ARUNAI ENGINEERING COLLEGE

(Affiliated to Anna University)
Velu Nagar,Thiruvannamalai-606 603
www.arunai.org

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BACHELOR OF TECHNOLOGY

2023-2024

THIRD SEMESTER

AD3301- DATA EXPLORATION AND VISUALIZATION LAB

ARUNAI ENGINEERING COLLEGE

THIRUVANAMALAI -606 603

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

CERTIFIED THAT THIS IS A BONAFIDE RECORD OF WORK DONE BY

NAME :

UNIVERSITY REG.NO :

SEMESTER :

BRANCH :

YEAR :

STAFF-IN-CHARGE **HEAD OF THE DEPARTMENT**

SUBMITTED FOR THE _____

PRACTICAL EXAMINATION HELD ON _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

Ex.No		Name Of The Experiment	Page.No	Staff Sign

Ex.No:1 **Installation of Data Analysis And Visualization Tool : Python**
DATE

Packages that we will need

Python3 and the following Python libraries / packages are needed for data exploration and visualization:

- jupyter
- jupyterlab
- numpy
- scipy
- pandas
- matplotlib
- seaborn

How to install Python and the packages

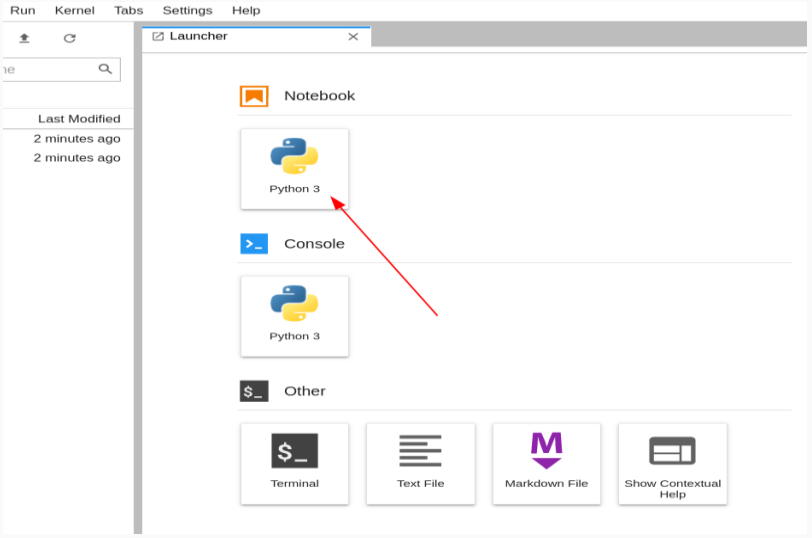
Install Anaconda which will give you a **Python3** environment and all the above required

Packages . After you have installed Anaconda , please verify the installation.
$ conda install -cconda - forgealtairvega_datasets

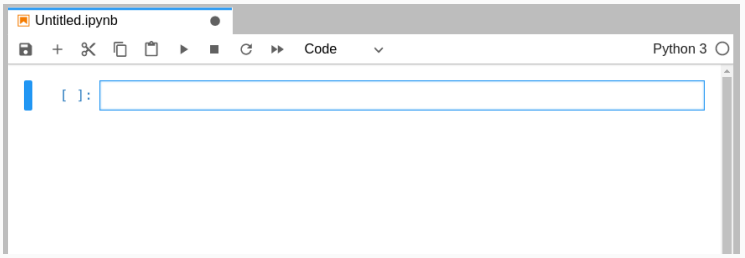
Howtoverifyyourinstallation

1. Open the AnacondaNavigator.
2. Findthe**JupyterLab**tileand“launch”it.

IfyouareonLinuxormacOS,youcanopenJupyterLabfromthe terminalbytypingjupyter-lab.

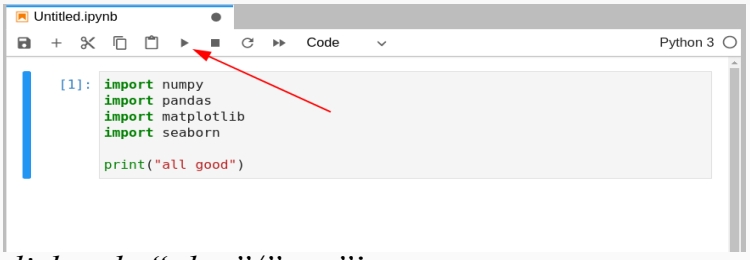


OnceyouclickedthePython3tileitshouldlook like this:

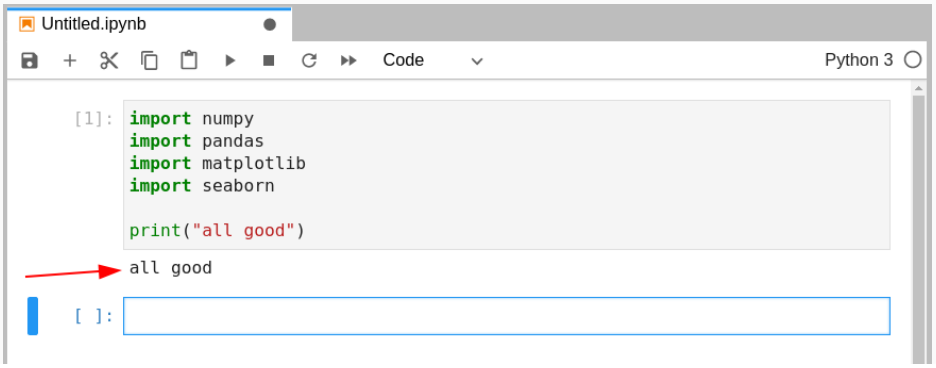


Type thefollowing:

import numpy
import scipy
import pandas
importmatplotli
b import
seaborn
print("allgood")



clickonthe"play"/"run"icon.



Result:

Thus the python tool were installed and verified successfully.

Ex.No:2 **Exploratory Data Analysis**
DATE

Aim:

To perform exploratory data analysis(EDA) on with datasets.

Procedure:

1. Import the dataset
2. View the head of the data
3. View the basic information of data and description of data
4. Find the unique value of data and verify the duplication of data
5. Plot a graph for unique value of dataset
6. Verify the presence of null value and replace the null value
7. Visualize the needed data

Program:

```
#Load the required
libraries
importpandasaspdimp
ort numpy as np
importseabornassns

#Loadthedata
df=pd.read_csv('titanic.csv')

#Viewthedata
df.head()


df.info()

df.describe()



#Findtheduplicate

s

df.duplicated().sum(

)


#unique values
df['Pclass'].unique()
df['Survived'].unique()
df['Sex'].unique()

#Plot the unique values

sns.countplot(df['Pclass']).unique()

#Findtheduplicates
```

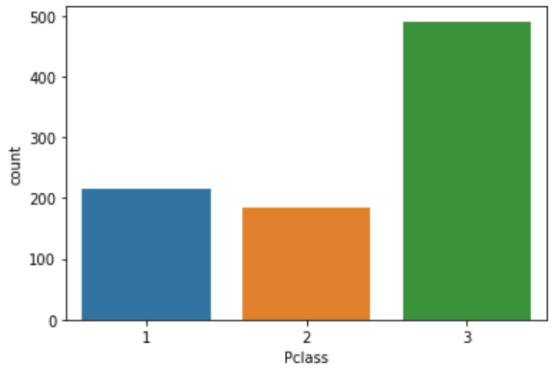
```
df.duplicated().sum()

#unique values
df['Pclass'].unique()
df['Survived'].unique()
df['Sex'].unique()

#Plot the unique values

sns.countplot(df['Pclass']).unique()



#Findnullvalues
df.isnull().sum()

#Replace null values

df.replace(np.nan,'0',inplace = True)

#Checkthechangesnow
df.isnull().sum()

#Filter data

df[df['Pclass']==1].head(

)

#Boxplotdf[['Fare']]

.boxplot()
```

OUTPUT:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   PassengerId     891 non-null    int64
 1   Survived        891 non-null    int64
 2   Pclass          891 non-null    int64
 3   Name            891 non-null    object
 4   Sex             891 non-null    object
 5   Age             714 non-null    float64
 6   SibSp           891 non-null    int64
 7   Parch           891 non-null    int64
 8   Ticket          891 non-null    object
 9   Fare            891 non-null    float64
 10  Cabin           204 non-null    object
 11  Embarked        889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

	PassengerId	Survived	Pclass	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	8.000000	6.000000	512.329200



array([3,1,2],dtype=int64)
array([0, 1], dtype=int64)
array(['male','female'],dtype=object)

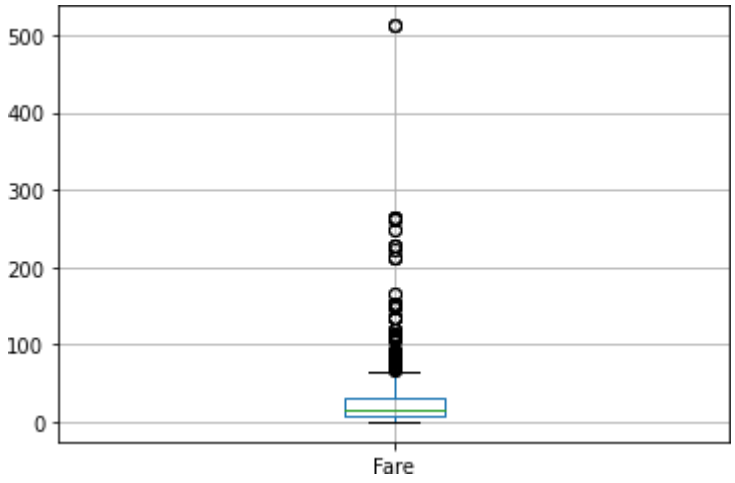
0

PassengerId 0
Survived 0
Pclass 0
Name 0
Sex 0
Age 177
SibSp 0
Parch 0
Ticket 0
Fare 0
Cabin 687
Embarked 2

dtype: int64

PassengerId0
Survived 0
Pclass 0
Name 0
Sex 0
Age 0
SibSp 0
Parch 0
Ticket 0
Fare 0
Cabin 0
Embarked 0

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5000	A6	S


500
400
300
200
100
0
Fare

Result:

Thus the program to perform exploratory data analysis (EDA) on with datasets was executed.

Ex.No:3.1 **NumpyArrays**
DATE

Aim:

To write a program to work with Numpy arrays.

Procedure:

1. Create array using numpy
2. Access the element in the array
3. Retrieve element using slice operation
4. Compute calculation in the array

Program:

```
import numpy as np

a = np.array([1, 2, 3]) # Create a rank 1 array
print(type(a))
                    #Prints"<class'numpy.ndarra
y'>"print(a.shape)         # Prints "(3,)"
print(a[0],a[1],a[2])#Prints"123"
a[0]=5              #Changeanelementofthearray
print(a)            #Prints"[5,2,3]"

b = np.array([[1,2,3],[4,5,6]])  # Create a
rank 2 array print(b.shape)      # Prints "(2,
3)"
print(b[0,0],b[0,1],b[1,0])
#Createthefollowingrank2arraywithshape(3,4
) # [[ 1234]
#[5678]
#[9101112]]
a=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])

#Use slicingto pull out thesubarray consisting
ofthefirst2 rows # and columns 1 and 2; b is the
following array of shape (2, 2):
#[[23]
#[67]]
b=a[:2, 1:3]

#Asliceofanarrayisaviewintothesamedata,somodifyin
git # will modify the original array.
print(a[0,1])#Prints"2"
b[0, 0] = 77 # b[0, 0] is the same piece of data
as a[0, 1] print(a[0, 1])
a=np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])

print(a)#prints"array([[1,2,3],
          #            [4,5,6],
          #            [7,8,9],
        #            [10,11, 12]])"

#Createanarrayofindic
es b = np.array([0, 2,
0, 1])
```

```
#Selectoneelementfromeachrowofausingtheindicesinb
print(a[np.arange(4), b])# Prints "[ 167 11]"

# Mutate one element from each row of a using the
indices in ba[np.arange(4), b] += 10

print(a)
x = np.array([1, 2]) # Let numpy choose
the datatype print(x.dtype)       # Prints
"int64"
x=np.array([[1,2],[3,4]],dtype=np.float64)
y=np.array([[5,6],[7,8]],dtype=np.float64)

# Elementwise
sum;bothproducethearray # [[ 6.08.0]
#[10.012.0]]
print(x + y)
print(np.add(x,y))


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

print(np.sum(x)) # Compute sum ofallelements;
prints"10" print(np.sum(x,axis=0)) #Computesum of each
column; prints"[4 6]"
print(np.sum(x,axis=1))#Computesumofeachrow;prints"[
37]"
```

Output:

```
<class'numpy.nparra
y>
(3,)
123
[523]
(2,3)
124
2
77
[[123]
 [456]
 [789]
 [101112]]
[16711]
[[1123]
 [4516]
 [1789]
 [102112]]
int32
[[6.8.]
 [10.12.]]
[[6.8.]
 [10.12.]]
10
[46]
[37]
```

Result:

Thus, the program is executed successfully

Ex.No:3.2 **Pandas Data Frames**
DATE

Aim:

To write a program for working with pandas data frames.

Procedure:

1. Import pandas library
2. Construct a pandas data frame
3. Modify ,drop columns in data frame
4. Calculate median in the data frame

Program:

```
import pandas as pd
data=pd.DataFrame({"x1":["y","x","y","x","x","y"],#ConstructapandasDataFrame
                   "x2":range(16,22),
                   "x3":range(1,7),
                   "x4":["a","b","c","d","e","f
                   "], "x5":range(30, 24, -
                   1)})
print(data)

data_row=data[data.x2<20]              #Removeparticularrows
print(data_row)                        #PrintpandasDataFramesubset
data_col=data.drop("x1",axis=1)        #DropcertainvariablefromDataFrame
print(data_col)



data_col=data.drop("x1",axis=1)        #DropcertainvariablefromDataFrame
print(data_col)


data_med=data["x5"].median()           #Calculatemedian
print(data_med)
```

Output:

27.5

Table 1

	x1	x2	x3	x4	x5
0	y	16	1	a	30
1	x	17	2	b	29
2	y	18	3	c	28
3	x	19	4	d	27
4	x	20	5	e	26
5	y	21	6	f	25

Table 2

	x1	x2	x3	x4	x5
0	y	16	1	a	30
1	x	17	2	b	29
2	y	18	3	c	28
3	x	19	4	d	27

	x2	x3	x4	x5
0	16	1	a	30
1	17	2	b	29
2	18	3	c	28
3	19	4	d	27
4	20	5	e	26
5	21	6	f	25

Result:

Thus, the program to work with pandas dataframe was executed.

Ex.No:3.3 **Basic Plots Using Matplotlib**
DATE

Aim:

To write a program to visualize basic plots using Matplotlib.

Procedure:

1. Import matplotlib library
2. Define x , y axis
3. Label the axis
4. Visualize the data using lineplot

Program:

```
import matplotlib.pyplot s plt
import numpy as np
x=[20,25,37]
y=[25000,40000,600]
plt.plot(x,y)
plt.xlabel("Age")
plt.ylabel('salary')
plt.title('Salarybyage')
plt.show()
```

OUTPUT:


Salary by age
60000
55000
50000
45000
40000
35000
30000
25000
salary
20.0
22.5
25.0
27.5
30.0
32.5
35.0
37.5
Age

Result:

Thus, the program to plot the basic plots using Matplotlib was executed.

Ex.No:4 **Data Cleaning using R**

DATE

Aim:

To explore various variable and row filters, plot features in R for cleaning data and visualize it.

Procedure:

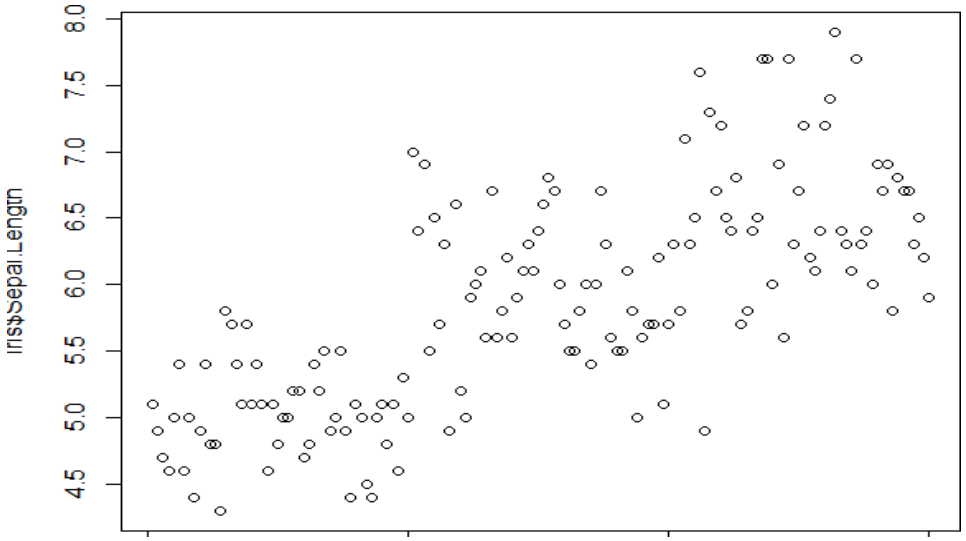
1. Import dplyr and ggplot2 library
2. Import iris dataset
3. Using dplyr select and filter functions rearrange the data
4. Using plotting visualize the selected data

Program:

```
> data(iris)
> head(iris,5)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> mean(iris$Sepal.Length)
[1] 5.843333
> library(dplyr)
> a<-select(iris,Sepal.Length)
> head(a,5)
  Sepal.Length
1          5.1
2          4.9
3          4.7
4          4.6
5          5.0
> c<-filter(iris,Sepal.Length>5)
> tail(c,5)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
114          6.7         3.0          5.2         2.3 virginica
115          6.3         2.5          5.0         1.9 virginica
116          6.5         3.0          5.2         2.0 virginica
117          6.2         3.4          5.4         2.3 virginica
118          5.9         3.0          5.1         1.8 virginica
> library(ggplot2)
```

plot(iris$Sepal.Length)

OUTPUT:



Result:

Thus, the program for cleaning and visualizing the data using R was executed.

Ex.No:5 **Time Series**
DATE

Aim:

To write a program to visualize time series analysis.

Procedure:

1. Import the temperature dataset
2. Import pandas and matplotlib library
3. Visualize the data using lineplot, histogram and boxplot

Program:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#reading the data set using read_csv
df =pd.read_csv("stock_data.csv",
                parse_dates=True,
               index_col="Date")

# displaying the first five rows of dataset
df.head()
```

```
from pandas import read_csv
from matplotlibimport pyplot
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0,
parse_dates=True,squeeze=True)print(series.head())
```

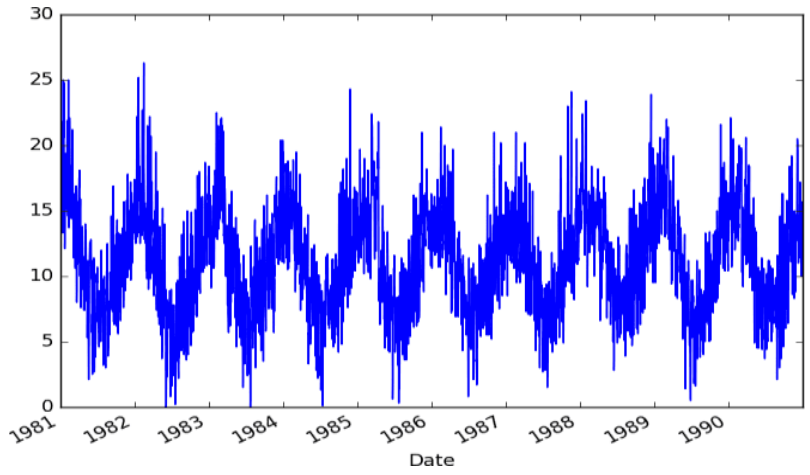
```
from pandas import read_csv
from matplotlibimport pyplot
series= read_csv('daily-minimum-temperatures.csv', header=0,
index_col=0,parse_dates=True, squeeze=True)
series.plot()pyplot.show()
```

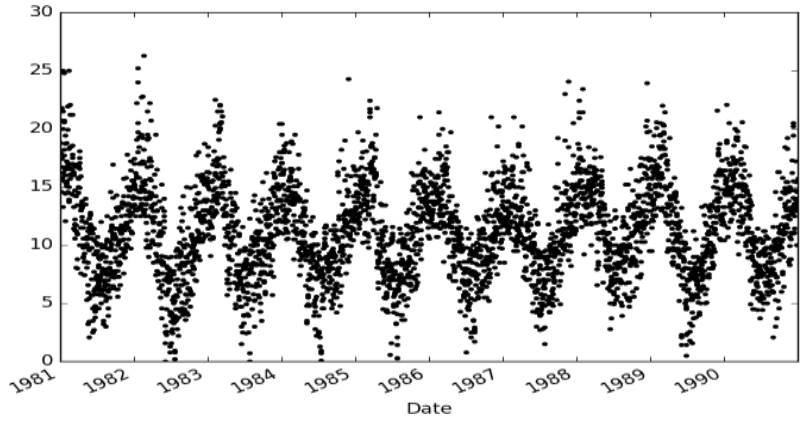
```
from pandas import read_csv
from matplotlibimport pyplot
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0,
parse_dates=True,squeeze=True)
series.plot(style='k.')
pyplot.show()
```

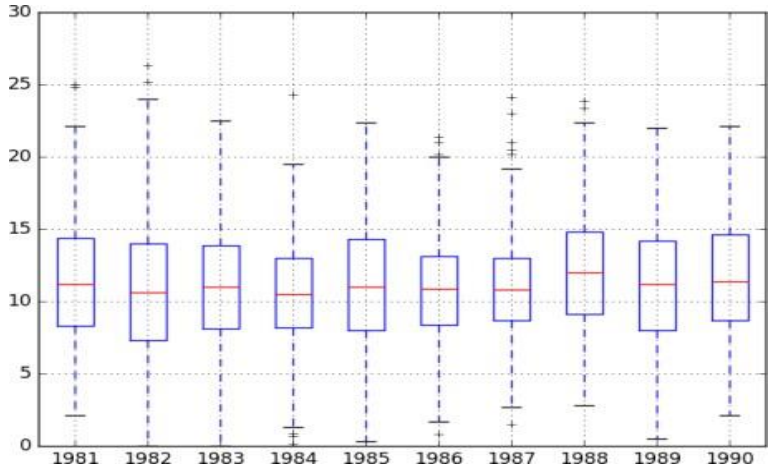
```
from pandas import read_csv
from matplotlibimport pyplot
series=read_csv('daily-minimum-
temperatures.csv',header=0,index_col=0,parse_dates=True,squeeze=True) series.hist()
pyplot.show()
```

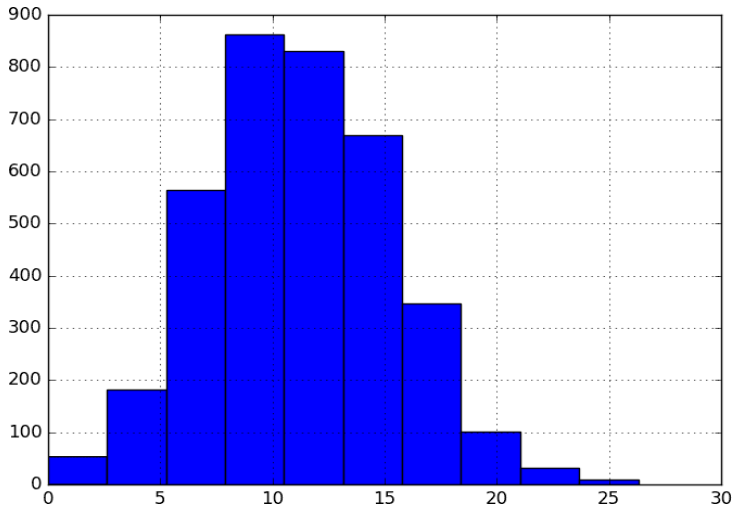
```
from pandas import read_csv
from pandas import DataFrame
from pandas import Grouper
from matplotlib import pyplot
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0,
parse_dates=True,squeeze=True)
groups=series.groupby(Grouper(freq='A'))yea
rs = DataFrame()
forname,groupingroups:
        years[name.year]=group.values
years.boxplot()
pyplot.show()
```

OUTPUT:


30
25
20
15
10
5
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
Date


30
25
20
15
10
5
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
Date


30
25
20
15
10
5
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990


900
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30

Result:

Thus, the program to visualize time series analysis was executed.

Ex.No:6 **InteractiveMapVisualization**
DATE

Aim:

To represent on a Map using various Map datasets with Mouse Rollover effect.

Procedure:

1. Import the library
2. Import map dataset
3. Specify the width , height , title form a use rollover
4. visualizethe map

Program:

```
pip install pyecharts
pip install echarts-countries-pypkg
pip install echarts-china-provinces-
pypkg pip install echarts-china-
cities-pypkg
pipinstallecharts-china-counties-
pypkg importpyecharts
print(pyecharts.version
) importpandas as pd
frompyecharts.charts
importMap
frompyechartsimportoptionsas
opts data =
pd.read_excel('GDP.xlsx')
province =
list(data["province"])
gdp=list(data["2019_gdp"])
list=
[list(z)forzinzip(province,gdp)]
c = (
   Map(init_opts=opts.InitOpts(width="1000px",height="600px"))#Initializemapsize
   .set_global_opts(
      title_opts=opts.TitleOpts(title="2019ProvincesinGDPDistributionunit:100mill
ionyuan"), #Configuration title
      visualmap_opts=opts.VisualMa
         pOpts(type_=
         "scatter"#Scatter type
      )
)
   .add("GDP",list,maptype="china")#takelistImported,maptypeisChinaMap
   .render("Map1.html")
)
```

OUTPUT:


新疆
黑龙江
吉林
辽宁
内蒙古
北京
天津
宁夏
山西
河北
山东
青海
甘肃
陕西
河南
安徽
江苏
上海
浙江
四川
湖北
西藏
重庆
湖南
江西
贵州
福建
云南
GDP
湖南：39,752.12
广西
香港
海南
澳门

Result:

Thus, the program form a userollover in map visualization was ex

Ex.No:7 **CartographicVisualization**
DATE

Aim:

Tobuildcartographicvisualizationformultipledatasetsinvolvingstatesanddistrictsin India .

Procedure:

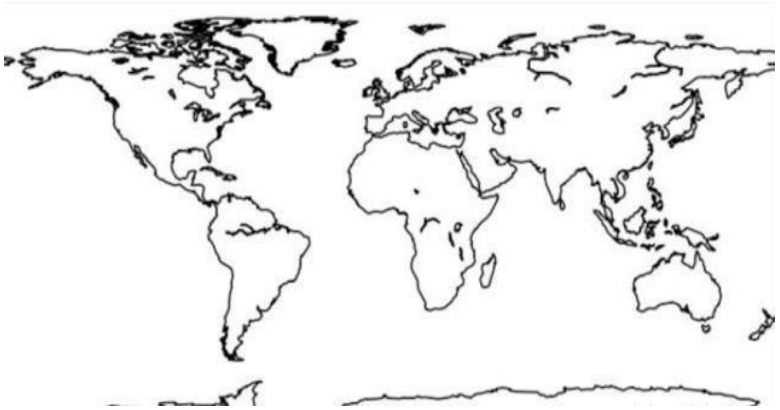
1. Import base map and library
2. Import the state data and map
3. Using matplotlib add the title and attributes to display the map

Program:

```
from mpl_toolkits.basemap import
Basemap import matplotlib.pyplot as
plt
map = Basemap()
map.drawcoastline
s() plt.show()
plt.savefig('test.png')


pipinstallgeopand
as import numpy
as np import
pandas as pd
importmatplotlib.pyplotas
plt import seaborn as sns
importgeopandasasgp
d import shapefile as
shp
from shapely.geometry import Point
sns.set_style('whitegrid')
fp=r'Maps_with_python\india-
polygon.shp' map_df =
gpd.read_file(fp)
map_df_copy=gpd.read_file(fp)
map_df.head()

map_df.plot()
df=pd.read_csv('globallandslides.csv')


state_df = state_df.to_frame()
state_df.reset_index(level=0,inplace=
True) state_df.columns = ['State',
'Count'] state_df.at[15,"Count"] = 69
state_df.at[0,"State"] = "Jammu and Kashmir"
                                        state_df.at[20,"State"] =
"Delhi"state_df.drop(7)
```

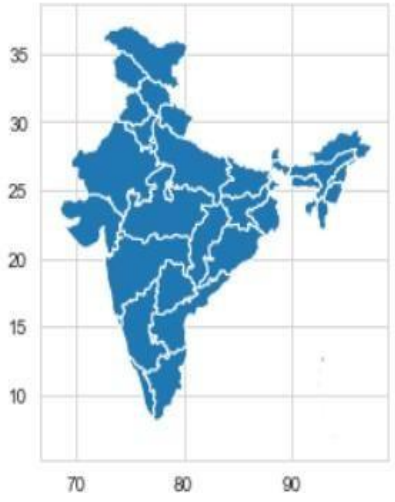
```
pd.set_option('display.max_column
s',None) df =
df[df.country_name=="India"]
df["Year"]=pd.to_datetime(df["event_dat
e"]).dt.year df =
df[df.landslide_category=="landslide"]
ls_df["admin_division_name"].replace("Nāgāland","Nagaland",inplace=True)
ls_df["admin_division_name"].replace("Meghālaya", "Meghalaya",inplace = True)
ls_df["admin_division_name"].replace("TamilNādu","TamilNadu",inplace=True)
ls_df["admin_division_name"].replace("Karnātaka", "Karnataka",inplace = True)
ls_df["admin_division_name"].replace("Gujarāt","Gujarat",inplace=True)
ls_df["admin_division_name"].replace("ArunāchalPradesh","ArunachalPradesh",i
nplace=True) state_df = ls_df["admin_division_name"].value_counts()

#Mergingthedata
merged=map_df.set_index('st_nm').join(state_df.set_i
ndex('State')) merged['Count'] =
merged['Count'].replace(np.nan,0) merged.head()



#CreatefigureandaxesforMatplotlibandsetth
etitle fig, ax = plt.subplots(1, figsize=(10,
10))
ax.axis('off')
ax.set_title('Number of landslides in India state-wise', fontdict={'fontsize':'20',
'fontweight' : '10'})# Plot the figure
merged.plot(column='Count',cmap='YlOrRd', linewidth=0.8, ax=ax,
edgecolor='0',legend=True,markersize=[39.739192,-
104.990337],legend_kwds={'label':"Numberof landslides"})
```

OUTPUT:



	id	st_nm	geometry
0	None	Andaman and Nicobar Islands	MULTIPOLYGON (((93.84831 7.24028, 93.92705 7.0...
1	None	Arunachal Pradesh	POLYGON ((95.23643 26.68105, 95.19594 27.03612...
2	None	Assam	POLYGON ((95.19594 27.03612, 95.08795 26.94578...
3	None	Bihar	POLYGON ((88.11357 26.54028, 88.28006 26.37640...
4	None	Chandigarh	POLYGON ((76.84208 30.76124, 76.83758 30.72552...

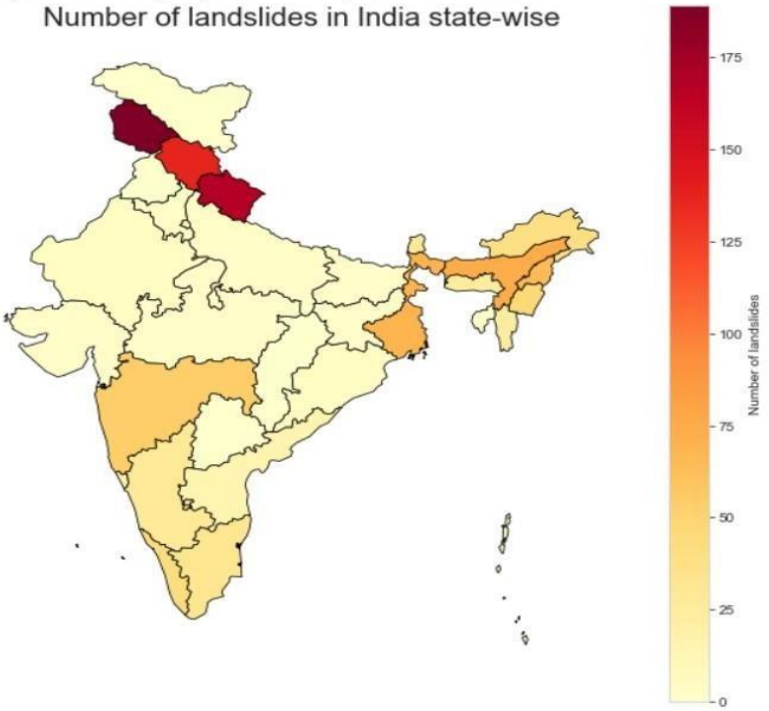
```
<matplotlib.axes._subplots.AxesSubplot at 0x254015d94e0>
```



	State	Count
0	Kashmir	189
1	Uttarakhand	167
2	Himachal Pradesh	137
3	Assam	74
4	Nagaland	65
5	Maharashtra	54
6	Manipur	47
8	Kerala	43
9	Arunachal Pradesh	40
10	Tamil Nadu	32
11	Sikkim	28
12	Karnataka	28
13	Meghalaya	25
14	Mizoram	25
15	West Bengal	69
16	Goa	19
17	Andhra Pradesh	15
18	Rajasthan	5
19	Odisha	5
20	NCT	3
21	Orissa	3
22	Uttar Pradesh	2
23	Gujarat	2
24	Tripura	2
25	Haryana	2
26	State of Odisha	1
27	Madhya Pradesh	1
28	Bihar	1
29	Jharkhand	1

st_nm	id	geometry	Count
Andaman and Nicobar Islands	None	MULTIPOLYGON (((93.84831 7.24028, 93.92705 7.0...	0.0
Arunachal Pradesh	None	POLYGON ((95.23643 26.68105, 95.19594 27.03612...	40.0
Assam	None	POLYGON ((95.19594 27.03612, 95.08795 26.94578...	74.0
Bihar	None	POLYGON ((88.11357 26.54028, 88.28006 26.37640...	1.0
Chandigarh	None	POLYGON ((76.84208 30.76124, 76.83758 30.72552...	0.0

```
<matplotlib.axes._subplots.AxesSubplot at 0x2230d29d908>
```



Result:

Thus, the program to display the cartographic visualization of India was executed successfully.

Ex.No:8 **EDAonWineQualityDataSet**
DATE

Aim:

To write a python program for EDA on Wine Quality Data Set.

Procedure:

1. Import library
2. Import wine dataset
3. Perform EDA to display information, description of data.
4. Analyse the content of alcohol consumption and visualize it

Program:

```
Import pandas as pd
df_red = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learningdatabases/wine-quality/winequality-red.csv", delimiter=";")
df_white=pd.read_csv("https://archive.ics.uci.edu/ml/machine-learningdatabases/wine-quality/winequality-white.csv", delimiter=";")
df_red.columns
```

Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar','chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density','pH','sulphates','alcohol','quality'],dtype ='object')

```
df_red.iloc[100:110]

df_red.dtypes
```

fixed acidity
float64
volatileacidityfloat6
4 citric acid float64
residual sugar
float64 chlorides
float64
free sulfur dioxide
float64
totalsulfurdioxidefloat6
4 density float64
pH float64
sulphatesfloat
64 alcohol
float64
quality int64
dtype: object

```
df_red.describe()

df_red.info()
```

<class
'pandas.core.frame.DataFrame'
>RangeIndex: 1599 entries, 0
to 1598 Data columns (total
12 columns):
fixedacidity1599non-nullfloat64

volatileacidity1599non-nullfloat64
citric acid 1599 non-null float64
residualsugar1599non-nullfloat64
chlorides 1599 non-null float64
free sulfur dioxide 1599 non-null float64
totalsulfurdioxide1599non-nullfloat64
density 1599 non-null float64
pH 1599 non-null float64
sulphates1599non-nullfloat64
alcohol 1599 non-null float64
quality 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB

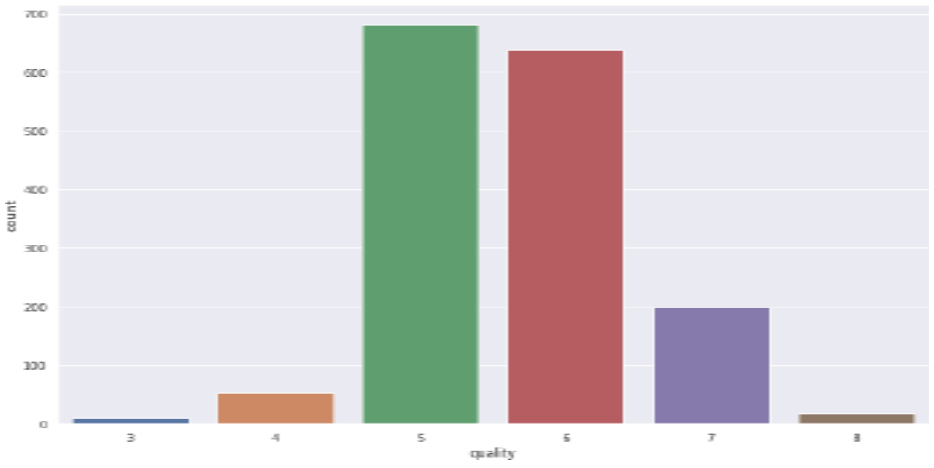
```
import seaborn as sns
sns.set(rc={'figure.figsize': (14, 8)})
sns.countplot(df_red['quality'])
```

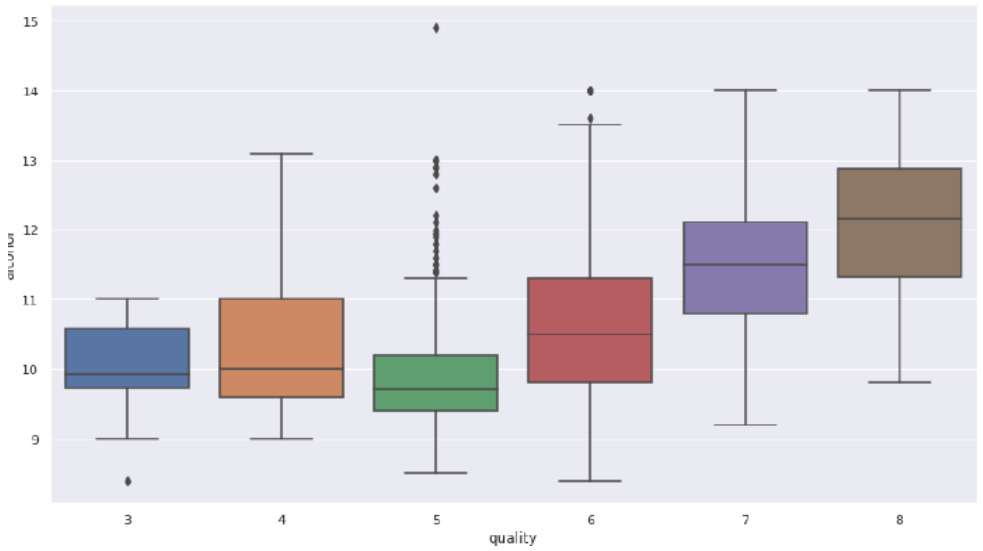
```
Sns.distplot(df_red[‘alchol’])
```

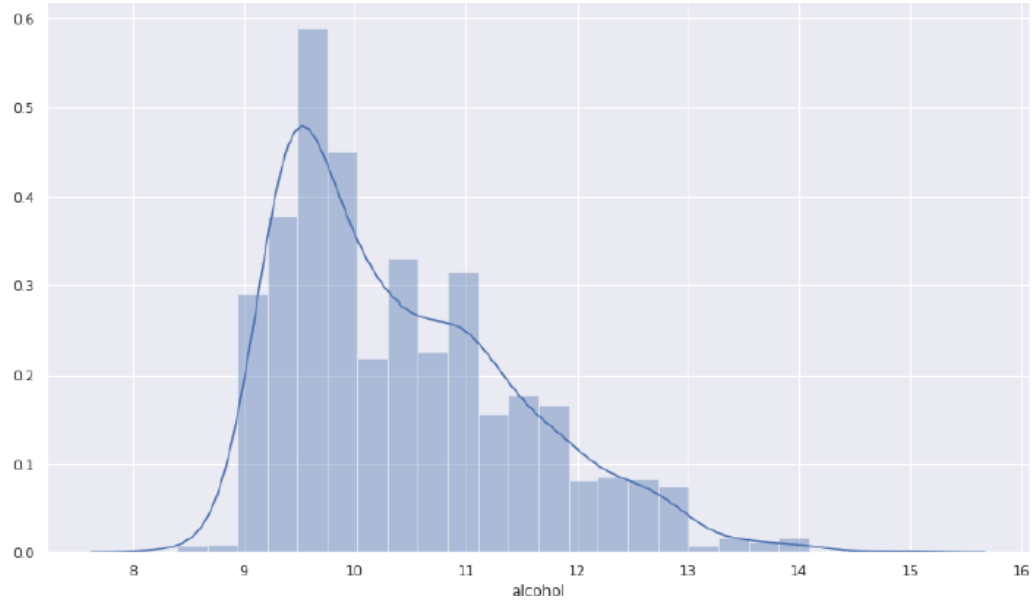
OUTPUT:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
100	8.3	0.610	0.30	2.1	0.084	11.0	50.0	0.9972	3.40	0.61	10.2	6
101	7.8	0.500	0.30	1.9	0.075	8.0	22.0	0.9959	3.31	0.56	10.4	6
102	8.1	0.545	0.18	1.9	0.080	13.0	35.0	0.9972	3.30	0.59	9.0	6
103	8.1	0.575	0.22	2.1	0.077	12.0	65.0	0.9967	3.29	0.51	9.2	5
104	7.2	0.490	0.24	2.2	0.070	5.0	36.0	0.9960	3.33	0.48	9.4	5
105	8.1	0.575	0.22	2.1	0.077	12.0	65.0	0.9967	3.29	0.51	9.2	5
106	7.8	0.410	0.68	1.7	0.467	18.0	69.0	0.9973	3.08	1.31	9.3	5
107	6.2	0.630	0.31	1.7	0.088	15.0	64.0	0.9969	3.46	0.79	9.3	5
108	8.0	0.330	0.53	2.5	0.091	18.0	80.0	0.9976	3.37	0.80	9.6	6
109	8.1	0.785	0.52	2.0	0.122	37.0	153.0	0.9969	3.21	0.69	9.3	5

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065868	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000




15
14
13
12
11
10
9
3
4
5
6
7
8
quality


0.6
0.5
0.4
0.3
0.2
0.1
0.0
8
9
10
11
12
13
14
15
16
alcohol

Result:

Thus, the program to execute the EDA on wine data set was executed successfully.

Ex.No:9 **Case Study on aDataSettopresentanAnalysis Report**
DATE

Aim:
To analysis following using the diabetes dataset from UCI and Pima Indians Diabetes dataset for performing the following

Procedure:

a. Univariateanalysis:Frequency,Mean,Median,Mode,Variance,StandardDeviation, Skewness and Kurtosis.
b. Bivariate analysis :Linear and logistic regression modeling
c. Multiple Regression analysis
d. Also compare the results of the above analysis for the two data sets.

Program:

a. Univariateanalysis:Frequency,Mean,Median,Mode,Variance,StandardDeviation, Skewness and Kurtosis.

```
import pandas as pd
import numpy as np
importstatisticsasst

# Loadthedata
df=pd.read_csv("data_desc.csv") print(df.shape)
print(df.info())
```

python
Output:

1Dependents	0.748333
2Income	705541.333333
3Loan_amount	323793.666667
4Term_months	183.350000
5Age	49.450000

Output:

(600, 10)

<class'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Datacolumns(total10columns):
Marital_status 600 non-null
object Dependents
600 non-null int64
Is_graduate 600 non-null object
Income 600 non-null int64
Loan_amount 600non-
nullint64 Term_months
600non-nullint64
Credit_score 600 non-null object
approval_status
600non-nullobject Age 600
non-null int64
Sex 600non-nullobject
dtypes: int64(5), object(5) memory
usage: 47.0+ KB
None

MeasuresofCentralTendency

Measures of central tendency describe the center of the data , and are often represented by the mean, the median, and the mode.

Mean

df.mean()

6dtype:float64

Itisalsopossibletocalculatethemeanofaparticularvariableinadata,as shownbelow,wherewe calculate the mean of the variables 'Age' and 'Income'.

```
print(df.loc[:,'Age'].mean())
print(df.loc[:,'Income'].mean())
```

python

Output:

149.45
2705541.33

It is also possible to calculate the mean of the rows by specifying the **(axis =1)** argument . The code below calculates the mean of the first five rows.

```
df.mean(axis=1)[0:5]
```

python

Output:
10 70096.0
21 161274.
0
32 125113.
4
43 119853.
8
54 120653.
8
6dtype:float64

Median df.median()

python
Output:
1 Dependents 0.0
2 Income 508350.0
3 Loan_amount 76000.0
4 Term_months 192.0
5 Age 51.0
6 dtype:float64

Mode

df.mode()

python
Output:

1	Marital_status	Dependents	Is_graduate	Income	Loan_amount	Term_months	Credit_score	approval_status	Age	Sex
2	---------	-------	-------	------	---------	---------	---------	----------	--	--
3	yes	0	yes	333300	70000	192.0	satisfactory	yes	55	M

MeasuresofDispersion

Themost popularmeasures ofdispersion arestanderddeviation, variance,and the interquartile range.

Median df.median()

python
Output:

7 Dependents 0.0
8 Income 508350.0
9 Loan_amount 76000.0
10 Term_months 192.0
11 Age 51.0
12 dtype:flo

at64 **Mode**

df.mode()

python
Output:

1	Marital_status	Dependents	Is_graduate	Income	Loan_amount	Term_months	Credit_score	approval_status	Age	Sex
2	----------	-------	-------	-----	--------	--------	--------	---------	--	--
3	yes	0	yes	333300	70000	192.0	satisfactory	yes	55	M

MeasuresofDispersion

Themost popularmeasures ofdispersion arestandarddeviation, variance,and the interquartile range.

StandardDeviation

df.std()

python
Output:
1Dependents 1.026362
2Income 711421.814154
3 Loan_amount 724293.4807ε2
4 Term_months 31.933949
5 Age 14.728511
6 dtype:float64

Variance

df.var()

python
Output:
1 Dependents 1.053420e+00
2 Income 5.061210e+11
3 Loan_amount 5.246010e+11
4 Term_months 1.019777e+03
5 Age 2.169290e+02
6 dtype:float64

InterquartileRange (IQR)

fromscipy.statsimportiqr
iqr(df['Age'])

python
Output:
125.0

Skewness

print(df.skew())

python
Output:
1 Dependents 1.169632
2 Income 5.344587
3 Loan_amount 5.006374
4 Term_months-2.471879
5 Age -0.055537
6 dtype:float64

Theskewnessvalues canbeinterpretedin thefollowingmanner:

- **Highlyskeweddistribution**:Iftheskewnessvalueislessthan−1orgreaterthan+1.
- **Moderatelyskeweddistribution**:Iftheskewnessvalueisbetween−1and−½orbetween+½ and +1.
- **Approximatelysymmetricdistribution**:Iftheskewnessvalueisbetween−½and +½.

PuttingEverythingTogether

df.describe()

python
Output:

	Dependents	Income	Loan_amount	Term_months	Age
count	600.000000	6.000000e+02	6.000000e+02	600.000000	600.000000
mean	0.748333	7.055413e+05	3.237937e+05	183.350000	49.450000
std	1.026362	7.114218e+05	7.242935e+05	31.933949	14.728511
min	0.000000	3.000000e+04	1.090000e+04	18.000000	22.000000
25%	0.000000	3.849750e+05	6.100000e+04	192.000000	36.000000
50%	0.000000	5.083500e+05	7.600000e+04	192.000000	51.000000
75%	1.000000	7.661000e+05	1.302500e+05	192.000000	61.000000
max	6.000000	8.444900e+06	7.780000e+06	252.000000	76.000000

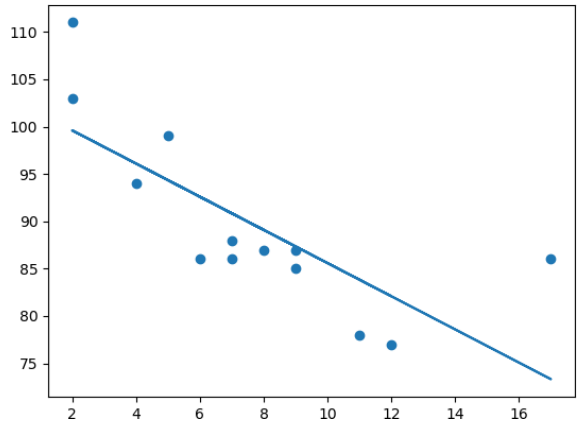
df.describe(include='all')

b. Bivariateanalysis:Linearandlogisticregressionmodeling

Linear Regression

```
importmatplotlib.pyplotasplt
from scipy import stats

x=[5,7,8,7,2,17,2,9,4,11,12,9,6]
y=[99,86,87,88,111,86,103,87,94,78,77,85,86]

slope,intercept,r,p,std_err=stats.linregress(x,y) def

myfunc(x):
  returnslope *x+intercept

mymodel=list(map(myfunc,x)) plt.scatter(x,

y)
plt.plot(x,mymodel)
plt.show()
```

OUTPUT:



Logistic Regression

```
import numpy
fromsklearnimportlinear_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)

deflogit2prob(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]]
```

Results Explained

3.780.61 Theprobabilitythat atumor withthesize3.78cm iscancerous is61%.
2.440.19 Theprobabilitythat atumor with thesize2.44cm iscancerous is19%.
2.90.13 Theprobabilitythat atumor with thesize2.09cm iscancerous is13%.

c.MultipleRegression analysis

Multipleregressionworksbyconsideringthevaluesoftheavailablemultipleindependent variables and predicting the value of one dependent variable.

```
importpandasaspd
fromsklearnimportlinear_model import
statsmodels.api as sm

data = {'year':
[2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2016,2016,2016,2016,2016,2016,20
16,2016,2016,2016,2016,2016,2016],
      'month':[12,11,10,9,8,7,6,5,4,3,2,1,12,11,10,9,8,7,6,5,4,3,2,1],
      'interest_rate':
[2.75,2.5,2.5,2.5,2.5,2.5,2.5,2.25,2.25,2.25,2,2,2,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75],
      'unemployment_rate':
[5.3,5.3,5.3,5.3,5.4,5.6,5.5,5.5,5.5,5.6,5.7,5.9,6,5.9,5.8,6.1,6.2,6.1,6.1,6.1,5.9,6.2,6.2,6.1],
      'index_price':
[1464,1394,1357,1293,1256,1254,1234,1195,1159,1167,1130,1075,1047,965,943,958,971,949,884,866,
876,822,704,719]
      }

df = pd.DataFrame(data)

x=df[['interest_rate','unemployment_rate']] y
= df['index_price']

# with sklearn
regr=linear_model.LinearRegression()
regr.fit(x, y)

print('Intercept:\n',regr.intercept_)
print('Coefficients: \n', regr.coef_)

# with statsmodels
x=sm.add_constant(x) #addingaconstant

model = sm.OLS(y, x).fit()
predictions=model.predict(x)

print_model=model.summary()
print(print_model)
```

OUTPUT:

Intercept:
1798.4039776258564
Coefficients:
[345.54008701 -250.14657137]

OLSRegressionResults

Dep. Variable:	index_price	R-squared:	0.898
Model:	OLS	Adj. R-squared:	0.888
Method:	LeastSquares	F-statistic:	92.07
Date:	Sat, 30 Jul 2022	Prob (F-statistic):	4.04e-11
Time:	13:47:01	Log-Likelihood:	-134.61
No. Observations:	24	AIC:	275.2
DfResiduals:	21	BIC:	278.8
Df Model:	2		
CovarianceType:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1798.4040	899.248	2.000	0.059	-71.685	3668.493
interest_rate	345.5401	111.367	3.103	0.005	113.940	577.140
unemployment_rate	-250.1466	117.950	-2.121	0.046	-495.437	-4.856

Omnibus:	2.691	Durbin-Watson:	0.530
Prob(Omnibus):	0.260	Jarque-Bera(JB):	1.551
Skew:	-0.612	Prob(JB):	0.461
Kurtosis:	3.226	Cond. No.	394.

Result:

Thus the Univariate, Bivariate and Multiple Regression Analysis using the diabetes data set from UCI and Pima Indians Diabetes data set was completed and verified successfully