

```

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn

data = pd.read_csv("Churn_Modelling.csv")

data.drop(['RowNumber'],axis=1,inplace=True)

data.head()

```

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	\
0	15634602	Hargrave	619	France	Female	42	2	
1	15647311	Hill	608	Spain	Female	41	1	
2	15619304	Onio	502	France	Female	42	8	
3	15701354	Boni	699	France	Female	39	1	
4	15737888	Mitchell	850	Spain	Female	43	2	

	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	0.00	1	1	1
1	83807.86	1	0	1
2	159660.80	3	1	0
3	0.00	2	0	0
4	125510.82	1	1	1

EstimatedSalary \

	Exited
0	1
1	0
2	1
3	0
4	0

```
data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   CustomerId            10000 non-null  int64
1   Surname                10000 non-null  object
2   CreditScore            10000 non-null  int64
3   Geography              10000 non-null  object
4   Gender                 10000 non-null  object

```

```
5   Age                10000 non-null   int64
6   Tenure              10000 non-null   int64
7   Balance             10000 non-null   float64
8   NumOfProducts       10000 non-null   int64
9   HasCrCard           10000 non-null   int64
10  IsActiveMember      10000 non-null   int64
11  EstimatedSalary     10000 non-null   float64
12  Exited               10000 non-null   int64
dtypes: float64(2), int64(8), object(3)
memory usage: 1015.8+ KB
```

```
data.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.
```

```
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
```

```
data.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.
```

```
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

data.mode()

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure
Balance \							
0	15565701	Smith	850.0	France	Male	37.0	2.0
0.0							
1	15565706	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
2	15565714	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
3	15565779	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
4	15565796	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
...
...							
9995	15815628	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
9996	15815645	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
9997	15815656	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
9998	15815660	NaN	NaN	NaN	NaN	NaN	NaN
NaN							
9999	15815690	NaN	NaN	NaN	NaN	NaN	NaN
NaN							

	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
Exited				
0	1.0	1.0	1.0	24924.92
0.0				
1	NaN	NaN	NaN	NaN
NaN				
2	NaN	NaN	NaN	NaN
NaN				
3	NaN	NaN	NaN	NaN
NaN				
4	NaN	NaN	NaN	NaN
NaN				
...
.				..
9995	NaN	NaN	NaN	NaN
NaN				
9996	NaN	NaN	NaN	NaN
NaN				
9997	NaN	NaN	NaN	NaN
NaN				

9998	NaN	NaN	NaN	NaN
NaN				
9999	NaN	NaN	NaN	NaN
NaN				

[10000 rows x 13 columns]

data.var()

/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.

CustomerId	5.174815e+09
CreditScore	9.341860e+03
Age	1.099941e+02
Tenure	8.364673e+00
Balance	3.893436e+09
NumOfProducts	3.383218e-01
HasCrCard	2.077905e-01
IsActiveMember	2.497970e-01
EstimatedSalary	3.307457e+09
Exited	1.622225e-01

dtype: float64

data.std()

/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.

CustomerId	71936.186123
CreditScore	96.653299
Age	10.487806
Tenure	2.892174
Balance	62397.405202
NumOfProducts	0.581654
HasCrCard	0.455840
IsActiveMember	0.499797
EstimatedSalary	57510.492818
Exited	0.402769

dtype: float64

data.describe()

	CustomerId	CreditScore	Age	Tenure
Balance \				
count	1.000000e+04	10000.000000	10000.000000	10000.000000
mean	1.569094e+07	650.528800	38.921800	5.012800
std	7.193619e+04	96.653299	10.487806	2.892174
min	1.556570e+07	350.000000	18.000000	0.000000
25%	1.562853e+07	584.000000	32.000000	3.000000
50%	1.569074e+07	652.000000	37.000000	5.000000
75%	1.575323e+07	718.000000	44.000000	7.000000
max	1.581569e+07	850.000000	92.000000	10.000000

	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary \
count	10000.000000	10000.000000	10000.000000	10000.000000
mean	1.530200	0.70550	0.515100	100090.239881
std	0.581654	0.45584	0.499797	57510.492818
min	1.000000	0.000000	0.000000	11.580000
25%	1.000000	0.000000	0.000000	51002.110000
50%	1.000000	1.000000	1.000000	100193.915000
75%	2.000000	1.000000	1.000000	149388.247500
max	4.000000	1.000000	1.000000	199992.480000

	Exited
count	10000.000000
mean	0.203700
std	0.402769
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
data['Geography'].unique()
```

```
array(['France', 'Spain', 'Germany'], dtype=object)
```

```
data['Age'].unique()
```

```
array([42, 41, 39, 43, 44, 50, 29, 27, 31, 24, 34, 25, 35, 45, 58, 32,
      46, 36, 33, 40, 51, 61, 49, 37, 19, 66, 56, 26, 21, 55, 75, 22,
      30, 28, 65, 48, 52, 57, 73, 47, 54, 72, 20, 67, 79, 62, 53, 80, 59,
      68,
```

```

23, 60, 70, 63, 64, 18, 82, 69, 74, 71, 76, 77, 88, 85, 84, 78,
81,
92, 83])

data['NumOfProducts'].unique()
array([1, 3, 2, 4])

data['Tenure'].unique()
array([ 2,  1,  8,  7,  4,  6,  3, 10,  5,  9,  0])

data['HasCrCard'].unique()
array([1, 0])

data['IsActiveMember'].unique()
array([1, 0])

data['Age'].value_counts()
37    478
38    477
35    474
36    456
34    447
...
92      2
82      1
88      1
85      1
83      1
Name: Age, Length: 70, dtype: int64

```

Data Visualization

uni variate analysis

```
sns.distplot(data['Age'])
```

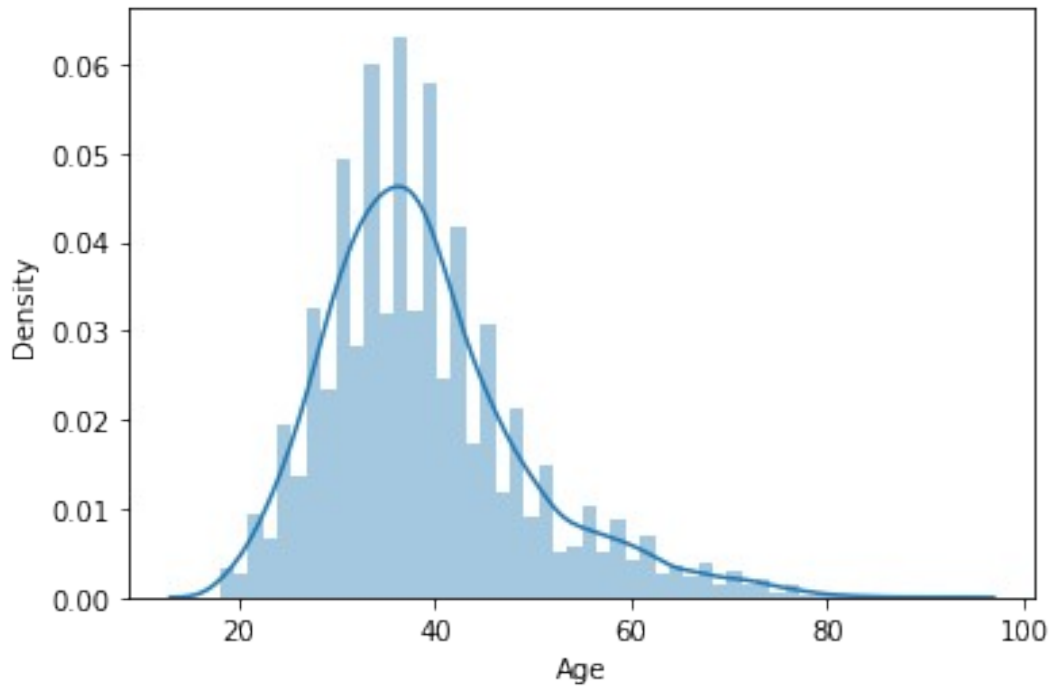
```

/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).

```

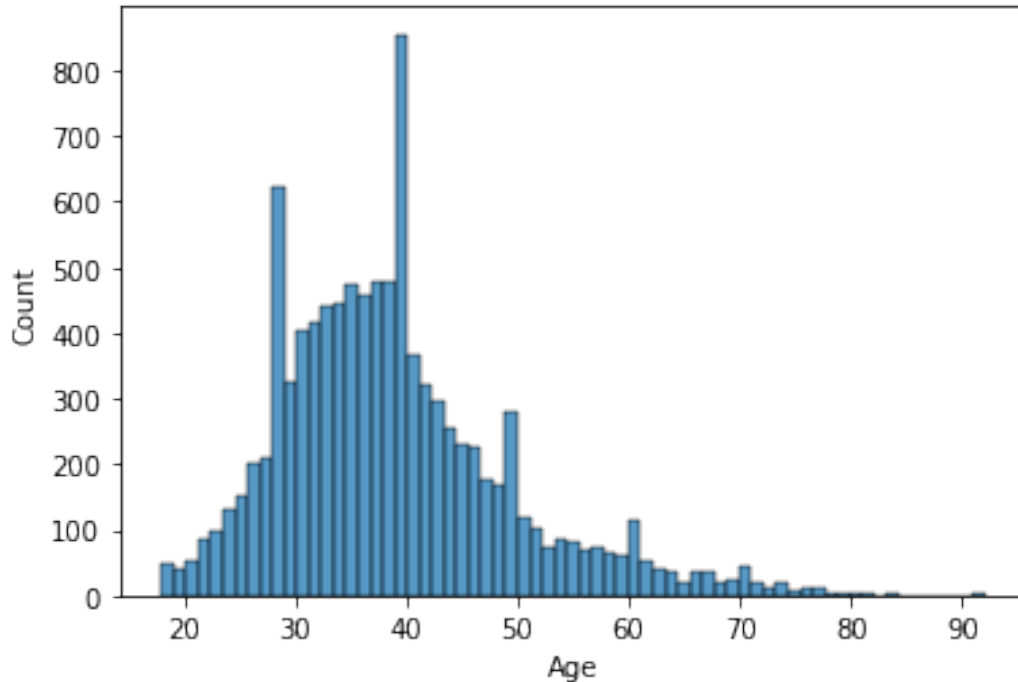
```
warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbae965cf10>
```



```
sns.histplot(data['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbae9543090>
```



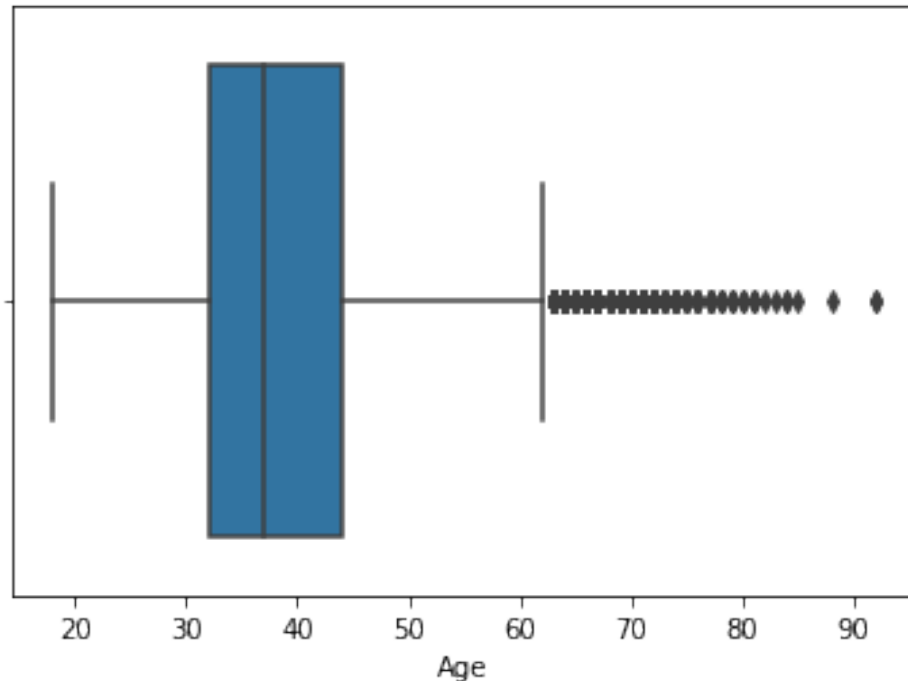
```
sns.boxplot(data['Age'])
```

```
/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 0x7fbae8eb9a90>



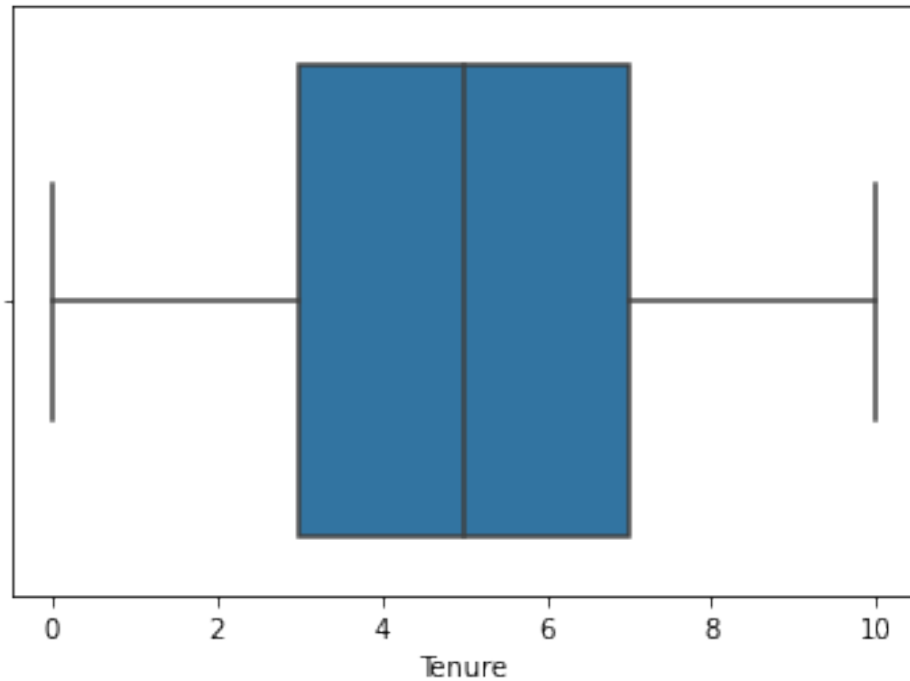
```
sns.boxplot(data['Tenure'])
```

/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 0x7fbae8e26e50>

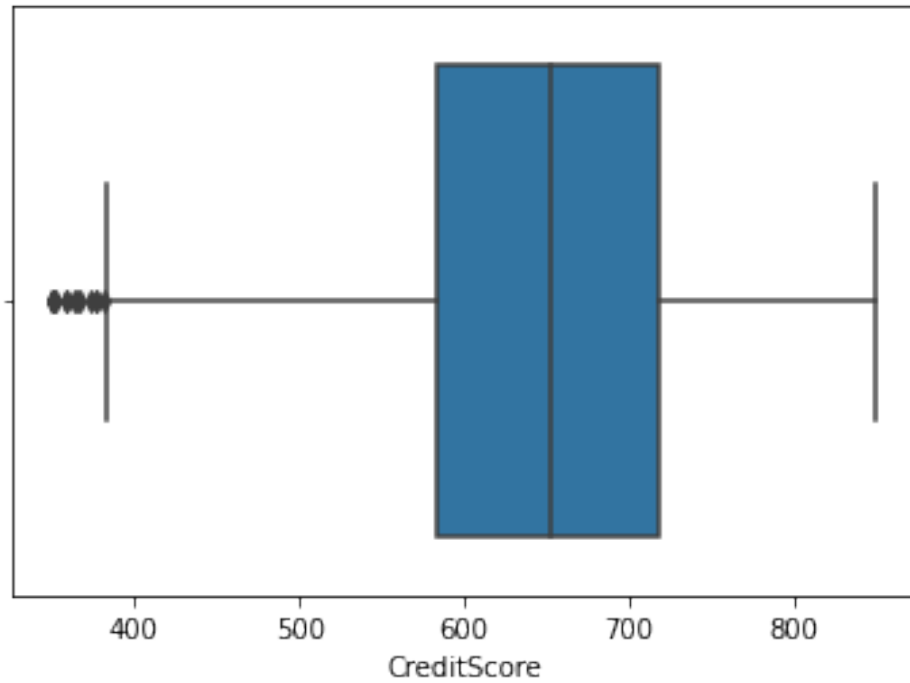


```
sns.boxplot(data['CreditScore'])
```

```
/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.
```

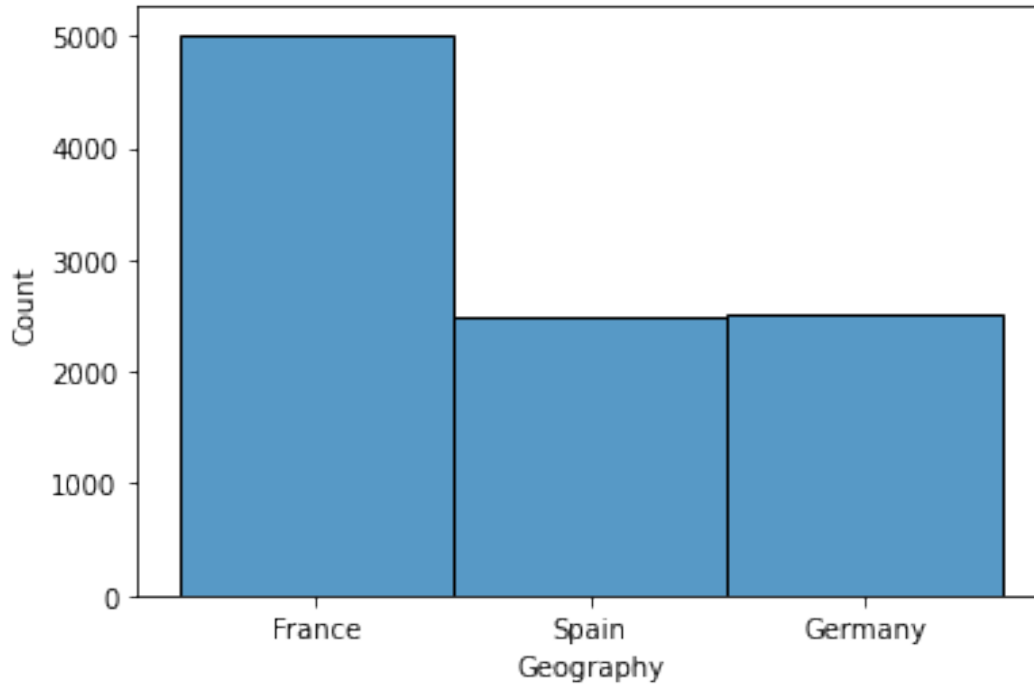
```
FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbae8e1e6d0>
```



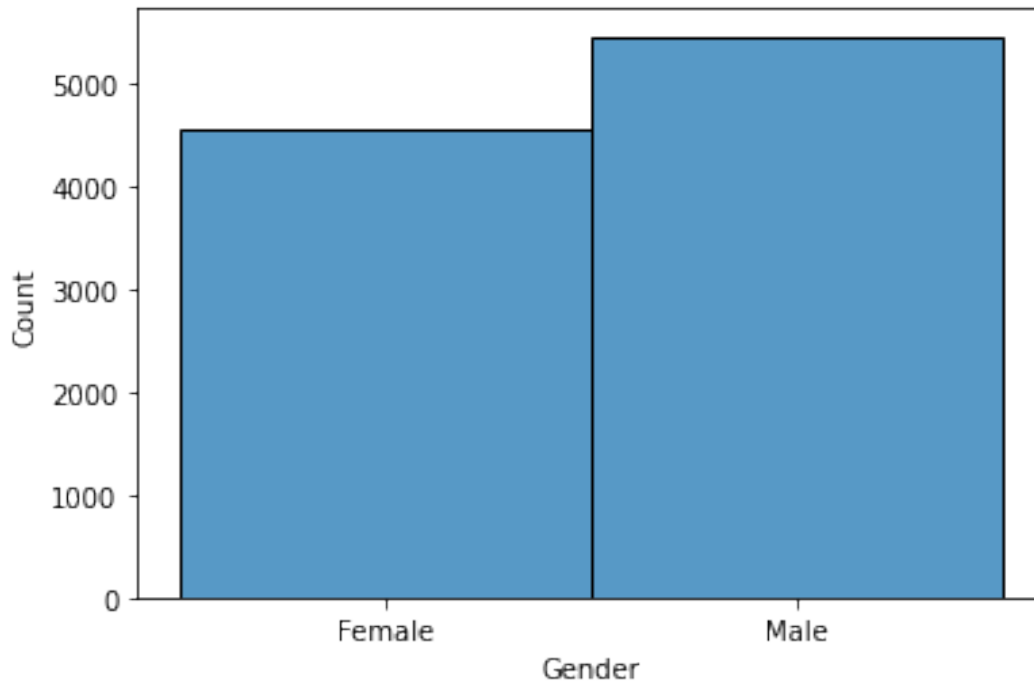
```
sns.histplot(data['Geography'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbae8d7aa90>
```



```
sns.histplot(data['Gender'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbae8cf4750>
```

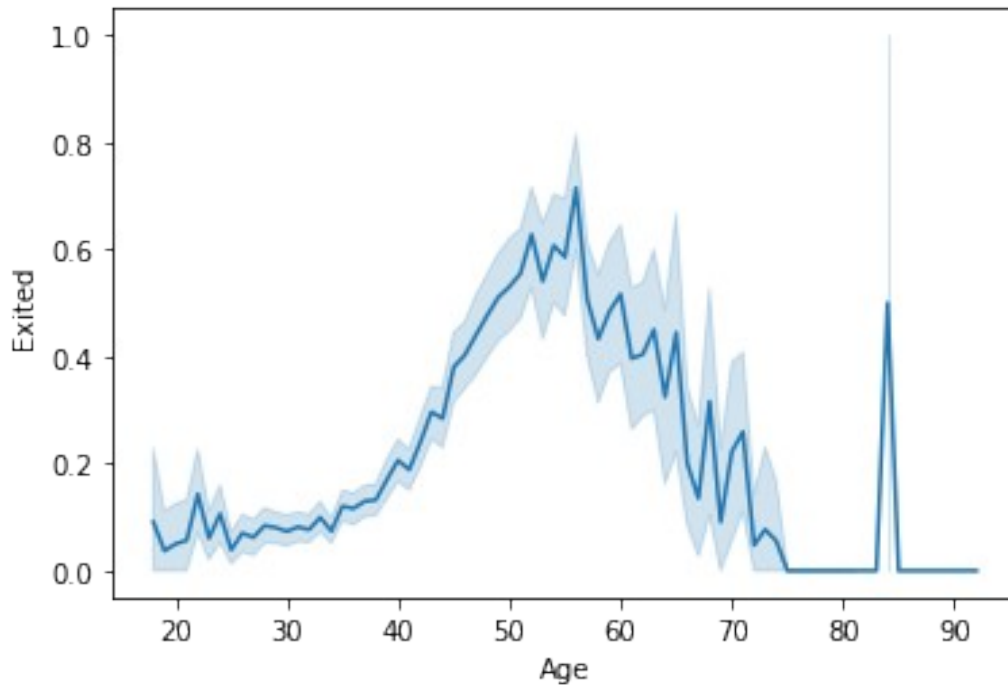


```
sns.lineplot(data['Age'], data['Exited'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:  
FutureWarning: Pass the following variables as keyword args: x, y.  
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 0x7fbae8cac050>
```

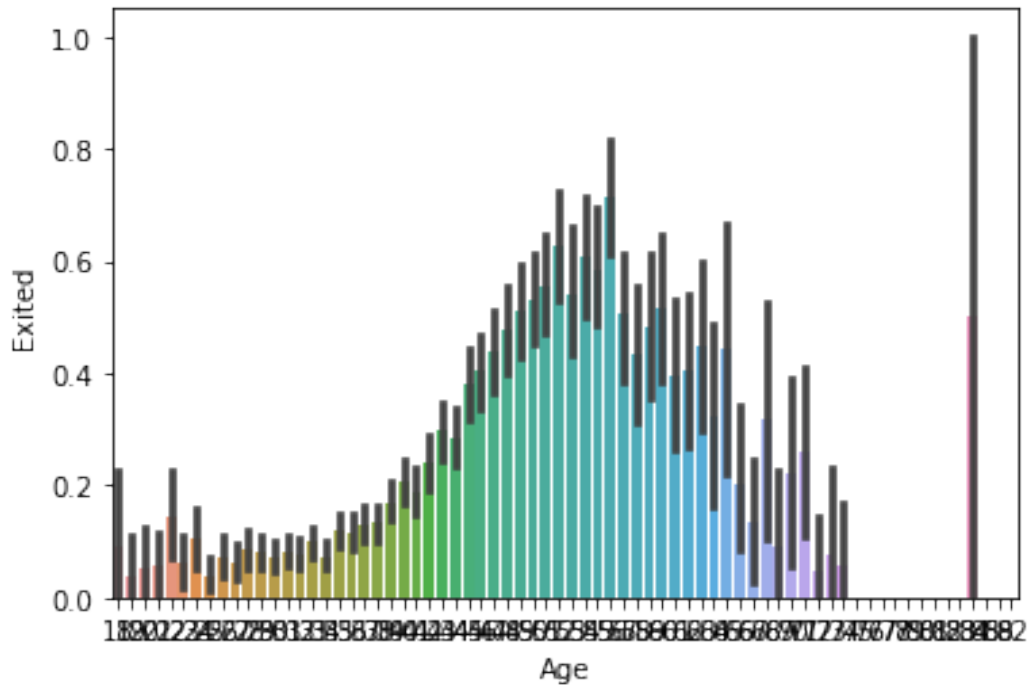


```
sns.barplot(data['Age'], data['Exited'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:  
FutureWarning: Pass the following variables as keyword args: x, y.  
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 0x7fbae8bf6e10>
```

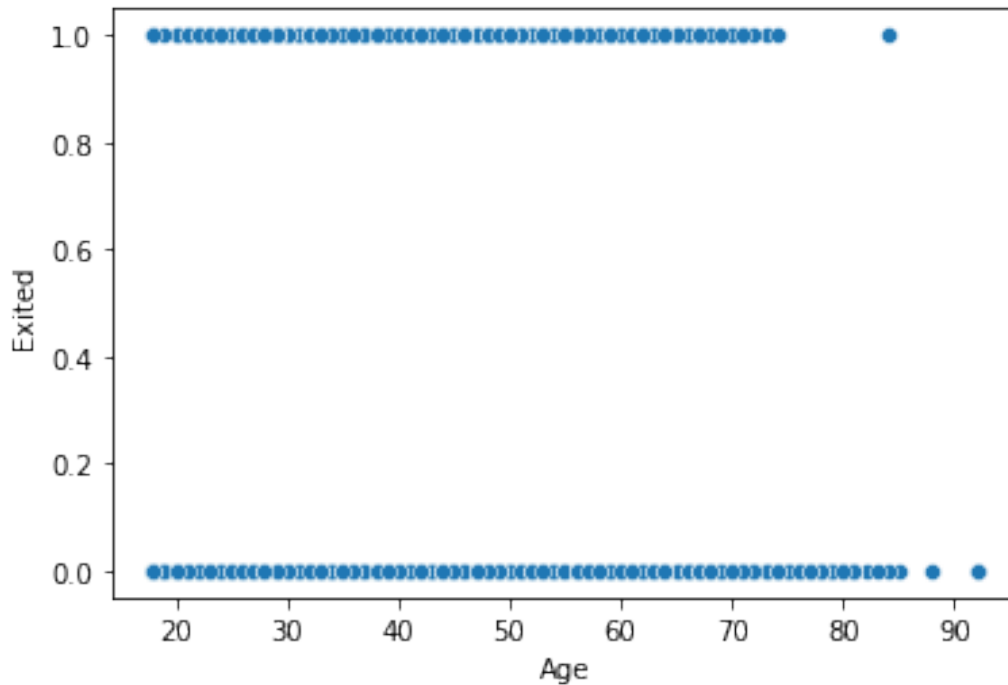


```
sns.scatterplot(data['Age'], data['Exited'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
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 0x7fbae89675d0>
```

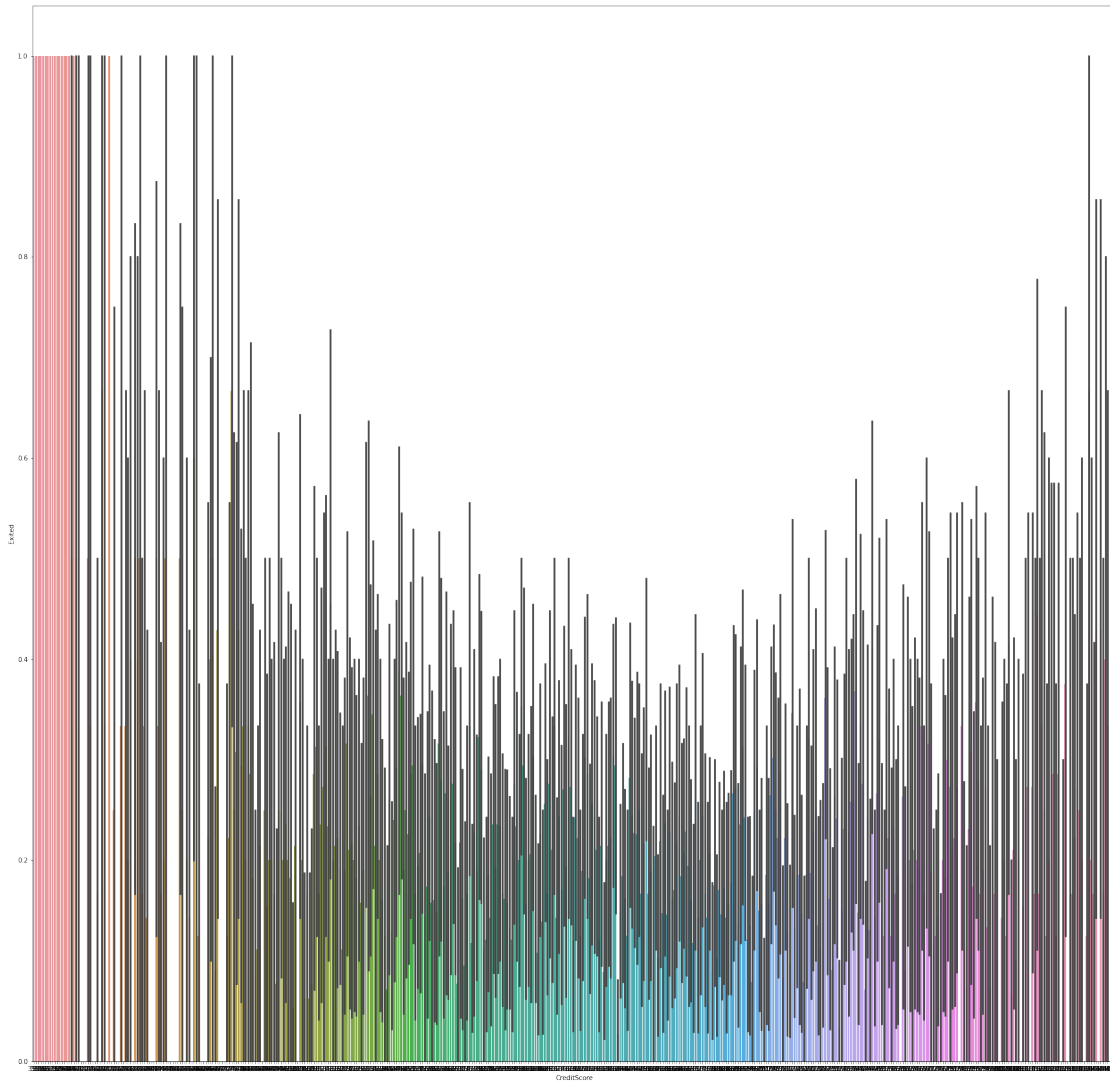


```
plt.figure(figsize=(30,30))
sns.barplot(data['CreditScore'], data['Exited'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
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 0x7fbae7827b10>
```



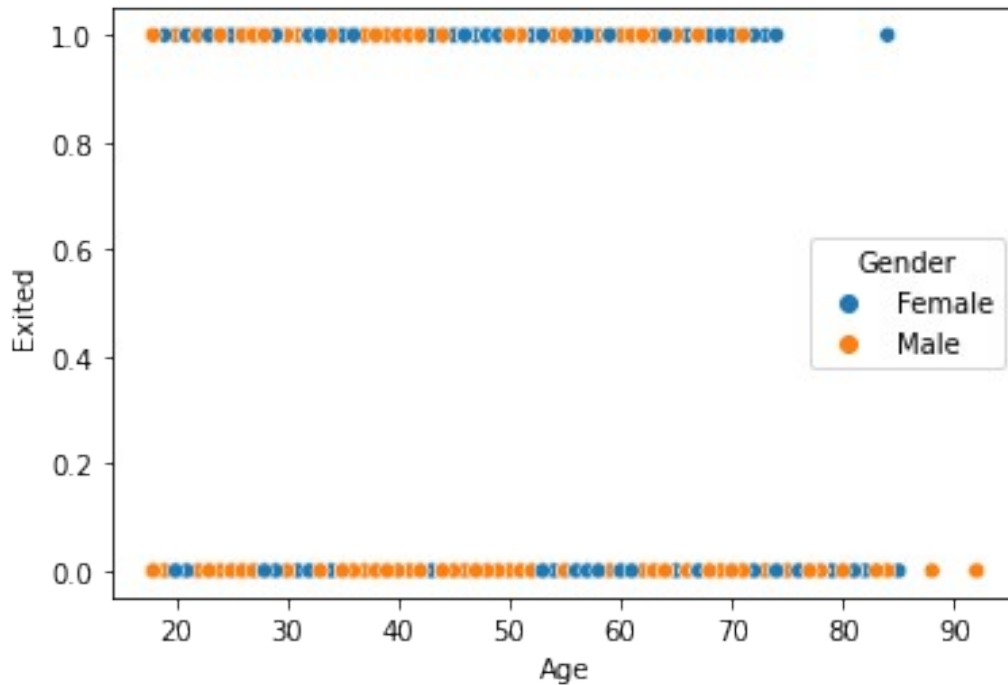
multivariate

```
sns.scatterplot(data['Age'], data['Exited'], hue=data['Gender'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:  
FutureWarning: Pass the following variables as keyword args: x, y.  
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 0x7fbae6609e10>
```

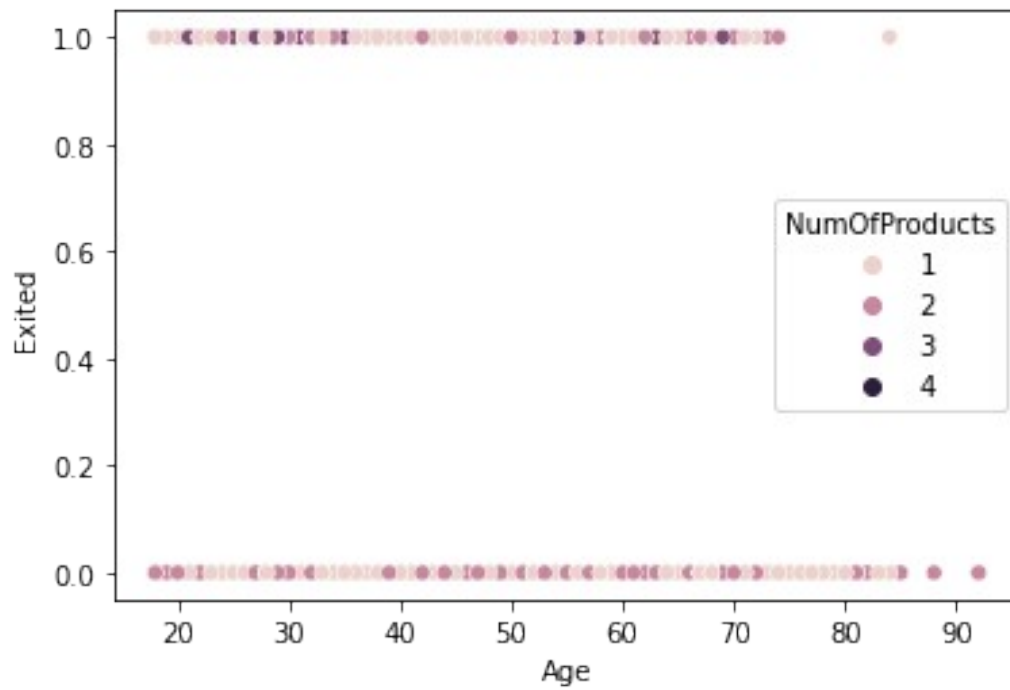


```
sns.scatterplot(data['Age'], data['Exited'], hue=data['NumOfProducts'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:  
FutureWarning: Pass the following variables as keyword args: x, y.  
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.
```

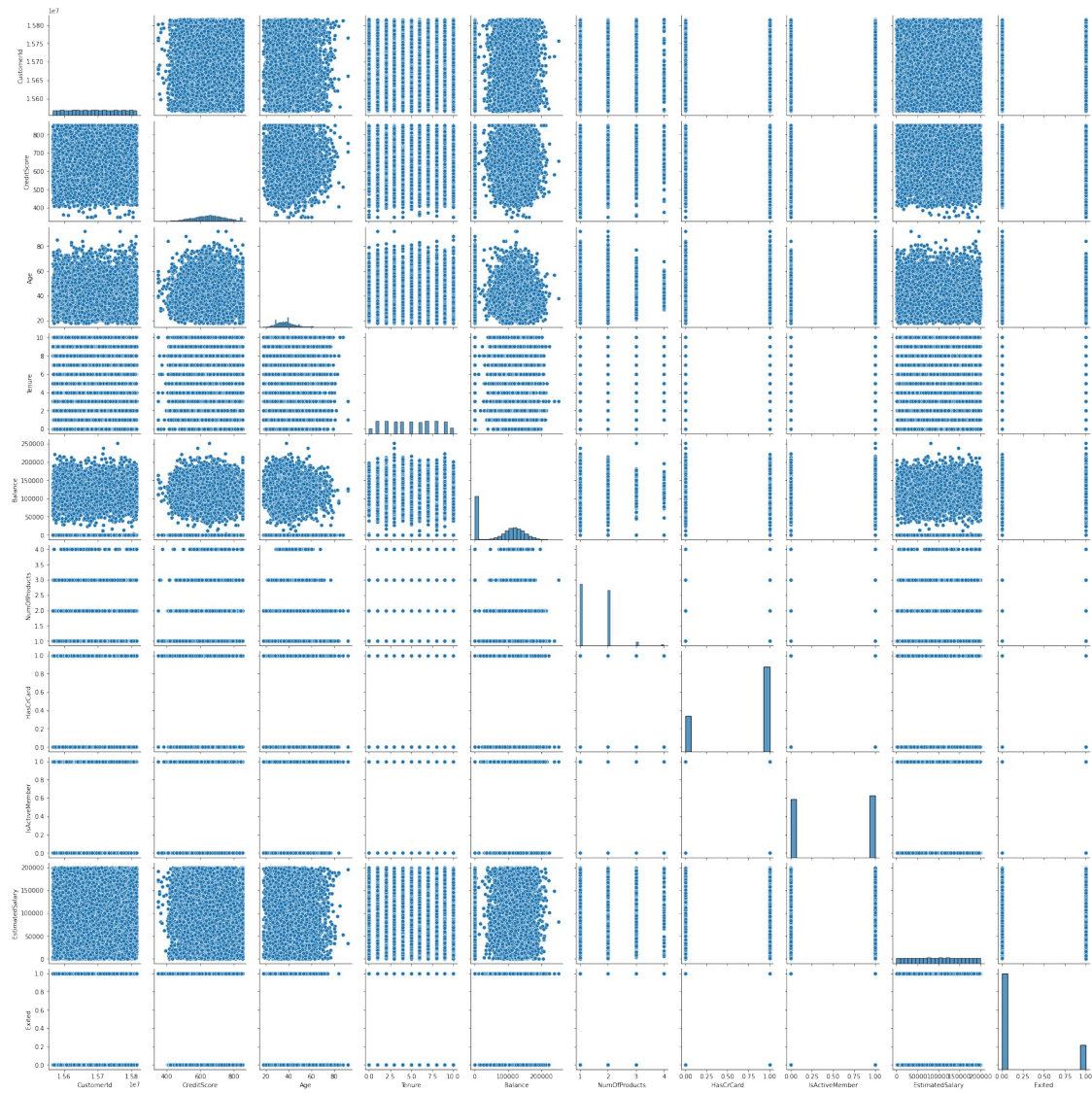
```
FutureWarning
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1d7b2c3b90>
```

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f1d7af87710>
```



```
data.corr()
```

	CustomerId	CreditScore	Age	Tenure	Balance
CustomerId	1.000000	0.005308	0.009497	-0.014883	-0.012419
CreditScore	0.005308	1.000000	-0.003965	0.000842	0.006268
Age	0.009497	-0.003965	1.000000	-0.009997	0.028308
Tenure	-0.014883	0.000842	-0.009997	1.000000	-0.012254
Balance	-0.012419	0.006268	0.028308	-0.012254	1.000000
NumOfProducts	0.016972	0.012238	-0.030680	0.013444	-0.304180

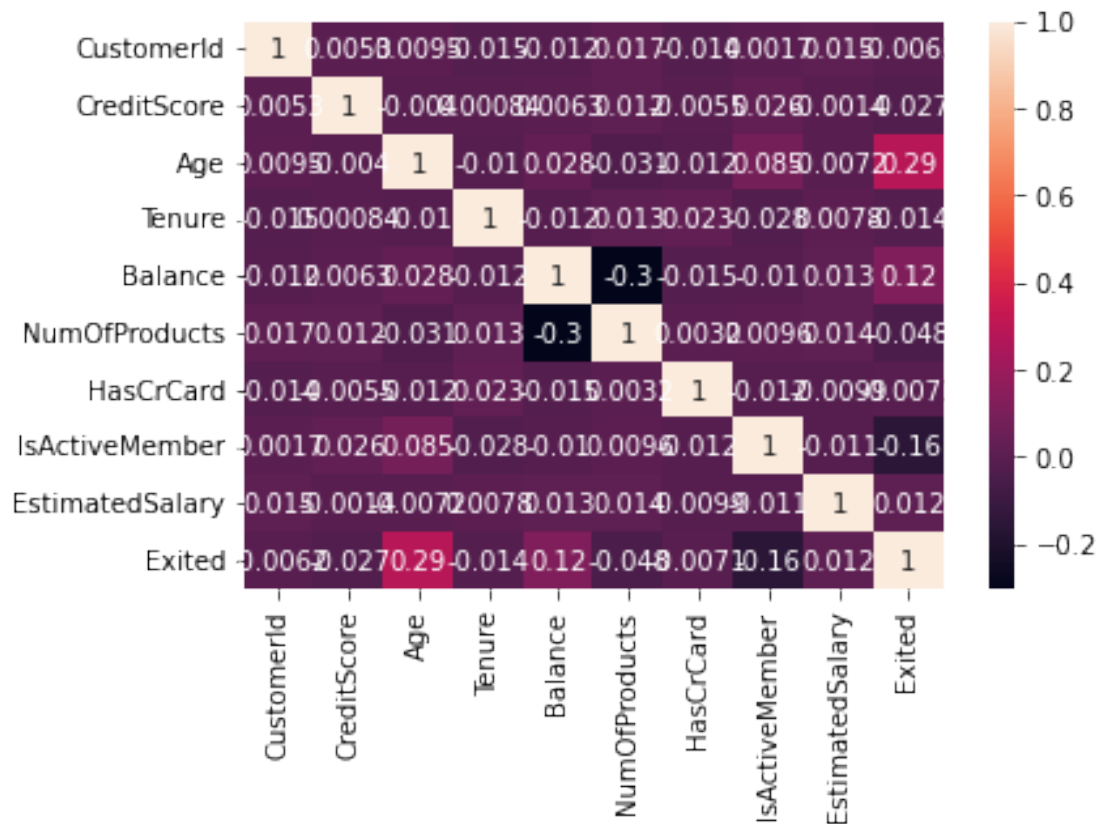
HasCrCard	-0.014025	-0.005458	-0.011721	0.022583	-0.014858
IsActiveMember	0.001665	0.025651	0.085472	-0.028362	-0.010084
EstimatedSalary	0.015271	-0.001384	-0.007201	0.007784	0.012797
Exited	-0.006248	-0.027094	0.285323	-0.014001	0.118533

	NumOfProducts	HasCrCard	IsActiveMember	
EstimatedSalary	\			
CustomerId	0.016972	-0.014025	0.001665	
0.015271				
CreditScore	0.012238	-0.005458	0.025651	-
0.001384				
Age	-0.030680	-0.011721	0.085472	-
0.007201				
Tenure	0.013444	0.022583	-0.028362	
0.007784				
Balance	-0.304180	-0.014858	-0.010084	
0.012797				
NumOfProducts	1.000000	0.003183	0.009612	
0.014204				
HasCrCard	0.003183	1.000000	-0.011866	-
0.009933				
IsActiveMember	0.009612	-0.011866	1.000000	-
0.011421				
EstimatedSalary	0.014204	-0.009933	-0.011421	
1.000000				
Exited	-0.047820	-0.007138	-0.156128	
0.012097				

	Exited
CustomerId	-0.006248
CreditScore	-0.027094
Age	0.285323
Tenure	-0.014001
Balance	0.118533
NumOfProducts	-0.047820
HasCrCard	-0.007138
IsActiveMember	-0.156128
EstimatedSalary	0.012097
Exited	1.000000

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1d7a1e7c90>
```



```
data.isnull().any()
```

```
CustomerId      False
Surname         False
CreditScore     False
Geography       False
Gender          False
Age            False
Tenure         False
Balance        False
NumOfProducts  False
HasCrCard      False
IsActiveMember False
EstimatedSalary False
Exited         False
dtype: bool
```

```
data.isnull().sum()
```

```
CustomerId      0
Surname         0
CreditScore     0
Geography       0
Gender          0
Age            0
```

```
Tenure          0
Balance         0
NumOfProducts  0
HasCrCard       0
IsActiveMember  0
EstimatedSalary 0
Exited          0
dtype: int64
```

```
data.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.
```

```
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
```

```
data['Age'].mode()
```

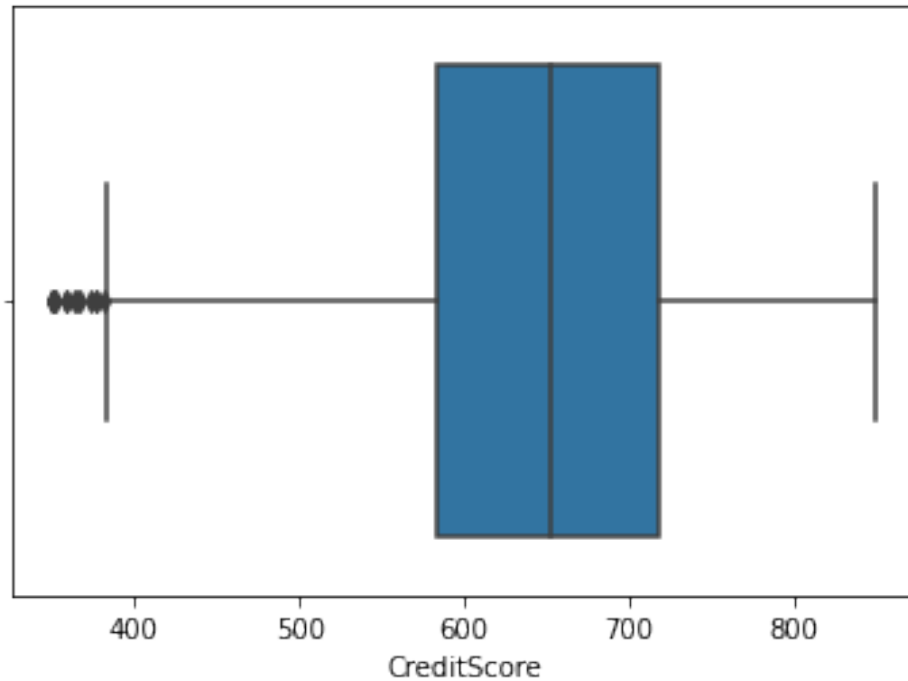
```
0    37
dtype: int64
```

```
sns.boxplot(data['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 0x7f1d73f49d10>
```



```
q = data.quantile([0.75,0.25])
```

```
q
```

	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts
0.75	15753233.75	718.0	44.0	7.0	127644.24	2.0
0.25	15628528.25	584.0	32.0	3.0	0.00	1.0

	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.75	1.0	1.0	149388.2475	0.0
0.25	0.0	0.0	51002.1100	0.0

```
iqr = q.iloc[0] - q.iloc[1]
```

```
iqr
```

CustomerId	124705.5000
CreditScore	134.0000
Age	12.0000
Tenure	4.0000
Balance	127644.2400
NumOfProducts	1.0000
HasCrCard	1.0000
IsActiveMember	1.0000
EstimatedSalary	98386.1375
Exited	0.0000
dtype:	float64

```
u = 584 + (1.5*134)
```

```
u
```

```
785.0
```

```
l = 718 - (1.5*134)
```

```
l
```

```
517.0
```

```
data['CreditScore']=
```

```
np.where(data['CreditScore']>785,517,data['CreditScore'])
```

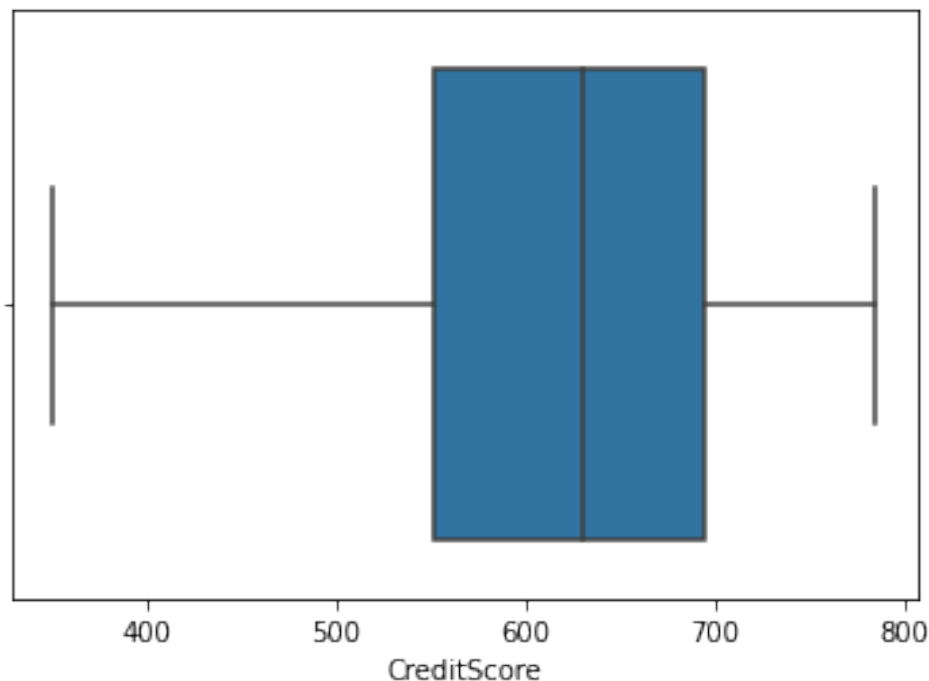
```
sns.boxplot(data['CreditScore'])
```

```
/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 0x7f1d7af87250>
```



```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
le = LabelEncoder()
```

```
oneh = OneHotEncoder()
```

```
data['Gender'] = le.fit_transform(data['Gender'])
```

```
data.head()
```

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	\
0	15634602	Hargrave	619	France	0	42	2	
1	15647311	Hill	608	Spain	0	41	1	
2	15619304	Onio	502	France	0	42	8	
3	15701354	Boni	699	France	0	39	1	
4	15737888	Mitchell	517	Spain	0	43	2	

	Balance	NumOfProducts	HasCrCard	IsActiveMember
EstimatedSalary	0.00	1	1	1
101348.88				
1	83807.86	1	0	1
112542.58				
2	159660.80	3	1	0
113931.57				
3	0.00	2	0	0
93826.63				
4	125510.82	1	1	1
79084.10				

	Exited
0	1
1	0
2	1
3	0
4	0

```
data['Geography'] = le.fit_transform(data['Geography'])
```

```
data.head()
```

	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	\
0	15634602	1115	619	0	0	42	2	
1	15647311	1177	608	2	0	41	1	
2	15619304	2040	502	0	0	42	8	
3	15701354	289	699	0	0	39	1	
4	15737888	1822	517	2	0	43	2	

	Balance	NumOfProducts	HasCrCard	IsActiveMember
EstimatedSalary	0.00	1	1	1
101348.88				
1	83807.86	1	0	1
112542.58				
2	159660.80	3	1	0
113931.57				
3	0.00	2	0	0
93826.63				
4	125510.82	1	1	1
79084.10				

	Exited
0	1
1	0
2	1
3	0
4	0

```
x=data.drop(['CustomerId','Surname','EstimatedSalary'],axis=1)
```

```
y=data['EstimatedSalary']
```

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
```

```
sc = StandardScaler()
```

```
x_scaled = sc.fit_transform(x)
```

```
x_scaled
```

```
array([[ -0.0569739 , -0.90188624, -1.09598752, ...,  0.64609167,
         0.97024255,  1.97716468],
       [ -0.18323566,  1.51506738, -1.09598752, ..., -1.54776799,
         0.97024255, -0.50577476],
       [ -1.39993993, -0.90188624, -1.09598752, ...,  0.64609167,
        -1.03067011,  1.97716468],
       ...,
       [  0.9760769 , -0.90188624, -1.09598752, ..., -1.54776799,
         0.97024255,  1.97716468],
       [  1.69921245,  0.30659057,  0.91241915, ...,  0.64609167,
        -1.03067011,  1.97716468],
       [ -1.2277648 , -0.90188624, -1.09598752, ...,  0.64609167,
        -1.03067011, -0.50577476]])
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y,
test_size = 0.3, random_state = 0)
```

```
x_train
```

```
array([[ 0.19554963, -0.90188624,  0.91241915, ...,  0.64609167,
         0.97024255,  1.97716468],
       [ -0.95228459, -0.90188624,  0.91241915, ...,  0.64609167,
        -1.03067011, -0.50577476],
       [ -0.38984582, -0.90188624, -1.09598752, ..., -1.54776799,
         0.97024255, -0.50577476],
       ...,
       [  1.30894882, -0.90188624,  0.91241915, ...,  0.64609167,
        -1.03067011, -0.50577476],
       [ -0.38984582,  1.51506738, -1.09598752, ...,  0.64609167,
         0.97024255, -0.50577476],
       [ -0.01106053,  0.30659057, -1.09598752, ...,  0.64609167,
        -1.03067011,  1.97716468]])
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(7000, 10)
(3000, 10)
(7000,)
(3000,)
(7000, 10)
(3000, 10)
(7000,)
(3000,)
```