How to Change the Legend Order in Power BI Visuals

Authored by

mohammed looti

January 10, 2026

The Legend Order in Power BI refers to the specific arrangement and sequence of categorical items displayed within a visual's key, significantly impacting how the user reads and interprets the presented data. While Power BI often defaults to alphabetical sorting, there are many scenarios where a custom, logical, or business-driven order is essential for effective communication. This order is crucial, as it dictates the layering of elements, especially in visuals like the stacked column chart, where sequence matters.

Although basic reordering can sometimes be handled through the Format pane (under the Legend properties), complex or custom sequential sorting--such as process stages or predetermined hierarchies--requires a more robust solution involving the data model itself. For instance, in a visualization tracking an operational workflow, ensuring that "Received" comes before "Located," which precedes "Packed," provides clarity that alphabetical sorting ("Located," "Packed," "Received") would destroy. We will demonstrate how to achieve precise, custom sorting using DAX calculated columns.

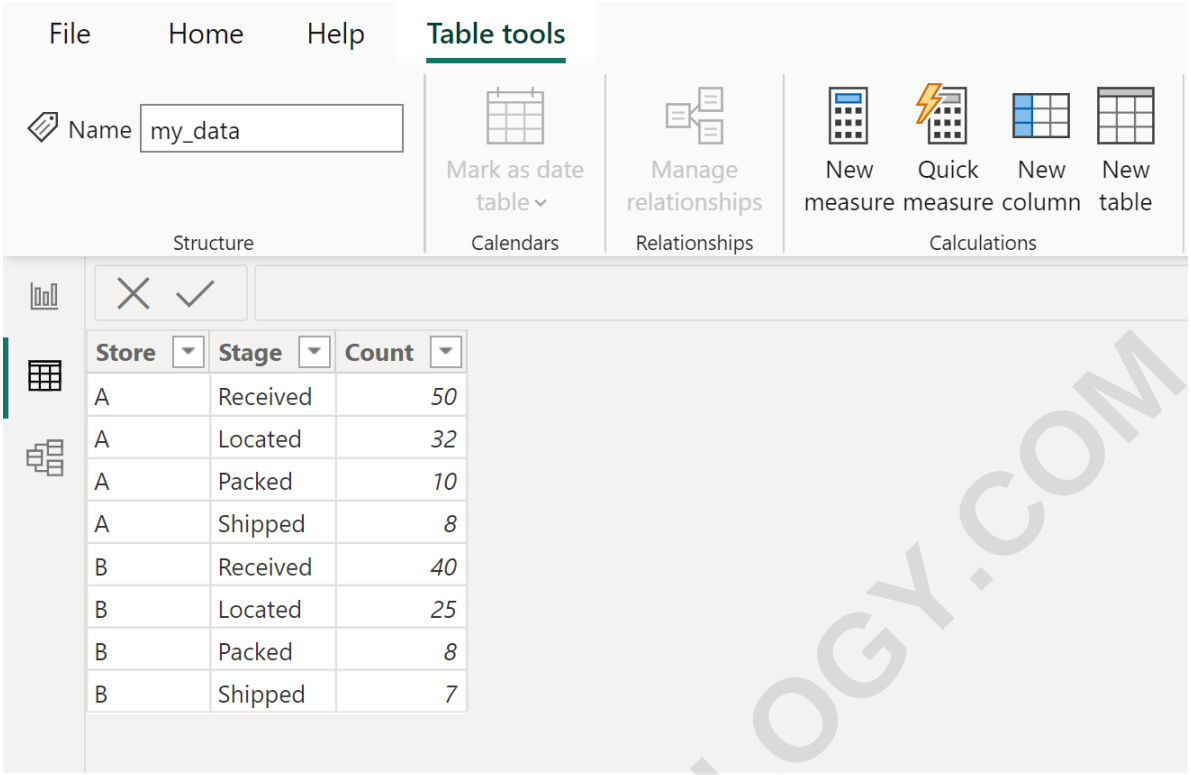
In advanced data visualization using Power BI, the ability to control the order of categories in the legend is frequently required by analysts. Standard sorting mechanisms often fail when the desired sequence is non-lexical--meaning the order is based on a defined business workflow or logical progression rather than simple alphabetical arrangement.

Fortunately, enforcing a specific custom sort order is entirely achievable. The accepted best practice involves manipulating the data model by creating a new calculated column that explicitly defines the sequence (a numeric sort key), and then instructing the original category column to use this new column for its sorting behavior. This technique ensures that the custom order persists across all visuals that utilize that category field.

The following comprehensive guide walks through a practical example, illustrating the exact steps needed to implement this robust custom sorting technique in a Power BI report, specifically addressing the challenge within a stacked column chart.

Example: Implementing Custom Legend Order in a Power BI Stacked Column Chart

Consider a common business scenario where we track inventory movement stages. Suppose we are working with a table in Power BI, named **my_data**, which contains essential metrics related to different stages of item fulfillment across two distinct retail stores. The data includes categorical stages such as Received, Located, Packed, and Shipped, alongside associated numerical values.



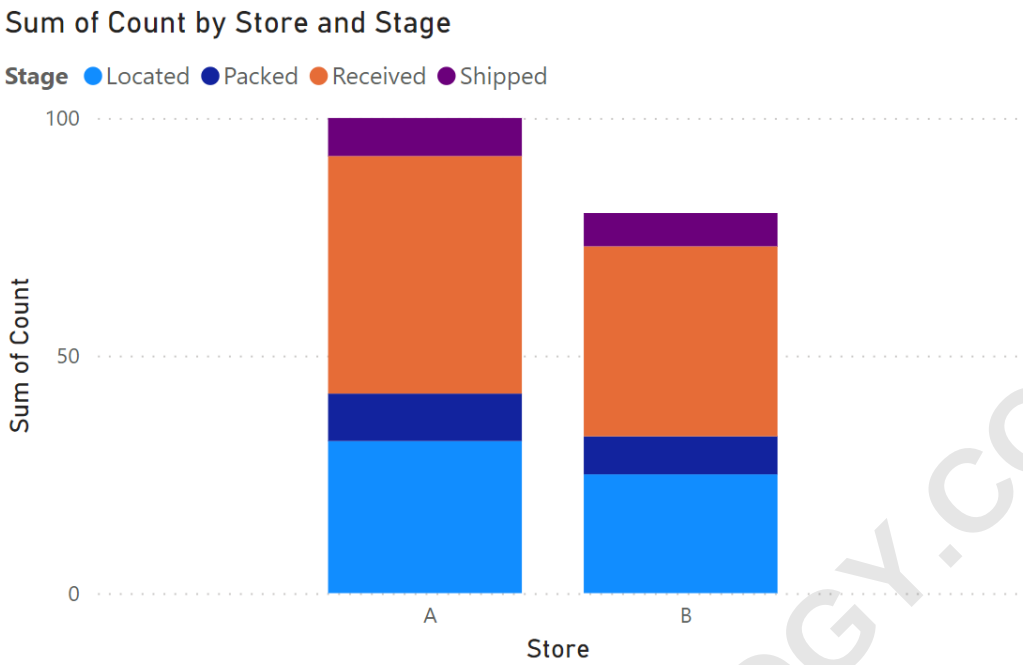
Store	Stage	Count
A	Received	50
A	Located	32
A	Packed	10
A	Shipped	8
B	Received	40
B	Located	25
B	Packed	8
B	Shipped	7

The Challenge of Default Alphabetical Sorting

Our objective is to visualize the counts of items in each stage using a stacked column chart. Critically, we need the legend items and the corresponding segments within the bars to follow a logical progression, reflecting the actual workflow: **Received**, then **Located**, then **Packed**, and finally **Shipped**.

If we initially create the stacked column chart by dropping the 'Stage' column into the Legend well, Power BI will apply its default sorting algorithm. This typically means sorting the text values alphabetically, which results in an illogical and confusing visual progression for operational data. The resulting order would likely be Located, Packed, Received, and Shipped, which misrepresents the business process flow.

Observe the resulting chart when default sorting is applied. The visual integrity of the process stages is compromised because the categorical elements are sorted alphabetically instead of sequentially:



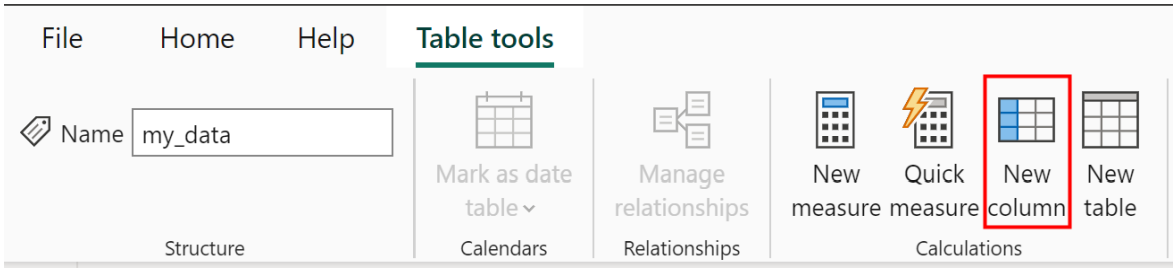
To override this default behavior and establish a custom order, we must utilize a powerful DAX mechanism that ties a numeric key to the textual category. This ensures the visualization respects the intended operational sequence:

Received (Stage 1)
Located (Stage 2)
Packed (Stage 3)
Shipped (Stage 4)

Step 1: Defining the Custom Order Key Using DAX

The first essential step is to introduce a calculated column into our **my_data** table. This column, which we will name **Stage Order**, will assign a unique, sequential numeric value to each corresponding text stage. These numeric values (1, 2, 3, 4) serve as the foundation for the custom sorting logic.

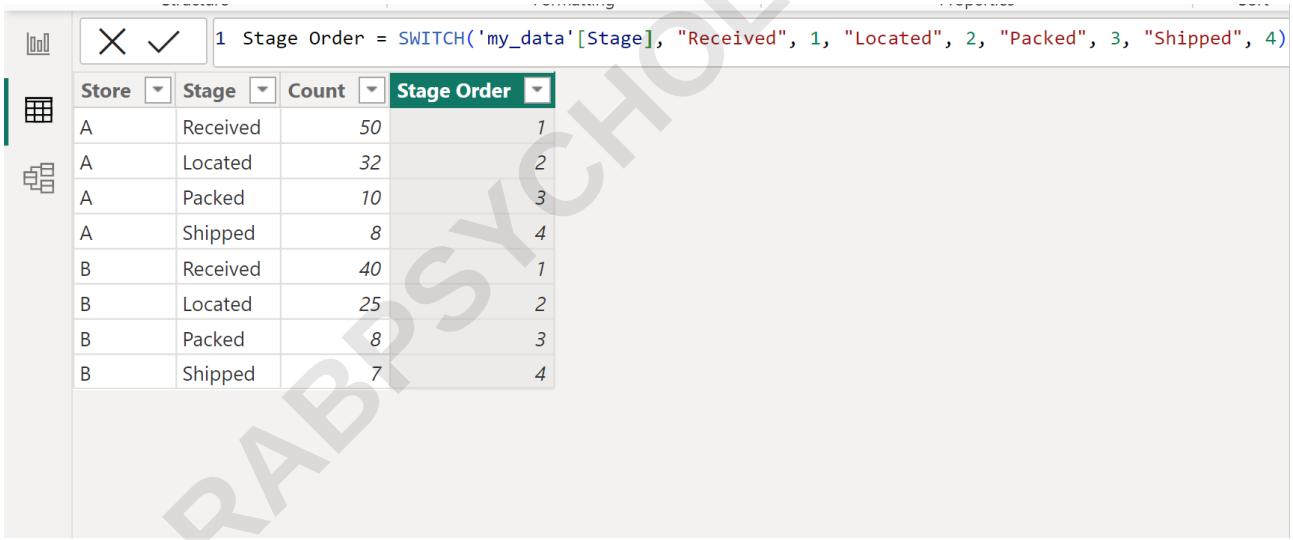
To begin, navigate to the **Table tools** tab within the Power BI desktop interface, ensuring you are in the Data view. Locate and click the **New column** icon. This action opens the formula bar, allowing us to input our Data Analysis Expressions (DAX) formula.



We will use the powerful **SWITCH** function in DAX, which is perfect for mapping text values to specific numeric outputs. Type the following formula precisely into the formula bar:

Stage Order = SWITCH('my_data', "Received", 1, "Located", 2, "Packed", 3, "Shipped", 4)

Upon confirming the formula, Power BI executes the calculation for every row in the table, resulting in the new **Stage Order** column being populated with the defined numeric sequence. This numeric column now holds the key to overriding the alphabetical sort behavior for the 'Stage' category.



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 2: Creating a Copy of the Category Column

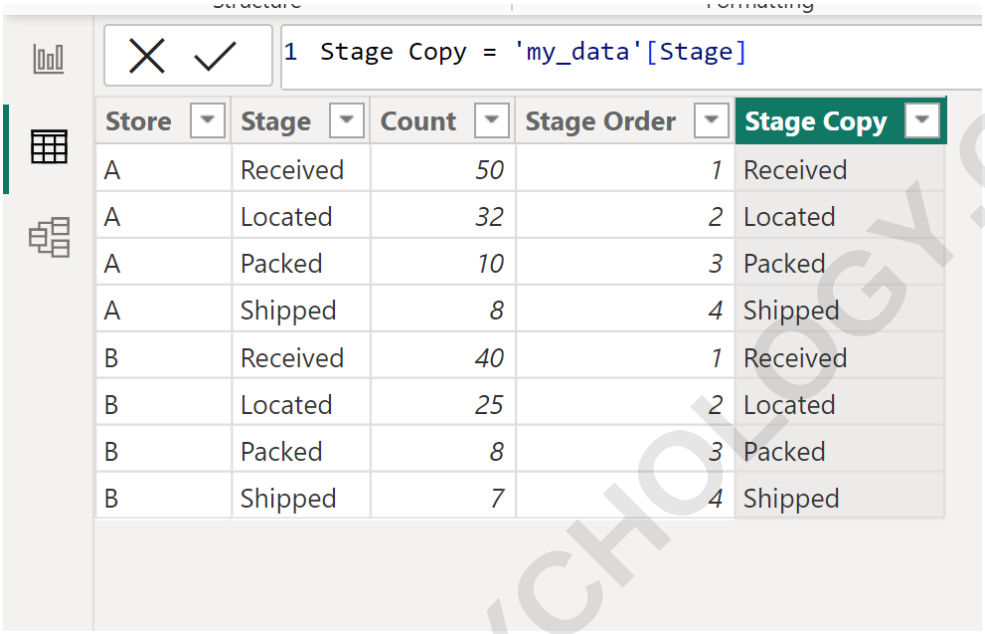
To ensure the sorting mechanism works reliably without causing circular dependencies, it is a recommended practice to create a copy of the original category column (in this case, 'Stage'). This copied column will be the one we utilize in the visual's legend well, and it is the field we will apply the custom sort rule to.

Return to the **Table tools** tab and click the **New column** icon again. We will name this column **Stage Copy**. The DAX expression for this step is straightforward, simply duplicating the values

from the original 'Stage' column:

Stage Copy = 'my_data'

This action creates the new column named **Stage Copy**, which is an exact replica of the original 'Stage' field but is independently structured to accept a custom sorting instruction linked to our numeric key.



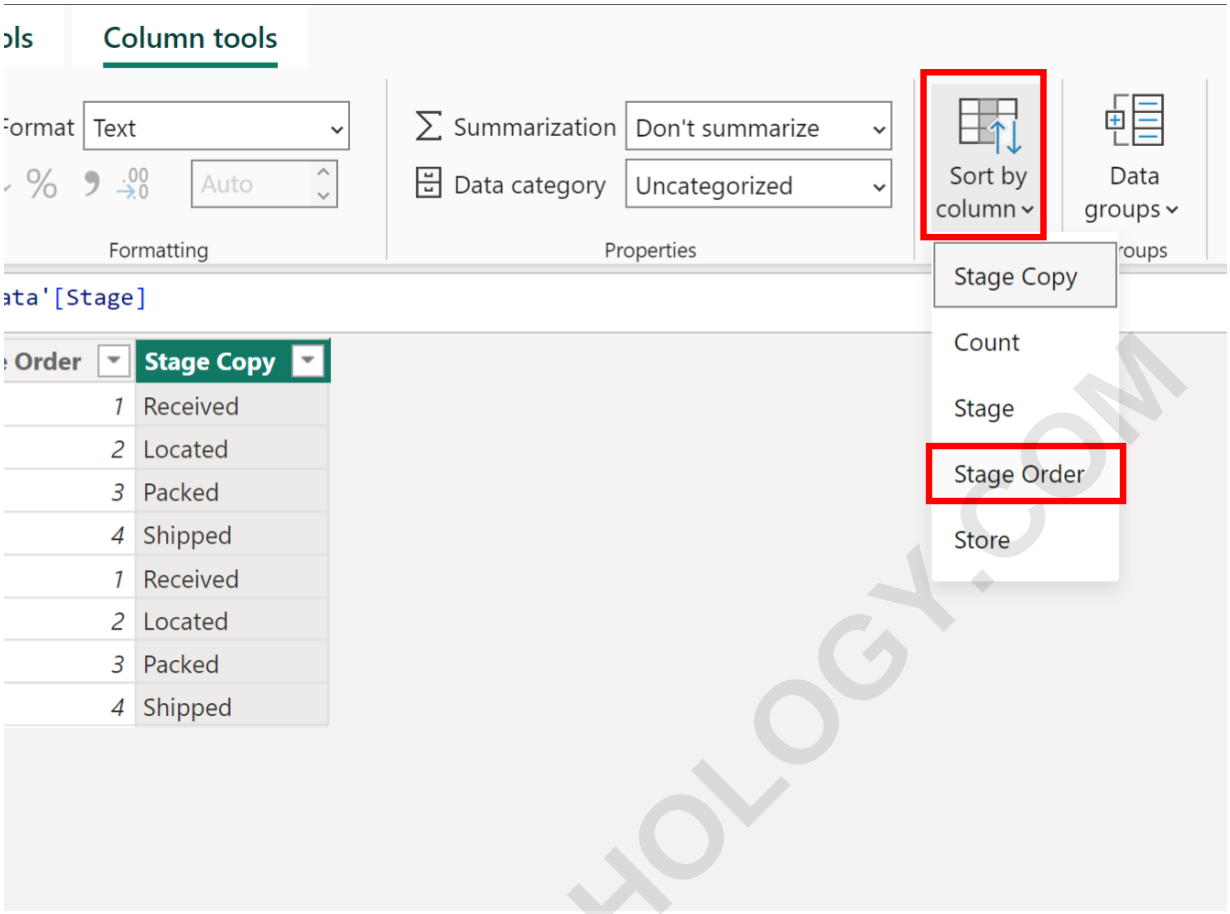
Store	Stage	Count	Stage Order	Stage Copy
A	Received	50	1	Received
A	Located	32	2	Located
A	Packed	10	3	Packed
A	Shipped	8	4	Shipped
B	Received	40	1	Received
B	Located	25	2	Located
B	Packed	8	3	Packed
B	Shipped	7	4	Shipped

Step 3: Linking the Category Copy to the Custom Sort Key

This is the critical step where we establish the relationship between the textual category we want to display and the numeric sequence we created to control its order. We must instruct Power BI that the **Stage Copy** field should no longer rely on alphabetical sorting but must instead use the values found in the **Stage Order** column.

In the Data view, ensure the **Stage Copy** column is selected. Navigate to the **Column tools** tab in the ribbon. Locate the **Sort by column** icon. Click this icon, and from the resulting dropdown menu, select **Stage Order**.

By linking **Stage Copy** to **Stage Order**, we permanently define the sorting logic for the **Stage Copy** field throughout the entire data model. Anytime this field is used in a visual, it will prioritize the numeric order 1, 2, 3, 4 over the inherent alphabetical text sorting.

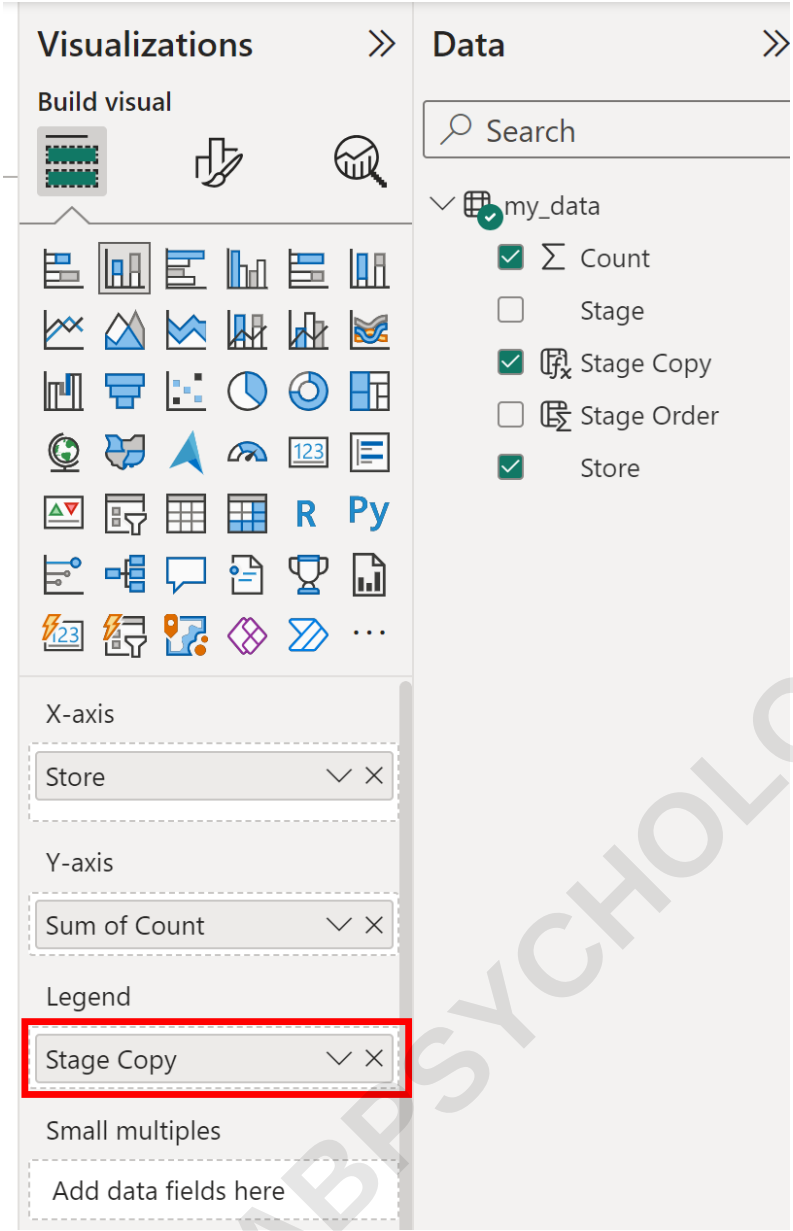


Step 4: Applying the Custom Sorted Field to the Visual

Once the sorting dependency has been established in the data model, we must switch back to the Report view to update our visualization. The key is to replace the original 'Stage' field that was using default alphabetical sorting with our newly customized field, **Stage Copy**.

Click the stacked column chart visual to make it the active element on the canvas. If the original 'Stage' field is still in the Legend well, remove it. Then, drag the **Stage Copy** field from the Fields pane and drop it into the **Legend** well of the visual.

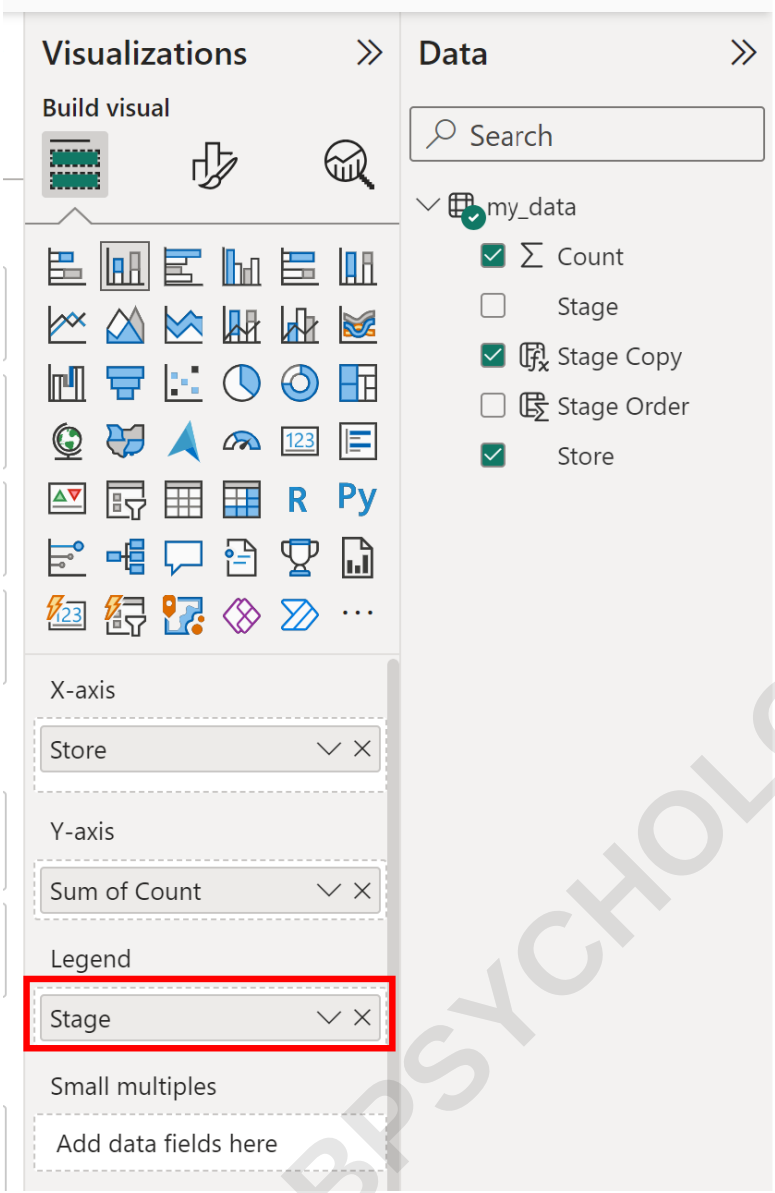
Immediately, the visual should update, displaying the categories in the order dictated by our numeric key (Received, Located, Packed, Shipped).



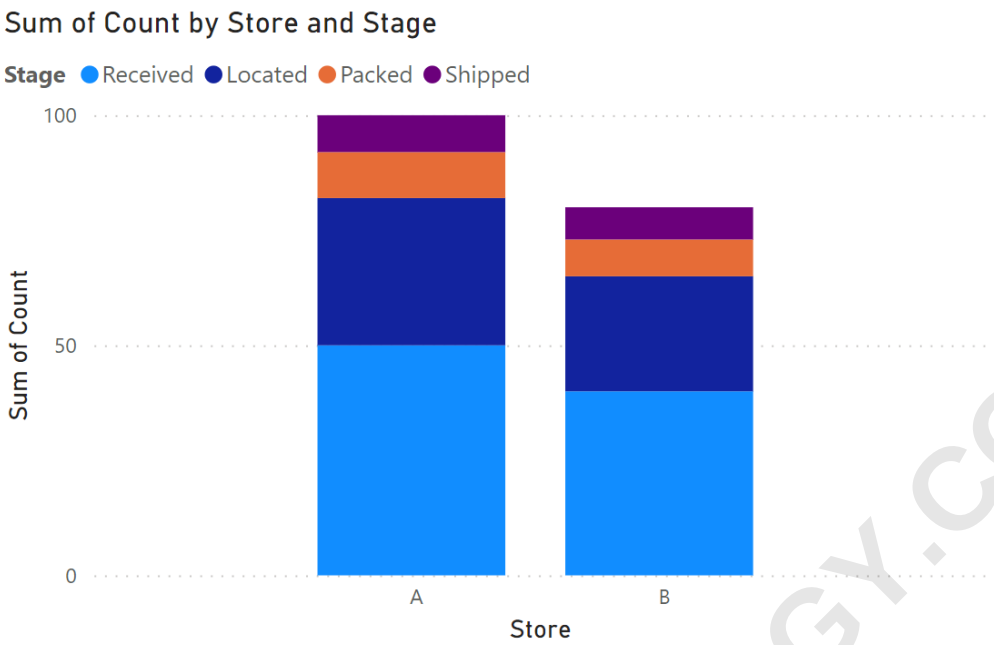
Step 5: Renaming the Legend Field for Clarity

Although the field is functionally named **Stage Copy** in the data model, displaying "Stage Copy" on the report legend itself is confusing for end-users. Power BI allows us to rename fields specifically for a single visual without altering the underlying data model name. This final step enhances report readability and professionalism.

Right-click on the **Stage Copy** field located within the **Legend** well of the visual. Select the option labeled **Rename for this visual**. Input a cleaner, user-friendly name, such as **Stage**.



The items in each bar of the stacked column chart will now be sorted in the order that we specified, resolving the issue of alphabetical defaults and ensuring the visual accurately reflects the business workflow:



Advanced Considerations and Further Power BI Tasks

The technique of using a calculated column as a sort key is considered the standard solution for enforcing custom, non-alphabetical sorting hierarchies in Power BI. This approach is superior to manual adjustments in the format pane because the custom order is defined at the data model level, meaning any future visual created using the **Stage Copy** field will automatically inherit the correct sorting.

If the categories are extremely numerous or change frequently, maintaining a DAX **SWITCH** statement can become cumbersome. In such scenarios, an alternative solution involves creating a separate, dedicated sort table. This small dimension table would contain the Category Name and the Sort Key, and then a relationship would be established between this sort table and the main data table. This external approach offers greater scalability and simplifies maintenance if the operational stages are frequently updated.

Mastering data modeling techniques such as custom sorting is vital for generating accurate and actionable insights. Understanding how fields interact, particularly through implicit and explicit sorting mechanisms, is a fundamental skill for any professional working with Power BI.

The following tutorials explain how to perform other common, yet essential, data manipulation tasks in Power BI: