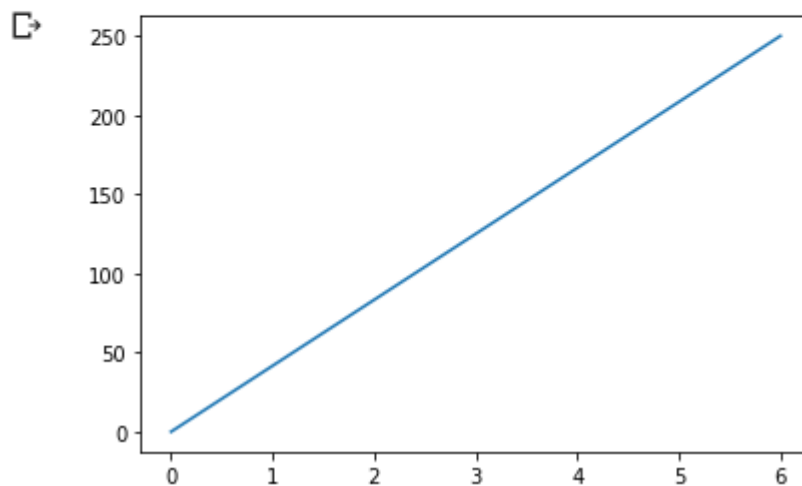


Week - 12

Visualize the datasets using matplotlib in python.(Histogram, Box plot, Bar chart, Pie chart etc.,)

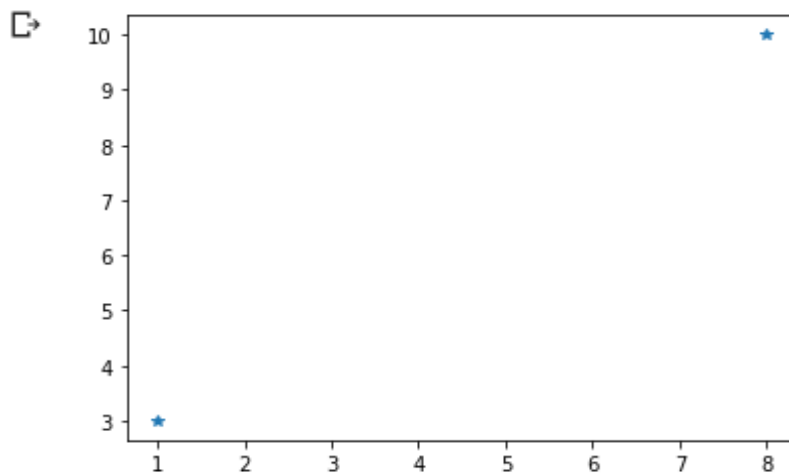
```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([0,6])
ypoints = np.array([0,250])
plt.plot(xpoints, ypoints)
plt.show()
```



The plot() function is used to draw points (markers) in a diagram. By default, the plot() function draws a line from point to point. The function takes parameters for specifying points in the diagram. Parameter 1 is an array containing the points on x-axis. Parameter 2 is an array containing the points on y-axis.

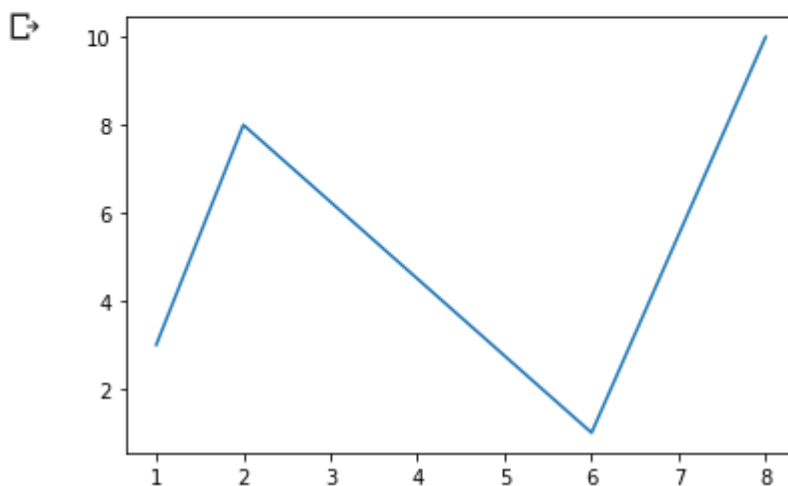
Plotting without a line

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([1,8])
ypoints = np.array([3,10])
plt.plot(xpoints, ypoints, '*')
plt.show()
```



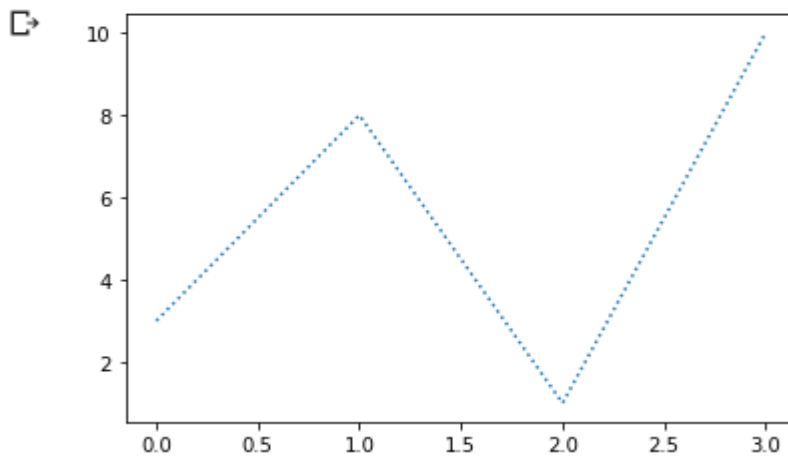
Multiple points

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([1,2,6,8])
ypoints = np.array([3,8,1,10])
plt.plot(xpoints, ypoints)
plt.show()
```



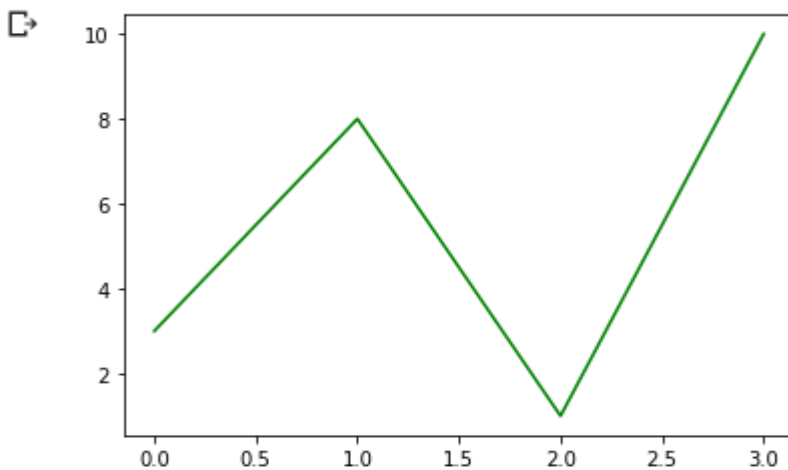
LineStyle : You can use the keyword argument `linestyle`, or shorter `ls`, to change the style of the plotted line

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3,8,1,10])
plt.plot(ypoints, linestyle = 'dotted')
plt.show()
```



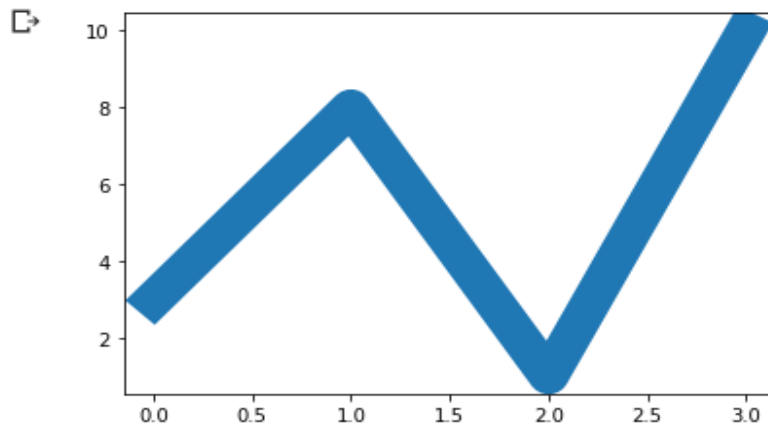
Color : You can use the keyword argument `color`, or shorter `c`, to change the color of the plotted line

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3,8,1,10])
plt.plot(ypoints, color = 'g')
plt.show()
```



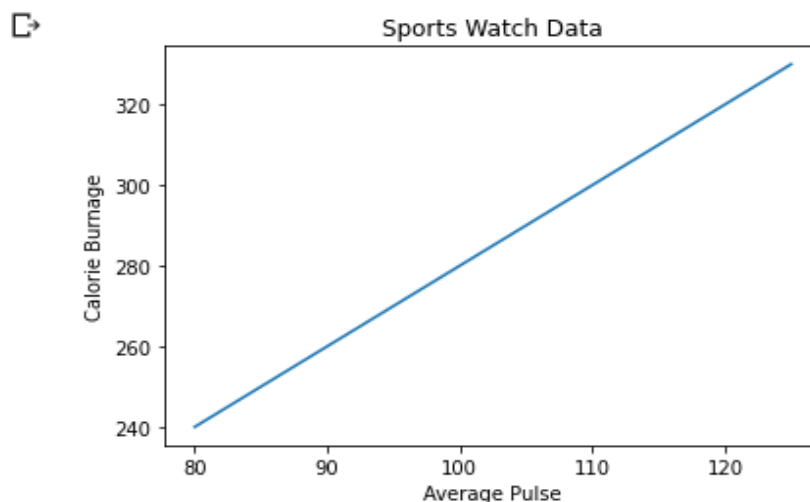
Line width: You can use the keyword argument linewidth to change the width or the thickness of the plotted line.

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3,8,1,10])
plt.plot(ypoints, linewidth = '20.5')
plt.show()
```



Label: You can give names to the x-axis and y-axis by using a keyword xlabel and ylabel respectively. You can also give the title to the graph by using the keyword title.

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array([80,85,90,95,100,105,110,115,120,125])
y = np.array([240,250,260,270,280,290,300,310,320,330])
plt.plot(x,y)
plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")
plt.show()
```



A) Histograms

```
import numpy as np
x = np.random.normal([170,10,250])
print(x)
```

```
[169.38116463  11.9161576  250.24784205]
```

A bar plot is also known as **Whisker plot** is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.

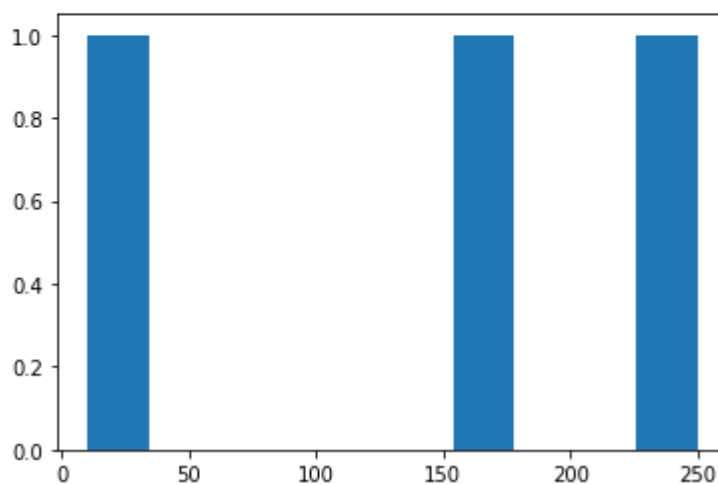
In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median.

Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

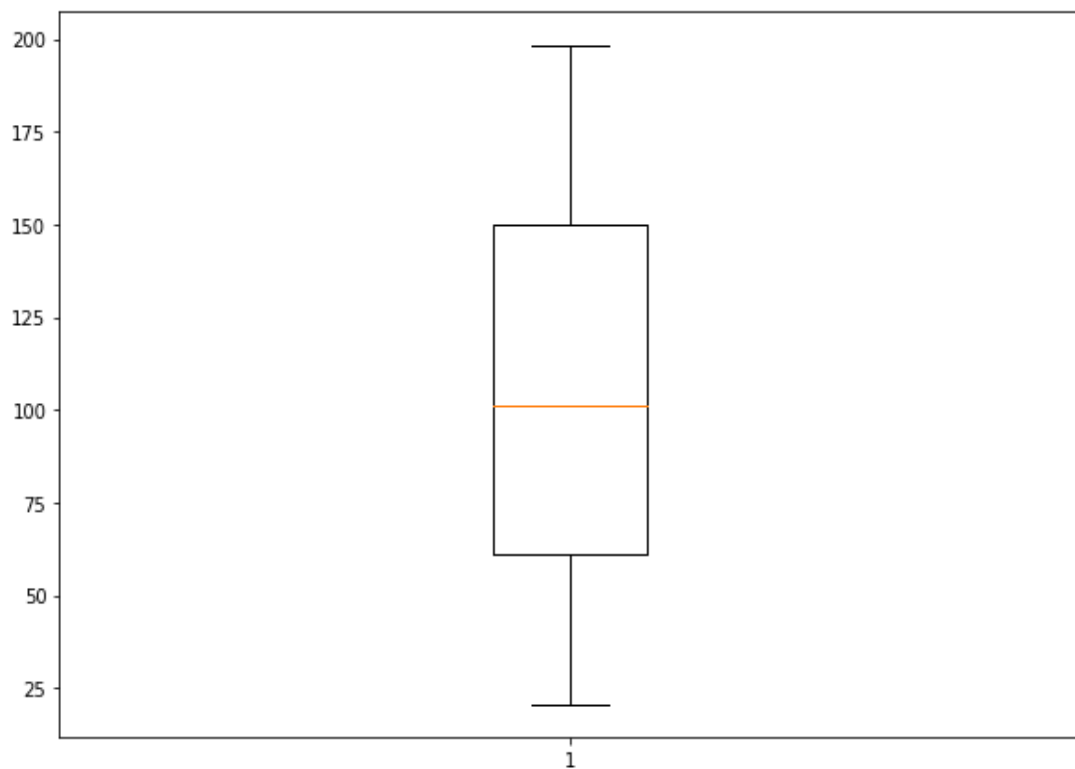
Syntax :

```
matplotlib.pyplot.boxplot(data, notch = None, vert = None,
patch_artist= None, widths= None)
```

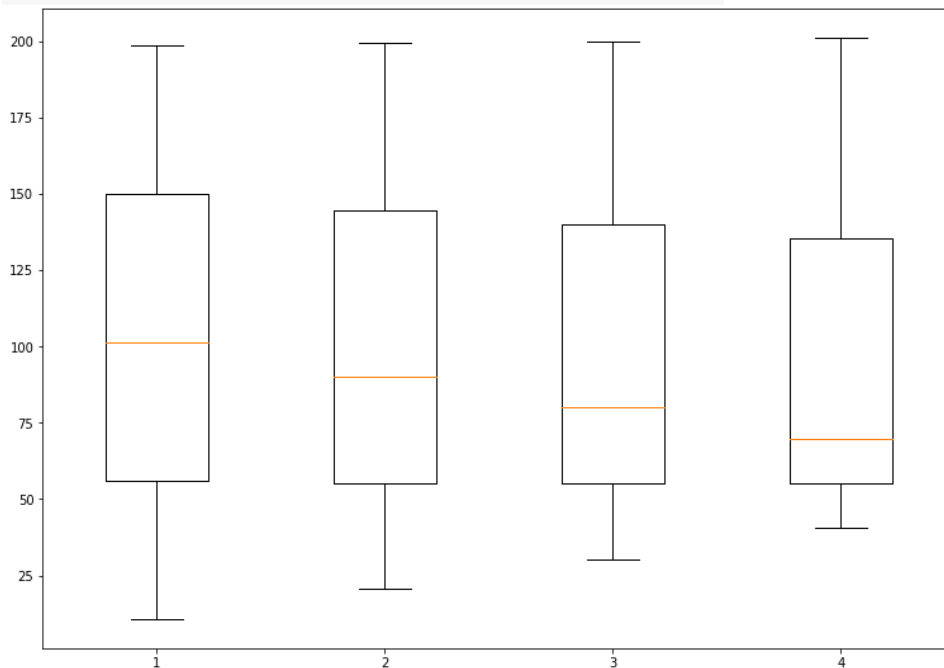
```
[41] import matplotlib.pyplot as plt
import numpy as np
x = np.random.normal([170,10,250])
plt.hist(x)
plt.show()
```



```
[44] import matplotlib.pyplot as plt
import numpy as np
np.random.seed(10)
data = np.random.normal([100,20,200])
fig = plt.figure(figsize = (10,7))
plt.boxplot(data)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
np.random.seed(10)
data_1 = np.random.normal([100,10,200])
data_2 = np.random.normal([90,20,200])
data_3 = np.random.normal([80,30,200])
data_4 = np.random.normal([70,40,200])
data = [data_1, data_2, data_3, data_4]
fig = plt.figure(figsize = (10,7))
ax = fig.add_axes([0,0,1,1])
bp = ax.boxplot(data)
plt.show()
```



Bar graph

A bar graph is a graphical representation of information. It uses bars that extend to different heights to depict value.

Bar graphs can be created with vertical bars, horizontal bars, grouped bars (multiple bars that compare values in a category), or stacked bars (bars containing multiple types of information).

Bars can also be grouped together for comparative purposes.

Data value is defined on the y-axis; data type is defined on the x-axis.

```
[53] import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure(0)
ax = fig.add_axes([0,0,1,1])
langs = ['C', 'C++', 'Java', 'Python', 'PHP']
students = [23,17,35,29,12]
ax.bar(langs,students)
plt.show()
```

