

```

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.info()
df.head()

<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

```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
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-CF0CW	Male	0	No	No	45
4	9237-HQITU	Female	0	No	No	2

	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 = pd.read_csv('Customer_Churn.csv')
df.info()
df.head()
```

```
<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

	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-CF0CW	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["TotalCharges"]=df["TotalCharges"].replace(" ", "0") # initially
putting 0 in quotes as its in integer
df["TotalCharges"]=df["TotalCharges"].astype("float") # changing the
datatype of this
```

```
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
```

```
df.isnull().sum().sum() # you can do 2 sum of find all the values null
in the df in one go
```

```
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

```
df.duplicated().sum() # so 0 rows are duplicated
# but not always just do this on the basis of the entire row we need
to check for some customer id or something
```

```
0
```

```
df["customerID"].duplicated().sum # as maybe same customer id in diff
rows but diff details so that is still incorrect data
```

```
0
```

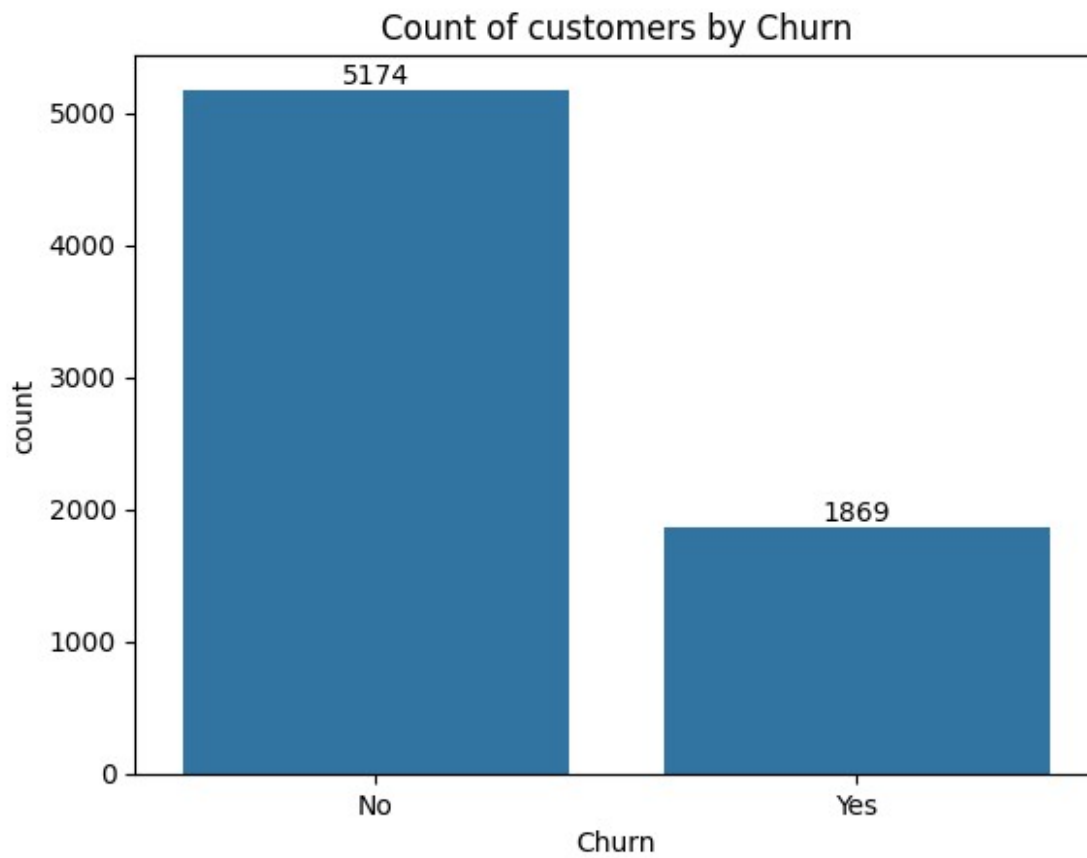
```
# converted 0 and 1 value of the senior citizen to yes and no to make
it easier to understand
```

```
def conv(value):
    if value==1:
        return "Yes"
    else:
        return "No"
```

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

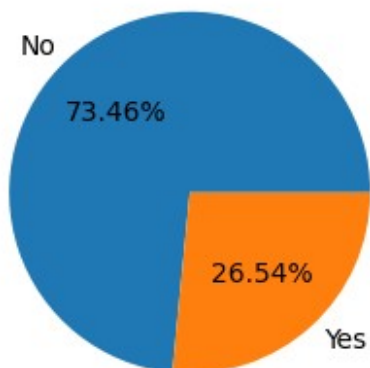
```
ax = sns.countplot(x='Churn',data=df)
```

```
ax.bar_label(ax.containers[0]) # to get the number on top of the bar
ax.set_title("Count of customers by Churn")
plt.show()
```



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

Percentage of Churned Customers



```

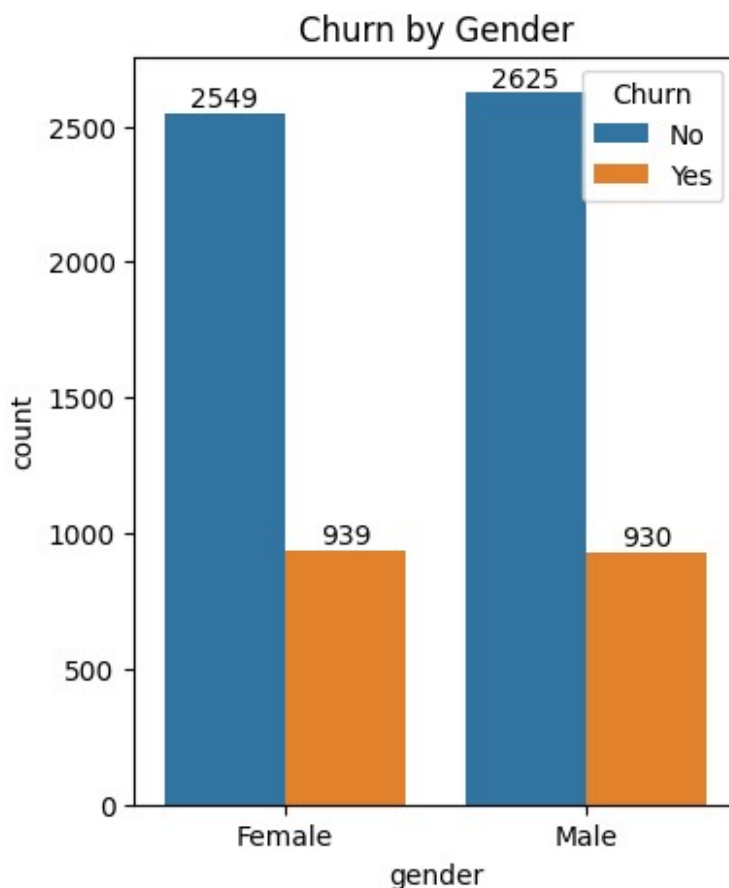
# from the given plot we can say that 25.54% of our customers have
churned out
# not lets explore the reason behind it

plt.figure(figsize=(4,5))
ax=sns.countplot(x = df['gender'],data=df, hue = "Churn")

# ax.bar_label(ax.containers[0]) # to get the number on top of the
# Add bar labels for all containers (groups of bars)
for container in ax.containers:
    ax.bar_label(container) # Add labels to each bar group

plt.title("Churn by Gender")
plt.show()

```



```

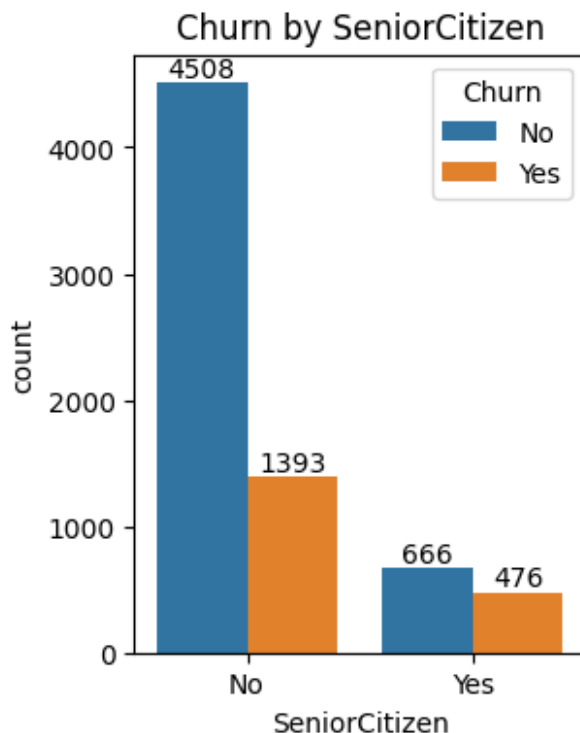
plt.figure(figsize=(3,4))
ax=sns.countplot(x = df['SeniorCitizen'],data=df, hue = "Churn")

ax.set_xticks([0, 1]) # Explicitly set tick positions we dont need it
but to be safe as seniorcitizen column is already 0 or 1
ax.set_xticklabels(["No", "Yes"])

```

```
# # Add bar labels for all containers (groups of bars)
for container in ax.containers:
    ax.bar_label(container) # Add labels to each bar group

plt.title("Churn by SeniorCitizen")
plt.show()
```

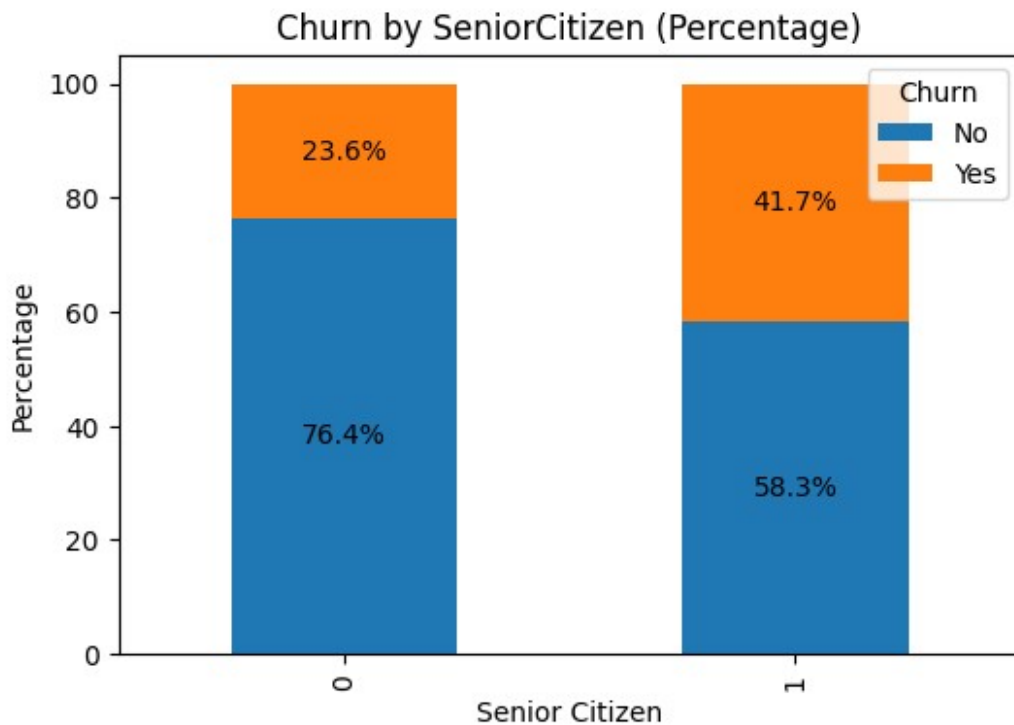


```
grouped = df.groupby(['SeniorCitizen', 'Churn']).size().unstack()
grouped = grouped.div(grouped.sum(axis=1), axis=0) * 100 # Calculate percentages

# Plot stacked bar chart
ax = grouped.plot(kind='bar', stacked=True, figsize=(6, 4))

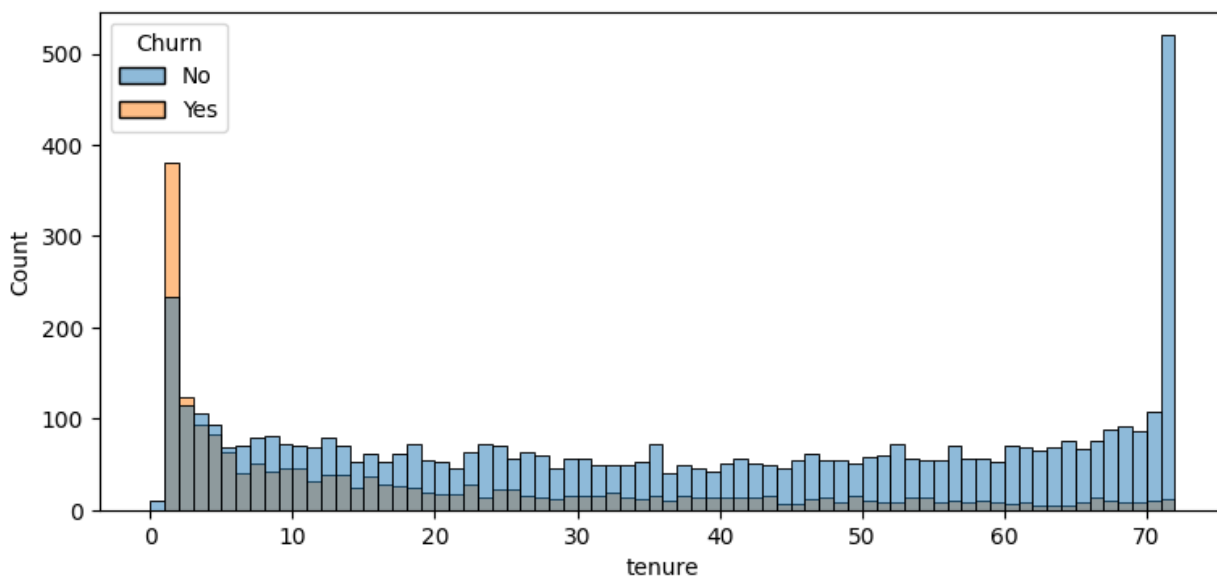
# Add percentage labels
for container in ax.containers:
    ax.bar_label(container, fmt='%.1f%%', label_type='center') # Add percentage labels

# Customize the plot
plt.title("Churn by SeniorCitizen (Percentage)")
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.legend(title="Churn", loc="upper right")
plt.show()
```

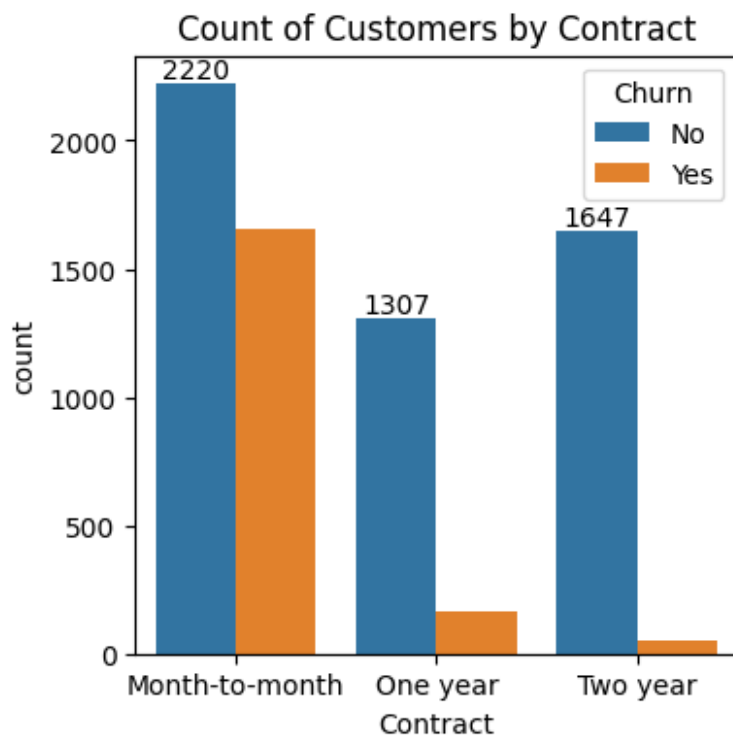
comparatively a greater percentage of people in the senior citizen category have churned out

```
plt.figure(figsize=(9,4))
sns.histplot(x="tenure" , data = df, bins=72, hue="Churn") # we did 72
bins as might look clearer as 72 people tenure we have
plt.show()
```



people who have used our services for a long time have stayed and people who have used our services for 1 or 2 months 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 that from those who have 1 or 2 years of 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 columns to plot (categorical columns)

```
columns_to_plot = ['gender', 'SeniorCitizen', 'Partner', 'Dependents',  
                  'PhoneService', 'MultipleLines', 'InternetService',
```

```

        'OnlineSecurity', 'OnlineBackup',
'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies',
        'Contract', 'PaperlessBilling', 'PaymentMethod',
'Churn']

# Create a grid of subplots
num_cols = 3 # Number of columns in the subplot grid
num_rows = (len(columns_to_plot) + num_cols - 1) // num_cols #
Calculate rows needed

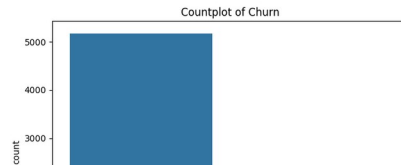
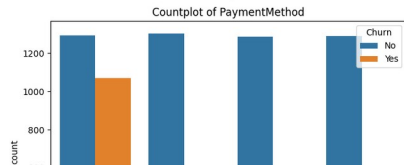
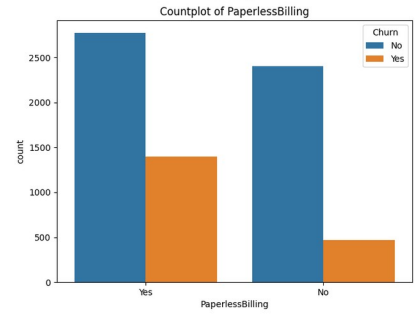
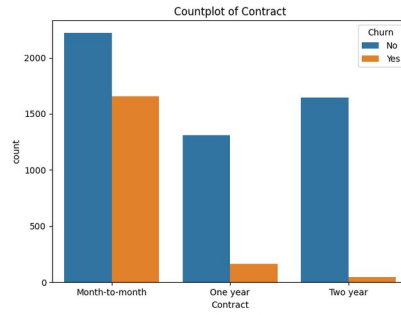
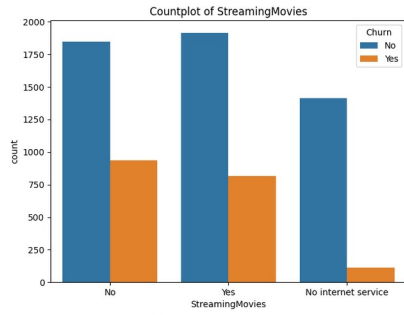
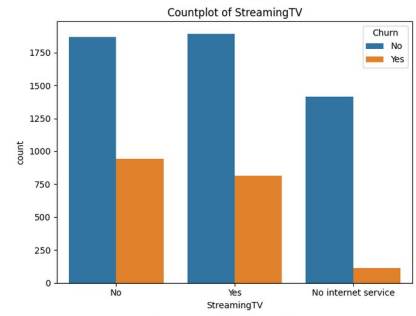
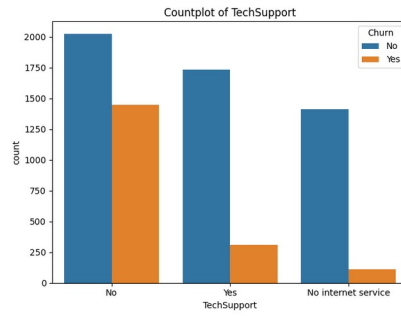
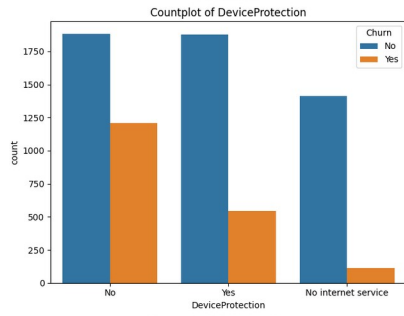
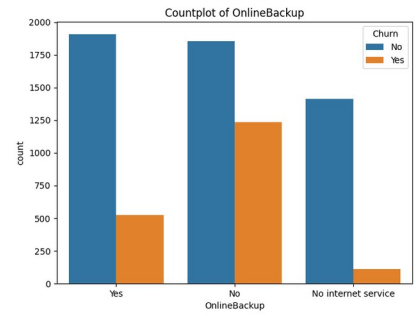
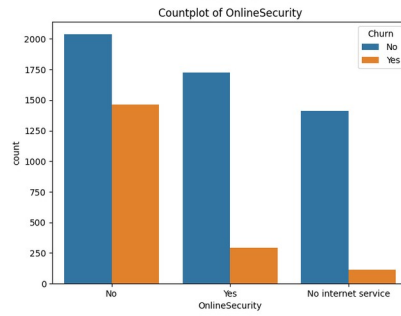
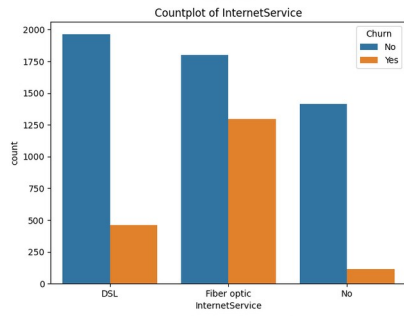
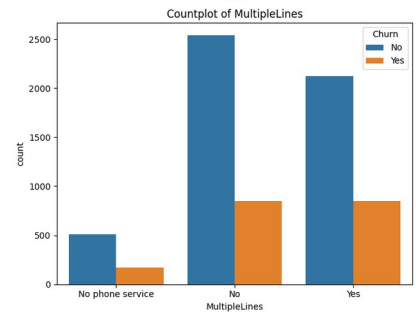
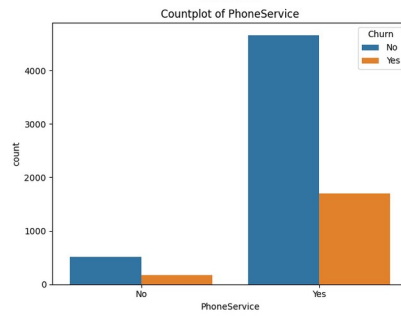
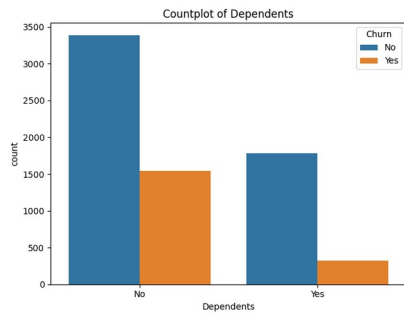
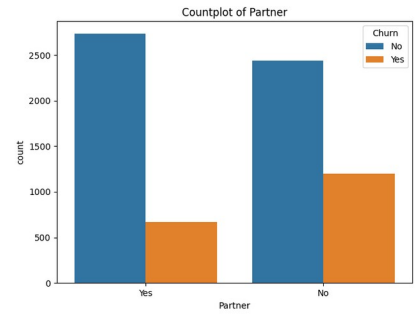
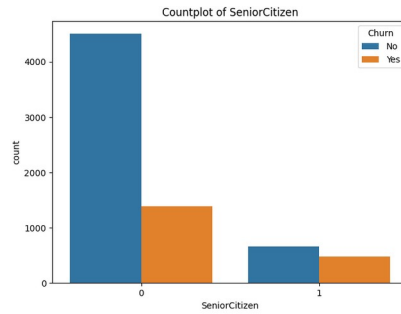
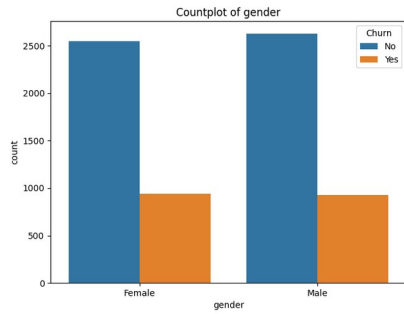
fig, axes = plt.subplots(num_rows, num_cols, figsize=(20, 5 *
num_rows))
axes = axes.flatten() # Flatten the 2D array of axes for easy
iteration

# Loop through columns and plot countplots
for i, column in enumerate(columns_to_plot):
    sns.countplot(x=df[column], ax=axes[i], hue = df["Churn"]) #
Create countplot in the ith subplot
    axes[i].set_title(f'Countplot of {column}') # Set title for the
subplot
    # axes[i].tick_params(axis='x', rotation=45) # Rotate x-axis
labels for readability

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

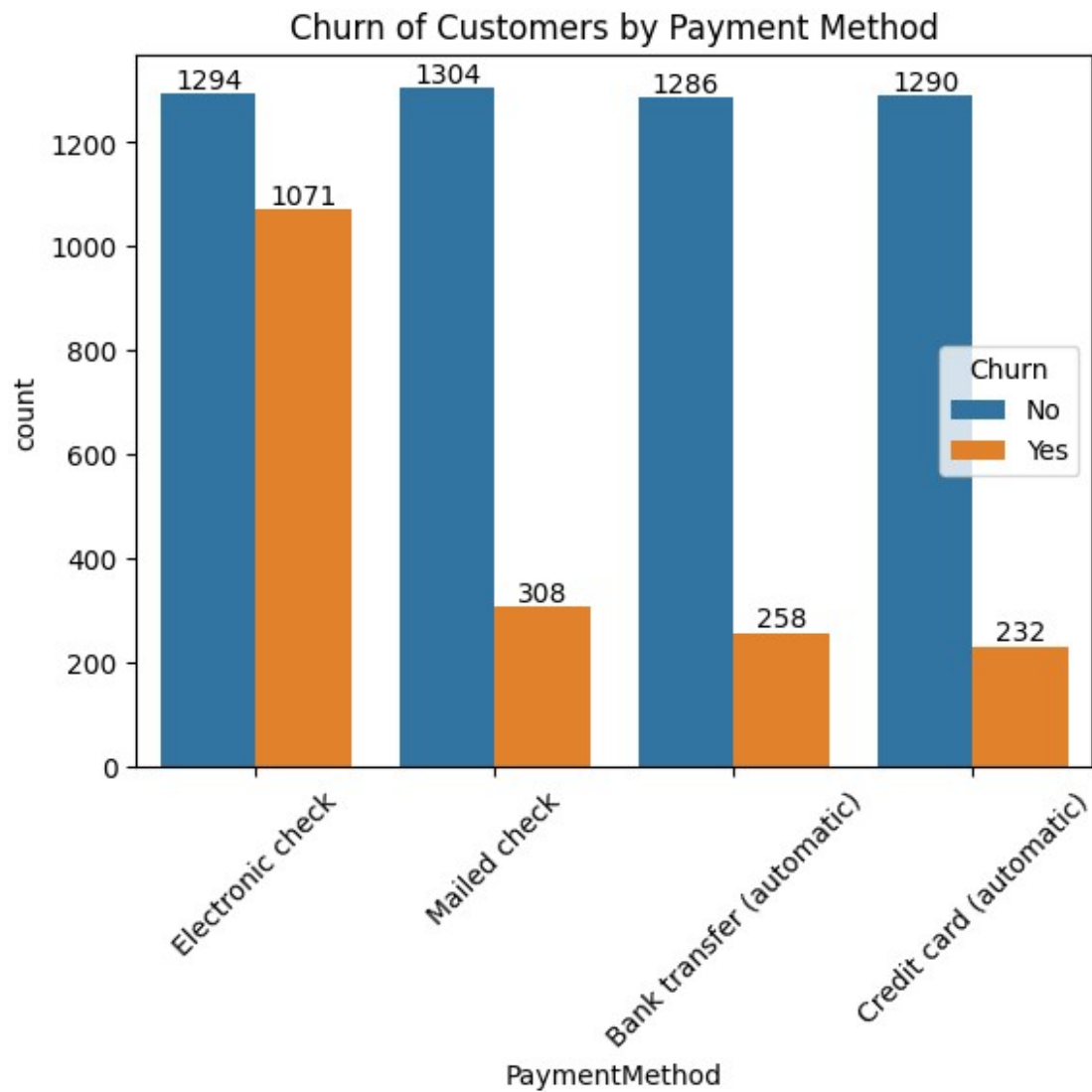
plt.tight_layout() # Adjust layout to prevent overlap
plt.show()

```



The visualizations present count plots showing customer churn (Yes or No) across various service features such as PhoneService, MultipleLines, InternetService, and OnlineSecurity. The majority of customers who do not churn tend to have services like PhoneService, InternetService (particularly DSL), and OnlineSecurity enabled. For services like OnlineBackup, TechSupport, and StreamingTV, churn rates are noticeably higher when these services are not used or are unavailable. The charts emphasize the relationship between service usage and customer churn.

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



Customer is likely to churn when he is using electronic check as a payment method