Assignment -2
Artificial Intelligence

Student Name	KANISHKESVAR A
Student Roll Number	73771921142
Maximum Marks	2 Marks

Question-1:

1. Download the dataset:
2. Load the dataset.

```
import pandas as pd import
numpy as np import
matplotlib.pyplot as plt

url =
'https://drive.google.com/file/d/1_HcM0K8wt4b7FMLkc1V1dv0y6I_9ULzy/
view?usp=sharing' path = 'https://drive.google.com/uc?
export=download&id='+url.split('/')[-2] df = pd.read_csv(path)

df.sample(20)
```

```
      RowNumber CustomerId        Surname CreditScore Geography Gender
 Age \
8075       8076    15745250     Simpson          850    France    Male
58
4957       4958    15600478      Watson          752    France    Male
39
6841       6842     15793491 Cherkasova           714 Germany     Male
26
4965       4966    15729515    McCarthy          782    France    Male
36
2828       2829    15716449      Fraser          527     Spain    Male
33
4732       4733    15653937    McIntyre             638 Germany Female
53
6210       6211    15592197     Simmons          522     Spain    Male
30
5505       5506     15802466 Donaldson          534     France Female
53
6450       6451    15781409     Lazarev          834     France Female
28
5407       5408    15714431         Yeh          561    France    Male
37
7529       7530    15575430      Robson          579     France Female
33
1887       1888    15680918     Freeman          613     Spain    Male
```

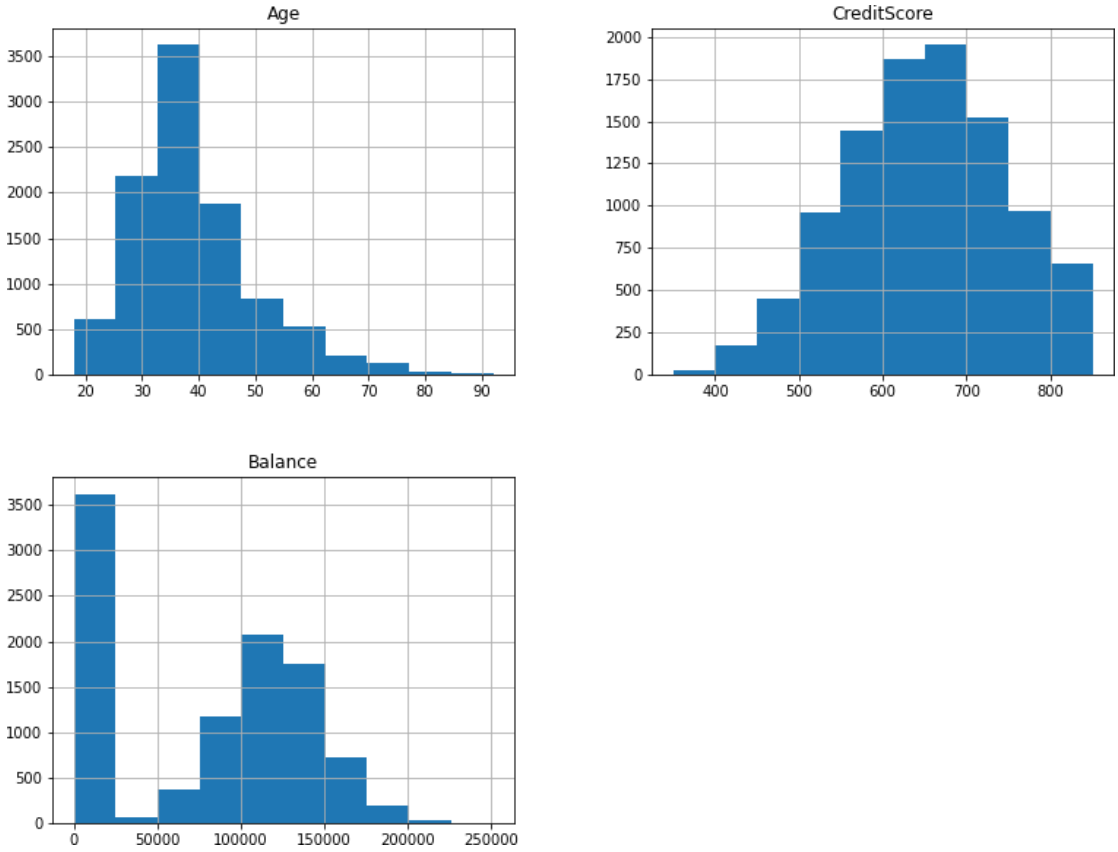
34
1590 1591 15651802 Day 632 Spain Female
39
7578 7579 15656417 Marsh 582 France Female
39
2692 2693 15736274 Prokhorova 751 France Male
31
7031 7032 15580914 Okechukwu 478 Spain Male
48
2158 2159 15685706 Bird 731 France Female

```
40
3549       3550     15647725 Napolitano               675 France Female
61
3772       3773     15699486     Johnson              745        Spain  Male
34
5328       5329     15680234        Bray              667 Germany  Male
27

      Tenure Balance NumOfProducts HasCrCard IsActiveMember \
8075       8 156652.13              1          0                0
4957       3       0.00             1          1                0
6841       3 119545.48              2          1                0
4965       1 148795.17              2          1                1
2828       9 132168.28              1          0                0
4732       1 123916.67              1          1                0
6210       3       0.00             2          1                0
5505       7       0.00             2          1                1
6450       6       0.00             1          1                0
5407       1 100443.36              2          0                1
7529       1 118392.75              1          1                1
1887       8 117300.02              1          1                0
1590       5 97854.37               2          1                0
7578       1 132077.48              2          1                0
2692       8       0.00             2          0                0
7031       0 83287.05               2          0                1
2158       7 118991.79              1          1                1
3549       5 62055.17               3          1                0
3772       7 132944.53              1          1                1
5328       2 138032.15              1          1                0

      EstimatedSalary Exited
8075         25899.21       1
4957        188187.05       0
6841         65482.94       0
4965        195681.43       0
2828         98734.15       0
4732         16657.68       1
6210        145490.85       0
5505         80619.17       0
6450         74287.53       0
5407        101693.73       0
7529        157564.75       0
1887        139410.08       0
1590         93536.38       0
7578        192255.15       0
```

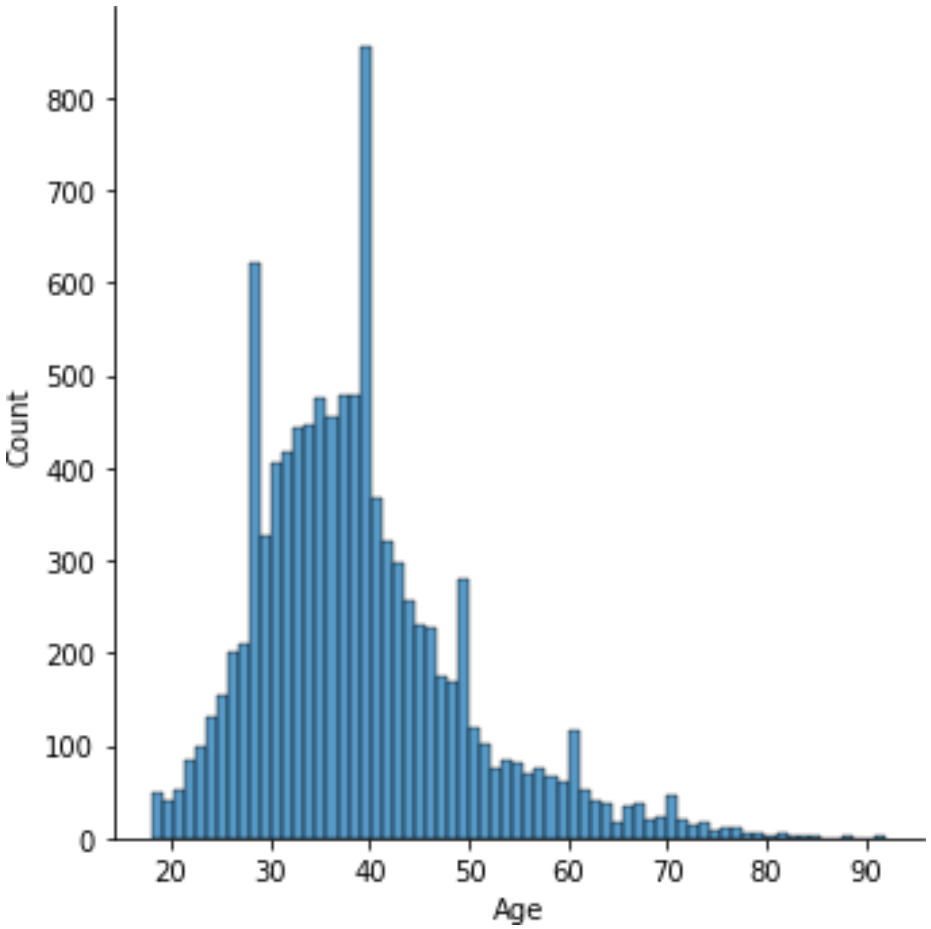
```
2692          17550.49       0
7031          44147.95       1
2158         156048.64       0
3549         166305.16       1
3772          31802.92       0
5328         166317.71       0
```

Perform Below Visualizations
Univariate Analysis

```
features =['Age', 'CreditScore', 'Balance']
df[features].hist(figsize=(13, 10));
```



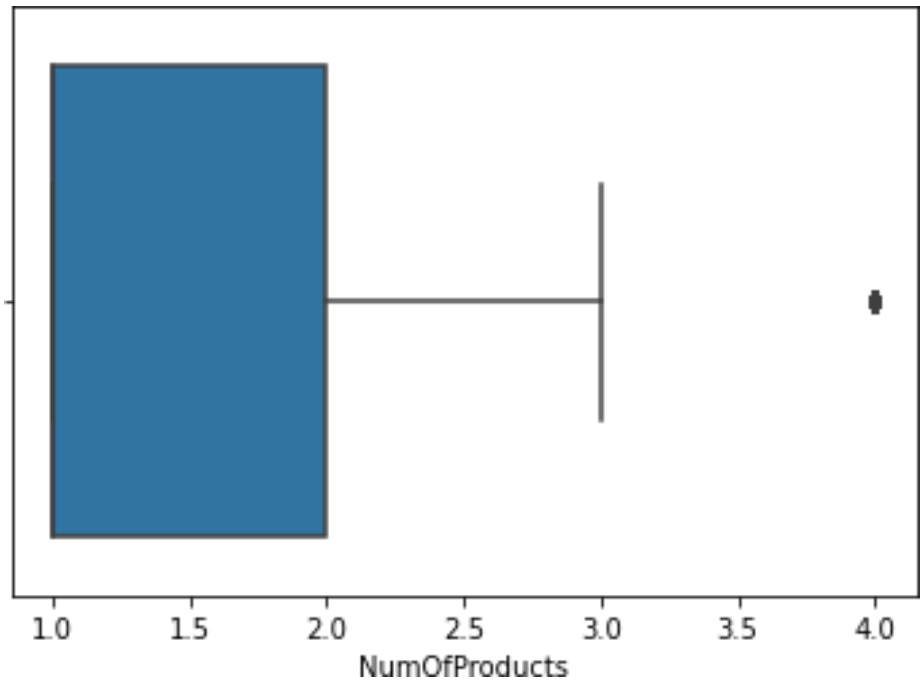

```
import seaborn as sns

sns.displot(df["Age"])

<seaborn.axisgrid.FacetGrid at 0x7fc07c40a350>
```




```
sns.boxplot(df["NumOfProducts"])
```

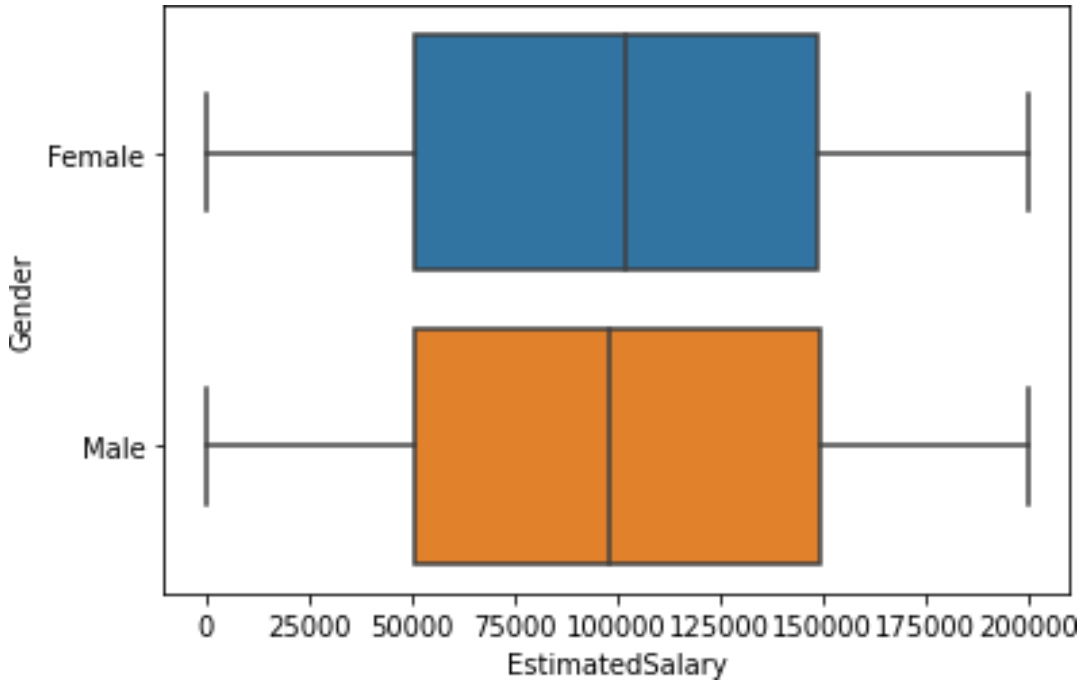
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0889c6a90>
```

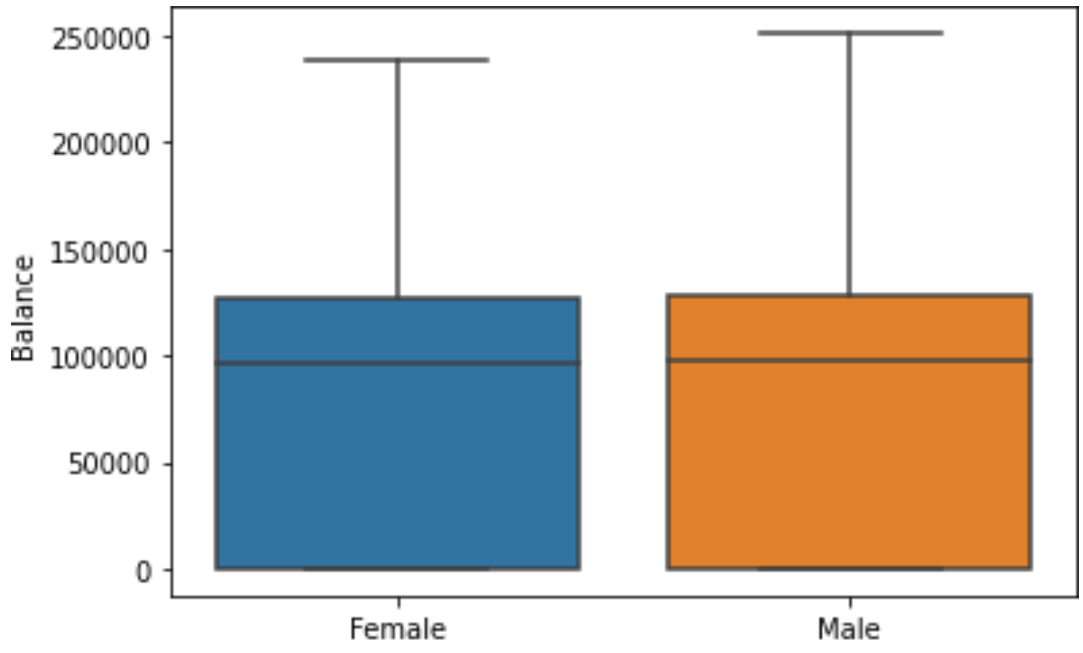


Bivariate Analysis

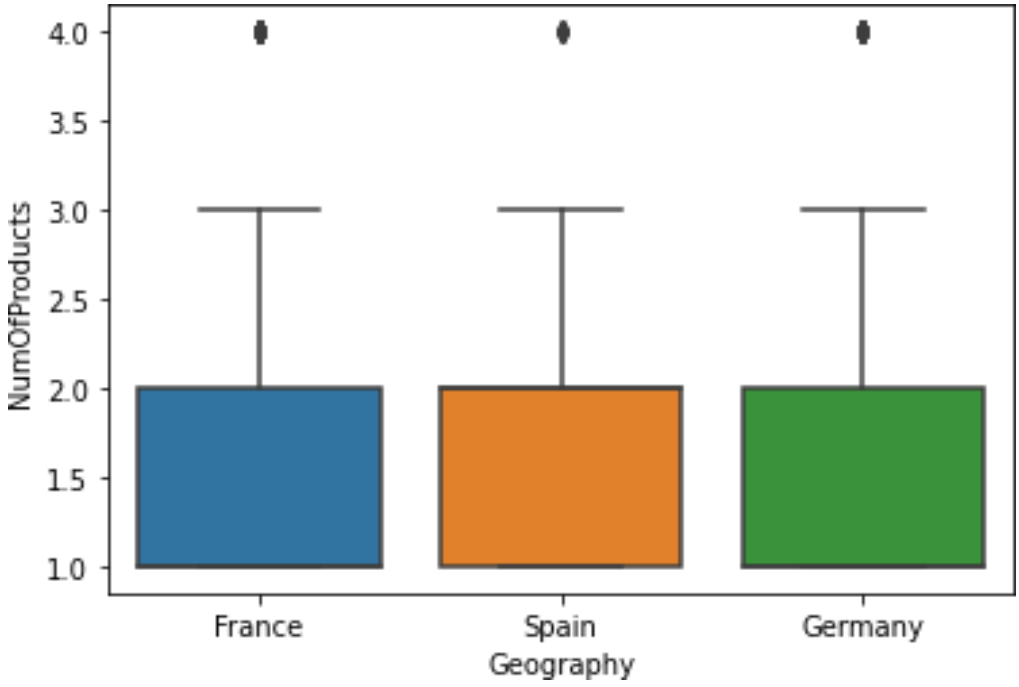
```
import seaborn as sns
sns.boxplot(x = df['EstimatedSalary'], y = df['Gender'] );
```




```
sns.boxplot(x=df['Gender'],y=df['Balance']);
```

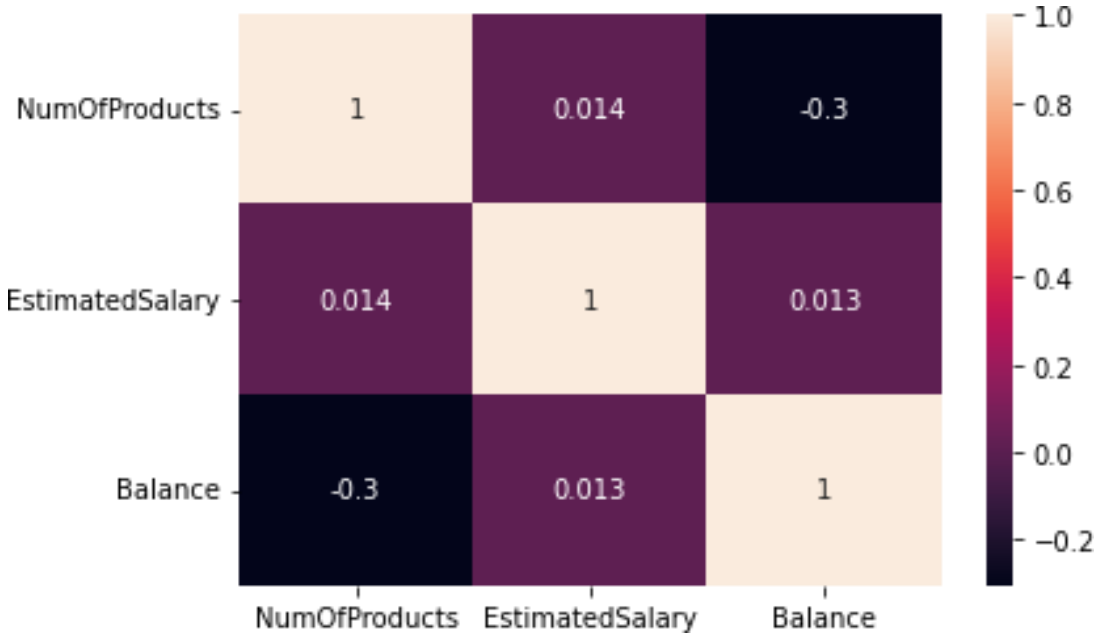



```
sns.boxplot(x=df['Geography'],y=df['NumOfProducts']);
```



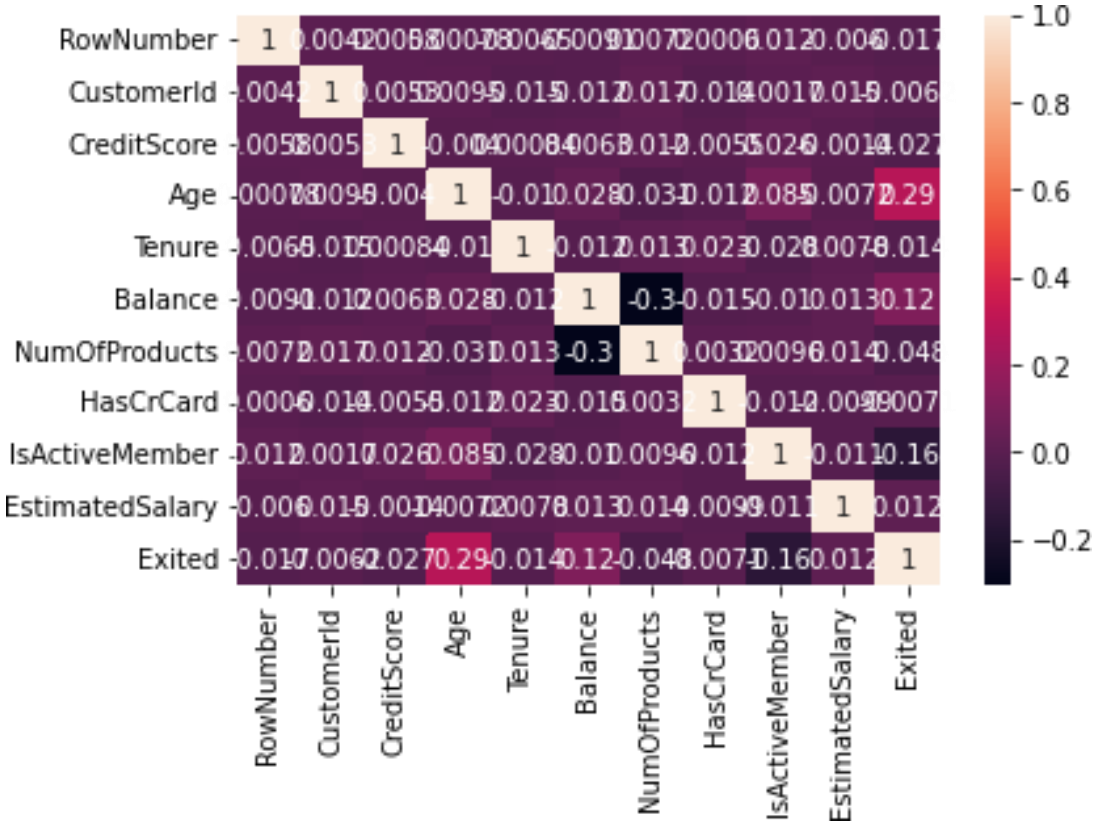
Multivariate Analysis

```
df_1 =
pd.DataFrame(df,columns=['NumOfProducts','EstimatedSalary','Balance'])
corrMatrix = df_1.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```




```
sns.heatmap(df.corr(),annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc079668750>
```



4. Perform descriptive statistics on the dataset.

```
df.describe(include=['object'])
```

```
       Surname Geography Gender
count    10000     10000 10000
unique    2932         3       2
top      Smith    France Male
freq        32        5014 5457
df['CreditScore'].value_counts()
df['CreditScore'].value_counts().to_frame()
df['Geography'].value_counts()
```

```
France     5014
Germany    2509
Spain    2477 Name:
Geography, dtype: int64
```

```
geography_counts=df['Geography'].value_counts().to_frame()
geography_counts.rename(columns={'Geography':'value_counts'},inplace=T
rue) geography_counts
```

```
         value_counts
France   5014
```

```
Germany          2509
Spain            2477
```

5. Handle the Missing values.

```
df.shape      (10000,      14)
```

```
df.isnull()
```

```
            RowNumber CustomerId Surname CreditScore Geography Gender
 Age \
0         False       False    False        False      False   False
False
1         False       False    False        False      False   False
False
2         False       False    False        False      False   False
False
3         False       False    False        False      False   False
False
4         False       False    False        False      False   False
False
...         ...         ...      ...          ...        ...     ...
...
9995      False       False    False        False      False   False
False
9996      False       False    False        False      False   False
False
9997      False       False    False        False      False   False
False
9998      False       False    False        False      False   False
False
9999      False       False    False        False      False   False
False


      Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0  False  False     False      False      False
1  False  False     False      False      False
2  False  False     False      False      False
3  False  False     False      False      False
4  False  False  False False False  ... ...  ...  ... ... ...
9995     False  False     False      False      False
9996     False  False     False      False      False
9997     False  False     False      False      False
```

```
9998     False False     False      False      False 9999 False  False
   False      False      False

       EstimatedSalary Exited
0  False False
1  False False
2  False False
3  False False
4  False False ... ...  ...
9995            False False
9996            False False
9997            False False
9998            False False
9999            False False

[10000 rows x 14 columns]
```

```
df.notnull()
```

```
         RowNumber CustomerId Surname CreditScore Geography Gender
 Age \
0           True     True  True  True  True  True
True
1           True     True  True  True  True  True
True
2           True     True  True  True  True  True
True
3           True     True  True  True  True  True
True
4           True     True  True  True  True  True
True
...          ...          ...      ...           ...         ...     ...
...
9995        True     True  True  True  True  True
True
9996        True     True  True  True  True  True
True
9997        True     True  True  True  True  True
True
9998        True     True  True  True  True  True
True
9999        True     True  True  True  True  True
True

      Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0  True  True  True  True  True
```

```
1  True  True  True  True  True
2  True  True  True  True  True
3  True  True  True  True  True
4  True  True  True True True  ... ...  ...  ... ... ...
9995    True  True  True  True  True
9996    True  True  True  True  True
9997    True  True  True  True  True
9998    True  True  True  True  True
9999    True  True  True  True  True


      EstimatedSalary Exited
0  True True
1  True True
2  True True
3  True True
4  True True  ...  ... ...
9995             True True
9996             True True
9997             True True
9998             True True
9999             True True

[10000 rows x 14 columns]
```

```
df.fillna(df.mean())
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError. Select only valid columns before calling the
reduction.
  """Entry point for launching an IPython kernel.


       RowNumber CustomerId        Surname CreditScore Geography Gender
 Age \
0              1    15634602 Hargrave     619   France Female
42
1              2    15647311   Hill  608   Spain Female
41
2              3    15619304   Onio  502   France Female
42
3              4    15701354   Boni  699   France Female
39
4              5    15737888 Mitchell     850   Spain Female
43
...          ...         ...        ...          ...       ...      ...
...
```

```
9995     9996  15606229 Obijiaku     771   France     Male
39
9996     9997  15569892 Johnstone    516   France     Male
35
9997     9998  15584532   Liu   709   France Female
36
9998     9999  15682355 Sabbatini    772 Germany     Male
42
9999     10000      15628319   Walker     792   France Female 28


      Tenure      Balance NumOfProducts HasCrCard IsActiveMember \
0           2  0.00  1     1     1
1           1 83807.86    1     0     1
2           8 159660.80   3     1     0
3           1 0.00 2 0 0  4 2 125510.82 1 1 1
... ... ... ... ... ...
9995     5     0.00  2     1     0
9996     10 57369.61      1     1     1
9997     7     0.00  1     0     1
9998     3 75075.31 2     1     0  9999     4 130142.79     1     1     0


      EstimatedSalary Exited
0  101348.88  1
1  112542.58  0
2  113931.57  1
3  93826.63   0
4  79084.10 0  ...  ... ...
9995         96270.64    0
9996         101699.77   0
9997         42085.58    1
9998         92888.52    1
9999         38190.78    0

[10000 rows x 14 columns]

df.fillna(df.median())

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError. Select only valid columns before calling the
reduction.
  """Entry point for launching an IPython kernel.

       RowNumber CustomerId        Surname CreditScore Geography Gender
Age \
```

```
0             1    15634602   Hargrave          619    France Female
42
1             2    15647311       Hill          608     Spain Female
41
2             3    15619304       Onio          502    France Female
42
3             4    15701354       Boni          699    France Female
39
4             5    15737888   Mitchell          850     Spain Female
43
...         ...         ...        ...          ...       ...      ...
...
9995       9996    15606229 Obijiaku     771   France     Male
39
9996       9997    15569892 Johnstone    516   France Male
35
9997       9998    15584532   Liu   709   France Female
36
9998       9999    15682355 Sabbatini    772 Germany Male
42
9999       10000   15628319   Walker     792   France Female
28

             Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0          2  0.00  1     1     1
1          1 83807.86    1     0     1
2          8 159660.80   3     1     0
3          1  0.00  2 0  0  4 2 125510.82  1 1  1
... ...  ..... ...  ...
9995     5     0.00  2     1     0
9996     10 57369.61       1     1     1
9997     7     0.00  1     0     1
9998     3 75075.31        2     1     0  9999     4 130142.79     1     1
   0


      EstimatedSalary Exited
0            101348.88   1
1            112542.58   0
2            113931.57   1
3            93826.63    0
4            79084.10  0
...  ... ...
9995         96270.64    0
9996         101699.77   0
9997         42085.58    1
9998         92888.52    1
9999         38190.78    0
```

```
[10000 rows x 14 columns]
```

```
df.isnull().sum
```

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of
   RowNumber CustomerId Surname CreditScore Geography Gender        Age
\
0  False      False False     False      False False
False
1  False False False False False False False
2  False      False      False      False      False False
False
3  False      False      False      False      False False
False
4  False      False      False      False      False False
False
...         ...         ...      ...          ...        ...     ...
...
9995      False     False      False      False      False False
False
9996      False     False      False      False      False False
False
9997      False     False      False      False      False False
False
9998      False     False      False      False      False False
False
9999      False     False      False      False      False False
False

       Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0       False  False      False      False      False
1       False  False      False      False      False
2       False  False      False      False      False
3       False  False      False      False      False
4       False  False      False      False      False
...       ...      ...            ...        ...              ...
9995    False  False      False      False      False
9996    False  False      False      False      False
9997    False  False      False      False      False
9998    False  False      False      False      False
9999    False  False      False      False      False

       EstimatedSalary Exited
0  False False
1  False False
2  False False
```

```
3  False False
4  False False ... ...  ...
9995            False False
9996            False False
9997            False False
9998            False False
9999            False False


[10000 rows x 14 columns]>
```

```
df[df.CreditScore.isnull()] Empty DataFrame
```

```
Columns: [RowNumber, CustomerId, Surname, CreditScore, Geography,
Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard,
IsActiveMember, EstimatedSalary,
```

```
Exited] Index: []
```

```
df.dropna(how='any').shape
```

```
(10000, 14)
```

```
df.dropna(subset=['CreditScore','Tenure'],how='any').shape
```

```
(10000, 14)
```

```
df.dropna(subset=['CreditScore','Tenure'],how='any')
```

```
      RowNumber CustomerId        Surname CreditScore Geography Gender
 Age \
0             1    15634602 Hargrave     619 France Female
42
1             2    15647311   Hill  608   Spain Female
41
2             3    15619304   Onio  502 France Female
42
3             4    15701354   Boni  699 France Female
39
4             5    15737888 Mitchell     850   Spain Female
43
...         ...         ...        ...           ...       ...      ...
...
9995       9996    15606229 Obijiaku     771 France      Male
39
9996       9997    15569892 Johnstone    516 France Male
35
9997       9998    15584532   Liu   709 France Female
36
9998       9999    15682355 Sabbatini    772 Germany Male
42
```

```
9999       10000    15628319   Walker     792 France Female
28

             Tenure Balance NumOfProducts HasCrCard IsActiveMember \
0           2  0.00  1     1     1
1           1  83807.86    1     0     1
2           8 159660.80   3     1     0
3           1  0.00  2 0  0  4 2 125510.82  1 1  1
... ...  ...  ... ... ...  ...
9995     5     0.00  2     1     0
9996     10  57369.61     1     1     1
9997     7     0.00  1     0     1
9998     3 75075.31 2     1     0 9999     4 130142.79      1     1     0

      EstimatedSalary Exited
0  101348.88  1
1  112542.58  0
2  113931.57  1
3  93826.63   0
4  79084.10 0  ...  ... ...
9995         96270.64    0
9996         101699.77   0
9997         42085.58    1
9998         92888.52    1
9999         38190.78    0

[10000 rows x 14 columns]
```

```
df.dropna(subset=['CreditScore','Tenure'],how='all').shape
```

```
(10000, 14)
```

```
df.dropna(subset=['CreditScore','Tenure'],how='all')
```

```
       RowNumber CustomerId        Surname CreditScore Geography Gender
 Age \
0              1 15634602 Hargrave   619 France Female
42
1              2 15647311      Hill  608   Spain Female
41
2              3 15619304      Onio  502 France Female
42
3              4 15701354      Boni  699 France Female
39
4              5 15737888 Mitchell   850   Spain Female
43
...          ...         ...        ...          ...       ...      ...
```

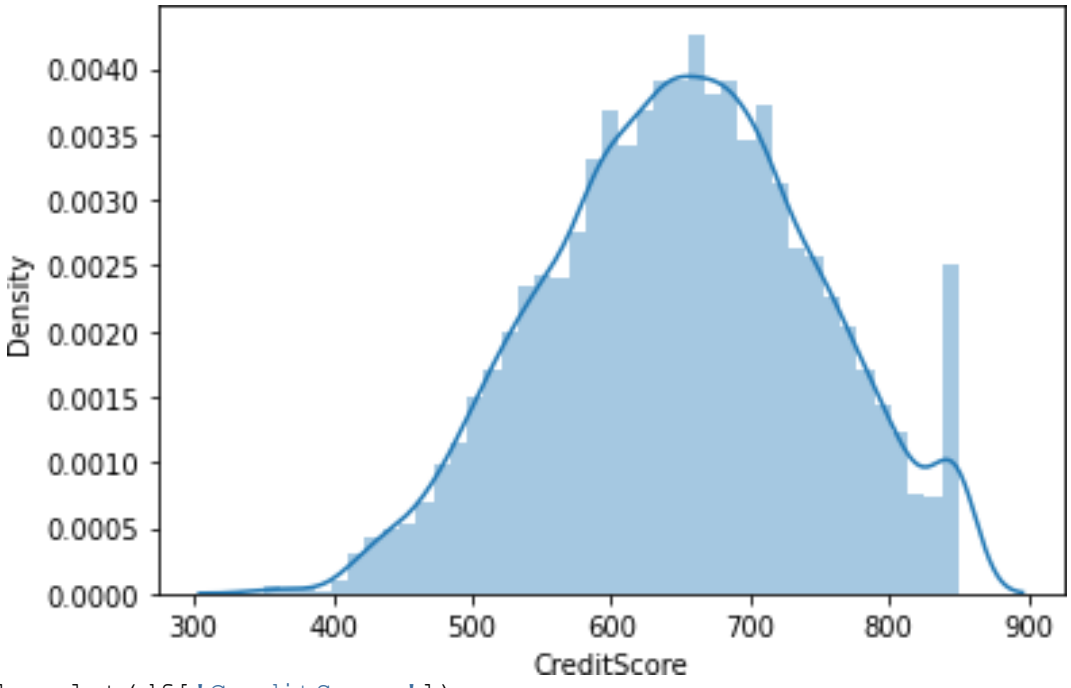
```
...
9995       9996    15606229 Obijiaku     771 France       Male
39
9996       9997 15569892 Johnstone  516 France Male
35
9997       9998 15584532       Liu   709 France Female
36
9998       9999 15682355 Sabbatini  772 Germany Male
42
9999       10000 15628319     Walker     792 France Female
28


      Tenure  Balance NumOfProducts HasCrCard IsActiveMember \ 0     2
   0.00  1     1     1
1          1 83807.86    1     0     1
2          8 159660.80   3     1     0
3          1 0.00 2 0 0 4 2 125510.82 1 1 1
... ... ... ... ... ...
9995     5     0.00  2     1     0
9996     10 57369.61      1     1     1
9997     7     0.00  1     0     1
9998     3 75075.31 2     1     0 9999     4 130142.79      1     1     0


      EstimatedSalary Exited
0            101348.88   1
1            112542.58   0
2            113931.57   1
3            93826.63    0
4            79084.10 0
...  ... ...
9995         96270.64    0
9996         101699.77   0
9997         42085.58    1
9998         92888.52    1
9999         38190.78    0


[10000 rows x 14 columns]
```

6. Find the outliers **and** replace the outliers

```
sns.distplot(df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms). warnings.warn(msg, FutureWarning)
```

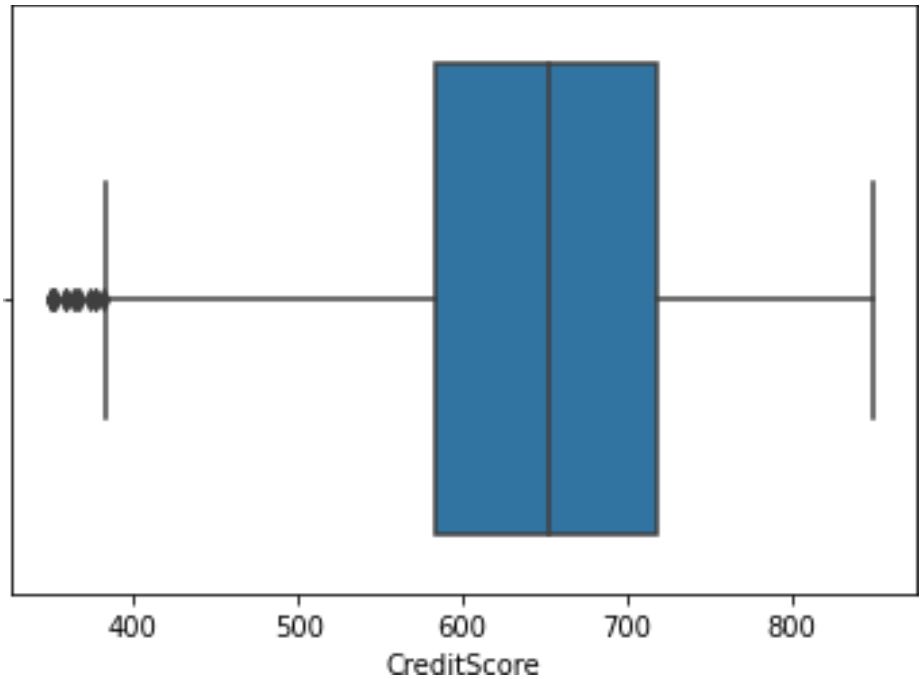
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0797203d0>
```




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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc07989acd0>
```




```
upper_limit = df['CreditScore'].mean() + 3*df['CreditScore'].std()
lower_limit = df['CreditScore'].mean() - 3*df['CreditScore'].std()
```

```
print('upper limit:', upper_limit) print('lower limit:',
lower_limit)
```

```
upper limit: 940.488696208391
lower limit:
360.568903791609
```

```
df.loc[(df['CreditScore'] > upper_limit) | (df['CreditScore'] <
lower_limit)]
```

```
      RowNumber CustomerId        Surname CreditScore Geography Gender
Age \
1405       1406     15612494 Panicucci           359      France Female
44
1631       1632     15685372 Azubuike            350     Spain     Male
54
1838       1839     15758813 Campbell            350 Germany      Male
39
1962       1963     15692416 Aikenhead           358       Spain Female
52
2473       2474    15679249       Chou              351 Germany Female
57
8723       8724     15803202 Onyekachi           350    France     Male
51
8762       8763    15765173        Lin           350      France Female
60
9624       9625    15668309     Maslow           350      France Female
40

              Tenure Balance NumOfProducts HasCrCard IsActiveMember \
1405        6 128747.69              1          1               0
1631        1 152677.48              1          1               1
1838        0 109733.20              2          0               0
1962        8 143542.36              3          1               0
2473        4 163146.46              1          1               0
8723      10       0.00              1          1               1
8762     3     0.00  1     0     0  9624     0 111098.85      1     1     1

      EstimatedSalary Exited
1405        146955.71       1
1631        191973.49       1
1838        123602.11       1
1962        141959.11       1
2473        169621.69       1
8723        125823.79       1
```

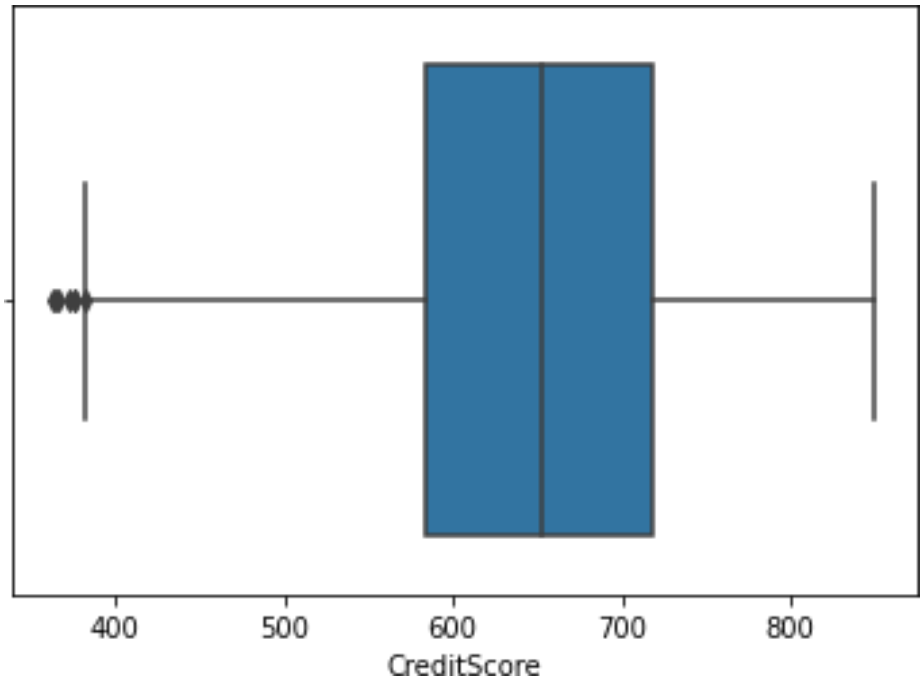
```
8762        113796.15       1
9624        172321.21       1
```

```
new_df = df.loc[(df['CreditScore'] <= upper_limit) &
(df['CreditScore'] >= lower_limit)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(new_df))
print('outliers:', len(df)-len(new_df))
```

```
before removing outliers: 10000
after removing outliers: 9992
outliers: 8
sns.boxplot(new_df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

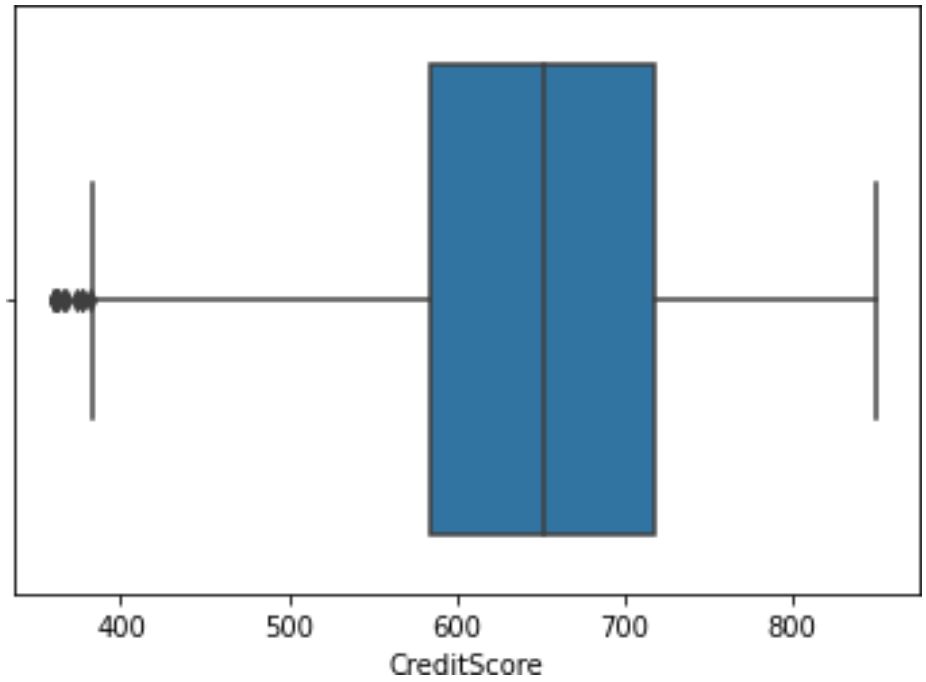
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0797e5310>
```




```
new_df = df.copy()
new_df.loc[(new_df['CreditScore']>=upper_limit), 'CreditScore'] =
upper_limit
new_df.loc[(new_df['CreditScore']<=lower_limit), 'CreditScore']
= lower_limit sns.boxplot(new_df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077c76a50>
```



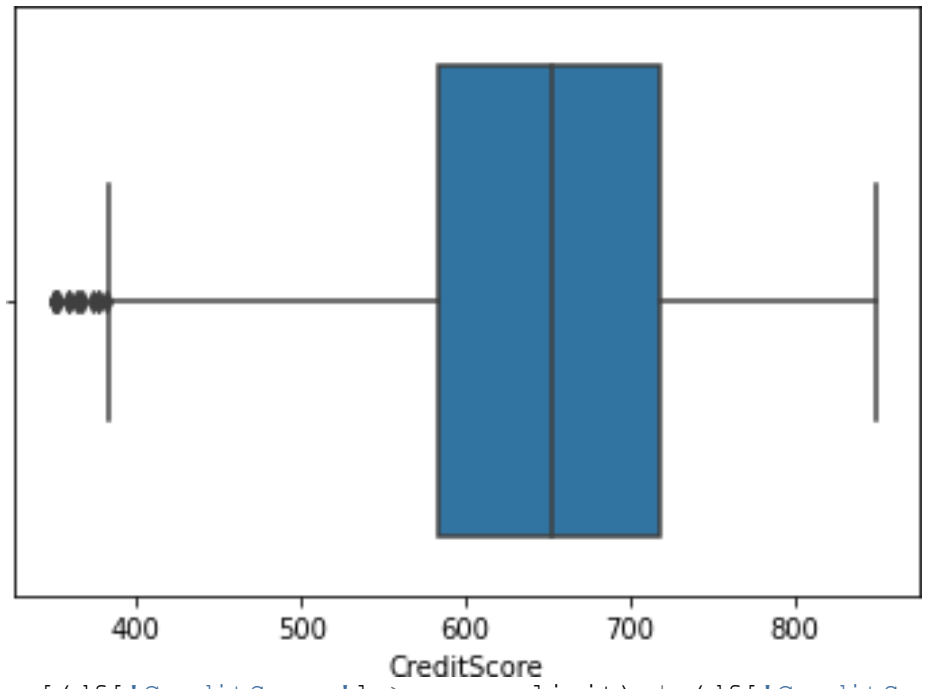

```
upper_limit = df['CreditScore'].quantile(0.99)
lower_limit = df['CreditScore'].quantile(0.01)
print('upper limit:', upper_limit)
print('lower limit:', lower_limit)
```

```
upper limit: 850.0 lower
limit: 432.0
```

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077c4bd90>
```




```
df.loc[(df['CreditScore'] > upper_limit) | (df['CreditScore'] <
 lower_limit)]
```

```
      RowNumber CustomerId        Surname CreditScore Geography Gender
 Age \
7             8    15656148      Obinna             376 Germany Female
29
29           30    15656300    Lucciano           411    France    Male
29
79           80    15803136      Postle             416 Germany Female
41
99          100    15633059     Fanucci           413    France    Male
34
149         150    15794413      Harris           416    France    Male
32
...         ...         ...         ...           ...       ...     ...
...
9357       9358     15814405 Chesnokova           418     France Female
46
9407       9408    15652835       Liang           419      Spain Female
27
9522       9523    15664504       Beede           418    France    Male
35
9624       9625    15668309      Maslow           350     France Female
40
9930       9931    15713604       Rossi           425 Germany     Male
40

      Tenure      Balance NumOfProducts HasCrCard IsActiveMember \
```

```
7           4 115046.74               4           1                0
29          0 59697.17                2           1                1
79         10 122189.66               2           1                0
99 9     0.00  2     0     0 149      0     0.00  2     0     1
...       ...         ...            ...         ...              ...
9357        9        0.00              1           1                1
9407        2 121580.42                1           0                1
9522        7        0.00              2           1                1
9624        0 111098.85                1           1                1
9930        9 166776.60                2           0                1


      EstimatedSalary Exited
7           119346.88       1
29           53483.21       0
79           98301.61       0
99            6534.18       0
149  878.87 0  ... ... ...
9357         81014.50       1
9407        134720.51       0
9522         88878.15       0
9624        172321.21       1
9930        172646.88       0


[99 rows x 14 columns]
```

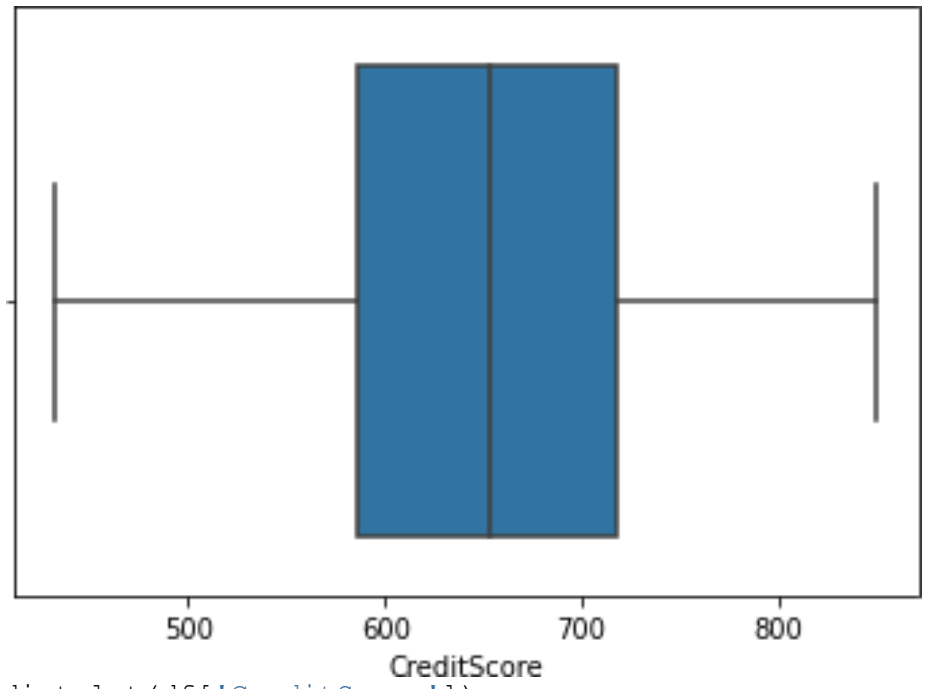
```
new_df = df.loc[(df['CreditScore'] <= upper_limit) &
(df['CreditScore'] >= lower_limit)]
print('before removing outliers:', len(df))
print('after removing outliers:',len(new_df))
print('outliers:', len(df)-len(new_df))
```

```
before removing outliers: 10000
after removing outliers: 9901
outliers: 99
```

```
sns.boxplot(new_df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in
an error or misinterpretation. FutureWarning
```

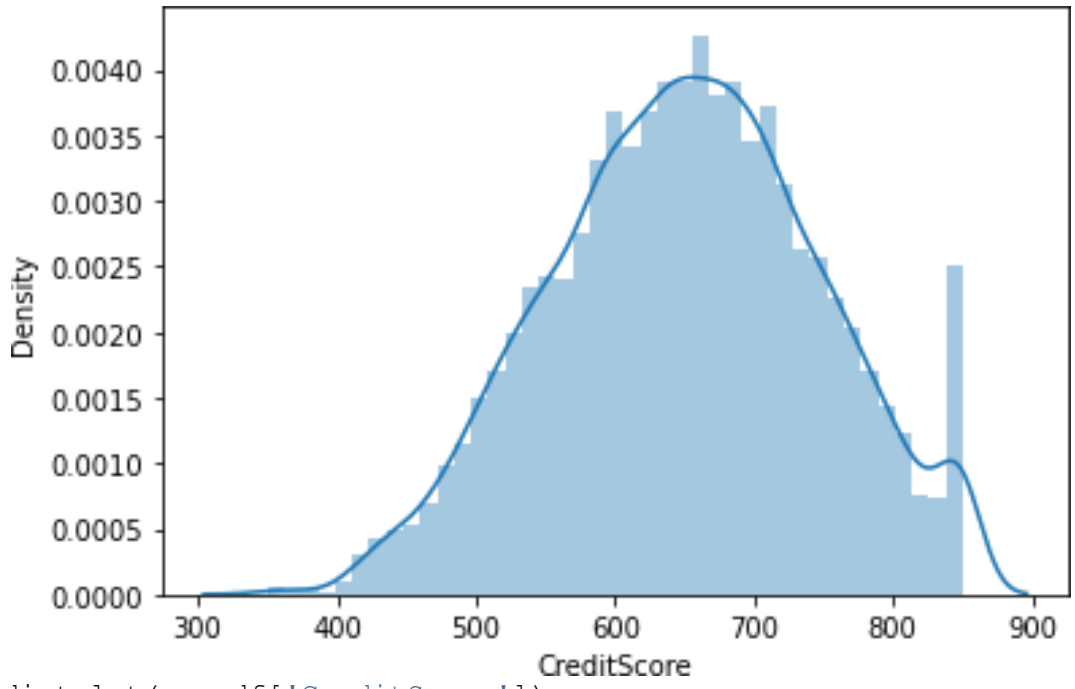
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077bc8550>
```




```
sns.distplot(df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077b2d510>
```

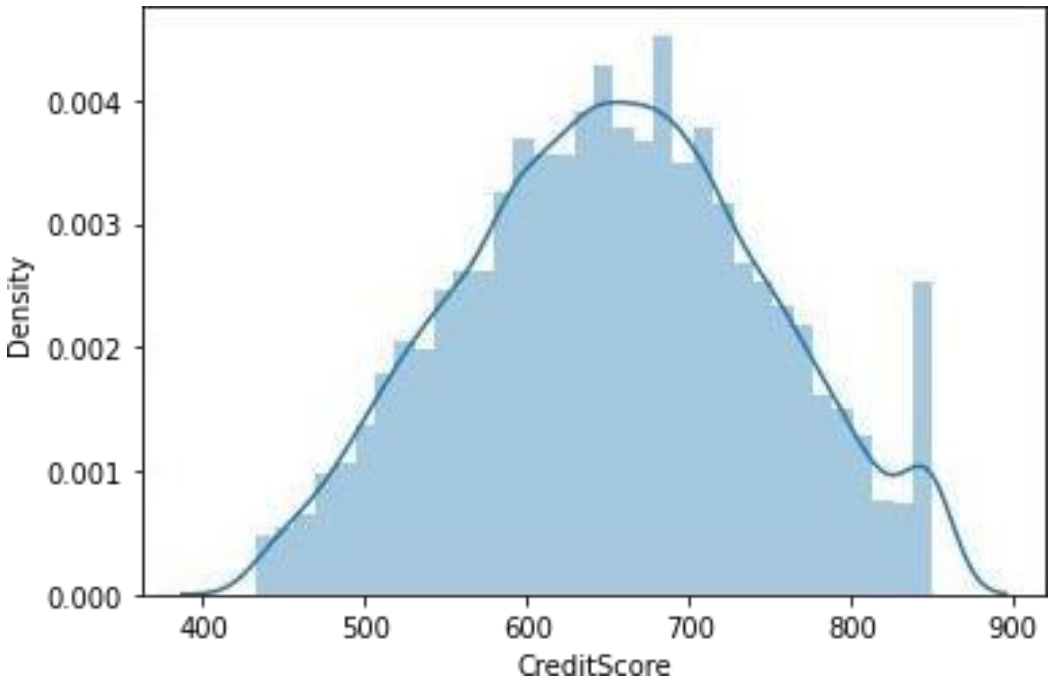



```
sns.distplot(new_df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
```

```
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077c61990>
```



7.)Check for Categorical columns and perform encoding.

```
df=df.iloc[:,:].values
df
```

```
array([[1, 15634602, 'Hargrave', ..., 1, 101348.88, 1],
       [2, 15647311, 'Hill', ..., 1, 112542.58, 0],
       [3, 15619304, 'Onio', ..., 0, 113931.57, 1],
       ...,
       [9998, 15584532, 'Liu', ..., 1, 42085.58, 1],
       [9999, 15682355, 'Sabbatini', ..., 0, 92888.52, 1],
       [10000, 15628319, 'Walker', ..., 0, 38190.78, 0]],
dtype=object)
```

8. Split the data into dependent and independent variables

```
url =
'https://drive.google.com/file/d/1_HcM0K8wt4b7FMLkc1V1dv0y6I_9ULzy/
view?usp=sharing' path = 'https://drive.google.com/uc?
export=download&id='+url.split('/')[-2] df = pd.read_csv(path)
```

```
x=df.iloc[:,4:7]
x
```

```
Geography Gender Age 0 France
 Female 42
1  Spain Female 41
```

```
2  France Female 42
3  France Female 39 4    Spain Female 43 ...   ...   ... ...
9995    France     Male 39
9996    France Male 35 9997 France Female 36
        9998 Germany Male 42 9999 France Female
        28


[10000 rows x 3 columns]
```

```
y=df.iloc[:,7]
y
```

```
0        2
1        1
2        8
3                                                              14
4 ..                                                           14
9995                                                           28
9996                                                           28

9997     7
9998     3
9999     4
Name: Tenure, Length: 10000, dtype: int64 9.
```

Scale the independent variables

```
import pandas as pd import
numpy as np import
matplotlib.pyplot as plt df
```

```
array([[1, 15634602, 'Hargrave', ..., 1, 101348.88, 1],
       [2, 15647311, 'Hill', ..., 1, 112542.58, 0],
       [3, 15619304, 'Onio', ..., 0, 113931.57, 1],
       ...,
       [9998, 15584532, 'Liu', ..., 1, 42085.58, 1],
       [9999, 15682355, 'Sabbatini', ..., 0, 92888.52, 1],
       [10000, 15628319, 'Walker', ..., 0, 38190.78, 0]],
dtype=object)
```

```
from sklearn.preprocessing import
scale x= scale(X) x
```

```
names=X.columns
names
```

10. Splitting the data into Training andTesting

```
x=np.array(df['CreditScore']).reshape(-1,1)
```

```
x.shape

(10000, 1)

print(x)

[[619]
 [608]
 [502]
 ...
 [709]
 [772]
 [792]]

y.shape

(10000,)

print(y)


0        2

1        1

2        8

3        1

4        2

          ..

9995     5

9996    10

9997     7

9998     3

9999     4

  Name: Tenure, Length: 10000, dtype: int64
from sklearn.model_selection import train_test_split
  x_train, x_test, y_train,

  y_test=train_test_split(x,y,test_size=0.30) x_train.shape (7000, 1)

  y_train.shape (7000,) y_test.shape (3000,) print(y_train.shape)

  (7000,)

  print(y_test.shape)
```

(3000,)