

# How to Change the Legend Order in Power BI: A Step-by-Step Guide

Authored by  
**stats writer**

January 26, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Change the Legend Order in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127810>

# Mastering Custom Legend Ordering in Power BI

## Introduction to Custom Legend Ordering in Power BI

Power BI is a leading platform in the field of Business Intelligence, enabling users to transform raw datasets into dynamic, interactive reports and visualizations. A crucial component of any visual is the legend, which serves to map the data categories or groups represented within the chart. While Power BI offers various standard sorting options, the ability to control the sequence of items in the legend is often essential for effective data storytelling and alignment with business logic. When dealing with stages, processes, or hierarchical categories, the default alphabetical sorting imposed by the tool can be counterproductive, obscuring the natural flow of the data.

Changing the default order of the legend is not merely a cosmetic adjustment; it is a vital step in ensuring that the visual narrative is logical and easily digestible for the end-user. For instance, in a visualization tracking an operational workflow, it is imperative that the process steps--such as "Received," "Located," "Packed," and "Shipped"--appear in their true chronological sequence, rather than being sorted alphabetically as "Located," "Packed," "Received," and "Shipped." This advanced level of control requires utilizing the inherent capabilities of the modeling layer in Power BI, specifically by creating calculated columns that enforce a predetermined, custom sort order on the categorical data.

The common misconception is that legend sorting can be fully managed solely within the Visual Format pane. While this pane allows for general adjustments and toggling of sort direction based on measures (e.g., sorting regions by sales amount), custom sorting of qualitative categories requires a structured approach using the underlying data model. The technique involves defining a corresponding numeric or ordinal column for the categorical column used in the legend, and then instructing Power BI to use this new column for sorting purposes. This method grants analysts precise governance over the presentation layer, enhancing the clarity and professional quality of the final reports.

## Understanding the Default Sorting Mechanism in Visualizations

When a categorical field is added to a visual's Legend or Axis well--particularly when measures are stacked or grouped--Power BI, by default, applies an alphabetical sort to that field. This inherent behavior is efficient for simple, non-sequential datasets but becomes problematic when the categories themselves represent steps in a process or have an intrinsic ordinal relationship that supersedes alphabetical designation. The default sort is applied both to the axis labels (if applicable) and the legend items themselves, potentially disrupting the intended data flow.

Consider a scenario involving a stacked column chart used to monitor inventory movement. If the

categories representing the item status are "Awaiting Inspection," "In Transit," and "Delivered," an alphabetical sort would display them as "Awaiting Inspection," "Delivered," and "In Transit." This sequence is illogical from a logistical perspective, making it difficult for stakeholders to quickly interpret the status progression. Relying on default sorting in such contexts often leads to confusion and misinterpretation of trends or operational bottlenecks.

To override this automatic sorting, we must leverage the data modeling capabilities provided by Power BI. The key insight is that Power BI can use one column to determine the visual sequence of another column, provided the relationship between the two columns is clearly defined. This critical process is managed outside the report canvas--specifically within the Data View or Model View--by defining the categorical column (the legend text) to be sorted by its related numerical index column (the desired order).

## The Necessity of Custom Sorting Columns using DAX

To achieve a custom legend order that aligns with business logic, the creation of specific calculated columns is indispensable. These columns act as the invisible backbone for sorting the visible categories. The language used for creating these calculation rules is Data Analysis Expressions, or DAX. While the implementation may seem complex initially, the use of functions like `SWITCH` allows for a straightforward mapping of categorical text values to specific numerical indexes.

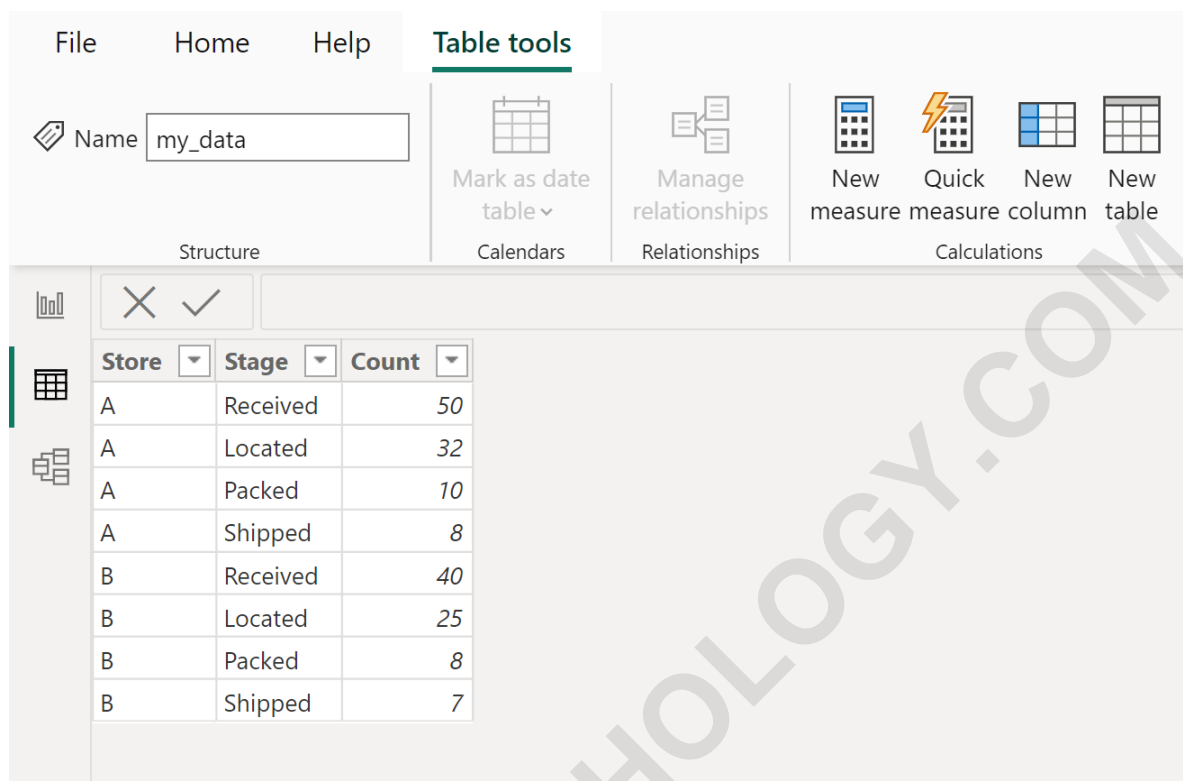
The primary objective of this step is to assign a unique, sequential number to each category in the order we wish them to appear. This numeric value is meaningless in terms of business metrics but is crucial for Power BI's sorting algorithm. For a set of categories like 'Received', 'Located', 'Packed', and 'Shipped', the corresponding numeric indices would be 1, 2, 3, and 4, respectively. This mapping ensures that when Power BI attempts to sort the text column, it relies on the defined numeric sequence instead of the alphabetical arrangement.

The structure for implementing this custom logic in Power BI involves creating at least one new calculated column. For stability and best practice, it is often recommended to create a dedicated 'Order' column (the numeric index) and sometimes a 'Copy' column of the stage name itself, ensuring that the original data column remains untouched and preserving data integrity. This structured approach, heavily relying on efficient DAX formulas, forms the foundation for advanced visual customization.

## Step 1: Defining the Sample Dataset and Visualization Goal

To illustrate this technique practically, let us work with a sample dataset named `my_data`. This table tracks the operational stages of inventory items across different retail locations. This table includes information detailing how many items have been received, located, packed, and

subsequently shipped from two distinct stores. The core categories we are concerned with are the stages themselves, which will serve as the legend items in our visualization.



The screenshot shows the 'Table tools' ribbon in Power BI. The table name is 'my\_data'. The ribbon includes the following options:

- Mark as date table (Calendars)
- Manage relationships (Relationships)
- New measure (Calculations)
- Quick measure (Calculations)
- New column (Calculations)
- New table (Calculations)

Store	Stage	Count
A	Received	50
A	Located	32
A	Packed	10
A	Shipped	8
B	Received	40
B	Located	25
B	Packed	8
B	Shipped	7

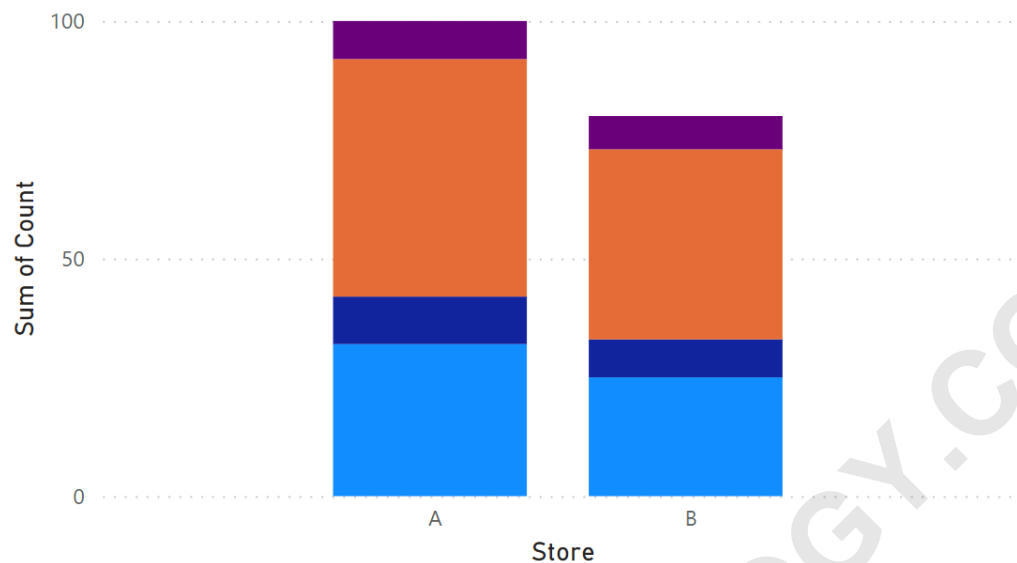
Our reporting requirement is to construct a stacked column chart that clearly visualizes the volume of items at each stage for each store. Crucially, the segments within each bar must be stacked according to the logical flow of the operation, ensuring the following custom order, which reflects the real-world process:

Received  
Located  
Packed  
Shipped

If we were to initially build the stacked column chart using the default settings, Power BI would automatically arrange the categories alphabetically. As demonstrated below, this results in an illogical visual order that starts with 'Located' and places 'Shipped' before 'Received', hindering quick analysis of the fulfillment process flow.

### Sum of Count by Store and Stage

Stage ● Located ● Packed ● Received ● Shipped

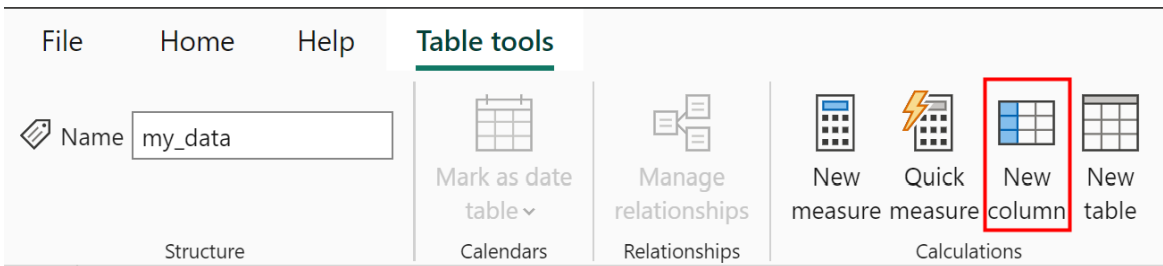


## Step 2: Implementing the Custom Sorting Logic using DAX

To enforce our desired operational sequence, we must define a new numerical column that assigns an ordinal value corresponding to each stage. This numerical column will dictate the sorting behavior of the stage names. The required mapping is:

- Received - 1
- Located - 2
- Packed - 3
- Shipped - 4

To begin this process, navigate to the **Table tools** tab within the Data View in Power BI Desktop and select the **New column** option. This action opens the formula bar where we will define the sorting logic using the DAX SWITCH function, which is ideal for conditional assignments based on specific text values.



The following DAX formula is entered into the formula bar. The function evaluates the value in the column of the `my_data` table and returns the corresponding numeric index.

**Stage Order = SWITCH('my\_data', "Received", 1, "Located", 2, "Packed", 3, "Shipped", 4)**

Executing this formula successfully creates the new numerical column named **Stage Order**, which contains the sequential numbers required for sorting the corresponding text stages. This column is the critical component for enforcing the custom sequence.

Store	Stage	Count	Stage Order
A	Received	50	1
A	Located	32	2
A	Packed	10	3
A	Shipped	8	4
B	Received	40	1
B	Located	25	2
B	Packed	8	3
B	Shipped	7	4

### Step 3: Creating a Stage Copy for Sorting Stability

Although technically we could sort the original column, a more robust practice in Power BI data modeling is to maintain a copy of the categorical column specifically for visualization purposes, especially when defining custom sort orders. This practice minimizes the risk of unintended consequences if the original column is used elsewhere or if complex relationships exist. Therefore, we will create a direct copy of the column.

To perform this duplication, navigate back to the **Table tools** tab and click the **New column** icon

again. Input the simple assignment formula below into the formula bar. This calculation merely duplicates the contents of the original column, creating an identical column for the stage names.

**Stage Copy = 'my\_data'**

The result of this operation is the creation of a second categorical column named **Stage Copy**, which mirrors the text values of the original Stage column. This column will be the field we use in our visual's legend well.

Store	Stage	Count	Stage Order	Stage Copy
A	Received	50	1	Received
A	Located	32	2	Located
A	Packed	10	3	Packed
A	Shipped	8	4	Shipped
B	Received	40	1	Received
B	Located	25	2	Located
B	Packed	8	3	Packed
B	Shipped	7	4	Shipped

#### Step 4: Applying the Custom Sort Column

The final modeling step is to instruct Power BI to use the newly created numerical index column (**Stage Order**) to sort the categorical copy column (**Stage Copy**). This linkage ensures that any visual using **Stage Copy** will adhere to the defined numerical sequence, thereby overriding the default alphabetical sorting.

With the **Stage Copy** column selected in the Data View, locate and click the **Sort by column** icon within the Column tools ribbon. From the resulting dropdown menu, select **Stage Order**. This action establishes the necessary dependency: Power BI will now sort the text values in **Stage Copy** based on the corresponding numbers in **Stage Order** (1, 2, 3, 4).

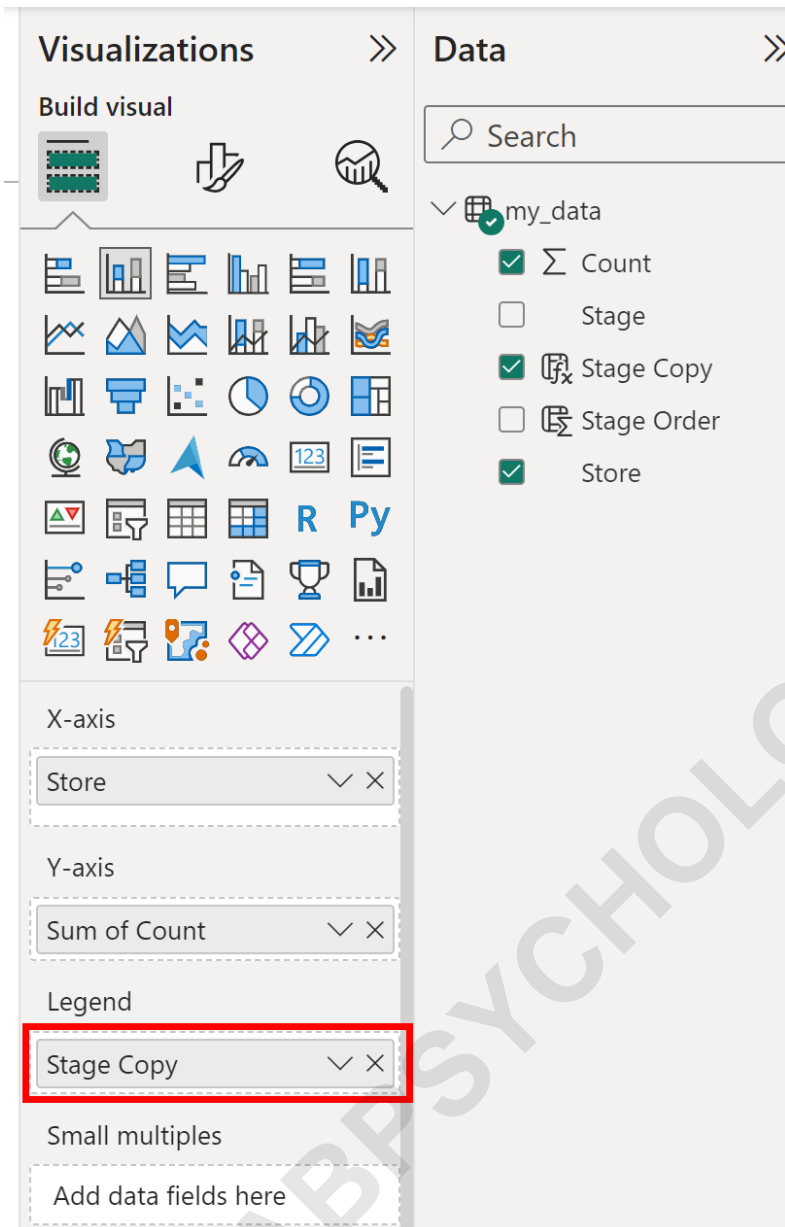
The screenshot shows the Power BI Column Tools ribbon with the 'Sort by column' dropdown menu open. The 'Sort by column' icon is highlighted with a red box. The dropdown menu is open, showing options: Stage Copy, Count, Stage, Stage Order (highlighted with a red box), and Store. Below the ribbon, a data table is visible with columns 'Order' and 'Stage Copy'.

Order	Stage Copy
1	Received
2	Located
3	Packed
4	Shipped
1	Received
2	Located
3	Packed
4	Shipped

## Step 5: Updating the Visualization and Final Review

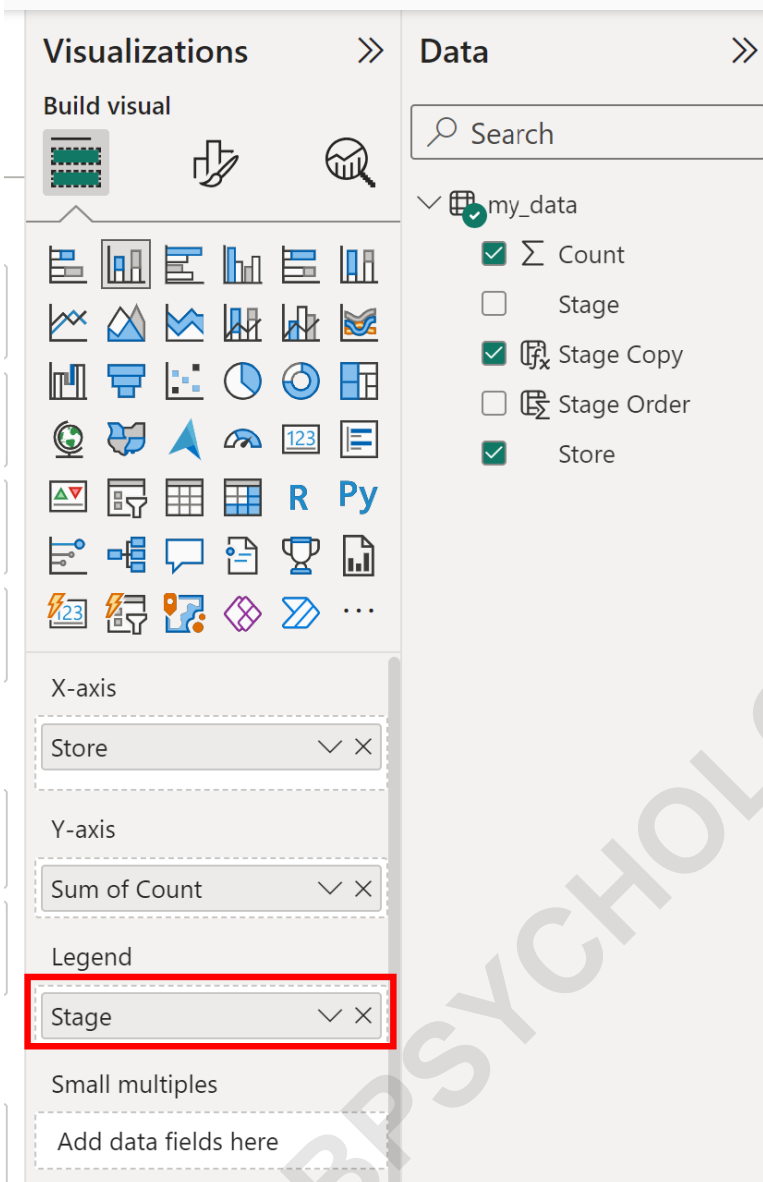
Once the custom sort relationship is established in the Data View, return to the Report View to update the visualization. Click on the existing stacked column chart to activate it. If the chart was previously using the original column in the Legend well, we must replace it with our newly engineered **Stage Copy** column.

Drag the **Stage Copy** field from the Fields pane and place it into the **Legend** well of the visualization. This substitutes the alphabetically sorted column with the custom-sorted column.



The screenshot displays the Power BI interface. On the left, the 'Visualizations' pane shows a matrix visual. The X-axis is labeled 'Store' and the Y-axis is labeled 'Sum of Count'. The Legend well contains 'Stage Copy', which is highlighted with a red box. Below the legend, there are options for 'Small multiples' and 'Add data fields here'. On the right, the 'Data' pane shows a table named 'my\_data' with the following columns: 'Count' (checked), 'Stage' (unchecked), 'Stage Copy' (checked), 'Stage Order' (unchecked), and 'Store' (checked).

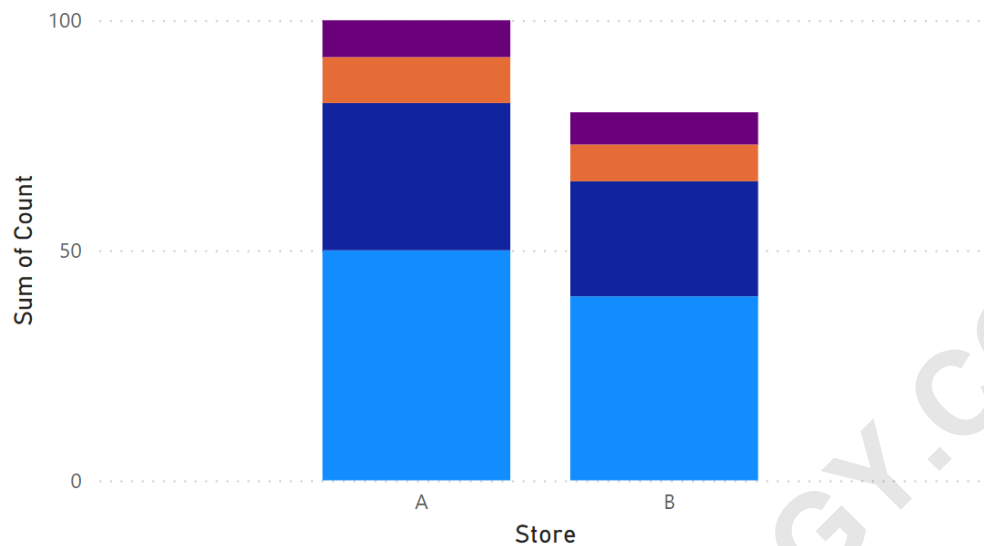
For presentation purposes, having 'Stage Copy' displayed in the legend might be distracting. To maintain a clean visual appearance without altering the underlying column name in the data model, right-click on **Stage Copy** within the Legend well and select **Rename for this visual**. Rename the field simply to **Stage**. This cosmetic adjustment ensures the visual is professional while the sorting logic remains intact.



The result is an enhanced visualization where the categories within each bar of the stacked column chart are now correctly sorted according to the operational flow we specified (Received, Located, Packed, Shipped). This precise control over the legend order significantly improves the chart's readability and aligns it perfectly with the intended Business Intelligence narrative.

Sum of Count by Store and Stage

Stage ● Received ● Located ● Packed ● Shipped



## Conclusion: Achieving Enhanced Data Storytelling

The ability to customize the sorting order of categorical fields, especially those used in the legend of complex visualizations like the stacked column chart, is a fundamental skill for advanced Power BI developers. While default settings prioritize alphabetical sorting, real-world data frequently demands ordinal or chronological sequencing to effectively communicate operational processes or defined hierarchies. By strategically employing DAX to create an auxiliary numerical index column and then linking this index to the categorical field using the "Sort by column" feature, developers gain absolute control over visual presentation.

This methodology ensures that data consumers are guided through the information in a logical and intuitive manner, minimizing cognitive load and reducing the potential for misinterpretation. Investing time in defining robust sorting mechanisms in the data model ultimately leads to higher quality, more trustworthy, and professional-grade Business Intelligence reports. Mastery of these sorting techniques is crucial for moving beyond basic visualization and into sophisticated data storytelling.

The following tutorials explain how to perform other common tasks in Power BI: