**Definition: Inner and Outer Product If u and v are column vectors with the same size, then u T v is the inner product of u and v; if u and v are column vectors of any size, then uvT is the outer product of u and v.**

**1.Data visualisation using matplotlib**

# initializing the data

x = [10, 20, 30, 40]

y = [20, 25, 35, 55]

# plotting the data

plt.plot(x, y)

plt.show()

**# Adding title to the plot**

plt.title("Linear graph")

plt.show()

plt.title("Linear graph", fontsize=25, color="green")

plt.show()

**Adding X-label, Y-Label**

**X label and the Y label are the titles given to X-axis and Y-axis respectively**

# Adding label on the y-axis

plt.ylabel('Y-Axis')

# Adding label on the x-axis

plt.xlabel('X-Axis')

plt.show()

**#Setting the limit of y-axis**

plt.ylim(0, 80)

# setting the labels of x-axis

plt.xticks(x, labels=["one", "two", "three", "four"])

plt.show()

**# Adding legends**

plt.legend(["GFG"])

plt.show()

# **Add Axes to a Figure**

Import libraries

import matplotlib

import numpy

# Create figure() objects

# This acts as a container

# for the different plots

fig = matplotlib.pyplot.figure()

# Generate line graph

x = numpy.arange(0, 1.414\*2, 0.05)

y1 = numpy.sin(x)

y2 = numpy.cos(x)

# Creating two axes

# add\_axes([xmin,ymin,dx,dy])

axes1 = fig.add\_axes([0, 0, 1, 1])

axes1.plot(x, y1)

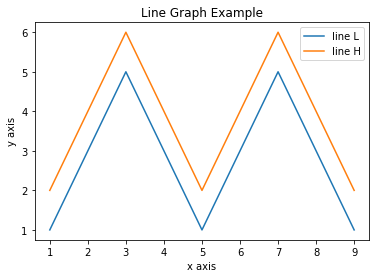
axes2 = fig.add\_axes([0, 1, 1, 1])

axes2.plot(x, y2)

# Show plot

plt.show()

### Line Plots



import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5, 6, 7, 8, 9]

y1 = [1, 3, 5, 3, 1, 3, 5, 3, 1]

y2 = [2, 4, 6, 4, 2, 4, 6, 4, 2]

plt.plot(x, y1, label="line L")

plt.plot(x, y2, label="line H")

plt.plot()

plt.xlabel("x axis")

plt.ylabel("y axis")

plt.title("Line Graph Example")

plt.legend()

plt.show()

### Bar Plots

import matplotlib.pyplot as plt

# Look at index 4 and 6, which demonstrate overlapping cases.

x1 = [1, 3, 4, 5, 6, 7, 9]

y1 = [4, 7, 2, 4, 7, 8, 3]

x2 = [2, 4, 6, 8, 10]

y2 = [5, 6, 2, 6, 2]

# Colors: https://matplotlib.org/api/colors\_api.html

plt.bar(x1, y1, label="Blue Bar", color='b')

plt.bar(x2, y2, label="Green Bar", color='g')

plt.plot()

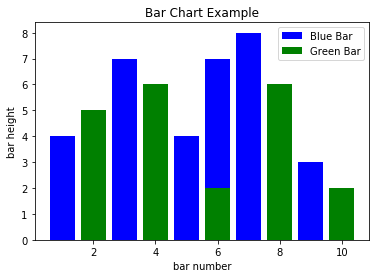
plt.xlabel("bar number")

plt.ylabel("bar height")

plt.title("Bar Chart Example")

plt.legend()

plt.show()



**Histograms**

import matplotlib.pyplot as plt

import numpy as np

# Use numpy to generate a bunch of random data in a bell curve around 5.

n = 5 + np.random.randn(1000)

m = [m for m in range(len(n))]

plt.bar(m, n)

plt.title("Raw Data")

plt.show()

plt.hist(n, bins=20)

plt.title("Histogram")

plt.show()

plt.hist(n, cumulative=True, bins=20)

plt.title("Cumulative Histogram")

plt.show()

### 

### 

### Scatter Plots

import matplotlib.pyplot as plt

x1 = [2, 3, 4]

y1 = [5, 5, 5]

x2 = [1, 2, 3, 4, 5]

y2 = [2, 3, 2, 3, 4]

y3 = [6, 8, 7, 8, 7]

# Markers: https://matplotlib.org/api/markers\_api.html

plt.scatter(x1, y1)

plt.scatter(x2, y2, marker='v', color='r')

plt.scatter(x2, y3, marker='^', color='m')

plt.title('Scatter Plot Example')

plt.show()

