Bank customer churn analysis: Loosing Bank customers

Business Problem Statement: ¶

- In the rapidly evolving banking sector, customer retention has become a critical concern.
- Banks are increasingly seeking to understand the factors that influence customer decisions to stay with or leave their banking service provider.
- This project focuses on analyzing a dataset containing various attributes of bank customers to identify **key predictors of customer churn.**
- By leveraging data analytics, we aim to uncover patterns and insights that could help devise strategies to enhance customer retention and reduce churn rates.

Data description:

Dataset link:https://drive.google.com/file/d/1xh7D0NDmxdg6IXTFzi_T-Oc5D-GtI44W/view?usp=sharing (https://drive.google.com/file/d/1xh7D0NDmxdg6IXTFzi_T-Oc5D-GtI44W/view?usp=sharing)

- **RowNumber**—corresponds to the record (row) number and has no effect on the output.
- **CustomerId**—contains random values and has no effect on customer leaving the bank.
- **Surname**—the surname of a customer has no impact on their decision to leave the bank.
- **CreditScore**—can have an effect on customer churn, since a customer with a higher credit score is less likely to leave the bank.
- **Geography**—a customer's location can affect their decision to leave the bank.
- **Gender**—it's interesting to explore whether gender plays a role in a customer leaving the bank.
- **Age**—this is certainly relevant, since older customers are less likely to leave their bank than younger ones.
- **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.
- **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.
- **NumOfProducts**—refers to the number of products that a customer has purchased through the bank.
- **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.
- **IsActiveMember**—active customers are less likely to leave the bank.
- **EstimatedSalary**—as with balance, people with lower salaries are more likely to leave the bank compared to those with higher salaries.
- **Exited**—whether or not the customer left the bank.
- **Complain**—customer has complaint or not.
- **Satisfaction Score**—Score provided by the customer for their complaint resolution.
- **Card Type**—type of card hold by the customer.
- **Points Earned**—the points earned by the customer for using credit card.

Approach and Solution Methodology:

1. **Descriptive Statistics**
   - Basic Statistics: Calculate mean, median, and mode for numerical columns like CreditScore, Age, Balance, NumOfProducts, EstimatedSalary, and Points Earned.
   - Distribution Analysis: Analyze the distribution of key numerical variables using histograms and box plots to understand the spread and central tendency.

2. **Exploratory Data Analysis (EDA)**
   - Correlation Analysis: Explore the correlation between numerical features and the Exited variable to identify potential predictors of churn.
   - Customer Profile Analysis: Segment customers based on key demographics (Age, Geography, Gender) to identify which groups are more likely to churn.

3. **Comparative Analysis**
   - Churn by Geography: Compare churn rates across different geographical locations to see if certain regions have higher churn rates.
   - Gender Differences in Churn: Analyze churn rates between different genders to explore if gender plays a significant role in churn.

4. **Behavioral Analysis**
   - Product and Services Usage: Examine how the number of products (NumOfProducts) a customer uses affects their likelihood to churn.
   - Activity Level Analysis: Investigate the relationship between being an IsActiveMember and customer churn.

5. **Financial Analysis**
   - Balance vs. Churn: Analyze how customer balance levels correlate with churn rates.

- Credit Card Ownership: Determine if owning a credit card (HasCrCard) impacts customer loyalty.

6. **Customer Satisfaction and Feedback**

- Complaint Analysis: Study the impact of having a complaint (Complain) on customer churn.
- Satisfaction and Churn: Explore how the Satisfaction Score relates to churn, especially among those who have filed complaints.

7. **Card Usage Analysis**

- Impact of Card Type on Churn: Examine if different Card Types have different churn rates.
- Loyalty Points Analysis: Investigate whether Points Earned from credit card usage influence customer retention.

8. **Salary Analysis**

- Salary and Churn: Analyze the relationship between EstimatedSalary and customer churn, focusing on how financial well-being might influence churn decisions.

9. **Insights and Recommendation**

Importing Libraries:

In [1]:
```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt # data visualization

import seaborn as sns # data visualization

import plotly.graph_objects as go # data visualization

from plotly.subplots import make_subplots # data visualization

import plotly.express as px # data visualization

from scipy import stats  # statistical analysis

from scipy.stats import chi2_contingency, ttest_ind, shapiro, levene, ttest_ind, mannwhitneyu # perform hypothesis testi

import statsmodels.api as sm  # statistical analysis

import warnings # control how warnings are handled
warnings.filterwarnings('ignore')
```

Reading the dataset:

In [2]:
```
df = pd.read_csv('Bank-Records.csv')
```

Looking at the dataset:

In [3]:
```
df.head()
```

Out[3]:

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

In [4]:
```
df.tail()
```

Out[4]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	E
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	

Shape of the dataset:

In [5]:
```
df.shape
```

Out[5]:
```
(10000, 18)
```

In [6]:
```
print(f"# rows: {df.shape[0]} \n# columns: {df.shape[1]}")
```

```
# rows: 10000
# columns: 18
```

Columns in the Dataset:

In [7]:
```
df.columns
```

Out[7]:
```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited', 'Complain',
       'Satisfaction Score', 'Card Type', 'Point Earned'],
      dtype='object')
```

Datatype of the columns:

In [8]:
```
df.dtypes
```

Out[8]:
```
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
Complain               int64
Satisfaction Score     int64
Card Type             object
Point Earned           int64
dtype: object
```

Basic information about the dataset:

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   RowNumber          10000 non-null  int64  
 1   CustomerId         10000 non-null  int64  
 2   Surname            10000 non-null  object 
 3   CreditScore        10000 non-null  int64  
 4   Geography          10000 non-null  object 
 5   Gender             10000 non-null  object 
 6   Age                10000 non-null  int64  
 7   Tenure             10000 non-null  int64  
 8   Balance            10000 non-null  float64
 9   NumOfProducts      10000 non-null  int64  
 10  HasCrCard          10000 non-null  int64  
 11  IsActiveMember     10000 non-null  int64  
 12  EstimatedSalary    10000 non-null  float64
 13  Exited             10000 non-null  int64  
 14  Complain           10000 non-null  int64  
 15  Satisfaction Score 10000 non-null  int64  
 16  Card Type          10000 non-null  object 
 17  Point Earned       10000 non-null  int64  
dtypes: float64(2), int64(12), object(4)
memory usage: 1.4+ MB
```

Missing value detection and perform imputation:

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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
```

Observations-

- There are no missing values in the dataset.

Identify and remove duplicate records:

In [11]:
```
df.duplicated()
```

Out[11]:
```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

In [12]:
```
np.any(df.duplicated())
```

Out[12]:
```
False
```

Observations-

- There are no duplicate values in the dataset.

Datatype of following attributes needs to changed to proper data type:

- **HasCrCard** - to categorical
- **IsActiveMember** - to categorical
- **Exited** - to categorical
- **Complain** - to categorical

In [13]:
```
df['HasCrCard'].replace({0 : 'No', 1 : 'Yes'}, inplace = True)

df['IsActiveMember'].replace({0 : 'No', 1 : 'Yes'}, inplace = True)

df['Exited'].replace({0 : 'No', 1 : 'Yes'}, inplace = True)

df['Complain'].replace({0 : 'No', 1 : 'Yes'}, inplace = True)
```

In [14]:
```
df.head()
```

Out[14]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estim
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	Yes	Yes	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	No	Yes	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	Yes	No	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	No	No	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	Yes	Yes	

---

1. Descriptive Statistics:

1A. Basic Statistics:

Calculating mean, median, and mode for numerical columns like CreditScore, Age, Balance, NumOfProducts, EstimatedSalary, and Points Earned.

Basic statistical information about the dataset:

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

Out[15]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	EstimatedSalary	Satisfaction Score	Point E
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	100090.239881	3.013800	606.5
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	57510.492818	1.405919	225.9
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	11.580000	1.000000	119.0
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	51002.110000	2.000000	410.0
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	100193.915000	3.000000	605.0
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	149388.247500	4.000000	801.0
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	199992.480000	5.000000	1000.0

In [16]:
```
df.describe(include = 'object')
```

Out[16]:

	Surname	Geography	Gender	HasCrCard	IsActiveMember	Exited	Complain	Card Type
count	10000	10000	10000	10000	10000	10000	10000	10000
unique	2932	3	2	2	2	2	2	4
top	Smith	France	Male	Yes	Yes	No	No	DIAMOND
freq	32	5014	5457	7055	5151	7962	7956	2507

Mean, Median, and Mode for numerical columns:

In [17]:
```
# Selecting the specified columns
selected_columns = ['CreditScore', 'Age', 'Balance', 'NumOfProducts', 'EstimatedSalary', 'Point Earned']

# Calculating mean, median, and mode for the selected columns
mean_values = df[selected_columns].mean()
median_values = df[selected_columns].median()
mode_values = df[selected_columns].mode().iloc[0]  # mode() returns a DataFrame, take the first row for the mode values

# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

print(f"{bold_start}Mean Values:{bold_end}")
print(mean_values)
print(f"\n{bold_start}Median Values:{bold_end}")
print(median_values)
print(f"\n{bold_start}Mode Values:{bold_end}")
print(mode_values)
```

```
Mean Values:
CreditScore           650.528800
Age                    38.921800
Balance             76485.889288
NumOfProducts           1.530200
EstimatedSalary    100090.239881
Point Earned          606.515100
dtype: float64

Median Values:
CreditScore           652.000
Age                    37.000
Balance             97198.540
NumOfProducts           1.000
EstimatedSalary    100193.915
Point Earned          605.000
dtype: float64

Mode Values:
CreditScore          850.00
Age                   37.00
Balance                0.00
NumOfProducts          1.00
EstimatedSalary    24924.92
Point Earned         408.00
Name: 0, dtype: float64
```

---

1. Descriptive Statistics:

1B. Distribution Analysis:

Analyze the distribution of key numerical variables using histograms and box plots to understand the spread and central tendency.

Non Graphical Analysis

In [18]:
```
# Columns to calculate value counts for
columns = ['Geography', 'Gender', 'NumOfProducts', 'HasCrCard', 'IsActiveMember',
           'Exited', 'Complain', 'Satisfaction Score', 'Card Type']

# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

for column in columns:
    print(f"{bold_start}{column} Value Counts:{bold_end}")
    print(df[column].value_counts())
    print('\n')
```

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

Gender Value Counts:
```
Male      5457
Female    4543
Name: Gender, dtype: int64
```

NumOfProducts Value Counts:
```
1    5084
2    4590
3     266
4      60
Name: NumOfProducts, dtype: int64
```

HasCrCard Value Counts:
```
Yes    7055
No     2945
Name: HasCrCard, dtype: int64
```

IsActiveMember Value Counts:
```
Yes    5151
No     4849
Name: IsActiveMember, dtype: int64
```

Exited Value Counts:
```
No     7962
Yes    2038
Name: Exited, dtype: int64
```

Complain Value Counts:
```
No     7956
Yes    2044
Name: Complain, dtype: int64
```

Satisfaction Score Value Counts:
```
3    2042
2    2014
4    2008
5    2004
1    1932
Name: Satisfaction Score, dtype: int64
```

Card Type Value Counts:
```
DIAMOND     2507
GOLD        2502
SILVER      2496
PLATINUM    2495
Name: Card Type, dtype: int64
```

Observations-

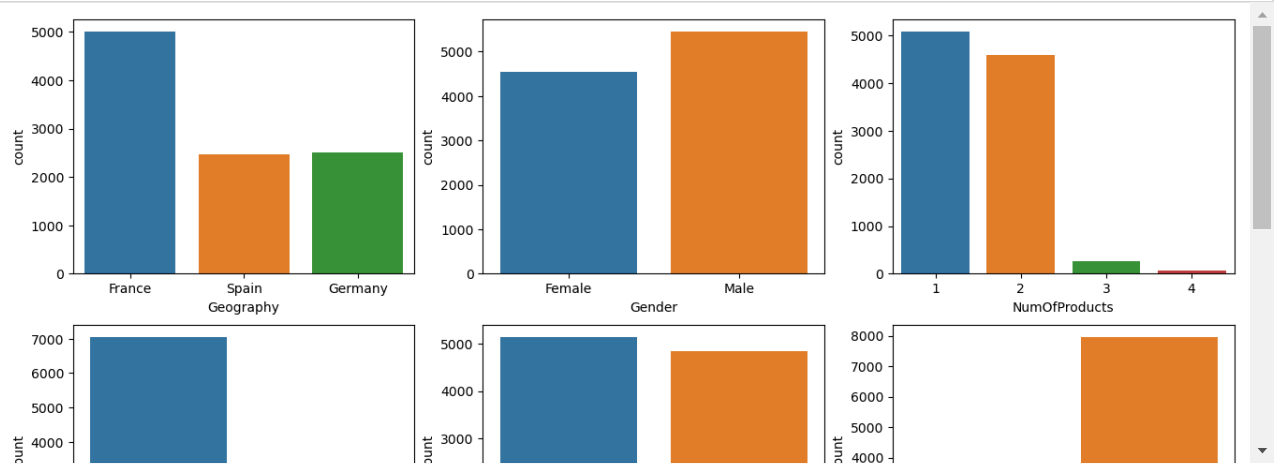
- 50% Customers are from France.
- 70% Customer have Credit Card.
- There seems to be more Male as compared to Females but by small margin.

- Complain and Exited seems to have some corelation since they have same numbers.
- Marginally have large number of active members as compared to non-active members.

Graphical Analysis

Bar charts for categorical variables.

In [19]:
```
# Define the categorical columns
cat_cols = ['Geography', 'Gender', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'Exited',
            'Complain', 'Satisfaction Score', 'Card Type']

fig, axis = plt.subplots(nrows = 3,  # Create a 3x3 grid for subplots
                         ncols = 3,
                         figsize = (16, 12))

index = 0

for row in range(3):  # Plotting countplots
    for col in range(3):
        sns.countplot(data = df,
                      x = cat_cols[index],
                      ax = axis[row, col])
        index += 1

plt.show()  # Displaying the plots
```



Observations-

The bar charts provide a clear visual summary of the distribution of various categorical variables in the dataset. Here are the observations from each bar chart:

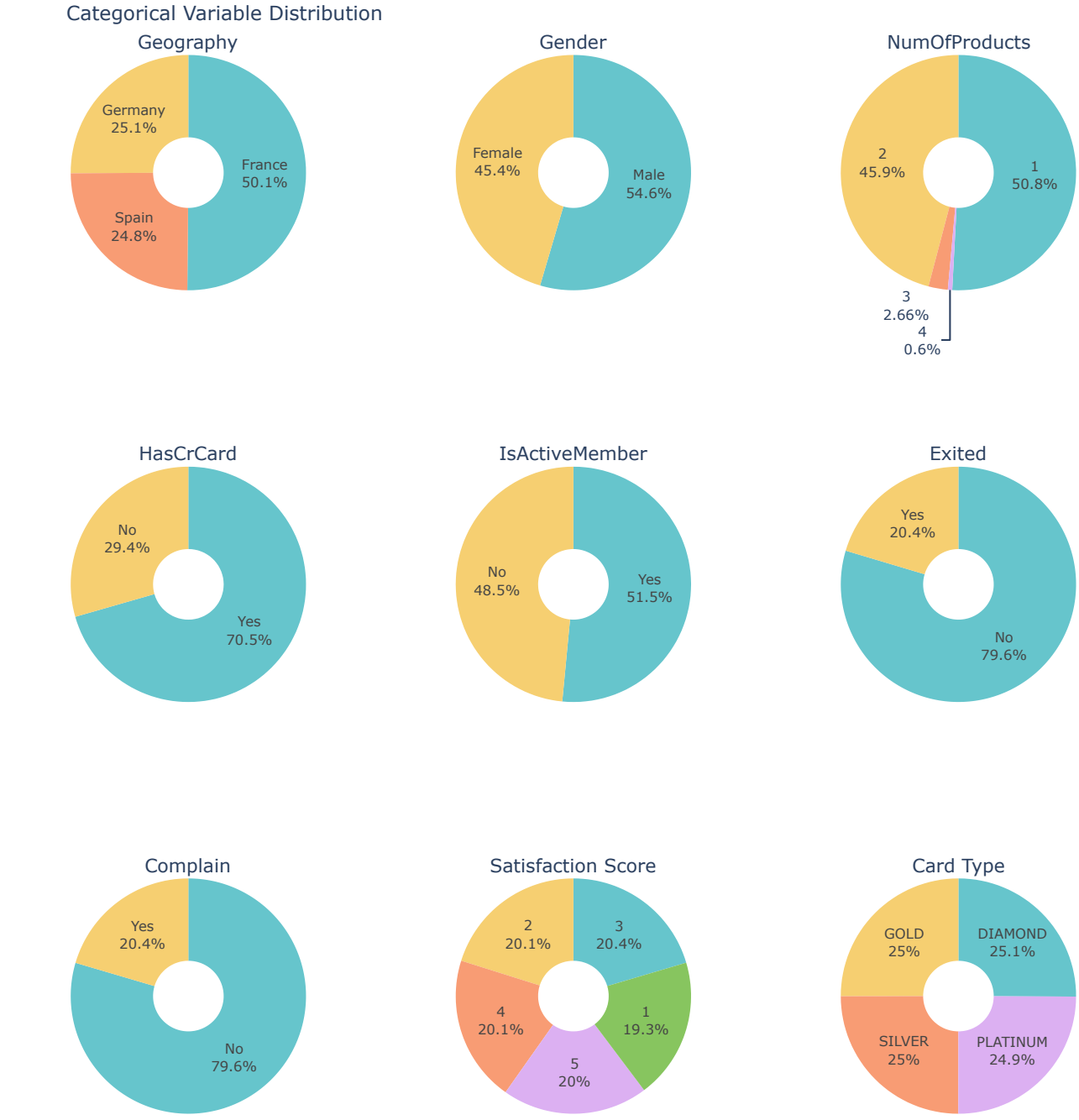
- Most customers are from France, followed by Germany and Spain.
- The dataset has a slightly higher proportion of male customers compared to female customers.
- The majority of customers have either 1 or 2 products, with very few having 3 or 4 products.
- A significant majority of customers have a credit card.
- The dataset has a nearly even split between active and inactive members, with slightly more active members.
- The majority of customers have not exited the service, with only a small percentage having exited.
- Most customers have not filed a complaint, similar to the exited distribution.
- The satisfaction scores are fairly evenly distributed across all possible scores, with no significant skew.
- The distribution of card types is almost evenly split among the four categories.

These observations highlight the demographic and behavioral characteristics of the customers in the dataset.

Graphical Analysis

Pie charts for categorical variables.

In [20]:
```
# Define the categorical columns
cat_cols = ['Geography', 'Gender', 'NumOfProducts', 'HasCrCard', 'IsActiveMember', 'Exited',
            'Complain', 'Satisfaction Score', 'Card Type']

# Create subplots: 3 rows, 3 columns
fig = make_subplots(rows=3, cols=3, subplot_titles=cat_cols, specs=[[{'type':'domain'}]*3]*3)

index = 0

for row in range(1, 4):  # Plotting pie charts
    for col in range(1, 4):
        if index < len(cat_cols):
            # Get the value counts for the current categorical column
            value_counts = df[cat_cols[index]].value_counts()

            # Add pie chart to the subplot
            fig.add_trace(go.Pie(
                labels=value_counts.index,
                values=value_counts,
                textinfo='percent+label',
                marker=dict(colors=px.colors.qualitative.Pastel),
                hole=.3  # Adding a hole to make it a donut chart, adjust as needed
            ), row=row, col=col)

            index += 1

# Adjust layout for larger pie charts
fig.update_layout(
    height=1000,  # Increase the figure height
    width=1000,  # Increase the figure width
    title_text="Categorical Variable Distribution",
    showlegend=False,
    grid=dict(rows=3, columns=3, pattern='independent'),
    margin=dict(l=20, r=20, t=60, b=20)  # Adjust margins to fit larger charts
)

# Display the plots
fig.show()
```



Observations-

The pie charts provide a visual summary of the distribution of various categorical variables in the dataset. Here are the observations from each pie chart:

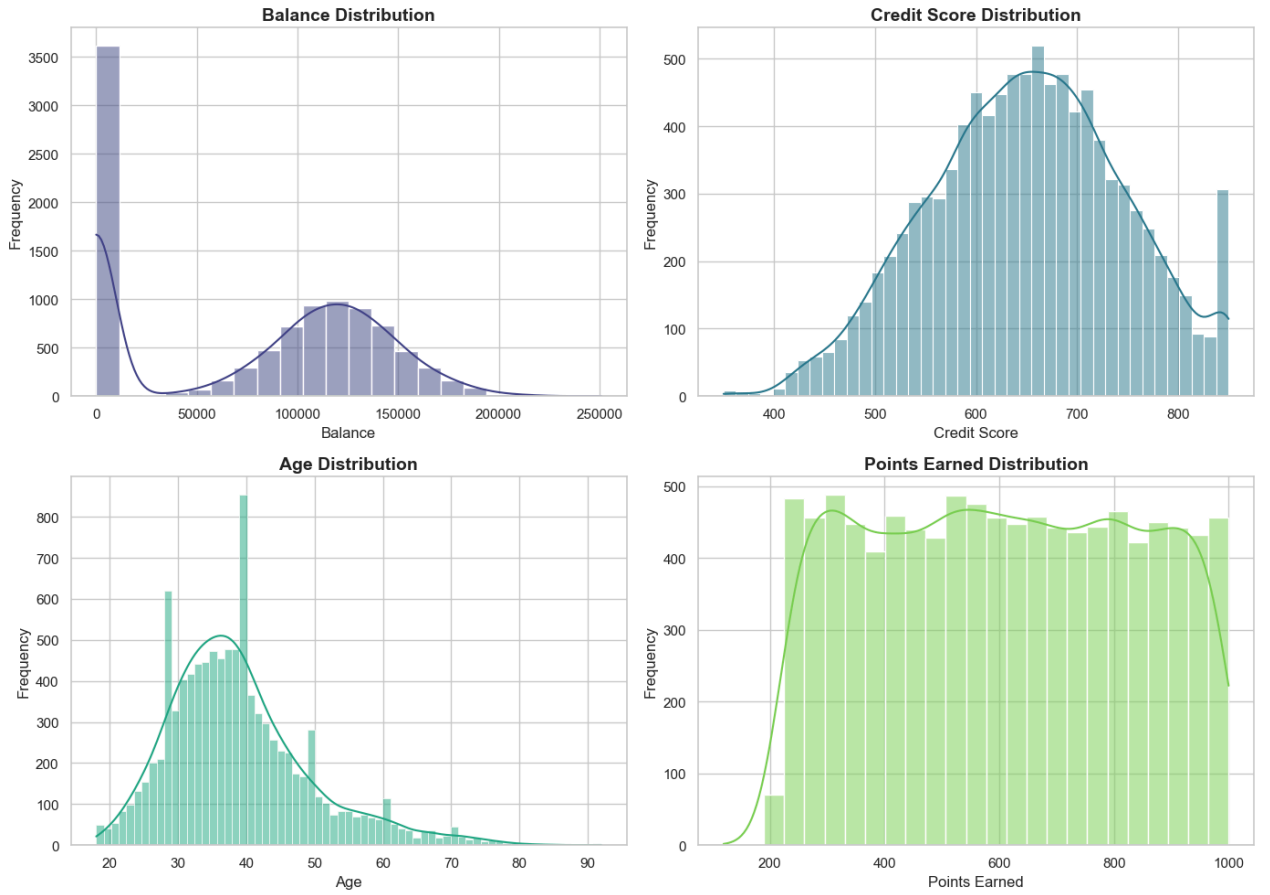
- Most customers are from France, followed by Germany and Spain.
- The dataset has a slightly higher proportion of male customers compared to female customers.
- The majority of customers have either 1 or 2 products, with very few having 3 or 4 products.
- A significant majority of customers have a credit card.
- The dataset has a nearly even split between active and inactive members, with slightly more active members.
- The majority of customers have not exited the service, with only a small percentage having exited.
- Most customers have not filed a complaint, similar to the exited distribution.
- The satisfaction scores are fairly evenly distributed across all possible scores, with no significant skew.
- The distribution of card types is almost evenly split among the four categories.

These observations highlight the demographic and behavioral characteristics of the customers in the dataset.

Graphical Analysis

Histograms for numerical variables.

In [21]:
```
# Setting the style and palette
sns.set(style="whitegrid")
palette = sns.color_palette("viridis", as_cmap=True)

fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(14, 10))

# Histogram for Balance
sns.histplot(data=df, x='Balance', kde=True, ax=axs[0, 0], color=palette(0.2))
axs[0, 0].set_title('Balance Distribution', fontsize=14, weight='bold')
axs[0, 0].set_xlabel('Balance', fontsize=12)
axs[0, 0].set_ylabel('Frequency', fontsize=12)

# Histogram for CreditScore
sns.histplot(data=df, x='CreditScore', kde=True, ax=axs[0, 1], color=palette(0.4))
axs[0, 1].set_title('Credit Score Distribution', fontsize=14, weight='bold')
axs[0, 1].set_xlabel('Credit Score', fontsize=12)
axs[0, 1].set_ylabel('Frequency', fontsize=12)

# Histogram for Age
sns.histplot(data=df, x='Age', kde=True, ax=axs[1, 0], color=palette(0.6))
axs[1, 0].set_title('Age Distribution', fontsize=14, weight='bold')
axs[1, 0].set_xlabel('Age', fontsize=12)
axs[1, 0].set_ylabel('Frequency', fontsize=12)

# Histogram for Point Earned
sns.histplot(data=df, x='Point Earned', kde=True, ax=axs[1, 1], color=palette(0.8))
axs[1, 1].set_title('Points Earned Distribution', fontsize=14, weight='bold')
axs[1, 1].set_xlabel('Points Earned', fontsize=12)
axs[1, 1].set_ylabel('Frequency', fontsize=12)

# Adjust layout for better spacing
plt.tight_layout()

# Show plot
plt.show()
```



Observations-

The histograms provide a detailed view of the distribution of several numerical variables in the dataset. Here are the observations from each histogram:

Balance Distribution:

- The majority of customers have a balance close to zero, with a sharp peak at the very low end.
- There is a noticeable number of customers with balances between 100,000 and 200,000, forming a secondary peak.
- The distribution shows a right skew, with most customers having lower balances and fewer customers with higher balances.

Credit Score Distribution:

- The credit scores range from approximately 350 to 850.
- The distribution appears to be roughly normal, with a peak around 650-700.
- The majority of customers have credit scores between 600 and 750, with fewer customers at the extremes of the range.

Age Distribution:

- The age of customers ranges from approximately 18 to 90.
- The distribution is right-skewed, with a majority of customers aged between 30 and 40.
- There are noticeable peaks around ages 35 and 40, indicating larger groups of customers in these age ranges.
- The frequency of customers decreases steadily as age increases beyond 40.

Points Earned Distribution:

- The points earned range from approximately 0 to 1000.
- The distribution is relatively uniform, with slight variations but no significant peaks or skewness.
- This suggests that customers are evenly distributed across the range of points earned.

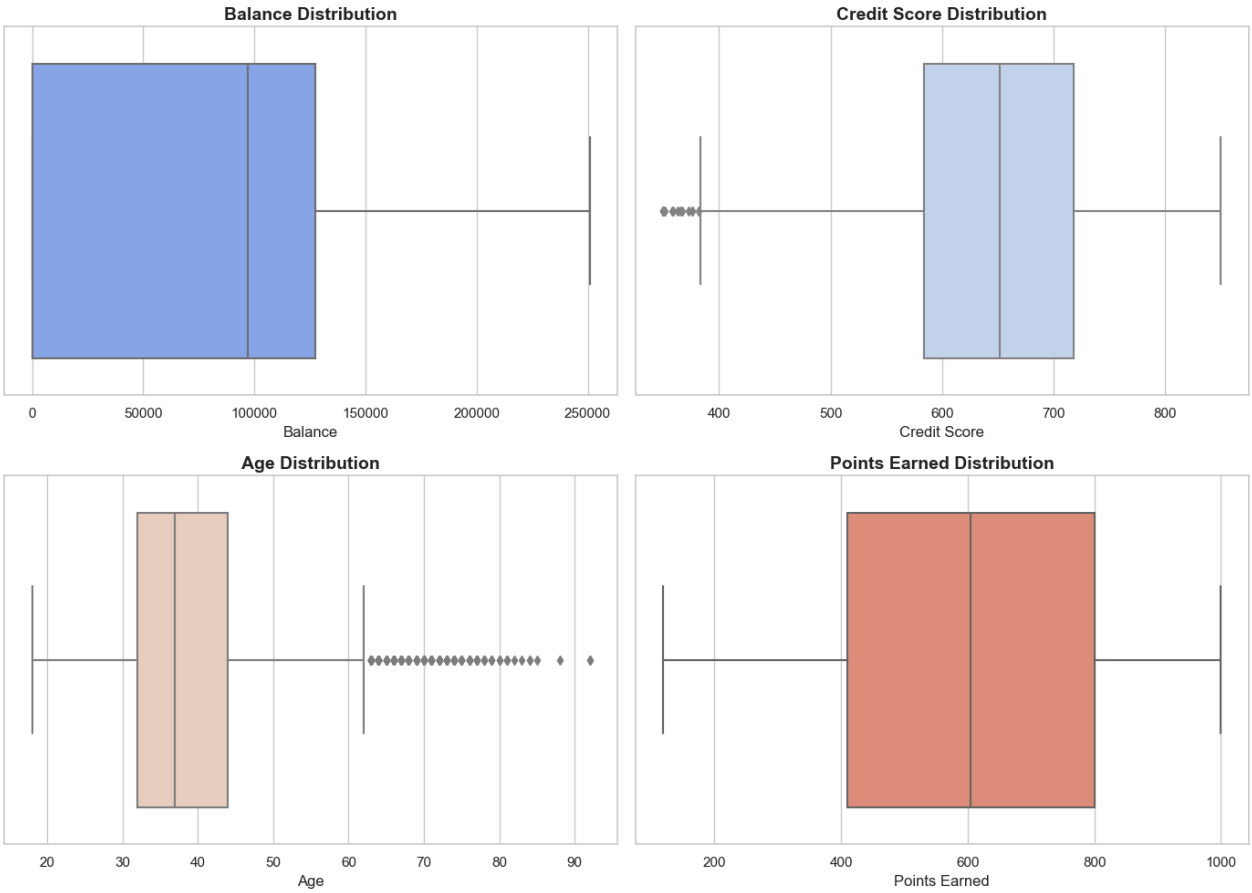
These observations provide insights into the financial and demographic characteristics of the customers in the dataset, highlighting trends and

---

Graphical Analysis

Box plots for numerical variables.

In [22]:
```
# Setting the style and palette
sns.set(style="whitegrid")
palette = sns.color_palette("coolwarm", as_cmap=True)

fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(14, 10))

# Box plot for Balance
sns.boxplot(data=df, x='Balance', ax=axs[0, 0], color=palette(0.2))
axs[0, 0].set_title('Balance Distribution', fontsize=14, weight='bold')
axs[0, 0].set_xlabel('Balance', fontsize=12)

# Box plot for CreditScore
sns.boxplot(data=df, x='CreditScore', ax=axs[0, 1], color=palette(0.4))
axs[0, 1].set_title('Credit Score Distribution', fontsize=14, weight='bold')
axs[0, 1].set_xlabel('Credit Score', fontsize=12)

# Box plot for Age
sns.boxplot(data=df, x='Age', ax=axs[1, 0], color=palette(0.6))
axs[1, 0].set_title('Age Distribution', fontsize=14, weight='bold')
axs[1, 0].set_xlabel('Age', fontsize=12)

# Box plot for Point Earned
sns.boxplot(data=df, x='Point Earned', ax=axs[1, 1], color=palette(0.8))
axs[1, 1].set_title('Points Earned Distribution', fontsize=14, weight='bold')
axs[1, 1].set_xlabel('Points Earned', fontsize=12)

# Adjust layout for better spacing
plt.tight_layout()

# Show plot
plt.show()
```



Observations-

These box plots show the distributions of four variables: Balance, Credit Score, Age, and Points Earned. Here's a detailed description and observations for each:

Balance Distribution:

- Description: The box plot shows the balance amounts, ranging from 0 to 250,000.
- Observations:
  - The median balance is slightly below 100,000.
  - The interquartile range (IQR), represented by the box, stretches from around 0 to 100,000.
  - There are no noticeable outliers, indicating that most balance values are within this range.
  - The whiskers extend up to 250,000, showing the range of the data.

Credit Score Distribution:

- Description: The box plot displays the distribution of credit scores, ranging from about 300 to 850.
- Observations:
  - The median credit score is approximately 650.
  - The IQR spans from around 600 to 700.
  - There are several outliers below 500, indicating some low credit scores.
  - The whiskers extend beyond the IQR, showing the full range of the data.

Age Distribution:

- Description: This box plot depicts the age distribution of the subjects, ranging from 20 to 90.
- Observations:
  - The median age is around 40.
  - The IQR ranges from about 30 to 50.
  - There are many outliers above 60, indicating some older subjects.
  - The whiskers extend to the minimum and maximum ages, with the data ranging from 20 to 90.

Points Earned Distribution:

- Description: The box plot shows the distribution of points earned, ranging from 0 to 1000.
- Observations:
  - The median points earned is approximately 600.
  - The IQR stretches from around 400 to 800.
  - There are no noticeable outliers, indicating most values are within this range.
  - The whiskers extend from 0 to 1000, showing the entire range of the data.

General Observations:

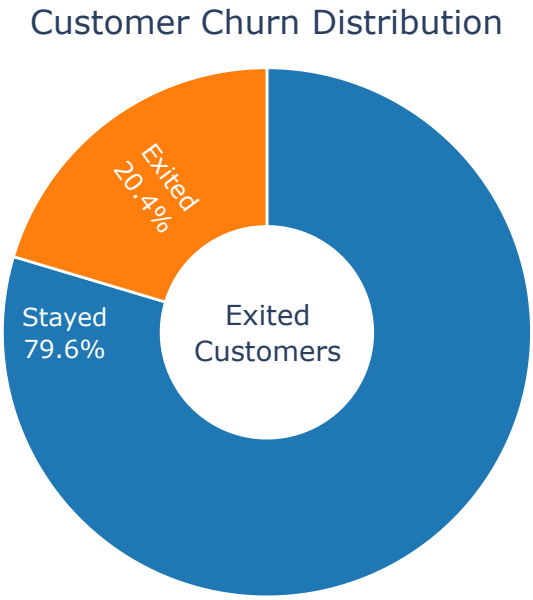
- The data distributions for balance and points earned are relatively wide, with large ranges and no significant outliers.
- The credit score and age distributions have noticeable outliers, particularly on the lower end for credit scores and the upper end for age.
- The age distribution shows a concentration of values in the middle age range (30-50), with outliers representing older ages.
- The balance and points earned distributions appear to be more symmetrical compared to the credit score and age distributions.

Graphical Analysis

Customer Churn Distribution.

In [23]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Identifying the number of customer who churned
# Value counts for the "Exited" column
df['Exited'] = df['Exited'].replace({1 : 'Churned', 0 : 'Not Churned'})
print(f'{bold_start}The number of customers who churned and who stayed:{bold_end}')
value_counts = df['Exited'].value_counts()
print(value_counts)

# Custom colors
colors = ['#1f77b4', '#ff7f0e']

# Create pie chart
fig = go.Figure(data = [go.Pie(
    labels = ['Stayed', 'Exited'],  # Using custom labels for clarity
    values = value_counts,
    textinfo = 'label+percent',
    insidetextorientation = 'radial',
    marker = dict(colors = colors, line = dict(color = '#FFFFFF', width = 2)),
    hoverinfo = 'label+percent+value',
    textfont = dict(size = 18, color = 'white'),
    showlegend = True,
    hole = 0.4  # Make it a donut chart for enhanced aesthetics
)])

# Add annotations
annotations = [
    dict(
        text = 'Exited<br>Customers',
        x = 0.5,
        y = 0.5,
        font_size = 20,
        showarrow = False
    )
]

# Update layout for more customizations
fig.update_layout(
    title = 'Customer Churn Distribution',
    annotations = annotations,
    title_x = 0.5,
    title_font = dict(size = 24),
    margin = dict(l = 50, r = 50, t = 50, b = 50),
    paper_bgcolor = 'rgba(0,0,0,0)',
    plot_bgcolor = 'rgba(0,0,0,0)',
    legend = dict(
        orientation = "h",
        yanchor = "bottom",
        y = -0.2,
        xanchor = "center",
        x = 0.5,
        font = dict(size=14)
    )
)

# Display the plot
fig.show()
```

```
The number of customers who churned and who stayed:
No     7962
Yes    2038
Name: Exited, dtype: int64
```



Observations-

This pie chart illustrates the distribution of customer churn, distinguishing between customers who stayed and those who exited.

Description:

The chart is divided into two segments:

- **Stayed:** Represented by a blue segment, accounting for 79.6% of the customers.
- **Exited:** Represented by an orange segment, accounting for 20.4% of the customers.

Observations:

- **Majority of Customers Stayed:** A significant majority, 79.6%, of the customers chose to stay.
- **Minority of Customers Exited:** A smaller portion, 20.4%, of the customers exited.
- **Churn Rate:** The churn rate (customers who exited) is 20.4%, which could be a concern for the business if the goal is to minimize churn.
- **Customer Retention:** The retention rate (customers who stayed) is relatively high at 79.6%, indicating that most customers are satisfied or see value in staying.

Overall, the pie chart shows that while most customers remain with the company, a non-negligible percentage (20.4%) has left, highlighting an area for potential improvement in customer retention strategies.

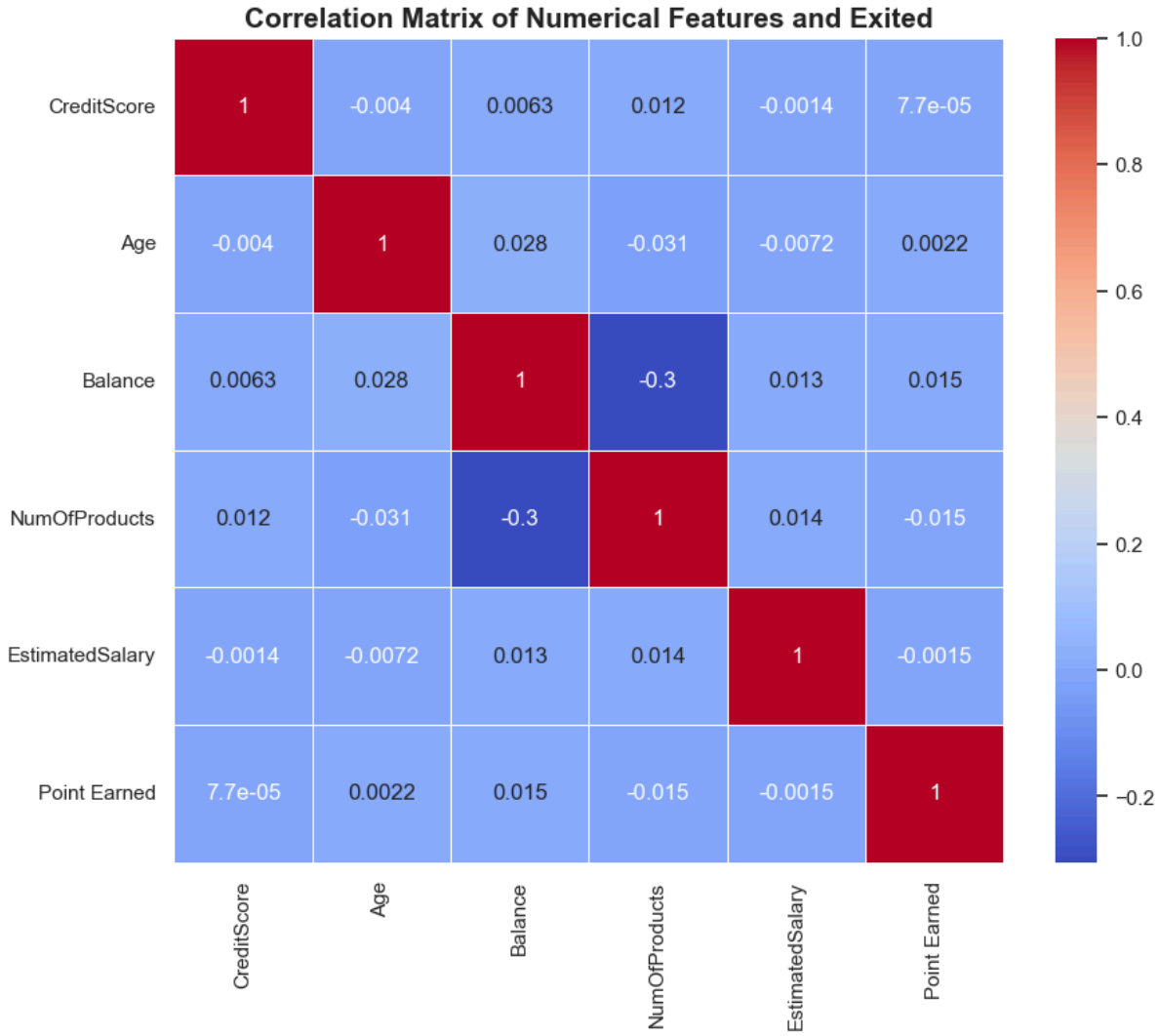
---

2. Exploratory Data Analysis (EDA):

2A. Correlation Analysis:

Exploring the correlation between numerical features and the Exited variable to identify potential predictors of churn.

In [24]:
```
# Selecting numerical columns and the Exited column
numerical_columns = ['CreditScore', 'Age', 'Balance', 'NumOfProducts', 'EstimatedSalary', 'Point Earned', 'Exited']

# Calculating the correlation matrix
correlation_matrix = df[numerical_columns].corr()

# Visualizing the correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix of Numerical Features and Exited', fontsize=15, weight='bold')
plt.show()
```



Observations-

This heatmap represents the correlation matrix for numerical features and customer churn status (Exited). The correlation coefficients range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.

- **Credit Score:**
  - Weak correlations with all other variables, with the highest being with the Number of Products (0.012).
- **Age:**
  - Weak correlations with all other variables, the highest being with Balance (0.028).
- **Balance:**
  - Negatively correlated with the Number of Products (-0.3), which is the strongest correlation in the matrix, suggesting that higher balances are associated with fewer products.
  - Weak correlations with other variables.
- **Number of Products (NumOfProducts):**
  - Negatively correlated with Balance (-0.3), as mentioned.
  - Very weak correlations with other variables.
- **Estimated Salary:**
  - Very weak correlations with all other variables.
- **Points Earned:**
  - Very weak correlations with all other variables.

General Observations

- **Weak Correlations:** Most correlations are weak, with values close to 0, indicating that these numerical features do not have strong linear relationships with each other.

- **Strongest Correlation:** The only moderate correlation is between Balance and Number of Products (-0.3), indicating an inverse relationship.
- **Independent Features:** The weak correlations suggest that these features can be considered relatively independent of each other in terms of their linear relationships.
- **Exited Variable:** If the "Exited" variable was included, its correlations with other features would be critical to understand churn drivers, but it is not explicitly shown here.

The weak correlations suggest that no strong linear relationships exist between these numerical features, which might imply that other types of

---

2. Exploratory Data Analysis (EDA):

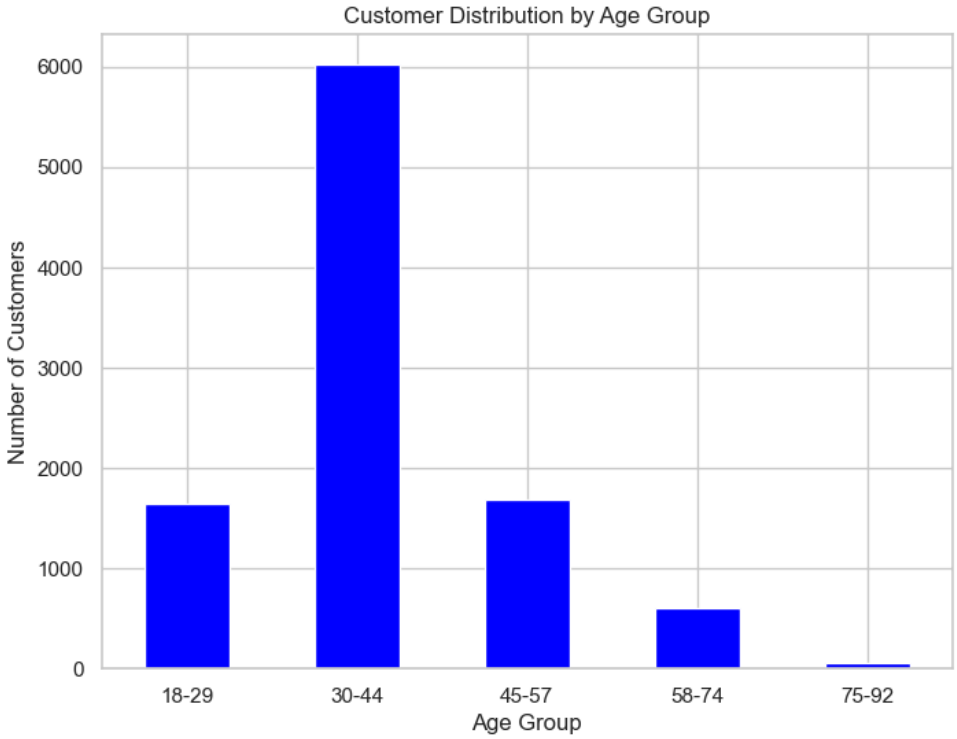
2B. Customer Profile Analysis:

Segment customers based on key demographics (Age, Geography, Gender) to identify which groups are more likely to churn.

In [25]:
```
# Converting 'Yes'/'No' to 1/0 in the Exited column
df['Exited'] = df['Exited'].apply(lambda x: 1 if x == 'Yes' else 0)
```

Segmenting customers by AgeGroup:

In [26]:
```
# Create age bins
age_bins = [18,30,45,58,75,93]
age_labels = ['18-29','30-44','45-57','58-74','75-92']

# Create AgeGroup column based on age brackets
df['AgeGroup'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=False)

# Count customers in each age group
age_counts = df['AgeGroup'].value_counts().sort_index()

# Plotting
plt.figure(figsize=(8, 6))
age_counts.plot(kind='bar', color='blue')
plt.title('Customer Distribution by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of Customers')
plt.xticks(rotation=0)
plt.show()
```



Observations-

The bar chart titled "Customer Distribution by Age Group" illustrates the number of customers across different age groups. Here are some observations:

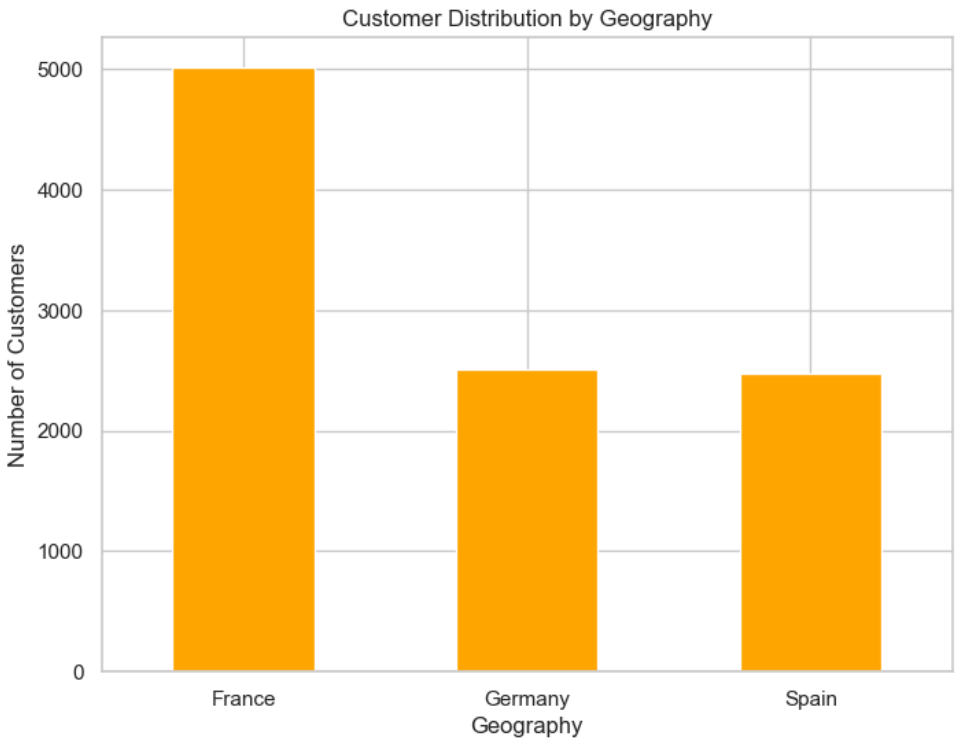
- **Dominant Age Group:** The 30-44 age group has the highest number of customers, with around 6000 individuals. This suggests that the majority of customers fall within this age range.

- **Secondary Age Group:** The next largest age group is the 45-57 range, with approximately 2000 customers. This indicates a significant drop from the 30-44 age group.
- **Young Adults:** The 18-29 age group has about 1500 customers, making it the third largest segment.
- **Older Adults:** The 58-74 age group has fewer customers, around 500. This shows a declining trend in the number of customers as age increases.
- **Senior Citizens:** The 75-92 age group has the least number of customers, with less than 100 individuals. This further supports the trend of decreasing customer numbers with increasing age.
- **Overall Trend:** There is a clear peak in the 30-44 age group, with a significant drop in customer numbers both before and after this age range.

The chart indicates that the majority of customers are middle-aged, with a sharp decline in both younger and older age groups. This could be useful

Segmenting customers by Geography:

In [27]:
```
# Count customers by Geography
geo_counts = df['Geography'].value_counts()

# Plotting
plt.figure(figsize=(8, 6))
geo_counts.plot(kind='bar', color='orange')
plt.title('Customer Distribution by Geography')
plt.xlabel('Geography')
plt.ylabel('Number of Customers')
plt.xticks(rotation=0)
plt.show()
```



Observations-

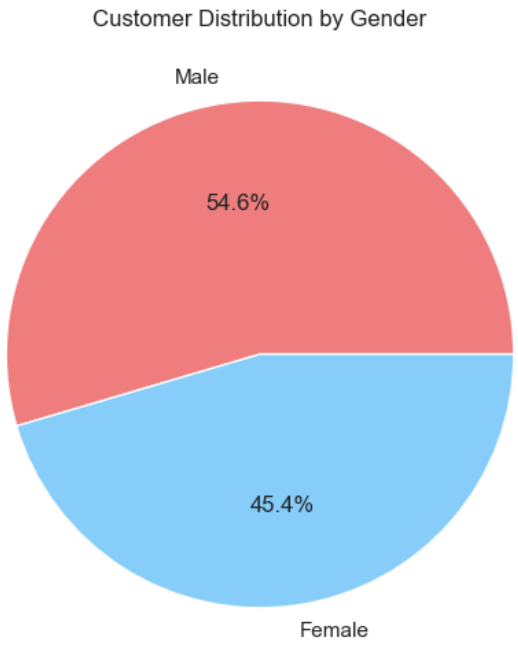
The bar chart titled "Customer Distribution by Geography" illustrates the number of customers across three countries: France, Germany, and Spain. Here are some observations:

- **France:** France has the highest number of customers, with around 5000 individuals. This suggests that France is the largest market among the three countries.
- **Germany:** Germany has the second-highest number of customers, with approximately 2500 individuals. This is about half the number of customers in France.
- **Spain:** Spain has slightly fewer customers than Germany, with around 2500 individuals. The difference between Germany and Spain is relatively small.
- **Comparison:** The number of customers in France is significantly higher than in Germany and Spain, indicating a strong presence or preference in the French market.
- **Overall Trend:** There is a clear leading country (France), with Germany and Spain having similar but lower customer numbers.

The chart indicates that France is a dominant market in terms of customer numbers, with Germany and Spain having a similar but considerably smaller customer base. This information could be useful for strategic decisions regarding market focus and resource allocation.

Segmenting customers by Gender:

In [28]:
```
# Count customers by Gender
gender_counts = df['Gender'].value_counts()

# Plotting
plt.figure(figsize=(8, 6))
gender_counts.plot(kind='pie', autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue'])
plt.title('Customer Distribution by Gender')
plt.ylabel('')
plt.show()
```



Observations-

The pie chart titled "Customer Distribution by Gender" illustrates the percentage of male and female customers. Here are some observations:

- **Male Customers:** Male customers make up 54.6% of the total customer base. This indicates that there are slightly more male customers compared to female customers.
- **Female Customers:** Female customers account for 45.4% of the total customer base. While this is less than the percentage of male customers, it is still a significant portion of the customer base.
- **Gender Distribution:** The difference between male and female customers is not very large, with males having a modest lead of 9.2%.
- **Overall Trend:** The distribution is relatively balanced, with neither gender overwhelmingly dominating the customer base.

The chart indicates a slight majority of male customers, but overall, the gender distribution is fairly even. This information can be useful for understanding the customer demographics and tailoring marketing strategies to address both genders effectively.

---

3. Comparative Analysis:

3A. Churn by Geography:

Comparing churn rates across different geographical locations to see if certain regions have higher churn rates.

Calculating churn rate by AgeGroup, Geography and Gender:

In [29]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Create age bins
age_bins = [18,30,45,58,75,93]
age_labels = ['18-29','30-44','45-57','58-74','75-92']
df['AgeGroup'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=False)

# Calculate churn rate by AgeGroup
age_churn_rate = df.groupby('AgeGroup')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by AgeGroup:{bold_end}")
print(age_churn_rate)

# Calculate churn rate by Geography
geography_churn_rate = df.groupby('Geography')['Exited'].mean() * 100
print(f"{bold_start}\nChurn Rate by Geography:{bold_end}")
print(geography_churn_rate)

# Calculate churn rate by Gender
gender_churn_rate = df.groupby('Gender')['Exited'].mean() * 100
print(f"{bold_start}\nChurn Rate by Gender:{bold_end}")
print(gender_churn_rate)
```

```
Churn Rate by AgeGroup:
AgeGroup
18-29     7.556368
30-44    14.454228
45-57    49.732938
58-74    34.109817
75-92     1.851852
Name: Exited, dtype: float64

Churn Rate by Geography:
Geography
France     16.174711
Germany    32.443204
Spain      16.673395
Name: Exited, dtype: float64

Churn Rate by Gender:
Gender
Female    25.071539
Male      16.474253
Name: Exited, dtype: float64
```

Plotting churn rate by Geography:

In [30]:
```
# Calculate churn rate by Geography
geography_churn_rate = df.groupby('Geography')['Exited'].mean() * 100
geography_churn_rate = geography_churn_rate.reset_index()

# Create bar chart for churn rate by Geography
fig_geo = px.bar(geography_churn_rate,
                 x='Geography',
                 y='Exited',
                 title='Churn Rate by Geography',
                 labels={'Exited': 'Churn Rate (%)'},
                 template='plotly_dark',
                 color='Geography',
                 color_discrete_sequence=px.colors.qualitative.Vivid)

# Update layout for better aesthetics
fig_geo.update_layout(
    title={'text': 'Churn Rate by Geography', 'x':0.5, 'xanchor': 'center'},
    xaxis_title='Geography',
    yaxis_title='Churn Rate (%)',
    font=dict(family='Arial', size=12, color='white'),
    bargap=0.2
)

# Show the plot
fig_geo.show()
```

Observations-

The bar chart titled "Churn Rate by Geography" illustrates the churn rates across three countries: France, Germany, and Spain. Here are some observations:

- **Germany:** Germany has the highest churn rate, with approximately 30%. This indicates that a significant portion of customers in Germany are leaving or discontinuing their service.
- **France:** France has a churn rate of around 15%. This is half the churn rate observed in Germany, suggesting that customers in France are more likely to remain.
- **Spain:** Spain's churn rate is around 20%. This places Spain in the middle, with a higher churn rate than France but lower than Germany.
- **Comparison:** The churn rate in Germany is notably higher compared to both France and Spain, indicating potential issues or dissatisfaction among customers in Germany.
- **Overall Trend:** There is a clear distinction in churn rates, with Germany having a significantly higher churn rate, followed by Spain and then France.

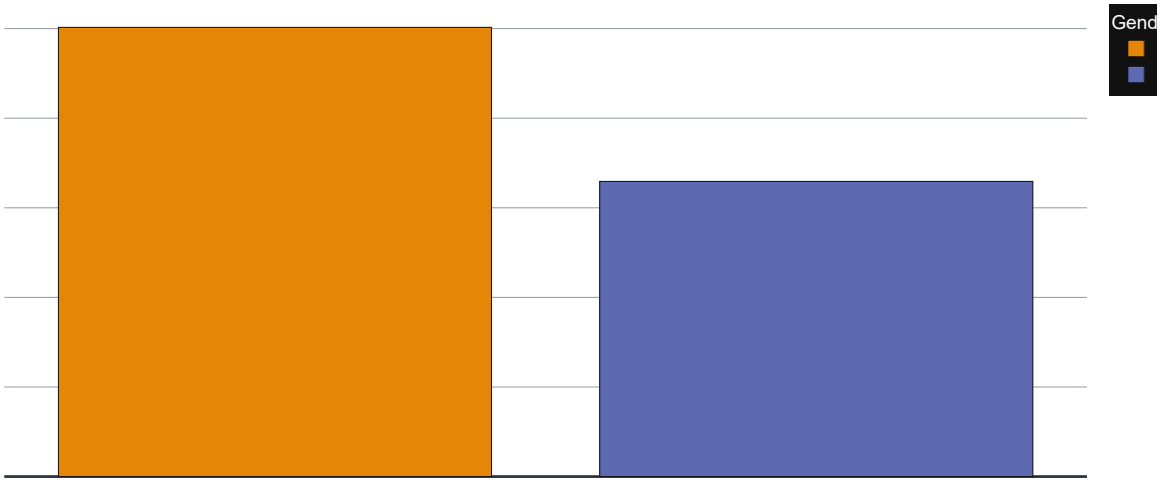
The chart indicates that Germany has the highest customer churn, which could be a cause for concern and may warrant further investigation into the reasons behind this high churn rate. France, with the lowest churn rate, suggests a relatively stable customer base, while Spain falls somewhere in between. This information can be crucial for devising targeted strategies to reduce churn rates, especially in Germany.

---

3. Comparative Analysis:

3B. Gender Differences in Churn:

In [31]:
```
# Calculate churn rate by Gender
gender_churn_rate = df.groupby('Gender')['Exited'].mean() * 100
gender_churn_rate = gender_churn_rate.reset_index()

# Create bar chart for churn rate by Gender
fig_gender = px.bar(gender_churn_rate,
                    x='Gender',
                    y='Exited',
                    title='Churn Rate by Gender',
                    labels={'Exited': 'Churn Rate (%)'},
                    template='plotly_dark',
                    color='Gender',
                    color_discrete_sequence=px.colors.qualitative.Vivid)

# Update layout for better aesthetics
fig_gender.update_layout(
    title={'text': 'Churn Rate by Gender', 'x':0.5, 'xanchor': 'center'},
    xaxis_title='Gender',
    yaxis_title='Churn Rate (%)',
    font=dict(family='Arial', size=12, color='white'),
    bargap=0.2
)

# Show the plot
fig_gender.show()
```



Observations-

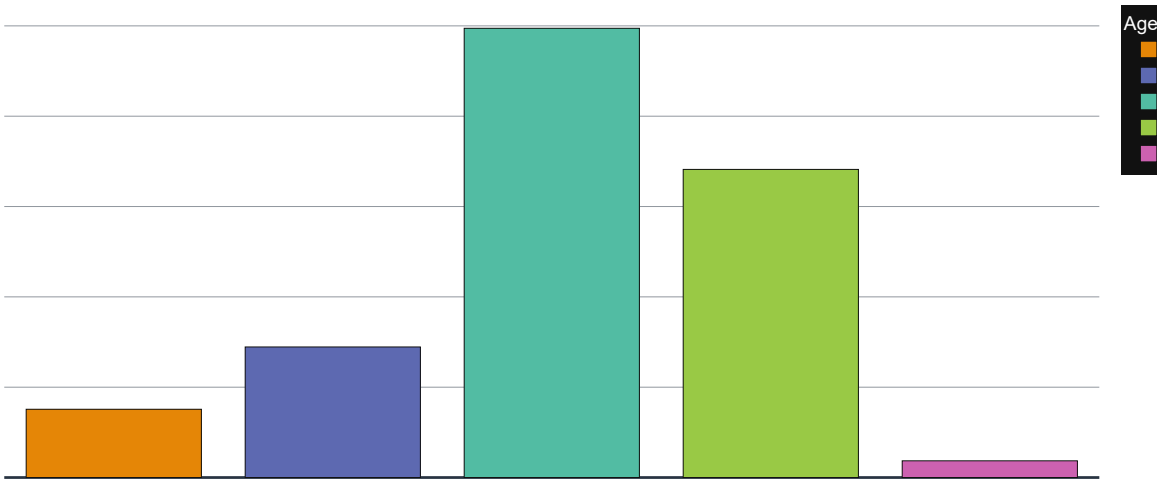
The bar chart titled "Churn Rate by Gender" illustrates the churn rates for male and female customers. Here are some observations:

- **Female Customers:** Female customers have a higher churn rate, around 25%. This indicates that a significant portion of female customers are leaving or discontinuing their service.
- **Male Customers:** Male customers have a lower churn rate, around 20%. This suggests that male customers are slightly more likely to stay compared to female customers.
- **Comparison:** There is a noticeable difference in churn rates between genders, with female customers having a higher churn rate than male customers.
- **Overall Trend:** The chart indicates that female customers are more prone to churn compared to male customers.

The chart shows a gender disparity in churn rates, with female customers leaving at a higher rate than male customers. This information can be valuable for identifying potential issues specific to female customers and developing targeted strategies to improve retention among this group.

Plotting churn rate by AgeGroup:

In [32]:
```
# Create age bins
age_bins = [18,30,45,58,75,93]
age_labels = ['18-29','30-44','45-57','58-74','75-92']
df['AgeGroup'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=False)

# Calculate churn rate by AgeGroup
age_churn_rate = df.groupby('AgeGroup')['Exited'].mean() * 100
age_churn_rate = age_churn_rate.reset_index()

# Plot churn rate by AgeGroup
fig_age = px.bar(age_churn_rate,
                 x='AgeGroup',
                 y='Exited',
                 title='Churn Rate by Age Group',
                 labels={'Exited': 'Churn Rate (%)', 'AgeGroup': 'Age Group'},
                 template='plotly_dark',
                 color='AgeGroup',
                 color_discrete_sequence=px.colors.qualitative.Vivid)

# Update layout for better aesthetics
fig_age.update_layout(
    title={'text': 'Churn Rate by Age Group', 'x':0.5, 'xanchor': 'center'},
    xaxis_title='Age Group',
    yaxis_title='Churn Rate (%)',
    font=dict(family='Arial', size=12, color='white'),
    bargap=0.2
)

# Show the plot
fig_age.show()
```



Observations-

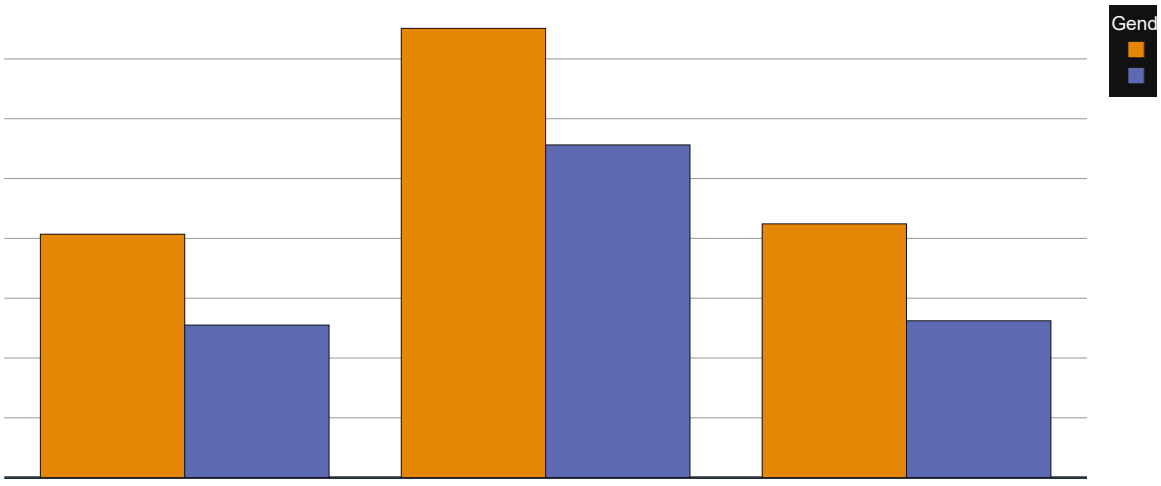
The bar chart titled "Churn Rate by Age Group" shows the churn rate percentages across different age groups. Here are some observations:

- **Age Group 45-57:** This group has the highest churn rate, close to 50%. This suggests that customers in this age range are more likely to leave compared to other age groups.
- **Age Group 58-74:** The second highest churn rate is observed in this group, with a churn rate around 35%. This indicates a significant number of customers in this age range are also leaving.
- **Age Group 30-44:** This age group has a moderate churn rate, approximately 15%. While not as high as the older age groups, it still indicates some level of churn.
- **Age Group 18-29:** The churn rate for this group is relatively low, around 5%. This suggests that younger customers are more likely to stay.
- **Age Group 75-92:** This age group has the lowest churn rate, almost negligible. It indicates that customers in this age range are the least likely to leave.

Overall, the churn rate appears to increase with age until the 45-57 age group and then decrease for the older age groups, with the 75-92 age group having the lowest churn rate. This pattern could be useful for targeted retention strategies based on age demographics.

Plotting churn rate by Geography and Gender:

In [33]:
```
# Calculate churn rate by Geography and Gender
geo_gender_churn_rate = df.groupby(['Geography', 'Gender'])['Exited'].mean() * 100
geo_gender_churn_rate = geo_gender_churn_rate.reset_index()

# Create bar chart for churn rate by Geography and Gender
fig_geo_gender = px.bar(geo_gender_churn_rate,
                        x='Geography',
                        y='Exited',
                        color='Gender',
                        barmode='group',
                        title='Churn Rate by Geography and Gender',
                        labels={'Exited': 'Churn Rate (%)'},
                        template='plotly_dark',
                        color_discrete_sequence=px.colors.qualitative.Vivid)

# Update layout for better aesthetics
fig_geo_gender.update_layout(
    title={'text': 'Churn Rate by Geography and Gender', 'x':0.5, 'xanchor': 'center'},
    xaxis_title='Geography',
    yaxis_title='Churn Rate (%)',
    font=dict(family='Arial', size=12, color='white'),
    bargap=0.2
)

# Show the plot
fig_geo_gender.show()
```



Observations-

The bar chart titled "Churn Rate by Geography and Gender" displays the churn rates for males and females in three different countries: France, Germany, and Spain. Here are some observations:

- **France:**
  - Female churn rate is around 20%.
  - Male churn rate is slightly above 10%.
  - Females have a higher churn rate compared to males in France.
- **Germany:**
  - Female churn rate is the highest among all groups at around 35%.
  - Male churn rate is around 30%.
  - Both genders in Germany have the highest churn rates compared to France and Spain, with females having a higher churn rate than males.
- **Spain:**
  - Female churn rate is approximately 20%.
  - Male churn rate is around 15%.
  - Similar to France, females in Spain have a higher churn rate than males.

Overall, across all three countries, females tend to have higher churn rates compared to males. Germany stands out with the highest churn rates for both genders, indicating a potential issue in this market that might need targeted attention.

4. Behavioral Analysis:

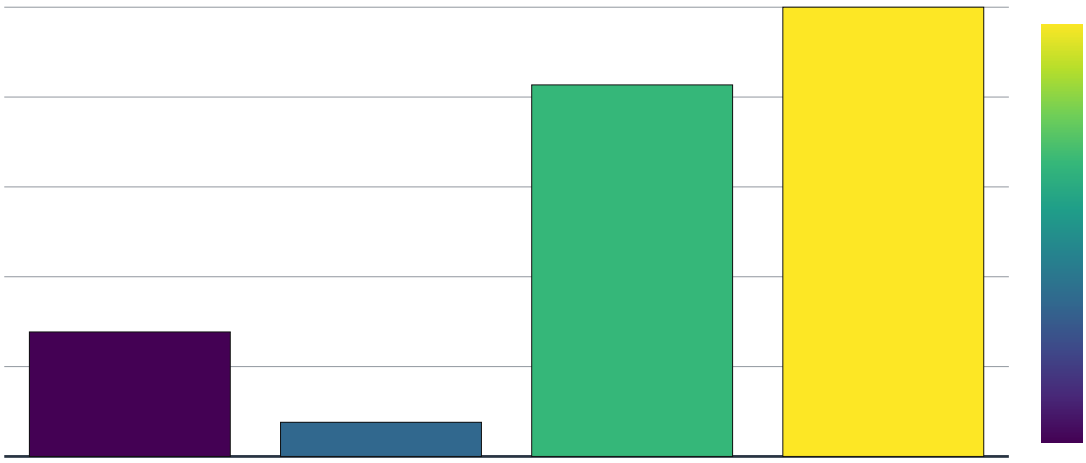
4A. Product and Services Usage:

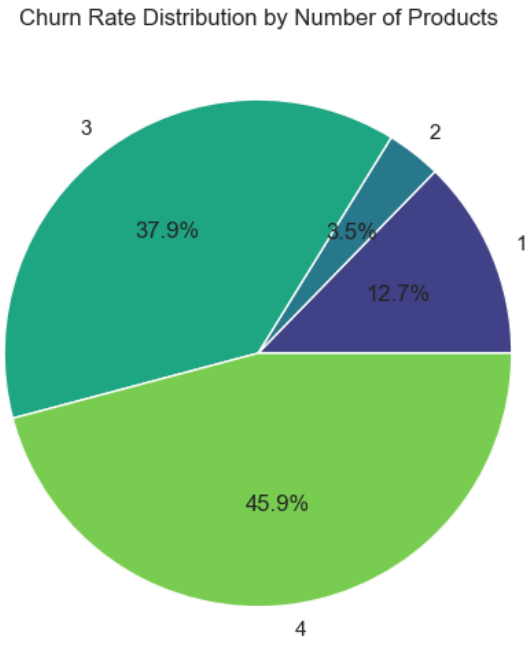
Examining how the number of products (NumOfProducts) a customer uses affects their likelihood to churn.

In [34]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by NumOfProducts
product_churn_rate = df.groupby('NumOfProducts')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Products and Services Usage:{bold_end}")
print(product_churn_rate)
```

```
Churn Rate by Products and Services Usage:
NumOfProducts
1     27.714398
2      7.603486
3     82.706767
4    100.000000
Name: Exited, dtype: float64
```

In [35]:
```
# Calculate churn rate by NumOfProducts
product_churn_rate = df.groupby('NumOfProducts')['Exited'].mean() * 100
product_churn_rate = product_churn_rate.reset_index()

# Create bar chart for churn rate by NumOfProducts using Plotly Express
fig_product = px.bar(product_churn_rate,
                     x='NumOfProducts',
                     y='Exited',
                     title='Churn Rate by Number of Products',
                     labels={'Exited': 'Churn Rate (%)', 'NumOfProducts': 'Number of Products'},
                     template='plotly_dark',
                     color='NumOfProducts',
                     color_continuous_scale=px.colors.sequential.Viridis)

# Update layout for better aesthetics
fig_product.update_layout(
    title={'text': 'Churn Rate by Number of Products', 'x':0.5, 'xanchor': 'center'},
    xaxis_title='Number of Products',
    yaxis_title='Churn Rate (%)',
    font=dict(family='Arial', size=12, color='white'),
    bargap=0.2
)

# Show the bar chart
fig_product.show()

# Create a DataFrame for pie chart
product_data = pd.DataFrame({
    'Number of Products': product_churn_rate['NumOfProducts'],
    'Churn Rate (%)': product_churn_rate['Exited']
})

# Plot the pie chart
plt.figure(figsize=(8, 6))
plt.pie(product_data['Churn Rate (%)'], labels=product_data['Number of Products'], autopct='%1.1f%%',
        colors=sns.color_palette('viridis', len(product_data['Number of Products'])))
plt.title('Churn Rate Distribution by Number of Products')
plt.show()
```





Observations-

The bar chart titled "Churn Rate by Number of Products" illustrates the churn rates for customers based on the number of products they use. Here are some observations:

- **1 Product:**
  - Churn rate is around 25%.
  - This indicates that customers with only one product have a moderate churn rate.
- **2 Products:**
  - Churn rate drops significantly to around 10%.
  - Customers with two products have the lowest churn rate among all groups, suggesting higher retention.
- **3 Products:**
  - Churn rate increases to around 60%.
  - There is a substantial increase in churn rate for customers with three products.
- **4 Products:**
  - Churn rate is the highest at around 95%.
  - Customers with four products have the highest likelihood of churning.

Overall, the churn rate decreases when customers use two products but increases sharply for those with three or four products. This pattern suggests that while having some engagement with multiple products reduces churn, a higher number of products might be associated with increased churn, potentially due to complexity or dissatisfaction with managing multiple products.

---

4. Behavioral Analysis:

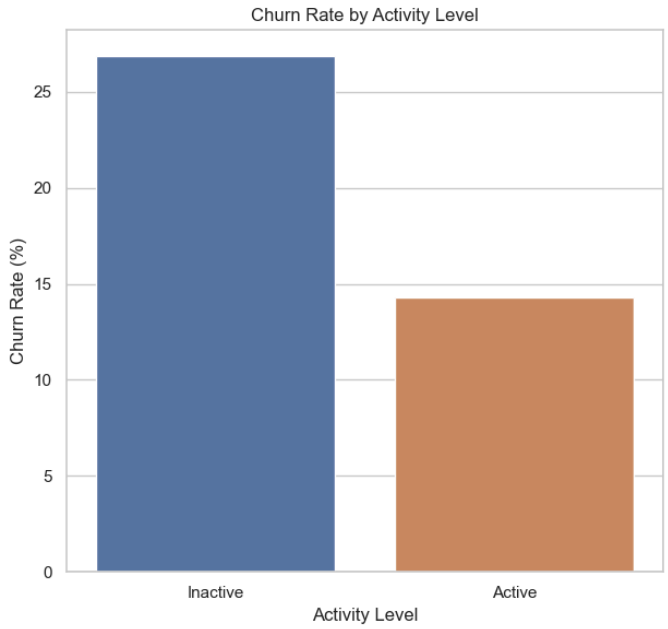
4B. Activity Level Analysis:

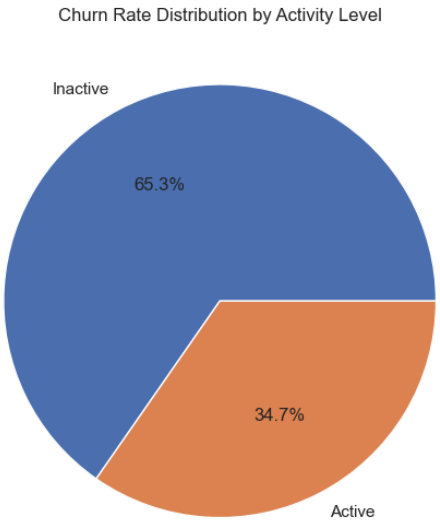
Investigating the relationship between being an IsActiveMember and customer churn.

In [36]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by IsActiveMember
activity_churn_rate = df.groupby('IsActiveMember')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Activity Level:{bold_end}")
print(activity_churn_rate)
```

```
Churn Rate by Activity Level:
IsActiveMember
No     26.871520
Yes    14.269074
Name: Exited, dtype: float64
```

In [37]:
```
# Create a DataFrame for plotting
activity_data = pd.DataFrame({
    'Activity Level': ['Inactive', 'Active'],
    'Churn Rate (%)': activity_churn_rate.values
})

# Plot the bar chart with 'deep' palette
plt.figure(figsize=(12, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=activity_data, x='Activity Level', y='Churn Rate (%)', palette='deep')
plt.title('Churn Rate by Activity Level')
plt.xlabel('Activity Level')
plt.ylabel('Churn Rate (%)')

# Pie chart
plt.subplot(1, 2, 2)
labels = activity_data['Activity Level']
sizes = activity_data['Churn Rate (%)']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('deep', len(labels)))
plt.title('Churn Rate Distribution by Activity Level')

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





Observations-

- Inactive users have a notably higher churn rate than active users.
- The majority of churned users are inactive, comprising more than two-thirds of the total churned user base.
- This suggests a strong correlation between user activity level and their likelihood of churning, with inactive users being more prone to churn.

These observations indicate that increasing user activity could be a key strategy in reducing overall churn rates.

---

5. Financial Analysis:

5A. Balance vs. Churn:

Analyzing how customer balance levels correlate with churn rates.

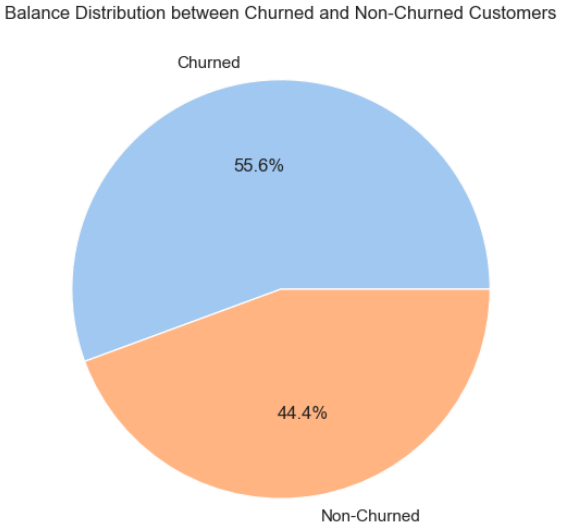
In [38]:
```
# Find the minimum balance
min_balance = df['Balance'].min()

# Find the maximum balance
max_balance = df['Balance'].max()

print(f"The minimum balance is: {min_balance}")
print(f"The maximum balance is: {max_balance}")
```

```
The minimum balance is: 0.0
The maximum balance is: 250898.09
```

In [39]:
```
churned_balance_avg = round((df[df['Exited'] == 1]['Balance'].mean() * 100),2)
print(f'Balance of Churned Customer: {churned_balance_avg }')

non_churned_balance_avg = round((df[df['Exited'] == 0]['Balance'].mean() * 100),2)
print(f'Balance of Non Churned Customer: {non_churned_balance_avg}')
```

```
Balance of Churned Customer: 9110947.6
Balance of Non Churned Customer: 7274275.07
```

In [40]:
```
# Create a DataFrame for plotting
balance_data = pd.DataFrame({
    'Customer Status': ['Churned', 'Non-Churned'],
    'Average Balance': [churned_balance_avg, non_churned_balance_avg]
})

# Plot the bar chart
plt.figure(figsize=(12, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=balance_data, x='Customer Status', y='Average Balance', palette='pastel')
plt.title('Average Balance for Churned vs. Non-Churned Customers')
plt.xlabel('Customer Status')
plt.ylabel('Average Balance')

# Pie chart
plt.subplot(1, 2, 2)
labels = balance_data['Customer Status']
sizes = balance_data['Average Balance']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('pastel', len(labels)))
plt.title('Balance Distribution between Churned and Non-Churned Customers')

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



Observations-

- Churned customers have a higher average balance compared to non-churned customers.
- The majority of the total balance is held by churned customers, comprising more than half of the total balance.
- This suggests that customers with higher balances are more likely to churn, or that when customers with higher balances churn, it significantly impacts the overall balance distribution.

These observations indicate a potential area of concern, as losing high-balance customers can have a significant financial impact. Strategies to retain high-balance customers could be critical in maintaining financial stability.

---

5. Financial Analysis:

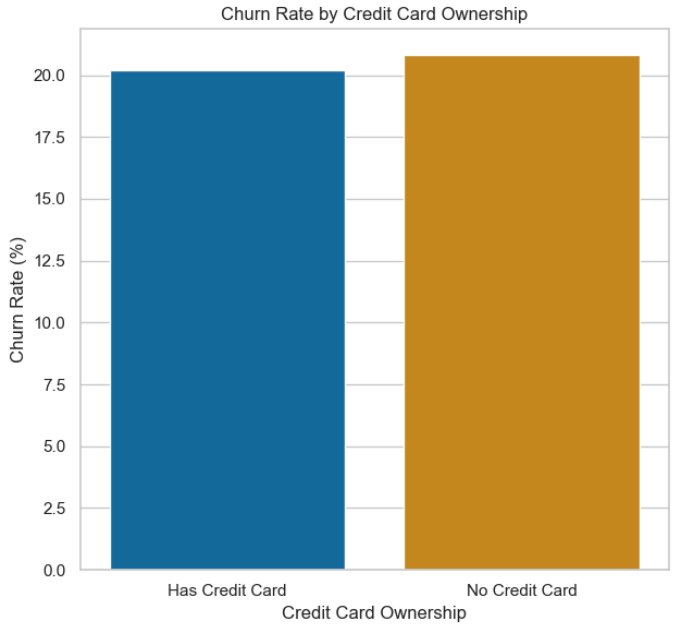
5B. Credit Card Ownership:

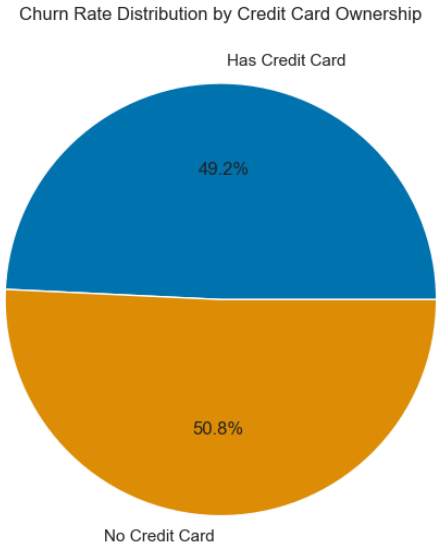
Determining if owning a credit card (HasCrCard) impacts customer loyalty.

In [41]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by HasCrCard
hascard_churn_rate = df.groupby('HasCrCard')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Credit Card Ownership:{bold_end}")
print(hascard_churn_rate)
```

```
Churn Rate by Credit Card Ownership:
HasCrCard
No     20.814941
Yes    20.198441
Name: Exited, dtype: float64
```

In [42]:
```
# Create a DataFrame for plotting
hascard_data = pd.DataFrame({
    'Credit Card Ownership': ['Has Credit Card', 'No Credit Card'],
    'Churn Rate (%)': [hascard_churn_rate[1], hascard_churn_rate[0]]
})

# Plot the bar chart with 'colorblind' palette
plt.figure(figsize=(12, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=hascard_data, x='Credit Card Ownership', y='Churn Rate (%)', palette='colorblind')
plt.title('Churn Rate by Credit Card Ownership')
plt.xlabel('Credit Card Ownership')
plt.ylabel('Churn Rate (%)')

# Pie chart
plt.subplot(1, 2, 2)
labels = hascard_data['Credit Card Ownership']
sizes = hascard_data['Churn Rate (%)']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('colorblind', len(labels)))
plt.title('Churn Rate Distribution by Credit Card Ownership')

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





Observations-

- There is a slight difference in churn rates between customers who have a credit card and those who do not, with those not having a credit card having a slightly higher churn rate.
- The distribution of churned customers is fairly even between those who have a credit card and those who do not.
- This suggests that credit card ownership does not have a significant impact on churn rate, as the churn rate and distribution are quite similar for both groups.

These observations indicate that while credit card ownership has a minor effect on churn rates, it is not a major differentiator in customer retention. Other factors might play a more significant role in influencing churn rates.

---

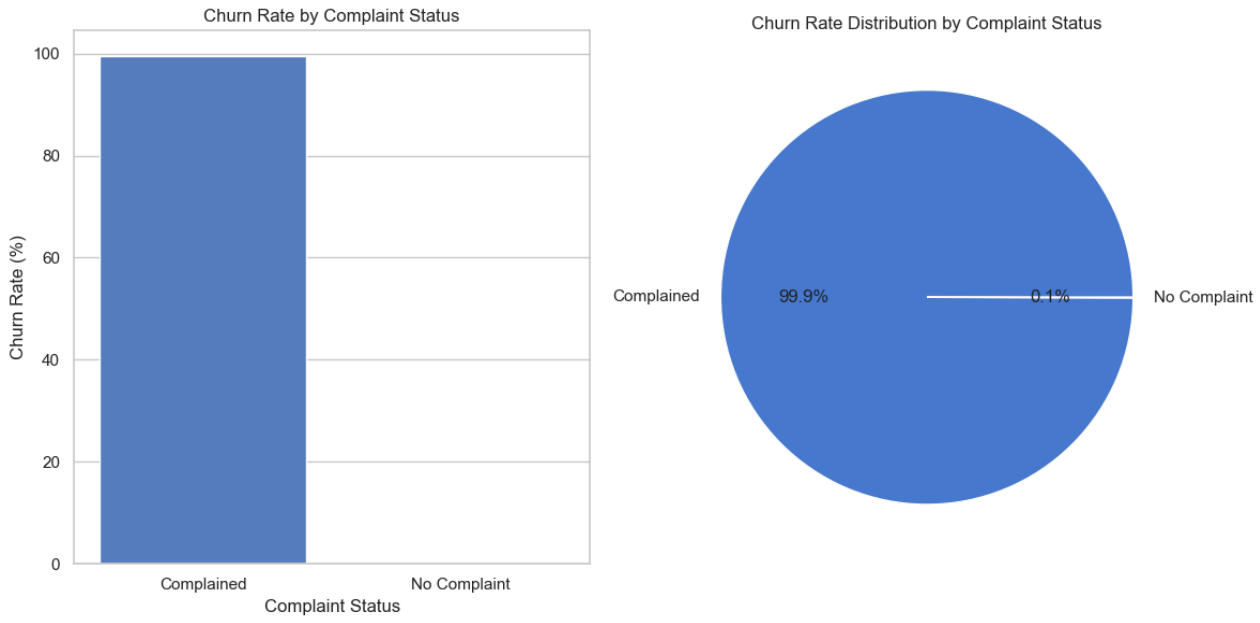
6. Customer Satisfaction and Feedback:

In [43]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by Complain
complain_churn_rate = df.groupby('Complain')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Complaint:{bold_end}")
print(complain_churn_rate)
```

Churn Rate by Complaint:

```
Complain
No      0.050277
Yes    99.510763
Name: Exited, dtype: float64
```

In [44]:
```
# Create a DataFrame for plotting
complain_data = pd.DataFrame({
    'Complaint Status': ['Complained', 'No Complaint'],
    'Churn Rate (%)': [complain_churn_rate[1], complain_churn_rate[0]]
})

# Plot the bar chart with 'muted' palette
plt.figure(figsize=(12, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=complain_data, x='Complaint Status', y='Churn Rate (%)', palette='muted')
plt.title('Churn Rate by Complaint Status')
plt.xlabel('Complaint Status')
plt.ylabel('Churn Rate (%)')

# Pie chart
plt.subplot(1, 2, 2)
labels = complain_data['Complaint Status']
sizes = complain_data['Churn Rate (%)']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('muted', len(labels)))
plt.title('Churn Rate Distribution by Complaint Status')

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



Observations-

- **High Churn Among Complainants:** Customers who file complaints have an exceptionally high churn rate, indicating dissatisfaction that likely leads to them leaving the service or product.
- **Minimal Churn Among Non-Complainants:** The churn rate among customers who do not complain is almost non-existent, suggesting they are either satisfied or at least not motivated enough to leave.
- **Disproportionate Impact:** The extremely high churn rate among complainants suggests that addressing customer complaints more effectively could significantly reduce overall churn rates.

---

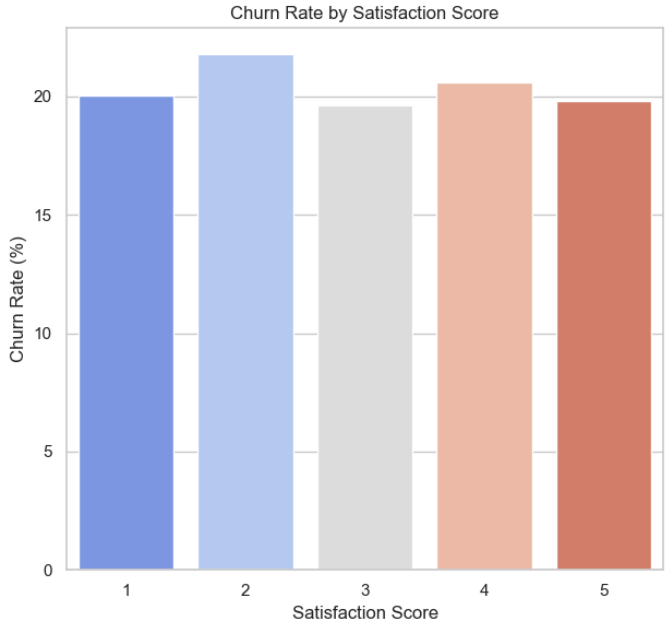
6. Customer Satisfaction and Feedback:

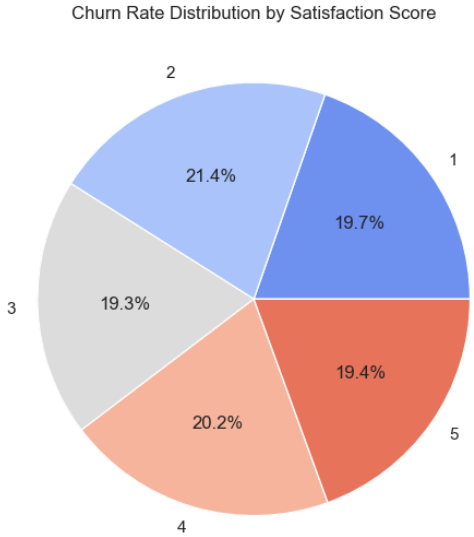
6B. Satisfaction and Churn:

In [45]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by Satisfaction Score
satisfaction_score_churn_rate = df.groupby('Satisfaction Score')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Satisfaction Score:{bold_end}")
print(satisfaction_score_churn_rate)
```

Churn Rate by Satisfaction Score:
```
Satisfaction Score
1    20.031056
2    21.797418
3    19.637610
4    20.617530
5    19.810379
Name: Exited, dtype: float64
```

In [46]:
```
# Create a DataFrame for plotting
satisfaction_score_data = pd.DataFrame({
    'Satisfaction Score': satisfaction_score_churn_rate.index,
    'Churn Rate (%)': satisfaction_score_churn_rate.values
})

# Plot the bar chart with 'coolwarm' palette
plt.figure(figsize=(12, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=satisfaction_score_data, x='Satisfaction Score', y='Churn Rate (%)', palette='coolwarm')
plt.title('Churn Rate by Satisfaction Score')
plt.xlabel('Satisfaction Score')
plt.ylabel('Churn Rate (%)')

# Pie chart
plt.subplot(1, 2, 2)
labels = satisfaction_score_data['Satisfaction Score']
sizes = satisfaction_score_data['Churn Rate (%)']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('coolwarm', len(labels)))
plt.title('Churn Rate Distribution by Satisfaction Score')

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





Observations-

- **Uniform Churn Rates:** Churn rates do not vary significantly across different satisfaction scores, suggesting that factors other than satisfaction score might be influencing churn.
- **Slight Peak at Satisfaction Score 2:** The churn rate and its distribution are slightly higher for satisfaction score 2, indicating that customers with this score might be more prone to churn compared to others.
- **Balanced Distribution:** The relatively even distribution of churn across all satisfaction scores implies that customers leave the service regardless of how satisfied or dissatisfied they are, though the small differences could be worth further investigation.

---

7. Card Usage Analysis:

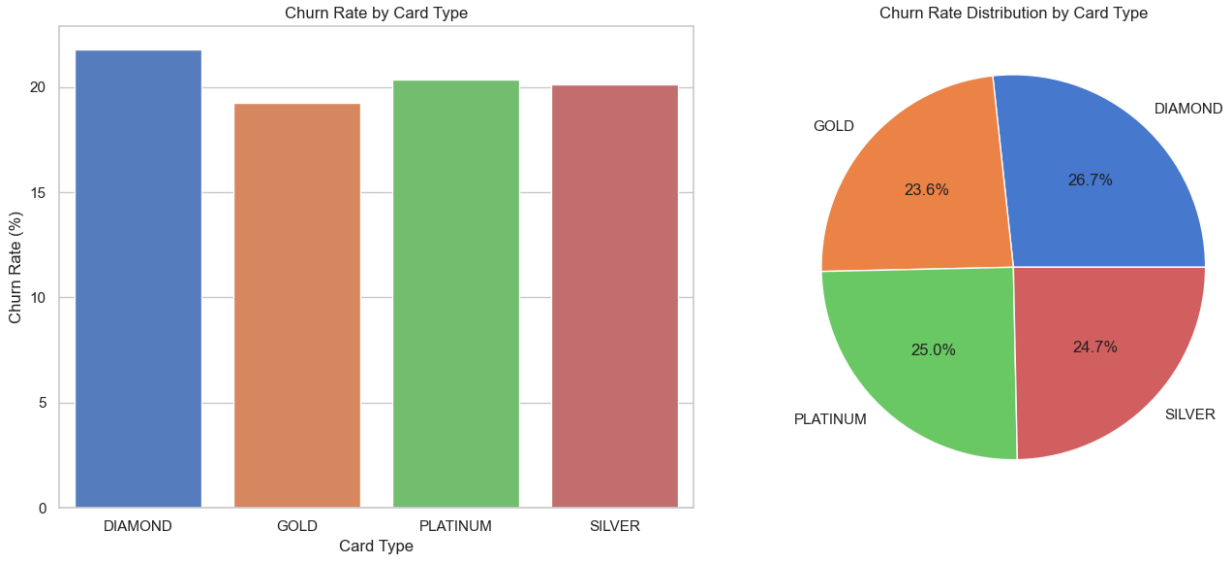
7A. Impact of Card Type on Churn:

Examining if different Card Types have different churn rates.

In [47]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by Card Type
card_type_churn_rate = df.groupby('Card Type')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Card Type:{bold_end}")
print(card_type_churn_rate)
```

```
Churn Rate by Card Type:
Card Type
DIAMOND     21.779019
GOLD        19.264588
PLATINUM    20.360721
SILVER      20.112179
Name: Exited, dtype: float64
```

In [48]:
```
# Create a DataFrame for plotting
card_type_score_data = pd.DataFrame({
    'Card Type': card_type_churn_rate.index,
    'Churn Rate (%)': card_type_churn_rate.values
})

# Plot the bar chart using Matplotlib and Seaborn
plt.figure(figsize=(14, 6))

# Bar chart
plt.subplot(1, 2, 1)
sns.barplot(data=card_type_score_data, x='Card Type', y='Churn Rate (%)', palette='muted')
plt.title('Churn Rate by Card Type')
plt.xlabel('Card Type')
plt.ylabel('Churn Rate (%)')

# Pie chart
plt.subplot(1, 2, 2)
labels = card_type_score_data['Card Type']
sizes = card_type_score_data['Churn Rate (%)']
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=sns.color_palette('muted', len(labels)))
plt.title('Churn Rate Distribution by Card Type')

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



Observations-

- **Slight Variation in Churn Rates:** Churn rates among different card types are quite similar, suggesting that card type does not have a strong influence on churn rate.
- **Higher Churn Among Diamond Card Holders:** Diamond card holders exhibit the highest churn rate and the largest distribution of churn, indicating potential dissatisfaction or other issues within this segment.
- **Balanced Distribution:** The distribution of churn among card types is relatively balanced, with no single card type dominating the churn rate. This implies that churn occurs across all card types fairly evenly.

---

7. Card Usage Analysis:

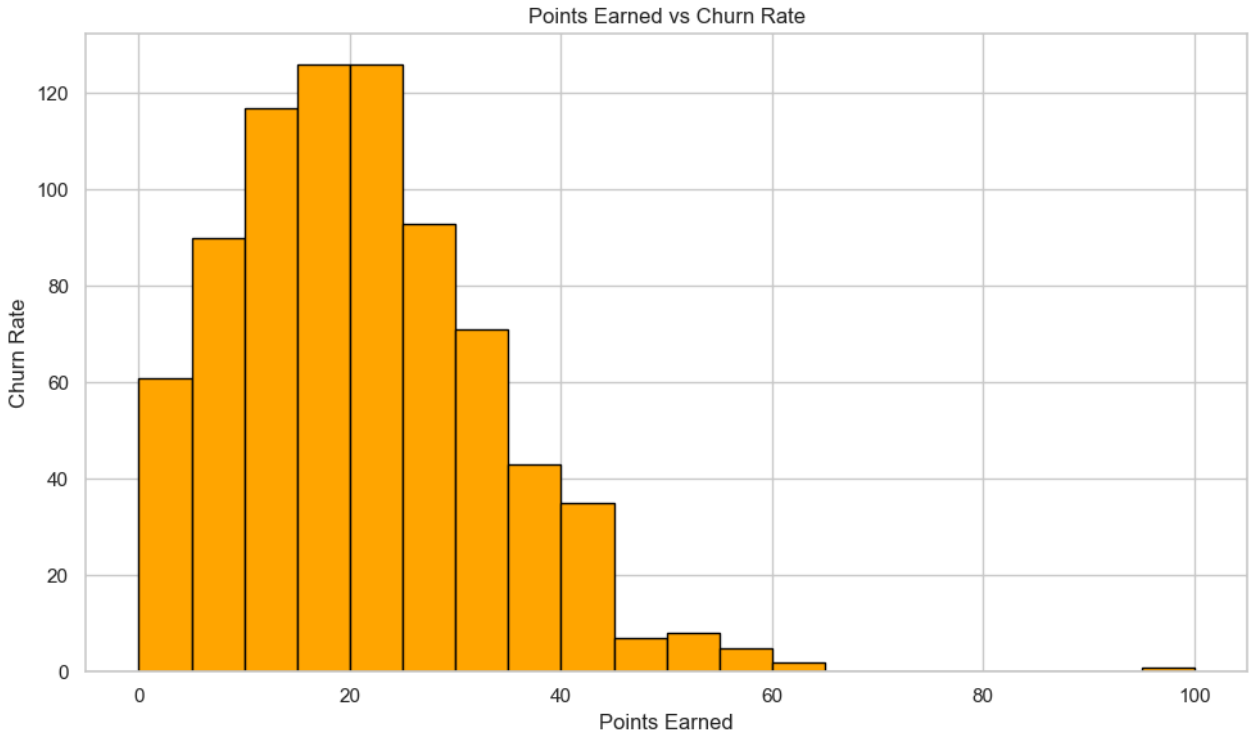
7B. Loyalty Points Analysis:

Investigating whether Points Earned from credit card usage influence customer retention.

In [49]:
```
# ANSI escape code for bold text
bold_start = "\033[1m"
bold_end = "\033[0m"

# Calculate churn rate by Points Earned
point_earned_churn_rate = df.groupby('Point Earned')['Exited'].mean() * 100
print(f"{bold_start}Churn Rate by Points Earned:{bold_end}")
print(point_earned_churn_rate)
```

```
Churn Rate by Points Earned:
Point Earned
119       0.000000
163     100.000000
206       0.000000
219      37.500000
220      14.285714
            ...
996       0.000000
997      26.666667
998       8.333333
999      14.285714
1000     23.076923
Name: Exited, Length: 785, dtype: float64
```

In [50]:
```
# Plot histogram
plt.figure(figsize=(10, 6))
plt.hist(point_earned_churn_rate, bins=20, color='orange', edgecolor='black')
plt.title('Points Earned vs Churn Rate')
plt.xlabel('Points Earned')
plt.ylabel('Churn Rate')
plt.grid(True)

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



Observations-

The histogram you provided shows the distribution of points earned versus churn rate. Here are some observations based on the histogram:

- **Peak Frequency:** The highest frequency of points earned falls in the range of 15-20, with a churn rate of around 120.
- **Skewness:** The distribution is right-skewed, with a majority of the data concentrated on the left side (lower points earned) and a long tail extending towards the right (higher points earned).
- **Churn Rate Decline:** As the points earned increase, the churn rate generally decreases, indicating that higher points earned are associated with lower churn rates.
- **Outliers:** There are some outliers in the higher points earned ranges (60-100), which have very low frequencies compared to the rest of the distribution.
- **Concentration:** The majority of the data points are concentrated between 0 to 40 points earned, where the churn rate is higher compared to the higher points earned range.

These observations suggest that customers earning fewer points tend to have a higher churn rate, while those earning more points have a lower churn rate, though fewer customers earn high points.

8. Salary Analysis:

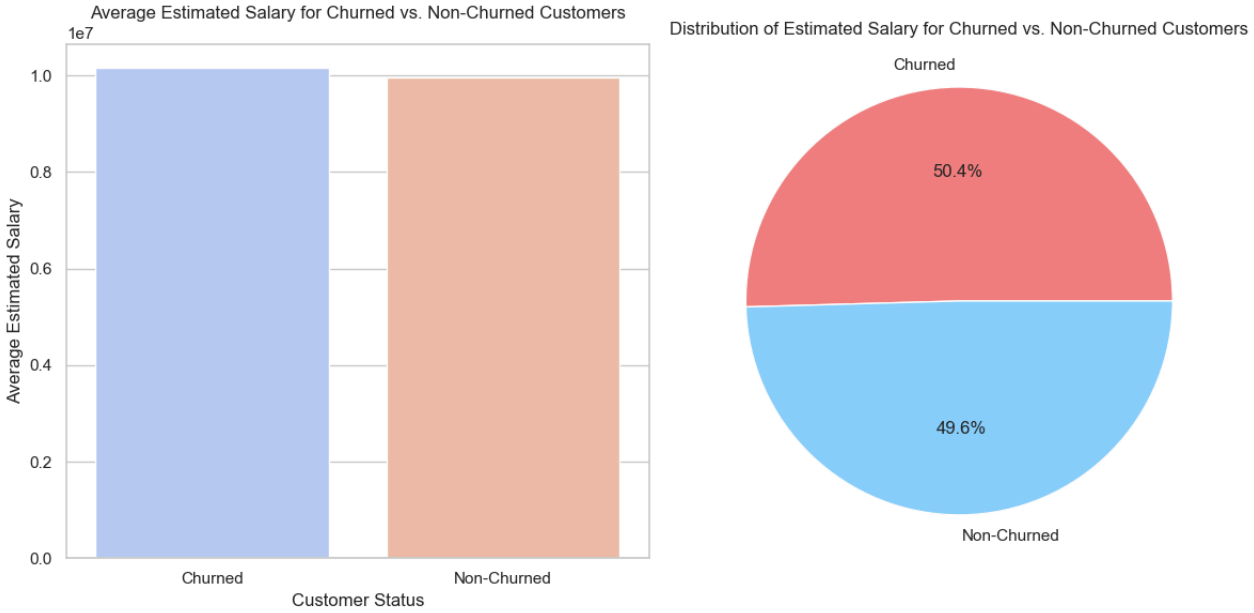
8A. Salary and Churn:

Analyzing the relationship between EstimatedSalary and customer churn, focusing on how financial well-being might influence churn decisions.

In [51]:
```
estimated_salary_churned = round((df[df['Exited'] == 1]['EstimatedSalary'].mean() * 100),2)
print(f'Estimated Salary of Churned Customer: {estimated_salary_churned}')

estimated_salary_non_churned = round((df[df['Exited'] == 0]['EstimatedSalary'].mean() * 100),2)
print(f'Estimated Salary of Non Churned Customer: {estimated_salary_non_churned}')
```

```
Estimated Salary of Churned Customer: 10150990.88
Estimated Salary of Non Churned Customer: 9972685.31
```

In [52]:
```
# Create a DataFrame for plotting
salary_data = pd.DataFrame({
    'Customer Status': ['Churned', 'Non-Churned'],
    'Average Estimated Salary': [estimated_salary_churned, estimated_salary_non_churned]
})

# Plotting both bar chart and pie chart together
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 6))

# Bar chart
sns.barplot(data=salary_data, x='Customer Status', y='Average Estimated Salary', ax=axes[0], palette='coolwarm')
axes[0].set_title('Average Estimated Salary for Churned vs. Non-Churned Customers')
axes[0].set_xlabel('Customer Status')
axes[0].set_ylabel('Average Estimated Salary')

# Pie chart
labels = ['Churned', 'Non-Churned']
sizes = [estimated_salary_churned, estimated_salary_non_churned]
axes[1].pie(sizes, labels=labels, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue'])
axes[1].set_title('Distribution of Estimated Salary for Churned vs. Non-Churned Customers')
axes[1].axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

# Adjust layout
plt.tight_layout()

# Show plots
plt.show()
```



Observations-

- **Salary Not a Strong Indicator:** Both visualizations suggest that estimated salary might not be a strong indicator of whether a customer will churn, as there is no substantial difference in average salaries and the proportions of churned vs. non-churned customers are nearly equal.
- **Potential Other Factors:** Other factors, such as customer service, product satisfaction, or usage patterns, might be more significant in determining customer churn.

Overall, these visualizations indicate that estimated salary alone is not a significant differentiator for customer churn in this dataset.

9. Hypothesis Testing:

9A. Satisfaction Score Vs Churn:

- **Null Hypothesis:** There is no significant difference in satisfaction score of a customer and the customer exiting the bank.
- **Alternative Hypothesis:** There is a significant difference in satisfaction score of a customer and the customer exiting the bank.

In [53]:
```
# Create a contingency table (cross-tabulation) of Geography vs. Exited
contingency_table = pd.crosstab(df['Satisfaction Score'], df['Exited'])

# Perform chi-square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Print results
print('Chi-square Test Results:')
print(f'Chi-square statistic: {chi2:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference in satisfaction score of a customer and the customer exiting the bank.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference in satisfaction score of a customer and the customer exiting the bank.')
```

```
Chi-square Test Results:
Chi-square statistic: 3.8027
P-value: 0.4334

Since p-value (0.4334) >= alpha (0.05), we fail to reject the null hypothesis.
There is no significant difference in satisfaction score of a customer and the customer exiting the bank.
```

Observation-

There is no significant difference in satisfaction score of a customer and the customer exiting the bank.

---

9. Hypothesis Testing:

9B. Complaint Vs Churn:

- **Null Hypothesis:** There is no significant difference in Complaint of a customer and the customer exiting the bank.
- **Alternative Hypothesis:** There is a significant difference in Complaint of a customer and the customer exiting the bank.

In [54]:
```
# Create a contingency table (cross-tabulation) of Geography vs. Exited
contingency_table = pd.crosstab(df['Complain'], df['Exited'])

# Perform chi-square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Print results
print('Chi-square Test Results:')
print(f'Chi-square statistic: {chi2:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference in Complaint of a customer and the customer exiting the bank.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference in Complaint of a customer and the customer exiting the bank.')
```

```
Chi-square Test Results:
Chi-square statistic: 9907.9070
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is a significant difference in Complaint of a customer and the customer exiting the bank.
```

Observation-

There is a significant difference in Complaint of a customer and the customer exiting the bank.

---

9. Hypothesis Testing:

9C. Geography vs Customer Churn:

- **Null Hypothesis:** There is no association between the Geography and the customers exiting the bank.
- **Alternative Hypothesis:** There is an association between the Geography and the customers exiting the bank.

In [55]:
```
# Create a contingency table (cross-tabulation) of Geography vs. Exited
contingency_table = pd.crosstab(df['Geography'], df['Exited'])

# Perform chi-square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Print results
print('Chi-square Test Results:')
print(f'Chi-square statistic: {chi2:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is an association between the Geography and the customers exiting the bank.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no association between the Geography and the customers exiting the bank.')
```

```
Chi-square Test Results:
Chi-square statistic: 300.6264
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is an association between the Geography and the customers exiting the bank.
```

Observation-

There is an association between the Geography and the customers exiting the bank.

---

9. Hypothesis Testing:

9D. Gender Vs Customer Churn:

- **Null Hypothesis:** There is no association between the Gender and the Customer exiting the bank.
- **Alternative Hypothesis:** These is an association between the Gender and the Customer exiting the bank.

In [56]:
```
# Create a contingency table (cross-tabulation) of Gender vs. Exited
contingency_table = pd.crosstab(df['Gender'], df['Exited'])

# Perform chi-square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Print results
print('Chi-square Test Results:')
print(f'Chi-square statistic: {chi2:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('These is an association between the Gender and the Customer exiting the bank.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no association between the Gender and the Customer exiting the bank.')
```

```
Chi-square Test Results:
Chi-square statistic: 112.3966
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
These is an association between the Gender and the Customer exiting the bank.
```

Observation-

These is an association between the Gender and the Customer exiting the bank.

---

9. Hypothesis Testing:

9E. Number Of Products Vs Customer Churn:

- **Null Hypothesis:** There is no significant difference between the Number of products bought by a customer and the customer exiting the bank.
- **Alternative Hypothesis:** There is a significant difference between the Number of products bought by a customer and the customer exiting the bank.

In [57]:
```
# Separate NumOfProducts for churned and non-churned customers
num_products_churned = df[df['Exited'] == 1]['NumOfProducts']
num_products_non_churned = df[df['Exited'] == 0]['NumOfProducts']

# Perform Mann-Whitney U test
statistic, p_value = mannwhitneyu(num_products_churned, num_products_non_churned)

# Print results
print('Mann-Whitney U Test Results:')
print(f'U statistic: {statistic:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference between the Number of products bought by a customer and the customer exiting
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference between the Number of products bought by a customer and the customer exitir
```

```
Mann-Whitney U Test Results:
U statistic: 6835967.0000
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is a significant difference between the Number of products bought by a customer and the customer exiting the ban
k.
```

Observation-

There is a significant difference between the Number of products bought by a customer and the customer exiting the bank.

---

9. Hypothesis Testing:

9F. IsActiveMember Vs Customer Churn:

- **Null Hypothesis:** There is no association betweeen Active Customer and the customer exiting the bank.
- **Alternative Hypothesis:** There is an association betweeen Active Customer and the customer exiting the bank.

In [58]:
```
# Create a contingency table (cross-tabulation) of IsActiveMember vs. Exited
contingency_table = pd.crosstab(df['IsActiveMember'], df['Exited'])

# Perform chi-square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Print results
print('Chi-square Test Results:')
print(f'Chi-square statistic: {chi2:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is an association betweeen Active Customer and the customer exiting the bank.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no association betweeen Active Customer and the customer exiting the bank.')
```

```
Chi-square Test Results:
Chi-square statistic: 243.6948
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is an association betweeen Active Customer and the customer exiting the bank.
```

Observation-

There is an association betweeen Active Customer and the customer exiting the bank.

---

9. Hypothesis Testing:

9G. Credit Score Vs Customer Churn:

- **Null Hypothesis:** There is no significant difference between the mean of credit score who exited the bank and not exited the bank.
- **Alternative Hypothesis:** There is significant difference between the mean of credit score who exited the bank and not exited the bank.

In [59]:
```
# Separate data into two groups based on Exited status
exited = df[df['Exited'] == 1]['CreditScore']
not_exited = df[df['Exited'] == 0]['CreditScore']

# Perform two-sample t-test
t_statistic, p_value = ttest_ind(exited, not_exited)

# Print results
print('Two-Sample T-Test Results:')
print(f'T-statistic: {t_statistic:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference in mean credit score between customers who exited and those who did not.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference in mean credit score between customers who exited and those who did not.')
```

```
Two-Sample T-Test Results:
T-statistic: -2.6778
P-value: 0.0074

Since p-value (0.0074) < alpha (0.05), we reject the null hypothesis.
There is a significant difference in mean credit score between customers who exited and those who did not.
```

Observation-

---

9. Hypothesis Testing:

9H. Age Vs Customer Churn:

- **Null Hypothesis:** There is no significant difference between the mean age of the customer who exited and not exited.
- **Alternative Hypothesis:** There is significant difference between the mean age of the customer who exited and not exited.

In [60]:
```
# Separate data into two groups based on Exited status
exited = df[df['Exited'] == 1]['Age']
not_exited = df[df['Exited'] == 0]['Age']

# Perform two-sample t-test
t_statistic, p_value = ttest_ind(exited, not_exited)

# Print results
print('Two-Sample T-Test Results:')
print(f'T-statistic: {t_statistic:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference in mean age between customers who exited and those who did not.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference in mean age between customers who exited and those who did not.')
```

```
Two-Sample T-Test Results:
T-statistic: 29.7638
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is a significant difference in mean age between customers who exited and those who did not.
```

Observation-

There is significant difference between the mean age of the customer who exited and not exited.

---

9. Hypothesis Testing:

9I. Balance Vs Customer Churn:

- **Null Hypothesis:** There is no significant difference between the mean balance of the customer who exited and not exited.
- **Alternative Hypothesis:** There is significant difference between the mean balance of the customer who exited and not exited.

In [61]:
```
# Separate data into two groups based on Exited status
exited = df[df['Exited'] == 1]['Balance']
not_exited = df[df['Exited'] == 0]['Balance']

# Perform two-sample t-test
t_statistic, p_value = ttest_ind(exited, not_exited)

# Print results
print('Two-Sample T-Test Results:')
print(f'T-statistic: {t_statistic:.4f}')
print(f'P-value: {p_value:.4f}')

# Set significance level
alpha = 0.05

# Interpret the results
if p_value < alpha:
    print(f'\nSince p-value ({p_value:.4f}) < alpha ({alpha}), we reject the null hypothesis.')
    print('There is a significant difference in mean balance between customers who exited and those who did not.')
else:
    print(f'\nSince p-value ({p_value:.4f}) >= alpha ({alpha}), we fail to reject the null hypothesis.')
    print('There is no significant difference in mean balance between customers who exited and those who did not.')
```

```
Two-Sample T-Test Results:
T-statistic: 11.9407
P-value: 0.0000

Since p-value (0.0000) < alpha (0.05), we reject the null hypothesis.
There is a significant difference in mean balance between customers who exited and those who did not.
```

Observation-

There is significant difference between the mean balance of the customer who exited and not exited.

---

10. Actionable Insights and Recommendations to retain users:

1. **Expand Marketing Efforts in Germany and Spain**: Since 50% of customers are from France, focus marketing campaigns on Germany and Spain to boost customer acquisition in these regions.

2. **Develop Targeted Offers for Female Customers**: Introduce specific products or offers aimed at attracting more female customers to balance the customer demographics.

3. **Enhance After-Sales Service**: Address the fact that almost 99% of customers who filed complaints have left the bank by significantly improving the after-sales service experience.

4. **Create Retention Strategies for Multi-Product Holders**: Implement targeted retention strategies for customers with three or more products, as they have a higher churn rate.

5. **Engage Zero Balance Account Holders**: Investigate why approximately 3,000 accounts have zero balance and develop offers or incentives to engage these customers and encourage account usage.

6. **Financial Counseling for At-Risk Customers**: Analyze factors influencing customer exit versus retention and offer financial counseling to customers in vulnerable salary brackets to reduce churn.

7. **Product Optimization:** Analyze usage patterns of users with multiple products and customize offerings to better meet their needs and preferences. Conduct user surveys to gather feedback and identify areas for product improvement or feature development.

8. **Localized Marketing Campaigns:** Develop region-specific marketing strategies tailored to address unique needs identified through geographical analysis. Utilize localized events or cultural insights to create targeted campaigns aimed at retaining users in high-churn regions.

9. **Personalized Engagement:** Implement personalized incentives or loyalty rewards based on demographic, financial, and activity-related characteristics to encourage user retention. Offer tailored financial management tools or advisory services to users with low balances or credit scores to enhance engagement with the service.

10. **Community Engagement Initiatives:** Foster a sense of community among users through localized events, forums, or online communities, particularly in regions with high churn rates. Encourage user participation and feedback to better understand regional challenges and adapt retention strategies accordingly.

11. **Continuous Monitoring and Adaptation:** Regularly monitor churn metrics and conduct ongoing analysis to identify changing trends or patterns. Maintain open communication channels with users to address concerns promptly and demonstrate a commitment to customer satisfaction.

12. **Localized Customer Support:** Establish dedicated customer support teams or resources for users in high-churn regions to provide personalized assistance and address issues effectively.