NSSA 220
Task Automation with Interpreted Languages

Matplotlib

Instructor: Dr. Fahed Jubair

RIT DUBAI

Matplotlib

- Matplotlib is a popular python library for graph plotting and data visualization

- To install Matplotlib, execute the command:

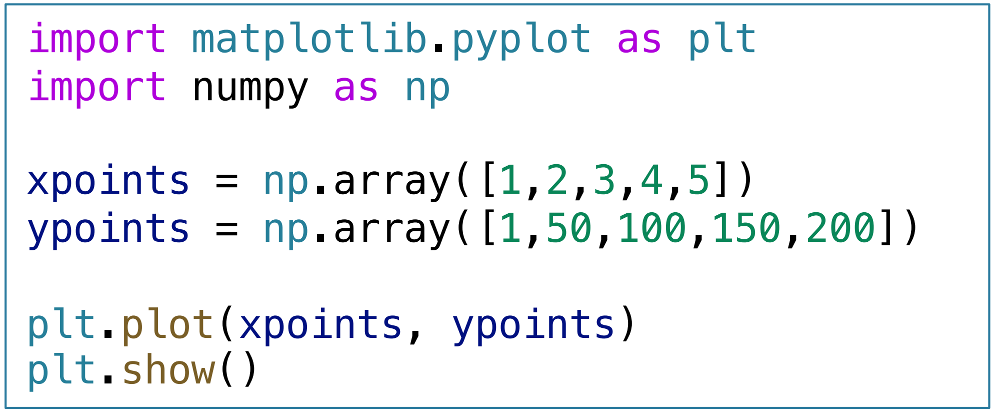
```
pip3 install matplotlib
```

- Slides Reference: https://www.w3schools.com/python/matplotlib_intro.asp

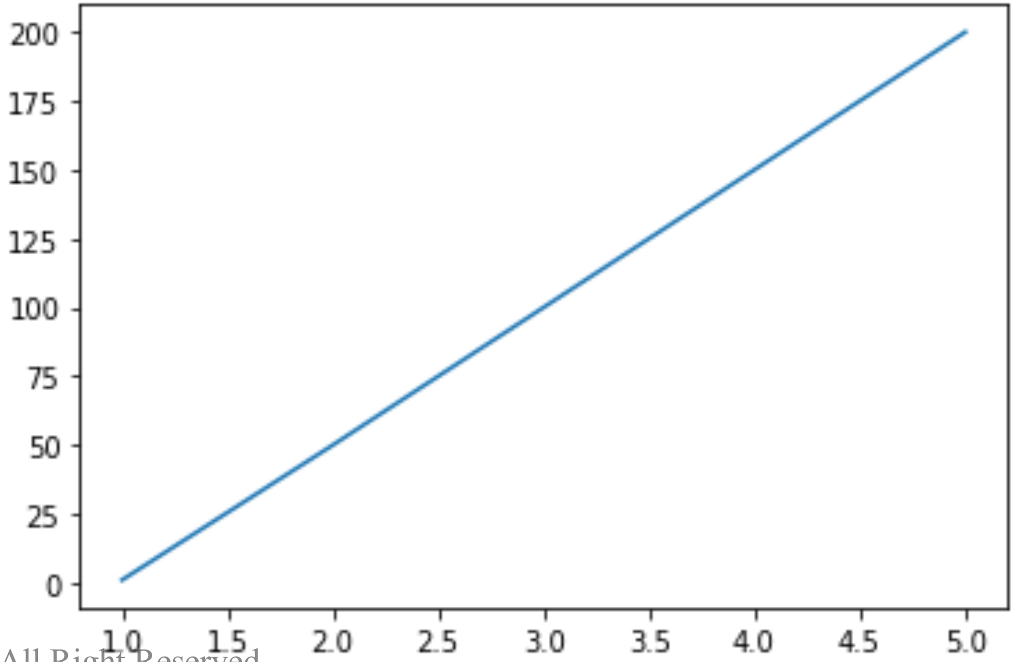
Pyplot

- Most of the Matplotlib utilities lies under the pyplot submodule

- Pyplot is usually imported under the *plt* alias



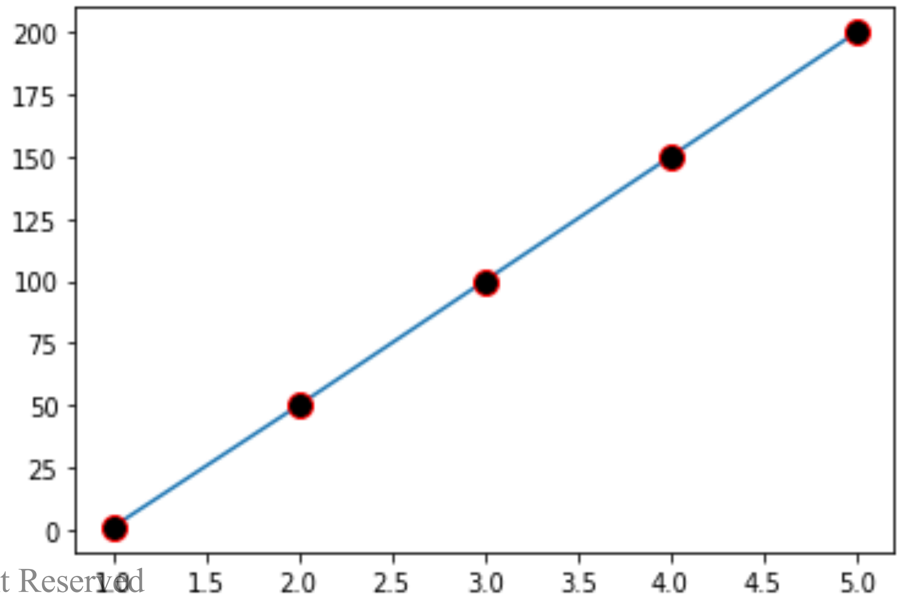
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1,2,3,4,5])
ypoints = np.array([1,50,100,150,200])

plt.plot(xpoints, ypoints)
plt.show()
```



Markers

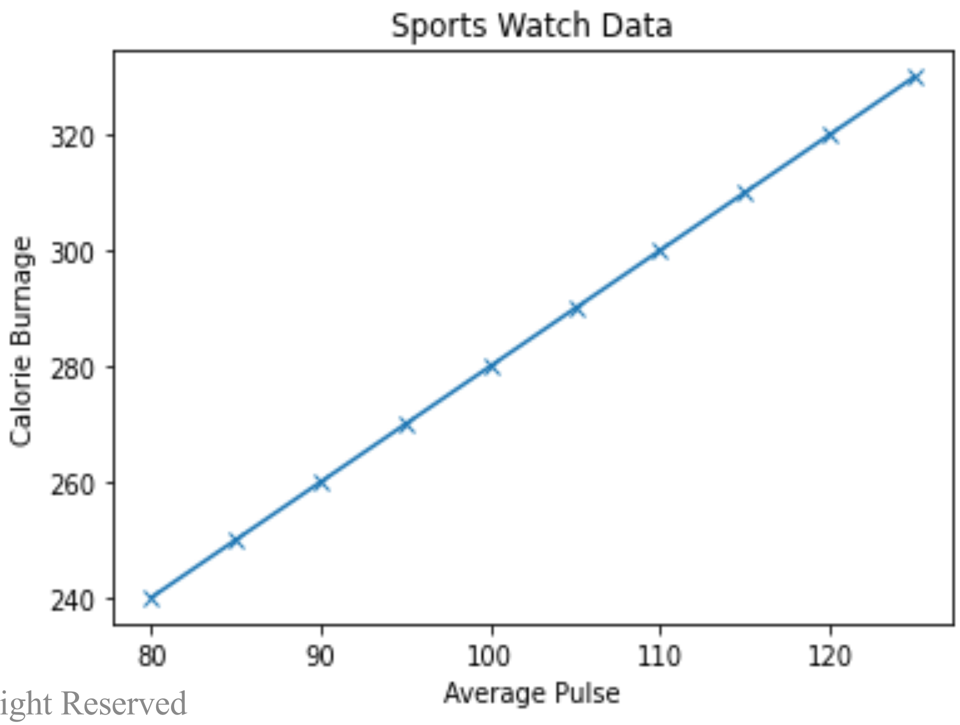
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1,2,3,4,5])
ypoints = np.array([1,50,100,150,200])

plt.plot(xpoints, ypoints, marker = 'o', ms = 10,
mec = 'red', mfc = 'black’)

plt.show()
```

- *marker* option specifies marker shape
- *ms* option specifies marker size
- *mec* option specifies marker edge color
- *mfc* option specifies marker face color



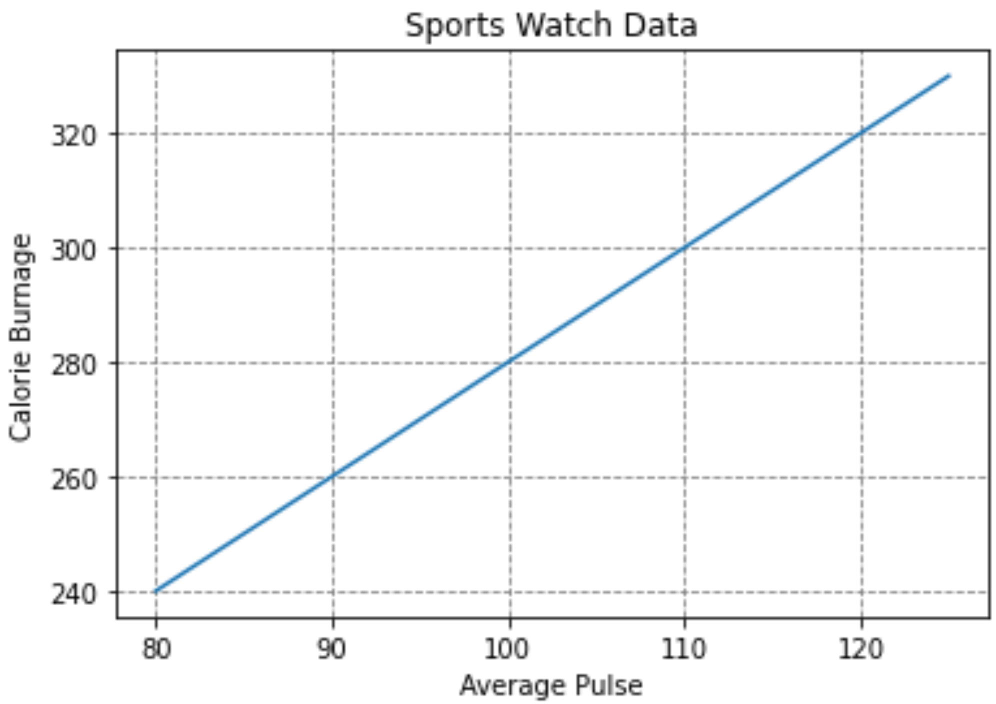
Labels

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

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y, marker='x')

plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.show()
```



Grid Lines

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

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.plot(x, y)

plt.grid(color = 'grey’,
         linestyle = '--')
plt.show()
```



Exercise

- Use Matplotlib to draw the plot of the below function:

$$y = x^2$$

  where x = [-100, -99, -98, … , 0, …, 99, 100]

- Show proper labels for x-axis and y-axis

- The plot curve should be in black color

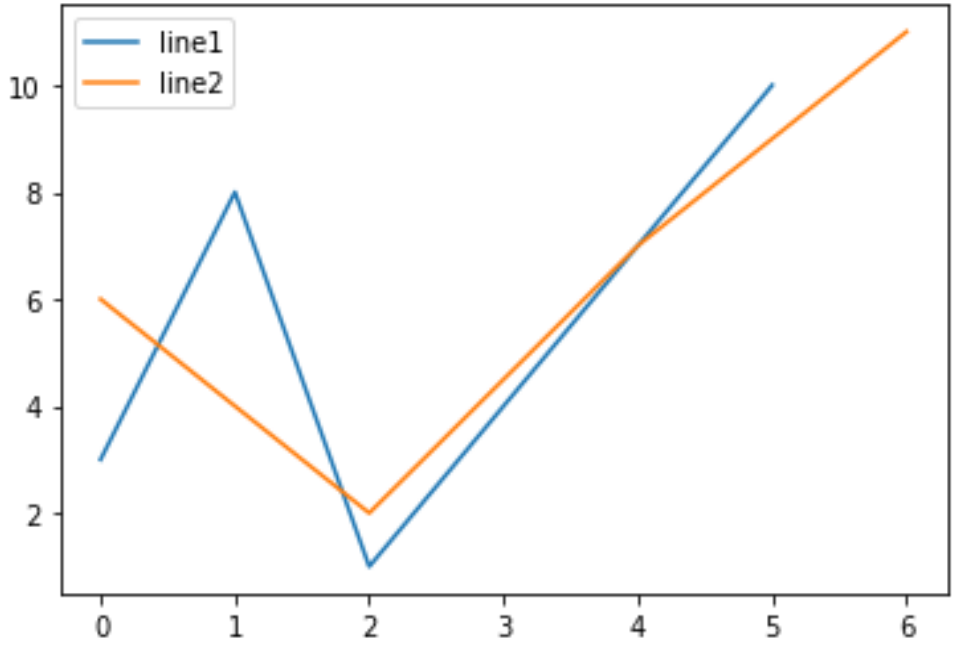
- The plot has grid lines

Exercise Solution

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

x = list(range(-100,100))
y = [i*i for i in x]

xpoints = np.array(x)
ypoints = np.array(y)

plt.plot(xpoints, ypoints, color='black')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid()
plt.show()
```

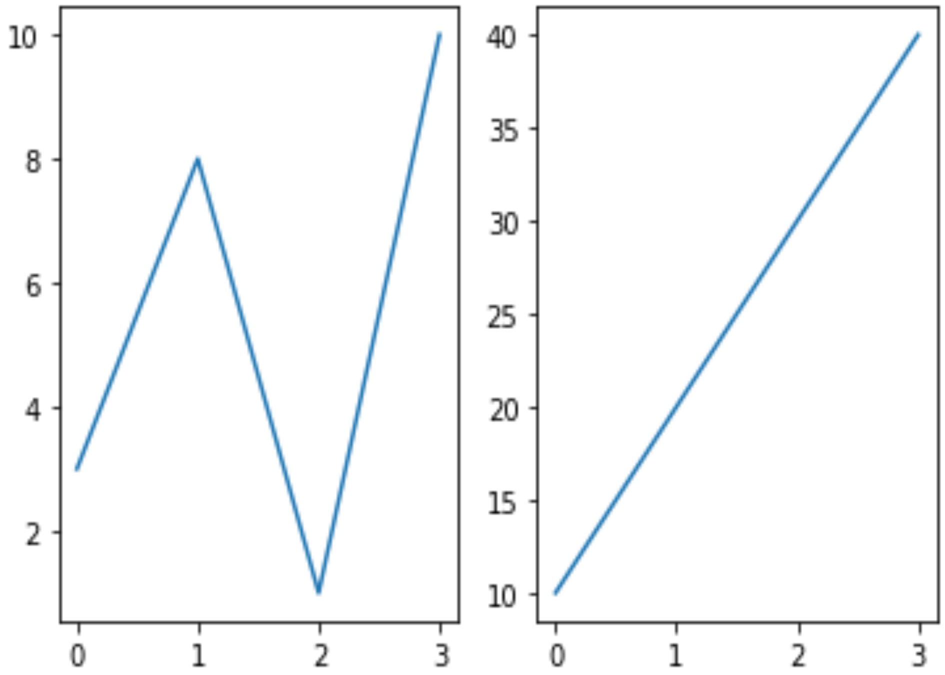
Multiple Plots

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

x1 = np.array([0, 1, 2, 5])
y1 = np.array([3, 8, 1, 10])
x2 = np.array([0, 2, 4, 6])
y2 = np.array([6, 2, 7, 11])

plt.plot(x1, y1, label="line1")
plt.plot(x2, y2, label="line2")
plt.legend(loc='upper left')
plt.show()
```



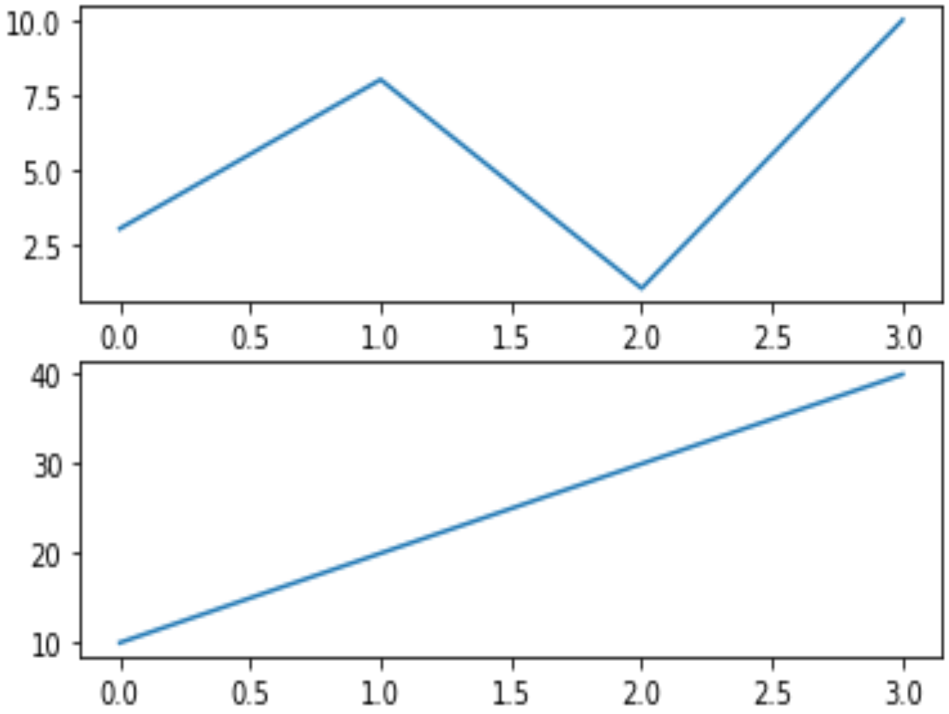
Subplots

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

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
#the figure has 1 row, 2 columns, and
this plot is the first plot.

plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
#the figure has 1 row, 2 columns, and
this plot is the second plot.

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



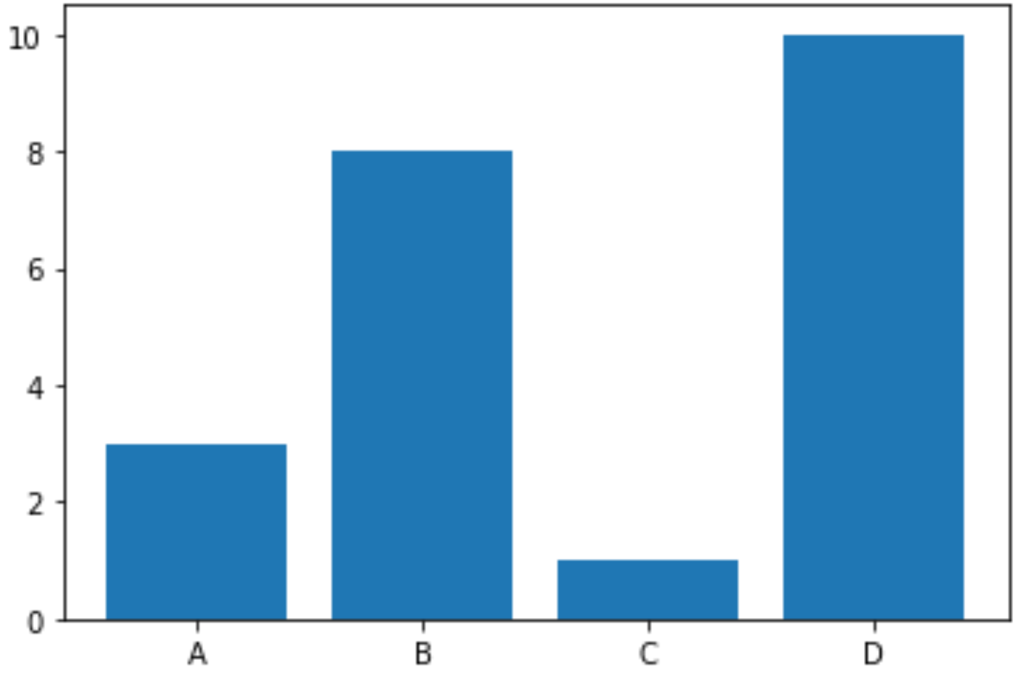
Subplots

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

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 1, 1)
#the figure has 2 rows, 1 column, and
this plot is the first plot.

plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 1, 2)
#the figure has 2 rows, 1 column, and
this plot is the second plot.

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



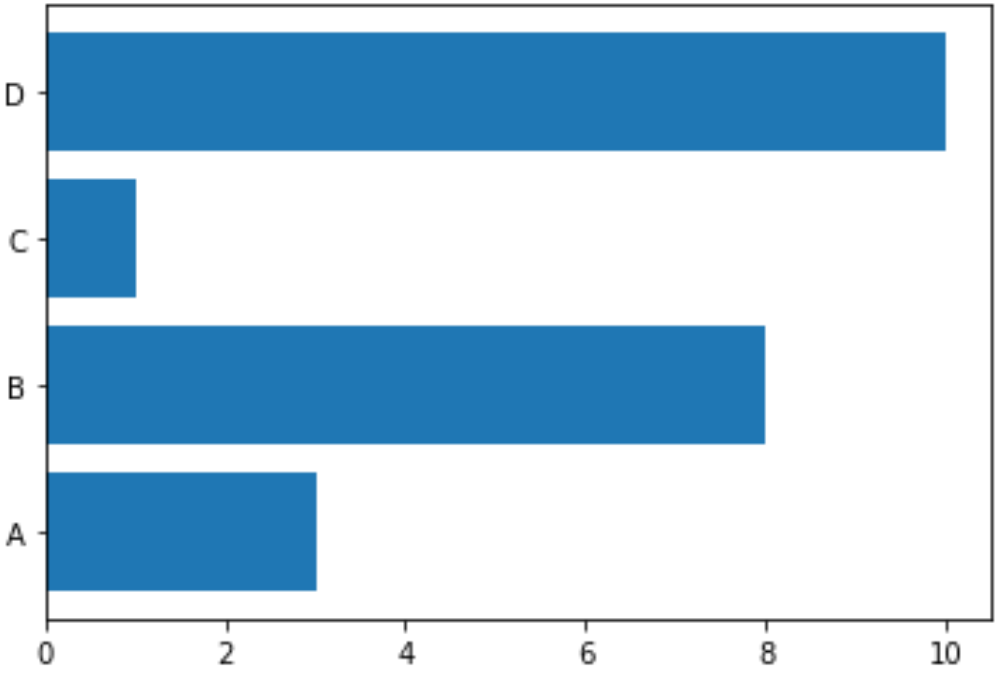
Bar Plot

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

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y)
plt.show()
```



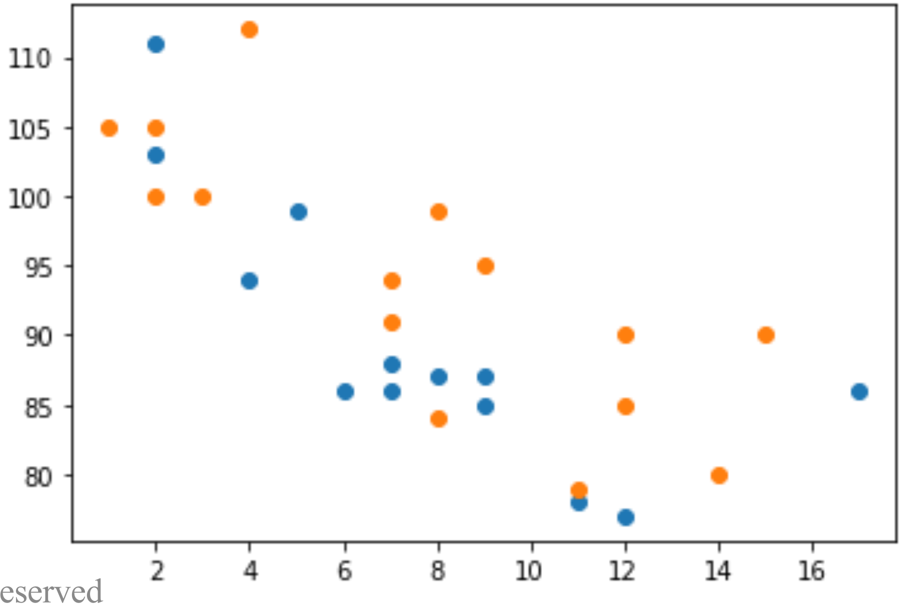
Horizontal Bar Plot

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

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.barh(x,y)
plt.show()
```



Scatter Plot

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

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y)

x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y)

plt.show()
```



Data Visualization Example
Iris Types

- We are given a excel sheet (iris.csv) that include five columns that describe information about three types of iris flowers, as follows:
  - Column 0: sepal length
  - Column 1: sepal width
  - Column 2: petal length
  - Column 3: petal width
  - Column 4: iris type

```
// the first five rows of iris.csv
5.1,3.5,1.4,0.2,1
4.9,3.0,1.4,0.2,1
4.7,3.2,1.3,0.2,1
4.6,3.1,1.5,0.2,1
5.0,3.6,1.4,0.2,1
```

Data Visualization Example
Reading the File

```
def read_data(filename, data) :

    # Read in data from file line by line
    infile = open(filename, 'r')
    line = infile.readline()

    while line :
        line = line.strip()
        data.append(line.split(','))
        line = infile.readline()
        infile.close()

    # Convert continuous attributes to float and class labels to integers
    for i in range(0, len(data)) :
        data[i][0] = float(data[i][0])
        data[i][1] = float(data[i][1])
        data[i][2] = float(data[i][2])
        data[i][3] = float(data[i][3])
        data[i][4] = int(data[i][4])
```

Data Visualization Example
Creating the Arrays

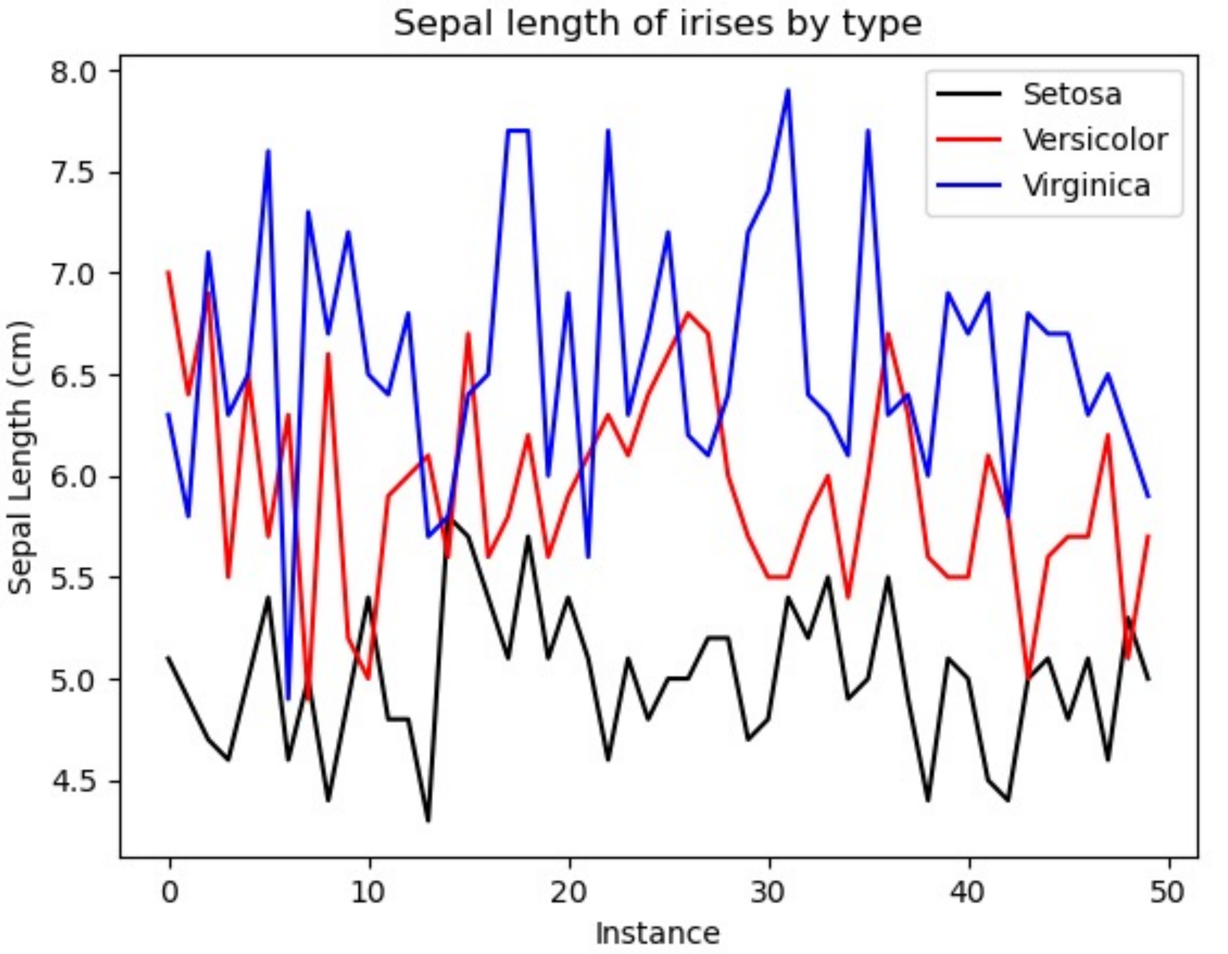
```
# make an empty data List
data = []
read_data('iris.csv', data)

# Divide up the data set by type of iris
setosa = []
versicolor = []
virginica = []

for instance in data :
    if(instance[4] == 1) :
        setosa.append(instance)
    elif(instance[4] == 2) :
        versicolor.append(instance)
    else :
        virginica.append(instance)

# convert to numpy arrays
setosa_arr = np.array(setosa)
versicolor_arr = np.array(versicolor)
virginica_arr = np.array(virginica)
```

Data Visualization Example
Plot of Sepal Length

```
# Sepal length line plot
plt.plot(setosa_arr[:, 0], color='black', label='Setosa')
plt.plot(versicolor_arr[:, 0], color='red', label='Versicolor')
plt.plot(virginica_arr[:, 0], color='blue', label='Virginica')
plt.legend(loc='upper right')
plt.ylabel('Sepal Length (cm)')
plt.xlabel('Instance')
plt.title('Sepal length of irises by type')
plt.savefig('sepal_length_type.png')
plt.close()
```

This line will save the plot as an image on your machine

Data Visualization Example
Plot of Sepal Length

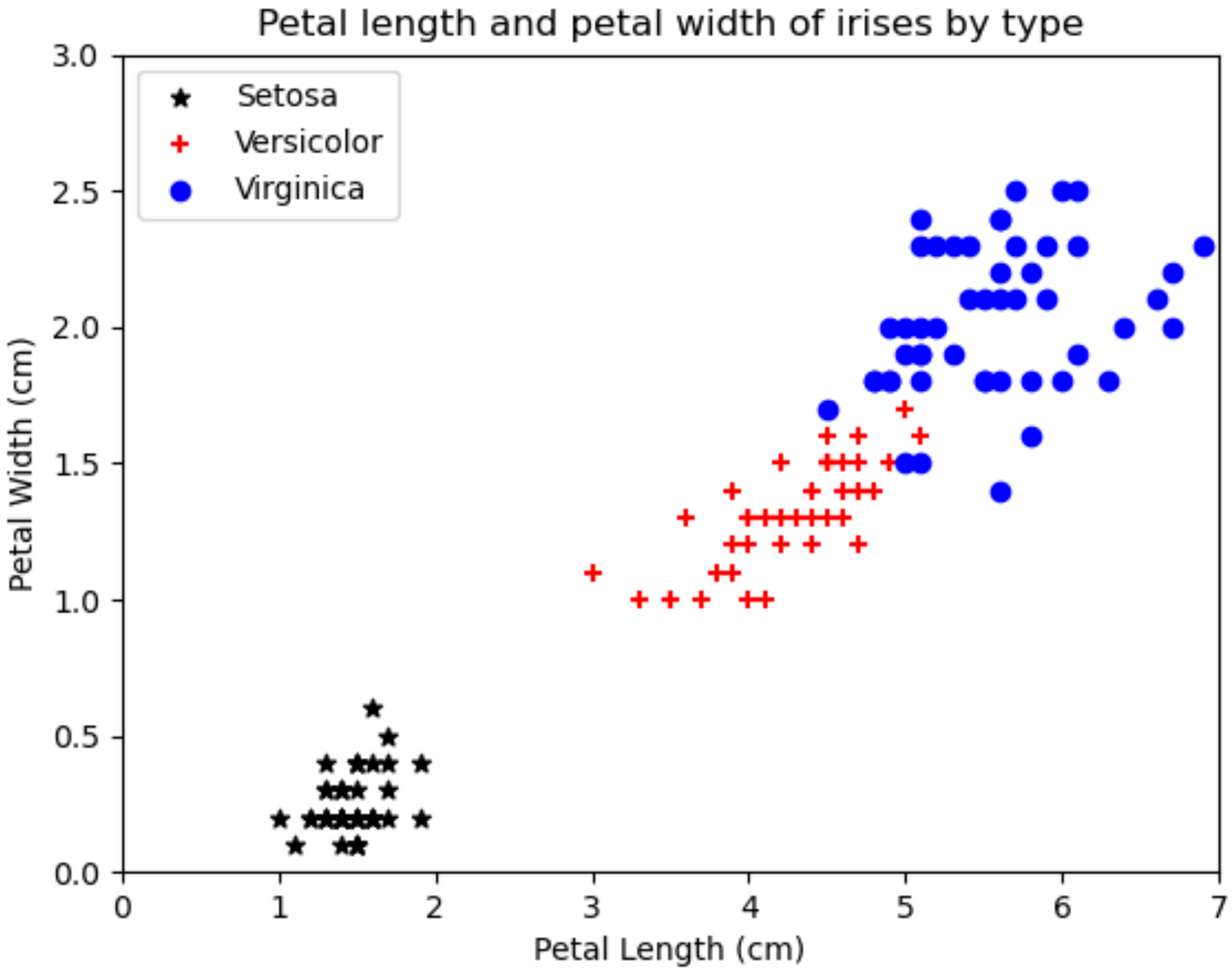


Data Visualization Example
Plot of Petal Length vs Petal Width

```
# Petal length and petal width scatter plot
plt.scatter(setosa_arr[:, 2], setosa_arr[:, 3],
            color='black', label='Setosa', marker='*’)

plt.scatter(versicolor_arr[:, 2], versicolor_arr[:, 3],
            color='red', label='Versicolor', marker='+’)

plt.scatter(virginica_arr[:, 2], virginica_arr[:, 3],
            color='blue', label='Virginica', marker='o’)

plt.legend(loc='upper left')
plt.ylabel('Petal Width (cm)')
plt.xlabel('Petal Length (cm)')
plt.title('Petal length and petal width of irises by type')
plt.xlim(0,7)
plt.ylim(0,3)
plt.savefig('petal_length_width_scatter.png')
plt.close()
```

Control x-axis and
y-axis limits

Data Visualization Example
Plot of Petal Length vs Petal Width



Summary

- Matplotlib is a popular data visualization library in Python
- Matplotlib provides customizable plots and layouts
- We covered some important features in Matplotlib, but there is still many features that you can explore yourself