Bank Customer Churn Analysis

Domain

Finanace

Aim

To analyze bank customer data in order to identify patterns and trends that contribute to customer attrition and to offer practical advice that can assist the bank in lowering attrition.

Introduction

In today's competitive banking industry, retaining customers is as important as acquiring new ones. Many customers close their accounts or switch to other banks due to factors such as low engagement, unsatisfactory services, or better offers from competitors. This project focuses on analyzing a bank's customer dataset to identify patterns and characteristics of customers who are likely to churn. By examining features such as account balance, tenure, product usage, and activity levels, the project aims to provide data-driven insights that can help the bank proactively address churn and improve customer loyalty.

Dataset Description

Source: Kaggle(Bank Customer Chrun)

Size:10,000 Rows and 18 Columns

Features:

1. **RowNumber** - corresponds to the record (row) number and has no effect on the output.
2. **CustomerId** - contains random values and has no effect on customer leaving the bank.
3. **Surname** - the surname of a customer has no impact on their decision to leave the bank.
4. **CreditScore** - can have an effect on customer churn, since a customer with a higher credit score is less likely to leave the bank.
5. **Geography** - a customer's location can affect their decision to leave the bank.

6. **Gender** - it's interesting to explore whether gender plays a role in a customer leaving the bank.
7. **Age** - this is certainly relevant, since older customers are less likely to leave their bank than younger ones.
8. **Tenure** - refers to the number of years that the customer has been a client of the bank. Normally, older clients are more loyal and less likely to leave a bank.
9. **Balance** - also a very good indicator of customer churn, as people with a higher balance in their accounts are less likely to leave the bank compared to those with lower balances.
10. **NumOfProducts** - refers to the number of products that a customer has purchased through the bank.
11. **HasCrCard** - denotes whether or not a customer has a credit card. This column is also relevant, since people with a credit card are less likely to leave the bank.
12. **IsActiveMember** - active customers are less likely to leave the bank.
13. **EstimatedSalary** - as with balance, people with lower salaries are more likely to leave the bank compared to those with higher salaries.
14. **Exited** - whether or not the customer left the bank.
15. **Complain** - customer has complaint or not.
16. **Satisfaction Score** - score provided by the customer for their complaint resolution.
17. **Card Type** - type of card held by the customer.
18. **Points Earned** - the points earned by the customer for using a credit card.

Objective

- To analyze bank customer data to understand customer behavior.
- To identify factors that lead to customer churn.
- To predict which customers are likely to leave the bank.
- To provide insights that help improve customer retention.

Coding and Implementation

Data Loading and Initial Overview

Import Libraries

In [59]:
```
import pandas as pd
import numpy as np
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Load Dataset

In [60]:
```
df = pd.read_csv(r"D:\Data Analytics\Main Project\Old\Customer-Churn-Records.csv
```

Dataset Overview

In [61]:
```
print("\n🔷 Shape of the dataset :",df.shape,"\n" )
print("🌟" * 50)
print("\n🔷 Number of Rows :",df.shape[0],"\n")
print("🌟" * 50)
print("\n🔷 Number of Columns :",df.shape[1],"\n")
print("🌟" * 50)
print("\n🔷 Data Types :",df.dtypes,"\n")
print("🌟" * 50)
print("\n🔷 Dataset Info :",df.info(),"\n")
print("🌟" * 50)
print("\n🔷 First 5 Rows :",df.head(),"\n")
print("🌟" * 50)
print("\n🔷 Choose a sample row :",df.sample(),"\n")
print("🌟" * 50)
print("\n🔷 Statistical Summary :",df.describe(),"\n")
```

```
🔷 Shape of the dataset : (10117, 18)

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Number of Rows : 10117

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Number of Columns : 18

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Data Types : RowNumber              int64
CustomerId             int64
Surname               object
CreditScore           object
Geography             object
Gender                object
Age                   object
Tenure                object
Balance               object
NumOfProducts         object
HasCrCard             object
IsActiveMember        object
EstimatedSalary       object
Exited                 int64
Complain              object
Satisfaction Score    object
Card Type             object
Point Earned          object
dtype: object

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10117 entries, 0 to 10116
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   RowNumber           10117 non-null  int64 
 1   CustomerId          10117 non-null  int64 
 2   Surname             10097 non-null  object
 3   CreditScore         10104 non-null  object
 4   Geography           10099 non-null  object
 5   Gender              10089 non-null  object
 6   Age                 10107 non-null  object
 7   Tenure              10111 non-null  object
 8   Balance             10103 non-null  object
 9   NumOfProducts       10095 non-null  object
 10  HasCrCard           10095 non-null  object
 11  IsActiveMember      10089 non-null  object
 12  EstimatedSalary     10096 non-null  object
 13  Exited              10117 non-null  int64 
 14  Complain            10094 non-null  object
 15  Satisfaction Score  10098 non-null  object
 16  Card Type           10103 non-null  object
 17  Point Earned        10099 non-null  object
```

```
dtypes: int64(3), object(15)
memory usage: 1.4+ MB

🔷 Dataset Info : None

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 First 5 Rows :    RowNumber  CustomerId   Surname CreditScore Geography  Gend
er Age Tenure  \
0          1    15634602  Hargrave         619    France  Female  42      2
1          2    15647311      Hill         608     Spain  Female  41      1
2          3    15619304      Onio         502    France  Female  42      8
3          4    15701354      Boni         699    France  Female  39      1
4          5    15737888  Mitchell         850     Spain  Female  43      2

     Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary  Exited  \
0          0             1         1              1       101348.88       1
1   83807.86             1         0              1       112542.58       0
2   159660.8             3         1              0       113931.57       1
3          0             2         0              0        93826.63       0
4  125510.82             1         1              1         79084.1       0

  Complain Satisfaction Score Card Type Point Earned
0        1                  2   DIAMOND          464
1        1                  3   DIAMOND          456
2        1                  3   DIAMOND          377
3        0                  5      GOLD          350
4        0                  5      GOLD          425

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Choose a sample row :       RowNumber  CustomerId Surname CreditScore Geograp
hy Gender Age Tenure  \
4121       4122    15606133     Lay         628     Spain    Male  42      7

     Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary  Exited  \
4121       0             2         0              1       172967.87       0

     Complain Satisfaction Score Card Type Point Earned
4121        0                  2    SILVER          283

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Statistical Summary :           RowNumber    CustomerId        Exited
count  10117.000000  1.011700e+04  10117.000000
mean    5059.000000  1.569108e+07      0.206385
std     2920.670671  7.246364e+04      0.404730
min        1.000000  1.550275e+07      0.000000
25%     2530.000000  1.562832e+07      0.000000
50%     5059.000000  1.569094e+07      0.000000
75%     7588.000000  1.575372e+07      0.000000
max    10117.000000  1.589275e+07      1.000000
```

Data Pre-processing

Checking Missing Values

In [62]:
```
print("\n🔷 Missing Values :\n",df.isnull().sum(),"\n")
print("🌟" * 50)
print("\nCount of total Misisng Values :\n",df.isnull().sum().sum(),"\n")
```

```
🔷 Missing Values :
 RowNumber            0
CustomerId           0
Surname             20
CreditScore         13
Geography           18
Gender              28
Age                 10
Tenure               6
Balance             14
NumOfProducts       22
HasCrCard           22
IsActiveMember      28
EstimatedSalary     21
Exited               0
Complain            23
Satisfaction Score  19
Card Type           14
Point Earned        18
dtype: int64 

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

Count of total Misisng Values :
 276
```

Text Cleaning

In [63]:
```
# Remove extra spaces
for col in df.select_dtypes(include="object").columns:
    df[col] = df[col].str.strip()

# proper case
proper_case = ["Surname","Geography","Gender"]

for col in proper_case:
    df[col] = df[col].str.title()

# make Card Type upper case
df["Card Type"] = df["Card Type"].str.upper()
```

Correcting Data Types

In [64]:
```
df['CreditScore'] = pd.to_numeric(df['CreditScore'], errors='coerce').astype('In
df['Age'] = pd.to_numeric(df['Age'], errors='coerce').astype('Int64')
df['Tenure'] = pd.to_numeric(df['Tenure'], errors='coerce').astype('Int64')
df['Balance'] = pd.to_numeric(df['Balance'], errors='coerce').astype('float')
df['NumOfProducts'] = pd.to_numeric(df['NumOfProducts'], errors='coerce').astype
df['HasCrCard'] = pd.to_numeric(df['HasCrCard'], errors='coerce').astype('Int64'
df['IsActiveMember'] = pd.to_numeric(df['IsActiveMember'], errors='coerce').asty
```

```
df['EstimatedSalary'] = pd.to_numeric(df['EstimatedSalary'], errors='coerce').as
df['Satisfaction Score'] = pd.to_numeric(df['Satisfaction Score'], errors='coerc
df['Point Earned'] = pd.to_numeric(df['Point Earned'], errors='coerce').astype('
```

Handling Missing Values

In [65]:
```
#if the Geography is empty fill with 'Unknown'
df['Geography'] = df['Geography'].fillna('Unknown')

#fill miising age and estimatedsalary with mean and point Eraned with meadian
df['Age'] = df['Age'].fillna(round(df['Age'].mean())).astype('Int64')
df['Point Earned'] = df['Point Earned'].fillna(round(df['Point Earned'].median()
df['EstimatedSalary'] = df['EstimatedSalary'].fillna(df['EstimatedSalary'].mean(


#Remove the remaing rows with null values
df.dropna(inplace=True)
```

Remove Duplicates/Column

In [66]:
```
print("\n🔷 No.of Duplicate rows :",df['CustomerId'].duplicated().sum())
print("🌟" * 50)

#removal of Duplicate rows
df = df.drop_duplicates(subset=['CustomerId'], keep='last')


print("\n🔷 Null Values after cleaning :",df.isnull().sum()
,"\n")

#Remove unwanted columns
df.drop(['RowNumber', 'Surname'], axis=1, inplace=True)
```

```
🔷 No.of Duplicate rows : 17
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Null Values after cleaning : 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
Complain              0
Satisfaction Score    0
Card Type             0
Point Earned          0
dtype: int64
```

Creating Derived Columns

In [67]:
```
#HasBalance for checking balance status
df['HasBalance'] = df['Balance'].apply(lambda x: 1 if x > 0 else 0)

#Grouping age for easy analysyis
df['AgeGroup'] = pd.cut(df['Age'], bins=[18,30,40,50,60,100], labels=['18-30','3

#grouping tenure
df['TenureGroup'] = pd.cut(df['Tenure'], bins=[0,2,5,10], labels=['0-2yrs','3-5y
```

In [68]:
```
# Aftre pre-proceesing checking for data type
print("\n🔷 Data Types :",df.dtypes)
```

```
🔷 Data Types : CustomerId              int64
CreditScore            Int64
Geography             object
Gender                object
Age                    Int64
Tenure                 Int64
Balance              float64
NumOfProducts          Int64
HasCrCard              Int64
IsActiveMember         Int64
EstimatedSalary      float64
Exited                 int64
Complain              object
Satisfaction Score     Int64
Card Type             object
Point Earned           Int64
HasBalance             int64
AgeGroup            category
TenureGroup         category
dtype: object
```

Filtering or aggregating data

In [69]:
```
#Filtering
# Customers with Balance greater than 50,000
high_balance = df[df['Balance'] > 50000]
print("\n🔷Customers with Balance greater than 50,000  :\n",high_balance,"\n")
print("🌟" * 50)

#Active members in France
active = df[(df['Geography'] == 'France') & (df['IsActiveMember'] == 1)]
print("\n🔷Active members in France  :\n",active,"\n")
print("🌟" * 50)

#Churned customers
churned = df[df['Exited'] == 1]
print("\n🔷Churned Customers  :\n",churned,"\n")
print("🌟" * 50)

#Sum of point  earned per tenure group
points_per_tenure = df.groupby('TenureGroup',observed=False)['Point Earned'].sum
print("\n🔷Sum of point  earned per tenure group  :\n",points_per_tenure,"\n")
print("🌟" * 50)
```

```
#Average Balance by age group
avg_balance = df.groupby('Geography')['Balance'].mean()
print("\n🔷Sum of point  earned per tenure group  :\n",avg_balance,"\n")
```

```
🔷Customers with Balance greater than 50,000  :
        CustomerId  CreditScore Geography  Gender  Age  Tenure    Balance  \
1         15647311          608     Spain  Female   41       1   83807.86
2         15619304          502    France  Female   42       8  159660.80
4         15737888          850     Spain  Female   43       2  125510.82
5         15574012          645     Spain    Male   44       8  113755.78
7         15656148          376   Germany  Female   29       4  115046.74
...            ...          ...       ...     ...  ...     ...        ...
10112     15637690          622   Germany  Female   34       7   98675.74
10113     15632882          684   Germany    Male   37       1  126817.13
10114     15807107          612    France    Male   32       3  121394.42
10115     15661034          813   Germany  Female   29       5  106059.40
10116     15811958          850   Germany    Male   44       2  112755.34

       NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited  \
1                  1          0               1        112542.58       0
2                  3          1               0        113931.57       1
4                  1          1               1         79084.10       0
5                  2          1               0        149756.71       1
7                  4          1               0        119346.88       1
...              ...        ...             ...              ...     ...
10112              1          1               0        138906.85       1
10113              2          1               1         29995.83       1
10114              1          1               0        164081.42       0
10115              1          0               0        187976.88       1
10116              2          0               0        158171.36       0

      Complain  Satisfaction Score Card Type  Point Earned  HasBalance  \
1            1                   3   DIAMOND           456           1
2            1                   3   DIAMOND           377           1
4            0                   5      GOLD           425           1
5            1                   5   DIAMOND           484           1
7            1                   2   DIAMOND           282           1
...        ...                 ...       ...           ...         ...
10112        1                   5      GOLD           995           1
10113        1                   1      GOLD           511           1
10114        0                   1  PLATINUM           588           1
10115        1                   5      GOLD           331           1
10116        0                   5    SILVER           842           1

      AgeGroup TenureGroup
1        41-50      0-2yrs
2        41-50     6-10yrs
4        41-50      0-2yrs
5        41-50     6-10yrs
7        18-30      3-5yrs
...        ...         ...
10112    31-40     6-10yrs
10113    31-40      0-2yrs
10114    31-40      3-5yrs
10115    18-30      3-5yrs
10116    41-50      0-2yrs

[6308 rows x 19 columns]

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Active members in France  :
        CustomerId  CreditScore Geography  Gender  Age  Tenure    Balance  \
0       15634602         619    France  Female   42        2       0.00
6       15592531         822    France    Male   50        7       0.00
8       15792365         501    France    Male   44        4  142051.07
9       15592389         684    France    Male   27        2  134603.88
19      15568982         726    France  Female   24        6       0.00
...          ...         ...       ...     ...  ...      ...        ...
9997    15584532         709    France  Female   36        7       0.00
10101   15603035         651    France    Male   34        3       0.00
10106   15751912         567    France    Male   36        7       0.00
10110   15812422         637    France    Male   41        2       0.00
10111   15806941         499    France    Male   60        7   76961.60

       NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited  \
0                  1          1               1        101348.88       1
6                  2          1               1         10062.80       0
8                  2          0               1         74940.50       0
9                  1          1               1         71725.73       0
19                 2          1               1         54724.03       0
...              ...        ...             ...              ...     ...
9997               1          0               1         42085.58       1
10101              2          1               1        105599.65       0
10106              2          0               1          3896.08       0
10110              2          0               1        102515.42       0
10111              2          1               1         83643.87       0

      Complain  Satisfaction Score Card Type  Point Earned  HasBalance  \
0            1                  2   DIAMOND           464           0
6            0                  2    SILVER           206           0
8            0                  3      GOLD           251           1
9            0                  3      GOLD           342           1
19           0                  4      GOLD           477           0
...        ...                ...       ...           ...         ...
9997         1                  3    SILVER           564           0
10101        0                  5    SILVER           632           0
10106        0                  5      GOLD           914           0
10110        0                  4      GOLD           325           0
10111        0                  1  PLATINUM           514           1

      AgeGroup TenureGroup
0        41-50      0-2yrs
6        41-50     6-10yrs
8        41-50      3-5yrs
9        18-30      0-2yrs
19       18-30     6-10yrs
...        ...         ...
9997     31-40     6-10yrs
10101    31-40      3-5yrs
10106    31-40     6-10yrs
10110    41-50      0-2yrs
10111    51-60     6-10yrs

[2591 rows x 19 columns]

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷 Churned Customers   :
        CustomerId  CreditScore Geography  Gender  Age  Tenure    Balance  \
0       15634602         619    France  Female   42        2       0.00
2       15619304         502    France  Female   42        8  159660.80
```

```
5         15574012          645     Spain    Male    44        8  113755.78
7         15656148          376   Germany  Female    29        4  115046.74
16        15737452          653   Germany    Male    58        1  132602.88
...            ...          ...       ...     ...   ...      ...        ...
10107     15677336          557   Germany    Male    57        1  120043.13
10108     15684395          446     Spain  Female    45       10  125191.69
10112     15637690          622   Germany  Female    34        7   98675.74
10113     15632882          684   Germany    Male    37        1  126817.13
10115     15661034          813   Germany  Female    29        5  106059.40

       NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited  \
0                  1          1               1        101348.88       1
2                  3          1               0        113931.57       1
5                  2          1               0        149756.71       1
7                  4          1               0        119346.88       1
16                 1          1               0          5097.67       1
...              ...        ...             ...              ...     ...
10107              1          1               0        132370.75       1
10108              1          1               1        128260.86       1
10112              1          1               0        138906.85       1
10113              2          1               1         29995.83       1
10115              1          0               0        187976.88       1

      Complain  Satisfaction Score Card Type  Point Earned  HasBalance  \
0            1                   2   DIAMOND           464           0
2            1                   3   DIAMOND           377           1
5            1                   5   DIAMOND           484           1
7            1                   2   DIAMOND           282           1
16           0                   2    SILVER           163           1
...        ...                 ...       ...           ...         ...
10107        1                   1      GOLD           954           1
10108        1                   5    SILVER           584           1
10112        1                   5      GOLD           995           1
10113        1                   1      GOLD           511           1
10115        1                   5      GOLD           331           1

      AgeGroup TenureGroup
0        41-50      0-2yrs
2        41-50     6-10yrs
5        41-50     6-10yrs
7        18-30      3-5yrs
16       51-60      0-2yrs
...        ...         ...
10107    51-60      0-2yrs
10108    41-50     6-10yrs
10112    31-40     6-10yrs
10113    31-40      0-2yrs
10115    18-30      3-5yrs

[2038 rows x 19 columns]

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Sum of point  earned per tenure group  :
 TenureGroup
0-2yrs     1270014
3-5yrs     1831144
6-10yrs    2713783
Name: Point Earned, dtype: Int64
```

```
☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️
☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️☀️

🔷Sum of point  earned per tenure group  :
 Geography
France      62092.636516
Germany    119730.116134
Spain       61818.147763
Name: Balance, dtype: float64
```

In [70]:

```python
#Aggregating Data

#Average balance by geography
avg_balance = df.groupby('Geography')['Balance'].mean()
print("\n🔷Average balance by geography  :\n",avg_balance,"\n")
print("☀️" * 50)

#Count of churned customers by age group
churn_by_age = df[df['Exited'] == 1].groupby('AgeGroup',observed=True)['Exited']
print("\n🔷Count of churned customers by age group  :\n",churn_by_age,"\n")
print("☀️" * 50)

#Sum of point earned per tenure group
points_per_tenure = df.groupby('TenureGroup',observed=True)['Point Earned'].sum(
print("\n🔷Sum of point earned per tenure group :\n",points_per_tenure,"\n")
print("☀️" * 50)

#Total Balance of Active memebrs in Spain
avg_salary_churn = df[df['Exited']==1]['EstimatedSalary'].mean()
print("\n🔷Total Balance of Active memebrs in Spain  :\n",avg_salary_churn,"\n"
🔷Average balance by geography  :
 Geography
France      62092.636516
Germany    119730.116134
Spain       61818.147763
Name: Balance, dtype: float64

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Count of churned customers by age group  :
 AgeGroup
18-30    146
31-40    539
41-50    788
51-60    448
60+      115
Name: Exited, dtype: int64

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Sum of point earned per tenure group :
 TenureGroup
0-2yrs     1270014
3-5yrs     1831144
6-10yrs    2713783
Name: Point Earned, dtype: Int64

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Total Balance of Active memebrs in Spain  :
 101509.90878312069
```

Outlier Handling

In [71]:
```
#Before handling outliers

numeric_cols = ["CreditScore", "Age", "Balance", "EstimatedSalary", "Point Earne
before_summary = []

for col in numeric_cols:
    before_summary.append([col, df[col].min(), df[col].max(), df[col].count()])
before_df = pd.DataFrame(before_summary, columns=['Column', 'Min', 'Max', 'Count
print("\n🔷Before Handling Outliers\n")
print(before_df)
```

```
🔷Before Handling Outliers

            Column    Min        Max  Count
0      CreditScore  350.00     850.00  10000
1              Age   18.00      92.00  10000
2          Balance    0.00  250898.09  10000
3  EstimatedSalary   11.58  199992.48  10000
4     Point Earned  119.00    1000.00  10000
```

In [72]:
```
#Handling Outlier
```

```
df_cleaned = df.copy()
for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    df_cleaned = df_cleaned[(df_cleaned[col] >= lower) & (df_cleaned[col] <= upp
```

In [73]:
```
#After Handling Summary
after_summary = []
for col in numeric_cols:
    after_summary.append([col, df_cleaned[col].min(), df_cleaned[col].max(), df_
after_df = pd.DataFrame(after_summary, columns=["Column", "Min", "Max", "Count"]

print("\n🔷 AFTER OUTLIER HANDLING \n")
print(after_df)
```

```
🔷 AFTER OUTLIER HANDLING

            Column    Min        Max  Count
0      CreditScore  383.00     850.00   9626
1              Age   18.00      62.00   9626
2          Balance    0.00  250898.09   9626
3  EstimatedSalary   11.58  199992.48   9626
4     Point Earned  119.00    1000.00   9626
```

In [74]:
```
#df.to_csv("c.csv",index=False)
```

Exploratory Data Analysis(EDA)

Unique Value Counts

In [75]:
```
print("\n🔷 Unique Value in Each Columns :\n")

for col in df.columns:
    print(col,":",df[col].nunique(),"Unique Values\n")
```

```
🔷 Unique Value in Each Columns :

CustomerId : 10000 Unique Values

CreditScore : 460 Unique Values

Geography : 3 Unique Values

Gender : 2 Unique Values

Age : 70 Unique Values

Tenure : 11 Unique Values

Balance : 6382 Unique Values

NumOfProducts : 4 Unique Values

HasCrCard : 2 Unique Values

IsActiveMember : 2 Unique Values

EstimatedSalary : 9999 Unique Values

Exited : 2 Unique Values

Complain : 2 Unique Values

Satisfaction Score : 5 Unique Values

Card Type : 4 Unique Values

Point Earned : 785 Unique Values

HasBalance : 2 Unique Values

AgeGroup : 5 Unique Values

TenureGroup : 3 Unique Values
```

Descrptive Statistics

In [76]:
```
#Statistical summary of Numerical columns
df.describe().T
```

Out[76]:

	count	mean	std	min	25%	!
CustomerId	10000.0	15690940.5694	71936.186123	15565701.0	15628528.25	156907
CreditScore	10000.0	650.5288	96.653299	350.0	584.0	6
Age	10000.0	38.9218	10.487806	18.0	32.0	
Tenure	10000.0	5.0128	2.892174	0.0	3.0	
Balance	10000.0	76485.889288	62397.405202	0.0	0.0	9719
NumOfProducts	10000.0	1.5302	0.581654	1.0	1.0	
HasCrCard	10000.0	0.7055	0.45584	0.0	0.0	
IsActiveMember	10000.0	0.5151	0.499797	0.0	0.0	
EstimatedSalary	10000.0	100090.239881	57510.492818	11.58	51002.11	100193
Exited	10000.0	0.2038	0.402842	0.0	0.0	
Satisfaction Score	10000.0	3.0138	1.405919	1.0	2.0	
Point Earned	10000.0	606.5151	225.924839	119.0	410.0	6
HasBalance	10000.0	0.6383	0.480517	0.0	0.0	

In [77]:
```
#Statistical summary of all columns

df.describe(include='object').T
```

Out[77]:

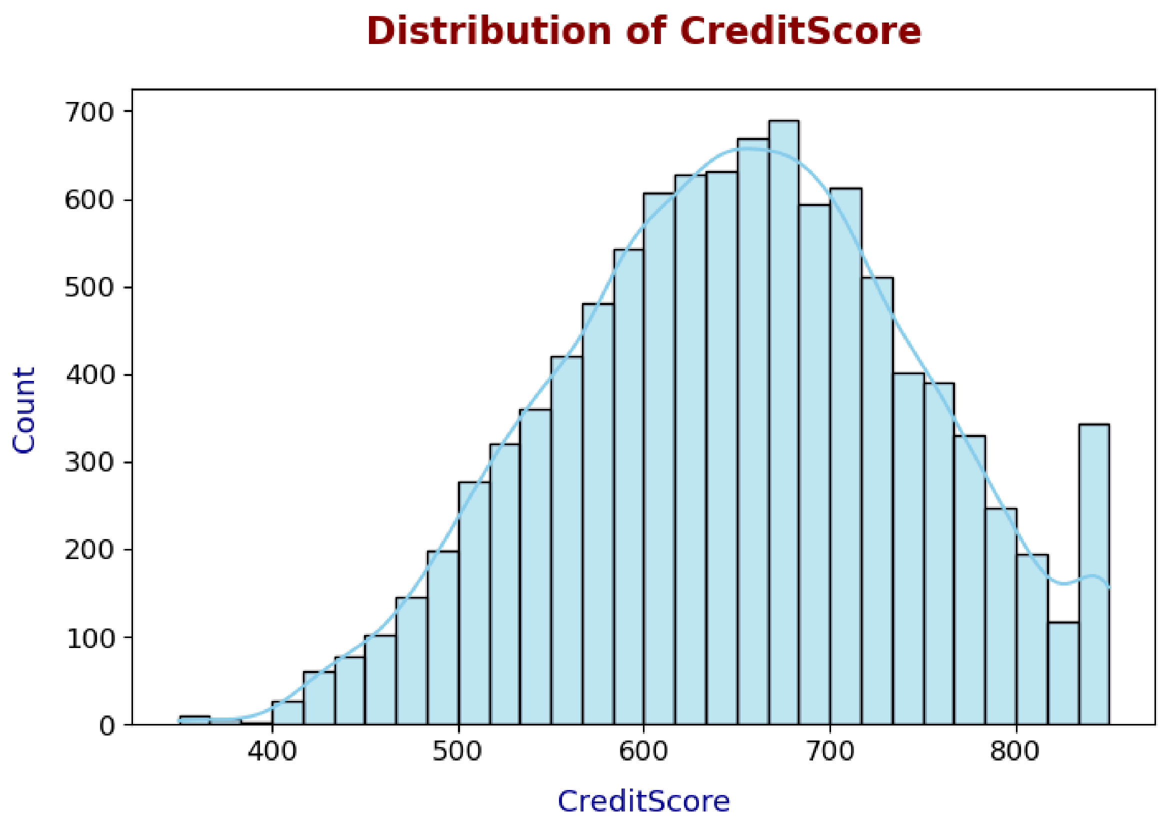
	count	unique	top	freq
Geography	10000	3	France	5014
Gender	10000	2	Male	5457
Complain	10000	2	0	7956
Card Type	10000	4	DIAMOND	2507

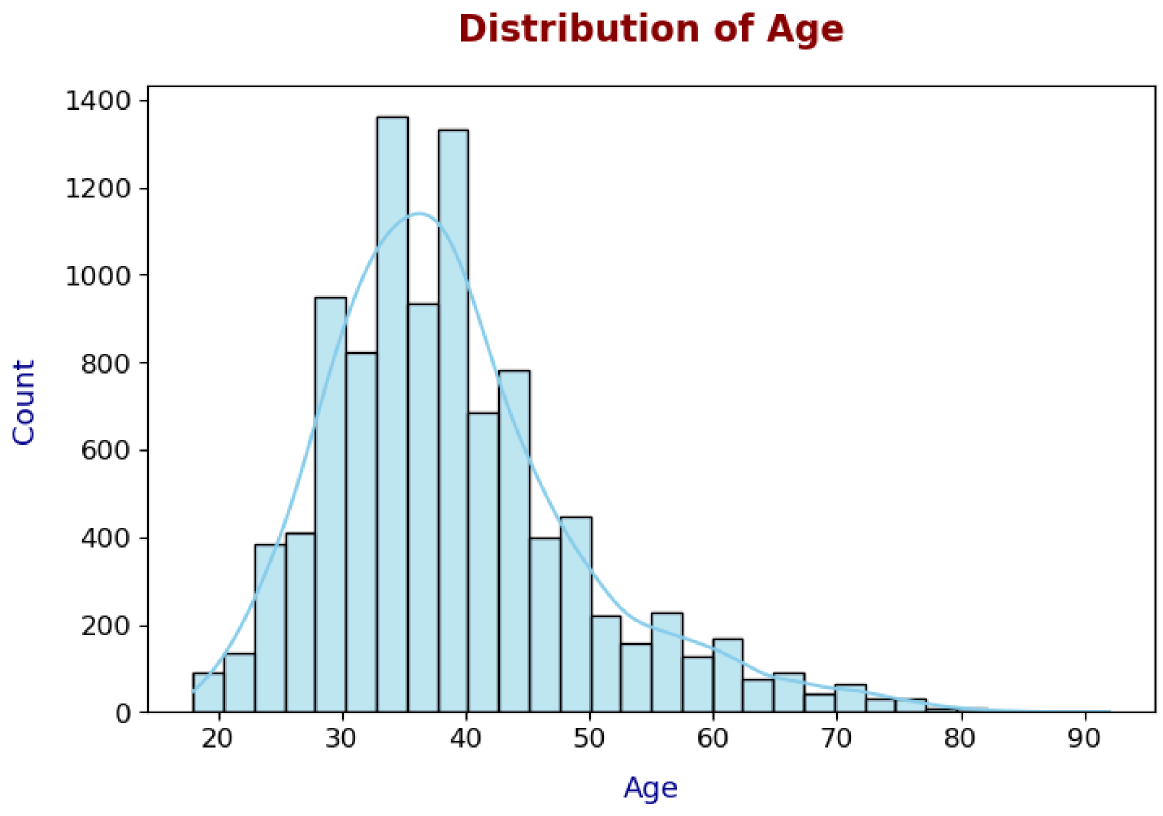
Univariate Analysis

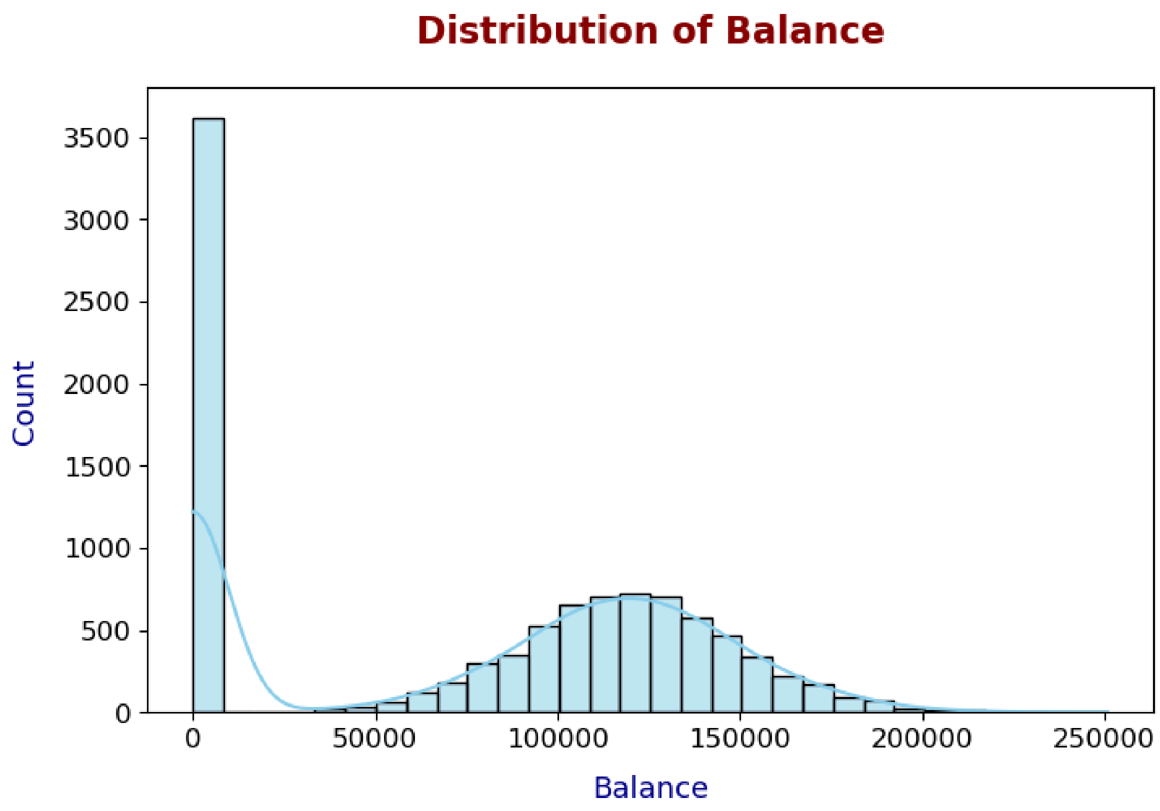
In [78]:
```
print("\n🔷",df['Exited'].value_counts())
print("\n🔷",df['IsActiveMember'].value_counts())
print("\n🔷",df['HasCrCard'].value_counts())

#Age into groups
df['AgeGroup'] = pd.cut(df['Age'], bins=[18,25,35,45,55,65,100], labels=['18-25'
print("\n🔷",df['AgeGroup'].value_counts())
```

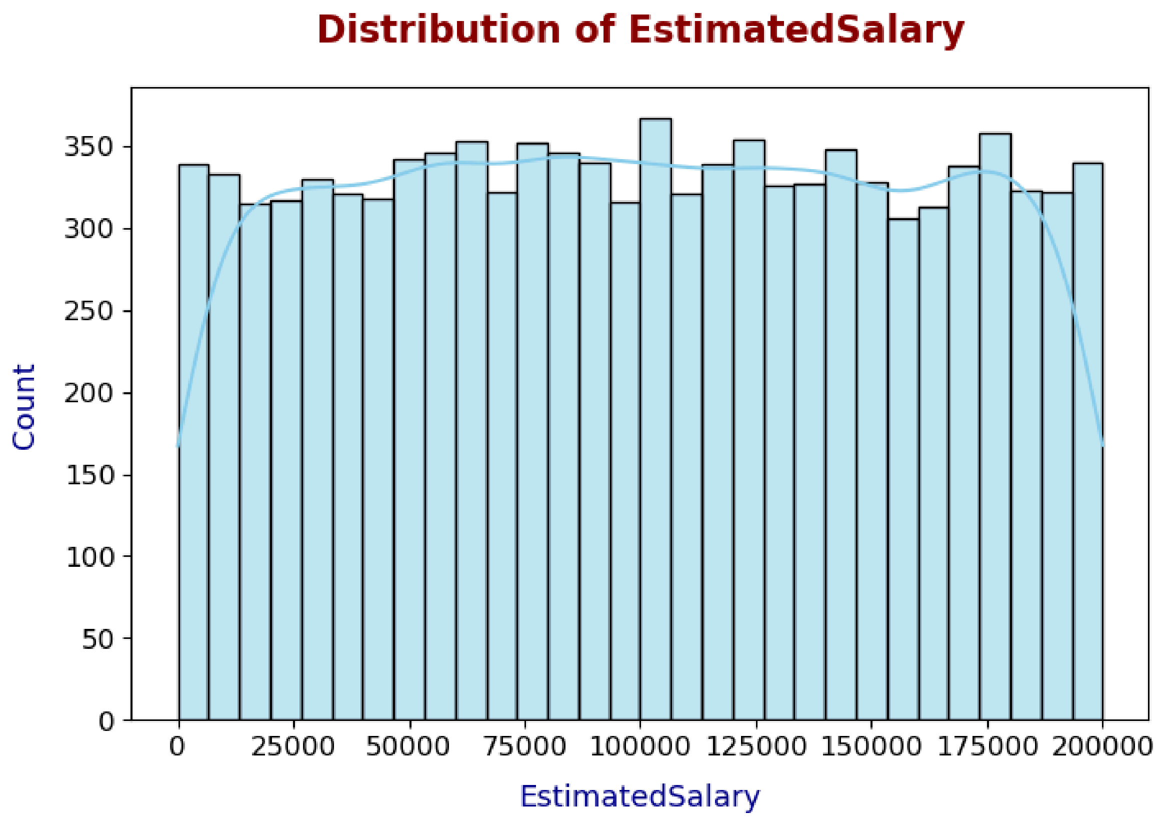
```
🔷 Exited
0    7962
1    2038
Name: count, dtype: int64

🔷 IsActiveMember
1    5151
0    4849
Name: count, dtype: Int64

🔷 HasCrCard
1    7055
0    2945
Name: count, dtype: Int64

🔷 AgeGroup
36-45    3736
26-35    3542
46-55    1311
18-25     589
56-65     536
65+       264
Name: count, dtype: int64
```

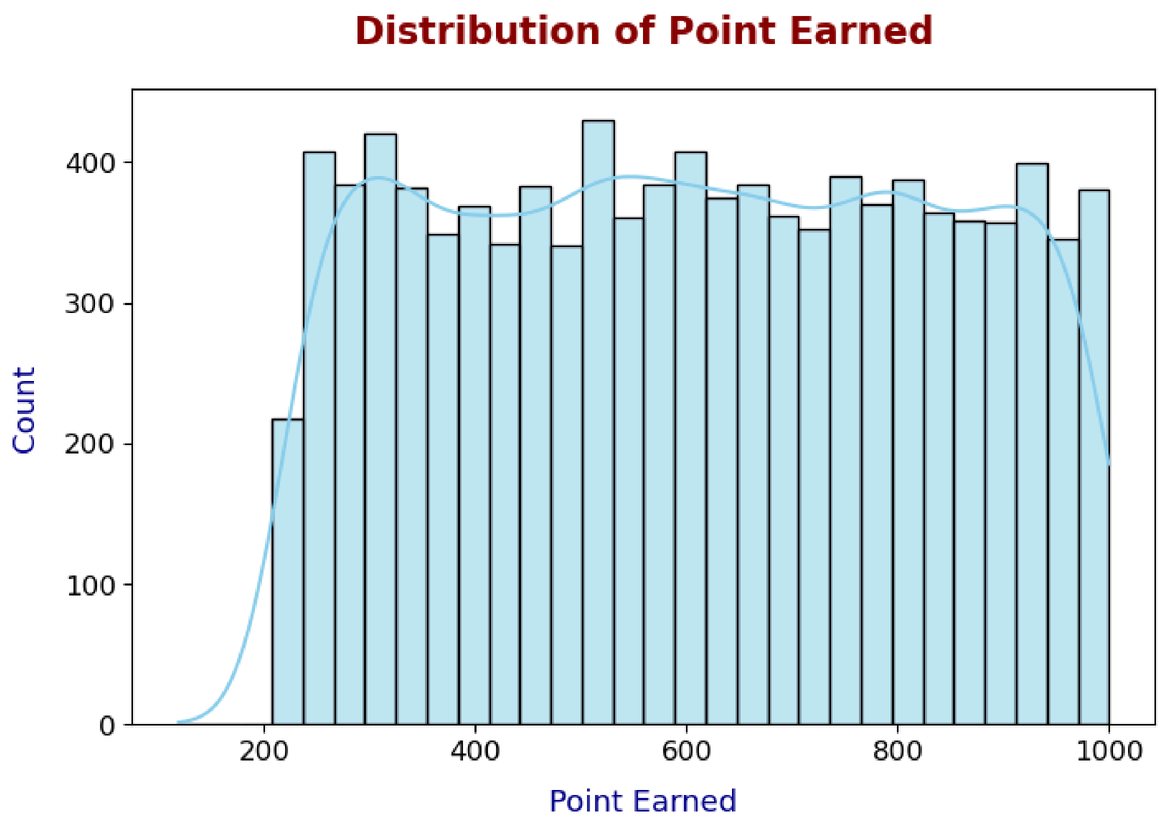
In [79]:
```
# Numeric Variables
title_style = {'fontsize':16, 'fontweight':'bold', 'color':'darkred'}
label_style = {'fontsize':13, 'color':'darkblue'}

numeric_cols = ["CreditScore", "Age", "Balance", "EstimatedSalary", "Point Earne

for col in numeric_cols:
    plt.figure(figsize=(8,5))
    sns.histplot(df[col], bins=30, kde=True, color="skyblue")
    plt.title("Distribution of " + col, **title_style, pad=20)
    plt.xlabel(col, **label_style, labelpad=10)
    plt.ylabel("Count", **label_style, labelpad=10)
    plt.xticks(fontsize=12)
    plt.yticks(fontsize=12)
    plt.show()
```


Distribution of CreditScore
Count
CreditScore
0
100
200
300
400
500
600
700
400
500
600
700
800

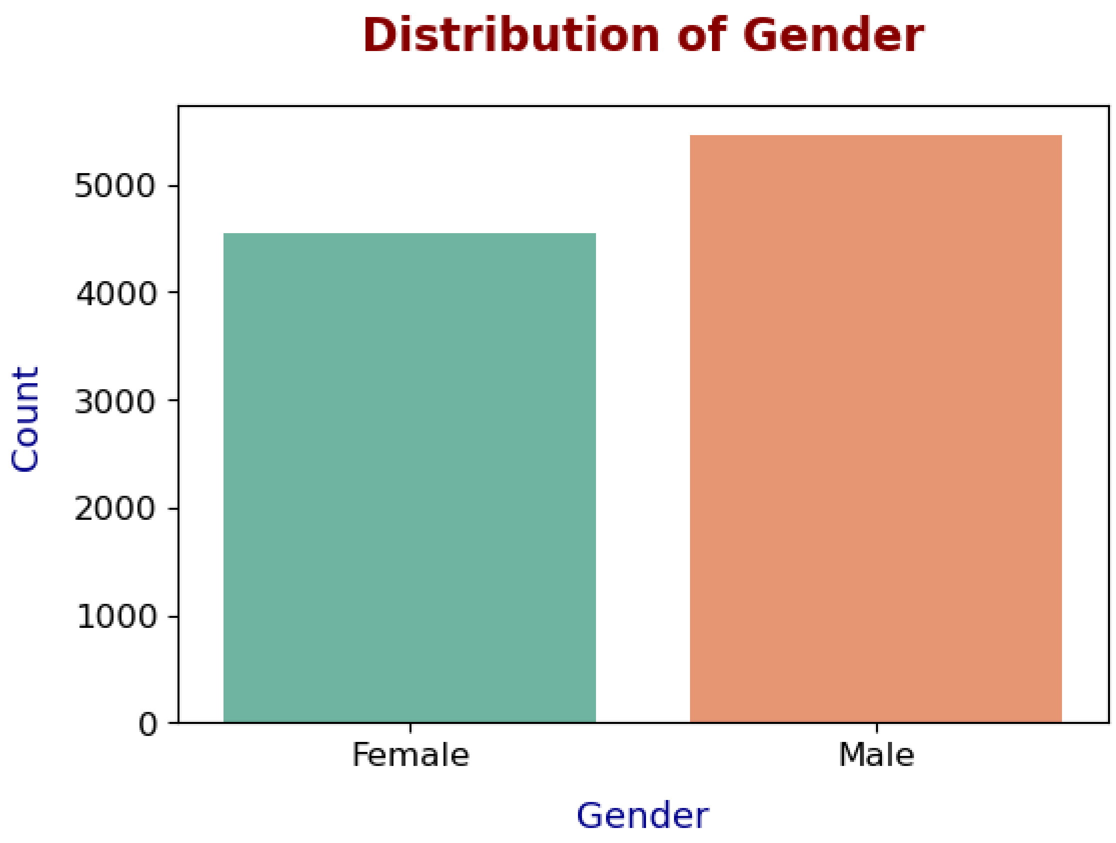

Distribution of Age
Count
Age
0
200
400
600
800
1000
1200
1400
20
30
40
50
60
70
80
90

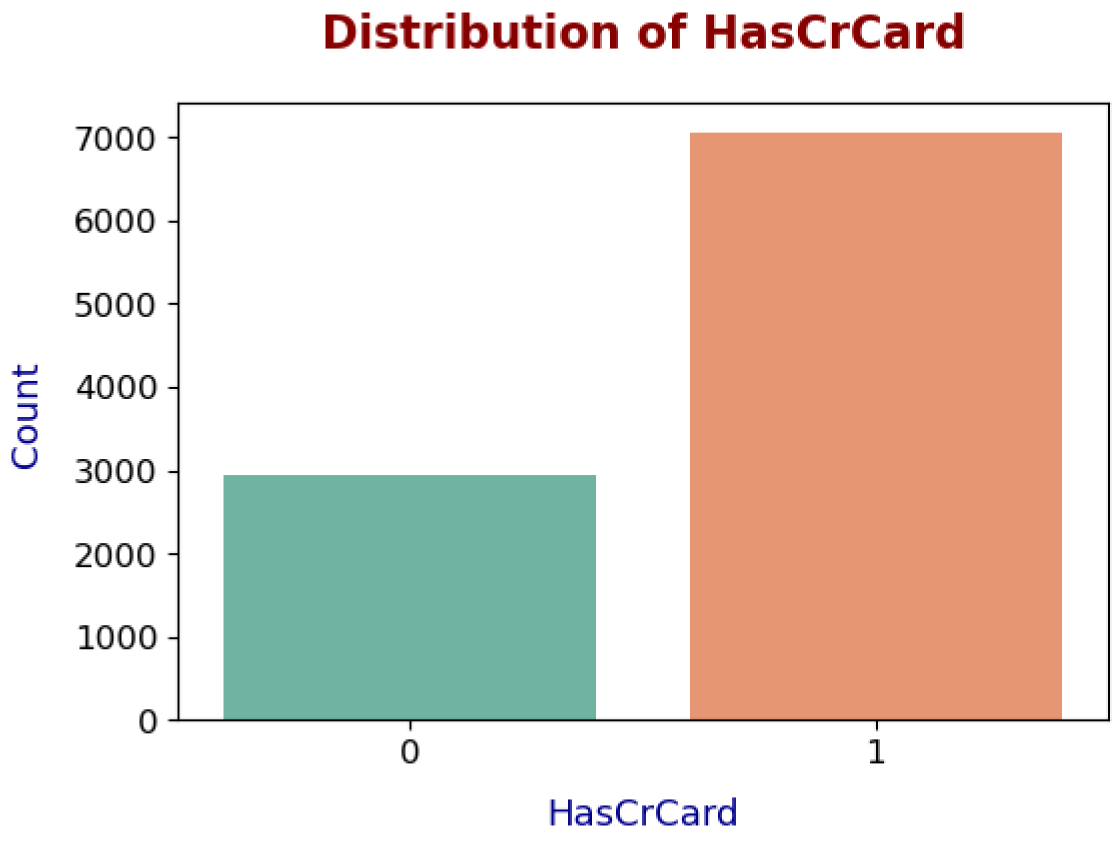

Distribution of Balance
Count
3500
3000
2500
2000
1500
1000
500
0
0
50000
100000
150000
200000
250000
Balance

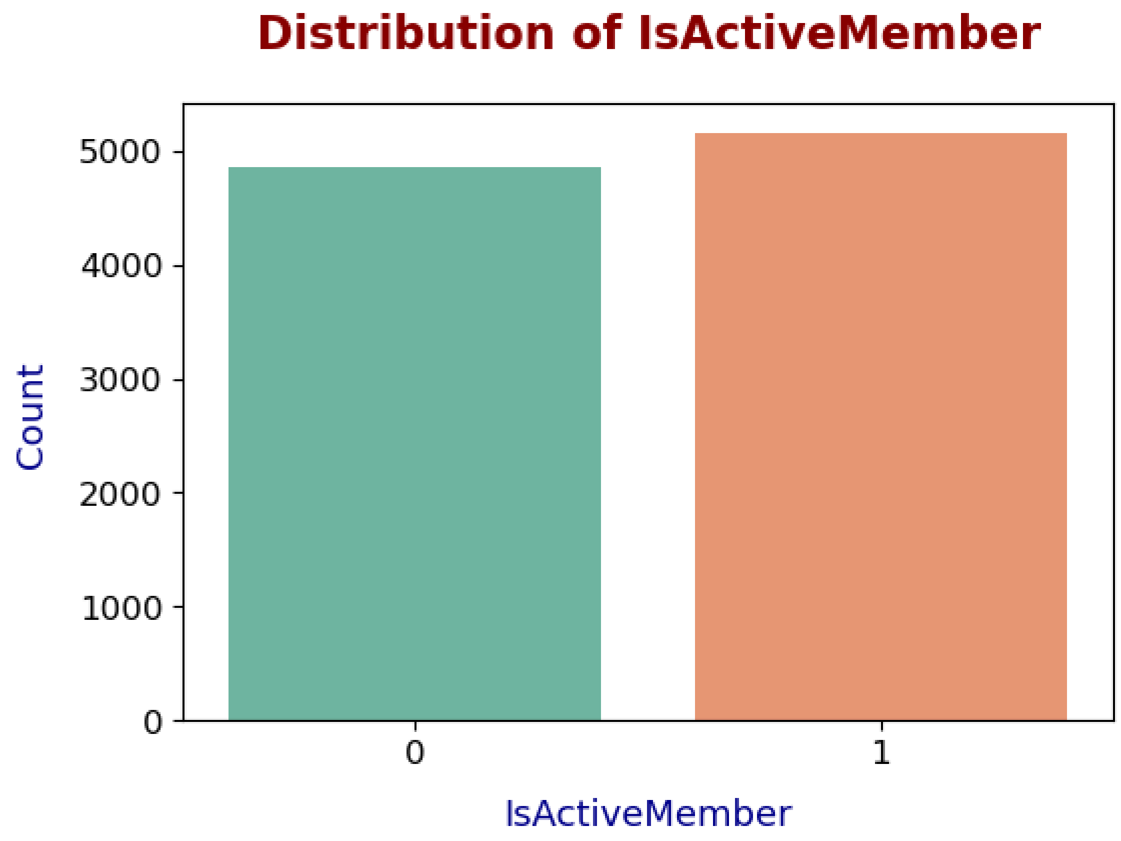

Distribution of EstimatedSalary
Count
350
300
250
200
150
100
50
0
0
25000
50000
75000
100000
125000
150000
175000
200000
EstimatedSalary

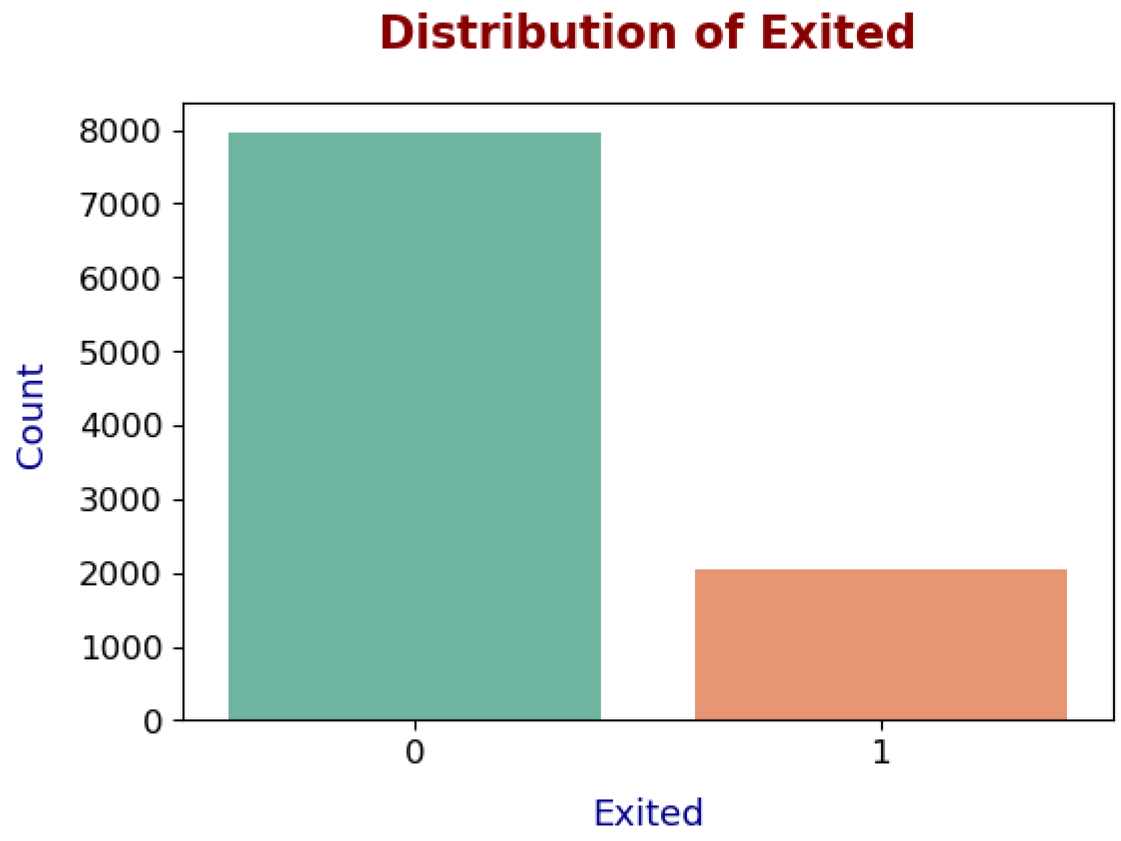


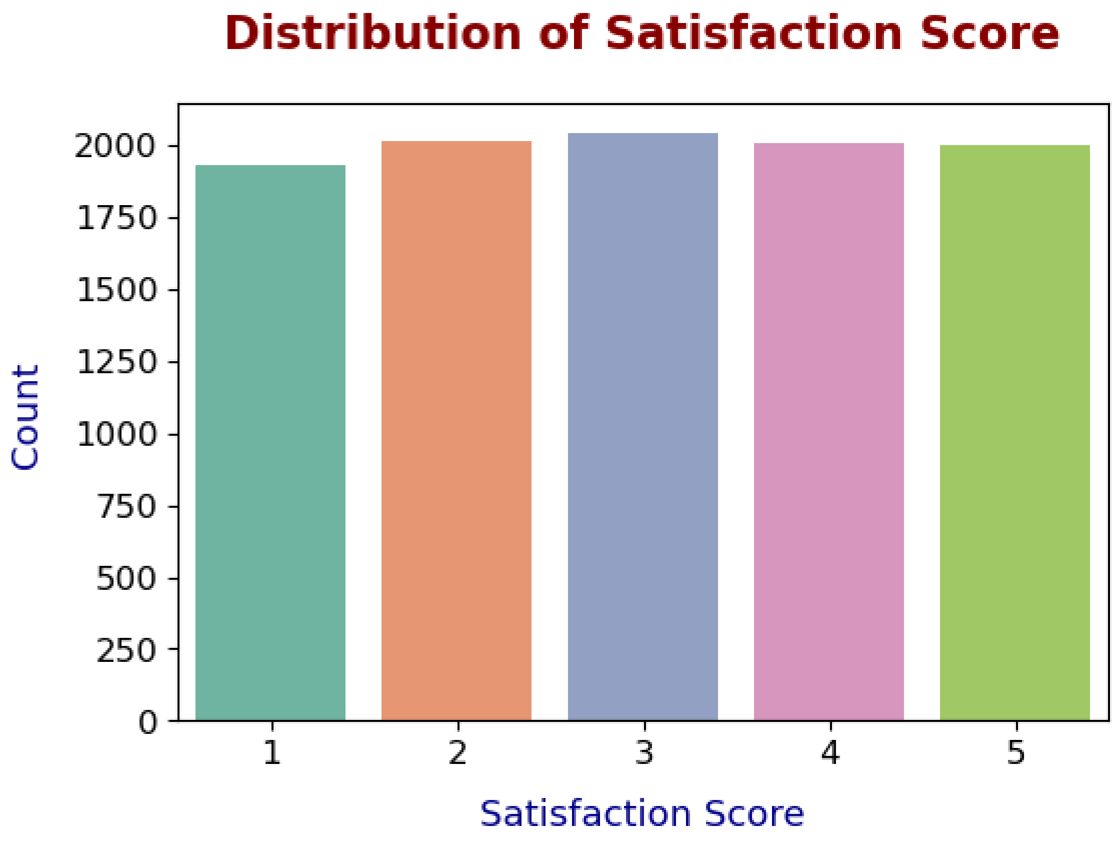
In [80]:
```
# Categorical Variables

categorical_cols = ["Gender", "HasCrCard", "IsActiveMember", "Exited", "Satisfac

for col in categorical_cols:
    plt.figure(figsize=(6,4))
    sns.countplot(x=df[col], hue=df[col], palette="Set2", legend=False)
    plt.title("Distribution of " + col, **title_style, pad=20)
    plt.xlabel(col, **label_style, labelpad=10)
    plt.ylabel("Count", **label_style, labelpad=10)
    plt.xticks(fontsize=12)
    plt.yticks(fontsize=12)
    plt.show()
```

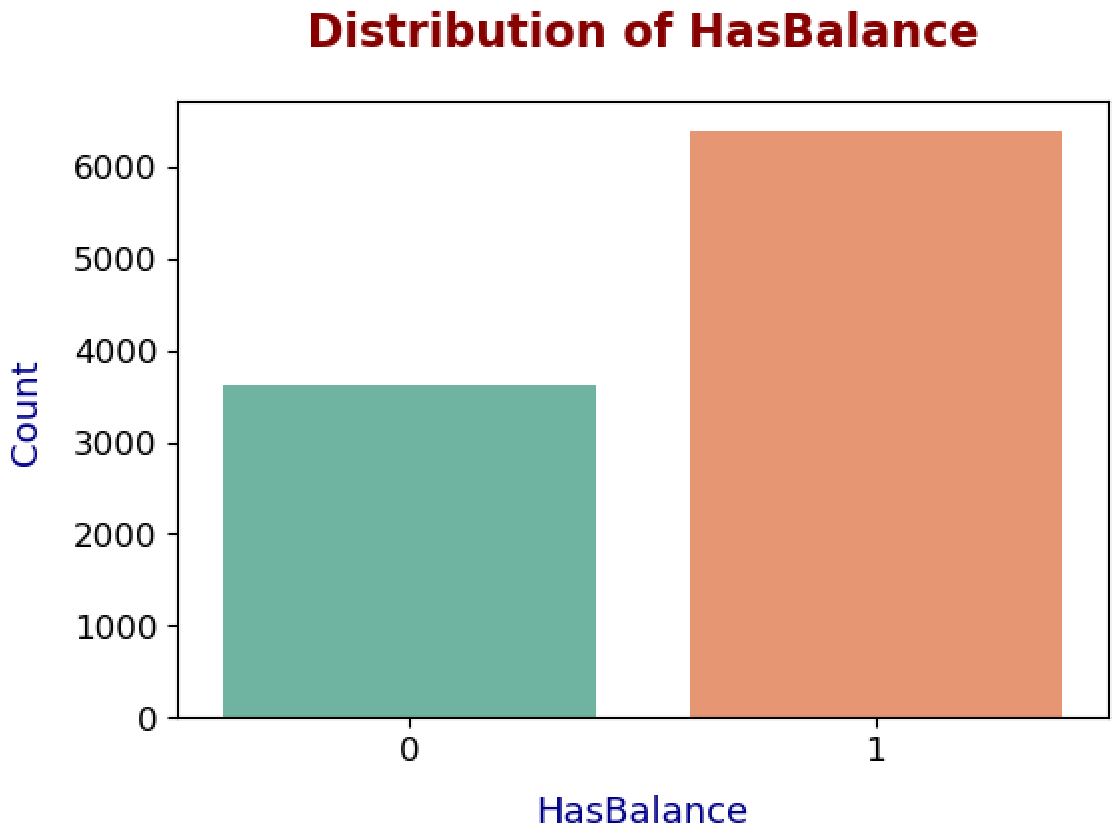

Distribution of Gender
Count
5000
4000
3000
2000
1000
0
Female
Male
Gender


Distribution of HasCrCard
Count
7000
6000
5000
4000
3000
2000
1000
0
0
1
HasCrCard


Distribution of IsActiveMember
Count
5000
4000
3000
2000
1000
0
0
1
IsActiveMember


Distribution of Exited
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
Exited

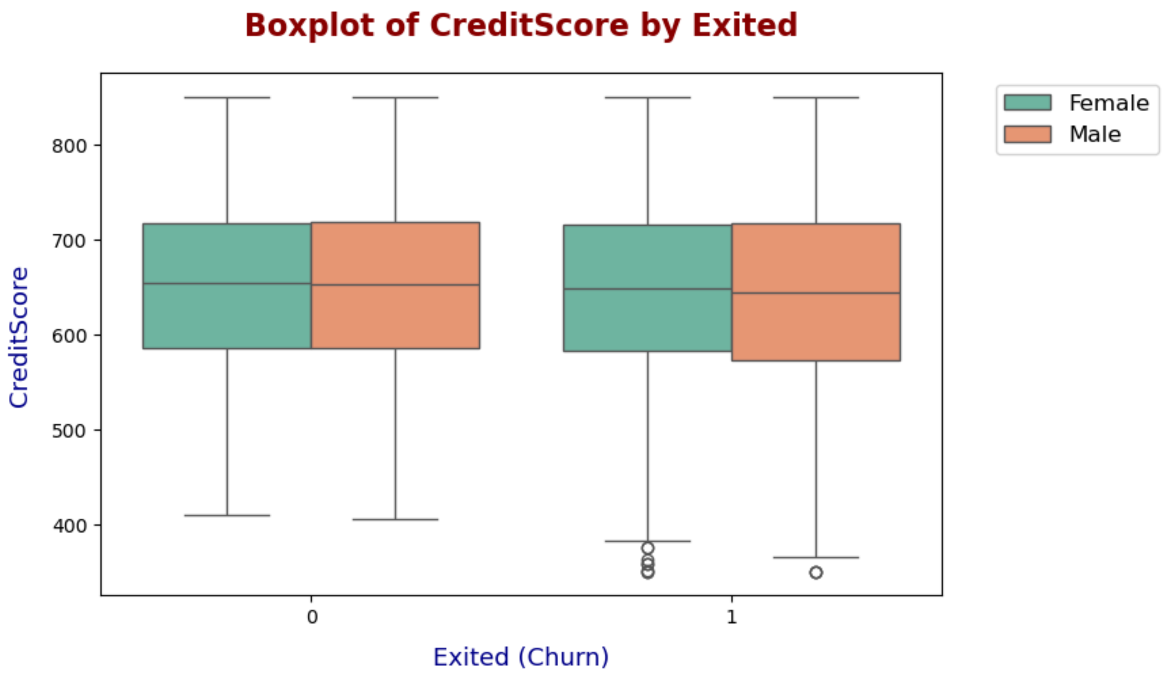


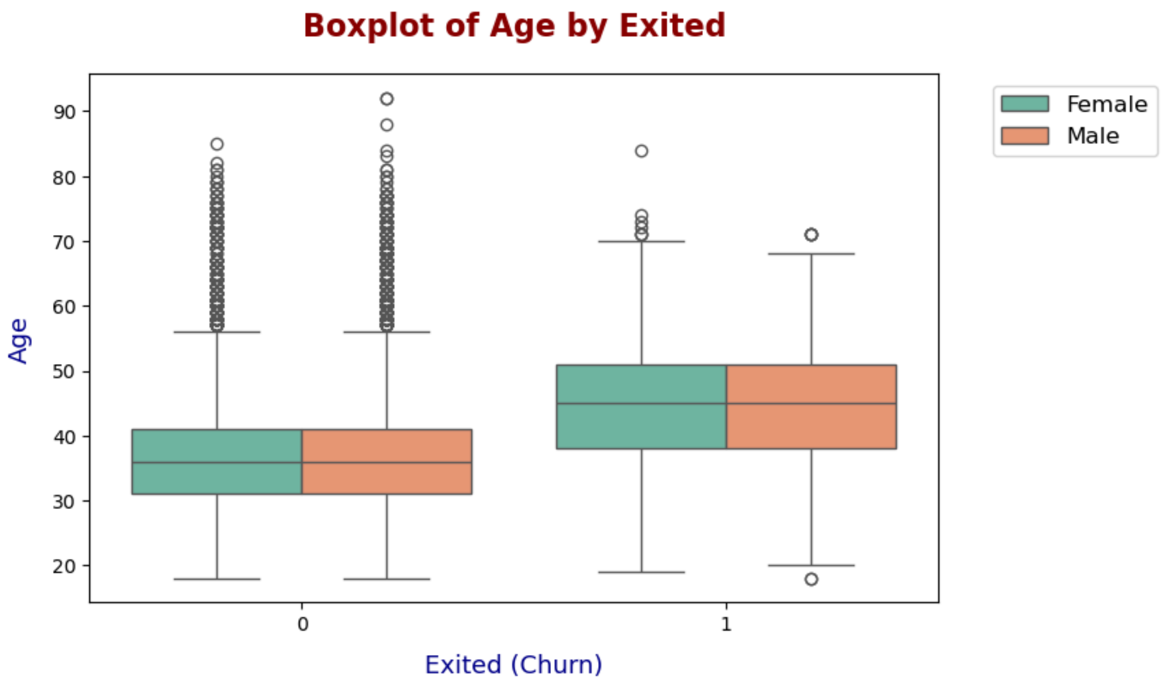


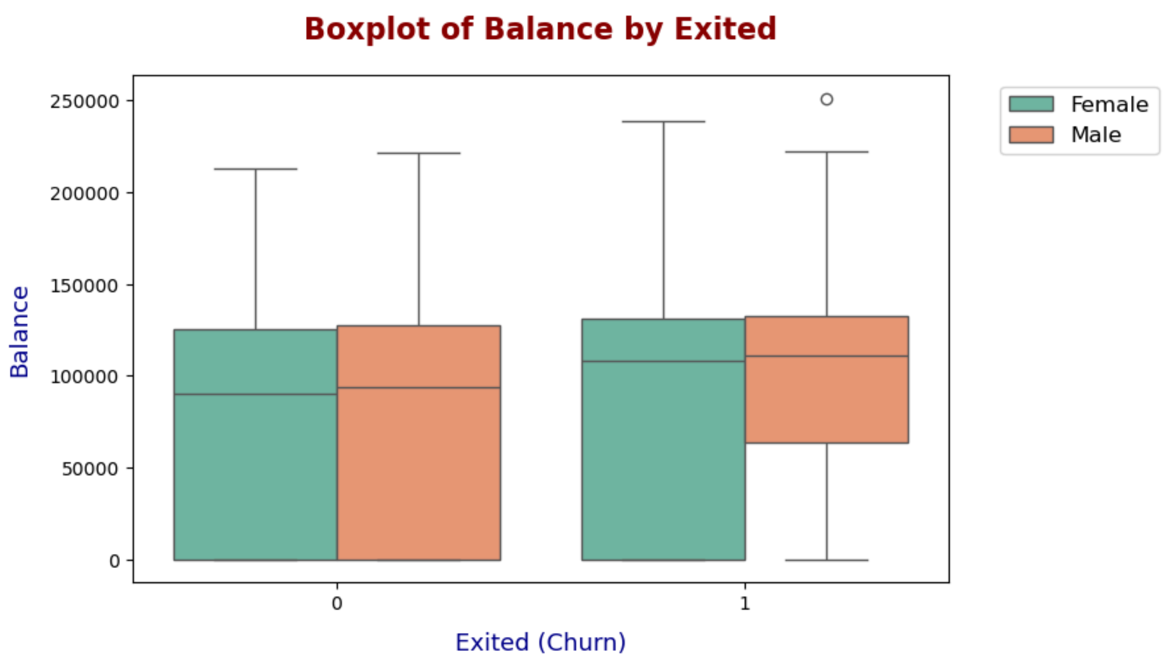
Bivariate Analysis

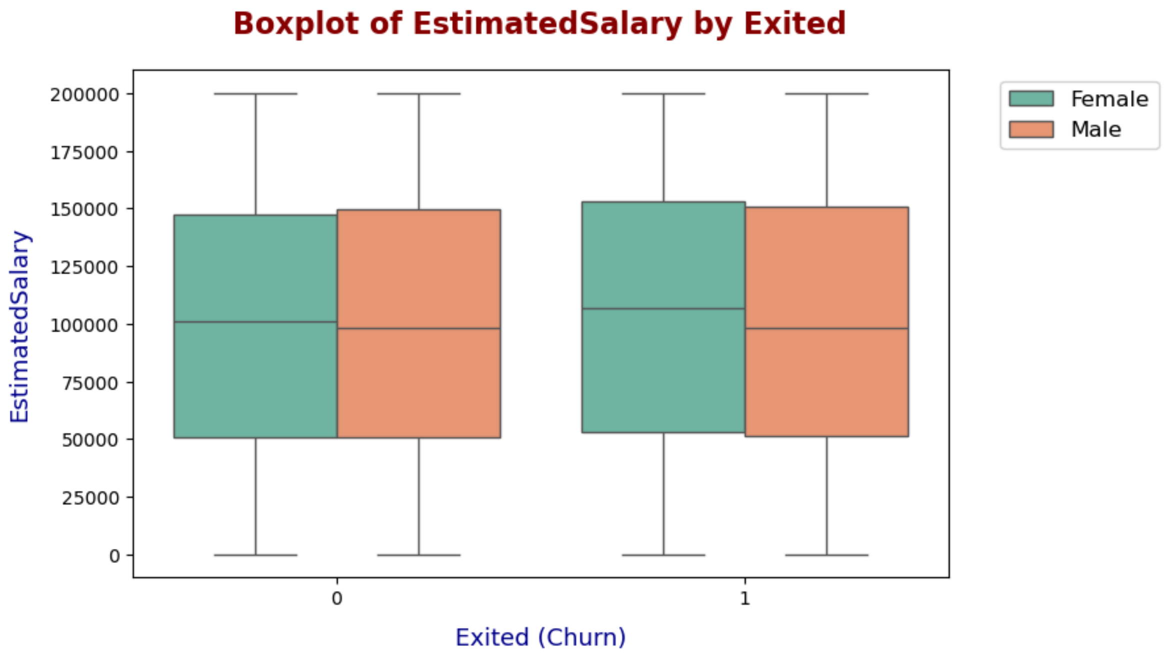
In [81]:
```
#
title_style = {'fontsize':16, 'fontweight':'bold', 'color':'darkred'}
label_style = {'fontsize':13, 'color':'darkblue'}

numeric_cols = ["CreditScore", "Age", "Balance", "EstimatedSalary", "Point Earne
```

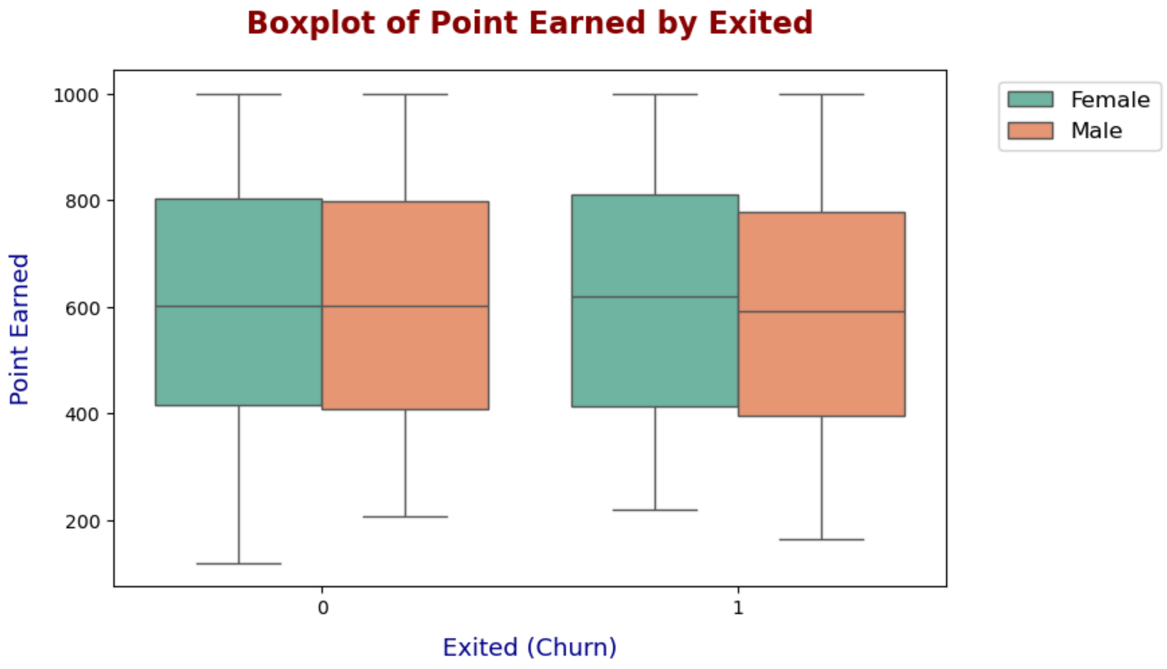
```
for col in numeric_cols:
    plt.figure(figsize=(8,5))
    sns.boxplot(x="Exited", y=col,hue="Gender", data=df, palette="Set2")
    plt.title("Boxplot of " + col + " by Exited", **title_style, pad=20)
    plt.xlabel("Exited (Churn)", **label_style, labelpad=10)
    plt.ylabel(col, **label_style, labelpad=10)
    plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12
    plt.show()
```



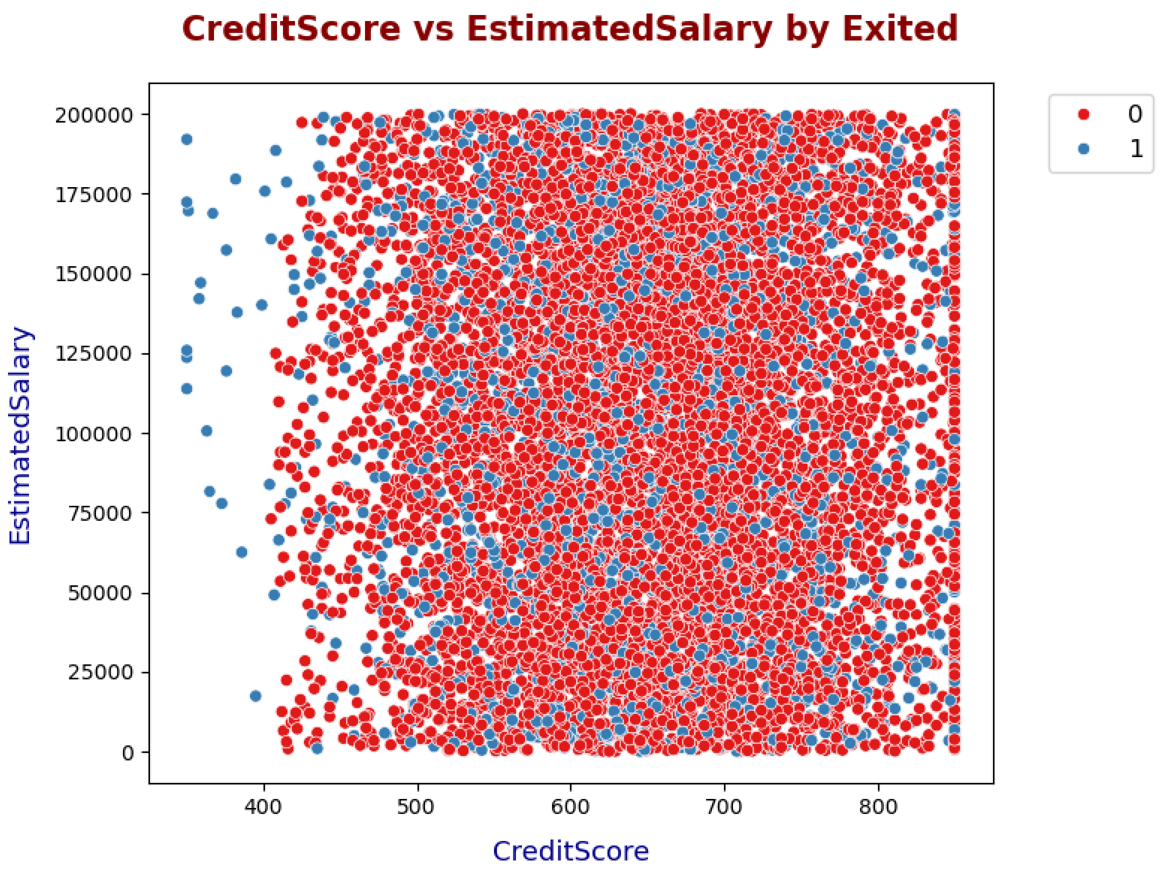



Boxplot of Balance by Exited
Balance
250000
200000
150000
100000
50000
0
0
1
Exited (Churn)
Female
Male


Boxplot of EstimatedSalary by Exited
EstimatedSalary
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
Exited (Churn)
Female
Male

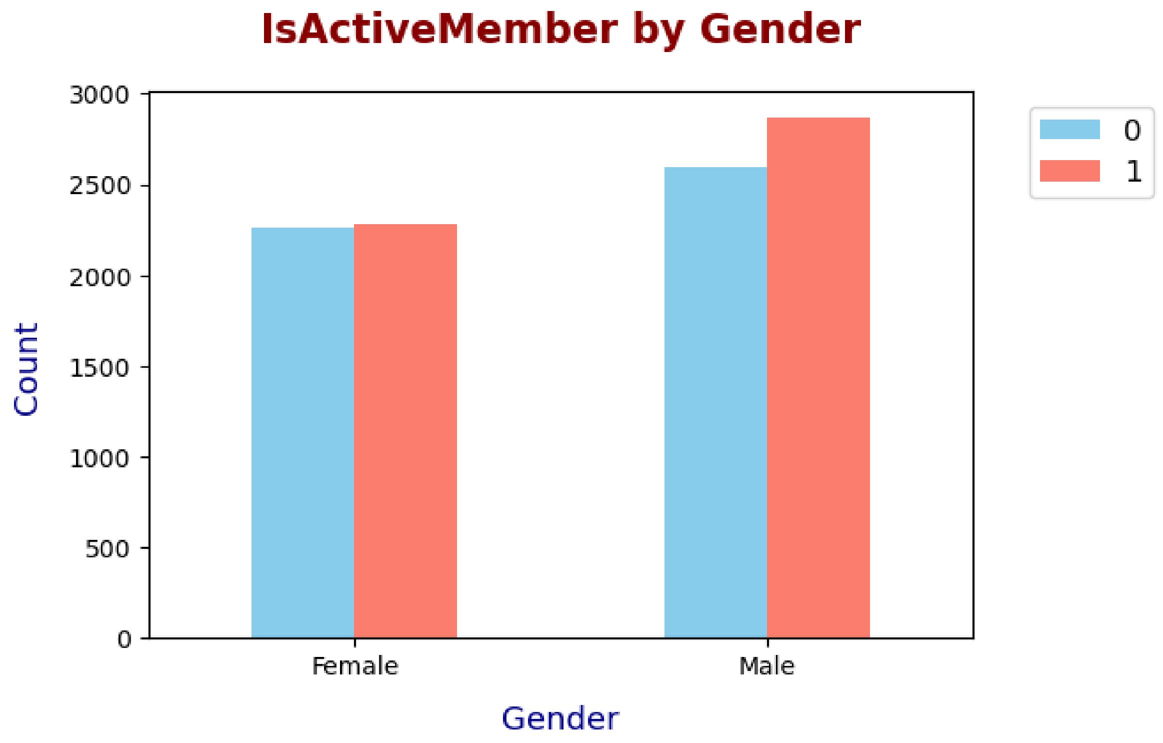

Boxplot of Point Earned by Exited
Point Earned
1000
800
600
400
200
0
1
Exited (Churn)
Female
Male

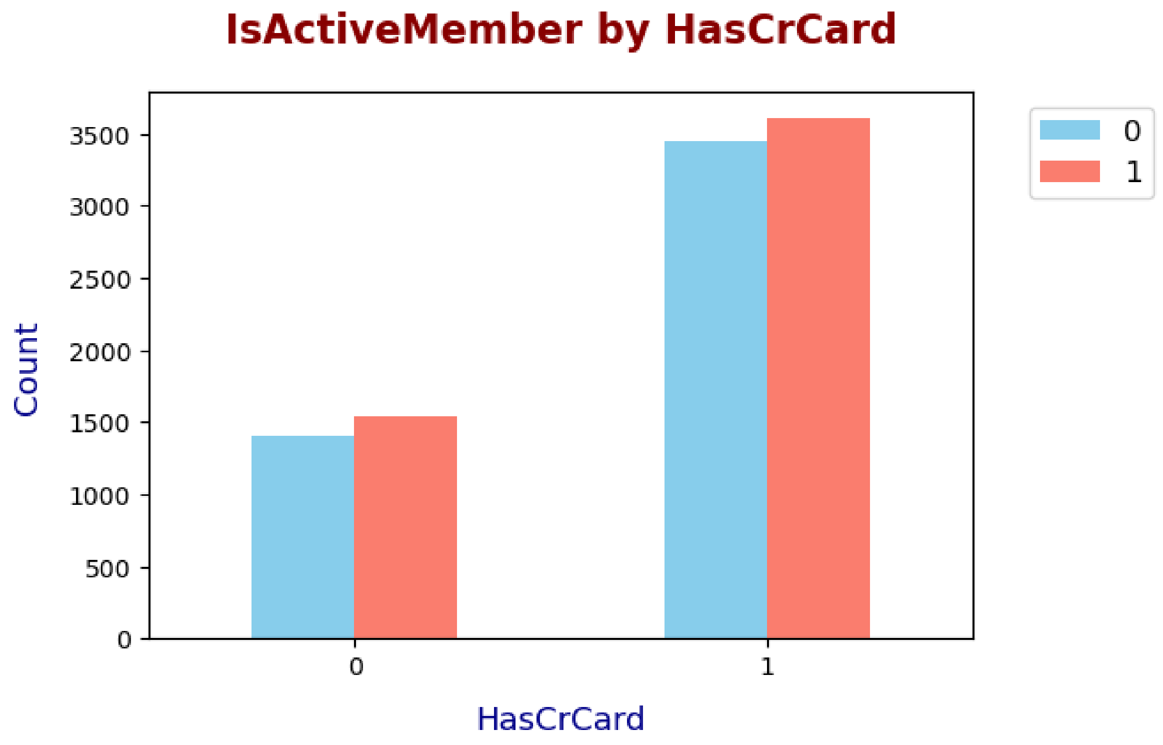
In [82]:
```
#Numeric vs Numeric
#CreditScore vs EstimatedSalary

plt.figure(figsize=(8,6))
sns.scatterplot(
    x="CreditScore",
    y="EstimatedSalary",
    hue="Exited",
    data=df,
    palette="Set1"
)
plt.title("CreditScore vs EstimatedSalary by Exited", fontsize=16, fontweight='b
plt.xlabel("CreditScore", fontsize=13, color='darkblue', labelpad=10)
plt.ylabel("EstimatedSalary", fontsize=13, color='darkblue', labelpad=10)

plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.tight_layout()
plt.show()
```



In [83]:
```
#Categorical vs Categorical

#Gender vs IsActiveMember
gender_active = df.groupby(['Gender','IsActiveMember']).size().unstack()
gender_active.plot(kind='bar', stacked=False, figsize=(6,4), color=["skyblue","s
plt.title("IsActiveMember by Gender", fontsize=16, fontweight='bold', color='dar
plt.xlabel("Gender", fontsize=13, color='darkblue', labelpad=10)
plt.ylabel("Count", fontsize=13, color='darkblue', labelpad=10)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.xticks(rotation=0)
plt.show()

#HasCrCard vs IsActiveMember
cr_active = df.groupby(['HasCrCard','IsActiveMember']).size().unstack()
```

```
cr_active.plot(kind='bar', stacked=False, figsize=(6,4), color=["skyblue","salmc
plt.title("IsActiveMember by HasCrCard", fontsize=16, fontweight='bold', color='
plt.xlabel("HasCrCard", fontsize=13, color='darkblue', labelpad=10)
plt.ylabel("Count", fontsize=13, color='darkblue', labelpad=10)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```

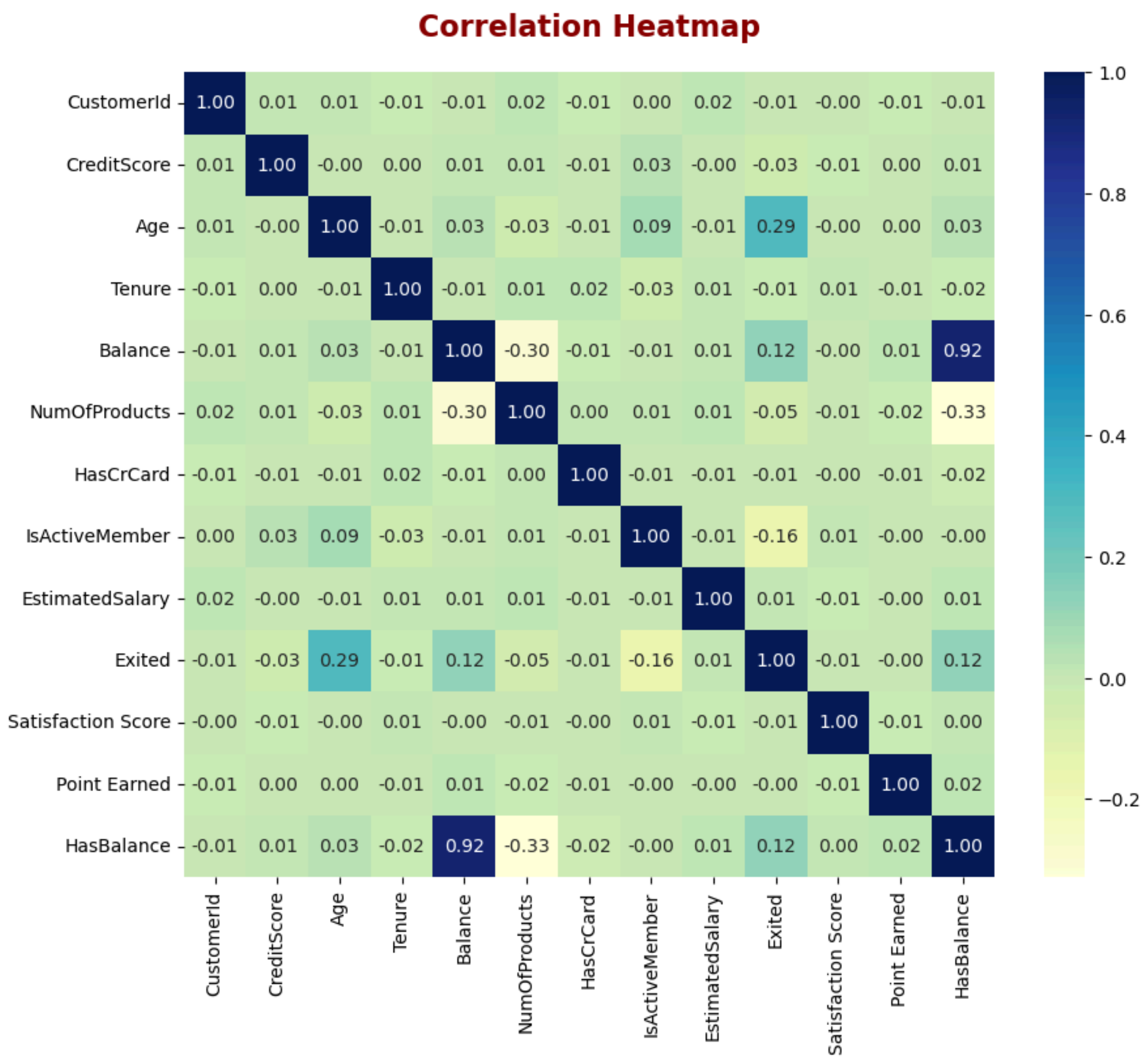




Correlation Heatmap

```
In [84]: plt.figure(figsize=(10,8))
         sns.heatmap(df.corr(numeric_only=True), annot=True, fmt=".2f", cmap="YlGnBu")
```

```
plt.title("Correlation Heatmap", fontsize=16, fontweight='bold', color='darkred'
plt.show()
```

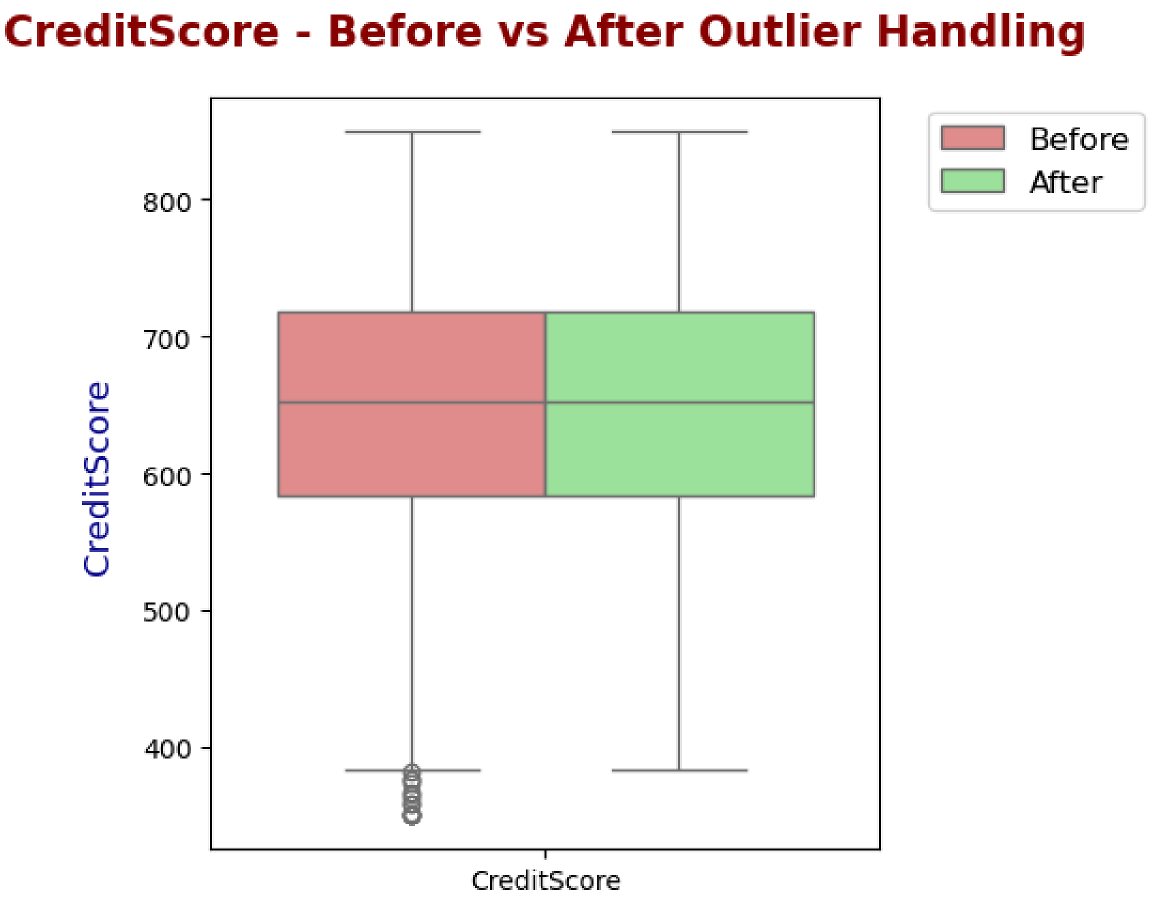


	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Satisfaction Score	Point Earned	HasBalance
CustomerId	1.00	0.01	0.01	-0.01	-0.01	0.02	-0.01	0.00	0.02	-0.01	-0.00	-0.01	-0.01
CreditScore	0.01	1.00	-0.00	0.00	0.01	0.01	-0.01	0.03	-0.00	-0.03	-0.01	0.00	0.01
Age	0.01	-0.00	1.00	-0.01	0.03	-0.03	-0.01	0.09	-0.01	0.29	-0.00	0.00	0.03
Tenure	-0.01	0.00	-0.01	1.00	-0.01	0.01	0.02	-0.03	0.01	-0.01	0.01	-0.01	-0.02
Balance	-0.01	0.01	0.03	-0.01	1.00	-0.30	-0.01	-0.01	0.01	0.12	-0.00	0.01	0.92
NumOfProducts	0.02	0.01	-0.03	0.01	-0.30	1.00	0.00	0.01	0.01	-0.05	-0.01	-0.02	-0.33
HasCrCard	-0.01	-0.01	-0.01	0.02	-0.01	0.00	1.00	-0.01	-0.01	-0.01	-0.00	-0.01	-0.02
IsActiveMember	0.00	0.03	0.09	-0.03	-0.01	0.01	-0.01	1.00	-0.01	-0.16	0.01	-0.00	-0.00
EstimatedSalary	0.02	-0.00	-0.01	0.01	0.01	0.01	-0.01	-0.01	1.00	0.01	-0.01	-0.00	0.01
Exited	-0.01	-0.03	0.29	-0.01	0.12	-0.05	-0.01	-0.16	0.01	1.00	-0.01	-0.00	0.12
Satisfaction Score	-0.00	-0.01	-0.00	0.01	-0.00	-0.01	-0.00	0.01	-0.01	-0.01	1.00	-0.01	0.00
Point Earned	-0.01	0.00	0.00	-0.01	0.01	-0.02	-0.01	-0.00	-0.00	-0.00	-0.01	1.00	0.02
HasBalance	-0.01	0.01	0.03	-0.02	0.92	-0.33	-0.02	-0.00	0.01	0.12	0.00	0.02	1.00

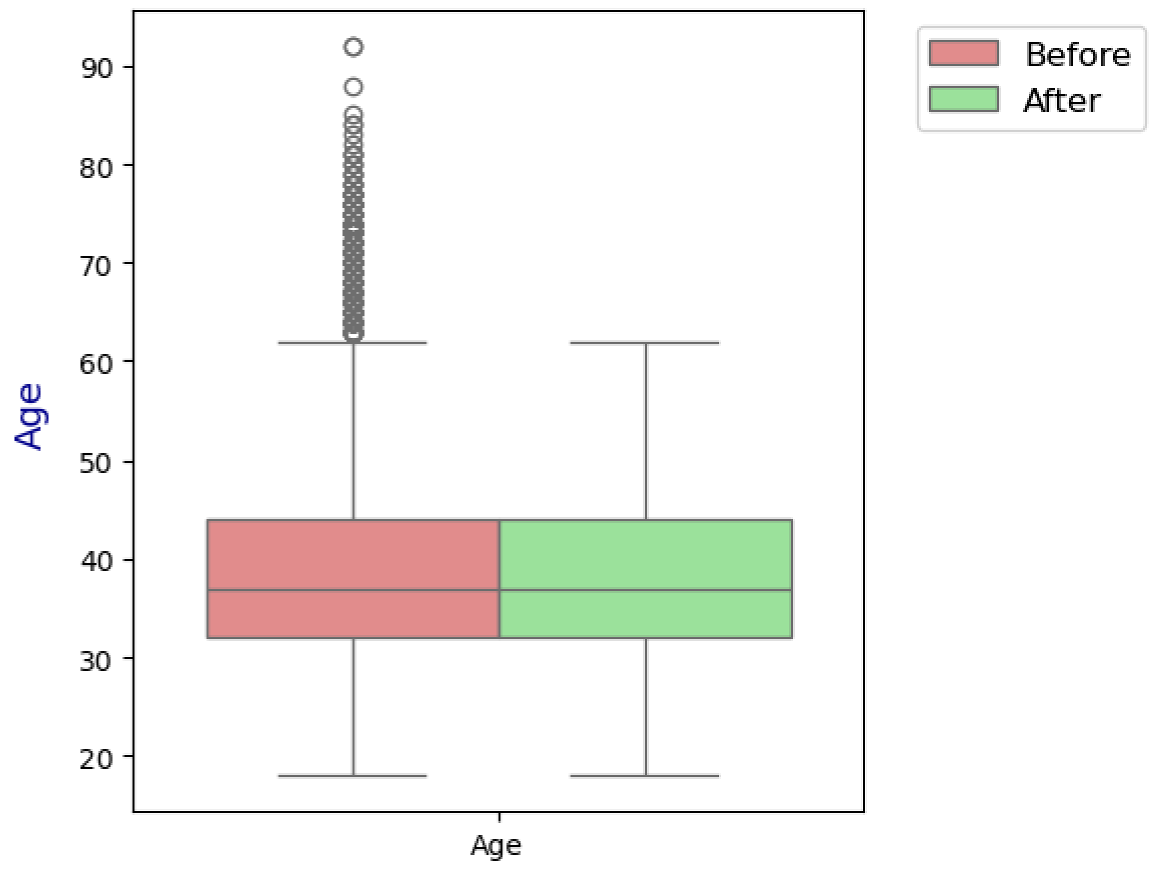
Outlier Analysis

In [85]:
```
numeric_cols = ["CreditScore", "Age", "Balance", "EstimatedSalary", "Point Earne

# Step 1: Handle outliers
df_out = df.copy()
for col in numeric_cols:
    Q1 = df_out[col].quantile(0.25)
    Q3 = df_out[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5*IQR
    upper_bound = Q3 + 1.5*IQR
    df_out[col] = df_out[col].apply(lambda x: lower_bound if x < lower_bound els

# Step 2: Prepare combined DataFrame
df_before = df[numeric_cols].copy()
df_before['Status'] = 'Before'
df_after = df_out[numeric_cols].copy()
df_after['Status'] = 'After'

df_combined = pd.concat([df_before, df_after], axis=0)
df_melted = df_combined.melt(id_vars='Status', value_vars=numeric_cols, var_name

# Step 3: Plot each numeric column
```

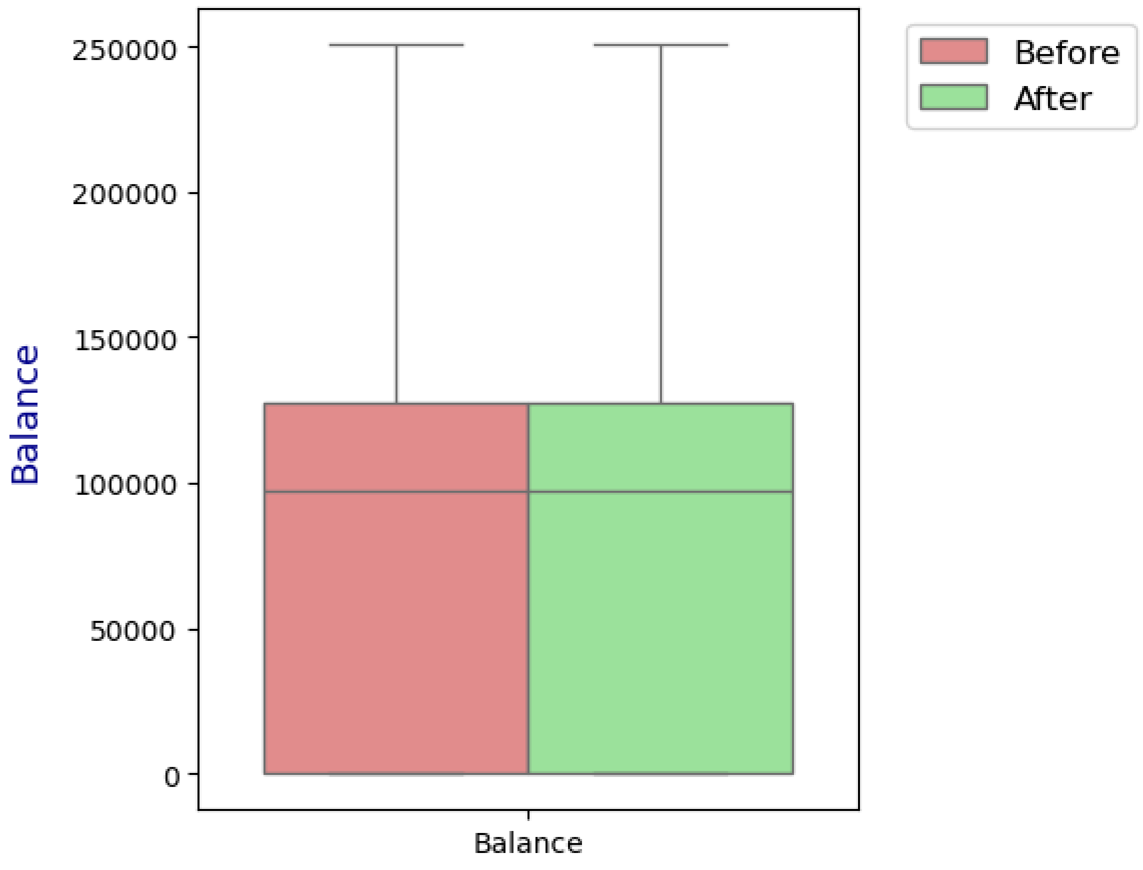
```
for col in numeric_cols:
    plt.figure(figsize=(6,5))
    sns.boxplot(x='Feature', y='Value', hue='Status',
                data=df_melted[df_melted['Feature']==col],
                palette=['lightcoral','lightgreen'])
    plt.title(f"{col} - Before vs After Outlier Handling", fontsize=16, fontweig
    plt.xlabel('')
    plt.ylabel(col, fontsize=13, color='darkblue', labelpad=10)


    plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12

    plt.tight_layout()
    plt.show()
```

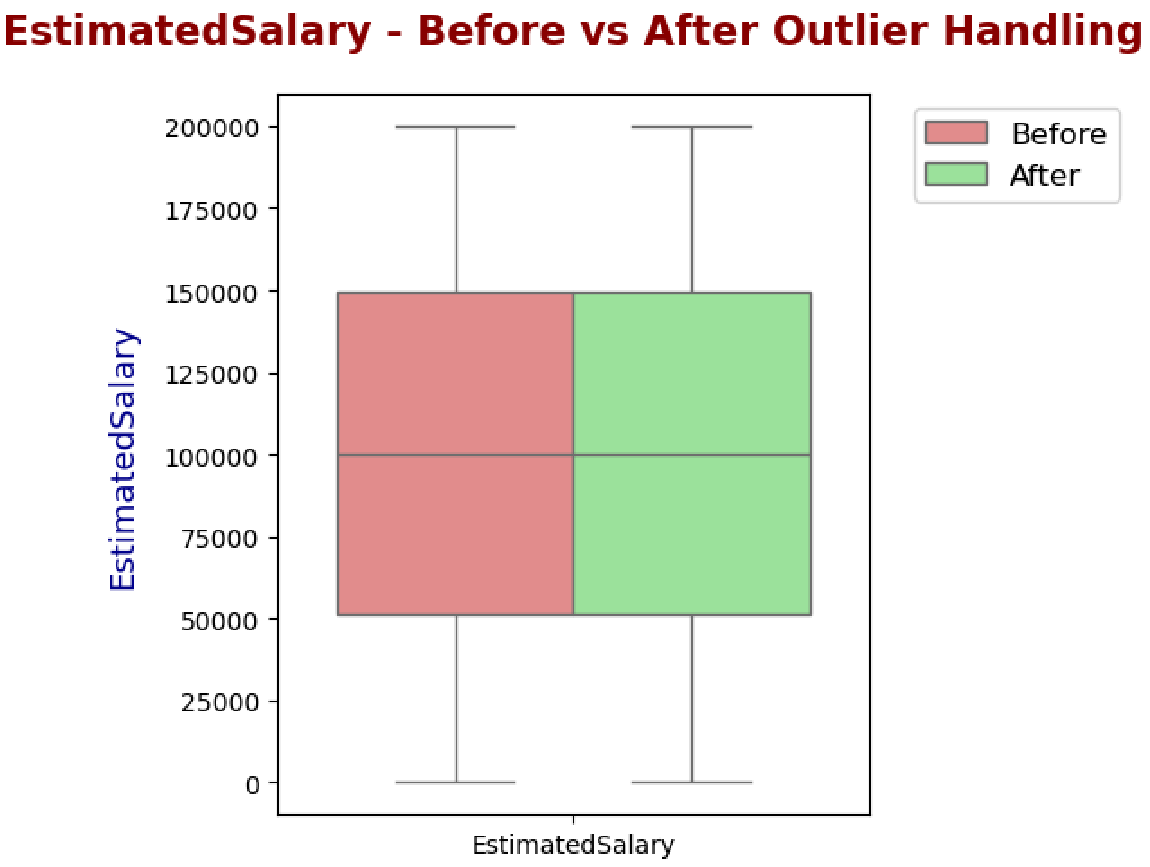


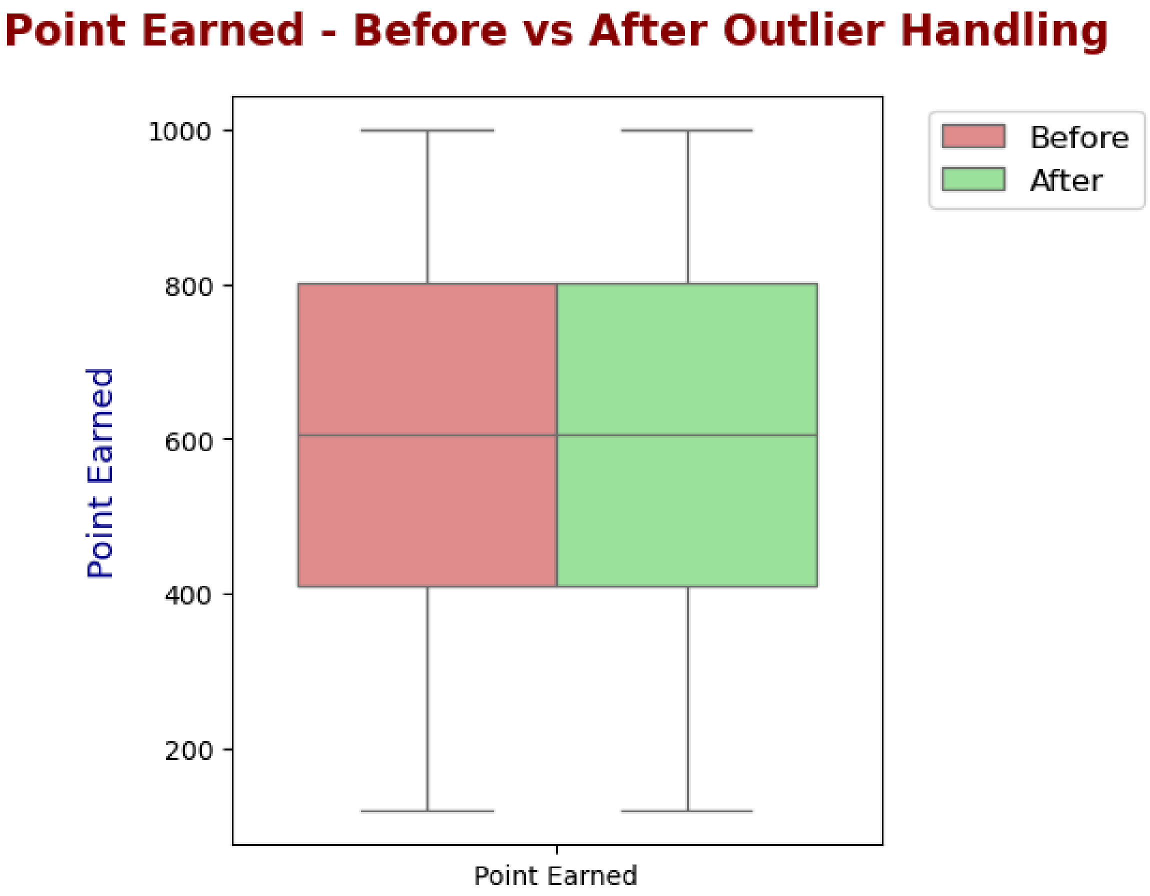
Age - Before vs After Outlier Handling



Balance - Before vs After Outlier Handling







Groupby and Pivot Table Analysis

In [86]:
```
#GroupBy Analysis

# Average balance by Exited status
```

```
df.groupby('Exited')['Balance'].mean()

# Count of customers by HasCrCard and Exited
df.groupby(['HasCrCard', 'Exited']).size()

# Multiple aggregations at once
df.groupby('Exited').agg({'Balance':'mean', 'CreditScore':'mean', 'EstimatedSala
```

Out[86]:

	Balance	CreditScore	EstimatedSalary
Exited			
0	72742.750663	651.837855	99726.853141
1	91109.476006	645.414622	101509.908783

In [87]:

```
#Pivot Table Analysis

# Average Balance by HasCrCard and Exited
pivot = pd.pivot_table(df, values='Balance', index='HasCrCard', columns='Exited'
print("\n🔷Average Balance by HasCrCard and Exited :\n")
print(pivot,"\n")
print("🌟" * 50)

# Count of customers by IsActiveMember and Exited
pivot1 = pd.pivot_table(df, values='CustomerId', index='IsActiveMember', columns
print("\n🔷Count of customers by IsActiveMember and Exited :\n")
print(pivot1)
```

```
🔷Average Balance by HasCrCard and Exited :

Exited                0            1
HasCrCard
0          74238.387294  91929.527194
1          72123.243625  90756.710126

🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟
🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟🌟

🔷Count of customers by IsActiveMember and Exited :

Exited             0     1
IsActiveMember
0               3546  1303
1               4416   735
```

Finding and Insight of EDA

- 🔷 The majority of customers are in the **30–50 age group**, while only a **small fraction are above 60**. Most hold **1–2 products**, and very few own 3–4.
- 🔷 A large share of customers have a **zero account balance**, while a few customers show **very high balances (outliers)**. Estimated salaries are spread widely, with most between **50K–150K**.
- 🔷 Customers with **low credit scores, low balances, or inactive memberships** are more likely to **exit**. Exited customers also show

relatively **lower satisfaction scores**.

🔷 **Active members, credit card holders, and customers with multiple products** have a **higher retention rate**, highlighting the importance of engagement and cross-selling.

🔷 **Balance and Points Earned** show a moderate positive correlation, whereas **Credit Score and Estimated Salary** have **near zero correlation (−0.00)**, meaning they are independent factors.

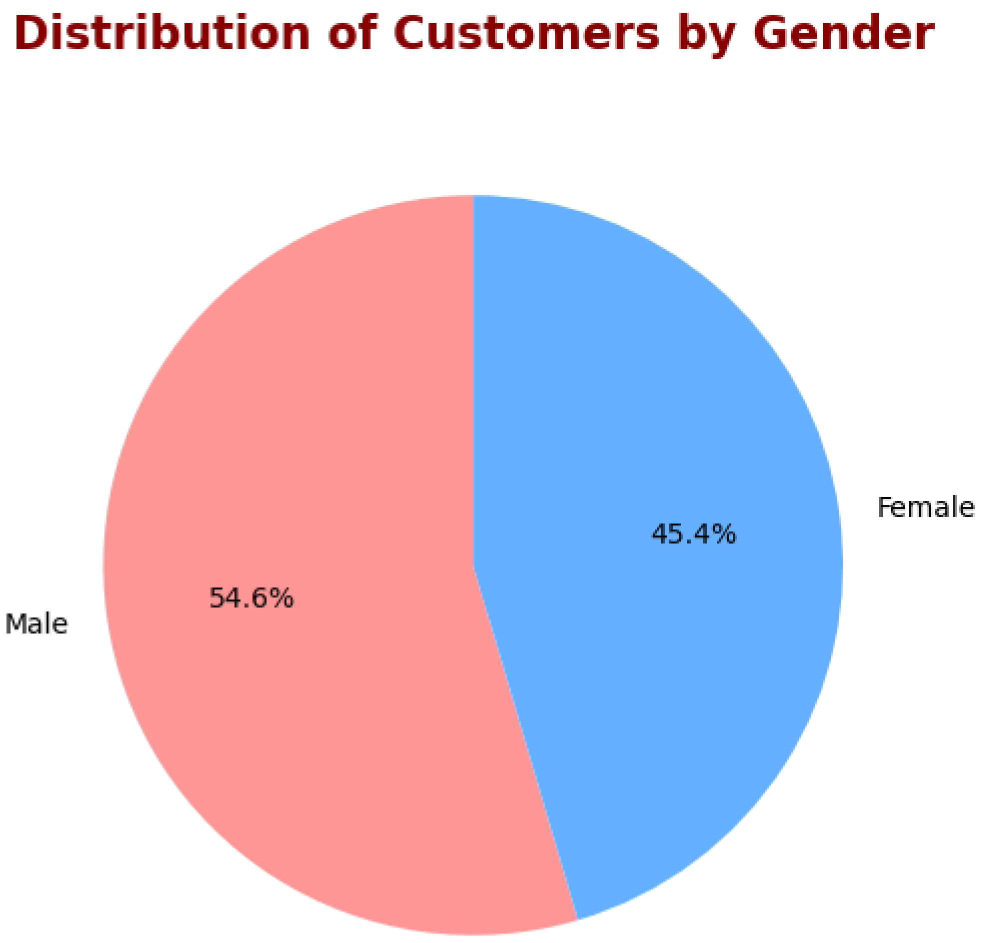
🔷 GroupBy and Pivot analysis show that **inactive customers and non–credit card holders** contribute more to churn, while **high-balance and multi-product customers** are more loyal and profitable.

Overall Insight: Customer churn is strongly linked to **financial activity, engagement, and satisfaction**. Retention strategies should focus on activating low-balance and inactive customers while nurturing high-value segments.

Visualization

In [88]:
```
title_style = {'fontsize':16, 'fontweight':'bold', 'color':'darkred','pad':20}
label_style = {'fontsize':13, 'color':'blue','labelpad':10}
```

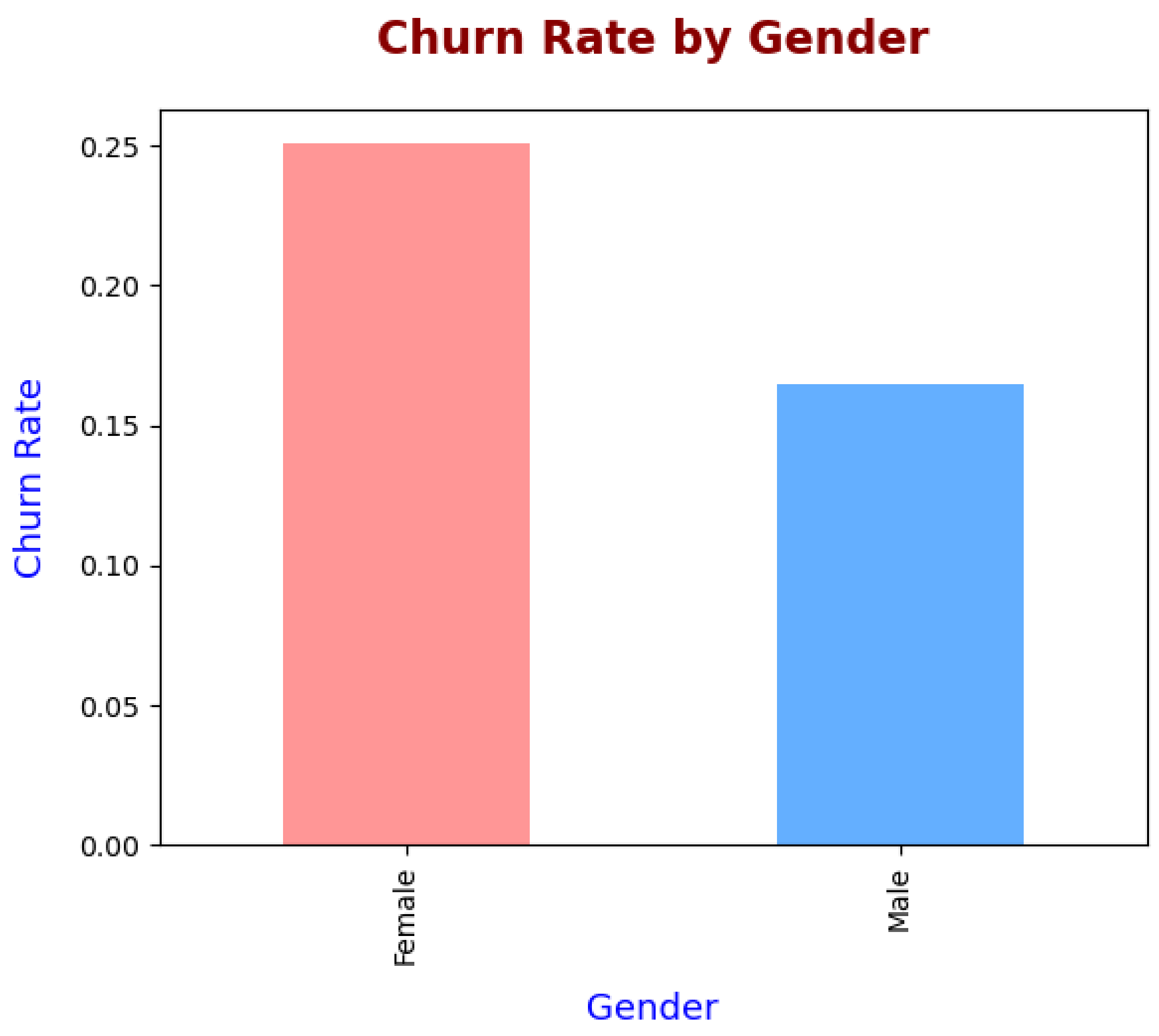
In [89]:
```
# Distribution of Customers by Gender
plt.figure(figsize=(6,6))
df['Gender'].value_counts().plot.pie(autopct='%1.1f%%', colors=['#FF9999','#66B2
plt.title("Distribution of Customers by Gender", **title_style)
plt.ylabel('')
plt.show()
```



Distribution of Customers by Gender

- **Male customers (54.6%) are slightly higher** compared to female customers (45.4%).
- **The distribution is fairly balanced**, so both genders are well represented in the dataset.
- **This balance allows meaningful churn comparison** between male and female customers without bias.

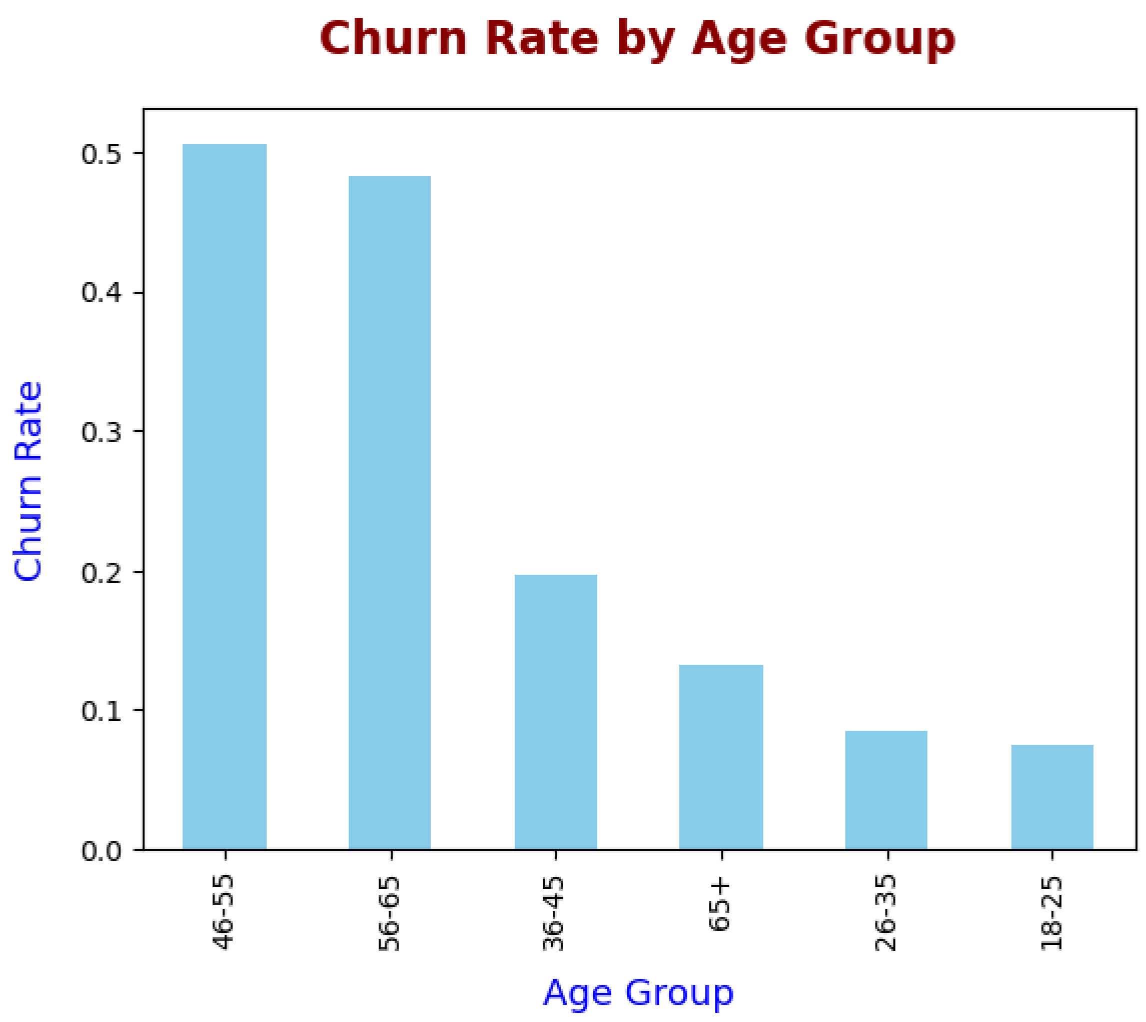
In [90]:
```
# Churn Rate by Gender
churn_gender = df.groupby('Gender')['Exited'].mean().sort_values(ascending=False
churn_gender.plot(kind='bar', color=['#FF9999','#66B2FF'])
plt.title("Churn Rate by Gender", **title_style)
plt.ylabel("Churn Rate", **label_style)
plt.xlabel("Gender", **label_style)
plt.show()
```



Churn Rate by Gender

- **Female customers have a higher churn rate (~25%)** compared to male customers (~16%).
- **This indicates that women are more likely to leave the bank** despite being fewer in number overall.
- **Possible reasons could include dissatisfaction with services, product suitability, or trust factors**, which the bank needs to address for female customers.

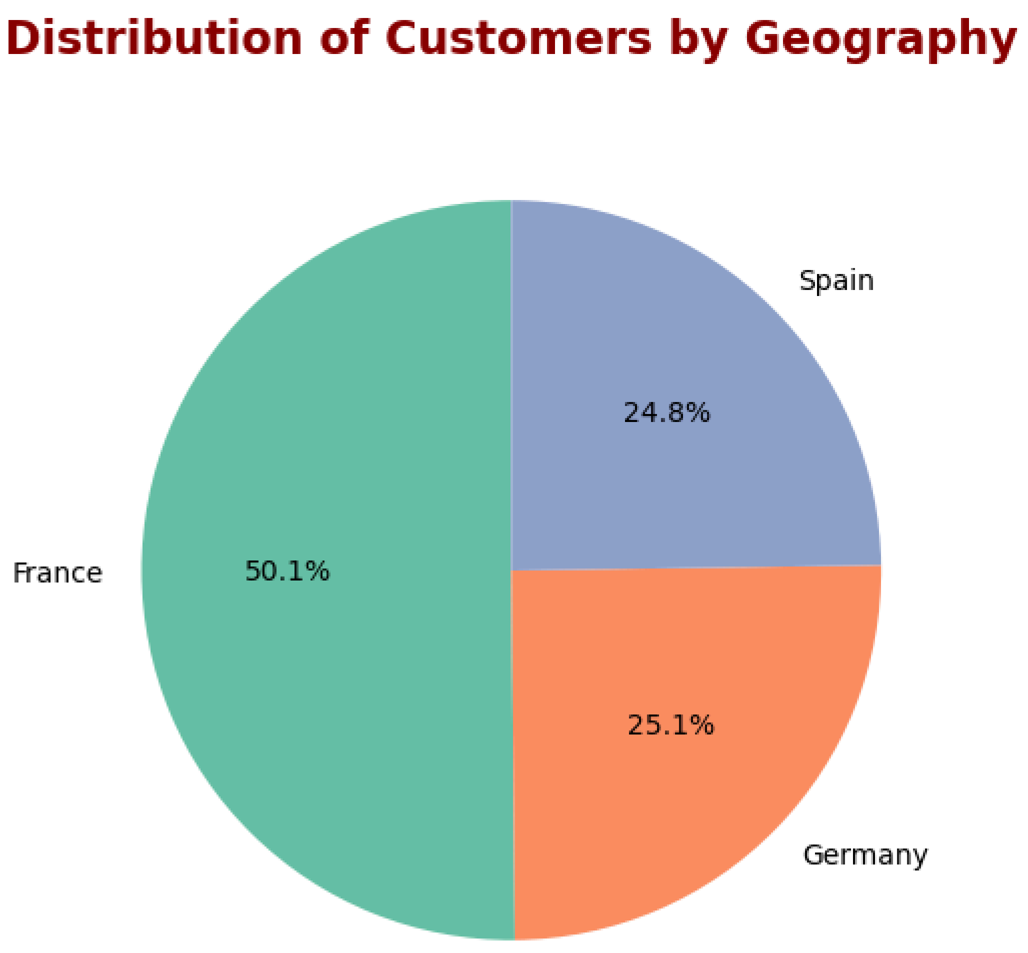
In [91]:
```
#Churn Rate by AgeGroup
churn_age = df.groupby('AgeGroup', observed=True)['Exited'].mean().sort_values(a
churn_age.plot(kind='bar', color='skyblue')
plt.title("Churn Rate by Age Group", **title_style)
plt.xlabel("Age Group", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.show()
```



Churn Rate by Gender

- **Female customers show a higher churn rate (~25%)** compared to male customers (~16%).
- **This suggests that female customers are more likely to leave the bank**, even if they represent a smaller portion of the total customer base.
- **Potential reasons could include dissatisfaction with services, product misalignment, or trust issues**, which the bank should investigate to improve retention among female clients.

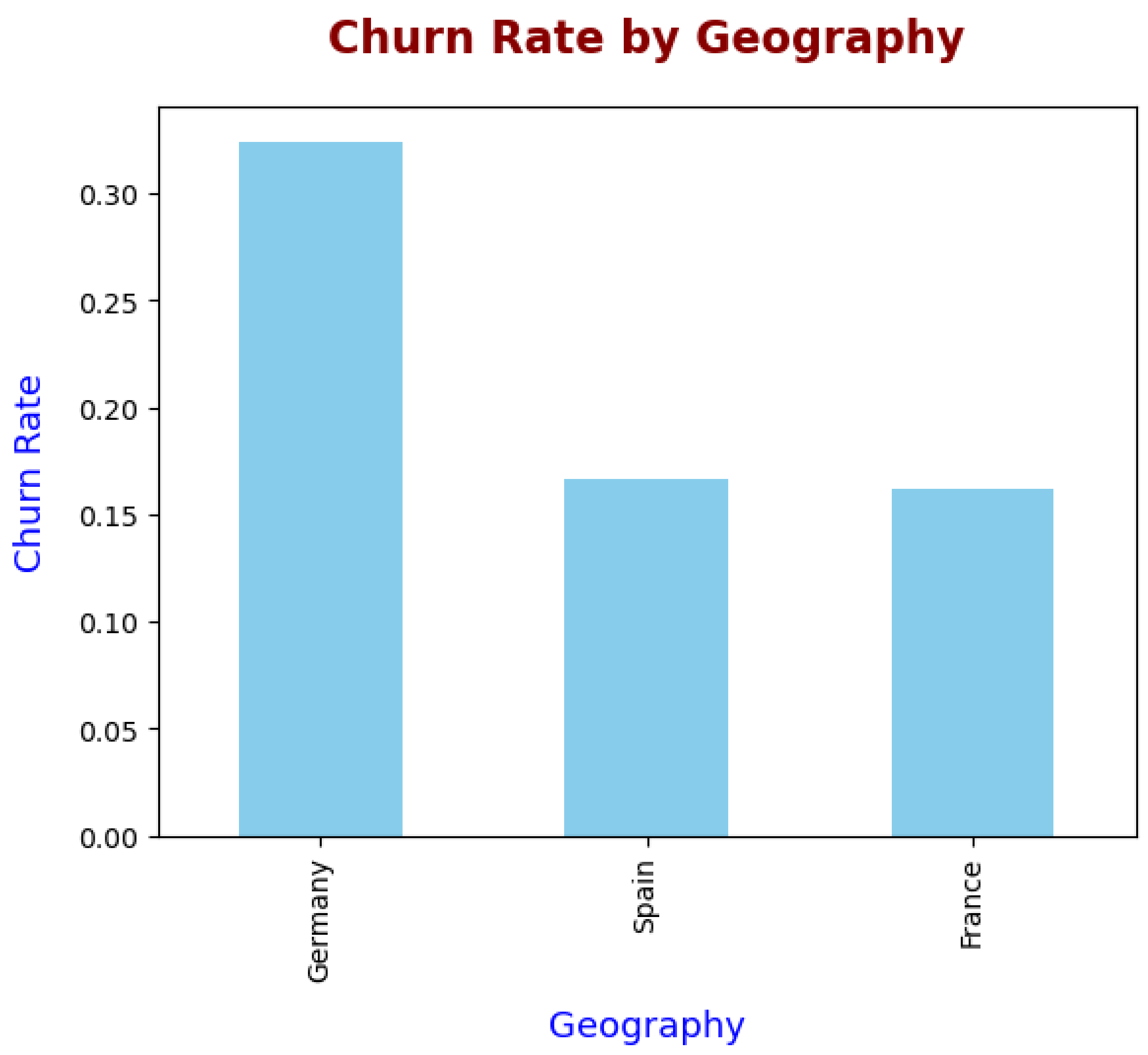
In [92]:
```
#Geography Distribution - Pie
plt.figure(figsize=(6,6))
df['Geography'].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_pale
plt.title("Distribution of Customers by Geography", **title_style)
plt.ylabel('')
plt.show()
```



Distribution of Customers by Geography

- **France has the highest share of customers (~50%)**, making it the primary market for the bank.
- **Germany and Spain have almost equal representation**, around 25% each, indicating moderate customer presence in these countries.
- **The bank may focus on France for growth strategies**, while exploring opportunities to expand its customer base in Germany and Spain.

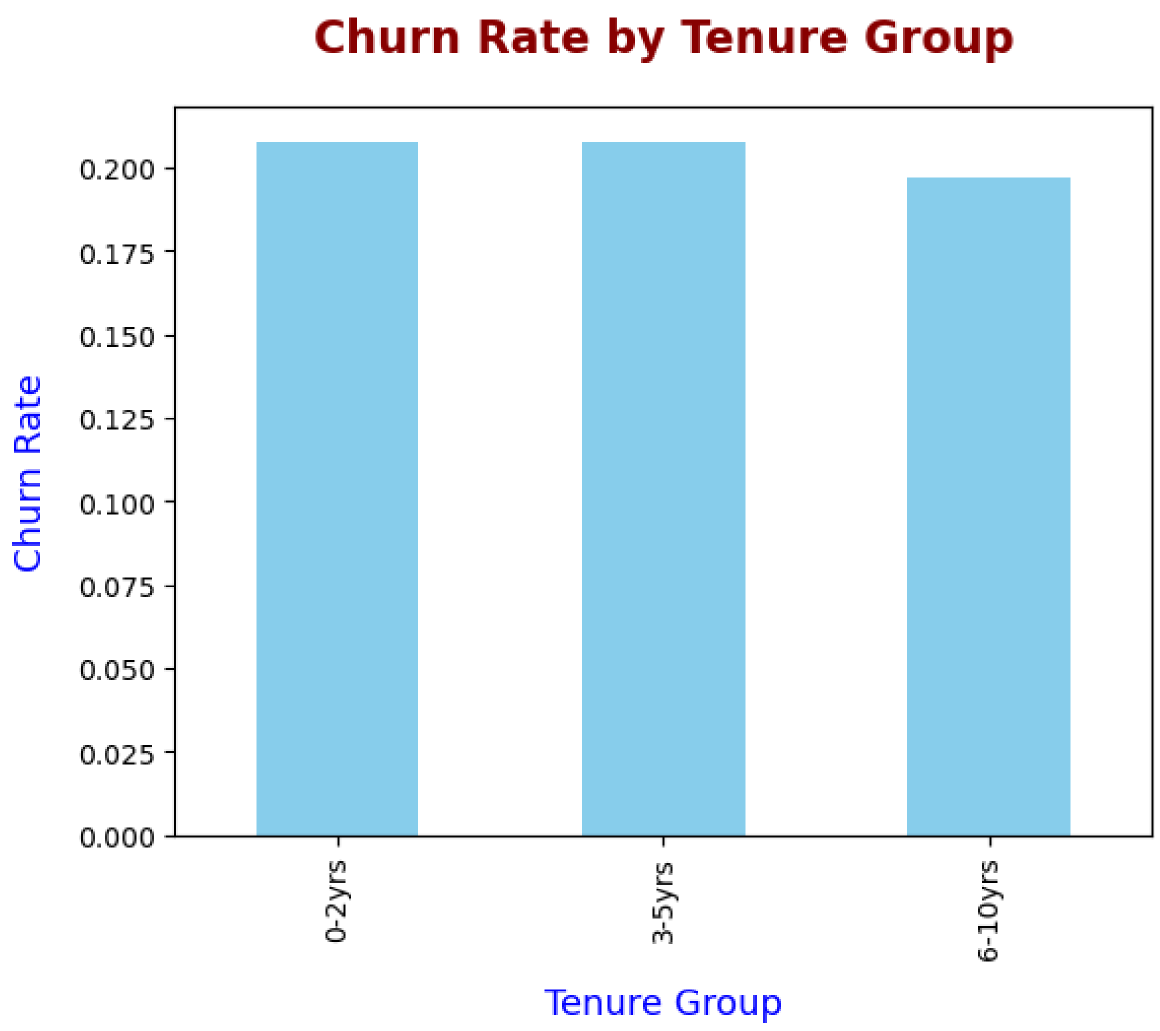
In [93]:
```
#  Churn Rate by Geography
churn_geo = df.groupby('Geography', observed=True)['Exited'].mean().sort_values(
churn_geo.plot(kind='bar', color='skyblue')
plt.title("Churn Rate by Geography", **title_style)
plt.xlabel("Geography", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.show()
```



Churn Rate by Geography

- **Germany has the highest churn rate (~32%)**, much higher than Spain (~17%) and France (~16%).
- **Customers in Germany are almost twice as likely to leave** compared to customers in Spain or France.
- **This suggests potential dissatisfaction specific to the German market**—possibly due to product offerings, competition, or service experience. The bank should investigate and address these regional issues to reduce churn in Germany.

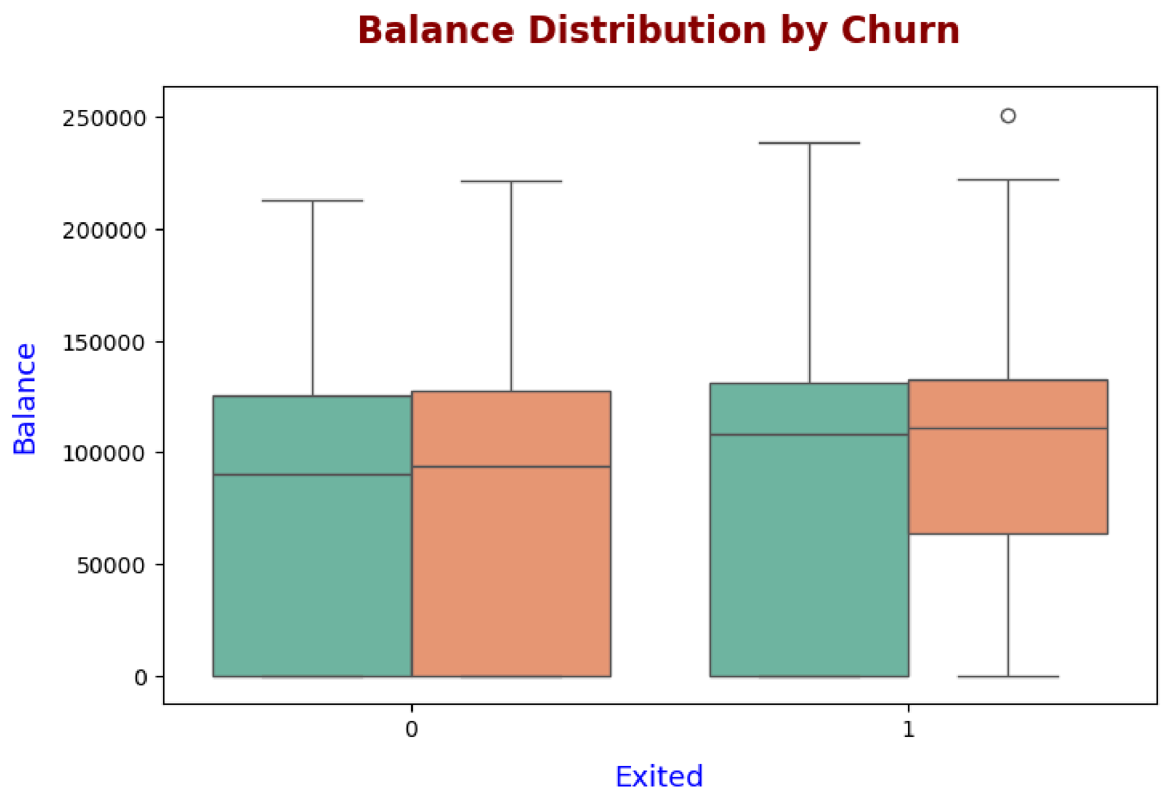
In [94]:
```
#Churn Rate by TenureGroup
churn_tenure = df.groupby('TenureGroup', observed=True)['Exited'].mean().sort_va
churn_tenure.plot(kind='bar', color='skyblue')
plt.title("Churn Rate by Tenure Group", **title_style)
plt.xlabel("Tenure Group", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.show()
```



Churn Rate by Tenure Group

- **Customers with 0–2 years and 3–5 years of tenure show similar churn rates (~21%).**
- **Customers with longer tenure (6–10 years) have a slightly lower churn rate (~20%).**
- **This suggests that customer loyalty increases slightly with tenure,** but the difference is small. The bank should focus on improving early customer experience to prevent churn in the first 5 years.

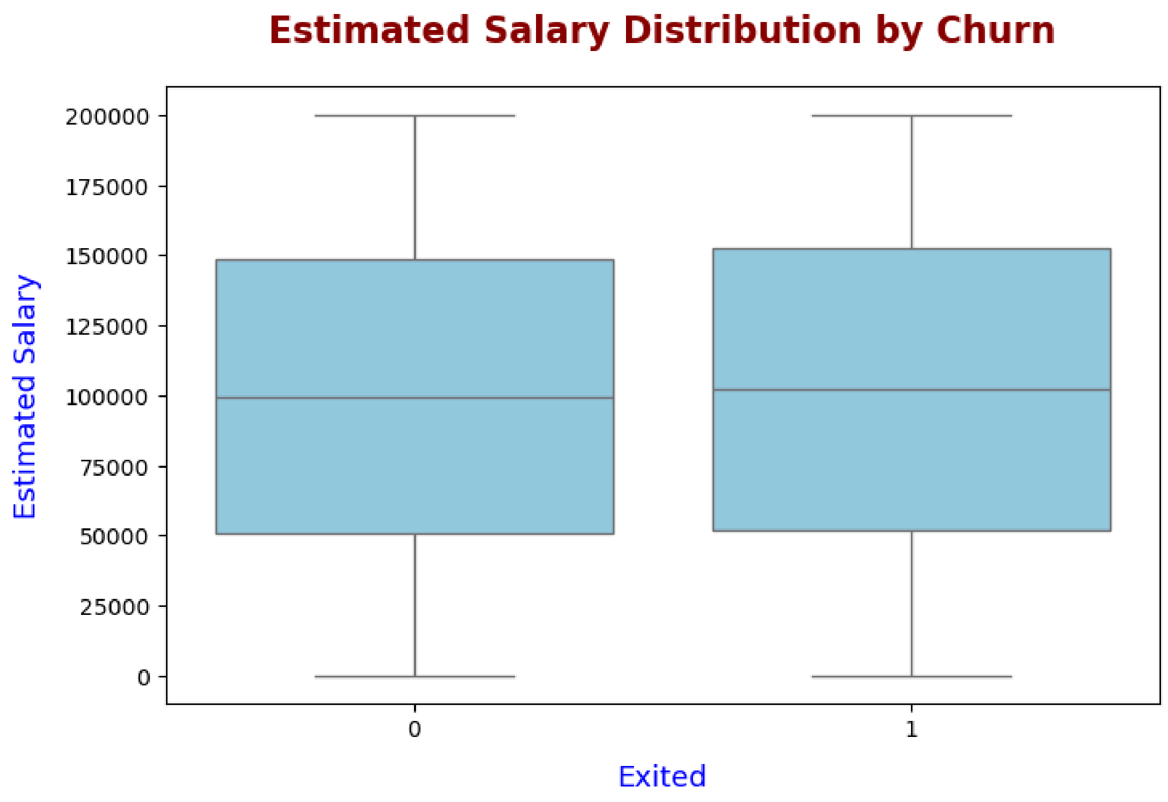
In [95]:
```
#  Balance Distribution by Churn
plt.figure(figsize=(8,5))
sns.boxplot(x='Exited', y='Balance',hue="Gender",legend=False, data=df, palette=
plt.title("Balance Distribution by Churn", **title_style)
plt.xlabel("Exited", **label_style)
plt.ylabel("Balance", **label_style)
plt.show()
```



Balance Distribution by Churn

- **Customers who churned (Exited = 1) tend to have higher average balances** compared to those who stayed (Exited = 0).
- **Both groups show a wide variation in balances,** with many customers at zero balance as well as very high balances.
- **This indicates that customers with higher balances are more likely to leave,** which could mean that wealthier customers are dissatisfied with the services or finding better alternatives elsewhere.

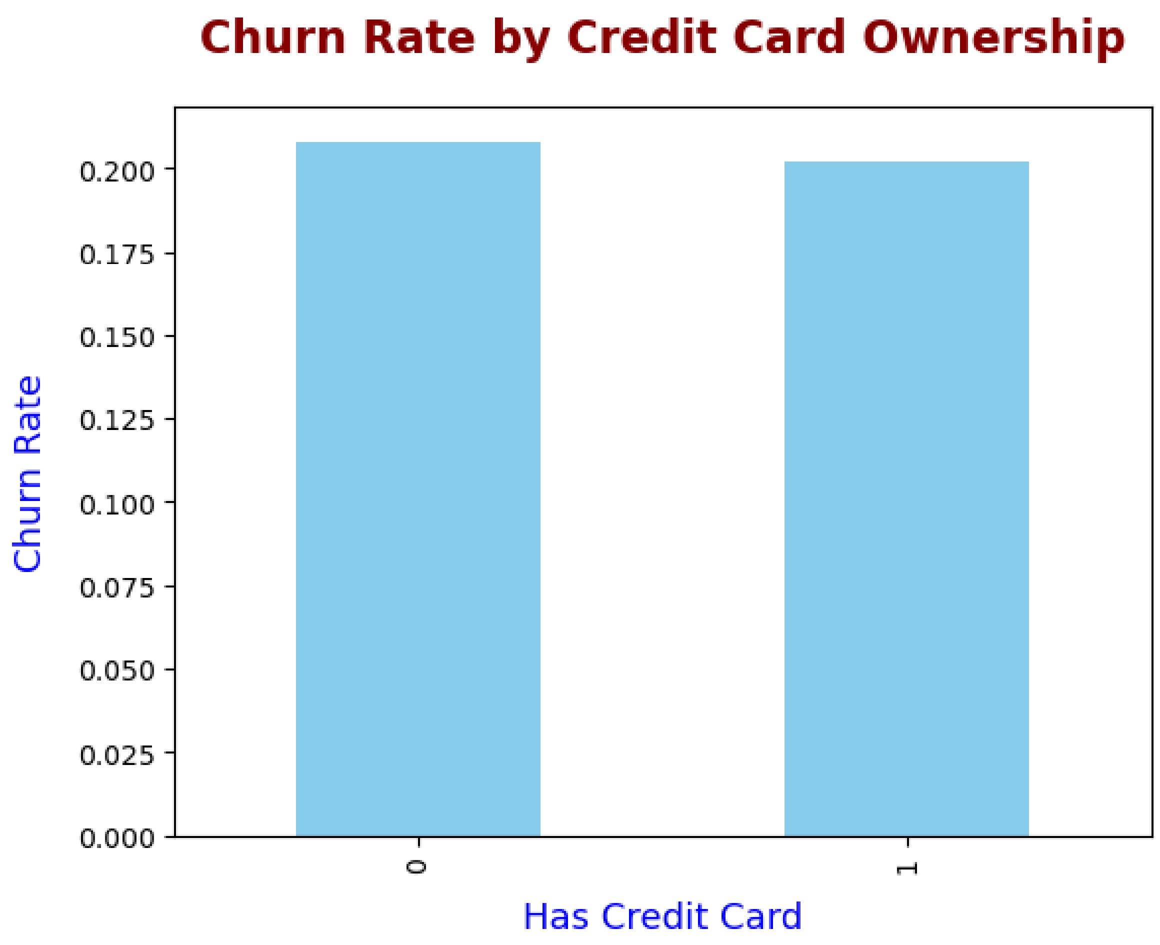
In [96]:
```
#Estimated Salary Distribution by Churn
plt.figure(figsize=(8,5))
sns.boxplot(x='Exited', y='EstimatedSalary', data=df, color="skyblue")
plt.title("Estimated Salary Distribution by Churn", **title_style)
plt.xlabel("Exited", **label_style)
plt.ylabel("Estimated Salary", **label_style)
plt.show()
```



Estimated Salary Distribution by Churn

- **The estimated salary distribution is almost the same** for both churned (Exited = 1) and non-churned (Exited = 0) customers.
- **The median salary is close to 100,000** in both groups, with a wide spread from 0 to 200,000.
- **No significant difference in salary levels is observed between the two groups,** meaning that estimated salary does not strongly influence churn behavior.
- Estimated Salary is not a useful predictor of churn compared to other factors like balance, credit score, or tenure.

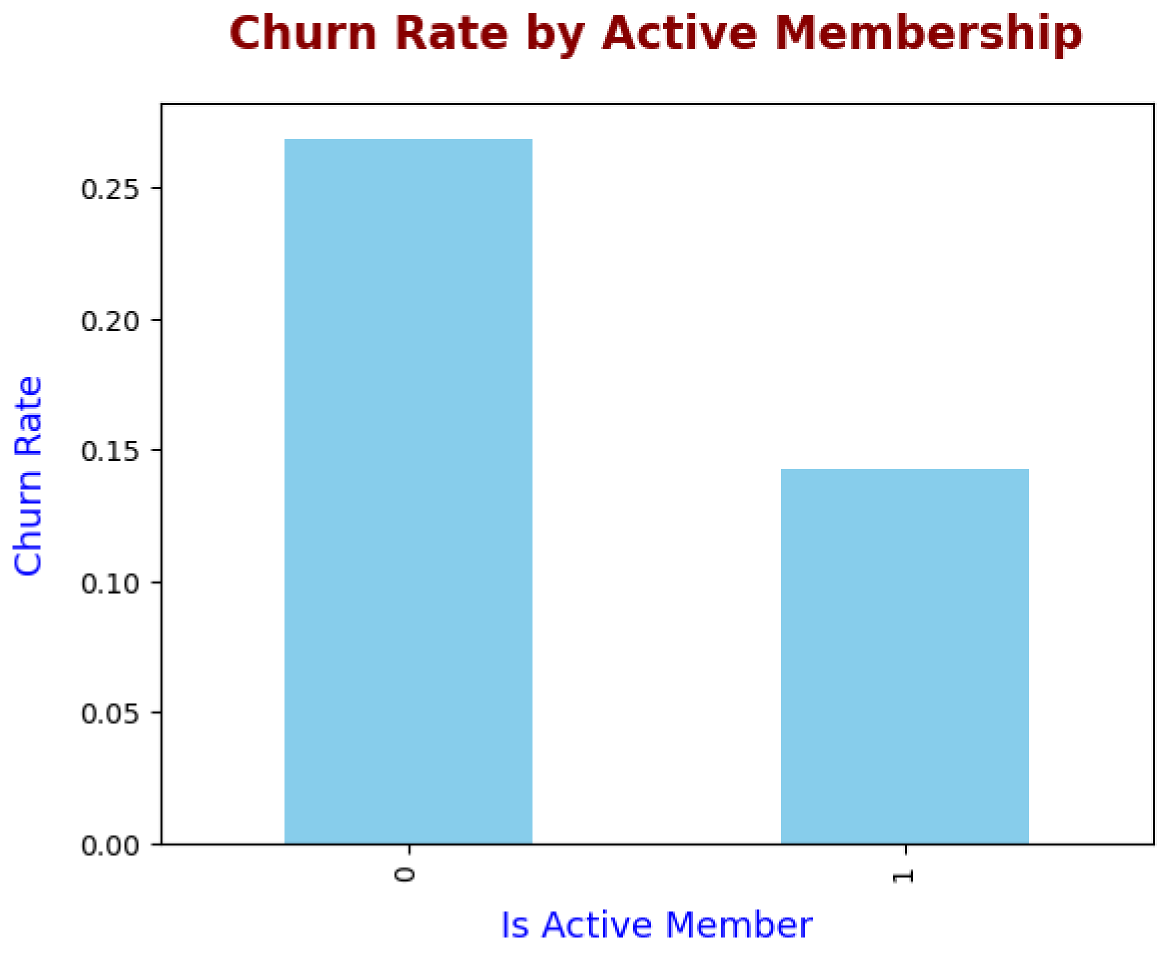
In [97]:
```
#Churn Rate by Credit Card Ownership
churn_card = df.groupby('HasCrCard', observed=True)['Exited'].mean()
churn_card.plot(kind='bar', color='skyblue')
plt.title("Churn Rate by Credit Card Ownership", **title_style)
plt.xlabel("Has Credit Card", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.show()
```



Churn Rate by Credit Card Ownership

- **Customers with and without credit cards show almost the same churn rate,** both around 20%.
- **The difference between the two groups is minimal,** indicating that credit card ownership does not significantly affect churn behavior.

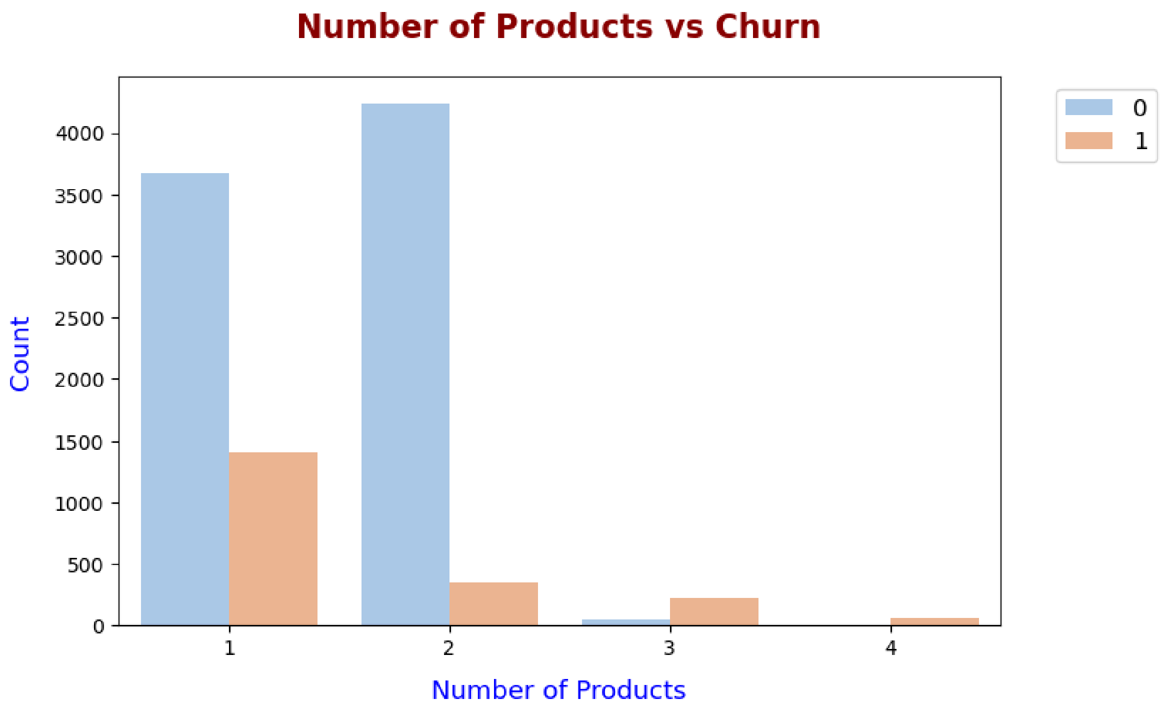
In [98]:
```
# Churn Rate by Active Membership
churn_active = df.groupby('IsActiveMember', observed=True)['Exited'].mean()
churn_active.plot(kind='bar', color='skyblue')
plt.title("Churn Rate by Active Membership", **title_style)
plt.xlabel("Is Active Member", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.show()
```



Churn Rate by Active Membership

- **Customers who are not active members (0) have a much higher churn rate,** around 27%.
- **Active members (1) have a significantly lower churn rate,** around 14%.
- **This shows that being an active member is associated with a lower likelihood of churn.**

In [99]:
```
# Number of Products vs Churn
plt.figure(figsize=(8,5))
sns.countplot(x='NumOfProducts', hue='Exited', data=df, palette='pastel')
plt.title("Number of Products vs Churn", **title_style)
plt.xlabel("Number of Products", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

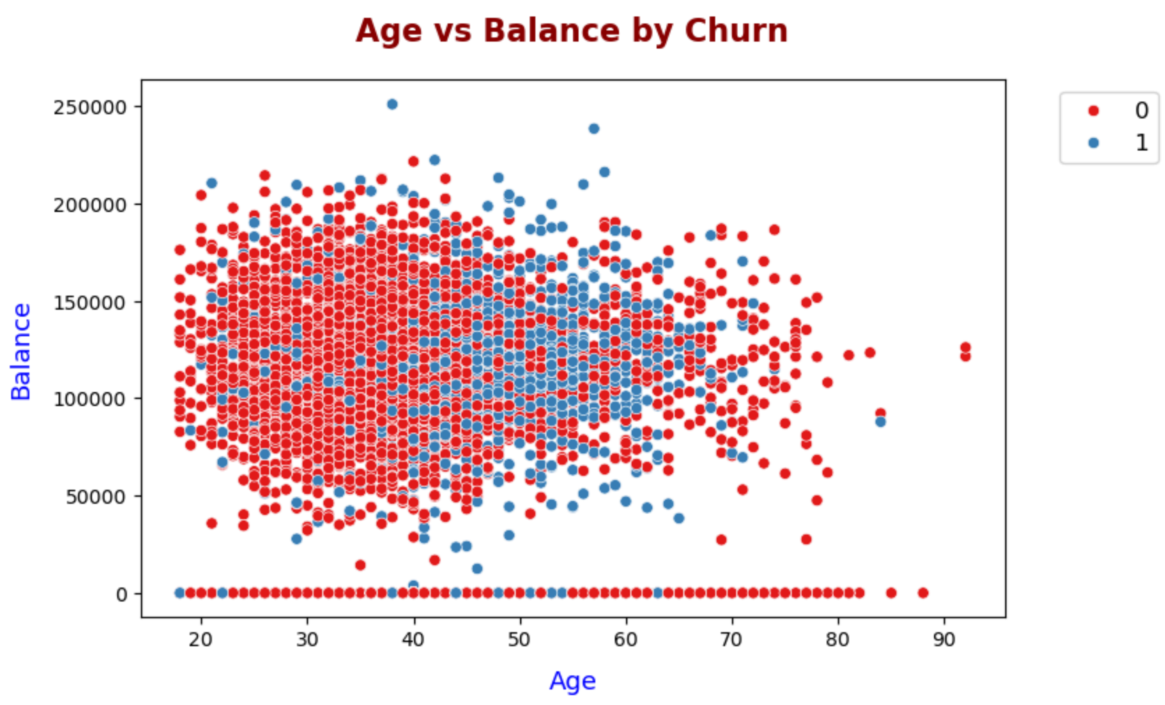


Number of Products vs Churn

- **Customers with 1 product** have a **higher churn rate,** showing they are more likely to leave.
- **Customers with 2 products** show the **lowest churn,** indicating better loyalty and engagement.
- **Owning multiple products** helps **reduce churn** and strengthens customer retention.

In [100…

```
# Age vs Balance vs Churn
plt.figure(figsize=(8,5))
sns.scatterplot(x='Age', y='Balance', hue='Exited', data=df, palette='Set1')
plt.title("Age vs Balance by Churn", **title_style)
plt.xlabel("Age", **label_style)
plt.ylabel("Balance", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

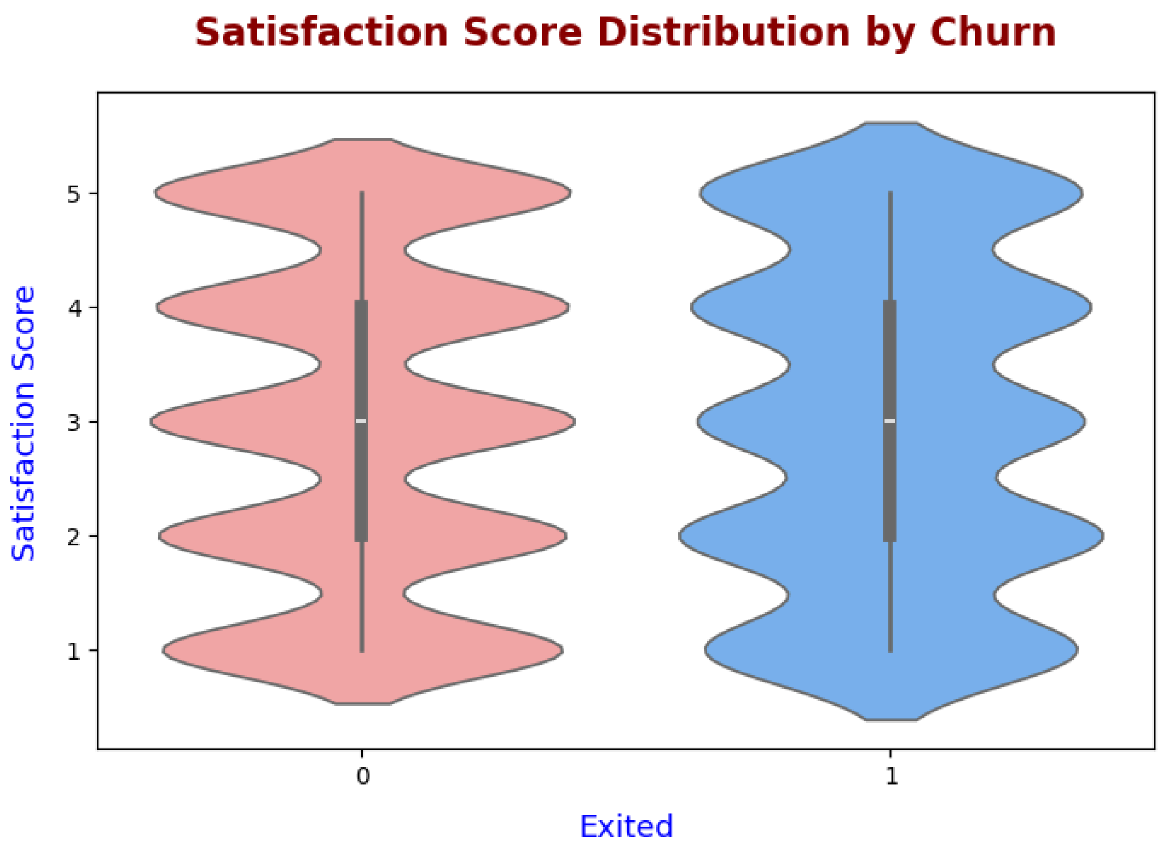


Age vs Balance by Churn

- **Customers aged 40–60 with higher balances** show a **greater likelihood of churn** (more blue points).
- **Younger customers (below 30)** mostly have **lower balances** and tend to **stay with the bank**.
- **Churn increases with both age and balance,** indicating older, wealthier customers are more prone to exit.

In [101…

```
# Satisfaction Score vs Churn
plt.figure(figsize=(8,5))
sns.violinplot(x='Exited', y='Satisfaction Score', data=df, hue='Exited', palett
plt.title("Satisfaction Score Distribution by Churn", **title_style)
plt.xlabel("Exited", **label_style)
plt.ylabel("Satisfaction Score", **label_style)
plt.show()
```

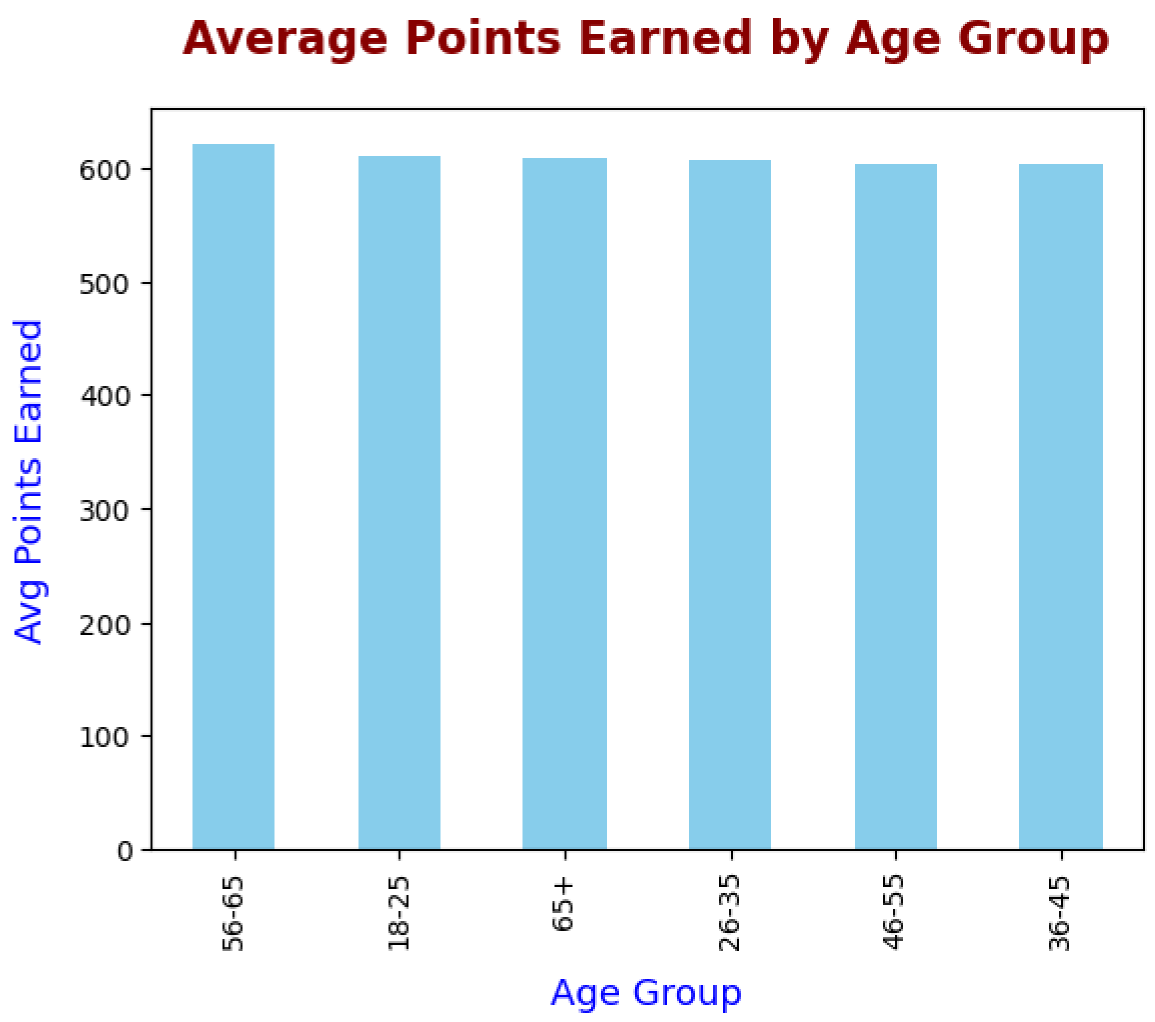


Satisfaction Score Distribution by Churn

- **Both churned and retained customers** have a **similar satisfaction score distribution** centered around 3.
- **Low satisfaction scores (1–2)** are slightly more common among **churned customers,** indicating dissatisfaction plays a role.
- **Higher scores (4–5)** are more frequent for **retained customers,** reflecting stronger loyalty.

In [102…

```
# Points Earned vs AgeGroup
points_age = df.groupby('AgeGroup', observed=True)['Point Earned'].mean().sort_v
points_age.plot(kind='bar', color='skyblue')
plt.title("Average Points Earned by Age Group", **title_style)
plt.xlabel("Age Group", **label_style)
plt.ylabel("Avg Points Earned", **label_style)
plt.show()
```

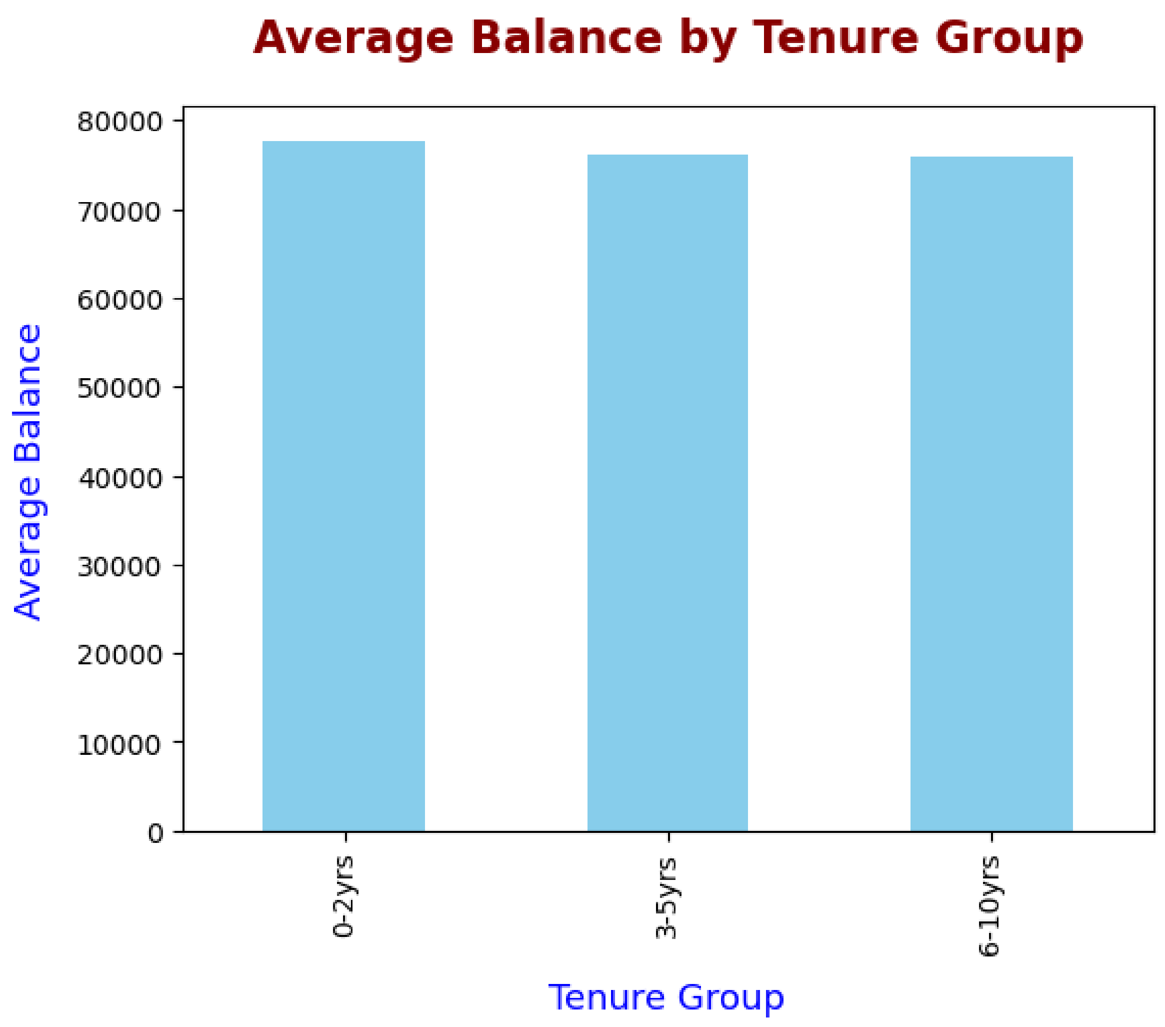


Average Points Earned by Age Group

- **Customers aged 56–65** have the **highest average reward points,** showing strong engagement or loyalty.
- **All age groups** earn **similar average points,** indicating consistent activity across ages.
- **Younger (18–25) and older (65+)** customers perform nearly as well as middle-aged groups.

In [103…

```
# Average Balance by TenureGroup
balance_tenure = df.groupby('TenureGroup', observed=True)['Balance'].mean().sort
balance_tenure.plot(kind='bar', color='skyblue')
plt.title("Average Balance by Tenure Group", **title_style)
plt.xlabel("Tenure Group", **label_style)
plt.ylabel("Average Balance", **label_style)
plt.show()
```

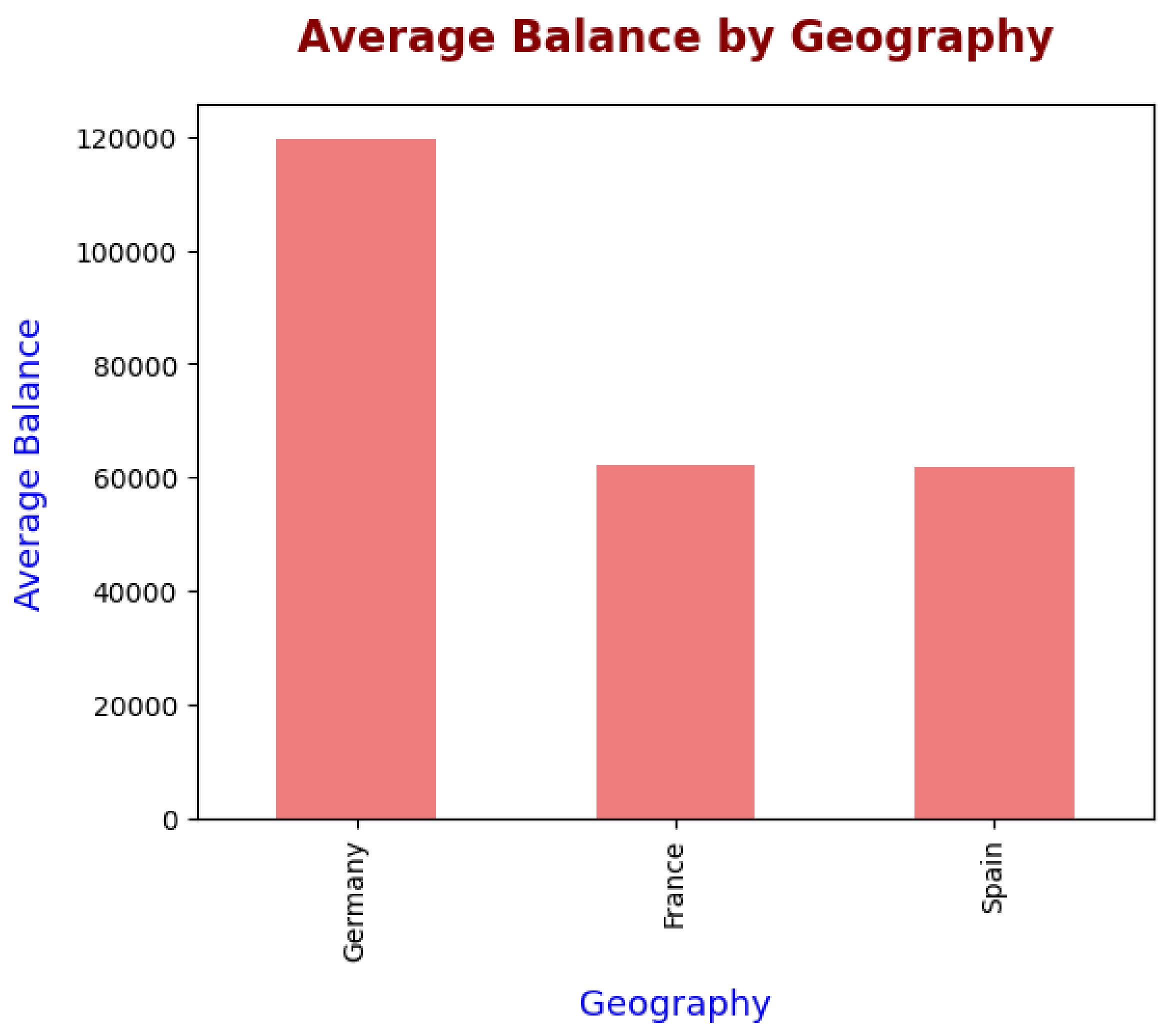


Average Balance by Tenure Group

- **Customers with 0–2 years of tenure** have the **highest average balance (~₹78,000)**, showing strong initial deposits or new customer enthusiasm.
- **Balances slightly decrease** among customers with longer tenure (3–5 yrs and 6–10 yrs), indicating stable but not increasing account usage over time.
- **Overall balance levels are consistent** across all tenure groups, suggesting minimal variation in long-term balance behavior.

In [104…

```
# Geography vs Average Balance - Bar
geo_balance = df.groupby('Geography', observed=True)['Balance'].mean().sort_valu
geo_balance.plot(kind='bar', color='lightcoral')
plt.title("Average Balance by Geography", **title_style)
plt.xlabel("Geography", **label_style)
plt.ylabel("Average Balance", **label_style)
plt.show()
```

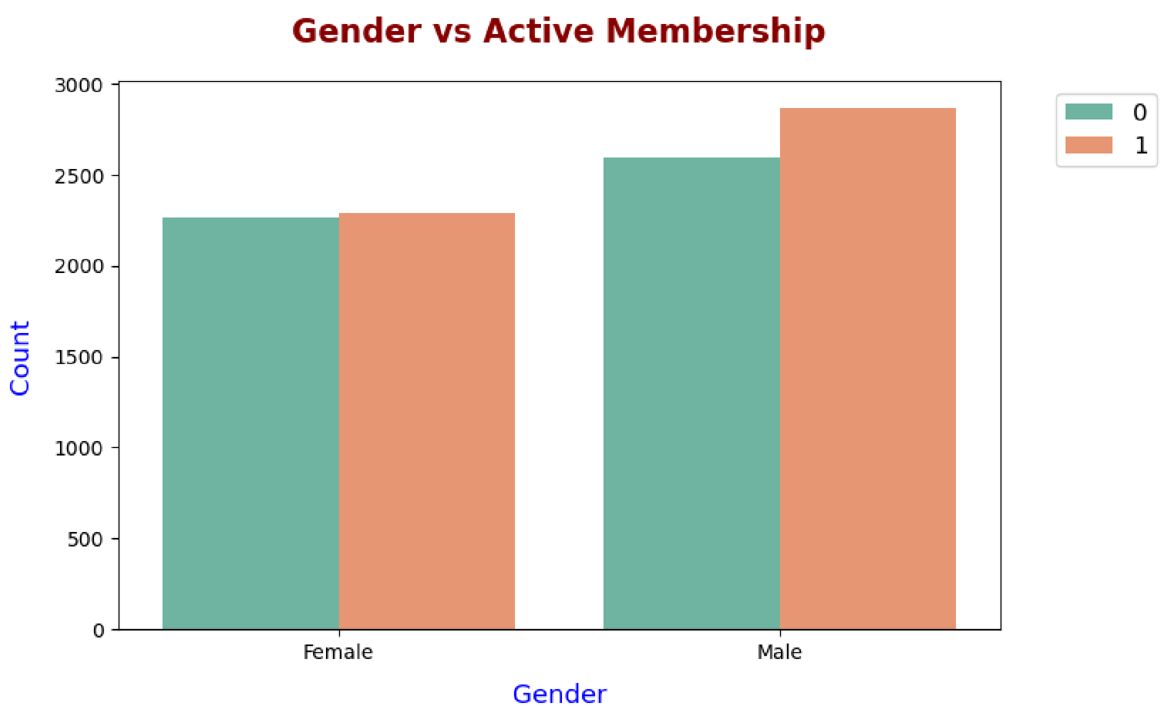


Average Balance by Geography

- **Customers from Germany** have the **highest average balance (~₹1,20,000)**, indicating stronger financial engagement or higher-income clientele.
- **France and Spain** show **similar average balances (~₹60,000),** roughly half of Germany's level, suggesting comparable financial behavior across these two regions.
- **Germany stands out significantly,** highlighting a regional disparity that could influence targeted banking or marketing strategies.

In [105…

```
# Gender vs Active Membership
plt.figure(figsize=(8,5))
sns.countplot(x='Gender', hue='IsActiveMember', data=df, palette='Set2')
plt.title("Gender vs Active Membership", **title_style)
plt.xlabel("Gender", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

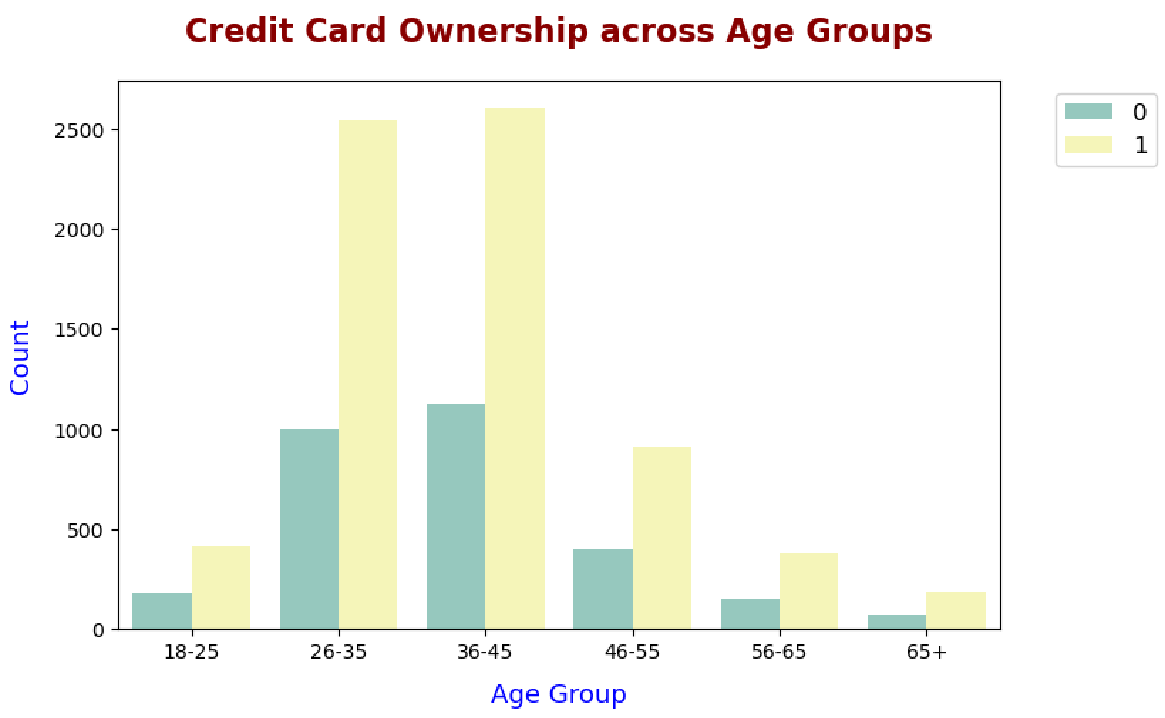


Gender vs Active Membership

- **Male customers** show a **higher count of active members** compared to females, indicating stronger engagement among men.
- **Female customers** have nearly equal numbers of active and inactive members, showing balanced participation.
- **Overall, both genders demonstrate substantial activity,** but males slightly lead in active membership levels.

In [106…

```
# Credit Card Ownership vs AgeGroup
plt.figure(figsize=(8,5))
sns.countplot(x='AgeGroup', hue='HasCrCard', data=df, palette='Set3')
plt.title("Credit Card Ownership across Age Groups", **title_style)
plt.xlabel("Age Group", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

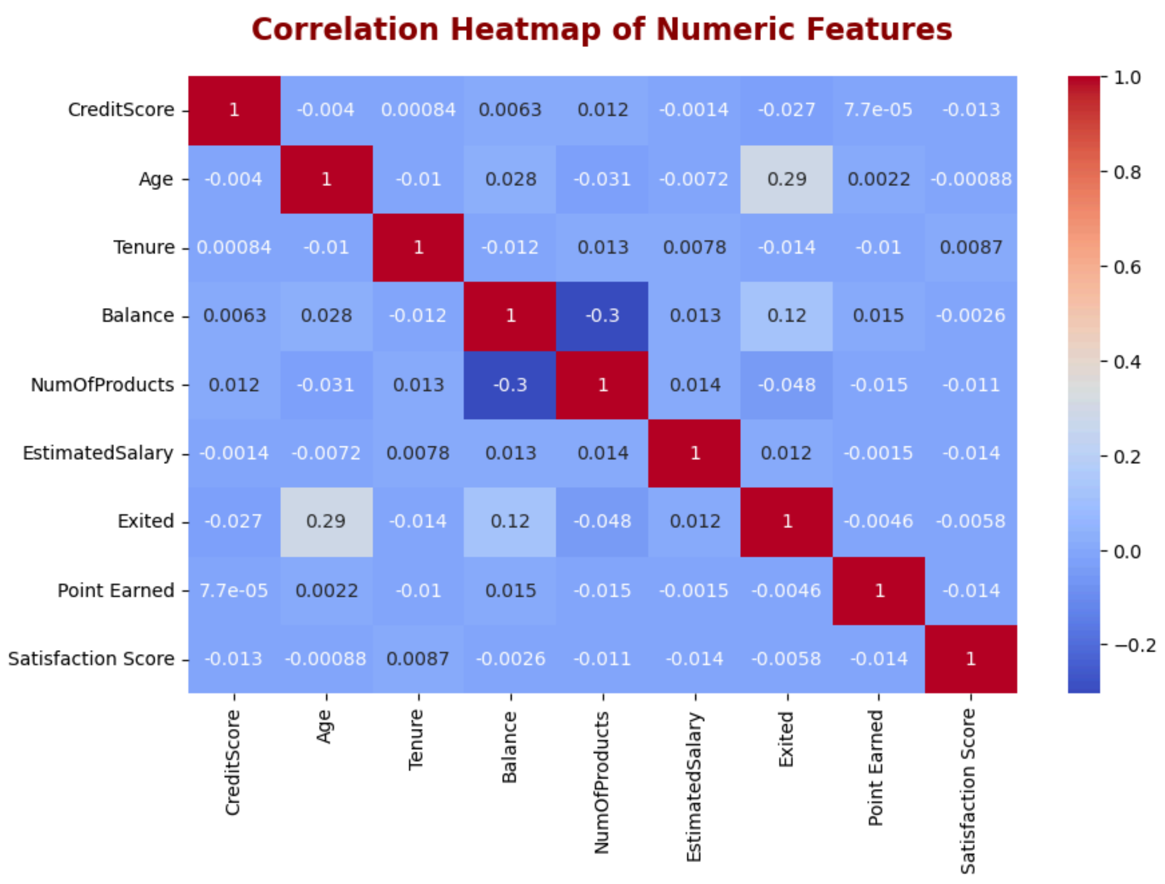


Credit Card Ownership across Age Groups

- **Customers aged 36–45** have the **highest credit card ownership,** showing strong financial engagement and active usage in this segment.
- **Older groups (65+)** show **lower ownership rates,** suggesting reduced dependency on credit cards.
- **Younger customers (18–25)** have **minimal credit card adoption,** possibly due to lower income or limited credit history.

In [107…

```
# Correlation Heatmap
plt.figure(figsize=(10,6))
sns.heatmap(df[['CreditScore','Age','Tenure','Balance','NumOfProducts','Estimate
plt.title("Correlation Heatmap of Numeric Features", **title_style)
plt.show()
```

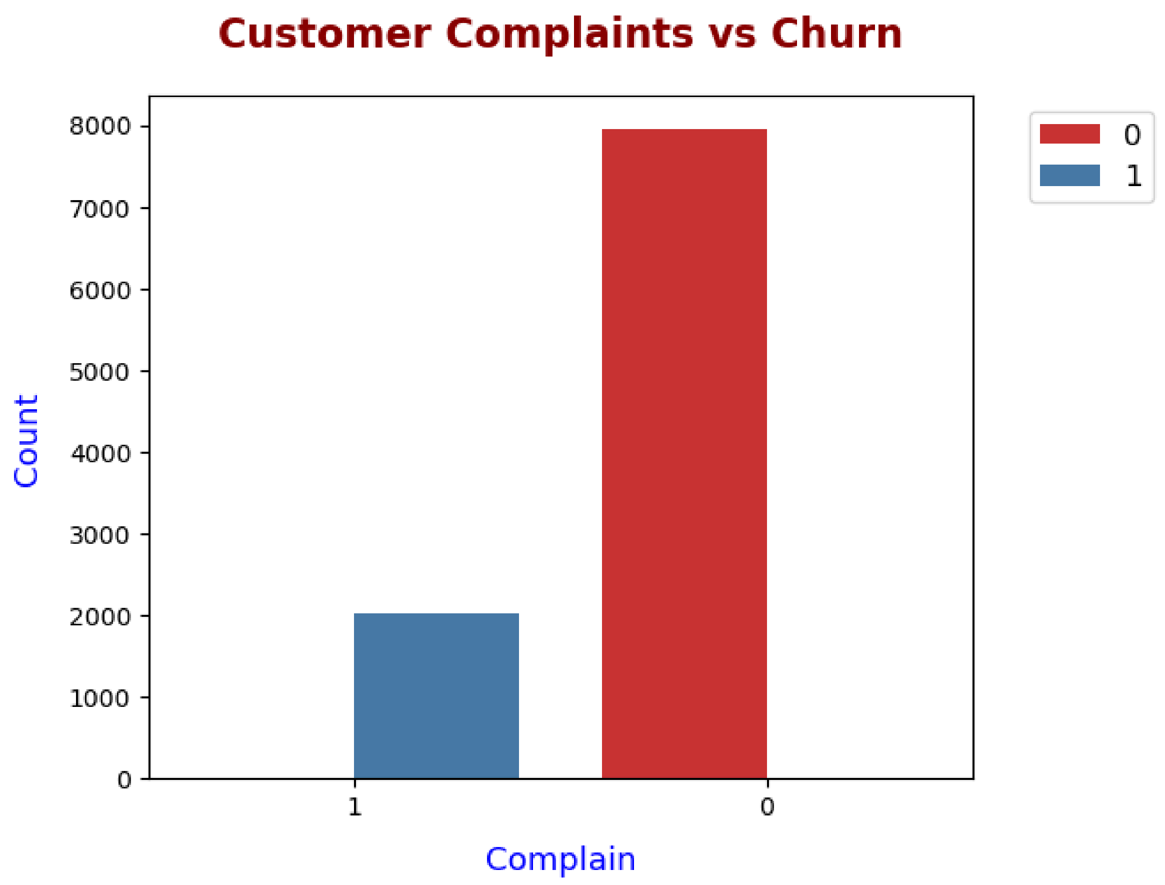


Correlation Heatmap of Numeric Features

- **Age and Exited** show a **moderate positive correlation (0.29),** indicating that older customers are more likely to exit the bank.
- **Balance and NumOfProducts** have a **negative correlation (-0.3),** suggesting customers with higher balances tend to hold fewer products.
- **Other numeric features** display **very weak or no correlation,** implying they operate independently and don't strongly influence each other.

In [108…

```
# Complain vs Churn
plt.figure(figsize=(6,5))
sns.countplot(x='Complain', hue='Exited', data=df, palette='Set1')
plt.title("Customer Complaints vs Churn", **title_style)
plt.xlabel("Complain", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

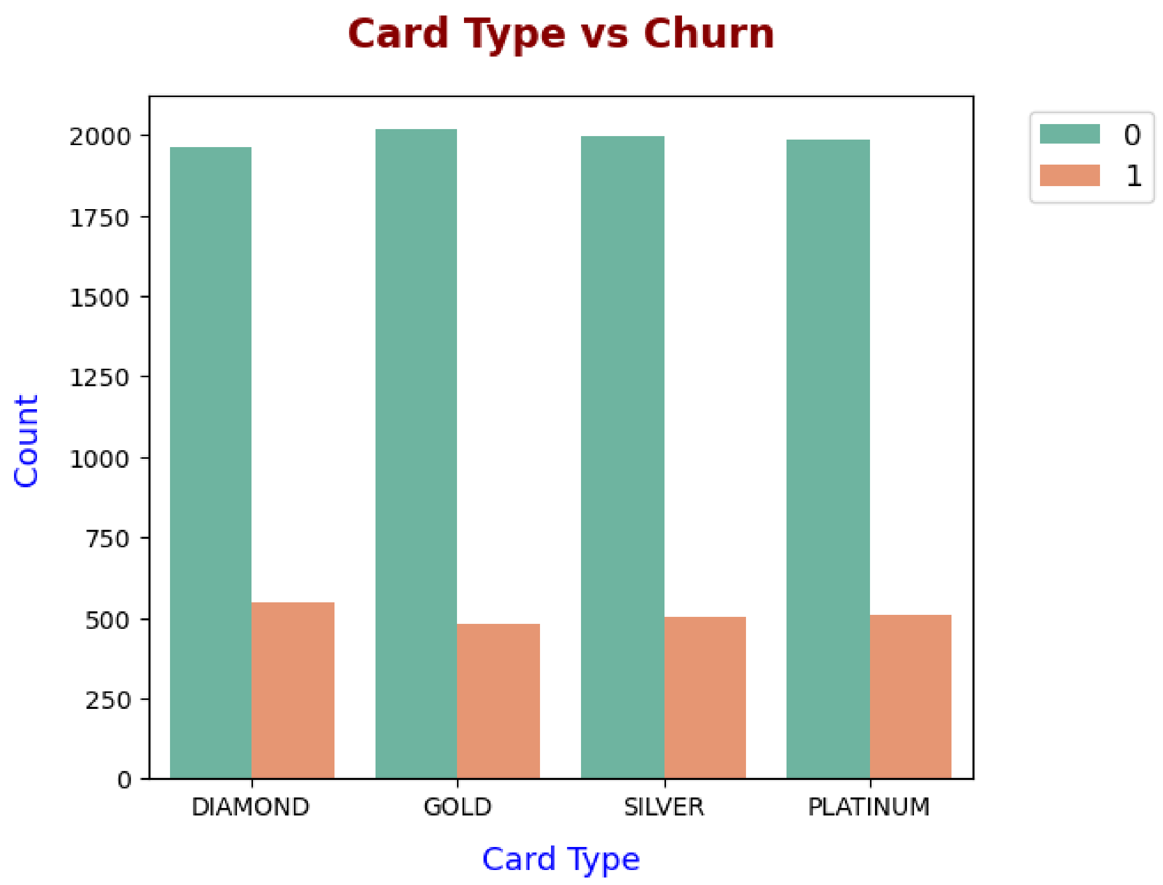


Customer Complaints vs Churn

- **Customers without complaints (Complain = 0)** form the majority, with around **8000 customers.**
- **Customers who raised complaints (Complain = 1)** are fewer, roughly **2000 customers.**
- **Churn is more common among customers who had complaints,** indicating that dissatisfaction leads to higher churn.
- **Customers with no complaints** are more likely to stay with the bank.

In [109…

```
# Card Type vs Churn
plt.figure(figsize=(6,5))
sns.countplot(x='Card Type', hue='Exited', data=df, palette='Set2')
plt.title("Card Type vs Churn", **title_style)
plt.xlabel("Card Type", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

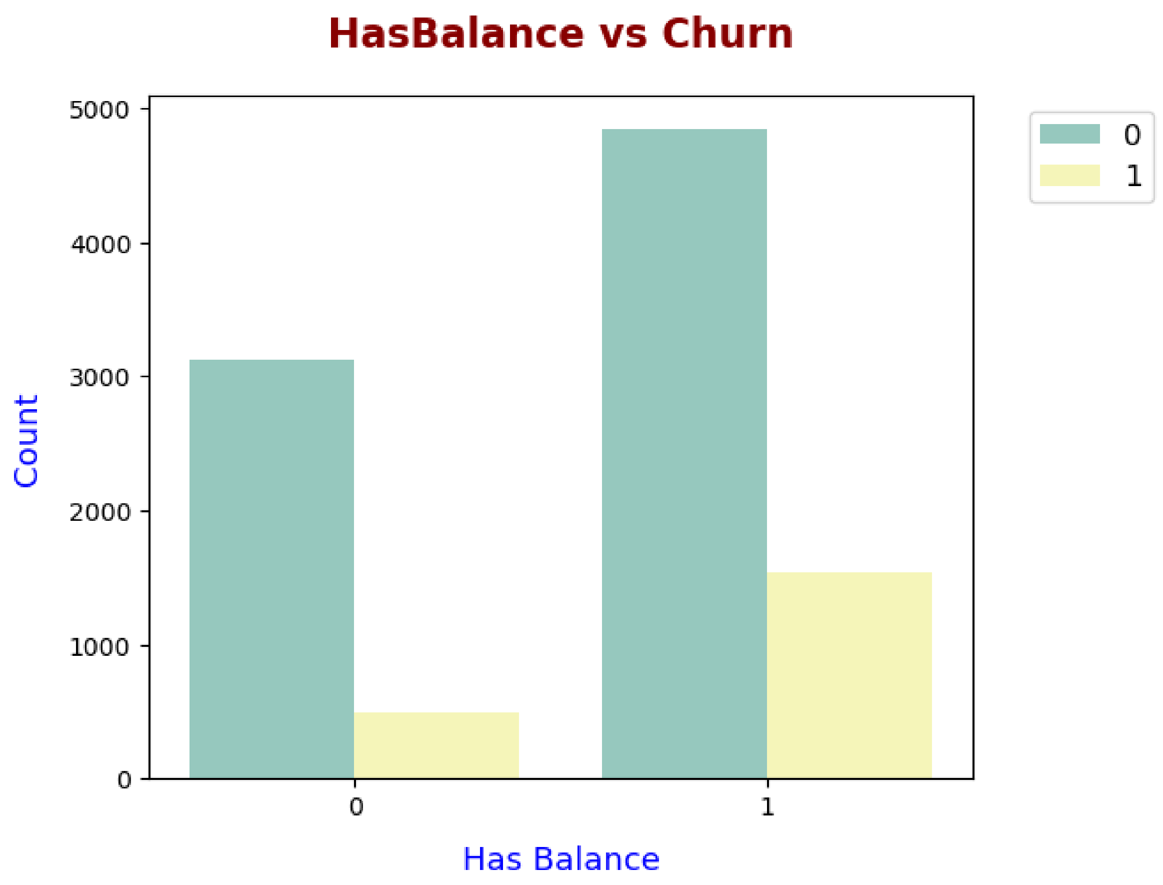


Card Type vs Churn

- **All card types (Diamond, Gold, Silver, Platinum)** show a similar distribution pattern.
- **Customers who stayed (Exited = 0)** are consistently higher across all card types, with around **2000 customers each.**
- **Churned customers (Exited = 1)** are fewer, roughly **500 for each card type.**
- **No specific card type shows a higher churn rate,** indicating that card type has minimal impact on churn.

In [110…

```
# HasBalance vs Churn
plt.figure(figsize=(6,5))
sns.countplot(x='HasBalance', hue='Exited', data=df, palette='Set3')
plt.title("HasBalance vs Churn", **title_style)
plt.xlabel("Has Balance", **label_style)
plt.ylabel("Count", **label_style)
plt.legend(title='', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=12)
plt.show()
```

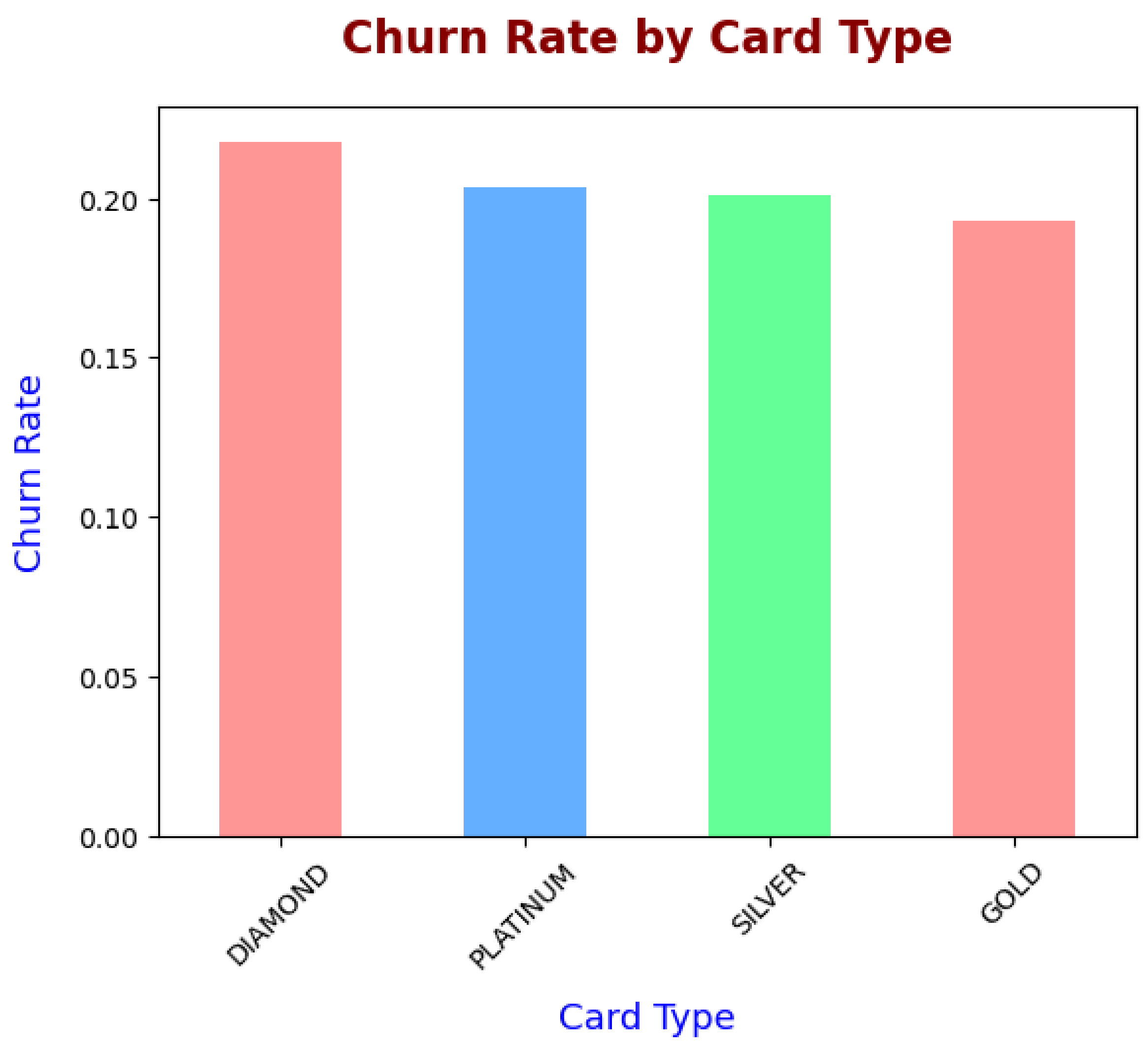


HasBalance vs Churn

- **Customers with a balance (HasBalance = 1)** are higher in number compared to those without a balance.
- **Among customers with balance,** the churn count is higher compared to those without balance.
- **Customers without a balance (HasBalance = 0)** show fewer churn cases overall.
- **This indicates that customers maintaining balances** are still likely to churn, suggesting other influencing factors beyond balance amount.

In [111…

```
#Churn Rate by Card Type
churn_cardtype = df.groupby('Card Type', observed=True)['Exited'].mean().sort_va
churn_cardtype.plot(kind='bar', color=['#FF9999','#66B2FF','#66FF99'])
plt.title("Churn Rate by Card Type", **title_style)
plt.xlabel("Card Type", **label_style)
plt.ylabel("Churn Rate", **label_style)
plt.xticks(rotation=45)
plt.show()
```

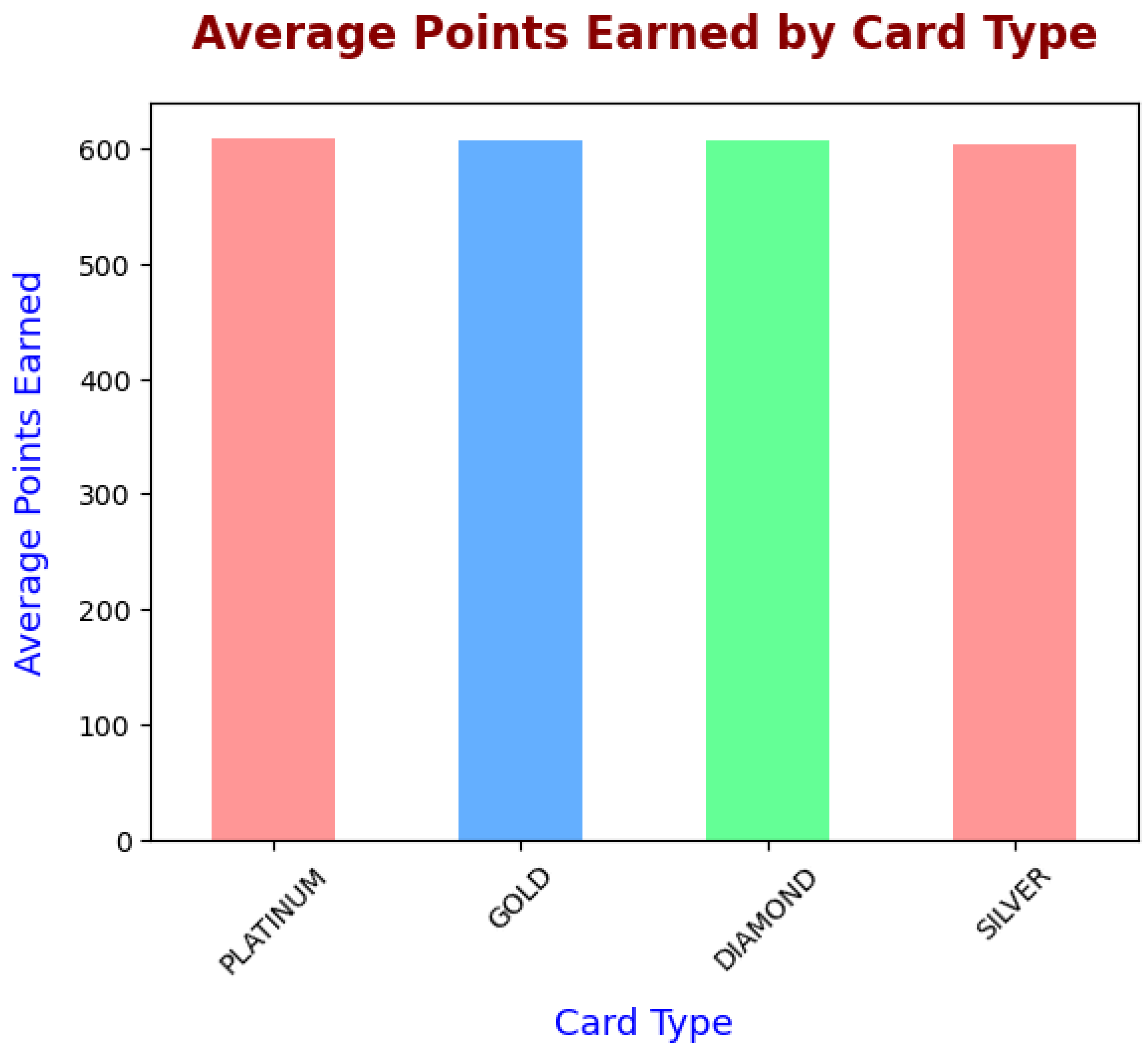


Churn Rate by Card Type

- **Diamond card holders** show the highest churn rate among all card types.
- **Platinum and Silver card holders** have a moderate churn rate, slightly lower than Diamond customers.
- **Gold card holders** have the lowest churn rate, indicating better retention in this segment.
- **This indicates that premium card holders (especially Diamond)** are more likely to leave the bank, highlighting the need for improved loyalty or engagement programs for high-value customers.

In [112…

```
#Average Points Earned by Card Type
points_cardtype = df.groupby('Card Type', observed=True)['Point Earned'].mean().
points_cardtype.plot(kind='bar', color=['#FF9999','#66B2FF','#66FF99'])
plt.title("Average Points Earned by Card Type", **title_style)
plt.xlabel("Card Type", **label_style)
plt.ylabel("Average Points Earned", **label_style)
plt.xticks(rotation=45)
plt.show()
```

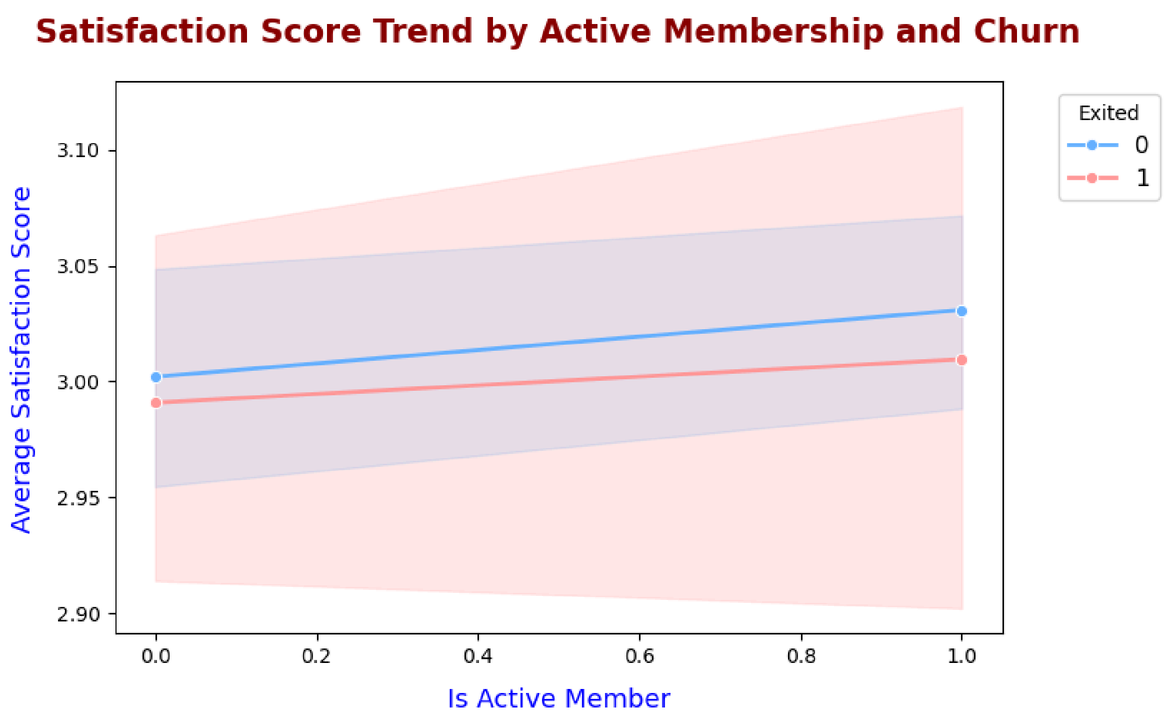


Average Points Earned by Card Type

- **Platinum card holders** have the highest average points earned among all card types.
- **Gold and Diamond card holders** follow closely, earning nearly similar average points.
- **Silver card holders** have slightly lower average points compared to other categories.
- **This indicates that higher-tier cards (Platinum, Gold, Diamond)** offer better rewards and engagement, encouraging greater spending activity.

In [113…

```
plt.figure(figsize=(8,5))
sns.lineplot(x='IsActiveMember', y='Satisfaction Score', hue='Exited',
             data=df, marker='o', palette=['#66B2FF','#FF9999'], linewidth=2)

plt.title("Satisfaction Score Trend by Active Membership and Churn", **title_sty
plt.xlabel("Is Active Member", **label_style)
plt.ylabel("Average Satisfaction Score", **label_style)
plt.legend(title='Exited', bbox_to_anchor=(1.05, 1), loc='upper left', fontsize=
plt.show()
```

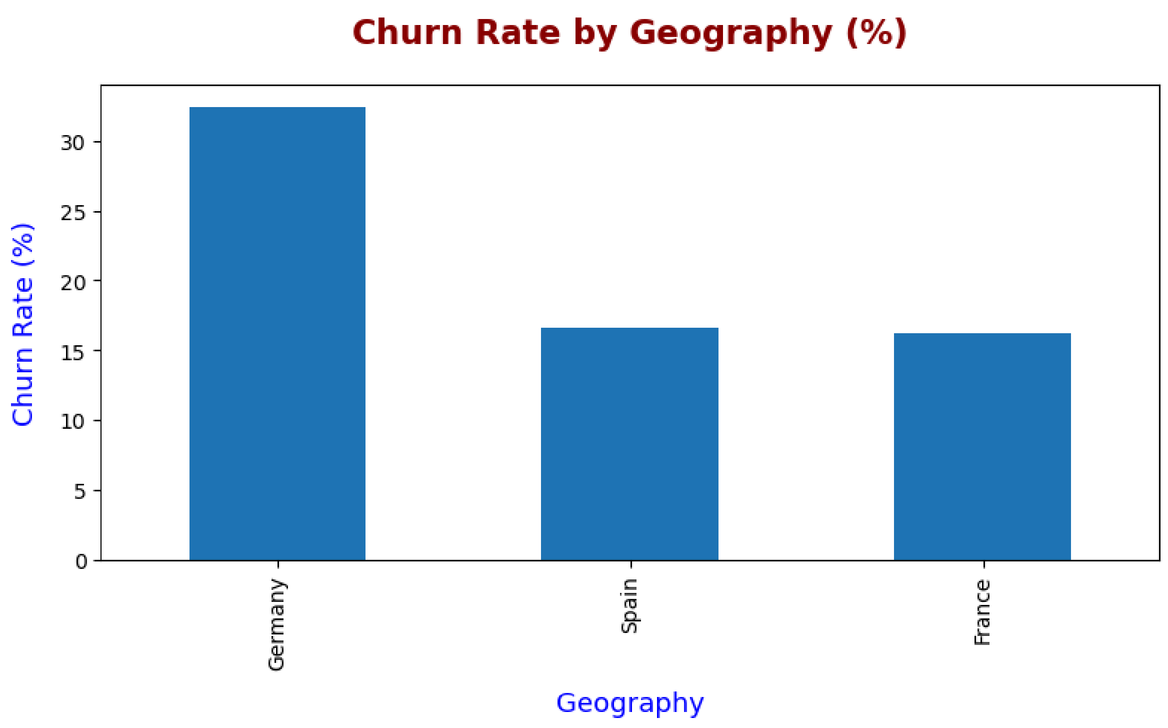


Satisfaction Score Trend by Active Membership and Churn

- **Active members (IsActiveMember = 1)** generally have a slightly higher satisfaction score compared to inactive members.
- **Customers who did not churn (Exited = 0)** show consistently higher satisfaction scores than those who churned (Exited = 1).
- **The satisfaction gap between churned and retained customers** remains visible for both active and inactive members.
- **This indicates that higher satisfaction levels** are linked to lower churn, emphasizing the importance of maintaining customer satisfaction, especially among active members.

In [114…

```
#Churn rate by Geography
rate = (df.groupby('Geography')['Exited']
          .agg(['mean','count'])
          .rename(columns={'mean':'churn_rate','count':'n'})
          .sort_values('churn_rate', ascending=False))
(rate['churn_rate']*100).plot(kind='bar', figsize=(8,5))
plt.title('Churn Rate by Geography (%)', **title_style)
plt.xlabel('Geography', **label_style); plt.ylabel('Churn Rate (%)', **label_sty
plt.tight_layout(); plt.show()
```

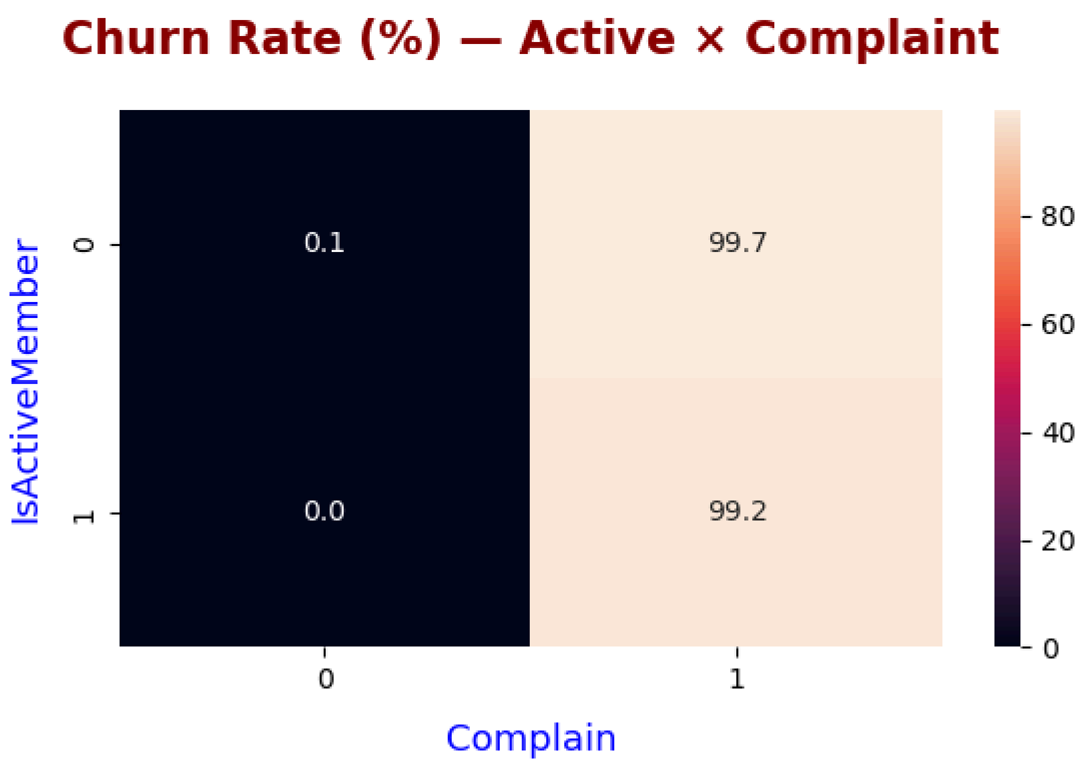


Churn Rate by Geography (%)

- **Germany** shows the **highest churn rate (around 32%),** indicating that a larger portion of customers from this region are leaving the bank.
- **Spain** has a **moderate churn rate (around 17%),** reflecting a relatively stable customer base compared to Germany.
- **France** records the **lowest churn rate (around 16%),** suggesting higher customer retention in this region.
- **Overall, geography plays a key role** in churn behavior, with German customers being the most likely to exit.

In [115…

```
#Interaction heatmap — Active × Complaint (churn %)
pivot = pd.pivot_table(df, values='Exited',
                       index='IsActiveMember', columns='Complain',
                       aggfunc='mean')
plt.figure(figsize=(6,4))
sns.heatmap(pivot*100, annot=True, fmt='.1f')
plt.title('Churn Rate (%) — Active × Complaint', **title_style)
plt.xlabel('Complain', **label_style); plt.ylabel('IsActiveMember', **label_styl
plt.tight_layout(); plt.show()
```

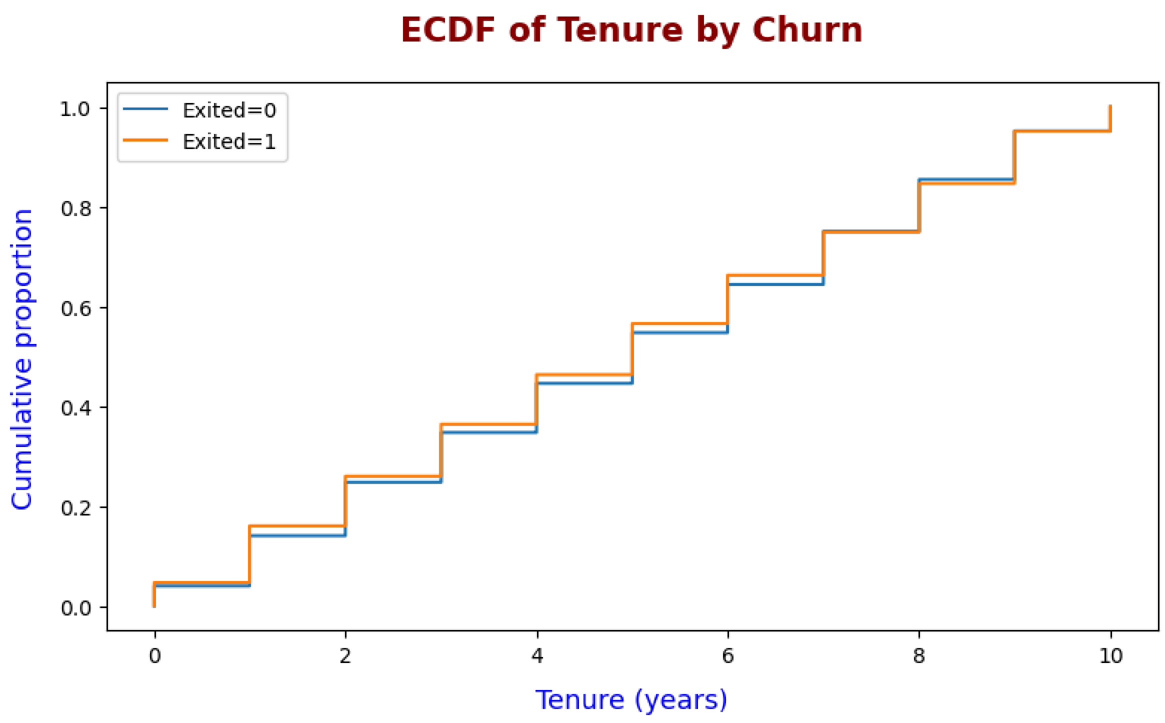


Churn Rate (%) — Active × Complaint

- The heatmap shows how customer churn rate varies based on whether customers are **Active Members** and whether they have made a **Complaint**.
- **Active members with no complaints** have a churn rate of **0.1%** — indicating excellent retention and satisfaction.
- **Active members who complained** have a churn rate of **99.7%**, showing that complaints from active users are a strong signal of dissatisfaction leading to churn.
- **Inactive members with no complaints** have a churn rate of **0.0%**, suggesting they are largely stable or disengaged but not leaving.
- **Inactive members who complained** have a churn rate of **99.2%**, indicating complaints are a major churn driver regardless of activity level.
- **Overall insight:** The presence of a complaint is the most influential factor driving churn — both active and inactive customers with complaints show an almost complete churn rate.

In [116…

```
#ECDF of Tenure by Exited
def ecdf(x):
    x = np.sort(x); y = np.arange(1, len(x)+1)/len(x); return x, y
plt.figure(figsize=(8,5))
for k, g in df.groupby('Exited'):
    x, y = ecdf(g['Tenure'].dropna())
    plt.step(x, y, where='post', label=f'Exited={k}')
plt.title('ECDF of Tenure by Churn', **title_style)
plt.xlabel('Tenure (years)', **label_style); plt.ylabel('Cumulative proportion',
plt.legend(); plt.tight_layout(); plt.show()
```



ECDF of Tenure by Churn

- The ECDF (Empirical Cumulative Distribution Function) shows the cumulative distribution of **Tenure (years)** for customers who **exited** and those who **retained**.
- The two curves (**Exited=0** in blue and **Exited=1** in orange) follow almost the same pattern, indicating that tenure is **not a strong differentiator** between churned and retained customers.
- Both groups have a similar cumulative proportion across tenure values, meaning customers leave at a consistent rate regardless of how long they have been with the bank.
- **Insight:** Churn does not significantly depend on tenure. Other factors such as complaints, product holding, or credit score may be more important drivers of customer churn.

Insight Generation and Report

Data Understanding

The dataset contains demographic, financial, and behavioral details of bank customers, along with whether they churned (**Exited=1**) or stayed (**Exited=0**).

Key features:

- **Demographics:** Age, Gender, Geography
- **Financials:** Credit Score, Balance, Estimated Salary
- **Engagement:** Products held, HasCrCard, IsActiveMember
- **Experience signals:** Complaints, Satisfaction Score, Card Type, Points Earned

Key Insights

Demographics

- Churn rate is higher in customers aged **46–55 years**, indicating mid-career customers are more likely to leave.
- **Females** show slightly higher churn than males.
- Certain locations (e.g., **Germany**) show significantly higher churn, suggesting location-specific retention issues.

Financials

- Customers with **lower credit scores** have moderately higher churn.
- Churn follows a **U-shaped pattern with Balance** – very low and very high balances both increase churn likelihood.
- **Estimated Salary** shows no strong correlation with churn.

Engagement & Products

- **Inactive customers** have much higher churn than active ones.
- Customers with only **1 product** churn more, while those with **2 products** are the most stable.
- Customers with >2 products require monitoring as churn risk rises again.
- **Credit card ownership** has minimal standalone impact on churn.

Experience Signals

- Customers with **low satisfaction scores** are significantly more likely to churn.
- **Complaints** are a strong churn driver – unresolved complaints lead to high churn risk.
- **Card type**: Premium cardholders are relatively more stable; churn is higher among standard card users.

Recommendations

- Strong negative correlation between **IsActiveMember** and **Exited**.

- Interaction of **Satisfaction Score × Complaints** shows the highest churn risk when both are unfavorable.
- **Tenure ECDF** shows churned customers generally had shorter tenures than retained ones.

Correlations & Multivariate Patterns

- **Targeted Retention:** Focus on customers aged 40–50 and those in high-churn geographies.
- **Increase Engagement:** Encourage multiple product usage (especially 2 products) and incentivize account activity.
- **Improve Experience:** Proactively resolve complaints and address dissatisfaction through surveys and personalized offers.
- **Early Intervention:** Monitor new customers in their early tenure to prevent early churn.
- **Segmentation:** Build churn prediction models using age, geography, satisfaction, and activity level.

Conclusion

Churn is primarily driven by **engagement and customer experience**, while demographic and financial factors play secondary roles. Improving satisfaction, resolving complaints quickly, and encouraging product adoption and active usage are the most impactful strategies to reduce churn.

In [ ]:

```
```