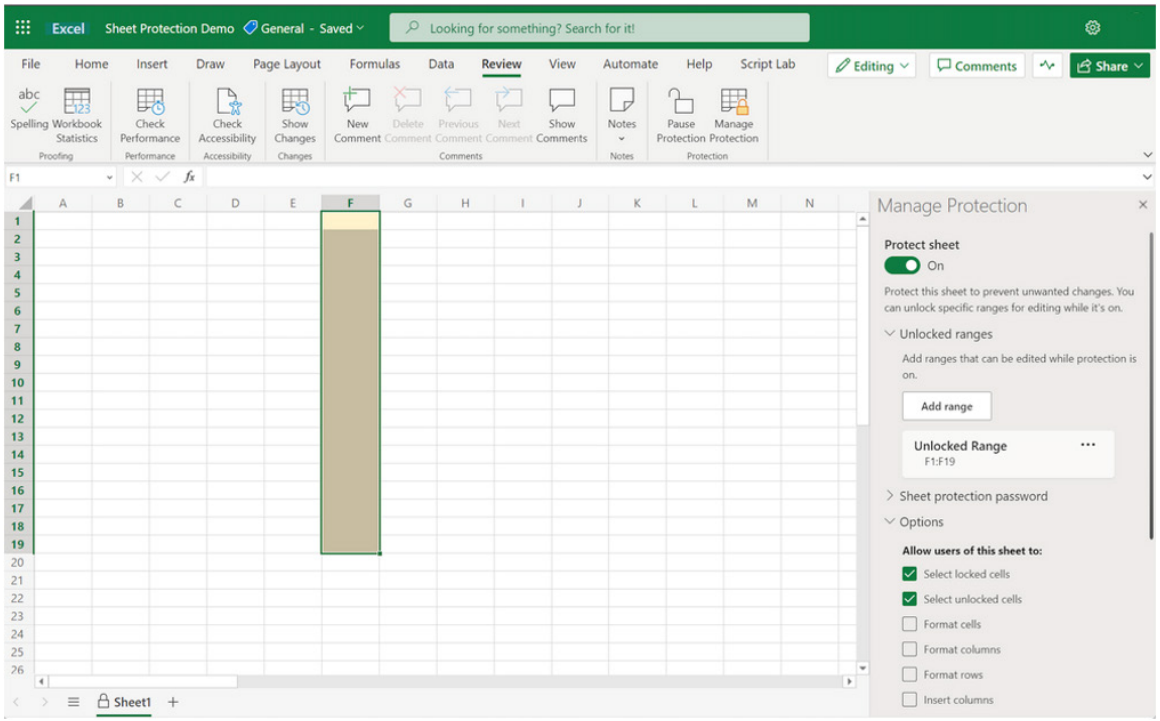
Excel for Mac

- Automatic alternative text suggestions on charts and PivotCharts
- Import from local text, CSV and XLSX files with data preview.

Let's plough through.

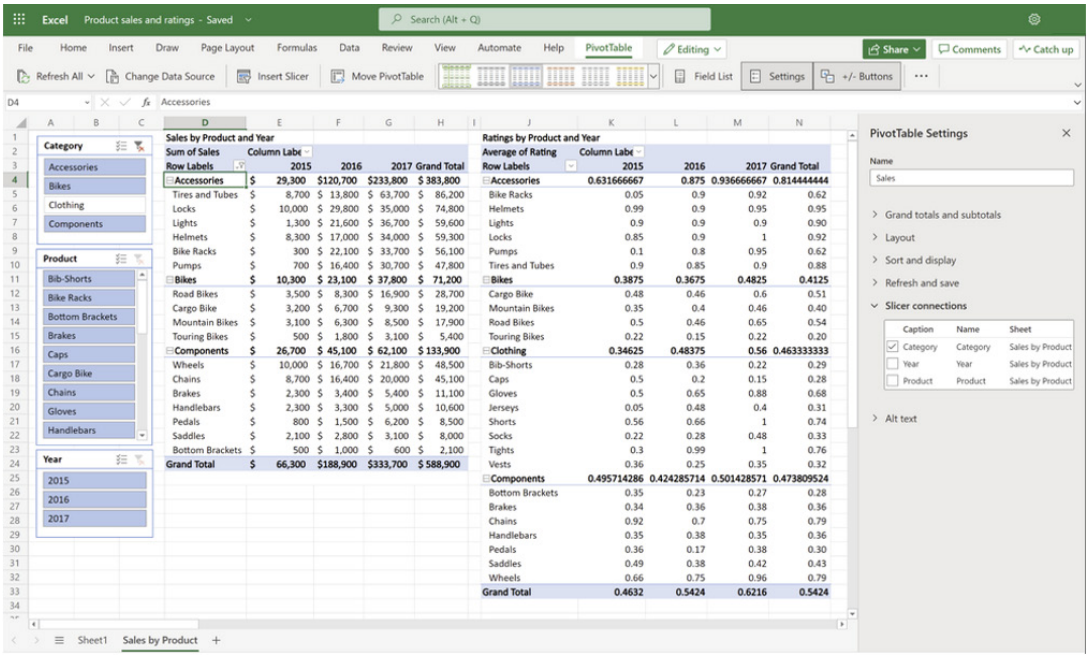
Sheet protection

Excel for the web now supports enabling and configuring sheet protection. Users may now turn sheet protection on and off, temporarily pause protection for just their session, and configure unlocked ranges and other sheet protection options.



'PivotTable Connections' in slicer settings pane

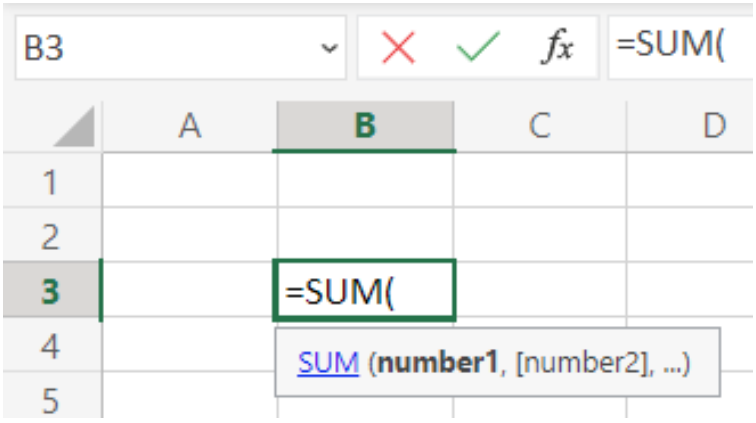
You may now customise which slicer applies to which PivotTable in Excel for the web. Simply choose your slicer connections in the PivotTable settings in Excel for the web:



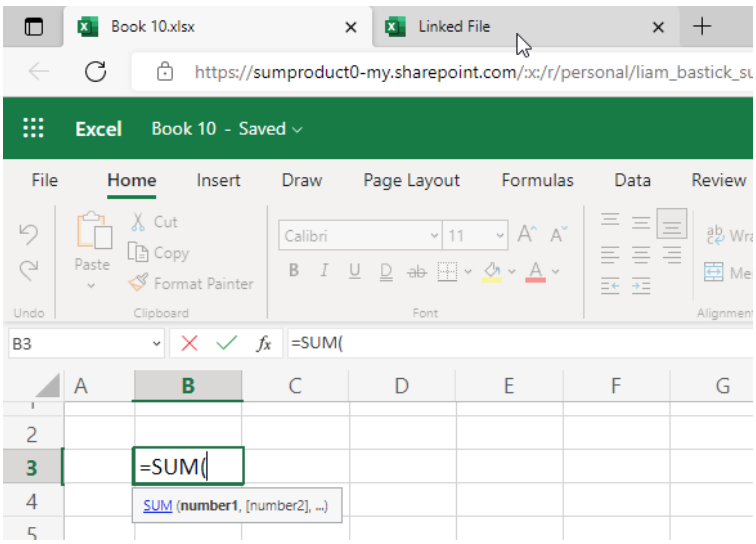
Semi-select for links creation

You may now create workbook links in Excel for the web using cross-workbook formula selection. This is also known as **semi-select**.

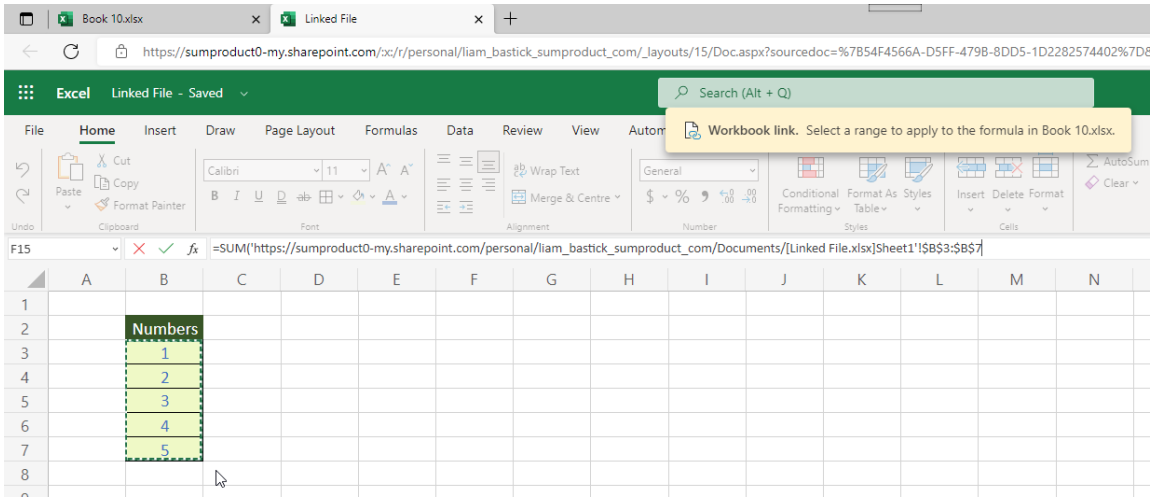
First, start typing your formula:



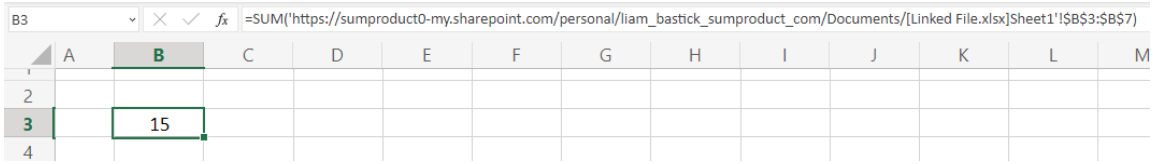
Then, click on the tab with the online Excel file you wish to link to:



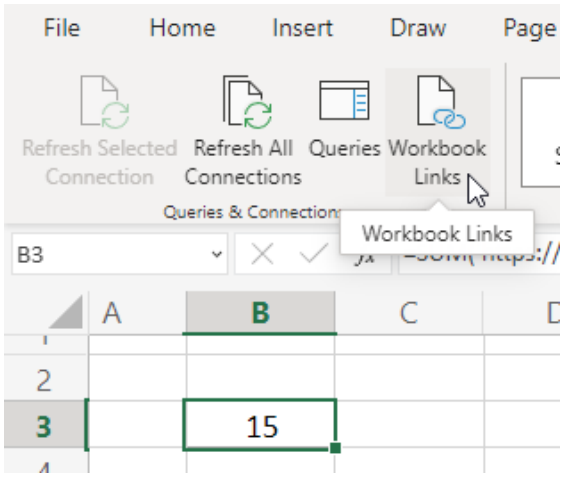
Select the cells to be linked in the usual way:



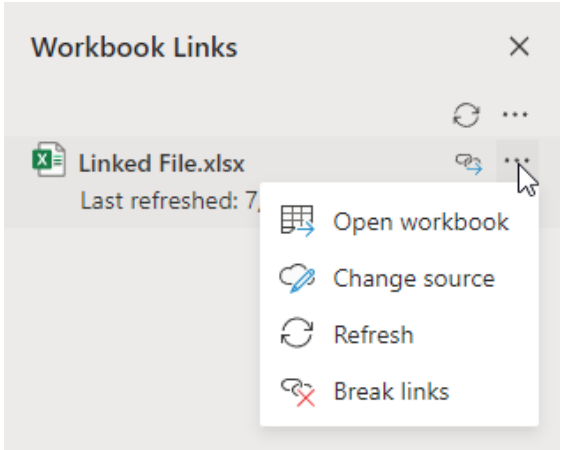
After closing any brackets *etc.* you should have your linked formula, *viz.*



You can manage links from the Data tab:



This opens the 'Workbook Links' pane, where you may refresh or modify links, *viz.*

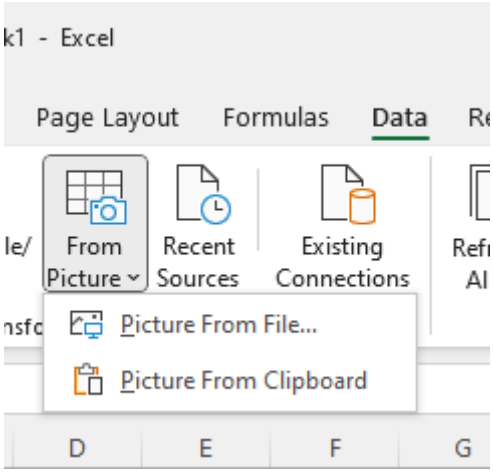


Data from Picture (Insider Beta)

With the Data from Picture feature in Excel for Windows, you can insert data from a picture on your clipboard or an image file from your computer. When it works, there's no need to type all the data: Excel can do it for you.

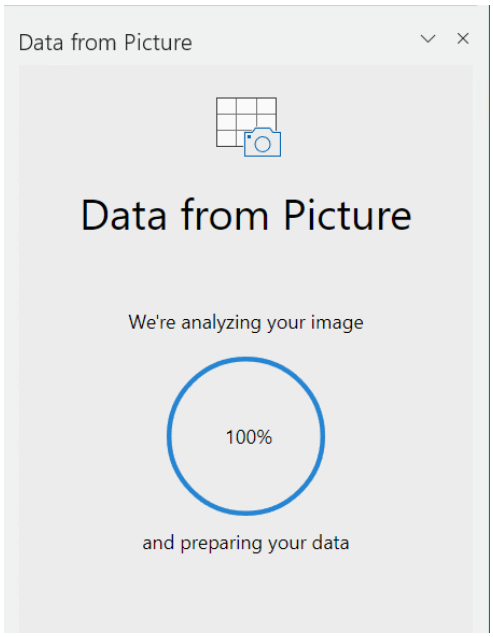
It works as follows:

1. Use one of the options below to capture the content you want to digitise:
    i. Select **Data -> From Picture -> Picture From File**

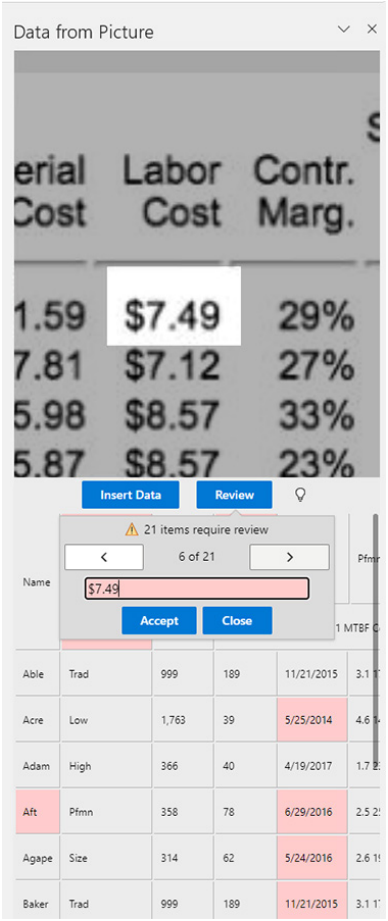


    ii. Copy an image of a table to your clipboard, *e.g.* take a screenshot of a table by pressing the **Windows Key + SHIFT + S**. Then, select **Data > From Picture -> Picture From Clipboard**

2. After you have captured the picture of the content you want to bring into Excel, the Data from Picture pane appears. This shows you the progress as the image is being analysed:



3. Review the results and make any corrections necessary, and then select Insert Data. The data is now in your Excel worksheet:



There are various ways you may use this feature:

- **Screen capture a table from a website:** if you've ever tried to copy and paste data from a website, you've likely noticed that it often results in formatting discrepancies. Instead, capture an image of the table (by pressing the **Windows Key + SHIFT + S**), and select **Data -> From Picture -> Picture From Clipboard**. Then, follow the instructions on the screen; you should get just what you're looking for
- **Take a picture of some printed data:** perhaps you'd like to get data from your previous tax returns into Excel and you only have printed copies. Simply take a picture of each one and transfer the pictures to your Windows computer (OneDrive may assist with this). Then, select **Data -> From Picture -> Picture From File**. Follow the on-screen instructions to convert the picture to data.

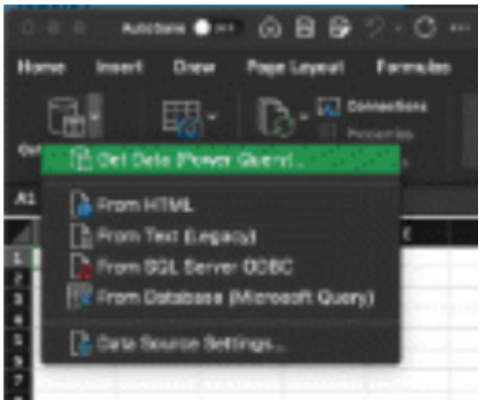
The Data from Picture feature is rolling out to Office Insiders running Beta Channel Version 2207 (Build 15402.20002) or later.

Provide automatic alternative (alt) text suggestions on charts and PivotCharts

As we try to adopt "better practice" in adding alternative text to charts, Excel for both Windows and Mac automatically recognise and generate "alt text" if you're using a screen reader and you land on a chart or Pivot Chart that is missing it.

Import from local text, CSV and XLSX files with data preview

You can now import data from local files such as Excel workbooks, Text and CSV files, using Power Query in Excel for Mac.



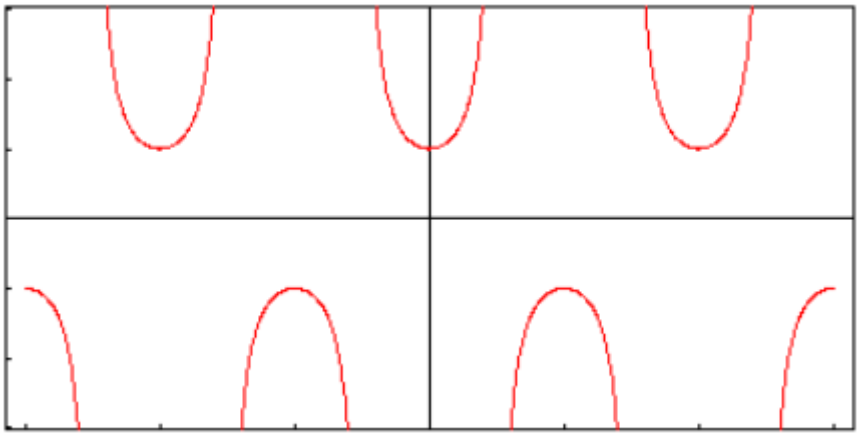
Microsoft has released an 'Excel Features Availability' flyer to detail all of these items:

Excel Features Availability

	Insider		Production				
Feature	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	Web
Data from picture	Version 2205 (Build 15316.20000) or later						
Sheet protection							June 2022
Semi-select for links creation							June 2022
Add "PivotTable Connections to Slicer settings pane							June 2022
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources							May 2022
Changing source file for workbook links							May 2022
Improved Recommended PivotTable experience	Version 2204 (Build 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later		Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20178) or later				
Shaping data with Power Query Editor		Version 16.61 (Build 22041701) or later					
Improved Find dialog and Find All						Version 16.60 (220410) or later	

You can find the updated version of this grid with the interactive links at aka.ms/ExcelFeaturesFlyer.

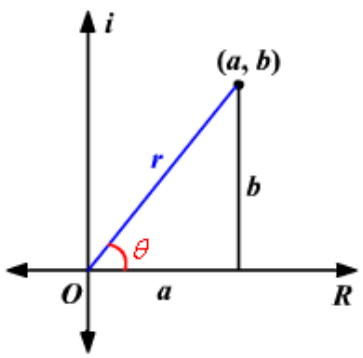
The A to Z of Excel Functions: IMSEC



An imaginary number is a complex number that can be written as a real number multiplied by the imaginary unit **i** (sometimes denoted **j**) which is defined by its property $\mathbf{i}^2 = -1$. In general, the square of an imaginary number **bi** is $-\mathbf{b}^2$. For example, 9**i** is an imaginary number, and its square is −81. Zero is considered to be both real and imaginary.

An **imaginary** number **bi** can be added to a **real** number **a** to form a **complex number** of the form **a + bi**, where the real numbers **a** and **b** are called, respectively, the **real** part and the **imaginary** part of the **complex number**.

The **polar form** of a complex number is another way to represent the number. The form **z = a + bi** is called the **rectangular form** of a complex number.



The horizontal axis is the real axis and the vertical axis is the imaginary axis. You can find the real and imaginary components in terms of **r** and **θ**, where **r** is the length of the vector and **θ** is the angle made with the real axis.

From the Pythagorean Theorem,

$$r^2 = a^2 + b^2$$

By using the basic trigonometric ratios,

$$\cos\theta = a / r \text{ and } \sin\theta = b / r$$

Therefore, multiplying each side by **r**:

$$r\cos\theta = a \text{ and } r\sin\theta = b$$

Therefore,

$$z = a + bi$$
$$\Leftrightarrow z = r\cos\theta + (r\sin\theta)i$$
$$\Leftrightarrow z = r(\cos\theta + i\sin\theta)$$

In the case of a complex number, **r** represents the **absolute value**, or **modulus** (where $r = |z| = \sqrt{a^2 + b^2}$), and the angle **θ** is called the **argument** of the complex number ($\theta = tan^{-1}\left(\frac{b}{a}\right)$ for **a** > 0 and $\theta = tan^{-1}\left(\frac{b}{a}\right)$ +π for **a** < 0).

The secant is simply the reciprocal of the cosine function. The **IMSEC** function returns the secant of a complex number in **x + yi or x + yj** text format.

$$\begin{aligned}
\sec(a+bi) &= \frac{1}{\cos(a+bi)} \\
&= \frac{1}{\cos a\cosh b - i\sin a\sinh b} \\
&= \frac{\cos a\cosh b + i\sin a\sinh b}{(\cos a\cosh b + i\sin a\sinh b)(\cos a\cosh b - i\sin a\sinh b)} \\
&= \frac{\cos a\cosh b + i\sin a\sinh b}{\cos^2 a\cosh^2 b - i^2\sin^2 a\sinh^2 b} \\
&= \frac{\cos a\cosh b + i\sin a\sinh b}{\cos^2 a\cosh^2 b + \sin^2 a\sinh^2 b}
\end{aligned}$$

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

IMSEC(inumber)

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

- **inumber:** this is required and represents the complex number for which you want to calculate the secant.

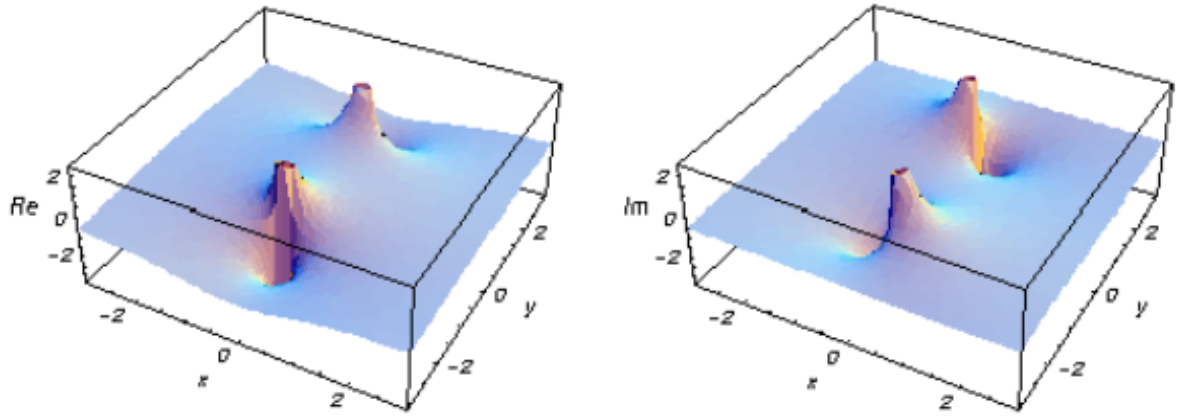
It should be further noted that:

- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMSEC** recognises either the **i** or **j** notation
- if **inumber** is a value that is not in the **x + yi or x + yj** text format, **IMSEC** returns the *#NUM!* error value
- if **inumber** is a logical value, **IMSEC** returns the *#VALUE!* error value
- if the complex number ends in **+i** or **-i** (or **j**), *i.e.* there is no coefficient between the operator and the imaginary unit, there must be no space, otherwise **IMSEC** will return an *#NUM!* error.

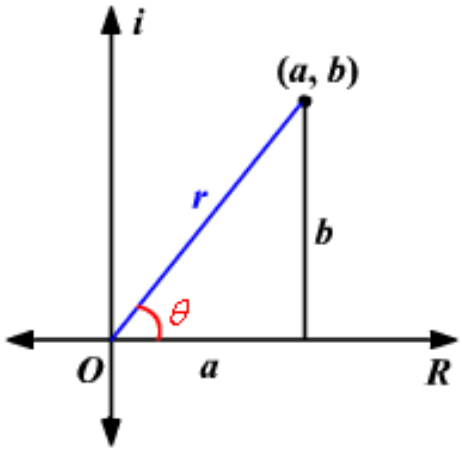
Please see our example below:

	A	B	C
1	Formula	Description	Result
2	=IMSEC("3 + 4i")	Secant of the complex number 3 + 4i	-0.0362534969158689+0.005164344607753 18i

The A to Z of Excel Functions: IMSECH



As stated above, the **polar form** of a complex number is another way to represent the number. The form **z = a + bi** is called the **rectangular form** of a complex number.



The horizontal axis is the real axis and the vertical axis is the imaginary axis. You can find the real and imaginary components in terms of **r** and **θ**, where r is the length of the vector and **θ** is the angle made with the real axis.

From the Pythagorean Theorem,

$$r^2 = a^2 + b^2$$

By using the basic trigonometric ratios,

$$\cos\theta = a / r \text{ and } \sin\theta = b / r$$

Therefore, multiplying each side by r:

$$r\cos\theta = a \text{ and } r\sin\theta = b$$

Therefore,

$$z = a + bi$$

$$\Leftrightarrow z = r\cos\theta + (r\sin\theta)i$$

$$\Leftrightarrow z = r(\cos\theta + i\sin\theta)$$

In the case of a complex number, r represents the **absolute value**, or **modulus** (where $r = |z| = \sqrt{a^2 + b^2}$), and the angle **θ** is called the **argument** of the complex number ($\theta = tan^{-1}\left(\frac{b}{a}\right)$ for **a** > 0 and $\theta = tan^{-1}\left(\frac{b}{a}\right)$ +π for **a** < 0).

The hyperbolic secant is the reciprocal of the hyperbolic cosine function.

$$\begin{aligned}
\operatorname{sech}(a+bi) &= \frac{1}{\cosh(a+bi)} \\
&= \frac{1}{\cosh a\cos b + i\sinh a\sin b} \\
&= \frac{\cosh a\cos b - i\sinh a\sin b}{(\cosh a\cos b + i\sinh a\sin b)(\cosh a\cos b - i\sinh a\sin b)} \\
&= \frac{\cosh a\cos b - i\sinh a\sin b}{\cosh^2 a\cos^2 b - i^2\sinh^2 a\sin^2 b} \\
&= \frac{\cosh a\cos b - i\sinh a\sin b}{\cosh^2 a\cos^2 b + \sinh^2 a\sin^2 b}
\end{aligned}$$

The **IMSECH** function returns the hyperbolic secant of a complex number in **x + yi** or **x + yj** text format.

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

IMSECH(inumber)

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

- **inumber:** this is required and represents the complex number for which you want to calculate the hyperbolic secant.

It should be further noted that:

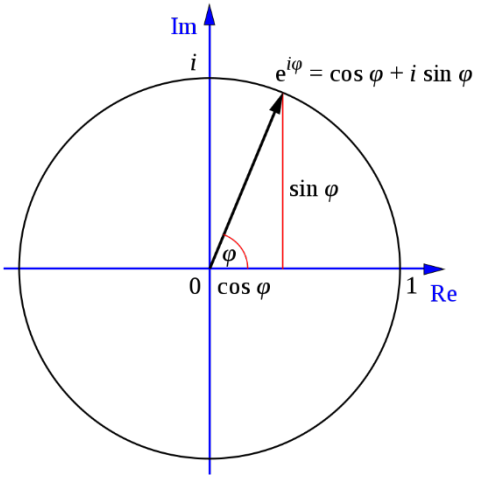
- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMSECH** recognises either the **i** or **j** notation
- if **inumber** is a value that is not in the **x + yi** or **x + yj** text format, **IMSECH** returns the *#NUM!* error value
- if **inumber** is a logical value, **IMSECH** returns the *#VALUE!* error value
- if the complex number ends in **+i** or **-i** (or j), *i.e.* there is no coefficient between the operator and the imaginary unit, there must be no space, otherwise **IMSECH** will return an *#NUM!* error.

Please see our next example below:

	A	B	C
1	Formula	Description	Result
2	=IMSECH("3 + 4i")	Hyperbolic secant of the complex number 3 + 4i	-0.065294027857947+0.07522496030277321

The A to Z of Excel Functions: IMSIN

Continuing with our imaginary numbers, using **Euler's Formula**,



$$e^{i\theta} = \cos\theta + i\sin\theta$$

Given $sin\ z = \frac{e^{iz}-e^{-iz}}{2i}$ by doing more mathematics than you would probably ever wish to read,

$$\begin{aligned}\sin a\cosh b + i\cos a\sinh b &= \frac{e^{ia}-e^{-ia}}{2i}\frac{e^{b}-e^{-b}}{2} + i\frac{e^{ia}+e^{-ia}}{2}\frac{e^{b}-e^{-b}}{2}\\ &= \frac{e^{b+ia}-e^{-b+ia}-e^{b-ia}+e^{-b-ia}-e^{b+ia}+e^{-b+ia}-e^{b-ia}+e^{-b-ia}}{4i}\\ &= \frac{e^{-b-ia}-e^{b-ia}}{2i}\\ &= \frac{e^{i(a+bi)}-e^{-i(a+bi)}}{2i}\\ &= \sin(a+bi)\end{aligned}$$

you eventually get:

$$\sin\left(x+yi\right) = \sin\left(x\right)\cosh\left(y\right) + \cos\left(x\right)\sinh\left(y\right)i$$

The **IMSIN** function returns the sine of a complex number in **x + yi** or **x + yj** text format.

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

IMSIN(inumber)

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

- inumber: this is required and represents the complex number for which you want to calculate the sine.

It should be further noted that:

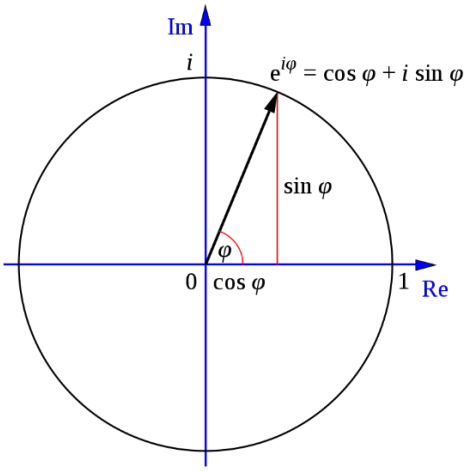
- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMSIN** recognises either the **i** or **j** notation
- if **inumber** is a value that is not in the **x + yi** or **x + yj** text format, **IMSIN** returns the *#NUM!* error value
- if **inumber** is a logical value, **IMSIN** returns the *#VALUE!* error value
- if the complex number ends in **+i** or **-i** (or **j**), *i.e.* there is no coefficient between the operator and the imaginary unit, there must be no space, otherwise **IMSIN** will return an *#NUM!* error.

Please see our penultimate example below:

	A	B	C
1	Formula	Description	Result
2	=IMSIN("3 + 4i")	Sine of the complex number 3 + 4i	3.85373803791938-27.0168132580039i

The A to Z of Excel Functions: IMSINH

From above,



$$e^{i\theta} = cos\,\theta + i\,sin\,\theta$$

Given $sin\ z = \frac{e^{iz}-e^{-iz}}{2i}$, then $sinh\ z = \frac{e^{z}-e^{-z}}{2}$. It then follows

$$\begin{aligned}\sinh a\cos b + i\cosh a\sin b &= \frac{e^{a}-e^{-a}}{2}\frac{e^{ib}+e^{-ib}}{2} + i\frac{e^{a}+e^{-a}}{2}\frac{e^{ib}-e^{-ib}}{2i}\\ &= \frac{e^{a+ib}-e^{-a+ib}+e^{a-ib}-e^{-a-ib}+e^{a+ib}+e^{-a+ib}-e^{a-ib}-e^{-a-ib}}{4}\\ &= \frac{e^{(a+bi)}-e^{-(a+bi)}}{2}\\ &= \sinh(a+bi)\end{aligned}$$

you eventually get:

sinh(x + yi) = sinh x cos y – i cosh x sin y

The **IMSINH** function returns the hyperbolic sine of a complex number in **x + yi** or **x + yj** text format.

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

IMSINH(inumber)

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

- inumber: this is required and represents the complex number for which you want to calculate the hyperbolic sine.

It should be further noted that:

- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMSINH** recognises either the i or j notation
- if **inumber** is a value that is not in the **x + yi** or **x + yj** text format, **IMSINH** returns the *#NUM!* error value
- if **inumber** is a logical value, **IMSINH** returns the *#VALUE!* error value
- if the complex number ends in **+i** or **-i** (or **j**), *i.e.* there is no coefficient between the operator and the imaginary unit, there must be no space, otherwise **IMSINH** will return an *#NUM!* error.

Please see our final example below:

	A	B	C
1	Formula	Description	Result
2	=IMSINH("3 + 4i")	Hyperbolic sine of the complex number 3 + 4i	-6.548120040911-7.619231720321 41i

More Excel Functions next month.

Beat the Boredom Suggested Solution

Sometimes when modelling you need to identify the location of the nth occurrence of a character in a text string, perhaps to truncate the text or to manipulate it in some other fashion.

The Challenge

Sometimes when modelling you need to identify the location of the **n**th occurrence of a character in a text string, perhaps to truncate the text or to manipulate it in some other fashion.

Character	l
Occurrence Number	3
Text	Hello Lesley
Result	10

This month's challenge was to write a formula in one cell that would identify the nth occurrence of a character in a text string. There were some constraints:

- the formula needed to be in just one cell (no "helper" cells)
- this was a formula challenge – so no Power Query / Get & Transform or Text to Columns!
- the formula must work in **all** current versions of Excel (so no VBA, dynamic arrays, **LAMBDA**, **LET** or user defined functions)
- the model may be large or unstable, so no volatile functions were allowed
- the formula must be case sensitive. For example, in the illustration above the third occurrence of "l" in "Hello Lesley" is in position 10, *i.e.* "Hello Leslley" – the capital "L" is ignored.

There were bonus points too: as an additional challenge, a second formula was sought to locate the **last** occurrence in the same text string too, subject to the same above restrictions.

Suggested Solution

There are two common functions in Excel that allow you to rummage through a given text string:

1. **SEARCH(find_text, within_text, [start_number])** is a search function which is **not** case sensitive, but does allow for wildcard characters. It seeks out the first instance of a character or characters (typed in inverted commas) in the **within_text** text string. The **start_number** argument is optional (hence the square brackets in the syntax), so that the first few characters in a text string may be ignored. If the **find_text** cannot be located within **within_text**, the error *#VALUE!* is returned
2. **FIND(find_text, within_text, [start_number])** is another search function which **is** case sensitive, but does **not** allow wildcard characters. It seeks out the first instance of a character or characters (typed in inverted commas) in the **within_text** text string. The **start_number** argument is optional (hence the square brackets in the syntax), so that the first few characters in a text string may be ignored. If the **find_text** cannot be located within **within_text** the error *#VALUE!* is returned.

Here, we need to create a formula that is case sensitive, which therefore forces us to use **FIND** rather than **SEARCH** (we do not need to consider wild cards here). The problem is, like its **SEARCH** counterpart, **FIND** seeks out the **first** occurrence of **find_text** – so we will need to be crafty.

If we cannot find the nth occurrence of a character (**find_text**) within a text string (**within_text**), then we need to locate the nth occurrence of the character and replace it with an alternative character that we KNOW will occur once and only once in the text string. Then, we may simply **FIND** this character.

But what character should we use as the replacement? We need it to be one that cannot be easily typed into a cell. A cursory glance of the internet will suggest **@** or **CHAR(160)**, the non-breaking space (often used in HTML code). I would suggest neither:

- **@** is now a special character associated with dynamic arrays / legacy formulae in some versions of Excel and may be problematic
- **CHAR(160)** causes problems with some text functions in Excel already, such as **TRIM** which cannot remove it, and therefore is best avoided.

I am going to suggest **CHAR(1)**, which is the unprintable "Start of Heading" character in the ASCII system. I don't think anyone will be typing this into a text string!

So how do we replace the nth occurrence of a given character with **CHAR(1)**? Two common functions come to mind:

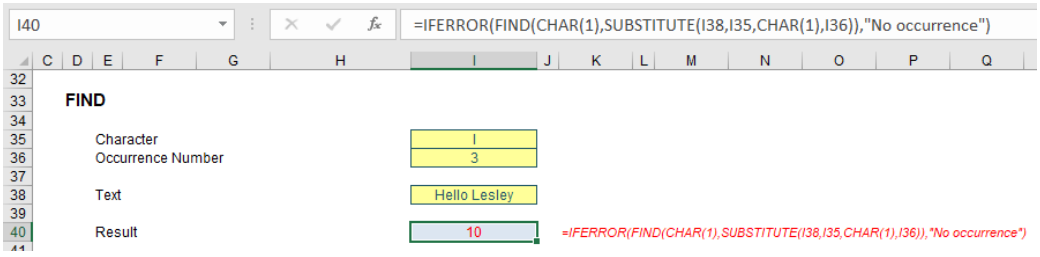
1. **REPLACE(old_text, start_number, number_of_characters, new text)** is a function that allows you to swap one or several characters in a text string with another character or a set of characters. In the **old_text**, it seeks out the characters to be swapped by starting at the **start_number** of the text string and replacing the **number_of_characters** with the new text. For example,

   =REPLACE("Get the answer next time",5,10,"it right")

   becomes "Get it right next time"
2. **SUBSTITUTE(text, old_text, new_text, [instance_number])** is similar to **REPLACE**, as it replaces one or more instances of a given character or text string (**old_text**) in a **text** string with a specified character or string (**new_text**). The optional **instance_number** cites the occurrence of **old_text** you wish to replace. If this is omitted, all occurrences will be replaced, *viz.*

	A	B	C
1	Text	Revised Result	Formula
2	Brazil 0 USA 0	Brazil 1 USA 1	=SUBSTITUTE(A2,"0","1")
3		Brazil 1 USA 0	=SUBSTITUTE(A2,"0","1",1)
4		Brazil 0 USA 1	=SUBSTITUTE(A2,"0","1",2)
5			

Only **SUBSTITUTE** allows us to specify the **instance_number** – but that's enough: we now have a plan of attack.



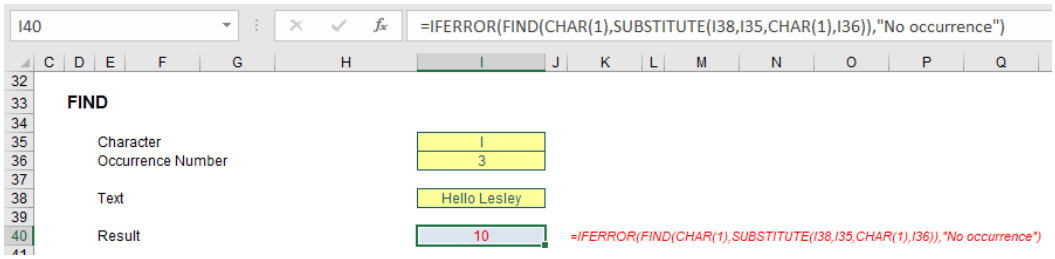
In the example illustrated *(above)*, I have used the formula

=IFERROR(FIND(CHAR(1),SUBSTITUTE(I38,I35,CHAR(1),I36)),"No occurrence")

To explain how this works:

- **SUBSTITUTE(I38,I35,CHAR(1),I36)** substitutes the third (**I36**) occurrence of "l" (**I35**) with **CHAR(1)** in the text "Hello Lesley" in cell **I38**. This now guarantees only one occurrence of the character which now occupies the position that needs to be identified
- **FIND(CHAR(1),SUBSTITUTE(I38,I35,CHAR(1),I36))** then returns the position of this unique character **CHAR(1)**
- **IFERROR(FIND(CHAR(1),SUBSTITUTE(I38,I35,CHAR(1),I36)),"No occurrence")** simply provides an error trap should there be no occurrence of the desired character (cell **I35**).

Taking this challenge one step further, to find the last occurrence is slightly trickier as we don't know how many occurrences there are. You could construct a calculation to reverse the text string, but the formula I have used is as follows:



Perhaps not the largest graphic ever committed to an article, but the formula is given by

=IFERROR(FIND(CHAR(1),SUBSTITUTE(I62,I60,CHAR(1),LEN(I62)-LEN(SUBSTITUTE(I62,I60,""))),"No occurrence")

This works similarly to the earlier calculation:

- **SUBSTITUTE(I62,I60,"")**: this element replaces the selected character with an empty string
- **LEN(I62)-LEN(SUBSTITUTE(I62,I60,""))**: this calculates how many times the selected character occurs in the string. This is because this formula subtracts the length of the string without any occurrences of the character (since all instances are replaced with an empty string) from the length of the original text string with all the original instances of the character intact
- **SUBSTITUTE(I62,I60,CHAR(1),LEN(I62)-LEN(SUBSTITUTE(I62,I60,"")))**: this is where we first came in! We now know how many occurrences there are of our chosen character, and this calculation substitutes the final occurrence of "l" (**I60**) with **CHAR(1)** in the text "Hello Lesley HALL" in cell **I62**. This now guarantees only one occurrence of the character which now occupies the position that needs to be identified
- **FIND(CHAR(1),SUBSTITUTE(I62,I60,CHAR(1),LEN(I62)-LEN(SUBSTITUTE(I62,I60,""))))**: then returns the position of this unique character **CHAR(1)**
- **IFERROR(FIND(CHAR(1),SUBSTITUTE(I62,I60,CHAR(1),LEN(I62)-LEN(SUBSTITUTE(I62,I60,"")))),"No occurrence")**: this simply provides an error trap should there be no occurrence of the desired character (cell **I60**).

It may seem horrible, but broken down, it's not so bad.

Word to the Wise

In case you are wondering why this challenge may be at all relevant in the real world, these sorts of issues occur all the time. For example, you may have serial numbers such as

ISBN 978-3-16-148410-0-SERIAL-78-8

ISBN 978-1-940313-1-0-2-RADIUS-15-19

ISBN 978-0-7334-2609-PUBL-2-4

and wish to extract the text strings "SERIAL", "RADIUS" and "PUBL". This would be possible using extrapolations of the ideas discussed above.

Until next time.

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Excel Tips and Tricks	29 Aug 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	30 - 31 Aug 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	28 -30 Sep 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	5 Oct 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	6 - 7 Oct 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	9 - 11 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	16 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	17 - 18 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	7 - 9 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	14 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	15 - 16 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we look again at the **CTRL** and **SHIFT** keys, but this time combined with the letters that Excel uses:

Keystroke	What it does
CTRL + SHIFT + A	Insert arguments in a formula
CTRL + SHIFT + B	Address Book (only when emailing, not as an attachment)
CTRL + SHIFT + F	Font face
CTRL + SHIFT + L	Toggle AutoFilter
CTRL + SHIFT + O	Select Comments
CTRL + SHIFT + P	Font size
CTRL + SHIFT + T	Toggle Total Row

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.

Sydney Address: SumProduct Pty Ltd, Suite 803, Level 8, 276 Pitt Street, Sydney NSW 2000
New York Address: SumProduct Pty Ltd, 48 Wall Street, New York, NY, USA 10005
London Address: SumProduct Pty Ltd, Office 7, 3537 Ludgate Hill, London, EC4M 7JN, UK
Melbourne Address: SumProduct Pty Ltd, Ground Floor, 470 St Kilda Road, Melbourne, VIC 3004
Registered Address: SumProduct Pty Ltd, Level 14, 440 Collins Street, Melbourne, VIC 3000

contact@sumproduct.com
www.sumproduct.com
+61 3 9020 2071