Descriptive Statistics

Python Programming Lab

05506231 Statistics and Probability

Asst. Prof. Dr.Anantaporn Hanskunatai

Outline

- Churn_Modelling Dataset
- Descriptive Statistics for Numeric Data
- Descriptive Statistics Categorical Data
- Data Visualization

Churn_Modelling Dataset

Row Number	CustomerId	Surname	Credit Score	Geography	Gender	Age	Tenure	Balance	NumOf Product	HasCr Card	IsActive Member	Estimated Salary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.6	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.8	1	1	1	79084.1	0
6	15574012	Chu	645	Spain	Male	44	8	113755.8	2	1	0	149756.7	1
7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
8	15656148	Obinna	376	Germany	Female	29	4	115046.7	4	1	0	119346.9	1
9	15792365	He	501	France	Male	44	4	142051.1	2	0	1	74940.5	0
10	15592389	H?	684	France	Male	27	2	134603.9	1	1	1	71725.73	0
11	15767821	Bearce	528	France	Male	31	6	102016.7	2	0	0	80181.12	0
12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0

Churn_Modelling Dataset

```python
from google.colab import files
uploaded = files.upload()
```

```python
import pandas as pd
df = pd.read_csv( 'Churn_Modelling.csv' )
```

```python
df.dtypes
```

```
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
```

Row Number	CustomerId	Surname	Credit Score	Geography	Gender	Age	Tenure	Balance	NumOf Product	HasCr Card	IsActive Member	Estimated Salary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.9	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.6	0

Descriptive Statistics for Numeric Data

```
df.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881	0.203700
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818	0.402769
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000	0.000000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000	0.000000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000	0.000000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500	0.000000
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000	1.000000

Mode

```
df.mode()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.92	0.0
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
df.mode(numeric_only=True)
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15565701	850.0	37.0	2.0	0.0	1.0	1.0	1.0	24924.92	0.0
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
df["Gender"].mode()
```

```
0    Male
dtype: object
```

```
df["Age"].mode()
```

```
0    37
dtype: int64
```

```
df["Surname"].mode()
```

```
0    Smith
dtype: object
```

Variance/ Coefficient of Variation

```
df.var()
```

```
df.var()
```

```
RowNumber          8.334167e+06
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
```

```
df.var()['Age']
```

```
df.var()['Age']
```

```
109.99408416841645
```

```
from scipy.stats import variation
variation(df['Age'])
```

```
from scipy.stats import variation
variation(df['Age'])
```

```
0.26944493955242593
```

Descriptive Statistics for Categorical Data

```
df.describe(exclude=['float', 'int64'])
```

```
df.describe(exclude=['float', 'int64'])
```

	Surname	Geography	Gender
count	10000	10000	10000
unique	2932	3	2
top	Smith	France	Male
freq	32	5014	5457

```
df.describe(include = 'object')
```

```
df.describe(include = 'object')
```

	Surname	Geography	Gender
count	10000	10000	10000
unique	2932	3	2
top	Smith	France	Male
freq	32	5014	5457

Convert Data Type

```
df.RowNumber=df.RowNumber.astype('category')
df.CustomerId=df.CustomerId.astype('category')
df.HasCrCard=df.HasCrCard.astype('category')
df.IsActiveMember=df.IsActiveMember.astype('category')
df.Exited=df.Exited.astype('category')
df.NumOfProducts=df.NumOfProducts.astype('category')
```

```
df.Geography = df.Geography.astype('category')
df.Surname = df.Surname.astype('category')
df.Gender = df.Gender.astype('category')
```

```
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.dtypes
```

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

Descriptive Statistics for Categorical Data

```
df.describe(include = 'category')
```

```
df.describe(include = 'category')
```

	RowNumber	CustomerId	Surname	Geography	Gender	NumOfProducts	HasCrCard	IsActiveMember	Exited
count	10000	10000	10000	10000	10000	10000	10000	10000	10000
unique	10000	10000	2932	3	2	4	2	2	2
top	10000	15815690	Smith	France	Male	1	1	1	0
freq	1	1	32	5014	5457	5084	7055	5151	7963

```
df.Geography.value_counts()
```

```
df.Geography.value_counts()
```

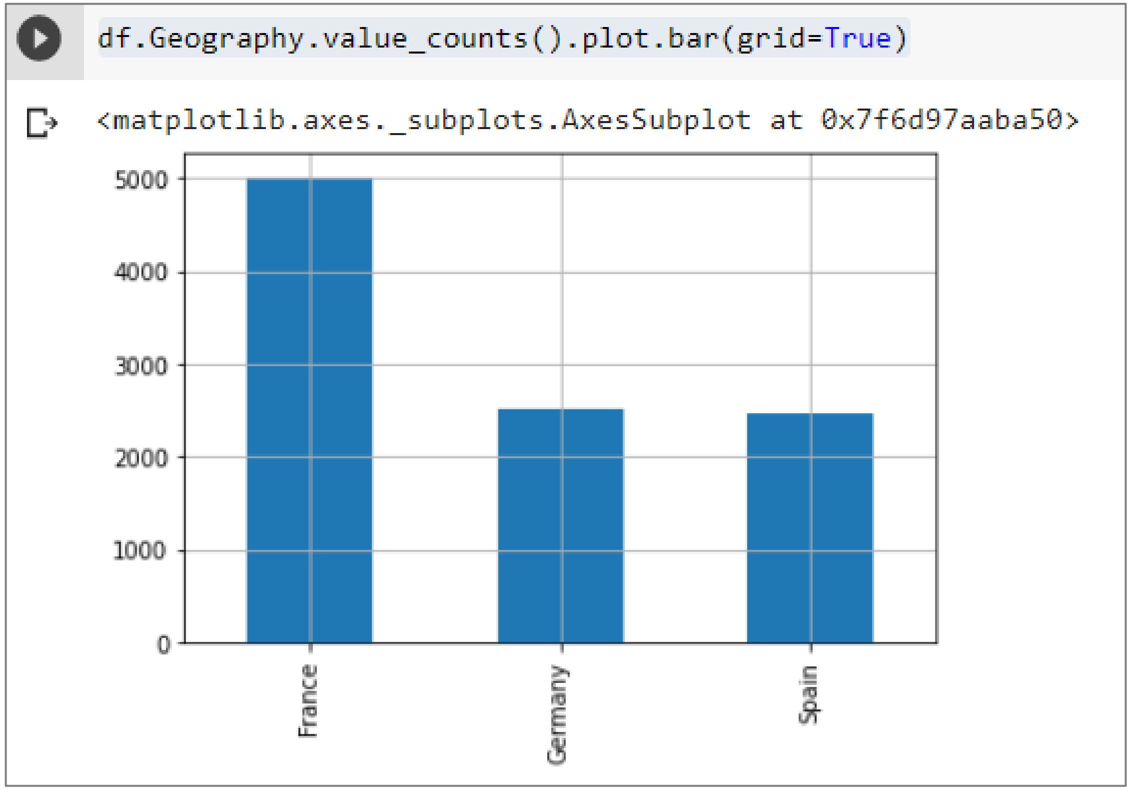
```
France     5014
Germany    2509
Spain      2477
Name: Geography, dtype: int64
```

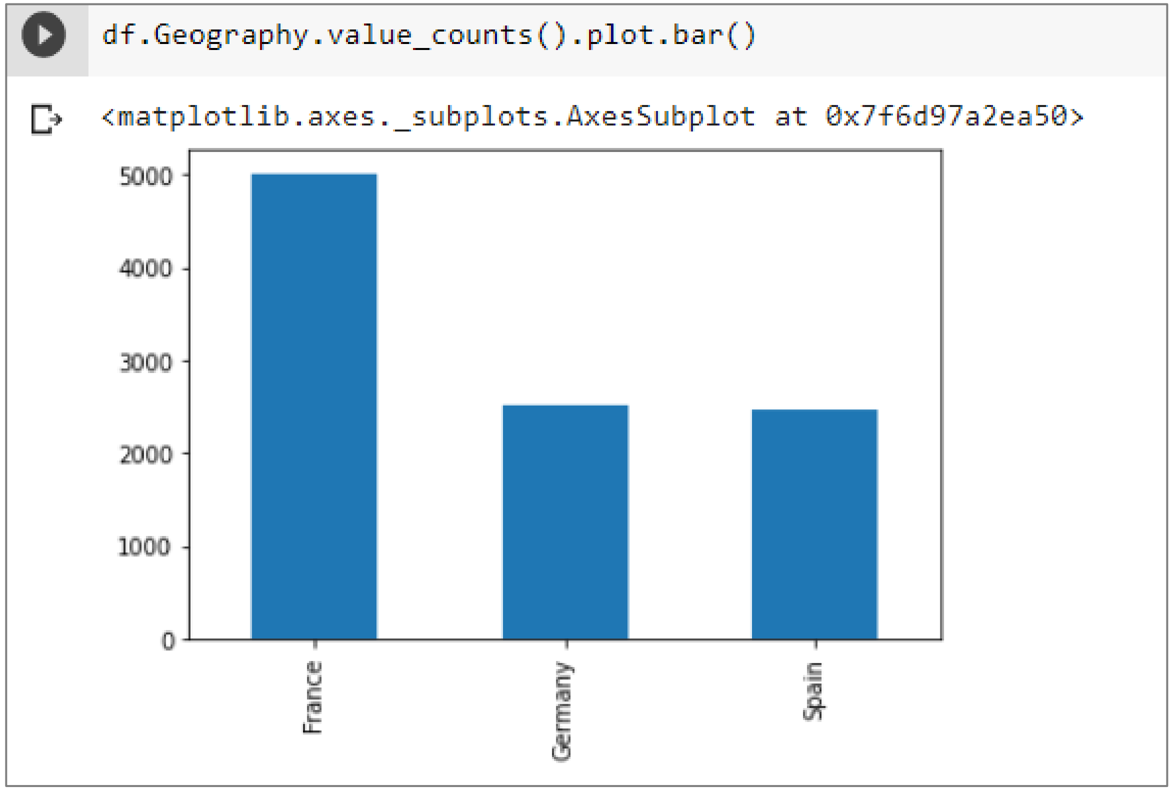
Data Visualization

- Bar charts
- Box plot
- Histogram

Bar Charts

```
df.Geography.value_counts().plot.bar(grid=True)
```





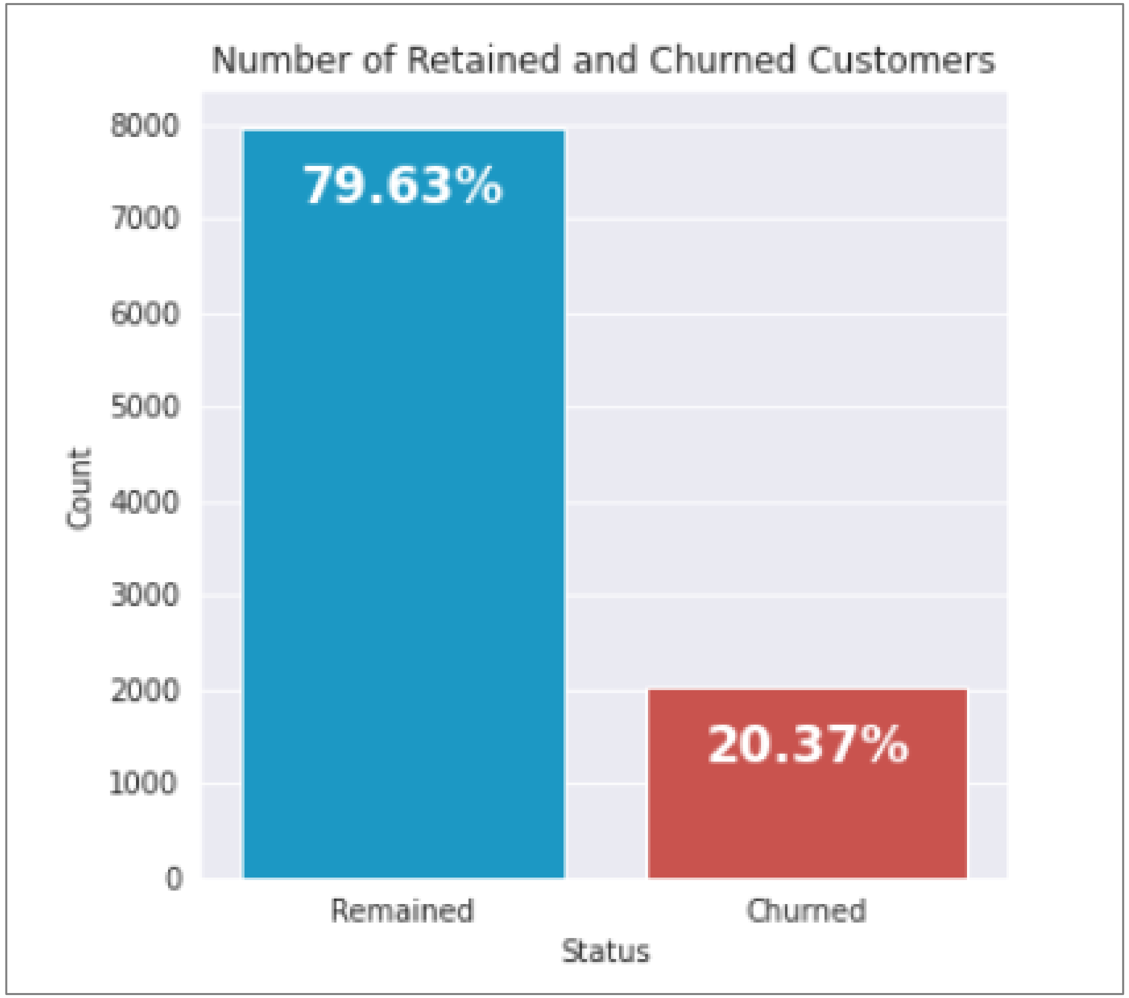
Bar Chart

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
colors = ['#00A5E0', '#DD403A']

fig = plt.figure(figsize = (5, 5))
sns.countplot(x = 'Exited', data = df, palette = colors)

for index, value in enumerate(df['Exited'].value_counts()):
    label =  '{}%'.format(round( (value/df['Exited'].shape[0])*100, 2))
    plt.annotate(label, xy = (index -0.25, value -800), color = 'w',fontweight='bold',size=17)
plt.title('Number of Retained and Churned Customers')
plt.xticks([0, 1], ['Remained', 'Churned'])
plt.xlabel('Status')
plt.ylabel('Count');
```

Bar Chart



Number of Retained and Churned Customers
8000
7000
6000
5000
4000
3000
2000
1000
0
Count
79.63%
20.37%
Remained
Churned
Status

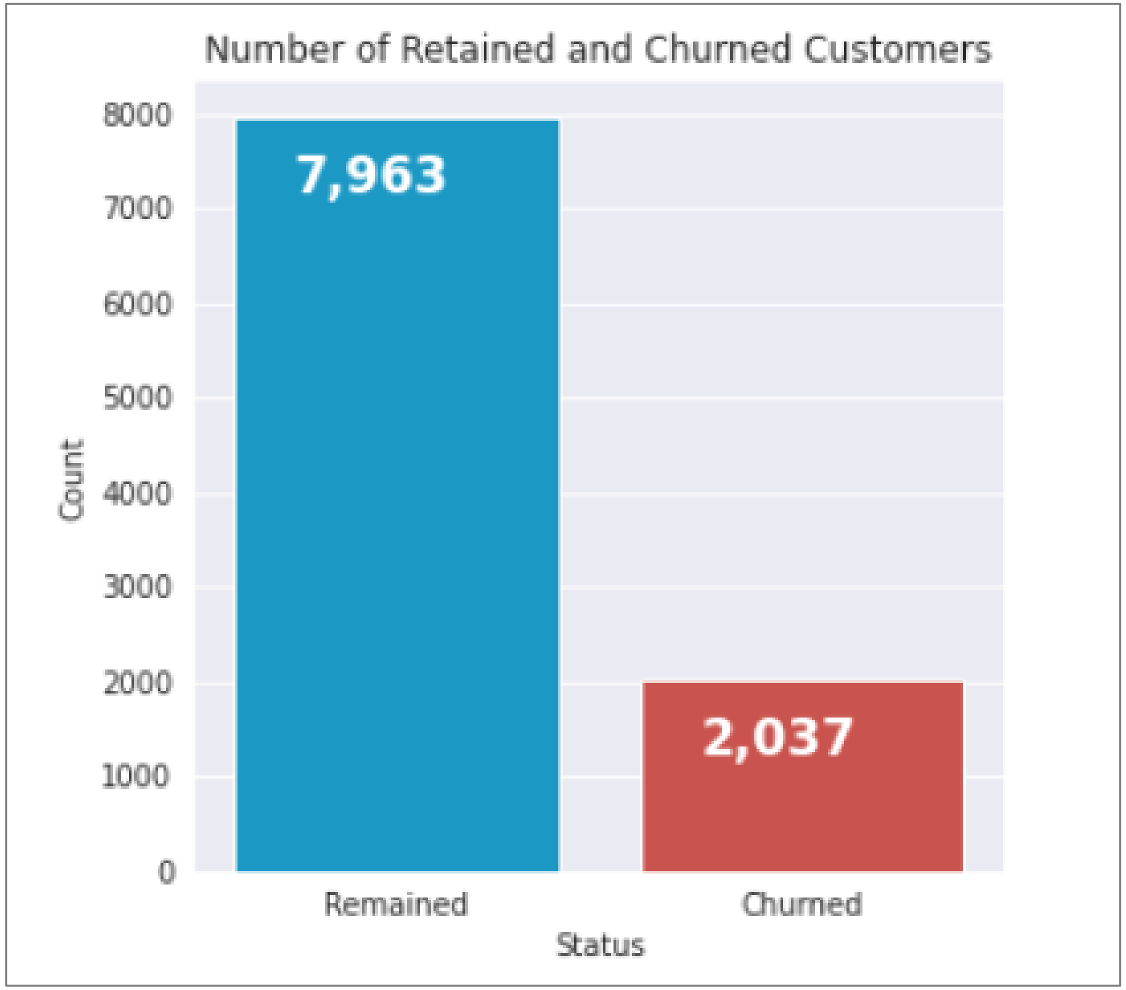
Bar Chart

```
sns.set_style('darkgrid')
colors = ['#00A5E0', '#DD403A']

fig = plt.figure(figsize = (5, 5))
sns.countplot(x = 'Exited', data = df, palette = colors)

for index, value in enumerate(df['Exited'].value_counts()):
    label = '{:,}'.format(value)
    plt.annotate(label, xy = (index -0.25, value -800), color = 'w',fontweight='bold',size=17)
plt.title('Number of Retained and Churned Customers')
plt.xticks([0, 1], ['Remained', 'Churned'])
plt.xlabel('Status')
plt.ylabel('Count');
```

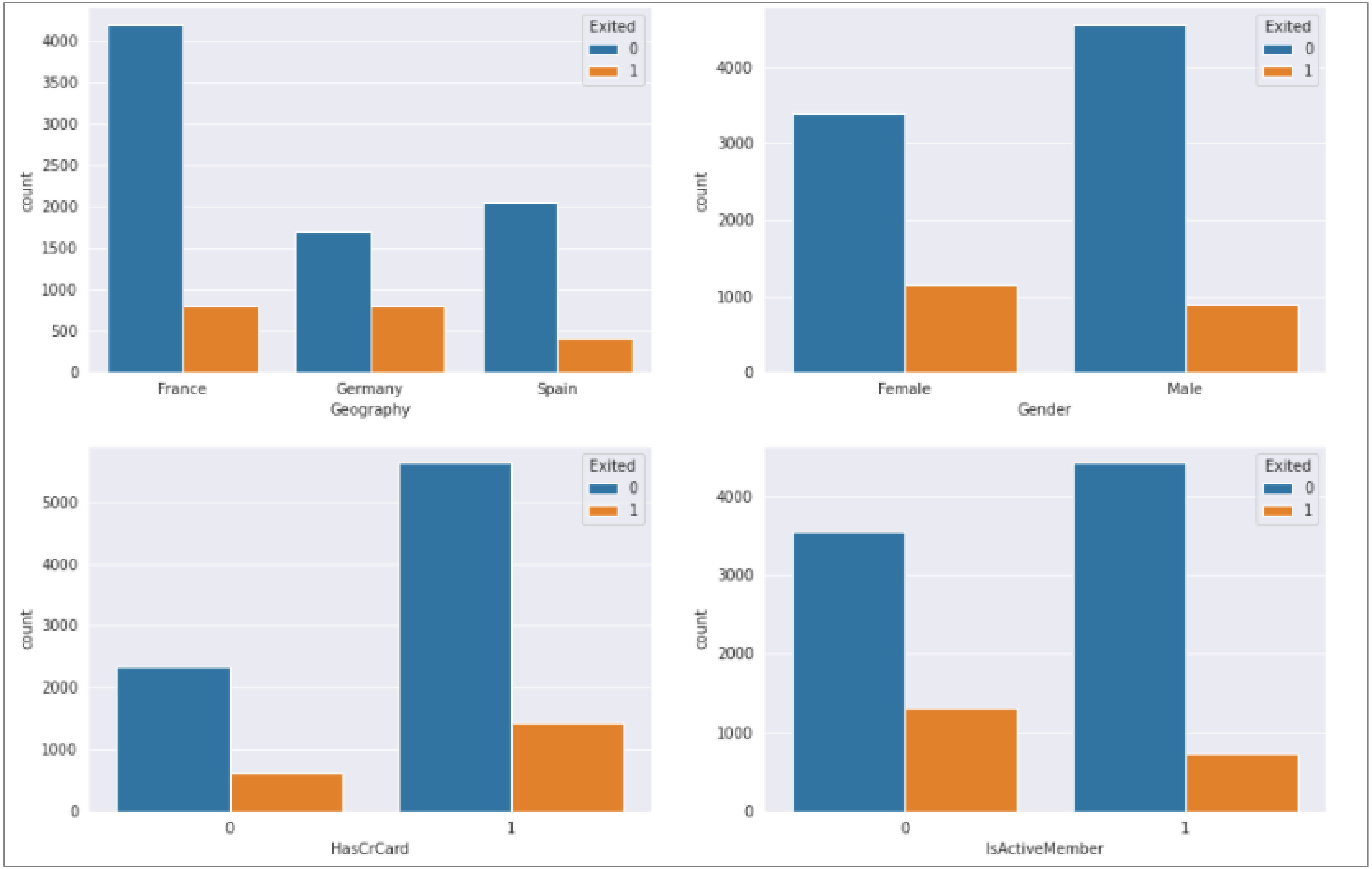
Bar Chart



Number of Retained and Churned Customers
8000
7000
6000
5000
4000
3000
2000
1000
0
Count
7,963
2,037
Remained
Churned
Status

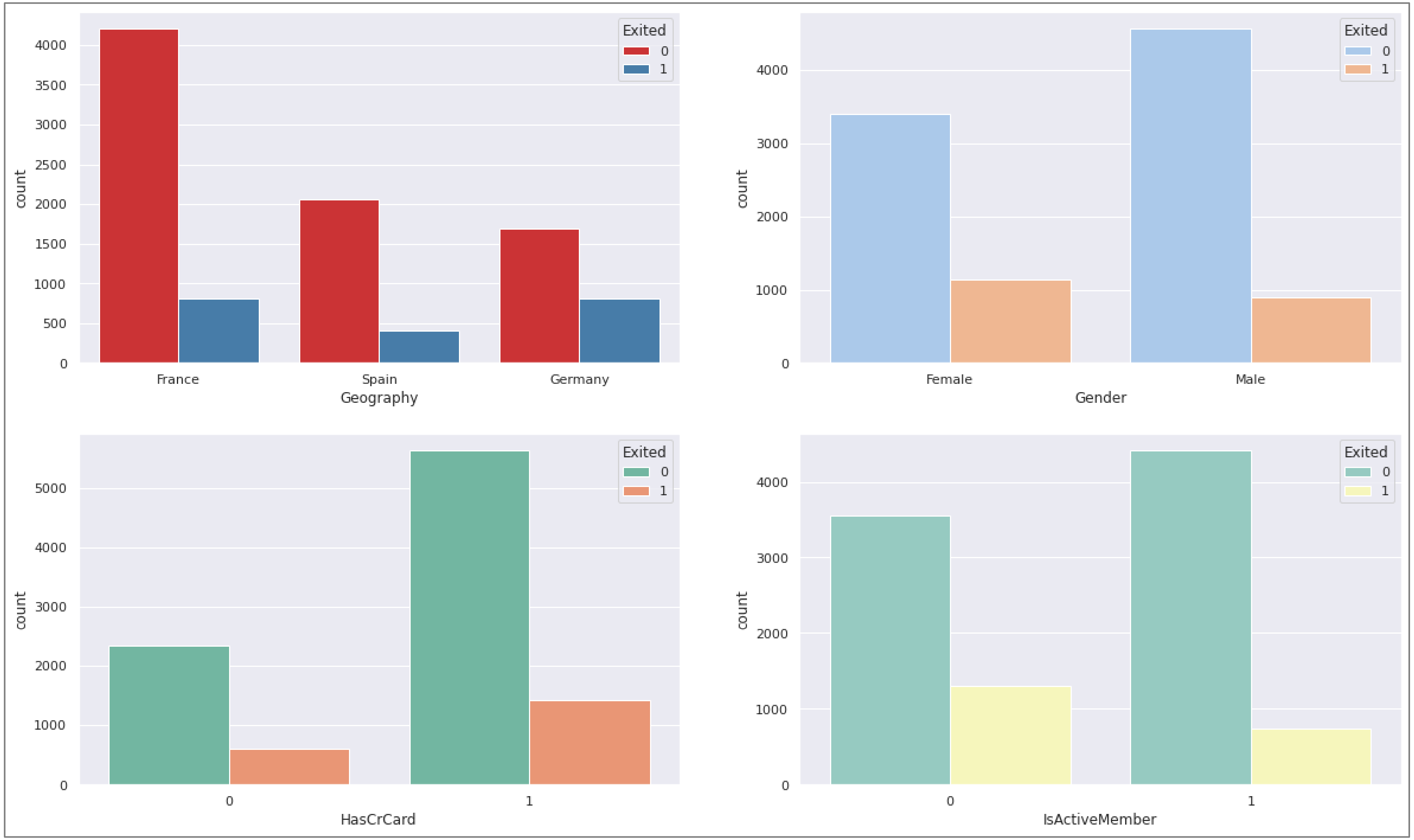
Bar Chart

```
fig, axarr = plt.subplots(2, 2, figsize=(20, 12))
sns.countplot(x='Geography', hue = 'Exited',data = df, ax=axarr[0][0])
sns.countplot(x='Gender', hue = 'Exited',data = df, ax=axarr[0][1])
sns.countplot(x='HasCrCard', hue = 'Exited',data = df, ax=axarr[1][0])
sns.countplot(x='IsActiveMember', hue = 'Exited',data = df, ax=axarr[1][1])
```



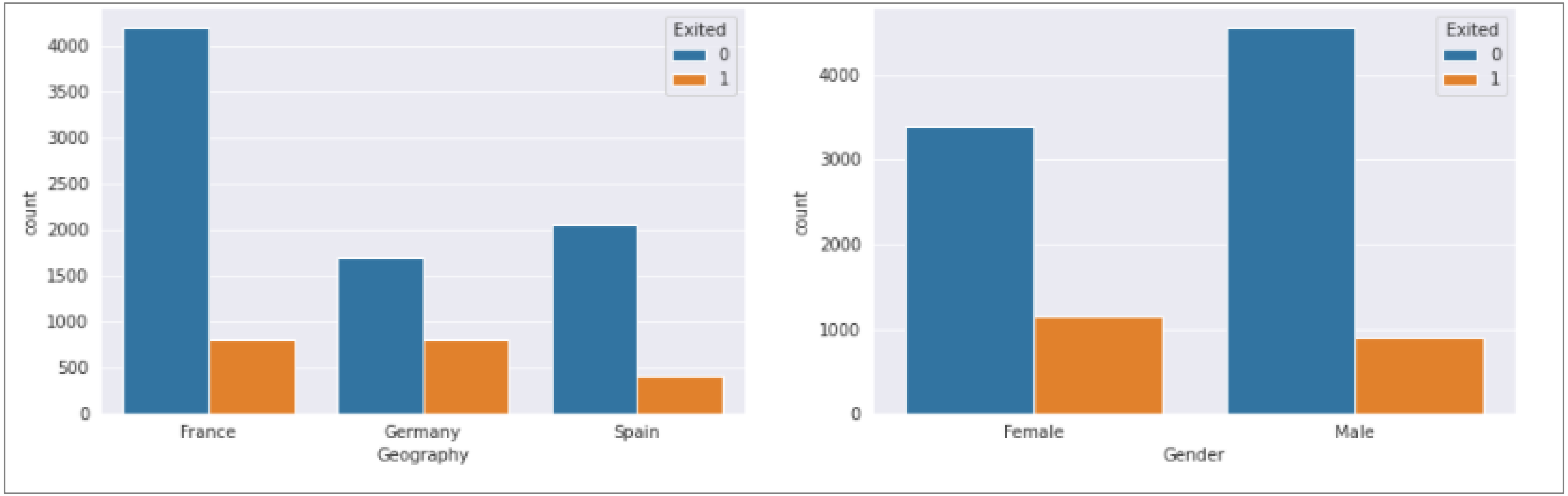
Exited
0
1
count
France
Germany
Spain
Geography
Female
Male
Gender
HasCrCard
IsActiveMember

```python
sns.countplot(x='Geography', hue='Exited',data = df, palette="Set1", ax=axarr[0][0])
sns.countplot(x='Gender', hue ='Exited',data = df, palette="pastel", ax=axarr[0][1])
sns.countplot(x='HasCrCard', hue='Exited',data = df, palette="Set2", ax=axarr[1][0])
sns.countplot(x='IsActiveMember', hue='Exited',data = df, palette="Set3", ax=axarr[1][1])
```



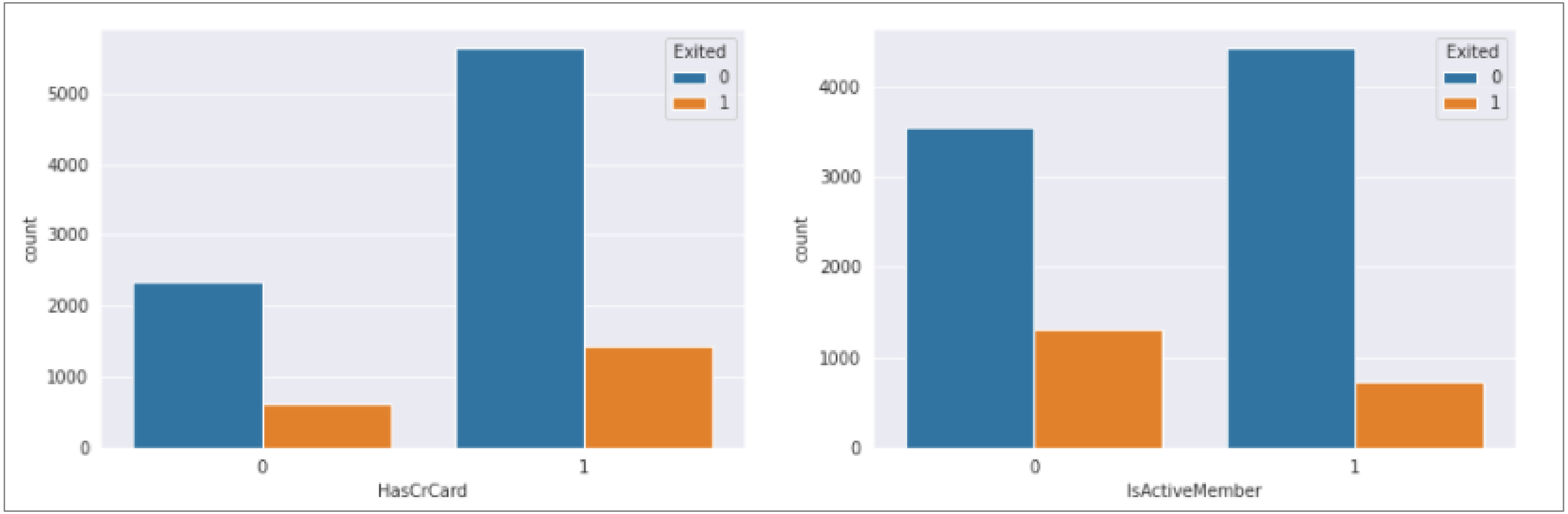
Bar Chart

- Majority of the data is from persons from France. Germany has the highest proportion of churned customers.
- The proportion of female customers churning is also greater than that of male customers



Bar Chart

- No different proportion of customer churning between HasCrCard and not have.
- The inactive members have a greater churn.



Box Plot

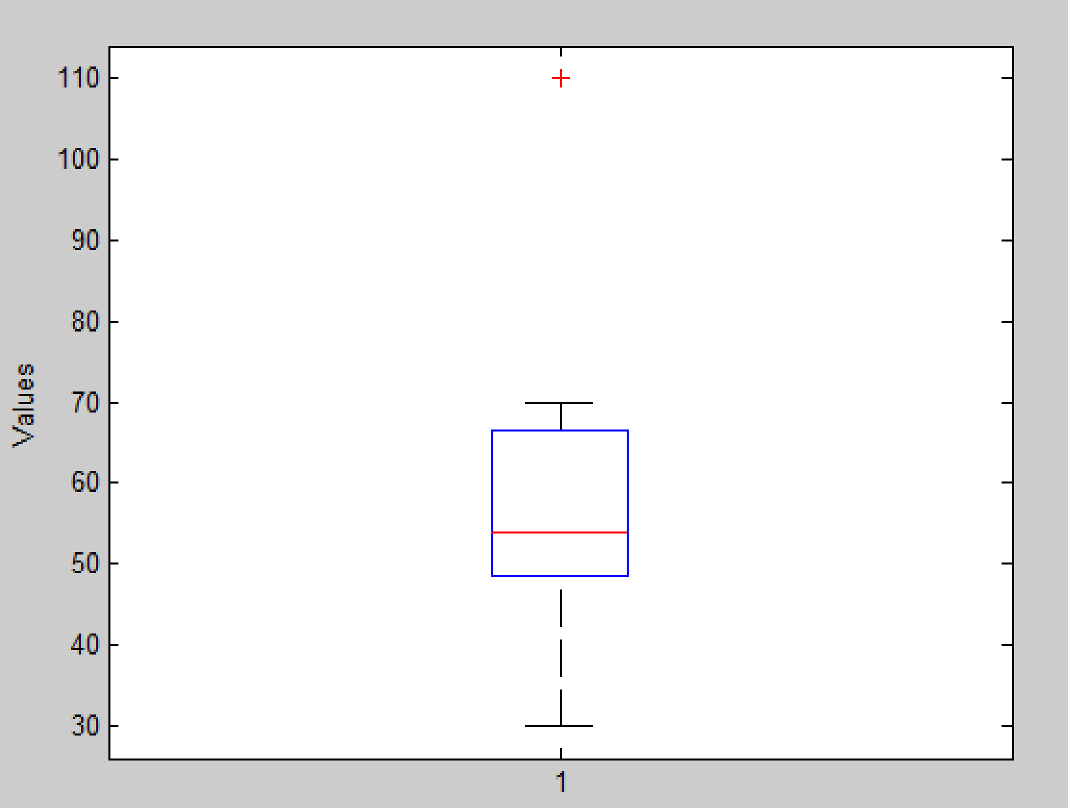
- Data is represented with a box
- The ends of the box are at the first and third quartiles, i.e., the height of the box is IQR

$$IQR = Q_3 - Q_1$$

- The median is marked by a line within the box
- Whiskers: two lines outside the box extended to Minimum and Maximum
- Outliers: points beyond a specified outlier threshold, plotted individually

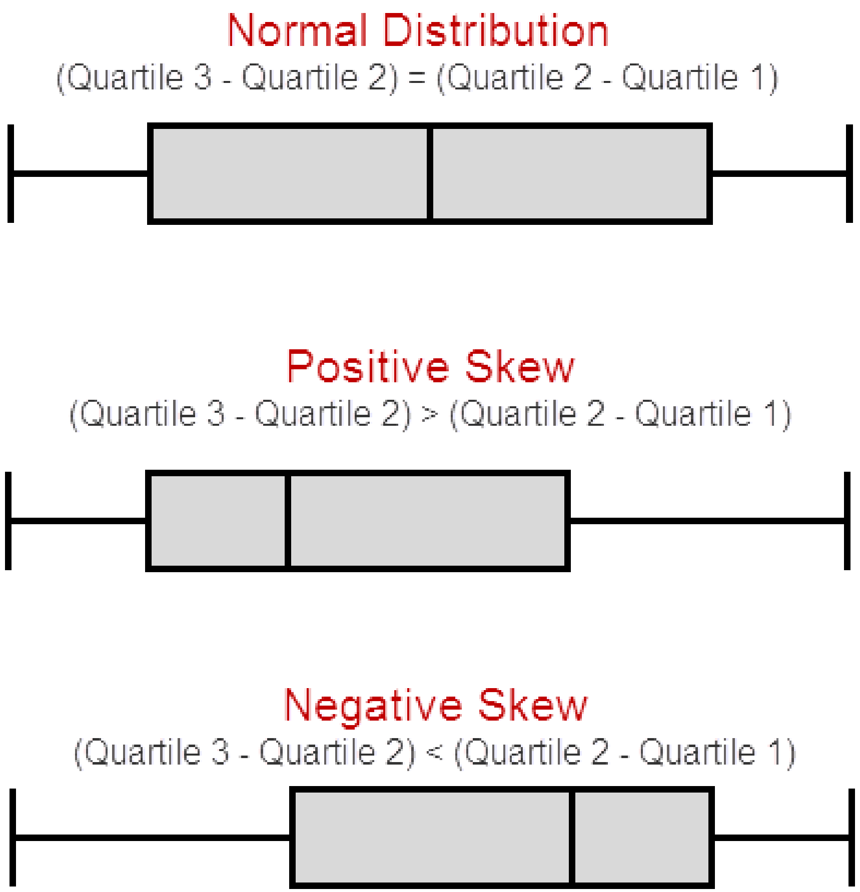
Box Plot

Example: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110

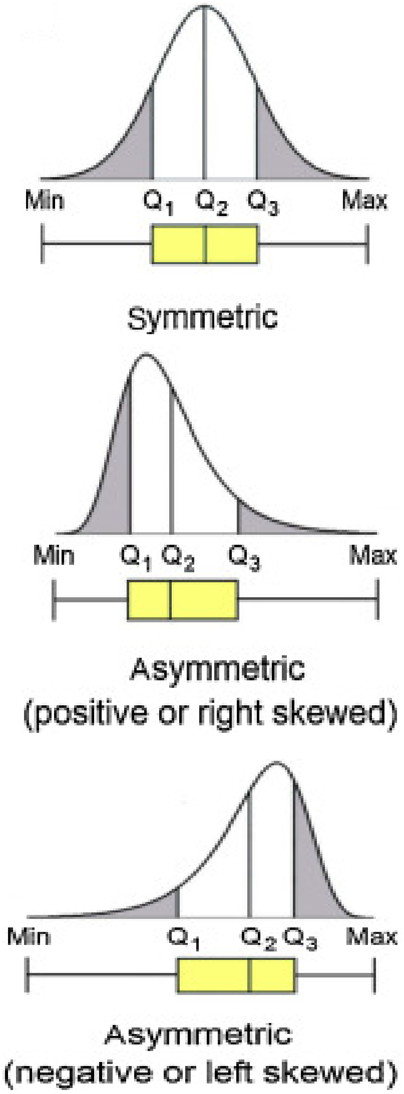


$$Q_1 = 48.5$$
$$Q_2 = 54$$
$$Q_3 = 66.5$$
$$IQR = 18$$

Box Plot



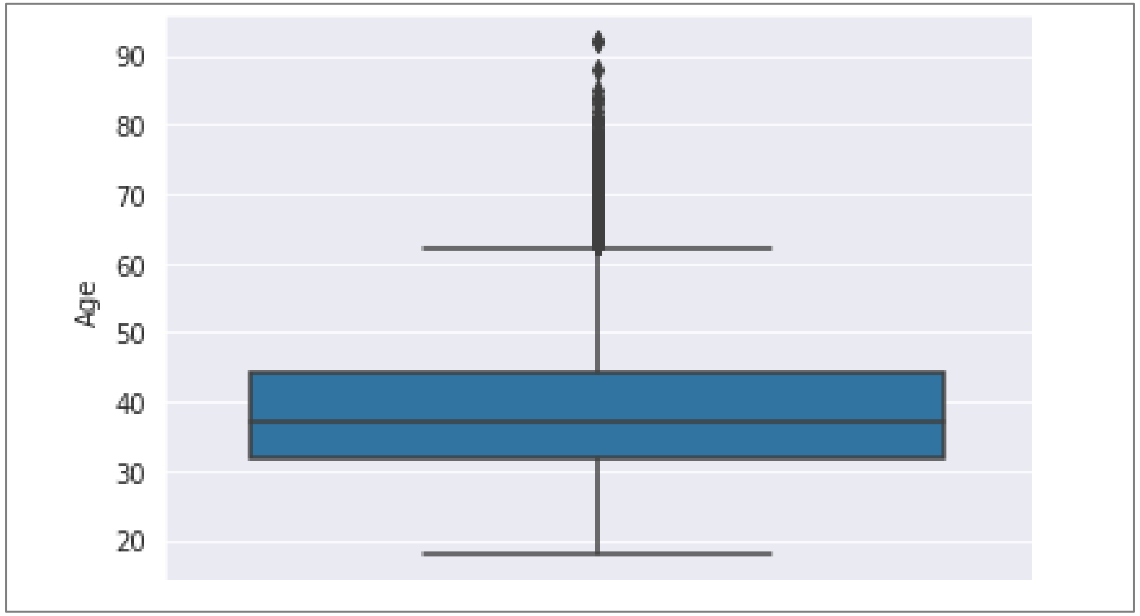
Normal Distribution
(Quartile 3 - Quartile 2) = (Quartile 2 - Quartile 1)
Positive Skew
(Quartile 3 - Quartile 2) > (Quartile 2 - Quartile 1)
Negative Skew
(Quartile 3 - Quartile 2) < (Quartile 2 - Quartile 1)



Min
Q1
Q2
Q3
Max
Symmetric
Asymmetric
(positive or right skewed)
Asymmetric
(negative or left skewed)

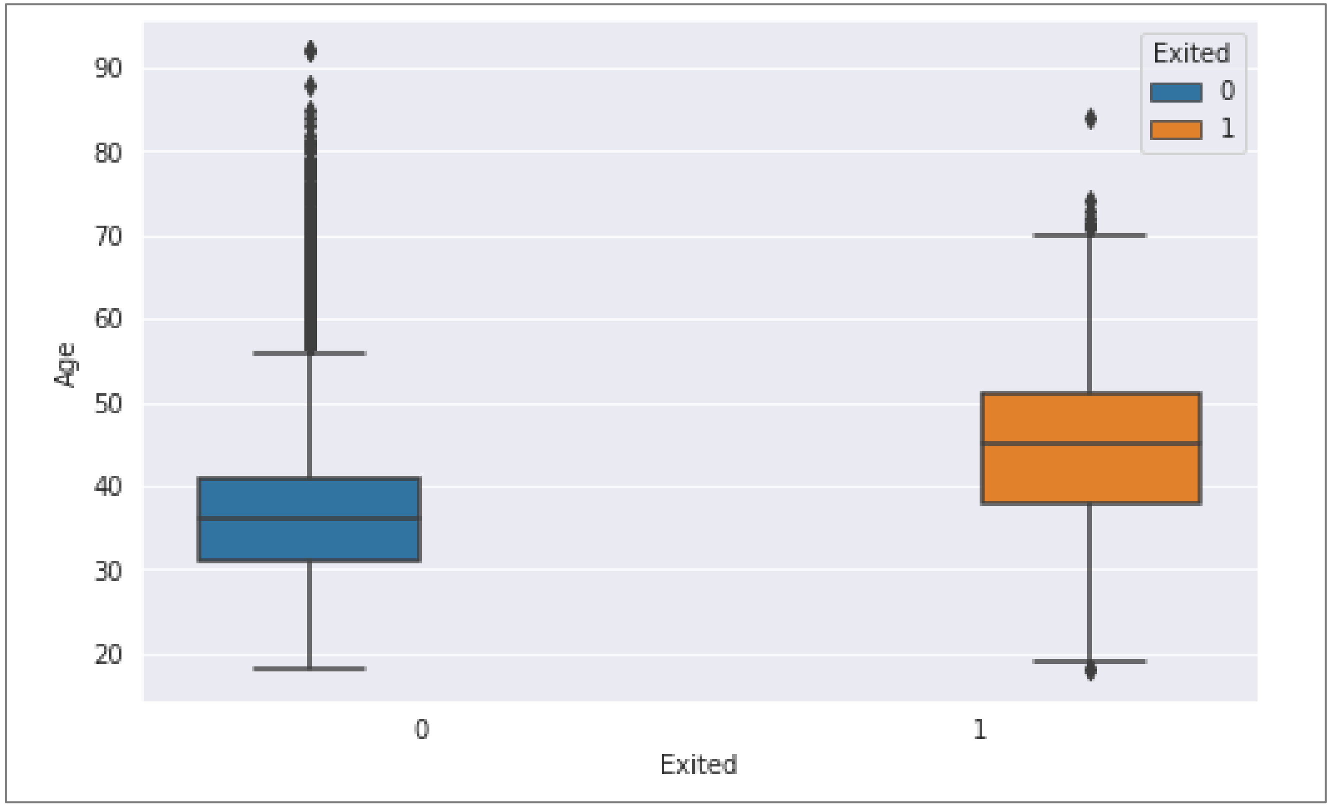
Box Plot

```
sns.boxplot(y='Age',data = df)
```



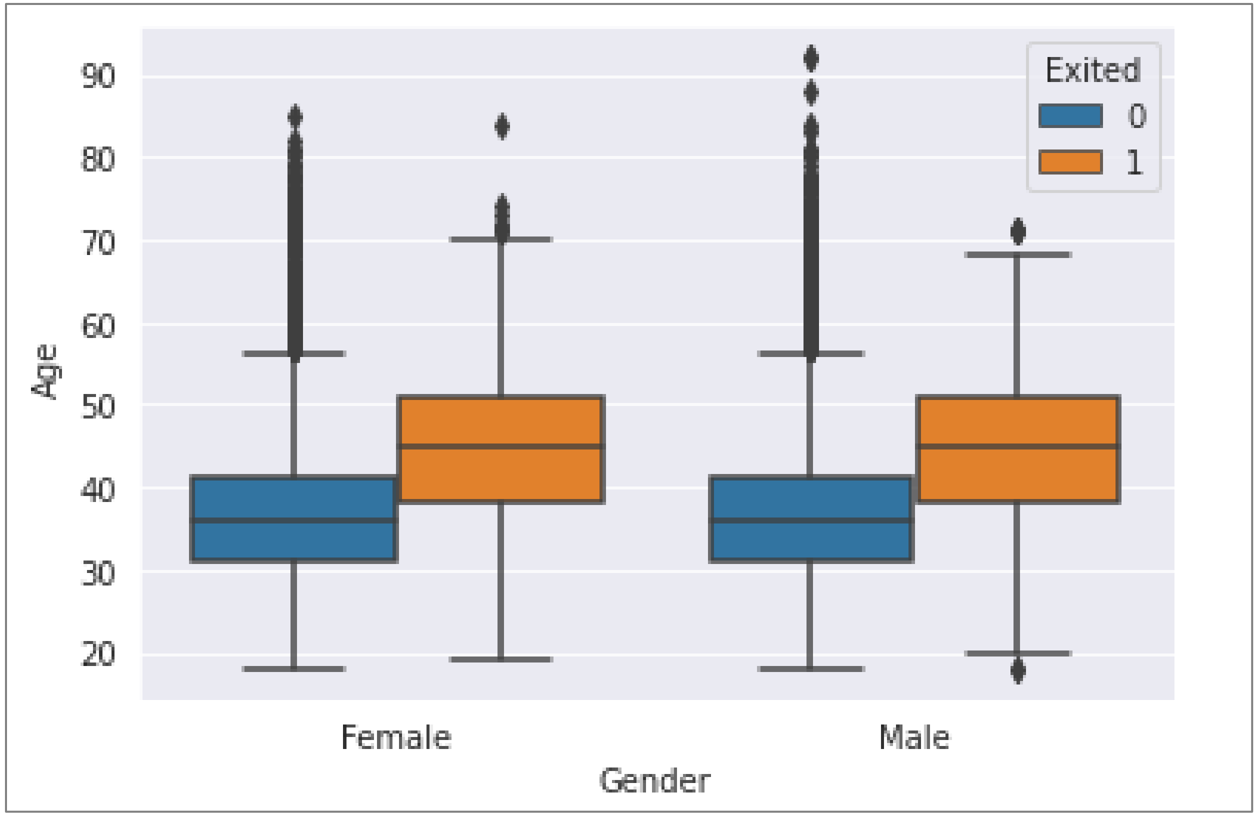
Box Plot

```
fig = plt.figure(figsize = (8, 5))
sns.boxplot(y='Age',x = 'Exited', hue = 'Exited',data = df)
```



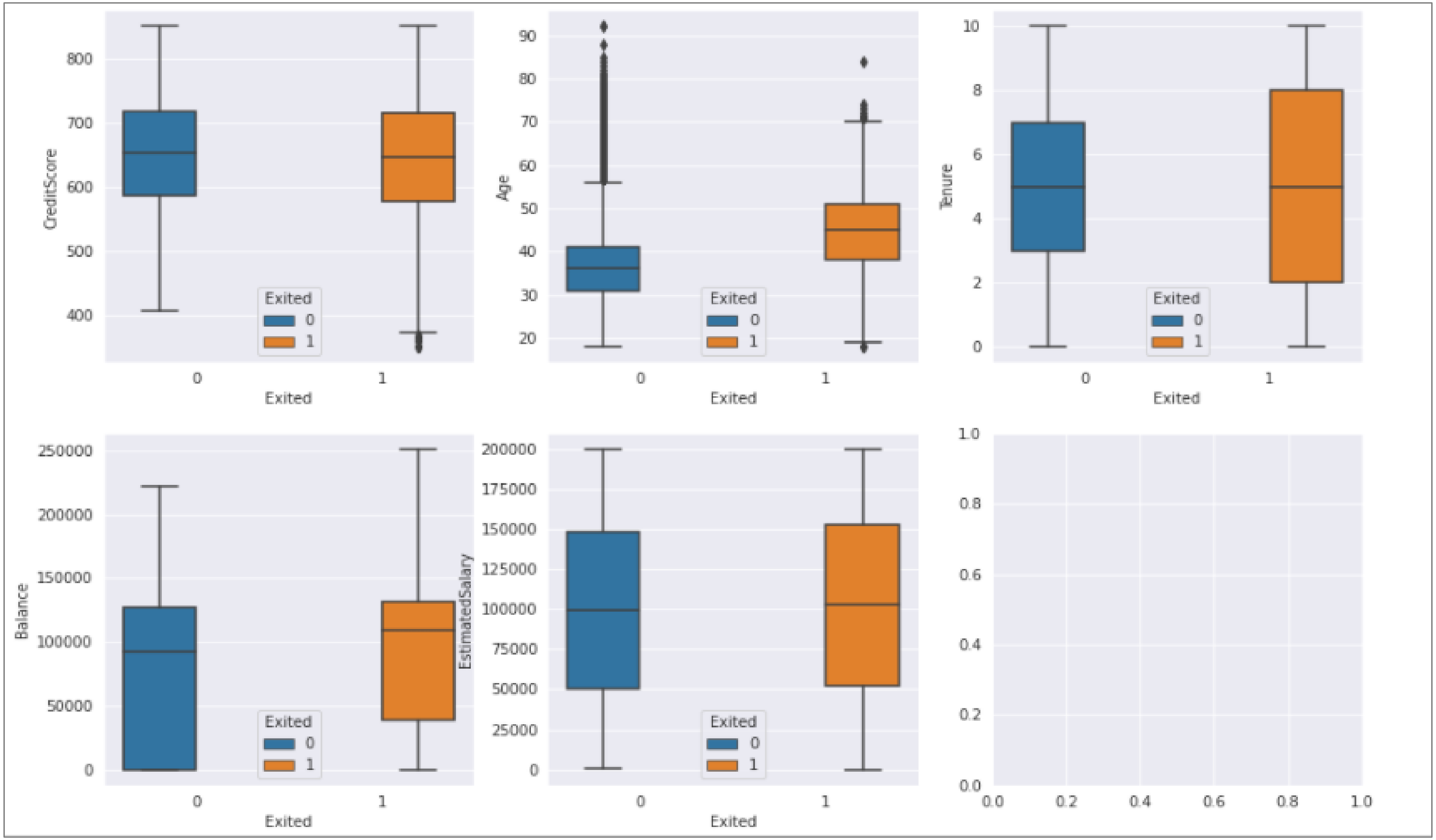
Box Plot

```
sns.boxplot(y='Age',x = 'Gender', hue = 'Exited', data = df)
```



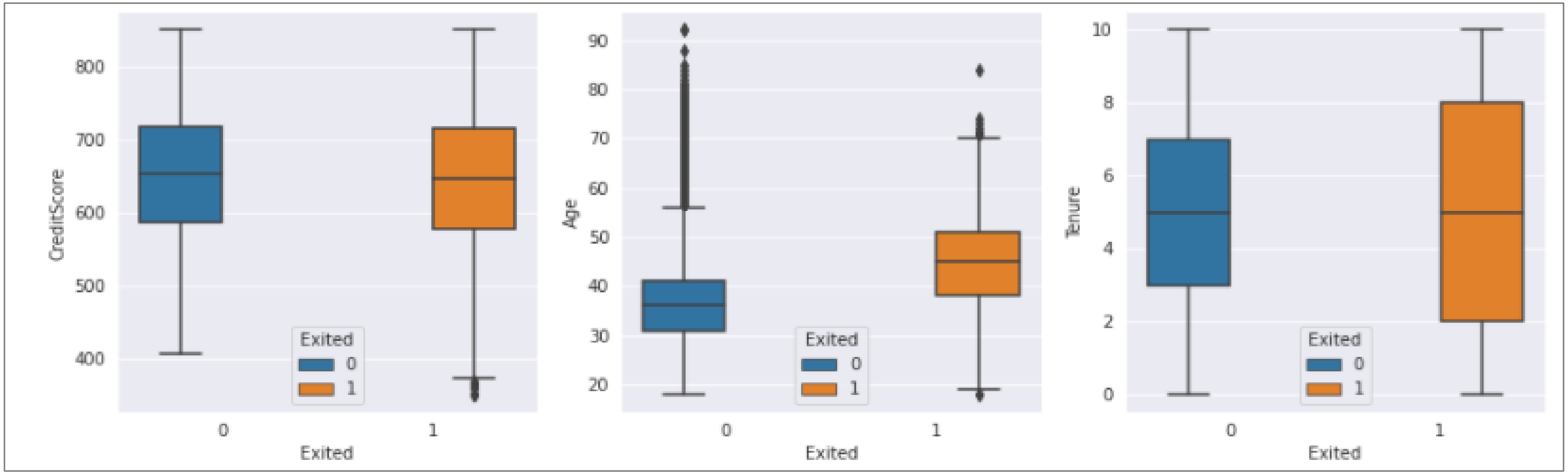
Box Plot

```
fig, axarr = plt.subplots( 2, 3, figsize=(15, 10))
sns.boxplot(y='CreditScore',x = 'Exited', hue = 'Exited',data = df, ax=axarr[0][0])
sns.boxplot(y='Age',x = 'Exited', hue = 'Exited',data = df , ax=axarr[0][1])
sns.boxplot(y='Tenure',x = 'Exited', hue = 'Exited',data = df, ax=axarr[0][2])
sns.boxplot(y='Balance',x = 'Exited', hue = 'Exited',data = df, ax=axarr[1][0])
sns.boxplot(y='EstimatedSalary',x = 'Exited', hue = 'Exited',data = df, ax=axarr[1][1])
```


CreditScore
800
700
600
500
400
Exited
0
1
0
1
Exited
Age
90
80
70
60
50
40
30
20
Exited
0
1
0
1
Exited
Tenure
10
8
6
4
2
0
Exited
0
1
0
1
Exited
Balance
250000
200000
150000
100000
50000
0
Exited
0
1
0
1
Exited
EstimatedSalary
200000
175000
150000
125000
100000
75000
50000
25000
0
Exited
0
1
0
1
Exited
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

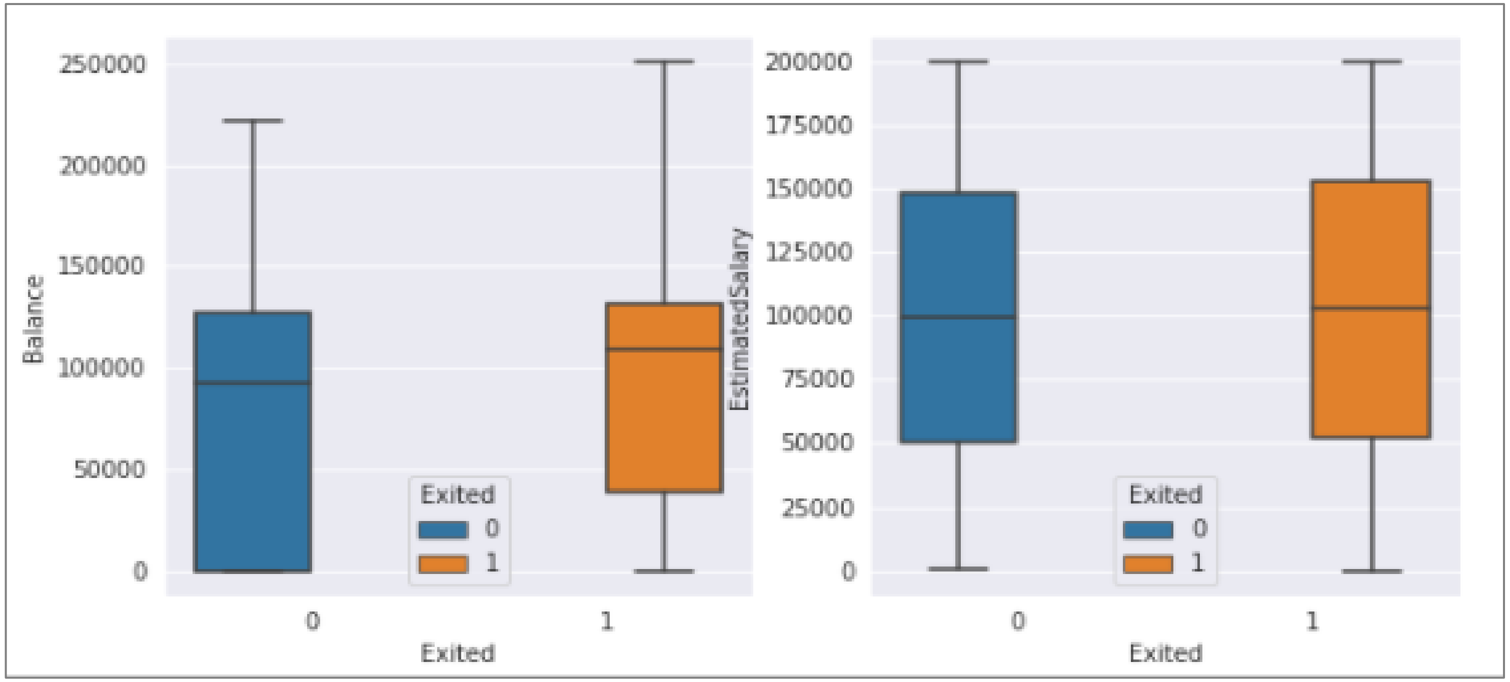
Box Plot

- There is no significant difference in the credit score distribution between retained and churned customers.
- The older customers are churning at more than the younger ones.
- The customers on either extreme end (spent little time with the bank or a lot of time with the bank) are more likely to churn compared to those that are of average tenure.



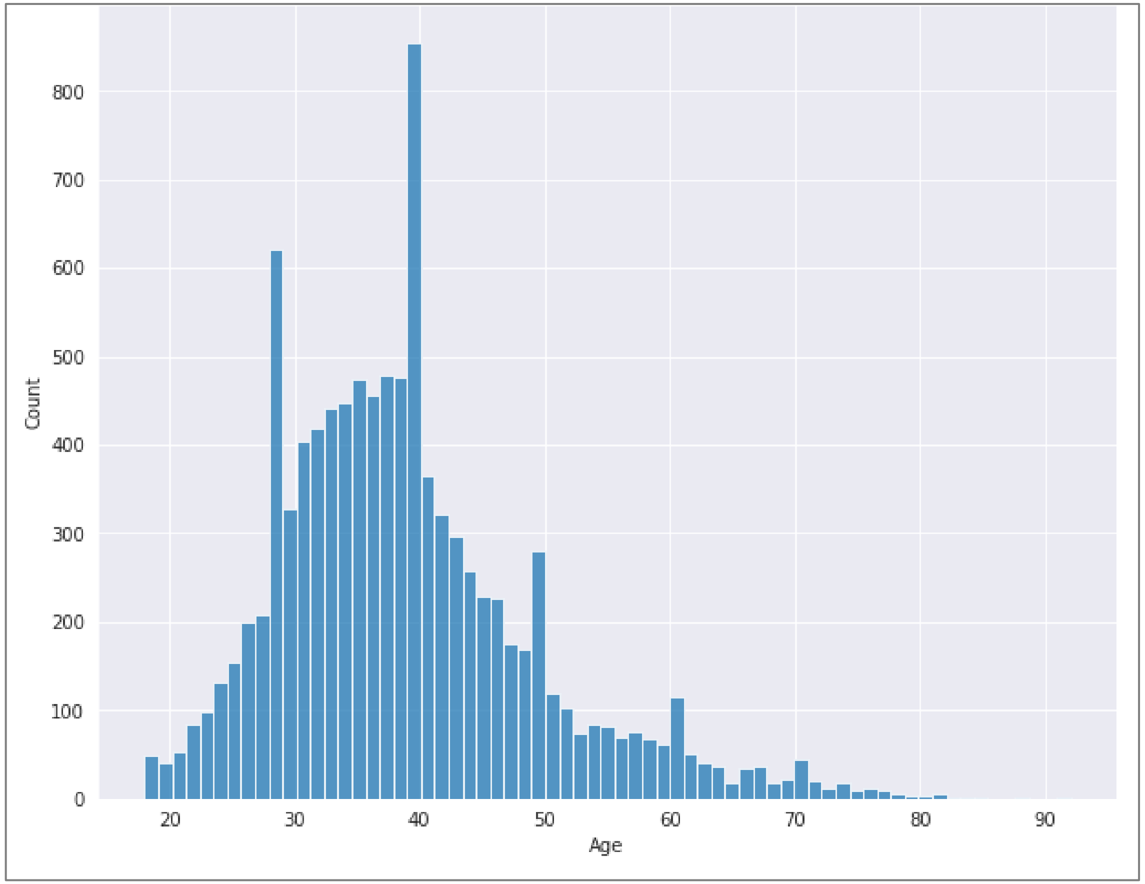
Box Plot

- The bank is losing customers with significant bank balances.
- The salary has no significant effect on the likelihood to churn.



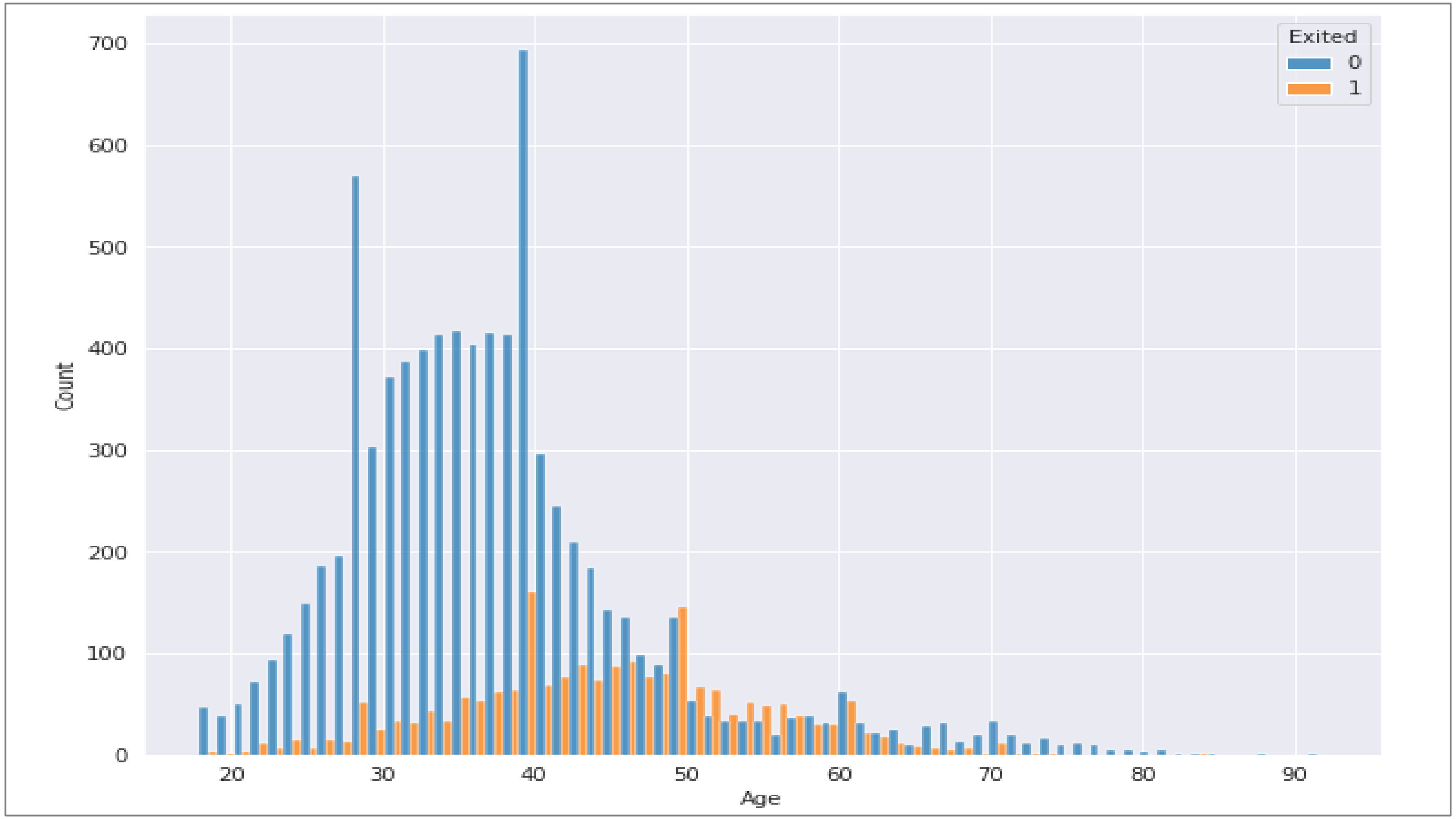
Histogram

```
fig = plt.figure(figsize = (10,8))
sns.histplot(df, x="Age")
```



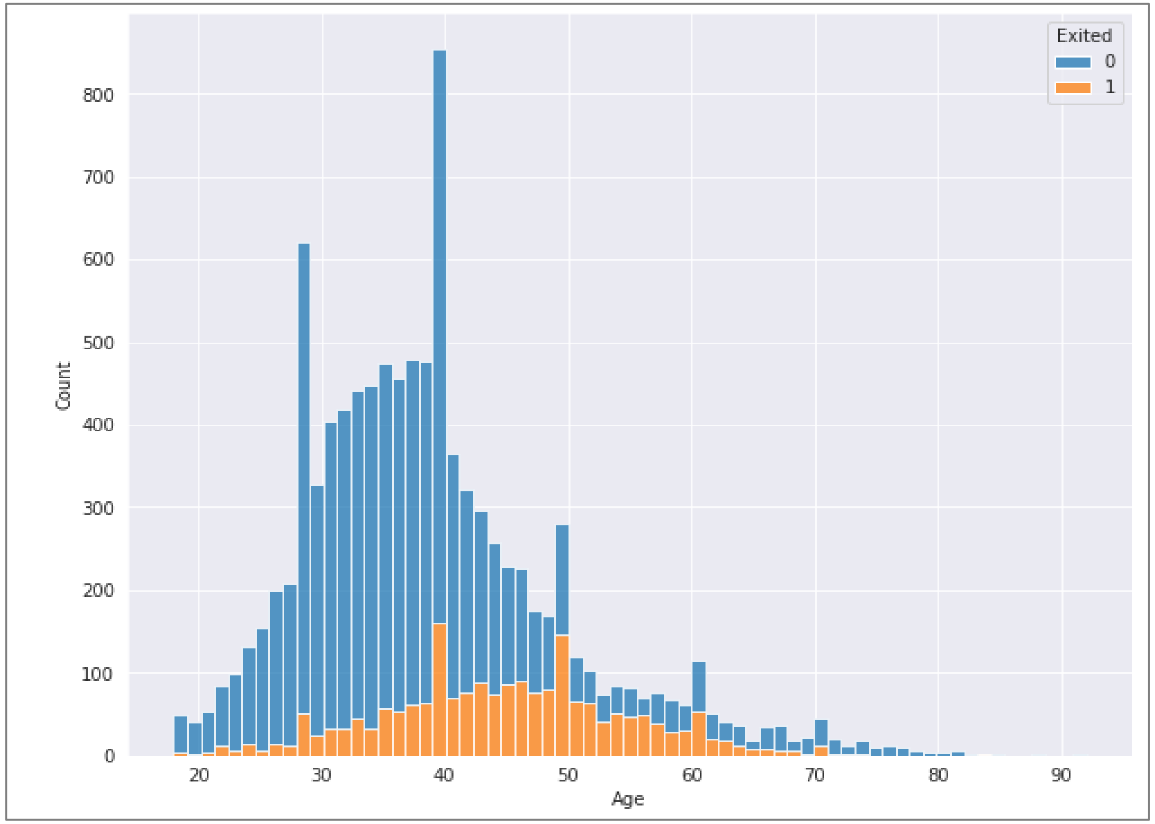
Histogram

```
fig = plt.figure(figsize = (10, 8))
sns.histplot(df, x="Age", hue = 'Exited',multiple="dodge")
```

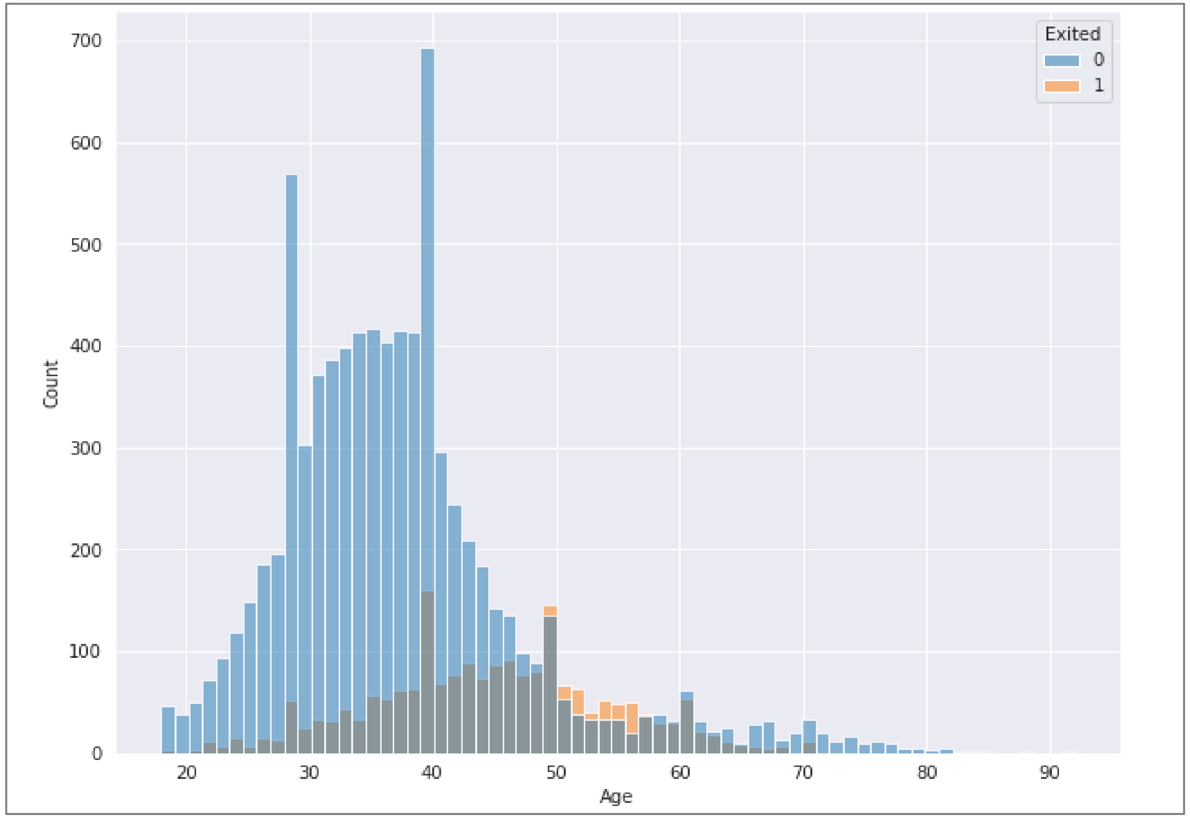


Histogram

`multiple="stack"`

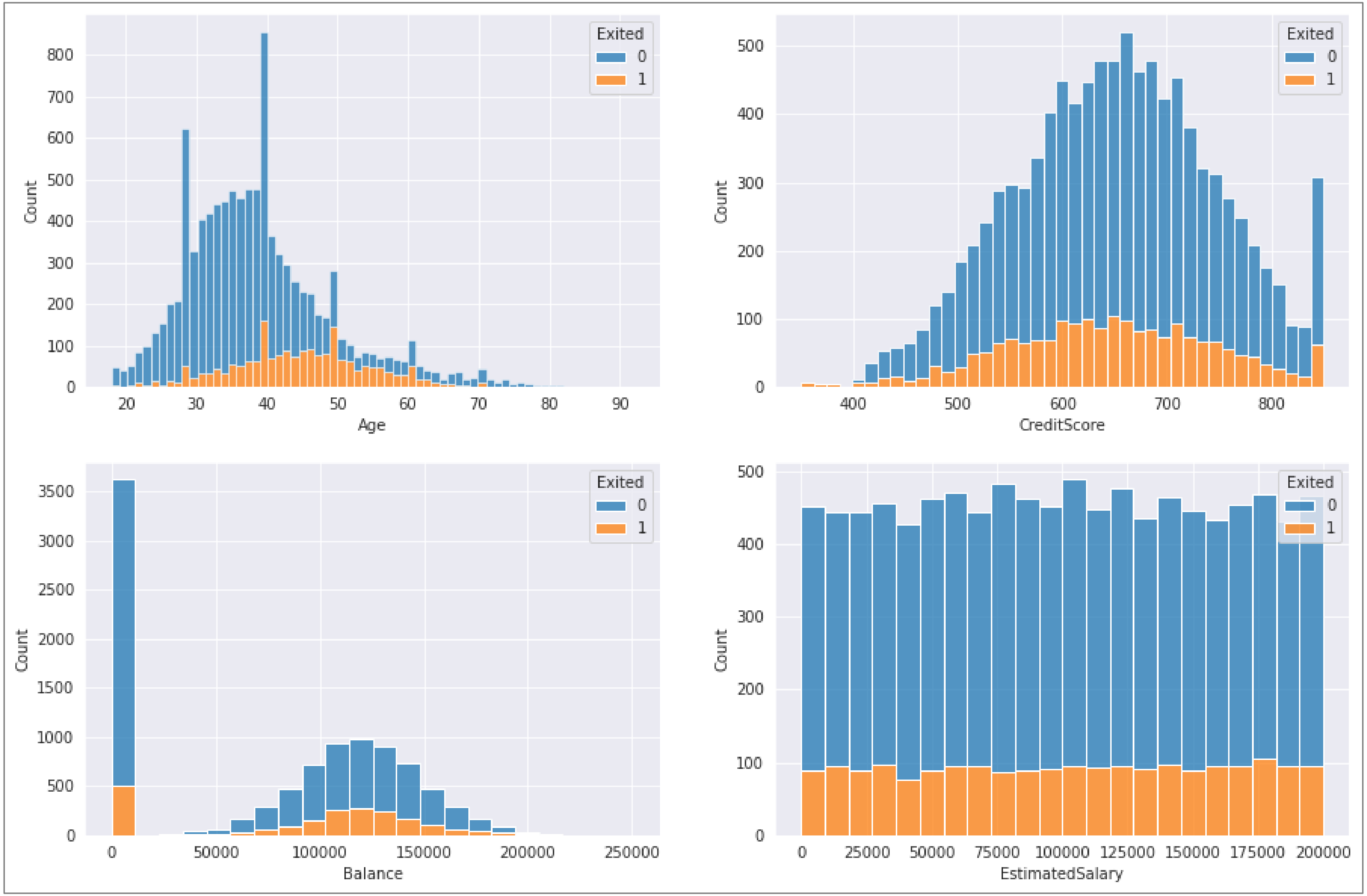


`multiple="layer"`



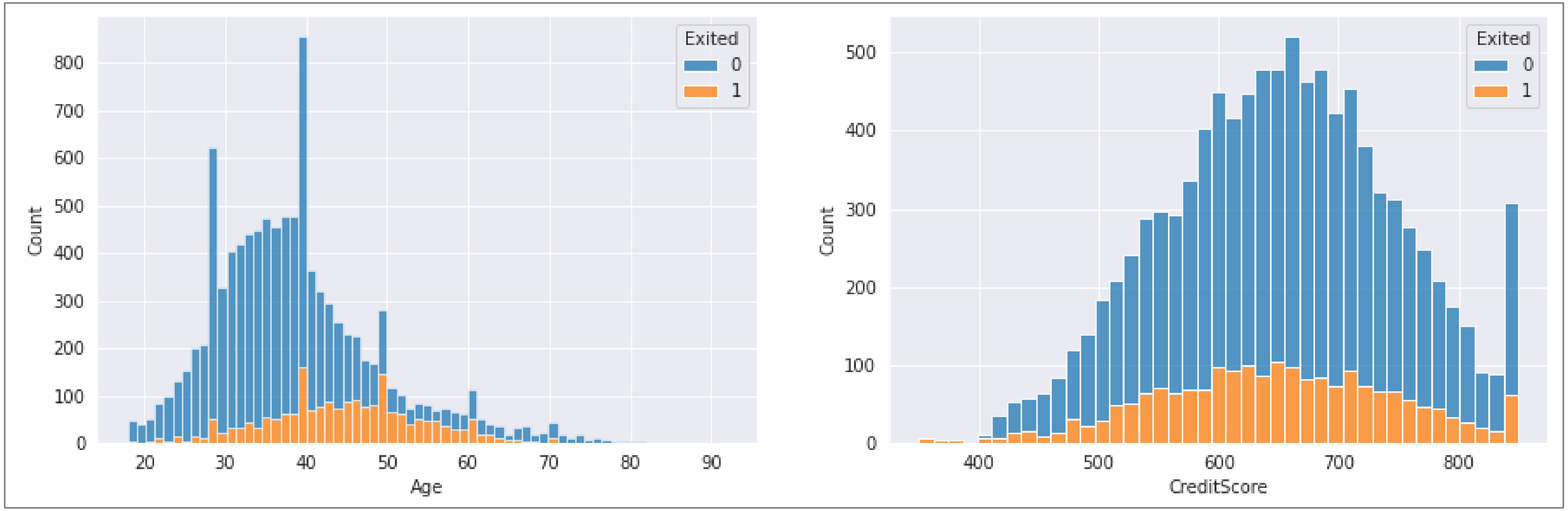
Histogram

```
fig, axarr = plt.subplots( 2, 2, figsize=(15, 10))
sns.histplot(df, x="Age", hue = 'Exited',multiple="stack", ax=axarr[0][0])
sns.histplot(df, x="CreditScore", hue = 'Exited',multiple="stack", ax=axarr[0][1])
sns.histplot(df, x="Balance", hue = 'Exited',multiple="stack", ax=axarr[1][0])
sns.histplot(df, x="EstimatedSalary", hue = 'Exited',multiple="stack", ax=axarr[1][1])
```


Exited
0
1
Count
Age
CreditScore
Balance
EstimatedSalary

Histogram

- Most of our customers are between the age of 28 to 40.
- Credit Score seems like left skewed.



Histogram

- Balances of the customers are seemed to be symmetrically distributed.
- There is not much variation in Estimated salary.