Power Query Editor (Filter Data)

Objectives:

This chapter would teach you how to

- Power Query Custom Column & Conditional Column
- Manage Parameter.
- Introduction to Filter and types of filter
- Trend analysis, Future forecast

Custom Column

Power Query Custom Column is a feature in Power BI that allows you to add a new column to a data table by creating a custom calculation or transformation using existing columns. With Custom Column, you can create a formula that calculates a new value for each row in the data table based on the values of one or more existing columns.

Add Custom Column (Power Query)

You can add a custom column to your current query by creating a formula. Power Query validates the formula syntax in the same way as the **Query Editing** dialog box. For more information about the Power Query Formula Language, see Create Power Query formulas.

Add a custom column

1. To open a query, locate one previously loaded from the Power Query Editor, select a cell in the data, and then select **Query** > **Edit**. For more information see Create, edit, and load a query in Excel.
2. Select **Add Column** > **Custom Column**. the **Custom Column** dialog box appears.
3. Enter a new column name.
4. Insert a column into the **Custom Column Formula** box by selecting a column from the **Available Columns** list, and then selecting **Insert**.

   Note You can reference multiple columns as long as you separate them with an operator. For example, to calculate a TotalSales column, you add Total and SalesTax using the formula = each [Total] + [SalesTax].
5. Select **OK**.
6. Once you add a custom column, make sure it has an appropriate data type. If you see the **Any** ABC 123 icon to the left of the column header, change the data type to what you want.

Tip You can try another approach to get the results you want. Use a custom column to merge values from two or more columns into a single custom column. For more information, see Merge columns.

Common example formulas

The following table summarizes common examples of custom formulas.

Formula	Description
"abc"	Creates a column with the text abc in all rows.
1+1	Creates a column with the result of 1 + 1 (2) in all rows.
[UnitPrice] * [Quantity]	Creates a column with the result of multiplying two table columns.
[UnitPrice] * (1 – [Discount]) * [Quantity]	Calculates the total price, considering the **Discount** column.

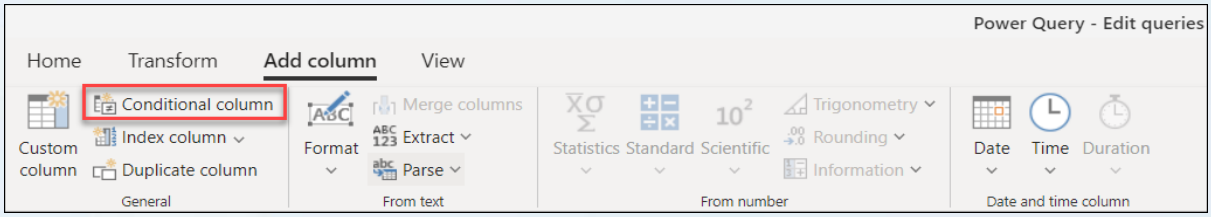
Formula	Description
"Hello" & [Name]	Combines Hello with the contents of the **Name** column in a new column.
Date.DayOfWeekName([DOB])	Creates a new column that displays a weekday name, such as Monday, derived from a **DOB** Date/Time column data type.
DateTime.Time([DOB])	Creates a new column that displays just the time derived from a **DOB** Date/Time column data type.
Date.Month([DOB])	Creates a new column that displays the month as a number from 1 to 12, such as 4 for April, derived from a **DOB** Date/Time column data type.

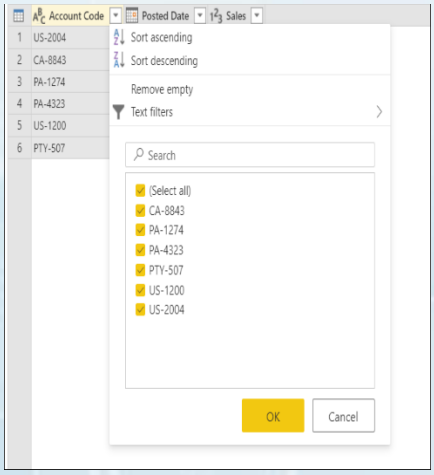
Conditional column

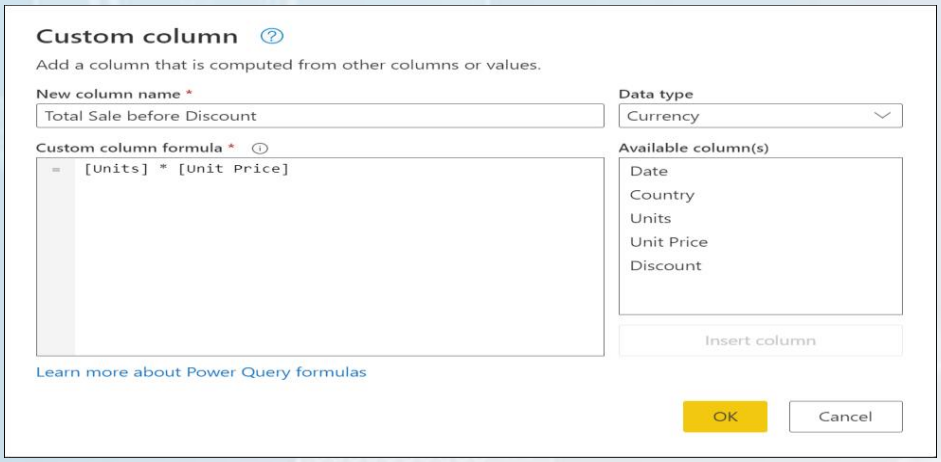
Conditional Column in Power BI is a feature in Power Query that allows you to add a new column to a data table based on a condition or set of conditions. With Conditional Column, you can create a formula that evaluates each row in the data table and returns a value for the new column based on whether the condition is met or not.

With Power Query, you can create new columns whose values will be based on one or more conditions applied to other columns in your table.
The **Conditional column** command is located on the **Add column** tab, in the **General** group.

In Power Query, you can include or exclude rows according to a specific value in a column. You can choose from multiple methods to filter the values in your column.







Add a Conditional Column (Power Query)

You can add a conditional column to your query by using a dialog box to create the formula.

1. To open a query, locate one previously loaded from the Power Query Editor, select a cell in the data, and then select **Query** > **Edit**. For more information see Create, load, or edit a query in Excel.
2. Select **Add Column** > **Conditional Column**. The **Add Conditional Column** dialog box appears to help you create a syntactically correct formula:
   if *condition* then *expression* else *expression*
3. In the **New column name** box, enter a unique name for your new conditional column. In the example, we change the name from **Custom** to **Region**.

4. In the **Column name** list box, select a column name. In the example, we select **Name & Postal,** which is a list of American states.
5. In the **Operator** list box, select an operator. In the example, we select **Contains**.
6. In the **Value** box, enter the appropriate value. In the example, we enter "Washington".
7. In the **Output** box, enter the output value that your conditional column should display when the if condition is true. In the example, we enter "West".
8. If you want to add an else expression when the condition is false, select **Add Clause**, and then repeat steps 4 to 6. In the example, additional regions are added including **South**, **Northeast**, and **Midwest**. This is commonly referred to as a nested condition.

   To delete or rearrange clauses, select **More** (...) next to the clause, and select a command.

   1. Optionally, add a final else expression. You can enter a value, another column, or a parameter.
   2. Select **OK**.

   Notes

- By default, a conditional column doesn't have a data type automatically defined. For more information, see Add or change data types.
- To modify the condition, select the **Edit Settings** icon next to the **Added Conditional column** step in the **Applied Steps** of the **Query Settings** pane.

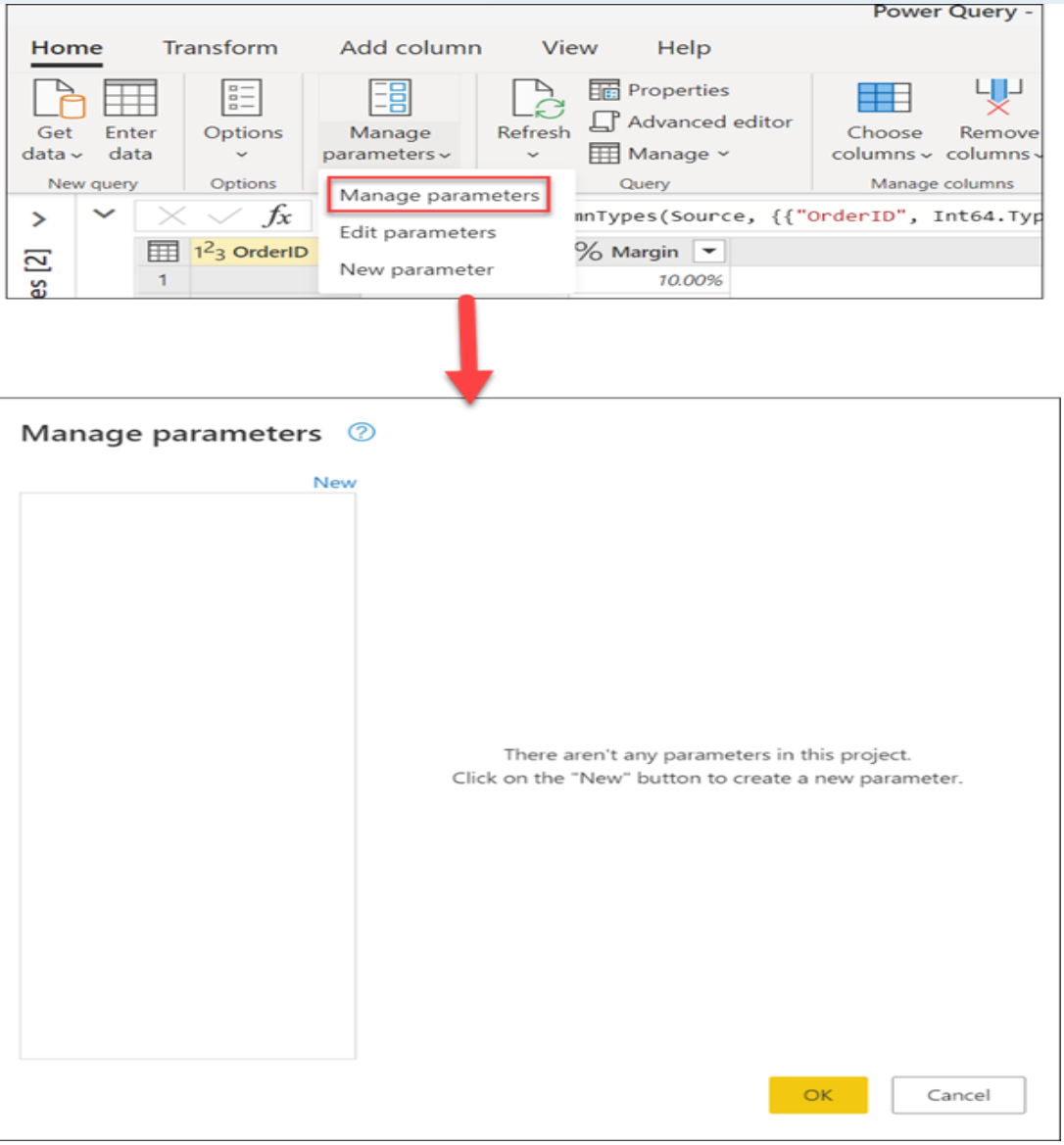
Using parameters

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.

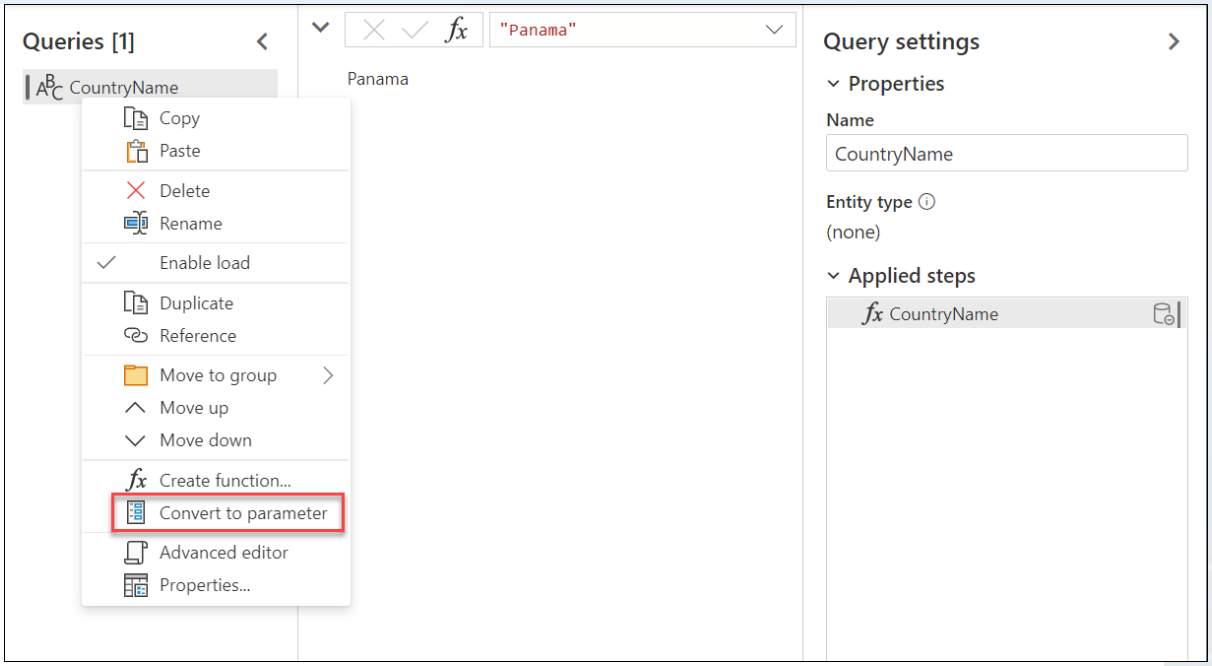
You can easily manage your parameters inside the **Manage Parameters** window. To get to the **Manage Parameters** window, select the **Manage Parameters** option inside **Manage Parameters** in the **Home** tab.



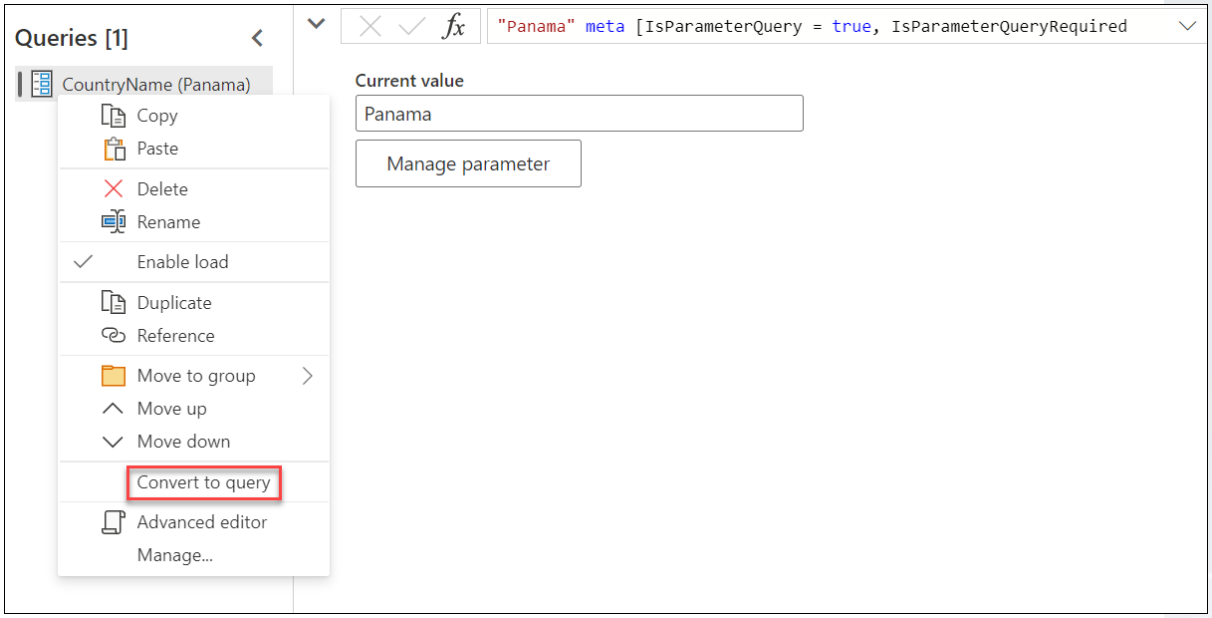
Creating a parameter

Power Query provides two easy ways to create parameters:

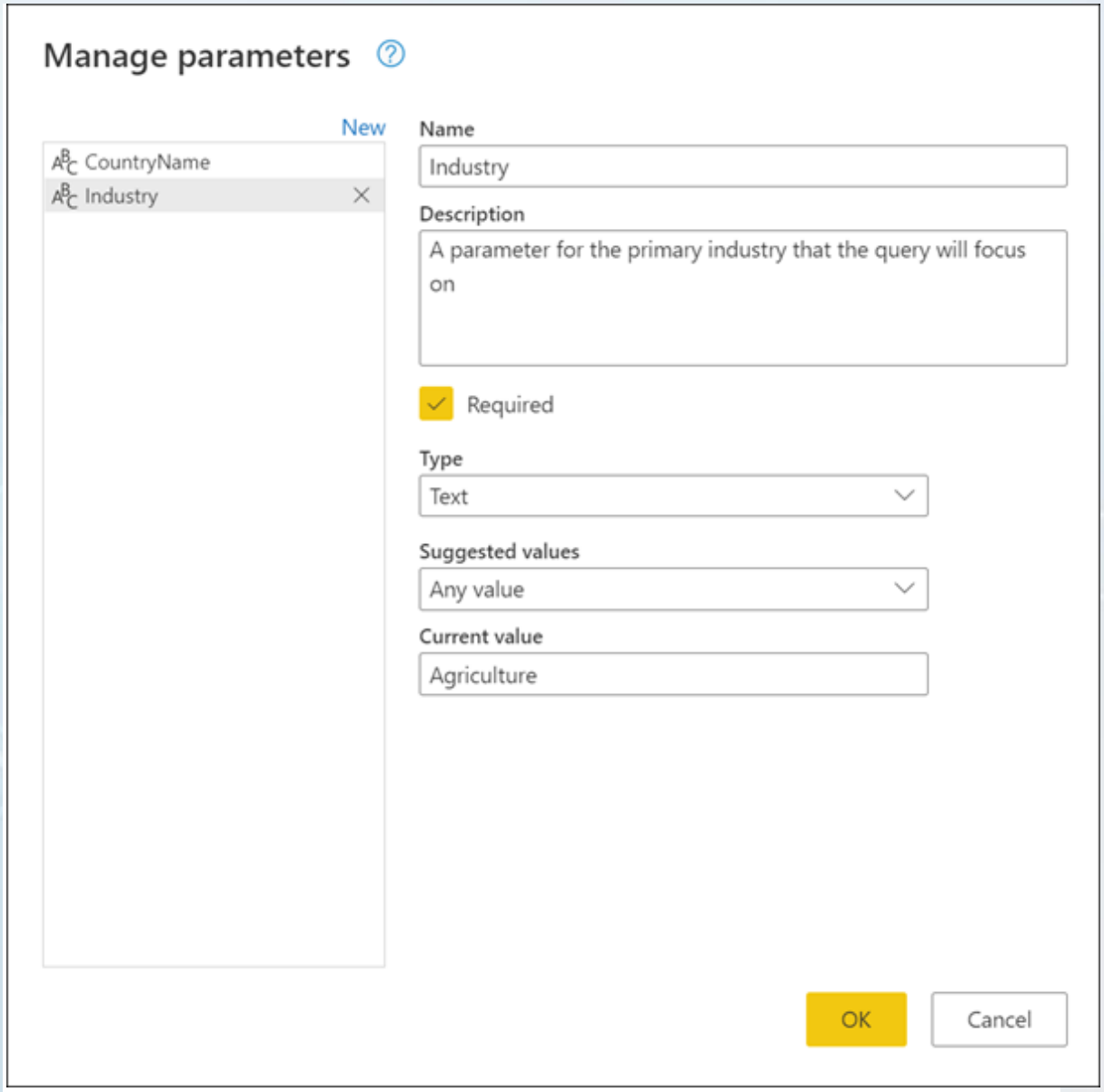
- **From an existing query**: Right-click a query whose value is a simple non-structured constant, such as a date, text, or number, and then select **Convert to Parameter**.



You can also convert a parameter to a query by right-clicking the parameter and then selecting **Convert to Query**.



- **Using the Manage Parameters window**: Select the **New Parameter** option from the dropdown menu of **Manage Parameters** in the **Home** tab. Or launch the **Manage Parameters** window and select **New** on the top to create a parameter. Fill in this form, and then select **OK** to create a new parameter.



After creating the parameter, you can always go back to the **Manage Parameters** window to modify any of your parameters at any moment.

Parameter properties

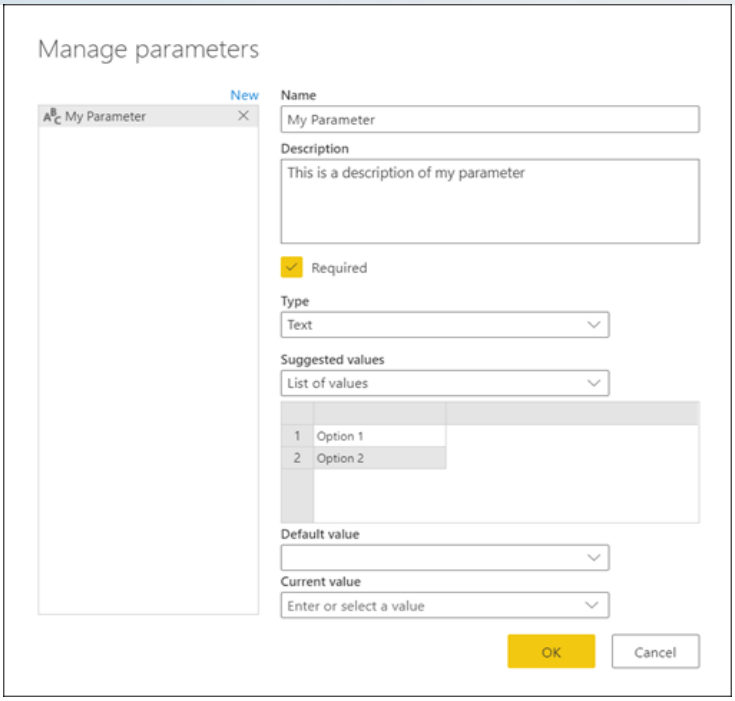
A parameter stores a value that can be used for transformations in Power Query. Apart from the name of the parameter and the value that it stores, it also has other properties that provide metadata to it. The properties of a parameter are:

- **Name**: Provide a name for this parameter that lets you easily recognize and differentiate it from other parameters you might create.

- **Description**: The description is displayed next to the parameter name when parameter information is displayed, helping users who are specifying the parameter value to understand its purpose and its semantics.
- **Required**: The checkbox indicates whether subsequent users can specify whether a value for the parameter must be provided.
- **Type**: Specifies the data type of the parameter. We recommended that you always set up the data type of your parameter

Suggested Values: Provides the user with suggestions to select a value for the **Current Value** from the available options:

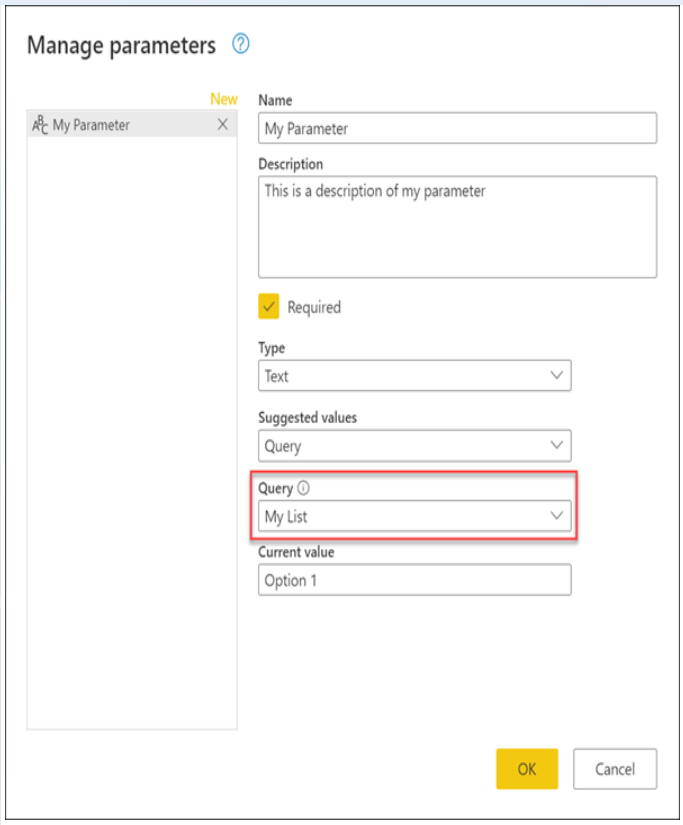
- **Any value**: The current value can be any manually entered value.
- **List of values**: Provides you with a simple table-like experience so you can define a list of suggested values that you can later select from for the **Current Value**. When this option is selected, a new option called **Default Value** will be made available. From here, you can select what should be the default value for this parameter, which is the default value shown to the user when referencing the parameter. This value isn't the same as the **Current Value**, which is the value that's stored inside the parameter and can be passed as an argument in transformations. Using the **List of values** provides a drop-down menu that's displayed in the **Default Value** and **Current Value** fields, where you can pick one of the values from the suggested list of values.



Note

You can still manually type any value that you want to pass to the parameter. The list of suggested values only serves as simple suggestions.

- **Query**: Uses a list query (a query whose output is a list) to provide the list of suggested values that you can later select for the **Current Value**.



- **Current Value**: The value that's stored in this parameter.

Where to use parameters

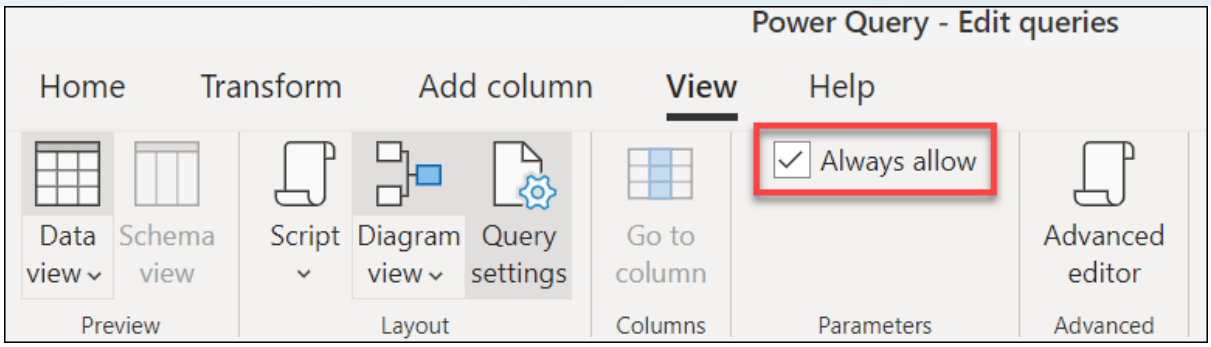
A parameter can be used in many different ways, but it's more commonly used in two scenarios:

- **Step argument**: You can use a parameter as the argument of multiple transformations driven from the user interface (UI).
- **Custom Function argument**: You can create a new function from a query and reference parameters as the arguments of your custom function.

In the next sections, you'll see an example for these two scenarios.

Step argument

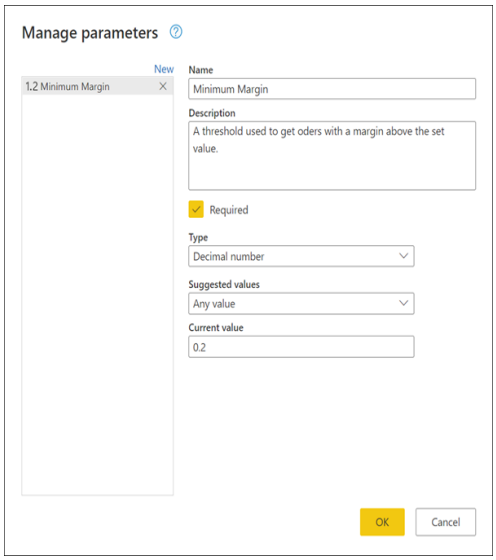
To enable this feature, first go to the **View** tab in the Power Query editor and select the **Always allow** option in the **Parameters** group.



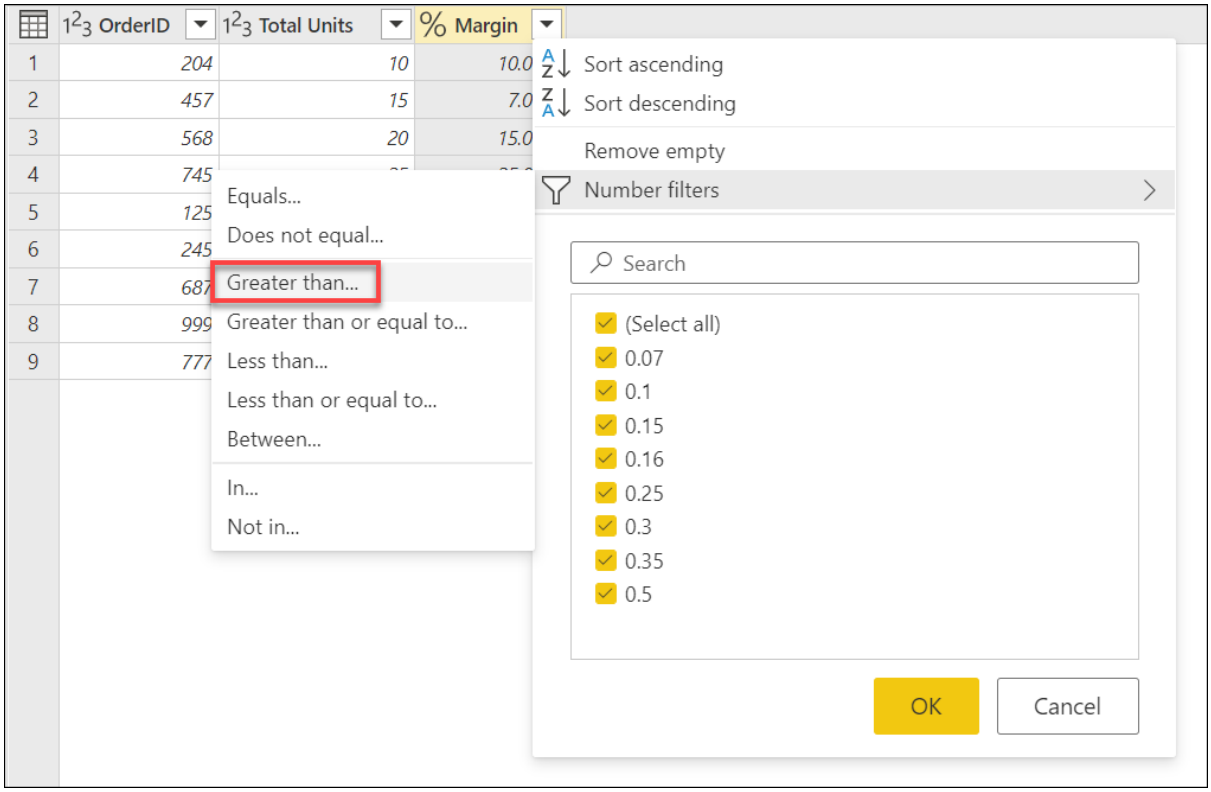
For example, the following **Orders** table contains the **OrderID**, **Units**, and **Margin** fields.

	OrderID	Total Units	Margin
1	204	10	10.00%
2	457	15	7.00%
3	568	20	15.00%
4	745	25	25.00%
5	125	30	30.00%
6	245	100	50.00%
7	687	42	16.00%
8	999	3000	16.00%
9	777	2500	35.00%

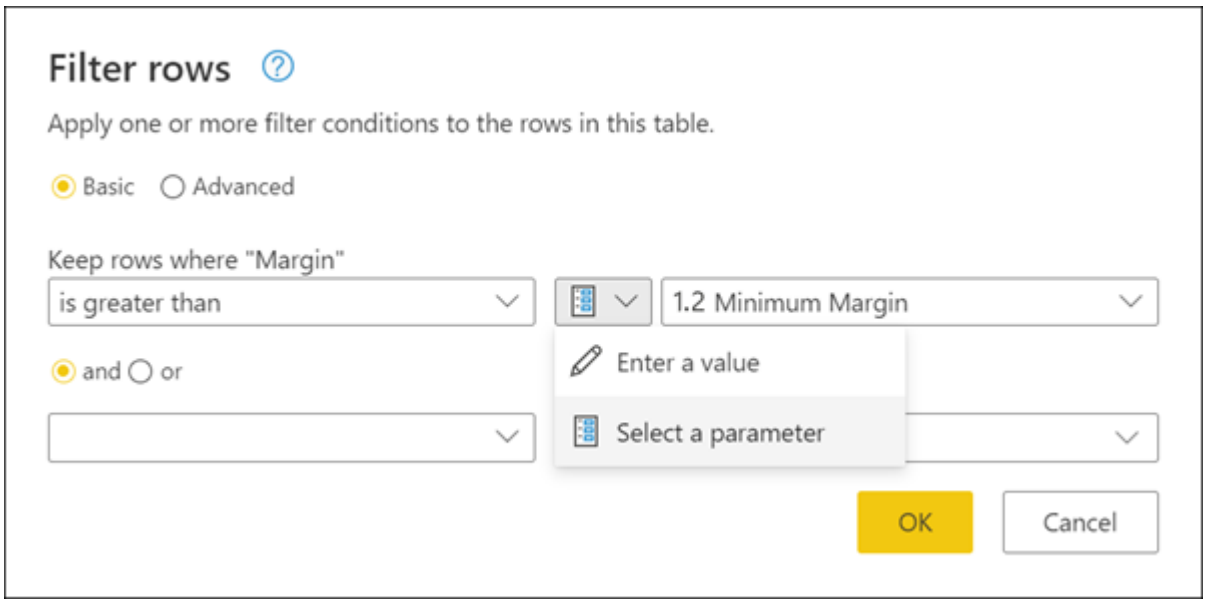
In this example, create a new parameter with the name **Minimum Margin** with a **Decimal Number** type and a **Current Value** of 0.2.



Go to the **Orders** query, and in the **Margin** field select the **Greater Than** filter option.



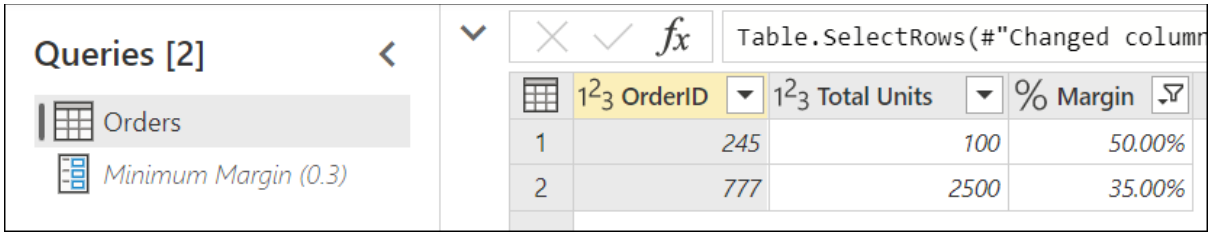
In the **Filter Rows** window, there's a button with a data type for the field selected. Select the **Parameter** option from the dropdown menu for this button. From the field selection right next to the data type button, select the parameter that you want to pass to this argument. In this case, it's the **Minimum Margin** parameter.



After you select **OK**, your table is filtered using the **Current Value** for your parameter.

	OrderID	Total Units	Margin
1	745	25	25.00%
2	125	30	30.00%
3	245	100	50.00%
4	777	2500	35.00%

If you modify the **Current Value** of your **Minimum Margin** parameter to be 0.3, your orders query gets updated immediately and shows you only the rows where the **Margin** is above 30%.

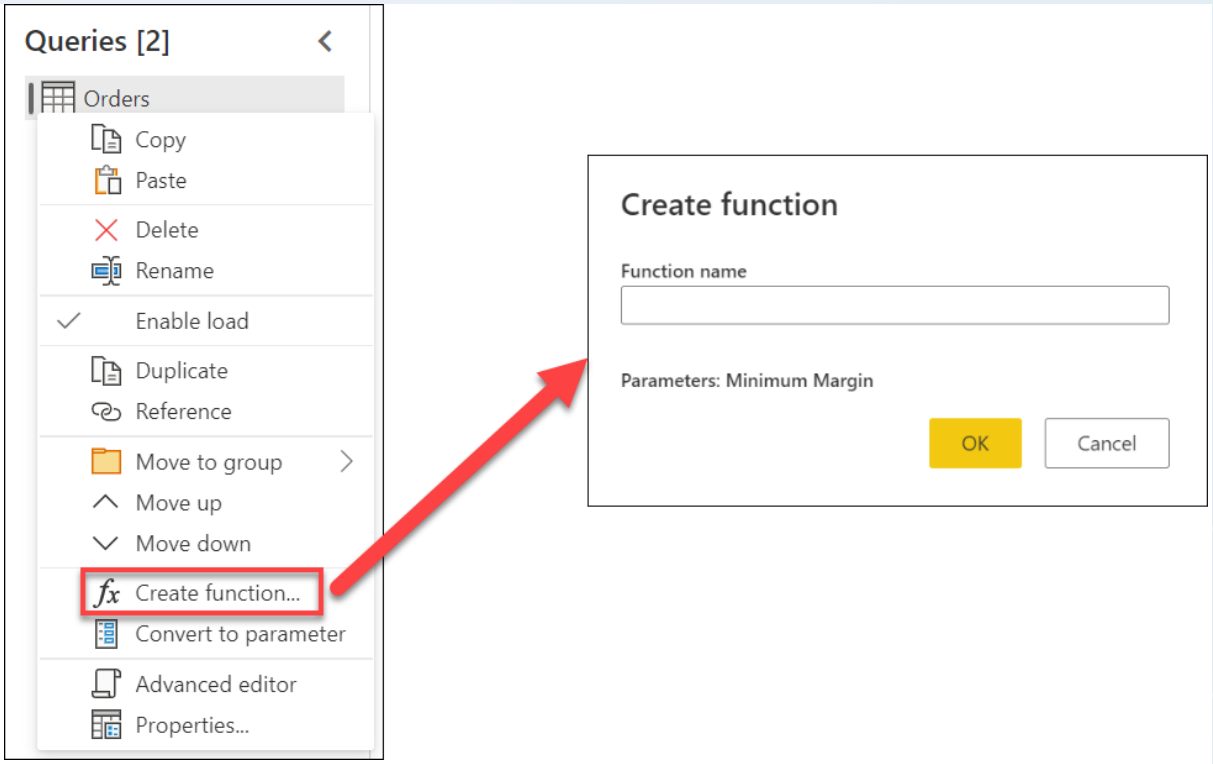


Tip

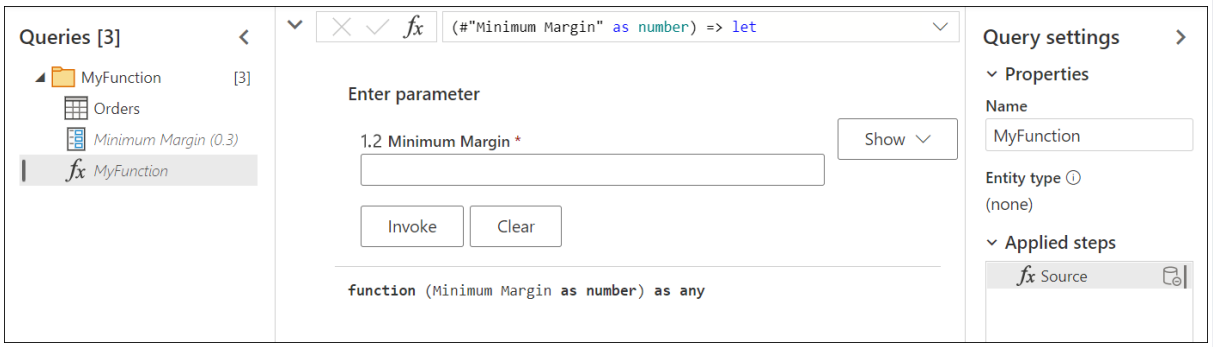
Many transformations in Power Query let you select your parameter from a dropdown.

Custom function argument

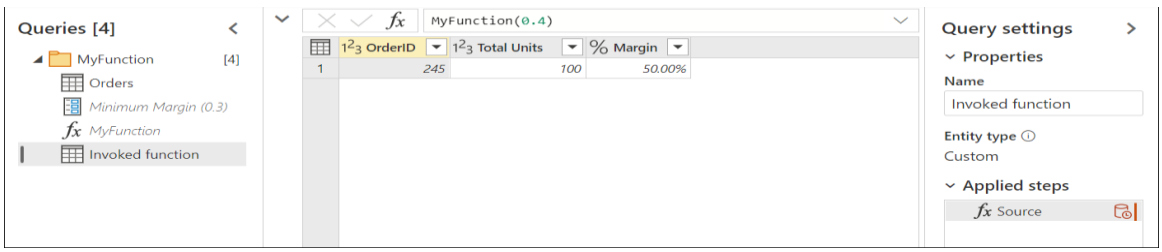
With Power Query, you can create a custom function from an existing query with a simple click. Following the previous example, right-click the **Orders** query and select **Create Function**. This action launches a new **Create Function** window. In this window, name your new function and it will tell you the parameters being referenced in your query. These parameters are used as the parameters for the custom function.



You can name this new function however you want. For demonstration purposes, the name of this new function is **MyFunction**. After you select **OK**, a new group is created in the **Queries** pane using the name of your new function. In this group, you'll find the parameters being used for the function, the query that was used to create the function, and the function itself.



To test this new function, enter a value, such as 0.4, in the field underneath the **Minimum Margin** label. Then select the **Invoke** button. This creates a new query with the name **Invoked Function**, effectively passing the value 0.4 to be used as the argument for the function and giving you only the rows where the margin is above 40%.



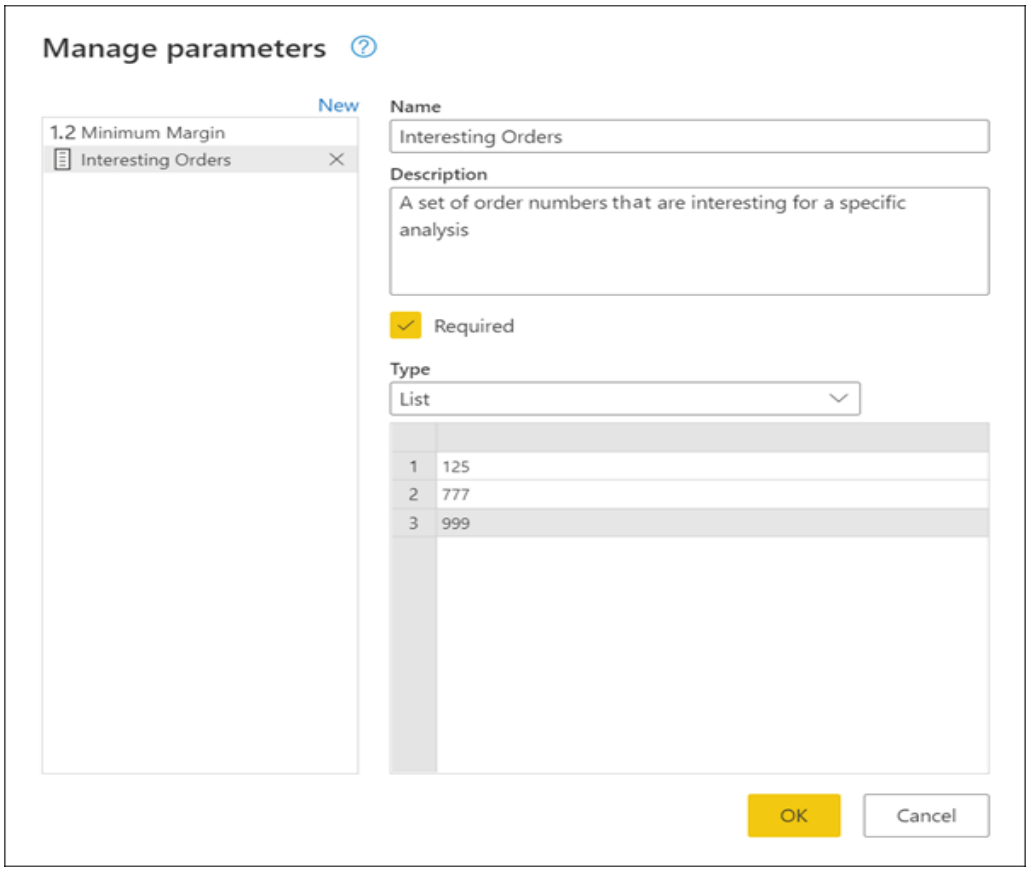
Multi-value or list parameters

A new type of parameter available only in Power Query Online is multi-value or list parameters. This section describes how to create a new list parameter and how to use it in your queries.

Following the previous example, change the current value for **Minimum Margin** from **0.3** to **0.1**. The new goal is to create a list parameter that can hold the order numbers of the orders that you're interested in analysing. To create the new parameter, go to **Manage Parameters** dialog and select **New** to create a new parameter. Fill in this new parameter with the following information:

- **Name:** Interesting Orders
- **Description:** A set of order numbers that are interesting for a specific analysis
- **Required:** True
- **Type:** List

After defining these fields, a new grid pops up where you can enter the values that you want to store for your parameter. In this case, those values are **125**, **777**, and **999**.



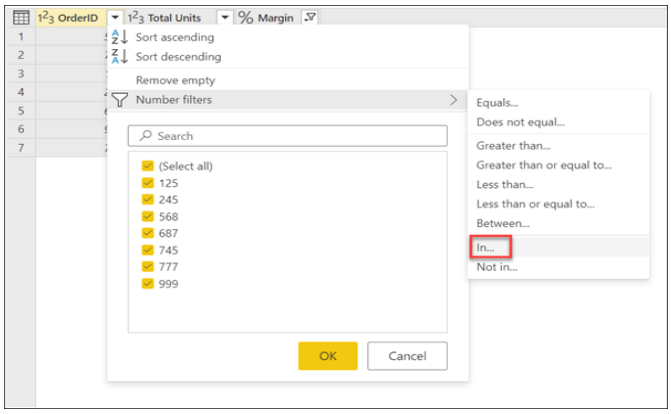
Note

While this example uses numbers, you can also store other data types in your list, such as text, dates, datetime, and more.

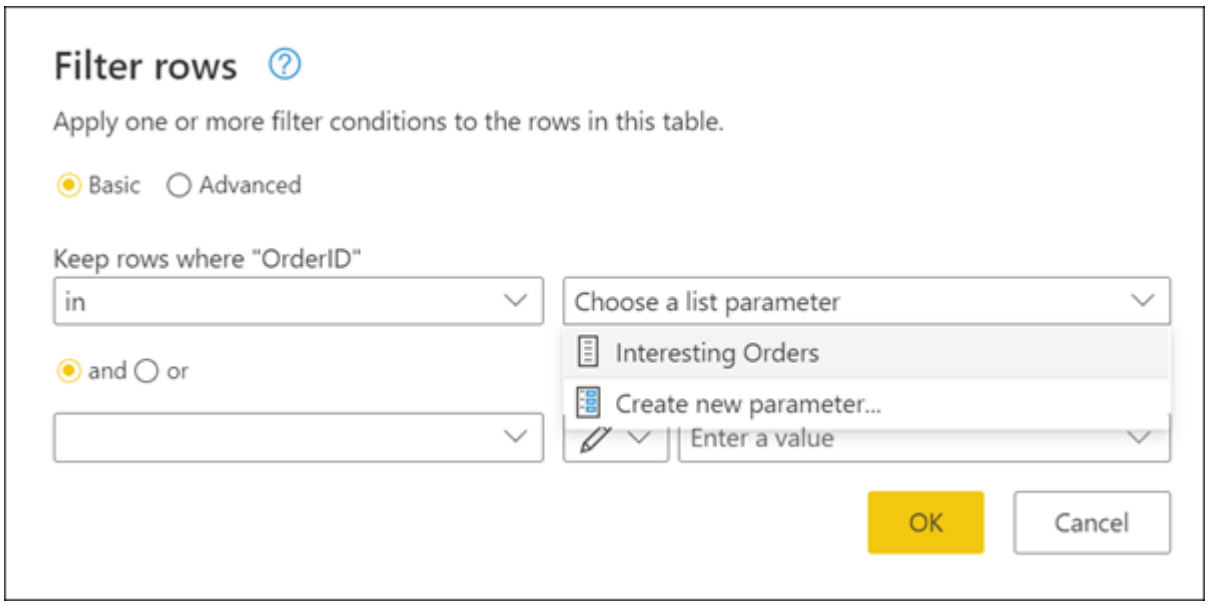
Tip

If you want to have more control over what values are used in your list parameter, you can always create a list with constant values and convert your list query to a parameter as showcased previously in this article.

With the new **Interesting Orders** list parameters in place, head back to the **Orders** query. Select the auto-filter menu of the **OrderID** field. Select **Number filters** > **In**.



After selecting this option, a new **Filter rows** dialog box appears. From here, you can select the list parameter from a drop-down menu.



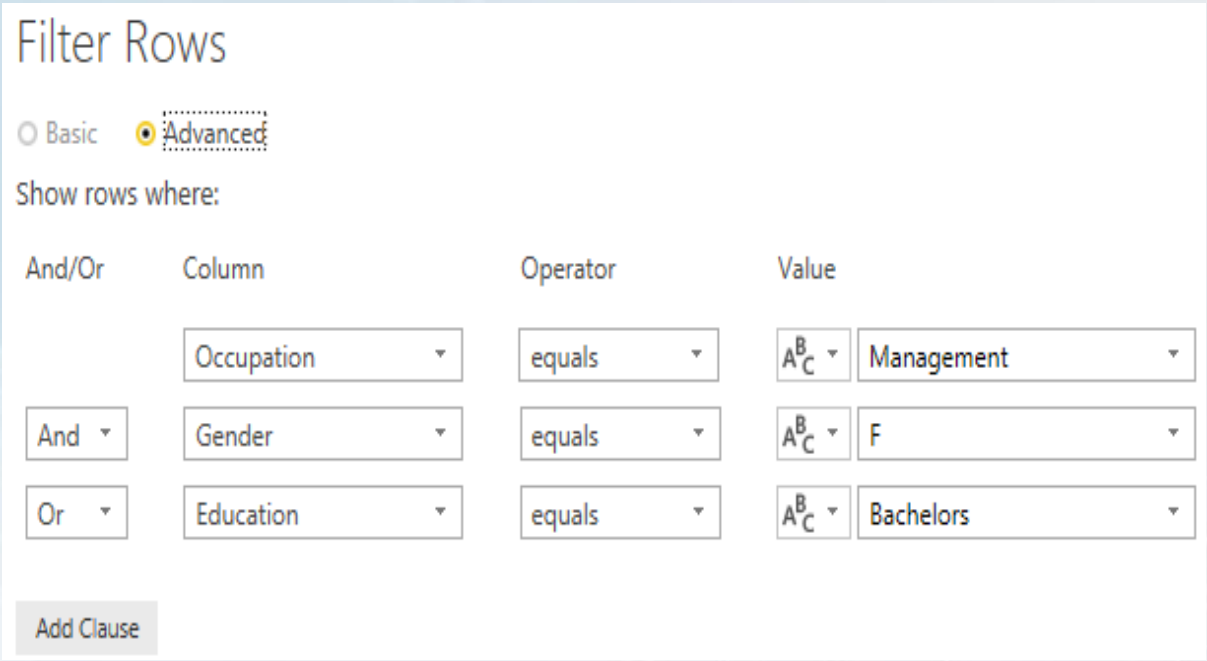
Note

List parameters can work with either the **In** or **Not in** options. **In** lets you filter only by the values from your list. **Not in** does exactly the opposite, and tries to filter your column to get all values that are not equal to the values stored in your parameter.

After selecting **OK**, you'll be taken back to your query. There, your query has been filtered using the list parameter that you've created, with the result that only the rows where the **OrderID** was equal to either **125**, **777**, or **999** was kept.

	1²3 OrderID	1²3 Total Units	% Margin
1	125	30	30.00%
2	999	3000	16.00%
3	777	2500	35.00%

Filter Rows

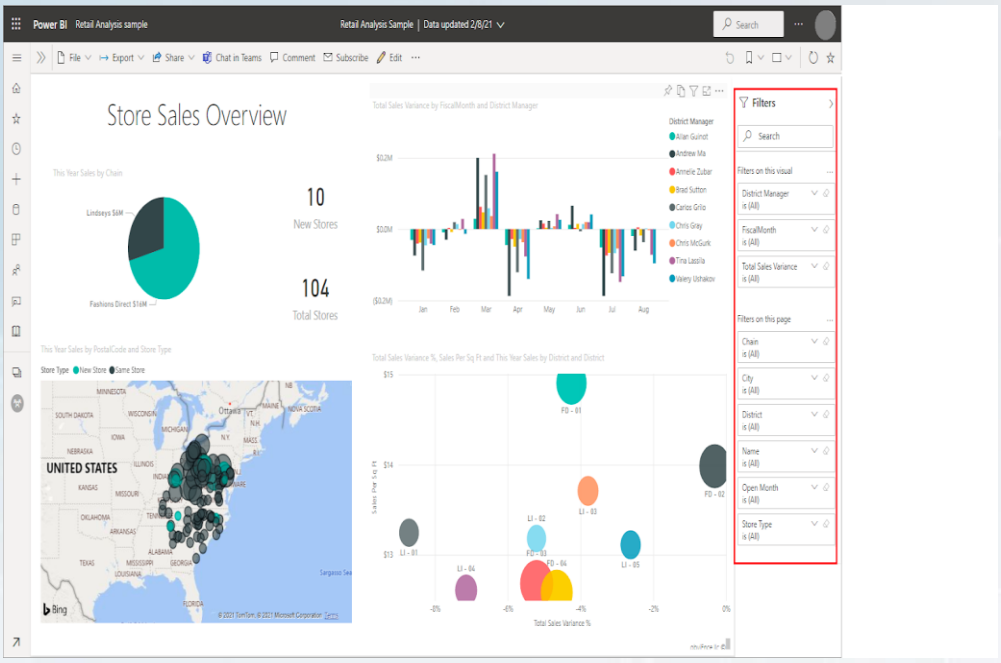


What is a Power BI Filter

Power BI filters are essentially functions that allow you to isolate a smaller part of your entire data set and analyse it more easily. Filters separate information that is relevant at the time from irrelevant or extraneous data that would not help you in decision-making.

Power BI filters don't delete your unwanted or extraneous data. Instead, this data is dimmed down and only the important fields get highlighted.

Filters pane



To apply Power BI filters to your data, you can visit the **Filters pane** on the right side of your Power BI screen. Here you can mark the criteria by which you want to sort your data and view new information related to that on your dashboard.

Types of Power BI Filters

There are four types of Power BI filters available in the Microsoft Power BI platform:

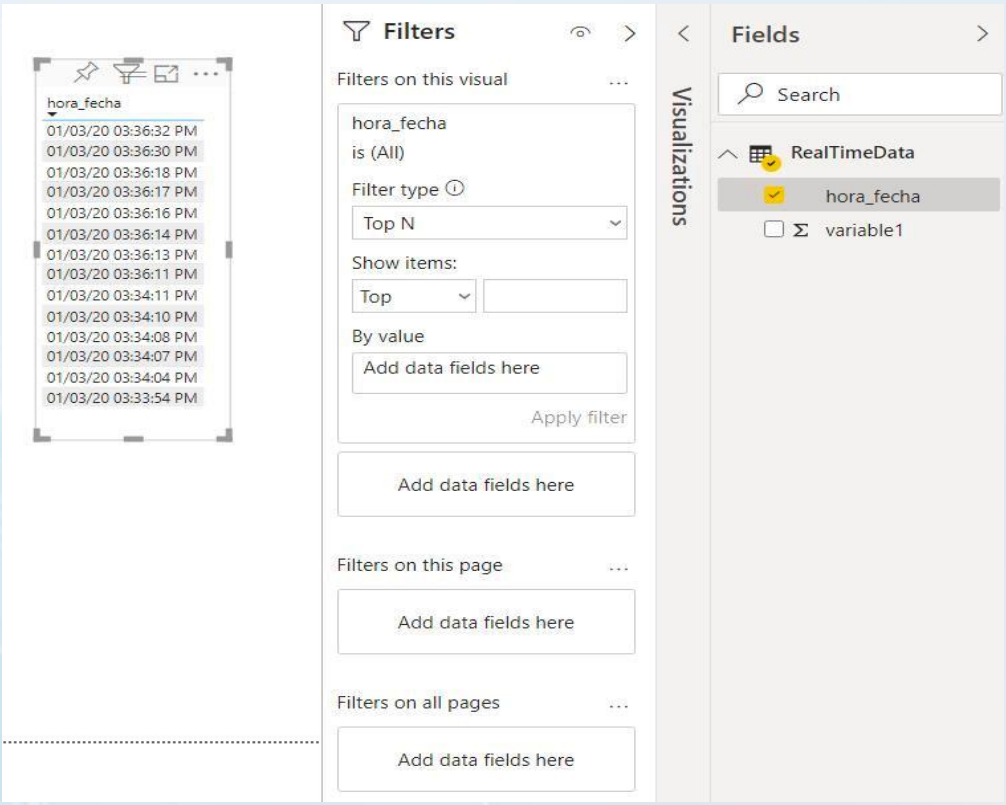
- **Visual Filters**: Visual filters sort data in a data visual like table, chart, card, slicer, etc. These are applied to both the data and the calculation conditions used within the visualization. To apply a visual filter, you just have to click a visual and select your filters from the Filters pane.

- **Page Filters**: Page filter applies filters to all its included visuals. To use your page-level filters, you can use the option "Filters on this page" from the Filters pane. Page filters are incredibly helpful while preparing company-wide reports, wherein you can create separate filters for separate pages of your Sales, Marketing, and Support.

- **Report Filters**: Report filters work on the entire report (collection of pages). These are universal filters that sort everything included in your report.

- **Drill through Filters**: Drill through filter allows you to create a destination report page that focuses on a specific entity. For example, if you would like to prepare data visualizations for your top-performing region, you can use drill through filters and prepare a separate page catering to all the important aspects of your region.

Here's a snapshot of the hierarchy available in the Power BI Filters pane. Starting from the top, you have "**Filters on this visual**" or Visual filters, then "**Filters on this page**" or Page filters followed by "**Filters on all pages**" or Report filters.



How to Add a Power BI Filter to a Visual?

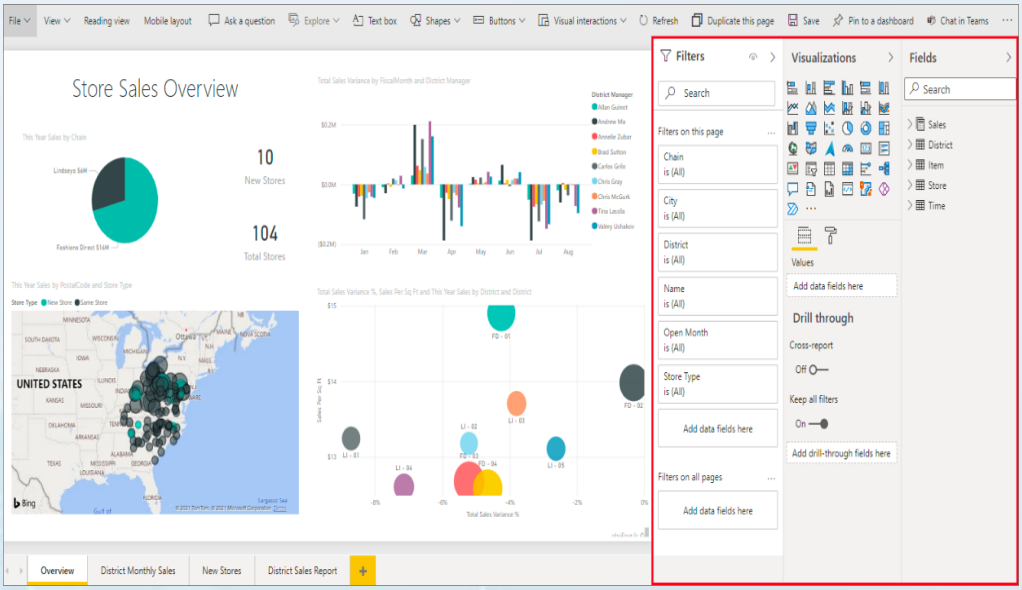
To be able to use Power BI filters, access your Power BI dashboard in the **Editing view**. If you access your report in the **Reading view**, you cannot edit, i.e., add or remove filters that are applied to the report.

Power BI filters for data visualizations can be added in two ways:

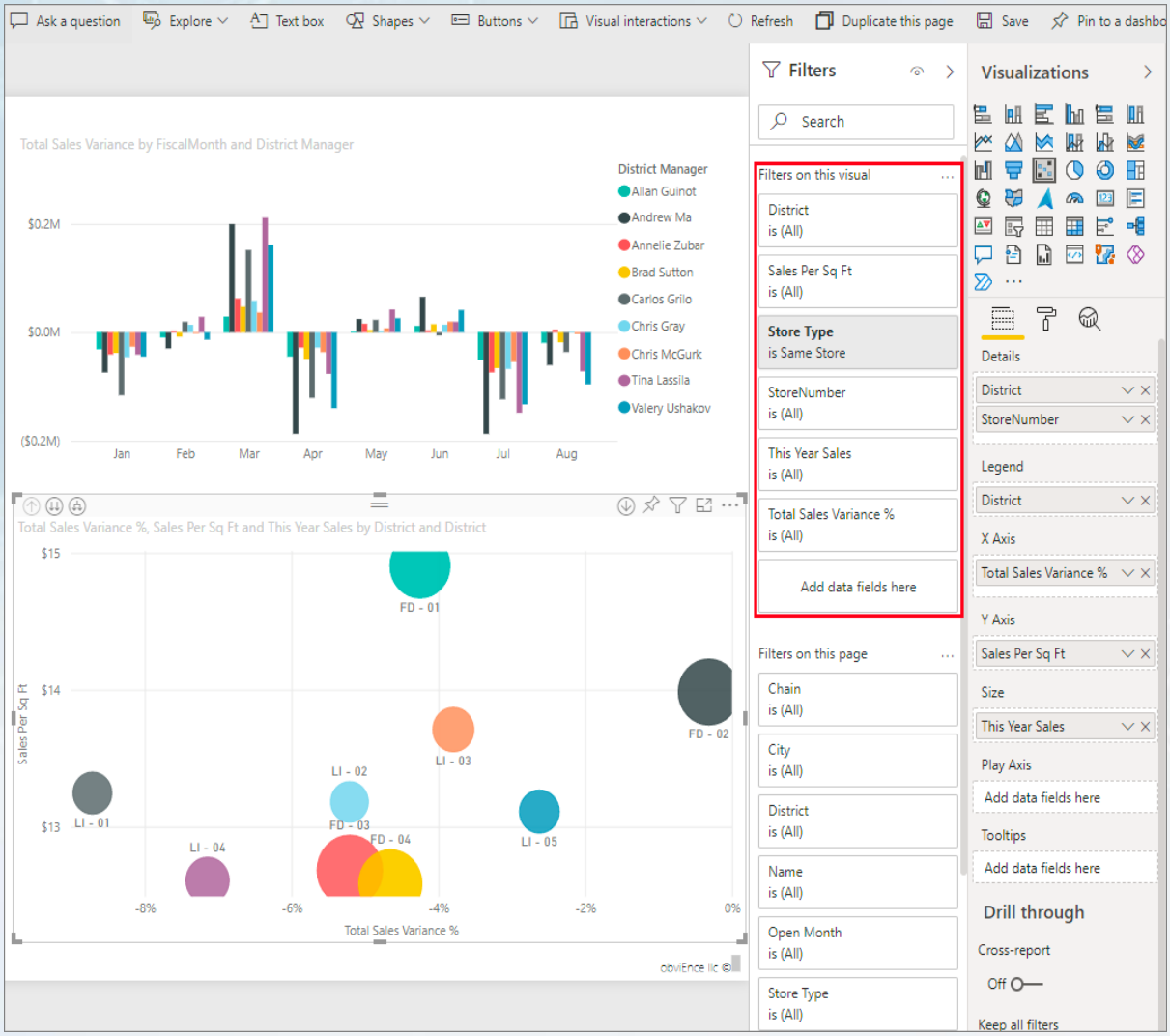
- Use **predefined data fields** as filters in your visual. These can be accessed from the available options under the header "Filters on this visual tab".
- Use a **new or custom field** that isn't available in the visual as a Power BI filter. To add such new filters, follow the steps mentioned below:

Step 1: Select the **Edit** option from the top pane in your Power BI dashboard.

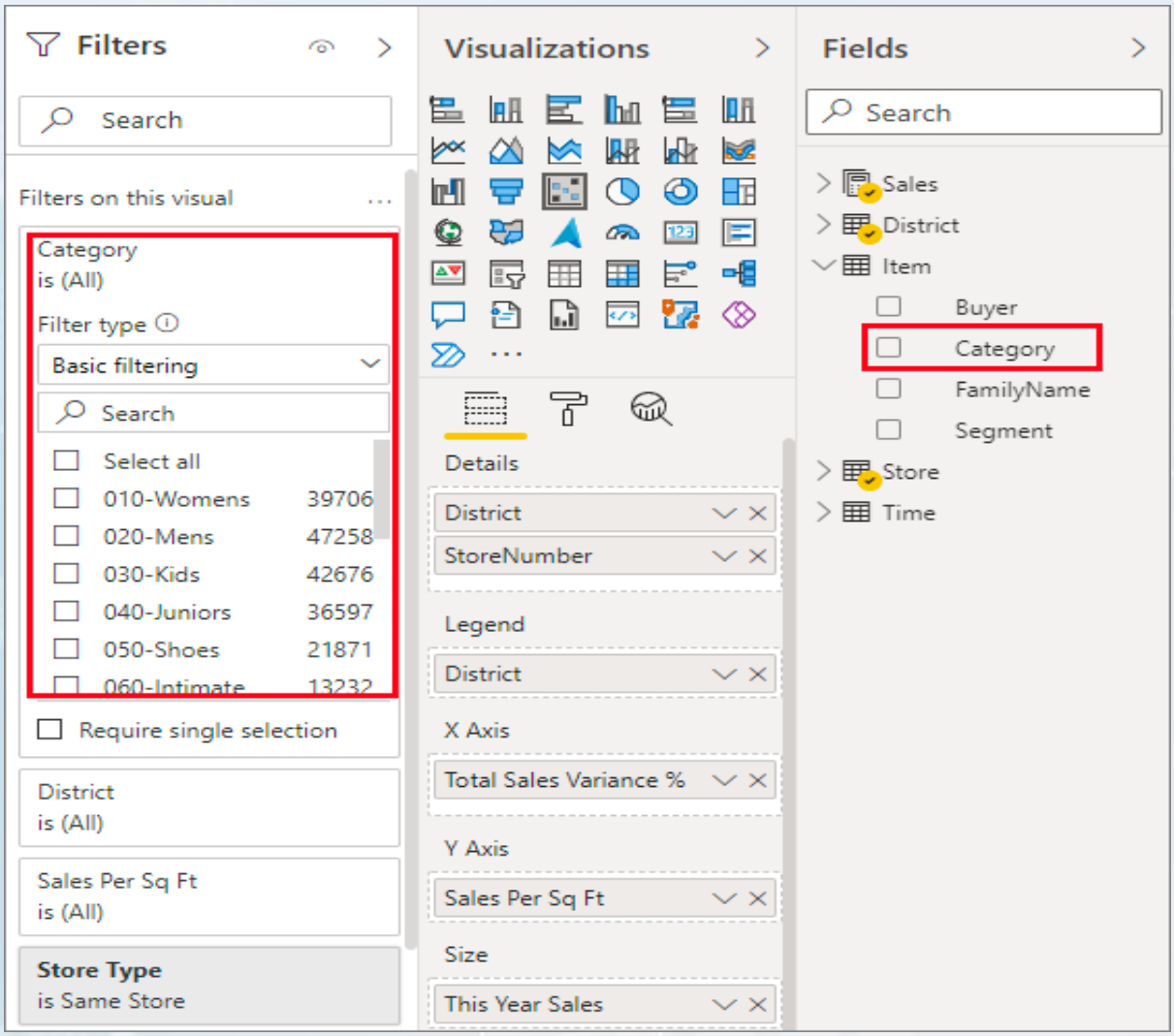
Step 2: Expand **Visualizations, Filters, and Fields** panes from the right side.



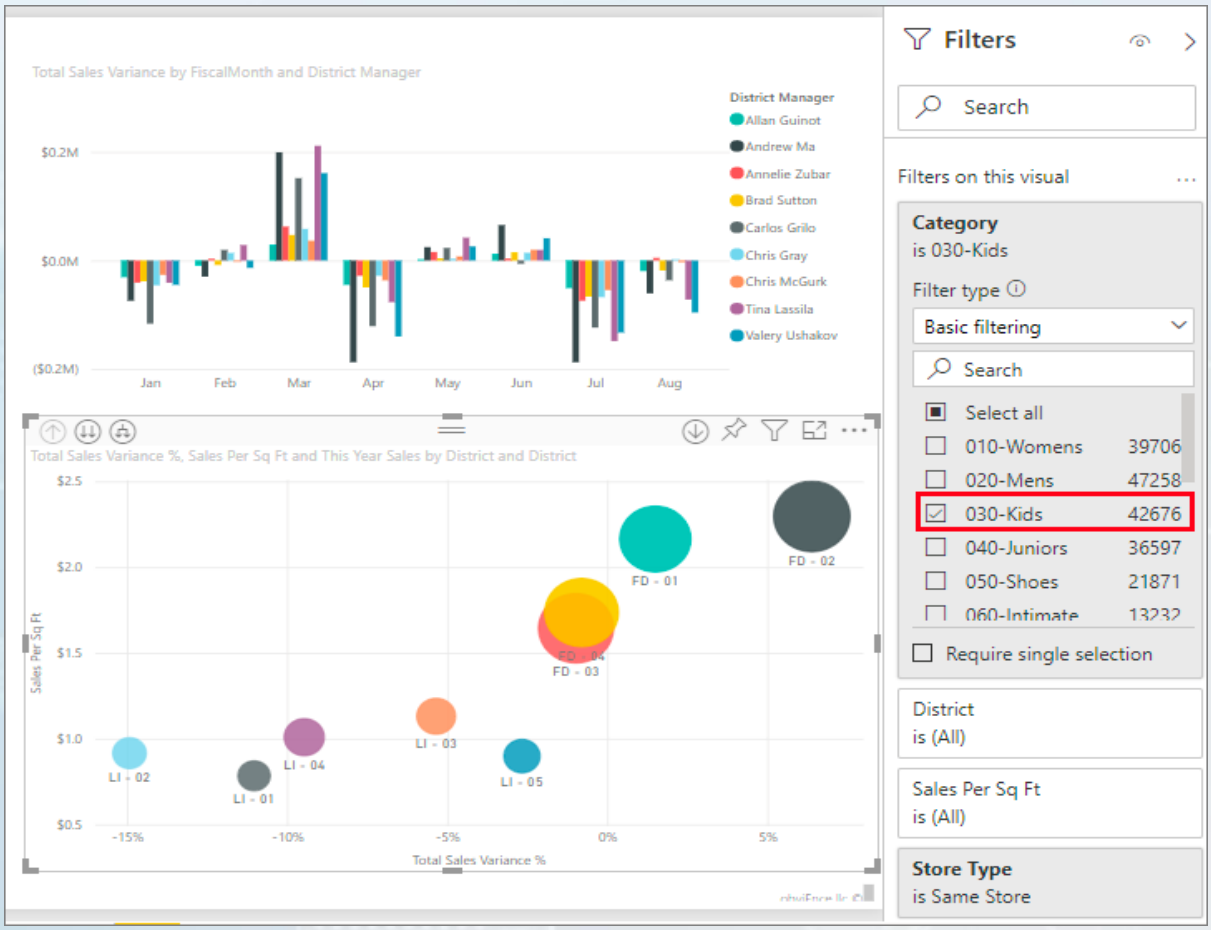
Step 3: Choose the graphic to which you wish to apply the new filter. When you pick a visual, the fields for that visual will appear in the **Visualizations** window. They are also listed in the Filters pane, under the Filters heading on this visual.



Step 4: Select the field you wish to add as a new visual-level filter from the Fields pane and drag it into the Filters on this visual area. In this example, we drag **Category** to the **Add data fields here** section.



Step 5: Pick your new filter from the scroll-down menu. The new filter(s) will be applied but the other visuals will remain the same.



Note: When you drag a numeric column to the Filters pane to create a visual-level filter, the filter is applied to the rows of data underlying it.

Trend analysis, Future forecast In Power BI

Power BI offers a range of features for trend analysis and future forecasting that allow you to analyse data over time and make predictions about future trends. Here are some steps to follow to perform trend analysis and future forecasting in Power BI:

1. **Import your data into Power BI:** The first step is to import your data into Power BI. This can be done using the Get Data function, which allows you to connect to a variety of data sources.

2. **Create a line chart:** Once your data is imported, create a line chart in Power BI. This will allow you to visualize trends in your data over time. You can customize the chart to show the specific metrics you are interested in tracking.

3. **Analyse trends:** Once your chart is created, you can analyse trends by looking for patterns in the data. Use the analytics pane to add trend lines and other statistical information to the chart. This will allow you to identify trends and patterns in the data.
4. **Create a forecast:** Power BI also allows you to create forecasts based on historical data. Use the forecasting feature to create a forecast for the future based on the trends you have identified. You can adjust the forecast by changing the confidence interval and other parameters.

5. **Monitor and refine:** Once your forecast is created, you can monitor it over time and refine it as necessary. Use the data analysis tools in Power BI to identify any changes in trends and adjust your forecast accordingly.

Overall, Power BI offers a range of tools for trend analysis and future forecasting. By importing your data, creating a line chart, analysing trends, creating a forecast, and monitoring and refining your analysis, you can make data-driven predictions about future trends in your business.