

How to Sort a Stacked Column Chart in Power BI

Authored by
mohammed loot

January 10, 2026

RECOMMENDED CITATION

mohammed loot (2026). *How to Sort a Stacked Column Chart in Power BI*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125368>

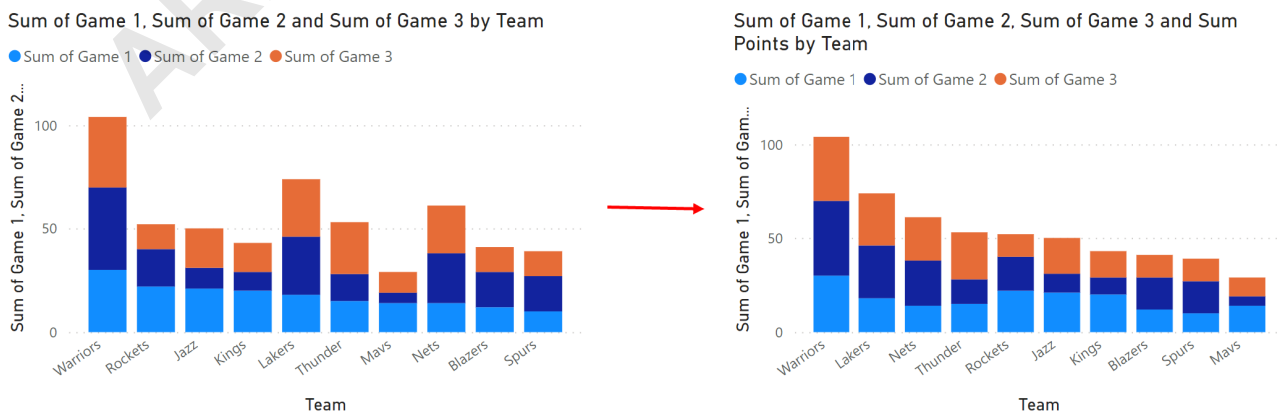
The Stacked Column Chart is one of the most frequently used visualizations within Power BI, offering a clear way to compare totals while simultaneously showing the composition of those totals. While sorting a simple column chart is straightforward, managing the sort order in a stacked chart often presents a challenge. Users typically need to sort the chart based not on a single segmented value, but on the accumulated total height of the bars.

Achieving this specific sorting requirement--ordering the columns by their cumulative height--cannot be done using standard field sorting options alone. This is because the visual attempts to sort based on the specific fields placed on the Y-axis (the segmentation), rather than the implied total. To correctly organize the data visualization by the total aggregate value, we must leverage the power of Data Analysis Expressions (DAX) by creating a dedicated aggregation Measure. This sophisticated approach ensures that the chart provides the most meaningful and insightful view of the underlying data trends.

Understanding the Challenge of Sorting Stacked Visuals

When dealing with complex visualizations like the Stacked Column Chart, the default sorting behavior in Power BI often focuses on the category axis (X-axis) alphabetically or on the first measure listed in the Y-axis. However, data analysts frequently require the columns to be ordered by the total value represented by the entire bar. For instance, if you are charting regional performance broken down by product category, you would want the regions to be ordered from the highest total performance to the lowest total performance.

The image below illustrates the typical scenario where we need to sort the columns according to their overall height, moving beyond the simple alphabetical or segmented sorting defaults. Without a custom sorting solution, identifying the largest contributors at a glance becomes difficult, hindering rapid data interpretation and executive decision-making.



This limitation highlights why a foundational understanding of DAX and the creation of explicit calculated fields is essential for advanced data manipulation in the Power BI environment. By defining a specific Measure that captures the total sum across all segments, we provide the visualization engine with the necessary key to govern the sorting behavior correctly.

The Importance of Custom Measures for Sorting Aggregates

The core solution to sorting stacked columns lies in creating a custom aggregation field. This field must be a Measure, not a calculated column, as it needs to dynamically calculate the sum based on the filter context applied by the visual's X-axis (in this case, the player or team name). This dedicated measure calculates the sum of all components that make up the stacked bar, regardless of which individual components are visible or segmented.

This technique is straightforward once the necessary calculation is defined. The approach involves using powerful DAX iterator functions, specifically combined with context manipulation, to ensure the summation occurs across the necessary fields for each unique column on the X-axis. This process effectively tells Power BI exactly how to rank the bars, based on their true cumulative value.

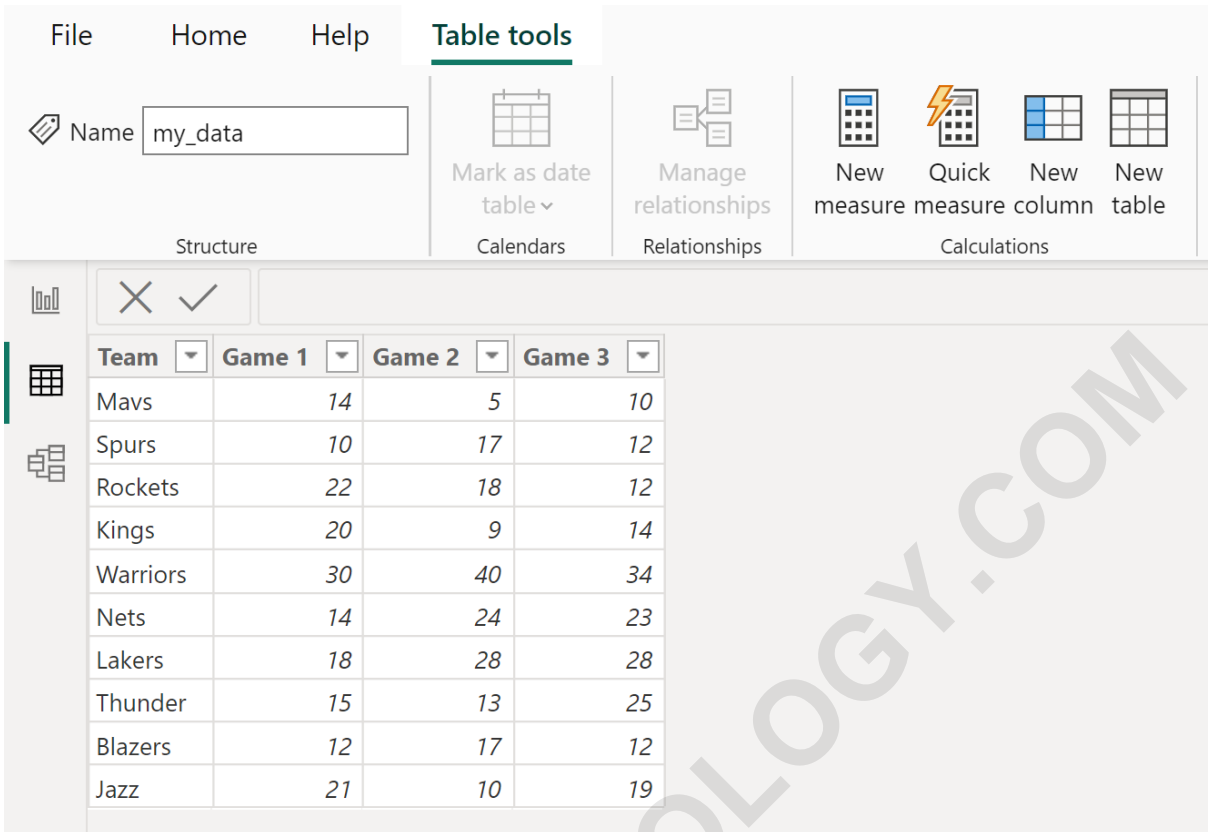
The following detailed example will demonstrate precisely how to implement this solution, transforming raw data into a dynamic and logically sorted Stacked Column Chart.

Preparing the Data Model in Power BI Desktop

Before creating the measure and the visualization, it is crucial to understand the structure of the source data. For this example, we assume we have a table named **my_data** loaded into Power BI. This table contains performance statistics, tracking points scored by various basketball players across three distinct games. Each row represents a player's statistics, and the columns represent the performance in each game.

The structure of the data table is essential because the subsequent DAX calculation relies directly on these column names. Note that we have separate columns for **Game 1**, **Game 2**, and **Game 3**, which will form the stacked segments in our final chart. If the data were already unpivoted (in a long format), the DAX calculation would be slightly different, but the principle of creating a total summation measure remains the same.

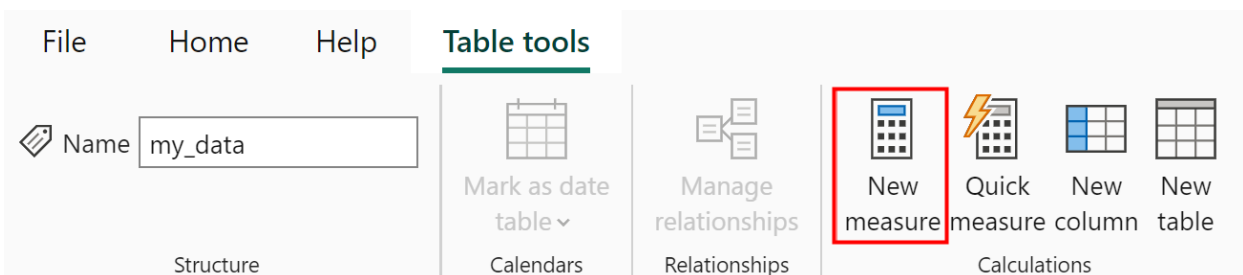
Review the sample data structure below, which serves as the foundation for our visualization exercise:



Step-by-Step Guide to Creating the Aggregation Measure

Our primary objective is to calculate the total points scored by each player across all three games. This total, encapsulated in a new Measure, will be the key sorting field. If we skip this step, Power BI would only be able to sort the chart based on the value of **Game 1** or an alphabetical ordering of the players, which is not aligned with our analytical goal.

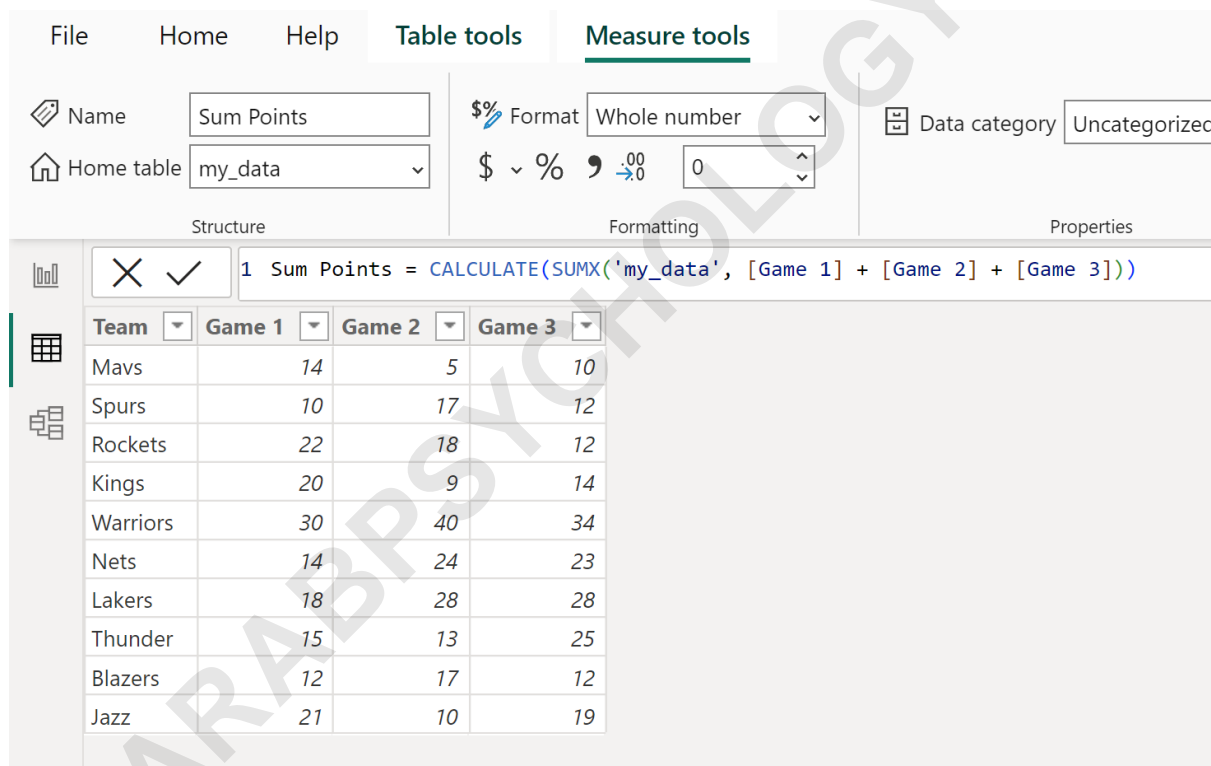
To initiate the creation of the measure, navigate to the **Table tools** tab in the ribbon at the top of the Power BI Desktop interface. Within this tab, locate and click the **New measure** icon. This action opens the formula bar, allowing us to input our custom DAX expression. This is where we define the calculation that will calculate the cumulative total for each player.



Enter the following formula into the formula bar. This expression utilizes both the CALCULATE and SUMX functions to correctly aggregate the scores within the context of the visual. The SUMX function iterates over the rows of the 'my_data' table, summing the points from all three game columns for each row, and the outer CALCULATE function ensures this total is evaluated correctly when used in the visualization context.

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

Once the formula is confirmed, a new measure named **Sum Points** will appear in your Fields pane. This field now holds the aggregated total for each player, derived across all three games, which is exactly what we need for sorting the stacked bars. The resulting data view, including the new measure, should look similar to the image below:



The screenshot displays the Power BI Desktop interface. The **Measure tools** ribbon is active, showing the formula bar with the measure definition: `1 Sum Points = CALCULATE(SUMX('my_data', [Game 1] + [Game 2] + [Game 3]))`. The ribbon also shows the Name field set to 'Sum Points', the Home table set to 'my_data', and the Format set to 'Whole number'. The Data category is set to 'Uncategorized'. Below the ribbon, a data table is visible 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

Visualizing the Data using the Stacked Column Chart

With the critical aggregation Measure defined, the next step is to construct the Stacked Column Chart itself in the Report View of Power BI Desktop. Switch to the Report View and select the Stacked Column Chart icon from the **Visualizations** pane to place an empty visual canvas on your report page.

The proper configuration of the visual fields is essential to display the data correctly. Follow these

steps to map the necessary fields:

Drag the **Team** (or Player) field onto the **X-axis** well. This defines the categories (the individual bars) we wish to sort.

Drag **Game 1**, **Game 2**, and **Game 3** onto the **Y-axis** well. These fields create the segmented stacks within each column.

Drag the newly created **Sum Points** measure onto the **Tooltips** well. Placing the measure in the Tooltips is crucial. Although it won't be displayed directly on the axes, its presence in the visual's context allows Power BI to use it for sorting purposes in the next step.

The configuration of the fields should mirror the setup shown in the image below, confirming that all segments and the sorting measure are correctly assigned:

ARABPSYCHOLOGY.COM

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 with the following data:

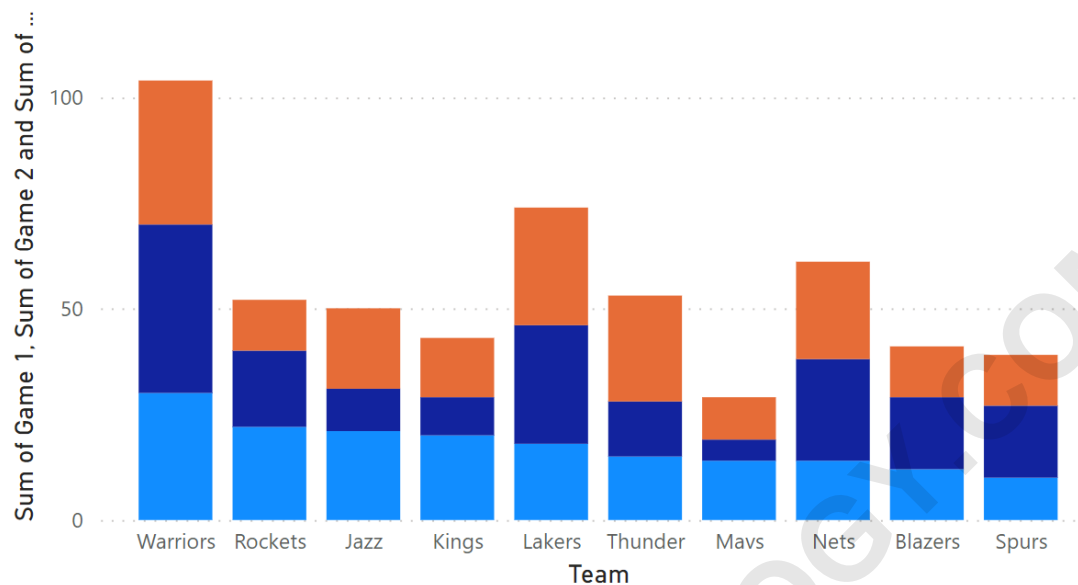
Game 1	Game 2	Game 3	Sum Points	Team
10	20	30	60	Team A
20	30	40	90	Team B
30	40	50	120	Team C

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.

Initially, this configuration produces the stacked column chart, but the bars are likely sorted by default settings--either alphabetically by team or by the value of the first segment (Game 1). This preliminary chart clearly demonstrates the need for applying our custom sorting logic, as the highest bars are not yet clustered together for easy 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



Implementing the Sorting Logic Using the Custom Measure

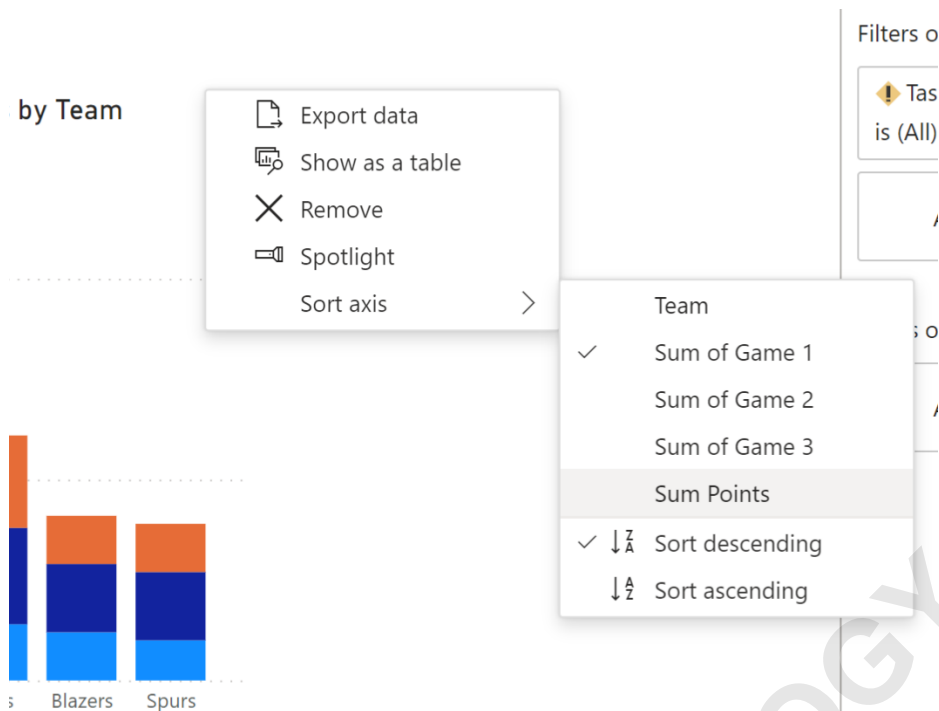
The final and most crucial step is instructing Power BI to use the **Sum Points** measure, rather than the default category or segment values, to determine the order of the columns. Since we have successfully included the **Sum Points** measure within the visual's context (in the Tooltips field), we can now access it via the visual's sorting options.

To change the sort order, follow these two simple actions:

Click the visual to select the chart. Locate the three horizontal dots (the "More options" menu) in the top right corner of the chart visualization.

Click **Sort axis** from the dropdown menu, and then select the custom measure **Sum Points**.

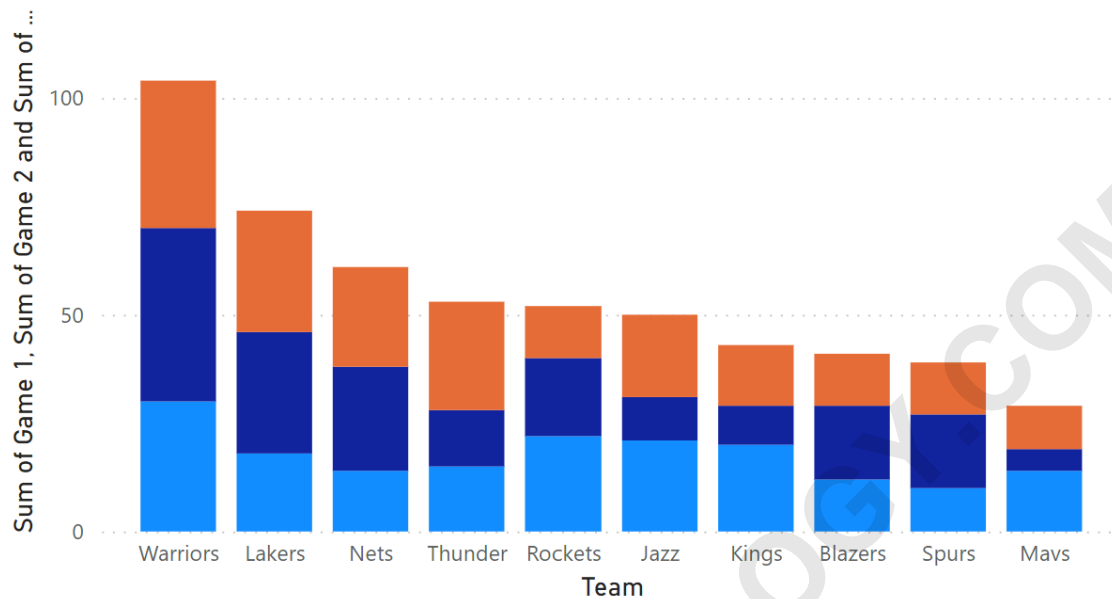
Selecting **Sum Points** immediately overrides the existing sorting logic. The chart will reorganize itself based on the total cumulative score calculated by our Measure. By default, Power BI usually applies descending sort order upon selection, placing the highest total score (the tallest bar) on the left.



Once this setting is applied, the bars in the chart will instantly be sorted based on the total aggregated score, typically in descending order. This optimized view makes the comparison between players intuitive and efficient. If you prefer ascending order, you can simply click **Sort descending** or **Sort ascending** from the same menu.

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 for Data Visualization

The ability to accurately sort a Stacked Column Chart by the cumulative total height is a fundamental skill for advanced users of Power BI. This technique transcends simple visualization setup, requiring a thoughtful application of DAX to control the context and calculations that govern the report's presentation. Using a calculated Measure ensures that the data is not just visually represented, but logically ordered for maximum analytical utility.

Remember these key best practices when implementing custom sorting in your reports:

Use Measures for Aggregates: Always rely on explicit Measures for complex sorting logic, especially when aggregation is required across multiple fields.

Placement in Tooltips: The custom sorting measure does not need to be on the Y-axis, but it must be included in the visual's field wells (Tooltips or Legend, depending on the visual) to be selectable in the sorting menu.

Clarity: Ensure your measure name (**Sum Points**) is descriptive so that other report designers immediately understand its purpose.

By mastering the creation and utilization of custom measures for sorting, analysts can ensure that their visualizations immediately draw attention to the most important data points, whether they represent the highest total revenue, the greatest number of incidents, or, as demonstrated here,

the overall top performers.

For further exploration of advanced Power BI functionalities and visualization techniques, consider the following related topics:

How to calculate running totals in Power BI using DAX.

Methods for integrating custom R visuals into Power BI reports.

Techniques for drilling through data hierarchies effectively.

ARABPSYCHOLOGY.COM