```
from google.colab import files
files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Customer Churn.csv to Customer Churn.csv
```

```
{"variable_name":"df","type":"dataframe"}
```

```
import pandas as pd

df = pd.read_csv("Customer Churn.csv")
df.head()
```

```
{"variable_name":"df","type":"dataframe"}
```

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

```
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 zero as tenure is 0 and no totalchare 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   object
 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(1), object(18)
memory usage: 1.1+ MB
```

```
df.isnull().sum().sum()
```

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

```
df.describe()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n    {\n      \"column\'
```

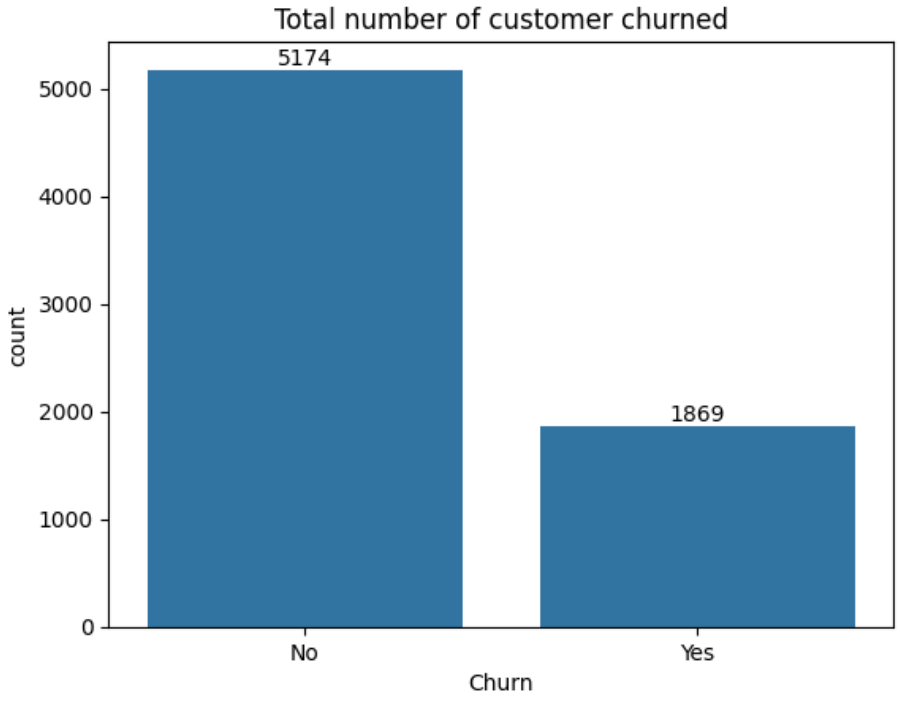
```
df['customerID'].duplicated().sum()
```

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

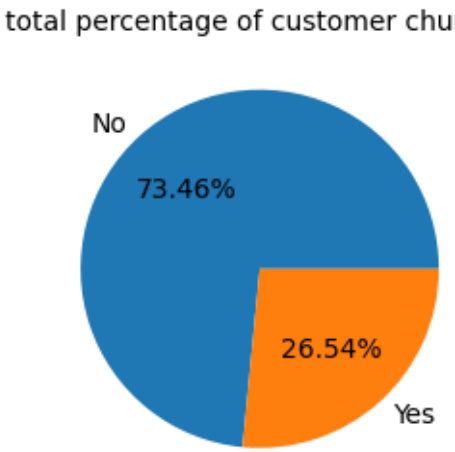
Converted 0 and 1 value of column 'senoirCtizen' to No/Yes to make easier to understand

```
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])
plt.title("Total number of customer churned")
plt.show()
```

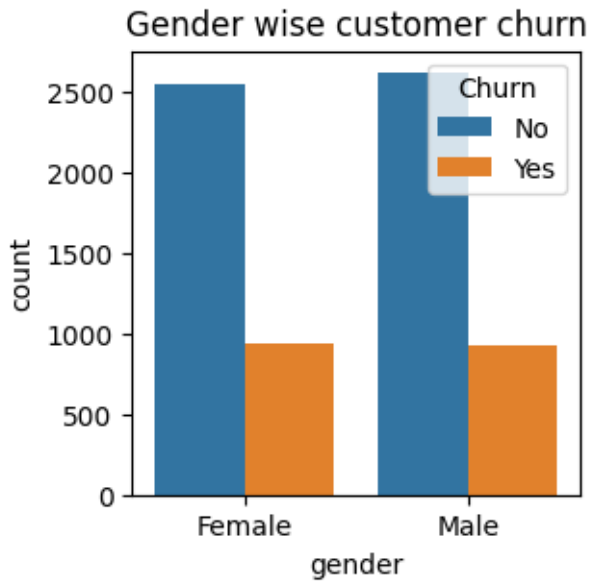



```
plt.figure(figsize=(3,4))
gb=df.groupby('Churn').agg({'Churn':'count'})
plt.pie(gb['Churn'],labels=gb.index,autopct='%1.2f%%')
plt.title("total percentage of customer churned",  fontsize=10)
plt.show()
```

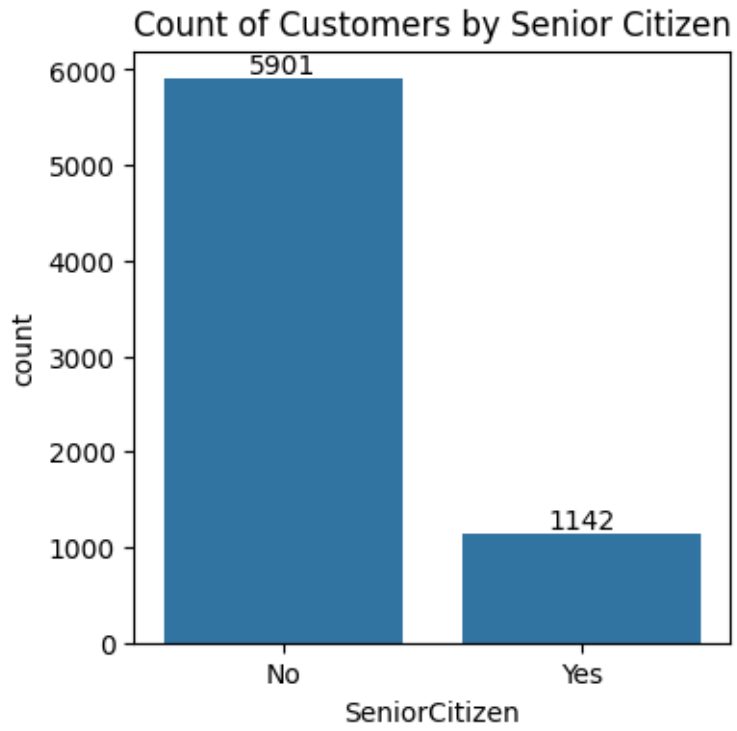


From the given pie chart we conclude that 26.54 percent cstomer are churned out now let explore the reason behond it

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

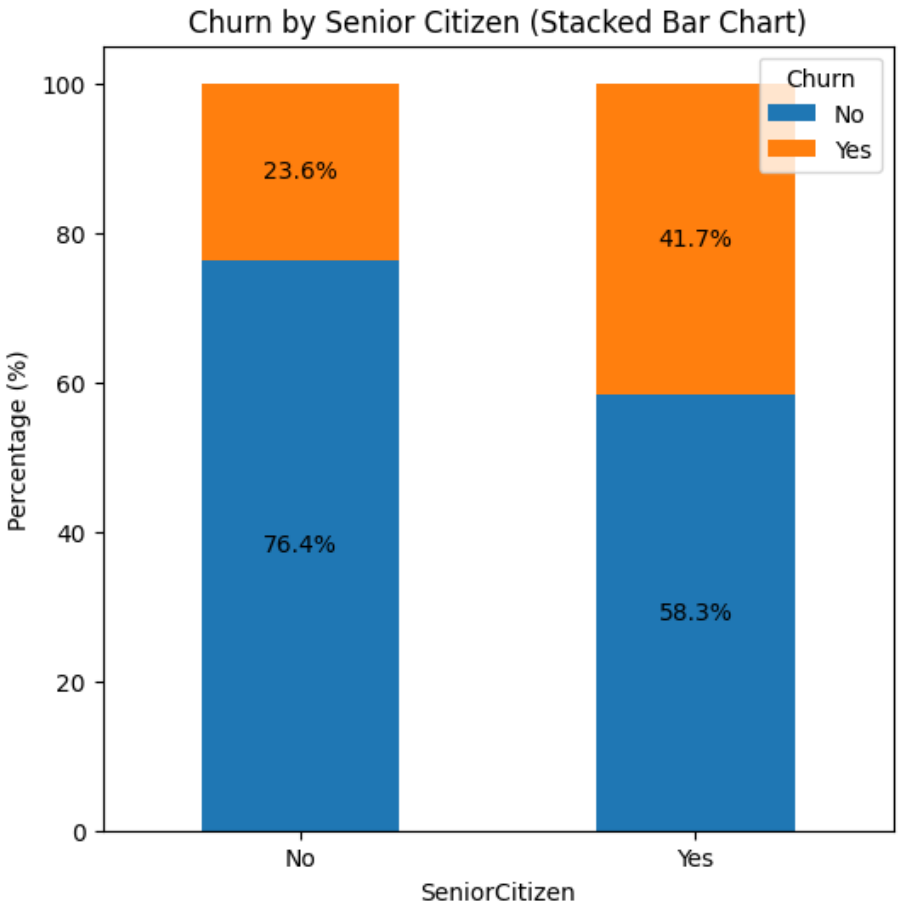


```
plt.figure(figsize=(4, 4))
ax = sns.countplot(x="SeniorCitizen", data=df)

ax.bar_label(ax.containers[0])

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



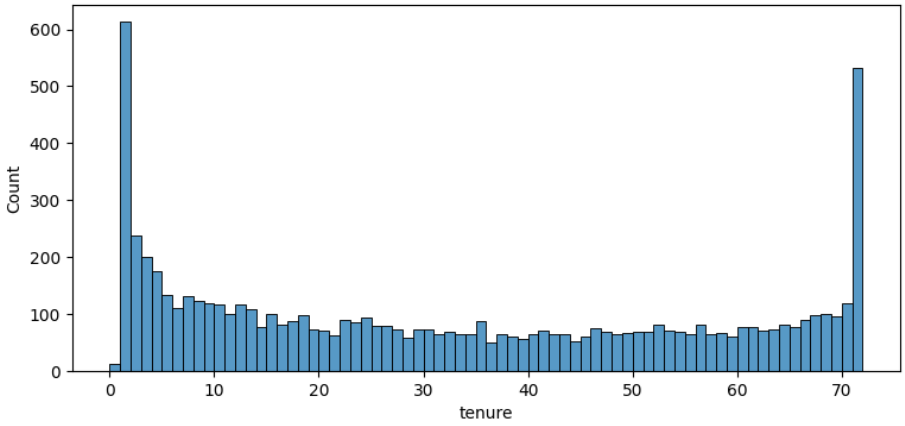

```
total_counts = df.groupby('SeniorCitizen')['Churn'].value_counts(normalize=True).unstack() *

# Plot
fig, ax = plt.subplots(figsize=(6, 6))  # Adjust figsize for better visualization

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

# 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', loc='upper right')  # Customize legend location
plt.show()
```

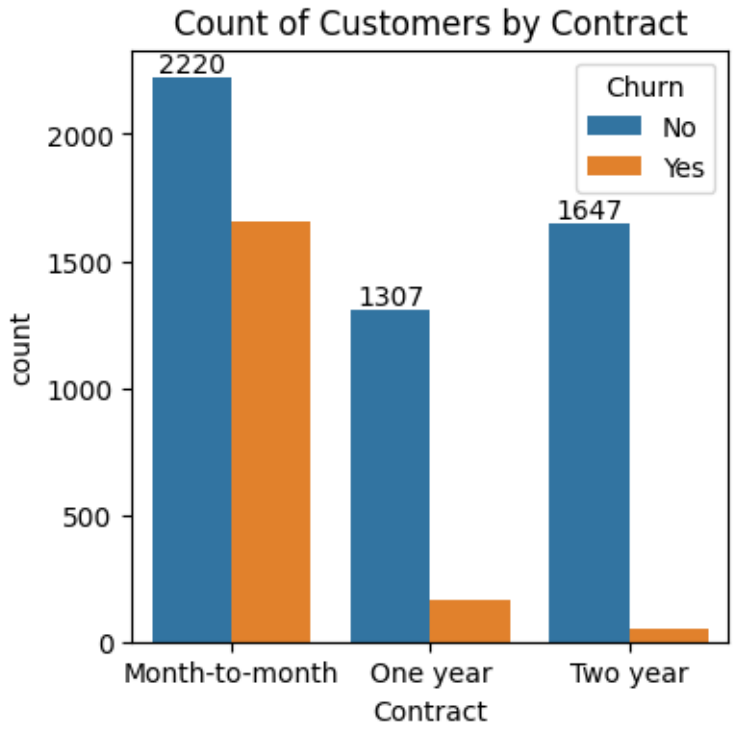


Comparitive a greater percentage of senoir citizen are churned

```
plt.figure(figsize=(9,4))
sns.histplot(x='tenure',data=df,bins=72)
plt.show()
```




```
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()
```

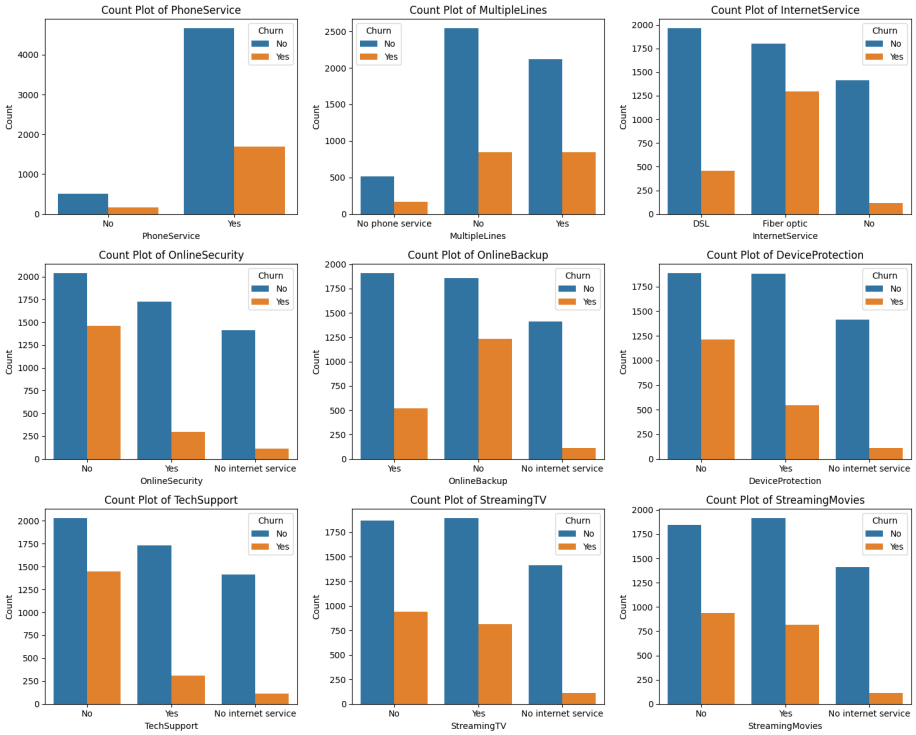


People who have month to month contract are likely to churn from those who have 1 or 2 years contract

```
df.columns.values
```

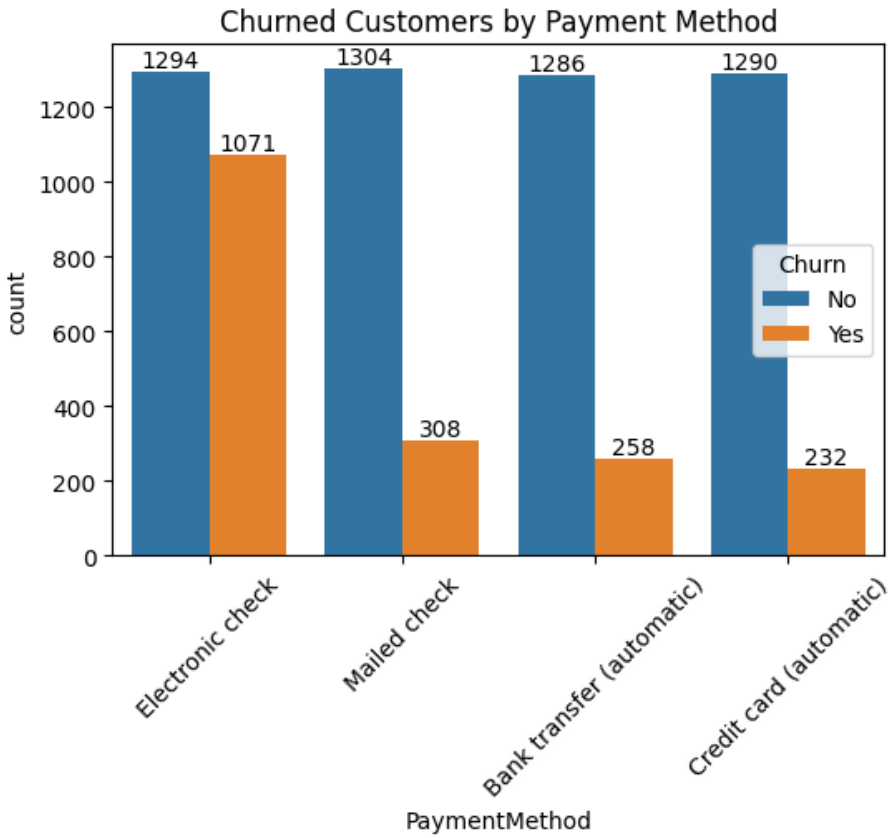
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling',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

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

# Number of columns for the subplot grid
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))

# 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=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()
```



####churn is disproportionately higher among users who have no add-on services such as OnlineSecurity, OnlineBackup, DeviceProtection, and TechSupport. Customers with bundled digital services show materially lower churn rates, indicating stronger engagement and switching costs. InternetService type also demonstrates churn variance, with fiber-optic users contributing a larger share of churn compared to DSL. This pattern suggests that service bundling and post-sale enablement are key levers for churn reduction.

```
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.title("Churned Customers by Payment Method")
plt.xticks(rotation=45)++
plt.show()
```


Churned Customers by Payment Method
1294
1304
1286
1290
1071
308
258
232
1200
1000
800
600
400
200
0
count
Churn
No
Yes
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod