In [1]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /opt/anaconda3/lib/python3.13/site-pac
kages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in /opt/anaconda3/lib/python3.13/s
ite-packages (from pandas) (2.1.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/pyt
hon3.13/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.13/si
te-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.13/
site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.13/site-p
ackages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in /opt/anaconda3/lib/python3.13/site-pack
ages (2.1.3)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /opt/anaconda3/lib/python3.13/sit
e-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.13/si
te-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/python
3.13/site-packages (from matplotlib) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/lib/python
3.13/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /opt/anaconda3/lib/python3.13/sit
e-packages (from matplotlib) (2.1.3)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.13/site-
packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/lib/pytho
n3.13/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.13/site-p
ackages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in /opt/anaconda3/lib/python3.13/site-pa
ckages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /opt/anaconda3/lib/pytho
n3.13/site-packages (from seaborn) (2.1.3)
Requirement already satisfied: pandas>=1.2 in /opt/anaconda3/lib/python3.13/sit
e-packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /opt/anaconda3/lib/py
thon3.13/site-packages (from seaborn) (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.13/si
te-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/python
3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/lib/python
3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.2)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.13/site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/lib/python3.1
3/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/lib/pytho
n3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.13/si
te-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.13/
site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.13/site-p
ackages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

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

In [9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('/Users/yashyadav/Downloads/Customer Churn.csv')
df.head(10)
```

Out[9]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneServ
0	7590-VHVEG	Female	0	Yes	No	1	
1	5575-GNVDE	Male	0	No	No	34	
2	3668-QPYBK	Male	0	No	No	2	
3	7795-CFOCW	Male	0	No	No	45	
4	9237-HQITU	Female	0	No	No	2	
5	9305-CDSKC	Female	0	No	No	8	
6	1452-KIOVK	Male	0	No	Yes	22	
7	6713-OKOMC	Female	0	No	No	10	
8	7892-POOKP	Female	0	Yes	No	28	
9	6388-TABGU	Male	0	No	Yes	62	

10 rows × 21 columns

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

In [13]:
```
df["TotalCharges"]=df["TotalCharges"].astype("float")
```

In [14]:
```
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 [17]:
```
df.isnull().sum().sum()
```

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

In [18]:
```
df.describe()
```

Out[18]:

	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 [21]:
```
df.duplicated().sum()
```

Out[21]: np.int64(0)

In [22]:
```
df["customerID"].duplicated().sum()
```

Out[22]: np.int64(0)

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

In [23]:
```
def conv(value):
    if value==1:
        return "yes"
    else:
        return "no"
df["SeniorCitizen"]=df["SeniorCitizen"].apply(conv)
```

In [24]:
```
df.head(15)
```

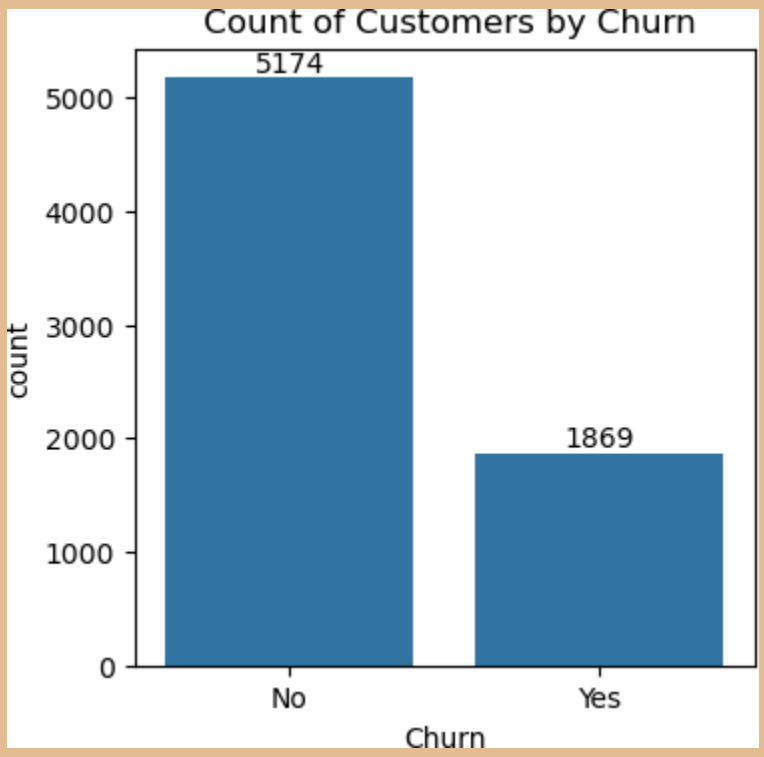
Out[24]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneSer
0	7590-VHVEG	Female	no	Yes	No	1	
1	5575-GNVDE	Male	no	No	No	34	
2	3668-QPYBK	Male	no	No	No	2	
3	7795-CFOCW	Male	no	No	No	45	
4	9237-HQITU	Female	no	No	No	2	
5	9305-CDSKC	Female	no	No	No	8	
6	1452-KIOVK	Male	no	No	Yes	22	
7	6713-OKOMC	Female	no	No	No	10	
8	7892-POOKP	Female	no	Yes	No	28	
9	6388-TABGU	Male	no	No	Yes	62	
10	9763-GRSKD	Male	no	Yes	Yes	13	
11	7469-LKBCI	Male	no	No	No	16	
12	8091-TTVAX	Male	no	Yes	No	58	
13	0280-XJGEX	Male	no	No	No	49	
14	5129-JLPIS	Male	no	No	No	25	

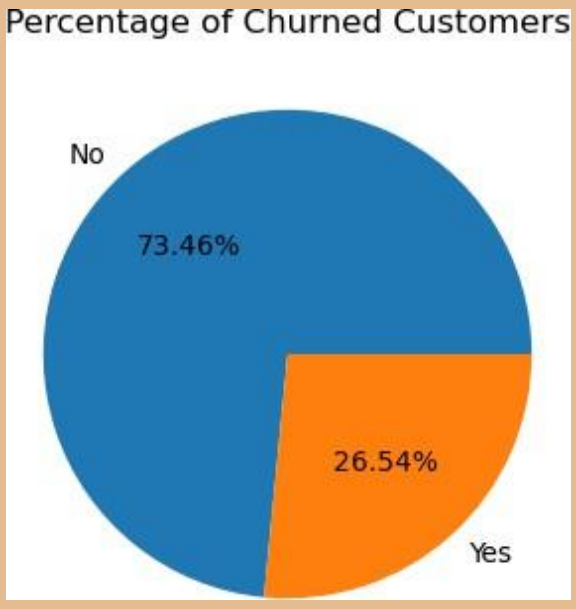
15 rows × 21 columns

In [36]:

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

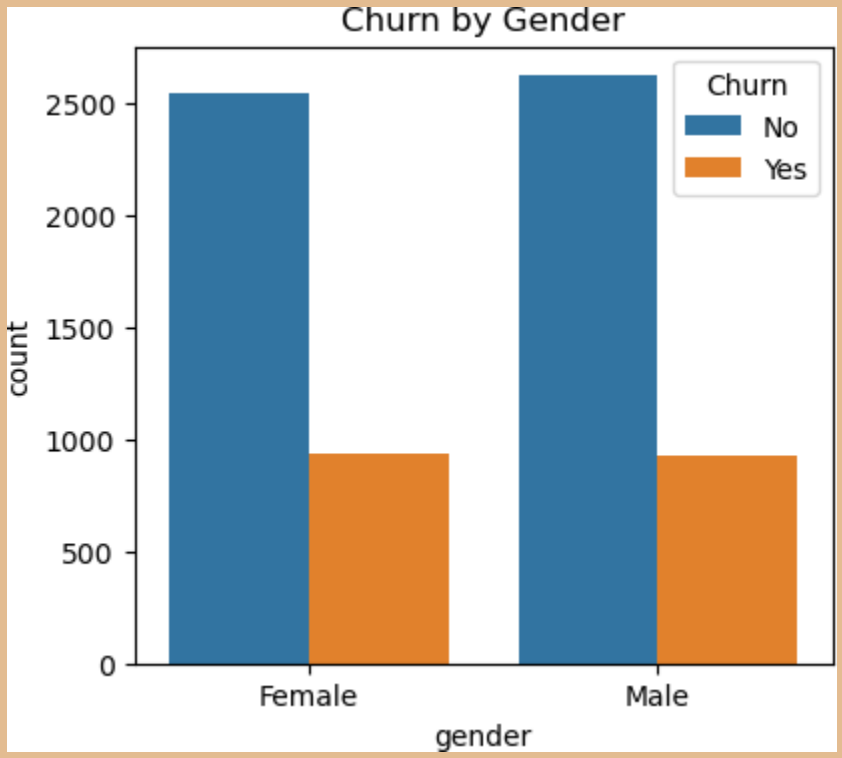


In [37]:
```
plt.figure(figsize=(4,4))
gb=df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb["Churn"],labels=gb.index,autopct="%1.2f%%")
plt.title(" Percentage of Churned Customers ")
plt.show()
```

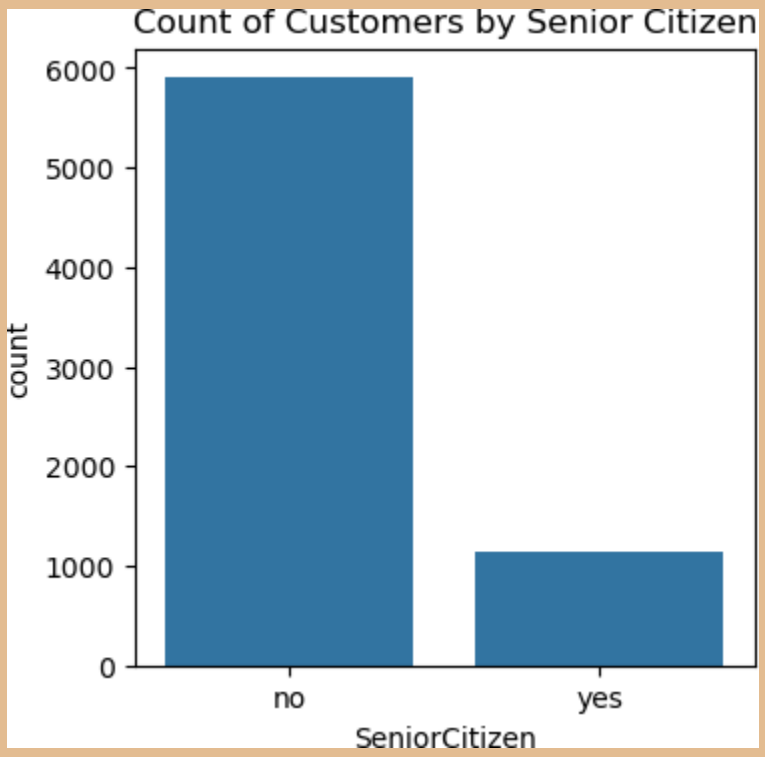


From the given pie chart we can conclude that 26.54% of our customers have Churned out. Now let's explore the reason behind it.

In [41]:
```
plt.figure(figsize=(4.5,4))
sns.countplot(x="gender",data=df,hue="Churn")
plt.title("Churn by Gender")
plt.show()
```

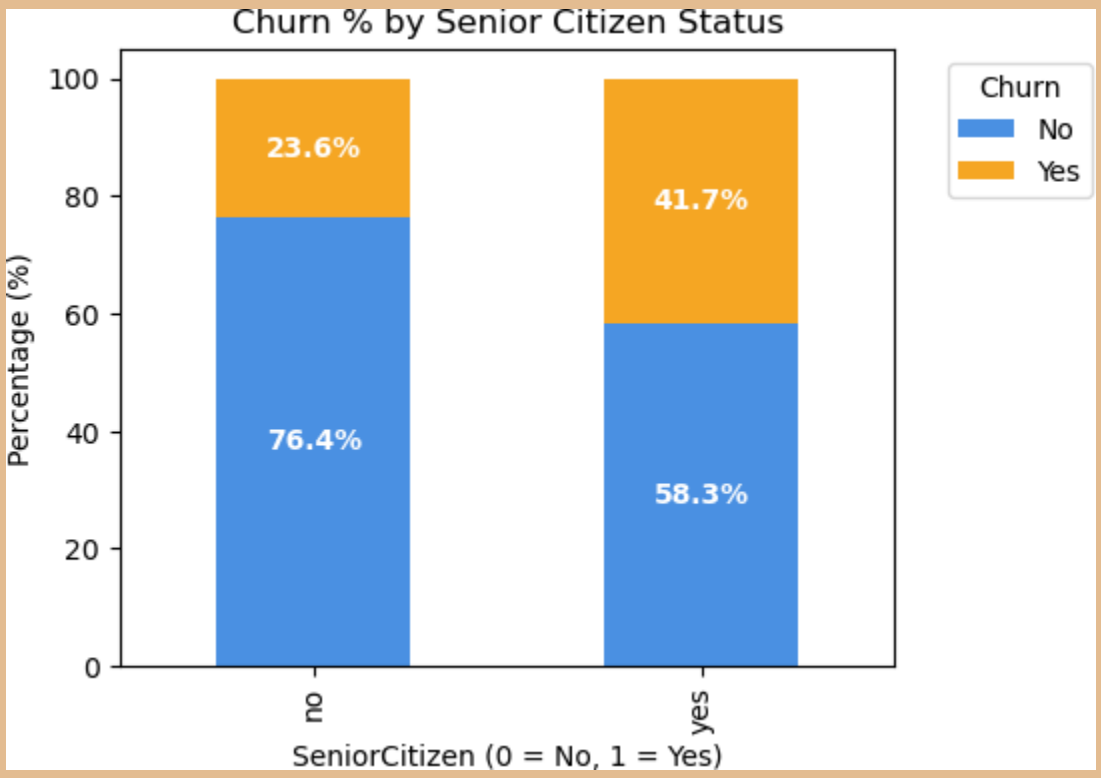


In [43]:
```
plt.figure(figsize=(4,4))
sns.countplot(x="SeniorCitizen",data=df)
plt.title("Count of Customers by Senior Citizen")
plt.show()
```



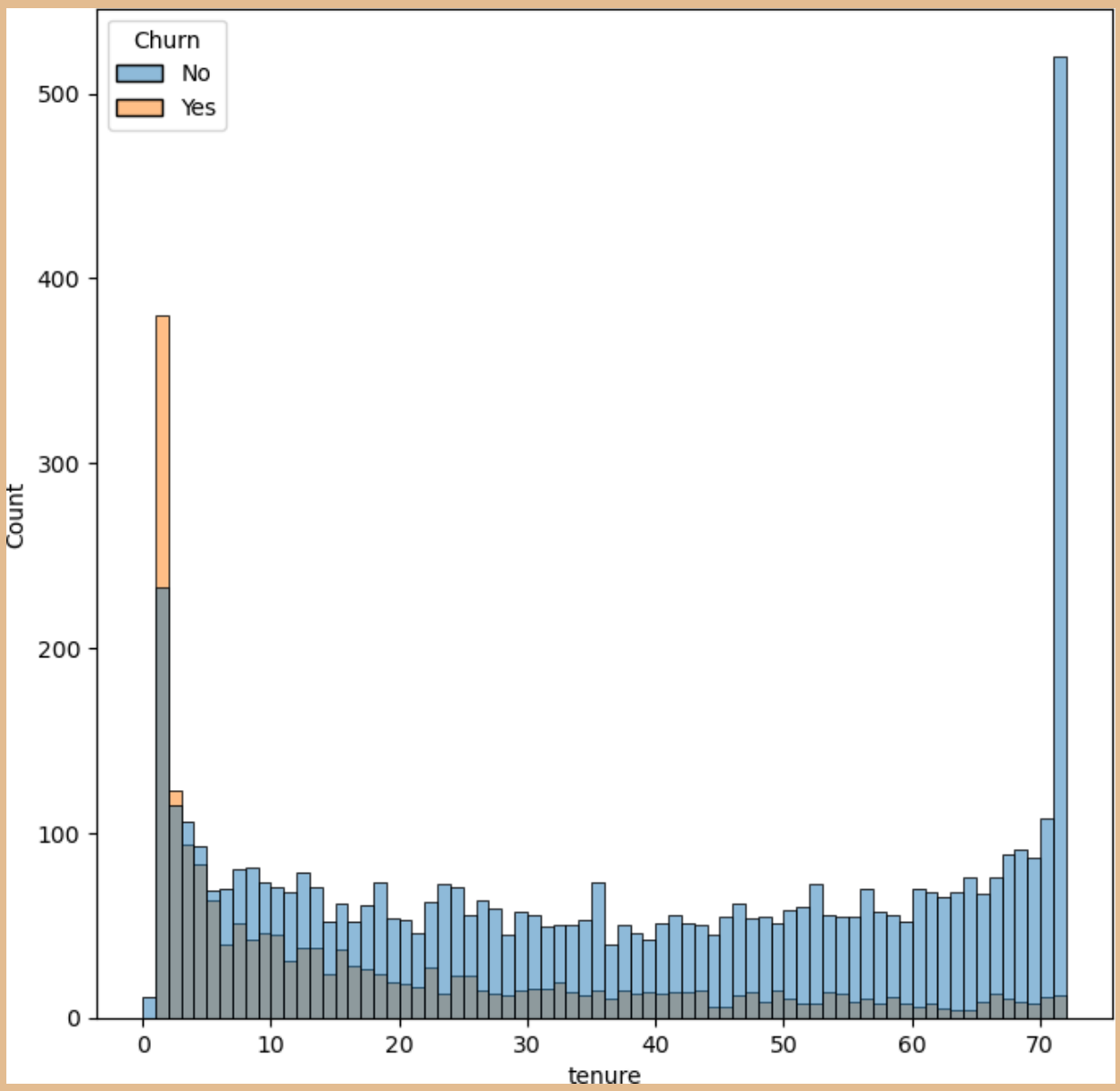
In [49]:

```
ct = pd.crosstab(df["SeniorCitizen"], df["Churn"], normalize="index") * 100

# Softer shades of blue & orange
colors = ["#4A90E2", "#F5A623"]    # muted blue and warm orange

# 2️⃣ Plot stacked bar chart
ax = ct.plot(
    kind="bar", stacked=True, figsize=(5,4), color=colors
)

plt.title("Churn % by Senior Citizen Status")
plt.ylabel("Percentage (%)")
plt.xlabel("SeniorCitizen (0 = No, 1 = Yes)")
plt.legend(title="Churn", bbox_to_anchor=(1.05, 1), loc='upper left')

# 3️⃣ Add percentage labels
for container in ax.containers:
    ax.bar_label(container, fmt="%.1f%%", label_type="center", color="white",

plt.show()
```



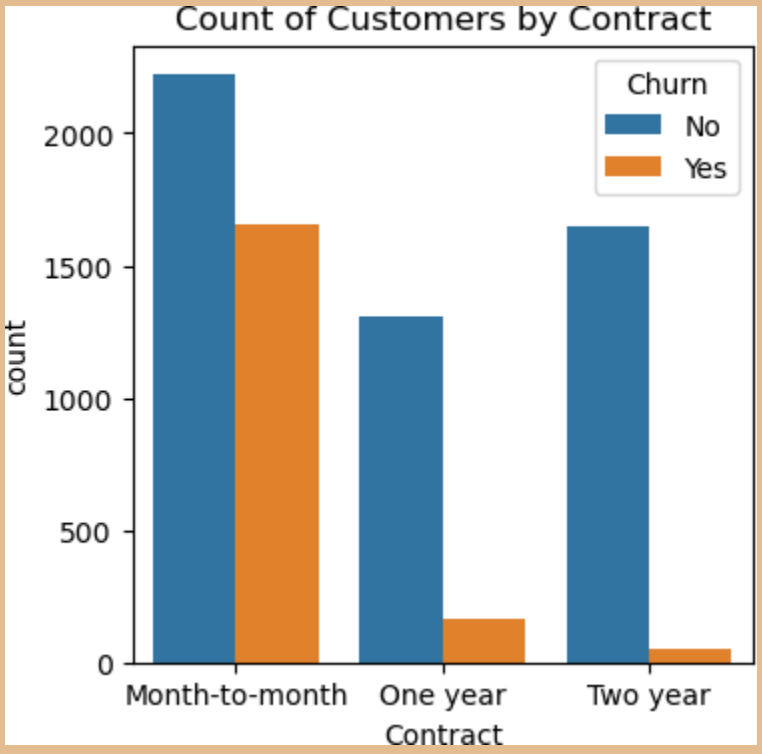
The chart shows the percentage of customers who churned vs stayed, split by whether they are Senior Citizens (1) or Non-Senior Citizens (0).

In [59]:
```
plt.figure(figsize=(8,8))
sns.histplot(x="tenure",data=df,bins=72,hue="Churn")
plt.show()
```



Most people who left did so in the first 1-2 months, while those who stayed kept using the service for a long time.

In [60]:
```
plt.figure(figsize=(4,4))
sns.countplot(x="Contract",data=df,hue="Churn")
plt.title("Count of Customers by Contract")
plt.show()
```



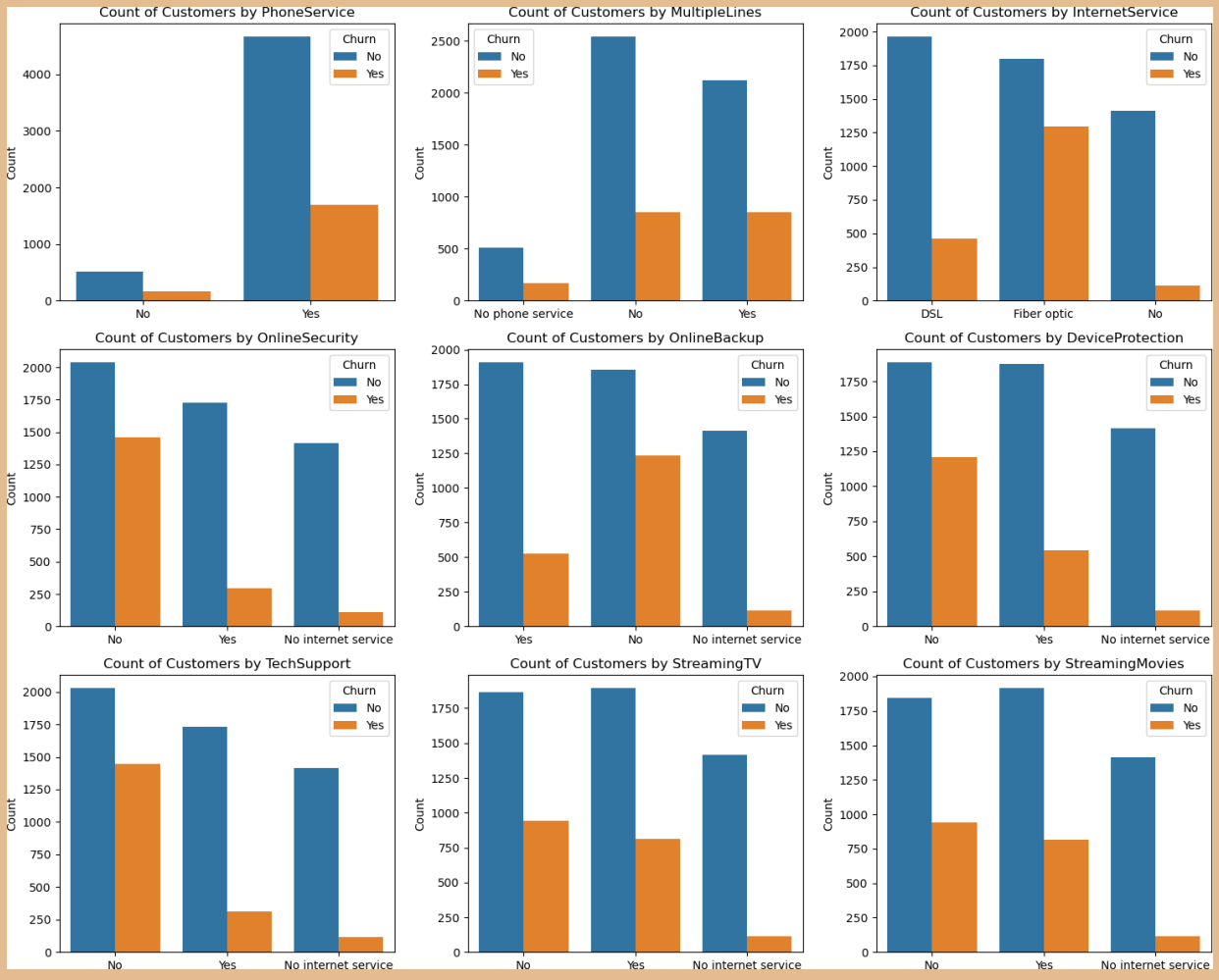
Customers with 1-2 year contracts mostly stayed, while those on month-to- month contracts were more likely to leave.

In [62]:
```
df.columns.values
```

Out[62]:
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

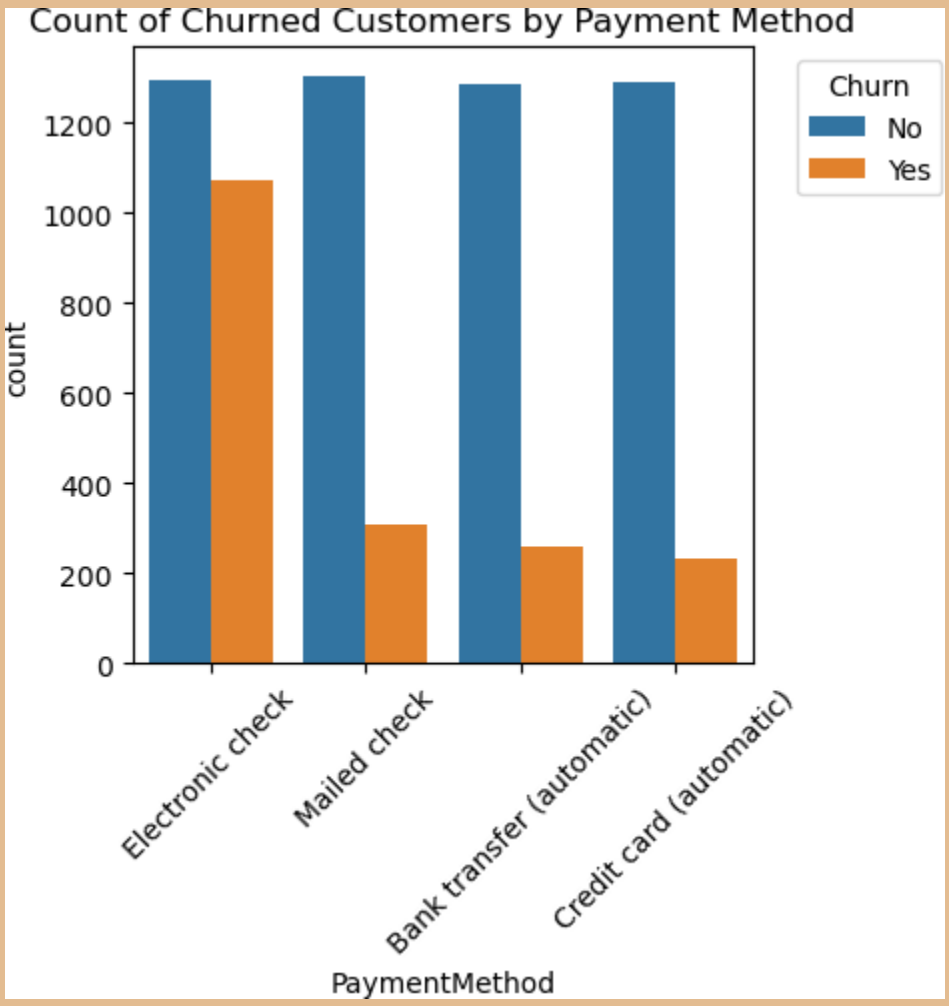
In [64]:
```
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Figure with subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15,12))
axes = axes.flatten()  # flatten 2D array of axes into 1D

# Loop through columns
```

```
for i, col in enumerate(cols):
    sns.countplot(x=col, data=df, ax=axes[i],hue="Churn")   # default color sch
    axes[i].set_title(f"Count of Customers by {col}")
    axes[i].set_xlabel("")  # optional: remove x-axis label for cleaner look
    axes[i].set_ylabel("Count")

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

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



Customers without extra services like Online Security, Backup, or Device Protection were more likely to churn, while those with these services tended to stay.

In [70]:
```
plt.figure(figsize=(4,4))
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.legend(title="Churn", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.title("Count of Churned Customers by Payment Method")
plt.show()
```



Customers paying by electronic check churned the most, while those using automatic payments (bank transfer or credit card) were more likely to stay.