# M Bharath kumar

# cl2-batch

# PROGRAMME

INPUT:

import matplotlib.pyplot as plt

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

height = [10,20,30,40,5]

tick\_label =['one','two','three','four','five']

plt.bar(left,height,tick\_label = tick\_label,width=0.8,color=['red','green'])

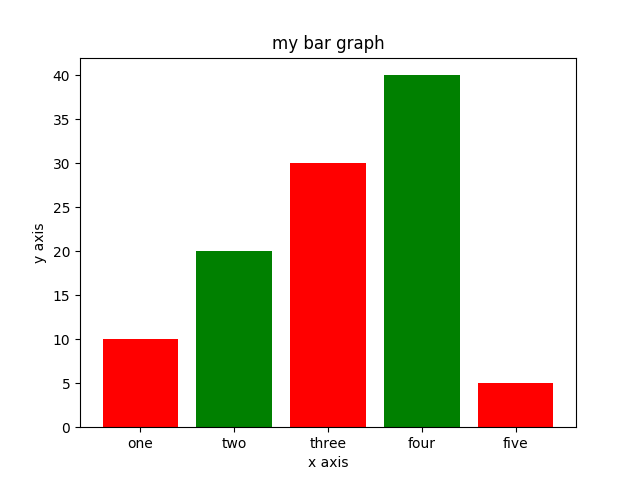
plt.xlabel('x axis')

plt.ylabel('y axis')

plt.title('my bar graph')

plt.show

OUTPUT:



# PROGRAMME

INPUT:

import matplotlib.pyplot as plt

x=[1,2,3]

y=[1,5,9]

plt.plot(x,y)

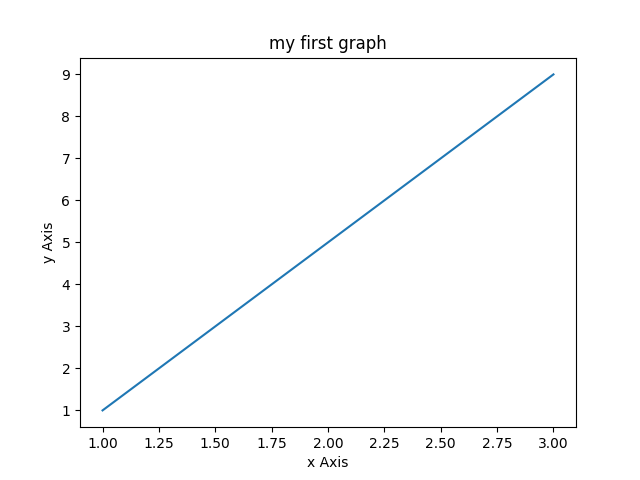
plt.xlabel("x Axis")

plt.ylabel("y Axis")

plt.title("my first graph")

plt.show()

OUTPUT:



# PROGRAMME

INPUT:

import numpy as np

import matplotlib.pyplot as plt

# set width of bar

barWidth = 0.25

# set height of bar

bars1 = [12, 30, 1, 8, 22]

bars2 = [28, 6, 16, 5, 10]

bars3 = [29, 3, 24, 25, 17]

# Set position of bar on X axis

r1 = np.arange(len(bars1))

r2 = [x + barWidth for x in r1]

r3 = [x + barWidth for x in r2]

# Make the plot

plt.bar(r1, bars1, color='#7f6d5f', width=barWidth, edgecolor='white', label='var1')

plt.bar(r2, bars2, color='#557f2d', width=barWidth, edgecolor='white', label='var2')

plt.bar(r3, bars3, color='#2d7f5e', width=barWidth, edgecolor='white', label='var3')

# Add xticks on the middle of the group bars

plt.xlabel('group', fontweight='bold')

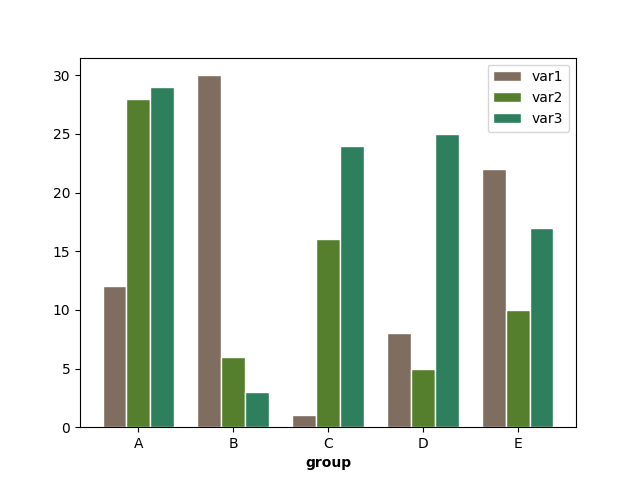
plt.xticks([r + barWidth for r in range(len(bars1))], ['A', 'B', 'C', 'D', 'E'])

# Create legend & Show graphicN

plt.legend()

plt.show()

OUTPUT:



# PROGRAMME

INPUT:

from matplotlib import pyplot as plt

from matplotlib import style

from numpy import genfromtxt

data = genfromtxt('graphi.csv',delimiter='o')

plt.plot(data)

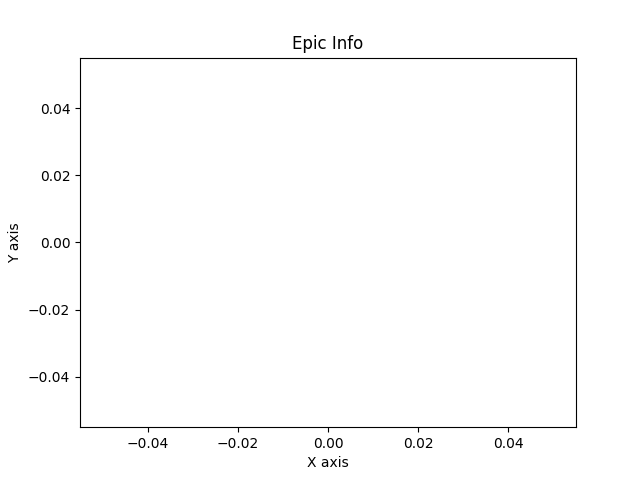
plt.title('Epic Info')

plt.ylabel('Y axis')

plt.xlabel('X axis')

plt.show()

OUTPUT:



# PROGRAMME

INPUT:

#graph

import matplotlib.pyplot as plt

import csv

x=[]

y=[]

z=[]

with open('graph.csv', 'r') as csvfile:

plots= csv.reader(csvfile, delimiter=',')

for row in plots:

x.append(int(row[0]))

y.append(int(row[1]))

z.append(int(row[2]))

plt.plot(x,y, marker='o')

plt.plot(x,z, marker='o')

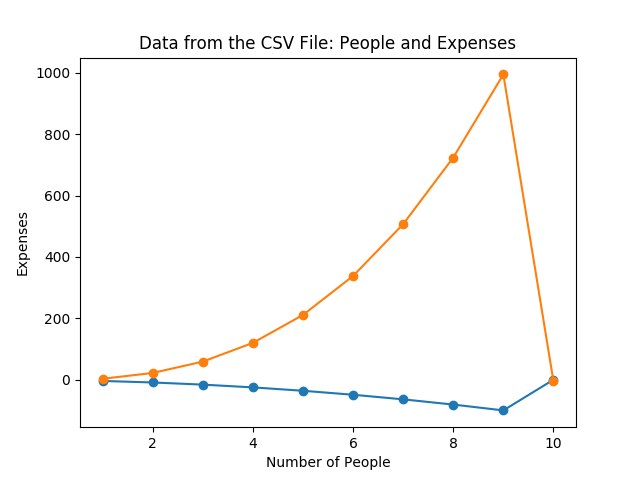
plt.title('Data from the CSV File: People and Expenses')

plt.xlabel('Number of People')

plt.ylabel('Expenses')

plt.show()

OUTPUT:



# PROGRAMME

INPUT:

class graph:

def \_\_init\_\_(self,gdict=None):

if gdict is None:

gdict = []

self.gdict = gdict

# Get the keys of the dictionary

def getVertices(self):

return list(self.gdict.keys())

# Create the dictionary with graph elements

graph\_elements = { "a" : ["b","c"],

"b" : ["a", "d"],

"c" : ["a", "d"],

"d" : ["e"],

"e" : ["d"]

}

g = graph(graph\_elements)

print(g.getVertices())

OUTPUT:

['a', 'b', 'c', 'd', 'e']

# PROGRAMME

INPUT:

import speech\_recognition as sr

r = sr.Recognizer()

with sr.Microphone() as source:

print("Say something!")

audio = r.listen(source)

try:

print("You said: " + r.recognize\_google(audio))

except sr.UnknownValueError:

print("Google Speech Recognition could not understand audio")

except sr.RequestError as e:

print("Could not request results from Google Speech Recognition service; {0}".format(e))

OUTPUT:

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.rear

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.center\_lfe

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.side

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

Say something!