

# How to Easily Add the Mean Value to Your Boxplot

Authored by  
**stats writer**

November 20, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Add the Mean Value to Your Boxplot*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=98514>

## 1. Introduction to Boxplots and Central Tendency

The boxplot, or box-and-whisker plot, stands as a fundamental tool in exploratory data analysis (EDA), offering a standardized way to visualize the distribution of numerical data based on the five-number summary: the minimum, the first quartile (Q1), the median (Q2), the third quartile (Q3), and the maximum. This visualization efficiently conveys the location, spread, and skewness of the data, and identifies potential outliers, all within a compact graphical format. However, while the standard boxplot excels at summarizing distributional shape and resistance, it traditionally emphasizes the median as its primary measure of central tendency.

Central tendency is a crucial concept in statistics, aiming to identify a single value that accurately describes the center point of a distribution. The two most common measures are the mean (average) and the median (midpoint). Data analysts often require both metrics to gain a comprehensive understanding of the dataset. For instance, if the distribution is perfectly symmetrical, the mean and the median will coincide. Deviations between these two values are essential indicators of distribution skewness, informing subsequent modeling decisions or conclusions drawn from the data.

Integrating the mean value directly onto the boxplot visualization resolves the limitation of relying solely on the five-number summary. By displaying both central tendency measures simultaneously, the analyst can instantly assess the relationship between the average and the true midpoint, thereby enriching the interpretive power of the plot. This capability is particularly vital in fields such as quality control, finance, and social sciences, where precise localization of the average value alongside distributional spread is required for reporting and decision-making.

## 2. Understanding the Mean vs. the Median in Data Visualization

The distinction between the mean and the median is statistically profound, and their graphical representation carries significant analytical weight. The mean is calculated by summing all data points and dividing by the total count, making it highly sensitive to extreme values or outliers. Consequently, in datasets that are heavily skewed--such as income distributions where a few extremely high earners exist--the mean can be pulled significantly away from the bulk of the data, potentially misrepresenting the 'typical' value.

In contrast, the median represents the 50th percentile, meaning 50% of the data falls below it and 50% falls above it. Because its calculation relies only on the order of the data points, the median is robust and highly resistant to the influence of outliers. This characteristic is precisely why traditional boxplots default to displaying the median: it provides a stable and reliable measure of the center that is not distorted by unusual data points identified in the whiskers or as separate outliers.

However, there are scenarios where the mean is the mandated metric for comparison, especially in parametric statistical tests or when the underlying theory assumes a normal distribution. In such cases, visualizing the mean alongside the median becomes a necessary visual audit. If the mean marker is noticeably closer to one of the quartiles (Q1 or Q3) than the other, it visually confirms the direction of data skewness. For instance, if the mean marker sits closer to Q3 (the upper quartile) than the median line, it suggests positive or right skewness.

### 3. The Role of the Boxplot's Internal Line

Understanding what each element of the boxplot represents is paramount to accurate interpretation. The rectangular box itself spans the interquartile range (IQR), from the first quartile (Q1) to the third quartile (Q3), encompassing the middle 50% of the data. The length of the box provides an indication of data variability or spread. Within this box, a single horizontal line is consistently drawn; this line is the visual anchor for the central tendency, specifically designating the median (Q2).

This default display choice reinforces the non-parametric nature of the boxplot--a visualization technique that makes no assumption about the underlying distribution of the data. The location of this median line relative to Q1 and Q3 provides further insight into intra-box skewness. If the line is positioned exactly in the center of the box, the central 50% of the data is symmetrically distributed. If the line is closer to Q1, the data is relatively denser on the lower end, and vice versa.

When preparing to add the mean indicator, it is essential to distinguish it visually from this inherent median line. Tools like Seaborn achieve this by using a distinct marker (a triangle by default) instead of a second line. This ensures that viewers can quickly differentiate between the robust midpoint (the line) and the arithmetic average (the marker), facilitating a clear comparison of these critical measures of location.

### 4. Implementing the Mean Indicator using Seaborn

You can utilize the powerful statistical visualization library, Seaborn, which is built upon Matplotlib, to generate highly informative boxplots. To explicitly display the mean value within the graphical representation, you must employ the **showmeans** parameter within the primary `boxplot()` function call. Setting this boolean argument to `True` instructs Seaborn to calculate and plot the average of the data points for each category represented by a box.

```
sns.boxplot(data=df, x='x_var', y='y_var', showmeans=True)
```

This concise syntax is the fundamental mechanism for overlaying the arithmetic mean onto the standard boxplot structure. While the default boxplot provides excellent insight into quartiles and

dispersion, the inclusion of the mean allows for immediate comparison between the central tendency measures, especially critical when evaluating the skewness of a distribution. The forthcoming sections will walk through a complete, runnable Python example demonstrating this functionality in practice.

## 5. Setting up the Data Environment (The Pandas DataFrame)

To illustrate the implementation of the **showmeans** parameter, we first require a suitable dataset structured as a Pandas DataFrame. For this example, we will simulate performance metrics--specifically, points scored--across three distinct competitive teams (Team A, Team B, and Team C). This categorical structure is ideal for visualization using boxplots, as it allows us to compare the distribution characteristics of a quantitative variable (points) across different groups (teams).

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,  
'points': })
```

```
#view head of DataFrame
```

```
print(df.head())
```

```
team points
```

```
0 A 3
```

```
1 A 4
```

```
2 A 6
```

```
3 A 8
```

```
4 A 9
```

The dataset contains fifteen observations, five for each team, providing a manageable but representative sample for our distributional analysis. Preparing the data in this tidy format, where columns represent variables and rows represent observations, is standard practice in Python data analysis pipelines utilizing libraries like Pandas and Seaborn. The structure of this DataFrame--a grouping variable ('team') and a numerical variable ('points')--perfectly aligns with the requirements of the categorical boxplot visualization.

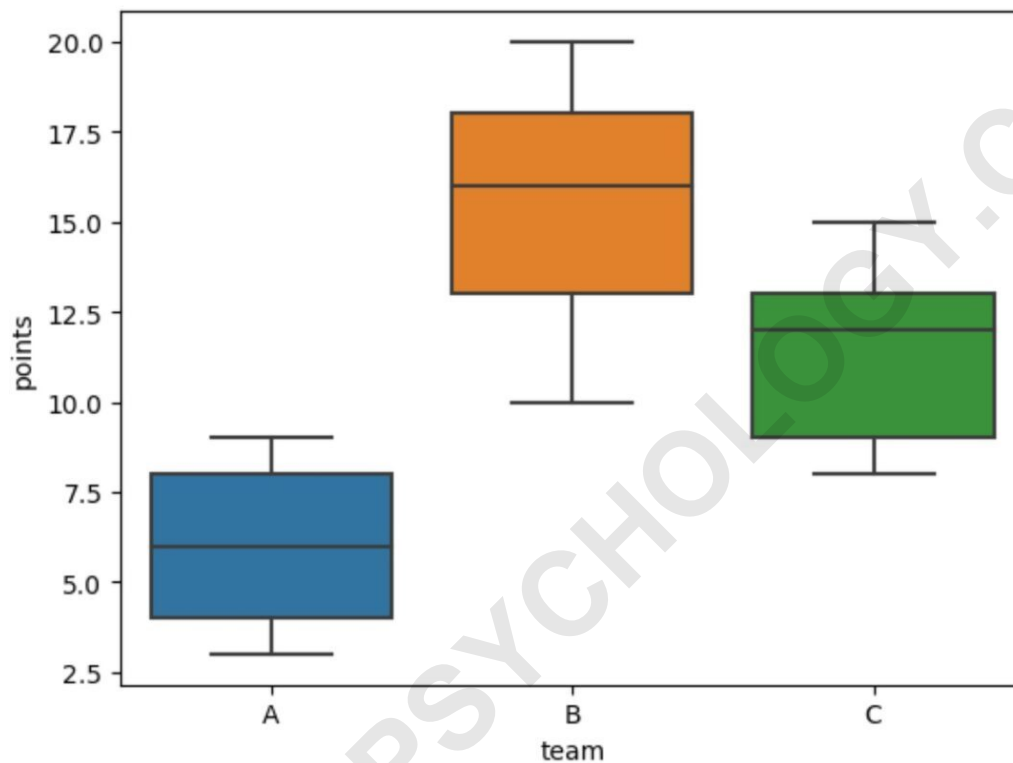
## 6. Step-by-Step Visualization: Default Boxplot (Median Display)

Before we introduce the mean marker, it is instructive to observe the default behavior of the Seaborn boxplot. By default, the visualization is designed to highlight the five-number summary, with its primary focus on robustness against outliers, which is best achieved by centering the box

around the median. This initial step serves as a baseline against which we can compare the subsequent plot that includes the arithmetic average.

### import seaborn as sns

```
#create boxplot to visualize points distribution by team  
sns.boxplot(data=df, x='team', y='points')
```



Upon examining the figure generated by the code above, one can clearly see the horizontal line segment positioned within the rectangular box for each team. This line consistently represents the median value (the 50th percentile) of the distribution of points for that specific team. The median is the preferred measure of central tendency in standard boxplots because it is resistant to extreme values or skewness in the underlying data, making it a reliable indicator of the true center regardless of data symmetry.

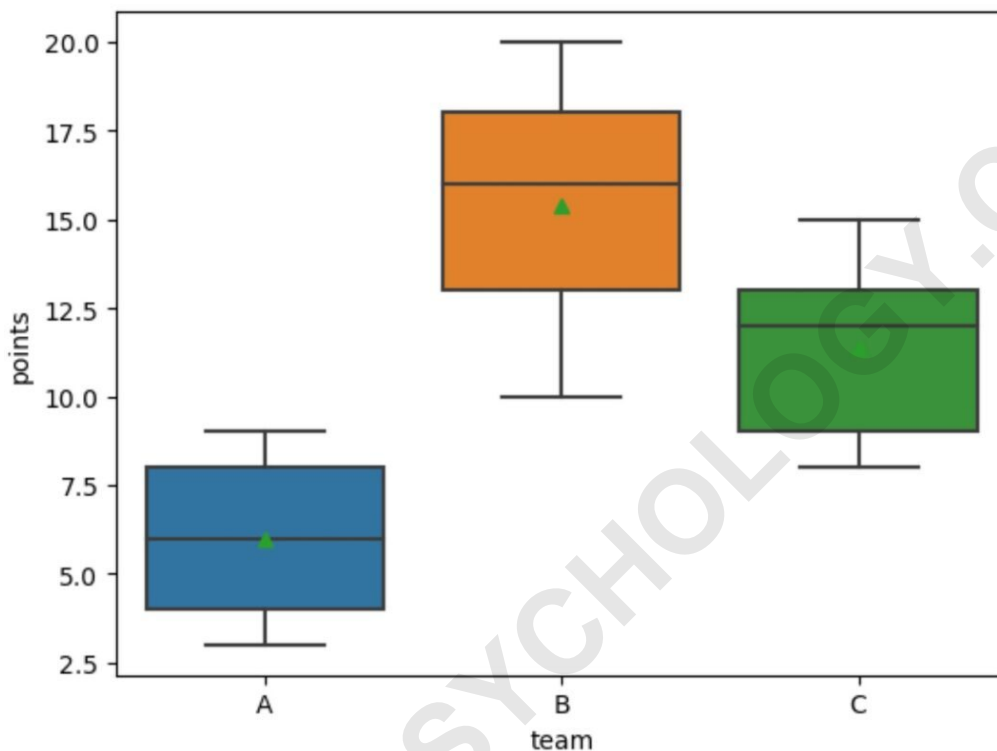
## 7. Displaying the Mean: Utilizing the `showmeans` Argument

To explicitly include the mean alongside the median, we must modify the function call by setting the **showmeans** parameter to `True`. This simple addition instructs Seaborn to calculate the arithmetic average for each group and display it visually on the plot. This is particularly useful when analyzing distributions, as the relative position of the mean versus the median immediately

indicates the direction and severity of skewness.

```
import seaborn as sns
```

```
#create boxplot to visualize points distribution by team (and display mean values)  
sns.boxplot(data=df, x='team', y='points', showmeans=True)
```



As visible in the resulting visualization, Seaborn defaults to using a distinct marker--specifically, a green triangle--to represent the calculated mean value for each respective box. When the green triangle aligns closely with the internal horizontal line, it suggests a relatively symmetric distribution. If the triangle is positioned significantly higher or lower than the line, it alerts the analyst to positive or negative skewness, respectively, highlighting the difference between the average and the midpoint of the data.

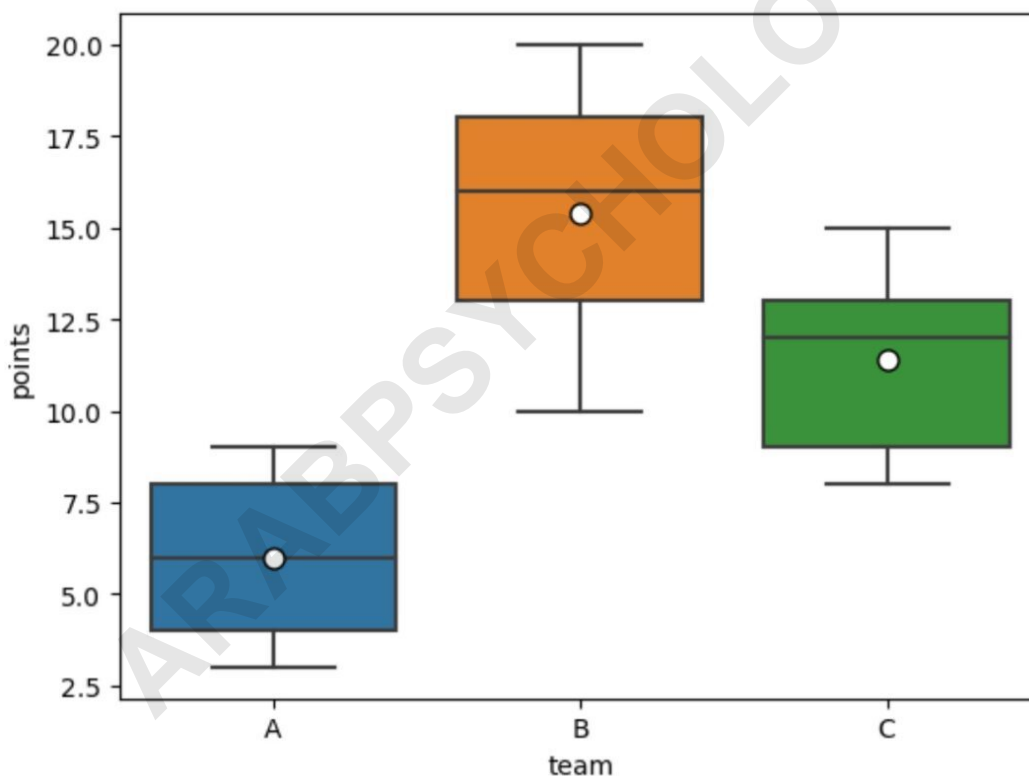
## 8. Advanced Customization: Fine-Tuning the Mean Marker with `meanprops`

While the default green triangle clearly indicates the mean, professional data visualizations often require specific aesthetics to match corporate standards or publication requirements. Seaborn facilitates this customization through the **meanprops** argument. This argument accepts a Python dictionary containing key-value pairs that are passed directly to the underlying Matplotlib scatter function responsible for drawing the mean marker.

The customizable properties include `marker` (the shape of the symbol), `markerfacecolor` (the fill color), `markeredgecolor` (the color of the border), and `markersize` (the scale of the marker). By leveraging **meanprops**, developers gain granular control over the visual presentation, ensuring the mean value stands out or blends in according to the visualization's purpose, thereby increasing clarity and visual appeal.

### import seaborn as sns

```
#create boxplot to visualize points distribution by team
sns.boxplot(data=df, x='team', y='points', showmeans=True,
            meanprops={'marker':'o',
                      'markerfacecolor':'white',
                      'markeredgecolor':'black',
                      'markersize':'8'})
```



The example above demonstrates transitioning from the default green triangle to a clearly defined white circle with a black border, set at a size of 8. This configuration ensures that the mean marker is highly visible against the background colors of the boxplots, significantly enhancing readability and aesthetic quality. Analysts are encouraged to experiment with different marker types, colors, and sizes within the **meanprops** dictionary to achieve optimal visual communication tailored to

their specific data context.

## 9. Conclusion and Best Practices

Incorporating the mean indicator into a boxplot transforms it from a tool focused solely on distribution percentiles and variability into a comprehensive summary of central tendency. By utilizing the **showmeans=True** parameter in Seaborn, and optionally customizing its appearance using **meanprops**, data scientists can quickly compare the arithmetic average with the median, providing immediate insights into data skewness and the influence of potential outliers.

Always consider the audience and the nature of the data when choosing visualization parameters. If the data is known to be highly skewed or contains significant outliers, the median remains the most robust measure of central tendency. However, for comparative statistical analysis or when the distribution is approximately normal, displaying the mean offers valuable, complementary information, ensuring that your visualization tells a complete story about the underlying data distributions. Mastering these subtle visualization techniques is essential for effective data communication in technical and academic environments.

**Note:** You can find the complete documentation for the seaborn boxplot() function on the official website, which details all available parameters and customization options.