**Data Visualization**

**Practical Journal**

**Submitted By**

**Deep Bhanushali**

**Roll No: 22-11705**

**TY BSc CS (Hons)**

**Department Of Computer Science**

**Somaiya Vidyavihar University**

**SK Somaiya college**

|  |  |  |
| --- | --- | --- |
| Practical No. | Title | Sign |
| 1 | Perform operations on series  a. Creation of series  b. Manipulation of series |  |
| 2 | Perform Operations on DataFrame  a. Creation of DataFrame  b. Manipulation of DataFrame |  |
| 3 | Creating a dataset in excel and importing it in python and performing statistical analysis on it |  |
| 4 | Create a file data.txt and perform read and write operation on the file |  |
| 5 | Applying arithmetic and conditional operation on series dataset |  |
| 6 | Creating the excel sheet and perform groupby on the data set |  |
| 7 | Creating an excel file and performing visualization by using line chart, bar chart to analyse it |  |
| 8 | Creating following visualization on dataset   1. Pie chart 2. Box plot 3. Histogram 4. Scatter plot |  |
| 9 | Plotting the graph and performing visualization and analysis using seaborn library |  |
| 10 | Plotting the graph and performing visualization and analysis using seaborn library. |  |
| 11 | Using stack plot for given dataset to perform analysis |  |
| 12 | Creating frequency table for series and dataframe dataset  Frequency Table |  |

**Practical 1:- Performing operations on series:**

* **Operation of series**
* **Manipulation on series**

1. Operation of series:-

* Creation of series
* Using Scalar values

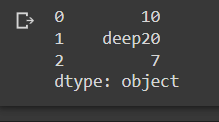
Code:

import pandas as pd

series1=pd.Series([10,'deep20',7])

print(series1)

Output:



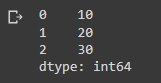
Code:

import pandas as pd

series1 = pd.Series([10,20,30])

print(series1)

Output:



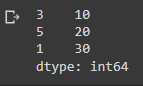
Code:

import pandas as pd

series1 = pd.Series([10,20,30],index =[3,5,1])

print(series1)

Output:



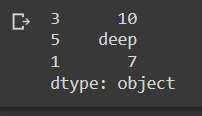
Code:

import pandas as pd

series2=pd.Series([10,'deep',7],index=[3,5,1])

print(series2)

Output:



* Using numpy

Code:

import numpy as np

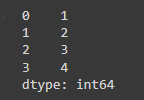
import pandas as pd

array1 = np.array([1,2,3,4])

series3 = pd.Series(array1)

print(series3)

Output:



Code:

import numpy as np

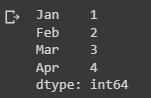
import pandas as pd

array1=np.array([1,2,3,4])

series5=pd.Series(array1, index=['Jan','Feb','Mar','Apr'])

print(series5)

Output:



* Using Dictionary

Code:

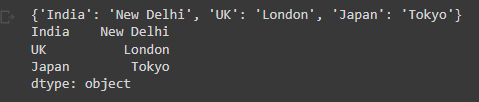
dict1 = {'India' : 'New Delhi', 'UK' :'London', 'Japan' :'Tokyo'}

print(dict1)

series3 = pd.Series(dict1)

print(series3)

Output:



**B. Manipulation of Series**

* **By Indexing**

Code:

import pandas as pd

seriesCapCntry = pd.Series(['New Delhi','Washington', 'London', 'Paris'],index=['India', 'USA', 'UK', 'France'])

seriesCapCntry['UK']

seriesCapCntry[2]

Output:



Code:

import pandas as pd

seriesNum=pd.Series([10,20,30])

seriesNum[2]

Output:



Code:

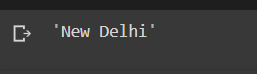
import pandas as pd

seriesCapCntry = pd.Series(['New Delhi','WashingtonDC','London','Paris'],index=['India','USA','UK','France'])

seriesCapCntry['India']

seriesCapCntry[0]

Output:



* By Slicing

Code:

import pandas as pd

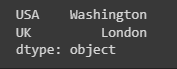
seriesCapCntry = pd.Series(['New Delhi','Washington', 'London', 'Paris'],index=['India', 'USA', 'UK', 'France'])

seriesCapCntry[1:3]

#seriesCapCntry['USA' : 'France]

#seriesCapCnrty[::-1]

Output:



Code:

import pandas as pd

seriesCapCntry = pd.Series(['New Delhi','Washington', 'London', 'Paris'],index=['India', 'USA', 'UK', 'France'])

seriesCapCntry[2:3]

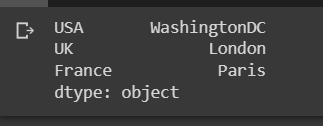
Output:



Code:

seriesCapCntry['USA':'France']

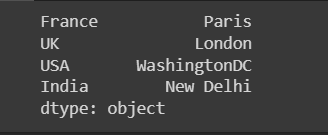
Output:



Code:

seriesCapCntry[::-1]

Output:



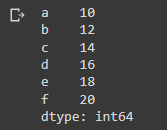
Code:

import numpy as np

seriesAlph = pd.Series(np.arange(10,21,2), index = ['a','b','c','d','e','f'])

seriesAlph

Output:



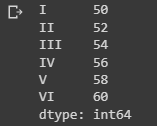
Code:

import numpy as np

seriesAlph = pd.Series(np.arange(50,61,2), index = ['I','II', 'III', 'IV', 'V', 'VI'])

seriesAlph

Output:



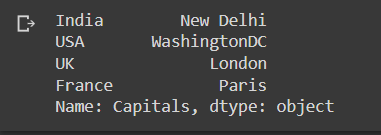
Code:

seriesCapCntry = pd.Series(['New Delhi','WashingtonDC','London','Paris'],index=['India','USA','UK','France'])

seriesCapCntry.name='Capitals'

print(seriesCapCntry)

Output:

****

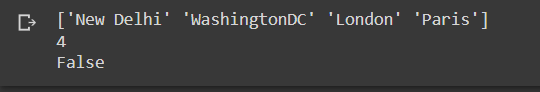
Code:

print(seriesCapCntry.values)

print(seriesCapCntry.size)

seriesCapCntry.empty

Output:



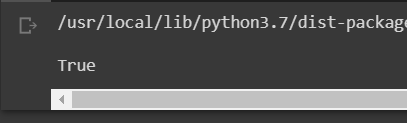
Code:Creating Empty Series

#create an empty series

seriesEmpt=pd.Series()

seriesEmpt.empty

Output:



Code:

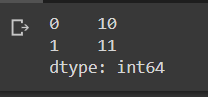
import pandas as pd

import numpy as np

seriesTenTwenty=pd.Series(np.arange(10,20,1),index=[0,1,2,3,4,5,6,7,8,9])

seriesTenTwenty.head(2)

Output:



Code:

seriesTenTwenty.count()

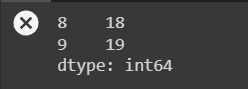
Output:



Code:

seriesTenTwenty.tail(2)

Output:



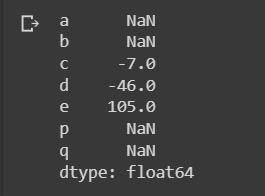
Code:Addition

seriesA=pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])

seriesB=pd.Series([10,20,-10,-50,100],index=['p','q','c','d','e'])

seriesA+seriesB

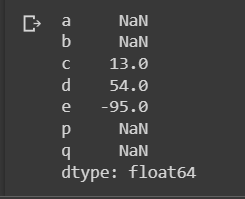
Output:



Code:Subtraction

seriesA-seriesB

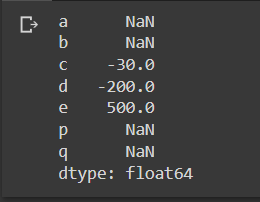
Output:



Code:Multiplication

seriesA\*seriesB

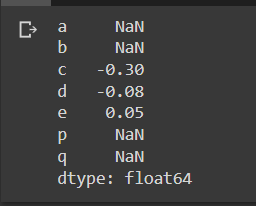
Output:



Code:Division

seriesA/seriesB

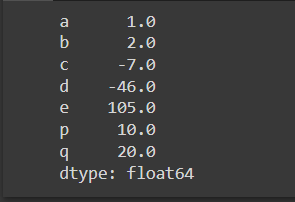
Output:



Code:addition using fill value

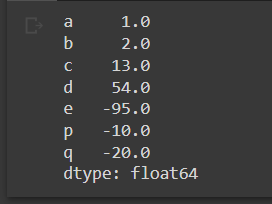
seriesA.add(seriesB,fill\_value=0)

Output:



Code:Subtraction using fill value

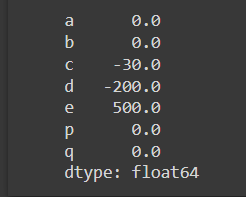
seriesA.sub(seriesB,fill\_value=0)

Output:

Code:Multiplication Using fill value

seriesA.mul(seriesB,fill\_value=0)

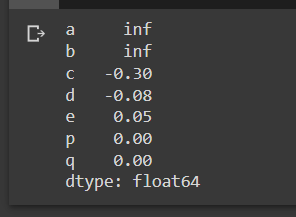
Output:



Code: Division using fill value

seriesA.div(seriesB,fill\_value=0)

Output:



**Practical 2:**

**Performing operations on dataframe:**

* **Operation of dataframe**
* **Manipulation on dataframe**
* **Creation of Dataframe**

**1.** **Create an empty data frame**

dFrameEmt=pd.DataFrame()

dFrameEmt



2. **creation of dataframe from numpy ndarrays**

import numpy as np

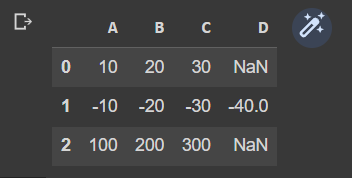
array1=np.array([10,20,30])

array2=np.array([100,200,300])

array3=np.array([-10,-20,-30,-40])

dFrame4=pd.DataFrame([array1,array3,array2],columns=['A','B','C','D'])

dFrame4

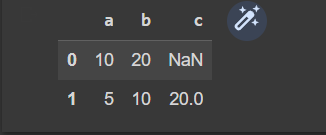


3. **creation of dataframe from list of dictionaries**

listDict=[{'a':10,'b':20},{'a':5,'b':10,'c':20}]

dFrameListDict=pd.DataFrame(listDict)

dFrameListDict

****

**4.** **From Series4**

seriesA=pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])

seriesB=pd.Series([1000,2000,-1000,-5000,1000],index=['a','b','c','d','e'])

seriesC=pd.Series([10,20,-10,-50,100],index=['z','y','a','c','e'])

#dFrame6=pd.DataFrame(seriesA)

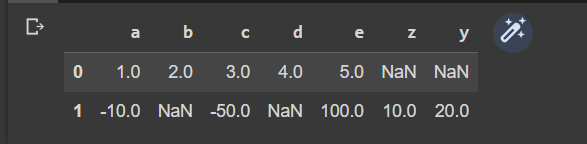
#print(dframe6)

#dFrame7=pd.DataFrame([seriesA,seriesB])

dFrame8=pd.DataFrame([seriesA,seriesC])

#print(dFrame7)

dFrame8

****

**5.** Creation of dataframe from dictionary of series

ResultSheet={

    'Deep':pd.Series([90,91,97],index=['Maths','Science','Hindi']),

    'Darren':pd.Series([90,91,97],index=['Maths','Science','Hindi']),

    'Sharvari':pd.Series([90,91,97],index=['Maths','Science','Hindi']),

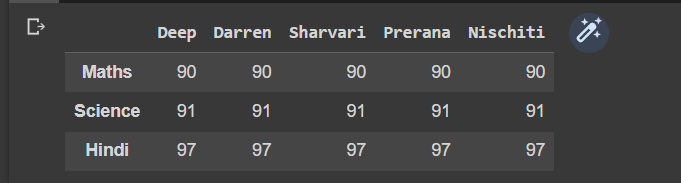
    'Prerana':pd.Series([90,91,97],index=['Maths','Science','Hindi']),

    'Nischiti':pd.Series([90,91,97],index=['Maths','Science','Hindi'])

}

ResultDF = pd.DataFrame(ResultSheet)

ResultDF

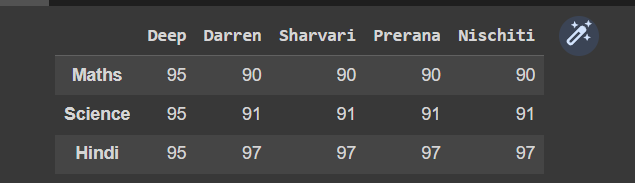
****

* **Manipulation of DataFrame**

**1.** modifying values of column

ResultDF['Deep']=95

ResultDF

****

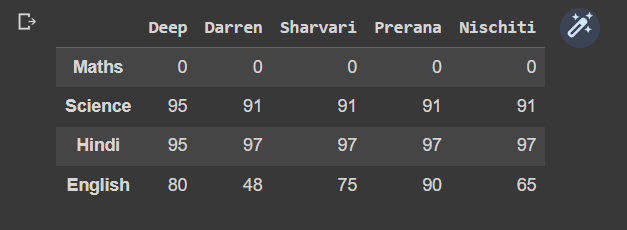
**2.** Adding a new row to DataFrame

ResultDF.loc['English']=[80,48,75,90,65]

ResultDF

#ResultDF.loc['Maths']=0

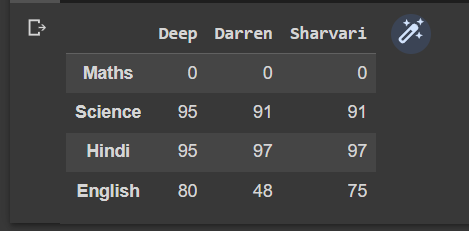
#ResultDF

****

**3.** delete the column

ResultDF = ResultDF.drop(['Prerana','Nischiti'],axis=1)

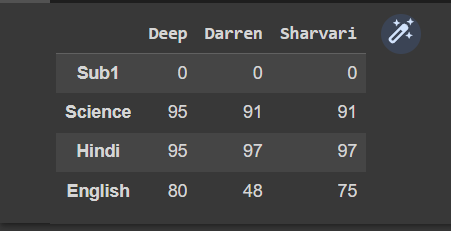
ResultDF

****

**4.** Remaining Row labels of a dataframe

ResultDF=ResultDF.rename({'Maths':'Sub1'},axis='index')

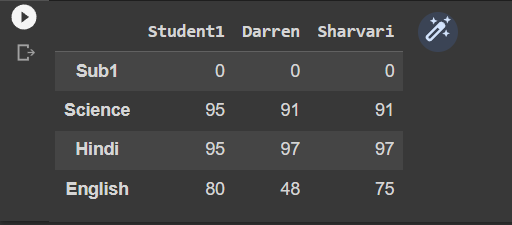
ResultDF

****

**5.** renaming columns labels of Dataframe

ResultDF=ResultDF.rename({'Deep':'Student1'},axis=1)

ResultDF

****

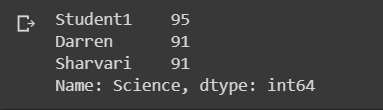
**6.** Accessing Dataframes element through indexing

ResultDF.loc['Science']

dFrame10Multiples = pd.DataFrame([10,20,30,40,50])

dFrame10Multiples.loc[2]

ResultDF.loc[:,'Student1']

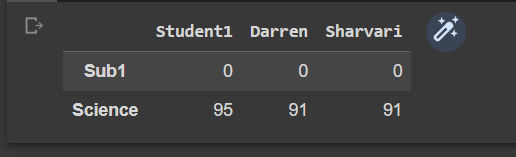


**7.** Accessing Data frame through slicing

Code:

ResultDF.loc['Sub1':'Science']

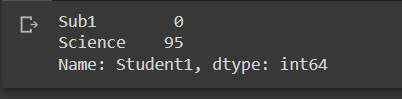
Output:

****

**Code:**

ResultDF.loc['Sub1':'Science','Student1']

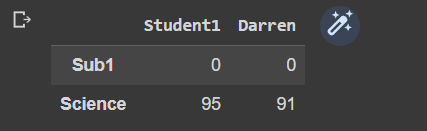
**Output:**

****

**Code:**

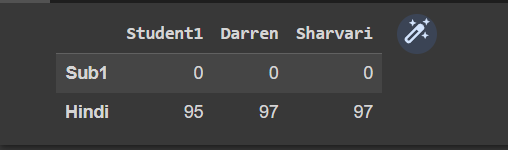
ResultDF.loc['Sub1':'Science','Student1':'Darren']

**Output:**

****

**8.** Filteering rows in Dataframes

ResultDF.loc[[True,False,True,False]]

****

**15.** Joining , merging and Concatenation of Dataframes

Code:

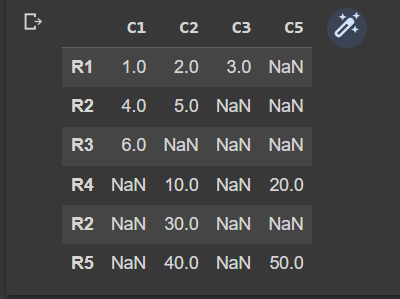
dFrame1=pd.DataFrame([[1,2,3],[4,5],[6]],columns=['C1','C2','C3'],index=['R1','R2','R3'])

dFrame2=pd.DataFrame([[10,20],[30],[40,50]],columns=['C2','C5'],index=['R4','R2','R5'])

dFrame1=dFrame1.append(dFrame2)

dFrame1

**Output:**

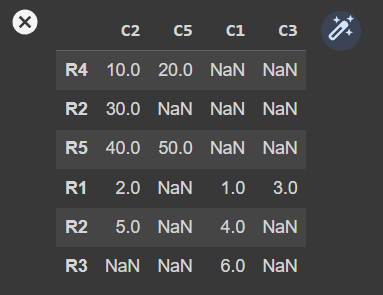
****

**Code:**

dFrame2 = dFrame2.append(dFrame1)

dFrame2

**Output:**

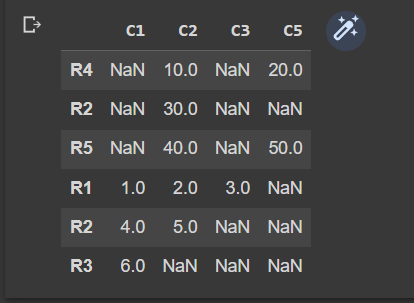
****

**Code:**

dFrame2 = dFrame2.append(dFrame1,sort='True')

dFrame2

**Output:**

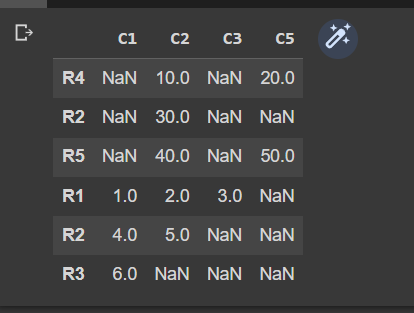
****

**Code:**

dFrame2 = dFrame2.append(dFrame1,sort='False')

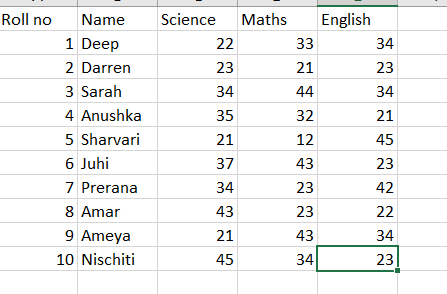
dFrame2

**Output:**

****

**Practical 3:-**

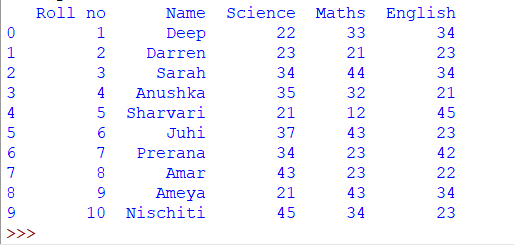
**Creating a database in excel and importing it in python and performing statistical analysis on it.**

****

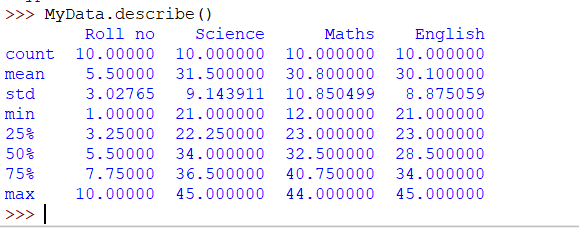
* import pandas as pd

>>> MyData=pd.read\_csv("C:\\TYCS\DV\Book2.csv")

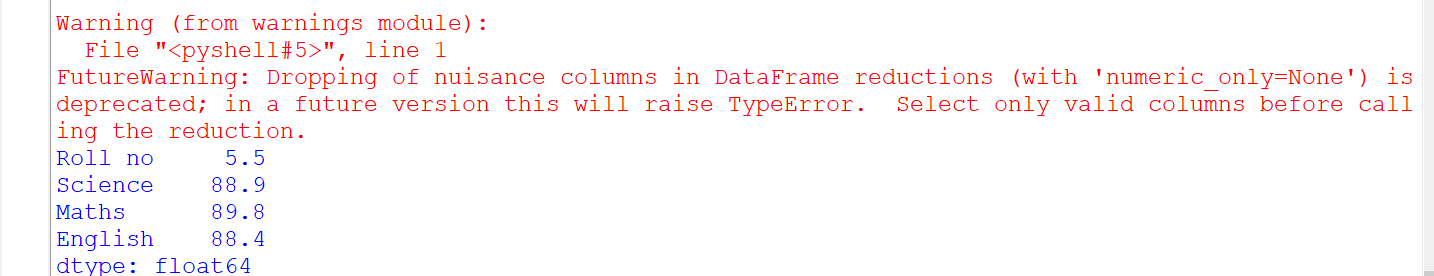
>>> MyData



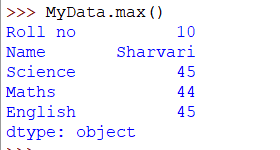
* MyData.describe()



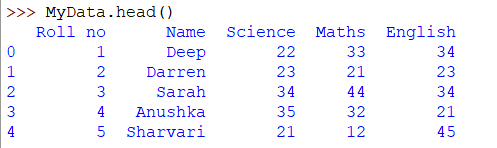
* MyData.mean()



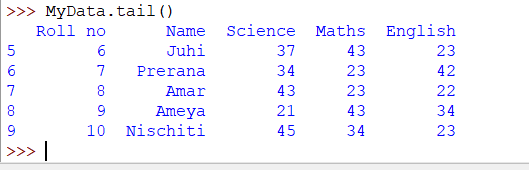
* MyData.max()



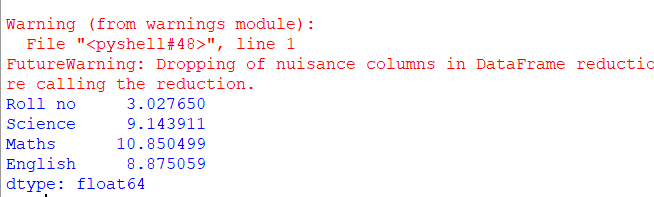
* MyData.head()



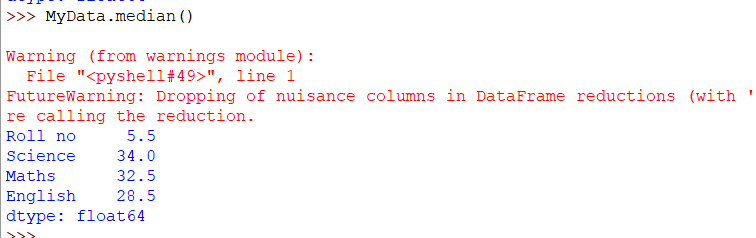
* MyData.tail()



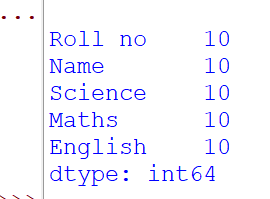
* MyData.std()



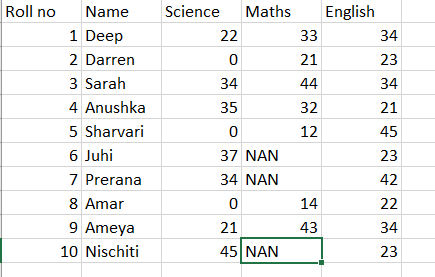
* MyData.median()



* MyData.count()



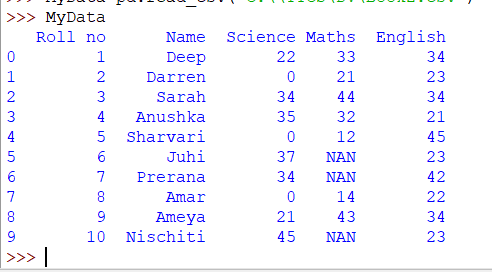
**If you don't enter the value and keep it null you will get the output as:**



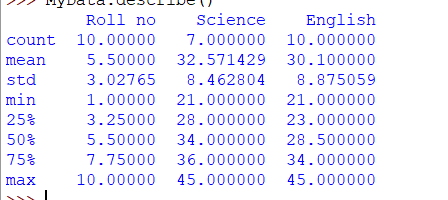
* import pandas as pd

>>> MyData=pd.read\_csv("C:\\TYCS\DV\Book2.csv")

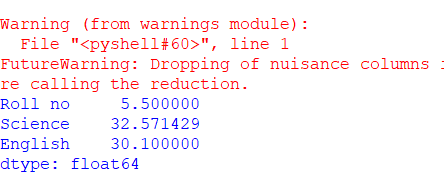
>>> MyData



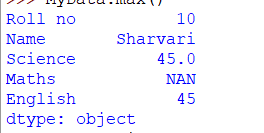
* MyData.describe()



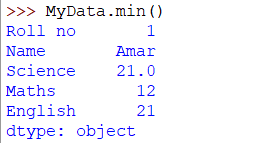
* MyData.mean()



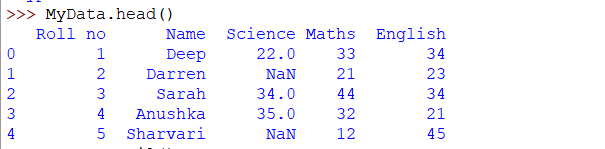
* MyData.max()



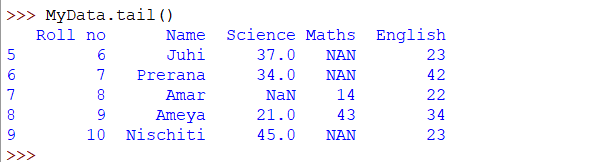
* MyData.min()



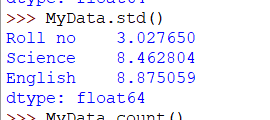
* MyData.head()



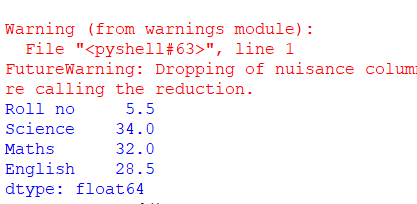
* MyData.tail()



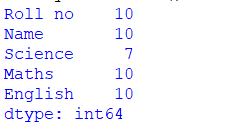
* MyData.std()



* MyData.median()



* MyData.count()



* Code:

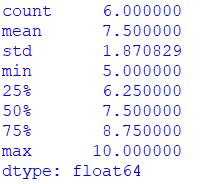
import pandas as pd

>>> import numpy as np

>>> S1=pd.Series([5,6,7,8,9,10])

>>> S1.describe()

Output:



**Practical 4:**

**Create a data.txt file and perform write and read operation on file**

**Code:**

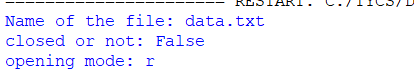
Filehndl=open("data.txt","r")

print("Name of the file:",Filehndl.name)

print("closed or not:",Filehndl.closed)

print("opening mode:",Filehndl.mode)

**Output:**

****

**Code:**

Filehndl=open("data.txt","r")

print("closed or not:",Filehndl.closed)

Filehndl.close()

print("closed or not:",Filehndl.closed)

**Output:**

****

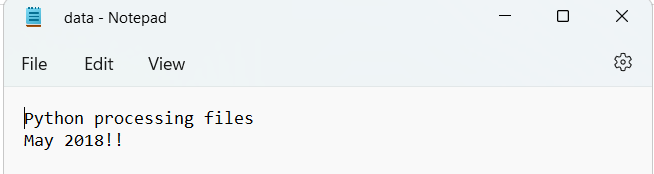
**Code:**

Filehndl=open("data.txt","w+")

Filehndl.write("Python processing files\nMay 2018!!\n")

Filehndl.close()

**Output:**

****

**Practical 5:**

**Applying arithmetic and conditional operations on dataset**

**Code:**

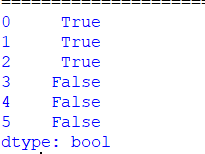
import pandas as pd

series1 = pd.Series([5,6,7,8,9,20])

temp=series1<8

print(temp)

**Output:**

****

**Code:**

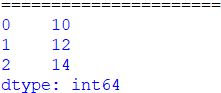
import pandas as pd

series1 = pd.Series([5,6,7,8,9,20])

temp=series1[series1<8]\*2

print(temp)

**Output:**

****

**Code:**

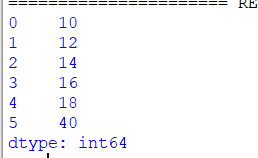
import pandas as pd

series1 = pd.Series([5,6,7,8,9,20])

temp=series1\*2

print(temp)

**Output:**

****

**Code:**

import pandas as pd

series1 = pd.Series([5,6,7,8,9,10])

series2 = pd.Series([11,12,13,14,15,16])

def AddSeries(x,y):

for i in range (len(x)):

print (x[i]+y[i])

def SubSeries(x,y):

for i in range (len(x)):

print(x[i]-y[i])

def MulSeries(x,y):

for i in range (len(x)):

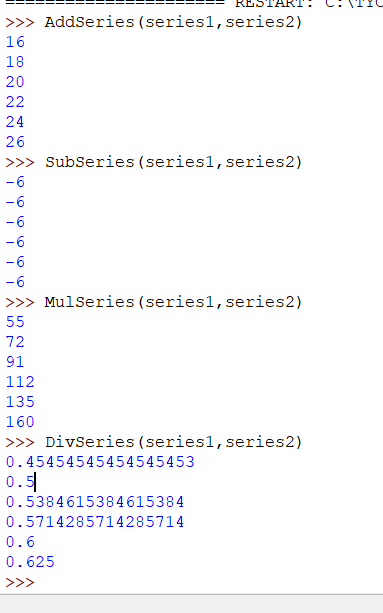
print (x[i]\*y[i])

def DivSeries(x,y):

for i in range (len(x)):

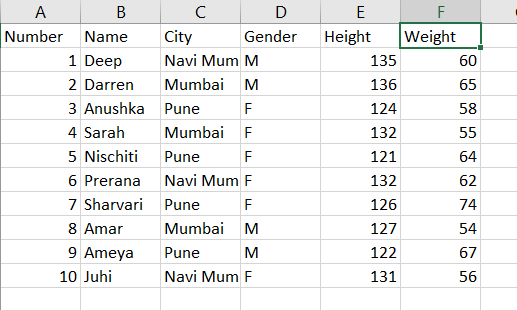
print (x[i]/y[i])

**Output:**

****

**Practical 6:**

**Aim:Crearting excel file and performing groupby on it.**

****

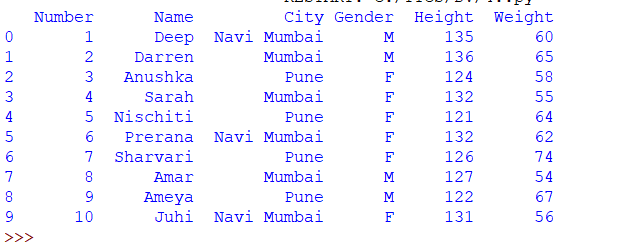
1. **Importing dataset into python IDLE.**

import pandas as pd

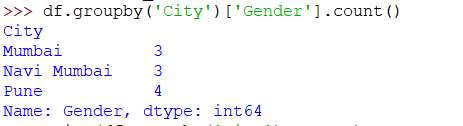
pd.read\_csv("C:\\TYCS\DV\Data.csv")

print(MyData)

MyData

****

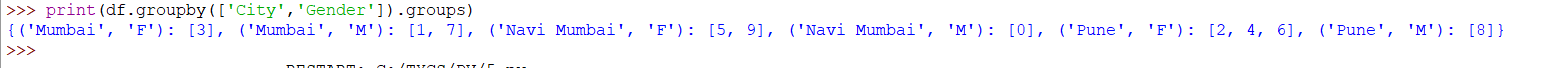
**Code and Output:**



**Code and Output:**

****

**Code and Output:**



**Practical 6.2:-**

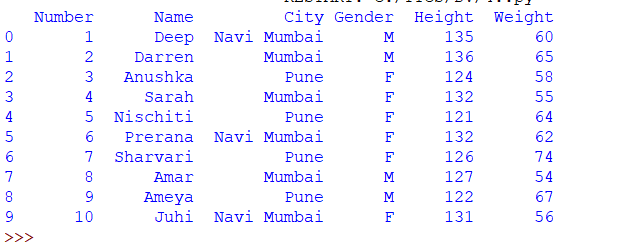
* **Importing dataset from excel.**

import pandas as pd

pd.read\_csv("C:\\TYCS\DV\Data.csv")

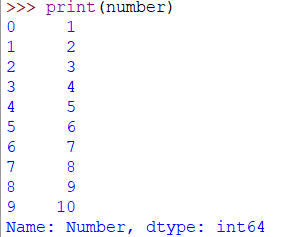
print(MyData)

MyData

****

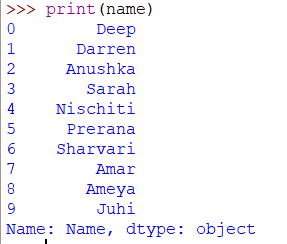
* number=dataset["Number"]

print(number)



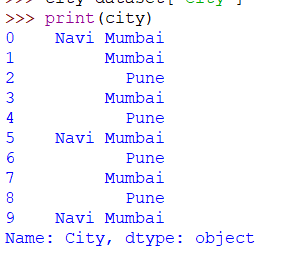
* name=dataset['Name']

print(name)



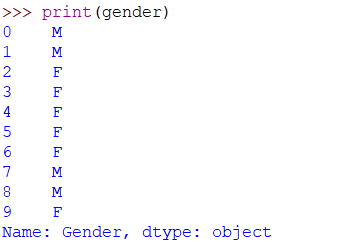
* city=dataset['City']

print(city)



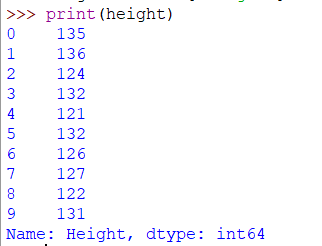
* gender=dataset['Gender']

print(gender)



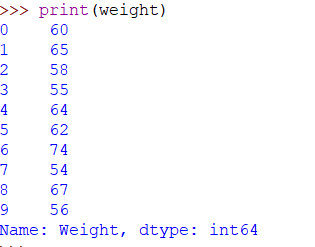
* height=dataset['Height']

print(height)

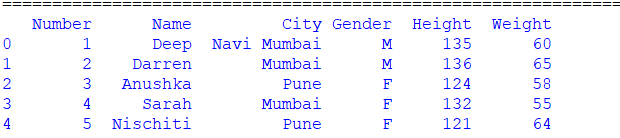


* weight=dataset['Weight']

print(weight)

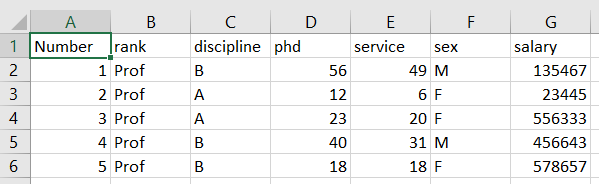


* print(dataset.head())



**Practical 7:-**

**Creating excel file and performing visualization by using line chart, bar plot to analyze it.**

****

**Code:**

import pandas as pd

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

#print(dataset)

rank=dataset["rank"]

discipline=dataset['discipline']

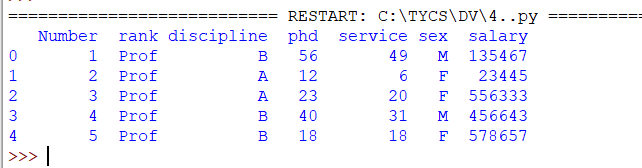
phd=dataset['phd']

service=dataset['service']

sex=dataset['sex']

salary=dataset['salary']

print(dataset.head())



* **Plotting a line chart:-**

import pandas as pd

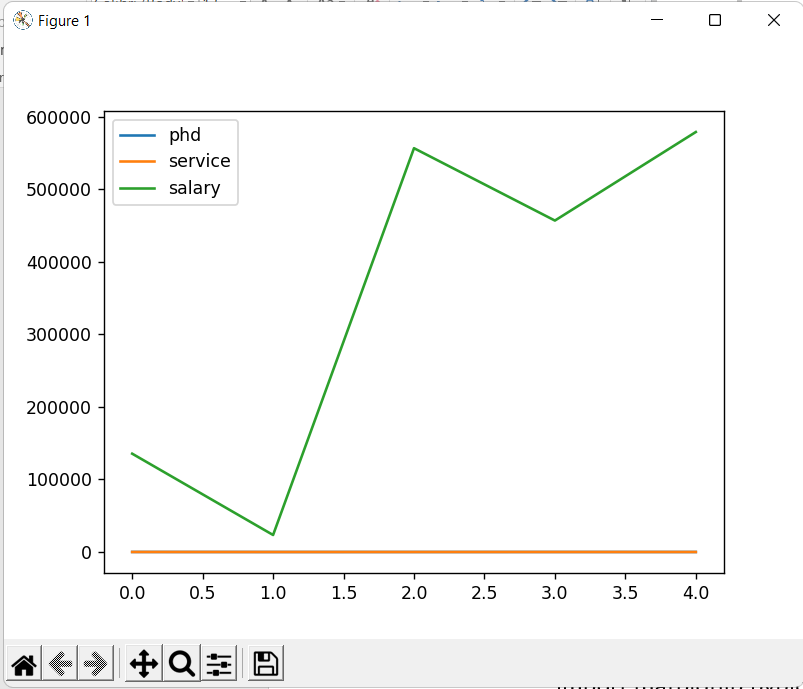
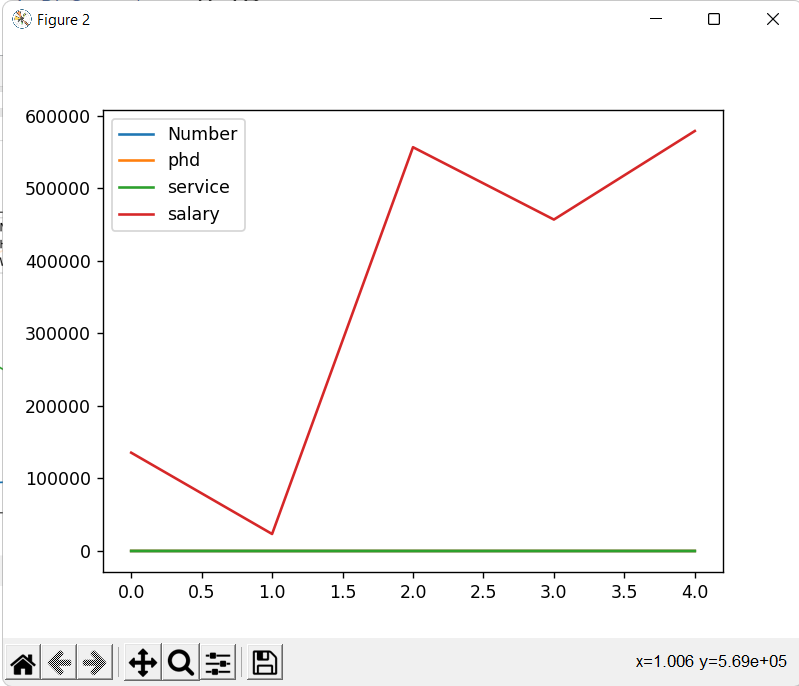
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset[["rank","discipline","phd","service","sex","salary"]].plot()

dataset.plot()

plt.show()

****

* **Line chart**

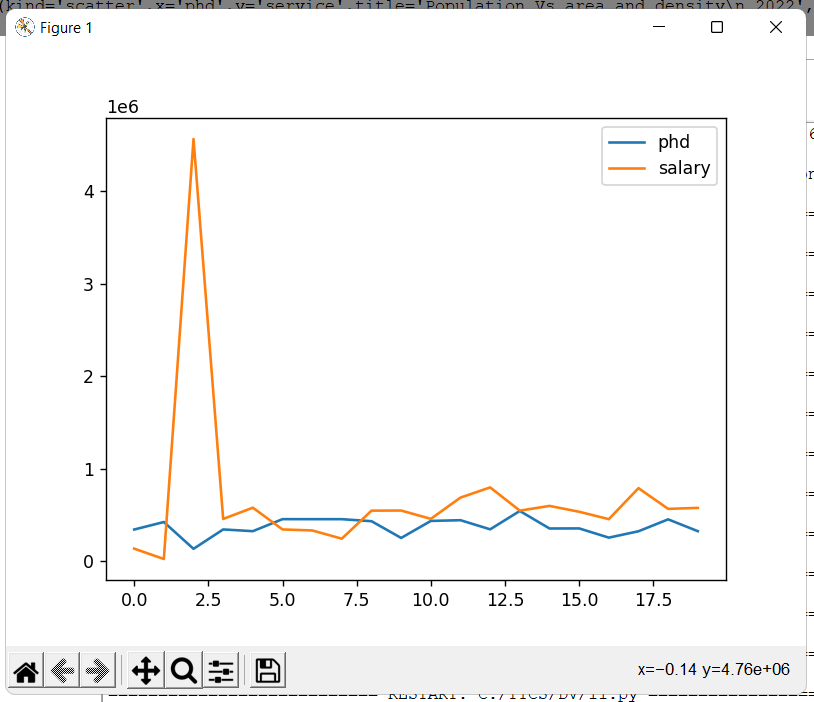
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset[["phd","salary"]].plot()

plt.show()

****

* **Bar chart**

import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Data.csv")

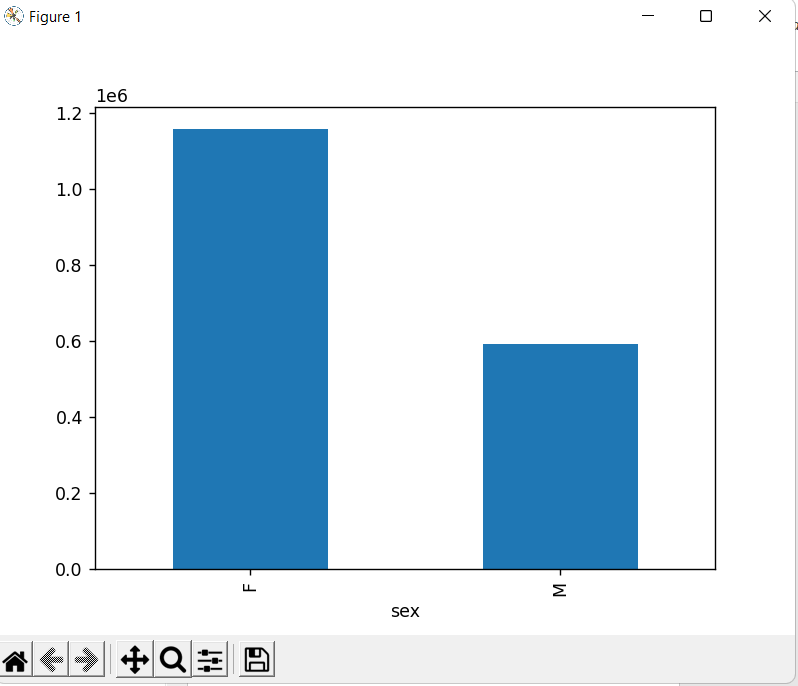
dataset1=dataset.groupby(['Height']).sum()

dataset1.sort\_values("Weight",ascending=False,inplace=True)

dataset1.head()

dataset1["Weight"].plot.bar()

plt.show()



* **Bar chart with and Without Title**

import pandas as pd

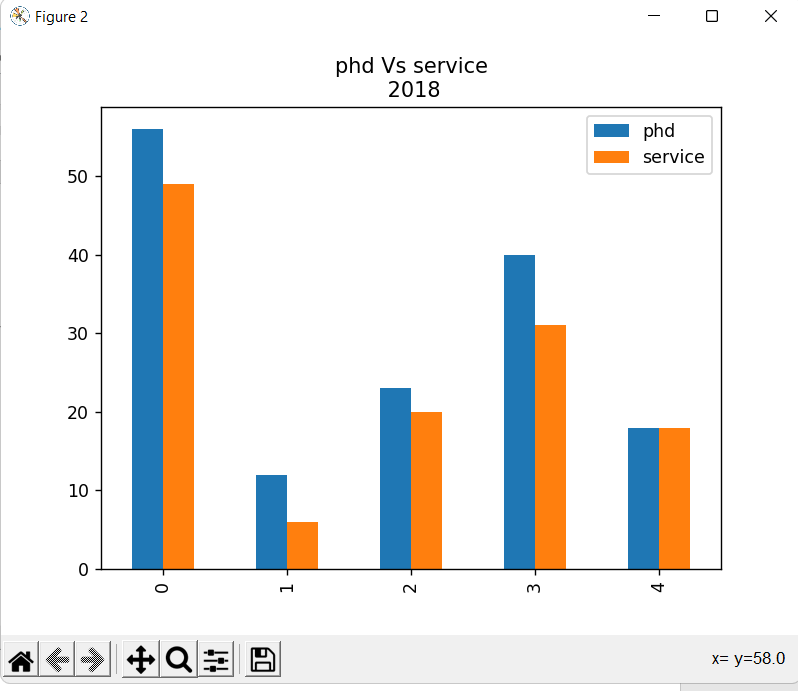
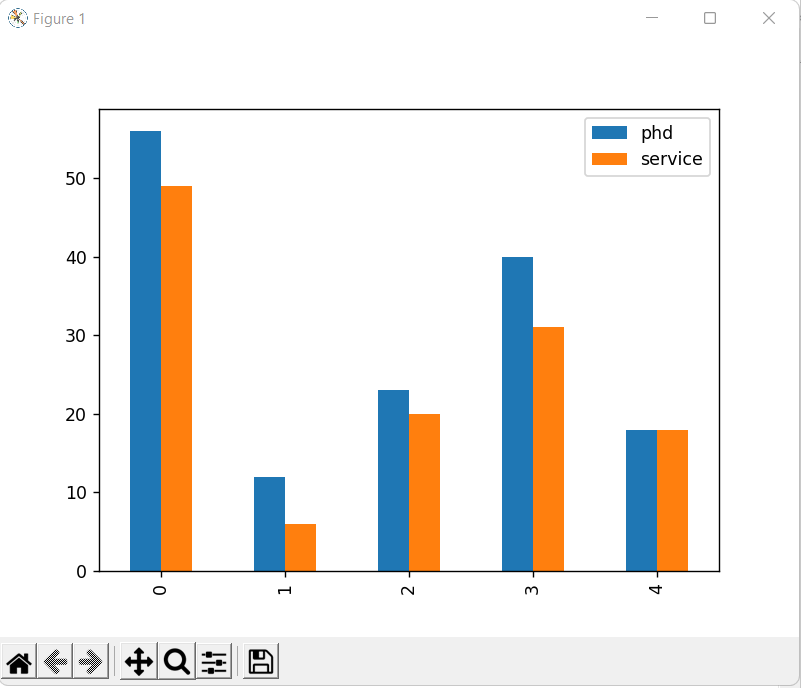
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset[['phd','service']].head(10).plot.bar()

dataset[['phd','service']].head(10).plot.bar(title="phd Vs service\n 2018")

plt.show()



* **Bar chart with color**

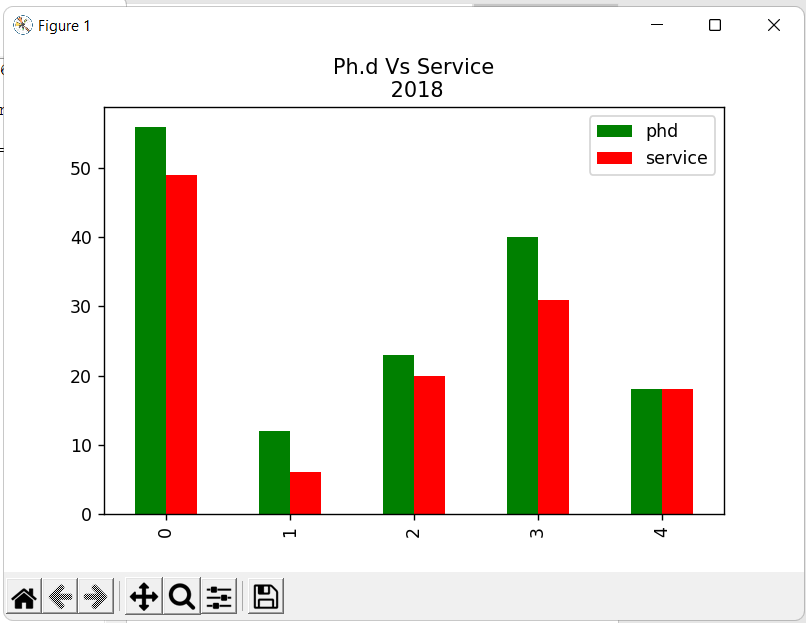
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset[['phd','service']].head(5).plot.bar(title="Ph.d Vs Service\n 2018",color=['g','red'])

plt.show()



**Practical 8: Create Excel sheet and perform following visualization**

* **Pie Chart**

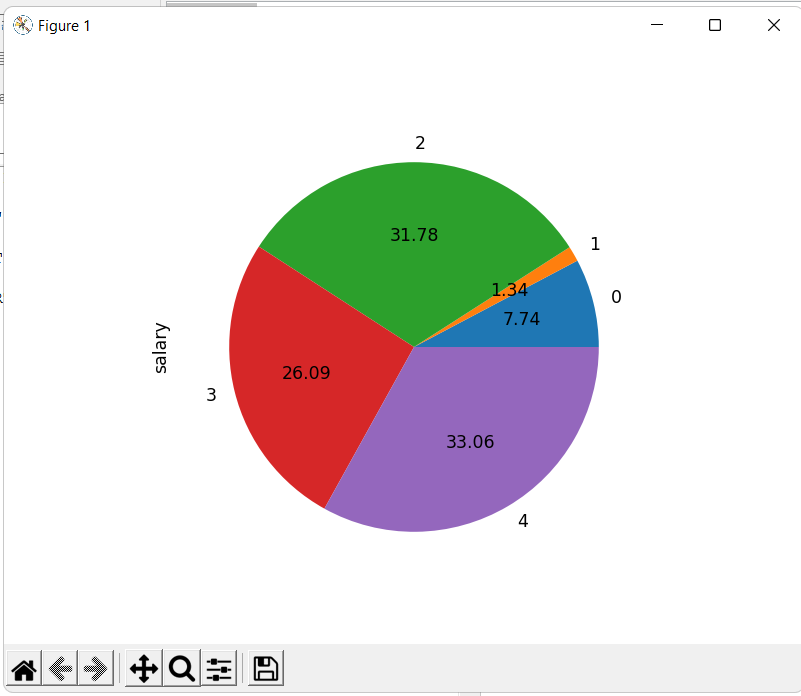
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset["salary"].head().plot.pie(autopct='%.2f')

plt.show()

****

* **Box Plot**

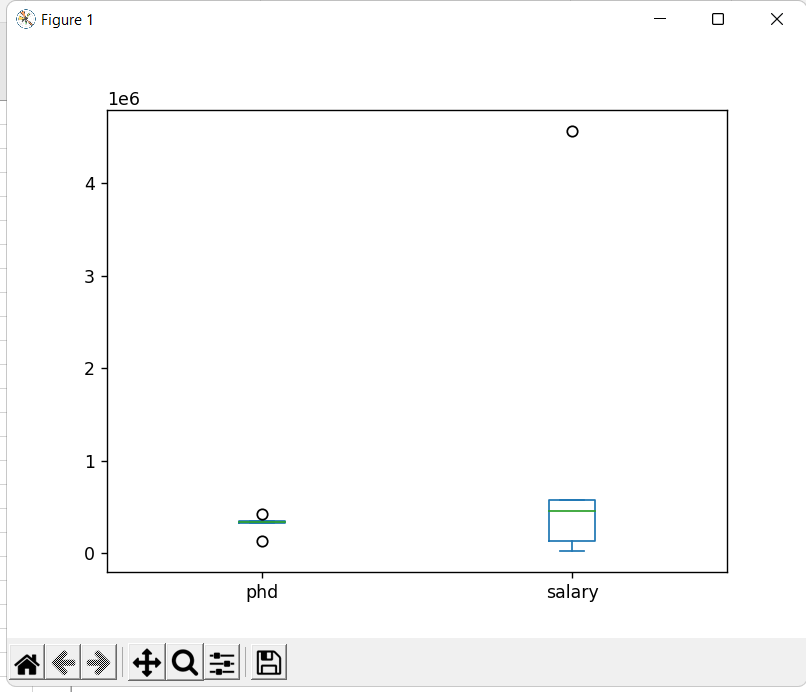
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset[["phd","salary"]].head(100).plot.box()

plt.show()



* **Histogram Plot**

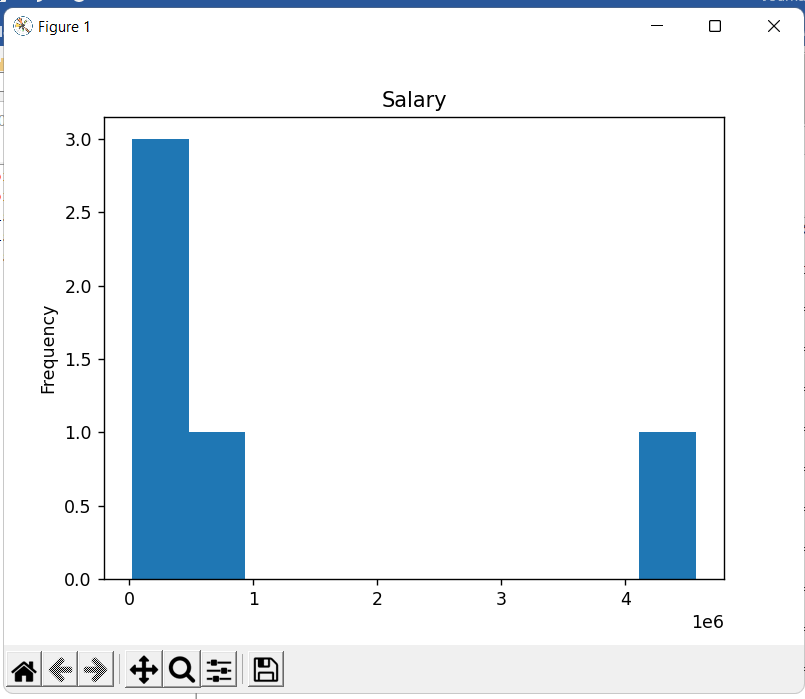
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset["salary"].head().plot.hist(title="Salary")

plt.show()



* **Scatter Plot**

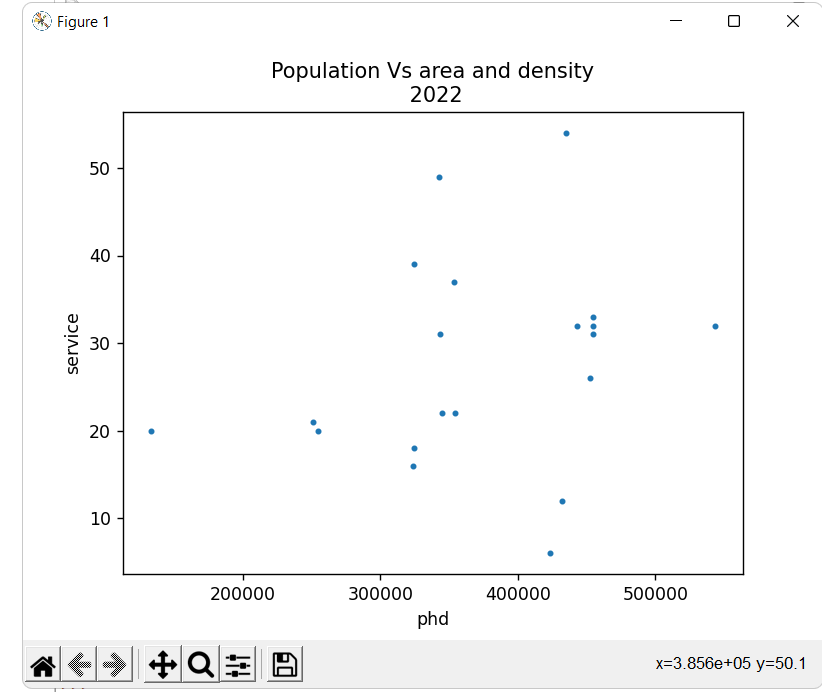
import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

dataset.plot(kind='scatter',x='phd',y='service',title='Population Vs area and density\n 2022',s=6.0)

plt.show()



**Practical 9:**

**Plotting the graph and performing visualization and analysis using seaborn library.**

**Code:**

import seaborn as sns

import pandas as pd

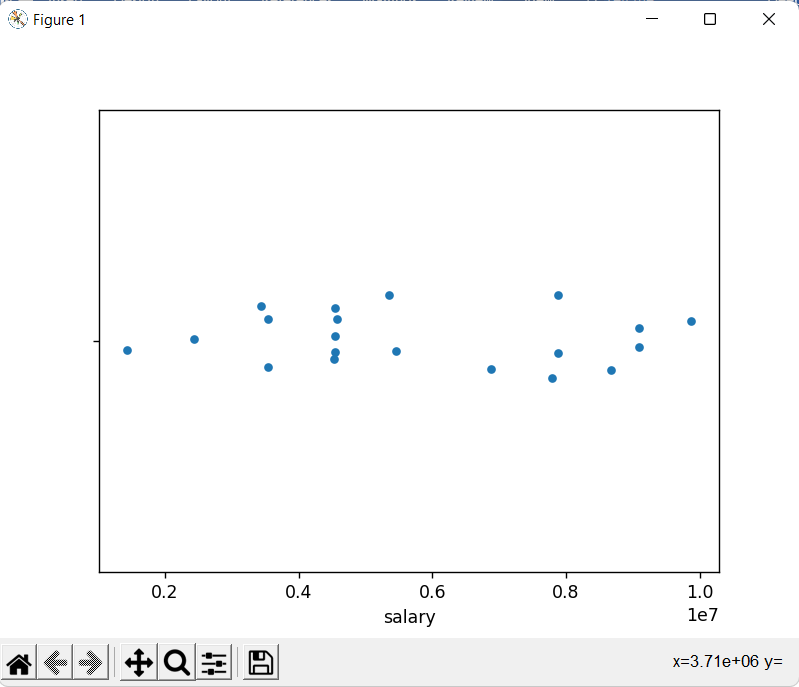
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x=dataset['salary'])

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

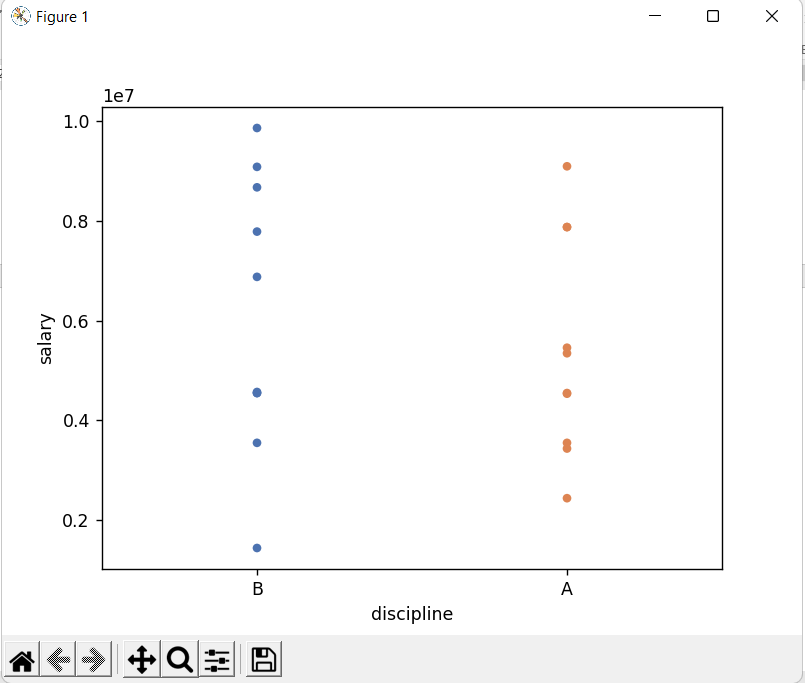
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x=dataset['discipline'],y=dataset['salary'],data=dataset,palette='deep',jitter=False)

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

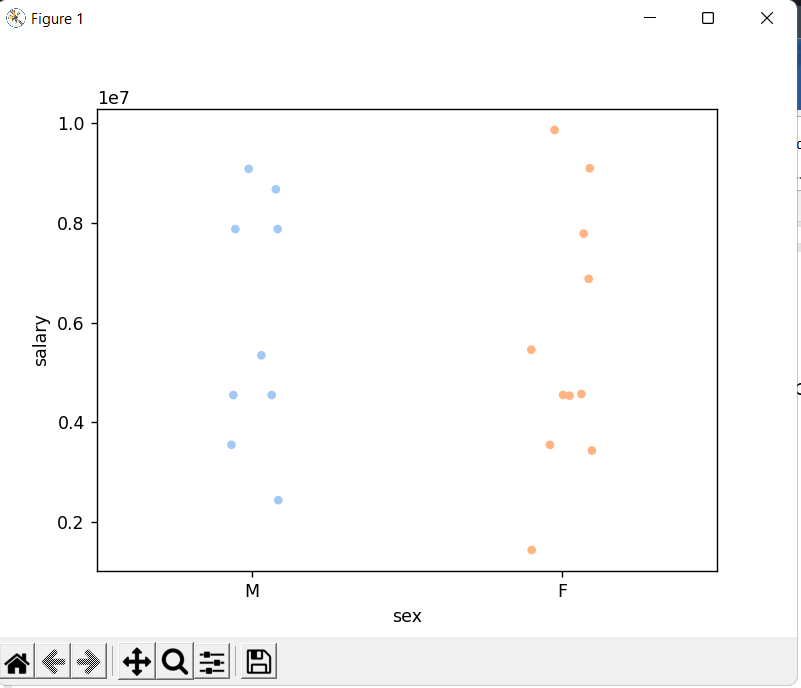
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x=dataset['sex'],y=dataset['salary'],data=dataset,palette='pastel')

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

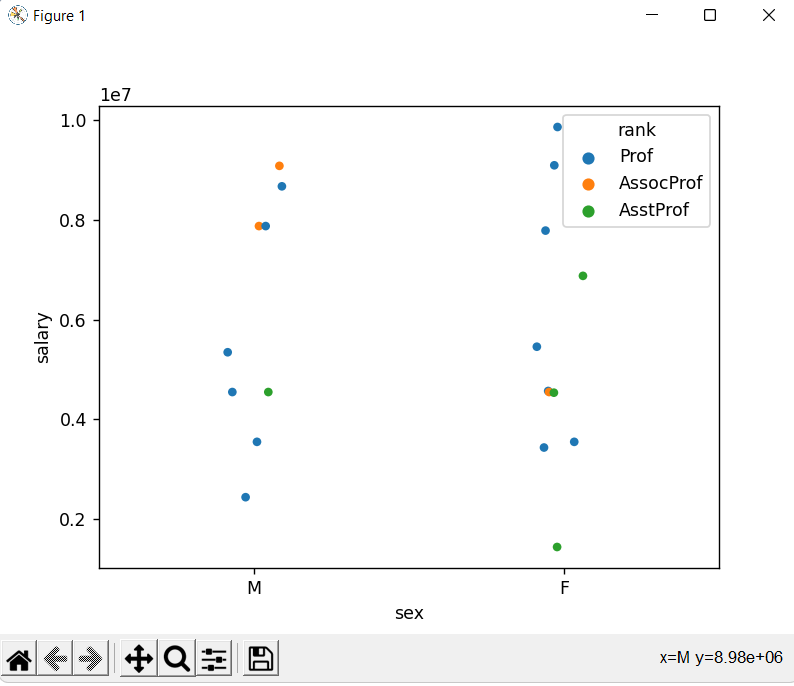
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x=dataset['sex'],y=dataset['salary'],data=dataset,hue=dataset['rank'],jitter=True)

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

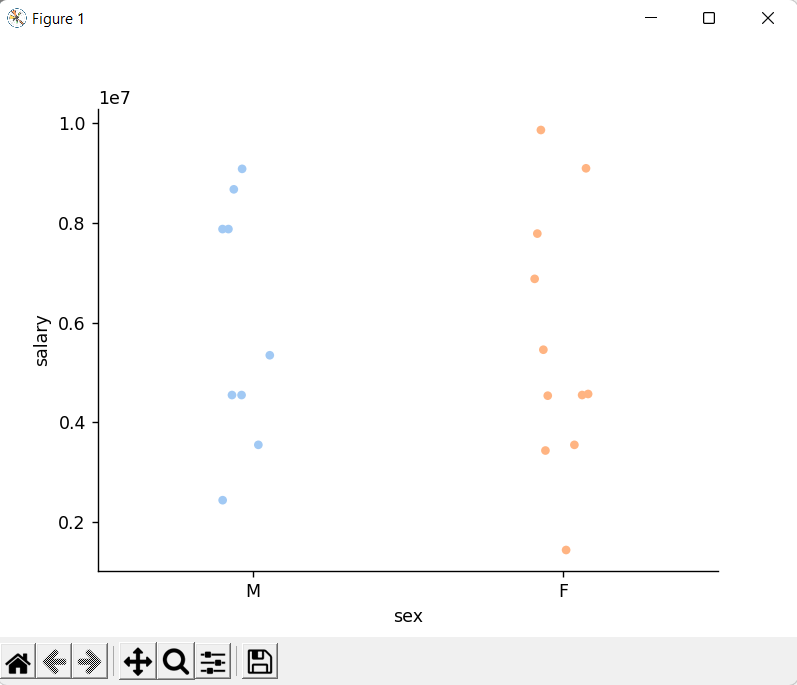
iris = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x='sex',y='salary',data=iris,palette='pastel',jitter=True,edgecolor='none')

sns.despine()

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

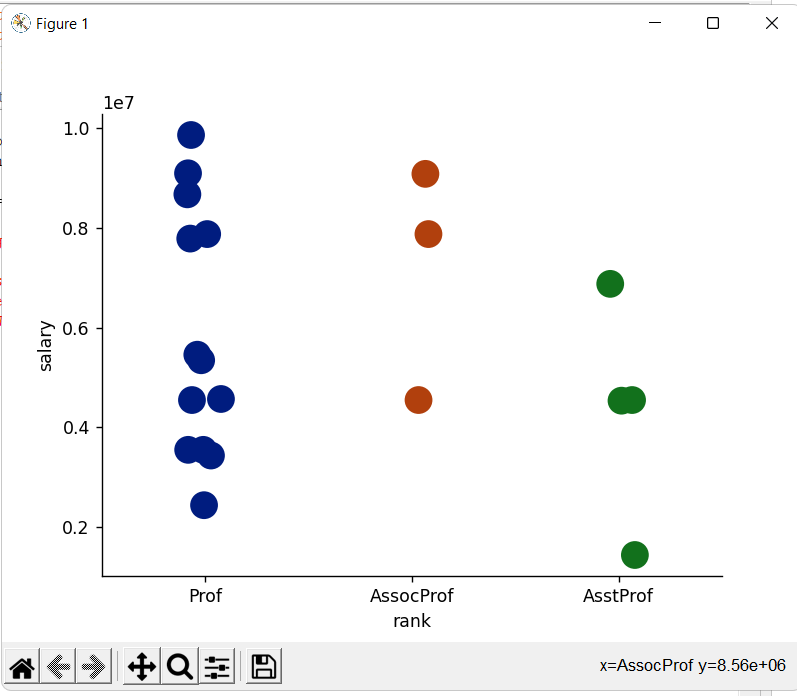
iris = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.stripplot(x='rank',y='salary',data=iris,palette='dark',size=16,jitter=True,edgecolor='none')

sns.despine()

plt.show()

**Output:**



**Code:Pie chart using seaborn**

import matplotlib.pyplot as plt

import seaborn as sns

data=[15,25,25,30,5]

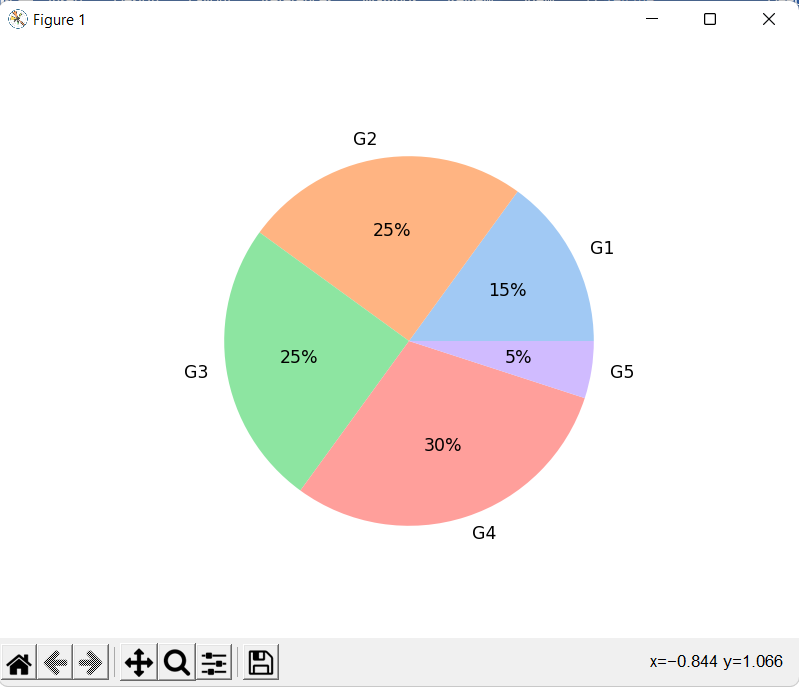
labels=['G1','G2','G3','G4','G5']

colors=sns.color\_palette('pastel')[0:5]

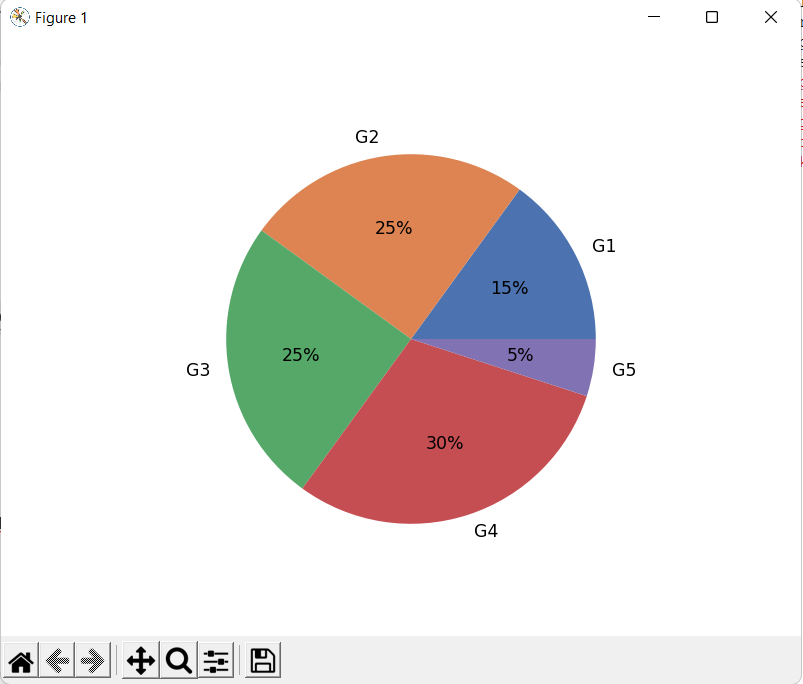
plt.pie(data,labels=labels,colors=colors,autopct='%.0f%%')

plt.show()

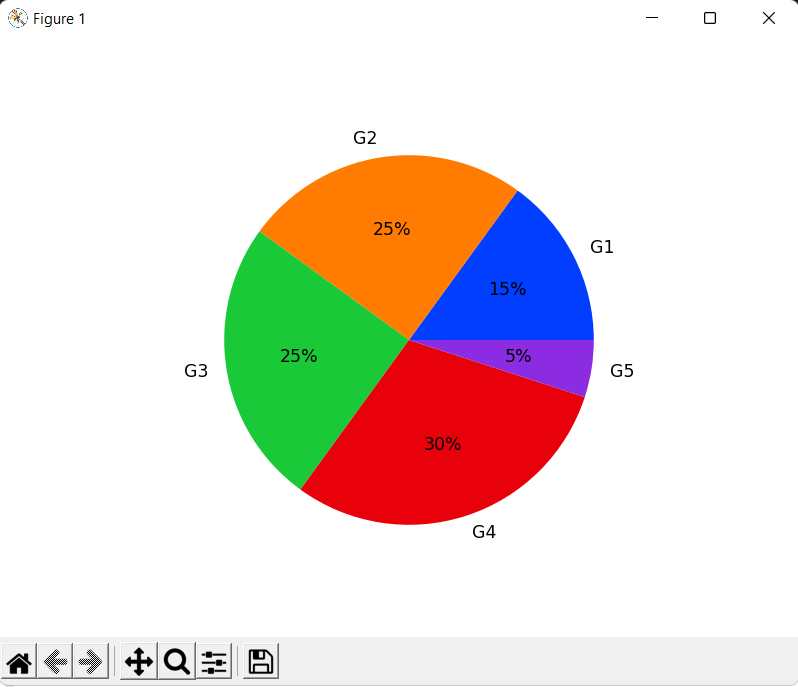
**Output:**



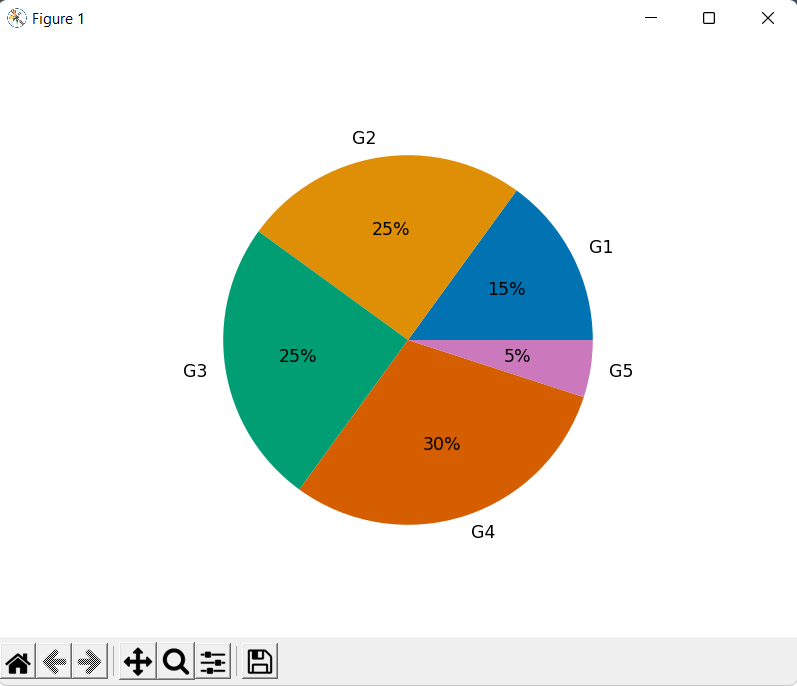
**b.deep**



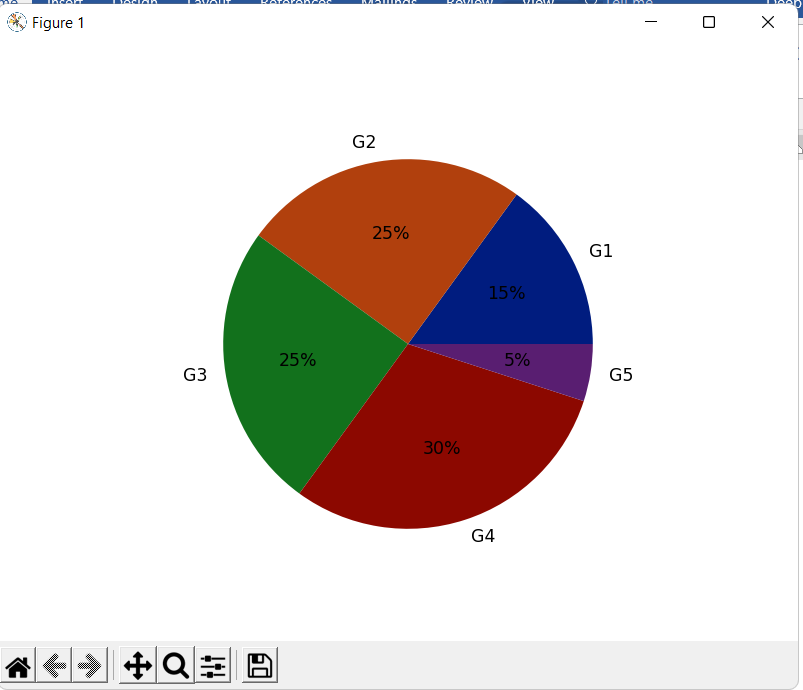
**c.bright**



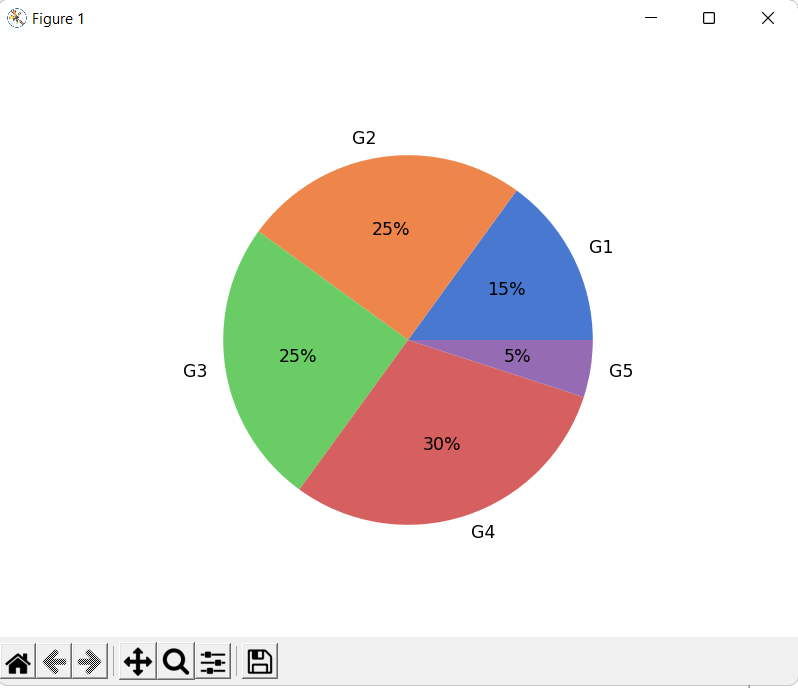
**d.colorblind**



**e.dark**



**f.muted**



**Code:**

import seaborn as sns

import pandas as pd

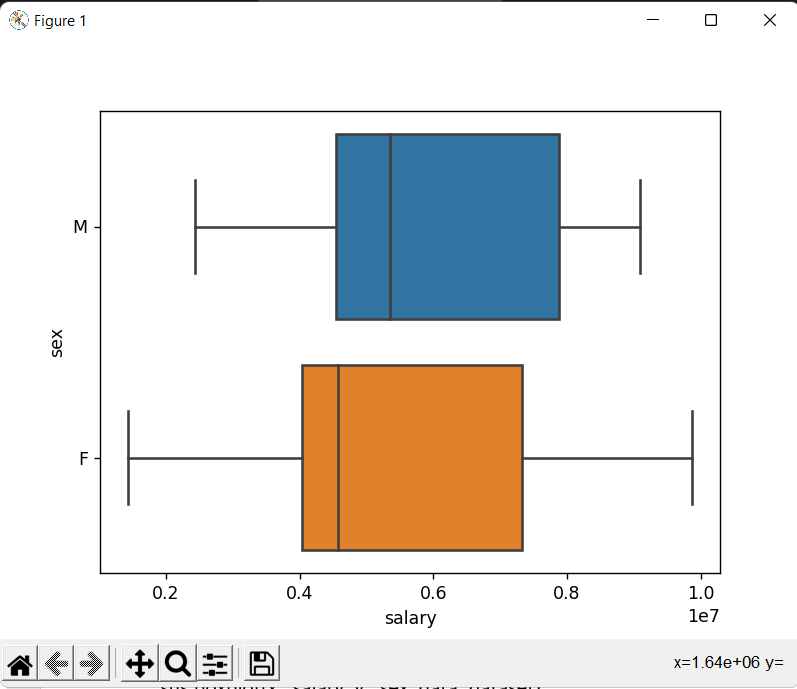
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x='salary',y='sex',data=dataset)

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

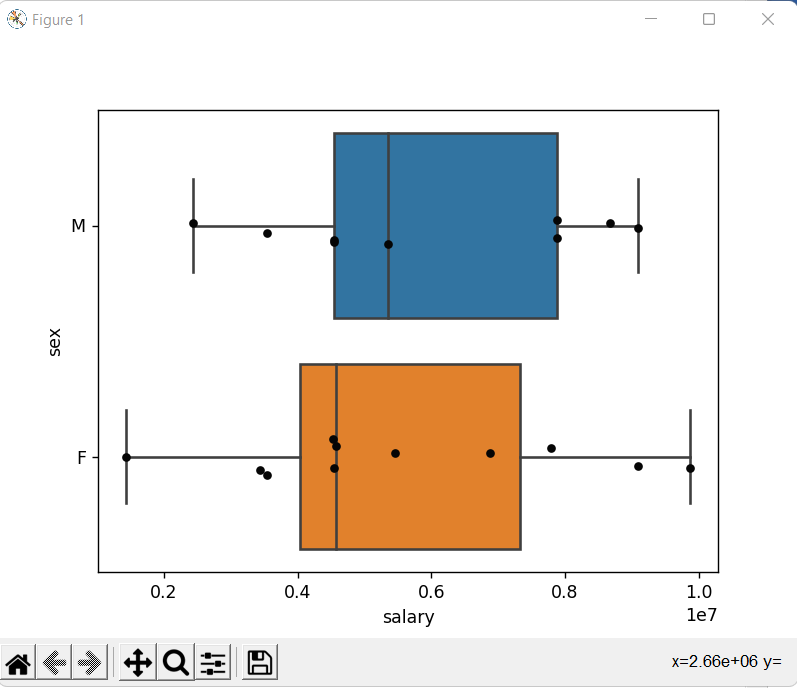
dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x='salary',y='sex',data=dataset)

sns.stripplot(x='salary',y='sex',data=dataset,jitter=True,color='0.02')

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

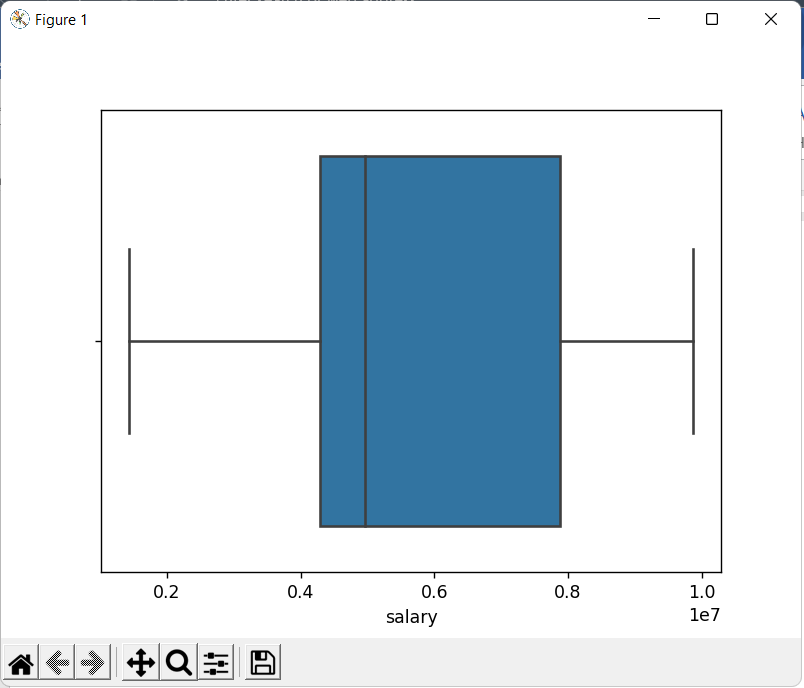
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x=dataset['salary'])

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

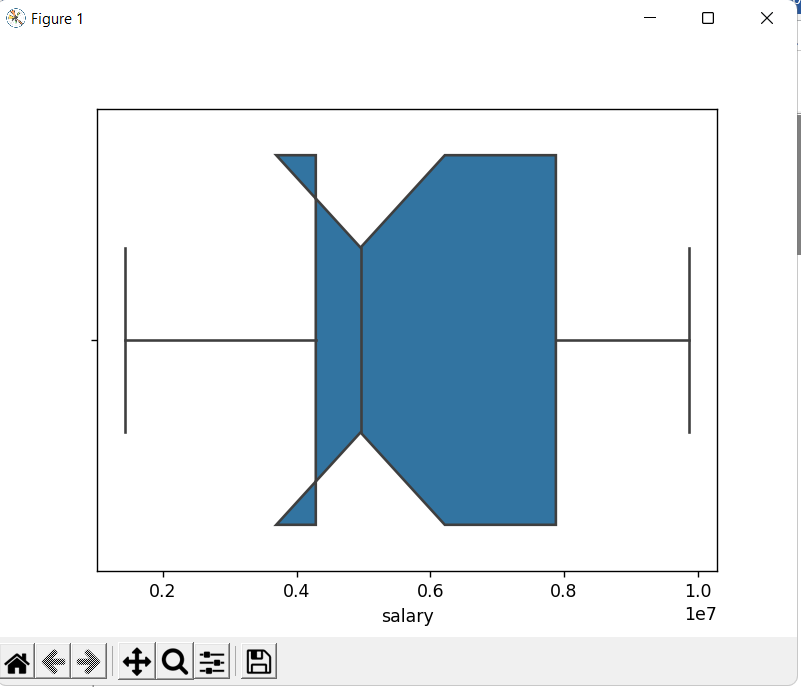
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x=dataset['salary'],notch=True)

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

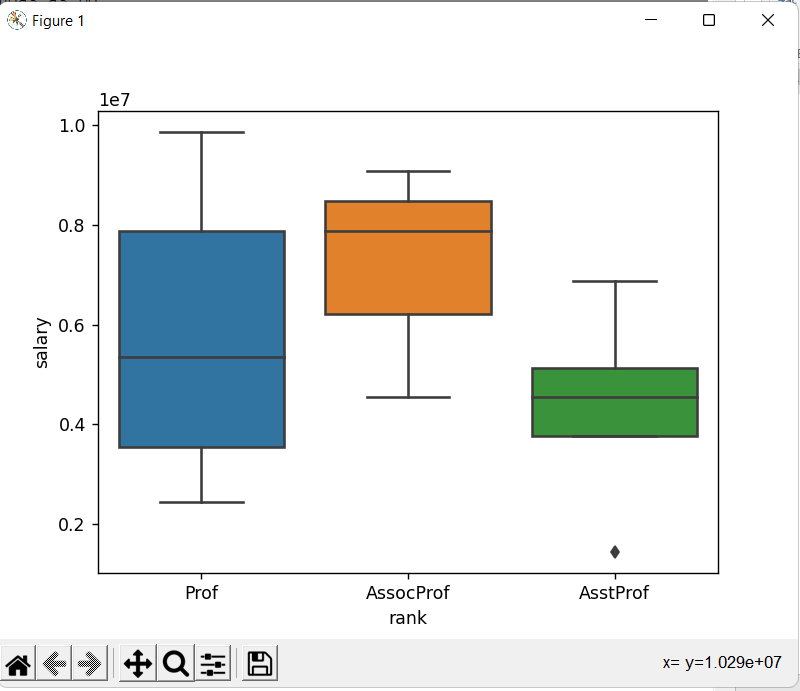
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x='rank',y='salary',data=dataset)

plt.show()

**Output:**



**Code:**

import seaborn as sns

import pandas as pd

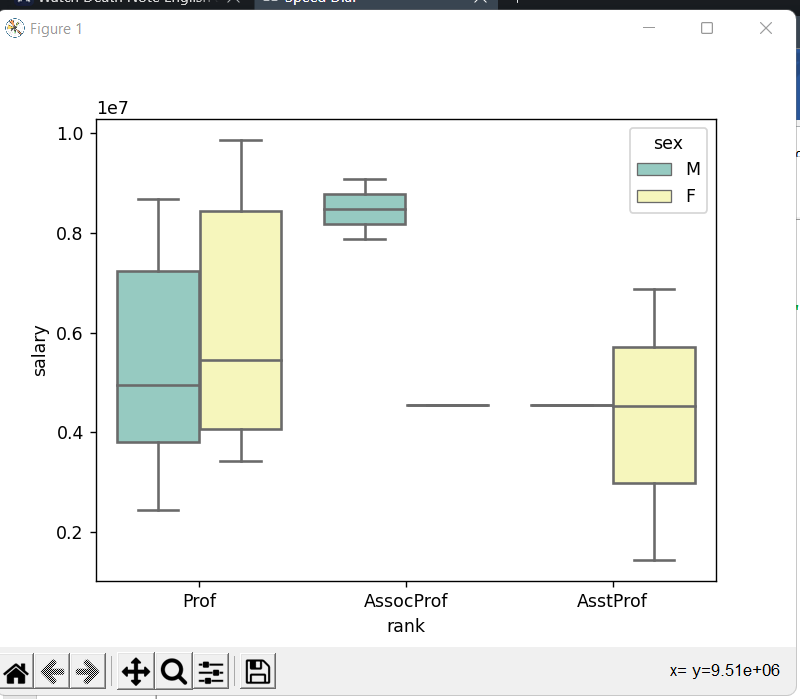
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x='rank',y='salary',hue='sex',data=dataset,palette='Set3')

plt.show()

**Output:**



**Practical 10: Plotting the graph and performing visualization and analysis using seaborn library.**

**Palette plot**

**Code:**

from matplotlib import pyplot as plt

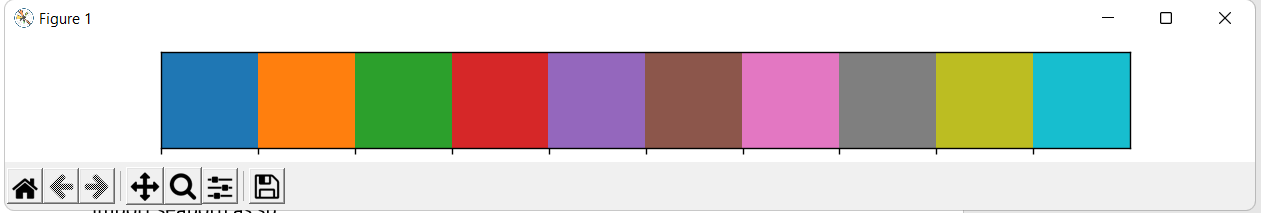
import seaborn as sb

current\_palette=sb.color\_palette()

sb.palplot(current\_palette)

plt.show()

**Output:**

****

**Code:**

from matplotlib import pyplot as plt

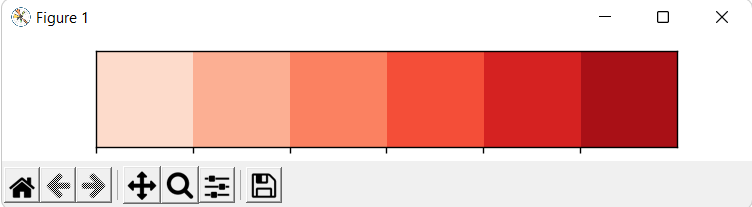
import seaborn as sb

current\_palette=sb.color\_palette()

sb.palplot(sb.color\_palette("Reds"))

plt.show()

**Output:**

****

**Code:**

from matplotlib import pyplot as plt

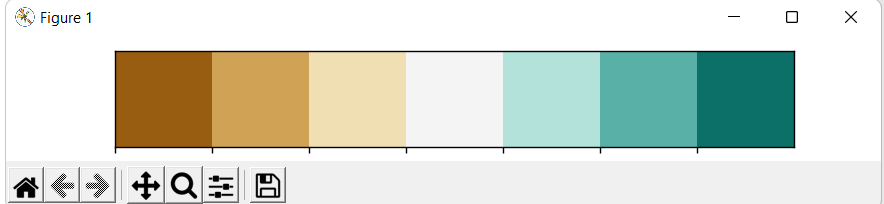
import seaborn as sb

current\_palette=sb.color\_palette()

sb.palplot(sb.color\_palette("BrBG",7))

plt.show()

**Output:**



**Swarm Plot**

**Code: box plot per rank**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

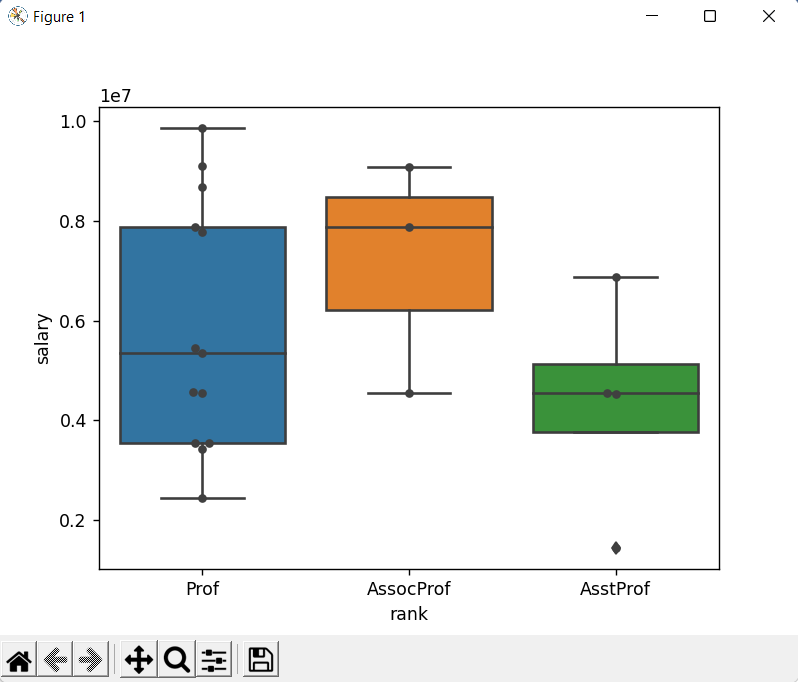
dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.boxplot(x='rank',y='salary',data=dataset)

sns.swarmplot(x='rank',y='salary',data=dataset,color='0.25')

plt.show()

**Output:**



**Code:swarm plotting of salary against gender.**

import seaborn as sns

import pandas as pd

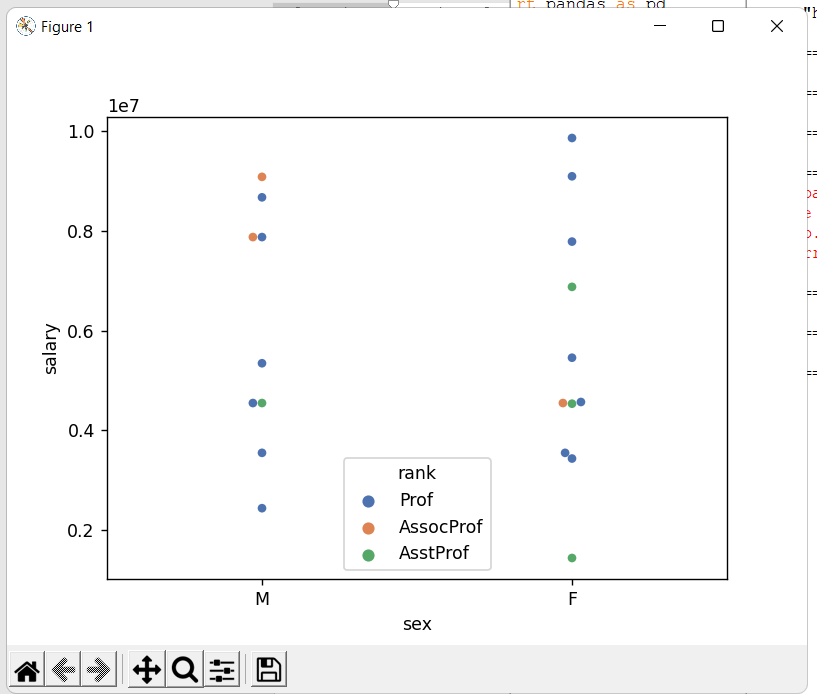
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.swarmplot(x='sex',y='salary',hue='rank',data=dataset,palette='deep',dodge=False)

plt.show()

**Output:**

****

**JointPlot**

**Code:**

import seaborn as sns

import pandas as pd

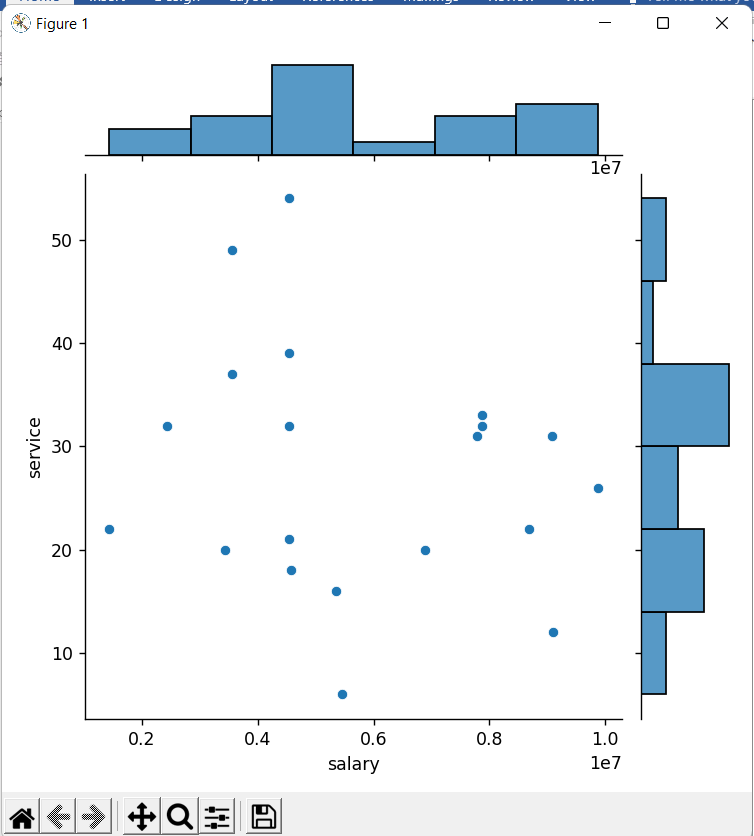
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.jointplot(x='salary',y='service',data=dataset)

plt.show()

**Output:**

****

**Code:**

import seaborn as sns

import pandas as pd

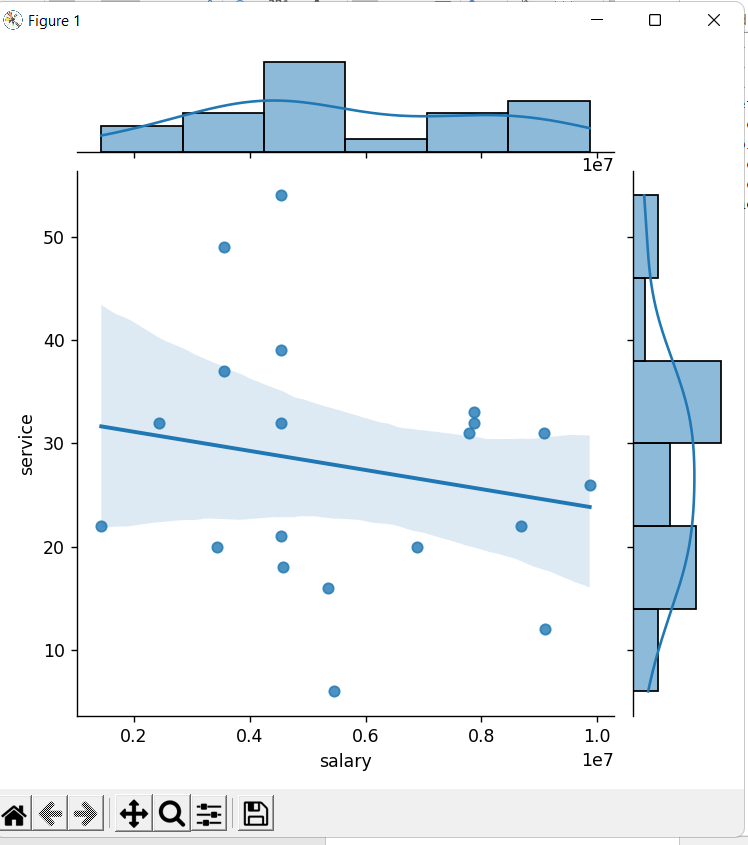
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.jointplot(x='salary',y='service',data=dataset,kind='reg')

plt.show()

**Output:**

****

**Code:**

import seaborn as sns

import pandas as pd

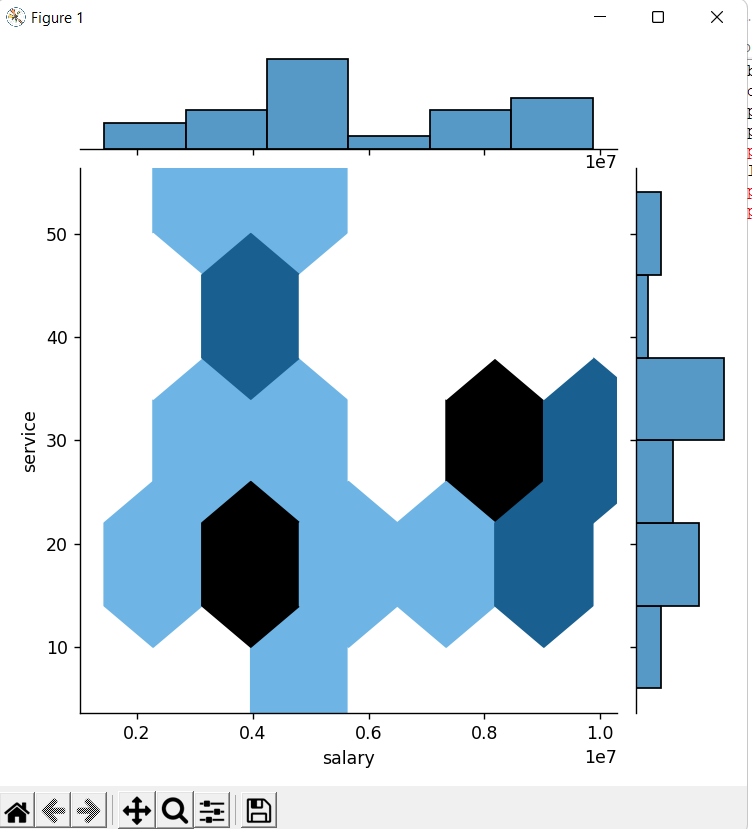
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.jointplot(x='salary',y='service',data=dataset,kind='hex')

plt.show()

**Output:**

****

**Code:**

import seaborn as sns

import pandas as pd

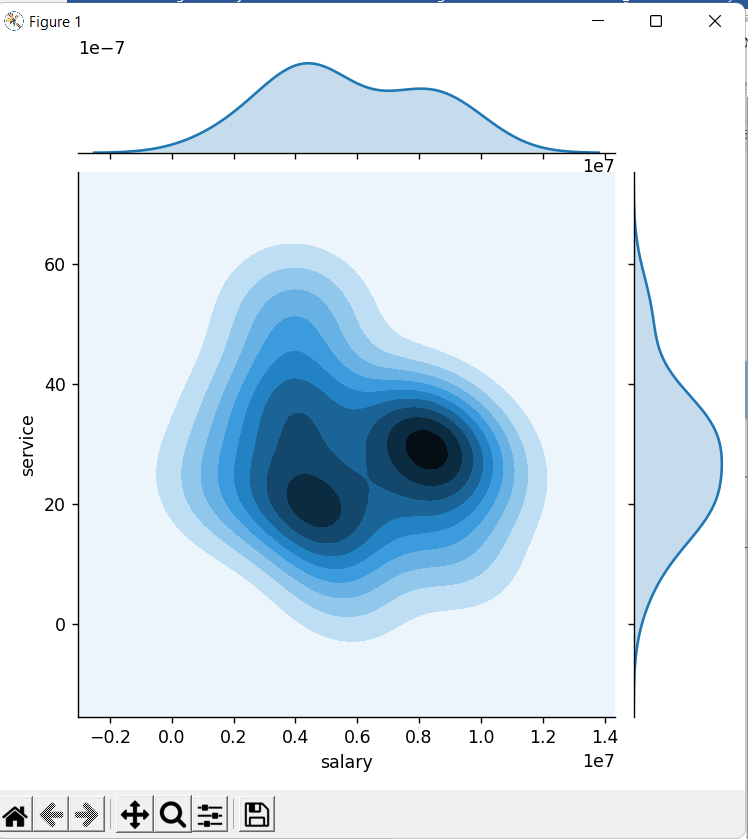
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.jointplot(x='salary',y='service',data=dataset,kind='kde')

plt.show()

**Output:**

****

**Code:**

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

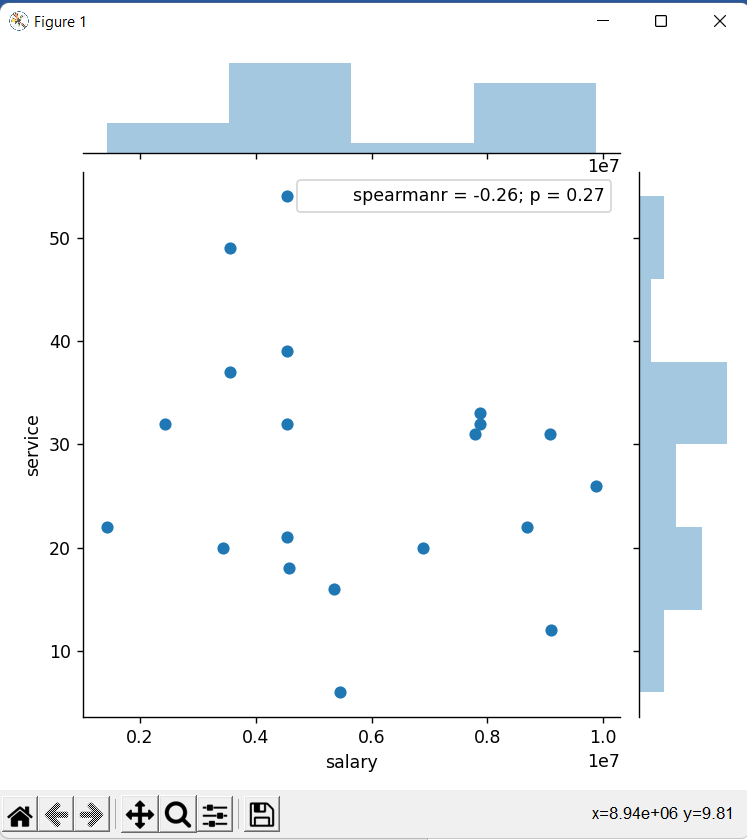
dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

from scipy.stats import spearmanr

sns.jointplot(x='salary',y='service',data=dataset,stat\_func=spearmanr)

plt.show()

**Output:**

****

**Code:**

from scipy.stats import spearmanr

import seaborn as sns

import pandas as pd

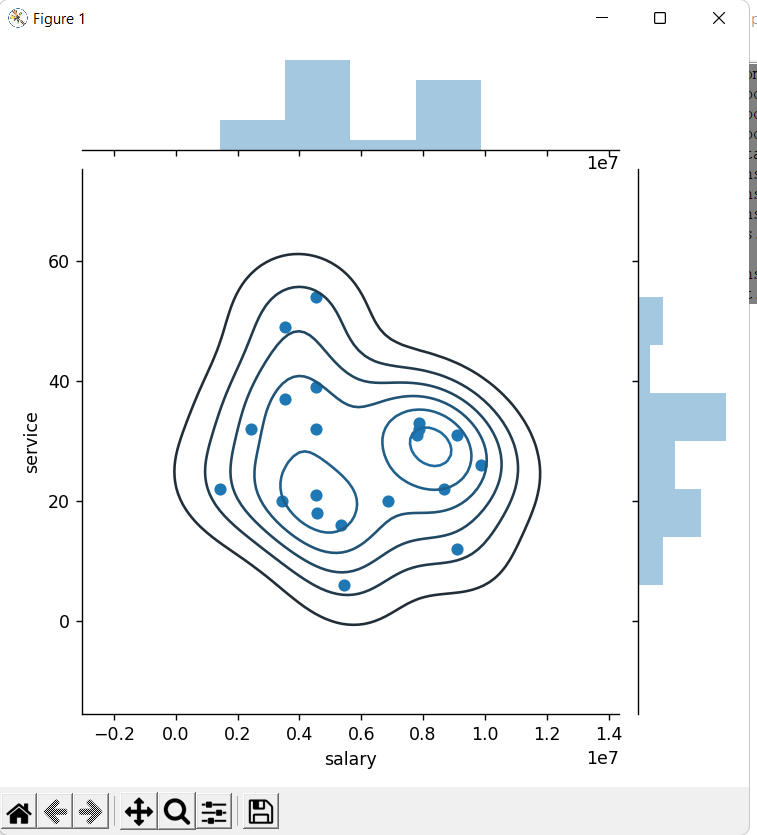
import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

sns.jointplot(x='salary',y='service',data=dataset).plot\_joint(sns.kdeplot,n\_levels=6)

plt.show()

**Output:**

****

Practical 11: **Using stack plot for given dataset to perform analysis**

**Code:**

from scipy.stats import spearmanr

import seaborn as sns

import pandas as pd

import matplotlib.pyplot as plt

dataset = pd.read\_csv("C:\\TYCS\DV\Salaries.csv")

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

sleeping= [7,8,6,11,7]

eating= [2,3,4,3,2]

working= [7,8,7,2,2]

playing= [8,5,7,8,13]

plt.plot([],[], color='m',label='sleeping')

plt.plot([],[], color='c',label='eating')

plt.plot([],[], color='r',label='working')

plt.plot([],[], color='k',label='playing')

plt.stackplot(days,sleeping,eating,working,playing,colors=['m','c','r','k'])

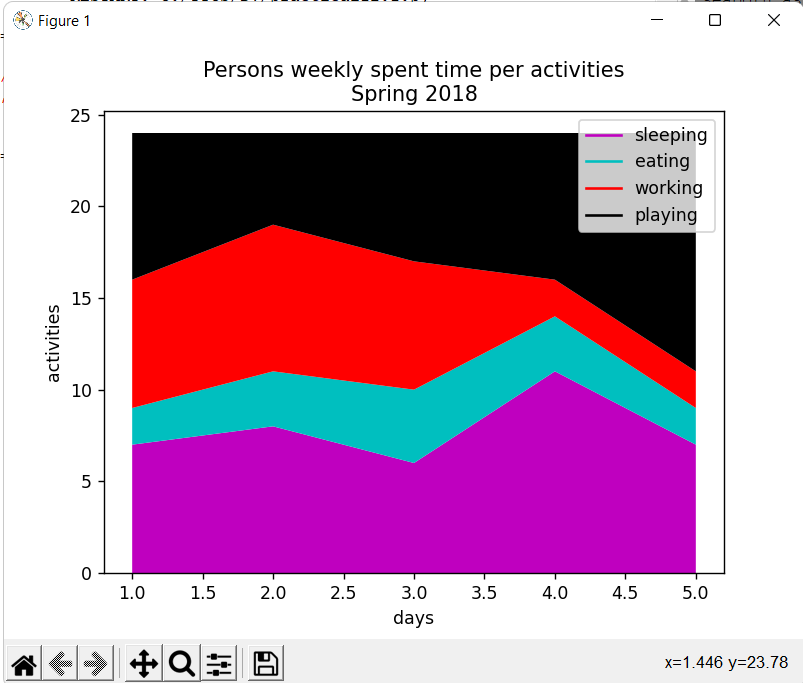
plt.xlabel('days')

plt.ylabel('activities')

plt.title('Persons weekly spent time per activities\nSpring 2018')

plt.legend()

plt.show()



Practical 12: **Creating frequency table for series and dataframe dataset**

**Frequency Table**

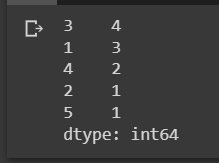
**Code:**

import pandas as pd

data=pd.Series([1,1,1,2,3,3,3,3,4,4,5])

data.value\_counts()

**Output:**

****

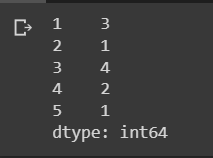
**Code:**

import pandas as pd

data=pd.Series([1,1,1,2,3,3,3,3,4,4,5])

data.value\_counts(sort=False)

**Output:**



**One Way Frequency Table frequency table for a dataframe**

**Code:**

import pandas as pd

df=pd.DataFrame({'Grade':['A','A','A','B','B','B','B','C','D','D'],

                 'Age':[18,18,18,19,19,20,18,18,19,19],

                 'Gender':['M','M','M','F','F','F','M','M','F','M']})

df

**Output:**

**Code:**

import pandas as pd

df=pd.DataFrame({'Grade':['A','A','A','B','B','B','B','C','D','D'],

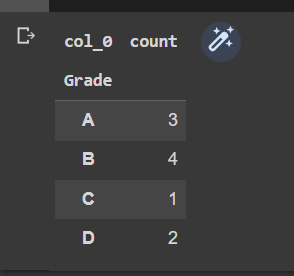
                 'Age':[18,18,18,19,19,20,18,18,19,19],

                 'Gender':['M','M','M','F','F','F','M','M','F','M']})

#df

pd.crosstab(index=df['Grade'],columns='count')

**Output:**

****

**Code:**

import pandas as pd

df=pd.DataFrame({'Grade':['A','A','A','B','B','B','B','C','D','D'],

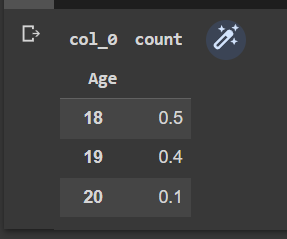
                 'Age':[18,18,18,19,19,20,18,18,19,19],

                 'Gender':['M','M','M','F','F','F','M','M','F','M']})

tab=pd.crosstab(index=df['Age'],columns='count')

tab/tab.sum()

**Output:**

****

**Two way Frequency table for a dataframe**

**Code:**

import pandas as pd

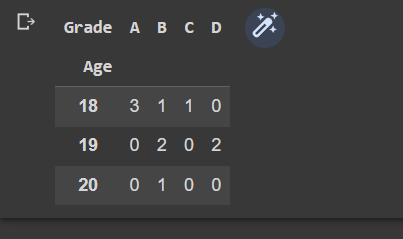
df=pd.DataFrame({'Grade':['A','A','A','B','B','B','B','C','D','D'],

                 'Age':[18,18,18,19,19,20,18,18,19,19],

                 'Gender':['M','M','M','F','F','F','M','M','F','M']})

pd.crosstab(index=df['Age'],columns=df['Grade

**Output:**

****