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

data=pd.read_csv("C:\\Users\\farha\\Downloads\\Telco-Customer-
Churn.csv")
data.head(10)
```

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

data.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
```

```
data.isnull().sum()
```

```
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
```

```
data.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
```

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

```
data.duplicated().sum()
```

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

```
data["customerID"].duplicated().sum()
```

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

```
# making function to convert senior citizen

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


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

```
data.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
```

```
data
```

```
      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
```

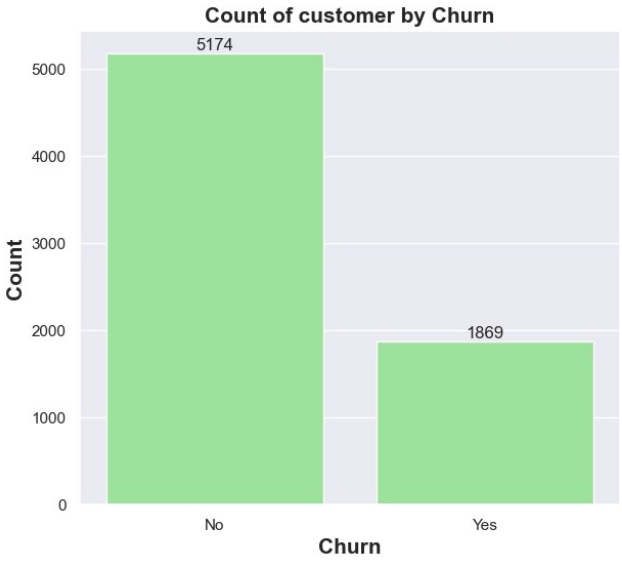
```
7042              Yes  Bank transfer (automatic)          105.65
6844.50

      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]
```

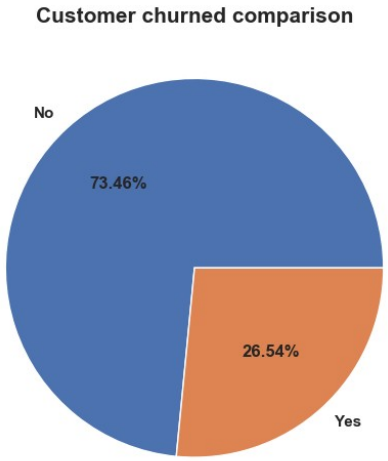
To check how many customer have churned out

```
gb=data.groupby("Churn").agg({'Churn':"count"})
gb
```

```
       Churn
Churn
No      5174
Yes     1869
```

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(15,6))

plt.subplot(1,2,1)
sns.set_theme(style="darkgrid")
ax=sns.countplot(x="Churn",data=data,color="lightgreen")
ax.bar_label(ax.containers[0])
plt.title("Count of customer by Churn",fontweight="bold",fontsize=15)
plt.xlabel("Churn",fontweight="bold",fontsize=15)
plt.ylabel("Count",fontweight="bold",fontsize=15)


plt.subplot(1,2,2)
plt.pie(gb["Churn"],labels=gb.index,autopct="%1.2f%
%",textprops={"fontweight":"bold"})
plt.title("Customer churned comparison",fontweight="bold",fontsize=15)


plt.show()
```



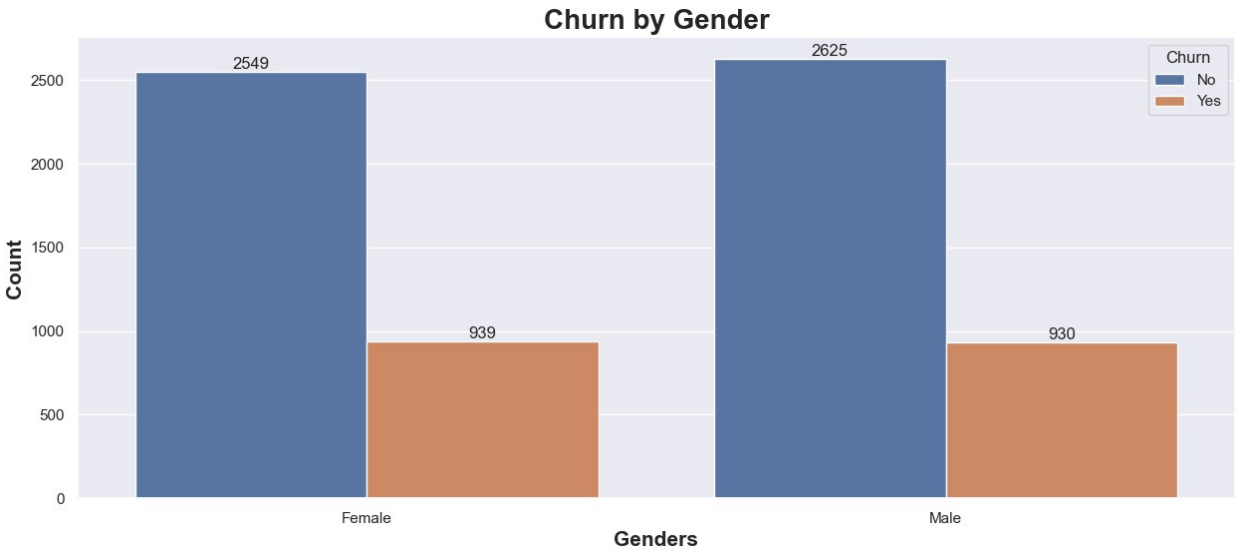



```
g=data.groupby("Churn").agg({"Churn":"count"})
gen=np.unique(data["gender"])
gen
```

```
array(['Female', 'Male'], dtype=object)
```

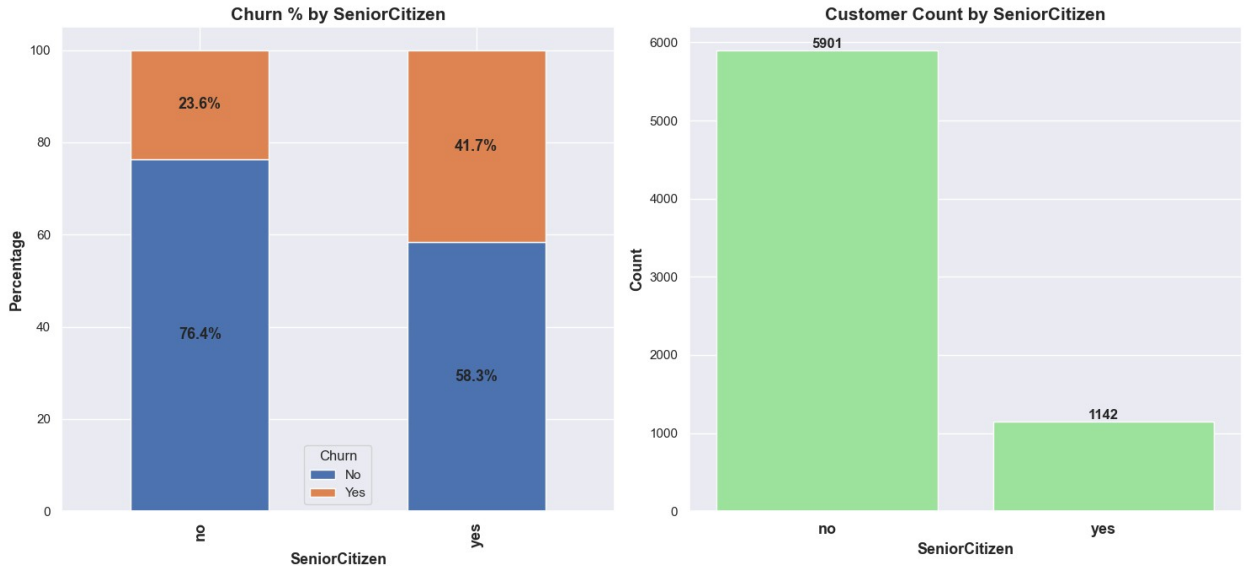
```
# On the gender basis how many churned
```

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(15,6))
ax = sns.countplot(x="gender", data=data, hue="Churn")
for container in ax.containers:
    ax.bar_label(container)


plt.title("Churn by Gender",fontweight="bold",fontsize=20)
plt.xlabel("Genders",fontweight="bold",fontsize=15)
plt.ylabel("Count",fontweight="bold",fontsize=15)
plt.show()
```




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

# Step 1: Group and calculate % churn
grouped = data.groupby(["SeniorCitizen",
"Churn"]).size().unstack(fill_value=0)
percent = grouped.div(grouped.sum(axis=1), axis=0) * 100

# Step 2: Define colors from uploaded image
color_map = {'No': '#4c72b0', 'Yes': '#dd8452'}
colors = [color_map[col] for col in percent.columns]

# Step 3: Create subplots
fig, axs = plt.subplots(1, 2, figsize=(15, 7))

# --- Left plot: Stacked percentage bar chart ---
percent.plot(kind='bar', stacked=True, color=colors, ax=axs[0])

# Add percentage labels
for i in range(percent.shape[0]):
    cum_sum = 0
    for j in range(percent.shape[1]):
        value = percent.iloc[i, j]
        axs[0].text(i, cum_sum + value / 2, f"{value:.1f}%",
ha='center', va='center',
                    fontsize=13, fontweight="bold")
        cum_sum += value

# Customize left plot
axs[0].set_title("Churn % by SeniorCitizen", fontsize=15,
fontweight="bold")
axs[0].set_ylabel("Percentage", fontweight="bold", fontsize=13)
```

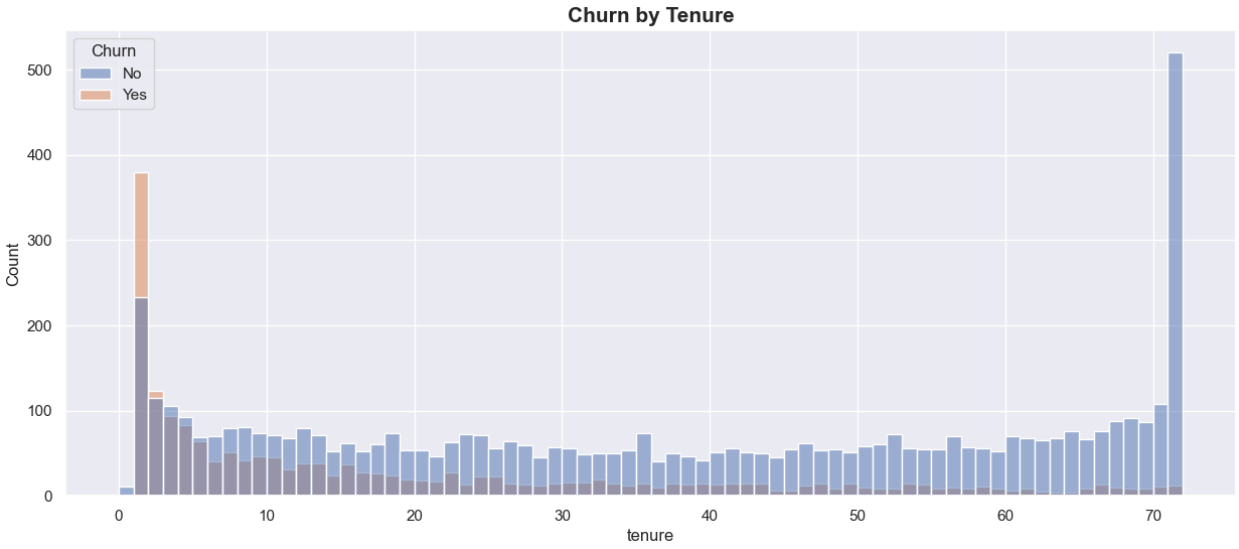
```
axs[0].set_xlabel("SeniorCitizen", fontweight="bold", fontsize=13)
axs[0].tick_params(axis='x', labelsize=12)
axs[0].legend(title="Churn")
for label in axs[0].get_xticklabels():
    label.set_fontsize(13)
    label.set_fontweight("bold")


# --- Right plot: Countplot of customers ---
sns.countplot(x="SeniorCitizen", data=data, ax=axs[1],
color="lightgreen")

# Add count labels on both bars
for container in axs[1].containers:
    for bar in container:
        height = bar.get_height()
        axs[1].text(bar.get_x() + bar.get_width() / 2, height,
f'{int(height)}',
                    ha='center', va='bottom', fontsize=12,
fontweight='bold')

# Customize right plot
axs[1].set_title("Customer Count by SeniorCitizen", fontsize=15,
fontweight="bold")
axs[1].set_ylabel("Count", fontweight="bold", fontsize=13)
axs[1].set_xlabel("SeniorCitizen", fontweight="bold", fontsize=13)
axs[1].tick_params(axis='x', labelsize=12)
for label in axs[1].get_xticklabels():
    label.set_fontsize(13)
    label.set_fontweight("bold")


# Layout adjustment
plt.tight_layout()
plt.show()
```




```
gender_count=data.groupby(data["gender"]).agg({"gender":"count"})
gender_count
```

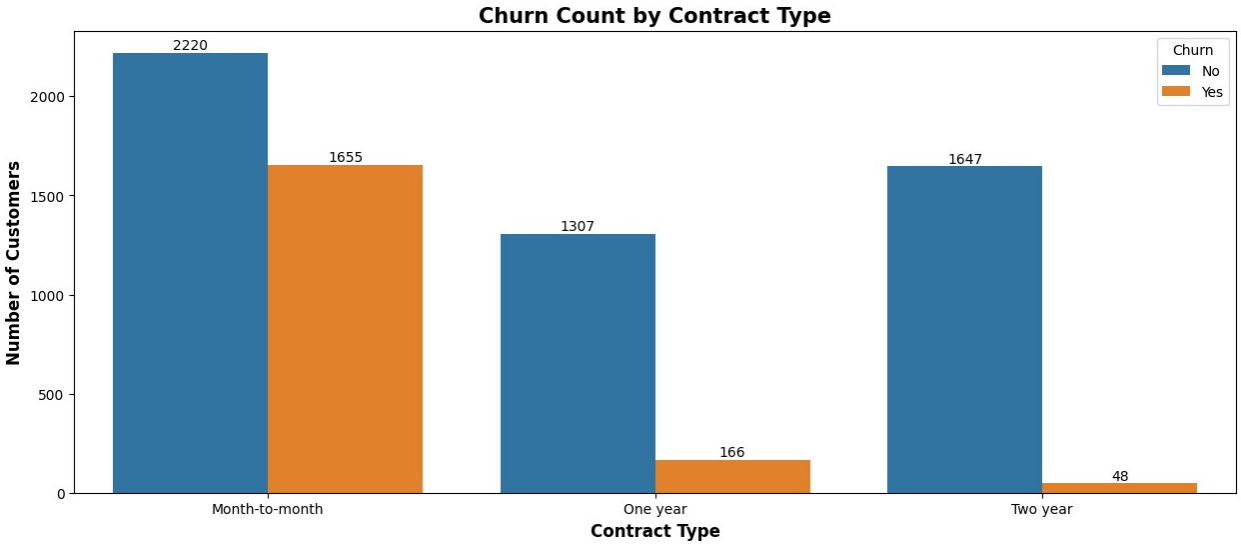
```
        gender
gender
Female    3488
Male      3555
```

```
# Tenure

plt.figure(figsize=(15,6))

sns.set_theme(style="darkgrid")
sns.histplot(x="tenure",data=data,bins=72,hue="Churn")
plt.title("Churn by Tenure",fontsize=15,fontweight="bold")
plt.show()
```



#People who have used our services for a long time have stayed and people who churned out are initial customers.

```
plt.figure(figsize=(15,6))
count=sns.countplot(x="Contract", data=data, hue="Churn")

for container in count.containers:
    count.bar_label(container)

plt.title("Churn Count by Contract
Type",fontweight="bold",fontsize=15)
plt.xlabel("Contract Type",fontweight="bold",fontsize=12)
plt.ylabel("Number of Customers",fontweight="bold",fontsize=12)
plt.show()
```

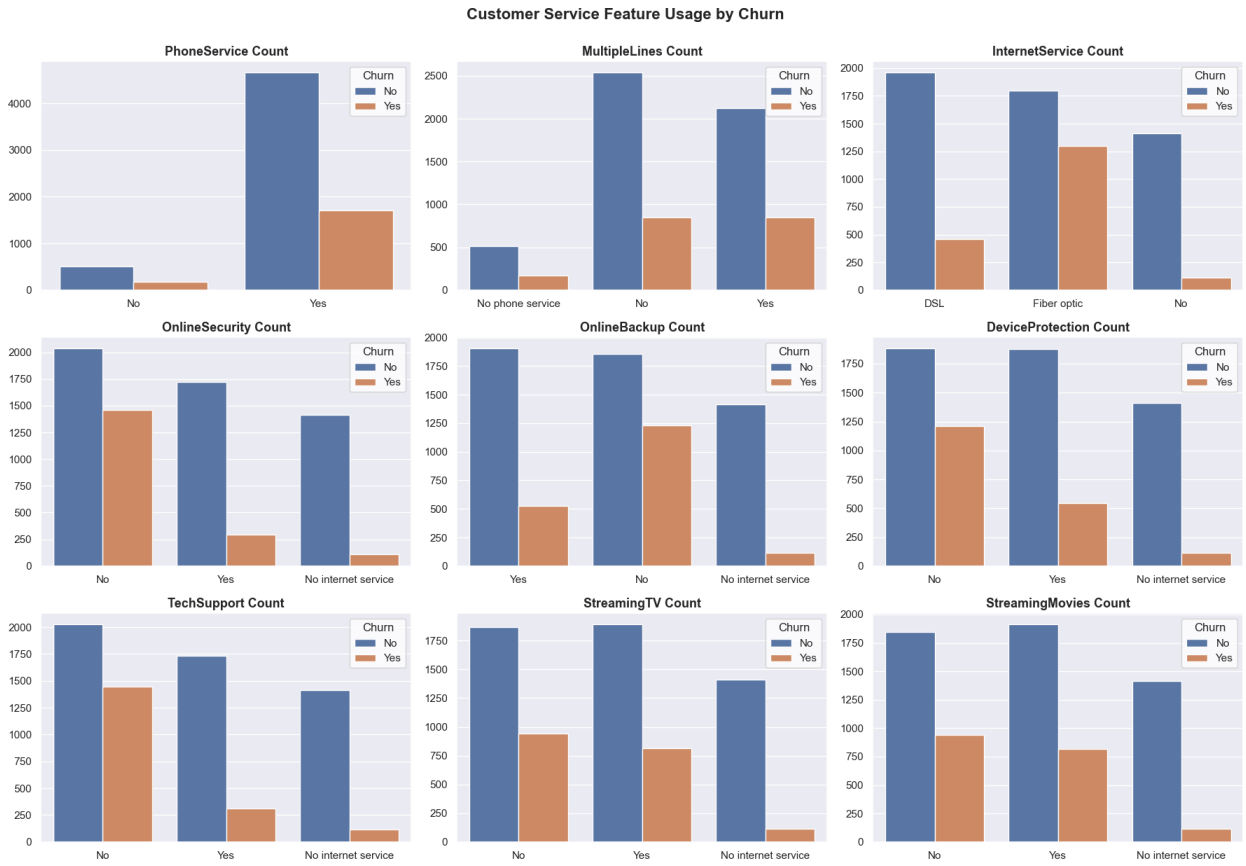


#People who have month-to-month contract are likely to churn then from those who have 1 or 2 year of contracts.

```
data.columns
```

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

```
import matplotlib.pyplot as plt
import seaborn as sns

# List of columns to plot
cols = ['PhoneService', 'MultipleLines', 'InternetService',
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create subplots
fig, axes = plt.subplots(3, 3, figsize=(18, 12))
axes = axes.flatten()

sns.set_theme(style="whitegrid")
# Loop through each column and create countplot with hue='Churn'
for i, col in enumerate(cols):
    sns.countplot(x=col, data=data, hue="Churn", ax=axes[i])
    axes[i].set_title(f"{col} Count", fontweight='bold', fontsize=13)
    axes[i].tick_params(axis='x', rotation=0)  # Ensure ticks are not
rotated
    axes[i].set_xlabel("")  # Optional: remove xlabel to reduce
clutter
    axes[i].set_ylabel("")  # Optional: remove ylabel to reduce
clutter
    # axes[i].grid(axis='y', linestyle='--', alpha=0.5)

# Adjust layout
plt.tight_layout()
plt.suptitle("Customer Service Feature Usage by Churn", fontsize=16,
fontweight='bold', y=1.03)
plt.show()
```



Key Insights by Service Feature

Phone Service

- Most customers who **did not churn** had phone service.
- Churn is relatively **higher among those who had phone service**, indicating phone service alone doesn't retain customers.

☎ Multiple Lines

- Churn is **slightly higher** for customers with **no additional lines** compared to those with multiple lines.

Internet Service

- **Fiber optic users churn more** than DSL or customers without internet.
- **DSL users** had the **lowest churn**, suggesting better retention with DSL.

Online Security

- Customers **without online security** had **higher churn**.
- Those **with online security** churned less, suggesting it may **improve retention**.

Online Backup

- Similar pattern as online security: **having backup service reduces churn**.

Device Protection

- Churn is **lower** among those who have **device protection services**.
- Lack of protection correlates with **higher churn**.

Tech Support

- Customers **without tech support** churn more.
- Availability of support services appears to **positively impact retention**.

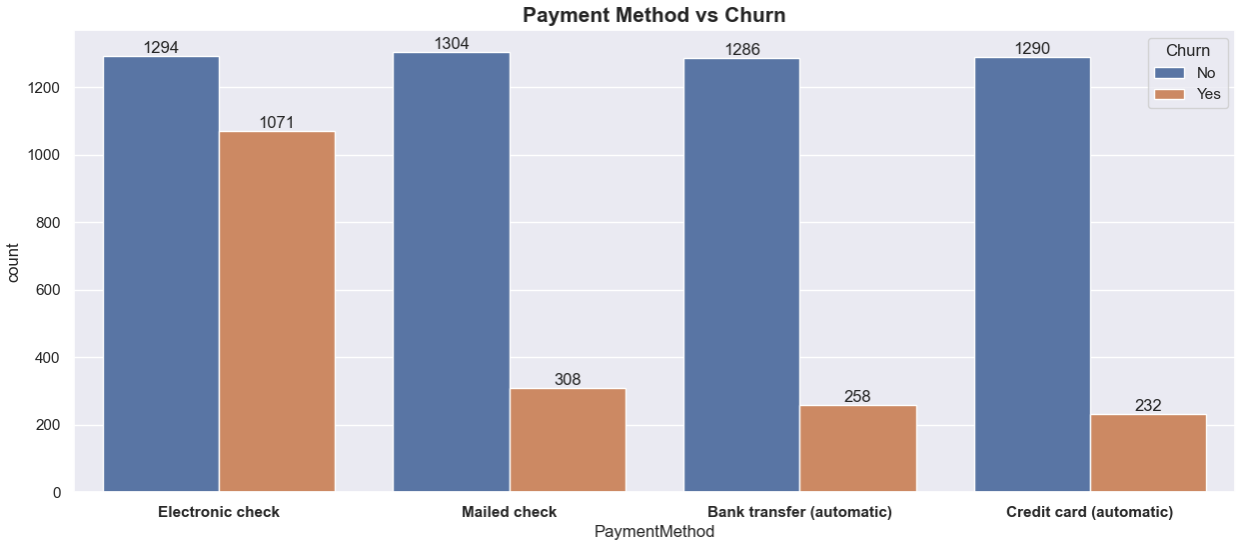
Streaming TV & Movies

- Churn is **higher in both groups (Yes/No)**, but slightly **lower for those who use streaming services**.
- Entertainment features may have a **moderate positive impact** on retention.

Overall Summary

- Customers who **lack additional services** (security, backup, protection, tech support) are **more likely to churn**.
- **Fiber optic internet users churn the most**, suggesting issues with satisfaction or cost.
- Offering **bundled services** like security, backup, and tech support can help **reduce churn**.

```
# churned analysis by payment method


plt.figure(figsize=(15,6))

sns.set_theme(style="darkgrid")
x=sns.countplot(x="PaymentMethod",data=data,hue="Churn")
for container in x.containers:
    x.bar_label(container)

plt.title("Payment Method vs Churn",fontweight="bold",fontsize=15)
plt.xticks(fontweight="bold")
plt.show()
```



Payment Method vs Churn Insight

- Customers using the **Electronic check** payment method show a significantly **higher churn rate** compared to other payment methods.
- In contrast, customers paying via **Mailed check**, **Bank transfer (automatic)**, or **Credit card (automatic)** have **lower churn rates**.
- This suggests that **automatic or more secure payment methods** may improve customer retention by offering greater convenience or satisfaction.