

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('Customer_Churn.csv')
df.head()
```

```
↗
```

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

5 rows × 21 columns

```
df.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
```

Replacing blanks with 0 as tenure is 0 and no total charges are recorded. And changing datatype as float.


```
df["TotalCharges"]=df["TotalCharges"].replace(" ", "0")
df["TotalCharges"]=df["TotalCharges"].astype(float)
```

```
df.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
```


```
df.isnull()
```



| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity |
|------|------------|--------|---------------|---------|------------|--------|--------------|---------------|-----------------|----------------|
| 0 | False | False | False | False | False | False | False | False | False | False |
| 1 | False | False | False | False | False | False | False | False | False | False |
| 2 | False | False | False | False | False | False | False | False | False | False |
| 3 | False | False | False | False | False | False | False | False | False | False |
| 4 | False | False | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 7038 | False | False | False | False | False | False | False | False | False | False |
| 7039 | False | False | False | False | False | False | False | False | False | False |
| 7040 | False | False | False | False | False | False | False | False | False | False |
| 7041 | False | False | False | False | False | False | False | False | False | False |
| 7042 | False | False | False | False | False | False | False | False | False | False |

7043 rows × 21 columns

```
df.isnull().sum()
```



| | |
|------------------|---|
| | 0 |
| 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

```
df.describe()
```

| | 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 |

```
df.duplicated().sum()
```

```
0
```

```
df["customerID"].duplicated().sum()
```

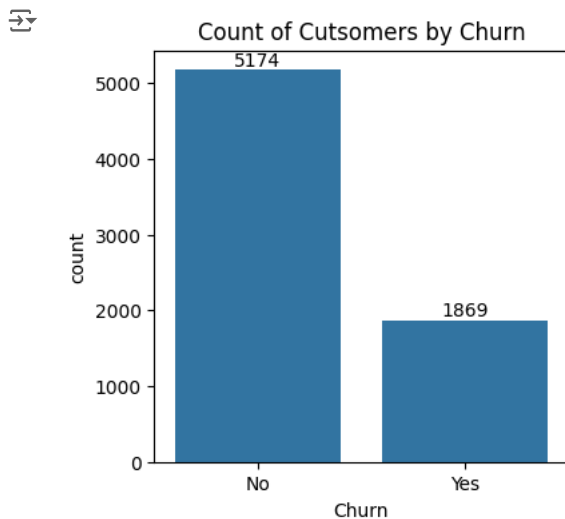
```
0
```

```
def conv(value):
    if value==1:
        return "Yes"
    else:
        return "No"
```

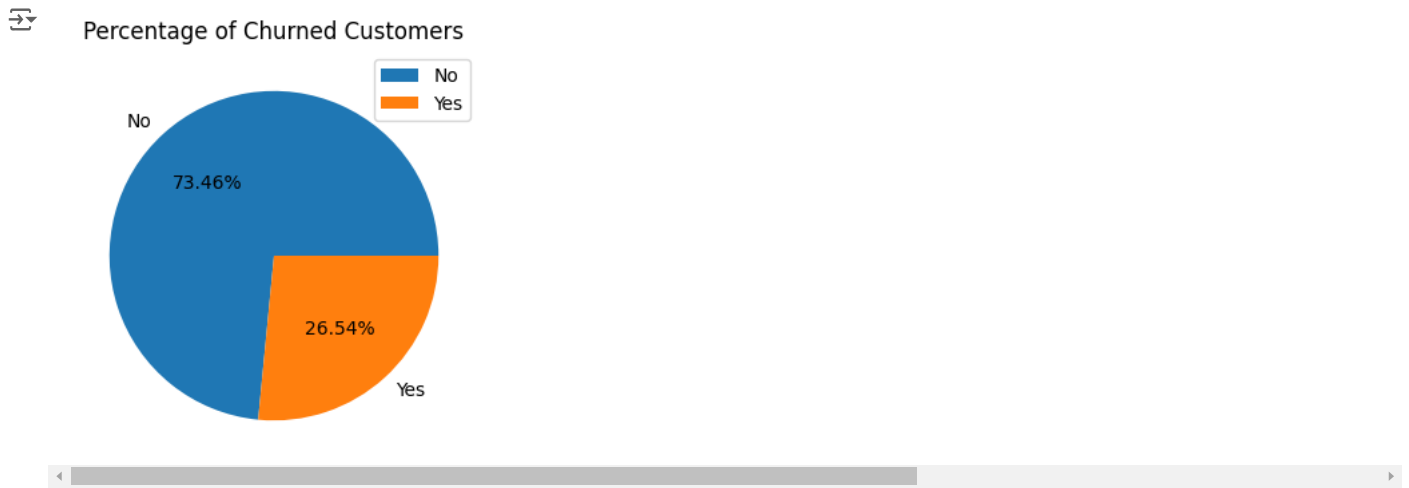
```
df["SeniorCitizen"]=df["SeniorCitizen"].apply(conv)
```

Converted 0 and 1 values of senior citizen into no/yes to understand the data easily.

```
plt.figure(figsize=(4,4))
ax = sns.countplot(x=df['Churn'])
ax.bar_label(ax.containers[0])
plt.title('Count of Cutsomers by Churn')
plt.show()
```



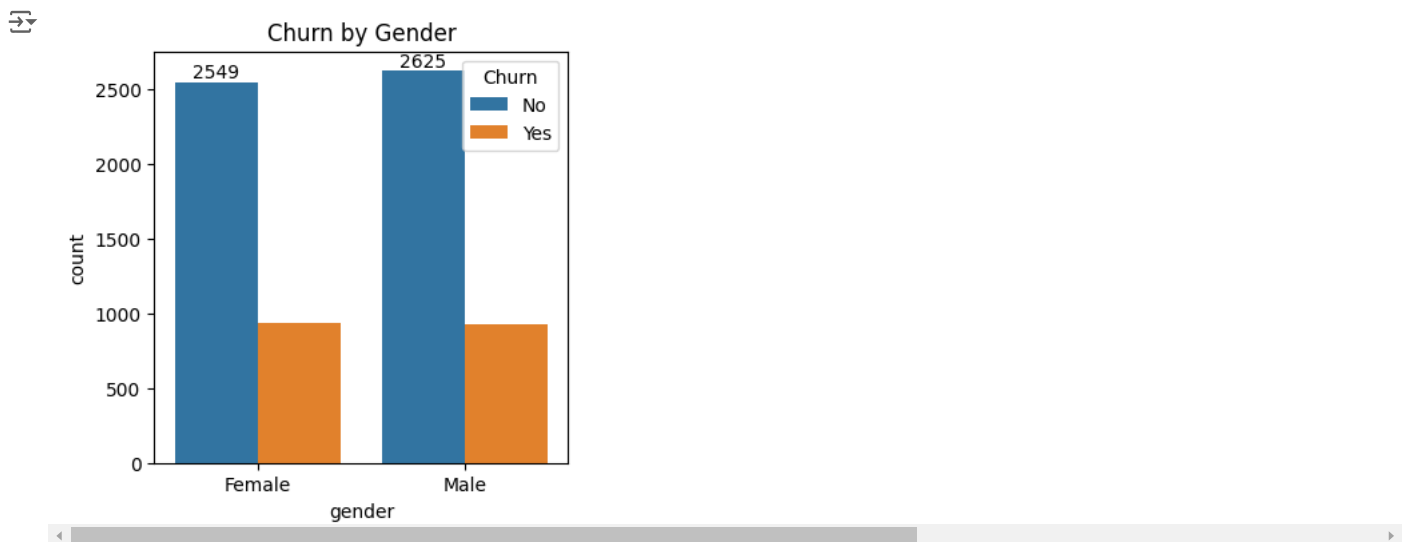
```
plt.figure(figsize=(4,4))
gb = df.groupby('Churn').agg({'Churn': 'count'})
plt.pie(gb['Churn'], labels = gb.index, autopct="%1.2f%%")
plt.legend()
plt.title('Percentage of Churned Customers')
plt.show()
```



From the given pie chart we can conclude that 26.54% customers have churned out.

Let's start exploring by Gender first.

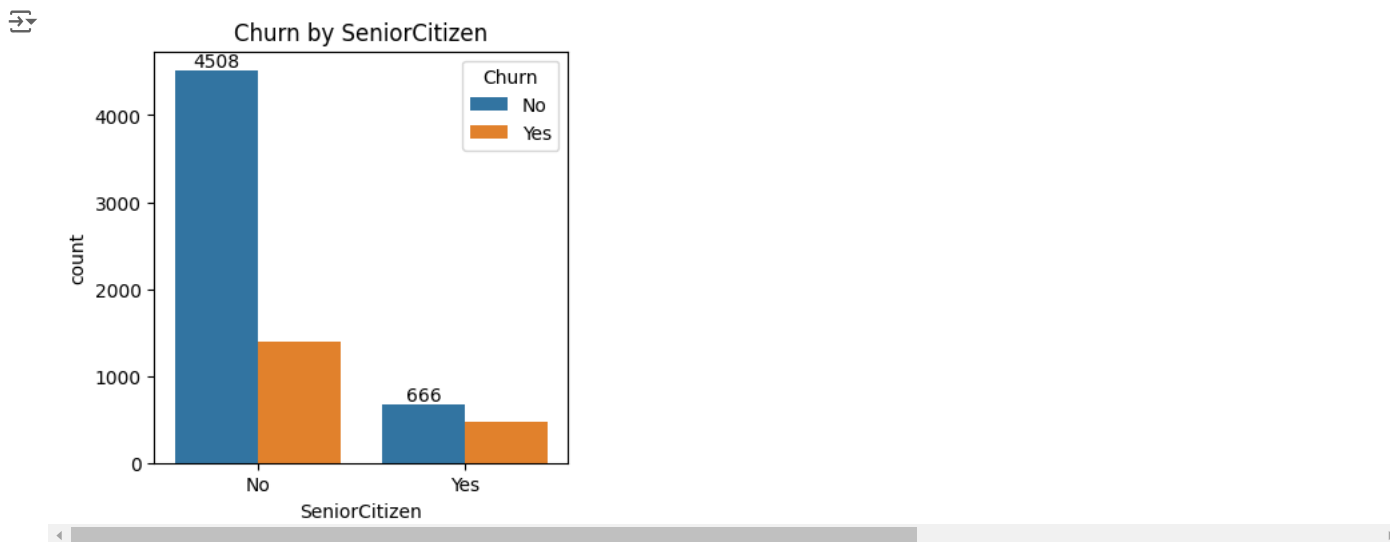
```
plt.figure(figsize=(4,4))
bx=sns.countplot(x=df['gender'],data=df,hue='Churn')
bx.bar_label(bx.containers[0])
plt.title('Churn by Gender')
plt.show()
```



From the above plot we can say that almost equal customers of male and female customers are churned out. So, Gender is not the reason for churned out customers.

Now let's check for Senior Citizen

```
plt.figure(figsize=(4,4))
bx=sns.countplot(x=df['SeniorCitizen'],data=df,hue='Churn')
bx.bar_label(bx.containers[0])
plt.title('Churn by SeniorCitizen')
plt.show()
```

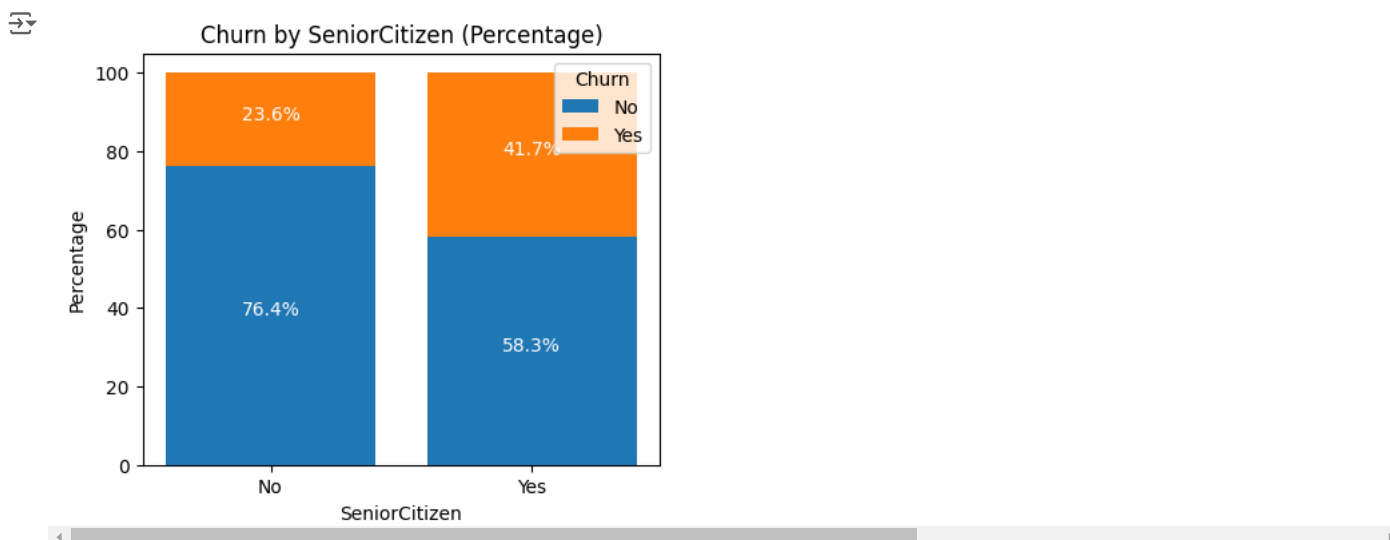


```
# Calculate percentages
percentages=df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)
percentages=percentages.div(percentages.sum(axis=1), axis=0) * 100

# Plotting
fig, ax = plt.subplots(figsize=(5, 4))
bottom = [0] * len(percentages)

for churn_status in percentages.columns:
    ax.bar(percentages.index,percentages[churn_status],label=churn_status,bottom=bottom)
    for i, value in enumerate(percentages[churn_status]):
        if value > 0:
            # Corrected: Use .iloc for accessing the value
            ax.text(i, bottom[i] + value / 2, f"{value:.1f}%", ha='center', color='white')
    bottom += percentages[churn_status].values # Ensure compatibility with array operations

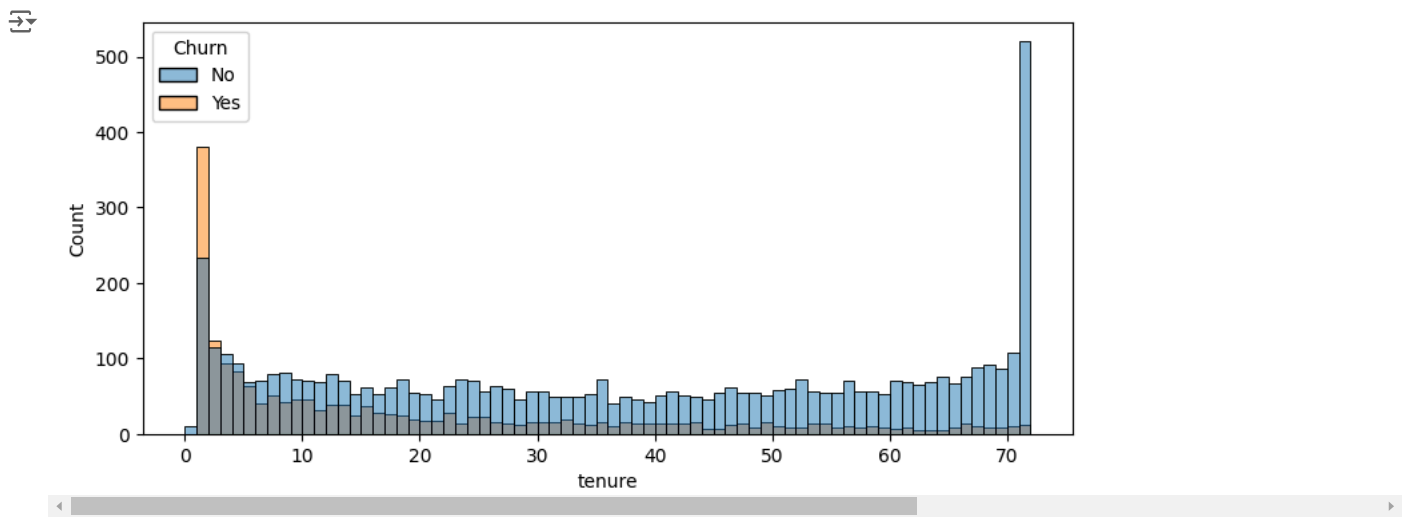
# Customizing
ax.set_title('Churn by SeniorCitizen (Percentage)')
ax.set_xlabel('SeniorCitizen')
ax.set_ylabel('Percentage')
ax.set_xticks(percentages.index)
ax.legend(title='Churn')
plt.show()
```



Comparatively a greater percentage of people in senior citizen have churned out.

Now lets analyze based on tenure

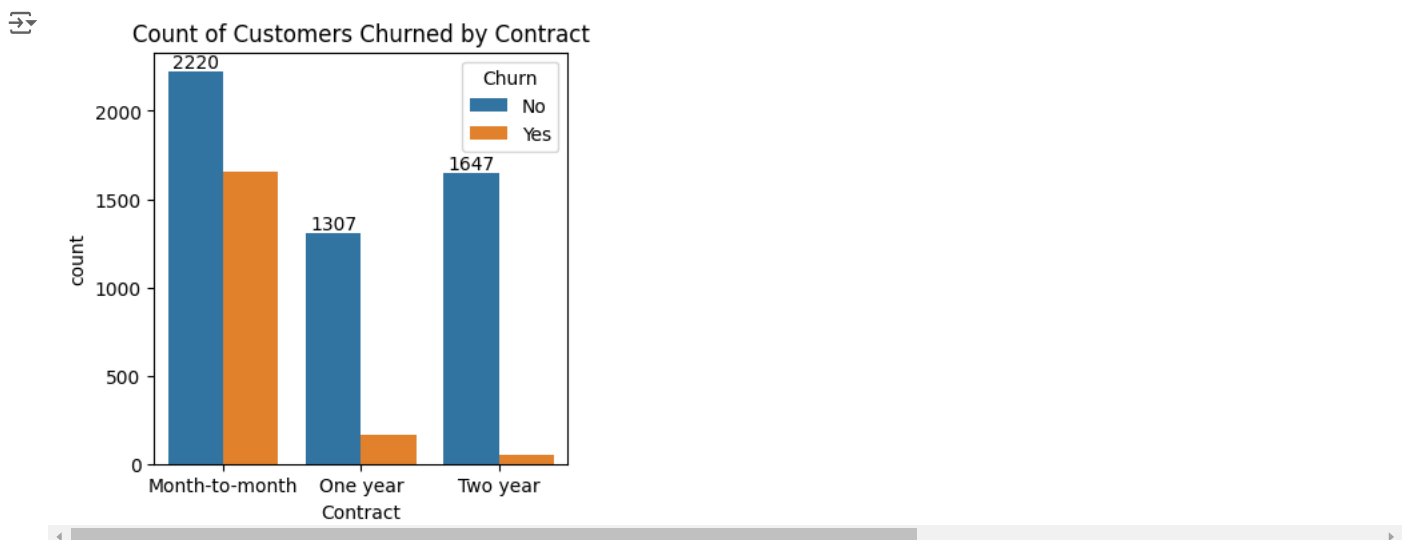
```
plt.figure(figsize=(9,4))
sns.histplot(x=df['tenure'],data=df,bins=72,hue='Churn')
plt.show()
```



People who have used our services for long time have stayed and people who have used our services for 1 or 2 months have churned.

Now lets analyze based on contract.

```
plt.figure(figsize=(4,4))
bx=sns.countplot(x=df['Contract'],data=df,hue='Churn')
bx.bar_label(bx.containers[0])
plt.title('Count of Customers Churned by Contract')
plt.show()
```



People who have month to month contract are likely to churn then from those who have one or two years of contract.

```
df.columns.values

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

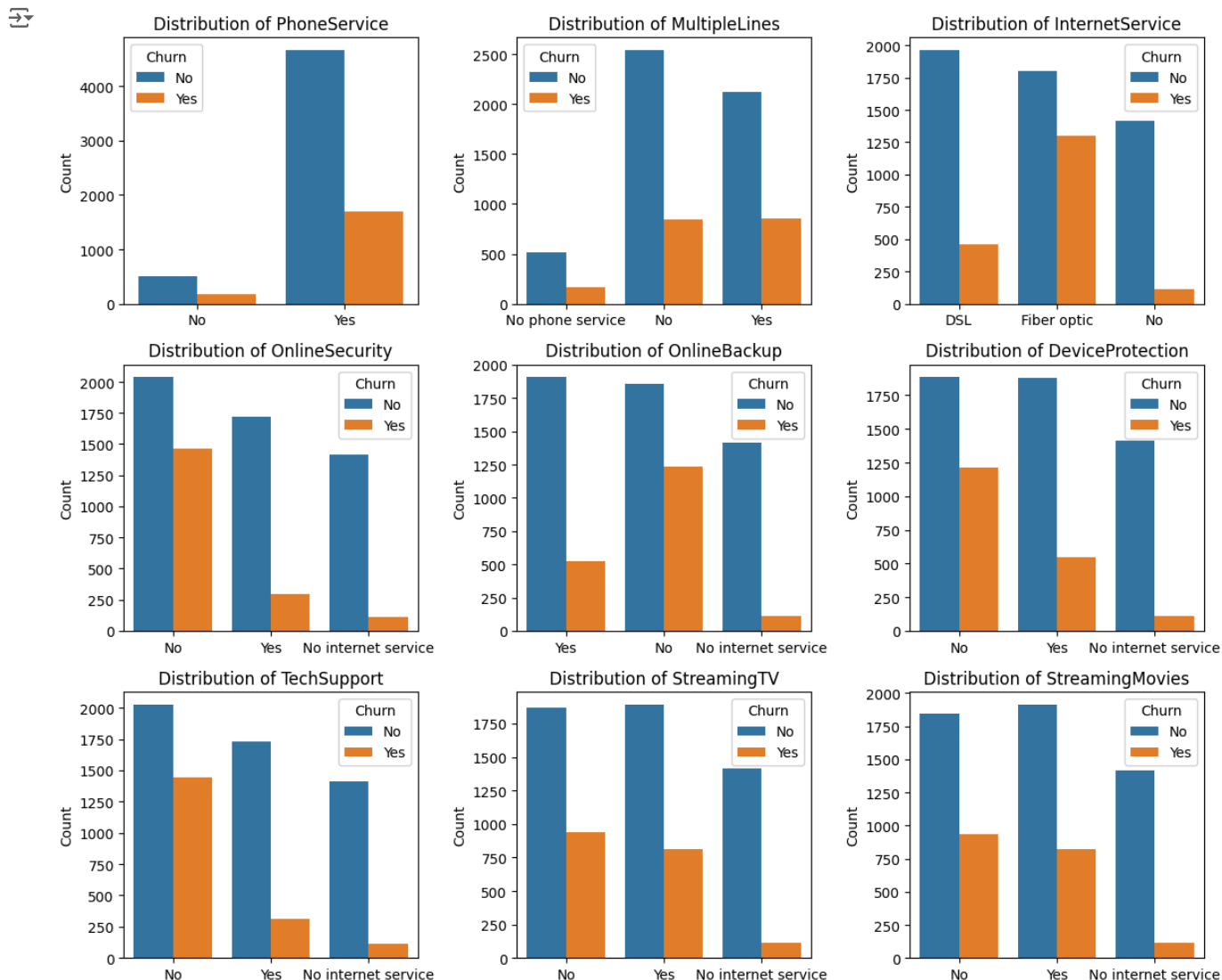
# Define the columns to plot
columns_to_plot = ['PhoneService', 'MultipleLines', 'InternetService',
                  'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
                  'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(12, 10))
axes = axes.flatten()

for i, col in enumerate(columns_to_plot):
    sns.countplot(data=df, x=col, ax=axes[i], hue='Churn')
    axes[i].set_title(f'Distribution of {col}')
    axes[i].set_xlabel('') # Remove x-axis label for clarity
    axes[i].set_ylabel('Count')
```

```
# Remove any empty subplot frames (if less than total subplots)
for j in range(len(columns_to_plot), len(axes)):
    fig.delaxes(axes[j])
```

```
# Adjust layout
plt.tight_layout()
plt.show()
```



Most customers have a phone service, but churn is higher among customers with "Multiple Lines" compared to those with a single line or no service. Customers using Fiber optic internet have a higher churn rate compared to those using DSL or no internet service. Services like OnlineSecurity, OnlineBackup, and TechSupport show higher churn when customers do not subscribe to these features. Churn rates are higher for customers who use streaming services like StreamingTV and StreamingMovies, particularly among those with active internet connections.

```
plt.figure(figsize=(10,4))
bx=sns.countplot(x=df[ 'PaymentMethod'],data=df,hue='Churn')
bx.bar_label(bx.containers[1])
plt.title('Count of Customers Churned by Payment Method')
plt.show()
```

