

How to Normalize Data Between 0 and 1 in Power BI

Authored by
mohammed looti

January 11, 2026

RECOMMENDED CITATION

mohammed looti (2026). *How to Normalize Data Between 0 and 1 in Power BI*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=125553>

The Importance of Data Scaling in Power BI

In the world of business intelligence and data analysis, ensuring data consistency is paramount. When working with Power BI, analysts often encounter datasets where different columns measure variables on vastly different scales. For instance, comparing revenue (in millions) with transaction count (in thousands) can skew visualizations and analytical models. This is where data normalization becomes critical.

Normalization scales numerical data into a standard range, typically between 0 and 1. This process, often referred to as Min-Max scaling, ensures that all data points are proportionally represented on a uniform scale. By normalizing the data, you eliminate the bias that large magnitude variables might introduce, leading to fairer comparisons and more robust analytical outcomes, especially in advanced analytical contexts like clustering or scoring models. Implementing this transformation directly within Power BI usually involves leveraging the powerful calculation engine known as DAX (Data Analysis Expressions).

While some specialized statistical software includes a built-in "Normalize" function for data transformation, in Power BI, this calculation is typically achieved through explicit formulas using DAX. This allows for precise control over the normalization scope, whether it applies across the entire column or within specific groups. The fundamental goal is simple: to make disparate measures comparable by bringing them onto the same playing field, 0 being the minimum observed value and 1 being the maximum observed value.

Understanding Min-Max Normalization (0 to 1 Scaling)

Min-Max normalization is the technique specifically used when the target range is 0 to 1. This method is exceptionally useful because it preserves the original distribution of the data while compressing the range. Unlike other scaling methods, such as standardization (Z-score), Min-Max scaling guarantees that the transformed values fall exactly within the desired 0 and 1 boundaries. This makes it ideal for scoring systems, machine learning features where non-negative inputs are required, or visualizations where relative contribution is key.

The core philosophy behind this technique is to measure how far away a specific data point is from the minimum value of the dataset, relative to the total range of the dataset. A value that is equal to the minimum in the original data will become 0 after transformation, and a value equal to the maximum will become 1. All intermediate values are scaled linearly between these two extremes. This transformation is mathematically straightforward but provides substantial analytical benefits by ensuring that outliers or extreme values do not disproportionately influence comparisons after scaling, provided the outliers themselves are not the intended focus of the analysis.

Using Min-Max scaling within Power BI allows the analyst to create derived measures that

integrate seamlessly with existing report structures. Since the calculation is performed using DAX, it is dynamic and refreshes automatically whenever the underlying source data is updated, ensuring that visualizations always reflect the current, normalized state of the metrics. This reliability is a major advantage of using calculated columns or measures over pre-processing data outside of the Power BI environment.

The Mathematical Foundation of Normalization

To effectively implement 0-to-1 normalization, one must understand the underlying mathematical formula, often known as the transformation equation for Min-Max scaling. This formula dictates how each individual observation (x_i) is mapped to its new normalized counterpart (z_i).

The calculation hinges on identifying the absolute minimum and maximum values within the entire data column being analyzed. Once these boundaries are established, the formula provides a clear path for transformation:

To normalize values in a dataset to be between 0 and 1, you can use the following formula:

$$z_i = (x_i - \min(x)) / (\max(x) - \min(x))$$

where:

z_i : The i th normalized value in the dataset

x_i : The i th value in the dataset

$\min(x)$: The minimum value in the dataset

$\max(x)$: The maximum value in the dataset

This formula ensures a linear transformation. The numerator calculates the distance of the current point from the minimum, and the denominator calculates the total range of the data. Dividing the distance by the range gives the proportional position of the current point within the 0-to-1 scale. It is crucial when applying this in DAX that the minimum and maximum values are calculated across the entire column context, regardless of any row context being evaluated, to maintain consistent scaling across the entire dataset.

Implementing Normalization using DAX Variables

While the mathematical formula is straightforward, translating it into functional DAX code requires careful use of variables and functions to manage context transitions properly. The most robust way to calculate normalization in Power BI is by defining the minimum and maximum values using variables, ensuring they are fixed for the entire column calculation, and then applying the scaling logic.

The use of the `VAR` function in `DAX` is essential here. Variables allow the complex calculation to be broken down into manageable, readable steps. By defining variables for the current value (X_i), the Minimum Value (`MinValue`), and the Maximum Value (`MaxValue`), the final return statement becomes simple and less prone to calculation errors, especially those related to calculation context.

To create a new calculated column containing these normalized values, you use the following `DAX` syntax structure. Note the use of the `DIVIDE` function, which is best practice in `DAX` to handle potential division by zero errors, although such an error is unlikely if the `MaxValue` equals the `MinValue` in a real-world scenario where a range is expected.

To create a new column of normalized values in Power BI, you can use the following syntax in `DAX`:

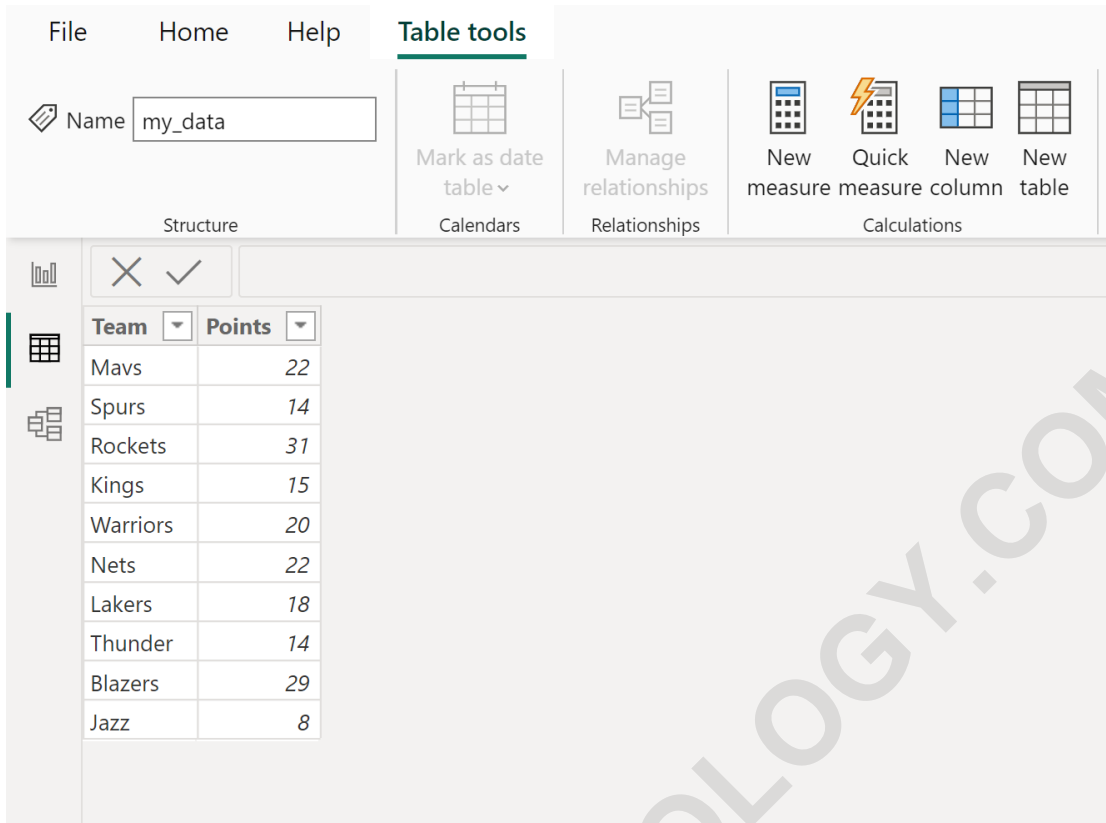
```
Normalized Points =  
VAR Xi = 'my_data'  
VAR MinValue = MIN('my_data')  
VAR MaxValue = MAX('my_data')  
RETURN DIVIDE(Xi - MinValue, MaxValue - MinValue)
```

This specific implementation creates a new column named **Normalized Points**. It correctly applies the Min-Max scaling methodology by referencing the **Points** column within the table named **my_data** to derive its normalized representation, ensuring the calculation is context-aware yet consistent across the entire table.

Practical Example: Setting Up the Dataset in Power BI

To illustrate this process, let us consider a practical dataset within Power BI. Imagine we are analyzing athletic performance data where the table, named **my_data**, tracks the points scored by various basketball players. The raw point totals range from relatively low scores to very high scores, making direct comparison across different groups potentially misleading until normalized.

Suppose we have the following table in Power BI named **my_data** that contains information about points scored by basketball players on various teams:



The screenshot shows the Power BI ribbon interface with the 'Table tools' tab selected. The ribbon includes a 'Name' field containing 'my_data', a 'Mark as date table' button, a 'Manage relationships' button, and a 'Calculations' group with four options: 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, a table is displayed with the following data:

Team	Points
Mavs	22
Spurs	14
Rockets	31
Kings	15
Warriors	20
Nets	22
Lakers	18
Thunder	14
Blazers	29
Jazz	8

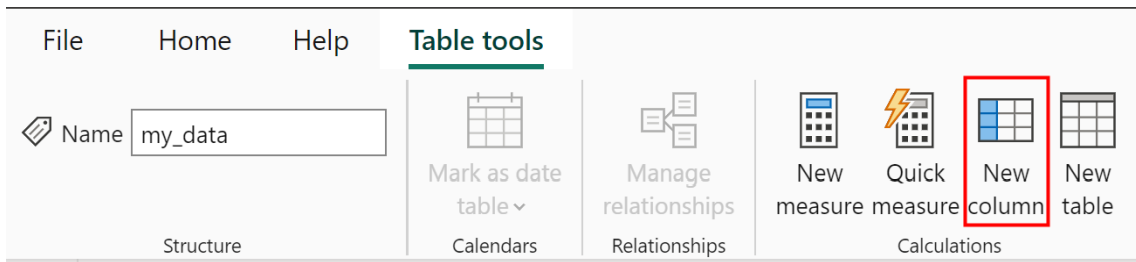
Our objective is to normalize each value in the **Points** column so that every score falls precisely within the range of 0 and 1. This transformation will allow us to immediately see a player's performance relative to the maximum and minimum performance recorded in this specific dataset.

Before proceeding with the DAX calculation, it is helpful to identify the global minimum and maximum values in the **Points** column. Observing the data set, we can see that the minimum score is 8 and the maximum score is 31. These two values define the range ($31 - 8 = 23$) that will be used as the divisor in the normalization formula for every single row, thus ensuring consistent and accurate normalization.

Step-by-Step DAX Implementation

The implementation process starts directly in the Power BI Data or Model view, where calculated columns are defined. This method ensures that the normalized values are stored alongside the original data, making them readily available for visual aggregation and filtering.

To initiate the creation of the new normalized column, navigate to the **Table tools** tab in the ribbon interface. Within this tab, locate and click the **New column** icon. This action opens the DAX formula bar, allowing us to input the required scaling logic.



Subsequently, the full DAX formula, utilizing variables for clarity and efficiency, must be typed or pasted into the formula bar:

Normalized Points =

VAR Xi = 'my_data'

VAR MinValue = MIN('my_data')

VAR MaxValue = MAX('my_data')

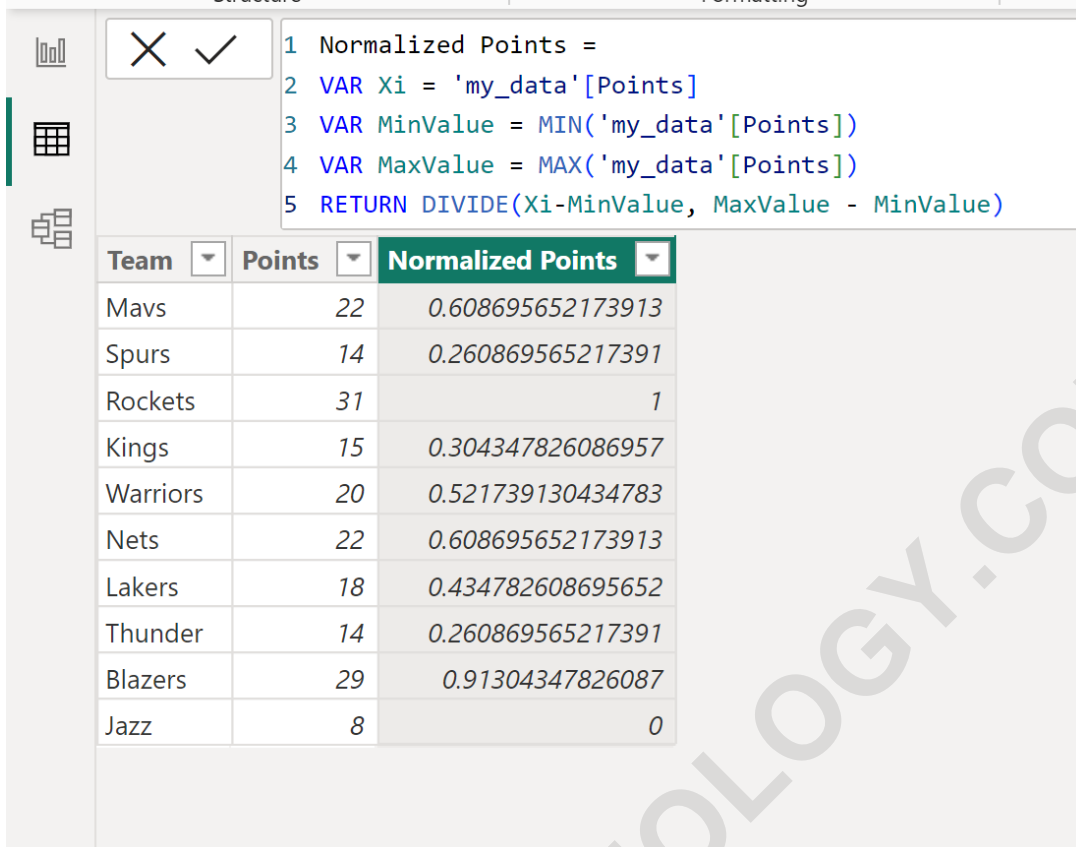
RETURN DIVIDE(Xi - MinValue, MaxValue - MinValue)

Executing this formula creates the new column. It is important to remember that because we used MIN and MAX functions without specific context modifiers like ALL, DAX automatically calculates these values across the entire column when defining a calculated column, achieving the required global Min-Max scaling across the dataset.

Verifying and Interpreting the Normalized Results

Upon successful execution of the DAX formula, a new column, **Normalized Points**, will appear in the **my_data** table. This column contains the proportional values of the original **Points** column, now constrained between 0 and 1.

This will create a new column named **Normalized Points** that contains the normalized values of the **Points** column of the table:



The screenshot shows a Power BI interface with a DAX formula bar and a data table. The formula bar contains the following DAX code:

```

1 Normalized Points =
2 VAR Xi = 'my_data'[Points]
3 VAR MinValue = MIN('my_data'[Points])
4 VAR MaxValue = MAX('my_data'[Points])
5 RETURN DIVIDE(Xi-MinValue, MaxValue - MinValue)

```

Below the formula bar is a table with three columns: Team, Points, and Normalized Points. The data is as follows:

Team	Points	Normalized Points
Mavs	22	0.608695652173913
Spurs	14	0.260869565217391
Rockets	31	1
Kings	15	0.304347826086957
Warriors	20	0.521739130434783
Nets	22	0.608695652173913
Lakers	18	0.434782608695652
Thunder	14	0.260869565217391
Blazers	29	0.91304347826087
Jazz	8	0

Crucially, every value in this new **Normalized Points** column adheres strictly to the defined range, running from a minimum of 0 (corresponding to the original score of 8) up to a maximum of 1 (corresponding to the original score of 31).

To confirm the calculation's accuracy, let us manually verify a specific data point. Consider the player with 22 points. We can trace how the normalization process applies the formula to this value:

$$z_i = (x_i - \min(x)) / (\max(x) - \min(x))$$

$$z_i = (22 - 8) / (31 - 8)$$

$$z_i = 14 / 23$$

$$z_i = 0.60869$$

This result, 0.60869, confirms that the original score of 22 is approximately 61% of the way between the minimum recorded score and the maximum recorded score. By applying this systematic normalization, we have successfully created a metric that provides immediate relative performance context, greatly enhancing the clarity and interpretability of subsequent Power BI reports and dashboards.

Advanced Applications and Alternatives to Normalization

While Min-Max normalization is exceptionally effective for creating proportional scales (0 to 1), it is important to recognize its limitations and alternative scaling methods available for advanced data transformation tasks. For instance, Min-Max scaling is highly sensitive to outliers; a single extreme value can compress all other data points into a very small portion of the 0-to-1 range.

For datasets where robust outlier handling is necessary, analysts often turn to Z-score standardization, which transforms data to have a mean of 0 and a standard deviation of 1. While Z-score standardization is not strictly normalization to a 0-to-1 range, it achieves a similar goal of bringing variables to a comparable scale. When choosing between these techniques, the choice depends heavily on the downstream modeling or visualization requirements. If absolute boundaries (0 and 1) are required, Min-Max scaling is the definitive choice.

Ultimately, mastering these techniques in Power BI, particularly through powerful DAX syntax and variable management, is crucial for any analyst seeking to build robust and insightful dashboards. The ability to perform complex analytical preparations, such as normalization, directly within the BI environment streamlines the data pipeline and ensures reproducible results.

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