Ex no:1 **Download, install and explore the features of NumPy, SciPy, Jupyter, Statsmodels and Pandas packages.**

Aim:

To download, install and explore the features of NumPy, SciPy, Jupyter, Statsmodels and Pandas packages.

Process:

NUMPY:

Install:

1. Press comment + space bar to open spothight search. Type in terminal and press enter.

2. In the terminal, use the pip command to install numpy package.

3. Once the package is install successfully, type python to get into python prompt. Notice the python version is displayed too.

Description:

Numpy is a python library used for working with arrays. It also has function for working in domain od linear algebra, farrier transform and matrices. Numpy was created in 2005 by travis oliphant. It is an open source project and your can use it freely numpy strands for numerical python.

Command:

```
pip install numpy
```

SCIPY

Description: Scipy is a scientific computation library that uses numpy underneath. Scipy stands for scientific python. It provides more utility functions for optimization, stats and signed processing. Like numpy, scipy is open source so we can use it freely.

Command:

```
pip install scipy
```

JUPYTER

Description:

The pupyter notebook is an open source web application that you use to wide and share documents that contain live code, equation visualization.

Command:

```
Pip install jupyter
```

PANDA PACKAGE

Description:

Pandas is a python package that provides fast, flexible and expressive data structures designed to make making with “relation” or “labelled” data path.

Command:

```
Pip install panda
```

SATE MODELS

Description:

State models is a python module that provide classes and functions for the estination of many different statical models as well as conducting statical.

Command:

```
pip install state model
```

Ex no :2 # Working With Numpy Arrays

Ex no :1

Program:

```
import numpy as np
arr=np.array([1,2,3,4,5])
print(arr)
print(type(arr))
```

output:

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Ex no :2

Program :

```
import numpy as np
arr=np.array([[1,2,3],[4,5,6]])
print(arr)
```

output:

```
[[1 2 3]
[4 5 6]]
```

Ex no :3

Program:

```
import numpy as np
```

```
    arr=np.array([[1,2,3],[3,4,5,],[4,5,6]])
    print(arr)
   print(“after slicing”)
   print(arr[1:])
```

output :

```
    [[1,2,3],
     [3,4,5],
     [4,5,6]]
     After slicing
    [[3,4,5],
     [4,5,6]]
```

Ex no :4

Program:

```
    import numpy as np
    arr=np.array([[1,2,3],[3,4,5,],[4,5,6]])
    print(“our array is:”)
    print(arr)
    print(“the items in the second column are:”)
    print(arr[...,1])
    print('\n')
    print(“the items in the second row are:”)
    print(arr[1,….])
```

output:

    our array is:

[[1 2 3]

[3 4 5]

[4 5 6]]

the items in the second column are:

[2 4 5]

The items in the second row are:

[3 4 5]

Ex.no:3 # working with pandas data

Ex no:1

Program:

```
import pandas as pd
S=pd.Series([11,28,72,3,5,8])
print(S)
```

output:

```
0   11
1   28
2   72
3   3
4   5
5   8
dtype: int64
```

Ex no :2

Program:

```
import pandas as pd
a=[1,7,2]
myvar=pd.Series(a)
print(myvar)
```

output:

```
0   1
1   7
2   2
dtype: int64
```

Ex no :3

Program:

```
import pandas as pd
data={
        "calories":[420,380,390],
        "duration":[50,40,45]
        }
df=pd.DataFrame(data)
print(data)
```

output:

	calories	duration
0	420	50
1	380	40
2	390	45

Ex no:4

Program:

```
import pandas as pd
data={
        "calories":[420,380,390],
        "duration":[50,40,45]
        }
df=pd.DataFrame(data,index=["day 1","day 2","day 3"])
print(df)
```

output:

	calories	duration
day 1	420	50
day 2	380	40
day 3	390	45

Ex no :5

Program:

```
import pandas as pd
a=[1,7,2]
myvar=pd.Series(a,index=["x","y","z"])
print(myvar)
```

output:

```
x    1
y    7
z    2
dtype: int64
```

Ex no:6

Program:

```
import pandas as pd
calories={“day 1”:420,”day 2”:380,”day 3”:390}
myvar=pd.Series(calories)
print(myvar)
```

output:

```
day 1          420
day 2          380
```

day 3 390

dtype:int64

Ex no :7

Program:

```
import pandas as pd
calories={“day 1”:420,”day 2”:380,”day 3”:390}
myvar=pd.Series(calories,index=[“day 1”,”day 2”])
print(myvar)
```

output:

```
day 1 420
day 2 380
dtype:int64
```

Ex. No: 4 a) READING DATA FROM TEXT FILES

PROGRAM:~

```
f=open('data.csv','rt',encoding ='Windows-1252')
line=f.read ()
print('File content:\n', line.strip())
f.close
line2=line.split()
from collections import Counter
import numpy as np
import matplotlib.pyplot as plt
word_list=line2
couunts =Counter(word_list)
labels=zip(*Counter.items())
values=zip(*Counter.items())
indsort=np.argsort(values)[::-1]
labels =np.arrray (labels)[indsort]
values=np.array (values)[indsort]
indexes = np.array(len(labels))
bar_width=0.35
plt.figure(figsize=(15,5))
plt.bar(indexes,values)
plt.xticks(indexes+bar_width,lables)
```

```
plt.show()
```

OUTPUT:-

```
File contents : Line 1  Sample text
Line 2  Good day
Line 3  Feeling happy
Line 4  Just be cool
Line 5  Good practice
Line 6  Happy day
```

b). Exploring various commands for doing descriptive analystics on the Data set.

Aim:-

PROGRAM:~

```
import pandas as pd
df=pd.read_csv('data.csv')
df.head()
df.shape
df.info()
df.describe()
df.isnull().sum()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  169 non-null    int64  
 1   Pulse     169 non-null    int64  
 2   Maxpulse  169 non-null    int64  
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
```

Ex no:5 Use the diabetes data set from UCI and Pima Indians Diabetes data set for performing the following

Ex.5(a) univariate analysis : frequency, mean, median, variance, mode,standard deviation, skenwness and kurtosis

PROGRAM:~

```
import pandas as pd
import numpy as np
import statistics as st
df=pd. read_csv ('data.csv')
print(df.shape);print(df.info())
print('MEAN:\n', df.mean())
print('MEDIAN:\n', df.median())
print('MODE:\n', df.mode())
print('STANDAR DEVIATION:\n',df.std())
print('VARIANCE:\n', df.var())
print('SKEWNESS:\n', df.skew())
print('KURTOSIS:\n', df.kurtosis())
df.describe()
```

0UTPUT:~

```
(169, 4)
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 169 entries, 0 to 168

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	Duration	169 non-null	int64
1	Pulse	169 non-null	int64
2	Maxpulse	169 non-null	int64
3	Calories	164 non-null	float64

dtypes: float64(1), int64(3)

memory usage: 5.4 KB

None

MEAN:

Duration 63.846154

Pulse 107.461538

Maxpulse 134.047337

Calories 375.800000

dtype: float64

MEDIAN:

Duration 60.0

Pulse 105.0

Maxpulse 131.0

Calories 318.6

dtype: float64

MODE:

	Duration	Pulse	Maxpulse	Calories
0	60	100	120	300.0

STANDAR DEVIATION:

Duration 42.299949

Pulse 14.510259

Maxpulse 16.450434

Calories 266.377134

dtype: float64

VARIANCE:

Duration 1789.285714

Pulse 210.547619

Maxpulse 270.616793

Calories 70956.777546

dtype: float64

SKEWNESS:

Duration 2.863888

Pulse 1.418405

Maxpulse 0.701439

Calories 3.102184

dtype: float64

KURTOSIS:

Duration 10.187516

Pulse 2.573407

Maxpulse 0.692226

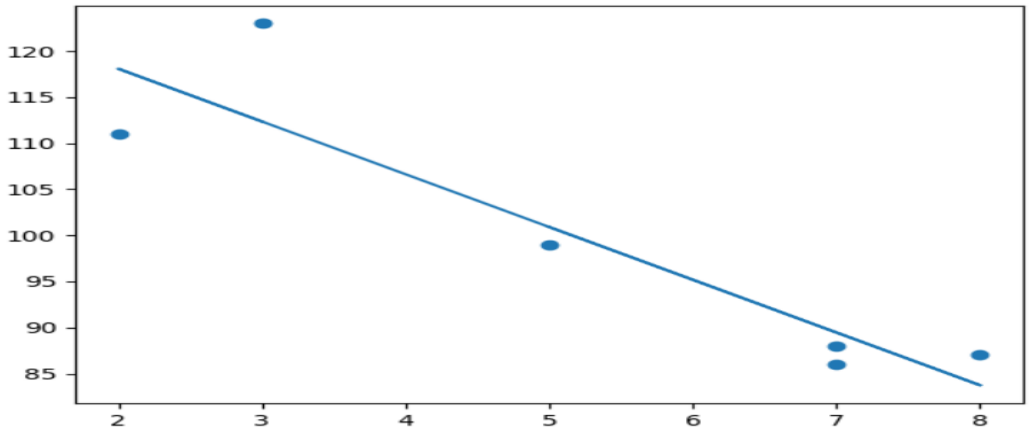
Calories 11.989747

dtype: float64

5. (b) Byvariate : Linear and logistic Regression modelling

Program 1:-

```
import sys
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
from scipy import stats
x = [5,7,8,7,2,3]
y = [99,86,87,88,111,123]
slope, intercept, r, p, std_err = stats.linregress(x, y)
def myfunc(x):
  return slope * x + intercept
mymodel = list(map(myfunc, x))
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```



Ex no:5(c) Multiple Regression analysis

Program :-

```
import numpy
from sklearn import linear_model
X = numpy.array([3.78, 2.44, 2.09, 0.14, 1.72, 1.65, 4.92, 4.37, 4.96, 4.52, 3.69, 5.88]).reshape(-1,1)
y = numpy.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
logr = linear_model.LogisticRegression()
logr.fit(X,y)
def logit2prob(logr, X):
  log_odds = logr.coef_ * X + logr.intercept_
  odds = numpy.exp(log_odds)
  probability = odds / (1 + odds)
  return(probability)
print(logit2prob(logr, X))
```

Output:~

```
[[0.60749955]
 [0.19268876]
 [0.12775886]
 [0.00955221]
 [0.08038616]
```

[0.07345637]

[0.88362743]

[0.77901378]

[0.88924409]

[0.81293497]

[0.57719129]

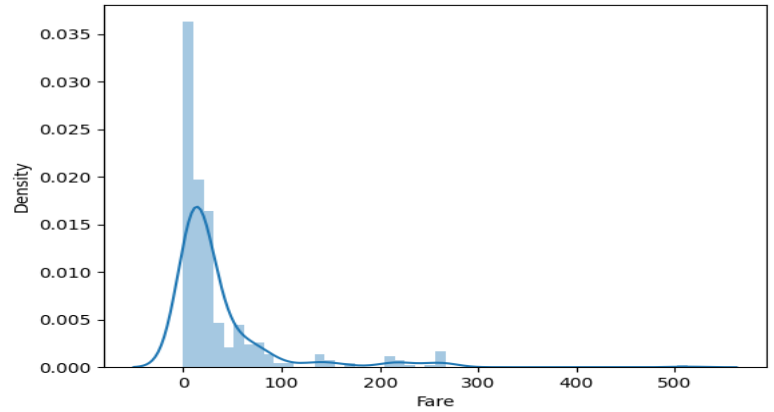
[0.96664243]]

EX.NO:6: Apply and Explore various ploting functions using Python

EX.NO:8A: BAR PLOT

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
sns.barplot(x='Age', y='Fare', data=df)
plt.savefig('EX8.png')
plt.show()
```

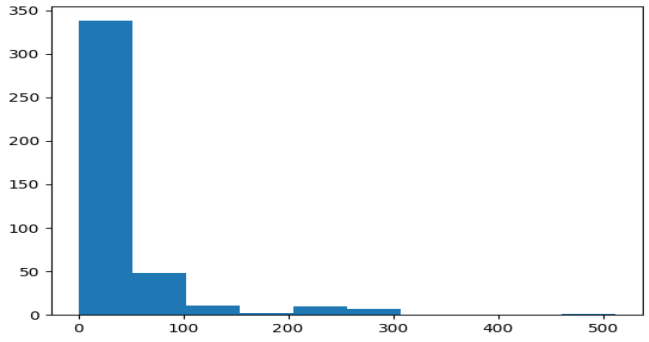


EX.NO:8B: HISTOGRAM

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
plt.hist(df["Fare"])
plt.savefig('EX8.1.png')
```

plt.show()

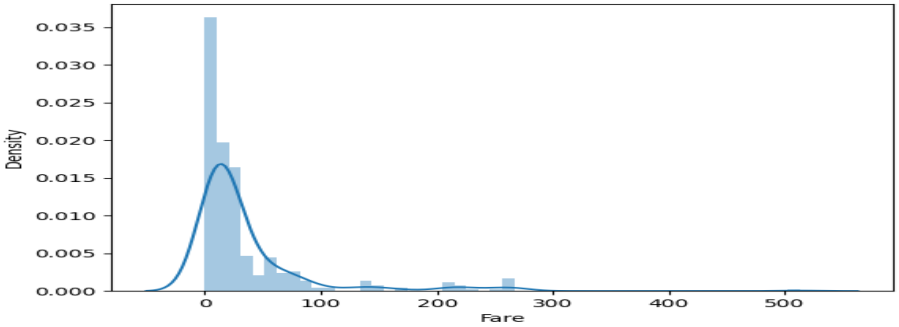


EX.NO:8C: DISTRIBUTION (or) DENSITY PLOT

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
sns.distplot(df["Fare"])
plt.savefig('EX8.2.png')
```

plt.show()

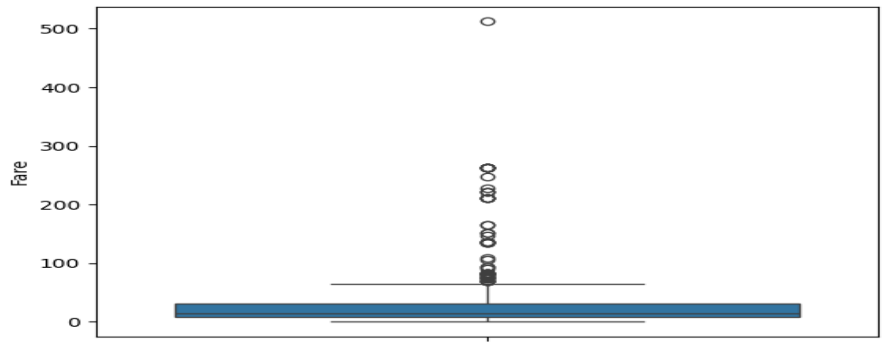


EX.NO:8D: BOX PLOT

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
sns.boxplot(df["Fare"])
plt.savefig('EX8.3.png')
```

plt.show()

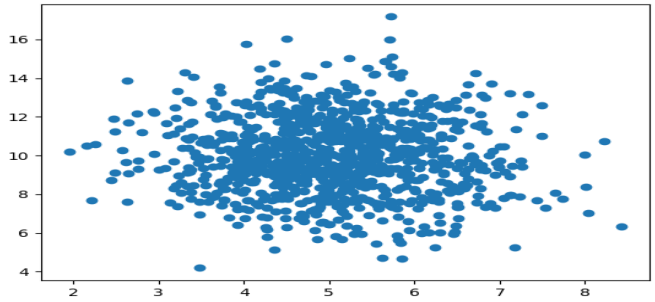


EX.NO:8E: SCATTER PLOT

Program:

```
import pandas as pd
import numpy
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
x=numpy.random.normal(5.0,1.0,1000)
y=numpy.random.normal(10.0,2.0,1000)
```

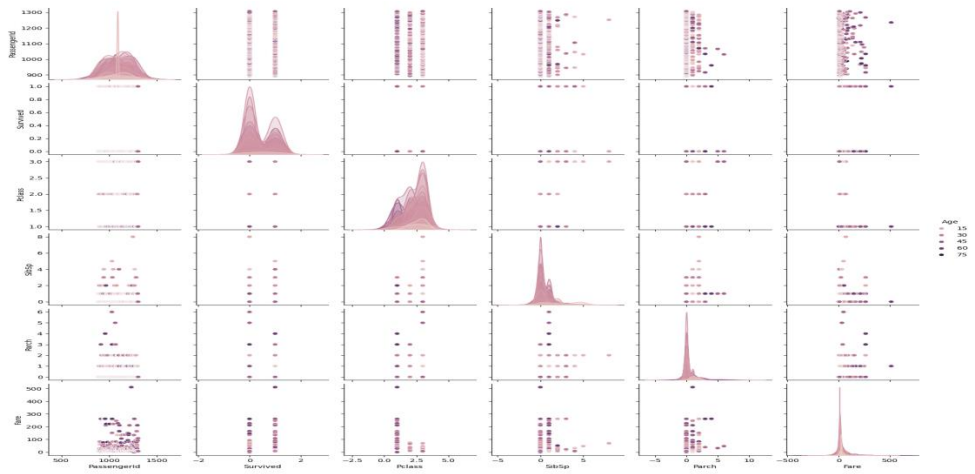
```
plt.scatter(x,y)
plt.savefig('EX8.4.png')
plt.show()
```



EX.NO:8F: PAIR PLOT

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
file_path = r'Z:\Data_Science\5\Titanic.csv'
df = pd.read_csv(file_path)
sns.pairplot(df,hue='Age')
plt.savefig('EX85.png')
plt.show()
```



EX.NO:8G: CORREALTION and HEAT MAP

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('Z:/DATA_SCIENCE/titanic.csv')
#plt.hist(df["Fare"])
sns.distplot(df["Fare"])
#box=plt.boxplot(df['Fare'])
df.corr()
```

```
dataplot=sns.heatmap(df.corr(),cmap="YlGnBu",annot=True)
plt.savefig('EX90')
plt.show()
```