|  |
| --- |
| **Name: MALAVIKA A**  **Roll No: 16**  **Batch: RMCA B**  **Date: 22-08-2022** |

**DATA SCIENCE LAB**

**Experiment No.: 1**

**Aim**

Scatter Plots and SVD

**Procedure**

1. **Bubble plot**

import matplotlib.pyplot as plt

import numpy as np

x = np.random.rand(40)

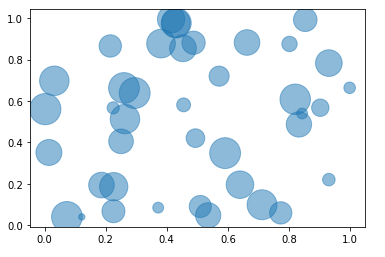
y = np.random.rand(40)

z = np.random.rand(40)

plt.scatter(x, y, s=z\*1000, alpha=0.5)

plt.show()

**OUTPUT**



1. **Graph-2 long**

import seaborn as sns

import matplotlib.pyplot as plt

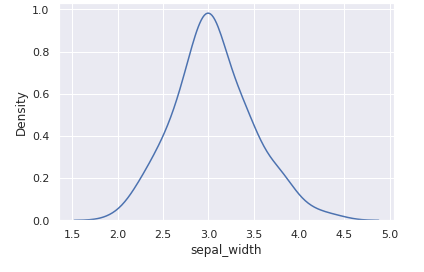
sns.set(style="darkgrid")

df = sns.load\_dataset('iris')

sns.kdeplot(df['sepal\_width'])

plt.show()

**OUTPUT**



1. **Probabaility plot**

import numpy as np

import pylab

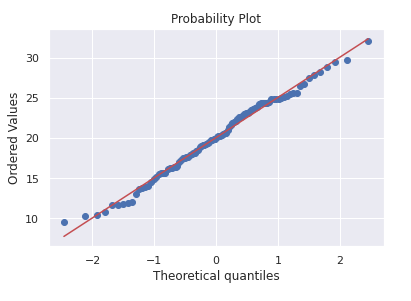
import scipy.stats as stats

measurements = np.random.normal(loc = 20, scale = 5, size=100)

stats.probplot(measurements, dist="norm", plot=pylab)

pylab.show()

**OUTPUT**



1. **Lined plot**

import numpy as np

import matplotlib.pyplot as plt

pos = 100

scale = 5

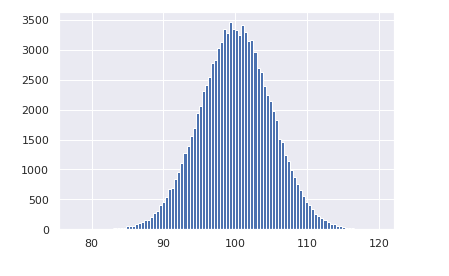
size = 100000

values = np.random.normal(pos, scale, size)

plt.hist(values, 100)

plt.show()

**OUTPUT**



1. **Zero matrix, all ones matrix, etc**

import numpy as np

a = np.array([1, 2, 3])

print("type: %s" %type(a))

print("shape: %s" %a.shape)

print(a[0], a[1], a[2])

a[0] = 5

print(a)

b = np.array([[1,2,3],[4,5,6]])

print("\n shape of b:",b.shape)

print(b[0, 0], b[0, 1], b[1, 0])

a = np.zeros((2,2))

print("All zeros matrix:\n  %s" %a)

b = np.ones((1,2))

print("\nAll ones matrix:\n  %s" %b)

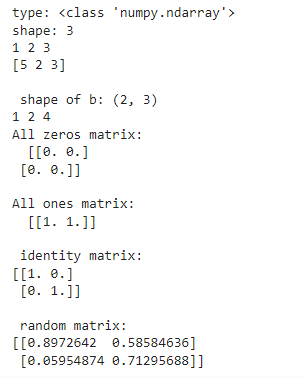
d = np.eye(2)

print("\n identity matrix: \n%s"%d)

e = np.random.random((2,2))

print("\n random matrix: \n%s"%e)

**OUTPUT**



1. **vectorized sum**

#vectorized sum

print("Vectorized sum example\n")

x = np.array([[1,2],[3,4]])

print("x:\n %s" %x)

print("sum: %s"%np.sum(x))  # Compute sum of all elements; prints "10"

print("sum axis = 0: %s" %np.sum(x, axis=0))  # Compute sum of each column; prints "[4 6]"

print(" sum axis = 1: %s" %np.sum(x, axis=1))  # Compute sum of each row; prints "[3 7]"

#matrix dot product

a = np.arange(10000)

b = np.arange(10000)

dp = np.dot(a,b)

print("Dot product: %s\n" %dp)

#outer product

op = np.outer(a,b)

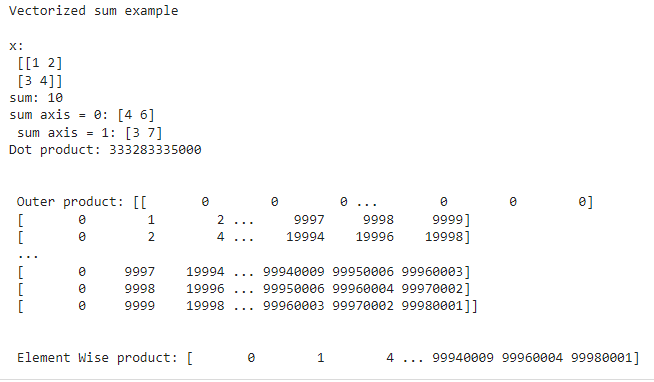
print("\n Outer product: %s\n" %op)

#elementwise product

ep = np.multiply(a, b)

print("\n Element Wise product: %s \n" %ep)

**OUTPUT**



1. **Transpose matrix**

import numpy as np

x = np.array([[1,2], [3,4]])

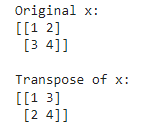
print("Original x: \n%s " %x)    # Prints "[[1 2]

            #          [3 4]]"

print("\nTranspose of x: \n%s" %x.T)  # Prints "[[1 3]

            #          [2 4]]"

**OUTPUT**



1. **SVD**

from numpy import array

from scipy.linalg import svd

A = array([[1, 2], [3, 4], [5, 6]])

print("A: \n%s" %A)

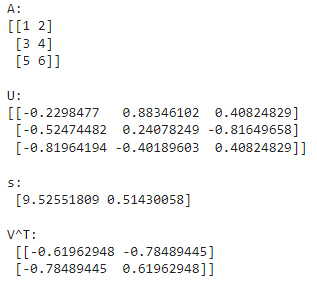
U, s, VT = svd(A)

print("\nU: \n%s" %U)

print("\ns: \n %s" %s)

print("\nV^T: \n %s" %VT)

**OUTPUT**



1. **Linear graph-1**

from matplotlib import pyplot as plt

x = [10, 20, 30, 40]

y = [20, 25, 35, 55]

plt.plot(x, y)

plt.title("Linear graph")

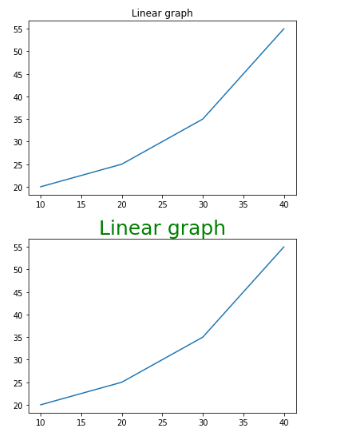
plt.show()

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

plt.plot(x, y)

plt.show()

**OUTPUT**



1. **Linear graph-2 (one,two,three)**

from matplotlib import pyplot as plt

plt.ylim(0, 80)

x = [10, 20, 30, 40]

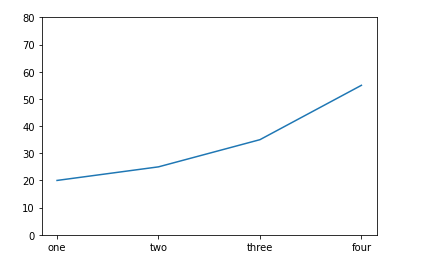
y = [20, 25, 35, 55]

plt.plot(x, y)

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

plt.show()

**OUTPUT**



1. **Linear graph-3(GFG)**

from matplotlib import pyplot as plt

x = [10, 20, 30, 40]

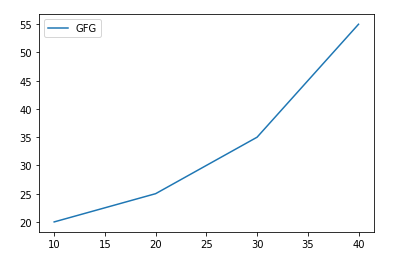
y = [20, 25, 35, 55]

plt.plot(x, y)

plt.legend(["GFG"])

plt.show()

**OUTPUT**



1. **Linear graph (axes)**

import matplotlib

import numpy

fig = matplotlib.pyplot.figure()

x = numpy.arange(0, 1.414\*2, 0.05)

y1 = numpy.sin(x)

y2 = numpy.cos(x)

axes1 = fig.add\_axes([0, 0, 1, 1])

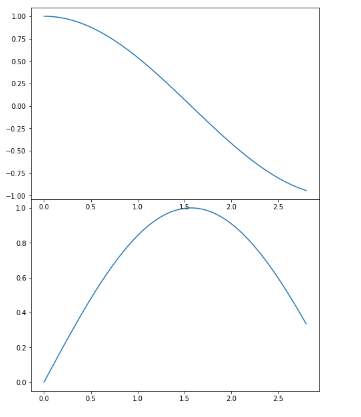
axes1.plot(x, y1)

axes2 = fig.add\_axes([0, 1, 1, 1])

axes2.plot(x, y2)

plt.show()

**OUTPUT**



1. **Line graph**

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")

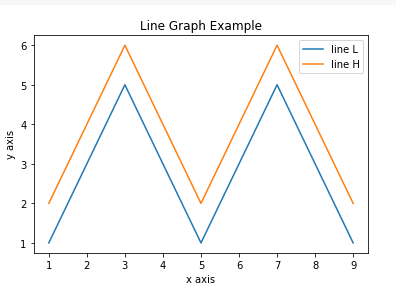
plt.ylabel("y axis")

plt.title("Line Graph Example")

plt.legend()

plt.show()

**OUTPUT**



1. **Bar chart**

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='b')

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

plt.plot()

plt.xlabel("bar number")

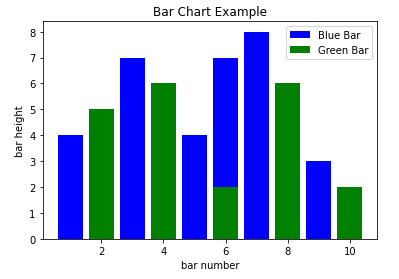
plt.ylabel("bar height")

plt.title("Bar Chart Example")

plt.legend()

plt.show()

**OUTPUT**



1. **Raw data**

import matplotlib.pyplot as plt

import numpy as np

n = 5 + np.random.randn(1000)

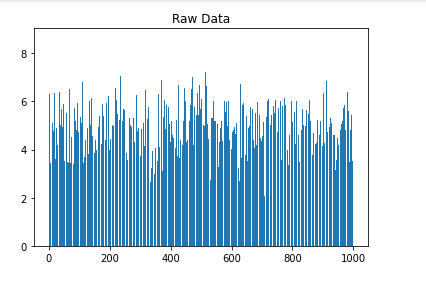
m = [m for m in range(len(n))]

plt.bar(m, n)

plt.title("Raw Data")

plt.show()

**OUTPUT**



1. **Scatter plot**

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)

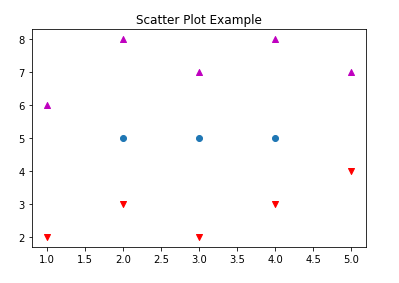
plt.scatter(x2, y2, marker='v', color='r')

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

plt.title('Scatter Plot Example')

plt.show()

**OUTPUT**



1. **Histogram**

import matplotlib.pyplot as plt

import numpy as np

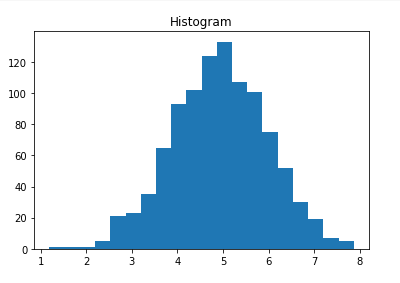
n = 5 + np.random.randn(1000)

plt.hist(n, bins=20)

plt.title("Histogram")

plt.show()

**OUTPUT**



1. **Cumulative histogram**

import matplotlib.pyplot as plt

import numpy as np

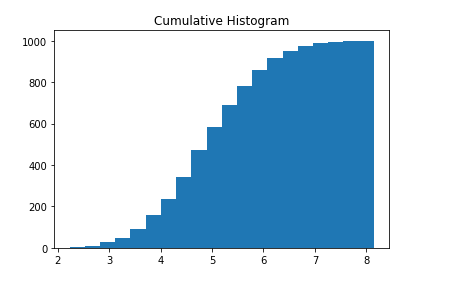
n = 5 + np.random.randn(1000)

plt.hist(n, cumulative=True, bins=20)

plt.title("Cumulative Histogram")

plt.show()

**OUTPUT**



1. **Graph (x-axis,y-axis)**

from matplotlib import pyplot as plt

x = [10, 20, 30, 40]

y = [20, 25, 35, 55]

plt.plot(x, y)

plt.ylabel('Y-Axis')

plt.xlabel('X-Axis')

plt.show()

**OUTPUT**

