

**DATA SCIENCE LAB**

**Experiment No.: 4**

**Aim**

Programs using matplotlib / plotly / bokeh / seaborn for data visualisation.

.

**Procedure**

**1.Histogram**

import matplotlib.pyplot as plt

import numpy as np

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")

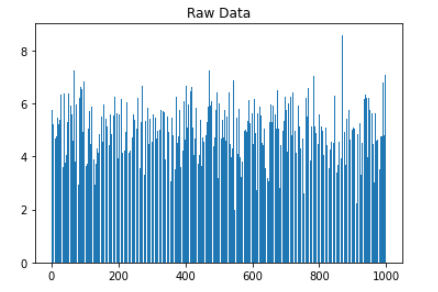
plt.show()

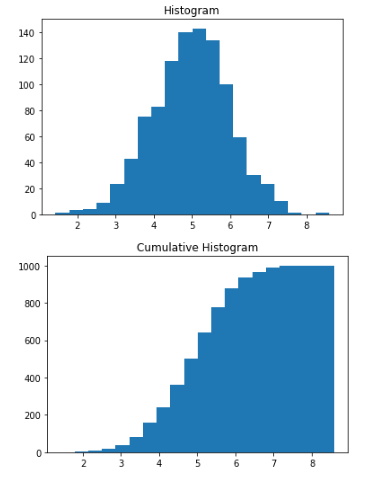
plt.hist(n, cumulative=True, bins=20)

plt.title("Cumulative Histogram")

plt.show()

**Output Screenshot**





**2. Distribution chart**

import numpy

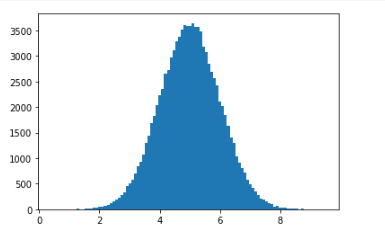
import matplotlib.pyplot as plt

x = numpy.random.normal(5.0, 1.0, 100000)

plt.hist(x, 100)

plt.show()

**Output Screenshot**



**2. 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]

plt.scatter(x1, y1)

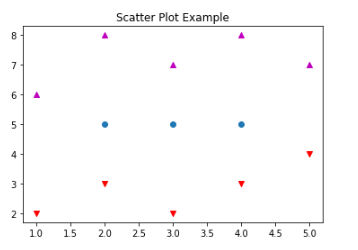
plt.scatter(x2, y2, marker='v', color='r')

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

plt.title('Scatter Plot Example')

plt.show()

**Output Screenshot**



**2. Bubble Chart**

import matplotlib.pyplot as plt

import numpy as np

x = np.random.rand(40)

y = np.random.rand(40)

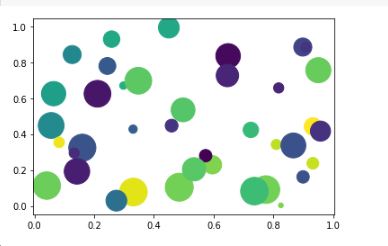
z = np.random.rand(40)

colors = np.random.rand(40)

plt.scatter(x, y, s=z\*1000,c=colors)

plt.show()

**Output Screenshot**



**Line graph**

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()

**output**



**Bar chart**

import matplotlib.pyplot as plt

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]

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

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

plt.plot()

plt.xlabel("bar number")

plt.ylabel("bar height")

plt.title("Bar Chart Example")

plt.legend()

plt.show()

**output**



**Box plot**

plt.figure()

plt.suptitle("Boxplot for X vs Y split into 5 bins")

ax = plt.gca()

df2.boxplot(showmeans=True)

# Rotate x axis text values

for tick in ax.get\_xticklabels():

    tick.set\_rotation(30)

print("\nIn the boxplot below, the box extends from the lower to upper quartile values of the data, with a line at the median.\n \

The whiskers extend from the box to show the range of the data. The triangle indicates the mean value.\n")

**output**

In the boxplot below, the box extends from the lower to upper quartile values of the data, with a line at the median.

 The whiskers extend from the box to show the range of the data. The triangle indicates the mean value.

