

```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('Customer Churn.csv')
df.head()
```

Out [4]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	29.83
1	5576-GNVEK	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year	No	Mailed check	56.16
2	3666-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month	Yes	Mailed check	53.17
3	7796-CFCOW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	70.27

5 rows × 21 columns

```
In [5]: 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(1), int64(2), object(18)
memory usage: 1.1+ MB
```

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

In [7]: df["TotalCharges"] = df["TotalCharges"].replace(" ", "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

```
In [8]: 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 [12]: df.isnull().sum().sum()

Out [12]: 0
```

In [14]:

df.describe()

Out [14]:

	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	28.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

```
In [18]: df["customerID"].duplicated().sum()

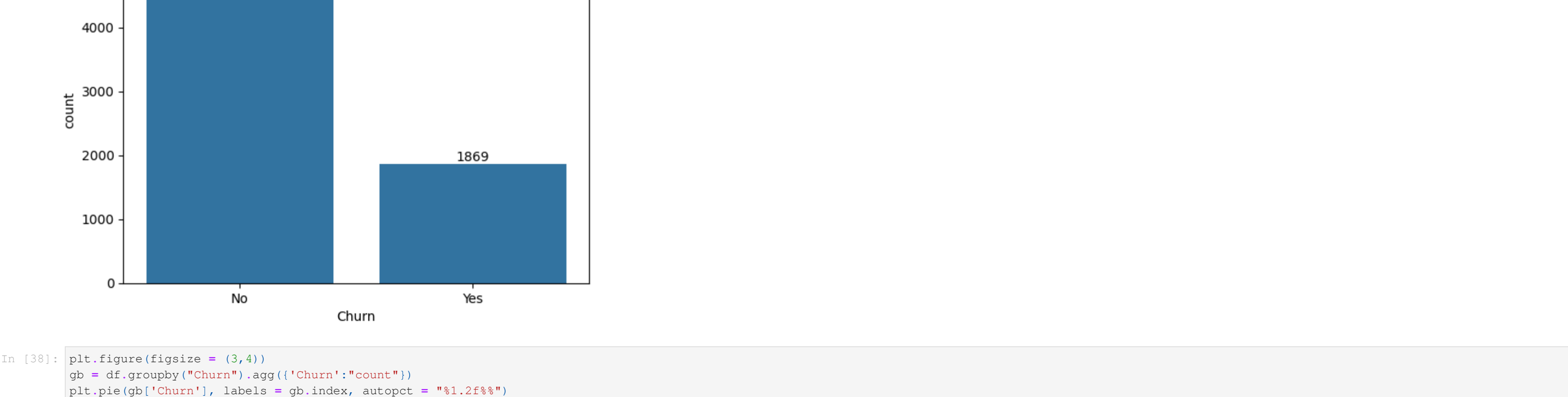
Out [18]: 0
```

```
In [19]: def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

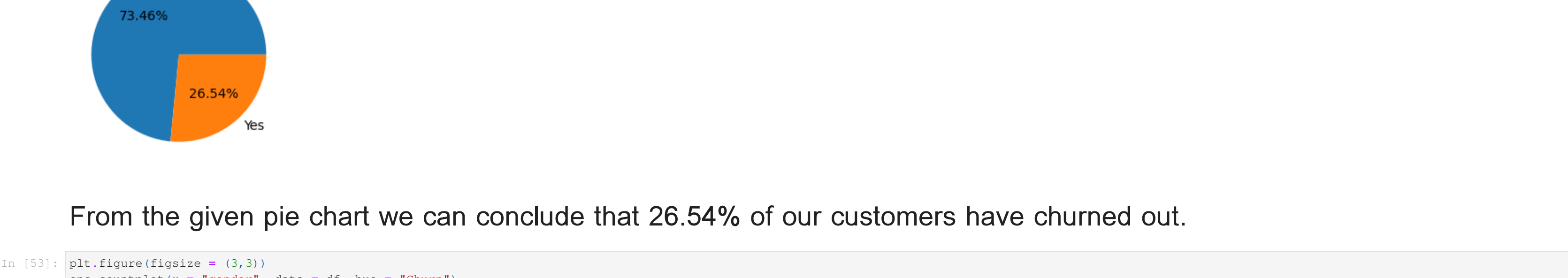
df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

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

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

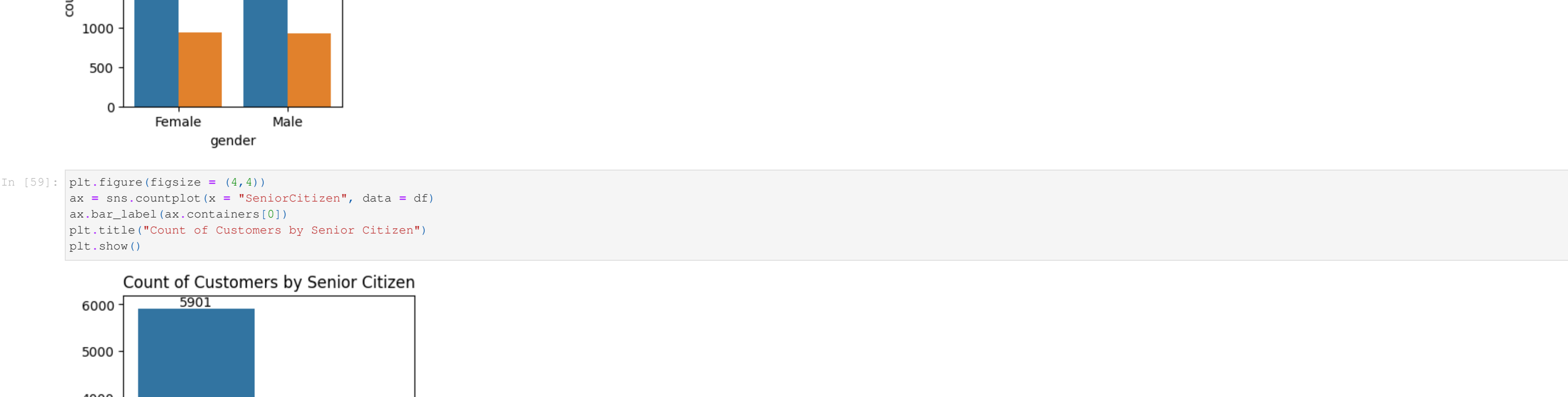


```
In [38]: plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({"Churn": "count"})
plt.pie(gb["Churn"], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage of Churned Customers", fontsize = 10)
plt.show()
```



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

```
In [53]: plt.figure(figsize = (3,3))
sns.countplot(x = "gender", data = df, hue = "Churn")
plt.title("Churn by gender")
plt.show()
```



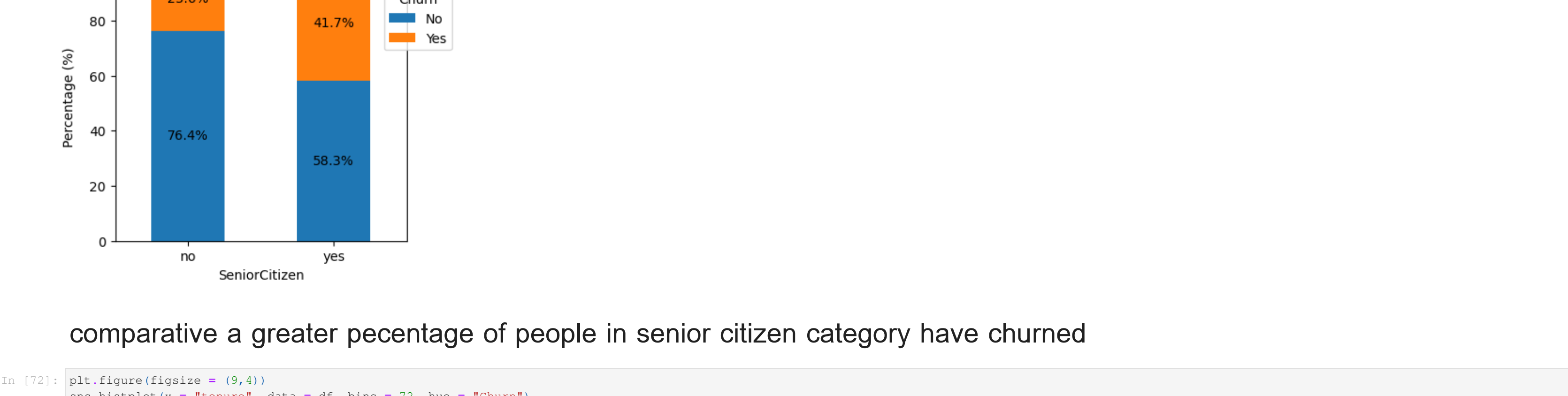
```
In [59]: plt.figure(figsize = (4,4))
ax = sns.countplot(x = "SeniorCitizen", data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Senior Citizen")
plt.show()
```



```
In [66]: total_counts = df.groupby('SeniorCitizen')['Churn'].value_counts(normalize=True).unstack() * 100
fig, ax = plt.subplots(figsize=(4, 4))

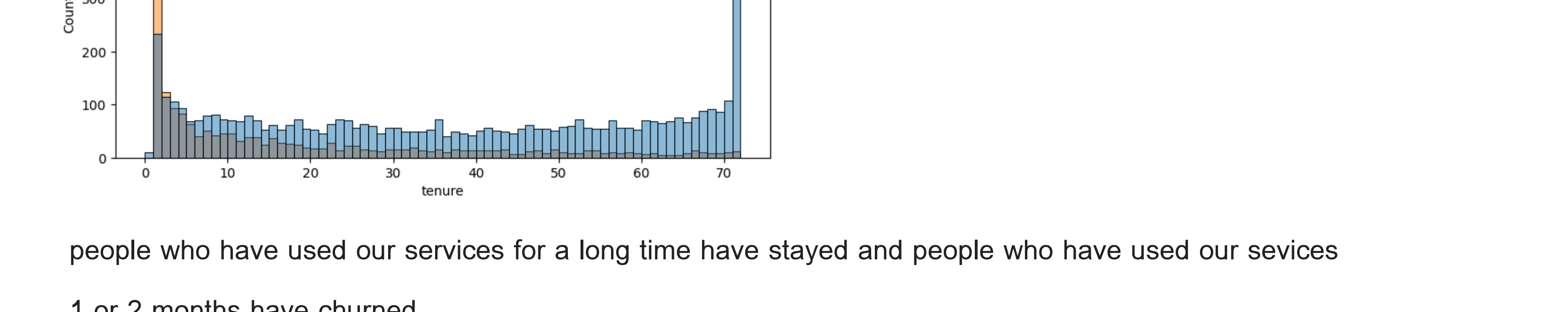
total_counts.plot(kind='bar', stacked=True, ax=ax, color=("#1f77b4", "#ff7f0e"))

for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    ax.y = p.get_y() + height / 2
    ax.text(x = width / 2, y = height / 2, f'({height:.1f}%', ha='center', va='center')
```



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

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



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

1 or 2 months have churned

```
In [74]: 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 then from those who have 1 or 2 years or contract.

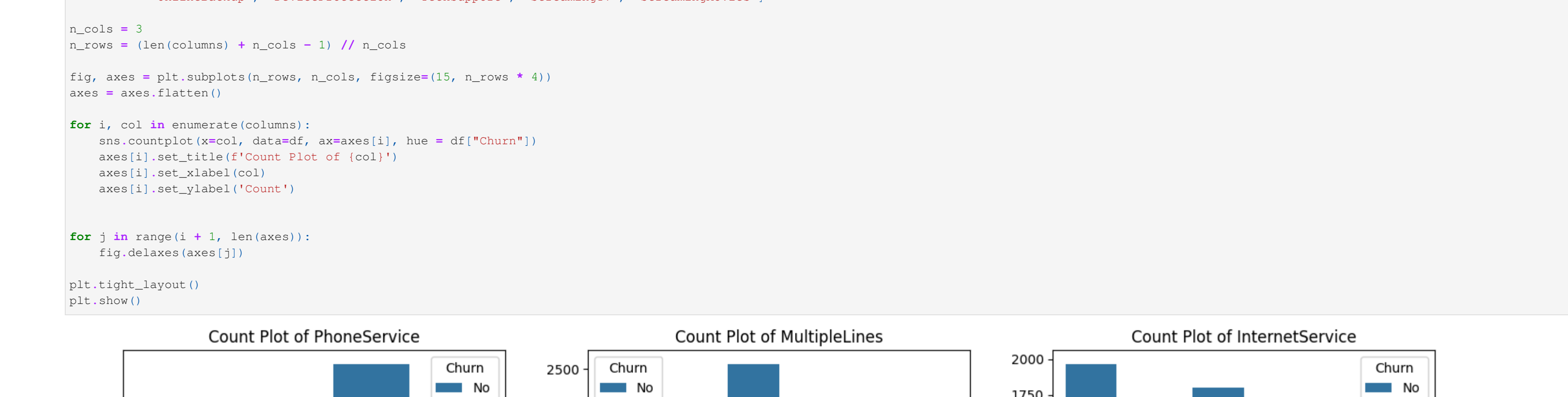
```
In [76]: df.columns.values

Out [76]: 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 [83]: columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']

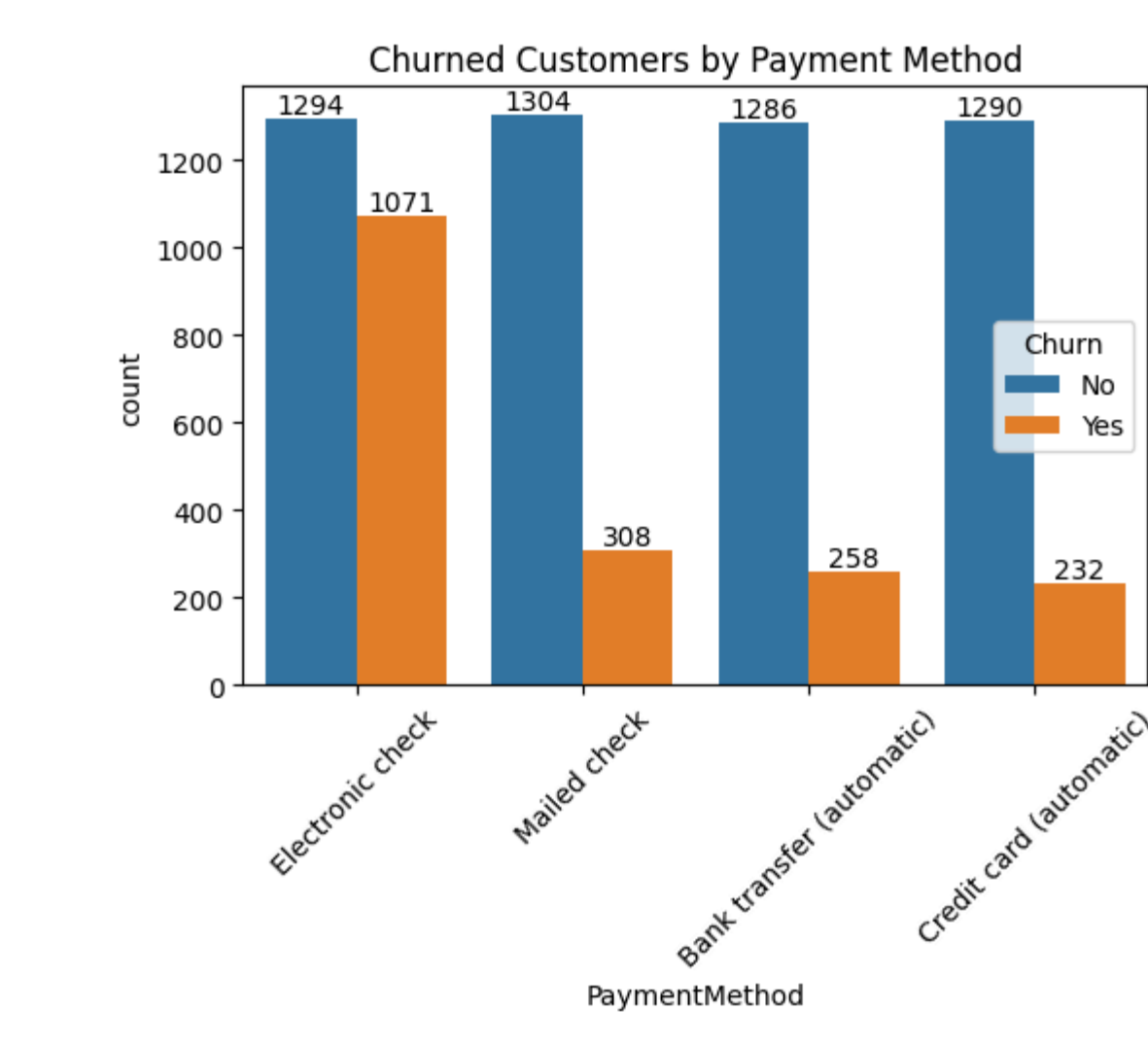
n_cols = 3
n_rows = (len(columns) + n_cols - 1) // n_cols

fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4))
axes = axes.flatten()
```



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 [90]: 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 Customers by Payment Method")
plt.xticks(rotation = 45)
plt.show()
```



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