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

In [3]:
```
df = pd.read_csv("Customer Churn.csv")
df
```

Out[3]:

	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
...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes
7040	4801-JZAZL	Female	0	Yes	Yes	11	No
7041	8361-LTMKD	Male	1	Yes	No	4	Yes
7042	3186-AJIEK	Male	0	No	No	66	Yes

7043 rows × 21 columns

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   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Replacing blank values in Total Charges as 0 so that we can convert the data type into float

In [8]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

In [10]:
```
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 [12]:
```
df.isnull().sum()
```

Out[12]:
```
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 [14]:
```
df.isnull().sum().sum()
```

Out[14]:
```
0
```

In [16]:
```
df.describe()
```

Out[16]:

	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 [18]:
```
df.duplicated()
```

Out[18]:
```
0       False
1       False
2       False
3       False
4       False
        ...
7038    False
7039    False
7040    False
7041    False
7042    False
Length: 7043, dtype: bool
```

In [24]:
```
df.duplicated().sum()
```

Out[24]:
```
0
```

In [26]:
```
df["customerID"].duplicated().sum()
```

Out[26]:
```
0
```

Converting 0 and 1 value of Senior Citizen column to Yes and No to make it easier to unerstand to everyone

In [28]:
```
def conv(value):
    if value == 1:
        return "Yes"
    else:
        return "No"
df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

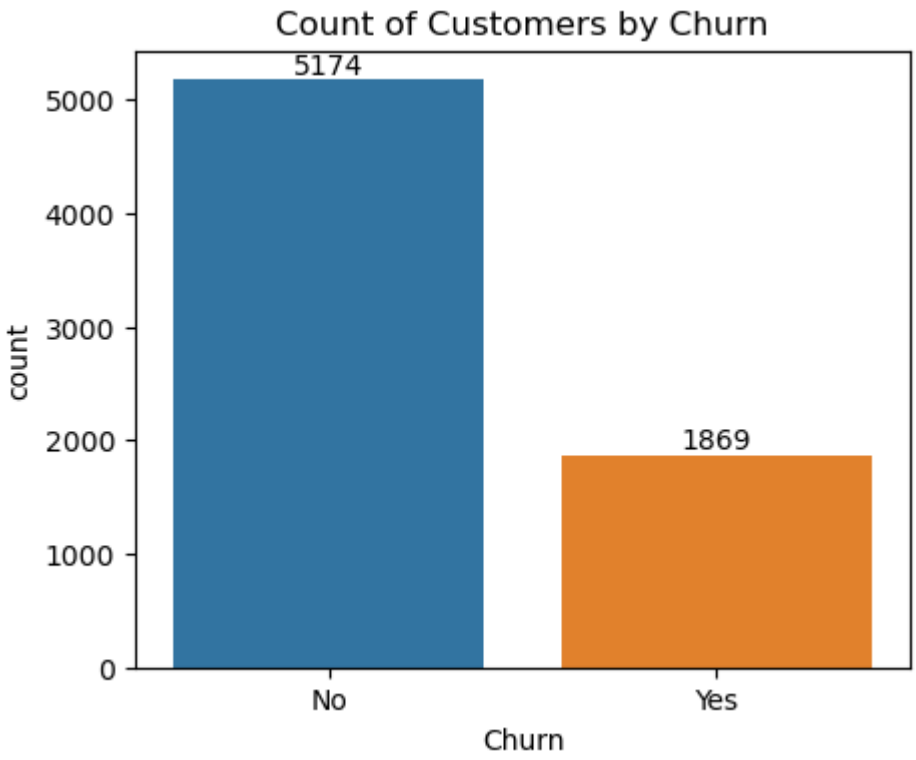
In [30]:
```
df.head()
```

Out[30]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
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	

5 rows × 21 columns

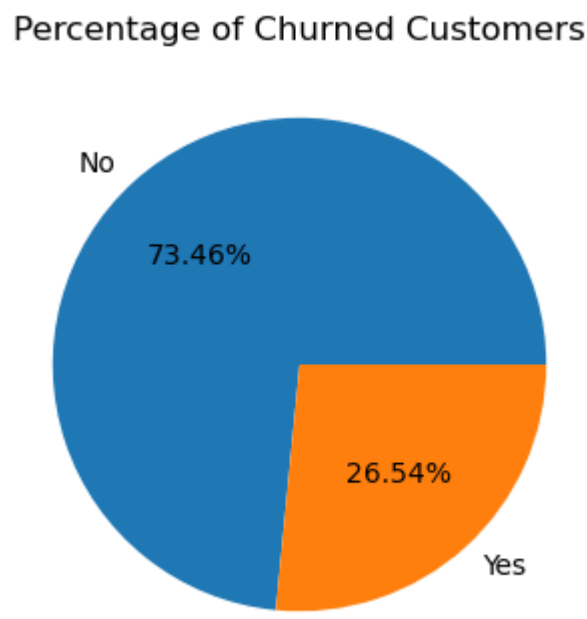
In [67]:
```
plt.figure(figsize=(5,4))
ax = sns.countplot(x = "Churn",data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn")
plt.show()
```



In [ ]:
```
```

In [62]:
```
plt.figure(figsize=(4,4))
gb = df.groupby("Churn").agg({"Churn":"count"})
plt.pie(gb["Churn"],labels=gb.index,autopct="%1.2f%%")
```

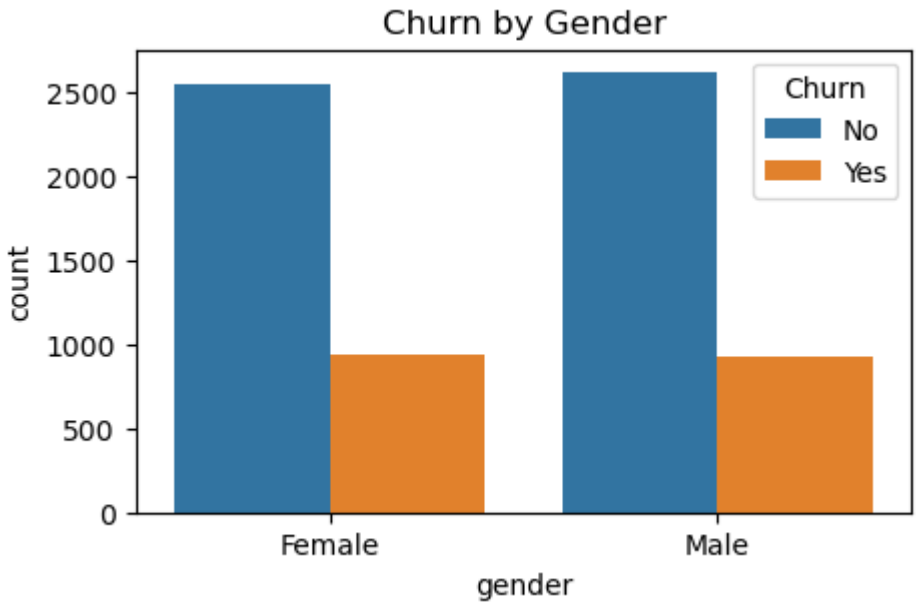
```
plt.title("Percentage of Churned Customers")
plt.show()
```



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

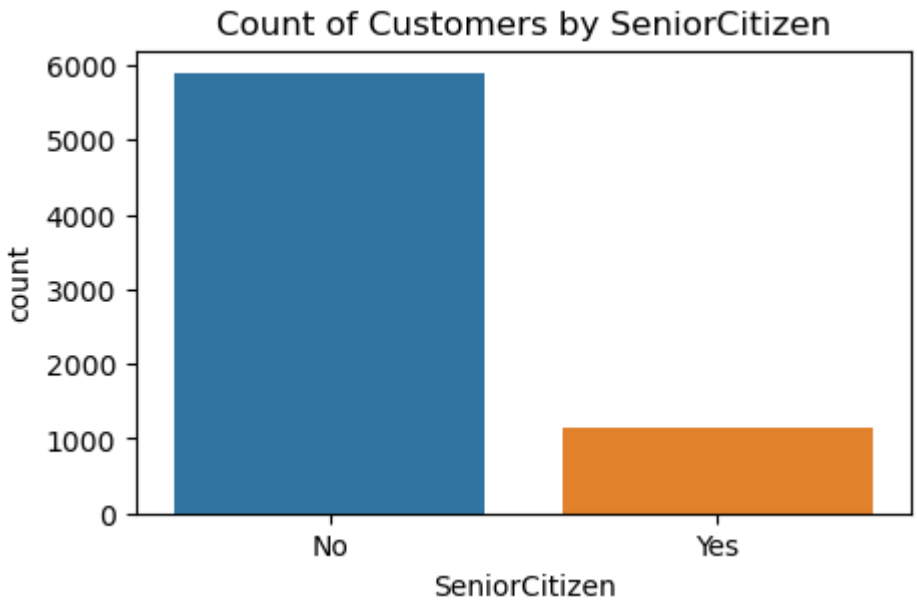
In [ ]:

In [74]:
```
plt.figure(figsize=(5,3))
sns.countplot(x="gender",data = df,hue = "Churn")
plt.title("Churn by Gender")
plt.show()
```



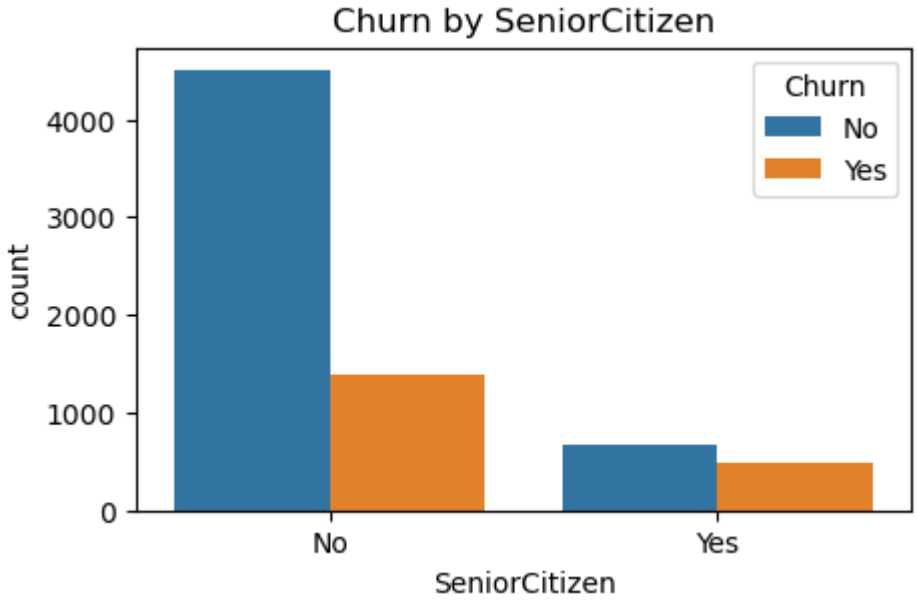
In [ ]:

In [113…
```
plt.figure(figsize=(5,3))
sns.countplot(x="SeniorCitizen",data = df)
plt.title("Count of Customers by SeniorCitizen")
plt.show()
```



In [ ]:

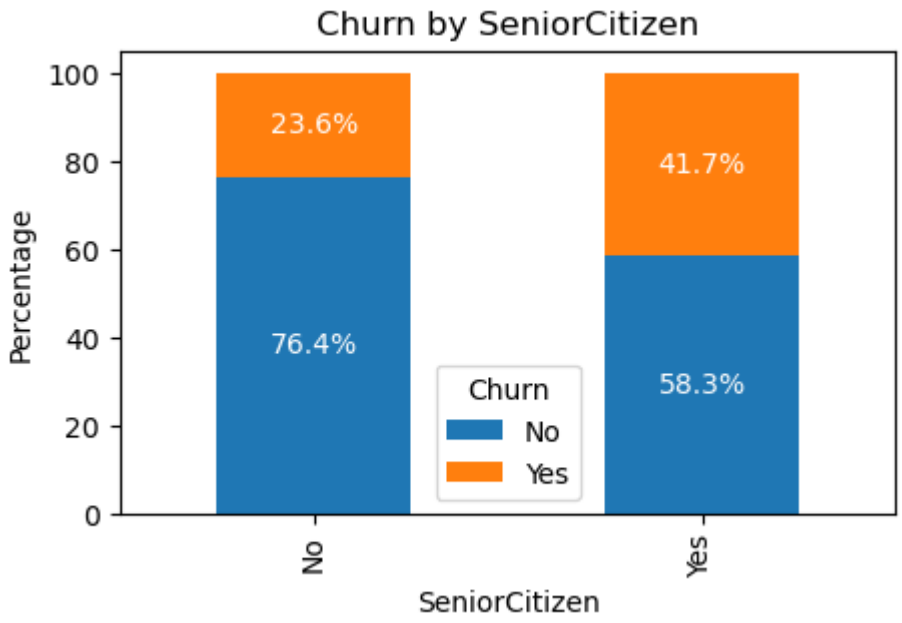
In [76]:
```
plt.figure(figsize=(5,3))
sns.countplot(x="SeniorCitizen",data = df,hue = "Churn")
plt.title("Churn by SeniorCitizen")
plt.show()
```



In [ ]:

In [96]:
```
grouped_data = df.groupby(["SeniorCitizen", "Churn"]).size().unstack(fill_value=
percent_data = grouped_data.div(grouped_data.sum(axis=1), axis=0) * 100

# Create the stacked bar plot
ax = percent_data.plot(kind='bar', stacked=True, figsize=(5, 3), color=["#1f77b4

# Set the title and labels
plt.title("Churn by SeniorCitizen")
plt.ylabel("Percentage")
plt.xlabel("SeniorCitizen")
```

```
# Add percentage labels on top of the bars
for p in ax.patches:
    width = p.get_width()
    height = p.get_height()
    x, y = p.get_xy()
    ax.text(x + width / 2, y + height / 2, f'{height:.1f}%', ha='center', va='ce

plt.show()
```

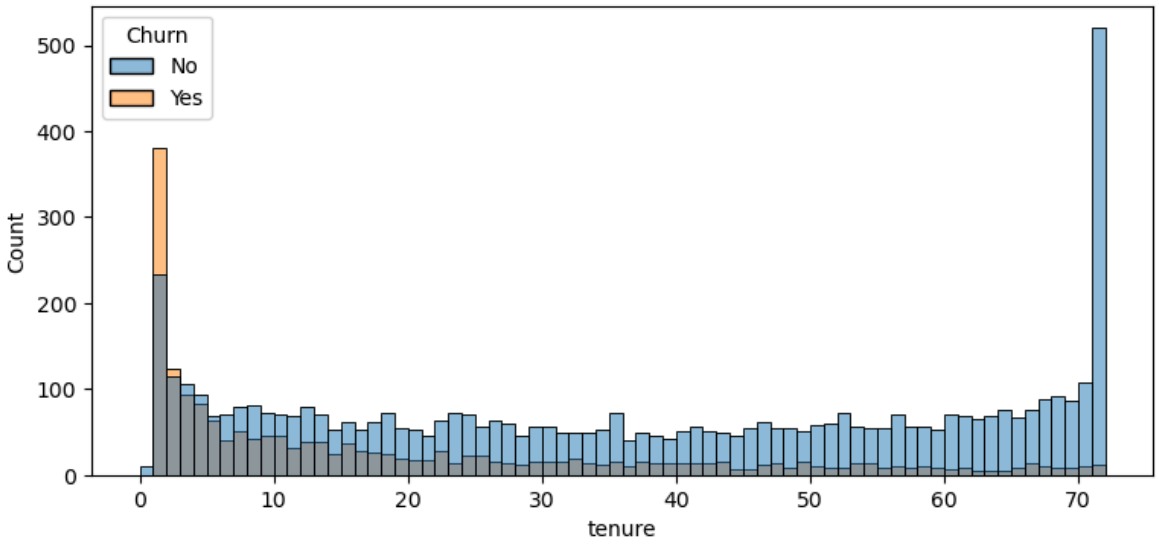


Comparitively a greater percentage of people in Senior Citizen Category have Churned

In [ ]:

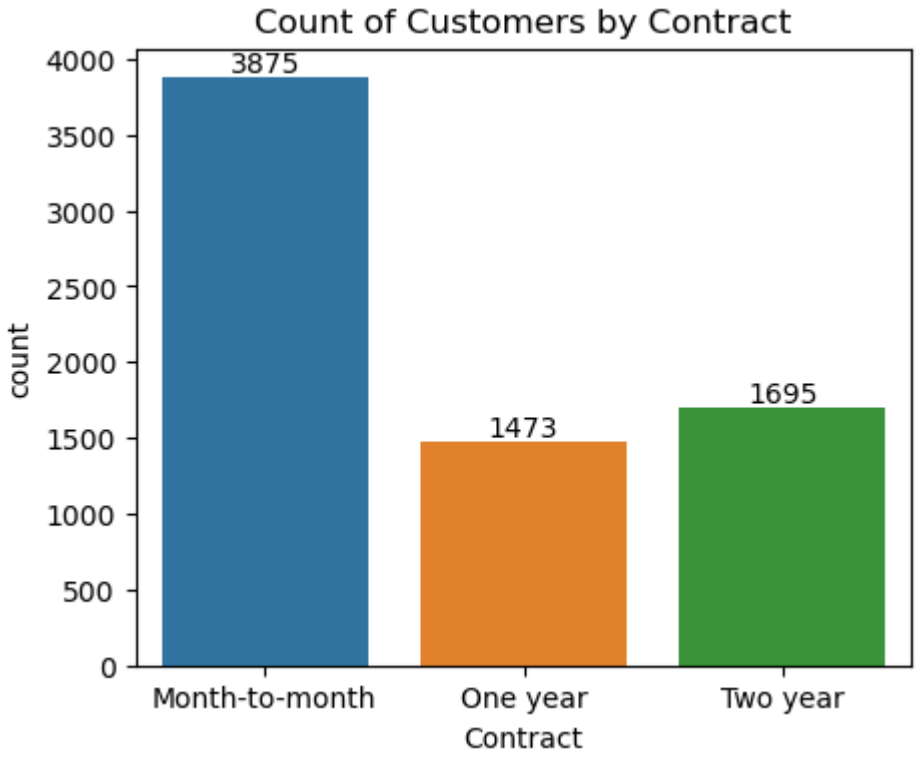
In [108…

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

df.replace([np.inf, -np.inf], np.nan, inplace=True)
```



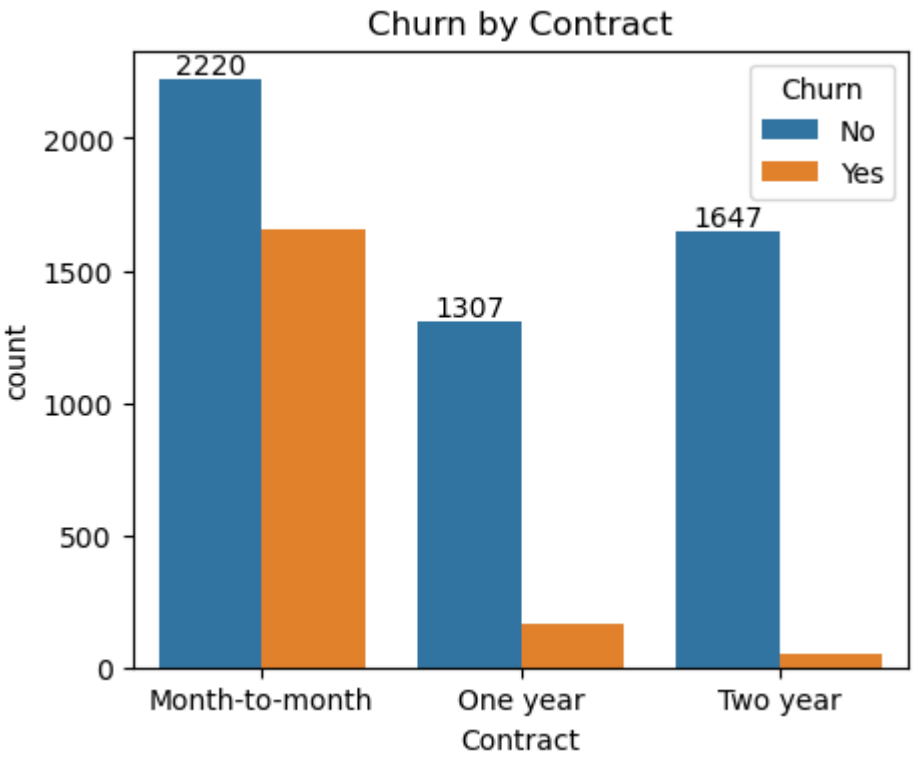
In [ ]:

In [126…
```
plt.figure(figsize=(5,4))
ax = sns.countplot(x="Contract",data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract")
plt.show()
```



In [ ]:
```
```

In [130…
```
plt.figure(figsize=(5,4))
ax = sns.countplot(x="Contract",data = df,hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Churn by Contract")
plt.show()
```

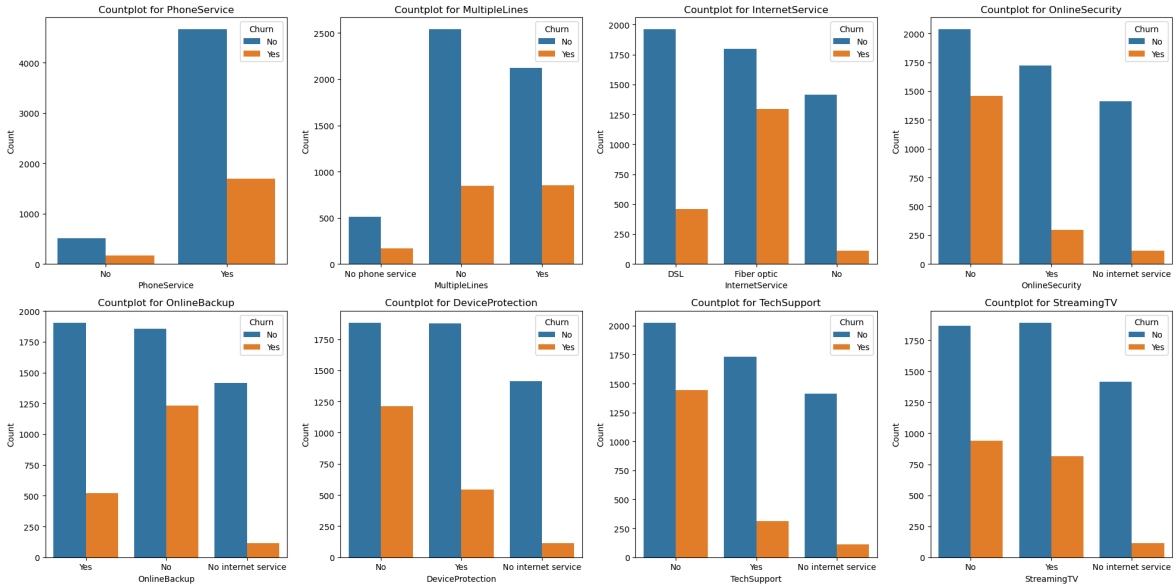


People who have month to month contract are likely to churn, than from those who have 1 or 2 years of contract

In [ ]:

In [140…

```
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV']

fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(20, 10))

axes = axes.flatten()

for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue = df["Churn"])
    axes[i].set_title(f'Countplot for {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Count')

plt.tight_layout()

plt.show()
```

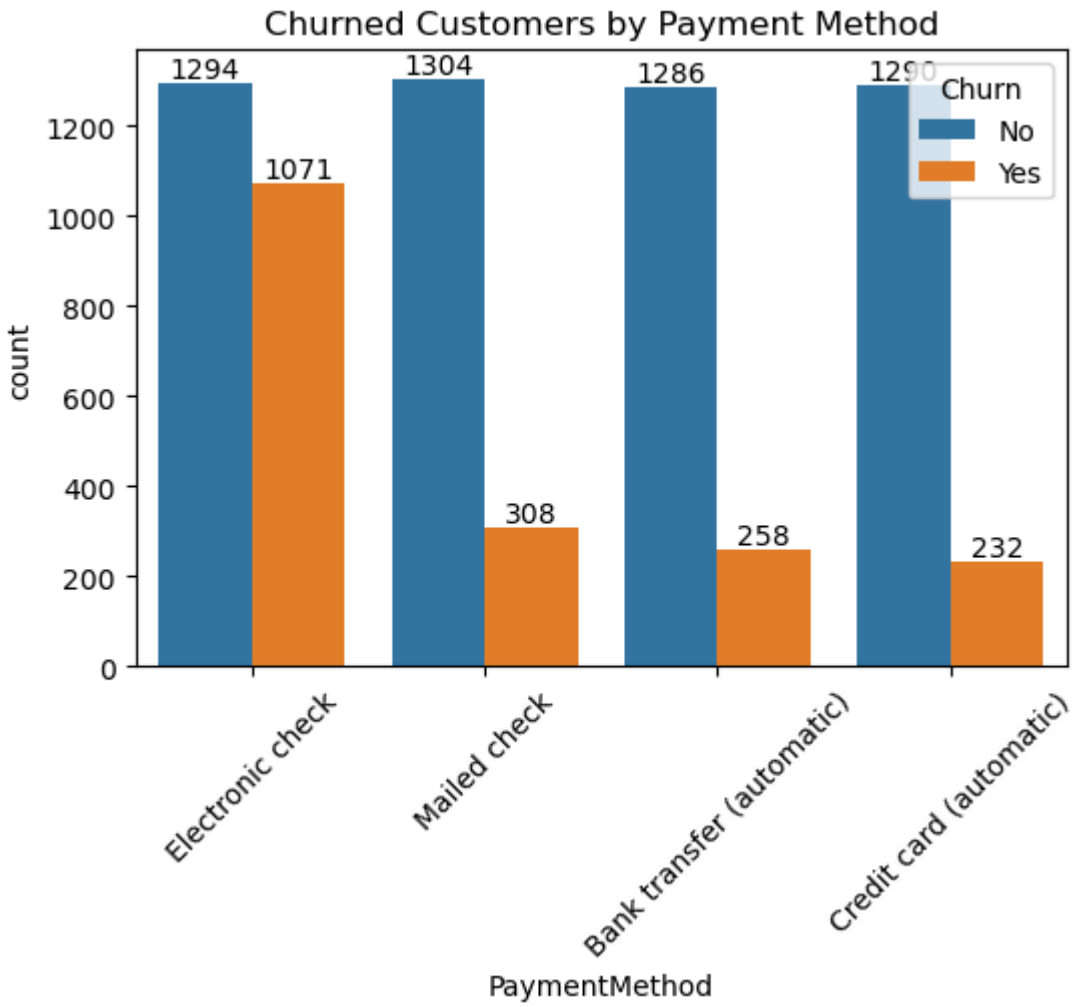


The data shows that customers without additional services like OnlineSecurity, TechSupport, and DeviceProtection tend to churn more. Fiber optic internet users and those without PhoneService or MultipleLines also show higher churn rates. In general, having these services seems to reduce customer churn.

In [ ]:

In [143…

```
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 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 [ ]: