



# **Exploratory Data Analysis (EDA)**

Understanding Customer Behaviour and Churn

**Mina Michel**



# What is EDA?



- EDA helps uncover **patterns, relationships, and trends** in the data.
- Critical for understanding which **features** influence customer churn.
- Sets the **foundation** for building predictive models.





## 1. Load and Explore the Dataset

We begin by loading the dataset to get an initial understanding of its structure and the nature of the data.

- Inspect the data to understand its structure, types, and completeness.
  - Summarize key statistics of the dataset (mean, median, missing values).
- 
- 

# 1. Load and Explore the Dataset

```
# Load the dataset
import pandas as pd
data = pd.read_csv('Customer Churn.csv')
data.head()
```

|   | Call Failure | Complains | Subscription Length | Charge Amount | Seconds of Use | Frequency of use | Frequency of SMS | Distinct Called Numbers | Age Group | Tariff Plan | Status | Age | Customer Value | Churn |
|---|--------------|-----------|---------------------|---------------|----------------|------------------|------------------|-------------------------|-----------|-------------|--------|-----|----------------|-------|
| 0 | 8            | 0         | 38                  | 0             | 4370           | 71               | 5                | 17                      | 3         | 1           | 1      | 30  | 197.640        | 0     |
| 1 | 0            | 0         | 39                  | 0             | 318            | 5                | 7                | 4                       | 2         | 1           | 2      | 25  | 46.035         | 0     |
| 2 | 10           | 0         | 37                  | 0             | 2453           | 60               | 359              | 24                      | 3         | 1           | 1      | 30  | 1536.520       | 0     |
| 3 | 10           | 0         | 38                  | 0             | 4198           | 66               | 1                | 35                      | 1         | 1           | 1      | 15  | 240.020        | 0     |
| 4 | 3            | 0         | 38                  | 0             | 2393           | 58               | 2                | 33                      | 1         | 1           | 1      | 15  | 145.805        | 0     |

# 1. Load and Explore the Dataset

- The dataset contains **3150** rows and **14** columns.
- No missing values.
- Most columns are **integers** except for Customer Value, is a float.
- The **Churn** column is the **target** variable for **prediction**.

```
data.info() # Check for data types and missing values
data.describe() # Summary statistics
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 14 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Call Failure                         3150 non-null   int64
 1   Complains                            3150 non-null   int64
 2   Subscription Length                  3150 non-null   int64
 3   Charge Amount                       3150 non-null   int64
 4   Seconds of Use                      3150 non-null   int64
 5   Frequency of use                    3150 non-null   int64
 6   Frequency of SMS                    3150 non-null   int64
 7   Distinct Called Numbers             3150 non-null   int64
 8   Age Group                          3150 non-null   int64
 9   Tariff Plan                        3150 non-null   int64
10   Status                             3150 non-null   int64
11   Age                                3150 non-null   int64
12   Customer Value                     3150 non-null   float64
13   Churn                             3150 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 344.7 KB
```

# 1. Load and Explore the Dataset

|       | Call Failure | Complains   | Subscription Length | Charge Amount | Seconds of Use | Frequency of use | Frequency of SMS | Distinct Called Numbers | Age Group   |
|-------|--------------|-------------|---------------------|---------------|----------------|------------------|------------------|-------------------------|-------------|
| count | 3150.000000  | 3150.000000 | 3150.000000         | 3150.000000   | 3150.000000    | 3150.000000      | 3150.000000      | 3150.000000             | 3150.000000 |
| mean  | 7.627937     | 0.076508    | 32.541905           | 0.942857      | 4472.459683    | 69.460635        | 73.174921        | 23.509841               | 2.826032    |
| std   | 7.263886     | 0.265851    | 8.573482            | 1.521072      | 4197.908687    | 57.413308        | 112.237560       | 17.217337               | 0.892555    |
| min   | 0.000000     | 0.000000    | 3.000000            | 0.000000      | 0.000000       | 0.000000         | 0.000000         | 0.000000                | 1.000000    |
| 25%   | 1.000000     | 0.000000    | 30.000000           | 0.000000      | 1391.250000    | 27.000000        | 6.000000         | 10.000000               | 2.000000    |
| 50%   | 6.000000     | 0.000000    | 35.000000           | 0.000000      | 2990.000000    | 54.000000        | 21.000000        | 21.000000               | 3.000000    |
| 75%   | 12.000000    | 0.000000    | 38.000000           | 1.000000      | 6478.250000    | 95.000000        | 87.000000        | 34.000000               | 3.000000    |
| max   | 36.000000    | 1.000000    | 47.000000           | 10.000000     | 17090.000000   | 255.000000       | 522.000000       | 97.000000               | 5.000000    |

## 2. Data Cleaning and Preparation

```
# Check for missing values
print("Missing values in each column:\n", data.isnull().sum())

# Check for duplicate rows
duplicate_rows = data.duplicated().sum()
print(f"\nNumber of duplicate rows: {duplicate_rows}")
```

Missing values in each column:

|                         |   |
|-------------------------|---|
| Call Failure            | 0 |
| Complains               | 0 |
| Subscription Length     | 0 |
| Charge Amount           | 0 |
| Seconds of Use          | 0 |
| Frequency of use        | 0 |
| Frequency of SMS        | 0 |
| Distinct Called Numbers | 0 |
| Age Group               | 0 |
| Tariff Plan             | 0 |
| Status                  | 0 |
| Age                     | 0 |
| Customer Value          | 0 |
| Churn                   | 0 |

dtype: int64

Number of duplicate rows: 300

➤ The data has **no missing values** but has **300 duplicate rows**.

## 2. Data Cleaning and Preparation

```
# Remove duplicate rows
data_cleaned = data.drop_duplicates()

# Verify that duplicates are removed
print(f'Number of rows after removing duplicates: {data_cleaned.shape[0]}')
```

Number of rows after removing duplicates: 2850

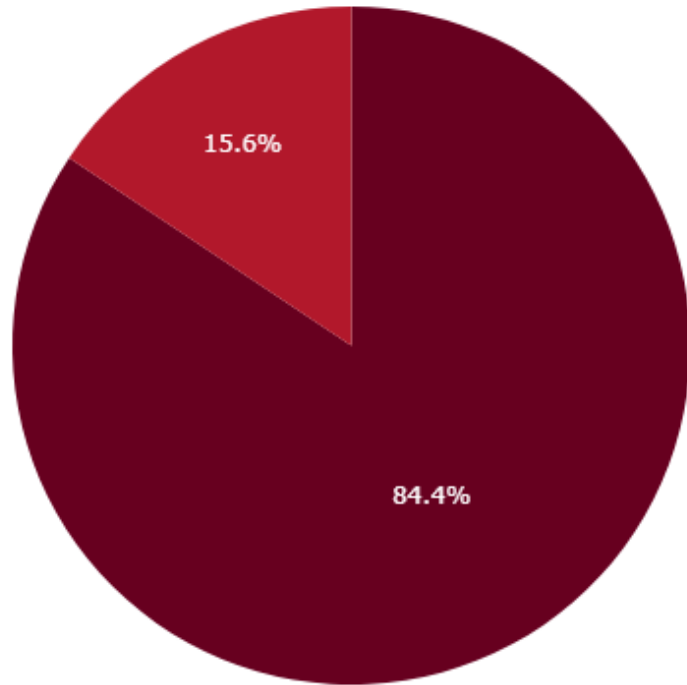
```
# Clean column names (remove extra spaces and strip them)
data_cleaned.columns = data_cleaned.columns.str.replace(' ', ' ').str.strip()

# Confirm cleaned column names
print("Cleaned column names:", data_cleaned.columns)
```

Cleaned column names: Index(['Call Failure', 'Complains', 'Subscription Length', 'Charge Amount', 'Seconds of Use', 'Frequency of use', 'Frequency of SMS', 'Distinct Called Numbers', 'Age Group', 'Tariff Plan', 'Status', 'Age', 'Customer Value', 'Churn'], dtype='object')



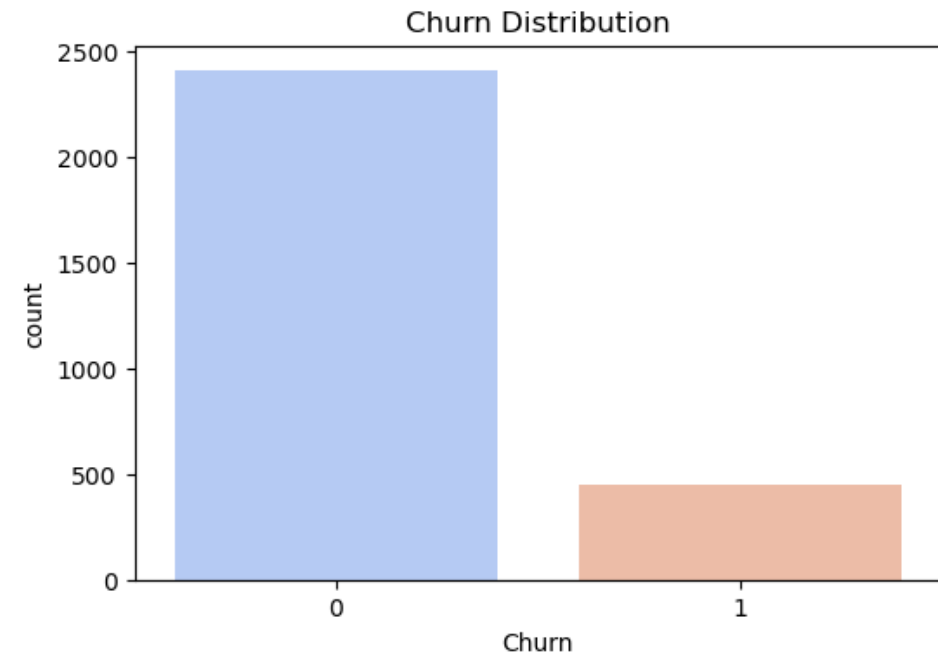
## Churn Rate Distribution:



- Class **imbalance** observed, which could affect model performance.

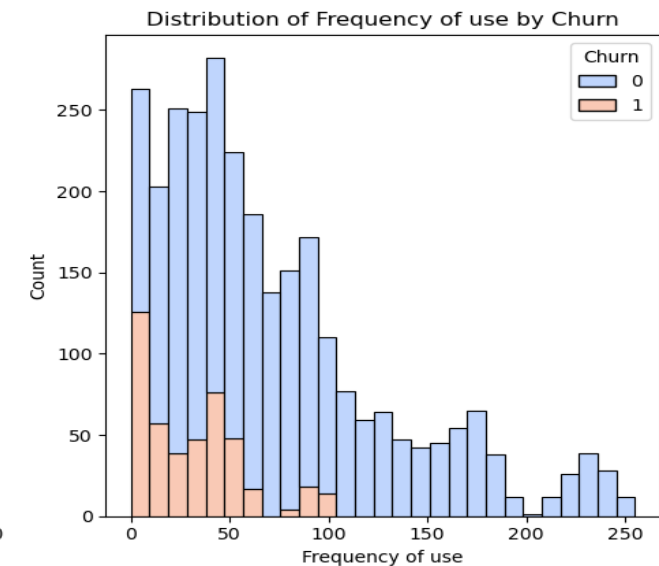
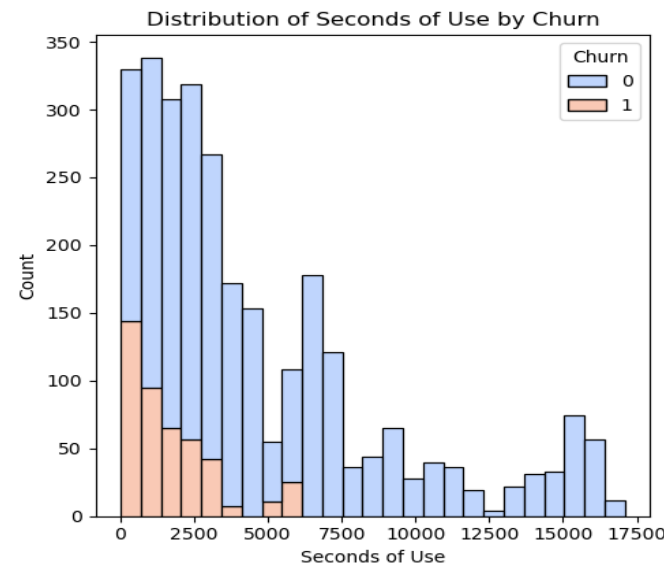
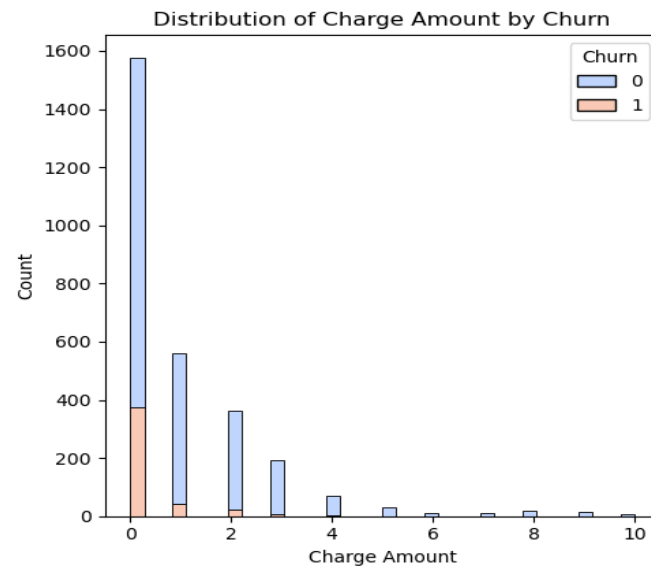
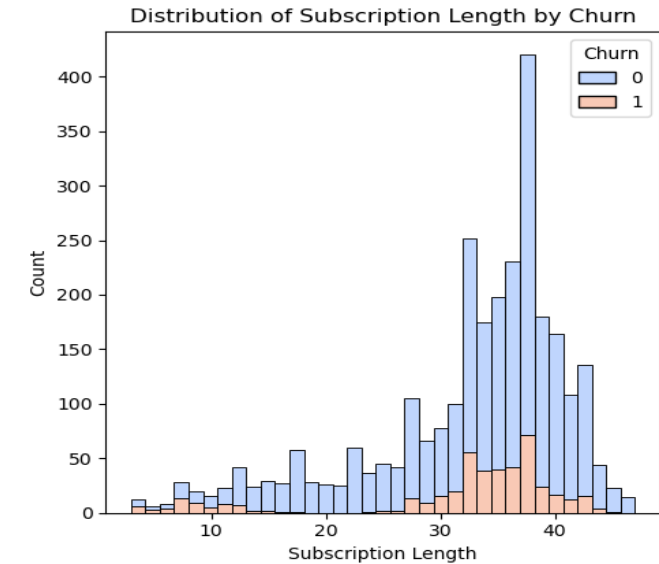
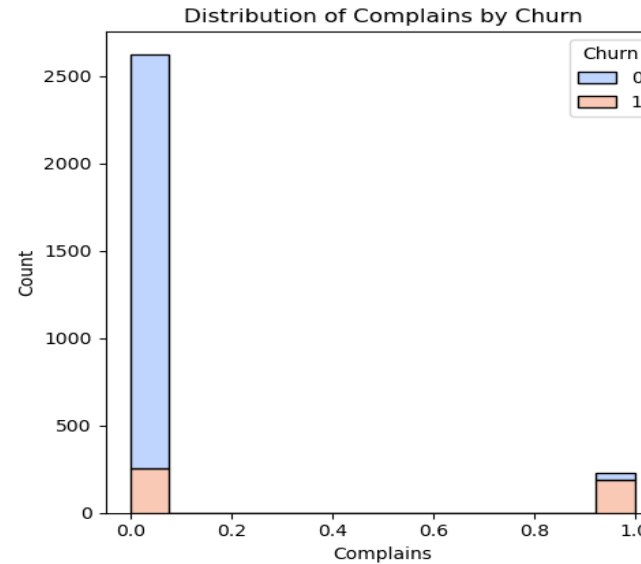
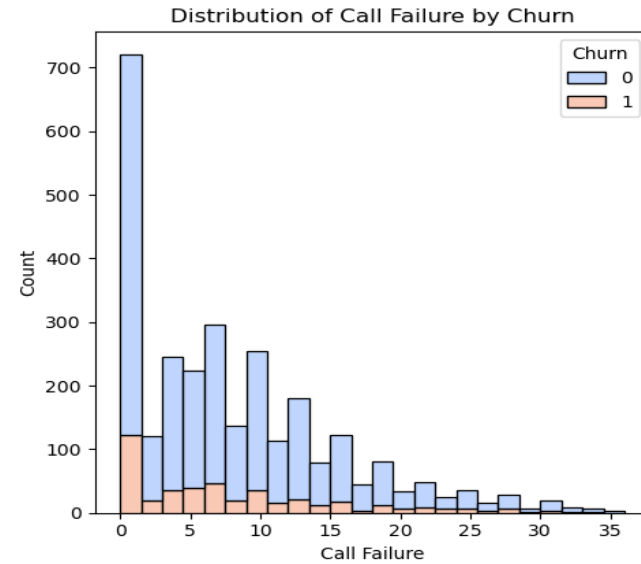
■ No Churn  
■ Churn

- **84.4%** of customers did not churn, while **15.6%** did churn.

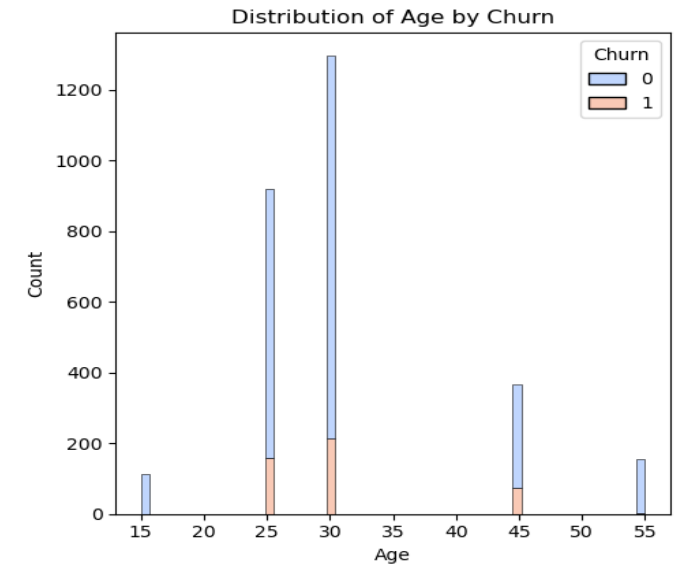
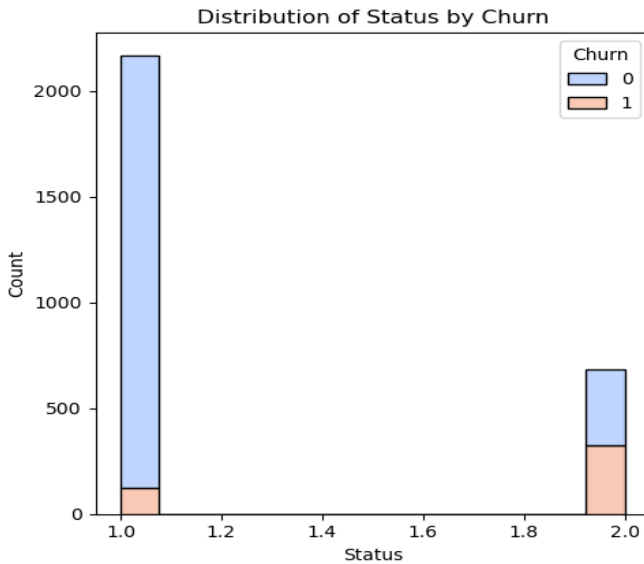
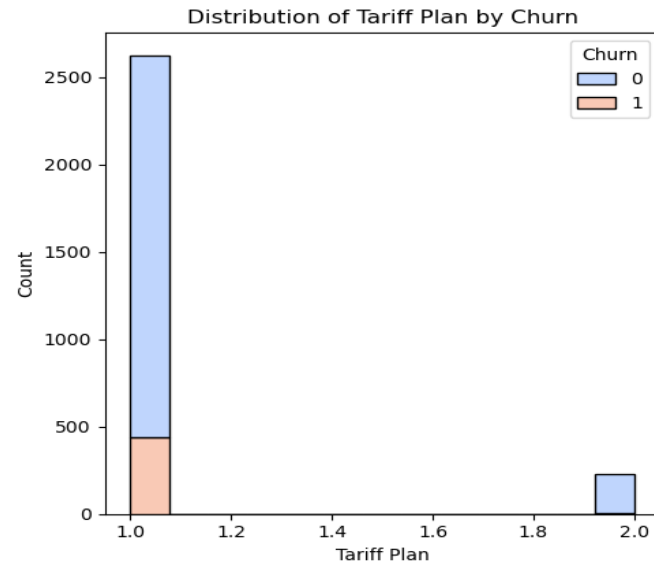
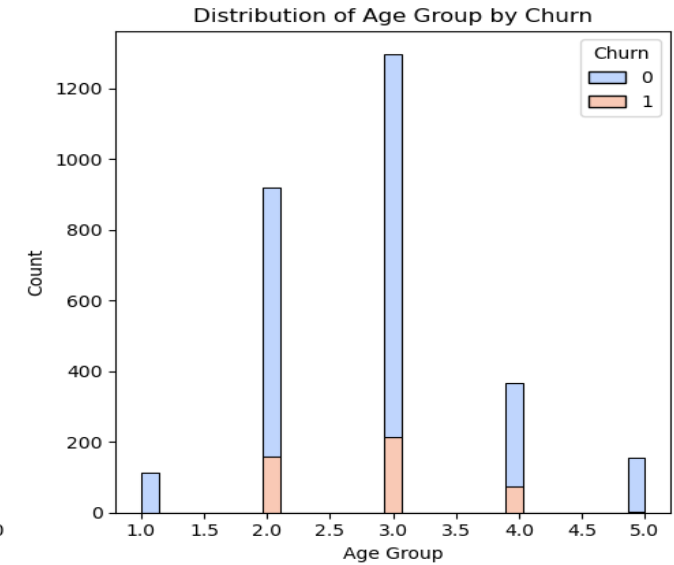
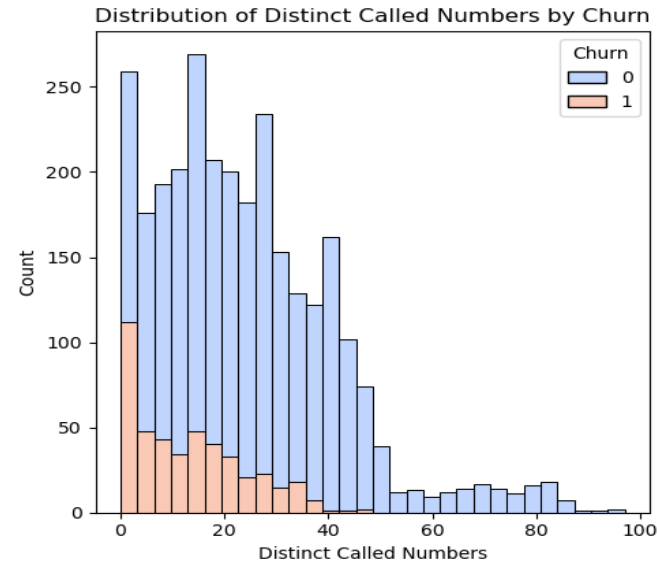
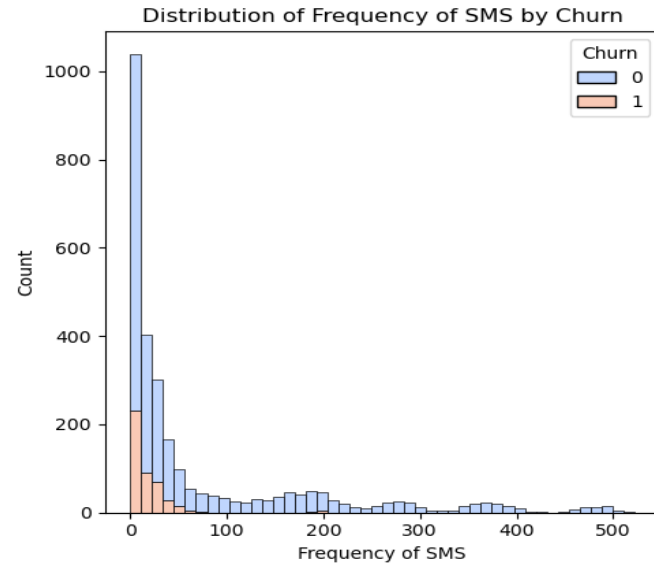


Number of Non-Churned Customers (0): **2404**  
Number of Churned Customers (1): **446**

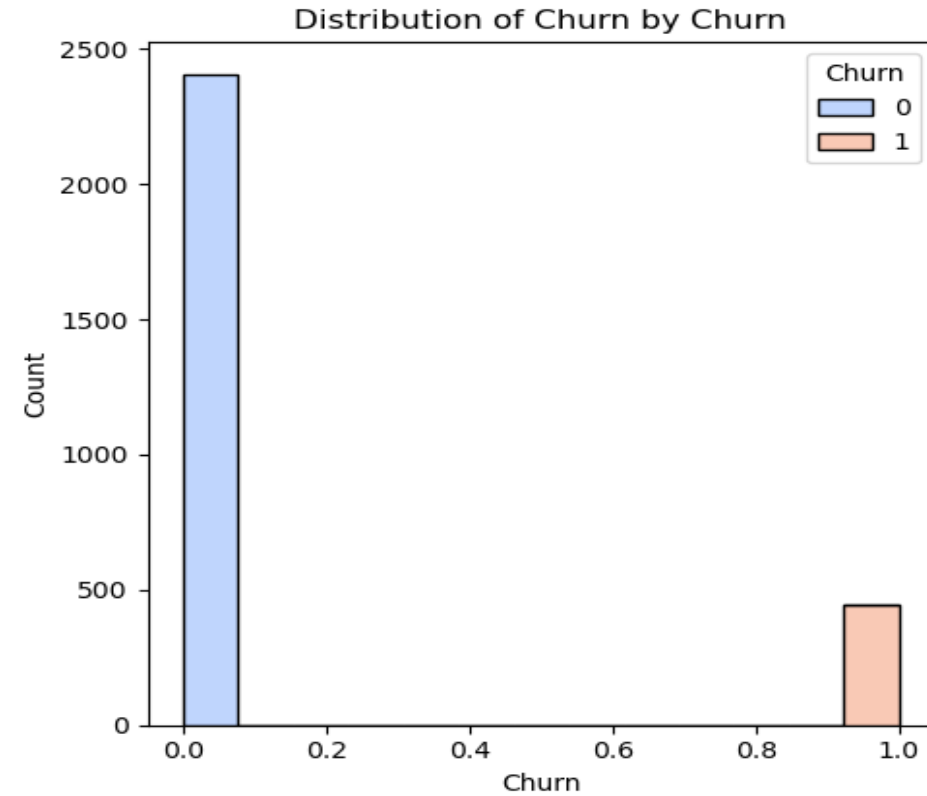
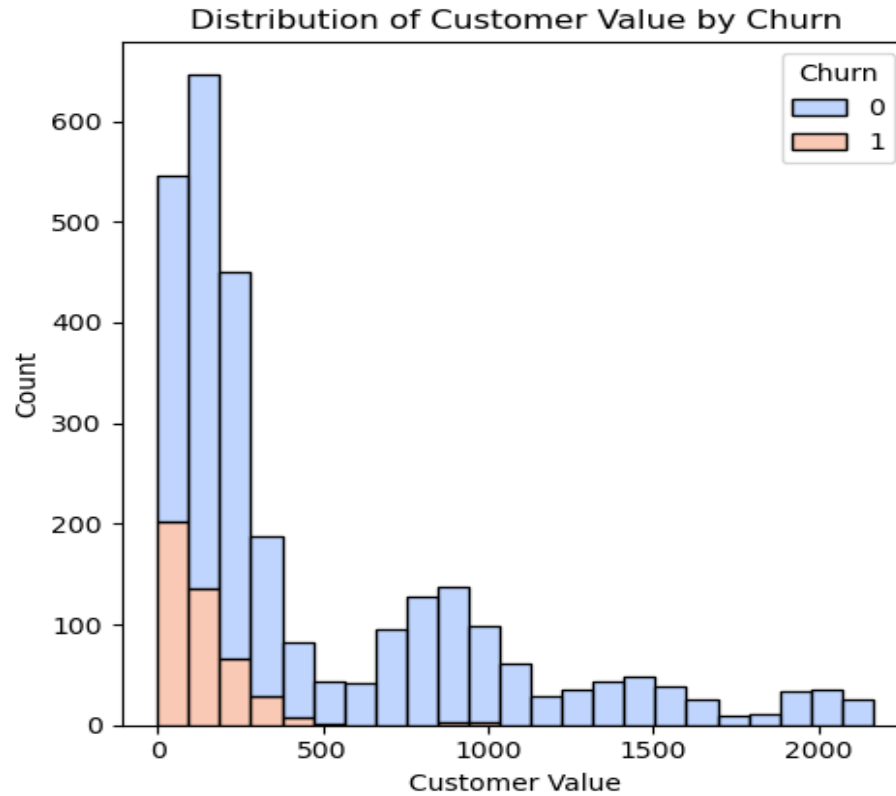
# Histograms for Distributions



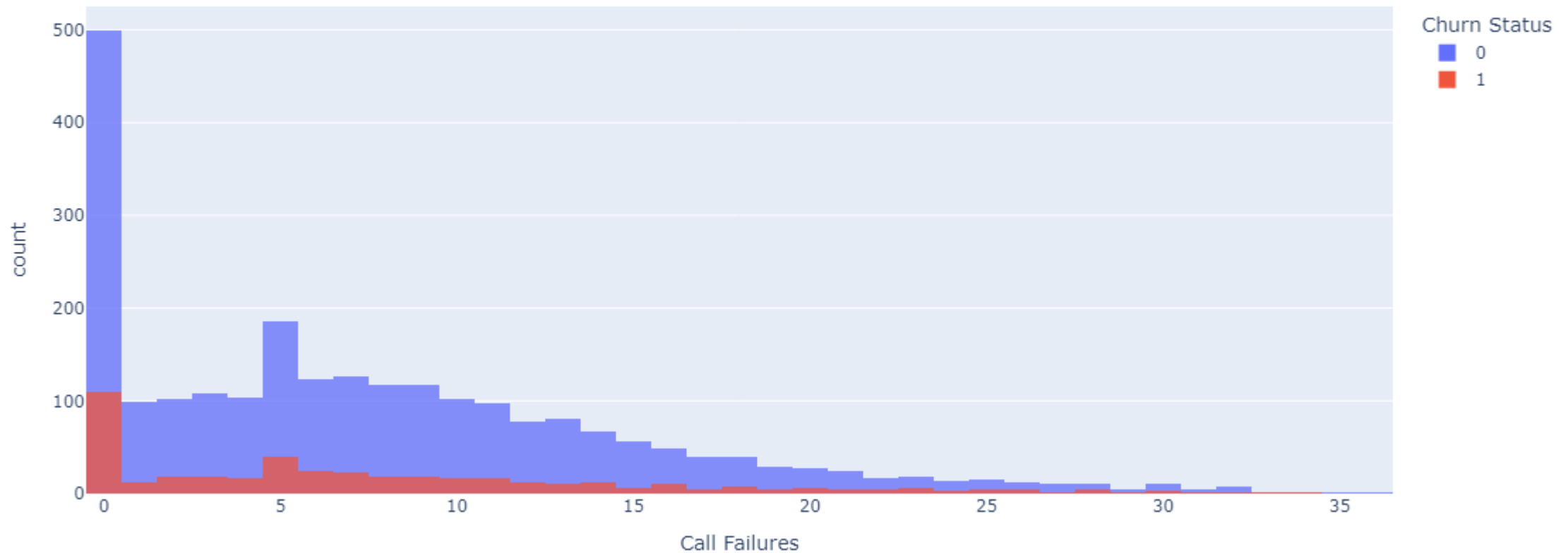
# Histograms for Distributions



# Histograms for Distributions

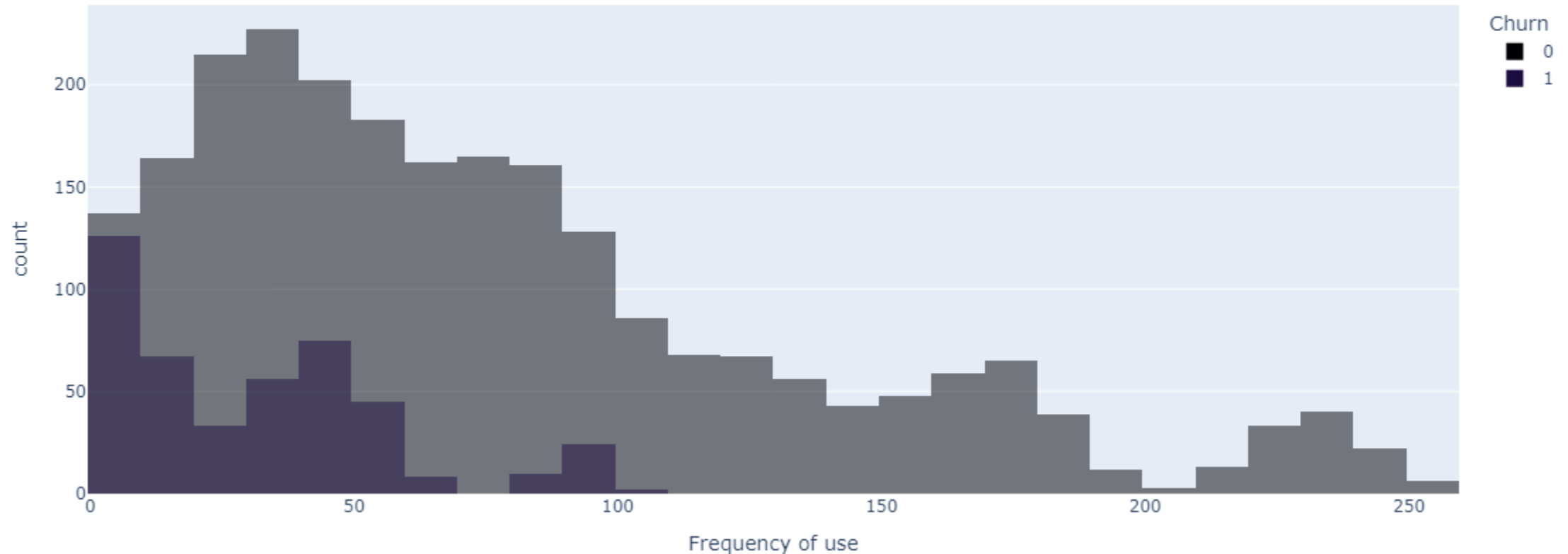


Call Failure Distribution by Churn Status



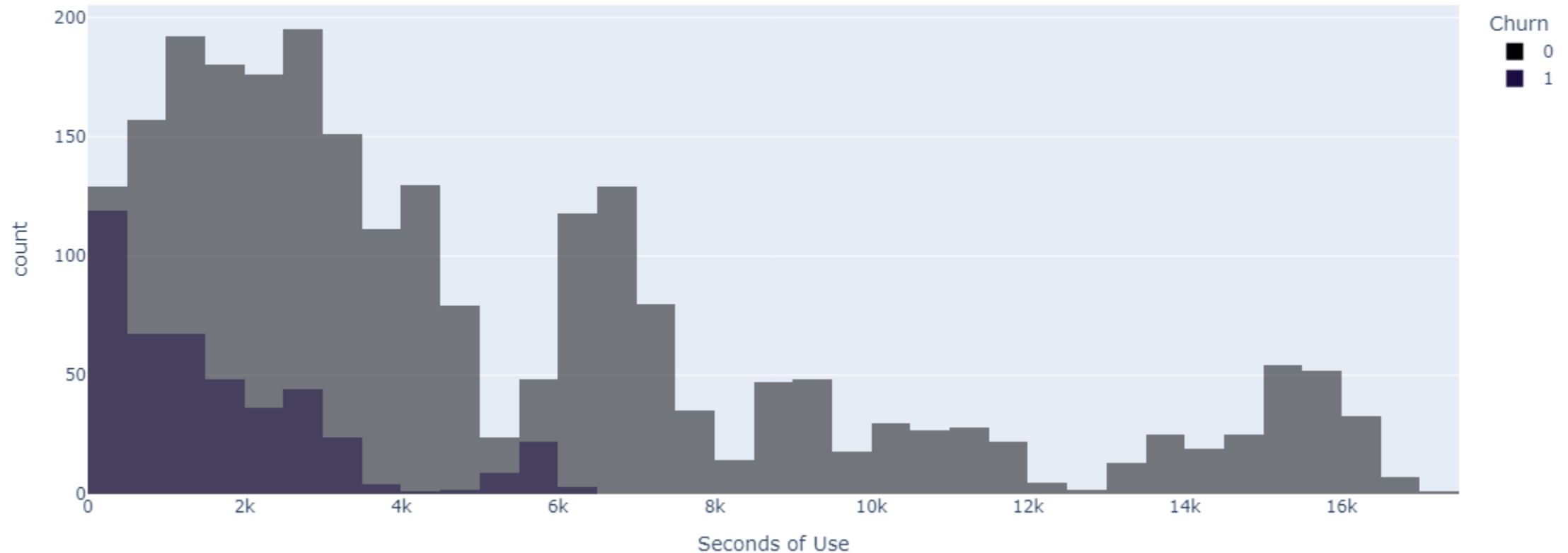
- A higher number of call failures correlates with increased churn rates.

Frequency of Use by Churn Status



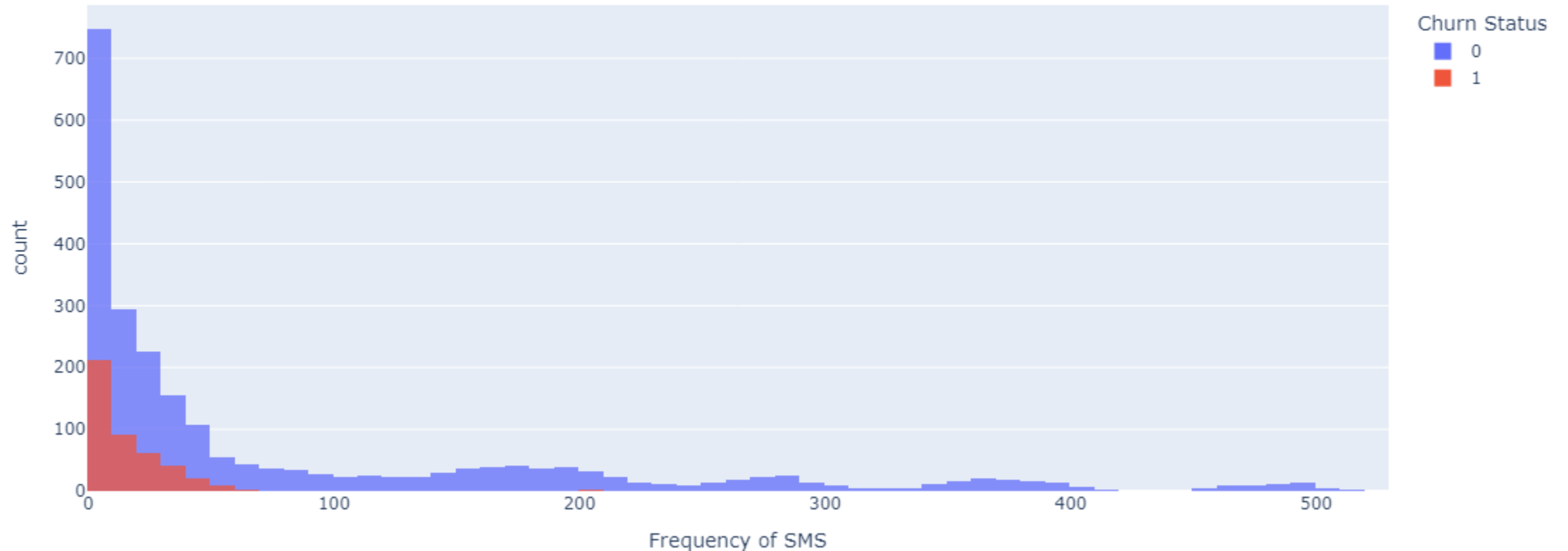
- Customers with *fewer* **service interactions** are more likely to **churn**. *Higher* frequency users are generally **retained**.

Seconds of Use by Churn Status



➤ **Churners** generally spend *less time on the service*. *Longer engagement* correlates with **retention**.

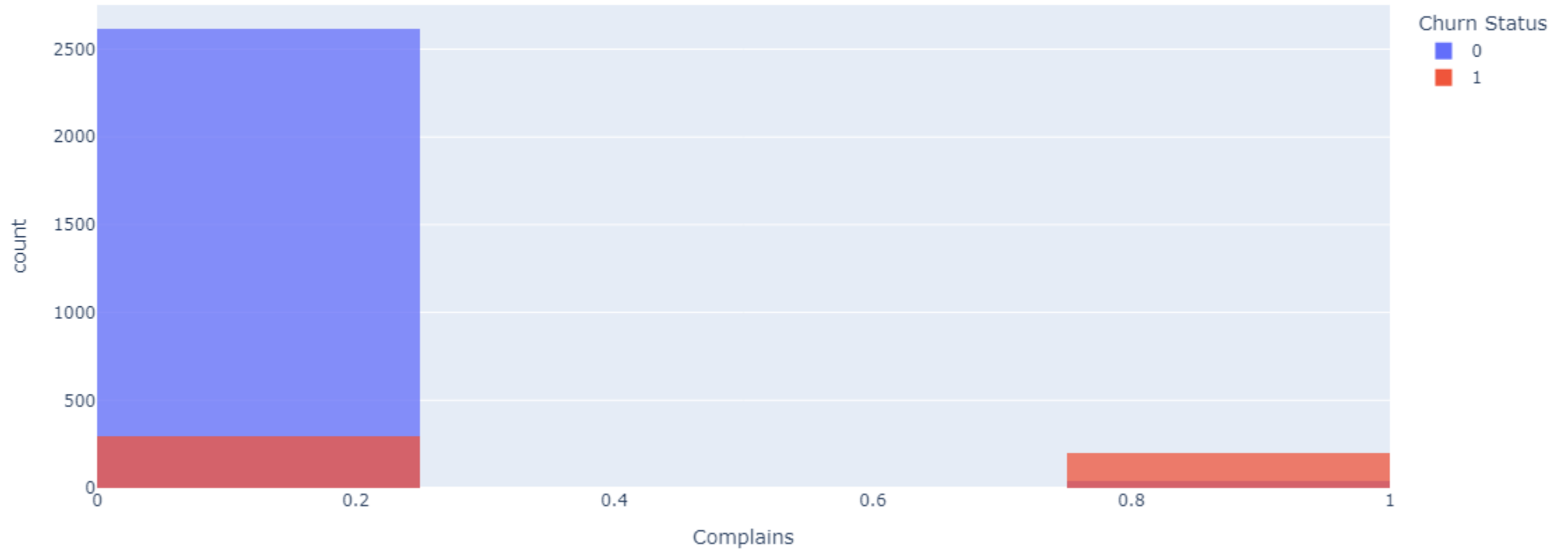
Frequency of SMS Distribution by Churn Status



- Customers with *fewer service interactions* are more likely to **churn**. *Higher* frequency users are generally **retained**.

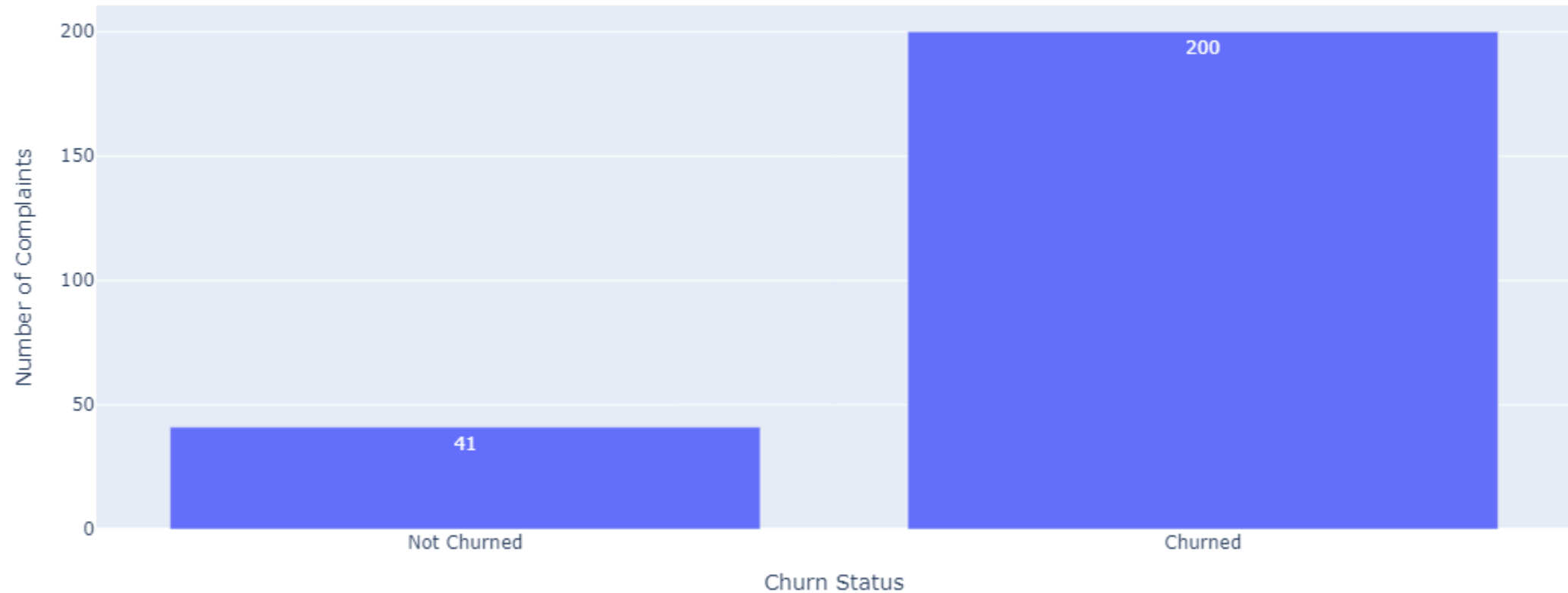


Complains Distribution by Churn Status



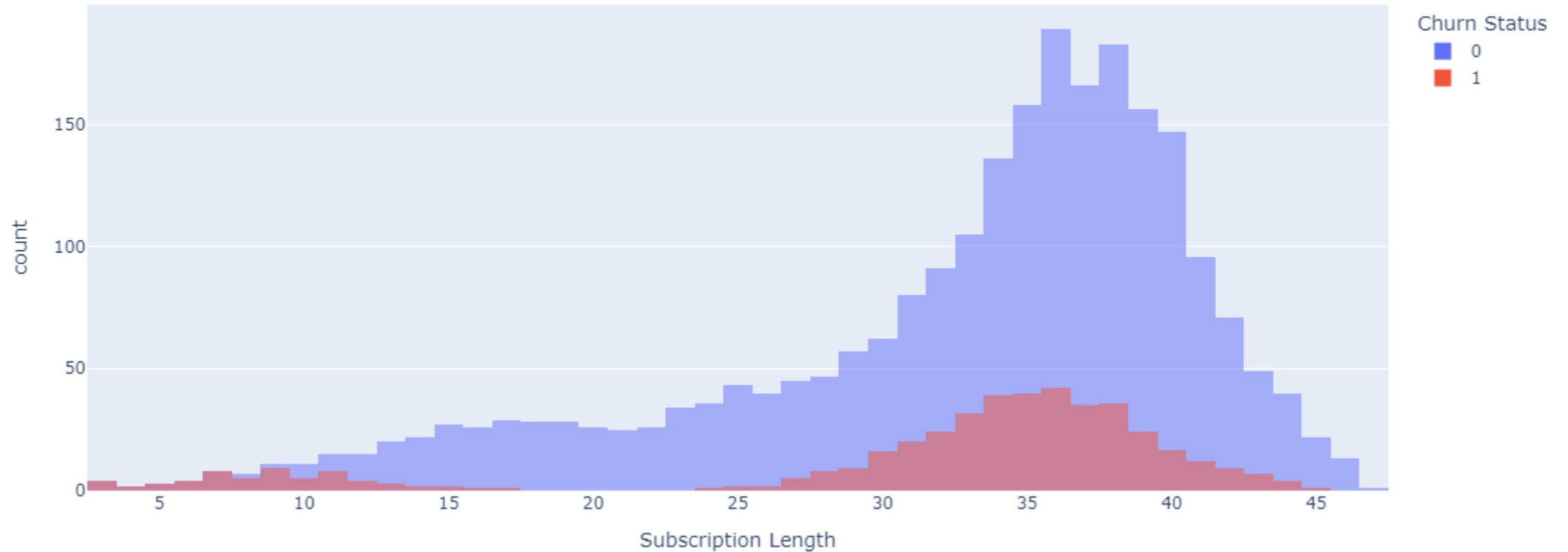
➤ Customers who register more complaints are more likely to churn.

Complains Distribution by Churn Status



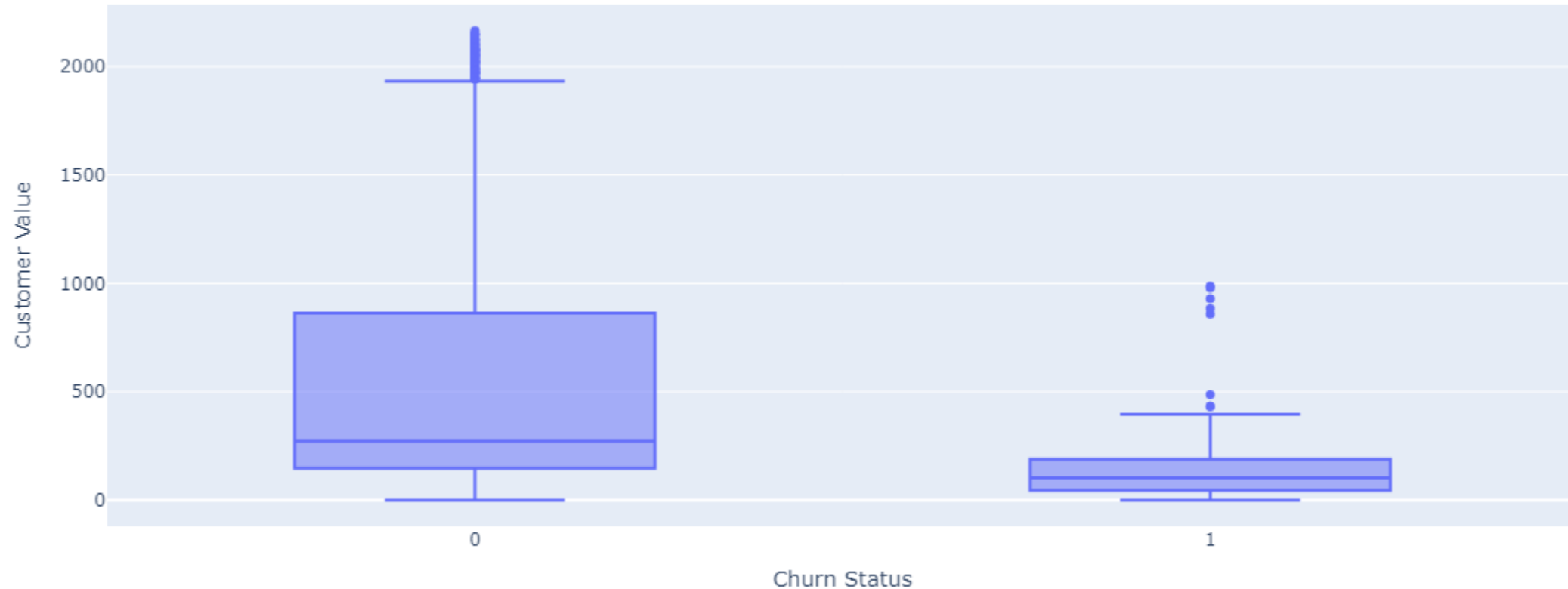
➤ Customers who register more complaints are more likely to churn.

Subscription Length Distribution by Churn Status



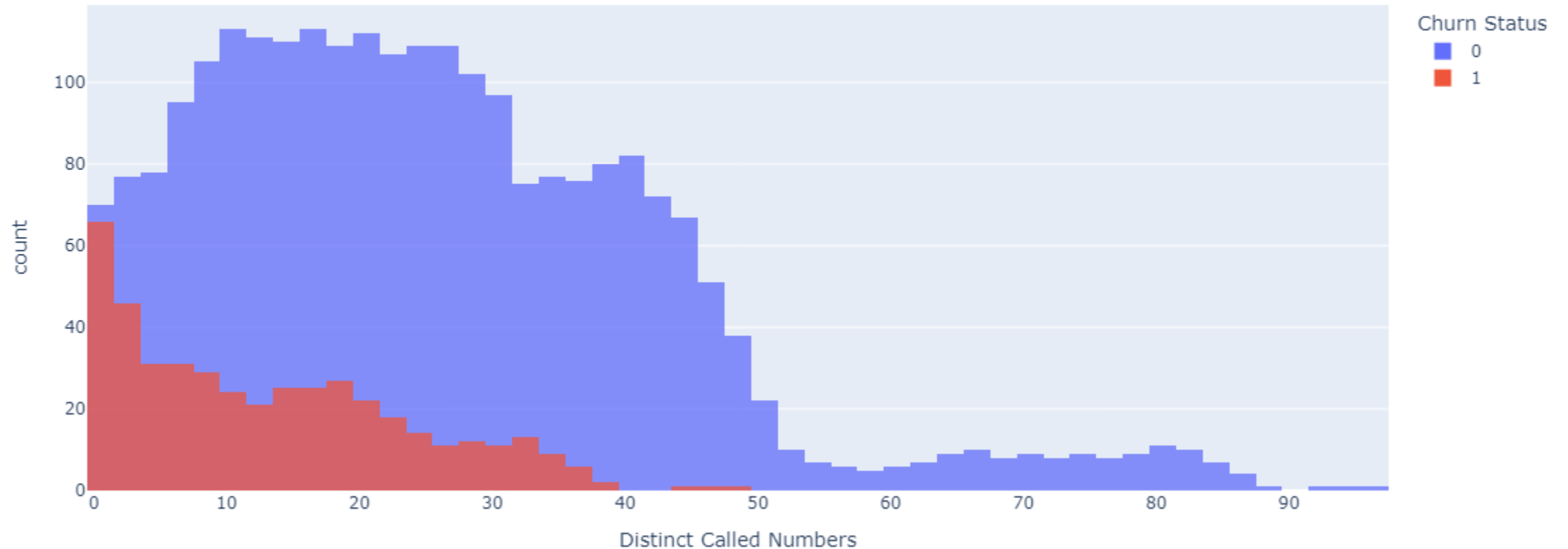
➤ Longer subscription lengths are associated with lower churn rates.

Customer Value Distribution by Churn Status



- Non-churned customers have a higher overall customer value distribution compared to churned customers.

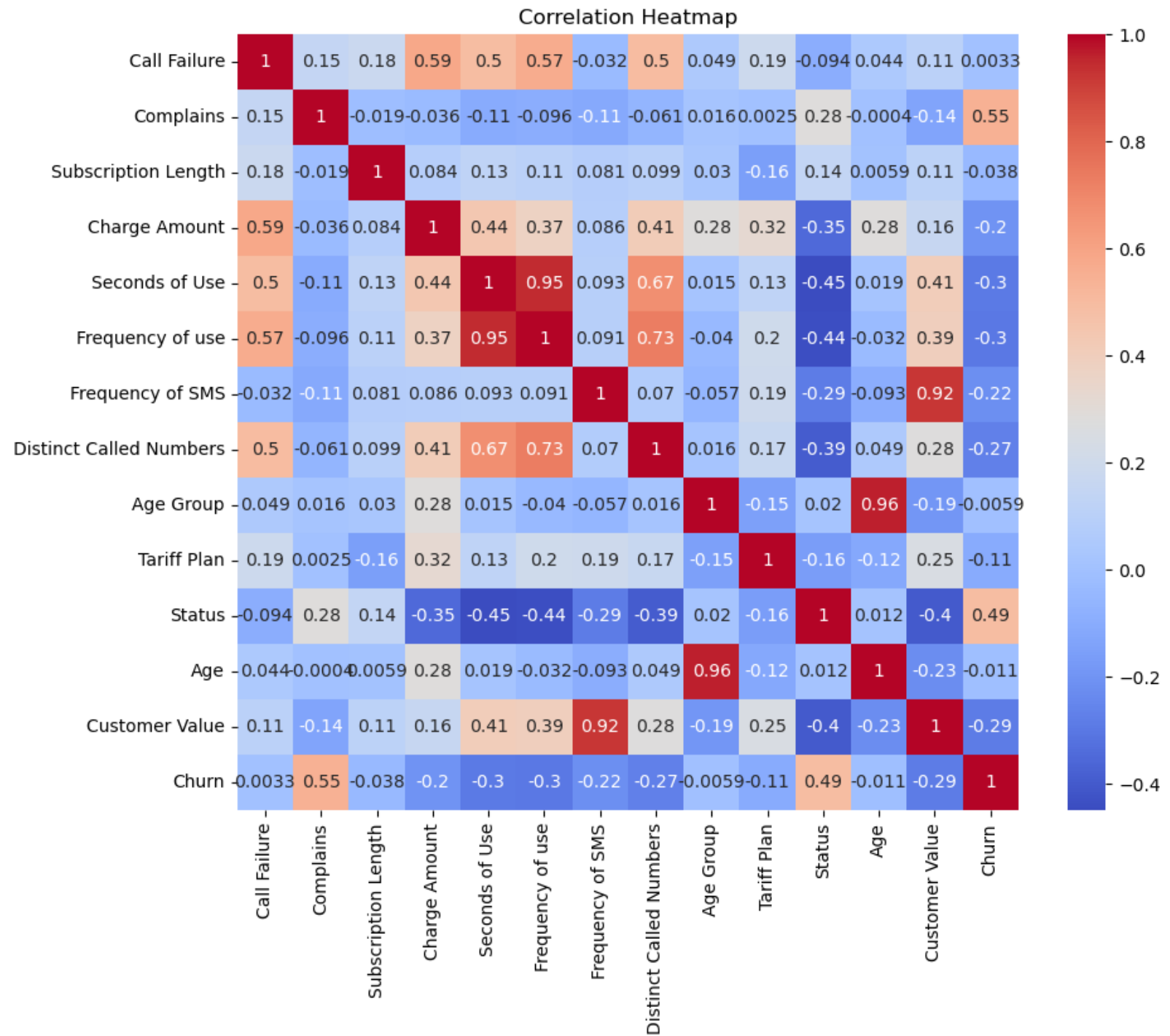
Distinct Called Numbers Distribution by Churn Status



➤ Customers who engage with a broader range of contacts are less likely to churn.



This heatmap visualizes the correlation between various features in the dataset and can guide feature selection for predictive models by highlighting the most relevant features.



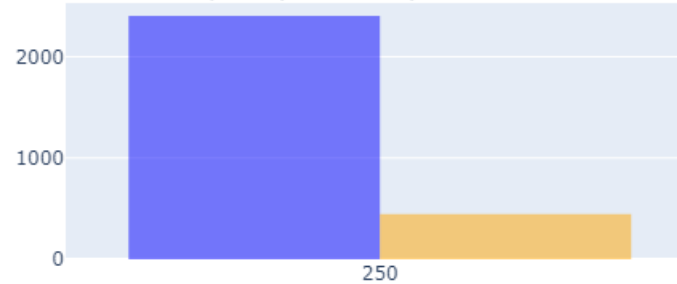
# Interactive Dashboard Using Plotly

This Dashboard visually highlights key metrics differentiating churned (in yellow) and non-churned (in blue) customers.

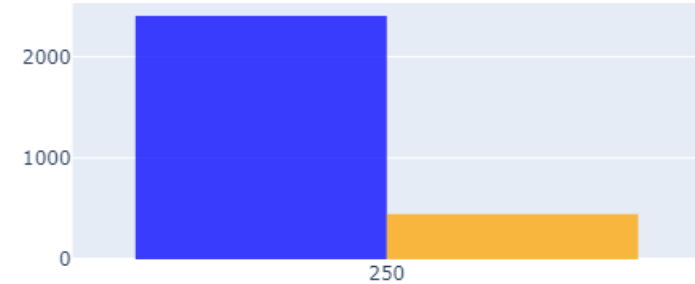
Customer Churn Dashboard

All Customers ▼

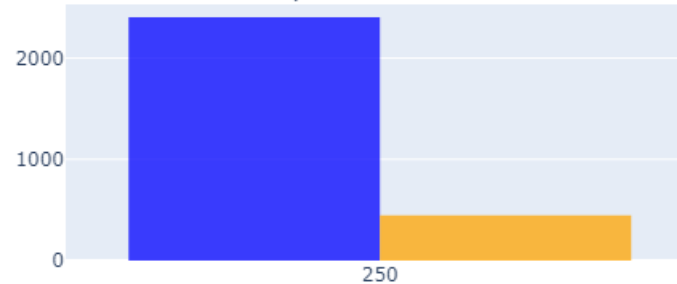
Frequency of Use by Churn Status



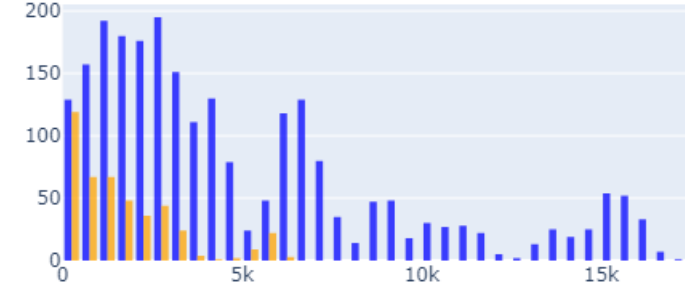
Call Failure Distribution



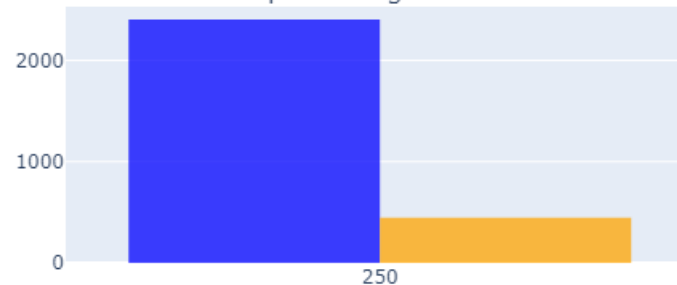
Complains Distribution



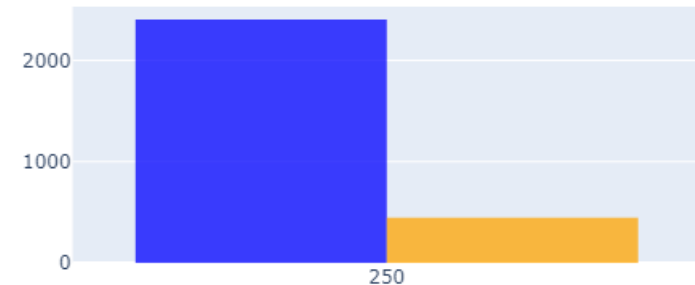
Seconds of Use



Subscription Length Distribution



Distinct Called Numbers Distribution



## Summary of Findings

- Strong correlation between **usage frequency** and **churn** likelihood. *Low engagement* (frequency/seconds of use) *strongly correlates* with **churn**.
- EDA provided **actionable insights** for the predictive model.
- Insights were used to guide **key recommendations**.