PRACTICAL-1

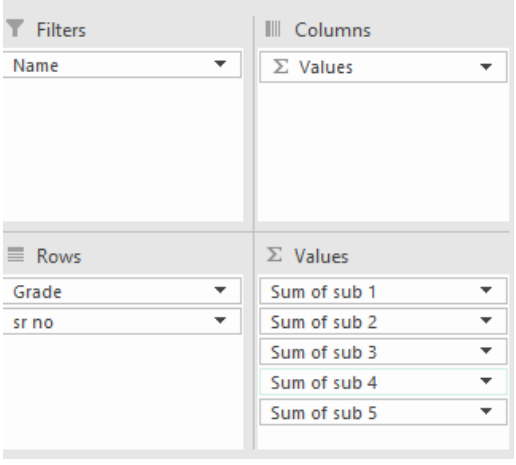
Aim:- Use MS-Excel to create pivot table & apply statistical measures to it..

Data:

sr no	Name	sub 1	sub 2	sub 3	sub 4	sub 5	Total	Avg	Percanta	Grade
1	hemil	40	37	43	59	50	229	45.8	91.6	O
2	raj	42	50	40	55	41	228	45.6	91.2	O
3	jenil	30	43	46	41	39	199	39.8	79.6	A
4	jay	40	29	49	36	28	182	36.4	72.8	A
5	romit	36	38	43	28	39	184	36.8	73.6	A
6	dhey	42	41	39	34	35	191	38.2	76.4	A
7	sahil	41	39	35	49	36	200	40	80	A+
8	mohan	36	28	36	48	37	185	37	74	A
9	radhika	28	39	37	40	38	182	36.4	72.8	A
10	utsav	34	35	35	0	0	104	20.8	41.6	Pass
11	milan	49	36	29	48	49	211	42.2	84.4	A+
12	jaynesh	48	37	49	25	48	207	41.4	82.8	A+
13	yash	40	38	40	29	42	189	37.8	75.6	A
14	krish	35	29	37	37	43	181	36.2	72.4	A
15	josh	48	49	42	40	42	221	44.2	88.4	A+
16	moksh	25	48	43	46	41	203	40.6	81.2	A+
17	henil	29	42	29	49	39	188	37.6	75.2	A
18	urmil	37	43	38	43	28	189	37.8	75.6	A
19	gaga	42	46	42	37	43	210	42	84	A+
20	jisas	43	46	41	42	46	218	43.6	87.2	A+
21	deep	29	49	39	43	46	206	41.2	82.4	A+
22	madhuran	38	43	28	29	49	187	37.4	74.8	A
23	nevil	49	42	39	38	43	211	42.2	84.4	A+
24	taksh	49	41	35	41	39	205	41	82	A+
25	deepak	47	39	36	39	35	196	39.2	78.4	A
26	shubam	25	25	37	28	25	140	28	56	B
27	krisha	28	28	38	28	17	139	27.8	55.6	B
28	parth	26	48	29	35	38	176	35.2	70.4	A
29	dhreej	27	50	49	36	29	191	38.2	76.4	A
30	tejal	29	42	38	37	49	195	39	78	A

Pivot table:

Row Labels	Sum of sub 1	Sum of sub 2	Sum of sub 3	Sum of sub 4	Sum of sub 5
A	520	592	574	561	568
A+	423	433	394	409	433
B	53	53	75	56	42
O	82	87	83	114	91
Pass	34	35	35	0	0
Grand Total	1112	1200	1161	1140	1134



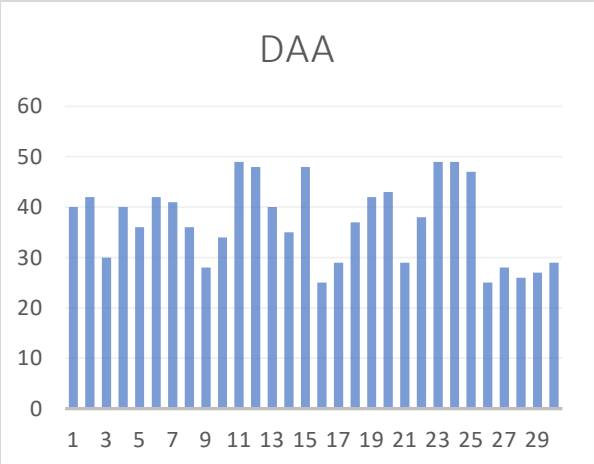
PRACTICAL-2

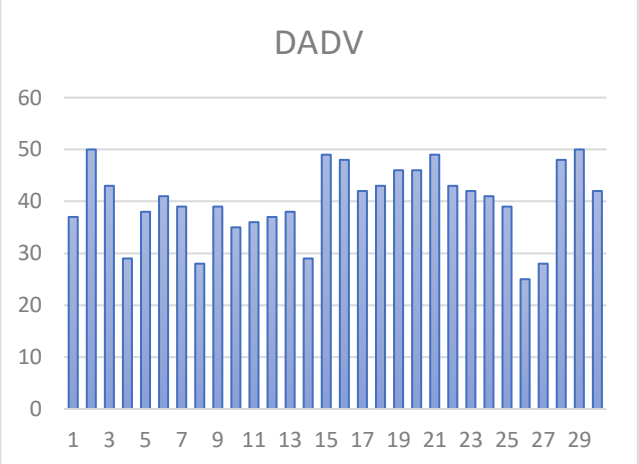
AIM: Use the Table Created in Above Practical to Generate Different Charts.

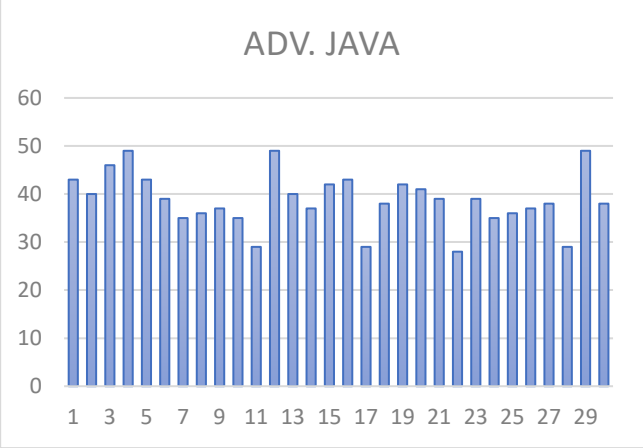
Data:

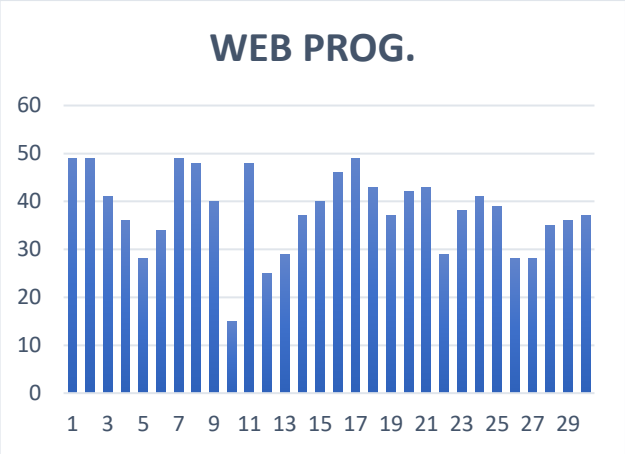
sr no	Name	DAA	DADV	ADV. JAV	WEB PRO	RES	Total	Avg	Percanta	Grade
1	hemil	40	37	43	49	50	219	43.8	87.6	A+
2	raj	42	50	40	49	41	222	44.4	88.8	A+
3	jenil	30	43	46	41	39	199	39.8	79.6	A
4	jay	40	29	49	36	28	182	36.4	72.8	A
5	romit	36	38	43	28	39	184	36.8	73.6	A
6	dhey	42	41	39	34	35	191	38.2	76.4	A
7	sahil	41	39	35	49	36	200	40	80	A+
8	mohan	36	28	36	48	37	185	37	74	A
9	radhika	28	39	37	40	38	182	36.4	72.8	A
10	utsav	34	35	35	15	45	164	32.8	65.6	B+
11	milan	49	36	29	48	49	211	42.2	84.4	A+
12	jaynesh	48	37	49	25	48	207	41.4	82.8	A+
13	yash	40	38	40	29	42	189	37.8	75.6	A
14	krish	35	29	37	37	43	181	36.2	72.4	A
15	josh	48	49	42	40	42	221	44.2	88.4	A+
16	moksh	25	48	43	46	41	203	40.6	81.2	A+
17	henil	29	42	29	49	39	188	37.6	75.2	A
18	urmil	37	43	38	43	28	189	37.8	75.6	A
19	gaga	42	46	42	37	43	210	42	84	A+
20	jisas	43	46	41	42	46	218	43.6	87.2	A+
21	deep	29	49	39	43	46	206	41.2	82.4	A+
22	madhuran	38	43	28	29	49	187	37.4	74.8	A
23	nevil	49	42	39	38	43	211	42.2	84.4	A+
24	taksh	49	41	35	41	39	205	41	82	A+
25	deepak	47	39	36	39	35	196	39.2	78.4	A
26	shubam	25	25	37	28	25	140	28	56	B
27	krisha	28	28	38	28	17	139	27.8	55.6	B
28	parth	26	48	29	35	38	176	35.2	70.4	A
29	dhreej	27	50	49	36	29	191	38.2	76.4	A
30	tejal	29	42	38	37	49	195	39	78	A

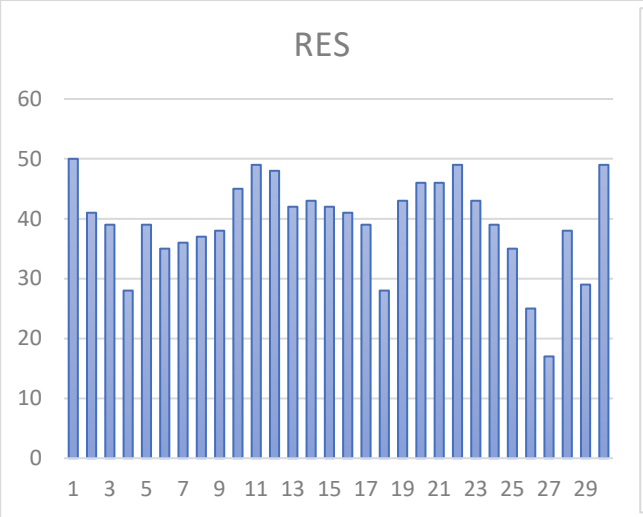
Output:

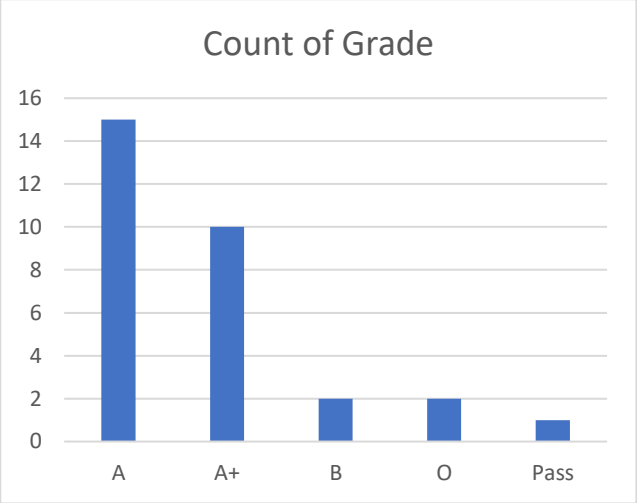


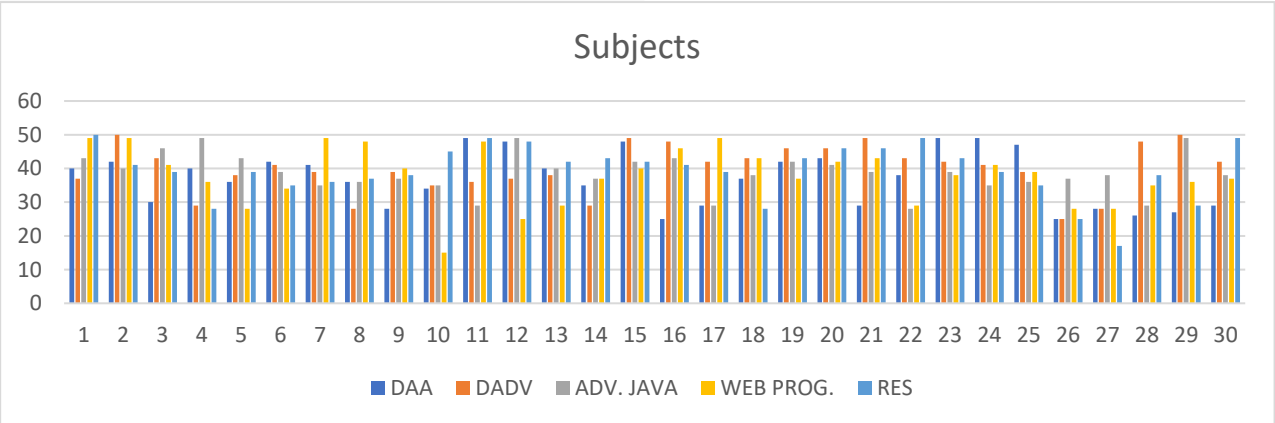


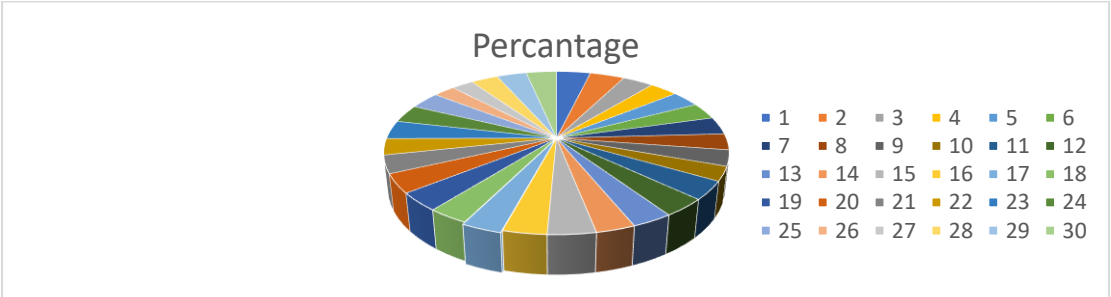

ADV. JAVA
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29


WEB PROG.
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29


RES
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29


Count of Grade
16
14
12
10
8
6
4
2
0
A
A+
B
O
Pass


Subjects
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
DAA
DADV
ADV. JAVA
WEB PROG.
RES


Percantage
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30

PRACTICAL-4

AIM: Use python libraries to generate chart from data stored in Excel.

Code:

```
import pandas as pd
from matplotlib import pyplot as plt
raw_data="/content/sample_data/california_housing_train.csv"
df = pd.read_csv(raw_data)
df.describe()
```

Output:

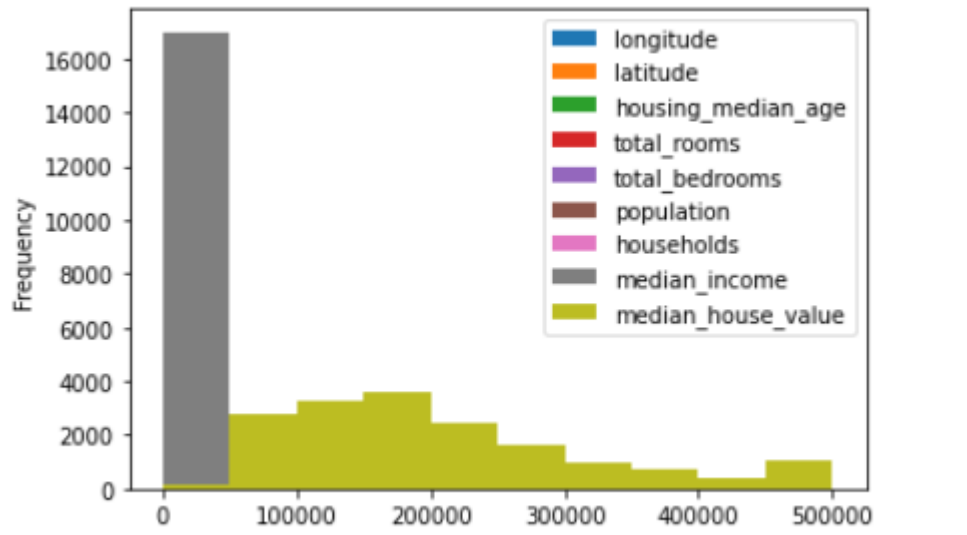
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000
mean	-119.562108	35.625225	28.589353	2643.664412	539.410824	1429.573941	501.221941	3.883578	207300.912353
std	2.005166	2.137340	12.586937	2179.947071	421.499452	1147.852959	384.520841	1.908157	115983.764387
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.790000	33.930000	18.000000	1462.000000	297.000000	790.000000	282.000000	2.566375	119400.000000
50%	-118.490000	34.250000	29.000000	2127.000000	434.000000	1167.000000	409.000000	3.544600	180400.000000
75%	-118.000000	37.720000	37.000000	3151.250000	648.250000	1721.000000	605.250000	4.767000	265000.000000
max	-114.310000	41.950000	52.000000	37937.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

Code:

```
df.plot.hist()
```

Output:

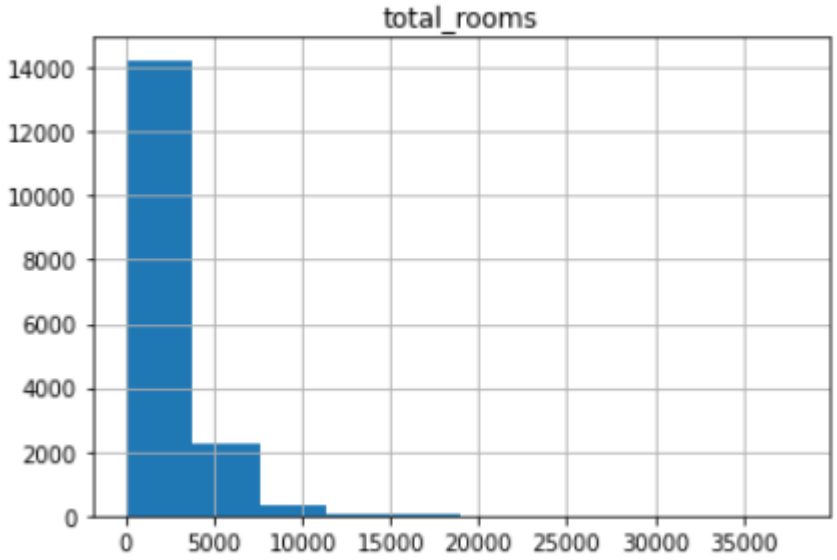
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb9e2956810>
```



Code:

```
df.hist(column='total_rooms');
```

Output:

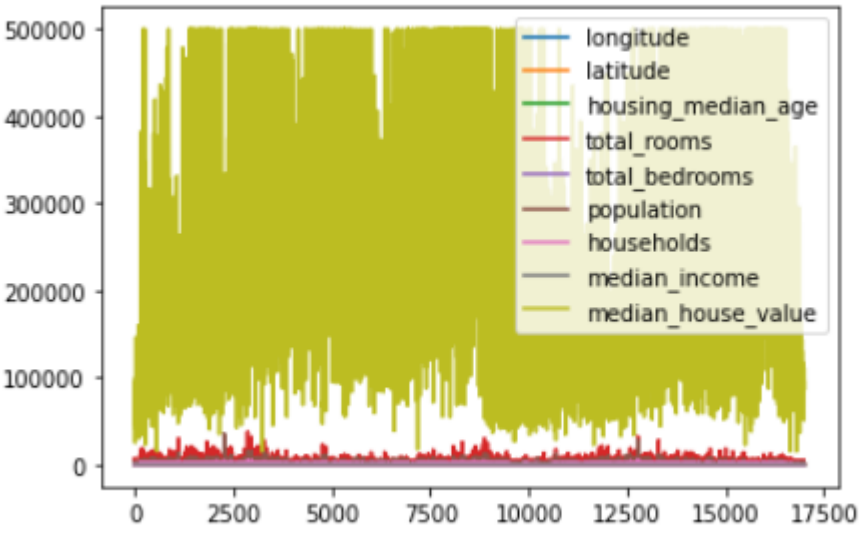


Code:

```
df.plot.line()
```

Output:

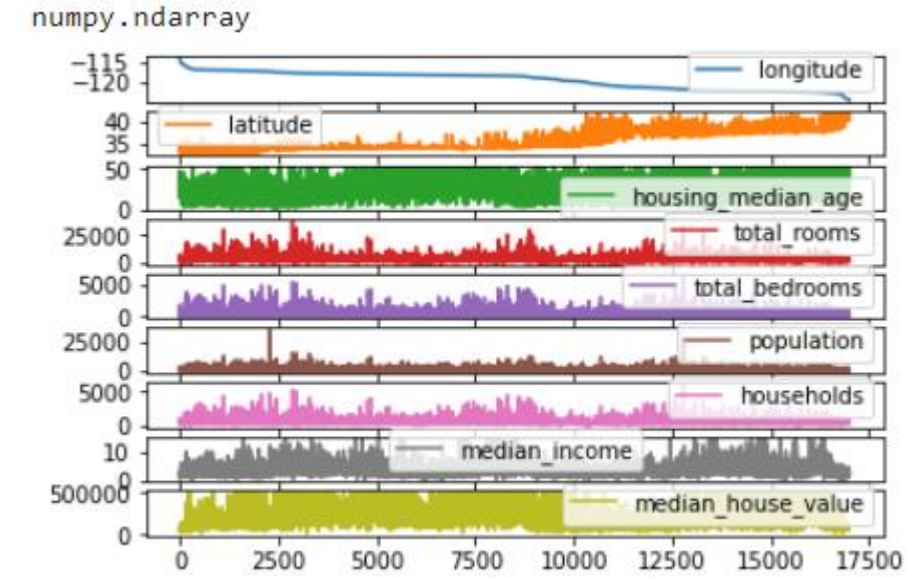
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb9e2af30d0>
```



Code:

```
axes = df.plot.line(subplots=True)
type(axes)
```

Output:


numpy.ndarray
longitude
latitude
housing_median_age
total_rooms
total_bedrooms
population
households
median_income
median_house_value
−115
−120
40
35
50
0
25000
0
5000
0
25000
0
5000
0
10
0
500000
0
0
2500
5000
7500
10000
12500
15000
17500

PRACTICAL-5

AIM: Perform Multiple Linear Regression on data.

Code:

```
# Import Libary
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# read data
df=pd.read_csv('sample_data/headbrain.csv')
df.head()
```

Output:

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

Code:

```
# Declare dependent variable(Y) and independent variable(X)
X=df['Head Size(cm^3)'].values
Y = df['Brain Weight(grams)'].values
np.corrcoef(X, Y)
```

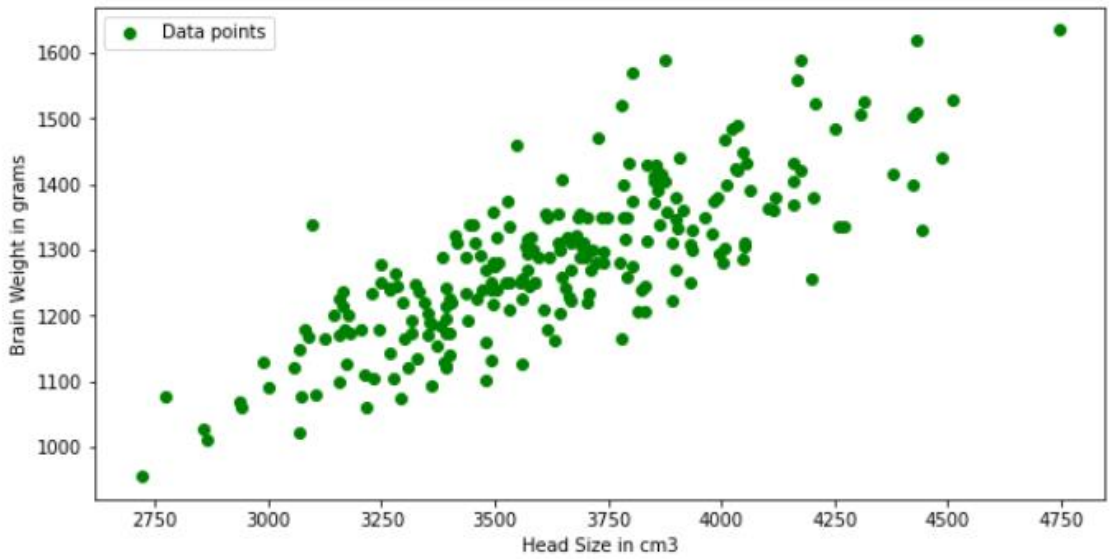
Output:

```
array([[1. , 0.79956971], [0.79956971, 1. ]])
```

Code:

```
# Plot the Input Data
plt.scatter(X, Y, c='green', label='Data points')
plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()
```

Output:



Code:

```
# Mean X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)
# Total number of values
n = len(X)
# Using the formula to calculate theta1 and theta2
numer = 0
denom = 0
for i in range(n):
    numer += (X[i] - mean_x) * (Y[i] - mean_y)
    denom += (X[i] - mean_x) ** 2
b1 = numer / denom
b0 = mean_y - (b1 * mean_x)
# Printing coefficients
print("coefficients for regression",b1, b0)
```

Output:

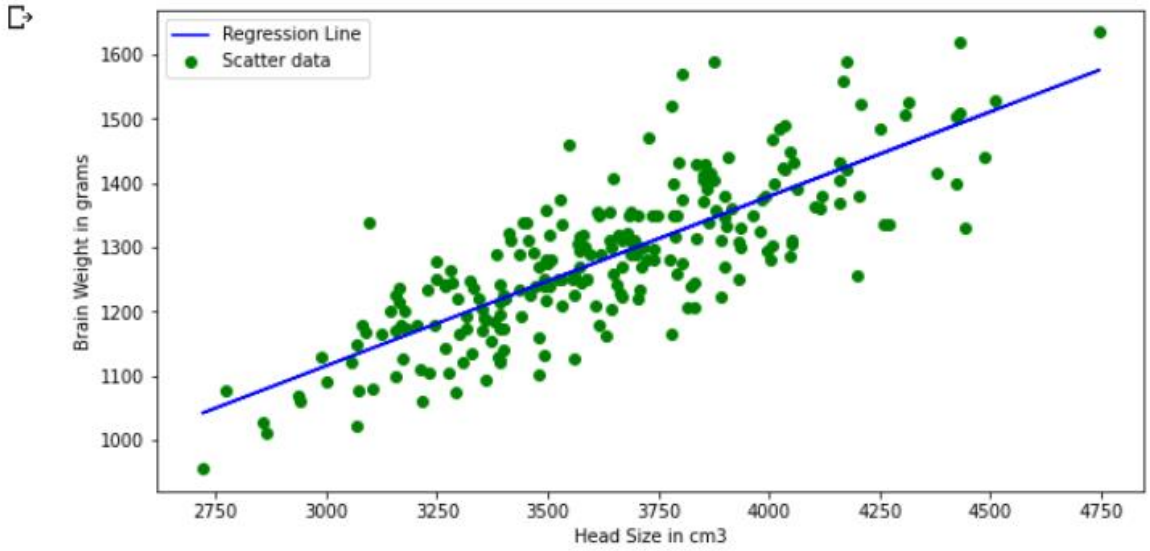
```
coefficients for regression 0.26342933948939945 325.57342104944223
```

Code:

```
# Plotting Values and Regression Line
%matplotlib inline
plt.rcParams['figure.figsize'] = (10.0, 5.0)
max_x = np.max(X) + 100
min_x = np.min(X) - 100
y = b0 + b1 * X
# Ploting Line
plt.plot(X, y, color='blue', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='green', label='Scatter data')
```

```
plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()
```

Output:



PRACTICAL-5

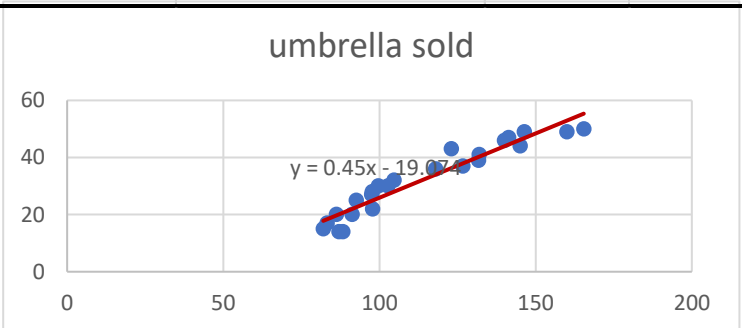
AIM: Perform the Logistic Regression on a dataset and Interpret the regression table.

Output:

month	rain fall	umbrella sold
jan	82	15
feb	92.5	25
mar	83.2	17
apr	97.7	28
may	131.9	41
jun	141.3	47
jul	165.4	50
aug	140	46
sep	126.7	37
oct	97.8	22
nov	86.2	20
dec	99.6	30
jan	87	14
feb	97.5	27
mar	88.2	14
apr	102.7	30
may	123	43
jun	146.3	49
jul	160	49
aug	145	44
sep	131.7	39
oct	118	36
nov	91.2	20
dec	104.6	32

RESIDUAL OUTPUT

Observation	Predicted umbrella sold	Residuals	Standard Residuals
1	17.82599924	-2.825999237	-0.806807846
2	22.5510131	2.448986904	0.699172817
3	18.36600082	-1.366000821	-0.389986015
4	24.89101996	3.10898004	0.887597369
5	40.2810651	0.7189349	0.205252114
6	44.51107751	2.488922493	0.710574217
7	55.35610932	-5.356109317	-1.529140901
8	43.92607579	2.073924208	0.5920944
9	37.94105824	-0.941058237	-0.268667153
10	24.93602009	-2.936020092	-0.838218218
11	19.71600478	0.283995219	0.081079134
12	25.74602247	4.253977532	1.214488101
13	20.07600584	-6.076005837	-1.734667552
14	24.8010197	2.198980304	0.627797254
15	20.61600742	-6.616007421	-1.88883515
16	27.14102656	2.858973441	0.816221806
17	36.27605335	6.723946647	1.919651229
18	46.76108411	2.238915893	0.639198654
19	52.92610219	-3.926102189	-1.120881424
20	46.17608239	-2.176082391	-0.621260021
21	40.19106484	-1.191064836	-0.340042715
22	34.02604675	1.973953246	0.563553219
23	21.96601138	-1.966011381	-0.561285858
24	27.99602907	4.003970933	1.143112539



SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.957666798
R Square	0.917125697
Adjusted R Square	0.913358683
Standard Error	3.58141382
Observations	24

rainfall/m	0.45000132
x	82
intercept/c	-19.07410899
y=mx+c	17.82599924

ANOVA

	df	SS	MS	F	Significance F
Regression	1	3122.774784	3122.774784	243.462262	2.21604E-13
Residual	22	282.1835489	12.82652495		
Total	23	3404.958333			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	-19.07410899	3.372182168	-5.656310378	1.0929E-05	-26.06758677	-12.08063122	-26.06758677	-12.08063122
rain fall	0.45000132	0.02884018	15.6032773	2.216E-13	0.390190448	0.509812192	0.390190448	0.509812192

PRACTICAL-8

AIM: Use a dataset & apply K means clustering to get insights from data.

Code:

```
from sklearn.cluster
import KMeans import pandas as pd
from sklearn.preprocessing
import MinMaxScaler from matplotlib
import pyplot as plt
%matplotlib inline
```

Code:

```
df = pd.read_csv("income.csv")
df.head()
```

Output:

	Name	Age	Income($)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

Code:

```
km.cluster_centers_
```

Output:

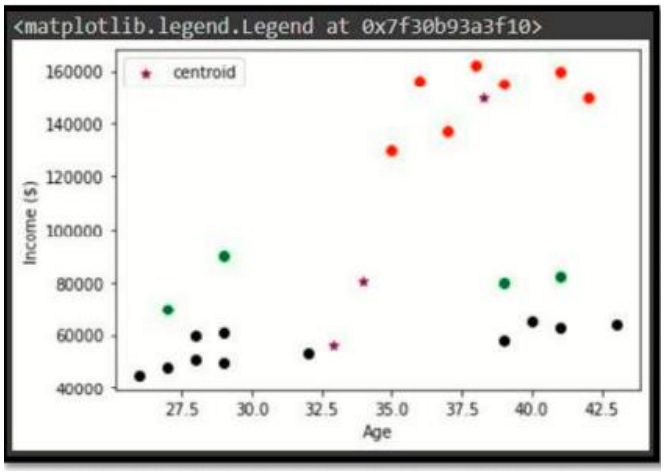
```
array([[3.40000000e+01, 8.05000000e+04],[3.82857143e+01, 1.50000000e+05],
[3.29090909e+01, 5.61363636e+04]])
```

Code:

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
```

```
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='c
entroid') plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()
```

Output:




```
scaler = MinMaxScaler()
scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])
scaler.fit(df[['Age']]) df['Age'] = scaler.transform(df[['Age']])
plt.scatter(df.Age,df['Income($)'])
```

Output:

```
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2],
dtype=int32)
```

Code:

```
df['cluster']=y_predicted
df.head()
```

Output:

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	0
3	Ismail	0.117647	0.128205	0
4	Kory	0.941176	0.897436	1

Code:

```
km.cluster_centers_
```

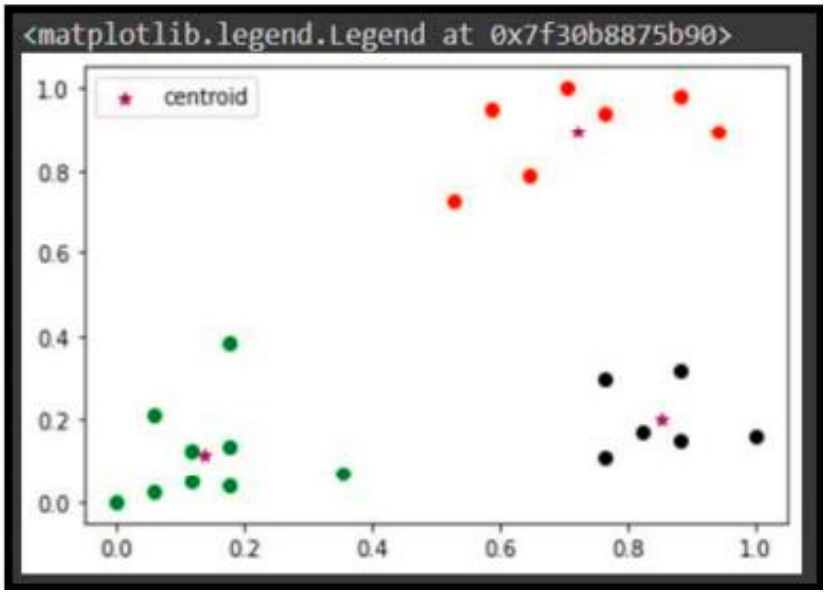
Output:

array([[0.1372549 , 0.11633428],[0.72268908, 0.8974359], [0.85294118, 0.2022792]])

Code:

Code: df1 = df[df.cluster==0] df2 = df[df.cluster==1] df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='
centroid') plt.legend()

Output:



PRACTICAL-9

AIM: Study about the tools like Orange, Tableau, Weka etc. tool for data Visualization.

Theory:

Orange Tool: Orange supports a flexible domain for developers, analysts, and data mining specialists. Python, a new generation scripting language and programming environment, where our data mining scripts may be easy but powerful. Orange employs a component-based approach for fast prototyping. We can implement our analysis technique simply like putting the LEGO bricks, or even utilize an existing algorithm. What are Orange components for scripting Orange widgets for visual programming?. Widgets utilize a specially designed communication mechanism for passing objects like classifiers, regressors, attribute lists, and data sets permitting to build easily rather complex data mining schemes that use modern approaches and techniques.

Orange core objects and Python modules incorporate numerous data mining tasks that are far from data preprocessing for evaluation and modeling. The operating principle of Orange is cover techniques and perspective in data mining and machine learning. For example, Orange's top-down induction of decision tree is a technique build of numerous components of which anyone can be prototyped in python and used in place of the original one. Orange widgets are not simply graphical objects that give a graphical interface for a specific strategy in Orange, but it includes an adaptable signaling mechanism that is for communication and exchange of objects like data sets, classification models, learners, objects that store the results of the assessment. All these ideas are significant and together recognize Orange from other data mining structures.

Widgets:

Orange widgets give us a graphical user interface to orange's data mining and machine learning techniques. They incorporate widgets for data entry and pre processing, classification, regression, association rules and clustering a set of widgets for model assessment and visualization of assessment results, and widgets for exporting the models into PMML.

Scripting:

If we want to access Orange objects, then we need to write our components and design our test schemes and machine learning applications through the script. Orange interfaces to Python, a model simple to use a scripting language with clear and powerful syntax and a broad set of additional libraries. Same as any scripting language, Python can be used to test a few ideas mutually or to develop more detailed scripts and programs.

Weka Tool:

Weka is one of the very popular open source data mining tools developed at the University of Waikato in New Zealand in 1992. It is a Java based tool and can be used to implement various machine learning and data mining algorithms written in Java. The simplicity of using Weka has made it a landmark for machine learning and data mining implementation. Weka supports reading of files from several different databases and also allows importing the data from the internet, from web pages or from a remotely located SQL database server by entering the URL of resource. Among all the available data mining tools, Weka is the most commonly used of all due to its fast performance and support for major classification and clustering algorithm. Weka can be easily downloaded and deployed. Weka provides both, a GUI and CLI for performing data mining and does a good job of providing support for all the data mining tasks. Weka supports a variety of data formats like CSV (Commaseparated Value), ARFF and Binary. Weka focuses more on textual representation of the data rather than visualization although it does provide support to display some visualization but those are very generic. Also, Weka does not provide visual representation of results of processing in an effective and understanding manner like Rapid Miner. Weka performs accurately when the size of the data set is not large. If the size is large, then Weka does experience some performance issues. Weka provides support for filtering out data or attributes. Weka supports the following three graphical user interfaces

1.The Explorer:

It is the most commonly used graphical user interface in Weka to implement data mining algorithms8. It supports exploratory data analysis to perform preprocessing, attribute selection, learning and visualization. This interface consists of different tabs to access various components for performing data mining. The different tabs are-

A) **Preprocessing** Using this tab, we can load input data files and perform preprocessing on this data using filters.
B) **Classify** This tab is used to implement different classification and regression algorithms. We can do this by selecting a particular classifier from this tab. For example, the K-NN or Naïve Bayesian algorithm can be implemented by using this tab.
C) **Associate** This tab is used to find out all association rules between different attributes of the data and which can be used for further mining. For example, Association rule mining, etc.

D) **Cluster** Using this tab, we can select a particular clustering algorithm to implement for our data set. Clustering algorithms like K-means can be implemented using this tab.
E) **Select attributes** This tab is used to select particular attributes from the data set useful for implementing the algorithm.
F) **Visualize** This tab is used to visualize the data whenever available or supported by a particular algorithm in the form of scatter plot matrix.

2. **The Experimenter**

This user interface provides experimental environment for testing and evaluating machine learning algorithms.

3. **The Knowledge**

Flow Knowledge flow is basically a component based interface similar to explorer. This interface is used for new process evaluations.

Tableau Tool:

Tableau is a powerful data visualization tool used in business intelligence and data analysis. Tableau Software was invented by Chris Stolte, Christian Chabot and Pat Hanrahan in January, 2003. The visualization provided by Tableau has completely enhanced the ability to gain more knowledge about the data we are working on and can be used to provide more accurate predictions. “The product queries relational databases, cubes, cloud databases, and spreadsheets and then generates a number of graph.Types that can be combined into dashboards which can be securely shared over a computer network or the internet”. Unlike Rapid Miner and Weka, Tableau does not implement data mining algorithms provides visualizations of the data. For this, Tableau provides integration with another popular statistical analysis tool R9, to provide support for data mining. “Tableau offers five main products namely Tableau Desktop, Tableau Server, Tableau Online,Tableau Reader and Tableau Public. Tableau Public and Tableau Reader are available freely, whereas Tableau Server and Tableau Desktop come with a free trial period afterwards which the user has to pay”. Tableau has made it possible to explore and present the data in a much simpler and beautiful manner. Working on projects using Tableau is less time consuming and easy to handle. Tableau uses a feature called Dashboard which is a collection of worksheets which can be easily imported from anywhere.

PRACTICAL-10

AIM: Given a case study: Interactive Data Analytics with Power BI.

Theory:

The Client:

The client is a data analytics specialist organization that uses artificial intelligence and data analytics to study professional services industry problems. The company is headquartered in London, England.

Client's Requirement:

The client was looking for a reliable and cost-effective service provider who could help them with the implementation of a Power BI solution for their business.

The Challenges:

The client had been facing several challenges before the implementation of the project. Some of the major challenges faced were:

1. The client wanted to create one dashboard that would give them all the information about their business that they needed in one, easy-to-access place.
2. They wanted to add some additional pages and visualizations to the report, as well as making it more concise, clear, and informative.

The Solution:

Flatworld Power BI development team meticulously studied the client's requirements and created the perfect reporting tool to accommodate all of them. To go above and beyond, Flatworld proposed multiple suggestions to improve the reports including shuffling reports around, adding more visual experiences, and other structural changes to make this dashboard both eye-catching and functional. Flatworld team worked with Power BI to enhance reports. The relationships in tables were made interactive, charts and graphs were redesigned, and the entire UI/UX was improved for an enhanced user experience.

The Results:

We estimated it would take 10 days to complete the first milestone of the project, but it ended up taking us only 7. The client was happy with the delivery and offered more Power BI work, all of which we will happily fulfill. The client was not only delighted to see the quality of the dashboard we created but also the cost-effectiveness of the entire project.