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

df = pd.read_csv("Customer Chur.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]
```

```
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

```
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
```

# Checking Null Value

```
df.isnull()
```

```
      customerID  gender  SeniorCitizen  Partner  Dependents
tenure  \
0          False   False          False    False       False   False

1          False   False          False    False       False   False

2          False   False          False    False       False   False

3          False   False          False    False       False   False

4          False   False          False    False       False   False

...          ...     ...            ...      ...         ...     ...

7038       False   False          False    False       False   False

7039       False   False          False    False       False   False
```

```
7040       False   False          False    False       False   False

7041       False   False          False    False       False   False

7042       False   False          False    False       False   False


      PhoneService  MultipleLines  InternetService
OnlineSecurity  ...  \
0            False          False            False
False  ...
1            False          False            False
False  ...
2            False          False            False
False  ...
3            False          False            False
False  ...
4            False          False            False
False  ...
...            ...            ...              ...              ...  ..
.
7038         False          False            False
False  ...
7039         False          False            False
False  ...
7040         False          False            False
False  ...
7041         False          False            False
False  ...
7042         False          False            False
False  ...

      DeviceProtection  TechSupport  StreamingTV  StreamingMovies
Contract  \
0                False        False        False            False
False
1                False        False        False            False
False
2                False        False        False            False
False
3                False        False        False            False
False
4                False        False        False            False
False
...                ...          ...          ...              ...
...
7038             False        False        False            False
False
7039             False        False        False            False
```

```
False
7040              False         False         False              False
False
7041              False         False         False              False
False
7042              False         False         False              False
False

      PaperlessBilling  PaymentMethod  MonthlyCharges  TotalCharges
Churn
0                False           False           False          False
False
1                False           False           False          False
False
2                False           False           False          False
False
3                False           False           False          False
False
4                False           False           False          False
False
...                ...             ...             ...            ...
...
7038             False           False           False          False
False
7039             False           False           False          False
False
7040             False           False           False          False
False
7041             False           False           False          False
False
7042             False           False           False          False
False

[7043 rows x 21 columns]
```

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

```
np.int64(0)
```

```
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
```

# Checking Duplicated Value

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

```
np.int64(0)
```

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

```
np.int64(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 to yes/no to make it
easier to uderstand
```

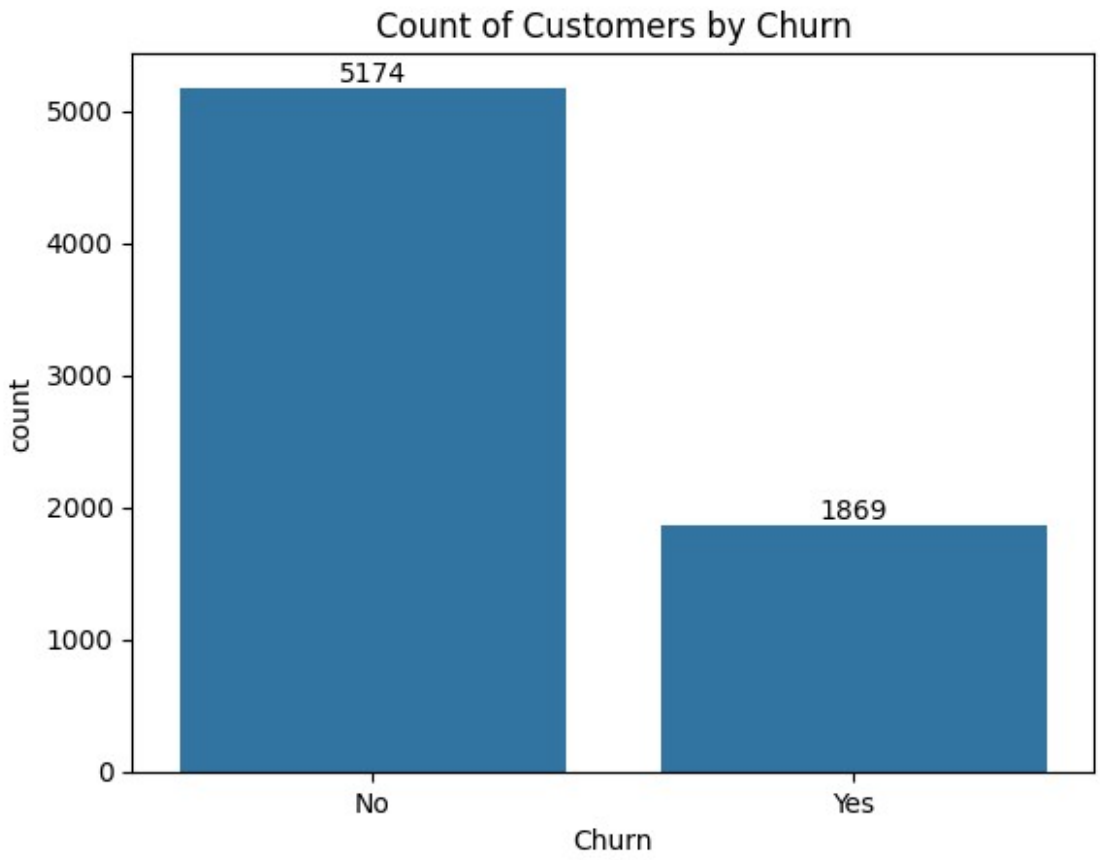
```
df.head()
```

```
   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  \
```

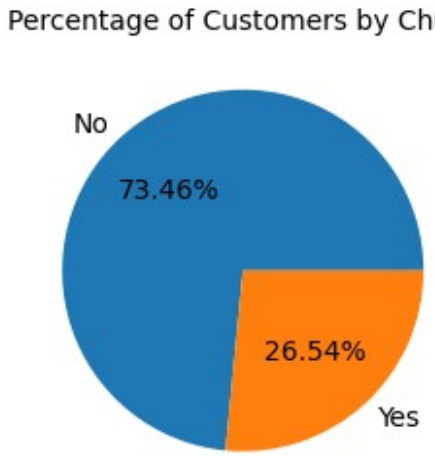
```
0           No           No               No  Month-to-month
Yes
1           No           No               No        One year
No
2           No           No               No  Month-to-month
Yes
3          Yes           No               No        One year
No
4           No           No               No  Month-to-month
Yes

                PaymentMethod MonthlyCharges  TotalCharges  Churn
0            Electronic check          29.85         29.85     No
1                Mailed check          56.95       1889.50     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]
```

# showing data yes/no how may customer have or churn

```
ax = sns.countplot(x = "Churn", data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn")
plt.show()
```



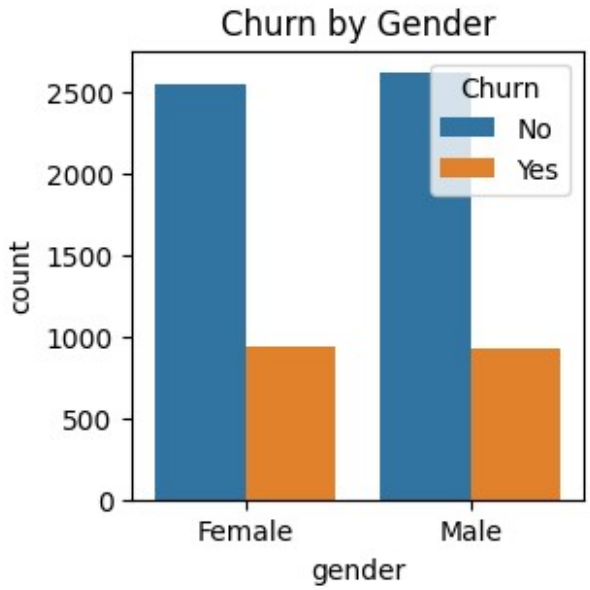

```
plt.figure(figsize = (3,4)) #height and width of pie chart
gb = df.groupby("Churn").agg({"Churn" : "count"})
plt.pie(gb["Churn"], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage of Customers by Churn", fontsize = 10)
plt.show()
```



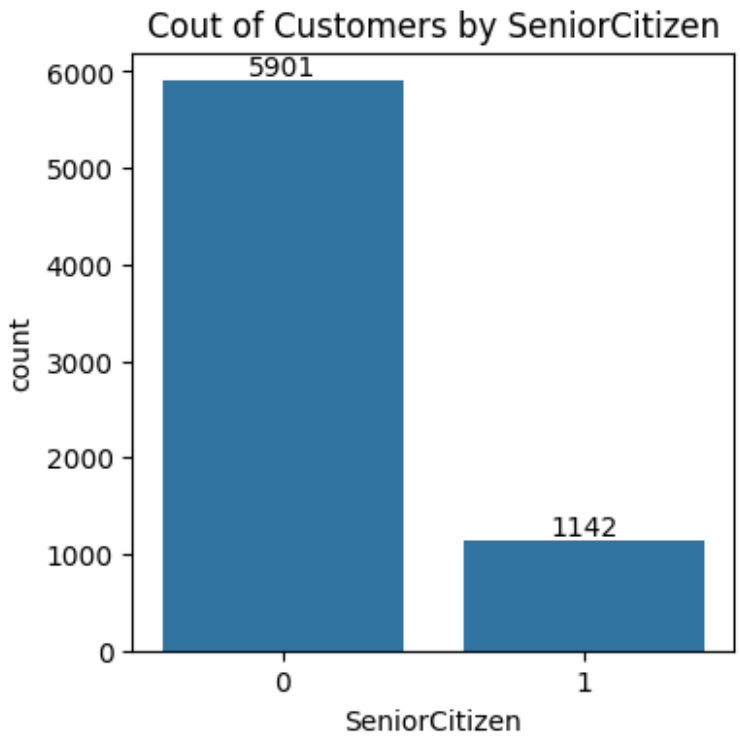
```
# from the given pie chart we can conclude that 26.54% of our
customers have churned out.
# now let's explore the reaso behind it
```

# count of male and Female customers

```
plt.figure(figsize = (3,3))
sns.countplot(x = "gender", data = df, hue = "Churn")
plt.title("Churn by Gender")
plt.show()
```

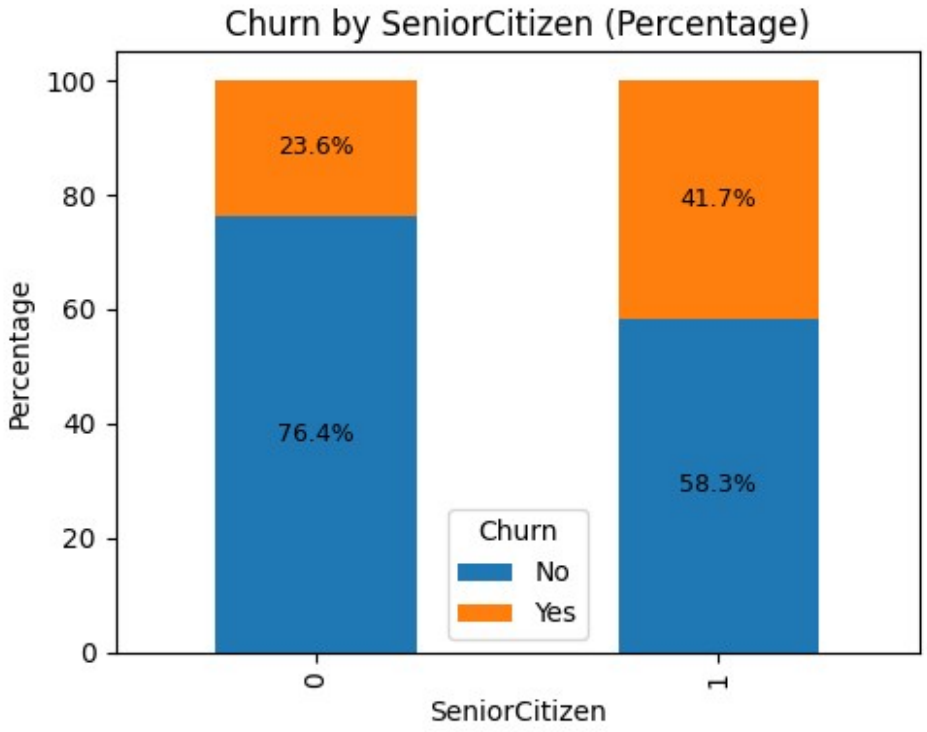



```
plt.figure(figsize=(4, 4))
ax = sns.countplot(x="SeniorCitizen", data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by SeniorCitizen")
plt.show()
```



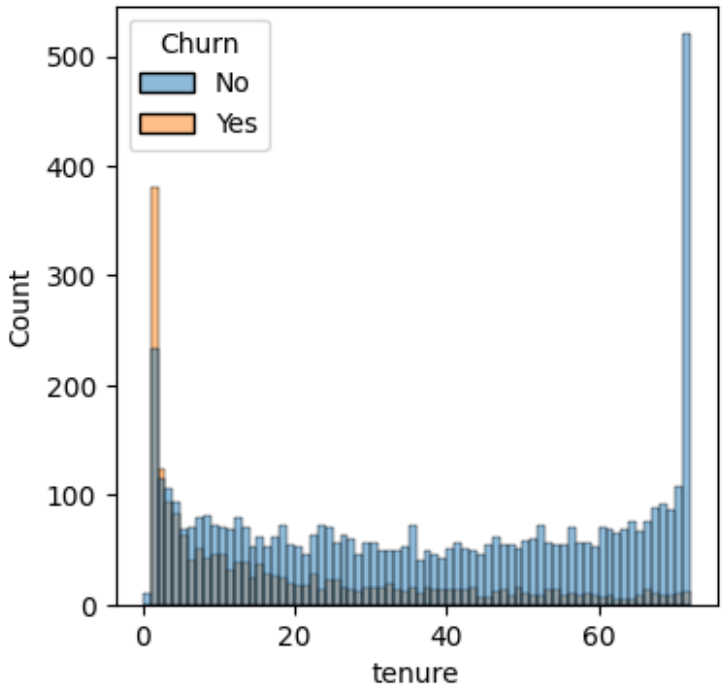

```
# Sample: load your dataset
df = pd.read_csv("Customer Chur.csv")


# Step 1: Group and calculate percentages
grouped = df.groupby(['SeniorCitizen',
'Churn']).size().unstack().fillna(0)

# Step 2: Calculate row-wise percentages
percentages = grouped.div(grouped.sum(axis=1), axis=0) * 100

# Step 3: Plot stacked bar chart
ax = percentages.plot(kind='bar', stacked=True, figsize=(5, 4))

# Step 4: Add percentage labels
for idx, row in percentages.iterrows():
    cum_sum = 0
    for col in percentages.columns:
        height = row[col]
        if height > 0:
            ax.text(idx, cum_sum + height / 2, f'{height:.1f}%',
ha='center', va='center', fontsize=9)
            cum_sum += height

# Labels and title
plt.title("Churn by SeniorCitizen (Percentage)")
plt.xlabel("SeniorCitizen")
```

```
plt.ylabel("Percentage")
plt.legend(title="Churn")
plt.tight_layout()
plt.show()
```



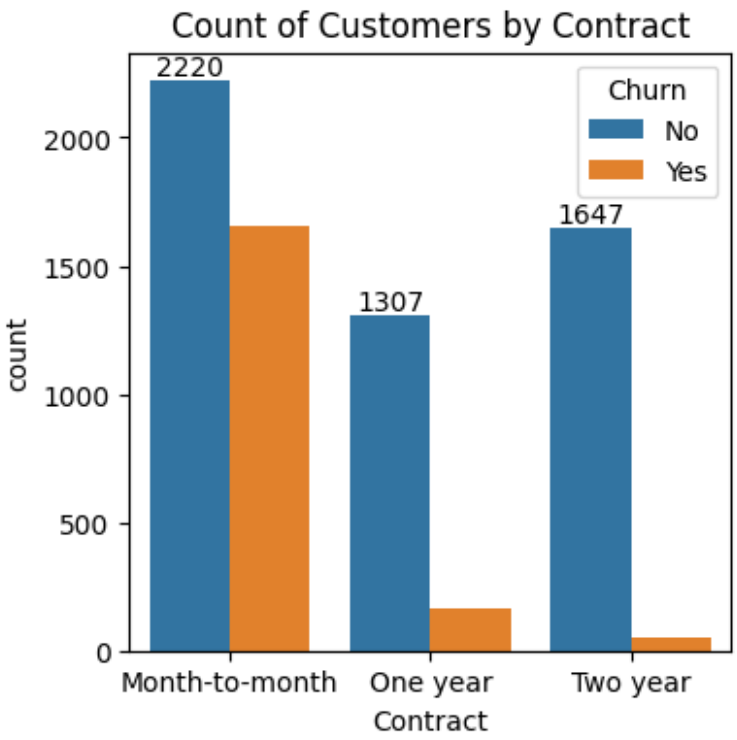

#comparative a gender percentage of people in serior citize category have churned

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




#people who have used our service for a long time have stayed and people who have used our service #1 or 2 month have churned

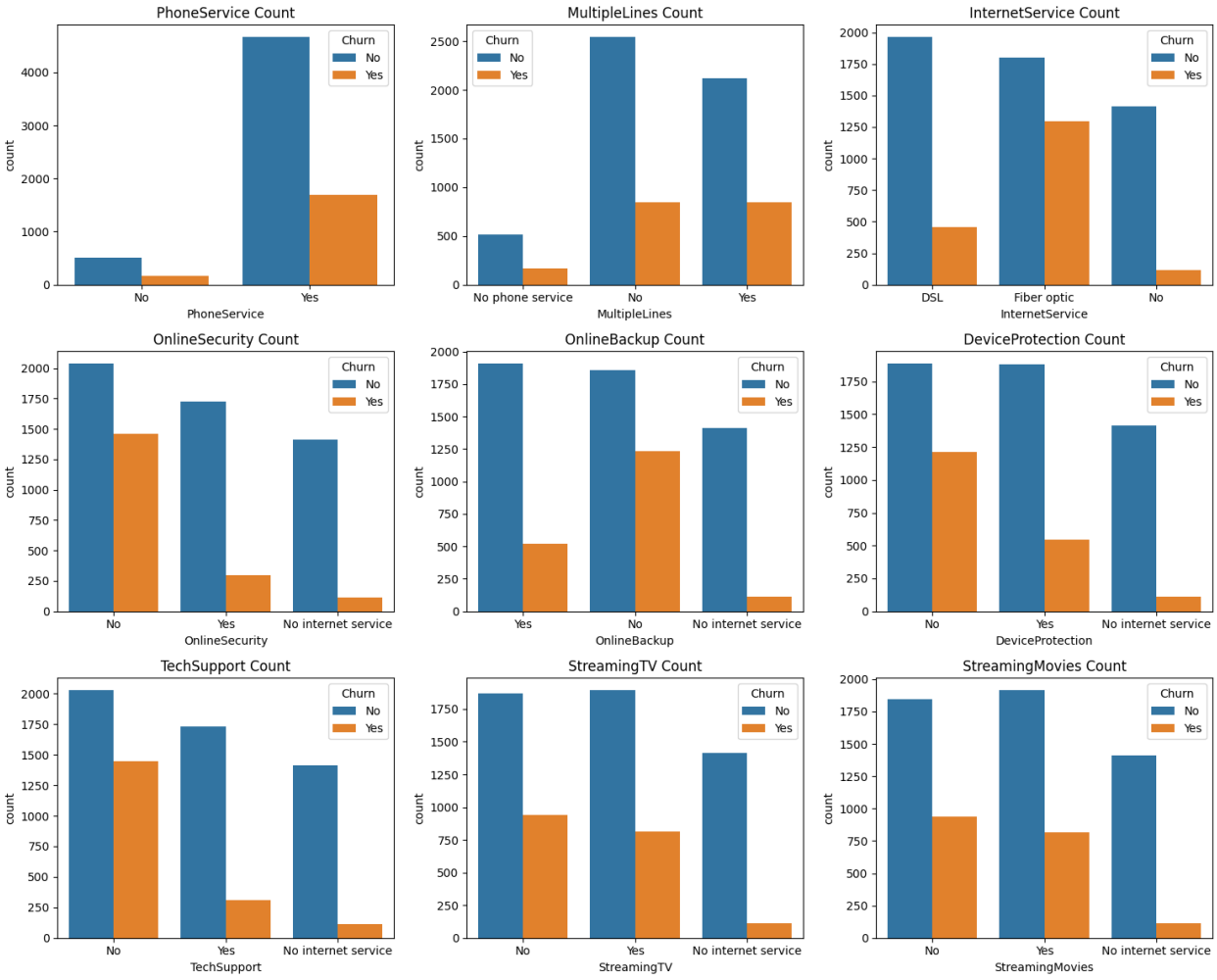
```
plt.figure(figsize=(4, 4))
ax = sns.countplot(x="Contract", data=df, hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract")
plt.show()
```




#People who have month to month contract are likely to churn the from those who have 1 and 2 years or 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)

# List of categorical service columns
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Number of plots
n_cols = 3
n_rows = (len(cols) + n_cols - 1) // n_cols  # Automatically compute
number of rows

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 12))
```

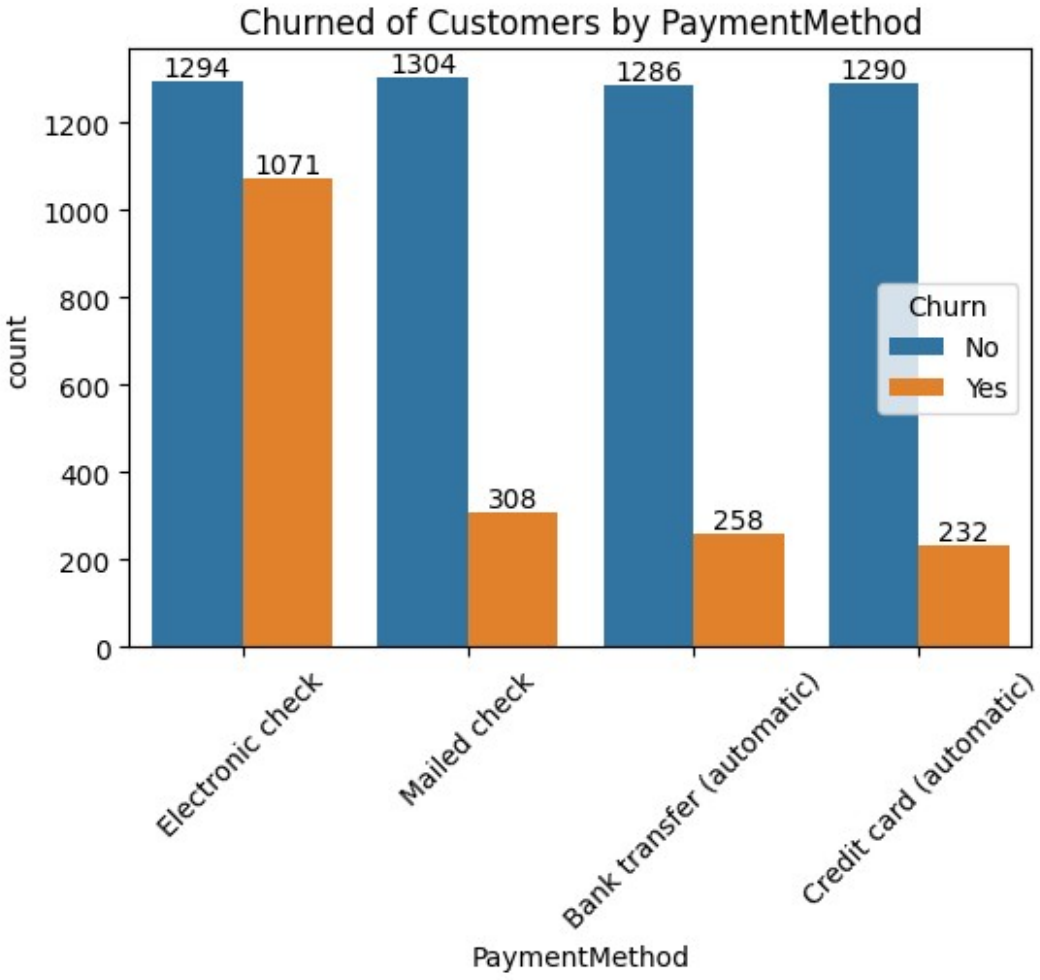
```
axes = axes.flatten()  # Flatten in case of 2D axes array

# Loop over each column and create a countplot
for i, col in enumerate(cols):
    sns.countplot(x=col, data=df, ax=axes[i], hue = df["Churn"])
    axes[i].set_title(f'{col} Count')
    axes[i].tick_params(axis='x', rotation=0)

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

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




#Customers with basic services like PhoneService and TechSupport show lower churn, while churn is higher among users with fiber optic internet. Optional services such as StreamingTV, OnlineSecurity, and DeviceProtection tend to correlate with higher churn when not used. "No

internet service" customers consistently show minimal churn, indicating a more stable user base.

```
plt.figure(figsize=(6, 4))
ax = sns.countplot(x="PaymentMethod", data=df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churned of Customers by PaymentMethod")
plt.xticks(rotation = 45)
plt.show()
```




#customers is likely to churn when is using electronic check as a payment method.