Use Of 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.

Use of Matplotlib-

→ Matplotlib is a Python Library used for plotting, this python library provides and objected-oriented APIs for integrating plots into applications.

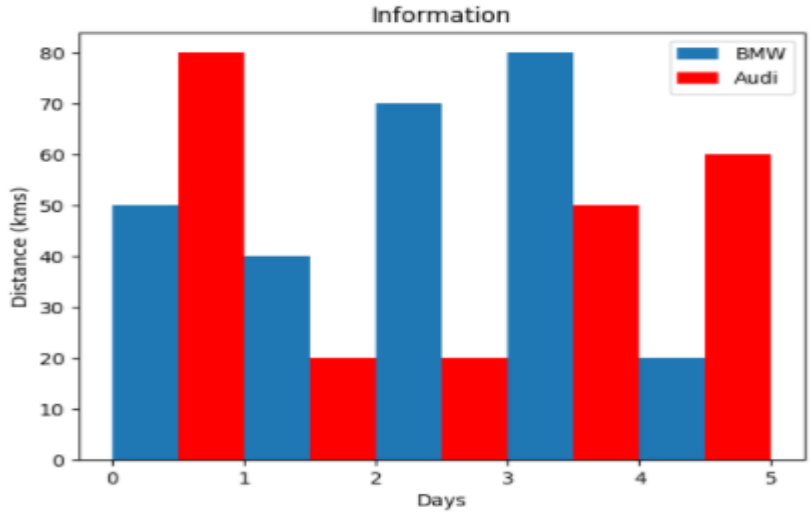
Python Matplotlib: Types of Plots

- **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. Longer the bar, greater is the value.

```
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()
```

Output-



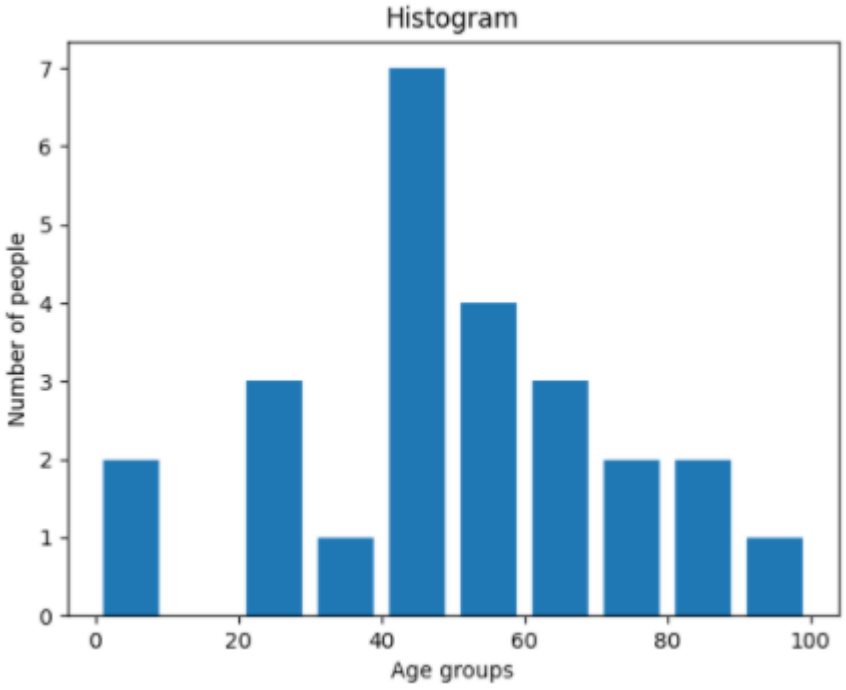
Use Of Matplotlib Library

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. Bin refers to the range of values that are divided into series of intervals. Bins are usually created of the same size.

```
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
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()
```

Output-



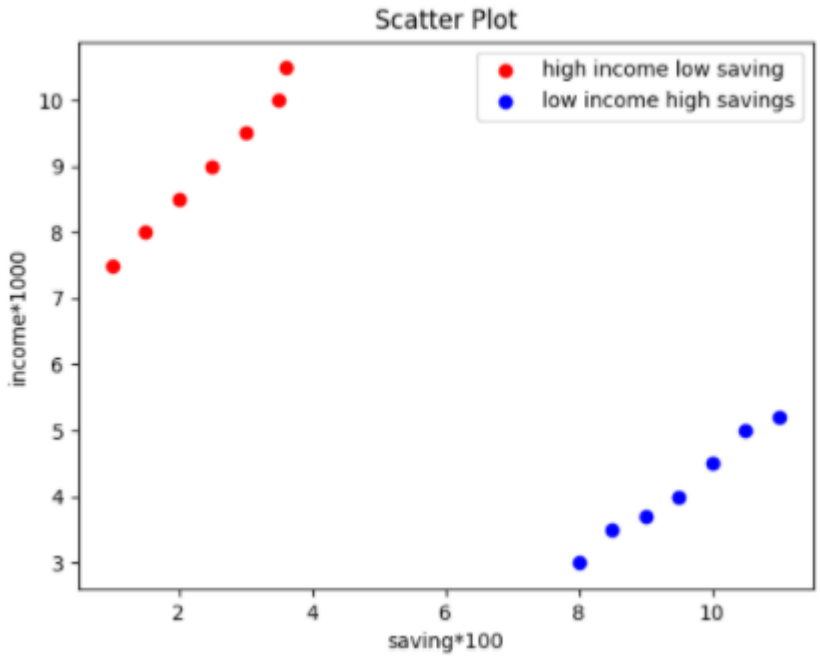
Use Of Matplotlib Library

Scatter Plot-

It is used to compare variables, for example, how much one variable is affected by another variable to build a relation out of it. The data is displayed as a collection of points, each having the value of one variable which determines the position on the horizontal axis and the value of other variable determines the position on the vertical axis.

```
1  import matplotlib.pyplot as plt
2  x = [1,1.5,2,2.5,3,3.5,3.6]
3  y = [7.5,8,8.5,9,9.5,10,10.5]
4
5  x1=[8,8.5,9,9.5,10,10.5,11]
6  y1=[3,3.5,3.7,4,4.5,5,5.2]
7
8  plt.scatter(x,y, label='high income low saving',color='r')
9  plt.scatter(x1,y1,label='low income high savings',color='b')
10 plt.xlabel('saving*100')
11 plt.ylabel('income*1000')
12 plt.title('Scatter Plot')
13 plt.legend()
14 plt.show()
```

Output-



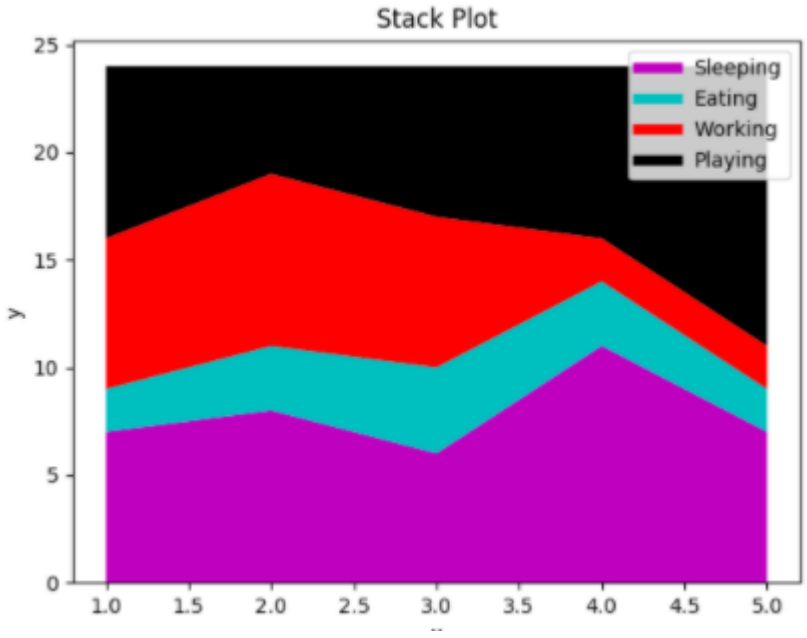
Use Of Matplotlib Library

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.

```
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()
```

Output-



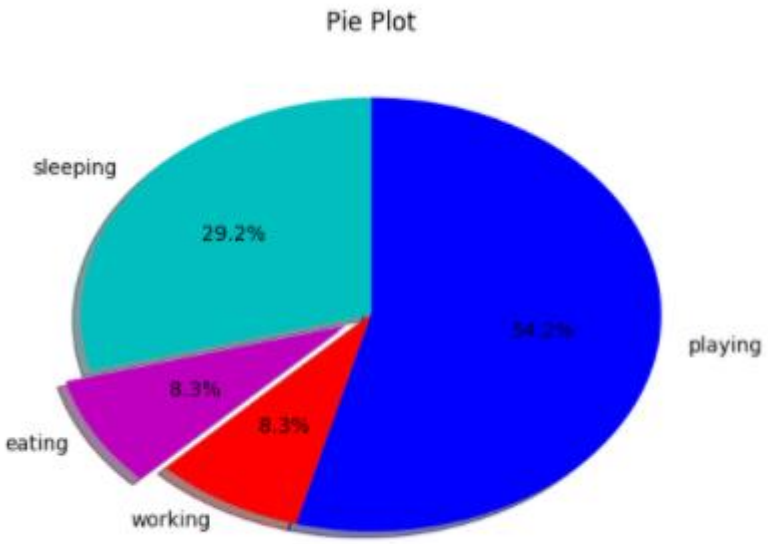
Use Of Matplotlib Library

Pie Plot-

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.

```
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()
```

Output-

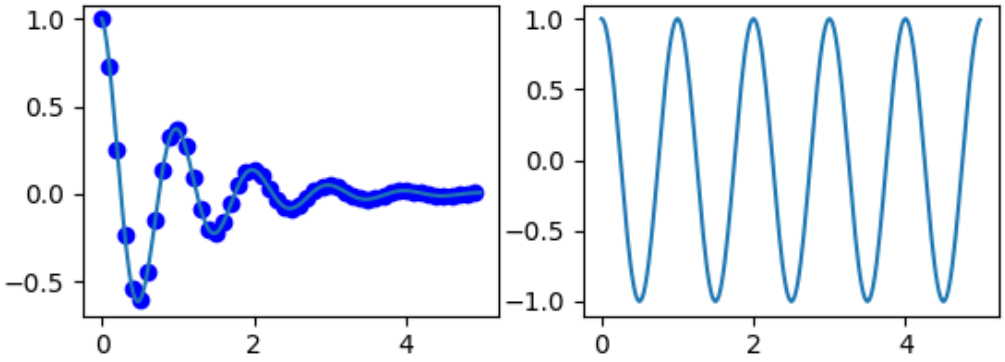


Use Of Matplotlib Library

Working With Multiple Plots-

```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)
t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)
plt.subplot(221)
plt.plot(t1, f(t1), 'bo', t2, f(t2))
plt.subplot(222)
plt.plot(t2, np.cos(2*np.pi*t2))
plt.show()
```

Output-



Simple Graph-

```
from matplotlib import pyplot as plt

 #Plotting to our canvas

 plt.plot([1,2,3],[4,5,1])

 #Showing what we plotted

 plt.show()
```

Output-