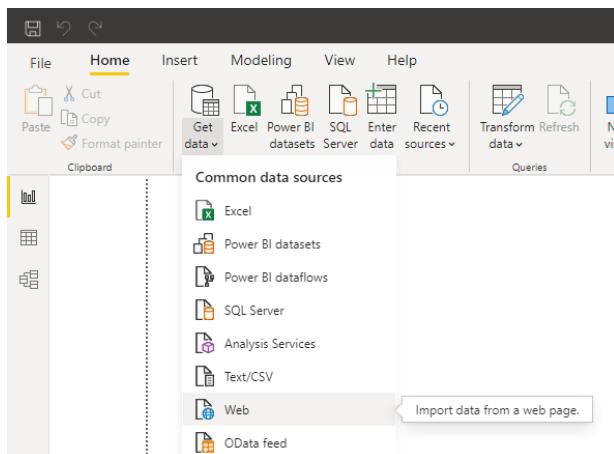


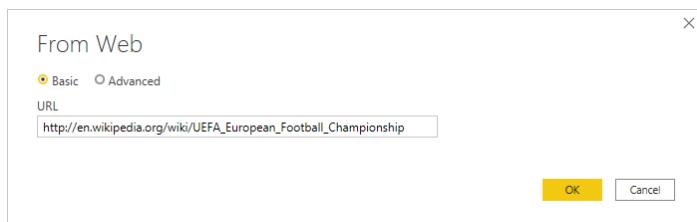
1. Create Report from Web Data Source

1.1 Add a Wikipedia Page Data Source

1. Select **Get Data → Web** in the **Home** ribbon tab.



2. Type the address (http://en.wikipedia.org/wiki/UEFA_European_Football_Championship) in the URL to import a Tournament Summary table from the UEFA European Football Championship Wikipedia, and click [OK] button to continue



3. After establishing a connection to the web page, you see a list of tables available on this Wikipedia page in the Navigator dialog. You can single-click on each of these tables to preview the data. In the **Navigator** left-pane, select the *Results[edit]* table and select [**Transform Data**].

Column1	Column2	Column3	Column4	Column5
Year	Host		Final	Final
1960Details	France	null	null Winners	Score
1964Details	Spain		null Soviet Union	2–1 (a.)
1968Details	Italy		null Spain	2–1
1972Details	Belgium		null Italy	1–1 (a.)
1976Details	Yugoslavia		null West Germany	3–0
1980Details	Italy		null Czechoslovakia	2–2 (a.)
1984Details	France		null West Germany	2–1
1988Details	West Germany		null Final	Final
1992Details	Sweden		null Winner	Score
1996Details	England		null France	2–0
2000Details	Belgium Netherlands		null Netherlands	2–0
2004Details	Portugal		null Germany	2–1 (g.)
2008Details	Austria Switzerland		null Greece	1–0
2012Details	Poland Ukraine		null Spain	4–0
2016Details	France		null Portugal	1–0 (a.)
2020(c)Details	Pan-European		null	
2024Details	Germany		null	

4. This will allow us to reshape this table before loading it to the Report, since the data is not in the shape that we need for our analysis

The screenshot shows the Power Query Editor interface with a query named 'Results[edit]'. The query is a transformation of the original table, adding columns for 'Year', 'Host', 'Final', 'Score', and various country names like 'Soviet Union', 'Spain', 'Italy', etc. A context menu is open on the right, showing options like 'Properties', 'Applied Steps', and 'Transform'. The 'Transform' section includes 'Use First Row as Headers' and 'Remove Other Columns'.

1.2 Data Sharpening and Cleansing

1.2.1 Remove Other Columns to only Display columns of interest

1. Select **Use First Row as Headers** in the Home ribbon tab.

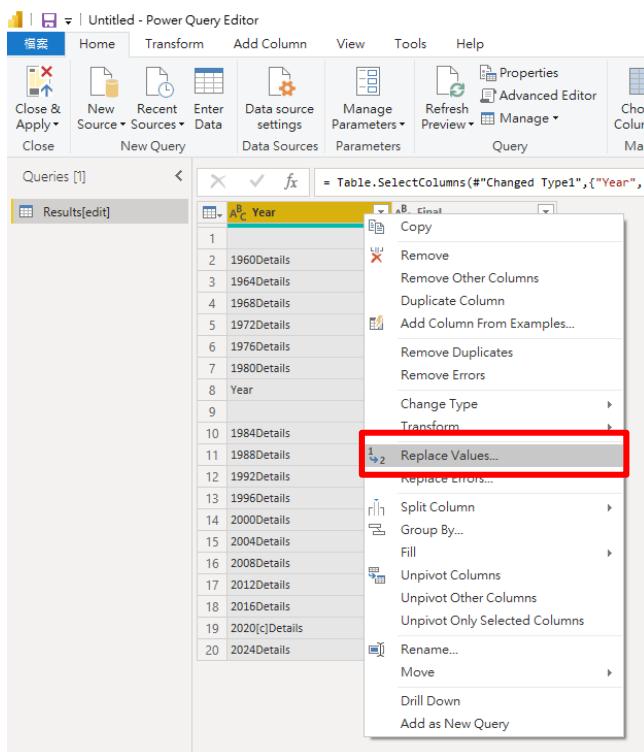
This screenshot shows the Power Query Editor's ribbon with the 'Home' tab selected. The 'Use First Row as Headers' button is highlighted with a red box. Below the ribbon, a tooltip says 'Promote the first row of this table into column headers.'

2. In the query preview grid, select the *Year* and *Final* columns. Right-click to select **Remove Other Columns** to remove the unselected columns.

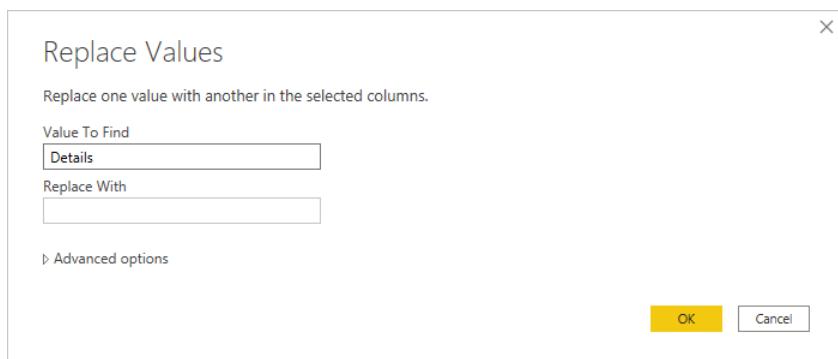
This screenshot shows the Power Query Editor with the 'Results[edit]' query. The 'Year' and 'Final' columns are selected. A context menu is open on the right, with 'Remove Other Columns' highlighted with a red box. Other options in the menu include 'Add Column from Examples...', 'Remove Duplicates', 'Remove Errors', 'Replace Values...', 'Fill', 'Change Type', 'Transform', 'Merge Columns', 'Group By...', 'Unpivot Columns', 'Unpivot Other Columns', 'Unpivot Only Selected Columns', and 'Move'.

1.2.2 Clean up Values in Selected Column

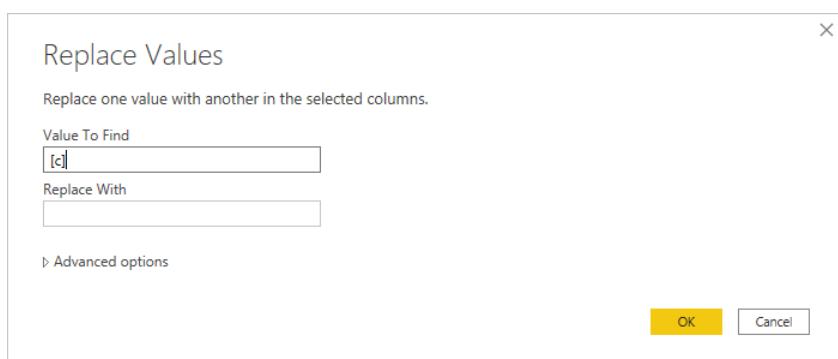
- Select the *Year* column, and right click to select **Replace Value**.



- In the **Replace Values** dialog box, type *Details* in the **Value to Find** text box and leave the **Replace With** text box empty. Click **[OK]** button to start.



- Select the *Year* column and right click to select **Replace Value** again. In the **Replace Values** dialog box, type */c* in the **Value to Find** text box and leave the **Replace With** text box empty. Click **[OK]** button to start.



1.2.3 Filter Values in Column

1. Click the filter drop down arrow on the *Year* column.
2. In the Filter drop-down, clear the *null* and *Year* option and click [OK].

The screenshot shows the Power Query Editor interface. A dropdown menu for the 'Year' column is open, specifically the 'Text Filters' section. Under 'Text Filters', the '1900' and 'Year' checkboxes are checked. At the bottom of the dialog, the 'OK' button is highlighted with a yellow background.

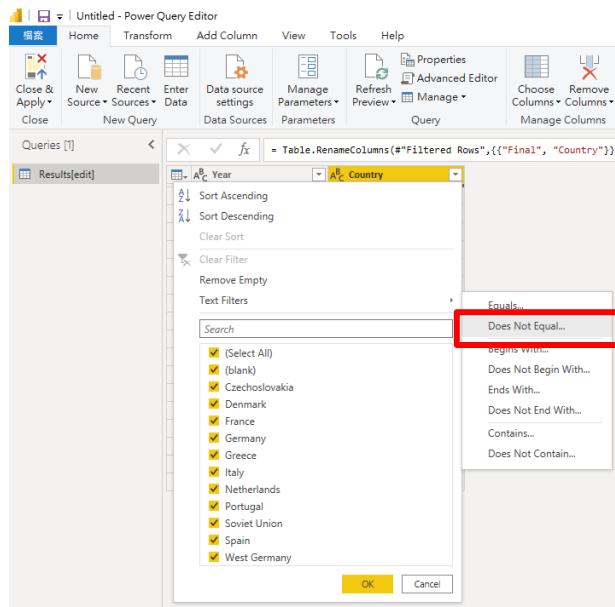
1.2.4 Rename Column

1. Select the *Final* column in the **Query Editor**, and then right click to select **Rename**.
2. Change the name of this column to “Country”.

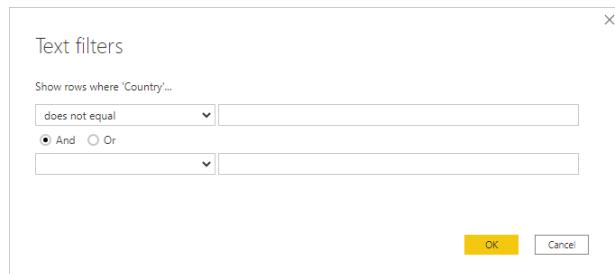
The screenshot shows the Power Query Editor interface. A context menu is open over the 'Final' column. The 'Rename...' option is highlighted with a red box. Other options visible in the menu include Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates, Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Split Column, Group By..., Fill, Unpivot Columns, Unpivot Other Columns, and Unpivot Only Selected Columns.

1.2.5 Filter out blank Values in Column

1. Right-click on one of the cells in the *Country* column that contain a null value.
2. Select **Text Filters → Does not Equal** in the context menu. This creates a new filter step to remove rows with space values in the Country column.



3. Press [OK] to start replacement.



1.2.6 Name a query

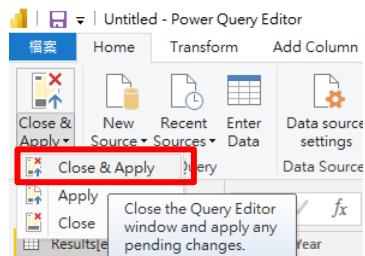
1. In the **Query Settings** pane, in the **Name** text box, enter **Euro Cup Winners**.

The screenshot shows the Power BI desktop interface. On the right side, there is a 'Query Settings' pane. Under the 'PROPERTIES' section, the 'Name' field is set to 'Euro Cup Winners'. Below that, the 'APPLIED STEPS' section lists various steps taken during the query transformation, including 'Extracted Table from Html', 'Changed Type', 'Promoted Headers', 'Added Index', 'Removed Other Columns', 'Replaced Value', 'Replaced Value1', 'Filtered Rows', 'Renamed Columns', 'Filtered Rows1', and 'Filtered Rows2'. The main workspace shows a table with columns 'Year' and 'Country', and 15 rows of data. The top ribbon has tabs for Home, Transform, Add Column, View, Tools, and Help. The ribbon below the tabs includes Close & Apply, New Source, Recent Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Choose Columns, Remove Columns, and Manage Columns.

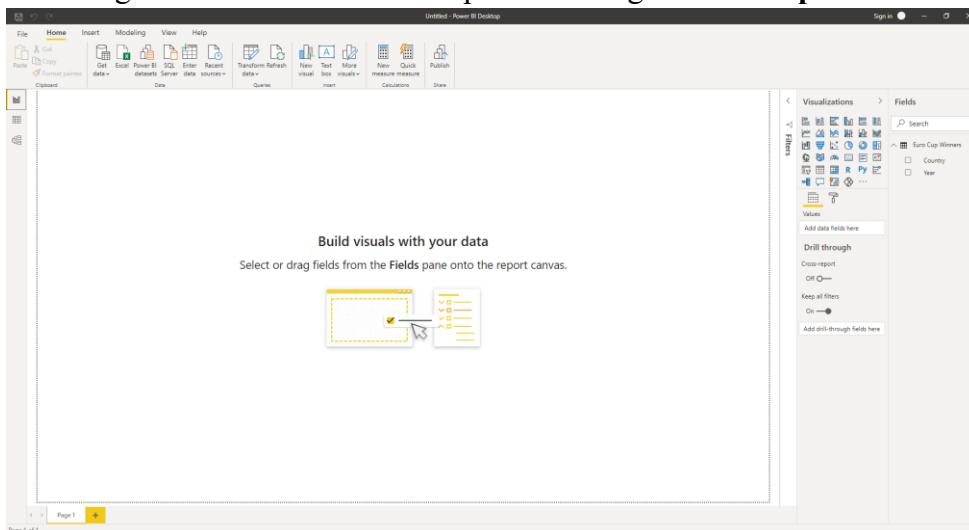
1.3 Create Visualizations using the Report View

1.3.1 Load the Query to Report

- To load the query results to Power BI Desktop and create a report, we select **Close & Apply** from the **Home** ribbon.

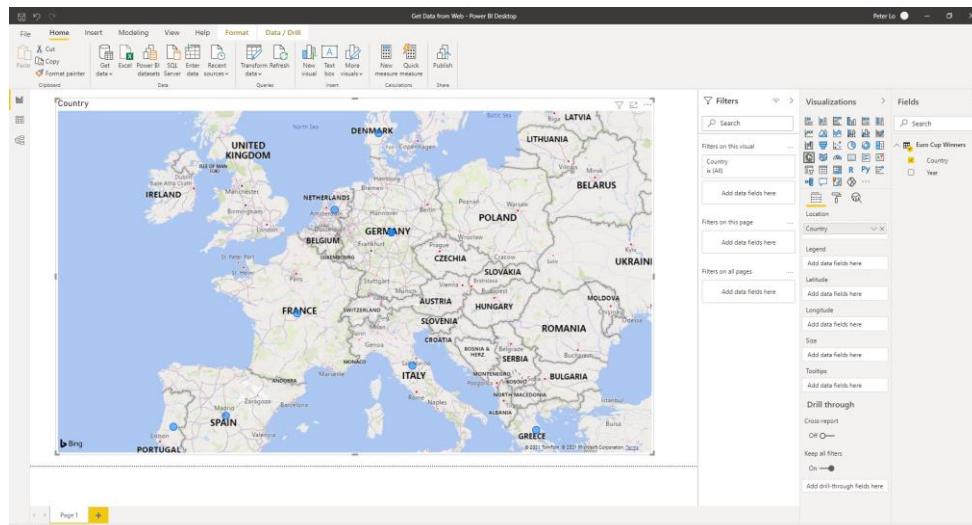


- This will trigger evaluation of the query and load of the table output to the Report. In Power BI Desktop, select the **Report** icon to see Power BI Desktop in **Report** view. You can see the resulting table fields in the **Fields** pane at the right of the **Report** view.

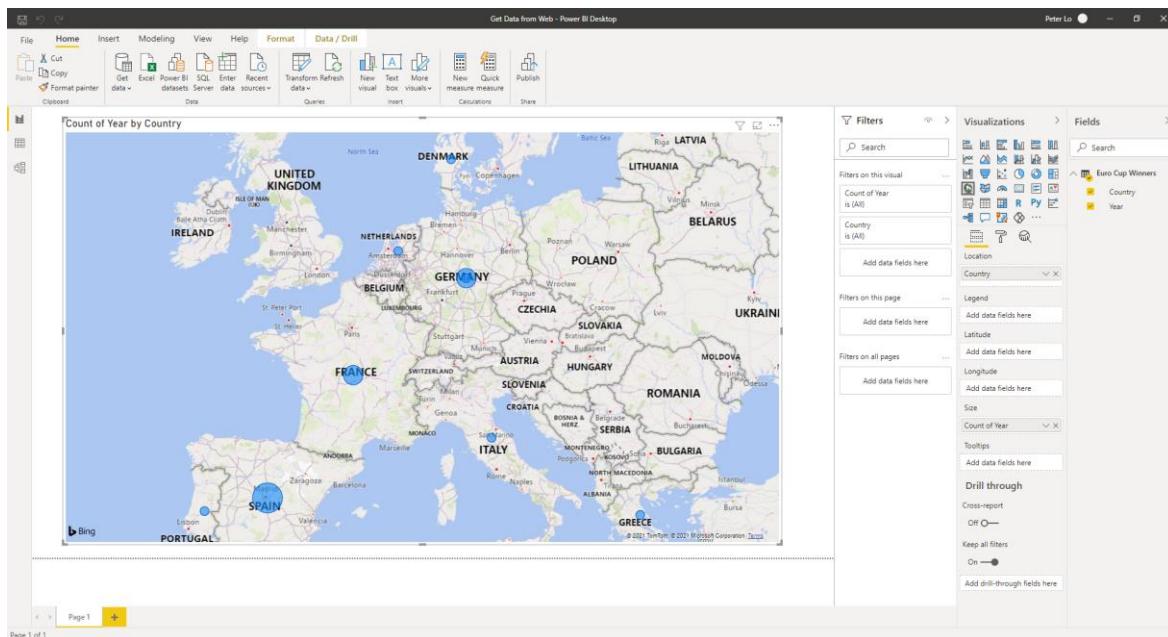


1.3.2 Create a Map Visualization

- Drag the *Country* field and drop it in the **Report** canvas. This will create a new visualization in the **Report** canvas. In this case, since we have a list of countries, it will create a **Map** visualization.



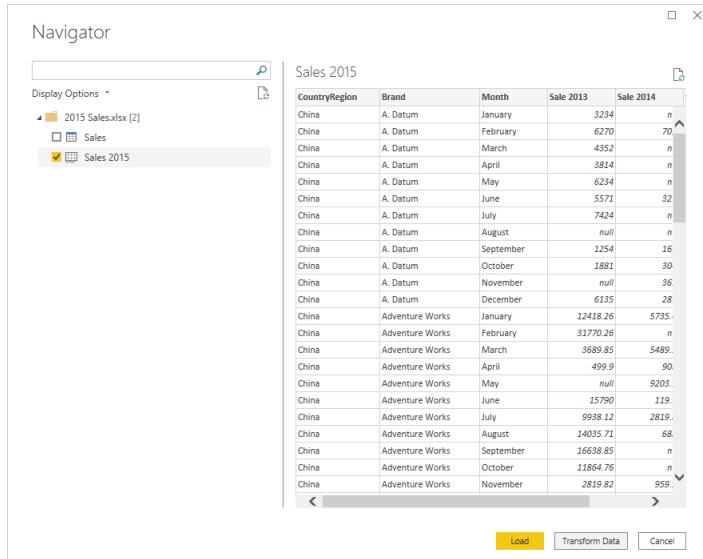
2. We can resize the visualization by dragging from one of the corners of the visualization up to the desired size. Note that currently all the points in the map have the same size. We want to change this so that countries with more Euro Cup tournaments won are represented with a larger point in the map. In order to do this, we can drag the *Year* field in the **Fields** list to the **Values** box in the lower half of the Fields pane.



2. Merge Query for Analysis

2.1 Open First Excel Workbook

1. Select **Get Data → Excel** in the **Home** ribbon tab,
2. In the **Open File** dialog box, select the *2015 Sales.xlsx* file. Then in the **Navigator** pane, select the *Sales 2015* table and then select **[Transform Data]**.

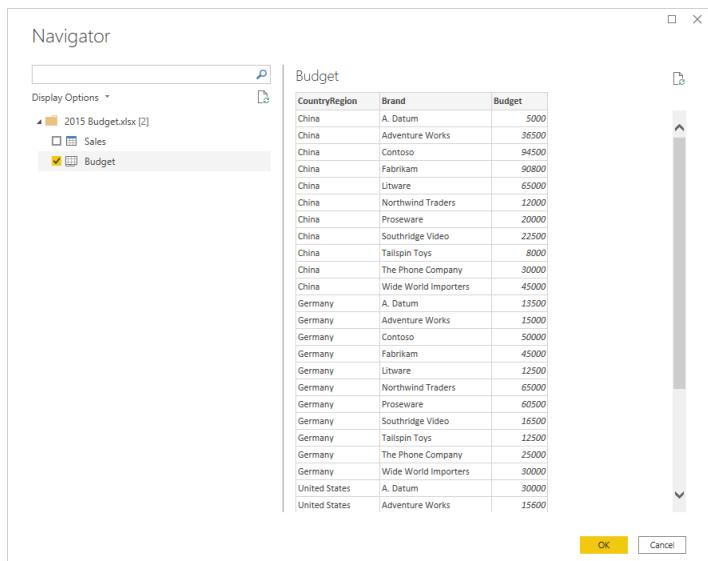


3. The file should be loaded successfully to the Power BI

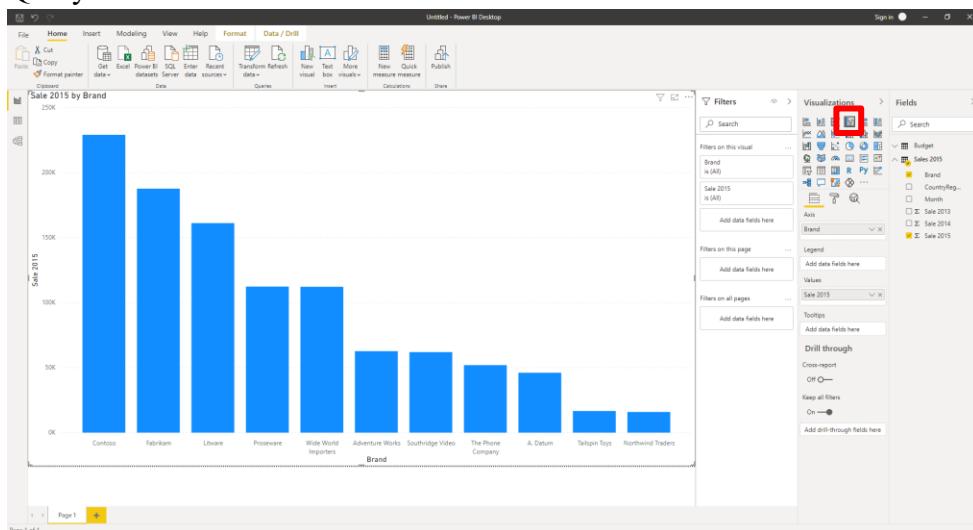
2.2 Open Second Excel Workbook

1. Select **New Source → Excel** in the **Home** ribbon tab

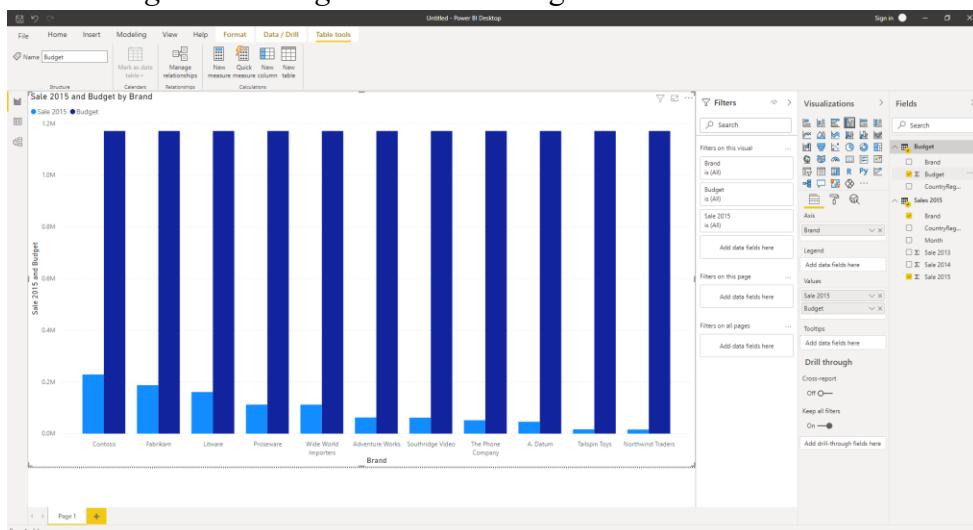
2. In the **Open File** dialog box, select the *2015 Budget.xlsx* file. Then in the **Navigator** pane, select the **Budget** table and then select [**Edit**].



3. The file should be loaded successfully to the Power BI. Select [**Close and Apply**] to confirm. Then create a **Clustered Column Chart** by selecting *Brand* and *Sales 2015* from *Sales 2015* Query.

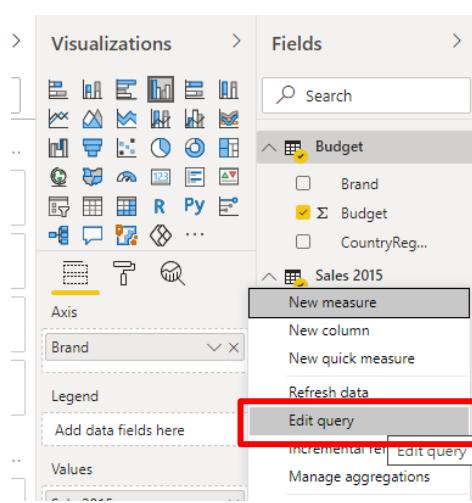


4. Select *Budget* from *Budget 2015*. The budget is failed to load.

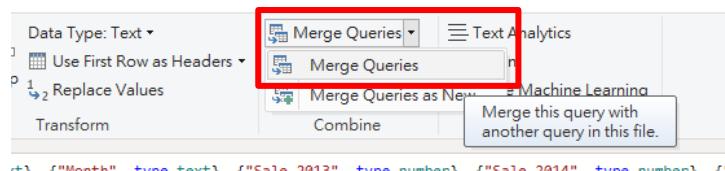


2.3 Merge Query

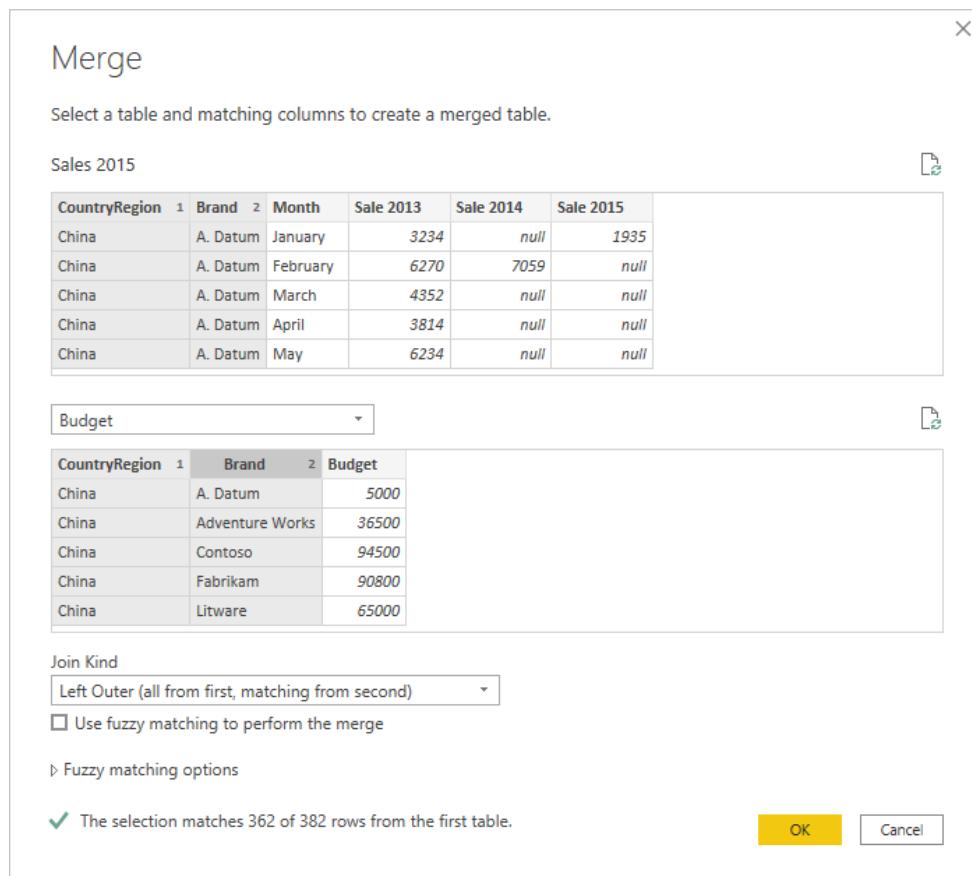
- Right click the query *Sales 2015*, and then select **Edit Query**.



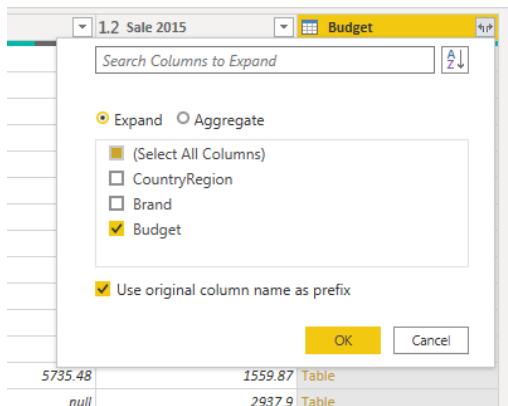
- Select **Merge Query → Merge Query in Home tab** in the Query Editor



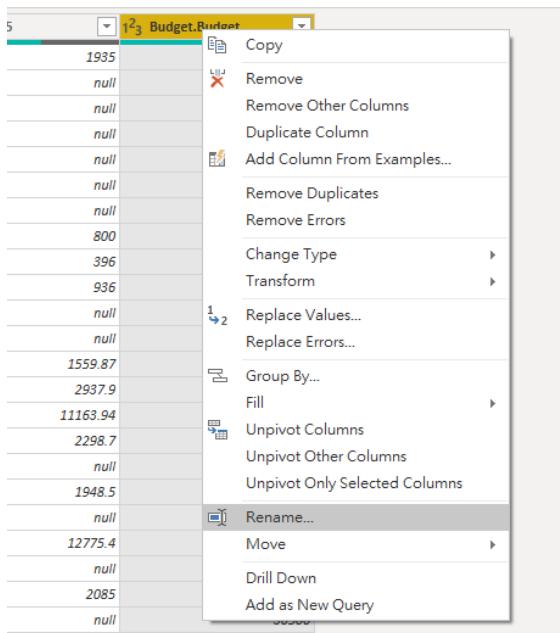
- Select field *CountryRegion* and *Brand* in query *Sales 2015*, and also select field *CountryRegion* and *Brand* in query *Budget*. Then select **Left Outer (all from first, matching from second)** as **Join Kind** and press **[OK]**.



4. Expand to *Budget* column and select *Budget* field only, and press [OK].



5. Right click the column *Budget.Budget* and select **Rename**.



6. Change the column name to **Budget 2015**.

The screenshot shows the Power Query Editor with the 'Budget' table loaded. The 'Budget' column has been renamed to 'Budget 2015'. The 'Renamed Columns' step is listed in the Applied Steps pane.

7. Press [Close & Apply].

The screenshot shows the Power Query Editor interface. The main area displays two tables: 'Sales 2015' and 'Budget'. The 'Budget' table has been modified with new column names: 'Brand', 'Month', '1.2 Sale 2013', '1.2 Sale 2014', '1.2 Sale 2015', and 'Budget 2015'. The 'CountryRegion' column is still present but has been moved to the left. The 'Sales 2015' table is also visible. The 'Query Settings' pane on the right shows the 'Renamed Columns' step.

2.4 Create Combined Report

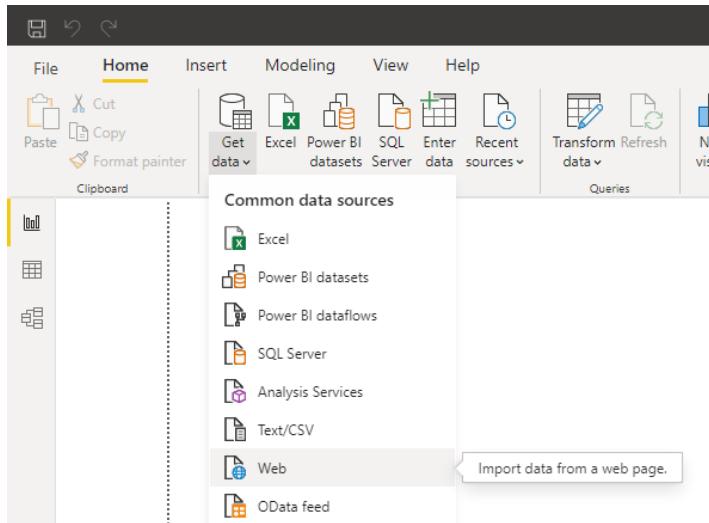
- Replace the value from *Budget* → *Budget* to *Sales* → *Budget 2015* to the Sales Report

The screenshot shows a bar chart in Power BI Desktop titled 'Sale 2015 and Budget 2015 by Brand'. The chart compares sales and budget for ten different brands. The Y-axis represents monetary values from 0.0M to 3.0M. The X-axis lists brands: Contoso, Fabrikam, Litware, Proseware, Wide World Importers, Adventure Works, Southridge Video, The Phone Company, A. Datum, Tailspin Toys, and Northwind Traders. The chart uses blue bars for Sales 2015 and dark blue bars for Budget 2015. The Power BI interface shows various tools like 'Filters', 'Visualizations', and 'Fields' on the right side.

3. Add New Column from Example

3.1 Import Data from Web

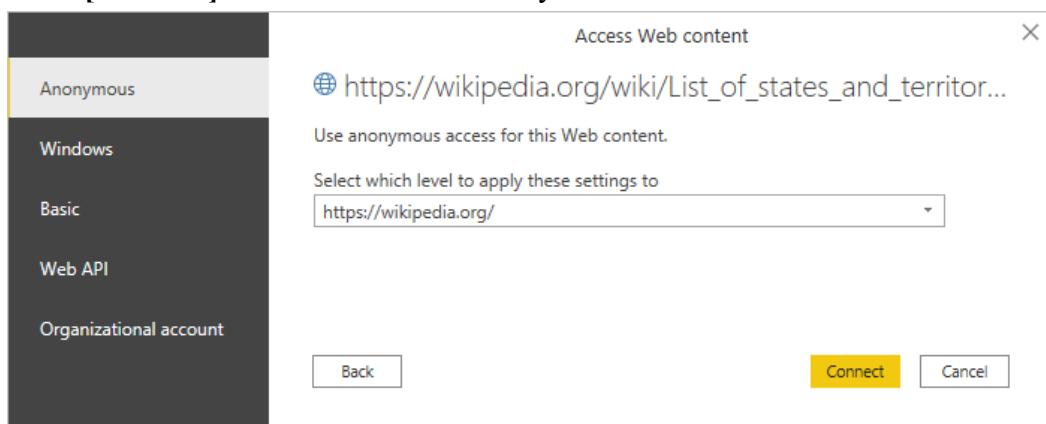
1. Select **Get Data → Web** in the **Home** ribbon tab.



2. Type https:// wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States in the URL to import a Tournament Summary table from the UEFA European Football Championship Wikipedia, and click [OK] button to continue



3. Press [Connect] to confirm that use anonymous access for this web content.



4. Select the *States of the United States of America* table and select [Transform Data].

The screenshot shows the Power BI Navigator window. On the left, there's a tree view of tables under 'HTML Tables [18]'. The 'States of the United States of America' table is selected and expanded, showing its columns and rows. The main area displays the table data:

Flag, name and postal abbreviation[12]	Flag, name and postal abbreviation[12]_1
null	null
Alabama	AL
Alaska	AK
Arizona	AZ
Arkansas	AR
California	CA
Colorado	CO
Connecticut	CT
Delaware	DE
Florida	FL
Georgia	GA
Hawaii	HI
Idaho	ID
Illinois	IL
Indiana	IN
Iowa	IA
Kansas	KS
Kentucky[E]	KY
Louisiana	LA
Maine	ME
Maryland	MD

At the bottom, there are buttons for 'Add Table Using Examples', 'Load', 'Transform Data', and 'Cancel'.

3.2 Add a New Column from Examples

- Once **Query Editor** is launched and you have some data loaded, you can get started adding a column from examples. To add a new column, select the **Add Column** tab on the ribbon and select **Column from Examples**.

The screenshot shows the Power Query Editor ribbon with the 'Add Column' tab selected. Below the ribbon, the 'Column From Examples' button is highlighted with a red box. A dialog box titled 'Column From Examples' is open, containing the text 'Use examples to create a new column in this table. (Ctrl+E)'. It shows the formula `= Table.TransformColumnType` and the column name `Flag, name and postal abbreviation[12]`. The formula bar also shows `ABC Flag, name and postal abbreviation[12]`.

- Double click **Column1** and change the column name to “*Part of State & Postal*”

The screenshot shows the 'Add Column From Examples' dialog box. It displays a table with multiple columns, each with a checkbox. The last column, labeled 'Part of State & Postal', has a yellow background and is highlighted. At the bottom right, there are 'OK' and 'Cancel' buttons.

3. Once you type your example in the new column, Power BI gives you a preview of how the column it's creating will appear, based on the transformations it detects. For example, we typed Alabama in the first row, corresponding to the Alabama value in the first column of the table. As soon as we hit [Enter] Power BI fills in the column based on that value.

The screenshot shows the Power Query Editor interface with the following details:

- File, Home, Transform, Add Column, View, Tools, Help** menu bar.
- Queries** pane: States of the United States.
- Add Column From Examples** dialog box:
 - Transform: "#Flag, name and postal abbreviation[12]"
 - Preview: Shows the first 10 rows of the table with the new column populated.
- Properties** pane: Name: States of the United States of America; Source: Extracted Table from Html; Promoted Headers: City, Capital, State Abbreviation, Largest[16], Founded[16], Population[0][14]; Applied Steps: Column From Examples.
- Applied Steps** pane: Contains the step "Column From Examples".
- Table Preview**: Shows 13 columns and 51 rows of state data, including columns for Flag, name and postal abbreviation, Cities, Cities_2, Ratification ordination, and Population.

4. Then we went to the row that included Massachusetts[E] and deleted that last [e] portion and Power BI detected the change, and used the example to create a transformation. Press [OK] once you confirm your input.

The screenshot shows the Power Query Editor interface with the following details:

- File, Home, Transform, Add Column, View, Tools, Help** menu bar.
- Queries** pane: States of the United States.
- Add Column From Examples** dialog box:
 - Transform: Text.BeforeDelimiter("#Flag, name and postal abbreviation[12]", ".")
 - Preview: Shows the first 10 rows of the table with the new column populated.
- Properties** pane: Name: States of the United States of America; Source: Extracted Table from Html; Promoted Headers: City, Capital, State Abbreviation, Largest[16], Founded[16], Population[0][14]; Applied Steps: Column From Examples.
- Applied Steps** pane: Contains the step "Column From Examples".
- Table Preview**: Shows 13 columns and 51 rows of state data, including columns for Flag, name and postal abbreviation, Cities, Cities_2, Ratification ordination, and Population.

4. Create Custom Column

4.1 Import Product Data from Excel

4.1.1 Connect to Excel Workbook

1. Select **Get Data → Excel** in the **Home** ribbon tab,
2. In the **Open File** dialog box, select the *product.xlsx* file. Then in the **Navigator** pane, select the **Products** table and then select **[Transform Data]**.

The screenshot shows the 'Navigator' pane in Power BI. On the left, there's a tree view of the 'Products.xlsx' file, with 'Products' selected. The main area displays the 'Products' table with the following data:

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	Chai	1	1	10 boxes x 20 bags
2	Chang	1	1	12 oz bottles
3	Aniseed Syrup	1	2	24 - 550 ml bottles
4	Chef Anton's Cajun Seasoning	2	2	24 - 6 oz jars
5	Chef Anton's Gumbo Mix	2	2	36 boxes
6	Grandma's Boysenberry Spread	3	2	10 boxes x 20 bags
7	Uncle Bob's Organic Dried Pears	3	3	12 - 8 oz jars
8	Northwoods Cranberry Sauce	3	3	12 oz jars
9	Mishi Kobe Niku	4	4	36 boxes
10	Ikura	4	4	8 12 - 200 ml jars
11	Queso Cabrales	5	4	12 - 1 lb pkgs.
12	Queso Manchego La Pastera	5	4	24 - 550 ml bottles
13	Konbu	6	5	1 kg pkgs.
14	Tofu	6	5	10 boxes x 20 bags
15	Genen Shouyu	6	5	12 - 8 oz jars
16	Pavlova	7	6	24 - 550 ml bottles
17	Alice Mutton	7	6	10 boxes x 20 bags
18	Carnarvon Tigers	7	6	12 oz bottles
19	Teatime Chocolate Biscuits	8	7	36 boxes
20	Sir Rodney's Marmalade	8	7	12 - 1 lb pkgs.
21	Sir Rodney's Scones	8	7	12 oz jars
22	Gusta's Knäckebrot	9	8	10 boxes x 20 bags
23	Tunnbröd	9	8	12 oz bottles

At the bottom right of the pane, there are 'Load', 'Transform Data', and 'Cancel' buttons.

4.1.2 Remove other Columns to only Display Columns of Interest

1. In **Query Editor**, select the *ProductID*, *ProductName*, *QuantityPerUnit* and *UnitsInStock* columns.
2. Right-click on a column header and click **Remove Other Columns**.

The screenshot shows the Power Query Editor with the 'Products' table loaded. The table has columns: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, and UnitsInStock. A context menu is open over the 'SupplierID' column, and the 'Remove Other Columns' option is highlighted with a red box.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	Chai	1	1	10 boxes x 20 bags	18	
2	Chang	1	1	12 oz bottles	19	
3	Aniseed Syrup	1	2	24 - 550 ml bottles	10	
4	Chef Anton's Cajun Seasoning	2	2	24 - 6 oz jars	22	
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25	
7	Uncle Bob's Organic Dried Pears	3	3	12 - 1 lb pkgs.	30	
8	Northwoods Cranberry Sauce	3	3	12 - 12 oz jars	40	
9	Mishi Kobe Niku	4	4	36 boxes	97	
10	Ikura	4	4	8 12 - 200 ml jars	31	
11	Queso Cabrales	5	4	1 kg pkgs.	21	
12	Queso Manchego La Pastera	5	4	10 - 500 g pkgs.	38	
13	Konbu	6	5	8 2 kg box	6	
14	Tofu	6	5	40 - 100 g pkgs.	28.25	
15	Genen Shouyu	6	5	24 - 250 ml bottles	15.5	
16	Pavlova	7	6	32 - 500 g boxes	17.45	
17	Alice Mutton	7	6	20 - 1 kg tins	39	
18	Carnarvon Tigers	7	6	16 kg plgs.	62.5	

4.1.3 Change the Data Type of Column

1. Select the *UnitsInStock* column.
2. Right-click on column header and select **Change Type** → **Whole Number**.

The screenshot shows the Power Query Editor interface with the 'Products' query selected. A context menu is open over the 'UnitsInStock' column header, specifically on the 'Change Type' submenu. The 'Whole Number' option is checked and highlighted with a red box. Other options like 'Decimal Number', 'Text', and 'True/False' are also visible in the dropdown menu.

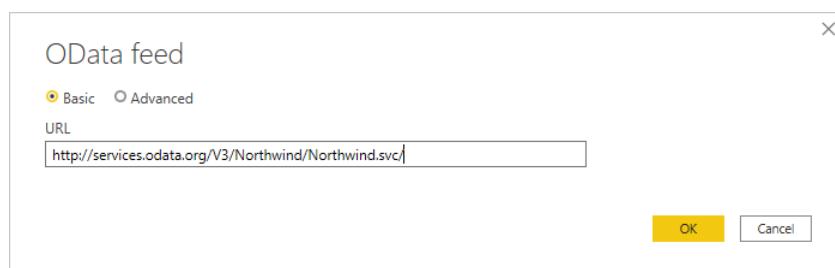
4.2 Import Order Data from OData Feed

4.2.1 Connect to OData Feed

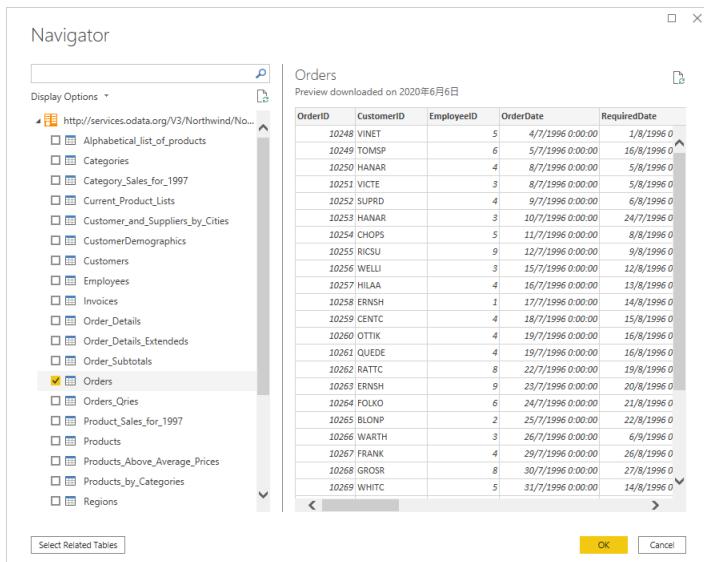
1. From the Home ribbon tab in Query Editor, select **New Source** → **OData feed**.

The screenshot shows the Power Query Editor ribbon with the 'Home' tab selected. Under the 'New Source' button, the 'OData feed' option is highlighted with a red box. Other options like 'Excel', 'SQL Server', 'Analysis Services', 'Text/CSV', and 'Web' are also listed.

2. Input <http://services.odata.org/V3/Northwind/Northwind.svc/> in the **URL** field of the **OData Feed** dialog box and press **[OK]** button.



3. In the Navigator pane, select the *Orders* table, and then select [OK].



4.2.2 Expand the Order Details Table

1. In the Query View, scroll to the *Order_Details* column and select the expand icon .

The screenshot shows the Power Query Editor interface. The 'Orders' query is selected. In the Data Preview pane, the 'Order_Details' column is highlighted with a red box. The column contains a list of records, each representing an order detail with fields like ProductID, UnitPrice, and Quantity.

2. Select *ProductID*, *UnitPrice*, and *Quantity* in the Expand drop-down, and then click [OK].

The screenshot shows the 'Expand' dialog box. Under the 'Select All Columns' section, 'ProductID', 'UnitPrice', and 'Quantity' are checked. The 'OK' button is highlighted with a yellow box.

4.2.3 Remove Other Columns to only Display Columns of Interest

- In the Query View, select all columns by click the first column (*OrderID*), and then press [Shift] to click the last column (*Shipper*).
- Press [Ctrl] and click to unselect the following columns: *OrderDate*, *ShipCity*, *ShipCountry*, *Order_Details.ProductID*, *Order_Details.UnitPrice* and *Order_Details.Quantity*.

3. Right-click on any selected column header and click Remove Columns.

The screenshot shows the Power Query Editor interface with a table loaded. A context menu is open over the 'Order_Details' column header, with the 'Remove Column' option highlighted by a red box. Other options like 'Add Column From Examples...', 'Merge Columns', and 'Unpivot' are also visible in the menu.

4.2.4 Calculate the Line Total for each Order_Details row

1. In the Add Column ribbon tab, click Custom Column.

The screenshot shows the Power Query Editor ribbon with the 'Add Column' tab selected. The 'Custom Column' button is highlighted with a red box. Below the ribbon, a 'Custom Column' dialog box is open, prompting the user to 'Create a new column in this table, based on a custom formula.'

2. In the Custom Column dialog box, enter `[Order_Details.UnitPrice] * [Order_Details.Quantity]` in the Custom Column Formula textbox. Then enter *LineTotal* in the New column name textbox. Click [OK] to continue.

The screenshot shows the 'Custom Column' dialog box. The 'New column name' field contains 'LineTotal'. The 'Custom column formula' field contains the formula `= [Order_Details.UnitPrice]*[Order_Details.Quantity]`. To the right, a list of available columns is shown, with 'Order_Details.Quantity' highlighted. At the bottom, a message says 'No syntax errors have been detected.' and there are 'OK' and 'Cancel' buttons.

4.2.5 Set the Datatype of the LineTotal field

- Right click the *LineTotal* column, and select **Change Type → Decimal Number**.

4.2.6 Rename and Reorder Columns in the Query

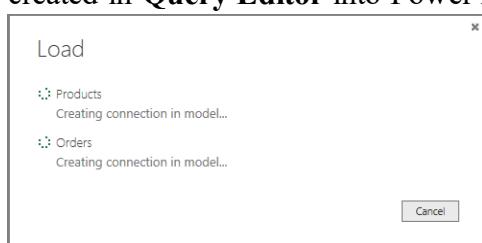
- In **Query Editor**, drag the *LineTotal* column to the left, after *ShipCountry*.

- Rename the column name by removing the *Order_Details.* prefix from the *Order_Details.ProductID*, *Order_Details.UnitPrice* and *Order_Details.Quantity* columns, by double-clicking on each column header, and then deleting that text from the column name.

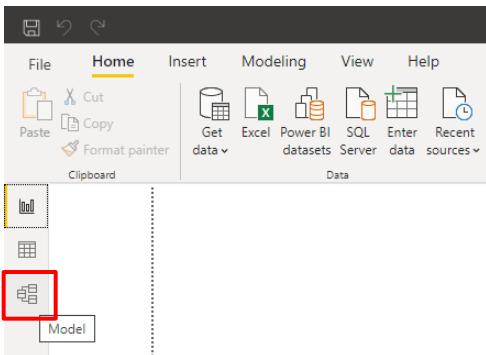
4.3 Combine the Products and Total Sales Queries

4.3.1 Confirm the Relationship between Products and Total Sales

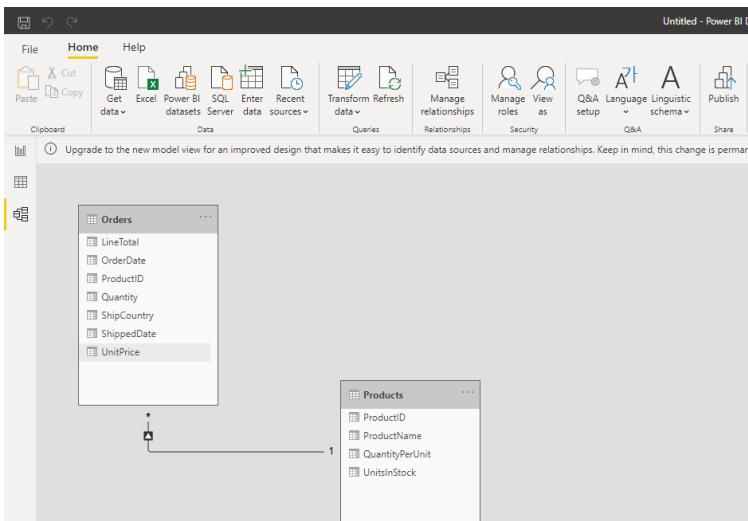
- From the **Home** ribbon of **Query Editor**, select **Close & Apply** to load the model that we created in **Query Editor** into Power BI Desktop.



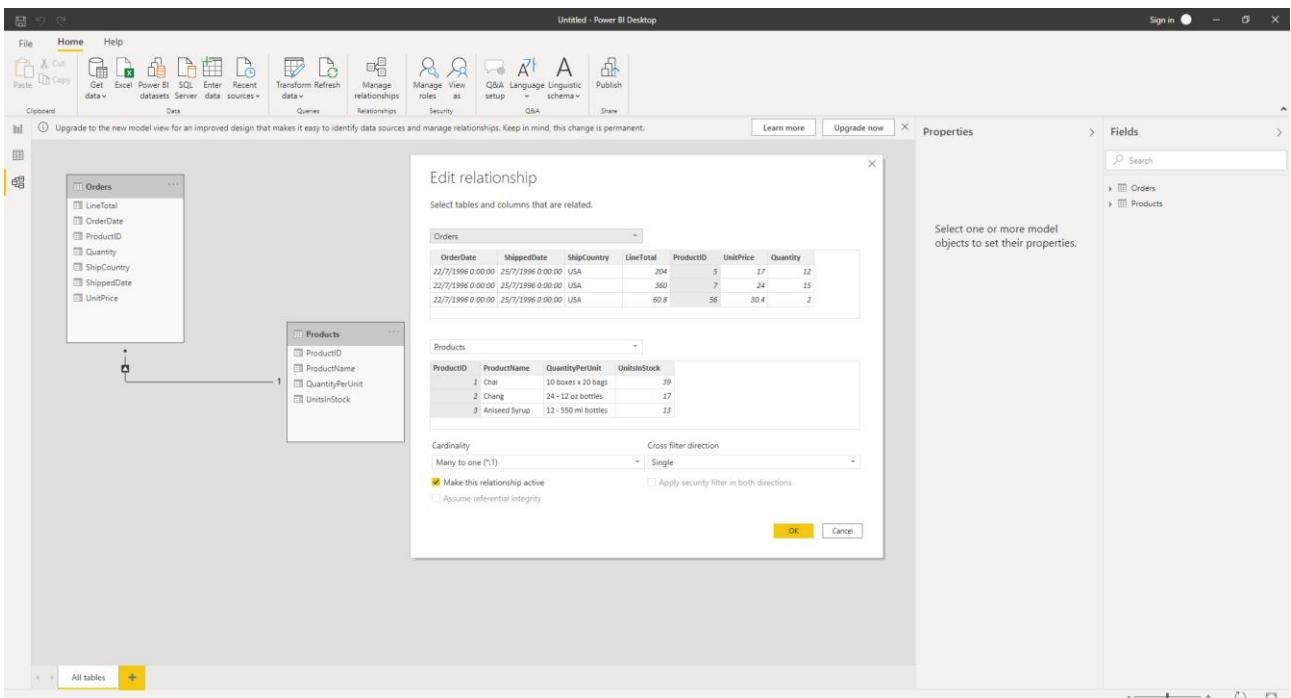
2. Once the data is loaded, Select **Relationship** view.



3. The relationship between the queries are visualized



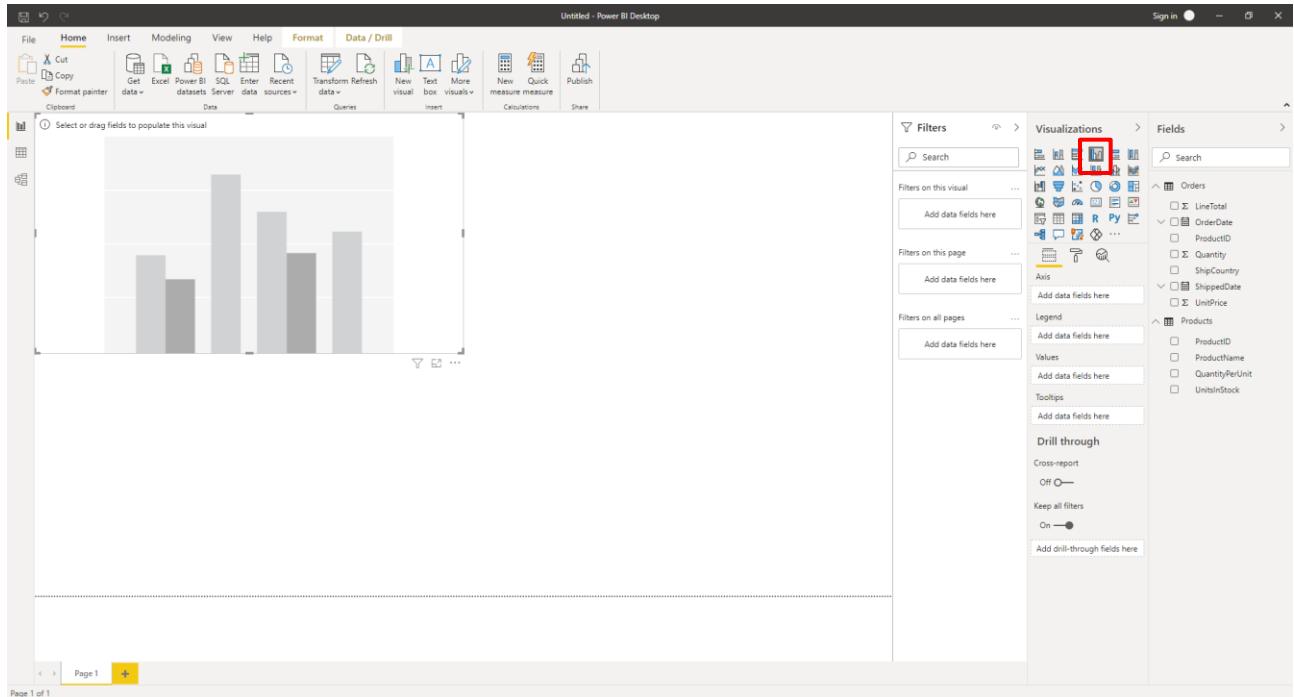
4. When you double-click the arrow on the line that connects the to queries, an **Edit Relationship** dialog appears. Since we don't need to make any changes, so we'll just select **[Cancel]** to close the **Edit Relationship** dialog.



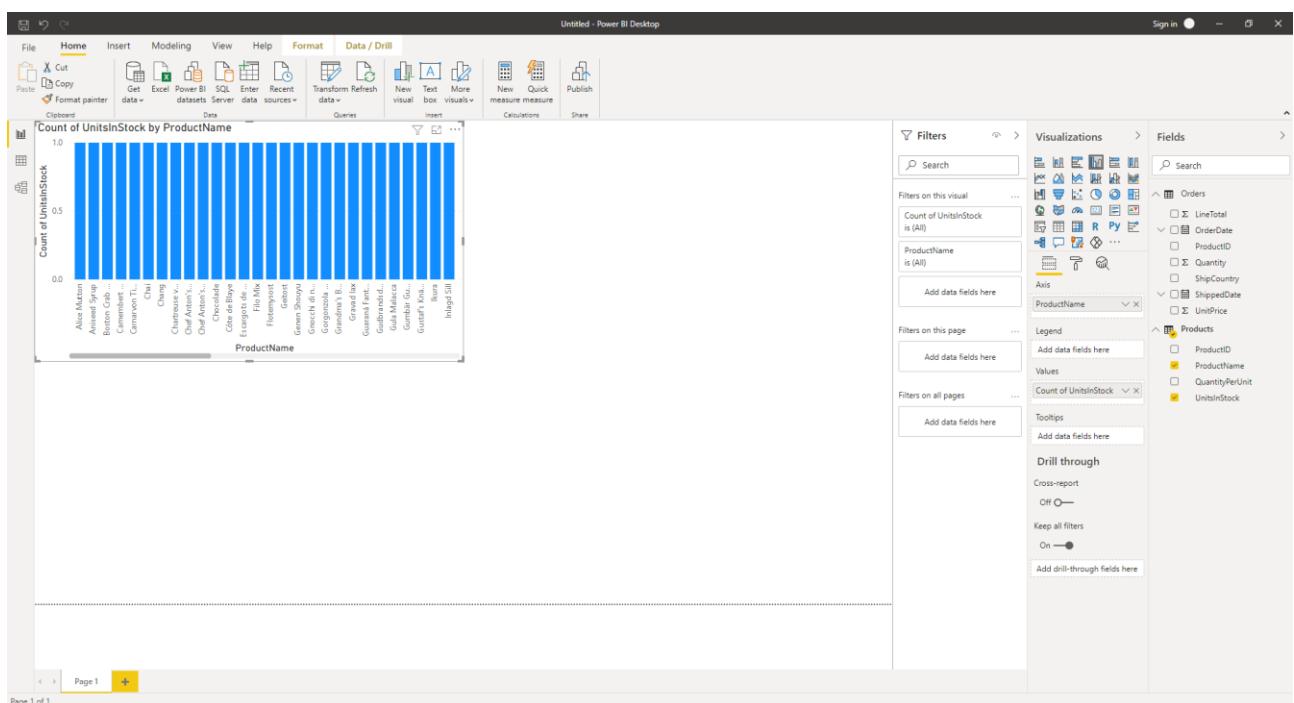
4.4 Build Visuals Using your Data

4.4.1 Clustered Bar Chart (Units in Stock by Product)

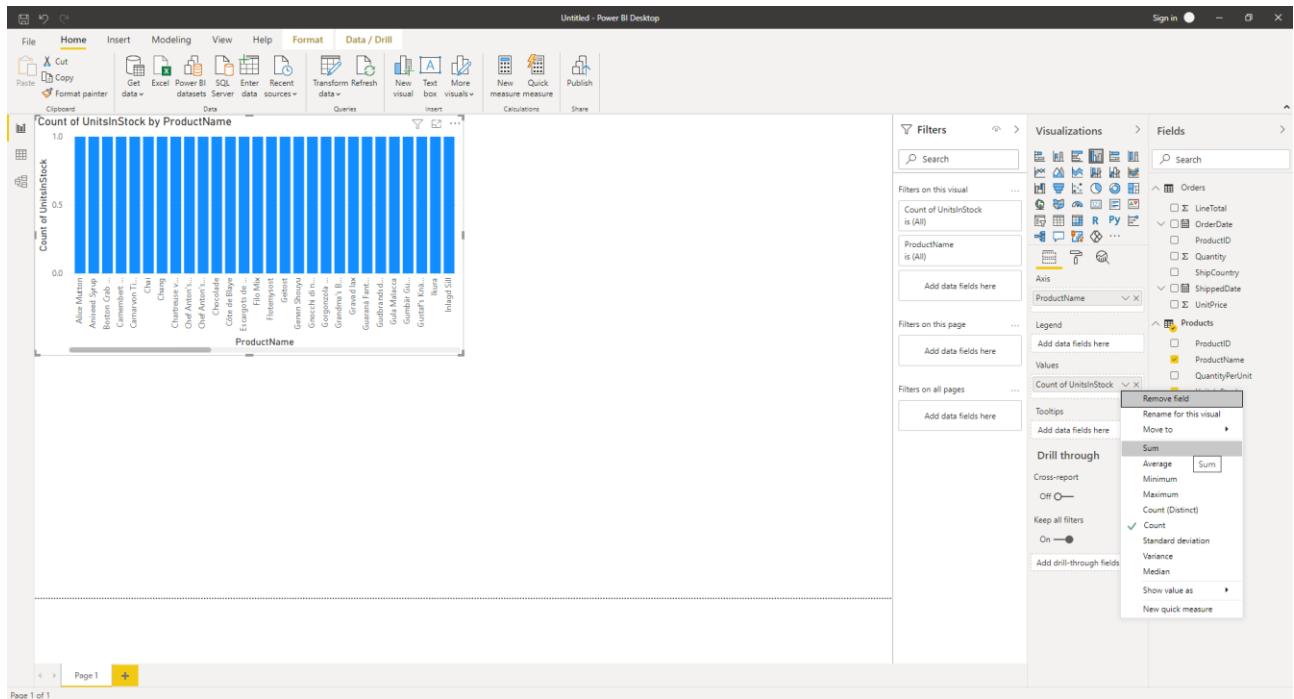
1. Add a **Clustered Bar Chart** visual, and then resize and reposition it.



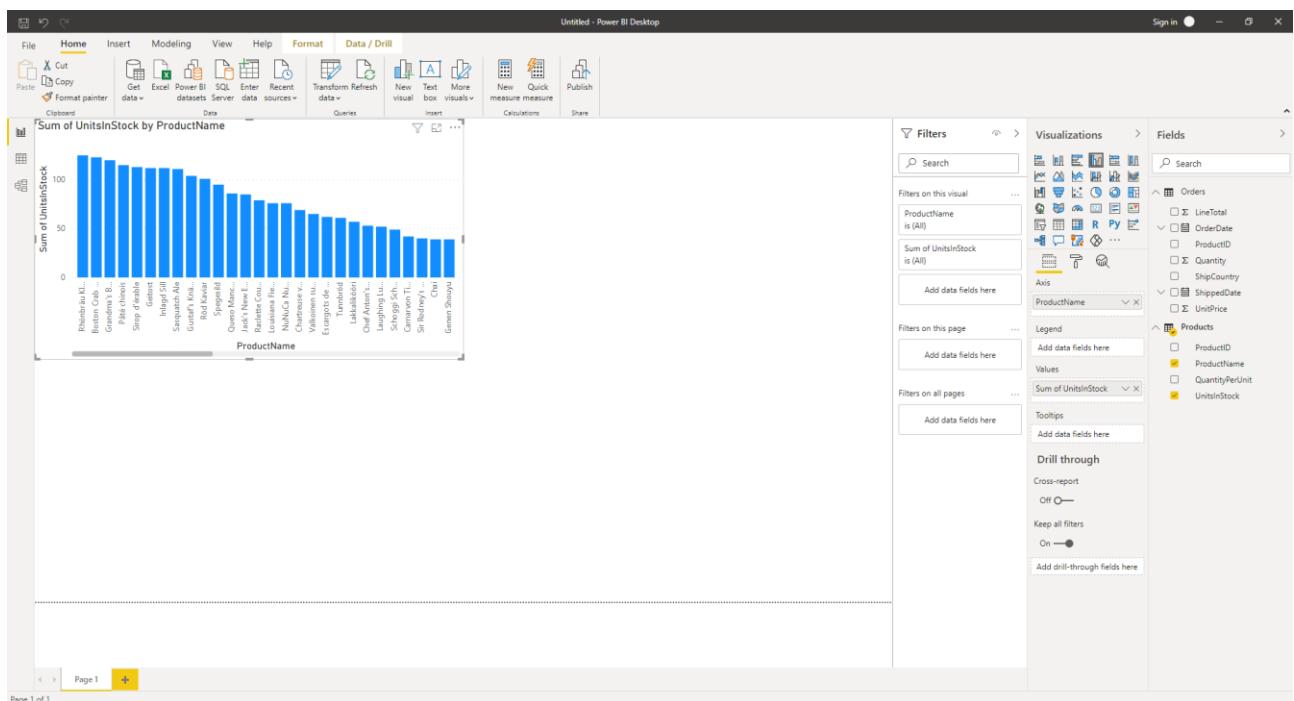
2. Add the following fields to the visual wells:
 - Assign **Product** → **ProductName** Month to **Axis**
 - Assign **Product** → **UnitsInStock** to **Value**



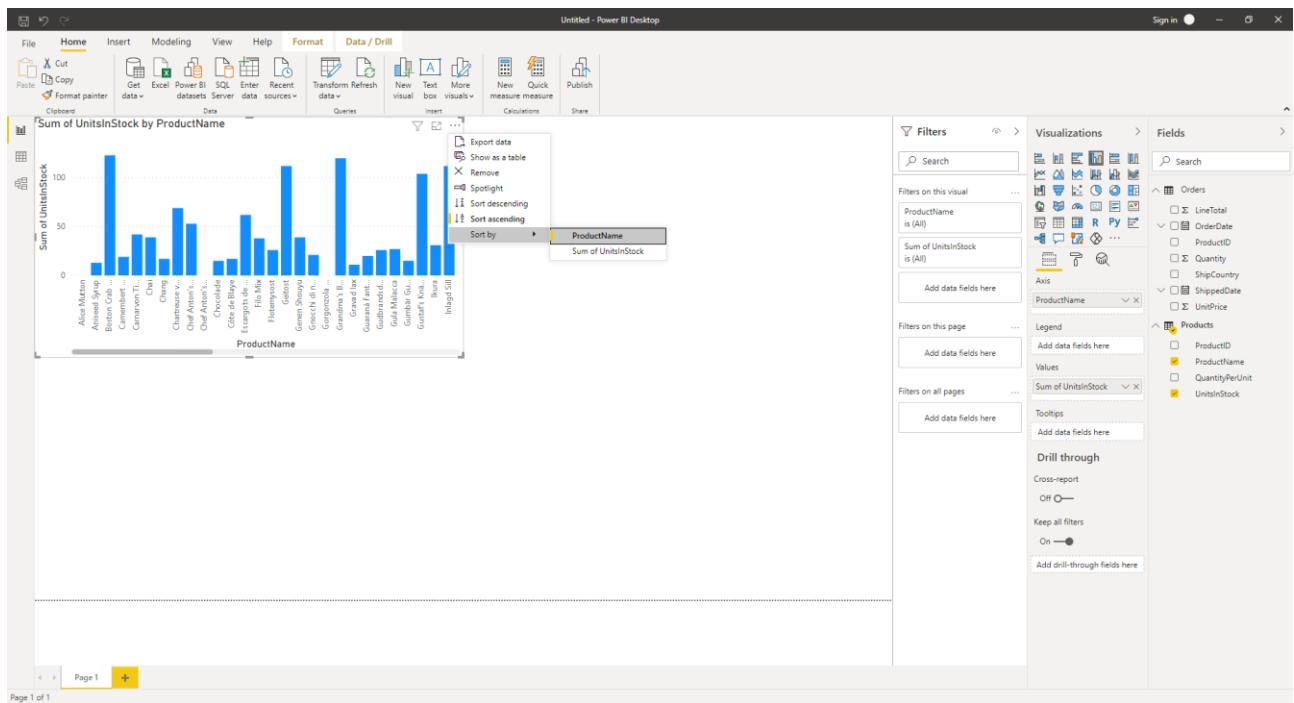
3. Change the *Count of UnitsInStock* in Values to Sum



4. Drag *UnitsInStock* from the Field pane onto a blank space on the canvas. A Table visualization is created.

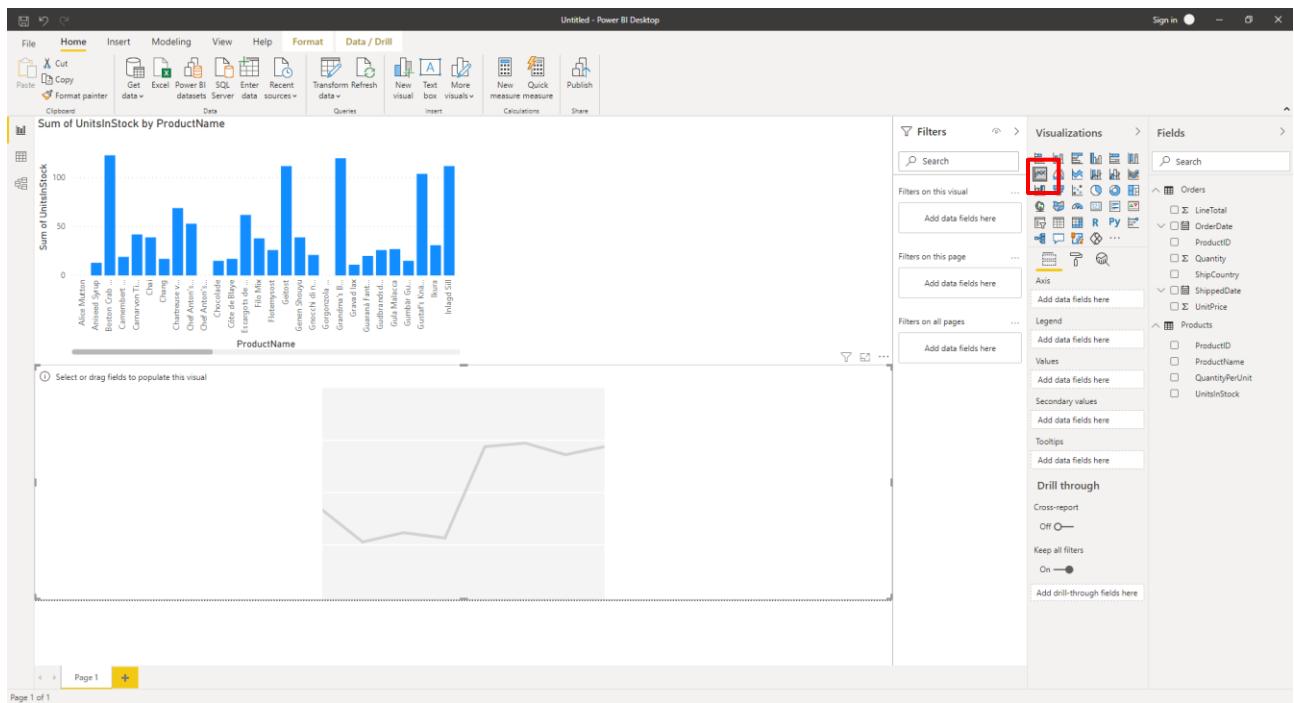


5. Select **Sort By → ProductName** in ascending order using the skittles in the top right corner of the visualization.



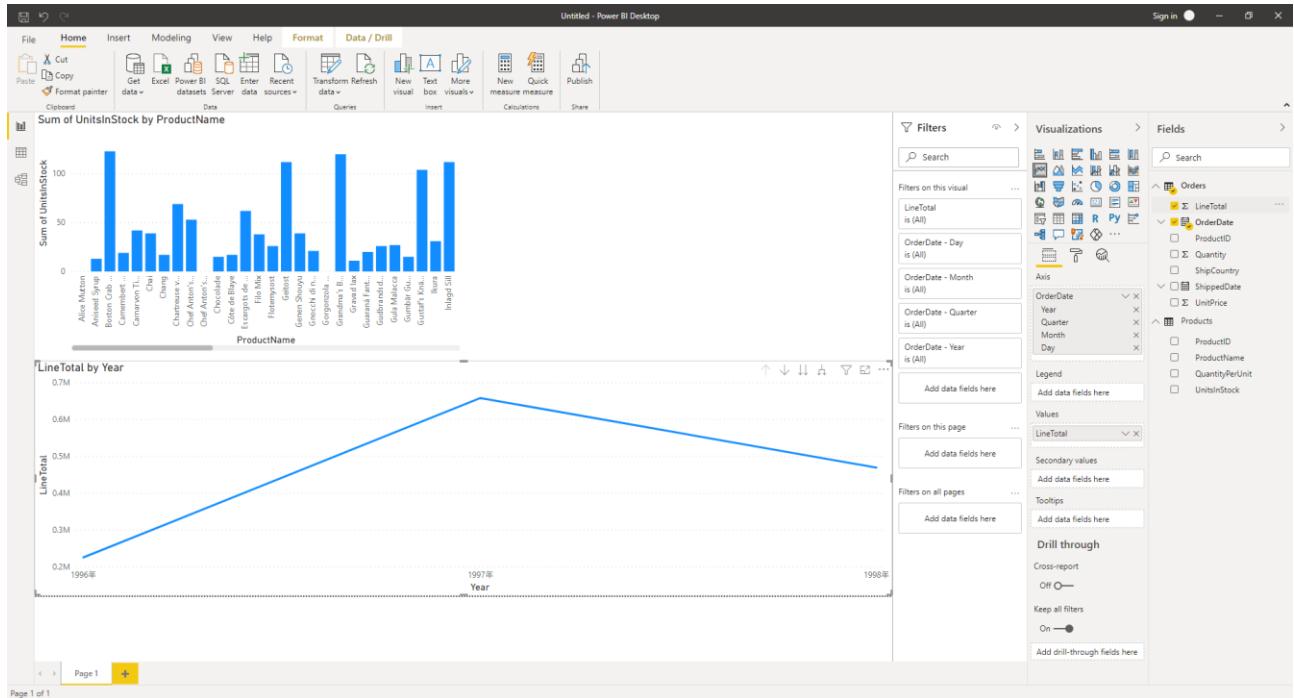
4.4.2 Line Chart (Total Sales by Year)

1. Add a **Line Chart** visual under the column chart, then resize and reposition it.

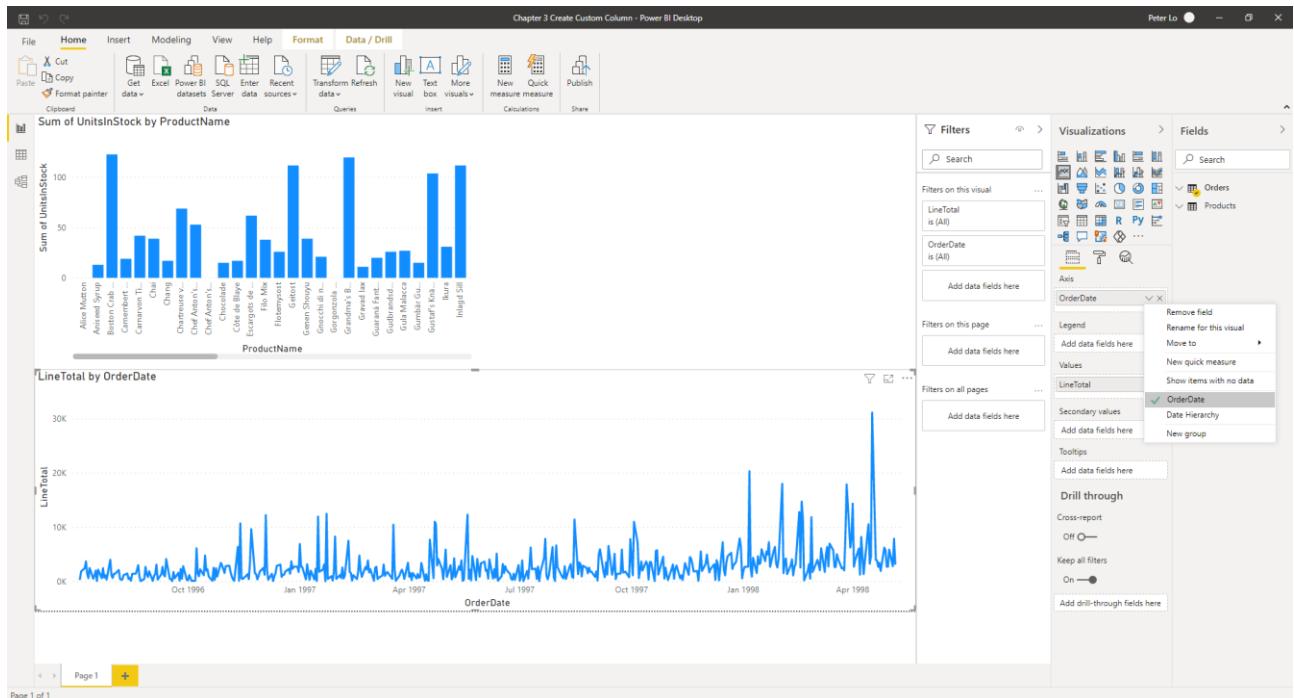


2. Add the following fields to the visual wells:

- Assign *Order* → *OrderDate* to **Axis**
- Assign *Order* → *LineTotal* to **Value**

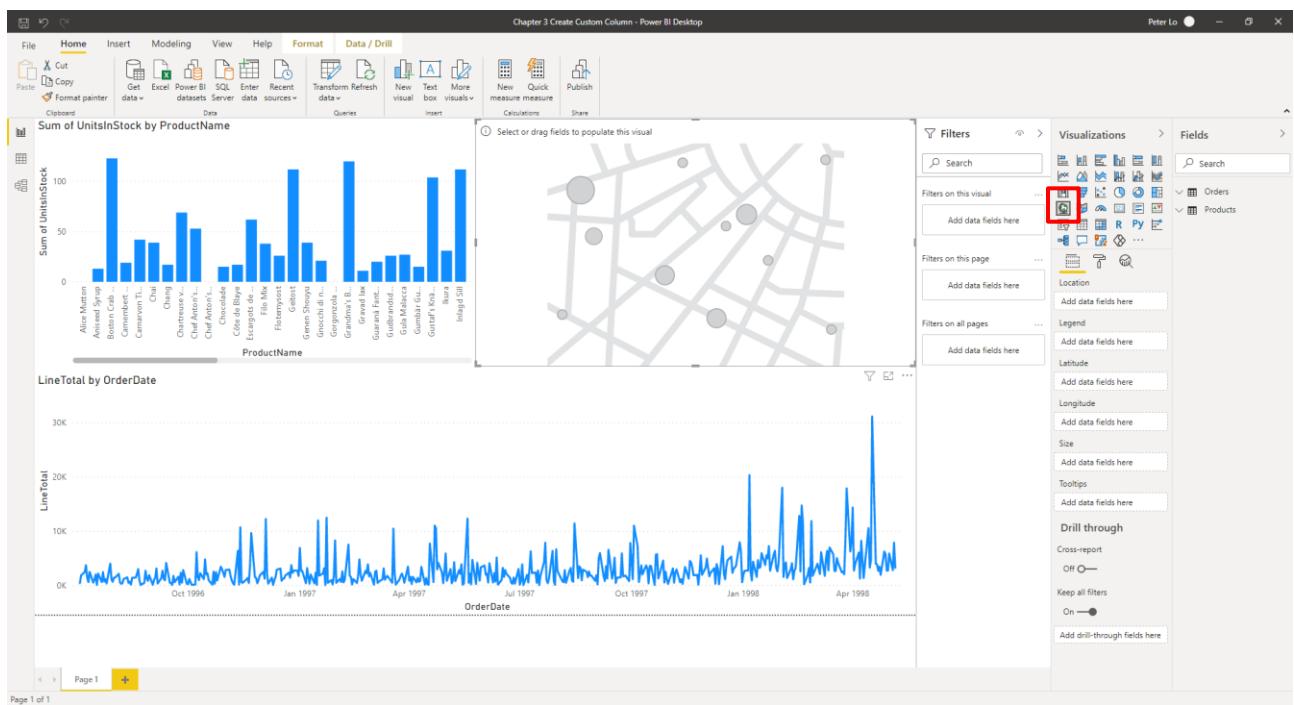


3. Select *OrderDate* in Axis and select **OrderDate**.



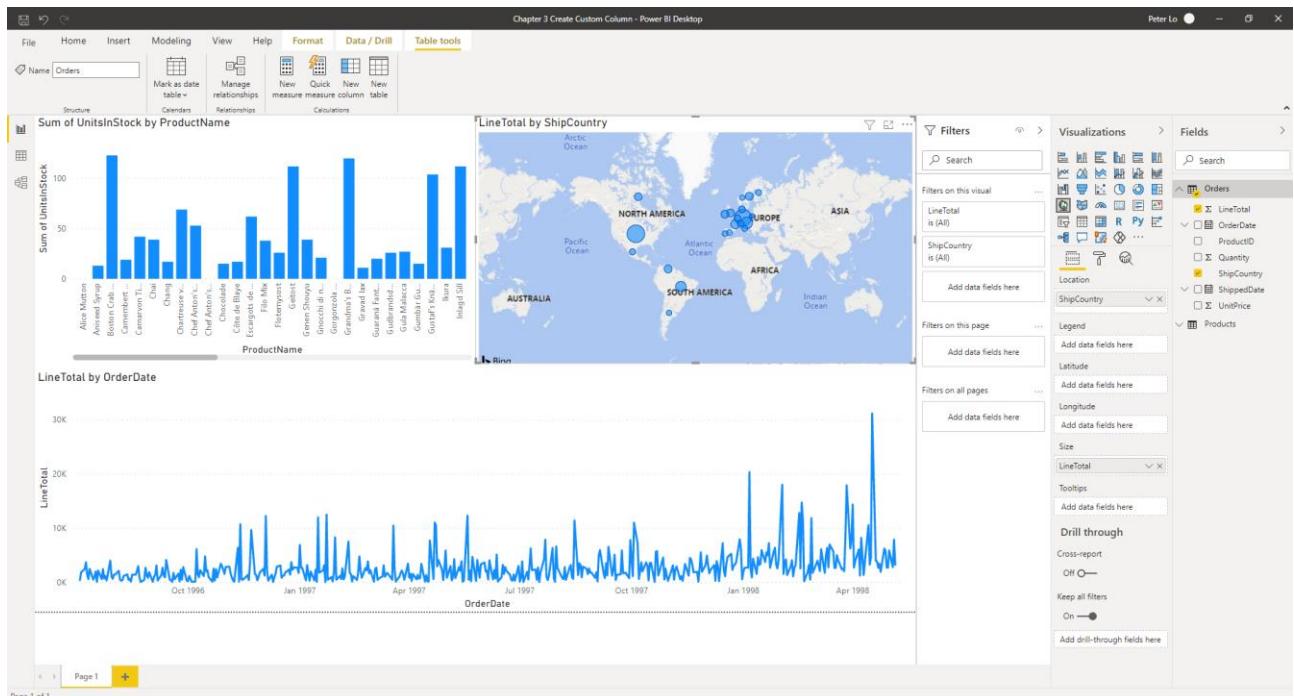
4.4.3 Map (Total Sales per Country)

- Add a Map visual beside the column chart, then resize and reposition it.



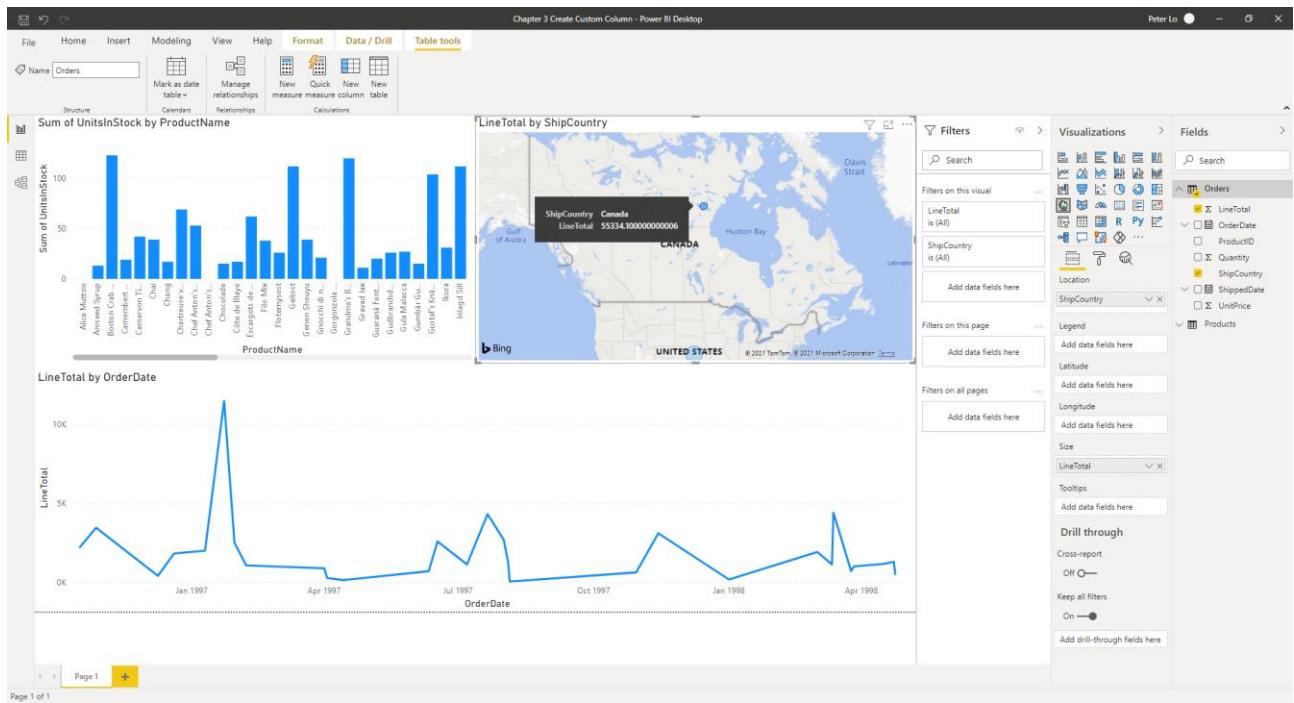
- Add the following fields to the visual wells:

- Assign *Order* → *ShipCountry* to **Location**
- Assign *Order* → *LineTotal* to **Size**



4.4.4 Interact with Report Visuals

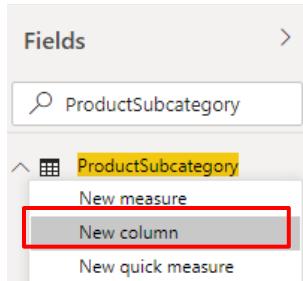
- Click on the light blue circle centered in Canada. Note how the other visuals are filtered to show Stock (*ShipCountry*) and Total Orders (*LineTotal*) just for Canada.



5. Create Calculated Columns

5.1 Create a Custom Column

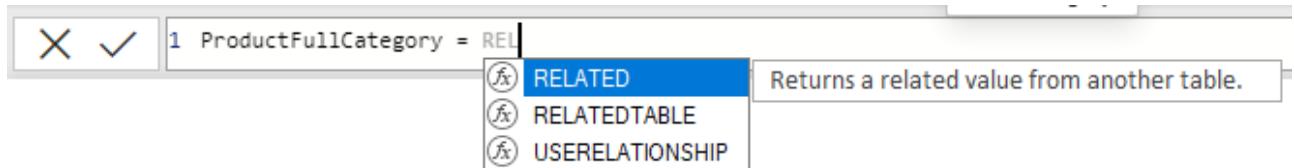
1. Select **File ➔ Open Report**, and open *Contoso Sales Sample for Power BI Desktop.pbix*
2. Click the down arrow on the *ProductSubcategory* table in the Fields list, and then click **New Column**. This will make sure our new column is added to the *ProductSubcategory* table.



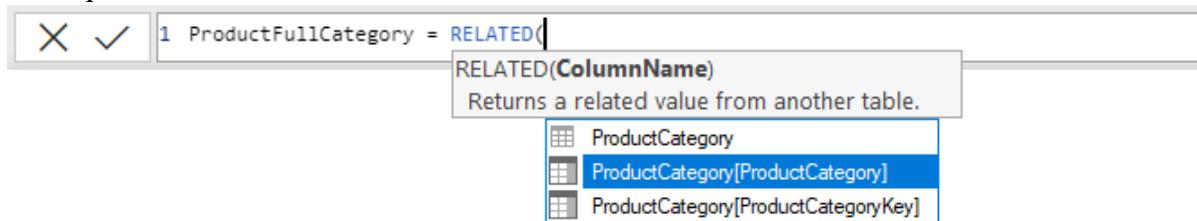
3. The formula bar appears along the top of the **Report** canvas. This is where we can rename our column and enter a DAX formula, type *ProductFullCategory*.



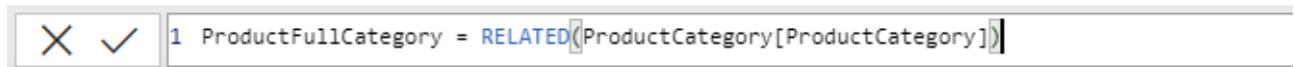
4. Select **RELATED** by scrolling down, and then pressing **[Enter]**.



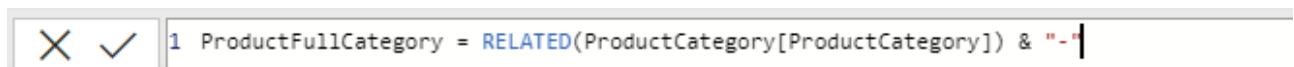
5. An opening parenthesis appears along with another suggestion list of all of the available columns we can pass to the **RELATED** function. A description and details on what parameters are expected is also shown.



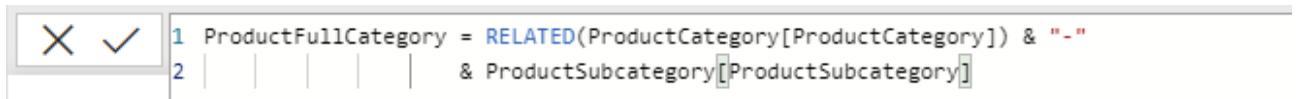
6. Select *ProductCategory[ProductCategory]*, and then type a closing parenthesis.



7. We want to add a dash symbol to separate each value, so after the closing parenthesis of the first expression, type a space, ampersand (&), quote, space, dash (-), another space, a closing quote, and then another ampersand.



8. Enter another opening bracket and then select the *[ProductSubcategory]* column to finish the formula. We didn't use another **RELATED** function in the second expression calling the *ProductSubcategory* column. This is because this column is already in the same table we're creating our new column in. We can enter *[ProductCategory]* with the table name (fully qualified) or without (non-qualified).



9. Complete the formula by pressing **[Enter]** or clicking on the checkmark in the formula bar. The formula is validated and added to the field list in the *ProductSubcategory* table.

The screenshot shows the Power BI Fields pane. Under the ProductSubcategory table, the *ProductFullCategory* column is listed with a red box around it, indicating it has been added.

5.2 Add New Column to Report

1. Start on a blank report page, Drag *ProductSubcategory* → *ProductFullCategory* onto the Report canvas, and then drag the *Sales* → *SalesAmount* field into the table.

The screenshot shows a Power BI report canvas with a table visual. The table contains the following data:

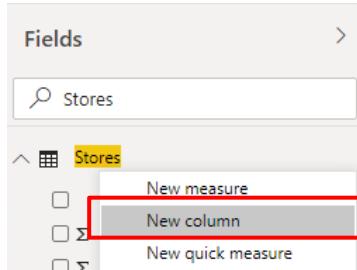
	SalesAmount
Audio-Bluetooth Headphones	\$41,907,488.9135
Audio-MP4&MP3	\$65,190,616.3964
Audio-Recording Pen	\$44,516,259.001
Cameras and camcorders-Camcorders	\$1,335,302,769.92
Cameras and camcorders-Cameras & Camcorders Accessories	\$51,643,775.8286
Cameras and camcorders-Digital Cameras	\$365,082,489.926
Cameras and camcorders-Digital SLR Cameras	\$809,994,738.386
Cell phones-Cell phones Accessories	\$120,017,198.2572
Cell phones-Home & Office Phones	\$48,333,160.9812
Cell phones-Smart phones & PDAs	\$423,389,457.48
Cell phones-Touch Screen Phones	\$300,493,447.58
Computers-Computers Accessories	\$111,023,802.1072
Computers-Desktops	\$508,196,937.084
Computers-Laptops	\$933,130,593.776
Computers-Monitors	\$268,114,052.77
Computers-Printers, Scanners & Fax	\$281,762,342.2
Computers-Projectors & Screens	\$1,107,199,413.48
Music, Movies and Audio Books-Movie DVD	\$165,804,705.9811
TV and Video-Car Video	\$306,818,844.52
TV and Video-Home Theater System	\$709,120,510.209
TV and Video-Televisions	\$307,373,914,4742
TV and Video-VCD & DVD	\$36,807,845.561
Total	\$8,341,224,364.8324

On the right side of the screen, the Power BI Fields pane is open, showing the filters applied to the table: *ProductFullCategory* is (All) and *SalesAmount* is (All). The SalesAmount filter is highlighted with a red box.

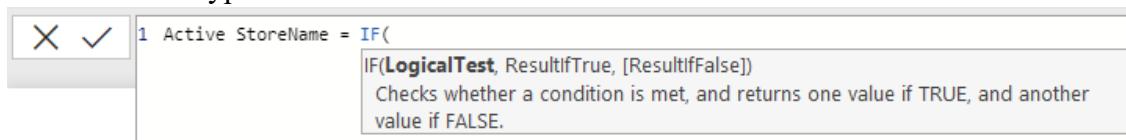
5.3 Create a Logical Column

Now we would like to create a new calculated column named *Active StoreName* in the *Stores* table. For this column, our DAX formula is going to check each stores status. If a stores status is On, our formula will return the stores name. If it's Off, it will have the name, "Inactive".

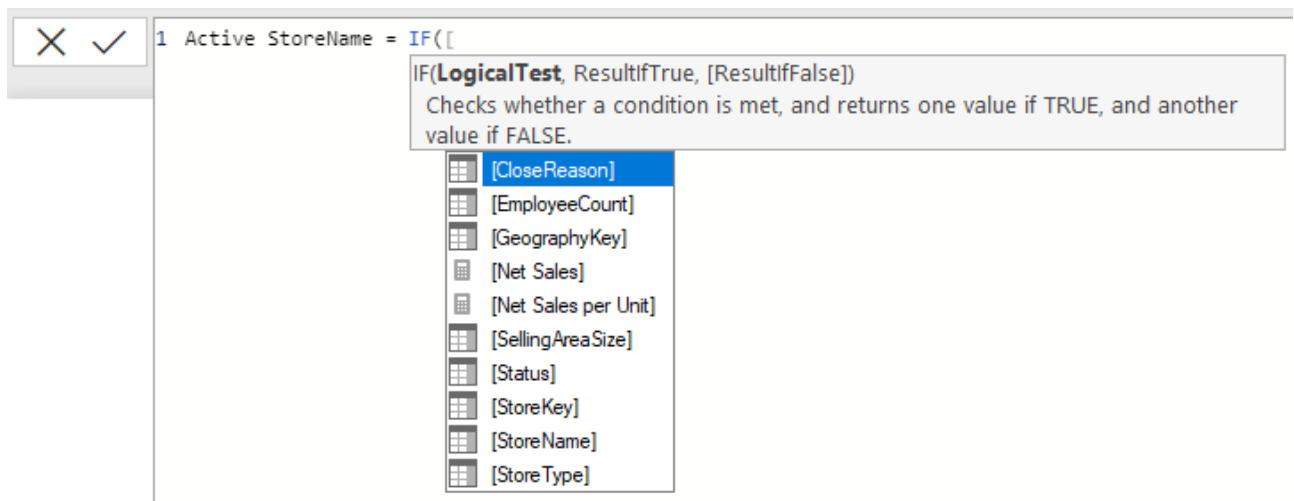
1. Click the down arrow on the *Stores* table in the Fields list, and then click **New Column**.



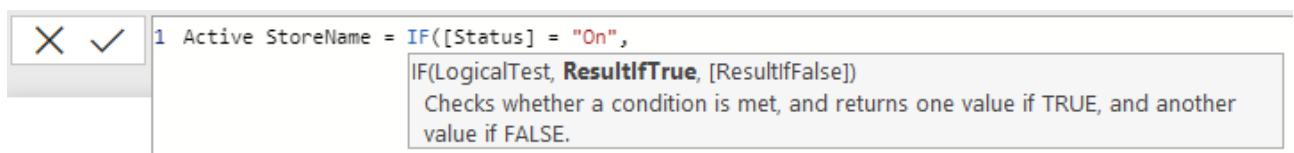
2. Select **IF** after type the column name *Active StoreName*.



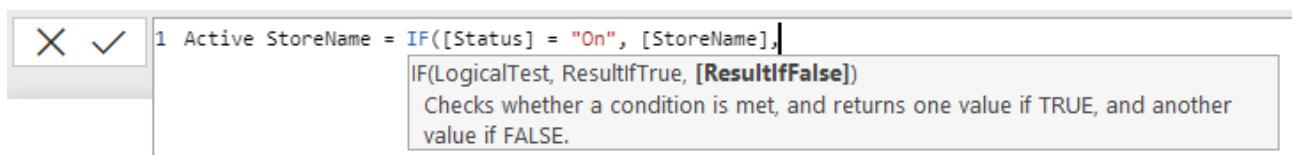
3. Type an opening bracket [, and select *[Status]*.



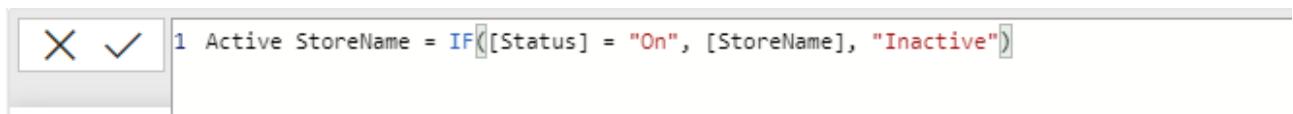
4. Right after *[Status]*, type = "On", then enter a comma (,) to enter the second argument. The tooltip suggests we need to add the value for when the result is true.



5. If the store is On, we want to show the store's name. Type an opening bracket [and select the *[StoreName]* column, and then type another comma so we can enter our third argument.



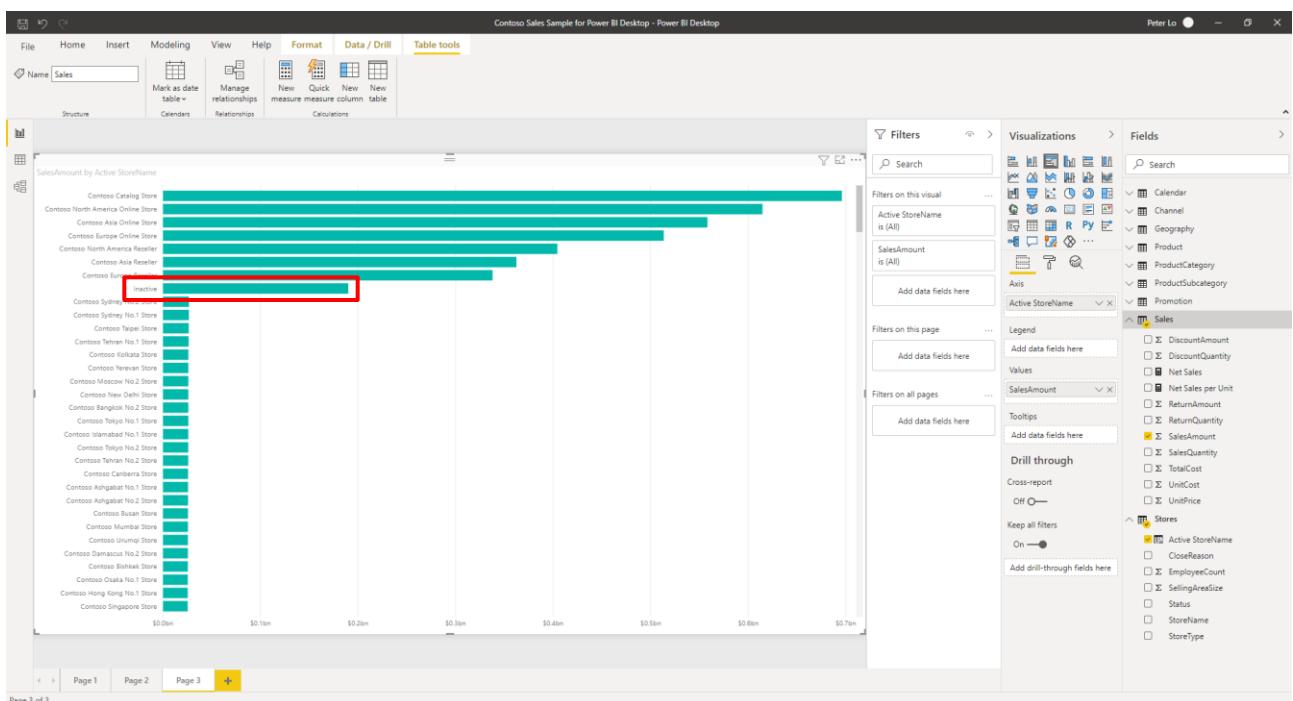
6. We need to add a value for when the result is false, in this case we want the value to be “**Inactive**”. Complete the formula by pressing **[Enter]** in the formula bar. The formula is validated and added to the field list in the *Stores* table.



7. Add a **Clustered Bar Chart** visual, and add the following fields to the visual wells:

- Assign *Store* → *Active StoreName* to **Axis**
- Assign *Sales* → *Sales Amount* to **Value**

In this chart, stores with a status of On are shown individually by name, but stores with a status of Off are grouped together and shown as Inactive.



6. Create Logical Column

6.1 Connect to Excel Workbook

1. Select **Get Data → Excel** in the **Home** ribbon tab,
2. In the **Open File** dialog box, select the **Skyscrapers.xlsx** file. Then in the Navigator pane, select the **Buildings** table and then select **[Load]**.

Building Name	City	Completed	Country
Burj Khalifa	Dubai	2010	AE
Shanghai Tower	Shanghai	2015	CN
Makkah Royal Clock Tower	Mecca	2012	SA
One World Trade Center	New York City	2014	US
Guangzhou CTF Finance Centre	Guangzhou	2016	CN
TAIPEI 101	Taipei	2004	TW
Shanghai World Financial Center	Shanghai	2008	CN
International Commerce Centre	Hong Kong	2010	CN
Petronas Twin Tower 1	Kuala Lumpur	1998	MY
Petronas Twin Tower 2	Kuala Lumpur	1998	MY
Zifeng Tower	Nanjing	2010	CN
Willis Tower	Chicago	1974	US
KK100	Shenzhen	2011	CN
Guangzhou International Finance Center	Guangzhou	2010	CN
432 Park Avenue	New York City	2015	US
Trump International Hotel & Tower	Chicago	2009	US
Jin Mao Tower	Shanghai	1999	CN
Princess Tower	Dubai	2012	AE
Al Hamra Tower	Kuwait City	2011	KW
Two International Finance Centre	Hong Kong	2003	CN
23 Marina	Dubai	2012	AE
CITIC Plaza	Guangzhou	1996	CN
Shun Hing Square	Shenzhen	1996	CN

6.2 Create Table Visual

1. Create a table for building name, city and use. There are 3 columns giving the uses to which each building is put. Add a **Table** visual, then add the following fields: *Building Name*, *City*, *Use1*, *Use2* and *Use3* to **Values**.

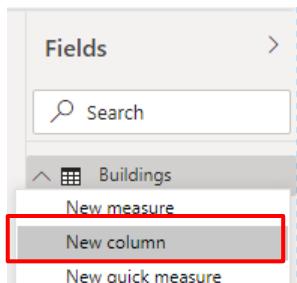
Building Name	City	Use1	Use3
23 Marina	Dubai	residential	
432 Park Avenue	New York City	residential	
ADNOC Headquarters	Abu Dhabi	office	
AI Hamra Tower	Kuwait City	office	
Al Yaquob Tower	Dubai	hotel	office
Almas Tower	Dubai	office	
Aon Center	Chicago	office	
Bayoke Tower II	Bangkok	hotel	
Bank of America Plaza	Atlanta	office	
Bank of America Tower	New York City	office	
Bank of China Tower	Hong Kong	office	
Burj Al Arab	Dubai	hotel	
Burj Khalifa	Dubai	office	residential hotel
Burj Mohammed Bin Rashid	Abu Dhabi	residential	
Burj Rafal	Riyadh	residential	hotel
Cayan Tower	Dubai	residential	
Central Plaza	Hong Kong	office	
China World Tower	Beijing	hotel	office
Chongqing World Financial Center	Chongqing	hotel	office
Chrysler Building	New York City	office	
CITIC Plaza	Guangzhou	office	
Deji Plaza	Nanjing	hotel	office
Diwang International Fortune Center	Luzhou	office	hotel
East Pacific Center	Shenzhen	residential	
Elite Residence	Dubai	residential	
Emirates Tower One	Dubai	office	
Emirates Tower Two	Dubai	hotel	
Empire State Building	New York City	office	
Eithad Towers T2	Abu Dhabi	residential	
Eton Place Dalian Tower 1	Dalian	hotel	office
Fortune Center	Guangzhou	office	
Forum 66 Tower 1	Shenyang	hotel	office
Global City Square	Guangzhou	office	
Greenland Puli Center	Jinan	office	

6.3 Using IF in Column

We would like to use the IF function to create a new column to show for each office its use type:

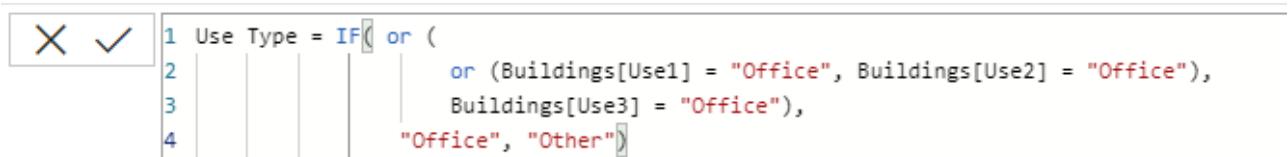
Rule	Use type
If any of the 3 Use columns is office	Office
Otherwise	Other

1. Click the down arrow on the *Building* table in the field list, and then click **New Column**.



2. Name this new column as Use Type. In order to perform this checking, IF and OR function is used in the function.

Use Type = IF(or (or (Buildings[Use1] = "Office", Buildings[Use2] = "Office"), Buildings[Use3] = "Office"), "Office", "Other")



3. Create a table by using *CountryCode* and *Use Type*.

Building Name	City	Use1	Use2	Use3
23 Marina	Dubai	residential		
432 Park Avenue	New York City	residential		
ADNOC Headquarters	Abu Dhabi	office		
Al Hamra Tower	Kuwait City	office		
AI Yas Tower	Dubai	hotel		
Almas Tower	Dubai	office		
Aon Center	Chicago	office		
Baytak Tower II	Bangkok	hotel		
Bank of America Plaza	Atlanta	office		
Bank of America Tower	New York City	office		
Bank of China Tower	Hong Kong	office		
Buri Al Arab	Dubai	hotel		
Buri Khalifa	Dubai	office	residential	hotel
Buri Mohammed Bin Rashid	Abu Dhabi	residential		
Buri Rafal	Riyadh	residential	hotel	
Cayan Tower	Dubai	residential		
Central Plaza	Hong Kong	office		
China World Tower	Beijing	hotel	office	
Chongqing World Financial Center	Chongqing	hotel	office	
Chrysler Building	New York City	office		
CITIC Plaza	Guangzhou	office		
Deji Plaza	Nanjing	hotel	office	
Diwang International Fortune Center	Liaozhou	office	hotel	
East Pacific Center Tower A	Shenzhen	residential		
Elite Residence	Dubai	residential		
Emirates Tower One	Dubai	office		
Emirates Tower Two	Dubai	hotel		
Empire State Building	New York City	office		
Eitha Towers T2	Abu Dhabi	residential		
Eton Place Dalian Tower 1	Dalian	hotel	office	
Fortune Center	Guangzhou	office		
Forum 66 Tower 1	Shenyang	hotel	office	
Global City Square	Guangzhou	office		
Greenland Puli Center	Jinan	office		

4. Change the calculate method for *Use Type* to **Count**

The screenshot shows the Power BI Desktop interface with a table visual titled "Logical Measure - Power BI Desktop". The table has columns: Building Name, City, Use1, Use2, and Use3. The Fields pane on the right is open, specifically the "Fields" section under the "Buildings" category. A context menu is open over the "Count of Use Type" field, with the "Count" option selected. Other options like "Remove field", "Move", "Conditional formatting", and "Show value as" are also visible.

Building Name	City	Use1	Use2	Use3
23 Marina	Dubai	residential		
432 Park Avenue	New York City	residential		
ADNOC Headquarters	Abu Dhabi	office		
Al Hamra Tower	Kuwait City	office		
Al Yaqoub Tower	Dubai	hotel	office	
Almas Tower	Dubai	office		
Aon Center	Chicago	office		
Baikoke Tower II	Bangkok	hotel		
Bank of America Plaza	Atlanta	office		
Bank of America Tower	New York City	office		
Bank of China Tower	Hong Kong	office		
Burj Al Arab	Dubai	hotel		
Burj Khalifa	Dubai	office	residential	hotel
Burj Mohammed Bin Rashid	Abu Dhabi	residential		
Burj Rafal	Riyadh	residential	hotel	
Cayan Tower	Dubai	residential		
Central Plaza	Hong Kong	office		
China World Tower	Beijing	hotel	office	
Chongqing World Financial Center	Chongqing	hotel	office	
Chrysler Building	New York City	office		
CITIC Plaza	Guangzhou	office		
Deji Plaza	Nanjing	hotel	office	
Diwang International Fortune Center	Liu Zhou	office	hotel	
East Pacific Center Tower A	Shenzhen	residential		
Elite Residence	Dubai	residential		
Emirates Tower One	Dubai	office		
Emirates Tower Two	Dubai	hotel		
Empire State Building	New York City	office		
Eihad Towers T2	Abu Dhabi	residential		
Eton Place Dalian Tower 1	Dalian	hotel	office	
Fortune Center	Guangzhou	office		
Forum 66 Tower 1	Shenyang	hotel	office	
Global City Square	Guangzhou	office		
Greenland Puli Center	Jinan	office		

5. Drag the *Use Type* to values again, then change the calculate method for *Use Type* to **Count** and show value as to **Percentage of Grant Total**.

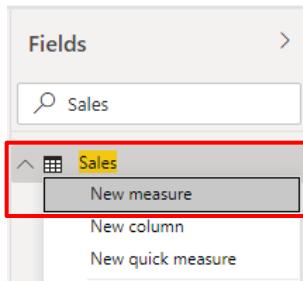
The screenshot shows the Power BI Desktop interface with a table visual titled "Logical Measure - Power BI Desktop". The table has columns: Building Name, City, Use1, Use2, and Use3. The Fields pane on the right is open, specifically the "Fields" section under the "Buildings" category. A context menu is open over the "Count of Use Type" field, with the "Percent of grand total" option selected. Other options like "Remove field", "Move", "Conditional formatting", and "Show value as" are also visible.

Building Name	City	Use1	Use2	Use3
23 Marina	Dubai	residential		
432 Park Avenue	New York City	residential		
ADNOC Headquarters	Abu Dhabi	office		
Al Hamra Tower	Kuwait City	office		
Al Yaqoub Tower	Dubai	hotel	office	
Almas Tower	Dubai	office		
Aon Center	Chicago	office		
Baikoke Tower II	Bangkok	hotel		
Bank of America Plaza	Atlanta	office		
Bank of America Tower	New York City	office		
Bank of China Tower	Hong Kong	office		
Burj Al Arab	Dubai	hotel		
Burj Khalifa	Dubai	office	residential	hotel
Burj Mohammed Bin Rashid	Abu Dhabi	residential		
Burj Rafal	Riyadh	residential	hotel	
Cayan Tower	Dubai	residential		
Central Plaza	Hong Kong	office		
China World Tower	Beijing	hotel	office	
Chongqing World Financial Center	Chongqing	hotel	office	
Chrysler Building	New York City	office		
CITIC Plaza	Guangzhou	office		
Deji Plaza	Nanjing	hotel	office	
Diwang International Fortune Center	Liu Zhou	office	hotel	
East Pacific Center Tower A	Shenzhen	residential		
Elite Residence	Dubai	residential		
Emirates Tower One	Dubai	office		
Emirates Tower Two	Dubai	hotel		
Empire State Building	New York City	office		
Eihad Towers T2	Abu Dhabi	residential		
Eton Place Dalian Tower 1	Dalian	hotel	office	
Fortune Center	Guangzhou	office		
Forum 66 Tower 1	Shenyang	hotel	office	
Global City Square	Guangzhou	office		
Greenland Puli Center	Jinan	office		

7. Create Custom Measures

7.1 Creating Custom Measure – Net Sales

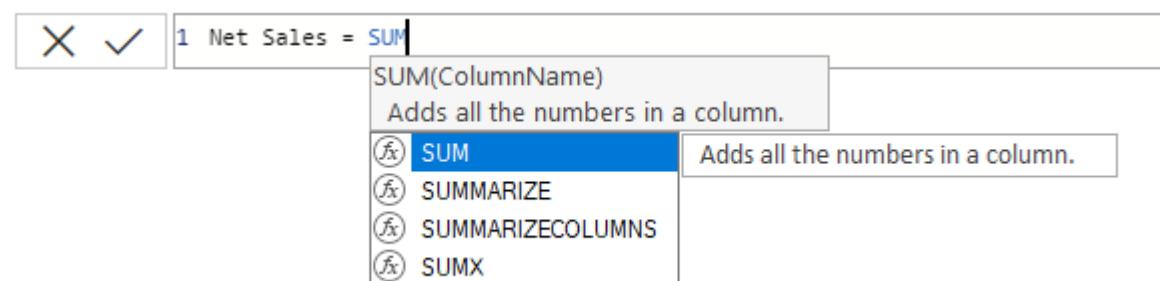
1. Select **File ➔ Open Report**, and open *Contoso Sales Sample for Power BI Desktop.pbix*
2. Click the down arrow on the *Sales* table in the field list, and then click **New Measure**. This will make sure our new measure is saved in the *Sales* table, where it will be easier to find.



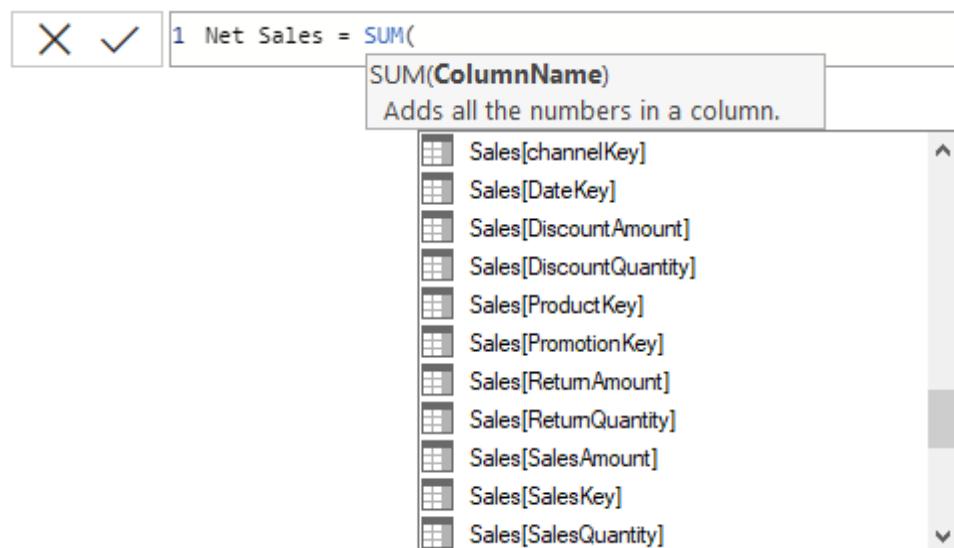
3. The formula bar appears along the top of the **Report Canvas**. This is where we can rename our measure and enter a DAX formula. Highlight *Measure* in the formula bar, and then type *Net Sales*.



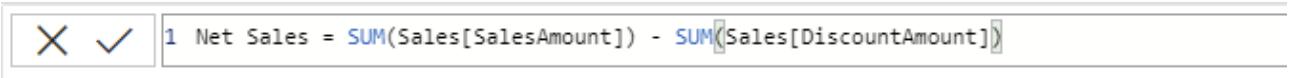
4. Select **SUM** by scrolling down, and then press **[Enter]**. An opening parenthesis appears along with another suggestion list of all of the available columns we can pass to the SUM function.



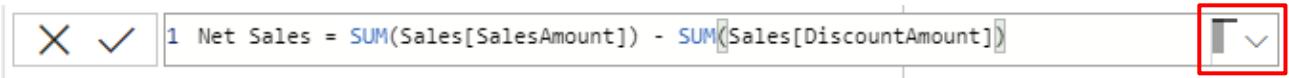
5. Select *Sales[SalesAmount]*, and then type a closing parenthesis



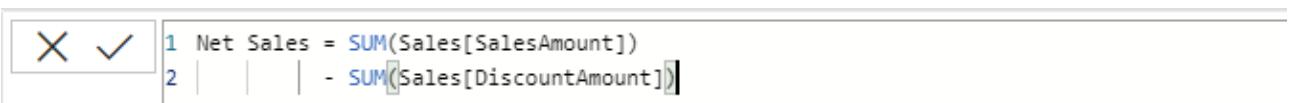
6. After the closing parenthesis for our first expression, type a space, and then a minus operator (-), followed by another space. Then enter another SUM function with the *Sales[DiscountAmount]* column as its argument.



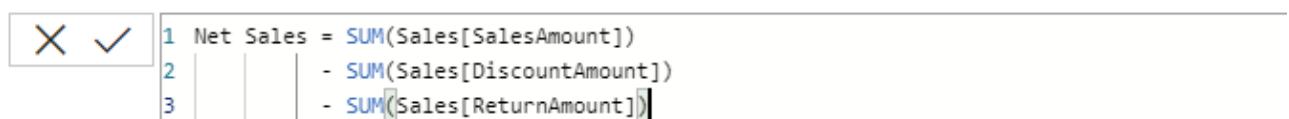
7. Click the down chevron on the right side of the formula bar.



8. We can enter new parts to our formula on a new line by pressing [Alt] + [Enter]. We can also move things over by using [Tab].



9. Add another minus operator followed by another SUM function and the *Sales[ReturnAmount]* column as its argument.

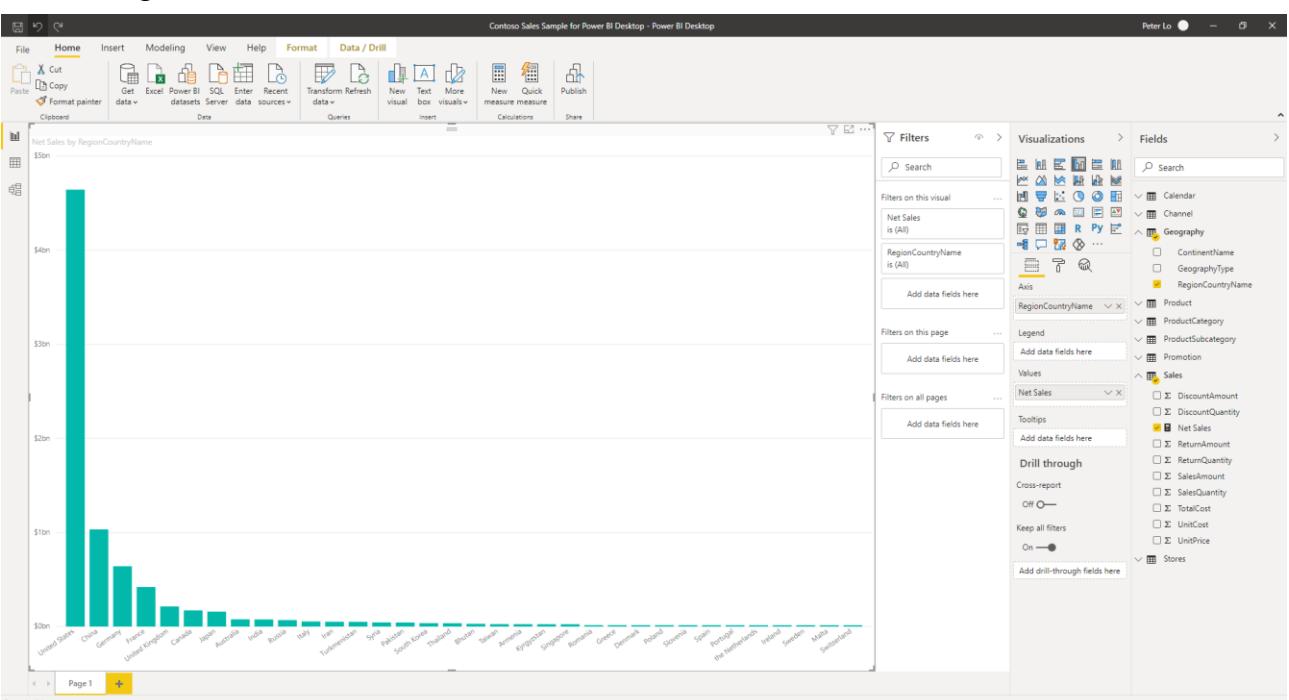


10. Press [Enter] in the formula bar to complete, the formula is validated and added to the field list in the *Sales* table.

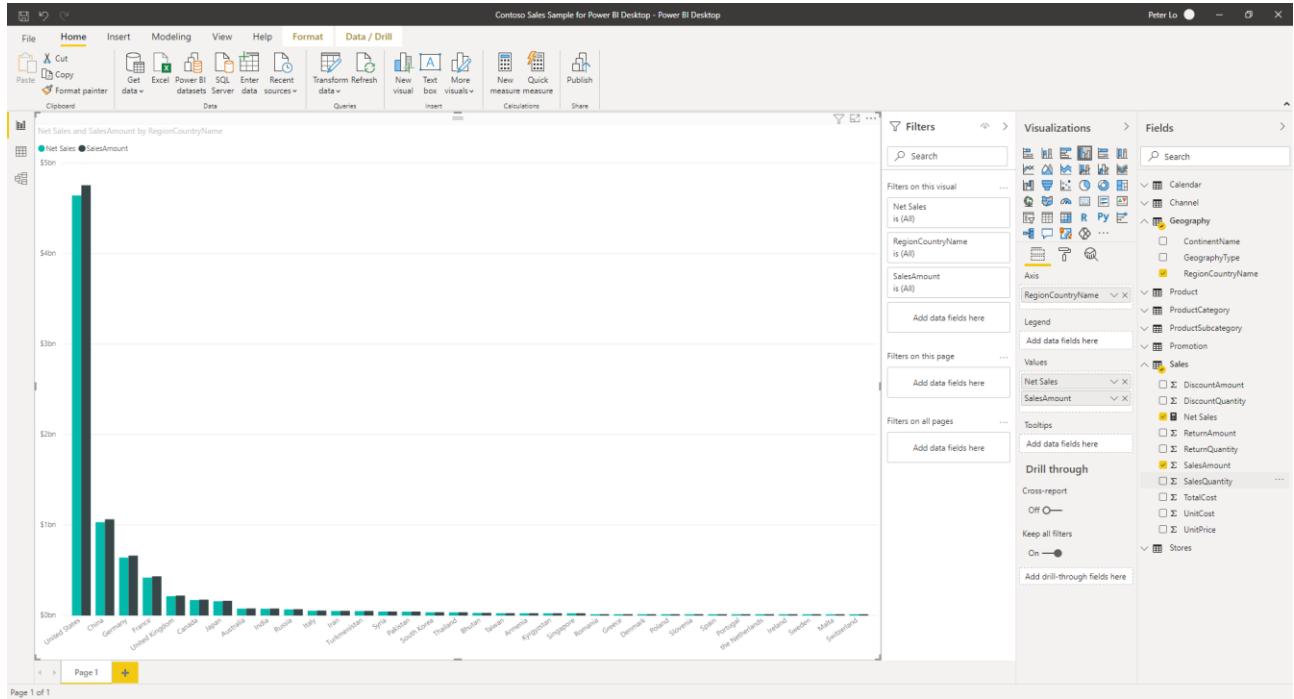
7.2 Add New Measure to Report

1. Add a **Clustered Bar Chart** visual and add the following fields to the visual wells:

- Assign *Geography* → *RegionCountryName* to **Axis**
- Assign *Sales* → *Net Sales* to **Value**



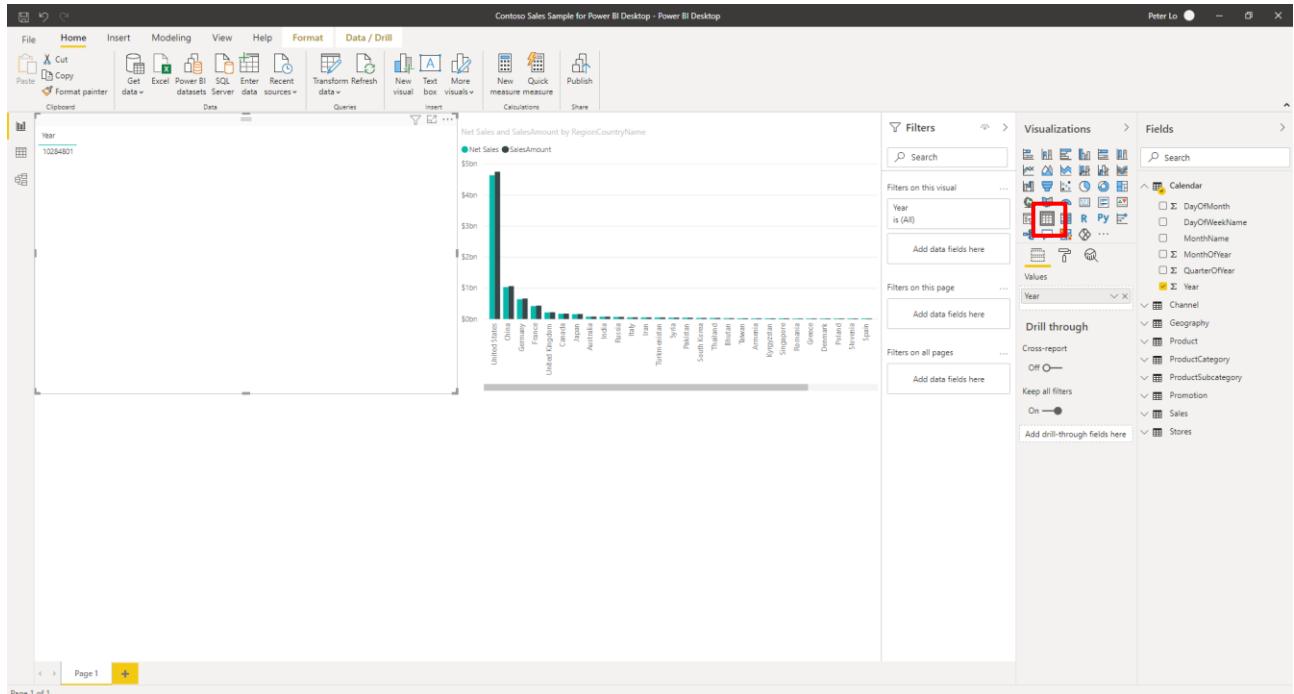
2. Drag the *Sales* → *SalesAmount* field into the chart, to see the difference between net sales and sales amount. We now really have two measures in our chart. *SalesAmount*, which was summed up automatically, and the *Net Sales* measure we created. In each case, the results were calculated in context of another field we have in the chart, the countries in *RegionCountryName*.



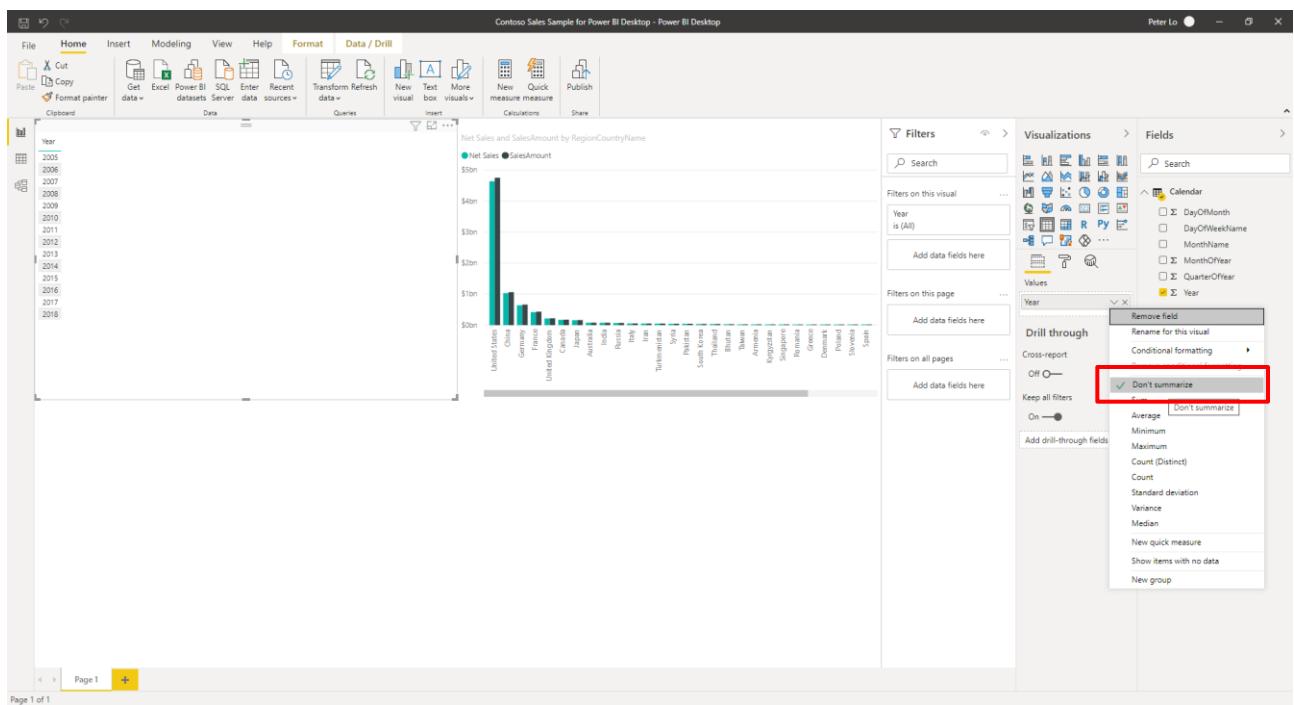
3. Add a **Table** visual and add the following fields to the visual wells.

- Assign *Calendar* → *Table* to Value

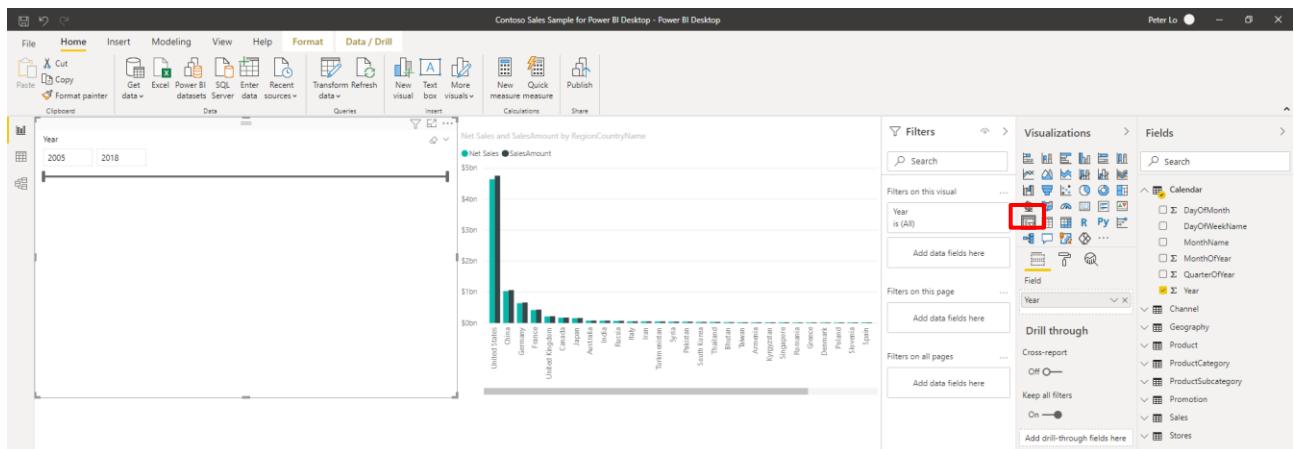
Because *Year* is a numeric field, Power BI Desktop summed up its values.



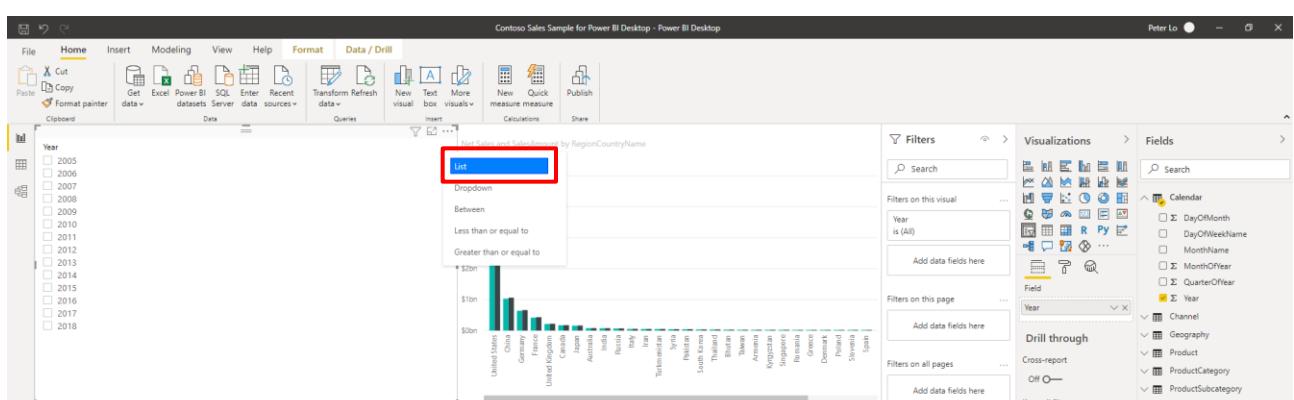
4. But that doesn't do us much good as a Slicer. Click the down arrow next to *Year* in **Values**, and then click **Don't Summarize**.



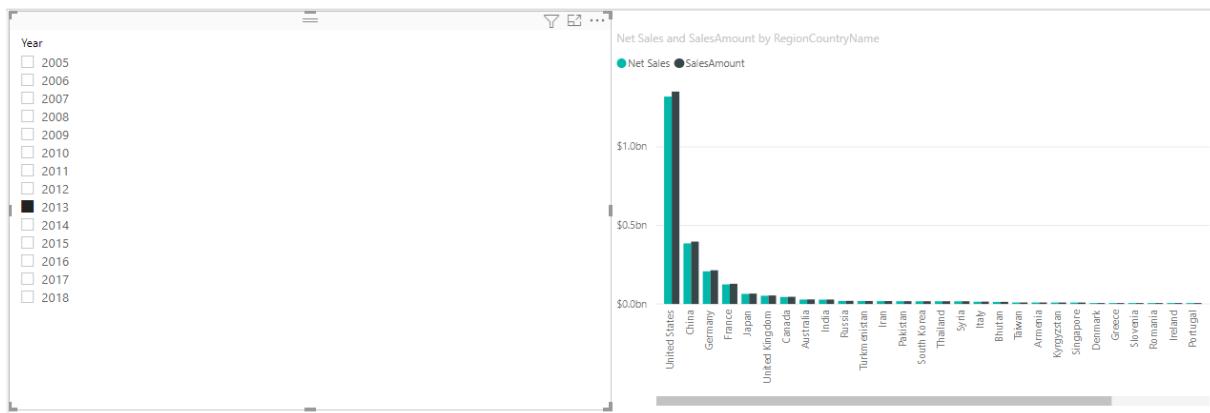
5. Now we can change the *Year* field in the table visualization into a **Slicer**. In **Visualizations**, click the **Slicer** visualization.



6. Change the *Year* Slicer to a **List**. We can select any individual or group of years and our report's visualizations will all be sliced accordingly.

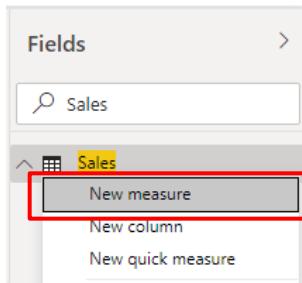


7. Go ahead and click on 2013. You'll see the chart change. Our *Net Sales* and *SalesAmount* measures are re-calculated, showing new results just for 2013.

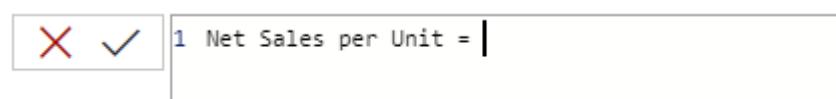


7.3 Create Another Measure – Net Sales per Unit

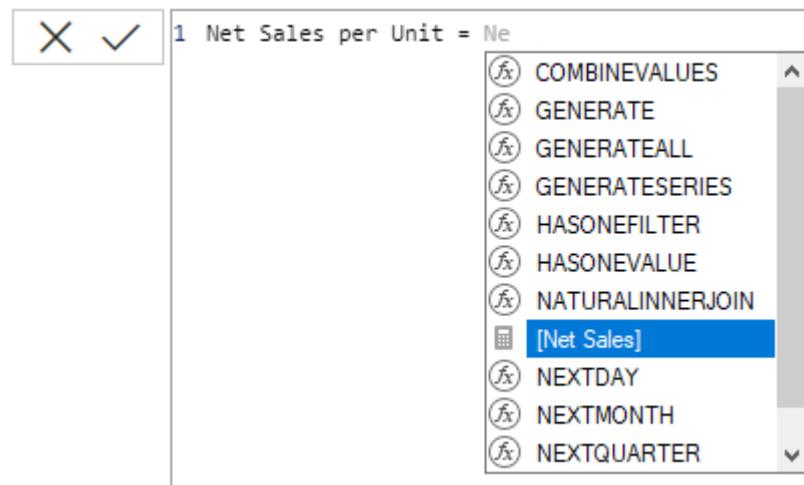
- In order to create another new measure, click the down arrow on the *Sales* table in the field list, and then click **New Measure**.



- Highlight Measure in the formula bar, and then rename as *Net Sales per Unit*.



- Begin typing *Net Sales*. The suggestion list will show what we can add. Select [*Net Sales*].

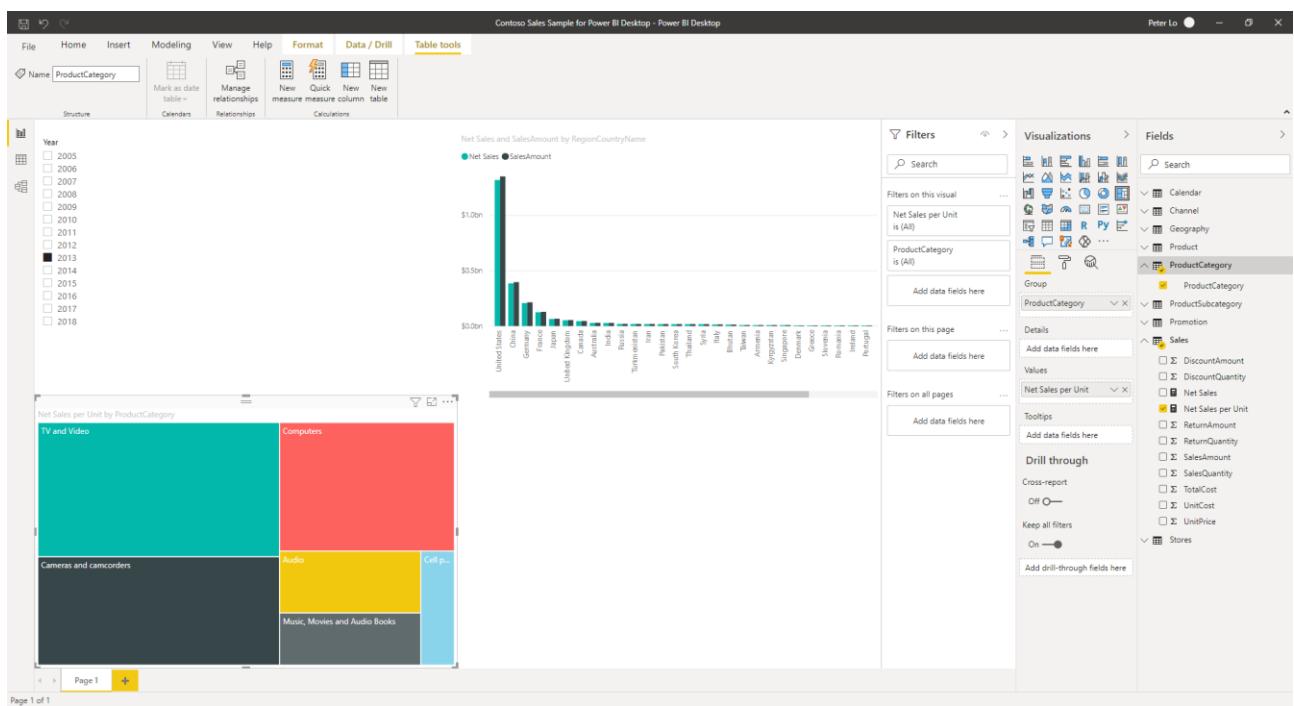


4. Right after [Net Sales], enter a space, then a divide operator (/), then enter a SUM function, then type Quantity. The suggestion list shows all of the columns with Quantity in the name. Select Sales[SalesQuantity]. The formula should now look like this:

1 Net Sales per Unit = [Net Sales] / Sum(Sales[SalesQuantity])

5. Add a **Treemap** visual and add the following fields to the visual wells:

- Assign *Sales* → *Net Sales per Unit* to **Values**
- Assign *ProductCategory* → *ProductCategory* to **Group**



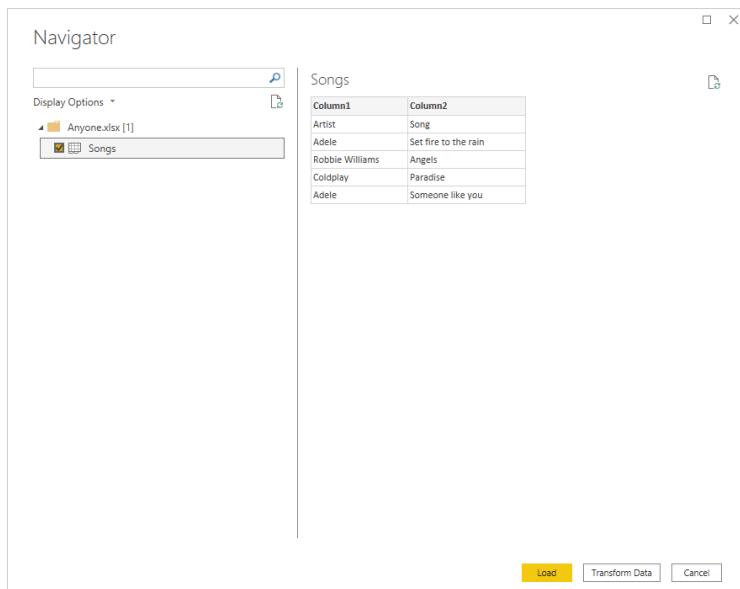
8. Parameter

A parameter serves as a way to easily store and manage a value that can be reused. Parameters give you the flexibility to dynamically change the output of your queries depending on their value, and can be used for:

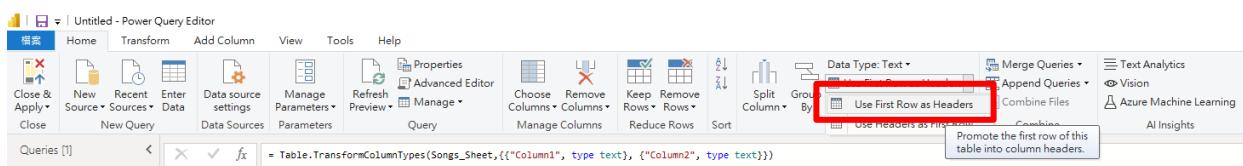
- Changing the argument values for particular transforms and data source functions
- Inputs in custom functions

You can easily manage your parameters inside the Manage Parameters window. You can get to the Manage Parameters window by selecting the Manage Parameters option inside Manage Parameters in the Home tab.

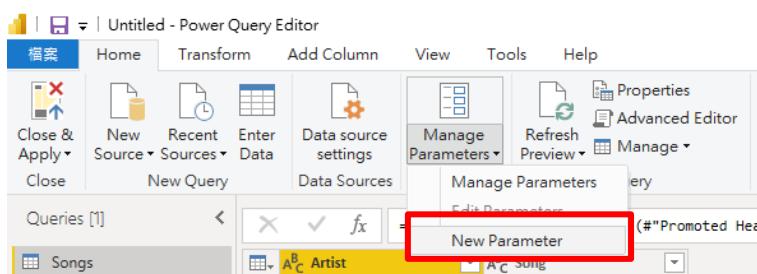
1. Select **Get Data → Excel** in the **Home** ribbon tab. In the **Open File** dialog box, select the *Anyone.xlsx* file. Then in the **Navigator** pane, select the **Songs** worksheet and click **[Transform Data]** button.



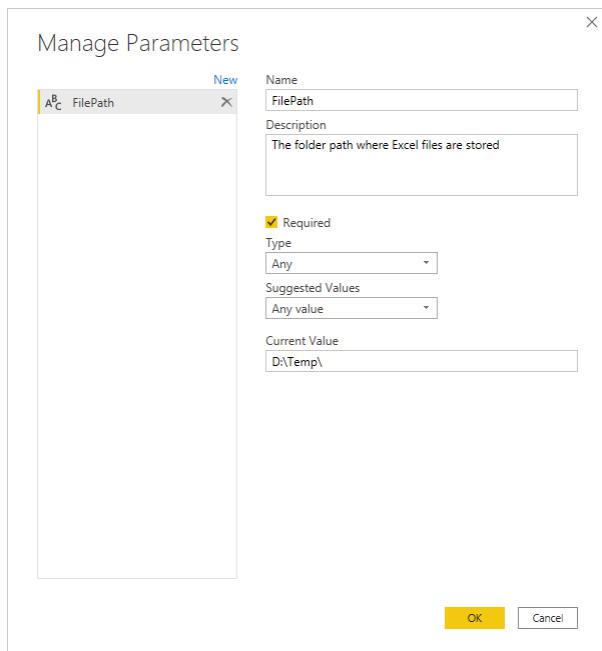
2. In Query Editor, select **Use First Row as Headers** in **Home** tab.



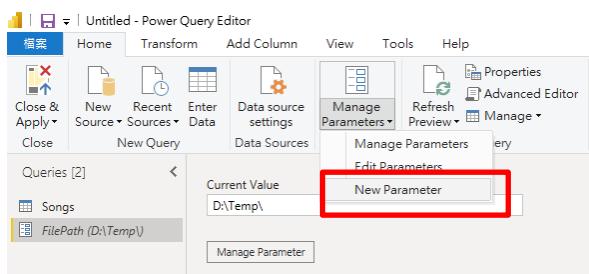
3. Select **Manage Parameter → New Parameters** in **Home** tab.



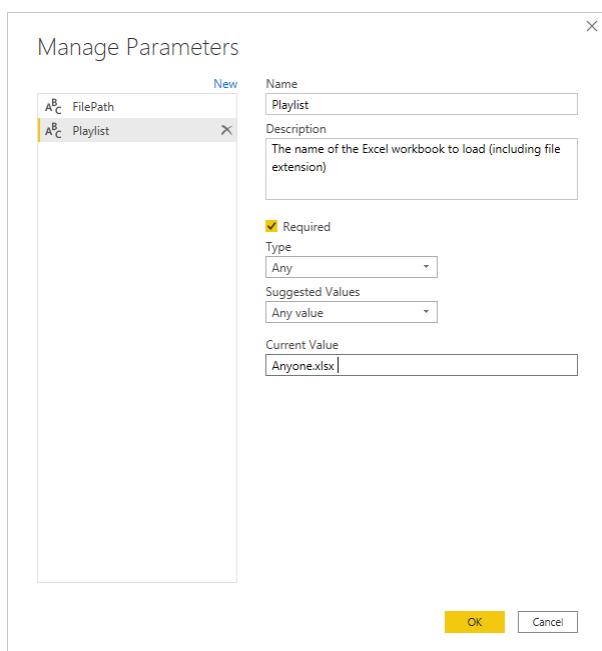
4. Create the first parameter “**FilePath**” to store the folder path where your Excel files are stored, then press [OK] to continue.



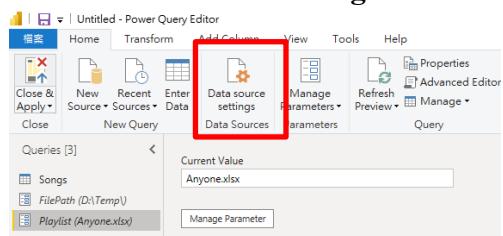
5. The parameter is created successfully, click **Manage Parameter** → **New Parameters** in Home tab again to create another parameter.



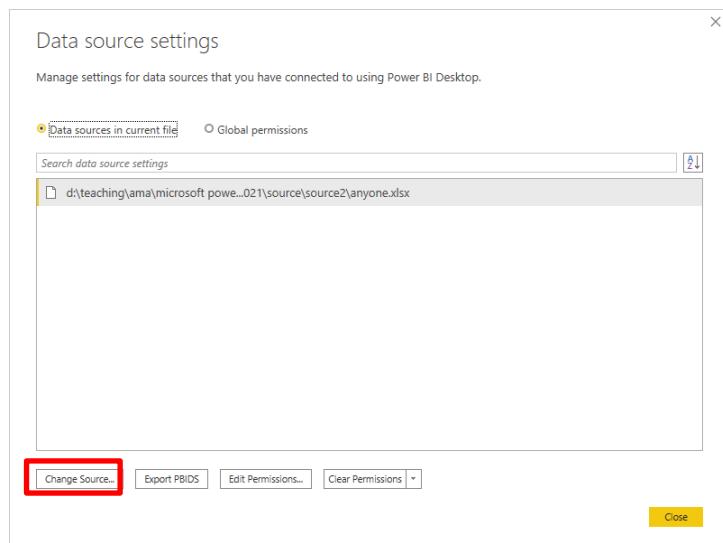
6. Create the second parameter “**Playlist**” to the name of the Excel workbook to load (including file extension)



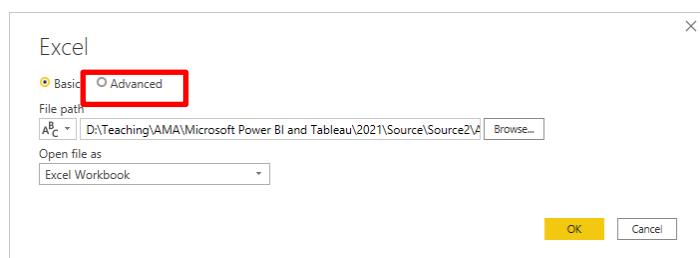
7. Select Data Source Setting in Home tab.



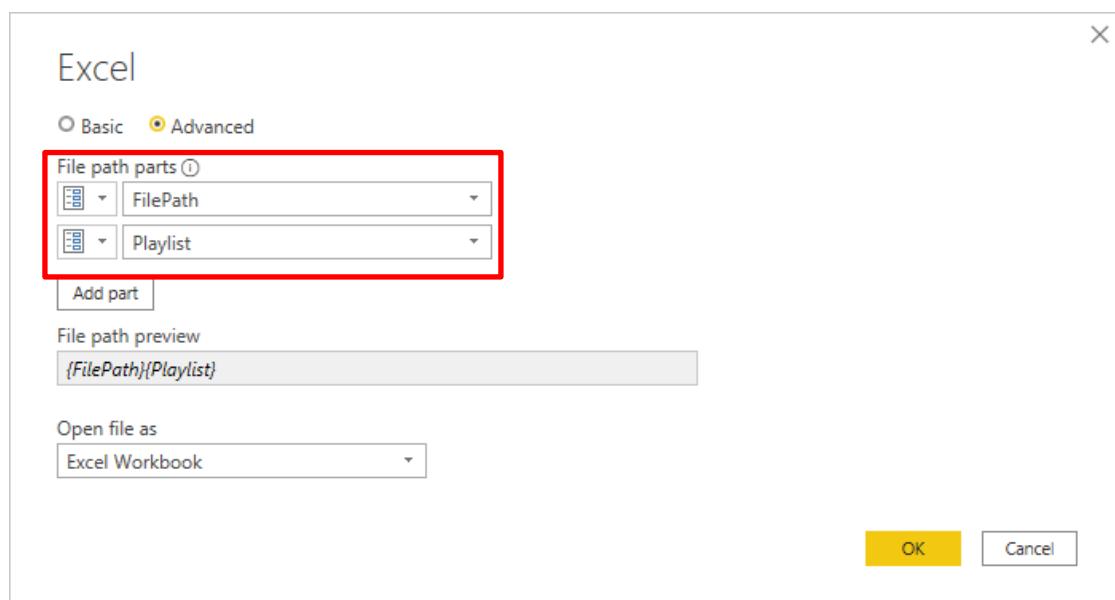
8. Select [Change Source]



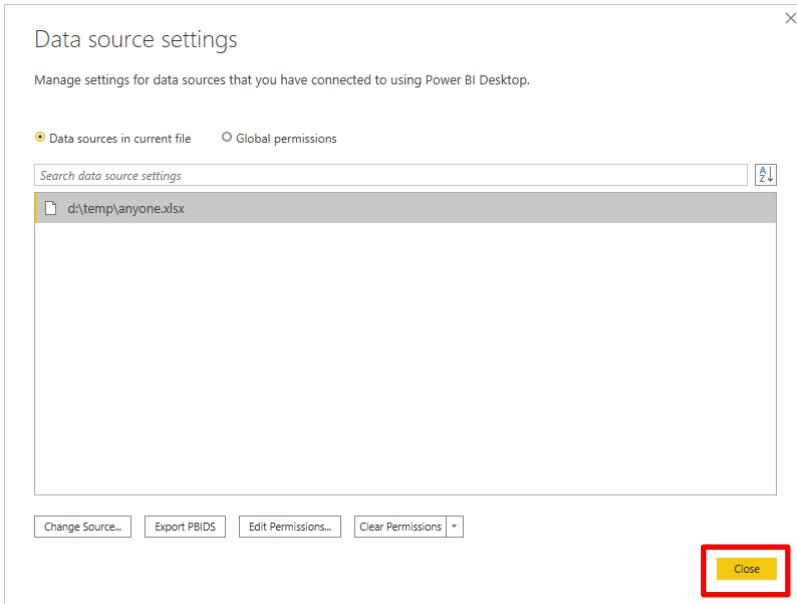
9. Select [Advanced].



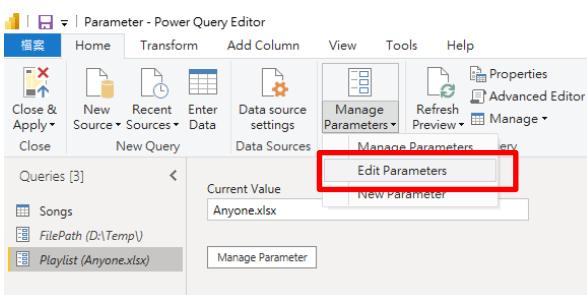
10. Change the file path part to parameter (*FilePath* and *Playlist*) and press [OK] to confirm



11. Press [Close] button to leave the Data source setting dialog.



12. Select Edit Queries → Edit Parameters.



13. Input the FilePath and Playlist and press [OK].



14. If you browse the Queries Songs, you should able to see the content from your new file. Press [Refresh Preview] to reload when necessary.

