November 20, 2022

```
[ ]: import numpy as np
     import pandas as pd
```

```
[ ]: df = pd.read_csv('/content/Churn_Modelling.csv')
```

```
[ ]: df
```

```
[ ]:       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]
```

3. Visualization

```
[ ]: import matplotlib.pyplot as plt
```

```
[ ]: import seaborn as sns
```

```
[ ]: %matplotlib inline
```

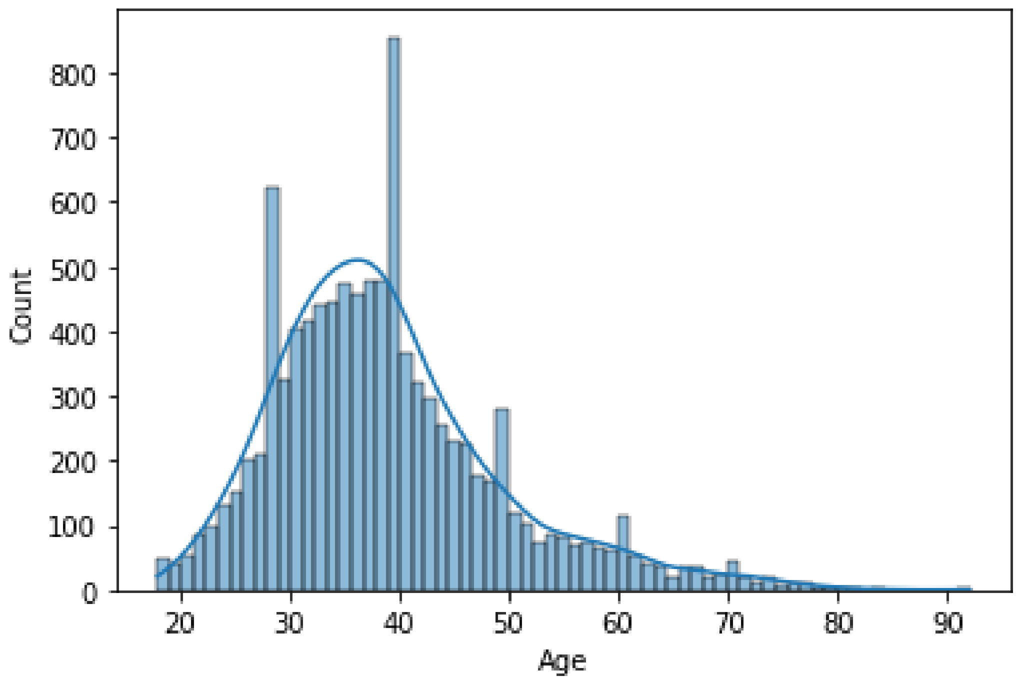
(i) Univariate Analysis

```
[ ]: df[['CustomerId','Surname','CreditScore','Geography','Age','Tenure']].describe()
```

```
[ ]:          CustomerId   CreditScore           Age        Tenure
     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
```

```
[ ]: sns.histplot(df.Age,kde=True)
```

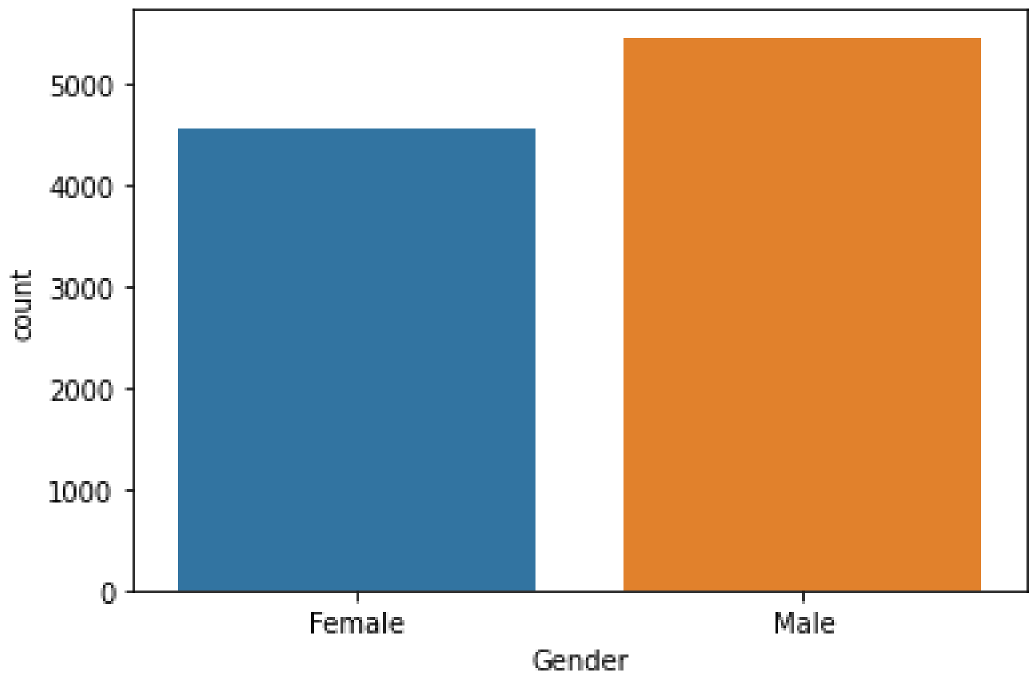
```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffbaff48d10>
```



```
[ ]: # plot count plot for the gender column
     sns.countplot(df.Gender)
```

```
/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 0x7ffbafcdbf90>
```



(ii) Bivariate Analysis

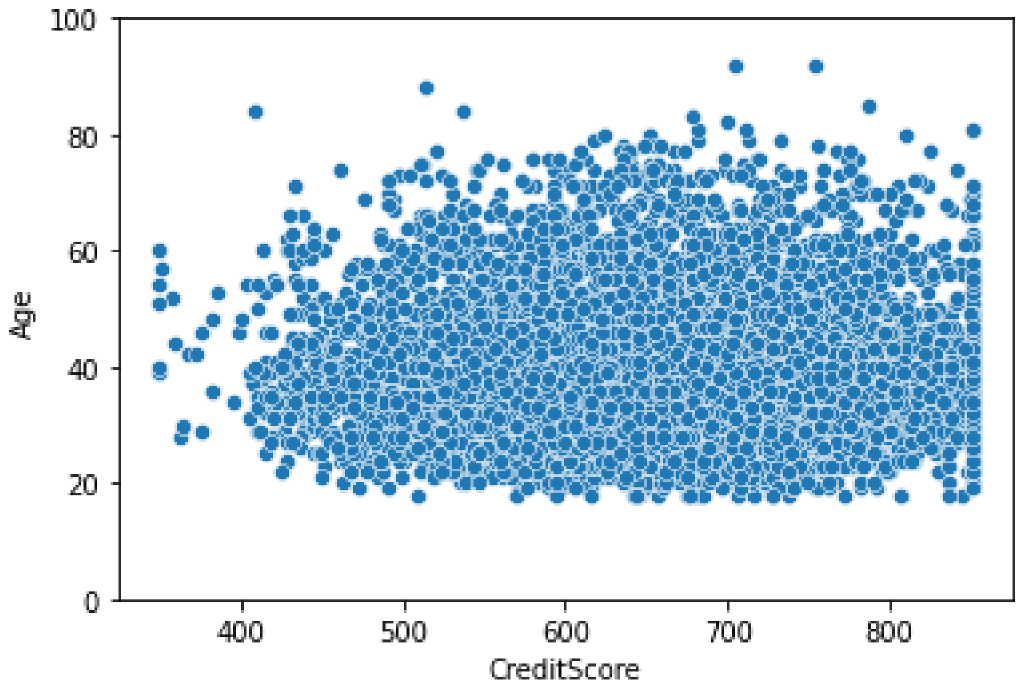
```
[ ]: df[['CustomerId','Surname','CreditScore','Geography','Gender','Age']].corr()
```

```
[ ]:              CustomerId  CreditScore       Age
     CustomerId     1.000000     0.005308  0.009497
     CreditScore    0.005308     1.000000 -0.003965
     Age            0.009497    -0.003965  1.000000
```

```
[ ]: sns.scatterplot(df.CreditScore,df.Age)
     plt.ylim(0,100)
```

```
/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
```

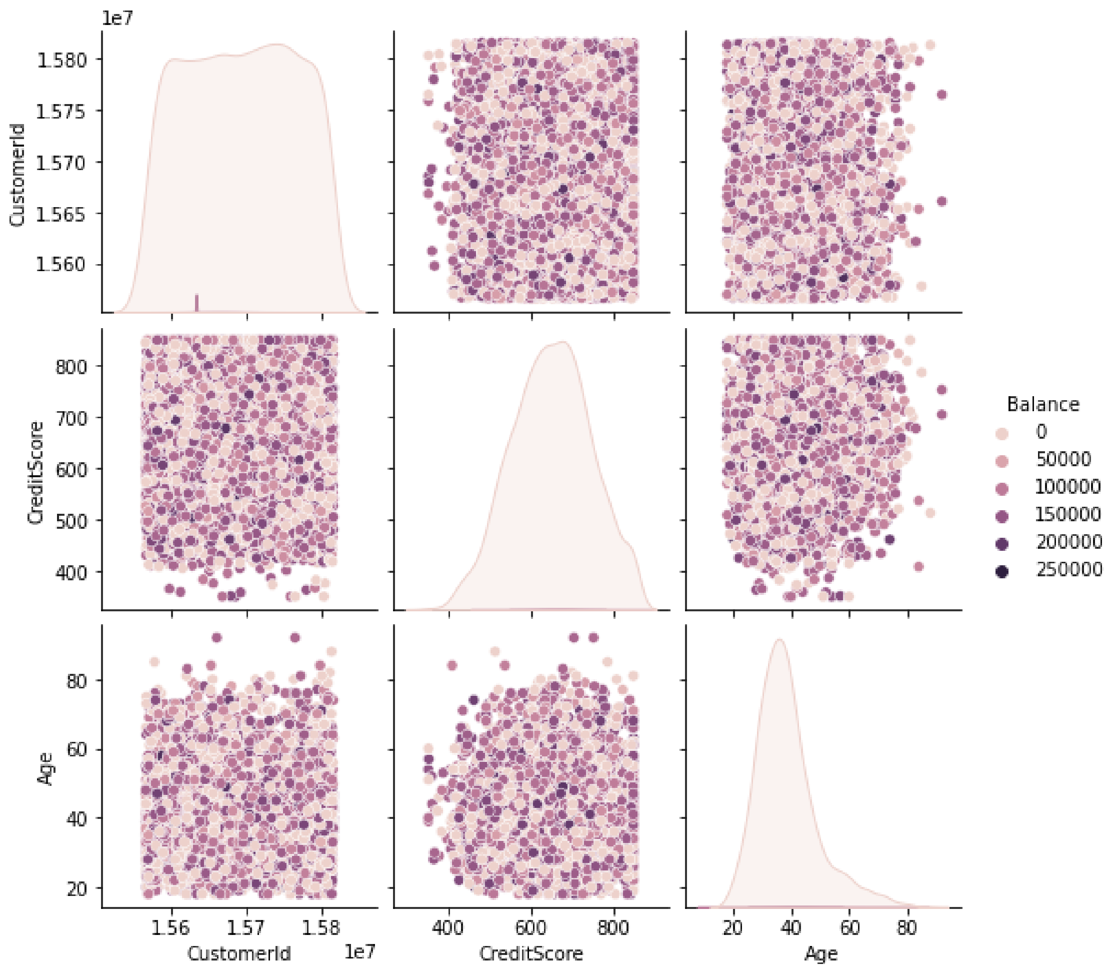
```
[ ]: (0.0, 100.0)
```



(iii) Multivariate Analysis

```
[ ]: sns.pairplot(data␣
      ↪=df[['CustomerId','Geography','Gender','CreditScore','Age','Balance']],hue =␣
      ↪'Balance')
```

```
[ ]: <seaborn.axisgrid.PairGrid at 0x7ffbaf7d2910>
```



4. Descriptive Statistics

```
[ ]: #mode
     df['Age'].mode()
```

```
[ ]: 0    37
     dtype: int64
```

```
[ ]: #calculation of the mean (for Age)
     df["Age"].mean()
```

```
[ ]: 38.9218
```

```
[ ]: #calculation of the mean and round the result(for Age)
     round(df["Age"].mean(), 2)
```

```
[ ]: 38.92
[ ]: #calculation of the median(for Age)
     df["Age"].median()
```

```
[ ]: 37.0
```

```
[ ]: df.columns
```

```
[ ]: Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
            'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
            'IsActiveMember', 'EstimatedSalary', 'Exited'],
           dtype='object')
```

```
[ ]: df["NumOfProducts"].value_counts()
```

```
[ ]: 1    5084
     2    4590
     3     266
     4      60
     Name: NumOfProducts, dtype: int64
```

```
[ ]: df.dtypes
```

```
[ ]: RowNumber            int64
     CustomerId           int64
     Surname             object
     CreditScore          int64
     Geography           object
     Gender              object
     Age                  int64
     Tenure               int64
     Balance            float64
     NumOfProducts        int64
     HasCrCard            int64
     IsActiveMember       int64
     EstimatedSalary    float64
     Exited               int64
     dtype: object
```

```
[ ]: df.head()
```

```
[ ]:    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
```

```
   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

   EstimatedSalary  Exited
0        101348.88       1
1        112542.58       0
2        113931.57       1
3         93826.63       0
4         79084.10       0
```

[ ]:
```
df.describe()
```

[ ]:
```
         RowNumber   CustomerId   CreditScore           Age        Tenure  \
count  10000.00000  1.000000e+04  10000.000000  10000.000000  10000.000000
mean    5000.50000  1.569094e+07    650.528800     38.921800      5.012800
std     2886.89568  7.193619e+04     96.653299     10.487806      2.892174
min        1.00000  1.556570e+07    350.000000     18.000000      0.000000
25%     2500.75000  1.562853e+07    584.000000     32.000000      3.000000
50%     5000.50000  1.569074e+07    652.000000     37.000000      5.000000
75%     7500.25000  1.575323e+07    718.000000     44.000000      7.000000
max    10000.00000  1.581569e+07    850.000000     92.000000     10.000000

             Balance  NumOfProducts    HasCrCard  IsActiveMember  \
count   10000.000000   10000.000000  10000.00000    10000.000000
mean    76485.889288       1.530200      0.70550        0.515100
std     62397.405202       0.581654      0.45584        0.499797
min         0.000000       1.000000      0.00000        0.000000
25%         0.000000       1.000000      0.00000        0.000000
50%     97198.540000       1.000000      1.00000        1.000000
75%    127644.240000       2.000000      1.00000        1.000000
max    250898.090000       4.000000      1.00000        1.000000

       EstimatedSalary        Exited
count     10000.000000  10000.000000
mean     100090.239881      0.203700
std       57510.492818      0.402769
min          11.580000      0.000000
25%       51002.110000      0.000000
50%      100193.915000      0.000000
75%      149388.247500      0.000000
max      199992.480000      1.000000
```

5. Handling missing values

```
[ ]: df.isna().any()
```

```
[ ]: RowNumber          False
     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
```

```
[ ]: df.isnull().sum()
```

```
[ ]: RowNumber          0
     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
```

```
[ ]: 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()
```

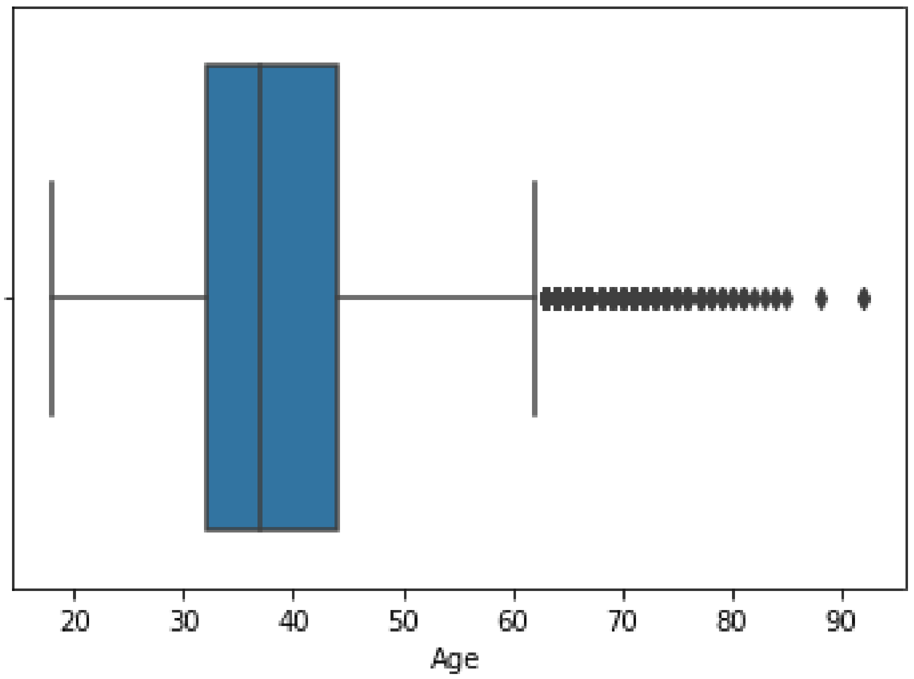
[ ]:
```
      RowNumber  CustomerId  Surname  CreditScore  Geography  Gender   Age  \
0          True        True     True         True       True    True  True
1          True        True     True         True       True    True  True
2          True        True     True         True       True    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]
```

6. Finding and replacing outliers

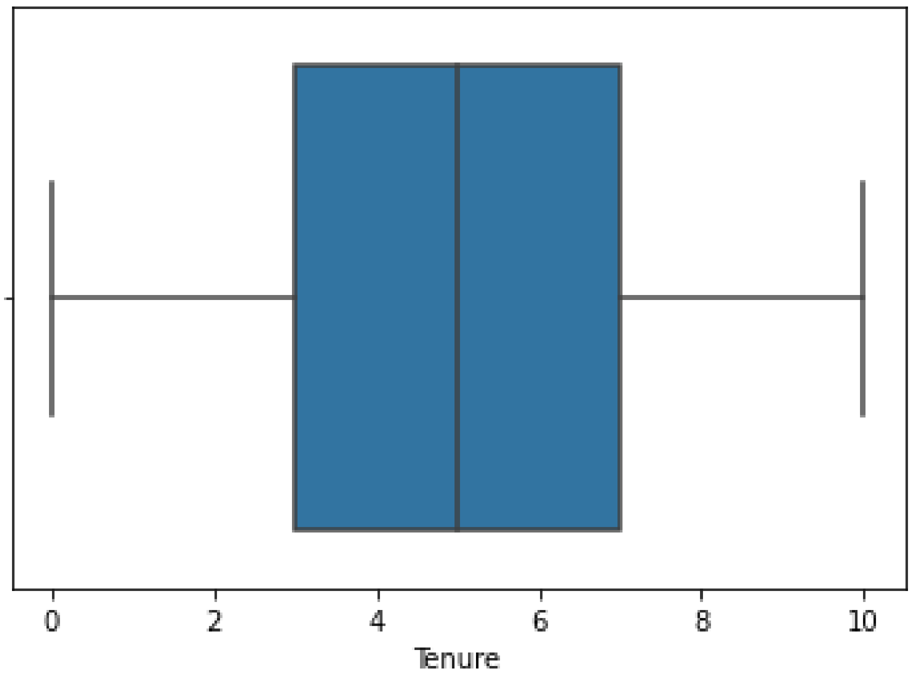
```
[ ]: import seaborn as sns
     sns.boxplot(x=df['Age'])
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffbade3d490>
```



```
[ ]: sns.boxplot(x=df['Tenure'])
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ffbaae4ac50>
```



7. Check for categorical column and perform encoding

```
[ ]: import pandas as pd
     df = pd.read_csv("Churn_Modelling.csv", header=None)
```

```
[ ]: cols = df.columns
     num_cols = df._get_numeric_data().columns
```

```
[ ]: num_cols
```

```
[ ]: Int64Index([], dtype='int64')
```

```
[ ]: list(set(cols) - set(num_cols))
```

```
[ ]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

8. Split the data into dependent and independent variables

```
[ ]: # x  -Independent
     # y  -Dependent
     x =df.drop('Exited',axis=1)
     y=df['Exited']
```

```
[ ]: x.head()
```

```
       ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-53-830ed5e65d76> in <module>
   ----> 1 x.head()


       NameError: name 'x' is not defined
```

```
[ ]: y.head()
```

```
       ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)
```

```
    <ipython-input-54-17b2b1f6e15b> in <module>
----> 1 y.head()


    NameError: name 'y' is not defined
```

9. Scale the independent variables

[ ]:
```python
from sklearn import linear_model
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
```

[ ]:
```python
X = df[['Balance', 'Tenure']]
scaledX = scale.fit_transform(X)
print(scaledX)
```