

Learning to Normalize Data Between 0 and 1 in Power BI

Authored by
Mohammed loot

November 12, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Normalize Data Between 0 and 1 in Power BI*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=17686>

Understanding Data Normalization

Data normalization is a critical step in the [data transformation](#) pipeline, especially when preparing datasets for advanced analysis or visualization. When working within platforms like [Power BI](#), datasets often contain features measured on vastly different scales. For instance, one column might represent customer age (ranging from 18 to 70), while another tracks quarterly revenue (ranging from \$1,000 to \$5,000,000). If these raw values are used directly in certain calculations, the metric with the larger magnitude can disproportionately influence the results, leading to skewed or misleading insights.

The core purpose of [normalization](#) is to scale all numerical features into a standardized range, typically between 0 and 1. This process ensures that every attribute contributes equally to the analysis, regardless of its original scale. By standardizing the range, we facilitate fair comparisons and improve the performance of specific analytical models, such as those used in machine learning or complex statistical aggregations within Power BI reports.

The method discussed here, known as Min-Max Scaling (or feature scaling), is the most straightforward and commonly implemented technique for bounding data within the 0 and 1 interval. It relies only on the minimum and maximum values present in the data column being analyzed, making it robust for implementation using basic DAX functions. Understanding this foundational concept is the first step toward mastering complex data preparation techniques necessary for high-quality business intelligence reporting.

The Min-Max Normalization Formula

To effectively normalize a set of values to fall precisely between 0 and 1, we employ the [Min-Max Scaling](#) formula. This technique linearly transforms the data such that the minimum value in the original dataset corresponds to 0, and the maximum value corresponds to 1. All intermediate values are scaled proportionally within this new range.

The specific mathematical formula used for this transformation is as follows:

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

This formula requires three crucial components derived from the original dataset, x :

z_i : The i th normalized value in the dataset, which will be the resulting output between 0 and 1.

x_i : The i th value of the specific data point you are currently transforming.

$\min(x)$: The absolute minimum value found across the entire dataset (or column) x .

$\max(x)$: The absolute maximum value found across the entire dataset (or column) x .

The numerator, $(x_i - \min(x))$, calculates the distance of the current value from the minimum

observed value. The denominator, $(\max(x) - \min(x))$, represents the total range of the dataset. By dividing the distance from the minimum by the total range, we effectively determine where the current value sits on a scale from 0% (minimum) to 100% (maximum). This simple yet powerful mathematical approach ensures a clean and interpretable standardization of the data scale.

Implementing Normalization using DAX in Power BI

When implementing analytical calculations directly within [Power BI](#), we rely on the powerful expression language known as [DAX](#) (Data Analysis Expressions). To create a persistent column of normalized values, a calculated column must be defined in the Data View. This approach is superior to using Power Query for this specific transformation because DAX allows us to easily calculate the required minimum and maximum values across the entire column context, which is necessary for the Min-Max formula.

The most efficient and readable way to structure this calculation in DAX is by utilizing variables. Variables (`VAR`) enhance performance by calculating complex values once and then reusing them throughout the calculation, making the formula easier to debug and understand. We need variables to capture the current value (`Xi`), the minimum value of the entire column (`MinValue`), and the maximum value of the entire column (`MaxValue`).

The following DAX syntax demonstrates how to create a new calculated column that applies the Min-Max formula to a column named within a table called `'my_data'`:

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

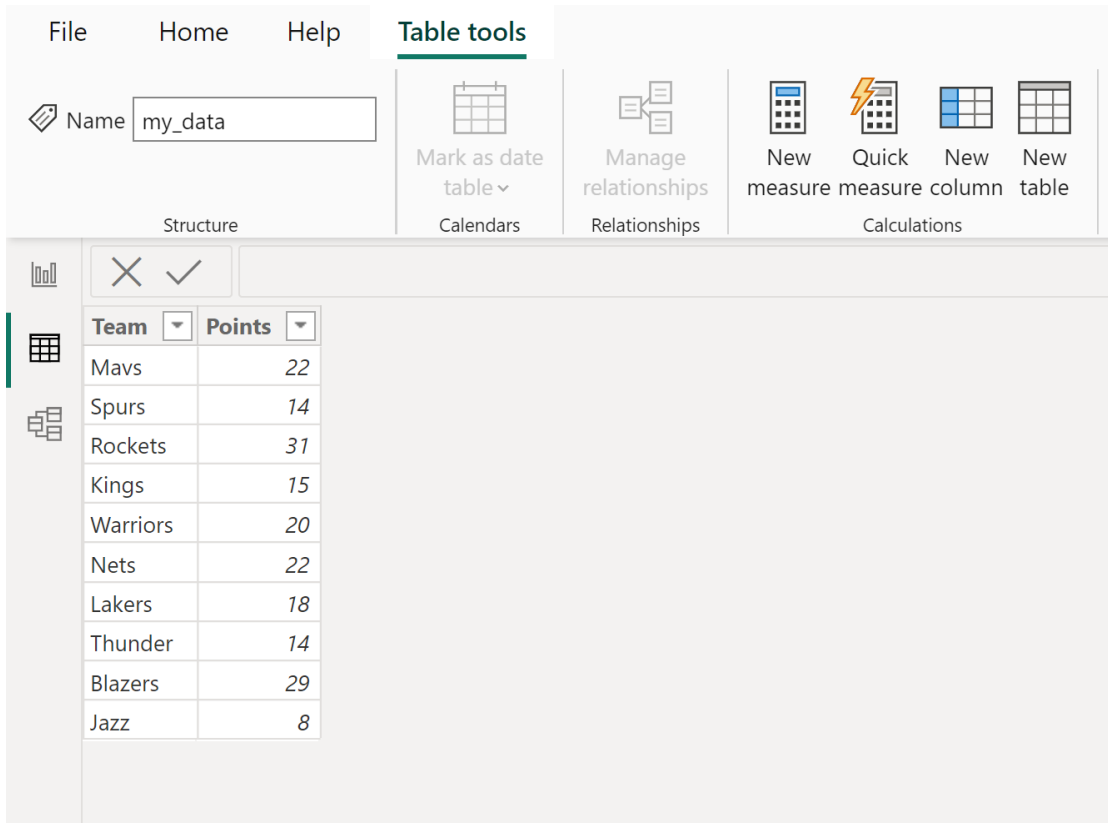
In this structure, the `MIN` and `MAX` functions operate over the entire column context by default, correctly retrieving the global minimum and maximum values needed for the denominator. The `DIVIDE` function is used instead of the standard division operator to safely handle potential division by zero errors, which might occur if the maximum and minimum values in the dataset are identical (i.e., if all values are the same). The resulting column, **Normalized Points**, provides the desired 0-to-1 representation of the original data.

Step-by-Step Example in Power BI Desktop

To solidify the understanding of this technique, let us walk through a practical example using a sample dataset within Power BI Desktop. Suppose we are analyzing performance data, and we

have a table named **my_data** that tracks the points scored by various individuals or teams.

This is the structure of our initial data table:

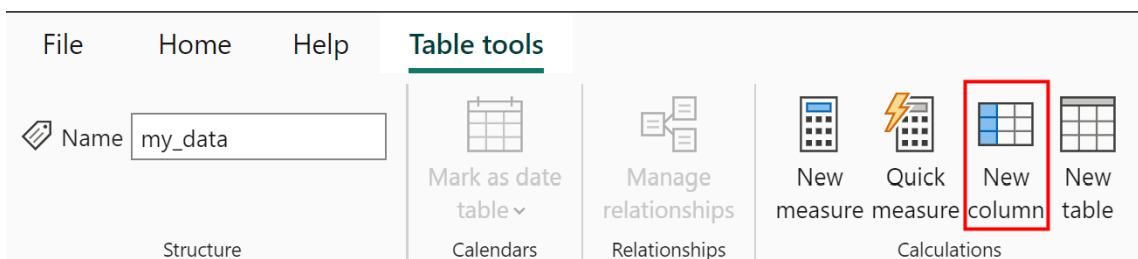


The screenshot shows the Power BI interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations'. Under 'Calculations', the 'New column' icon is highlighted. Below the ribbon, a data table is displayed with the following data:

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

Our goal is to apply [Min-Max Scaling](#) to the **Points** column. We want every value in this column to be rescaled such that the lowest point score becomes 0 and the highest score becomes 1, allowing for standardized comparative reporting.

To begin the transformation, navigate to the **Table tools** tab in the ribbon menu at the top of the Power BI interface while viewing the data table. From there, locate and click the **New column** icon. This action opens the [DAX](#) formula bar where you will input the calculation logic:



The screenshot shows the Power BI interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations'. Under 'Calculations', the 'New column' icon is highlighted with a red box.

Once the formula bar is active, paste or type the robust DAX formula we structured earlier. This

formula efficiently defines the variables and returns the scaled value for each row:

Normalized Points =

VAR Xi = 'my_data'

VAR MinValue = MIN('my_data')

VAR MaxValue = MAX('my_data')

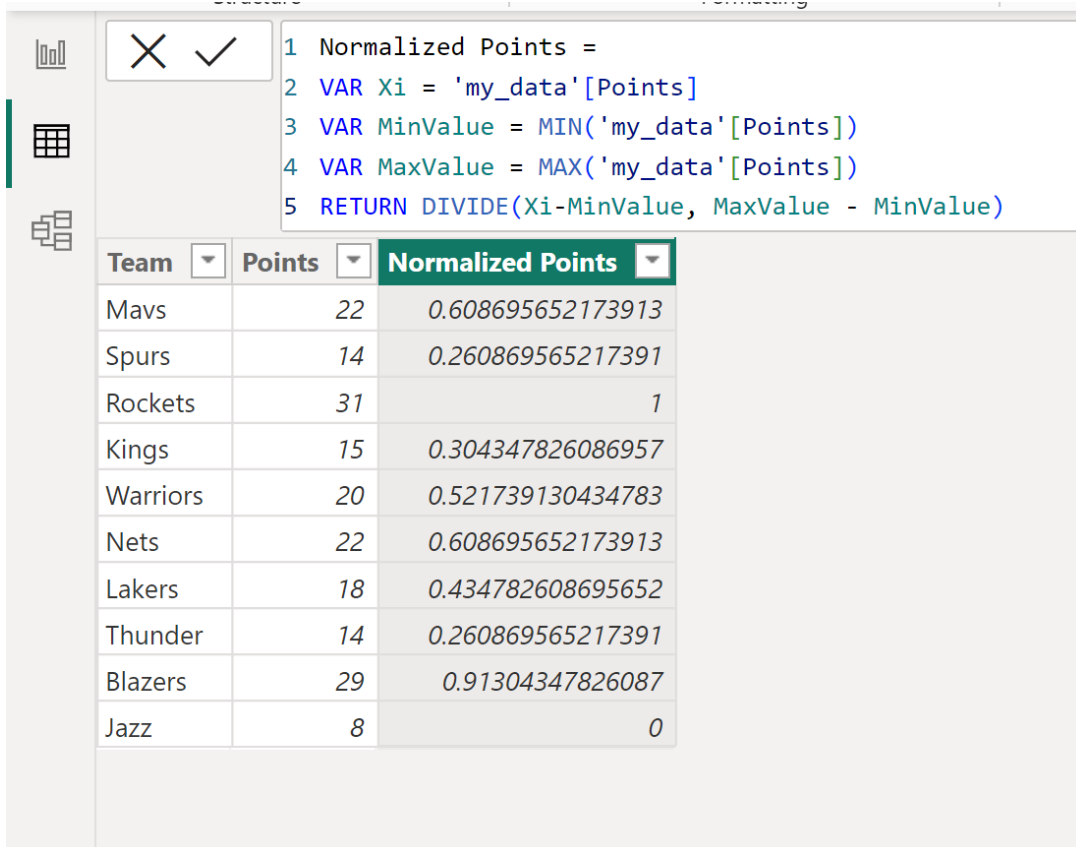
RETURN DIVIDE(Xi - MinValue, MaxValue - MinValue)

Executing this calculation creates the new column, **Normalized Points**, immediately demonstrating the results of the scaling operation alongside the original data. This new column is ready to be used in visualizations, measures, or other analytical steps where standardized input is required.

Verifying the Normalized Results

Upon successful execution of the DAX formula, a new column named **Normalized Points** will be appended to the **my_data** table. This column contains the normalized values, confirming that the [normalization](#) process has been completed.

The result should appear similar to the following visualization:



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)

```

The table below shows the result of this calculation for various teams:

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

A crucial observation here is that every value in the **Normalized Points** column now strictly falls within the range of 0 to 1. Specifically, if we examine the original data, the minimum value in the **Points** column is 8 (which corresponds to 0.00000 in the normalized column), and the maximum value is 31 (which corresponds exactly to 1.00000 in the normalized column).

To ensure full understanding, let us verify the calculation for the specific data point where the original score was 22. In the resulting table, the normalized value for this score is **0.60869**. We can confirm this value by manually applying the Min-Max formula using the identified minimum (8) and maximum (31) scores from the entire dataset:

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

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

$$z_i = 14 / 23$$

$$z_i \approx 0.60869$$

This verification confirms that the [DAX](#) calculation accurately implemented the Min-Max scaling logic. Every other value in the **Normalized Points** column has been calculated in a similar, mathematically consistent manner, providing a standardized basis for future reporting and comparisons within [Power BI](#).

Benefits and Considerations of Min-Max Scaling

The implementation of [Min-Max Scaling](#), as demonstrated in Power BI, offers significant advantages in the realm of business intelligence and analytics. Firstly, it is highly intuitive: the resulting scaled values are easily understood as a percentage of the total observed range. A normalized score of 0.75, for instance, immediately tells the user that the original value was 75% of the way between the minimum and maximum observed data points. This interpretability is invaluable for dashboard design and end-user comprehension.

Secondly, this form of [data transformation](#) is crucial when preparing data for algorithms that rely on distance metrics, such as clustering or certain machine learning models implemented via integrated R or Python scripts in Power BI. By preventing features with large scales from dominating the distance calculations, normalization ensures the model treats all inputs fairly, leading to more accurate and reliable outcomes.

However, it is essential to consider a major limitation of Min-Max Scaling: its sensitivity to outliers. Because the normalization relies strictly on the absolute minimum and maximum values, a single extreme outlier can compress the majority of the data points into a very small range (e.g., between 0 and 0.1). If your dataset is known to contain significant outliers, an alternative approach, such as Z-score standardization (which scales data based on the mean and standard deviation), might be a more robust choice. For datasets without extreme outliers, or when a strict 0-to-1 boundary is mandatory, Min-Max Scaling implemented through DAX is the definitive technique.

Further Data Transformation Resources

Mastering data preparation techniques like normalization is fundamental to becoming proficient in data modeling and business intelligence. While Min-Max scaling is perfect for setting strict boundaries, Power BI and DAX offer a wide array of functions to handle other common data preprocessing needs, including standardization, discretization, and feature engineering.

The following tutorials explain how to perform other common tasks in [Power BI](#), further expanding your capabilities in data analysis and reporting:

How to implement Z-score standardization using DAX.

Techniques for handling missing values in Power BI using Power Query.

Advanced feature engineering methods in DAX for time series analysis.

Calculating percentile ranks for comparative analysis without full [normalization](#).

By continually expanding your knowledge of data transformation, you ensure that the insights derived from your Power BI reports are always based on clean, relevant, and standardized data.