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

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

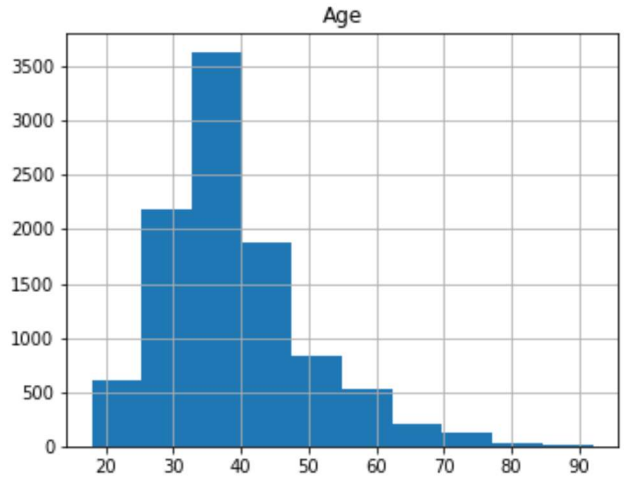
```
url = 'https://drive.google.com/file/d/1_HcM0K8wt4b7FMLkc1V1dv0y6I_9ULzy/view?usp=sharing'
path = 'https://drive.google.com/uc?export=download&id='+url.split('/')[-2]
df = pd.read_csv(path)
```

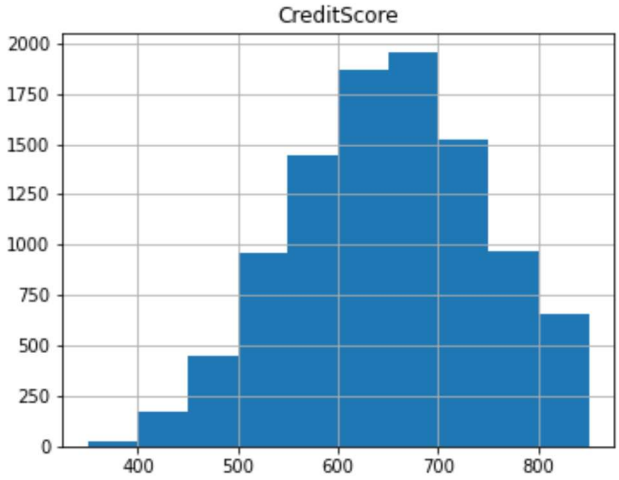
```
df.sample(20)
```

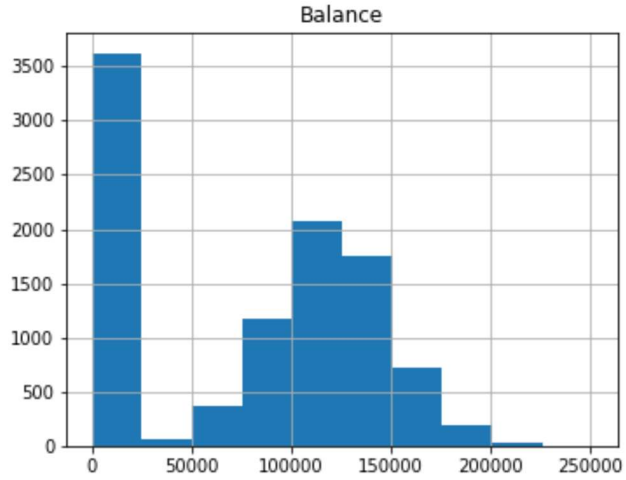
```
RowNumber  CustomerId    Surname  CreditScore  Geography  Gender  Age  Tenure
```

Perform Below Visualizations
Univariate Analysis

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



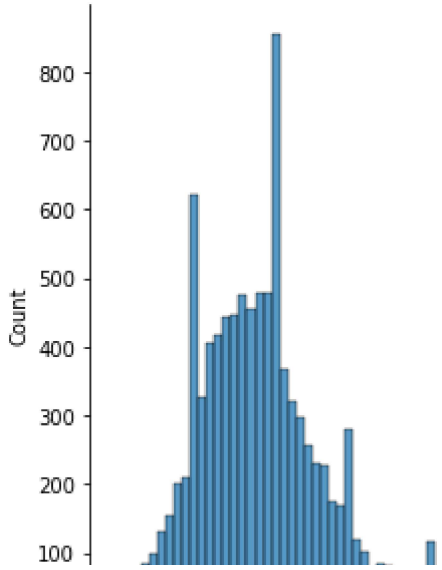





```
import seaborn as sns
```

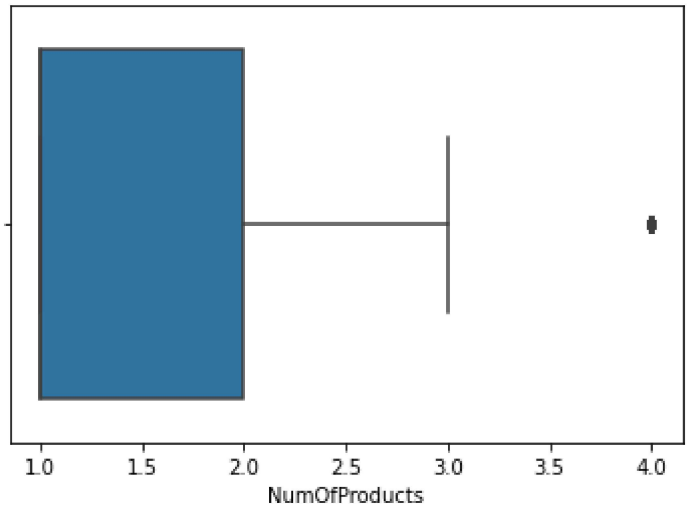
```
sns.displot(df["Age"])
```

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



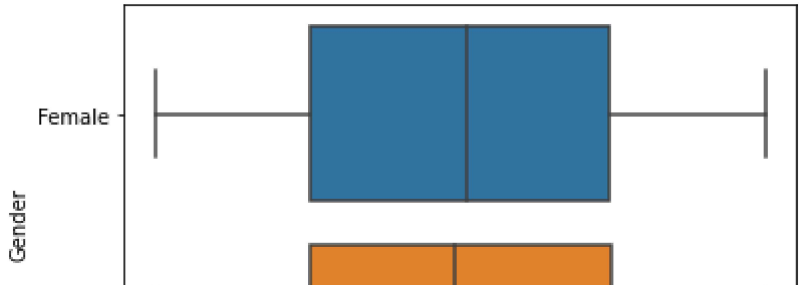

```
sns.boxplot(df["NumOfProducts"])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fc0889c6a90>
```

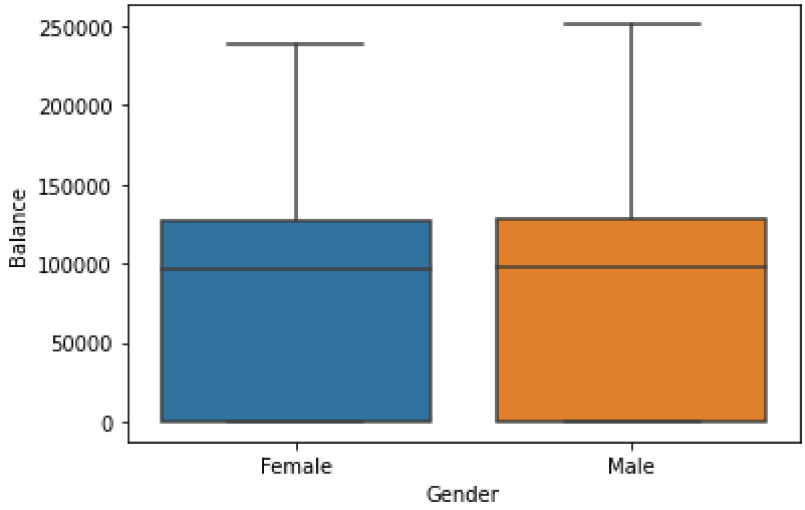


Bivariate Analysis

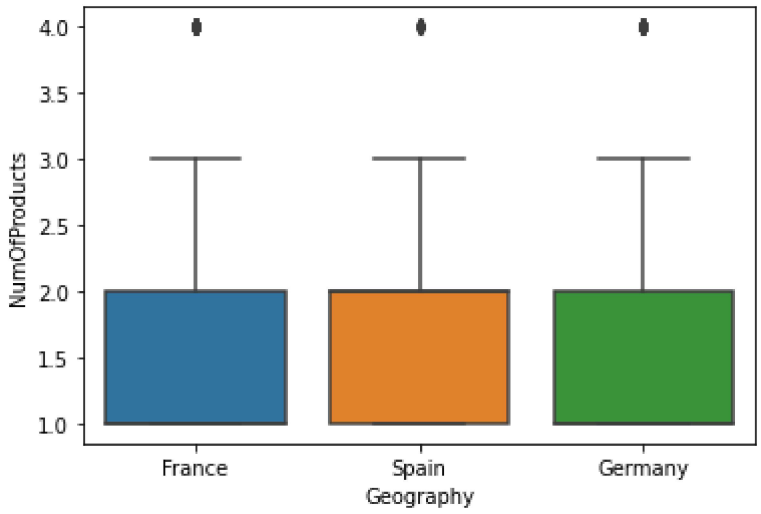
```
import seaborn as sns
sns.boxplot(x = df['EstimatedSalary'], y = df['Gender'] );
```




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

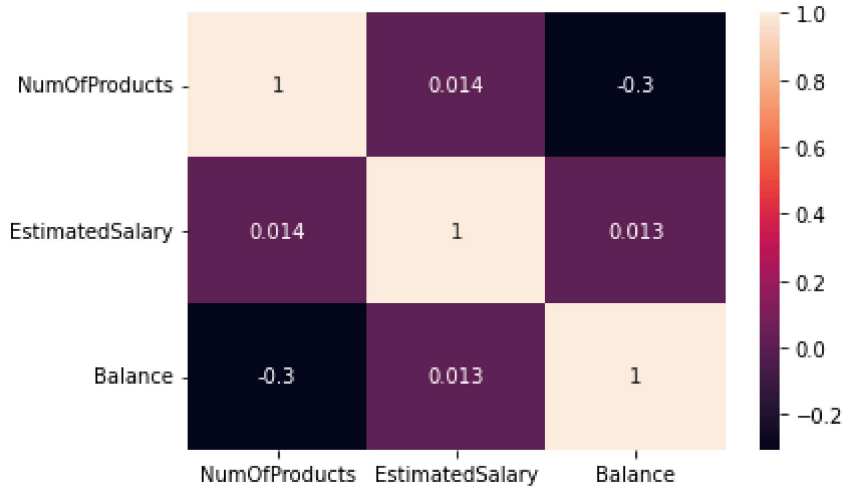



```
sns.boxplot(x=df['Geography'],y=df['NumOfProducts']);
```



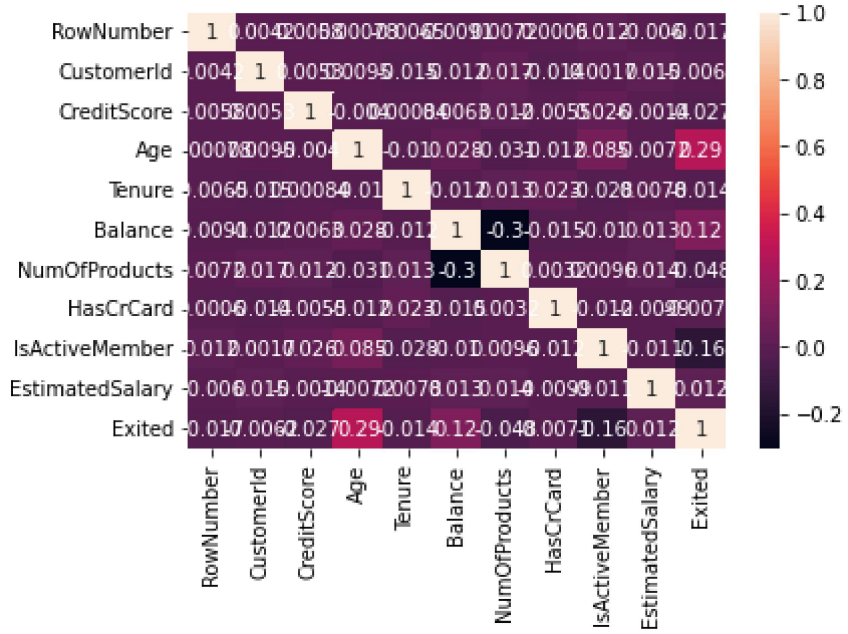
Multivariate Analysis

```
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=True)
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	Tenure	Bal
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	

```
df.notnull()
```

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

10000 rows × 14 columns

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping
  """Entry point for launching an IPython kernel.
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83

```
df.fillna(df.median())
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping
  """Entry point for launching an IPython kernel.
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75
9999	10000	15628319	Walker	792	France	Female	28	4	130

10000 rows × 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()]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance

```
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	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75
9999	10000	15628319	Walker	792	France	Female	28	4	130

10000 rows × 14 columns

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

```
(10000, 14)
```

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

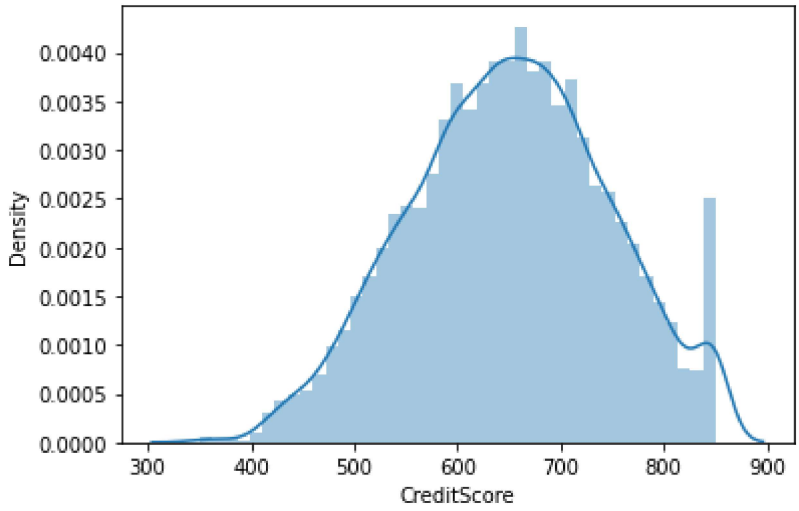
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Ba
0	1	15634602	Hargrave	619	France	Female	42	2	
1	2	15647311	Hill	608	Spain	Female	41	1	83
2	3	15619304	Onio	502	France	Female	42	8	159
3	4	15701354	Boni	699	France	Female	39	1	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125
...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57
9997	9998	15584532	Liu	709	France	Female	36	7	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75
9999	10000	15628319	Walker	792	France	Female	28	4	130

10000 rows × 14 columns

6. Find the outliers and replace the outliers

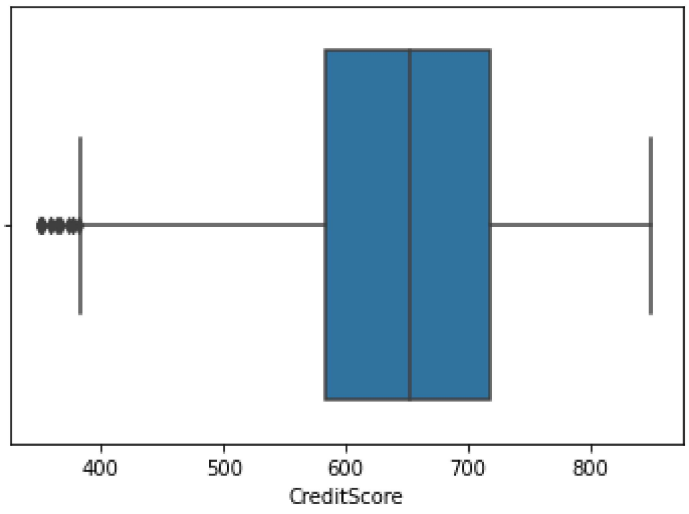
```
sns.distplot(df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  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 th
  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	Tenure	B
1405	1406	15612494	Panicucci	359	France	Female	44	6	128
1631	1632	15685372	Azubuike	350	Spain	Male	54	1	152
1838	1839	15758813	Campbell	350	Germany	Male	39	0	109
1962	1963	15692416	Aikenhead	358	Spain	Female	52	8	143
2473	2474	15679249	Chou	351	Germany	Female	57	4	163
8723	8724	15803202	Onyekachi	350	France	Male	51	10	
8762	8763	15765173	Lin	350	France	Female	60	3	
9624	9625	15668309	Maslow	350	France	Female	40	0	111

```
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))
```

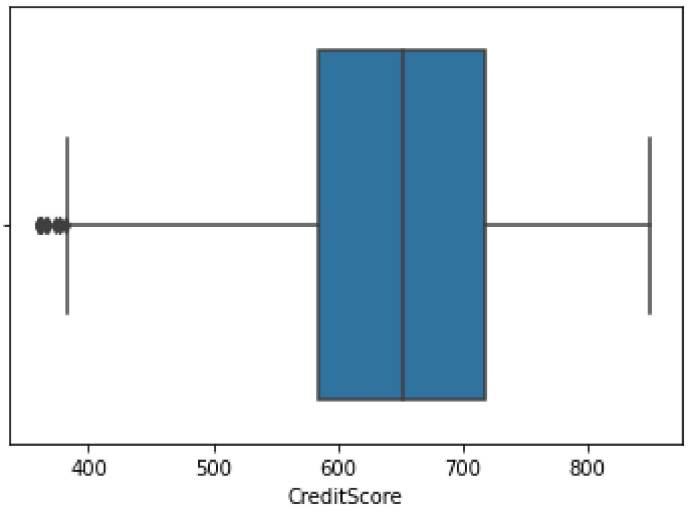
```
before removing outliers: 10000
after removing outliers: 9992
outliers: 8
```

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th
  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 th
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077c76a50>
```

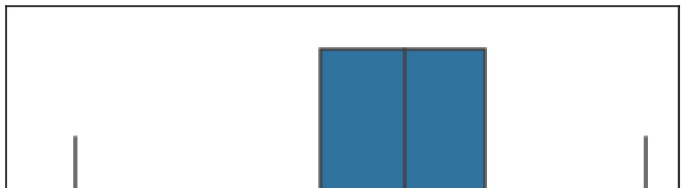



```
upper_limit = df['CreditScore'].quantile(0.99)
lower_limit = df['CreditScore'].quantile(0.01)
print('upper limit:', upper_limit)
print('lower limit:', lower_limit)
```

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



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

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	
7	8	15656148	Obinna	376	Germany	Female	29	4	1
29	30	15656300	Lucciano	411	France	Male	29	0	
79	80	15803136	Postle	416	Germany	Female	41	10	1
99	100	15633059	Fanucci	413	France	Male	34	9	
149	150	15794413	Harris	416	France	Male	32	0	
...	...	...	...	...	...	...	...	...	
9357	9358	15814405	Chesnokova	418	France	Female	46	9	
9407	9408	15652835	Liang	419	Spain	Female	27	2	1
9522	9523	15664504	Beede	418	France	Male	35	7	
9624	9625	15668309	Maslow	350	France	Female	40	0	1
9930	9931	15713604	Rossi	425	Germany	Male	40	9	1

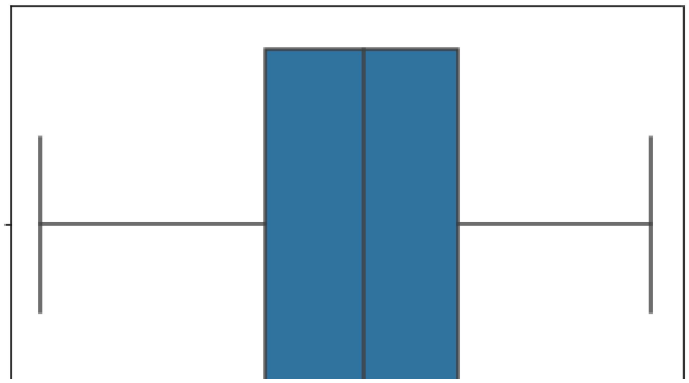
99 rows × 14 columns

```
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
```

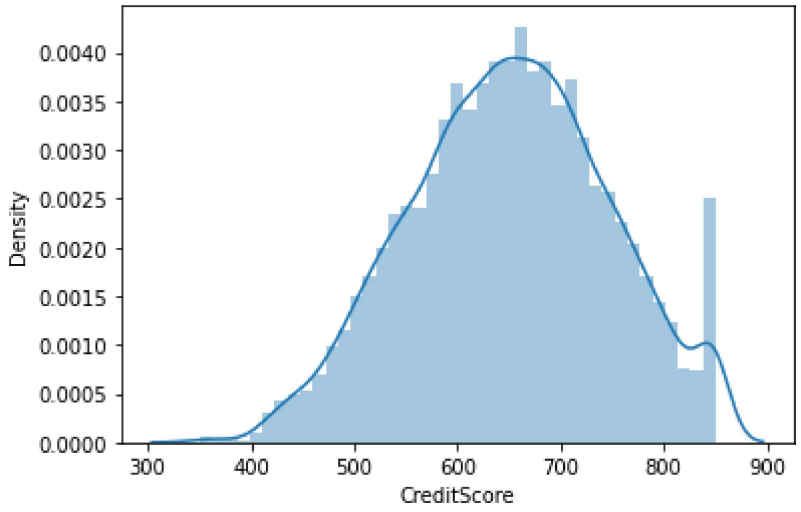
```
sns.boxplot(new_df['CreditScore'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass th
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fc077bc8550>
```



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

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `di
  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: `di
  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

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

9999	France	Female	28

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,)
```