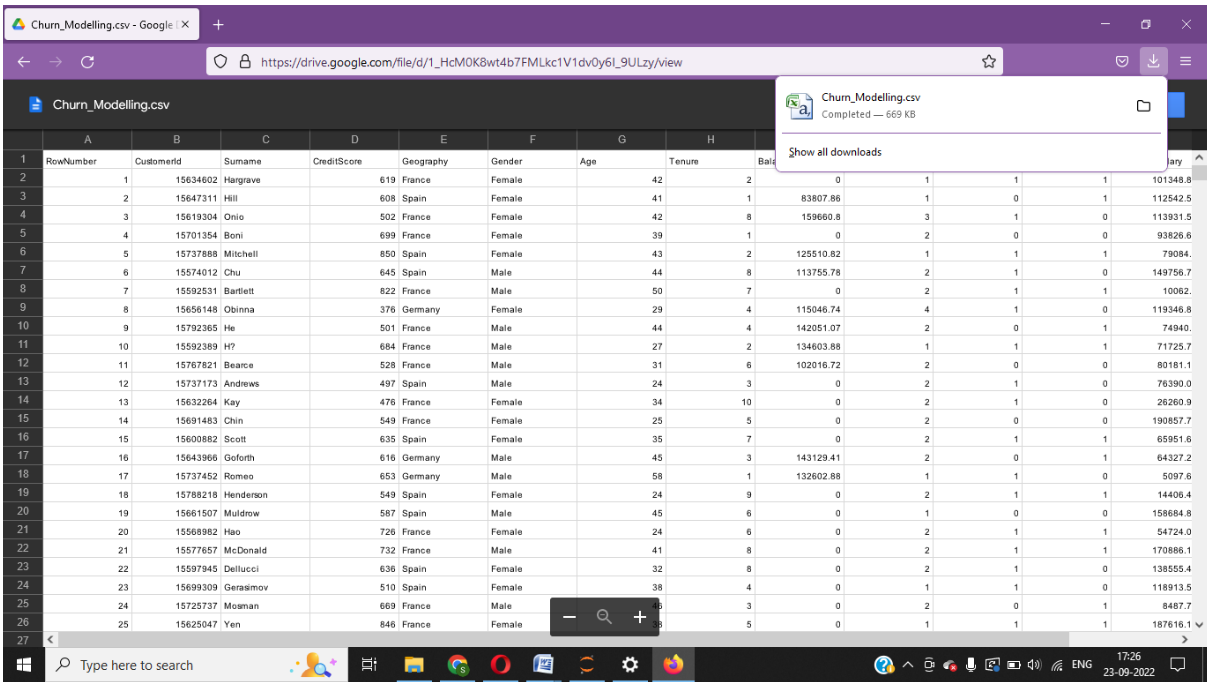
Assignment -2
Data Visualization and Data Pre-processing

Assignment Date	17 September 2022
Student Name	Manjunathan V
Student Roll Number	727719EUCS080
Maximum Marks	2 Marks

Question-1:

Download the dataset: Dataset



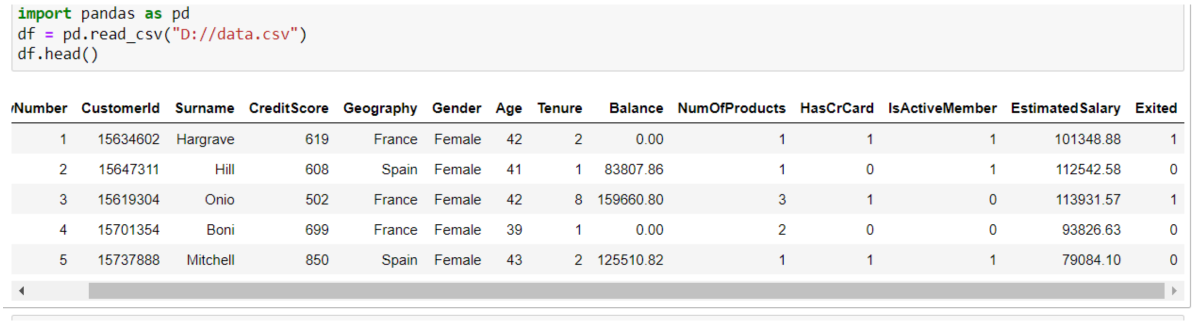
	A	B	C	D	E	F	G	H					
1	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bala				lary
2	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.8
3	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.5
4	3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.5
5	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.6
6	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.7
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.
9	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.8
10	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.7
12	11	15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.1
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.9
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.7
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.6
17	16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	0	1	64327.2
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.6
19	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.4
20	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158684.8
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.0
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.1
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.4
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.5
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.7
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	187616.1
27													

Question-2:

Load the dataset.

Solution:

import pandas as pd
df = pd.read_csv("D://data.csv")
df.head()



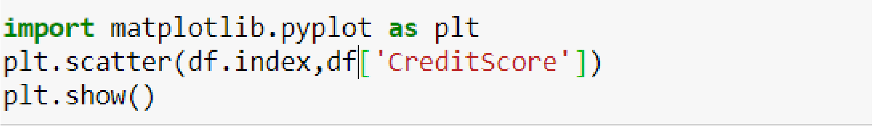
```
import pandas as pd
df = pd.read_csv("D://data.csv")
df.head()
```

Number	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

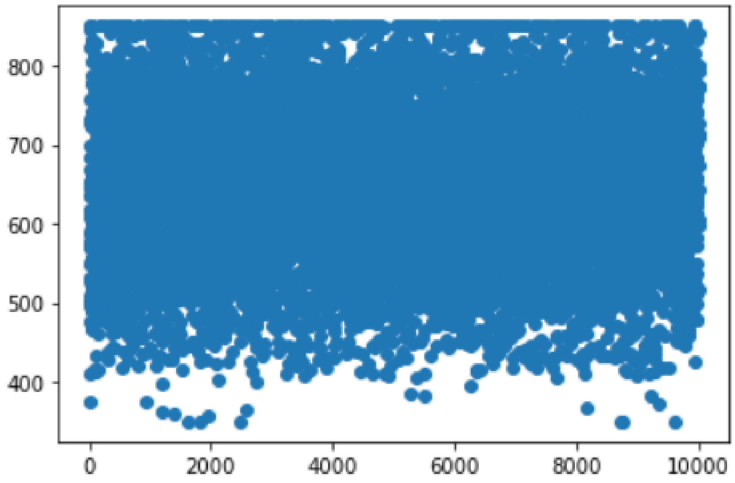
Question-3:

Perform Below Visualizations.

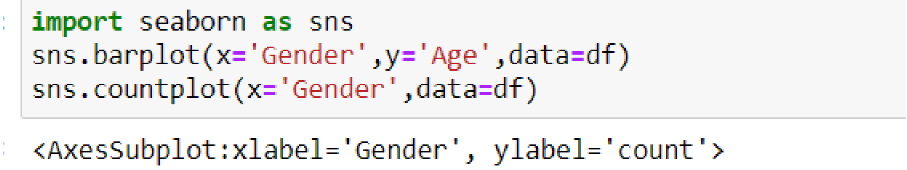
- Univariate Analysis



```
import matplotlib.pyplot as plt
plt.scatter(df.index,df['CreditScore'])
plt.show()
```

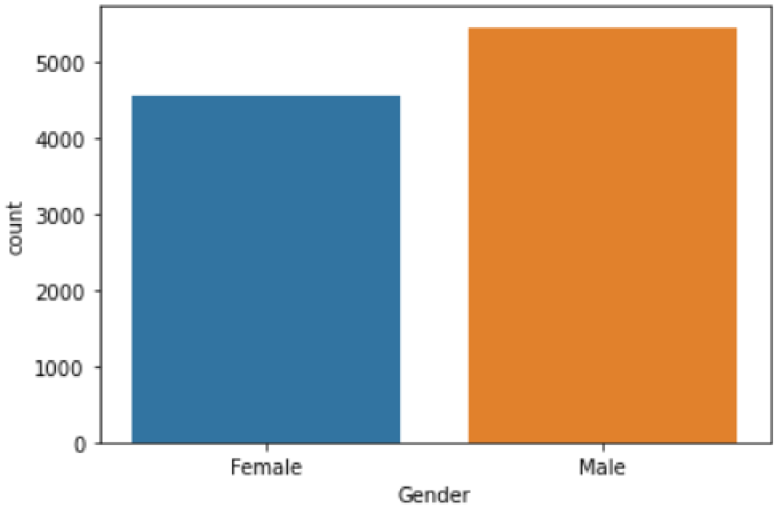


- Bi - Variate Analysis

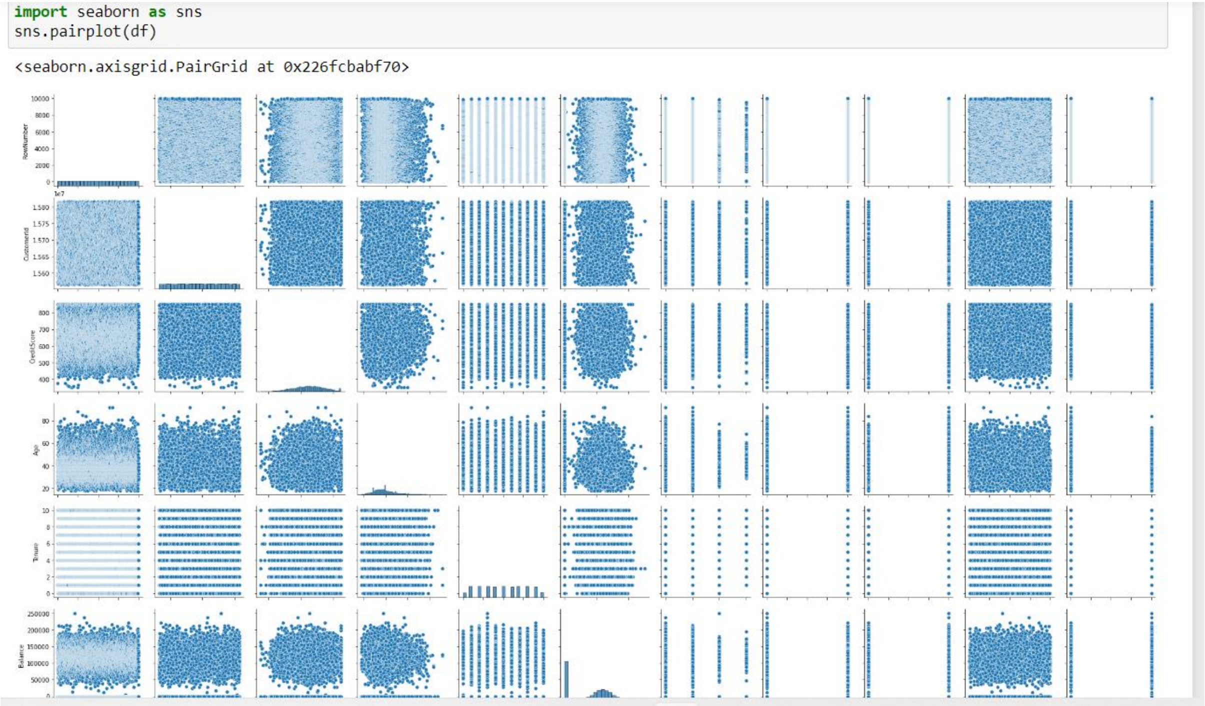


```
import seaborn as sns
sns.barplot(x='Gender',y='Age',data=df)
sns.countplot(x='Gender',data=df)
```

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



- Multi - Variate Analysis

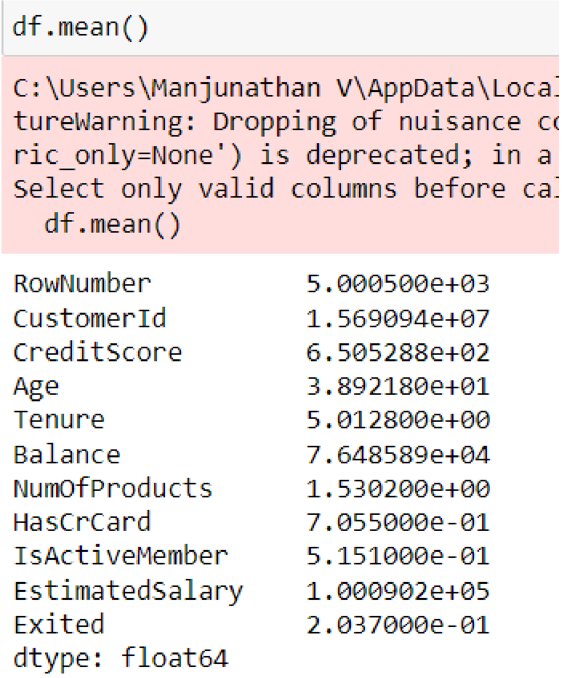


```
import seaborn as sns
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x226fcbabf70>
```

Question-4:

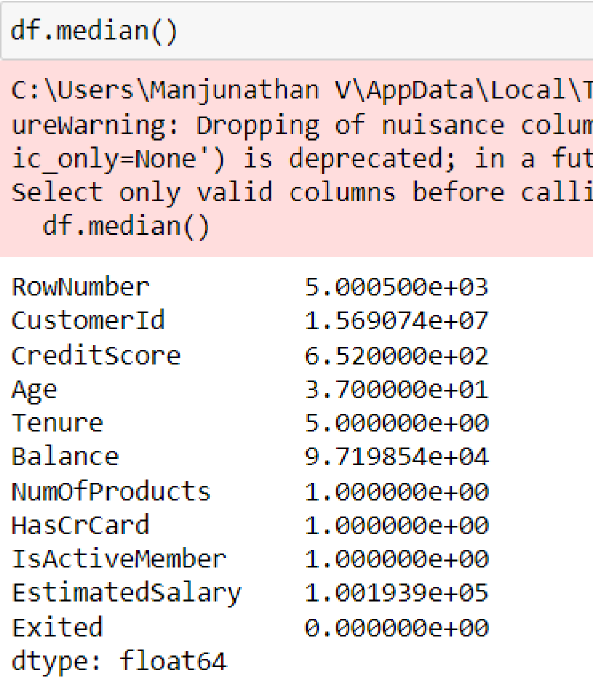
Perform descriptive statistics on the dataset.



```
df.mean()
```

```
C:\Users\Manjunathan V\AppData\Local
tureWarning: Dropping of nuisance co
ric_only=None') is deprecated; in a
Select only valid columns before cal
  df.mean()
```

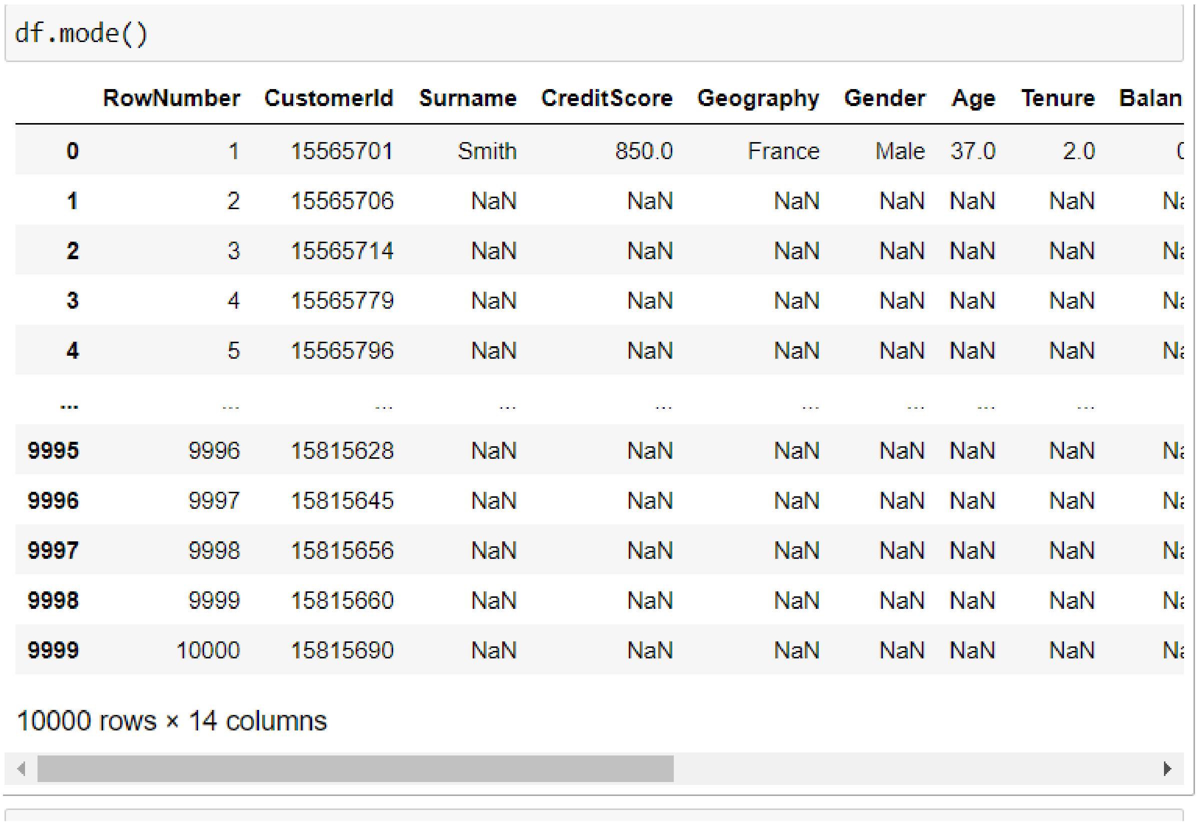
```
RowNumber          5.000500e+03
CustomerId         1.569094e+07
CreditScore        6.505288e+02
Age                3.892180e+01
Tenure             5.012800e+00
Balance            7.648589e+04
NumOfProducts      1.530200e+00
HasCrCard          7.055000e-01
IsActiveMember     5.151000e-01
EstimatedSalary    1.000902e+05
Exited             2.037000e-01
dtype: float64
```



```
df.median()
```

```
C:\Users\Manjunathan V\AppData\Local\T
ureWarning: Dropping of nuisance colum
ic_only=None') is deprecated; in a fut
Select only valid columns before calli
  df.median()
```

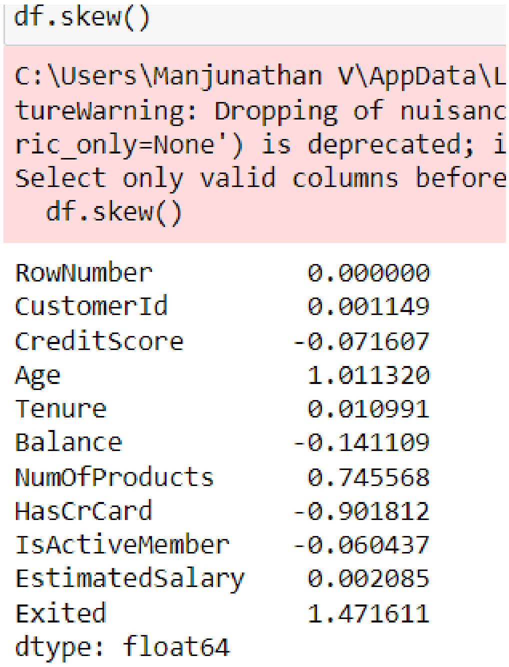
```
RowNumber          5.000500e+03
CustomerId         1.569074e+07
CreditScore        6.520000e+02
Age                3.700000e+01
Tenure             5.000000e+00
Balance            9.719854e+04
NumOfProducts      1.000000e+00
HasCrCard          1.000000e+00
IsActiveMember     1.000000e+00
EstimatedSalary    1.001939e+05
Exited             0.000000e+00
dtype: float64
```



```
df.mode()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balan
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	Na
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	Na
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	Na
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	Na
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	Na

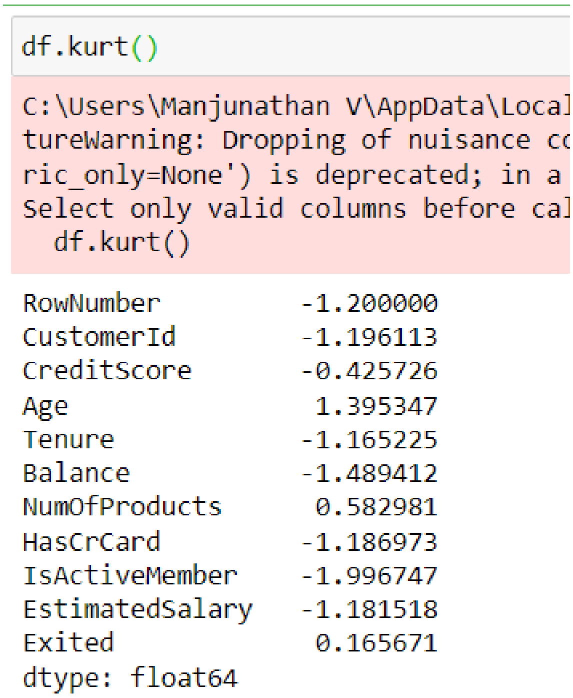
10000 rows × 14 columns



```
df.skew()
```

```
C:\Users\Manjunathan V\AppData\L
tureWarning: Dropping of nuisanc
ric_only=None') is deprecated; i
Select only valid columns before
  df.skew()
```

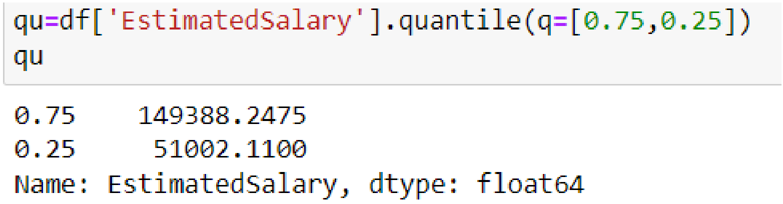
```
RowNumber          0.000000
CustomerId         0.001149
CreditScore       -0.071607
Age                1.011320
Tenure             0.010991
Balance           -0.141109
NumOfProducts      0.745568
HasCrCard         -0.901812
IsActiveMember    -0.060437
EstimatedSalary    0.002085
Exited             1.471611
dtype: float64
```



```
df.kurt()
```

```
C:\Users\Manjunathan V\AppData\Local
tureWarning: Dropping of nuisance co
ric_only=None') is deprecated; in a
Select only valid columns before cal
  df.kurt()
```

```
RowNumber         -1.200000
CustomerId        -1.196113
CreditScore       -0.425726
Age                1.395347
Tenure            -1.165225
Balance           -1.489412
NumOfProducts      0.582981
HasCrCard         -1.186973
IsActiveMember    -1.996747
EstimatedSalary   -1.181518
Exited             0.165671
dtype: float64
```



```
qu=df['EstimatedSalary'].quantile(q=[0.75,0.25])
qu
```

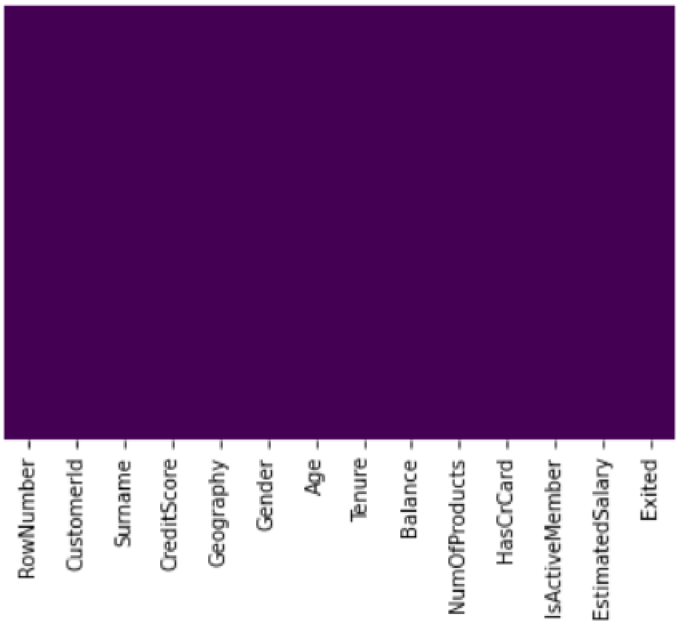
```
0.75    149388.2475
0.25     51002.1100
Name: EstimatedSalary, dtype: float64
```

Question-5:

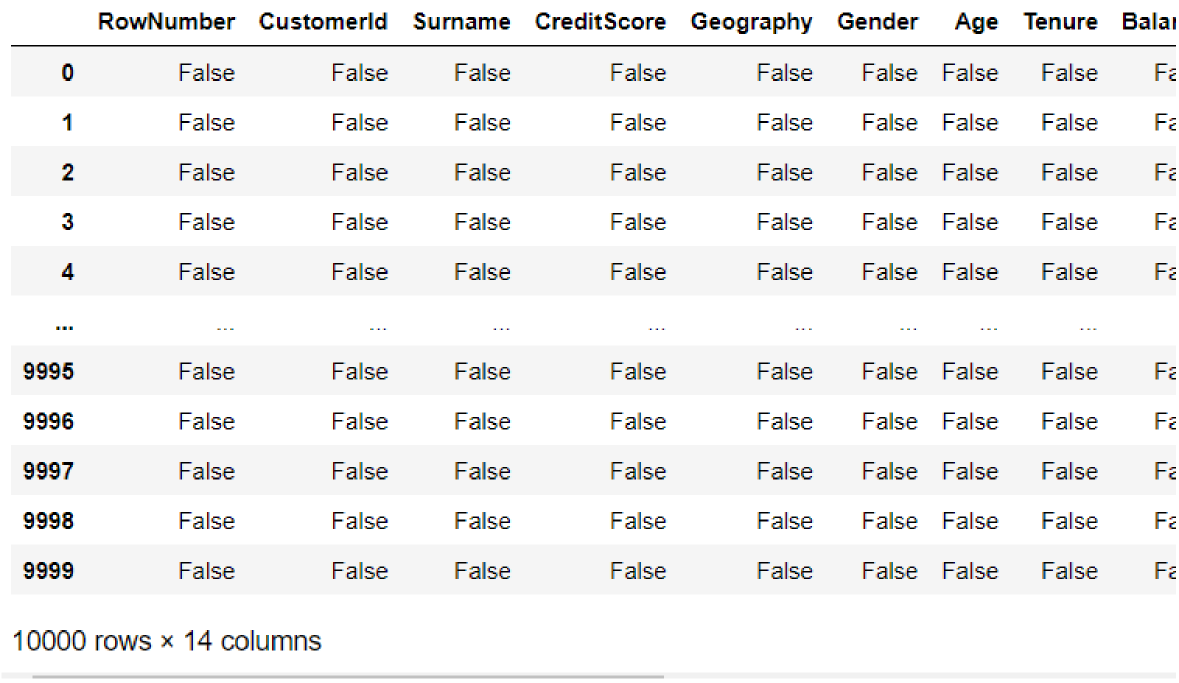
Handle the Missing values.

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap="viridis")
#there is no missing values as per heatmap
```

```
<AxesSubplot:>
```



```
df.isnull()
```

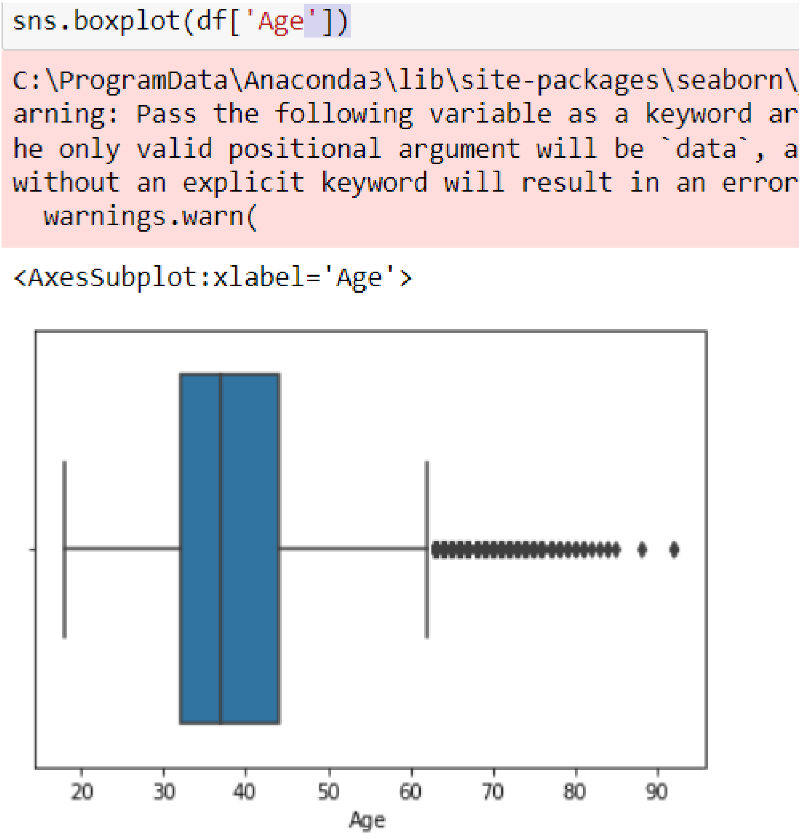


	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balaı
0	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	
9995	False	False	False	False	False	False	False	False	Fa
9996	False	False	False	False	False	False	False	False	Fa
9997	False	False	False	False	False	False	False	False	Fa
9998	False	False	False	False	False	False	False	False	Fa
9999	False	False	False	False	False	False	False	False	Fa

10000 rows × 14 columns

Question-6:

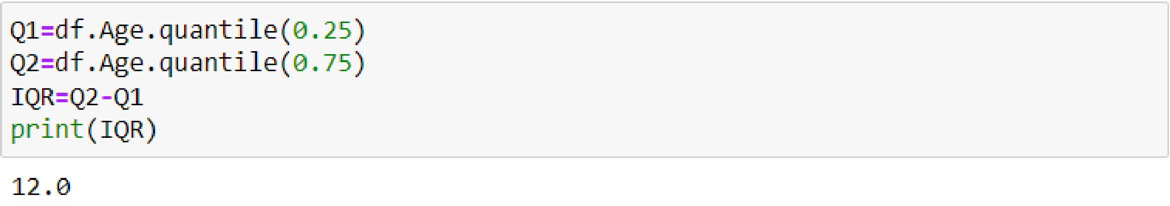
Find the outliers and replace the outliers



```
sns.boxplot(df['Age'])
```

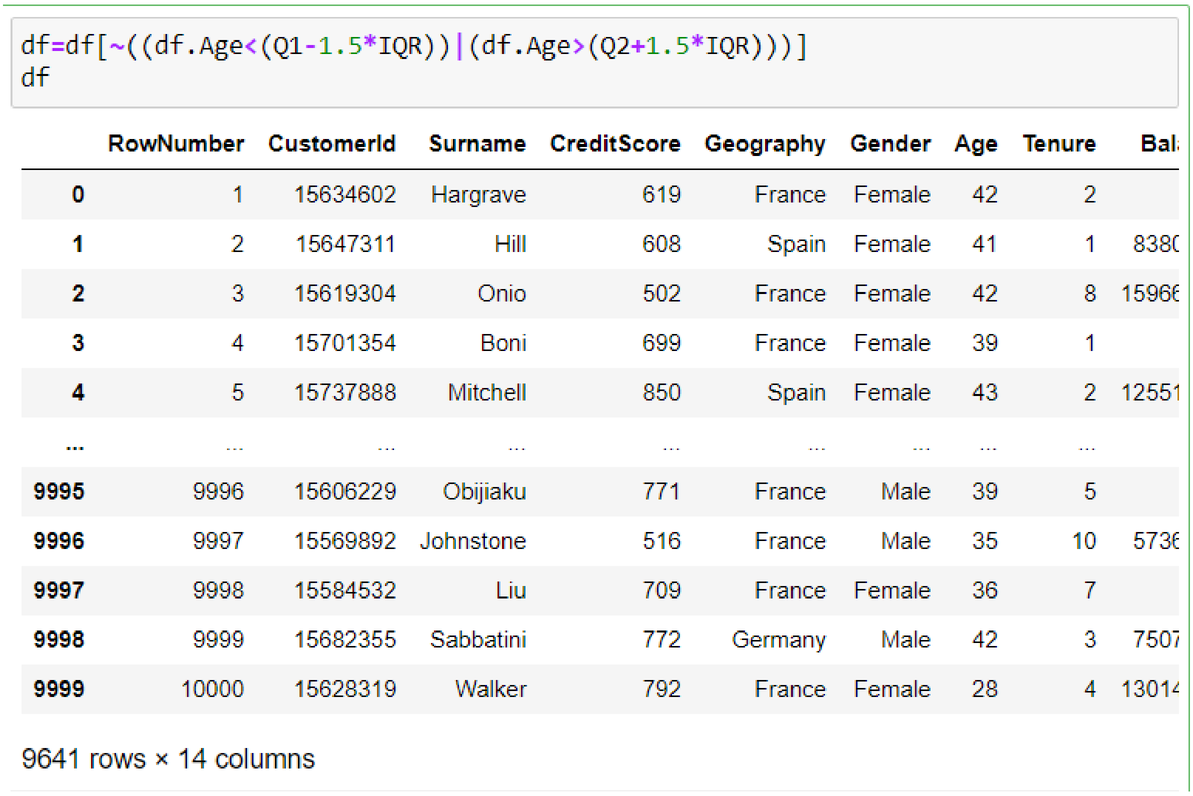
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\
arning: Pass the following variable as a keyword ar
he only valid positional argument will be `data`, a
without an explicit keyword will result in an error
  warnings.warn(
```

```
<AxesSubplot:xlabel='Age'>
```



```
Q1=df.Age.quantile(0.25)
Q2=df.Age.quantile(0.75)
IQR=Q2-Q1
print(IQR)
```

```
12.0
```



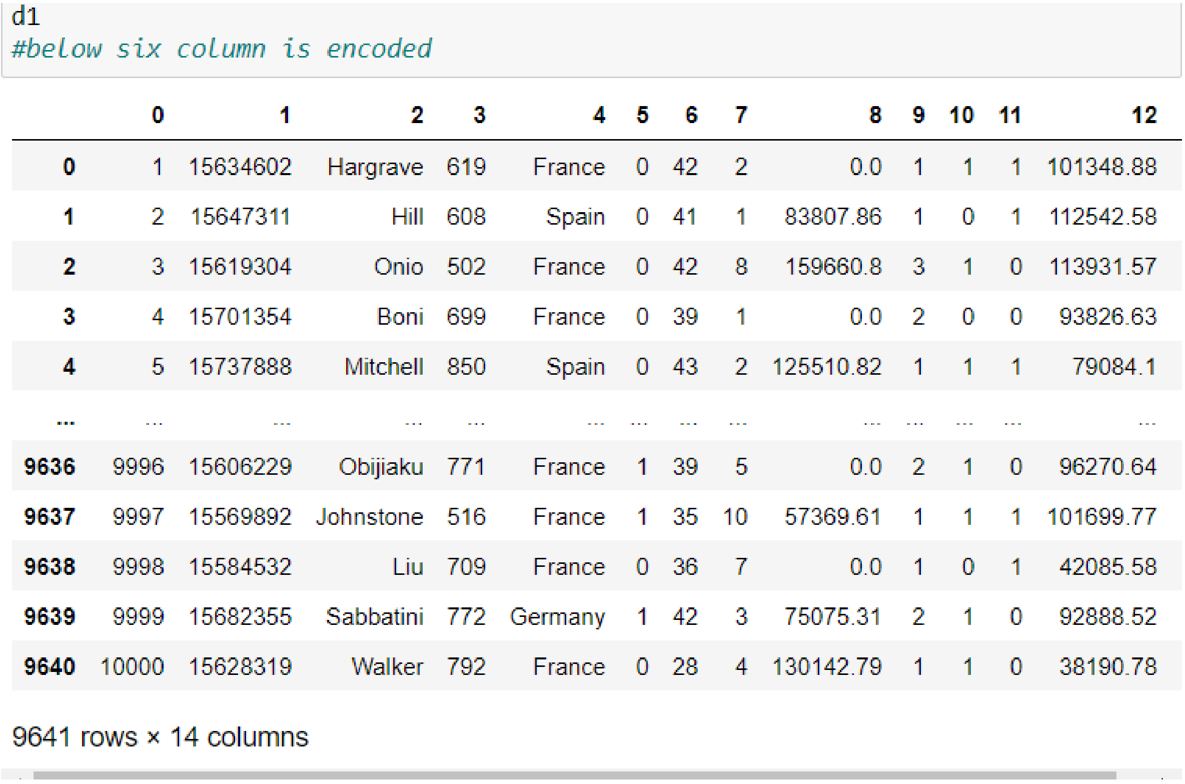
```
df=df[~((df.Age<(Q1-1.5*IQR))|(df.Age>(Q2+1.5*IQR)))]
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Bal
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	8380
2	3	15619304	Onio	502	France	Female	42	8	15966
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	12551
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	5736
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	7507
9999	10000	15628319	Walker	792	France	Female	28	4	13014

9641 rows × 14 columns

Question-7:

Check for Categorical columns and perform encoding.



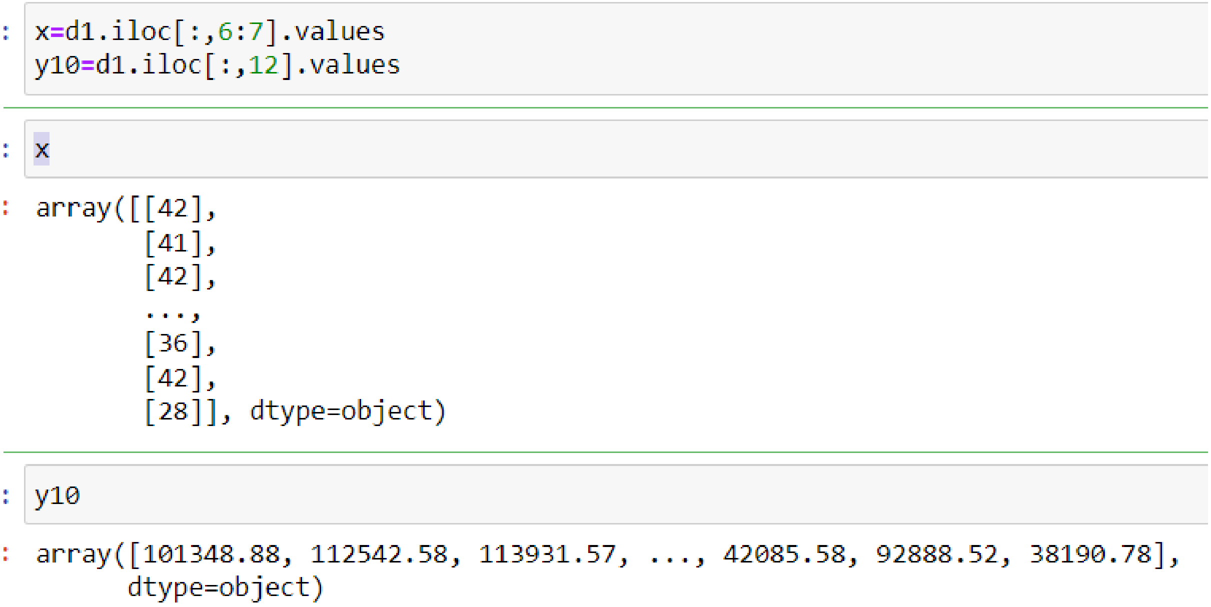
```
d1
#below six column is encoded
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	15634602	Hargrave	619	France	0	42	2	0.0	1	1	1	101348.88
1	2	15647311	Hill	608	Spain	0	41	1	83807.86	1	0	1	112542.58
2	3	15619304	Onio	502	France	0	42	8	159660.8	3	1	0	113931.57
3	4	15701354	Boni	699	France	0	39	1	0.0	2	0	0	93826.63
4	5	15737888	Mitchell	850	Spain	0	43	2	125510.82	1	1	1	79084.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9636	9996	15606229	Obijiaku	771	France	1	39	5	0.0	2	1	0	96270.64
9637	9997	15569892	Johnstone	516	France	1	35	10	57369.61	1	1	1	101699.77
9638	9998	15584532	Liu	709	France	0	36	7	0.0	1	0	1	42085.58
9639	9999	15682355	Sabbatini	772	Germany	1	42	3	75075.31	2	1	0	92888.52
9640	10000	15628319	Walker	792	France	0	28	4	130142.79	1	1	0	38190.78

9641 rows × 14 columns

Question-8:

Split the data into dependent and independent variables.



```
: x=d1.iloc[:,6:7].values
  y10=d1.iloc[:,12].values
```

```
: x
```

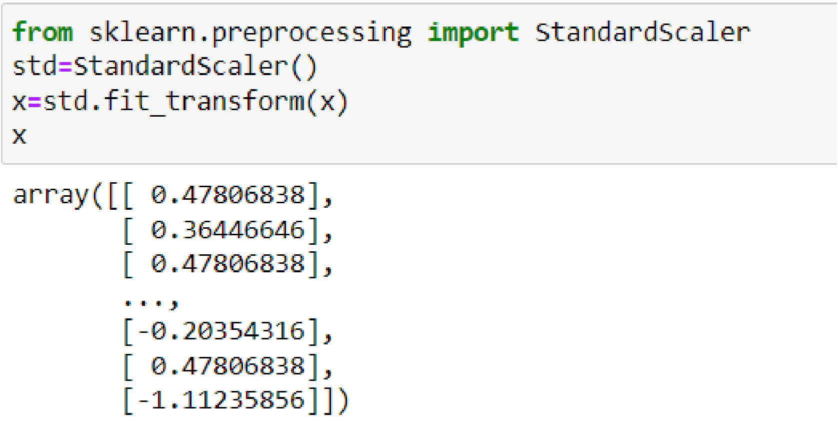
```
: array([[42],
         [41],
         [42],
         ...,
         [36],
         [42],
         [28]], dtype=object)
```

```
: y10
```

```
: array([101348.88, 112542.58, 113931.57, ..., 42085.58, 92888.52, 38190.78],
        dtype=object)
```

Question-9:

Scale the independent variables

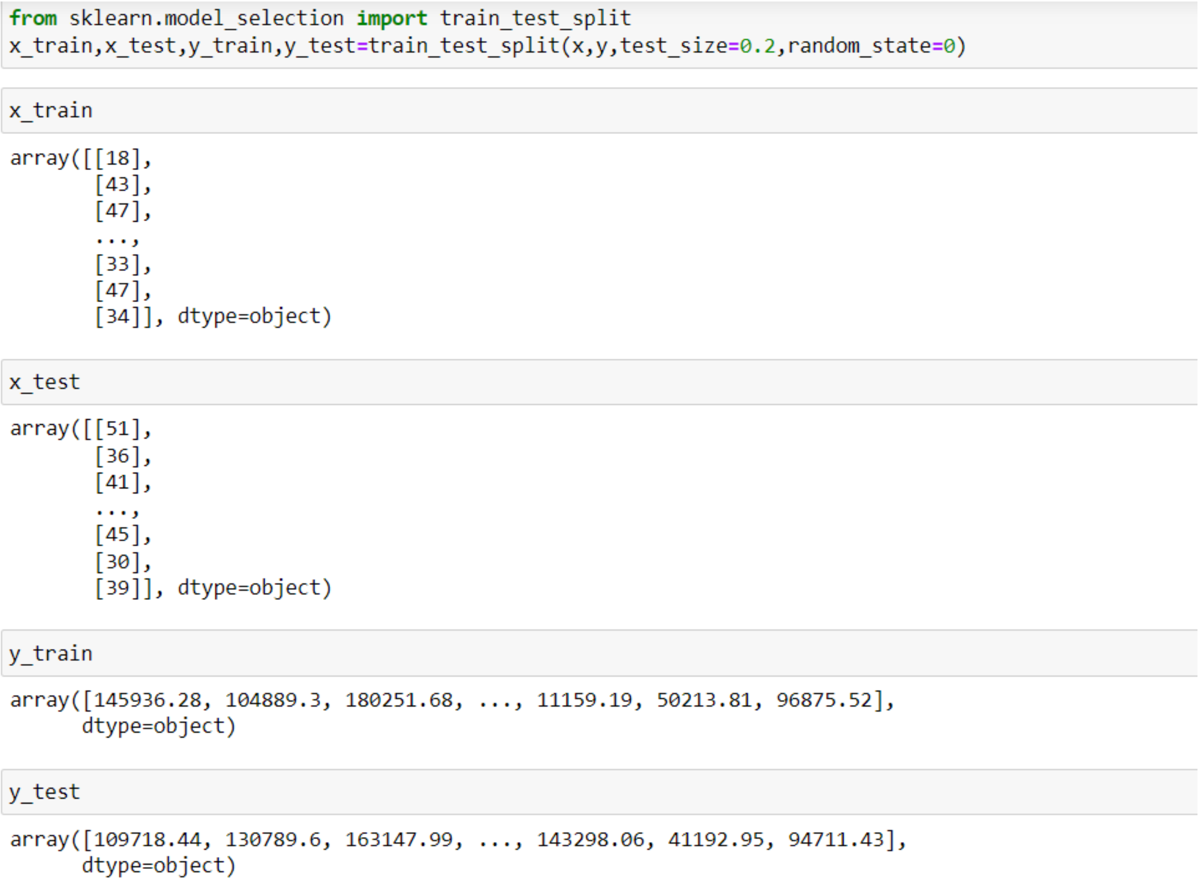


```
from sklearn.preprocessing import StandardScaler
std=StandardScaler()
x=std.fit_transform(x)
x
```

```
array([[ 0.47806838],
       [ 0.36446646],
       [ 0.47806838],
       ...,
       [-0.20354316],
       [ 0.47806838],
       [-1.11235856]])
```

Question-10:

Split the data into training and testing



```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
x_train
```

```
array([[18],
       [43],
       [47],
       ...,
       [33],
       [47],
       [34]], dtype=object)
```

```
x_test
```

```
array([[51],
       [36],
       [41],
       ...,
       [45],
       [30],
       [39]], dtype=object)
```

```
y_train
```

```
array([145936.28, 104889.3, 180251.68, ..., 11159.19, 50213.81, 96875.52],
      dtype=object)
```

```
y_test
```

```
array([109718.44, 130789.6, 163147.99, ..., 143298.06, 41192.95, 94711.43],
      dtype=object)
```