In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
df=pd.read_csv("https://github.com/YBIFoundation/Dataset/raw/main/TelecomCustomerCh
```

In [5]:
```
df
```

Out[5]:

	customerID	Gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	Mu
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	
...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

7043 rows × 21 columns

replacing blank with 0 as tenure is 0 and no total charges are recorded

In [10]:
```
df["TotalCharges"]= df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

In [11]:
```
df
```

Out[11]:

	customerID	Gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	Mu
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	
...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

7043 rows × 21 columns

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   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

In [12]:
```
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 [14]:
```
df.isnull().sum()
```

Out[14]:
```
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
```

In [15]:
```
df.describe()
```

Out[15]:

	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 [17]:
```
df["customerID"].duplicated().sum()
```

Out[17]:
```
np.int64(0)
```

In [22]:
```
#Coverted 0 and 1 value of senior citizen to no and yess to make it easier to under
def conv(value):
    if value==1:
        return"yes"
    else:
        return"no"
df["SeniorCitizen"]= df["SeniorCitizen"].apply(conv)
```

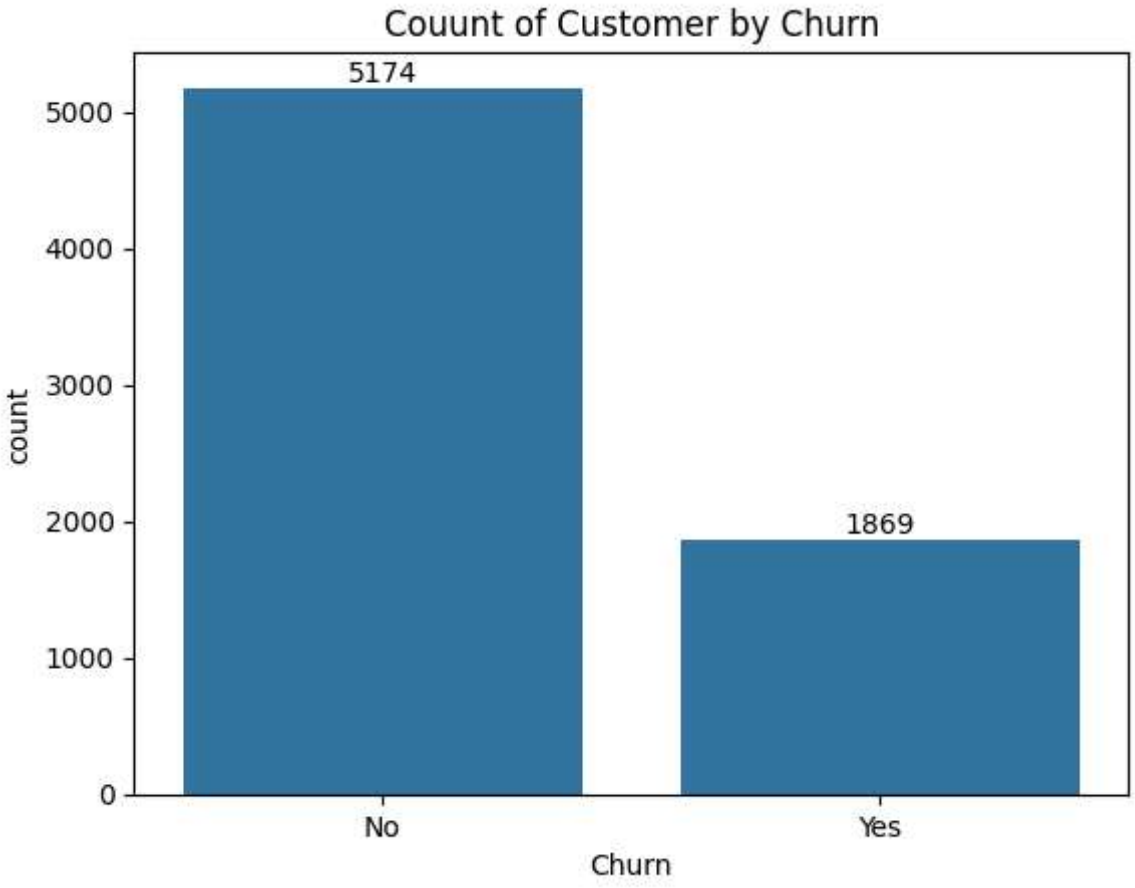
In [23]:
```
df
```

Out[23]:

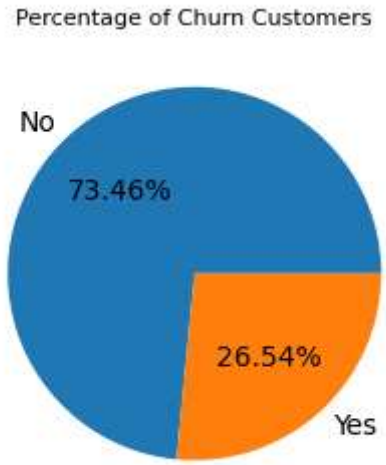
	customerID	Gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	Mu
0	7590-VHVEG	Female	no	Yes	No	1	No	
1	5575-GNVDE	Male	no	No	No	34	Yes	
2	3668-QPYBK	Male	no	No	No	2	Yes	
3	7795-CFOCW	Male	no	No	No	45	No	
4	9237-HQITU	Female	no	No	No	2	Yes	
...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	no	Yes	Yes	24	Yes	
7039	2234-XADUH	Female	no	Yes	Yes	72	Yes	
7040	4801-JZAZL	Female	no	Yes	Yes	11	No	
7041	8361-LTMKD	Male	yes	Yes	No	4	Yes	
7042	3186-AJIEK	Male	no	No	No	66	Yes	

7043 rows × 21 columns

In [39]:
```
ax=sns.countplot(x="Churn",data=df)
ax.bar_label(ax.containers[0])
plt.title("Couunt of Customer by Churn")
plt.show()
```



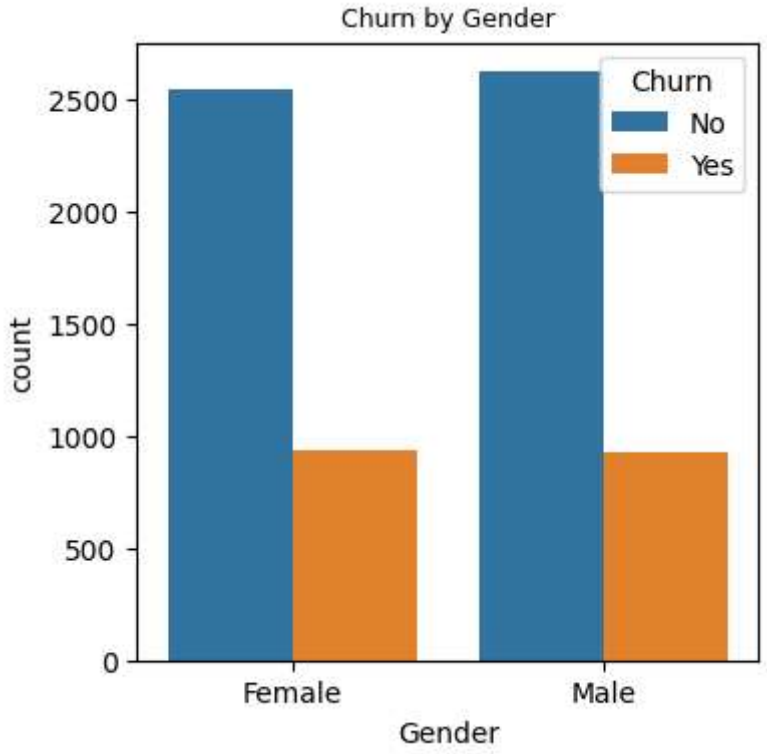
In [40]:
```
plt.figure(figsize =(3,4))
gb = df.groupby("Churn").agg({"Churn":"count"})
plt.pie(gb["Churn"], labels = gb.index,autopct= "%1.2f%%")
plt.title("Percentage of Churn Customers",fontsize=8)
plt.show()
```



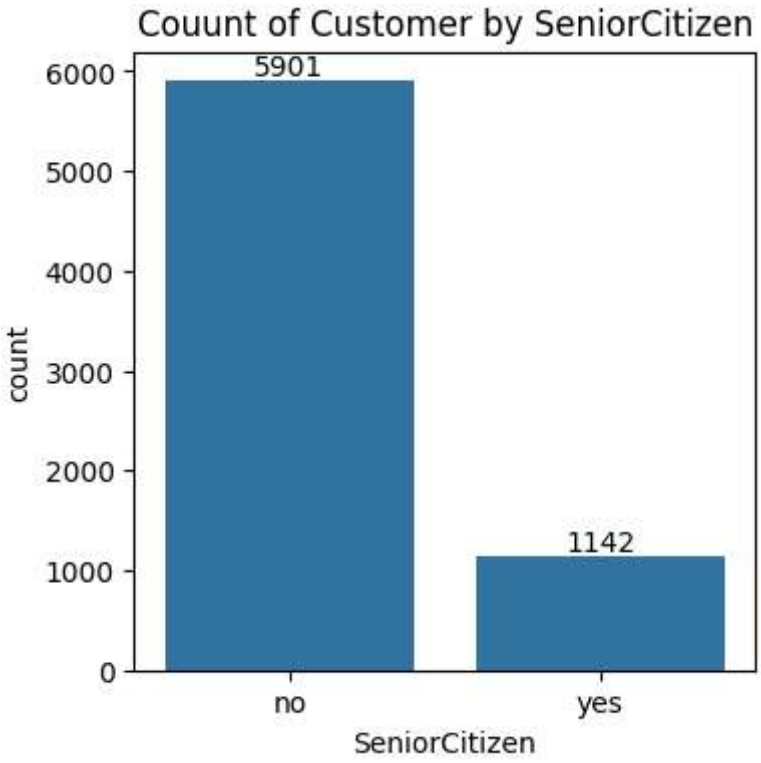
from the give pie chart we can conclude that 26.54% of our customers have

churned out not lets explore the reson behind it

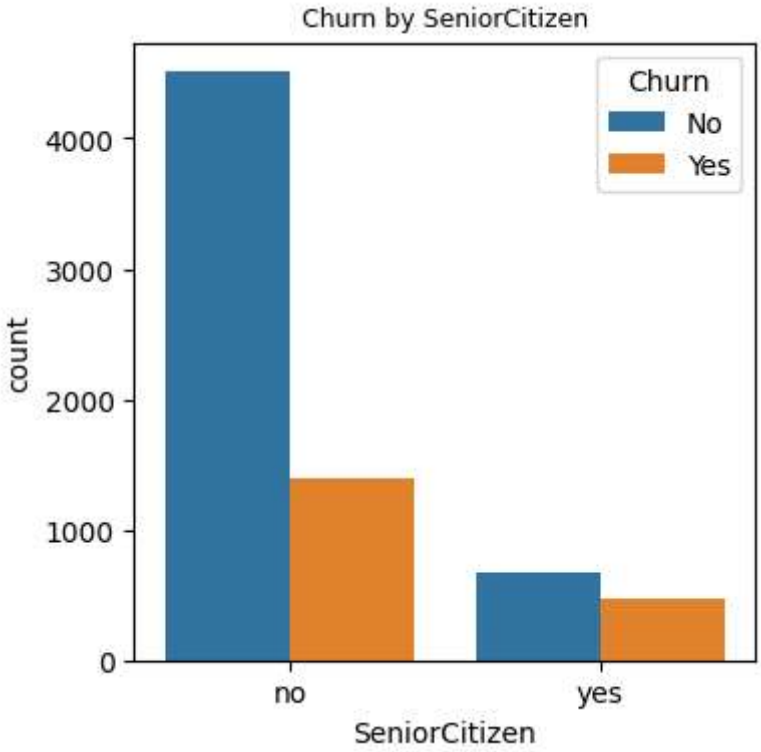
In [52]:
```
plt.figure(figsize=(4,4))
sns.countplot(x ="Gender",data = df,hue="Churn")
plt.title("Churn by Gender",fontsize=9)
plt.show()
```



In [56]:
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x="SeniorCitizen",data=df)
ax.bar_label(ax.containers[0])
plt.title("Couunt of Customer by SeniorCitizen")
plt.show()
```

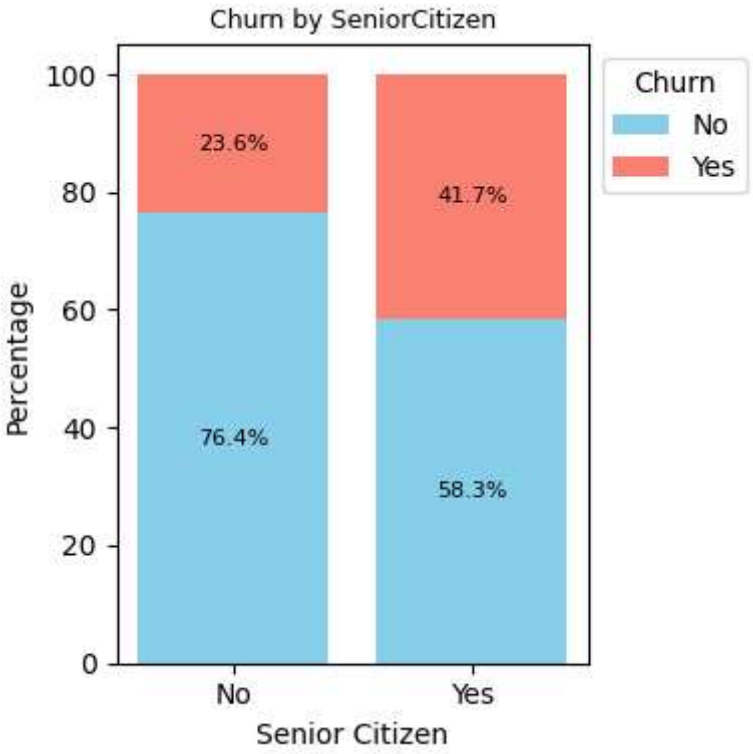


In [54]:
```
plt.figure(figsize=(4,4))
sns.countplot(x ="SeniorCitizen",data = df,hue="Churn")
plt.title("Churn by SeniorCitizen",fontsize=9)
plt.show()
```



In [59]:
```
# Step 1: Calculate the percentage data
churn_data = df.groupby(['SeniorCitizen', 'Churn']).size().unstack().fillna(0)
churn_percent = churn_data.div(churn_data.sum(axis=1), axis=0) * 100
```

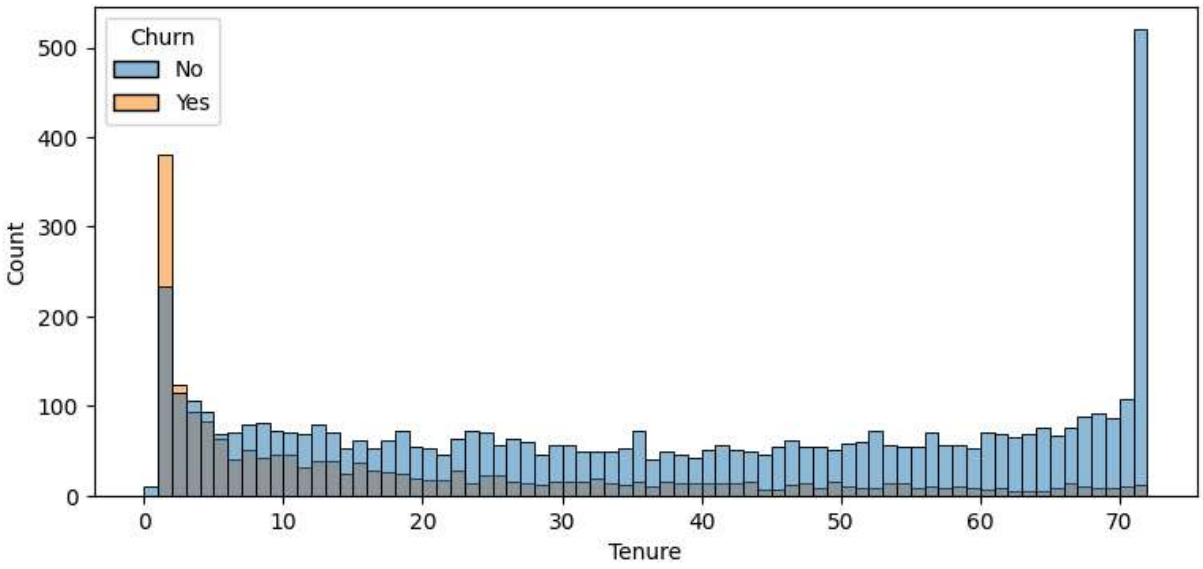
```
# Step 2: Plot stacked bar chart
plt.figure(figsize=(4, 4))

# Bottom bar (e.g., 'No')
plt.bar(churn_percent.index, churn_percent['No'], label='No', color='skyblue')

# Top bar (e.g., 'Yes') stacked on top
plt.bar(churn_percent.index, churn_percent['Yes'], bottom=churn_percent['No'], labe

# Add percentage text
for i in churn_percent.index:
    plt.text(i, churn_percent.loc[i, 'No']/2, f"{churn_percent.loc[i, 'No']:.1f}%",
    plt.text(i, churn_percent.loc[i, 'No'] + churn_percent.loc[i, 'Yes']/2, f"{chur

# Step 3: Styling
plt.title("Churn by SeniorCitizen", fontsize=9)
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.xticks([0, 1], ["No", "Yes"])  # Assuming 0 = Not Senior, 1 = Senior
plt.legend(title='Churn',bbox_to_anchor=(1,1))
plt.tight_layout()
plt.show()
```



comparatively a reater percenatge of people in SC category have churnedout

In [67]:
```
plt.figure(figsize=(9,4))
sns.histplot(x="Tenure", data =df,bins=72,hue="Churn")
```

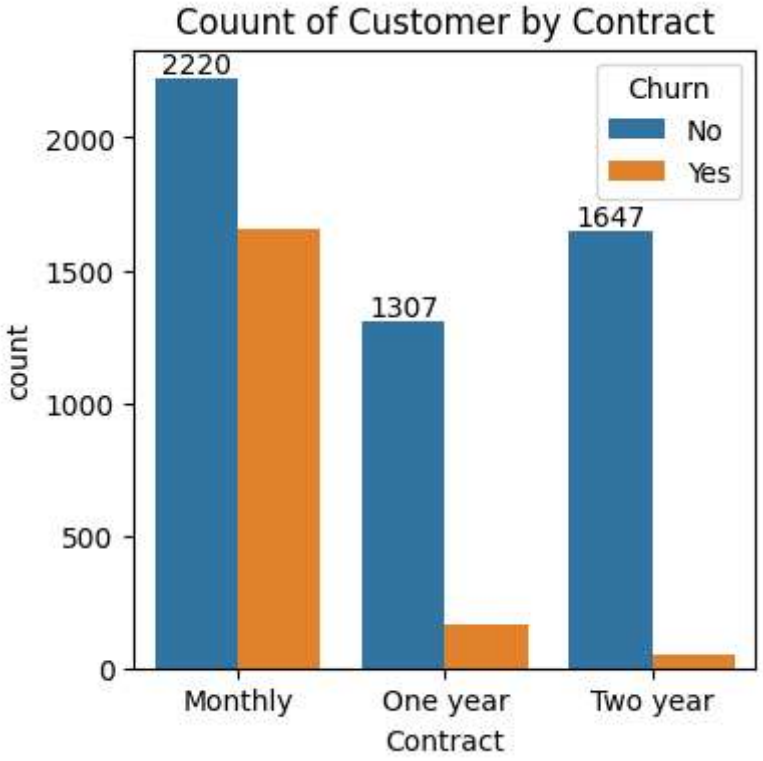
```
plt.show()
```



people who have used our services fro a long time have stayed and people who have used our services #1 or 2Month have churn out

usually the people are for short tenure have curned out

In [70]:
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x="Contract",data=df,hue="Churn")
ax.bar_label(ax.containers[0])
plt.title("Couunt of Customer by Contract")
plt.show()
```



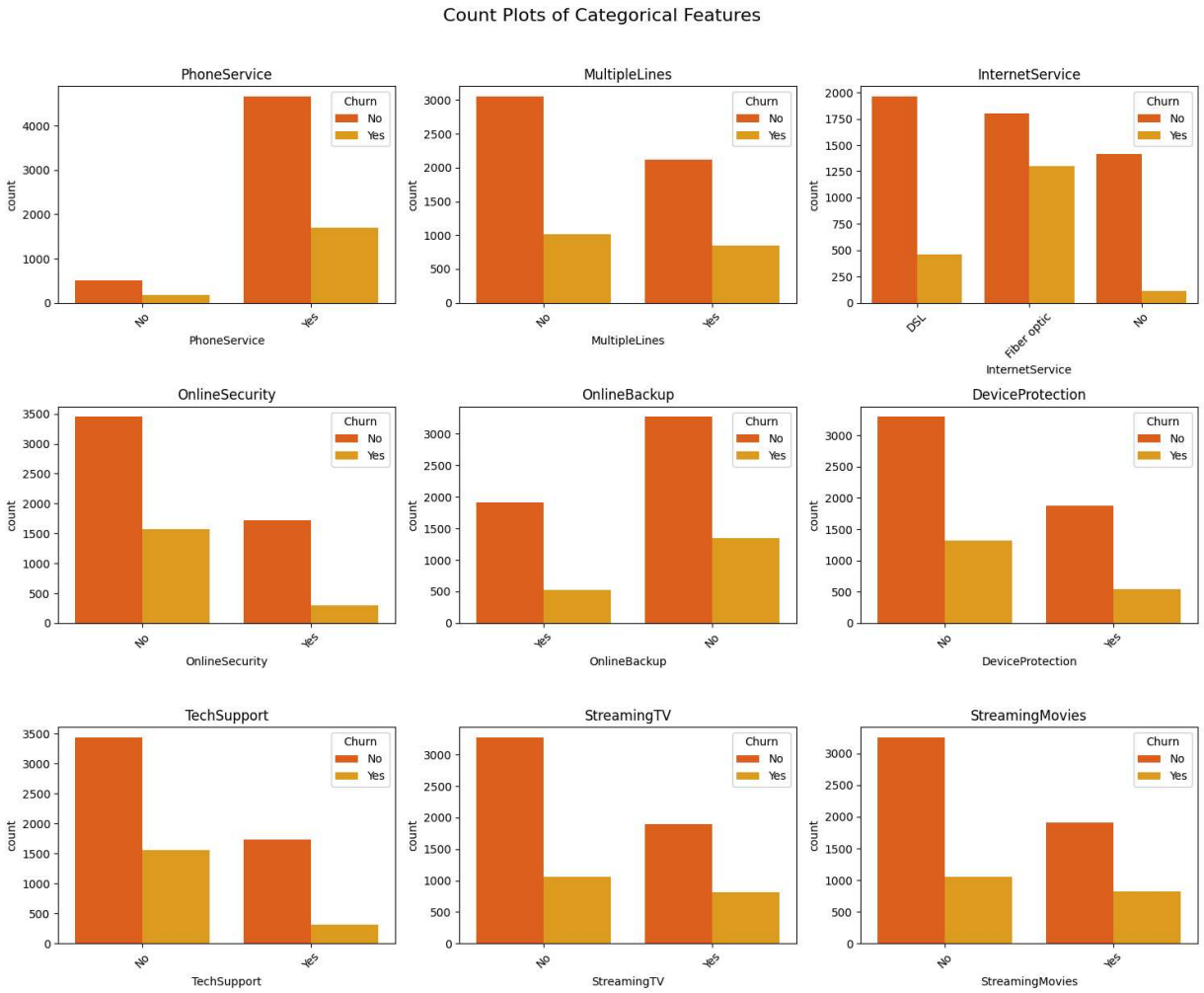
people who have month to month contract are likely to churn than those who have 1 or 2 year data

In [72]:
```
df.columns.values
```

Out[72]:
```
array(['customerID', 'Gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'Tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

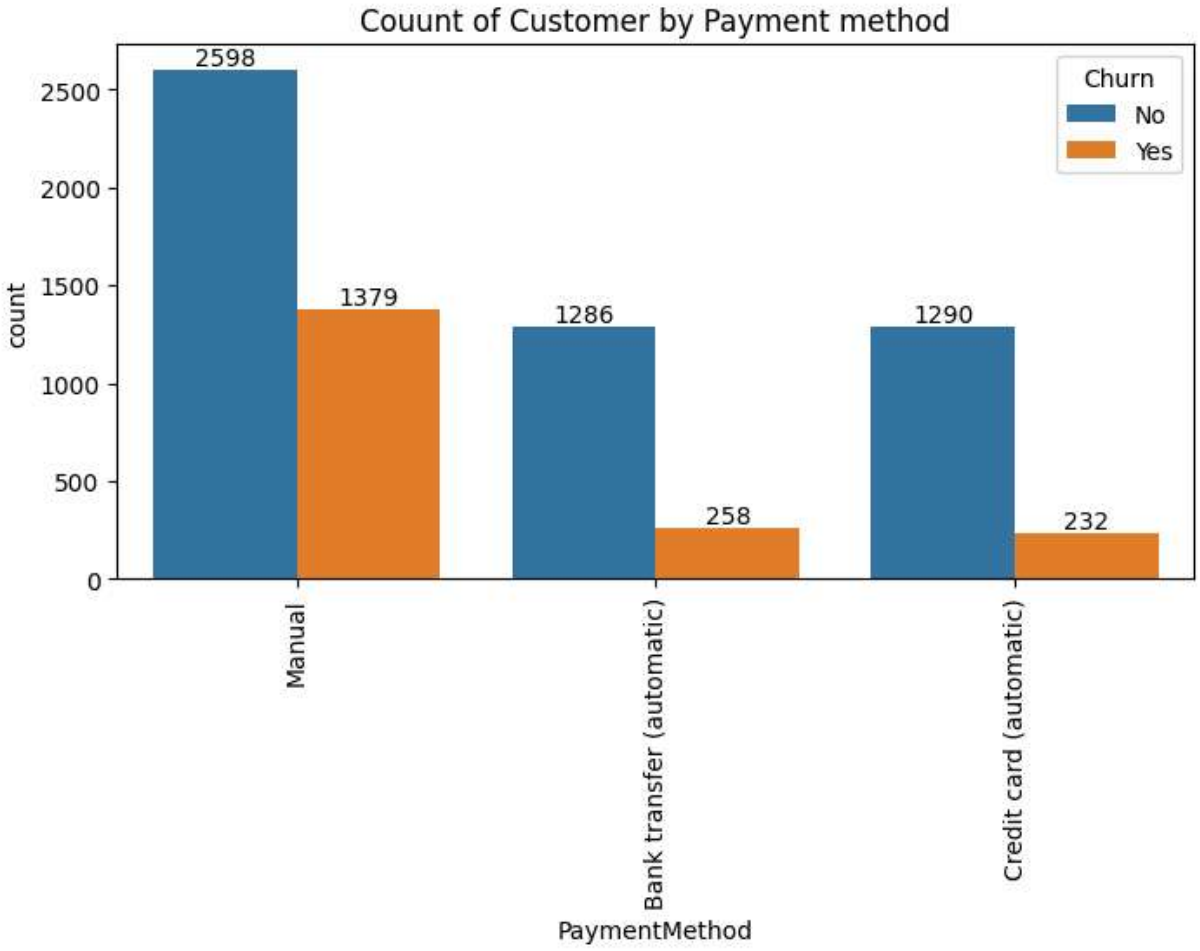
In [79]:
```
# List of features to plot
features = ['PhoneService', 'MultipleLines', 'InternetService',
            'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
            'TechSupport', 'StreamingTV', 'StreamingMovies']

# Setup subplot grid
plt.figure(figsize=(15, 12))
for i, feature in enumerate(features):
    plt.subplot(3, 3, i+1)  # 3 rows x 3 columns
    sns.countplot(data=df, x=feature, palette="autumn",hue="Churn")
    plt.title(feature)
    plt.xticks(rotation=45)
    plt.tight_layout()

plt.suptitle("Count Plots of Categorical Features", fontsize=16, y=1.02)
```

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



The count plots show that customers with No additional services like OnlineSecurity, TechSupport, DeviceProtection, and Streaming options tend to churn more than those who have them. Fiber optic users have a higher churn rate compared to DSL or no internet. While most customers have PhoneService, churn is relatively higher among those with MultipleLines. Overall, lacking optional services correlates with higher churn probability.

In [88]:
```
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("Couunt of Customer by Payment method")
plt.xticks(rotation =90)
plt.show()
```



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

Executive Summary – Customer Churn Analysis

This analysis investigates factors influencing customer churn, based on demographic, service usage, and contract features. The goal is to identify trends that can guide targeted retention strategies and reduce churn.

📌 Overall Churn Rate 26.54% of customers have churned from the company.

This highlights a substantial opportunity to improve customer retention and lifetime value.

👴 Senior Citizen Status 42% of senior citizens have churned, compared to 24% of non-senior citizens.

This suggests older customers may require more tailored communication, service support, or pricing incentives.

⏳ Tenure Impact Customers with 1–2 months of tenure exhibit a churn rate above 50%.

In contrast, customers with tenure > 60 months have a churn rate below 10%.

Long-term users show loyalty, indicating a strong case for onboarding support and early-stage engagement programs.

📄 Contract Type Month-to-month contract users have a churn rate of nearly 45%.

Customers on one-year and two-year contracts have significantly lower churn rates at around 11% and 3%, respectively.

This indicates that encouraging longer-term contracts can substantially reduce churn.

🌐 Internet Service Type Fiber optic users show a churn rate of approximately 42%, much higher than DSL users (~19%) and those with no internet (~7%).

The churn from fiber optic users could be related to cost, performance issues, or competition, warranting deeper investigation.

💳 Payment Method Customers using Electronic Check have a churn rate of 45%, which is the highest among all payment methods.

Those using credit cards or bank transfers have churn rates around 15–20%.

The high churn from electronic check users may be linked to user experience or financial behavior.

📡 Additional Services Influence Customers without value-added services are consistently more likely to churn:

Online Security: 33% churn without it vs. 11% with it.

Tech Support: 32% churn without vs. 10% with support.

Streaming TV/Movies: Higher churn rates among those who do not use these services.

Device Protection and Online Backup show similar trends.

This strongly suggests that upselling additional services can improve engagement and retention.

✅ Key Recommendations Focus on retaining new customers during the first 3 months with onboarding campaigns.

Offer incentives to move month-to-month customers to longer contracts.

Investigate and optimize fiber optic plans or service satisfaction.

Promote value-added services to improve stickiness.

Consider personalized retention plans for senior citizens.

Review and improve the electronic check payment experience or encourage alternative methods.

In [ ]: