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

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\furqu\appdata\roaming\python\p
ython313\site-packages (2.3.2)
Note: you may need to restart the kernel to use updated packages.
```

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

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\furqu\appdata\roaming\python\p
ython313\site-packages (2.3.2)
Note: you may need to restart the kernel to use updated packages.
```

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

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\users\furqu\appdata\roaming\python\
python313\site-packages (2.3.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\furqu\appdata\roaming\
python\python313\site-packages (from pandas) (2.3.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\furqu\appdat
a\roaming\python\python313\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\furqu\appdata\roaming\p
ython\python313\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\furqu\appdata\roamin
g\python\python313\site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\furqu\appdata\roaming\pytho
n\python313\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

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

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\furqu\appdata\roaming\python\p
ython313\site-packages (2.3.2)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
pip install matplotlib
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: matplotlib in c:\users\furqu\appdata\roaming\pyt
hon\python313\site-packages (3.10.7)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\furqu\appdata\roami
ng\python\python313\site-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\users\furqu\appdata\roaming\p
ython\python313\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\furqu\appdata\roam
ing\python\python313\site-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\furqu\appdata\roam
ing\python\python313\site-packages (from matplotlib) (1.4.9)
Requirement already satisfied: numpy>=1.23 in c:\users\furqu\appdata\roaming\py
thon\python313\site-packages (from matplotlib) (2.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\furqu\appdata\roamin
g\python\python313\site-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in c:\users\furqu\appdata\roaming\pyth
on\python313\site-packages (from matplotlib) (12.0.0)
Requirement already satisfied: pyparsing>=3 in c:\users\furqu\appdata\roaming\p
ython\python313\site-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\furqu\appdata\r
oaming\python\python313\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\furqu\appdata\roaming\pytho
n\python313\site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:

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	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 rows × 21 columns

In [11]:
```
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
```

Replace blank with 0 as tenure is 0 and no total charges are recorded

Also Converting object (show due to blank rows) convert into a float

In [9]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" " , "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

In [17]:
```
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
```

Understading What is isnull()

In [10]:
```
#df.isnull() we dont understant what is false so we use sum() here
df.isnull().sum()        #This is show in our data there is null value or not
df.isnull().sum().sum()  # This show the overall data is their any null value
```

Out[10]:
```
np.int64(0)
```

Now we use descriptive analysis to understand

In [11]:
```
df.describe()
```

Out[11]:

	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

Check Duplicate Values

In [12]:
```
#df.duplicated().sum()  #check entire data
df["customerID"].duplicated().sum()   # this helps to find a particular columr
```

Out[12]: np.int64(0)

Here we Convert 0/1 value into yes/no to make it easier to understand

In [13]:
```
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

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

In [17]:
```
df.head(10)   # here we check 10 starting row is here any seniorcitizen or not
```

Out[17]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneSe
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	3655-SNQYZ	Female	no	Yes	Yes	69	
16	8191-XWSZG	Female	no	No	No	52	
17	9959-WOFKT	Male	no	No	Yes	71	
18	4190-MFLUW	Female	no	Yes	Yes	10	
19	4183-MYFRB	Female	no	No	No	21	
20	8779-QRDMV	Male	yes	No	No	1	

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneSe
21	1680-VDCWW	Male	no	Yes	No	12	
22	1066-JKSGK	Male	no	No	No	1	
23	3638-WEABW	Female	no	Yes	No	58	
24	6322-HRPFA	Male	no	Yes	Yes	49	
25	6865-JZNKO	Female	no	No	No	30	
26	6467-CHFZW	Male	no	Yes	Yes	47	
27	8665-UTDHZ	Male	no	Yes	Yes	1	
28	5248-YGIJN	Male	no	Yes	No	72	
29	8773-HHUOZ	Female	no	No	Yes	17	

30 rows × 21 columns

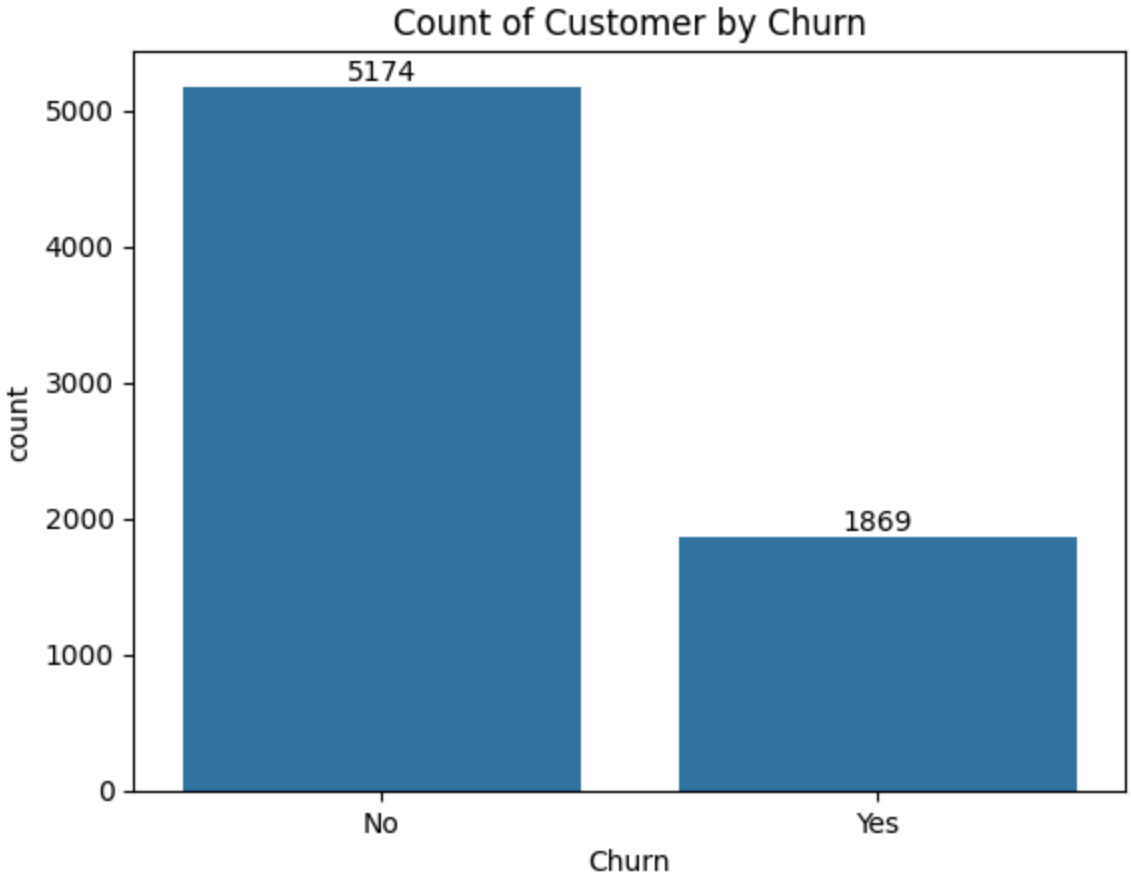
In [19]:
```
df.tail() #is called to retrieve the last rows of the DataFrame. # here easily
```

Out[19]:

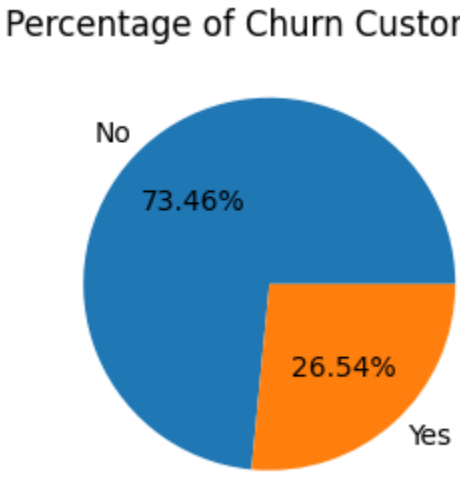
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneS
7038	6840-RESVB	Male	no	Yes	Yes	24	
7039	2234-XADUH	Female	no	Yes	Yes	72	
7040	4801-JZAZL	Female	no	Yes	Yes	11	
7041	8361-LTMKD	Male	yes	Yes	No	4	
7042	3186-AJIEK	Male	no	No	No	66	

5 rows × 21 columns

In [58]:
```
ax = sns.countplot(x = 'Churn', data=df)
ax.bar_label(ax.containers[0])
plt.title('Count of Customer by Churn')
plt.show()
```



In [61]:
```
plt.figure(figsize = (3,4))
plt.title('Percentage of Churn Customers')
gb = df.groupby('Churn').agg({'Churn':'count'})
plt.pie(gb['Churn'], labels = gb.index,  autopct ='%1.2f%%')
plt.show()
```

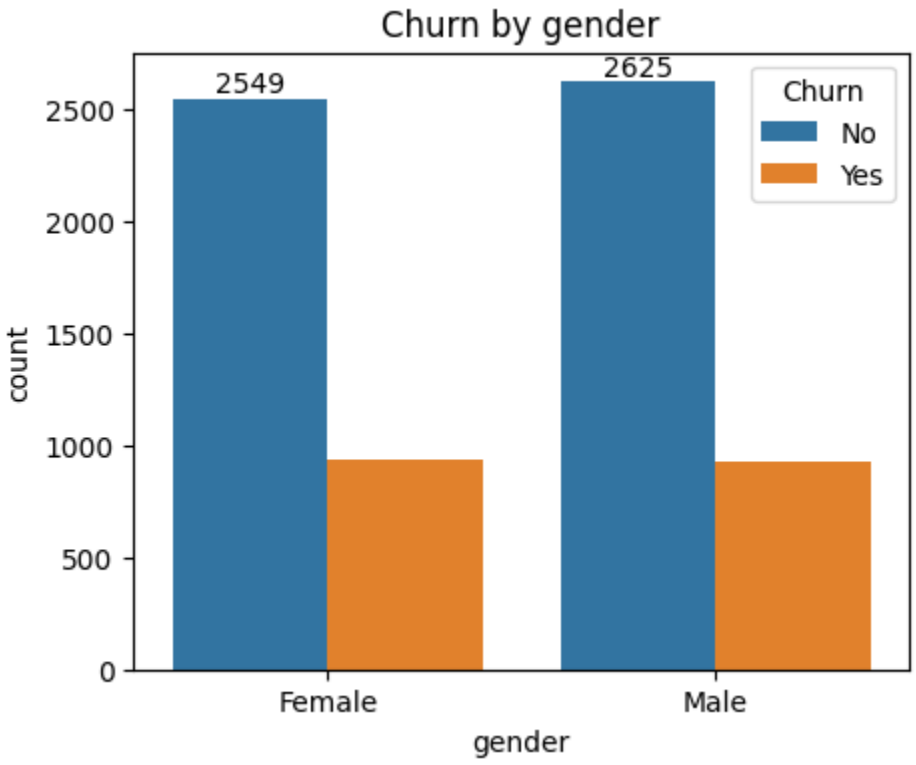


From the given pie chart we can conclude that 26.54% of our customers have churned out

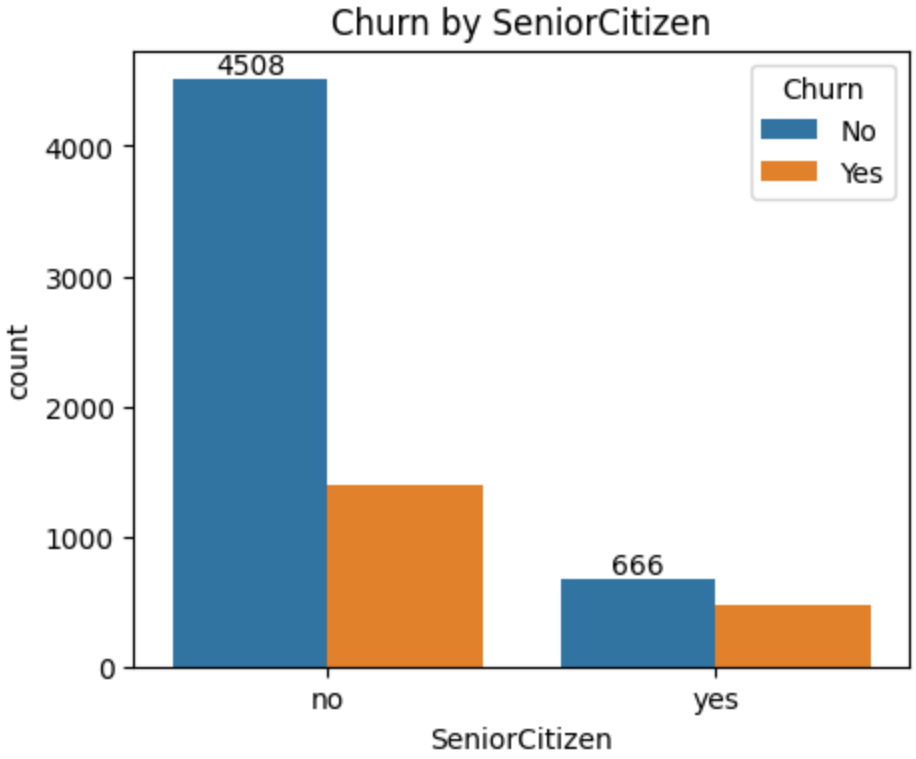
Now let explore reason behind it

Now we explore by gender

In [78]:
```
plt.figure(figsize = (5,4))
ax=sns.countplot(x = "gender" ,data = df, hue ="Churn")
plt.title("Churn by gender")
ax.bar_label(ax.containers[0])
plt.show()
```



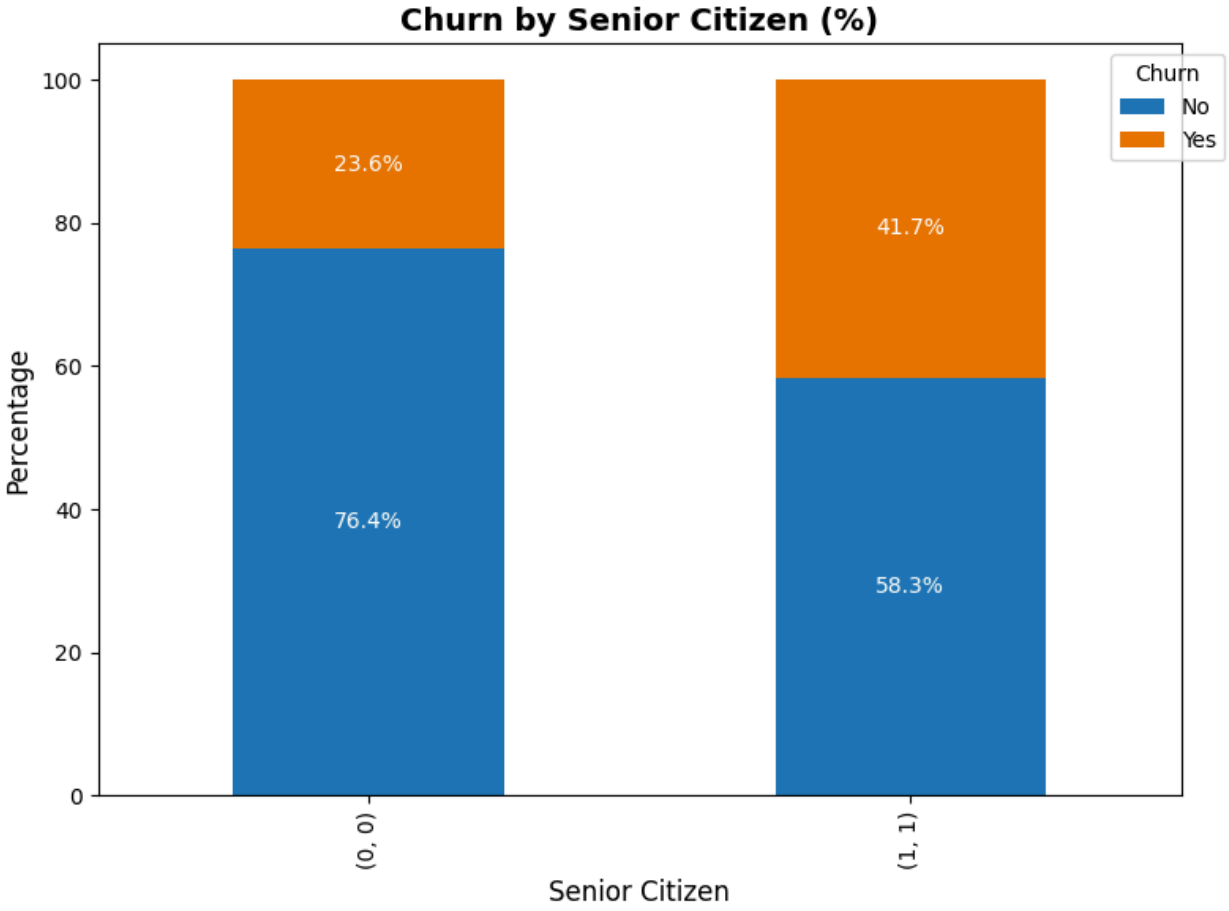
In [81]:
```
plt.figure(figsize = (5,4))
ax=sns.countplot(x = "SeniorCitizen" ,data = df, hue ="Churn")
plt.title("Churn by SeniorCitizen")
ax.bar_label(ax.containers[0])
plt.show()
```



In [23]:

```
import pandas as pd
import matplotlib.pyplot as plt

# 1️⃣ Group and calculate percentage by SeniorCitizen & Churn
data = (
    df.groupby(['SeniorCitizen', 'Churn'])
    .size()
    .groupby(level=0)
    .apply(lambda x: 100 * x / x.sum())
    .unstack(fill_value=0)
)

# 2️⃣ Create a larger figure
plt.figure(figsize=(6,5))  # You can increase width or height as you like

# 3️⃣ Plot stacked bar chart
ax = data.plot(kind='bar', stacked=True, color=['#1f77b4', '#e67300'], figsize

# 4️⃣ Add % labels
for container in ax.containers:
    ax.bar_label(container, fmt='%.1f%%', label_type='center', color='white',

# 5️⃣ Add titles and labels
plt.title("Churn by Senior Citizen (%)", fontsize=14, fontweight='bold')
plt.ylabel("Percentage", fontsize=12)
plt.xlabel("Senior Citizen", fontsize=12)
plt.legend(title="Churn", bbox_to_anchor=(1.05, 1))
plt.tight_layout()
plt.show()
```

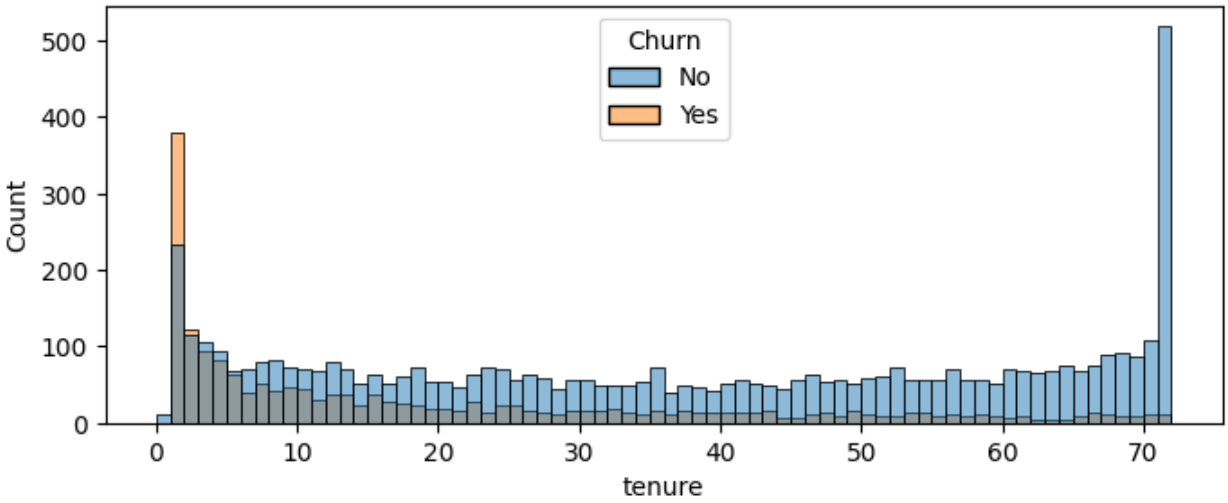
```
<Figure size 600x500 with 0 Axes>
```



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

In [27]:
```
plt.figure(figsize=(8,3))
sns.histplot(x="tenure", data=df, bins=72, hue="Churn")
plt.show
```

Out[27]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



People who have use our services for a long time have stayed and people who have used our services for a one or two month

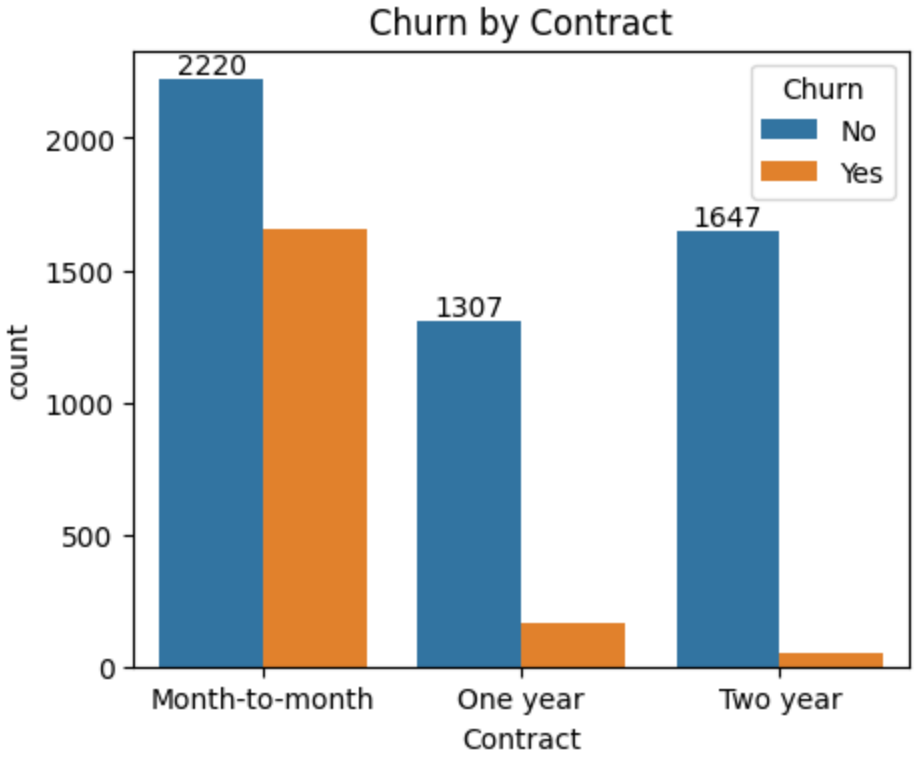
In [28]:
```
df.head()
```

Out[28]:

	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 rows × 21 columns

In [32]:
```
plt.figure(figsize = (5,4))
ax=sns.countplot(x = "Contract" ,data = df, hue ="Churn")
plt.title("Churn by Contract")
ax.bar_label(ax.containers[0])
plt.show()
```



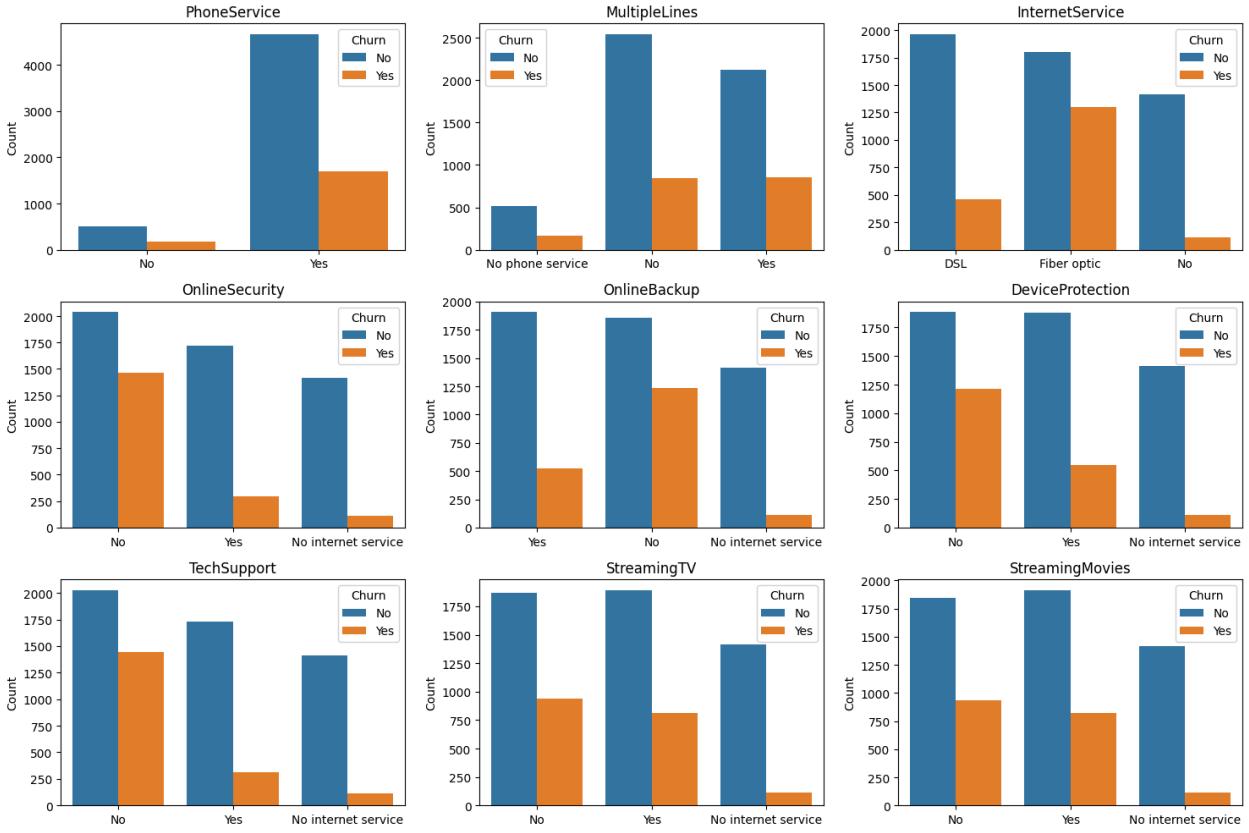
People who have month to month contract likely to churn then from those who have 1 or 2 year contract

In [35]:
```
df.columns.values
```

Out[35]:
```
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 [41]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# List of categorical columns
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Set up the subplot grid (3 rows × 3 columns)
fig, axes = plt.subplots(3, 3, figsize=(15, 10))
axes = axes.flatten()
```

```
# Loop through each column and plot countplot
for i, col in enumerate(cols):
    sns.countplot(x=col, data=df, ax=axes[i], hue='Churn')
    axes[i].set_title(col, fontsize=12)
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')

# Remove empty subplot if number of plots < total grid cells
for j in range(len(cols), len(axes)):
    fig.delaxes(axes[j])

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

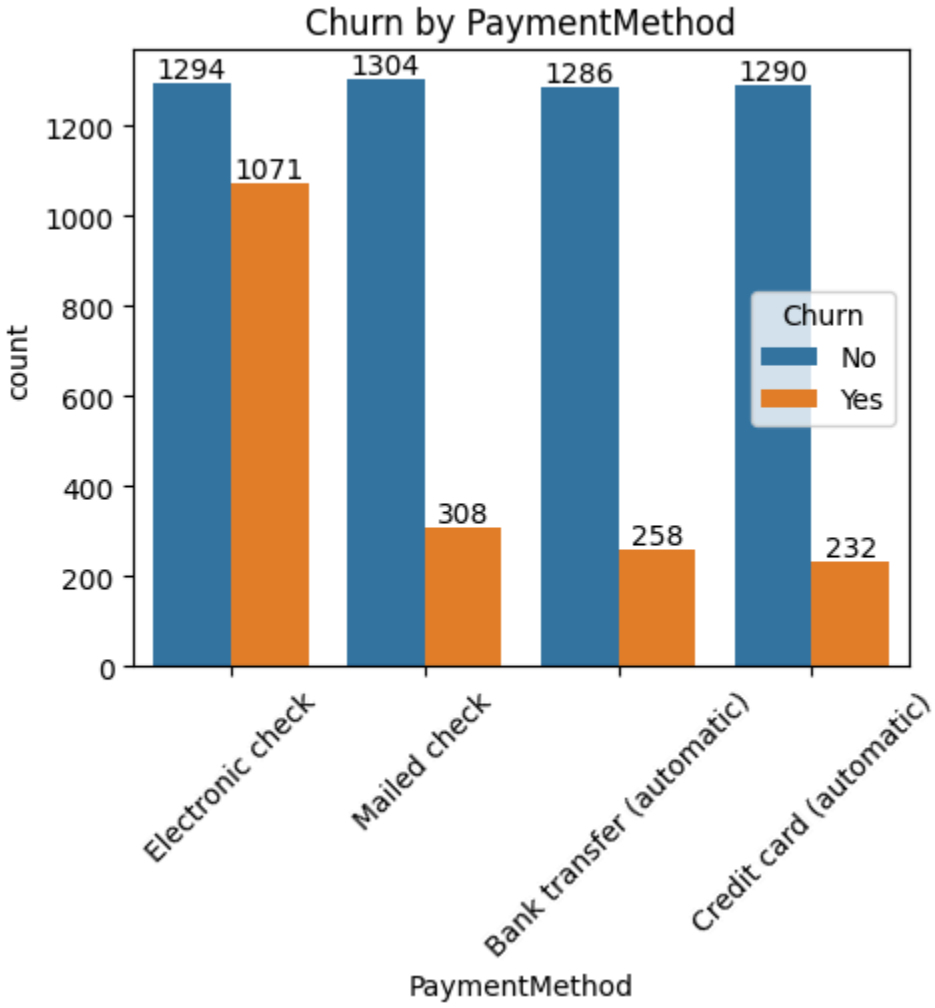


Those using **Fiber optic** internet also have a higher churn rate than DSL users. Overall, customers lacking additional services tend to leave more often, while users with multiple services are generally more loyal.

In [46]:
```
df.PaymentMethod.values
```

Out[46]:
```
array(['Electronic check', 'Mailed check', 'Mailed check', ...,
       'Electronic check', 'Mailed check', 'Bank transfer (automatic)'],
      shape=(7043,), dtype=object)
```

In [66]:
```
plt.figure(figsize = (5,4))
ax=sns.countplot(x = "PaymentMethod" ,data = df, hue ="Churn")
plt.title("Churn by PaymentMethod")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.xticks(rotation = 45)
plt.show()
```



Customer is likely to churn when he is using Electronic check as a pyment method

In [ ]: