

# How to Easily Create a Custom Legend in Matplotlib

Authored by  
**stats writer**

November 24, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Create a Custom Legend in Matplotlib*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100444>

Creating a manual legend in Matplotlib is a simple process. First, create a plot using the desired markers and lines for each dataset in the plot. Next, use the `pyplot.legend()` method to create the legend and specify the desired location. Finally, add labels to each plot element using the `label` argument. An example of creating a legend in Matplotlib is the following code, which adds a legend to a line plot with three different line styles: `plt.plot(x, y1, label='Line 1')` `plt.plot(x, y2, label='Line 2')` `plt.plot(x, y3, label='Line 3')` `plt.legend(loc='upper left')` `plt.show()`

You can use functions from the **matplotlib.lines** and **matplotlib.patches** sub-modules to create a manual legend in a matplotlib plot.

The following example shows how to do so.

## Example: Create a Manual Legend in Matplotlib

The following code shows how to create a scatter plot in matplotlib with a default legend:

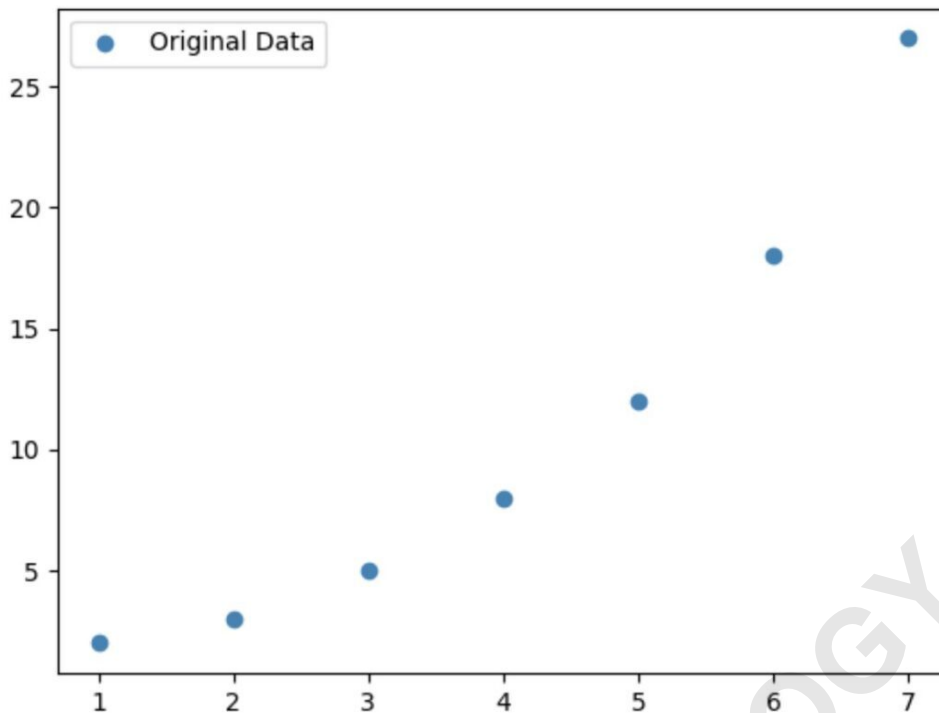
```
import matplotlib.pyplot as plt

#define data to plot
x =
y =

#create scatter plot of x vs. y
plt.scatter(x, y, label='Original Data', color='steelblue')

#add legend
plt.legend()

#display plot
plt.show()
```



To create a manual legend with custom lines and squares, we need to import the **matplotlib.lines** and **matplotlib.patches** sub-modules.

The following code shows how to use these sub-modules to create a manual legend:

```
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import matplotlib.patches as mpatches

#define data to plot
x =
y =

#create scatter plot of x vs. y
plt.scatter(x, y, label='Original Data', color='steelblue')

#define handles and labels that will get added to legend
handles, labels = plt.gca().get_legend_handles_labels()

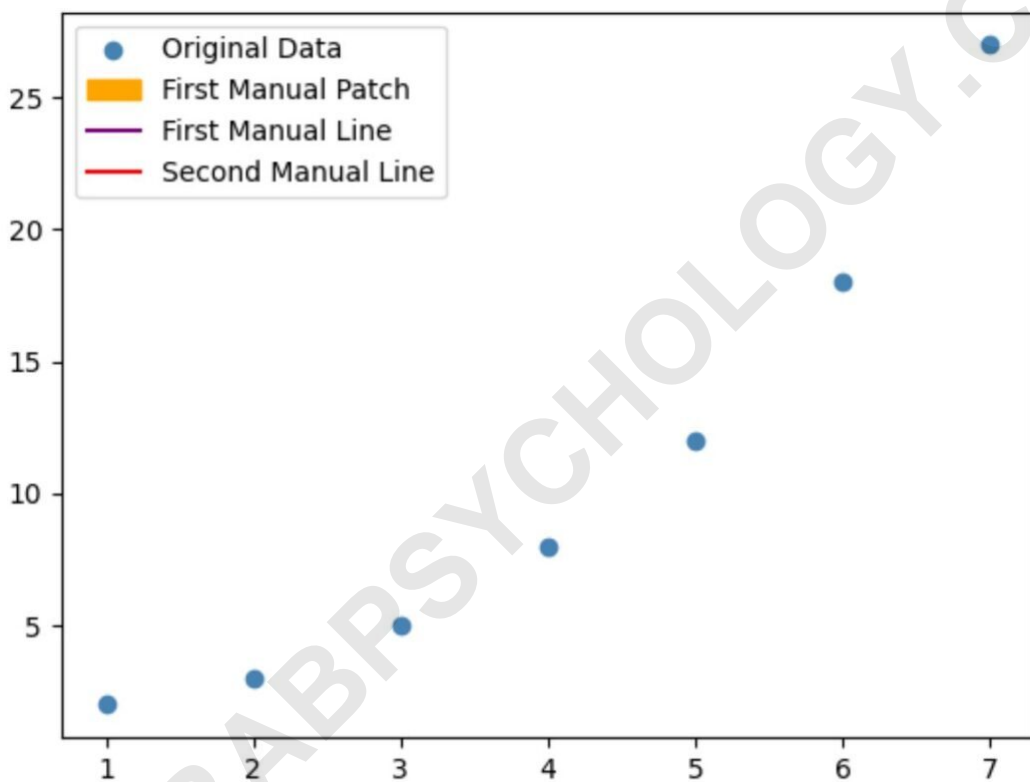
#define patches and lines to add to legend
patch1 = mpatches.Patch(color='orange', label='First Manual Patch')
patch2 = mpatches.Patch(color='orange', label='First Manual Patch')
line1 = Line2D(, , label='First Manual Line', color='purple')
```

```
line2 = Line2D( , , label='Second Manual Line', color='red')

#add handles
handles.extend()

#add legend
plt.legend(handles=handles)

#display plot
plt.show()
```



Notice that this legend includes the label for the original data but also includes labels and shapes for items we added manually.

To change the labels or colors of any of the items, simply modify the values for the **label** and **color** arguments in the previous chunk of code.

**Note:** Refer to to learn how to change the position of the legend within the plot.

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