Sum Product

NEWSLETTER #89 - April 2020

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

It may be dark times around the world

but we will try to light up your Excel, Power BI and modelling realms here as we seek to make analytical sense of the world around us.

All the usual stuff is here. You know the drill by now. For as long as we can, it's Business As Usual. Stay safe, stay happy, stay healthy.

Until next month.

Liam Bastick, Managing Director, SumProduct

Coronavirus: We Are Here To Help

There is only one way we can start this month's newsletter off. As COVID-19 continues to make an impact around the world, we hope all readers, their loved ones, their colleagues and their friends, all stay safe and healthy, as the pandemic takes an unrelenting, odious grip of the planet.

Little else matters.

We are not medics. We will continue to wash our hands and some of us may wear masks, but we will not wash our hands of our clients or mask their needs. We appreciate there is more to life than business, and that everyone needs to do their bit at this juncture.

This is NOT a sales pitch. SumProduct is committed to helping where we can:

- **Physical contactless consulting and assistance:** we have been working remotely with clients all over the world for many years. We can provide modelling and strategic services without meeting face to face, using Skype, Teams, Zoom, WebEx, GoToMeeting and other similar software products. The world is having to come to terms with social distancing, but we can still cater for all your needs using real-time, internet connectivity, wherever you may be in the world
- **Simple cashflow modelling:** the world is changing. Businesses are seeing a drop-off in custom and short-term, forecasting models (such as 13-week, rolling, daily cashflow forecasts) are becoming an urgent requirement for all businesses, large or small. We can help build simple bespoke Excel / Power BI forecasting tools which can analyse the impacts on your business and help you manage your cash flow before it manages you – simple, cheap solutions <u>are</u> possible
- **On-line training, virtual training and webinars:** as more and more staff are having to work in isolation, from home or elsewhere, it is becoming increasingly important to ensure that staff have the requisite skills to do their work with less physical supervision. SumProduct has a wealth of on-line, interactive courses that cater for the individual, small / medium enterprises and large corporates.

  Topics include (but are not limited to):

  - charts and dashboards
  - critical skills for decision-making
  - Excel tricks and tips
  - financial modelling
  - PivotTables
  - Power BI
  - Power Pivot
  - Power Query
  - valuations modelling
  - VBA.

  We also offer virtual training courses, where staff can attend our usual physical training courses from their PC with full trainer support and webinars which provide live and pre-recorded presentations. We can even design bespoke sessions for you and your colleagues

Microsoft® Most Valuable Professional

- **Discounted products and services:** we know cashflow is king and businesses are struggling to survive. We are cutting our prices on all training, consulting, modelling and auditing work. If you need something, let us know. We are unable to provide much for free, but we endeavour to assist as much as we can on a heavily discounted, value for money basis, so all businesses – ourselves included – can get through these dark times and come out the other side more informed, stronger and in a better position to take advantage of the possibilities that will inevitably arise
- **Liam will actually commit to tell a funny joke at some point in 2020:** OK, maybe we've stretched credibility too far here. That one we probably can't deliver upon, but the rest we certainly can.

If you want to find out more, drop us a line at contact@sumproduct.com. Yes, we are a business, but we are here to help, not just make some quick buck. We have been trusted advisers and trainers for over 10 years at SumProduct, and for many more before that. Here's to another 10 years+ for all of us – from both a personal and a business perspective.

To LET

If you are the victim of slow-running spreadsheets, you must read this.

Mid-March, Microsoft announced a new Excel function **LET** for Office 365, albeit at the time of writing, only to the Insiders Channel. This allows you to stop writing the same expressions time and time again in a formula or allows portability of segments of a computation for different formulae. As Microsoft puts it, it's "…names on a formula level".

In essence, this function assigns values or expressions to defined names and then passes these to calculation results. It can store intermediate calculations, values or defining names inside a formula. The defined names only apply within the scope of the **LET** function in a similar way to variables in general programming scenarios.

If you reuse the same expression multiple times in a formula, Excel calculates that expression multiple times. That's *not* a good thing. **LET** allows you to name the expression and refer to it using that name, similar to **VAR** in a DAX expression, for those that know the language. Any named expression is calculated only once, even though it may be referred to many times in the formula. This can significantly improve performance considerably and speed up your spreadsheets.

To use the **LET** function, we must define pairs of names and associated values, and a calculation that uses them all, as the final argument. At least one name / value pair must be defined.

It has following syntax to operate:

LET(name1, value1, [name2…], [value2…], calculation)

where:

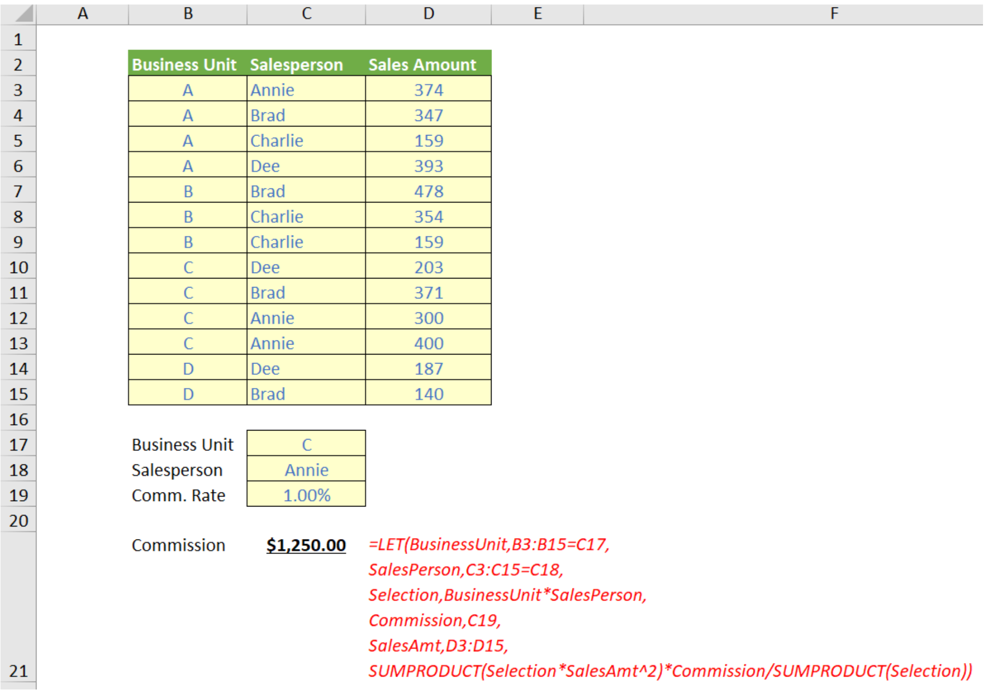
- **name1:** the name for the first value
- **value1:** the value to associate with the first name
- **name2** (optional): additional names for second and subsequent values
- **value2** (optional): additional values for the second and subsequent values
- **calculation:** this is the calculation to perform. This is always the final argument and it can refer to any of the defined names in the LET.

The main benefits of using LET function includes:

- **readability: LET** function tracks the changes in defined cells or expression in dynamic way, so there is no need to remember what a specific range / cell reference referred to. With the ability to define variables, it can provide more meaningful context to end users
- **performance: LET** allows named expression in the formula and calculated only once, even if it is referred to many times in the formula. This can significantly improve performance for complex expressions.

Let's look at an example. If I am considering **LET**, how about we use the property rental market! Suppose we run a company where salespeople make commission based upon 1% of the average of the square of the weekly rent, *e.g.* if Annie makes two sales of $400 and $300, then the average of the amounts squared would be ($300^2 + 400^2$) / 2 = 125,000 and 1% of that would be $1,250. Not much incentive to make lots of sales, is there? It's not my fault she negotiated her contract badly…

Anyway, it's convoluted, I know, but I am trying to construct an example which I cannot simply solve using a standard Excel formula like **AVERAGEIFS**!



Business Unit	Salesperson	Sales Amount
A	Annie	374
A	Brad	347
A	Charlie	159
A	Dee	393
B	Brad	478
B	Charlie	354
B	Charlie	159
C	Dee	203
C	Brad	371
C	Annie	300
C	Annie	400
D	Dee	187
D	Brad	140

Business Unit	C
Salesperson	Annie
Comm. Rate	1.00%

Commission **$1,250.00**

```
=LET(BusinessUnit,B3:B15=C17,
SalesPerson,C3:C15=C18,
Selection,BusinessUnit*SalesPerson,
Commission,C19,
SalesAmt,D3:D15,
SUMPRODUCT(Selection*SalesAmt^2)*Commission/SUMPRODUCT(Selection))
```

The formula in cell **C21** is given by

=LET(BusinessUnit, B3:B15=C17,
SalesPerson, C3:C15=C18,
Selection, BusinessUnit*SalesPerson,
Commission, C19,
SalesAmt, D3:D15,
SUMPRODUCT(Selection*SalesAmt^2)*Commission/SUMPRODUCT(Selection))

The final argument is the formula – but more on that shortly. Before that, there are pairs of expressions where the first element is the name and the second is an expression for a value. Let's go through all of these arguments (mainly in pairs):

- **BusinessUnit, B3:B15=C17** define the array **BusinessUnit**, which is an array of TRUE and FALSE values depending on whether the Business Unit is equal to the value in cell **C17** (which is "C" in the illustration). Therefore, we have {FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE, FALSE, FALSE}
- **SalesPerson, C3:C15=C18** generate a similar array (named **SalesPerson**), which provides TRUE and FALSE values depending on whether the salesperson was Annie. Therefore, we have {TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, FALSE}
- **Selection, BusinessUnit*SalesPerson** now shows the value of **LET**. This names the value of the product of the first two parameters **Selection** without having to write out the formula again and, more importantly, *without having to calculate them a second time unnecessarily*. This results in TRUE/FALSE expressions multiplied by other TRUE/FALSE expressions, which results in a one if both are TRUE and zero (0) otherwise. Therefore, we have {0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0}
- **Commission, C19** is relatively simple, naming the value in cell **C19** (1%) **Commission**
- Similarly, **SalesAmt, D3:D15** apply the name **SalesAmt** to the range **D3:D15** *for the purposes of this formula*. Therefore, we have {374, 347, 159, 393, 478, 354, 159, 203, 371, 300, 400, 187, 140}
- Finally, the formula **SUMPRODUCT(Selection*SalesAmt^2)*Commission/SUMPRODUCT(Selection)** calculates the commission similarly to the approach explained above. However, Selection, SalesAmt and **Commission** are merely referenced and neither need to be recalculated nor written out in full.

This may seem verbose, but it speeds up calculation time considerably, and makes formulae easier to read (once you get your head around this new approach). Further, **LET** only defines the range name *within* the formula. These are not general range names in the spreadsheet – Excel will not recognise them outside of this **LET** formula.

To be crystal clear, I see two key differences between defined range names and this new LET function:

1. **LET** allows you to see the definition of the name there and then without jumping to the definition elsewhere. Yes, the formula may be longer in the cell, but the user can hopefully follow it without audit tools
2. When Excel sees a defined name in a formula, it pauses the current formula, evaluates the defined name's formula to get a result, and returns to the original formula. A variable defined using **LET** is only ever evaluated once.

   An example that shows the difference is **=LET(x, RANDBETWEEN(1, 1000000), x – x)**. This formula will always return zero, since **RANDBETWEEN(1, 1000000)** gets evaluated and then assigned to x, which never changes, so **x – x** = 0.

   If you instead use the name manager to define the name **x** as **=RANDBETWEEN(1, 1000000)**, then entering **=x – x** is almost never going to be zero, since each **x** in the formula causes **RANDBETWEEN** to be evaluated.

It should also be noted that you must be careful with your order of arguments and defining reference names. Name definitions can only make use of prior and not subsequent names. In the example above, **Selection** could only be defined once **BusinessUnit** and **SalesAmt** had both been defined.

For those with Excel Insiders Fast, you really should have a play. Once you realise how this will be of benefit, there will be no stopping you!

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 the evergreen IF statement.

It's time to cover the use of the IF statement in VBA. This is a very useful tool to help apply conditional rules in macros. For example, if you want a macro to run A if a cell is positive, and B if it is negative, you can use the IF statement to check the cell value, and choose what will happen in the true and false cases.

The basic command looks like this:

```
If <X is true> Then
<Do something>
Else
<Do something else>
End If
```

You can also use it to run only if the check is true, as long as what you are doing will fit on one line, such as:

```
If <X is true> Then <Do something>
```

In this instance, if X is false, nothing will happen.

Let's look at a specific example. Suppose we want to check if the cell **A1** contains a value. If it does, let's clear the content of the cell. If not, we will copy the value from **B1** into **A1**. We could do this as follows:

```
If Range("A1").Value <> "" Then
Range("A1").ClearContents
Else
Range("B1").Copy
Range("A1").PasteSpecial xlPasteAll
End If
```

In this, the use of "<>" in the first line means "not equal to"; that is, that the value in cell **A1** is not equal to a blank space.

More next time!

Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we discuss a key function used for developing measures.

Any avid Excel user will know the sheer utility you get from the **SUMIF** function, unfortunately, **SUMIF** does not work with Power Pivot! Good thing for us there is an alternative – and it is actually *more* powerful.

The **CALCULATE** function evaluates an expression in a context that is modified by filters. Sounding similar to a famous Excel function namely **SUMIF**? **SUMIF** is not recognised in the Power Pivot programming language (Data Analysis eXpressions, DAX), hence we use the **CALCULATE** function instead.

Similar to **SUMIF**, the syntax for **CALCULATE** is:

CALCULATE (expression, filter1[, filter2,] …)

The filters can be located in any table imported to Power Pivot, as long as the tables have had proper relationships established. The really powerful thing? **CALCULATE** is not "context specific" (*i.e.* the data used does not need to pertain to the row or column headers of the PivotTable) and filters used do not need to be included in the PivotTable either!

Example

Let's create a measure that retrieves the sale for the year of 2014:

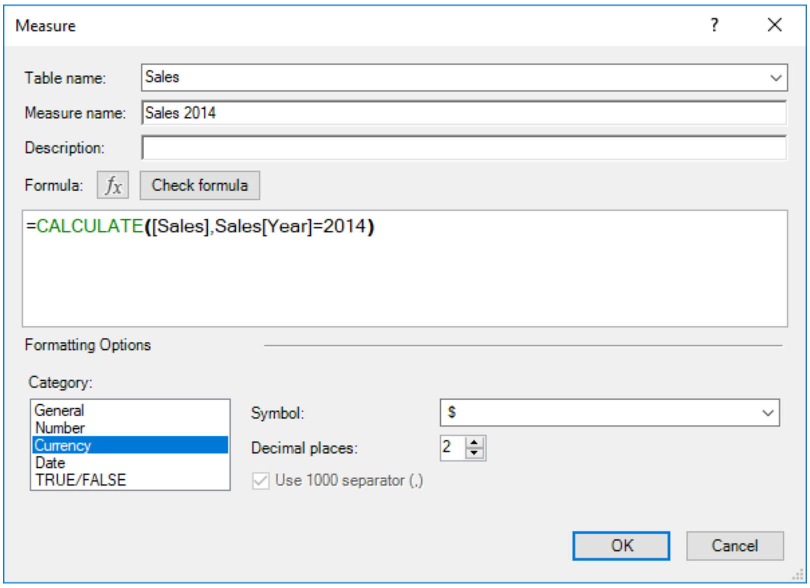


Table name: Sales

Measure name: Sales 2014

Description:

Formula: *fx*

```
=CALCULATE([Sales],Sales[Year]=2014)
```

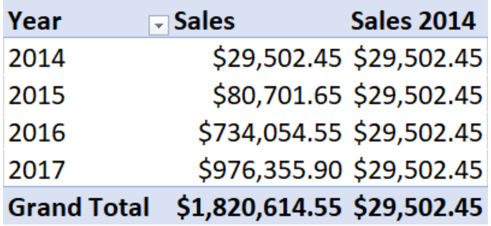
Category: General, Number, Currency (selected), Date, TRUE/FALSE

Symbol: $

Decimal places: 2

☑ Use 1000 separator (,)

We can check that our **CALCULATE** filter is working correctly by creating the following PivotTable:



Year	Sales	Sales 2014
2014	$29,502.45	$29,502.45
2015	$80,701.65	$29,502.45
2016	$734,054.55	$29,502.45
2017	$976,355.90	$29,502.45
Grand Total	**$1,820,614.55**	**$29,502.45**

As you can see, no matter what the year in each row of the PivotTable the **CALCULATE**-based measure always relates to 2014 data. We can also apply multiple filters:

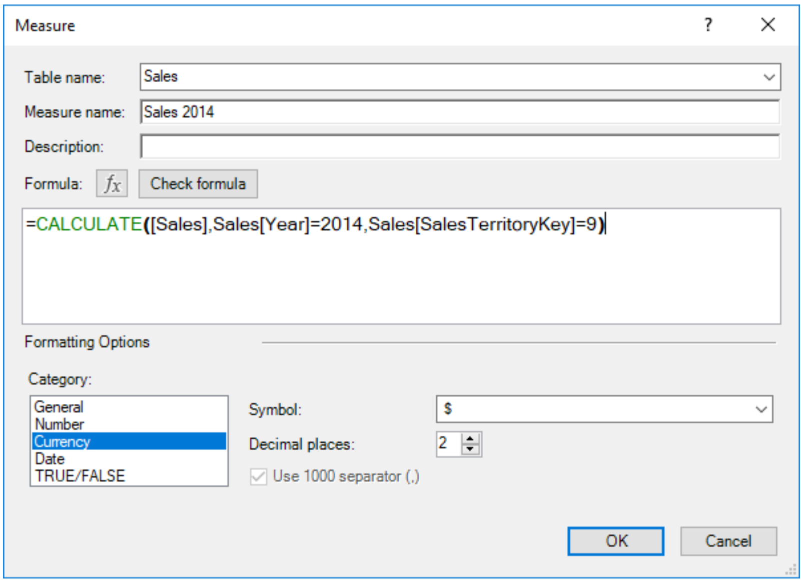


Table name: Sales

Measure name: Sales 2014

Description:

Formula: *fx*

```
=CALCULATE([Sales],Sales[Year]=2014,Sales[SalesTerritoryKey]=9)
```

Category: General, Number, Currency (selected), Date, TRUE/FALSE

Symbol: $

Decimal places: 2

☑ Use 1000 separator (,)

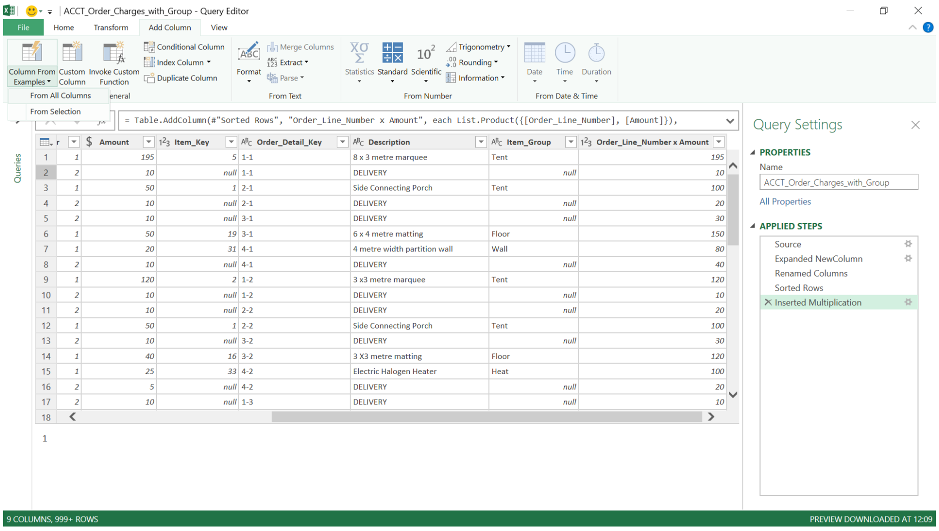
Do note that we are not limited to filter the Calendar Year or Product Category, we can also filter based on Product Types, Weeks, Months or even Sales Region. Furthermore, applying filters through this method may not be very transparent, so if you wish to apply multiple filters it is recommended that you use filters (but more on this anon).

More *Power Pivot Principles* next month.

Power Query Pointers

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

'Column From Examples' essentially pulls together a number of Power Query functions into one place, so it's very user friendly. In the Query Editor, the 'Column From Examples' can be found in the 'Add Column' tab:



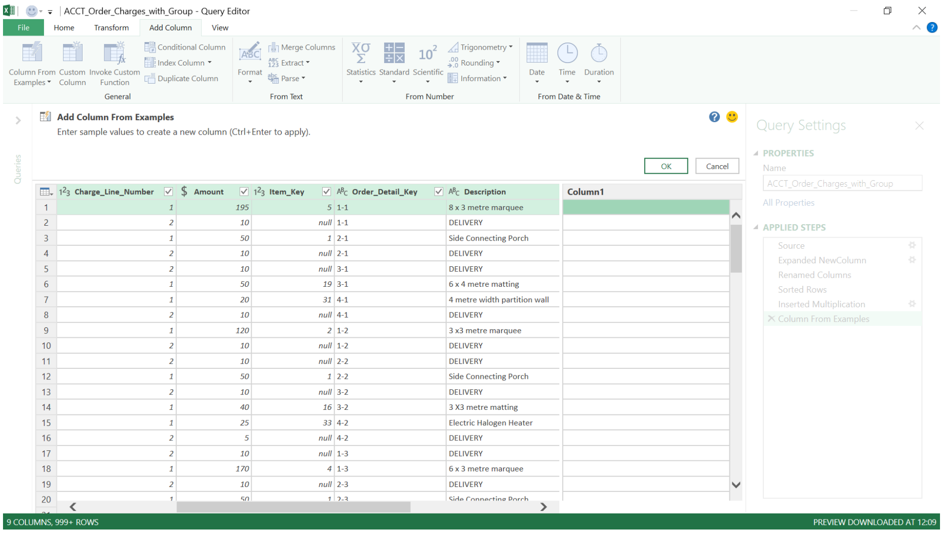
```
= Table.AddColumn(#"Sorted Rows", "Order_Line_Number x Amount", each List.Product({[Order_Line_Number], [Amount]}),
```

	r	Amount	Item_Key	Order_Detail_Key	Description	Item_Group	Order_Line_Number x Amount
1	1	195	5	1-1	8 x 3 metre marquee	Tent	195
2	2	10	null	1-1	DELIVERY	null	10
3	1	50	1	2-1	Side Connecting Porch	Tent	100
4	2	10	null	2-1	DELIVERY	null	20
5	2	10	null	3-1	DELIVERY	null	30
6	1	50	19	3-1	6 x 4 metre matting	Floor	150
7	1	20	31	4-1	4 metre width partition wall	Wall	80
8	2	10	null	4-1	DELIVERY	null	40
9	1	120	2	1-2	3 x3 metre marquee	Tent	120
10	2	10	null	1-2	DELIVERY	null	10
11	2	10	null	2-2	DELIVERY	null	20
12	1	50	1	2-2	Side Connecting Porch	Tent	100
13	2	10	null	3-2	DELIVERY	null	30
14	1	40	16	3-2	3 X3 metre matting	Floor	120
15	1	25	33	4-2	Electric Halogen Heater	Heat	100
16	2	5	null	4-2	DELIVERY	null	20
17	2	10	null	1-3	DELIVERY	null	10
18							

Name: ACCT_Order_Charges_with_Group

Applied Steps:
- Source
- Expanded NewColumn
- Renamed Columns
- Sorted Rows
- Inserted Multiplication

You have the option of 'From All Columns', which allows you to create a column by using data from any of the columns or you may use 'From Selection', which will only allow my new column to transform data from the columns that have been selected. Given we want to have a look at what we can do with this feature, we will opt for 'From All Columns':



Add Column From Examples

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

	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description	Column1
1	1	195	5	1-1	8 x 3 metre marquee	
2	2	10	null	1-1	DELIVERY	
3	1	50	1	2-1	Side Connecting Porch	
4	2	10	null	2-1	DELIVERY	
5	2	10	null	3-1	DELIVERY	
6	1	50	19	3-1	6 x 4 metre matting	
7	1	20	31	4-1	4 metre width partition wall	
8	2	10	null	4-1	DELIVERY	
9	1	120	2	1-2	3 x3 metre marquee	
10	2	10	null	1-2	DELIVERY	
11	2	10	null	2-2	DELIVERY	
12	1	50	1	2-2	Side Connecting Porch	
13	2	10	null	3-2	DELIVERY	
14	1	40	16	3-2	3 X3 metre matting	
15	1	25	33	4-2	Electric Halogen Heater	
16	2	5	null	4-2	DELIVERY	
17	2	10	null	1-3	DELIVERY	
18	1	170	4	1-3	6 x 3 metre marquee	
19	2	10	null	2-3	DELIVERY	
20	1	50	1	2-3	Side Connecting Porch	

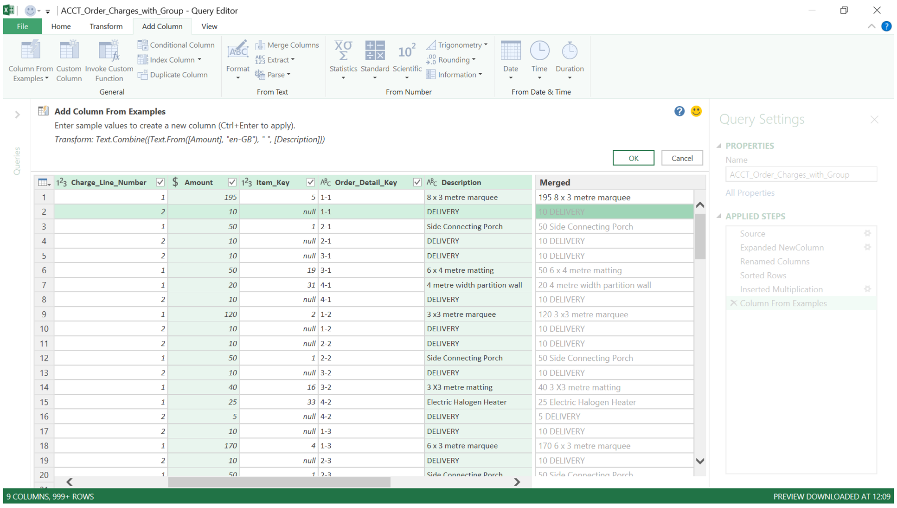
Name: ACCT_Order_Charges_with_Group

Applied Steps:
- Source
- Expanded NewColumn
- Renamed Columns
- Sorted Rows
- Inserted Multiplication
- Column From Examples

We are presented with a new empty column, which we can type into. Let's try combining columns of different types to see what happens. We will start with typing the following:

195 8x3 metre marquee

This effectively combines the **Amount** (a currency amount) with **Description** (a text field). The instruction in the pane above my query tells us to type **CTRL + ENTER** to enter my value, but you will find that **ENTER** works too.



Add Column From Examples

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

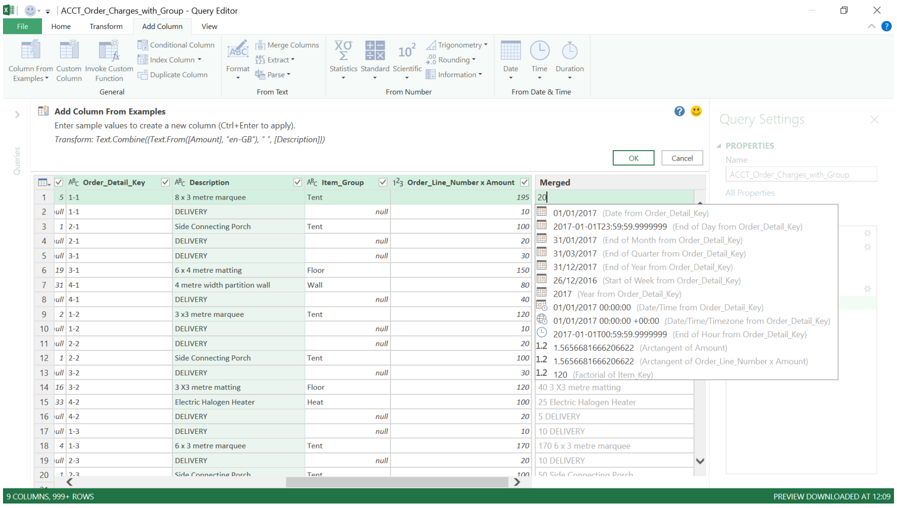
Transform: Text.Combine({Text.From([Amount], "en-GB"), " ", [Description]})

	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description	Merged
1	1	195	5	1-1	8 x 3 metre marquee	195 8 x 3 metre marquee
2	2	10	null	1-1	DELIVERY	10 DELIVERY
3	1	50	1	2-1	Side Connecting Porch	50 Side Connecting Porch
4	2	10	null	2-1	DELIVERY	10 DELIVERY
5	2	10	null	3-1	DELIVERY	10 DELIVERY
6	1	50	19	3-1	6 x 4 metre matting	50 6 x 4 metre matting
7	1	20	31	4-1	4 metre width partition wall	20 4 metre width partition wall
8	2	10	null	4-1	DELIVERY	10 DELIVERY
9	1	120	2	1-2	3 x3 metre marquee	120 3 x3 metre marquee
10	2	10	null	1-2	DELIVERY	10 DELIVERY
11	2	10	null	2-2	DELIVERY	10 DELIVERY
12	1	50	1	2-2	Side Connecting Porch	50 Side Connecting Porch
13	2	10	null	3-2	DELIVERY	10 DELIVERY
14	1	40	16	3-2	3 X3 metre matting	40 3 X3 metre matting
15	1	25	33	4-2	Electric Halogen Heater	25 Electric Halogen Heater
16	2	5	null	4-2	DELIVERY	5 DELIVERY
17	2	10	null	1-3	DELIVERY	10 DELIVERY
18	1	170	4	1-3	6 x 3 metre marquee	170 6 x 3 metre marquee
19	2	10	null	2-3	DELIVERY	10 DELIVERY
20	1	50	1	2-3	Side Connecting Porch	50 Side Connecting Porch

Not only has Power Query recognised what we are trying to do, it has filled in the rest of the values in the column so that we may check. It's like Flash Fill (Excel 2013 onwards) but with formulae! The new column has now been given the name **Merged**, and the **M** code behind the transformation is displayed in the pane above our query:

Text.Combine({Text.From[Amount] "en-GB"), " ", [Description]})

This is a satisfying result. We can choose to accept the values and move on or we can go back and type something else into the first row of my column instead.



Add Column From Examples

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

Transform: Text.Combine({Text.From([Amount], "en-GB"), " ", [Description]})

		Order_Detail_Key	Description	Item_Group	Order_Line_Number x Amount	Merged
1	5	1-1	8 x 3 metre marquee	Tent	195	20
2	ull	1-1	DELIVERY	null	10	
3	1	2-1	Side Connecting Porch	Tent	100	
4	ull	2-1	DELIVERY	null	20	
5	ull	3-1	DELIVERY	null	30	
6	19	3-1	6 x 4 metre matting	Floor	150	
7	31	4-1	4 metre width partition wall	Wall	80	
8	ull	4-1	DELIVERY	null	40	
9	2	1-2	3 x3 metre marquee	Tent	120	
10	ull	1-2	DELIVERY	null	10	
11	ull	2-2	DELIVERY	null	20	
12	1	2-2	Side Connecting Porch	Tent	100	
13	ull	3-2	DELIVERY	null	30	
14	16	3-2	3 X3 metre matting	Floor	120	40 3 X3 metre matting
15	33	4-2	Electric Halogen Heater	Heat	100	25 Electric Halogen Heater
16	ull	4-2	DELIVERY	null	20	5 DELIVERY
17	ull	1-3	DELIVERY	null	10	10 DELIVERY
18	4	1-3	6 x 3 metre marquee	Tent	170	170 6 x 3 metre marquee
19	ull	2-3	DELIVERY	null	20	10 DELIVERY
20	1	2-3	Side Connecting Porch	Tent	100	50 Side Connecting Porch

01/01/2017 (Date from Order_Detail_Key)
2017-01-01T23:59:59.9999999 (End of Day from Order_Detail_Key)
31/01/2017 (End of Month from Order_Detail_Key)
31/03/2017 (End of Quarter from Order_Detail_Key)
31/12/2017 (End of Year from Order_Detail_Key)
26/12/2016 (Start of Week from Order_Detail_Key)
2017 (Year from Order_Detail_Key)
01/01/2017 00:00:00 (Date/Time from Order_Detail_Key)
01/01/2017 00:00:00 +00:00 (Date/Time/Timezone from Order_Detail_Key)
2017-01-01T00:59:59.9999999 (End of Hour from Order_Detail_Key)
1.2 1.5656681666206622 (Arctangent of Amount)
1.2 1.5656681666206622 (Arctangent of Order_Line_Number x Amount)
1.2 120 (Factorial of Item_Key)

Power Query tries to make sense of what you are entering as soon as you start typing. '20' is likely to be a date, so various date functions are suggested. You may find that you are able to type data in up to three rows before Power Query either follows the pattern of what I am doing or issues a message telling me that the transformation is not recognised (and to start again). Later versions allow more iterations.

The transformations available are listed on the Microsoft help pages for the 'Column from Examples' description:

Number transformations

Average
Count distinct values
Count values
Minimum
Maximum
Median
Percent of
Power

Text transformations

Text.Combine (supports combination of literal strings and entire column values)
Text.Replace
Text.Start
Text.Middle
Text.End
Text.BeforeDelimiter
Text.AfterDelimiter
Text.BetweenDelimiters

Date transformations

Date.Day
Date.DayOfWeek
Date.DayOfWeekName
Date.DayOfYear
Date.Month
Date.MonthName
Date.QuarterOfYear
Date.WeekOfMonth
Date.WeekOfYear
Date.Year
Date > Age
Date > Year > Start of Year
Date > Year > End of Year
Date > Month > Start of Month
Date > Month > End of Month
Date > Quarter > Start of Quarter
Date > Month > Days in Month
Date > Quarter > End of Quarter
Date > Week > Start of Week
Date > Week > End of Week
Date > Day > Day of Month
Date > Day > Start of Day
Date > Day > End of Day

Time transformations

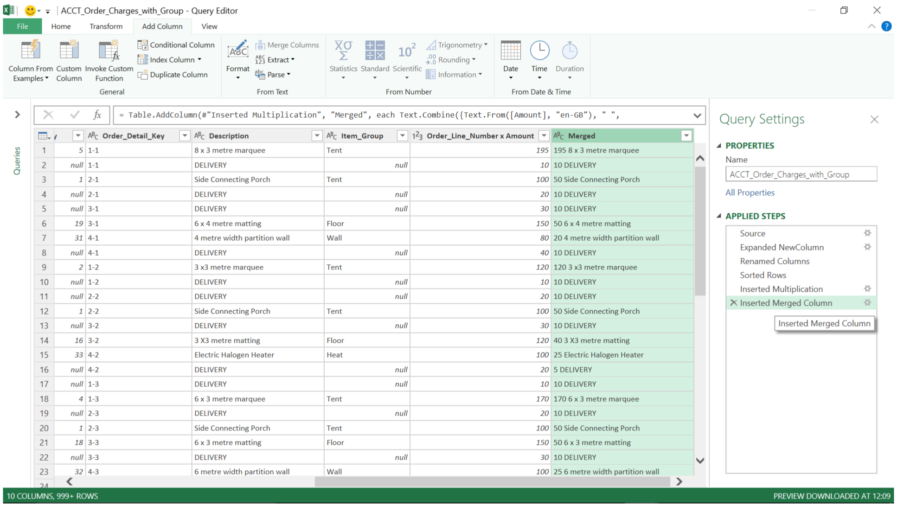
Time.Hour
Time.Minute
Time.Second

Date/Time transformations

Subtract days
Combine Date and Time
Earliest
Latest
Subtract time

This is a substantial list for now, but it will only improve over time!

Back in our query, let's decide to go ahead and create a **Merge** column which combines the **Amount** and **Description** columns. Click 'OK' in the pane above the query, and a new column appears in the query. The transformation step is added as 'Inserted Merged Column':



ACCT_Order_Charges_with_Group - Query Editor

```
= Table.AddColumn(#"Inserted Multiplication", "Merged", each Text.Combine({Text.From([Amount], "en-GB"), " ",
```

	y	Order_Detail_Key	Description	Item_Group	Order_Line_Number x Amount	Merged
1	5	1-1	8 x 3 metre marquee	Tent	195	195 8 x 3 metre marquee
2	null	1-1	DELIVERY	null	10	10 DELIVERY
3	1	2-1	Side Connecting Porch	Tent	100	50 Side Connecting Porch
4	null	2-1	DELIVERY	null	20	10 DELIVERY
5	null	3-1	DELIVERY	null	30	10 DELIVERY
6	19	3-1	6 x 4 metre matting	Floor	150	50 6 x 4 metre matting
7	31	4-1	4 metre width partition wall	Wall	80	20 4 metre width partition wall
8	null	4-1	DELIVERY	null	40	10 DELIVERY
9	2	1-2	3 x3 metre marquee	Tent	120	120 3 x3 metre marquee
10	null	1-2	DELIVERY	null	10	10 DELIVERY
11	null	2-2	DELIVERY	null	20	10 DELIVERY
12	1	2-2	Side Connecting Porch	Tent	100	50 Side Connecting Porch
13	null	3-2	DELIVERY	null	30	10 DELIVERY
14	16	3-2	3 X3 metre matting	Floor	120	40 3 X3 metre matting
15	33	4-2	Electric Halogen Heater	Heat	100	25 Electric Halogen Heater
16	null	4-2	DELIVERY	null	20	5 DELIVERY
17	null	1-3	DELIVERY	null	10	10 DELIVERY
18	4	1-3	6 x 3 metre marquee	Tent	170	170 6 x 3 metre marquee
19	null	2-3	DELIVERY	null	20	10 DELIVERY
20	1	2-3	Side Connecting Porch	Tent	100	50 Side Connecting Porch
21	18	3-3	6 x 3 metre matting	Floor	150	50 6 x 3 metre matting
22	null	3-3	DELIVERY	null	30	10 DELIVERY
23	32	4-3	6 metre width partition wall	Wall	100	25 6 metre width partition wall

Query Settings

PROPERTIES

Name

ACCT_Order_Charges_with_Group

All Properties

APPLIED STEPS

- Source
- Expanded NewColumn
- Renamed Columns
- Sorted Rows
- Inserted Multiplication
- Inserted Merged Column

10 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 12:09

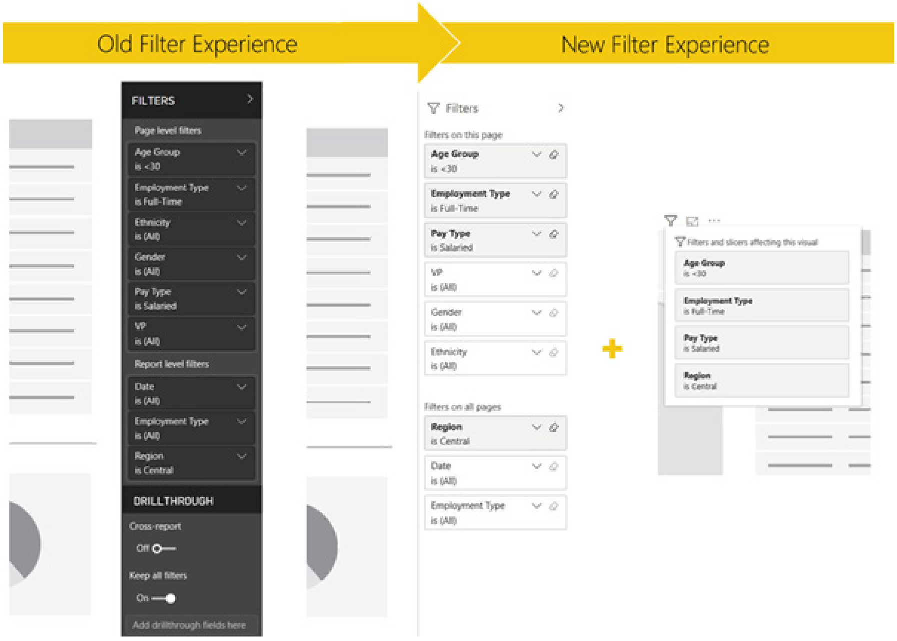
One of the main benefits of this feature is that it allows access to a lot of **M** code functions without you needing to understand **M** code or needing to know where all the separate functions are on the Ribbon. We look forward to seeing more functions added to this facility in due course.

More next month!

Power BI Desktop Update for March

March saw the release of new button actions, a multi-column sort for tables, a dual axis for line chart, a search for the Filter pane, updates to the decomposition tree and plenty more. In this update, Microsoft also released a Preview of an enhanced dataset metadata feature, which they deemed "…will be foundational for XMLA read / write and for management of Power BI dataset and the migration of Analysis Services workloads to Power BI".

Whilst you read these updates, do bear in mind an important update scheduled for April 2020. From the next update, the modern Filter experience will automatically go live in all reports for all users in Power BI Desktop and Power BI Service. This also means that the old Filter pane will no longer be available. Therefore, if you have not yet upgraded some (all?) of your reports, we strongly recommend doing so NOW, so you can tweak or customise any of the formatting before your reports get auto-upgraded in April.



Anyway, here's the complete list of March updates:

Reporting

- New action types for buttons
    - Page navigation
    - Drill through (Preview)
- Multi-column sort for tables
- Dual axis for line chart
- Filter pane search
- Updates to decomposition tree
    - Bars per level formatting option
    - Updates to behaviour of data bar length
- New Ribbon is now on by default

Modelling

- New **DAX** function
    - **COALESCE**

Visualisations

- Waterfall chart by xViz
- Updates to ArcGIS Maps
    - New connection options
    - Enterprise support is now Generally Available
    - Support for multiple reference layers
    - New table of contents

Template Applications

- Azure DevOps dashboard by Data Maru
- TeamsPower by Encamina

Data Preparation

- Query diagnostics is now Generally Available

Data Connectivity

- Hive LLAP connector
- Cognite connector

Other

- Enhanced dataset metadata (Preview)
- Using default system credentials for web proxy.

Let's take a look at each in turn.

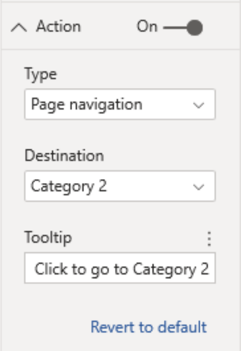
New action types for buttons

Two new buttons have been added with this latest update.

PAGE NAVIGATION

Previously, when you wanted to create a button to navigate to another page you first had to save a bookmark for your destination page, and then you had to create a button with a bookmark action that references that bookmark you saved. To help reduce the number of steps needed for you to create such a navigation experience, this release introduces a built-in page navigation action. With this new action you can quickly build an entire navigation experience without having to save or manage any bookmarks at all.

To set up a page navigation button, create a button with Page navigation as the action type, and select the Destination page:



Action On

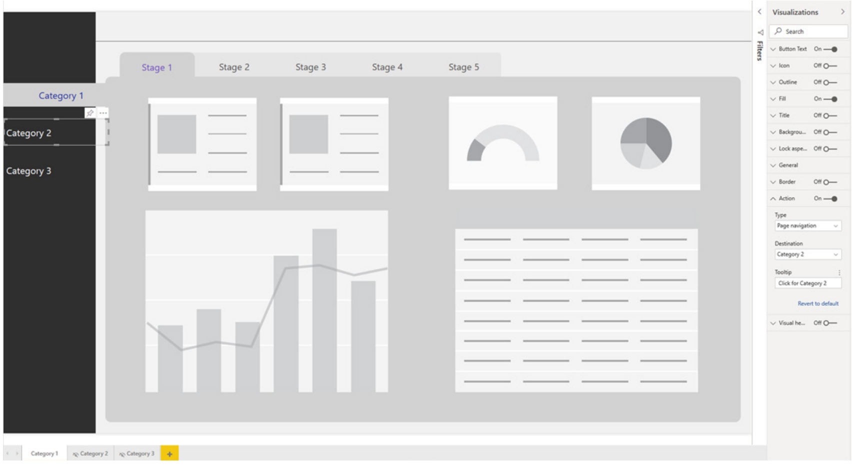
Type: Page navigation

Destination: Category 2

Tooltip: Click to go to Category 2

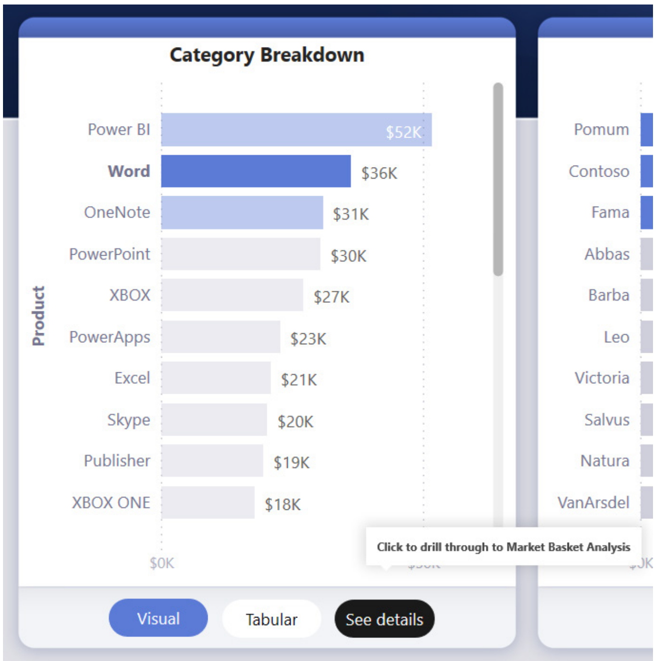
Revert to default

With this new action you, can quickly build a Navigation pane and avoid having to edit and manage bookmarks if you want to change which pages to show in your Navigation pane.

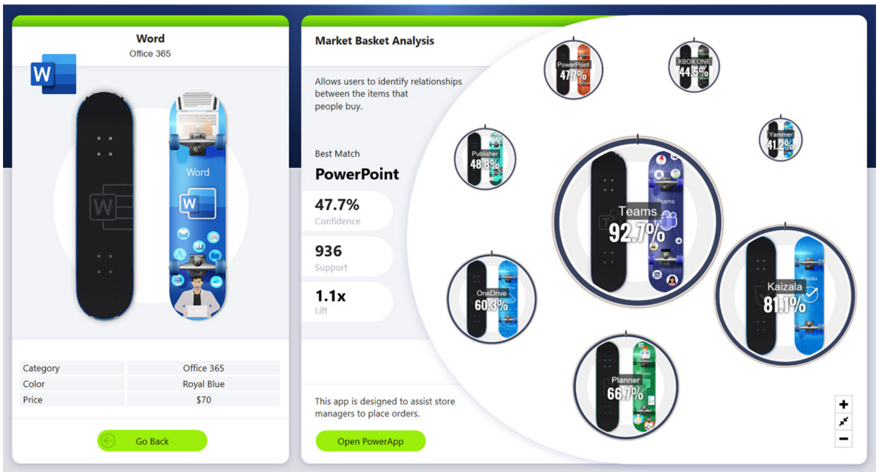


DRILL THROUGH (PREVIEW)

In addition to page navigation, there's a new drill through action as a Preview. This action type will allow you to contextually navigate to a drill through page. This feature can be useful if you want to increase the discoverability of important drill through scenarios in your reports.



In this example, after the user selects Word, the button will be enabled, and they can drill through to the Market Basket Analysis page, *viz*.


Word
Office 365

Category	Office 365
Color	Royal Blue
Price	$70

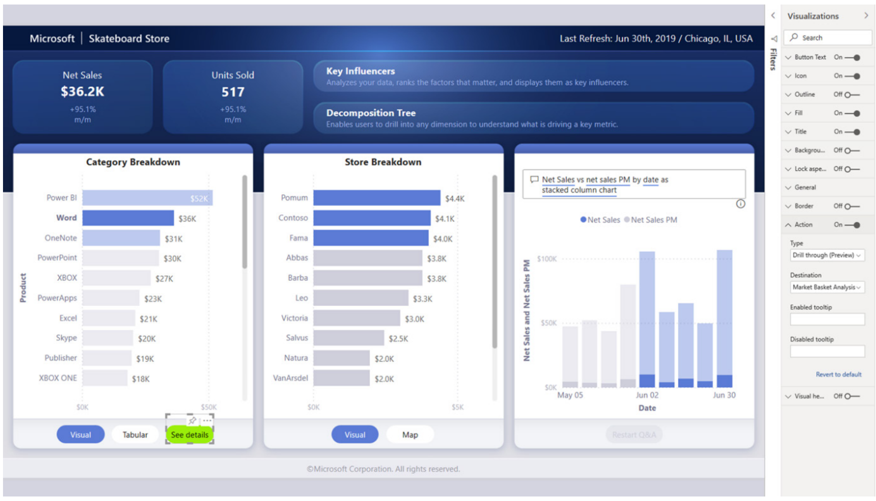
Market Basket Analysis

Allows users to identify relationships between the items that people buy.

Best Match
PowerPoint
47.7% Confidence
936 Support
1.1x Lift

This app is designed to assist store managers to place orders.

To set up a drill through button, you will first need to have a valid drill through page within your report. Then, you will need to create a button with 'Drill through' as the action type and select the drill through page as the Destination. Because the drill through button has dual states (when drill through is enabled compared to disabled), you will see that there are two ToolTip options:


Microsoft | Skateboard Store
Last Refresh: Jun 30th, 2019 / Chicago, IL, USA

Net Sales $36.2K +95.1% m/m
Units Sold 517 +95.1% m/m

Key Influencers
Analyzes your data, ranks the factors that matter, and displays them as key influencers.

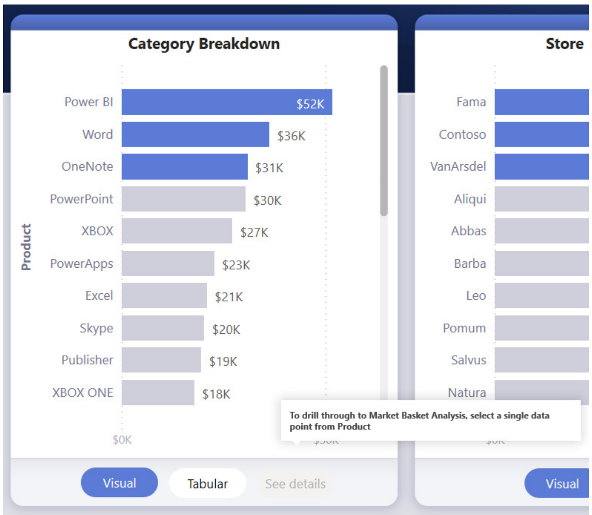
Decomposition Tree
Enables users to drill into any dimension to understand what is driving a key metric.

Net Sales vs net sales PM by date as stacked column chart

©Microsoft Corporation. All rights reserved.

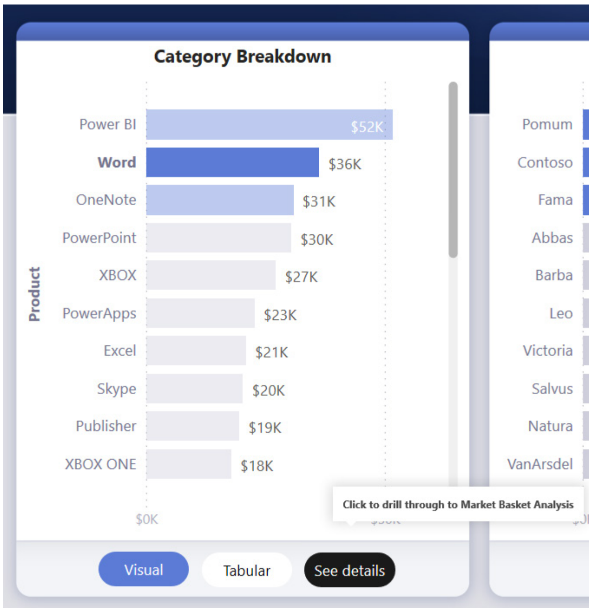
Visualizations
Button Text On
Icon On
Outline Off
Fill On
Title On
Backgrou... Off
Lock aspe... Off
General
Border Off
Action On
Type: Drill through (Preview)
Destination: Market Basket Analysis
Enabled tooltip
Disabled tooltip
Revert to default
Visual he... Off

However, feel free to leave them blank to use the auto-generated ToolTips, which are based on the destination and drill through field(s). Here's an example of the auto-generated tooltip when the button is disabled:



To drill through to Market Basket Analysis, select a single data point from Product

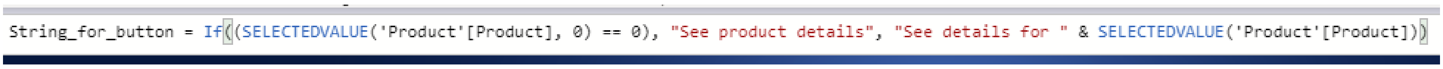
In comparison, here's an example of the auto-generated tooltip when the button is enabled:



Click to drill through to Market Basket Analysis

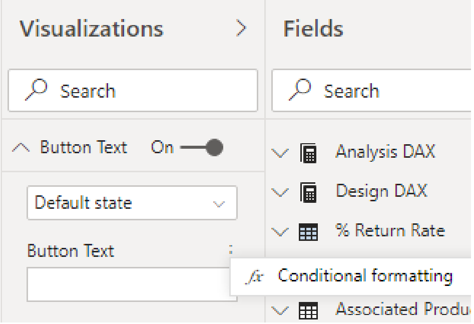
However, if you would like to provide custom ToolTips, you can always input a static string. You can also use conditional formatting to change the button text based on the selected value of a field. To do this you will need to create a measure that will output the desired string based on the DAX function, **SELECTEDVALUE**.

Here's an example measure that will output "See product details" if a single Product is NOT selected; otherwise, it will output "See details for [the selected Product]":

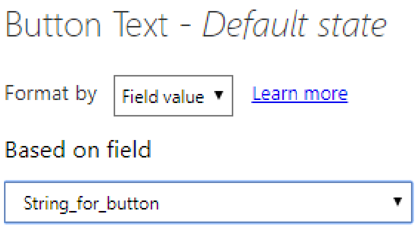


```
String_for_button = IF((SELECTEDVALUE('Product'[Product], 0) == 0), "See product details", "See details for " & SELECTEDVALUE('Product'[Product]))
```

Once you've created this measure, you will select the conditional formatting option for the button text:



Then, you just select the measure you created for the button text:

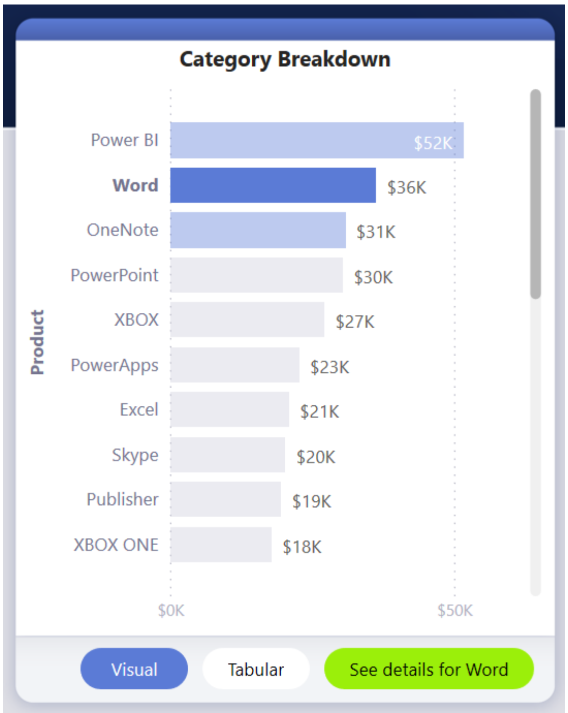


Button Text - *Default state*

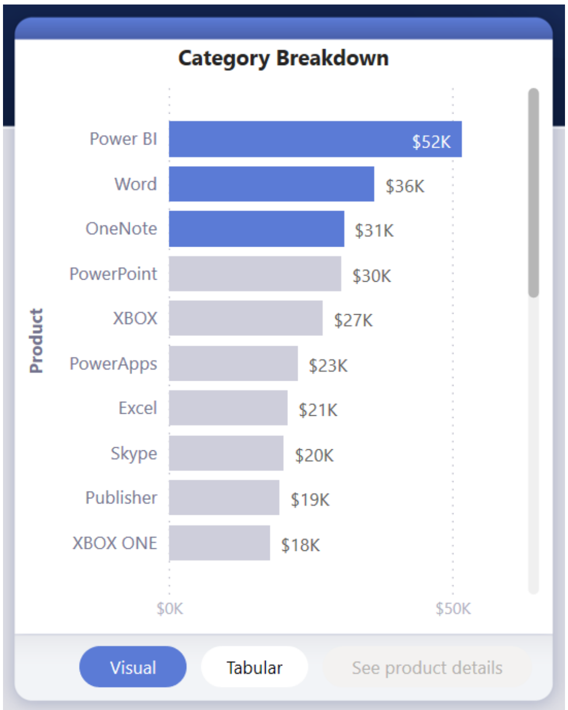
Format by: Field value ▾ Learn more

Based on field: String_for_button ▾

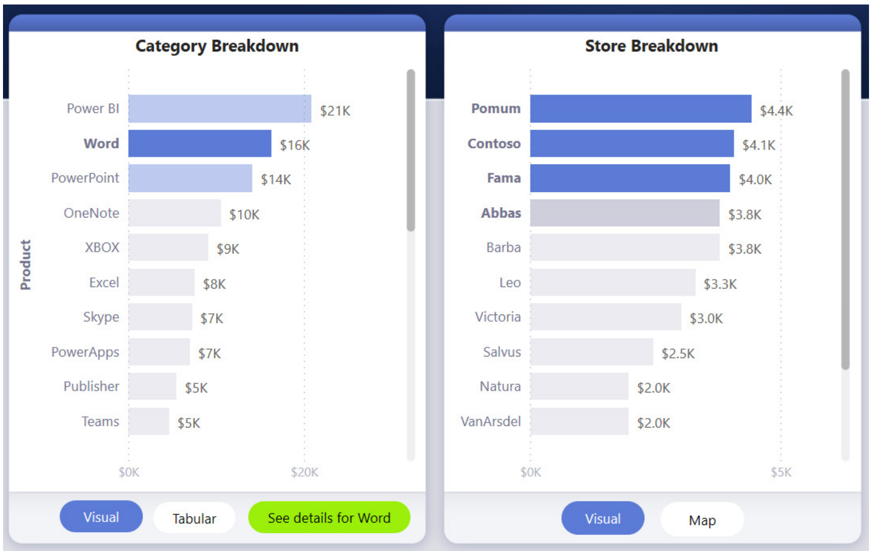
So, here's the result when a single product is selected:



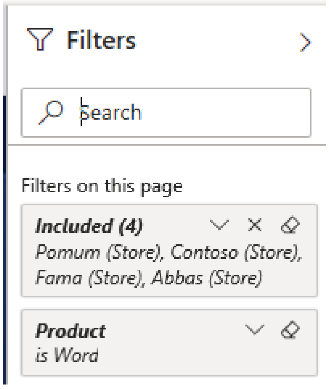
Then, here's the result when either no products are selected, or more than one product is selected:



The button will work like normal drill through, so you can also pass filters on additional fields by cross-filtering the visual(s) which contain the drill through field. For example, using **CTRL + Click** and cross-filtering, you can pass multiple filters on **Store** to the drill through page, because the selections cross-filter the visual that contains **Product** (the drill through field):



Therefore, when you invoke the drill through, you will see filters on both **Store** and **Product** being passed through:

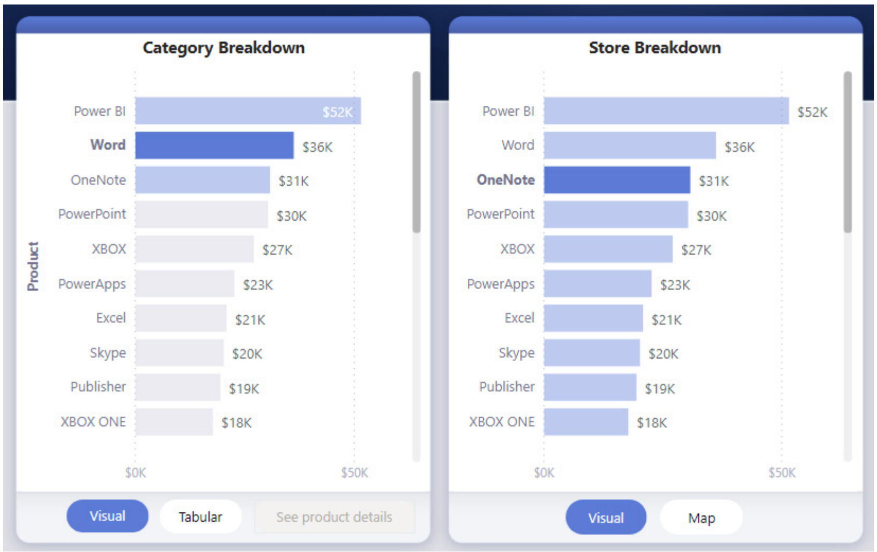


Filters on this page

Included (4)
Pomum (Store), Contoso (Store), Fama (Store), Abbas (Store)

Product
is Word

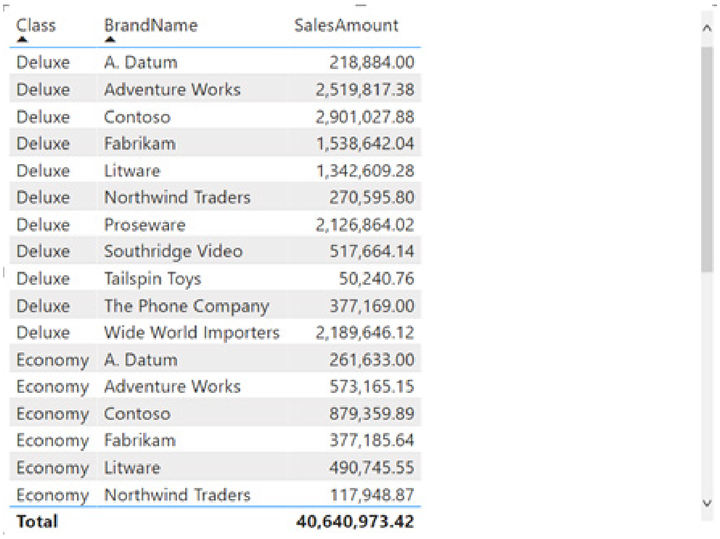
Since the drill through button is not directly tied to a single visual, if there is ambiguity in your selection, then the button will be disabled. In this example, the button is disabled because there are two visuals that both contain a single-selection on **Product**, so there is ambiguity on which data point from which visual to tie the drill through action to:



Multi-column sort for tables

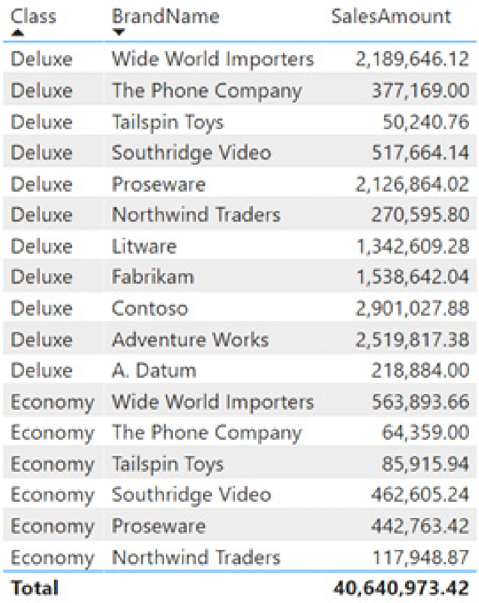
To add more columns to the sort order, **SHIFT + Click** the column header you would like to add next in the sort order. If you **SHIFT + Click** a second time on the same column, this will change the sort direction for that column. Furthermore, if you **SHIFT + Click** a column you have previously added to the sort order, this will move that column to the back of the sort order.

For example, if you click **Class** and then **Shift + click BrandName**, then the table is sorted first by **Class**, then by **BrandName**:



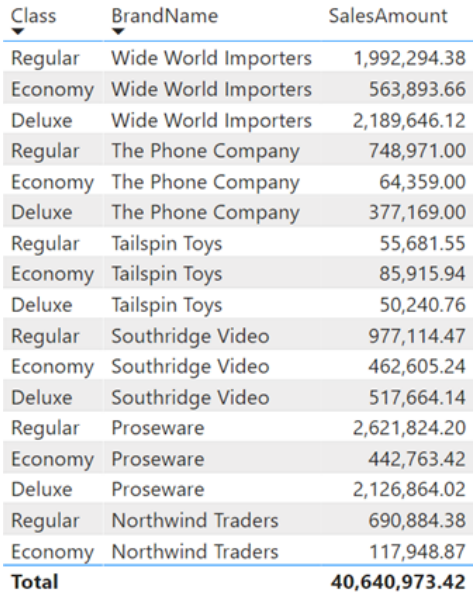
Class ▲	BrandName ▲	SalesAmount
Deluxe	A. Datum	218,884.00
Deluxe	Adventure Works	2,519,817.38
Deluxe	Contoso	2,901,027.88
Deluxe	Fabrikam	1,538,642.04
Deluxe	Litware	1,342,609.28
Deluxe	Northwind Traders	270,595.80
Deluxe	Proseware	2,126,864.02
Deluxe	Southridge Video	517,664.14
Deluxe	Tailspin Toys	50,240.76
Deluxe	The Phone Company	377,169.00
Deluxe	Wide World Importers	2,189,646.12
Economy	A. Datum	261,633.00
Economy	Adventure Works	573,165.15
Economy	Contoso	879,359.89
Economy	Fabrikam	377,185.64
Economy	Litware	490,745.55
Economy	Northwind Traders	117,948.87
Total		**40,640,973.42**

Next, if you **SHIFT + Click** on **BrandName** again, this will change the sort direction for **BrandName**:



Class ▲	BrandName ▼	SalesAmount
Deluxe	Wide World Importers	2,189,646.12
Deluxe	The Phone Company	377,169.00
Deluxe	Tailspin Toys	50,240.76
Deluxe	Southridge Video	517,664.14
Deluxe	Proseware	2,126,864.02
Deluxe	Northwind Traders	270,595.80
Deluxe	Litware	1,342,609.28
Deluxe	Fabrikam	1,538,642.04
Deluxe	Contoso	2,901,027.88
Deluxe	Adventure Works	2,519,817.38
Deluxe	A. Datum	218,884.00
Economy	Wide World Importers	563,893.66
Economy	The Phone Company	64,359.00
Economy	Tailspin Toys	85,915.94
Economy	Southridge Video	462,605.24
Economy	Proseware	442,763.42
Economy	Northwind Traders	117,948.87
Total		**40,640,973.42**

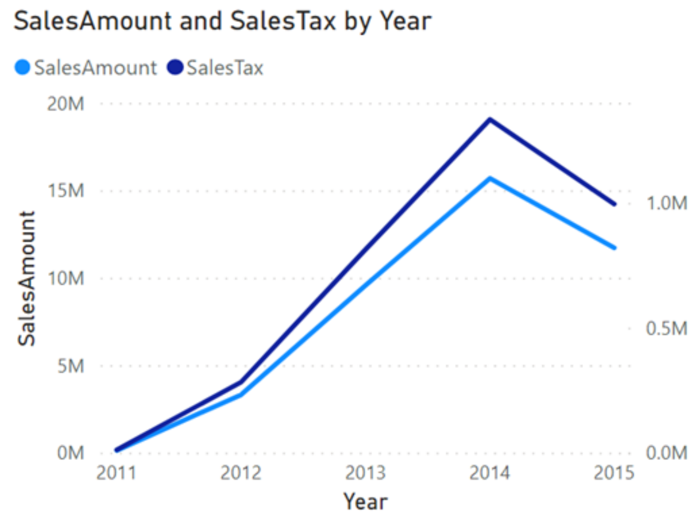
Finally, If you **SHIFT + Click** on **Class** again, this will move it to the back of the sort order, meaning that the table is now sorted by **BrandName** first and then **Class**. This will also reverse the sort direction of **Class**, since the column had been previously selected.



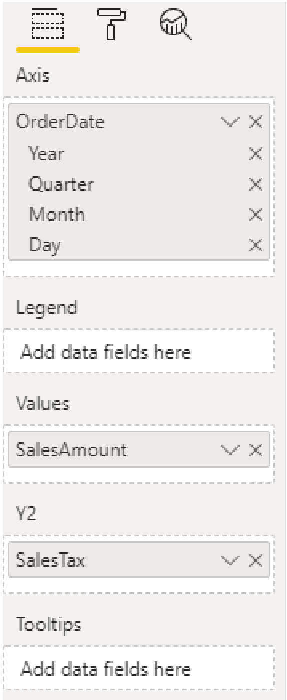
Class ▼	BrandName ▼	SalesAmount
Regular	Wide World Importers	1,992,294.38
Economy	Wide World Importers	563,893.66
Deluxe	Wide World Importers	2,189,646.12
Regular	The Phone Company	748,971.00
Economy	The Phone Company	64,359.00
Deluxe	The Phone Company	377,169.00
Regular	Tailspin Toys	55,681.55
Economy	Tailspin Toys	85,915.94
Deluxe	Tailspin Toys	50,240.76
Regular	Southridge Video	977,114.47
Economy	Southridge Video	462,605.24
Deluxe	Southridge Video	517,664.14
Regular	Proseware	2,621,824.20
Economy	Proseware	442,763.42
Deluxe	Proseware	2,126,864.02
Regular	Northwind Traders	690,884.38
Economy	Northwind Traders	117,948.87
Total		**40,640,973.42**

Dual axis for line chart

This update allows you to now include a second **y**-axis on your line charts, enabling you to plot two trends with different ranges along the same **x**-axis progression.



To use the second axis, drag fields into the new secondary **y**-axis field well, and they will be drawn into the visual against a **y**-axis on the right side of the chart. You can format these lines and this axis in the associated card of the Formatting pane. Otherwise, the formatting will mirror that of your primary **y**-axis.



Axis

OrderDate
Year
Quarter
Month
Day

Legend

Add data fields here

Values

SalesAmount

Y2

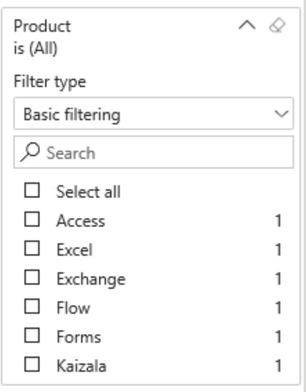
SalesTax

Tooltips

Add data fields here

Filter pane search

For a while now, Power BI has supported Filter card search, which allows you to search across the values within a field.



Product
is (All)

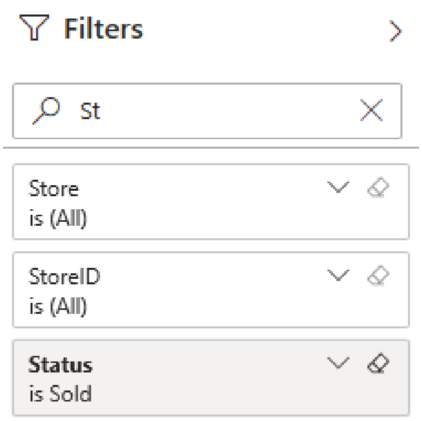
Filter type

Basic filtering

Search

□ Select all	
□ Access	1
□ Excel	1
□ Exchange	1
□ Flow	1
□ Forms	1
□ Kaizala	1

This update introduces additional support for Filter *pane* search, which will allow you to search across your Filter cards. This feature is particularly useful if you have several different Filter cards in your Filter pane and need help finding the ones of interest.



Filters

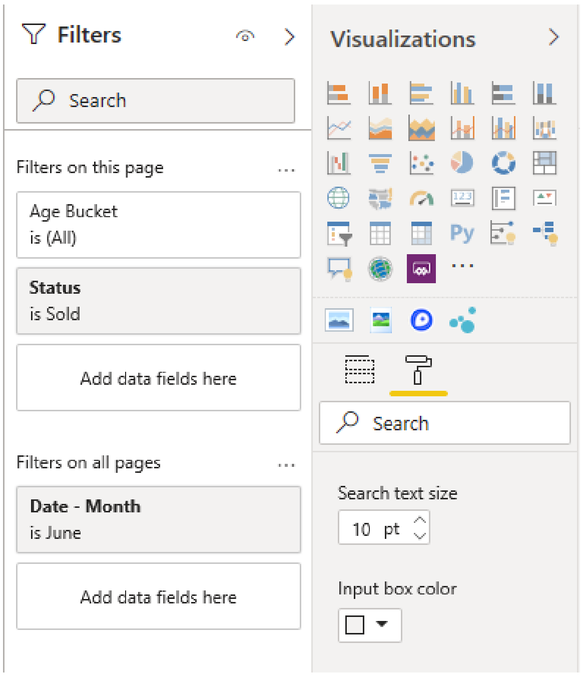
St

Store
is (All)

StoreID
is (All)

Status
is Sold

Additionally, you can format and tweak the search box, just as can with the other elements of the Filter pane.



Filters

Search

Filters on this page

Age Bucket
is (All)

Status
is Sold

Add data fields here

Filters on all pages

Date - Month
is June

Add data fields here

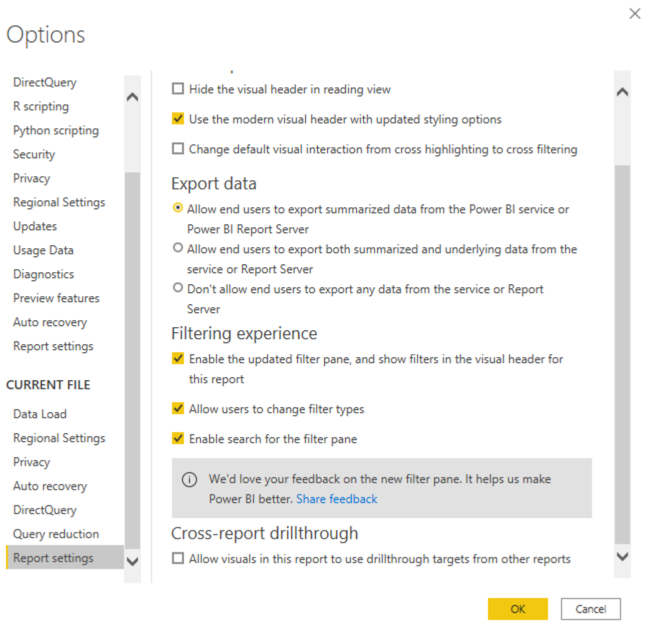
Visualizations

Search

Search text size: 10 pt

Input box color: □

While this Filter pane search feature will be on by default, you can also choose to turn it on or off by selecting ‘Enable search for Filter pane’ in the ‘Report settings’ of the Options dialog.



Options

DirectQuery
R scripting
Python scripting
Security
Privacy
Regional Settings
Updates
Usage Data
Diagnostics
Preview features
Auto recovery
Report settings

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto recovery
DirectQuery
Query reduction
Report settings

☐ Hide the visual header in reading view

☑ Use the modern visual header with updated styling options

☐ Change default visual interaction from cross highlighting to cross filtering

Export data

● Allow end users to export summarized data from the Power BI service or Power BI Report Server

○ Allow end users to export both summarized and underlying data from the service or Report Server

○ Don't allow end users to export any data from the service or Report Server

Filtering experience

☑ Enable the updated filter pane, and show filters in the visual header for this report

☑ Allow users to change filter types

☑ Enable search for the filter pane

ⓘ We'd love your feedback on the new filter pane. It helps us make Power BI better. Share feedback

Cross-report drillthrough

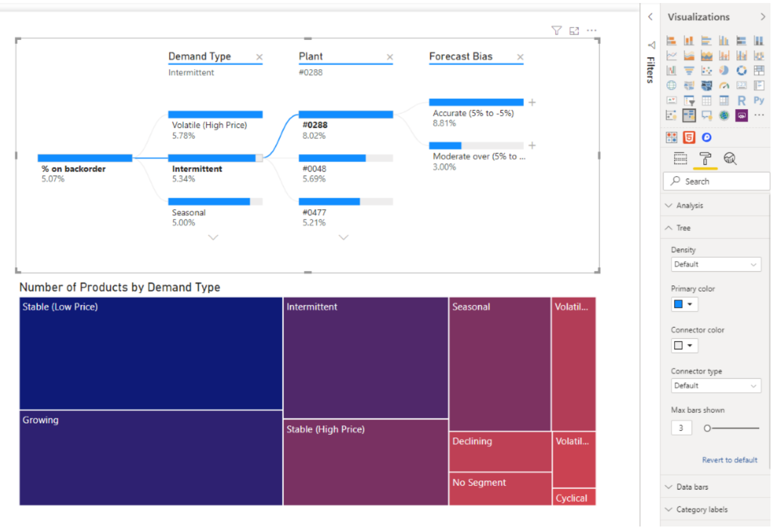
☐ Allow visuals in this report to use drillthrough targets from other reports

OK Cancel

Updates to decomposition tree

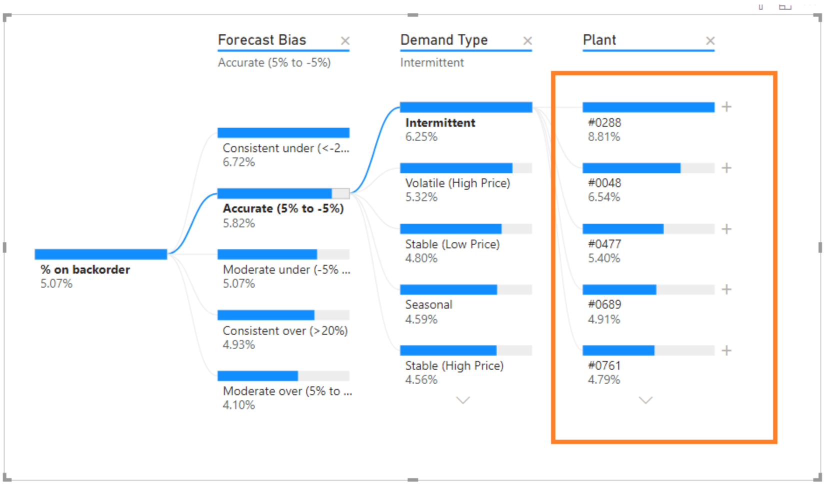
BARS PER LEVEL FORMATTING OPTION

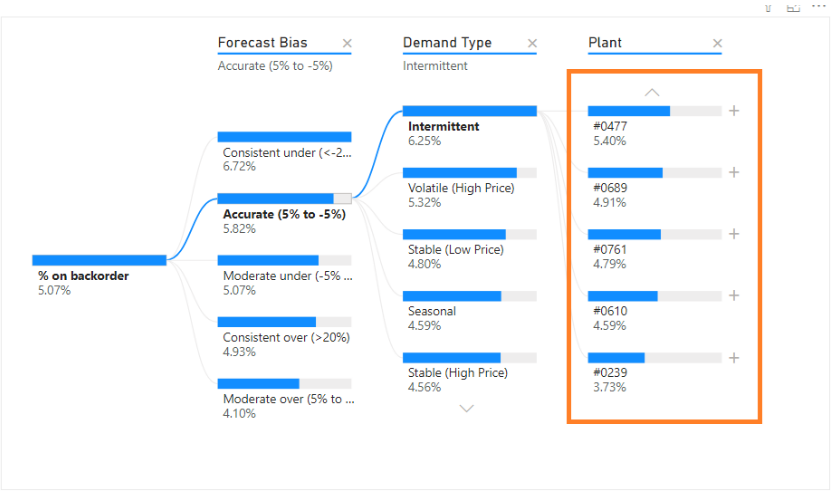
The decomposition tree now supports modifying the maximum bars shown per level. This is a formatting option found in the Tree card. The default is 10, and users MAY select values between three (3) and 30. Setting a low number is particularly handy if you don't want the decomposition tree to take up too much space on the canvas. It's easy to cycle through the bars by using the up and down arrows in the visual.


Density: Default
Primary color
Connector color
Connector type: Default
Max bars shown: 3

UPDATES TO BEHAVIOUR OF DATA BAR LENGTH

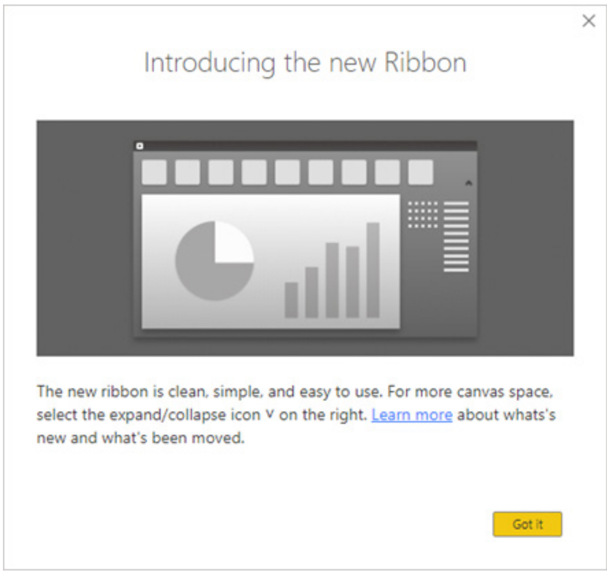
Previously, the data bar length was based on the values currently shown in the tree and not all the values in the range. If a user scrolled up or down the tree, the first data bar would return to the maximum. This behaviour has been improved and the global maximum is now remembered when scrolling up and down the visual. In the example below, a user selects the down arrow in 'Plant' and the next batch of results is shown. The data bar for Plant #0477 is no longer full as its value is calculated based on the maximum value in the whole tree (Plant #0288):





New Ribbon is now on by default

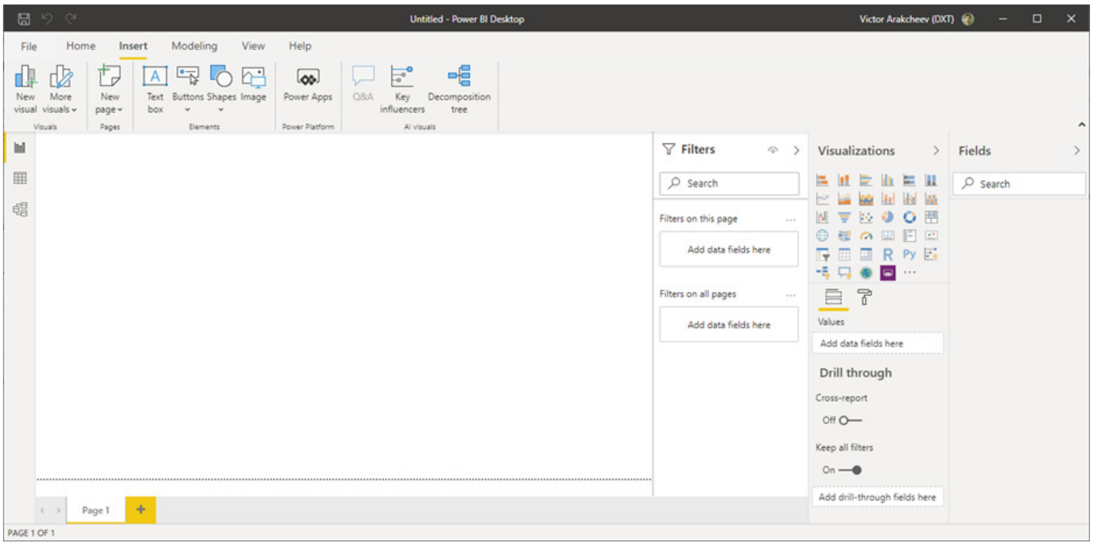
Starting this month, the modern Ribbon will be on by default. You'll see a dialog announcing this change:



Introducing the new Ribbon

The new ribbon is clean, simple, and easy to use. For more canvas space, select the expand/collapse icon ˅ on the right. Learn more about whats's new and what's been moved.

There are also some new updates to the Insert tab in the Ribbon. You can now insert the Power Apps visual and AI visuals directly from the Ribbon:



New DAX function: COALESCE

The **COALESCE** function returns the first expression that does not evaluate to **BLANK**. If all expressions evaluate to **BLANK**, **BLANK** is returned. Its syntax is as follows:

COALESCE(expression, expression[, expression]…)

It has the following arguments:

- **expression:** any DAX expression that returns a scalar expression. It should be noted that input expressions may be of different data types.

This function will return a scalar value coming from one of the **expressions**, or **BLANK** if all expressions evaluate to **BLANK**.

For example,

EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }

returns 10, which is the first **expression** that does not evaluate to **BLANK**.

As another example,

EVALUATE { COALESCE(SUM(FactInternetSales[SalesAmount]), 0) }

returns the sum of all values in the **SalesAmount** column in the **FactInternetSales** table, or else zero (0). This can be used to convert **BLANK** values of the total sales to zero (0).

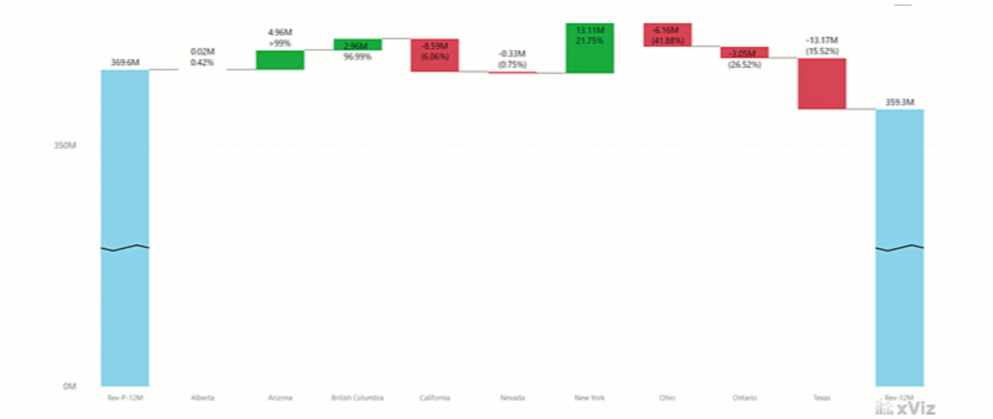
Waterfall chart by xViz

The waterfall chart by xViz is a visual that displays the cumulative effect of sequential positive and negative values contributing to the final total. Examples where it may be used:

- **variance analysis:** drill down multiple levels to track absolute and relative variance across members
- **breakdown analysis:** monitor each member's contribution to the increase or decrease in totals (sales, revenue, *etc.*)
- **display running totals:** track the updated summation each time a new member adds to the total
- **display multiple subtotals and total column:** ability to display the sum or the cumulative total for any set of members.

Some of the key features include:

- Custom sort option: sort the waterfall with another column
- Breadcrumb for drill down scenario
- Switch between horizontal and vertical orientation
- Runtime deviation bar
- Axis break
- Semantic number formatting
- Summary table: tabular view of data.



Updates to ArcGIS Maps

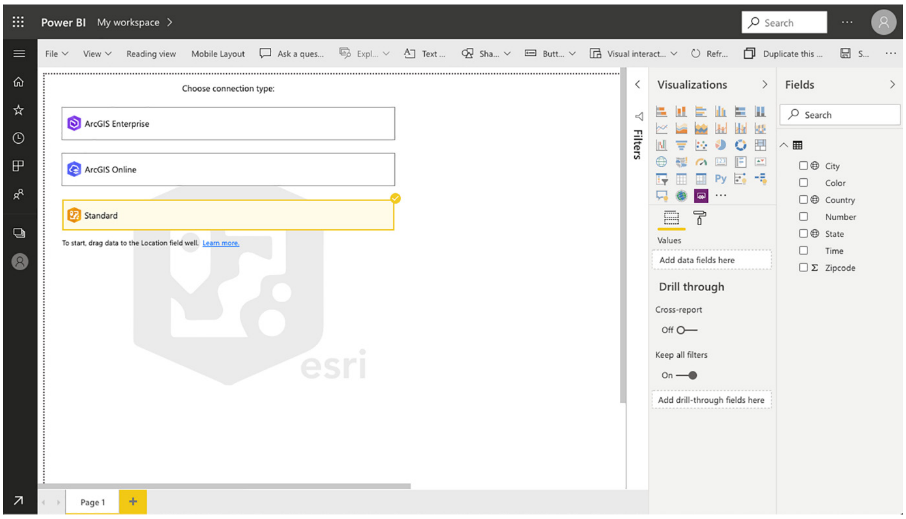
ArcGIS Maps provides you with the ability to build informative maps using your data within Microsoft Power BI. It works with all three versions of Power BI: Free, Pro and Premium. This release sees four key updates:

- new connection options
- Enterprise support is now Generally Available
- support for multiple reference layers
- new table of contents.

NEW CONNECTION OPTIONS

With this release of Power BI, when you click on the ArcGIS Maps for Power BI visual within Power BI, you will now see three connection options:

- ArcGIS Enterprise
- ArcGIS Online
- Standard.


Choose connection type:

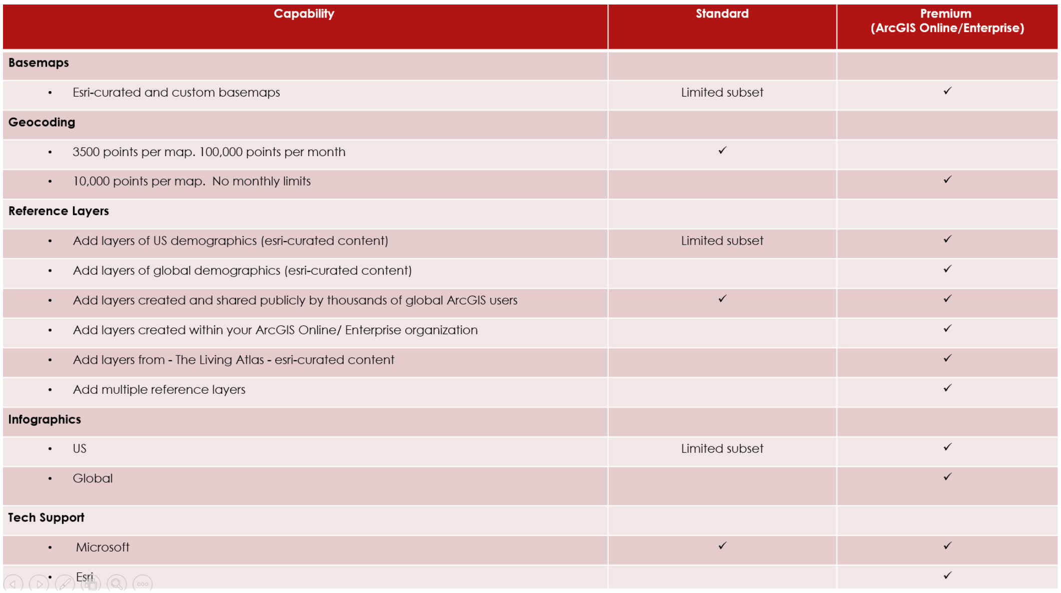
ArcGIS Enterprise

ArcGIS Online

Standard

To start, drag data to the Location field well. Learn more.

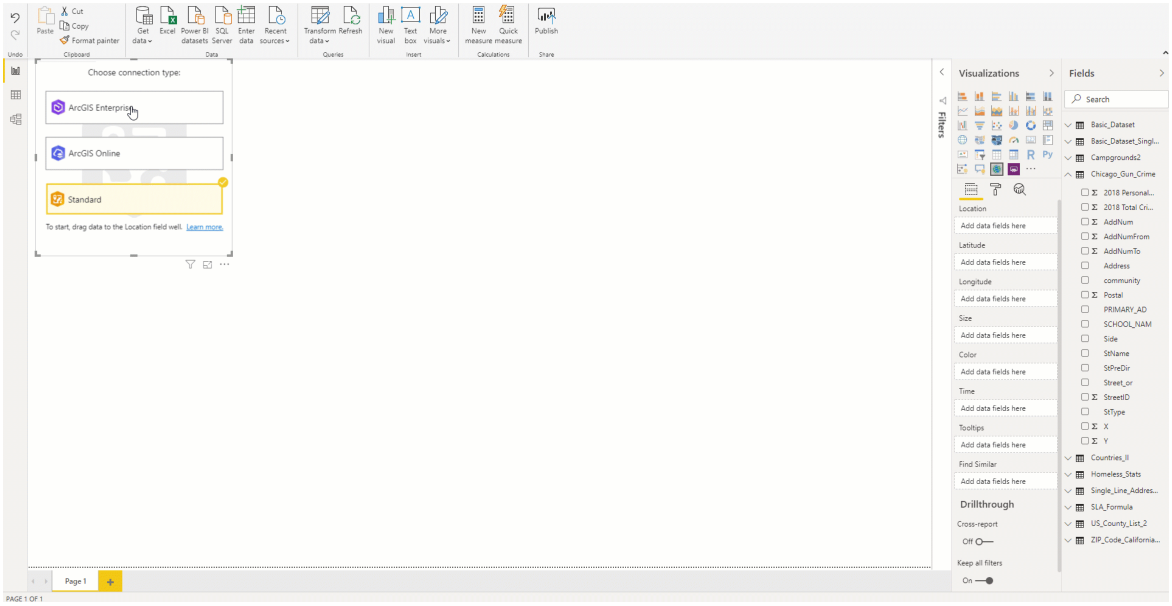
The ArcGIS Enterprise and ArcGIS Online options are for users who have the premium application subscription. They provide all the capabilities of Standard and Extra capabilities, including additional geocoding, technical support and access to mapping reference layers, and more. The Standard option is free and provides basic mapping capabilities.



Capability	Standard	Premium (ArcGIS Online/Enterprise)
Basemaps		
• Esri-curated and custom basemaps	Limited subset	✓
Geocoding		
• 3500 points per map. 100,000 points per month	✓	
• 10,000 points per map. No monthly limits		✓
Reference Layers		
• Add layers of US demographics (esri-curated content)	Limited subset	✓
• Add layers of global demographics (esri-curated content)		✓
• Add layers created and shared publicly by thousands of global ArcGIS users	✓	✓
• Add layers created within your ArcGIS Online/ Enterprise organization		✓
• Add layers from - The Living Atlas - esri-curated content		✓
• Add multiple reference layers		✓
Infographics		
• US	Limited subset	✓
• Global		✓
Tech Support		
• Microsoft	✓	✓
• Esri		✓

ENTERPRISE SUPPORT IS NOW GENERALLY AVAILABLE

In October 2019, ArcGIS Enterprise support was released as a public Preview. In this release, Enterprise support has now become Generally Available. This means that all ArcGIS Enterprise 10.7.1 and above users may connect to their organisation's Enterprise account and use their secure GIS data in dashboards and reports within Power BI. To get started, you would need the ArcGIS Maps Premium app subscription, and then just connect to the ArcGIS visual within Power BI and start mapping.



Choose connection type:

ArcGIS Enterprise

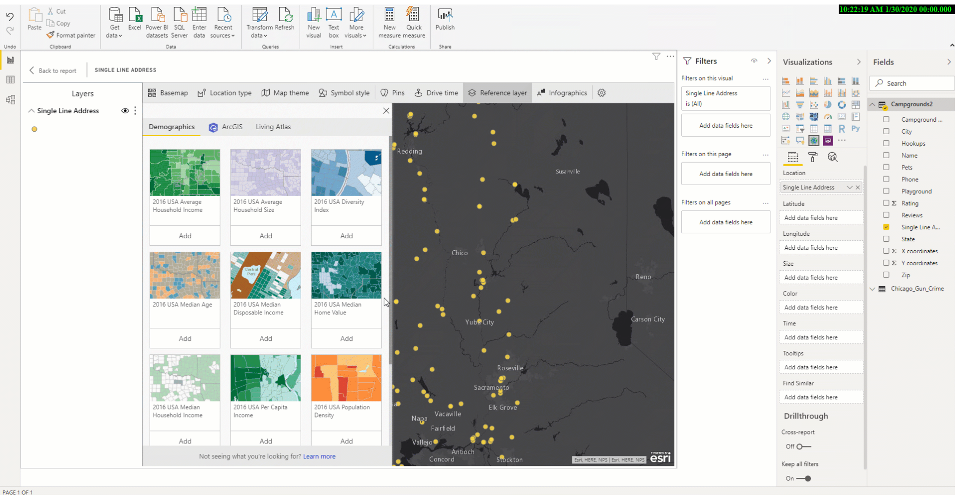
ArcGIS Online

Standard

To start, drag data to the Location field well. Learn more.

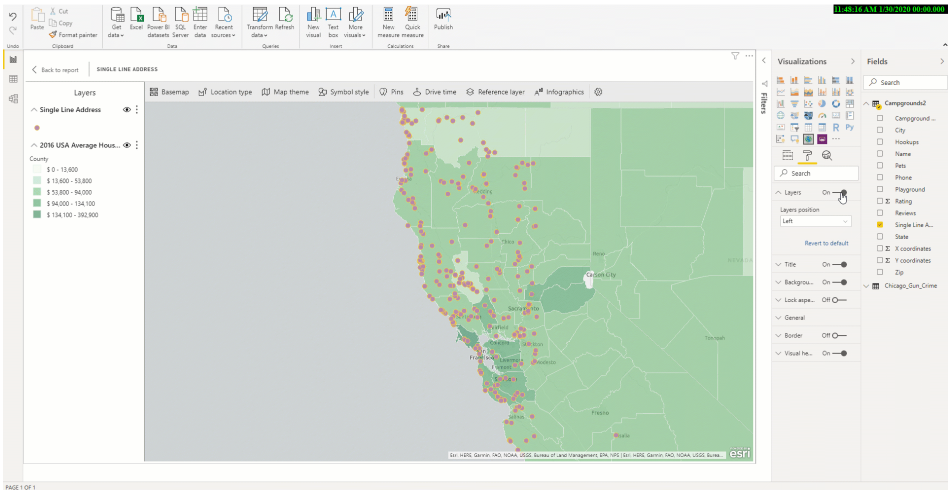
SUPPORT FOR MULTIPLE REFERENCE LAYERS

All Premium app users can now add multiple reference layers to a single map visualisation within Power BI. A reference layer is information represented on a map. It adds context to your operational business data. For example, let's say you have mapped your store locations in Power BI. You can now overlay it against reference layers such as income, age, competitor locations or other demographics to gain valuable insights. You can add data and layers that are published and shared online by the ArcGIS community as well as layers from your ArcGIS Online or ArcGIS Enterprise organisation.


10:22:19 AM 1/30/2020 00:00.000
Back to report
SINGLE LINE ADDRESS
Layers
Single Line Address
Basemap
Location type
Map theme
Symbol style
Pins
Drive time
Reference layer
Infographics
Demographics
ArcGIS
Living Atlas
2016 USA Average Household Income
2016 USA Average Household Size
2016 USA Diversity Index
2016 USA Median Age
2016 USA Median Disposable Income
2016 USA Median Home Value
2016 USA Median Household Income
2016 USA Per Capita Income
2016 USA Population Density
Add
Not seeing what you're looking for? Learn more
Filters
Visualizations
Fields
Campgrounds2
Chicago_Gun_Crime
PAGE 1 OF 1

NEW TABLE OF CONTENTS

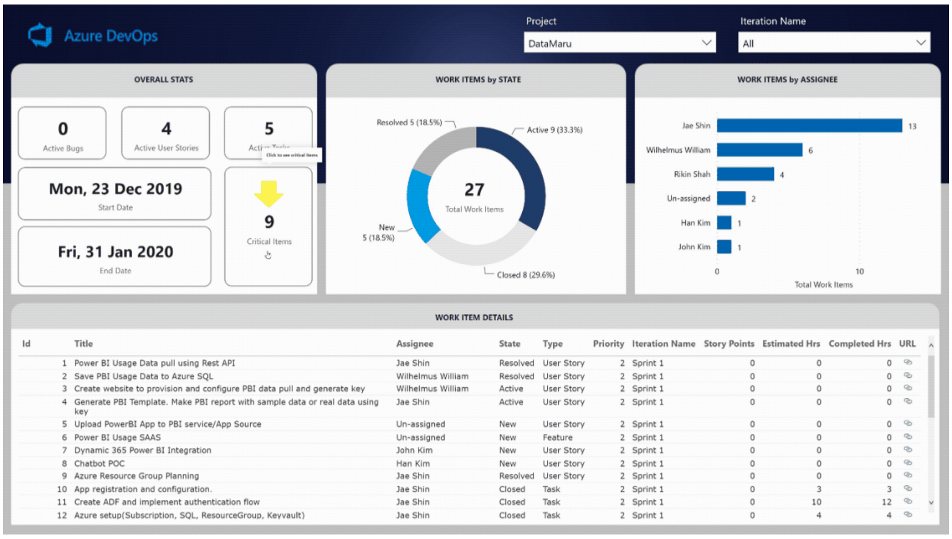
There is also a new table of contents that will help all ArcGIS Maps for Power BI users (Free and Premium) better visualise their data on a map. Now, when you drag data to a location field well and see it on a map, you can also see a table of contents that lists all the layers on the map and shows the features represented by the layers. This allows your report viewers to quickly understand the data that they are seeing.



Azure DevOps dashboard by Data Maru

Azure DevOps dashboard by Data Maru will help stakeholders to assess iteration health and get insight into development details. Measurements will help evaluate:

- Current and historic iteration status (number of user stories and bugs, and breakdown of all tasks)
- Holistic view of current iteration (including current progress, critical items, and outstanding bugs)
- Work item by state
- Work item assignment breakdown by user.



WORK ITEM DETAILS

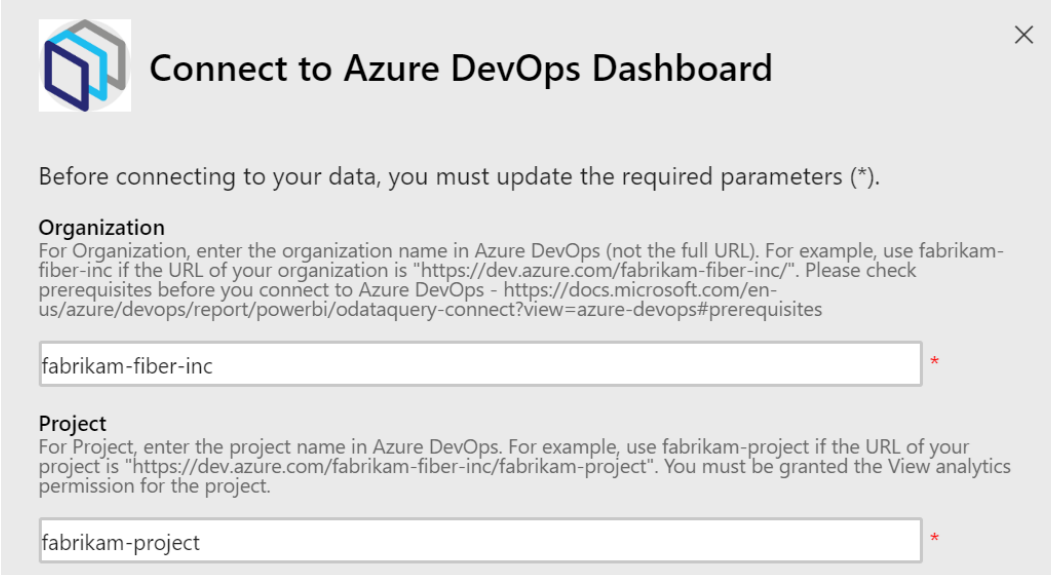
Id	Title	Assignee	State	Type	Priority	Iteration Name	Story Points	Estimated Hrs	Completed Hrs	URL
1	Power BI Usage Data pull using Rest API	Jae Shin	Resolved	User Story	2	Sprint 1	0	0	0	🔗
2	Save PBI Usage Data to Azure SQL	Wilhelmus William	Resolved	User Story	2	Sprint 1	0	0	0	🔗
3	Create website to provision and configure PBI data pull and generate key	Wilhelmus William	Active	User Story	2	Sprint 1	0	0	0	🔗
4	Generate PBI Template. Make PBI report with sample data or real data using key	Jae Shin	Active	User Story	2	Sprint 1	0	0	0	🔗
5	Upload PowerBI App to PBI service/App Source	Un-assigned	New	User Story	2	Sprint 1	0	0	0	🔗
6	Power BI Usage SAAS	Un-assigned	New	Feature	2	Sprint 1	0	0	0	🔗
7	Dynamic 365 Power BI Integration	John Kim	New	User Story	2	Sprint 1	0	0	0	🔗
8	Chatbot POC	Han Kim	New	User Story	2	Sprint 1	0	0	0	🔗
9	Azure Resource Group Planning	Jae Shin	Resolved	User Story	2	Sprint 1	0	0	0	🔗
10	App registration and configuration.	Jae Shin	Closed	Task	2	Sprint 1	0	3	3	🔗
11	Create ADF and implement authentication flow	Jae Shin	Closed	Task	2	Sprint 1	0	10	12	🔗
12	Azure setup(Subscription, SQL, ResourceGroup, Keyvault)	Jae Shin	Closed	Task	2	Sprint 1	0	4	4	🔗

With the details gained above, stakeholders will be able to understand iteration details and examine the data from many different angles. They can easily compare multiple iterations and combine details for the whole. To use this template application:

- you must be a member of a project in Azure DevOps
- set the 'View Analytics' permission to Allow
- enable the analytics views preview feature either for individual users or for the organisation
- boards must be enabled. If it is disabled, analytics views won't be displayed
- the process of your project should be the agile process only.

To connect to your data:

- Click Connect
- Enter your Organization name and your Project name in Azure DevOps



Connect to Azure DevOps Dashboard

Before connecting to your data, you must update the required parameters (*).

Organization
For Organization, enter the organization name in Azure DevOps (not the full URL). For example, use fabrikam-fiber-inc if the URL of your organization is "https://dev.azure.com/fabrikam-fiber-inc/". Please check prerequisites before you connect to Azure DevOps - https://docs.microsoft.com/en-us/azure/devops/report/powerbi/odataquery-connect?view=azure-devops#prerequisites

fabrikam-fiber-inc *

Project
For Project, enter the project name in Azure DevOps. For example, use fabrikam-project if the URL of your project is "https://dev.azure.com/fabrikam-fiber-inc/fabrikam-project". You must be granted the View analytics permission for the project.

fabrikam-project *

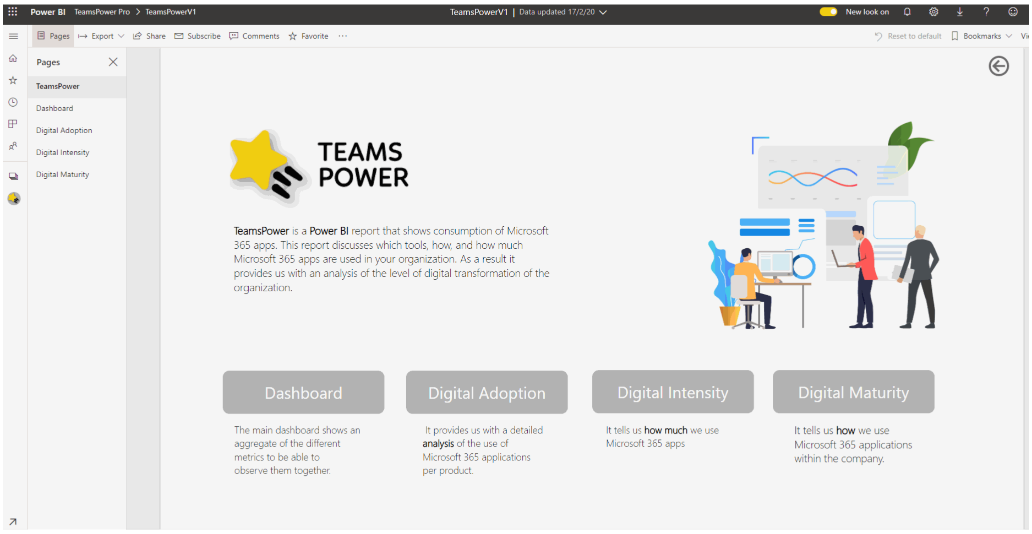
- Choose OAuth2 for Authentication method and sign in with an account that has 'View Analytics' permission for the project.

TeamsPower by Encamina

This report discusses which, how, and how much, the M365 tools are used in your organisation. As a result, it provides an analysis of the company's digital transformation level. It offers the user a traffic light-style view that places the company on one level or another, based on configurable thresholds. If the company is above the set threshold, the indicator will have a green colour, if it is in the average, yellow, and if it is below, it will appear in red.

TeamsPower analyses three key indicators of digital transformation:

- **digital adoption:** provides a detailed analysis of the use of Microsoft 365 applications
- **digital intensity:** indicates how much use is made of Microsoft 365 applications, based on the frequency of user activity
- **digital maturity:** shows how Microsoft 365 apps are being used within the company.



TeamsPower is a Power BI report that shows consumption of Microsoft 365 apps. This report discusses which tools, how, and how much Microsoft 365 apps are used in your organization. As a result it provides us with an analysis of the level of digital transformation of the organization.

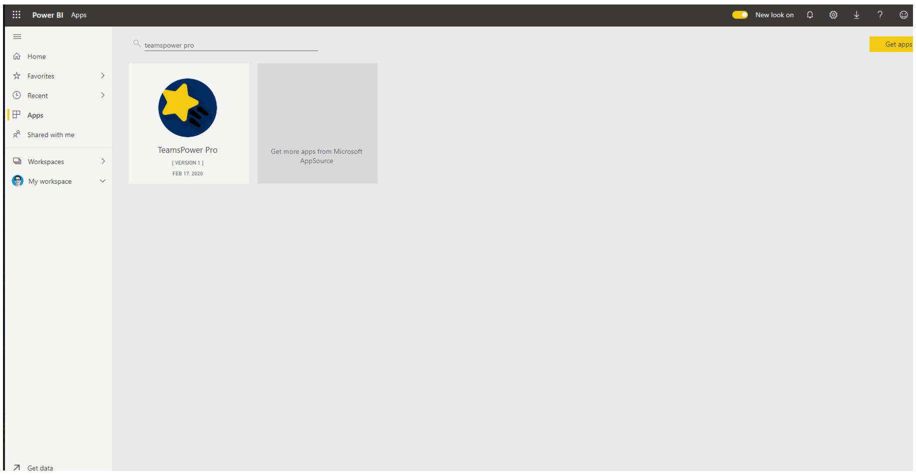
Dashboard
The main dashboard shows an aggregate of the different metrics to be able to observe them together.

Digital Adoption
It provides us with a detailed analysis of the use of Microsoft 365 applications per product.

Digital Intensity
It tells us how much we use Microsoft 365 apps

Digital Maturity
It tells us how we use Microsoft 365 applications within the company.

Once the application is installed, your data should be connected to TeamsPower by using your tenant ID and choosing a scenario to modify the target thresholds set in the application. After you connect your company data, the application will display your own information.

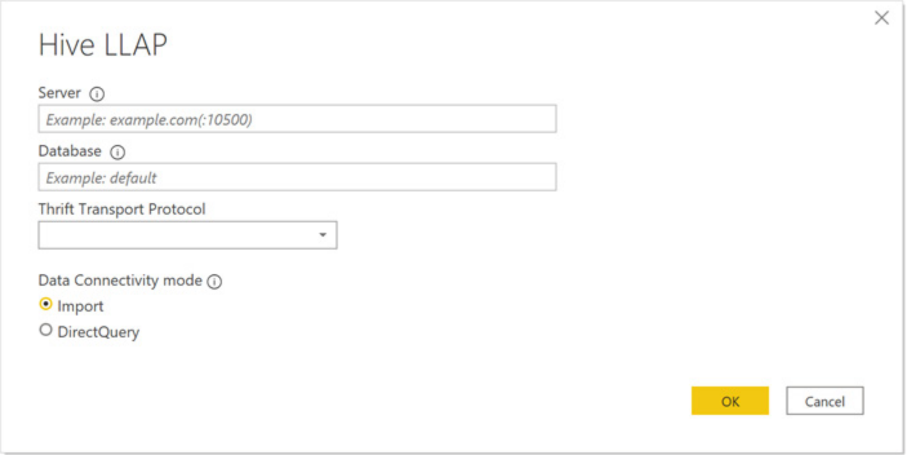


Query diagnostics is now Generally Available

Query diagnostics is now Generally Available. This update has also added the capability to diagnose single steps and improved the experience of using query diagnostics.

Hive LLAP connector

This connector provides both Import and Direct Query capabilities, and the ability to specify Thrift Transport Protocol as 'Standard' or 'HTTP'. Clear as mud, yes?



Hive LLAP

Server ⓘ

Example: example.com(:10500)

Database ⓘ

Example: default

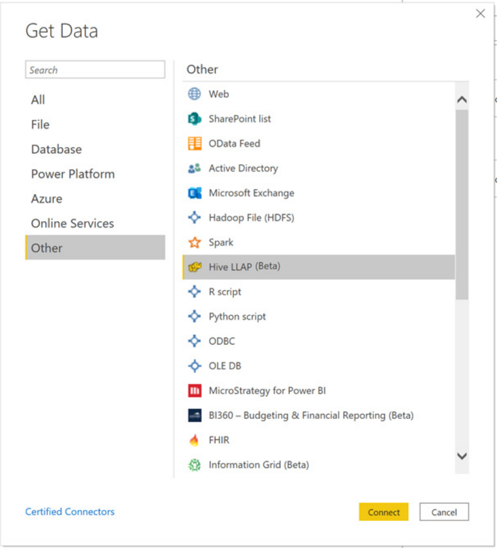
Thrift Transport Protocol

Data Connectivity mode ⓘ

● Import

○ DirectQuery

This connector can be found in the Other category of the 'Get Data' dialog.



Cognite connector

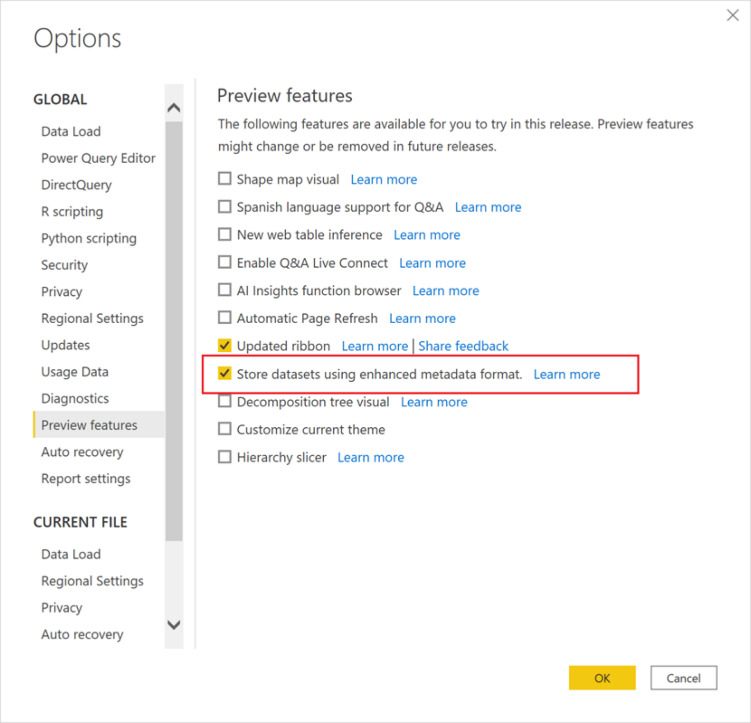
This update sees the public Preview of the Cognite connector. The Cognite Power BI connector enables data consumers to read, analyse and present data from Cognite Data Fusion (CDF). CDF is a software package used to manage massive quantities of industrial data. Users can easily access data such as assets, data points, events, files and time series, and then analyse and aggregate it. This connector can be found in the Other category of the Get Data dialog.

Enhanced dataset metadata (Preview)

Power BI Desktop creates dataset metadata in PBIX and PBIT files. Until now, the metadata was stored in a format very specific to Power BI Desktop. The enhanced dataset metadata Preview feature uses a standard format closer to that used for Analysis Services tabular models, based on the Tabular Object Model. In the coming weeks, this will be a foundational feature because it will enable capabilities such as:

- XMLA read / write for management of Power BI datasets
- the migration of Analysis Services workloads to Power BI to benefit from next-generation features.

You can turn this feature on in the Preview features section of the Options dialog:



Options

Preview features

The following features are available for you to try in this release. Preview features might change or be removed in future releases.

□ Shape map visual Learn more
□ Spanish language support for Q&A Learn more
□ New web table inference Learn more
□ Enable Q&A Live Connect Learn more
□ AI Insights function browser Learn more
□ Automatic Page Refresh Learn more
☑ Updated ribbon Learn more | Share feedback
☑ Store datasets using enhanced metadata format. Learn more
□ Decomposition tree visual Learn more
□ Customize current theme
□ Hierarchy slicer Learn more

Using default system credentials for web proxy

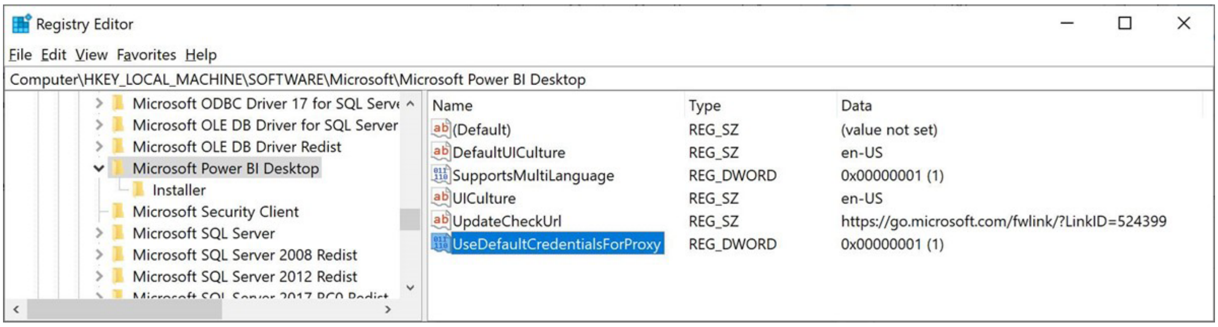
Previously, web requests issued by Power BI Desktop do not use any web proxy credentials, so if you were using a proxy server, Desktop may not be able to make web requests. With this release, system or network administrators may allow users to use default system credentials for web proxy authentication.

The administrators can create a **UseDefaultCredentialsForProxy** registry keys to enable it, under:

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Microsoft Power BI Desktop]

or

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Power BI Desktop].



Registry Editor
File Edit View Favorites Help

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Power BI Desktop

Microsoft ODBC Driver 17 for SQL Serv
Microsoft OLE DB Driver for SQL Server
Microsoft OLE DB Driver Redist
Microsoft Power BI Desktop
Installer
Microsoft Security Client
Microsoft SQL Server
Microsoft SQL Server 2008 Redist
Microsoft SQL Server 2012 Redist
Microsoft SQL Server 2017 RC0 Redist

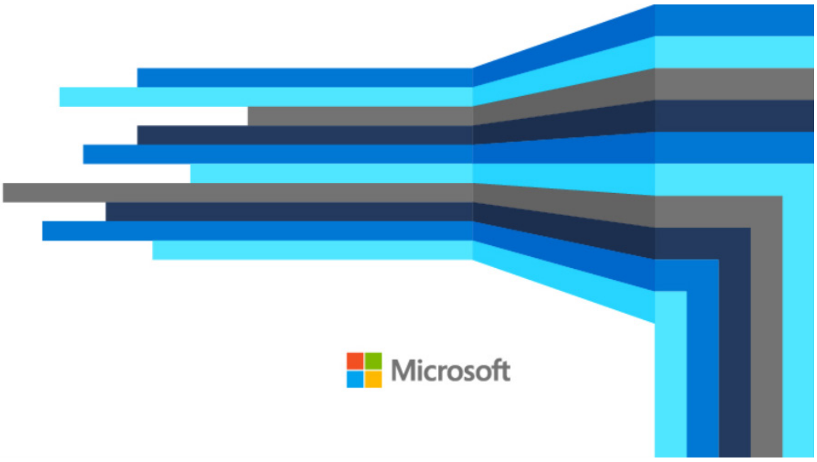
Name	Type	Data
(Default)	REG_SZ	(value not set)
DefaultUICulture	REG_SZ	en-US
SupportsMultiLanguage	REG_DWORD	0x00000001 (1)
UICulture	REG_SZ	en-US
UpdateCheckUrl	REG_SZ	https://go.microsoft.com/fwlink/?LinkID=524399
UseDefaultCredentialsForProxy	REG_DWORD	0x00000001 (1)

Then, the proxy settings, as you would see defined in Internet Explorer, will get used when Desktop makes web requests as well. As with any change to proxy or credential settings, there are security implications to this, so be sure that your administrators have configured these proxies correctly before turning on this feature!

MVP Summit

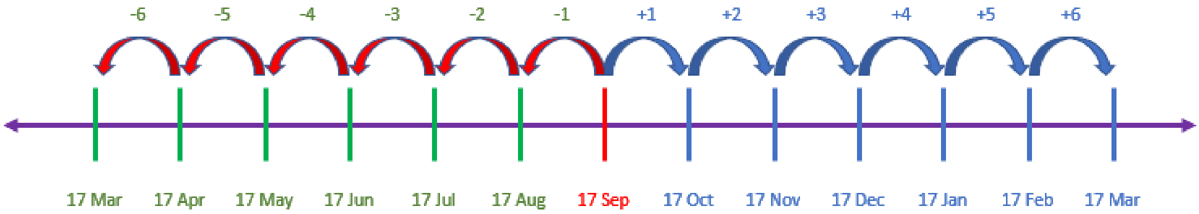
The MVP Summit **DID** go ahead – not with a physical, but a virtual presence – as presenters and attendees alike made use of the interweb thingy, Teams, telephones *et al*. There is some exciting new stuff on the way for Excel and Power BI, and as it becomes public knowledge, we'll bring you the announcements, as and when they occur.

Watch this space.



The A to Z of Excel Functions: EDATE

This function returns the serial number that represents the date that is the indicated number of months before or after a specified date (the **start_date**). **EDATE** should be used to calculate maturity dates or due dates that fall on the same day of the month as the date of issue / first date. This can be useful for infrastructure analysis too.



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

EDATE(start_date, months)

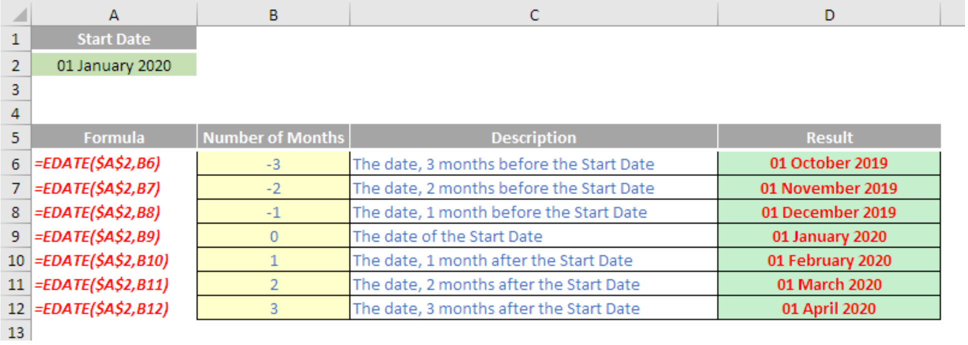
The **EDATE** function has the following arguments:

- **start_date:** this is required and represents the start date. Dates should be entered by using the **DATE** function, or as results of other formulas or functions. For example, consider using **DATE(**2020,5,23**)** for the 23rd day of May, 2020. Problems can occur if dates are entered as text
- **months:** this is also required. This represents the number of months before or after the **start_date**. A positive value for months yields a future date; a negative value yields a past date.

It should be further noted that:

- Excel stores dates as sequential serial numbers so that they can be used in calculations. January 1, 1900 is serial number 1, and July 6, 2009 is serial number 40000 because it is 39,999 days after January 1, 1900
- if **start_date** is not a valid date, **EDATE** returns the *#VALUE!* error value
- if **months** is not an integer, it is truncated
- you will need to change the number format (**CTRL + 1**, 'Format Cells') in order to display a proper date
- February 29, 1900 is recognised as day 60 on the 1900 date system. This date does not exist (years ending in "00" must be divisible by 400 to be a leap year), but this error has been perpetuated to be consistent / compatible with Lotus 1-2-3.

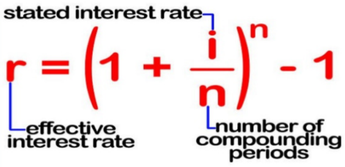
Please see our example below:



	A	B	C	D
1	Start Date			
2	01 January 2020			
3				
4				
5	Formula	Number of Months	Description	Result
6	=EDATE(A2,B6)	-3	The date, 3 months before the Start Date	01 October 2019
7	=EDATE(A2,B7)	-2	The date, 2 months before the Start Date	01 November 2019
8	=EDATE(A2,B8)	-1	The date, 1 month before the Start Date	01 December 2019
9	=EDATE(A2,B9)	0	The date of the Start Date	01 January 2020
10	=EDATE(A2,B10)	1	The date, 1 month after the Start Date	01 February 2020
11	=EDATE(A2,B11)	2	The date, 2 months after the Start Date	01 March 2020
12	=EDATE(A2,B12)	3	The date, 3 months after the Start Date	01 April 2020
13				

The A to Z of Excel Functions: EFFECT

Does Excel sometimes not have the desired **EFFECT**? This function returns the effective annual interest rate, given the nominal annual interest rate and the number of compounding periods per year.


stated interest rate
$r = \left(1 + \frac{i}{n}\right)^n - 1$
effective interest rate
number of compounding periods

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

EFFECT(nominal_rate, npery)

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

- **nominal_rate:** this is required and represents the nominal interest rate
- **npery:** this is also required. This is the number of compounding periods per year.

It should be further noted that:

- **npery** is truncated to an integer
- if either argument is nonnumeric, **EFFECT** returns the *#VALUE!* error value
- if **nominal_rate** ≤ 0 or if **npery** < 1, **EFFECT** returns the *#NUM!* error value.

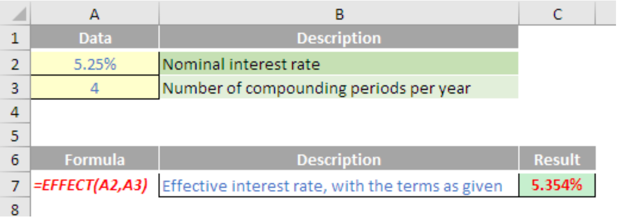
EFFECT is calculated as follows:

$$EFFECT = \left(1 + \frac{Nominal\_rate}{Npery}\right)^{Npery} - 1$$

EFFECT (**nominal_rate, npery**) is related to the function **NOMINAL**(**effect_rate, npery**) through the identity

effective_rate = (1 + (nominal_rate / npery))* npery - 1.

Please see the example below:



	A	B	C
1	Data	Description	
2	5.25%	Nominal interest rate	
3	4	Number of compounding periods per year	
4			
5			
6	Formula	Description	Result
7	=EFFECT(A2,A3)	Effective interest rate, with the terms as given	5.354%
8			

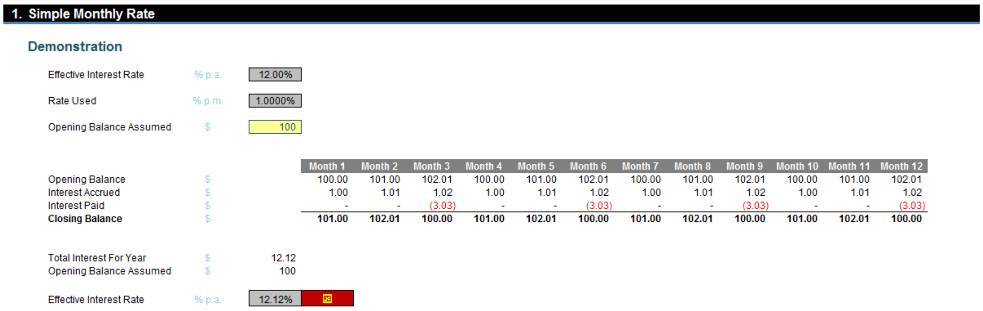
These types of calculations give rise to common errors in financial spreadsheets. One common issue is the inability of many analysts to convert an annual interest rate into a monthly or quarterly rate correctly. Sometimes the error is immaterial; other times, it can cause major issues (*e.g.* for bank forecasts). This isn't so much an Excel issue as a mathematical problem, but it is still relevant to financial modellers and more often than not, it's calculated incorrectly.

The key thing any modeller charged with this exercise must do is read the debt term sheet or deposit account prospectus to see what the effective annual rate is. Accountants talk about nominal interest rates and such like, but the **<u>effective annual rate</u>** is the amount of interest expressed as a percentage of the opening debt or cash balance if interest were to be paid and calculated as per the terms of the underlying document.

It's easier said than done.

Let me demonstrate with an example: consider a loan of $100 where interest is calculated in arrears on a monthly compounding basis paid quarterly at the end of each quarter. The effective rate is determined to be 12%. The question is, what is the appropriate monthly rate be for cashflow calculations?

Who said 1%? That's a common answer, derived as 12% divided by the number of months in a year (12). It sounds good, but if I crunch the numbers I will get the following result:



1. Simple Monthly Rate

Demonstration

Effective Interest Rate	% p.a.	12.00%
Rate Used	% p.m.	1.0000%
Opening Balance Assumed	$	100

		Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
Opening Balance	$	100.00	101.00	102.01	100.00	101.00	102.01	100.00	101.00	102.01	100.00	101.00	102.01
Interest Accrued	$	1.00	1.01	1.02	1.00	1.01	1.02	1.00	1.01	1.02	1.00	1.01	1.02
Interest Paid	$	-	-	(3.03)	-	-	(3.03)	-	-	(3.03)	-	-	(3.03)
Closing Balance	$	**101.00**	**102.01**	**100.00**	**101.00**	**102.01**	**100.00**	**101.00**	**102.01**	**100.00**	**101.00**	**102.01**	**100.00**

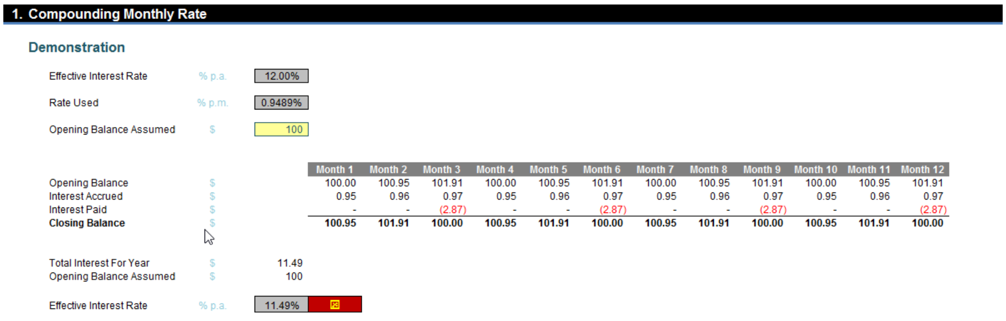
Total Interest For Year	$	12.12
Opening Balance Assumed	$	100
Effective Interest Rate	% p.a.	12.12% ☒

On a $100 balance with interest of 1% monthly paid quarterly, the total interest for the year will be $12.12 – that's an effective annual rate of 12.12% rather than 12% (12.00%). This is not correct, as interest is being rolled up at the end of each of the first two months in the quarter and is attracting further interest, i.e. interest is compounding and has not been taken into account by simply dividing the annual rate by the number of months in a year.

So, what about using the compounding formula instead? Interest can then be calculated as

$$\mathbf{=(1+12\%)^{1/12} - 1}$$

This equates to 0.9489% per month. This will take into account compounding – well, sort of:



1. Compounding Monthly Rate

Demonstration

Effective Interest Rate % p.a. 12.00%

Rate Used % p.m. 0.9489%

Opening Balance Assumed $ 100

		Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
Opening Balance	$	100.00	100.95	101.91	100.00	100.95	101.91	100.00	100.95	101.91	100.00	100.95	101.91
Interest Accrued	$	0.95	0.96	0.97	0.95	0.96	0.97	0.95	0.96	0.97	0.95	0.96	0.97
Interest Paid	$	-	-	(2.87)	-	-	(2.87)	-	-	(2.87)	-	-	(2.87)
Closing Balance	$	**100.95**	**101.91**	**100.00**	**100.95**	**101.91**	**100.00**	**100.95**	**101.91**	**100.00**	**100.95**	**101.91**	**100.00**

Total Interest For Year $ 11.49
Opening Balance Assumed $ 100

Effective Interest Rate % p.a. 11.49% ☒

This hasn't worked either: interest has been under-accrued (that's a good word I've just made up) as only $11.49 has been computed for the year – an effective rate of 11.49%.

Here, the calculation is again too simplistic as it does not take into account that the interest is paid quarterly. This means that compounding throughout the year does not occur, hence the shortfall of 0.51%.

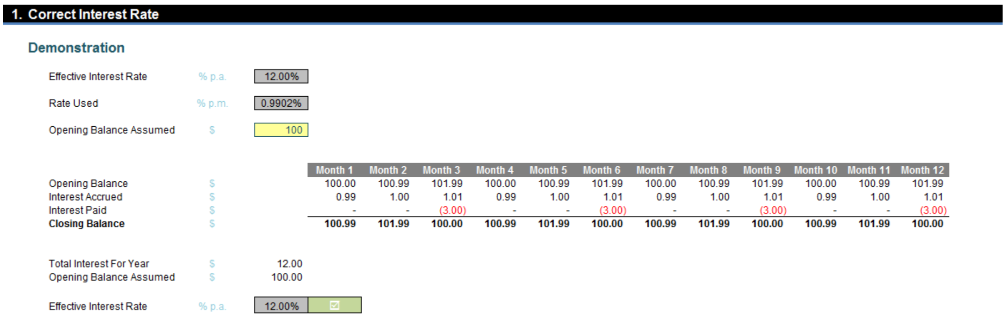
The correct formula is:

$$\mathbf{=(1+(Effective\ Annual\ Interest\ Rate\ x\ Payment\ Frequency\ /\ Months\ in\ Year))^{1/Payment\ Frequency} - 1}$$

If payments are made once every three months then there are four payments (equal to **Months in Year / Payment Frequency** or 12 / 3) each year. At these points, compounding stops. Therefore, the interest rate of 12% per annum is effectively 3% per quarter. The compounding formula can then be applied to the quarterly rate to get the monthly rate accordingly:

$$\mathbf{=(1+3\%)^{1/3} - 1}$$

This equates to 0.9902%. This is correct, *viz.*



1. Correct Interest Rate

Demonstration

Effective Interest Rate % p.a. 12.00%

Rate Used % p.m. 0.9902%

Opening Balance Assumed $ 100

		Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
Opening Balance	$	100.00	100.99	101.99	100.00	100.99	101.99	100.00	100.99	101.99	100.00	100.99	101.99
Interest Accrued	$	0.99	1.00	1.01	0.99	1.00	1.01	0.99	1.00	1.01	0.99	1.00	1.01
Interest Paid	$	-	-	(3.00)	-	-	(3.00)	-	-	(3.00)	-	-	(3.00)
Closing Balance	$	**100.99**	**101.99**	**100.00**	**100.99**	**101.99**	**100.00**	**100.99**	**101.99**	**100.00**	**100.99**	**101.99**	**100.00**

Total Interest For Year $ 12.00
Opening Balance Assumed $ 100.00

Effective Interest Rate % p.a. 12.00% ☑

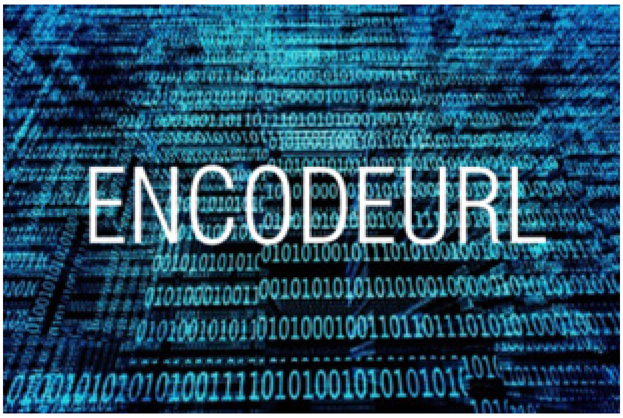
Success!

This is the formula that should be applied to calculate the appropriate monthly interest rate from a quoted effective annual one. The attached Excel file provides examples and a template that may be used.

It should be noted that if interest is paid monthly, the formula above reduces to the simple interest rate method (*i.e.* simply divide the rate by 12). Similarly, if interest is only paid at the end of the year, the compounding formula is correct too. Essentially, the formula explained above is the "halfway house" between the two.

The A to Z of Excel Functions: ENCODEURL

URL encoding is carried to replace the special character, such as ",", "/", "#" and so on, which either has its own special meaning or not a valid character for URL. You may encode your desired URL by using either **ENCODEURL** function in Excel or from websites like encodeurl.com *etc.*



This aforementioned function returns a URL-encoded string.

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

ENCODEURL(text)

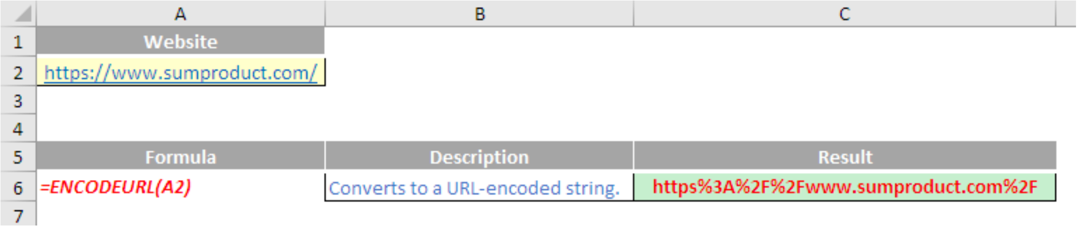
The **ENCODEURL** function has the following arguments:

- **text:** this represents a string to be URL encoded.

It should be noted that:

- this function was introduced in Excel 2013
- the **ENCODEURL** function is not available in Excel Online, and it is not available in Excel 2016 for Mac
- this function may appear in the function gallery in Excel for Mac, but it relies on features of the Windows operating system, so it will not return results on Mac.

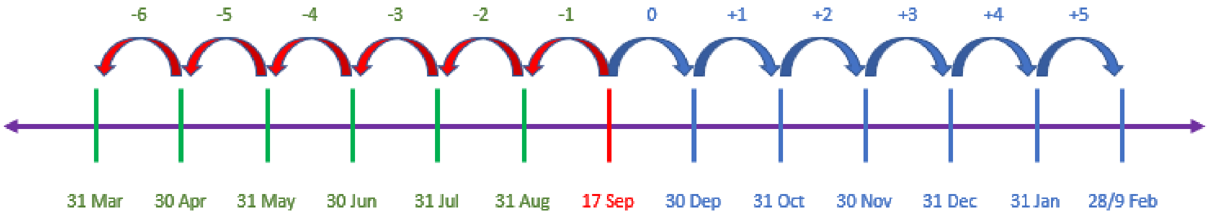
Please see the example below:



	A	B	C
1	Website		
2	https://www.sumproduct.com/		
3			
4			
5	Formula	Description	Result
6	=ENCODEURL(A2)	Converts to a URL-encoded string.	https%3A%2F%2Fwww.sumproduct.com%2F
7			

The A to Z of Excel Functions: EOMONTH

This function returns the serial number for the last day of the month that is the indicated number of months before or after **start_date**. Use **EOMONTH** to calculate calendar dates, maturity dates or due dates that fall on the last day of the month. This is a particularly useful function for creating time series in financial models.



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

EOMONTH(start_date, months)

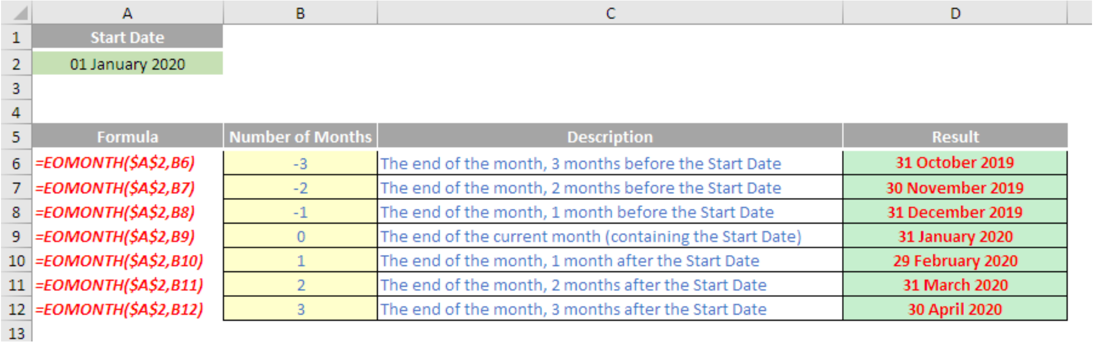
The **EOMONTH** function has the following arguments:

- **start_date:** this is required and represents the start date. Dates should be entered by using the **DATE** function, or as results of other formulas or functions. For example, consider using **DATE(**2020,5,23**)** for the 23rd day of May, 2020. Problems can occur if dates are entered as text
- **months:** this is also required. This represents the number of months before or after the **start_date**. A positive value for months yields a future date; a negative value yields a past date.

It should be further noted that:

- Excel stores dates as sequential serial numbers so that they can be used in calculations. January 1, 1900 is serial number 1, and July 6, 2009 is serial number 40000 because it is 39,999 days after January 1, 1900
- if **start_date** is not a valid date, **EOMONTH** returns the *#VALUE!* error value
- if **months** is not an integer, it is truncated
- **EOMONTH(start_date,0)** produces the end of the current month
- you will need to change the number format (**CTRL + 1,** 'Format Cells') in order to display a proper date
- February 29, 1900 is recognised as day 60 on the 1900 date system. This date does not exist (years ending in "00" must be divisible by 400 to be a leap year), but this error has been perpetuated to be consistent / compatible with Lotus 1-2-3.

Please see our final example for this month below:



	A	B	C	D
1	Start Date			
2	01 January 2020			
3				
4				
5	Formula	Number of Months	Description	Result
6	=EOMONTH(A2,B6)	-3	The end of the month, 3 months before the Start Date	31 October 2019
7	=EOMONTH(A2,B7)	-2	The end of the month, 2 months before the Start Date	30 November 2019
8	=EOMONTH(A2,B8)	-1	The end of the month, 1 month before the Start Date	31 December 2019
9	=EOMONTH(A2,B9)	0	The end of the current month (containing the Start Date)	31 January 2020
10	=EOMONTH(A2,B10)	1	The end of the month, 1 month after the Start Date	29 February 2020
11	=EOMONTH(A2,B11)	2	The end of the month, 2 months after the Start Date	31 March 2020
12	=EOMONTH(A2,B12)	3	The end of the month, 3 months after the Start Date	30 April 2020
13				

More Excel Functions next month…

Upcoming SumProduct Training Courses

In the current environment, many if not all of these courses may not be available as physical events. However, if there is sufficient demand, they will proceed as virtual training sessions. Social distancing will not prevent us from offering our training programs. Furthermore, if you require bespoke training, alternative dates or an "in-house virtual" event, simply drop us a line at contact@sumproduct.com and ask about discounted offerings. We might not be going anywhere, but for all your training needs, we aren't going anywhere!

Location	Course	Date	Duration
Sydney	Excel Tips and Tricks	6 Apr 2020	1 Day
Sydney	Financial Modelling	7 - 8 Apr 2020	2 Days
Melbourne	Excel Tips and Tricks	7 Apr 2020	1 Day
Melbourne	Financial Modelling	8 - 9 Apr 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	11 - 13 May 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	18 - 20 May 2020	3 Days
Melbourne	Excel Tips and Tricks	25 May 2020	1 Day
Melbourne	Financial Modelling	26 - 27 May 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	16 - 18 Jun 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	22 - 24 Jun 2020	3 Days
Melbourne	Excel Tips and Tricks	6 Jul 2020	1 Day
Melbourne	Financial Modelling	8 - 9 Jul 2020	2 Days
Sydney	Excel Tips and Tricks	14 Jul 2020	1 Day
Sydney	Financial Modelling	15 - 16 Jul 2020	2 Days
Melbourne	Power Pivot, Power Query and Power BI	20 - 22 Jul 2020	3 Days
Sydney	Power Pivot, Power Query and Power BI	27 - 29 Jul 2020	3 Days
Sydney	Excel Tips and Tricks	17 Aug 2020	1 Day

Location	Course	Date	Duration
Sydney	Financial Modelling	18 - 19 Aug 2020	2 Days
Melbourne	Excel Tips and Tricks	18 Aug 2020	1 Day
Melbourne	Financial Modelling	19 - 20 Aug 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	24 - 26 Aug 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	7 - 9 Sep 2020	3 Days
Sydney	Excel Tips and Tricks	21 Sep 2020	1 Day
Sydney	Financial Modelling	22 - 23 Sep 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	6 - 8 Oct 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	9 - 11 Nov 2020	3 Days
Sydney	Excel Tips and Tricks	9 Nov 2020	1 Day
Sydney	Financial Modelling	10 - 11 Nov 2020	2 Days
Melbourne	Excel Tips and Tricks	16 Nov 2020	1 Day
Melbourne	Financial Modelling	17 - 18 Nov 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	8 - 10 Dec 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	14 - 16 Dec 2020	3 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we thought we would see what may be achieved with multiplication (*****) and division (**/**):

Keystroke	What it does
CTRL + MULTIPLY (*)	Select current region
CTRL + DIVIDE (/)	Select array
CTRL + BACKSLASH (\)	Select inconsistent cells in row of selection
CTRL + SHIFT + BACKSLASH (\)	Select inconsistent cells in column of selection

There are over 540 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file a 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 http://www.sumproduct.com/training.

Sydney Address: SumProduct Pty Ltd, Suite 803, Level 8, 276 Pitt Street, Sydney NSW 2000
New York Address: SumProduct Pty Ltd, 48 Wall Street, New York, NY, USA 10005
London Address: SumProduct Pty Ltd, Office 7, 3537 Ludgate Hill, London, EC4M 7JN, UK
Melbourne Address: SumProduct Pty Ltd, Level 9, 440 Collins Street, Melbourne, VIC 3000
Registered Address: SumProduct Pty Ltd, Level 6, 468 St Kilda Road, Melbourne, VIC 3004

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