

Learning to Sort Stacked Column Charts by Total Value in Power BI

Authored by
Mohammed looti

November 12, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learning to Sort Stacked Column Charts by Total Value in Power BI*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=17708>

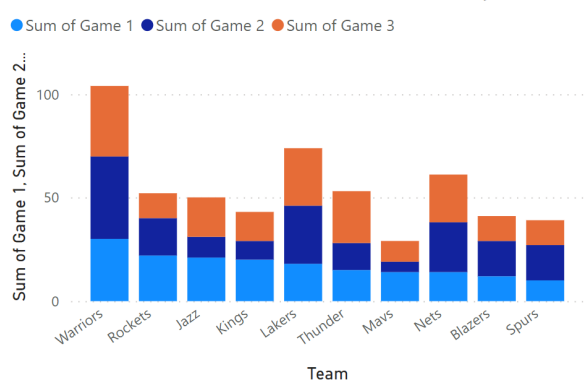
The Challenge of Sorting Stacked Column Charts

When designing visualizations in [Power BI](#), report creators frequently encounter the need to customize the sorting logic, particularly for elements like the [stacked column chart](#). By default, Power BI may sort these charts alphabetically based on the category on the X-axis, or it might sort based on the value of the first segment within the stack. Neither of these default methods typically provides the most insightful visual representation, especially when the goal is to quickly compare the overall magnitude of different categories.

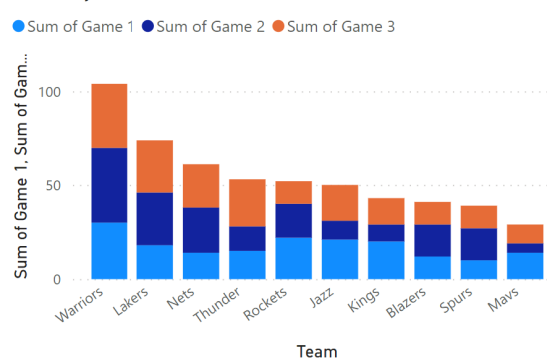
The most common requirement for optimizing readability in a stacked column chart is to sort the bars based on their combined or **total height**. This technique immediately highlights the categories that contribute the most or the least to the overall dataset magnitude, allowing stakeholders to grasp key performance indicators or distributional differences instantly. Achieving this requires a specific approach within the Power BI environment: creating a dedicated measure that calculates the [summation](#) of all components within the stack, ensuring that the visual ranking reflects the true aggregate value.

The visual below illustrates the desired outcome: a stacked column chart where the bars are ordered from largest to smallest based on the aggregated value of all components within that column. This example demonstrates how a simple, calculated measure can unlock powerful sorting functionality, transforming a standard visualization into an effective analytical tool capable of answering critical ranking questions efficiently.

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



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



This tutorial provides a practical, step-by-step guide on implementing this total-based sorting mechanism, focusing on the essential data modeling and measure creation techniques required in [Power BI](#) Desktop.

Prerequisites and Data Preparation in Power BI

Before we can apply custom sorting logic, we must ensure our data is properly imported and structured within the [Power BI data model](#). For this detailed example, we will use a hypothetical dataset named **my_data**. This table tracks points scored by various basketball players across three distinct games. Each row represents a player or team, and the columns contain the respective points scored in Game 1, Game 2, and Game 3.

This initial setup is crucial because the visual we intend to create--the stacked column chart--will rely on aggregating these individual game scores (Game 1, Game 2, and Game 3) along the Y-axis, while the player or team name defines the X-axis categories. While this data structure is suitable for immediate visualization, it inherently lacks a single, explicit column representing the total score, which is necessary for effective sorting based on the complete bar height.

Review the structure of our sample data, **my_data**, as displayed below. Although calculating the sum externally (e.g., in the source file) is possible, defining the sum as a dynamic measure within [Power BI](#) using Data Analysis Expressions (DAX) is the recommended best practice. This approach ensures the total remains responsive to potential filters or slicers applied elsewhere in the report, maintaining both data integrity and report flexibility.

File

Home

Help

Table tools

Name

my_data

Mark as date table

Calendars

Manage relationships

Relationships

New measure

Quick measure

New column

New table

Structure

Calculations

×

✓

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

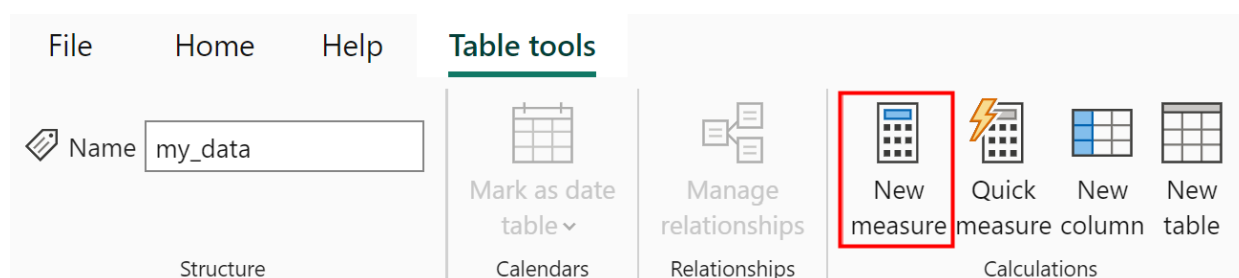
19

Our primary objective here is to calculate the sum of the points scored in Game 1, Game 2, and Game 3 for each player, and subsequently use this newly calculated total as the underlying metric that dictates the order of the columns in the resulting chart.

The Core Solution: Defining a Calculation Measure (DAX)

To enable sorting by the overall column height--the aggregated total of all stacked segments--we must introduce a new measure using [DAX](#) (Data Analysis Expressions). This measure will calculate the aggregated points for each player across all three games, regardless of how those game scores are individually segmented in the visual. This calculated field, which we will name **Sum Points**, will serve as the essential, invisible sorting key for our visualization.

The process of creating a new measure begins in the Power BI Desktop interface. Navigate to the **Table tools** tab within the ribbon, then click the **New measure** icon. This action opens the [DAX](#) formula bar, allowing us to define our calculation. It is important to remember that measures are dynamic and calculated at query time, making them far superior to calculated columns for aggregated metrics intended for visualization and sorting.



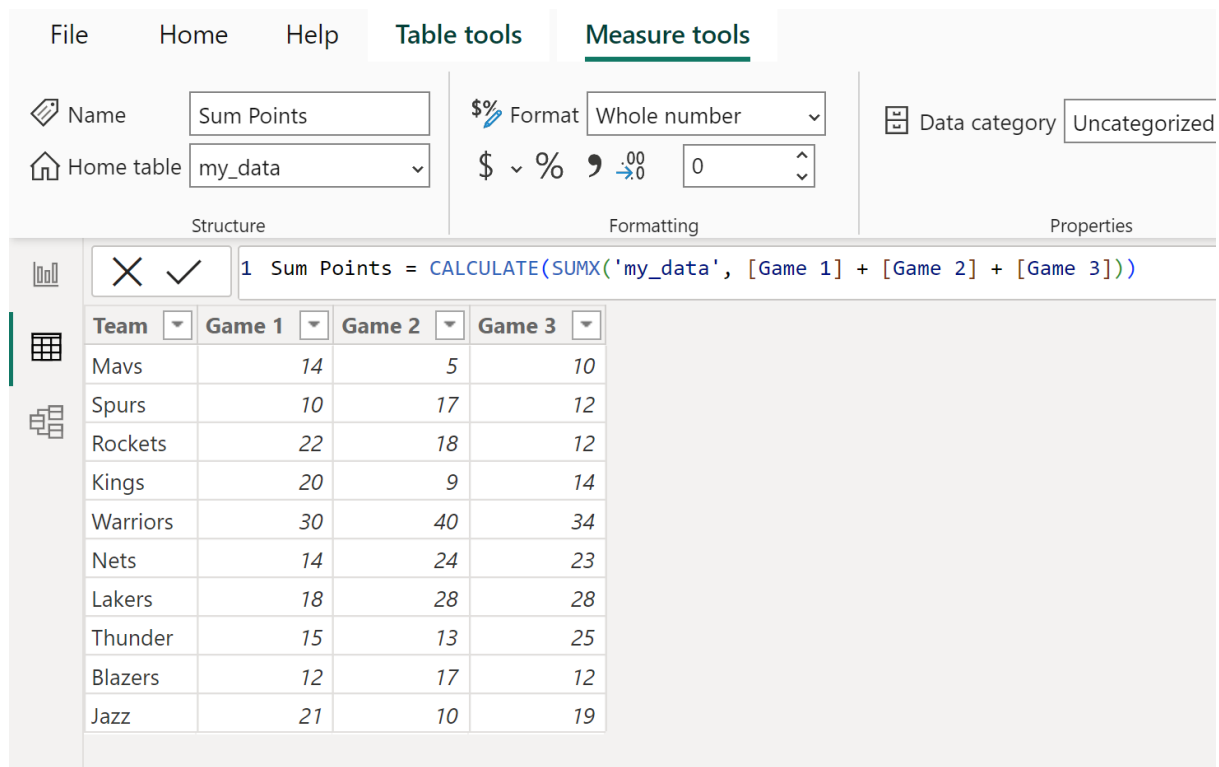
The following formula utilizes two powerful DAX functions: [CALCULATE](#) and [SUMX](#). The [SUMX](#) function is an iterator that steps through the rows of the specified table ('**my_data**') and performs the summation of the three game columns for each row. The [CALCULATE](#) function, while often used for context modification, here ensures the measure correctly aggregates the total sum without being unduly constrained by external filters that might target only specific games.

Input the calculation below exactly into the formula bar. This measure aggregates the total points and is now ready to be deployed as a sorting mechanism within the visual:

Sum Points = CALCULATE(SUMX('my_data', + +))

Once confirmed, the new measure, **Sum Points**, is created and automatically added to the fields list in your data model. Although it will not appear as a visible column in the data view, it holds the essential total aggregated score for each player, forming the basis for our subsequent sorting

operation.



The screenshot shows the Power BI Desktop interface with the 'Measure tools' tab selected. The 'Name' field is 'Sum Points', 'Home table' is 'my_data', 'Format' is 'Whole number', and 'Data category' is 'Uncategorized'. The formula bar shows: 1 Sum Points = CALCULATE(SUMX('my_data', [Game 1] + [Game 2] + [Game 3])). Below the formula bar is a table with columns: Team, Game 1, Game 2, Game 3.

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

Visualizing the Data: Creating the Stacked Column Chart

With the **Sum Points** measure successfully defined and residing in our data model, the next step involves constructing the primary visualization in the Report View. Switch to the Report View interface and select the **Stacked column chart** icon located under the **Visualizations** tab. This action initializes a blank canvas for the visual, which we will now populate with our relevant data fields.

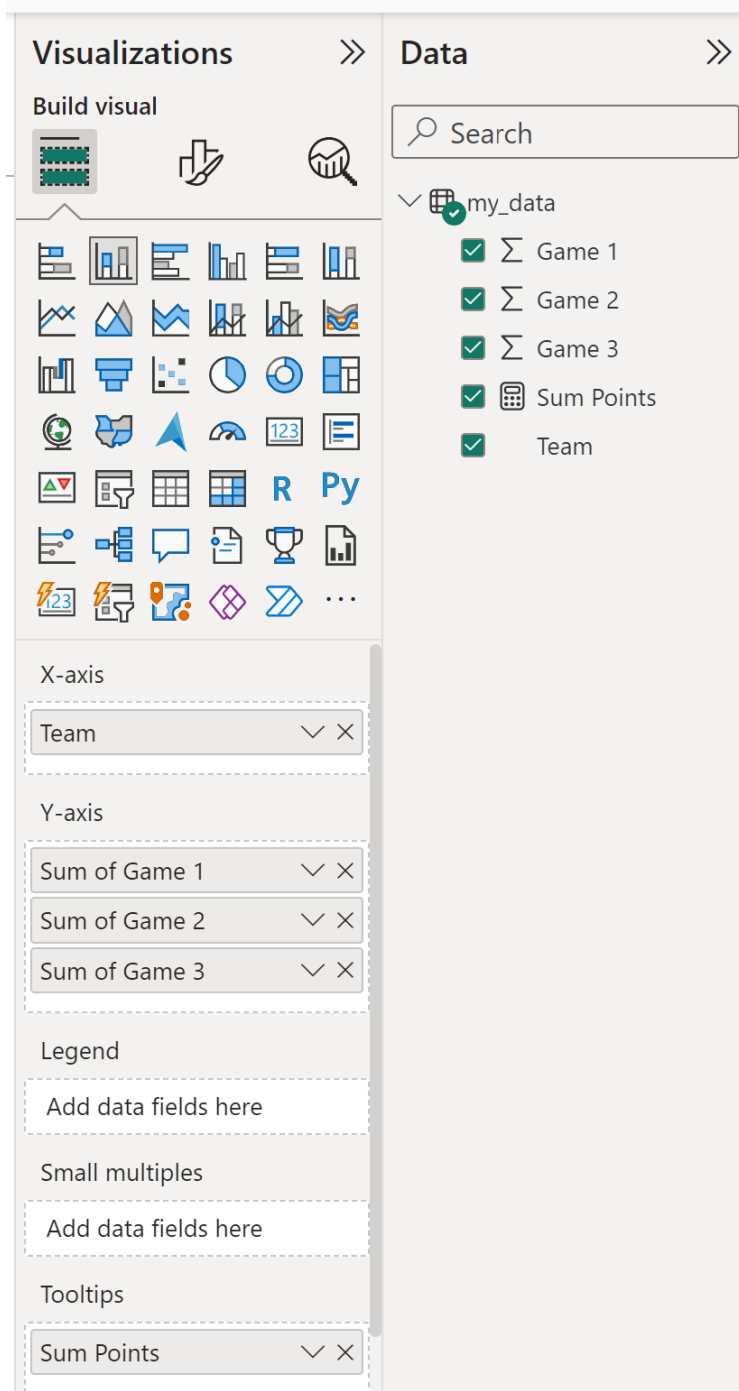
Proper configuration of the visual fields is essential to achieve the desired stacking effect. The categorical variable (the entity we wish to compare and sort) must be assigned to the X-axis, while the individual metrics that form the stack should be assigned to the Y-axis. In our specific scenario, the data fields are mapped as follows:

Assign the **Team** field to the **X-axis** (the category being compared).

Assign **Game 1**, **Game 2**, and **Game 3** fields to the **Y-axis** (the values that make up the stacked segments).

Crucially, the new measure, **Sum Points**, must be included in the visualization's data context, but not necessarily on the visible axes. The ideal and most efficient location for a measure intended

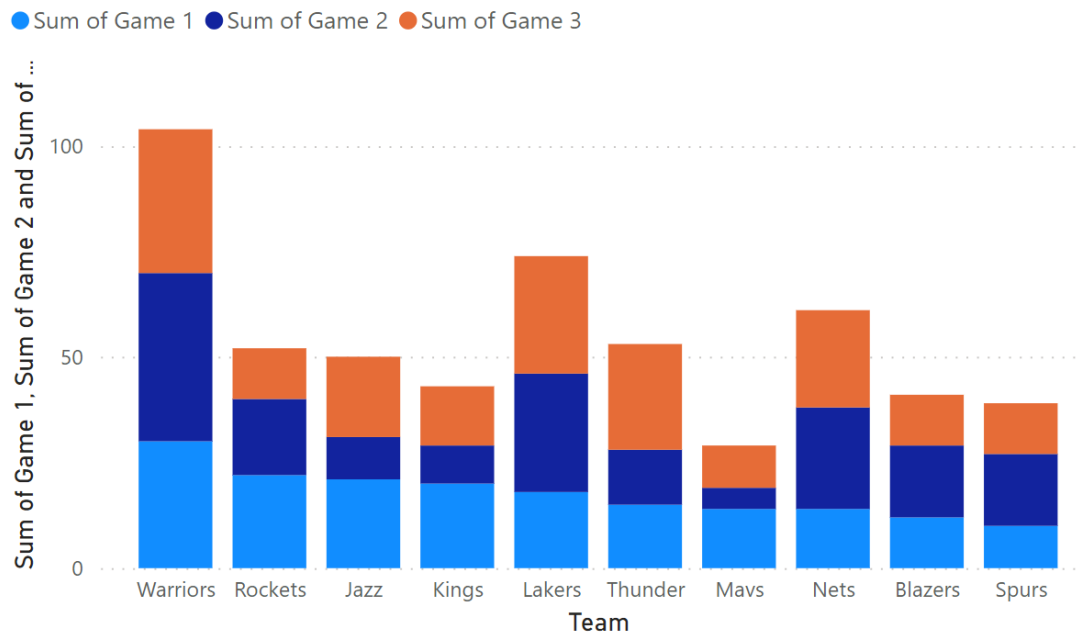
purely for sorting or informational purposes is the **Tooltips** well. Adding **Sum Points** to the Tooltips ensures that the measure is present within the visual's calculation context, making it readily available for the sorting function without cluttering the main axes.



Upon configuring the fields as described, Power BI renders the initial [stacked column chart](#). At this stage, the chart displays the points stacked correctly, but the order of the columns (the X-axis categories) is likely determined by a default setting--often an alphabetical sort or, in this case,

descending order based on the value of the first segment, **Game 1**. This initial arrangement, as shown below, does not reflect the total aggregated performance, making it difficult to immediately identify the top-performing teams based on overall points.

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



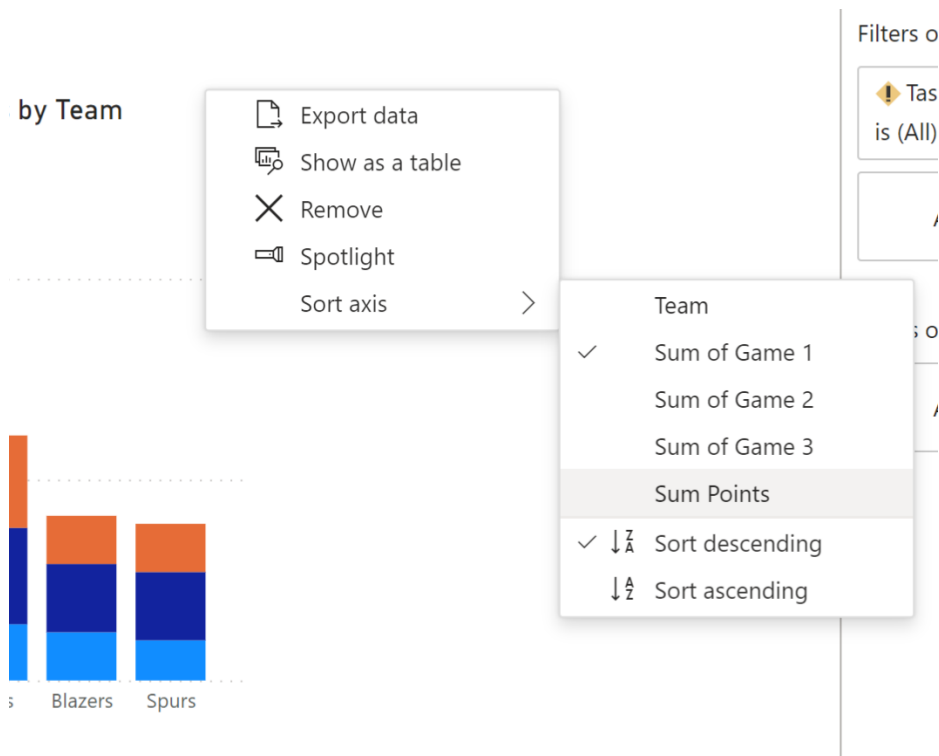
Implementing the Sort Axis by Measure

The final and most critical step is instructing the visualization to use our custom measure, **Sum Points**, as the definitive basis for ordering the columns. This is achieved through the chart's advanced sorting options, which are accessible directly from the visual interface in the Report View. This technique overrides the default sorting method that previously ordered the columns based on a single variable segment, such as **Game 1** or **Team** name.

To change the sorting logic, locate the ellipsis (three horizontal dots) in the top-right corner of the stacked column chart visualization. Click this menu to reveal the contextual options available for the visual. Within this menu, select the **Sort axis** option. Power BI will then present a list of all available fields and measures that can be used for sorting the X-axis categories. Crucially, this list includes the original categorical field (Team) and any measures or fields added to the visualization's wells, including the **Tooltips** well.

From the list presented under **Sort axis**, select **Sum Points**. This definitive action explicitly tells [Power BI](#) to recalculate the order of the X-axis categories (Teams) based on the magnitude of the **Sum Points** measure associated with each category. By default, Power BI often applies a

descending sort upon selection, immediately placing the category with the highest total value at the start of the axis.

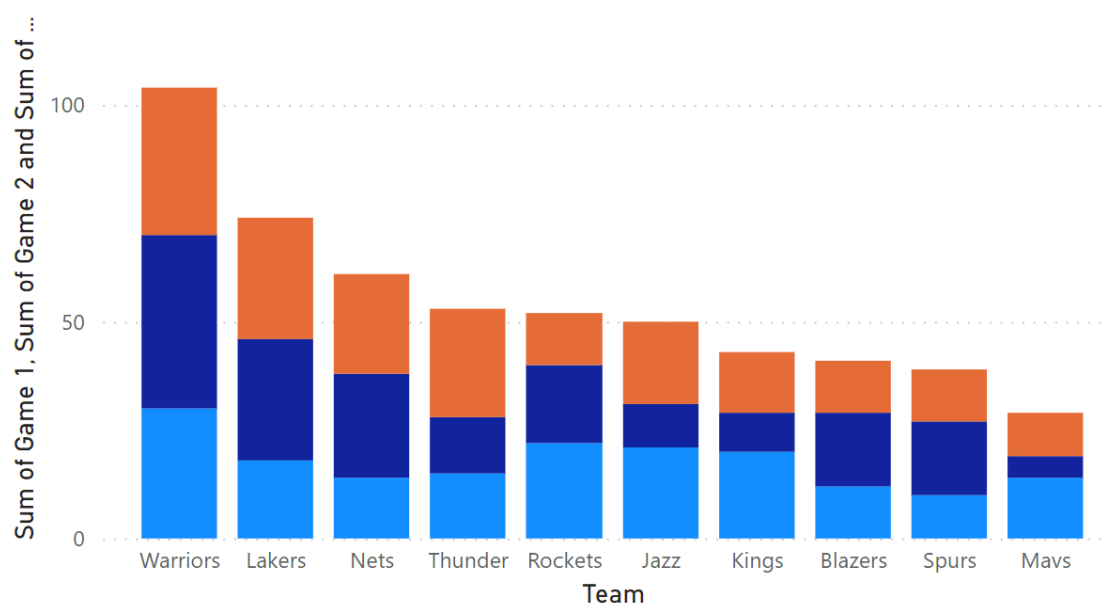


Immediately after selecting **Sum Points**, the chart visualization will update automatically. The columns are now correctly ordered based on their total height, moving from the highest aggregate score to the lowest. This sorted arrangement significantly enhances the chart's analytical utility, providing a clear, ranked view of overall player performance. If ascending order is preferred (i.e., starting with the lowest score), simply revisit the **Sort axis** menu and click **Sort descending** or **Sort ascending** to toggle the direction.

The resulting visualization confirms that the sorting process was successful, providing an accurate, total-based ranking of the teams, which is invaluable for performance comparison.

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



Conclusion and Best Practices

The ability to effectively sort complex visualizations like the stacked column chart is paramount for creating insightful and actionable business intelligence reports in [Power BI](#). Relying solely on default alphabetical or category-specific sorting often misrepresents the underlying data trends and hampers rapid analysis. By introducing a dedicated [DAX](#) measure to calculate the total aggregation of all components within a stack, report developers gain precise, dynamic control over the visual hierarchy.

This methodology of using a measure for sorting is a versatile and critical best practice in professional Power BI development. It ensures that the visual order remains consistent and meaningful, even as the source data changes or as users interact with slicers and filters across the report. Key best practices derived from this process include always defining aggregated totals as dynamic measures rather than static calculated columns, especially when those totals are intended to influence visual behavior such as sorting or conditional formatting.

Furthermore, a crucial technical detail to remember is that the measure used for sorting does not need to be displayed on the axes; merely including it in the **Tooltips** well is sufficient to make it available as a valid sorting criterion for the visual's axis. Mastering this technique allows for the creation of far more intuitive and professional-grade dashboards that guide the user's focus immediately to the most relevant data points.

Additional Resources

To further enhance your skills in Power BI data modeling and visualization, explore the following relevant tutorials and documentation, which cover complementary tasks necessary for creating compelling reports:

Understanding Row Context vs. Filter Context in DAX calculations.

Advanced usage of the [CALCULATE](#) function for modifying evaluation context.

Techniques for applying dynamic conditional formatting in Power BI visuals.