

LOAD THE DATASET

```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
import sklearn as sk
dataset = pd.read_csv("Churn_Modelling.csv")
dataset.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

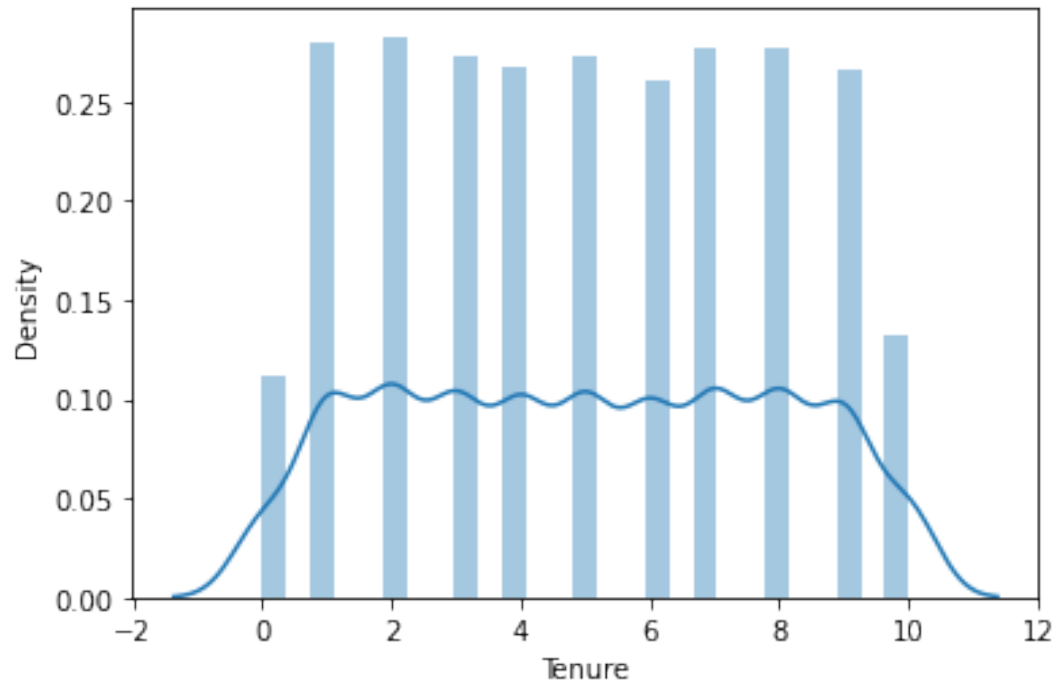
	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	
3	1	0.00	2	0	0	
4	2	125510.82	1	1	1	

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

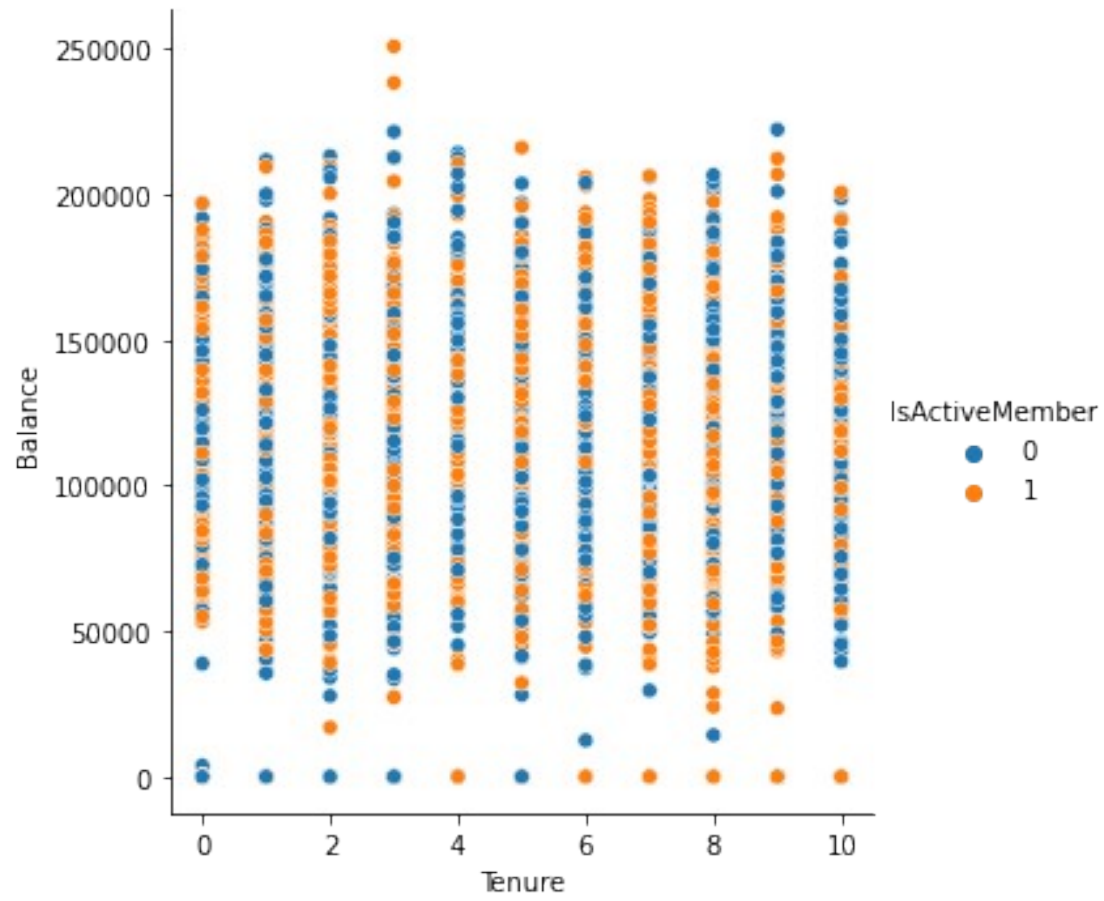
```
import warnings
warnings.filterwarnings("ignore")
```

```
sb.distplot(dataset["Tenure"])
```

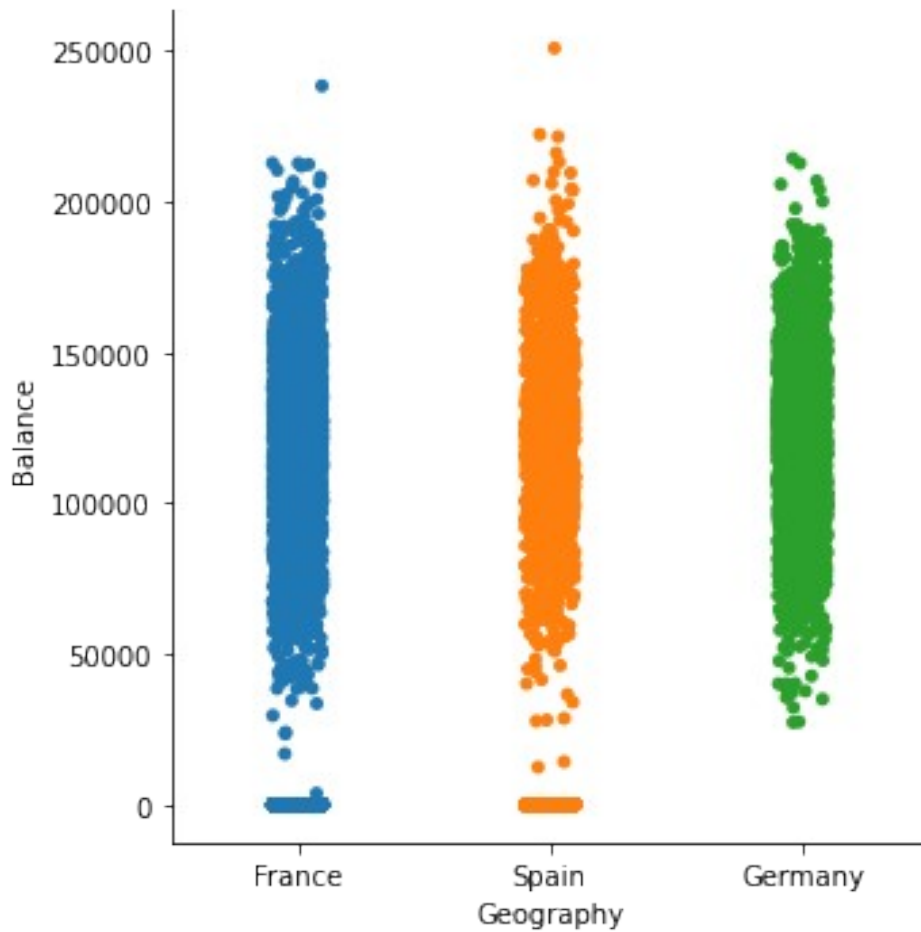
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f92389efa10>
```



```
sb.relplot(x="Tenure",y="Balance",data=dataset,hue="IsActiveMember")  
<seaborn.axisgrid.FacetGrid at 0x7f9238846510>
```



```
sb.catplot(x="Geography",y="Balance",data=dataset)  
<seaborn.axisgrid.FacetGrid at 0x7f9238846690>
```



```
dataset = pd.read_csv("Churn_Modelling.csv")
```

```
dataset.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

dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 10000 entries, 0 to 9999

Data columns (total 14 columns):

#	Column	Non-Null Count	Dtype
0	RowNumber	10000 non-null	int64
1	CustomerId	10000 non-null	int64
2	Surname	10000 non-null	object
3	CreditScore	10000 non-null	int64
4	Geography	10000 non-null	object
5	Gender	10000 non-null	object
6	Age	10000 non-null	int64
7	Tenure	10000 non-null	int64
8	Balance	10000 non-null	float64
9	NumOfProducts	10000 non-null	int64
10	HasCrCard	10000 non-null	int64
11	IsActiveMember	10000 non-null	int64
12	EstimatedSalary	10000 non-null	float64
13	Exited	10000 non-null	int64

dtypes: float64(2), int64(9), object(3)

memory usage: 1.1+ MB

dataset.describe()

	RowNumber	CustomerId	CreditScore	Age
Tenure \				
count	10000.000000	1.000000e+04	10000.000000	10000.000000
mean	5000.500000	1.569094e+07	650.528800	38.921800
std	2886.89568	7.193619e+04	96.653299	10.487806
min	1.000000	1.556570e+07	350.000000	18.000000
25%	2500.750000	1.562853e+07	584.000000	32.000000
50%	5000.500000	1.569074e+07	652.000000	37.000000
75%	7500.250000	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	10000.000000	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

```
dataset.mean()
```

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

```
dataset["EstimatedSalary"].mean()
```

```
100090.239881
```

```
dataset.mode()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age \						
0	1	15565701	Smith	850.0	France	Male
37.0						
1	2	15565706	NaN	NaN	NaN	NaN
NaN						

2	3	15565714	NaN	NaN	NaN	NaN
NaN						
3	4	15565779	NaN	NaN	NaN	NaN
NaN						
4	5	15565796	NaN	NaN	NaN	NaN
NaN						
...
.						
9995	9996	15815628	NaN	NaN	NaN	NaN
NaN						
9996	9997	15815645	NaN	NaN	NaN	NaN
NaN						
9997	9998	15815656	NaN	NaN	NaN	NaN
NaN						
9998	9999	15815660	NaN	NaN	NaN	NaN
NaN						
9999	10000	15815690	NaN	NaN	NaN	NaN
NaN						

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2.0	0.0	1.0	1.0		1.0
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

	EstimatedSalary	Exited
0	24924.92	0.0
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN
...
9995	NaN	NaN
9996	NaN	NaN
9997	NaN	NaN
9998	NaN	NaN
9999	NaN	NaN

[10000 rows x 14 columns]

dataset["Balance"].median()

97198.540000000001

```
dataset.median()
```

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

```
dataset.skew()
```

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

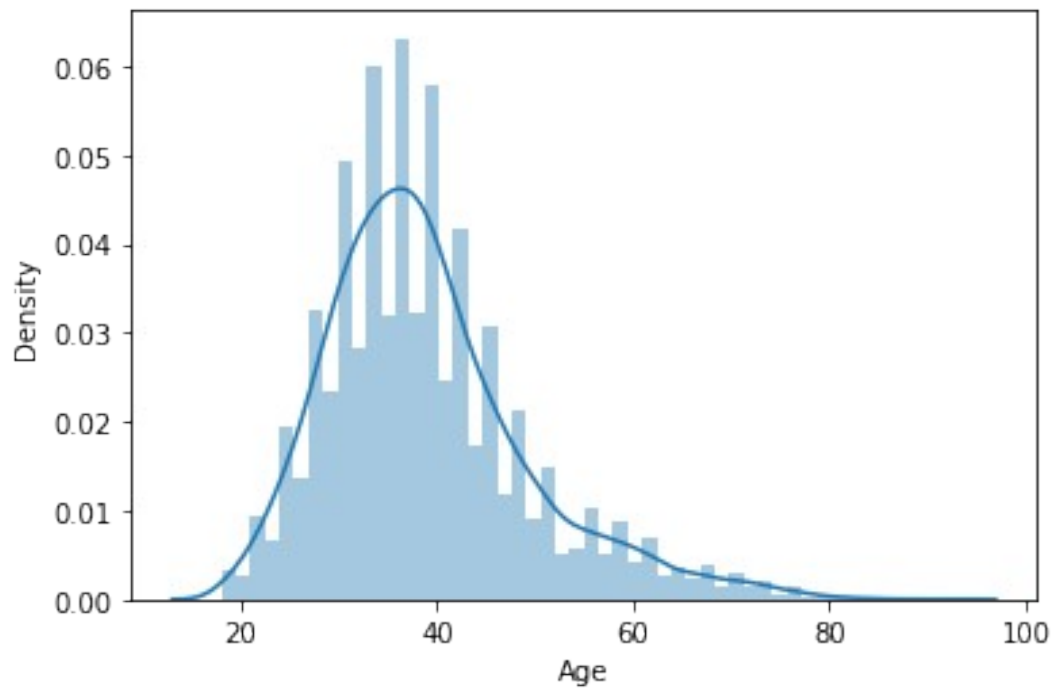
```
dataset.kurtosis()
```

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

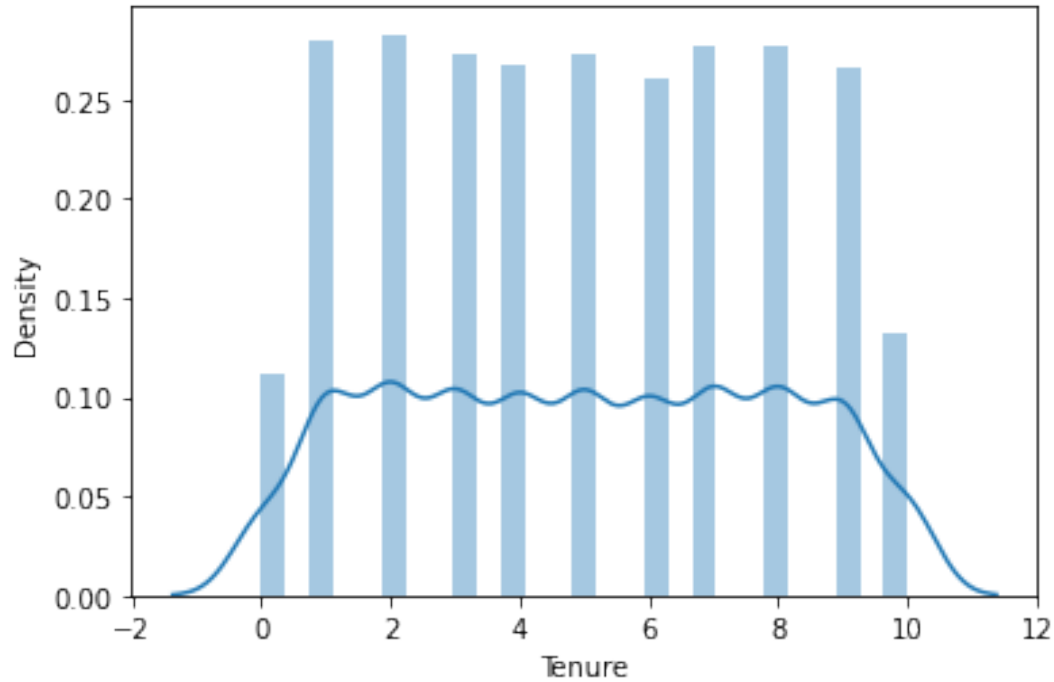
```
import seaborn as sns
```

```
print(sns.distplot(dataset["Age"]))
```

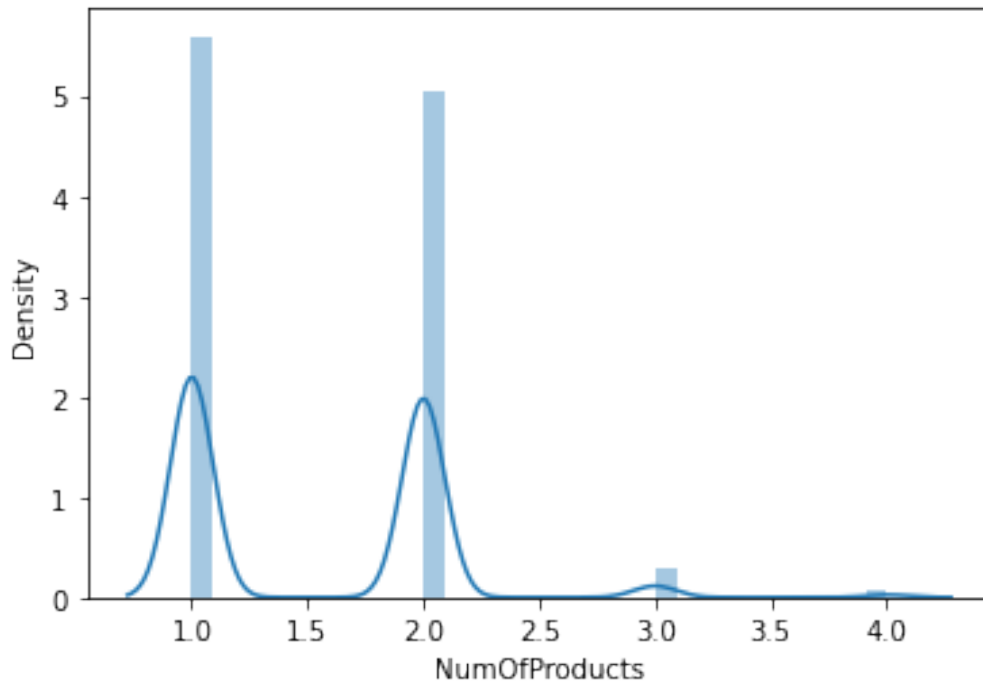
```
AxesSubplot(0.125,0.125;0.775x0.755)
```

```
print(sns.distplot(dataset["Tenure"]))
AxesSubplot(0.125,0.125;0.775x0.755)
```



```
print(sns.distplot(dataset["NumOfProducts"]))
AxesSubplot(0.125,0.125;0.775x0.755)
```



```
dataset.max()
```

```

RowNumber      10000
CustomerId     15815690
Surname        Zuyeva
CreditScore    850
Geography      Spain
Gender         Male
Age            92
Tenure         10
Balance        250898.09
NumOfProducts  4
HasCrCard      1
IsActiveMember 1
EstimatedSalary 199992.48
Exited         1
dtype: object

```

```
dataset.min()
```

```

RowNumber      1
CustomerId     15565701
Surname        Abazu
CreditScore    350
Geography      France
Gender         Female
Age            18
Tenure         0
Balance        0.0

```

```
NumOfProducts      1
HasCrCard           0
IsActiveMember      0
EstimatedSalary    11.58
Exited              0
dtype: object
```

```
dataset.columns
```

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

```
dataset.count()
```

```
RowNumber      10000
CustomerId     10000
Surname        10000
CreditScore    10000
Geography      10000
Gender         10000
Age            10000
Tenure         10000
Balance        10000
NumOfProducts  10000
HasCrCard      10000
IsActiveMember 10000
EstimatedSalary 10000
Exited         10000
dtype: int64
```

```
dataset.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
```

```
dataset.isnull().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
```

```
dataset.dropna()
```

	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]

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

quantile

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	\
0.75	7500.25	15753233.75	718.0	44.0	7.0	127644.24	
0.25	2500.75	15628528.25	584.0	32.0	3.0	0.00	

	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
Exited				
0.75	2.0	1.0	1.0	149388.2475
0.0				
0.25	1.0	0.0	0.0	51002.1100
0.0				

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

iqr

RowNumber	4999.5000
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
```

```
upper = quantile.iloc[0] + (1.5*iqr)
upper
```

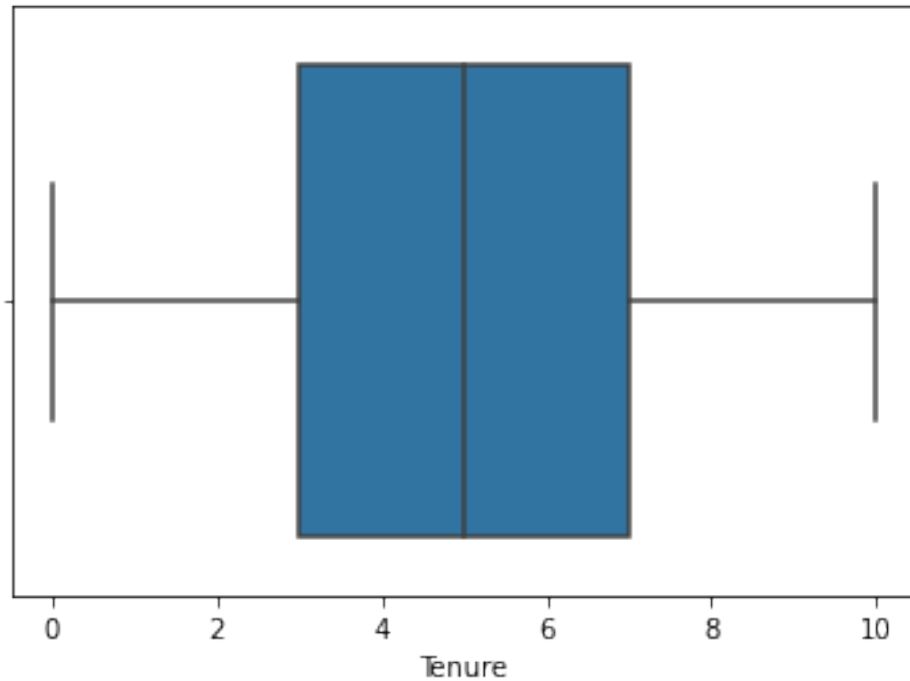
```
RowNumber      1.499950e+04
CustomerId      1.594029e+07
CreditScore     9.190000e+02
Age             6.200000e+01
Tenure          1.300000e+01
Balance         3.191106e+05
NumOfProducts   3.500000e+00
HasCrCard       2.500000e+00
IsActiveMember  2.500000e+00
EstimatedSalary 2.969675e+05
Exited          0.000000e+00
dtype: float64
```

```
lower = quantile.loc[0.25] - 1.5*iqr
lower
```

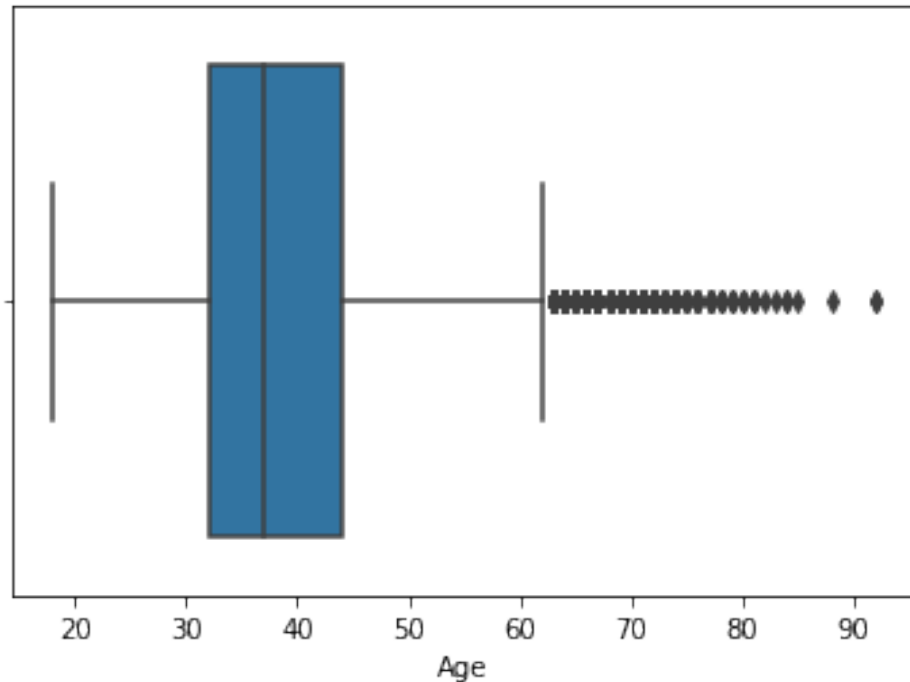
```
RowNumber      -4.998500e+03
CustomerId      1.544147e+07
CreditScore     3.830000e+02
Age             1.400000e+01
Tenure          -3.000000e+00
Balance        -1.914664e+05
NumOfProducts   -5.000000e-01
HasCrCard       -1.500000e+00
IsActiveMember  -1.500000e+00
EstimatedSalary -9.657710e+04
Exited          0.000000e+00
dtype: float64
```

```
sb.boxplot(dataset.Tenure)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9233068090>
```

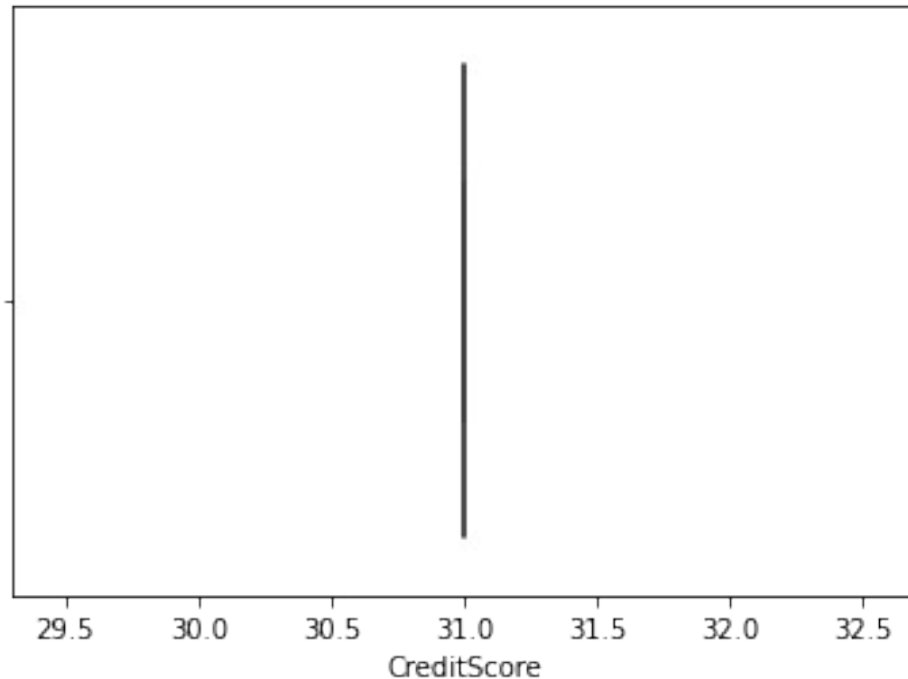


```
sb.boxplot(dataset["Age"])
<matplotlib.axes._subplots.AxesSubplot at 0x7f9233082290>
```



```
dataset["Age"] = np.where(dataset["Age"]>87,40,dataset["Age"])
dataset["CreditScore"] =
np.where(dataset["CreditScore"]>45,31,dataset["CreditScore"])
sb.boxplot(dataset["CreditScore"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9234228e90>
```



```
from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
dataset["Gender"] = label.fit_transform(dataset["Gender"])
dataset.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	31	France	0	42
1	2	15647311	Hill	31	Spain	0	41
2	3	15619304	Onio	31	France	0	42
3	4	15701354	Boni	31	France	0	39
4	5	15737888	Mitchell	31	Spain	0	43

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	1
1	1	83807.86	1	0	1	1
2	8	159660.80	3	1	0	0
3	1	0.00	2	0	0	0
4	2	125510.82	1	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

```

```
dataset.Gender.unique()
```

```
array([0, 1])
```

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

```

x = data.iloc[:,0:4].values
y = data.iloc[:,4:5].values

```

```

from sklearn.preprocessing import OneHotEncoder
one = OneHotEncoder()
z = one.fit_transform(x[:,3:4]).toarray()
z

```

```

array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],

```

```

...
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.])

```

```

x = data.iloc[:,1:]
y = data.iloc[:,0]
print(x)
print(y)

```

	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						
...
.						
9995	15606229	Obijiaku	771	France	Male	39
5						
9996	15569892	Johnstone	516	France	Male	35
10						
9997	15584532	Liu	709	France	Female	36
7						
9998	15682355	Sabbatini	772	Germany	Male	42
3						
9999	15628319	Walker	792	France	Female	28
4						

	Balance	NumOfProducts	HasCrCard	IsActiveMember
EstimatedSalary \				
0	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				
...
...				
9995	0.00	2	1	0

```

96270.64
9996    57369.61      1      1      1
101699.77
9997      0.00      1      0      1
42085.58
9998    75075.31      2      1      0
92888.52
9999   130142.79      1      1      0
38190.78

```

```

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

```

```
[10000 rows x 13 columns]
```

```

0          1
1          2
2          3
3          4
4          5

```

```

...
9995      9996
9996      9997
9997      9998
9998      9999
9999     10000

```

```
Name: RowNumber, Length: 10000, dtype: int64
```

```

x = dataset.drop(columns=["Exited"],axis = 1)
x.head()

```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
0	1	15634602	Hargrave	31	France	0	42
1	2	15647311	Hill	31	Spain	0	41
2	3	15619304	Onio	31	France	0	42
3	4	15701354	Boni	31	France	0	39

4	5	15737888	Mitchell	31	Spain	0	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
0	101348.88
1	112542.58
2	113931.57
3	93826.63
4	79084.10

```
Name = x.columns
Name
```

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

```
from sklearn.preprocessing import scale
x = pd.DataFrame(x,columns = Name)
x
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age \						
0	1	15634602	Hargrave	31	France	0
42						
1	2	15647311	Hill	31	Spain	0
41						
2	3	15619304	Onio	31	France	0
42						
3	4	15701354	Boni	31	France	0
39						
4	5	15737888	Mitchell	31	Spain	0
43						
...
...						
9995	9996	15606229	Obijiaku	31	France	1
39						
9996	9997	15569892	Johnstone	31	France	1
35						
9997	9998	15584532	Liu	31	France	0
36						

9998	9999	15682355	Sabbatini	31	Germany	1
42						
9999	10000	15628319	Walker	31	France	0
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
0	101348.88
1	112542.58
2	113931.57
3	93826.63
4	79084.10
...	...
9995	96270.64
9996	101699.77
9997	42085.58
9998	92888.52
9999	38190.78

[10000 rows x 13 columns]

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

(8000, 13) (2000, 13)