1. Write a Pandas program to select distinct department id from employees file.

import pandas as pd

employees=pd.read\_csv(r"EMPLOYEES.csv")

print("Distinct department\_id:")

print(employees.department\_id.unique())

2 Write a Pandas program to display the ID for those employees who did two or more jobs in the past

import pandas as pd

employees=pd.read\_csv(r"EMPLOYEES.csv")

result=job\_history.groupby(['employee\_id'])

print(result.filter(lambda x:len(x)>1).groupby('employee\_id').size().sort\_values(ascending=False))

**3** Write a Pandas program to display the details of jobs in descending sequence on job title.

import pandas as pd

employees=pd.read\_csv(r"EMPLOYEES.csv")

result=jobs.sort\_values('job\_title')

for index, row inresult.iterrows():

print(row['job\_id'].ljust(15),row['job\_title'].ljust(35),str(row['min\_salary']).ljust(9),row['max\_salary'])

1. Write a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates.

.import pandas aspd

importmatplotlib.pyplotasplt

df=pd.read\_csv("alphabet\_stock\_data.csv")

start\_date=pd.to\_datetime('2020-4-1')

end\_date=pd.to\_datetime('2020-09-30')

df['Date']=pd.to\_datetime(df['Date'])

new\_df=(df['Date']>=start\_date)&(df['Date']<=end\_date)

df1 =df.loc[new\_df]

df2 = df1.set\_index('Date')

plt.figure(figsize=(5,5))

plt.suptitle('Stock prices of Alphabet Inc.,\n01-04-2020 to 30-09-2020', \

fontsize=18,color='black')

plt.xlabel("Date",fontsize=16,color='black')

plt.ylabel("$ price",fontsize=16,color='black')

df2['Close'].plot(color='green');

plt.show()

5.Write a Pandas program to create a bar plot of the trading volume of Alphabet Inc. stock between two specific dates.

import pandas aspd

importmatplotlib.pyplotasplt

df=pd.read\_csv("alphabet\_stock\_data.csv")

start\_date=pd.to\_datetime('2020-4-1')

end\_date=pd.to\_datetime('2020-4-30')

df['Date']=pd.to\_datetime(df['Date'])

new\_df=(df['Date']>=start\_date)&(df['Date']<=end\_date)

df1 =df.loc[new\_df]

df2 = df1.set\_index('Date')

plt.figure(figsize=(6,6))

plt.suptitle('Trading Volume of Alphabet Inc. stock,\n01-04-2020 to 30-04-2020',fontsize=16,color='black')

plt.xlabel("Date",fontsize=12,color='black')

plt.ylabel("Trading Volume",fontsize=12,color='black')

df2['Volume'].plot(kind='bar');

plt.show()

6.Write a Pandas program to create a scatter plot of the trading volume/stock prices of Alphabet Inc. stock between two specific dates.

**alphabet\_stock\_data:**

import pandas aspd

importmatplotlib.pyplotasplt

df=pd.read\_csv("alphabet\_stock\_data.csv")

start\_date=pd.to\_datetime('2020-4-1')

end\_date=pd.to\_datetime('2020-9-30')

df['Date']=pd.to\_datetime(df['Date'])

new\_df=(df['Date']>=start\_date)&(df['Date']<=end\_date)

df1 =df.loc[new\_df]

df2 = df1.set\_index('Date')

x=['Close']; y =['Volume']

plt.figure(figsize=[15,10])

df2.plot.scatter(x, y, s=50);

plt.grid(True)

plt.title('Trading Volume/Price of Alphabet Inc. stock,\n01-04-2020 to 30-09-2020',fontsize=14,color='black')

plt.xlabel("Stock Price",fontsize=12,color='black')

plt.ylabel("Trading Volume",fontsize=12,color='black')

plt.show()

7.Write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.(refer sales\_data table)

import pandas aspd

importnumpyas np

df=pd.read\_excel('E:\SaleData.xlsx')

table=pd.pivot\_table(df, index='Item', values='Sale\_amt',aggfunc=[np.max,np.min])

print(table)

8.Write a Pandas program to create a Pivot table and find the item wise unit sold. .(refer sales\_data table)

import pandas aspd

importnumpyas np

df=pd.read\_excel('E:\SaleData.xlsx')

print(pd.pivot\_table(df,index=["Item"], values="Units",aggfunc=np.sum))

9.Write a Pandas program to create a Pivot table and find the total sale amount region wise, manager wise, sales man wise. .(refer sales\_data table)

. importnumpy as np

import pandas as pd

df = pd.read\_excel('E:\SaleData.xlsx')

print(pd.pivot\_table(df,index=["Region","Manager","SalesMan"], values="Sale\_amt", aggfunc=np.sum))

10.Create a dataframe of ten rows, four columns with random values. Write a Pandas program to highlight the negative numbers red and positive numbers black.

import pandas aspd

importnumpyas np

np.random.seed(24)

df=pd.DataFrame({'A':np.linspace(1,10,10)})

df=pd.concat([df,pd.DataFrame(np.random.randn(10,4), columns=list('BCDE'))],

axis=1)

print("Original array:")

print(df)

defcolor\_negative\_red(val):

color='red'ifval<0else'black'

return'color: %s'%color

print("\nNegative numbers red and positive numbers black:")

df.style.applymap(color\_negative\_red)

11.Create a dataframe of ten rows, four columns with random values. Convert some values to nan values. Write a Pandas program which will highlight the nan values.

import pandas aspd

importnumpyas np

np.random.seed(24)

df=pd.DataFrame({'A':np.linspace(1,10,10)})

df=pd.concat([df,pd.DataFrame(np.random.randn(10,4), columns=list('BCDE'))],

axis=1)

df.iloc[0,2]=np.nan

df.iloc[3,3]=np.nan

df.iloc[4,1]=np.nan

df.iloc[9,4]=np.nan

print("Original array:")

print(df)

defcolor\_negative\_red(val):

color='red'ifval<0else'black'

return'color: %s'%color

print("\nNegative numbers red and positive numbers black:")

df.style.highlight\_null(null\_color='red')

12.Create a dataframe of ten rows, four columns with random values. Write a Pandas program to set dataframe background Color black and font color yellow.

import pandas aspd

importnumpyas np

np.random.seed(24)

df=pd.DataFrame({'A':np.linspace(1,10,10)})

df=pd.concat([df,pd.DataFrame(np.random.randn(10,4), columns=list('BCDE'))],

axis=1)

df.iloc[0,2]=np.nan

df.iloc[3,3]=np.nan

df.iloc[4,1]=np.nan

df.iloc[9,4]=np.nan

print("Original array:")

print(df)

print("\nBackground:black - fontcolor:yelow")

df.style.set\_properties(\*\*{'background-color':'black',

'color':'yellow'})

13.Write a Pandas program to detect missing values of a given DataFrame. Display True or False

import pandas as pd

Import numpy as np

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

df=pd.DataFrame({

'ord\_no':[70001,np.nan,70002,70004,np.nan,70005,np.nan,70010,70003,70012,np.nan,70013],

'purch\_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,1983.43,2480.4,250.45,75.29,3045.6],

'ord\_date':['2012-10-05','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-10','2012-06-27','2012-08-17','2012-04-25'],

'customer\_id':[3002,3001,3001,3003,3002,3001,3001,3004,3003,3002,3001,3001],

'salesman\_id':[5002,5003,5001,np.nan,5002,5001,5001,np.nan,5003,5002,5003,np.nan]})

print("Original Orders DataFrame:")

print(df)

print("\nMissing values of the said dataframe:")

print(df.isna())

14. Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

.import pandas aspd

importnumpyas np

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

df=pd.DataFrame({

'ord\_no':[70001,np.nan,70002,70004,np.nan,70005,"--",70010,70003,70012,np.nan,70013],

'purch\_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,"?",12.43,2480.4,250.45,3045.6],

'ord\_date':['?','2012-09-10',np.nan,'2012-08-17','2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-10','2012-06-27','2012-08-17','2012-04-25'],

'customer\_id':[3002,3001,3001,3003,3002,3001,3001,3004,"--",3002,3001,3001],

'salesman\_id':[5002,5003,"?",5001,np.nan,5002,5001,"?",5003,5002,5003,"--"]})

print("Original Orders DataFrame:")

print(df)

print("\nReplace the missing values with NaN:")

result=df.replace({"?":np.nan,"--":np.nan})

print(result)

15.Write a Pandas program to keep the rows with at least 2 NaN values in a given DataFrame.

import pandas aspd

importnumpyas np

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

df=pd.DataFrame({

'ord\_no':[np.nan,np.nan,70002,np.nan,np.nan,70005,np.nan,70010,70003,70012,np.nan,np.nan],

'purch\_amt':[np.nan,270.65,65.26,np.nan,948.5,2400.6,5760,1983.43,2480.4,250.45,75.29,np.nan],

'ord\_date':[np.nan,'2012-09-10',np.nan,np.nan,'2012-09-10','2012-07-27','2012-09-10','2012-10-10','2012-10-10','2012-06-27','2012-08-17',np.nan],

'customer\_id':[np.nan,3001,3001,np.nan,3002,3001,3001,3004,3003,3002,3001,np.nan]})

print("Original Orders DataFrame:")

print(df)

print("\nKeep the rows with at least 2 NaN values of the said DataFrame:")

result=df.dropna(thresh=2)

print(result)

16.Write a Pandas program to split the following dataframe into groups based on school code. Also check the type of GroupBy object.

.import pandas aspd

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

student\_data=pd.DataFrame({

'school\_code':['s001','s002','s003','s001','s002','s004'],

'class':['V','V','VI','VI','V','VI'],

'name':['Alberto Franco','GinoMcneill','Ryan Parkes','Eesha Hinton','Gino Mcneill','David Parkes'],

'date\_Of\_Birth ':['15/05/2002','17/05/2002','16/02/1999','25/09/1998','11/05/2002','15/09/1997'],

'age':[12,12,13,13,14,12],

'height':[173,192,186,167,151,159],

'weight':[35,32,33,30,31,32],

'address':['street1','street2','street3','street1','street2','street4']},

index=['S1','S2','S3','S4','S5','S6'])

print("Original DataFrame:")

print(student\_data)

print('\nSplit the said data on school\_code wise:')

result=student\_data.groupby(['school\_code'])

forname,groupin result:

print("\nGroup:")

print(name)

print(group)

print("\nType of the object:")

print(type(result))

17.Write a Pandas program to split the following dataframe by school code and get mean, min, and max value of age for each school.

import pandas aspd

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

student\_data=pd.DataFrame({

'school\_code':['s001','s002','s003','s001','s002','s004'],

'class':['V','V','VI','VI','V','VI'],

'name':['Alberto Franco','GinoMcneill','Ryan Parkes','Eesha Hinton','Gino Mcneill','David Parkes'],

'date\_Of\_Birth ':['15/05/2002','17/05/2002','16/02/1999','25/09/1998','11/05/2002','15/09/1997'],

'age':[12,12,13,13,14,12],

' height ':[173,192,186,167,151,159],

'weight':[35,32,33,30,31,32],

'address':['street1','street2','street3','street1','street2','street4']},

index=['S1','S2','S3','S4','S5','S6'])

print("Original DataFrame:")

print(student\_data)

print('\nMean, min, and max value of age for each school with customized column names:')

grouped\_single=student\_data.groupby('school\_code').agg(Age\_Mean=('age','mean'),Age\_Max=('age',max),Age\_Min=('age',min))

print(grouped\_single)

18.Write a Pandas program to split the following given dataframe into groups based on school code and class.

import pandas aspd

pd.set\_option('display.max\_rows',None)

#pd.set\_option('display.max\_columns', None)

student\_data=pd.DataFrame({

'school\_code':['s001','s002','s003','s001','s002','s004'],

'class':['V','V','VI','VI','V','VI'],

'name':['Alberto Franco','GinoMcneill','Ryan Parkes','Eesha Hinton','Gino Mcneill','David Parkes'],

'date\_Of\_Birth ':['15/05/2002','17/05/2002','16/02/1999','25/09/1998','11/05/2002','15/09/1997'],

'age':[12,12,13,13,14,12],

'height':[173,192,186,167,151,159],

'weight':[35,32,33,30,31,32],

'address':['street1','street2','street3','street1','street2','street4']},

index=['S1','S2','S3','S4','S5','S6'])

print("Original DataFrame:")

print(student\_data)

print('\nSplit the said data on school\_code wise:')

result=student\_data.groupby(['school\_code'])

forname,groupin result:

print("\nGroup:")

print(name)

print(group)

print("\nType of the object:")

print(type(result))

19.Write a Pandas program to display the dimensions or shape of the World alcohol consumption dataset. Also extract the column names from the dataset.

import pandas aspd

# World alcohol consumption data

w\_a\_con=pd.read\_csv('world\_alcohol.csv')

print("World alcohol consumption sample data:")

print(w\_a\_con.head())

print('\nShape of the dataframe: ',w\_a\_con.shape)

print('\nNumber of rows: ',w\_a\_con.shape[0])

print('\nNumber of column: ',w\_a\_con.shape[1])

print("\nExtract Column Names:")

print(w\_a\_con.columns)

20 Write a Pandas program to find the index of a given substring of a DataFrame column

import pandas aspd

df=pd.DataFrame({

'name\_code':['c0001','1000c','b00c2','b2c02','c2222'],

'date\_of\_birth ':['12/05/2002','16/02/1999','25/09/1998','12/02/2022','15/09/1997'],

'age':[18.5,21.2,22.5,22,23]

})

print("Original DataFrame:")

print(df)

print("\nIndex of a substring in a specified column of a dataframe:")

df['Index']=list(map(lambda x:x.find('c',0,5),df['name\_code']))

print(df)

21.Write a Pandas program to swap the cases of a specified character column in a given DataFrame.

import pandas aspd

df=pd.DataFrame({

'company\_code':['Abcd','EFGF','zefsalf','sdfslew','zekfsdf'],

'date\_of\_sale':['12/05/2002','16/02/1999','25/09/1998','12/02/2022','15/09/1997'],

'sale\_amount':[12348.5,233331.2,22.5,2566552.0,23.0]

})

print("Original DataFrame:")

print(df)

print("\nSwapp cases in comapny\_code:")

df['swapped\_company\_code']=list(map(lambda x:x.swapcase(),df['company\_code']))

print(df)

22.Write a Python program to draw a line with suitable label in the x axis, y axis and a title.

import matplotlib.pyplot as plt

X =range(1,50)

Y =[value \*3for value in X]

print("Values of X:")

print(\*range(1,50))

print("Values of Y (thrice of X):")

print(Y)

# Plot lines and/or markers to the Axes.

plt.plot(X, Y)

# Set the x axis label of the current axis.

plt.xlabel('x - axis')

# Set the y axis label of the current axis.

plt.ylabel('y - axis')

# Set a title

plt.title('Draw a line.')

# Display the figure.

plt.show()

23.Write a Python program to draw a line using given axis values taken from a text file, with suitable label in the x axis, y axis and a title.  
  
importmatplotlib.pyplotasplt

# x axis values

x =[1,2,3]

# y axis values

y =[2,4,1]

# Plot lines and/or markers to the Axes.

plt.plot(x, y)

# Set the x axis label of the current axis.

plt.xlabel('x - axis')

# Set the y axis label of the current axis.

plt.ylabel('y - axis')

# Set a title

plt.title('Sample graph!')

# Display a figure.

plt.show()

24.Write a Python program to draw line charts of the financial data of Alphabet Inc. between October 3, 2016 to October 7, 2016.  
Sample Financial data (fdata.csv):  
Date,Open,High,Low,Close  
10-03-16,774.25,776.065002,769.5,772.559998  
10-04-16,776.030029,778.710022,772.890015,776.429993  
10-05-16,779.309998,782.070007,775.650024,776.469971  
10-06-16,779,780.47998,775.539978,776.859985  
10-07-16,779.659973,779.659973,770.75,775.080017

importmatplotlib.pyplotasplt

import pandas aspd

df=pd.read\_csv('fdata.csv',sep=',',parse\_dates=True,index\_col=0)

df.plot()

plt.show()

25.Write a Python program to plot two or more lines with legends, different widths and colors.

importmatplotlib.pyplotasplt

# line 1 points

x1=[10,20,30]

y1 =[20,40,10]

# line 2 points

x2=[10,20,30]

y2 =[40,10,30]

# Set the x axis label of the current axis.

plt.xlabel('x - axis')

# Set the y axis label of the current axis.

plt.ylabel('y - axis')

# Set a title

plt.title('Two or more lines with different widths and colors with suitable legends ')

# Display the figure.

plt.plot(x1,y1,color='blue', linewidth =3, label ='line1-width-3')

plt.plot(x2,y2,color='red', linewidth =5, label ='line2-width-5')

# show a legend on the plot

plt.legend()

plt.show()

26.Write a Python program to create multiple plots

importmatplotlib.pyplotasplt

fig=plt.figure()

fig.subplots\_adjust(bottom=0.020, left=0.020, top =0.900, right=0.800)

plt.subplot(2,1,1)

plt.xticks(()),plt.yticks(())

plt.subplot(2,3,4)

plt.xticks(())

plt.yticks(())

plt.subplot(2,3,5)

plt.xticks(())

plt.yticks(())

plt.subplot(2,3,6)

plt.xticks(())

plt.yticks(())

plt.show()

27.Write a Python programming to display a bar chart of the popularity of programming Languages.

importmatplotlib.pyplotasplt

x =['Java','Python','PHP','JavaScript','C#','C++']

popularity=[22.2,17.6,8.8,8,7.7,6.7]

x\_pos=[ifori, \_ inenumerate(x)]

plt.bar(x\_pos, popularity,color='blue')

plt.xlabel("Languages")

plt.ylabel("Popularity")

plt.title("PopularitY of Programming Language\n"+"Worldwide, Oct 2017 compared to a year ago")

plt.xticks(x\_pos, x)

# Turn on the grid

plt.minorticks\_on()

plt.grid(which='major',linestyle='-', linewidth='0.5',color='red')

# Customize the minor grid

plt.grid(which='minor',linestyle=':', linewidth='0.5',color='black')

plt.show()

28.Write a Python programming to display a horizontal bar chart of the popularity of programming Languages.  
  
importmatplotlib.pyplotasplt

x =['Java','Python','PHP','JS','C#','C++']

popularity=[22.2,17.6,8.8,8,7.7,6.7]

x\_pos=[ifori, \_ inenumerate(x)]

plt.barh(x\_pos, popularity,color='green')

plt.xlabel("Popularity")

plt.ylabel("Languages")

plt.title("PopularitY of Programming Language\n"+"Worldwide, Oct 2017 compared to a year ago")

plt.yticks(x\_pos, x)

# Turn on the grid

plt.minorticks\_on()

plt.grid(which='major',linestyle='-', linewidth='0.5',color='red')

# Customize the minor grid

plt.grid(which='minor',linestyle=':', linewidth='0.5',color='black')

plt.show()

29.Write a Python programming to display a bar chart of the popularity of programming Languages. Use different color for each bar.

importmatplotlib.pyplotasplt

x =['Java','Python','PHP','JavaScript','C#','C++']

popularity=[22.2,17.6,8.8,8,7.7,6.7]

x\_pos=[ifori, \_ inenumerate(x)]

plt.bar(x\_pos, popularity,color=['red','black','green','blue','yellow','cyan'])

plt.xlabel("Languages")

plt.ylabel("Popularity")

plt.title("PopularitY of Programming Language\n"+"Worldwide, Oct 2017 compared to a year ago")

plt.xticks(x\_pos, x)

# Turn on the grid

plt.minorticks\_on()

plt.grid(which='major',linestyle='-', linewidth='0.5',color='red')

# Customize the minor grid

plt.grid(which='minor',linestyle=':', linewidth='0.5',color='black')

plt.show()

30.Write a Python program to create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.  
  
importnumpyas np

importmatplotlib.pyplotasplt

# data to plot

n\_groups=5

men\_means=(22,30,33,30,26)

women\_means=(25,32,30,35,29)

# create plot

fig,ax=plt.subplots()

index=np.arange(n\_groups)

bar\_width=0.35

opacity=0.8

rects1 =plt.bar(index,men\_means,bar\_width,

alpha=opacity,

color='g',

label='Men')

rects2 =plt.bar(index +bar\_width,women\_means,bar\_width,

alpha=opacity,

color='r',

label='Women')

plt.xlabel('Person')

plt.ylabel('Scores')

plt.title('Scores by person')

plt.xticks(index +bar\_width,('G1','G2','G3','G4','G5'))

plt.legend()

plt.tight\_layout()

plt.show()

31.Write a Python program to create a stacked bar plot with error bars.  
Note: Use bottom to stack the women?s bars on top of the men?s bars.  
.importnumpyas np

importmatplotlib.pyplotasplt

N =5

menMeans=(22,30,35,35,26)

womenMeans=(25,32,30,35,29)

menStd=(4,3,4,1,5)

womenStd=(3,5,2,3,3)

# the x locations for the groups

ind=np.arange(N)

# the width of the bars

width=0.35

p1 =plt.bar(ind,menMeans, width,yerr=menStd,color='red')

p2 =plt.bar(ind,womenMeans, width,

bottom=menMeans,yerr=womenStd,color='green')

plt.ylabel('Scores')

plt.xlabel('Groups')

plt.title('Scores by group\n'+'and gender')

plt.xticks(ind,('Group1','Group2','Group3','Group4','Group5'))

plt.yticks(np.arange(0,81,10))

plt.legend((p1[0], p2[0]),('Men','Women'))

plt.show()

32.Write a Python program to draw a scatter graph taking a random distribution in X and Y and plotted against each other.

importnumpyas np

importmatplotlib.pyplotasplt

N =5

menMeans=(22,30,35,35,26)

womenMeans=(25,32,30,35,29)

menStd=(4,3,4,1,5)

womenStd=(3,5,2,3,3)

# the x locations for the groups

ind=np.arange(N)

# the width of the bars

width=0.35

p1 =plt.bar(ind,menMeans, width,yerr=menStd,color='red')

p2 =plt.bar(ind,womenMeans, width,

bottom=menMeans,yerr=womenStd,color='green')

plt.ylabel('Scores')

plt.xlabel('Groups')

plt.title('Scores by group\n'+'and gender')

plt.xticks(ind,('Group1','Group2','Group3','Group4','Group5'))

plt.yticks(np.arange(0,81,10))

plt.legend((p1[0], p2[0]),('Men','Women'))

plt.show()

33.Write a Python program to draw a scatter plot with empty circles taking a random distribution in X and Y and plotted against each other

importmatplotlib.pyplotasplt

importnumpyas np

x =np.random.randn(50)

y =np.random.randn(50)

plt.scatter(x, y, s=70,facecolors='none',edgecolors='g')

plt.xlabel("X")

plt.ylabel("Y")

plt.show()

34.Write a Python program to draw a scatter plot using random distributions to generate balls of different sizes.

import math

import random

importmatplotlib.pyplotasplt

# create random data

no\_of\_balls=25

x =[random.triangular()foriinrange(no\_of\_balls)]

y =[random.gauss(0.5,0.25)foriinrange(no\_of\_balls)]

colors=[random.randint(1,4)foriinrange(no\_of\_balls)]

areas=[math.pi\*random.randint(5,15)\*\*2foriinrange(no\_of\_balls)]

# draw the plot

plt.figure()

plt.scatter(x, y, s=areas, c=colors, alpha=0.85)

plt.axis([0.0,1.0,0.0,1.0])

plt.xlabel("X")

plt.ylabel("Y")

plt.show()

35.Write a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students.

importmatplotlib.pyplotasplt

import pandas aspd

math\_marks=[88,92,80,89,100,80,60,100,80,34]

science\_marks=[35,79,79,48,100,88,32,45,20,30]

marks\_range=[10,20,30,40,50,60,70,80,90,100]

plt.scatter(marks\_range,math\_marks, label='Math marks',color='r')

plt.scatter(marks\_range,science\_marks, label='Science marks',color='g')

plt.title('Scatter Plot')

plt.xlabel('Marks Range')

plt.ylabel('Marks Scored')

plt.legend()

plt.show()

36.Write a Python program to draw a scatter plot for three different groups comparing weights and heights

importmatplotlib.pyplotasplt

importnumpyas np

weight1=[67,57.2,59.6,59.64,55.8,61.2,60.45,61,56.23,56]

height1=[101.7,197.6,98.3,125.1,113.7,157.7,136,148.9,125.3,114.9]

weight2=[61.9,64,62.1,64.2,62.3,65.4,62.4,61.4,62.5,63.6]

height2=[152.8,155.3,135.1,125.2,151.3,135,182.2,195.9,165.1,125.1]

weight3=[68.2,67.2,68.4,68.7,71,71.3,70.8,70,71.1,71.7]

height3=[165.8,170.9,192.8,135.4,161.4,136.1,167.1,235.1,181.1,177.3]

weight=np.concatenate((weight1,weight2,weight3))

height=np.concatenate((height1,height2,height3))

plt.scatter(weight, height, marker='\*',color=['blue'])

plt.xlabel('weight',fontsize=16)

plt.ylabel('height',fontsize=16)

plt.title('Group wise Weight vs Height scatter plot',fontsize=20)

plt.show()

37.import pandas aspd

df=pd.DataFrame({'X':[78,85,96,80,86],'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]});

print(df)

37.Write a Pandas program to create a dataframe from a dictionary and display it.

38.Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.  
Sample Python dictionary data and list labels:

import pandas aspd

importnumpyas np

exam\_data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],

'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],

'attempts':[1,3,2,3,2,3,1,1,2,1],

'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}

labels =['a','b','c','d','e','f','g','h','i','j']

df=pd.DataFrame(exam\_data, index=labels)

print(df)

39.Write a Pandas program to get the first 3 rows of a given DataFrame.  
Sample Python dictionary data and list labels:  
exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

import pandas aspd

importnumpyas np

exam\_data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],

'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],

'attempts':[1,3,2,3,2,3,1,1,2,1],

'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}

labels =['a','b','c','d','e','f','g','h','i','j']

df=pd.DataFrame(exam\_data, index=labels)

print("First three rows of the data frame:")

print(df.iloc[:3])

1. Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.  
   Sample Python dictionary data and list labels:  
   exam\_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],  
   'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
   'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
   'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
   labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

import pandas aspd

importnumpyas np

exam\_data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],

'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],

'attempts':[1,3,2,3,2,3,1,1,2,1],

'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}

labels =['a','b','c','d','e','f','g','h','i','j']

df=pd.DataFrame(exam\_data, index=labels)

print("Select specific columns:")

print(df[['name','score']])