In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv("c:/mydata/Customer Churn.csv")
df
```

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
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 [4]:
```
df.head()
```

Out[4]:

	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 [5]:
```
df.tail()
```

Out[5]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
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	

5 rows × 21 columns

In [6]:
```
df.describe()
```

Out[6]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

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

Out[7]:
```
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 [13]:
```
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   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
```

In [9]:
```python
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"
df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

converted o and 1 value senior citizen to yes/no to make it easier to understand.

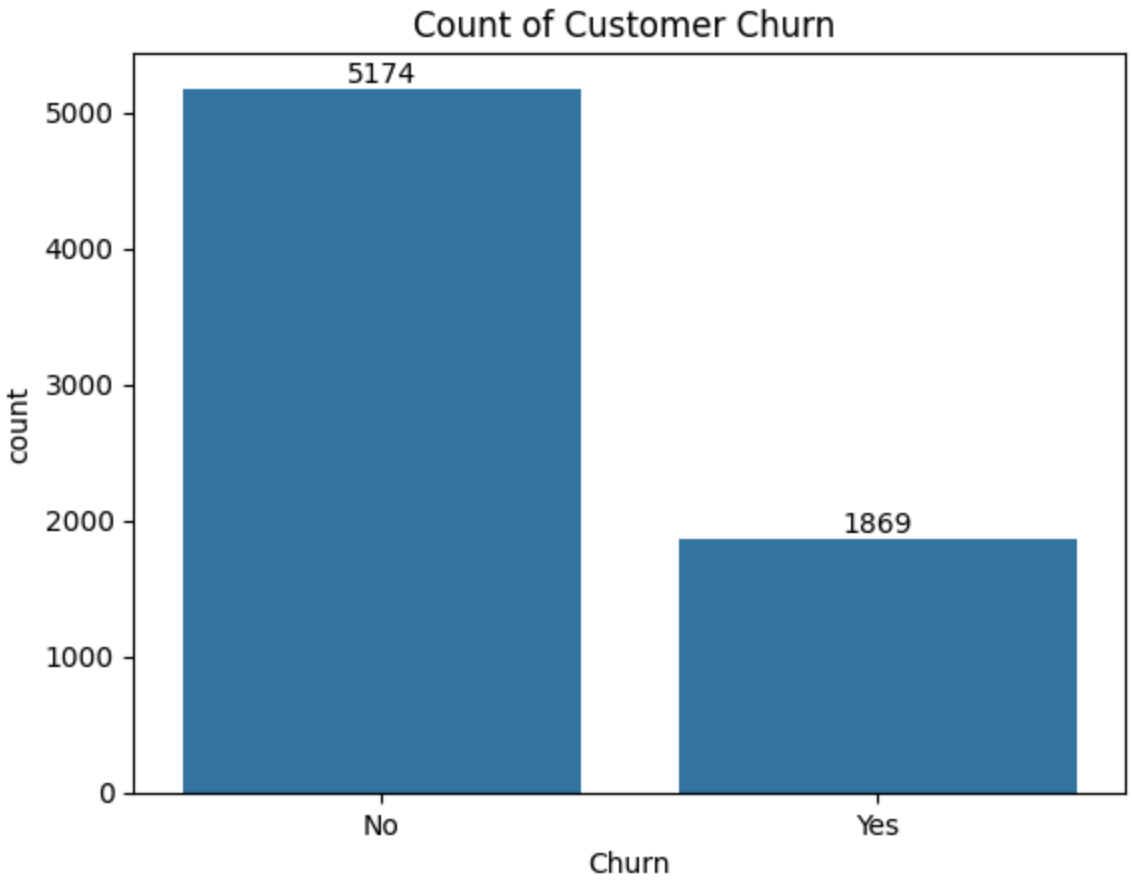
In [12]:
```python
df["TotalCharges"] = df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

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

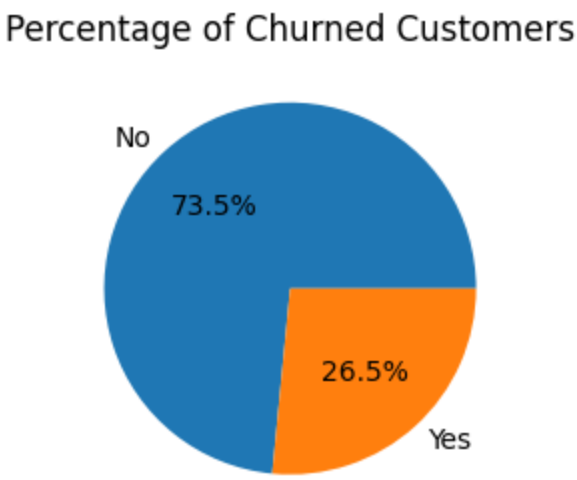
In [14]:
```python
df["Churn"].value_counts()
```

Out[14]:
```
Churn
No     5174
Yes    1869
Name: count, dtype: int64
```

In [28]:
```
ax = sns.countplot(x = "Churn", data = df)

ax.bar_label(ax.containers[0])
plt.title("Count of Customer Churn ")
plt.show()
```



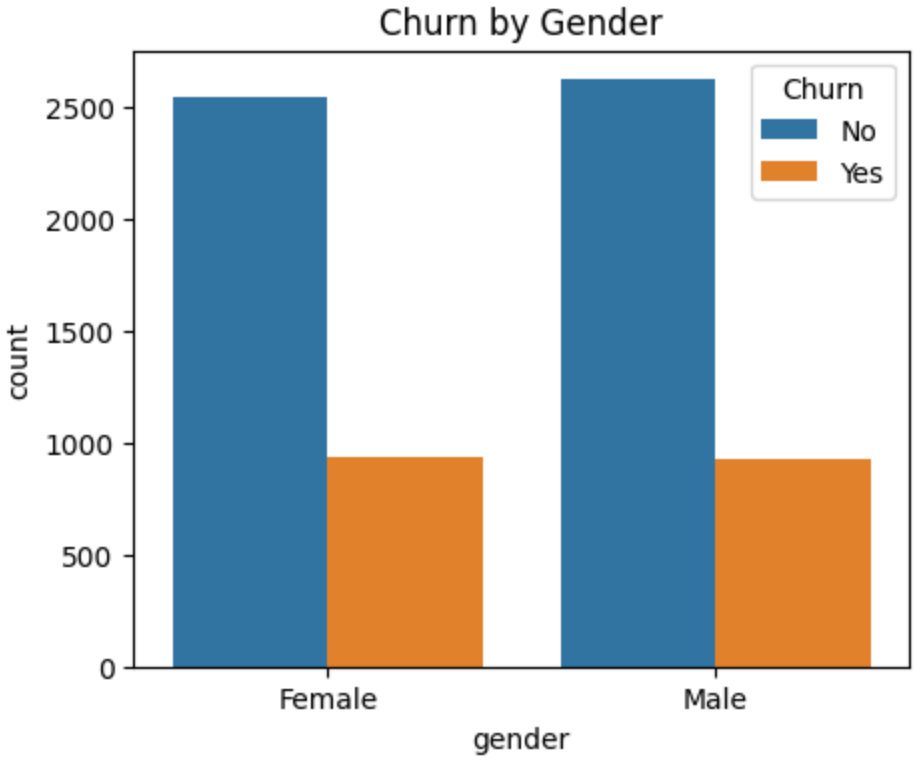
In [29]:
```
plt.figure(figsize=(3,4))
gb = df.groupby("Churn").agg({"Churn": "count"})

plt.pie(gb["Churn"], labels = gb.index, autopct = "%1.1f%%")
plt.title("Percentage of Churned Customers")
plt.show()
```



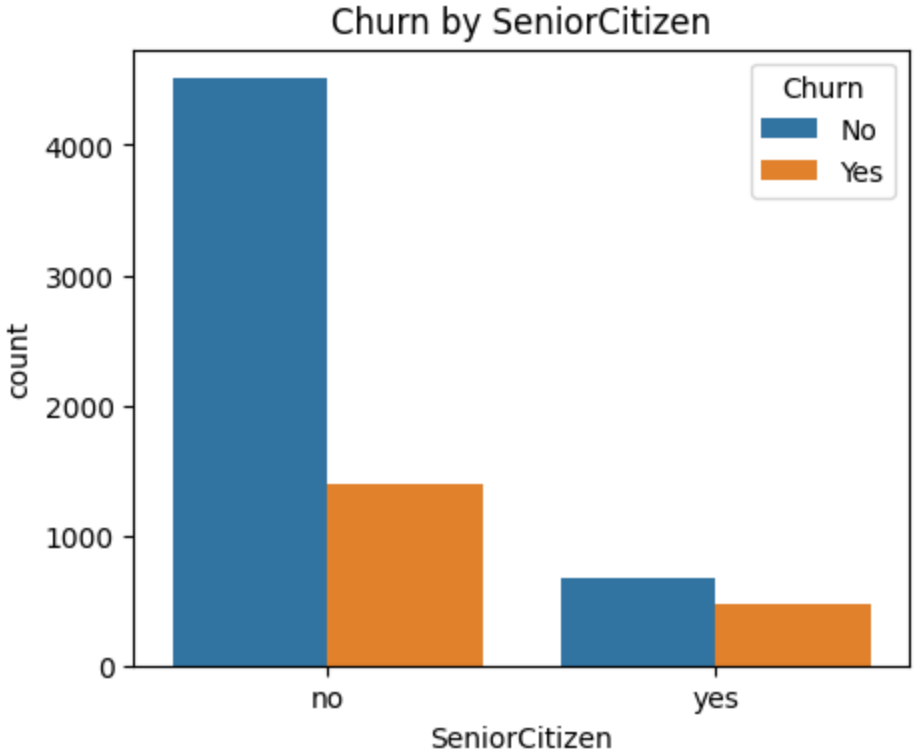
from the given pie chart we can conclude that 26.54% of our customers have churned ont

not let's explore the reason behind it

In [34]:
```
plt.figure(figsize=(5,4))
sns.countplot(x = "gender", data = df, hue = "Churn")
plt.title("Churn by Gender")
plt.show()
```

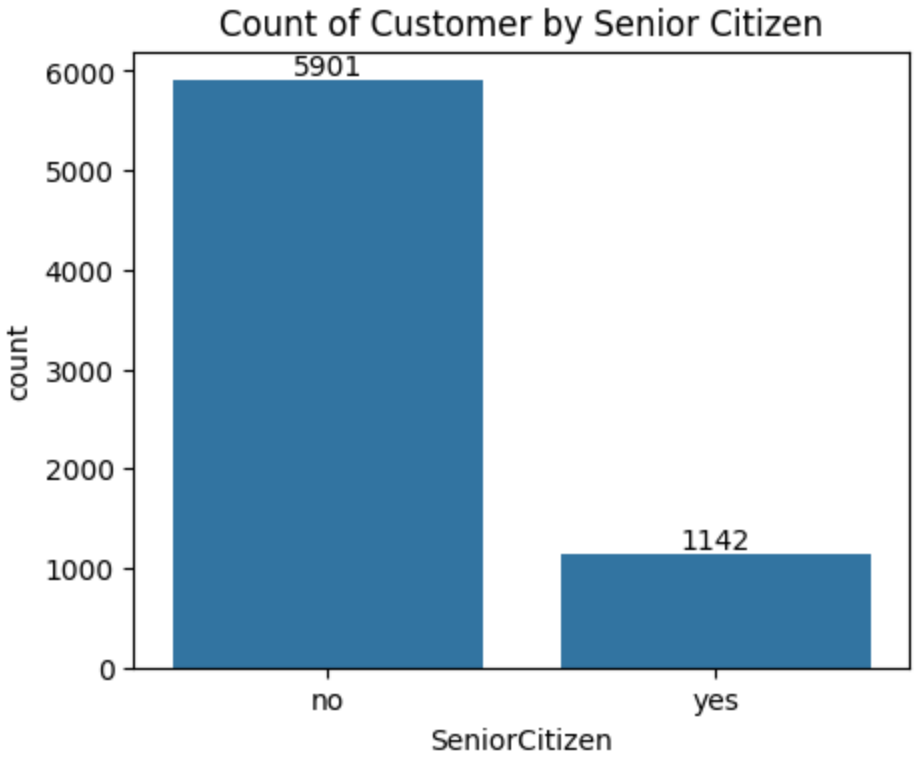


In [35]:
```
plt.figure(figsize=(5,4))
sns.countplot(x = "SeniorCitizen", data = df, hue = "Churn")
plt.title("Churn by SeniorCitizen")
plt.show()
```



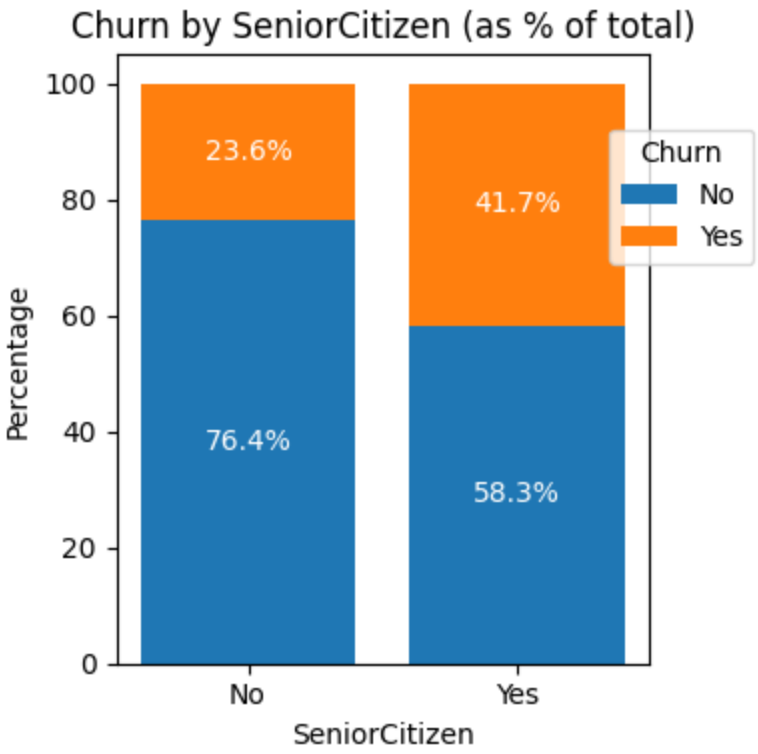
In [48]:
```
plt.figure(figsize=(5,4))
ax = sns.countplot(x = "SeniorCitizen", data = df)
ax.bar_label(ax.containers[0])
```

```
plt.title("Count of Customer by Senior Citizen")
plt.show()
```



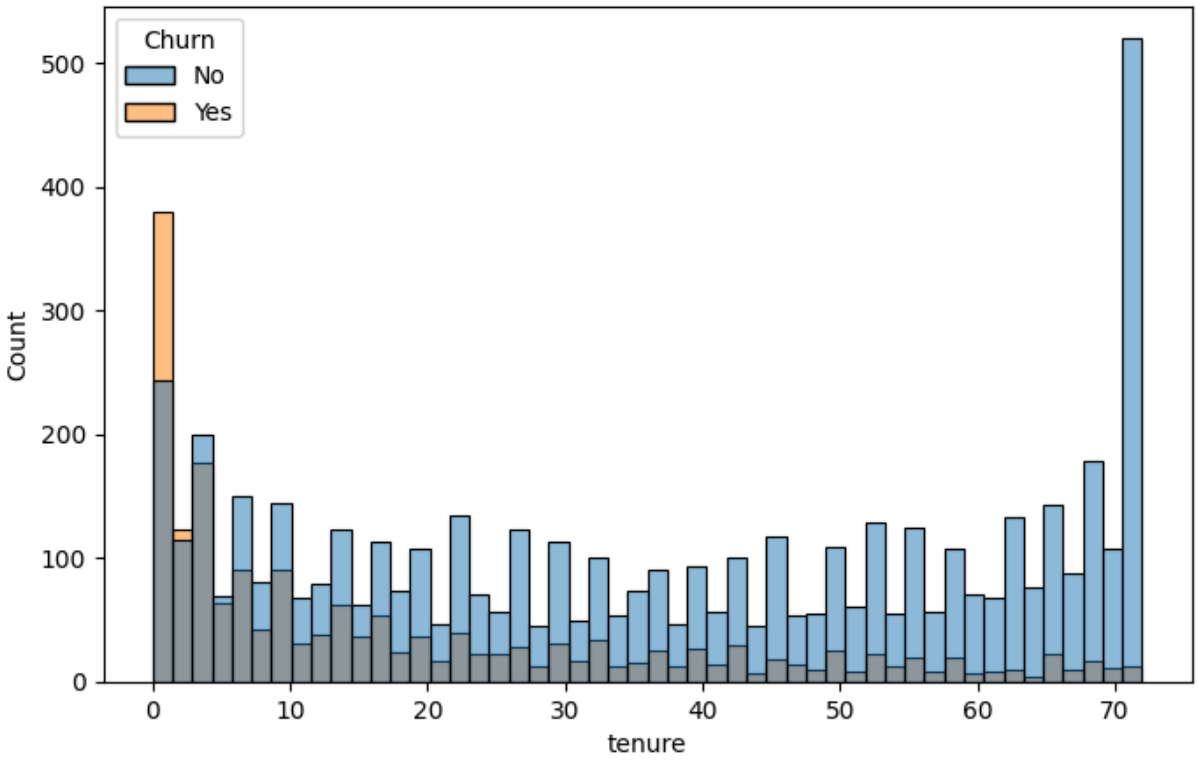
In [43]:
```
counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)
percentages = counts.div(counts.sum(axis=1), axis=0) * 100

# Plot
fig, ax = plt.subplots(figsize=(4,4))

# Stacked bar chart
bottoms = [0] * len(percentages)
for churn_category in percentages.columns:
    ax.bar(percentages.index, percentages[churn_category], bottom=bottoms, label=ch
    bottoms += percentages[churn_category]

# Add percentages as labels
for i, senior in enumerate(percentages.index):
    bottom = 0
    for churn_category in percentages.columns:
        value = percentages.loc[senior, churn_category]
        if value > 0:  # Only label non-zero values
            ax.text(i, bottom + value / 2, f"{value:.1f}%", ha="center", va="center
        bottom += value

# Customization
ax.set_title("Churn by SeniorCitizen (as % of total)")
ax.set_xlabel("SeniorCitizen")
ax.set_ylabel("Percentage")
ax.set_xticks(percentages.index)
ax.set_xticklabels(["No", "Yes"])  # Assuming 0 = No, 1 = Yes
ax.legend(title="Churn", bbox_to_anchor = (0.9,0.9))
```

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



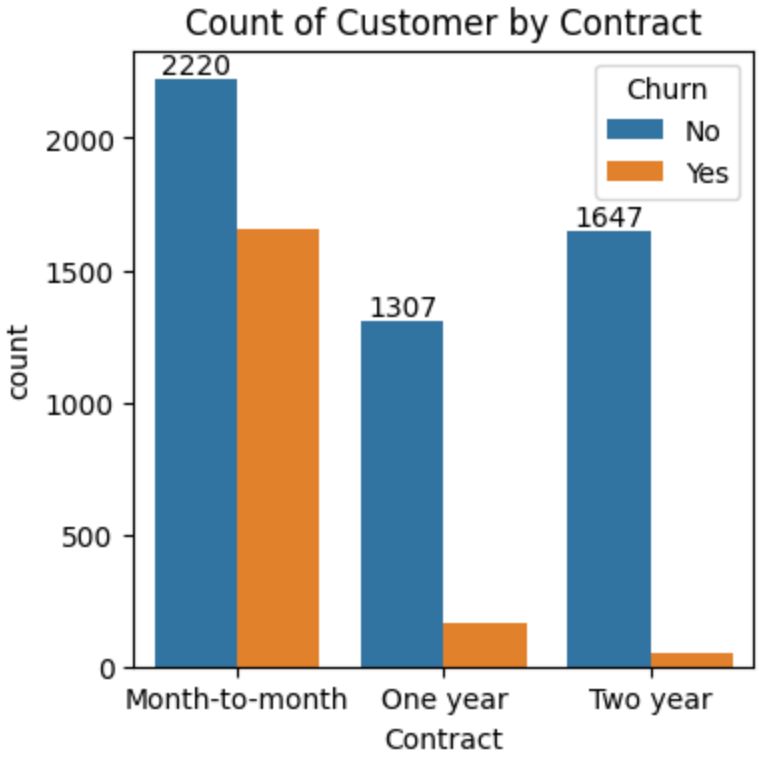
comparative a greated percentage of people in senior citizen category have churned

In [47]:
```
plt.figure(figsize=(8,5))
sns.histplot(x = "tenure", data = df, hue = "Churn", bins = 50)
plt.show()
```



people who have used our servies for a long time have stayed and people who have used our servieces 1 and 2 months

In [51]:
```
plt.figure(figsize=(4,4))
ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of Customer by Contract")
plt.show()
```



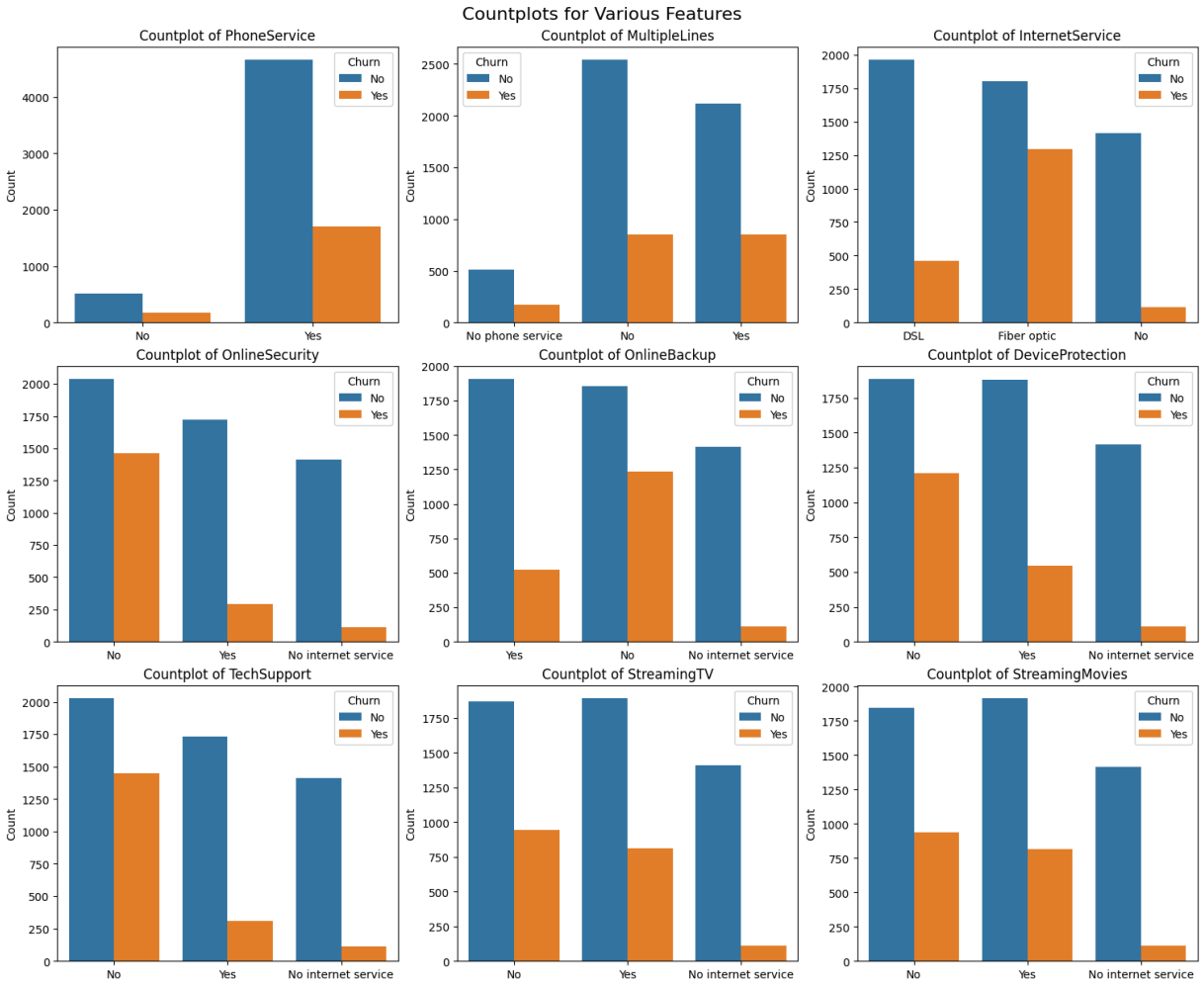
people who have month to month contract are likely to churn then from those who have 1 or 2 years of contract

In [52]:
```
df.columns.values
```

Out[52]:
```
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 [56]:
```
columns = [
    'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMov
]

# Number of subplots (3 rows x 3 columns for 9 features)
n_cols = 3
n_rows = -(-len(columns) // n_cols)  # Ceiling division for rows

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 12), constrained_layout=True)

# Flatten axes for easier indexing
axes = axes.flatten()
```

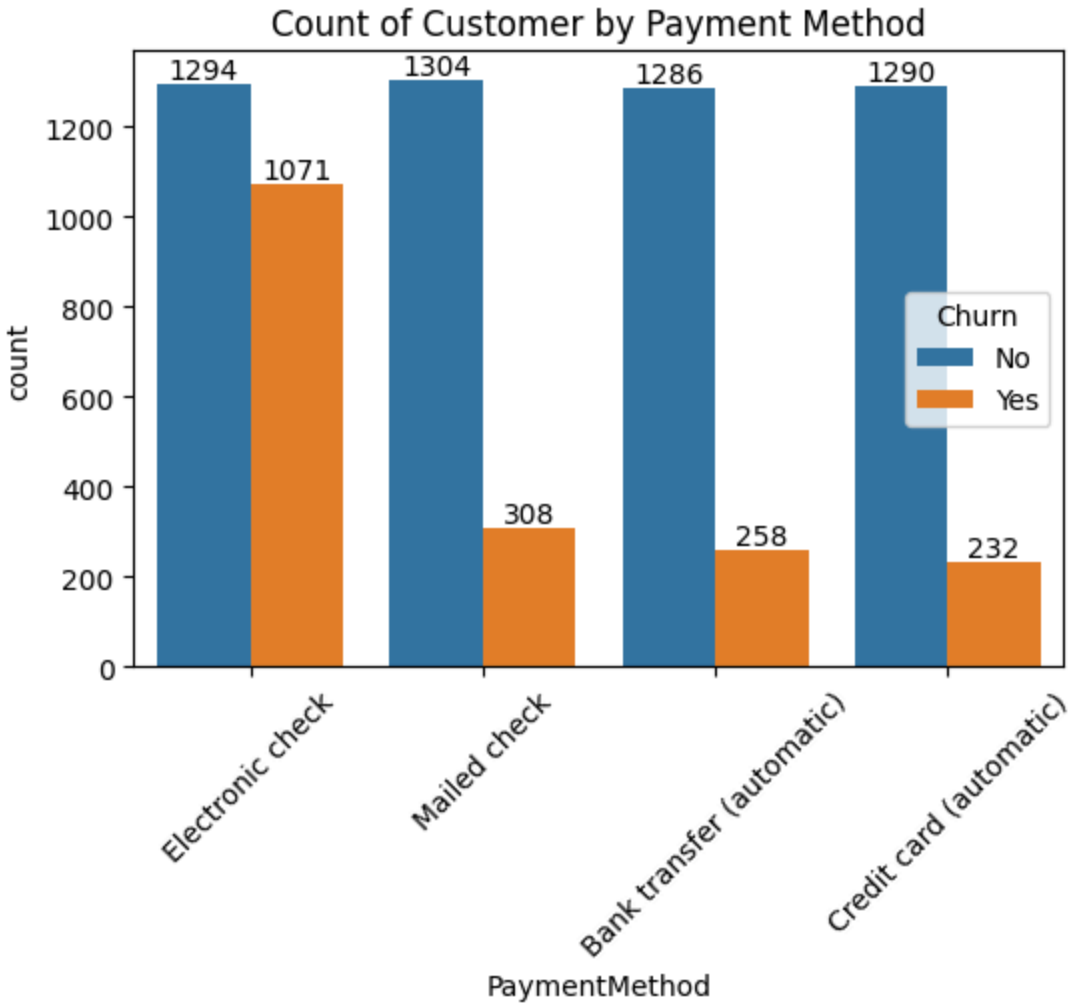
```python
# Loop through columns and plot
for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue = "Churn")
    axes[i].set_title(f'Countplot of {col}')
    axes[i].set_xlabel('')  # Remove x-axis label for cleaner appearance
    axes[i].set_ylabel('Count')

# Hide any unused subplots if the grid is larger than the number of columns
for j in range(len(columns), len(axes)):
    axes[j].set_visible(False)

# Display the plots
plt.suptitle('Countplots for Various Features', fontsize=16, y=1.02)
plt.show()
```



The majority of customers who do nt churn tend to have services like phoneService, InternetService(Particularly DSL), and OnlineSecurity enbled

OnlineBackup, Techsupport, and StreamingTv, churn rates are noticeably higher when these services are not used or are unavailable

In [61]:
```
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.title("Count of Customer by Payment Method")
plt.xticks(rotation = 45)
plt.show()
```



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

In [ ]:
```
```