

How to Calculate the Average in Power BI, Excluding Zeros

Authored by
stats writer

January 28, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Calculate the Average in Power BI, Excluding Zeros*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128236>

The calculation of averages is a fundamental task in data analysis, yet it frequently presents subtle challenges, especially when dealing with large datasets containing disparate values. In Power BI, the standard average function efficiently computes the mean of a specified column. However, the presence of zero values can significantly distort the resulting average, leading to misrepresentative metrics and flawed interpretations. This distortion is particularly critical in fields like finance, performance tracking, or inventory management, where a zero often signifies a non-event or an entry that should be excluded from performance calculations, rather than being counted as a minimal value. To ensure the accuracy and reliability of analytical findings, analysts must employ advanced techniques to perform context-aware calculations, specifically learning how to calculate the average while explicitly ignoring these unwanted zeros. This capability, achieved through custom formulas using DAX, is essential for generating meaningful insights and enhancing the quality of data-driven decisions within the Microsoft Power Platform ecosystem.

The standard methodology in descriptive statistics dictates that all numerical values within a defined set contribute equally to the average calculation. While mathematically correct, this approach often fails to align with the real-world business logic. For instance, if you are averaging sales performance, a zero might indicate a day off or a period when a product was unavailable, not necessarily a poor performance point that should drag down the true average of active sales days. To accurately reflect performance or utilization rates, the calculation must be conditional. Power BI provides powerful tools, primarily through the DAX language, which enable users to establish specific filtering criteria--such as ignoring zero values--before the aggregation occurs. By mastering this conditional averaging technique, users can obtain a statistic that is not only mathematically sound but also contextually appropriate and representative of the underlying data trends, thereby strengthening the empirical foundation of their analyses.

Power BI: Calculate Average and Ignore Zeros

The Challenge of Zero Values in Data Aggregation

In many analytical scenarios, the simple arithmetic mean provided by the default `AVERAGE` function in Power BI is insufficient because it treats every numerical entry equally, including those with a value of zero. When zero represents the absence of data or an irrelevant data point (e.g., a non-score or an inactive period), its inclusion unjustly reduces the overall calculated average. This is a common issue in performance dashboards where analysts seek to determine the average performance only during periods of activity, effectively requiring a calculation that ignores values where no activity was recorded.

Consider a dataset tracking athlete performance, where some games resulted in zero points due to injury or absence. If we calculate the standard average across all games, the high-scoring games are artificially diluted by the zero entries. To get a true measure of the athlete's scoring potential

during active play, we need to implement a mechanism that dynamically removes rows where the score column equals zero before the aggregation process begins. This transformation requires moving beyond simple column aggregation and leveraging the context modification capabilities inherent in DAX.

Addressing this challenge necessitates creating a custom measure. This measure will employ advanced DAX functions to establish a new calculation context. Specifically, we use a combination of functions to tell Power BI: "First, identify all rows where the value is not zero, and only then apply the average calculation to that reduced set of rows." This structured approach ensures that the resulting average truly reflects the metrics we intend to track, providing a clean and reliable statistical summary.

Introducing DAX: The Language of Power BI Calculations

DAX, or Data Analysis Expressions, is the powerful formula language used throughout Microsoft Power BI, Power Pivot in Excel, and Analysis Services. Unlike simple column operations, DAX allows users to define custom calculations, known as measures, which evaluate data based on the current filtering context. This ability to manipulate context is the core mechanism we will exploit to ignore zero values effectively. Standard functions like SUM or AVERAGE operate on the default row context, but the powerful combination of functions allows us to introduce complex, conditional logic.

To implement conditional averaging, we must utilize two primary DAX functions: CALCULATE and FILTER. The CALCULATE function is arguably the most important function in DAX, as it evaluates an expression within a modified filter context. It takes an expression (like our AVERAGE calculation) and one or more filters. The FILTER function, conversely, is an iterator that evaluates a condition over every row of a specified table, returning only the rows that satisfy the condition. By nesting the AVERAGE function inside CALCULATE and applying a FILTER that excludes zeros, we effectively create an "AVERAGEIF" functionality that is not natively available as a single function in DAX.

The resulting measure is reusable across different visuals and contexts within the Power BI report, ensuring consistency and efficiency. This specific structure provides the necessary control to perform selective aggregations, moving beyond basic summary statistics to sophisticated, context-sensitive calculations that are vital for advanced business intelligence applications.

The Advanced Solution: Combining CALCULATE and FILTER

The recommended approach for calculating the average while explicitly ignoring zero values involves leveraging the power of context transition within DAX. The following formula syntax demonstrates how to calculate the average value in a column while rigorously excluding any values equal to zero. This formula creates a new, sophisticated measure that modifies the

evaluation context of the standard AVERAGE function.

```
Avg Points =  
CALCULATE (  
AVERAGE ( 'my_data' ),  
FILTER ( 'my_data', 'my_data' <> 0 )  
)
```

Let's break down this powerful formula piece by piece to understand its mechanism. First, the entire operation is wrapped in the CALCULATE function, which is tasked with evaluating the first argument--the expression--under the conditions defined by the subsequent arguments--the filters. The expression here is the simple AVERAGE of the `Points` column: `AVERAGE('my_data')`. This expression, on its own, would calculate the average including zeros.

The key modification comes from the second argument: the FILTER function. The FILTER function takes the entire table (`'my_data'`) and applies a boolean condition (`'my_data' <> 0`). This condition instructs FILTER to only pass through rows where the value in the `Points` column is not equal to zero. The result of the FILTER function is a virtual table containing only the non-zero rows, and this virtual table is then used by the CALCULATE function to execute the AVERAGE calculation. This structure ensures that the average is computed exclusively on the data points that hold meaningful, non-zero values, successfully bypassing the skew caused by irrelevant zero entries. This particular example creates a new measure named **Avg Points** that calculates the average value in the **Points** column of the table named **my_data** while precisely ignoring any values equal to zero.

Practical Application Example: Setting Up the Scenario

To illustrate this technique in a practical context, let us consider a dataset within Power BI named **my_data**. This table simulates tracking performance scores--specifically, points scored by basketball players across various games or teams. As is typical in real-world data collection, this table contains multiple entries where the score is recorded as zero, representing games missed or performances resulting in no points, which we wish to exclude from our performance metric calculation.

The data structure includes columns for identification and the core numerical data we are interested in, the **Points** column. Below is a representation of the data within the **my_data** table:

The screenshot shows the Power BI Desktop interface. The 'Table tools' ribbon is active, displaying 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 visible with the following data:

Team	Points	Assists
Mavs	22	4
Rockets	0	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	0	7
Spurs	0	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

A close inspection of the data reveals several values equal to zero within the **Points** column. If we were to calculate the standard arithmetic mean of this column using the default AVERAGE function, these zero values would pull the overall average downwards, potentially suggesting a lower typical performance than is actually representative of the players during active play. Our goal is to calculate the average value in the **Points** column while ensuring these specific zero values are entirely disregarded, thus providing a more accurate representation of the scoring average achieved.

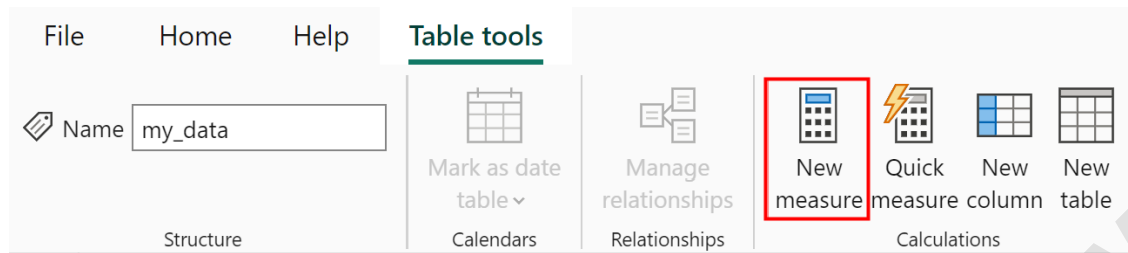
The following steps demonstrate the procedure for implementing the CALCULATE and FILTER solution directly within the Power BI Desktop interface, transforming this analytical requirement into a robust and deployable measure that can be used throughout the report.

Step-by-Step Implementation: Creating the Custom Measure

Implementing the custom averaging logic begins in the Power BI Desktop environment, where we define a new measure using the DAX formula previously discussed. Measures are crucial for dynamic calculations that respond to report filters and interactions.

Navigate to Measure Creation: Begin by ensuring you are on the data model view or the report view where the table **my_data** is present. To initiate the creation of a new measure, locate and

click the **Table tools** tab positioned along the top ribbon of the Power BI interface. Within this tab, select the **New measure** icon.



Input the DAX Formula: A formula bar will appear, prompting you to define the new measure. Into this bar, carefully type or paste the complete DAX formula that utilizes the CALCULATE and FILTER functions to exclude zero values from the averaging process:

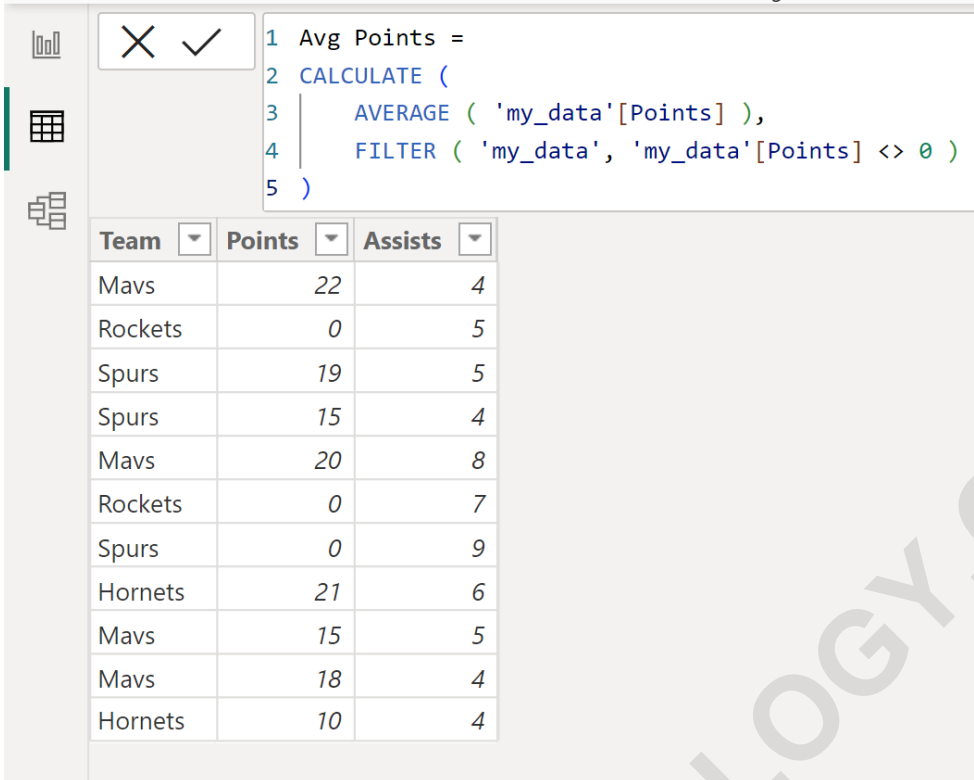
```
Avg Points =  
CALCULATE (  
  AVERAGE ( 'my_data' ),  
  FILTER ( 'my_data', 'my_data' <> 0 )  
)
```

This formula defines a new calculation called **Avg Points**. This name clearly indicates the purpose of the measure. Once the formula is entered and confirmed (by pressing Enter or clicking the checkmark), the measure is saved to your data model, typically appearing under the **my_data** table fields.

The successful implementation of this step will result in a new field in your Fields pane. This field, **Avg Points**, encapsulates the custom logic, ensuring that whenever it is used in a visual, it automatically calculates the average of the **Points** column while systematically excluding all entries where the point value is zero. This powerful combination of functions allows for precise control over data aggregation, delivering an accurate statistical result based on the required business rules.

Displaying and Validating the Results

Once the **Avg Points** measure has been successfully created, the next essential step is to display and validate the calculated value to confirm its accuracy against manual calculation. The measure itself contains the average of values in the **Points** column, specifically ignoring zero entries, as shown in the data model view:



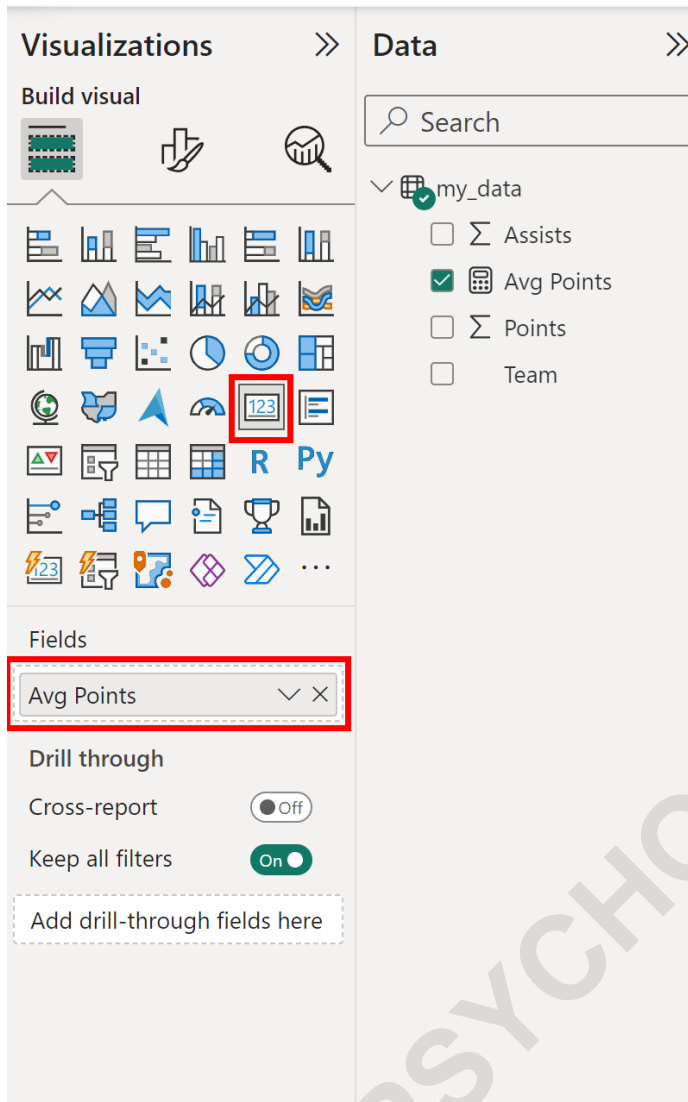
The screenshot shows the DAX editor in Power BI. The formula bar contains the following DAX code:

```
1 Avg Points =  
2 CALCULATE (  
3     AVERAGE ( 'my_data'[Points] ),  
4     FILTER ( 'my_data', 'my_data'[Points] <> 0 )  
5 )
```

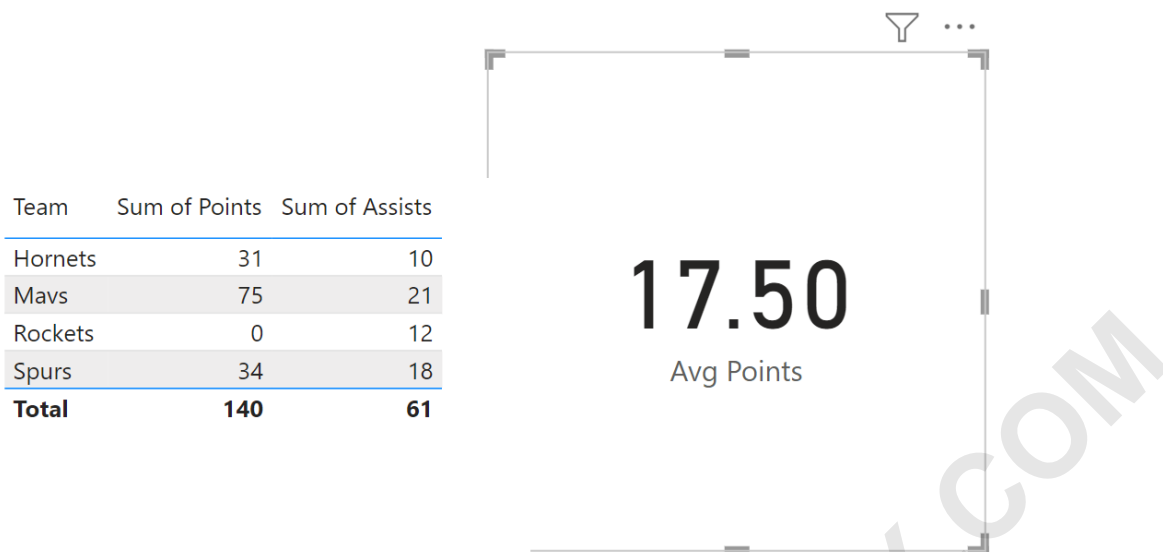
Below the formula bar, a table is displayed with the following data:

Team	Points	Assists
Mavs	22	4
Rockets	0	5
Spurs	19	5
Spurs	15	4
Mavs	20	8
Rockets	0	7
Spurs	0	9
Hornets	21	6
Mavs	15	5
Mavs	18	4
Hornets	10	4

To visualize this result, transition to the **Report View** in Power BI. The Card visualization is the most straightforward method for displaying a single numerical aggregate like an average. To add the visualization, navigate to the **Visualizations** pane, click on the **Card** icon, and then drag the newly created **Avg Points** measure into the **Fields** well of the Card visual.



The final visualization, displaying the result of the conditional average, will clearly show the computed value, which is based only on the non-zero rows in the dataset. This outcome confirms that the CALCULATE and FILTER functions worked as intended, restricting the context before the AVERAGE function was applied. The resulting Card visual will display the average points, which in this example, is **17.5**.



We can decisively verify the correctness of this calculation by manually excluding the zero values from the raw data and calculating the average. Reviewing the **Points** column, the non-zero values are: 22, 19, 15, 20, 21, 15, 18, and 10. There are 8 such values. Manually calculating the average confirms the result: Average of Points while Ignoring Zeros: $(22 + 19 + 15 + 20 + 21 + 15 + 18 + 10) / 8 = 140 / 8 = 17.5$. This manual calculation precisely matches the value computed by our custom measure, confirming the accuracy and effectiveness of the DAX formula used to ignore zero entries.

Conclusion: Benefits of Context-Aware Averaging

Mastering conditional aggregation techniques in Power BI, particularly the ability to calculate averages while excluding specific values like zero, is indispensable for generating sophisticated and accurate business intelligence. The use of the CALCULATE function in conjunction with the FILTER function provides analysts with granular control over the data evaluation context, allowing the report to reflect complex business logic accurately. This powerful combination moves beyond basic statistical summaries, ensuring that metrics such as average performance, utilization rates, or transaction values are not unfairly skewed by irrelevant data points.

By implementing the **Avg Points** measure, we have demonstrated a robust and repeatable solution that yields a true, context-aware average. The clarity and precision gained from excluding zeros lead directly to more reliable reports and more informed strategic decisions. This technique is highly transferable and can be easily adapted to exclude other outlier values, such as negative numbers or specific error codes, simply by modifying the filtering condition within the FILTER argument.

Understanding and applying these advanced DAX constructs is a hallmark of expert Power BI development. These methods not only solve immediate data integrity challenges but also future-proof reports against common data quality issues, ensuring that the analytical output remains consistent, valid, and highly valuable across the organization.

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

ARABPSYCHOLOGY.COM