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

df = pd.read_csv('Customer Churn (1).csv')

df.head()
```

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

      MultipleLines InternetService OnlineSecurity  ...
DeviceProtection  \
0  No phone service             DSL             No  ...
No
1                No             DSL            Yes  ...
Yes
2                No             DSL            Yes  ...
No
3  No phone service             DSL            Yes  ...
Yes
4                No     Fiber optic             No  ...
No

  TechSupport StreamingTV StreamingMovies        Contract
PaperlessBilling  \
0          No          No              No  Month-to-month
Yes
1          No          No              No        One year
No
2          No          No              No  Month-to-month
Yes
3         Yes          No              No        One year
No
4          No          No              No  Month-to-month
Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
```

```
1                Mailed check          56.95        1889.5    No
2                Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4            Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

```
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"
df['SeniorCitizen'] = df['SeniorCitizen'].apply(conv)

df.head()
```

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

      MultipleLines InternetService OnlineSecurity  ...
DeviceProtection  \
0  No phone service             DSL             No  ...
No
1                No             DSL            Yes  ...
Yes
2                No             DSL            Yes  ...
No
3  No phone service             DSL            Yes  ...
Yes
4                No     Fiber optic             No  ...
No

  TechSupport StreamingTV StreamingMovies        Contract
PaperlessBilling  \
0          No          No              No  Month-to-month
Yes
1          No          No              No        One year
No
2          No          No              No  Month-to-month
Yes
```

```
3         Yes          No               No         One year
No
4          No          No               No  Month-to-month
Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

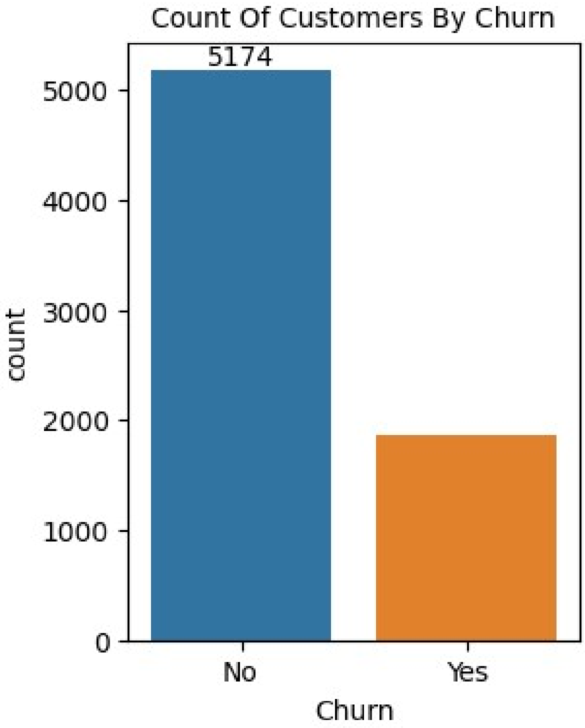
```
df.info()
```

```
<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   object 
 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(1), object(19)
memory usage: 1.1+ MB
```

```
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   object 
 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(1), object(18)
memory usage: 1.1+ MB
```

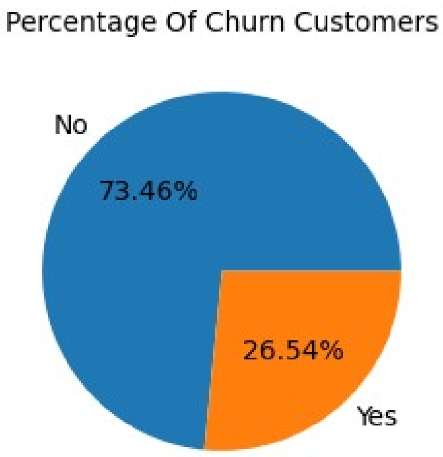
```
plt.figure(figsize = (3,4))
ax = sns.countplot( x = 'Churn' , data = df , hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Count Of Customers By Churn" , fontsize = 10)
plt.show()
```



```
gb = df.groupby("Churn").agg({'Churn' : "count"})
gb
```

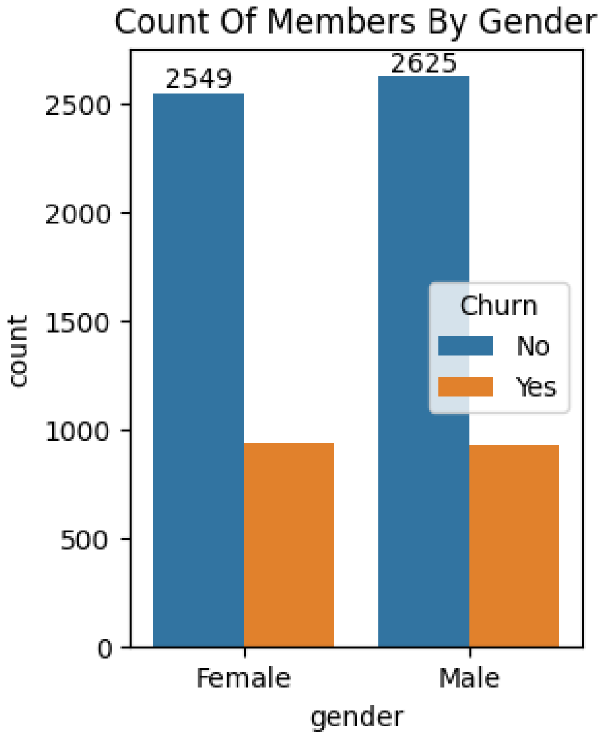
```
       Churn
Churn
No      5174
Yes     1869
```

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

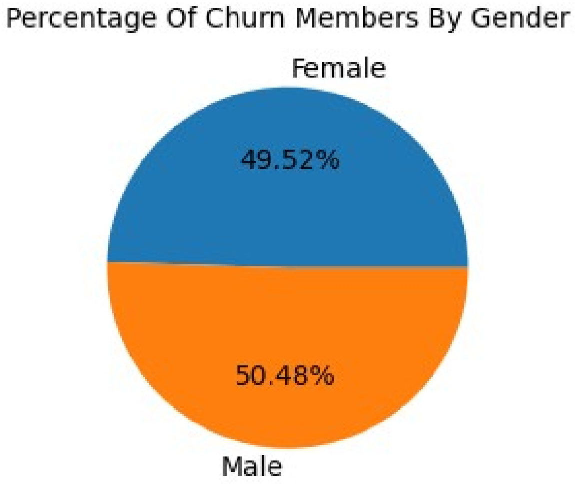


```
       Churn
Churn
No      5174
Yes     1869
```

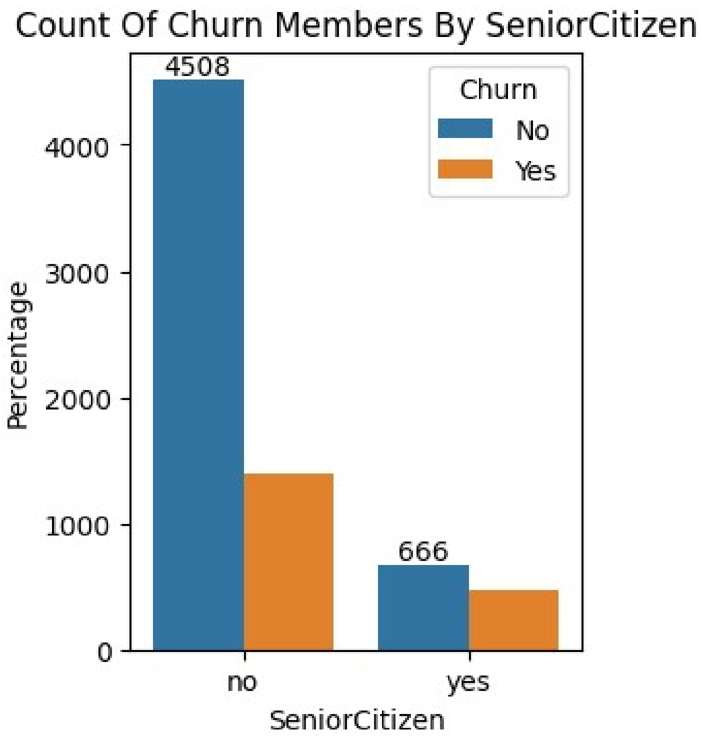
```
plt.figure(figsize = (3,4))
ax = sns.countplot( x = 'gender' , data = df , hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Count Of Members By Gender")
plt.show()
```



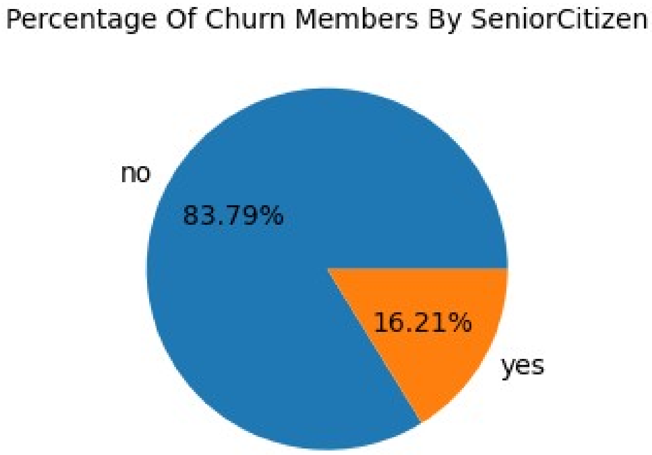
```
plt.figure(figsize = (3,4))
gb = df.groupby("gender").agg({'gender' : "count"})
plt.pie(gb['gender'] , labels = gb.index , autopct = "%1.2f%%")
plt.title("Percentage Of Churn Members By Gender" , fontsize = 10)
plt.show()
```



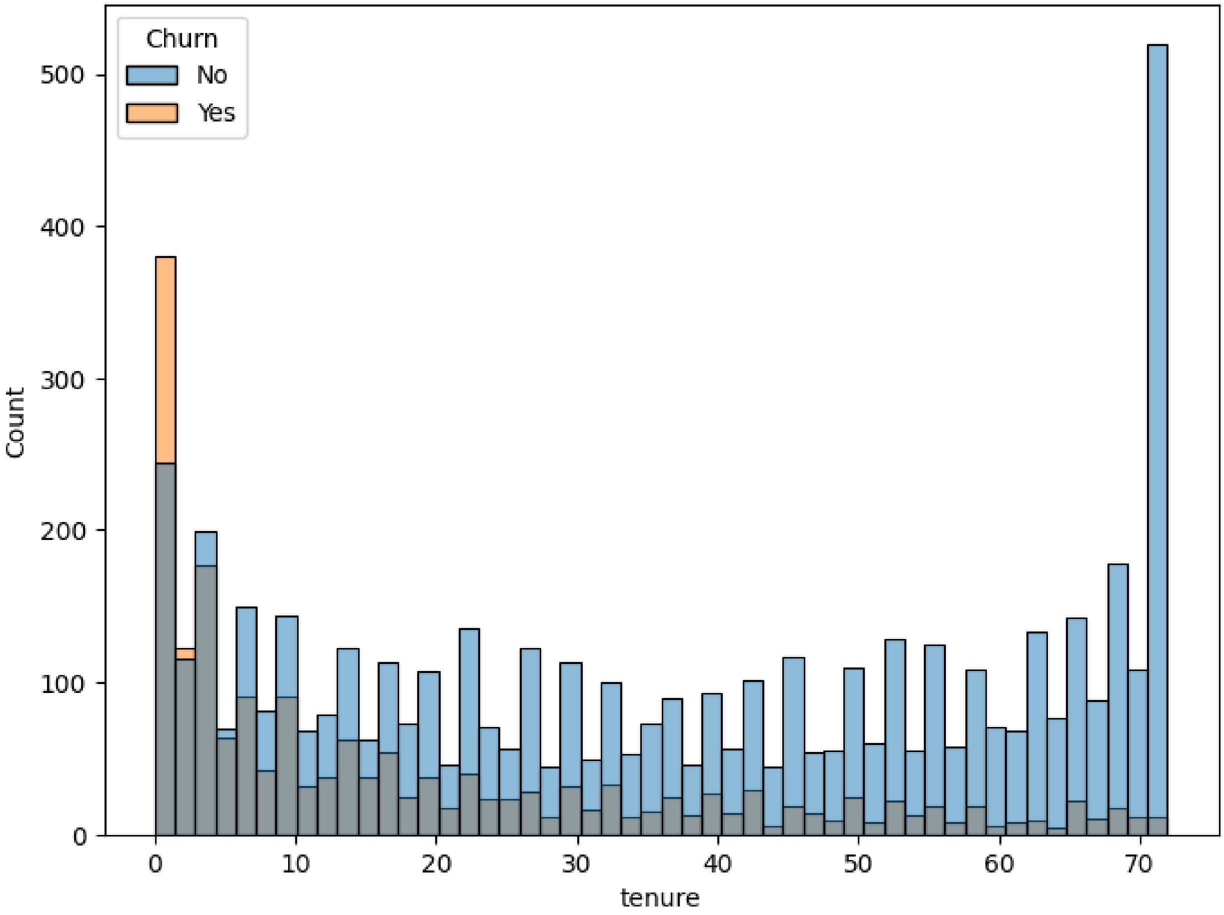
```
plt.figure(figsize = (3,4))
plt.xlabel('SeniorCitizen')
plt.ylabel('Percentage')
ax = sns.countplot( x = 'SeniorCitizen' , data = df , hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Count Of Churn Members By SeniorCitizen")
plt.show()
```



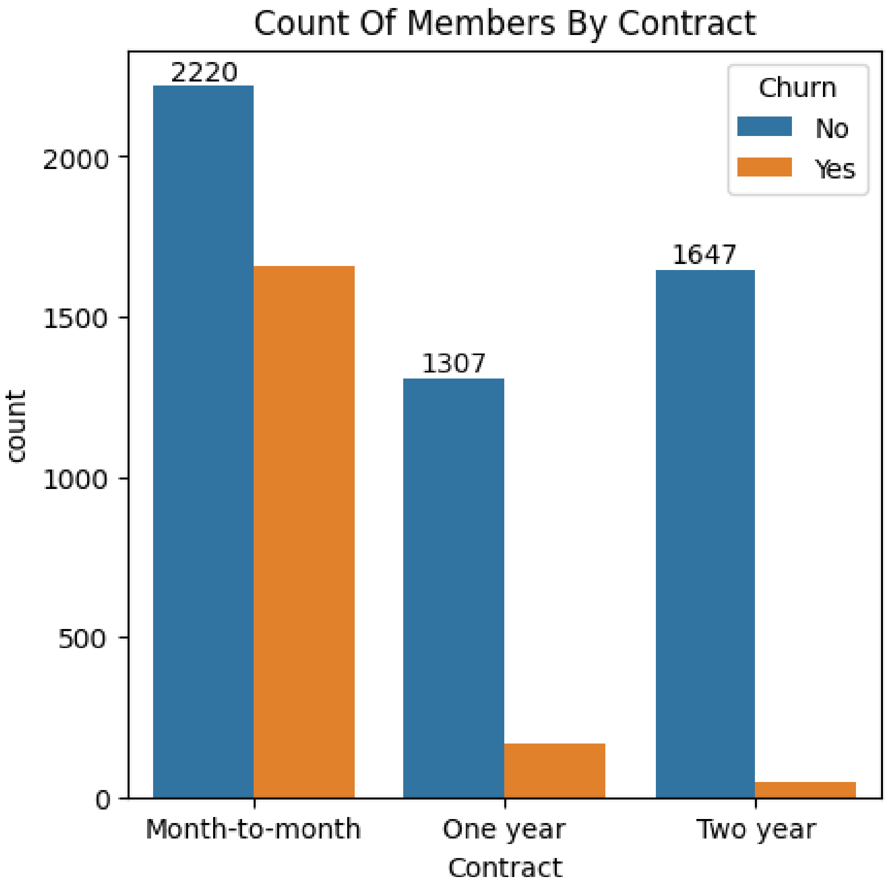
```
plt.figure(figsize = (3,4))
gb = df.groupby("SeniorCitizen").agg({'SeniorCitizen'  : "count"})
plt.pie(gb['SeniorCitizen'] , labels = gb.index , autopct = "%1.2f%%")
plt.title("Percentage Of Churn Members By SeniorCitizen" , fontsize =
10)
plt.show()
```



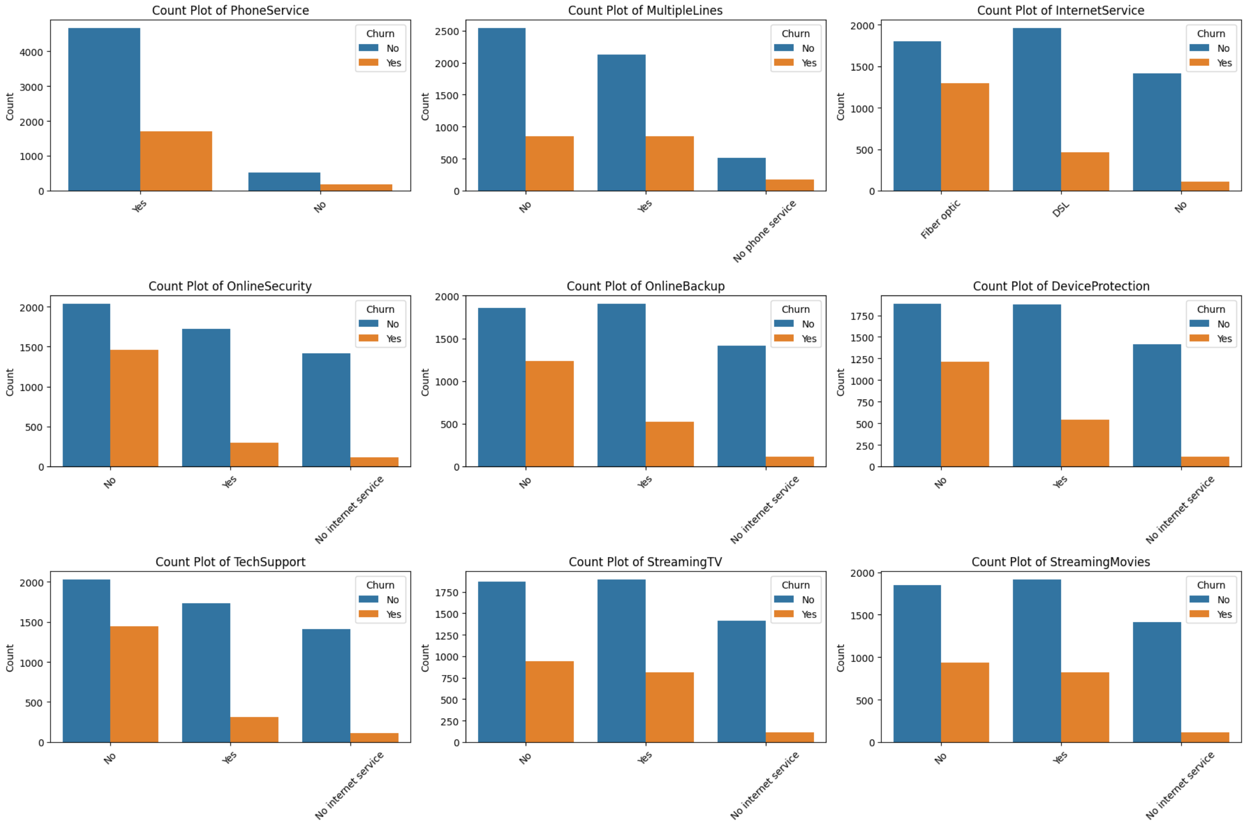
```
plt.figure(figsize = (8,6))
sns.histplot( x = "tenure" , data = df , bins = 50 , hue = 'Churn')
plt.show()
```



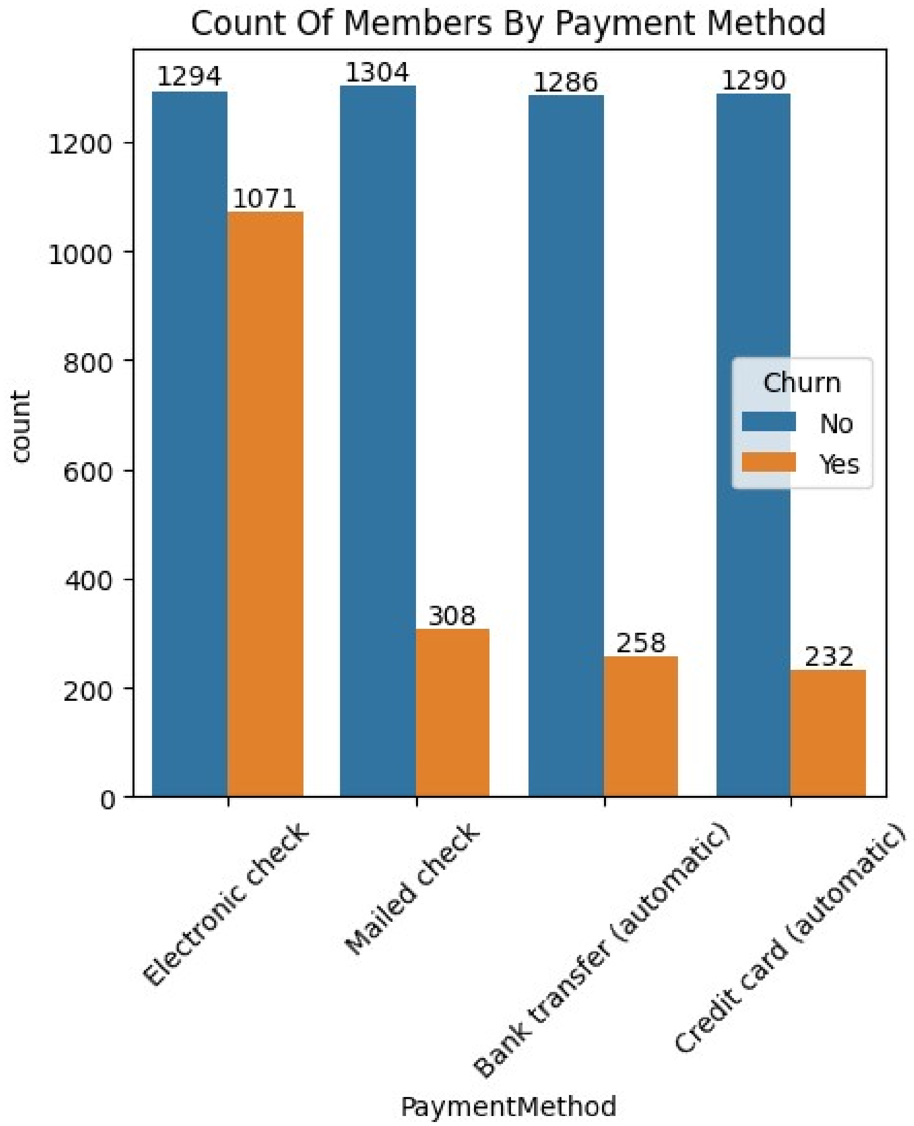
```
plt.figure(figsize = (5,5))
ax = sns.countplot( x = 'Contract' , data = df , hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Count Of Members By Contract")
plt.show()
```



```
# Assuming df is your DataFrame
# List of columns for count plots
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Setup subplot grid
n_cols = 3
n_rows = (len(cols) + n_cols - 1) // n_cols  # Ceiling division
fig, axes = plt.subplots(n_rows, n_cols, figsize=(18, 12))
axes = axes.flatten()

# Plotting each count plot
for i, col in enumerate(cols):
    sns.countplot(data=df, x=col, ax=axes[i],
order=df[col].value_counts().index , hue = df['Churn'])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')
    axes[i].tick_params(axis='x', rotation=45)

# Remove unused axes
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```



```
plt.figure(figsize = (5,5))
ax = sns.countplot( x = 'PaymentMethod' , data = df , hue = 'Churn')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.xticks(rotation = 45)
plt.title("Count Of Members By Payment Method")
plt.show()
```


Count Of Members By Payment Method
count
0
200
400
600
800
1000
1200
1294
1071
1304
308
1286
258
1290
232
Churn
No
Yes
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod