In [27]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [28]:
```
df=pd.read_csv("Customer Churn.csv")
df
```

Out[28]:

	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 [29]:
```
df.head()
```

Out[29]:

	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

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

In [30]:
```
df['TotalCharges']=df['TotalCharges'].replace(" ","0")
df['TotalCharges']=df['TotalCharges'].astype("float")
```

In [31]:
```
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 [32]:
```
df.isnull().sum().sum()
```

Out[32]:
```
np.int64(0)
```

In [33]:
```
df.describe()
```

Out[33]:

	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 [34]:
```
df.duplicated().sum()
```

Out[34]:
```
np.int64(0)
```

In [35]:
```
df["customerID"].duplicated().sum()
```

Out[35]:
```
np.int64(0)
```

In [36]:
```
def conv(value):
    if value==1:
        return "Yes"
    else:
        return "No"

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

converted 0 and 1 values of senior citizen to yes/no to make it easier to understand

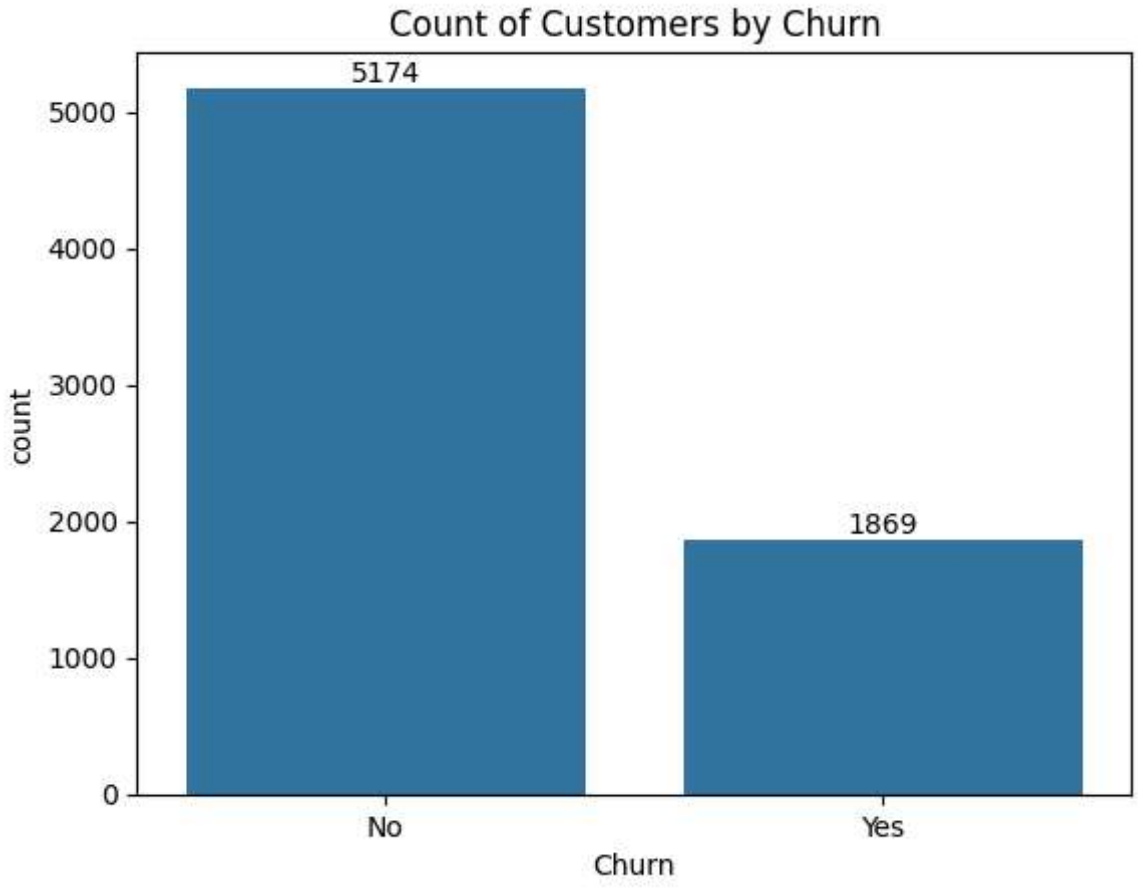
In [37]:
```
df.head()
```

Out[37]:

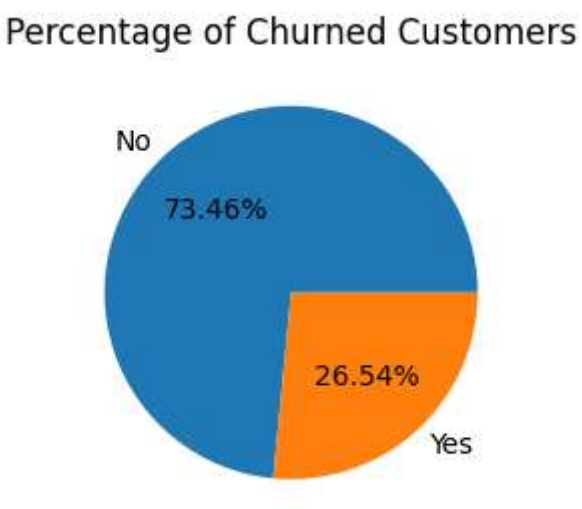
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Multipl
0	7590-VHVEG	Female	No	Yes	No	1	No	No
1	5575-GNVDE	Male	No	No	No	34	Yes	
2	3668-QPYBK	Male	No	No	No	2	Yes	
3	7795-CFOCW	Male	No	No	No	45	No	No
4	9237-HQITU	Female	No	No	No	2	Yes	

5 rows × 21 columns

In [38]:
```
ax=sns.countplot(x='Churn',data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn")
plt.show()
```



In [79]:
```
# 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 Churned Customers")
# plt.show()
# gb
import matplotlib.pyplot as plt

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 Churned Customers")
plt.show()
gb
```



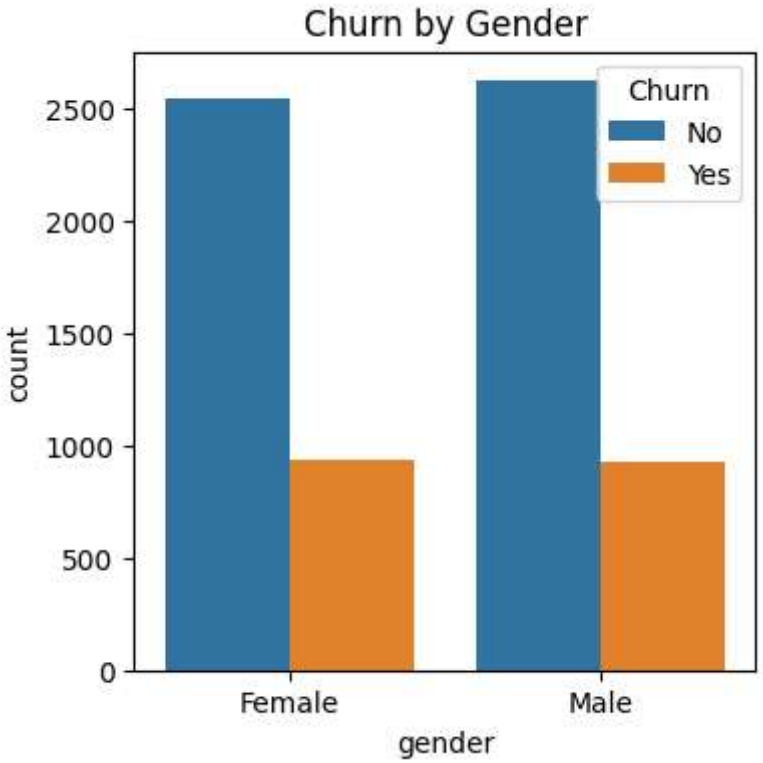
Out[79]:

Churn	
No	5174
Yes	1869

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

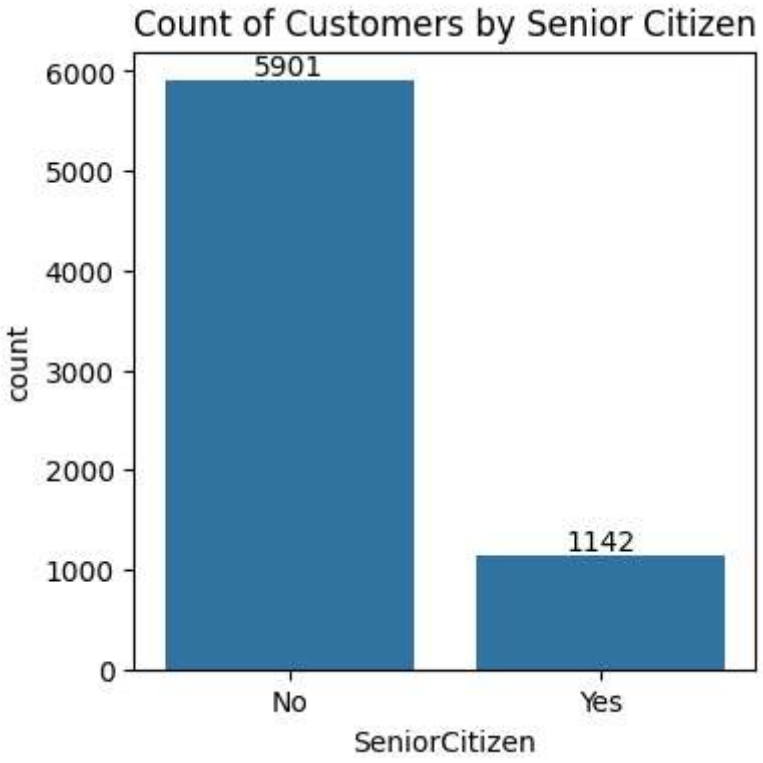
now let's explore the reason behind it

In [80]:
```
plt.figure(figsize=(4,4))
sns.countplot(x="gender", data =df, hue='Churn')
plt.title("Churn by Gender")
plt.show()
```

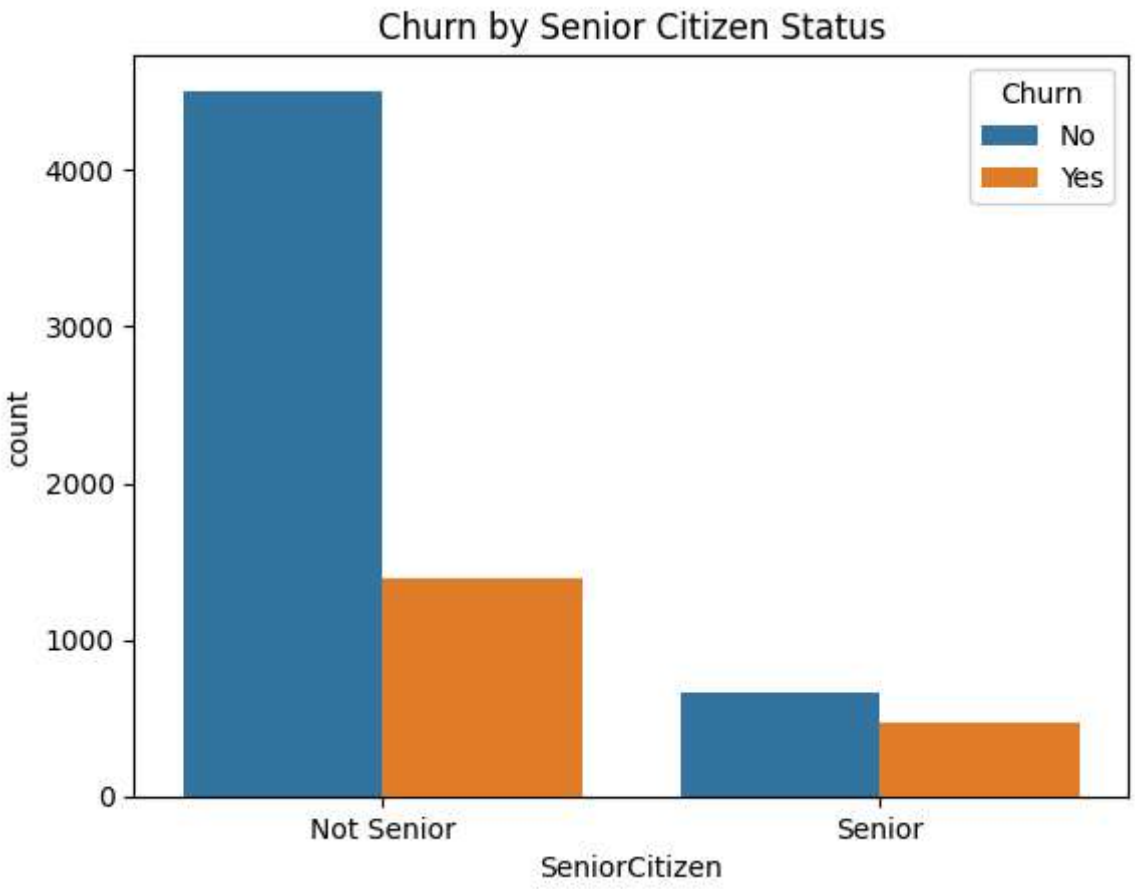


In [ ]:

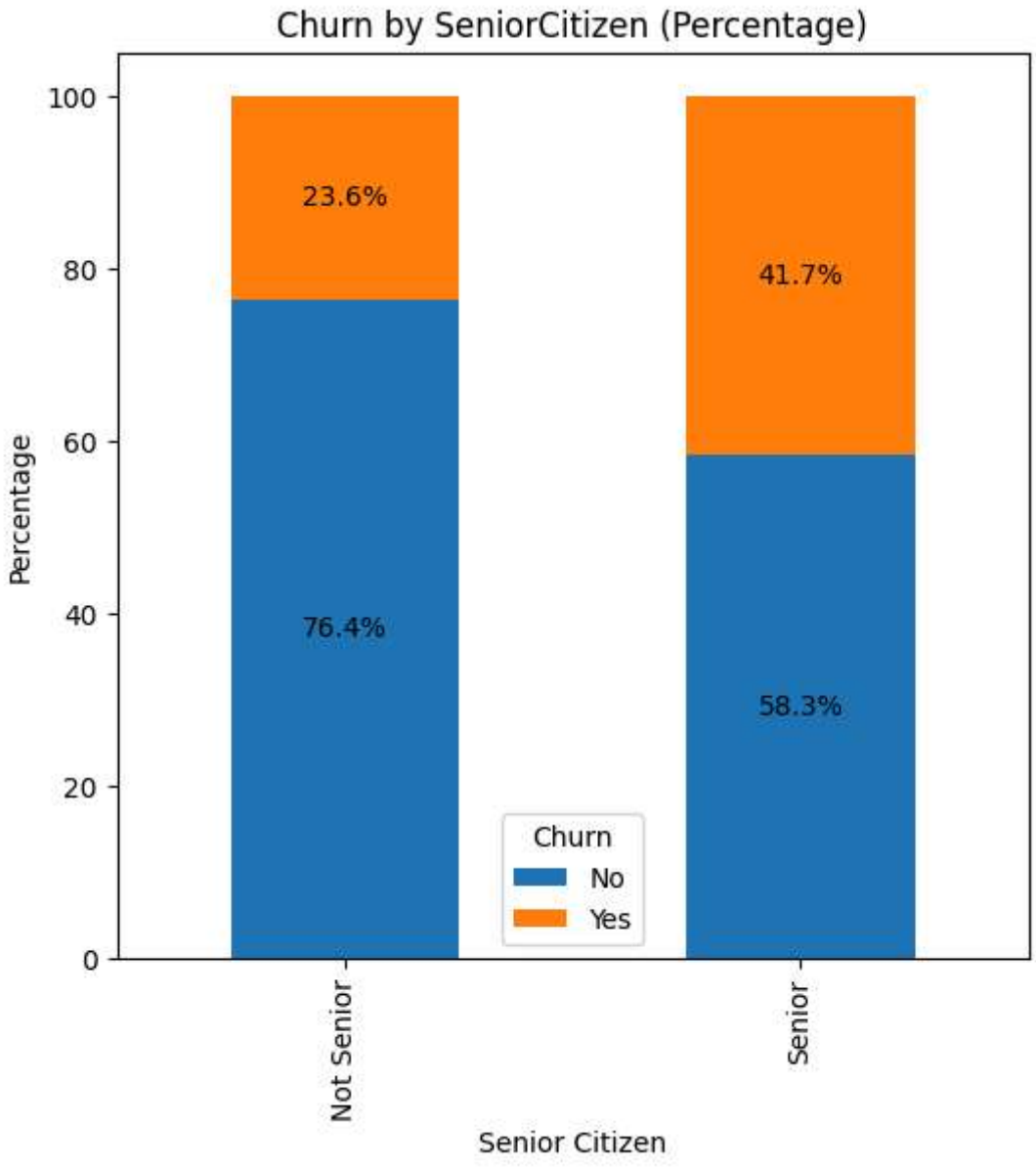
In [81]:
```
plt.figure(figsize = (4,4))
ax= sns.countplot(x = "SeniorCitizen", data = df)
ax.bar_label(ax.containers [0])
plt.title("Count of Customers by Senior Citizen")
plt.show()
```



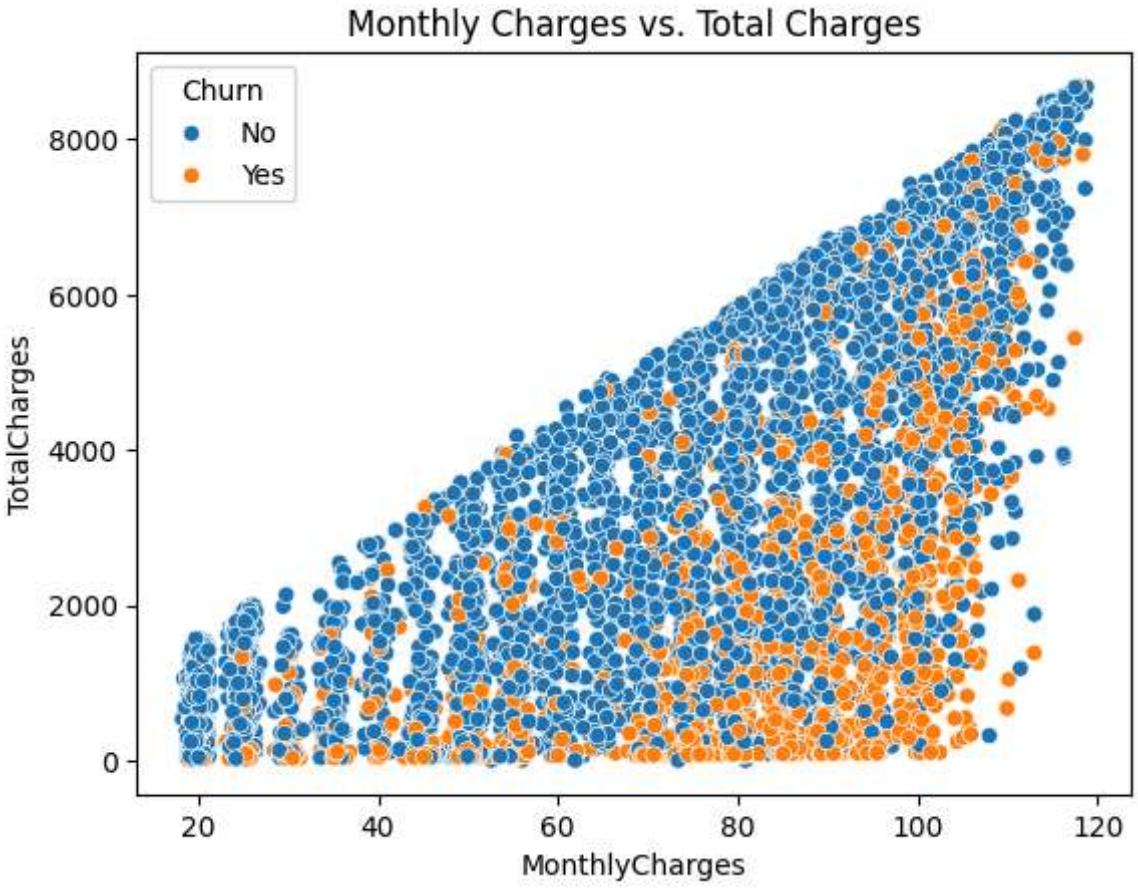
In [82]:
```
sns.countplot(data=df, x='SeniorCitizen', hue='Churn')
plt.title("Churn by Senior Citizen Status")
plt.xticks([0, 1], ['Not Senior', 'Senior'])
plt.show()
```



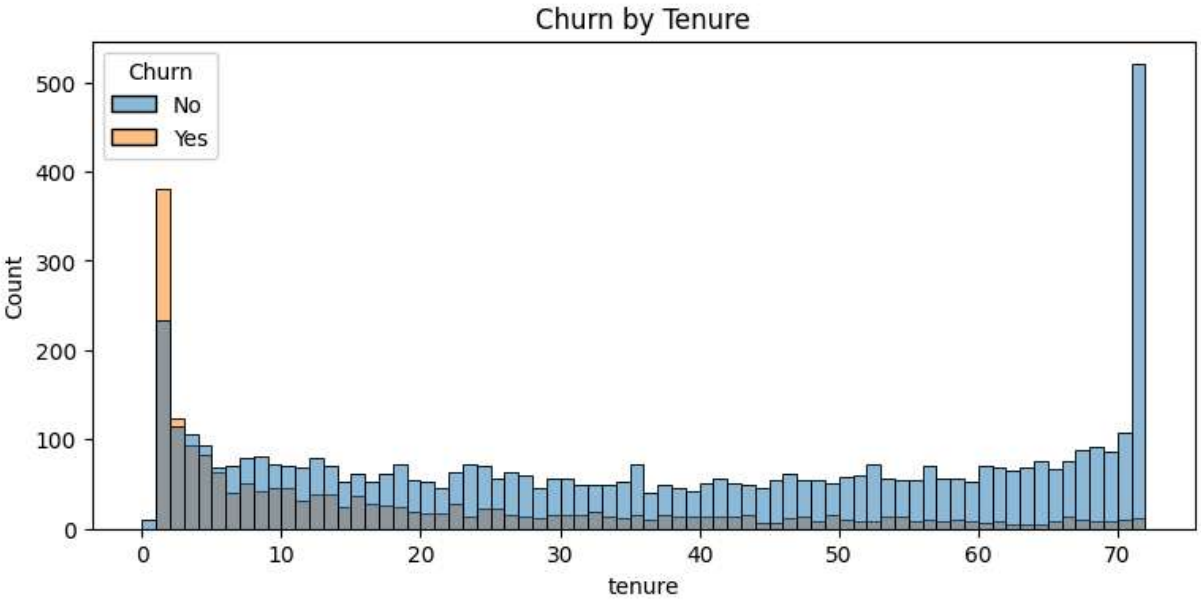
In [83]:
```
# Create a crosstab for SeniorCitizen vs Churn with percentages
count_data = pd.crosstab(df["SeniorCitizen"], df["Churn"], normalize='index')  * 10

# Plot stacked bar chart
ax = count_data.plot(kind='bar', stacked=True, figsize=(6, 6))

# Add percentage labels to the bars
for p in ax.patches:
    height = p.get_height()
    if height > 0:
        ax.annotate(f'{height:.1f}%',
                    xy=(p.get_x() + p.get_width() / 2, p.get_y() + height / 2),
                    xytext=(0, 0),
                    textcoords='offset points',
                    ha='center', va='center')

plt.title("Churn by SeniorCitizen (Percentage)")
plt.ylabel("Percentage")
plt.xlabel("Senior Citizen")
plt.xticks([0, 1], ['Not Senior', 'Senior'])
plt.show()
```



In [86]:
```
sns.scatterplot(data=df, x='MonthlyCharges', y='TotalCharges', hue='Churn')
plt.title("Monthly Charges vs. Total Charges")
plt.show()
```



In [87]:
```
plt.figure(figsize=(9,4))
sns.histplot(x = "tenure", data = df, bins=72 , hue= 'Churn')
plt.title("Churn by Tenure")
plt.show()
```



people who have used our services for a long time have stayed and people who

have used our servies 1 or months have churned

In [88]:
```
plt.figure(figsize = (4,4))
ax = sns.countplot(x = "Contract", data = df, hue='Churn')
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract")
plt.show()
```

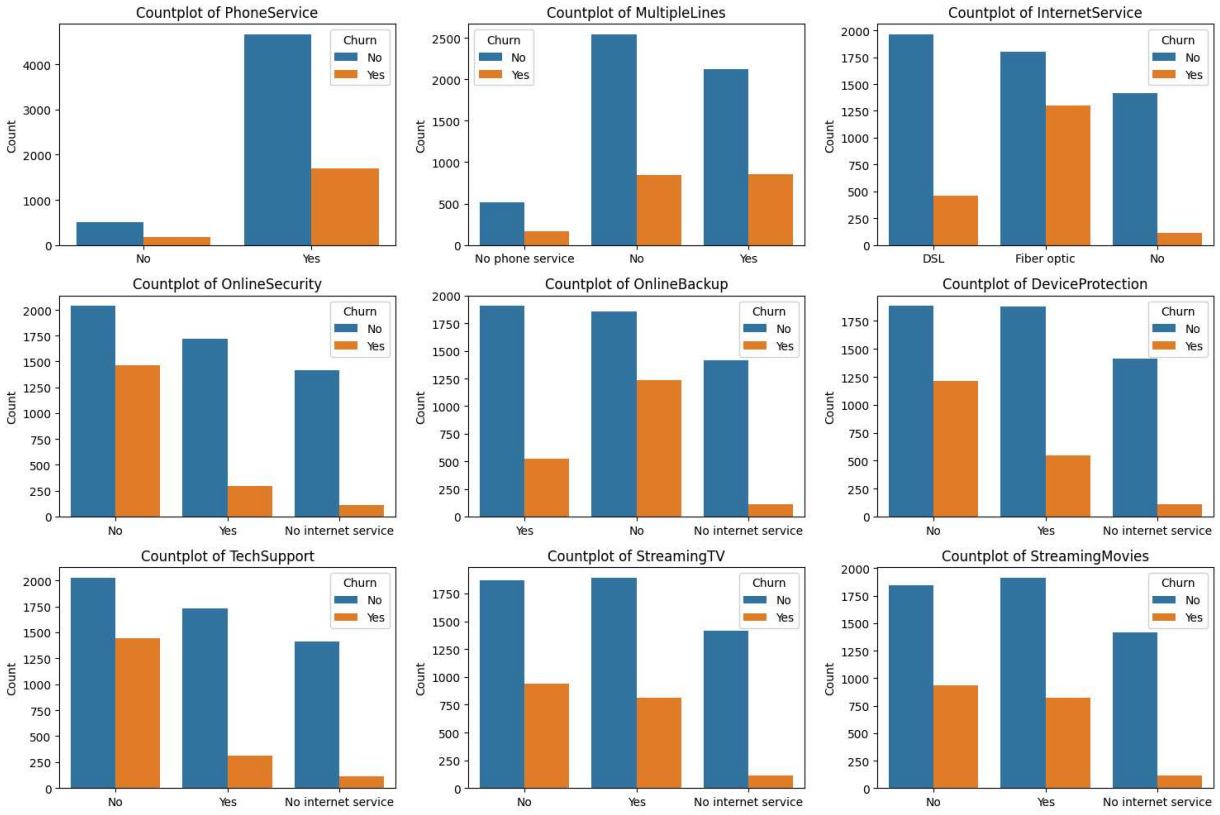
people who have month to month contract are likely to churn than from those who have 1 or 2 years of contract

In [89]:
```
df.columns.values
```

Out[89]:
```
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 [90]:
```
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV', 'StreamingMovies']

# Number of rows and columns for the subplots
```

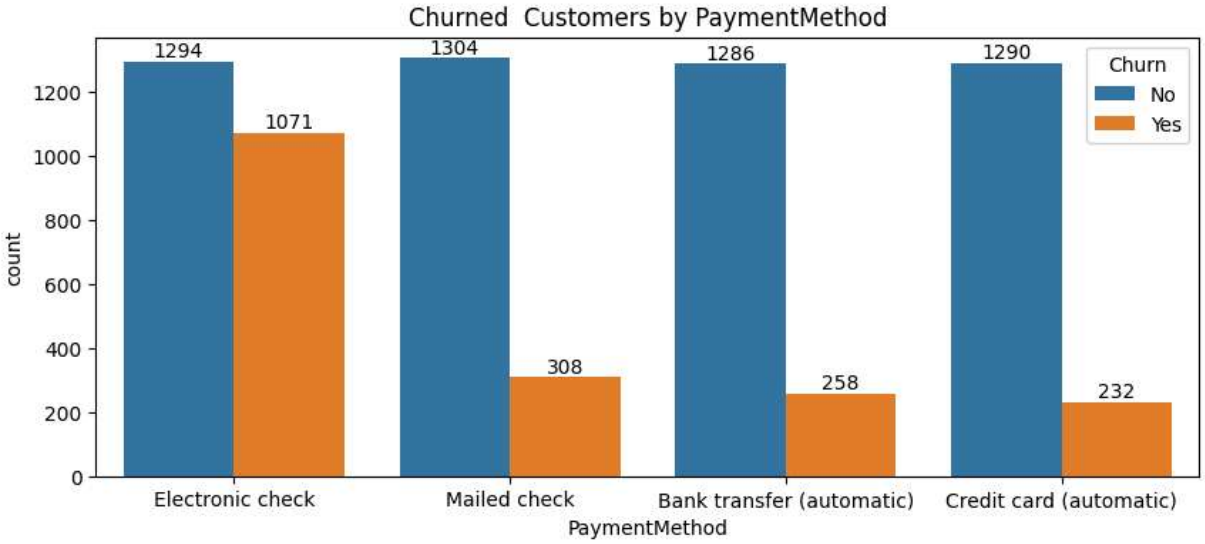
```
n_cols = 3  # Number of columns in subplot grid
n_rows = -(-len(columns) // n_cols)  # Compute rows needed

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))
axes = axes.flatten()  # Flatten the axes array for easy iteration

# Generate a countplot for each column
for i, col in enumerate(columns):
    sns.countplot(data=df, x=col, ax=axes[i], hue=df['Churn'])
    axes[i].set_title(f"Countplot of {col}")
    axes[i].set_xlabel("")
    axes[i].set_ylabel("Count")

# Hide any extra subplot axes
for j in range(len(columns), len(axes)):
    fig.delaxes(axes[j])

# Adjust layout
plt.tight_layout()
plt.show()
```



The majority of customers who do not churn tend to have services like PhoneService, InternetService (particularly DSL), and OnlineSecurity enabled. For services like OnlineBackup, TechSupport, and StreamingTV, churn rates are noticeably higher when these services are not used or are unavailable

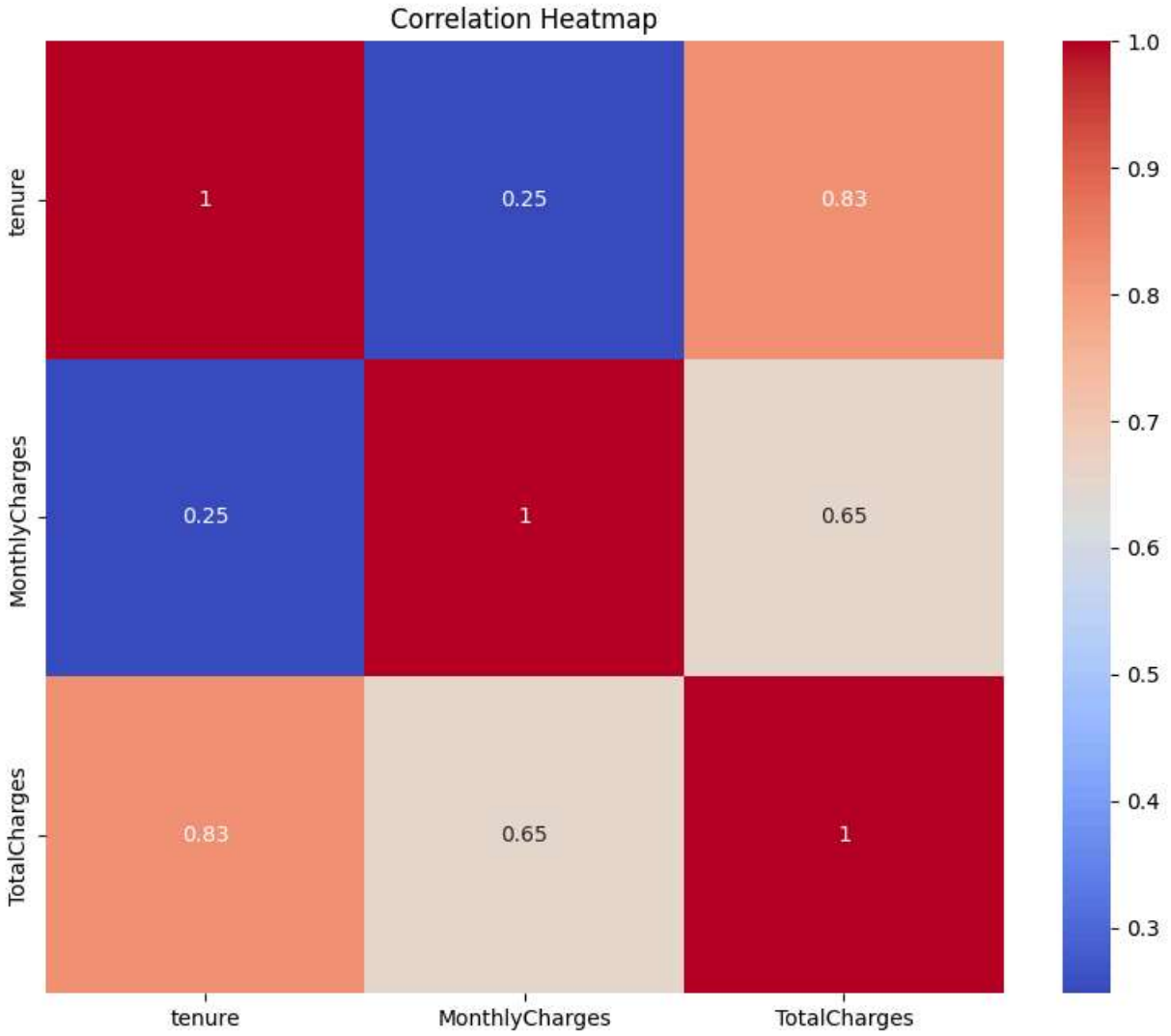
In [91]:
```
plt.figure(figsize = (10,4))
ax = sns.countplot(x = "PaymentMethod", data = df, hue='Churn')
ax.bar_label(ax.containers[0])
```

```
ax.bar_label(ax.containers [1])
plt.title("Churned  Customers by PaymentMethod")
plt.show()
```



Customers are likely to churn when they were using Eloctronic Check as their payment method

In [92]:
```
# Filter only numeric columns
numeric_df = df.select_dtypes(include=['number'])

# Create the heatmap on only the numeric data
plt.figure(figsize=(10, 8))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```


Correlation Heatmap
tenure
MonthlyCharges
TotalCharges
1
0.25
0.83
0.25
1
0.65
0.83
0.65
1
tenure
MonthlyCharges
TotalCharges
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3