OPEN SOURCE SOFTWARE LAB (15B17CI575)

Lab Assignment 5 (Practice Lab)

Odd 2021

Week 5 & 6: 28 Sept- 9 Oct

Topic Coverage: Python- Matplotlib and SciPy package

Patil Amit Gurusidhappa

19104004

B11

Practice Questions

1.

import matplotlib.pyplot as plt

import numpy as np

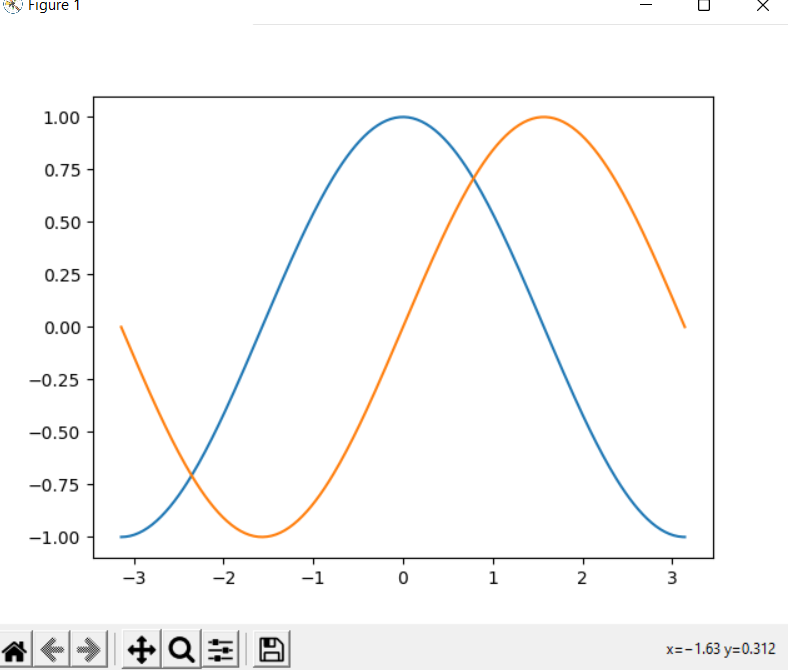
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)

C, S = np.cos(X), np.sin(X)

plt.plot(X, C)

plt.plot(X, S)

plt.show()



2.

import matplotlib.pyplot as plt

import numpy as np

X = np.linspace(-np.pi, np.pi, 256, endpoint=True)

C, S = np.cos(X), np.sin(X)

# Create a figure of size 8x6 inches, 80 dots per inch

plt.figure(figsize=(8, 6), dpi=80)

# Create a new subplot from a grid of 1x1

plt.subplot(1, 1, 1)

# Plot cosine with a blue continuous line of width 1 (pixels)

plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-")

# Plot sine with a green continuous line of width 1 (pixels)

plt.plot(X, S, color="green", linewidth=1.0, linestyle="-")

# Set x limits

plt.xlim(-4.0, 4.0)

# Set x ticks

plt.xticks(np.linspace(-4, 4, 9, endpoint=True))

# Set y limits

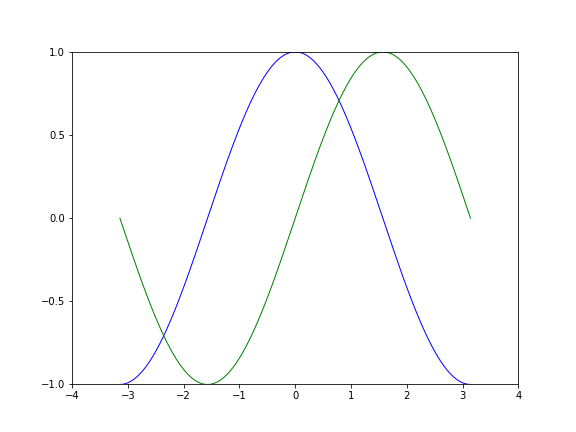
plt.ylim(-1.0, 1.0)

# Set y ticks

plt.yticks(np.linspace(-1, 1, 5, endpoint=True))

# Save figure using 72 dots per inch

plt.savefig("exercise\_2.png", dpi=72)



3.

import matplotlib.pyplot as plt

import numpy as np

X = np.linspace(-np.pi, np.pi, 256, endpoint=True)

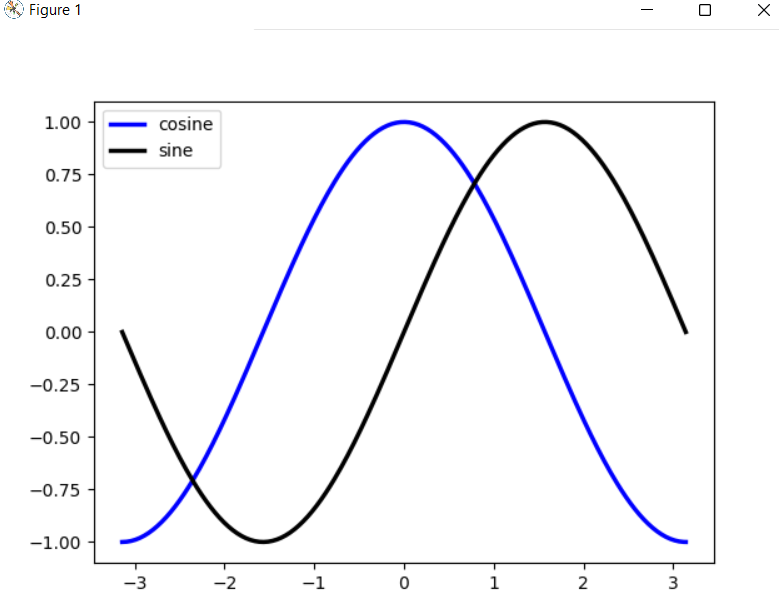
C, S = np.cos(X), np.sin(X)

plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-", label="cosine")

plt.plot(X, S, color="black", linewidth=2.5, linestyle="-", label="sine")

plt.legend(loc='upper left')

plt.show()



4.

import matplotlib.pyplot as plt

import numpy as np

n = 256

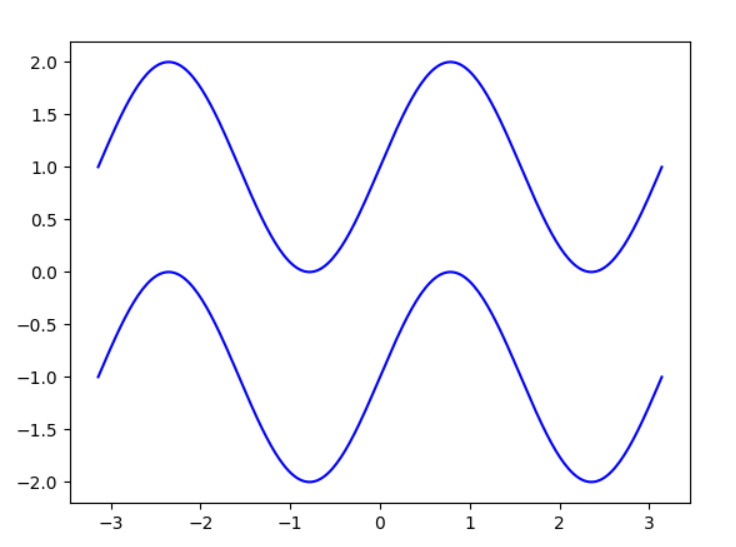
X = np.linspace(-np.pi, np.pi, n, endpoint=True)

Y = np.sin(2 \* X)

plt.plot(X, Y + 1, color='blue', alpha=1.00)

plt.plot(X, Y - 1, color='blue', alpha=1.00)

plt.show()



5.

import matplotlib.pyplot as plt

import numpy as np

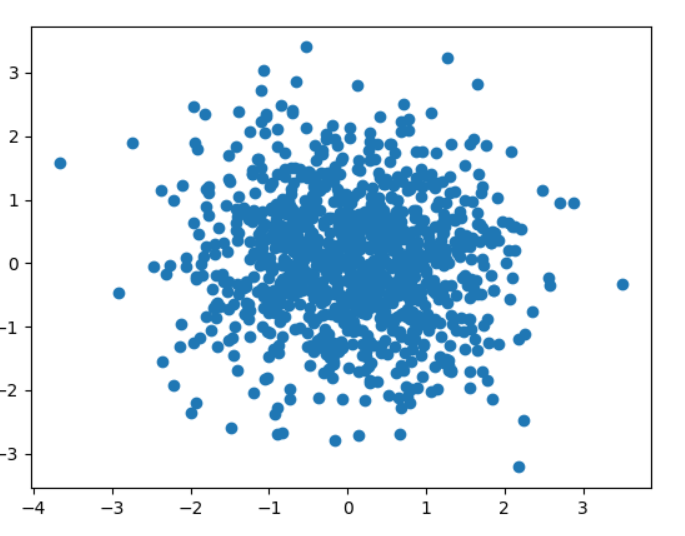
n = 1024

X = np.random.normal(0, 1, n)

Y = np.random.normal(0, 1, n)

plt.scatter(X, Y)

plt.show()



6.

import matplotlib.pyplot as plt

import numpy as np

n = 12

X = np.arange(n)

Y1 = (1 - X / float(n)) \* np.random.uniform(0.5, 1.0, n)

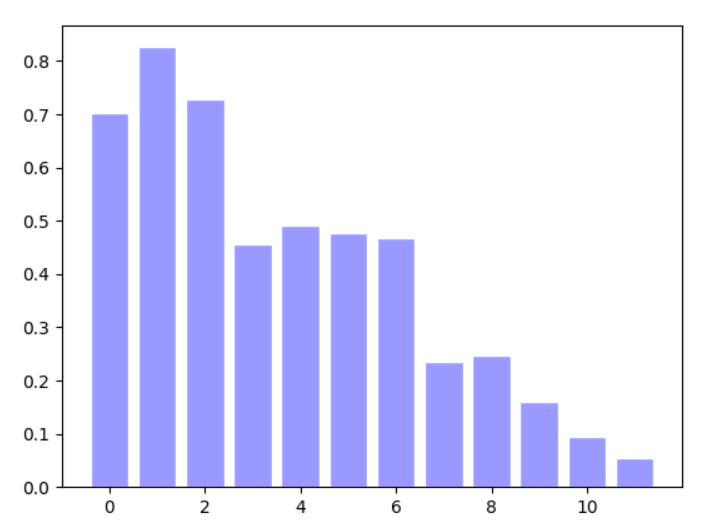
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')

plt.show()

Y2 = (1 - X / float(n)) \* np.random.uniform(0.5, 1.0, n)

plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')

plt.show()



Exercise Questions

1.

import numpy as np

from scipy import io as sci

array = np.ones((4, 4))

sci.savemat('example.mat', {'ar': array})

data = sci.loadmat('example.mat', struct\_as\_record=True)

print(data['ar'])

"""

[[1. 1. 1. 1.]

 [1. 1. 1. 1.]

 [1. 1. 1. 1.]

 [1. 1. 1. 1.]]

"""

2.

from scipy.special import cbrt

cubeRoot = cbrt(2764891)

print(cubeRoot)

"""

140.35439126462808

"""

3.

import numpy as np

from scipy import linalg

A = np.array([[4, 5], [3, 2]])

det = linalg.det(A)

print(det)

"""

-7.0

"""

4.

import numpy as np

from scipy import linalg

A = np.array([[4, 5], [3, 2]])

inverse = linalg.inv(A)

print(A)

print(inverse)

"""

[[4 5]

 [3 2]]

[[-0.28571429  0.71428571]

 [ 0.42857143 -0.57142857]]

"""

5.

import numpy as np

from scipy import linalg

A = np.array([[4, 5], [3, 2]])

eval, evector = linalg.eig(A)

print(eval)

print(evector)

"""

[ 7.+0.j -1.+0.j]

[[ 0.85749293 -0.70710678]

 [ 0.51449576  0.70710678]]

"""