Sum Product

NEWSLETTER #124 - March 2023

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

I need a link for my main article this month as it's about the errors made in linking files together in Excel. Do you have a foolproof method for linking files? We explore one alternative below. This newsletter also redresses the misbalance from last one month when we learned just a tad too late for the printing deadline that Value Preview ToolTips have come to Excel. We explain what they are as our lead article this month. Check it out below.

Otherwise, March roars on with all the regulars: there's our Beat the Boredom Challenge, Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates and Excel Updates all ready for the reading: have you got the stamina to plough through?

We mention a new update for Power Query in Excel for Mac and the A to Z of Excel functions series finally gets to the end of the "I" functions (anyone know functions beginning with "J"?? Find out next month!). Finally, we end taking **CTRL** of the numbers in our Keyboard Shortcuts tips.

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

Liam Bastick, Managing Director, SumProduct

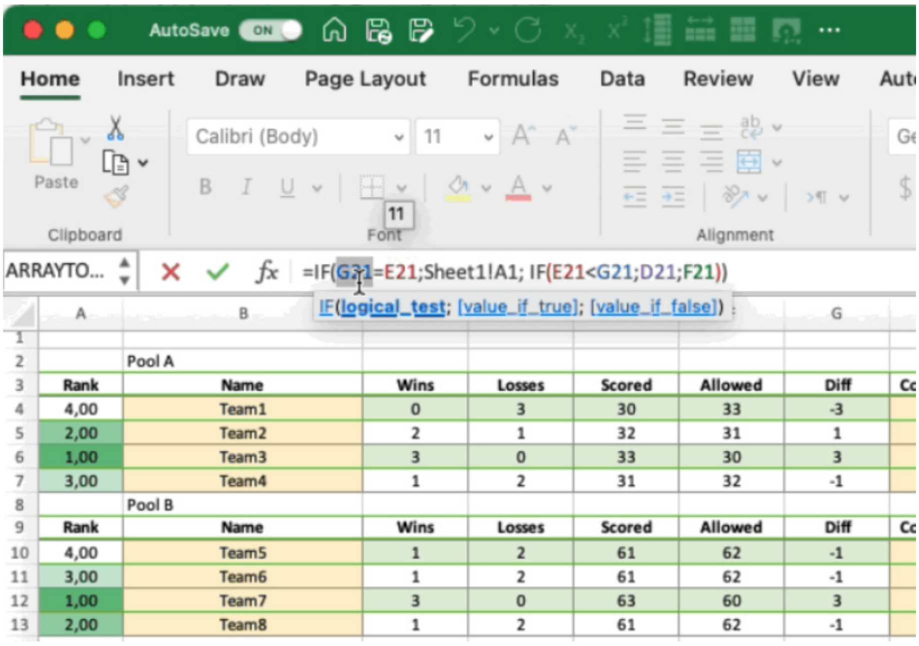
Value Preview ToolTips in Excel

Sadly, the news came too late to put it in last month's newsletter so this one goes right to the top this month! Microsoft has added ToolTips to help you check your formulae directly in the Formula bar or in the cell you're editing. All you need to do is select part of the formula, and Excel will display a ToolTip that contains the current value of the part you highlighted. They're called **Value Preview ToolTips**, a catchy title if ever there wasn't one. But don't let that put you off!

Up until now, a common technique modellers employed to obtain the current value of (part of) a formula was to select the formula (or part thereof) and press **F9**. Doing so would replace (the selected part of) the formula with its current value. You'd probably only do this to check a value, and then you press **ESC** to avoid keeping it as a hard-coded value in your formula.

Alternatively, another approach was to use the 'Evaluate Formula' dialog (**ALT + M + V** or **Formulas -> Evaluate Formula**), which let you check the current value of (each part of) your formula. ToolTips are intended to be easier to use and less burdensome should you just want to focus on one part of the formula.

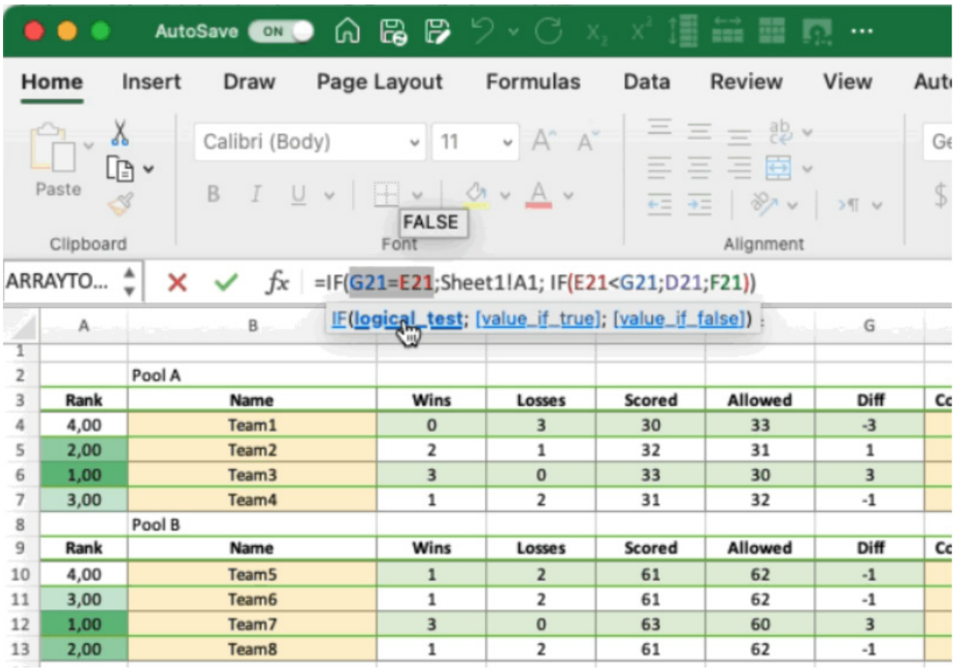
To get this new feature to work, in any formula, simply select the part you wish to evaluate. For example, in a ToolTip that shows the function syntax, click any of the function parameters to select it, and then take note of the ToolTip that appears which shows the current value of that parameter.



=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Co
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Co
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	

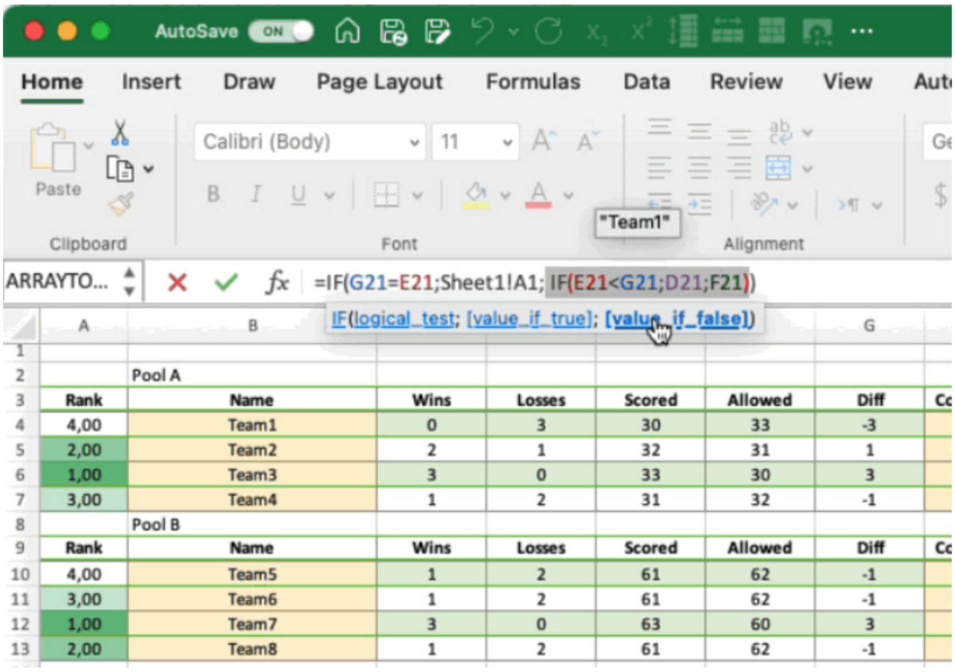


FALSE

=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	



"Team1"

=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	

It should be noted that:

- you may select references, functions, parameters within a function or even the entire formula
- if you'd like to turn off the ToolTips, you can press **CTRL + ALT + P** toggle the feature on or off, but note:
  - on Excel for Mac, this only works when you are not editing a cell
  - on Excel for Windows, it may be toggled on / off anytime (if a ToolTip is visible, pressing the shortcut will not immediately hide the ToolTip, but it will switch the feature off so that subsequent selections won't show the ToolTip).

Thus, in any formula that contains range or table references, you may select a reference and see that the ToolTip shows the value or values in the referenced cells. Further, while editing a formula that has one or more functions, you may also place your cursor inside the brackets of the function so that the syntax ToolTip appears. Then, click one of the parameter names in the syntax to select that part of your formula.

If you select a part of the formula that can't be evaluated, you won't see a ToolTip. For example, if you select only part of a reference or part of a parameter, no ToolTip will be displayed.

It's possible to see a tooltip for part of the formula that doesn't get calculated as part of the cell value. For example, if your formula is =101+201, you could select 1+20 and see a tooltip that shows 21, even though that part isn't relevant when calculating the entire formula.

Dates presently show as serial numbers in Excel, rather than how they would be displaced under current locale settings. This *may* change in the future – watch this space!

Finally, you can move the ToolTip if it's covering something you need to see, or if you just want to locate it somewhere else.

This feature is presently rolling out to Beta Channel users running:

- **Windows:** Version 2302 (Build 16116.20000) or later
- **Mac:** Version 16. 70 (Build 230116) or later.

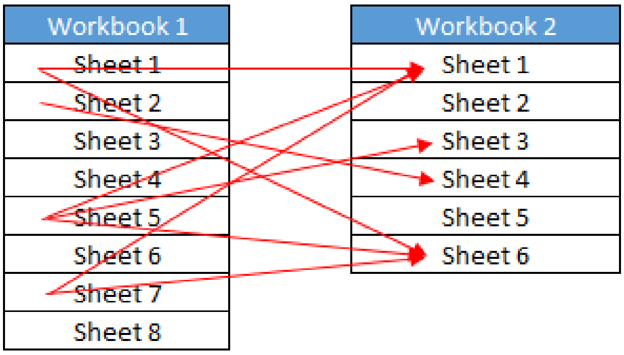
Linking Files

Sometimes our newsletters are inspired by issues we see in recent models encountered and this month's topic has been triggered in this fashion. We have seen a few errors in linking files which could have been avoided.

Before we start, we'd like to emphasis try *not* to link workbooks, where possible. Managing links and detecting the dreaded "phantom links" (*i.e.* links which are unintended) may become a major preoccupation if proper care is not taken. Nonetheless, linking workbooks is sometimes required. Legitimate reasons may include:

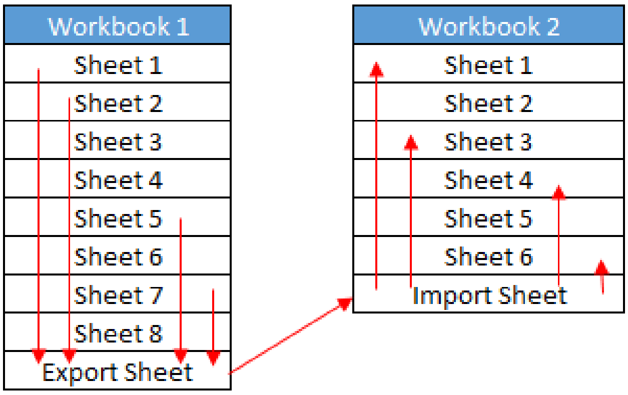
- not sharing confidential data (don't make the mistake of trying to "hide" data – there are countless stories the stuff of legend where businesses have been embarrassed attempting to adopt this approach)
- files are too large to open in a 32-bit environment
- files are too large to email.

Standard industry practice to link two workbooks together tends to conform to the "URM" approach, *viz.*



That's right, meet "URM": the **U**tterly **R**andom **M**ethod. This technique is inadvisable and dangerous. It makes it difficult for end users to identify external links or not be able to differentiate between intended and accidental links. In fact, if links go both ways and only one workbook is open, circular references may be created and worse, be disguised.

If you do have to link workbooks, consider the following approach instead:



This approach is much more uniform:

- identify all cells that need to be exported (*i.e.* linked to another workbook)
- link these cells all to a dedicated 'Export' worksheet
- create a mirror image, dedicated 'Import' worksheet in the destination workbook
- refer to these imported cells in the other sheets.

This way, it is easy to trace and locate all external references. Should you have multiple worksheets to link to, simply create separate import / export pairs for each link (do not put them all in the same worksheet as this will confuse end users).

Some commentators and "Best Practice" guidelines will recommend using a range name for either the source and / or destination cell(s) of the relationship link. Personally, I am not a fan of this method but understand the main reason for this. Using range names will address the issue of inserting or deleting rows and / or columns whilst the linked file is closed; standard cell referencing will not.

It is true range names will overcome this issue. However, the problem is many users cannot readily locate range names and problems may arise with multiple workbooks open with similar or the same range names, especially if you have multiple copies of a particular worksheet. This can, in fact, lead in a worst-case scenario to file corruption.

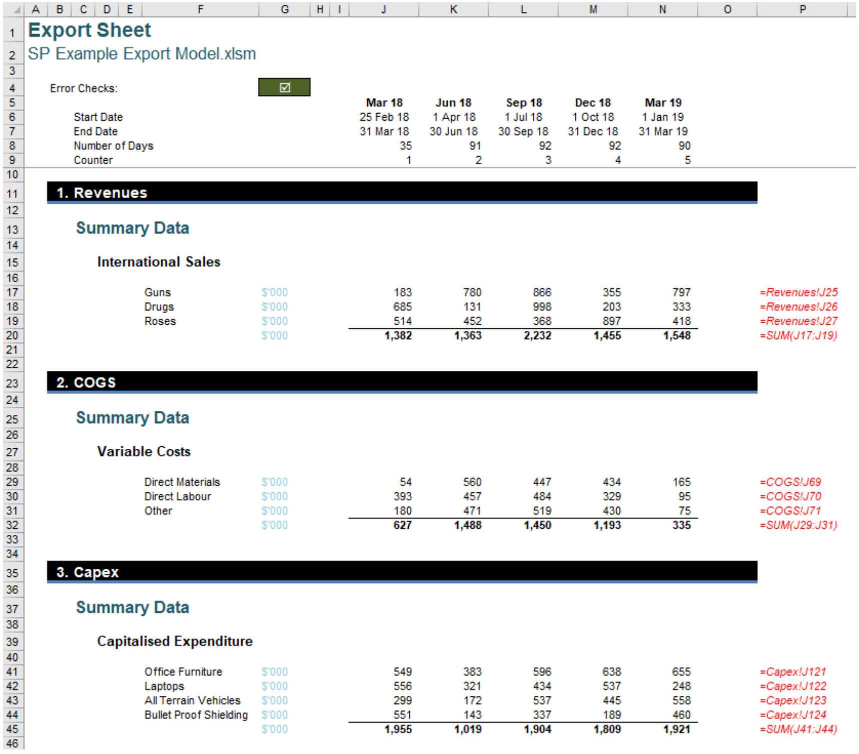
So what should you do instead?

It's very simple, if not ideal. After creating import / export sheets in the way explained above, either hide or protect these said sheets. Their sole purpose is for linking to other workbooks so there is no need to revise the worksheet structure after they have been agreed and created. Yes, you might insert rows or delete columns within an internal worksheet of one workbook whilst the other associated files are closed – but the links will remain intact to the relevant import or export sheet. Simple!

This just leaves one question: how do you create a "mirror image" worksheet? Here's an example of how easy it is to do this…

Example

The first thing to do is clearly identify the data that needs to be linked to another workbook (*i.e.* the data to be exported data). Create an export worksheet and link all the data in:



Export Sheet

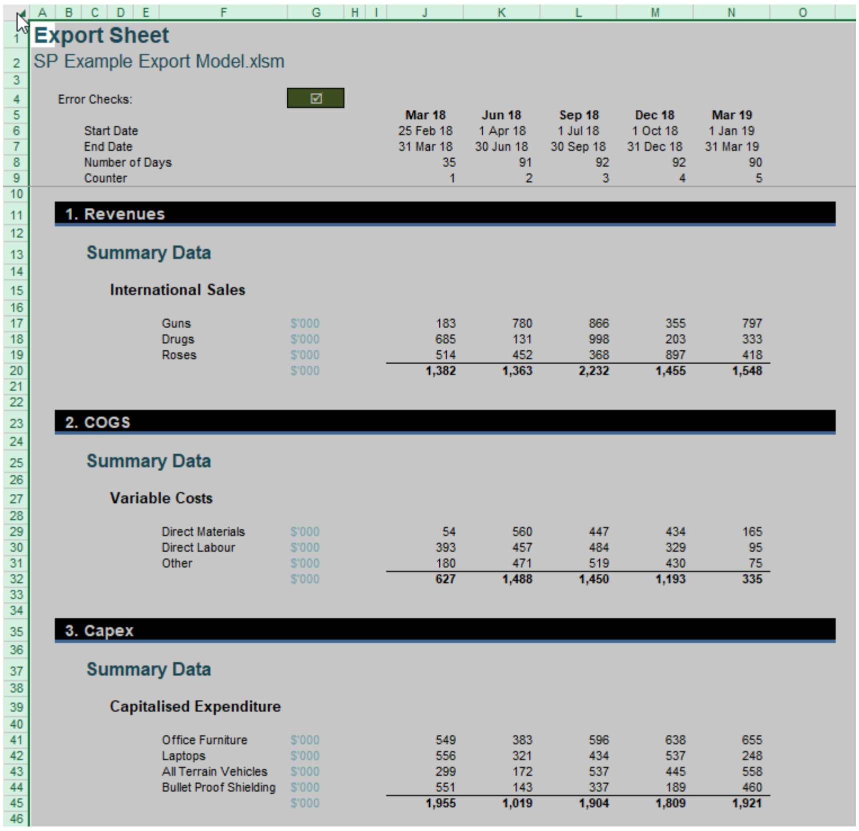
SP Example Export Model.xlsm

Error Checks: ☑

		Mar 18	Jun 18	Sep 18	Dec 18	Mar 19	
Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19	
End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19	
Number of Days		35	91	92	92	90	
Counter		1	2	3	4	5	
1. Revenues							
Summary Data							
International Sales							
Guns	$'000	183	780	866	355	797	=Revenues!J25
Drugs	$'000	685	131	998	203	333	=Revenues!J26
Roses	$'000	514	452	368	897	418	=Revenues!J27
	$'000	**1,382**	**1,363**	**2,232**	**1,455**	**1,548**	=SUM(J17:J19)
2. COGS							
Summary Data							
Variable Costs							
Direct Materials	$'000	54	560	447	434	165	=COGS!J69
Direct Labour	$'000	393	457	484	329	95	=COGS!J70
Other	$'000	180	471	519	430	75	=COGS!J71
	$'000	**627**	**1,488**	**1,450**	**1,193**	**335**	=SUM(J29:J31)
3. Capex							
Summary Data							
Capitalised Expenditure							
Office Furniture	$'000	549	383	596	638	655	=Capex!J121
Laptops	$'000	556	321	434	537	248	=Capex!J122
All Terrain Vehicles	$'000	299	172	537	445	558	=Capex!J123
Bullet Proof Shielding	$'000	551	143	337	189	460	=Capex!J124
	$'000	**1,955**	**1,019**	**1,904**	**1,809**	**1,921**	=SUM(J41:J44)

(The formulae in the red cells are displayed to show where the data may have come from and these cells do not form part of what should be exported.)

After creating a blank sheet in the destination workbook, it's now time to create the "mirror image" import sheet. To do this, highlight the entire 'Export Sheet' by selecting the grey corner cell in the row / column headers:



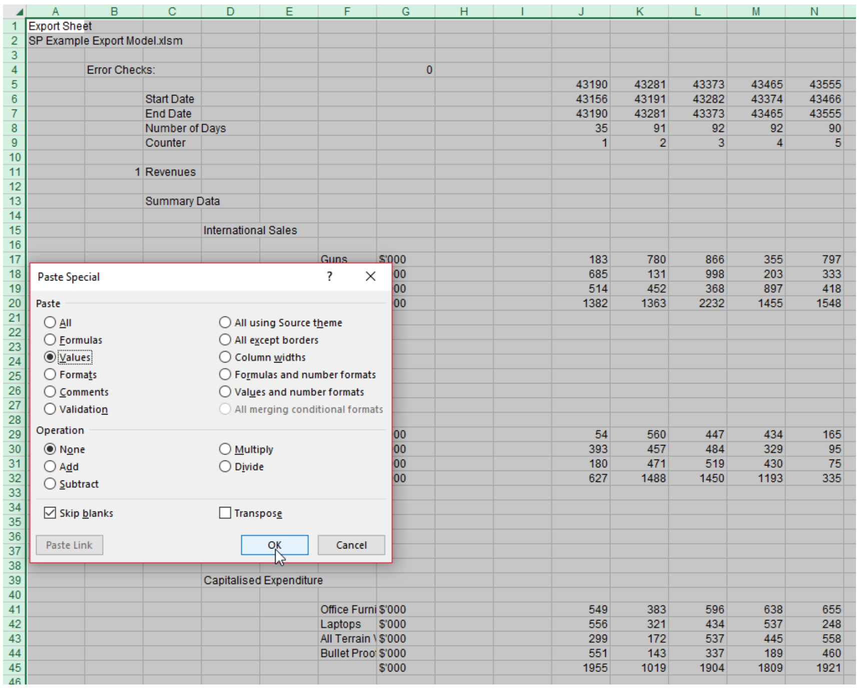
Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

		Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days		35	91	92	92	90
Counter		1	2	3	4	5
1. Revenues						
Summary Data						
International Sales						
Guns	$'000	183	780	866	355	797
Drugs	$'000	685	131	998	203	333
Roses	$'000	514	452	368	897	418
	$'000	**1,382**	**1,363**	**2,232**	**1,455**	**1,548**
2. COGS						
Summary Data						
Variable Costs						
Direct Materials	$'000	54	560	447	434	165
Direct Labour	$'000	393	457	484	329	95
Other	$'000	180	471	519	430	75
	$'000	**627**	**1,488**	**1,450**	**1,193**	**335**
3. Capex						
Summary Data						
Capitalised Expenditure						
Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	**1,955**	**1,019**	**1,904**	**1,809**	**1,921**

Now copy this sheet (**CTRL + C**). Select cell **A1** in the destination worksheet and paste special as <u>values</u> (**ALT + E + S + V + ENTER**):



	A	B	C	D	F	G	J	K	L	M	N
1	Export Sheet										
2	SP Example Export Model.xlsm										
3											
4		Error Checks:				0					
5							43190	43281	43373	43465	43555
6			Start Date				43156	43191	43282	43374	43466
7			End Date				43190	43281	43373	43465	43555
8			Number of Days				35	91	92	92	90
9			Counter				1	2	3	4	5
10											
11		1	Revenues								
12											
13			Summary Data								
14											
15				International Sales							
16											
17					Guns	$'000	183	780	866	355	797
18						00	685	131	998	203	333
19						00	514	452	368	897	418
20						00	1382	1363	2232	1455	1548
29						00	54	560	447	434	165
30						00	393	457	484	329	95
31						00	180	471	519	430	75
32						00	627	1488	1450	1193	335
39				Capitalised Expenditure							
40											
41					Office Furni	$'000	549	383	596	638	655
42					Laptops	$'000	556	321	434	537	248
43					All Terrain \	$'000	299	172	537	445	558
44					Bullet Proo	$'000	551	143	337	189	460
45						$'000	1955	1019	1904	1809	1921

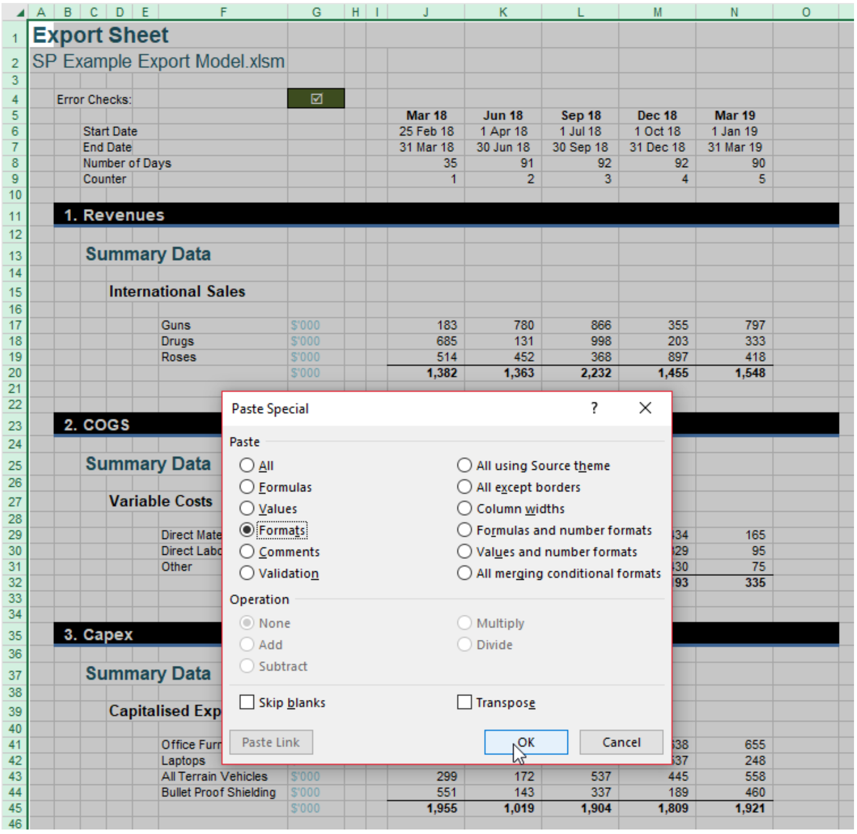
Paste Special

Paste
○ All ○ All using Source theme
○ Formulas ○ All except borders
● Values ○ Column widths
○ Formats ○ Formulas and number formats
○ Comments ○ Values and number formats
○ Validation ○ All merging conditional formats

Operation
● None ○ Multiply
○ Add ○ Divide
○ Subtract

☑ Skip blanks ☐ Transpose

Immediately afterwards, copy and paste special as <u>formats</u> (**ALT + E + S + T + ENTER**) too:



	B–E	F	G	J	K	L	M	N
1	**Export Sheet**							
2	SP Example Export Model.xlsm							
3								
4	Error Checks:		☑					
5				**Mar 18**	**Jun 18**	**Sep 18**	**Dec 18**	**Mar 19**
6	Start Date			25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
7	End Date			31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
8	Number of Days			35	91	92	92	90
9	Counter			1	2	3	4	5
10								
11	**1. Revenues**							
12								
13	**Summary Data**							
14								
15	**International Sales**							
16								
17		Guns	$'000	183	780	866	355	797
18		Drugs	$'000	685	131	998	203	333
19		Roses	$'000	514	452	368	897	418
20			$'000	**1,382**	**1,363**	**2,232**	**1,455**	**1,548**
21								
22								
23	**2. COGS**							
24								
25	**Summary Data**							
26								
27	**Variable Costs**							
28								
29		Direct Mate					34	165
30		Direct Labc					29	95
31		Other					30	75
32							**93**	**335**
33								
34								
35	**3. Capex**							
36								
37	**Summary Data**							
38								
39	**Capitalised Exp**							
40								
41		Office Furr					38	655
42		Laptops					37	248
43		All Terrain Vehicles	$'000	299	172	537	445	558
44		Bullet Proof Shielding	$'000	551	143	337	189	460
45			$'000	**1,955**	**1,019**	**1,904**	**1,809**	**1,921**

Paste Special

Paste
○ All ○ All using Source theme
○ Formulas ○ All except borders
○ Values ○ Column widths
● Formats ○ Formulas and number formats
○ Comments ○ Values and number formats
○ Validation ○ All merging conditional formats

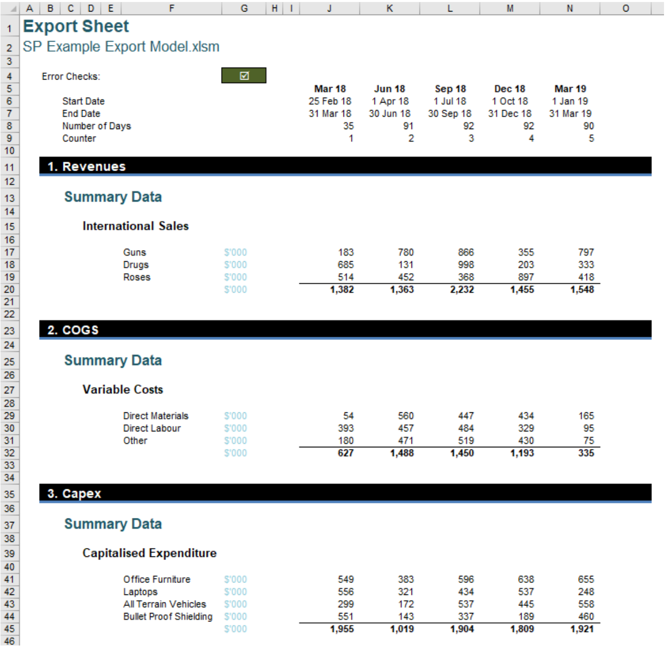
Operation
● None ○ Multiply
○ Add ○ Divide
○ Subtract

☐ Skip blanks ☐ Transpose

Note that if you had gridlines turned off in the source worksheet, this setting will <u>not</u> be replicated in the destination sheet. These may be turned off easily though by going to the 'View' tab in the Ribbon and unchecking 'Gridlines' in the 'Show' group (**ALT + W + VG**).

You <u>must</u> paste as values before you format – this order is important. This is because if you have any merged cells, formulae will not copy across after merging.

Once these actions have been taken, you should have a worksheet that looks like this:



Export Sheet

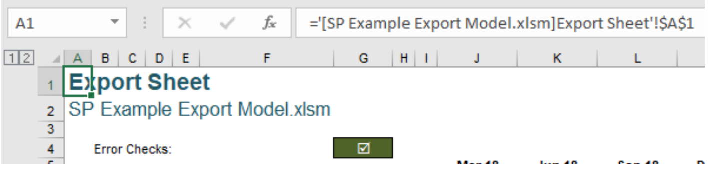
SP Example Export Model.xlsm

Error Checks: ☑

		Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days		35	91	92	92	90
Counter		1	2	3	4	5
1. Revenues						
Summary Data						
International Sales						
Guns	$'000	183	780	866	355	797
Drugs	$'000	685	131	998	203	333
Roses	$'000	514	452	368	897	418
	$'000	**1,382**	**1,363**	**2,232**	**1,455**	**1,548**
2. COGS						
Summary Data						
Variable Costs						
Direct Materials	$'000	54	560	447	434	165
Direct Labour	$'000	393	457	484	329	95
Other	$'000	180	471	519	430	75
	$'000	**627**	**1,488**	**1,450**	**1,193**	**335**
3. Capex						
Summary Data						
Capitalised Expenditure						
Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	**1,955**	**1,019**	**1,904**	**1,809**	**1,921**

It looks like the original, but with all of the data, labels and dates hard coded. This is intentional – we're not finished yet!

Do you see the sheet title in cell **A1** is incorrect? Well, guess what, we're not going to fix it yet. In fact, we're going to "reinforce" this error. Let's link cell **A1** in the import sheet back to cell **A1** of the source export sheet. Given the other workbook must be open by definition, the formula in cell **A1** will look something like this:



A1

='[SP Example Export Model.xlsm]Export Sheet'!$A$1

Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

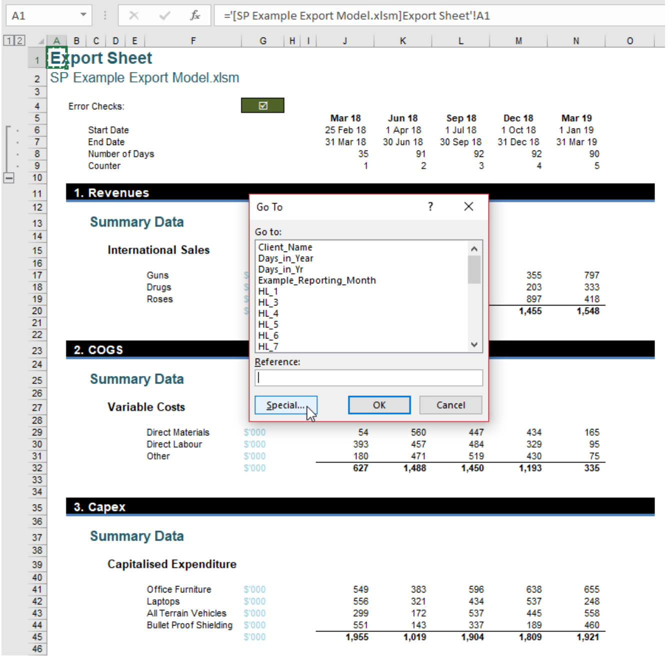
='[SP Example Export Model.xlsm]Export Sheet'!$A$1

Note that all cells linked to other workbooks are made *absolute* (*i.e.* use dollar "$" signs) by default. I need to remove this pair of dollar signs, so I have *relative* referencing:

='[SP Example Export Model.xlsm]Export Sheet'!A1

This will mean as we copy this formula into other cells in the import sheet we will always be linking back to the corresponding cell (*e.g.* cell **B17** will link to **B17**, cell **AA686** will link to **AA686** and so on). This is what we want.

Once this change has been made, copy this formula (**CTRL + C**) into cell **A1** of the import sheet. Now, I need to select where I am pasting this cell. I do not want to paste it into all cells in this sheet, just the non-blank ones. To do this, I load the 'Go To' dialog box (function key **F5** or the keyboard shortcut **CTRL + G**):



A1

='[SP Example Export Model.xlsm]Export Sheet'!A1

Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

		Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days		35	91	92	92	90
Counter		1	2	3	4	5
1. Revenues						
Summary Data						
International Sales						
Guns					355	797
Drugs					203	333
Roses					897	418
					1,455	**1,548**
2. COGS						
Summary Data						
Variable Costs						
Direct Materials	$'000	54	560	447	434	165
Direct Labour	$'000	393	457	484	329	95
Other	$'000	180	471	519	430	75
	$'000	**627**	**1,488**	**1,450**	**1,193**	**335**
3. Capex						
Summary Data						
Capitalised Expenditure						
Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	**1,955**	**1,019**	**1,904**	**1,809**	**1,921**

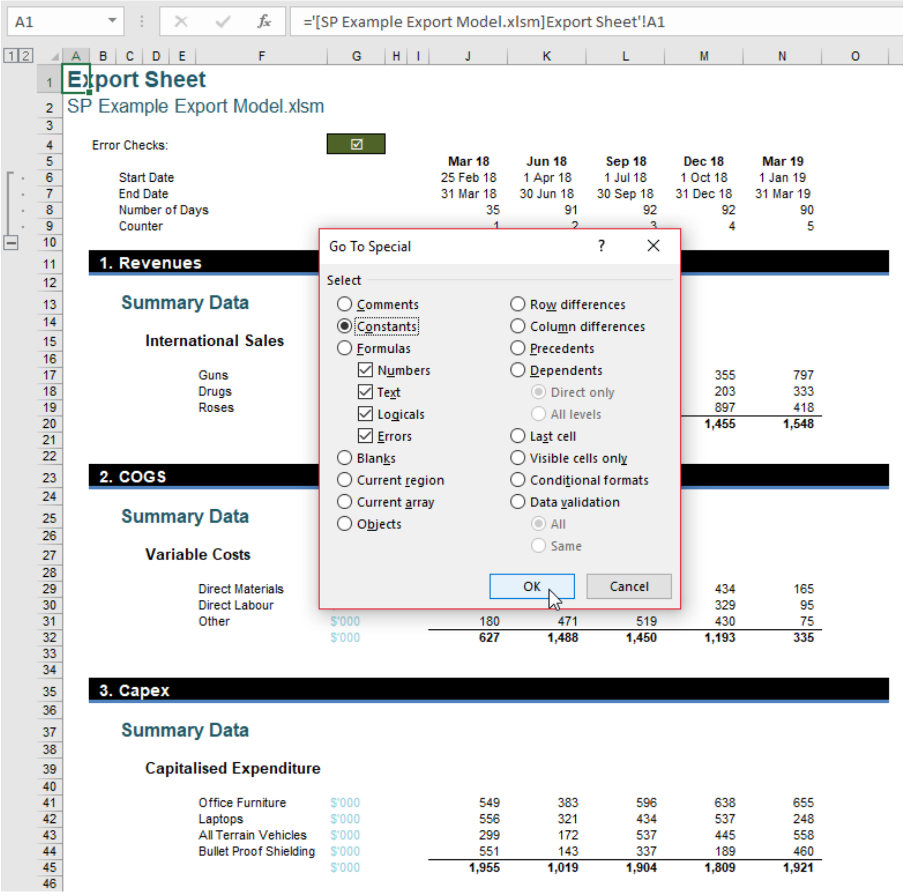
Go To

Go to:

Client_Name
Days_in_Year
Days_in_Yr
Example_Reporting_Month
HL_1
HL_3
HL_4
HL_5
HL_6
HL_7

Reference:

Next, click on the ‘Special…’ button in the bottom left-hand corner:


A1
='[SP Example Export Model.xlsm]Export Sheet'!A1
A B C D E F G H I J K L M N O

Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

	Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date	25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date	31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days	35	91	92	92	90
Counter	1	2	3	4	5

Go To Special

Select

- ○ Comments
- ● Constants
- ○ Formulas
  - ☑ Numbers
  - ☑ Text
  - ☑ Logicals
  - ☑ Errors
- ○ Blanks
- ○ Current region
- ○ Current array
- ○ Objects
- ○ Row differences
- ○ Column differences
- ○ Precedents
- ○ Dependents
  - ○ Direct only
  - ○ All levels
- ○ Last cell
- ○ Visible cells only
- ○ Conditional formats
- ○ Data validation
  - ○ All
  - ○ Same

1. Revenues

Summary Data

International Sales

Guns	355	797
Drugs	203	333
Roses	897	418
	1,455	1,548

2. COGS

Summary Data

Variable Costs

Direct Materials					434	165
Direct Labour					329	95
Other	$'000	180	471	519	430	75
	$'000	627	1,488	1,450	1,193	335

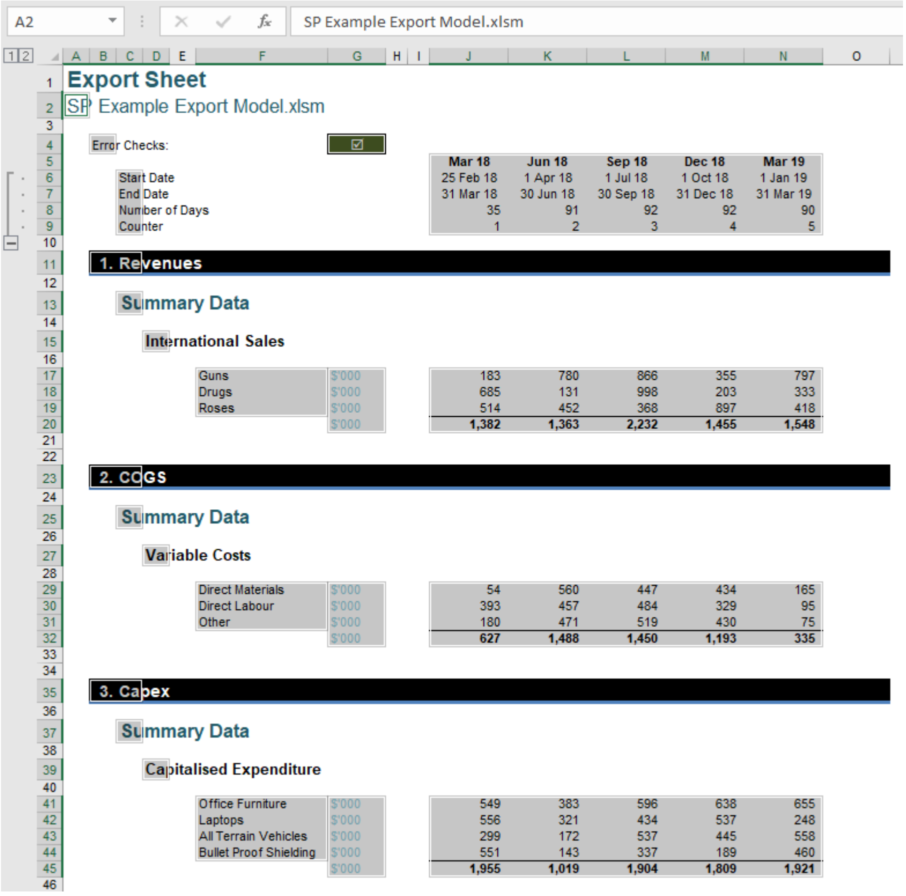
3. Capex

Summary Data

Capitalised Expenditure

Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	1,955	1,019	1,904	1,809	1,921

Select the ‘Constants’ radio button and ensure the ‘Numbers’, ‘Text’, ‘Logicals’ and ‘Errors’ checkboxes are all checked as displayed *(above)*. This means every non-blank cell will now be selected, *viz.*



Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

	Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date	25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date	31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days	35	91	92	92	90
Counter	1	2	3	4	5

1. Revenues

Summary Data

International Sales

Guns	$'000	183	780	866	355	797
Drugs	$'000	685	131	998	203	333
Roses	$'000	514	452	368	897	418
	$'000	1,382	1,363	2,232	1,455	1,548

2. COGS

Summary Data

Variable Costs

Direct Materials	$'000	54	560	447	434	165
Direct Labour	$'000	393	457	484	329	95
Other	$'000	180	471	519	430	75
	$'000	627	1,488	1,450	1,193	335

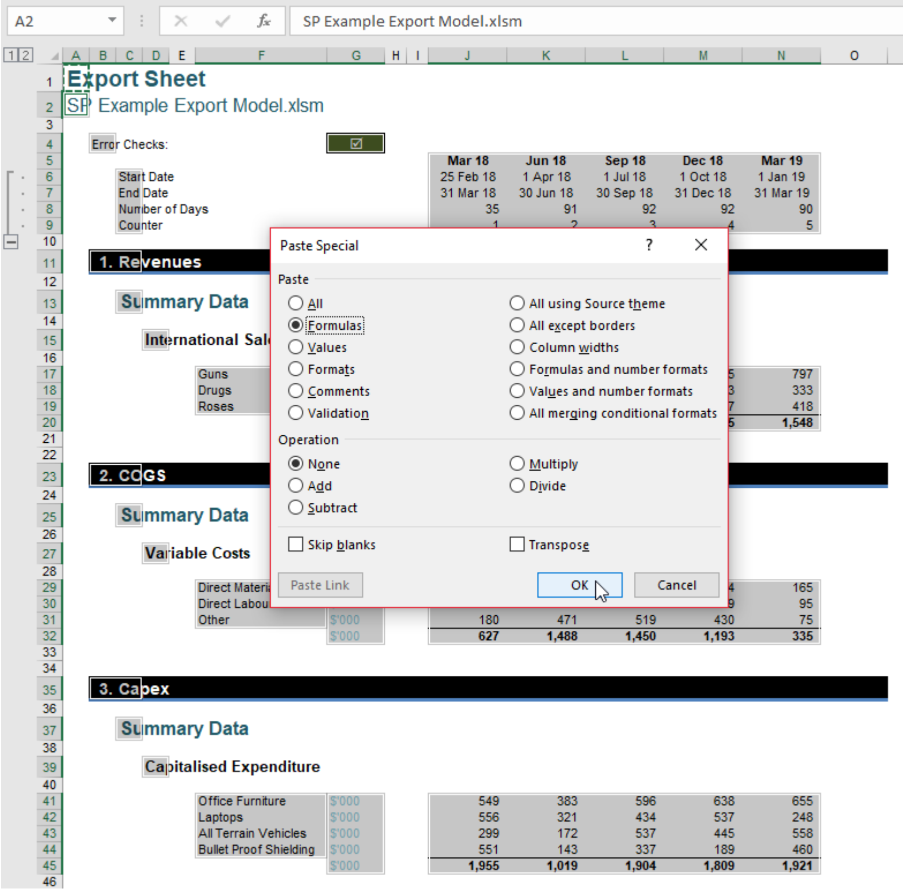
3. Capex

Summary Data

Capitalised Expenditure

Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	1,955	1,019	1,904	1,809	1,921

Note cell **A1** is not selected (it's a formula!). Now, just paste special as formulas *(sic)* (**ALT + E + S + F + ENTER**):



A2 | SP Example Export Model.xlsm

Export Sheet

SP Example Export Model.xlsm

Paste Special

Paste

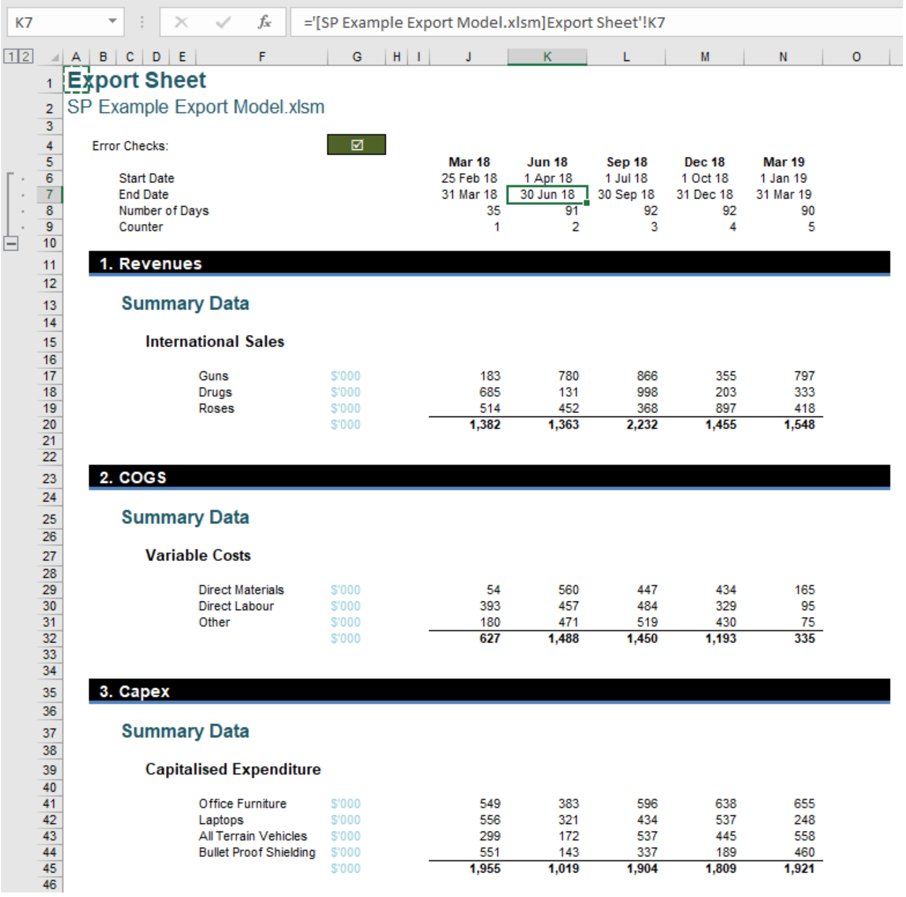
- ○ All
- ● Formulas
- ○ Values
- ○ Formats
- ○ Comments
- ○ Validation
- ○ All using Source theme
- ○ All except borders
- ○ Column widths
- ○ Formulas and number formats
- ○ Values and number formats
- ○ All merging conditional formats

Operation

- ● None
- ○ Add
- ○ Subtract
- ○ Multiply
- ○ Divide

□ Skip blanks □ Transpose

We now have an import sheet where all non-blank cells link back to the export worksheet:



K7 | ='[SP Example Export Model.xlsm]Export Sheet'!K7

Export Sheet

SP Example Export Model.xlsm

Error Checks: ☑

		Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
Number of Days		35	91	92	92	90
Counter		1	2	3	4	5

1. Revenues

Summary Data

International Sales

Guns	$'000	183	780	866	355	797
Drugs	$'000	685	131	998	203	333
Roses	$'000	514	452	368	897	418
	$'000	1,382	1,363	2,232	1,455	1,548

2. COGS

Summary Data

Variable Costs

Direct Materials	$'000	54	560	447	434	165
Direct Labour	$'000	393	457	484	329	95
Other	$'000	180	471	519	430	75
	$'000	627	1,488	1,450	1,193	335

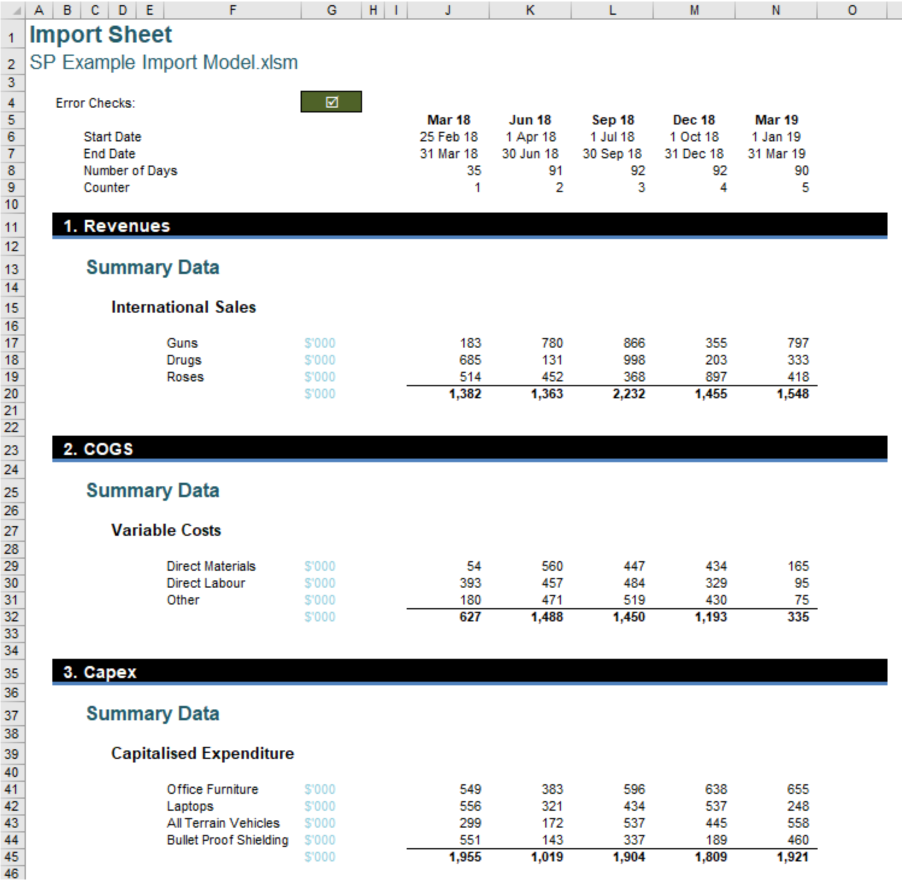
3. Capex

Summary Data

Capitalised Expenditure

Office Furniture	$'000	549	383	596	638	655
Laptops	$'000	556	321	434	537	248
All Terrain Vehicles	$'000	299	172	537	445	558
Bullet Proof Shielding	$'000	551	143	337	189	460
	$'000	1,955	1,019	1,904	1,809	1,921

All you have to do now is correct cells **A1** and **A2**, which need to say, “Import Sheet” and cite the filename of the destination workbook respectively.



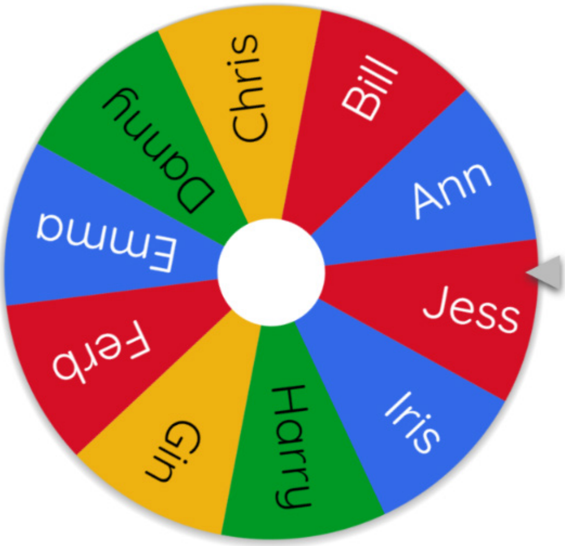
	A–F	G	J	K	L	M	N
1	Import Sheet						
2	SP Example Import Model.xlsm						
3							
4	Error Checks:	☑					
5			Mar 18	Jun 18	Sep 18	Dec 18	Mar 19
6	Start Date		25 Feb 18	1 Apr 18	1 Jul 18	1 Oct 18	1 Jan 19
7	End Date		31 Mar 18	30 Jun 18	30 Sep 18	31 Dec 18	31 Mar 19
8	Number of Days		35	91	92	92	90
9	Counter		1	2	3	4	5
10							
11	1. Revenues						
12							
13	Summary Data						
14							
15	International Sales						
16							
17	Guns	$'000	183	780	866	355	797
18	Drugs	$'000	685	131	998	203	333
19	Roses	$'000	514	452	368	897	418
20		$'000	1,382	1,363	2,232	1,455	1,548
21							
22							
23	2. COGS						
24							
25	Summary Data						
26							
27	Variable Costs						
28							
29	Direct Materials	$'000	54	560	447	434	165
30	Direct Labour	$'000	393	457	484	329	95
31	Other	$'000	180	471	519	430	75
32		$'000	627	1,488	1,450	1,193	335
33							
34							
35	3. Capex						
36							
37	Summary Data						
38							
39	Capitalised Expenditure						
40							
41	Office Furniture	$'000	549	383	596	638	655
42	Laptops	$'000	556	321	434	537	248
43	All Terrain Vehicles	$'000	299	172	537	445	558
44	Bullet Proof Shielding	$'000	551	143	337	189	460
45		$'000	1,955	1,019	1,904	1,809	1,921
46							

If you have more than one link, this process will need to be repeated for each import / export pair. Don’t try and have just one import or export worksheet – it gets too confusing.

This may seem like an over-the-top solution to a common problem, but with practice, this process takes only seconds to perform and can save hours later as you go seeking those pesky links when you have forgotten where they might be.

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…*



If you host prize draws on your website, social media, or live meetings, you may need to demonstrate a fair way to select random winner(s).

It is easy to randomly select just one winner as a suggested solution is simple as follows:

=INDEX(List[Name], RANDBETWEEN(1, ROWS(List[Name])))

ROWS counts number of names in the list. **RANDBETWEEN** will generate a random integer (*i.e.* position number) between one [1] and the number of names. Then, **INDEX** will help us pick the name at that position.

In this challenge, the objective is to create a random selector in Excel to choose three winners for a prize draw from a list of 10 people. The tricky part here is that a person cannot win the game more than once. Hence, the winners list must not contain any duplicates.

As always, there are some requirements:

- this is a formula challenge; no Power Query / Get & Transform or VBA!
- there must be no duplicate in the result list
- if we change the number of winners, the formula should still work.

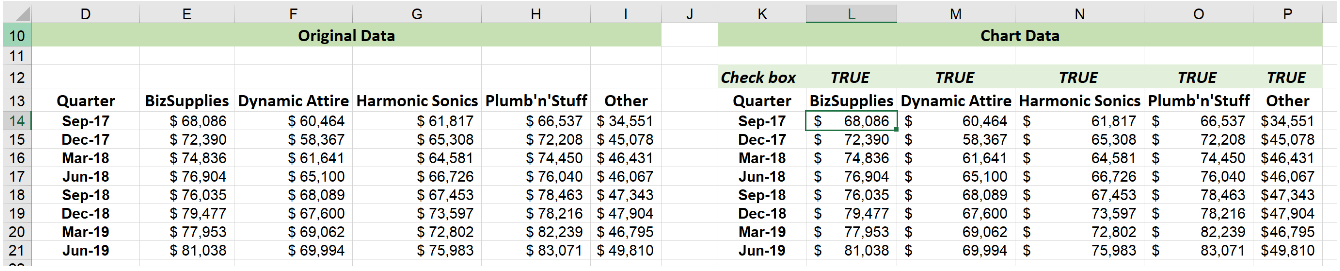
Sounds easy? Try it. One solution *just might* be found later in this newsletter – but no reading ahead!

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 start to look at interactive charts by considering Form Controls.

Excel offers a lot of options to interact with charts, one of which is Form Controls. With Form Controls, we can add a data selection option to charts, such as a Check Box or a Drop-down List.

Let's consider an example. To create Check Boxes for a chart, one option might be to create a linked duplicate table of the original data. We create a row for Check Box value **TRUE** or **FALSE**. In each cell of the data, we apply a condition that if the Check Box shows **FALSE**, it will return *#N/A* error, otherwise, it will return the original value. The reason for using the **NA()** function rather than returning a zero or blank cell is that cells with *#N/A* will not be displayed in line charts, while cells with zero or blank will still be shown, as a straight line at the bottom of the chart. However, for a column chart, this would not have the same impact.

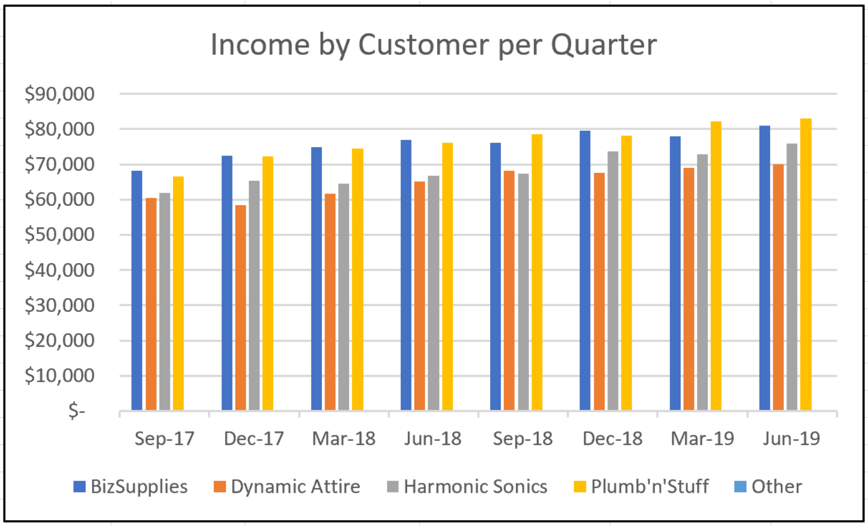


	D	E	F	G	H	I	J	K	L	M	N	O	P
10			Original Data							Chart Data			
11													
12								*Check box*	*TRUE*	*TRUE*	*TRUE*	*TRUE*	*TRUE*
13	Quarter	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff	Other		Quarter	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff	Other
14	Sep-17	$ 68,086	$ 60,464	$ 61,817	$ 66,537	$ 34,551		Sep-17	$ 68,086	$ 60,464	$ 61,817	$ 66,537	$34,551
15	Dec-17	$ 72,390	$ 58,367	$ 65,308	$ 72,208	$ 45,078		Dec-17	$ 72,390	$ 58,367	$ 65,308	$ 72,208	$45,078
16	Mar-18	$ 74,836	$ 61,641	$ 64,581	$ 74,450	$ 46,431		Mar-18	$ 74,836	$ 61,641	$ 64,581	$ 74,450	$46,431
17	Jun-18	$ 76,904	$ 65,100	$ 66,726	$ 76,040	$ 46,067		Jun-18	$ 76,904	$ 65,100	$ 66,726	$ 76,040	$46,067
18	Sep-18	$ 76,035	$ 68,089	$ 67,453	$ 78,463	$ 47,343		Sep-18	$ 76,035	$ 68,089	$ 67,453	$ 78,463	$47,343
19	Dec-18	$ 79,477	$ 67,600	$ 73,597	$ 78,216	$ 47,904		Dec-18	$ 79,477	$ 67,600	$ 73,597	$ 78,216	$47,904
20	Mar-19	$ 77,953	$ 69,062	$ 72,802	$ 82,239	$ 46,795		Mar-19	$ 77,953	$ 69,062	$ 72,802	$ 82,239	$46,795
21	Jun-19	$ 81,038	$ 69,994	$ 75,983	$ 83,071	$ 49,810		Jun-19	$ 81,038	$ 69,994	$ 75,983	$ 83,071	$49,810

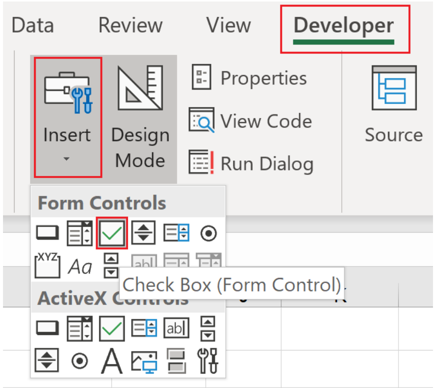
A suitable formula for cell **L14** might be:

=IF(L$12,E14,NA())

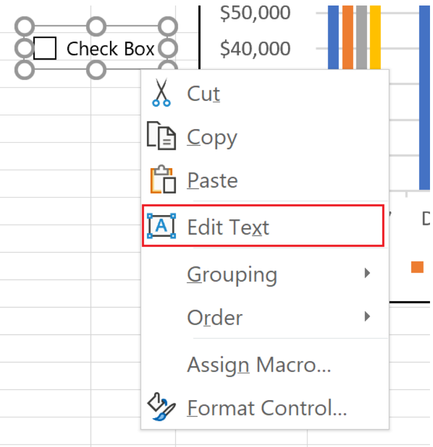
Next, we will insert a column chart based on the Chart Data:



In order to create Check Boxes for this chart, go to the Developer tab on the Ribbon (you may need to display it first using **File -> Options -> Customize Ribbon**), and below the Insert menu, choose the 'Check Box' icon:



We will drag and drop the icon on to the worksheet. We may rename this Check Box by right-clicking on the Check Box and choosing 'Edit Text':



Now, we have to set up a control by right-clicking on the Check Box, choosing 'Format Control'. A dialog will appear. On the final tab, Control, choose Checked, and choose 'Cell link' at the Check Box row according to the cell being referred to (*e.g.* for BizSupplies, we would select **L12**). The Check Box returns **TRUE** in the 'Cell Link' cell when checked, and **FALSE** when unchecked.

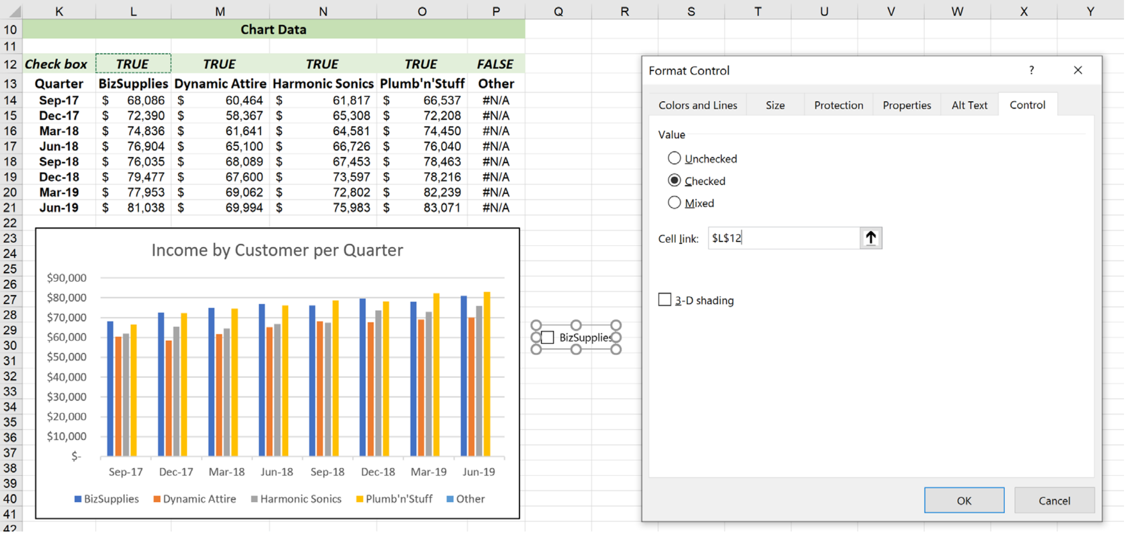


Chart Data

Check box	TRUE	TRUE	TRUE	TRUE	FALSE
Quarter	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff	Other
Sep-17	$ 68,086	$ 60,464	$ 61,817	$ 66,537	#N/A
Dec-17	$ 72,390	$ 58,367	$ 65,308	$ 72,208	#N/A
Mar-18	$ 74,836	$ 61,641	$ 64,581	$ 74,450	#N/A
Jun-18	$ 76,904	$ 65,100	$ 66,726	$ 76,040	#N/A
Sep-18	$ 76,035	$ 68,089	$ 67,453	$ 78,463	#N/A
Dec-18	$ 79,477	$ 67,600	$ 73,597	$ 78,216	#N/A
Mar-19	$ 77,953	$ 69,062	$ 72,802	$ 82,239	#N/A
Jun-19	$ 81,038	$ 69,994	$ 75,983	$ 83,071	#N/A

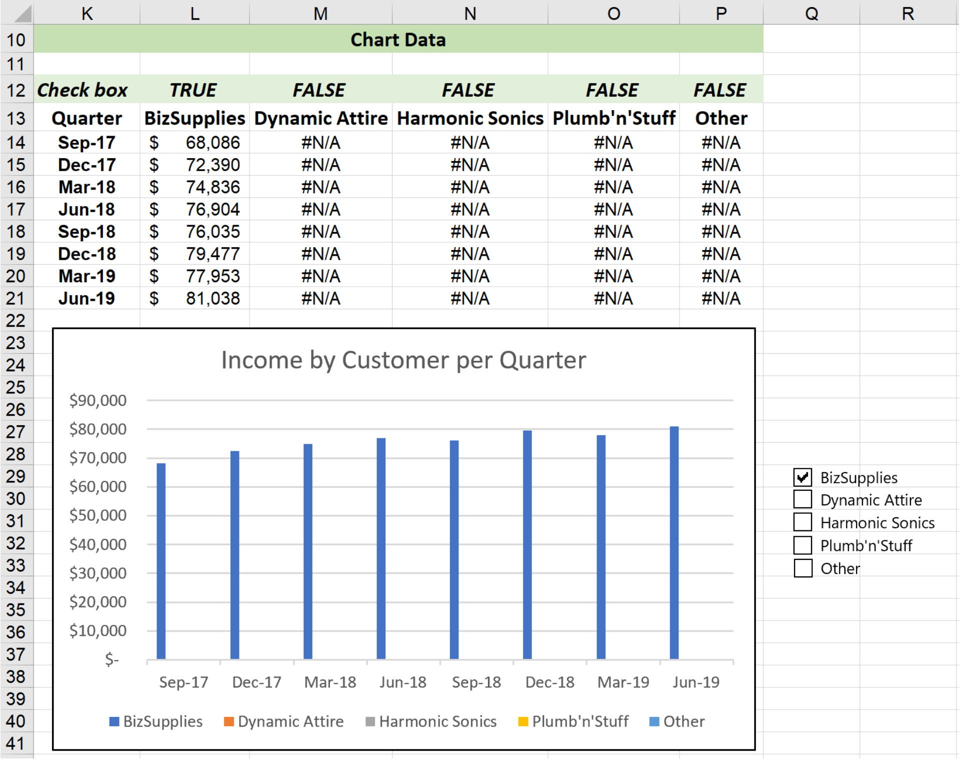
Format Control

Control tab — Value: ○ Unchecked ● Checked ○ Mixed

Cell link: $L$12

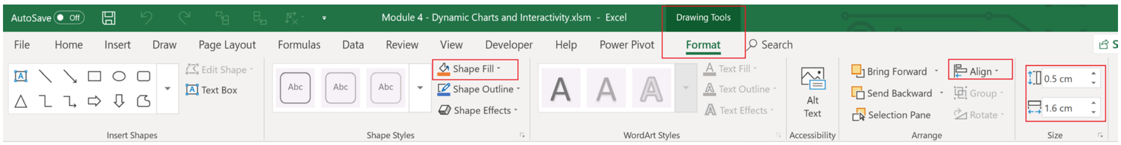
□ 3-D shading

We will do the same for the rest of the Check Boxes. After inserting Check Boxes for all of the fields, we may perform a test. Below, if we check the BizSupplies box only, in the chart, only the BizSupplies data will be shown. In the Chart Data, the Check Box cell for BizSupplies will also show TRUE. The remaining references will still display **FALSE**. As a consequence, their corresponding data will be displayed as *#N/A*, meaning there will be no corresponding lines or columns shown on the chart. Whilst it is true that zero [0] would have the same effect here, this would not be the case for all chart types.

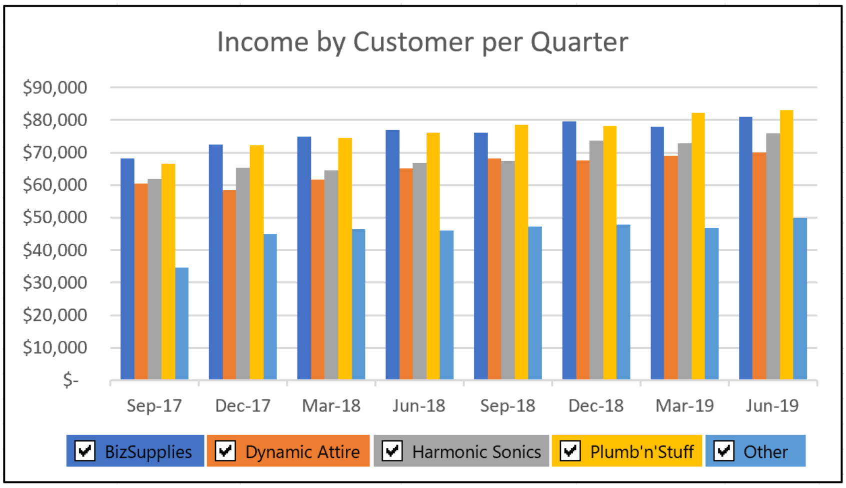


	K	L	M	N	O	P
10	Chart Data					
11						
12	*Check box*	*TRUE*	*FALSE*	*FALSE*	*FALSE*	*FALSE*
13	**Quarter**	**BizSupplies**	**Dynamic Attire**	**Harmonic Sonics**	**Plumb'n'Stuff**	**Other**
14	**Sep-17**	$ 68,086	#N/A	#N/A	#N/A	#N/A
15	**Dec-17**	$ 72,390	#N/A	#N/A	#N/A	#N/A
16	**Mar-18**	$ 74,836	#N/A	#N/A	#N/A	#N/A
17	**Jun-18**	$ 76,904	#N/A	#N/A	#N/A	#N/A
18	**Sep-18**	$ 76,035	#N/A	#N/A	#N/A	#N/A
19	**Dec-18**	$ 79,477	#N/A	#N/A	#N/A	#N/A
20	**Mar-19**	$ 77,953	#N/A	#N/A	#N/A	#N/A
21	**Jun-19**	$ 81,038	#N/A	#N/A	#N/A	#N/A

Now, we need to create the Check Boxes on the chart legend. Right-click on the Check Boxes and a 'Drawing Tools, Format' tab will appear on the Ribbon. Here, we may choose the shape fills, outlines and effects as we wish.



Particularly in this example, we choose the shape fill with matching colour for each Check Box (one tip is that you can apply RGB [red, green, blue] number so that the colours are exactly matched). Then we drag the Check Box to overwrite the existing legends, then adjust their height and alignment, so that the interaction looks much better:



More next month…

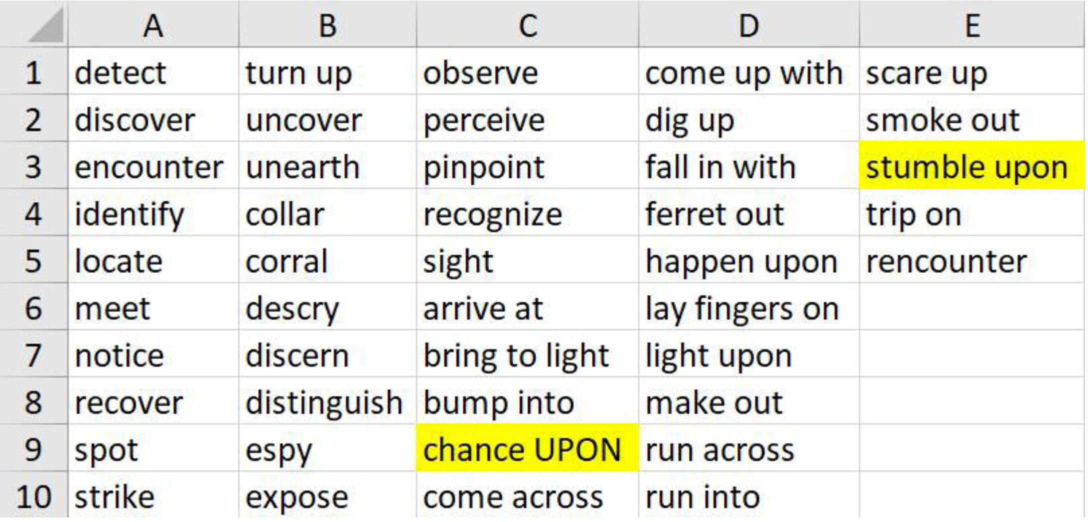
Visual Basics

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

Last month, we used the **LookAt** parameter with the **Find** method to match the complete word. Of the three methods of matching words listed in the previous newsletter, this time out we tackle the second method, matching the case of the word, *viz.*

1. Matching the complete word
2. **Matching the case of the word (*i.e.* the capitalisation)**
3. Matching a specific cell formatting.

Similar to last time, we are going to be matching words found in this field.



	A	B	C	D	E
1	detect	turn up	observe	come up with	scare up
2	discover	uncover	perceive	dig up	smoke out
3	encounter	unearth	pinpoint	fall in with	stumble upon
4	identify	collar	recognize	ferret out	trip on
5	locate	corral	sight	happen upon	rencounter
6	meet	descry	arrive at	lay fingers on	
7	notice	discern	bring to light	light upon	
8	recover	distinguish	bump into	make out	
9	spot	espy	chance UPON	run across	
10	strike	expose	come across	run into	

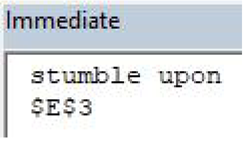
We want it to return with the cell **C9** "chance UPON". Let's search for "UPON" from cell **A1** onwards:

```
Sub FindMatchCase()

    Dim searchRange As Range
    Set searchRange = Range("A1:E10")

    Dim foundrange As Range
    Set foundrange = searchRange.Find("UPON", LookAt:=xlPart)

    If foundrange Is Nothing Then
        Debug.Print "not found!"
    Else
        Debug.Print foundrange
        Debug.Print foundrange.Address
    End If
End Sub
```



```
stumble upon
$E$3
```

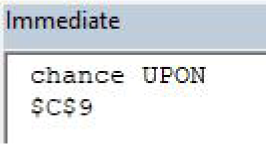
It found 'stumble upon' from cell **E3**. This suggests that we have to add another parameter, **MatchCase**. The **MatchCase** parameter will force VBA to match the case of the word. Let's search of "UPON" again from cell **A1**:

```
Sub FindMatchCase()

    Dim searchRange As Range
    Set searchRange = Range("A1:E10")

    Dim foundrange As Range
    Set foundrange = searchRange.Find("UPON", LookAt:=xlPart, MatchCase:=True)

    If foundrange Is Nothing Then
        Debug.Print "not found!"
    Else
        Debug.Print foundrange
        Debug.Print foundrange.Address
    End If
End Sub
```



```
chance UPON
$C$9
```

This change in the code will give us the expected result of 'chance UPON' in cell **C9**.

A noteworthy part of the code to point out is that we are setting the **LookAt** parameter to 'xlPart' and not 'xlWhole'. Therefore, if we wish to find the entire word we would have to change 'xlPart' to 'xlWhole'.

Next time, we shall turn our attention to matching specific cell formatting.

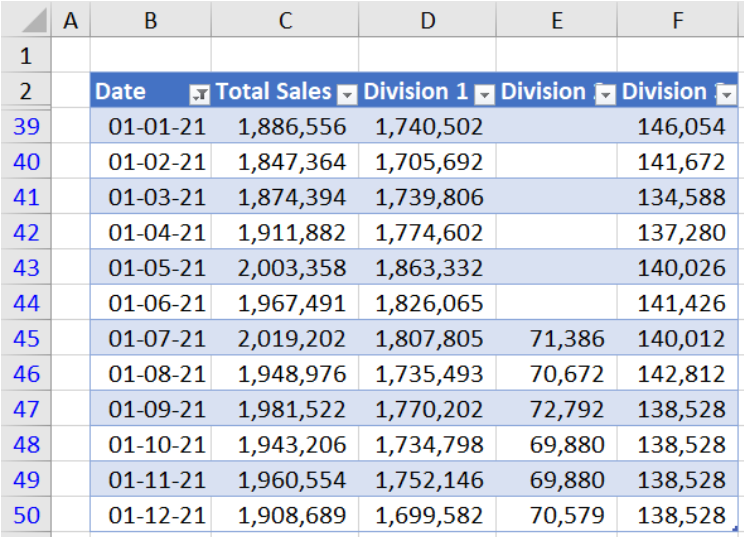
Until then.

Power Pivot Principles

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

A while ago, we ran into the issue of division by zero errors in Power Pivot. We dealt with that issue with a combination of the **IF** and **BLANK** functions to error trap the division by zero errors.

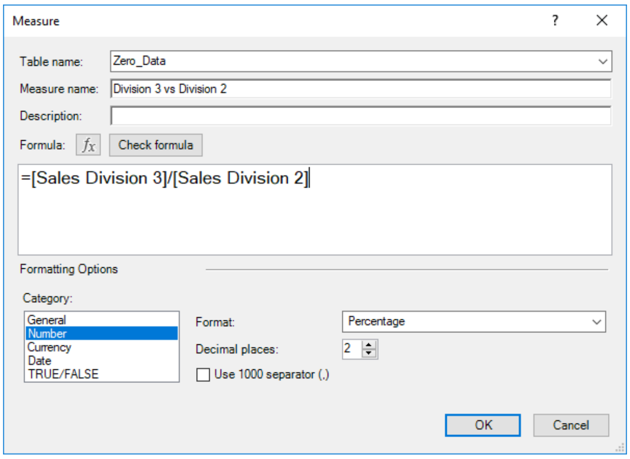
As a quick and dirty example, let's use the following data:



	A	B	C	D	E	F
1						
2		Date	Total Sales	Division 1	Division	Division
39		01-01-21	1,886,556	1,740,502		146,054
40		01-02-21	1,847,364	1,705,692		141,672
41		01-03-21	1,874,394	1,739,806		134,588
42		01-04-21	1,911,882	1,774,602		137,280
43		01-05-21	2,003,358	1,863,332		140,026
44		01-06-21	1,967,491	1,826,065		141,426
45		01-07-21	2,019,202	1,807,805	71,386	140,012
46		01-08-21	1,948,976	1,735,493	70,672	142,812
47		01-09-21	1,981,522	1,770,202	72,792	138,528
48		01-10-21	1,943,206	1,734,798	69,880	138,528
49		01-11-21	1,960,554	1,752,146	69,880	138,528
50		01-12-21	1,908,689	1,699,582	70,579	138,528

Say we wanted to calculate the percentage difference between the sales in Division 2 and Division 3 using the following measure (assuming that we have aggregation measures for the sales of both divisions):

=[Sales Division 3]/[Sales Division 2]



Measure

Table name: Zero_Data

Measure name: Division 3 vs Division 2

Description:

Formula:

```
=[Sales Division 3]/[Sales Division 2]
```

Formatting Options

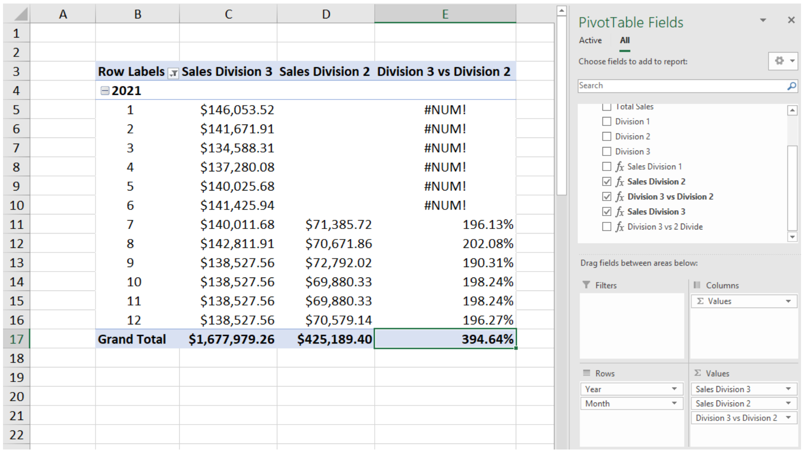
Category: General, **Number**, Currency, Date, TRUE/FALSE

Format: Percentage

Decimal places: 2

□ Use 1000 separator (,)

We would get the following PivotTable output:



	A	B	C	D	E
1					
2					
3		**Row Labels**	**Sales Division 3**	**Sales Division 2**	**Division 3 vs Division 2**
4		⊟**2021**			
5		1	$146,053.52		#NUM!
6		2	$141,671.91		#NUM!
7		3	$134,588.31		#NUM!
8		4	$137,280.08		#NUM!
9		5	$140,025.68		#NUM!
10		6	$141,425.94		#NUM!
11		7	$140,011.68	$71,385.72	196.13%
12		8	$142,811.91	$70,671.86	202.08%
13		9	$138,527.56	$72,792.02	190.31%
14		10	$138,527.56	$69,880.33	198.24%
15		11	$138,527.56	$69,880.33	198.24%
16		12	$138,527.56	$70,579.14	196.27%
17		**Grand Total**	**$1,677,979.26**	**$425,189.40**	**394.64%**
18					
19					
20					
21					
22					

PivotTable Fields

Active **All**

Choose fields to add to report:

Search

□ Total Sales
□ Division 1
□ Division 2
□ Division 3
□ fx Sales Division 1
☑ fx **Sales Division 2**
☑ fx **Division 3 vs Division 2**
☑ fx **Sales Division 3**
□ fx Division 3 vs 2 Divide

Drag fields between areas below:

Filters:

Columns: Σ Values

Rows: Year, Month

Values: Sales Division 3, Sales Division 2, Division 3 vs Division 2

The nasty *#NUM!* error is caused by the denominator, **[Sales Division 2]**, being zero [0] for those time periods. Rather than use the **BLANK** and **IF** functions to trap these errors, we may use the **DIVIDE** function instead.

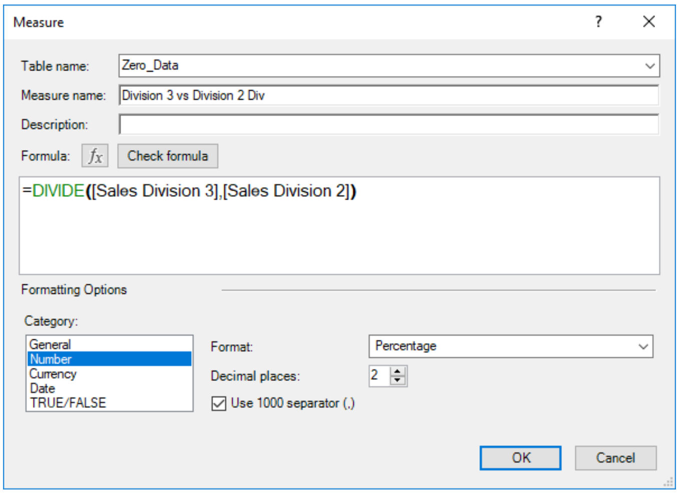
The **DIVIDE** function uses the following syntax to operate:

DIVIDE(numerator, denominator, [alternateresult])

The **[alternateresult]** is an optional argument which the **DIVIDE** function will return when the denominator is zero [0]. If no value is specified, the default setting is to return with **BLANK**.

Let's create a new measure using the **DIVIDE** function:

=DIVIDE([Sales Division 3],[Sales Division 2])



Measure

Table name: Zero_Data

Measure name: Division 3 vs Division 2 Div

Description:

Formula:

```
=DIVIDE([Sales Division 3],[Sales Division 2])
```

Formatting Options

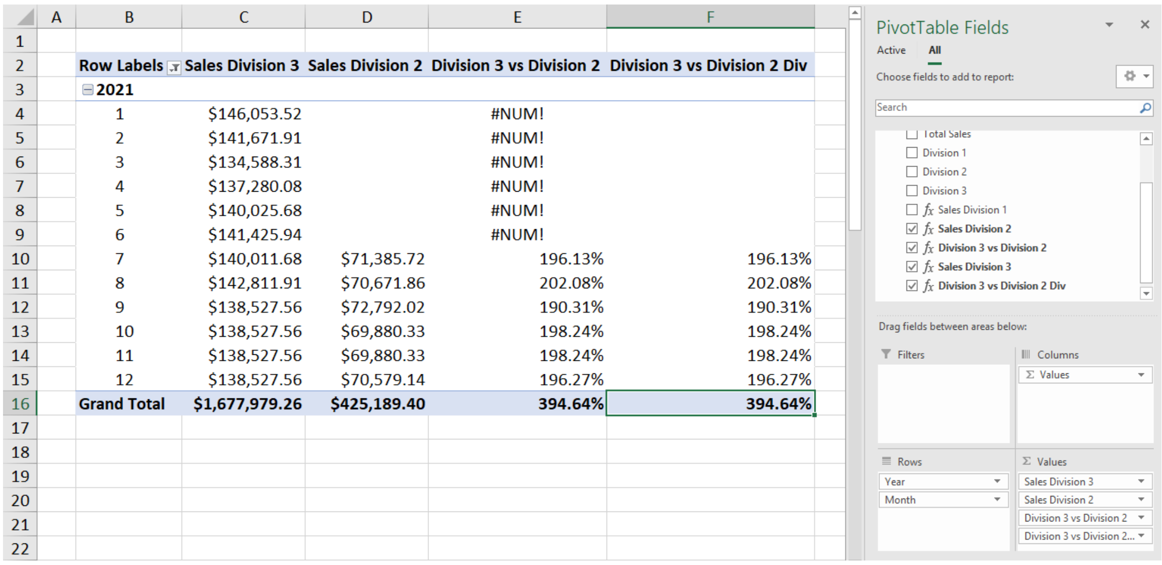
Category: General, **Number**, Currency, Date, TRUE/FALSE

Format: Percentage

Decimal places: 2

☑ Use 1000 separator (,)

This results in the following output from our PivotTable:



Row Labels	Sales Division 3	Sales Division 2	Division 3 vs Division 2	Division 3 vs Division 2 Div
2021				
1	$146,053.52		#NUM!	
2	$141,671.91		#NUM!	
3	$134,588.31		#NUM!	
4	$137,280.08		#NUM!	
5	$140,025.68		#NUM!	
6	$141,425.94		#NUM!	
7	$140,011.68	$71,385.72	196.13%	196.13%
8	$142,811.91	$70,671.86	202.08%	202.08%
9	$138,527.56	$72,792.02	190.31%	190.31%
10	$138,527.56	$69,880.33	198.24%	198.24%
11	$138,527.56	$69,880.33	198.24%	198.24%
12	$138,527.56	$70,579.14	196.27%	196.27%
Grand Total	$1,677,979.26	$425,189.40	394.64%	394.64%

There we go: the **DIVIDE** function has done our error trapping for us!

More *Power Pivot Principles* next month.

Power Query Pointers

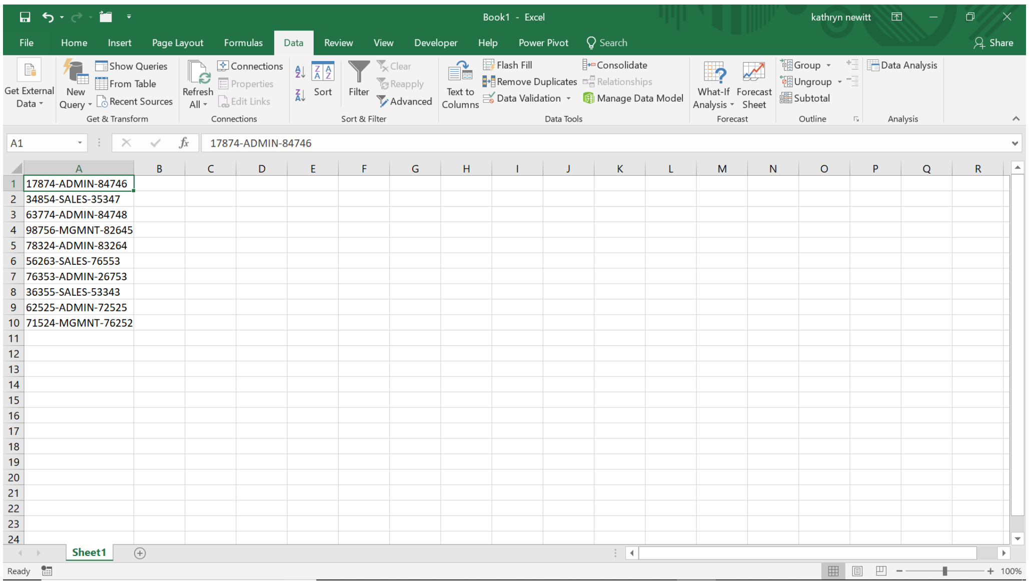
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog*. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at some useful extraction*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

In our final look at text functions, we will describe some **Text()** functions that extract part of a string. In this way, we may create useful new columns that can help us to link our data to other tables. As we did for Boolean and transforming functions, we will give an example for each one.

Text.Middle

Text.Middle(text as nullable text, **start** as number, optional **count** as nullable number**)** as nullable text

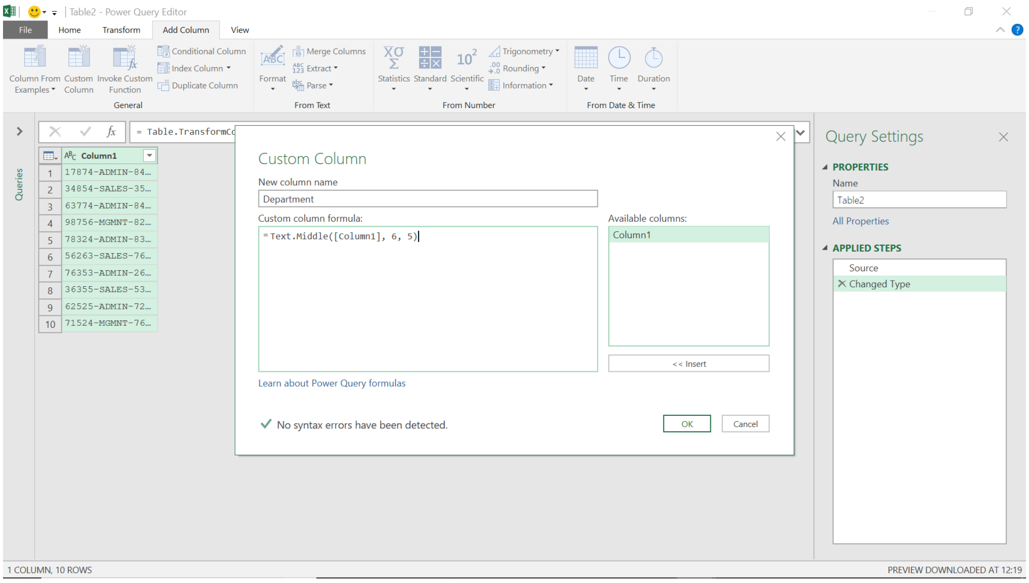
This returns **count** characters, or through the end of **text**; at the offset **start**, and may be used to extract a consecutive portion of a text string. This is particularly useful for extracting information from references; for example, the department from a **Product ID**. The following screenshot shows some references that we are going to extract the **Department** from:



A1 17874-ADMIN-84746

	A
1	17874-ADMIN-84746
2	34854-SALES-35347
3	63774-ADMIN-84748
4	98756-MGMNT-82645
5	78324-ADMIN-83264
6	56263-SALES-76553
7	76353-ADMIN-26753
8	36355-SALES-53343
9	62525-ADMIN-72525
10	71524-MGMNT-76252

We use the ‘From Table’ option on the ‘Get and Transform’ section of the ‘Data’ tab. We can now use the **Text.Middle()** formula to extract a new column from our original data



```
= Table.TransformCo
```

	Column1
1	17874-ADMIN-84…
2	34854-SALES-35…
3	63774-ADMIN-84…
4	98756-MGMNT-82…
5	78324-ADMIN-83…
6	56263-SALES-76…
7	76353-ADMIN-26…
8	36355-SALES-53…
9	62525-ADMIN-72…
10	71524-MGMNT-76…

Custom Column

New column name: Department

Custom column formula:

```
= Text.Middle([Column1], 6, 5)
```

Available columns: Column1

✓ No syntax errors have been detected.

Query Settings

PROPERTIES

Name: Table2

APPLIED STEPS

- Source
- Changed Type

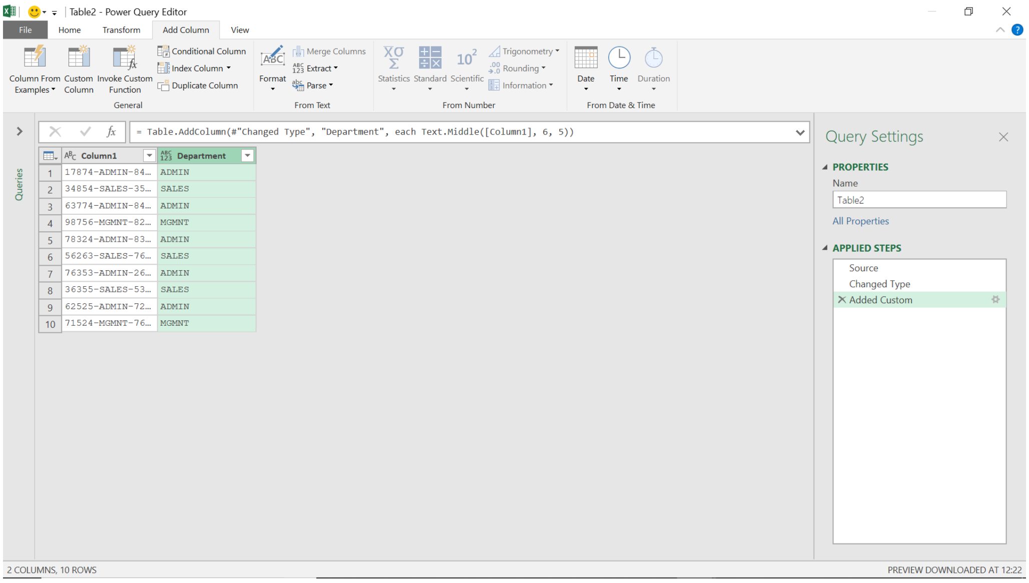
1 COLUMN, 10 ROWS

PREVIEW DOWNLOADED AT 12:19

The **M** formula we have used is

= Text.Middle([Column1], 6, 5)

This tells Power Query to start at position six [6] (counting from position zero [0]) and select the next five [5] characters.



```
= Table.AddColumn(#"Changed Type", "Department", each Text.Middle([Column1], 6, 5))
```

	Column1	Department
1	17874-ADMIN-84…	ADMIN
2	34854-SALES-35…	SALES
3	63774-ADMIN-84…	ADMIN
4	98756-MGMNT-82…	MGMNT
5	78324-ADMIN-83…	ADMIN
6	56263-SALES-76…	SALES
7	76353-ADMIN-26…	ADMIN
8	36355-SALES-53…	SALES
9	62525-ADMIN-72…	ADMIN
10	71524-MGMNT-76…	MGMNT

Query Settings

PROPERTIES

Name: Table2

APPLIED STEPS

- Source
- Changed Type
- Added Custom

2 COLUMNS, 10 ROWS

PREVIEW DOWNLOADED AT 12:22

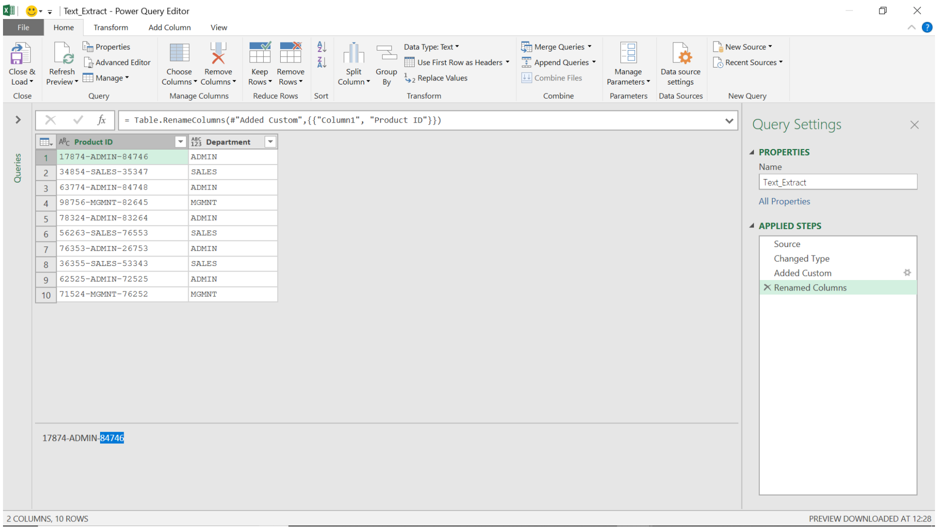
This selects the department, and we may link to other data which uses the same department codes.

Text.AfterDelimiter

Text.AfterDelimiter(text as nullable text, **delimiter** as text, optional **index** as any**)** as any

This returns the portion of **text** after the specified **delimiter**. An optional numerical **index** indicates which occurrence of the **delimiter** should be considered.

We may use this function to extract the second serial number from the **Product ID**.



Text_Extract - Power Query Editor

```
= Table.RenameColumns(#"Added Custom",{{"Column1", "Product ID"}})
```

	Product ID	Department
1	17874-ADMIN-84746	ADMIN
2	34854-SALES-35347	SALES
3	63774-ADMIN-84748	ADMIN
4	98756-MGMNT-82645	MGMNT
5	78324-ADMIN-83264	ADMIN
6	56263-SALES-76553	SALES
7	76353-ADMIN-26753	ADMIN
8	36355-SALES-53343	SALES
9	62525-ADMIN-72525	ADMIN
10	71524-MGMNT-76252	MGMNT

17874-ADMIN-84746

Query Settings

PROPERTIES

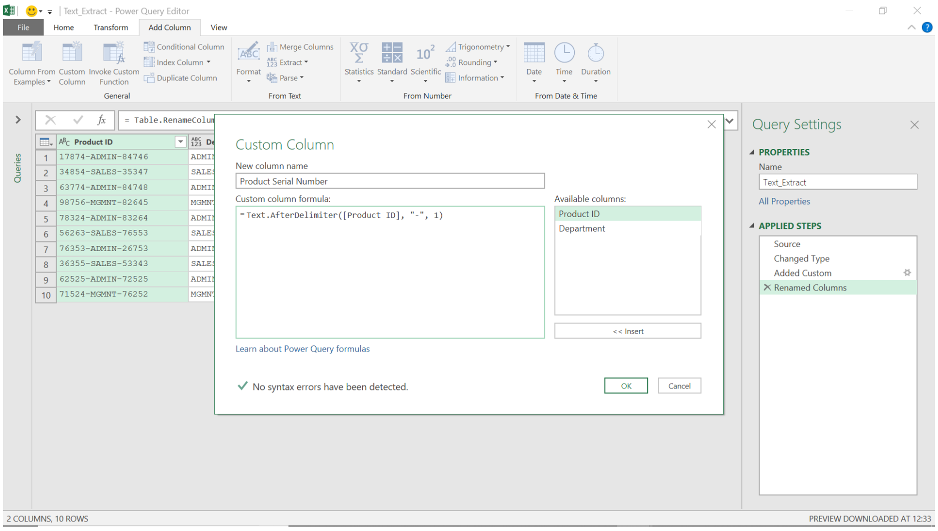
Name: Text_Extract

APPLIED STEPS

- Source
- Changed Type
- Added Custom
- Renamed Columns

2 COLUMNS, 10 ROWS — PREVIEW DOWNLOADED AT 12:28

We can create a new column using **Text.AfterDelimiter()**:



Custom Column

New column name: Product Serial Number

Custom column formula:

```
= Text.AfterDelimiter([Product ID], "-", 1)
```

Available columns:

- Product ID
- Department

<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

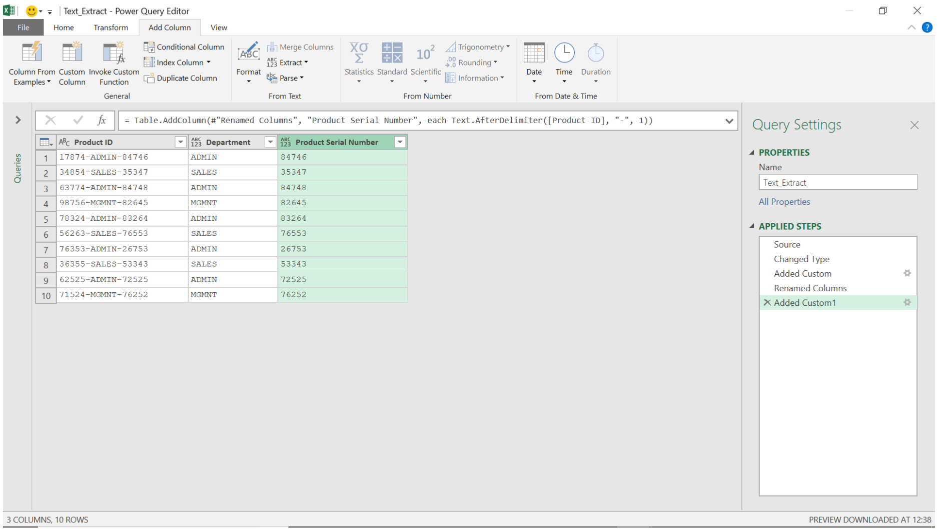
OK Cancel

2 COLUMNS, 10 ROWS — PREVIEW DOWNLOADED AT 12:33

The **M** formula is

= Text.AfterDelimiter([Product ID],"-", 1)

This means, select what comes after the second occurrence of “-” (since the first occurrence is counted as zero [0]).



Text_Extract - Power Query Editor

```
= Table.AddColumn(#"Renamed Columns", "Product Serial Number", each Text.AfterDelimiter([Product ID], "-", 1))
```

	Product ID	Department	Product Serial Number
1	17874-ADMIN-84746	ADMIN	84746
2	34854-SALES-35347	SALES	35347
3	63774-ADMIN-84748	ADMIN	84748
4	98756-MGMNT-82645	MGMNT	82645
5	78324-ADMIN-83264	ADMIN	83264
6	56263-SALES-76553	SALES	76553
7	76353-ADMIN-26753	ADMIN	26753
8	36355-SALES-53343	SALES	53343
9	62525-ADMIN-72525	ADMIN	72525
10	71524-MGMNT-76252	MGMNT	76252

APPLIED STEPS

- Source
- Changed Type
- Added Custom
- Renamed Columns
- Added Custom1

3 COLUMNS, 10 ROWS — PREVIEW DOWNLOADED AT 12:38

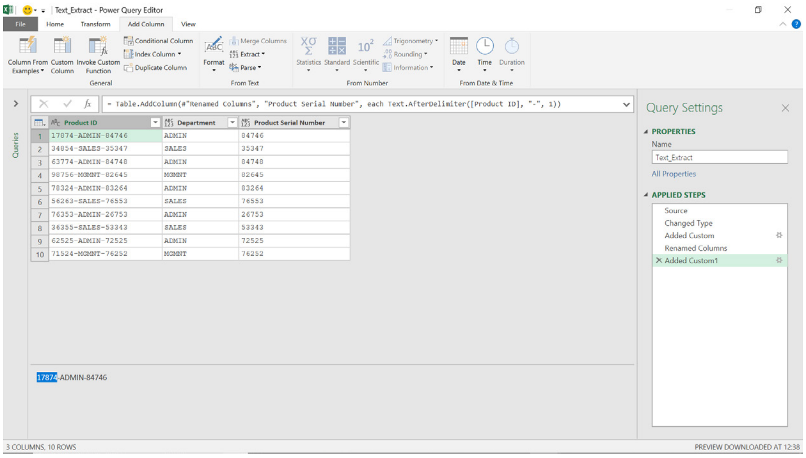
We have our **Product Serial Number**. In order to get the first number, the nominal code, we will need another function, which is coming up next.

Text.BeforeDelimiter

Text.BeforeDelimiter(text as nullable text, **delimiter** as text, optional **index** as any**)** as any

This returns the portion of **text** before the specified **delimiter**. An optional numerical **index** indicates which occurrence of the **delimiter** should be considered.

It’s very similar to the previous function: this time we get the **text** before the **delimiter**.



Text_Extract - Power Query Editor

```
= Table.AddColumn(#"Renamed Columns", "Product Serial Number", each Text.AfterDelimiter([Product ID], "-", 1))
```

	Product ID	Department	Product Serial Number
1	17874-ADMIN-84746	ADMIN	84746
2	34854-SALES-35347	SALES	35347
3	63774-ADMIN-84748	ADMIN	84748
4	98756-MGMNT-82645	MGMNT	82645
5	78324-ADMIN-83264	ADMIN	83264
6	56263-SALES-76553	SALES	76553
7	76353-ADMIN-26753	ADMIN	26753
8	36355-SALES-53343	SALES	53343
9	62525-ADMIN-72525	ADMIN	72525
10	71524-MGMNT-76252	MGMNT	76252

17874-ADMIN-84746

Query Settings

PROPERTIES

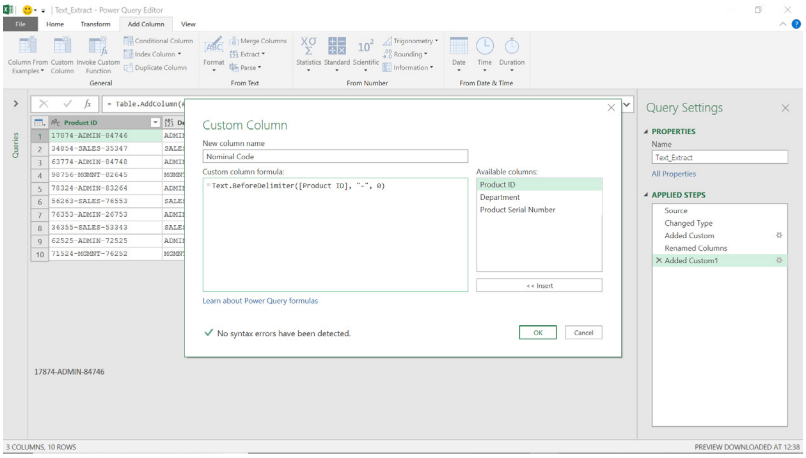
Name: Text_Extract

APPLIED STEPS: Source, Changed Type, Added Custom, Renamed Columns, Added Custom1

3 COLUMNS, 10 ROWS

PREVIEW DOWNLOADED AT 12:38

We add a column that uses the **M** function **Text.BeforeDelimiter()**.



Custom Column

New column name: Nominal Code

Custom column formula:

```
= Text.BeforeDelimiter([Product ID], "-", 0)
```

Available columns: Product ID, Department, Product Serial Number

No syntax errors have been detected.

17874-ADMIN-84746

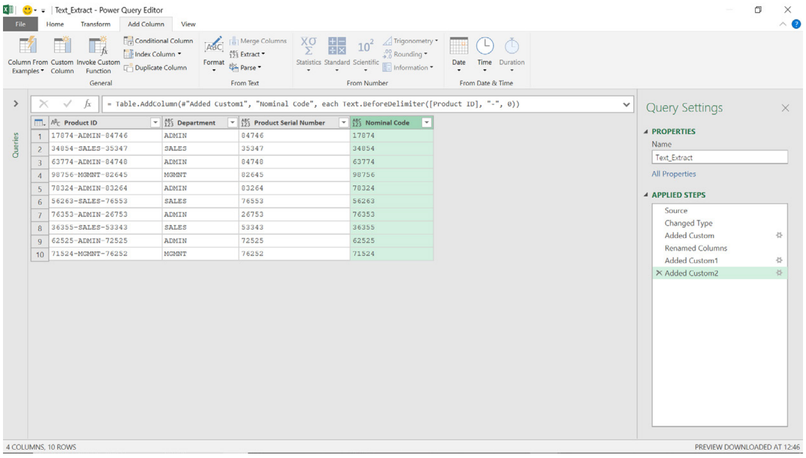
3 COLUMNS, 10 ROWS

PREVIEW DOWNLOADED AT 12:38

The **M** formula used is

= Text.BeforeDelimiter([Product ID], "-", 0)

This tells Power Query to get the **text** before the first occurrence of “-” (the first occurrence is counted as zero [0]).



Text_Extract - Power Query Editor

```
= Table.AddColumn(#"Added Custom1", "Nominal Code", each Text.BeforeDelimiter([Product ID], "-", 0))
```

	Product ID	Department	Product Serial Number	Nominal Code
1	17874-ADMIN-84746	ADMIN	84746	17874
2	34854-SALES-35347	SALES	35347	34854
3	63774-ADMIN-84748	ADMIN	84748	63774
4	98756-MGMNT-82645	MGMNT	82645	98756
5	78324-ADMIN-83264	ADMIN	83264	78324
6	56263-SALES-76553	SALES	76553	56263
7	76353-ADMIN-26753	ADMIN	26753	76353
8	36355-SALES-53343	SALES	53343	36355
9	62525-ADMIN-72525	ADMIN	72525	62525
10	71524-MGMNT-76252	MGMNT	76252	71524

Query Settings

PROPERTIES

Name: Text_Extract

APPLIED STEPS: Source, Changed Type, Added Custom, Renamed Columns, Added Custom1, Added Custom2

4 COLUMNS, 10 ROWS

PREVIEW DOWNLOADED AT 12:46

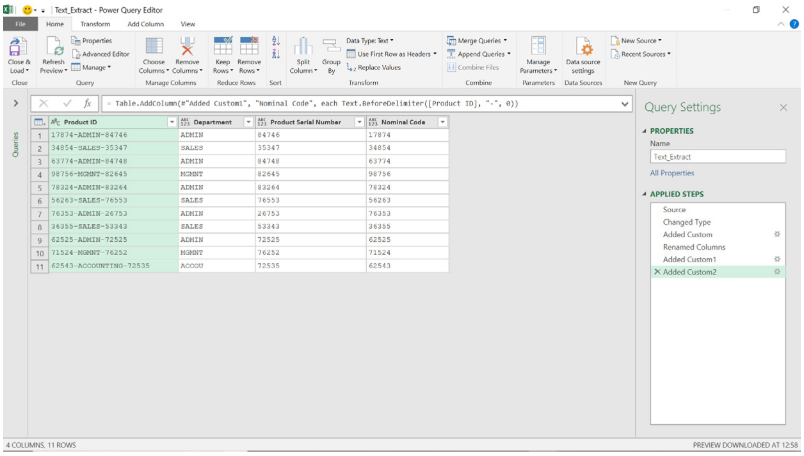
We have now extracted useful columns from the **Product ID**. The **Text.AfterDelimiter()** and **Text.BeforeDelimiter()** functions do not rely on the length of the **Product ID** segments being consistent, whereas the **Text.Middle()** function does. There is a way to extract the **Department** which does not assume it always occurs at the same position in the **text** – and that function is coming next.

Text.BetweenDelimiters

Text.BetweenDelimiters(text as nullable text, **startDelimiter** as text, **endDelimiter** as text, optional **startIndex** as any, optional **endIndex** as any**)** as any

This returns the portion of **text** between the specified **startDelimiter** and **endDelimiter**. An optional numerical **startIndex** indicates which occurrence of the **startDelimiter** should be considered and an optional numerical **endIndex** indicates which occurrence of the **endDelimiter** should be considered.

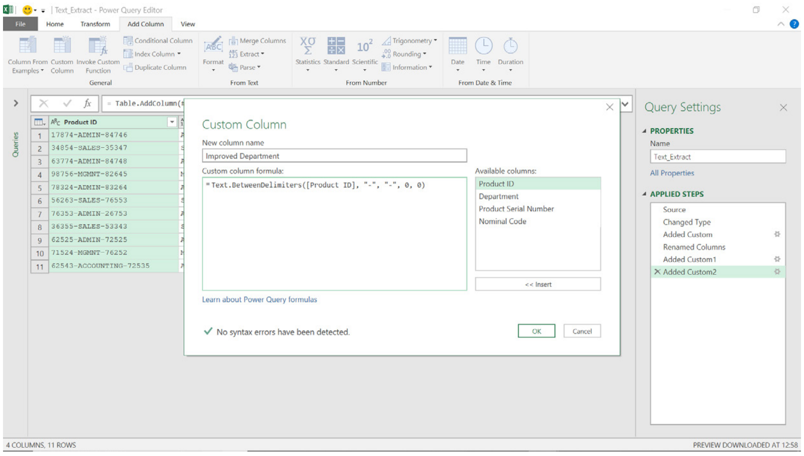
This is much more flexible than we need it to be for our example, because our delimiters are the same. This function would allow us to get the **text** between two different delimiters, *e.g.* "**-**" and "**;**", and specify the occurrence of each one. The occurrence for the **startDelimiter** is from the beginning, and the occurrence of the **endDelimiter** is from the end. We may use this function to get the **Department**, even if it is longer than the standard five [5] characters we have used. A new reference code has been added to demonstrate.



```
= Table.AddColumn(#"Added Custom1", "Nominal Code", each Text.BeforeDelimiter([Product ID], "-", 0))
```

	Product ID	Department	Product Serial Number	Nominal Code
1	17874-ADMIN-84746	ADMIN	84746	17874
2	34854-SALES-35347	SALES	35347	34854
3	63774-ADMIN-84748	ADMIN	84748	63774
4	98756-MGMNT-82645	MGMNT	82645	98756
5	78324-ADMIN-83264	ADMIN	83264	78324
6	56263-SALES-76553	SALES	76553	56263
7	76353-ADMIN-26753	ADMIN	26753	76353
8	36355-SALES-53343	SALES	53343	36355
9	62525-ADMIN-72525	ADMIN	72525	62525
10	71524-MGMNT-76252	MGMNT	76252	71524
11	62543-ACCOUNTING-72535	ACCOU	72535	62543

We use **Text.BetweenDelimiters()** to create a new column:



Custom Column

New column name: Improved Department

Custom column formula:

```
= Text.BetweenDelimiters([Product ID], "-", "-", 0, 0)
```

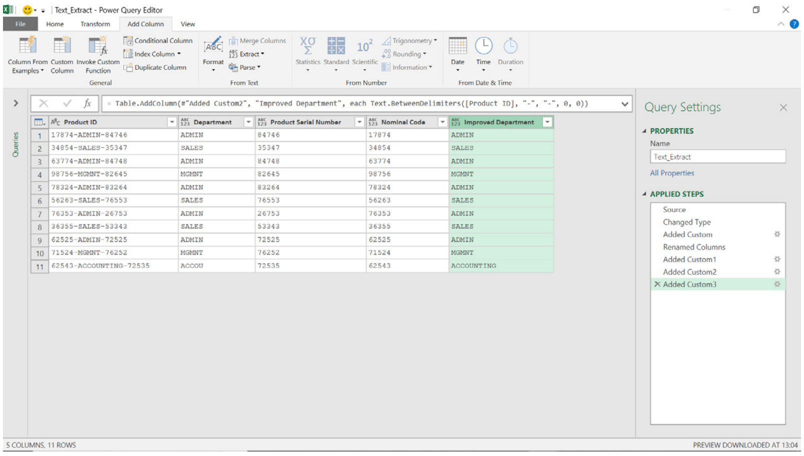
Available columns: Product ID, Department, Product Serial Number, Nominal Code

✓ No syntax errors have been detected.

The **M** formula used is

= Text.BetweenDelimiters([Product ID], "-", "-", 0, 0)

This will provide the **text** between the first "**-**" (count zero [0] from the beginning) and the second "**-**" (count zero [0] from the end).



```
= Table.AddColumn(#"Added Custom2", "Improved Department", each Text.BetweenDelimiters([Product ID], "-", "-", 0, 0))
```

	Product ID	Department	Product Serial Number	Nominal Code	Improved Department
1	17874-ADMIN-84746	ADMIN	84746	17874	ADMIN
2	34854-SALES-35347	SALES	35347	34854	SALES
3	63774-ADMIN-84748	ADMIN	84748	63774	ADMIN
4	98756-MGMNT-82645	MGMNT	82645	98756	MGMNT
5	78324-ADMIN-83264	ADMIN	83264	78324	ADMIN
6	56263-SALES-76553	SALES	76553	56263	SALES
7	76353-ADMIN-26753	ADMIN	26753	76353	ADMIN
8	36355-SALES-53343	SALES	53343	36355	SALES
9	62525-ADMIN-72525	ADMIN	72525	62525	ADMIN
10	71524-MGMNT-76252	MGMNT	76252	71524	MGMNT
11	62543-ACCOUNTING-72535	ACCOU	72535	62543	ACCOUNTING

We have all three of our segments extracted to separate columns so that we may analyse the data further.

More next time.

Power BI Updates

This latest set of updates see TypeScript support added in the developer playground sandbox and several new formatting features amongst other tweaks.

The full list is as follows:

Reporting

- Conditional formatting based on string fields
- Smart Narrative visual summary icon
- Formatting image width in table and matrix
- Update your base theme in Power BI
- Report theme validation on custom theme import
- Text box visual indentation
- New accessible report themes
- Customise visible pages in the Page navigator visual
- Sensitivity labels now supported in PDF export from Desktop in Preview
- Enhanced Row Level Security editor in Preview

Analytics

- Quick Create SDK

Modelling

- New DAX functions: **LINEST** and **LINESTX**

Data connectivity

- Denodo (connector update)
- Digital Construction Works Insights (connector update)
- Profisee (connector update)
- Cosmos DB V2 (connector update)

Service

- New 'Get Data' experience in Power BI Service
- Compact view
- Linked metrics
- Follow metric

Paginated Reports

- Paginated Reports Formatted Table authoring experience
- Create a Paginated Report from a datamart
- Capacity planning for Paginated Reports

Embedded

- TypeScript support in the playground developer sandbox

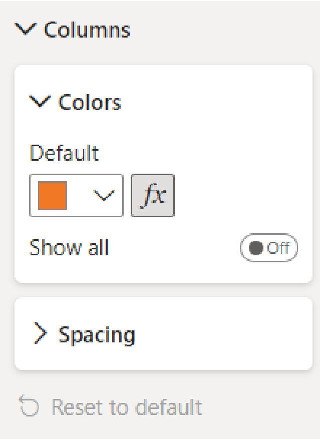
Visualisations

- New visuals in AppSource
- Milestone Trend Analysis Chart by Nova Silva
- Multiple Sparklines
- Inforiver Charts 2.1 is now Microsoft Power BI certified
- accoPLANNING by Accobat
- Drill Down Donut PRO by ZoomCharts.

Let's look at each in turn.

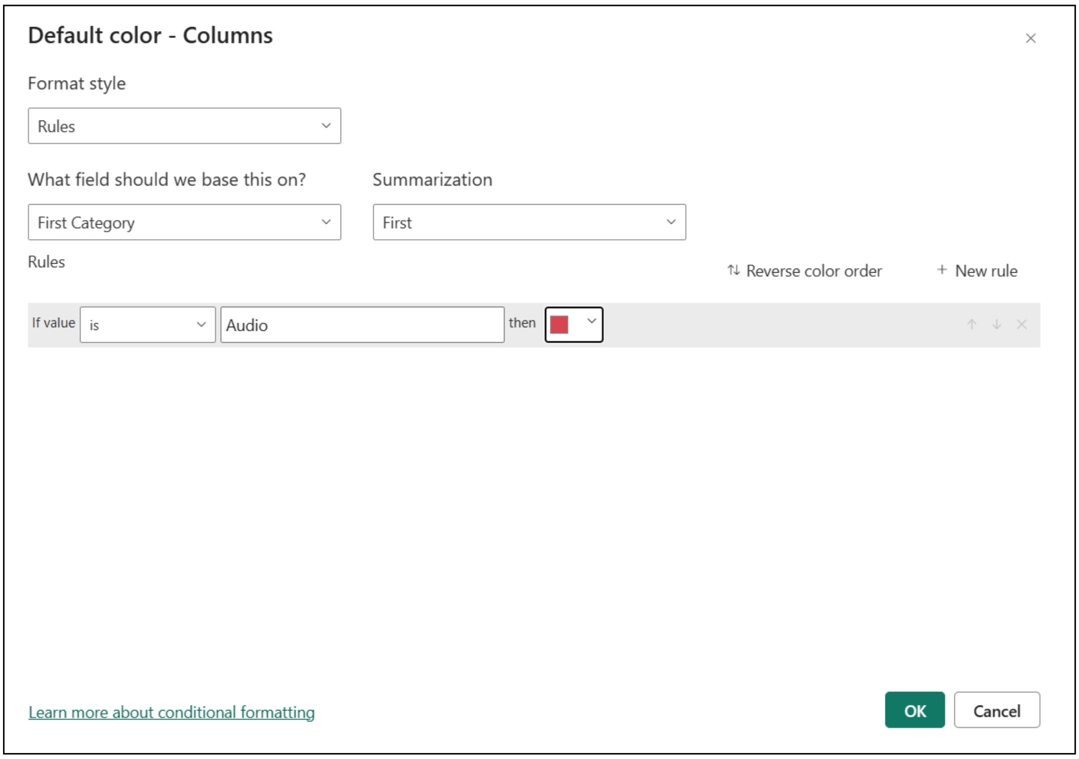
Conditional formatting based on string fields

This latest update allows you to set conditional formatting rules based upon string fields. You may access this new capability in the same way as all other conditional formatting: through the dialog which opens up from selecting an option in the Formatting pane. For example, in column charts, you can conditionally format column colours by selecting the **fx** button in the Columns card of the Formatting pane, *viz.*



In the dialog, select 'Rules' through the 'Format style' dropdown, then base these rules upon a string field and select a summarisation. Before, your summarisation options were only ever numerical, *e.g.* Count and Count (Distinct). Now you'll see First and Last options as well, which will give you strings to write logic around.

In the example below, we add a simple rule to check that the value of the string is Audio, and if it is, we wish to colour the column red:


Default color - Columns

Format style: Rules

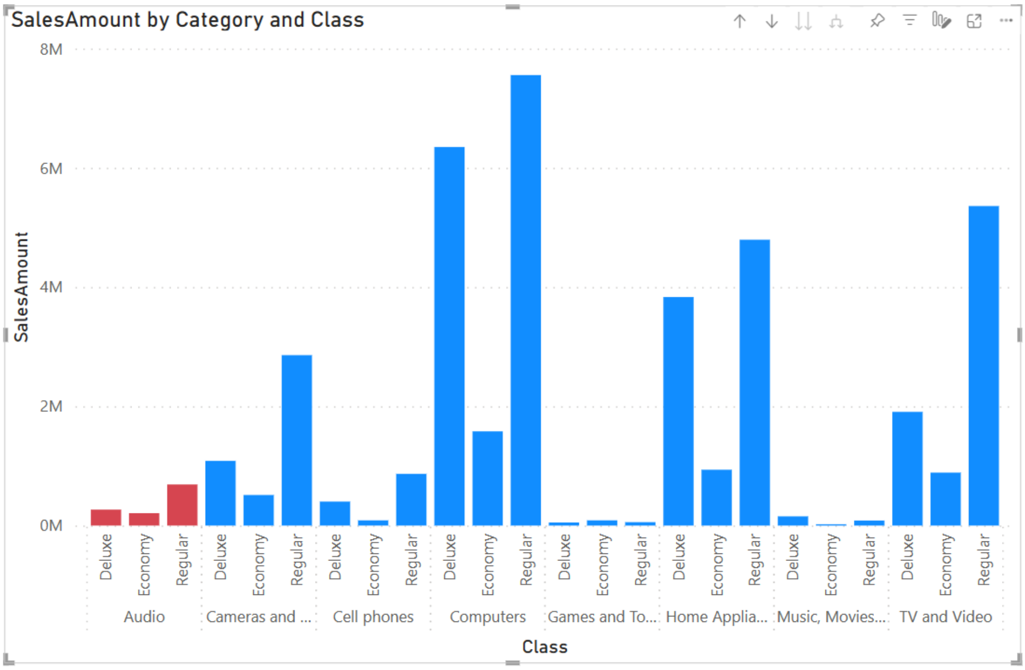
What field should we base this on?: First Category

Summarization: First

Rules

If value is Audio then 🟥

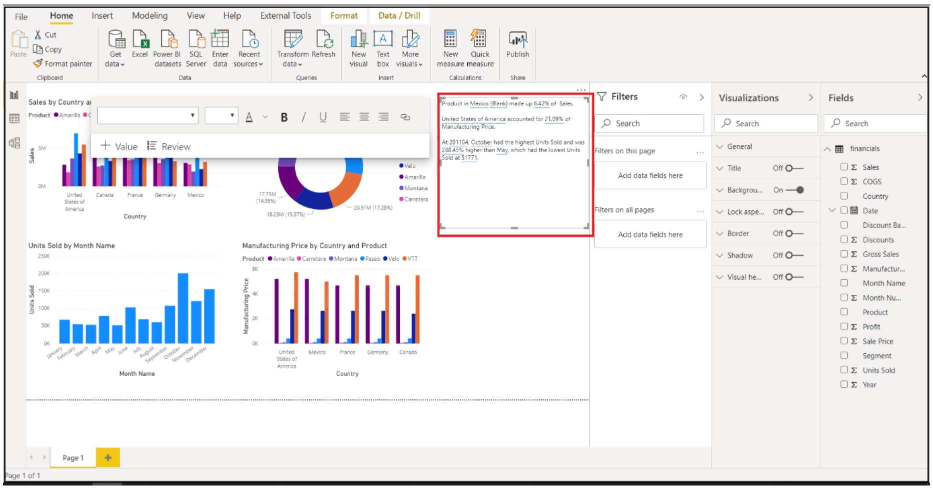
Then press OK, and the Audio columns of the visual will now be coloured red:



Power BI has added a variety of comparator options to the 'Conditional formatting' dialog, including 'is', 'starts with', 'ends with' and more. Do keep in mind that these comparisons will check your fields against your inputs for exact character matches, **including** case-sensitivity.

Smart Narrative visual summary icon

As many of you will already be aware, Smart Narrative technology is already available within Power BI, and with this new feature this update adds an optional icon to the visual header that triggers an on-demand summary of the visual contents for Accessibility purposes. It will announce results to any assistive technology. You can enable it in the Format pane for individual visuals or add it to your custom theme file for the visual types you choose. It should be noted that it has the same limitations as the existing Smart Narrative visual.



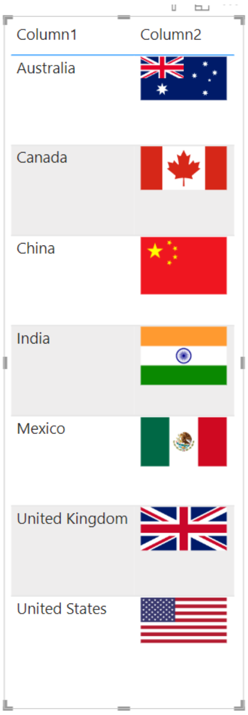
Product in Mexico (Blank) made up 6.42% of Sales.

United States of America accounted for 21.09% of Manufacturing Price.

At 201104, October had the highest Units Sold and was 288.45% higher than May, which had the lowest Units Sold at 51771.

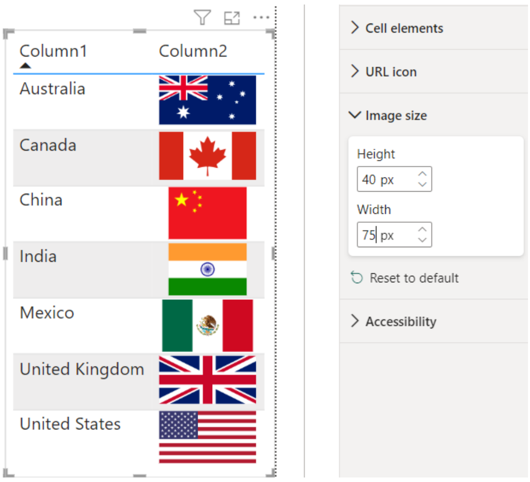
Formatting image width in table and matrix

Image dimensions in table and matrix visuals may now be individually formatted. Prior to this release, there was only one control for image dimensions in table and matrix cells. Despite being labelled as 'image height', this setting in the Formatting pane controlled both the width and height of images, causing them to take up square space regardless of their actual dimensions. This caused layout problems, especially when images were wider than they were tall, as rows then took up far more vertical space than necessary.



Column1	Column2
Australia	
Canada	
China	
India	
Mexico	
United Kingdom	
United States	

Now that you may change the image height and width separately:



Column1	Column2
Australia	
Canada	
China	
India	
Mexico	
United Kingdom	
United States	

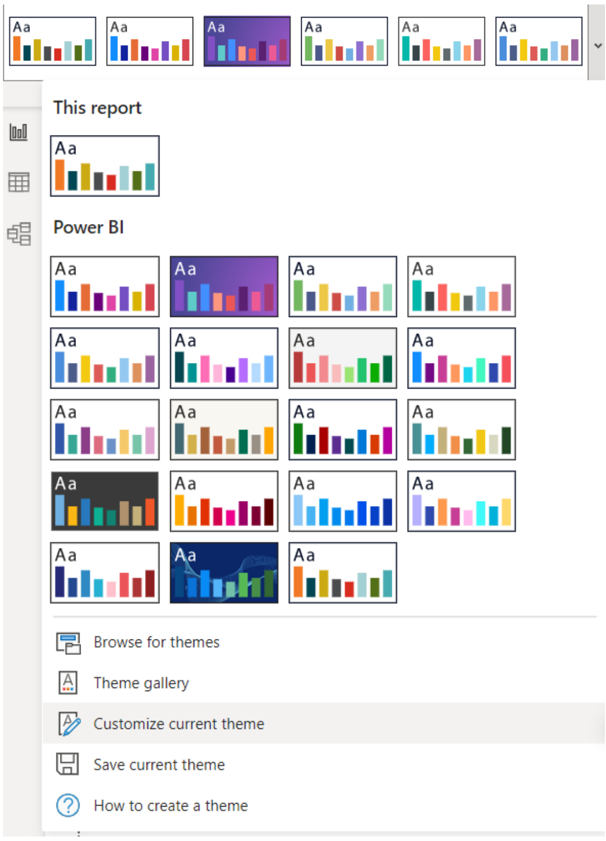
Height 40 px

Width 75 px

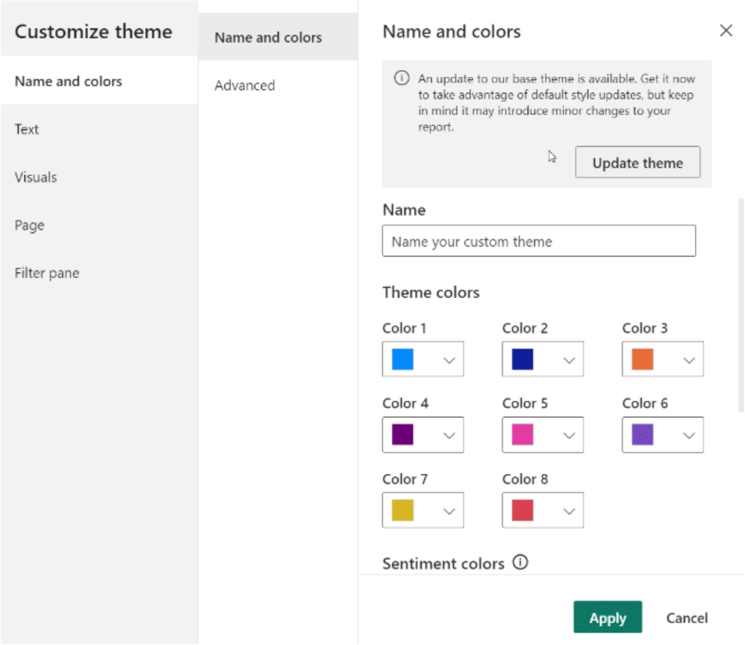
Update your base theme in Power BI

This update has now added an easy way for you to update your current theme in Power BI Desktop. Microsoft ships many changes to visuals and reports, including new defaults for formatting settings, as a part of a new base theme so as not to disturb existing reports. These changes don't propagate to some authors with custom themes, who may not see fixes to issues or updates to their reports without updating their base themes. Up until now, doing so involves opening up the theme gallery, switching to one of the default themes, then switching back to a custom theme, which is an unintuitive process.

Now, if you are using an outdated theme, simply enter the 'Customize current theme' menu from the theme dropdown in the View tab of the Ribbon:



You will find a banner at the top of the out-of-date themes encouraging you to update to the latest base theme. All you need do is click 'Update theme' to see the changes, then hit Apply when ready.



An update to our base theme is available. Get it now to take advantage of default style updates, but keep in mind it may introduce minor changes to your report.

Keep in mind that this change will update all of the default behaviours that Microsoft has changed since your current theme was created, so any specific behaviour that your visuals were reliant upon may change as well. You should check to make sure your report still functions the way you expect after updating your base theme.

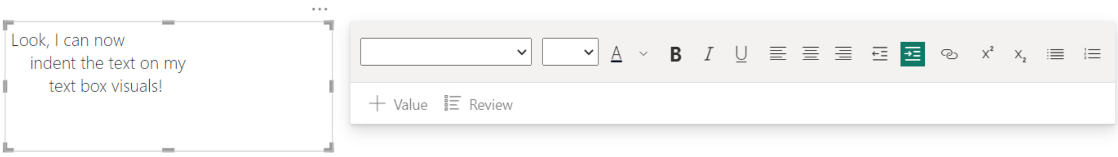
Report theme validation on custom theme import

Creating custom themes can be a daunting task, especially when there are so many visuals and formatting properties you can configure. Mistakes can occur when authoring the theme JSON, property names can change with time or any number of other issues could render the custom theme file invalid when it's imported into Power BI.

This latest release introduces a feature which will validate your theme file upon import to make sure Power BI can successfully read its full contents. If Power BI finds fields it doesn't understand, it will show you a message letting you know that the theme file is invalid, and invite you to re-import with a corrected theme file. Microsoft will also be publishing the schema Power BI uses to check your theme file in a public location, updating it automatically whenever we change it in Power BI, so that you can ensure your theme files are always current.

Text box visual indentation

This latest update provides new support for indentation to your text box visuals in Power BI:

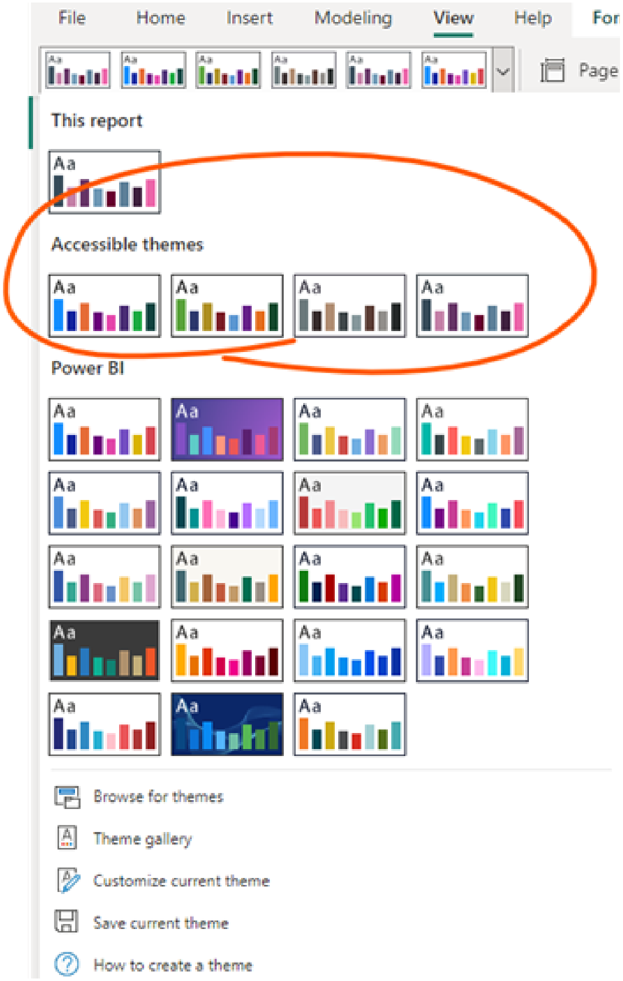


Look, I can now
indent the text on my
text box visuals!

Now, using the increase and decrease indent buttons in the Formatting popup next to the text box, you can adjust the indentation of specific lines of your text boxes. You can also use the **TAB** or **SHIFT + TAB** keys to accomplish the same effect.

New accessible report themes

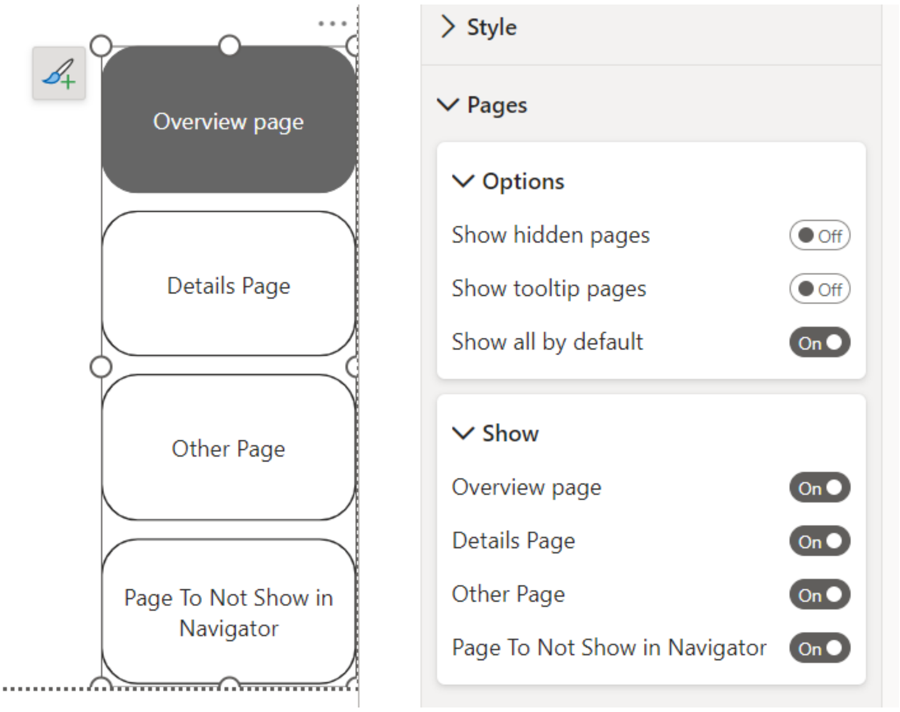
To promote authors creating accessible reports with good contrast across their colours, this update adds some new accessible report themes to the theme dropdown in the View tab of the Ribbon:



You'll find them in a new section labelled Accessible themes.

Customise visible pages in the Page navigator visual

You can now readily decide which report pages are visible within the Page navigator visual. If you want to exclude specific pages, all you need to do is expand the Show card in the Pages section of the Formatting pane. Here, you'll see the list of pages that are not already hidden through the 'Show hidden pages' or 'Show tooltip pages' options.



> Style

Pages

Options

Show hidden pages Off

Show tooltip pages Off

Show all by default On

Show

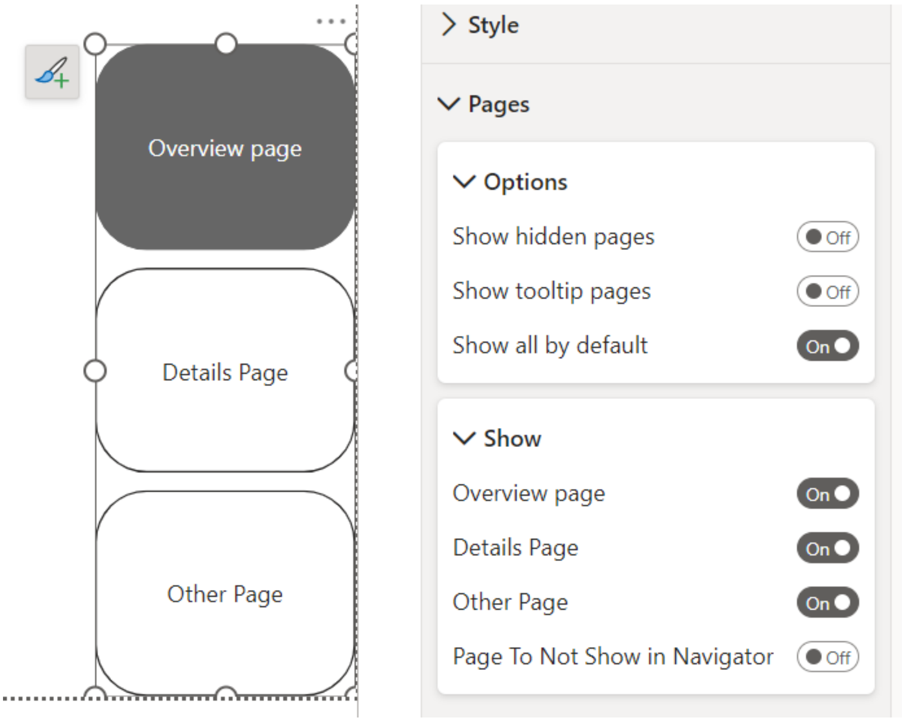
Overview page On

Details Page On

Other Page On

Page To Not Show in Navigator On

All you need to do to hide specific pages is turn the toggle for that page to Off.



> Style

Pages

Options

Show hidden pages Off

Show tooltip pages Off

Show all by default On

Show

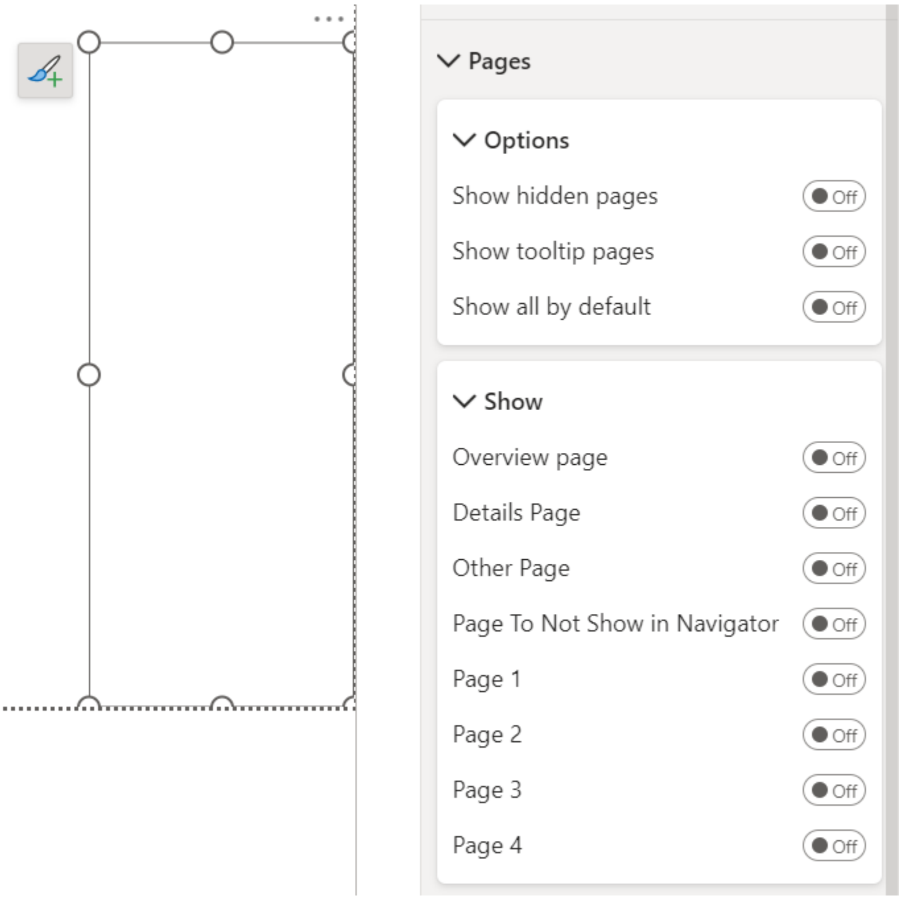
Overview page On

Details Page On

Other Page On

Page To Not Show in Navigator Off

If you have many pages in your report that you want to hide within the Page navigator visual, it can be faster to instead just specify which ones you specifically want to show. To do this, turn the 'Show all by default' toggle to Off in the Options card, which will immediately turn all pages within the Show card to Off.



Pages

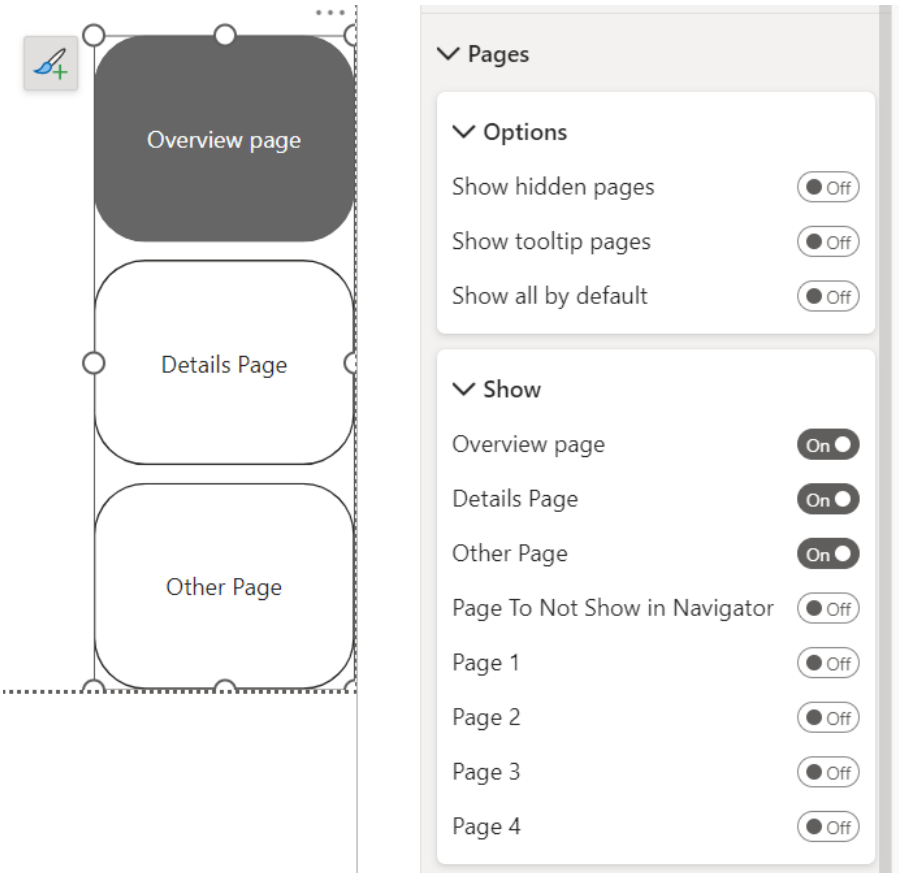
Options

Show hidden pages	Off
Show tooltip pages	Off
Show all by default	Off

Show

Overview page	Off
Details Page	Off
Other Page	Off
Page To Not Show in Navigator	Off
Page 1	Off
Page 2	Off
Page 3	Off
Page 4	Off

From there, you can then go in and just turn back on the specific pages you want visible.



Pages

Options

Show hidden pages	Off
Show tooltip pages	Off
Show all by default	Off

Show

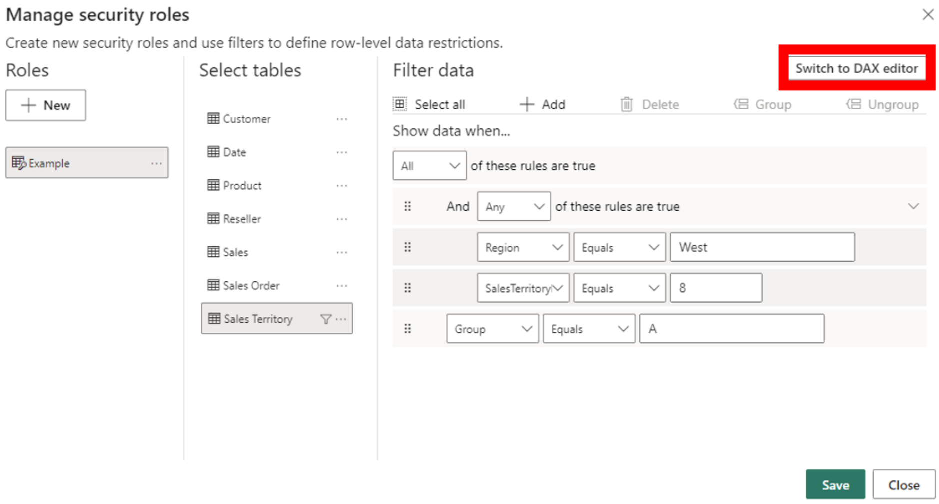
Overview page	On
Details Page	On
Other Page	On
Page To Not Show in Navigator	Off
Page 1	Off
Page 2	Off
Page 3	Off
Page 4	Off

Sensitivity labels now supported in PDF export from Desktop in Preview

Power BI Desktop now joins Power BI Service in supporting sensitivity labels from Microsoft Purview Information Protection for export to PDF. Now, when you export to PDF from Desktop, the sensitivity label on the PBIX file (if any) is applied to the exported PDF file, thus ensuring that your sensitive data remains protected even after it has left Power BI. Sensitivity label support in Desktop for export to PDF is currently a Preview feature that is on by default.

Enhanced Row Level Security editor in Preview

Now you can quickly and easily define Row Level Security (RLS) roles and filters without having to write any DAX. Select 'Manage roles' in the Ribbon to use the new drop-down interface to create and edit security roles. If you prefer DAX or require DAX for your filter definitions, the software supports toggling between using the default drop-down editor and a DAX editor.


Manage security roles

Create new security roles and use filters to define row-level data restrictions.

Show data when...

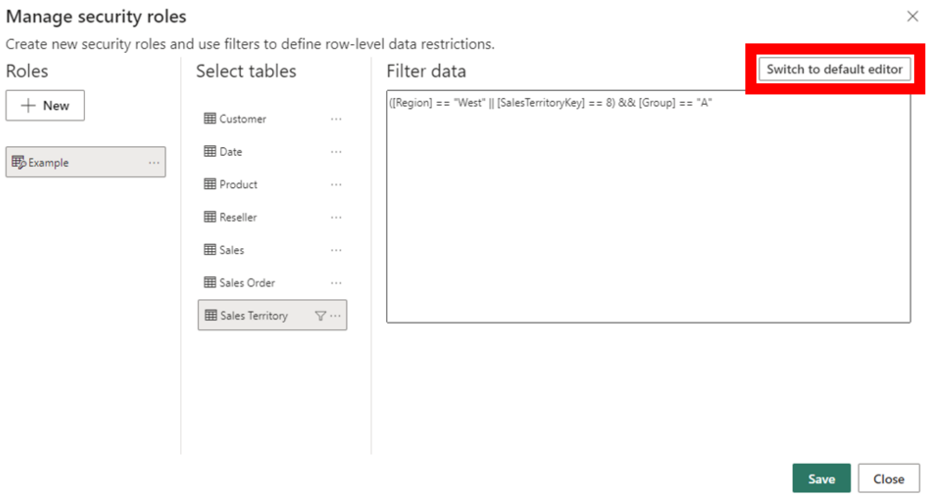
All of these rules are true

And Any of these rules are true

Region Equals West

SalesTerritory Equals 8

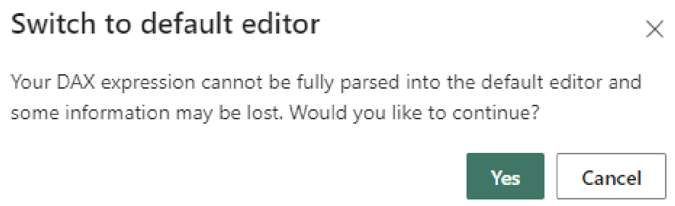
Group Equals A


Manage security roles

Create new security roles and use filters to define row-level data restrictions.

```
([Region] == "West" || [SalesTerritoryKey] == 8) && [Group] == "A"
```

All changes made in either editor interface will persist when switching interfaces when possible. However, not all RLS filters supported in Power BI may be defined using the default editor. Limitations include expressions that can only be defined using DAX including dynamic rules such as **USERNAME()** or **USERPRINCIPALNAME()**. When defining a role with these limitations in the DAX editor, if you attempt to switch to the default editor you will be prompted with a warning that switching editors may result in some information being lost. To keep this information simply select 'Cancel' and continue only editing this role in the DAX editor.

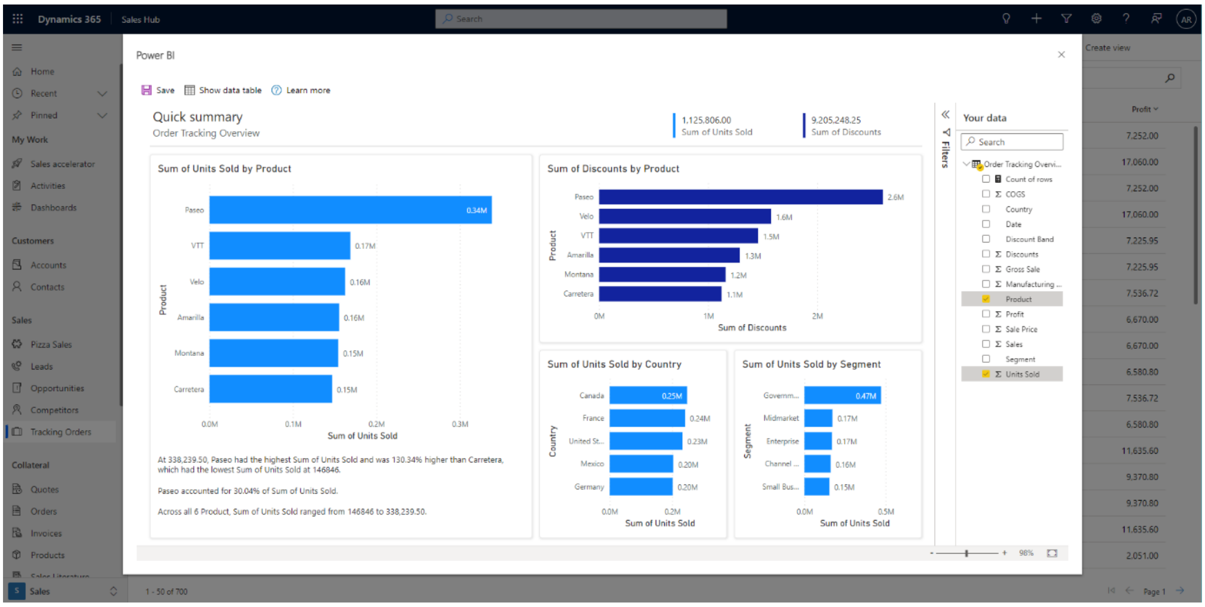

Switch to default editor

Your DAX expression cannot be fully parsed into the default editor and some information may be lost. Would you like to continue?

You can try this Preview feature right now by going to **Files -> Options and Settings -> Options -> Preview features** and turn on 'Enhanced row-level security editor'.

Quick Create SDK

This update announces a new form of Power BI embedded analytics that enables you to embed an interactive data exploration and report creation experience in your applications. With this solution, you'll be able to provide your users a similar experience to integrations in Dynamics 365 and SharePoint.



At 338,239.50, Paseo had the highest Sum of Units Sold and was 130.34% higher than Carretera, which had the lowest Sum of Units Sold at 146846.

Paseo accounted for 30.04% of Sum of Units Sold.

Across all 6 Product, Sum of Units Sold ranged from 146846 to 338,239.50.

Profit
7,252.00
17,060.00
7,252.00
17,060.00
7,225.95
7,225.95
7,536.72
6,670.00
6,670.00
6,580.80
7,536.72
6,580.80
11,635.60
9,370.80
9,370.80
11,635.60
2,051.00

By using this new set of Power BI client APIs, your application can pass in data source information, and Power BI will automatically generate a dataset and report for your users to explore, modify and if they choose, save to a workspace in their Power BI tenant.

To set this up, you'll need to create a configuration object to store information about your data source and customise the report that is initially created. As part of this, you'll be able to pick how you want to connect your application's data to Power BI. You can either pass an array of data directly to Power BI or else you may use a Power Query connector by passing in a mashup document.

Passing in the data directly is a very easy way to get started. It's aimed at smaller amounts of data, up to 16MB in size, and doesn't require you to develop your own connector. Since you're passing in the data directly, it's not refreshable in the future, so more suited towards enabling your users to explore their data to answer a point in time question.

You'll pass in data as rows using a simple format, *e.g.*

```
const data: IDataTable = [
{
     "name": "Table",
        "rows": [
         ["Product1", "20", "2022"],
         ["Product2", "35", "2021"],
        ]
    }
]
```

In addition to name and **dataType**, you can specify default aggregation on columns by providing an optional property **aggregateFunction** for each column. The number for supported aggregate functions are based on the **aggregateFunction** Enum type.

Alternatively, by using a data connector, you'll be able to enable your users to create refreshable, reusable reports and datasets. This route requires you to create a Power Query connector or expose your data through one of our existing data connectors, but users will be able to visualise much more data and return to their reports in the future to see the latest data. Microsoft currently supports anonymous data connections or authenticating with AAD tokens, and in the future, they plan to enable support for other authentication methods too.

To use this method, you'll just need to pass in a mashup document and the data model configurations, *e.g.*

```
const datasetCreateConfig: IDatasetCreateConfiguration = {

    "modelName": "temp model",

    "locale": "en_US",

    "mashupDocument": "section Section1; shared Table = let Source = Csv.Document(Web.
Contents(\"https://yourdomain.com/sample.csv\"),[Delimiter=\",\", QuoteStyle=QuoteStyle.None]),
#\"Promoted Headers\" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]) in #\"Promoted
Headers\";",

    "dataSourceConnection": {

        "credentialType": CredentialType.Anonymous, // =1

        "dataCacheMode": DataCacheMode.Import // =1

    }

}
```

This new embedding option is a form of SaaS embedding, so your app users will need to sign into their own Power BI account to generate the report and dataset. This means it's free for you to include in your application. While your users will need to sign up for Power BI, if they don't already have an account, they'll only need to have a free licence to explore their data and save it into their personal workspace.

New DAX functions: LINEST and LINESTX

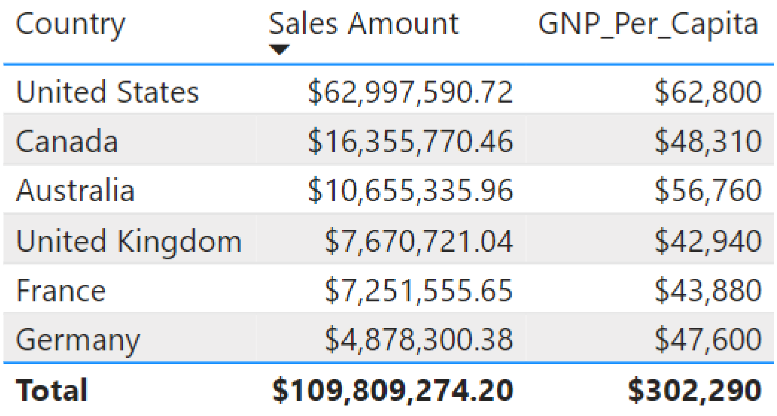
This month, two new statistical DAX functions have been added to the Power BI repertoire: **LINEST** and **LINESTX**. These two functions perform linear regression, leveraging the Least Squares method, to calculate a straight line that best fits the given data and return a table describing that line. These functions are especially useful in predicting unknown dependent values (**y**) given known independent values (**x**).

Both functions return a single-row table describing the line and additional statistics. The resulting table includes columns such as slopes, intercepts, standard errors and the coefficient of determination. The equation of the fitted line can be constructed as follows

$$\mathbf{y = Slope1 * x_1 + Slope2 * x_2 + \ldots + Intercept}.$$

The difference between **LINEST** and **LINESTX** is that **LINEST** expects columns of known **x** and **y** values to be provided, whereas **LINESTX** expects a table and expressions to be evaluated for each row of the table to obtain the **x** and **y** values.

For the following examples, consider the following data, which includes **Sales Amount** and Gross National Product, **GNP_Per_Capita**:



Country	Sales Amount	GNP_Per_Capita
United States	$62,997,590.72	$62,800
Canada	$16,355,770.46	$48,310
Australia	$10,655,335.96	$56,760
United Kingdom	$7,670,721.04	$42,940
France	$7,251,555.65	$43,880
Germany	$4,878,300.38	$47,600
Total	**$109,809,274.20**	**$302,290**

In the example below, we will use **LINESTX** to predict total sales based upon GNP per capita:

```
LinestX_example =

VAR CountryGNP = SUMMARIZE(

Sales,

'GNP_Country'[Country],

'GNP_Country'[GNP_Per_Capita],

"Total Sales", SUM(Sales[Sales Amount])

)
```

```
VAR SalesPrediction = LINESTX(
'CountryGNP',
[Total Sales],
[GNP_Per_Capita]
)

VAR Example_GNP_Per_Capita = 50000

RETURN SELECTCOLUMNS(
SalesPrediction,
[Slope1]
) * Example_GNP_Per_Capita +
SELECTCOLUMNS(
SalesPrediction,
[Intercept]
)
```

This expression not only leverages **LINESTX** but also leverages the result to perform a prediction for a fictitious country with gross national product per capita of $50,000. The result is a predicted total sales of $17,426,123.29. Of course, this is a fabricated scenario and it's rare to have a fixed value such as the $50,000 above as part of the expression.

We may do the same using **LINEST** assuming the required tables are all in the model, *e.g.* as calculated tables. In this example, we've added the following calculated tables:

- **CountryDetails**, defined as:

  ```
  CountryDetails = SUMMARIZECOLUMNS(
  'GNP_Country'[Country],
  'GNP_Country'[GNP_Per_Capita],
  "Total Sales", SUM(Sales[Sales Amount]))
  ```

- **SalesPredictionLINEST,** defined as:

  SalesPredictionLINEST = LINEST('CountryDetails'[Total Sales], 'CountryDetails'[GNP_Per_Capita]).

Now we may use the following measure expression to obtain the same result as above:

```
Linest_example =
VAR Example_GNP_Per_Capita = 50000

RETURN
MAX ( SalesPredictionLINEST[Slope1] ) * Example_GNP_Per_Capita
+ MAX ( SalesPredictionLINEST[Intercept] )
```

Denodo (connector update)

The Denodo connector has been updated. This update adds an option to the **Denodo.Contents** function to work around issues with relationship discovery in tables with a large amount of relationships.

Digital Construction Works Insights (connector update)

This connector's Beta flag has been removed.

Profisee (connector update)

The Profisee connector has been updated. This update has:

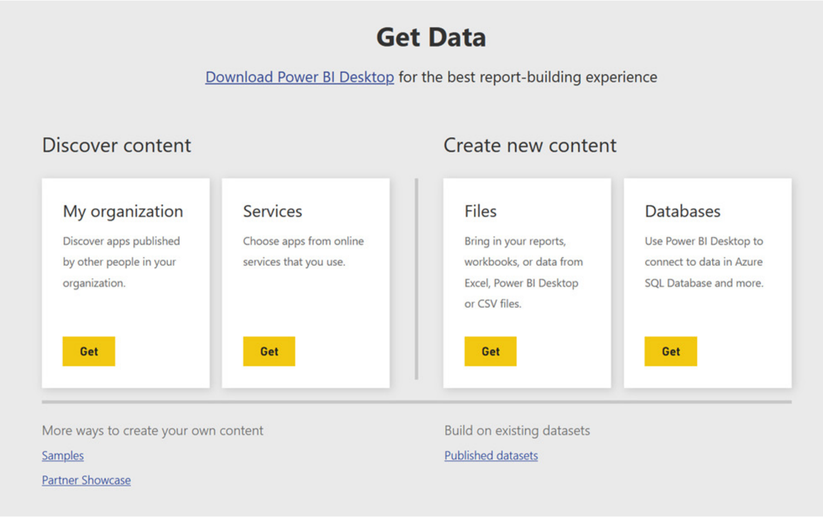
- updated error handling to support changes in the 2022 R2 release
- removed the Beta flag.

Cosmos DB V2 (connector update)

This release contains an update to the Cosmos DB V2 connector that improves DirectQuery performance when a filter on partition key is specified.

New 'Get Data' experience in Power BI Service

As mentioned a couple of months ago, Microsoft has completed the process of removing the older 'Get Data' page in Power BI Service in favour of the new, comparable features available within workspaces. The change removed the entry points to the old 'Get Data' page as shown below:



Get Data

Download Power BI Desktop for the best report-building experience

Discover content

My organization
Discover apps published by other people in your organization.
Get

Services
Choose apps from online services that you use.
Get

Create new content

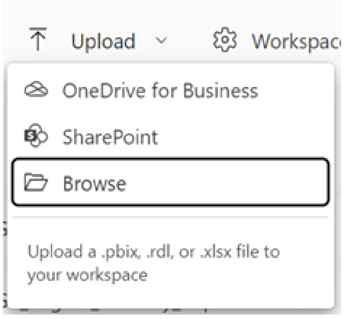
Files
Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.
Get

Databases
Use Power BI Desktop to connect to data in Azure SQL Database and more.
Get

More ways to create your own content
Samples
Partner Showcase

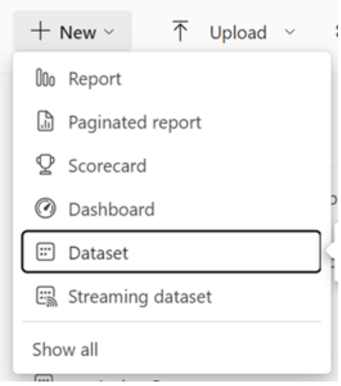
Build on existing datasets
Published datasets

From now on, you'll be able to access comparable features within workspaces. If you want to upload a file to Power BI, such as a .pbix, .xlsx or .rdl files to your workspace, you can use the Upload option that was released in November. This option lets you upload files from your local computer or connect to files on OneDrive or a SharePoint site. With this change, you'll no longer be able to connect to files on personal OneDrive accounts.

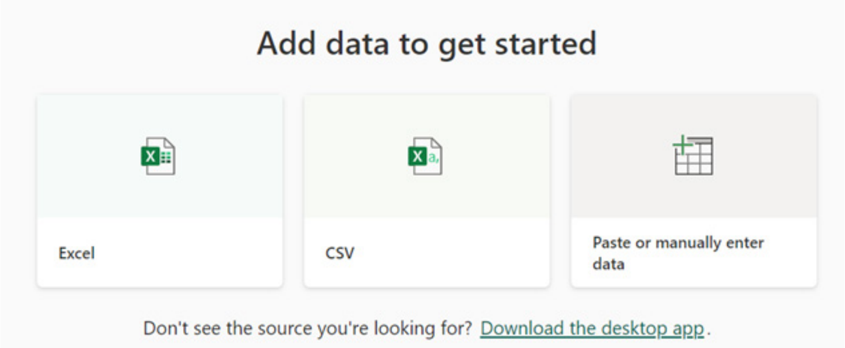


Upload a .pbix, .rdl, or .xlsx file to your workspace

If instead you want to create a dataset from Excel or CSV data, you can now access that functionality through the **New -> Dataset** option in the workspace you want to create the dataset in.



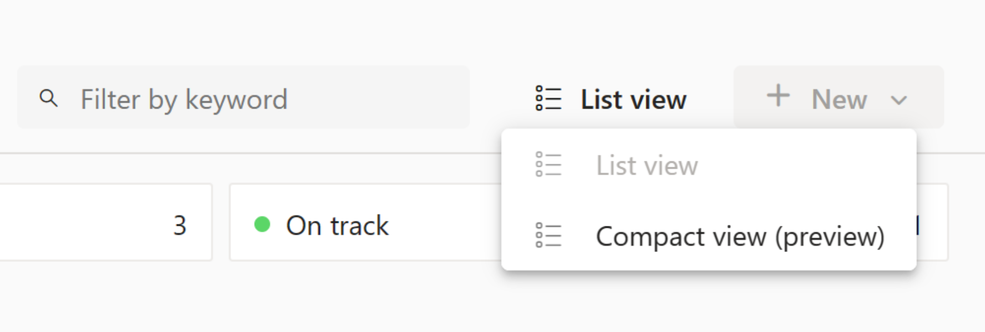
This Dataset option has also been updated to take you to a new page with options to create a dataset off an Excel, CSV or pasting in data. Once you select the file, the behaviour used to generate the dataset is the same as previously used on the 'Get Data' page. Once the dataset is created, you'll be taken to the dataset's details page in the Data hub.

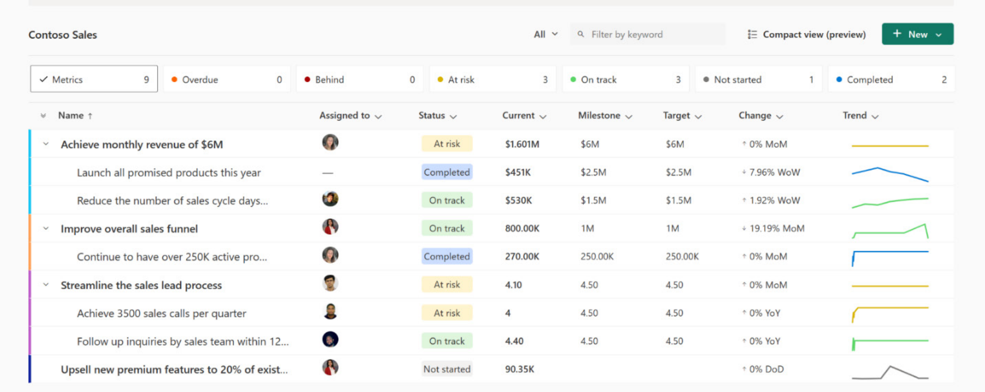

Add data to get started
Excel
CSV
Paste or manually enter data
Don't see the source you're looking for? Download the desktop app.

Compact view

Scorecards were built for easy readability on larger screens, but they let you view only a few metrics at a time which led to significant scrolling when there were a lot of metrics. To make it easier to use, a denser version has now been released that doubles the number of metrics that may be viewed on the screen. By tightening the scorecard, removing the spacing between metrics and the padding within a metric, Microsoft has designed a condensed view of the scorecard called the Compact view in addition to the list (Original) view.

You may switch to the Compact view of the scorecard by using the view selector in the scorecard header:





Name ↑	Assigned to	Status	Current	Milestone	Target	Change	Trend
Achieve monthly revenue of $6M		At risk	$1.601M	$6M	$6M	↑ 0% MoM	
Launch all promised products this year	—	Completed	$451K	$2.5M	$2.5M	↓ 7.96% WoW	
Reduce the number of sales cycle days...		On track	$530K	$1.5M	$1.5M	↑ 1.92% WoW	
Improve overall sales funnel		On track	800.00K	1M	1M	↓ 19.19% MoM	
Continue to have over 250K active pro...		Completed	270.00K	250.00K	250.00K	↑ 0% MoM	
Streamline the sales lead process		At risk	4.10	4.50	4.50	↑ 0% MoM	
Achieve 3500 sales calls per quarter		At risk	4	4.50	4.50	↑ 0% YoY	
Follow up inquiries by sales team within 12...		On track	4.40	4.50	4.50	↑ 0% YoY	
Upsell new premium features to 20% of exist...		Not started	90.35K			↑ 0% DoD	

The compact view is responsive, adjusting smoothly to various screen sizes and supports resizing of columns. You can do all the functionality that is available in the list view of the scorecard except editing and creating a new metric. This is disabled in the edit mode of Compact view and can only be performed in the List view.

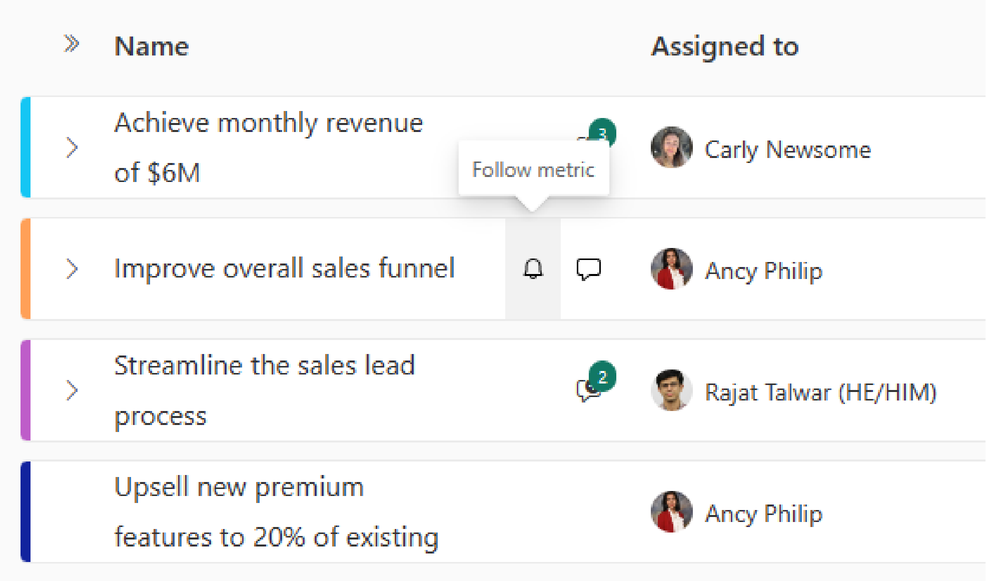
Shortly (if not already by the time you read this), Power BI will be adding two new settings in the column settings pane to customise the layout: text wrapping for metric name and horizontal scroll. Toggling on the horizontal scroll option will let you navigate left to right until you reach the final column. However, by setting it to off, your view of this scorecard will leave off the columns that don't fit on one screen without scrolling.

Linked metrics

Recently, linked metrics were shipped. These let you show the same metric on multiple scorecards, across multiple workspaces. There are a few improvements to linked metrics this month. If you now link multiple metrics at once, the parent-child relationships and the metrics' order from the source scorecard are preserved in the destination scorecard. Further, you may now use the 'Set For All' option to apply permissions to descendant metrics on the source scorecard, even if there are linked metrics in between the parent and descendants.

Follow metric

Last year saw the introduction of 'Follow metric' functionality to quickly access metrics you're interested in and stay up to date on the activity on these metrics through Teams notifications. The icon of 'Follow metric' has now been changed to an alert icon for better association with notifications, *viz.*

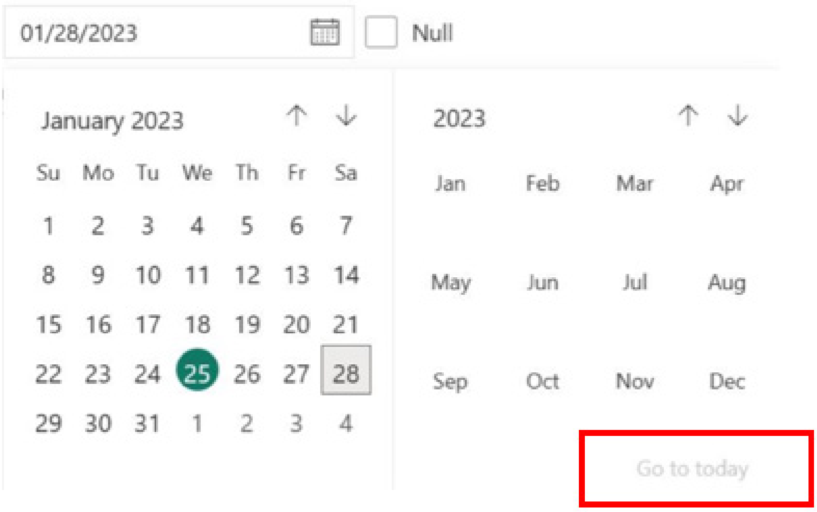


Name	Assigned to
Achieve monthly revenue of $6M	Carly Newsome
Improve overall sales funnel	Ancy Philip
Streamline the sales lead process	Rajat Talwar (HE/HIM)
Upsell new premium features to 20% of existing	Ancy Philip

Paginated Reports Formatted Table authoring experience

In the last update, Microsoft introduced accessible navigation, moving and resizing, as well as grand totals. This month, there are additional improvements for Paginated Reports and additional capabilities when using a Formatted Table.

Paginated Reports are known for their use of parameters. Many use parameters a lot, so Microsoft has made some accessibility improvements for better screen reader support and keyboard navigation. The date picker now has a 'Go to today', month, and year view to make it easier for date selection.



01/28/2023 ☐ Null

January 2023

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

2023

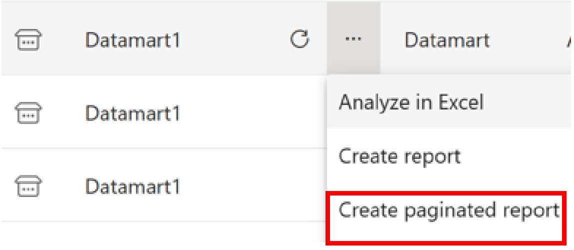
Jan	Feb	Mar	Apr
May	Jun	Jul	Aug
Sep	Oct	Nov	Dec

Microsoft has also made all dropdowns list values resizable, so you may drag and resize values with longer lengths. You should also notice performance improvements for parameters with long lists.

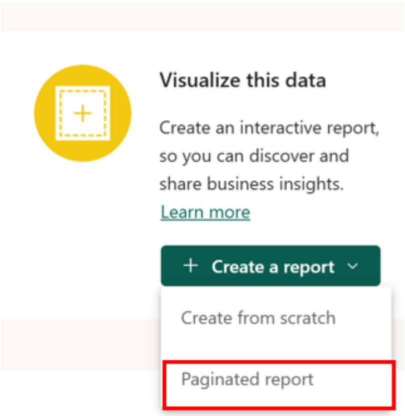
Create a Paginated Report from a datamart

End users may now create a paginated report directly from a datamart. There are two entry points:

1. directly from the context menu of a datamart



2. from the Data hub when you click on the datamart to open datamart details.



Visualize this data

Create an interactive report, so you can discover and share business insights.

Capacity planning for Paginated Reports

For users interested in migrating their enterprise on-premises Business Intelligence reports to Power BI, Microsoft advise that you should plan your Premium capacity to get the best performance out of your Paginated Reports.

TypeScript support in the playground developer sandbox

Typescript is now available in the developer sandbox in the Power BI embedded playground. Until now, only JavaScript was available in the sandbox, but now, Microsoft has made it possible to code in TypeScript as well.

The Power BI embedded playground is useful for Power BI embedded, including the developer sandbox, showcasing different capabilities and sample scenarios, and providing resources for learning. The playground assists you in experiencing Power BI embedded.

The developer sandbox provides hands-on coding experience, allowing you to embed your own reports (or use our sample report) and interact with the Power BI client APIs. This means you can see instant results without having to build your own environment.

To switch to TypeScript in the sandbox, simply click on the new dropdown in the upper right corner and select your preferred language. If you are logged in and you choose to save your code in the playground, it will keep your selected language and load it the next time you visit the playground.

New visuals in AppSource

This month sees the following new visuals in AppSource:

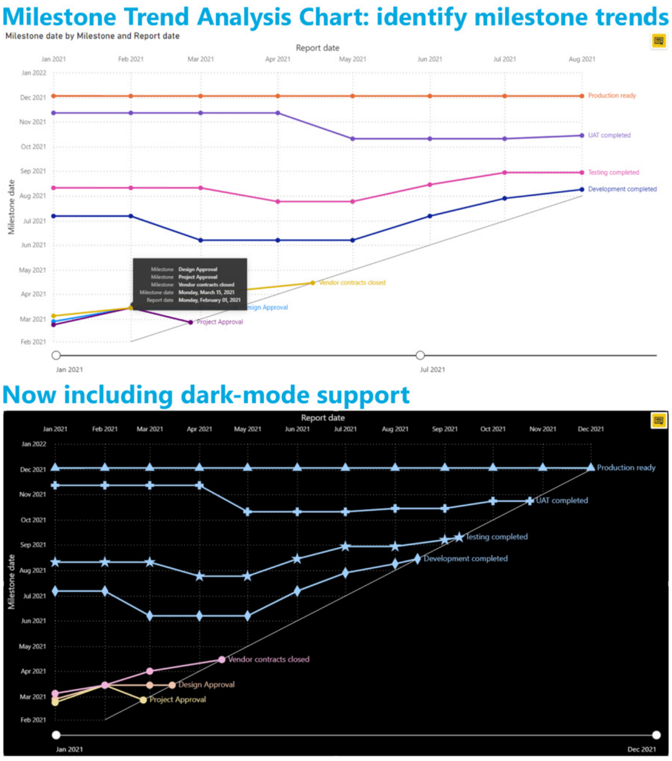
- Beeswarm Chart
- Cumulative by sio2Graphs
- DoubleYaxis
- PictureThis
- sw3D_StackedBarDiagram
- swBoxPlot
- swUniaxialScatterPlot.

Milestone Trend Analysis Chart by Nova Silva

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

Key updates in the Milestone Trend Analysis Chart are:

- zoom slider support to allow for easy navigation through time
- support for different marker icons
- direct labelling, allowing the user to utilise series labels to replace the legend
- ToolTip will include details of all overlapping markers
- AppSource eCommerce transactability: you may now purchase and manage your licenses like any Microsoft license
- dark mode support: all elements can be formatted to support a dark background.

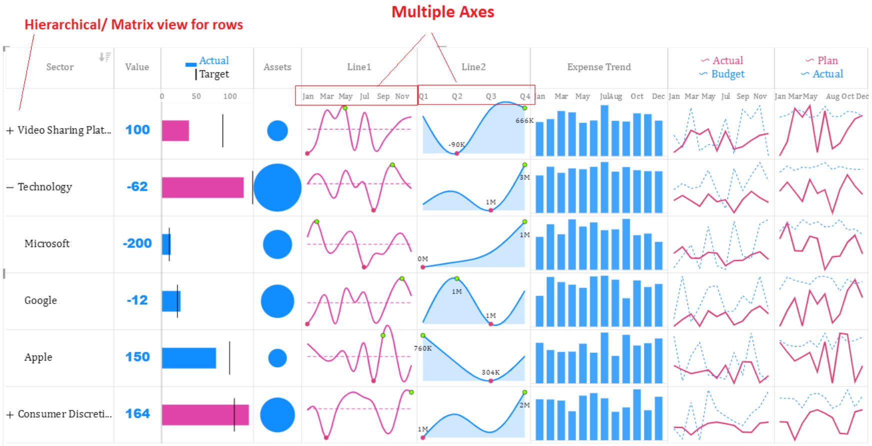


The Milestone Trend Analysis Chart makes use of the familiar standard Power BI user interface and may be downloaded from AppSource.

Multiple Sparklines

New features were added to Multiple sparklines:

- Matrix format allowing you to display hierarchical data in rows
- Multiple Axes for trend charts (sparklines, column chart, stacked column, combo chart).



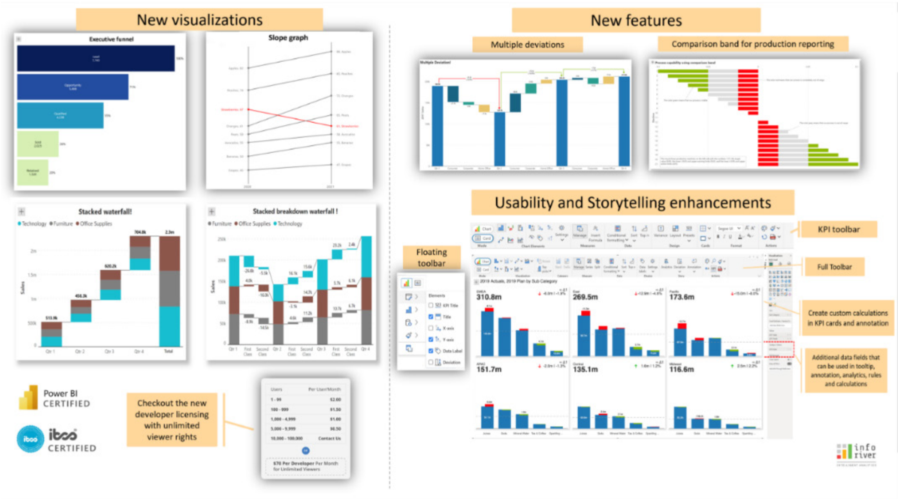
This update may be downloaded from AppSource.

Inforiver Charts 2.1 is now Microsoft Power BI certified

Inforiver Charts 2.1 is now Microsoft Power BI certified. The latest version 2.1 features significant feature updates, user experience (UX) enhancements and a new developer-only pricing plan that can be directly purchased from Microsoft AppSource. This new pricing plan allows for unlimited viewers and avoids licence management and administration hassles.

What's new in Inforiver Charts 2.1:

- **new visualisations:**
  - Executive Horizontal / Vertical Funnel chart with secondary KPI
  - Stacked / paired breakdown waterfall charts
  - Slopegraph
  - Charts with integrated KPI card headers


New visualizations
Executive funnel
Slope graph
Stacked waterfall!
Technology
Furniture
Office Supplies
Stacked breakdown waterfall !
Power BI CERTIFIED
ibcs CERTIFIED
Checkout the new developer licensing with unlimited viewer rights
Users
Per User/Month
1 - 99
$2.00
100 - 999
$1.50
1,000 - 4,999
$1.00
5,000 - 9,999
$0.50
10,000 - 100,000
Contact Us
$70 Per Developer Per Month for Unlimited Viewers
New features
Multiple deviations
Comparison band for production reporting
Usability and Storytelling enhancements
Floating toolbar
KPI toolbar
Full Toolbar
Create custom calculations in KPI cards and annotation
Additional data fields that can be used in tooltip, annotation, analytics, rules and calculations
310.8m
269.5m
173.6m
151.7m
135.1m
116.6m
inforiver

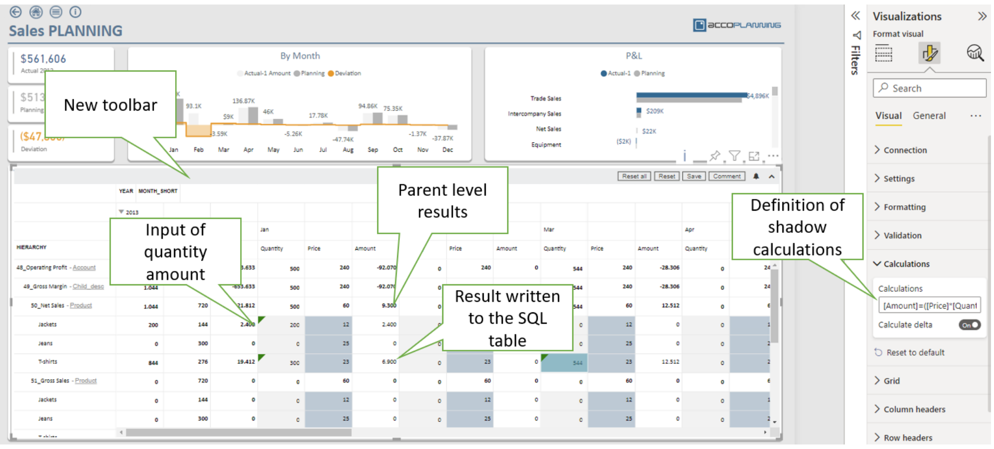
- **new use cases:**
  - Banded Bar and Column charts for manufacturing and production KPI reporting
  - create and display secondary KPIs on visuals (using 'Others' data field) with integrated formula engine
  - create stamp-sized, micro layouts and KPI spark cards
- **enhanced user experience (UX) and productivity:**
  - *for developers:* multiple toolbar modes (full or floating), quick feature search, on-canvas element interactivity, responsiveness, font scaling and more
  - *for viewers:* increased reading mode options, contextual drilldown for exploratory analysis and more
- **storytelling enhancements:**
  - customisable legends, titles, KPI card headers, annotations, analytics and more
- **themes:**
  - choose between classic Power BI-like colour themes, IBCS standards or custom themes matching your corporate branding.

accoPLANNING by Accobat

This enables writeback for any Power BI model. Therefore, you could use Power BI for planning and forecasting. With this visual you can now writeback numbers, text and dates on any existing or new Power BI model.

Key new functionalities include:

- **shadow calculations:**
  - enables you to avoid performance overhead on your leaf calculations like **Price * Quantity**
  - benefit from having the result of your calculations stored in the writeback table for use in other models or systems
  - real-time calculations while you are typing
- **general enhancements:**
  - optimised and extended functionality for copying and pasting of data from Excel, providing you with a fast and easy way to add data to your model
  - better handling of typing long comments and text with cell scrollbar and mouse over preview of text
- **AppSource licence handling:**
  - you can now purchase license directly from the AppSource



New toolbar

Parent level results

Input of quantity amount

Definition of shadow calculations

Result written to the SQL table

Drill Down Donut PRO by ZoomCharts

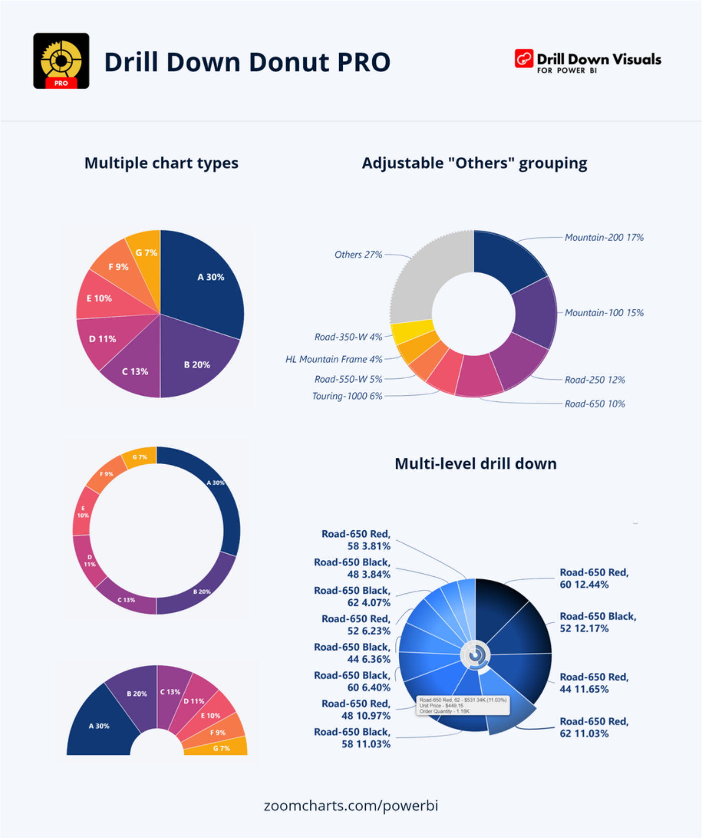
Drill Down Donut PRO for Power BI lets you create do(ugh)nut charts for easy data exploration. All interactions take place on the chart itself using just the mouse. You can fully customise the look of your chart with advanced formatting options and even choose from different chart types.

Main features:

- **adjustable ‘Others’ slice:** set the number of visible slices and group the rest
- **cross-chart filtering:** use the chart as a filter for the rest of the report
- **multi-level drill down:** drill down up to nine [9] levels
- **full customisation:** modify slices, labels and the legend
- **desktop and mobile device navigation:** explore charts the same way on any device.

Popular use cases include:

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



Again, available from AppSource.

That’s it for this month: more next newsletter.

New Features for Excel

The latest updates see the release of PivotTable 'Show Details' in Excel for the web and the ability to automate your tasks with the Power Automate tab in Excel for Windows and Mac. For Insiders users on Windows and Mac, you can try recording your worksheet actions using Office Scripts and checking your formula with the Value Preview ToolTip.

The full list is as follows:

Excel for the web

- PivotTable Show Details

Excel for Windows

- Automate your tasks with the 'Power Automate' tab
- Record worksheet actions using Office Scripts (Insiders)
- Check your formula with Value Preview ToolTips (Insiders)

Excel for Mac

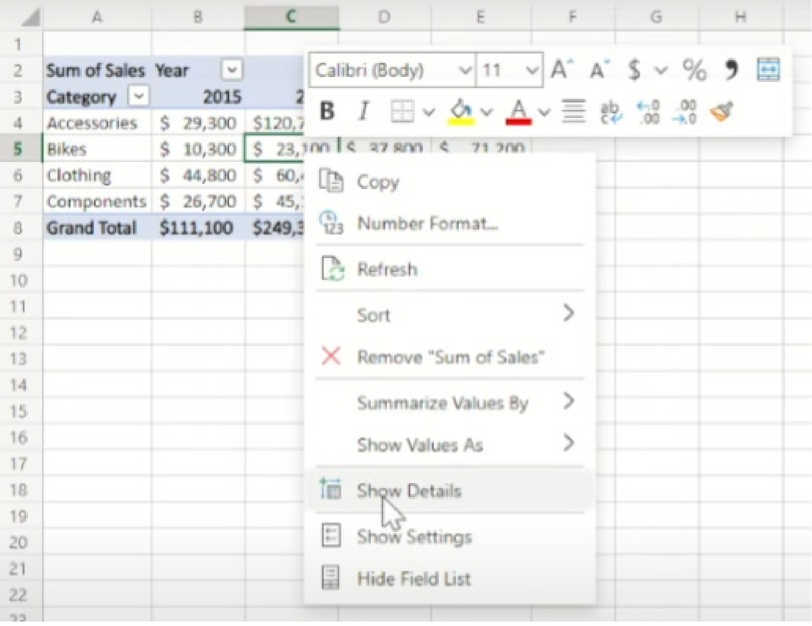
- Automate your tasks with the 'Power Automate' tab
- Record worksheet actions using Office Scripts (Insiders)
- Check your formula

Let's plough through.

PivotTable Show Details

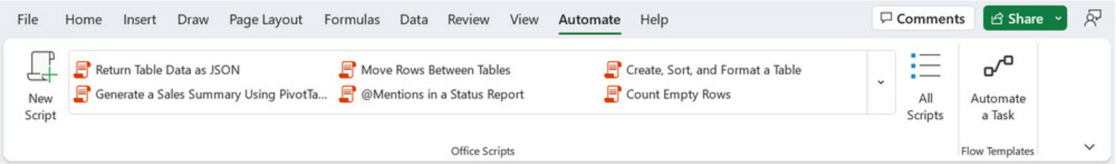
In Excel for the web, you can now view the source data for a PivotTable value cell. Simply double-click or right-click and choose 'Show Details from a value cell' in your PivotTable to create a new sheet with the detailed rows for that cell. There is also a new button to activate the same functionality from the PivotTable tab of the Ribbon.

'Show Details' is useful to help you understand how the values in your PivotTable were calculated. For example, if you see an unexpected value in your PivotTable, you can use 'Show Details' to confirm whether the source data used to calculate that value is as expected or if there is an issue with the underlying data.



Automate your tasks with the 'Power Automate' tab

Office Scripts enables you to automate repetitive tasks in your spreadsheet workflows and automate repetitive tasks in your spreadsheet workflow and automate repetitive tasks in your spreadsheet workflow. The following tab has now come to Excel for Windows and Excel for Mac:

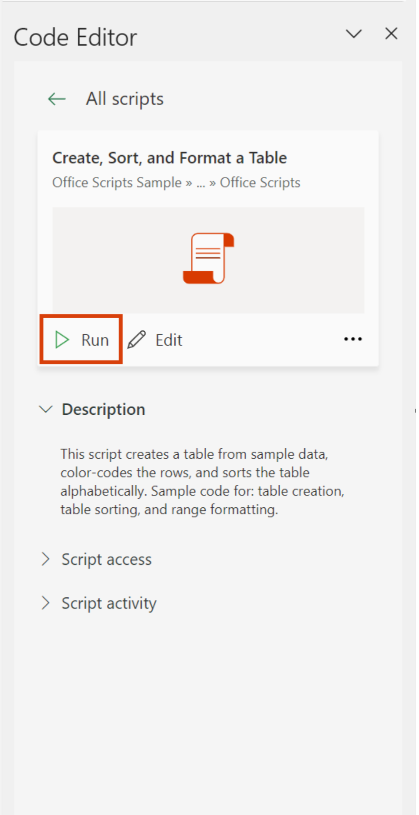


These scripts help you save time by increasing efficiency and reducing errors in your workflow, and you can even schedule your scripts so that your automations can run even while you're away.

Up until now, you could only create these helpful scripts in Excel on the web. With this update, the Automate tab is now available for all eligible enterprise users in Excel for Windows and Mac. Previously, this tab was only available in Excel on the web. This tab represents the first stage of uniting automation solutions across platforms.

To view and run scripts:

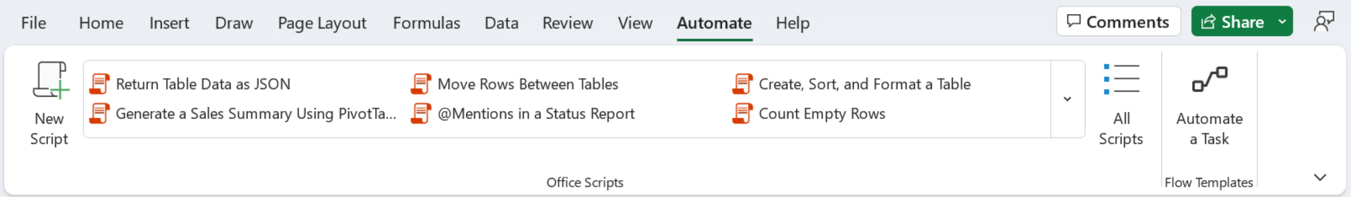
- open any workbook in Excel for Windows or for Mac and select the Automate tab
- to run an existing script, select one in the gallery or in the 'All Scripts' task pane
- in the 'All Scripts' task pane, select any script in the gallery or click the 'All Scripts' button to view more detail
- to run the script you've selected, click the Run button on the script's detail page.


Code Editor
All scripts
Create, Sort, and Format a Table
Office Scripts Sample » ... » Office Scripts
Run
Edit
Description
Script access
Script activity

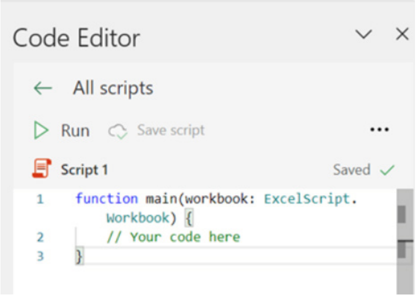
This script creates a table from sample data, color-codes the rows, and sorts the table alphabetically. Sample code for: table creation, table sorting, and range formatting.

To create and modify scripts:

- open any workbook in Excel for Windows or for Mac and navigate to the Automate tab. You should note that all the scripts in your workbook are available as well as various Microsoft samples



- you may start making your own scripts by selecting the 'New Script' button

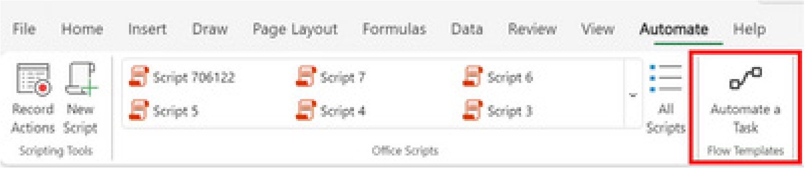


```
function main(workbook: ExcelScript.
    Workbook) {
    // Your code here
}
```

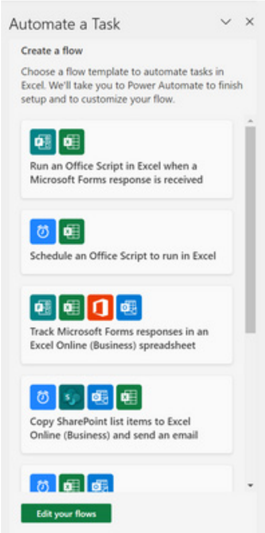
- to modify an existing script, select Edit on the script's details page or select the pencil icon by hovering over any script in the 'All Scripts' task pane.

To connect your automations to other applications:

- in Excel on the web, for Windows or for Mac, open an Excel workbook
- select Automate -> Automate a Task



- select the template you wish to use



Create a flow

Choose a flow template to automate tasks in Excel. We'll take you to Power Automate to finish setup and to customize your flow.

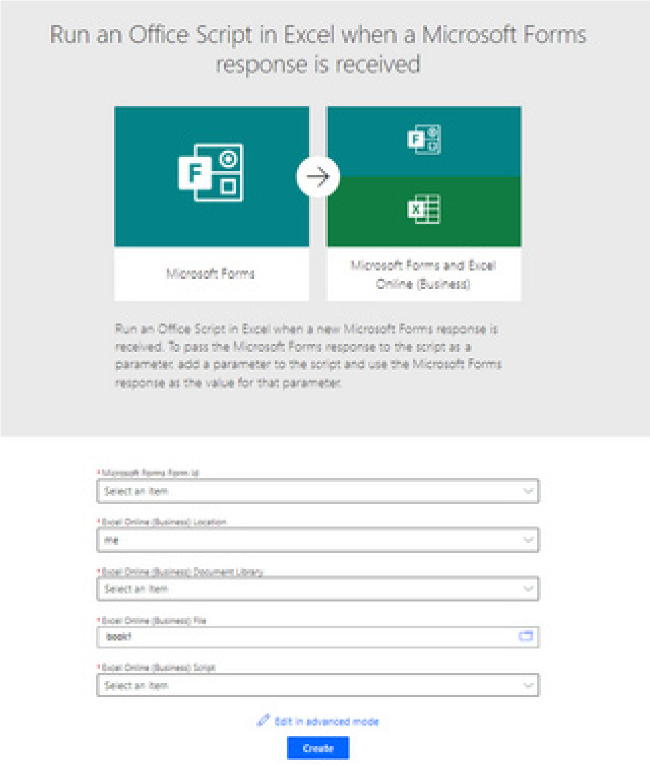
Run an Office Script in Excel when a Microsoft Forms response is received

Schedule an Office Script to run in Excel

Track Microsoft Forms responses in an Excel Online (Business) spreadsheet

Copy SharePoint list items to Excel Online (Business) and send an email

- Sign in, provide the required information, and then select the Create button.



Run an Office Script in Excel when a Microsoft Forms response is received

Run an Office Script in Excel when a new Microsoft Forms response is received. To pass the Microsoft Forms response to the script as a parameter, add a parameter to the script and use the Microsoft Forms response as the value for that parameter.

* Microsoft Forms Form Id: Select an item

* Excel Online (Business) Location: me

* Excel Online (Business) Document Library: Select an item

* Excel Online (Business) File: book1

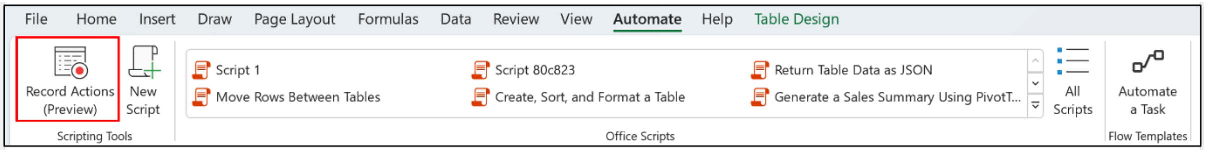
* Excel Online (Business) Script: Select an item

Record worksheet actions using Office Scripts (Insiders)

Up until now, in Excel for Windows and Excel for Mac, you could create and modify automations using JavaScript and TypeScript using the Office Scripts Code Editor only. With the Action Recorder, you can now automate repetitive worksheet tasks without needing any programming experience.

It's very simple:

- open any existing workbook
- select **Automate -> Record Actions (Preview)**



- record actions in your workbook
- when you're done recording, click the 'Stop recording' button in the 'Record Actions (Preview)' task pane to save the recorded actions into a script that you can run at any time on any workbook.

It should be noted that not all actions are recordable at this time: Microsoft is continuing to add support to increase the number of recordable actions. They ask that you continue trying to record actions useful to your workflow as the telemetry received will help them prioritise the order in which support is added.

To use Office Scripts in Excel for Windows or Excel for Mac, you must have the following:

- a stable internet connection
- a commercial (E3 / E5) or Education Microsoft 365 subscription
- for Windows users only: Microsoft Edge WebView2 installed.

This feature is available to Office Insiders running:

- Windows: Version 2212 (Build 15922.20000) or later
- Mac: Version 16.68 (Build 22120101) or later.

Check your formula with Value Preview ToolTips (Insiders)

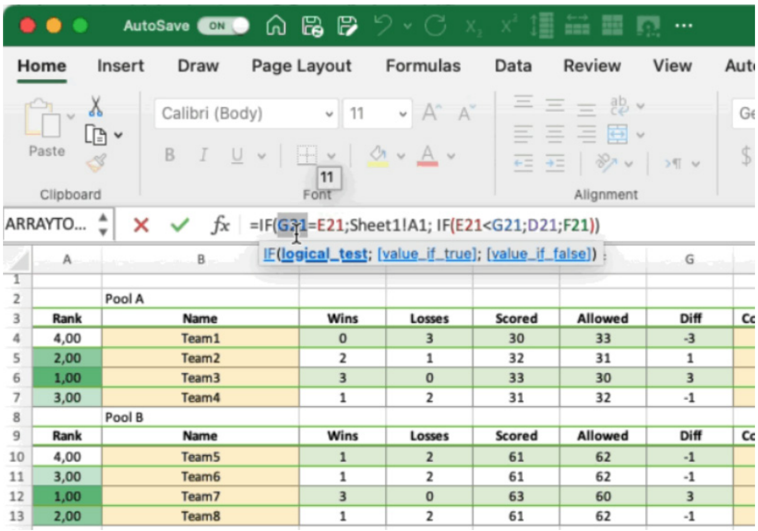
Yes, we know we have already mentioned this right at the beginning of this newsletter, but we also appreciate not everyone reads these things cover to cover *(Even I don't – Ed.)*.

An exciting new feature for Excel for Windows and for Mac, Microsoft has added ToolTips to help you check your formulae directly in the Formula bar or in the cell you're editing. All you need to do is select part of the formula, and Excel will display a ToolTip that contains the current value of the part you highlighted. They're called **Value Preview ToolTips**, a catchy title if ever there wasn't one. But don't let that put you off!

Up until now, a common technique modellers employed to obtain the current value of (part of) a formula was to select the formula (or part thereof) and press **F9**. Doing so would replace (the selected part of) the formula with its current value. You'd probably only do this to check a value, and then you press **ESC** to avoid keeping it as a hard-coded value in your formula.

Alternatively, another approach was to use the 'Evaluate Formula' dialog (**ALT + M + V** or **Formulas -> Evaluate Formula**), which let you check the current value of (each part of) your formula. ToolTips are intended to be easier to use and less burdensome should you just want to focus on one part of the formula.

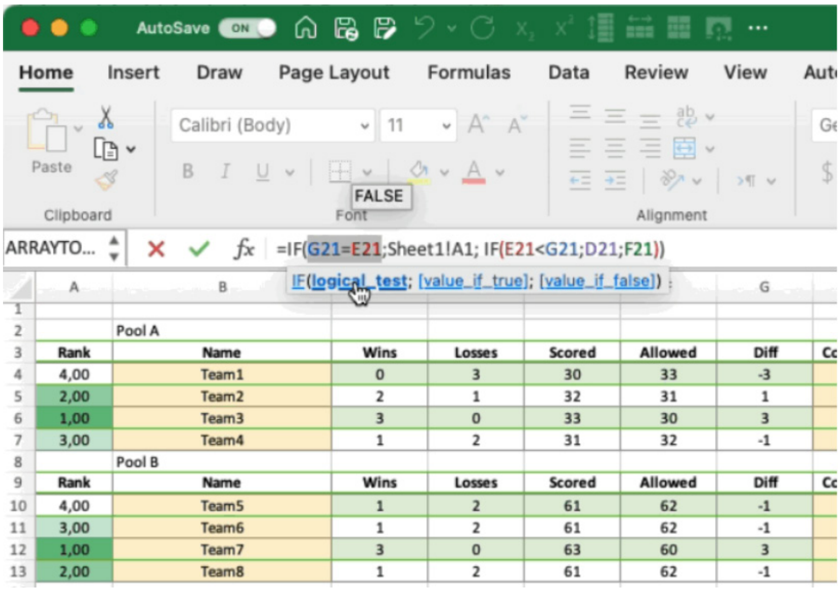
To get this new feature to work, in any formula, simply select the part you wish to evaluate. For example, in a ToolTip that shows the function syntax, click any of the function parameters to select it, and then take note of the ToolTip that appears which shows the current value of that parameter.



=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

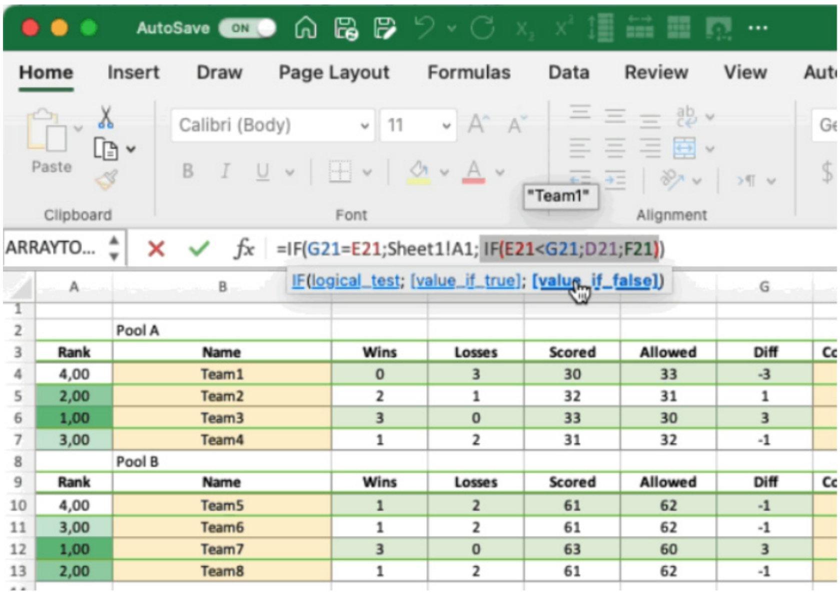
	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Co
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Co
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	



=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	



=IF(G21=E21;Sheet1!A1; IF(E21<G21;D21;F21))

IF(logical_test; [value_if_true]; [value_if_false])

	A	B					G	
1								
2		Pool A						
3	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
4	4,00	Team1	0	3	30	33	-3	
5	2,00	Team2	2	1	32	31	1	
6	1,00	Team3	3	0	33	30	3	
7	3,00	Team4	1	2	31	32	-1	
8		Pool B						
9	Rank	Name	Wins	Losses	Scored	Allowed	Diff	Cc
10	4,00	Team5	1	2	61	62	-1	
11	3,00	Team6	1	2	61	62	-1	
12	1,00	Team7	3	0	63	60	3	
13	2,00	Team8	1	2	61	62	-1	

It should be noted that:

- you may select references, functions, parameters within a function or even the entire formula
- if you'd like to turn off the ToolTips, you can press **CTRL + ALT + P** toggle the feature on or off, but note:
  - on Excel for Mac, this only works when you are not editing a cell
  - on Excel for Windows, it may be toggled on / off anytime (if a ToolTip is visible, pressing the shortcut will not immediately hide the ToolTip, but it will switch the feature off so that subsequent selections won't show the ToolTip).

Thus, in any formula that contains range or table references, you may select a reference and see that the ToolTip shows the value or values in the referenced cells. Further, while editing a formula that has one or more functions, you may also place your cursor inside the brackets of the function so that the syntax ToolTip appears. Then, click one of the parameter names in the syntax to select that part of your formula.

If you select a part of the formula that can't be evaluated, you won't see a ToolTip. For example, if you select only part of a reference or part of a parameter, no ToolTip will be displayed.

It's possible to see a tooltip for part of the formula that doesn't get calculated as part of the cell value. For example, if your formula is =101+201, you could select 1+20 and see a tooltip that shows 21, even though that part isn't relevant when calculating the entire formula.

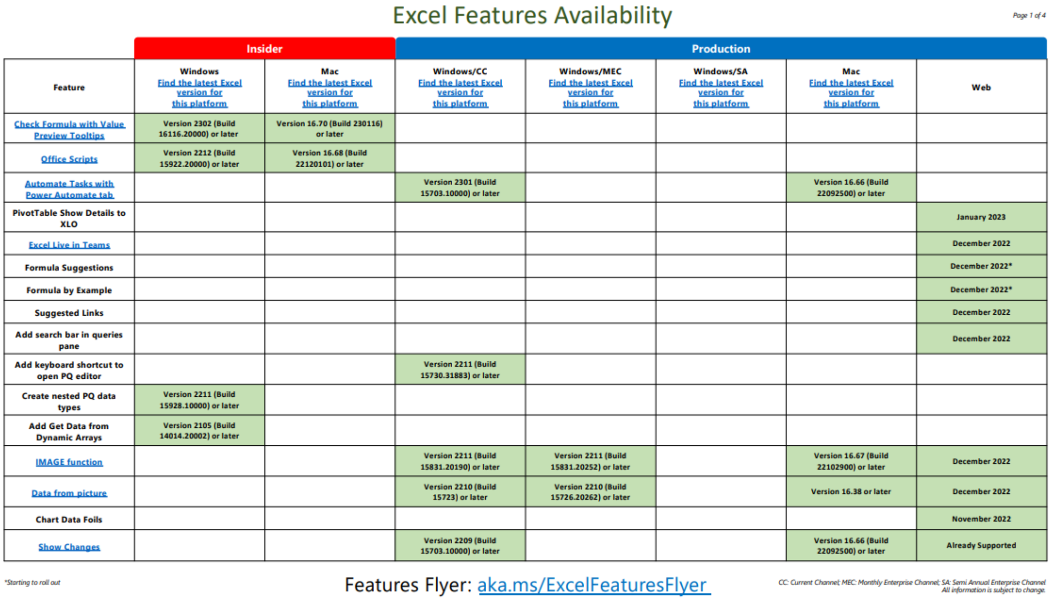
Dates presently show as serial numbers in Excel, rather than how they would be displayed under current locale settings. This *may* change in the future – watch this space!

Finally, you can move the ToolTip if it's covering something you need to see, or if you just want to locate it somewhere else.

This feature is presently rolling out to Beta Channel users running:

- **Windows:** Version 2302 (Build 16116.20000) or later
- **Mac:** Version 16. 70 (Build 230116) or later.

The updated version of the grid with all the new features is fast becoming too complicated to show clearly here. Nonetheless, you can find the interactive links at aka.ms/ExcelFeaturesFlyer.



Excel Features Availability

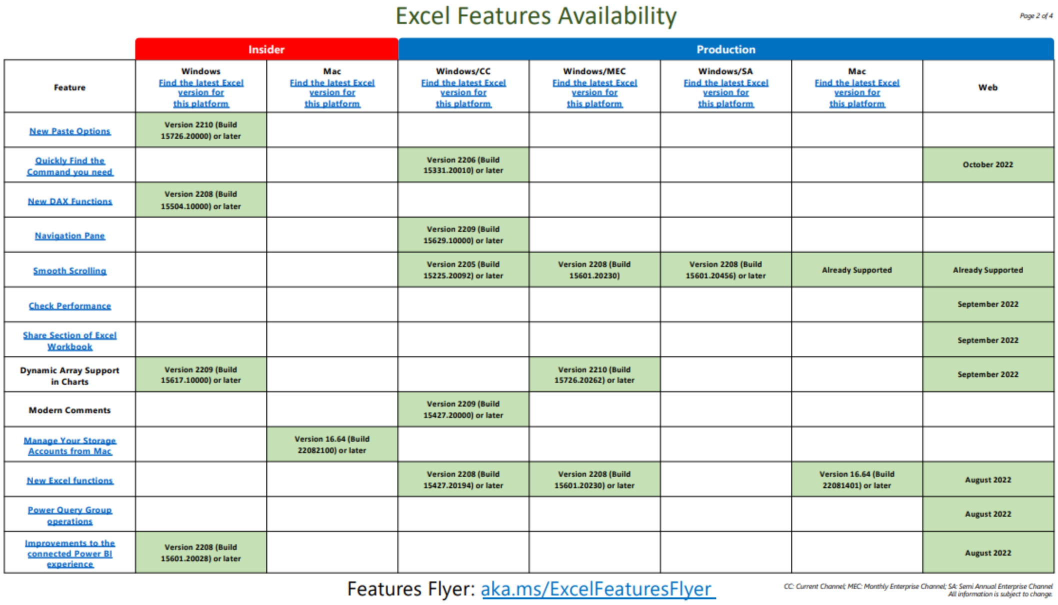
Page 1 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Check Formula with Value Preview Tooltips	Version 2302 (Build 16116.20000) or later	Version 16.70 (Build 230116) or later					
Office Scripts	Version 2212 (Build 15922.20000) or later	Version 16.68 (Build 22120101) or later					
Automate Tasks with Power Automate tab			Version 2301 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	
PivotTable Show Details to XLO							January 2023
Excel Live in Teams							December 2022
Formula Suggestions							December 2022*
Formula by Example							December 2022*
Suggested Links							December 2022
Add search bar in queries pane							December 2022
Add keyboard shortcut to open PQ editor			Version 2211 (Build 15730.31883) or later				
Create nested PQ data types	Version 2211 (Build 15928.10000) or later						
Add Get Data from Dynamic Arrays	Version 2105 (Build 14014.20002) or later						
IMAGE function			Version 2211 (Build 15831.20190) or later	Version 2211 (Build 15831.20252) or later		Version 16.67 (Build 22102900) or later	December 2022
Data from picture			Version 2210 (Build 15723) or later	Version 2210 (Build 15726.20262) or later		Version 16.38 or later	December 2022
Chart Data Foils							November 2022
Show Changes			Version 2209 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	Already Supported

*Starting to roll out

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



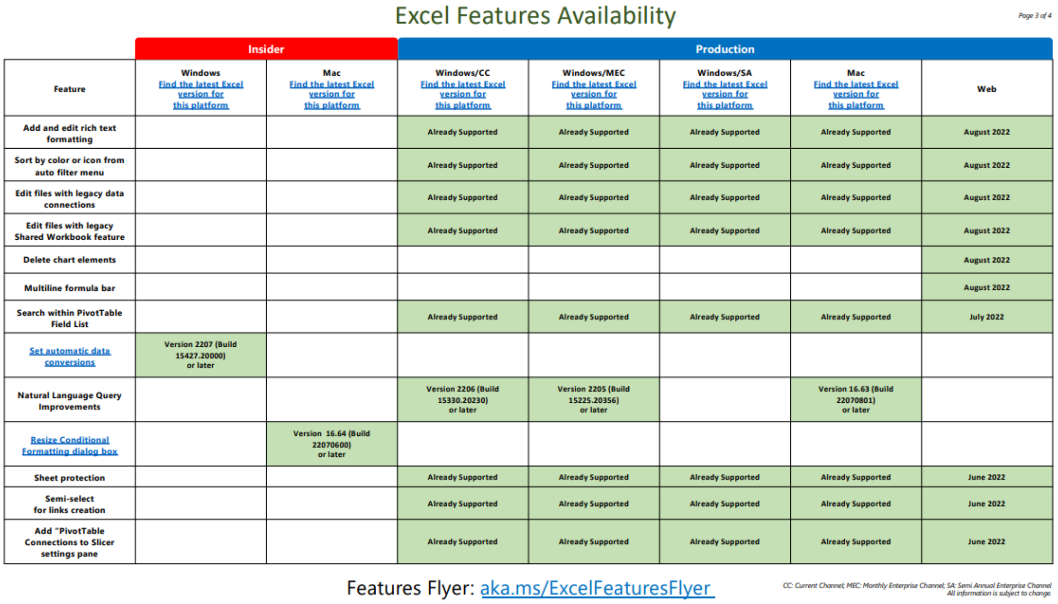
Excel Features Availability

Page 2 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
New Paste Options	Version 2210 (Build 15726.20000) or later						
Quickly Find the Command you need			Version 2206 (Build 15331.20010) or later				October 2022
New DAX Functions	Version 2208 (Build 15504.10000) or later						
Navigation Pane			Version 2209 (Build 15629.10000) or later				
Smooth Scrolling			Version 2205 (Build 15225.20092) or later	Version 2208 (Build 15601.20230)	Version 2208 (Build 15601.20456) or later	Already Supported	Already Supported
Check Performance							September 2022
Share Section of Excel Workbook							September 2022
Dynamic Array Support in Charts	Version 2209 (Build 15617.10000) or later			Version 2210 (Build 15726.20262) or later			September 2022
Modern Comments			Version 2209 (Build 15427.20000) or later				
Manage Your Storage Accounts from Mac		Version 16.64 (Build 22082100) or later					
New Excel functions			Version 2208 (Build 15427.20194) or later	Version 2208 (Build 15601.20230) or later		Version 16.64 (Build 22081401) or later	August 2022
Power Query Group operations							August 2022
Improvements to the connected Power BI experience	Version 2208 (Build 15601.20028) or later						August 2022

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



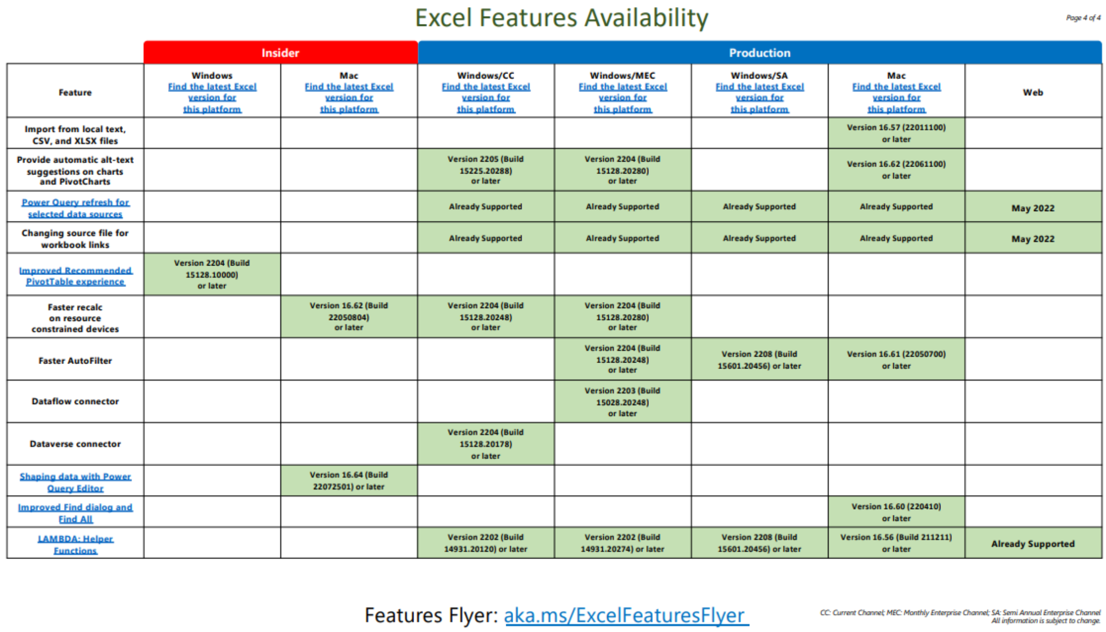
Excel Features Availability

Page 3 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Add and edit rich text formatting			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Sort by color or icon from auto filter menu			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy data connections			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy Shared Workbook feature			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Delete chart elements							August 2022
Multiline formula bar							August 2022
Search within PivotTable Field List			Already Supported	Already Supported	Already Supported	Already Supported	July 2022
Set automatic data conversions	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20356) or later		Version 16.63 (Build 22070801) or later	
Resize Conditional Formatting dialog box		Version 16.64 (Build 22070600) or later					
Sheet protection			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Semi-select for links creation			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Add "PivotTable Connections to Slicer settings pane			Already Supported	Already Supported	Already Supported	Already Supported	June 2022

Features Flyer: aka.ms/ExcelFeaturesFlyer

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



Excel Features Availability

Page 4 of 4

	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Changing source file for workbook links			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Improved Recommended PivotTable experience	Version 2204 (Build 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20178) or later				
Shaping data with Power Query Editor		Version 16.64 (Build 22072501) or later					
Improved Find dialog and Find All						Version 16.60 (220410) or later	
LAMBDA: Helper Functions			Version 2202 (Build 14931.20120) or later	Version 2202 (Build 14931.20274) or later	Version 2208 (Build 15601.20456) or later	Version 16.56 (Build 211211) or later	Already Supported

Features Flyer: aka.ms/ExcelFeaturesFlyer

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

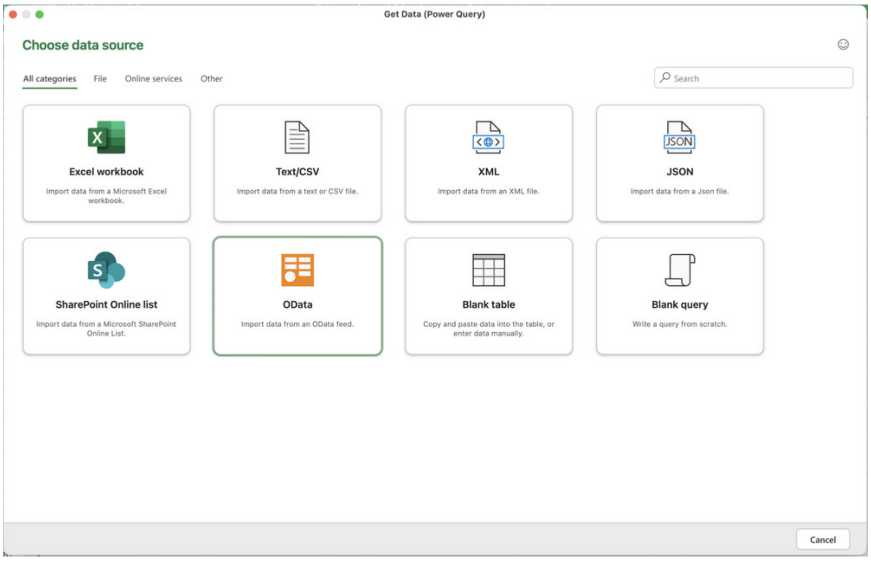
More next month, we're sure.

Import Data from Additional Data Sources using Power Query in Excel for Mac

Power Query in Excel for Mac allows you to get and transform your data from various data sources. Up until now, importing data in Power Query in Excel for Mac was available only from Excel workbooks and Text / CSV files. Following the release of Power Query Editor to the General Audience, you can now import data from XML and JSON files, OData, SharePoint Online List, Blank Query and Blank Table. Being able to import your data from additional sources, helps you gather data quickly and easily, and expands the amount of data you will be able to work with within Excel.

As an example, consider the importation steps for the OData connector:

- select **Data -> Get Data (Power Query)**
- in the 'Choose data source' dialog box, select OData



Excel workbook
Import data from a Microsoft Excel workbook.

Text/CSV
Import data from a text or CSV file.

XML
Import data from an XML file.

JSON
Import data from a Json file.

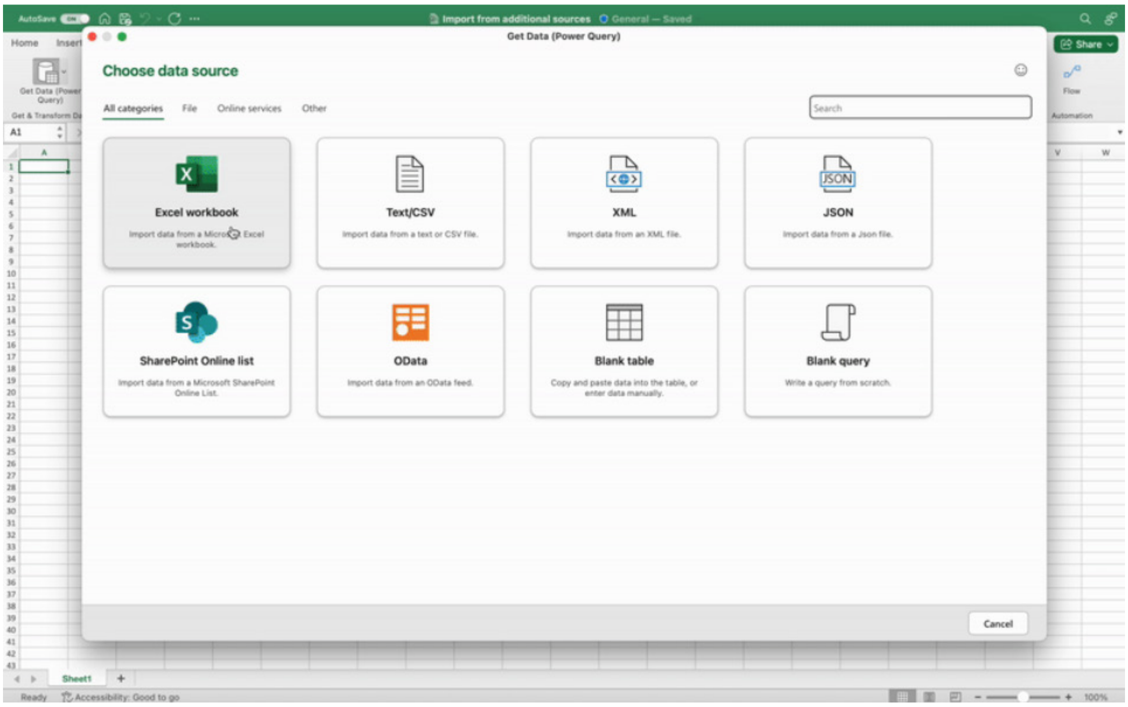
SharePoint Online list
Import data from a Microsoft SharePoint Online List.

OData
Import data from an OData feed.

Blank table
Copy and paste data into the table, or enter data manually.

Blank query
Write a query from scratch.

- in the 'OData Feed' dialog box, enter the URL for the OData feed
- if the OData feed requires user credentials, in the 'Connection credentials' dialog box under Authentication kind:
  - select 'Basic' if the OData feed requires your username and password
  - select 'Organizational account' if the OData feed requires federated access credentials, and select 'Sign in'.
- select Next.



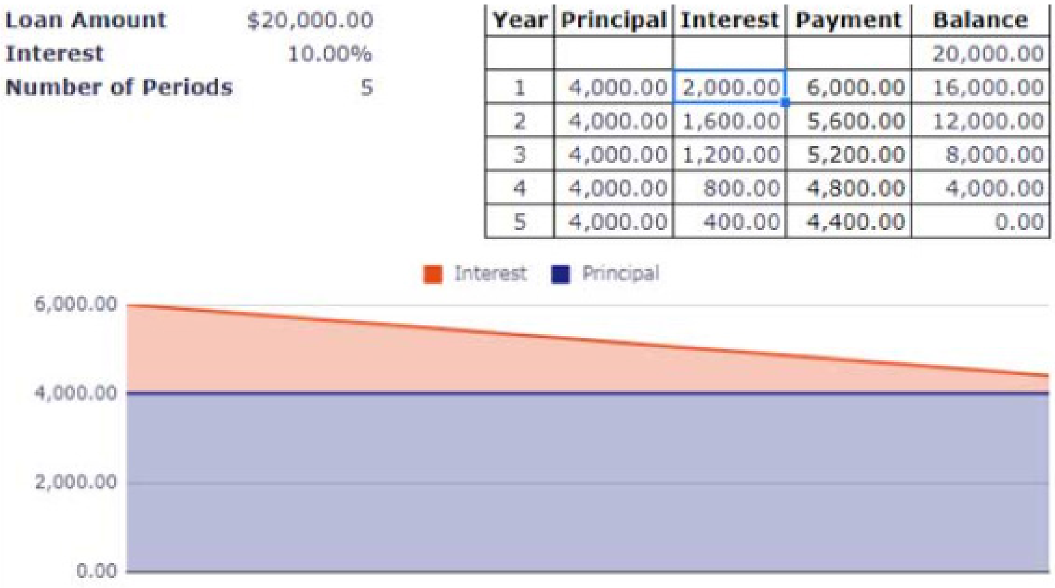
Choose data source

- Excel workbook: Import data from a Microsoft Excel workbook.
- Text/CSV: Import data from a text or CSV file.
- XML: Import data from an XML file.
- JSON: Import data from a Json file.
- SharePoint Online list: Import data from a Microsoft SharePoint Online List.
- OData: Import data from an OData feed.
- Blank table: Copy and paste data into the table, or enter data manually.
- Blank query: Write a query from scratch.

You may explore the various categories of data sources shown in the 'Get Data' dialog, or alternatively, use the Search bar on the top right-hand side of the dialog to easily find the data source you wish to import data from.

This feature is Generally Available in Excel for Microsoft 365 for Mac version 16.69 (23010700) or later.

The A to Z of Excel Functions: ISPMT



Loan Amount	$20,000.00
Interest	10.00%
Number of Periods	5

Year	Principal	Interest	Payment	Balance
				20,000.00
1	4,000.00	2,000.00	6,000.00	16,000.00
2	4,000.00	1,600.00	5,600.00	12,000.00
3	4,000.00	1,200.00	5,200.00	8,000.00
4	4,000.00	800.00	4,800.00	4,000.00
5	4,000.00	400.00	4,400.00	0.00

This function calculates the interest paid (or received) for the specified period of a loan (or an investment) for a constant instant rate with equal principal repayments. In reality, this is quite an easy financial instrument to calculate using basic formulae, but the **ISPMT** function makes it slightly simpler than computing from first principles.

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

ISPMT(rate, per, nper, pv)

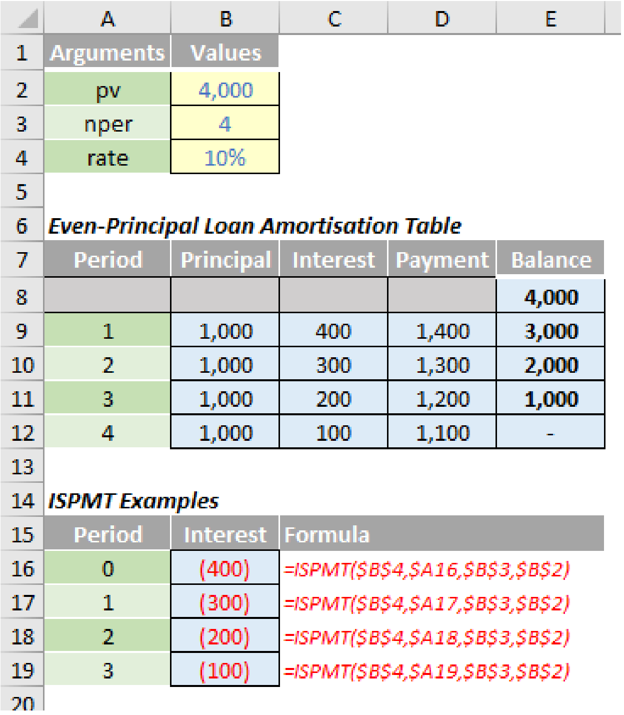
The **ISPMT** function has the following arguments:

- **rate:** this is required and represents the constant interest rate for the loan or investment
- **per:** this is required, and specified the period to be considered, between periods 1 and **nper**
- **nper:** this is also required and denotes the total number of payments for the loan or investment
- **pv:** also necessary, this is the present value, or the total amount that a series of future payments is worth now, also known as the principal (*i.e.* what you are borrowing).

It should be further noted that:

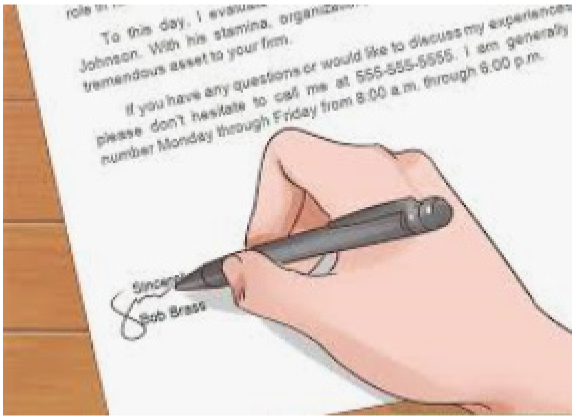
- the payment returned by **PPMT** relates to the principal but considers no taxes, reserve payments or other fees sometimes associated with loans
- make sure that you are consistent about the units you use for specifying **rate** and **nper**. If you make monthly payments on a four-year loan at an annual interest rate of 12%, use 12%/12 for **rate** and 4*12 for **nper**. If you make annual payments on the same loan, use 12% for the **rate** and 4 for **nper**
- **ISPMT** counts each period beginning with zero (0), not with one (1)
- Most loans use a repayment schedule with even periodic payments. The **IPMT** function returns the interest payment for a given period for this type of loan
- Some loans use a repayment schedule with even principal payments. The **ISPMT** function returns the interest payment for a given period for this type of loan
- This is one of Excel's financial functions which distinguishes between cash inflows (positive) and outflows (negative).

Please see our example below:



	A	B	C	D	E
1	Arguments	Values			
2	pv	4,000			
3	nper	4			
4	rate	10%			
5					
6	*Even-Principal Loan Amortisation Table*				
7	Period	Principal	Interest	Payment	Balance
8					4,000
9	1	1,000	400	1,400	3,000
10	2	1,000	300	1,300	2,000
11	3	1,000	200	1,200	1,000
12	4	1,000	100	1,100	-
13					
14	*ISPMT Examples*				
15	Period	Interest	Formula		
16	0	(400)	=ISPMT($B$4,$A16,$B$3,$B$2)		
17	1	(300)	=ISPMT($B$4,$A17,$B$3,$B$2)		
18	2	(200)	=ISPMT($B$4,$A18,$B$3,$B$2)		
19	3	(100)	=ISPMT($B$4,$A19,$B$3,$B$2)		
20					

The A to Z of Excel Functions: ISREF



At the time of writing, there are 12 **IS** functions, *i.e.* functions that give rise to a TRUE or FALSE value depending upon whether a certain condition is met:

1. **ISBLANK(reference):** checks whether the **reference** is to an empty cell
2. **ISERR(value):** checks whether the **value** is an error (*e.g. #REF!, #DIV/0!, #NULL!*). This check specifically excludes *#N/A*
3. **ISERROR(value):** checks whether the **value** is an error (*e.g. #REF!, #DIV/0!, #NULL!*). This is probably the most commonly used of these functions in financial modelling
4. **ISEVEN(number):** checks to see if the **number** is even
5. **ISFORMULA(reference):** checks to see whether the **reference** is to a cell containing a formula
6. **ISLOGICAL(value):** checks to see whether the **value** is a logical (TRUE or FALSE) value
7. **ISNA(value):** checks to see whether the **value** is *#N/A*. This gives us the rather crude identity **ISERR + ISNA = ISERROR**
8. **ISNONTEXT(value):** checks whether the **value** is not text (*N.B.* blank cells are not text)
9. **ISNUMBER(value):** checks whether the **value** is a number
10. **ISODD(number):** checks to see if the **number** is odd. Personally, I find the number 46 very odd, but Excel doesn't
11. **ISREF(value):** checks whether the **value** is a reference
12. **ISTEXT(value):** checks whether the **value** is text.

As stated above, the **ISREF** function checks to see whether the value is a reference. It has the following syntax:

ISREF(value)

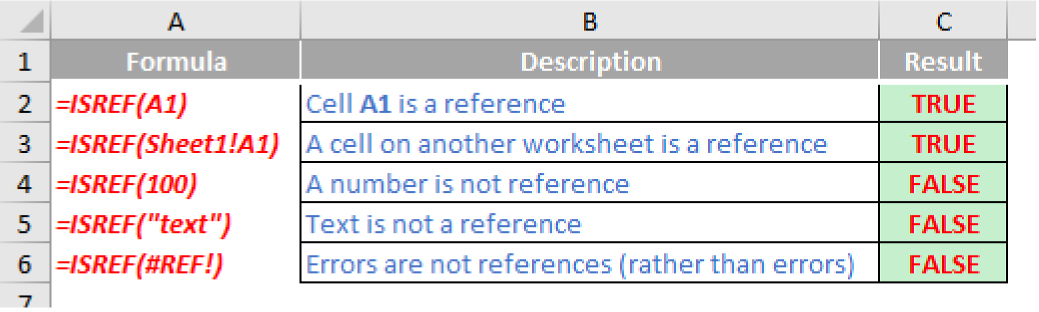
The **ISREF** function has the following argument:

- **value:** this is required and represents the **value** for which you wish to determine whether it is a reference.

It should be further noted that:

- if **value** is not a valid data type, such as a defined name that is not a reference, **ISREF** returns the *#VALUE!* error value.

Please see my example below:



	A	B	C
1	Formula	Description	Result
2	=ISREF(A1)	Cell **A1** is a reference	TRUE
3	=ISREF(Sheet1!A1)	A cell on another worksheet is a reference	TRUE
4	=ISREF(100)	A number is not reference	FALSE
5	=ISREF("text")	Text is not a reference	FALSE
6	=ISREF(#REF!)	Errors are not references (rather than errors)	FALSE
7			

The A to Z of Excel Functions: ISTEXT

The final **IS** function, **ISTEXT**, checks to see whether the value is text. It has the following syntax:

ISTEXT(value)

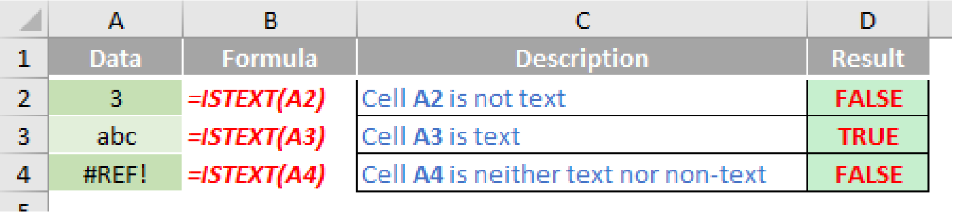
The **ISTEXT** function has the following argument:

- **value:** this is required and represents the **value** for which you wish to determine whether it contains text.

It should be further noted that:

- if **ISTEXT** is FALSE, this does not mean **ISNONTEXT** is TRUE. For example, *#DIV/0!* Is neither text nor non-text; it is an error
- if **value** is not a valid data type, such as a defined name that is not a reference, **ISTEXT** returns the *#VALUE!* error value.

Please see our final example for this month below:



	A	B	C	D
1	Data	Formula	Description	Result
2	3	*=ISTEXT(A2)*	Cell **A2** is not text	FALSE
3	abc	*=ISTEXT(A3)*	Cell **A3** is text	TRUE
4	#REF!	*=ISTEXT(A4)*	Cell **A4** is neither text nor non-text	FALSE

More Excel Functions next month.

Beat the Boredom Suggested Solution

If you host prize draws on your website, social media, or live meetings, you may need to demonstrate a fair way to select random winner(s).

In this challenge, we required you to create a random selector in Excel to choose three winners for a prize draw from a list of 10 people. The tricky part here was that a person could not win the game more than once. Hence, the winners list must not contain any duplicates.

As always, there were some requirements:

- this was a formula challenge; no Power Query / Get & Transform or VBA!
- there must be no duplicate in the result list
- if we changed the number of winners, the formula should still work.

Suggested Solutions

Before explaining our solutions, we will clarify the common idea how we came up with them first.

Brainstorming

Before picking the winners, we need to shuffle the given list first. Therefore, an Excel formula that helps generate random numbers is needed. For example:

- **RAND**: generates a random number between zero [0] and one [1]
- **RANDARRAY**: produces an array of random numbers based upon specific conditions
- **RANDBETWEEN**: returns a random integer between two specified numbers.

Then, we can get a new shuffled list by sorting the old list by random number list. This can be done using **RANK** / **RANK.EQ** or **SORTBY**

Finally, we can pick names as many as we need from the new list from the top to bottom.

Returning to the Suggested Solutions

For Excel Office 365, Excel 2021, Excel on the Web and Office Beta / Insider versions:

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

First, we can generate a dynamic list of random numbers using **RANDARRAY** with the help of **ROWS** as follows.

RANDARRAY(ROWS(List[Name]))

Second, in order to shuffle the name list, we use **SORTBY** function to sort it by random numbers above.

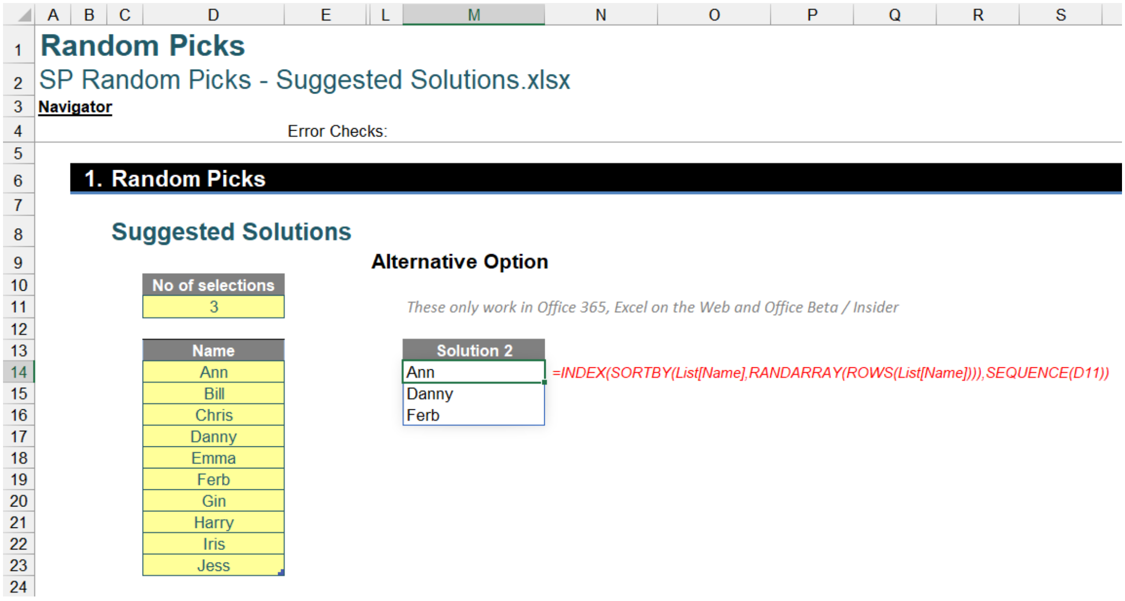
SORTBY(List[Name], RANDARRAY(ROWS(List[Name])))

Finally, **INDEX** and **SEQUENCE** are used to get first three winners from the new sorted list.

=INDEX(SORTBY(List[Name], RANDARRAY(ROWS(List[Name]))), SEQUENCE(D11))

where **D11** is the 'Number of selections' input cell location. This will have the effect of creating the top **X** in a list going down a column.

The resulting list will be a spilled range created by the formula above written in only cell **M14** as below:



Random Picks

SP Random Picks - Suggested Solutions.xlsx

Navigator

Error Checks:

1. Random Picks

Suggested Solutions

Alternative Option

No of selections
3

These only work in Office 365, Excel on the Web and Office Beta / Insider

Name
Ann
Bill
Chris
Danny
Emma
Ferb
Gin
Harry
Iris
Jess

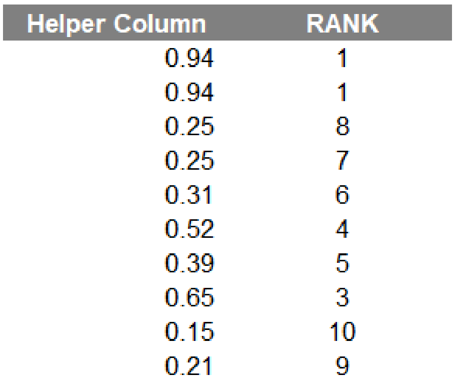
Solution 2
Ann
Danny
Ferb

=INDEX(SORTBY(List[Name],RANDARRAY(ROWS(List[Name]))),SEQUENCE(D11))

For all other Excel versions:

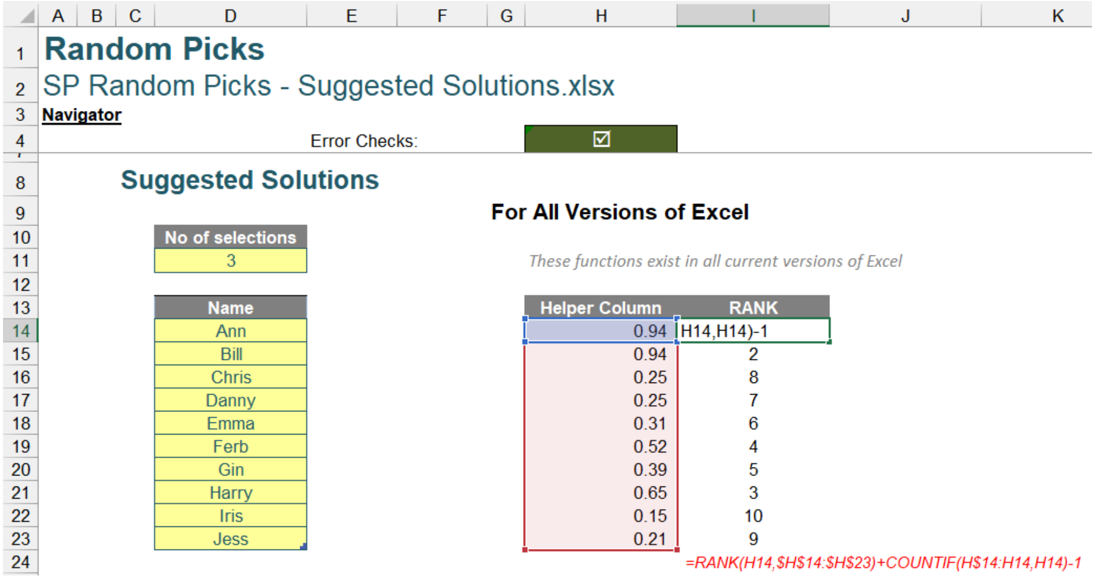
As there is no **SORT** or **SORTBY** (*i.e.* Dynamic Array functions) in other Excel versions, we need a helper column using =**RAND()** to generate a list of random numbers.

Then, we need to find a ranking for each random number above with the help of the **RANK** function. However, as there is a chance (although very small) that two random numbers created by **RAND** are the same, **RANK** will provide the same results for them. For instance, **RANK** ranks the first two rows as one [1], skips two [2] and continues with three [3].



Helper Column	RANK
0.94	1
0.94	1
0.25	8
0.25	7
0.31	6
0.52	4
0.39	5
0.65	3
0.15	10
0.21	9

Hence, we need to modify **RANK** function using **COUNTIF** as follows. **COUNTIF** will help, adding one [1] to any duplicated random numbers that appears subsequently in the helper column.



Random Picks

SP Random Picks - Suggested Solutions.xlsx

Navigator

Error Checks: ☑

Suggested Solutions

For All Versions of Excel

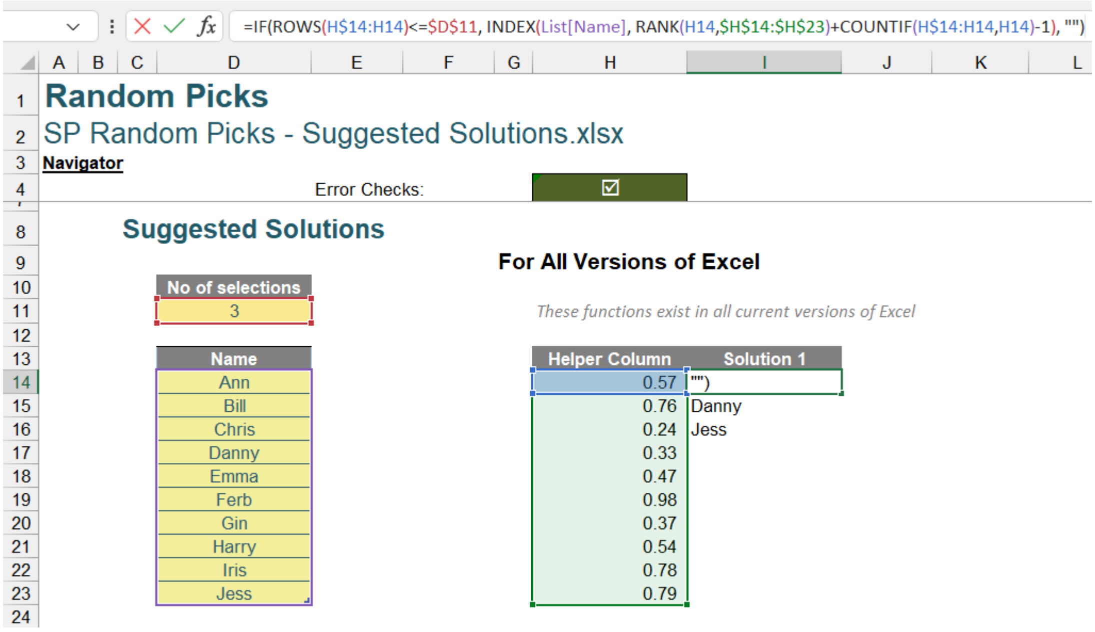
These functions exist in all current versions of Excel

No of selections
3

Name	Helper Column	RANK
Ann	0.94	H14,H14)-1
Bill	0.94	2
Chris	0.25	8
Danny	0.25	7
Emma	0.31	6
Ferb	0.52	4
Gin	0.39	5
Harry	0.65	3
Iris	0.15	10
Jess	0.21	9

=RANK(H14,$H$14:$H$23)+COUNTIF(H$14:H14,H14)-1

Then, we can use **INDEX** within **IF** function to select three winners with the ranked positions above. These ranked numbers can also be considered as a new random list without duplicates.



```
=IF(ROWS(H$14:H14)<=$D$11, INDEX(List[Name], RANK(H14,$H$14:$H$23)+COUNTIF(H$14:H14,H14)-1), "")
```

Random Picks

SP Random Picks - Suggested Solutions.xlsx

Navigator

Error Checks: ☑

Suggested Solutions

For All Versions of Excel

These functions exist in all current versions of Excel

No of selections
3

Name	Helper Column	Solution 1
Ann	0.57	"")
Bill	0.76	Danny
Chris	0.24	Jess
Danny	0.33	
Emma	0.47	
Ferb	0.98	
Gin	0.37	
Harry	0.54	
Iris	0.78	
Jess	0.79	

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	20 - 22 Mar 2023	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	27 Mar 2023	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Online (Australia)	Financial Modelling	28 - 29 Mar 2023	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we look at the **CTRL** and **ALT** keys:

Keystroke	What it does
CTRL + 0	Hide column
CTRL + 1	Format cells
CTRL + 2	Bold (toggle)
CTRL + 3	Italics (toggle)
CTRL + 4	Underline (toggle)
CTRL + 5	Strikethrough (toggle)
CTRL + 6	Toggle Show, Hide and 'Show Placeholders for Objects'
CTRL + 7	Excel 2003 and earlier: toggle Standard toolbar visibility
CTRL + 8	Toggle showing outline symbols
CTRL + 9	Hide row

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

Our Services

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

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

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

Link to Others

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

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

Any Questions?

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

Training

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

Check out our more popular courses in our training brochure:

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

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

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