Mount Drive & Import Libraries

```
from google.colab import drive
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Mount Google Drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Double-click (or enter) to edit

```
# Load dataset (update path if needed)
file_path = "/content/drive/MyDrive/datasetnws/Telco-Customer-Churn/train.csv"
df = pd.read_csv(file_path)

# Check first rows
df.head()
```

	Age	Avg Monthly GB Download	Avg Monthly Long Distance Charges	Churn Category	Churn Reason	Churn Score	City	CLTV	Contract	Country	...	Tenure in Months	Total Charges	Total Extra Data Charges	Total Long Distance Charges	Total Refunds
0	72	4	19.44	NaN	NaN	51	San Mateo	4849	Two Year	United States	...	25	2191.15	0	486.00	0.0
1	27	59	45.62	NaN	NaN	27	Sutter Creek	3715	Month-to-Month	United States	...	35	3418.20	0	1596.70	0.0
2	59	0	16.07	NaN	NaN	59	Santa Cruz	5092	Month-to-Month	United States	...	46	851.20	0	739.22	0.0
3	25	27	0.00	NaN	NaN	49	Brea	2068	One Year	United States	...	27	1246.40	30	0.00	0.0
4	31	21	17.22	Dissatisfaction	Network reliability	88	San Jose	4026	One Year	United States	...	58	3563.80	0	998.76	0.0

5 rows × 52 columns

Data Overview

```
# Shape of dataset
print("Rows:", df.shape[0], " | Columns:", df.shape[1])

# Info & missing values
df.info()
df.isnull().sum()

# Churn rate
df['Churn'].value_counts(normalize=True) * 100
```

```
Rows: 4225  | Columns: 52
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4225 entries, 0 to 4224
Data columns (total 52 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Age                                4225 non-null   int64  
 1   Avg Monthly GB Download            4225 non-null   int64  
 2   Avg Monthly Long Distance Charges  4225 non-null   float64
 3   Churn Category                     1121 non-null   object 
 4   Churn Reason                       1121 non-null   object 
 5   Churn Score                        4225 non-null   int64  
 6   City                               4225 non-null   object 
 7   CLTV                               4225 non-null   int64  
 8   Contract                           4225 non-null   object 
 9   Country                            4225 non-null   object 
 10  Customer ID                        4225 non-null   object 
 11  Customer Status                    4225 non-null   object 
 12  Dependents                         4225 non-null   int64  
 13  Device Protection Plan             4225 non-null   int64  
 14  Gender                             4225 non-null   object 
 15  Internet Service                   4225 non-null   int64  
 16  Internet Type                      3339 non-null   object 
 17  Lat Long                           4225 non-null   object 
 18  Latitude                           4225 non-null   float64
 19  Longitude                          4225 non-null   float64
 20  Married                            4225 non-null   int64  
 21  Monthly Charge                     4225 non-null   float64
 22  Multiple Lines                     4225 non-null   int64  
 23  Number of Dependents               4225 non-null   int64  
 24  Number of Referrals                4225 non-null   int64  
 25  Offer                              1901 non-null   object 
 26  Online Backup                      4225 non-null   int64  
 27  Online Security                    4225 non-null   int64  
 28  Paperless Billing                  4225 non-null   int64  
 29  Partner                            4225 non-null   int64  
 30  Payment Method                     4225 non-null   object 
 31  Phone Service                      4225 non-null   int64  
 32  Population                         4225 non-null   int64  
 33  Premium Tech Support               4225 non-null   int64  
 34  Quarter                            4225 non-null   object 
 35  Referred a Friend                  4225 non-null   int64  
 36  Satisfaction Score                 4225 non-null   int64  
 37  Senior Citizen                     4225 non-null   int64  
 38  State                              4225 non-null   object 
 39  Streaming Movies                   4225 non-null   int64  
 40  Streaming Music                    4225 non-null   int64  
 41  Streaming TV                       4225 non-null   int64  
 42  Tenure in Months                   4225 non-null   int64  
 43  Total Charges                      4225 non-null   float64
 44  Total Extra Data Charges           4225 non-null   int64  
 45  Total Long Distance Charges        4225 non-null   float64
 46  Total Refunds                      4225 non-null   float64
 47  Total Revenue                      4225 non-null   float64
 48  Under 30                           4225 non-null   int64  
 49  Unlimited Data                     4225 non-null   int64  
 50  Zip Code                           4225 non-null   int64  
 51  Churn                              4225 non-null   int64  
dtypes: float64(8), int64(30), object(14)
memory usage: 1.7+ MB
```

	proportion
Churn	
0	73.467456
1	26.532544

dtype: float64

```
# ----- DATA CLEANING -----

# 1. Check missing values
print("Missing values per column:\n", df.isnull().sum())

# 2. Handle missing values in 'Churn Category', 'Churn Reason', 'Offer', 'Internet Type'
#    -> Fill NaN with "Unknown"
for col in ['Churn Category','Churn Reason','Offer','Internet Type']:
    df[col] = df[col].fillna("Unknown")

# 3. Convert categorical columns with Yes/No to integers (if needed)
# Example: Paperless Billing (Yes=1, No=0)
```

```
df['Paperless Billing'] = df['Paperless Billing'].map({'Yes':1,'No':0})

# 4. Ensure numeric columns are correct
numeric_cols = ['Total Charges','Monthly Charge','Total Revenue']
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')

# Fill any numeric NaN with median
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].median())

# 5. Drop duplicates (if any)
df = df.drop_duplicates()

# Final check
print("Dataset shape after cleaning:", df.shape)
df.head()
```

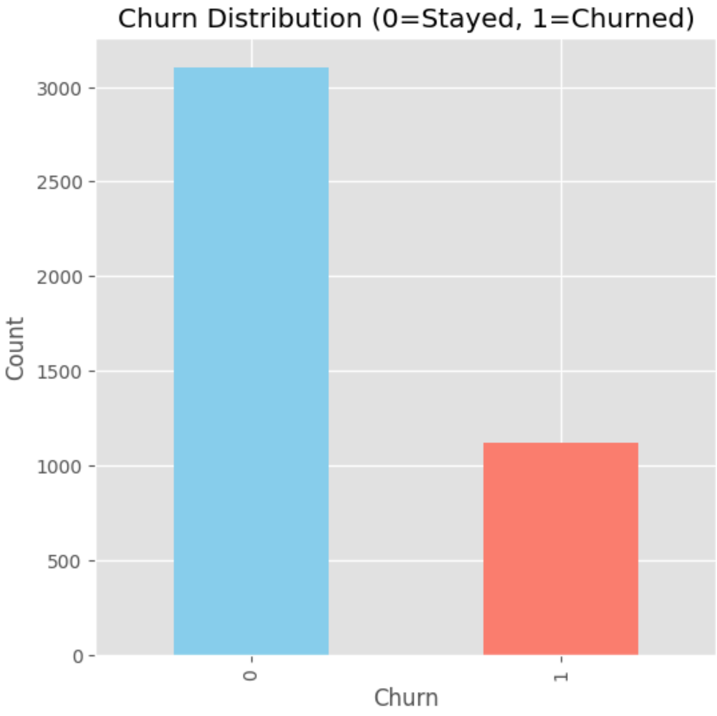
```
Missing values per column:
 Age                                     0
Avg Monthly GB Download                 0
Avg Monthly Long Distance Charges       0
Churn Category                       3104
Churn Reason                         3104
Churn Score                             0
City                                    0
CLTV                                    0
Contract                                0
Country                                 0
Customer ID                             0
Customer Status                         0
Dependents                              0
Device Protection Plan                  0
Gender                                  0
Internet Service                        0
Internet Type                         886
Lat Long                                0
Latitude                                0
Longitude                               0
Married                                 0
Monthly Charge                          0
Multiple Lines                          0
Number of Dependents                    0
Number of Referrals                     0
Offer                                2324
Online Backup                           0
Online Security                         0
Paperless Billing                       0
Partner                                 0
Payment Method                          0
Phone Service                           0
Population                              0
Premium Tech Support                    0
Quarter                                 0
Referred a Friend                       0
Satisfaction Score                      0
Senior Citizen                          0
State                                   0
Streaming Movies                        0
Streaming Music                         0
Streaming TV                            0
Tenure in Months                        0
Total Charges                           0
Total Extra Data Charges                0
Total Long Distance Charges             0
Total Refunds                           0
Total Revenue                           0
Under 30                                0
Unlimited Data                          0
Zip Code                                0
Churn                                   0
dtype: int64
Dataset shape after cleaning: (4225, 52)
```

	Age	Avg Monthly GB Download	Avg Monthly Long Distance Charges	Churn Category	Churn Reason	Churn Score	City	CLTV	Contract	Country	...	Tenure in Months	Total Charges	Total Extra Data Charges	Total Long Distance Charges	Tota Refund
0	72	4	19.44	Unknown	Unknown	51	San Mateo	4849	Two Year	United States	...	25	2191.15	0	486.00	0.
1	27	59	45.62	Unknown	Unknown	27	Sutter Creek	3715	Month-to-Month	United States	...	35	3418.20	0	1596.70	0.
2	59	0	16.07	Unknown	Unknown	59	Santa Cruz	5092	Month-to-Month	United States	...	46	851.20	0	739.22	0.
3	25	27	0.00	Unknown	Unknown	49	Brea	2068	One Year	United States	...	27	1246.40	30	0.00	0.
4	31	21	17.22	Dissatisfaction	Network reliability	88	San Jose	4026	One Year	United States	...	58	3563.80	0	998.76	0.

5 rows × 52 columns

Basic Churn Distribution

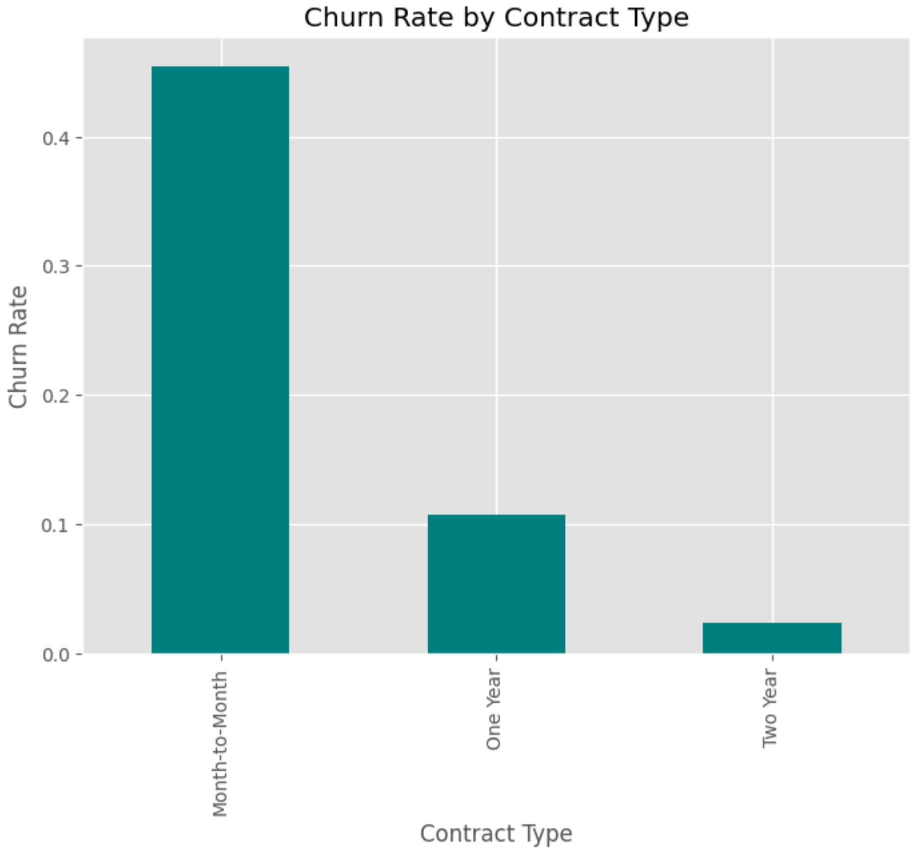
```
plt.figure(figsize=(6,6))
df['Churn'].value_counts().plot(
    kind='bar'
    kind='bar',
    color=['skyblue','salmon']
)
plt.title("Churn Distribution (0=Stayed, 1=Churned)")
plt.xlabel("Churn")
plt.ylabel("Count")
plt.show()
```



Churn by Contract Type

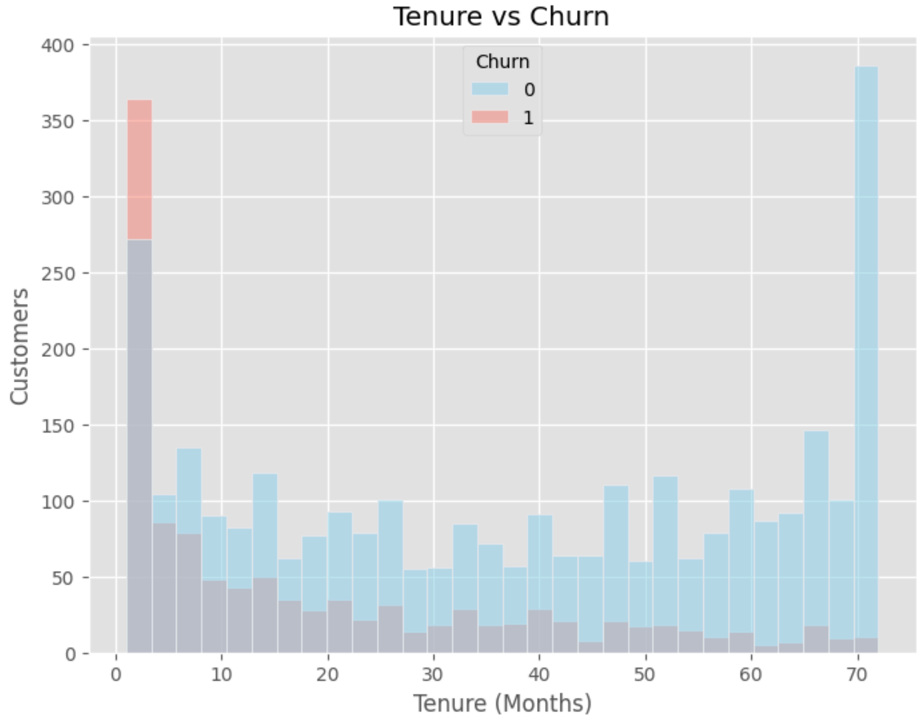
```
contract_churn = df.groupby("Contract")['Churn'].mean()

plt.figure(figsize=(8,6))
contract_churn.plot(kind='bar', color="teal")
plt.title("Churn Rate by Contract Type")
plt.ylabel("Churn Rate")
plt.xlabel("Contract Type")
plt.show()
```



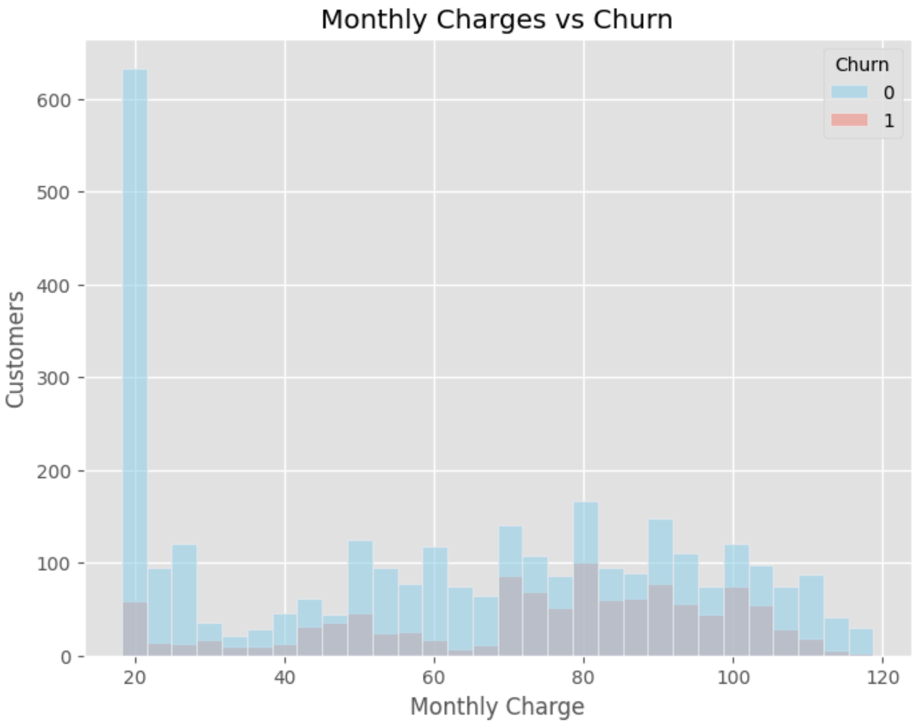
Tenure vs Churn

```
plt.figure(figsize=(8,6))
sns.histplot(data=df, x="Tenure in Months", hue="Churn", bins=30, kde=False, palette=["skyblue","salmon"])
plt.title("Tenure vs Churn")
plt.xlabel("Tenure (Months)")
plt.ylabel("Customers")
plt.show()
```



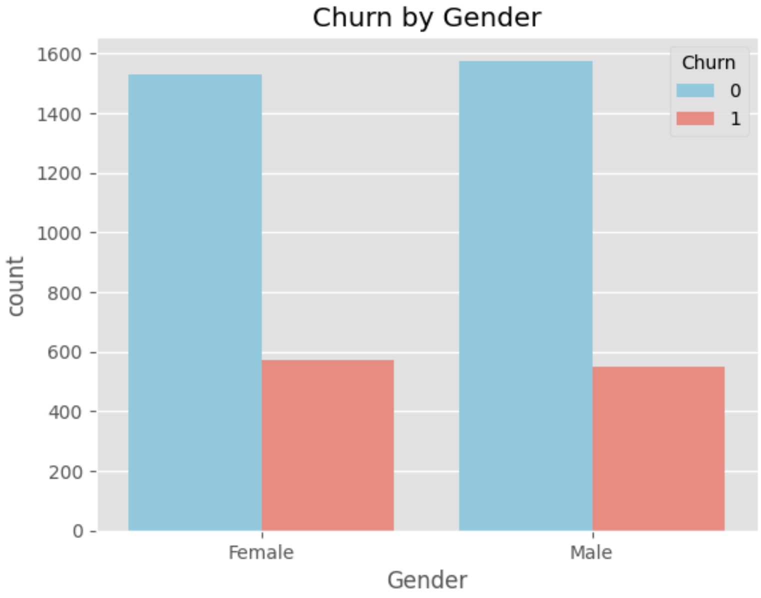
Monthly Charges vs Churn

```
plt.figure(figsize=(8,6))
sns.histplot(data=df, x="Monthly Charge", hue="Churn", bins=30, kde=False, palette=["skyblue","salmon"])
plt.title("Monthly Charges vs Churn")
plt.xlabel("Monthly Charge")
plt.ylabel("Customers")
plt.show()
```



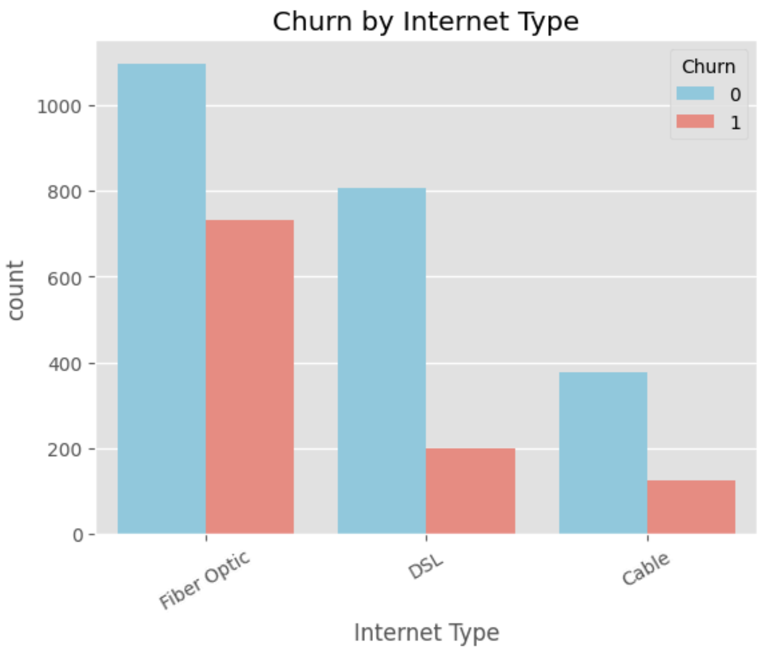
Cross Analysis

```
sns.countplot(data=df, x="Gender", hue="Churn", palette=["skyblue","salmon"])
plt.title("Churn by Gender")
plt.show()
```



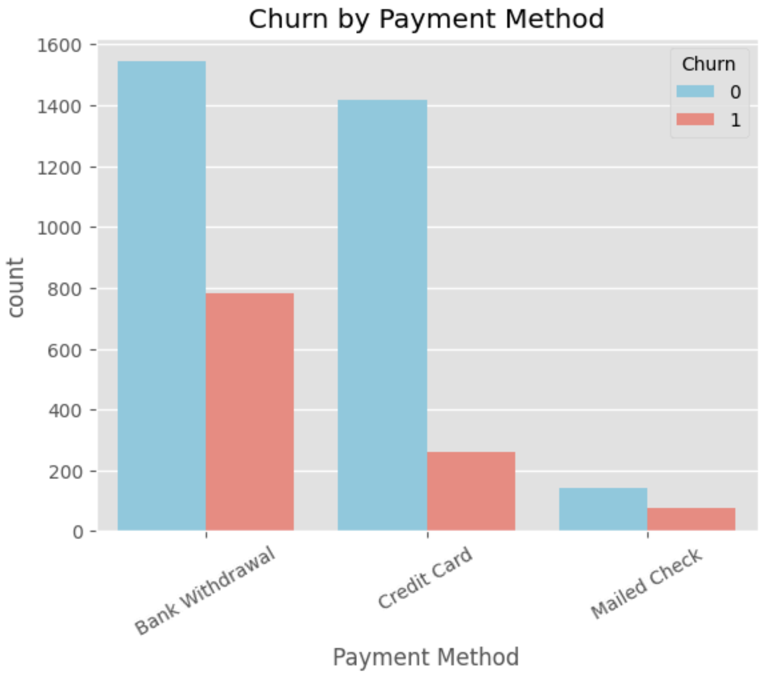
```
#Internet Type
```

```
sns.countplot(data=df, x="Internet Type", hue="Churn", palette=["skyblue","salmon"])
plt.title("Churn by Internet Type")
plt.xticks(rotation=30)
plt.show()
```



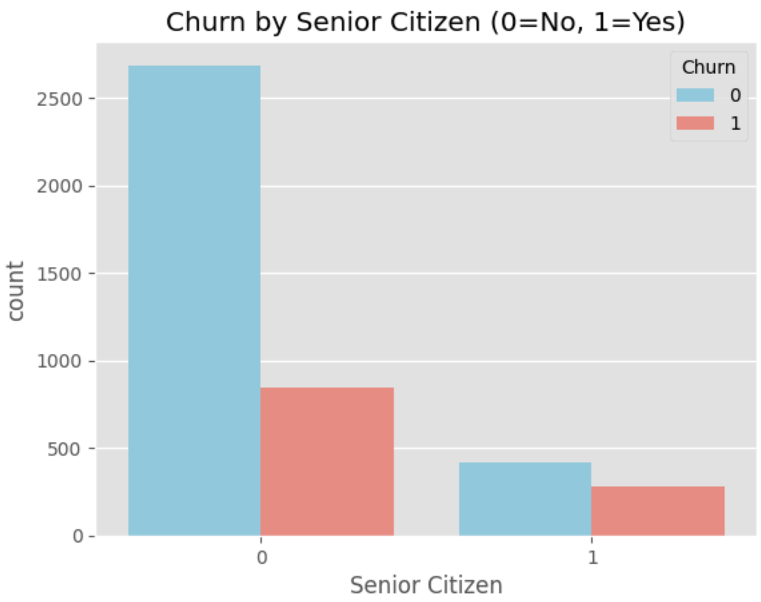
Payment Method

```
sns.countplot(data=df, x="Payment Method", hue="Churn", palette=["skyblue","salmon"])
plt.title("Churn by Payment Method")
plt.xticks(rotation=30)
plt.show()
```



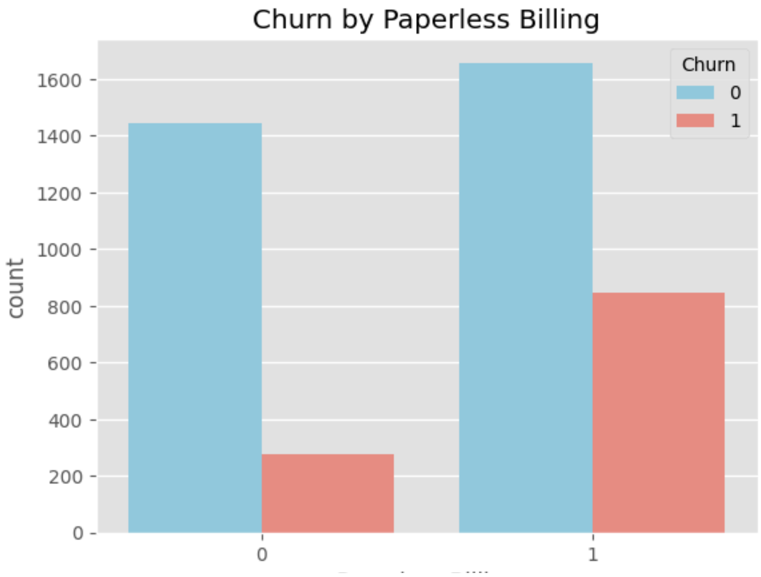
Senior Citizens

```
sns.countplot(data=df, x="Senior Citizen", hue="Churn", palette=["skyblue","salmon"])
plt.title("Churn by Senior Citizen (0=No, 1=Yes)")
plt.show()
```



Paperless Billing

```
sns.countplot(data=df, x="Paperless Billing", hue="Churn", palette=["skyblue","salmon"])
plt.title("Churn by Paperless Billing")
plt.show()
```


Churn by Paperless Billing
count
Churn