



# Telco Customers Churn.

---

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: df = pd.read_csv('Customer_Churn.csv')
df
```

```
Out[2]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	Phone5
0	7590-VHVEG	Female	0	Yes	No	1	
1	5575-GNVDE	Male	0	No	No	34	
2	3668-QPYBK	Male	0	No	No	2	
3	7795-CFOCW	Male	0	No	No	45	
4	9237-HQITU	Female	0	No	No	2	
...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	
7039	2234-XADUH	Female	0	Yes	Yes	72	
7040	4801-JZAZL	Female	0	Yes	Yes	11	
7041	8361-LTMKD	Male	1	Yes	No	4	
7042	3186-AJIEK	Male	0	No	No	66	

7043 rows × 21 columns

```
In [3]: 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 blank with 0 as tenure is 0 and no total charges are recorded.

```

In [4]: df["TotalCharges"] = df["TotalCharges"].replace(" ", "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")

```

```

In [5]: 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

```

```
In [6]: df.isnull().sum()
```

```

Out[6]: 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 [7]: df.describe()
```

```
Out[7]:
```

	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 [8]: df.duplicated().sum()
```

```
Out[8]: np.int64(0)
```

Check customerID duplications.

```
In [9]: df["customerID"].duplicated().sum()
```

```
Out[9]: np.int64(0)
```

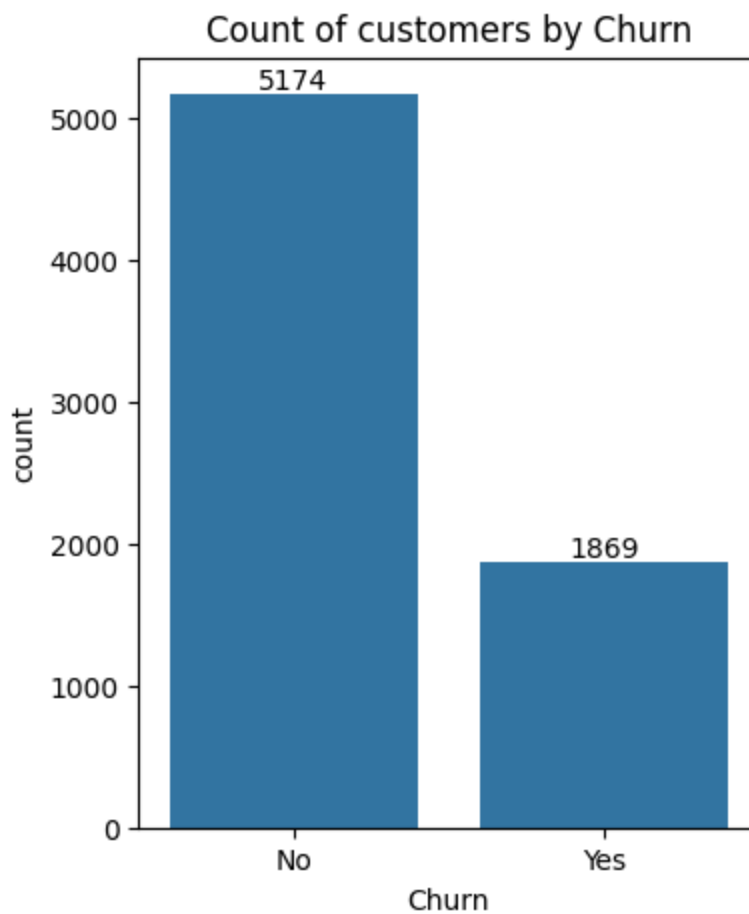
Convert 0 & 1 values of SeniorCitizen to Yes/No to make it easier to understand

```
In [10]: def conv(value):
          if value == 1:
              return "Yes"
          else:
              return "No"

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

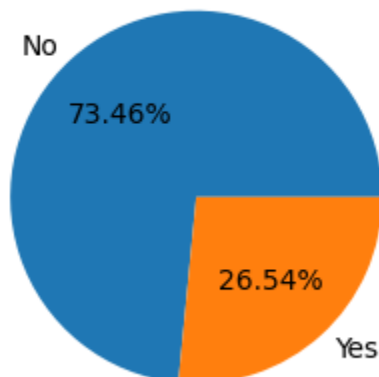
```
In [11]: plt.figure(figsize = (4,5))
          ax = sns.countplot(x = 'Churn', data = df)

          ax.bar_label(ax.containers[0])
          plt.title("Count of customers by Churn")
          plt.show()
```



```
In [12]: plt.figure(figsize = (3,4))  
gb = df.groupby("Churn").agg({'Churn' : 'count'})  
  
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")  
plt.title("Percentage of Churned Customers", fontsize = 10)  
plt.show()
```

Percentage of Churned Customers

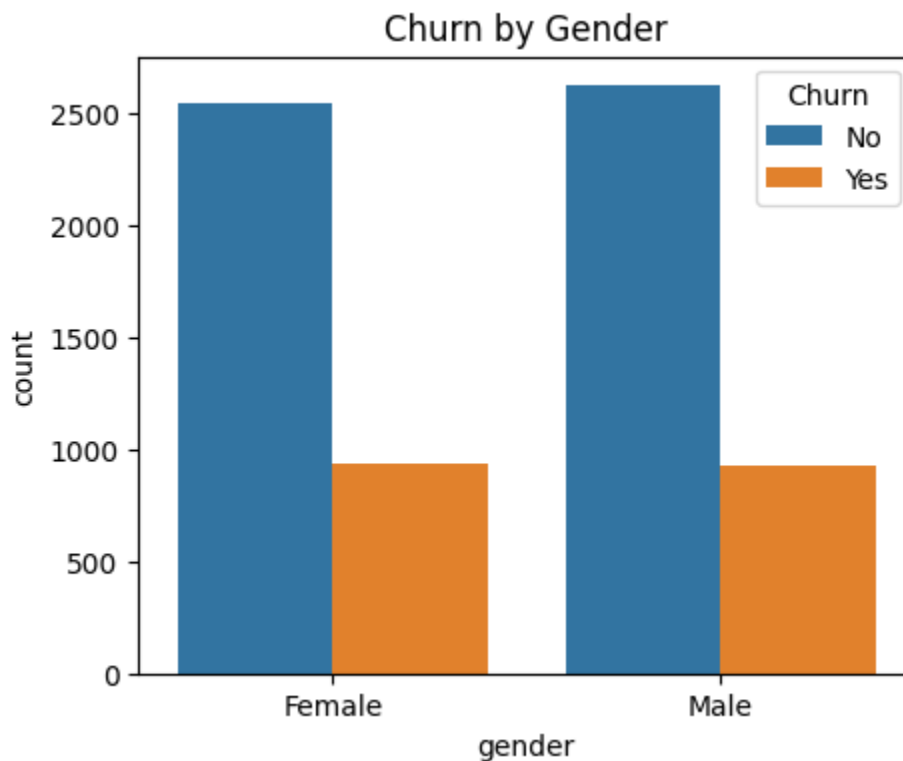


From the given pie chart we can conclude that 26.54% of our customers have churned out.

---

Now let's explore the reason behind it.

```
In [13]: plt.figure(figsize = (5 , 4))
sns.countplot(x = "gender", data =df, hue = "Churn")
plt.title("Churn by Gender")
plt.show()
```



```
In [14]: data_counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)
data_percent = data_counts.div(data_counts.sum(axis=1), axis=0) * 100

fig, ax = plt.subplots(figsize=(4,4))
bottom = [0, 0]

for idx, column in enumerate(data_percent.columns):
    ax.bar(data_percent.index, data_percent[column], bottom=bottom, label=column)

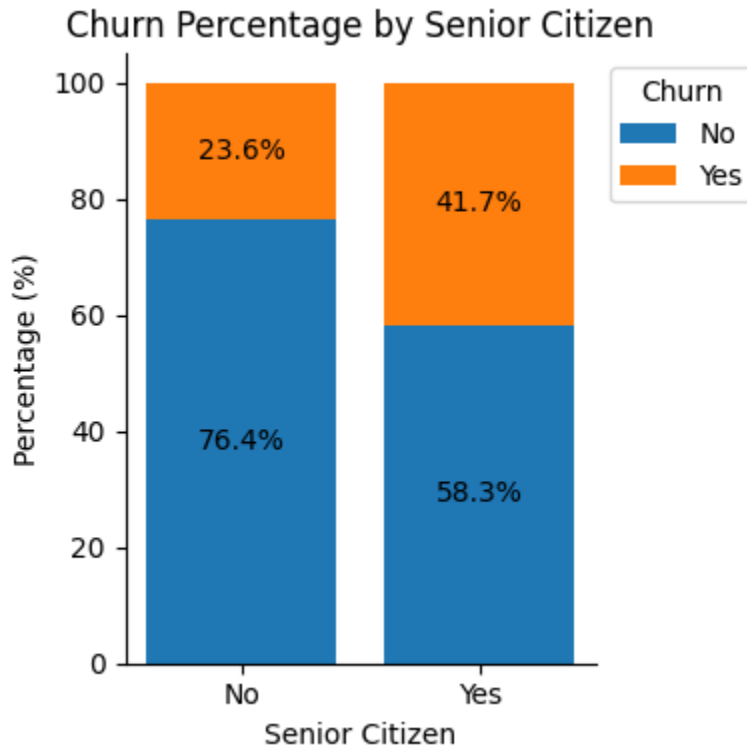
    for i, val in enumerate(data_percent[column]):
        if val > 0:
            ax.text(i, bottom[i] + val / 2, f'{val:.1f}%', ha='center', va='center')
            bottom = [i + j for i, j in zip(bottom, data_percent[column])]

ax.set_title("Churn Percentage by Senior Citizen")
ax.set_xlabel("Senior Citizen")
```

```

ax.set_ylabel("Percentage (%)")
ax.set_xticks([0, 1])
ax.set_xticklabels(["No", "Yes"])
ax.legend(title="Churn", bbox_to_anchor = (1,1))
sns.despine()
plt.tight_layout()
plt.show()

```

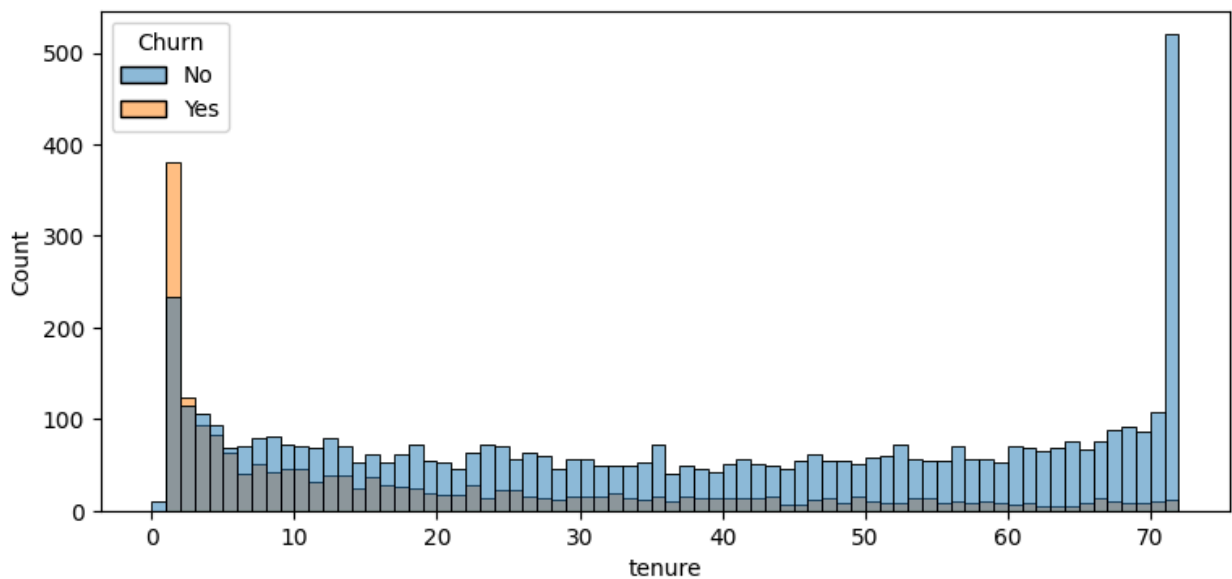


Comparative a greater percentage of people in senior citizen category have churned.

```

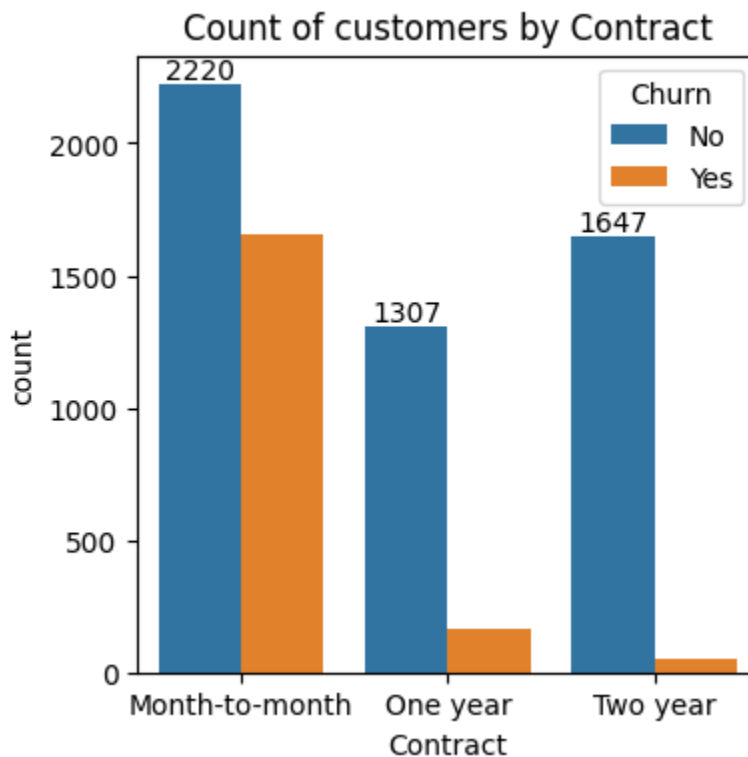
In [15]: plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()

```



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

```
In [16]: plt.figure(figsize = (4,4))
ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of customers by Contract")
plt.show()
```



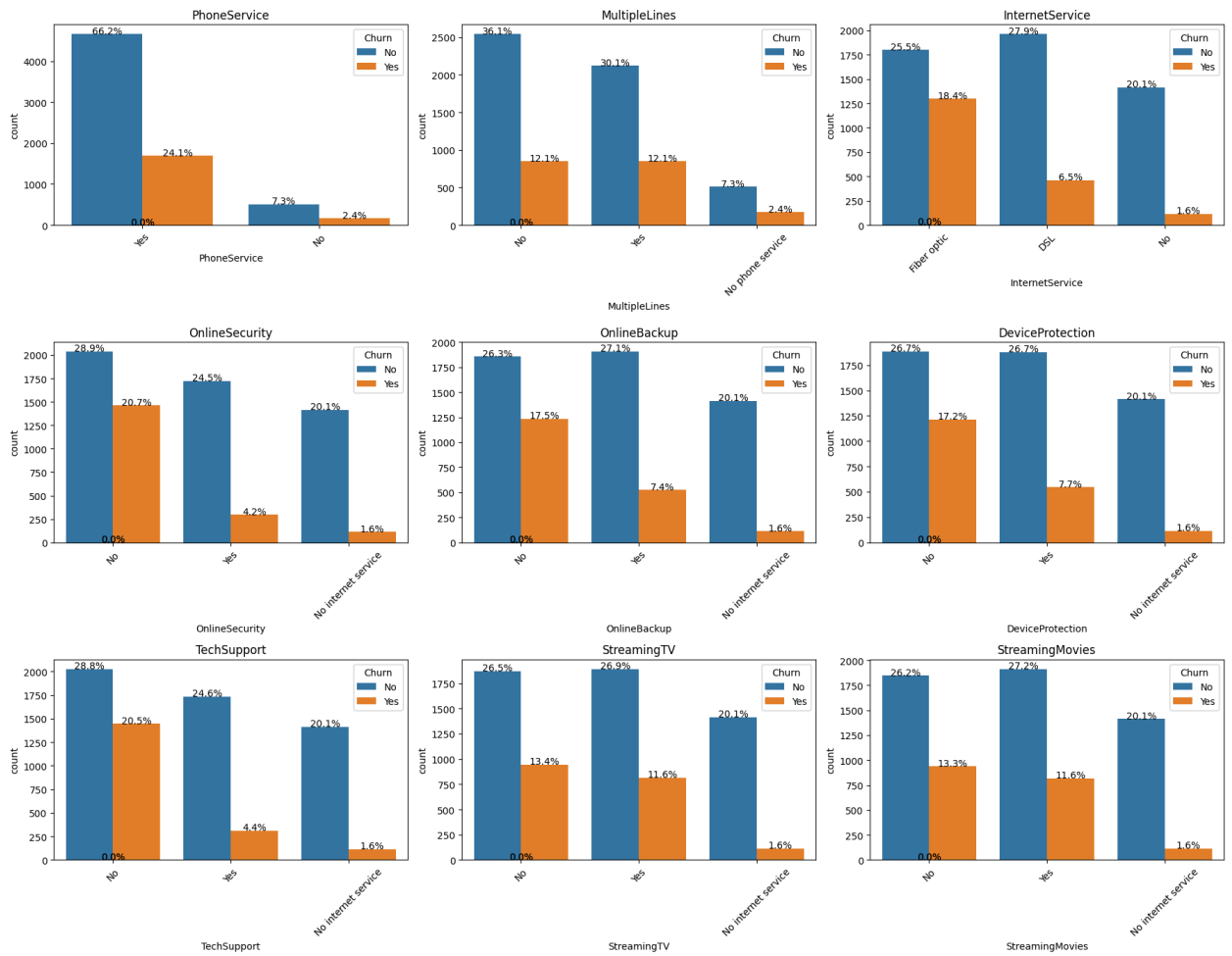


People who have month to month contract are likely to churn then from those who have 1 or 2 years of contract.

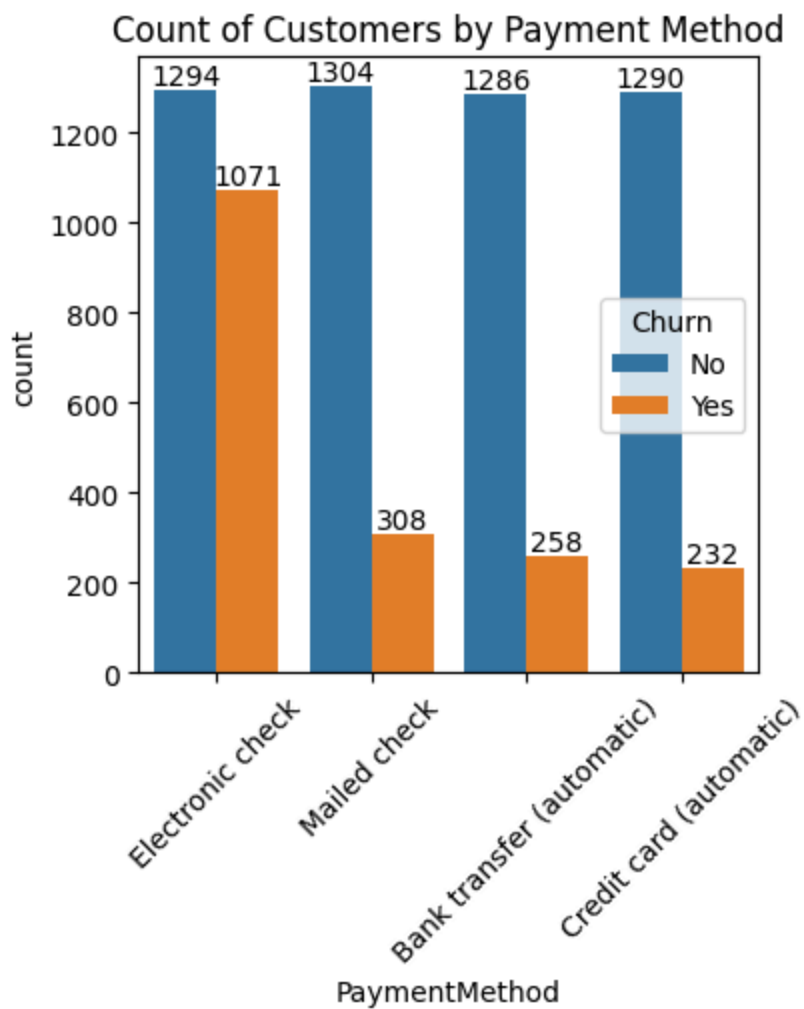
```
In [17]: df.columns.values
```

```
Out[17]: array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',  
               'tenure', 'PhoneService', 'MultipleLines', 'InternetService',  
               'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',  
               'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',  
               'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',  
               'TotalCharges', 'Churn'], dtype=object)
```

```
In [18]: cols = ['PhoneService', 'MultipleLines', 'InternetService',  
                'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',  
                'TechSupport', 'StreamingTV', 'StreamingMovies']  
  
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 14))  
axes = axes.flatten()  
  
total_rows = len(df)  
  
for i, col in enumerate(cols):  
    ax = axes[i]  
    order = df[col].value_counts().index  
    sns.countplot(x=col, data=df, ax=ax, order=order, hue = df["Churn"])  
  
    for p in ax.patches:  
        count = p.get_height()  
        percentage = 100 * count / total_rows  
        ax.text(p.get_x() + p.get_width() / 2, count + 5,  
                f'{percentage:.1f}%', ha='center', fontsize=10)  
  
    ax.set_title(col, fontsize=12)  
    ax.tick_params(axis='x', rotation=45)  
  
plt.tight_layout()  
plt.show()
```



```
In [19]: plt.figure(figsize = (4,4))
ax = sns.countplot(x = "PaymentMethod", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Customers by Payment Method")
plt.xticks(rotation = 45)
plt.show()
```



Customer is likely to churn when he is using electronic check as a payment method.

```
In [22]: plt.figure(figsize = (4,4))
ax = sns.countplot(x = "PaperlessBilling", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Customers by PaperlessBilling")
plt.xticks(rotation = 45)
plt.show()
```

