

```
In [155]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [156]: sns.set(style = "whitegrid")
```

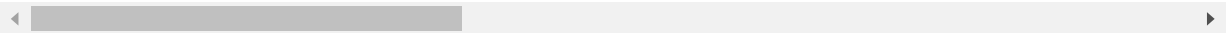
```
In [157]: data =pd.read_csv(f"Customer Churn.csv")
```

```
In [158]: data.head()
```

Out[158]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Inter
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	
4	9237-HQITU	Female	0	No	No	2	Yes	No	

5 rows × 21 columns

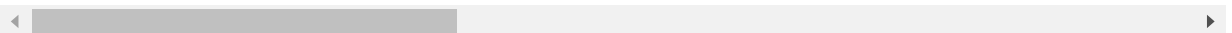


```
In [159]: data.tail()
```

Out[159]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Ir
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	

5 rows × 21 columns



In [160]: data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                7043 non-null   object
2   SeniorCitizen         7043 non-null   int64
3   Partner               7043 non-null   object
4   Dependents            7043 non-null   object
5   tenure               7043 non-null   int64
6   PhoneService          7043 non-null   object
7   MultipleLines         7043 non-null   object
8   InternetService       7043 non-null   object
9   OnlineSecurity        7043 non-null   object
10  OnlineBackup          7043 non-null   object
11  DeviceProtection      7043 non-null   object
12  TechSupport           7043 non-null   object
13  StreamingTV           7043 non-null   object
14  StreamingMovies       7043 non-null   object
15  Contract              7043 non-null   object
16  PaperlessBilling      7043 non-null   object
17  PaymentMethod         7043 non-null   object
18  MonthlyCharges        7043 non-null   float64
19  TotalCharges          7043 non-null   object
20  Churn                 7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

In [161]: data.isnull().sum()

```
Out[161]: customerID            0
gender                0
SeniorCitizen         0
Partner               0
Dependents            0
tenure               0
PhoneService          0
MultipleLines         0
InternetService       0
OnlineSecurity        0
OnlineBackup          0
DeviceProtection      0
TechSupport           0
StreamingTV           0
StreamingMovies       0
Contract              0
PaperlessBilling      0
PaymentMethod         0
MonthlyCharges        0
TotalCharges          0
Churn                 0
dtype: int64
```

```
In [162]: #If change any column data type then use this code:
data["TotalCharges"] = data["TotalCharges"].replace(" ", "0")
data["TotalCharges"] = data["TotalCharges"].astype("float")
```

```
In [163]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customerID            7043 non-null   object
1   gender                7043 non-null   object
2   SeniorCitizen         7043 non-null   int64
3   Partner               7043 non-null   object
4   Dependents            7043 non-null   object
5   tenure               7043 non-null   int64
6   PhoneService          7043 non-null   object
7   MultipleLines         7043 non-null   object
8   InternetService       7043 non-null   object
9   OnlineSecurity        7043 non-null   object
10  OnlineBackup          7043 non-null   object
11  DeviceProtection      7043 non-null   object
12  TechSupport           7043 non-null   object
13  StreamingTV           7043 non-null   object
14  StreamingMovies       7043 non-null   object
15  Contract              7043 non-null   object
16  PaperlessBilling      7043 non-null   object
17  PaymentMethod         7043 non-null   object
18  MonthlyCharges        7043 non-null   float64
19  TotalCharges          7043 non-null   float64
20  Churn                 7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

```
In [164]: data.isnull().sum().sum()
```

```
Out[164]: 0
```

In [165]: `data.isnull().sum()`

```
Out[165]: customerID      0
gender          0
SeniorCitizen   0
Partner         0
Dependents      0
tenure          0
PhoneService    0
MultipleLines   0
InternetService 0
OnlineSecurity  0
OnlineBackup    0
DeviceProtection 0
TechSupport     0
StreamingTV     0
StreamingMovies 0
Contract        0
PaperlessBilling 0
PaymentMethod   0
MonthlyCharges  0
TotalCharges    0
Churn           0
dtype: int64
```

In [166]: `data.describe()`

```
Out[166]:
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

In [167]: `data.duplicated().sum()`

Out[167]: 0

In [168]: *# Change the data type of SeniorCitizen column using this code:*

```
def con(value):
    if value == "Yes":
        return "Yes"
    else:
        return "No"

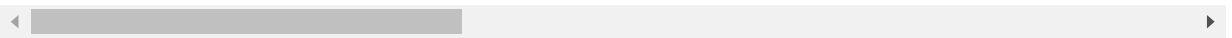
data["SeniorCitizen"] = data["SeniorCitizen"].apply(con)
```

```
In [169]: data.head()
```

```
Out[169]:
```

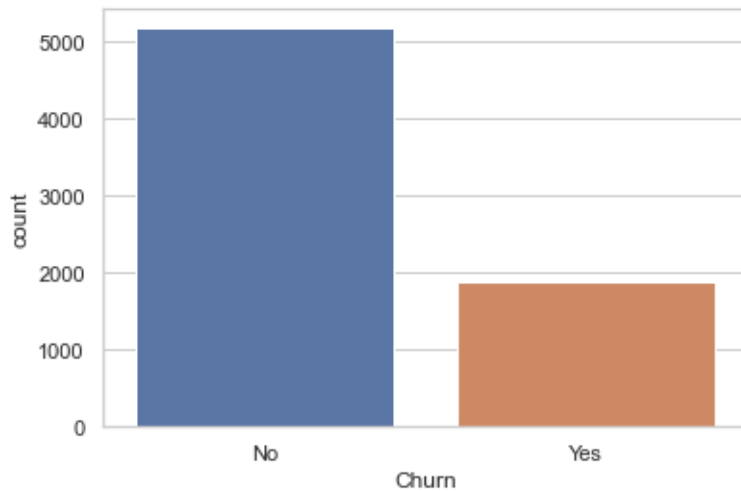
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Inter
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	
1	5575-GNVDE	Male	No	No	No	34	Yes	No	
2	3668-QPYBK	Male	No	No	No	2	Yes	No	
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	
4	9237-HQITU	Female	No	No	No	2	Yes	No	

5 rows × 21 columns

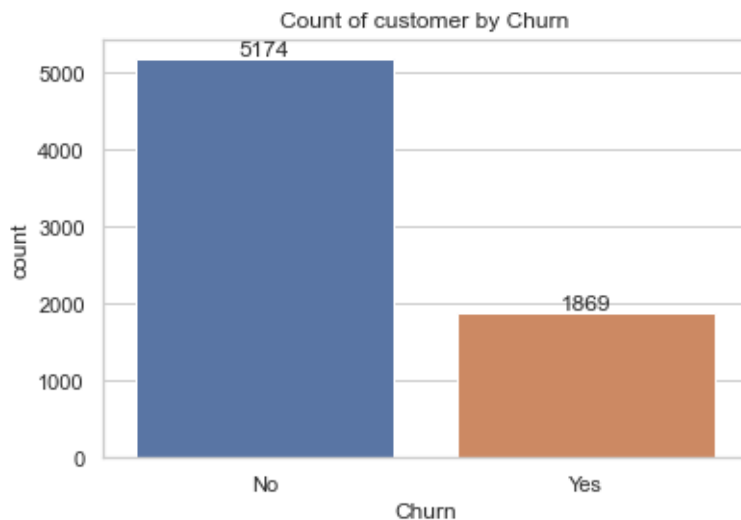


Compare all Cell and visualization using seaborn

```
In [170]: sns.countplot(x=data["Churn"])  
plt.show()
```



```
In [171]: ax = sns.countplot(x=data["Churn"])
ax.bar_label(ax.containers[0])
plt.title("Count of customer by Churn")
plt.show()
```

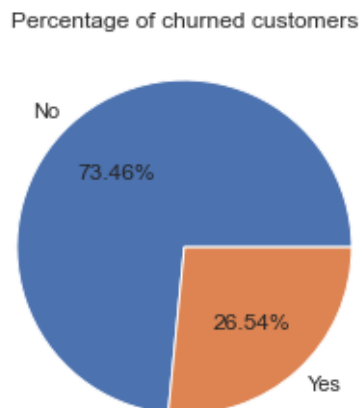


```
In [172]: groupby = data.groupby("Churn").agg({"Churn": "count"})
groupby
```

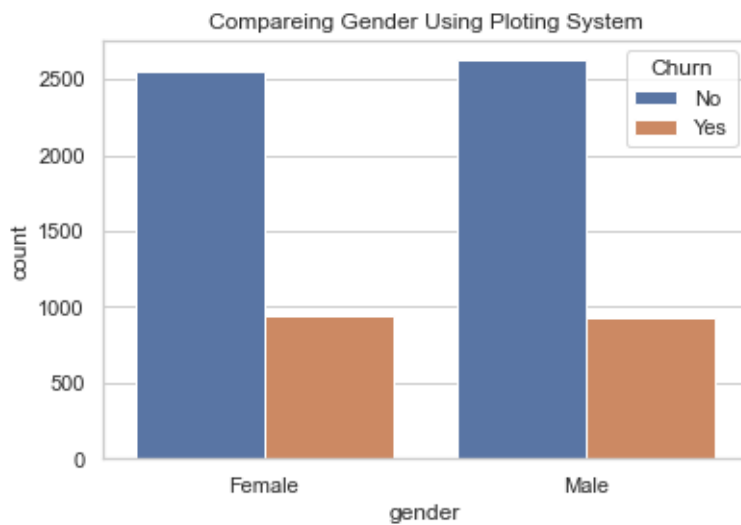
Out[172]:

Churn	
Churn	
No	5174
Yes	1869

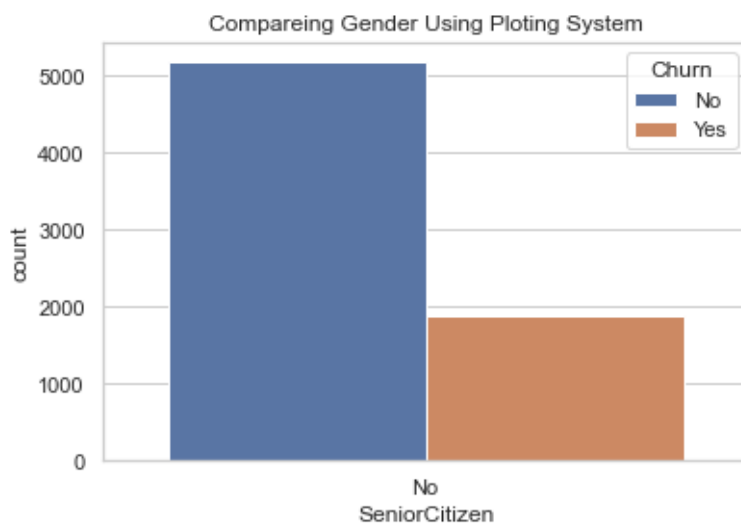
```
In [173]: # At first create a pie chart using this code and using groupby function:
groupby = data.groupby("Churn").agg({"Churn": "count"})
plt.pie (groupby["Churn"], labels = groupby.index , autopct = "%1.2f%%" )
plt.savefig('pie.png')
plt.title("Percentage of churned customers")
plt.show()
```



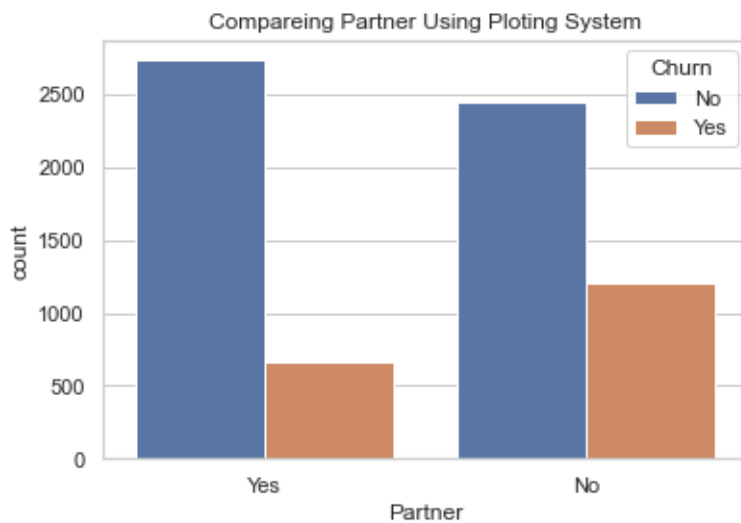
```
In [174]: sns.countplot(x='gender',data=data,hue="Churn")  
plt.title("Compareing Gender Using Ploting System")  
plt.show()
```



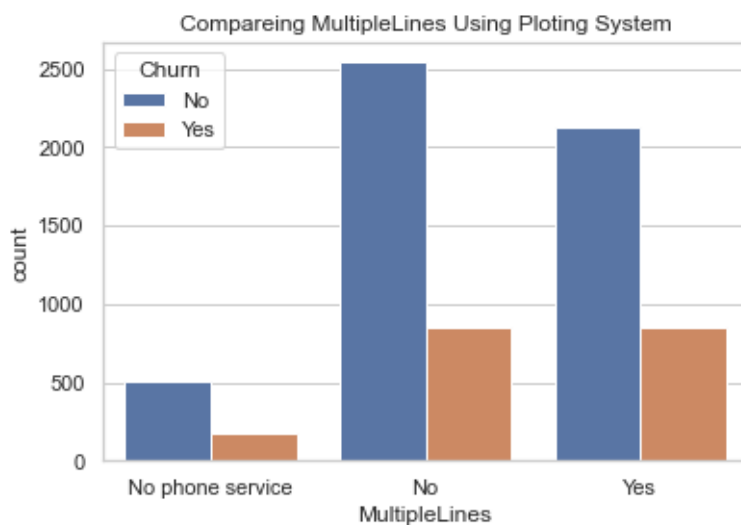
```
In [175]: sns.countplot(x='SeniorCitizen',data=data,hue="Churn")  
plt.title("Compareing Gender Using Ploting System")  
plt.show()
```



```
In [176]: sns.countplot(x = 'Partner', data=data, hue="Churn")
plt.title("Compareing Partner Using Ploting System")
plt.show()
```



```
In [177]: sns.countplot(x = 'MultipleLines', data=data, hue="Churn")
plt.title("Compareing MultipleLines Using Ploting System")
plt.show()
```



```
In [178]: # ax = sns.countplot(x = 'SeniorCitizen', data=data)
# ax.bar_label(ax.containers[0])
# plt.title("Count of Customer by Senior citizen ")
# plt.show()
```



```

In [179]: import pandas as pd
import matplotlib.pyplot as plt

# Sample data creation for illustration
data = pd.DataFrame({
    'SeniorCitizen': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Churn': [0, 1, 0, 1, 1, 0, 1, 0, 1, 0]
})

# Calculate the total counts of Churn for each SeniorCitizen category
total_counts = data.groupby('SeniorCitizen')['Churn'].value_counts(normalize=True)

# Create a stacked bar chart
fig, ax = plt.subplots(figsize=(10, 7)) # Adjust figsize for better visualization

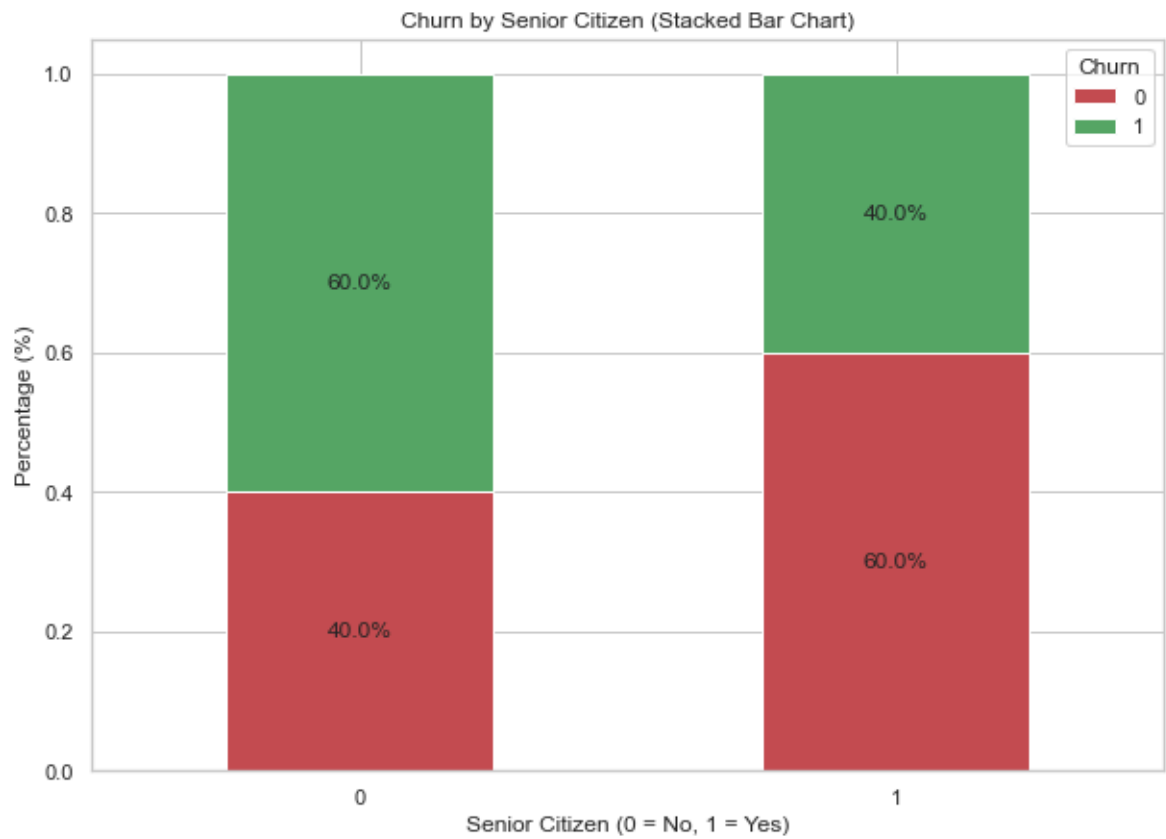
# Plot the bars
total_counts.plot(kind='bar', stacked=True, ax=ax, color=['r', 'g']) # Customize colors

# Add percentage labels on the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy() # Get the position of the rectangle
    ax.text(x + width / 2, y + height / 2, f'{height:.1%}', ha='center', va='center')

# Add titles and labels
plt.title('Churn by Senior Citizen (Stacked Bar Chart)')
plt.xlabel('Senior Citizen (0 = No, 1 = Yes)')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=0)
plt.legend(title='Churn', loc='upper right')

# Show the plot
plt.show()

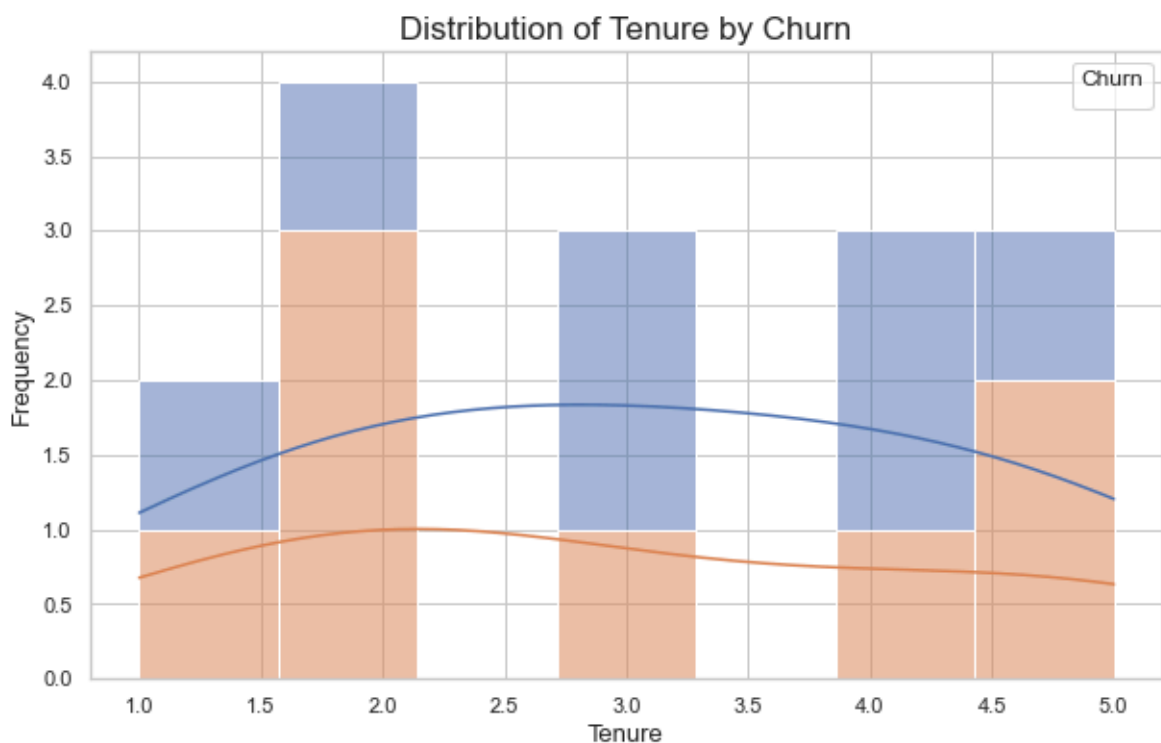
```



```
In [180]: # Sample data creation for illustration
data = pd.DataFrame({
    'tenure': [1, 2, 3, 4, 5, 2, 1, 4, 5, 3, 2, 5, 3, 4, 2],
    'Churn': [0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1] # Example Churn data
})

# Create a histogram
plt.figure(figsize=(10, 6)) # Set the figure size
sns.histplot(x="tenure", data=data, bins=7, hue="Churn", multiple="stack", kde=True)
plt.title('Distribution of Tenure by Churn', fontsize=17) # Title of the plot
plt.xlabel('Tenure', fontsize=13) # X-axis Label
plt.ylabel('Frequency', fontsize=13) # Y-axis Label
plt.legend(title='Churn') # Legend title
plt.show() # Show the plot
```

No handles with labels found to put in legend.

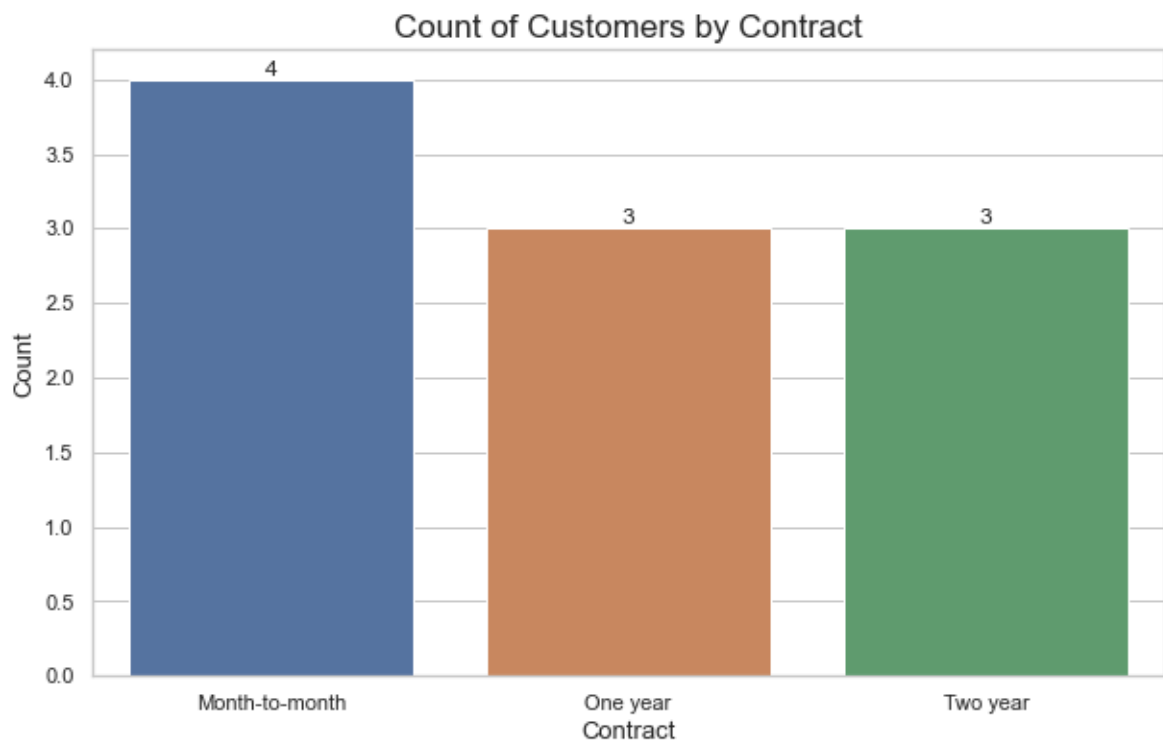


```
In [181]: # Sample data creation for illustration
data = pd.DataFrame({
    'Contract': ['Month-to-month', 'One year', 'Two year', 'Month-to-month', 'One year',
                'Two year', 'Month-to-month', 'Two year', 'One year', 'Month-to-month']
})

# Create a count plot
plt.figure(figsize=(10, 6)) # Set the figure size
ax = sns.countplot(x="Contract", data=data) # Create count plot

# Adding Labels on the bars
for container in ax.containers:
    ax.bar_label(container) # Add Labels to each bar

# Add title and show the plot
plt.title("Count of Customers by Contract", fontsize=17) # Title of the plot
plt.xlabel("Contract", fontsize=13) # X-axis Label
plt.ylabel("Count", fontsize=13) # Y-axis Label
plt.show() # Show the plot
```



```

In [182]: # Sample data creation for illustration with an added 'Churn' column
data = pd.DataFrame({
    'PhoneService': ['Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes'],
    'MultipleLines': ['No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'Yes'],
    'InternetService': ['DSL', 'Fiber optic', 'DSL', 'DSL', 'Fiber optic', 'DSL', 'Fiber optic', 'DSL', 'DSL', 'DSL'],
    'OnlineSecurity': ['Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes'],
    'OnlineBackup': ['No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No'],
    'DeviceProtection': ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'],
    'TechSupport': ['No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes'],
    'StreamingTV': ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'],
    'StreamingMovies': ['No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'Yes'],
    'Churn': ['Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes'] # Added Churn column
})

# List of columns to plot
columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
           'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create a 3x3 grid of subplots
fig, axes = plt.subplots(3, 3, figsize=(15, 15)) # Adjust grid and figure size as needed
fig.suptitle("Count Plots of Customer Services by Churn", fontsize=20)

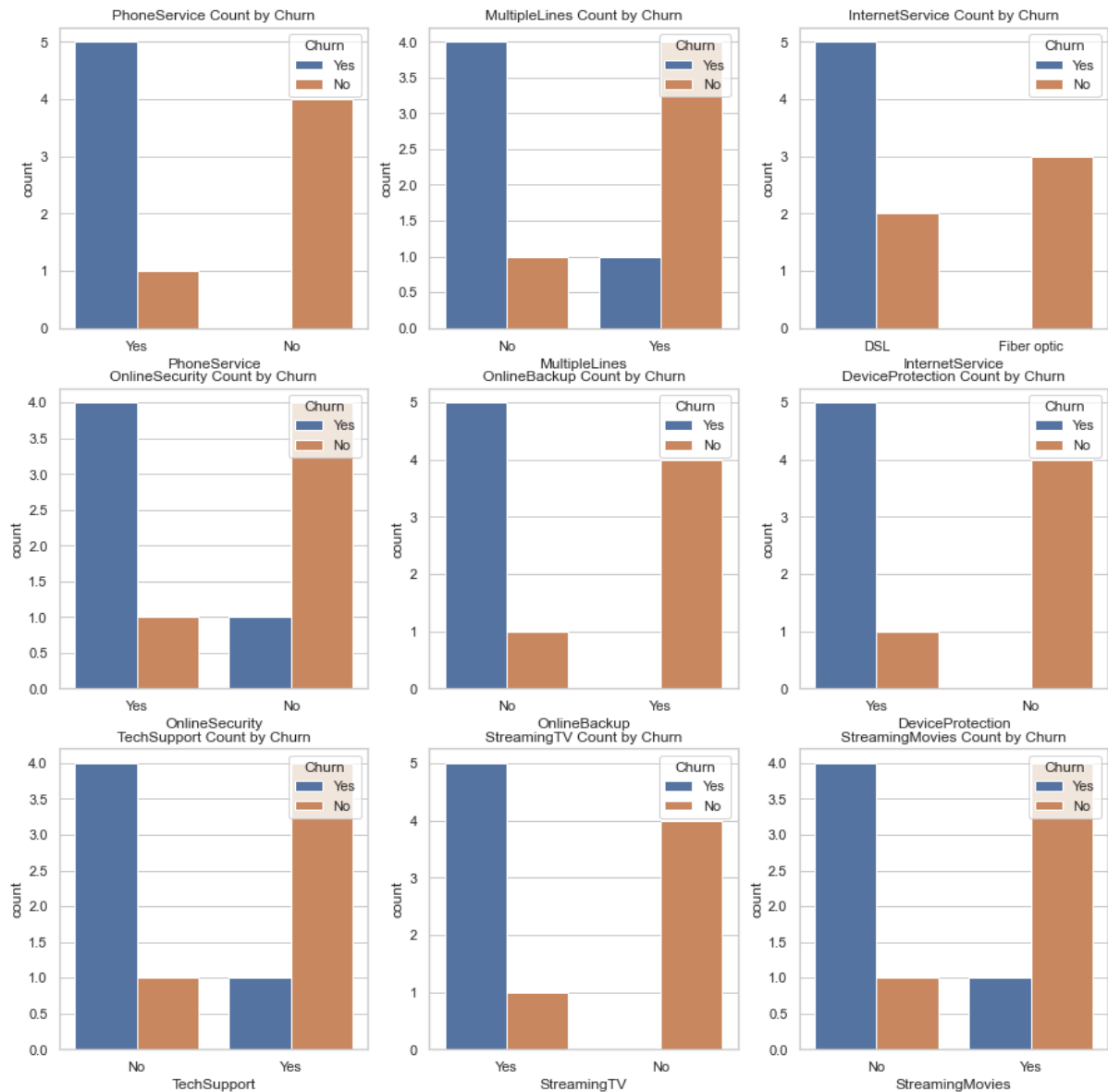
# Flatten the axes array for easy iteration
axes = axes.flatten()

# Loop through each column and plot with hue
for i, col in enumerate(columns):
    sns.countplot(x=col, data=data, hue='Churn', ax=axes[i]) # Added hue='Churn'
    axes[i].set_title(f'{col} Count by Churn')
    axes[i].legend(title='Churn', loc='upper right')

# Hide any unused subplots (in case there are fewer than 9 columns)

```

Count Plots of Customer Services by Churn



The visualizations display count plots illustrating customer churn status (Yes or No) across various service features, including PhoneService, MultipleLines, InternetService, and OnlineSecurity. A trend is observed where customers who do not churn often have active subscriptions to services like PhoneService, DSL InternetService, and OnlineSecurity. In contrast, higher churn rates are associated with customers who lack services such as OnlineBackup, TechSupport, and StreamingTV, suggesting these services may play a role in customer retention. The charts highlight a potential relationship between service usage and customer loyalty, providing insights into how service availability might influence churn behavior.

```
In [183]: import matplotlib.pyplot as plt
import seaborn as sns

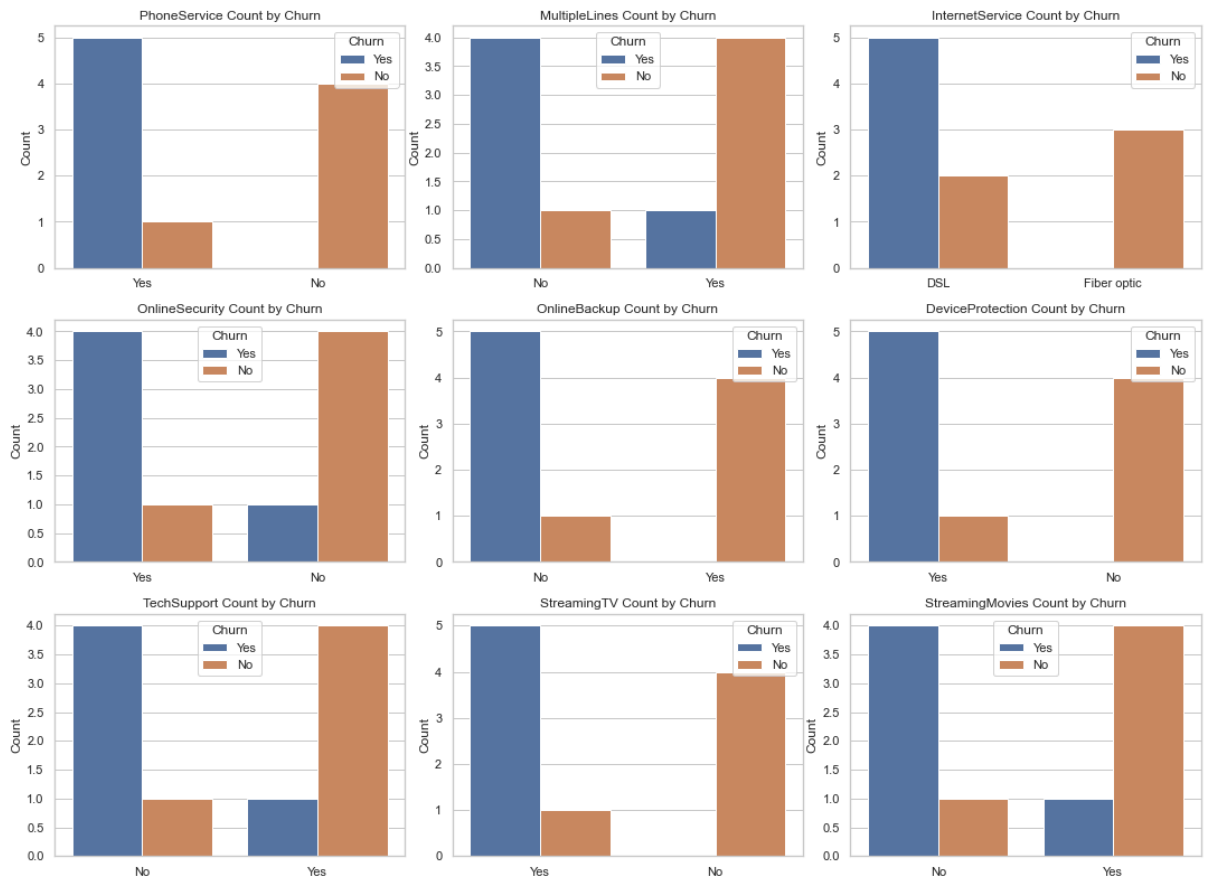
# List of columns to plot (update as per your dataset)
columns_to_plot = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
                  'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV']

# Set the figure size and layout for subplots
fig, axes = plt.subplots(3, 3, figsize=(15, 12)) # Adjust grid size as needed
fig.suptitle("Customer Churn by Service Features", fontsize=16)

# Loop over columns and create count plots for each one
for col, ax in zip(columns_to_plot, axes.flatten()):
    sns.countplot(x=col, data=data, hue="Churn", ax=ax)
    ax.set_title(f"{col} Count by Churn")
    ax.set_xlabel("") # Remove x-axis label for clarity
    ax.set_ylabel("Count")

# Adjust layout and show plot
plt.tight_layout(rect=[0, 0, 1, 0.96]) # Adjust for main title
plt.show()
```

Customer Churn by Service Features



END

In []: