

# How to Easily Place Your Seaborn Plot Legend Outside the Plot

Authored by  
**stats writer**

December 6, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Place Your Seaborn Plot Legend Outside the Plot*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106179>

As data visualization professionals, we frequently rely on the robust capabilities of libraries like [Seaborn](#) and [Matplotlib](#) to create insightful statistical graphics. A common challenge arises when the default legend placement obscures critical data points or unnecessarily shrinks the plotting area. To solve this, we must strategically position the legend outside of the main axes using the powerful `bbox_to_anchor()` argument.

This technique grants precise control over the legend's location relative to the plot boundaries. By defining specific coordinate pairs, developers can ensure that the legend enhances readability without compromising the visualization itself. This comprehensive guide will detail the mechanism behind external legend placement and provide practical, executable Python examples demonstrating various anchor points.

## The Mechanism: Using `bbox_to_anchor` for External Placement

The ability to position the legend externally is facilitated through the Matplotlib function `plt.legend()`, which [Seaborn](#) utilizes internally. Specifically, the argument `bbox_to_anchor()` is key. This argument accepts a tuple of coordinates, usually (x, y), which defines a fixed location in relation to the axes bounding box where the legend box should be anchored.

When aiming to place the legend outside the plot area, the coordinates provided to `bbox_to_anchor()` must extend beyond the standard plot boundaries, which range from (0, 0) in the bottom-left corner to (1, 1) in the top-right corner. For instance, an x-coordinate greater than 1.0 (e.g., 1.05) will move the anchor point slightly to the right of the plot's boundary, ensuring the legend is drawn in the margin space.

For example, the following syntax is typically employed to position the legend slightly outside the top-right corner of the plot area:

```
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
```

In this configuration, the tuple **(1.05, 1)** specifies the (x, y) coordinates for the anchor point. The **loc** parameter, set to `'upper left'`, determines which part of the legend box should be aligned with that anchor point. By setting the legend's upper-left corner to the coordinate (1.05, 1), the entire legend box shifts outward, beginning just beyond the right edge (x=1.0) and exactly at the top edge (y=1.0).

## Understanding the Coordinate System and Parameters

To effectively manipulate legend placement, a solid understanding of the coordinate system used by `bbox_to_anchor()` is essential. By default, these coordinates are normalized relative to the axes bounding box. A value of 0.0 represents the bottom or left edge, and 1.0 represents the top or

right edge. Thus, coordinates outside the range push the anchor point outside the plot.

The **loc** parameter is critically important, as it dictates the specific point on the legend box that will be placed at the defined **bbox\_to\_anchor** coordinates. Common values include `'upper left'`, `'center right'`, or `'lower right'`. Choosing the right combination of **bbox\_to\_anchor** coordinates and **loc** values is necessary to achieve the desired orientation and prevent overlap. For instance, if you anchor the legend at (1.05, 1.0) and use `loc='upper left'`, the legend will extend downward and inward from that anchor point.

Furthermore, the argument **borderaxespadd** controls the amount of padding between the axes and the border of the legend. Setting **borderaxespadd** to 0 (as shown in the examples) ensures that the legend is placed immediately adjacent to the defined coordinate, eliminating any default whitespace padding that might otherwise be introduced. Understanding how these three parameters--**bbox\_to\_anchor**, **loc**, and **borderaxespadd**--interact is fundamental for advanced plot customization in [Matplotlib](#) visualizations.

## Practical Application: Placing the Legend Outside the Top Right Corner

We begin with a practical demonstration using a sample dataset generated with the [Pandas](#) library. This setup ensures a clear context for visualizing the effect of the legend positioning parameters. We will create a simple [scatterplot](#) where points are colored based on a categorical variable (`'team'`), requiring a descriptive legend.

The objective for this first example is to shift the legend entirely outside the plotting area, aligning it neatly in the upper-right quadrant of the figure space. This position is often preferred because it utilizes the whitespace typically available to the right of the plot, maximizing the data display area.

The following code snippet demonstrates the initialization of the data, the creation of the [Seaborn scatterplot](#), and the precise command sequence required to position the legend using a minimal offset:

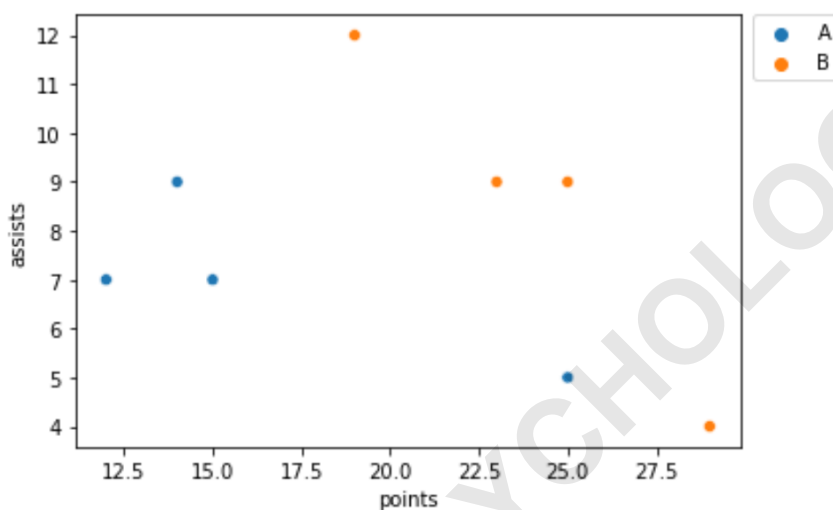
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#create fake data
df = pd.DataFrame({'points': ,
'assists': ,
'team': })
```

```
#create scatterplot
```

```
sns.scatterplot(data=df, x='points', y='assists', hue='team')  
  
#place legend outside top right corner of plot  
plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left', borderaxespad=0)
```

In this specific instance, we use `bbox_to_anchor=(1.02, 1)`. The x-coordinate of 1.02 places the anchor point just 2% of the axes width beyond the right edge, providing a slight separation. The y-coordinate of 1.0 aligns the anchor point precisely with the top edge. Coupled with `loc='upper left'`, the legend's top-left corner adheres to this anchor, causing the legend box to fall perfectly outside the upper-right corner of the plot.



## Adjusting Location: Centering the Legend Along the Right Border

While anchoring the legend in the top corner is effective, sometimes a more central vertical alignment is aesthetically preferable, especially when the plot is taller or when the legend contains many entries. To achieve vertical centering along the right border, we maintain an x-coordinate slightly greater than 1.0 but adjust the y-coordinate closer to 0.5 (the vertical midpoint of the axes).

However, simply setting the y-coordinate to 0.5 when using `loc='upper left'` would result in the legend starting its downward draw from the midpoint, meaning the legend would be centered above the midpoint, not centered overall. To truly center the entire legend box vertically, we must use `loc='center left'` or estimate the necessary y-offset.

In this demonstration, we slightly adjust the y-coordinate to account for the size of the legend box, ensuring that the visual center of the legend aligns approximately with the vertical center of the plot ( $y=0.5$ ). We will continue to use `loc='upper left'`, making a small empirical adjustment to the y-value:

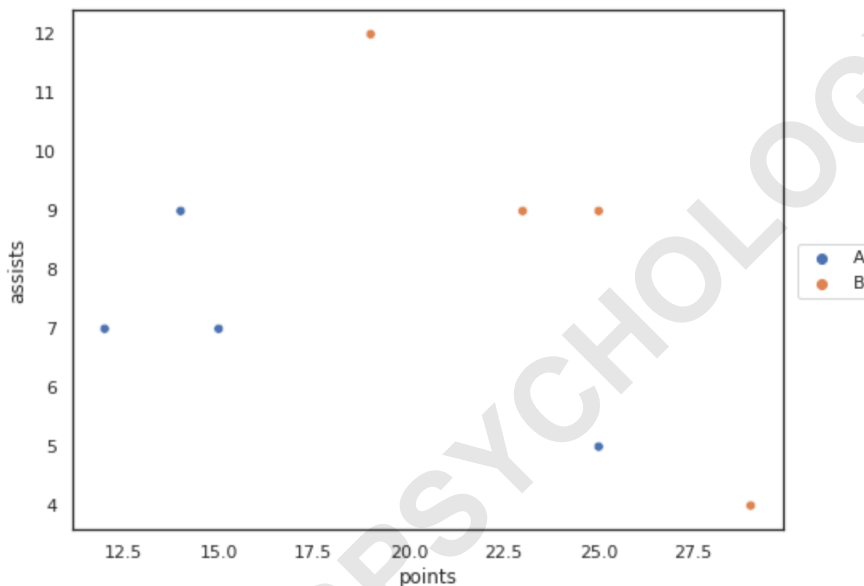
### #create scatterplot

```
sns.scatterplot(data=df, x='points', y='assists', hue='team')
```

```
#place legend outside center right border of plot
```

```
plt.legend(bbox_to_anchor=(1.02, 0.55), loc='upper left', borderaxespad=0)
```

By setting `bbox_to_anchor=(1.02, 0.55)`, we position the top-left corner of the legend box at a point slightly above the vertical midpoint. This slight upward shift compensates for the height of the legend itself, resulting in an appearance that is visually centered along the right edge of the plot. This illustrates the iterative nature of using `bbox_to_anchor()`; precise placement often involves minor trial-and-error adjustments based on the size of the generated legend.



### Fine-Tuning Placement: Anchoring to the Bottom Right Corner

Another frequent requirement is placing the legend near the bottom edge of the plot, often utilized when there is available space below the figure or when the upper area is reserved for annotations or titles. To achieve placement outside the bottom-right corner, we must set the y-coordinate close to 0, ensuring the legend's lowest point aligns near the bottom of the axes.

When aiming for the bottom-right corner, utilizing `loc='upper left'` means the legend will extend downward from its anchor point. Therefore, if we set the y-coordinate too low (e.g., 0.0), the legend will partially or fully disappear below the figure boundary. We must raise the y-coordinate enough so that the legend remains entirely visible, while still anchoring near the bottom margin.

In this final positioning example, we select coordinates that place the anchor point slightly above

the bottom axis, allowing the legend to be fully displayed outside the plot area along the right edge:

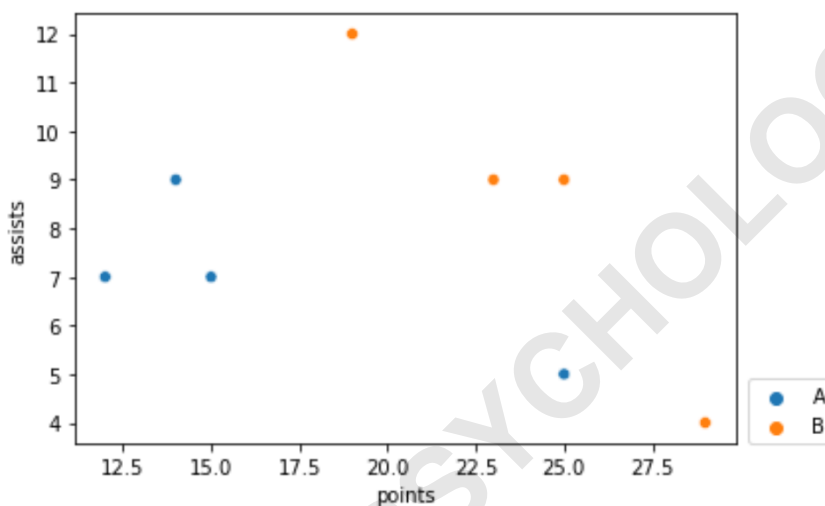
### #create scatterplot

```
sns.scatterplot(data=df, x='points', y='assists', hue='team')
```

```
#place legend outside bottom right corner of plot
```

```
plt.legend(bbox_to_anchor=(1.02, 0.15), loc='upper left', borderaxespad=0)
```

Here, the combination of `bbox_to_anchor=(1.02, 0.15)` and `loc='upper left'` ensures that the legend begins its draw at 15% of the plot height from the bottom, resulting in a low placement adjacent to the bottom-right corner. This demonstrates how minor adjustments to the y-coordinate can drastically change the vertical alignment when using a fixed `loc` value.



## Summary of Best Practices for Legend Control

Placing a legend outside a Seaborn plot is achieved by leveraging the underlying Matplotlib functionality, primarily through the `bbox_to_anchor()` parameter. Mastery of this technique requires understanding the interplay between the relative coordinate system (0 to 1 for plot boundaries) and the `loc` parameter (which determines the alignment of the legend box relative to the anchor point).

When customizing visualizations, always remember the following key rules for using `bbox_to_anchor()`:

**External Placement:** Use values greater than 1.0 (for the x-axis) or less than 0.0 (less common, for left/bottom placement) to push the anchor point outside the axes boundaries.

**Coordinate Relativity:** The (x, y) values are relative to the size of the plotting axes, not the entire

figure.

**Alignment Control:** Adjust the **loc** parameter (e.g., 'upper left', 'lower right') to dictate which corner of the legend box lands on the specified anchor point.

**Padding:** Set **borderaxespad=0** to eliminate default padding between the legend and the axes border, allowing for tighter control over separation.

For a detailed explanation of the arguments and options available for full legend customization, refer to the [official Matplotlib documentation](#). Further details on styling the overall aesthetics of the plot, including color palettes and font sizes, can be found in the comprehensive [Seaborn tutorial on plot aesthetics](#).

[How to Change Legend Font Size in a Seaborn Plot](#)

ARABPSYCHOLOGY.COM