How can I use bold font in Matplotlib?

Authored by

stats writer

June 27, 2024

Matplotlib is a popular Python library used for creating visualizations and plots. One of the key features of Matplotlib is the ability to customize the appearance of the graphs and plots. One way to do this is by using bold font in the plot labels, titles, and legends. This can be achieved by adding an additional parameter, 'fontweight', to the text properties in the code. By setting the fontweight to 'bold', the text will appear in a thicker and more prominent font. This can be useful in highlighting important information or creating emphasis in the visualizations. Overall, using bold font in Matplotlib is a simple and effective way to enhance the appearance of your graphs and make them stand out.

Use Bold Font in Matplotlib (With Examples)

You can use the weight argument to create a bold font in Matplotlib.

This argument is commonly used with titles and annotated text in Matplotlib:

Method 1: Use Bold Font in Title

plt.title('My Title', weight='bold')

Method 2: Use Bold Font in Annotated Text

plt.text(6, 10, 'some text', weight='bold')

The following examples show how to use each method in practice.

Example 1: Use Bold Font in Title

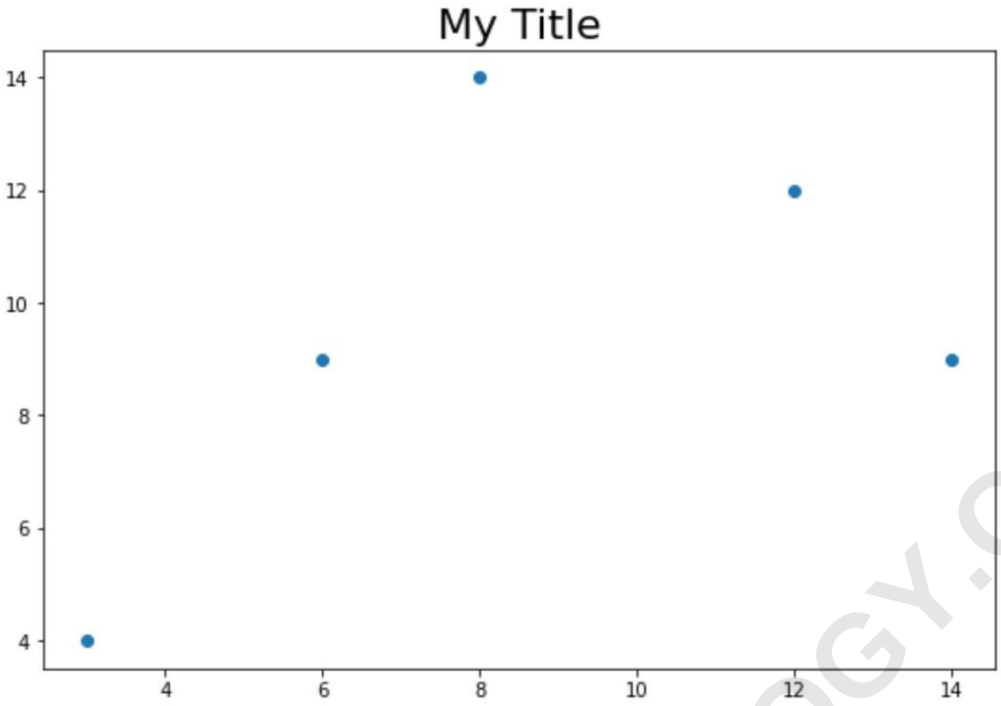
The following code shows how to create a scatterplot with a title in Matplotlib that uses regular font:

```
import matplotlib.pyplotas plt

#create data
x =
y =

#create scatterplot
plt.scatter(x, y)

#add title
plt.title('My Title', fontsize=22)
```



And the following code shows how to create a scatterplot with a title in Matplotlib and use the weight argument to enable bold font:

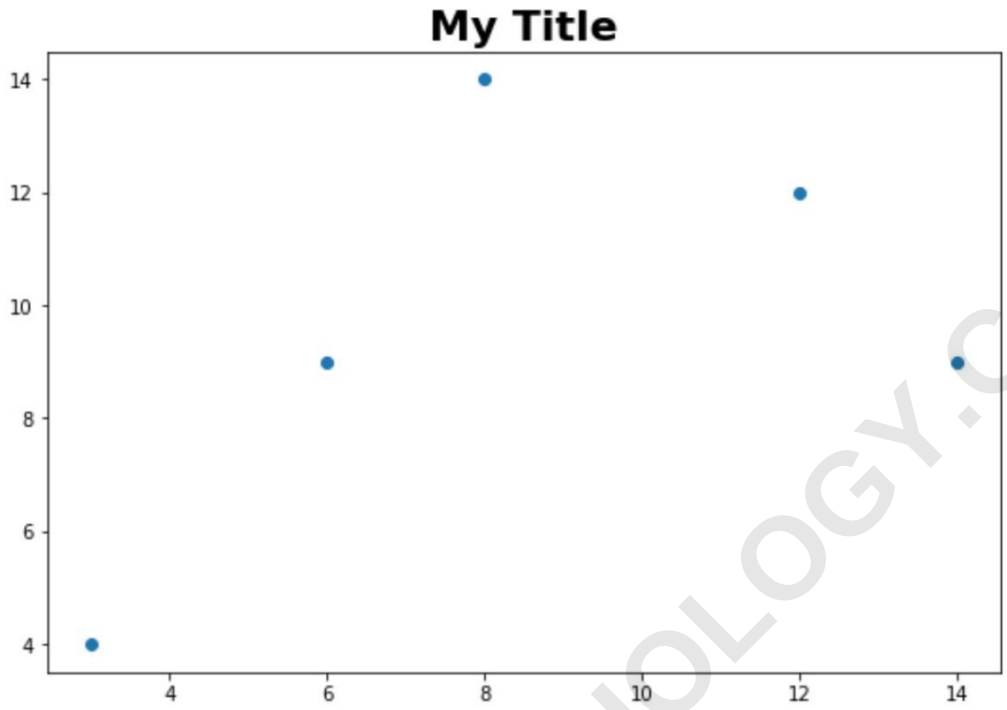
```
import matplotlib.pyplotas plt

#create data
x =
y =

#create scatterplot
plt.scatter(x, y)

#add title
```

plt.title('My Title', fontsize=22, weight='bold')



Example 2: Use Bold Font in Annotated Text

The following code shows how to use the weight argument to enable bold font in an annotated text in a Matplotlib scatterplot:

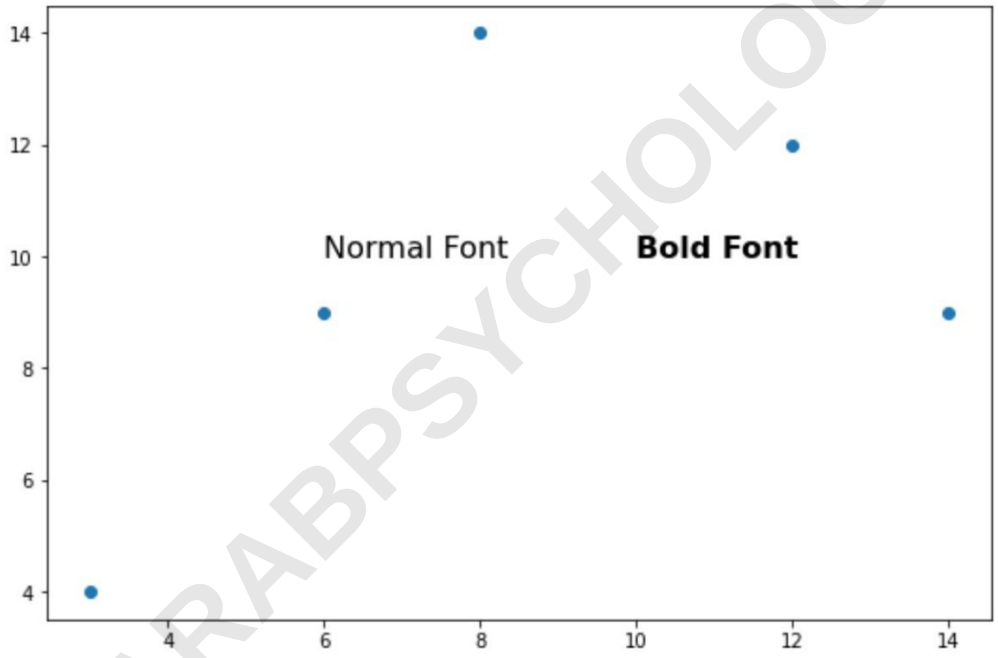
```
import matplotlib.pyplot as plt

#create data
x =
y =
```

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

#add regular text
plt.text(6, 10, 'Normal Font', fontsize=16)

#add bold text
plt.text(10, 10, 'Bold Font', fontsize=16, weight='bold')
```



Notice the difference between the normal font and the bold font.

The following tutorials explain how to perform other

common tasks in Matplotlib: