```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files

from google.colab import drive
drive.mount('/content/drive')

# Read the CSV file
df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/Customer
Churn.csv') # Adjust path if necessary
df.head()
```

```
Mounted at /content/drive
```

{"type":"dataframe","variable_name":"df"}

New Section

```
# Convert binary categorical columns
df['gender'] = df['gender'].map({'Male': 0, 'Female': 1})
df['Partner'] = df['Partner'].map({'Yes': 1, 'No': 0})
df['Dependents'] = df['Dependents'].map({'Yes': 1, 'No': 0})
df['PhoneService'] = df['PhoneService'].map({'Yes': 1, 'No': 0})
df['MultipleLines'] = df['MultipleLines'].map({'Yes': 1, 'No': 0, 'No
phone service': 0})
df['OnlineSecurity'] = df['OnlineSecurity'].map({'Yes': 1, 'No': 0,
'No internet service': 0})
df['OnlineBackup'] = df['OnlineBackup'].map({'Yes': 1, 'No': 0, 'No
internet service': 0})
df['DeviceProtection'] = df['DeviceProtection'].map({'Yes': 1, 'No':
0, 'No internet service': 0})
df['TechSupport'] = df['TechSupport'].map({'Yes': 1, 'No': 0, 'No
internet service': 0})
df['StreamingTV'] = df['StreamingTV'].map({'Yes': 1, 'No': 0, 'No
internet service': 0})
df['StreamingMovies'] = df['StreamingMovies'].map({'Yes': 1, 'No': 0,
'No internet service': 0})
df['PaperlessBilling'] = df['PaperlessBilling'].map({'Yes': 1, 'No':
0})
df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})

# Handle multi-category columns using one-hot encoding
df = pd.get_dummies(df, columns=['InternetService', 'Contract',
'PaymentMethod'], drop_first=True)

# Convert TotalCharges to numeric and handle missing values
```

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'],
errors='coerce')
df['TotalCharges'] = df['TotalCharges'].fillna(0)

# Drop the 'customerID' column if not needed
df = df.drop(columns=['customerID'])

# Check the updated DataFrame
print(df.info())
```

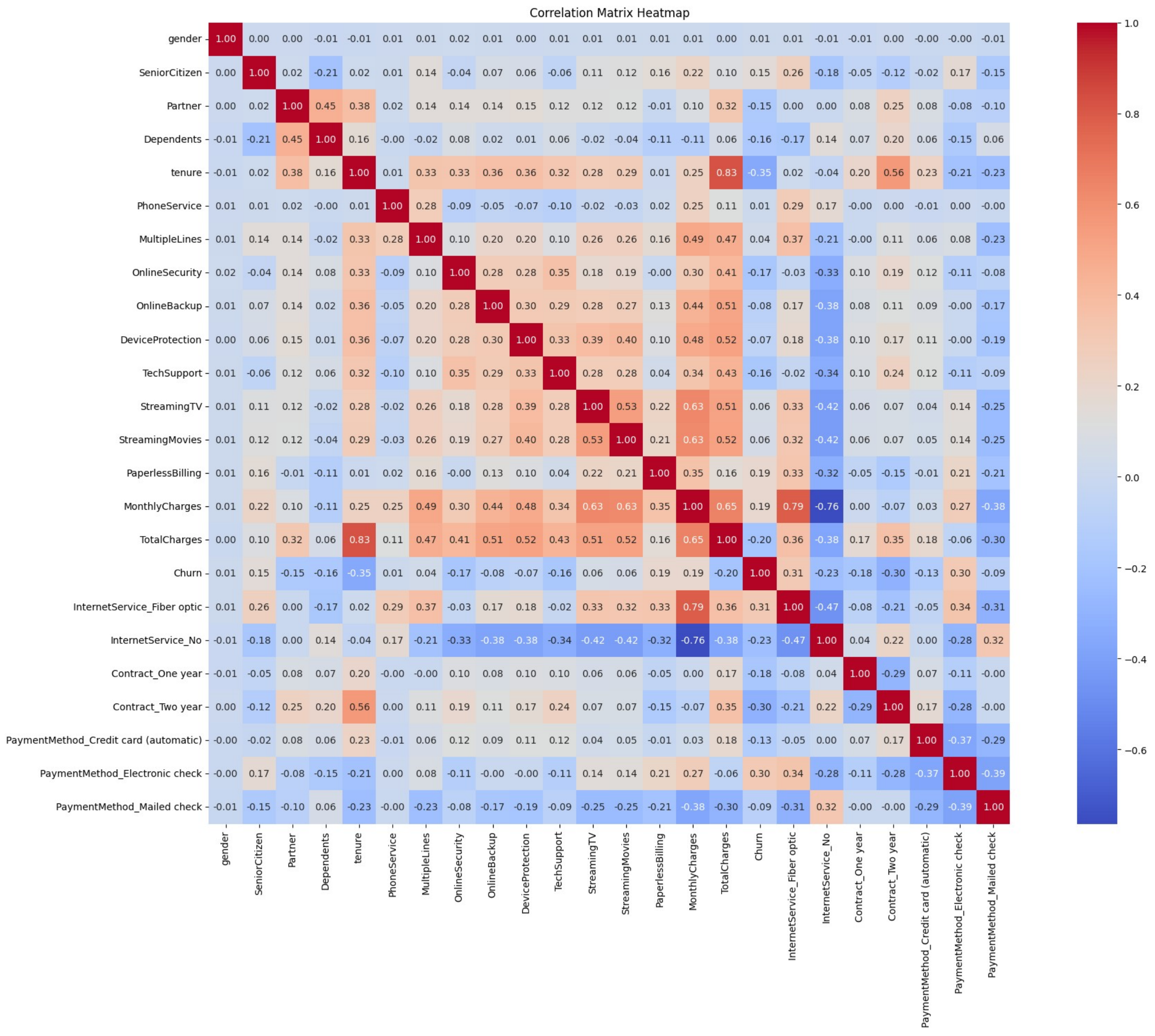
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 24 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   gender                                 7043 non-null   int64  
 1   SeniorCitizen                          7043 non-null   int64  
 2   Partner                                7043 non-null   int64  
 3   Dependents                             7043 non-null   int64  
 4   tenure                                 7043 non-null   int64  
 5   PhoneService                           7043 non-null   int64  
 6   MultipleLines                          7043 non-null   int64  
 7   OnlineSecurity                         7043 non-null   int64  
 8   OnlineBackup                           7043 non-null   int64  
 9   DeviceProtection                       7043 non-null   int64  
 10  TechSupport                            7043 non-null   int64  
 11  StreamingTV                            7043 non-null   int64  
 12  StreamingMovies                        7043 non-null   int64  
 13  PaperlessBilling                       7043 non-null   int64  
 14  MonthlyCharges                         7043 non-null   float64
 15  TotalCharges                           7043 non-null   float64
 16  Churn                                  7043 non-null   int64  
 17  InternetService_Fiber optic            7043 non-null   bool   
 18  InternetService_No                     7043 non-null   bool   
 19  Contract_One year                      7043 non-null   bool   
 20  Contract_Two year                      7043 non-null   bool   
 21  PaymentMethod_Credit card (automatic)  7043 non-null   bool   
 22  PaymentMethod_Electronic check         7043 non-null   bool   
 23  PaymentMethod_Mailed check             7043 non-null   bool   
dtypes: bool(7), float64(2), int64(15)
memory usage: 983.7 KB
None
```

```
df.head()
```

{"type":"dataframe","variable_name":"df"}

```
# Assume your dataset is in a pandas DataFrame called df
corr = df.corr()  # Calculate correlation matrix
```

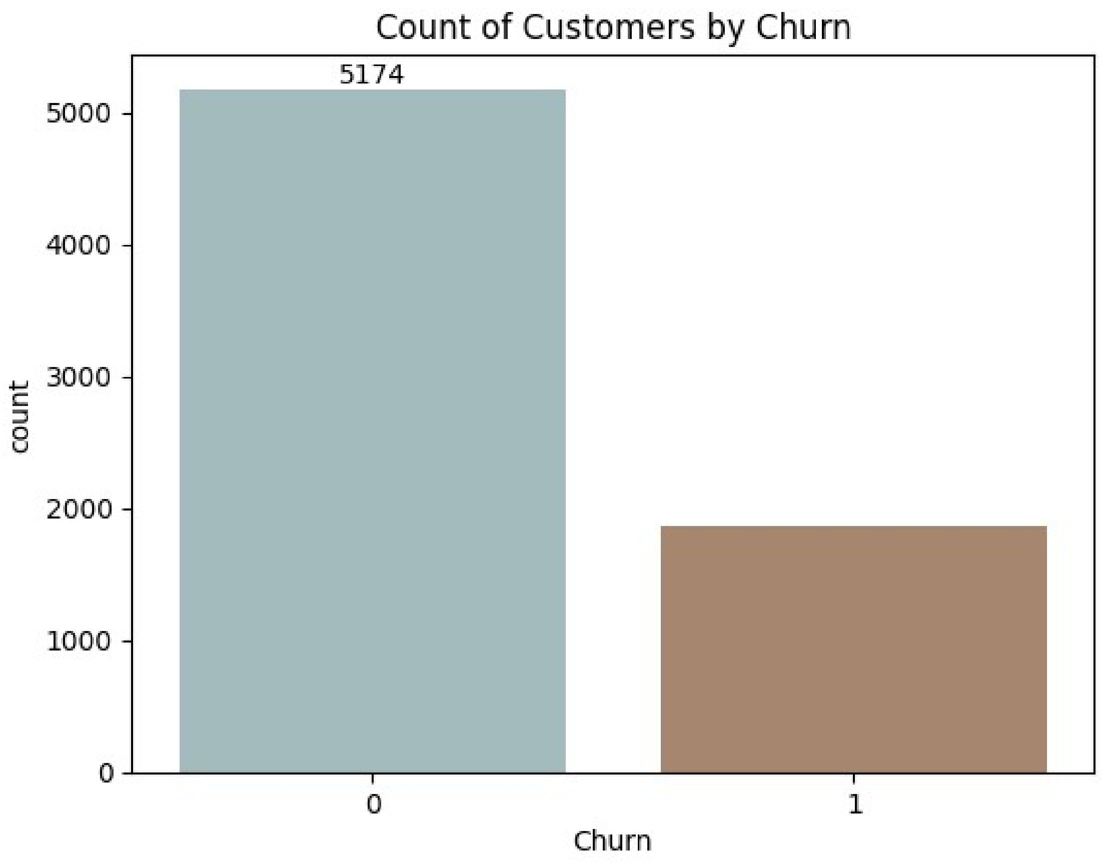
```python
plt.figure(figsize=(25, 15))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', square=True)
plt.title('Correlation Matrix Heatmap')
# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```



```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>
```

```python
# Set custom colors: for example, blue for '0', red for '1'
custom_colors = ['#9fbfc0', '#b08466'] # You can use named colors or
hex codes

# Plot with custom colors
ax = sns.countplot(x='Churn', data=df, palette=custom_colors)

# Add count labels on top of the bars
ax.bar_label(ax.containers[0])

# Add title
plt.title("Count of Customers by Churn")

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```

```
/tmp/ipython-input-5-1480726103.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  ax = sns.countplot(x='Churn', data=df, palette=custom_colors)
```

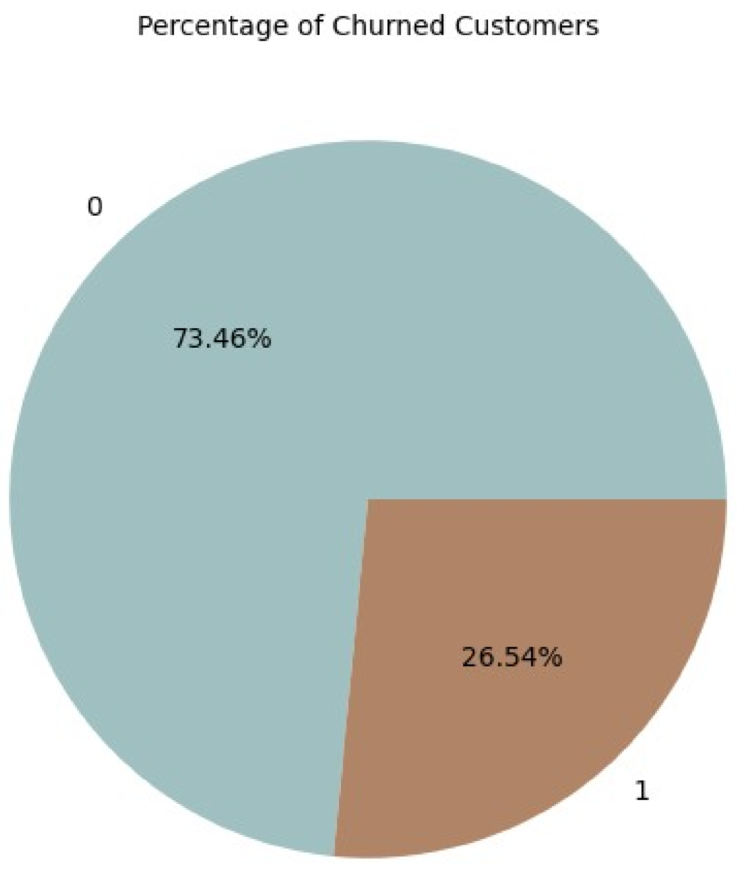


```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

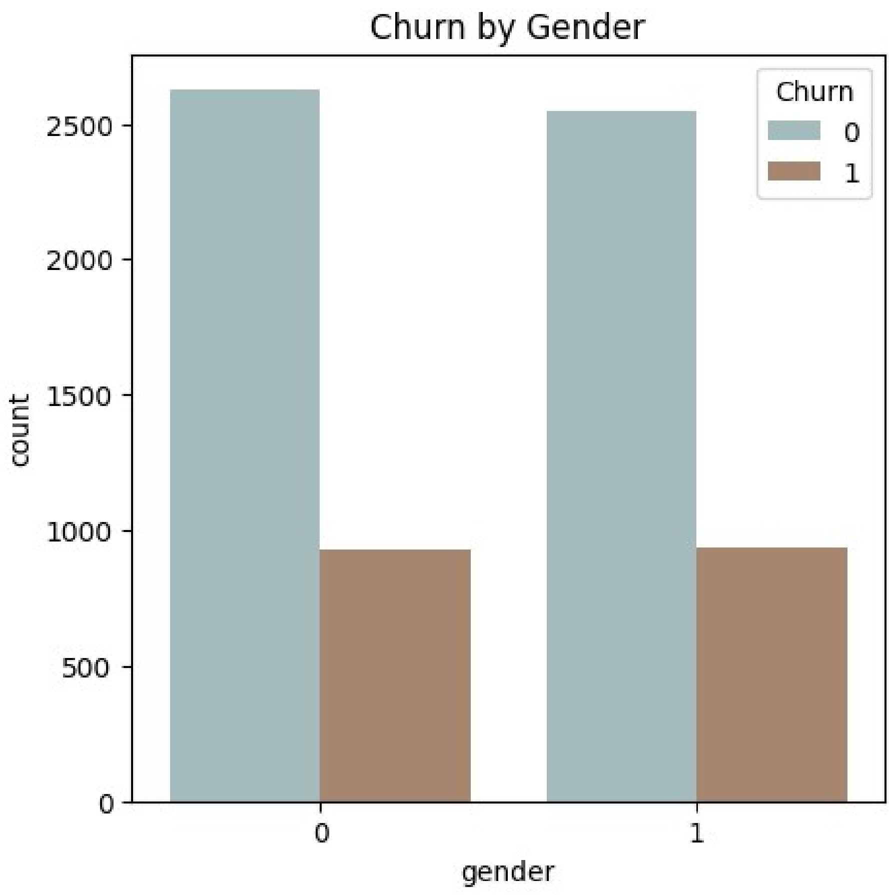
```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 6))

# Group and aggregate
gb = df.groupby('Churn').agg({'Churn': 'count'})

# Define custom colors
colors = ['#9fbfc0', '#b08466']  # Green for 0, Orange for 1

# Plot pie chart
plt.pie(gb['Churn'], labels=gb.index, autopct="%1.2f%%",
colors=colors)

# Add title
plt.title("Percentage of Churned Customers", fontsize=10)

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')
```

```
# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```

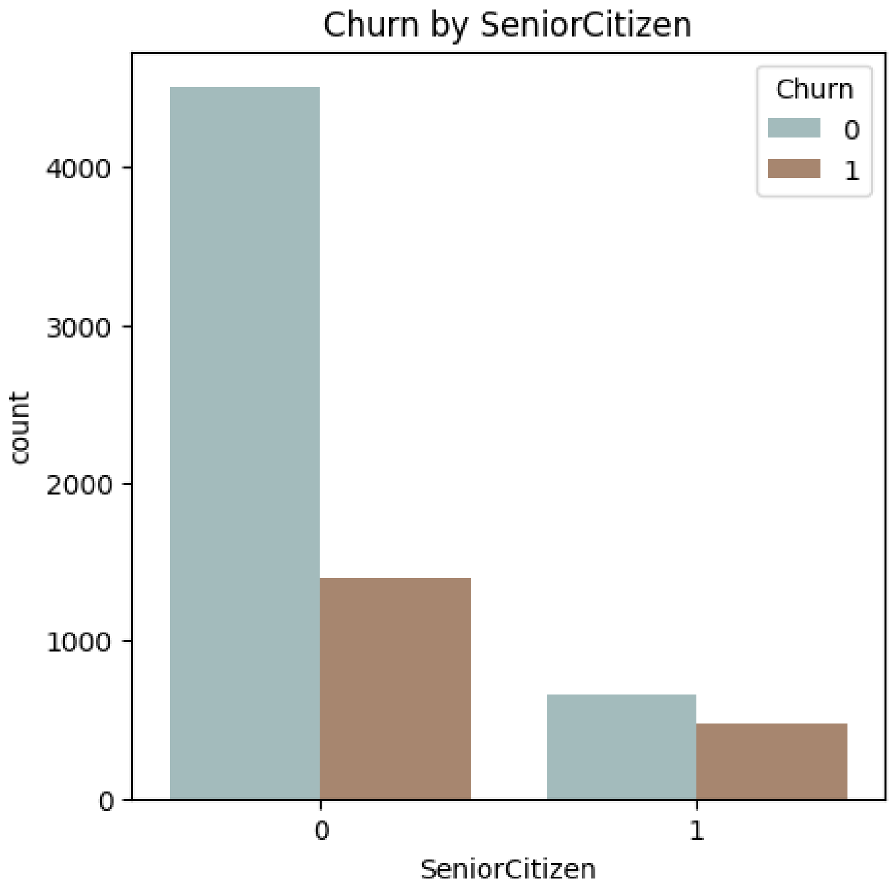


```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

plt.figure(figsize=(5, 5))

# Set custom colors for hue='Churn'
custom_colors = ['#9fbfc0', '#b08466']  # Green for 0, Orange for 1

# Plot
sns.countplot(x="gender", data=df, hue='Churn', palette=custom_colors)

plt.title("Churn by Gender")
```

```
# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```

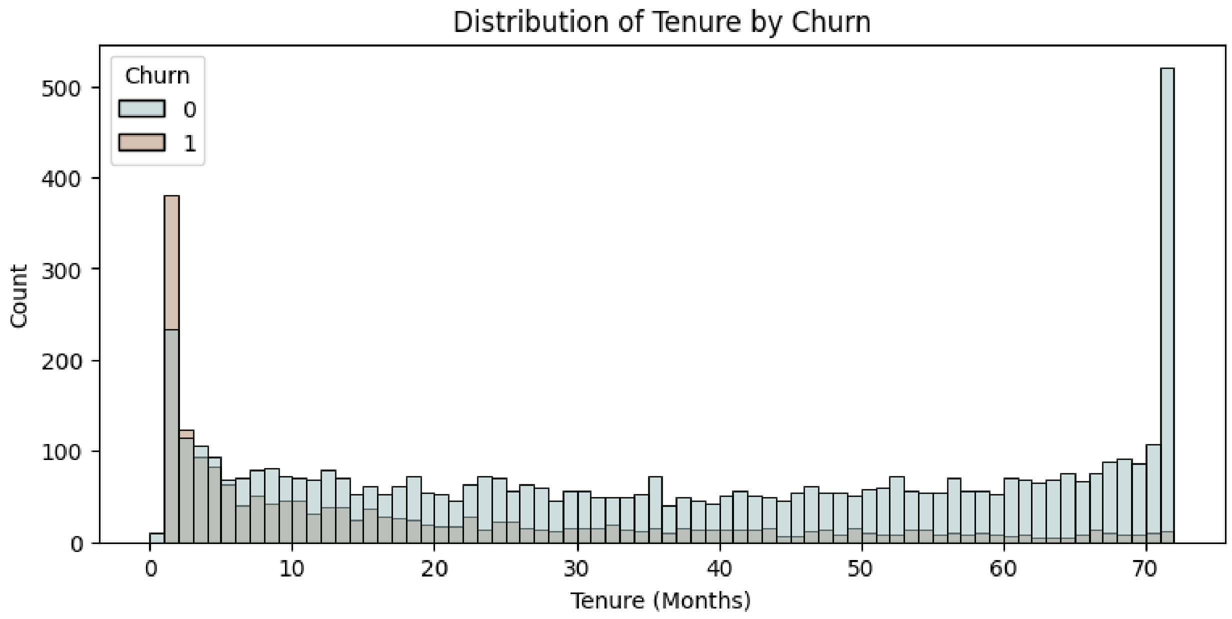


```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

plt.figure(figsize=(5, 5))

# Define custom colors
custom_colors = ['#9fbfc0', '#b08466']  # Green for 0 (Not Churned),
Orange for 1 (Churned)

# Plot with custom colors
sns.countplot(x="SeniorCitizen", data=df, hue='Churn',
palette=custom_colors)

plt.title("Churn by SeniorCitizen")
# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```



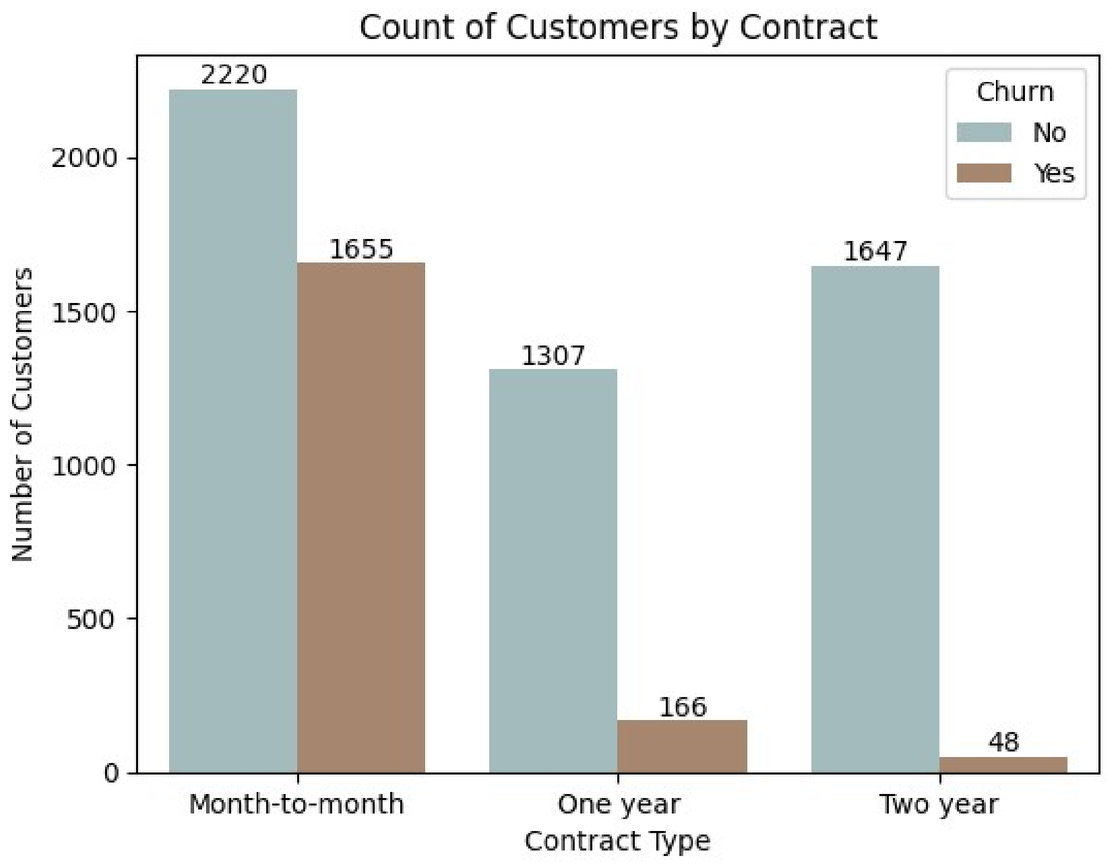
```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

plt.figure(figsize=(9, 4))

# Define custom colors for Churn levels
custom_colors = ['#9fbfc0', '#b08466']   # Green for 0, Orange for 1
```

```
# Plot histogram with custom hue colors
sns.histplot(x="tenure", data=df, bins=72, hue="Churn",
palette=custom_colors)

plt.title("Distribution of Tenure by Churn")
plt.xlabel("Tenure (Months)")
plt.ylabel("Count")

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```



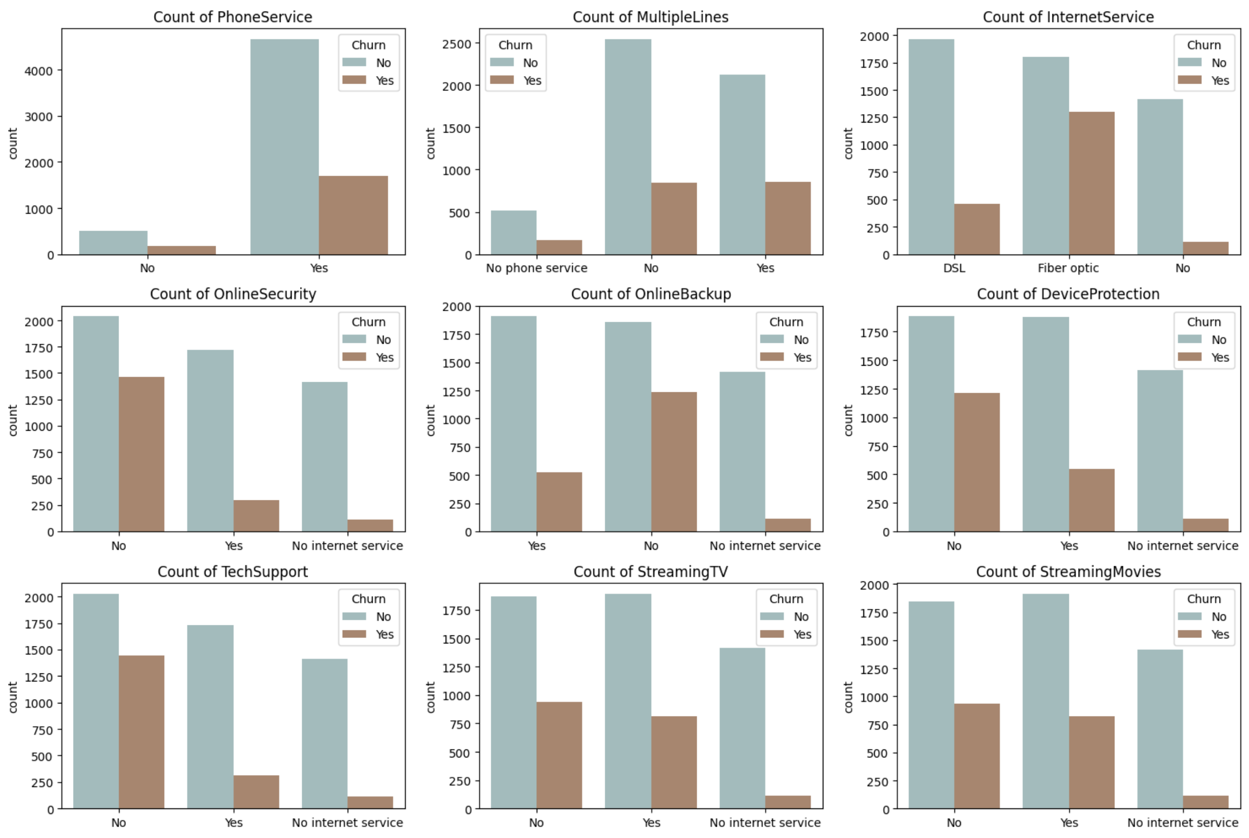
```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

drive.mount('/content/drive')

# Read the CSV file
df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/Customer
Churn.csv')  # Adjust path if necessary

# Define custom colors for Churn
custom_colors = ['#9fbfc0', '#b08466']  # Green for 0 (No Churn),
Orange for 1 (Churn)

# Plot with custom colors
ax = sns.countplot(x='Contract', data=df, hue="Churn",
palette=custom_colors)

# Add value labels on bars
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])

plt.title("Count of Customers by Contract")
plt.xlabel("Contract Type")
plt.ylabel("Number of Customers")

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```

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

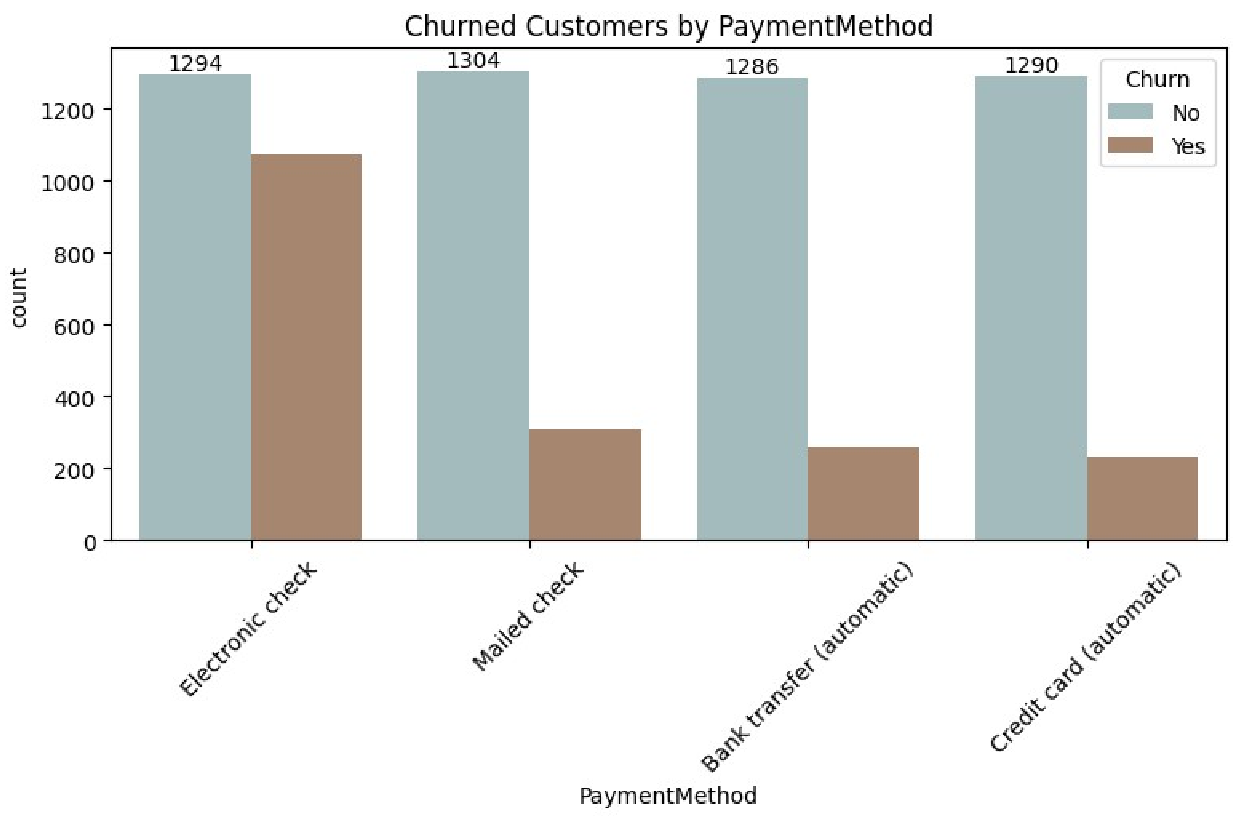


```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

df = df.drop(columns=['customerID'])
df.columns.values

array(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)


# Assuming you have a DataFrame 'df' with the columns
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV', 'StreamingMovies']

# Number of rows and columns for the subplots
n_rows = 3
n_cols = 3
```

```python
# Define custom colors for Churn
custom_colors = ['#9fbfc0', '#b08466']  # Green for 0, Orange for 1

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))

# Flatten the axes array for easy iteration
axes = axes.flatten()

# Loop through each column and create a count plot
for i, col in enumerate(columns):
    sns.countplot(data=df, x=col, ax=axes[i], hue=df["Churn"],
palette=custom_colors)
    axes[i].set_title(f'Count of {col}')
    axes[i].set_xlabel('')  # Optional: to remove x-labels for a
cleaner look

# Adjust layout
plt.tight_layout()

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```



```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>

plt.figure(figsize=(9,4))
ax = sns.countplot(x='PaymentMethod', data=df, hue='Churn',
palette=['#9fbfc0', '#b08466'])
ax.bar_label(ax.containers[0])
plt.title("Churned Customers by PaymentMethod")
plt.xticks(rotation=45)

# Save plot to file
plt.savefig("performance_plot_1.png", dpi=300, bbox_inches='tight')

# Display plot
plt.show()


# Download the saved file
files.download("performance_plot_1.png")
```



Churned Customers by PaymentMethod
1294
1304
1286
1290
Churn
No
Yes
count
0
200
400
600
800
1000
1200
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod

```
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>
```