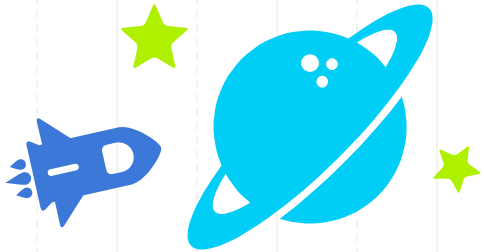


Matplotlib

Plotting of data includes advance concepts of matplotlib and allows you to create, clean and manipulate data plotting.

Data Plotting OPERATIONS

- Plotting Demo Data (First Plot)
- Files Operation
- Merge
- Data Transformation
- Groupby and Data aggregator

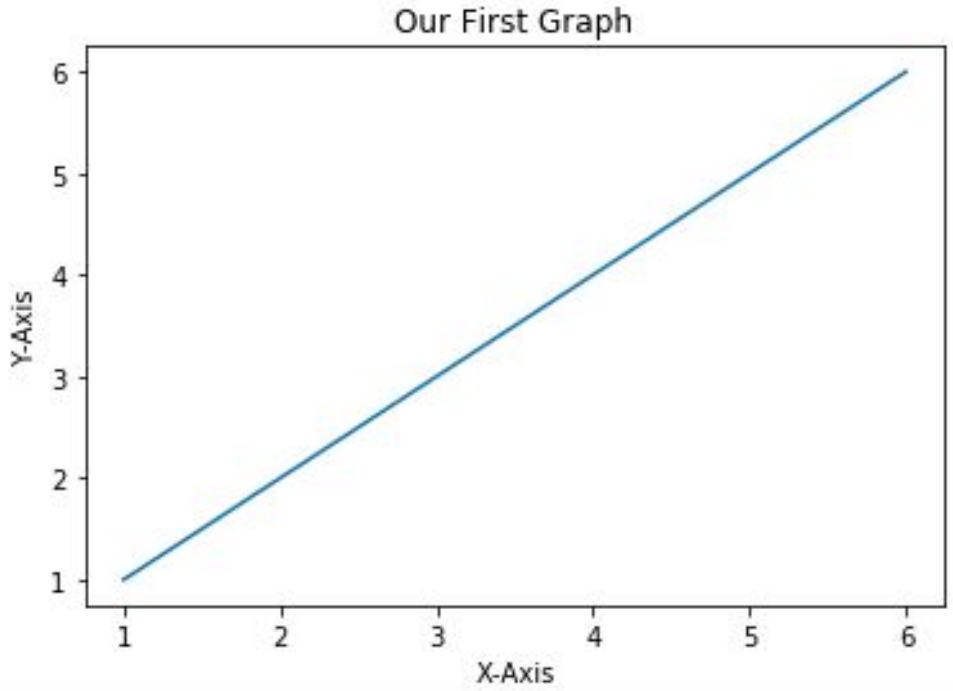
“

import matplotlib.pyplot as plt

Plotting First Graph

```
import matplotlib.pyplot as plt
x = [1,2,3,4,5,6]
y = [1,2,3,4,5,6]
plt.plot(x,y)
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.title('Our First Graph')
plt.show()
```

Out First Plot



Changing Y Plot

```
import matplotlib.pyplot as plt
x = [1,2,3,4,5,6]
y = [2,5,6,4,7,8]
plt.plot(x,y)
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.title('Our First Graph')
plt.show()
```

Cosine Wave Plot

```
t = np.arange(0.0,2.0,0.01)
s = 1 + np.cos(2*np.pi*t)
plt.plot(t,s,'r')
plt.xlabel('TIME(t)')
plt.ylabel('Voltage(mV)')
plt.title('Cosine wave plot COS(x)')
plt.show()
```

Cosine Wave Plot with Grid

```
t = np.arange(0.0,2.0,0.01)
s = 1 + np.cos(2*np.pi*t)
plt.plot(t,s,'--')
plt.grid()
plt.xlabel('TIME(t)')
plt.ylabel('Voltage(mV)')
plt.title('Cosine wave plot COS(x)')
plt.show()
```

Subplot in Matplotlib

```
x1 = np.linspace(0.0,5.0)
x2 = np.linspace(0.0,2.0)
y1 = np.cos(2*np.pi*x1)*np.exp(-x1)
y2 = np.cos(2*np.pi*x2)
plt.subplot(2,1,1)
plt.plot(x1,y1,'o-')
plt.title('Subplot-1')
plt.xlabel('x1')
plt.ylabel('Amp(y1)')
plt.show()
```

Subplot in Matplotlib

```
plt.subplot(2,1,2)
plt.plot(x2,y2,'.-')
plt.title('Subplot-2')
plt.xlabel('x2')
plt.ylabel('Amp(y2)')
plt.show()
```

Bar Graph

```
x = [1,2,3,4,5]
y = [10,20,30,50,80]
tick_label = ['one','two','three','four','five']
plt.bar(x,y,tick_label=tick_label,color=['red','green','blue'])
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.title('Bar Graph')
plt.show()
```

Histogram

```
temp = [30,30,30,43,43,44,45,56,67,78,89,32]
bins = 10

rangee = (0,100)
plt.hist(temp,bins,rangee,color='red',histtype='bar',rwidth=0.7)
plt.show()
```

Scatter Graph

```
x = [1,2,3,4,5,6,7,8,9,10]
y = [10,20,30,20,10,22,54,43,21,23]
plt.scatter(x,y,label='Starts',color='red',marker='*',s=20)
plt.legend()
plt.show()
```

Pie Chart

```
activities = ['eat','sleep','work','repeat']
slices = [3,7,8,2]
color = ['r','g','y','b']
plt.pie(slices,labels=activities,colors=color,startangle=90,shadow=True,explode=(0.2,0,0,0),autopct='%1.0f%%')
plt.legend()
plt.show()
```

Plotting multiple graphs

```
x = np.arange(0,2*np.pi,0.01)
y1 = np.sin(x)
y2 = np.cos(x)
plt.plot(x,y1,color='red',label='Sin')
plt.plot(x,y2,color='green',label='Cos')
plt.legend()
plt.show()
```

Plotting from CSV

```
import matplotlib.pyplot as plt
import csv
x= []
y= []
with open('test.csv','r')as csvfile:
    plots = csv.reader(csvfile)
    for col in plots:
        x.append(col[0])
        y.append(col[1])
```

Plotting from CSV

```
plt.plot(x,y,label='File',color='green')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Test Graph')
plt.legend()
plt.show()
```

Live Plot

```
import requests
import matplotlib.animation as animation
from matplotlib import style
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
fig1 = plt.figure()
ax1 = fig1.add_subplot(1,1,1)
```

Live Plot

```
def animate(p):
    data =
requests.get('https://demo-live-data.highcharts.com/vs-load.csv')
    plot_data = data.text
    line_data = plot_data.split('\n')
    x1=[]
    y1=[]
```

Live Plot

```
for line in range(1,len(line_data)):
    if len(line_data[line])>1:
        y,x = line_data[line].split(',')
        x1.append(x)
        y1.append(y)
    ax1.clear()
    ax1.plot(x1,y1)
```

Live Plot

```
anime_data = animation.FuncAnimation(fig1, animate, interval = 500)

plt.show()
```