Summary & Recommendations

Executive Summary

The analysis focuses on understanding the factors contributing to customer churn within a telecommunications company by examining a dataset containing customer demographics, service usage, and account details. Various visualizations, such as count plots, pie charts, and histograms, are used to analyze churn behavior across multiple variables.

Key Findings

1. Churn By Contract Type:
- Customers with month-to-month contracts are the most likely to churn, with a churn rate of approximately 42%. In contrast, customers with one-year and two-year contracts have much lower churn rates of 11% and 3%, respectively. This suggests that short-term contracts lead to higher instability in customer retention.

2. Churn By Service Availability:
- The absence of certain services is a strong indicator of churn. For instance:

- Customers without online security show a churn rate of 45%, whereas those with online security have a much lower churn rate of 15%.

- Similarly, customers without tech support have a churn rate of 41%, compared to 12% for those with tech support.

- The lack of multiple lines and device protection also correlates with higher churn rates.

- Interestingly, customers without internet service are the least likely to churn (around 7%), indicating that internet services might not be the core driver for this segment.

3. Payment Method Impact:
- Electronic check users have the highest churn rate at 44%, suggesting that customers using this method might have lower loyalty or higher dissatisfaction.

- By contrast, customers paying via credit card or bank transfer churn at rates closer to 16-17%, indicating more stable payment patterns.

4. Churn By Tenure:

- Newer customers (with tenure less than 12 months) have the highest churn rate at 50%. This suggests dissatisfaction or unmet expectations during the initial period.

- Interestingly, customers with a tenure between 12 to 24 months have a churn rate of 15-18%, and those with a tenure of more than 24 months exhibit the lowest churn rate at 5-7%. This points to customer loyalty strengthening over time.

5. Churn By Gender And Senior Citizen:

- Gender does not appear to have a significant impact on churn rates, as the churn percentages for both male and female customers are similar (around 27-28%).

- However, Senior Citizens have a notably higher churn rate of 42%, compared to 25% for non-senior citizens, indicating that older customers may be more inclined to leave the service.

Visual Insights:

I have created several count plots and pie charts to visualize churn by different customer attributes. Key insights from these visualizations include:

- Churn distribution shows that 26.5% of customers in the dataset have churned, with the remaining 73.5% being retained.

- Service-related factors like internet service, multiple lines, and online security play a crucial role in whether customers leave or stay. Higher churn rates are seen when these services are absent.

- The stacked bar chart for Senior Citizens highlights that a greater proportion of this demographic tends to churn.

Conclusion:

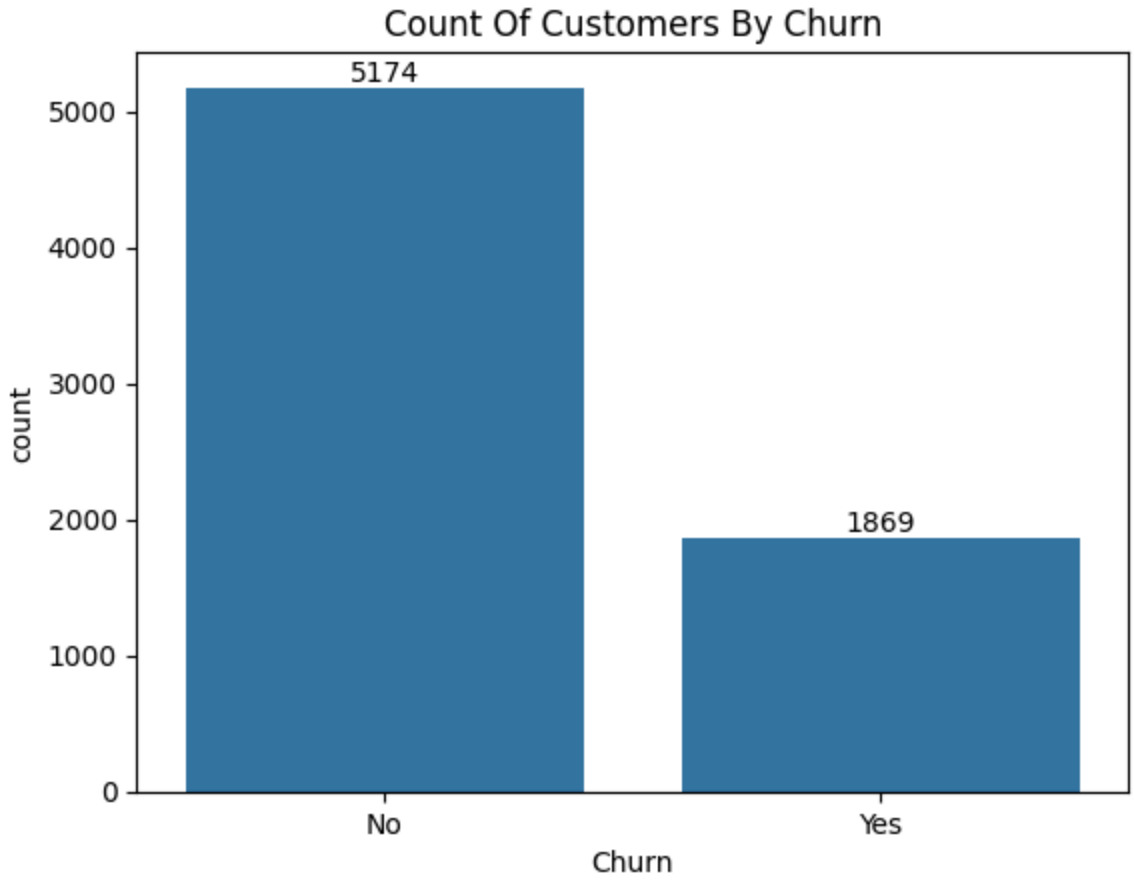
The analysis reveals that churn is heavily influenced by service features (like online security, tech support), contract length, payment methods, and customer tenure.

Special attention should be given to month-to-month contract holders, senior citizens, and new customers to reduce churn rates. By improving service offerings and incentivizing longer-term contracts, the company could significantly enhance customer retention.

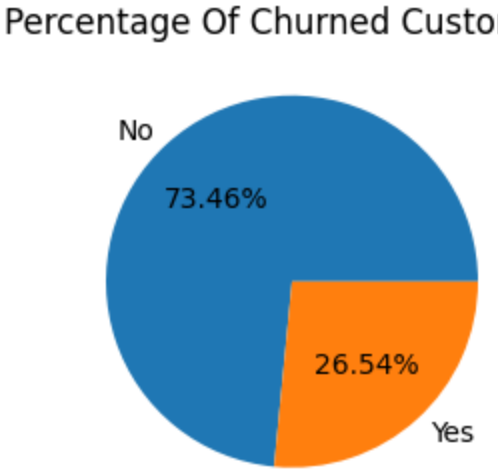
The visual representations offer a comprehensive understanding of the factors driving customer churn, providing valuable insights for making strategic business decisions.

Following are the charts that I created for my analysis:-

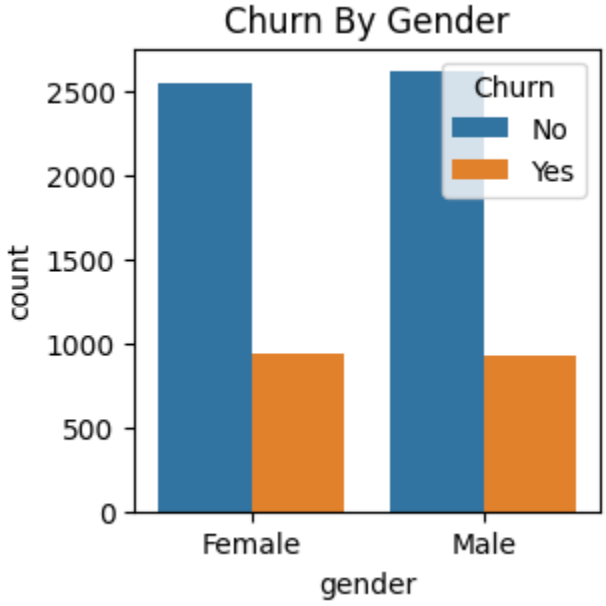
1. **Count Of Customers By Churn**



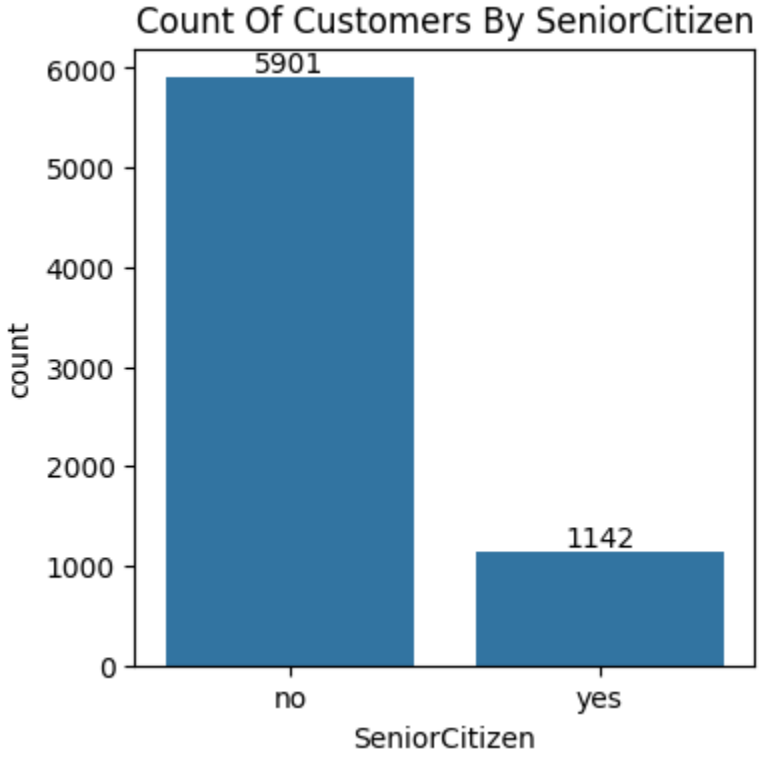
2. **Percentage Of Churned Customers**



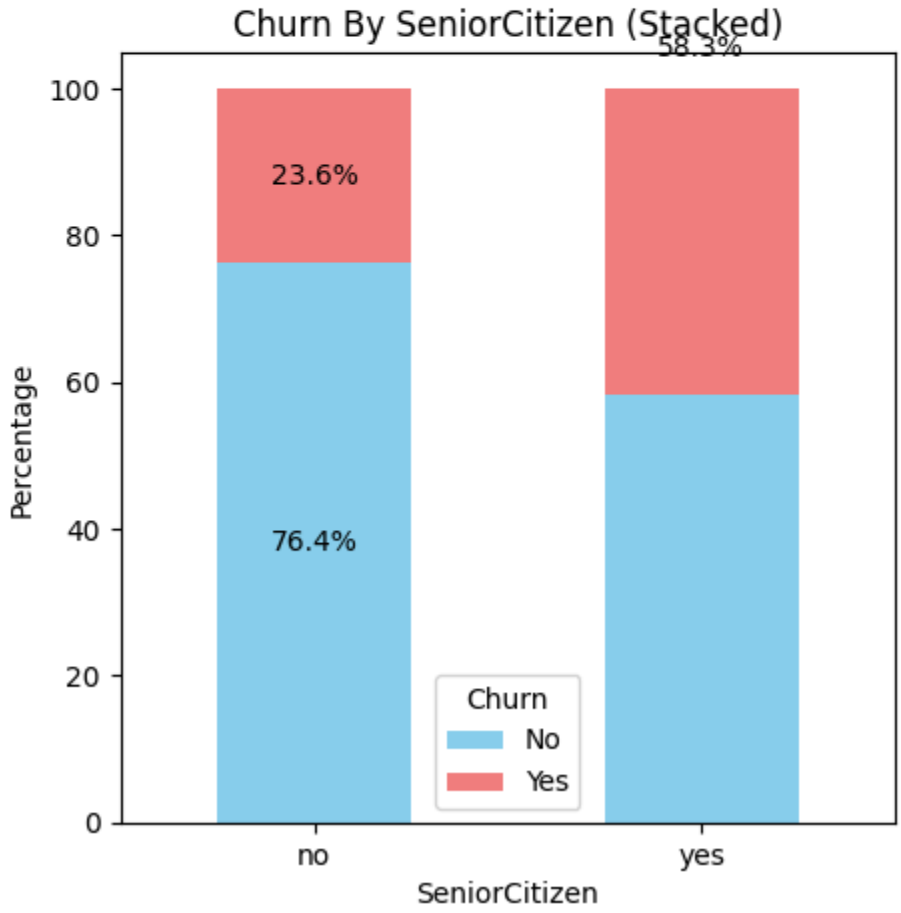
3. Churn By Gender



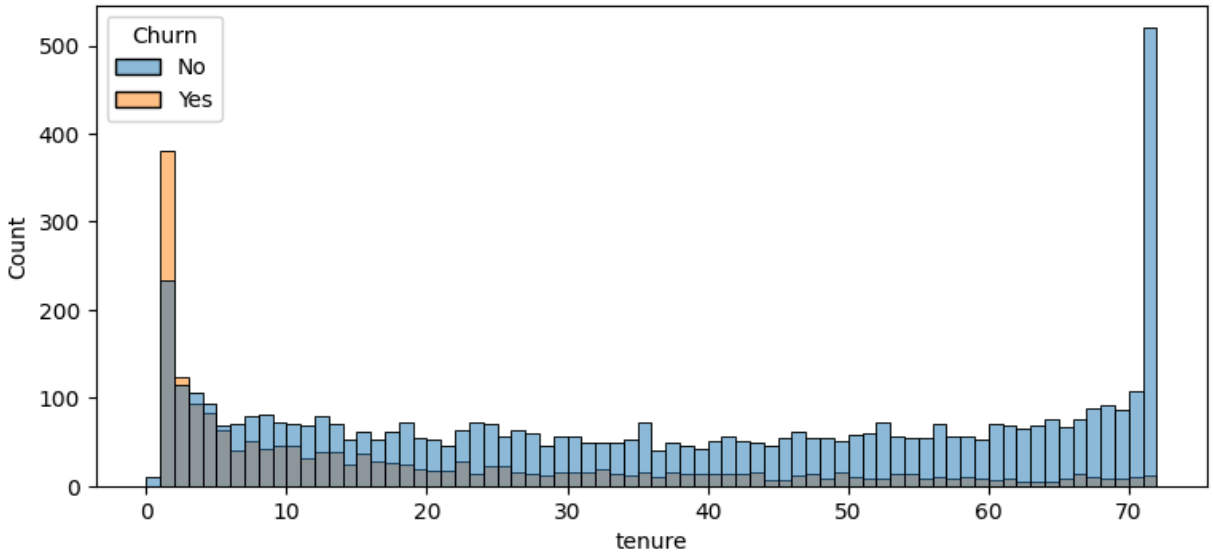
4. Count Of Customers By SeniorCitizen



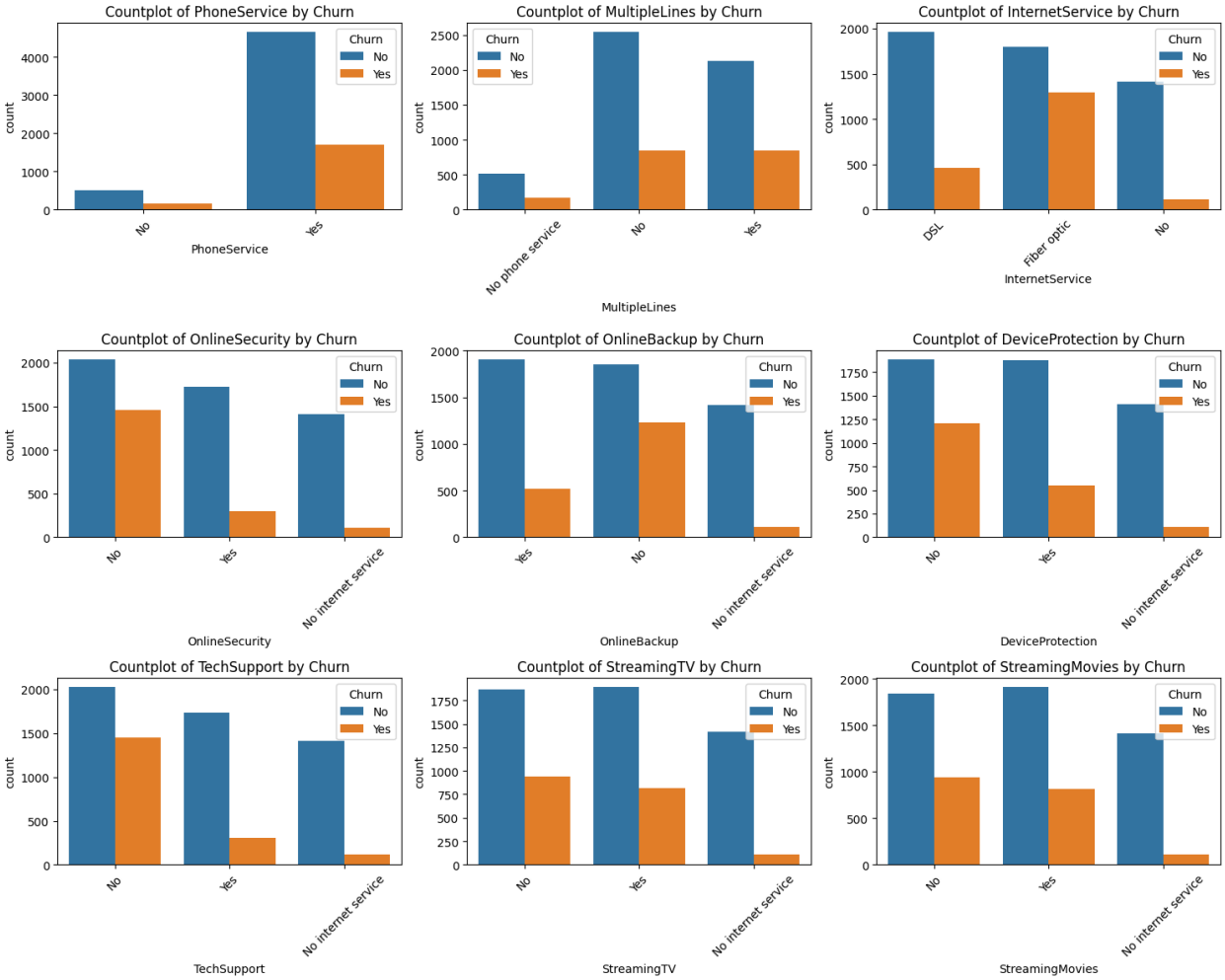
5. Churn By SeniorCitizen



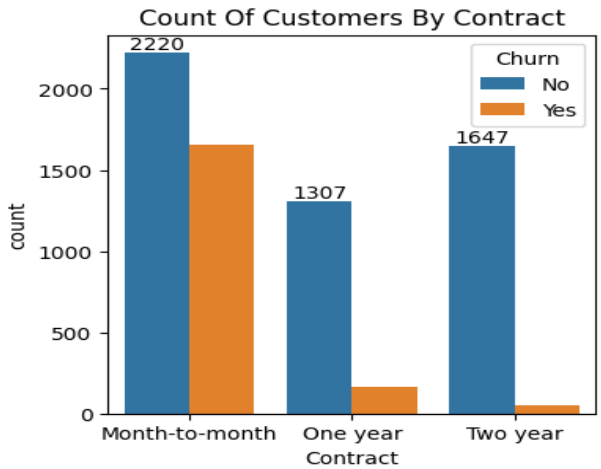
6. Churn By Tenure



7. Countplot Of All The Services



8. Count Of Customers By Contract



Code:-

```
# Installing and importing libraries
#As this is a python script running in visual studio code, we always need to use print
#command to see the output but in other IDEs like jupyter notebook it does not require
# to use print command, it will automatically show the output

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


#Now understand your dataset i.e Exploratory Data Analysis(EDA)

#importing file by creating dataframe using df command
#I got an error when i wrote the name of the csv file, therefore give full path
df=pd.read_csv('D:\Shivranjan Jogwar\Coding\Python\Python Project\Churn Analysis\Customer
Churn.csv')

#we can simply write "df" but it was not showing the output,
# therefore i used the print command
print(df)

#This function allows us to see first 5 rows in the data
df.head(10)

#In EDA first step is to inspect our data
#This function gives us the not-null values and displays the total
# number of rows and columns with its datatype
df.info()

#We have a column called "TotalCharges" with the datatype as "object"
# we will convert it to float
#Also when we view the data in excel after inserting it into the table
# we can see "TotalCharges" column has some blank values
# and tenure column is filled with 0s, so we will fill the empty values of
# "TotalCharges" column with 0s

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

df.info()
```

```
#This command is used to check whether there are blank of empty values in the dataset
#if there are blank values it will return value >0 else 0
print(df.isnull().sum())

#This command is used for descriptive analysis
print(df.describe())

#This is command is used to check duplicate values
print(df.duplicated().sum())

#This command is used to check duplicates for a specified column
print(df["customerID"].duplicated().sum())

#We are creating a function to convert values of "SeniorCitizen" column "0" & "1" to
# "yes" & "no"

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

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


#We are using this command to check whether we have yes/no
#in "SeniorCitizen" column in first 30 rows & columns
print(df.head(30))

#Now we will mention our analysis "on why customers churned out",
#"why they left using our services", "type of customers" & there characteristics

#This command is used to get a countplot
ax = sns.countplot(x='Churn', data=df)

ax.bar_label(ax.containers[0]) #We used this command to see exact
#number of customers who churned out etc...
print(plt.title("Count Of Customers By Churn"))
plt.show()

#This code is used to generate pie chart displayed with percentage
plt.figure(figsize=(3,4))
gb=df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels=gb.index, autopct="%1.2f%%")
```

```
print(plt.title("Percentage Of Churned Customers"))
plt.show()

#From the given pie chart for "Gender" we can conclude that 26.5% customers has churned out
#Lets 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()

#-----------------------------------------------Code:1-----------------------------------------------
#plt.figure(figsize=(3,3))
#sns.countplot(x="SeniorCitizen", data=df, hue="Churn")
#plt.title("Churn By SeniorCitizen")
#plt.show()
#-----------------------------------------------------------------------------------------------------

#Later the code was realtered for explanation
plt.figure(figsize=(4,4))
ax=sns.countplot(x="SeniorCitizen", data=df)
ax.bar_label(ax.containers[0])
plt.title("Count Of Customers By SeniorCitizen")
plt.show()

#Code from gpt after just providing the Code:1 and giving a prompt as:- "i want to create a stack bar chart which
#Creating a stack bar chart
# gives me labels as percentage of total"

# Assuming "churn_demo" is your DataFrame
# Example DataFrame
#This is new data that has only 2 columns in your dataframe, earlier we were working on 20 columns
# and now we are creating stack bar for the analysis of only these 2 columns
data = {'SeniorCitizen': [0, 1, 0, 1, 0, 1, 0, 1, 0],
        'Churn': ['Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'Yes', 'No']}

#this is new data frame
churn_demo = pd.DataFrame(data)

# Calculate the counts and percentages
churn_counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)
churn_percent = churn_counts.div(churn_counts.sum(axis=1), axis=0) * 100
```

```
# Plot stacked bar chart with percentages
fig, ax = plt.subplots(figsize=(5,5))

# Stacking bars using the churn percentages
churn_percent.plot(kind='bar', stacked=True, ax=ax, color=['skyblue', 'lightcoral'])

# Adding percentage labels
for i in range(len(churn_percent)):
    for j in range(len(churn_percent.columns)):
        percentage = churn_percent.iloc[i, j]
        if percentage > 0:  # Show label only if percentage is > 0
            ax.text(i, churn_percent.iloc[:i+1, :j+1].sum().sum() - percentage / 2, f'{percentage:.1f}%',
                    ha='center', va='center', color='black')

# Customize the plot
plt.title("Churn By SeniorCitizen (Stacked)")
plt.ylabel("Percentage")
plt.xlabel("SeniorCitizen")
plt.xticks(rotation=0)
plt.legend(title='Churn')
plt.show()

#Creating histogram of "tenure" column
plt.figure(figsize=(9,4))
sns.histplot(x="tenure", data=df, bins=72, hue="Churn")
plt.show()

#from the above chart we can conclude that people who have used our services for a long time
and people who
#hav used our services for 1 or 2 months has churned.

#Reason who have stayed might have had long duration of the contract, following is the count
on the basis of contract
plt.figure(figsize=(4,4))
ax=sns.countplot(x="Contract", data=df)
ax.bar_label(ax.containers[0])
plt.title("Count Of Customers By Contract")
plt.show()

#From the above analysis we can conclude that people who have month to month contract are
likely to churn then
#from those who have 1 or 2 years of contract
```

```
#-----------------------------------------------------------------------------------------------------------

#Taking the code from chatgpt to crete multiple countplots for the 9 columns for which business is providing
#its services

# List of columns for which we want to create countplots
columns_to_plot = ['PhoneService', 'MultipleLines', 'InternetService',
                   'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
                   'TechSupport', 'StreamingTV', 'StreamingMovies']

# Set up the number of rows and columns for subplots
n_cols = 3  # Number of columns for the subplot grid
n_rows = len(columns_to_plot) // n_cols + (len(columns_to_plot) % n_cols > 0)

# 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()

# Loop over the columns and create a countplot for each one
for i, column in enumerate(columns_to_plot):
    sns.countplot(x=column, data=df, hue="Churn", ax=axes[i])
    axes[i].set_title(f'Countplot of {column} by Churn')
    axes[i].tick_params(axis='x', rotation=45)

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

# Adjust the layout to prevent overlapping
plt.tight_layout()

# Display the plot
plt.show()

#From the above code we can conclude that:- The visualizations show countplots of several service-related columns
# (`PhoneService`, `MultipleLines`, `InternetService`, etc.) with the distribution of customers who have churned
# (`Yes`) versus those who have not (`No`). Generally, churn tends to be higher for customers with certain services,
```

```
# such as those without internet service or those who have no online security, backup, or tech support.
# The plots highlight significant contrasts in service usage between customers who churn and those who remain.

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.xticks(rotation=45)
plt.title("Churned Customers By Payment Method")
plt.show()

#Customers are likely to Churn when they are using electronic check as a payment method
```