Plotting Graph with Python and MatPlotLib Library

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Installing Matplot Library: **$ pip install matplotlib**

*pip is the Python Package Manager for installing library hosted in pypi.org

Python Graphs Extra Tasks for Week-5 in Computational Mathematics (4MM013).

1. Line Plot,
2. X-Y Plot,
3. Scatter Plot,
4. Bar plot,
5. Histogram,
6. Pie-Chart,
7. Sub-plot,
8. Titles and Axis-Labels,
9. Legends, and
10. Save Plot as Image Files.

Remember : To use the functions or variables of library, we import that library.

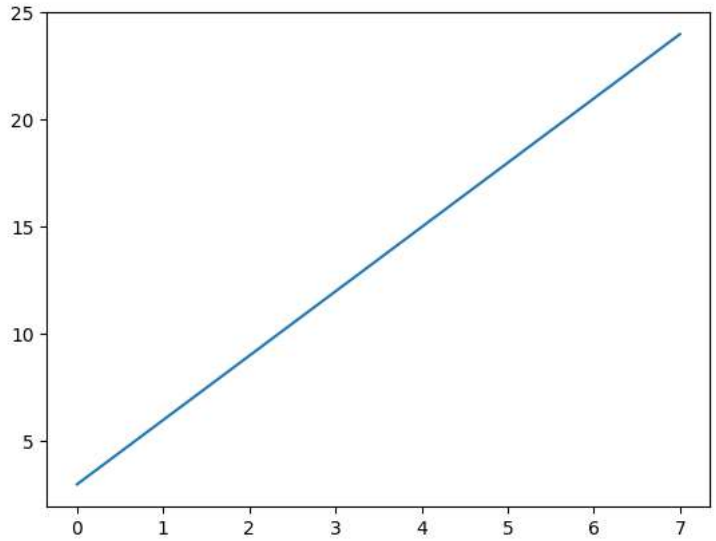
1. Create a simple line plot with Pyplot by plotting the 'data' list of values against the corresponding indexes (0 to 7).

Take : data = [3, 6, 9, 12, 15, 18, 21, 24]

```
#Akriti Kumari Dev
import matplotlib.pyplot as plt

data = [3, 6, 9, 12, 15, 18, 21, 24]
x = list(range(len(data)))

plt.plot(x,data)
plt.show()
```



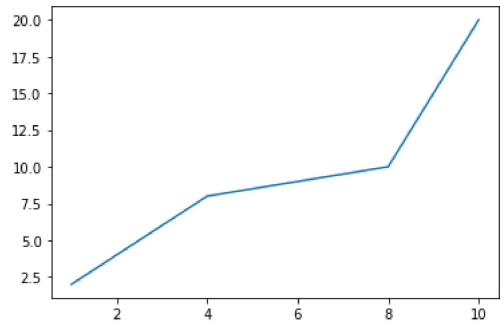
2. Plot two lists of data points against each other to form an X-Y plot.

x = [1, 2, 4, 8, 10] y = [2, 4, 8, 10, 20]

```
#Akriti Kumari Dev
x=[1, 2, 4, 8, 10]
```

```
y = [2, 4, 8, 10, 20]
plt.plot(x,y)
plt.show
```

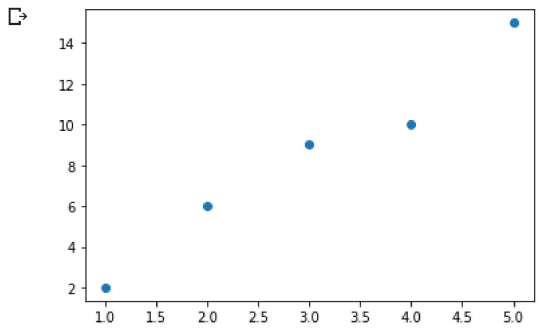
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



3. Create a scatter plot of data points, where each point is represented by a coordinate (x, y).

x = [1, 2, 3, 4, 5] y = [2, 6, 9, 10, 15]

```
#Akriti Kumari Dev
x = [1, 2, 3, 4, 5]
y = [2, 6, 9, 10, 15]
plt.scatter(x,y)
plt.show()
```

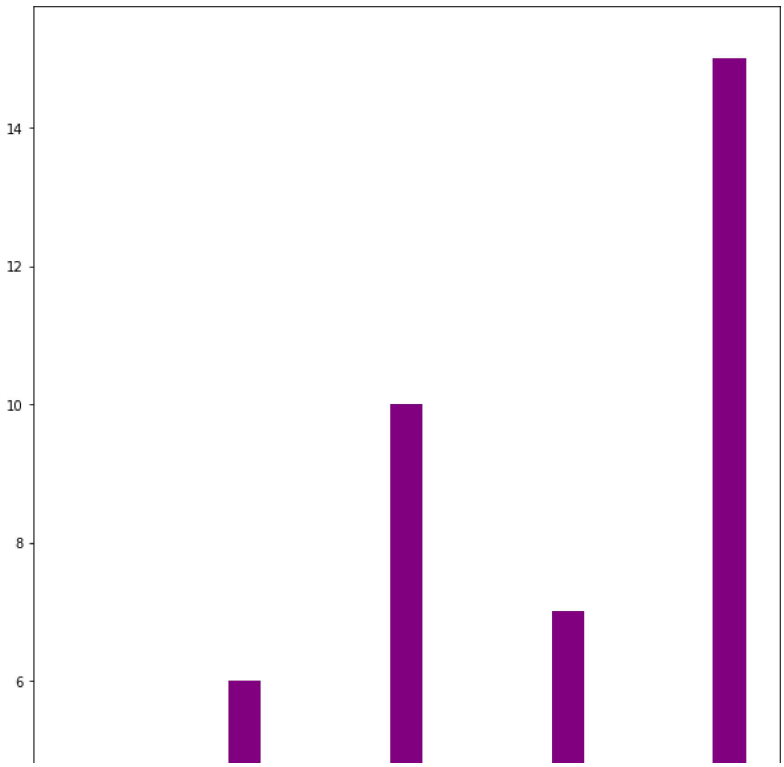


4. Create a bar plot to compare the frequency of items in a list.

items = ['A', 'B', 'C', 'D', 'E'] freq = [3, 6, 10, 7, 15]

Add 3 items with 3 additional frequencies to the graph.

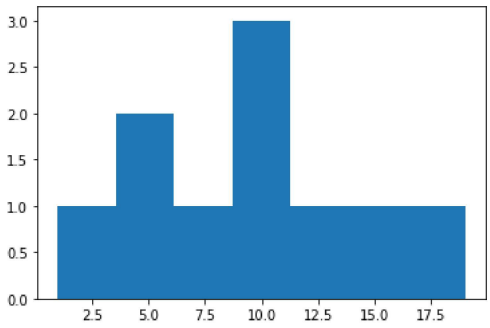
```
#Akriti Kumari Dev
items = ['A', 'B', 'C', 'D', 'E']
freq = [3, 6, 10, 7, 15]
fig=plt.figure(figsize=(10,15))
plt.bar(items,freq,color='purple' ,width=0.2)
plt.show()
```



5. Plot a histogram to visualize the distribution of a list of numerical values.

values = [1, 4, 4, 7, 10, 11, 11, 13, 14, 19] and bins=5

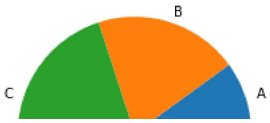
```
#Akriti Kumari Dev
values = [1, 4, 4, 7, 10, 11, 11, 13, 14, 19]
bins=7
plt.hist(values,bins)
plt.show()
```



6. Create a pie chart to represent percentages of various categories within a dataset.

labels = ['A', 'B', 'C', 'D'] sizes = [10, 20, 30, 40]

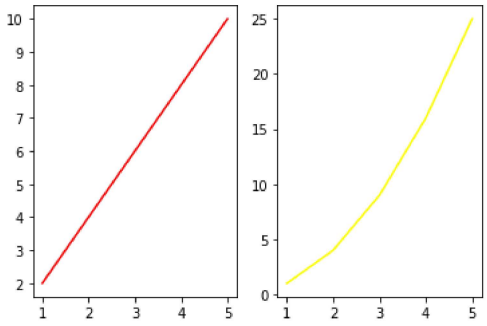
```
#Akriti Kumari Dev
labels = ['A', 'B', 'C', 'D']
sizes = [10, 20, 30, 40]
plt.pie(sizes,labels=labels)
plt.show()
```



7. Use subplots to display multiple plots in a single figure. For this task, create two line plots and display them side by side.

x1 = [1, 2, 3, 4, 5], y1 = [2, 4, 6, 8, 10], x2 = [1, 2, 3, 4, 5] and y2 = [1, 4, 9, 16, 25]

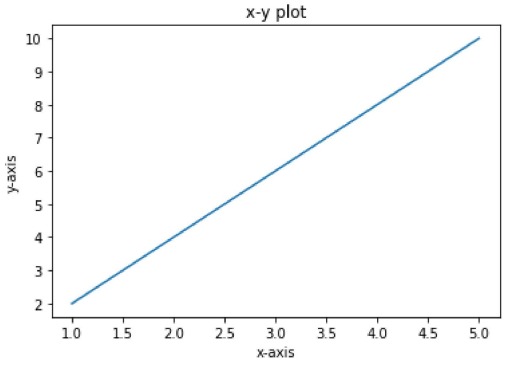
```
#Akriti Kumari Dev
x1 = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
x2 = [1, 2, 3, 4, 5]
y2 = [1, 4, 9, 16, 25]
fig,(ax1,ax2)=plt.subplots(1,2)
ax1.plot(x1,y1,color='red')
ax2.plot(x2,y2,color='yellow')
plt.show()
```



8. Titles and Axis Labels: Supplement a standard X-Y plot with titles and axis labels.

x = [1, 2, 3, 4, 5] y = [2, 4, 6, 8, 10]

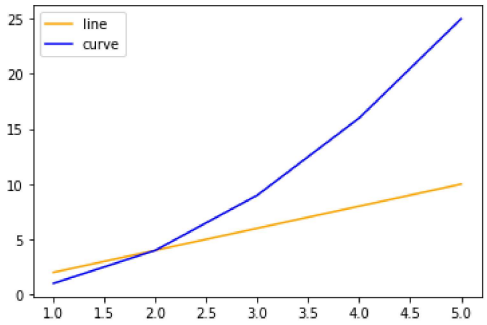
```
#Akriti Kumari Dev
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
plt.plot(x,y)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("x-y plot")
plt.show()
```



9. Create a plot with multiple lines or datasets and add a legend to differentiate them.

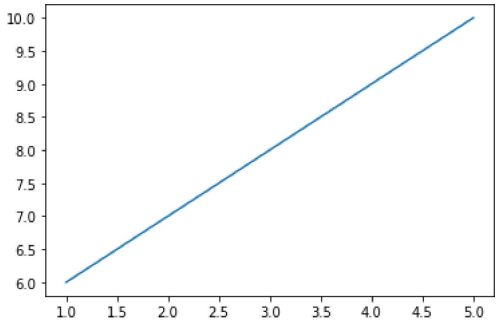
x = [1, 2, 3, 4, 5], y1 = [2, 4, 6, 8, 10], y2 = [1, 4, 9, 16, 25]

```
#Akriti Kumari Dev
x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 4, 9, 16, 25]
plt.plot(x,y1,label="line",color="orange")
plt.plot(x,y2,label="curve",color="blue")
plt.legend()
plt.show()
```



10. Save Plot as Image File: After creating an X-Y plot, save the resulting graph as an image file.

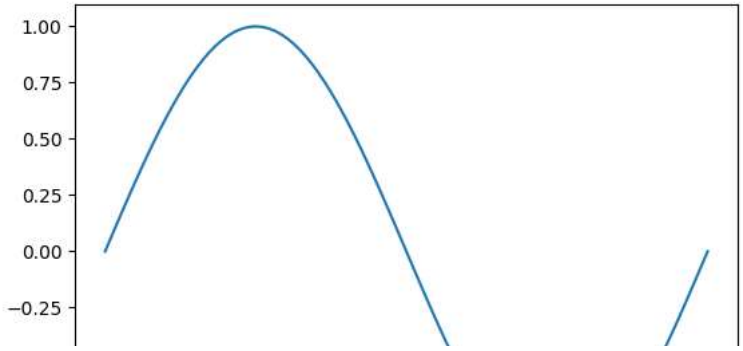
```
#Akriti Kumari Dev
x=[1,2,3,4,5]
y=[6,7,8,9,10]
plt.plot(x,y)
plt.savefig('work5.png')
```



Home Assignment:

1. Create a simple plot of sin(x). Using Numpy linspace to generate values of x and numpy sin(x) to generate y.

```
#Akriti Kumari Dev
import numpy as np
x=np.linspace(0,2*np.pi,100)
y=np.sin(x)
plt.plot(x,y)
plt.show()
```

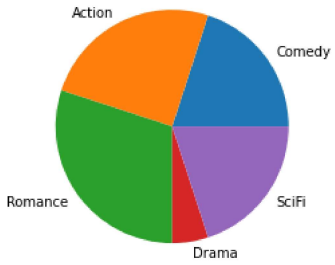


2. Create a pie-chart from data below:

Labels: Comedy, Action, Romance, Drama, SciFi

Data : 4, 5, 6, 1, 4

```
#Akriti Kumari Dev
import matplotlib.pyplot as plt
Labels= ['Comedy', 'Action', 'Romance', 'Drama', 'SciFi']
Data =[ 4, 5, 6, 1, 4]
plt.pie(Data,labels=Labels)
plt.show()
```



3. Create a scatter plot from kaggle.com(covid-19 data).