How can I remove outliers from a boxplot in Seaborn?

Authored by

stats writer

November 21, 2025

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

stats writer (2025). *How can I remove outliers from a boxplot in Seaborn?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=99185

To effectively manage and visualize data distributions, especially when dealing with anomalies, the Seaborn statistical visualization library provides powerful tools. When generating a boxplot, the presence of extreme values, often termed outliers, can sometimes skew the visual representation or require dedicated filtering for a clearer understanding of the core data spread. The most straightforward method to address this in Seaborn is by utilizing the built-in boxplot function's **showfliers** option.

Setting the **showfliers** parameter to **False** instructs Seaborn to omit any data points that satisfy the statistical definition of an outlier from the resulting plot. This technique is indispensable for generating streamlined visualizations where the primary focus is the interquartile range (IQR) and the overall symmetry of the distribution, rather than the individual extreme values themselves. Understanding this option is fundamental for data analysts seeking clean, publication-ready graphics.

Understanding Boxplots and the Definition of Outliers

A boxplot, or box-and-whisker plot, is a standardized way of displaying the distribution of data based on a five-number summary: the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. This visualization is crucial in exploratory data analysis (EDA) because it quickly highlights central tendency, spread, and symmetry, which are often obscured in simpler charts like histograms.

Statistically, outliers are defined as observations that fall outside the whiskers of the plot. Specifically, an observation is typically classified as an outlier if it is 1.5 times the Interquartile Range (IQR = Q3 - Q1) above the third quartile or below the first quartile. While these points are statistically relevant and should not be discarded lightly during analysis, they often represent extreme cases or potential data collection errors. When visualizing data for presentation, removing these markers can significantly clean up the graph, drawing attention to the main body of the distribution.

The decision to exclude outliers from the visualization is a methodological choice that depends heavily on the analytical objective. If the goal is to show the raw, unfiltered data, outliers must be included. However, if the goal is to compare the core distributions of several groups--say, comparing the average performance of different teams--removing visual noise caused by extreme, infrequent scores ensures a clearer, more focused comparison of the central tendencies and variability between those groups.

Implementing the showfliers Parameter in Seaborn

The Seaborn library, built upon Matplotlib, offers extensive customization for statistical graphics. The **sns.boxplot()** function manages the drawing of the plot, and the key to outlier control lies

ARABPSYCHOLOGY.COM

within its parameters. The parameter **showfliers** is specifically designed to toggle the visibility of these extreme observations.

By default, **showfliers** is set to **True**, meaning Seaborn calculates and plots all observations classified as outliers, usually marking them with small diamond shapes. To suppress this behavior and remove outlier points entirely from the visualization, we must explicitly set this argument to **False** during the function call. This modification ensures that the resulting plot only displays the box (containing the IQR and median) and the whiskers (extending to the farthest non-outlier data points).

The syntax for implementing this removal is concise and is applied directly within the plot generation command, regardless of whether the data is univariate or multivariate (grouped boxplots). This streamlined approach makes it easy for practitioners to quickly iterate between views showing and hiding extreme values, aiding in robust data interpretation. When creating a boxplot in Seaborn, you can use the argument **showfliers=False** to remove outlier observations from the plot:

sns.boxplot(x='variable', y='value', data=df, showfliers=False)

Fine-Tuning Outlier Visualization using fliersize

While complete removal of outliers is often desirable, there are scenarios where analysts prefer to keep the outliers visible but minimize their visual impact. For these cases, Seaborn provides the **fliersize** argument. This parameter controls the size of the markers used to represent the outliers on the plot. By adjusting this value, one can make the outlier points extremely small or, conversely, highly prominent.

The default size for **fliersize** is typically **5** (in points). Reducing this value, for instance to 2 or 3, allows the outliers to remain on the plot, confirming their existence, but reduces their dominance relative to the box and whisker structure. This serves as an excellent intermediate option between full visibility and complete suppression.

If you'd like to simply change the size of the outlier markers without removing them, you can use the **fliersize** argument:

sns.boxplot(x='variable', y='value', data=df, fliersize=3)

Note that the default size for **fliersize** is **5**. Choosing the right size often requires experimentation to balance clarity with the need to acknowledge extreme observations.

ARABPSYCHOLOGY.COM

Practical Example: Data Preparation using Pandas

To illustrate these concepts, we will use a common data structure scenario involving comparisons across different categories. Suppose we have a Pandas DataFrame that details the points scored by players across three distinct basketball teams (A, B, and C). Since Seaborn prefers data in a 'long' format for categorical comparisons, we will use the **pd.melt()** function to transform the wide format data into a stacked format, suitable for plotting.

The process begins by importing necessary libraries and defining our initial dataset. Notice that we intentionally include very high values (78 and 60) in teams A and C to ensure the generation of clear outliers in the subsequent visualizations. This synthetic data provides a perfect test case for demonstrating the effects of **showfliers=False**.

The following Python code initializes the data structure and transforms it into the required long format:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'A': ,
'B': ,
'C': })

#melt data frame into long format
df_melted = pd.melt(df)

#view head of DataFrame
print(df_melted.head())

variable value
0 A 5
1 A 7
2 A 7
3 A 19
4 A 22

The resulting **df_melted** dataframe is now ready for visualization, featuring a 'variable' column indicating the team (A, B, or C) and a 'value' column containing the scores.

Default Visualization: Identifying Outliers

ARABPSYCHOLOGY.COM

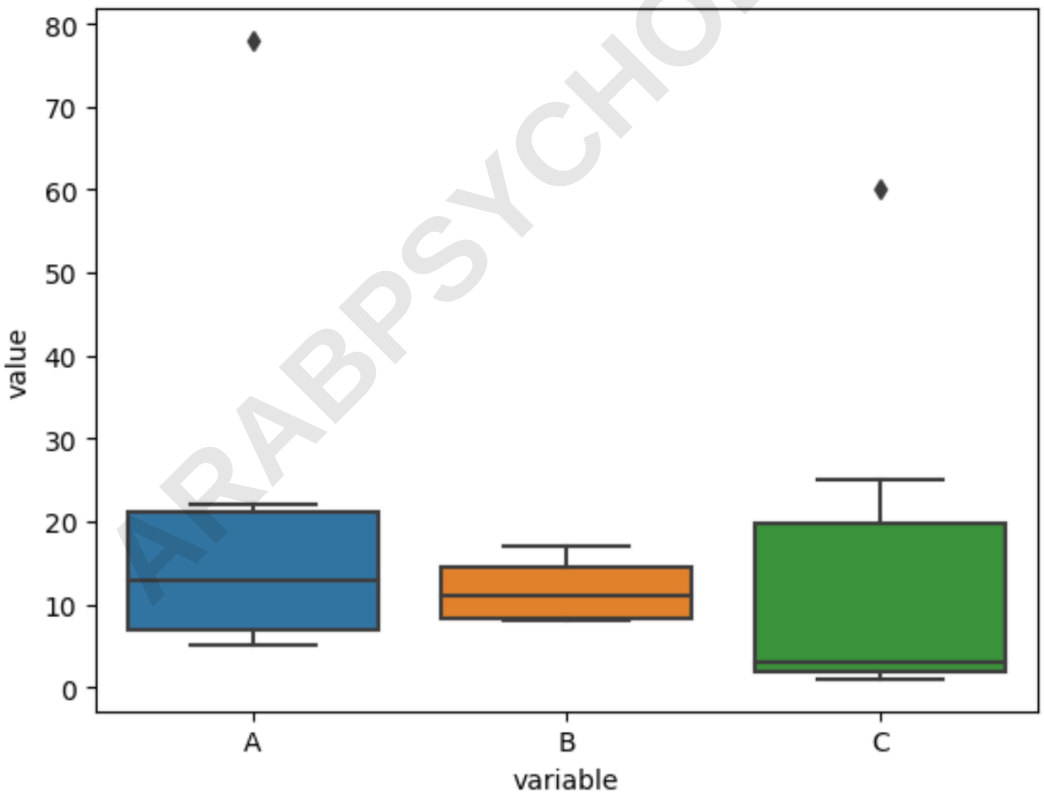
Before modifying the plot, it is essential to establish a baseline visualization that includes all data points, specifically the calculated outliers. We utilize the standard **sns.boxplot()** function call, mapping the team variable to the x-axis and the score value to the y-axis. Since we omit the **showfliers** argument, the default behavior is maintained.

This initial plot allows us to visually confirm which data points Seaborn statistically identifies as outliers. In the context of our basketball score example, teams A and C, due to their single very high scores (78 and 60 respectively), are expected to display distinct outlier markers far above their respective boxes, while Team B, which has a tight distribution, might show none.

We use the following syntax to create three boxplots that show the distribution of points scored by each team:

import seaborn as sns

#create boxplot to visualize distribution of points by team
sns.boxplot(x='variable', y='value', data=df_melted)



As anticipated, by default, Seaborn uses diamond-shaped markers to display outlier observations, clearly visible here for groups A and C. These points often distort the overall scale of the y-axis, making subtle differences between the core distributions of A, B, and C harder to discern.

ARABPSYCHOLOGY.COM

Removing Outliers for a Focused Visualization

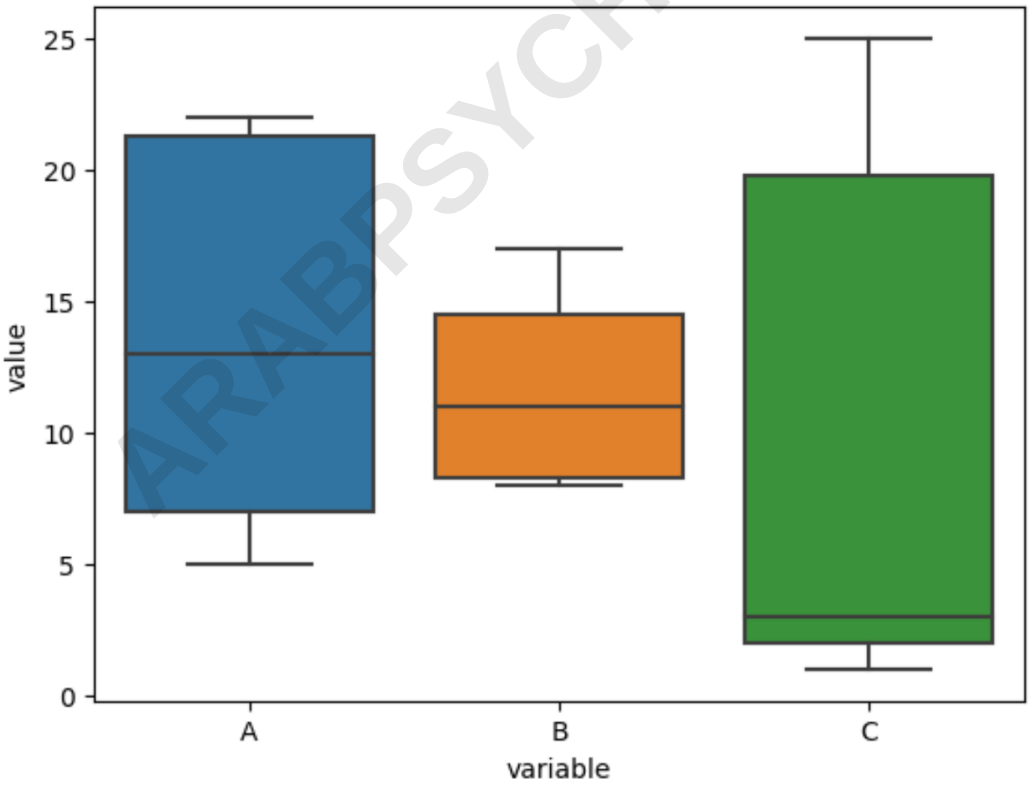
To achieve a cleaner visual representation that focuses solely on the bulk of the data distribution, we now apply the key technique: setting **showfliers=False**. This adjustment tells the **boxplot()** function to ignore those extreme points during rendering. Crucially, while the visualization omits them, the underlying statistical definition of the box and whiskers (Q1, Median, Q3, and the non-outlier maximum/minimum) remains based on the full dataset.

The resulting plot will still accurately represent the quartiles but will truncate the whiskers at the nearest non-outlier observation, effectively zooming in on the central distribution. This is particularly useful when presenting data to a non-technical audience who might be distracted by extreme values that are not central to the primary analytical conclusion.

To remove these outliers from the plot, we use the argument **showfliers=False** in the function call:

import seaborn as sns

#create boxplots and remove outliers
sns.boxplot(x='variable', y='value', data=df_melted, showfliers=False)



ARABPSYCHOLOGY.COM

Notice that the outliers have been removed from the plot entirely. The whiskers now terminate at

the highest and lowest scores that are not deemed statistical outliers, providing a much tighter focus on the typical scoring range for each team.

Adjusting Visual Weight: Minimizing Outlier Markers

As an alternative to complete removal, if the analyst wishes to retain the outlier information while minimizing its visual weight, the **fliersize** parameter is applied. By setting this value significantly lower than the default 5, the diamond markers become less conspicuous on the visualization, achieving a balance between data completeness and visual clarity.

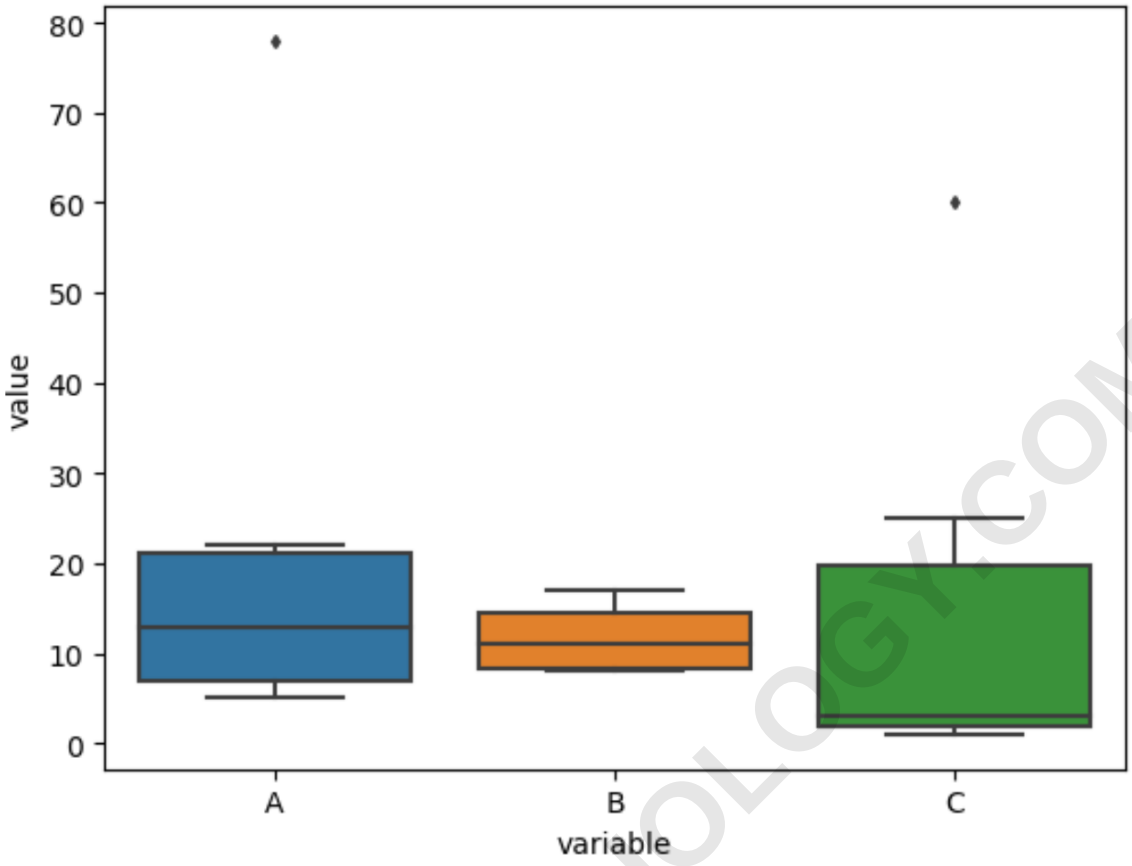
For example, setting **fliersize=3** makes the markers half the size of the default **fliersize=5**. This technique ensures that if an outlier is present, it is still plotted, satisfying statistical rigor, but it avoids overwhelming the visualization, which is particularly useful when comparing many distributions simultaneously.

Here is the syntax for adjusting the marker size:

import seaborn as sns

#create boxplots and adjust markers for outliers to be smaller
sns.boxplot(x='variable', y='value', data=df_melted, fliersize=3)

ARABPSYCHOLOGY.COM



ARABPSYCHOLOGY.COM

Notice that the marker sizes for the outliers are much smaller compared to the initial default plot. Analysts are free to adjust the numerical value for the **fliersize** to make the markers as small or large as appropriate for their specific presentation needs.

Summary of Customization Options

Controlling the display of outliers in Seaborn boxplots offers flexibility in how data distributions are communicated. The choice between removing outliers, shrinking them, or retaining the default size should be guided by the audience and the analytical purpose of the visualization.

The two primary methods discussed provide distinct levels of control:

Complete Removal: Achieved by setting **showfliers=False**. This results in the cleanest possible visualization of the core data distribution, ideal for comparing central tendencies without distraction.

Size Adjustment: Achieved by setting **fliersize** to a small integer (e.g., 2 or 3). This is a compromise, allowing the existence of extreme values to be recorded visually while minimizing their dominance on the plot scale.

For advanced customization and a full list of parameters, including options for color, line style, and

statistical aggregation, analysts should consult the official Seaborn documentation for the **seaborn.boxplot()** function. Mastering these parameters is key to creating insightful and high-quality statistical graphics in Python.

The following tutorials explain how to create other common visualizations in Seaborn:

ARABPSYCHOLOGY.COM