1

import numpy as np

import matplotlib.pyplot as plt

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

#Figure of size 8x6 inches, 80 dots per inch

plt.figure(figsize=(8, 6), dpi=80)

#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

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

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

plt.legend(loc='upper left')

plt.show()

4

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

n = 1024

X = np.random.normal(0,1,n)

Y = np.random.normal(0,1,n)

plt.scatter(X,Y)

plt.show()

6

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()

Without Solution Questn

1)

import numpy as np

import matplotlib.pyplot as plt

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

T=np.tan(X)

SEC=np.arccos(X)

COT=np.arctan(X)

COSEC=np.arcsin(X)

plt.plot(X, T)

plt.show()

plt.plot(X,SEC)

plt.show()

plt.plot(X,COT)

plt.show()

plt.plot(X,COSEC)

plt.show()

2)

import numpy as np

import matplotlib.pyplot as plt

x = ['A','B','C','D']

Result1 = [2,5,8,5]

Result2 = [3,2,5,7]

x\_axis = np.arange(len(x))

plt.bar(x\_axis - 0.2, Result1, 0.4, color = 'green', label = 'Result1')

plt.bar(x\_axis + 0.2, Result2, 0.4, color = 'yellow', label = 'Result2')

plt.xticks(x\_axis, x)

plt.xlabel('Method')

plt.ylabel('Result')

plt.legend()

plt.show()