

```
In [10]: data=pd.read_csv("Customer Churn.csv")
```

```
In [13]: df=pd.read_csv('customer churn.csv')  
df
```

Out[13]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No
...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes

7043 rows × 21 columns

```
In [15]: 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
```

```
In [22]: #replacing blank with 0 as tenure is 0 and no tattle charges are recorded.
```

```
In [20]: df["TotalCharges"] =df["TotalCharges"].replace(" ", "0")  
df["TotalCharges"] =df["TotalCharges"].astype("float")
```

```
In [21]: 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 [24]: `df.isnull().sum()`

```

Out[24]: 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 [25]: `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

In [29]: `df["customerID"].duplicated().sum()`

Out[29]: `np.int64(0)`

```

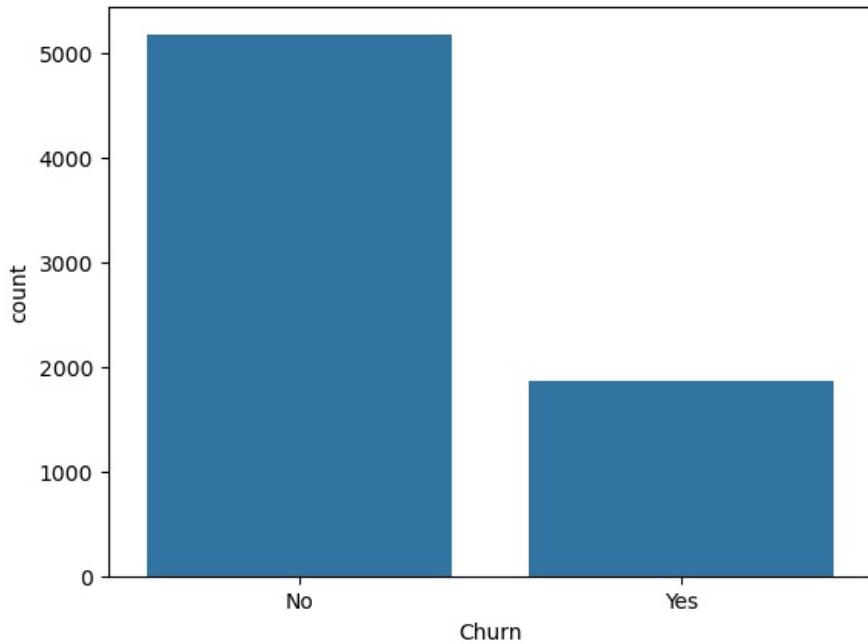
In [32]: def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

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

```

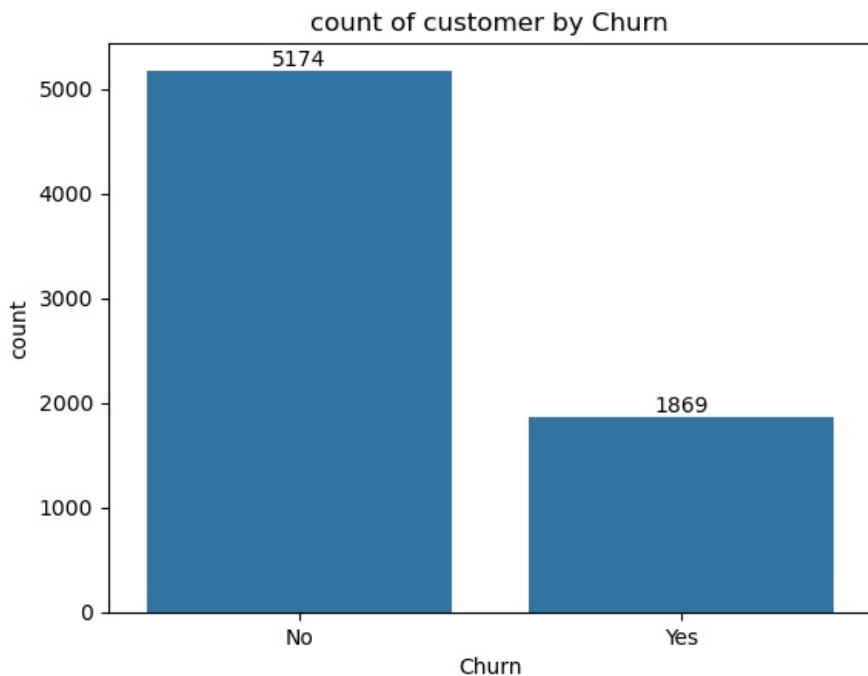
```
In [35]: # converted 0 and 1 value into yes or no for easy to understand.
```

```
In [44]: sns.countplot(x='Churn',data=df)
plt.show()
```



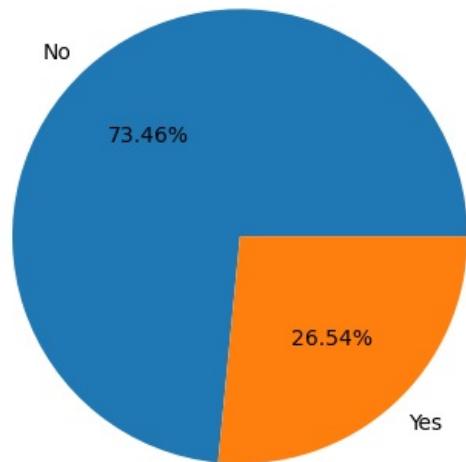
```
In [57]: ax = sns.countplot(x='Churn',data=df)
```

```
ax.bar_label(ax.containers[0])
plt.title("count of customer by Churn")
plt.show()
```



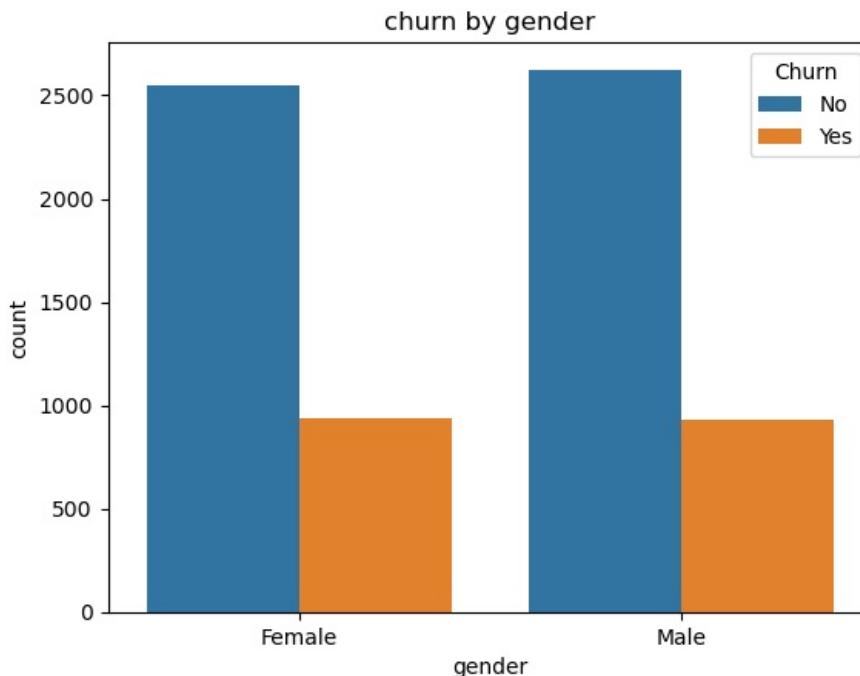
```
In [59]: gb = df.groupby("Churn").agg({'Churn':'count'})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%")
plt.title("Percentage of churn customers")
plt.show()
```

Percentage of churn customers

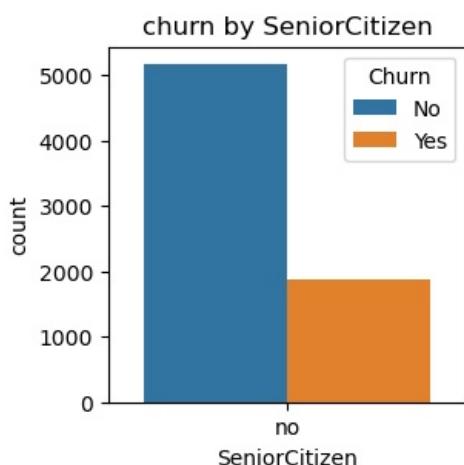


```
In [61]: # from the given pie chart we can conclude that 26.54% of our customer have churned out now explore reason behind it
```

```
In [64]: sns.countplot(x = "gender", data = df, hue= "Churn")
plt.title("churn by gender")
plt.show()
```



```
In [67]: plt.figure(figsize= (3,3))
sns.countplot(x = "SeniorCitizen", data = df, hue= "Churn")
plt.title("churn by SeniorCitizen")
plt.show()
```



```
In [66]: count_df = df.groupby(["SeniorCitizen", "Churn"]).size().unstack(fill_value=0)

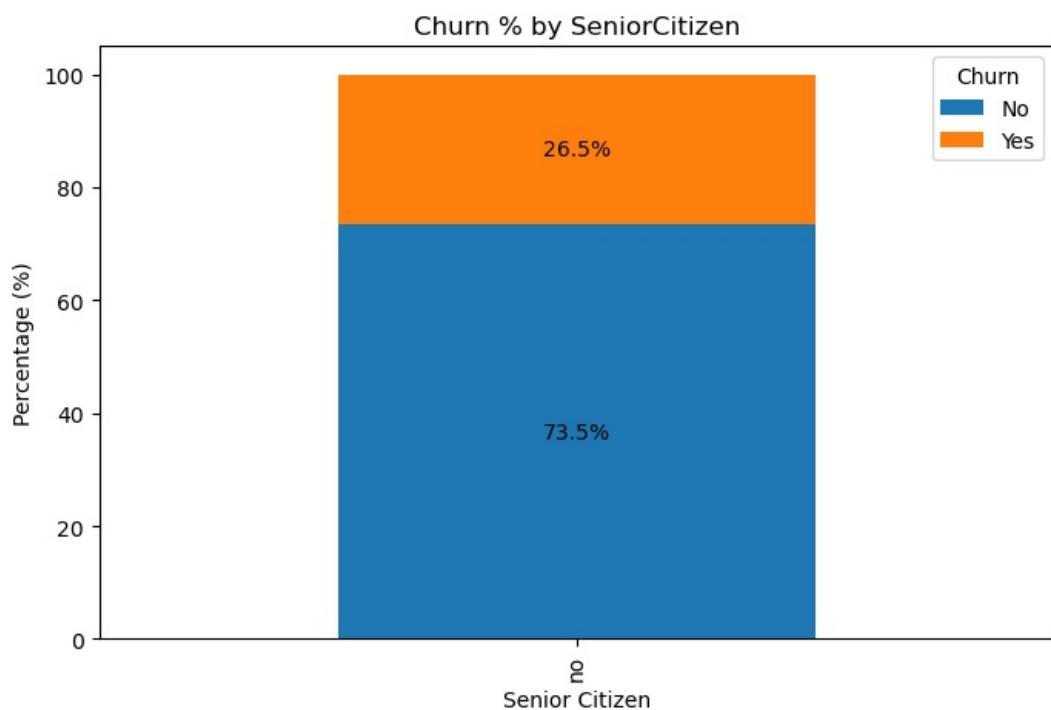
# 2. Convert counts → % of total
percent_df = count_df.div(count_df.sum(axis=1), axis=0) * 100

# 3. Plot stacked bar chart
ax = percent_df.plot(
    kind="bar",
    stacked=True,
    figsize=(8,5)
)

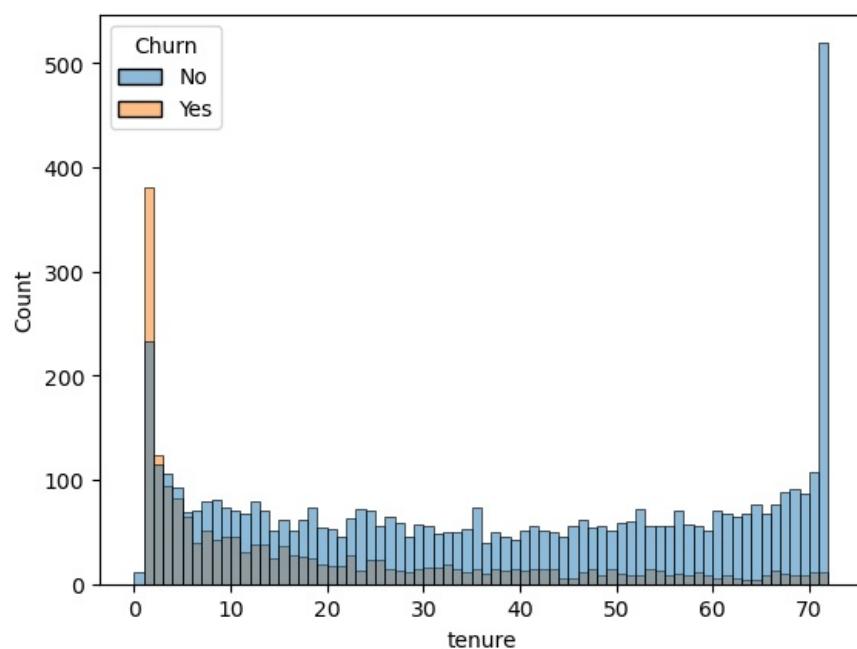
plt.title("Churn % by SeniorCitizen")
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage (%)")
plt.legend(title="Churn")

# 4. Add % labels on bars
for c in ax.containers:
    ax.bar_label(c, fmt=".1f%%", label_type="center")

plt.show()
```



```
In [70]: sns.histplot(x="tenure", data=df, bins=72, hue="Churn")
plt.show()
```

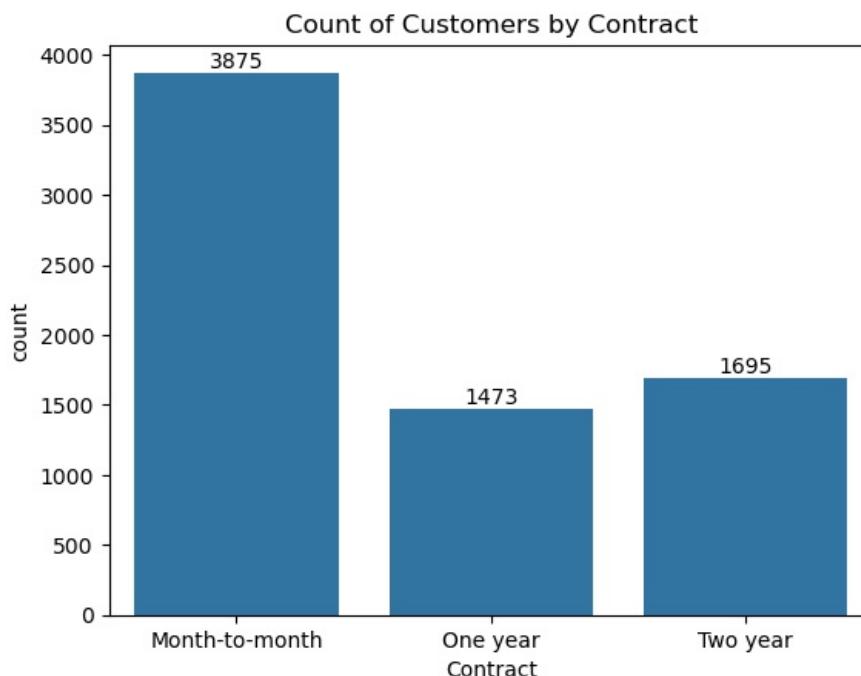


```
In [71]: #People who have used our services for a long time have stayed and people who have used our services #1 or 2 mon
```

```
In [80]: ax = sns.countplot(x="Contract", data=df)

# Add labels to bars
for container in ax.containers:
    ax.bar_label(container)

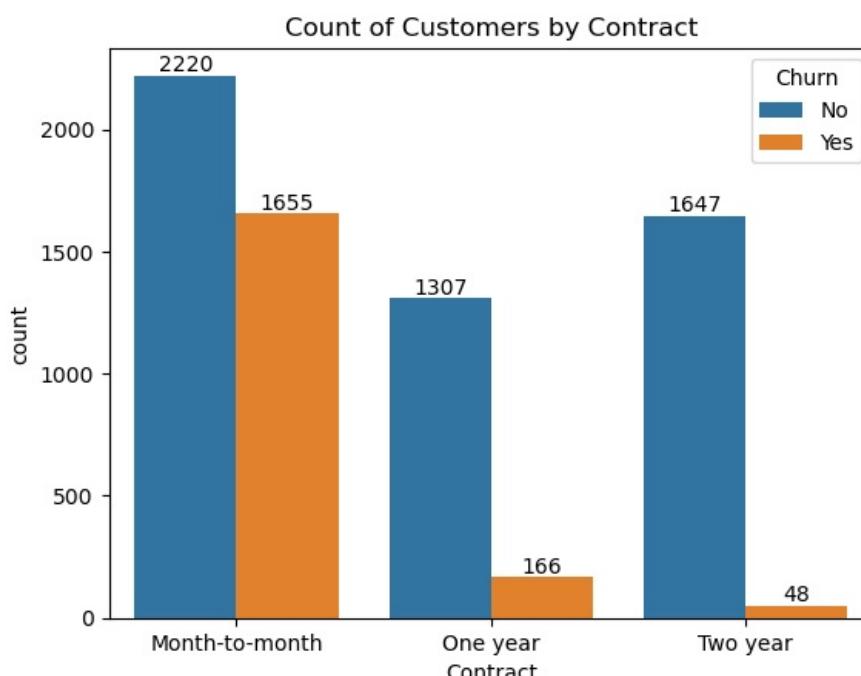
plt.title("Count of Customers by Contract")
plt.show()
```



```
In [81]: ax = sns.countplot(x="Contract", data=df,hue="Churn")

# Add labels to bars
for container in ax.containers:
    ax.bar_label(container)

plt.title("Count of Customers by Contract")
plt.show()
```



```
In [82]: # people who have month to month contract are likely to churn then from those who have 1 or 2 year of contract
```

```
In [83]: df.columns.values
```

```
Out[83]: array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

```
In [85]: cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

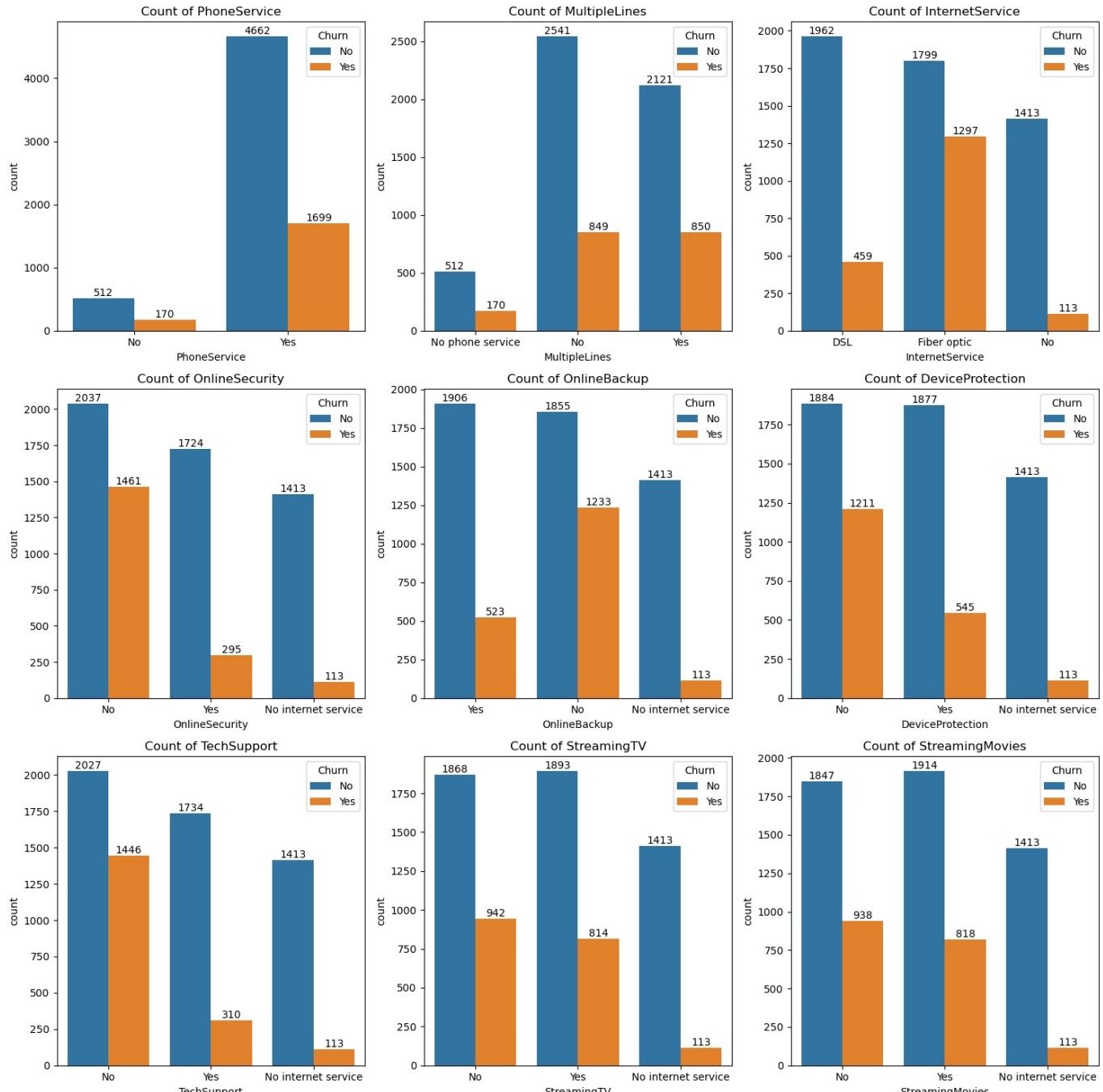
plt.figure(figsize=(15, 15))

for i, col in enumerate(cols, 1):
    plt.subplot(3, 3, i)
    ax = sns.countplot(x=col, data=df, hue="Churn")

    # Add bar labels
    for container in ax.containers:
        ax.bar_label(container)

    plt.title(f"Count of {col}")
    plt.tight_layout()

plt.show()
```

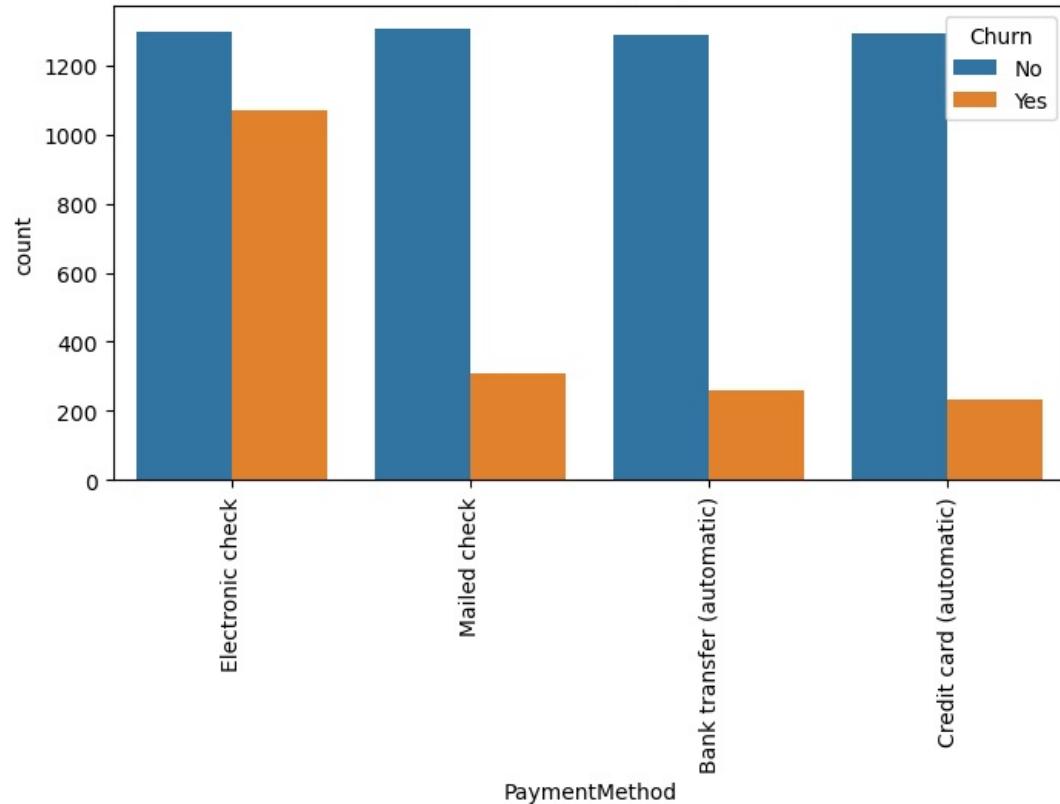


```
In [86]: #Customers without security or tech-support services churn the most, making these the strongest churn drivers. !
```

```
In [93]: plt.figure(figsize=(8,4))
sns.countplot(x = "PaymentMethod", data = df, hue="Churn")
plt.title("Churned of customer by PaymentMethod")
plt.xticks(rotation=90)

plt.show()
```

Churned of customer by PaymentMethod



```
In [94]: # customer is likely to churn when he is using electronic check as payment method.
```

```
In [ ]:
```