Install python libraries

pip install pandas pip install seaborn pip install matplotlib pip install numpy

Importing libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("E:\Projects\EDA\Customer-Churn-analysis-main\Customer Cl
df.head()
```

```
<>:6: SyntaxWarning: invalid escape sequence '\P'
<>:6: SyntaxWarning: invalid escape sequence '\P'
C:\Users\dines\AppData\Local\Temp\ipykernel_16804\817859210.py:6: SyntaxWarni
invalid escape sequence '\P'
  df = pd.read_csv("E:\Projects\EDA\Customer-Churn-analysis-main\Customer
Churn.csv")
```

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Multipl
0	7590-VHVEG	Female	0	Yes	No	1	No	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	No
4	9237-HQITU	Female	0	No	No	2	Yes	

5 rows × 21 columns

In [4]:
```
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 is 0 and no t
Charges are Recorded

In [5]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" ",0)
df["TotalCharges"] = df["TotalCharges"].astype("float")
print("Excuted")
```

```
Excuted
```

In [6]:
```
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
```

In [7]:
```
df.isnull().sum().sum()
```

Out[7]:
```
0
```

In [8]:
```
df.describe()
```

Out[8]:

	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

In [9]:
```
df["customerID"].duplicated().sum()
```

Out[9]:
```
0
```

converted 1 and 0 as Yes and No respectively SeniorCitizen

In [10]:
```
def conv(value):
    if value == 1:
        return "Yes"
    else:
        return "No"

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

```
Completed
```

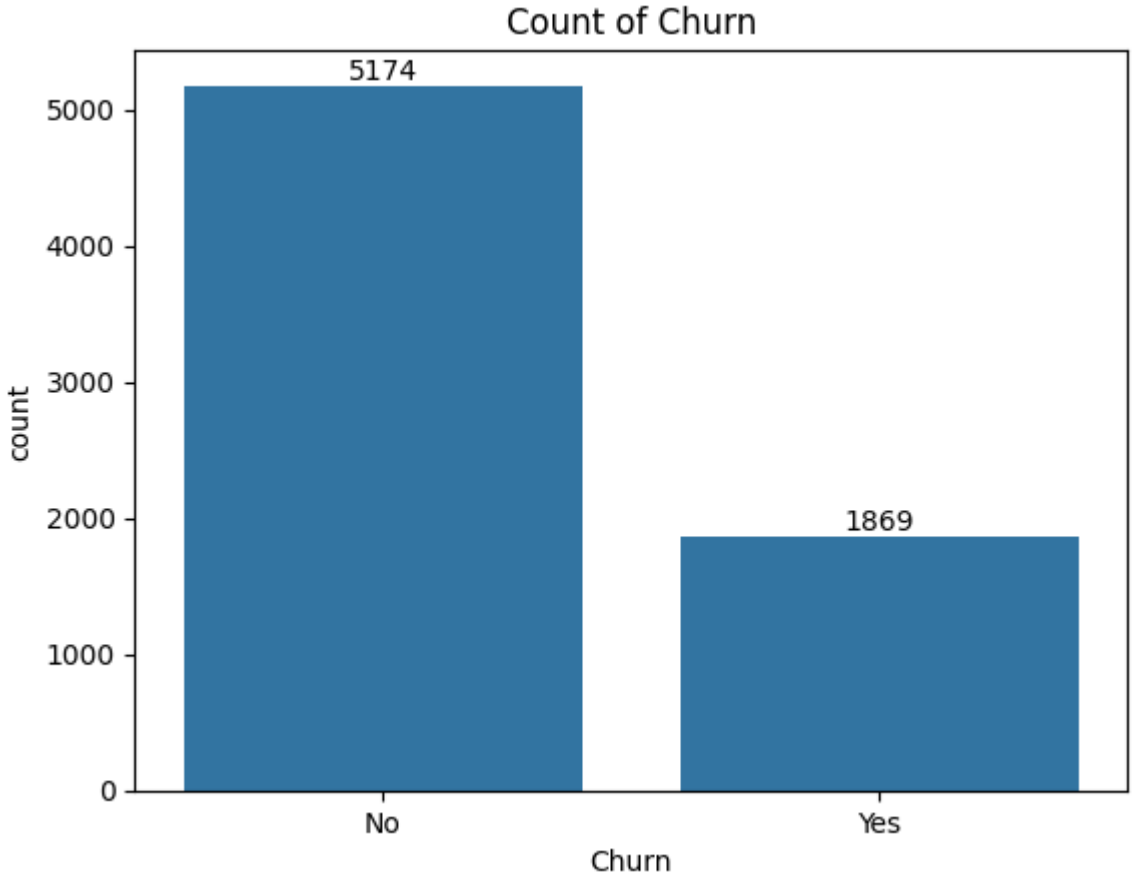
In [11]:
```
df.head()
```

Out[11]:

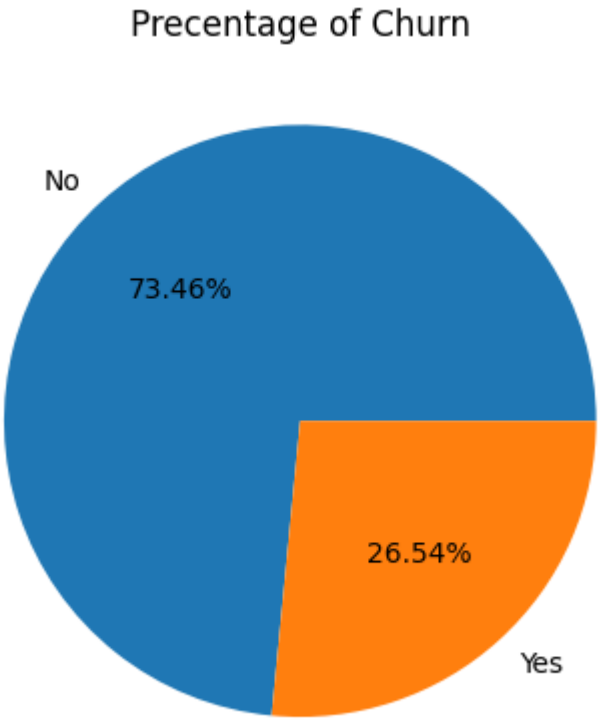
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Multipl
0	7590-VHVEG	Female	No	Yes	No	1	No	No
1	5575-GNVDE	Male	No	No	No	34	Yes	
2	3668-QPYBK	Male	No	No	No	2	Yes	
3	7795-CFOCW	Male	No	No	No	45	No	No
4	9237-HQITU	Female	No	No	No	2	Yes	

5 rows × 21 columns

In [12]:
```
ax = sns.countplot(x = 'Churn' , data = df )
plt.title("Count of Churn")
ax.bar_label(ax.containers[0])
plt.show()
```

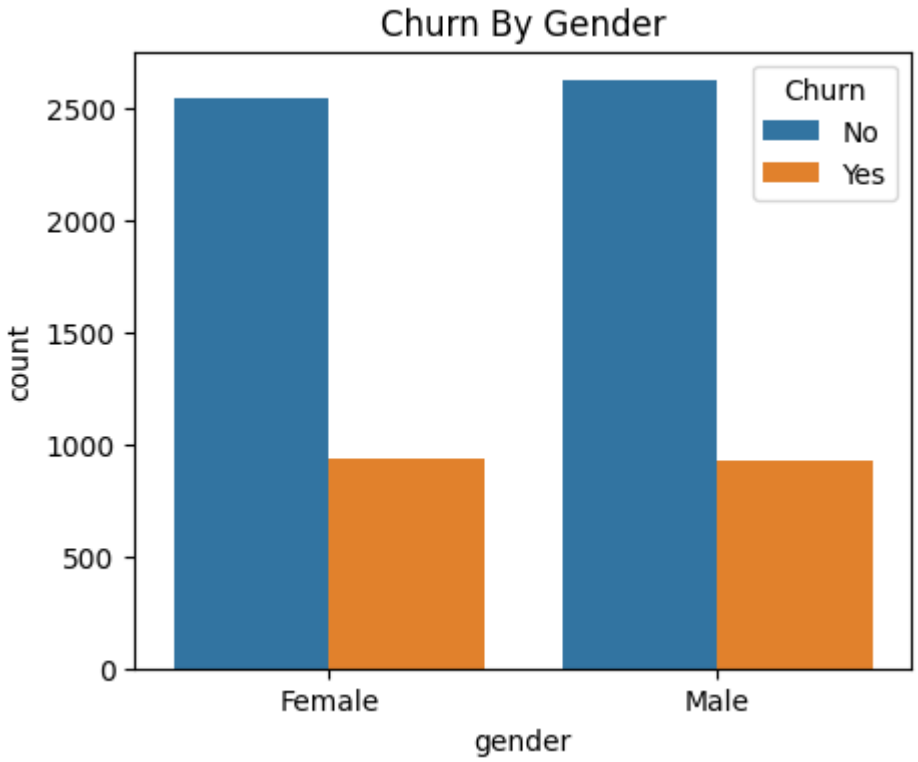


In [13]:
```
gb = df.groupby("Churn").agg({'Churn': "count"})
plt.pie(gb['Churn'] , labels = gb.index , autopct = '%1.2f%%')
plt.title("Precentage of Churn")
plt.show()
```

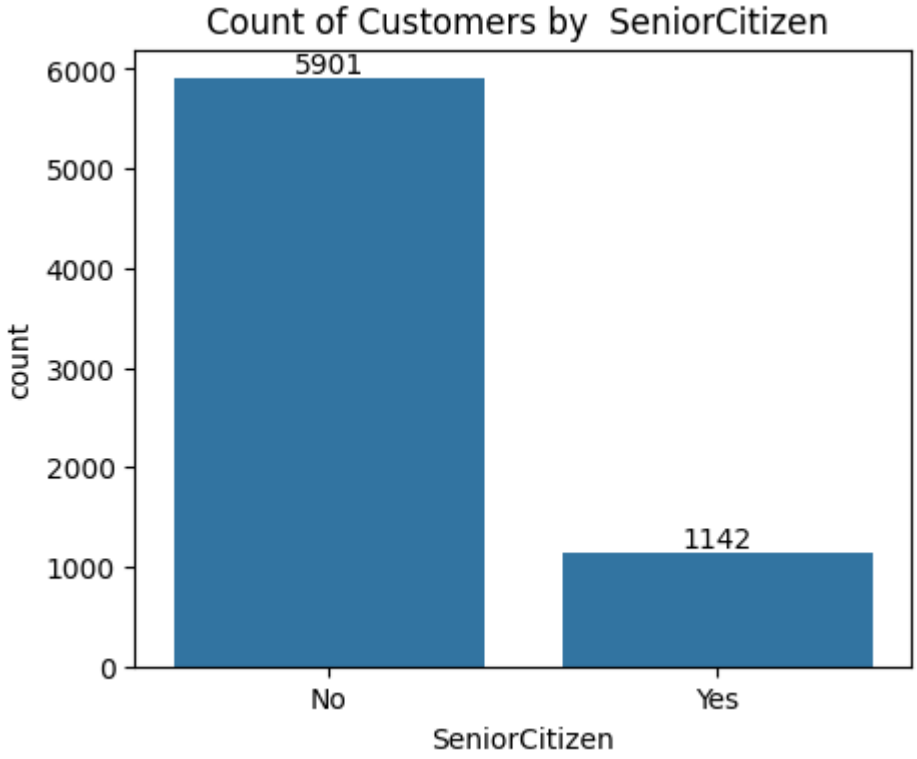


from above pic chart we can conclude that 26.5
of our Customers have Churned Out lets us fin
reason about it

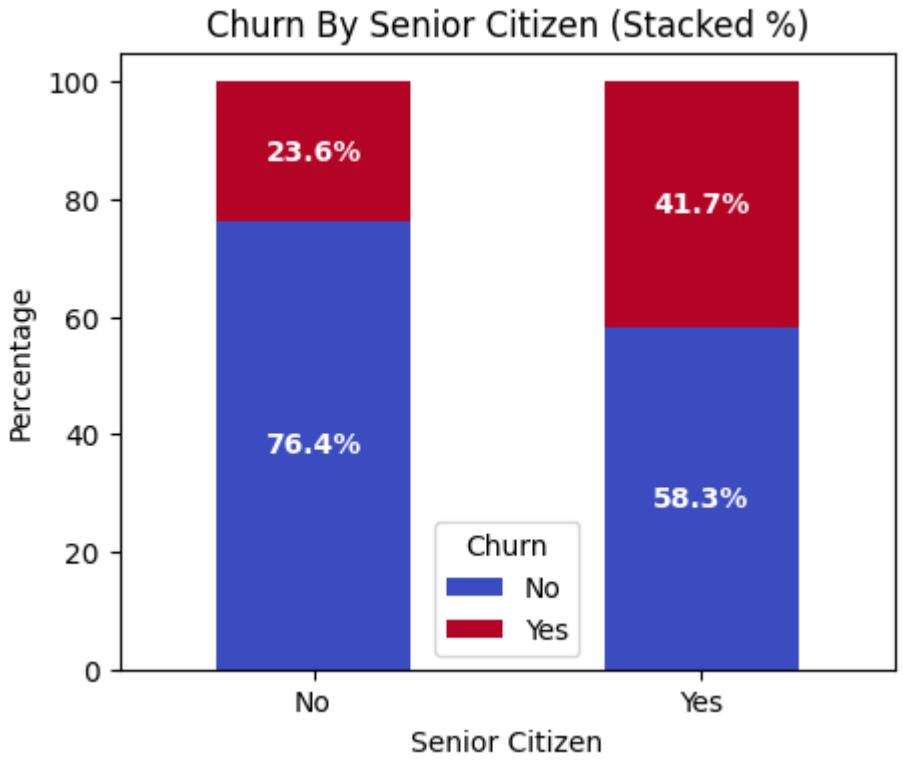
In [24]:
```
plt.figure(figsize = (5,4))
sns.countplot(x = "gender" , data = df , hue = "Churn")
plt.title("Churn By Gender")
plt.show()
```



In [33]:
```
plt.figure(figsize = (5,4))
ax = sns.countplot(x = "SeniorCitizen" , data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by  SeniorCitizen ")
plt.show()
```

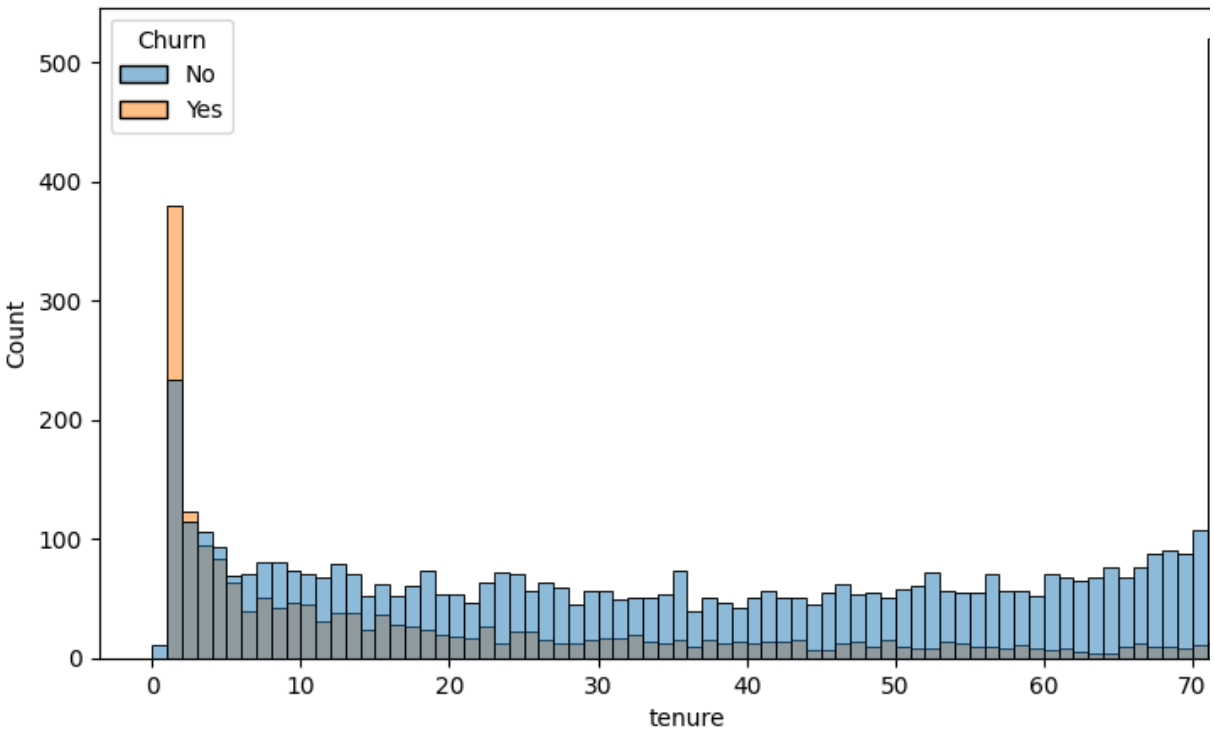


In [28]:
```
counts = df.groupby(["SeniorCitizen", "Churn"]).size().unstack()
percentages = counts.div(counts.sum(axis=1), axis=0) * 100

fig, ax = plt.subplots(figsize=(5, 4))
percentages.plot(kind="bar", stacked=True, ax=ax, colormap="coolwarm")

for bar_container in ax.containers:
    for bar in bar_container:
        height = bar.get_height()
        if height > 0:
            ax.text(
                bar.get_x() + bar.get_width() / 2,
                bar.get_y() + height / 2,
                f"{height:.1f}%",
                ha="center",
                va="center",
                color="white",
                fontsize=10,
                fontweight="bold"
            )

plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.title("Churn By Senior Citizen (Stacked %)")
plt.xticks(rotation=0)
plt.legend(title="Churn")
plt.show()
```



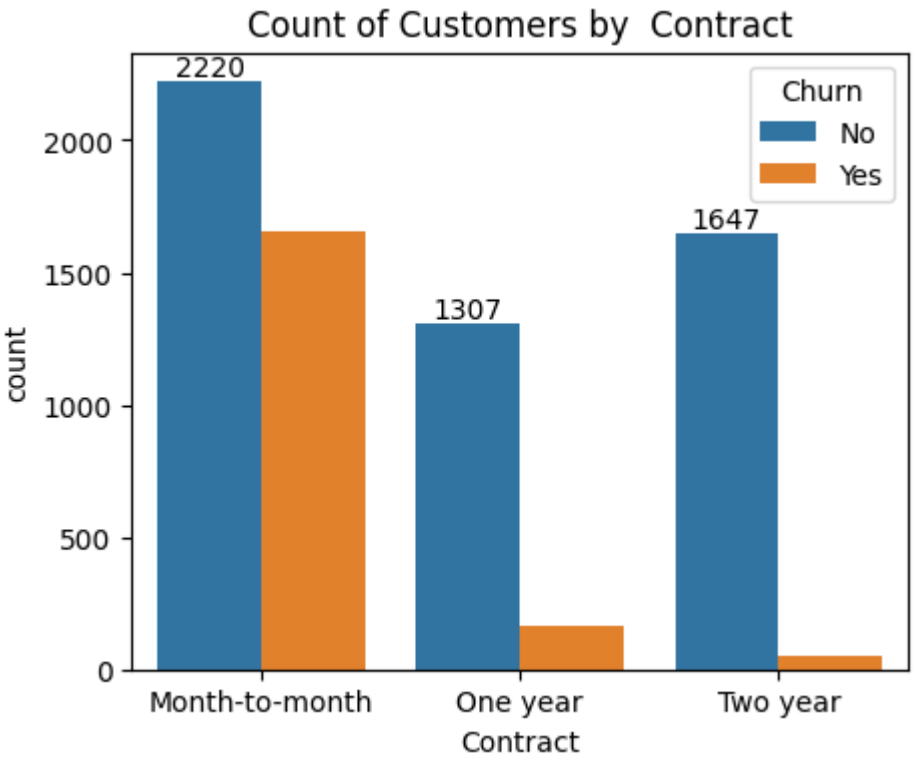
comparative a greated percentage of people in Senior Citizen Category have churned

In [40]:
```
plt.figure(figsize = (9,5))
sns.histplot(x = "tenure" , data = df , bins = 72 , hue = "Churn")
plt.show()
```



poeple who have used our serverice for long ti have Stayed and people who have used our services for 1 or 2 months have churned

In [42]:
```
plt.figure(figsize = (5,4))
ax = sns.countplot(x = "Contract" , data = df , hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by  Contract")
plt.show()
```

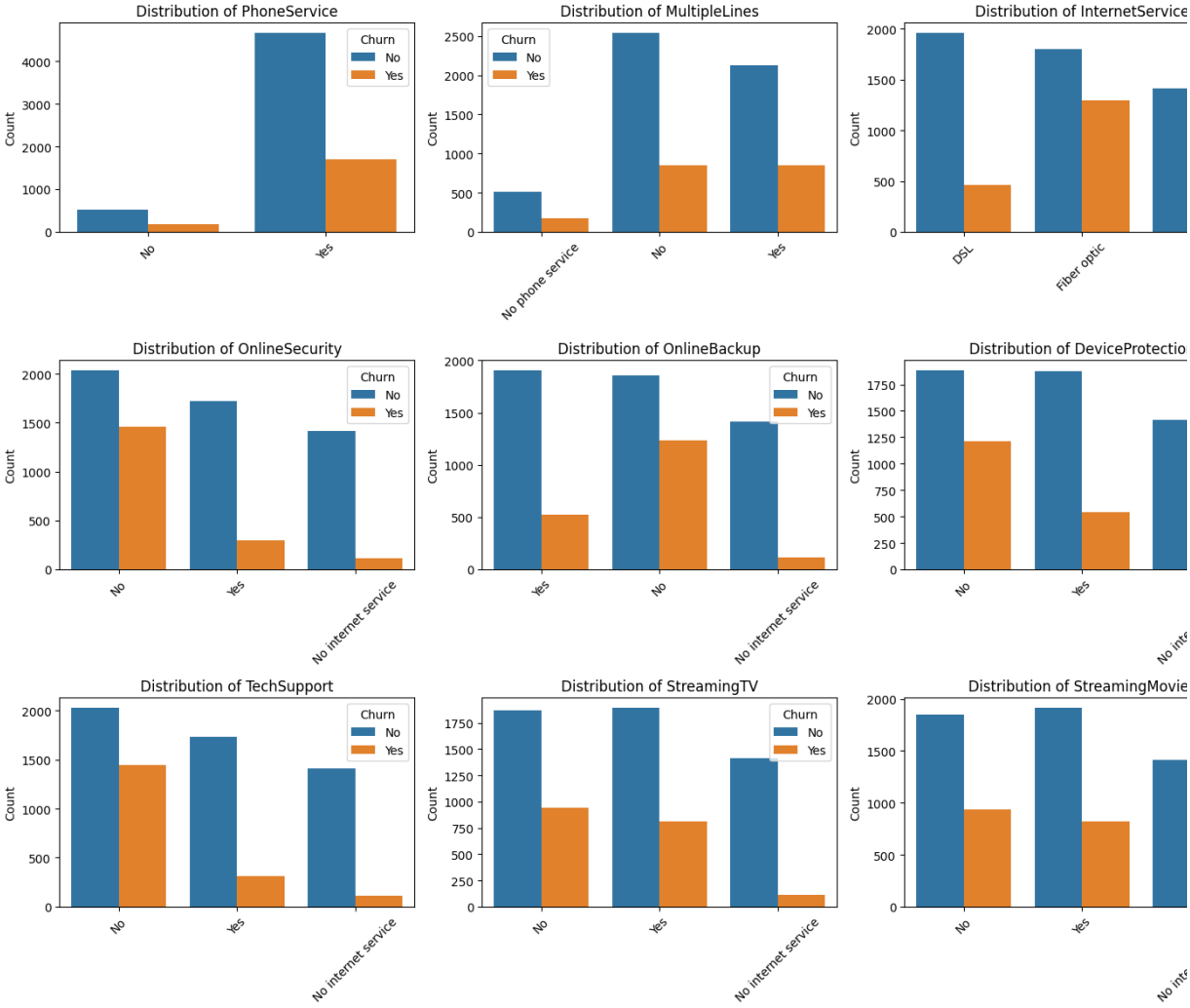


people who have month-month likely to churn t people who have 1 or 2 year or Contract

In [43]:
```
df.columns.values
```

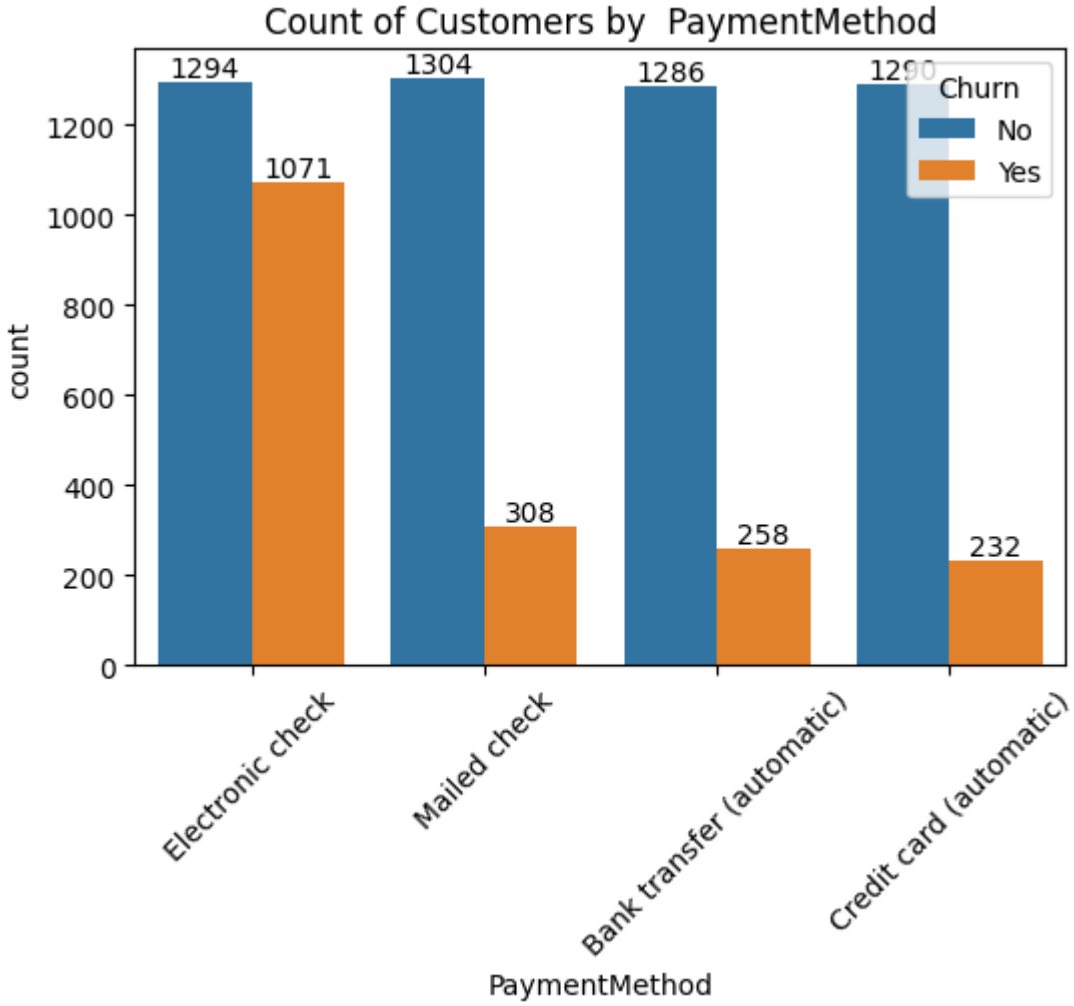
Out[43]:
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

In [46]:
```
columns = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))  # 3x3 grid
axes = axes.flatten()

for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i] , hue = "Churn")
    axes[i].set_title(f"Distribution of {col}")
    axes[i].set_xlabel("")
    axes[i].set_ylabel("Count")
    axes[i].tick_params(axis="x", rotation=45)

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



The visualizations display the distribution of various telecom services (e.g., PhoneService, InternetService, OnlineSecurity, etc.) among customers who churned versus those who did In general, customers with no additional servic (e.g., OnlineSecurity, TechSupport) show a hig churn rate. Fiber optic internet users also have relatively higher churn rate than DSL users. Th suggests that customers lacking value-added services are more likely to churn, highlighting opportunities for retention strategies.

In [55]:
```
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.xticks(rotation = 45)
plt.title("Count of Customers by  PaymentMethod")
plt.show()
```



Customers is likely to churn when he is using electronic Check as a payment method

In [ ]:

In [ ]: