```python
import seaborn as sns
print("Seaborn is working!")
```

Seaborn is working!

```python
import pandas as pd
import numpy as np
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  \
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 0 as tenure is 0 and no total charges are recorded

```python
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
```

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

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

```python
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

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

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

```python
def conv(x):
    # Example conversion logic
```

```
    return "Yes" if x == 1 else "No"

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

#converted 0 and 1 value of senior citizen to yes/no to make it easier to understand

```
df.head(21)
```

	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
5	9305-CDSKC	Female	0	No	No	8	Yes
6	1452-KIOVK	Male	0	No	Yes	22	Yes
7	6713-OKOMC	Female	0	No	No	10	No
8	7892-POOKP	Female	0	Yes	No	28	Yes
9	6388-TABGU	Male	0	No	Yes	62	Yes
10	9763-GRSKD	Male	0	Yes	Yes	13	Yes
11	7469-LKBCI	Male	0	No	No	16	Yes
12	8091-TTVAX	Male	0	Yes	No	58	Yes
13	0280-XJGEX	Male	0	No	No	49	Yes
14	5129-JLPIS	Male	0	No	No	25	Yes
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes
16	8191-XWSZG	Female	0	No	No	52	Yes
17	9959-WOFKT	Male	0	No	Yes	71	Yes
18	4190-MFLUW	Female	0	Yes	Yes	10	Yes
19	4183-MYFRB	Female	0	No	No	21	

```
Yes
20  8779-QRDMV    Male              1       No        No        1
No
```

	MultipleLines	InternetService	OnlineSecurity	...
0	No phone service	DSL	No	...
1	No	DSL	Yes	...
2	No	DSL	Yes	...
3	No phone service	DSL	Yes	...
4	No	Fiber optic	No	...
5	Yes	Fiber optic	No	...
6	Yes	Fiber optic	No	...
7	No phone service	DSL	Yes	...
8	Yes	Fiber optic	No	...
9	No	DSL	Yes	...
10	No	DSL	Yes	...
11	No	No	No internet service	...
12	Yes	Fiber optic	No	...
13	Yes	Fiber optic	No	...
14	No	Fiber optic	Yes	...
15	Yes	Fiber optic	Yes	...
16	No	No	No internet service	...
17	Yes	Fiber optic	Yes	...
18	No	DSL	No	...
19	No	Fiber optic	No	...
20	No phone service	DSL	No	...

	DeviceProtection	TechSupport	StreamingTV
0	No	No	No
1	Yes	No	No
2	No	No	No
3	Yes	Yes	No
4	No	No	No
5	Yes	No	Yes
6	No	No	Yes
7	No	No	No
8	Yes	Yes	Yes
9	No	No	No
10	No	No	No
11	No internet service	No internet service	No internet service
12	Yes	No	Yes
13	Yes	No	Yes
14	Yes	Yes	Yes
15	Yes	Yes	Yes
16	No internet service	No internet service	No internet service
17	Yes	No	Yes
18	Yes	Yes	No
19	Yes	No	No
20	Yes	No	No

```
      StreamingMovies          Contract PaperlessBilling  \
0                  No  Month-to-month              Yes
1                  No        One year               No
2                  No  Month-to-month              Yes
3                  No        One year               No
4                  No  Month-to-month              Yes
5                 Yes  Month-to-month              Yes
6                  No  Month-to-month              Yes
7                  No  Month-to-month               No
8                 Yes  Month-to-month              Yes
9                  No        One year               No
10                 No  Month-to-month              Yes
11  No internet service        Two year               No
12                Yes        One year               No
13                Yes  Month-to-month              Yes
14                Yes  Month-to-month              Yes
15                Yes        Two year               No
16  No internet service        One year               No
17                Yes        Two year               No
18                 No  Month-to-month               No
19                Yes  Month-to-month              Yes
20                Yes  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            Electronic check          99.65        820.50    Yes
6     Credit card (automatic)          89.10       1949.40     No
7               Mailed check          29.75        301.90     No
8            Electronic check         104.80       3046.05    Yes
9   Bank transfer (automatic)          56.15       3487.95     No
10              Mailed check          49.95        587.45     No
11    Credit card (automatic)          18.95        326.80     No
12    Credit card (automatic)         100.35       5681.10     No
13  Bank transfer (automatic)         103.70       5036.30    Yes
14           Electronic check         105.50       2686.05     No
15    Credit card (automatic)         113.25       7895.15     No
16              Mailed check          20.65       1022.95     No
17  Bank transfer (automatic)         106.70       7382.25     No
18    Credit card (automatic)          55.20        528.35    Yes
19           Electronic check          90.05       1862.90     No
20           Electronic check          39.65         39.65    Yes

[21 rows x 21 columns]
```
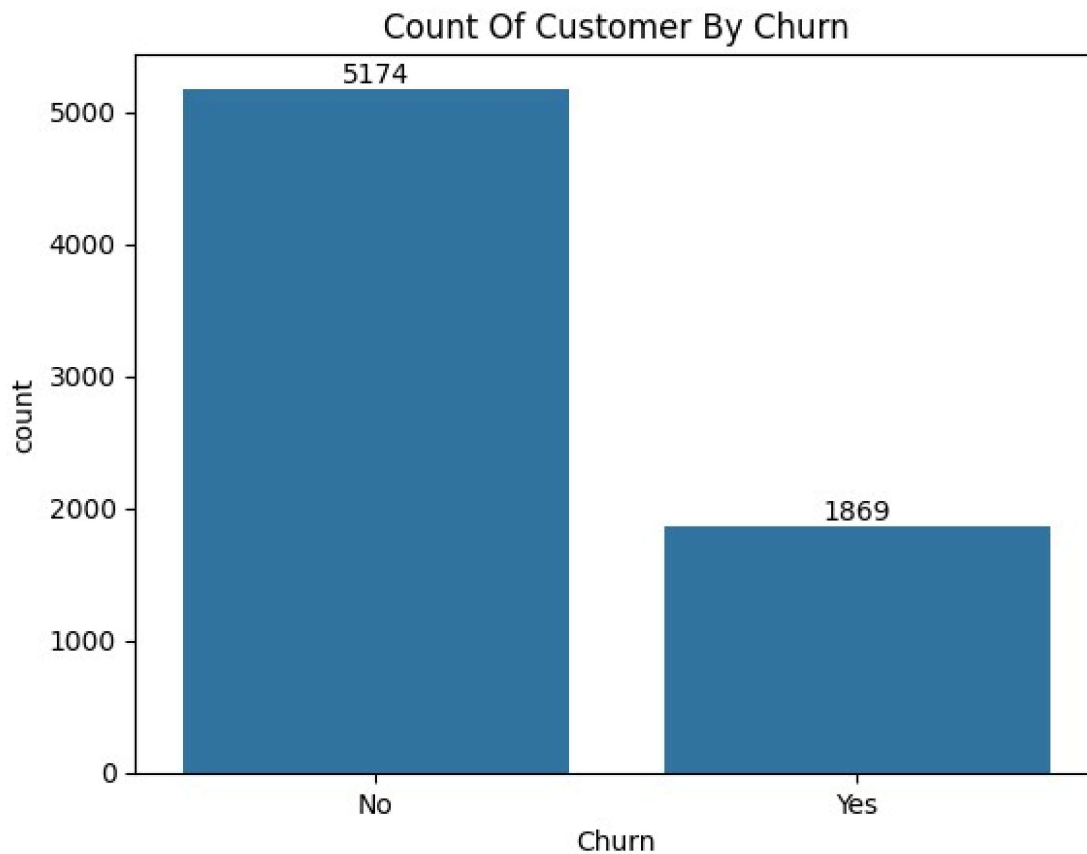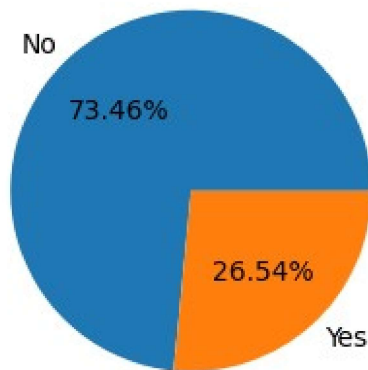
```python
ax = sns.countplot(x = 'Churn',data = df)
```

```
ax.bar_label(ax.containers[0])
plt.title("Count Of Customer By Churn")
plt.show()
```


Count Of Customer By Churn

```
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 Churn Customers",fontsize = 10)
plt.show()
```
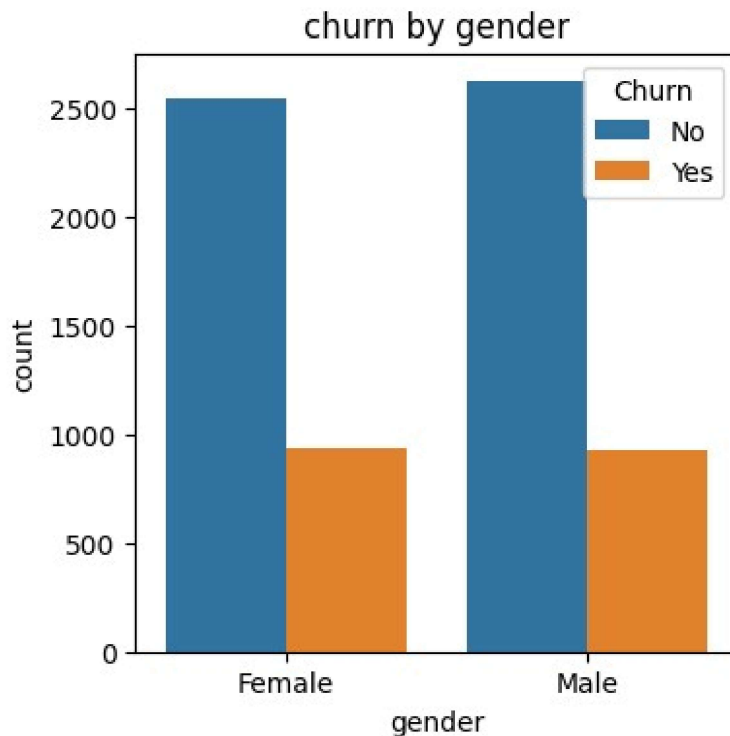
Percentage Of Churn Customers

from the given pie chart we can conclude that 26.54% of our have churned out

now let's explore the reason behind it

```python
plt.figure(figsize = (4,4))
sns.countplot(x = "gender",data = df, hue = "Churn")
plt.title ("churn by gender")
plt.show()
```

churn by gender

```python
counts = df.groupby(["SeniorCitizen", "Churn"]).size().unstack()

# Convert counts to percentages
percentages = counts.div(counts.sum(axis=1), axis=0) * 100

# Plot stacked bar chart
fig, ax = plt.subplots(figsize=(4, 4))  # Adjust size as needed
bottom = None  # Initialize bottom for stacking

# Iterate through each churn category
for churn_status in percentages.columns:
    bars = ax.bar(percentages.index, percentages[churn_status],
bottom=bottom, label=churn_status)
    bottom = percentages[churn_status] if bottom is None else bottom +
percentages[churn_status]

    # Add percentage labels
    for bar in bars:
        height = bar.get_height()
        if height > 0:  # Avoid labeling empty bars
            ax.text(bar.get_x() + bar.get_width() / 2, bar.get_y() +
height / 2,
                    f"{height:.1f}%", ha='center', va='center',
color='white', fontsize=10)

# Labels and title
ax.set_xlabel("Senior Citizen")
```

```
ax.set_ylabel("Percentage")
ax.set_title("Churn by Senior Citizen (Stacked)")
ax.legend(title="Churn")

plt.show()
```


Churn by Senior Citizen (Stacked)
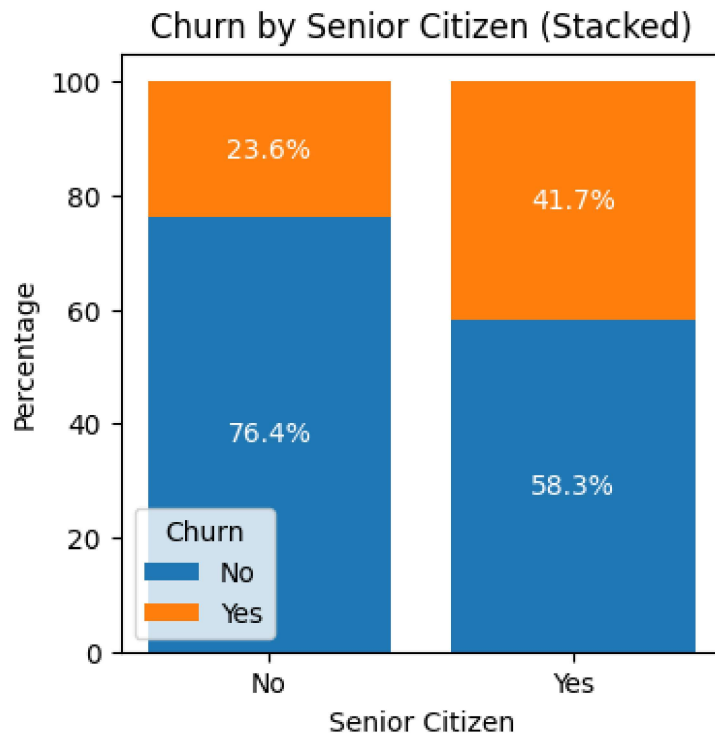
#comparative a greater percentage of people in senior citizen category have churned

```
plt.figure(figsize = (9,4))
sns.histplot(x = "tenure",data = df, bins = 72, hue = 'Churn')
plt.show()
```
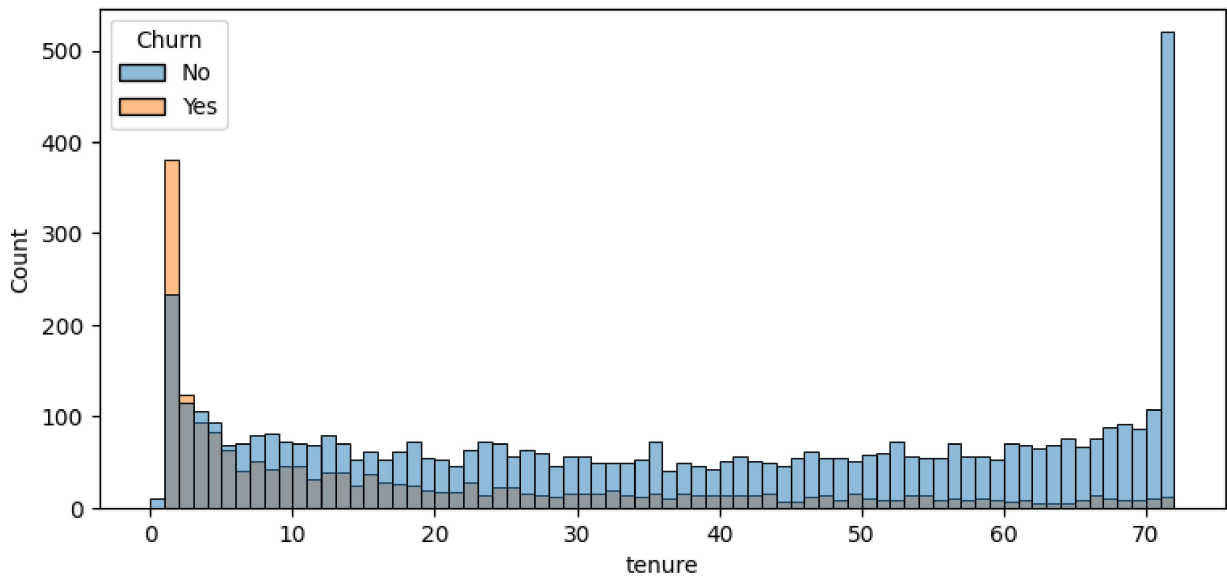
#people who have used our services for a long time have stayed and people who have used our services #1 or 2 months have churned

```python
plt.figure(figsize = (3,4))
ax = sns.countplot(x = "Contract", data = df, hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("count of customer by Contract")
plt.show()
```
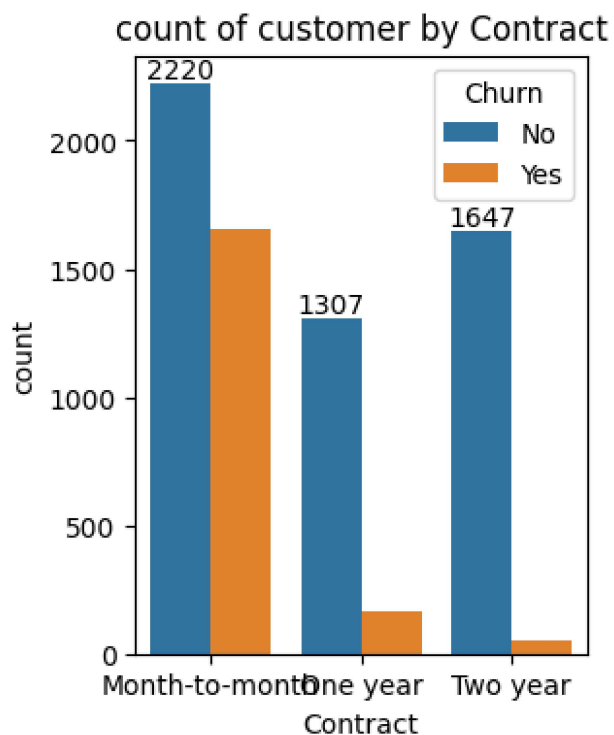
#people who have month to month contract are likely to churn then from those who have 1 or 2 years contract

```
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)


# Sample DataFrame (replace with your actual dataset)
data = {
    'PhoneService': ['Yes', 'No', 'Yes', 'Yes', 'No', 'Yes'],
    'MultipleLines': ['No', 'No', 'Yes', 'No', 'Yes', 'Yes'],
    'InternetService': ['DSL', 'Fiber optic', 'DSL', 'No', 'Fiber
optic', 'DSL'],
    'OnlineSecurity': ['No', 'Yes', 'No', 'No', 'Yes', 'No'],
    'OnlineBackup': ['Yes', 'No', 'Yes', 'No', 'No', 'Yes'],
    'DeviceProtection': ['No', 'Yes', 'No', 'Yes', 'No', 'No'],
    'TechSupport': ['No', 'Yes', 'Yes', 'No', 'Yes', 'No'],
    'StreamingTV': ['Yes', 'No', 'Yes', 'No', 'No', 'Yes'],
    'StreamingMovies': ['No', 'Yes', 'Yes', 'No', 'Yes', 'Yes'],
    'Churn': ['Yes', 'No', 'Yes', 'No', 'No', 'Yes']  # Churn column
}

df = pd.DataFrame(data)

# Define the columns for subplots
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(8, 6))
axes = axes.flatten()  # Flatten for easy iteration

# Loop through columns and create count plots
for i, col in enumerate(columns):
    sns.countplot(x=df[col], ax=axes[i], hue=df["Churn"],
palette='Set2')
    axes[i].set_title(col)
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')

# Adjust layout for better spacing
```
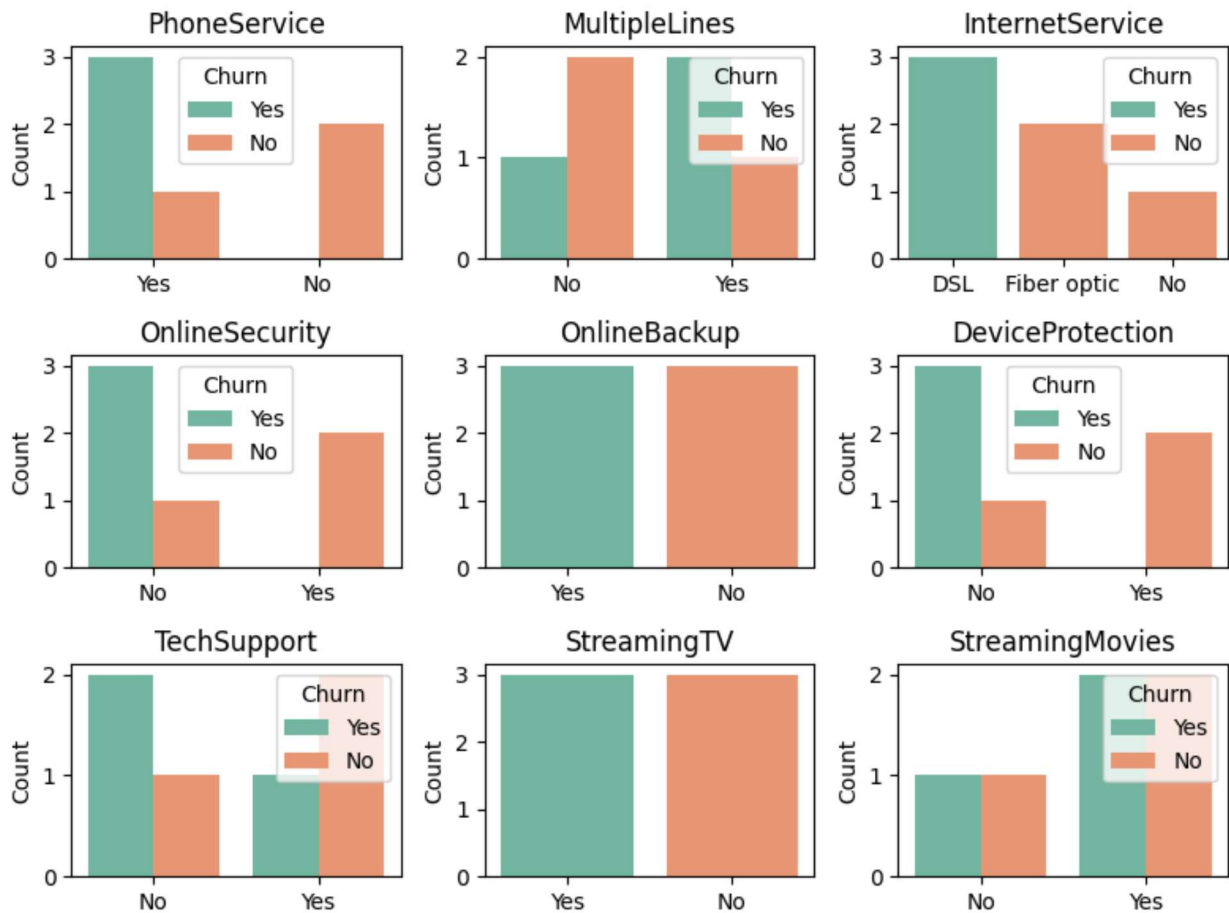
```
plt.tight_layout()
plt.show()
```


#The charts indicate that customers with certain services, such as fiber optic internet and no
security services, have higher churn rates.

```
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 customer PaymentMethod")
plt.xticks(rotation = 45)
plt.show()
```

Churned customer PaymentMethod
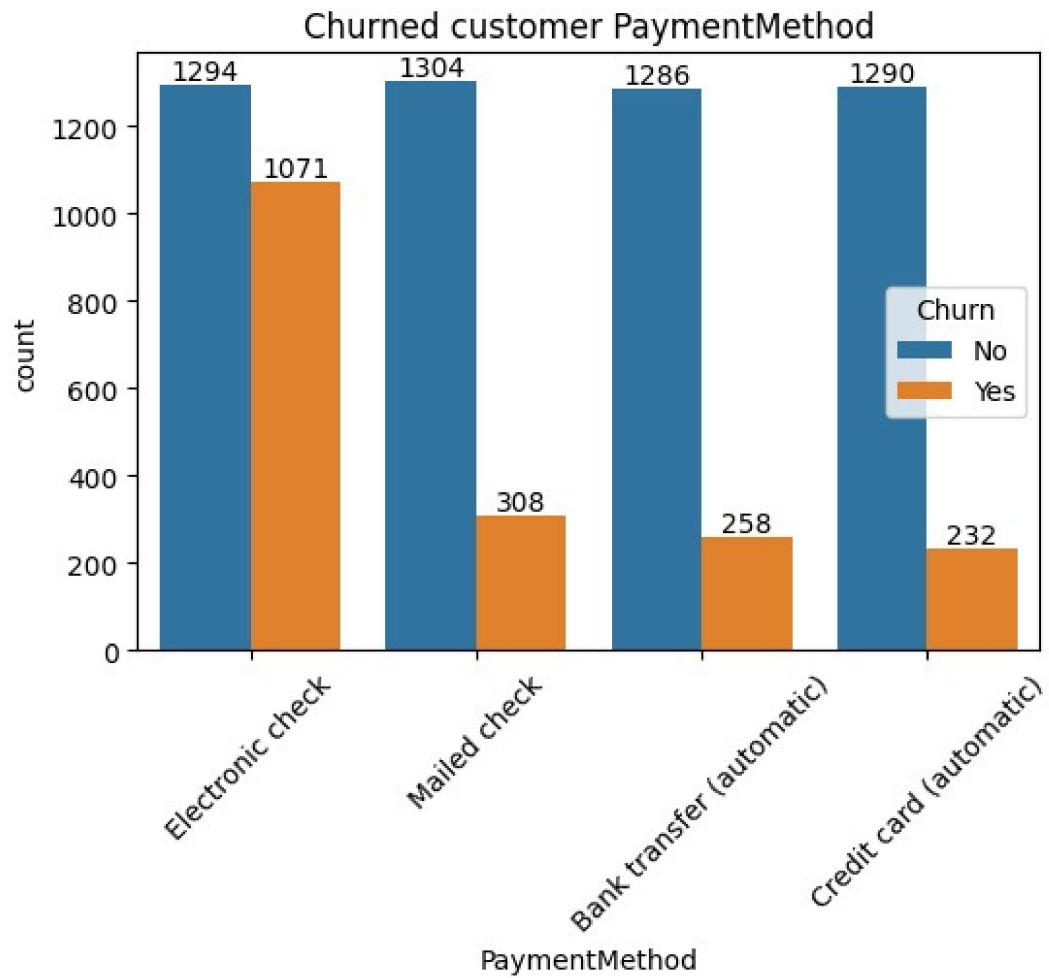
#customer is likely to churn when he isusing electronic check as a payment