Sum Product

NEWSLETTER #135 - February 2024

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

As 2024 gets fully underway new features and updates are coming thick and fast like an out of control milkshake machine! This month's newsletter looks at some of the big news stories, including great news about Copilot, debugging formulae with Excel Labs and a new experimental feature (Sheet Defined Functions), plus news on creating PivotTables *on an iPad* and the new **DAX INFO** functions for DAX query view in Power BI.

We could stop right there – but of course, we won't!

There are *most* of the usual suspects too. Whilst it's all quiet regarding New Features in Excel, there is plenty more to get your teeth into. We have our latest Beat the Boredom Challenge, together with the usual tips on Charts & Dashboards, Excel for Mac, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates, the A to Z of Keyboard Shortcuts and we "minute" more of the Excel functions.

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

Liam Bastick, Managing Director, SumProduct

Copilot for All (sort of...)

As we have reported recently, Microsoft's Artificial Intelligence (AI) tool, Copilot, has been released into the wild, albeit for companies wishing to make a purchase for at least 300 users. Even so, the software giant announced it had reached another milestone in trying to connect with the world with more than 5 billion chats and more than 5 billion images to date.

Microsoft is starting to learn how people are using it. Firstly, there is a set of Copilot power users including creators, researchers, programmers and others who want more rapid access to the very latest Microsoft has to offer. Secondly, there is another group of Microsoft 365 users who wish to use it for personal use.

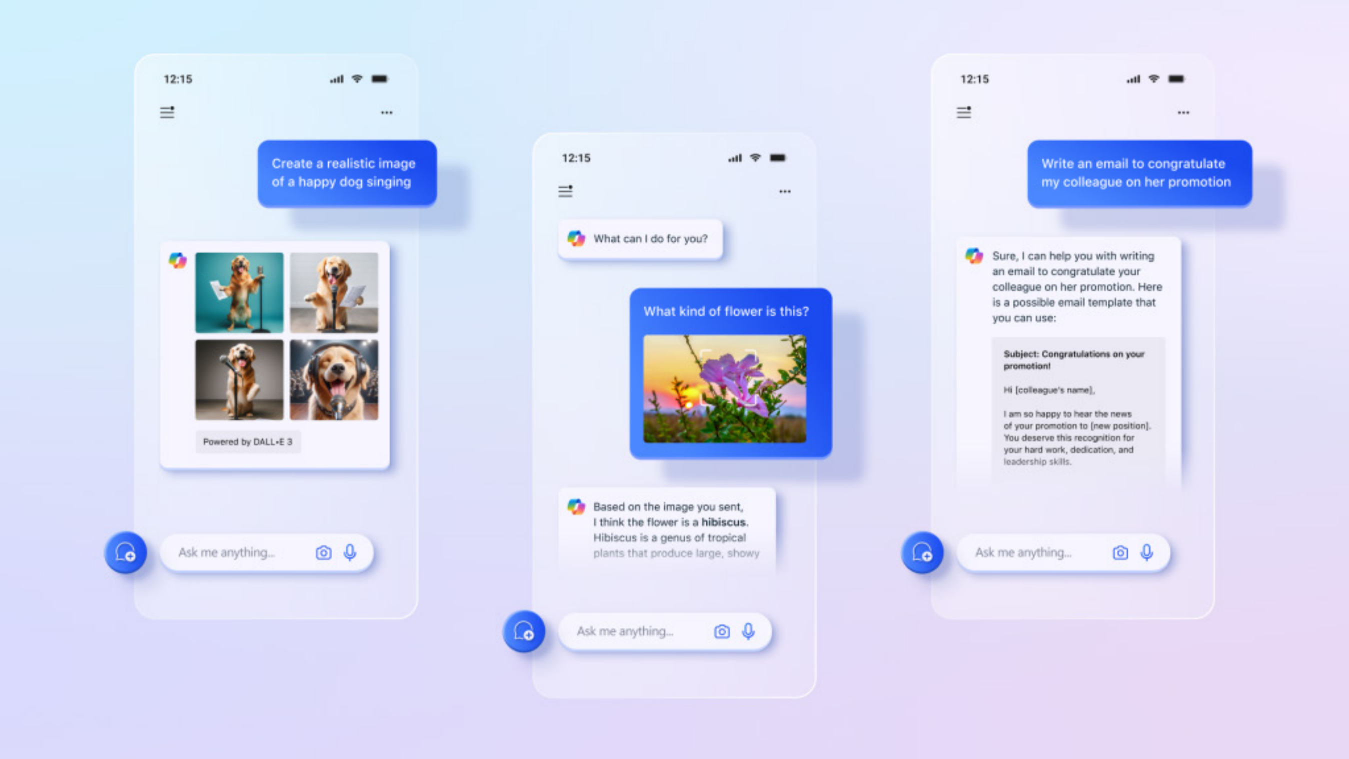
To help address those needs, Microsoft has just announced more options for power users, creators and anyone looking to take their Copilot experience to the next level. This begins with the introduction of Copilot Pro, a new premium subscription for individuals that provides a higher tier of service for AI capabilities, brings Copilot AI capabilities to Microsoft 365 Personal and Family subscribers, and new capabilities, such as the ability to create Copilot GPTs. Furthermore, they are also announcing the General Availability of their Copilot app for iOS and Android phones. Finally, and arguably most importantly, Copilot for Microsoft 365 is to be brought to more commercial customers by expanding the availability to businesses of all sizes, including small- and medium-sized businesses, starting right now.

Copilot Pro is a new subscription that delivers the most advanced features and capabilities of Microsoft Copilot to individuals looking to obtain Copilot. This software can assist with writing, coding, designing, researching or learning, as required.

Copilot Pro provides:

- a single AI experience that runs across your devices, understanding your context on the web, on your PC, across your apps and soon on your phone
- access to Copilot in Word, Excel (Preview only, just in English), PowerPoint, Outlook and OneNote on PC, Mac and iPad for Microsoft 365 Personal and Family subscribers
- priority access to the very latest models, beginning with OpenAI's GPT-4 Turbo
- enhanced AI image creation with Image Creator from Designer (formerly Bing Image Creator)
- the ability to build your own Copilot GPT (Generative Pre-trained Transformer), a customised Copilot tailored for a specific topic – in the coming soon Copilot GPT Builder with just a simple set of prompts.

While Copilot Pro (priced at US$20 per month per user) is aimed at individuals, Copilot for Microsoft 365 is aimed at organisations. Copilot for Microsoft 365 became Generally Available for enterprises in November, with 40% of the Fortune 100 participated in the Early Access Program. However, Copilot for Microsoft 365 is now available for organisations of all sizes — with no seat minimum. About time!



Create a realistic image of a happy dog singing

What can I do for you?

What kind of flower is this?

Based on the image you sent, I think the flower is a **hibiscus**. Hibiscus is a genus of tropical plants that produce large, showy

Write an email to congratulate my colleague on her promotion

Sure, I can help you with writing an email to congratulate your colleague on her promotion. Here is a possible email template that you can use:

Subject: Congratulations on your promotion!

Hi [colleague's name],

I am so happy to hear the news of your promotion to [new position]. You deserve this recognition for your hard work, dedication, and leadership skills.

These latest updates include the following information:

- Copilot for Microsoft 365 is now Generally Available for small businesses with Microsoft 365 Business Premium and Business Standard Customers can purchase between one and 299 seats for a price of US$30 per month per user
- The 300-seat purchase minimum for commercial plans has been removed. Microsoft has made Copilot available for Office 365 E3 and E5 customers (a Microsoft 365 license was previously required)
- This builds on December's announcement of Copilot for Microsoft 365 being available for education faculty and staff.

Copilot for Microsoft 365 has greater power for organisations because it works across your entire universe of data at work including emails, meetings, chats, documents and more, plus the web. With natural language prompts like "Tell my team how we updated the product strategy," Copilot can generate a status update based on the morning's meetings, emails and chat threads.

As the availability of Copilot expands to more people, there is still a free experience for anyone interested in exploring how Copilot can transform productivity and creativity using AI. You can find out more by visiting copilot.microsoft.com.

The latest features include:

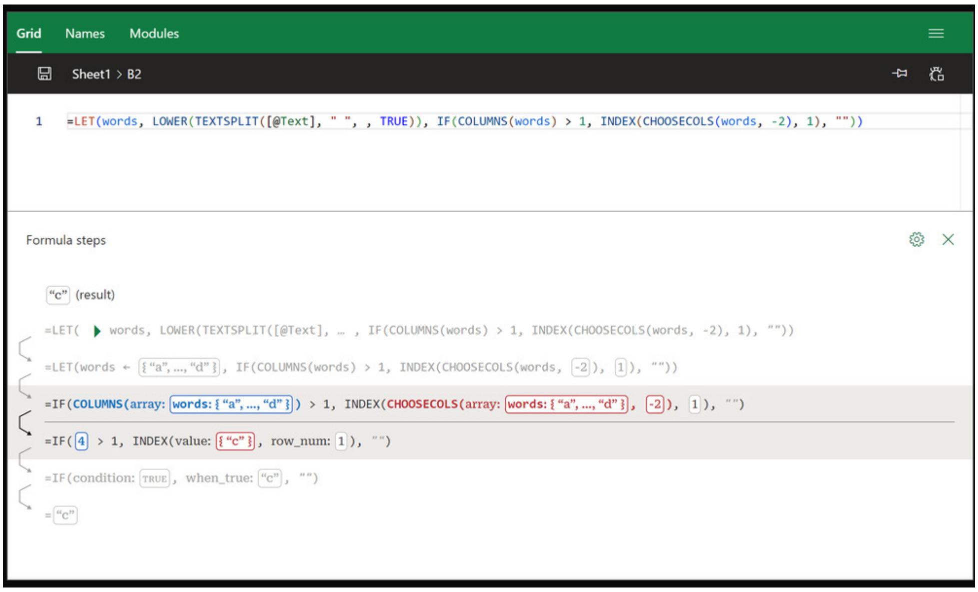
- **Copilot GPTs:** these let you customise the behaviour of Microsoft Copilot on a topic that is of particular interest to you. A handful of Copilot GPTs will start to roll out with specific purposes such as fitness, travel, cooking and more. Soon, Copilot Pro users will also be able to create their own Copilot GPTs using Copilot GPT Builder
- **Copilot mobile app:** this is now available for Android and iOS. The Copilot app gives you the power of Copilot on the go as your Copilot queries and chats will roam across your phone and PC. The Copilot mobile app includes the same capabilities of Copilot on your PC including access to GPT-4, Dall-E 3 for image creation, and the ability to use images from your phone when chatting with Copilot

- **Copilot in the Microsoft 365 mobile app:** Microsoft is also adding Copilot to the Microsoft 365 mobile app for Android and iOS for individuals with a Microsoft account. This new feature is rolling out over the next month. Access Copilot right inside the app and easily export the content you create to a Word or PDF document.

With these latest announcements, Copilot is coming to more of us. Let's hope it gets to all who want it very soon.

Debugging Formulae with Excel Labs

You may know SumProduct work as model auditors amongst other duties: have you ever thought of calling us in? On occasion, it's likely you have found yourself tangled in a web of complex Excel formulae, trying to identify the source of an error. Well, you might want to know that the latest experiment from Excel Labs is a new formula debugger, available within the Advanced Formula Environment (AFE).



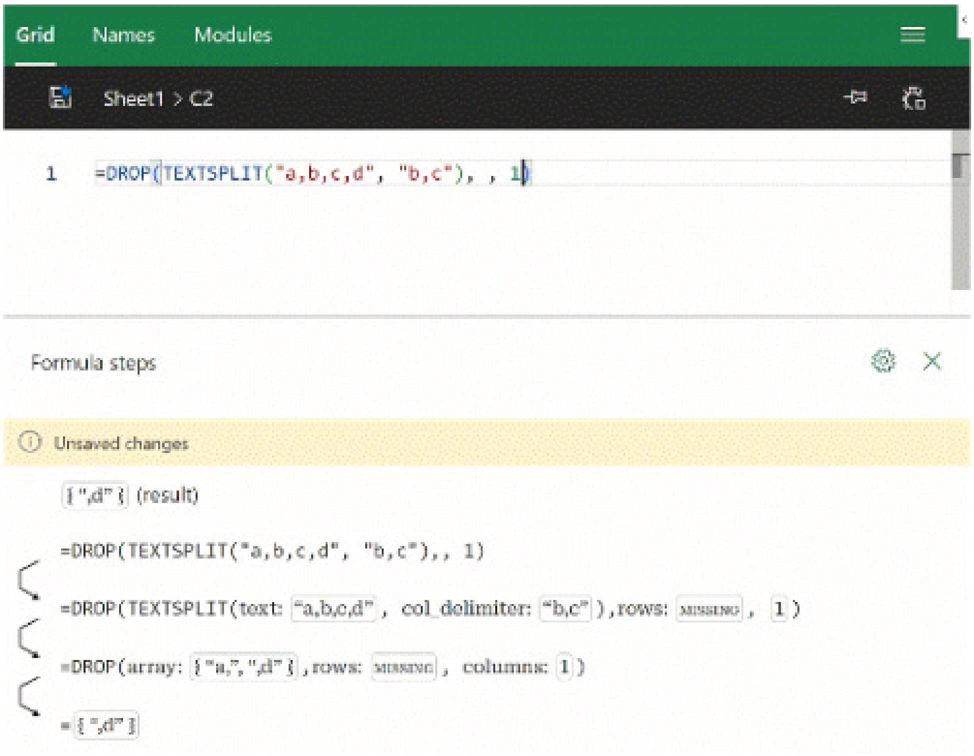
```
=LET(words, LOWER(TEXTSPLIT([@Text], " ", , TRUE)), IF(COLUMNS(words) > 1, INDEX(CHOOSECOLS(words, -2), 1), ""))
```

```
"c" (result)
=LET( words, LOWER(TEXTSPLIT([@Text], … , IF(COLUMNS(words) > 1, INDEX(CHOOSECOLS(words, -2), 1), ""))
=LET(words ← { "a", ..., "d" }, IF(COLUMNS(words) > 1, INDEX(CHOOSECOLS(words, -2), 1), ""))
=IF(COLUMNS(array: words: { "a", ..., "d" }) > 1, INDEX(CHOOSECOLS(array: words: { "a", ..., "d" }, -2), 1), "")
=IF(4 > 1, INDEX(value: { "c" }, row_num: 1), "")
=IF(condition: TRUE, when_true: "c", "")
="c"
```

The debugging capabilities are activated using the new button on the Grid page in AFE. When the button is activated, the debugger will be visible in the same editing pane, and you are ready to go.

Here are some of the key features to look out for:

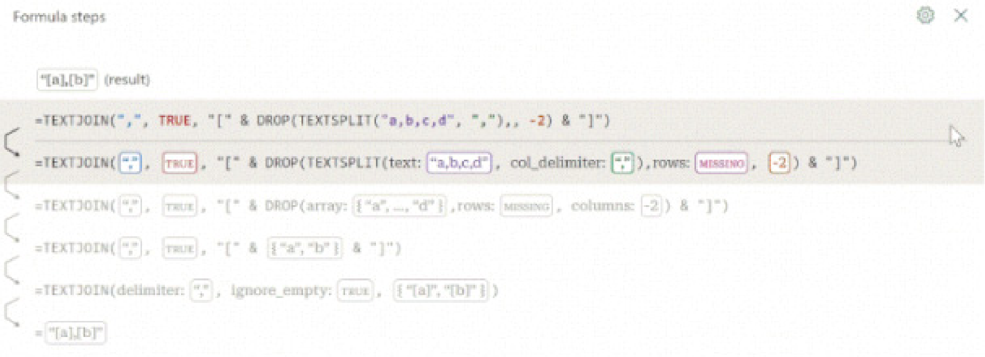
- **Live debugging:** the debugger updates as you type, making it quick to explore how different variations of a formula are evaluated



```
=DROP(TEXTSPLIT("a,b,c,d", "b,c"), , 1)
```

```
{ ",d" } (result)
=DROP(TEXTSPLIT("a,b,c,d", "b,c"),, 1)
=DROP(TEXTSPLIT(text: "a,b,c,d", col_delimiter: "b,c"),rows: MISSING, 1)
=DROP(array: { "a,", ",d" },rows: MISSING, columns: 1)
={ ",d" }
```

- **Evaluation steps:** each evaluation step is shown, with highlights and underlines making it easy to see what changed at every step

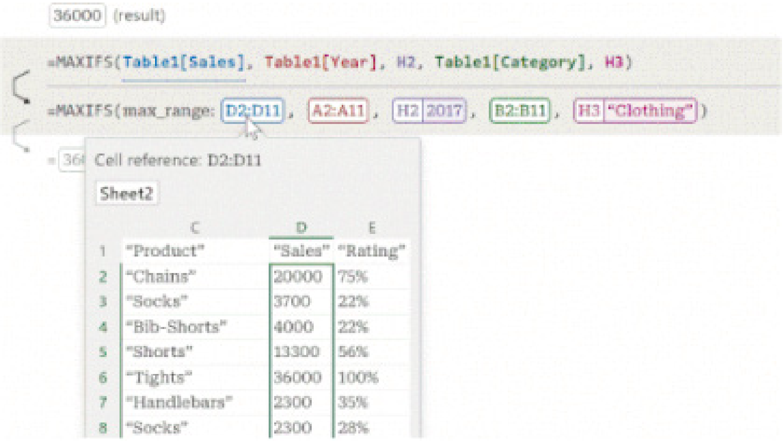


Formula steps

"[a],[b]" (result)

```
=TEXTJOIN(",", TRUE, "[" & DROP(TEXTSPLIT("a,b,c,d", ","),, -2) & "]")
=TEXTJOIN(",", TRUE, "[" & DROP(TEXTSPLIT(text: "a,b,c,d", col_delimiter: ","),rows: MISSING, -2) & "]")
=TEXTJOIN(",", TRUE, "[" & DROP(array: {"a", ..., "d"},rows: MISSING, columns: -2) & "]")
=TEXTJOIN(",", TRUE, "[" & {"a", "b"} & "]")
=TEXTJOIN(delimiter: ",", ignore_empty: TRUE, {"[a]", "[b]"})
="[a],[b]"
```

- **Range preview:** view a preview of the grid when hovering over a reference. Previews show the surrounding context to make it easier to navigate



36000 (result)

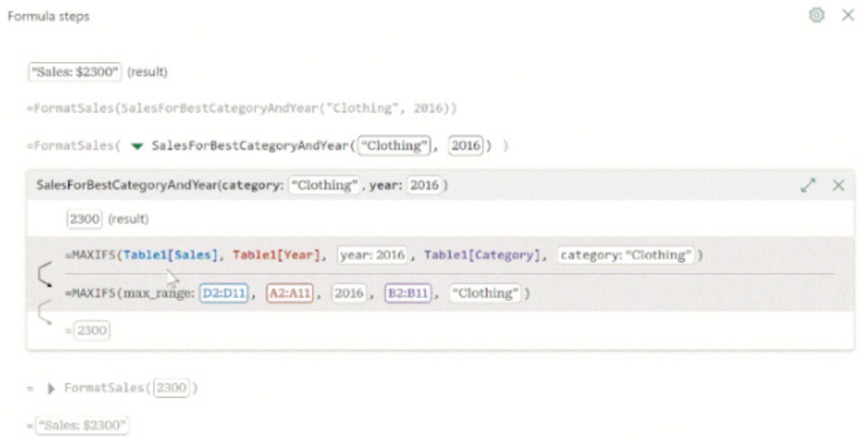
```
=MAXIFS(Table1[Sales], Table1[Year], H2, Table1[Category], H3)
=MAXIFS(max_range: D2:D11, A2:A11, H2 2017, B2:B11, H3 "Clothing")
= 36(
```

Cell reference: D2:D11

Sheet2

	C	D	E
1	"Product"	"Sales"	"Rating"
2	"Chains"	20000	75%
3	"Socks"	3700	22%
4	"Bib-Shorts"	4000	22%
5	"Shorts"	13300	56%
6	"Tights"	36000	100%
7	"Handlebars"	2300	35%
8	"Socks"	2300	28%

- **LAMBDA debugging:** you may debug **LAMBDA** formulae, such as **SalesForBestCategory**, shown below. Every function call is contained in an expandable card. This lets you dive into the particular functions you are interested in, to see how they evaluate.



Formula steps

"Sales: $2300" (result)

```
=FormatSales(SalesForBestCategoryAndYear("Clothing", 2016))
=FormatSales( ▼ SalesForBestCategoryAndYear("Clothing", 2016) )
```

SalesForBestCategoryAndYear(category: "Clothing" , year: 2016)

2300 (result)

```
=MAXIFS(Table1[Sales], Table1[Year], year: 2016 , Table1[Category], category: "Clothing" )
=MAXIFS(max_range: D2:D11, A2:A11, 2016 , B2:B11, "Clothing" )
=2300
```

```
= ▶ FormatSales(2300)
="Sales: $2300"
```

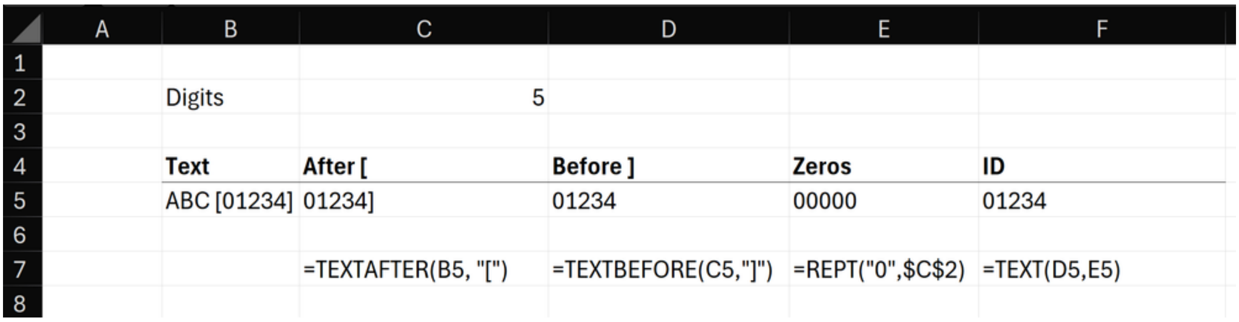
Not all functions are supported just yet (**XLOOKUP** is one notable absentee). If a function or scenario is blocking your work, Microsoft has asked that you let them know through the usual feedback channels.

The debugger will be pushed to Excel Labs automatically and there is no need to explicitly update the add-in. If you do not have Excel Labs, you can install the add-in from the Office store.

Sheet-defined Functions within the Advanced Formula Environment (AFE)

Another experimental feature in the Advanced Formula Environment (AFE) is the ability to automatically convert a multi-cell calculation into a function, also known as sheet-defined functions. This capability has been part of AFE for a while, but we thought we'd take this opportunity to explain it.

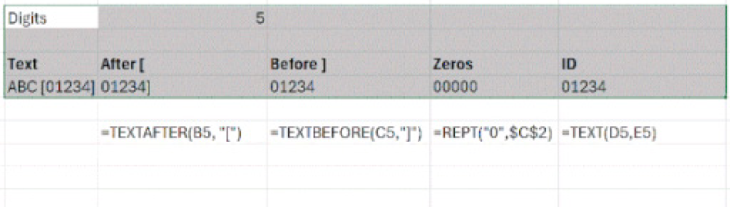
Often, modellers find themselves spreading a complex calculation over many cells ("Rule of Thumb" anyone?), like this one, that extracts an ID from a piece of text:



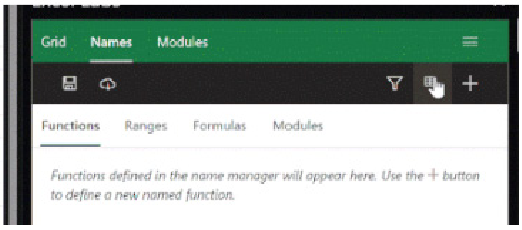
	A	B	C	D	E	F
1						
2		Digits	5			
3						
4		**Text**	**After [**	**Before]**	**Zeros**	**ID**
5		ABC [01234]	01234]	01234	00000	01234
6						
7			=TEXTAFTER(B5, "[")	=TEXTBEFORE(C5,"]")	=REPT("0",C2)	=TEXT(D5,E5)
8						

After writing the calculation, we are faced with either many intermediate cells or we must roll the formulae into one single "mega-formula". With AFE and the 'Add function from grid' feature, we can automatically convert these formulae into a tidy function using three simple steps:

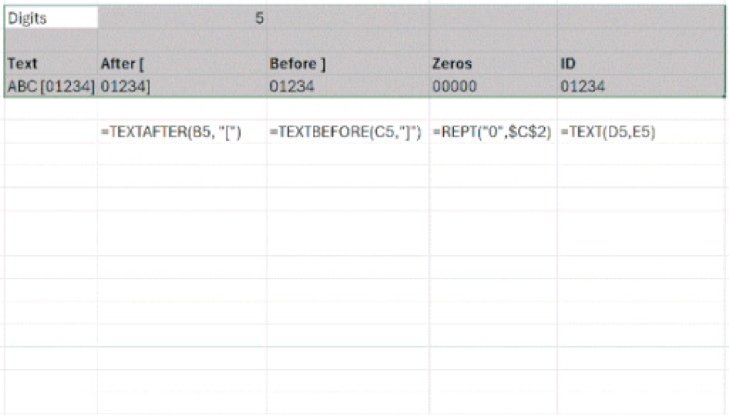
1. Select the range containing the formulae that will make up the function
2. Click the 'Add function from grid' button and provide the references that will define the inputs and outputs. Sometimes, AFE can automatically detect the inputs and outputs based upon the dependencies
3. AFE will present you with the generated function that combines the formulae in the range. If you use labels adjacent to formulae, AFE can also use those to generate friendly parameter names, like 'Text' or 'After'.



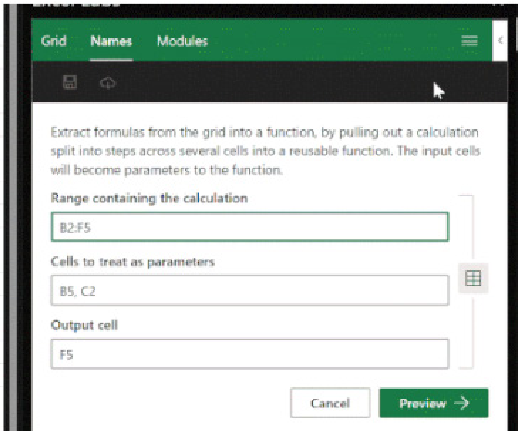
Digits	5			
Text	After [	Before]	Zeros	ID
ABC [01234]	01234]	01234	00000	01234
	=TEXTAFTER(B5, "[")	=TEXTBEFORE(C5,"]")	=REPT("0",C2)	=TEXT(D5,E5)



Functions defined in the name manager will appear here. Use the + button to define a new named function.



Digits	5			
Text	After [	Before]	Zeros	ID
ABC [01234]	01234]	01234	00000	01234
	=TEXTAFTER(B5, "[")	=TEXTBEFORE(C5,"]")	=REPT("0",C2)	=TEXT(D5,E5)

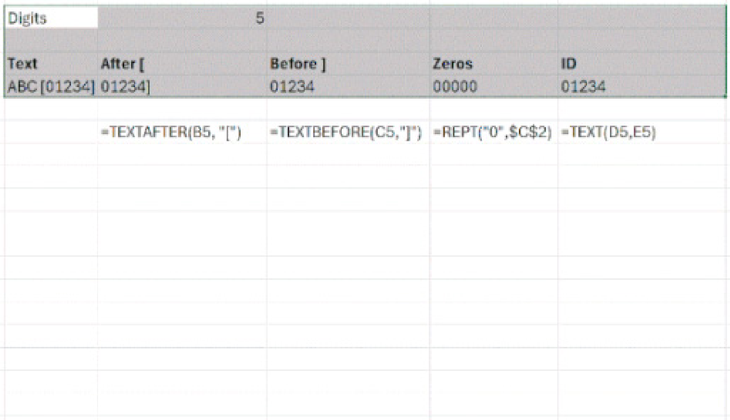


Extract formulas from the grid into a function, by pulling out a calculation split into steps across several cells into a reusable function. The input cells will become parameters to the function.

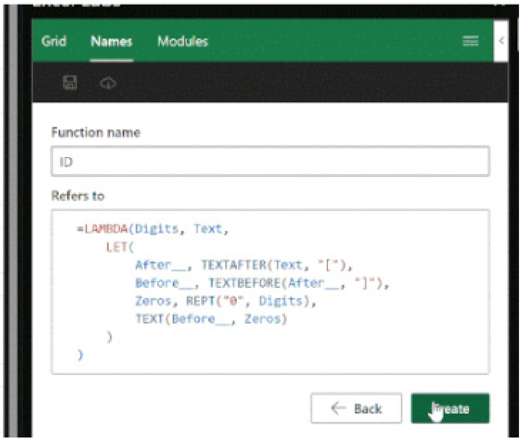
Range containing the calculation: B2:F5

Cells to treat as parameters: B5, C2

Output cell: F5



Digits		5		
Text	After [	Before]	Zeros	ID
ABC [01234]	01234]	01234	00000	01234
	=TEXTAFTER(B5, "[")	=TEXTBEFORE(C5,"]")	=REPT("0",C2)	=TEXT(D5,E5)

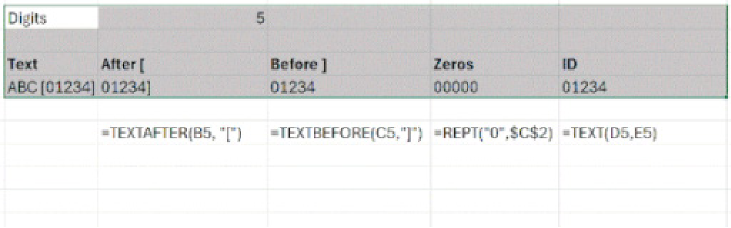


Function name

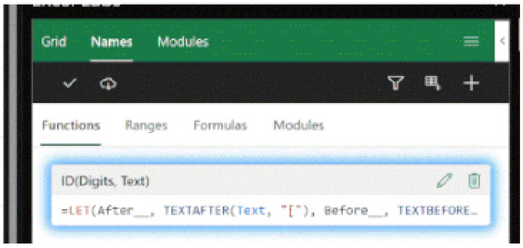
ID

Refers to

```
=LAMBDA(Digits, Text,
    LET(
        After__, TEXTAFTER(Text, "["),
        Before__, TEXTBEFORE(After__, "]"),
        Zeros, REPT("0", Digits),
        TEXT(Before__, Zeros)
    )
)
```



Digits		5		
Text	After [	Before]	Zeros	ID
ABC [01234]	01234]	01234	00000	01234
	=TEXTAFTER(B5, "[")	=TEXTBEFORE(C5,"]")	=REPT("0",C2)	=TEXT(D5,E5)



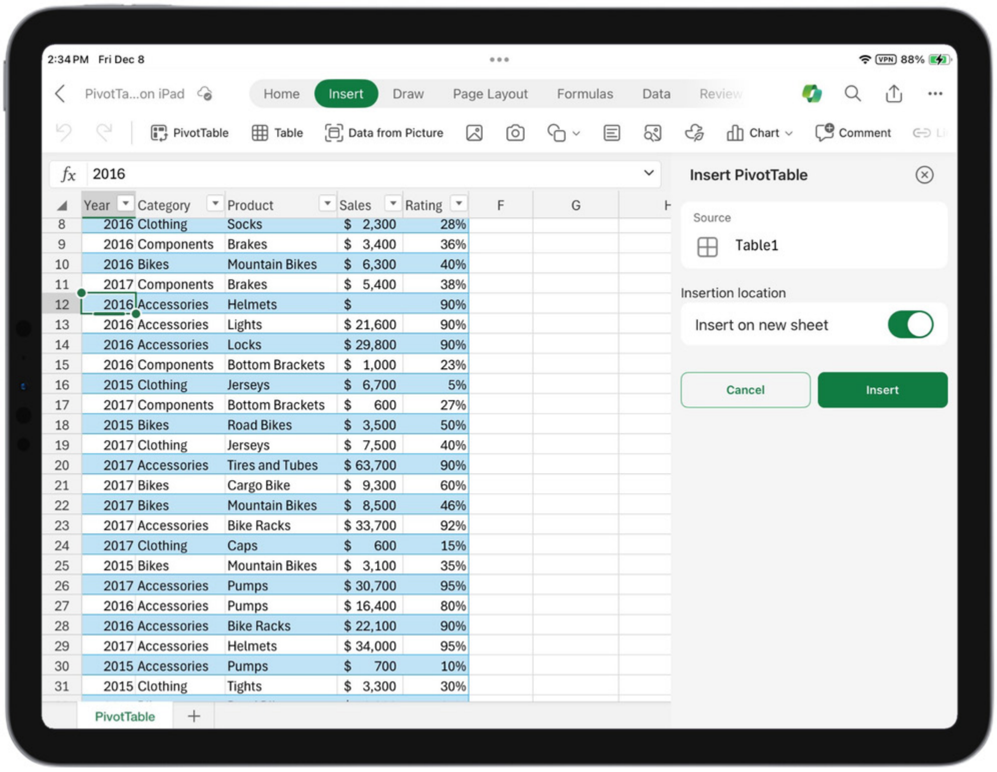
ID(Digits, Text)

```
=LET(After__, TEXTAFTER(Text, "["), Before__, TEXTBEFORE…
```

PivotTables Now on iPad

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

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



fx 2016

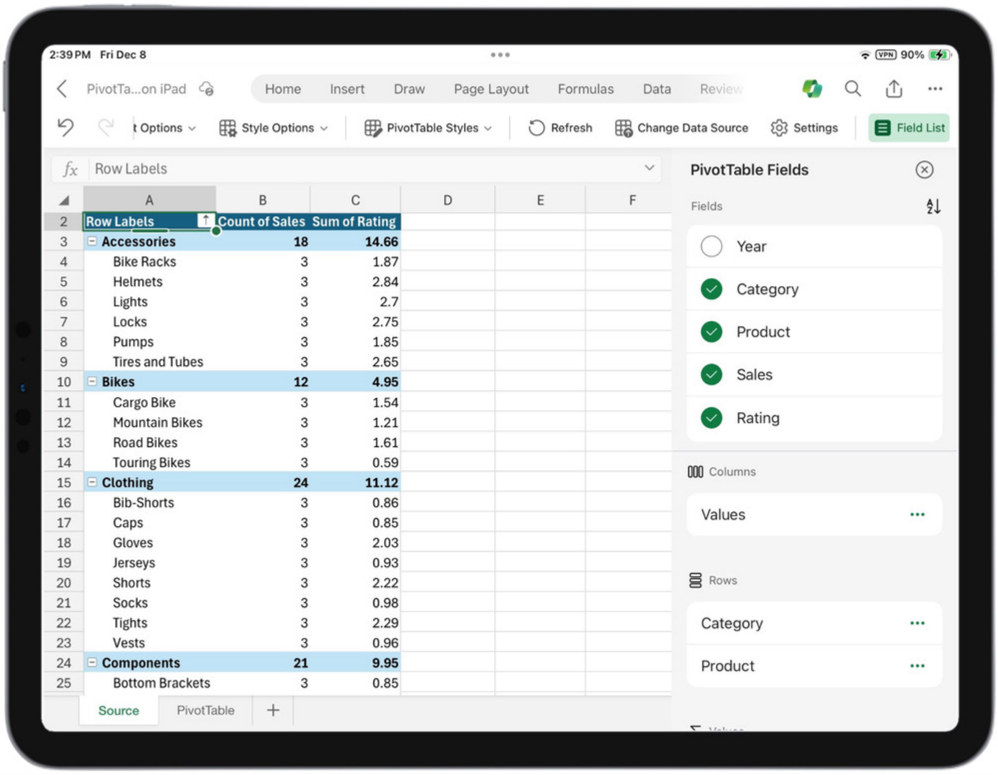
	Year	Category	Product	Sales	Rating
8	2016	Clothing	Socks	$ 2,300	28%
9	2016	Components	Brakes	$ 3,400	36%
10	2016	Bikes	Mountain Bikes	$ 6,300	40%
11	2017	Components	Brakes	$ 5,400	38%
12	2016	Accessories	Helmets	$	90%
13	2016	Accessories	Lights	$ 21,600	90%
14	2016	Accessories	Locks	$ 29,800	90%
15	2016	Components	Bottom Brackets	$ 1,000	23%
16	2015	Clothing	Jerseys	$ 6,700	5%
17	2017	Components	Bottom Brackets	$ 600	27%
18	2015	Bikes	Road Bikes	$ 3,500	50%
19	2017	Clothing	Jerseys	$ 7,500	40%
20	2017	Accessories	Tires and Tubes	$ 63,700	90%
21	2017	Bikes	Cargo Bike	$ 9,300	60%
22	2017	Bikes	Mountain Bikes	$ 8,500	46%
23	2017	Accessories	Bike Racks	$ 33,700	92%
24	2017	Clothing	Caps	$ 600	15%
25	2015	Bikes	Mountain Bikes	$ 3,100	35%
26	2017	Accessories	Pumps	$ 30,700	95%
27	2016	Accessories	Pumps	$ 16,400	80%
28	2016	Accessories	Bike Racks	$ 22,100	90%
29	2017	Accessories	Helmets	$ 34,000	95%
30	2015	Accessories	Pumps	$ 700	10%
31	2015	Clothing	Tights	$ 3,300	30%

Insert PivotTable

Source: Table1

Insertion location: Insert on new sheet ●

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



Row Labels	Count of Sales	Sum of Rating
⊟ Accessories	18	14.66
Bike Racks	3	1.87
Helmets	3	2.84
Lights	3	2.7
Locks	3	2.75
Pumps	3	1.85
Tires and Tubes	3	2.65
⊟ Bikes	12	4.95
Cargo Bike	3	1.54
Mountain Bikes	3	1.21
Road Bikes	3	1.61
Touring Bikes	3	0.59
⊟ Clothing	24	11.12
Bib-Shorts	3	0.86
Caps	3	0.85
Gloves	3	2.03
Jerseys	3	0.93
Shorts	3	2.22
Socks	3	0.98
Tights	3	2.29
Vests	3	0.96
⊟ Components	21	9.95
Bottom Brackets	3	0.85

PivotTable Fields

Fields

○ Year
✓ Category
✓ Product
✓ Sales
✓ Rating

Columns

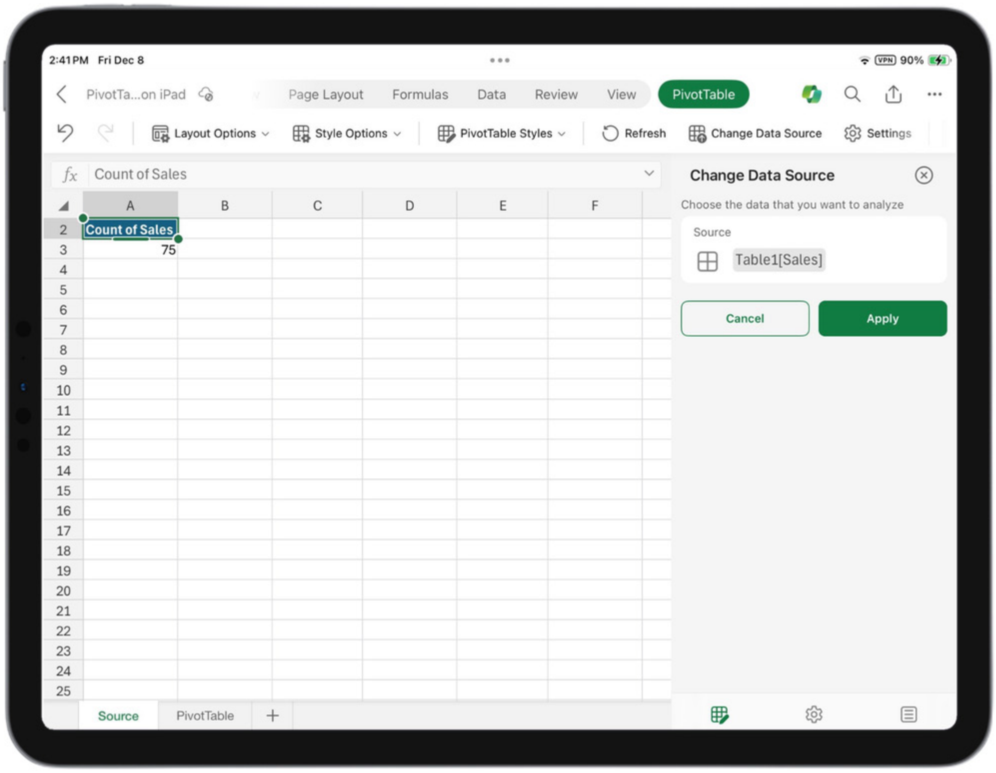
Values

Rows

Category

Product

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



Count of Sales
75

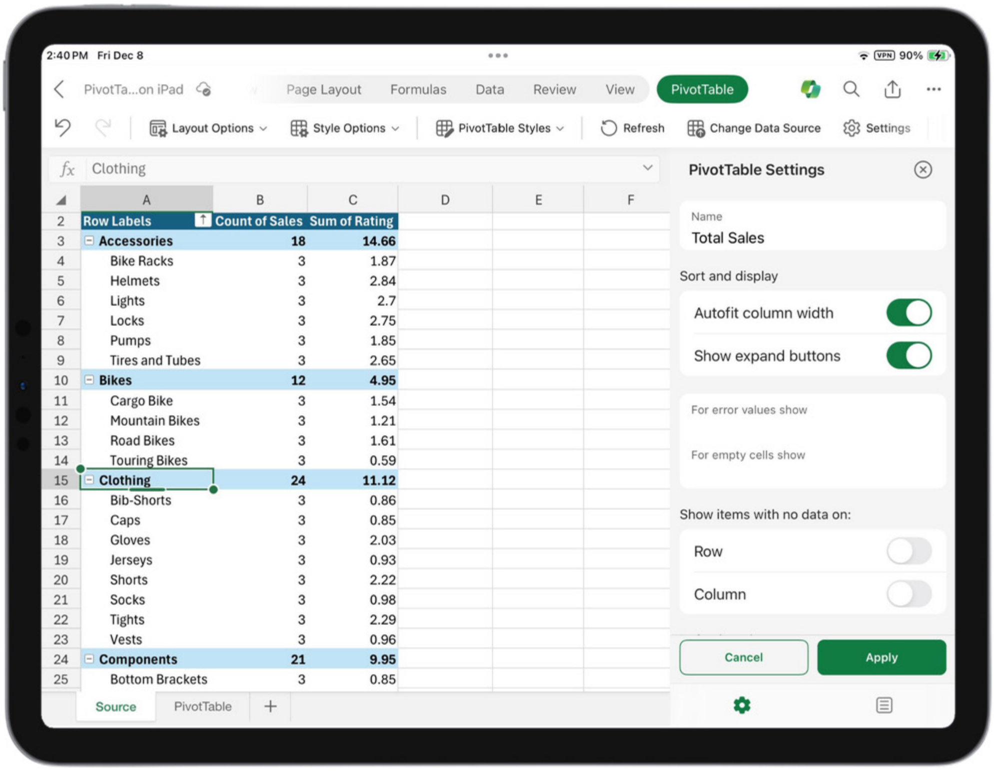
Change Data Source

Choose the data that you want to analyze

Source

Table1[Sales]

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



fx Clothing

	A	B	C
2	Row Labels	Count of Sales	Sum of Rating
3	⊟ Accessories	18	14.66
4	Bike Racks	3	1.87
5	Helmets	3	2.84
6	Lights	3	2.7
7	Locks	3	2.75
8	Pumps	3	1.85
9	Tires and Tubes	3	2.65
10	⊟ Bikes	12	4.95
11	Cargo Bike	3	1.54
12	Mountain Bikes	3	1.21
13	Road Bikes	3	1.61
14	Touring Bikes	3	0.59
15	⊟ Clothing	24	11.12
16	Bib-Shorts	3	0.86
17	Caps	3	0.85
18	Gloves	3	2.03
19	Jerseys	3	0.93
20	Shorts	3	2.22
21	Socks	3	0.98
22	Tights	3	2.29
23	Vests	3	0.96
24	⊟ Components	21	9.95
25	Bottom Brackets	3	0.85

PivotTable Settings

Name
Total Sales

Sort and display

Autofit column width ●

Show expand buttons ●

For error values show

For empty cells show

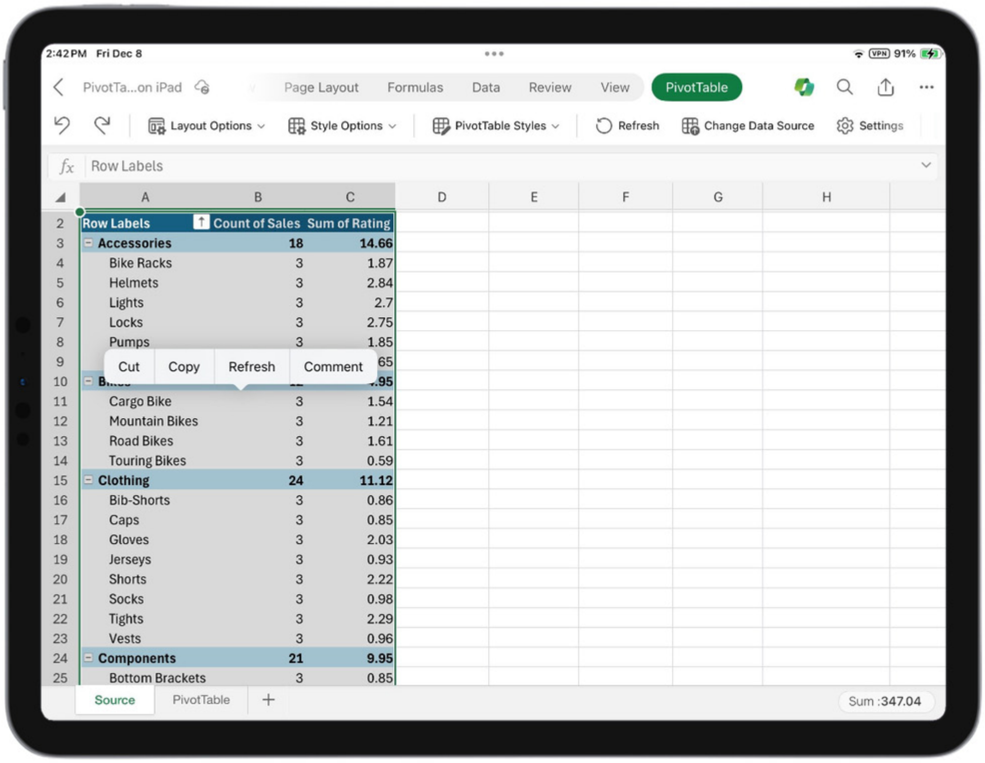
Show items with no data on:

Row ○

Column ○

Cancel | Apply

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



fx Row Labels

	A	B	C
2	Row Labels	Count of Sales	Sum of Rating
3	⊟ Accessories	18	14.66
4	Bike Racks	3	1.87
5	Helmets	3	2.84
6	Lights	3	2.7
7	Locks	3	2.75
8	Pumps	3	1.85
9			65
10	⊟ B		.95
11	Cargo Bike	3	1.54
12	Mountain Bikes	3	1.21
13	Road Bikes	3	1.61
14	Touring Bikes	3	0.59
15	⊟ Clothing	24	11.12
16	Bib-Shorts	3	0.86
17	Caps	3	0.85
18	Gloves	3	2.03
19	Jerseys	3	0.93
20	Shorts	3	2.22
21	Socks	3	0.98
22	Tights	3	2.29
23	Vests	3	0.96
24	⊟ Components	21	9.95
25	Bottom Brackets	3	0.85

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

New INFO DAX Functions

We've talked about DAX query view quite a bit recently. Since it was released in public Preview back in November, upcoming / current Power BI Desktop releases will now include the new **INFO DAX** functions and improvements to the Update model CodeLens.

It shouldn't be understated just what has hit Power BI. There are more than 50 new **INFO DAX** functions now available! If you're familiar with the Dynamic Management Views (DMVs) of models in Power BI, Azure Analysis Services and SQL Server Analysis Services, you might recognise these as the **TMSCHEMA** DMVs. The existing **TMSCHEMA** DMVs are now available as **DAX** functions. Providing them as **DAX** functions gives the following benefits:

- no longer do you have to use a different query syntax than **DAX** to see information about your semantic model. These are native **DAX** functions and show in IntelliSense when you type **INFO**
- they may be combined with other **DAX** functions. Existing DMV query syntax does not allow you to join them. As a **DAX** function, the output is a Table data type and existing **DAX** functions that join tables or summarise tables may be used.

Here's the full list of the new **INFO DAX** functions (deep breath):

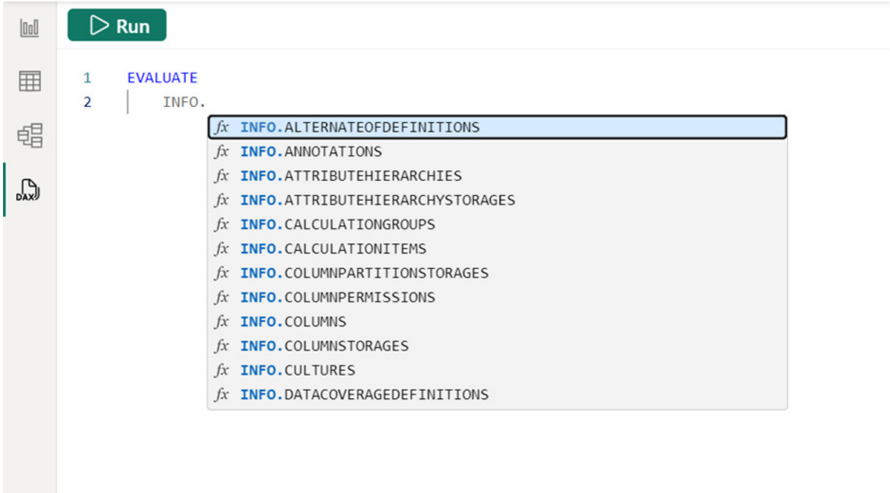
- **INFO.ALTERNATEOFDEFINITIONS**
- **INFO.ANNOTATIONS**
- **INFO.ATTRIBUTEHIERARCHIES**
- **INFO.ATTRIBUTEHIERARCHYSTORAGES**
- **INFO.CALCULATIONGROUPS**
- **INFO.CALCULATIONITEMS**
- **INFO.COLUMNPARTITIONSTORAGES**
- **INFO.COLUMNPERMISSIONS**
- **INFO.COLUMNS**
- **INFO.COLUMNSTORAGES**
- **INFO.CULTURES**
- **INFO.DATACOVERAGEDEFINITIONS**
- **INFO.DATASOURCES**
- **INFO.DELTATABLEMETADATASTORAGES**
- **INFO.DETAILROWSDEFINITIONS**
- **INFO.DICTIONARYSTORAGES**
- **INFO.EXPRESSIONS**
- **INFO.EXTENDEDPROPERTIES**
- **INFO.FORMATSTRINGDEFINITIONS**
- **INFO.GENERALSEGMENTMAPSEGMENTMETADATASTORAGES**
- **INFO.GROUPBYCOLUMNS**
- **INFO.HIERARCHIES**
- **INFO.HIERARCHYSTORAGES**
- **INFO.KPIS**
- **INFO.LEVELS**
- **INFO.MEASURES**
- **INFO.MODEL**
- **INFO.OBJECTTRANSLATIONS**
- **INFO.PARQUETFILESTORAGES**
- **INFO.PARTITIONS**
- **INFO.PARTITIONSTORAGES**
- **INFO.PERSPECTIVECOLUMNS**
- **INFO.PERSPECTIVEHIERARCHIES**
- **INFO.PERSPECTIVEMEASURES**
- **INFO.PERSPECTIVES**
- **INFO.PERSPECTIVETABLES**
- **INFO.QUERYGROUPS**
- **INFO.REFRESHPOLICIES**
- **INFO.RELATEDCOLUMNDETAILS**
- **INFO.RELATIONSHIPINDEXSTORAGES**
- **INFO.RELATIONSHIPS**
- **INFO.RELATIONSHIPSTORAGES**
- **INFO.ROLEMEMBERSHIPS**
- **INFO.ROLES**
- **INFO.SEGMENTMAPSTORAGES**
- **INFO.SEGMENTSTORAGES**
- **INFO.STORAGEFILES**
- **INFO.STORAGEFOLDERS**
- **INFO.TABLEPERMISSIONS**
- **INFO.TABLES**
- **INFO.TABLESTORAGES**
- **INFO.VARIATIONS.**

In particular:

- **INFO.TABLES** contains information about the tables in the model, such as the table name, description and whether it is hidden or not
- **INFO.COLUMNS** contains information about the columns in a model, such as the column name, data type and whether it is hidden or not
- **INFO.MEASURES** contains information about the measures in the model, such as the measure name, expression and format string.

For example, let's look at **INFO.MEASURES** and demonstrate now it may be joined to other **INFO** functions in DAX query view. Microsoft has a file, **Store Sales** (**PBIX**), that you may download from https://learn.microsoft.com/power-bi/create-reports/sample-datasets#updated-samples if you wish to follow along.

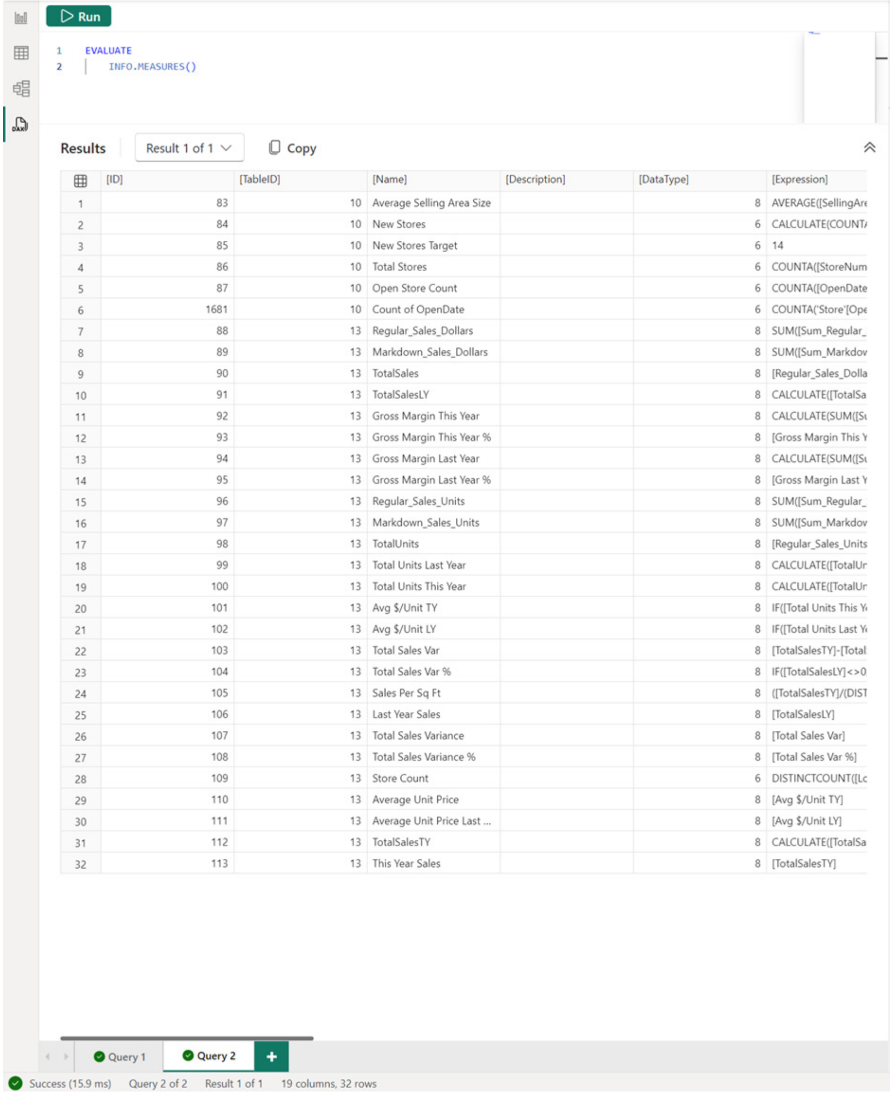
In Power BI Desktop. go to DAX query view and type in **EVALUATE** followed by **INFO** to see them all listed:



```
EVALUATE
    INFO.
```

```
fx INFO.ALTERNATEOFDEFINITIONS
fx INFO.ANNOTATIONS
fx INFO.ATTRIBUTEHIERARCHIES
fx INFO.ATTRIBUTEHIERARCHYSTORAGES
fx INFO.CALCULATIONGROUPS
fx INFO.CALCULATIONITEMS
fx INFO.COLUMNPARTITIONSTORAGES
fx INFO.COLUMNPERMISSIONS
fx INFO.COLUMNS
fx INFO.COLUMNSTORAGES
fx INFO.CULTURES
fx INFO.DATACOVERAGEDEFINITIONS
```

I can continue typing or scroll down to **INFO.MEASURES**. Then I click run or **F5** (now recognised in Power BI). You can also use the maybe lesser-known keyboard shortcut, **CTRL + SHIFT + E**, to run the **DAX** query too.



```
EVALUATE
    INFO.MEASURES()
```

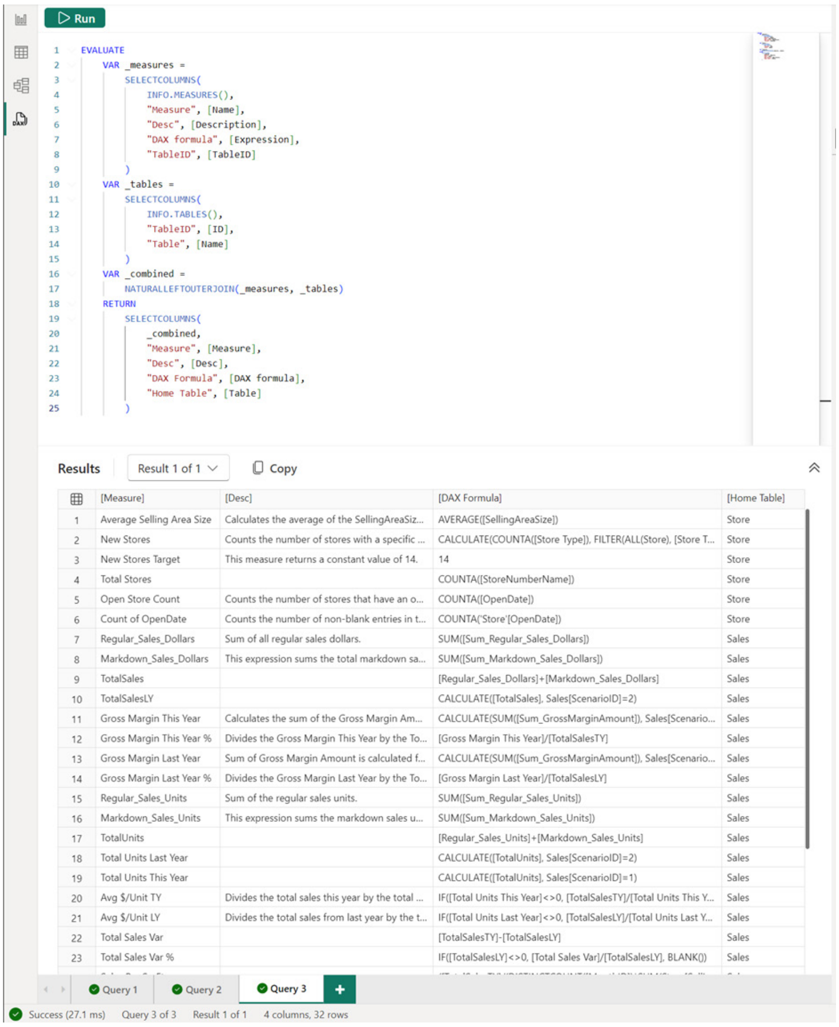
	[ID]	[TableID]	[Name]	[Description]	[DataType]	[Expression]
1	83	10	Average Selling Area Size		8	AVERAGE([SellingAre
2	84	10	New Stores		6	CALCULATE(COUNT/
3	85	10	New Stores Target		6	14
4	86	10	Total Stores		6	COUNTA([StoreNum
5	87	10	Open Store Count		6	COUNTA([OpenDate
6	1681	10	Count of OpenDate		6	COUNTA('Store'[Ope
7	88	13	Regular_Sales_Dollars		8	SUM([Sum_Regular_
8	89	13	Markdown_Sales_Dollars		8	SUM([Sum_Markdov
9	90	13	TotalSales		8	[Regular_Sales_Dolla
10	91	13	TotalSalesLY		8	CALCULATE([TotalSa
11	92	13	Gross Margin This Year		8	CALCULATE(SUM([Su
12	93	13	Gross Margin This Year %		8	[Gross Margin This Y
13	94	13	Gross Margin Last Year		8	CALCULATE(SUM([Su
14	95	13	Gross Margin Last Year %		8	[Gross Margin Last Y
15	96	13	Regular_Sales_Units		8	SUM([Sum_Regular_
16	97	13	Markdown_Sales_Units		8	SUM([Sum_Markdov
17	98	13	TotalUnits		8	[Regular_Sales_Units
18	99	13	Total Units Last Year		8	CALCULATE([TotalUr
19	100	13	Total Units This Year		8	CALCULATE([TotalUr
20	101	13	Avg $/Unit TY		8	IF([Total Units This Y
21	102	13	Avg $/Unit LY		8	IF([Total Units Last Y
22	103	13	Total Sales Var		8	[TotalSalesTY]-[Total
23	104	13	Total Sales Var %		8	IF([TotalSalesLY]<>0
24	105	13	Sales Per Sq Ft		8	([TotalSalesTY]/(DIST
25	106	13	Last Year Sales		8	[TotalSalesLY]
26	107	13	Total Sales Variance		8	[Total Sales Var]
27	108	13	Total Sales Variance %		8	[Total Sales Var %]
28	109	13	Store Count		6	DISTINCTCOUNT([Lc
29	110	13	Average Unit Price		8	[Avg $/Unit TY]
30	111	13	Average Unit Price Last ...		8	[Avg $/Unit LY]
31	112	13	TotalSalesTY		8	CALCULATE([TotalSa
32	113	13	This Year Sales		8	[TotalSalesTY]

Success (15.9 ms)   Query 2 of 2   Result 1 of 1   19 columns, 32 rows

Now I can see all the measures in my model, their expression, and other useful information about them. It also gives me their home table but in maybe a slightly less useful **TableID** column. Thankfully, there is an **INFO.TABLES DAX** function which has the names of tables 10 and 13. To join them, I will use **SELECTCOLUMNS** and **NATURALLEFTOUTERJOIN** shown in the example below.

```
EVALUATE
    VAR _measures =
        SELECTCOLUMNS(
            INFO.MEASURES(),
            "Measure", [Name],
            "Desc", [Description],
            "DAX formula", [Expression],
            "TableID", [TableID]
        )
    VAR _tables =
        SELECTCOLUMNS(
            INFO.TABLES(),
            "TableID", [ID],
            "Table", [Name]
        )
    VAR _combined =
        NATURALLEFTOUTERJOIN(_measures, _tables)
    RETURN
        SELECTCOLUMNS(
            _combined,
            "Measure", [Measure],
            "Desc", [Desc],
            "DAX Formula", [DAX formula],
            "Home Table", [Table]
        )
```

If you run this **DAX** query, the description (when available), **DAX** formulae and the home table of all the measures in the model will be displayed:

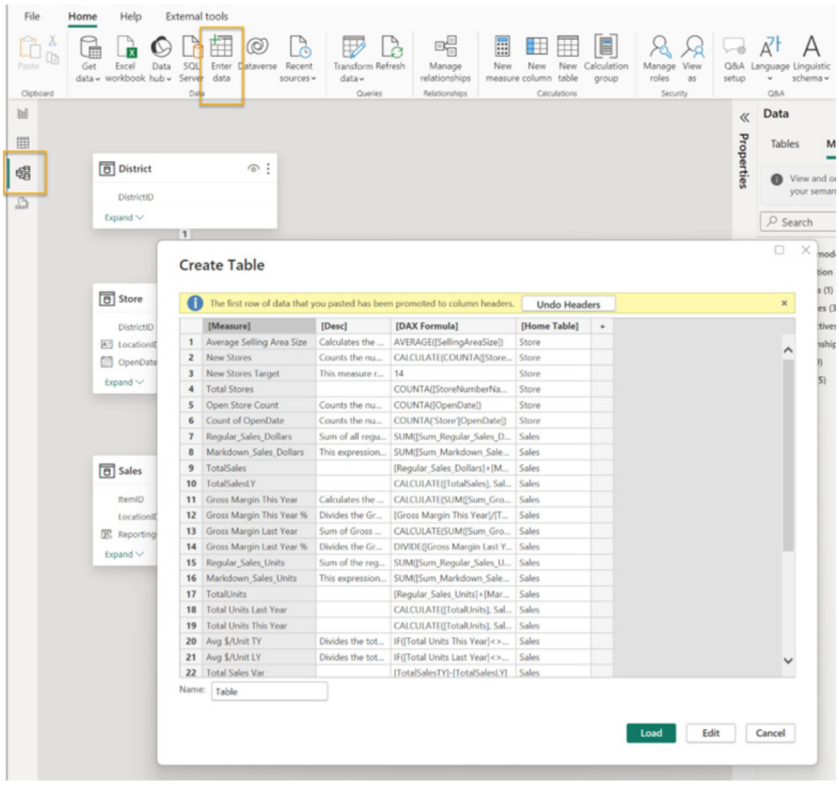


```
EVALUATE
    VAR _measures =
        SELECTCOLUMNS(
            INFO.MEASURES(),
            "Measure", [Name],
            "Desc", [Description],
            "DAX formula", [Expression],
            "TableID", [TableID]
        )
    VAR _tables =
        SELECTCOLUMNS(
            INFO.TABLES(),
            "TableID", [ID],
            "Table", [Name]
        )
    VAR _combined =
        NATURALLEFTOUTERJOIN(_measures, _tables)
    RETURN
        SELECTCOLUMNS(
            _combined,
            "Measure", [Measure],
            "Desc", [Desc],
            "DAX Formula", [DAX formula],
            "Home Table", [Table]
        )
```

	[Measure]	[Desc]	[DAX Formula]	[Home Table]
1	Average Selling Area Size	Calculates the average of the SellingAreaSiz...	AVERAGE([SellingAreaSize])	Store
2	New Stores	Counts the number of stores with a specific ...	CALCULATE(COUNTA([Store Type]), FILTER(ALL(Store), [Store T...	Store
3	New Stores Target	This measure returns a constant value of 14.	14	Store
4	Total Stores		COUNTA([StoreNumberName])	Store
5	Open Store Count	Counts the number of stores that have an o...	COUNTA([OpenDate])	Store
6	Count of OpenDate	Counts the number of non-blank entries in t...	COUNTA('Store'[OpenDate])	Store
7	Regular_Sales_Dollars	Sum of all regular sales dollars.	SUM([Sum_Regular_Sales_Dollars])	Sales
8	Markdown_Sales_Dollars	This expression sums the total markdown sa...	SUM([Sum_Markdown_Sales_Dollars])	Sales
9	TotalSales		[Regular_Sales_Dollars]+[Markdown_Sales_Dollars]	Sales
10	TotalSalesLY		CALCULATE([TotalSales], Sales[ScenarioID]=2)	Sales
11	Gross Margin This Year	Calculates the sum of the Gross Margin Am...	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[Scenario...	Sales
12	Gross Margin This Year %	Divides the Gross Margin This Year by the To...	[Gross Margin This Year]/[TotalSalesTY]	Sales
13	Gross Margin Last Year	Sum of Gross Margin Amount is calculated f...	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[Scenario...	Sales
14	Gross Margin Last Year %	Divides the Gross Margin Last Year by the To...	[Gross Margin Last Year]/[TotalSalesLY]	Sales
15	Regular_Sales_Units	Sum of the regular sales units.	SUM([Sum_Regular_Sales_Units])	Sales
16	Markdown_Sales_Units	This expression sums the markdown sales u...	SUM([Sum_Markdown_Sales_Units])	Sales
17	TotalUnits		[Regular_Sales_Units]+[Markdown_Sales_Units]	Sales
18	Total Units Last Year		CALCULATE([TotalUnits], Sales[ScenarioID]=2)	Sales
19	Total Units This Year		CALCULATE([TotalUnits], Sales[ScenarioID]=1)	Sales
20	Avg $/Unit TY	Divides the total sales this year by the total ...	IF([Total Units This Year]<>0, [TotalSalesTY]/[Total Units This Y...	Sales
21	Avg $/Unit LY	Divides the total sales from last year by the t...	IF([Total Units Last Year]<>0, [TotalSalesLY]/[Total Units Last Y...	Sales
22	Total Sales Var		[TotalSalesTY]-[TotalSalesLY]	Sales
23	Total Sales Var %		IF([TotalSalesLY]<>0, [Total Sales Var]/[TotalSalesLY], BLANK())	Sales

Success (27.1 ms) Query 3 of 3 Result 1 of 1 4 columns, 32 rows

Documenting your model just got a lot easier. You may copy the results as you need with the Copy button. For example, I can go to Model view and choose ‘Enter data’ to Paste them into the model.



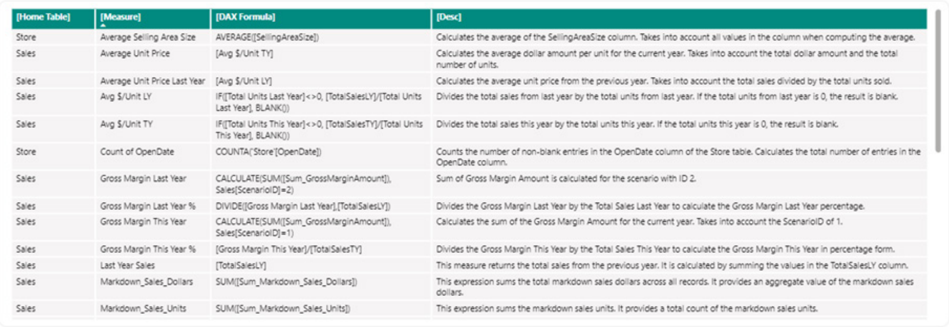
Create Table

The first row of data that you pasted has been promoted to column headers. Undo Headers

	[Measure]	[Desc]	[DAX Formula]	[Home Table]
1	Average Selling Area Size	Calculates the ...	AVERAGE([SellingAreaSize])	Store
2	New Stores	Counts the nu...	CALCULATE(COUNTA([Store...	Store
3	New Stores Target	This measure r...	14	Store
4	Total Stores		COUNTA([StoreNumberNa...	Store
5	Open Store Count	Counts the nu...	COUNTA([OpenDate])	Store
6	Count of OpenDate	Counts the nu...	COUNTA('Store'[OpenDate])	Store
7	Regular_Sales_Dollars	Sum of all regu...	SUM([Sum_Regular_Sales_D...	Sales
8	Markdown_Sales_Dollars	This expression...	SUM([Sum_Markdown_Sale...	Sales
9	TotalSales		[Regular_Sales_Dollars]+[M...	Sales
10	TotalSalesLY		CALCULATE([TotalSales], Sal...	Sales
11	Gross Margin This Year	Calculates the ...	CALCULATE(SUM([Sum_Gro...	Sales
12	Gross Margin This Year %	Divides the Gr...	[Gross Margin This Year]/[T...	Sales
13	Gross Margin Last Year	Sum of Gross ...	CALCULATE(SUM([Sum_Gro...	Sales
14	Gross Margin Last Year %	Divides the Gr...	DIVIDE([Gross Margin Last Y...	Sales
15	Regular_Sales_Units	Sum of the reg...	SUM([Sum_Regular_Sales_U...	Sales
16	Markdown_Sales_Units	This expression...	SUM([Sum_Markdown_Sale...	Sales
17	TotalUnits		[Regular_Sales_Units]+[Mar...	Sales
18	Total Units Last Year		CALCULATE([TotalUnits], Sal...	Sales
19	Total Units This Year		CALCULATE([TotalUnits], Sal...	Sales
20	Avg $/Unit TY	Divides the tot...	IF([Total Units This Year]<>...	Sales
21	Avg $/Unit LY	Divides the tot...	IF([Total Units Last Year]<>...	Sales
22	Total Sales Var		[TotalSalesTY]-[TotalSalesLY]	Sales

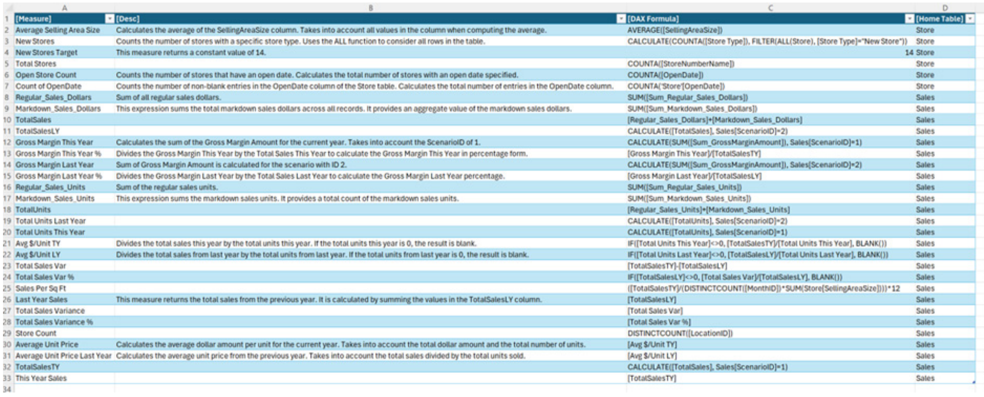
Name: Table

I can then add a Report page with a visual showing information about this model:



[Home Table]	[Measure]	[DAX Formula]	[Desc]
Store	Average Selling Area Size	AVERAGE([SellingAreaSize])	Calculates the average of the SellingAreaSize column. Takes into account all values in the column when computing the average.
Sales	Average Unit Price	[Avg $/Unit TY]	Calculates the average dollar amount per unit for the current year. Takes into account the total dollar amount and the total number of units.
Sales	Average Unit Price Last Year	[Avg $/Unit LY]	Calculates the average unit price from the previous year. Takes into account the total sales divided by the total units sold.
Sales	Avg $/Unit LY	IF([Total Units Last Year]<>0, [TotalSalesLY]/[Total Units Last Year], BLANK())	Divides the total sales from last year by the total units from last year. If the total units from last year is 0, the result is blank.
Sales	Avg $/Unit TY	IF([Total Units This Year]<>0, [TotalSalesTY]/[Total Units This Year], BLANK())	Divides the total sales this year by the total units this year. If the total units this year is 0, the result is blank.
Store	Count of OpenDate	COUNTA('Store'[OpenDate])	Counts the number of non-blank entries in the OpenDate column of the Store table. Calculates the total number of entries in the OpenDate column.
Sales	Gross Margin Last Year	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=2)	Sum of Gross Margin Amount is calculated for the scenario with ID 2.
Sales	Gross Margin Last Year %	DIVIDE([Gross Margin Last Year],[TotalSalesLY])	Divides the Gross Margin Last Year by the Total Sales Last Year to calculate the Gross Margin Last Year percentage.
Sales	Gross Margin This Year	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=1)	Calculates the sum of the Gross Margin Amount for the current year. Takes into account the ScenarioID of 1.
Sales	Gross Margin This Year %	[Gross Margin This Year]/[TotalSalesTY]	Divides the Gross Margin This Year by the Total Sales This Year to calculate the Gross Margin This Year in percentage form.
Sales	Last Year Sales	[TotalSalesLY]	This measure returns the total sales from the previous year. It is calculated by summing the values in the TotalSalesLY column.
Sales	Markdown_Sales_Dollars	SUM([Sum_Markdown_Sales_Dollars])	This expression sums the total markdown sales dollars across all records. It provides an aggregate value of the markdown sales dollars.
Sales	Markdown_Sales_Units	SUM([Sum_Markdown_Sales_Units])	This expression sums the markdown sales units. It provides a total count of the markdown sales units.

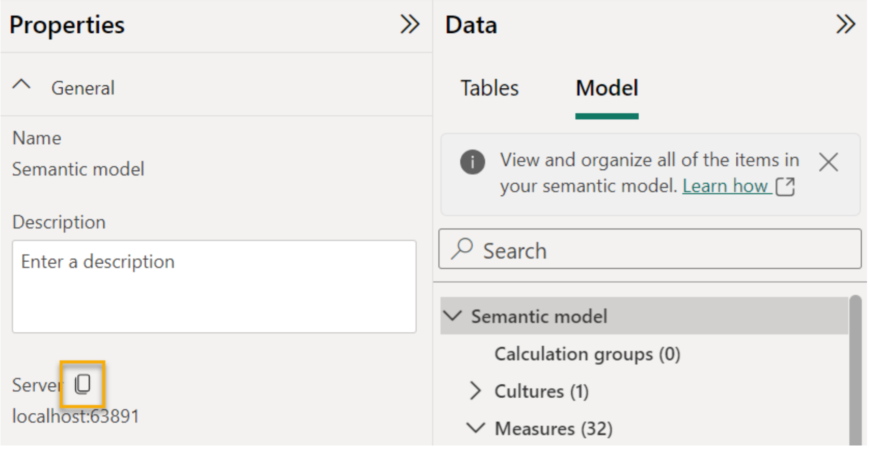
Here, I have pasted them into Excel and made a Table with **CTRL + T**:



[Measure]	[Desc]	[DAX Formula]	[Home Table]
Average Selling Area Size	Calculates the average of the SellingAreaSize column. Takes into account all values in the column when computing the average.	AVERAGE([SellingAreaSize])	Store
New Stores	Counts the number of stores with a specific store type. Uses the ALL function to consider all rows in the table.	CALCULATE(COUNTA([Store Type]), FILTER(ALL(Store), [Store Type]="New Store"))	Store
New Stores Target	This measure returns a constant value of 14.	14	Store
Total Stores		COUNTA([StoreNumberName])	Store
Open Store Count	Counts the number of stores that have an open date. Calculates the total number of stores with an open date specified.	COUNTA([OpenDate])	Store
Count of OpenDate	Counts the number of non-blank entries in the OpenDate column of the Store table. Calculates the total number of entries in the OpenDate column.	COUNTA('Store'[OpenDate])	Store
Regular_Sales_Dollars	Sum of all regular sales dollars.	SUM([Sum_Regular_Sales_Dollars])	Sales
Markdown_Sales_Dollars	This expression sums the total markdown sales dollars across all records. It provides an aggregate value of the markdown sales dollars.	SUM([Sum_Markdown_Sales_Dollars])	Sales
TotalSales		[Regular_Sales_Dollars]+[Markdown_Sales_Dollars]	Sales
TotalSalesLY		CALCULATE([TotalSales], Sales[ScenarioID]=2)	Sales
Gross Margin This Year	Calculates the sum of the Gross Margin Amount for the current year. Takes into account the ScenarioID of 1.	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=1)	Sales
Gross Margin This Year %	Divides the Gross Margin This Year by the Total Sales This Year to calculate the Gross Margin This Year in percentage form.	[Gross Margin This Year]/[TotalSalesTY]	Sales
Gross Margin Last Year	Sum of Gross Margin Amount is calculated for the scenario with ID 2.	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=2)	Sales
Gross Margin Last Year %	Divides the Gross Margin Last Year by the Total Sales Last Year to calculate the Gross Margin Last Year percentage.	[Gross Margin Last Year]/[TotalSalesLY]	Sales
Regular_Sales_Units	Sum of the regular sales units.	SUM([Sum_Regular_Sales_Units])	Sales
Markdown_Sales_Units	This expression sums the markdown sales units. It provides a total count of the markdown sales units.	SUM([Sum_Markdown_Sales_Units])	Sales
TotalUnits		[Regular_Sales_Units]+[Markdown_Sales_Units]	Sales
Total Units Last Year		CALCULATE([TotalUnits], Sales[ScenarioID]=2)	Sales
Total Units This Year		CALCULATE([TotalUnits], Sales[ScenarioID]=1)	Sales
Avg $/Unit TY	Divides the total sales this year by the total units this year. If the total units this year is 0, the result is blank.	IF([Total Units This Year]<>0, [TotalSalesTY]/[Total Units This Year], BLANK())	Sales
Avg $/Unit LY	Divides the total sales from last year by the total units from last year. If the total units from last year is 0, the result is blank.	IF([Total Units Last Year]<>0, [TotalSalesLY]/[Total Units Last Year], BLANK())	Sales
Total Sales Var		[TotalSalesTY]-[TotalSalesLY]	Sales
Total Sales Var %		IF([TotalSalesLY]<>0, [Total Sales Var]/[TotalSalesLY], BLANK())	Sales
Sales Per Sq Ft		([TotalSalesTY]/(DISTINCTCOUNT([MonthID])*SUM(Store[SellingAreaSize])))*12	Sales
Last Year Sales	This measure returns the total sales from the previous year. It is calculated by summing the values in the TotalSalesLY column.	[TotalSalesLY]	Sales
Total Sales Variance		[Total Sales Var]	Sales
Total Sales Variance %		[Total Sales Var %]	Sales
Store Count		DISTINCTCOUNT([LocationID])	Sales
Average Unit Price	Calculates the average dollar amount per unit for the current year. Takes into account the total dollar amount and the total number of units.	[Avg $/Unit TY]	Sales
Average Unit Price Last Year	Calculates the average unit price from the previous year. Takes into account the total sales divided by the total units sold.	[Avg $/Unit LY]	Sales
TotalSalesTY		CALCULATE([TotalSales], Sales[ScenarioID]=1)	Sales
This Year Sales		[TotalSalesTY]	Sales

It should be noted that copy then paste may not work well if you have **DAX** formulae on multiple lines in your measures. In that case, you may run the **DAX** query in Excel directly:

- Open Excel and navigate to the Data tab
- Click 'Get Data'
- Click 'Database' in the drop down, then 'SQL Server Analysis Services' (not the 'Analysis Services' one)
- In Power BI Desktop, go to Model view, then go to the **Data pane -> Model**, and click the **Semantic model**. If you don't see this, ensure 'Model Explorer' is enabled in your Preview features section of Options
- In the Properties pane, click the copy icon next to Server to add it to your clipboard (another new feature)



Properties

General

Name
Semantic model

Description
Enter a description

Server
localhost:63891

Data

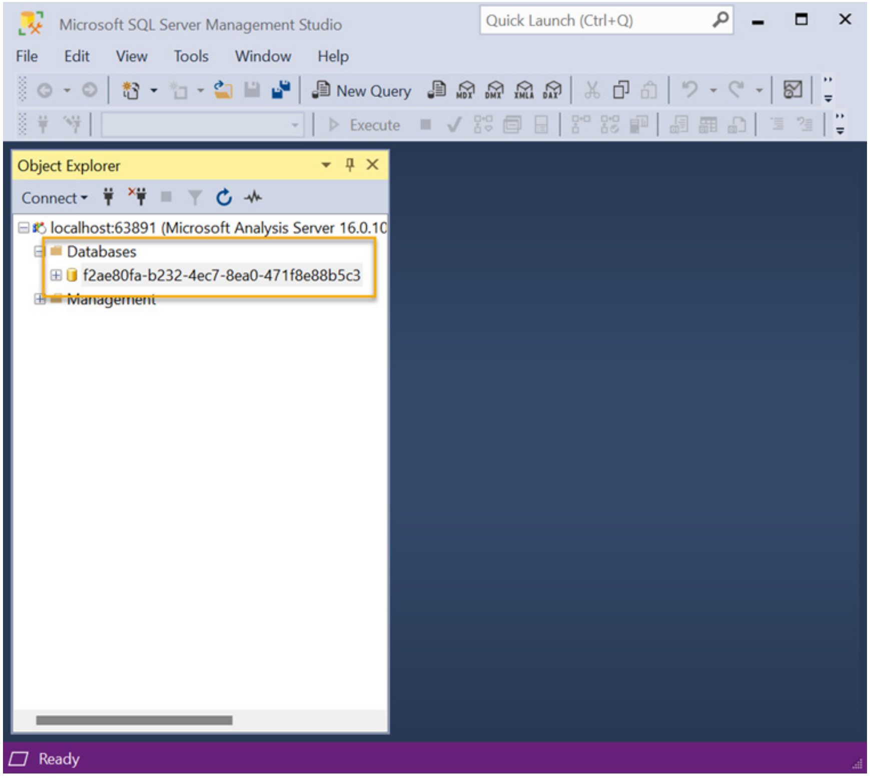
Tables Model

View and organize all of the items in your semantic model. Learn how

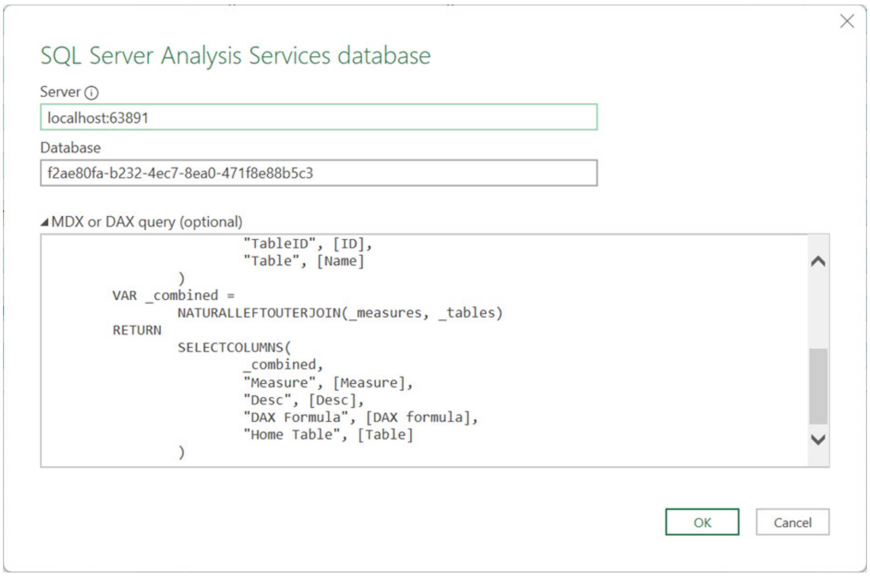
Search

Semantic model
Calculation groups (0)
Cultures (1)
Measures (32)

- Returning to Excel, paste the **localhost:xxxxx** (most likely it will be a different number for you) into the Server
- You also need the Database, which is not in the Properties pane yet with Server (planned). You can get the Database ID from an external tool, such as DAX Studio, or another free Microsoft tool, SQL Server Management Studio (change Server type to Analysis Services). The database ID is not the name of the **PBIX** file, but instead a GUID. If you have previously published the **PBIX** to the Power BI Service, the Database GUID is also in the URL of the published version


Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File
Edit
View
Tools
Window
Help
New Query
Execute
Object Explorer
Connect
localhost:63891 (Microsoft Analysis Server 16.0.10
Databases
f2ae80fa-b232-4ec7-8ea0-471f8e88b5c3
Management
Ready

- Back in Excel, copy in the **DAX** query to the **DAX** or **MDX** query box and click OK



SQL Server Analysis Services database

Server ⓘ

localhost:63891

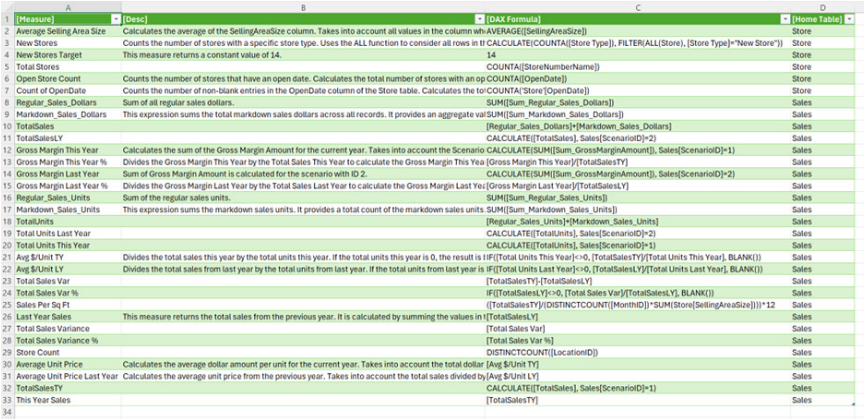
Database

f2ae80fa-b232-4ec7-8ea0-471f8e88b5c3

◢ MDX or DAX query (optional)

```
                "TableID", [ID],
                "Table", [Name]
        )
    VAR _combined =
        NATURALLEFTOUTERJOIN(_measures, _tables)
    RETURN
        SELECTCOLUMNS(
                _combined,
                "Measure", [Measure],
                "Desc", [Desc],
                "DAX Formula", [DAX formula],
                "Home Table", [Table]
        )
```

- Click Load
- The data will show in an Excel Table.



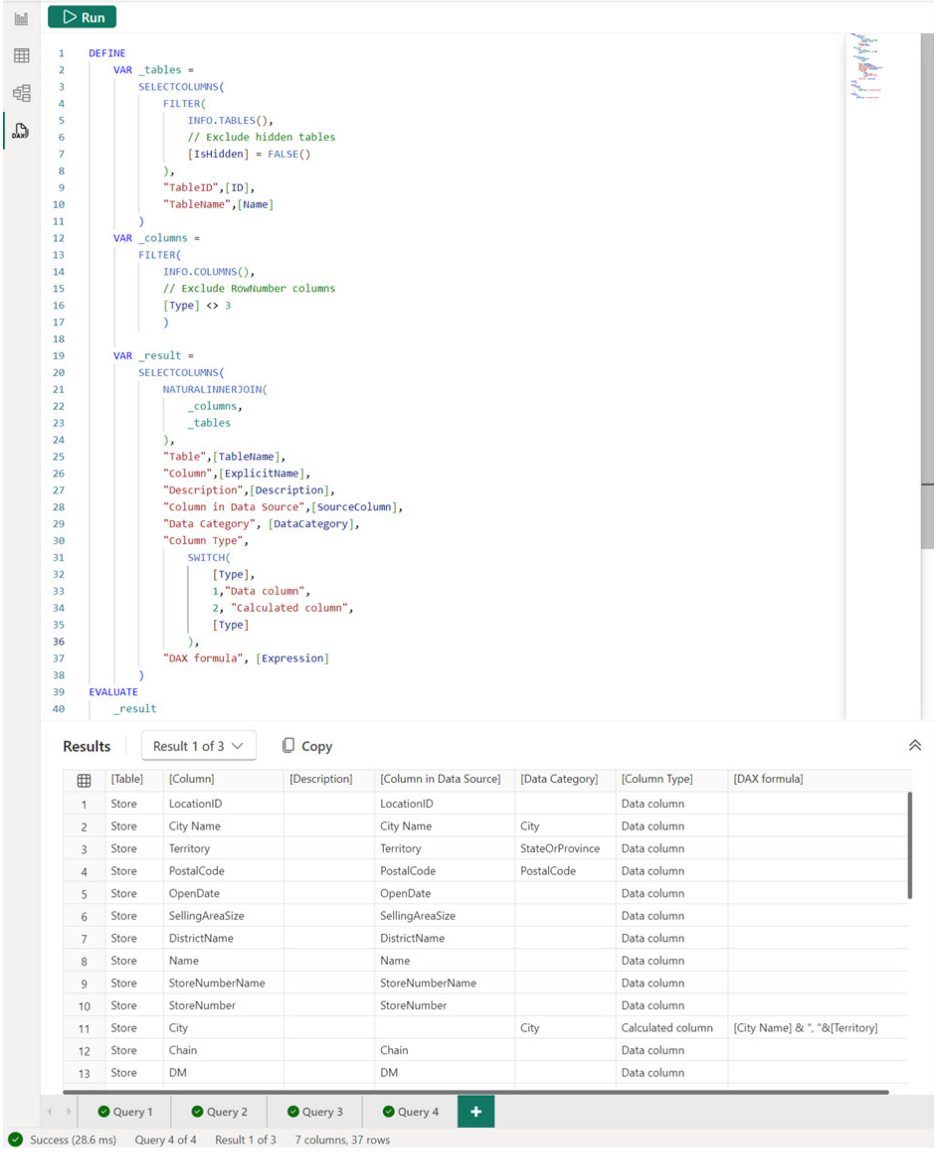
	A	B	C	D
1	[Measure]	[Desc]	[DAX Formula]	[Home Table]
2	Average Selling Area Size	Calculates the average of the SellingAreaSize column. Takes into account all values in the column wh	AVERAGE([SellingAreaSize])	Store
3	New Stores	Counts the number of stores with a specific store type. Uses the ALL function to consider all rows in tl	CALCULATE(COUNTA([Store Type]), FILTER(ALL(Store), [Store Type]="New Store"))	Store
4	New Stores Target	This measure returns a constant value of 14.	14	Store
5	Total Stores		COUNTA([StoreNumberName])	Store
6	Open Store Count	Counts the number of stores that have an open date. Calculates the total number of stores with an op	COUNTA([OpenDate])	Store
7	Count of OpenDate	Counts the number of non-blank entries in the OpenDate column of the Store table. Calculates the tot	COUNTA('Store'[OpenDate])	Store
8	Regular_Sales_Dollars	Sum of all regular sales dollars.	SUM([Sum_Regular_Sales_Dollars])	Sales
9	Markdown_Sales_Dollars	This expression sums the total markdown sales dollars across all records. It provides an aggregate val	SUM([Sum_Markdown_Sales_Dollars])	Sales
10	TotalSales		[Regular_Sales_Dollars]+[Markdown_Sales_Dollars]	Sales
11	TotalSalesLY		CALCULATE([TotalSales], Sales[ScenarioID]=2)	Sales
12	Gross Margin This Year	Calculates the sum of the Gross Margin Amount for the current year. Takes into account the Scenario	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=1)	Sales
13	Gross Margin This Year %	Divides the Gross Margin This Year by the Total Sales This Year to calculate the Gross Margin This Yea	[Gross Margin This Year]/[TotalSalesTY]	Sales
14	Gross Margin Last Year	Sum of Gross Margin Amount is calculated for the scenario with ID 2.	CALCULATE(SUM([Sum_GrossMarginAmount]), Sales[ScenarioID]=2)	Sales
15	Gross Margin Last Year %	Divides the Gross Margin Last Year by the Total Sales Last Year to calculate the Gross Margin Last Yea	[Gross Margin Last Year]/[TotalSalesLY]	Sales
16	Regular_Sales_Units	Sum of the regular sales units.	SUM([Sum_Regular_Sales_Units])	Sales
17	Markdown_Sales_Units	This expression sums the markdown sales units. It provides a total count of the markdown sales units.	SUM([Sum_Markdown_Sales_Units])	Sales
18	TotalUnits		[Regular_Sales_Units]+[Markdown_Sales_Units]	Sales
19	Total Units Last Year		CALCULATE([TotalUnits], Sales[ScenarioID]=2)	Sales
20	Total Units This Year		CALCULATE([TotalUnits], Sales[ScenarioID]=1)	Sales
21	Avg $/Unit TY	Divides the total sales this year by the total units this year. If the total units this year is 0, the result is t	IF([Total Units This Year]<>0, [TotalSalesTY]/[Total Units This Year], BLANK())	Sales
22	Avg $/Unit LY	Divides the total sales from last year by the total units from last year. If the total units from last year is	IF([Total Units Last Year]<>0, [TotalSalesLY]/[Total Units Last Year], BLANK())	Sales
23	Total Sales Var		[TotalSalesTY]-[TotalSalesLY]	Sales
24	Total Sales Var %		IF([TotalSalesLY]<>0, [Total Sales Var]/[TotalSalesLY], BLANK())	Sales
25	Sales Per Sq Ft		([TotalSalesTY]/(DISTINCTCOUNT([MonthID])*SUM(Store[SellingAreaSize])))*12	Sales
26	Last Year Sales	This measure returns the total sales from the previous year. It is calculated by summing the values in t	[TotalSalesLY]	Sales
27	Total Sales Variance		[Total Sales Var]	Sales
28	Total Sales Variance %		[Total Sales Var %]	Sales
29	Store Count		DISTINCTCOUNT([LocationID])	Sales
30	Average Unit Price	Calculates the average dollar amount per unit for the current year. Takes into account the total dollar	[Avg $/Unit TY]	Sales
31	Average Unit Price Last Year	Calculates the average unit price from the previous year. Takes into account the total sales divided by	[Avg $/Unit LY]	Sales
32	TotalSalesTY		CALCULATE([TotalSales], Sales[ScenarioID]=1)	Sales
33	This Year Sales		[TotalSalesTY]	Sales
34				

What is useful about this approach is you can now refresh it, if you make an update to your model during this editing session. When you close Power BI Desktop it will not refresh. However, if you re-open Power BI Desktop, you can simply edit the connection with the new Server to refresh it while Power BI Desktop is running. Also, once published to the Power BI Service you may edit the connection's Server to be the Workspace connection and refresh it too. To change the Server, right-click the table, click on Table, then Edit Query. A familiar Power Query window will open, and you can update the Server and Database there. You can do this with any of your **DAX** queries, not just these new **INFO DAX** functions.

Another combination that may be useful for the new **INFO DAX** functions is to explore the columns in your model.

```
DEFINE
    VAR _tables =
        SELECTCOLUMNS(
            FILTER(
                INFO.TABLES(),
                // Exclude hidden tables
                [IsHidden] = FALSE()
            ),
            "TableID",[ID],
            "TableName",[Name]
        )
    VAR _columns =
        FILTER(
            INFO.COLUMNS(),
            // Exclude RowNumber columns
            [Type] <> 3
            )

    VAR _result =
        SELECTCOLUMNS(
            NATURALINNERJOIN(
                _columns,
                _tables
            ),
            "Table",[TableName],
            "Column",[ExplicitName],
            "Description",[Description],
            "Column in Data Source",[SourceColumn],
            "Data Category", [DataCategory],
            "Column Type",
                SWITCH(
                    [Type],
                    1,"Data column",
                    2, "Calculated column",
                    [Type]
                ),
            "DAX formula", [Expression]
        )
EVALUATE
    _result
```

When I run this **DAX** query, I can see information about all the columns in my model. I can see if the name is different than the source name, if they have a data category and even the **DA**X formula for calculated columns.



```
DEFINE
    VAR _tables =
        SELECTCOLUMNS(
            FILTER(
                INFO.TABLES(),
                // Exclude hidden tables
                [IsHidden] = FALSE()
            ),
            "TableID",[ID],
            "TableName",[Name]
        )
    VAR _columns =
        FILTER(
            INFO.COLUMNS(),
            // Exclude RowNumber columns
            [Type] <> 3
            )

    VAR _result =
        SELECTCOLUMNS(
            NATURALINNERJOIN(
                _columns,
                _tables
            ),
            "Table",[TableName],
            "Column",[ExplicitName],
            "Description",[Description],
            "Column in Data Source",[SourceColumn],
            "Data Category", [DataCategory],
            "Column Type",
                SWITCH(
                    [Type],
                    1,"Data column",
                    2, "Calculated column",
                    [Type]
                ),
            "DAX formula", [Expression]
        )
EVALUATE
    _result
```

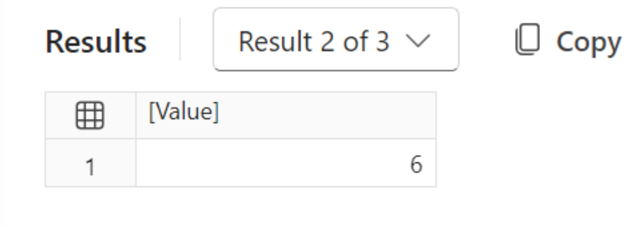
⊞	[Table]	[Column]	[Description]	[Column in Data Source]	[Data Category]	[Column Type]	[DAX formula]
1	Store	LocationID		LocationID		Data column	
2	Store	City Name		City Name	City	Data column	
3	Store	Territory		Territory	StateOrProvince	Data column	
4	Store	PostalCode		PostalCode	PostalCode	Data column	
5	Store	OpenDate		OpenDate		Data column	
6	Store	SellingAreaSize		SellingAreaSize		Data column	
7	Store	DistrictName		DistrictName		Data column	
8	Store	Name		Name		Data column	
9	Store	StoreNumberName		StoreNumberName		Data column	
10	Store	StoreNumber		StoreNumber		Data column	
11	Store	City			City	Calculated column	[City Name] & ", "&[Territory]
12	Store	Chain		Chain		Data column	
13	Store	DM		DM		Data column	

Success (28.6 ms)   Query 4 of 4   Result 1 of 3   7 columns, 37 rows

I can take this even further. I am concerned about the number of calculated columns, so I want to see how many there are and then show only calculated columns. I can do this by adding this to my **DAX** query:

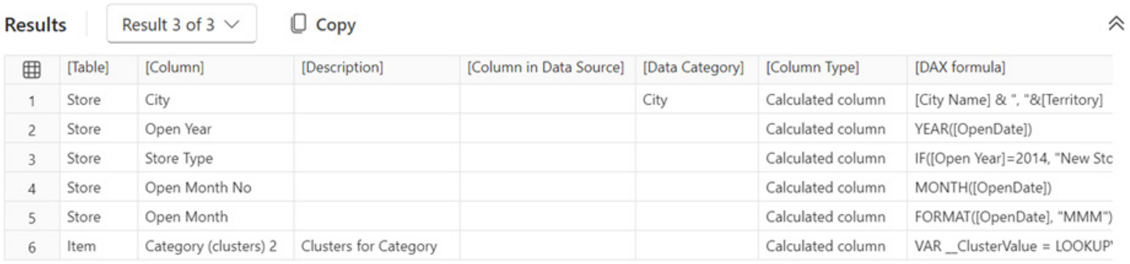
```
EVALUATE
    {COUNTROWS(
        FILTER(
            _result,
            [Column Type] = "Calculated column")
    )}


EVALUATE
    FILTER(
        _result,
        [Column Type] = "Calculated column")
```

Changing the **DAX** query view result grid to the second result, I can see there are six [6] calculated columns:



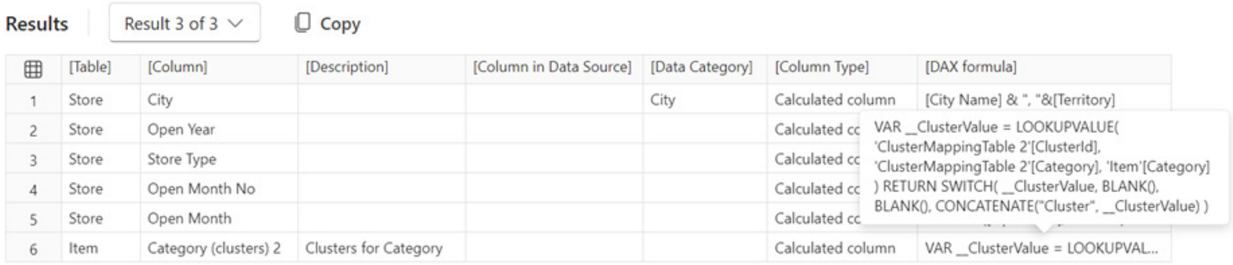
	[Value]
1	6

It should be noted that the third result shows just the calculated columns:



	[Table]	[Column]	[Description]	[Column in Data Source]	[Data Category]	[Column Type]	[DAX formula]
1	Store	City			City	Calculated column	[City Name] & ", "&[Territory]
2	Store	Open Year				Calculated column	YEAR([OpenDate])
3	Store	Store Type				Calculated column	IF([Open Year]=2014, "New Stc
4	Store	Open Month No				Calculated column	MONTH([OpenDate])
5	Store	Open Month				Calculated column	FORMAT([OpenDate], "MMM")
6	Item	Category (clusters) 2	Clusters for Category			Calculated column	VAR _ClusterValue = LOOKUP'

If you are wondering what the **DAX** formula for Category (clusters) 2 is, simply hover over it to see the full **DAX** formula:



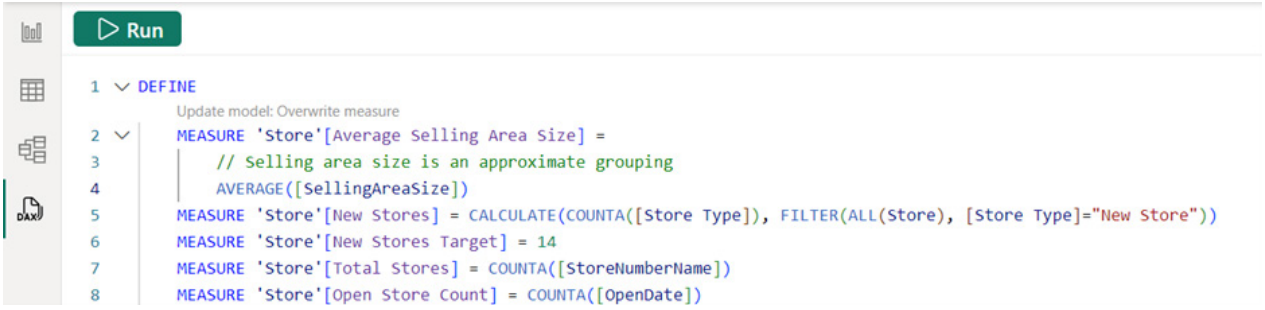
	[Table]	[Column]	[Description]	[Column in Data Source]	[Data Category]	[Column Type]	[DAX formula]
1	Store	City			City	Calculated column	[City Name] & ", "&[Territory]
2	Store	Open Year				Calculated cc	
3	Store	Store Type				Calculated cc	
4	Store	Open Month No				Calculated cc	
5	Store	Open Month				Calculated cc	
6	Item	Category (clusters) 2	Clusters for Category			Calculated column	VAR _ClusterValue = LOOKUPVAL...

```
VAR __ClusterValue = LOOKUPVALUE(
'ClusterMappingTable 2'[ClusterId],
'ClusterMappingTable 2'[Category], 'Item'[Category]
) RETURN SWITCH( __ClusterValue, BLANK(),
BLANK(), CONCATENATE("Cluster", __ClusterValue) )
```

As you can see, there are many new ways to explore the metadata of your model with these **INFO DAX** functions.

You cannot use these functions in calculated tables, calculated columns and measures. They will show an error if you try. Do note that the **INFO** functions will show in the **DAX** Formula bar IntelliSense, but they may result in an error when used. If you use them in the **DEFINE MEASURE**, DAX query view will currently still show the Update model CodeLens, but it will result in an error when clicked. These measures can still be run as **DAX** query in DAX query view without error.

To end, you should also notice that there is an update to the Update model CodeLens. Comments are now kept when using the CodeLens and the CodeLens will show when the **DAX** formula formatting has been changed. For example, when you use 'Quick queries' to 'Evaluate and define' a measure, then click 'Format query', you will see this update. Now, those with new formatting will show a CodeLens to Update model to quickly format your measure **DAX** formulae.



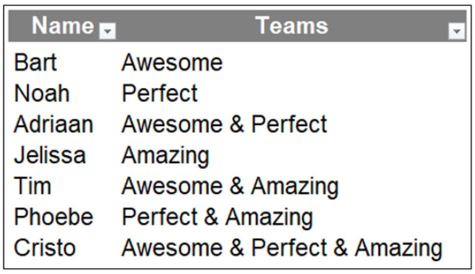
```
DEFINE
    Update model: Overwrite measure
    MEASURE 'Store'[Average Selling Area Size] =
        // Selling area size is an approximate grouping
        AVERAGE([SellingAreaSize])
    MEASURE 'Store'[New Stores] = CALCULATE(COUNTA([Store Type]), FILTER(ALL(Store), [Store Type]="New Store"))
    MEASURE 'Store'[New Stores Target] = 14
    MEASURE 'Store'[Total Stores] = COUNTA([StoreNumberName])
    MEASURE 'Store'[Open Store Count] = COUNTA([OpenDate])
```

Beat the Boredom Challenge

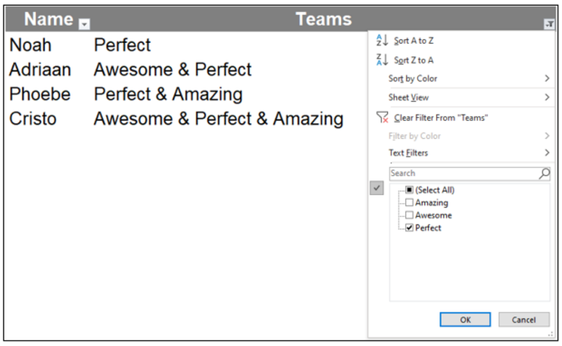
With many of us currently "working from home" / quarantined, there are only so Zoom / Teams calls and virtual parties you can make before you reach your (data) limit. Perhaps they should measure data allowance in blood pressure millimetres of mercury (mmHg). To try and keep our readers engaged, we will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

Imagine that you need to filter out rows in a table that have specific keywords within the text strings contained therein. Manually filtering every single keyword and copying / pasting them to a new location can be a tedious and time-consuming process. To address this, we challenge you to develop a user-friendly solution that allows users to select the desired keywords and return a list having all the text strings associated with those keywords.

Your aim is to create a filter using the keywords "Awesome", "Amazing", and "Perfect" as filter criteria, as shown in the picture below:



Name	Teams
Bart	Awesome
Noah	Perfect
Adriaan	Awesome & Perfect
Jelissa	Amazing
Tim	Awesome & Amazing
Phoebe	Perfect & Amazing
Cristo	Awesome & Perfect & Amazing



Name	Teams
Noah	Perfect
Adriaan	Awesome & Perfect
Phoebe	Perfect & Amazing
Cristo	Awesome & Perfect & Amazing

As always, there are some requirements:

- no Power Query / Get & Transform or VBA is allowed
- the formula(e) should be dynamic so that they shall update when a new entry is added.

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

Excel for Mac

With Steve Kraynak joining the team, we thought we would exploit his knowledge and recant all about Microsoft Excel for Mac. Each month, we'll cover a different topic to help you understand how Excel for Mac is different than Excel for Windows. This month, we show a few differences between the experience of writing formulae on a Mac versus a PC.

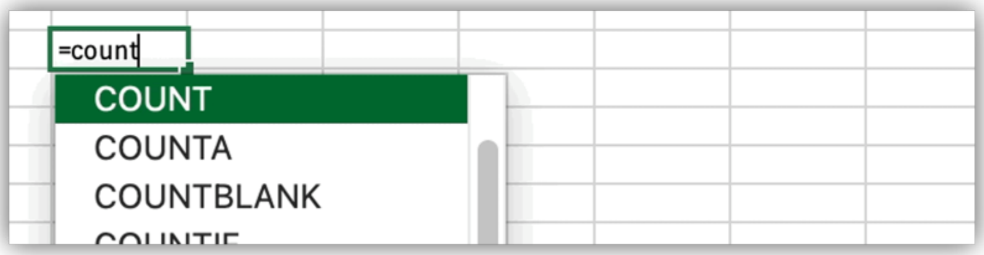
Thankfully, your formulae will give you precisely the same result on a Mac as they do on Windows. You don't need to worry about this. However, there are some minor differences in how Excel helps you to write and edit your formulae.

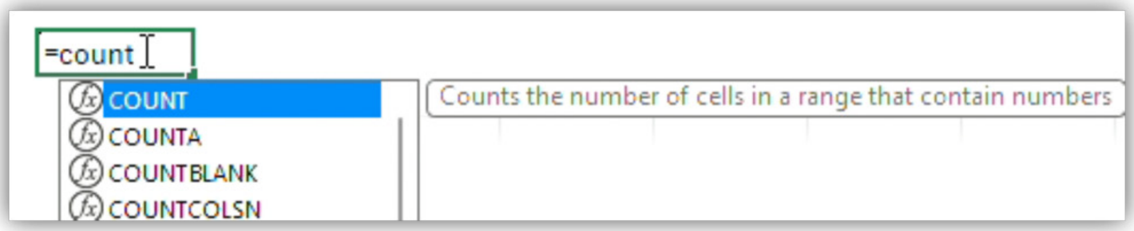
[It should be noted that there are a few special functions that only work on Windows.]

Function ToolTips

When writing a formula, you can begin typing the name of the function you'd like to use, and Excel will show you a list of functions that match. You can use your arrow keys to select a function from the list, and if you're using Windows, you'll see a helpful ToolTip to show you a short description of the function. Unfortunately, the ToolTip won't appear on a Mac.

As you can see in the examples below, a ToolTip helps you understand the difference between the **COUNT** and **COUNTA** functions. To determine this on a Mac, you can use the Formula Builder, which we'll show next.





=count

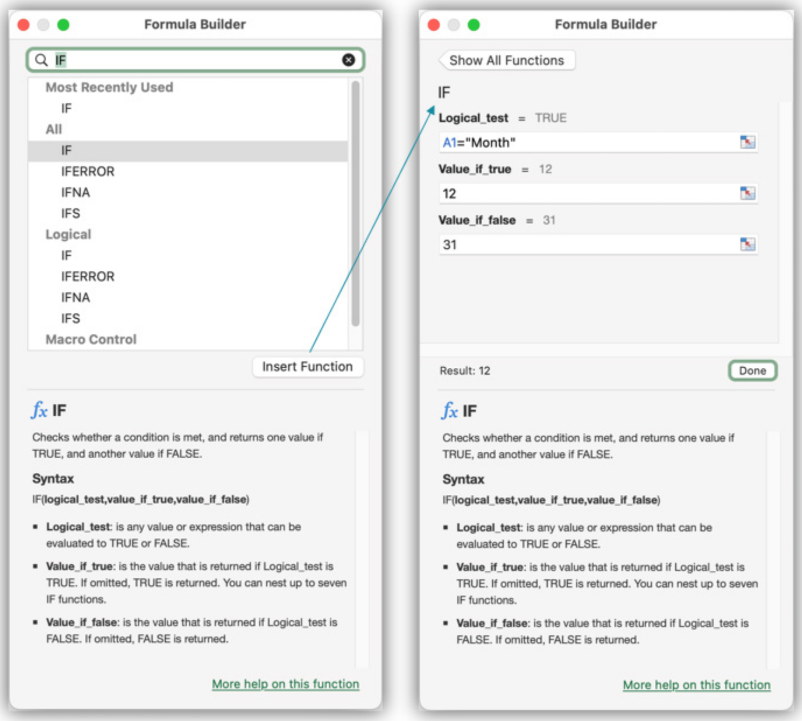
Counts the number of cells in a range that contain numbers

Formula Builder

Excel for Mac has a 'Formula Builder' pane to help you create formulae. This is different than Windows, which has a modal dialog for the same purpose.

It's a two-step process for a new formula or a single step if you're editing a formula. To open it, just click the **Fx** button next to the Formula bar or press **SHIFT + F3**. At first, it shows the function list. You can start typing in the search field and then select a function from the list to see the function's description and syntax, including the descriptions for each parameter.

Once you've found the function you want, press the 'Insert Function' button. The function signature will appear in the pane, showing you the parameters with text fields where you can enter the appropriate references, functions, and / or values. When you're finished, press the 'Done' button and it will insert the formula into the active cell.



Formula Builder

IF

Most Recently Used
- IF

All
- IF
- IFERROR
- IFNA
- IFS

Logical
- IF
- IFERROR
- IFNA
- IFS

Macro Control

Insert Function

fx **IF**

Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.

Syntax

IF(**logical_test,value_if_true,value_if_false**)

- **Logical_test**: is any value or expression that can be evaluated to TRUE or FALSE.
- **Value_if_true**: is the value that is returned if Logical_test is TRUE. If omitted, TRUE is returned. You can nest up to seven IF functions.
- **Value_if_false**: is the value that is returned if Logical_test is FALSE. If omitted, FALSE is returned.

More help on this function

Formula Builder

Show All Functions

IF

Logical_test = TRUE

A1="Month"

Value_if_true = 12

12

Value_if_false = 31

31

Result: 12

Done

fx **IF**

Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.

Syntax

IF(**logical_test,value_if_true,value_if_false**)

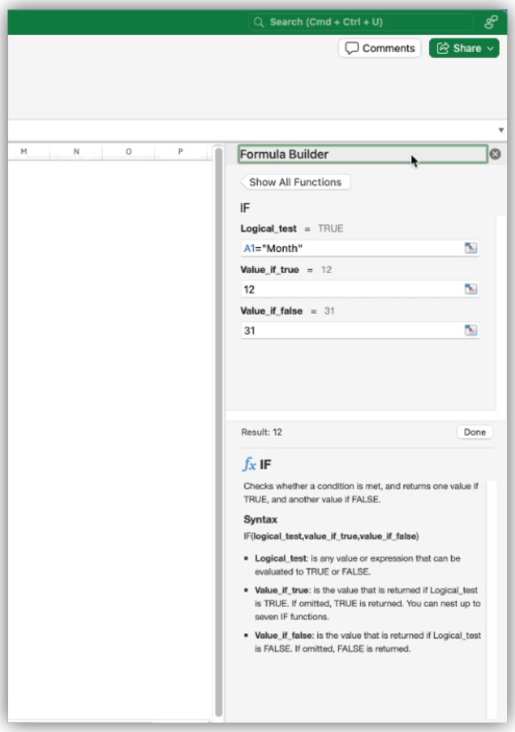
- **Logical_test**: is any value or expression that can be evaluated to TRUE or FALSE.
- **Value_if_true**: is the value that is returned if Logical_test is TRUE. If omitted, TRUE is returned. You can nest up to seven IF functions.
- **Value_if_false**: is the value that is returned if Logical_test is FALSE. If omitted, FALSE is returned.

More help on this function

You can use the 'Formula Builder' to write and edit, but also to help debug your formulae. When you've entered enough information for Excel to start calculating, it will show you the current values above each field in the pane. Notice in the example shown above, the '**Logical_test**' parameter has a value of TRUE. This is the current calculation for **A1**="Month". It also shows the result of the entire formula, if it can be calculated. Again, in the example above, you can see 'Result: 12' near the 'Done' button. This is the current value being calculated by the formula that's being edited.

Positioning the Formula Builder

It doesn't have to be a pane – or a pain. The 'Formula Builder' on a Mac will open as a side pane, but you can tear it away from the side and use it as a dialog if you like. This gives you the flexibility to position it anywhere you like. Move it back to the side if you want to dock it later.



Formula Builder

Show All Functions

IF

Logical_test = TRUE

A1="Month"

Value_if_true = 12

12

Value_if_false = 31

31

Result: 12

Done

fx **IF**

Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.

Syntax

IF(**logical_test,value_if_true,value_if_false**)

- **Logical_test**: is any value or expression that can be evaluated to TRUE or FALSE.
- **Value_if_true**: is the value that is returned if Logical_test is TRUE. If omitted, TRUE is returned. You can nest up to seven IF functions.
- **Value_if_false**: is the value that is returned if Logical_test is FALSE. If omitted, FALSE is returned.

Check back for more details about Excel for Mac and how it's different than Excel for Windows.

[It should be noted that the **ENCODEURL()**, **FILTERXML()** and **WEBSERVICE()** functions rely on operating system capability on Windows, so they won't work on a Mac. You'll see an *#NAME?* error and the Formula bar will show the function names with a '**_xlfn.**' prefix.

We'll continue next month…

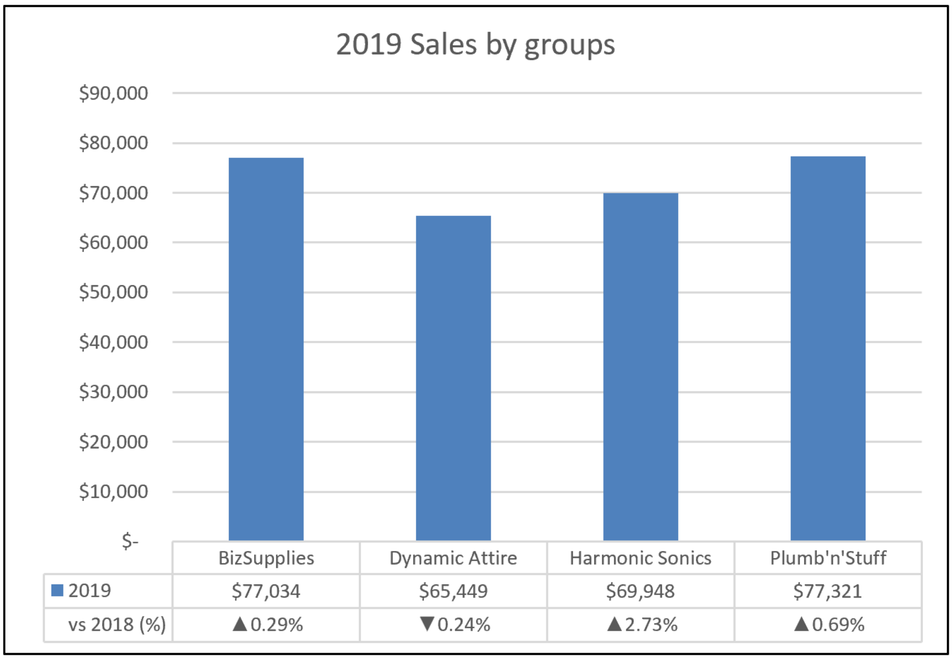
Charts and Dashboards

It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we will consider how to link charts from Excel to a presentation in PowerPoint.

There is a straightforward way of exporting information from Excel to PowerPoint. This method will allow you to link your Excel image in PowerPoint back to the source file.

It is noted that there is a PowerPoint option for a slide layout containing text and a chart object. This is not suitable for pasting an existing Excel chart and is only necessary if you wish to use Microsoft Chart for building a new graph from scratch. We will not consider that option here.

Instead, let's consider an option that will update automatically, to allow for new categories, changes in numbers or even a change of chart type. The first thing to do is to set up our chart in Excel. An example could look something similar to the following work of fiction:

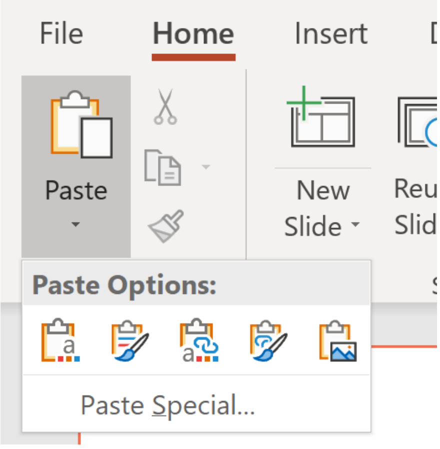


	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff
■ 2019	$77,034	$65,449	$69,948	$77,321
vs 2018 (%)	▲0.29%	▼0.24%	▲2.73%	▲0.69%

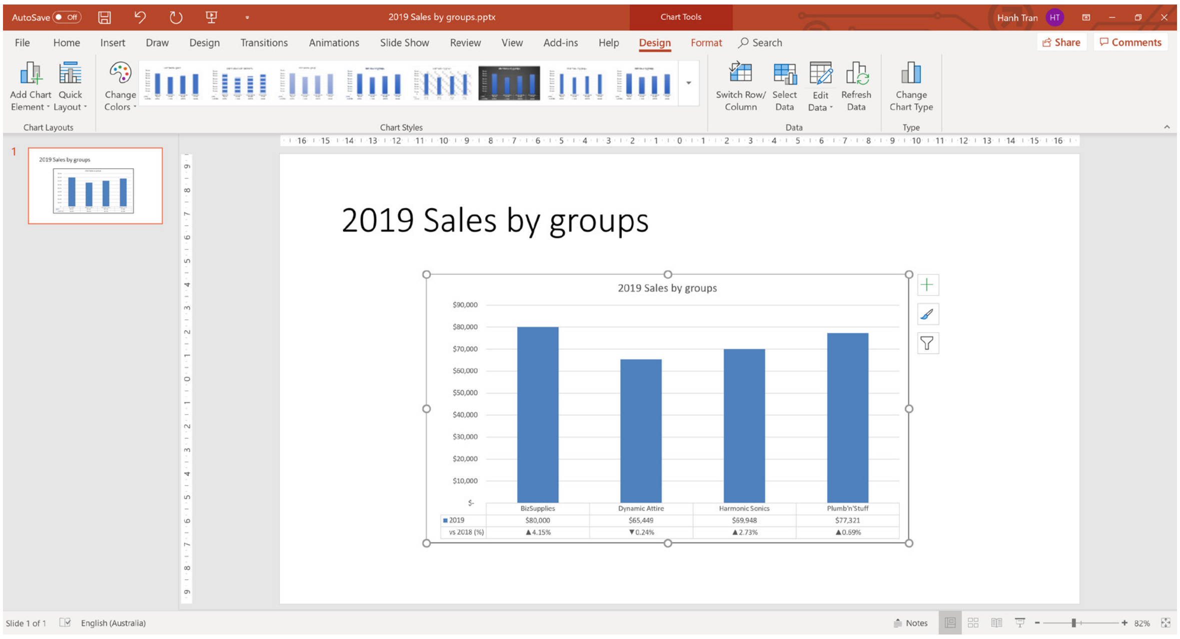
Next, the chart should be selected and the copied in the usual way *e.g.* click on the chart, right-click and select 'Copy'.

In PowerPoint, select the slide which will be used to house the graphic and then go to the Home tab on the Ribbon and on the Clipboard group, select Paste, where there are a few options to choose:

- Use Destination Theme & Embed Workbook
- Keep Source Formatting & Embed Workbook
- Use Destination Theme & Link Data
- Keep Source Formatting & Link Data
- Picture.



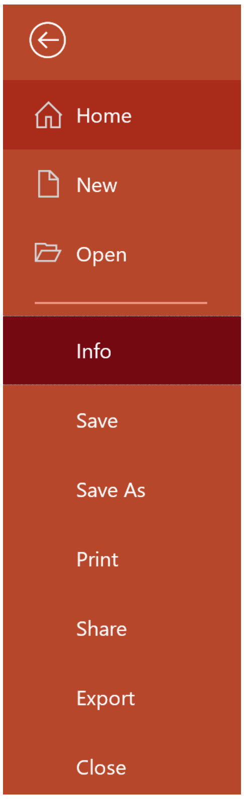
Choosing ‘Keep Source Formatting & Link Data’ will paste the chart from Excel without changing how it looks and establish the link to the data in the Excel file. Pasting the chart to the PowerPoint slide and click on it, a contextual ‘Chart Tools’ tab will appear on the Ribbon. Here, the chart can be reformatted, and chart data may also be refreshed from the linked Excel file:



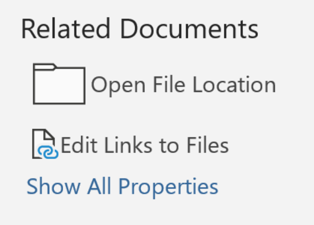
2019 Sales by groups

	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff
■ 2019	$80,000	$65,449	$69,948	$77,321
vs 2018 (%)	▲4.15%	▼0.24%	▲2.73%	▲0.69%

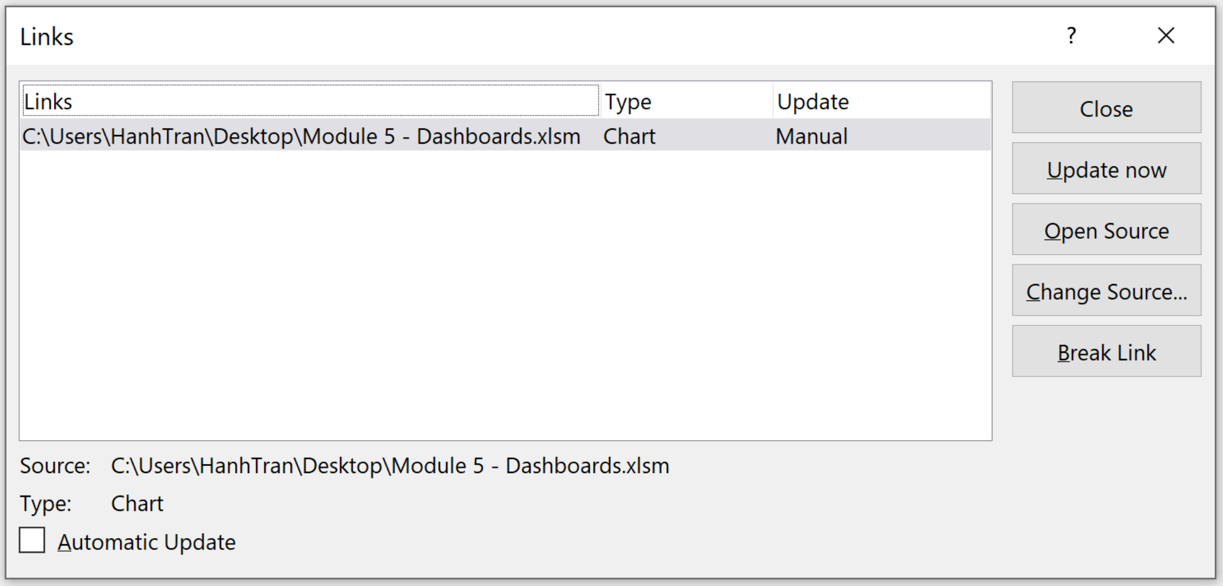
Any changes made in Excel will be reflected in this PowerPoint presentation as well, given that the links are not broken. To edit the links if they break, go to the File tab on the Ribbon, and then click on the Info tab:



We can navigate to the bottom right corner to find ‘Related Documents’, then click on ‘Edit Links to Files’:



A ‘Links’ dialog box will appear, allowing us to make changes or maintain the links if they were to break or the file is relocated.


Links

Links	Type	Update
C:\Users\HanhTran\Desktop\Module 5 - Dashboards.xlsm	Chart	Manual

Source: C:\Users\HanhTran\Desktop\Module 5 - Dashboards.xlsm

Type: Chart

□ Automatic Update

More next month.

Visual Basics

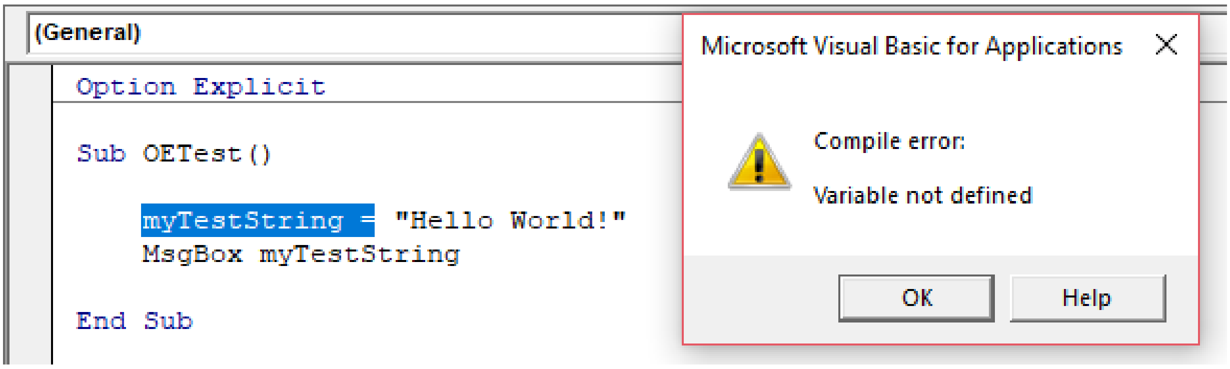
We thought we’d run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month we look at declaring variables.

To maintain good coding practice to declare variables, it is easy to ask VBA to force the coder to declare them all the time. This is done using the opening statement **Option Explicit**.

When **Option Explicit** appears in a file, you must explicitly declare all variables by using the **Dim** or **ReDim** statements. If you try to use an undeclared variable name, an error occurs at compile time.

As an example, let’s start with the following code:

```
Option Explicit

Sub OETest()

    myTestString = "Hello World!"

    MsgBox myTestString

End Sub
```


```
Option Explicit

Sub OETest()

    myTestString = "Hello World!"
    MsgBox myTestString

End Sub
```

Microsoft Visual Basic for Applications

Compile error:

Variable not defined

The compiler stops with an error alert and snaps back to the Visual Basic Editor to the variable that has not been declared. The procedure must be stopped and the declaration statements added before it will successfully execute.

More next time.

Power Pivot Principles

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

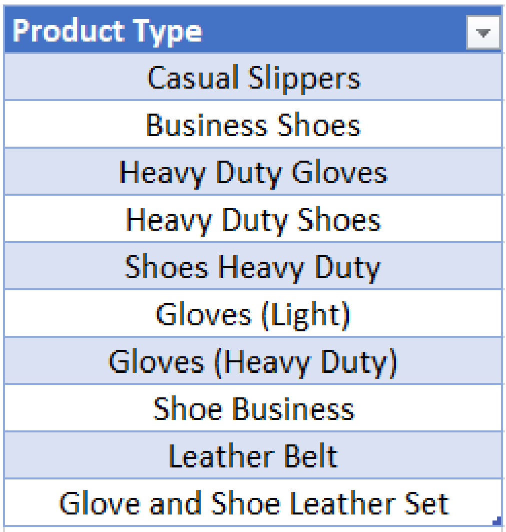
The **SEARCH** function takes the input text character or text string and looks for it in a data value. It returns with the character position number of where it found within the data value. **SEARCH** is case sensitive and looks from left to right.

The **SEARCH** function uses the following syntax to operate:

SEARCH(find_text, within_text [, [start_num][, NotFoundValue]])

- **find_text** is the text that we wish to find. Note that we can use the '**?**' and '*****' operators. The question mark '**?**' matches any single character to the string and the asterisk '*****' matches any sequence of characters. If we wish to find an actual question mark or asterisk, we have to include the tilde '**~**' before the question mark or asterisk
- **within_text** is the text that we wish to search for **find_text**, or a column containing text.
- **start_num** is an optional input, is the character position in **within_text**, where we want to start searching
- **NotFoundValue** is an optional input; this has to be a number, or we can use the **BLANK**() function, and it will be returned when the operation does not find a matching value.

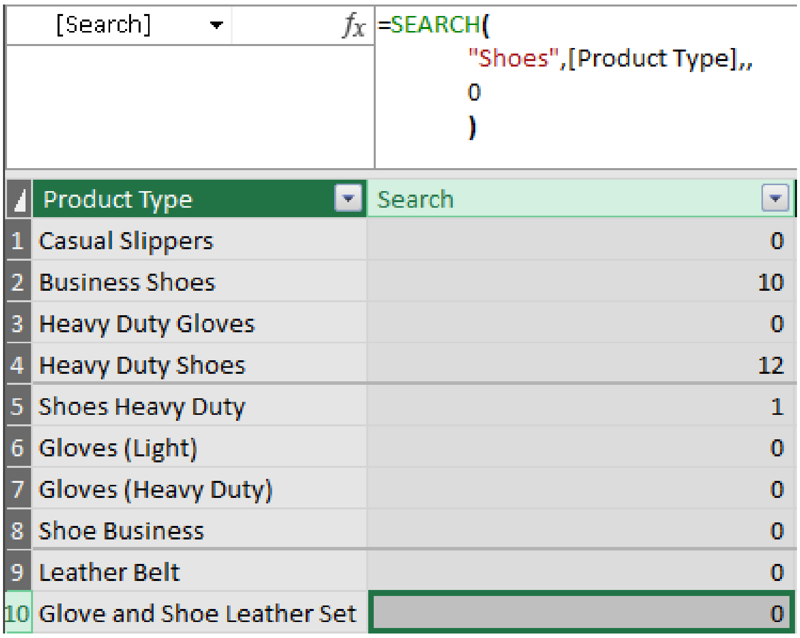
Let's say we want to search for the text "Shoes" in the following data set:



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

We can use the following DAX code to search for "Shoes":

```
=SEARCH(
            "Shoes",[Product Type],,
            0
            )
```



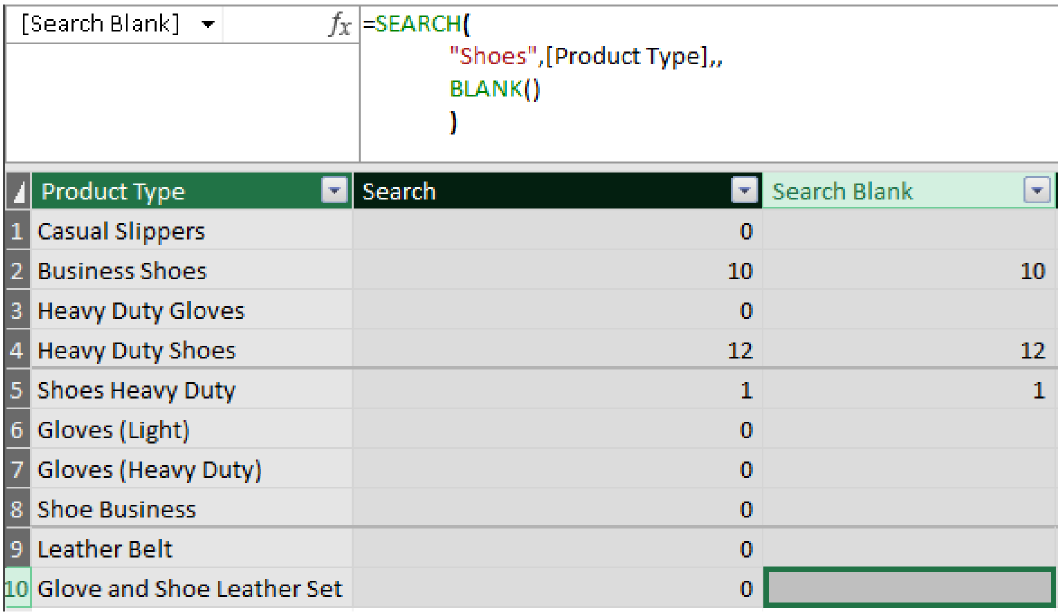
```
=SEARCH(
        "Shoes",[Product Type],,
        0
        )
```

	Product Type	Search
1	Casual Slippers	0
2	Business Shoes	10
3	Heavy Duty Gloves	0
4	Heavy Duty Shoes	12
5	Shoes Heavy Duty	1
6	Gloves (Light)	0
7	Gloves (Heavy Duty)	0
8	Shoe Business	0
9	Leather Belt	0
10	Glove and Shoe Leather Set	0

The **SEARCH** function returns with the text location of the first instance where it finds "Shoes" in the **Product Type** text string. In row 2 it is 10, because there are eight characters in "Business", space " " is the ninth character, therefore the 10th character is where the text string "Shoes" begins.

The current formula has been set to return with zero [0] when no matches are found.

We can also use the **BLANK**() function in place of zero:



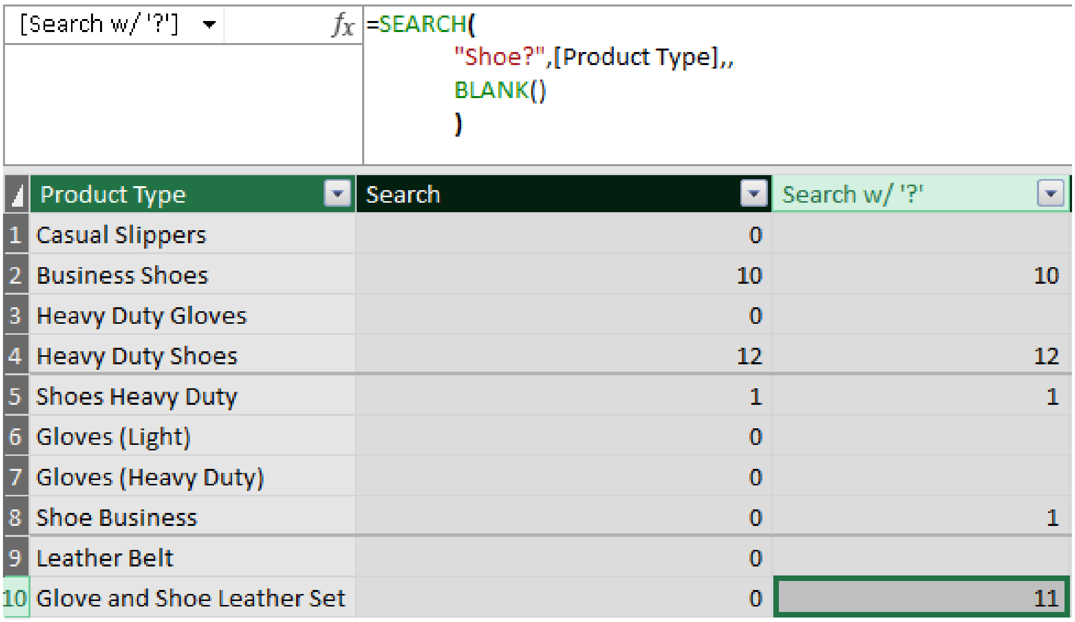
[Search Blank]

```
=SEARCH(
        "Shoes",[Product Type],,
        BLANK()
        )
```

	Product Type	Search	Search Blank
1	Casual Slippers	0	
2	Business Shoes	10	10
3	Heavy Duty Gloves	0	
4	Heavy Duty Shoes	12	12
5	Shoes Heavy Duty	1	1
6	Gloves (Light)	0	
7	Gloves (Heavy Duty)	0	
8	Shoe Business	0	
9	Leather Belt	0	
10	Glove and Shoe Leather Set	0	

Notice that our column is not picking up rows 8 and 10. That's because the data values in those rows contain "Shoe" rather than "Shoes". We can adjust our DAX formula to account for this:

```
=SEARCH(
        "Shoe?",[Product Type],,
        BLANK()
        )
```



[Search w/ '?']

```
=SEARCH(
        "Shoe?",[Product Type],,
        BLANK()
        )
```

	Product Type	Search	Search w/ '?'
1	Casual Slippers	0	
2	Business Shoes	10	10
3	Heavy Duty Gloves	0	
4	Heavy Duty Shoes	12	12
5	Shoes Heavy Duty	1	1
6	Gloves (Light)	0	
7	Gloves (Heavy Duty)	0	
8	Shoe Business	0	1
9	Leather Belt	0	
10	Glove and Shoe Leather Set	0	11

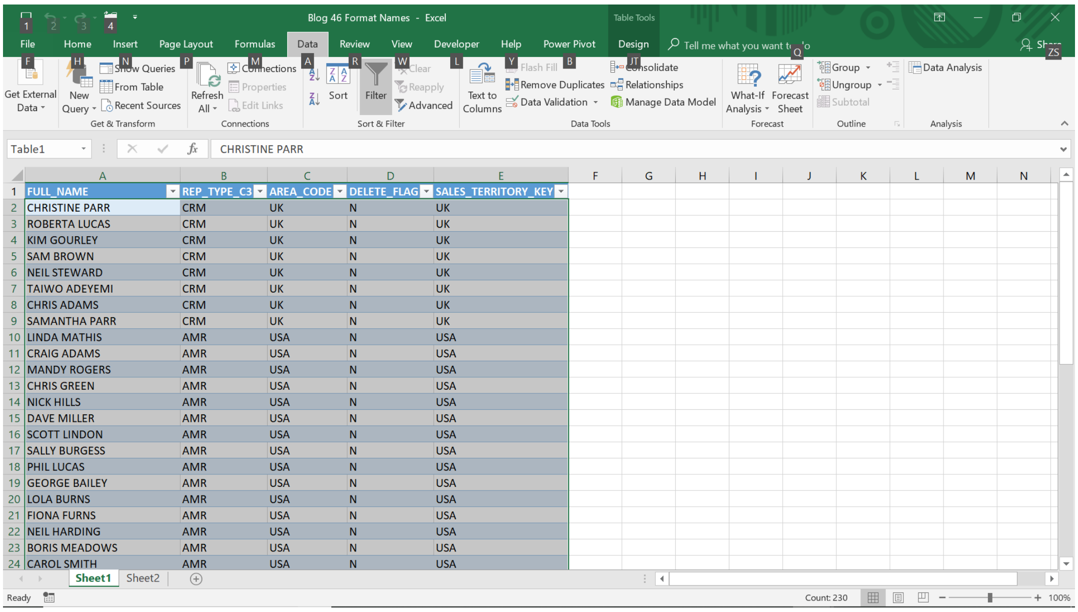
We included '**?**' in our **find_text** term so that we will pick up rows 8 and 10 where it's "Shoe " with a space and not "Shoe(s)".

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

Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we note some improvements to Power Query/Get and Transform, in particular looking at how creating a column from examples has developed.

'Column from Examples' does appear to keep getting improved. The good news is that you may now multitask by using this feature, as you can use it to combine transformations. This reminded me of a past problem detailed in this newsletter where we wanted to reformat some names:



Blog 46 Format Names - Excel

Table1 | CHRISTINE PARR

	FULL_NAME	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITORY_KEY
2	CHRISTINE PARR	CRM	UK	N	UK
3	ROBERTA LUCAS	CRM	UK	N	UK
4	KIM GOURLEY	CRM	UK	N	UK
5	SAM BROWN	CRM	UK	N	UK
6	NEIL STEWARD	CRM	UK	N	UK
7	TAIWO ADEYEMI	CRM	UK	N	UK
8	CHRIS ADAMS	CRM	UK	N	UK
9	SAMANTHA PARR	CRM	UK	N	UK
10	LINDA MATHIS	AMR	USA	N	USA
11	CRAIG ADAMS	AMR	USA	N	USA
12	MANDY ROGERS	AMR	USA	N	USA
13	CHRIS GREEN	AMR	USA	N	USA
14	NICK HILLS	AMR	USA	N	USA
15	DAVE MILLER	AMR	USA	N	USA
16	SCOTT LINDON	AMR	USA	N	USA
17	SALLY BURGESS	AMR	USA	N	USA
18	PHIL LUCAS	AMR	USA	N	USA
19	GEORGE BAILEY	AMR	USA	N	USA
20	LOLA BURNS	AMR	USA	N	USA
21	FIONA FURNS	AMR	USA	N	USA
22	NEIL HARDING	AMR	USA	N	USA
23	BORIS MEADOWS	AMR	USA	N	USA
24	CAROL SMITH	AMR	USA	N	USA

Ready | Count: 230 | 100%

My aim was to transform the **FULL_NAME** field so that we could have 'C. Parr' instead of 'CHRISTINE PARR'. The problem we came up against previously was that 'Column from Examples' couldn't cope (forgive the old Excel 2013 screenshot!):

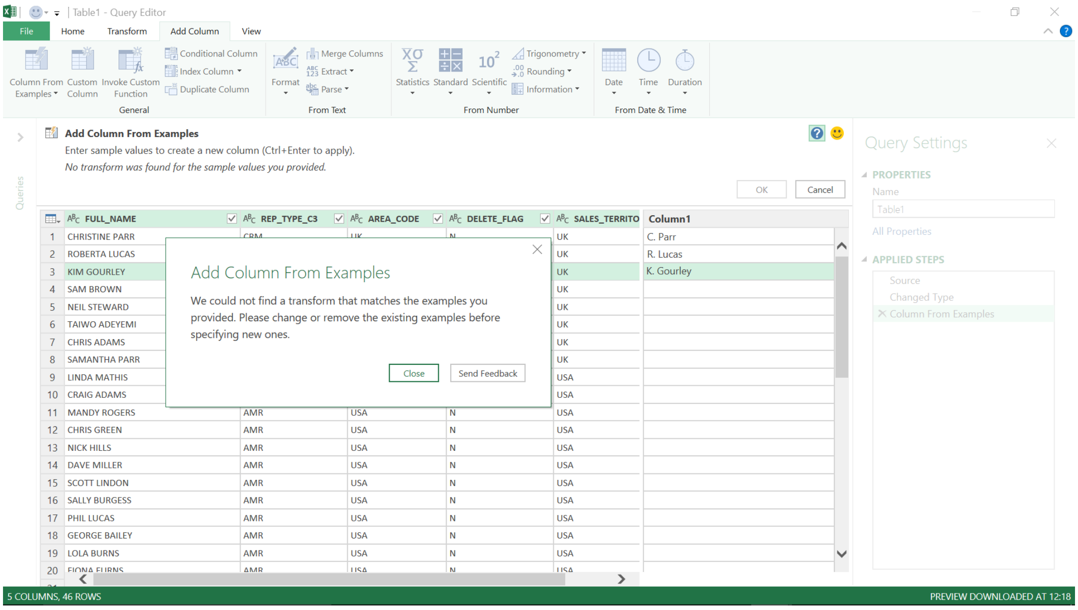


Table1 - Query Editor

Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

No transform was found for the sample values you provided.

Add Column From Examples

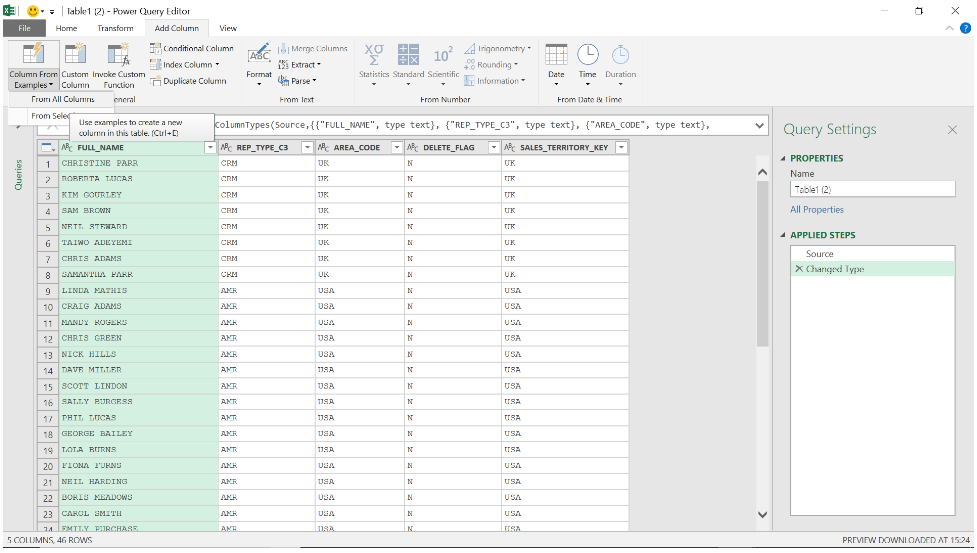
We could not find a transform that matches the examples you provided. Please change or remove the existing examples before specifying new ones.

	FULL_NAME	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITO	Column1
1	CHRISTINE PARR	CRM	UK	N	UK	C. Parr
2	ROBERTA LUCAS				UK	R. Lucas
3	KIM GOURLEY				UK	K. Gourley
4	SAM BROWN				UK	
5	NEIL STEWARD				UK	
6	TAIWO ADEYEMI				UK	
7	CHRIS ADAMS				UK	
8	SAMANTHA PARR				UK	
9	LINDA MATHIS				USA	
10	CRAIG ADAMS				USA	
11	MANDY ROGERS	AMR	USA	N	USA	
12	CHRIS GREEN	AMR	USA	N	USA	
13	NICK HILLS	AMR	USA	N	USA	
14	DAVE MILLER	AMR	USA	N	USA	
15	SCOTT LINDON	AMR	USA	N	USA	
16	SALLY BURGESS	AMR	USA	N	USA	
17	PHIL LUCAS	AMR	USA	N	USA	
18	GEORGE BAILEY	AMR	USA	N	USA	
19	LOLA BURNS	AMR	USA	N	USA	
20	FIONA FURNS	AMR	USA	N	USA	

Query Settings — Name: Table1 — Applied Steps: Source, Changed Type, Column From Examples

5 COLUMNS, 46 ROWS | PREVIEW DOWNLOADED AT 12:18

So, we are going to try that again!



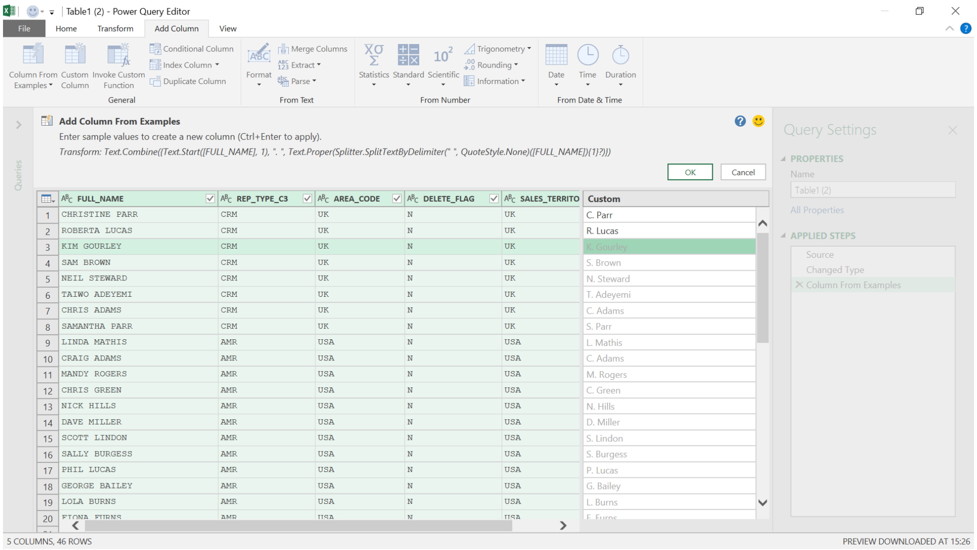
Use examples to create a new column in this table. (Ctrl+E)

```
ColumnTypes(Source,{{"FULL_NAME", type text}, {"REP_TYPE_C3", type text}, {"AREA_CODE", type text},
```

Name: Table1 (2)

	FULL_NAME	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITORY_KEY
1	CHRISTINE PARR	CRM	UK	N	UK
2	ROBERTA LUCAS	CRM	UK	N	UK
3	KIM GOURLEY	CRM	UK	N	UK
4	SAM BROWN	CRM	UK	N	UK
5	NEIL STEWARD	CRM	UK	N	UK
6	TAIWO ADEYEMI	CRM	UK	N	UK
7	CHRIS ADAMS	CRM	UK	N	UK
8	SAMANTHA PARR	CRM	UK	N	UK
9	LINDA MATHIS	AMR	USA	N	USA
10	CRAIG ADAMS	AMR	USA	N	USA
11	MANDY ROGERS	AMR	USA	N	USA
12	CHRIS GREEN	AMR	USA	N	USA
13	NICK HILLS	AMR	USA	N	USA
14	DAVE MILLER	AMR	USA	N	USA
15	SCOTT LINDON	AMR	USA	N	USA
16	SALLY BURGESS	AMR	USA	N	USA
17	PHIL LUCAS	AMR	USA	N	USA
18	GEORGE BAILEY	AMR	USA	N	USA
19	LOLA BURNS	AMR	USA	N	USA
20	FIONA FURNS	AMR	USA	N	USA
21	NEIL HARDING	AMR	USA	N	USA
22	BORIS MEADOWS	AMR	USA	N	USA
23	CAROL SMITH	AMR	USA	N	USA
24	EMILY PURCHASE	AMR	USA	N	USA

In the 'Add Column' tab, there is a section called 'Column from Examples'. Here, we're going to allow Power Query to get data for the new column 'From All Columns'.



Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

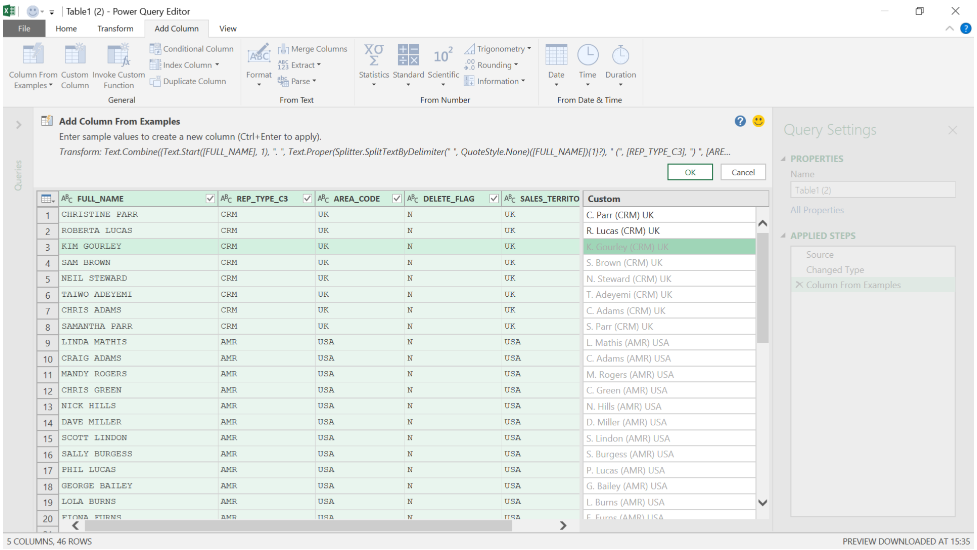
Transform: Text.Combine({Text.Start([FULL_NAME], 1), ". ", Text.Proper(Splitter.SplitTextByDelimiter(" ", QuoteStyle.None)([FULL_NAME]){1}?)})

	FULL_NAME	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITO	Custom
1	CHRISTINE PARR	CRM	UK	N	UK	C. Parr
2	ROBERTA LUCAS	CRM	UK	N	UK	R. Lucas
3	KIM GOURLEY	CRM	UK	N	UK	K. Gourley
4	SAM BROWN	CRM	UK	N	UK	S. Brown
5	NEIL STEWARD	CRM	UK	N	UK	N. Steward
6	TAIWO ADEYEMI	CRM	UK	N	UK	T. Adeyemi
7	CHRIS ADAMS	CRM	UK	N	UK	C. Adams
8	SAMANTHA PARR	CRM	UK	N	UK	S. Parr
9	LINDA MATHIS	AMR	USA	N	USA	L. Mathis
10	CRAIG ADAMS	AMR	USA	N	USA	C. Adams
11	MANDY ROGERS	AMR	USA	N	USA	M. Rogers
12	CHRIS GREEN	AMR	USA	N	USA	C. Green
13	NICK HILLS	AMR	USA	N	USA	N. Hills
14	DAVE MILLER	AMR	USA	N	USA	D. Miller
15	SCOTT LINDON	AMR	USA	N	USA	S. Lindon
16	SALLY BURGESS	AMR	USA	N	USA	S. Burgess
17	PHIL LUCAS	AMR	USA	N	USA	P. Lucas
18	GEORGE BAILEY	AMR	USA	N	USA	G. Bailey
19	LOLA BURNS	AMR	USA	N	USA	L. Burns
20	FIONA FURNS	AMR	USA	N	USA	F. Furns

Well this is definitely an improvement! After two examples, the new column has been created. We may click 'OK' to get my new column:

= Text.Combine({Text.Start([FULL_NAME], 1), ". ", Text.Proper(Splitter.SplitTextByDelimiter(" ", QuoteStyle.None)([FULL_NAME]){1}?)})

Let's delete this column and try combining data from other columns. This time, not only do we want to format the name, we also want to include the job title and region:



Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

Transform: Text.Combine({Text.Start([FULL_NAME], 1), ". ", Text.Proper(Splitter.SplitTextByDelimiter(" ", QuoteStyle.None)([FULL_NAME]){1}?), " (", [REP_TYPE_C3], ") ", [ARE...

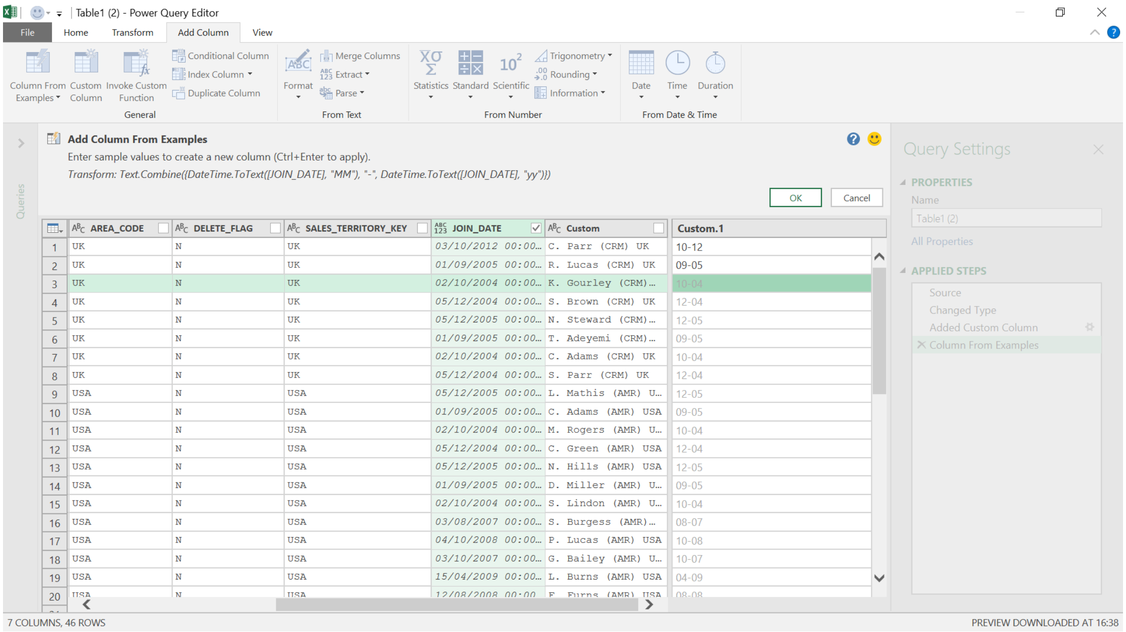
	FULL_NAME	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITO	Custom
1	CHRISTINE PARR	CRM	UK	N	UK	C. Parr (CRM) UK
2	ROBERTA LUCAS	CRM	UK	N	UK	R. Lucas (CRM) UK
3	KIM GOURLEY	CRM	UK	N	UK	K. Gourley (CRM) UK
4	SAM BROWN	CRM	UK	N	UK	S. Brown (CRM) UK
5	NEIL STEWARD	CRM	UK	N	UK	N. Steward (CRM) UK
6	TAIWO ADEYEMI	CRM	UK	N	UK	T. Adeyemi (CRM) UK
7	CHRIS ADAMS	CRM	UK	N	UK	C. Adams (CRM) UK
8	SAMANTHA PARR	CRM	UK	N	UK	S. Parr (CRM) UK
9	LINDA MATHIS	AMR	USA	N	USA	L. Mathis (AMR) USA
10	CRAIG ADAMS	AMR	USA	N	USA	C. Adams (AMR) USA
11	MANDY ROGERS	AMR	USA	N	USA	M. Rogers (AMR) USA
12	CHRIS GREEN	AMR	USA	N	USA	C. Green (AMR) USA
13	NICK HILLS	AMR	USA	N	USA	N. Hills (AMR) USA
14	DAVE MILLER	AMR	USA	N	USA	D. Miller (AMR) USA
15	SCOTT LINDON	AMR	USA	N	USA	S. Lindon (AMR) USA
16	SALLY BURGESS	AMR	USA	N	USA	S. Burgess (AMR) USA
17	PHIL LUCAS	AMR	USA	N	USA	P. Lucas (AMR) USA
18	GEORGE BAILEY	AMR	USA	N	USA	G. Bailey (AMR) USA
19	LOLA BURNS	AMR	USA	N	USA	L. Burns (AMR) USA
20	FIONA FURNS	AMR	USA	N	USA	F. Furns (AMR) USA

We create the column to see the full transformation.

= Text.Combine({Text.Start([FULL_NAME], 1), ". ", Text.Proper(Splitter.SplitTextByDelimiter(" ", QuoteStyle.None)([FULL_NAME]){1}?), " (", [REP_TYPE_C3], ") ", [AREA_CODE]})

It shows that it's worth trying out new methods to solve old problems, as Power Query is constantly improving.

There are also developments in using dates in 'Column from Examples' for date formats used in specific domains though we found this functionality a bit hit and miss. Hopefully, it will improve. We've tried it with **JOIN_DATE** in the following example:



Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

Transform: Text.Combine({DateTime.ToText([JOIN_DATE], "MM"), "-", DateTime.ToText([JOIN_DATE], "yy")})

OK Cancel

	AREA_CODE	DELETE_FLAG	SALES_TERRITORY_KEY	JOIN_DATE ☑	Custom	Custom.1
1	UK	N	UK	03/10/2012 00:00...	C. Parr (CRM) UK	10-12
2	UK	N	UK	01/09/2005 00:00...	R. Lucas (CRM) UK	09-05
3	UK	N	UK	02/10/2004 00:00...	K. Gourley (CRM)...	10-04
4	UK	N	UK	05/12/2004 00:00...	S. Brown (CRM) UK	12-04
5	UK	N	UK	05/12/2005 00:00...	N. Steward (CRM)...	12-05
6	UK	N	UK	01/09/2005 00:00...	T. Adeyemi (CRM)...	09-05
7	UK	N	UK	02/10/2004 00:00...	C. Adams (CRM) UK	10-04
8	UK	N	UK	05/12/2004 00:00...	S. Parr (CRM) UK	12-04
9	USA	N	USA	05/12/2005 00:00...	L. Mathis (AMR) U...	12-05
10	USA	N	USA	01/09/2005 00:00...	C. Adams (AMR) USA	09-05
11	USA	N	USA	02/10/2004 00:00...	M. Rogers (AMR) U...	10-04
12	USA	N	USA	05/12/2004 00:00...	C. Green (AMR) USA	12-04
13	USA	N	USA	05/12/2005 00:00...	N. Hills (AMR) USA	12-05
14	USA	N	USA	01/09/2005 00:00...	D. Miller (AMR) U...	09-05
15	USA	N	USA	02/10/2004 00:00...	S. Lindon (AMR) U...	10-04
16	USA	N	USA	03/08/2007 00:00...	S. Burgess (AMR)...	08-07
17	USA	N	USA	04/10/2008 00:00...	P. Lucas (AMR) USA	10-08
18	USA	N	USA	03/10/2007 00:00...	G. Bailey (AMR) U...	10-07
19	USA	N	USA	15/04/2009 00:00...	L. Burns (AMR) USA	04-09
20	USA	N	USA	12/08/2008 00:00	F. Furns (AMR) USA	08-08

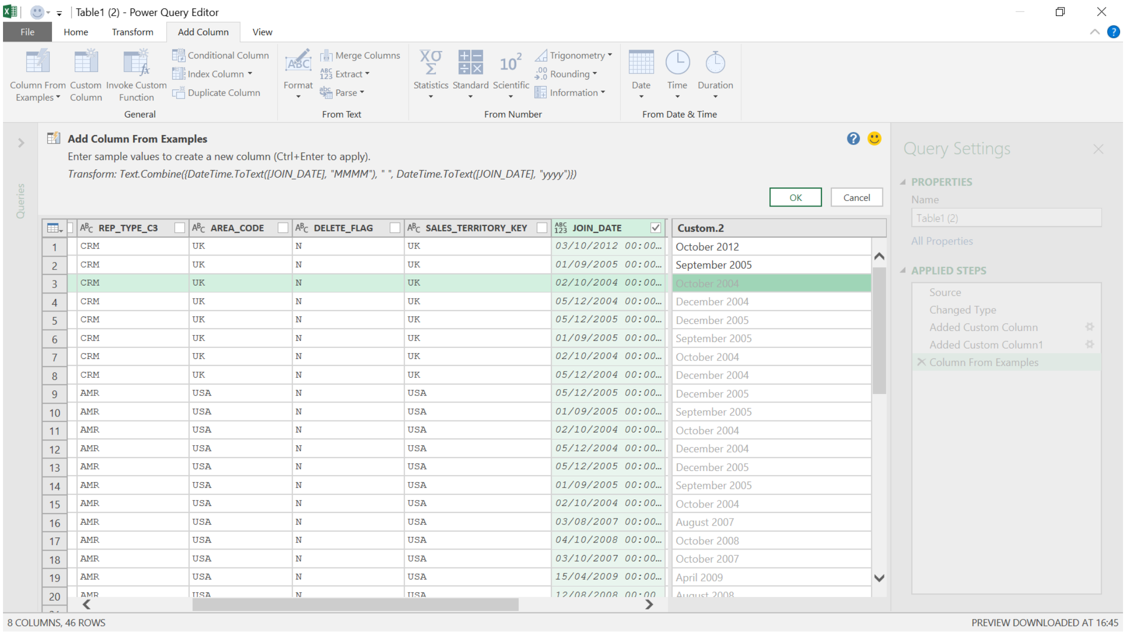
7 COLUMNS, 46 ROWS PREVIEW DOWNLOADED AT 16:38

The date is translated to MM-YY from a full datetime format.

The **M** formula is

= Text.Combine({DateTime.ToText([JOIN_DATE], "MM"), "-", DateTime.ToText([JOIN_DATE], "yy")})

We can also extract the name of the month:



Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

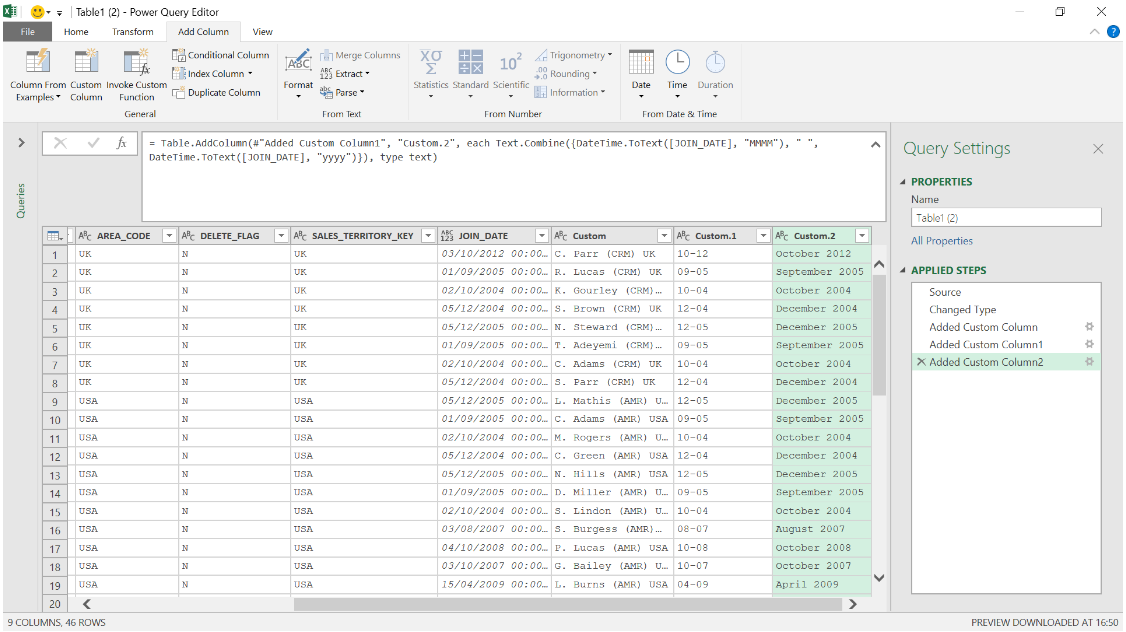
Transform: Text.Combine({DateTime.ToText([JOIN_DATE], "MMMM"), " ", DateTime.ToText([JOIN_DATE], "yyyy")})

OK Cancel

	REP_TYPE_C3	AREA_CODE	DELETE_FLAG	SALES_TERRITORY_KEY	JOIN_DATE ☑	Custom.2
1	CRM	UK	N	UK	03/10/2012 00:00...	October 2012
2	CRM	UK	N	UK	01/09/2005 00:00...	September 2005
3	CRM	UK	N	UK	02/10/2004 00:00...	October 2004
4	CRM	UK	N	UK	05/12/2004 00:00...	December 2004
5	CRM	UK	N	UK	05/12/2005 00:00...	December 2005
6	CRM	UK	N	UK	01/09/2005 00:00...	September 2005
7	CRM	UK	N	UK	02/10/2004 00:00...	October 2004
8	CRM	UK	N	UK	05/12/2004 00:00...	December 2004
9	AMR	USA	N	USA	05/12/2005 00:00...	December 2005
10	AMR	USA	N	USA	01/09/2005 00:00...	September 2005
11	AMR	USA	N	USA	02/10/2004 00:00...	October 2004
12	AMR	USA	N	USA	05/12/2004 00:00...	December 2004
13	AMR	USA	N	USA	05/12/2005 00:00...	December 2005
14	AMR	USA	N	USA	01/09/2005 00:00...	September 2005
15	AMR	USA	N	USA	02/10/2004 00:00...	October 2004
16	AMR	USA	N	USA	03/08/2007 00:00...	August 2007
17	AMR	USA	N	USA	04/10/2008 00:00...	October 2008
18	AMR	USA	N	USA	03/10/2007 00:00...	October 2007
19	AMR	USA	N	USA	15/04/2009 00:00...	April 2009
20	AMR	USA	N	USA	12/08/2008 00:00	August 2008

8 COLUMNS, 46 ROWS PREVIEW DOWNLOADED AT 16:45

We create the new column.



```
= Table.AddColumn(#"Added Custom Column1", "Custom.2", each Text.Combine({DateTime.ToText([JOIN_DATE], "MMMM"), " ",
DateTime.ToText([JOIN_DATE], "yyyy")}), type text)
```

	AREA_CODE	DELETE_FLAG	SALES_TERRITORY_KEY	JOIN_DATE	Custom	Custom.1	Custom.2
1	UK	N	UK	03/10/2012 00:00...	C. Parr (CRM) UK	10-12	October 2012
2	UK	N	UK	01/09/2005 00:00...	R. Lucas (CRM) UK	09-05	September 2005
3	UK	N	UK	02/10/2004 00:00...	K. Gourley (CRM)...	10-04	October 2004
4	UK	N	UK	05/12/2004 00:00...	S. Brown (CRM) UK	12-04	December 2004
5	UK	N	UK	05/12/2005 00:00...	N. Steward (CRM)...	12-05	December 2005
6	UK	N	UK	01/09/2005 00:00...	T. Adeyemi (CRM)...	09-05	September 2005
7	UK	N	UK	02/10/2004 00:00...	C. Adams (CRM) UK	10-04	October 2004
8	UK	N	UK	05/12/2004 00:00...	S. Parr (CRM) UK	12-04	December 2004
9	USA	N	USA	05/12/2005 00:00...	L. Mathis (AMR) U...	12-05	December 2005
10	USA	N	USA	01/09/2005 00:00...	C. Adams (AMR) USA	09-05	September 2005
11	USA	N	USA	02/10/2004 00:00...	M. Rogers (AMR) U...	10-04	October 2004
12	USA	N	USA	05/12/2004 00:00...	C. Green (AMR) USA	12-04	December 2004
13	USA	N	USA	05/12/2005 00:00...	N. Hills (AMR) USA	12-05	December 2005
14	USA	N	USA	01/09/2005 00:00...	D. Miller (AMR) U...	09-05	September 2005
15	USA	N	USA	02/10/2004 00:00...	S. Lindon (AMR) U...	10-04	October 2004
16	USA	N	USA	03/08/2007 00:00...	S. Burgess (AMR)...	08-07	August 2007
17	USA	N	USA	04/10/2008 00:00...	P. Lucas (AMR) USA	10-08	October 2008
18	USA	N	USA	03/10/2007 00:00...	G. Bailey (AMR) U...	10-07	October 2007
19	USA	N	USA	15/04/2009 00:00...	L. Burns (AMR) USA	04-09	April 2009
20							

Name: Table1 (2)

Applied steps:
- Source
- Changed Type
- Added Custom Column
- Added Custom Column1
- Added Custom Column2

9 COLUMNS, 46 ROWS

PREVIEW DOWNLOADED AT 16:50

The **M** formula used is:

= Text.Combine({DateTime.ToText([JOIN_DATE], "MMMM"), " ", DateTime.ToText([JOIN_DATE], "yyyy")})

It's good to see the functionality is expanding, making 'Column from Example' an even more useful feature – especially for beginners.

Until next month.

Power BI Updates

Unfortunately, the latest set of Power BI updates were announced after our January newsletter had to be prepared. So apologies for our tardiness, but this month's updates include various updates to reporting (*e.g.* more options for styling your column and bar charts) and improvements for inserting insert Power BI content into your PowerPoint presentations.

The full list of updates is as follows:

Reporting

- On-Object Interaction updates (in Preview)
- More styling options for Column and Bar charts
- Extended customisation for data labels
- Azure Maps clustering aggregation support for bubble layer
- Alerting on your Power BI reports with Data Activator

Data connectivity

- OneLake data hub in Power BI Desktop
- Power BI connector updates

Service

- Storytelling in PowerPoint: suggested content
- Ability to search and sort in paginated reports

Developers

- Power BI custom visuals authentication API
- Dynamic drill down API
- Git integration conflict resolution

Visualisations

- New visuals in AppSource
- Activity Gauge by Powerviz
- Navigate Financial Risks and Rewards with Profitbase's Power BI Visualisation
- Intuitive multi-level Donut charts for your reports
- Inforiver Analytics+ 3.2
- The Microsoft Purview hub in Fabric for creators in Preview
- VideoPlayer Visual
- Ridgeline Plot by Powerviz.

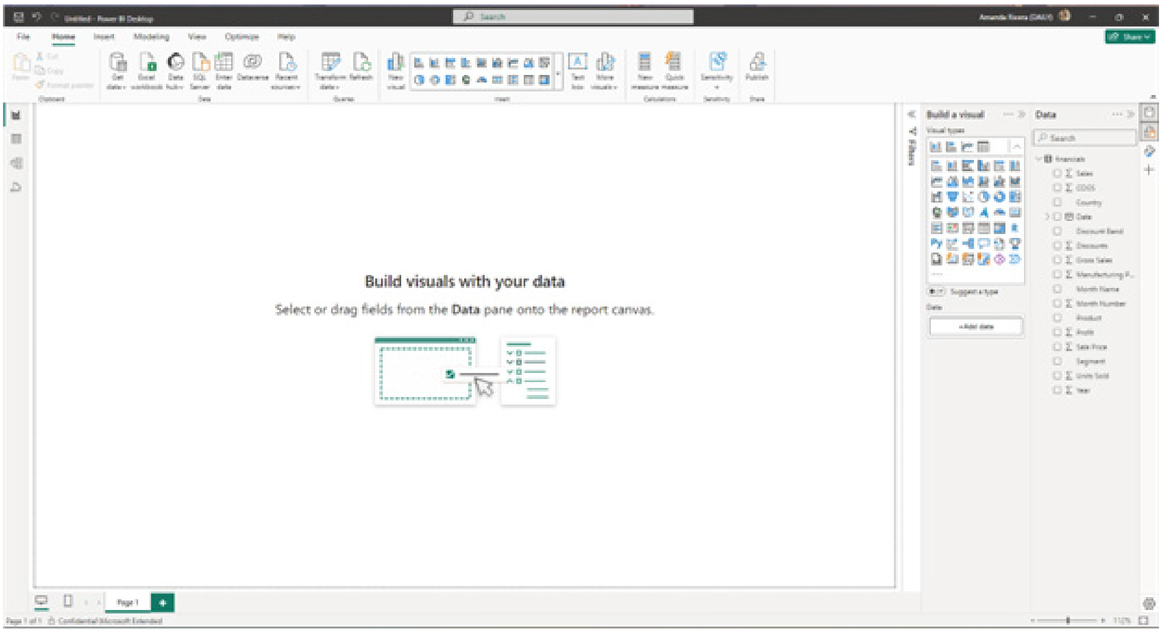
Let's look at each in turn.

On-Object Interaction updates (in Preview)

NEW ON-OBJECT DEFAULT SETUP

This update sees the introduction of a new default setup that feels closer to the previous classic look. With this new default, you will be able to preset the following options with just one click. Examples include:

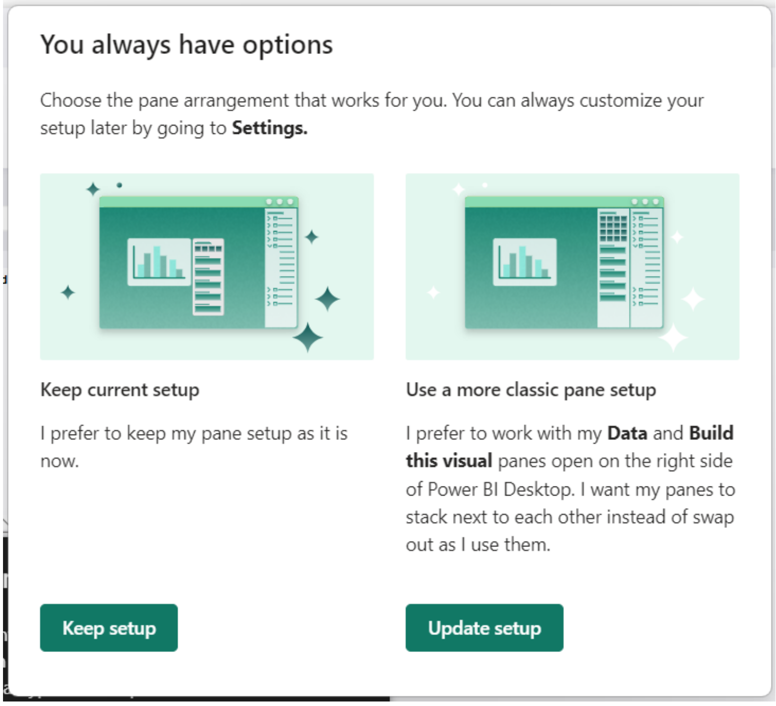
- Move the 'Build a visual' menu to the pane switcher by default
- Show both Build and Data panes expanded by default
- Format pane in the pane switcher by default
- 'Always open in new pane' set to on, thereby stacking panes by default
- Collapse panes to the pane switcher instead of close 'X' by default.



Build visuals with your data

Select or drag fields from the Data pane onto the report canvas.

Upon updating to the new build, with the new On-Object Preview switch set to 'On', you'll be greeted with a dialog to update to the new default setup. If you like your existing on-object setup and do not wish to have a more classic look, choose 'Keep setup' and no changes will be made.



You always have options

Choose the pane arrangement that works for you. You can always customize your setup later by going to **Settings.**

Keep current setup

I prefer to keep my pane setup as it is now.

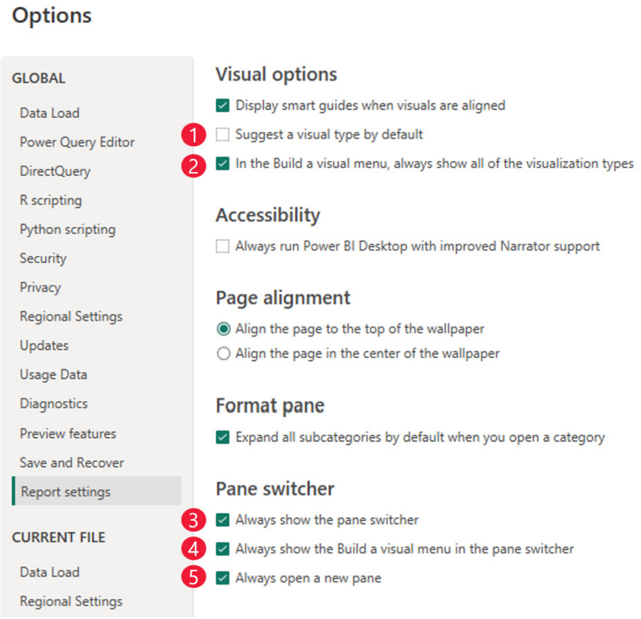
Use a more classic pane setup

I prefer to work with my **Data** and **Build this visual** panes open on the right side of Power BI Desktop. I want my panes to stack next to each other instead of swap out as I use them.

If you change your mind, you can always go back to the Options menu to customise the experience to your needs. Here's a walkthrough of what each of these options do:

- **Suggest a type by default:** if you prefer the option in the Build menu to be off by default, expanding all the field wells always, make sure this is not checked
- **In the Build a visual menu, always show all the visualisation types:** if you prefer to see all the visualisation types shown by default in the Build menu, be sure to check this option
- **Always show pane switcher:** when checked, this provides the right-hand pane switcher shown even when working with zero panes. It provides access to the '+' menu and setting gear always
- **Always show the Build a visual menu in the pane switcher:** this moves the Build menu from showing next to the visual, to the pane switcher on the right side instead. Check this option if you prefer to work with the Build menu as a pane
- **Always open a new pane:** instead of switching between panes, check this option if you prefer the stacking behaviour as before. With this option on, opening a new pane will open side-by-side by default.

To summarise, here's the state of these settings for the new default setup:



Options

GLOBAL
Data Load
Power Query Editor
DirectQuery
R scripting
Python scripting
Security
Privacy
Regional Settings
Updates
Usage Data
Diagnostics
Preview features
Save and Recover
Report settings

CURRENT FILE
Data Load
Regional Settings

Visual options
☑ Display smart guides when visuals are aligned
1 □ Suggest a visual type by default
2 ☑ In the Build a visual menu, always show all of the visualization types

Accessibility
□ Always run Power BI Desktop with improved Narrator support

Page alignment
● Align the page to the top of the wallpaper
○ Align the page in the center of the wallpaper

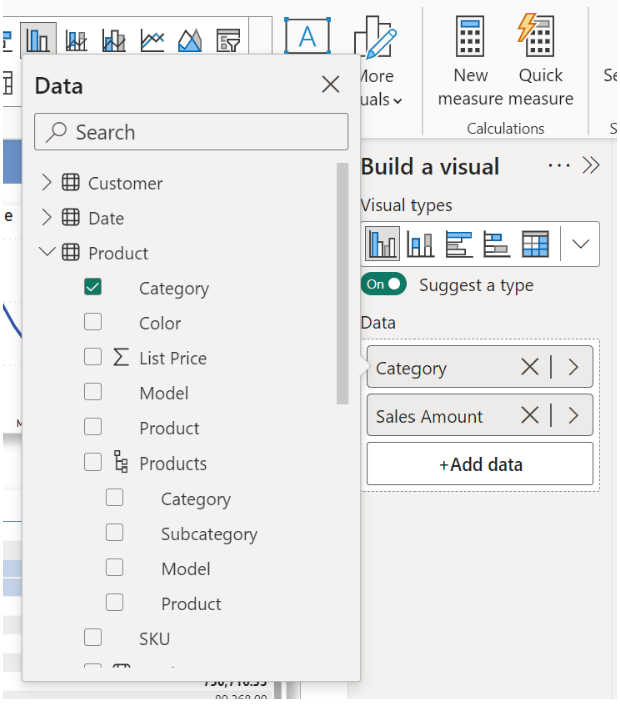
Format pane
☑ Expand all subcategories by default when you open a category

Pane switcher
3 ☑ Always show the pane switcher
4 ☑ Always show the Build a visual menu in the pane switcher
5 ☑ Always open a new pane

MULTI-SELECT IN DATA FLYOUT

To cut down on the extra clicks when first adding data using the new 'add data' button, the data flyout now allows you to multi-select fields to add fields in bulk to your visual, just like the Data pane. Simply check all the fields you'd like to add to your visual in one go.

This works best when 'suggest a type' is switched to 'on' so you can add as many fields as you like and Power BI will place the fields in the appropriate field wells for you.



Data

Search

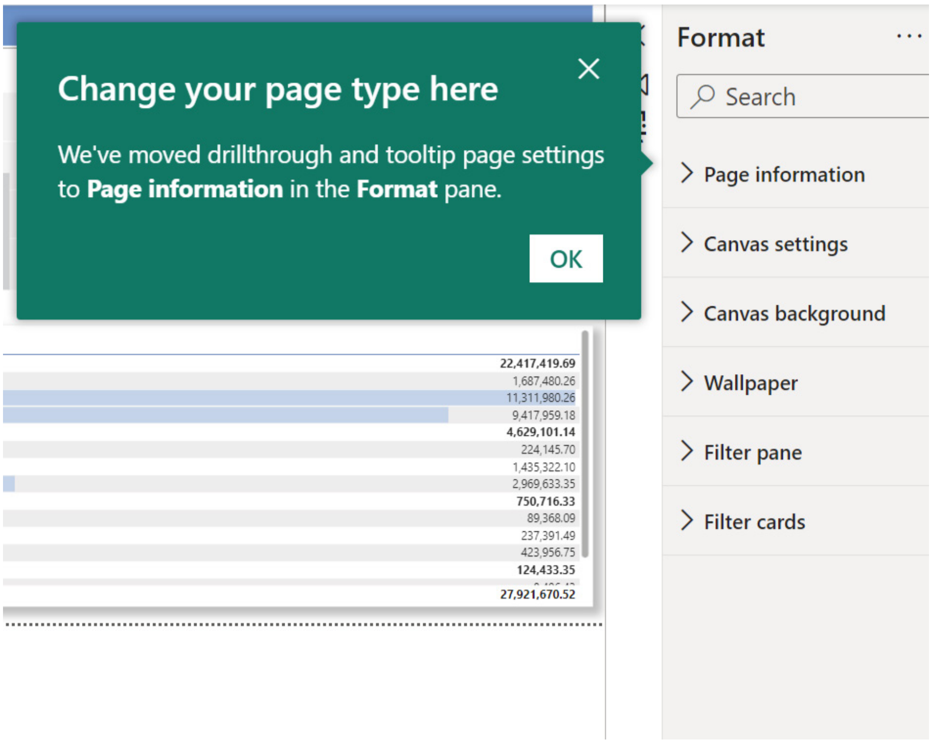
Customer
Date
Product
☑ Category
□ Color
□ List Price
□ Model
□ Product
□ Products
□ Category
□ Subcategory
□ Model
□ Product
□ SKU

Build a visual
Visual types
On Suggest a type
Data
Category
Sales Amount
+Add data

IN-PRODUCT USER EDUCATION

With the many changes the On-Object interaction introduced, it can hard to know where certain features may have moved. This month, four [4] teaching bubbles have been added to help users along the way.

Question	Answer
Where is the Drillthrough and ToolTip page set up?	A teaching bubble will now point to the new location within **Format -> Page information**.
Where did the Analytics pane go?	Analytics options like error bars, max, min or percentile lines are now consolidated in the Format pane at the bottom. You can set a max, min or other reference line using the new ‘Reference line’ card in the Format pane.
What happened to the right click options for a field like ‘rename’ or ‘show value as’?	They are still available on the right click. Simply right click on the field name to see the same options as before.
How do you bring back the Data pane if the pane switcher is gone?	You shouldn’t run into this as frequently now that the behaviour has been updated to collapse panes by default instead of closing. However, in case you do, you can always re-open a pane from the View tab in the Ribbon.



Change your page type here

We've moved drillthrough and tooltip page settings to **Page information** in the **Format** pane.

OK

22,417,419.69
1,687,480.26
11,311,980.26
9,417,959.18
4,629,101.14
224,145.70
1,435,322.10
2,969,633.35
750,716.33
89,368.09
237,391.49
423,956.75
124,433.35
27,921,670.52

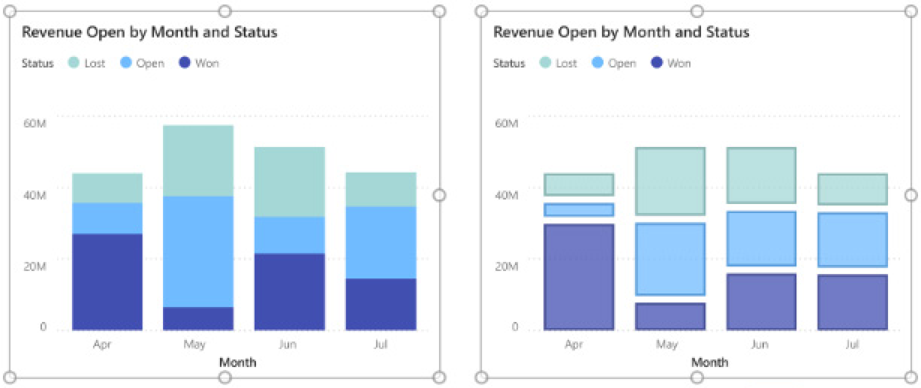
Format

Search

- Page information
- Canvas settings
- Canvas background
- Wallpaper
- Filter pane
- Filter cards

More styling options for Column and Bar charts

This update sees many new formatting enhancements for Column and Bar charts in Power BI. These additions are designed to improve your data visualisation, enabling more impactful analysis and storytelling capabilities.



Visual shapes
Default

Visual shapes New
Enhanced

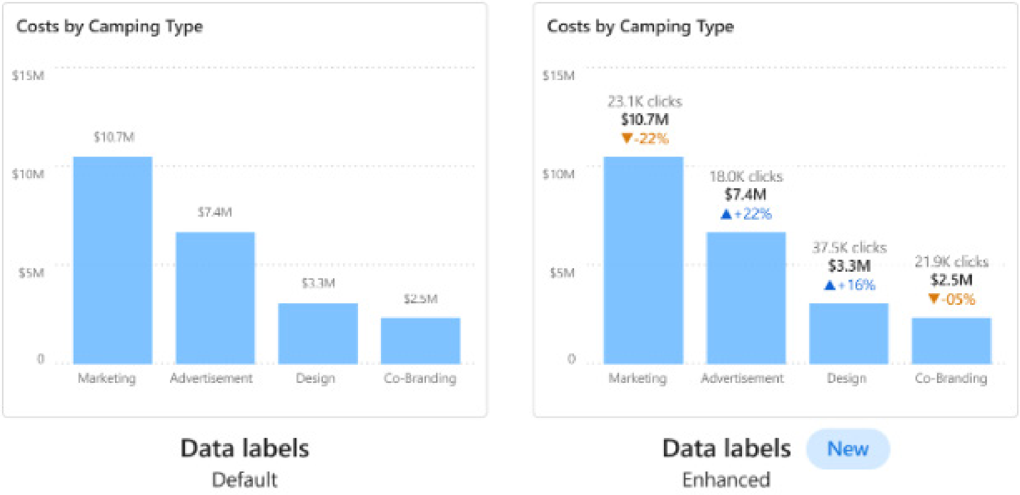
Key enhancements include:

- **Apply settings to All or Series:** you may now apply formatting options globally across all categories or tailor them for each unique individual series
- **Transparency Control:** you can now adjust the fill colour transparency to achieve a better balance of visibility and design
- **Border customisation:** you can now introduce borders with precision, choosing a distinct colour or match it to the column's fill, and define both the border's width and transparency for a more polished look
- **Layout:** the Spacing card has been renamed the Layout card. This update also has relocated the familiar 'Reverse Order' toggle that allows you to invert series, and introduced 'Sort by Value', 'Space between categories' and 'Space between series'. Furthermore, with Clustered columns and Bar charts, you now have the option to Erode or Explode the Stacked series
- **Advanced Ribbon settings:** you can highlight a series or create visual continuity with an array of options including matching series colour or choosing a colour selection, transparency, border colour matching or colour selection, border width and border spacing.

These features aim to provide you with greater flexibility and creative freedom in presenting your data, ensuring that your insights stand out more.

Extended customisation for data labels

Data labels are also getting more options too. With this update, there are now some new enhanced capabilities for Data labels, giving you extended customisation choices. Further, it brings the simple integration of additional metrics, dramatically improving your data analysis experience.



Key updates include:

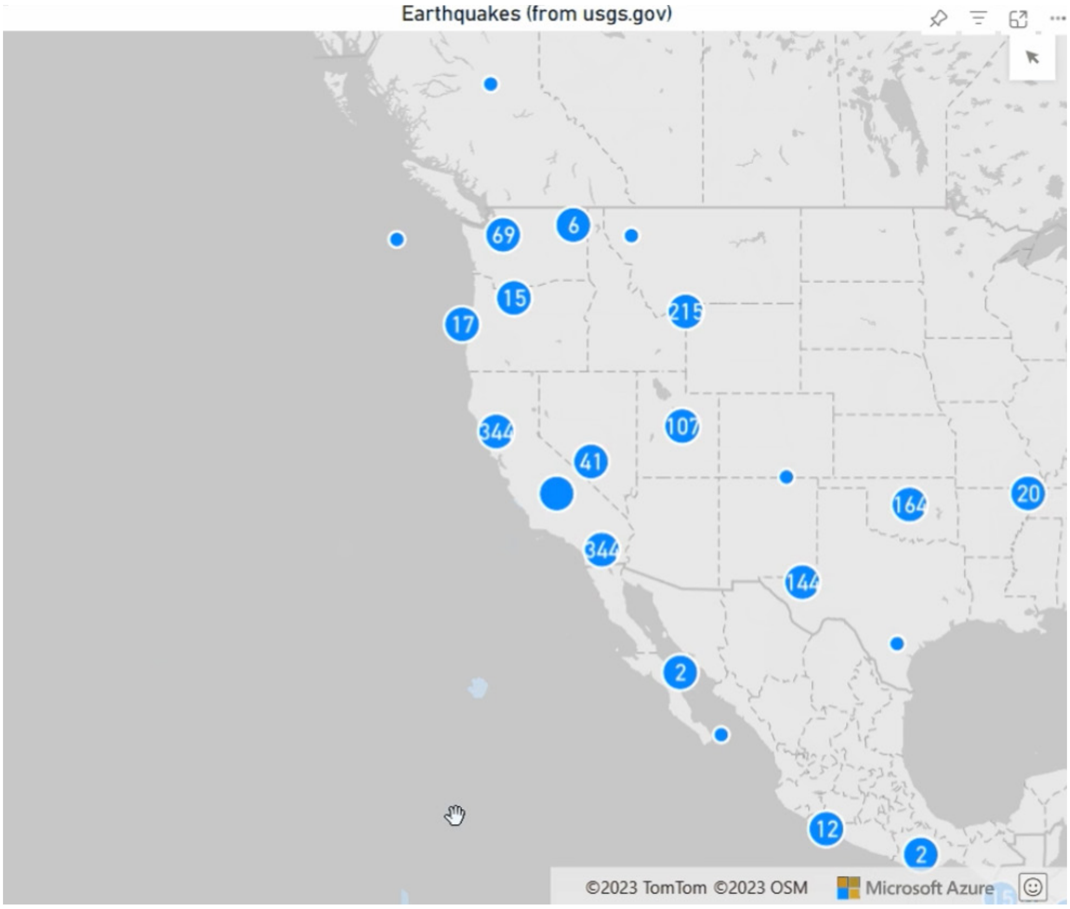
- **Title:** The new Title card feature enables the display of legend fields on your data labels, offering the option to omit the legend for a cleaner look. It also allows customised data label titles using different data fields, adjustable font style, colour and transparency
- **Value:** you can now easily spot the active field for your data label in the field well. Modifying it to another field is quick and easy, with customisation options like font, colour, transparency and more, plus a new feature for blank value display
- **Detail:** The new Detail card feature unlocks new potential by allowing the inclusion of a secondary metric to your data labels. You may choose any field to add an extra value, with plenty of formatting options at your disposal
- **Visual label layout:** this is a new feature that lets you choose between a single-line data label or a multi-line data label, where desired.

These features are available for Columns, Bars, Lines and Ribbon charts.

Azure Maps clustering aggregation support for bubble layer

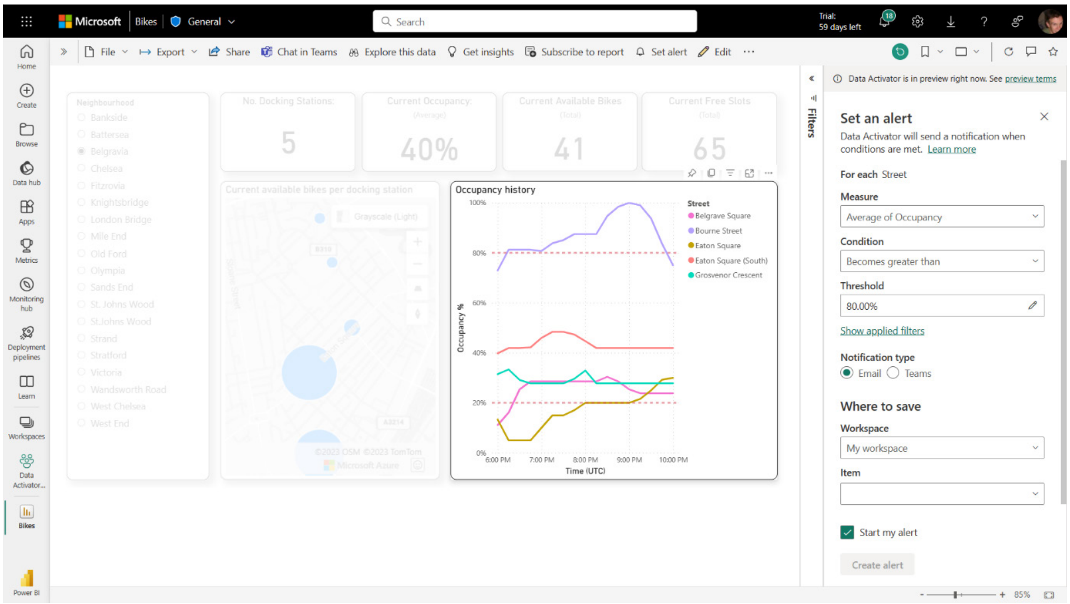
Microsoft has introduced an enhancement to the Power BI Azure Maps visual: the clustering aggregation feature for the bubble layer with customisable styling options. This capability not only enables you to dynamically aggregate data within bubble layers based upon distinct zoom levels, but also allows you to tailor the visual representation of clustered data.

You may now style your cluster bubbles, choosing from a range of options including bubble colours, font size and more. This level of customisation assists you with visually highlighting and emphasising specific aspects of the data, creating more impactful and informative geospatial visualisations.



Alerting on your Power BI reports with Data Activator

Back in October, Microsoft announced the public Preview of Data Activator. Many Power BI users have since been asking for alerting capabilities within reports. From the visual context menu or Report toolbar you can choose ‘Set alert’. This opens a pane where you may set the measure you want to monitor and the conditions you wish to detect:



Data Activator is in preview right now. See preview terms

Set an alert

Data Activator will send a notification when conditions are met. Learn more

For each Street

Measure: Average of Occupancy

Condition: Becomes greater than

Threshold: 80.00%

Show applied filters

Notification type: ● Email ○ Teams

Where to save

Workspace: My workspace

Item:

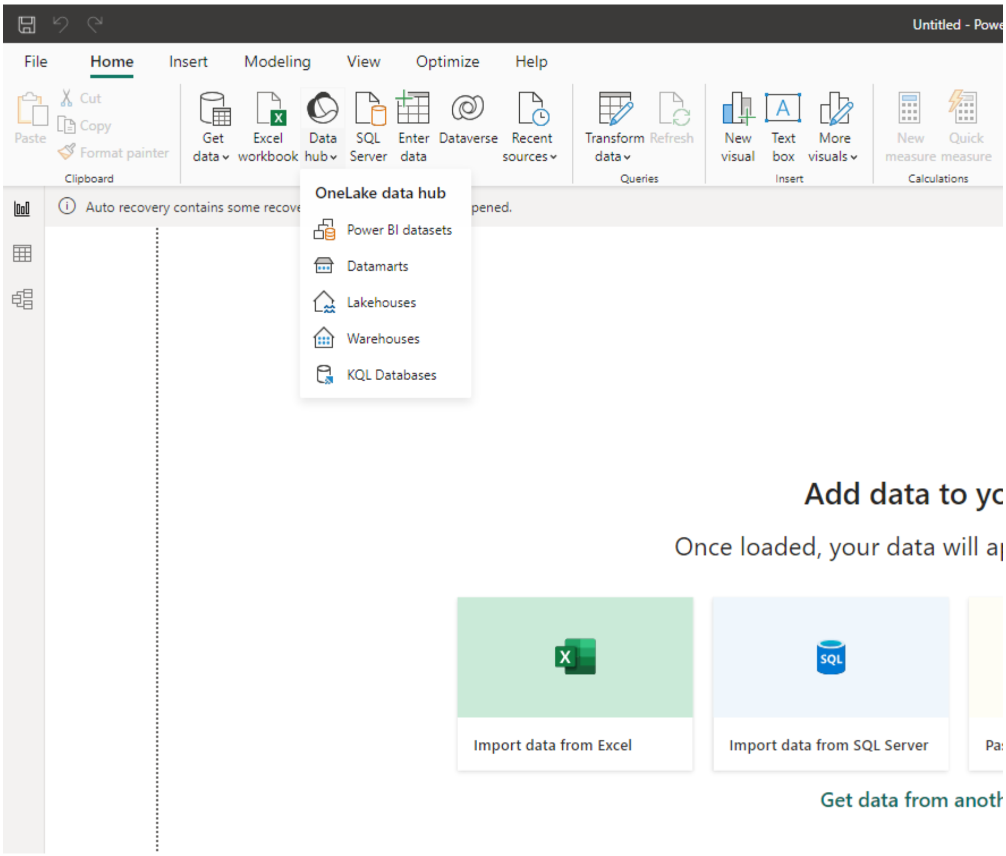
☑ Start my alert

Create alert

When you select Continue, it will create a Data Activator reflex and connect it to the data in your Power BI dataset. Data Activator will automatically start monitoring your data and notify you when the condition is met.

OneLake data hub in Power BI Desktop

Connectivity to Fabric items (Datamarts, Lakehouses, Warehouses, KQL Databases) via the OneLake data hub is now Generally Available, following the earlier announcement of Fabric's General Availability.



Power BI connector updates

AMAZON ATHENA (CONNECTOR UPDATE)

The Amazon Athena connector has been updated. In this Power BI connector release, it has eliminated an issue that prevented users from connecting to Athena using AAD authentication without a role parameter.

BQE CORE (CONNECTOR UPDATE)

The BQE Core connector has been updated. It has now added 'Target Utilization' and 'Birth Date' properties to the Employee table.

DATABRICKS (CONNECTOR UPDATE)

The Databricks connector has been updated. In this release, AAD authoring support has been added for Databricks AWS workspace.

INWINK (NEW CONNECTOR)

This update witnesses the release of the inwink connector. You may now visualise your inwink data and create dashboards using the inwink connector for Power BI. Further, you may combine data in inwink with data from other business apps (like Salesforce, SAP, *etc.*) to drive better decision making and analyse the performance of your events and communities.

STARBURST ENTERPRISE (CONNECTOR UPDATE)

The Starburst Enterprise connector has been updated. All changes are backwards compatible:

- added support for cancelling queries using 'CancelQueryExplicity'
- added 'Use default proxy' checkbox
- added support for Starburst Galaxy OAuth authentication
- removed 'Safe metadata read (optional)' from the connector dialog. It is now enabled by default for all queries.

SNOWFLAKE (CONNECTOR UPDATE)

The Snowflake connector has been updated to enable cancellation of a Snowflake query when the operation is cancelled in the Power BI user interface.

EDUFRAME REPORTING (CONNECTOR UPDATE)

The Eduframe Reporting connector has been updated. Modifications include:

- added teacher enrolments
- added teacher roles
- added catalog variants
- added planning event materials
- added meeting materials
- fixed the loading of teachers.

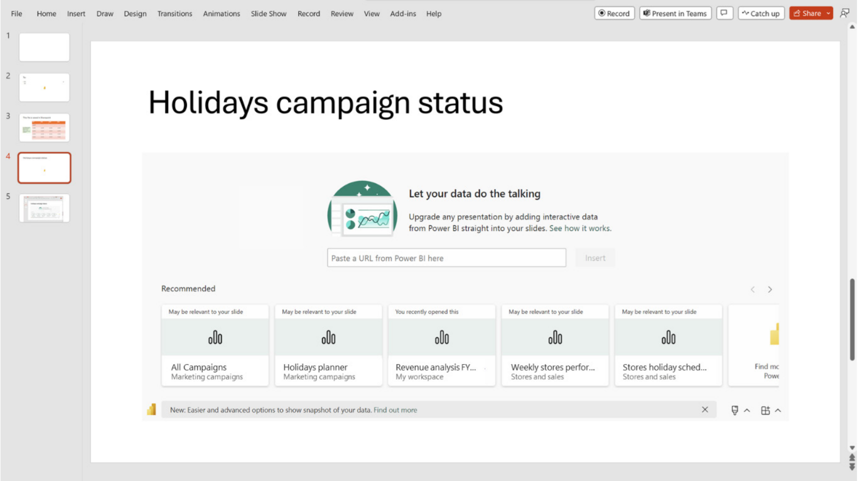
Storytelling in PowerPoint: suggested content

This update sees a new feature, a Power BI add-in for PowerPoint that makes it easier to find and insert the Power BI reports you need for your presentation.

When you add the Power BI add-in to your slide, the add-in automatically scans the title of the slide and suggests Power BI content that might be relevant. For example, if the title of your slide is "Holiday campaign status", the add-in will show you a list of Power BI reports that contain data about campaigns.

The add-in also shows you reports you've recently visited in Power BI, so you can easily access reports you're working on or frequently use.

It should be noted that if the report you wanted to add doesn't appear in the list of suggestions, you can just copy the report's URL and paste it into the input box, as you've always done.



Holidays campaign status

Let your data do the talking

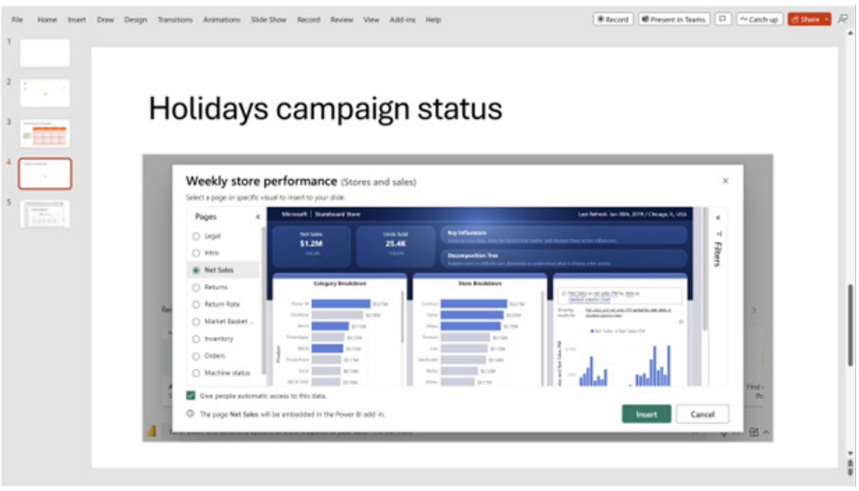
Upgrade any presentation by adding interactive data from Power BI straight into your slides. See how it works.

Paste a URL from Power BI here

May be relevant to your slide	May be relevant to your slide	You recently opened this	May be relevant to your slide	May be relevant to your slide
All Campaigns Marketing campaigns	Holidays planner Marketing campaigns	Revenue analysis FY... My workspace	Weekly stores perfor... Stores and sales	Stores holiday sched... Stores and sales

New: Easier and advanced options to show snapshot of your data. Find out more

Once you find the report you want to insert, you can choose to insert a complete page or a specific visual from the report. You can also update filters or slicers before you insert your selection. Filter and slicer values are saved together with the report or visual you picked.



Holidays campaign status

Weekly store performance (Stores and sales)

Select a page or specific visual to insert to your slide.

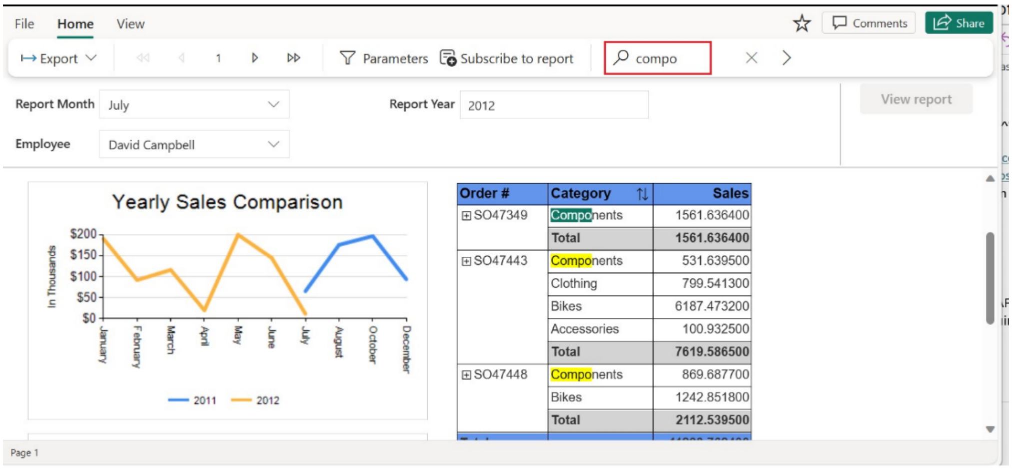
☑ Give people automatic access to this data.

The page Net Sales will be embedded in the Power BI add-in.

If you have permission to grant others access to this report, you can also select the 'Give people automatic access to this data' checkbox, so that when others in your organisation view this presentation, they'll be able to see the data you've just added to the slide.

Ability to search and sort in paginated reports

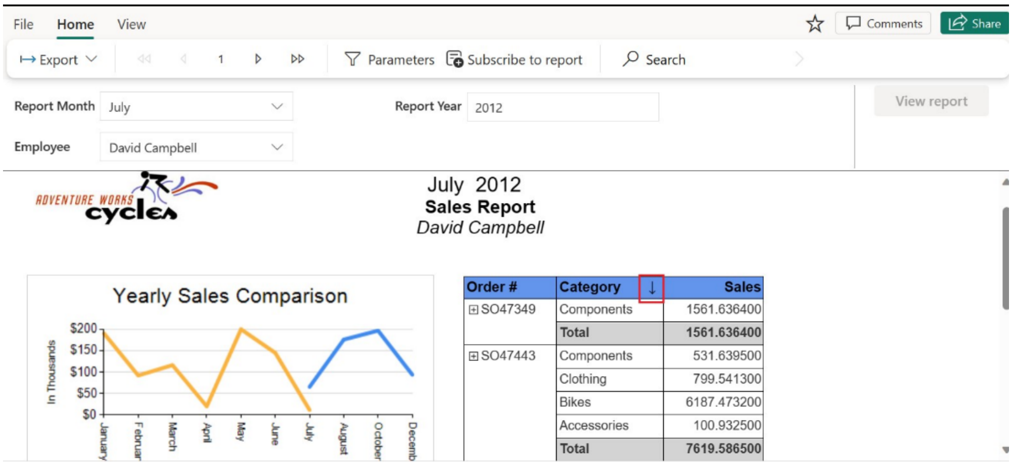
This update introduces the ability to search for content across all pages of a paginated report in the Power BI Service. The paginated report can either be published to the Service or can be authored on the Power BI Service. You may specify any required parameters to view the report before searching. The search works across all pages of the report.



Report Month: July
Report Year: 2012
Employee: David Campbell

Order #	Category	Sales
⊞SO47349	Components	1561.636400
	Total	1561.636400
⊞SO47443	Components	531.639500
	Clothing	799.541300
	Bikes	6187.473200
	Accessories	100.932500
	Total	7619.586500
⊞SO47448	Components	869.687700
	Bikes	1242.851800
	Total	2112.539500

You can also sort on a column by simply clicking on the sort icon when the report is rendered in the Power BI Service.



Report Month: July
Report Year: 2012
Employee: David Campbell

July 2012
Sales Report
David Campbell

Order #	Category ↓	Sales
⊞SO47349	Components	1561.636400
	Total	1561.636400
⊞SO47443	Components	531.639500
	Clothing	799.541300
	Bikes	6187.473200
	Accessories	100.932500
	Total	7619.586500

It should be noted that both sort and search are available only in the Web layout.

Power BI custom visuals authentication API

This API allows Custom Visuals to obtain Azure Active Directory (AAD) access tokens through single sign-on (SSO), facilitating secure and efficient user-contextual operations. The API will be controlled by a global administrator setting.

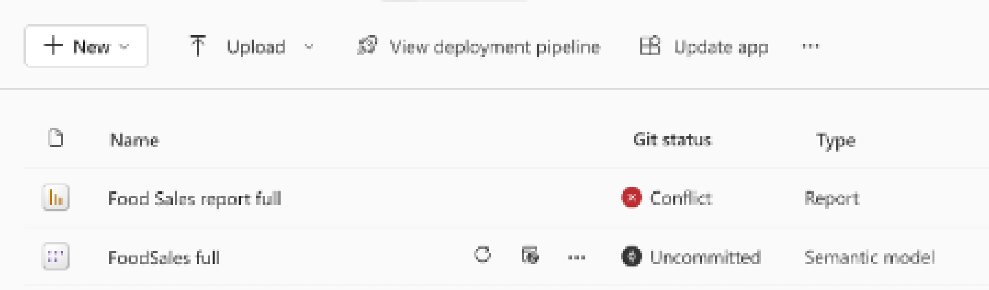
Dynamic drill down API

This feature allows the visual to enable or disable the drill feature dynamically using an API call.

When the drill feature is enabled, all the functionalities of drilldown and expand / collapse features are available. These functionalities include API calls, context menu commands, header drill buttons and support for hierarchy data. When the drill feature is disabled, these functionalities aren't available. I would never have figured that.

Git integration conflict resolution

When working with Git integration in Fabric, conflicts may occur if the same items are edited in both the Fabric workspace and in the connected Git repository. To avoid any mistakes, items in conflict are marked as such, and the conflicts must be resolved before any further changes can be made. In these cases, it is up to you to decide which changes to keep and which to discard.



Name	Git status	Type
Food Sales report full	Conflict	Report
FoodSales full	Uncommitted	Semantic model

To make things easier, Microsoft has added the option to resolve conflicts and select which content to keep directly within the Fabric workspace, so that you don't have to navigate elsewhere or revert to previous versions.

Accessed by updating items from the 'Source control' pane, resolving conflicts is as simple as selecting whether to accept incoming changes or keep the current content for each item in conflict. Once the selections are made, all items may be merged and updated, and any further changes can be made in the workspace.

Resolving conflicts can also be undertaken by reverting either the workspace or the Git branch to a previous state, or directly in Git if you need to further inspect what changes were made and which version to choose.

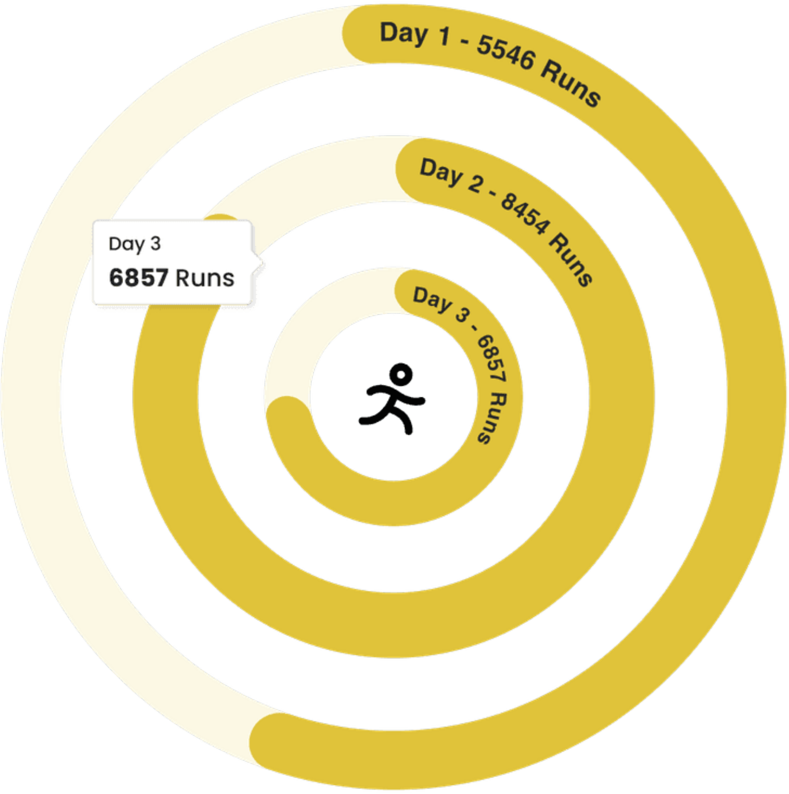
New visuals in AppSource

This month's new visuals include:

- Risk Matrix by Profitbase
- PowerBI User Activity Tracker
- ilionx ZorgControl KPI Gauge
- Tabulator for PowerBI
- Funnel Charts.

Activity Gauge by Powerviz

The Activity Gauge by Powerviz is a visual used to measure the progress against a set target or goal for multiple categories. It is typically presented in the form of a Radial Bar chart / gauge, displaying a range of values or percentages. This visual is commonly used for project management, performance tracking and monitoring Key Performance Indicators (KPIs).



Key features include:

- **Gauge customisation:** shape, style and add icons for a unique design
- **Targets:** choose from options or use a custom field to set a target
- **Data colours:** offers seven [7] schemes and 30+ colour palettes
- **Centre circle:** customise the centre circle with text, icons, measures and images
- **Interactive mouseover:** hover for 'Interactive Info Display'
- **Conditional formatting:** create rules based upon measure or category rules.

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

Business use cases might include:

- **Project management:** track task progress for efficient management
- **Financial dashboard:** gain insights into key financial metrics
- **Employee performance:** evaluate contributions and identify areas for improvement.

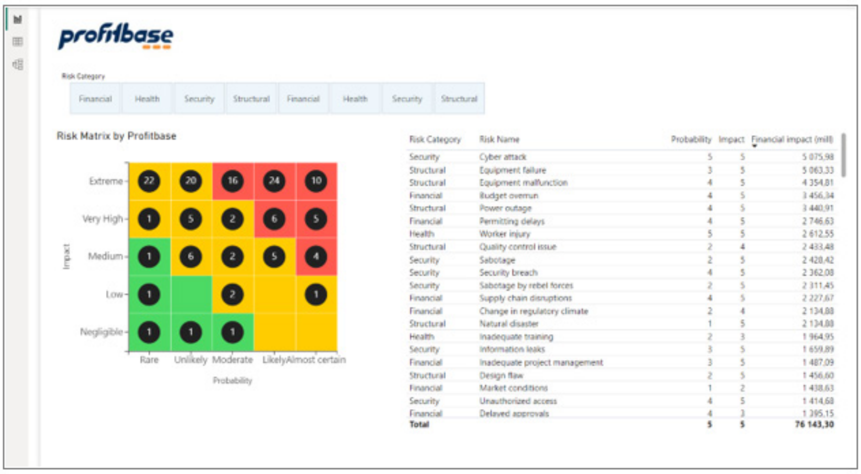
Activity Gauge is available in AppSource.

Navigate Financial Risks and Rewards with Profitbase's Power BI Visualisation

Profitbase's new Risk Matrix visualisation for Power BI provides businesses with a better way to visualise and assess potential risks. This helps them handle and minimise potential damage, in addition to discovering opportunities that the business may benefit from.

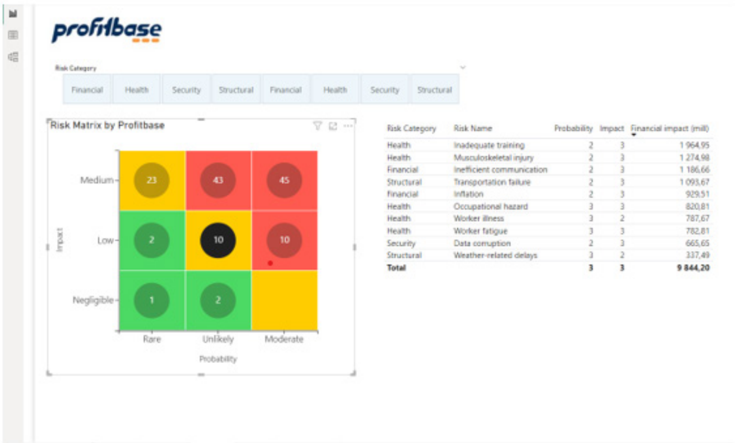
Advantages of the Risk Matrix by Profitbase:

- **Efficient risk assessment:** evaluate risks by impact and likelihood
- **Focused prioritisation:** spotlight and tackle high-priority risks, focusing your resources effectively
- **Strategic allocation:** employ the tool to guide resource distribution
- **Harmonised communication:** establish a common understanding among teams and stakeholders
- **Flexible customisation:** personalise the matrix size, from 5×5 to 3×3 or 3×5, and colours may be changed to align with your company's branding
- **Dynamic visualisation:** adjustable bubble sizes animate your data, offering a clear financial perspective.



Risk Category	Risk Name	Probability	Impact	Financial impact (mill)
Security	Cyber attack	5	5	5 075,98
Structural	Equipment failure	3	5	5 063,33
Structural	Equipment malfunction	4	5	4 354,81
Financial	Budget overrun	4	5	3 456,34
Structural	Power outage	4	5	3 440,91
Financial	Permitting delays	4	5	2 746,63
Health	Worker injury	5	5	2 612,55
Structural	Quality control issue	2	4	2 433,48
Security	Sabotage	2	5	2 428,42
Security	Security breach	4	5	2 362,08
Security	Sabotage by rebel forces	2	5	2 311,45
Financial	Supply chain disruptions	4	5	2 227,67
Financial	Change in regulatory climate	2	4	2 134,88
Structural	Natural disaster	1	5	2 134,88
Health	Inadequate training	2	3	1 964,95
Security	Information leaks	3	5	1 659,89
Financial	Inadequate project management	3	5	1 487,09
Structural	Design flaw	2	5	1 456,60
Financial	Market conditions	1	2	1 438,63
Security	Unauthorized access	4	5	1 414,68
Financial	Delayed approvals	4	3	1 395,15
Total		**5**	**5**	**76 143,30**

Figure 1: Default layout with 5x5 matrix



Risk Category	Risk Name	Probability	Impact	Financial impact (mill)
Health	Inadequate training	2	3	1 964,95
Health	Musculoskeletal injury	2	3	1 274,98
Financial	Inefficient communication	2	3	1 186,66
Structural	Transportation failure	2	3	1 093,67
Financial	Inflation	2	3	929,51
Health	Occupational hazard	3	3	820,81
Health	Worker illness	3	2	787,67
Health	Worker fatigue	3	3	782,81
Security	Data corruption	2	3	665,65
Structural	Weather-related delays	3	2	337,49
Total		**3**	**3**	**9 844,20**

Figure 2: Adjusted layout with 3x3 layout. Click a bubble to filter other visuals.

Intuitive multi-level Donut charts for your reports

Drill Down Donut PRO for Power BI allows you to use multi-level Donut (Doughnut) charts. All interactions take place as on-chart interactions, ensuring a quick and easy data exploration experience. You can control every aspect of the chart through various customisation and also select from multiple chart types (Donut, Pie, Gauge). Drill Down Donut PRO is available from AppSource.

Main features:

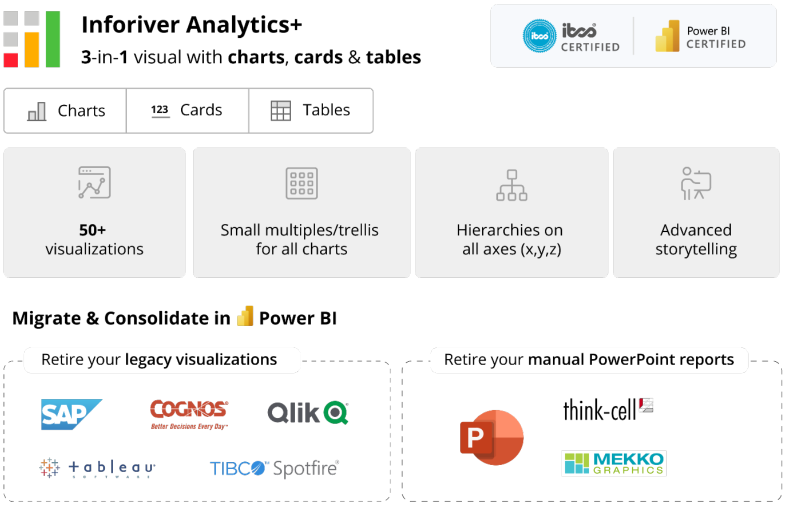
- **Adjustable 'Others' slice:** set the number of visible slices and group the rest
- **On-chart interactions:** no need to learn hidden controls
- **Custom ToolTips:** select ZoomCharts custom ToolTip or Power BI's built-in ToolTip
- **Full customisation:** you may modify every slice, label and legend
- **Desktop and mobile device navigation:** explore charts the same way on any device.

Popular use cases include:

- **Sales and marketing:** measuring campaign performance and research results
- **Human resources:** staff composition, salary distribution, performance data
- **Accounting and finance:** income and expense analysis, billings, creditors and debtors
- **Project management:** risk distribution, resource allocation.

Inforiver Analytics+ 3.2

Inforiver Analytics+ (50+ charts, cards and tables) by Lumel is now a complete No-Code storytelling visualisation platform that is designed to help customers migrate legacy dashboards from Tableau, Spotfire, Qlik, Cognos, SAP Lumira or their static PowerPoint dashboards from Think-cell or Mekko Graphics and consolidate them within Microsoft Power BI.


Inforiver Analytics+
3-in-1 visual with charts, cards & tables
ibcs CERTIFIED
Power BI CERTIFIED
Charts
123 Cards
Tables
50+ visualizations
Small multiples/trellis for all charts
Hierarchies on all axes (x,y,z)
Advanced storytelling
Migrate & Consolidate in Power BI
Retire your legacy visualizations
SAP
COGNOS
Qlik
tableau
TIBCO Spotfire
Retire your manual PowerPoint reports
think-cell
MEKKO GRAPHICS

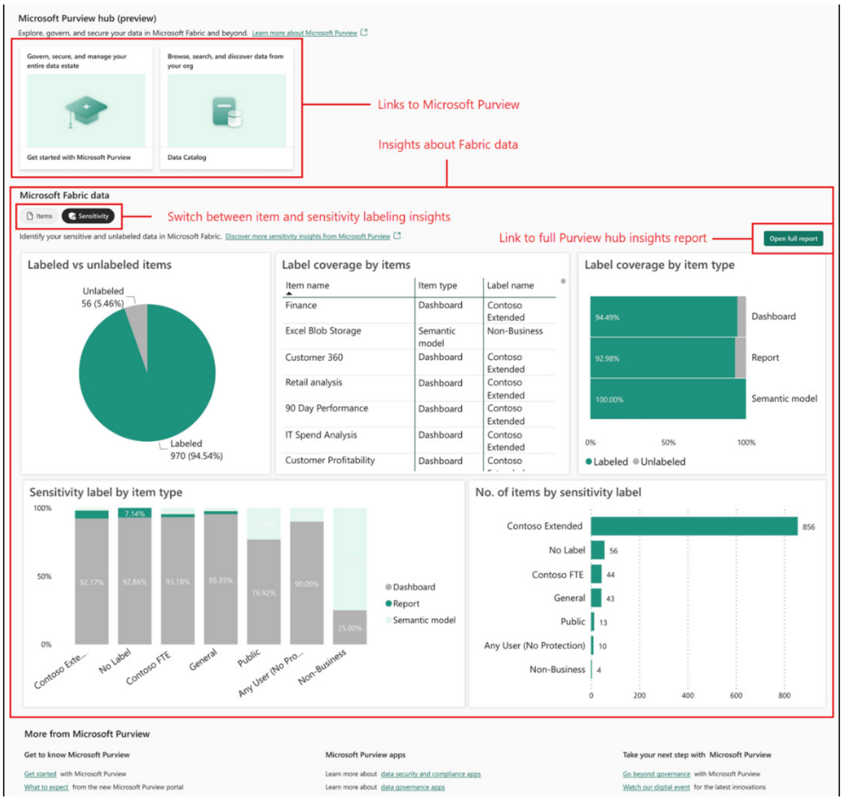
Benefits include:

- Small Multiple and Trellis, similar to Tableau and Spotfire
- "Visual PivotTable" for hierarchical data on all three axes (x, y and z) like Spotfire
- split and group measures into Trellis panels like Tableau
- storytelling features like Think-cell and Mekko Graphics:
  - Dynamic Deviations (CAGR)
  - Annotations
  - Series connector lines
  - Automatic series and categorical sorting
- Trellis Tables with Excel-like formatting
- Advanced visualisations:
  - Box and Whisker Plot: dynamic and pre-calculated quartiles
  - Marimekko Chart (variable width Column or Cascade chart): stacking and 100% scales
  - Advanced Waterfall: measure-based Interim, Stacked & Dual Breakdown, Variance and Combination
- Special Charts for Story Telling:
  - Slope graph
  - Lollipop and Dumbbell
  - Dot Plot,
  - Arrow & Range Plots.

Multivariate visualisations (Bubble / Scatter and Radar / Polar) and Pie / Donut types will be added to this visual soon.

The Microsoft Purview hub in Fabric for creators in Preview

The Microsoft Purview hub is a centralised page in Fabric that helps Fabric administrators and users gain insights about their Fabric data estate. It contains reports that provide insights into sensitive data and item endorsement and serves as a gateway to more advanced capabilities in Microsoft Purview. Following the public Preview release of the Microsoft Purview hub for Fabric tenant admins back in May 2023, the hub is now available in public Preview for all Fabric users.



Microsoft Purview hub (preview)

Explore, govern, and secure your data in Microsoft Fabric and beyond. Learn more about Microsoft Purview

Govern, secure, and manage your entire data estate

Get started with Microsoft Purview

Browse, search, and discover data from your org

Data Catalog

Links to Microsoft Purview

Insights about Fabric data

Microsoft Fabric data

Switch between item and sensitivity labeling insights

Identify your sensitive and unlabeled data in Microsoft Fabric. Discover more sensitivity insights from Microsoft Purview

Link to full Purview hub insights report

Label coverage by items

Item name	Item type	Label name
Finance	Dashboard	Contoso Extended
Excel Blob Storage	Semantic model	Non-Business
Customer 360	Dashboard	Contoso Extended
Retail analysis	Dashboard	Contoso Extended
90 Day Performance	Dashboard	Contoso Extended
IT Spend Analysis	Dashboard	Contoso Extended
Customer Profitability	Dashboard	Contoso

More from Microsoft Purview

Get to know Microsoft Purview

Get started with Microsoft Purview

What to expect from the new Microsoft Purview portal

Microsoft Purview apps

Learn more about data security and compliance apps

Learn more about data governance apps

Take your next step with Microsoft Purview

Go beyond governance with Microsoft Purview

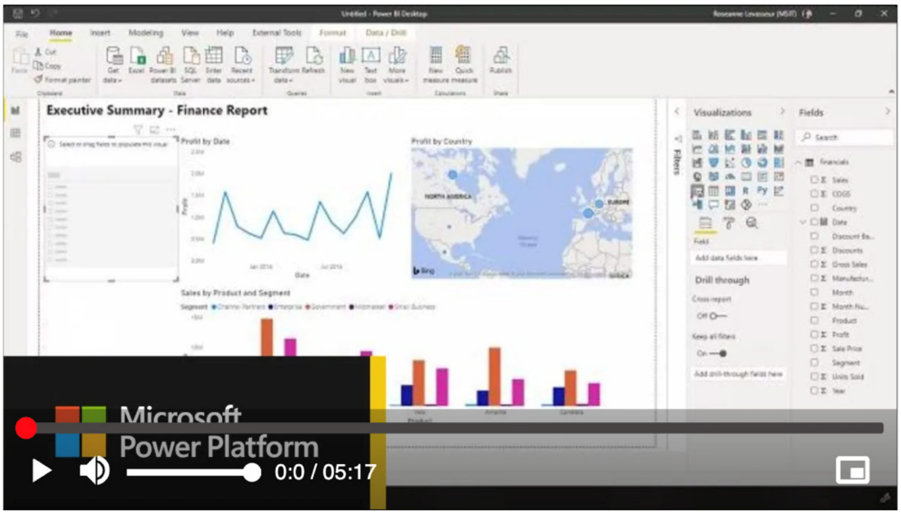
Watch our digital event for the latest innovations

VideoPlayer Visual

The VideoPlayer visual for Power BI Desktop has now joined us. This new custom visual allows you to integrate videos into your Power BI reports, enhancing your data storytelling capabilities. With VideoPlayer, you can enrich your reports with dynamic multimedia content to engage your audience effectively.

Key features:

- **video integration:** easily embed videos from various sources, such as YouTube, Vimeo (including password protected videos) and many others, directly into your Power BI reports
- **interactive playback:** control video playback using intuitive play, pause and seeking functionality, ensuring your audience can interact with the content as needed
- **customisation:** modify the appearance of the VideoPlayer to match your report's style and design, including changing the video player's dimensions and controls
- **data-driven video selection:** utilise data-driven techniques to determine which video to play based on your dataset, making your reports more insightful.



Ridgeline Plot by Powerviz

A Ridgeline Plot or Joy Plot is an advanced Power BI visual that visually represents numerical variable distributions across multiple categories on a continuous axis. The overlapping mountain ranges allow simple comparison. Ridgeline Plot effectively highlights patterns of distribution change across different categories, providing insights into the variability and trends within the data.

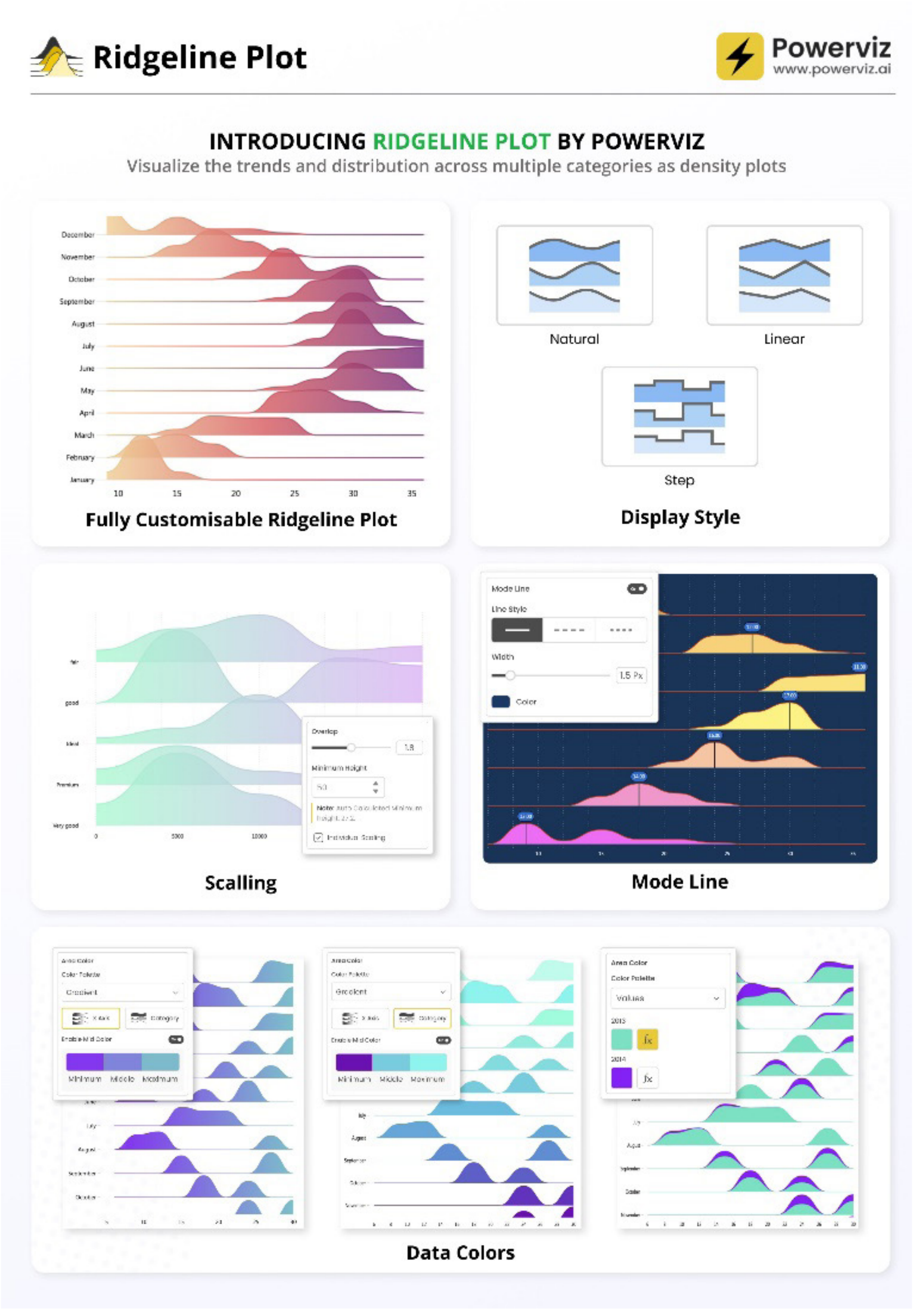
Key features include:

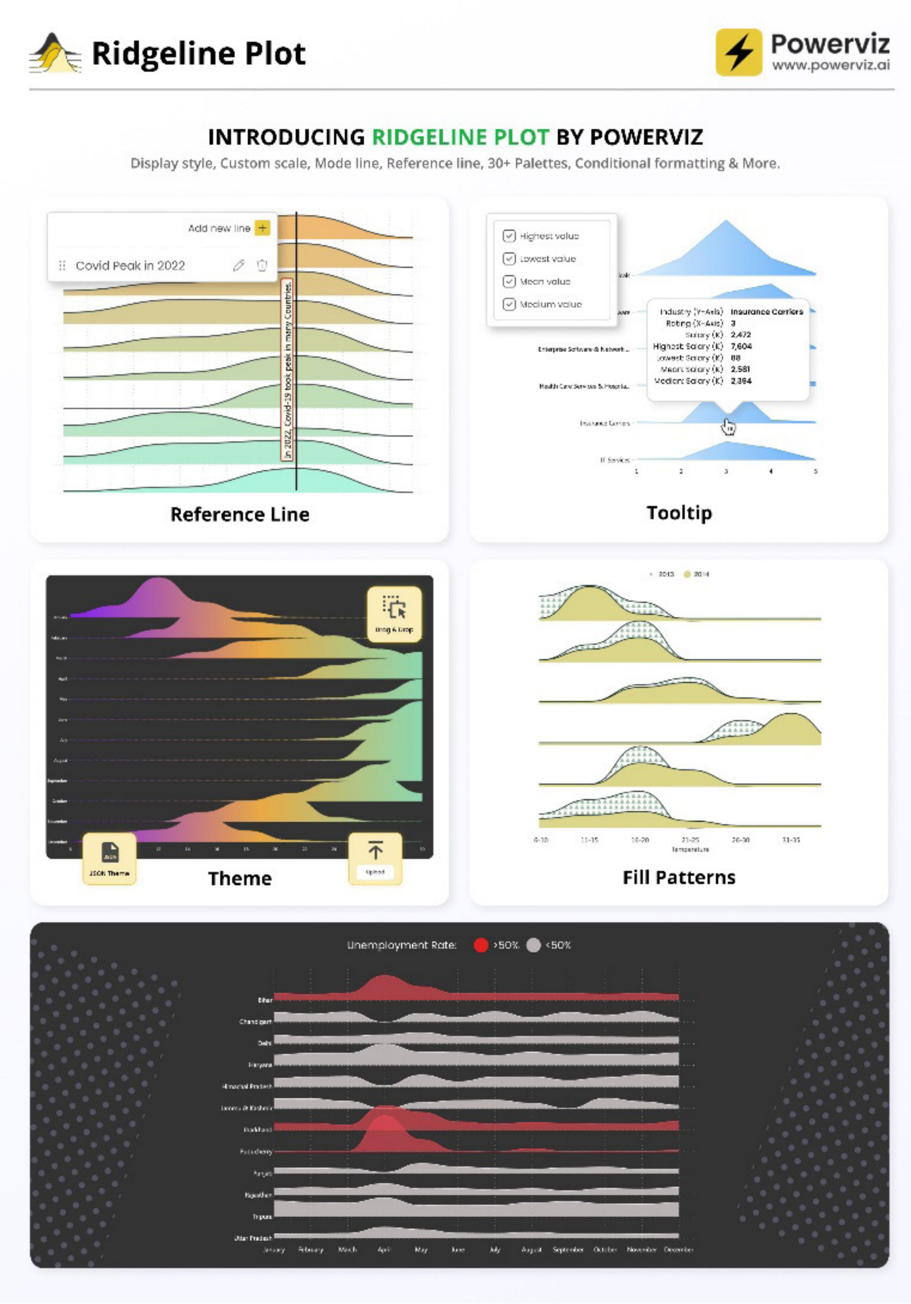
- **multiple distribution:** ability to plot more than one value in measures
- **Ridge style:** different lines and marker styles available
- **scaling:** control **y**-axis scaling and overlaps
- **colours:** apply colour schemes across **x**-axis, **y**-axis and based upon FX rules
- **ranking:** filter Top / Bottom **N** values, with “Others”
- **reference line:** highlight ranges or important data points on **x** and **y** axes
- **mode line:** emphasise peak points and display labels
- **custom ToolTip:** add highest, lowest, mean and median points without any **DAX**
- **themes:** save designs and share with JSON files.

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

Business use cases might include time series data, statistical analysis and data distribution.

You can download Ridgeline Plot for free from AppSource.





Ridgeline Plot

Powerviz
www.powerviz.ai

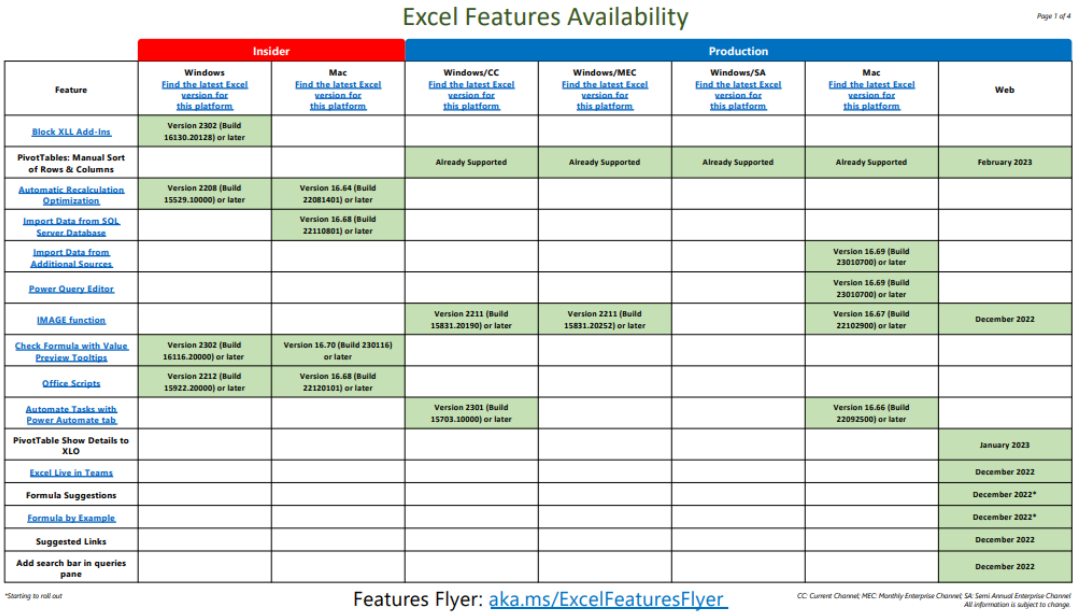
INTRODUCING RIDGELINE PLOT BY POWERVIZ

Display style, Custom scale, Mode line, Reference line, 30+ Palettes, Conditional formatting & More.

More next month.

New Features for Excel

Sadly, it’s all quiet on the Excel front this month, but don’t fret. We’re sure it will be back with a vengeance next month. In the meantime, just to keep you apprised here is the latest version of the grid with all the new features. Nonetheless, you can find the interactive links at aka.ms/ExcelFeaturesFlyer.



Excel Features Availability

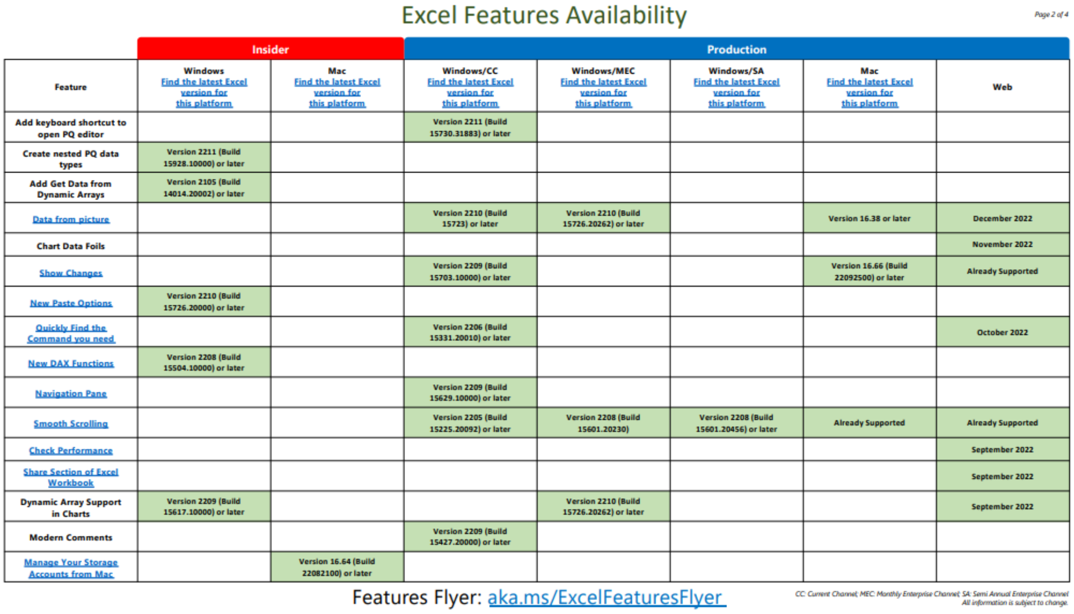
Page 1 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Block XLL Add-Ins	Version 2302 (Build 16130.20128) or later						
PivotTables: Manual Sort of Rows & Columns			Already Supported	Already Supported	Already Supported	Already Supported	February 2023
Automatic Recalculation Optimization	Version 2208 (Build 15529.10000) or later	Version 16.64 (Build 22081401) or later					
Import Data from SQL Server Database		Version 16.68 (Build 22110801) or later					
Import Data from Additional Sources						Version 16.69 (Build 23010700) or later	
Power Query Editor						Version 16.69 (Build 23010700) or later	
IMAGE function			Version 2211 (Build 15831.20190) or later	Version 2211 (Build 15831.20252) or later		Version 16.67 (Build 22102900) or later	December 2022
Check Formula with Value Preview Tooltips	Version 2302 (Build 16116.20000) or later	Version 16.70 (Build 230116) or later					
Office Scripts	Version 2212 (Build 15922.20000) or later	Version 16.68 (Build 22120101) or later					
Automate Tasks with Power Automate tab			Version 2301 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	
PivotTable Show Details to XLO							January 2023
Excel Live in Teams							December 2022
Formula Suggestions							December 2022*
Formula by Example							December 2022*
Suggested Links							December 2022
Add search bar in queries pane							December 2022

*Starting to roll out

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



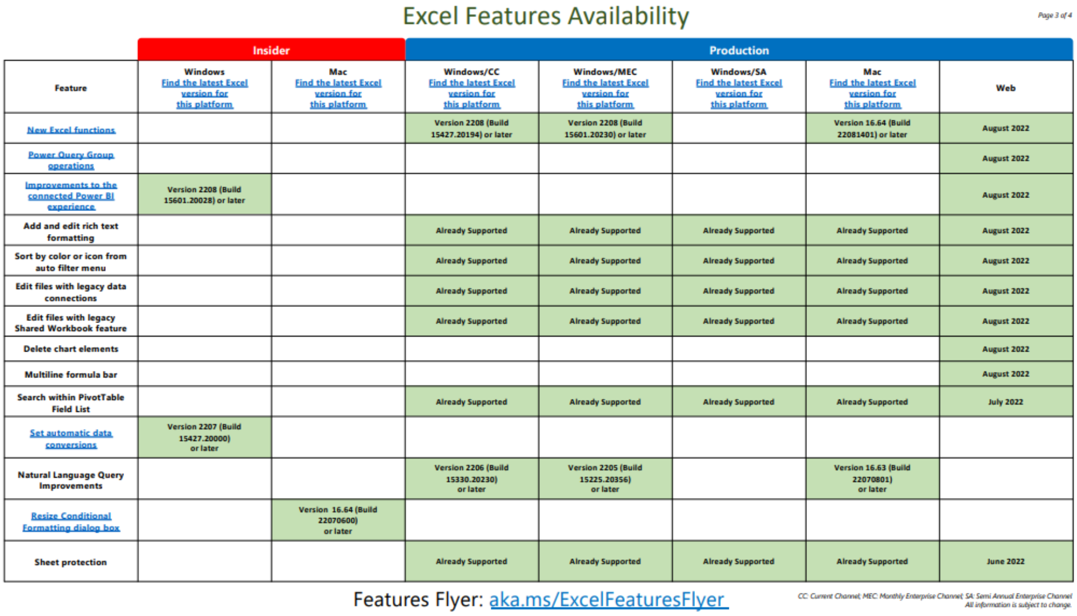
Excel Features Availability

Page 2 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Add keyboard shortcut to open PQ editor			Version 2211 (Build 15730.31883) or later				
Create nested PQ data types	Version 2211 (Build 15928.10000) or later						
Add Get Data from Dynamic Arrays	Version 2105 (Build 14014.20002) or later						
Data from picture			Version 2210 (Build 15723) or later	Version 2210 (Build 15726.20262) or later		Version 16.38 or later	December 2022
Chart Data Foils							November 2022
Show Changes			Version 2209 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	Already Supported
New Paste Options	Version 2210 (Build 15726.20000) or later						
Quickly Find the Command you need			Version 2206 (Build 15331.20010) or later				October 2022
New DAX Functions	Version 2208 (Build 15504.10000) or later						
Navigation Pane			Version 2209 (Build 15629.10000) or later				
Smooth Scrolling			Version 2205 (Build 15225.20092) or later	Version 2208 (Build 15601.20230)	Version 2208 (Build 15601.20456) or later	Already Supported	Already Supported
Check Performance							September 2022
Share Section of Excel Workbook							September 2022
Dynamic Array Support in Charts	Version 2209 (Build 15617.10000) or later			Version 2210 (Build 15726.20262) or later			September 2022
Modern Comments			Version 2209 (Build 15427.20000) or later				
Manage Your Storage Accounts from Mac		Version 16.64 (Build 22082100) or later					

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



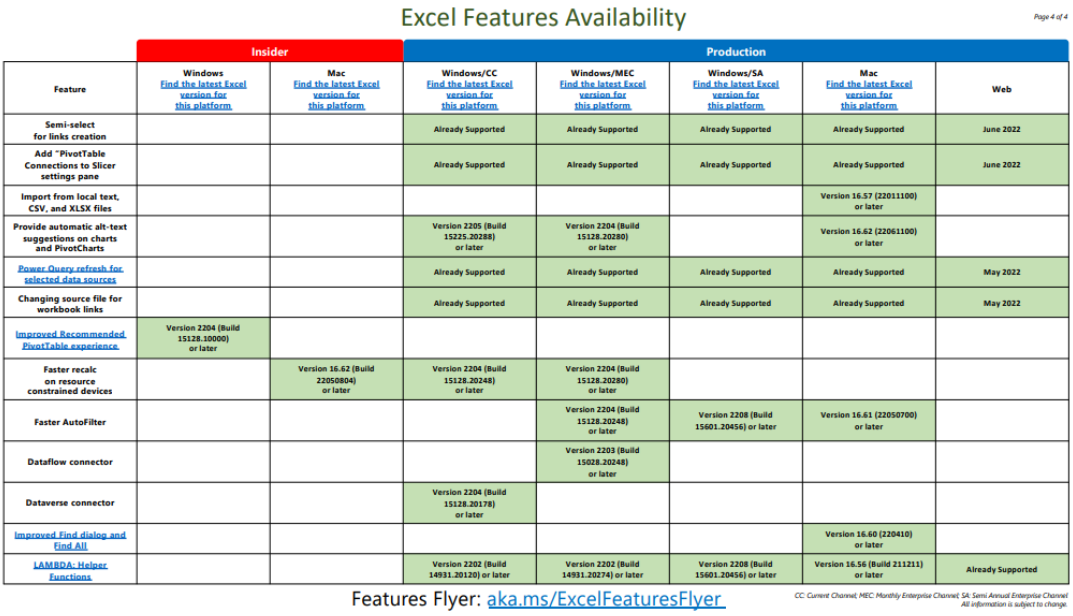
Excel Features Availability

Page 3 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
New Excel functions			Version 2208 (Build 15427.20194) or later	Version 2208 (Build 15601.20230) or later		Version 16.64 (Build 22081401) or later	August 2022
Power Query Group operations							August 2022
Improvements to the connected Power BI experience	Version 2208 (Build 15601.20028) or later						August 2022
Add and edit rich text formatting			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Sort by color or icon from auto filter menu			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy data connections			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy Shared Workbook feature			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Delete chart elements							August 2022
Multiline formula bar							August 2022
Search within PivotTable Field List			Already Supported	Already Supported	Already Supported	Already Supported	July 2022
Set automatic data conversions	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20356) or later		Version 16.63 (Build 22070801) or later	
Resize Conditional Formatting dialog box		Version 16.64 (Build 22070600) or later					
Sheet protection			Already Supported	Already Supported	Already Supported	Already Supported	June 2022

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



Excel Features Availability

Page 4 of 4

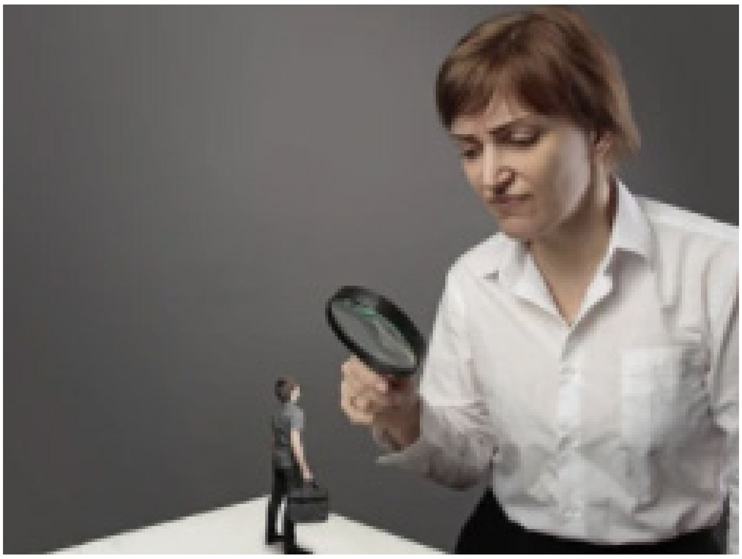
	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Semi-select for links creation			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Add "PivotTable Connections to Slicer settings pane			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Changing source file for workbook links			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Improved Recommended PivotTable experience	Version 2204 (Build 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20178) or later				
Improved Find dialog and Find All						Version 16.60 (220410) or later	
LAMBDA: Helper Functions			Version 2202 (Build 14931.20120) or later	Version 2202 (Build 14931.20274) or later	Version 2208 (Build 15601.20456) or later	Version 16.56 (Build 211211) or later	Already Supported

Features Flyer: aka.ms/ExcelFeaturesFlyer

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

Back next month, we're sure.

The A to Z of Excel Functions: MINUTE



This function returns the minutes of a time value (expecting a serial number). The **MINUTE** is given as an integer, ranging from zero [0] to 59. The **MINUTE** function employs the following syntax to operate:

MINUTE(serial_number)

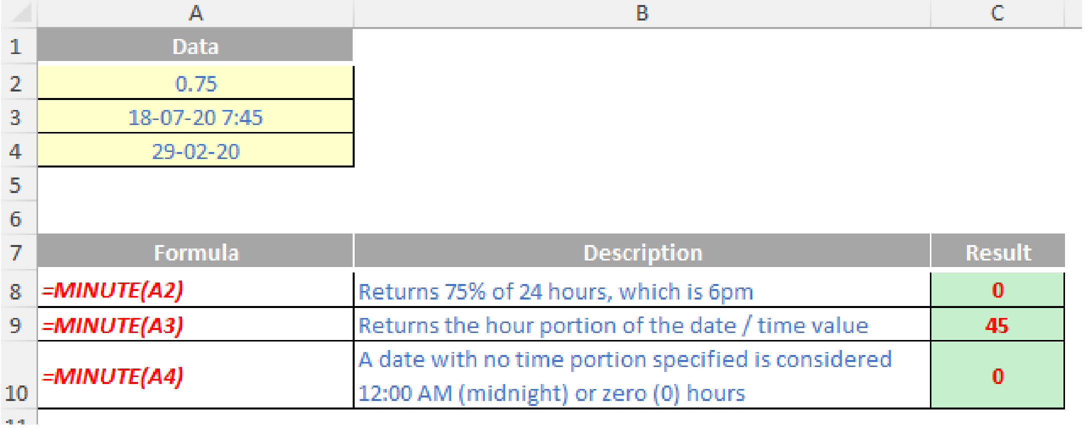
The **MINUTE** function has the following argument:

- **serial_number:** this is required and represents the time that contains the minute you want to find. Times may be entered as text strings within quotation marks (*e.g.* "6:45 PM"), as decimal numbers (for example, 0.78125, which represents 6:45 PM, or as results of other formulas or functions (for example, **TIMEVALUE("6:45 PM")**).

It should be further noted that:

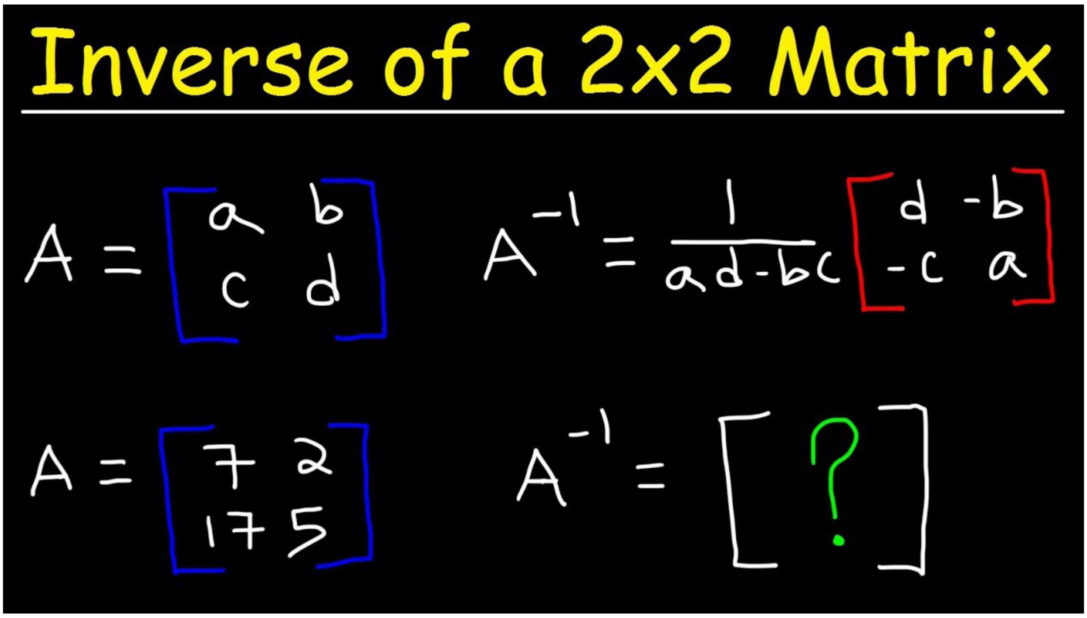
- time values are a portion of a date value and are represented by a decimal number (*e.g.* 12:00 PM is represented as 0.5 because it is half of a day).

Please see some examples below:



	A	B	C
1	Data		
2	0.75		
3	18-07-20 7:45		
4	29-02-20		
5			
6			
7	Formula	Description	Result
8	=MINUTE(A2)	Returns 75% of 24 hours, which is 6pm	0
9	=MINUTE(A3)	Returns the hour portion of the date / time value	45
10	=MINUTE(A4)	A date with no time portion specified is considered 12:00 AM (midnight) or zero (0) hours	0

The A to Z of Excel Functions: MINVERSE


Inverse of a 2x2 Matrix
$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
$A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$
$A = \begin{bmatrix} 7 & 2 \\ 17 & 5 \end{bmatrix}$
$A^{-1} = \begin{bmatrix} ? \end{bmatrix}$

In mathematics, especially in areas such as linear algebra, matrices may be used to solve simultaneous equations. For the record, a matrix is not just a movie it's a rectangular arrangement of **m** x **n** elements, in the dimensions of **m** rows by **n** columns, *e.g.* a matrix **A** *(say)* may be represented as

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}_{m \times n}$$

It is often written in compact form as

$$A = [a_{ij}]_{m \times n}, 1 \leqslant i < m, 1 \leqslant j < n$$

An **n** x **n** square matrix **A** is called invertible (also non-singular or non-degenerate), if there exists an **n** x **n** square matrix **B** such that

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$$

where $\mathbf{I}_n$ denotes the **n** x **n** identity matrix and the multiplication used is ordinary matrix multiplication. If this is the case, then the matrix **B** is uniquely determined by **A**, and is called the (multiplicative) inverse of **A**, denoted by $\mathbf{A}^{-1}$. Matrix inversion is the process of finding the matrix **B** that satisfies the prior equation for a given invertible matrix **A**.

A square matrix that is not invertible is called singular or degenerate. A square matrix is singular if and only if its determinant is zero. Singular matrices are rare in the sense that if a square matrix's entries are randomly selected from any finite region on the number line or complex plane, the probability that the matrix is singular is zero [0], that is, it will "almost never" be singular. Non-square matrices (**m** x **n** matrices for which **m** ≠ **n**) do not have an inverse. However, in some cases such a matrix may have a left inverse or right inverse. If **A** is **m** x **n** and the rank of **A** is equal to **n** (**n** ≤ **m**), then **A** has a *left* inverse, an **n** x **m** matrix **B** such that $\mathbf{BA} = \mathbf{I}_n$. If **A** has rank **m** (**m** ≤ **n**), then it has a *right* inverse, an **n** x **m** matrix **B** such that $\mathbf{AB} = \mathbf{I}_m$.

The Excel function **MINVERSE** returns the inverse matrix for a matrix stored in an array. It has the following syntax:

MINVERSE(array)

where:

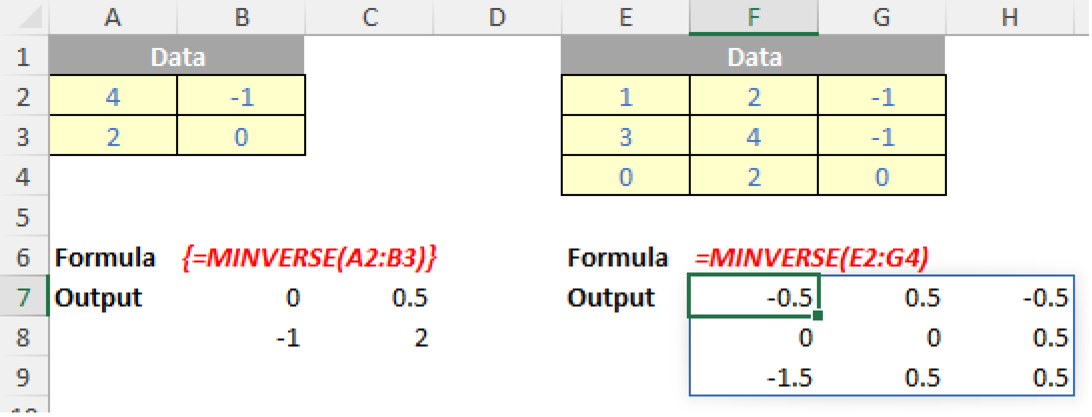
- **array** is required, and represents a numerical **array** with an equal number of rows and columns.

It should be noted that:

- **array** may be given as:
  - a cell range, *e.g.* **A1:C3**
  - an **array** constant, such as **{1,2,3;4,5,6;7,8,9}**
  - a name to either of these
- **MINVERSE** returns the *#VALUE!* error when:
  - any cells in array are empty or contain text
  - **array** does not have an equal number of rows and columns
- matrix determinants are generally used for solving systems of mathematical equations that involve several variables
- **MINVERSE** is calculated with an accuracy of approximately 16 digits, which may lead to a small numeric error when the calculation is not complete. For example, the determinant of a singular matrix may differ from zero by 1E-16
- some square matrices cannot be inverted and will return the *#NUM!* error value with **MINVERSE**. The determinant for a non-invertible matrix is zero [0].

If you have a current version of Microsoft 365, then you can simply enter the formula in the top-left-cell of the output range, then press **ENTER** to confirm the formula as a dynamic array formula. Otherwise, the formula must be entered as a legacy array formula by first selecting the output range, entering the formula in the top-left-cell of the output range, and then pressing **CTRL + SHIFT + ENTER** to confirm it. Excel inserts curly brackets ("braces") at the beginning and end of the formula.

As an example:



	A	B	C	D	E	F	G	H
1	Data				Data			
2	4	-1			1	2	-1	
3	2	0			3	4	-1	
4					0	2	0	
5								
6	Formula	{=MINVERSE(A2:B3)}			Formula	=MINVERSE(E2:G4)		
7	Output		0	0.5	Output	-0.5	0.5	-0.5
8			-1	2		0	0	0.5
9						-1.5	0.5	0.5

The A to Z of Excel Functions: MIRR

The **Internal Rate of Return** (IRR) is the name given to the discount rate that makes the Net Present Value (NPV) of a range of cashflows zero. For example, if I invest \$100 now and receive \$121 back in two years' time this would give me an annual IRR of 10% since:

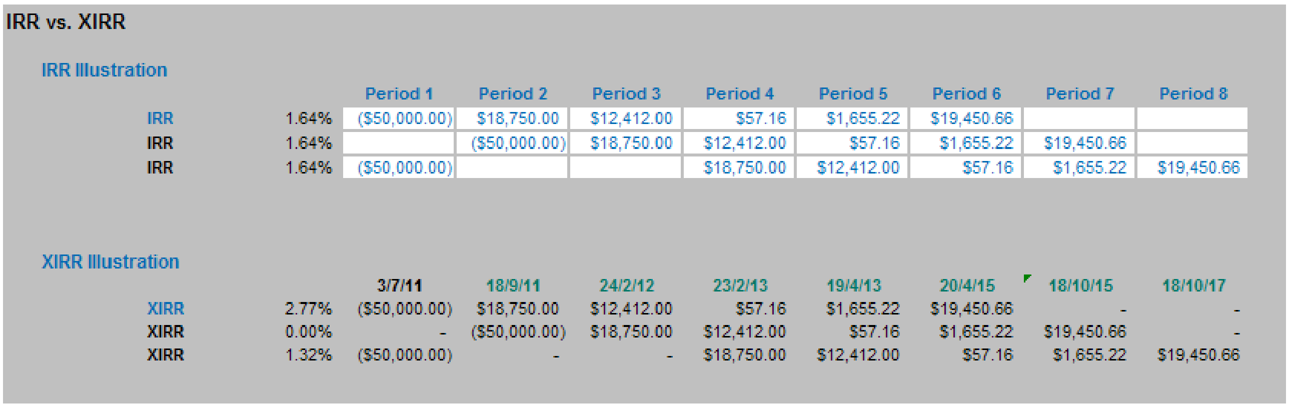
$$
\begin{aligned}
(\$100) + PV(\$121) &= (\$100) + \$121 / (1 + 10\%)^2 \\
&= (\$100) + \$100 \\
NPV &= 0
\end{aligned}
$$

It is often used to calculate cost of capital hurdle rates, bond rates, discount rates implicit in leases and Compound Annual Growth Rates, amongst other tasks.

Excel has two functions that estimate IRR:

- **IRR(values,** [guess]**)** returns the periodic IRR for a set of sequential cashflows occurring on a regular periodic basis. There must be at least one positive and one negative value in the range. **IRR** will then cycle through an iterative technique (up to 20 times) to try and find an answer which is accurate to within 0.00001%
- **XIRR(values, dates,** [guess]**)** returns the annual IRR (assuming a 365-day year) for a set of cashflows which may neither be sequential nor occur on a regular periodic basis. Again, there must be at least one positive and one negative value in the range. **XIRR** will then cycle though a similar iterative technique (this time up to 100 times) to try and find an answer which is accurate to within 0.000001%
- if the guess is not specified, Excel will assume that it is 10% (0.1) for both functions
- watch out with **XIRR**: in Excel 2003 and earlier versions it is found in the Analysis ToolPak (go to **Tools -> Add Ins…** or **ALT + T + I**). If the ToolPak has not been added in, using **XIRR** in a formula will give rise to the *#NAME?* error. If this applies to you, it's really time to invest in some new software.

Both are fraught with problems in practice and the associated Excel file provides several documented examples. However, before we discuss some of the problems with the functions, let me compare and contrast with a simple example:



IRR vs. XIRR

IRR Illustration

		Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8
IRR	1.64%	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66		
IRR	1.64%		(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	
IRR	1.64%	(\$50,000.00)			\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66

XIRR Illustration

		3/7/11	18/9/11	24/2/12	23/2/13	19/4/13	20/4/15	18/10/15	18/10/17
XIRR	2.77%	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	-	-
XIRR	0.00%	-	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	-
XIRR	1.32%	(\$50,000.00)	-	-	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66

In the illustration above, we consider three slightly different cashflows. The first one has the old faithful "hockey stick" projections of an outflow followed by five successive cash inflows. In our example, this would give us a periodic IRR of 1.64%. If the periods were months and we wished to convert this to annual rate then, using the compounding formula, this would be:

$$(1 + 1.64\%)^{12} -1 = 21.54\%.$$

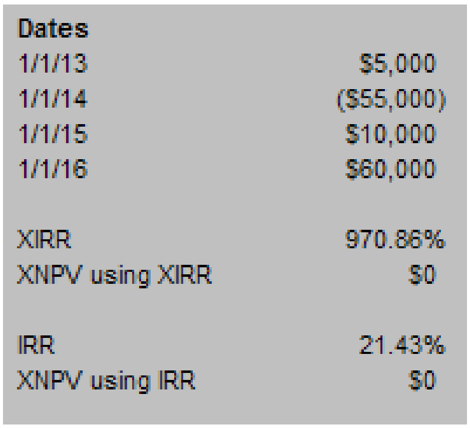
The second example for IRR gives the same rate – which is correct. Start dates for IRR are irrelevant: the only thing that matters is how long after the initial cashflow subsequent flows occur.

Unfortunately, the third scenario is not right and demonstrates a weakness in the IRR function. Here, the outflow still occurs in the first period, but then there is a two-month delay before the inflows are received. Clearly, this will impact both the NPV and the IRR. The IRR is unaffected however – *because blanks are ignored by the **IRR** function*. To get Excel to calculate the IRR correctly here, zeroes should be entered into the two blank cells.

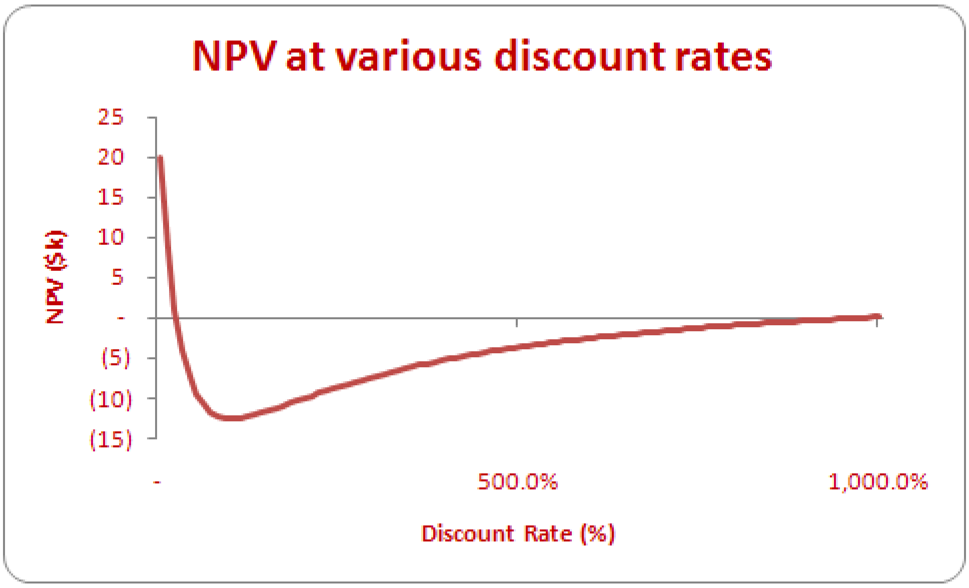
The **XIRR** function uses the same three cashflow scenarios and calculates correctly for the first and third scenarios. They demonstrate what happens when cashflows are aperiodic. For example, in the first scenario, instead of an annual IRR of 21.54% (as calculated above) we get 2.77%, reflecting the irregular payment profile.

The second scenario should be different now too, as the timing of the cashflows is different. The rate cited is 0.00%. This means that with no discounting the NPV should be zero, *i.e.* the numbers should add up to zero. They do not: the sum is actually \$2,325.05 and there is a clue in the number formatting as to what is going on here.

Every time a cashflow series changes sign (*i.e.* goes from positive to negative or vice versa) there is potentially another solution to the Internal Rate of Return (IRR). Consider the following:



Dates	
1/1/13	\$5,000
1/1/14	(\$55,000)
1/1/15	\$10,000
1/1/16	\$60,000
XIRR	970.86%
XNPV using XIRR	\$0
IRR	21.43%
XNPV using IRR	\$0



Here, prompted by a guess in the **XIRR** function (albeit of the other solution 21.43%), the two common Excel functions **XIRR** and **IRR** return the two IRRs associated with this cashflow scenario. It is important to not only check that an IRR gives an NPV of zero but that it is also the correct one in the circumstances.

This is the first problem with the concept of IRR. However, before we look at an objective way to generate just one meaningful solution for analysis, I'd like to consider another key issue. Forget the almost nonsensical IRR of 970.86% quoted in the above example. The other solution, 21.43%, seems more "realistic", yes?

As explained in the previous section, the Internal Rate of Return (IRR) is the name given to the discount rate that makes the Net Present Value (NPV) of a range of cashflows zero. For example, if I invest $100 now and receive $121 back in two years' time this would give me an annual IRR of 10% since:

$$\begin{aligned} (\$100) + PV(\$121) \quad &= (\$100) + \$121 / (1 + 10\%)^2 \\ &= (\$100) + \$100 \\ NPV \quad &= 0 \end{aligned}$$

It is nothing more than this. Put simply, if all cash required is borrowed at 21.43% and all surplus cash is reinvested at 21.43%, my project would neither create nor destroy cash value.

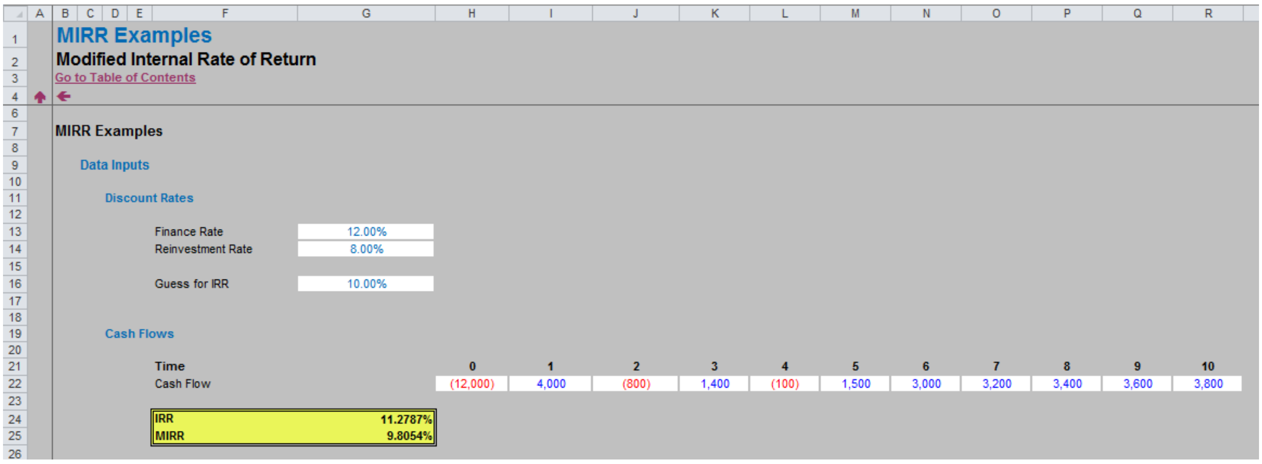
Wait a minute. Reinvest at 21.43%..? If I could find a risk-free investment returning this sort of money I would be depositing my pension in it never mind stakeholders' funds. The symmetry of the finance rate (cost of borrowing, typically the Weighted Average Cost of Capital) and the reinvestment rate (the return surplus funds can generate) is usually an absurd notion in the real world: if gains could be made in a free market, the principle of arbitrage would soon erode this advantage.

If we are looking for a measure to address the multiple solutions issue of IRR, perhaps we should also ensure it considers the fact that finance rates tend to be greater than reinvestment rates.

This is where we now use the

Modified Internal Rate of Return (MIRR). To explain how this works, I will be using the following example, which is included in the associated Excel file.

Consider the following assumptions:



MIRR Examples

Modified Internal Rate of Return

Go to Table of Contents

MIRR Examples

Data Inputs

Discount Rates

Finance Rate	12.00%
Reinvestment Rate	8.00%
Guess for IRR	10.00%

Cash Flows

Time	0	1	2	3	4	5	6	7	8	9	10
Cash Flow	(12,000)	4,000	(800)	1,400	(100)	1,500	3,000	3,200	3,400	3,600	3,800

IRR	11.2787%
MIRR	9.8054%

Let's keep this example nice and simple. Here, I have assumed a finance rate of 12%, a reinvestment rate of a more realistic 8% (say) and cash flows generated periodically at 11 points of time (Time 0 being "now" to Time 10 being 10 periods from "now"). Notice that the cash flows change sign a total of five times, which means there could be potentially five different IRRs.

This is the reason for the "Guess" cell (**G16**) in the illustration above. The IRR formula in cell **G24** is

=IRR(H22:R22,Guess)

where changing the value of **Guess** may cause the IRR calculated to vary (*i.e.* generate an alternative solution).

The **MIRR** calculation (cell **G25**) is simply

=MIRR(H22:R22,Finance_Rate,Reinvestment_Rate)

where the **Finance_Rate** is entered in cell **G13** and the **Reinvestment_Rate** is entered in cell **G14**. The formula for **MIRR** is defined as follows:

$$\left(\frac{-\mathrm{NPV}(rrate, values[positive]) * (1 + rrate)^{n}}{\mathrm{NPV}(frate, values[negative]) * (1 + frate)}\right)^{\frac{1}{n-1}}$$

where:

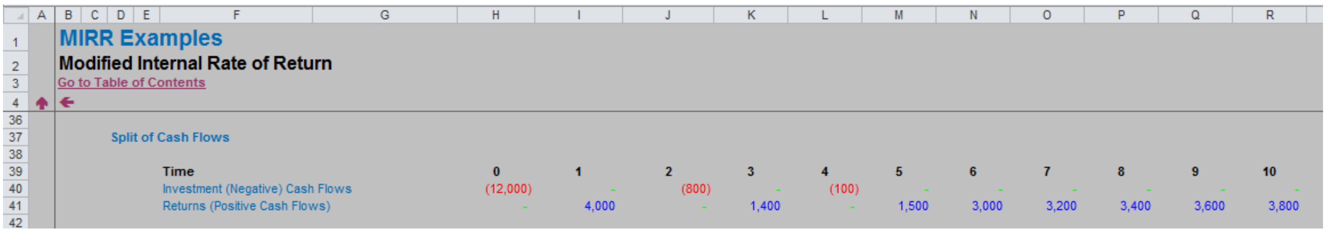
- **NPV()** is the Excel NPV function
- **rrate** is the reinvestment rate
- **frate** is the finance rate
- **values[positive]** is the positive values in the array only
- **values[negative]** is the negative values in the array only
- **n** is the number of periods.

This formula will always give the same value regardless of the number of changes of sign in the cash flow. It also takes into account the disparity between reinvestment and finance rates. It ticks the boxes – the only question is: what on earth does it do?

It's quite simple actually in concept. Let's ignore the formula and perform the calculation manually with the example above.

The first problem we have is the number of sign changes (five). To get an objective measure, we need just the one change of sign to ensure a unique solution. How do we do that?

To begin with, the cash flows should be split in two as follows:



MIRR Examples

Modified Internal Rate of Return

Go to Table of Contents

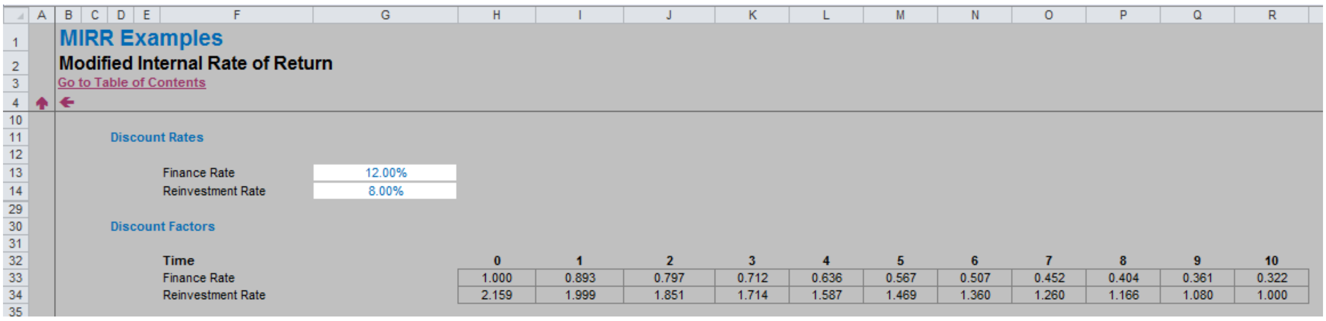
Split of Cash Flows

Time	0	1	2	3	4	5	6	7	8	9	10
Investment (Negative) Cash Flows	(12,000)	-	(800)	-	(100)	-	-	-	-	-	-
Returns (Positive Cash Flows)	-	4,000	-	1,400	-	1,500	3,000	3,200	3,400	3,600	3,800

The associated Excel file clearly shows how this breakdown was arrived at using **MAX()** and **MIN()** functions.

The intention now is to replace the values in one or both rows by an equivalent single positive or negative number. This is not simply the summation of the rows as this does not take into account the time value of money (*e.g.* \$100 reinvested would be worth \$108 = \$100 x (1 + 8%) in the next period using the above assumptions).

To work out what the discount factor should be, we need to determine the appropriate rate (finance rate for negative cash flows and reinvestment rate for positive cash flows) and at what point in time the cash flows are to be collated. We do this using the following table:



MIRR Examples

Modified Internal Rate of Return

Go to Table of Contents

Discount Rates

Finance Rate	12.00%
Reinvestment Rate	8.00%

Discount Factors

Time	0	1	2	3	4	5	6	7	8	9	10
Finance Rate	1.000	0.893	0.797	0.712	0.636	0.567	0.507	0.452	0.404	0.361	0.322
Reinvestment Rate	2.159	1.999	1.851	1.714	1.587	1.469	1.360	1.260	1.166	1.080	1.000

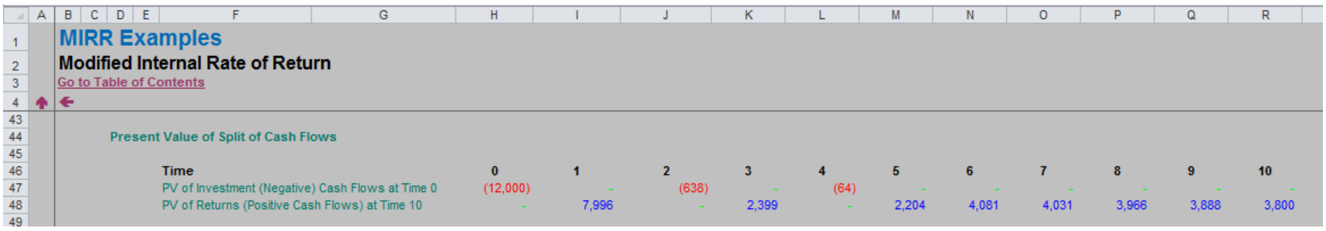
The Finance Rate calculates the appropriate discount factor required to generate the present values for Time 0 (*i.e.* the value of all negative cash flows as if they had arisen in the first period). This is because for most projects, companies will invest first (negative cash flows in early periods) to receive positive cash flows in later periods.

For example, the Time 2 factor (0.797, cell J33) is calculated as $1/1.12^2$, *i.e.* discounting for two periods at the finance rate of 12%.

The Reinvestment Rate calculates the appropriate discount factor required to generate the present values for the final period (here, Time 10 or the 11th period) (*i.e.* the value of all positive cash flows as if they had arisen in the last period). As before, this is because for most projects, companies will invest first (negative cash flows in early periods) to receive positive cash flows in later periods – including the final period.

For example, the Time 7 factor (1.260, cell O34) is calculated as 1.08^3, *i.e.* inflating for three periods (= 10 – 7) at the reinvestment rate of 8%.

Now, we simply cross-multiply the discount factors (rows 33 and 34 in our example) by the split cash flows (rows 40 and 41), *viz*.



MIRR Examples
Modified Internal Rate of Return
Go to Table of Contents

Present Value of Split of Cash Flows

Time	0	1	2	3	4	5	6	7	8	9	10
PV of Investment (Negative) Cash Flows at Time 0	(12,000)	-	(638)	-	(64)	-	-	-	-	-	-
PV of Returns (Positive Cash Flows) at Time 10	-	7,996	-	2,399	-	2,204	4,081	4,031	3,966	3,888	3,800

The negative numbers after Time 0 become smaller (reflecting the discounting), whereas positive cash flows are increasingly inflated the earlier they are to the final period (Time 10).

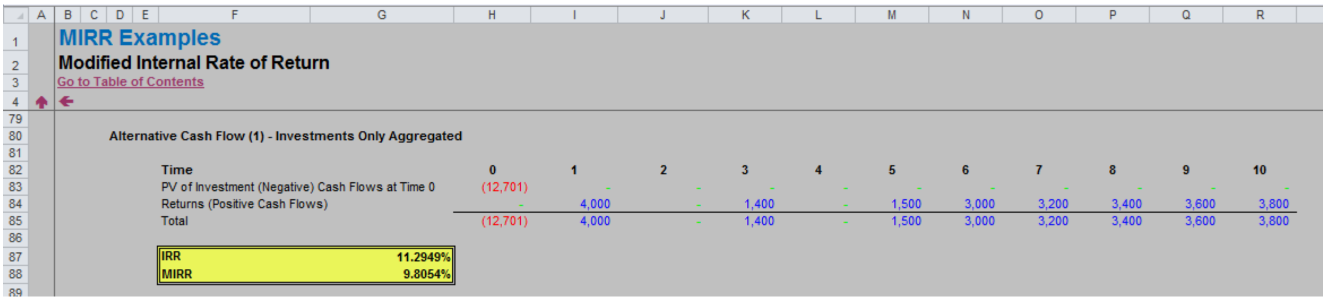
We now have three alternative cash flows we can consider:

1. aggregate the investment (negative) cash flows only;
2. aggregate the returns (positive cash flows) only;
3. aggregate both the investment cash flows and the returns.

Each of these options will only create one change of sign and consider the disparate discount rates. I now consider each one in turn.

1. Aggregate the Investment (Negative) Cash Flows Only

The associated Excel file calculates the following cash flow:



MIRR Examples
Modified Internal Rate of Return
Go to Table of Contents

Alternative Cash Flow (1) - Investments Only Aggregated

Time	0	1	2	3	4	5	6	7	8	9	10
PV of Investment (Negative) Cash Flows at Time 0	(12,701)	-	-	-	-	-	-	-	-	-	-
Returns (Positive Cash Flows)	-	4,000	-	1,400	-	1,500	3,000	3,200	3,400	3,600	3,800
Total	(12,701)	4,000	-	1,400	-	1,500	3,000	3,200	3,400	3,600	3,800

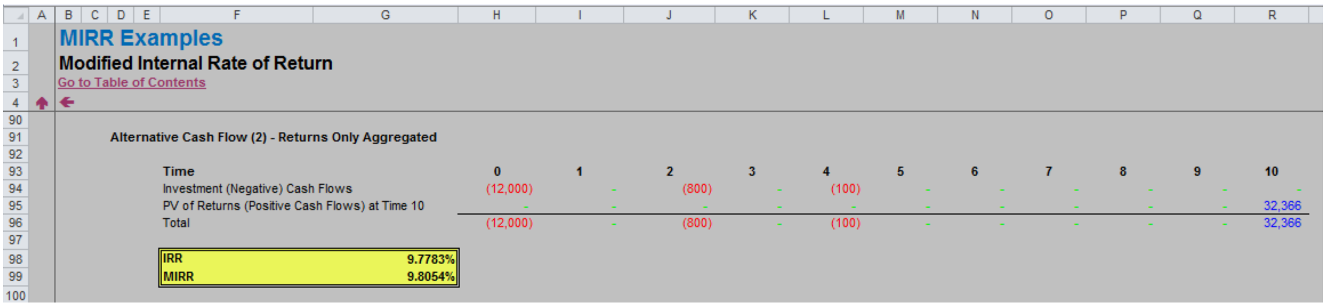
IRR	11.2949%
MIRR	9.8054%

Row 85 shows a negative cash flow in the first period (being the sum of row 47) with non-negative cash flows thereafter (from row 41). Having zero in a period does not constitute a change of sign, but these cells must be zero rather than blank else the Excel functions will not calculate correctly.

Note that whilst the IRR changes slightly from the original calculation, the MIRR is precisely the same. This IRR is unique (only one change of sign).

2. Aggregate the Returns (Positive Cash Flows) Only

The following cash flow may be calculated:



MIRR Examples
Modified Internal Rate of Return
Go to Table of Contents

Alternative Cash Flow (2) - Returns Only Aggregated

Time	0	1	2	3	4	5	6	7	8	9	10
Investment (Negative) Cash Flows	(12,000)	-	(800)	-	(100)	-	-	-	-	-	-
PV of Returns (Positive Cash Flows) at Time 10	-	-	-	-	-	-	-	-	-	-	32,366
Total	(12,000)	-	(800)	-	(100)	-	-	-	-	-	32,366

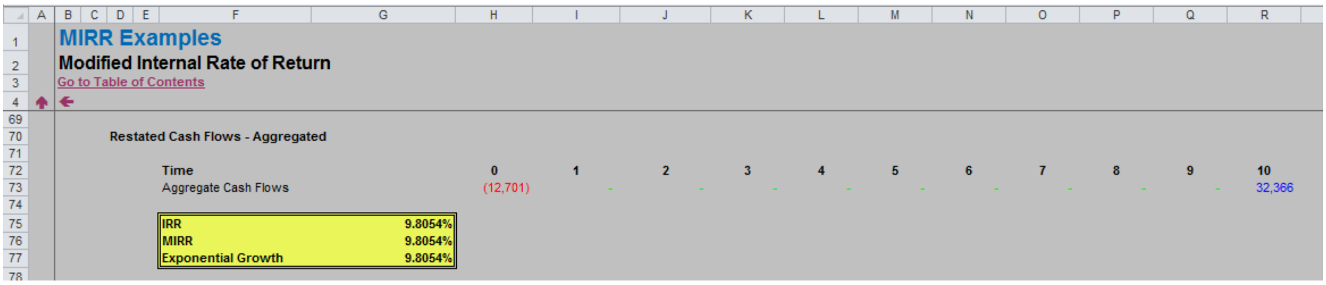
IRR	9.7783%
MIRR	9.8054%

Row 96 shows several negative cash flows (referenced to row 40) with a non-negative cash flow in the final period (being the sum of row 48).

Note that again the IRR changes from the original calculation (it is reduced since all positive cash flows have been moved to the final period), the MIRR is precisely the same. As before, this IRR is unique (only one change of sign).

3. Aggregate Both the Investments and the Returns (MIRR Approach)

This is the first cash flow shown in the Outputs section of the associated Excel file:



MIRR Examples

Modified Internal Rate of Return

Go to Table of Contents

Restated Cash Flows - Aggregated

	0	1	2	3	4	5	6	7	8	9	10
Time	0	1	2	3	4	5	6	7	8	9	10
Aggregate Cash Flows	(12,701)	-	-	-	-	-	-	-	-	-	32,366

IRR	9.8054%
MIRR	9.8054%
Exponential Growth	9.8054%

Row 73 contains only two non-zero flows: the present value of all investments at Time 0 and the future value of all returns At Time 10. As before, for this to work correctly, the interim period cash flows must be zero rather than blank.

As above, the IRR will be unique, but this time the IRR equals the MIRR. **This is how the MIRR is calculated**. Indeed, cell **G77** contains an alternative method of calculation, the "Exponential Growth" approach, calculated as:

$$=(32{,}366/12{,}701)^{1/10}$$

This is essentially the **MIRR** formula from earlier:

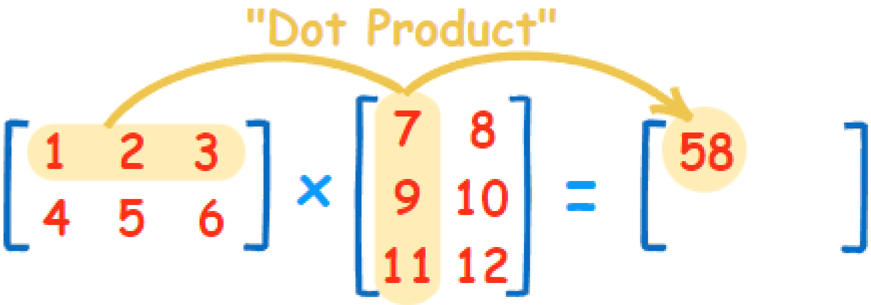
$$\left(\frac{-\mathrm{NPV}(rrate, values[positive]) * (1+rrate)^{n}}{\mathrm{NPV}(frate, values[negative]) * (1+frate)}\right)^{\frac{1}{n-1}}$$

Using the formula, MIRR is arguably quicker to calculate than IRR, more objective (only one solution) and takes into account the differing rates implicit in the cash flows. **MIRR** is usually lower than IRR (assuming the reinvestment rate will be lower than the finance rate), unless the reinvestment rate equals the finance rate, whereby altering the cash flows as depicted above will neither affect the NPV nor the IRR.

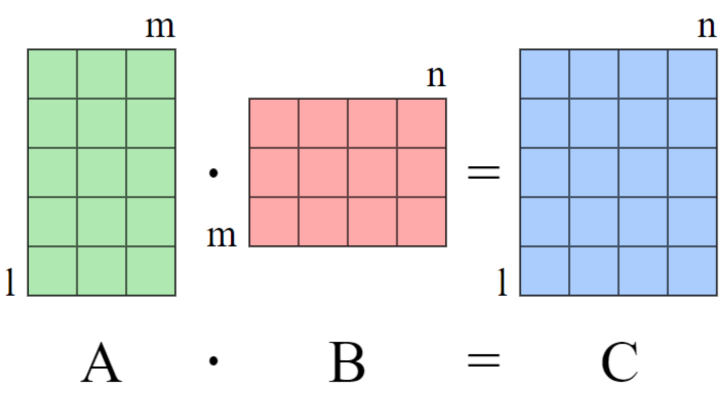
The MIRR is often seen as a financial measure of an investment's attractiveness. It is used frequently in capital budgeting to rank alternative investments of similar size (although this may not always be an appropriate approach: NPV or NPV per $ invested (the so-called "bang for buck" key factor analysis) may be more suitable metrics).

There is much confusion about what the reinvestment rate implies. However, both the NPV and the IRR techniques assume the cash flows generated by a project are reinvested within that same project. This is not always the case; often, they are often reinvested elsewhere within the business and it is not a necessary assumption that the firm is capable of generating that IRR elsewhere. Indeed, one implication of the MIRR is that the project is not capable of generating cash flows as predicted and that the project's NPV is overstated.

The A to Z of Excel Functions: MMULT



In mathematics, matrix multiplication is what's known as a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the matrix product, has the number of rows of the first and the number of columns of the second matrix, *i.e.*



Algebraically, if **A** is an **m** x **n** matrix and **B** is an **n** x **p** matrix, w

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

then the matrix product **C** = **AB** is determined to be the **m** x **p** matrix

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

i.e.

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

The Excel function **MMULT** returns the matrix product of two arrays, **array1** and **array2** (the Excel equivalent of a matrix). The result is an array with the same number of rows as **array1** and the same number of columns as **array2**.

It has the following syntax:

MMULT(array1, array2)

where:

- **array1** and **array2** are required, and represent the two arrays you wish to multiply.

It should be noted that:

- as explained above, the number of columns in **array1** must be the same as the number of rows in **array2**, and both arrays must contain only numbers
- **array1** and **array2** may be given as cell ranges, array constants or references
- **MMULT** returns the *#VALUE!* error when:
  - any cells are empty or contain text
  - the number of columns in **array1** is different from the number of rows in **array2**
- The matrix product array, **a**, of two arrays, **b** and **c**, is:

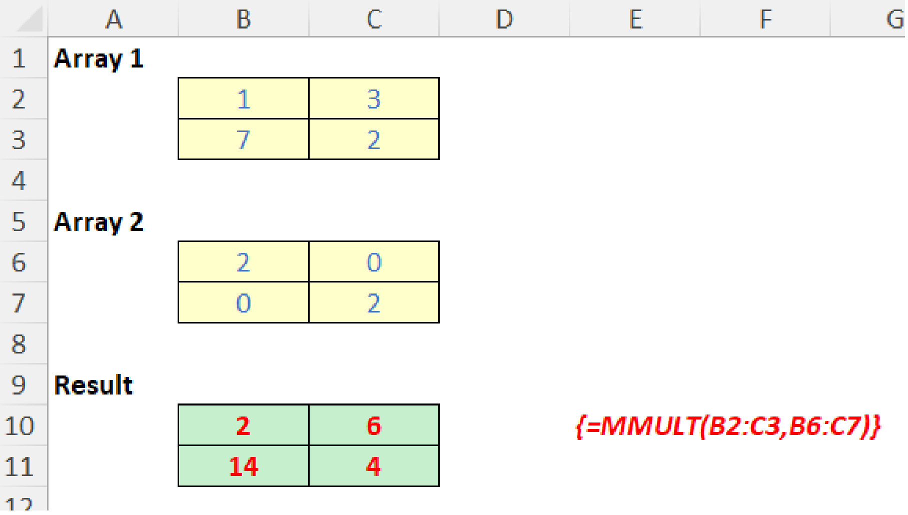
$$a_{ij} = \sum_{k=1}^{n} b_{ik} c_{kj}$$

where **i** is the row number and **j** is the column number.

If you are using Microsoft 365, then you can simply enter the formula in the top left cell of the output range, then press **ENTER** to confirm the formula as a dynamic array formula. Otherwise, the formula must be entered as a legacy array formula by first selecting the output range, entering the formula in the top-left-cell of the output range, and then pressing **CTRL+SHIFT+ENTER** to confirm it. Excel inserts braces (curly brackets) automatically.

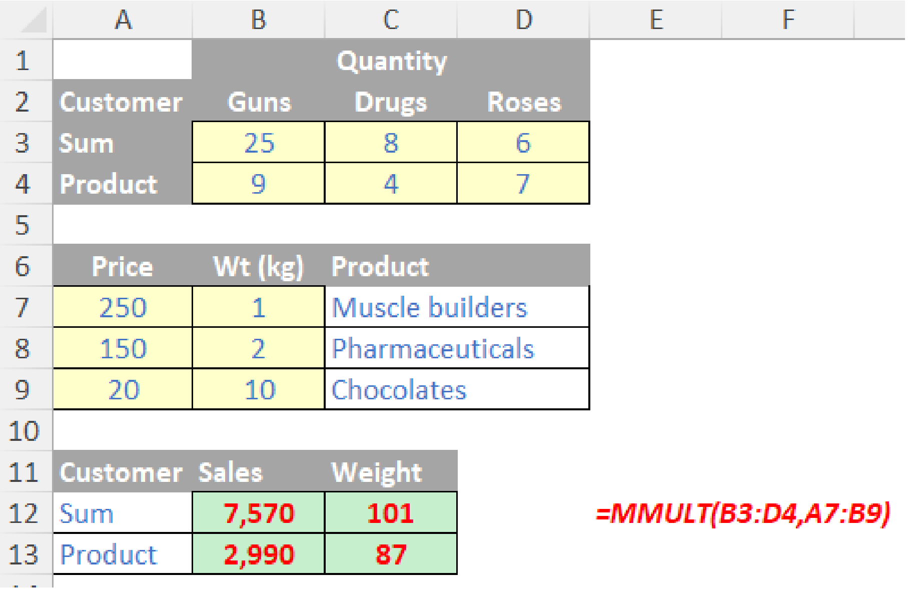
As examples:

Legacy Approach



	A	B	C	D	E	F	G
1	**Array 1**						
2		1	3				
3		7	2				
4							
5	**Array 2**						
6		2	0				
7		0	2				
8							
9	**Result**						
10		**2**	**6**		*{=MMULT(B2:C3,B6:C7)}*		
11		**14**	**4**				
12							

Using Dynamic Arrays



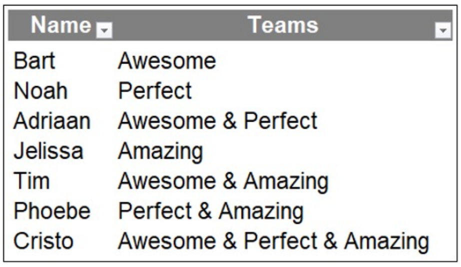
	A	B	C	D	E	F
1		Quantity				
2	Customer	Guns	Drugs	Roses		
3	Sum	25	8	6		
4	Product	9	4	7		
5						
6	Price	Wt (kg)	Product			
7	250	1	Muscle builders			
8	150	2	Pharmaceuticals			
9	20	10	Chocolates			
10						
11	Customer	Sales	Weight			
12	Sum	**7,570**	**101**		*=MMULT(B3:D4,A7:B9)*	
13	Product	**2,990**	**87**			

More Excel Functions next month.

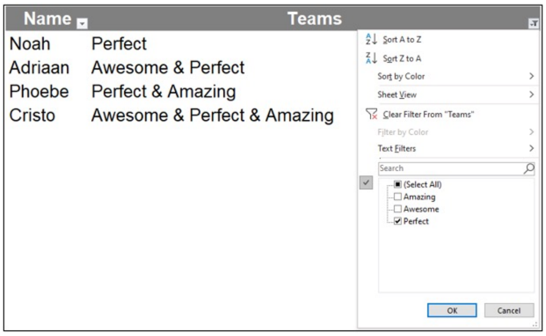
Beat the Boredom Suggested Solution

This month, we asked you to imagine that you needed to filter rows in a table that have specific keywords within the text strings contained therein. Manually filtering every single keyword and copying / pasting them to a new location would be a tedious and time-consuming process. To address this, we challenged you to develop a user-friendly solution that allows users to select the desired keywords and return a list having all the text strings associated with those keywords.

Your aim was to create a filter using the keywords "Awesome", "Amazing", and "Perfect" as filter criteria, as shown in the picture below:



Name	Teams
Bart	Awesome
Noah	Perfect
Adriaan	Awesome & Perfect
Jelissa	Amazing
Tim	Awesome & Amazing
Phoebe	Perfect & Amazing
Cristo	Awesome & Perfect & Amazing



Name	Teams
Noah	Perfect
Adriaan	Awesome & Perfect
Phoebe	Perfect & Amazing
Cristo	Awesome & Perfect & Amazing

As always, there were some requirements:

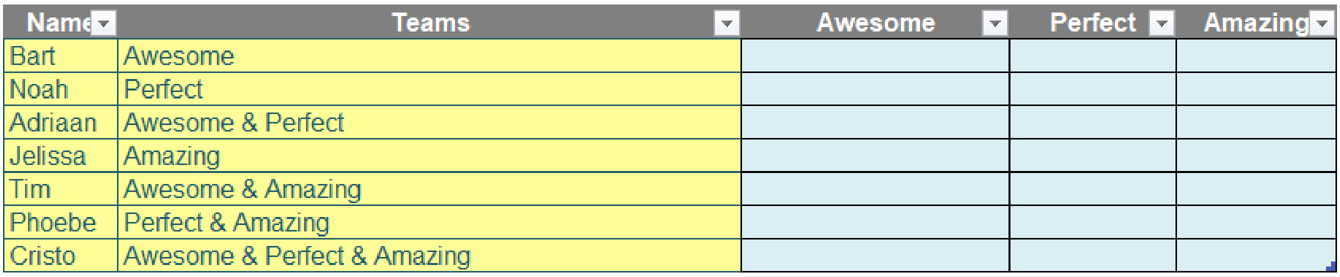
- no Power Query / Get & Transform or VBA was allowed
- the formula(e) should be dynamic so that they should update when a new entry was added.

Suggested Solution

The steps are detailed below.

Find the keywords

To begin, we create three [3] new columns in the **Data** Table (**CTRL + T**) with the heading names based upon the keywords provided:



Name	Teams	Awesome	Perfect	Amazing
Bart	Awesome			
Noah	Perfect			
Adriaan	Awesome & Perfect			
Jelissa	Amazing			
Tim	Awesome & Amazing			
Phoebe	Perfect & Amazing			
Cristo	Awesome & Perfect & Amazing			

We accept this is a manual interaction as Table headers may not contain formulae.

To check whether the text strings in the **Data[Teams]** column have specific keywords, we need to create a formula that uses the **ISNUMBER** and **SEARCH** functions. These functions work together to confirm the keywords are within the text strings (note that the **FIND** function could also be used, but beware that it's case-sensitive and requires an exact match between the capitalisation of the keywords and text strings).

=ISNUMBER(SEARCH(Data[[#Headers],[Awesome]],[@Teams]))*1

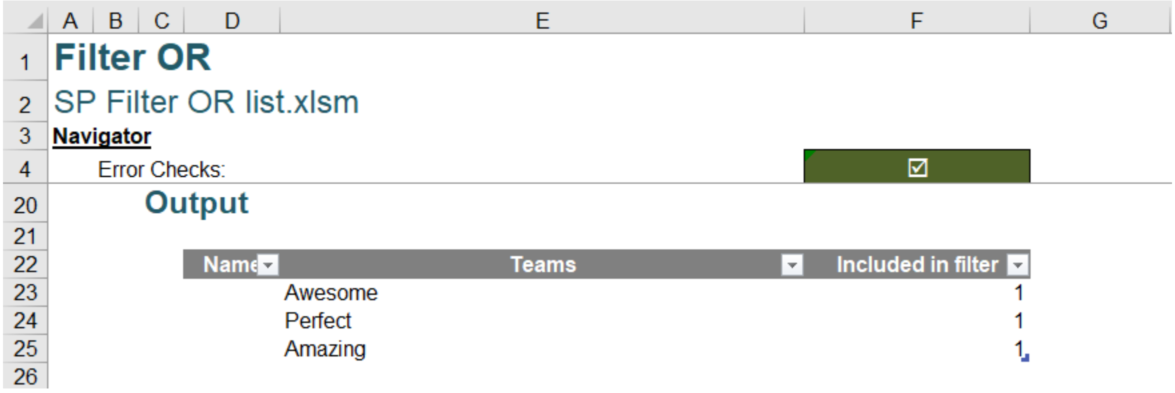
The **SEARCH** function will return a number if the keywords is found within the text string. If the text string did not contain the keyword, then it will return *#VALUE!*. Then the **ISNUMBER** checks whether the output of the **SEARCH** function is a number or not and will return TRUE or FALSE accordingly. At this point, we can choose to keep it as logical value or turn it into number. We will change all these logical values into numbers, so we multiply the numbers by one [1]. Therefore, all TRUE values will be restated as one [1] and all FALSE values will turn to zero [0]. Similarly, we will do the same for the other two [2] columns, *viz.*

=ISNUMBER(SEARCH(Data[[#Headers],[Perfect]],[@Teams]))*1

=ISNUMBER(SEARCH(Data[[#Headers],[Amazing]],[@Teams]))*1

Hidden power of SUBTOTAL

Let's move on the **Output** table. In this table, we first enter the keywords of this challenged in the **Data[Teams]** column and we will leave the **Data[Name]** column blank:



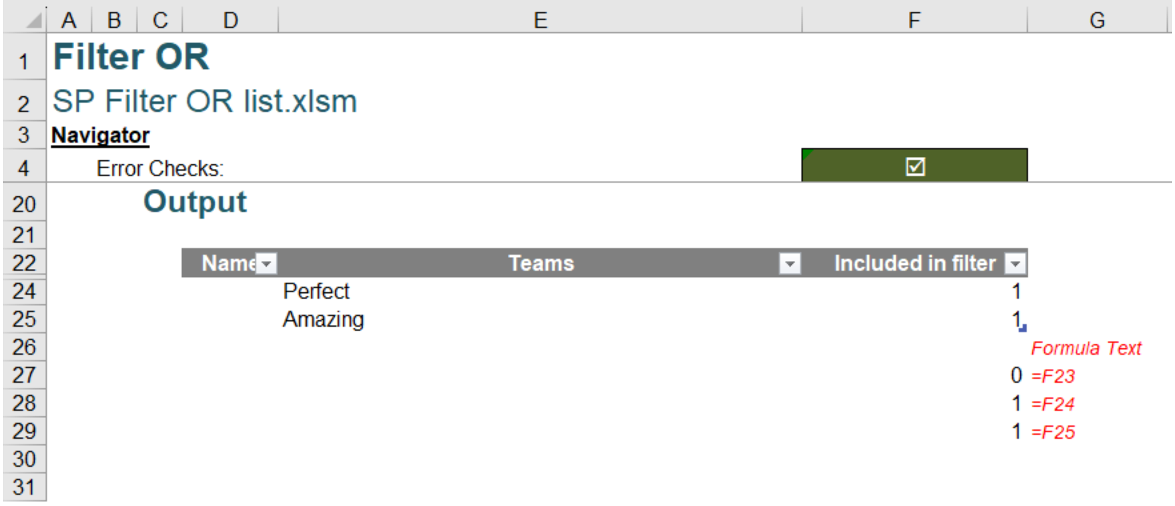
	A	B	C	D	E	F	G
1	Filter OR						
2	SP Filter OR list.xlsm						
3	Navigator						
4		Error Checks:				☑	
20			Output				
21							
22				Name	Teams	Included in filter	
23					Awesome	1	
24					Perfect	1	
25					Amazing	1	
26							

Then, we added one column to this Table and we call this column 'Included in filter'. This column will specify which keywords to include in the filter. The formula for this new column will be:

=SUBTOTAL(103,[@Teams])

The **SUBTOTAL** function has the advantage of being able to exclude hidden cells from calculations. When the **function_num** is greater than 100, the **SUBTOTAL** function will not include any hidden cells in the calculation.

Using **function_num** 103 of **SUBTOTAL**, which is **COUNTA**, will return one [1] if a cell is not hidden and zero [0] if the cell is hidden. For instance, if we completely hide the row 23, the result will be as follows:



	A	B	C	D	E	F	G
1	Filter OR						
2	SP Filter OR list.xlsm						
3	Navigator						
4		Error Checks:				☑	
20			Output				
21							
22				Name	Teams	Included in filter	
24					Perfect	1	
25					Amazing	1	
26							Formula Text
27						0	=F23
28						1	=F24
29						1	=F25
30							
31							

As we can see that cell **F23** value is zero [0] while another unhidden cell is one [1].

Matching keywords

We will now add another column to the **Data** Table called 'Included'. This column will determine whether to include a particular row in our **Output** table based upon a formula. The formula is as follows:

=MIN(MMULT(Data[@[Awesome]:[Amazing]],Output[Included in filter]),1)

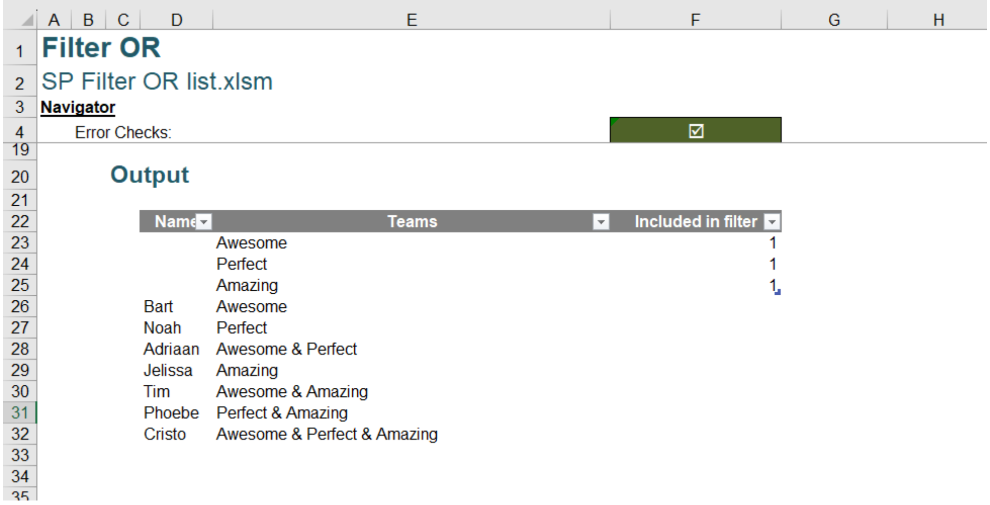
The first part of the formula, **MMULT(Data[@[Awesome]:[Amazing]],Output[Included in filter])**, multiplies two [2] selected vectors to obtain the dot product. You can read more about **MMULT** in our *A to Z of Excel Functions* (earlier in this newsletter). The resulting dot product shows how many matches the text strings have with the filter. We then wrap this dot product in the **MIN** function to ensure that our output is either one [1] or zero [0], like a TRUE or FALSE value.

Filter

Below the Output table we will use **FILTER** function to filter out the list that contains the keywords:

=FILTER(Data[[Name]:[Teams]],Data[Included])

This will use the Included column in **Data** table to filter out all the match we have on the **Data** table and output it out in form of an array. Therefore, we have the following output table:



Filter OR

SP Filter OR list.xlsm

Navigator

Error Checks: ☑

Output

Name	Teams	Included in filter
	Awesome	1
	Perfect	1
	Amazing	1
Bart	Awesome	
Noah	Perfect	
Adriaan	Awesome & Perfect	
Jelissa	Amazing	
Tim	Awesome & Amazing	
Phoebe	Perfect & Amazing	
Cristo	Awesome & Perfect & Amazing	

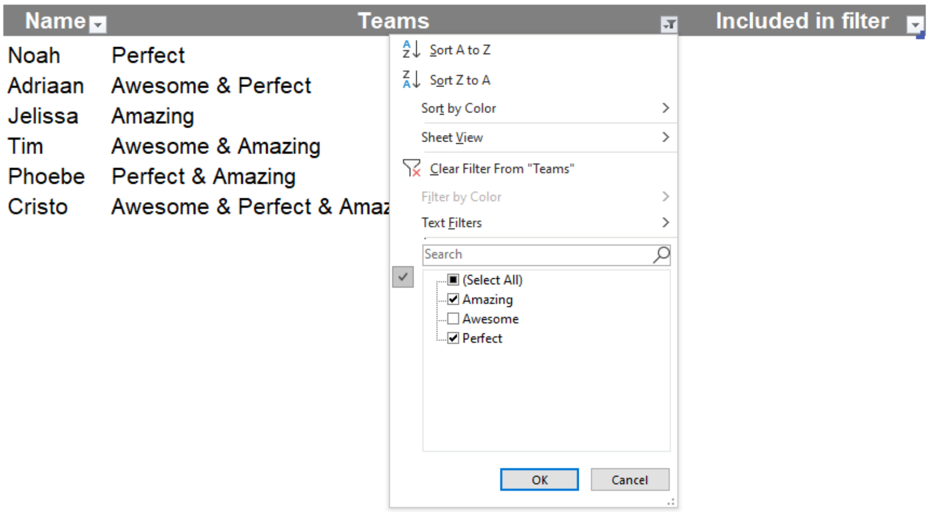
It is important to note that anything typed below a Table will automatically add a new row to the Table. Therefore, you may need to resize the **Output** table to exclude the new formula that was just added.

Hide, but not completely

The project is currently 80% complete, but we need to hide rows 23 to 25 from our output. However, we need to be careful when hiding these rows. We do not want to completely hide them, as this could cause the **SUBTOTAL** to become zero [0] or the filter to stop working.

The lowest pixel height for a row that will still allow the **SUBTOTAL** function to work is 1/3 pixel (0.25 points). To achieve this, we can select the rows we want to hide and go to **Home -> Cells -> Format -> Row Height** (or use the shortcut key **Alt + H + O + H**) and enter 0.25 points in the box. However, the filter of Excel will not work if cell is under one [1] pixel (0.75 points). Therefore, we need to adjust the row height to one [1] pixel to ensure that our filter works properly.

After setting the row height to one [1] pixel, our filter is now complete:



Name	Teams	Included in filter
Noah	Perfect	
Adriaan	Awesome & Perfect	
Jelissa	Amazing	
Tim	Awesome & Amazing	
Phoebe	Perfect & Amazing	
Cristo	Awesome & Perfect & Amaz	

Now, we may use the drop-down menu from the Table to filter out **Data[Teams]** that contains the keywords we need. The drop-down filter will automatically set the row height to zero [0] pixels for unselected criteria and maintain one [1] pixel for selected criteria.

Word to the Wise

We appreciate there are many, many ways this could have been achieved. If you have come up with an alternative, radically different approach, congratulations – that's half the fun of Excel!

Until next month.

Upcoming SumProduct Training Courses

Location	Course	Date	Date	Duration	Duration
Melbourne Australia	Excel Tips and Tricks	5 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Melbourne Australia	Financial Modelling	6 - 7 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days
Melbourne Australia	Power Pivot, Power Query and Power BI	8 - 9 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days
Sydney Australia	Power Pivot, Power Query and Power BI	19 - 20 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days
Sydney Australia	Excel Tips and Tricks	21 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Sydney Australia	Financial Modelling	22 - 23 Feb 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days
Melbourne Australia	Power Pivot, Power Query and Power BI	18 - 19 Mar 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days
Melbourne Australia	Excel Tips and Tricks	20 Mar 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Melbourne Australia	Financial Modelling	21 - 22 Mar 2024	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. We've started going through the alphabet actions. This month, it's "S":

Keystroke	What it does
ALT + SPACE	Displays control menu
CTRL + SPACE	Select column
SHIFT + SPACE	Select row
CTRL + SHIFT + SPACE	Select contiguous area
Subtract (-)	Extend Selection mode
CTRL + Subtract (-)	Delete cells
SHIFT + Subtract (-)	Add to Selection mode

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

Our Services

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

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

If you require modelling assistance of any kind, please do not hesitate to contact us at contact@sumproduct.com.

Link to Others

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

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

Any Questions?

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

Training

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

Check out our more popular courses in our training brochure:

Drop us a line at training@sumproduct.com for a copy of the brochure or download it directly from www.sumproduct.com/training.

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

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