In [1]:

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

In [2]:

```python
df =pd.read_csv("Customer Churn.csv")
df
```

Out[2]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneS
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]:

```python
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.

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 [7]:

```python
df.isnull().sum() # column wise sum of null values.
```

Out[7]:

```
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 [6]:

```
df.isnull().sum().sum() # total null values sum.
```

Out[6]:

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

In [8]:

```
df.describe()
```

Out[8]:

	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 [9]:

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

Out[9]:

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

In [10]:

```
df["customerID"].duplicated().sum() # Always check duplicate values with uniqu
```

Out[10]:

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

Replacing values(1 ,0) of senior citizen column with yes and no

In [11]:

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

In [12]:

```
df.head()
```

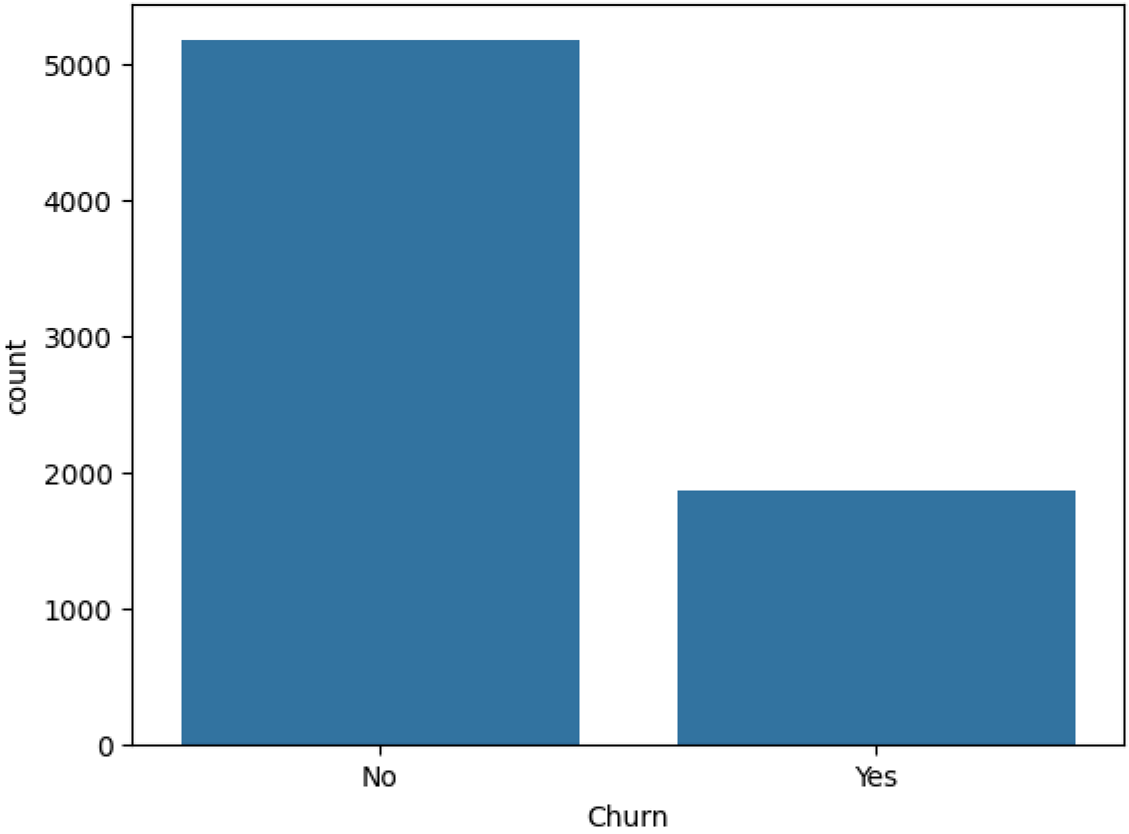
Out[12]:

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

5 rows × 21 columns

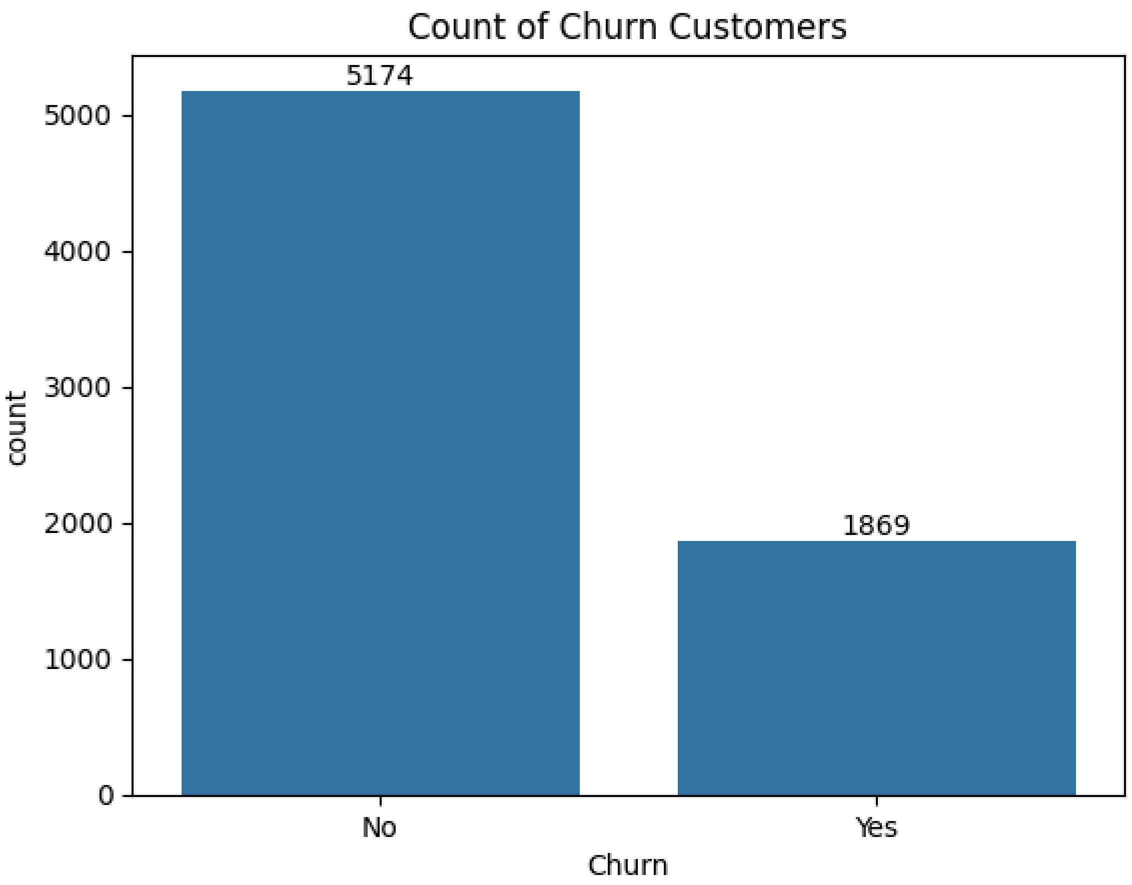
In [13]:

```
sns.countplot( x = "Churn" , data =df)
plt.show()
```



In [14]:

```
ax = sns.countplot( x = "Churn" , data =df)
ax.bar_label(ax.containers[0])
plt.title("Count of Churn Customers" , fontsize= 12)
plt.show()
```



In [15]:

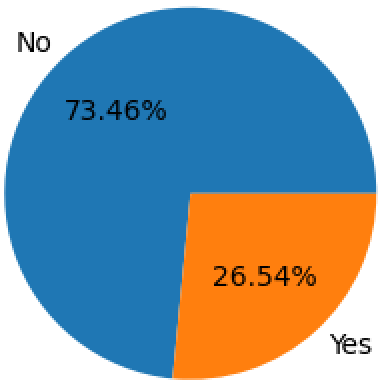
```
gb = df.groupby("Churn").agg({"Churn" : "count"})
gb
```

Out[15]:

	Churn
Churn	
No	5174
Yes	1869

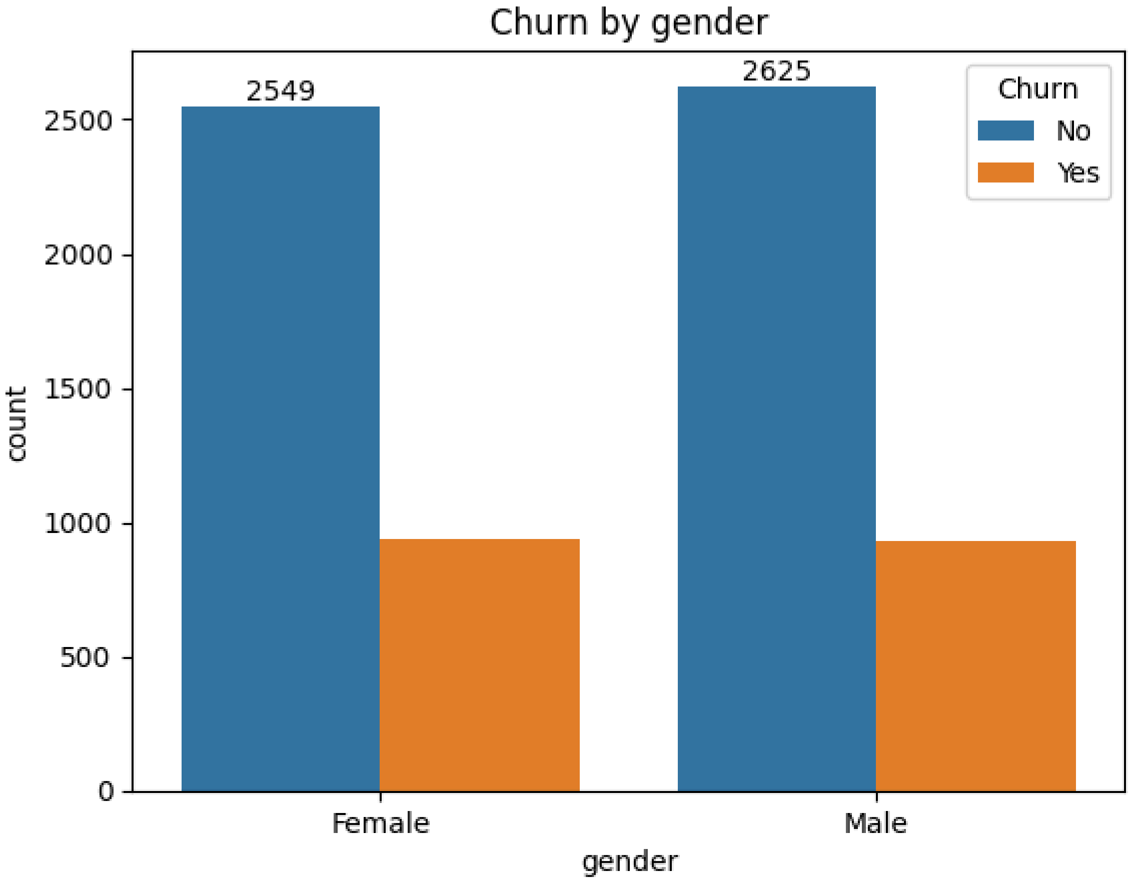
In [16]:

```
plt.figure(figsize=(3,4))
gb = df.groupby("Churn").agg({"Churn" : "count"})
plt.pie(gb["Churn"], labels = gb.index , autopct = "%1.2f%%")
plt.show()
```



Let's explore the reason behind the churn rate (26.54%)

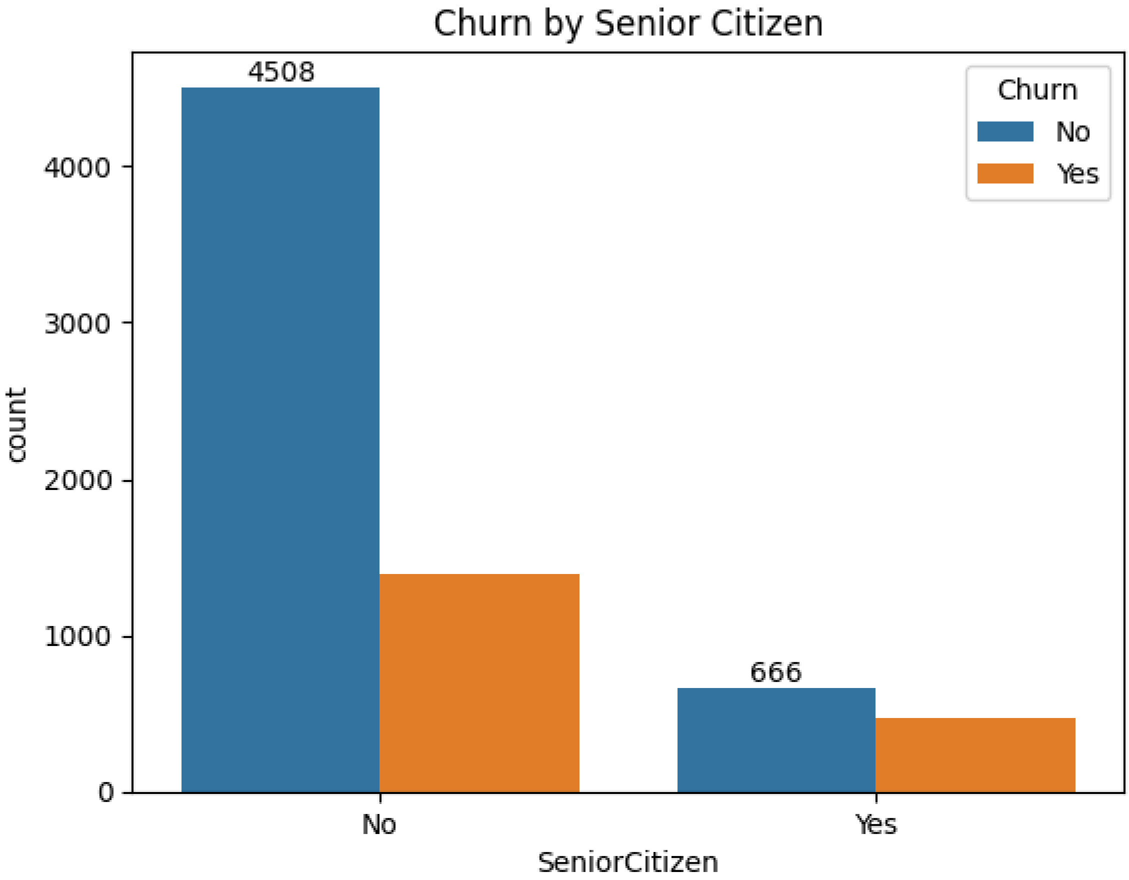
In [17]:
```
ax = sns.countplot(x= "gender" , data =df , hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Churn by gender")
plt.show()
# Churn values are not gender specific , almost same for both genders.
```



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

In [18]:

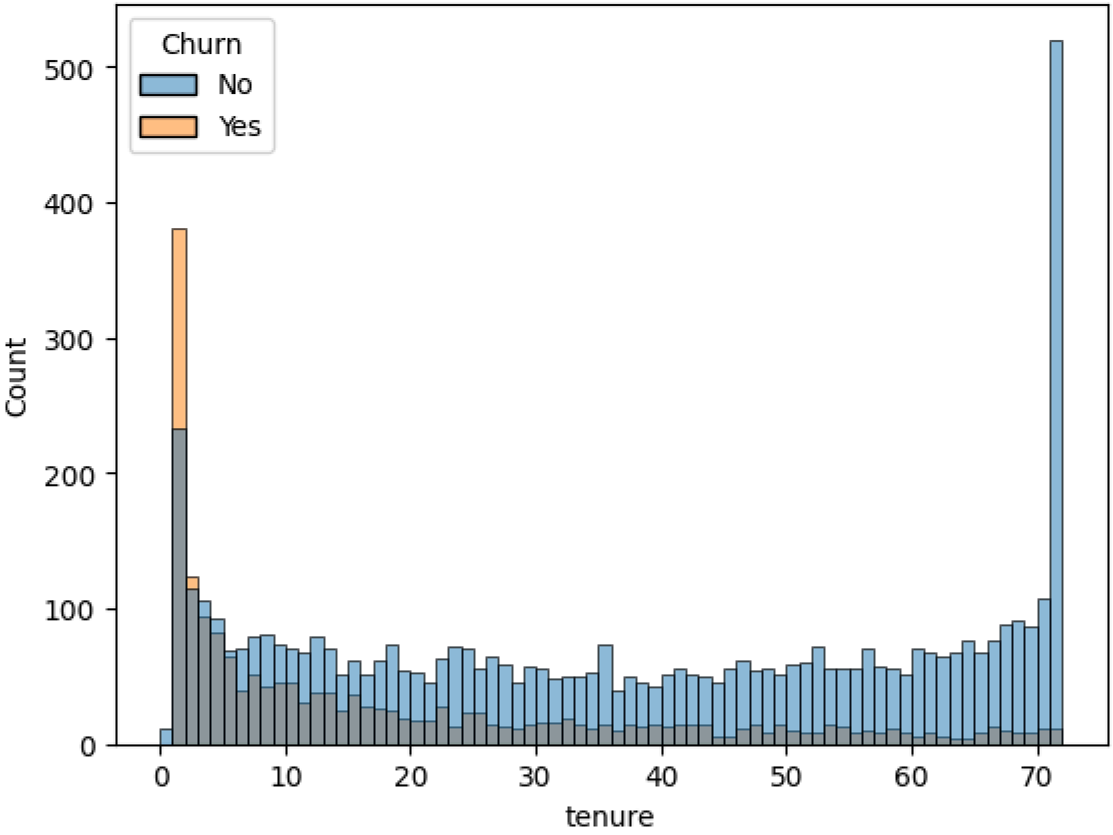
```
ax = sns.countplot(x= "SeniorCitizen" , data =df , hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Churn by Senior Citizen")
plt.show()
```



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

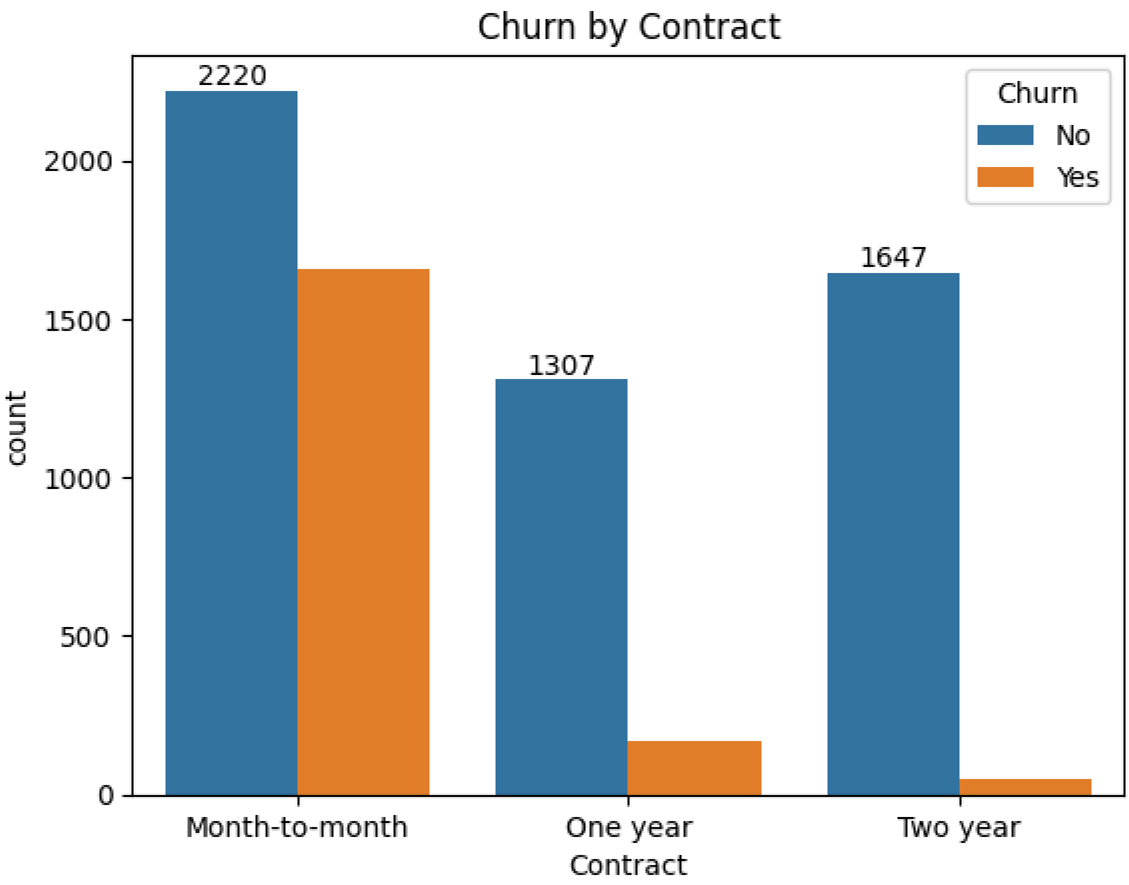
In [20]:

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



As contract duration increases churn rate decreases , short term tenure leads to more churn.

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



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)
```

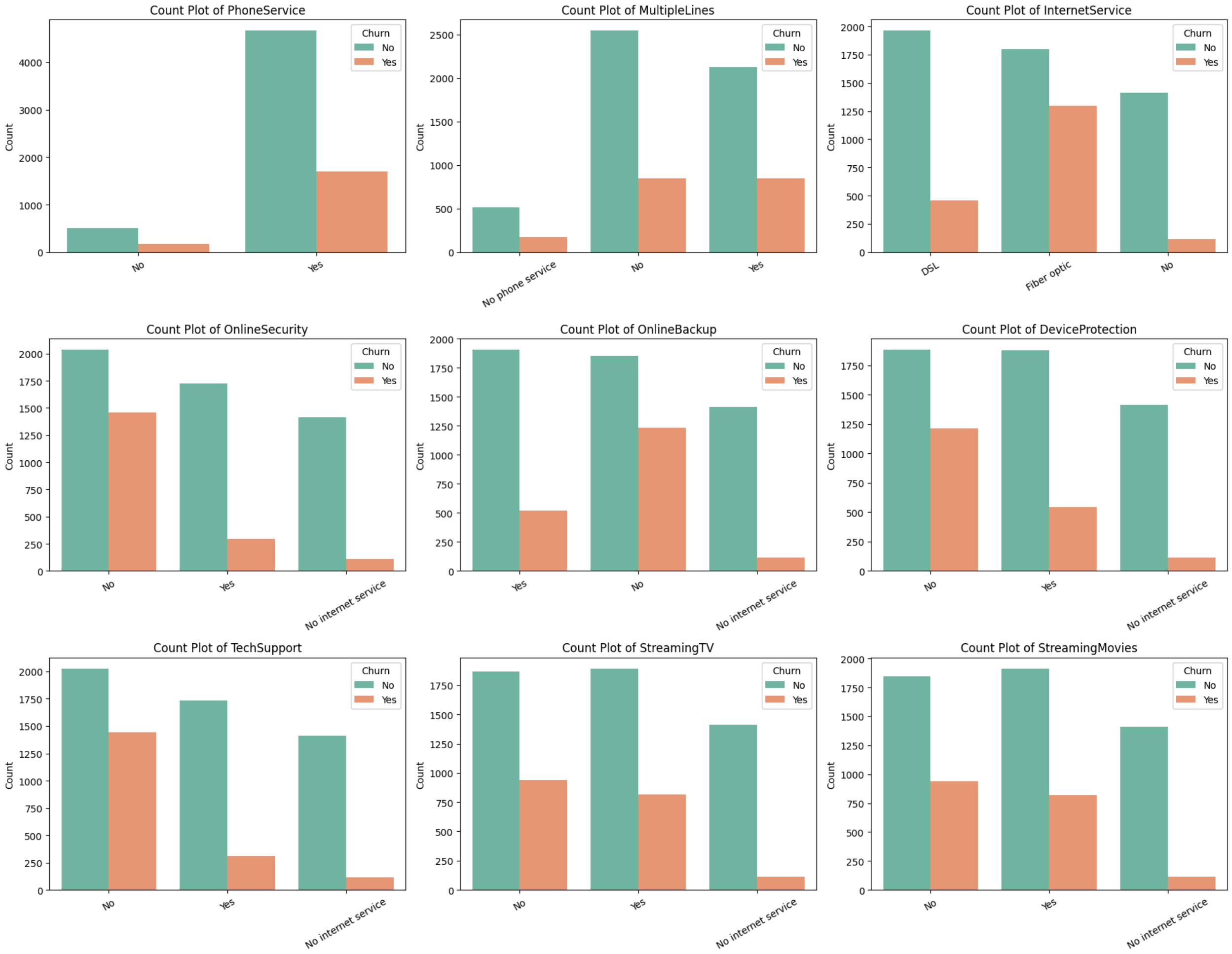
The majority of customers who do not churn tend to have services like PhoneService, InternetService(particularly DSL), OnlineSecurity enabled. For services like OnlineBackup , TechSupport, and StreamingTV , churn rates are noticeably higher when these sevices are not used or are unavailable.

In [21]:
```
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]


# Set figure size and grid
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(18, 14))
axes = axes.flatten()
```

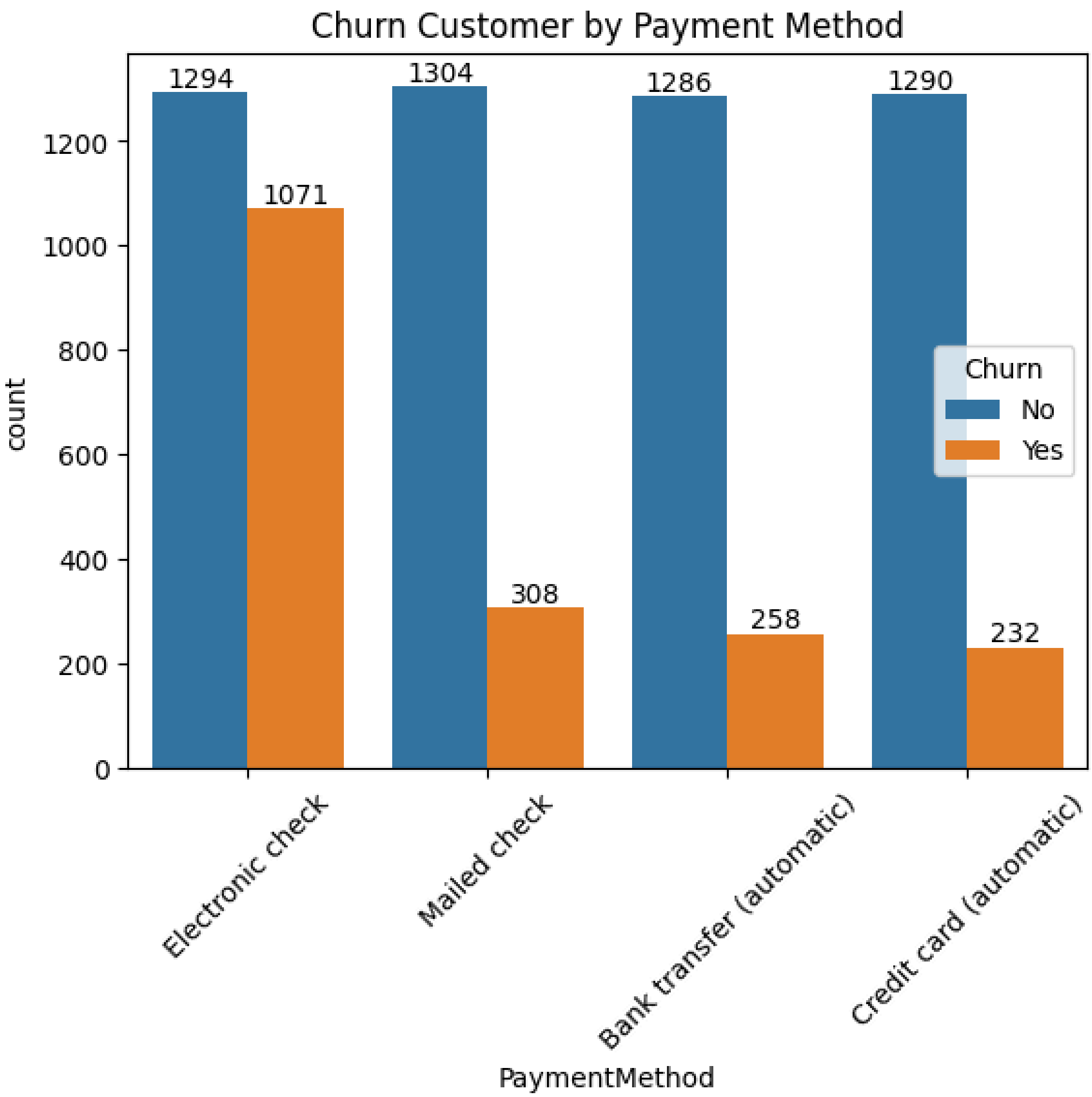
```
for i, col in enumerate(cols):
    sns.countplot(
        data=df,
        x=col,
        ax=axes[i], hue ="Churn" ,
        palette="Set2"
    )
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')
    axes[i].tick_params(axis='x', rotation=30)

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



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

In [24]:
```
ax = sns.countplot(x= "PaymentMethod" , data =df , hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churn Customer by Payment Method")
plt.xticks(rotation = 45)
plt.show()
```



In [ ]:

In [ ]: