```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.22.4 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
abhishek upadhyay\appdata\local\programs\python\python310\lib\site-
packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from pandas) (2025.1)
Requirement already satisfied: six>=1.5 in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (2.2.3)
Requirement already satisfied: packaging>=20.0 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib) (3.2.1)
```

```
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
abhishek upadhyay\appdata\local\programs\python\python310\lib\site-
packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\
abhishek upadhyay\appdata\local\programs\python\python310\lib\site-
packages (from seaborn) (2.2.3)
Requirement already satisfied: pandas>=1.2 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\
abhishek upadhyay\appdata\local\programs\python\python310\lib\site-
packages (from seaborn) (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
abhishek upadhyay\appdata\local\programs\python\python310\lib\site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
(from pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\abhishek
upadhyay\appdata\local\programs\python\python310\lib\site-packages
```

```
(from pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: six>=1.5 in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\abhishek upadhyay\
appdata\local\programs\python\python310\lib\site-packages (2.2.3)
Note: you may need to restart the kernel to use updated packages.
```

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

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

```
      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-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   

     PhoneService     MultipleLines InternetService
OnlineSecurity  ...  \
0              No  No phone service             DSL
No  ...   
1             Yes                No             DSL
Yes  ...   
2             Yes                No             DSL
Yes  ...   
3              No  No phone service             DSL
Yes  ...   
4             Yes                No     Fiber optic
No  ...   
...           ...               ...             ...             ...  ..
.   
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  \
0                  No          No          No              No  Month-
to-month
1                 Yes          No          No              No
One year
2                  No          No          No              No  Month-
to-month
3                 Yes         Yes          No              No
One year
4                  No          No          No              No  Month-
to-month
...               ...         ...         ...             ...
...
7038              Yes         Yes         Yes             Yes
One year
7039              Yes          No         Yes             Yes
One year
7040               No          No          No              No  Month-
to-month
7041               No          No          No              No  Month-
to-month
7042              Yes         Yes         Yes             Yes
Two year

     PaperlessBilling              PaymentMethod MonthlyCharges
TotalCharges  \
0                 Yes           Electronic check          29.85
29.85
1                  No               Mailed check          56.95
1889.5
2                 Yes               Mailed check          53.85
108.15
3                  No  Bank transfer (automatic)          42.30
1840.75
4                 Yes           Electronic check          70.70
151.65
...               ...                        ...            ...
...
7038              Yes               Mailed check          84.80
1990.5
```

```
7039              Yes    Credit card (automatic)         103.20
7362.9
7040              Yes           Electronic check          29.60
346.45
7041              Yes               Mailed check          74.40
306.6
7042              Yes  Bank transfer (automatic)         105.65
6844.5

     Churn
0       No
1       No
2      Yes
3       No
4      Yes
...    ...
7038    No
7039    No
7040    No
7041   Yes
7042    No

[7043 rows x 21 columns]
```

```
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 blank with 0 as tenure is 0 and no charges are recorded

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

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()

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

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

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

#converted 0 and 1 values of the senior citizen to yes/No to make easy to understand

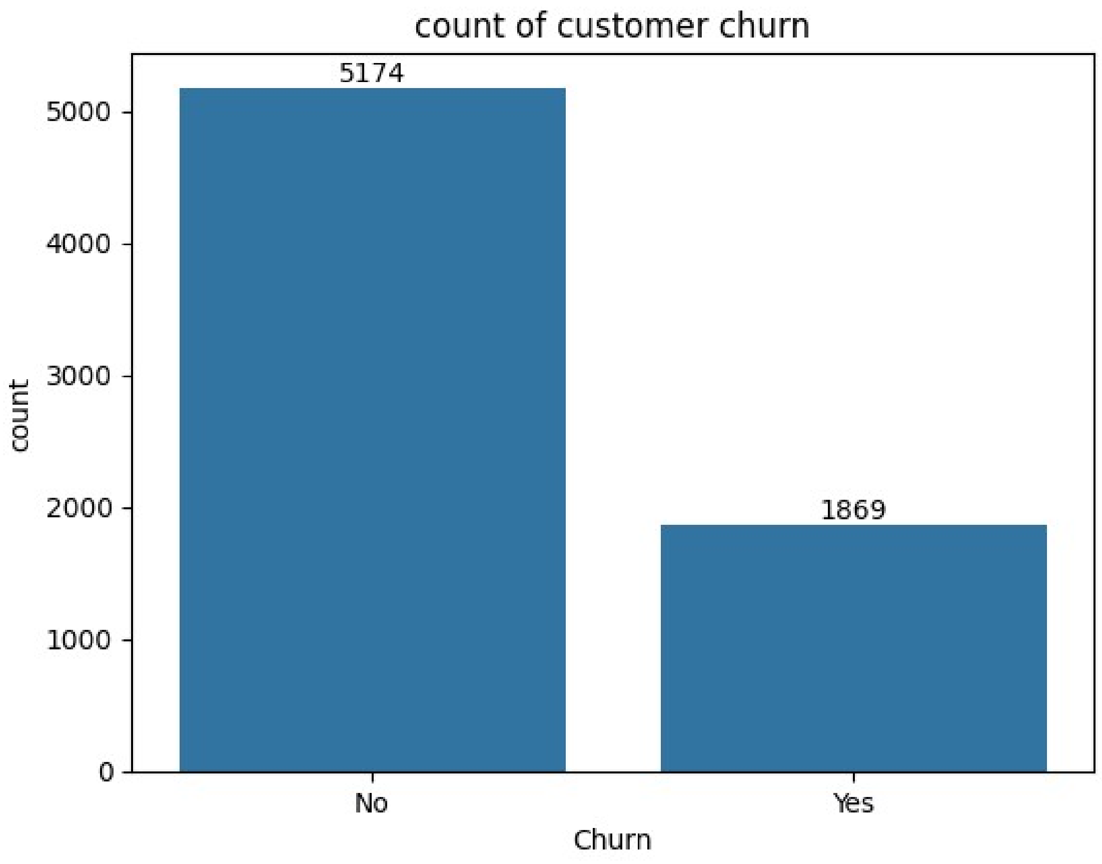
```
df.head(10)
```

```
   customerID  gender SeniorCitizen Partner Dependents  tenure 
PhoneService  \
0  7590-VHVEG  Female            No     Yes         No       1 
No   
1  5575-GNVDE    Male            No      No         No      34 
Yes   
2  3668-QPYBK    Male            No      No         No       2 
Yes   
3  7795-CFOCW    Male            No      No         No      45 
No   
4  9237-HQITU  Female            No      No         No       2 
Yes   
5  9305-CDSKC  Female            No      No         No       8 
Yes   
6  1452-KIOVK    Male            No      No        Yes      22 
Yes   
7  6713-OKOMC  Female            No      No         No      10 
No   
8  7892-POOKP  Female            No     Yes         No      28 
Yes   
9  6388-TABGU    Male            No      No        Yes      62 
Yes   

      MultipleLines InternetService OnlineSecurity  ... 
DeviceProtection  \
0  No phone service             DSL             No  ...
No   
1                No             DSL            Yes  ...
```

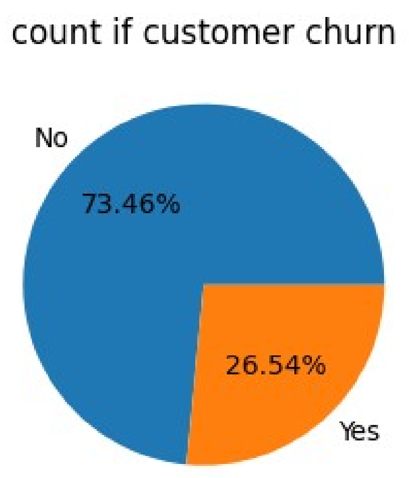
```
Yes
2                No              DSL             Yes  ...
No
3  No phone service              DSL             Yes  ...
Yes
4                No      Fiber optic              No  ...
No
5               Yes      Fiber optic              No  ...
Yes
6               Yes      Fiber optic              No  ...
No
7  No phone service              DSL             Yes  ...
No
8               Yes      Fiber optic              No  ...
Yes
9                No              DSL             Yes  ...
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
5          No         Yes             Yes  Month-to-month
Yes
6          No         Yes              No  Month-to-month
Yes
7          No          No              No  Month-to-month
No
8         Yes         Yes             Yes  Month-to-month
Yes
9          No          No              No        One year
No

               PaymentMethod MonthlyCharges  TotalCharges  Churn
0           Electronic check          29.85         29.85     No
1               Mailed check          56.95       1889.50     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           Electronic check          99.65        820.50    Yes
6    Credit card (automatic)          89.10       1949.40     No
```

```
7                 Mailed check          29.75         301.90     No
8             Electronic check         104.80        3046.05    Yes
9    Bank transfer (automatic)          56.15        3487.95     No

[10 rows x 21 columns]
```

```
ax = sns.countplot(x = 'Churn', data = df)
plt.title('count of customer churn')
ax.bar_label(ax.containers[0])
plt.show()
```



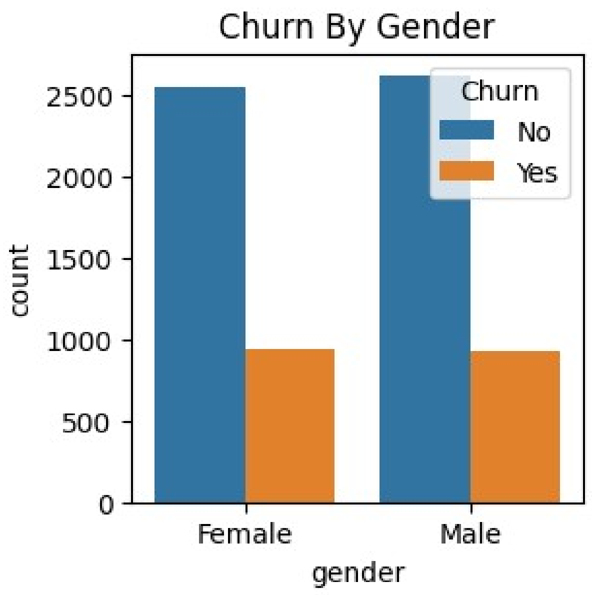
```
plt.figure(figsize = (3,4))
plt.title("count if customer churn")
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
plt.show()
```



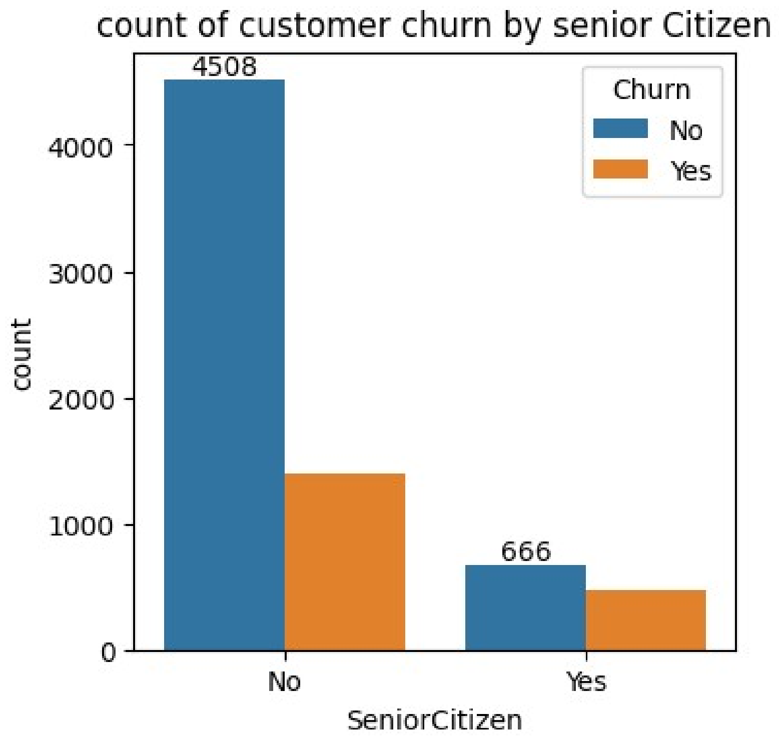
#from the given pie chart we can conclude that 26.54% of our customers have churned out.

#now let's explore the reason behind it

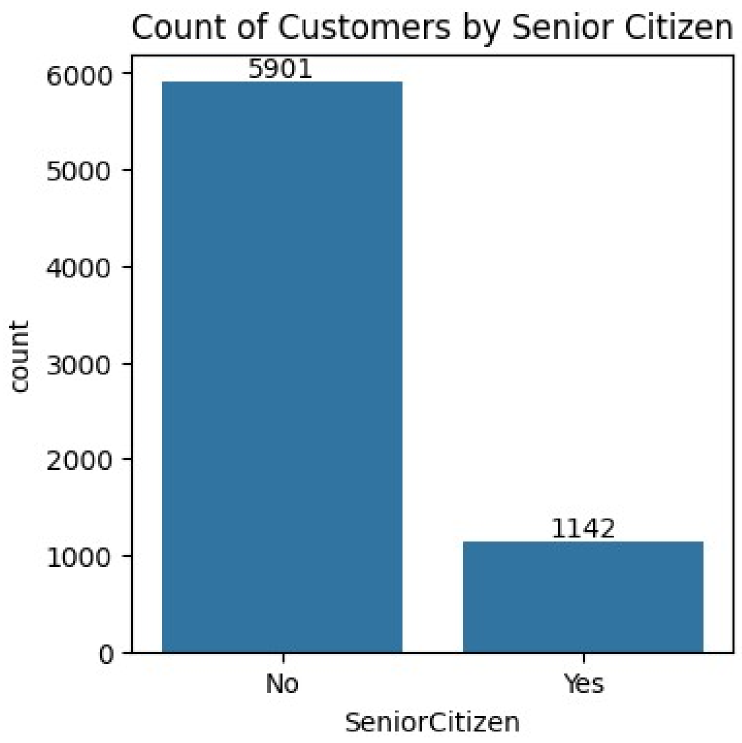
```
plt.figure(figsize = (3,3))
sns.countplot(x = "gender", data = df, hue = "Churn")
plt.title("Churn By Gender")
plt.show()
```



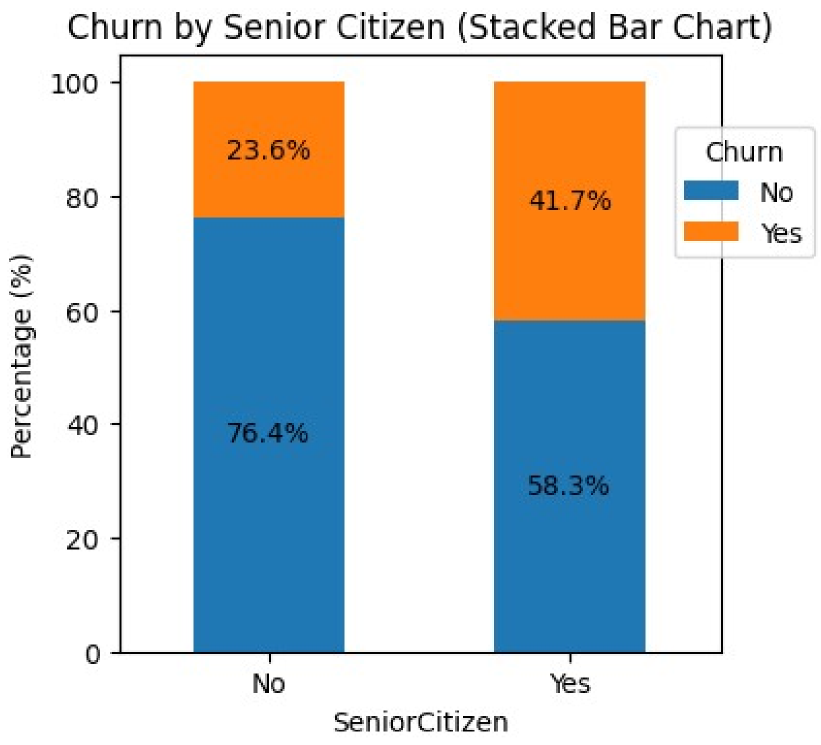
```
plt.figure(figsize = (4,4))
ax = sns.countplot(x = 'SeniorCitizen', data = df, hue = "Churn")
plt.title('count of customer churn by senior Citizen')
ax.bar_label(ax.containers[0])
plt.show()
```



```
plt.figure(figsize = (4,4))
ax = sns.countplot(x = "SeniorCitizen", data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Senior Citizen")
plt.show()
```

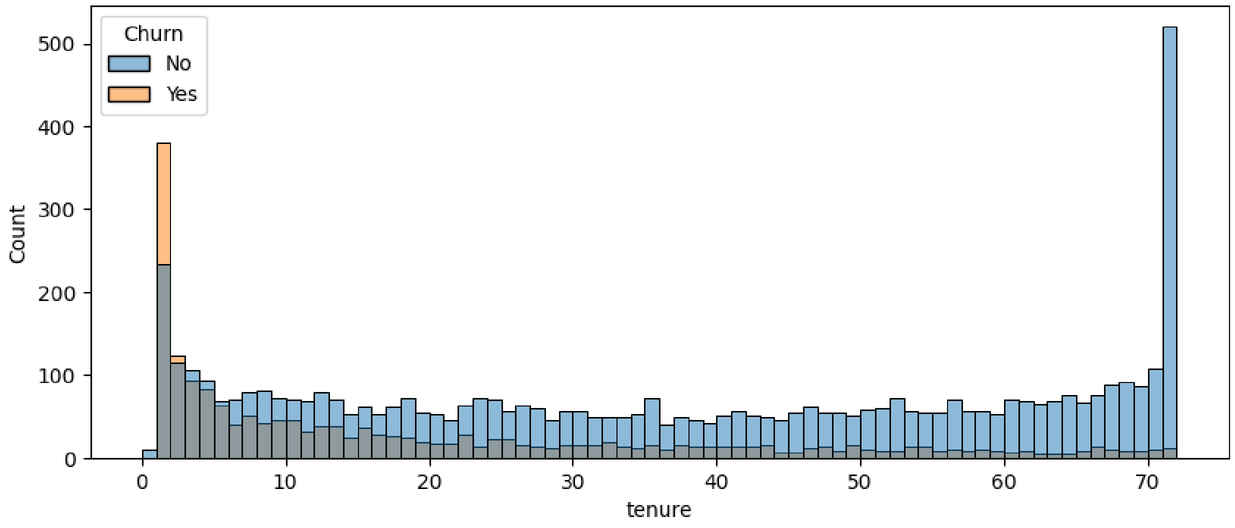


```
total_counts = df.groupby('SeniorCitizen')
['Churn'].value_counts(normalize=True).unstack() * 100

# Plot
fig, ax = plt.subplots(figsize=(4, 4))  # Adjust figsize for better
visualization

# Plot the bars
total_counts.plot(kind='bar', stacked=True, ax=ax, color=['#1f77b4',
'#ff7f0e'])  # Customize colors if desired

# Add percentage labels on the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x + width / 2, y + height / 2, f'{height:.1f}%',
ha='center', va='center')

plt.title('Churn by Senior Citizen (Stacked Bar Chart)')
plt.xlabel('SeniorCitizen')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=0)
plt.legend(title='Churn', bbox_to_anchor = (0.9,0.9))  # Customize
legend location

plt.show()
```



#comparative a greater pecentage of people in senior citizen category have churned

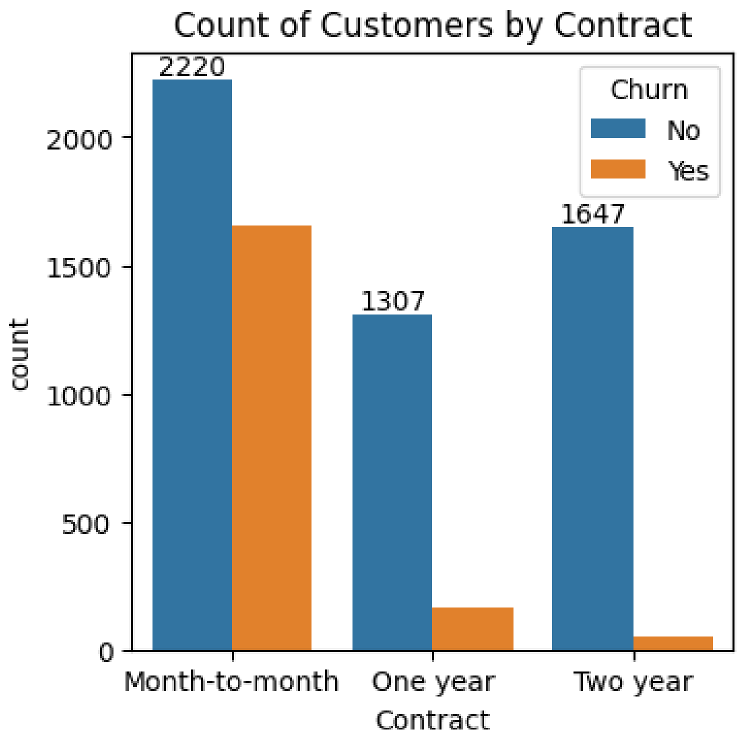
```
plt.figure(figsize = (10,4))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()
```



#people who have used our services for a long time have stayed and people who have used our sevices

#1 or 2 months have churned

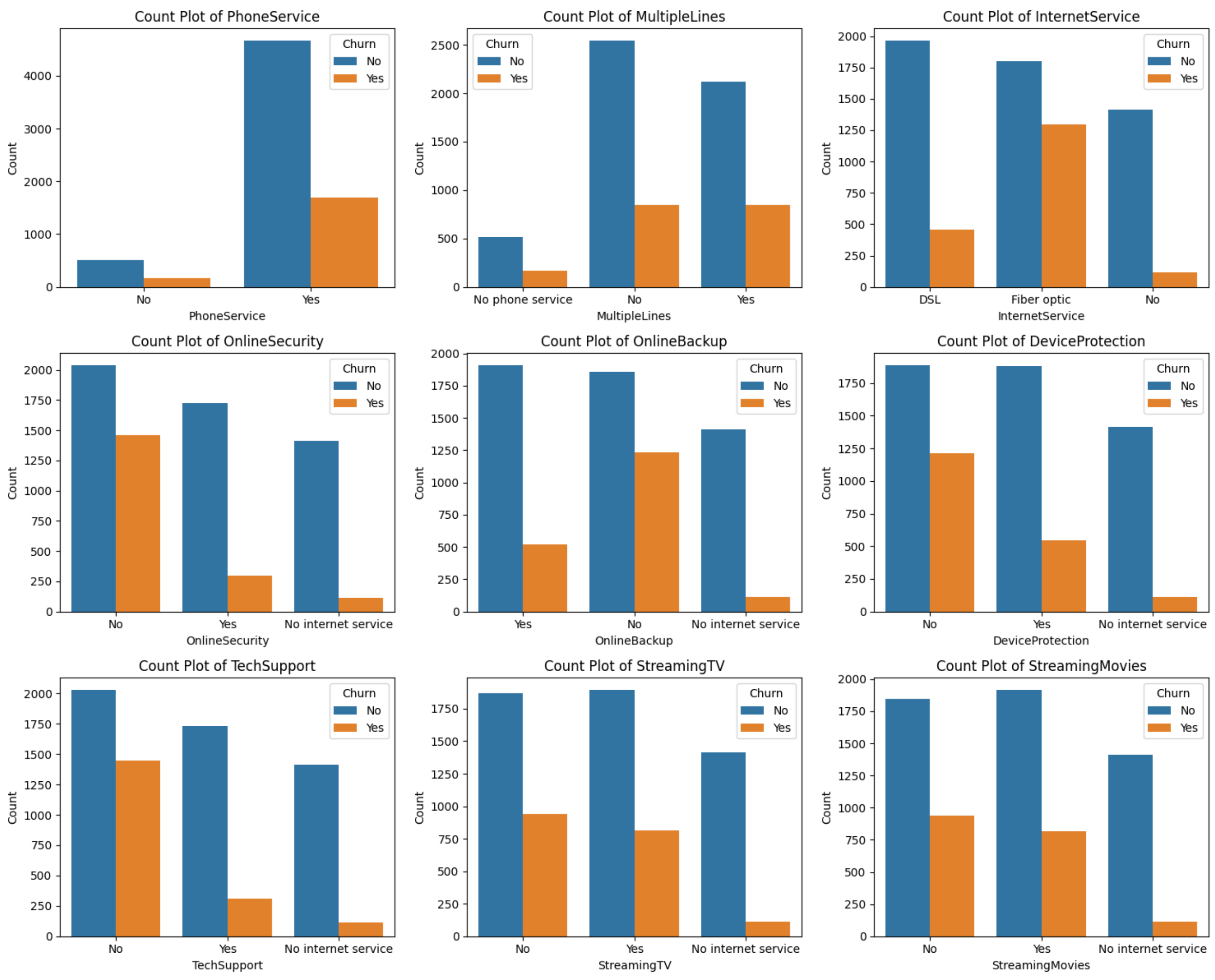
```
plt.figure(figsize = (4,4))
ax = sns.countplot(x = "Contract", data = df, hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract")
plt.show()
```



#people who have month to month contract are likely to churn then from those who have 1 or 2 years of contract

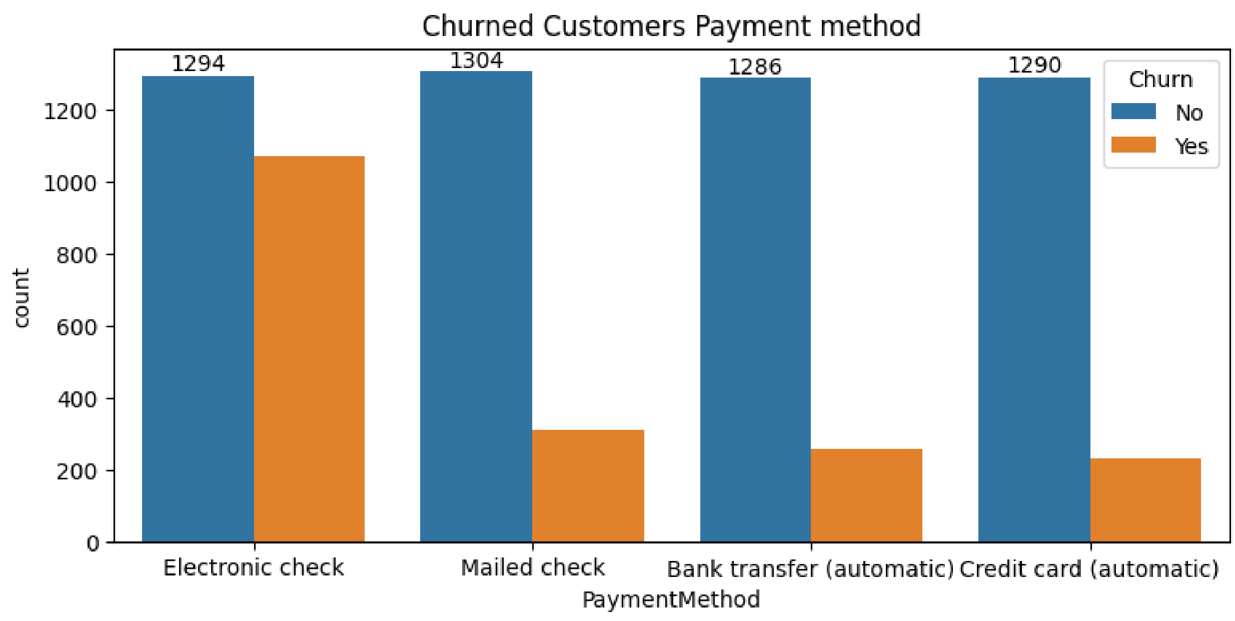
```
df.columns.values

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

columns = ['PhoneService', 'MultipleLines', 'InternetService',
'OnlineSecurity',
           'OnlineBackup', 'DeviceProtection', 'TechSupport',
'StreamingTV', 'StreamingMovies']

# Number of columns for the subplot grid (you can change this)
n_cols = 3
```

```
n_rows = (len(columns) + n_cols - 1) // n_cols  # Calculate number of
rows needed

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

# Flatten the axes array for easy iteration (handles both 1D and 2D
arrays)
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 = df["Churn"])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Count')

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

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



#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.

```
plt.figure(figsize = (9,4))
ax = sns.countplot(x = "PaymentMethod", data = df, hue = 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Churned Customers Payment method")
plt.show()
```



#customer is likely to churn when he is using electronic check as a payment method.