In [4]:
```
import pandas as pd
df = pd.read_csv("Churn_Modelling.csv")
df.head()
```

Out[4]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86
2	3	15619304	Onio	502	France	Female	42	8	159660.80
3	4	15701354	Boni	699	France	Female	39	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82

UNIVARIATE ANALYSIS:

In [6]:
```
df['Age'].mean()
```

Out[6]:
```
38.9218
```

In [7]:
```
df['Age'].median()
```

Out[7]:
```
37.0
```

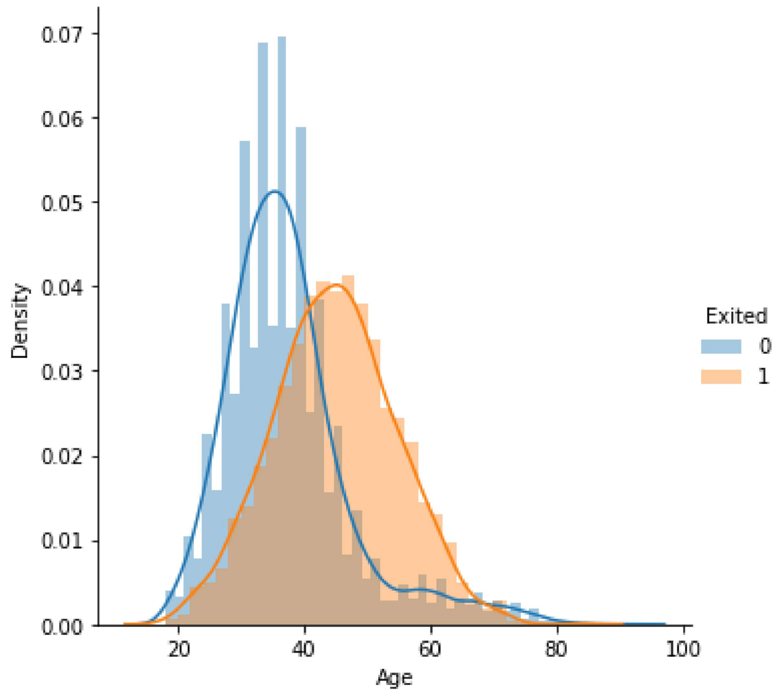
In [8]:
```
df['Age'].mode()
```

Out[8]:
```
0    37
dtype: int64
```

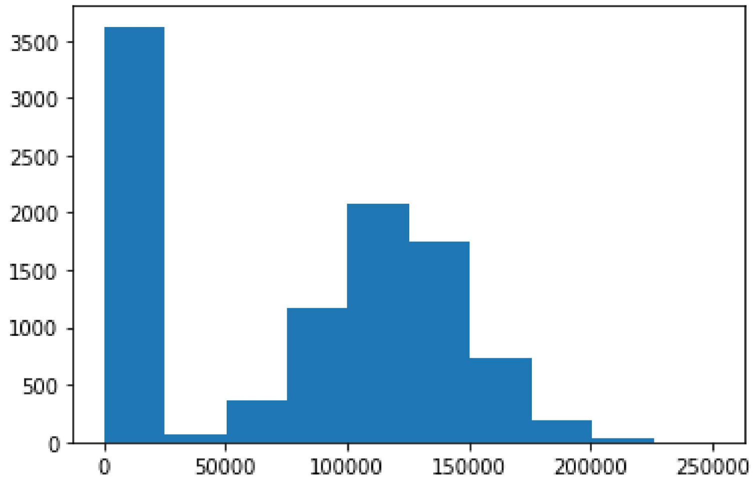
In [9]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
sns.FacetGrid(df, hue = "Exited", size=5).map(sns.distplot, 'Age').add_legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning: Th
e `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWar
ning: `distplot` is a deprecated function and will be removed in a future versi
on. Please adapt your code to use either `displot` (a figure-level function wit
h similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWar
ning: `distplot` is a deprecated function and will be removed in a future versi
on. Please adapt your code to use either `displot` (a figure-level function wit
h similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

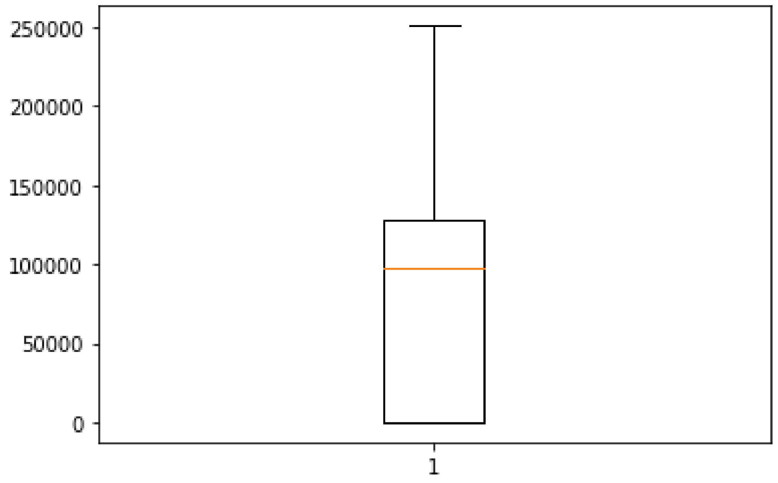
Out[9]:
```
<seaborn.axisgrid.FacetGrid at 0x7f292601b690>
```



In [10]:
```
plt.hist(df['Balance'])
plt.show()
```

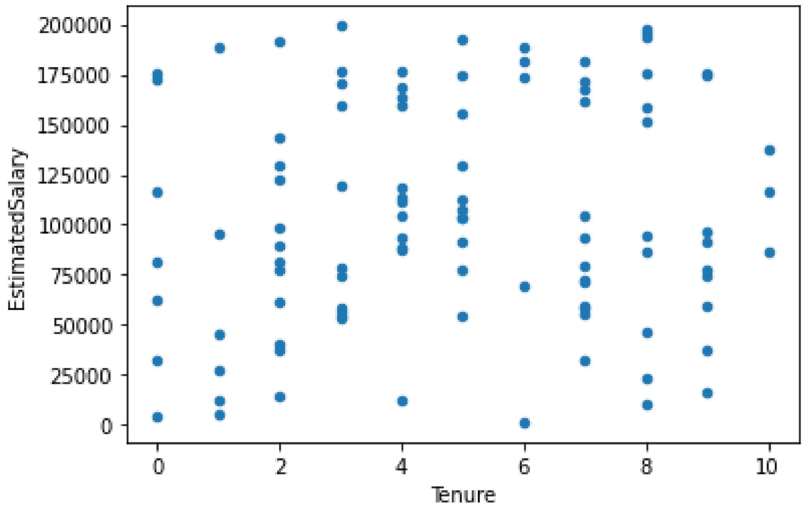


In [11]:
```
plt.boxplot(df['Balance'])
plt.show()
```

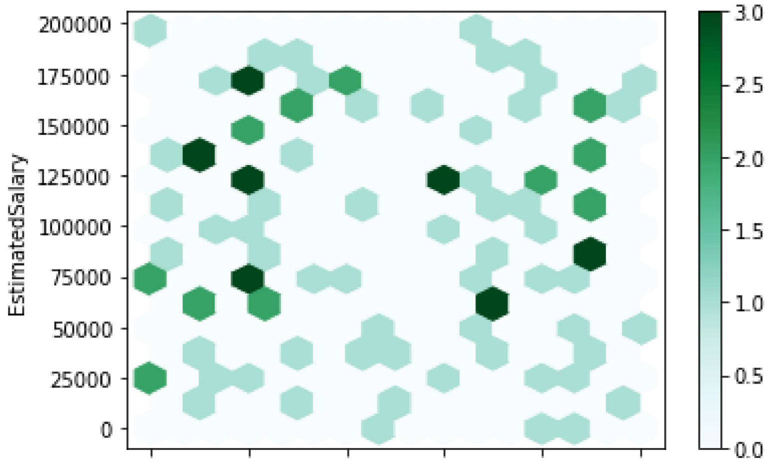


BIVARIATE ANALYSIS:

```
In [12]: df.sample(100).plot.scatter(x='Tenure',y='EstimatedSalary')
         plt.show()
```

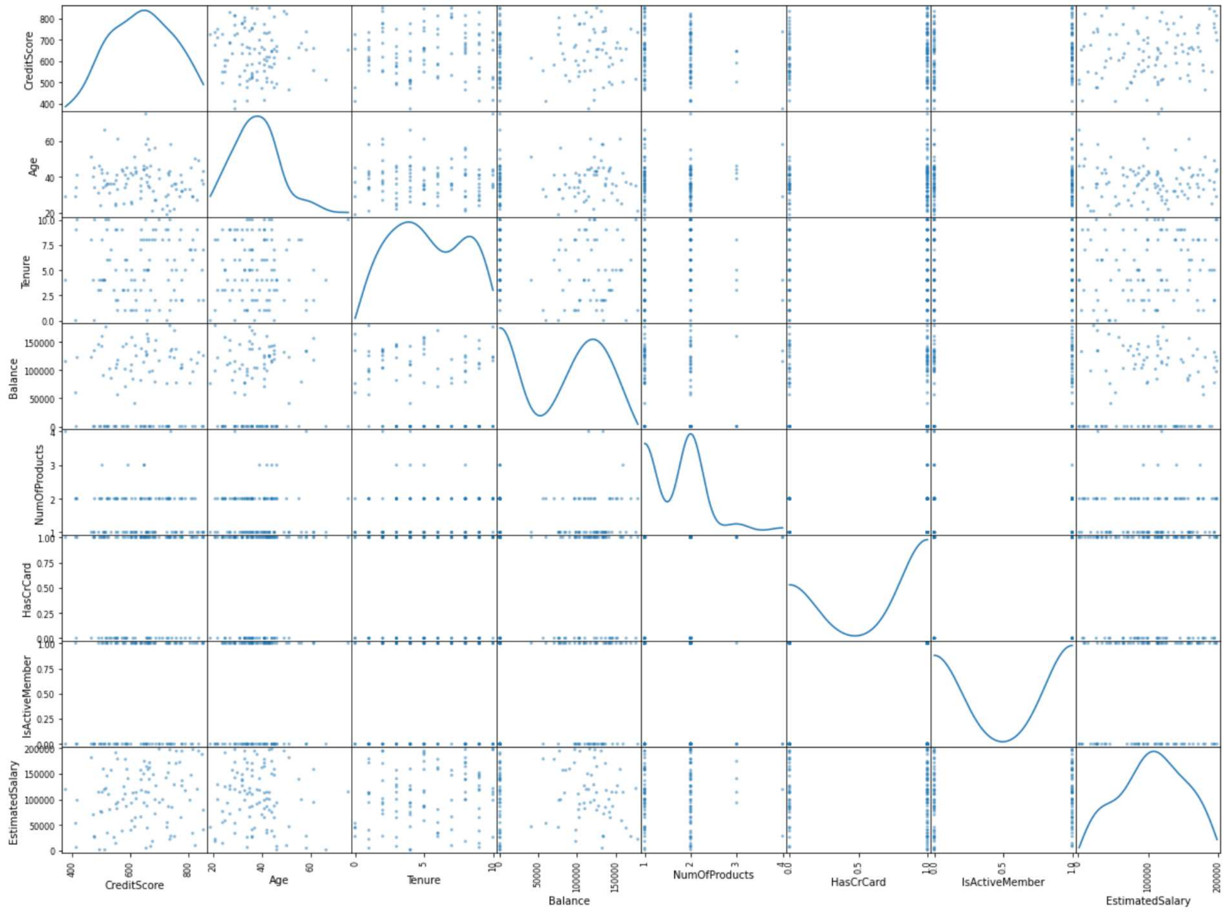


```
In [13]: df.sample(100).plot.hexbin(x='Tenure',y='EstimatedSalary',gridsize=15)
         plt.show()
```



MULTI-VARIATE ANALYSIS

In [14]:
```
pd.plotting.scatter_matrix(df.loc[:100,'CreditScore':'EstimatedSalary'], diagonal
plt.show()
```



In [15]:
```
df.describe()
```

Out[15]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumC
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	

HANDLING MISSING VALUES

In [16]:
```
df.isnull().any()
```

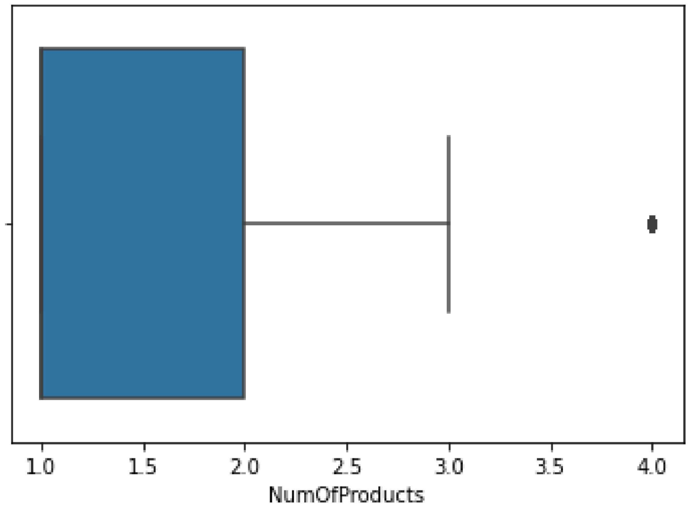
Out[16]:
```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

FINDING OUTLIERS AND REPLACING THEM USING IQR

```
In [17]: sns.boxplot(df.NumOfProducts)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: 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 a
n explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

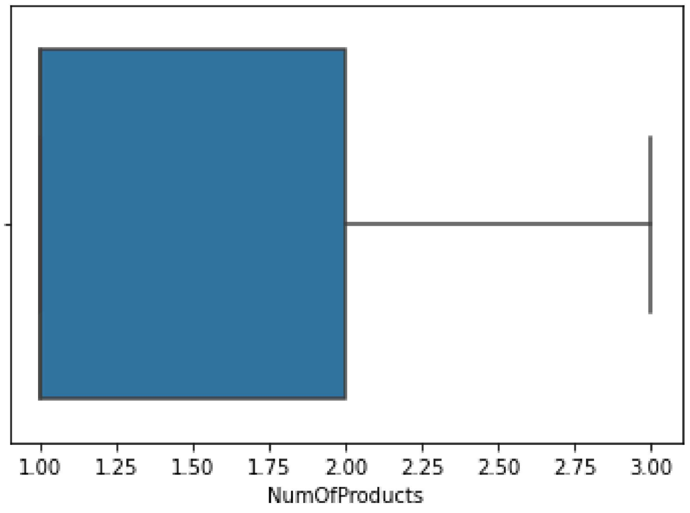
```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28f59523d0>
```



In [18]:
```
Q1 = df.NumOfProducts.quantile(0.25)
Q3 = df.NumOfProducts.quantile(0.75)
IQR = Q3 - Q1
upperLimit = Q3 + 1.5 * IQR
lowerLimit = Q1 - 1.5 * IQR
df = df[df.NumOfProducts < upperLimit]
sns.boxplot(df.NumOfProducts)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: 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 a
n explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f28f58d9cd0>
```



ENCODING

In [19]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df.Geography = le.fit_transform(df.Geography)
df.head()
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:5516: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  self[name] = value
```

Out[19]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	0	Female	42	2	0.00
1	2	15647311	Hill	608	2	Female	41	1	83807.86
2	3	15619304	Onio	502	0	Female	42	8	159660.80
3	4	15701354	Boni	699	0	Female	39	1	0.00
4	5	15737888	Mitchell	850	2	Female	43	2	125510.82

One hot encoding:

In [20]:
```
df_main = pd.get_dummies(df,columns = ['Gender'])
df_main.head()
```

Out[20]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Age	Tenure	Balance	NumOfPr
0	1	15634602	Hargrave	619	0	42	2	0.00	
1	2	15647311	Hill	608	2	41	1	83807.86	
2	3	15619304	Onio	502	0	42	8	159660.80	
3	4	15701354	Boni	699	0	39	1	0.00	
4	5	15737888	Mitchell	850	2	43	2	125510.82	

SPLITTING DATA INTO DEPENDANT AND INDEPENDANT VARIABLES:

In [21]:

```python
#Independent variable X
X = df_main.drop(columns = ['Exited','Surname'], axis = 1)
X.head()
```

Out[21]:

	RowNumber	CustomerId	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	Ha
0	1	15634602	619	0	42	2	0.00	1	
1	2	15647311	608	2	41	1	83807.86	1	
2	3	15619304	502	0	42	8	159660.80	3	
3	4	15701354	699	0	39	1	0.00	2	
4	5	15737888	850	2	43	2	125510.82	1	

In [22]:

```python
#Dependent variable Y
Y = df_main.Exited
Y.head()
```

Out[22]:

```
0    1
1    0
2    1
3    0
4    0
Name: Exited, dtype: int64
```

SCALING THE INDEPENDANT VARIABLE:

In [23]:

```python
from sklearn.preprocessing import scale
X_scaled = pd.DataFrame(scale(X), columns = X.columns)
X_scaled.head()
```

Out[23]:

	RowNumber	CustomerId	CreditScore	Geography	Age	Tenure	Balance	NumOfProduc
0	-1.730861	-0.784231	-0.326110	-0.901890	0.297483	-1.041259	-1.223855	-0.9356
1	-1.730515	-0.607593	-0.439952	1.512868	0.202106	-1.387070	0.118987	-0.9356
2	-1.730169	-0.996853	-1.536977	-0.901890	0.297483	1.033605	1.334368	2.6959
3	-1.729823	0.143532	0.501833	-0.901890	0.011351	-1.387070	-1.223855	0.8801
4	-1.729476	0.651305	2.064576	1.512868	0.392860	-1.041259	0.787188	-0.9356

TRAING THE TEST SPLIT:

In [24]:
```python
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,
X_train.head()
```

Out[24]:

	RowNumber	CustomerId	CreditScore	Geography	Age	Tenure	Balance	NumOfPr
5833	0.301456	0.603674	-0.916020	0.305489	-0.751667	-1.041259	0.582053	-0.
9935	1.730207	-1.178578	1.246982	-0.901890	0.011351	-0.003827	-1.223855	0.
863	-1.431329	-0.056206	0.284498	0.305489	-1.323931	1.725226	0.009395	0.
8866	1.357263	-1.559984	-0.098426	-0.901890	1.251256	-0.349638	-1.223855	-0.
3761	-0.421923	-0.079736	-0.574493	0.305489	0.678992	-0.003827	1.057170	0.

In [25]:
```python
X_train.shape
```

Out[25]:
```
(6958, 13)
```

In [26]:
```python
Y_train.shape
```

Out[26]:
```
(6958,)
```