

eb83zxhso

December 8, 2024

```
[11]: pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\asus\anaconda3\lib\site-  
packages (2.1.4)  
Requirement already satisfied: numpy<2,>=1.23.2 in  
c:\users\asus\anaconda3\lib\site-packages (from pandas) (1.26.4)  
Requirement already satisfied: python-dateutil>=2.8.2 in  
c:\users\asus\anaconda3\lib\site-packages (from pandas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\anaconda3\lib\site-  
packages (from pandas) (2023.3.post1)  
Requirement already satisfied: tzdata>=2022.1 in  
c:\users\asus\anaconda3\lib\site-packages (from pandas) (2023.3)  
Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\lib\site-  
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)  
Note: you may need to restart the kernel to use updated packages.
```

```
[8]: pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\asus\anaconda3\lib\site-  
packages (0.1.9)  
Requirement already satisfied: pyloco>=0.0.134 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib) (0.0.139)  
Requirement already satisfied: matplotlib>=3.1.1 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib) (3.8.0)  
Requirement already satisfied: contourpy>=1.0.1 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)  
(1.2.0)  
Requirement already satisfied: cycycler>=0.10 in c:\users\asus\anaconda3\lib\site-  
packages (from matplotlib>=3.1.1->matplotlib) (0.11.0)  
Requirement already satisfied: fonttools>=4.22.0 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)  
(4.25.0)  
Requirement already satisfied: kiwisolver>=1.0.1 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)  
(1.4.4)  
Requirement already satisfied: numpy<2,>=1.21 in  
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)  
(1.26.4)
```

Requirement already satisfied: packaging>=20.0 in
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)
(23.1)

Requirement already satisfied: pillow>=6.2.0 in
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)
(10.2.0)

Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)
(3.0.9)

Requirement already satisfied: python-dateutil>=2.7 in
c:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.1.1->matplotlib)
(2.8.2)

Requirement already satisfied: ushlex in c:\users\asus\anaconda3\lib\site-
packages (from pyloco>=0.0.134->matplotlib) (0.99.1)

Requirement already satisfied: websocket-client in
c:\users\asus\anaconda3\lib\site-packages (from pyloco>=0.0.134->matplotlib)
(0.58.0)

Requirement already satisfied: twine in c:\users\asus\anaconda3\lib\site-
packages (from pyloco>=0.0.134->matplotlib) (6.0.1)

Requirement already satisfied: typing in c:\users\asus\anaconda3\lib\site-
packages (from pyloco>=0.0.134->matplotlib) (3.7.4.3)

Requirement already satisfied: SimpleWebSocketServer in
c:\users\asus\anaconda3\lib\site-packages (from pyloco>=0.0.134->matplotlib)
(0.1.2)

Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.1.1->matplotlib) (1.16.0)

Requirement already satisfied: pkginfo>=1.8.1 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(1.9.6)

Requirement already satisfied: readme-renderer>=35.0 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(44.0)

Requirement already satisfied: requests>=2.20 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(2.31.0)

Requirement already satisfied: requests-toolbelt!=0.9.0,>=0.8.0 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(1.0.0)

Requirement already satisfied: urllib3>=1.26.0 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(2.0.7)

Requirement already satisfied: keyring>=15.1 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(23.13.1)

Requirement already satisfied: rfc3986>=1.4.0 in
c:\users\asus\anaconda3\lib\site-packages (from twine->pyloco>=0.0.134->matplotlib)
(2.0.0)

Requirement already satisfied: rich>=12.0.0 in c:\users\asus\anaconda3\lib\site-

```

packages (from twine->pyloco>=0.0.134->matplotlib) (13.3.5)
Requirement already satisfied: jaraco.classes in
c:\users\asus\anaconda3\lib\site-packages (from
keyring>=15.1->twine->pyloco>=0.0.134->matplotlib) (3.2.1)
Requirement already satisfied: importlib-metadata>=4.11.4 in
c:\users\asus\anaconda3\lib\site-packages (from
keyring>=15.1->twine->pyloco>=0.0.134->matplotlib) (7.0.1)
Requirement already satisfied: pywin32-ctypes>=0.2.0 in
c:\users\asus\anaconda3\lib\site-packages (from
keyring>=15.1->twine->pyloco>=0.0.134->matplotlib) (0.2.0)
Requirement already satisfied: nh3>=0.2.14 in c:\users\asus\anaconda3\lib\site-
packages (from readme-renderer>=35.0->twine->pyloco>=0.0.134->matplotlib) (0.2.19)
Requirement already satisfied: docutils>=0.21.2 in
c:\users\asus\anaconda3\lib\site-packages (from readme-
renderer>=35.0->twine->pyloco>=0.0.134->matplotlib) (0.21.2)
Requirement already satisfied: Pygments>=2.5.1 in
c:\users\asus\anaconda3\lib\site-packages (from readme-
renderer>=35.0->twine->pyloco>=0.0.134->matplotlib) (2.15.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\asus\anaconda3\lib\site-packages (from
requests>=2.20->twine->pyloco>=0.0.134->matplotlib) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\asus\anaconda3\lib\site-
packages (from requests>=2.20->twine->pyloco>=0.0.134->matplotlib) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\asus\anaconda3\lib\site-packages (from
requests>=2.20->twine->pyloco>=0.0.134->matplotlib) (2024.2.2)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in
c:\users\asus\anaconda3\lib\site-packages (from
rich>=12.0.0->twine->pyloco>=0.0.134->matplotlib) (2.2.0)
Requirement already satisfied: zipp>=0.5 in c:\users\asus\anaconda3\lib\site-
packages (from importlib-
metadata>=4.11.4->keyring>=15.1->twine->pyloco>=0.0.134->matplotlib) (3.17.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\asus\anaconda3\lib\site-
packages (from markdown-it-
py<3.0.0,>=2.2.0->rich>=12.0.0->twine->pyloco>=0.0.134->matplotlib) (0.1.0)
Requirement already satisfied: more-itertools in
c:\users\asus\anaconda3\lib\site-packages (from
jaraco.classes->keyring>=15.1->twine->pyloco>=0.0.134->matplotlib) (10.1.0)
Note: you may need to restart the kernel to use updated packages.

```

```
[12]: pip install seaborn
```

```

Requirement already satisfied: seaborn in c:\users\asus\anaconda3\lib\site-
packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
c:\users\asus\anaconda3\lib\site-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=0.25 in c:\users\asus\anaconda3\lib\site-
packages (from seaborn) (2.1.4)

```

Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\asus\anaconda3\lib\site-packages (from seaborn) (3.8.0)
 Requirement already satisfied: contourpy>=1.0.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.2.0)
 Requirement already satisfied: cycycler>=0.10 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
 Requirement already satisfied: fonttools>=4.22.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.25.0)
 Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
 Requirement already satisfied: packaging>=20.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
 Requirement already satisfied: pillow>=6.2.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (10.2.0)
 Requirement already satisfied: pyparsing>=2.3.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
 Requirement already satisfied: python-dateutil>=2.7 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
 Requirement already satisfied: pytz>=2020.1 in c:\users\asus\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3.post1)
 Requirement already satisfied: tzdata>=2022.1 in c:\users\asus\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)
 Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
 Note: you may need to restart the kernel to use updated packages.

```
[13]: pip install numpy
```

Requirement already satisfied: numpy in c:\users\asus\anaconda3\lib\site-packages (1.26.4)
 Note: you may need to restart the kernel to use updated packages.

```
[14]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

#Reading csv file

```
[95]: df = pd.read_csv(r'C:\Users\Asus\Downloads\archive\Customer Churn.csv')
df
```

[95]:

	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]

#To inspect the data

[96]: `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 blanks with 0 as tenure in 0 and no total charges are recorded

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

```
[98]: 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

```

#To check for null values

```
[99]: df.isnull().sum()
```

```

[99]: customerID      0
      gender          0
      SeniorCitizen   0
      Partner         0

```

```

Dependents      0
tenure          0
PhoneService    0
MultipleLines    0
InternetService  0
OnlineSecurity   0
OnlineBackup     0
DeviceProtection 0
TechSupport      0
StreamingTV      0
StreamingMovies  0
Contract         0
PaperlessBilling 0
PaymentMethod    0
MonthlyCharges   0
TotalCharges     0
Churn            0
dtype: int64

```

```
[100]: df.isnull().sum().sum()
```

```
[100]: 0
```

#To get descriptive analysis

```
[101]: df.describe()
```

```
[101]:
```

	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

#To check duplicate records

```
[102]: df.duplicated().sum()
```

```
[102]: 0
```

```
[103]: df["customerID"].duplicated().sum()
```

```
[103]: 0
```

#Converting 0 and 1 value of senior citizen to yes/no values for better understanding


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

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

```
[104]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	\
0	7590-VHVEG	Female	no	Yes	No	1	
1	5575-GNVDE	Male	no	No	No	34	
2	3668-QPYBK	Male	no	No	No	2	
3	7795-CFOCW	Male	no	No	No	45	
4	9237-HQITU	Female	no	No	No	2	
...	
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	...	\
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	

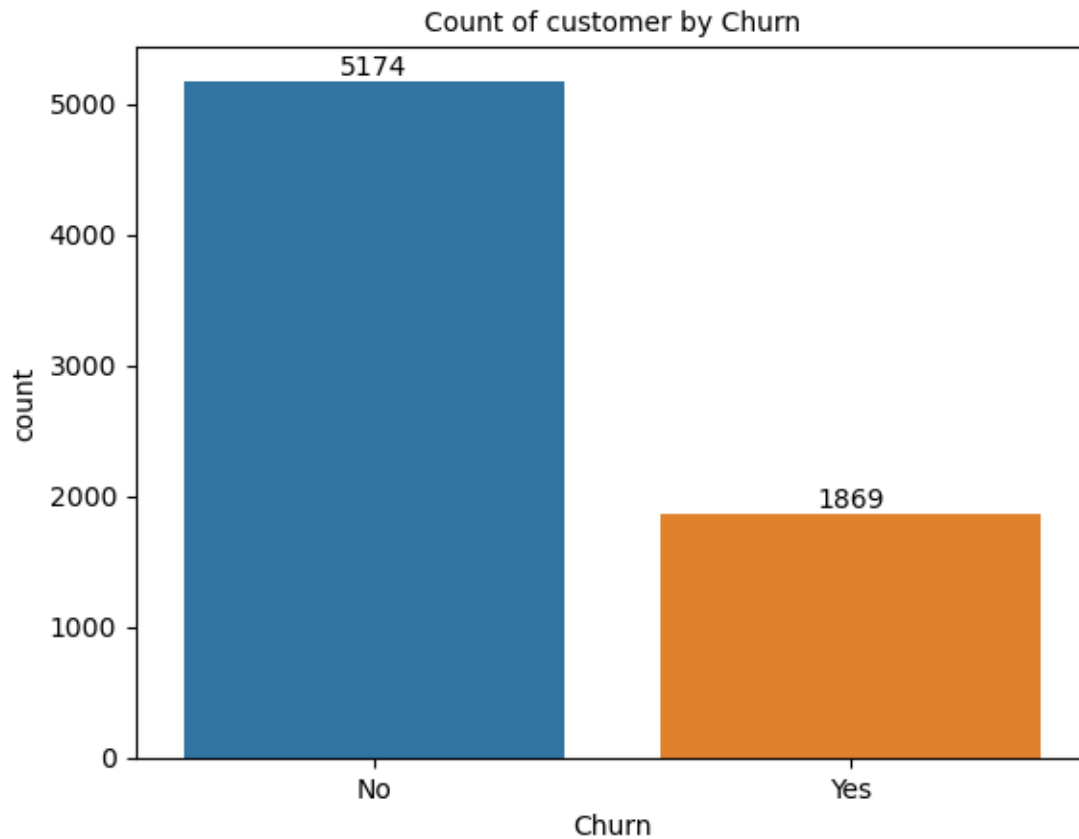
	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	\
0	Yes	Electronic check	29.85	29.85	
1	No	Mailed check	56.95	1889.50	
2	Yes	Mailed check	53.85	108.15	
3	No	Bank transfer (automatic)	42.30	1840.75	
4	Yes	Electronic check	70.70	151.65	
...	
7038	Yes	Mailed check	84.80	1990.50	
7039	Yes	Credit 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
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]

#Plotting count and percentage of Customer Churned out

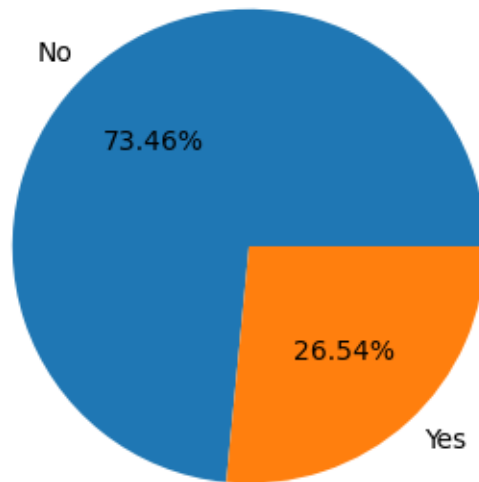
```
[58]: ax = sns.countplot(x = 'Churn', data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of customer by Churn", fontsize = 10)
plt.show()
```



#From the given bar graph we can conclude that 1869 customers have Churned out.

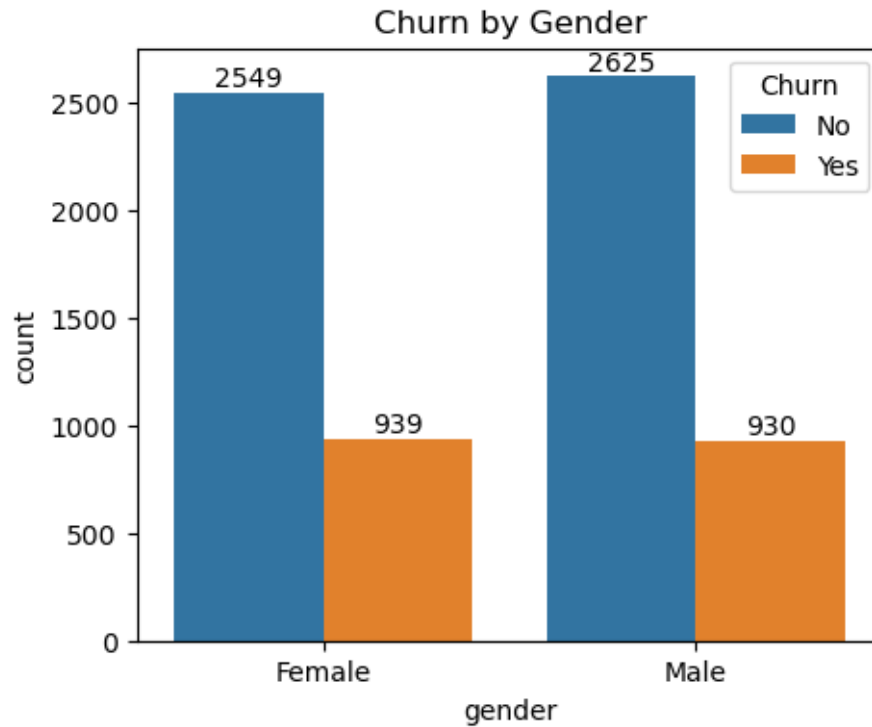
```
[57]: plt.figure(figsize = (4,4))
      gb = df.groupby('Churn').agg({'Churn':"count"})
      plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
      plt.title("Percentage of Churned Customer", fontsize = 10)
      plt.show()
```

Percentage of Churned Customer



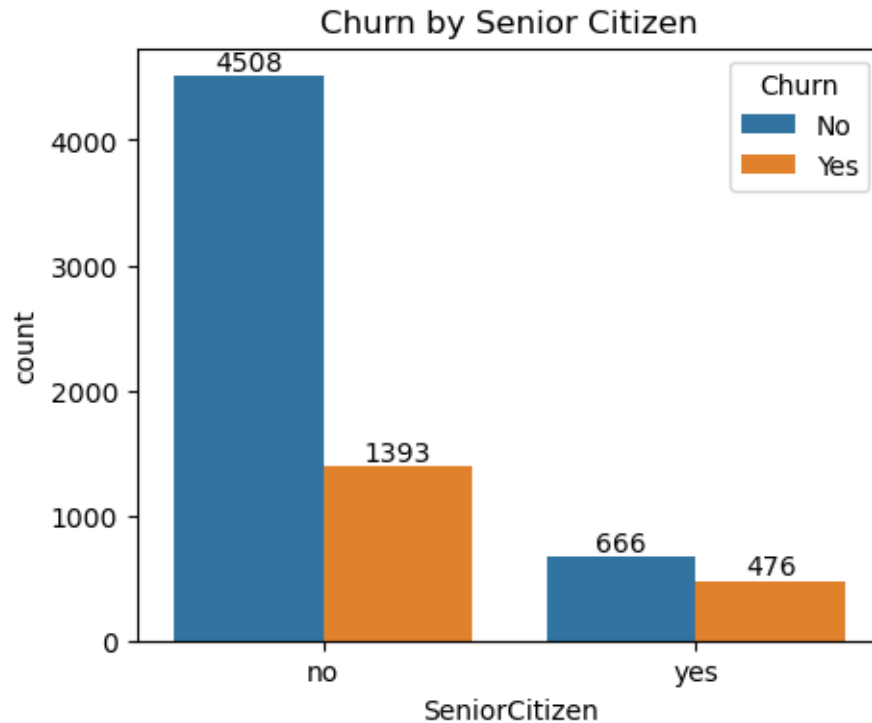
#From the given pie chart we can conclude that 26.54% of our customer have churned out.

```
[70]: plt.figure(figsize = (5,4))
      ax = sns.countplot(x = "gender", data = df, hue = "Churn")
      ax.bar_label(ax.containers[1])
      ax.bar_label(ax.containers[0])
      plt.title("Churn by Gender")
      plt.show()
```



#From the given bar graph we can conclude that the churned customer are not gender specific and both are almost equal

```
[108]: plt.figure(figsize=(5, 4))
ax = sns.countplot(x = "SeniorCitizen", data = df, hue = "Churn")
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[0])
plt.title("Churn by Senior Citizen")
plt.show()
```



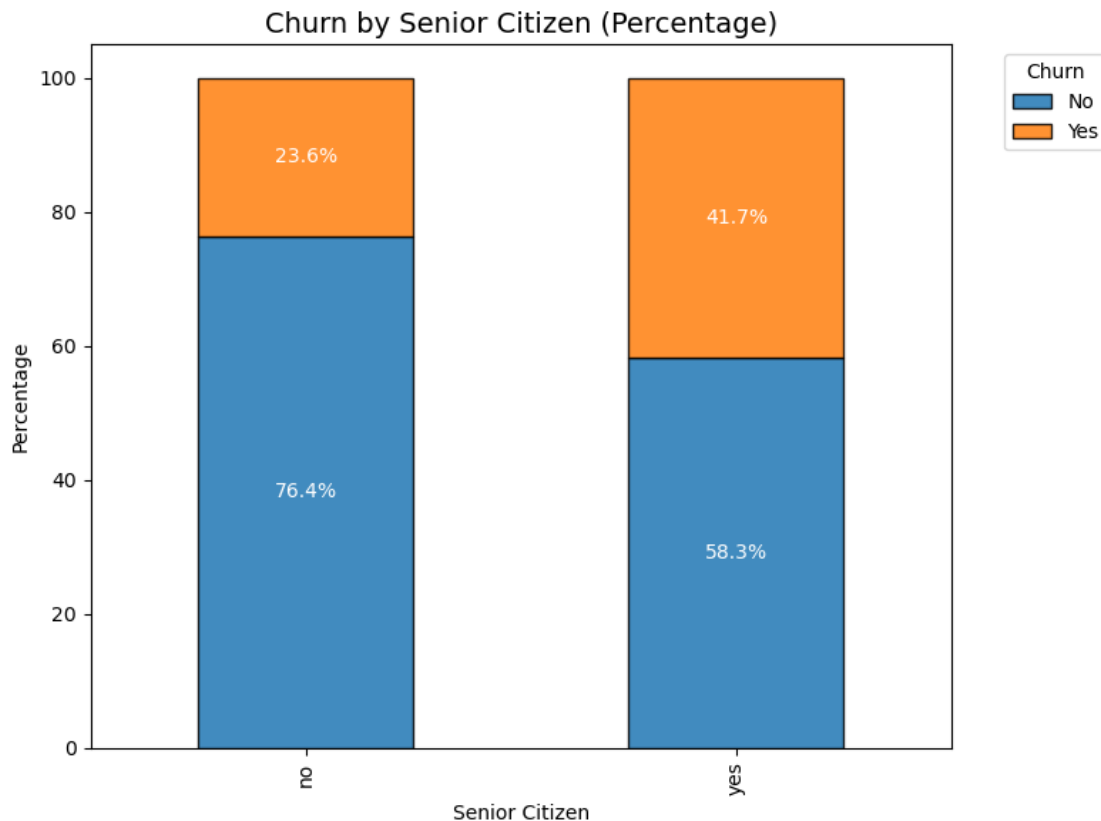
```
[110]: data_percent = df.groupby(["SeniorCitizen", "Churn"]).size().
        ↪reset_index(name="count")
total_counts = data_percent.groupby("SeniorCitizen")["count"].transform("sum")
data_percent["percentage"] = (data_percent["count"] / total_counts) * 100

# Step 2: Pivot the data for stacking
data_pivot = data_percent.pivot(index="SeniorCitizen", columns="Churn",
        ↪values="percentage").fillna(0)

# Step 3: Plot the stacked bar chart
ax = data_pivot.plot(
    kind="bar",
    stacked=True,
    figsize=(8, 6),
    color=["#1f77b4", "#ff7f0e"], # Colors for segments
    edgecolor="black",
    alpha=0.85,
)

# Step 4: Add percentage labels
for container in ax.containers:
    ax.bar_label(container, fmt="%.1f%%", label_type="center", fontsize=10,
        ↪color="white")
```

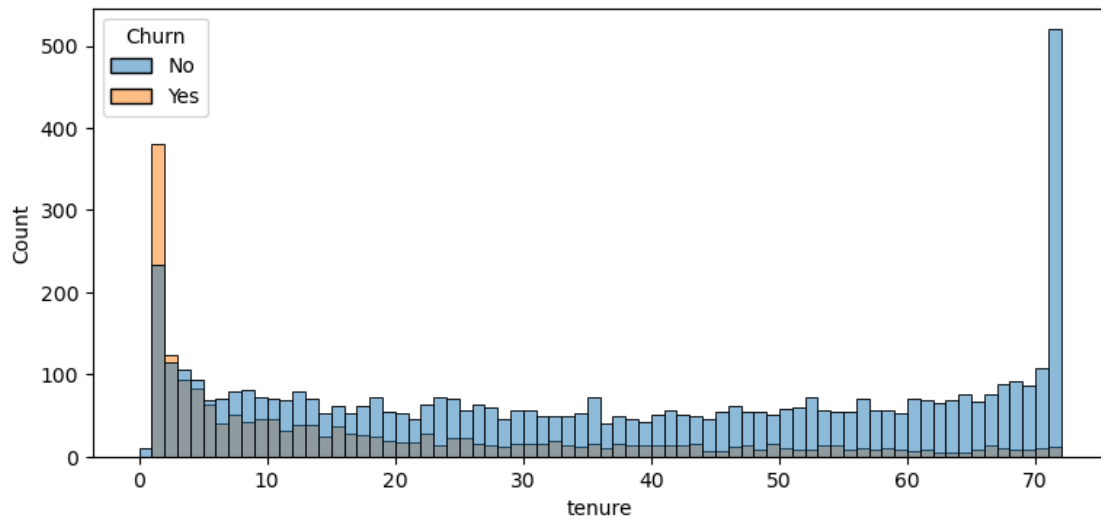
```
# Step 5: Customize the plot
plt.title("Churn by Senior Citizen (Percentage)", fontsize=14)
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.legend(title="Churn", bbox_to_anchor=(1.05, 1), loc="upper left")
plt.tight_layout()
plt.show()
```



#Comparitive a great percentage of people in Senior Citizen have churned.

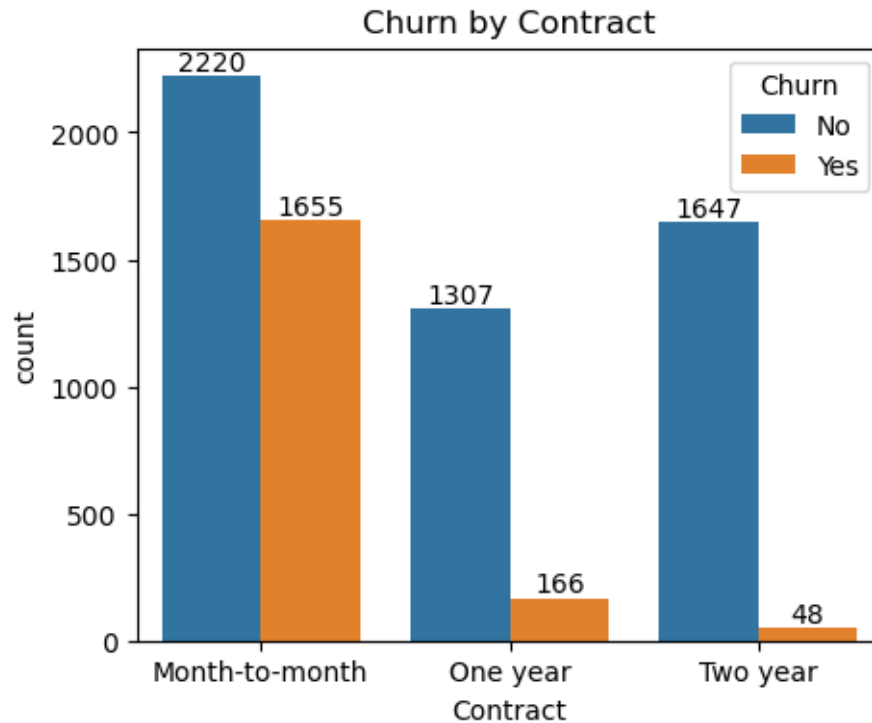
```
[112]: plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()
```

C:\Users\Asus\anaconda3\Lib\site-packages\seaborn_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):



#People who have used our services for a long time have stayed and people who have used our services 1 or 2 months have churned

```
[113]: plt.figure(figsize=(5, 4))
ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[0])
plt.title("Churn 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.

```
[115]: df.columns.values
```

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

```
[121]: 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()

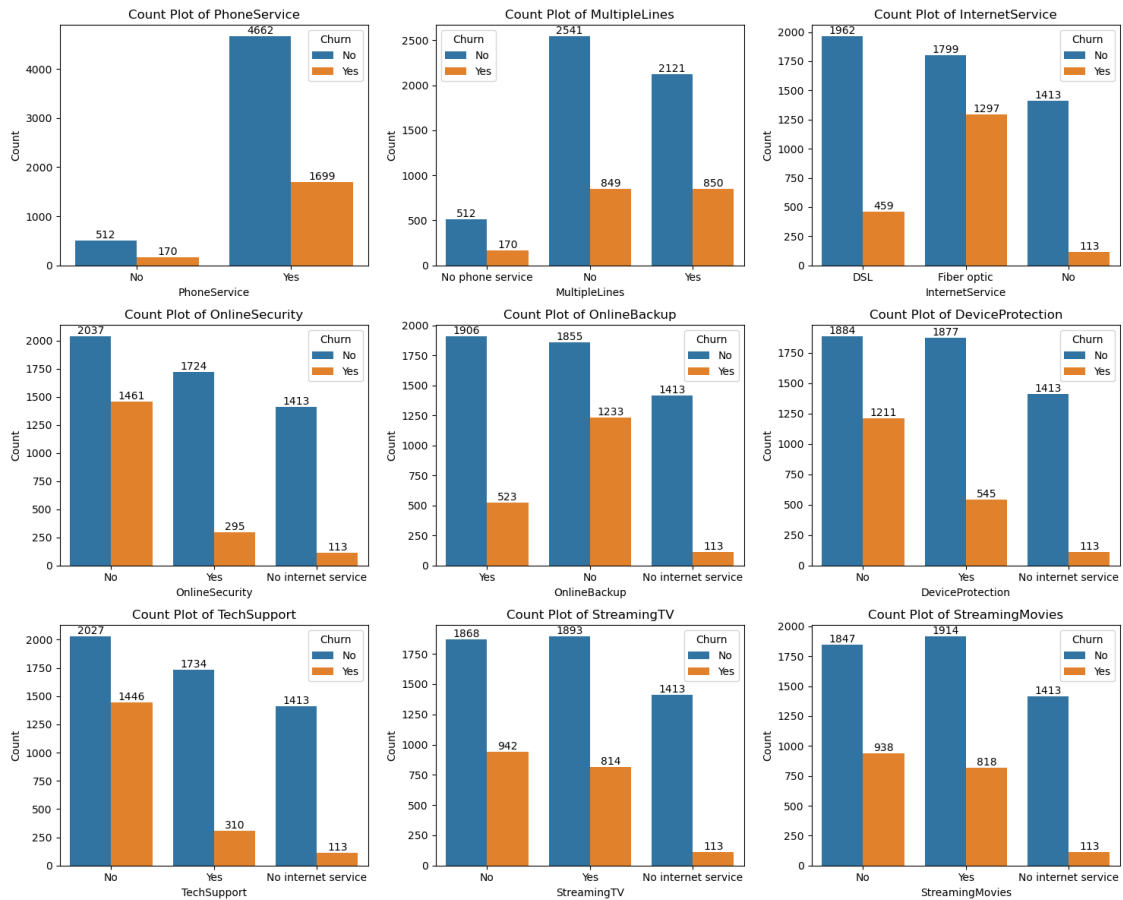
# Iterate over columns and plot count plots
for i, col in enumerate(columns):
    ax = sns.countplot(x=col, data=df, ax=axes[i], hue=df["Churn"])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Count')

    # Add count labels to the bars
    for container in ax.containers:
        ax.bar_label(container, fmt='%d', label_type='edge')

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

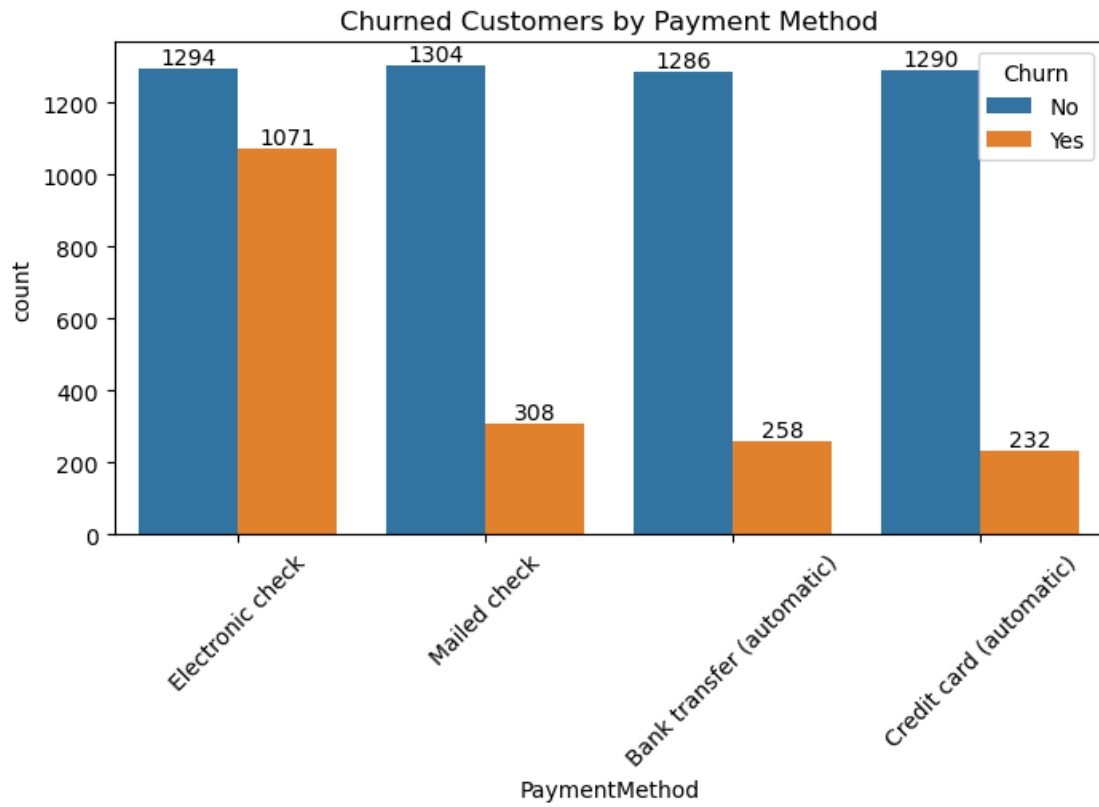
plt.tight_layout()
plt.show()

```



#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.

```
[123]: plt.figure(figsize = (8,4))
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()
```



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

[]: