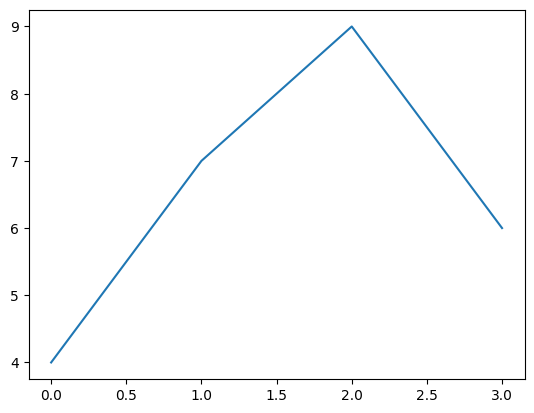
import matplotlib.pyplot as plt

lst = [4,7,9,6]  
plt.plot(lst)  
plt.show()  
#profit



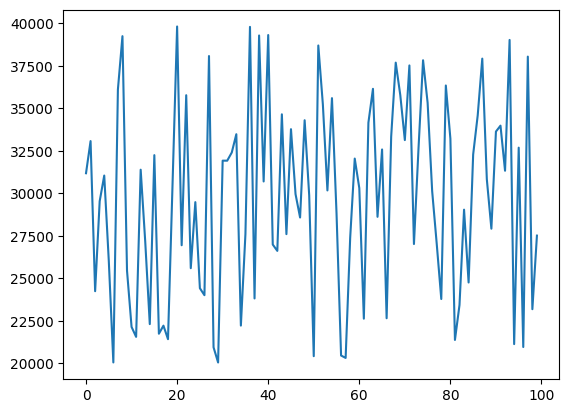
import numpy as np

salary = np.random.randint(20000,40000,100)  
salary

array([31186, 33077, 24249, 29541, 31050, 26034, 20060, 36109, 39246,  
 25462, 22149, 21564, 31394, 27167, 22314, 32250, 21748, 22223,  
 21427, 30572, 39816, 26952, 35774, 25602, 29487, 24425, 24015,  
 38078, 20951, 20058, 31926, 31924, 32405, 33480, 22227, 27610,  
 39793, 23823, 39287, 30702, 39313, 26994, 26620, 34651, 27605,  
 33777, 29959, 28583, 34303, 29890, 20428, 38698, 35238, 30175,  
 35602, 28839, 20468, 20325, 27397, 32048, 30312, 22634, 34160,  
 36150, 28623, 32584, 22659, 33341, 37690, 35778, 33135, 37525,  
 27023, 32642, 37830, 35330, 30132, 27036, 23790, 36346, 33221,  
 21383, 23439, 29044, 24761, 32290, 34594, 37926, 30833, 27925,  
 33636, 33985, 31333, 39025, 21138, 32690, 20967, 38044, 23191,  
 27519])

plt.plot(salary)

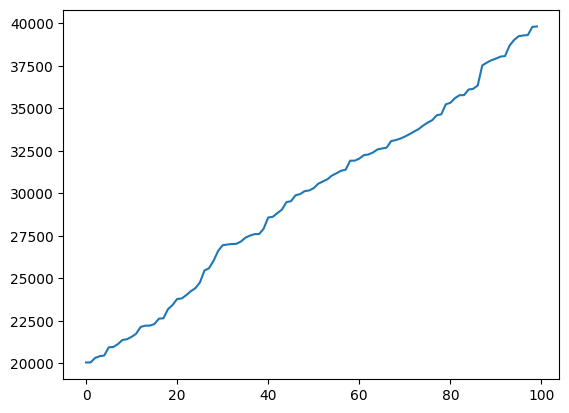
[<matplotlib.lines.Line2D at 0x1c23f465100>]



salary.sort()

plt.plot(salary)

[<matplotlib.lines.Line2D at 0x1c23f40bf50>]



Basic Markers

'.' : Point

',' : Pixel

'\*' : Star

'o' : Circle

's' : Square

'D' : Diamond

'v' : Triangle down

'^' : Triangle up

'<' : Triangle left

'>' : Triangle right

'p' : Pentagon

'h' : Hexagon

Basic Line Styles

Matplotlib supports several predefined line styles:

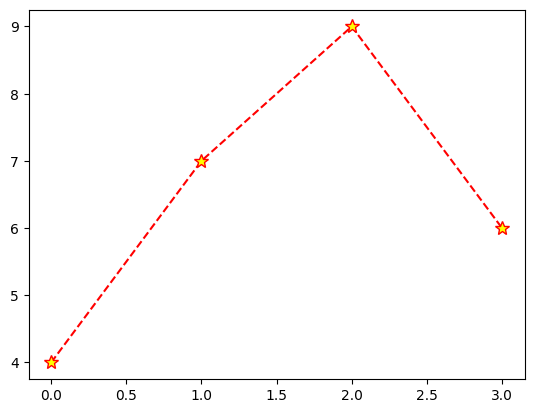
'-' or 'solid': Solid line

'--' or 'dashed': Dashed line

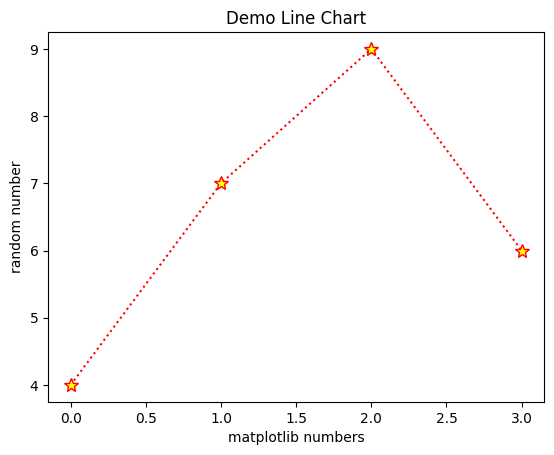
'-.' or 'dashdot': Dash-dot line

':' or 'dotted': Dotted line

lst = [4,7,9,6]  
plt.plot(lst,marker='\*',ms=10,mfc='yellow',color='red',linestyle='dashed')  
plt.show()

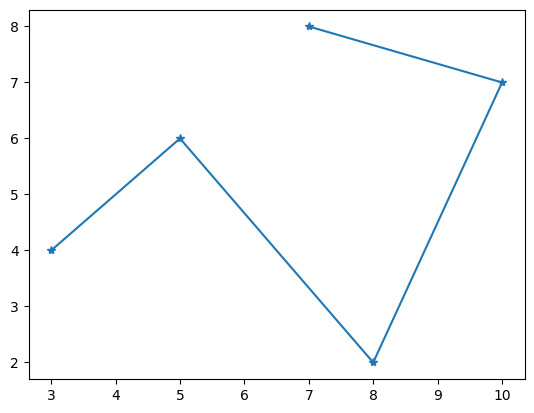


lst = [4,7,9,6]  
plt.plot(lst,marker='\*',ms=10,mfc='yellow',color='red',ls='dotted')  
plt.title("Demo Line Chart")  
plt.ylabel('random number')  
plt.xlabel('matplotlib numbers')  
plt.show()

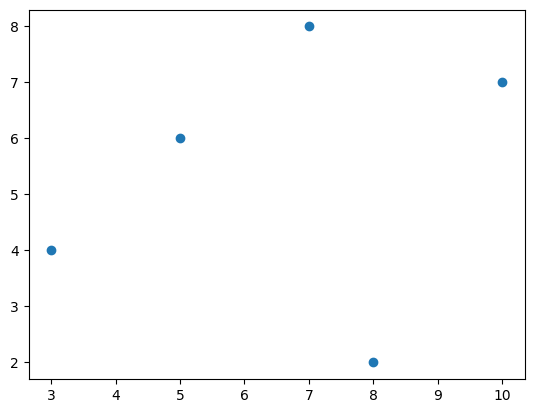


lst1 = [3,5,8,10,7]  
lst2 = [4,6,2,7,8]  
  
plt.plot(lst1,lst2,marker='\*')

[<matplotlib.lines.Line2D at 0x1c24173be30>]



lst1 = [3,5,8,10,7]  
lst2 = [4,6,2,7,8]  
plt.scatter(x=lst1,y=lst2)  
plt.show()

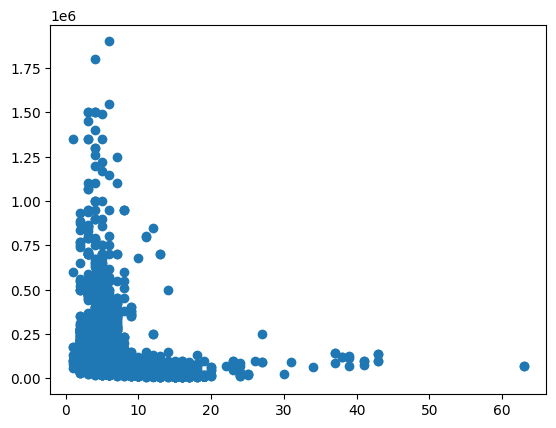


import pandas as pd

bike = pd.read\_csv('Used\_Bikes.csv')  
bike.head()

bike\_name price city kms\_driven \  
0 TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad 17654.0   
1 Royal Enfield Classic 350cc 119900.0 Delhi 11000.0   
2 Triumph Daytona 675R 600000.0 Delhi 110.0   
3 TVS Apache RTR 180cc 65000.0 Bangalore 16329.0   
4 Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore 10000.0   
  
 owner age power brand   
0 First Owner 3.0 110.0 TVS   
1 First Owner 4.0 350.0 Royal Enfield   
2 First Owner 8.0 675.0 Triumph   
3 First Owner 4.0 180.0 TVS   
4 First Owner 3.0 150.0 Yamaha

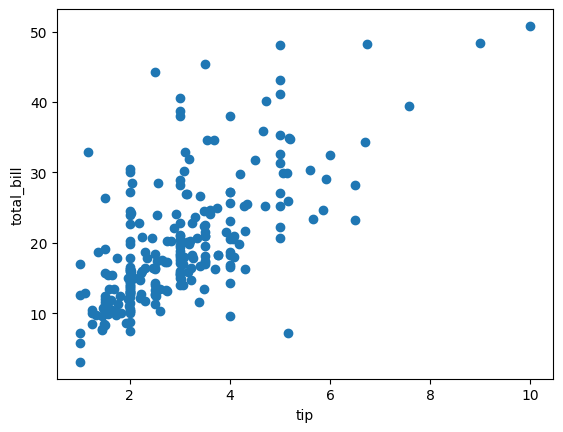
plt.scatter(bike['age'],bike['price'])  
plt.show()



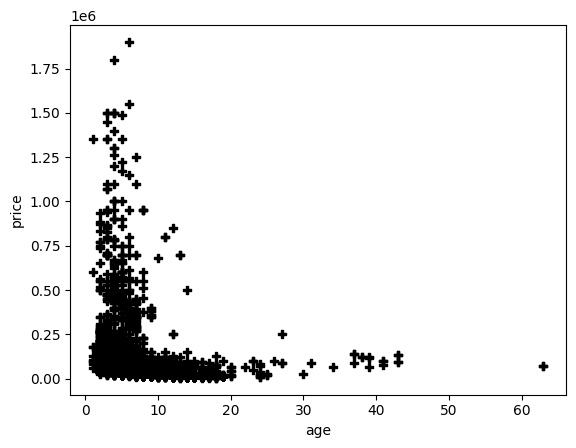
import seaborn as sns   
df = sns.load\_dataset('tips')  
df.head()

total\_bill tip sex smoker day time size  
0 16.99 1.01 Female No Sun Dinner 2  
1 10.34 1.66 Male No Sun Dinner 3  
2 21.01 3.50 Male No Sun Dinner 3  
3 23.68 3.31 Male No Sun Dinner 2  
4 24.59 3.61 Female No Sun Dinner 4

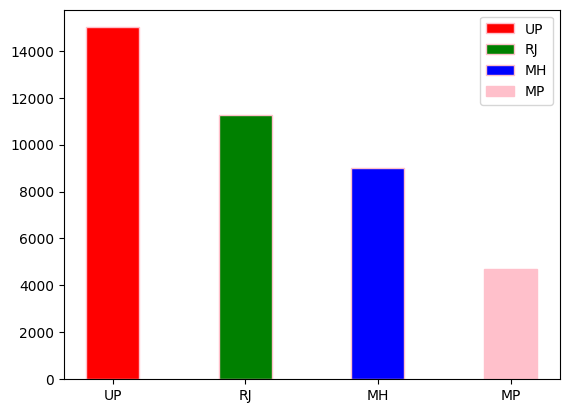
plt.scatter(df['tip'],df['total\_bill'])  
plt.xlabel('tip')  
plt.ylabel('total\_bill')  
plt.show()



plt.scatter(bike['age'],bike['price'],color='k',marker="P") #v , D , P ,  
plt.xlabel('age')  
plt.ylabel('price')  
plt.show()



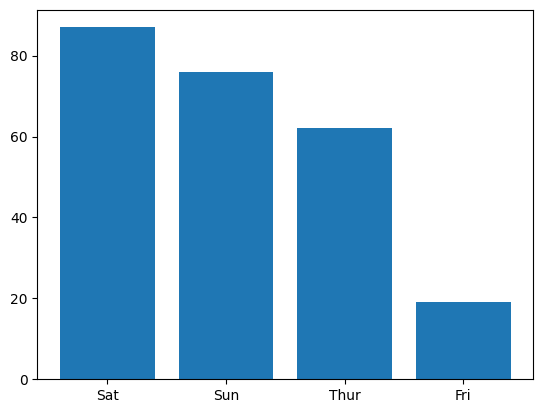
# x --> categorical  
# y --> numerical  
  
  
label = ['UP','RJ','MH','MP']  
population = [15000,11243,9008,4687]  
color\_lst = ['red','green','blue','pink']  
plt.bar(label,population,color=color\_lst,edgecolor='pink',width=0.4,label=label)  
plt.legend()  
plt.show()



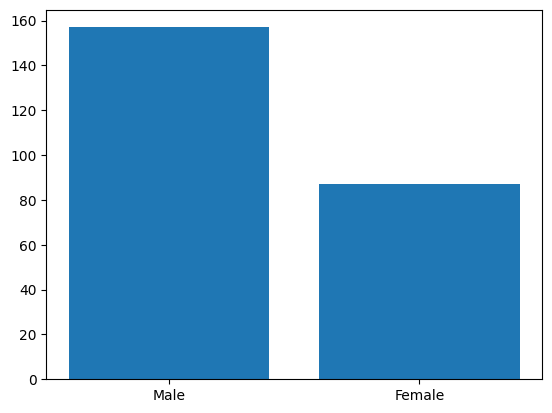
df

total\_bill tip sex smoker day time size  
0 16.99 1.01 Female No Sun Dinner 2  
1 10.34 1.66 Male No Sun Dinner 3  
2 21.01 3.50 Male No Sun Dinner 3  
3 23.68 3.31 Male No Sun Dinner 2  
4 24.59 3.61 Female No Sun Dinner 4  
.. ... ... ... ... ... ... ...  
239 29.03 5.92 Male No Sat Dinner 3  
240 27.18 2.00 Female Yes Sat Dinner 2  
241 22.67 2.00 Male Yes Sat Dinner 2  
242 17.82 1.75 Male No Sat Dinner 2  
243 18.78 3.00 Female No Thur Dinner 2  
  
[244 rows x 7 columns]

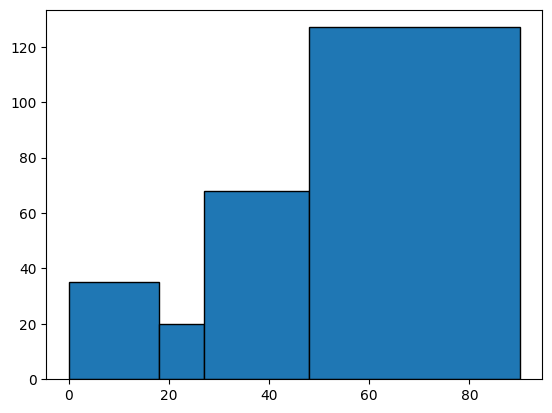
days = list(df['day'].value\_counts().index)  
counting = list(df['day'].value\_counts().values)  
  
plt.bar(days,counting)  
plt.show()



days = list(df['sex'].value\_counts().index)  
counting = list(df['sex'].value\_counts().values)  
  
plt.bar(days,counting)  
plt.show()



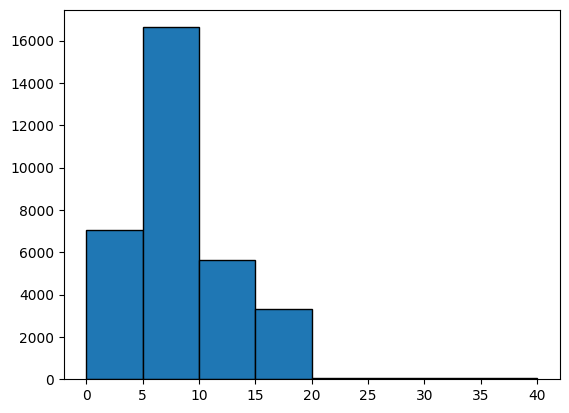
# histogram   
  
age = np.random.randint(5,90,250)  
bin = [0,18,27,48,90]  
  
plt.hist(age,bin,edgecolor='k')  
plt.show()



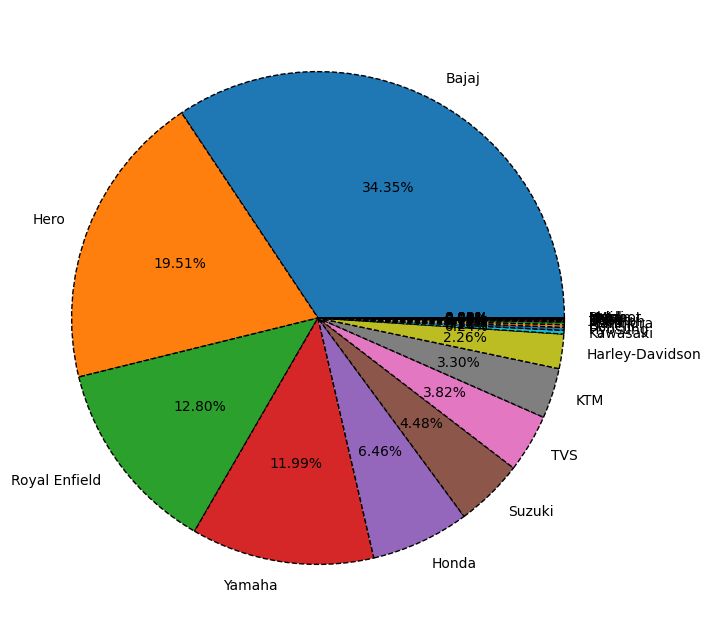
len(age[age<18])

35

bin = [0,5,10,15,20,40]  
plt.hist(bike['age'],bin,edgecolor='k')  
plt.show()



## pie plot  
ownership = list(bike['brand'].value\_counts().index)  
no\_of\_bike = list(bike['brand'].value\_counts().values)  
plt.figure(figsize=(8,8))  
plt.pie(no\_of\_bike,labels=ownership,autopct="%2.2f%%",wedgeprops={'edgecolor':'k','linestyle':'dashed'})  
plt.savefig('pieplot.png')  
plt.show()



https://www.kaggle.com/datasets/imdevskp/corona-virus-report

Cell In[26], line 1  
 https://www.kaggle.com/datasets/imdevskp/corona-virus-report  
 ^  
SyntaxError: invalid syntax