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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

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

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

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

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

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No

5 rows × 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
```

Converting the Data Type of TotalCharges as Float and replacing blanks with 0

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

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

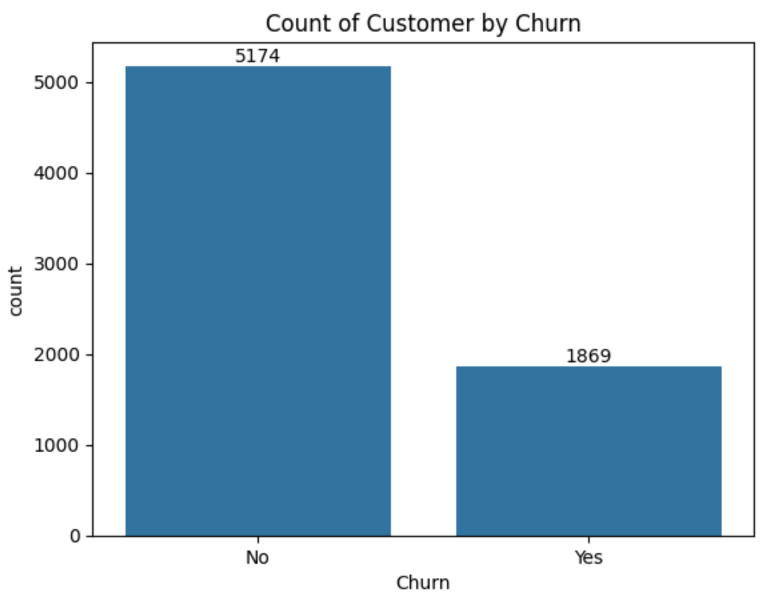
Coverted 0 and 1 values of Senior citizen to yes/no

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

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

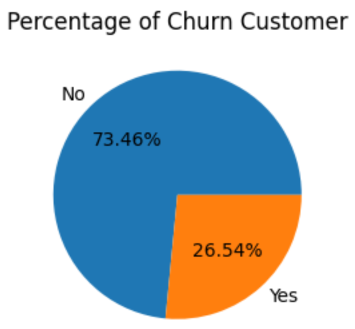
```
df[["SeniorCitizen"]].head()
```

	SeniorCitizen
0	no
1	no
2	no
3	no
4	no

```
ax = sns.countplot(x=df["Churn"],data=df)
#for count
ax.bar_label(ax.containers[0])
plt.title("Count of Customer by Churn")
plt.show()
```

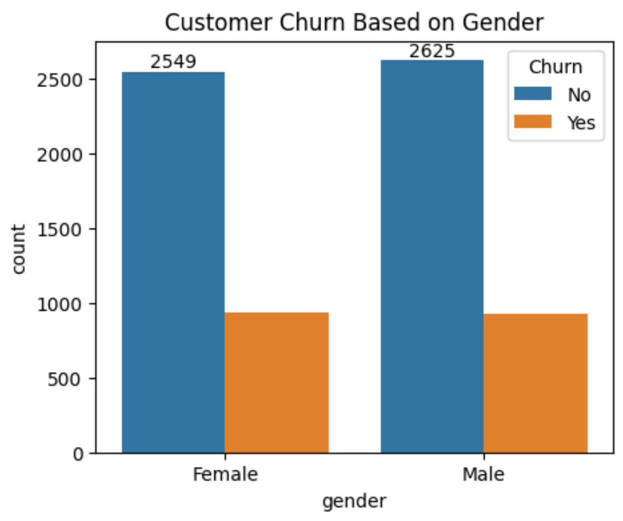


```
plt.figure(figsize=(3,4))
gb = df.groupby("Churn").agg({'Churn':'count'})
plt.pie(gb["Churn"], labels=gb.index, autopct="%1.2f%%")
plt.title("Percentage of Churn Customer")
plt.show()
```

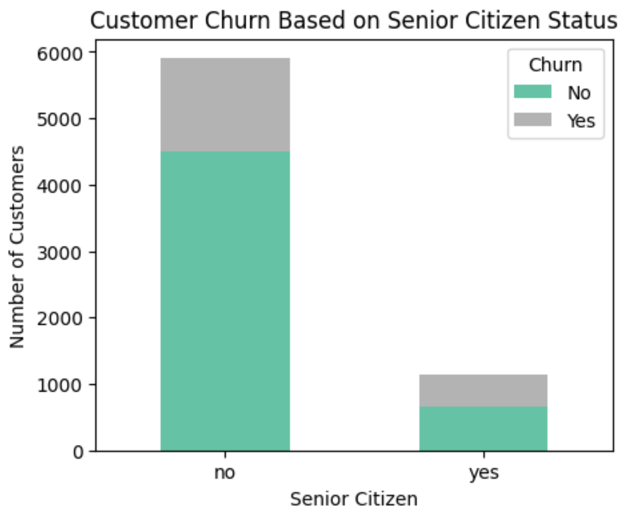


```
#from the above Pie Chart we can conclude that 26% OF the Customers have Churned Out
#exploring the reason behind it
```

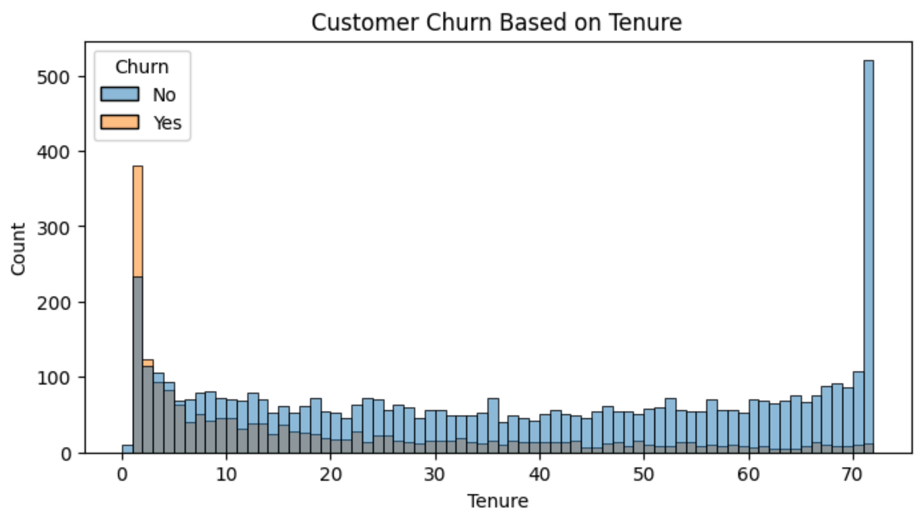
```
plt.figure(figsize =(5,4))
ax = sns.countplot(x="gender",data= df,hue="Churn")
ax.bar_label(ax.containers[0])
plt.title("Customer Churn Based on Gender")
plt.show()
```



```
# Prepare the stacked data
stack_data = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)

# Plot stacked bar chart
stack_data.plot(kind='bar', stacked=True, figsize=(5,4), colormap='Set2')

plt.title("Customer Churn Based on Senior Citizen Status")
plt.xlabel("Senior Citizen")
plt.ylabel("Number of Customers")
plt.legend(title="Churn")
plt.xticks(rotation=0)
plt.show()
```

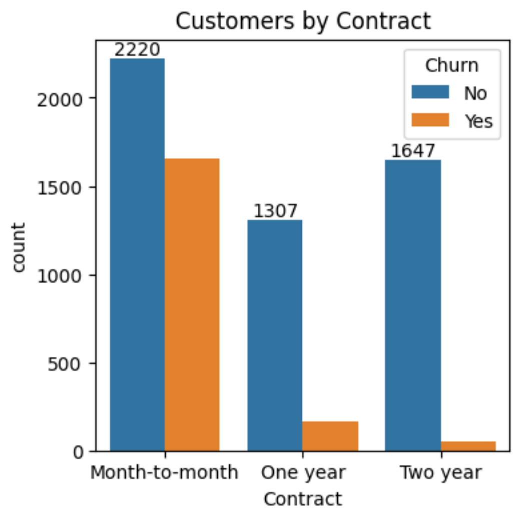


```
plt.figure(figsize=(8,4))
sns.histplot(x="tenure", data=df,bins=72,hue="Churn")
plt.xlabel("Tenure")
plt.title("Customer Churn Based on Tenure")
plt.show()
```



```
#people who have used our services for long month have stayed and people who have used our services for one or two months have
```

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



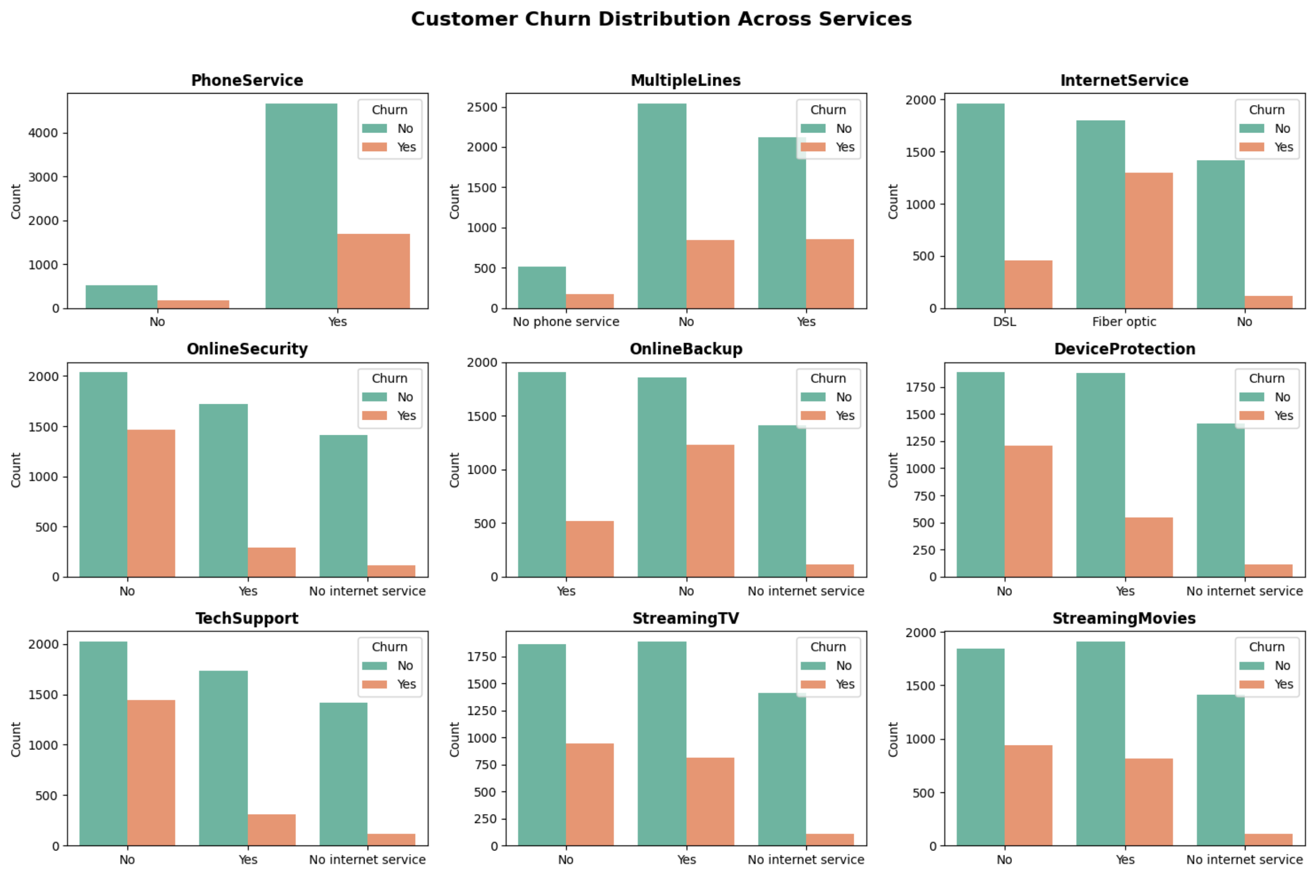
```
#People who have month to month Contrct have Churn Earlier than one Year Contract People
```

```
df.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')
```

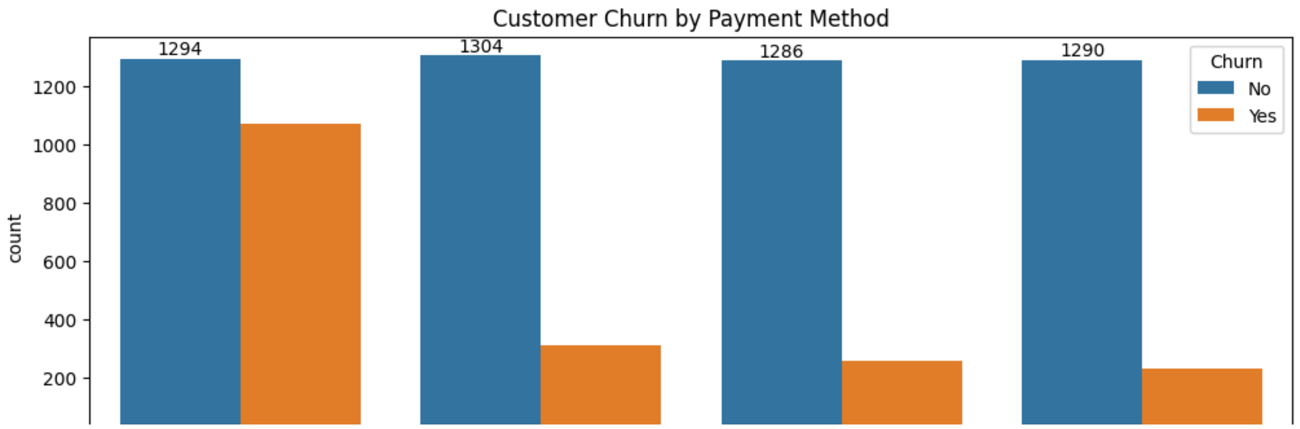
```
# List of columns to plot
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Create a grid of subplots (3 rows × 3 columns)
```

```
fig, axes = plt.subplots(3, 3, figsize=(15, 10))
fig.suptitle("Customer Churn Distribution Across Services", fontsize=16, fontweight='bold')

# Flatten axes for easy iteration
axes = axes.flatten()

# Loop through each column and plot a countplot
for i, col in enumerate(cols):
    sns.countplot(data=df, x=col, hue="Churn", ax=axes[i], palette="Set2")
    axes[i].set_title(col, fontsize=12, fontweight='bold')
    axes[i].set_xlabel("")
    axes[i].set_ylabel("Count")
    axes[i].legend(title="Churn", loc="upper right")

# Remove any unused subplot (if number of cols < grid size)
for j in range(len(cols), len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



```
#Customers with value-added services like Online Security, Tech Support, and Device Protection show much lower churn, while thc
#Fiber optic users have the highest churn, indicating possible pricing or service issues.
#Overall, higher service engagement strongly correlates with better customer retention.
```

```
plt.figure(figsize=(12,4))
ax.bar_label(ax.containers[0])
ax= sns.countplot(x ="PaymentMethod",data=df, hue="Churn")
ax.bar_label(ax.containers[0])
plt.title("Customer Churn by Payment Method")
plt.show()
```


Customer Churn by Payment Method
count
1200
1000
800
600
400
200
1294
1304
1286
1290
Churn
No
Yes