How can I manually create a legend in Matplotlib with an example?”

Authored by

stats writer

June 27, 2024

The process of manually creating a legend in Matplotlib involves adding a graphical representation of the data elements in a plot, along with their corresponding labels. This can be done by using the "legend" function in Matplotlib, which allows for customization of the legend's position, labels, and style. An example of manually creating a legend in Matplotlib would be adding a legend to a scatter plot, where each point represents a different category and is labeled accordingly in the legend.

Create a Manual Legend in Matplotlib (With Example)

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.pyplotas plt

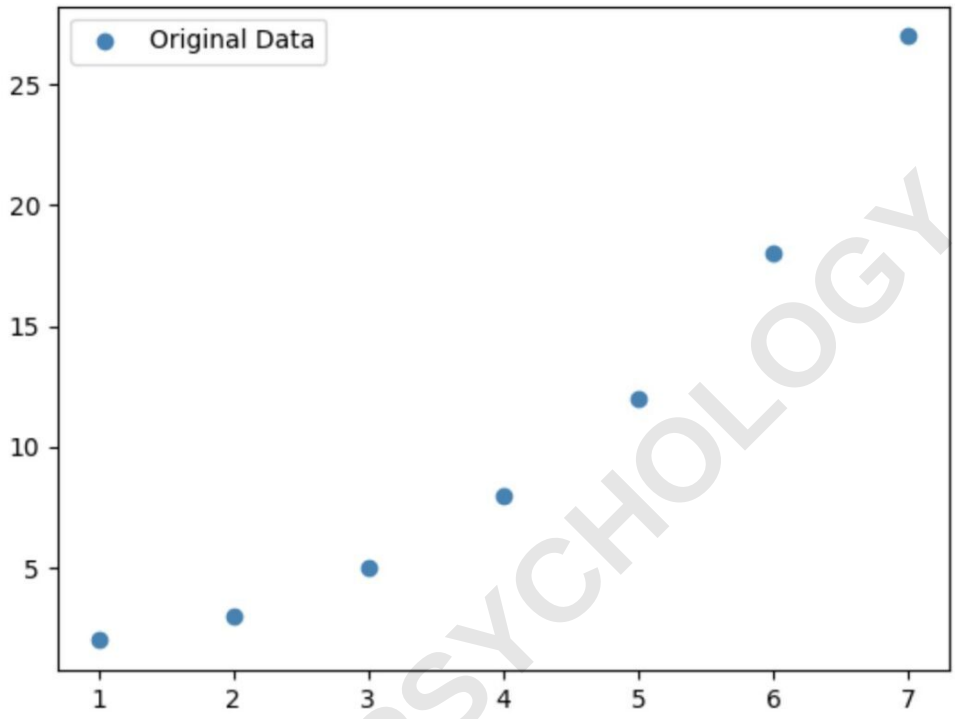
#define data to plot
x =
y =

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

#add legend

plt.legend()

#display plot
plt.show()



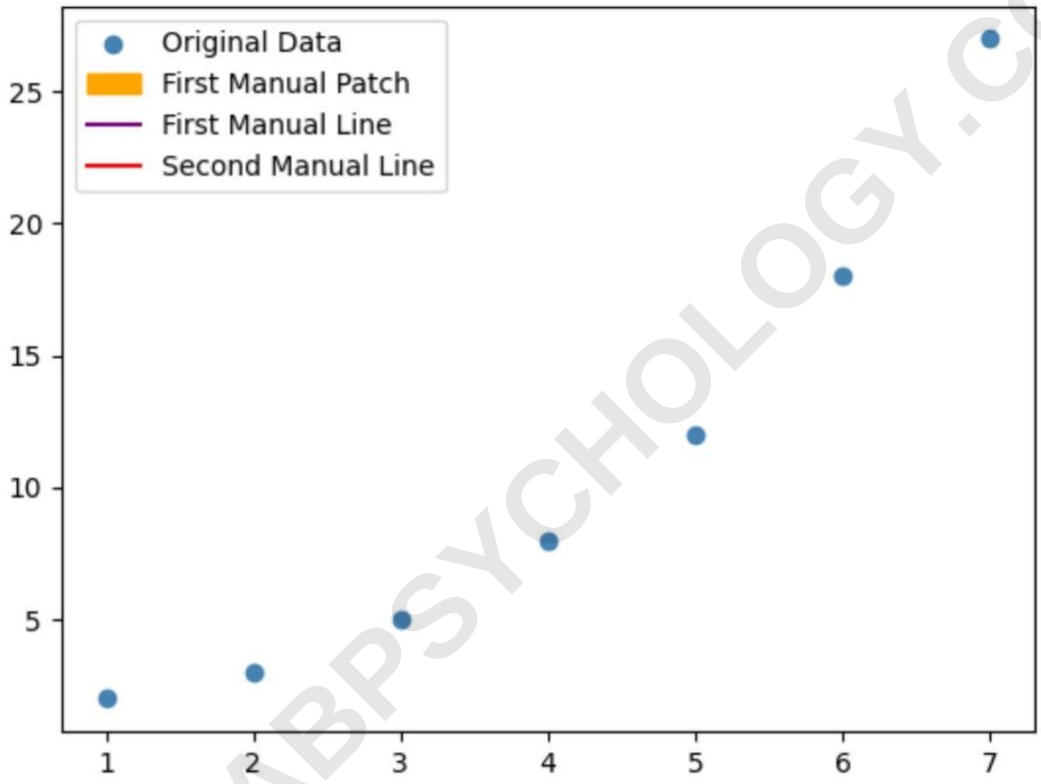
To create a manual legend with custom lines and squares, we need to import the matplotlib.lines and matplotlib.patches sub-modules.

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

import matplotlib.pyplotas plt

```
from matplotlib.linesimport Line2D
import matplotlib.patchesas 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: