

```

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

df = pd.read_csv(r"C:\Users\badri\OneDrive\Desktop\Customer churn
analysis\archive (1)\WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.info()
df.head()
df.dtypes

```

```

<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	object
gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object

```
OnlineBackup      object
DeviceProtection  object
TechSupport       object
StreamingTV       object
StreamingMovies   object
Contract          object
PaperlessBilling  object
PaymentMethod     object
MonthlyCharges    float64
TotalCharges      object
Churn             object
dtype: object
```

#replacing blanks with 0 as tenure is 0 and no total charges are recorded

#Converting TotalCharges(Object) into float data type

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

df.dtypes

```
customerID      object
gender          object
SeniorCitizen   int64
Partner         object
Dependents      object
tenure          int64
PhoneService    object
MultipleLines   object
InternetService object
OnlineSecurity  object
OnlineBackup    object
DeviceProtection object
TechSupport     object
StreamingTV     object
StreamingMovies object
Contract        object
PaperlessBilling object
PaymentMethod   object
MonthlyCharges  float64
TotalCharges    float64
Churn           object
dtype: object
```

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

#Check whether our data contain any null values

```
df.isnull().sum().sum()
np.int64(0)
```

#check whether the data contains any duplicates.

```
df.duplicated().sum()
df["customerID"].duplicated().sum()
np.int64(0)
```

#Converting Seniorcitizen column values(0 and 1) to Yes/no for better understanding

```
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

df['SeniorCitizen'] = df['SeniorCitizen'].apply(conv)
df.tail()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure \
7038	6840-RESVB	Male	no	Yes	Yes	24
7039	2234-XADUH	Female	no	Yes	Yes	72
7040	4801-JZAZL	Female	no	Yes	Yes	11
7041	8361-LTMKD	Male	yes	Yes	No	4
7042	3186-AJIEK	Male	no	No	No	66

	PhoneService	MultipleLines	InternetService
OnlineSecurity ... \			
7038	Yes	Yes	DSL
Yes ...			
7039	Yes	Yes	Fiber optic
No ...			
7040	No	No phone service	DSL
Yes ...			
7041	Yes	Yes	Fiber optic
No ...			

7042	Yes	No	Fiber optic
Yes ...			

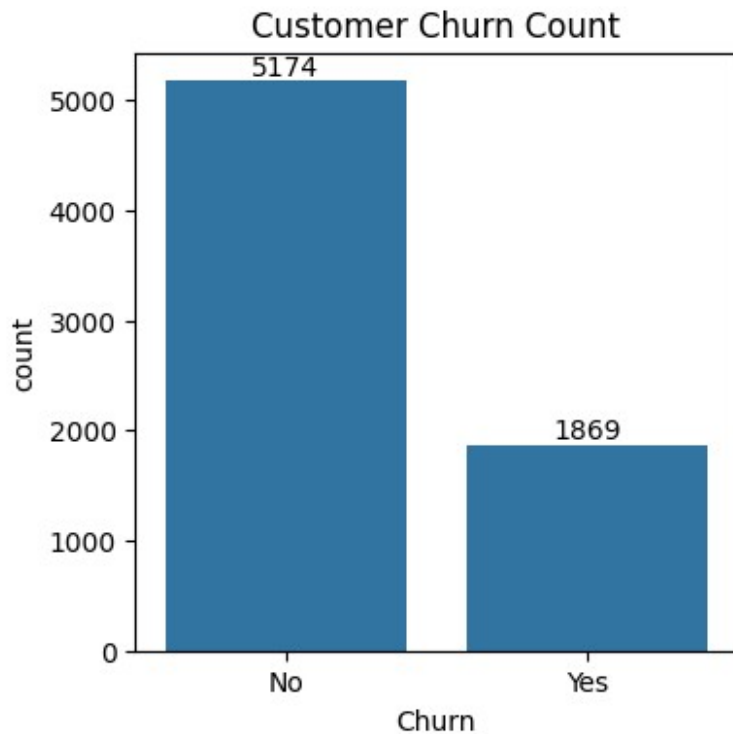
	DeviceProtection	TechSupport	StreamingTV	StreamingMovies
Contract \				
7038	Yes	Yes	Yes	Yes
One year				
7039	Yes	No	Yes	Yes
One year				
7040	No	No	No	No
to-month				
7041	No	No	No	No
to-month				
7042	Yes	Yes	Yes	Yes
Two year				

	PaperlessBilling	PaymentMethod	MonthlyCharges
TotalCharges \			
7038	Yes	Mailed check	84.80
1990.50			
7039	Yes	Credit card (automatic)	103.20
7362.90			
7040	Yes	Electronic check	29.60
346.45			
7041	Yes	Mailed check	74.40
306.60			
7042	Yes	Bank transfer (automatic)	105.65
6844.50			

	Churn
7038	No
7039	No
7040	No
7041	Yes
7042	No

[5 rows x 21 columns]

```
plt.figure(figsize = (4,4))
ax = sns.countplot(x='Churn', data=df)
plt.title('Customer Churn Count')
ax.bar_label(ax.containers[0])
plt.show()
```

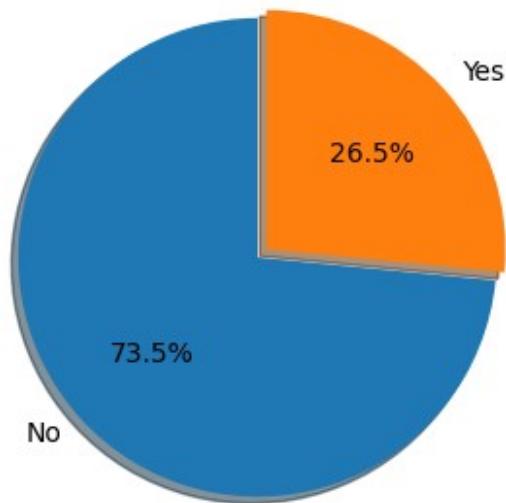


```
churn_counts = df['Churn'].value_counts()

# Plot pie chart
plt.figure(figsize=(4,4))
plt.pie(churn_counts,
        labels=churn_counts.index,
        autopct='%1.1f%%',           # show percentages (like 26.5%)
        startangle=90,              # rotate start
        explode=[0, 0.05],          # slight separation
        shadow=True)

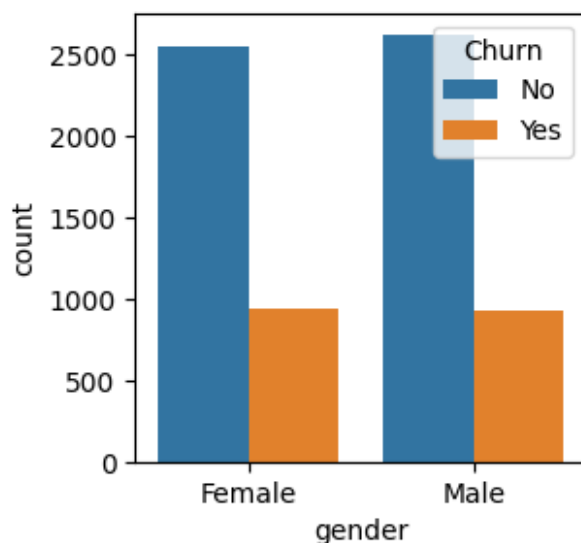
plt.title('Customer Churn Percentage')
plt.show()
```

Customer Churn Percentage



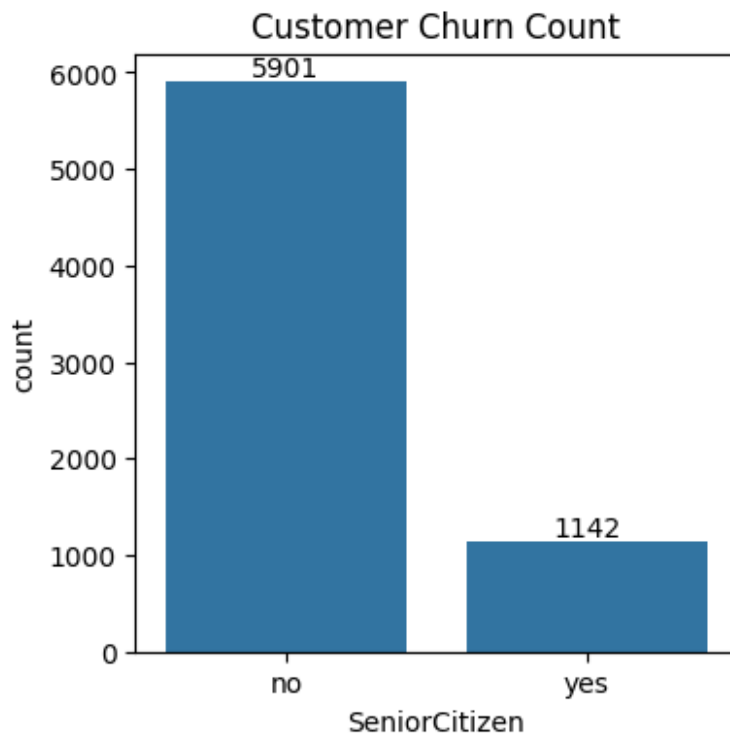
#From the above chart we can conclude that 26.5% of our customers have churned out. Now explore the reason behind it

```
plt.figure(figsize= (3,3))
sns.countplot(x = "gender", data =df, hue = "Churn")
plt.title("Churn by gender")
plt.show()
```



```
plt.figure(figsize = (4,4))
ax = sns.countplot(x='SeniorCitizen', data=df)
plt.title('Customer Churn Count')
```

```
ax.bar_label(ax.containers[0])
plt.show()
```



```
# Step 1: Create a cross-tab (counts)
churn_senior = pd.crosstab(df['SeniorCitizen'], df['Churn'])

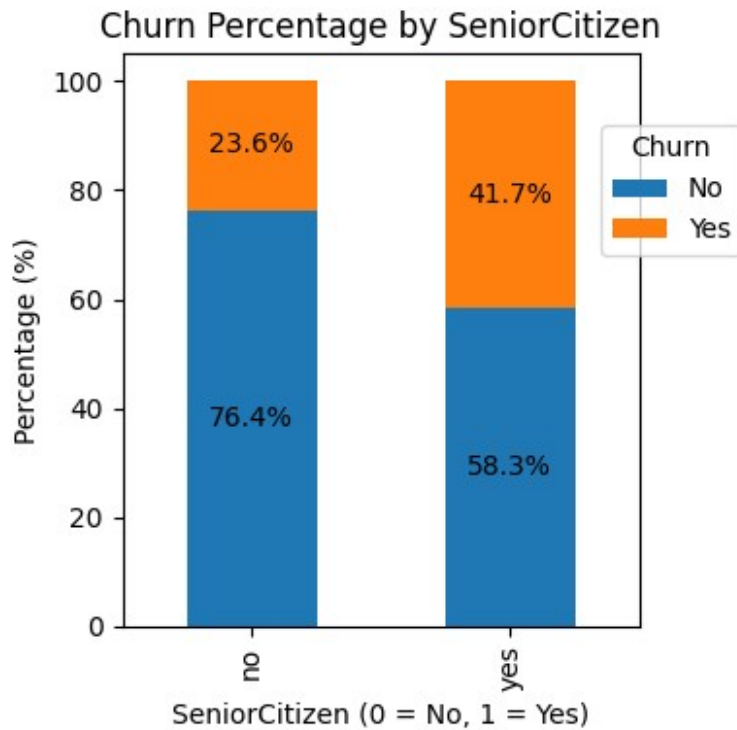
# Step 2: Convert to percentage (row-wise)
churn_senior_pct = churn_senior.div(churn_senior.sum(axis=1), axis=0)
* 100

# Step 3: Plot stacked bar chart
ax = churn_senior_pct.plot(kind='bar', stacked=True, figsize=(4,4))

plt.title('Churn Percentage by SeniorCitizen')
plt.xlabel('SeniorCitizen (0 = No, 1 = Yes)')
plt.ylabel('Percentage (%)')

# Step 4: Add percentage labels on each segment
for c in ax.containers:
    ax.bar_label(c, fmt='%.1f%', label_type='center')

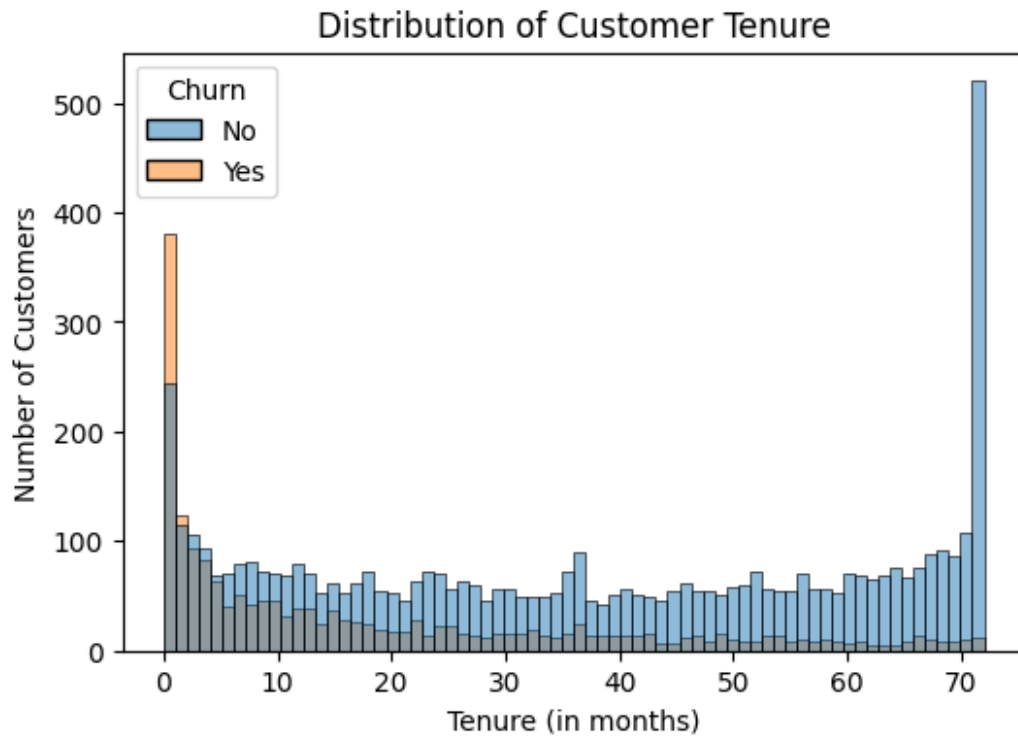
plt.legend(title='Churn', bbox_to_anchor= (0.9,0.9))
plt.tight_layout()
plt.show()
```



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

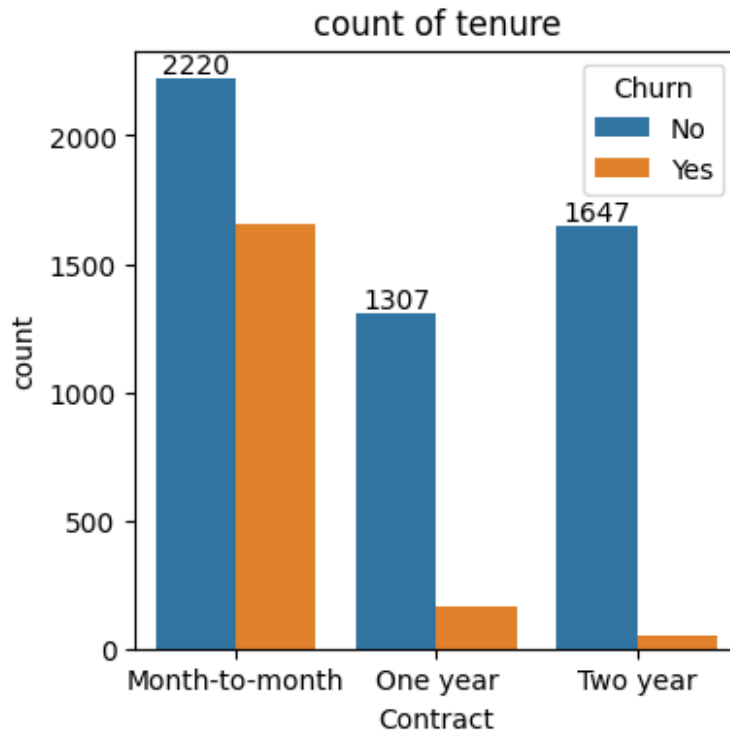
```
plt.figure(figsize=(6,4))
sns.histplot(x= "tenure",data= df, bins=70, hue = "Churn")

plt.title('Distribution of Customer Tenure')
plt.xlabel('Tenure (in months)')
plt.ylabel('Number of Customers')
plt.show()
```

#People who have used our services for longer time have stayed and people who have used our services for one or two months have churned out.

```
plt.figure(figsize= (4,4))
ax= sns.countplot(x = "Contract", data = df, hue = "Churn")
plt.title("count of tenure")
ax.bar_label(ax.containers[0])
plt.show()
```



#People who have month to month contract are likely to churn then from those who have one or two year 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 categorical columns
columns = ['PhoneService', 'MultipleLines', 'InternetService',
          'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
          'TechSupport', 'StreamingTV', 'StreamingMovies']

# Number of columns per row for subplots
n_cols = 3
n_rows = (len(columns) + n_cols - 1) // n_cols # Calculate rows based
on number of plots

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4))

# Flatten the axes array for easy iteration
```

```

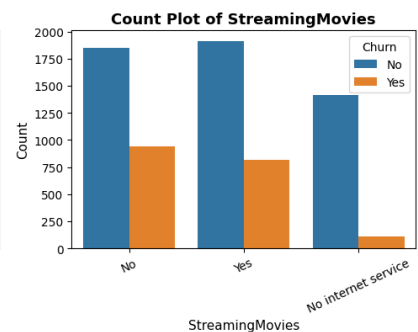
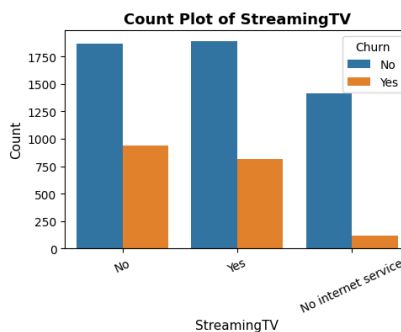
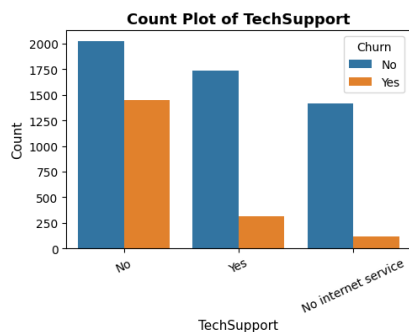
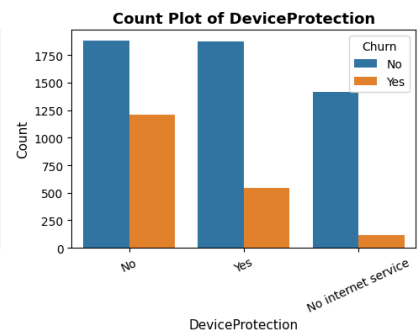
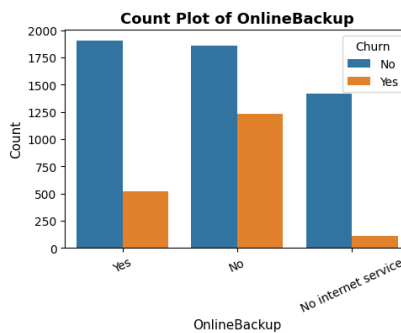
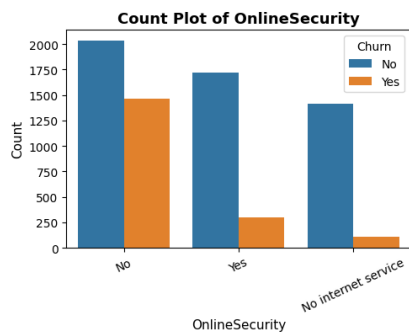
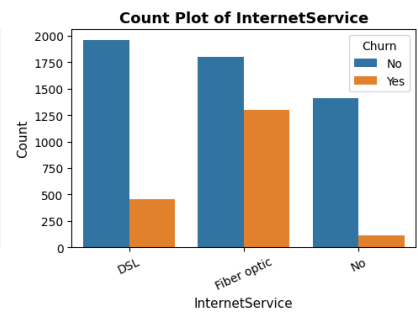
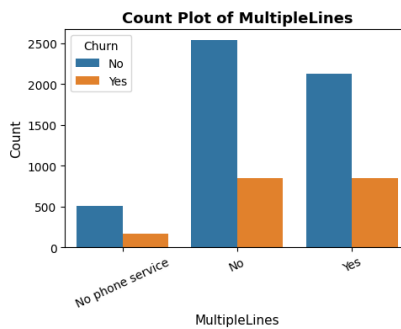
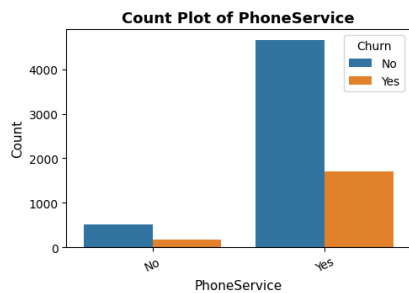
axes = axes.flatten()

# Iterate over columns and plot count plots
for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue = "Churn")
    axes[i].set_title(f'Count Plot of {col}', fontsize=13,
fontweight='bold')
    axes[i].set_xlabel(col, fontsize=11)
    axes[i].set_ylabel('Count', fontsize=11)
    axes[i].tick_params(axis='x', rotation=25)

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

plt.tight_layout()
plt.show()

```

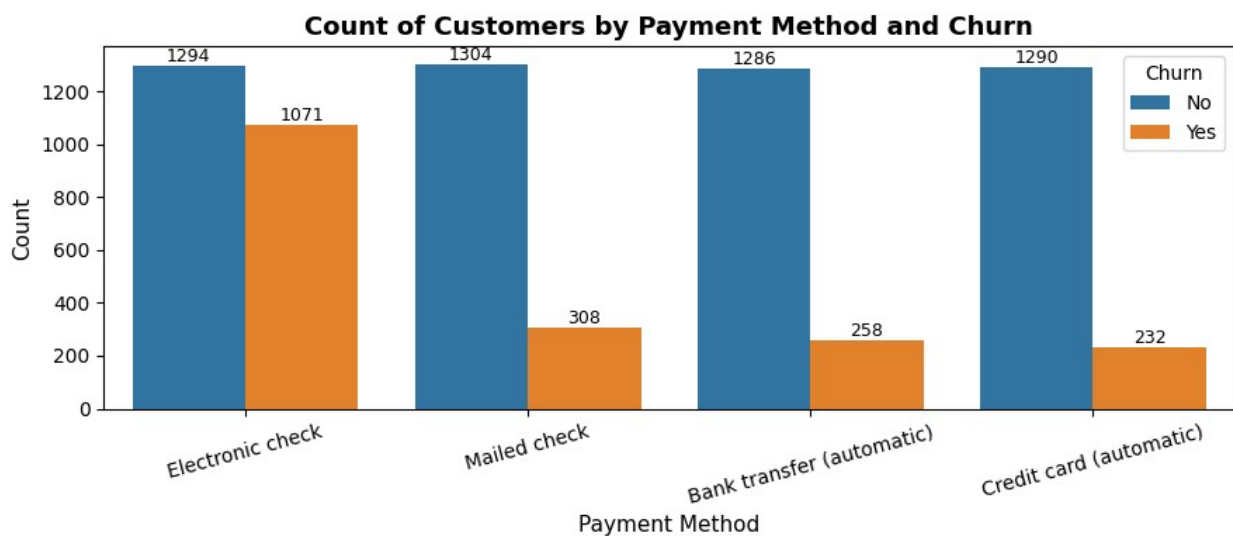


#The visual analysis shows that customers who lack additional services like OnlineSecurity, TechSupport, and DeviceProtection are more likely to churn. Fiber optic users exhibit higher churn rates compared to DSL users.

```
plt.figure(figsize=(9,4))
ax = sns.countplot(x="PaymentMethod", data=df, hue="Churn")
plt.title("Count of Customers by Payment Method and Churn",
          fontsize=13, fontweight='bold')
plt.xlabel("Payment Method", fontsize=11)
plt.ylabel("Count", fontsize=11)

# Add count labels on both churn bars
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', fontsize=9)

plt.xticks(rotation=15)
plt.tight_layout()
plt.show()
```



#Customers paying via Electronic Check churn the most, likely due to inconvenience, lack of automation, and fewer loyalty benefits compared to automatic payment users.

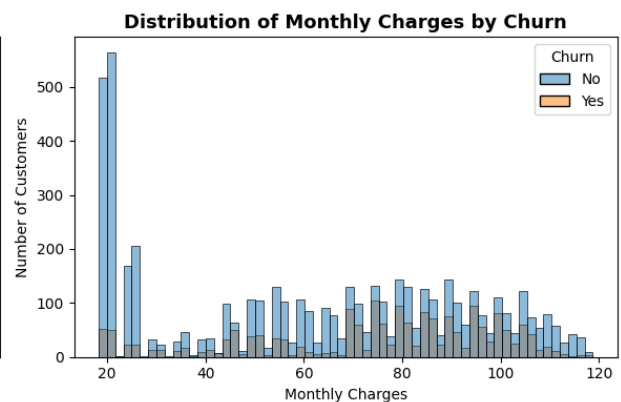
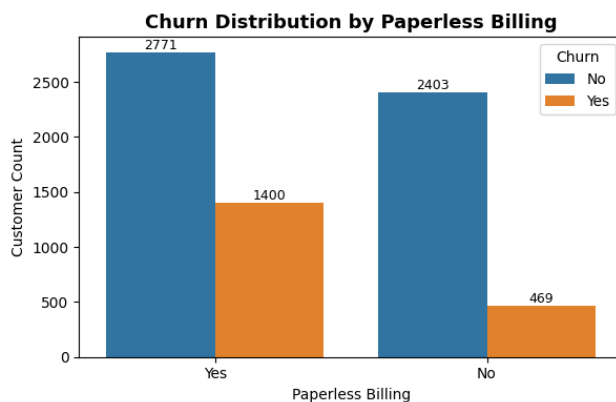
```
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

# --- | Count Plot for Paperless Billing ---
ax1 = sns.countplot(x='PaperlessBilling', hue='Churn', data=df,
                    ax=axes[0])
ax1.set_title('Churn Distribution by Paperless Billing', fontsize=13,
              fontweight='bold')
ax1.set_xlabel('Paperless Billing')
ax1.set_ylabel('Customer Count')
```

```
# Add count labels
for container in ax1.containers:
    ax1.bar_label(container, fmt='%d', label_type='edge', fontsize=9)

2 --- 2 Histogram for Monthly Charges ---
sns.histplot(data=df, x='MonthlyCharges', hue='Churn', bins= 60,
ax=axes[1])
axes[1].set_title('Distribution of Monthly Charges by Churn',
fontsize=13, fontweight='bold')
axes[1].set_xlabel('Monthly Charges')
axes[1].set_ylabel('Number of Customers')

# Adjust layout
plt.tight_layout()
plt.show()
```



#Customers who opted for paperless billing have a higher churn rate compared to those who receive paper bills and From the histogram, customers with higher monthly charges (above ₹70) show a noticeably higher churn rate.