# What are the benefits of using annotations in Matplotlib scatterplots?

Authored by

**stats writer**

April 19, 2024

Annotations in Matplotlib scatterplots provide a way to enhance the visual representation of data by adding additional information or highlighting specific data points. They can be used to label points with their corresponding values, add titles or descriptions to the plot, or even draw attention to outliers or important data points. This makes the scatterplot more informative and easier to interpret for the viewer. Additionally, annotations allow for customization of the plot by changing the color, size, or style of the annotation text and marker. This helps to improve the overall aesthetic and readability of the scatterplot. Overall, annotations in Matplotlib scatterplots provide various benefits such as improving data understanding, enhancing visual appeal, and allowing for customization, making them a valuable tool for data visualization.

## Annotate Matplotlib Scatterplots

## You can use the following basic syntax to annotate scatter plots in Matplotlib:

```
#add 'my text' at (x, y) coordinates = (6, 9.5)
plt.text(6, 9.5, 'my text')
```

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

### Create Basic Scatterplot

## The following code shows how to create a basic scatterplot using Matplotlib:

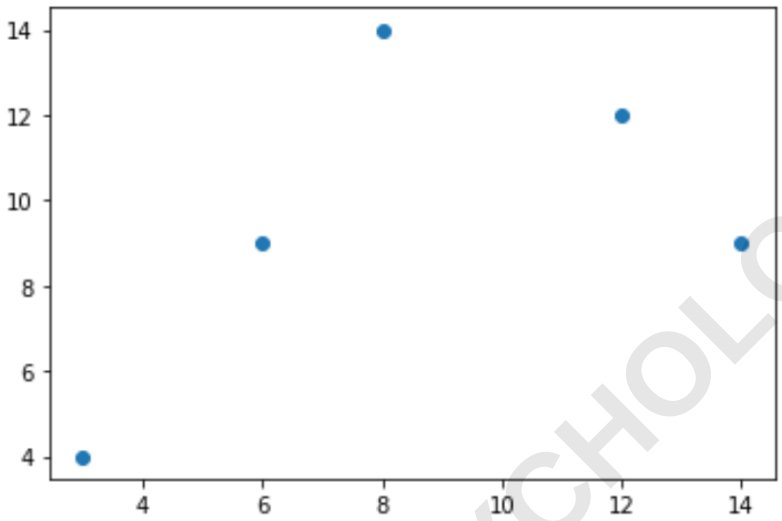
```
import matplotlib.pyplot as plt

#create data
```

**x =**

**y =**

**#create scatterplot**

**plt.scatter(x, y)**






## Annotate a Single Point

**We can use the following code to add an annotation to a single point in the plot:**

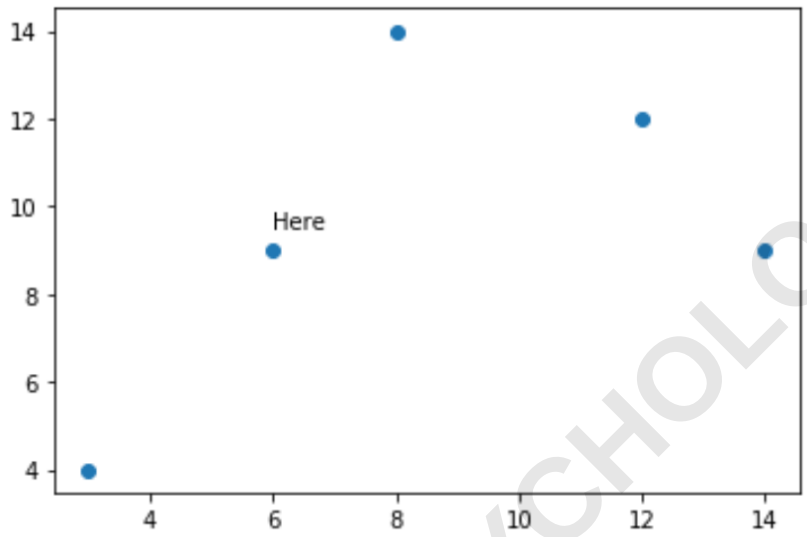
**import matplotlib.pyplot as plt**

**#create data**

**x =**

**y =**

```
#create scatterplot
plt.scatter(x, y)

#add text 'Here' at (x, y) coordinates = (6, 9.5)
plt.text(6, 9.5, 'Here')
```



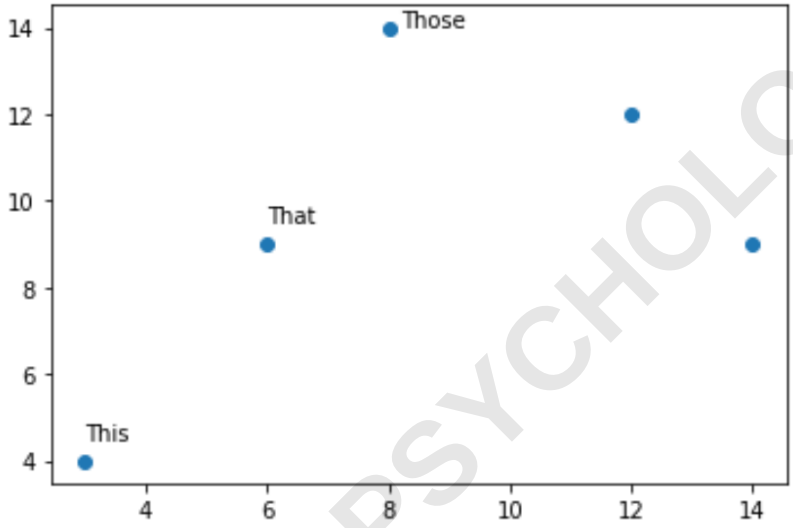



## Annotate Multiple Points

**We can use the following code to add annotations to multiple points in the plot:**

```
import matplotlib.pyplot as plt

#create data
x =
y =
```

```
#create scatterplot
plt.scatter(x, y)

#add text to certain points
plt.text(3, 4.5, 'This')
plt.text(6, 9.5, 'That')
plt.text(8.2, 14, 'Those')
```



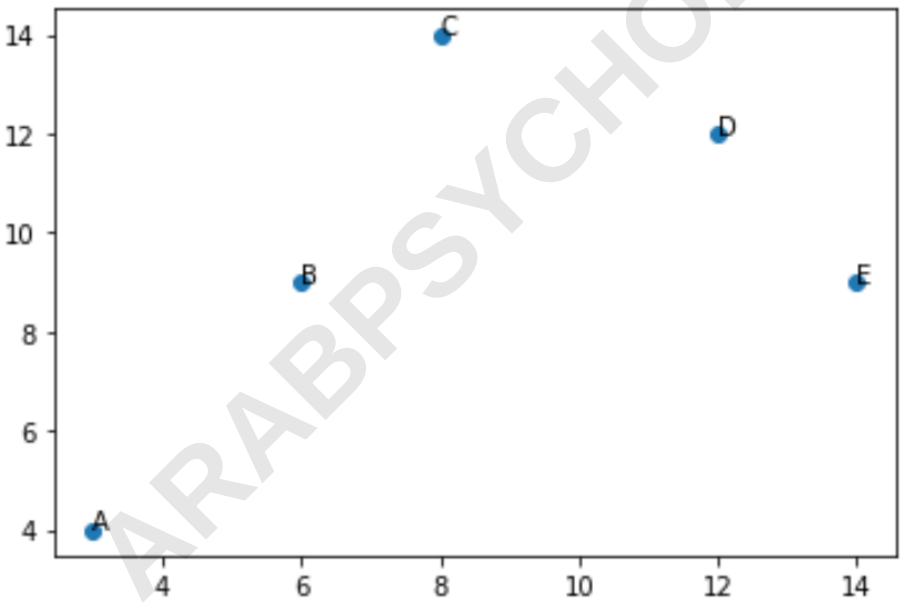



## Annotate All Points

**We can use the following code to add annotations to every single point in the plot:**

```
import matplotlib.pyplot as plt

#create data
```

```
x =
y =
labs =

#create scatterplot
plt.scatter(x, y)

#use for loop to add annotations to each point in plotfor
i, txt in enumerate(labs):
plt.annotate(txt, (x, y))
```

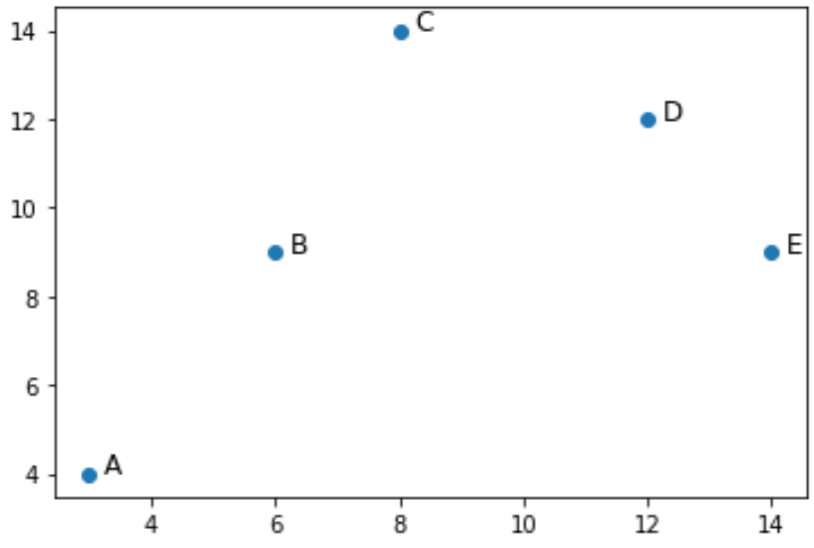





**By default, the annotations are placed directly on top of the points in the scatterplot and the default font size is 10.**

**The following code shows how to adjust both of these settings so the annotations are slightly to the right of the points and the font size is slightly larger:**

```
import matplotlib.pyplot as plt

#create data
x =
y =
labs =

#create scatterplot
plt.scatter(x, y)

#use for loop to add annotations to each point in plotfor
i, txt in enumerate(labs):
plt.annotate(txt, (x+.25, y), fontsize=12)
```




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