

Sum Product

NEWSLETTER #114 - May 2022

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



We are full on security alerts this month! Excel takes on VBA this month as it is starting to make a more proactive stance to blocking macros from untrusted sources, so we thought we would show you how to re-enable coded routines caught up innocently in this latest security measure. And whilst we're on the subject of looking at ever-present security warnings, we thought we'd revisit the perennial links issue in Excel too!

The regulars are all here in abundance too: there is another Beat the Boredom Challenge, plus our usual articles on Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers and Power BI Updates. There is another tranche of **IMAGINARY A to Z** of Excel Functions this month, and we even look UP some Keyboard Shortcuts.

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

Liam Bastick, Managing Director, SumProduct



Blocked!

Microsoft has recently upped its game with blocking macros. They argue that macro-based malware is on the rise and thus they are blocking macros from more and more external / third party locations. Protected

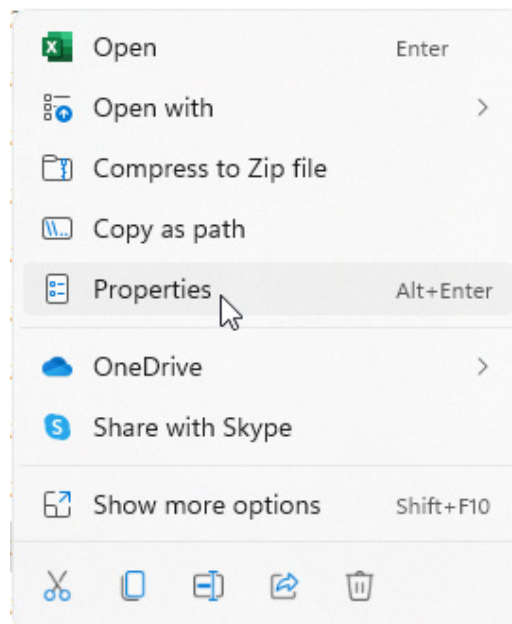
mode seems insufficient as we have noted the following message becoming more and more prevalent when opening files (we have opened time and time again previously!):



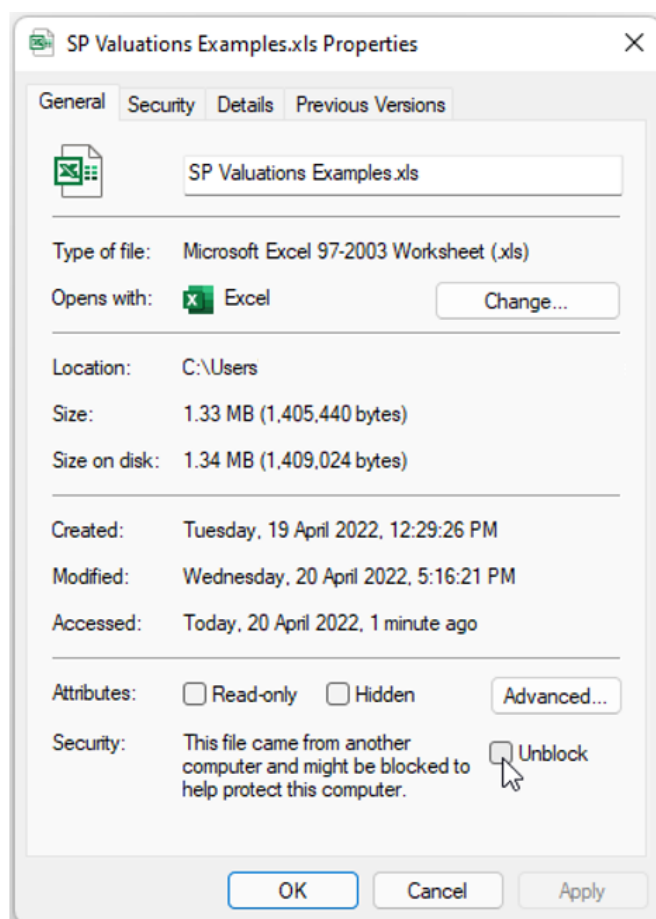
SECURITY RISK Microsoft has blocked macros from running because the source of this file is untrusted.

[Learn More](#)

Of course, everything to avoid digital nasties is to be applauded, but innocent files are being ensnared too. Trying to enable macros in the usual way through the Trust Center (*sic*) will not work here. In this instance, close the file and locate it in File Explorer. Then, right-click on the file:



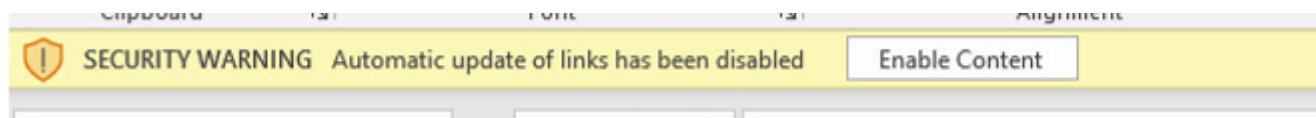
Click on 'Properties' (ALT + ENTER), which will produce the following dialog for the selected file:



At this stage, click on Unblock in the bottom right-hand corner of the General tab. The file may now be opened with the option to enable macros restored.

You're welcome.

Identifying Links – Part 967



Talking of those security warnings, sometimes deliberate and sometimes inadvertent, external links often work their way into our Excel workbooks. To understand what they are, where they are and why they exist, we need to first know which external sources are referenced, because we can't find what we are looking for if we don't know what we are looking

for. I feel a U2 song coming on. That's right, I can't live with or without my links – of which there may be more than One *[please stop – Ed.]*.

Many years ago, I wrote an article that explained how a macro could be employed to list all external links in an Excel workbook:

```
Sub ListExternalLinks()  
Dim ws As Worksheet, TargetWS As Worksheet, SourceWB As Workbook  
If ActiveWorkbook Is Nothing Then Exit Sub  
Application.ScreenUpdating = False  
With ActiveWorkbook  
On Error Resume Next  
Set TargetWS = .Worksheets.Add(Before:=.Worksheets(1))  
If TargetWS Is Nothing Then ' the workbook is protected  
Set SourceWB = ActiveWorkbook
```

```

        Set TargetWS = Workbooks.Add.Worksheets(1)
        SourceWB.Activate
        Set SourceWB = Nothing
    End If
    With TargetWS
        .Range("A1").Formula = "Link No."
        .Range("B1").Formula = "Cell"
        .Range("C1").Formula = "Formula"
        .Range("A1:C1").Font.Bold = True
    End With
    For Each ws In .Worksheets
        If Not ws Is TargetWS Then
            ListLinksInWS ws, TargetWS
        End If
    Next ws
    Set ws = Nothing
End With
With TargetWS
    .Parent.Activate
    .Activate
    .Columns("A:C").AutoFit
    On Error Resume Next
    .Name = "Link List"
    On Error GoTo 0
End With
Set TargetWS = Nothing
Application.ScreenUpdating = True
End Sub

Private Sub ListLinksInWS(ws As Worksheet, TargetWS As Worksheet)
Dim cl As Range, cFormula As String, tRow As Long
    If ws Is Nothing Then Exit Sub
    If TargetWS Is Nothing Then Exit Sub
    Application.StatusBar = "Finding external formula references in " & _
        ws.Name & "... "
    For Each cl In ws.UsedRange
        cFormula = cl.Formula
        If Len(cFormula) > 0 Then
            If Left$(cFormula, 1) = "=" Then
                If InStr(cFormula, "[") > 1 Then
                    With TargetWS
                        tRow = .Range("A" & .Rows.Count).End(xlUp).Row + 1
                        .Range("A" & tRow).Formula = tRow - 1
                        .Range("B" & tRow).Formula = ws.Name & "!" & _
                            cl.Address(False, False, xlA1)
                        .Range("C" & tRow).Formula = "'" & cFormula
                    End With
                End If
            End If
        End If
    Next cl
End Sub

```

```

        End With
    End If
End If
End If
Next cl
Set cl = Nothing
Application.StatusBar = False
End Sub

```

If you are terrified of VBA and hate programming, don't worry – feel free to ignore the above: I have no plans to explain it. I reproduce it for *effect* only.

Whilst working on a client project recently, I came across this wonderful trick that combines old, “outdated” Excel with one of the very latest features in Excel. It is so *simple* and just blows the above solution out of the water. I would love to take the credit for this, but I cannot: long-time Excel Most Valuable Professional (MVP) **Bob Umlas**, please take a bow, because the crux of this trick belongs to you.

Before Visual Basic for Applications (VBA) came to Excel, coding was undertaken using what were known as Excel 4.0 (“xl4”) macro functions. These old xl4 macro functions are still doing the rounds because Microsoft cannot get rid of anything, because the software giant knows some spreadsheets still in use were developed probably before the wheel was.

I have written about **EVALUATE** before, which is a very useful function (it essentially converts text strings into formulae that may be, er, evaluated). For example, consider the following complex spreadsheet:

	A	
1		1
2	+	
3		2

Theoretically,

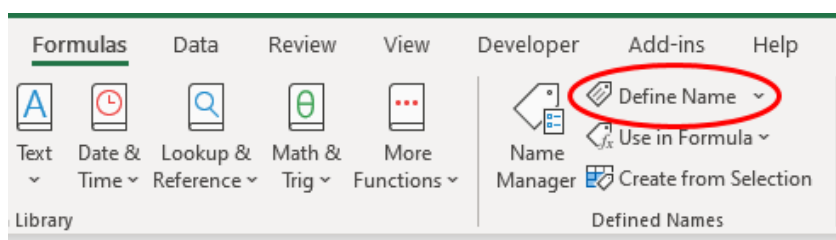
=EVALUATE(A1&A2&A3)

would be **EVALUATE(1+2)** which is three [3]. That's all good, except it doesn't work unless you use it via a range name definition.

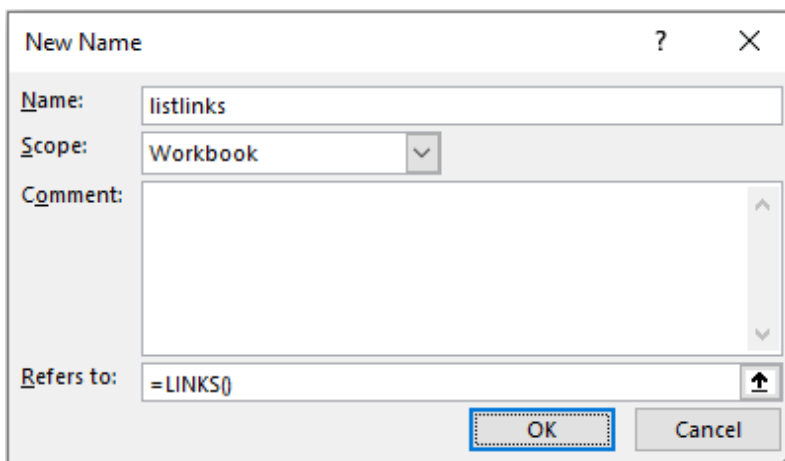
You won't find it in Excel Help (“That function isn't valid.”), but as I say, it is recognised as long as you use it inside an Excel range name. And its sister function, **LINKS**, which recognises external links in an Excel workbook, behaves very similarly.

The process for identifying and listing your links in an Excel workbook is very easy.

First, let's define a range name. I can do this using 'Defined Name' in the Formulas tab of the Ribbon:



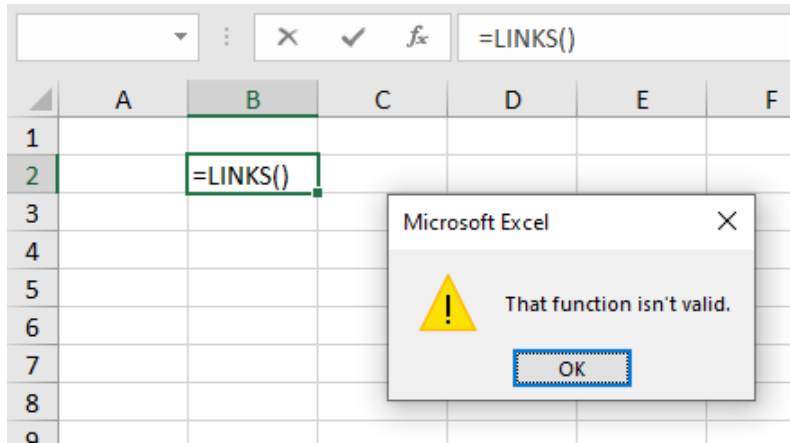
This calls the 'New Name' dialog:



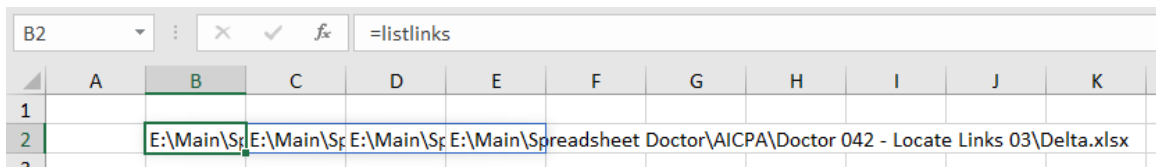
Here, I have created a new range name, called **listlinks**, which refers to the formula

=LINKS()

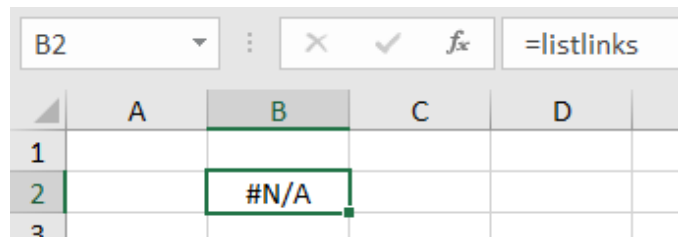
If I were to type this formula straight into a cell, I would get the following (aforementioned) error:



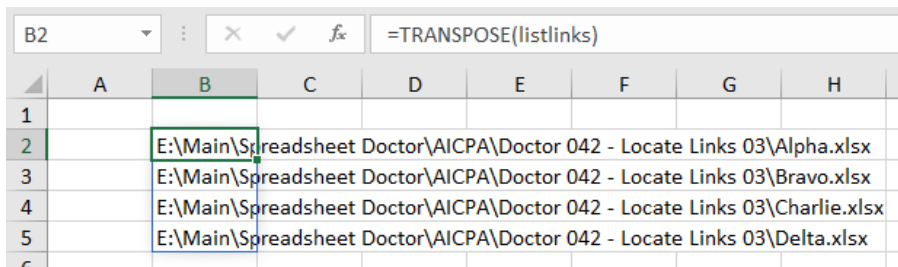
but that's sort of untrue *indirectly*. If instead I were to type in `=listlinks` (*i.e.* my freshly minted range name), I wouldn't get an error if the model contains external links and you have an Office 365 version of Excel:



However, if the model *doesn't* contain links, I would receive a *prima facie* error:



In the first instance, Office 365 has *spilled* the references, *i.e.* it has listed the references in adjacent cells along the same row. Good old Excel: it does like to default on the incorrect choice. To counter this great default "feature", if I were to use **TRANSPOSE**, suddenly things become much more readable:



Voila! All your links presented dynamically.

This is just so much simpler than convoluted VBA code or using third party software. All you need is the ability to spill your results, *i.e.* your version of Excel supports dynamic arrays (presently, this means using an Office 365 version of Excel).

That's all there is to it.

Word to the Wise

Excel 4.0 functions stored in defined names may only be saved in macro-enabled workbooks (.xlsm or .xlsb). If using this feature in conjunction with dynamic arrays, the file will have to be generated using Excel 365 too, so do be aware of these limitations when incorporating this functionality into existing workbooks.

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...

This is based upon a real-life problem, which was not as easy to solve as thought it would be.

Numbers with leading zeroes may sometimes cause problems. In order to find duplicates from a list, you might think of using **FILTER** with **COUNTIF** for example. However, COUNTIF cannot distinguish between numbers

with and without leading zeroes, or “real” numbers and numbers which have been stored as text. Surprised? Try it and see!

As you can see below, the number of occurrences of each number is calculated incorrectly by **COUNTIF**:

List	COUNTIF
12	4
123	5
012	4
0123	5
0123	5
12	4
12	4
123	5
123	5

=COUNTIF(Number[List],Number[@List])

Therefore, this challenge is trickier than it might at first seem.

This month’s challenge is to write a **formula in one cell** using Dynamic Arrays that will spill down to generate a list of duplicates (*i.e.* all numbers that show up more than once) from the number list in the file above. The result should be similar to the list generated on the right (*below*):

List	Solution
12	0123
123	12
012	123
0123	
0123	
12	
12	
123	
123	

As always, there are some requirements:

- the formula needs to be in just one cell (no “helper” cells)
- it should treat these kinds of numbers differently:
 - o with and without leading zeroes
 - o “real” numbers as opposed to numbers stored or formatted as text
- this is a formula challenge – no Power Query / Get & Transform or VBA!

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

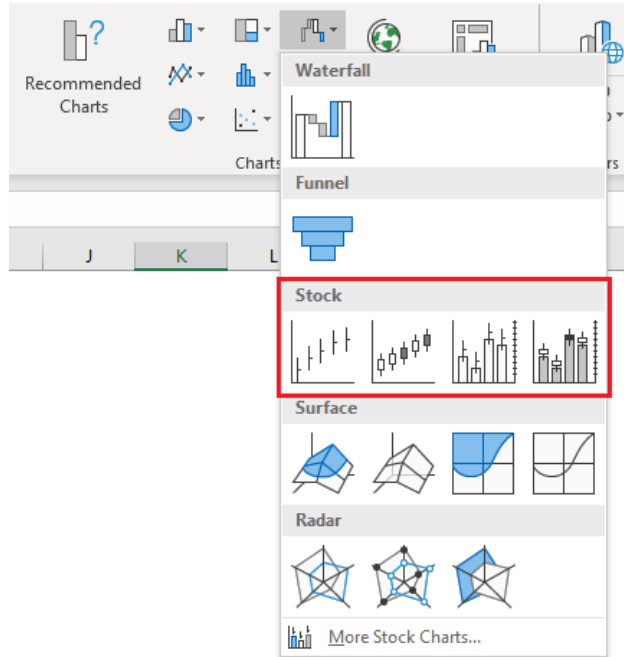
Charts and Dashboards

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

A Stock chart, or Candlestick chart, is a form of Column chart specifically designed to plot movements in stock market commodities (e.g. currencies, securities) over time.

There are four different types of Stock charts available:

1. the first chart type requires three data series, being the high, low and closing prices of the commodity
2. as well as these three series, to use the second Stock Chart, you must also include the opening prices
3. the third chart type requires the same three price series as the first chart plus the volume traded
4. the final Stock Chart type requires all five series - volume traded and the opening, high, low and closing prices.



We're going to use the fourth Stock chart, so let's create a data table which includes the relevant columns:

Commodity Trading Information					
Date	Volume Traded	Opening Price	High Price	Low Price	Closing Price
4-Feb-19	21,434,900	\$ 4.78	\$ 6.36	\$ 3.96	\$ 6.04
5-Feb-19	14,376,500	\$ 6.04	\$ 8.04	\$ 5.75	\$ 7.39
6-Feb-19	13,646,300	\$ 7.39	\$ 8.05	\$ 7.20	\$ 7.38
7-Feb-19	19,707,900	\$ 7.38	\$ 8.18	\$ 6.56	\$ 7.14
8-Feb-19	24,691,800	\$ 7.14	\$ 8.83	\$ 6.21	\$ 8.51
11-Feb-19	18,185,400	\$ 8.51	\$ 9.83	\$ 8.10	\$ 9.46
12-Feb-19	13,807,000	\$ 9.46	\$ 11.17	\$ 8.68	\$ 10.80
13-Feb-19	18,895,500	\$ 10.80	\$ 11.80	\$ 9.53	\$ 10.45
14-Feb-19	23,170,600	\$ 10.45	\$ 10.64	\$ 10.17	\$ 10.47
15-Feb-19	16,845,900	\$ 10.47	\$ 11.23	\$ 10.26	\$ 10.88

Note the order of the columns in the table above. This is very important as Excel is expecting the data to be laid out a certain way in order to properly graph the figures. The date must be in the first column, the volume traded in the next column, the opening prices in the third column, etc.

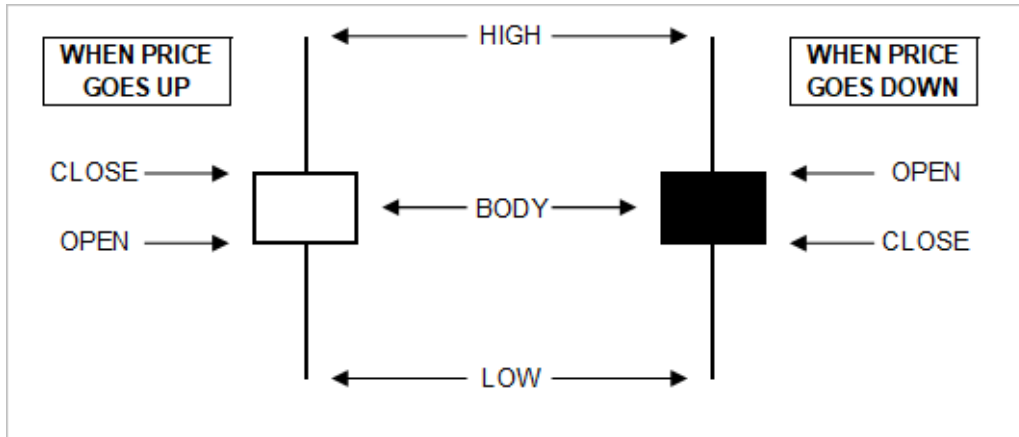
Once the table is ready, highlight the data and create the Stock chart applying the fourth chart type. The initial chart should resemble the following:



So how do you read this chart? There is a fair bit to take in with this chart. There is a gap in the data, lots of horizontal gridlines and several floating black and white boxes with lines going through them.

Let's start with the blue columns. As the legend indicates, these represent the volumes traded. The gap in the data is because the stock market does not trade on weekends, but given this is a chart based on a time series, it needs to have a continuous sequence of dates, regardless

of whether they are provided in the data or not. The multitude of gridlines is purely due to the two vertical axes on the sides of the chart, each displaying gridlines relevant to their numeric scales. The black and white boxes with lines sticking out of them are generated from the four pricing series in my data table. The best way to read this part of the chart is by referring to the following diagram:

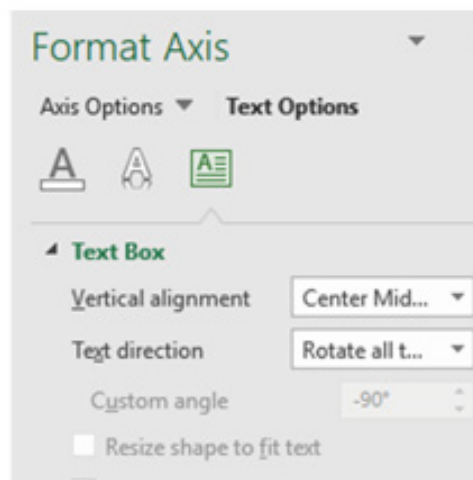
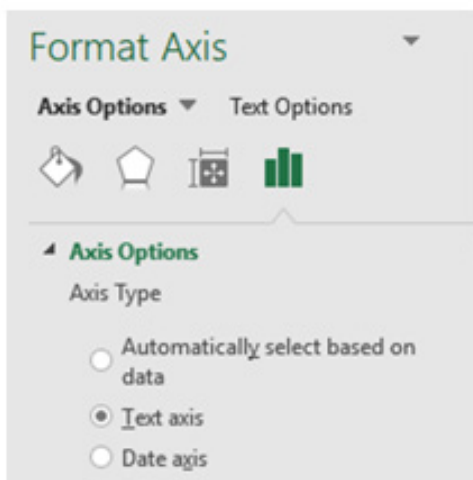


If you look at the data table, on 4 February the opening price was \$4.78 and the closing price was \$6.04, so the price went up that day, resulting in a white box, or body as it is known. The highest price for the day was \$6.36, so only 32 cents higher than the closing price resulting in a short line extending upwards out of the body. The lowest price traded was \$3.96, which is 82 cents below the opening price, so the line extending down from the body is longer. If the price goes down, as occurred on 7 February, the body is represented by a black box.

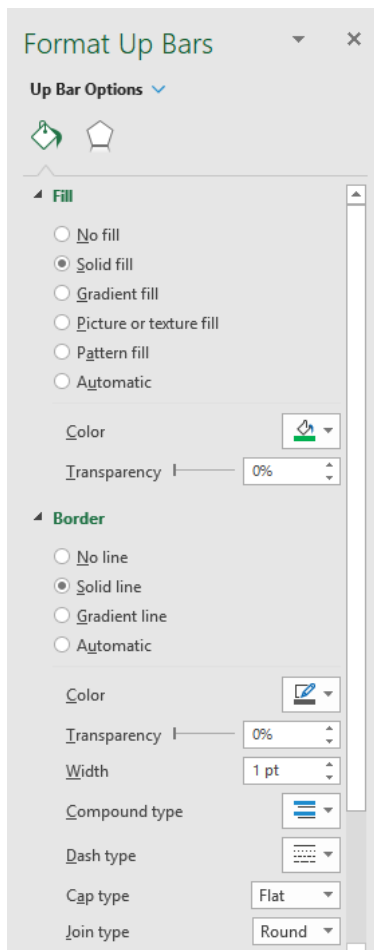
So now that you understand how to read the chart, you might wish to apply some formats to make it "easier on the eyes".

- We will add 'Axes Titles' on both sides by clicking anywhere on the chart, then go to the Design tab on the Ribbon, click on 'Add Chart Element' and choose 'Axis Titles' and 'Primary Vertical'. Click on each title then change the wording to "Volume Traded" on the left and "Price (Open, High, Low, Close)" on the right. While we're changing text, also change the 'Chart Title' to 'Commodity Trading Results'

- With 'Axes Titles' in place, we no longer need the legend so we may click on the legend and then press the **DELETE** key or right-click and choose Delete
- To simplify the gridlines, we need to alter the units in the scales on both axes. If we set the volumes to range from 0 to 30,000,000 in increments of 5,000,000 and set the prices to range from \$0 to \$12 in increments of \$2, then both scales will have six gridlines each, providing one set of gridlines that apply to both axes. To arrange this, click on the left vertical axis, right click, choose 'Format Axis', then under Bounds, set the Minimum to zero and Maximum to 30,000,000 and under Units set the Major to 5,000,000. Next, without closing the 'Format Axis' panel, click on the right vertical axis and set the Minimum to zero, the Maximum to 12 and the Major to two [2]



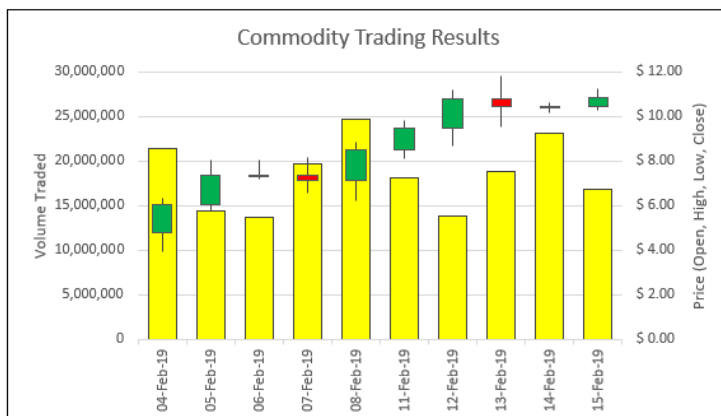
- We also want to remove the weekend dates so the chart can use all the space available and not have this gap. To resolve this, click on the horizontal axis, right-click and choose 'Format Axis'. Under 'Axis Options', set the 'Axis Type' to be 'Text Axis'. If the dates on the horizontal axis are at an angle and we want to straighten them up, click on 'Text Options' and set the 'Text Direction' to be 'Rotate All Text 270'
- Next, it is recognised in the trading industry that the standard colours used to indicate movements in commodity prices is green when the price goes up and red when it goes down. We will put aside potential issues with visual impairment for the moment. Click on one of the white boxes, right click and choose Format Up Bars, and change the fill colour from white to green. While we are in the Formatting panel, go to where it says, 'Up Bar Options' at the top on the panel, click on the drop-down arrow and select 'Down Bars 2' and change the fill colour so the black boxes are now red.



- We would also like to tone down the formatting of the volume traded series, so at the top of the Formatting panel, where it presently states 'High-Low Line Options', use the drop-down arrow and select 'Series "Volume Traded" '. Change the fill to a softer colour and add a black border on the data series
- Also, note that the width of the volume columns and the width of the open-close boxes is the same, which means that if the

volume traded column happens to stop behind an open-close box, we may not be able to work out what the volume was. To fix this, while still in the 'Format Data Series' panel, click on the 'Series Options' icon (it looks like a small Column Chart) and reduce the 'Gap Width' to 75%.

With all the modifications complete, here is the chart we have now:



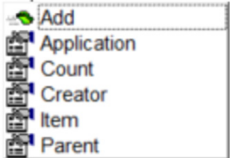
More next month...

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 will be reviewing **ListColumns** and how it applies to **ListRows**.

ListColumns can be really confusing. Using **ListColumns** references ALL the columns, but using **ListColumns(Index)** only references a specific column. Remember in VBA you may press "." and the applicable method / properties that apply to that object come up? This is true with **ListColumns**:

```
Sub Test ()  
  
Dim MyTable As ListObject  
Set MyTable = Range("MyTable").ListObject  
  
MyTable.ListColumns.  
  
End Sub
```



- **Add**: this allows you to add a column. By default it will be added at the end of the table
- **Application**: this returns the name of the application from which the object belongs to (in this case Excel)
- **Count**: returns the number of columns
- **Creator**: returns the name of the author of the object
- **Item**: this returns the column as an object. **ListColumns.Item(Index)** is identical to using **ListColumns(Index)** in terms of referencing a specific column
- **Parent**: this returns the overall owner of the object, in this case, being "MyTable".

All of these properties are identical in **ListRows**: simply interchange column in the above list with row!

Notice how there is no 'Delete' property here? This is done on a row / column indexed basis. This is because invoking 'Delete' would either delete your entire data set (on **ListRows**) or the entire table (on **ListColumns**). If that is indeed what you wish to do, then using **DataBodyRange** or the **ListObject** table itself would be more intuitive and understandable.

More next time.

Power Pivot Principles

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

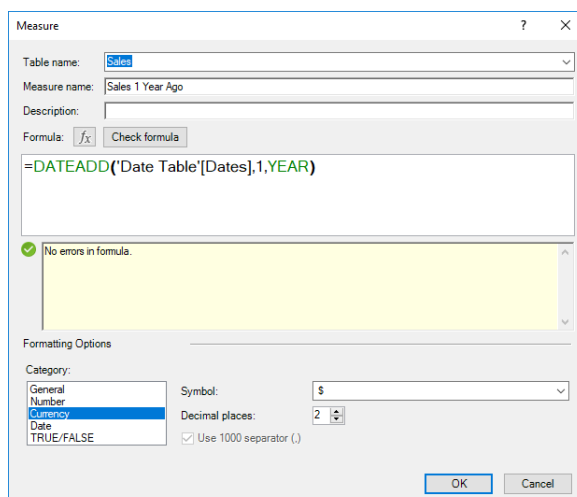
The **DATEADD** function is a time intelligence function just like the **TOTALYTD** and **DATESYTD** functions. They all require a calendar table with strict ascending and contiguous dates with no gaps.

The syntax of this function is:

DATEADD(dates, number_of_intervals, interval)

The **DATEADD** function returns the table of column of dates that is shifted either forward or backwards in time specified by the **number_of_intervals**.

To illustrate, let's create a simple measure with the **DATEADD** function:



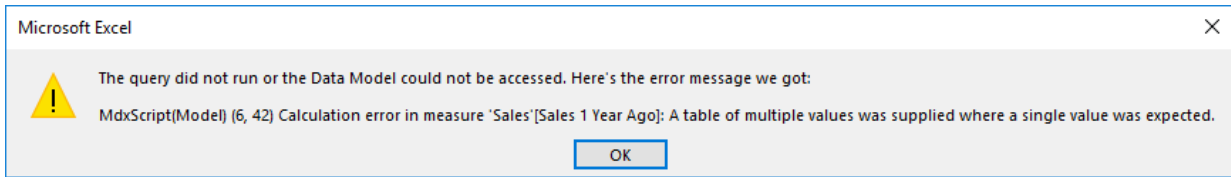
=DATEADD('Date Table'[Dates],1,YEAR)

The first parameter we have to specify is the **dates** argument; in this case, we shall use our contiguous date table **'Date Table'[Dates]**.

The next parameter is the **number_of_intervals**. This can be any integer which Power Pivot will use to add or subtract the from the dates. In our example we use '1'.

The final parameter to define is the **interval**. We can specify the **interval** to be **YEAR, QUARTER, MONTH** or **DAY**. In this case, we are looking for the sales one year ago so we shall use **YEAR**.

Upon checking the formula Power Pivot returns with 'no errors in formula'. However, after clicking 'OK' we receive this error:



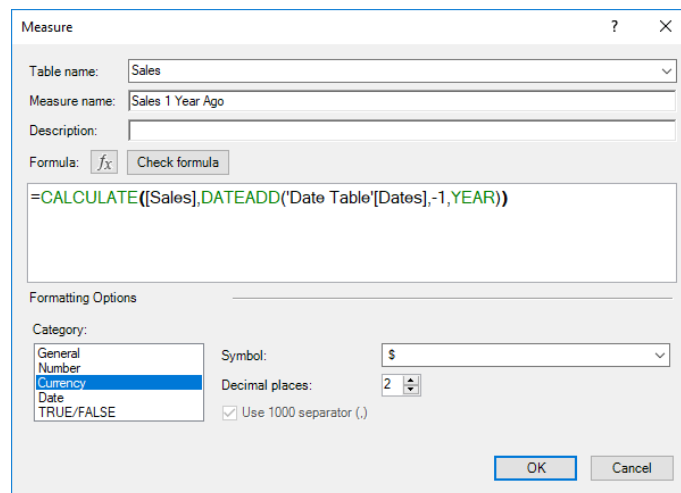
This is because we have not specified what value we want Power Pivot to calculate. Including **'CALCULATE([Sales],'** (with an additional closed bracket at the end) to our formula should suffice:

=CALCULATE([Sales],DATEADD('Date Table'[Dates],1,YEAR))

As long as the dates are contiguous, our measure should work fine now:

Row Labels	Sales Amount	Sales 1 Year Ago
2014		
7	\$4,128.10	\$7,669.80
8	\$4,239.10	\$9,016.25
9	\$4,298.25	\$5,861.15
10	\$4,580.40	\$7,105.85
11	\$5,334.95	\$7,008.80
12	\$6,921.65	\$10,197.60
2015		
1	\$5,231.30	\$6,779.00
2	\$5,097.90	\$7,749.55
3	\$5,667.55	\$7,925.10
4	\$6,063.90	\$9,260.70
5	\$6,029.40	\$10,556.85
6	\$5,752.15	\$10,230.45
7	\$7,669.80	\$42,138.05
8	\$9,016.25	\$115,169.70
9	\$5,861.15	\$117,152.60
10	\$7,105.85	\$124,468.30
11	\$7,008.80	\$125,286.20
12	\$10,197.60	\$157,315.05

Hang on, it seems that our measure is returning with the sales one year in the future not one year ago. Oops. Another slight modification to our measure should address this:



=CALCULATE([Sales],DATEADD('Date Table'[Dates],-1,YEAR))

Note, if we wish for **DATEADD** to return with the sales amounts from the past we have to use a negative integer and a positive integer for sales amounts in the future.

Row Labels	Sales Amount	Sales 1 Year Ago
2014		
7	\$4,128.10	
8	\$4,239.10	
9	\$4,298.25	
10	\$4,580.40	
11	\$5,334.95	
12	\$6,921.65	
2015		
1	\$5,231.30	
2	\$5,097.90	
3	\$5,667.55	
4	\$6,063.90	
5	\$6,029.40	
6	\$5,752.15	
7	\$7,669.80	\$4,128.10
8	\$9,016.25	\$4,239.10
9	\$5,861.15	\$4,298.25
10	\$7,105.85	\$4,580.40
11	\$7,008.80	\$5,334.95
12	\$10,197.60	\$6,921.65

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 **M** function **List.Zip** and its uses.

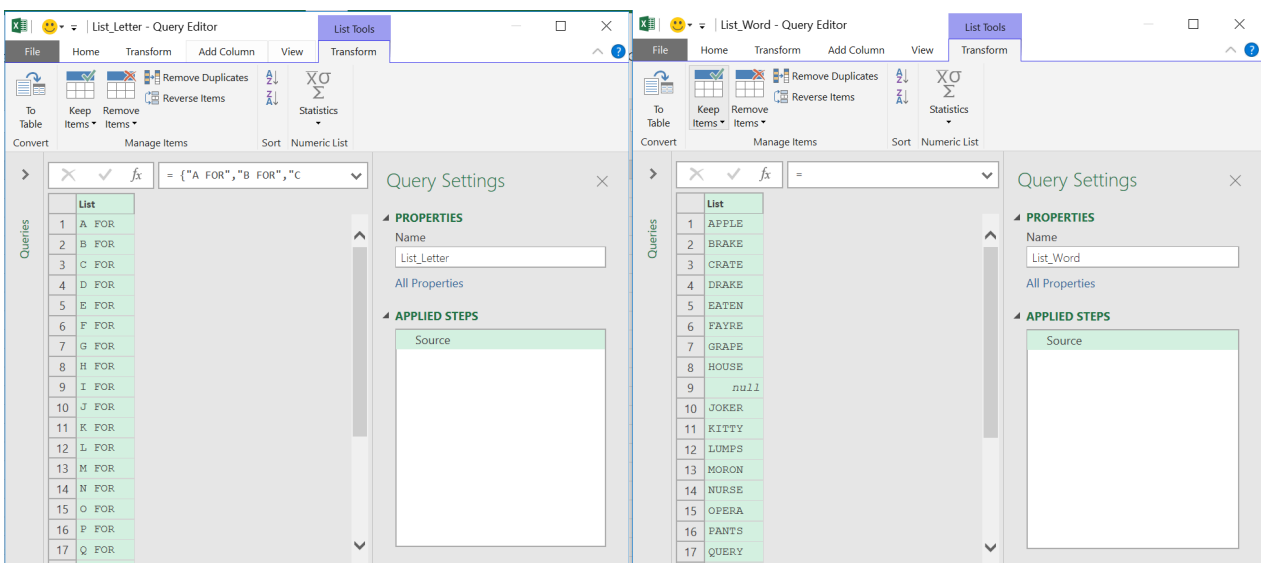
We have been looking at **List** functions in the **M** language for a few newsletters now, and this one reminds us of past problems with *null* values in lists.

The description of the function in the Microsoft help pages is straightforward:

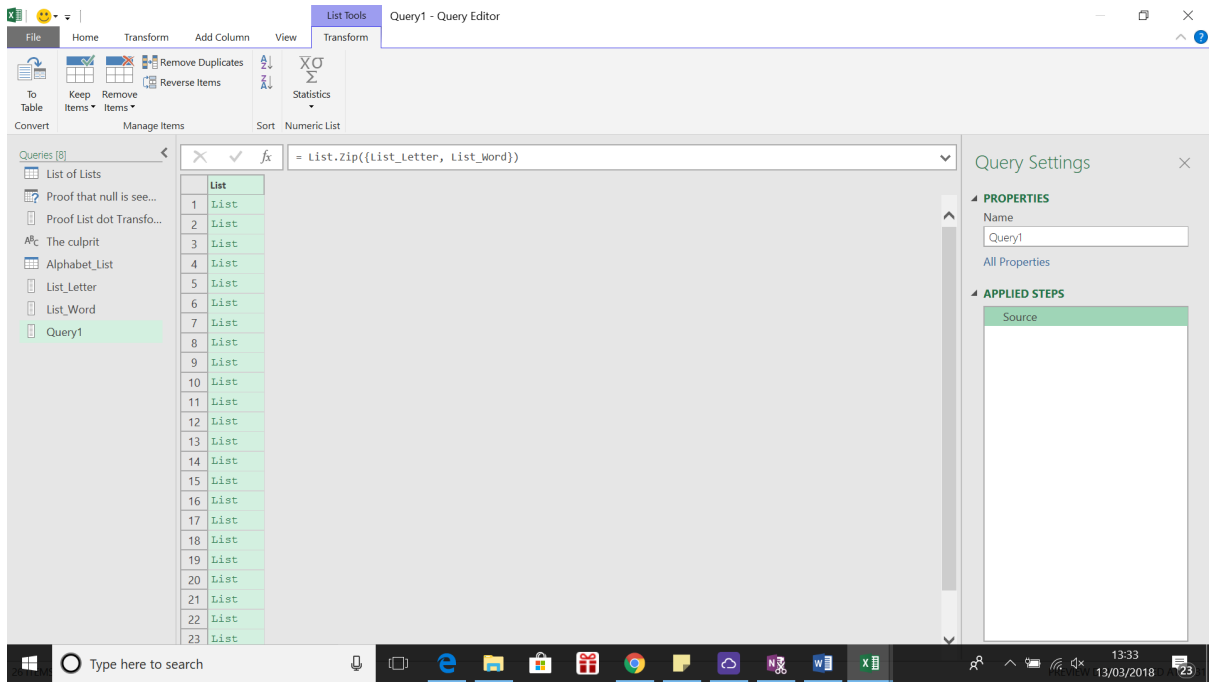
List.Zip(lists as list) as list

This takes a list of lists and returns a list of lists combining items in the same position.

Consider a simple example with two lists below. Let's see if the function will cope with a *null* value.



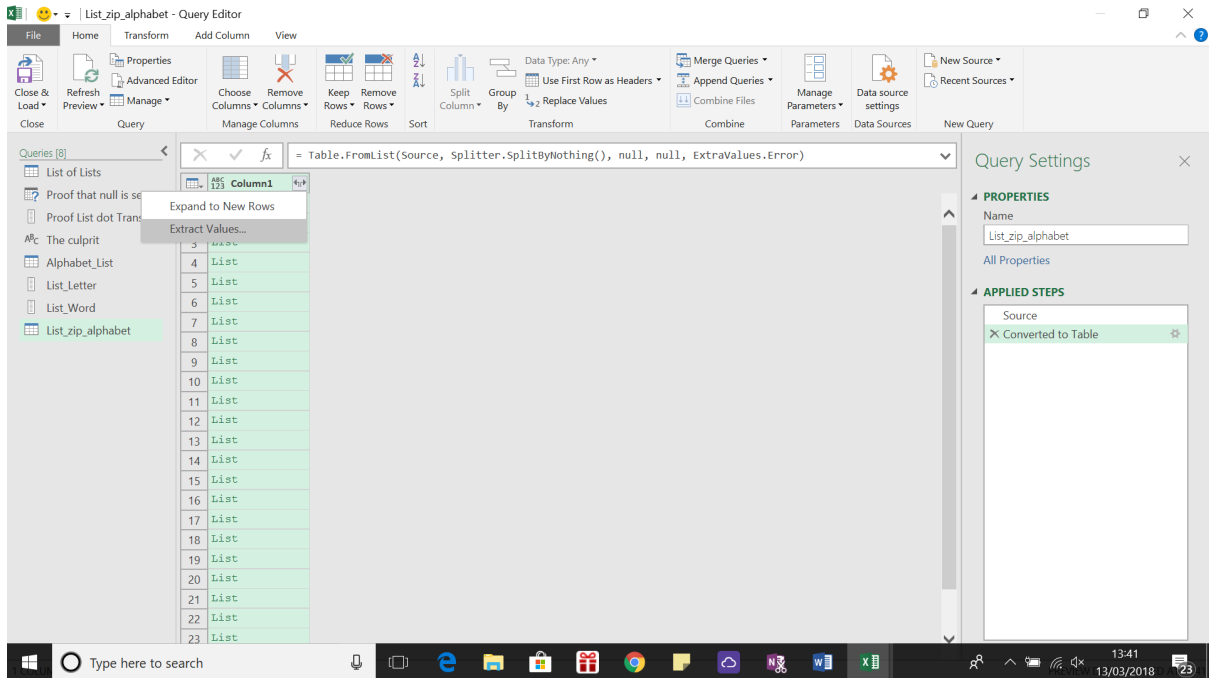
We'll create a new blank query from the 'New Query' option on the 'Get & Transform' section of the 'Data' tab by choosing 'Blank Query' from the dropdown in the 'From Other Sources' section.



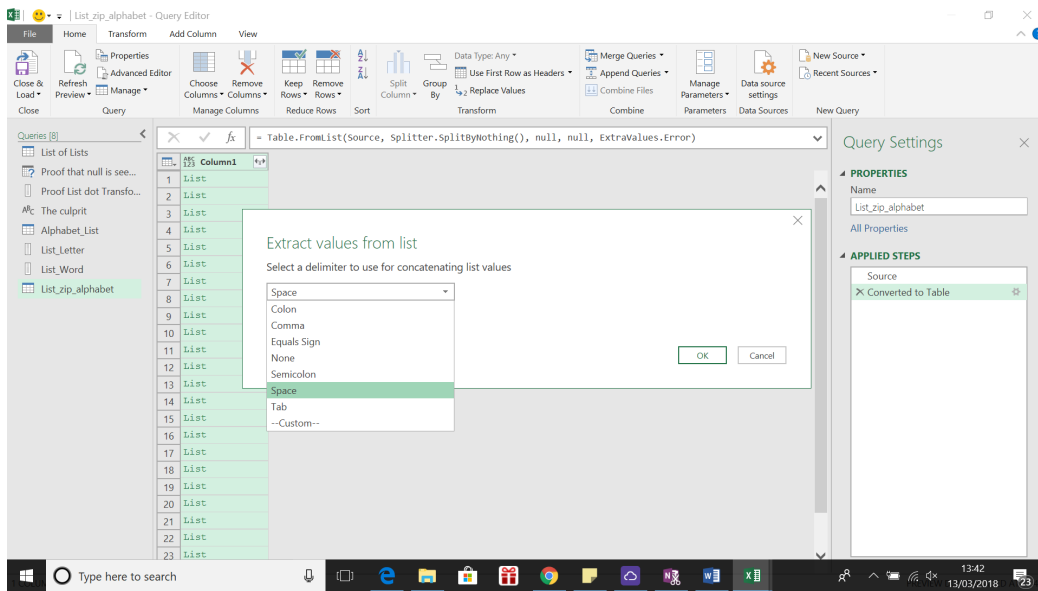
The syntax we have used is:

= List.Zip({List_Letter, List_Word})

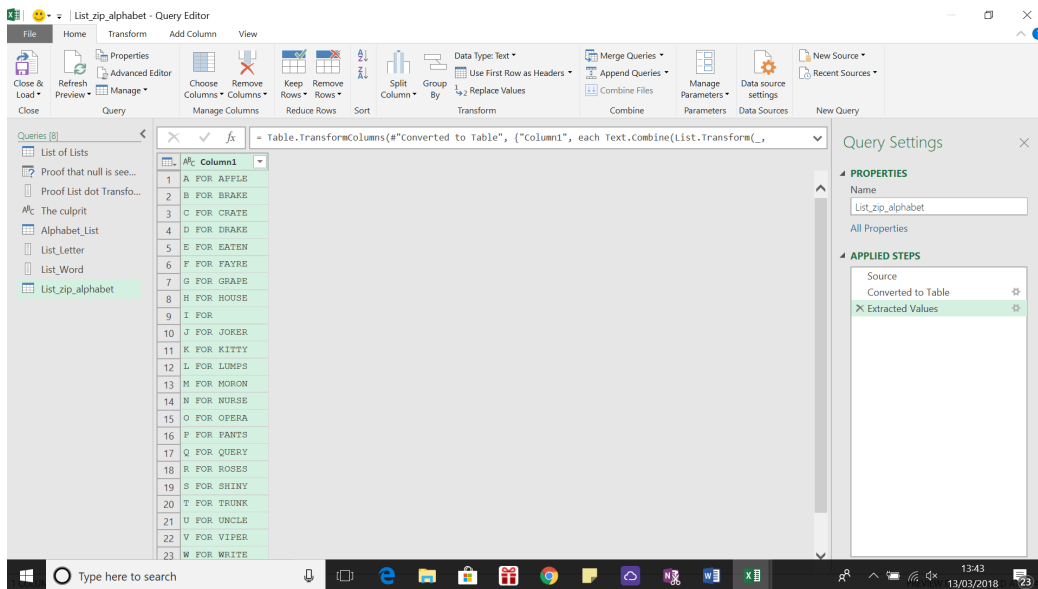
The functionality has created a list of lists. To extract the values, we may convert to a table using the 'To Table' option in the 'Convert' section on the 'List Tools / Transform' tab.



Now, we have a column which we may extract the values from, using the split arrows icon.

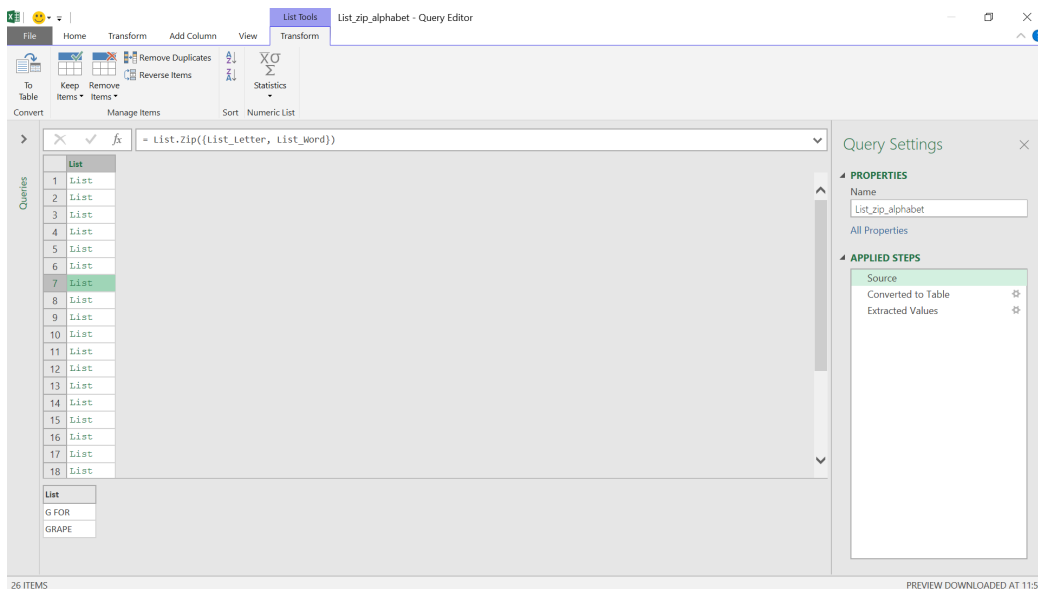


Let's choose to use a space as a delimiter between our values.

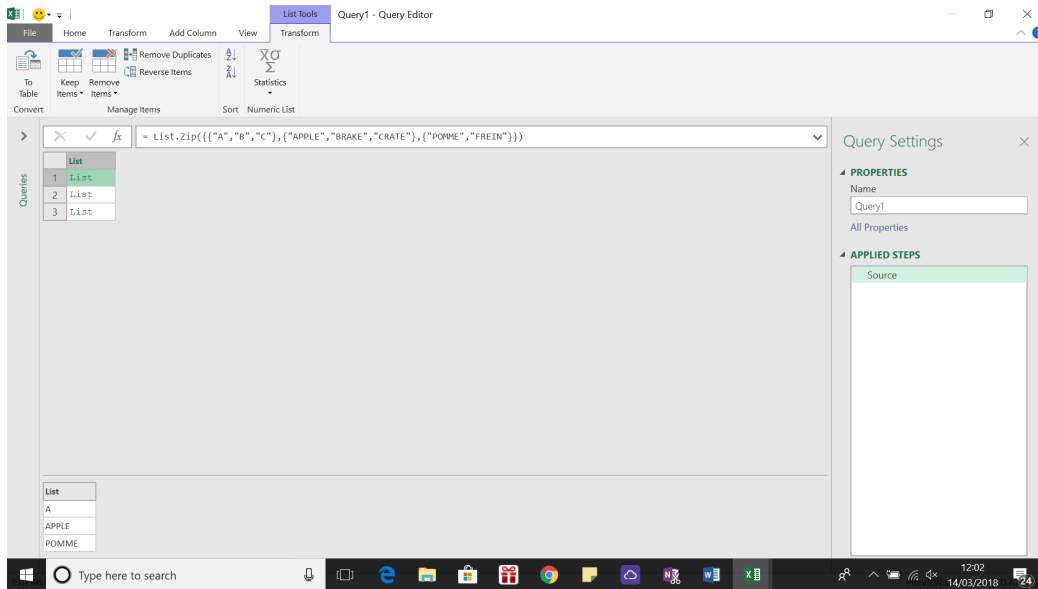


Our values are extracted, and this time the *null* value appears against the 'I FOR' instead of offsetting the values. If the lists were of different lengths, then *null* would also be used in place of any 'missing' values at the end.

If we wanted to check what was in each of our 'list' values in the original list without going through the table conversion, we may simply click on the list we wish to see.



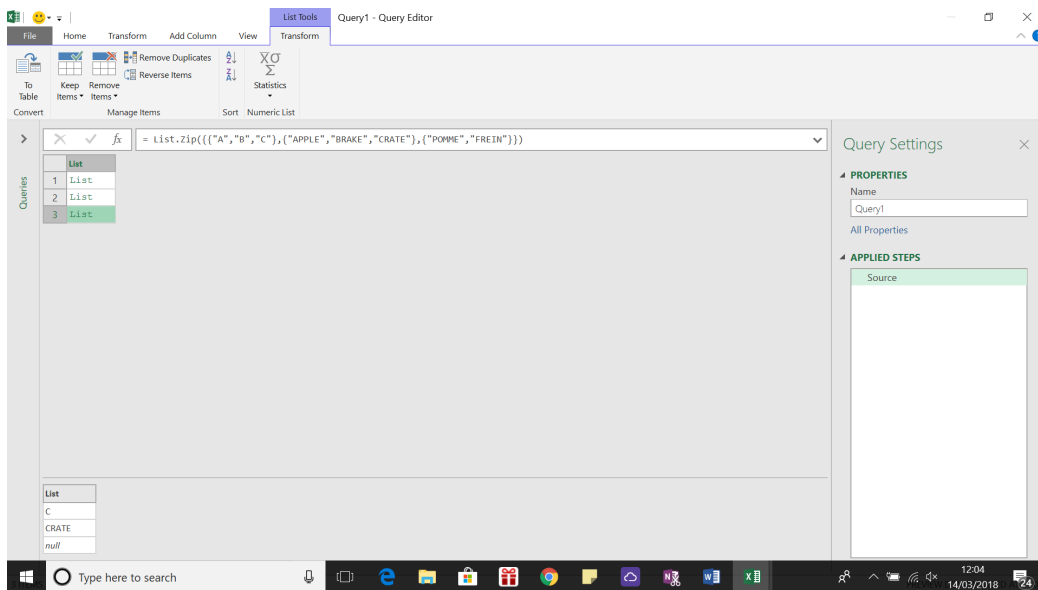
Our example has two lists, but we may enter multiple lists, as shown in the example below:



This time we have used a list of three simple lists, viz.

`= List.Zip({{"A","B","C"}, {"APPLE","BRAKE","CRATE"}, {"POMME","FREIN"}})`

Since my third list is two entries instead of three, a *null* is inserted at the end of the last list.



More next month!

Power BI Updates

This month's revisions include updates for the new Format pane and Power BI Goals, amongst other improvements / new features. The full list is as follows:

Reporting

- Updates for the new Format pane (in Preview)
- ToolTips now support drill actions for matrix, line and area charts (in Preview)
- Updates to required permissions when using composite models on Power BI datasets
- Error bars for clustered column and bar charts (in Preview)
- Bookmark navigator now shows last selected bookmark per group

Data Connectivity and Preparation

- Dynamic M query parameters are now Generally Available

Data connectivity

- Autodesk Construction Cloud (New Connector)
- Digital Construction Works Insights (Connector Update)
- Dremio, Dremio Cloud (Connector Updates)

Service

- Power BI Goals updates:
 - Bulk updating
 - Updating sharing and access banner

Mobile

- Mobile formatting options support visual interaction
- Goals push notifications

Visualisations

- New visuals in AppSource
- Drill Down TimeSeries PRO by ZoomCharts
- PowerGantt Chart by Nova Silva

Other

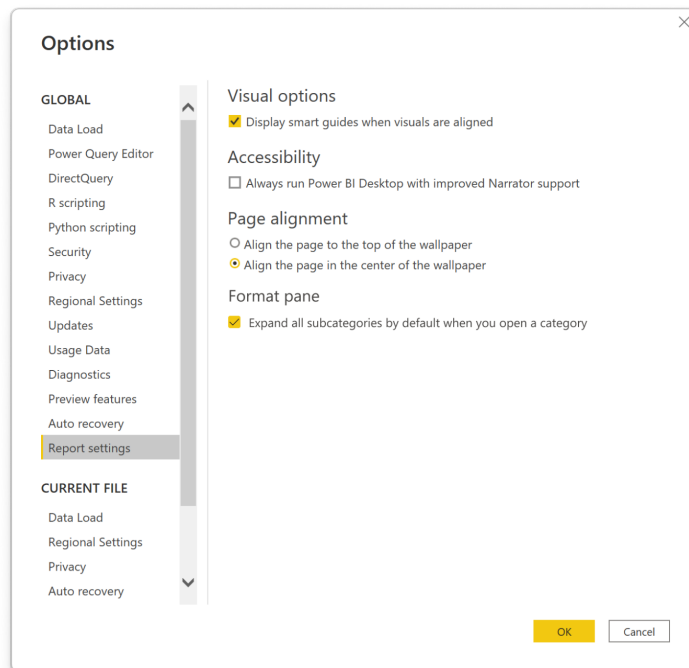
- Windows 11 support.

Let's now go through each in turn.

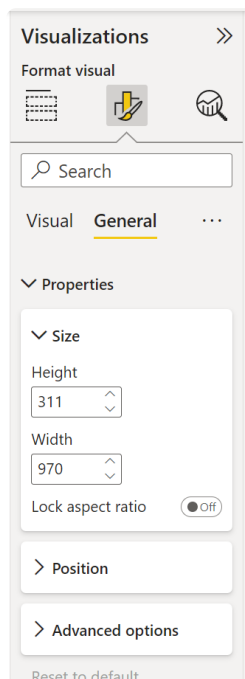
Updates for the new Format pane (in Preview)

As you may recall, the new Format pane has been in Preview since last November and was turned on by default for users in February's release. Microsoft has announced they plan to make this Generally Available this month, which means you will no longer be able to switch back to the old Format pane.

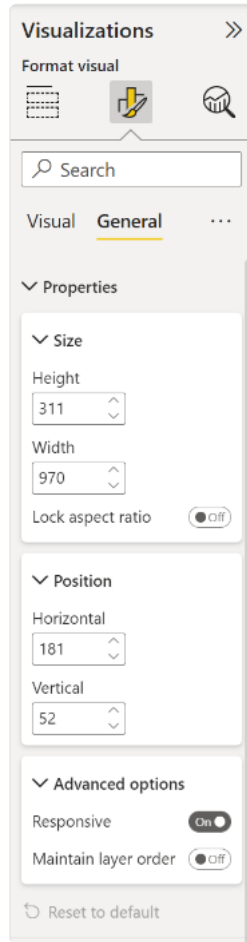
There is more though: Microsoft has received "significant feedback" to expand all sub-categories by default and has cited they will add a user preference setting in next month's release (later in May) to allow all sub-categories to remain expanded when you open a card (yay!). This should feel familiar to the old Format pane behaviour.



Presently, with the current default (setting is off), only the first sub-category is expanded by default to allow for a quick scanning of contents:



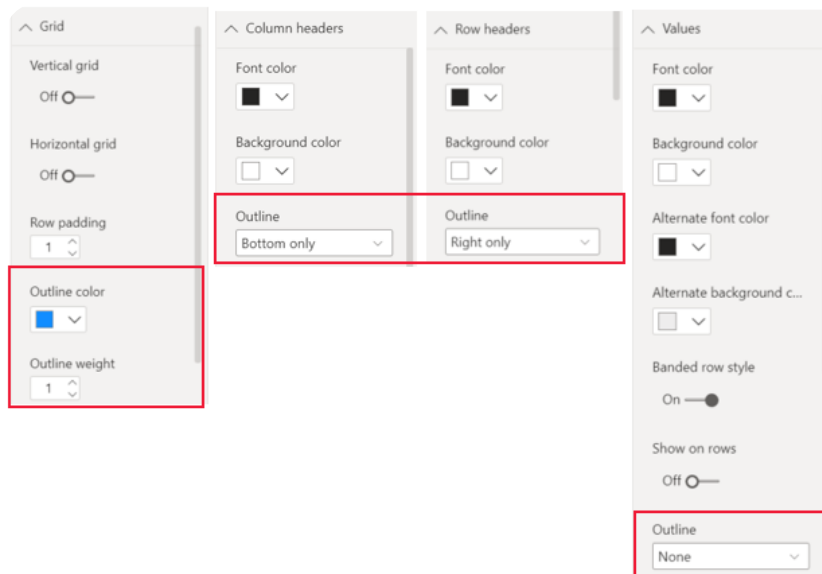
However, if you prefer, you may switch the setting on and all sub-categories will be expanded to reduce the extra clicks (surely only a glutton for punishment would not select this option?):



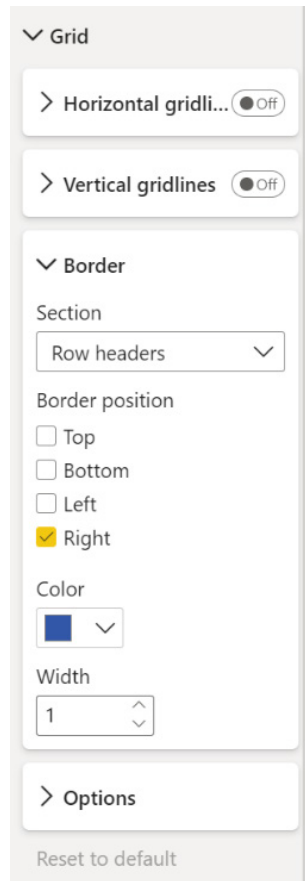
That's not all, either. Microsoft has been busy with the following further improvements:

- Shape map custom colour has been re-added
- Scatter chart 'show blank values' toggle has been re-added
- Slicers with slider type (between, before, after) responsive toggle has been re-added
- button icon size setting has been re-added
- the bug causing a delay in cursor position for text input boxes has been fixed.

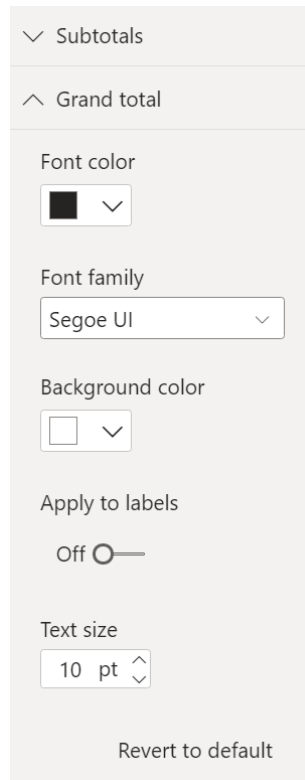
In addition, the outline settings for matrices and tables have been consolidated, so that they may be set from the Grid card. Previously, users would need to go to the Grid card to set the colour and width, and then to each section of the matrix and adjust the border location.



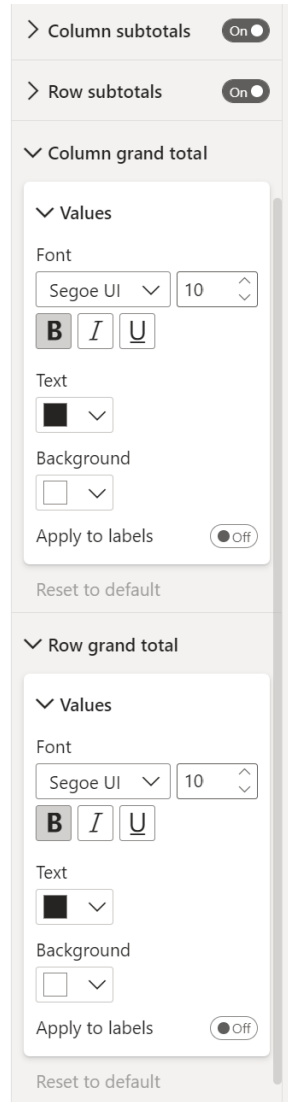
Now, users may adjust the borders for the whole matrix and tweak each section from the same place within the Grid card.



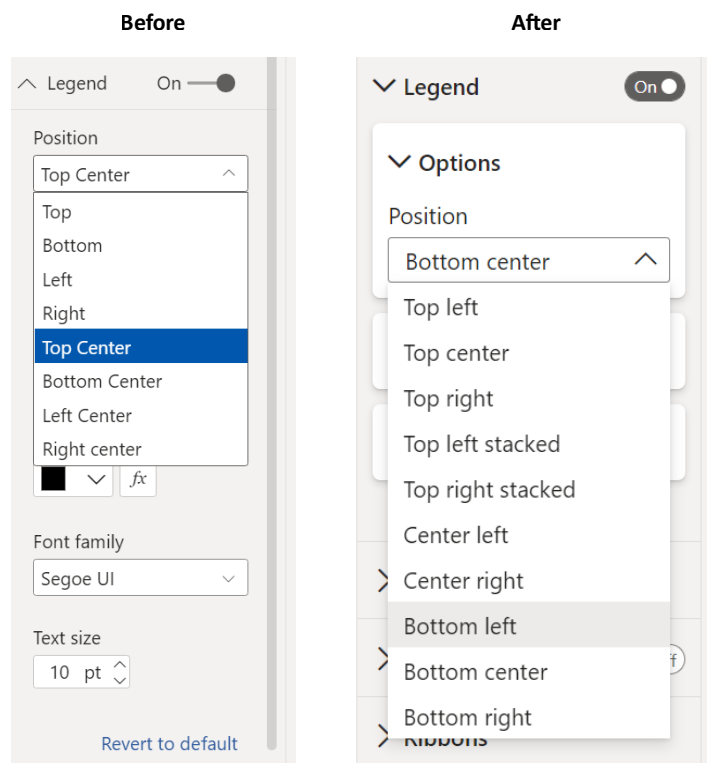
Another improvement is that Power BI now allows you to split the treatment for column grand totals and row grand totals for matrix visuals. Previously, changing grand totals would affect both row and columns.



Now, you may format each separately:



Finally, legend options have been improved with top right and bottom right positioning now permissible:



ToolTips now support drill actions for matrix, line and area charts (in Preview)

Last year, Microsoft released a Preview of the Modern visual ToolTips feature that allowed for datapoint drill down or drill through actions from the tooltip itself. The initial Preview did not include drill support for a few of the built-in visuals. However, this update sees this capability added to the following visuals:

- Matrix
- Line charts
- Area charts
- Stacked area charts.

State	TotalDue	TaxAmt
California	\$346,517.6072	\$25,087.2476
New Mexico	\$15,275.1977	\$1,105.8967
Colorado	\$14,017.9083	
Nevada	\$7,330.0000	
Utah	\$660.0000	
Total	\$383,800.0000	

State Colorado
TotalDue \$14,017.9083

Drill down
Drill through >

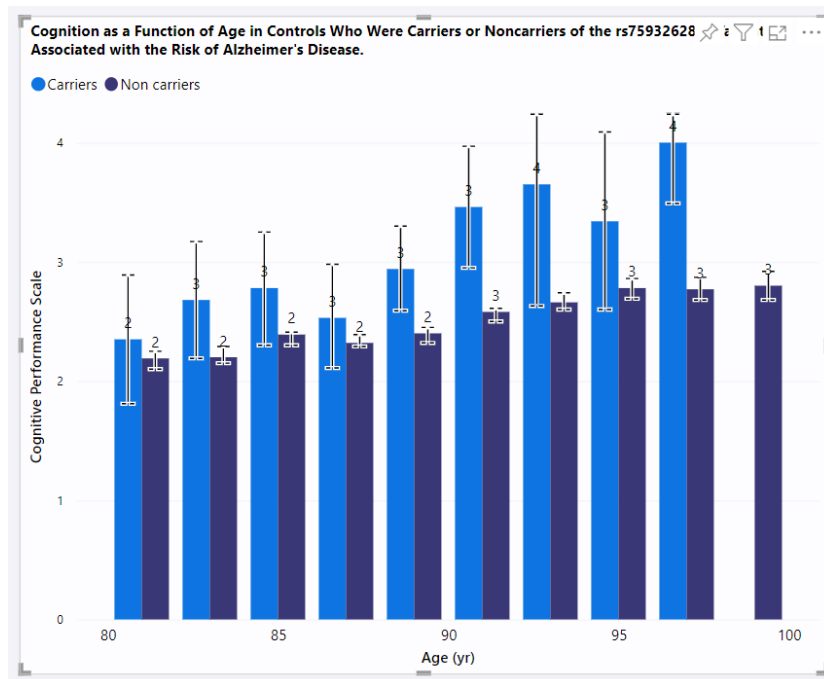
Updates to required permissions when using composite models on Power BI datasets

So far, a user working with a report that leverages a composite model based upon a Power BI dataset required Build permissions (or Contributor role) on all datasets in the chain. Many have complained that the Build permission is overly permissive for a user that just needs to consume

a report. Therefore, starting today, for datasets in Premium or PPU workspaces, a user that consumes a report that leverages a composite model based upon a Power BI dataset is required to have Read permissions (or have the Viewer role) on all datasets in the chain.

Error bars for clustered column and bar charts (in Preview)

Last month's updates introduced the error bars Preview feature for Line charts to help users visualise the uncertainty in their data. This functionality has now been added for Clustered Column and Bar charts as well.



You can create error bars using the same method as with Line charts: simply drag upper and lower bound fields into the field wells in the error bars card of the Analytics pane. You'll find a more limited set of options than with the Line charts, given shade areas are not very meaningful with

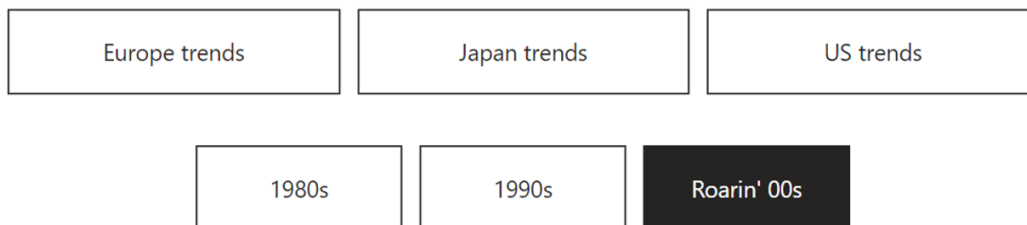
discrete columns. You'll also notice that error bars are enabled specifically for Clustered charts and not Stacked, where such a visualisation would be unclear about the categories being represented.

Bookmark navigator now shows last selected bookmark per group

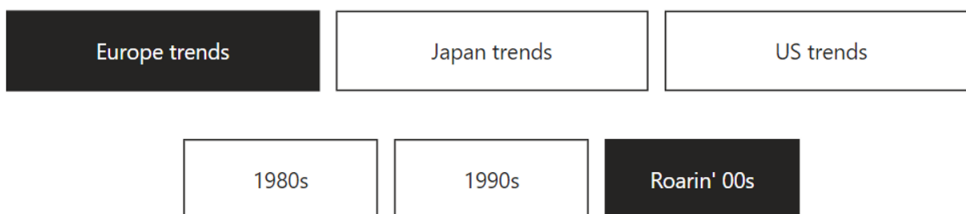
To better support interaction between multiple navigators, bookmark groups and interaction with elements on the report, this update ensures that once a bookmark is clicked on the navigator, it will continue to stay selected, regardless of changes in the report state. The “active” bookmark in any navigator will continue to stay active until another bookmark contained within the navigator is selected, either from the navigator or the Bookmarks pane. Before, selecting any other bookmark, including those not in the bookmark group represented by a navigator, would clear the selections in the bookmark navigator. This small behaviour change

enables the use of multiple navigators targeting separate bookmark groups to operate and have independent “active” bookmarks.

For example, imagine you have two bookmark navigators, one which reflects the bookmark group you’ve made to filter a report page by country, and one which reflects the bookmark group you’ve made to filter by decade. Before, selecting a decade would clear the country selection, and vice versa, even if the filter being applied from the country selection is still active:



Now, however, both bookmarks (selected country and selected decade) will remain selected:



Note that you can still change the report state without selecting another bookmark to leave the state captured by that bookmark (e.g. by manually changing a filter away from Europe trends to US trends). Your bookmark navigators will still maintain their selection state.

Dynamic M query parameters are now Generally Available

The Dynamic M query parameters feature is now Generally Available. You may recall that this feature allows report viewers to dynamically set the value(s) for an M query parameter using filters or slicers. This can be especially helpful for those needing query performance optimisations without sacrificing report interactivity.

As part of this Generally Available release, Dynamic M query parameters are now compatible with the following AI features:

- Q&A
- Decomposition Tree
- Key Influencers
- Anomaly Detection.

Consider the following illustration, using a Dynamic M query parameter and setting the value of the M query parameter using the Color (sic) options slicer. In this example, the colour options may be Blue, Multi or Red, and the Q&A visual and Decomposition Tree are filtered to those values.



Autodesk Construction Cloud (New Connector)

The Autodesk Construction Cloud (ACC) Power BI connector allows customers with unified ACC accounts and BIM360 accounts to import their Data Connector extractions directly into Power BI. The data connector allows teams to quickly extract project data across their entire account and with this certified connector, customers can now streamline

the data extraction process and set a schedule for their Power BI data set to be updated automatically with the latest Data Connector extract. This simplifies users' ability to gain valuable insights on their project data so they may make more informed business decisions.

Digital Construction Works Insights (Connector Update)

This release fixes an issue where the Digital Construction Works Insights connector was not able to successfully connect in Power BI Service.

Dremio, Dremio Cloud (Connector Updates)

The Power BI Certified Connector to Dremio has been updated. Users may now connect to Dremio from Power BI Service without requiring the on-premises data gateway.

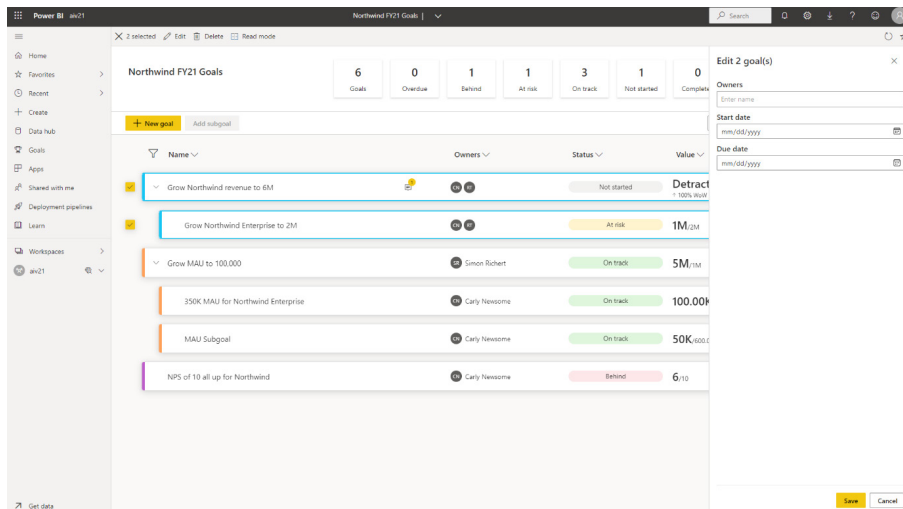
existing connections and reports. You may need to re-enter your credentials and possibly re-publish existing reports once you have upgraded to use this new connector. Whether or not you are affected will depend on how your previous gateway was set up.

Some users may experience a backward-compatibility issue with their

Power BI Goals updates

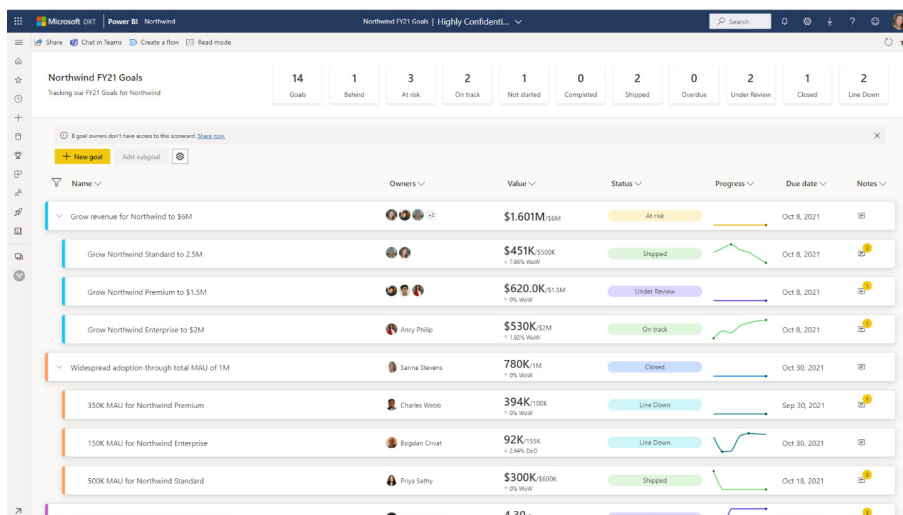
This month sees the announcement of two updates for Power BI Goals: additions for bulk updating capabilities and a helpful access and sharing banner.

BULK UPDATING



With bulk updating, you will be able to select multiple goals to make changes to due dates and owners, as well as delete multiple goals at once. More capabilities are to be rolled out in the coming weeks such as status changes, status rules, etc.

UPDATING SHARING AND ACCESS BANNER

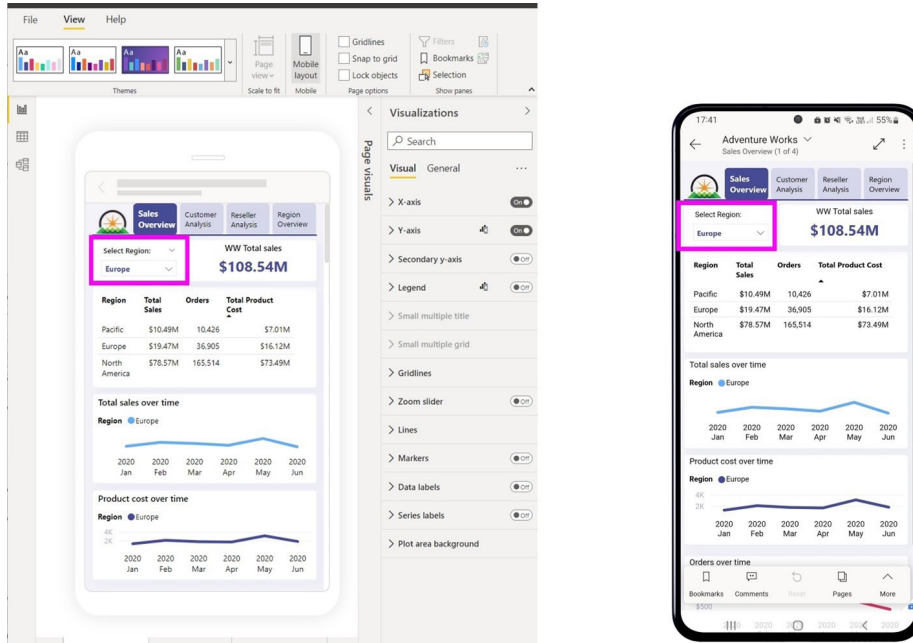


The new updates have included a banner that calls out when goal owners do not have access to the scorecard that contains their goals. With a few clicks, you can now ensure that all owners can access and provide

updates on their goals' progress, making it easier to keep your scorecard up to date.

Mobile formatting options support visual interaction

In this release, the visual interaction support has been extended to the new Mobile formatting. Now, when you use the new mobile formatting options to create a mobile optimised report, your visual interaction settings will be applied both in the Mobile layout and when you view the report on your mobile phone.



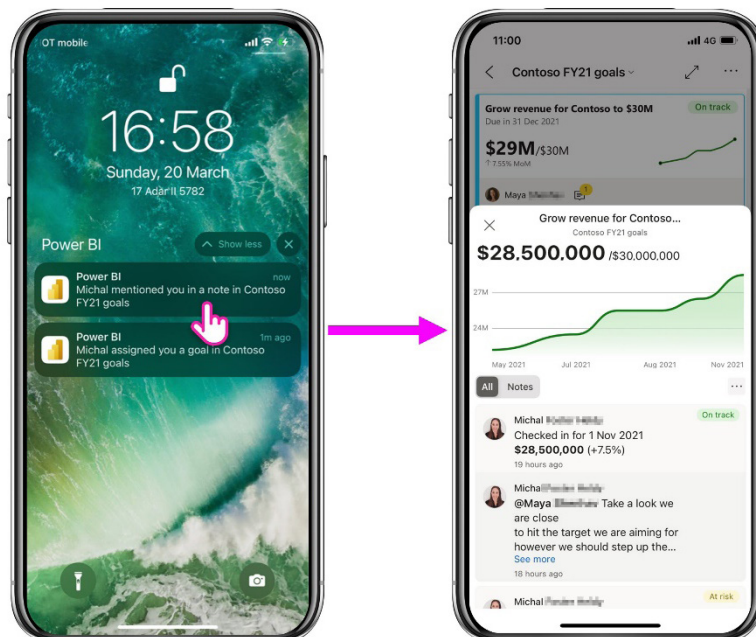
Goals push notifications

In this release, the visual interaction support has been extended to the new Mobile formatting. Now, when you use the new mobile formatting options to create a mobile optimised report, your visual interaction settings will be applied both in the Mobile layout and when you view the report on your mobile phone.

1. **Goal assignment notification:** whenever someone assigns you a goal, you'll receive a goal notification on your mobile device. This way you'll know when you've become a goal owner and are now accountable for that goal. Holidays will never be the same again!

2. **Notifications for mentions:** whenever you're mentioned in a note attached to a new or existing check-in, you'll receive a notification on your mobile device. Now, when someone mentions you, requesting your attention or comment, you won't miss it. You may reply directly from the app too.

For both kinds of notification, tapping the notification takes you right to the goal's details pane inside the scorecard, where you can see all the necessary information.



New visuals in AppSource



For the record, the following are new visuals this update:

- Animator for Power BI
- Data Label Placement.

Drill Down TimeSeries PRO by ZoomCharts

Visualising a Power BI timeline may be achieved now using the Drill Down TimeSeries PRO by ZoomCharts. This visual lets you explore time-based data right down to the millisecond. Featuring vast interaction options

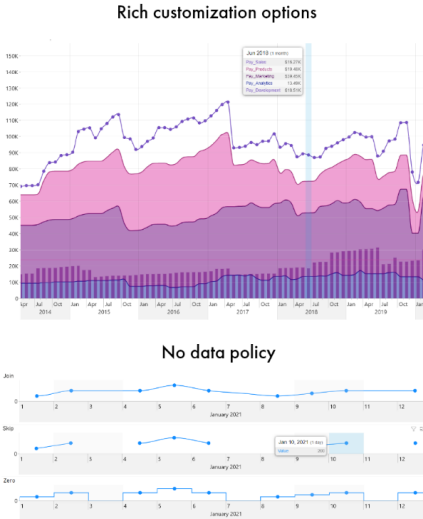
and smooth animations, you may combine multiple chart types. You can switch from months to weeks, to days to drill down into the data, and choose between panning and selecting as your filtering method.



Drill Down TimeSeries PRO

- Multi-touch device friendly
- Multiple chart types and combinations
- Cross-chart filtering
- Static and dynamic thresholds
- Full customization

zoomcharts.com/powerbi



The features of Drill Down TimeSeries PRO include:

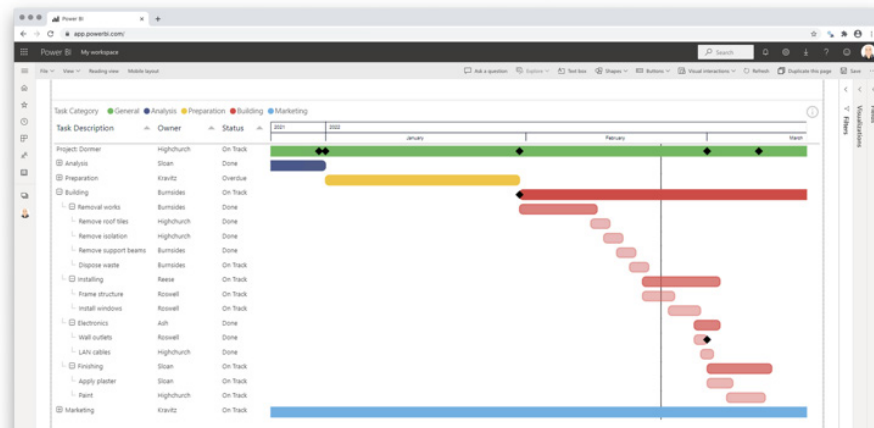
- **multi-touch device friendly:** get the same experience on any device
- **multiple chart types and combinations:** e.g. Column, Area and Line charts. This includes stacking and clustering for all chart types
- **cross-chart filtering:** instead of using slicers, you may select data points on multiple charts
- **static and dynamic thresholds:** you can set up multiple thresholds to demonstrate targets or benchmarks
- **full customisation:** you may use customisation options to combine multiple chart types, set thresholds and convert GMT data to local time.

To evaluate the Power BI timeline capabilities of Drill Down TimeSeries PRO for free, you may start your 30 day trial period now by downloading the Drill Down TimeSeries PRO from the AppSource.

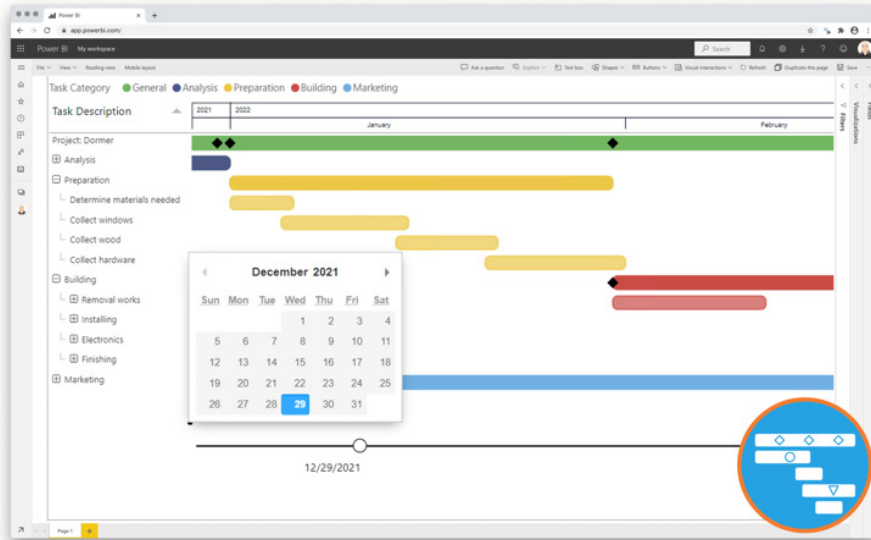
PowerGantt Chart by Nova Silva

The Gantt chart is a useful tool to visualise your program and project schedules and task dependencies. The horizontal bars show for each task / activity the period between the start and end date (or time). Icons may be used to represent milestones: i.e. important moments within your project timeline.

PowerGantt Chart by Nova Silva



Including the Zoom Slider with date picker support



The PowerGantt Chart allows you to review both tasks / activities and milestones in your Power BI reports. The PowerGantt Chart supports an unlimited number of task / activities, hierarchy levels and number of milestones.

Key features of the PowerGantt Chart include:

- **insight into your program and project schedules:** high level overview at first and dive into details using the collapse / expand buttons. Further, you may zoom into specific periods using the dynamic zoom slider
- **customisable:** you can add or remove columns, change colours of bars based upon your own criteria, and change the date granularity
- **no need to learn a new interface:** configuration is the same as a standard Power BI Bar chart. Theme colours and settings are supported.

You may download this visual from the AppSource.

Windows 11 support

Windows 11 introduces four high-contrast themes: Aquatic, Desert, Dusk and Night sky, each with its new colour palette that is distinctive from its counterpart in Windows 10. In this release, Power BI fully supports the new Windows 11 high-contrast themes and is compatible with Windows 10 high-contrast themes and custom themes. Colours of the window background, body text, highlighted text and background,

hyperlink, disabled text, button text and background, and icons will adapt to the user's system colours at runtime. As a result, users with low vision will have a better experience navigating through the Power BI app and utilising all kinds of dialogs, regardless of the Windows operation systems they use. This change does not impact any report themes.

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

The A to Z of Excel Functions: IMDIV

$$\begin{aligned} \frac{a + bi}{c + di} &= \frac{(a + bi)(c - di)}{(c + di)(c - di)} \\ &= \frac{(ac + bd) + i(bc - ad)}{c^2 + d^2} \\ &= \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2} \end{aligned}$$

An imaginary number is a complex number that can be written as a real number multiplied by the imaginary unit i (sometimes denoted j) which is defined by its property $i^2 = -1$. In general, the square of an imaginary

number bi is $-b^2$. For example, $9i$ is an imaginary number, and its square is -81 . Zero is considered to be both real and imaginary.

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

Sometimes you wish to divide one imaginary number (**inumber1**), $z^1 = a + bi$, by another (**inumber2**), $z^2 = c + di$. The **IMDIV** function can help you do this (i.e. z_1 / z_2).

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

$$\text{IMDIV}(\text{inumber1}, \text{inumber2})$$

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

- **inumber1**: this is required and represents the numerator of the required division
- **inumber2**: this is required and represents the denominator of the required division.

It should be further noted that:

- you should use **COMPLEX** to convert real and imaginary coefficients into a complex number
- **IMDIV** recognises either the **i** or **j** notation
- if **inumber1** or **inumber2** is a value that is not in the **x + yi** or **x + yj** text format, **IMDIV** returns the **#NUM!** error value
- if **inumber1** or **inumber2** is a logical value, **IMDIV** returns the **#VALUE!** error value
- if the complex number ends in **+i** or **-i** (or **j**), i.e. there is no coefficient between the operator and the imaginary unit, there must be no space, otherwise **IMDIV** will return an **#NUM!** error
- The quotient of the two complex numbers is calculated as

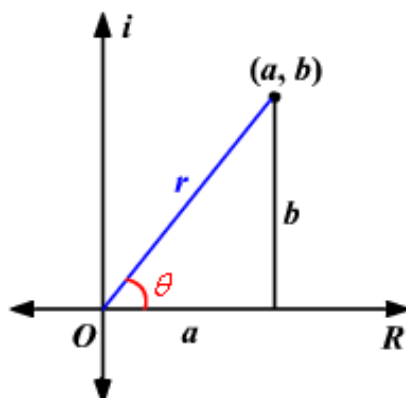
$$\text{IMDIV}(z_1, z_2) = \frac{(a + bi)}{(c + di)} = \frac{(ac + bd) + (bc - ad)i}{c^2 + d^2}$$

Please see our example below:

	A	B	C
1	Formula	Description	Result
2	=IMDIV("-238+240i", "10+24i")	Quotient of the two complex numbers	5+12i

The A to Z of Excel Functions: IMEXP

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



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

From the Pythagorean Theorem,

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

By using the basic trigonometric ratios,

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

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

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

Therefore,

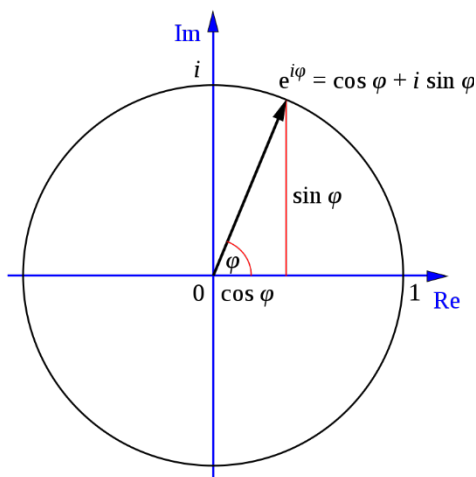
$$z = a + bi$$

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

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

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

Using **Euler's Formula**,



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

The exponential of $a + bi$ is therefore calculated as

$$e^{(a + bi)} = e^a e^{bi} = e^a (\cos b + i \sin b).$$

The **IMEXP** function returns the exponential of a complex number in $x + yi$ or $x + yj$ text format and employs the following syntax to operate:

$$\text{IMEXP}(\text{inumber})$$

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

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

It should be further noted that:

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

$$\text{IMEXP}(z) = e^{(x + yi)} = e^x e^{yi} = e^x (\cos y + i \sin y).$$

Please see our last example below:

	A	B	C
1	Formula	Description	Result
2	=IMEXP("3 + 4i")	Exponential of the complex number 3 + 4i	-13.1287830814622-15.200784463068i

More Excel Functions next month.

Beat the Boredom Suggested Solution

Let's recap our problem from earlier in the newsletter where we wanted you to generate a list of duplicates from the original list containing a mix of "real" numbers, numbers with leading zeroes, and numbers in text strings

As mentioned earlier numbers with leading zeroes may sometimes cause problems. In order to find duplicates from a list, you might think of using

FILTER with **COUNTIF**. However, **COUNTIF** cannot distinguish between numbers with and without leading zeroes, or "real" numbers and numbers stored as text or else in text format.

As you can see below, the number of occurrences of each number is calculated incorrectly by **COUNTIF**:

List	COUNTIF
12	4
123	5
012	4
0123	5
0123	5
12	4
12	4
123	5
123	5

`=COUNTIF(Number[List],Number[@List])`

Therefore, this challenge was trickier than it may have first appeared.

This month's challenge was to write a **formula in one cell** using Dynamic Arrays that would spill down to generate a list of duplicates (*i.e.* all numbers that show up more than once) from the number list in the file above. The result had to be similar to the list on the right (*below*):

List	Solution
12	0123
123	12
012	123
0123	
0123	
12	
12	
123	
123	

As always, there were the "usual" requirements:

- the formula should be in just one cell (no "helper" cells)
- it should treat these kinds of numbers differently:
 - o with and without leading zeroes
 - o "real" numbers as opposed to text strings
- this was a formula challenge – no Power Query / Get & Transform or VBA!

Suggested Solutions

When finding duplicates, you may think of some of the following common ways:

- **COUNTIF / COUNTIFS**
- **SUMPRODUCT**
- **EXACT**
- conditional formatting to highlight duplicates
- *etc.*

However, after we had tried all the options above, we realised that they could not meet the requirements of the question.

For example, if we use Dynamic Arrays with **COUNTIF**, the result may be wrong. As numbers 12 and 123, and string "012" only show up once in the list, they should not be considered as duplicates.

List
12
123
012
0123
0123
12
12
123
123

DAs & COUNTIF
12
123
012
0123
12
123

=FILTER(UNIQUE(Number[List]),COUNTIF(Number[List],UNIQUE(Number[List]))>1)

Before explaining our solution, we will clarify how we came up with it first.

Brainstorming

The aim was to count the number of occurrences of each number in the list to check whether those numbers were considered as duplicates or not.

Firstly, we think of using a matrix with:

- **x axis** being the number list (i.e. **G12#**) with the help of **TRANSPOSE**
- **y axis** being its unique list (i.e. **F13#**) with the help of **UNIQUE**
- counting only numbers from the unique list is enough.

The values in the matrix will be 1 (TRUE) or 0 (FALSE) depending on whether numbers on the **y axis** show up on the **x axis** (i.e. the number list) or not.

	12	123	012	0123	0123	12	12	123	123
12	1	0	0	0	0	0	0	0	0
123	0	1	0	0	0	0	0	0	0
012	0	0	1	0	0	0	0	0	0
0123	0	0	0	1	1	0	0	0	0
12	0	0	0	0	0	1	1	0	0
123	0	0	0	0	0	0	0	1	1

Secondly, we need to sum up each row of the matrix to get the frequency each number show up.

As you may know, we cannot use **SUM** with Dynamic Arrays to calculate each row's result as it will aggregate the result of all rows. Hence, we will apply a trick with the **MMULT** function (see below) to multiply the matrix above by a matrix containing only number ones [1's].

The second matrix is created by **SEQUENCE**. The number of rows of this matrix needs to be the same as the number of columns of the first matrix above. The formula is as follows:

=SEQUENCE(COLUMNS(G13#),,1,0)

	12	123	012	0123	0123	12	12	123	123
12	1	0	0	0	0	0	0	0	0
123	0	1	0	0	0	0	0	0	0
012	0	0	1	0	0	0	0	0	0
0123	0	0	0	1	1	0	0	0	0
12	0	0	0	0	0	1	1	0	0
123	0	0	0	0	0	0	0	1	1

The syntax of the **MMULT** function is as below. It will return the matrix product of two specified arrays or matrices:

=MMULT(array1, array2)

Hence, we will get a result matrix on column **R** below, which is a list of the number of occurrences.

Finally, we just need to use the **FILTER** function to remove all the numbers which only occur once.

=FILTER(F13#,R13#>1)

Returning to the Suggested Solution

You may wonder why the challenge only allows a formula cell while there are quite a number of working steps above.

Our solution is a combination of all steps above within a **LET** formula as follows:

=LET(List,Number[List], matrix,(UNIQUE(List)=TRANSPOSE(List))+0, FILTER(UNIQUE(List),MMULT(matrix,SEQUENCE(ROWS(List),,,0))>1))

There are two variables, namely **List** and **matrix**. **List** is the original number list in the question, whereas **matrix** is the first **matrix** to calculate the number of occurrences. Then, the final part of the formula is the calculation to get the list of duplicates, viz.

Although it is a long and complex formula, you can apply it to your own list by only replacing the value (i.e. **Number[List]**) of the variable **List**.

Until next time.

Upcoming SumProduct Training Courses - COVID-19 update

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

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

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

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	10 - 12 May 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	17 May 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	1 Day

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Financial Modelling	18 - 19 May 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	19 - 21 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	26 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	27 - 28 Jul 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	2 Days
Online (Australia)	Excel Tips and Tricks	29 Aug 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	30 - 31 Aug 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	28 -30 Sep 2022	09:00-17:00 AEST	(-1 day) 23:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	5 Oct 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	6 - 7 Oct 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	9 - 11 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	16 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	17 - 18 Nov 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	7 - 9 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	14 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	1 Day
Online (Australia)	Financial Modelling	15 - 16 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-06:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month we thought we'd do the **RIGHT** thing with this month's keyboard shortcuts:

Keystroke	What it does
UP	Move up one cell
ALT + UP	Close drop-down (auto-complete, filter or validation)
CTRL + UP	Select the cell at the top of the region
SHIFT + UP	Extend selection up one cell
CTRL + ALT + UP	Intel Chipset: Display screen normally (zero [0] degrees)
CTRL + SHIFT + UP	Extend selection up to the end of the region

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