

# Sum Product

NEWSLETTER #116 - July 2022

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



**Finally – a book update!!** The third in our continuing series of *Financial Modelling* books is closer to release! We have fielded MANY queries since we first revealed it was in the pipeline and after a multitude of delays, it is finally set to be issued (although you might not wish to hold your breath for a physical copy!).

This month, we also look at the updated Recommended PivotTables and provide greater detail on last month's announcements regarding the updated **TEXT** functions. Furthermore, Microsoft has started a new Excel Updates section which we have decided to pass on too!

Combined with another Beat the Boredom Challenge, plus our usual articles on Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers and Power BI Updates, this newsletter refuses to diminish in size. We continue with our imaginary Excel Functions (all too real as they tell us, e.g. **IMREAL**) this month, and we even have some more Keyboard Shortcuts if you **SHIFT** down to the end of the newsletter and promise to **CTRL** yourselves!

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

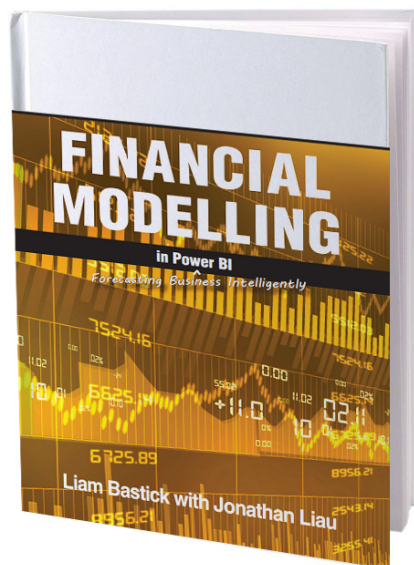
*Liam Bastick*, Managing Director, SumProduct



## Financial Modelling in Power BI: finally, coming soon!!

Originally announced in our 100th newsletter (!), pandemics, shipping issues, Power BI updates and other unforeseen technical delays saw this landmark book delayed and delayed further. Quite embarrassing, but not our fault!!

But no more: final editing checks have been made – and *it is coming out soon!!*



You may or may not be familiar with the first two books in this series, *Introduction to Financial Modelling* and *Continuing Financial Modelling*. These tomes covered how to model in Excel. But time moves on. And more and more of us are realising that whilst the humble spreadsheet is a powerful enabler, there's a young(er) upstart in town, gaining traction every day.

Power BI – Microsoft's self-service business intelligence and data visualisation tool – is becoming both more versatile and more popular each month. Today's analysts need to use the best, most powerful and flexible analytical software for their work. More and more of us are realising Power BI fits the bill. We'd be foolish to stand still.

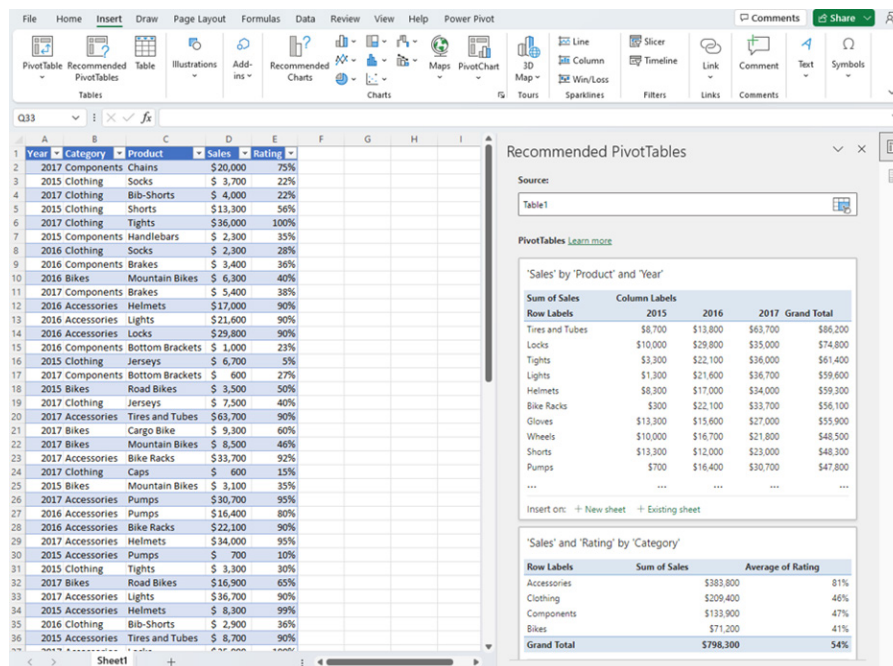
This book addresses the way forward for financial modelling. Many working in banking and finance are creating their financial models in Excel and then importing them into Power BI for graphical interpretation and further analysis. We're not going to do that. This book seeks to build the model *in Power BI*.

Whilst not a "basic" book on Power BI, it is looking to lay the foundations of building a model in Power BI. It's not going to cover everything, but it will set the scene and construct the basics. To be released electronically within the next month or so, physical copies *should* be available before the end of the year.

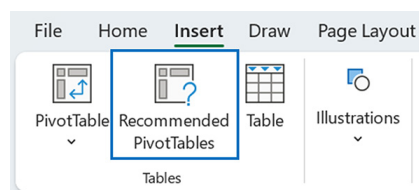
# Improved Recommended PivotTables

Recommended PivotTables have been updated to become more intelligent by utilising the same technology as the Analyze Data feature, also in Excel. In addition, the dialog box interface has been replaced

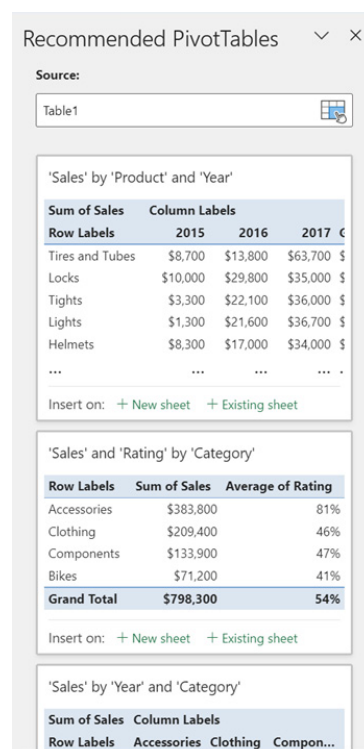
with an all-new redesigned panel. This updated user experience makes it easier to view all of your options and simpler to change your data selection before inserting a recommended PivotTable.



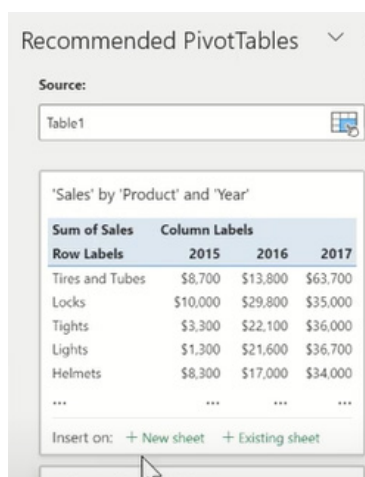
To see how the new experience works, open an Excel workbook containing data and then select **Insert -> Recommended PivotTables**.



In the Recommended PivotTables pane, review the suggested options:



If desired, place your cursor in the Source box and choose a different data selection on the grid, and then finally select either **+New sheet** or **+Existing sheet** to insert the recommended PivotTable in your workbook.



This feature is currently available to Microsoft 365 subscribers in English, French, Spanish, German, Simplified Chinese and Japanese. Microsoft advises support for more languages is coming soon.

It should be noted that Recommended PivotTables currently does not support the analysis of datasets larger than 1.5 million cells. As an alternative, you can filter and reduce your data set to a supported number of cells, copy it to another location, and then run Recommended PivotTables on that smaller data set.

For best results:

- use clean, tabular data for best results
- format your data as an Excel table (click anywhere in your data and press **CTRL + T** or **Insert -> Table**)
- make sure all columns have headers, with a single row of unique, non-blank labels for each column
- avoid double rows of headers or merged cells

- if you have complicated or nested data, use Power Query to transform it (for example, to unpivot your data) so it is organised in columns with a single header row.

To take advantage of this feature, you must have:

- a reliable internet connection
- 'Optional connected experiences' switched on.

The improved Recommended PivotTables experience analyses your data with intelligent services. For those so concerned, you may access the Microsoft Privacy Statement for additional details on data security. In particular, you should note that Microsoft advises that Recommended PivotTables uses material from third parties.

This updated feature will be rolled out to Office Insiders running Beta Channel Version 2204 (Build 15128.10000) or later.

## Recent Text and Array Functions: an Update

It all came in too late for last month's newsletter, but regular readers may recall I did mention that Microsoft has elected to "...enhance the signatures of the text functions..." as follows:

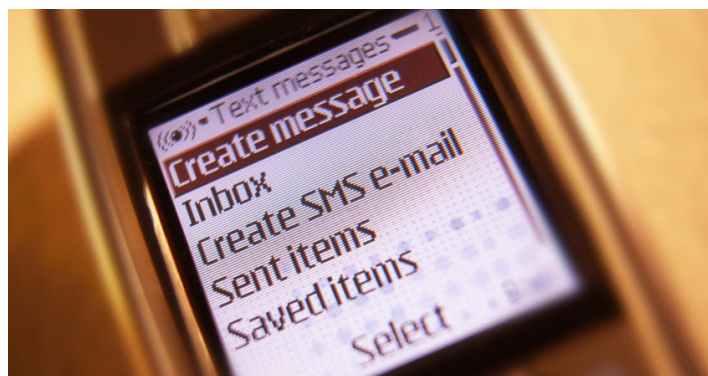
- added **[match end]** to **TEXTBEFORE** and **TEXTAFTER**. This will allow you to request a match against the end of the text
- renamed **[ignore case]** to **[match mode]** for **TEXTBEFORE** and **TEXTAFTER**
- added **[match mode]** to **TEXTSPLIT** to be more consistent with **TEXTBEFORE** and **TEXTAFTER**

- added **[if not found]** to **TEXTBEFORE** and **TEXTAFTER**. This expression is returned instead of **#N/A** when there is no delimiter match.

The updated text functions will become available to Beta channel users starting with build 15316.20000. The function documentation will be updated to reflect these changes in the coming weeks. Please note that you will need to update your **TEXTSPLIT** functions if you used the **[pad with]** argument, as this argument has moved from the fifth to the sixth argument. All of the other changes are backwards compatible.

Let's take a look at the three functions that have been updated.

### The TEXTBEFORE function



The **TEXTBEFORE** function returns the string of text that occurs before a given substring (*i.e.* a character or set of characters) in that string. It is the opposite of the **TEXTAFTER** function. **TEXTBEFORE** has the following syntax:

**TEXTBEFORE(text, delimiter, [instance number], [match mode], [match end], [if not found])**

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

- **text**: this is required and represents the text string you are searching within. Wildcard characters are not allowed
- **delimiter**: this is also required and represents the **text** in the text string that marks the point before which you wish to extract



- **instance number**: this is the first optional argument and denotes the nth instance of the delimiter before which you wish to extract. By default, this is equal to one [1]. If a negative number is used here, the function starts searching for the **delimiter** from the end rather than the beginning
- **match mode**: this is the second optional argument, and determines whether the **text** search is case-sensitive. The default is case-sensitive. If used, you should enter one of the following two [2] values:
  - **0**: case sensitive
  - **1**: case insensitive
- **match end**: also optional, this treats the end of the text as a delimiter. By default this is viewed as an exact match. If used, you should enter one of the following two [2] values:
  - **0**: do not match the delimiter against the end of the text
  - **1**: match the delimiter against the end of the text
- **if not found**: this too is an optional argument and provides the

value should no match be found. If not used and no match is found, #N/A will be returned.

It should be further noted that:

- when searching with an empty **delimiter** value, **TEXTBEFORE** matches immediately. It returns empty **text** when searching from the front (if **instance number** is positive) and the entire **text** when searching from the end (if **instance number** is negative)
- Excel returns an **#VALUE!** error if the instance number is zero (the default is one) or if the **instance number** is greater than the length of the **text**
- Excel returns an **#N/A** error if the **delimiter** does not occur within the **text**
- Excel returns an **#N/A** error if the **instance number** is greater than the number of occurrences of the **delimiter** within the **text**.

Please see the examples below:

	A	B	C
1	Data		
2	Many many occurrences of "many" in this sentence.		
3		empty	
4	Liam Bastick		
5	SumProduct		
6			
7			
8	Formula	Description	Result
9	=TEXTBEFORE(A2, "Many")	Identifies there is no text before "Many" is used as the <b>delimiter</b> (case sensitive is default).	
10	=TEXTBEFORE(A2, "many")	Identifies there is text before "many" is used as the <b>delimiter</b> (case sensitive is default).	Many
11	=TEXTBEFORE(A2, "many", FALSE)	Identifies the text before "many" is used as the <b>delimiter</b> (case sensitive using FALSE).	Many
12	=TEXTBEFORE(A2, "many", TRUE)	Identifies there is no text before "many" is used as the <b>delimiter</b> (case insensitive using TRUE).	
13	=TEXTBEFORE(A2, "many", 2, FALSE)	Identifies text before second occurrence of "many" (case sensitive).	Many many occurrences of f
14	=TEXTBEFORE(A2, "many", 2, TRUE)	Identifies text before second occurrence of "many" (case insensitive).	Many
15	=TEXTBEFORE(A2, "Many", 2, FALSE)	No second occurrence of "Many" (case sensitive).	#N/A
16	=TEXTBEFORE(A2, "Many", 2)	No second occurrence of "Many" (case sensitive).	#N/A
17	=TEXTBEFORE(A2, "many", -1, FALSE)	Identifies there is text before the final occurrence of "many" is used as the <b>delimiter</b> (case sensitive).	Many many occurrences of f
18	=TEXTBEFORE(A2, "many", -2, FALSE)	Identifies there is text before the penultimate occurrence of "many" is used as the <b>delimiter</b> (case sensitive).	Many
19	=TEXTBEFORE(A2, "Many", -2, FALSE)	Insufficient occurrences of "Many" (case sensitive).	#N/A
20	=TEXTBEFORE(A2, )	The <b>delimiter</b> is not specified.	
21	=TEXTBEFORE(A2, "")	The <b>delimiter</b> is not specified.	
22	=TEXTBEFORE(A3, "Many")	Insufficient occurrences of "Many" (case sensitive) (cell blank).	#N/A
23	=TEXTBEFORE(A4, " ")	Insufficient occurrences of "Many" (case sensitive) (cell blank).	Liam
24	=TEXTBEFORE(A5, " ", "No space you idiot")	Insufficient occurrences of "Many" (case sensitive) (cell blank).	No space you idiot
25	=TEXTBEFORE(A5, " ", 1)	Insufficient occurrences of "Many" (case sensitive) (cell blank).	SumProduct

## The TEXTAFTER function



The **TEXTAFTER** function returns the string of text that occurs after a given substring (*i.e.* a character or set of characters) in that string. It is the opposite of the **TEXTBEFORE** function. **TEXTAFTER** has the following syntax:

**TEXTAFTER**(text, delimiter, [instance number], [match mode], [match end], [if not found])

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

- **text:** this is required and represents the **text** string you are searching within. Wildcard characters are not allowed
- **delimiter:** this is also required and represents the text in the text string that marks the point after which you wish to extract
- **instance number:** this is the first optional argument and denotes the **n**th instance of the **delimiter** after which you wish to extract. By default, this is equal to one [1]. If a negative number is used here, the function starts searching for the **delimiter** from the end rather than the beginning



- **match mode:** this is the second optional argument, and determines whether the **text** search is case-sensitive. The default is case-sensitive. If used, you should enter one of the following two [2] values:
  - o **0:** case sensitive
  - o **1:** case insensitive
- **match end:** also optional, this treats the end of the **text** as a **delimiter**. By default this is viewed as an exact match. If used, you should enter one of the following two [2] values:
  - o **0:** do not match the **delimiter** against the end of the **text**
  - o **1:** match the **delimiter** against the end of the **text**
- **if not found:** this too is an optional argument and provides the value should no match be found. If not used and no match is found, #N/A will be returned.

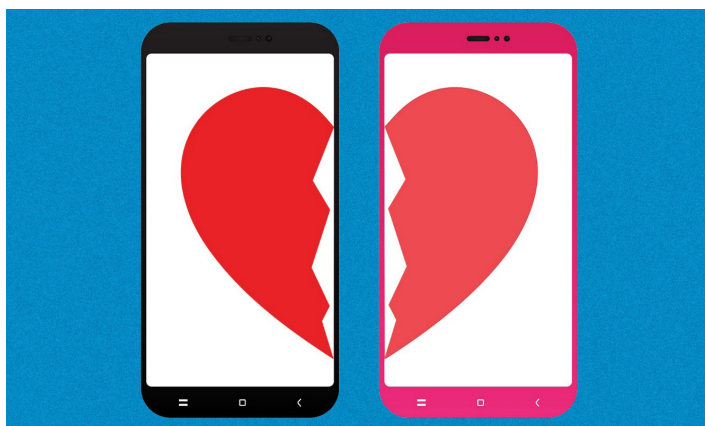
It should be further noted that:

- when searching with an empty **delimiter** value, **TEXTAFTER** matches immediately. It returns the entire **text** when searching from the front (if **instance number** is positive) and empty **text** when searching from the end (if **instance number** is negative)
- Excel returns an #VALUE! error if the **instance number** is zero (the default is one) or if the **instance number** is greater than the length of the **text**
- Excel returns an #N/A error if the **delimiter** does not occur within the **text**
- Excel returns an #N/A error if the **instance number** is greater than the number of occurrences of the **delimiter** within the **text**.

Please see relevant examples below:

A	B	C
1		
2	Data	
3	Many many occurrences of "many" in this sentence.	
4	Liam Bastick	
5	SumProduct	
6		
7		
8	Formula	Description
9	=TEXTAFTER(A2, "Many")	Identifies there is text after "Many" is used as the <b>delimiter</b> (case sensitive is default).
10	=TEXTAFTER(A2, "many")	Identifies there is text after "many" is used as the <b>delimiter</b> (case sensitive is default).
11	=TEXTAFTER(A2, "many", FALSE)	Identifies the text after "many" is used as the <b>delimiter</b> (case sensitive using FALSE).
12	=TEXTAFTER(A2, "many", TRUE)	Identifies there is text after "many" is used as the <b>delimiter</b> (case insensitive using TRUE).
13	=TEXTAFTER(A2, "many", 2, FALSE)	Identifies text after second occurrence of "many" (case sensitive).
14	=TEXTAFTER(A2, "many", 2, TRUE)	Identifies text after second occurrence of "many" (case insensitive).
15	=TEXTAFTER(A2, "Many", 2, FALSE)	No second occurrence of "Many" (case sensitive).
16	=TEXTAFTER(A2, "Many", 2)	No second occurrence of "Many" (case sensitive).
17	=TEXTAFTER(A2, "many", -1, FALSE)	Identifies there is text after the final occurrence of "many" is used as the <b>delimiter</b> (case sensitive).
18	=TEXTAFTER(A2, "many", -2, FALSE)	Identifies there is text after the penultimate occurrence of "many" is used as the <b>delimiter</b> (case sensitive).
19	=TEXTAFTER(A2, "Many", -2, FALSE)	Insufficient occurrences of "Many" (case sensitive).
20	=TEXTAFTER(A2, )	The <b>delimiter</b> is not specified.
21	=TEXTAFTER(A2, "")	The <b>delimiter</b> is not specified.
22	=TEXTAFTER(A3, "Many")	Insufficient occurrences of "Many" (case sensitive) (cell blank).
23	=TEXTAFTER(A4, " ")	Insufficient occurrences of "Many" (case sensitive) (cell blank).
24	=TEXTAFTER(A5, " ", "No space you idiot")	Insufficient occurrences of "Many" (case sensitive) (cell blank).
25	=TEXTAFTER(A5, " ", 1)	Insufficient occurrences of "Many" (case sensitive) (cell blank).

## The TEXTSPLIT function



The **TEXTSPLIT** function is intended to work like the Text to Columns button on the Data tab of the Ribbon, almost like the “inverse” of the **TEXTJOIN** function. It allows you to split a given text across rows or down columns. **TEXTSPLIT** has the following syntax:

**TEXTSPLIT(text, [column delimiter], [row delimiter], [ignore empty], [match mode], [pad with])**

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

- **text:** this is required and represents the text string you wish to split
- **column delimiter:** this is optional and denotes one or more characters that specify where to spill the **text** across columns
- **row delimiter:** this is optional and denotes one or more characters that specify where to spill the **text** down rows

- **ignore empty:** another optional argument, you should specify TRUE to create an empty cell when two delimiters are used. This argument defaults to FALSE, which means don't create an empty cell
- **match mode:** this is the second optional argument, and determines whether the text search is case-sensitive. The default is case-sensitive. If used, you should enter one of the following two [2] values:
  - o **0:** case sensitive
  - o **1:** case insensitive

- **pad with:** not to be confused with Pad Thai, this final optional argument “pads” the resulting text range where cells would otherwise be blank. The default is N/A.

Please note that you will need to update any **TEXTSPLIT** functions previously written if you used the **pad with** argument, as this argument has moved from the fifth to the sixth argument.

If there is more than one delimiter (row or column), then an array

constant must be used. For example, to split by both a comma (,) and a period (full stop, .), use **=TEXTSPLIT(text, {",", "."})**.

Just for a change, some more examples:

Just for a change, some more examples:

	A	B	C	D	E	F	G	H	I	J
1	Data									
2	Many sentences. There are three. Just three.									
3	padding									
4										
5										
6		Many	sentences.		There	are	three.		Just	three.
7										
8		<b>=TEXTSPLIT(A2, " ")</b>								

	A	B	C	D
1	Data			
2	Many sentences. There are three. Just three.			
3	padding			
4				
5				
6		Many sentences	There are three	Just three
7				
8		<b>=TEXTSPLIT(A2, ". ")</b>		

	A	B	C
1	Data		
2	Many sentences. There are three. Just three.		
3	padding		
4			
5			
6		Many sentences	<b>=TEXTSPLIT(A2,, ". ")</b>
7		There are three	
8		Just three	

	A	B	C	D	E	F
1	Data					
2	Many sentences. There are three. Just three.					
3	padding					
4						
5						
6		Many	sentences	#N/A	#N/A	#N/A
7				There	are	three
8				Just	three	#N/A
9			#N/A	#N/A	#N/A	#N/A
10						
11		<b>=TEXTSPLIT(A2, " ", " ", ". ")</b>				
12						

	A	B	C	D
1	Data			
2	Many sentences. There are three. Just three.			
3	padding			
4				
5				
6		Many	sentences	#N/A
7		There	are	three
8		Just	three	#N/A
9				
10				
11		<b>=TEXTSPLIT(A2, " ", " ", ". ", TRUE)</b>		
12				

	A	B	C	D	E	F
1	Data					
2	Many sentences. There are three. Just three.					
3	padding					
4						
5						
6		Many	sentences	#N/A	#N/A	#N/A
7				There	are	three
8				Just	three	#N/A
9			#N/A	#N/A	#N/A	#N/A
10						
11						
12						

**=TEXTSPLIT(A2, " ", ".", FALSE)**

	A	B	C	D
1	Data			
2	Many sentences. There are three. Just three.			
3	padding			
4				
5				
6		Many sen	ences	#N/A
7		There are	hree	#N/A
8		Jus		hree
9				
10				
11				

**=TEXTSPLIT(A2, "t", ".", TRUE, 0)**

	A	B	C	D
1	Data			
2	Many sentences. There are three. Just three.			
3	padding			
4				
5				
6		Many	sentences	padding
7		There	are	three
8		Just	three	padding
9				
10				
11				

**=TEXTSPLIT(A2, " ", ".", TRUE,,A3)**

	A	B	C	D	E	F	G	H	I	J	K	L
1	Data											
2	Many sentences. There are three. Just three.											
3	padding											
4												
5												
6		Many	sentences		There	are		three		Just		three
7												
8												
9												

**=TEXTSPLIT(A2, {" ", ".", " "})**

	A	B	C	D	E
1	Data				
2	Many sentences. There are three. Just three.				
3	padding				
4					
5					
6		Many			
7		sentences			
8					
9					
10		There			
11		are			
12		three			
13					
14					
15		Just			
16		three			
17					

**=TEXTSPLIT(A2,, {" ", ".", " "})**



## 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...

Filtered month	March
Month	
March	
April	
May	
August	
September	
October	
November	

We have gone through Dynamic Arrays and the value they provide to Excel in detail, including the solutions we can create with them. However, as we continue to try to tackle ever more perplexing challenges, we occasionally reach some awkward obstacles.

Which brings us to this month’s challenge.

We have a list of months in a year in this case. We now use filters to see specific data daily. What would you do if you wanted to derive the first visible cell after applying the filter, which is commonly performed formulaically using dynamic arrays?

For example, if someone is looking at the sale of a company in a specific country while using the **FILTER** function, the Table’s and reports’ headings alter in line with the country name for which data is shown.

The use of macros and VBA could be a simple solution to this problem, but it is also a well-known reality that most Excel users are unfamiliar with VBA and macros, and that they will not run in all situations / environments. As a result, we must also look for a solution that does not require the use of such automated wizardry.

- This is a formula challenge, so no Power Query / Get & Transform or VBA!

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 Funnel charts.*

The Funnel chart is effectively a form of Bar chart. The difference between the two is that the bars in a Funnel chart are centred across the width of the Plot Area instead of starting from the vertical axis.

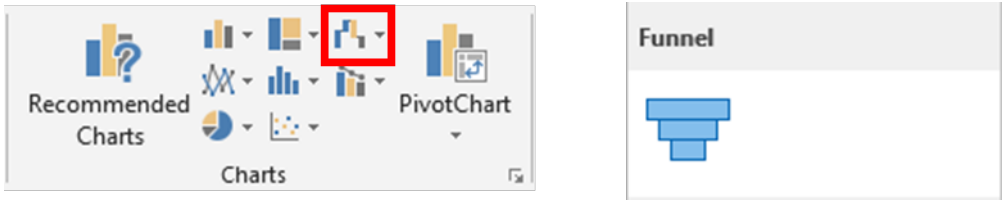
A Funnel chart can be used to show values across stages or categories. Typically, the values are sorted from highest to lowest so that when they are graphed, the chart resembles a funnel. An example of when a Funnel chart could be used is to show the volumes at each stage in a sales process, commencing with the number of general enquiries, then those that become genuine prospects, those who respond to quotes provided, those that make contact to discuss / negotiate, and finally those who place orders. Such a scenario would naturally result in the numbers

being in descending order, but not all data that is used to create Funnel charts follows this rule.

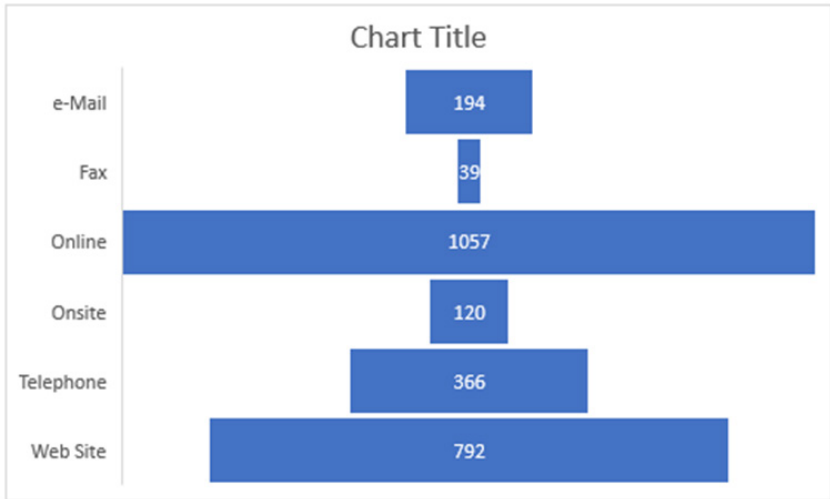
The data table required for a Funnel chart is very straightforward. The stages or categories are entered into one column and the values are entered in a second column. For example, I have the following information about the six methods by which customers of a company place work orders / requests. Management is trying to ascertain which are the most popular methods that customers use to place work orders and, given recent issues with their online system and their web pages, whether they should consider investing more in upgrading their internet-based offerings.

Work Orders / Requests	
Source	Volume
e-Mail	194
Fax	39
Online	1,057
Onsite	120
Telephone	366
Web Site	792

I highlight the data table and create a Funnel chart. The Funnel chart can be found under the Insert tab of the Ribbon, under the last small icon along the top of the Charts section. There is only one configuration option for the Funnel chart in Excel:

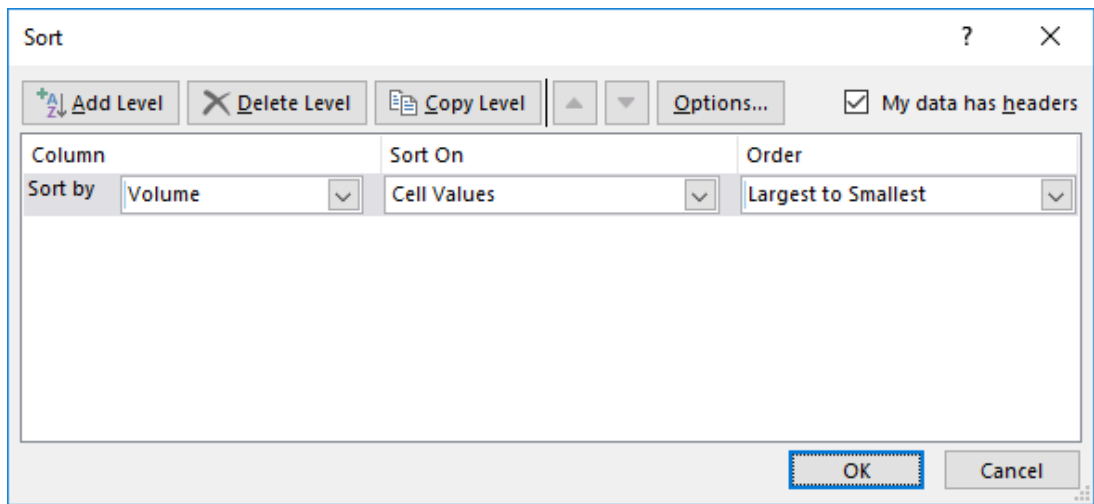


My initial chart looks like the one below:



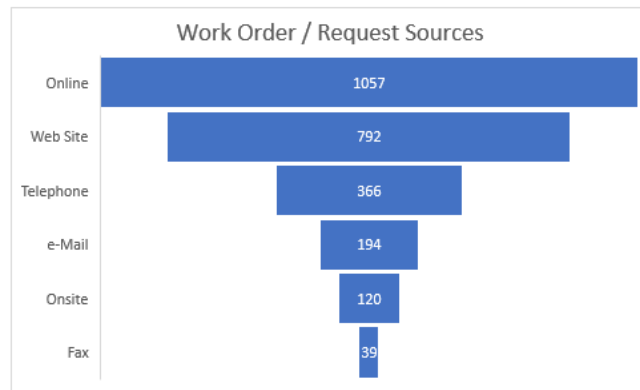
The first obvious thing I notice is that this Funnel chart does not resemble a funnel. Clearly, this is due to the figures not being in descending order. The scenario used is one where the data is not reflecting stages in a process, but merely counts of occurrences, so my figures do not naturally reduce as I move from one category to the next. It would be more meaningful though if the figures were in descending order given management's

specific concerns. Therefore, I highlight the data table including the column headings but excluding the table header, then go to the Home tab on the Ribbon, select 'Sort & Filter', and finally, 'Custom Sort'. Within the Sort screen that appears, tick the 'My data has headers' option, then sort by the Volume column, from largest to smallest, and click OK.

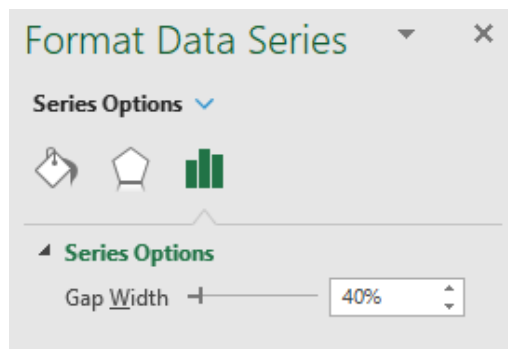


The data and chart will reflect the change in the order of the information:

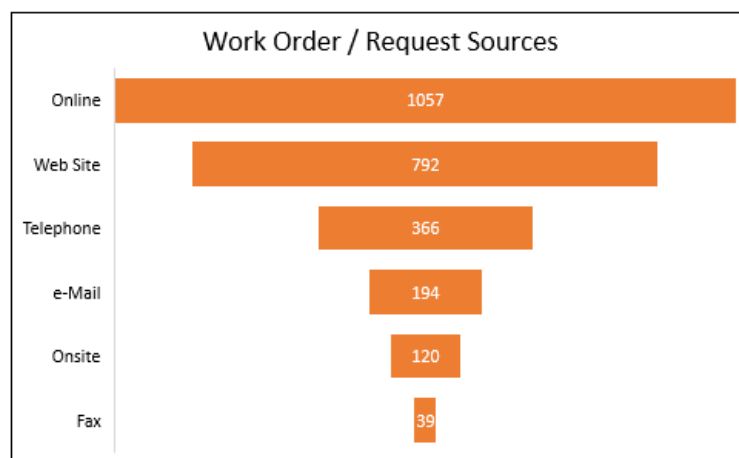
Work Orders / Requests	
Source	Volume
Online	1057
Web Site	792
Telephone	366
e-Mail	194
Onsite	120
Fax	39



To further format this chart, I right-click on the bar and choose 'Format Data Series'. In here, I change the 'Gap Width' to 40% and then I change the colour of the bar to orange.



The final product is shown below:



Do note that this differs from a Tornado chart insofar as the latter shows deviations from a given norm, both positive and negative, whereas corresponding bars are merely centred here.

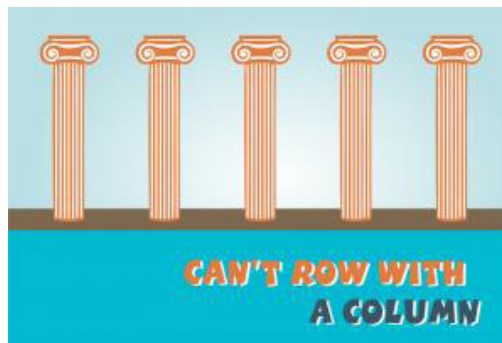
More next month...

## 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. Continuing our series about using **ListObjects** to manipulate Tables within an Excel workbook in VBA, this month we consider the unique properties of **ListColumn**.



Whilst doing some research to eloquently explain the difference between rows and columns, we came across this quote:

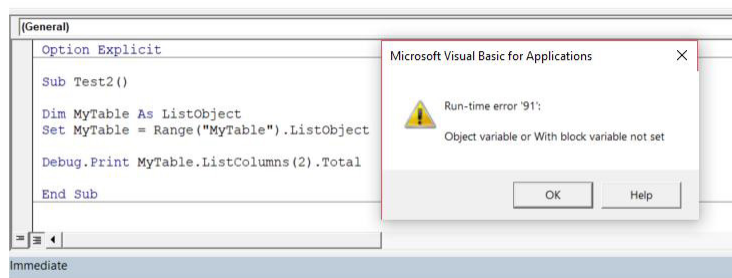


There are a few things that may be done with **ListColumn** that are not possible with **ListRow** due to the nature of how you think to store data. Consider each row as a record in a database or an item. There may be an unlimited number of records as data is added. Columns are the fields of

data which are defined. They are attributes of the record storing a single piece of information in relation to that item. The column is the structure of your data whereas the row is the data itself.

The properties that are unique to **ListColumn** are as follows:

- **DataBodyRange**: remember that **Range** refers to the whole column including the header. This returns just the information in the column under the header, the data portion of the table
- **Name**: this returns the name stored in the header of the column
- **Total**: this returns the value in the Totals row. Note, the table must have a total otherwise it will return the following error as it is looking for an object that does not exist:



- **TotalsCalculation**: this gives the ability to change the type of calculation in the Totals row which we covered in an earlier newsletter
- **XPath**: **XPath** provides the ability to change the **XPath** information stored about the table. **XPath** uses path expressions to select nodes or node-sets in an XML document. XML is designed to manage and share structured data. Using **XPath** in VBA, you may manipulate the structure and tags in order to format in a certain way for importing / exporting between applications.

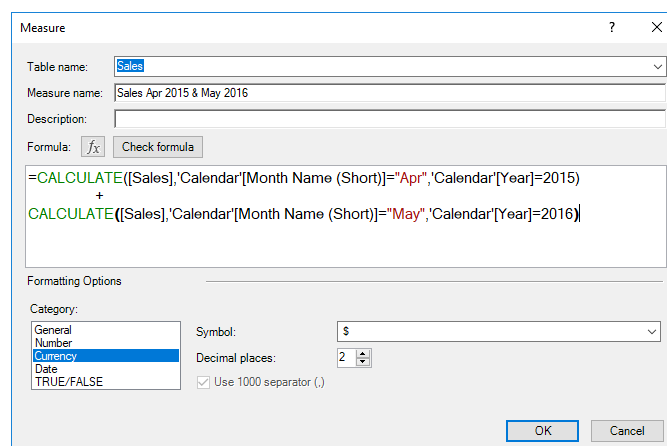
That wraps up all things **ListColumns** and **ListRows** and concludes the distinct ranges in a table!

More next time.

## Power Pivot Principles

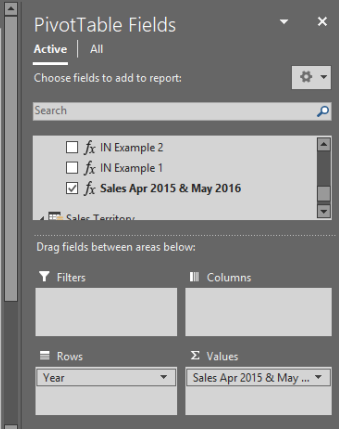
We continue our series on the Excel COM add-in, Power Pivot. This month, we look at using concatenation with the **FILTER** function to combine filter criteria.

Imagine we wanted to create a measure that calculates the total sales amount for April 2015 and May 2016 from our data set. How would we do that? One simple solution would combine two **CALCULATE** functions together:

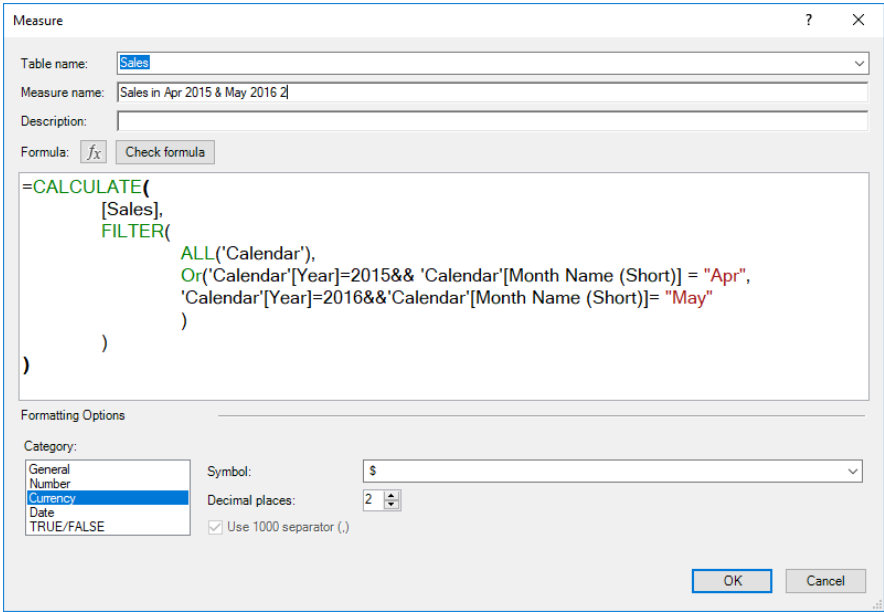


That would yield the following:

	A	B	C	D
1				
2				
3		Year	Sales Apr 2015 & May 2016	
4		2014	\$16,620.75	
5		2015	\$16,620.75	
6		2016	\$16,620.75	
7		2017	\$16,620.75	
8		2018	\$16,620.75	
9		Grand Total	\$16,620.75	
10				
11				
12				
13				
14				



A more delicate solution would involve a combination of the **CALCULATE**, **FILTER**, **ALL** and **OR** functions to create the measure.



```

=CALCULATE(
    [Sales],
    FILTER(
        ALL('Calendar'),
        OR('Calendar'[Year]=2015&&'Calendar'[Month Name (Short)] = "Apr",
        'Calendar'[Year]=2016&&'Calendar'[Month Name (Short)] = "May"
    )
)

```

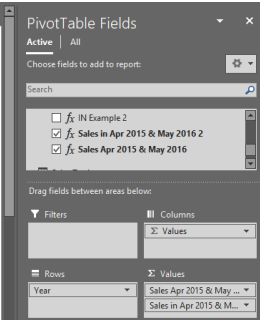
Note that we can concatenate two criteria into one by using the '&&' syntax:

'Calendar'[Year]=2015&&'Calendar'[Month Name (Short)] = "Apr",

'Calendar'[Year]=2016&&'Calendar'[Month Name (Short)] = "May"

This allows us to combine the two criteria into one **CALCULATE** function.

	A	B	C	D
1				
2				
3		Year	Sales Apr 2015 & May 2016	Sales in Apr 2015 & May 2016 2
4		2014	\$16,620.75	\$16,620.75
5		2015	\$16,620.75	\$16,620.75
6		2016	\$16,620.75	\$16,620.75
7		2017	\$16,620.75	\$16,620.75
8		2018	\$16,620.75	\$16,620.75
9		Grand Total	\$16,620.75	\$16,620.75
10				
11				
12				
13				



The type of criteria that can be used in this manner is not limited to dates. We can also use **Product Category Keys** and **Standard Costs** criteria in the same manner to build a similar measure.

Measure

Table name: Sales

Measure name: Sales of Product Cat 1, Std Cost < 50 & Cat 2 Std Cost < 100

Description:

Formula:

```
=CALCULATE(
    [Sales],
    FILTER(
        ALL('Product SubCategory'),
        OR('Product SubCategory'[ProductCategoryKey]=1&&'Product SubCategory'[Standard Cost]<50,
        'Product SubCategory'[ProductCategoryKey]=2&&'Product SubCategory'[Standard Cost]<100
        )
    )
)
```

Formatting Options

Category: General Number Currency Date TRUE/FALSE

Symbol: \$

Decimal places: 2

☒ Use 1000 separator (,)

This measure yields:

Year	Sales of Product Cat 1, Std Cost < 50 & Cat 2 Std Cost < 100
2014	\$18,554.30
2015	\$50,294.15
2016	\$458,309.80
2017	\$603,822.85
<b>Grand Total</b>	<b>\$1,130,981.10</b>

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](http://www.sumproduct.com/blog). If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at the **M** function **Table.Join**.

Whilst there are often multiple ways to achieve the same objective in Power Query, the purpose of this month's example is to become familiar with the **M** function, **Table.Join**. The description in the Microsoft help pages is as follows:

**Table.Join**(table1 as table,

key1 as any,

table2 as table,

key2 as any,

optional joinKind as nullable number,

optional joinAlgorithm as nullable number,

optional keyEqualityComparers as nullable list) as table

This joins the rows of **table1** with the rows of **table2** based on the equality of the values of the key columns selected by **key1** (for **table1**) and **key2** (for **table2**).

By default, what is known as an "inner join" is performed, however an optional **joinKind** may be included to specify the type of join. Options include:

- **JoinKind.Inner**
- **JoinKind.LeftOuter**
- **JoinKind.RightOuter**
- **JoinKind.FullOuter**
- **JoinKind.LeftAnti** and
- **JoinKind.RightAnti**.

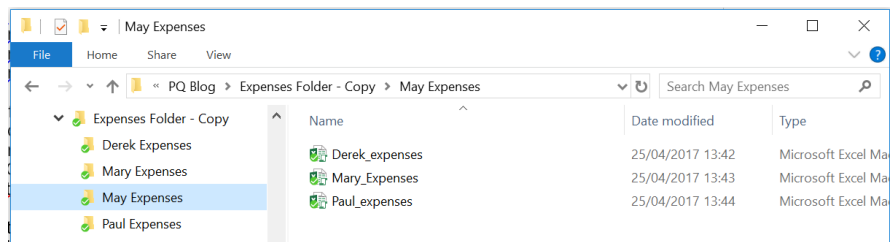


The different types of join work as follows:

- **Inner:** only matching rows
- **Left Outer:** all from first, matching from second table
- **Right Outer:** all from second, matching from first table
- **Full Outer:** all rows from both tables
- **Left Anti:** rows only in first table
- **Right Anti:** rows only in second table.

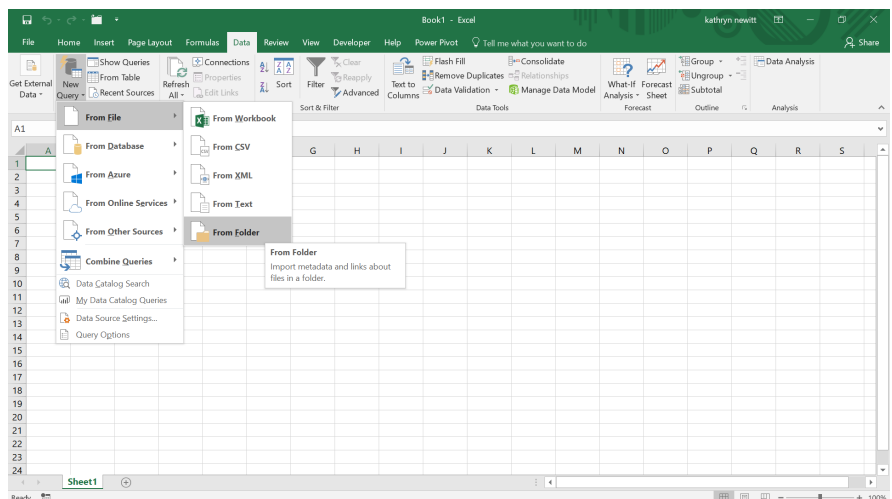
In order to understand how **Table.Join** works, let's consider an example of where it may be used.

Our fictional salespeople have been busy sending in their expenses, but the administration has not been ideal. We have a folder which contains all the expenses that we need to upload, but there have been inconsistencies that mean some of the expense files appear in more than one of the subfolders.

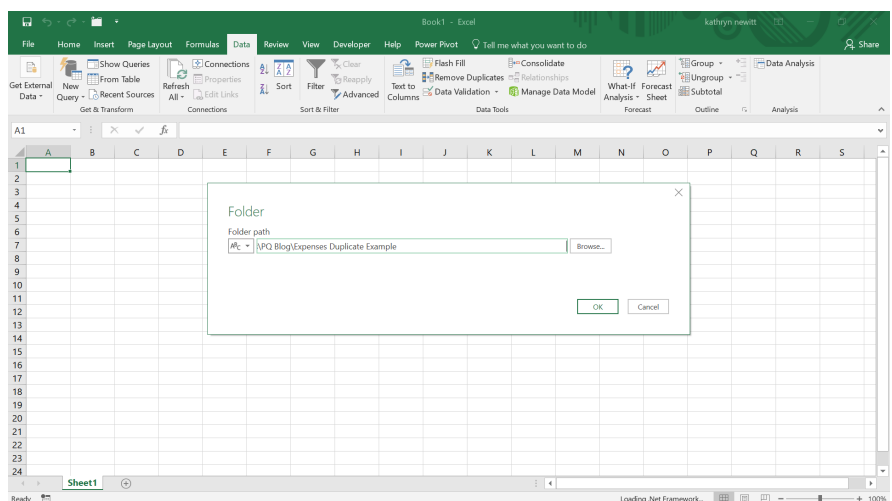


May is not one of our imaginary salespeople – ‘May Expenses’ holds expenses created in May. However, not all of the expense files are in here, so how can we see which files are duplicated? The example is simple, but this method will apply to much larger folders.

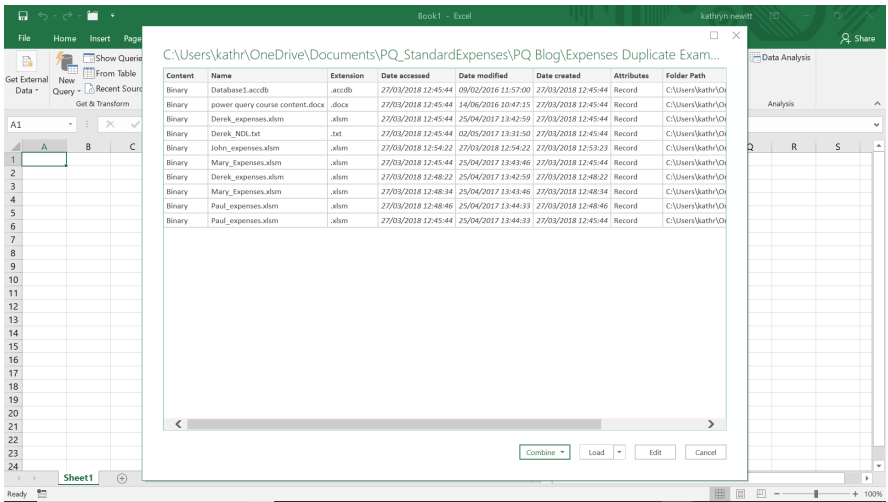
In a new workbook, let's create a new query ‘From Folder’ on the ‘From File’ option in the ‘Get and Transform’ section of the ‘Data Tab’:



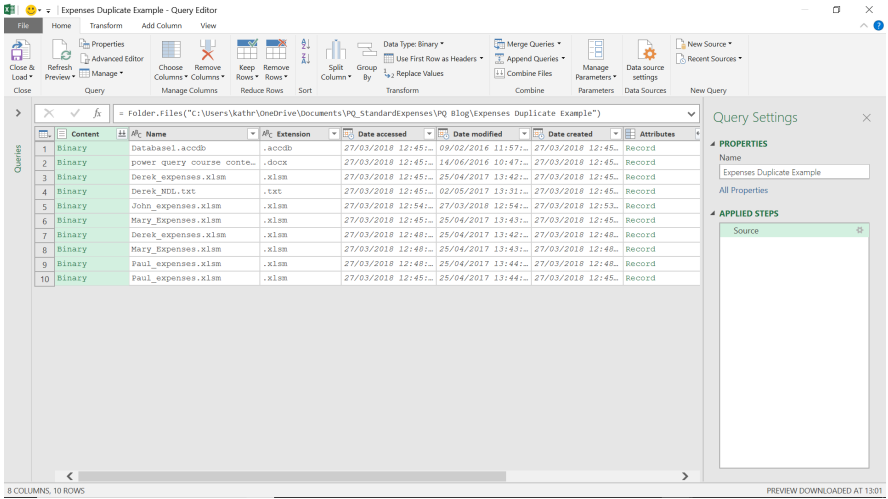
For the purposes of illustration, the folder here is located in ‘...PQ Blog\Expenses Duplicate Example’; this folder contains all the expense folders where we are looking for duplicates.



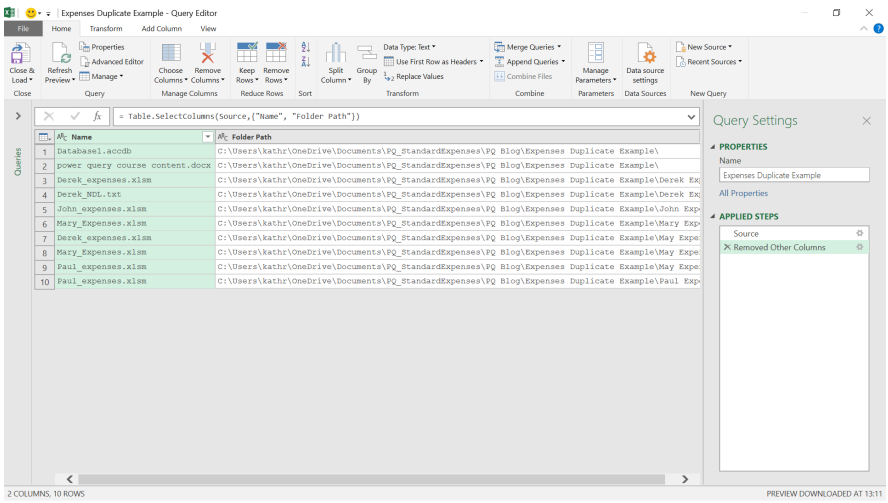
When we click 'OK', we can see a list of all the files in the folders beneath the specified folder:



For this task, we want to see the duplicates, not combine or load all the files, so we will choose to 'Edit'. In fact, you should always choose to edit first, to make sure you are getting the data that you expect.

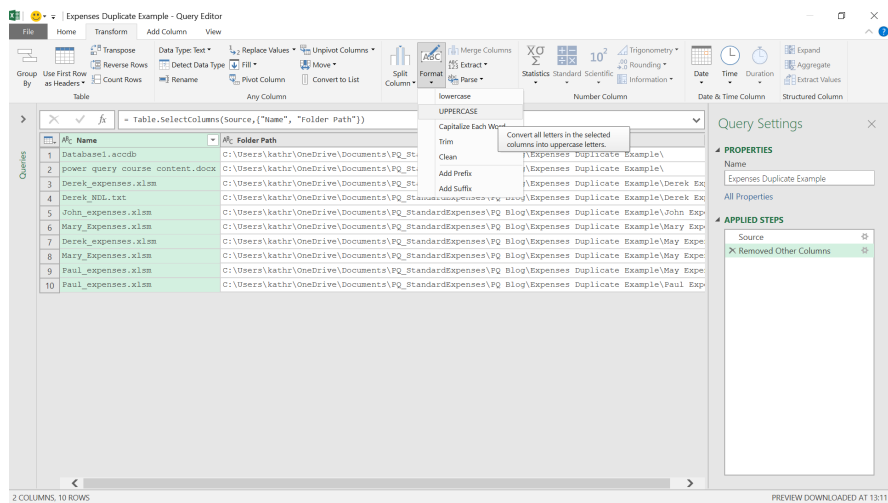


For this exercise, we only need the **Name** and **Folder Path** columns, so we will select those columns and choose to 'Remove Other Columns' from the 'Remove Columns' dropdown on the 'Manage Columns' section of the 'Home' tab.



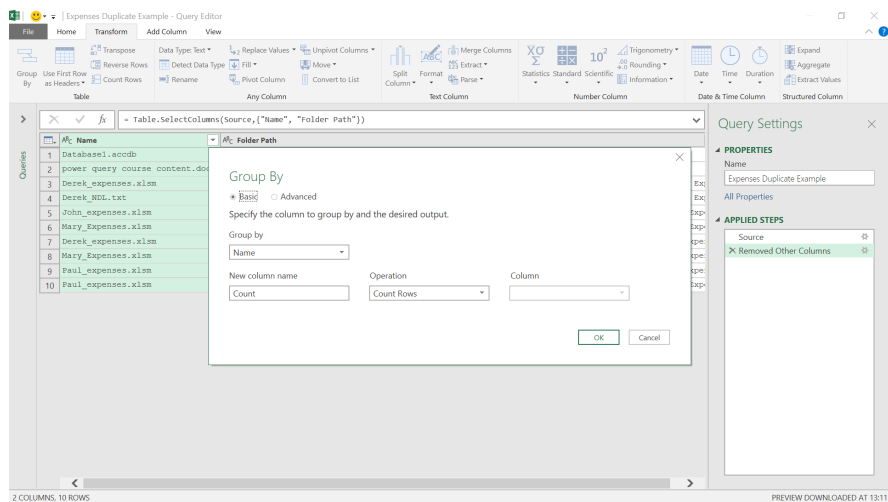
Arguably, we should also upshift the **Name** column for comparison. This really depends whether a file with uppercase text is really the same file as one with the same name, but using lowercase text. This will depend

upon the practices of the users concerned. If required, we would then select the **Name** column and use the 'UPPERCASE' option in the 'Format' dropdown on the 'Transform' menu as shown below.

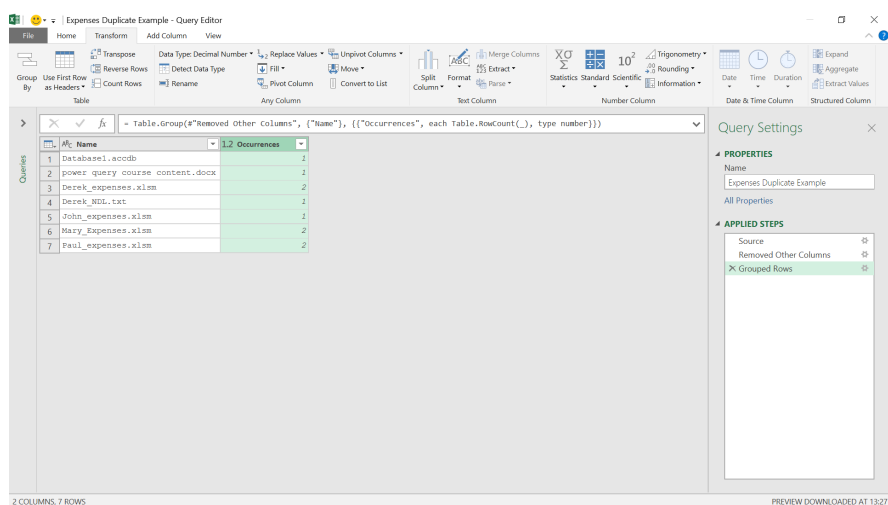


We will not be upshifting the text, so our final step for this part of the process is 'Removed Other Columns'. We need to remember this step, as will be shown later.

The next step is to group by name and to do this, we use the 'Group By' option in the 'Table' section of the 'Transform' tab. It should be added at this point that there are other ways to do this grouping, but we are doing it this way to allow us to use **Table.Join** later!

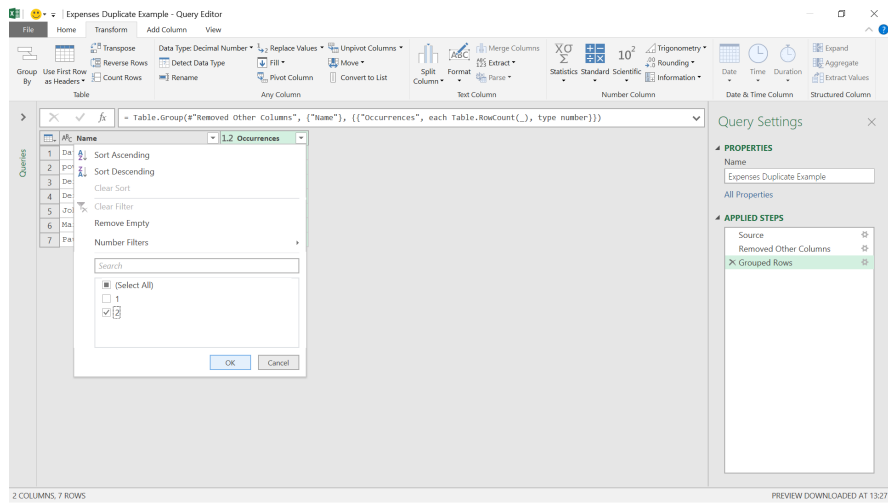


We will call our new column **Occurrences**, which will count the number of rows for each name.

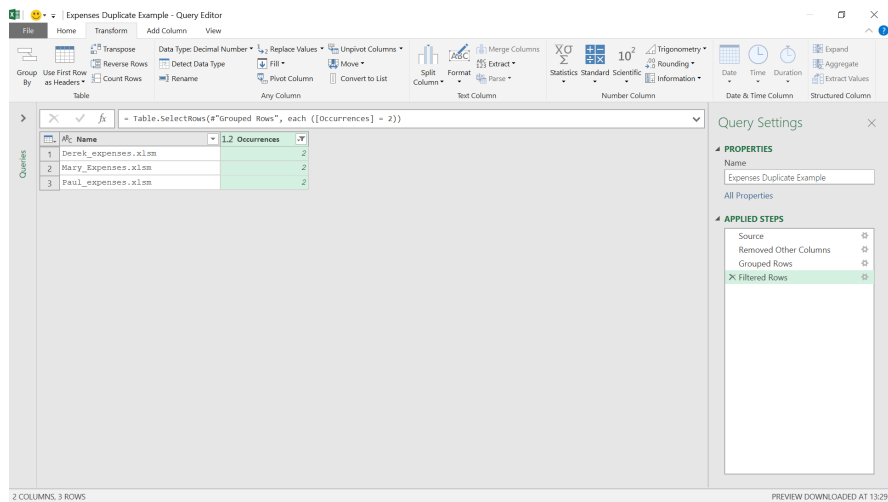


We may filter out those names that only appear once by applying a filter to **Occurrences**.



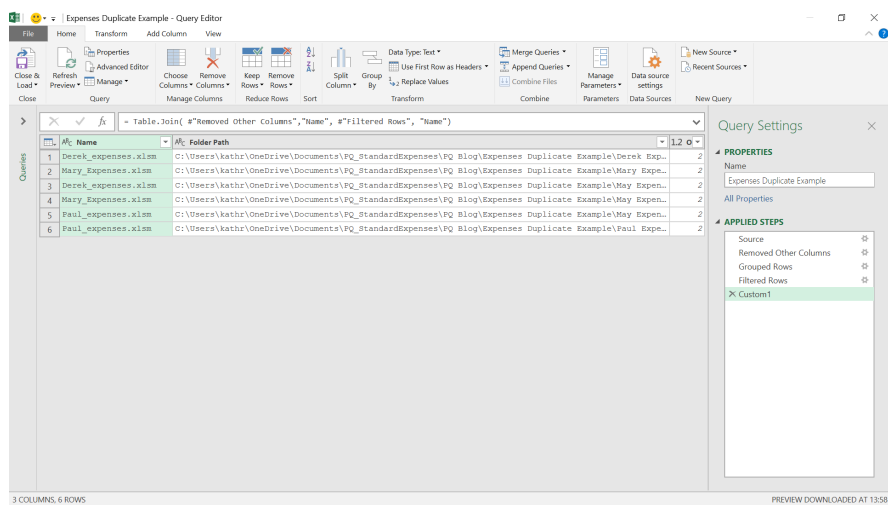


This then leaves us with the duplicates.



Thus, we have the names of the files that are duplicated, but now we want to go back and look at the folder path. We want to merge the data we had in step 'Removed Other Columns' (see now why we had to remember this step?) with the data we have in step 'Filtered Rows'. This is where we may use **Table.Join**.

Our first table is at the step 'Removed Other Columns' and our second table is at 'Filtered Rows'.



Wenow have the file name, folder path and the number of occurrences – so we can see where every duplicate file is. The syntax we used is:

**= Table.Join( #"Removed Other Columns", "Name", #"Filtered Rows", "Name")**

The reason for the #’s is because we have names with gaps, and the # simply tells **M** to ignore the spaces in the names. We could also rename our steps, but I prefer to do it this way, so I can see exactly where we are combining our data. “**Name**” tells Power Query how to join the two tables – if the name matches, then join the rows. We are using the default join here: the inner join.

Now we know where the duplicate files are, we may sort out the administration.

More next month!

## Power BI Updates

The latest set of updates finally sees the General Availability of the new Format Pane, plus error bars and information protection updates, table navigation improvements, Connect to datamarts and Power BI Data Storytelling features. But, as always (and of course!), there’s more. The full list is as follows:

### Reporting

- New Format Pane now Generally Available
- Updates to error bars in Preview
- Information protection updates
- Table navigation improvements
- Canvas Zoom in Teams and Quick create

### Modelling

- Composite models on SQL Server Analysis Services

### Data connectivity and preparation

- Connect to Datamarts in Preview
- Anaplan (Connector Update)
- Azure Databricks (Connector Update)
- Databricks (New Connector)
- Dremio Cloud (Connector Update)
- FactSet Analytics (Connector Update)
- Hexagon PPM Smart API (Connector Update)
- SingleStore (New Connector)
- Snowflake (Connector Update)
- Vessel Insight (Connector Update)
- Workplace Analytics / Viva Insights (Connector Update)

### Service

- Power BI Data Storytelling
- Announcing the Data Hub
- Power BI Metrics updates

Let’s now go through each in turn.

### New Format Pane now Generally Available

The new Format pane is finally Generally Available. This has also been shipped to the Power BI Service. The improvements have continued though:

- a blocking bug has been fixed to ensure the Analytics pane works for custom visuals (where applicable)
- another bug has been fixed that removed log scale applied to the horizontal axis. This affected line, area, stacked area, combo, or scatter charts set prior to April 2022 reports. The issue has now been resolved
- when changing a blank page to ToolTip size, Power BI now automatically flips the “allow use as ToolTip” toggle on.

### Updates to error bars in Preview

This update also revamps the Error bars card in the Analytics pane to simplify the structure of the formatting options. With fewer sub-cards to have to scroll through, you can more easily find and customise the

### Embedded Analytics

- Hierarchy slicers support in JS APIs
- Support embedding Paginated Report with data sources containing single sign on (SSO)
- Support passing report parameters for Paginated Reports in JS SDK
- DirectQuery for Power BI datasets is now supported in Power BI Embedded
- Service principal profile

### Developers

- Define the privileged permissions your visual is requesting in the capabilities.json
- Improving the custom visual load time

### Visualisations

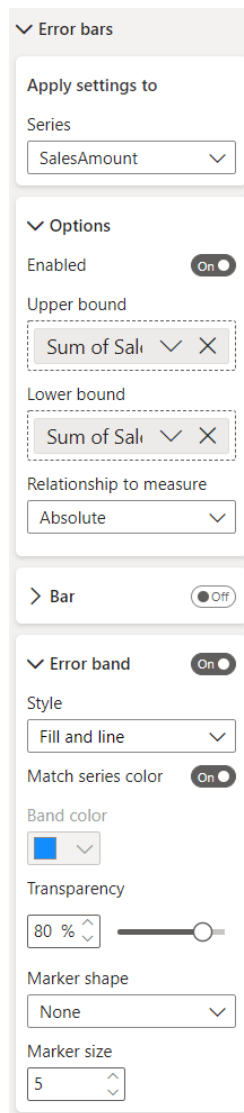
- New visuals in AppSource
- Milestone Trend Analysis Chart by Nova Silva
- Inforiver Premium Edition now certified
- Zebra BI Cards 1.2

### Other

- Coming soon: Dataverse Display Name Support
- Power BI Desktop infrastructure update (WebView2)
- Improved support for single sign-on (SSO) for all users.

various elements you want to see on your error bars. The major changes

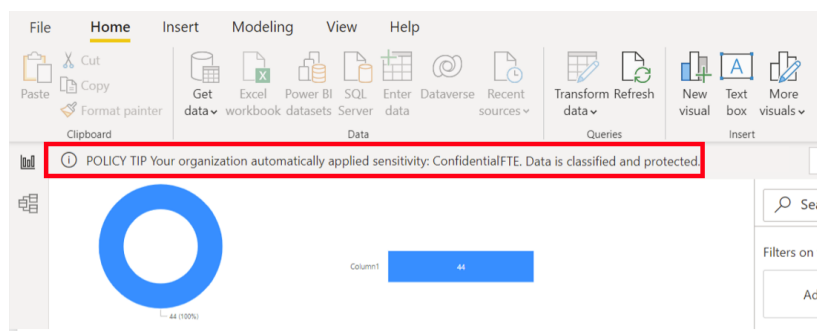
are that line options have now been moved into the Error band sub card, and marker options have also now moved into the bar and band cards.



### Information protection updates

In Power BI Desktop, you can establish a live connection to a shared dataset in the Power BI Service and create many different reports from the same dataset. If the dataset has a sensitivity label, Power BI will

automatically apply the live dataset's sensitivity label to the PBIX file in order to maintain the data's classification and protection as it leaves the Power BI Service.



### Table navigation improvements

There are also several enhancements to the Table visual for existing and new reports to improve performance, usability and accessibility for all users. These updates include:

- overall performance improvements for scrolling and rendering, as well as the look and feel of grid lines and selection styles
- new keyboard shortcuts for easier navigation, including **SHIFT + Up / Down Arrow** to select multiple contiguous rows, **Page Up / Down** for scrolling, and **HOME / END** to move focus to the first / last cell in the current row
- enhanced row selection experience that includes a left indicator and improved contrast to help distinguish selected rows from non-selected rows

- a new two-toned focus outline so on any colour background, the focus will always be visible even on darker themes meeting colour contrast requirements
- updated conditional formatting icons that are conformant with colour contrast requirements and now announces alternative text with screen readers
- the ability to associate images with alternative (*alt*) text in the Formatting pane to describe images to blind and low vision users.

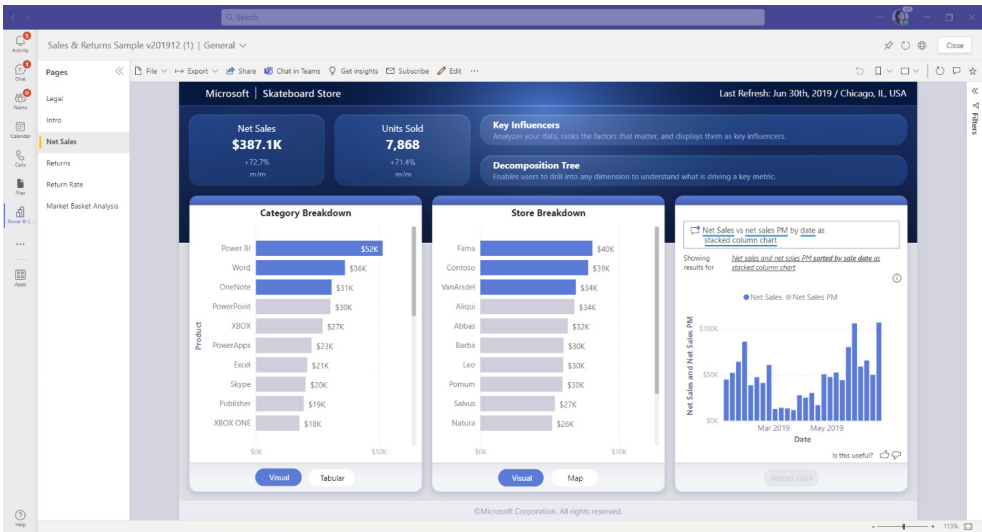
Region	Country	Item Type	Total Cost	Total Profit	Total Revenue	Unit Cost	Units Sold	Unit Price
Europe	Albania	Household	18,792,985.84	6,197,639.08	24,990,624.92	2,010.16	37396	2,673.08
Europe	Estonia	Household	17,409,995.76	5,741,550.12	23,151,545.88	2,010.16	34644	2,673.08
Sub-Saharan Africa	Kenya	Cosmetics	7,130,976.40	4,708,399.60	11,839,376.00	1,053.32	27080	1,748.80
Europe	Ireland	Household	13,645,971.16	4,500,232.42	18,146,203.58	2,010.16	27154	2,673.08
Europe	Spain	Household	13,236,903.60	4,365,328.20	17,602,231.80	1,507.62	26340	2,004.81
Europe	Malta	Cosmetics	6,338,616.43	4,185,224.77	10,523,841.20	789.99	24071	1,311.60
Europe	Moldova	Household	12,618,779.40	4,161,480.30	16,780,259.70	2,010.16	25110	2,673.08
Sub-Saharan Africa	Madagascar	Household	11,812,705.24	3,895,649.38	15,708,354.62	2,010.16	23506	2,673.08
Middle East and North Africa	Egypt	Office Supplies	16,065,350.88	3,863,628.75	19,928,979.63	2,624.80	30603	3,256.05
Sub-Saharan Africa	Seychelles	Office Supplies	15,922,036.80	3,829,162.50	19,751,199.30	2,099.84	30330	2,604.84
Sub-Saharan Africa	Tanzania	Cosmetics	5,724,794.20	3,779,933.80	9,504,728.00	789.99	21740	1,311.60
Middle East and North Africa	Jordan	Household	11,263,429.02	3,714,506.49	14,977,935.51	1,507.62	22413	2,004.81
Sub-Saharan Africa	Equatorial Guinea	Cosmetics	5,622,095.50	3,712,124.50	9,334,220.00	789.99	21350	1,311.60
North America	Greenland	Household	10,554,847.62	3,480,827.19	14,035,674.81	1,507.62	21003	2,004.81
Middle East and North Africa	Oman	Cosmetics	5,231,577.11	3,454,275.29	8,685,852.40	789.99	19867	1,311.60

### Canvas Zoom in Teams and Quick create

Last month’s update shipped the canvas zoom feature to both Power BI Desktop and Service. This feature is now also available for Power BI in Teams and Quick create. For report readers, this is especially important for improving readability. For report creators, this helps magnify the canvas to make pixel perfect tweaks. You may drag the slider to set the zoom level or

click on the zoom % to launch the zoom level dialog and type in a custom input. Furthermore, you may use the new quick ‘fit to page’ button to get back to the default view.

One thing to note though: the zoom level is not saved with the report.



### Composite models on SQL Server Analysis Services

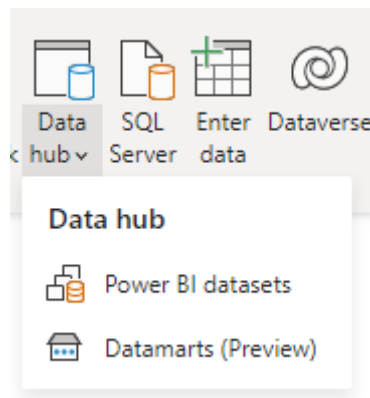
This update also allows you to create composite Power BI models by connecting to SQL Server 2022 Analysis Services tabular models. SQL Server 2022 is available now for public Preview. When you upgrade or deploy tabular models to SQL Server 2022 Analysis Services, this DirectQuery

connection becomes available in Power BI Desktop beginning with the May 2022 release. Existing or new tabular models hosted on SQL Server 2022 Analysis Services can be joined with other data in Power BI Desktop to create a composite model.

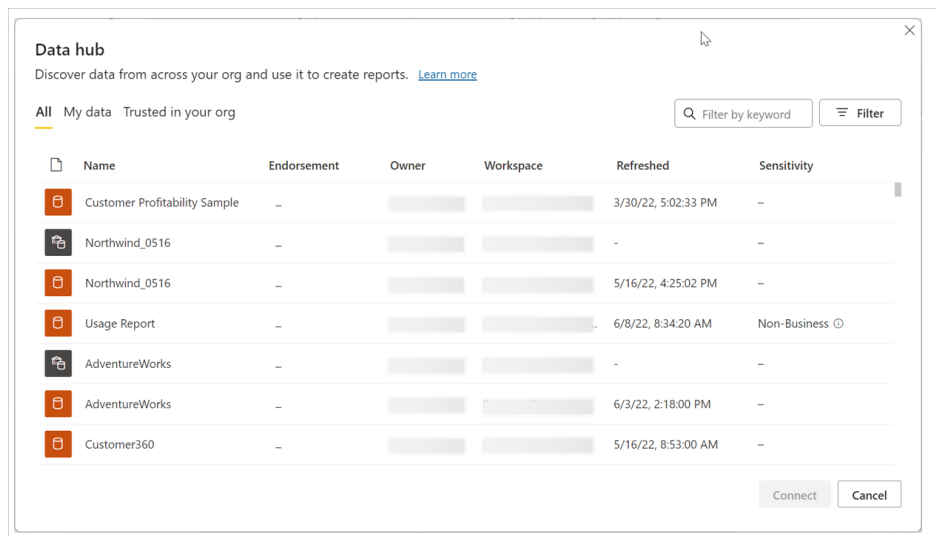
### Connect to Datamarts in Preview

As a reminder, Datamarts in Power BI was officially announced in May 2022 as a new Power BI Premium self-service capability that enables you to perform relational database analytics and uncover actionable insights from their data.

You can now easily discover and connect to Datamarts from the ‘Data hub’ within the Power BI Desktop Ribbon. This feature allows users to discover datamarts that have already been created and reuse them to easily build reports.



You will be able to see datamarts that you have created or were shared with you by others in the organisation. This will connect you to the datamart's underlying auto-generated dataset (using live connect) so that you can easily create reports.



To create datasets on top of the datamart, you can use the Azure SQL database connector and specify the Datamart's SQL connection string for Server name. If you created the datamart, you can get the SQL Connection string from datamart settings (context menu action from the datamart in your workspace):

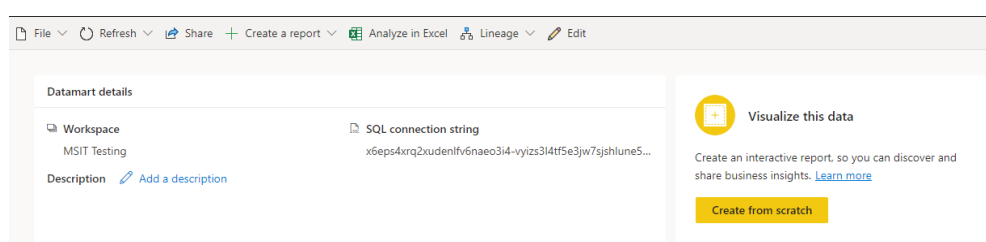
## Settings for Customer360

- Datamart Description
- ◄ Server settings

### Connection string:

x6eps4xrq2xudenlfv6naeo3i4-vyizs3l4tf5e3jw7sjshlun... [Copy](#)

If this datamart was shared with you, you can get the SQL connection string from the Datamart info page below from the Data Hub (in Power BI Service) (see more on the Data Hub below).





Apparently, Microsoft has state that the plan is to next simplify the experience to connect to the datamart's SQL endpoint so that advanced use cases may be served easily without the hassle of having to find the connection string.

### ***Anaplan (Connector Update)***

The Anaplan connector has been updated. This version of Anaplan connector for Power BI now supports Anaplan OAuth APIs for authenticating with Anaplan. Users will be able to use either Basic Authentication (User ID, Password) or the Anaplan configured IDP for

logging into Anaplan from the connector, accessing and running Export actions. With this new connector option, customers do not need to be an Anaplan exception user to use Power BI.

### ***Azure Databricks (Connector Update)***

The Azure Databricks connector has been updated:

- fast evaluation option is removed from the User Interface (UI) and is used by default now
- various performance improvements have also been made.

### ***Databricks (New Connector)***

A new Databricks connector has also been released. This update contains a Preview generic datasource 'Databricks (Beta)', which supports OAuth authentication on AWS. The OAuth on AWS Preview will support users managed by Databricks directly.

### ***Dremio Cloud (Connector Update)***

The Dremio Cloud connector has been updated:

- users may now connect Power BI Service to Dremio without needing Power BI Gateway
- some customers may experience a back compatibility issue with their existing connections and reports. They may need to re-enter their credentials and possibly recreate existing reports once they have upgraded to this new connector.

### ***FactSet Analytics (Connector Update)***

Minor internal fixes for table representation have been made.

### ***Hexagon PPM Smart API (Connector Update)***

This update contains a fix resolving an issue where Okta authentication no longer worked with the WebView2 browser control.

### ***SingleStore (New Connector)***

There is a new SingleStore Power BI connector. SingleStore is a fast database for all data-intensive applications. It provides a single unified database for analytical and transactional workloads with multi-model support for text, geospatial, time-series and more. The SingleStore

Power BI connector provides a live connection to the database with support for DirectQuery as well as the ability to import data into Power BI with standard and Custom SQL Import modes.

### ***Snowflake (Connector Update)***

This update contains a minor bug fix around mixed case database names and quotations causing compilation errors.

### ***Vessel Insight (Connector Update)***

The new update in the connector enables a user to navigate through fleets with different asset trees when applicable and only sensors with data will be visible. Vessel Insight has a new version of the asset tree where the hierarchy structure has been updated. It is now possible to navigate through the old and the new asset tree or more, while

the previous version of the connector only allows the user to navigate through one asset structure. Sensors without data will be hidden and ensure the user when selecting sensors for import that there will be no empty sensors.

### ***Workplace Analytics / Viva Insights (Connector Update)***

The Workplace Analytics connector has been updated and is now rebranded as Viva Insights. The updated connector is now Generally

Available. The connector includes a new optional parameter, which lets you import raw query output from certain data sources.

## Power BI Data Storytelling

Back in May, Microsoft announced the public Preview of Power BI Data Storytelling, which represents a new integration between Power BI and PowerPoint.

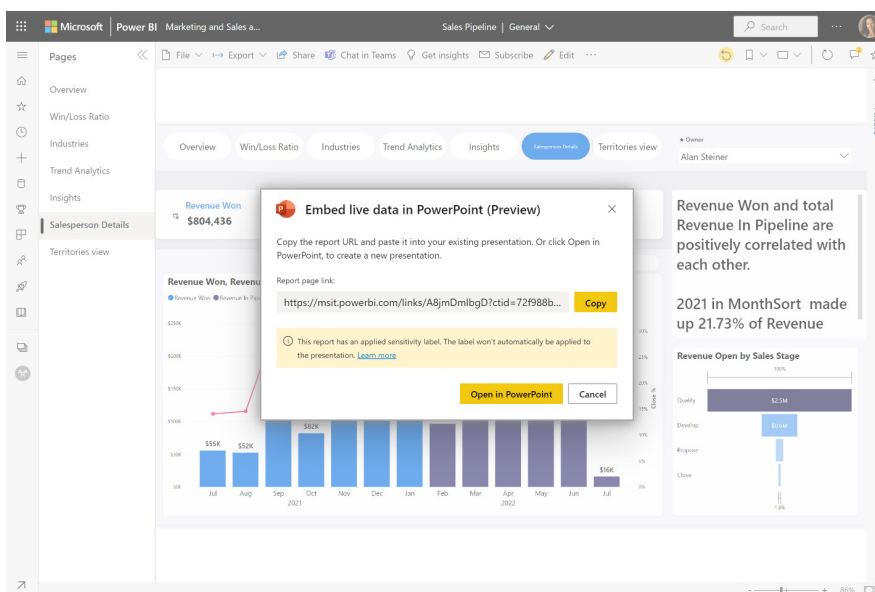
Power BI Data storytelling is a PowerPoint add-in developed by the Power BI team, which lets you add Power BI reports into PowerPoint slides and utilise the interactive experience of Power BI inside your presentations. With data storytelling, data will always be up to date in your slides, whether you're building a presentation or presenting it live to others.

This update enables Storytelling starting points inside the Power BI report experience, making it easier and simpler for you to get started with storytelling directly from the report you want to use in your presentation.

First, open the report you wish to add to your presentation, navigate to the desired page, and make any data selection that reflects the view you want to use. Then, click **Share -> PowerPoint or Export -> PowerPoint -> Embed live data**.

Check the 'Embed report with data filters you selected' box if you want to make sure that the same report state will be added to the presentation. This will open a new window, with two options:

1. copy the link and paste into a Power BI add-in you added to an existing presentation
2. open a new presentation with the report automatically added to it and continue to build your presentation.



## Announcing the Data Hub

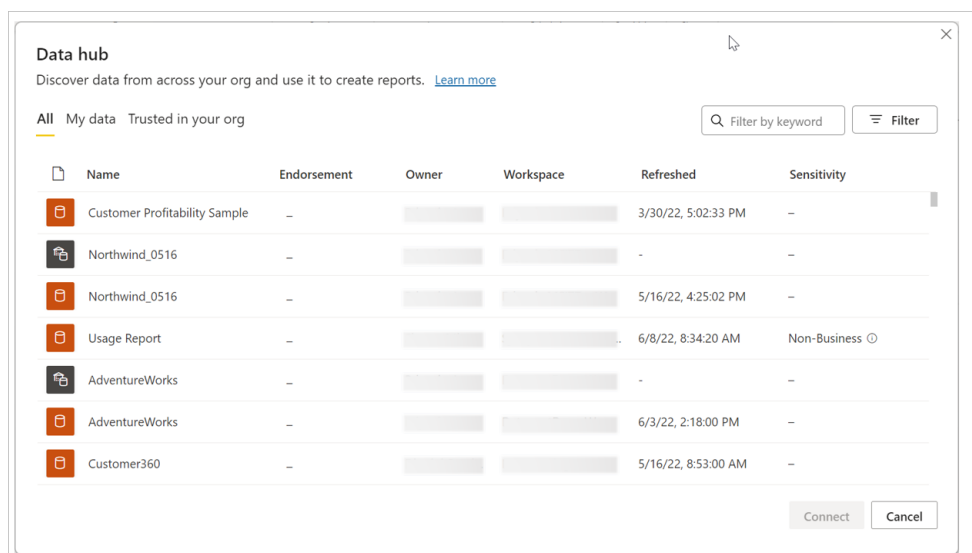
The Data Hub is the evolution of the Datasets Hub that was announced in December 2020. In the Data Hub, in addition to Power BI datasets, you can now find the newly released Datamarts.

The Data Hub is a central location for data owners, report creators and report consumers to manage, discover and reuse data items across all workspaces:

- data item owners can see usage metrics, refresh status, related reports, lineage and impact analysis to help monitor and manage their data items

- report creators can use the Hub to find suitable items to build their reports on and use links to easily create the reports
- report consumers can use the Hub to find reports based on trustworthy data items.

This was in response to feedback Microsoft received from many customers that users often build similar data repeatedly, resulting in resource overload, governance complexity, and difficulty in identifying sources of truth.



The Data Hub addresses these challenges by making it easier for users to discover existing, quality data and reuse it to answer business questions and gain insights. Once you discover a data item that seems to have the right data, a click on the item opens the data details page. On the details page you can see the following:

- metadata about the data item, including description, endorsement and sensitivity label
- actions such as share, refresh, create new, etc.
- related reports, which are reports that were built on top of the dataset.

Considering data discovery best practices, before creating new reports, you should look at the existing related reports. If there are no such reports that are suitable, there are multiple ways to create a new report:

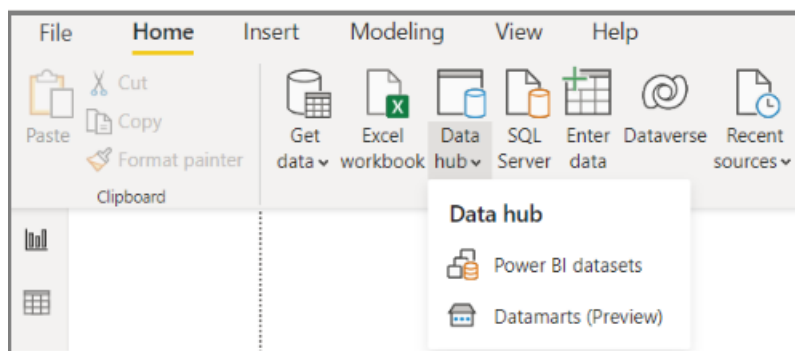
- create a report from a template, if a template has been defined (a template is defined by adding the suffix '(template)' to a report built on top of the dataset / datamart)

- create a report from scratch, to open a blank canvas and start creating your report
- auto-create a report. It should be noted that this is currently supported only for single-table datasets; moving forward, support for datasets with multiple tables will be added in due course
- create a formatted table.

Power BI report creators primarily use Power BI Desktop to build new content and then publish it to the Power BI Service. When building new reports, instead of connecting to external data sources, the best practice is to connect to curated data available to them in Power BI.

In Power BI Desktop, just as in the Power BI Service, Microsoft would like to bring all Power BI data items into a single experience. For this reason, they have brought the Data Hub into Power BI Desktop.

With the Data Hub, users may find datasets and datamarts side by side, filter and search to find the data they need, and then connect to create a report.



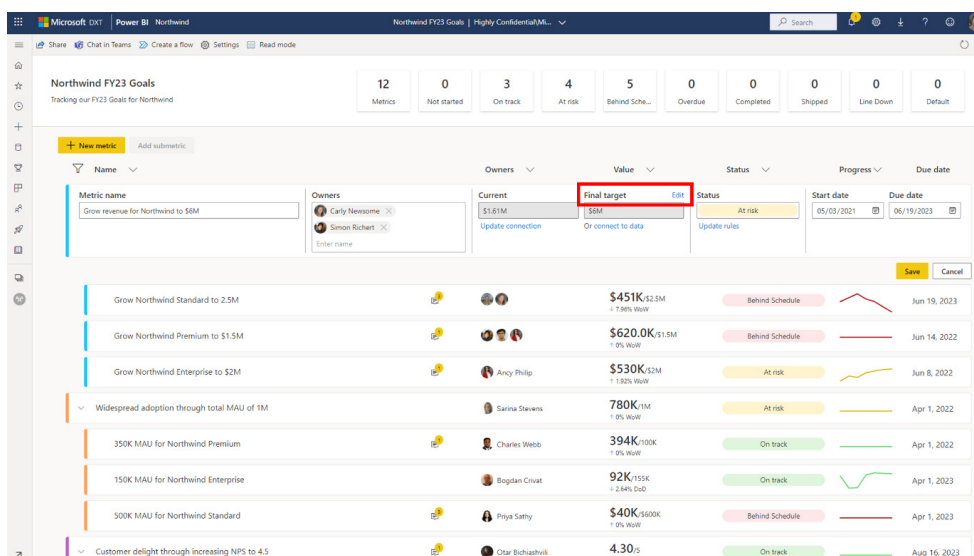
## Power BI Metrics updates

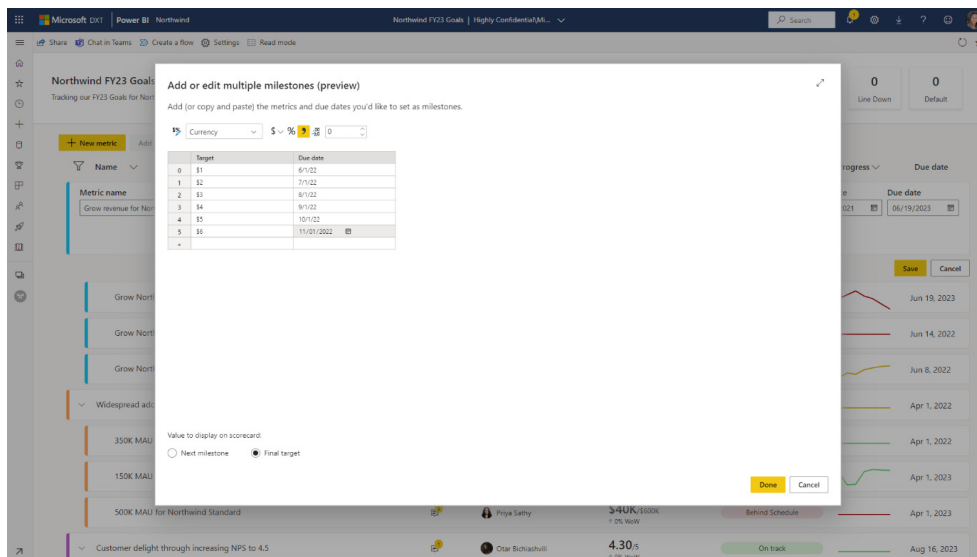
There are several Power BI Metrics (the feature formerly known as 'Goals') updates this month. There are now additions of multiple milestones that allow you to add date associated milestones within a

given metric, categorical metrics that help you track non-numerical items, and enrichment of the bulk updating feature. Further, Power BI Metrics now supports customer managed keys.

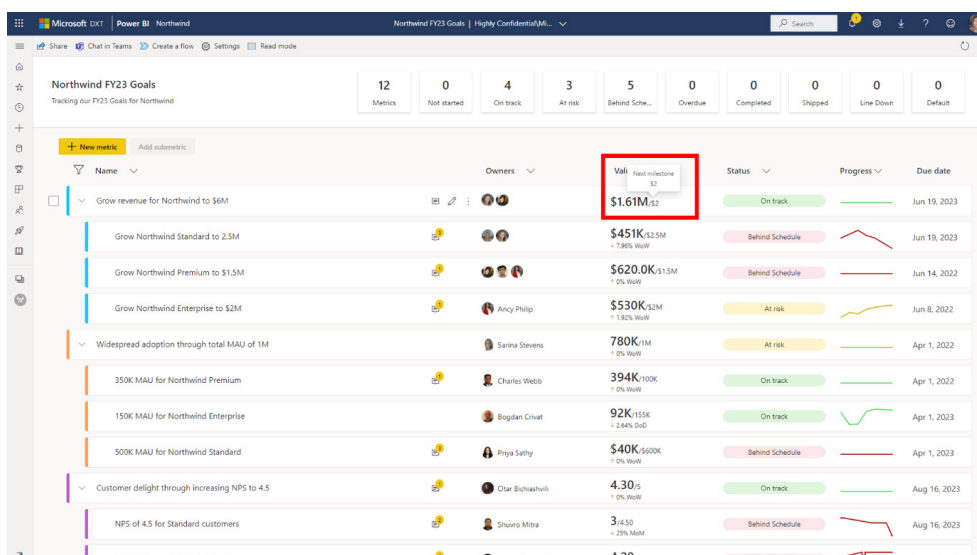
## Milestones

Now, date associated milestones can be added to a metric to help track incremental progress toward an overall end goal. With this new feature, scorecards support more nuanced tracking scenarios such as cumulative monthly targets, stepped revenue goals, and more.

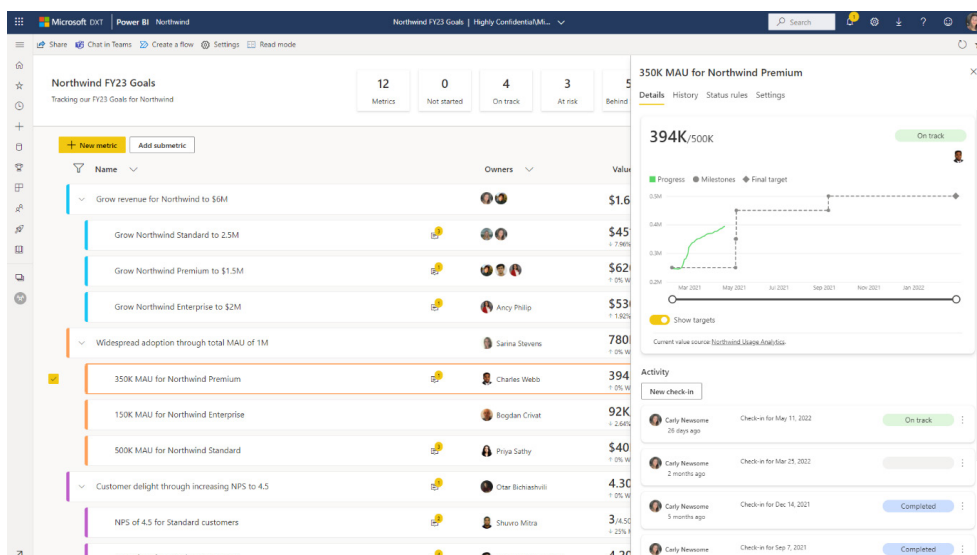




Each milestone you set is associated with a date so that you can accurately track your progress in your scorecard the way you do in the rest of your organisation. In the setup process, you will have the flexibility to select whether you want to display the next milestone or the final target for your metric on the scorecard. This may be customised per metric.

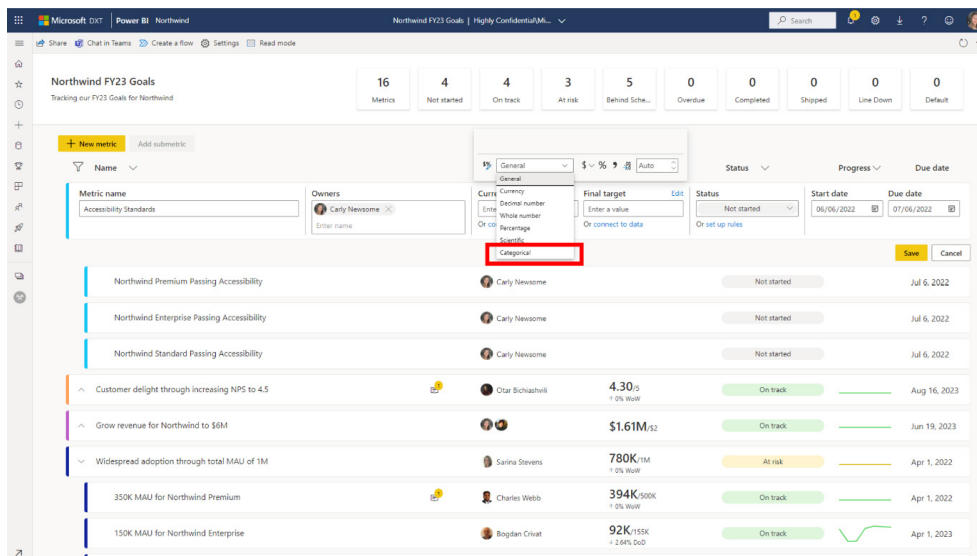


You can see the milestones as well as the final target visualised in the Details pane alongside your current progress, so that it's easier than ever to get a snapshot of how you're performing against your metrics.

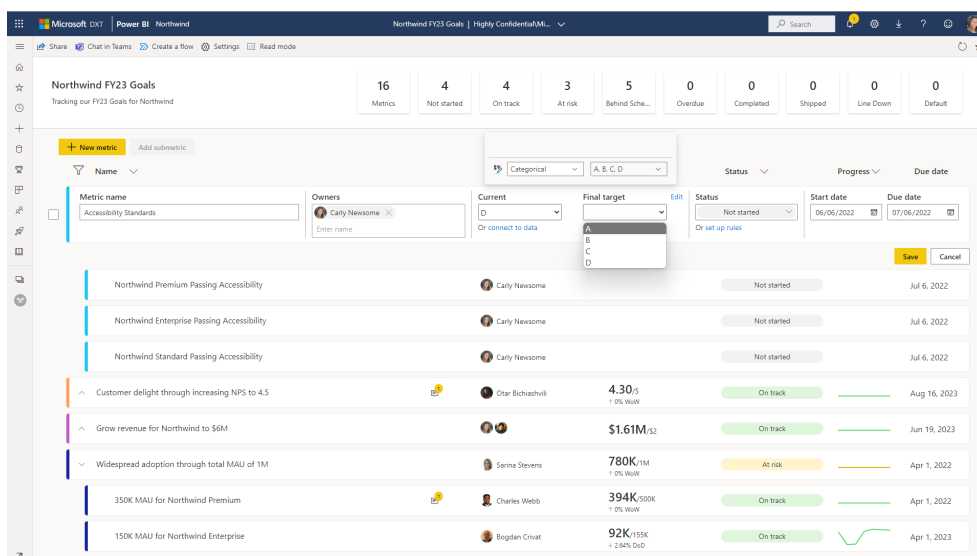
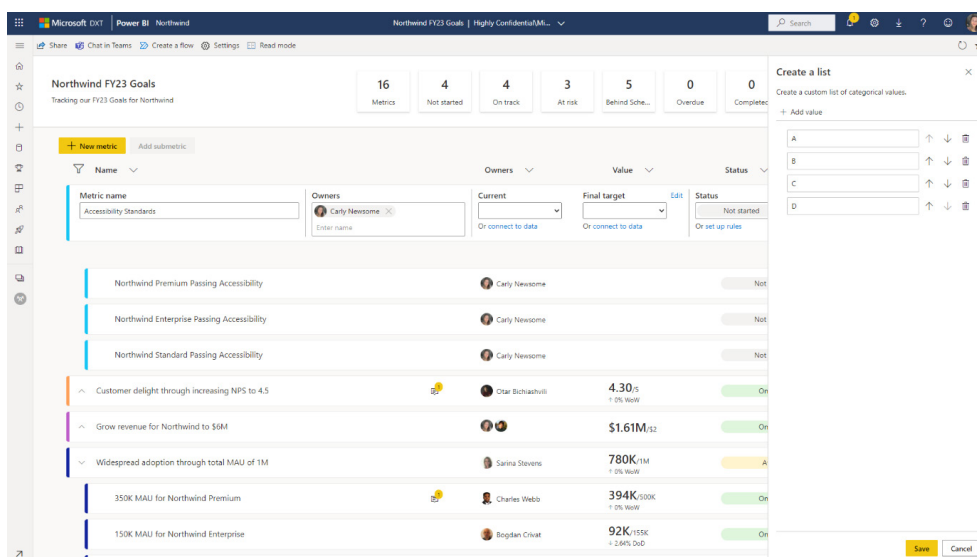


## Categorical Metrics

Now you may begin to track metrics that are not numerical. With categorical metrics, scorecards support scenarios like accessibility, NPS category scores (detractor, promotor, *etc.*). To get started, select categorical from the floating formatting bar when choosing the current and target values.

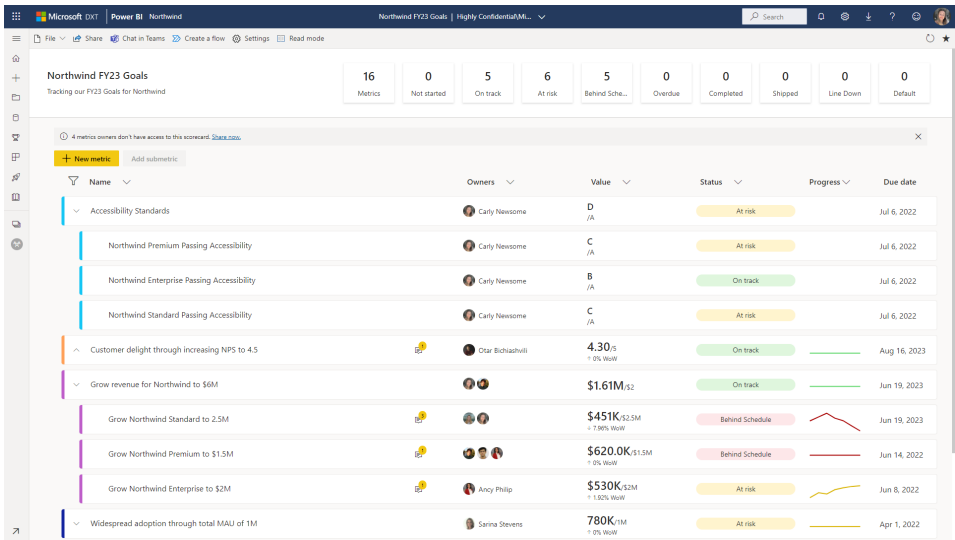


You will be able to select from existing lists of categories or create new categories with accessibility scores. After creating my categories, you may then fill out my current and target values with the appropriate selections.

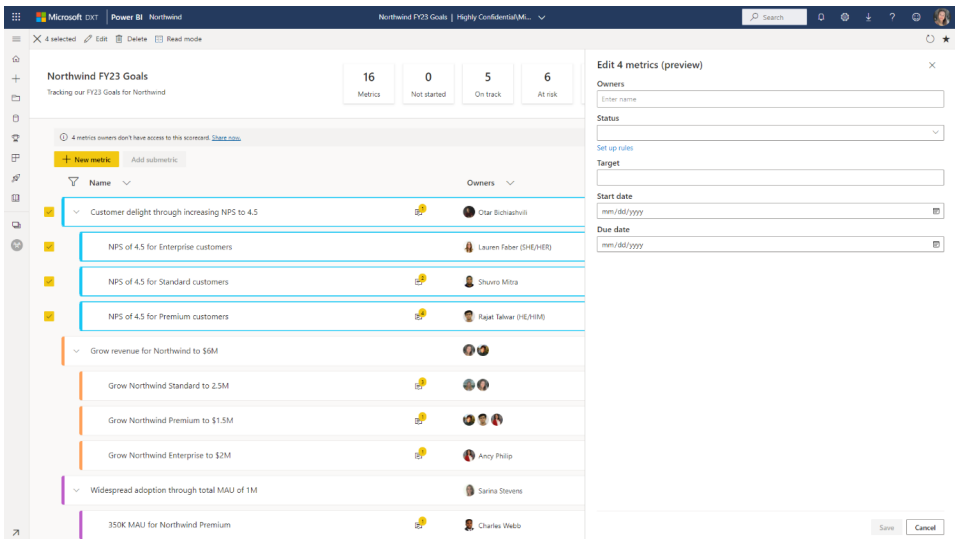




Here, you can see a group of metrics with accessibility scores B-D, with final targets of A.



Bulk Updating



With new additions to bulk updating, you will be able to:

- assign owners
- update statuses (if no rules exist)
- create new or replace existing status rules
- update start dates
- update due dates.

Hierarchy slicers support in JS APIs

Slicers in Power BI are a type of on canvas visual filters. The slicers enable a user to sort and filter a packed report and view only the information they want. Independent software vendors (ISVs) and developers may now interact with hierarchy slicers using slicer APIs in JS APIs.

Using the slicer APIs, you can get and set the state of a Power BI slicer. In addition, you can use load configuration to change the slicer state when loading a report.

Support embedding Paginated Report with data sources containing single sign on (SSO)

Partners and customers may now automate Power BI Embedded paginated reports with data sources containing SSO support when the data source has the SSO option enabled and your users access reports built atop the data source. Power BI will send the related identity

blob, which was received through the GenerateTokenV2 request in the queries, to the data source.

This option enables Power BI to respect the security settings that are configured at the data source level.

## Support passing report parameters for Paginated Reports in JS SDK

Developers may now set up parameters to paginated reports in Power BI by passing them in the Payload settings through JS SDK. They can further customise how Power BI processes the paginated report by embedding parameters for filtering data that meets the parameters criteria.

## DirectQuery for Power BI datasets is now supported in Power BI Embedded

ISVs may now enable their customers to create their own specific dataset that can be combined with a central dataset that ISV uses for all their customers. The customers can combine their own data coming from any datasource connected to PBI in any fashion (including with SSO enabled for supported data sources) and create their own measures and calculations.

For example, ISVs can let their tenant customers create tenant specific measures via their code, using TOM over XMLA to the customer tenant specific dataset.

To embed Power BI report with a dataset which has a Direct Query connection to another Power BI dataset two separate things must be done:

1. in Power BI portal, set the 'XMLA endpoint' to 'Read Only' or 'Read Write' in the dataset's capacity: this only needs to be done once per capacity
2. generate a Multi-Resource Embed Token with all relevant dataset IDs specified in the request and the XmlaPermissions set to ReadOnly.

## Service principal profile

Until now, ISVs and other Power BI Embedded application owners were able to deploy the embedded solution to production by using one of the following two [2] options:

1. create one service principal for all their customers, where they have a Power BI limit of 1,000 workspaces per each service principal
2. create one service principal per each customer, where they have the burden of managing thousands of service principals.

Now, this update has added a new method that allows a much larger number of customers per one service principal by introducing a new

entity named service principal profile. Each service principal profile can administer a customer, and the number of profiles per service principal can reach hundreds of thousands.

The service principal profile allows the ISV to build a scalable application that enables better customer data isolation and establishes tighter security boundaries between customers. Using the service principal profile enables the ISV application to host multiple customers on a single Power BI tenant. Each profile represents one customer in Power BI. In other words, each profile creates and manages datasets, reports, dashboards and other artifacts that are connected only to one specific customer's data.

## Define the privileged permissions your visual is requesting in the capabilities.json

Starting from API 4.6 and above, developers must declare any privileged API calls their visual requires access to operate. These currently include **web access** and **downloading to a file** from the custom visual.

These privileges are granted in the *capabilities.json* file and will be displayed in the 'About' dialog in the Desktop and web when in Edit

mode. The visual will be permitted to access any special privileges only if it was predefined in the *capabilities.json*. Administrators may enable custom visuals downloading files through a tenant setting.

New APIs are introduced to allow visuals to query the host at runtime to determine which of the privileges is enabled.

## Improving the custom visual load time

With API 4.6, Microsoft has improved the load time for custom visuals.

## New visuals in AppSource

The following are new visuals in this update:

- Mapbox Custom visual
- SCvation-Flow-Basic
- 2WriteBack
- Easy SPC.

## Milestone Trend Analysis Chart by Nova Silva

Milestones in project management are points in time when important deliverables are due, such as beginning and end dates of project phases, formal approvals, etc. The Milestone Trend Analysis Chart for Power BI allows you to monitor these milestones.

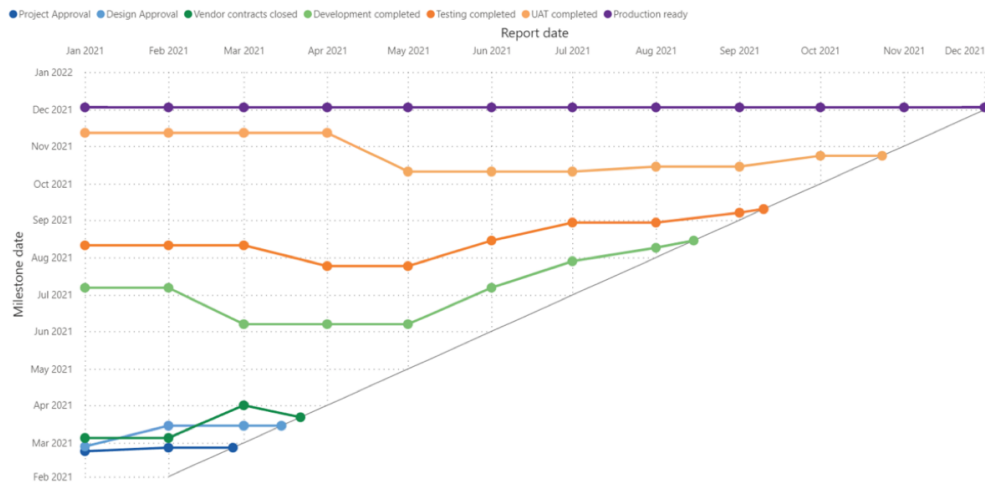
Key signals in the Milestone Trend Analysis Chart include:

- horizontal line: no change in the (expected) completion date
- increasing line: delay in the (expected) completion date
- decreasing line: (expected) completion date brought forward.

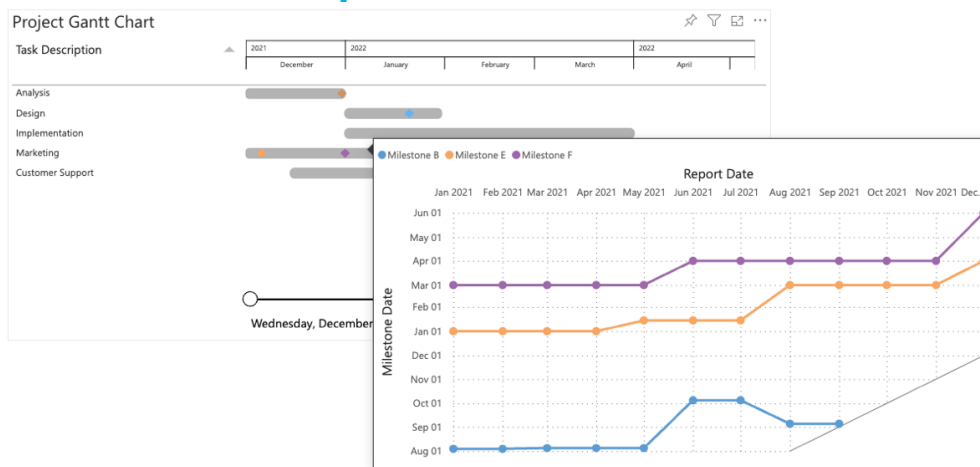
Milestone Trend Analysis (MTA) is a simple method of early identifying deadline trends. It allows you to react in time with corrective actions. It will also raise the deadline awareness of all project participants and stakeholders.

## Milestone Trend Analysis Chart: identify milestone trends

Milestone Trend Analysis



## Combined as tooltip with the PowerGantt Chart



All features of the Milestone Trend Analysis Chart are available in the familiar standard Power BI user interface. Themes, interactive selection and ToolTips are all supported.

All features are available for free to evaluate this visual within Power BI Desktop from within AppSource.

### Inforiver Premium Edition now certified

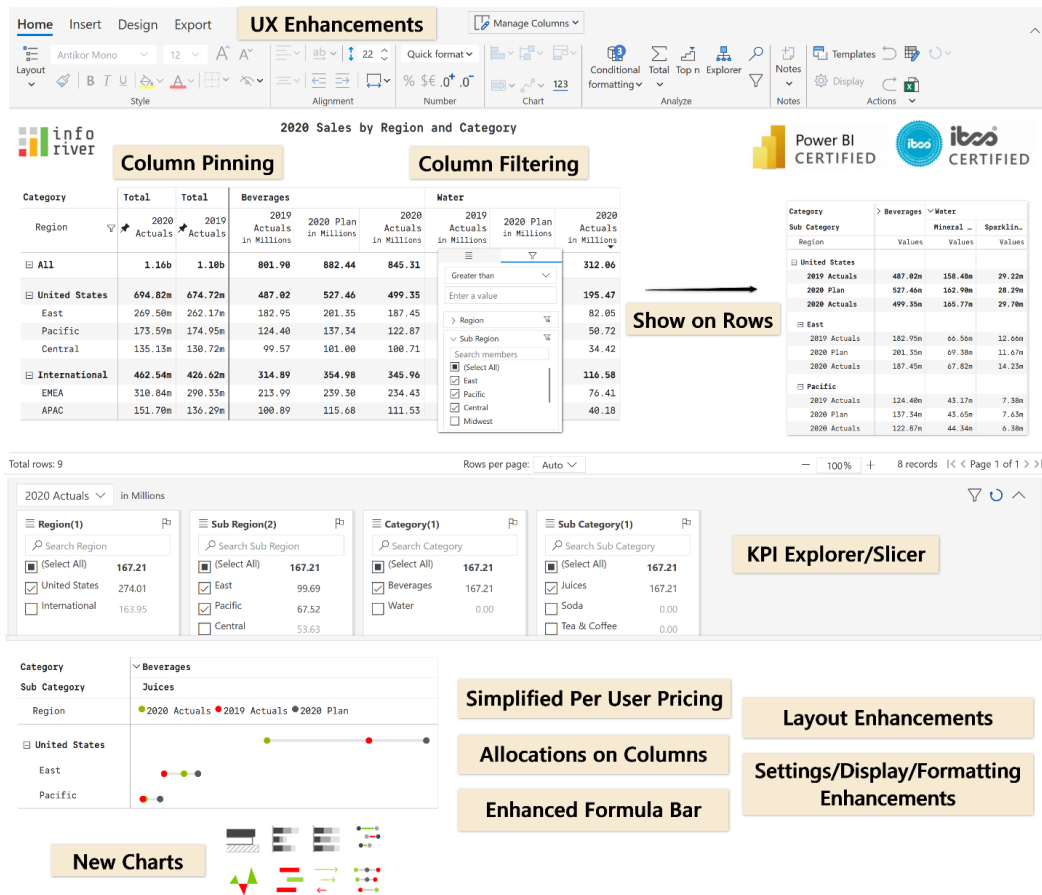
Inforiver Premium Edition, which delivers a quick way to build Power BI reports, has now been Microsoft Power BI Certified. The latest release incorporates several new features and enhancements and is available for download at Microsoft AppSource.

The Premium Edition leverages a no-code / low-code architecture to deliver a wide variety of use cases such as:

- paginated reports
- management, financial and IBCS variance reports
- graphical tables (in-cell visualisations)
- pivot analysis with expand and collapse available for both row and columns
- Excel-like formulae at row, column and cell level for visual-level calculations
- manual data entry for plan, forecast or comments / cell level annotations
- what-if? simulations with popular distributions and spread methods
- formatted export to Excel / PDF.

Additional use cases such as writeback, paginated report subscription scheduling, and collaborative cell level commenting are available (but only in the enterprise edition).

Inforiver Premium Edition is completely free for Power BI Desktop, with no feature restrictions. Apparently, Inforiver pricing has also been simplified for wider adoption.



Key enhancements in this 1.5 release include:

- KPI explorer / slicer for data exploration
- new chart types
- column pinning and freezing
- column filtering
- show measure on rows
- User Experience (UX) enhancements
- display and formatting enhancements.

## Zebra BI Cards 1.2

The 1.2 version of the Zebra BI Cards visual brings some new features, such as the ability to display the year-to-date (YTD) value for the KPIs, more flexibility with different font settings for elements on the cards, and new interaction settings.

### Year-to-date value

With Zebra BI Cards 1.2 you may now choose to display the YTD value for the KPIs. In practice, this brings several benefits:

- seamlessly switch between different stages for each KPI: YTD total, total and last data point
- compare data with the same period last year
- put your data in perspective by having all information about KPI performance on one page.

### Font settings

To create even more flexible reports, you may now apply your preferred font settings, such as font style, size, and colour. Card elements like KPI title, value, variances and chart labels may be customised separately.

### Interaction settings

By adding the Interaction group of settings, report designers now have more control over which interaction is allowed to the end-user once the report is published.



### Coming Soon: Dataverse Display Name Support

With next month's release, Microsoft has announced that they are planning to support display names through the Dataverse connector. Once released, you will see display names automatically applied to all columns in the model when creating reports using the Dataverse connector, in either Import or Direct Query mode.

This should save Dataverse users a lot of time, as you won't need to manually name every column to match the friendly names end users are accustomed to seeing in their applications, like Dynamics. However, Microsoft did want to give everyone advanced warning of this change, as there will be some impact on existing reports once the change rolls out.

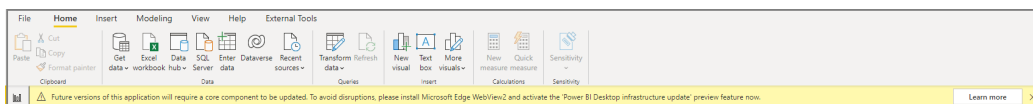
For all existing reports, you'll need to open the report up in Power BI Desktop and refresh the model for display names to show up. On the first schema refresh, the display names will override any existing renames you've made in the report, so if you want to retain those unique names, you'll need to manually reapply your name changes in the model. This will only happen when the display names are first applied, so future schema refreshes should not require any rework.

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.

### Power BI Desktop infrastructure update (WebView2)

The switch to WebView2 as part of the infrastructure update continues. As Microsoft is getting closer to General Availability, they are hoping to have solved as many issues as possible. Starting with this release, you

will see the following warning every time you launch Power BI Desktop if you don't have WebView2 installed yet:



If you see this, you should install WebView2 and enable it.

### Improved support for single sign-on (SSO) for all users

Power BI Desktop will now prompt users to sign in and use SSO where applicable to better align with Microsoft Office. This is the first of many small tweaks the start-up experience. In future updates, Microsoft will use signed in users to deliver more tailored experiences and allow users to consume files from the Service.

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

## New Features for Excel

Similar to the Power BI Updates, the Excel gang has commenced summarising recent updates in everyone's favourite spreadsheeting software. We've no idea whether this will continue, but we thought we'd make hay whilst the sun shines and provide an overview of the features divulged across Excel for the web, Windows, Mac and mobile.

In this first update (of this kind anyway!), Microsoft has shared the General Availability of Power Query refresh for selected data sources

and the ability to change the source file for workbook links in Excel for the web and performance improvements across Excel for Windows and Excel for Mac making AutoFilter noticeably faster, for example. For Insiders, updates see an improved Recommended PivotTable experience (already noted earlier in this newsletter) in Excel for Windows and the ability to get and shape data with the Power Query Editor in Excel for Mac.



In summary:

### Excel for the web

- Power Query refresh for selected data sources
- Changing source file for workbook links

### Excel for Windows

- Improved recommended PivotTable experience (Insider Beta)
- Faster recalculation on resource constrained devices (Current Channel)
- Faster AutoFilter (Monthly Enterprise Channel)
- Dataflow and Dataverse connectors (Monthly Enterprise Channel and Current Channel)
- Running Office Scripts from a workbook button (Current Channel)

### Excel for Mac

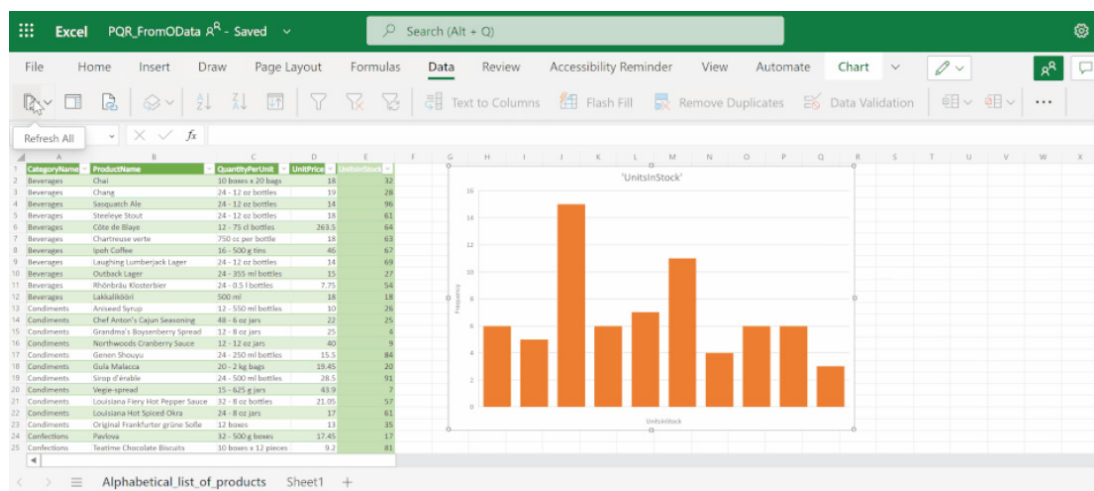
- Shaping data with Power Query Editor (Insider Beta)
- Faster AutoFilter
- Improved Find dialog and Find All.

Let's plough through.

### Power Query refresh for selected data sources

In Excel for the web, Power Query refresh is now available for queries sourcing data from the current workbook and anonymous OData feeds. Furthermore, it should be noted that Queries may refresh:

- in the background, so you may continue working on the spreadsheet
- asynchronously, instead of one-by-one, enabling a faster refresh.



As stated above, to refresh the Power Query queries in your workbook situated in Excel for the web, the source data must either emanate from the current workbook (i.e. 'From Table / Range') or else anonymous OData feeds. In addition, you may also start with a Blank Query, write your own M code and then refresh it.

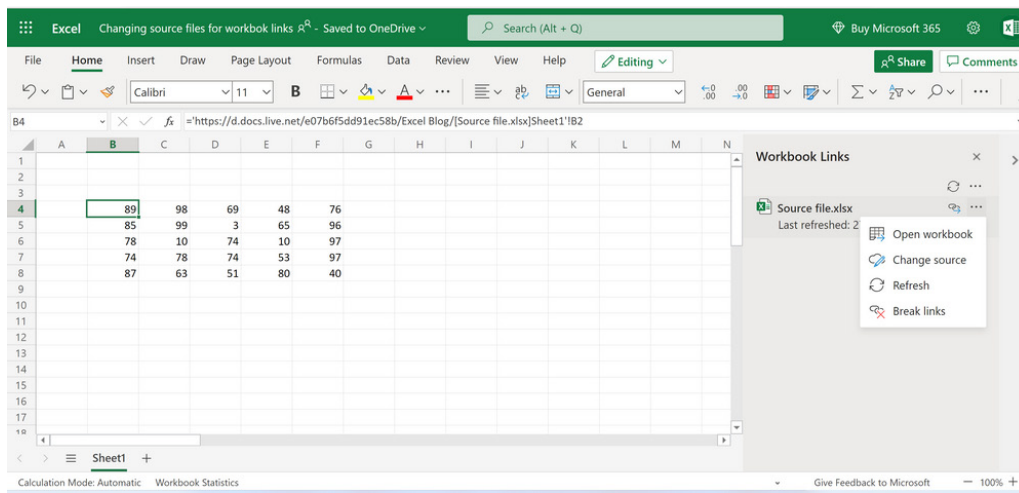
The refresh happens behind the scenes so you may keep editing the workbook while refreshing. It is done in a usual way, e.g.

- Select the **Data tab -> Refresh All**

Obviously, Microsoft is continuing to work on enabling refresh for additional data sources as well as with other features. Future plans include Power Query Refresh for additional data sources, automation support, and eventually, releasing the full Power Query Editor experience to Excel for the web.

### Changing source file for workbook links

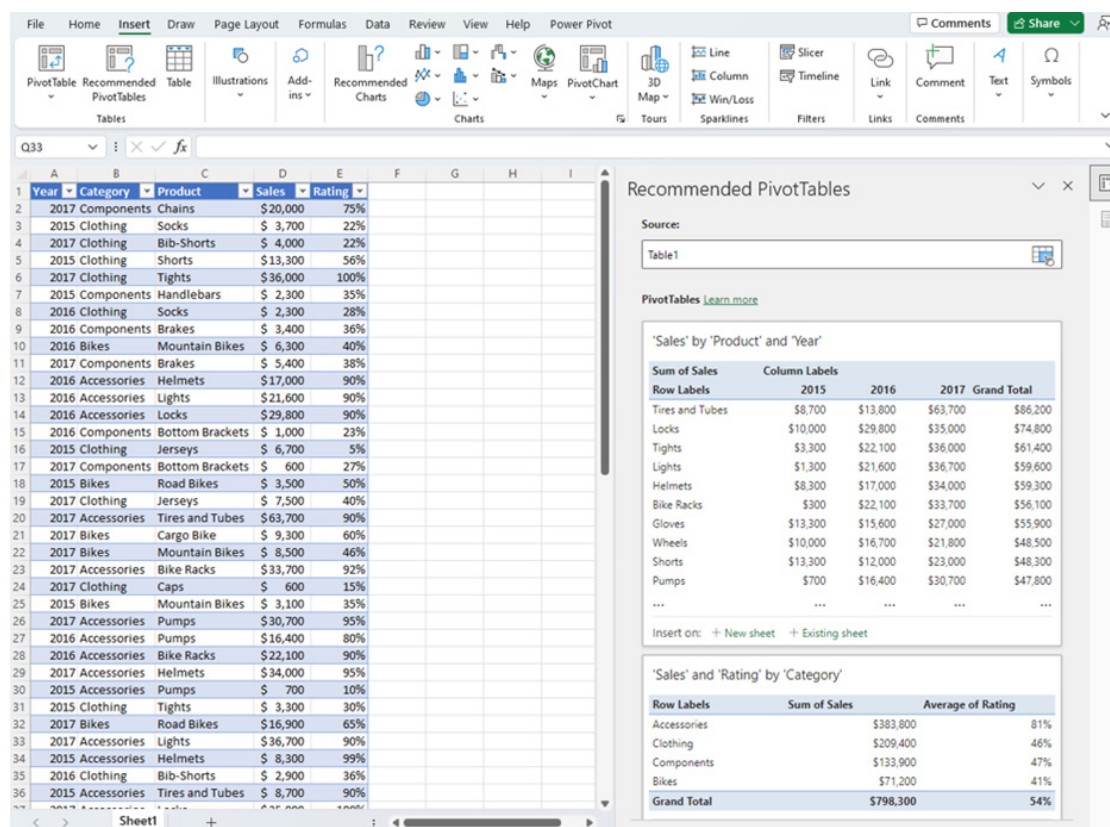
In addition to opening, refreshing and breaking links, you may also now change the source file in workbook links in Excel for the web.



### Improved recommended PivotTable experience (Insider Beta)

As mentioned earlier in this newsletter, Recommended PivotTables in Excel for Windows are now more intelligent, utilising the same technology as the Analyze Data feature. A redesigned panel should

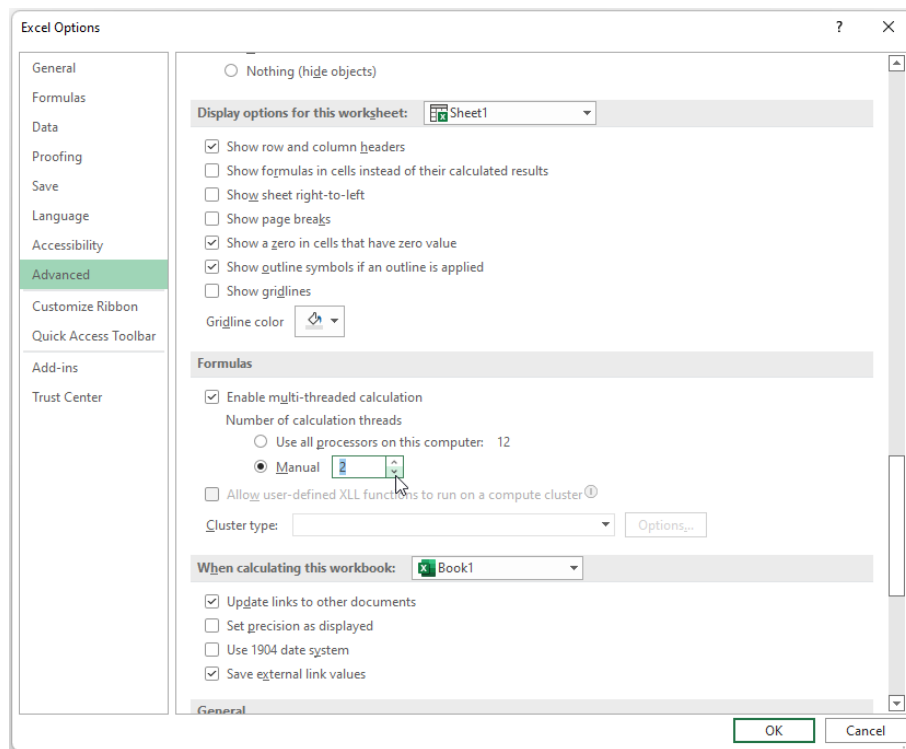
make it simpler to view all of your options, and easier to change your data selection before inserting a recommended PivotTable.



### Faster recalculation on resource constrained devices (Current Channel)

On resource constrained devices, Excel for Windows has now made recalculation more optimal by default, by running calculations on a single thread. In most cases, users should see noticeably faster calculation on these devices. It should be noted that in some cases that require

computationally intensive calculations, users may want to consider overriding this default by setting the 'Number of calculation threads' to two [2] in **File -> Options -> Advanced -> Formulas**, viz.



### ***Faster AutoFilter (Monthly Enterprise Channel)***

Users should now find AutoFilter noticeably faster in Excel for Windows too. Improvements have been made to reduce memory usage and optimise calls made by the filter's comparison algorithm. The

optimisations should be especially noticeable on low-end devices that have less memory or slower CPU-memory throughput.

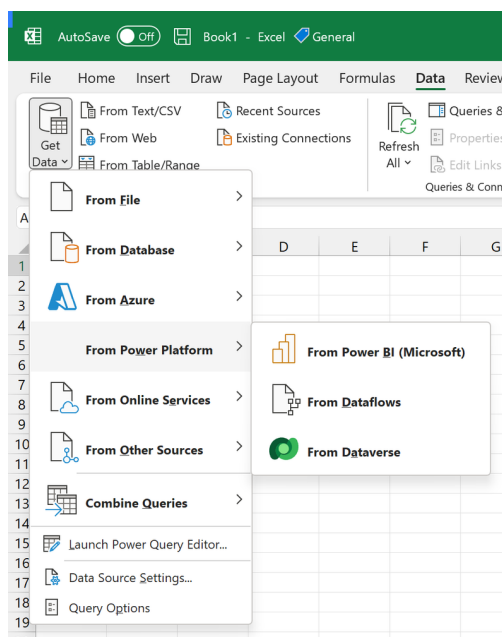
### ***Dataflow and Dataverse connectors (Monthly Enterprise Channel and Current Channel)***

Dataflows are a self-service, cloud-based, data preparation technology. They enable customers to ingest, transform and load data, and are featured in multiple Microsoft products. With the new 'From Dataflows' connector in Excel for Windows, you can now connect and import data from Dataflows directly into Excel. Users are able to immediately load data into Excel or use Power Query to transform the data prior to loading it into Excel.

Don't fear the 'verse. Dataverse is a secure and cloud-based storage for your data. It allows you store a business-focused definition of your

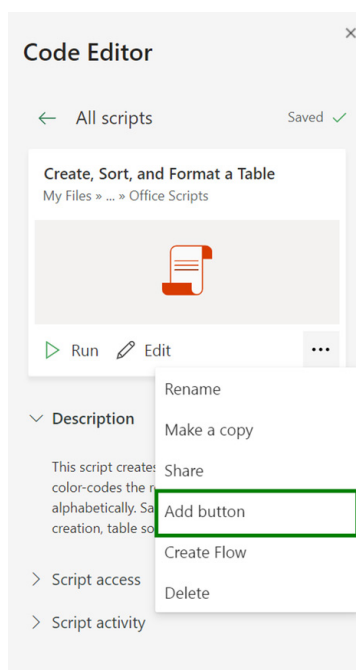
organisation's data for use within applications. With the new 'From Dataverse' connector in Excel for Windows, you can now connect and import data from a Dataverse environment into Excel. Users are able to immediately load data into Excel or use Power Query to transform the data prior to loading it into Excel.

To start using these connectors, go to Data tab on the Ribbon, click 'Get Data', go to 'From Power Platform category', and select 'From Dataflows' or 'From Dataverse' options.



## Running Office Scripts from a workbook button (Current Channel)

Mentioned in last month's newsletter, Microsoft has extended the notion of their recently introduced script buttons feature, allowing the same button to run Office Scripts in Excel for Windows.



With script buttons in Excel for Windows, Office Scripts moves ever nearer to becoming an automation solution you can use to manage your workflow anytime and anywhere. While you still need to use Excel on the web to create and manage your scripts, people use a variety of platforms to complete their tasks and Microsoft has stated that they “...intend to investigate more cross-platform investments as part of [their] future work”.

To create a button in your workbook:

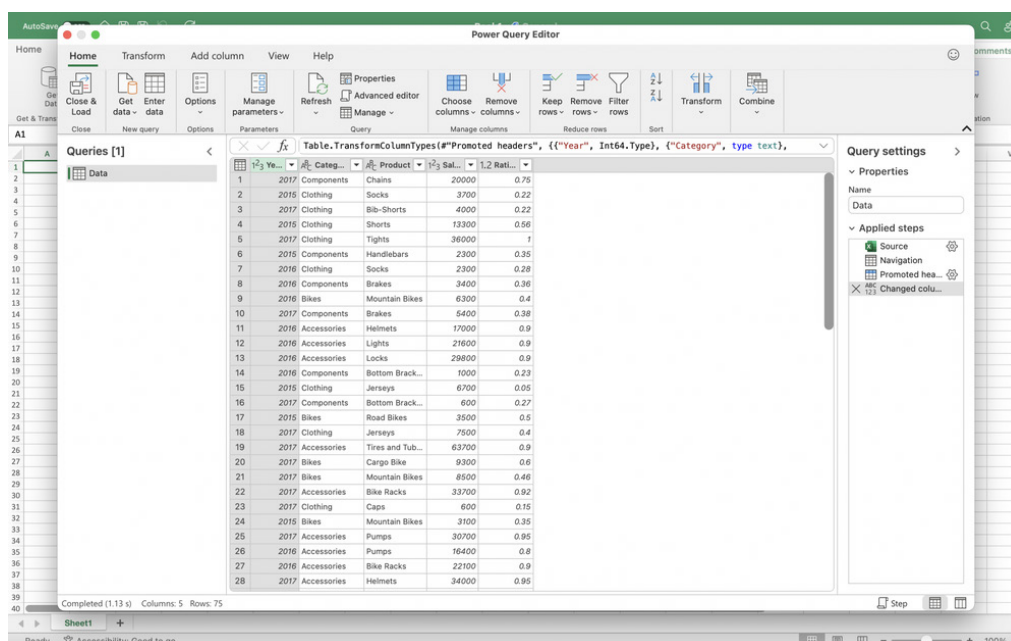
- from the Automate tab in Excel on the web, select a script
- go to the ‘More options’ (...) menu in either the ‘Script Details’, ‘All Scripts’ or ‘Code Editor’ pane
- select ‘Add button’.

To run the button in Windows:

- open a file containing the button in Excel on Windows
- select the button in the workbook.

## Shaping data with Power Query Editor (Insider Beta)

Quite rightly, many users are requesting the ability to transform data using the Query Editor in Excel for Mac be added. Again, we mentioned this last month!

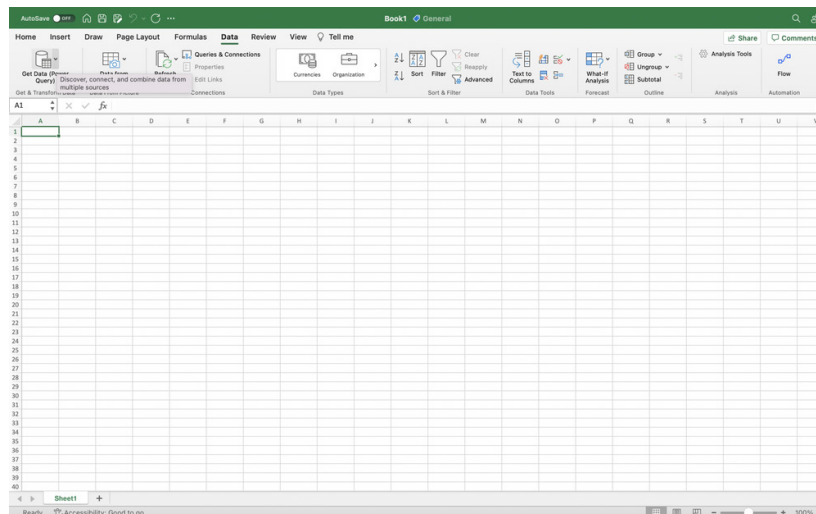


When Power Query was first released in Excel for Mac, there was hope that it wouldn't take long for the features to mirror their Windows counterparts. It's taking time, but the capabilities are improving. Beginning with the ability to refresh data two years ago, importing data was introduced this time last year, so this does seem to be an annual event! As you may imagine, Mac users have desperately requested the ability to transform data using the Query Editor, which would provide

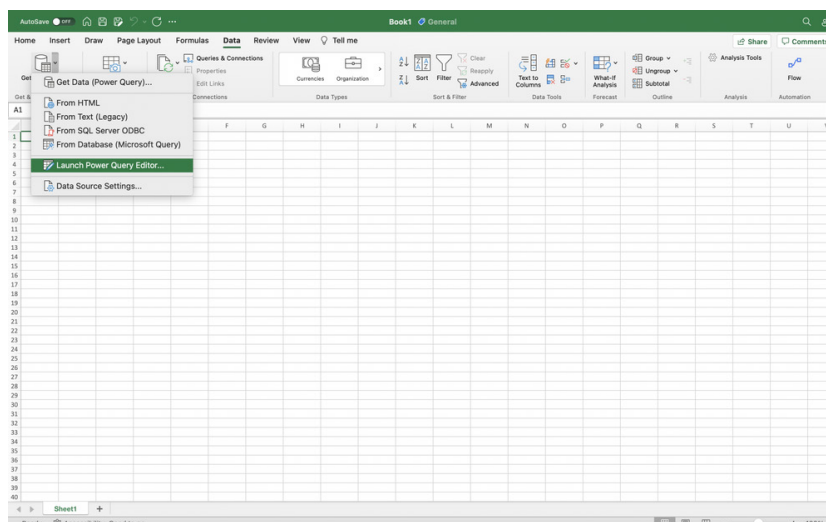
users with the full Power Query experience in Excel for Mac.

Well... it's getting there! You may now clean and shape your data with hundreds of transformations available in Power Query Editor in Excel for Mac.

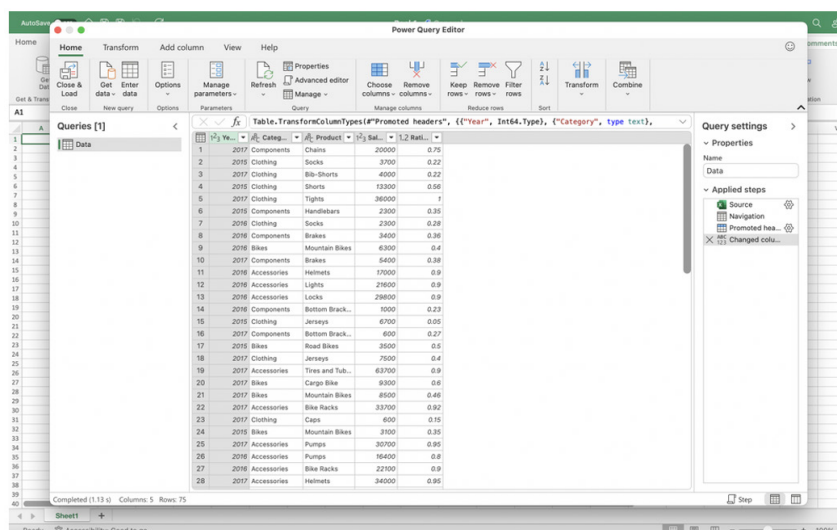
To access, on the Data tab, click the Get Data (Power Query) button.



Click Launch Power Query Editor to open the Query Editor.



You can shape and transform your data using the Query Editor similarly to Excel for Windows. When you're done, click the 'Close & Load' button on the Home tab.



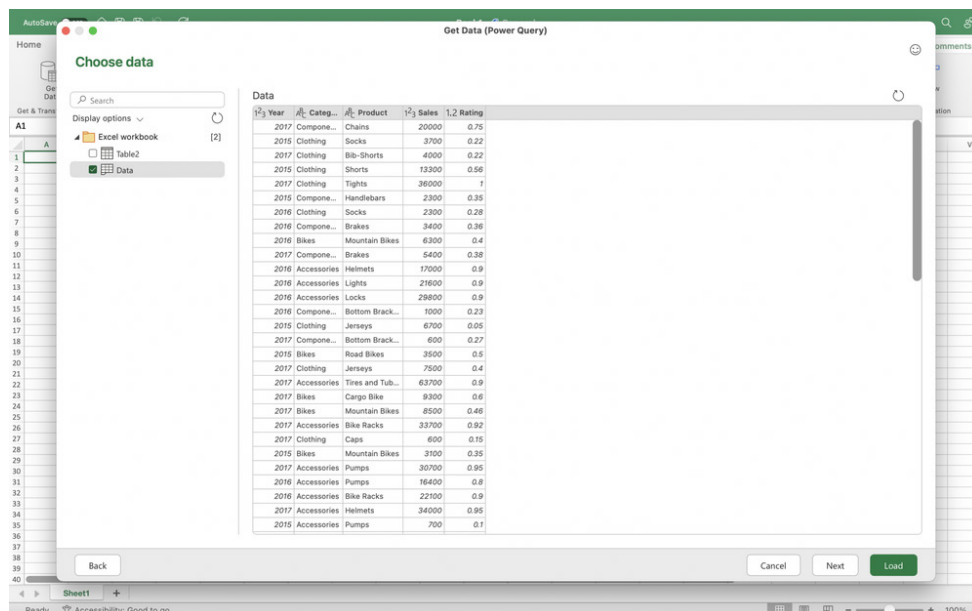


The newly imported data appears in a new sheet.

Supported data sources include:

- text, CSV, XLSX, XML and JSON files
- SharePoint, SharePoint Lists, SharePoint Folders and OData
- local tables, Tables and ranges
- Microsoft SQL Server.

You may also access the Query Editor from the data import flow by clicking the Get Data (Power Query) button, choosing a data source, and clicking the Next button.



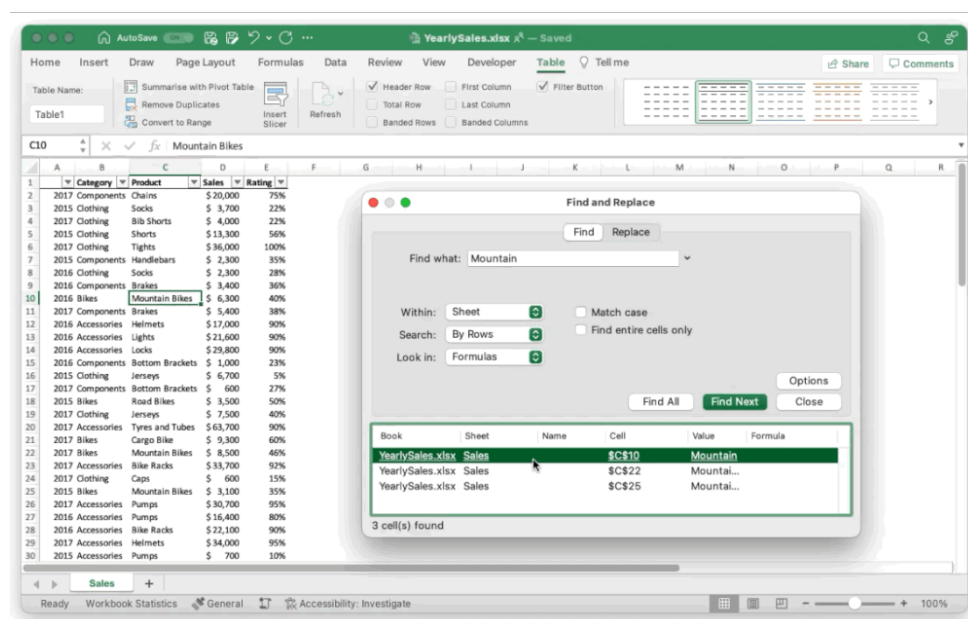
### Faster AutoFilter

Similar to Excel for Windows, users should now find AutoFilter noticeably faster on Excel for Mac too. Improvements were made to reduce memory usage and optimise calls made by the filter's comparison algorithm. The

optimisations should be especially noticeable on low-end devices that have less memory or slower CPU-memory throughput.

### Improved Find dialog and Find All

Excel users should now find the Find dialog in Excel for Mac more consistent with that in Excel for Windows including the option to Find All.





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

## Excel Features Availability

Feature	Insider		Production				Web
	Windows <small>Find the latest Excel version for this platform</small>	Mac <small>Find the latest Excel version for this platform</small>	Windows/CC <small>Find the latest Excel version for this platform</small>	Windows/MEC <small>Find the latest Excel version for this platform</small>	Windows/SA <small>Find the latest Excel version for this platform</small>	Mac <small>Find the latest Excel version for this platform</small>	
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				
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	

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

You can find the updated version of this grid with the interactive links at [aka.ms/ExcelFeaturesFlyer](https://aka.ms/ExcelFeaturesFlyer).

## The A to Z of Excel Functions: IMPOWER



This function is kind of **IMPOWER**ing...

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  $i^2 = -1$ . In general, the square of an imaginary number  $bi$  is  $-b^2$ . For example,  $9i$  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**.

Sometimes you might wish to raise a complex number to a power. **IMPOWER** returns a complex number that is in the  $x + yi$  or  $x + yj$  text format to a given power.

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

**IMPOWER(number, number)**

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

- **number**: this is required and represents the complex number you wish to raise to a power
- **number**: this is also required. This is the power to which you wish to raise the complex number.

It should be further noted that:

- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMPOWER** recognises either the  $i$  or  $j$  notation
- if **number** is a value that is not in the  $x + yi$  or  $x + yj$  text format, **IMPOWER** returns the **#NUM!** error value

- if **inumber** is a logical value, **IMPOWER** returns the #VALUE! error value
- if **number** is non-numeric, **IMPOWER** 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 **IMPOWER** will return an #NUM! error
- a complex raised to a power is calculated as follows:

$$(x + yi)^n = r^n e^{in\theta} = r^n \cos n\theta + ir^n \sin n\theta$$

where:

$$r = \sqrt{x^2 + y^2}$$

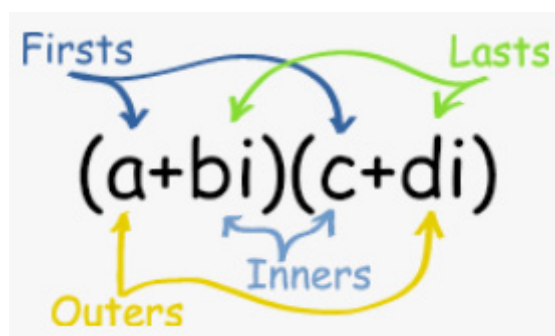
$$\theta = \tan^{-1} \left( \frac{y}{x} \right) \text{ and}$$

$$\theta \in (-\pi; \pi]$$

Please see our example below:

	A	B	C
1	Formula	Description	Result
2	=IMPOWER("3 + 4i", 1.7)	3 + 4i raised to the power 1.7	-0.0864698058642243+15.4256042122232i

## The A to Z of Excel Functions: IMPRODUCT



Sometimes you might wish to multiply complex numbers together. **IMPRODUCT** returns the product of one to 255 complex numbers in the **x + yi** or **x + yj** text format.

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

**IMPRODUCT(inumber1, [inumber2], ...)**

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

- **inumber1, inumber2, ...**: the first argument is required (the remainder are optional). This / these represent(s) the complex number(s) you wish to multiply.

It should be further noted that:

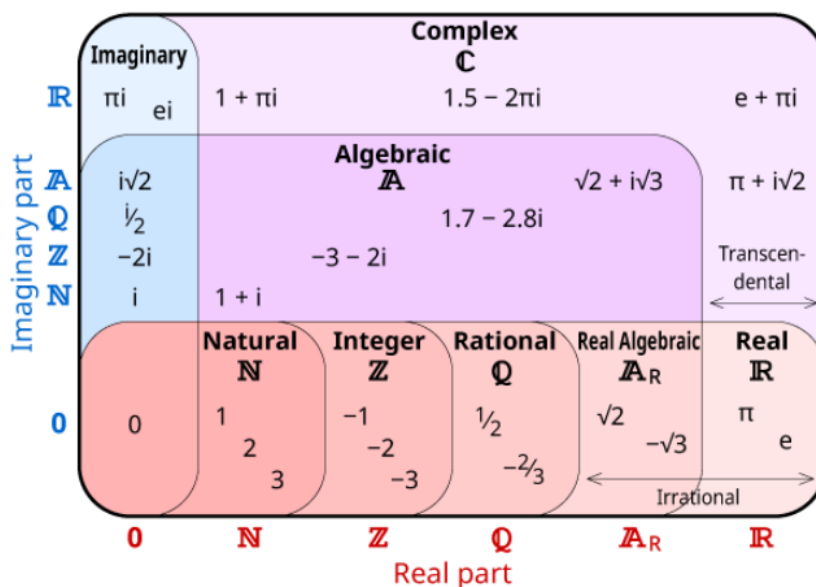
- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMPRODUCT** recognises either the i or j notation
- if any of **inumber1, [inumber2], ...** is a value that is not in the **x + yi** or **x + yj** text format, **IMPRODUCT** returns the #NUM! error value
- if any of **inumber1, [inumber2], ...** is a logical value, **IMPRODUCT** returns the #VALUE! error value
- if any of **inumber1, [inumber2], ...** is non-numeric, **IMPRODUCT** returns the #VALUE! error value
- if any 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 **IMPRODUCT** will return an #NUM! error
- the product of two complex numbers is:

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$$

Please see our second example below:

	A	B	C
1	Formula	Description	Result
2	<code>=IMPRODUCT("3 + 4i", "5 - 6i")</code>	Product of two complex numbers	39+2i
3	<code>=IMPRODUCT("1+j",2)</code>	1+ (no spaces) doubled	2+2j

## The A to Z of Excel Functions: IMREAL



$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{A}_R \subset \mathbb{R} \subset \mathbb{C}$$

**IMREAL** – are you?

The **IMREAL** function returns the real coefficient of a complex number in the form **a + bi**. It employs the following syntax to operate:

**IMREAL(number)**

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

- **number**: this is required and represents the complex number for which you want the imaginary coefficient.

It should be further noted that:

- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMREAL** recognises either the **i** or **j** notation
- 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 **IMREAL** will return an **#NUM!** error.

Please see our final example for this month below:

	A	B	C
1	Formula	Description	Result
2	<code>=IMREAL("3 + 4i")</code>	Real coefficient of the complex number 3 + 4i	3
3	<code>=IMREAL("0 - j")</code>	Real coefficient of the complex number 0 - j (causes an error when there is a space between the operator and the imaginary unit)	#NUM!
4	<code>=IMREAL("0-j")</code>	Real coefficient of the complex number 0-j (works when there is no space between the operator and the imaginary unit)	0
5	<code>=IMREAL("81j")</code>	Real coefficient of the real number 81j	0

More Excel Functions next month.

# Beat the Boredom Suggested Solution

The challenge this month was to create a formula for finding the first visible cell in a range. Easy, yes?

## The Challenge

Filtered month

Month
March
April
May
August
September
October
November

As a reminder, in this challenge, earlier in the newsletter we required you to create a formula for finding first visible cell in a filtered range without using VBA and macros. As always, there was a requirement: this time, it was emphasised that this was a formula challenge; no Power Query / Get & Transform or VBA!

## Suggested Solutions

We have more than one this month! Before explaining our solutions, we will clarify the common idea regarding how we came up with them first.

### For Excel Office 365, Excel on the Web and Office Beta / Insider Versions

These Excel versions allow us to use Dynamic Arrays which help shorten a lot of steps.

To begin with, we can generate a new column using the **SUBTOTAL** function which produces a subtotal in a given list, table or database. Do not forget to use the **COUNTA** function in the **SUBTOTAL** function as the given data consists of text.

The screenshot shows an Excel spreadsheet with the following content:

- Row 1: **First Visible Cell**
- Row 2: **First Visible Cell-Suggested Solutions.xlsx**
- Row 3: **Navigator**
- Row 4: Error Checks: ☒
- Row 8: **Suggested Solutions**
- Row 10: **Alternative Option**
- Row 11: *These only work in Office 365, Excel on the Web and Office Beta / Insider*
- Row 12: **=SUBTOTAL(3,D13)**
- Row 13: **Value**
- Row 14: **1**
- Row 15: **1**
- Row 16: **1**
- Row 17: **1**
- Row 18: **1**
- Row 19: **1**
- Row 20: **1**
- Row 21: **1**
- Row 22: **1**
- Row 23: **1**
- Row 24: **1**
- Row 25: **1**
- Row 26: **1**

The **FILTER** dynamic array function is used in order to find the first visible cell in a given range and uses the following formula:

**=FILTER(Example[Months],Example1[@Value]=1)**

The resulting list will be spilled range created by the formula above written in only cell **N13** as below:

The screenshot shows an Excel spreadsheet with the following content:

- Row 1: **First Visible Cell**
- Row 2: **First Visible Cell-Suggested Solutions.xlsx**
- Row 3: **Navigator**
- Row 4: Error Checks: ☒
- Row 8: **Suggested Solutions**
- Row 10: **Alternative Option**
- Row 11: *These only work in Office 365, Excel on the Web and Office Beta / Insider*
- Row 12: **=FILTER(Example[Months],Example1[@Value]=1)**
- Row 13: **Value**
- Row 14: **Output**
- Row 15: **1 January**
- Row 16: **1 February**
- Row 17: **1 March**
- Row 18: **1 April**
- Row 19: **1 May**
- Row 20: **1 June**
- Row 21: **1 July**
- Row 22: **1 August**
- Row 23: **1 September**
- Row 24: **1 October**
- Row 25: **1 November**
- Row 26: **1 December**

## For All Current Excel Versions

As there is no **FILTER** function (*i.e.* Dynamic Array function) in other versions of Excel, firstly, we need to create few helper columns. To begin with, we will be using the row function in order to find out the rule number of the given data using the formula

**=ROW(Example[Months])**

Months	ROW
January	13
February	14
March	15
April	16
May	17
June	18
July	19
August	20
September	21
October	22
November	23
December	24

Then, the next step would be to find out the counter number for each month using the given formula

**=F13-ROW(INDEX(Example[Months],1))**

Months	ROW	Counter
January	13	0
February	14	1
March	15	2
April	16	3
May	17	4
June	18	5
July	19	6
August	20	7
September	21	8
October	22	9
November	23	10
December	24	11

The next step would be to use the **INDEX** function inside the **OFFSET** function in order to find out the given month for each counter number using formula

**=OFFSET(INDEX(Example[Months],1),G13,0)**

Months	ROW	Counter	OFF SET
January	13	0	January
February	14	1	February
March	15	2	March
April	16	3	April
May	17	4	May
June	18	5	June
July	19	6	July
August	20	7	August
September	21	8	September
October	22	9	October
November	23	10	November
December	24	11	December

The next step would be to use the **COUNTA** function inside the **SUBTOTAL** function in order to find the subtotal of text within the 'OFFSET' column using

**=SUBTOTAL(3,H13)**

**First Visible Cell**  
First Visible Cell-Suggested Solutions.xlsx

Navigator

Error Checks: ☒

**1. First Visible Cell**

**Suggested Solutions**

For All Versions of Excel

Months	ROW	Counter	OFFSET	SUBTOTAL
January	13	0	January	1
February	14	1	February	1
March	15	2	March	1
April	16	3	April	1
May	17	4	May	1
June	18	5	June	1
July	19	6	July	1
August	20	7	August	1
September	21	8	September	1
October	22	9	October	1
November	23	10	November	1
December	24	11	December	1

Then, we may use the **IF** function in order to return the row number against the subtotal column using formula

**=IF(I13,F13)**

**First Visible Cell**  
First Visible Cell-Suggested Solutions.xlsx

Navigator

Error Checks: ☒

**1. First Visible Cell**

**Suggested Solutions**

For All Versions of Excel

Months	ROW	Counter	OFFSET	SUBTOTAL	IF
January	13	0	January	1	13
February	14	1	February	1	14
March	15	2	March	1	15
April	16	3	April	1	16
May	17	4	May	1	17
June	18	5	June	1	18
July	19	6	July	1	19
August	20	7	August	1	20
September	21	8	September	1	21
October	22	9	October	1	22
November	23	10	November	1	23
December	24	11	December	1	24

Then, we may find the minimum value

**=MIN(J13:J24)**

**First Visible Cell**  
First Visible Cell-Suggested Solutions.xlsx

Navigator

Error Checks: ☒

**1. First Visible Cell**

**Suggested Solutions**

For All Versions of Excel

Months	ROW	Counter	OFFSET	SUBTOTAL	IF
January	13	0	January	1	13
February	14	1	February	1	14
March	15	2	March	1	15
April	16	3	April	1	16
May	17	4	May	1	17
June	18	5	June	1	18
July	19	6	July	1	19
August	20	7	August	1	20
September	21	8	September	1	21
October	22	9	October	1	22
November	23	10	November	1	23
December	24	11	December	1	24

**MIN**

**=MIN(J13:J24)**



The final step would be to use **INDEX** in order to find out the first visible cell in the given range using

**=INDEX(Example[Months],G26-ROW(INDEX(Example[Months],1))+1)**

Putting this all together we get the following behemoth:

**=INDEX(Example[Months], MIN(IF(SUBTOTAL(3,OFFSET(INDEX(Example[Months],1),ROW(Example[Months])-ROW(INDEX(Example[Months],1)),0),ROW(Example[Months])-ROW(INDEX(Example[Months],1))+1))))**

There we have it, did you come up with a better solution? Let us know at [contact@sumproduct.com](mailto:contact@sumproduct.com).

Until next time.

## Upcoming SumProduct Training Courses - COVID-19 update

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

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

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

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	19 - 21 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	26 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	27 - 28 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	2 Days
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

Location	Course	Date	Date	Duration	Duration
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 at quick formatting using the **CTRL** and **SHIFT** keys:

Keystroke	What it does
CTRL + SHIFT + 0	Show column
CTRL + SHIFT + 1	Fixed decimal and comma format
CTRL + SHIFT + 2	Time (AM/PM) format
CTRL + SHIFT + 3	Date format
CTRL + SHIFT + 4	Currency format
CTRL + SHIFT + 5	Percentage format
CTRL + SHIFT + 6	Exponential format
CTRL + SHIFT + 7	Outline border
CTRL + SHIFT + 8	Select current region
CTRL + SHIFT + 9	Unhide 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](http://www.sumproduct.com/thought/keyboard-shortcuts). Also, check out our new daily **Excel Tip of the Day** feature on the [www.sumproduct.com](http://www.sumproduct.com) homepage.

## Our Services

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

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

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

## Link to Others

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

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

## Any Questions?

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

## Training

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

Check out our more popular courses in our training brochure:



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

**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**