

## 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.

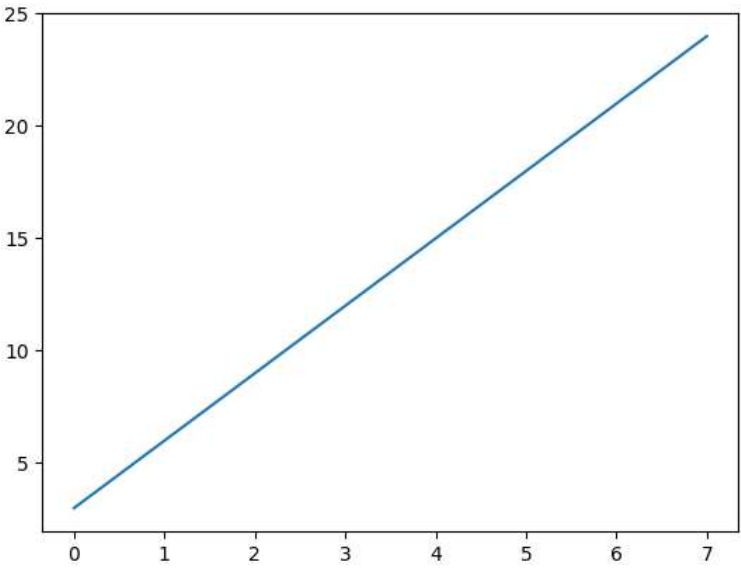
**Remember : To use the functions or variables of library, we import that library.**

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]

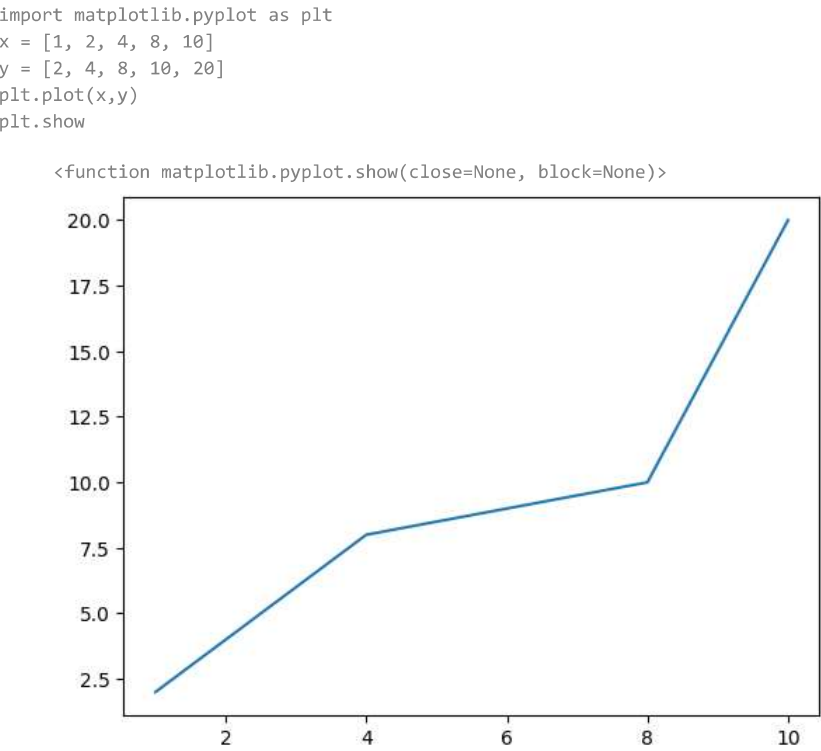
```
import matplotlib.pyplot as plt
data = [3, 6, 9, 12, 15, 18, 21, 24]
x = list (range(len(data)))
```

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

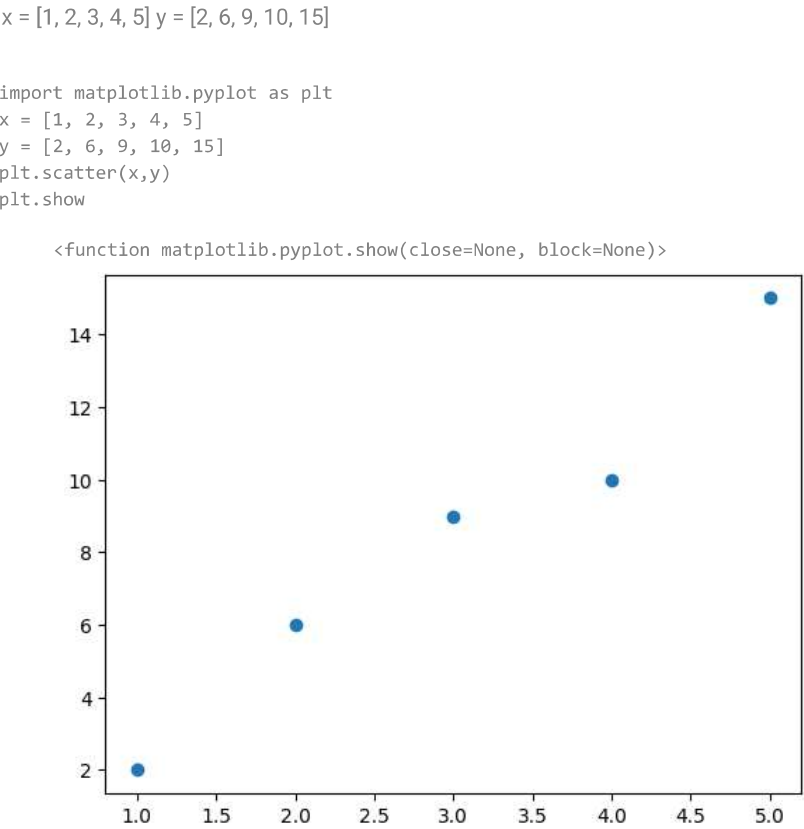


2. Plot two lists of data points against each other to form an X-Y plot.

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



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



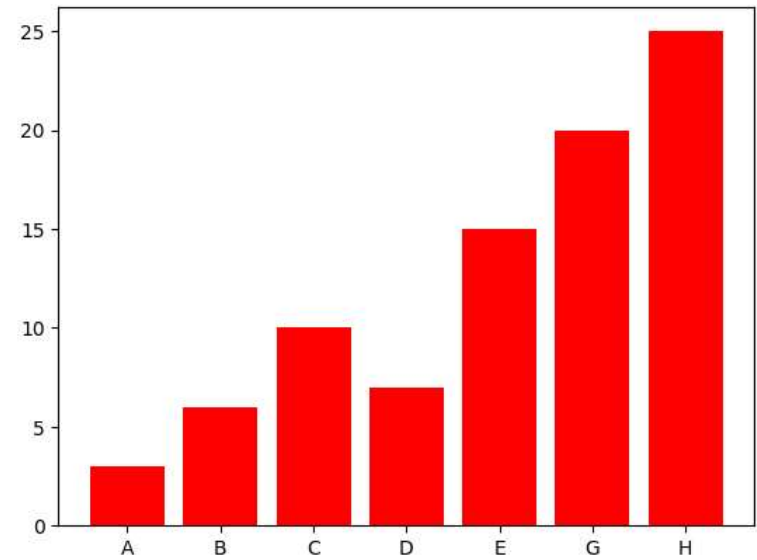
4. Create a bar plot to compare the frequency of items in a list.

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

Add 3 items with 3 additional frequencies to the graph.

```
import matplotlib.pyplot as plt
items = ['A', 'B', 'C', 'D', 'E', 'G','H']
freq = [3, 6, 10, 7, 15 ,20 ,25]
plt.bar(items,freq,color="red")
plt.show
```

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

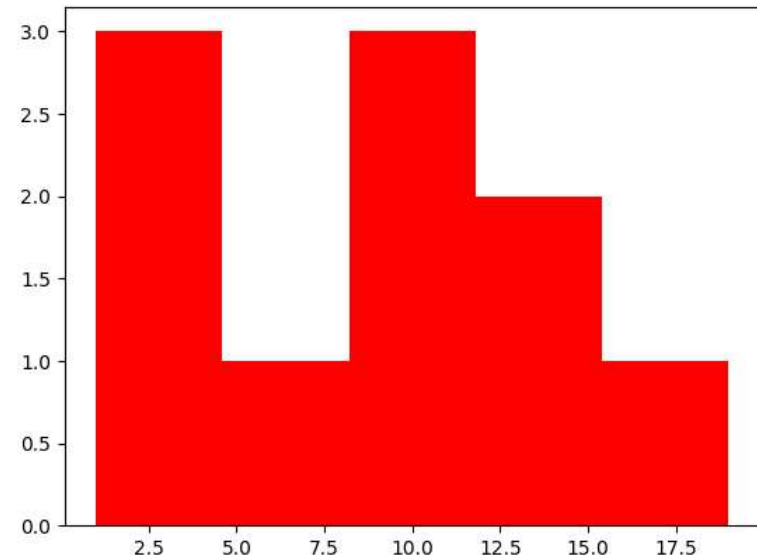


5. Plot a histogram to visualize the distribution of a list of numerical values.

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

```
import matplotlib.pyplot as plt
values = [1, 4, 4, 7, 10, 11, 11, 13, 14, 19]
bins=5
plt.hist(values,bins, color = "red")
plt.show
```

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



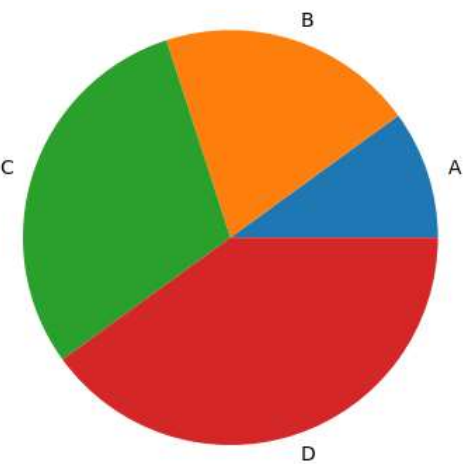
6. Create a pie chart to represent percentages of various categories within a dataset.

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

```
import matplotlib.pyplot as plt
labels = ['A', 'B', 'C', 'D']
```

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

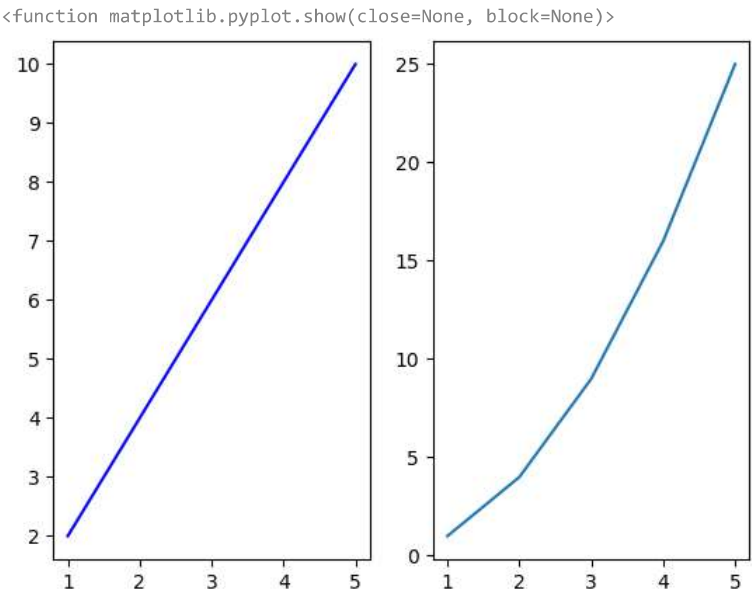
<function matplotlib.pyplot.show(close=None, block=None)>
```



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]
```

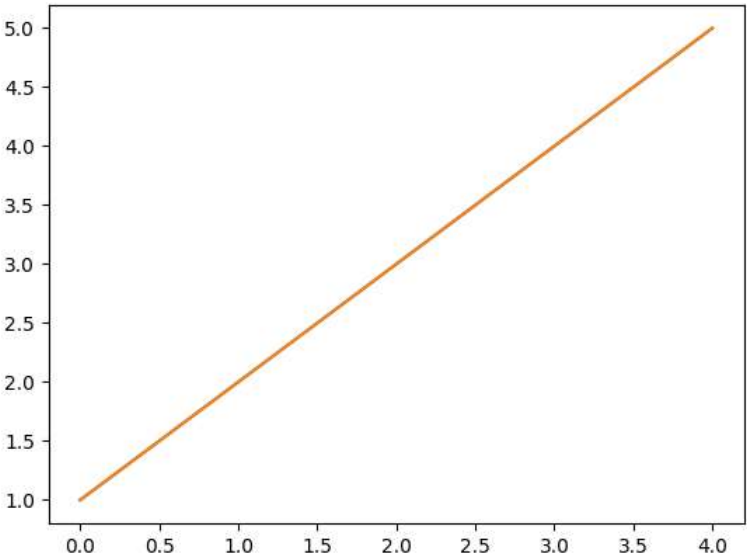
```
import matplotlib.pyplot as plt
x1 = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
x2 = [1, 2, 3, 4, 5]
y2 = [1, 4, 9, 16, 25]
fig,(ax1,ax2)=plt.subplots(1,2)
ax1.plot(x1,y1,color="blue")
ax2.plot(x2,y2)
plt.show
```



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]
```

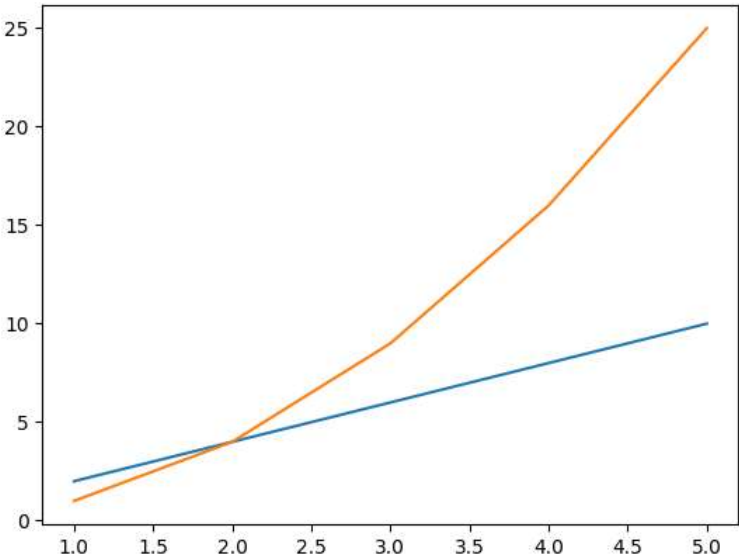
```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
plt.plot(x,label='first')
plt.plot(x,label='second')
plt.show()
```



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

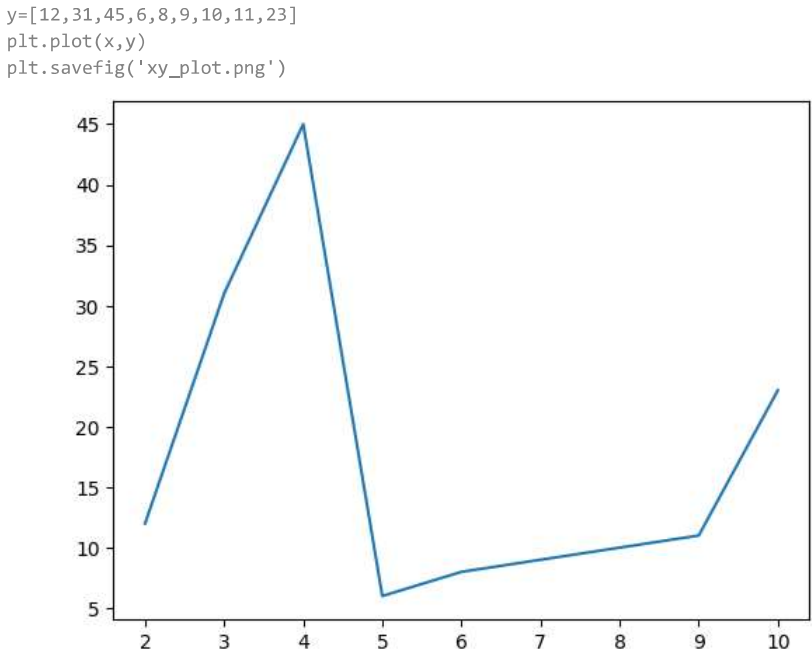
```
x = [1, 2, 3, 4, 5], y1 = [2, 4, 6, 8, 10], y2 = [1, 4, 9, 16, 25]
```

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 4, 9, 16, 25]
plt.plot(x1,y1,label='orange')
plt.plot(x,y2,label='red')
plt.show()
```



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

```
import matplotlib.pyplot as plt
x=[2,3,4,5,6,7,8,9,10]
```

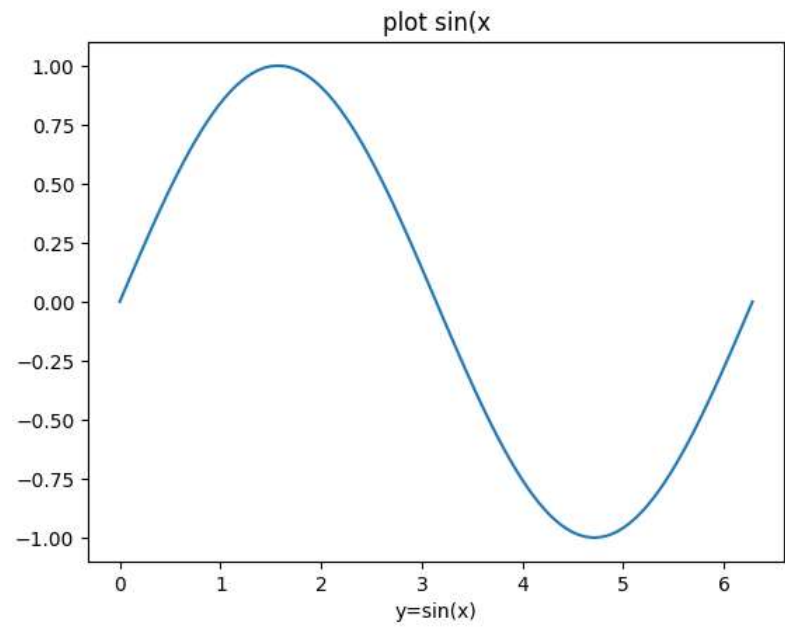


Home Assignment:

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

```
import numpy as np
import matplotlib.pyplot as plt
x= np.linspace(0, 2*np.pi,110)
y=np.sin(x)
plt.plot(x,y,)
plt.xlabel('x')
plt.ylabel('y=sin(x)')
plt.title('plot sin(x)')
```

Text(0.5, 1.0, 'plot sin(x)')



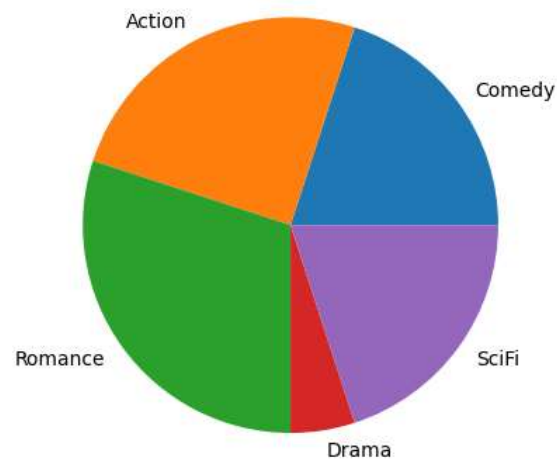
2. Create a pie-chart from data below:

Labels: Comedy, Action, Romance, Drama, SciFi

Data : 4, 5, 6, 1, 4

```
import matplotlib.pyplot as plt
labels = ['Comedy', 'Action', 'Romance', 'Drama', 'SciFi']
Data = [4, 5, 6, 1, 4]
fig, ax = plt.subplots()
ax.pie(Data,labels=labels)
plt.show
```

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



3. Create a scatter plot from kaggle.com covid -19 file as csv

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv(r'C:\Users\User\Desktop')
x = df['total_cases']
y = df['total_deaths']
fig, ax = plt.subplots()
ax.scatter(x,y)
ax.set_xlabel('cases')
ax.set_ylabel('deaths')
ax.set_title('covid19')
plt.show
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-20-1111a0a4f3d4> in <cell line: 3>()
      1 import pandas as pd
      2 import matplotlib.pyplot as plt
----> 3 df = pd.read_csv(r'C:\Users\User\Desktop')
      4 x = df['total_cases']
      5 y = df['total_deaths']
```

5 frames

```
/usr/local/lib/python3.9/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    784     if ioargs.encoding and "b" not in ioargs.mode:
    785         # Encoding
--> 786         handle = open(
    787             handle,
    788             ioargs.mode,
```

FileNotFoundError: [Errno 2] No such file or directory: 'C:\\Users\\User\\Desktop'

