

# How to Sort a Stacked Column Chart in Power BI for Clearer Insights

Authored by  
**stats writer**

January 27, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Sort a Stacked Column Chart in Power BI for Clearer Insights*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127945>

**Power BI** serves as a cornerstone **business intelligence** platform, empowering professionals to transform raw data into dynamic, insightful visualizations. Among the array of available visual elements, the **stacked column chart** stands out as an indispensable tool for comparing proportional data contributions across various categorical segments. While these charts inherently offer compelling comparative analysis, their true analytical power is often unlocked by applying a strategic sorting mechanism. Standard sorting options in Power BI typically default to ordering by the category axis (X-axis) or by the value of the first series (Y-axis), which may obscure the overall magnitude of the categories being compared. For enhanced data interpretation and to highlight the most significant contributors, users frequently need to sort the columns based on their aggregate total--the overall height of the stacked bar. This nuanced sorting capability is crucial for effective visual communication.

Achieving this specific sorting behavior for a **stacked column chart** requires a methodological approach that extends beyond the standard sorting parameters found in the visualization pane. The conventional "Sort By Column" feature is generally effective for sorting discrete values or simple categories. However, when dealing with stacks where the total height is a composite of multiple data series, the native sorting functions often prove insufficient for achieving the desired outcome. Therefore, we must engineer a solution that calculates this cumulative value before applying it as the designated sorting criterion. This comprehensive process involves leveraging **DAX** (Data Analysis Expressions) to create a custom computational element, formally referred to as a **Measure**, which precisely represents the total height of each stack within the visualization. This technique provides users with a flexible and highly efficient method to organize and analyze complex datasets within Power BI.

## Sorting Stacked Column Charts in Power BI: A Comprehensive Guide

### The Fundamental Challenge: Sorting by Total Stack Height

In advanced data analysis, the primary requirement is often to sort a **stacked column chart** in **Power BI** according to the cumulative or total height of the individual bars, rather than just the first series value. This allows the user to immediately identify which category contributes the most overall value, regardless of how that value is distributed across the segments.

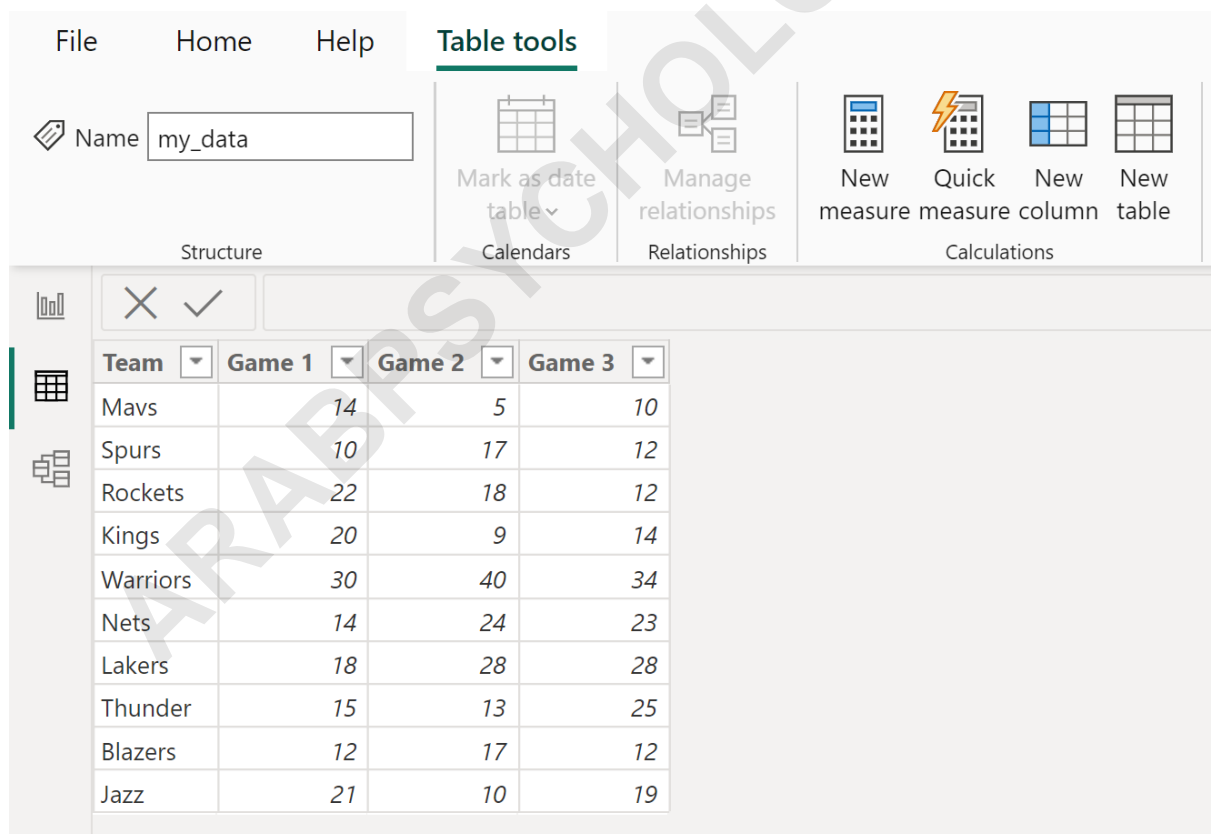
When a visualization is initially created, Power BI applies default sorting rules. For a stacked column chart, this typically means sorting either alphabetically by the X-axis category or numerically by the value of the first element placed on the Y-axis. While these defaults are functional, they fail to provide a holistic view when the aggregation of all segments within a bar is the most critical metric for sorting. For example, if you are tracking sales across regions and

products, sorting by total sales (the full bar height) is far more informative than sorting by sales of only Product A.

To successfully override this default behavior and sort by the total height of the stacked bars, we must first calculate that total height explicitly for every category present on the X-axis. This necessitates the creation of a specialized computational element--a **Measure**--that aggregates the values of all stacked series. Once this measure is established, Power BI can utilize it as the basis for sorting the visual elements, providing a clear and efficient prioritization of data points.

## Practical Example: Preparing the Data and Defining the Goal

To illustrate the process effectively, consider a dataset within Power BI named **my\_data**. This table contains crucial performance metrics, specifically the points scored by various basketball players across three distinct games. Our objective is to visually compare the players' performance using a stacked column chart, but the ultimate goal is to order the players based on their overall scoring totals accumulated across all three games.



The screenshot shows the Power BI interface with the 'Table tools' ribbon active. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a data table is displayed with the following data:

Team	Game 1	Game 2	Game 3
Mavs	14	5	10
Spurs	10	17	12
Rockets	22	18	12
Kings	20	9	14
Warriors	30	40	34
Nets	14	24	23
Lakers	18	28	28
Thunder	15	13	25
Blazers	12	17	12
Jazz	21	10	19

As shown in the data table above, each player (represented by the 'Team' category, which functions as our categorical axis) has three distinct values corresponding to 'Game 1', 'Game 2', and 'Game 3'. When visualized, these three columns will form the individual segments of the

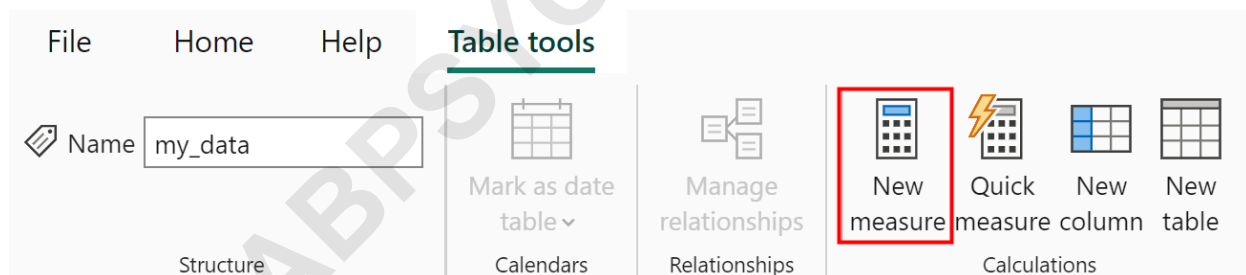
stacked column. If we attempt to create the chart without a preparatory measure, the sorting options available will not allow us to naturally sort by the sum of 'Game 1' + 'Game 2' + 'Game 3' for each player.

The primary motivation here is clarity and actionable insight. If Player A scored 50 total points and Player B scored 80 total points, the chart should visually reflect this hierarchy by placing Player B's bar higher than Player A's, regardless of how the points were distributed across the individual games. This requirement dictates that the summation of points must be calculated outside the visualization's default aggregation behavior, making the creation of a dedicated **Measure** mandatory.

## Creating the Aggregate Measure using DAX

If we wish to enable the sorting of the bars based on the cumulative height--meaning the total points scored by each player across all games--we must first introduce a new computational column that explicitly calculates this aggregated value. This calculation will serve as our custom sorting criterion, allowing us to accurately prioritize players based on their overall performance.

To initiate the creation of this aggregate value using **DAX**, navigate to the **Table tools** tab within the Power BI interface. Once there, locate and click the **New measure** icon, which prepares the system for defining a custom calculation.



The formula for calculating the sum of points must account for all three game columns and ensure the calculation is performed row-by-row before aggregation. The following **DAX** expression is used to define the new measure, which we will name **Sum Points**:

**Sum Points = CALCULATE(SUMX('my\_data', + + ))**

This formula utilizes the **SUMX** iterator function, which iterates over every row in the **my\_data** table and calculates the sum of 'Game 1', 'Game 2', and 'Game 3' for that specific row (player). The outer **CALCULATE** function ensures that this sum is evaluated correctly within the filter context of the visualization. Upon successful creation, a new **Measure** named **Sum Points** will appear in

your Fields pane, containing the crucial aggregate score for each player across all three games.

The screenshot shows the Power BI ribbon with the 'Measure tools' tab selected. The 'Name' field is set to 'Sum Points', the 'Home table' is 'my\_data', and the 'Format' is 'Whole number'. The 'Data category' is 'Uncategorized'. Below the ribbon, the DAX formula for the measure is displayed: `1 Sum Points = CALCULATE(SUMX('my_data', [Game 1] + [Game 2] + [Game 3]))`. A data table is shown below the formula, listing teams and their scores across three games.

Team	Game 1	Game 2	Game 3
Mavs	14	5	10
Spurs	10	17	12
Rockets	22	18	12
Kings	20	9	14
Warriors	30	40	34
Nets	14	24	23
Lakers	18	28	28
Thunder	15	13	25
Blazers	12	17	12
Jazz	21	10	19

## Building the Stacked Column Chart Visual

Once the custom sorting **Measure** (Sum Points) has been successfully defined and added to the data model, the next phase involves switching to the Report View and constructing the visualization itself. This process ensures all necessary fields are correctly mapped to generate a functional stacked column chart ready for the custom sorting procedure.

Begin by locating the **Visualizations** pane and clicking the icon designated for the **Stacked column chart**. With the visual active on the canvas, proceed to assign the following data fields from the **my\_data** table and the newly created measure to the corresponding visual wells:

Drag the **Team** field to the **X-axis** well, defining the categories that will form the individual bars.

Drag **Game 1**, **Game 2**, and **Game 3** fields to the **Y-axis** well. These three fields will stack vertically to form the height of each column.

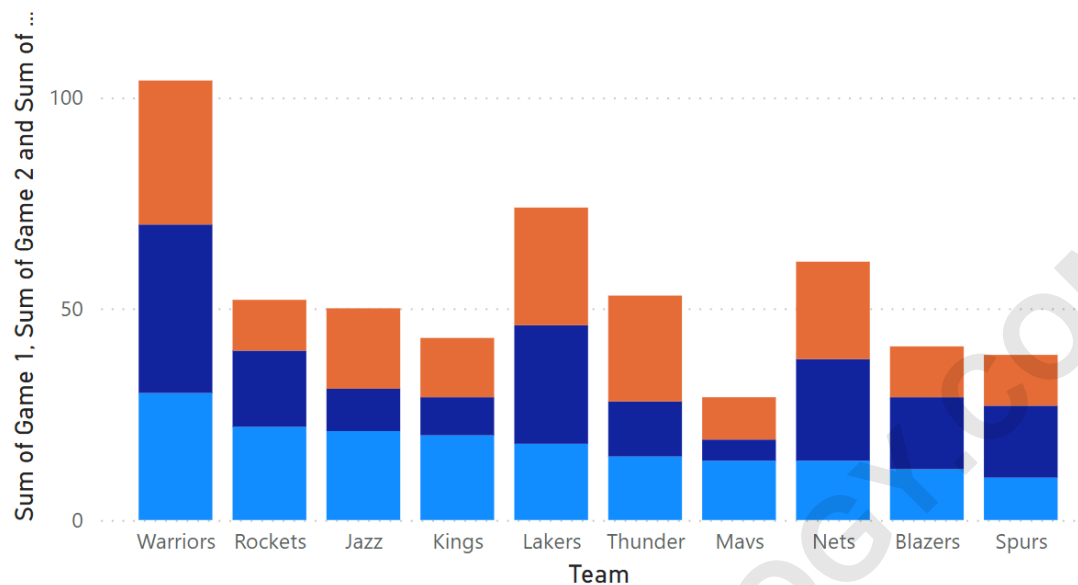
Crucially, drag the newly created **Measure**, **Sum Points**, to the **Tooltips** well. Placing the sorting measure in the Tooltips well is essential because it includes the measure in the visual's data context without directly affecting the visual stacking, thus making it available as a sorting option.

The image shows the Power BI interface with the Visualizations pane on the left and the Data pane on the right. The Visualizations pane is set to 'Stacked Column Chart'. The Data pane shows a table named 'my\_data' with the following fields: Game 1, Game 2, Game 3, Sum Points, and Team. The X-axis is set to 'Team' and the Y-axis is set to 'Sum of Game 1', 'Sum of Game 2', and 'Sum of Game 3'. The Legend, Small multiples, and Tooltips sections are also visible.

Following the field assignments, the initial stacked column chart will render on the report canvas. At this stage, the chart will display the total points scored by each player, segmented by game. However, the initial sort order will likely be based on a default value, often the first series on the Y-axis (Game 1), which may not reflect the true overall ranking of the players.

### Sum of Game 1, Sum of Game 2, Sum of Game 3 and Sum Points by Team

● Sum of Game 1 ● Sum of Game 2 ● Sum of Game 3

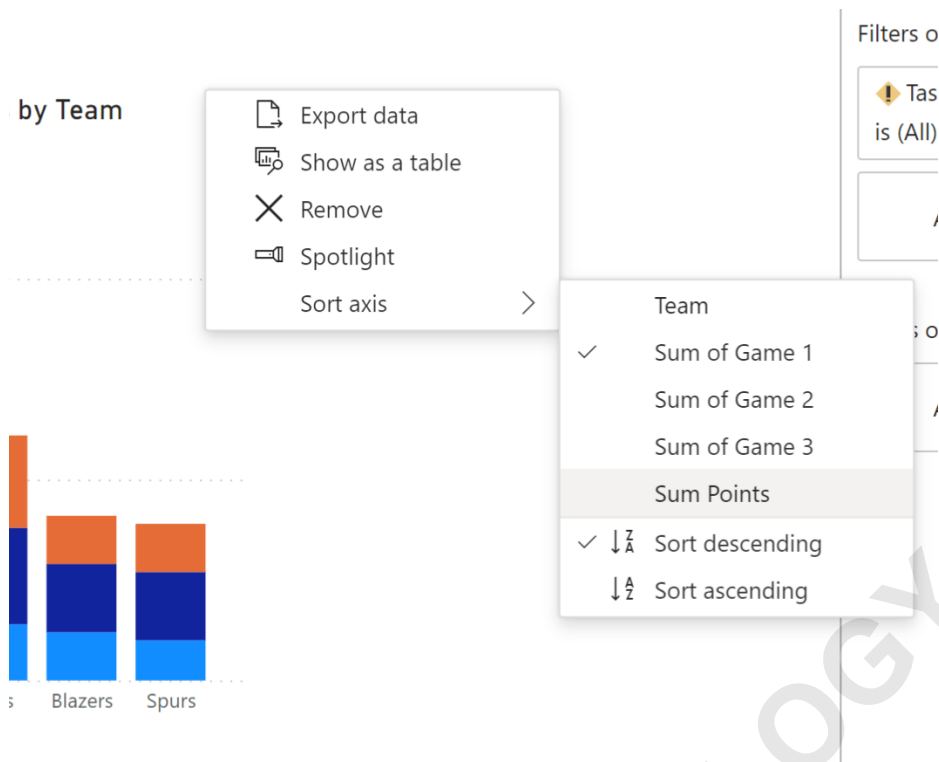


### Applying the Custom Sort Axis

As demonstrated in the visualization above, the bars are currently sorted in descending order based on the values in the **Game 1** column, meaning the player with the highest score in Game 1 appears first. Our objective is to disregard this initial sort order and instead prioritize the bars based on the cumulative value defined by our custom **Sum Points** measure.

To apply the custom sort criterion, interact directly with the visualization. Click the three horizontal ellipsis (dots) located in the top right corner of the chart, which opens the contextual menu for the visual. Within this menu, select the option labeled **Sort axis**.

A list of available fields will populate, including all fields currently used in the visual, crucially including the **Sum Points** measure placed in the Tooltips well. Click **Sum Points** from this list. This action instructs **Power BI** to use the aggregate total, rather than the discrete game scores, as the reference for ordering the categories along the X-axis.

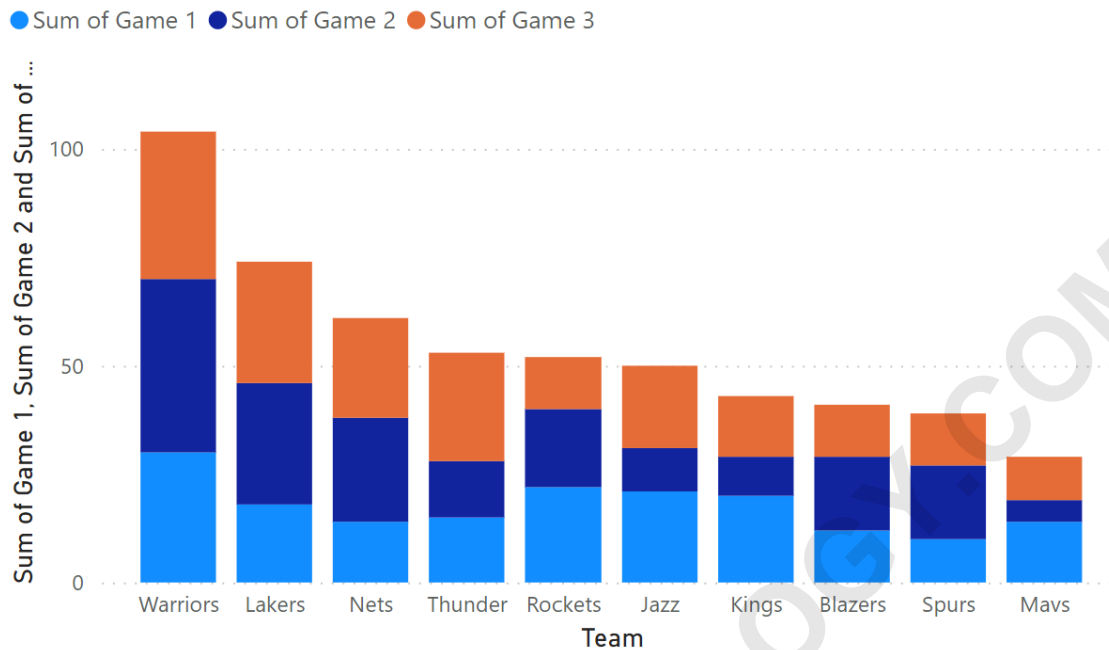


If the resulting sort direction is not as desired (e.g., you prefer ascending instead of the default descending), simply return to the **Sort axis** menu and click **Sort descending** or **Sort ascending** to toggle the order. By selecting the **Sum Points** measure and setting the desired direction, the chart is immediately reorganized to reflect the true total magnitude of each stacked column.

## Verifying the Results and Enhancing Interpretation

Upon successful execution of the custom sorting steps, the visualization will automatically refresh. The bars in the **stacked column chart** will now be correctly positioned based on the total points scored by each player, organized in descending order by default. This transformation significantly enhances the chart's readability and analytical value, as the most successful players are now instantly identifiable.

Sum of Game 1, Sum of Game 2, Sum of Game 3 and Sum Points by Team



This final, sorted visual provides a clear hierarchy based on cumulative performance. Not only can the reader ascertain the overall ranking (from highest to lowest total score), but they can also simultaneously analyze the compositional breakdown of that total score (the proportion contributed by Game 1, Game 2, and Game 3). This synthesis of aggregate comparison and granular detail is precisely what makes custom sorting in **Power BI** an invaluable technique for powerful **business intelligence** reporting.

## Concluding Insights on Advanced Sorting

The ability to sort a complex visualization like a stacked column chart based on an aggregate total, rather than an arbitrary segment, is a cornerstone of sophisticated data visualization. By leveraging **DAX** to create a dedicated computational **Measure**, users gain granular control over how their data narratives are presented. This methodology ensures that the visualization prioritizes the most meaningful metric--the cumulative sum--thereby guiding the audience toward the most relevant conclusions immediately.

While the initial process may seem slightly complex due to the requirement of creating a custom measure, the resulting clarity and analytical precision it provides are well worth the investment. This technique is universally applicable across various datasets where stacked charts are employed, offering a robust solution for enhancing report quality and ensuring accurate data interpretation within the Power BI environment.

## Further Power BI Resources

The following tutorials explain how to perform other common and advanced analytical tasks within the **Power BI** ecosystem, helping users further unlock the potential of their data models and visualizations:

Detailed guides on advanced **DAX** functions.

Instructions for implementing dynamic filtering and conditional formatting.

Strategies for optimizing data models for improved report performance.

ARABPSYCHOLOGY.COM