

# Use of Python Matplotlib Library

- **Python Matplotlib -**

matplotlib.pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

It is used for plotting, this python library provides and objected-oriented APIs for integrating plots into applications.

Matplotlib is not a part of the Standard Libraries which is installed by default when Python, there are several toolkits which are available that extend python matplotlib functionality. Some of them are separate downloads, others can be shipped with the matplotlib source code but have external dependencies.

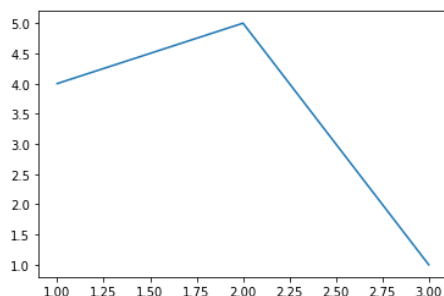
- **Types of Plots -**

There are various plots which can be created using python matplotlib. Some of them are listed below:

1. Bar Graph
2. Histogram
3. Scatter Plot
4. Area Plot
5. Pie Plot

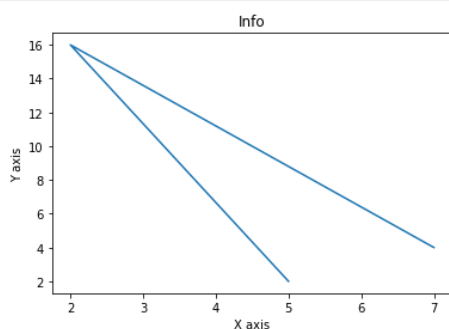
This is a code to generate basic graphs.

```
In [2]: from matplotlib import pyplot as plt  
plt.plot([1,2,3],[4,5,1])  
plt.show()
```



After adding title and labels to graph.

```
In [3]: from matplotlib import pyplot as plt  
x = [5,2,7]  
y = [2,16,4]  
plt.plot(x,y)  
plt.title('Info')  
plt.ylabel('Y axis')  
plt.xlabel('X axis')  
plt.show()
```



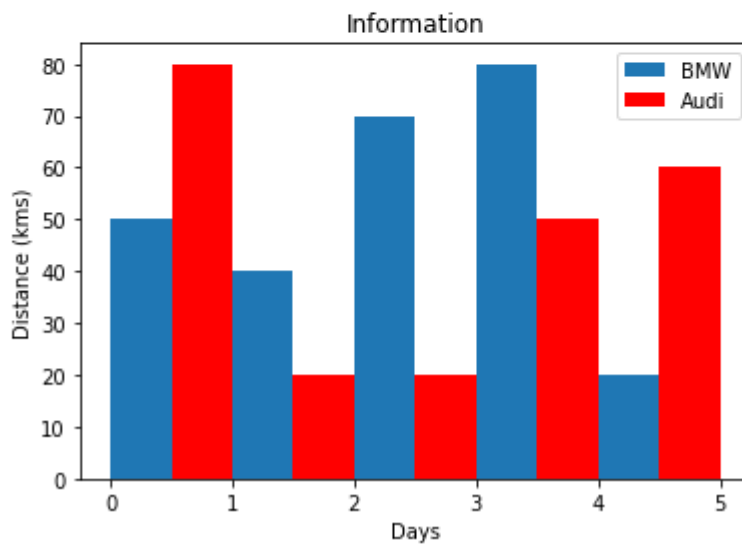
# Use of Python Matplotlib Library

## 1. Bar Graph:

A bar graph uses bars to compare data among different categories. It is well suited when you want to measure the changes over a period of time. It can be represented horizontally or vertically.

```
In [4]: from matplotlib import pyplot as plt

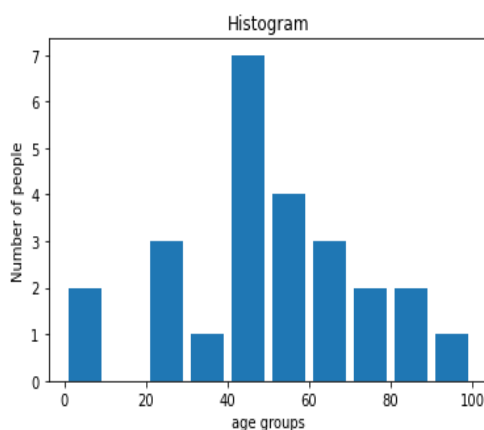
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],
        label="BMW",width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],
        label="Audi", color='r',width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```



## 2. Histogram:

Histograms are used to show a distribution whereas a bar chart is used to compare different entities. Histograms are useful when you have arrays or a very long list.

```
In [5]: import matplotlib.pyplot as plt
population_age = [22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,111,115,80,75,65,54,44,43,42,48]
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```



# Use of Python Matplotlib Library

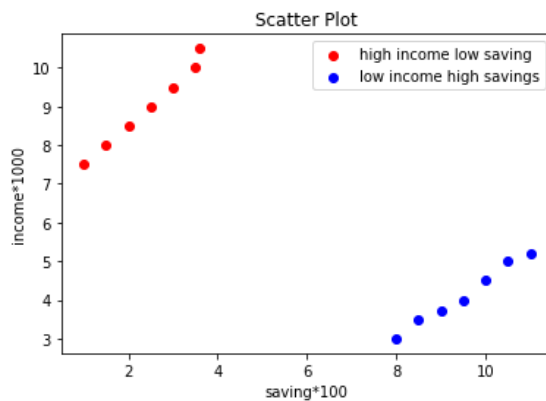
## 3. Scatter Plot:

We need scatter plots in order to compare variables.

```
In [6]: import matplotlib.pyplot as plt
x = [1,1.5,2,2.5,3,3.5,3.6]
y = [7.5,8,8.5,9,9.5,10,10.5]

x1=[8,8.5,9,9.5,10,10.5,11]
y1=[3,3.5,3.7,4,4.5,5,5.2]

plt.scatter(x,y, label='high income low saving',color='r')
plt.scatter(x1,y1,label='low income high savings',color='b')
plt.xlabel('saving*100')
plt.ylabel('income*1000')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```



## 4. Area Plot:

Area plots are pretty much similar to the line plot. They are also known as stack plots. These plots can be used to track changes over time for two or more related groups that make up one whole category.

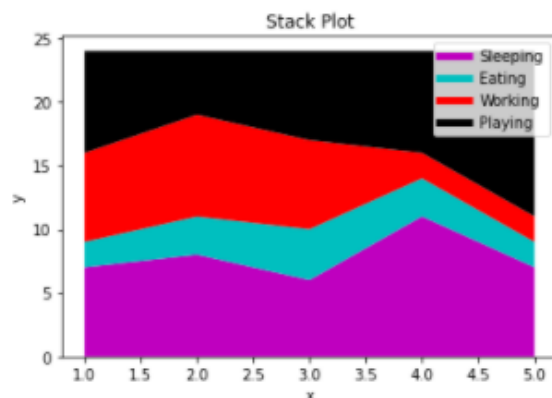
```
In [8]: import matplotlib.pyplot as plt
days = [1,2,3,4,5]

sleeping = [7,8,6,11,7]
eating = [2,3,4,3,2]
working = [7,8,7,2,2]
playing = [8,5,7,8,13]

plt.plot([],[],color='m', label='Sleeping', linewidth=5)
plt.plot([],[],color='c', label='Eating', linewidth=5)
plt.plot([],[],color='r', label='Working', linewidth=5)
plt.plot([],[],color='k', label='Playing', linewidth=5)

plt.stackplot(days, sleeping,eating,working,playing, colors=['m','c','r','k'])

plt.xlabel('x')
plt.ylabel('y')
plt.title('Stack Plot')
plt.legend()
plt.show()
```



## Use of Python Matplotlib Library

### 5. Pie Chart:

A pie chart refers to a circular graph which is broken down into segments i.e. slices of pie. It is basically used to show the percentage or proportional data where each slice of pie represents a category.

```
In [9]: import matplotlib.pyplot as plt

days = [1,2,3,4,5]

sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b']

plt.pie(slices,
        labels=activities,
        colors=cols,
        startangle=90,
        shadow= True,
        explode=(0,0.1,0,0),
        autopct='%1.1f%%')

plt.title('Pie Plot')
plt.show()
```

