How can I add a legend to a scatterplot using Matplotlib?

Authored by

stats writer

May 3, 2024

Adding a legend to a scatterplot using Matplotlib is a simple process that allows for clear and concise communication of the data being presented. A legend is a visual aid that explains the meaning of the different elements or variables in a plot. To add a legend to a scatterplot using Matplotlib, one can utilize the "legend" function and specify the labels and associated colors for each data point. This will create a visual key that helps viewers understand the relationship between the plotted data points and their respective labels. By incorporating a legend, the scatterplot becomes more informative and easier to interpret, making it a valuable tool in data visualization.

Add Legend to Scatterplot in Matplotlib

You can use the following syntax to add a legend to a scatterplot in Matplotlib:

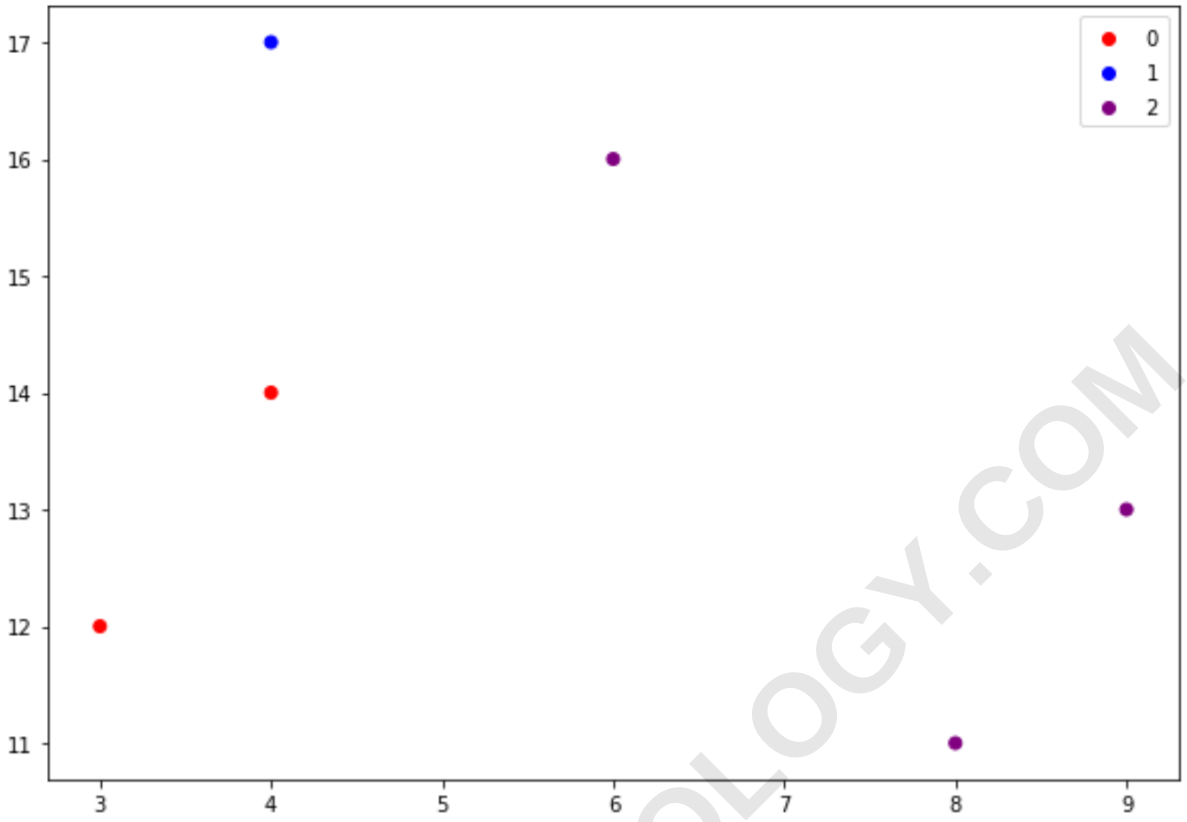
```
import matplotlib.pyplotas plt
from matplotlib.colorsimport ListedColormap

#define values, classes, and colors to map
values =
classes =
colors = ListedColormap()

#create scatterplot
scatter = plt.scatter(x, y, c=values, cmap=colors)

#add legend
plt.legend(*scatter.legend_elements())
```

The following examples show how to use this syntax in practice.

Example 1: Scatterplot Legend with Values

The following example shows how to create a scatterplot where the legend displays values:

```
import matplotlib.pyplotas plt
from matplotlib.colorsimport ListedColormap

#define data
x =
y =

#define values, classes, and colors to map
values =
classes =
colors = ListedColormap()

#create scatterplot
scatter = plt.scatter(x, y, c=values, cmap=colors)

#add legend with values
plt.legend(*scatter.legend_elements())
```

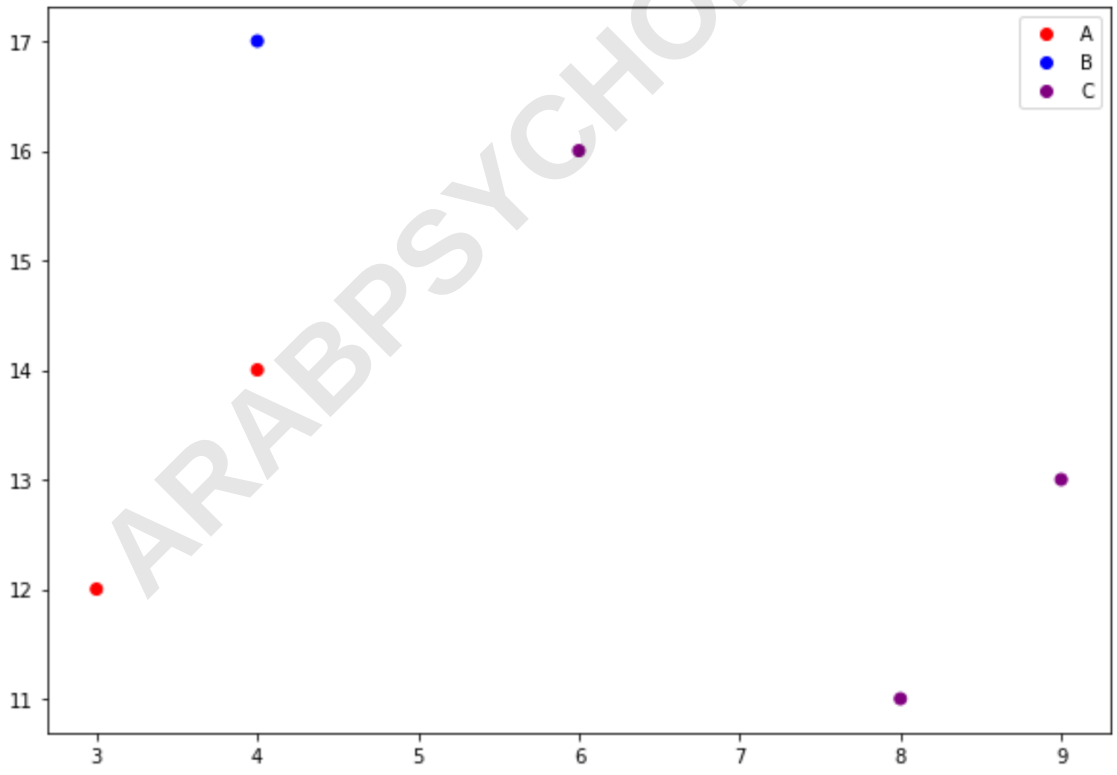


Example 2: Scatterplot Legend with Classes

The following example shows how to create a scatterplot where the legend displays class names:

```
import matplotlib.pyplotas plt
from matplotlib.colorsimport ListedColormap

#define data
x =
y =

#define values, classes, and colors to map
```

```
values =
classes =
colors = ListedColormap()

#create scatterplot
scatter = plt.scatter(x, y, c=values, cmap=colors)

#add legend with class names
plt.legend(handles=scatter.legend_elements(),
labels=classes)
```



Notice that this legend displays the class names that we

specified (A, B, C) as opposed to the values (0, 1, 2) that we specified.