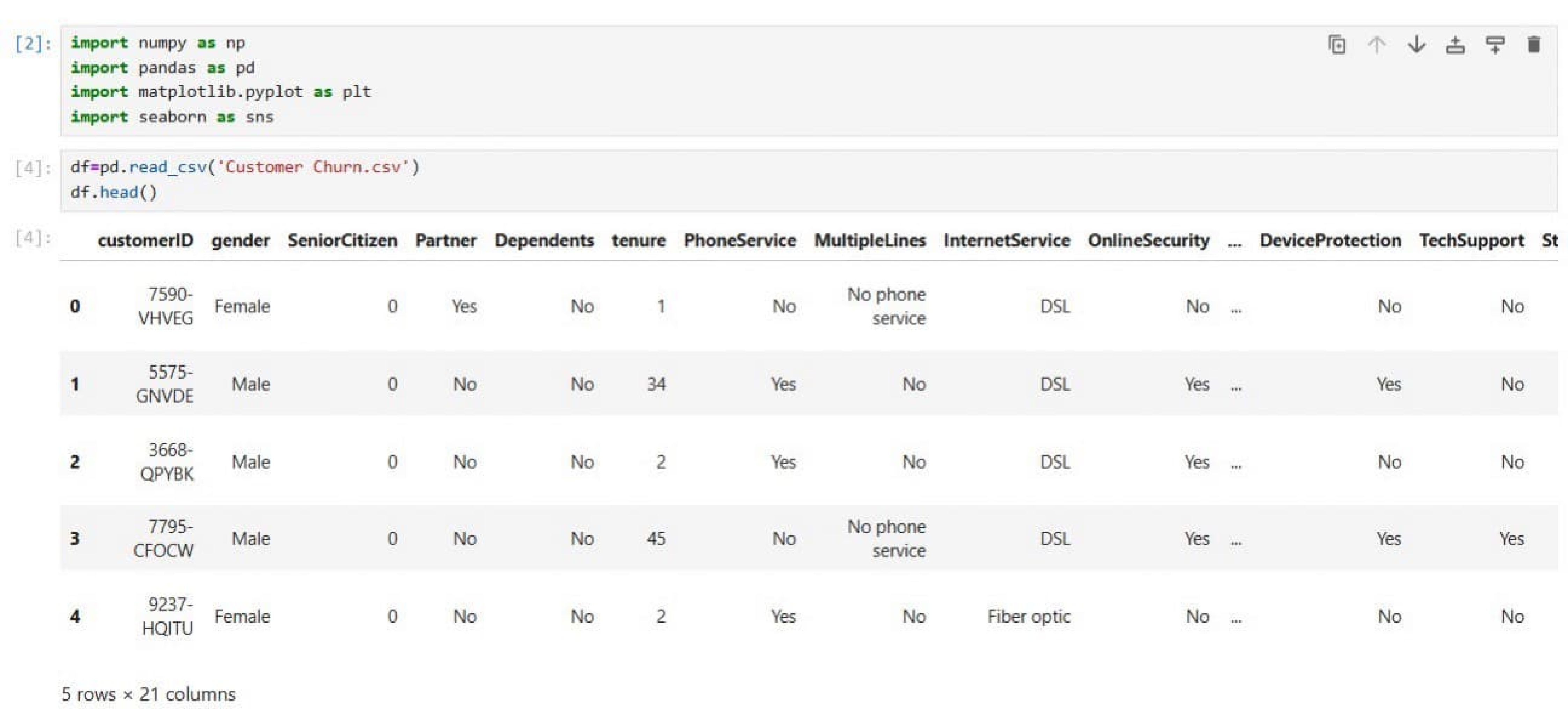


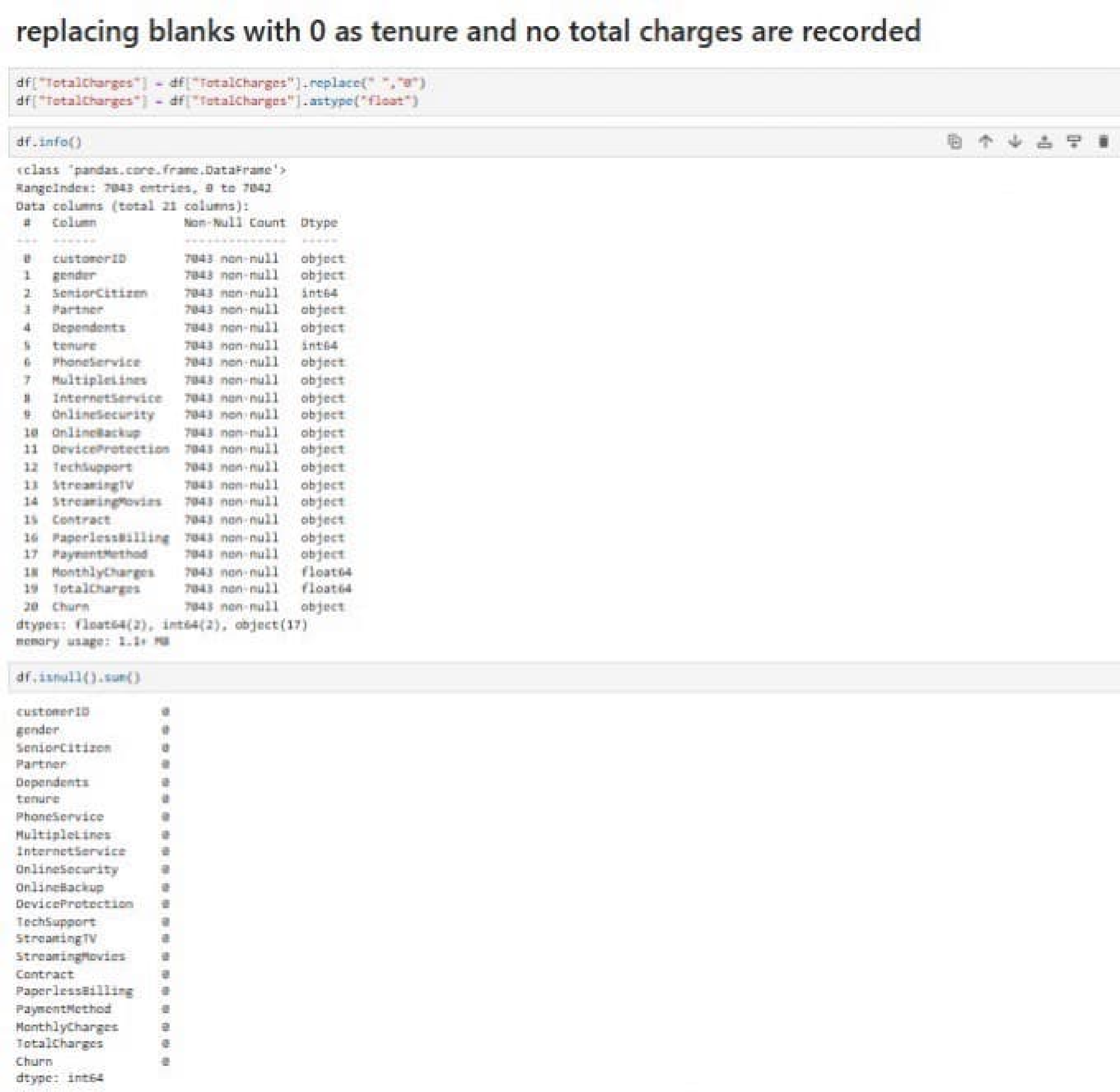
[2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```python
df=pd.read_csv('Customer Churn.csv')
df.head()
```

[4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	St
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	

5 rows × 21 columns



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

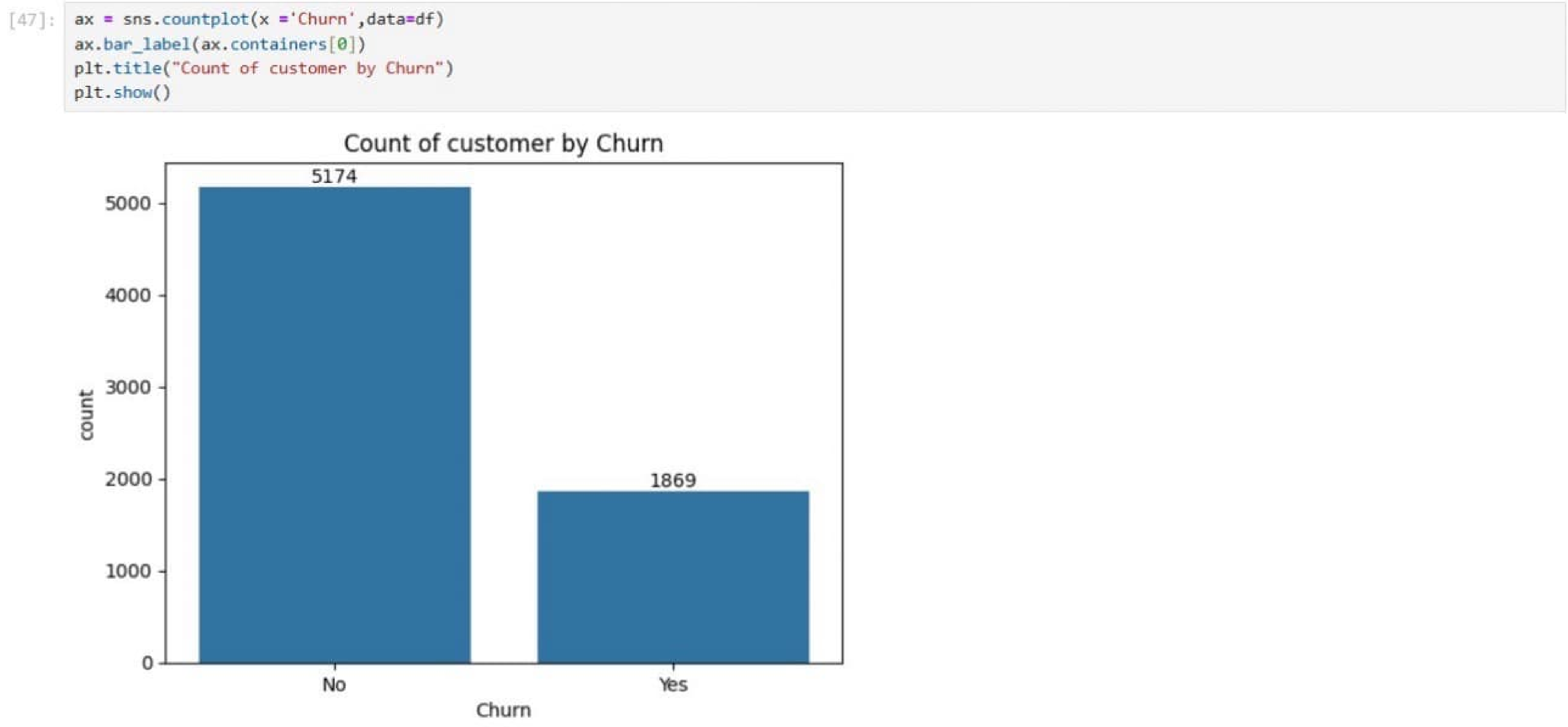
```
df["TotalCharges"] = df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

```
df.info()
```

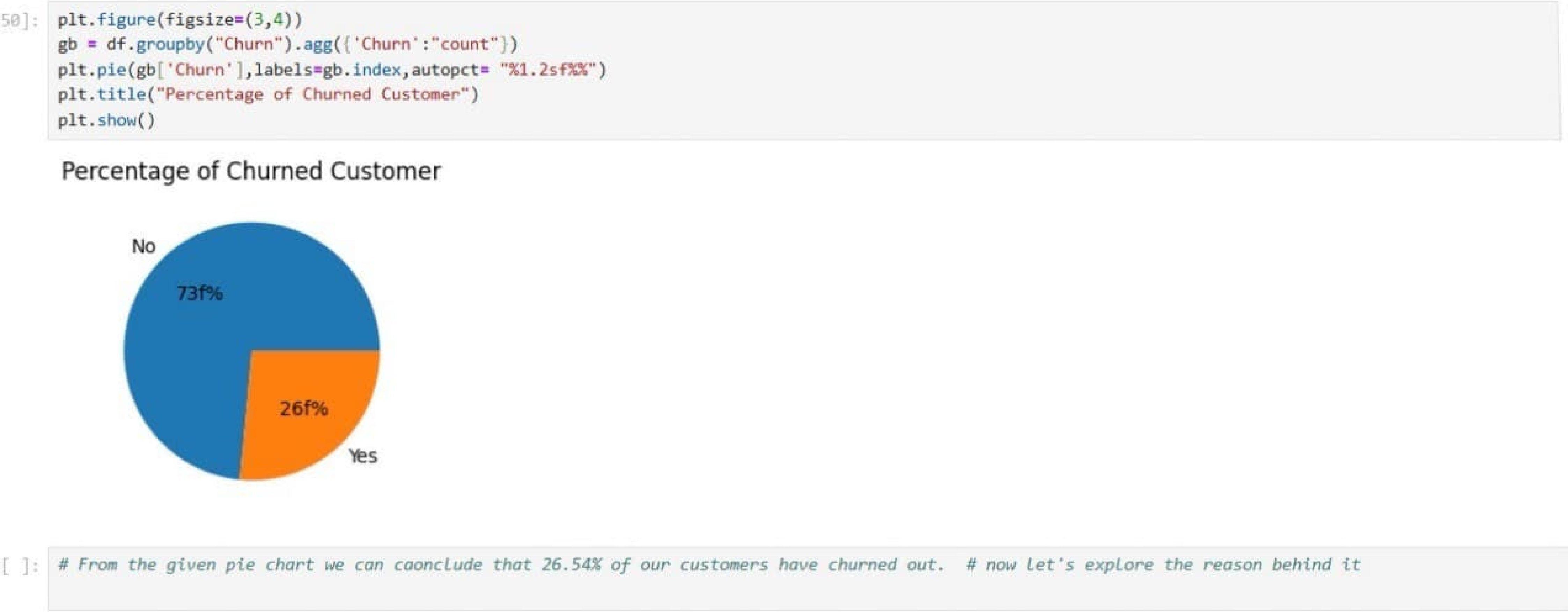
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   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
```

```
df.isnull().sum()
```

```
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
```



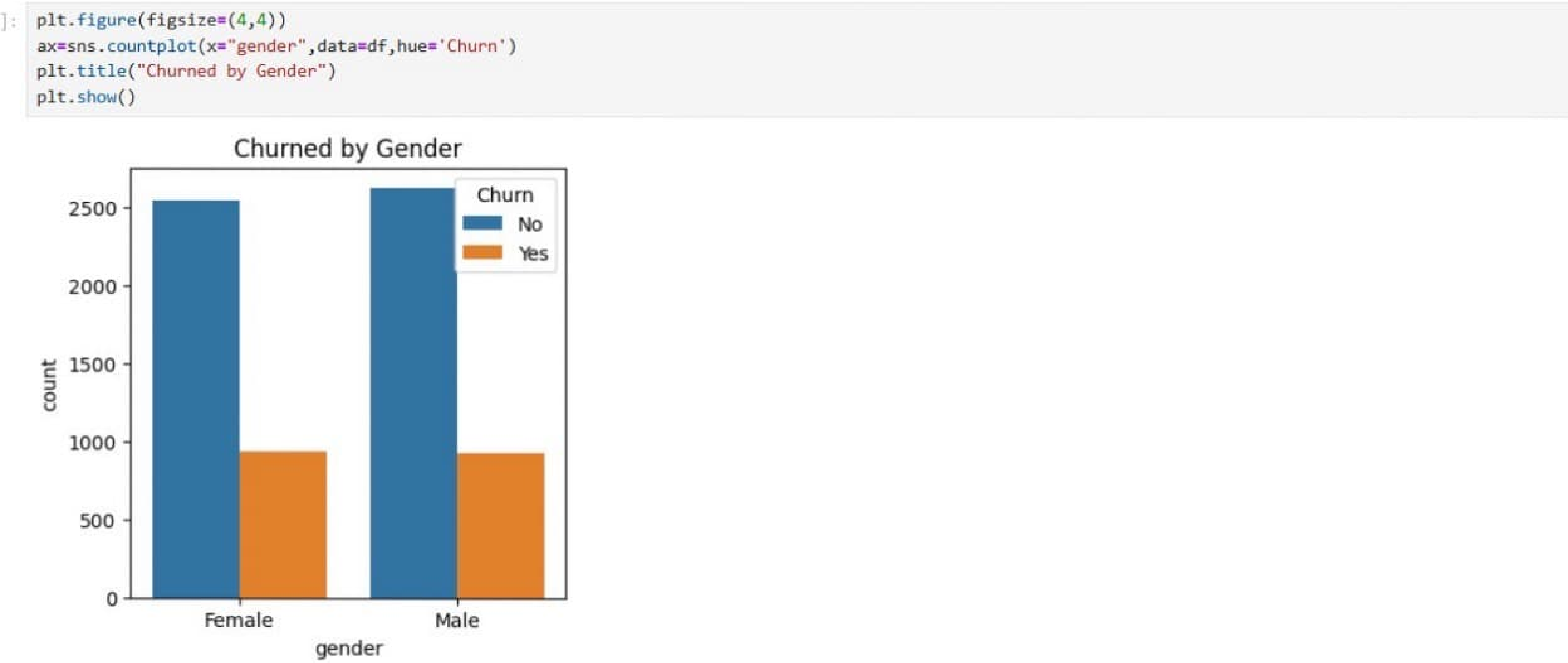
```
[47]: ax = sns.countplot(x ='Churn',data=df)
      ax.bar_label(ax.containers[0])
      plt.title("Count of customer by Churn")
      plt.show()
```



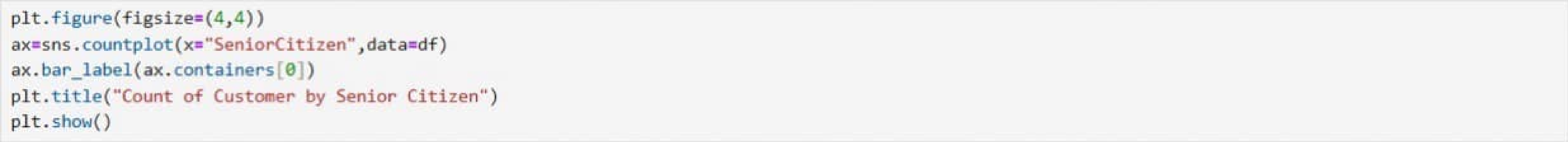
```
50]: plt.figure(figsize=(3,4))
     gb = df.groupby("Churn").agg({'Churn':"count"})
     plt.pie(gb['Churn'],labels=gb.index,autopct= "%1.2sf%%")
     plt.title("Percentage of Churned Customer")
     plt.show()
```

Percentage of Churned Customer

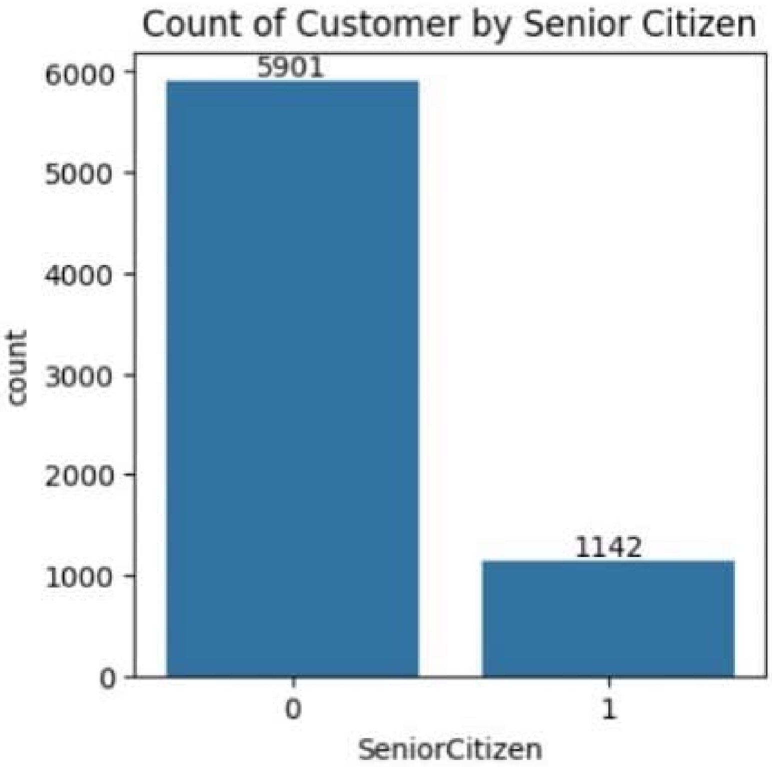
```
[ ]: # From the given pie chart we can caonclude that 26.54% of our customers have churned out.  # now let's explore the reason behind it
```

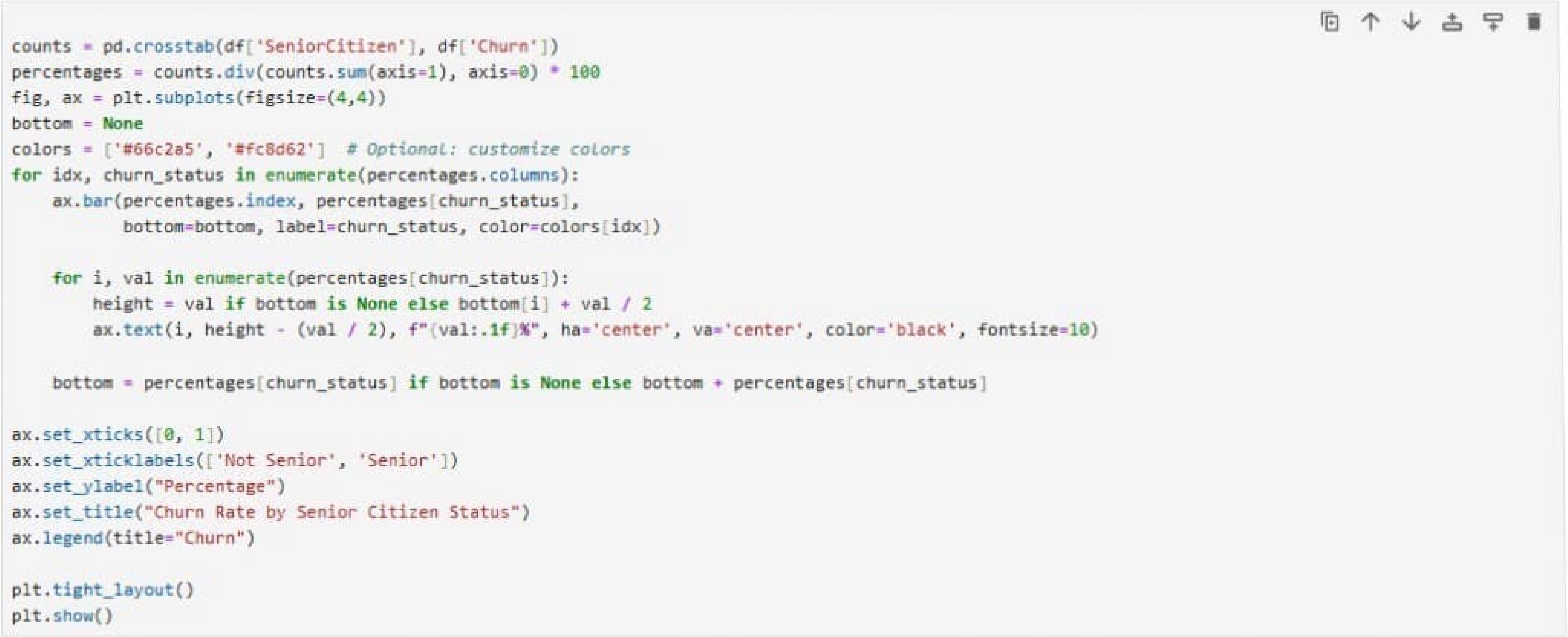


```
plt.figure(figsize=(4,4))
ax=sns.countplot(x="gender",data=df,hue='Churn')
plt.title("Churned by Gender")
plt.show()
```

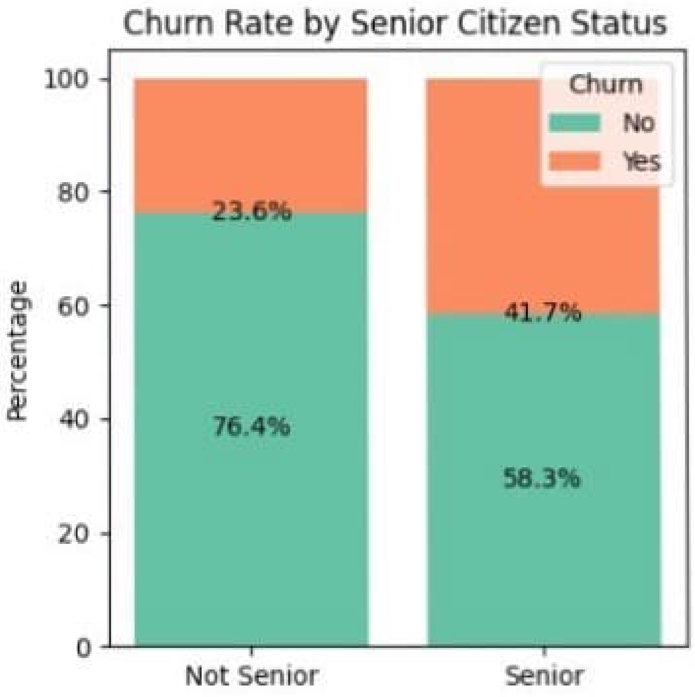


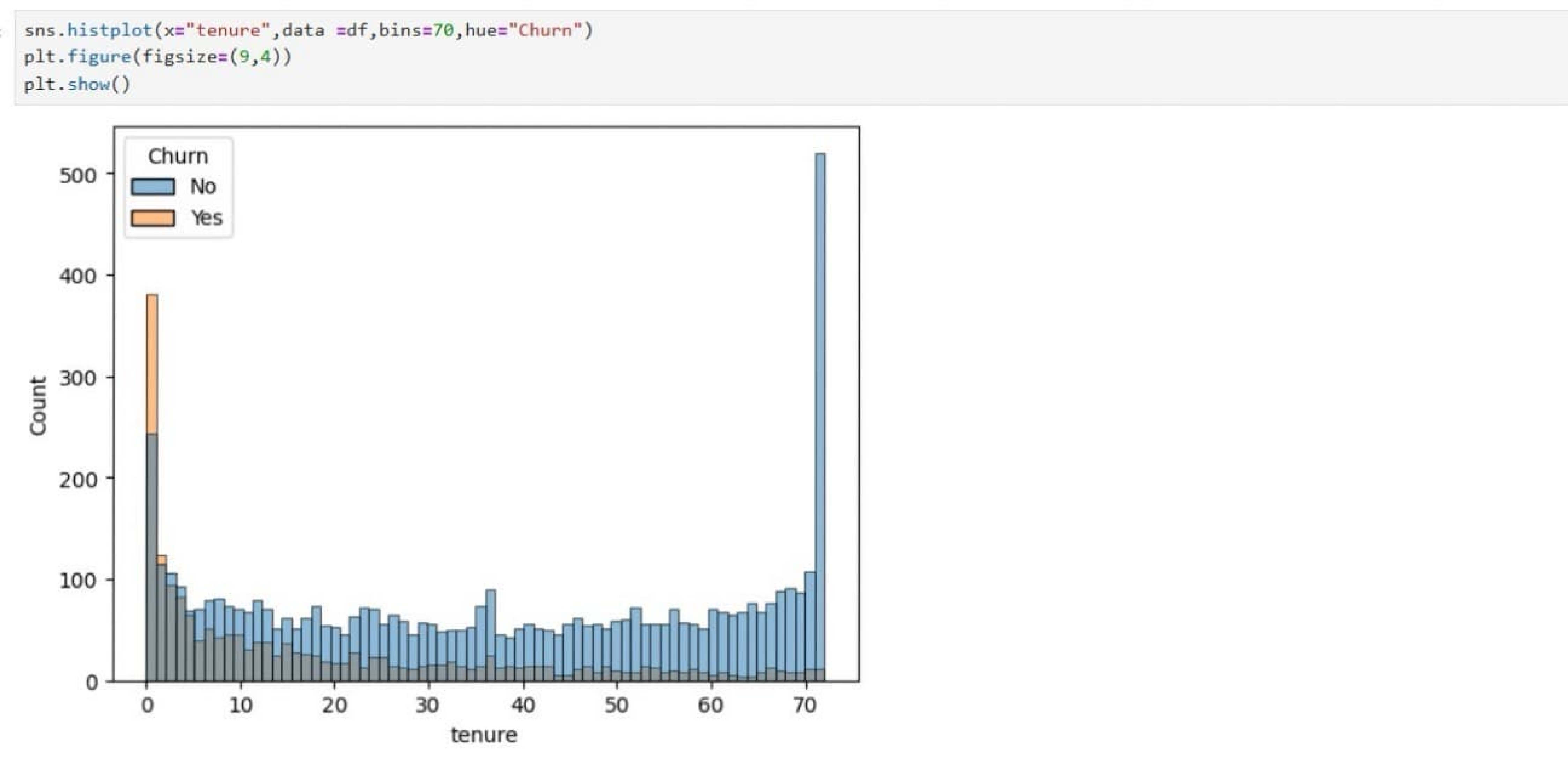
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x="SeniorCitizen",data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customer by Senior Citizen")
plt.show()
```



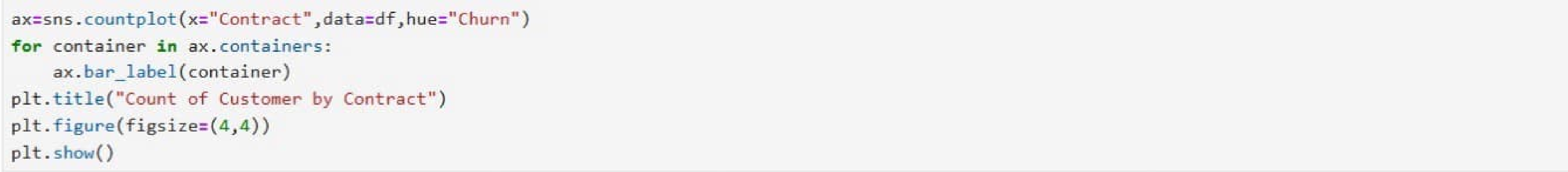


```python
counts = pd.crosstab(df['SeniorCitizen'], df['Churn'])
percentages = counts.div(counts.sum(axis=1), axis=0) * 100
fig, ax = plt.subplots(figsize=(4,4))
bottom = None
colors = ['#66c2a5', '#fc8d62']  # Optional: customize colors
for idx, churn_status in enumerate(percentages.columns):
    ax.bar(percentages.index, percentages[churn_status],
           bottom=bottom, label=churn_status, color=colors[idx])

    for i, val in enumerate(percentages[churn_status]):
        height = val if bottom is None else bottom[i] + val / 2
        ax.text(i, height - (val / 2), f"{val:.1f}%", ha='center', va='center', color='black', fontsize=10)

    bottom = percentages[churn_status] if bottom is None else bottom + percentages[churn_status]

ax.set_xticks([0, 1])
ax.set_xticklabels(['Not Senior', 'Senior'])
ax.set_ylabel("Percentage")
ax.set_title("Churn Rate by Senior Citizen Status")
ax.legend(title="Churn")

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

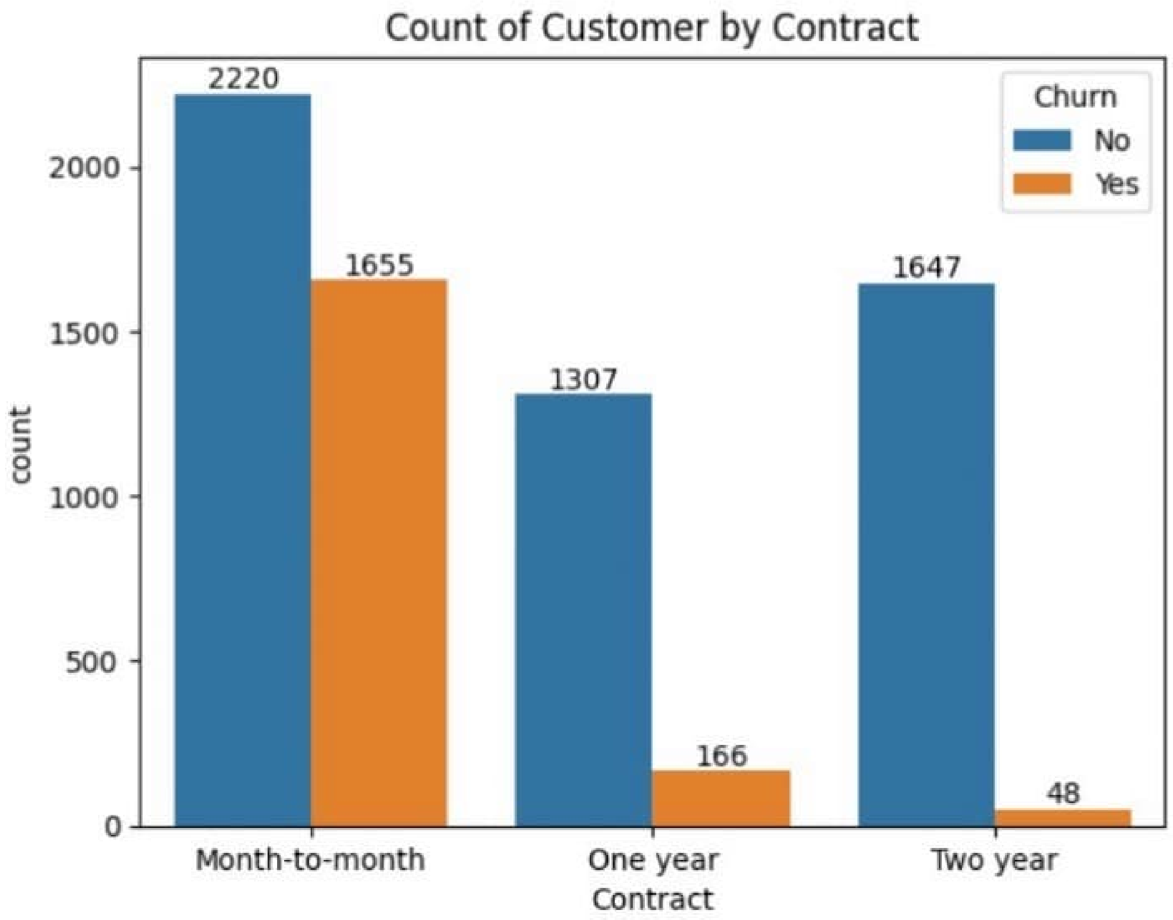


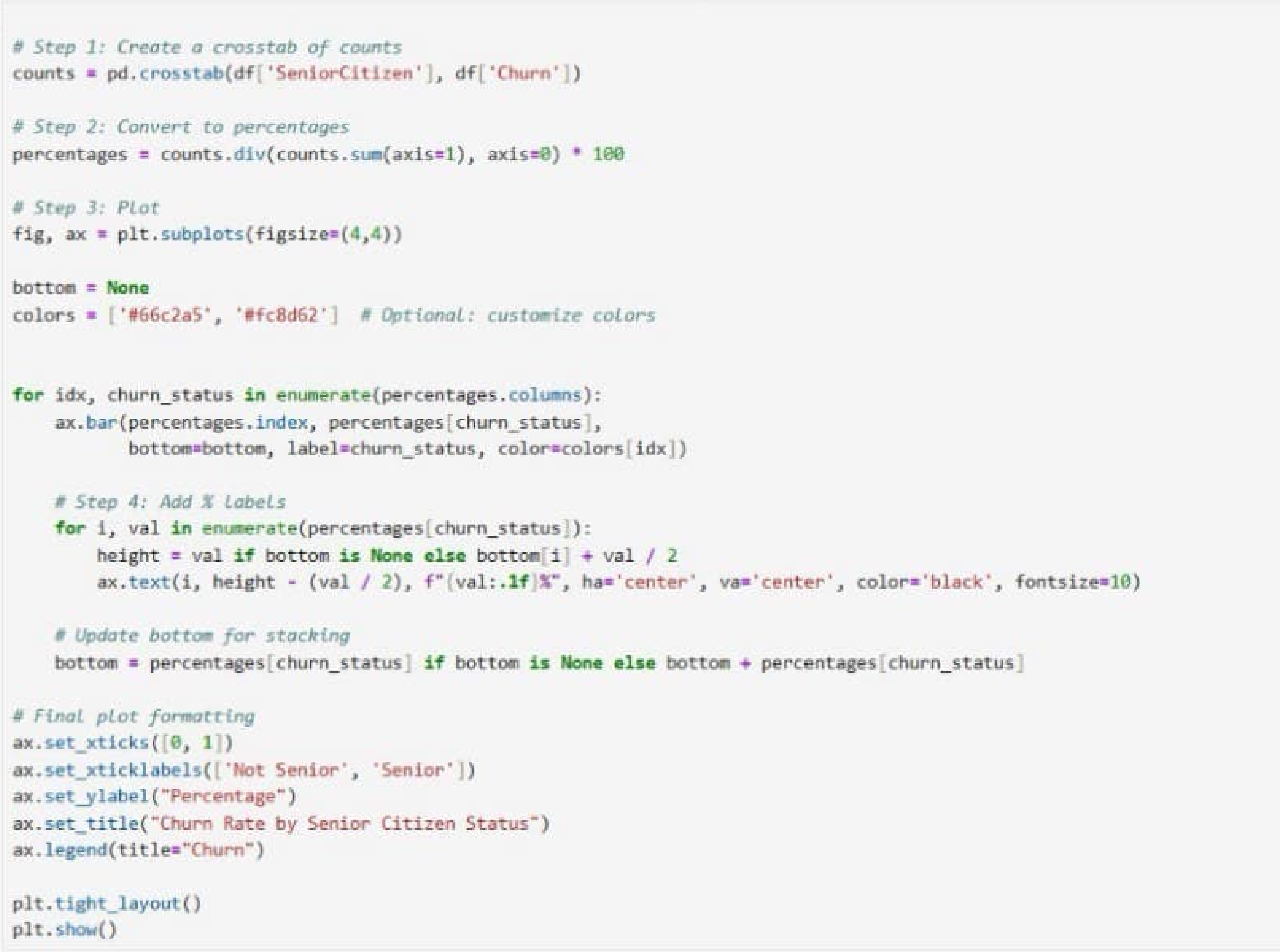


```
sns.histplot(x="tenure",data =df,bins=70,hue="Churn")
plt.figure(figsize=(9,4))
plt.show()
```

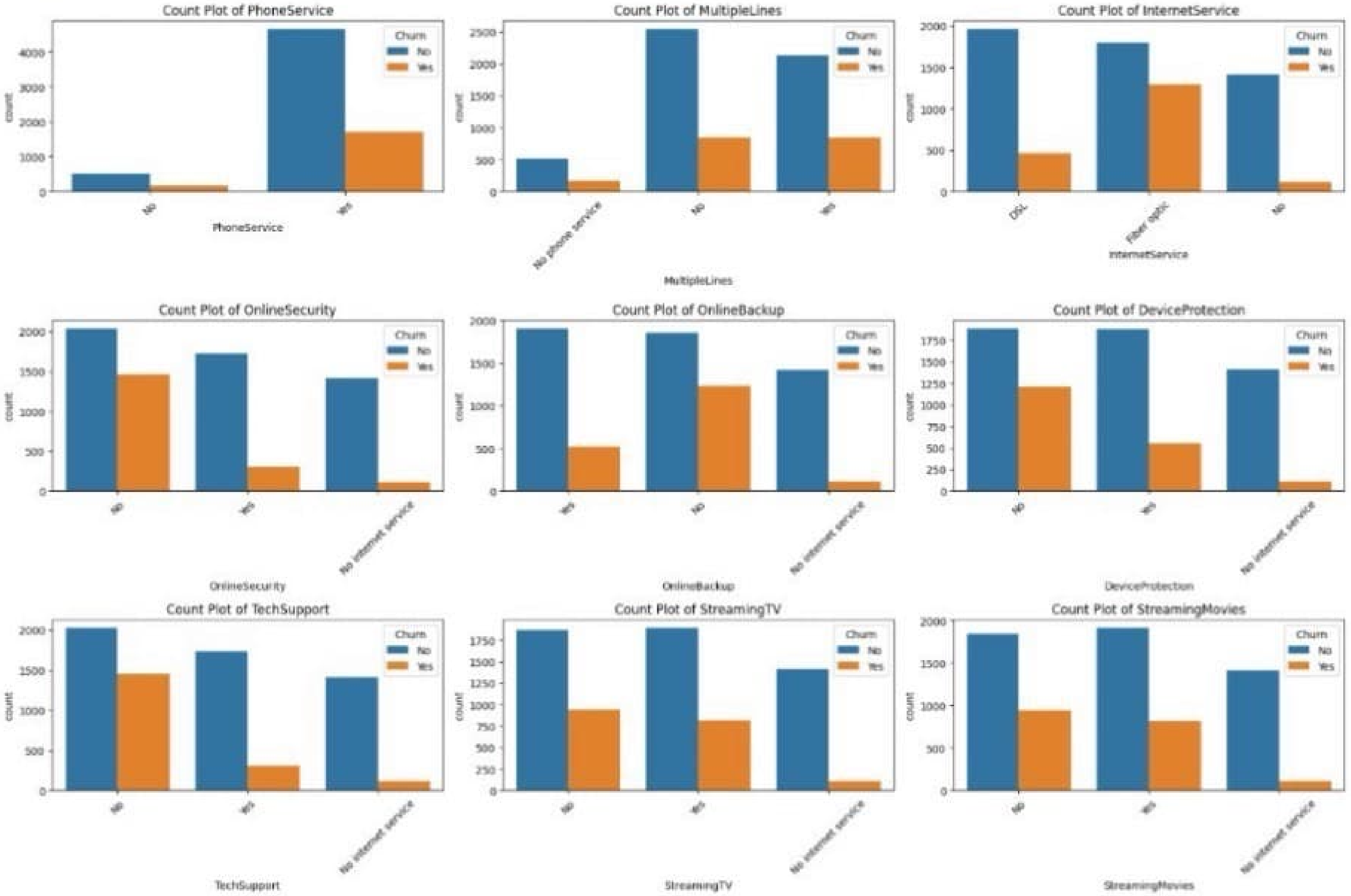


```python
ax=sns.countplot(x="Contract",data=df,hue="Churn")
for container in ax.containers:
    ax.bar_label(container)
plt.title("Count of Customer by Contract")
plt.figure(figsize=(4,4))
plt.show()
```



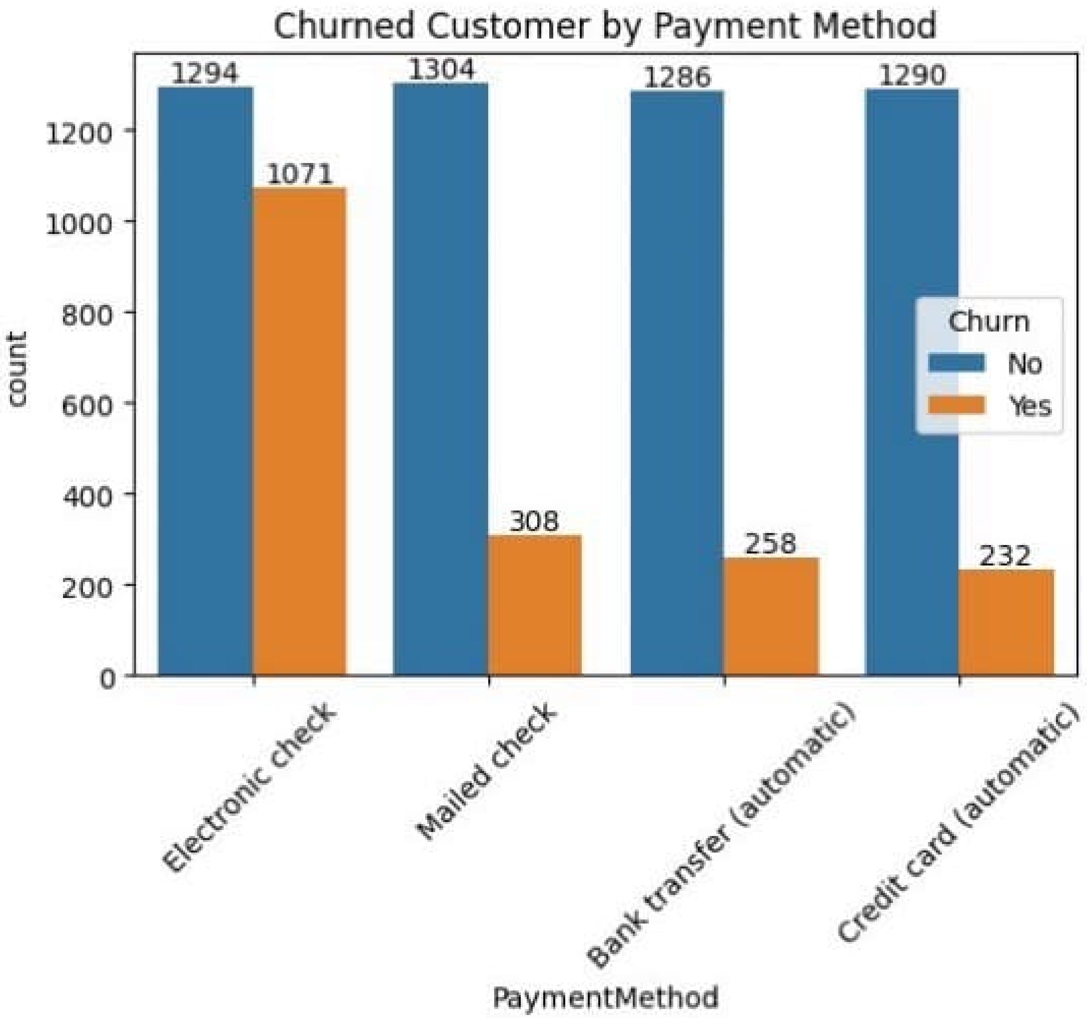


```
# Step 1: Create a crosstab of counts
counts = pd.crosstab(df['SeniorCitizen'], df['Churn'])

# Step 2: Convert to percentages
percentages = counts.div(counts.sum(axis=1), axis=0) * 100

# Step 3: Plot
fig, ax = plt.subplots(figsize=(4,4))

bottom = None
colors = ['#66c2a5', '#fc8d62']  # Optional: customize colors


for idx, churn_status in enumerate(percentages.columns):
    ax.bar(percentages.index, percentages[churn_status],
           bottom=bottom, label=churn_status, color=colors[idx])

    # Step 4: Add % Labels
    for i, val in enumerate(percentages[churn_status]):
        height = val if bottom is None else bottom[i] + val / 2
        ax.text(i, height - (val / 2), f"{val:.1f}%", ha='center', va='center', color='black', fontsize=10)

    # Update bottom for stacking
    bottom = percentages[churn_status] if bottom is None else bottom + percentages[churn_status]

# Final plot formatting
ax.set_xticks([0, 1])
ax.set_xticklabels(['Not Senior', 'Senior'])
ax.set_ylabel("Percentage")
ax.set_title("Churn Rate by Senior Citizen Status")
ax.legend(title="Churn")

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



```
# The majority of customers who don not churn tend to have services like Phone Services, InternetServices, and onlineSecurity enabled. For services
```

```
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("Churned Customer by Payment Method")
plt.xticks(rotation= 45)
plt.show()
```



```
# customer is likely tp churn when he is using electrnoic check as a payment method.
```