In [102…

```
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
```

Out[102…

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
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 [61]:

```
df.head()
```

Out[61]:

	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	

5 rows × 21 columns

In [62]:
```
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 [63]:
```
#replacing blanks with 0 as tenure is 0 and no total charges recorded
```

In [64]:
```
df["TotalCharges"]=df["TotalCharges"].replace(" ","0")
df["TotalCharges"]=df["TotalCharges"].astype("float")
```

In [65]:
```
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 [66]:
```
df.isnull()
```

Out[66]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
7038	False	False	False	False	False	False	False
7039	False	False	False	False	False	False	False
7040	False	False	False	False	False	False	False
7041	False	False	False	False	False	False	False
7042	False	False	False	False	False	False	False

7043 rows × 21 columns

In [67]:
```
df.isnull().sum()
```

Out[67]:
```
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
```

In [68]:
```
df.isnull().sum().sum()
```

Out[68]:
```
0
```

In [69]:
```
df.describe()
```

Out[69]:

	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 [70]:
```
df["customerID"].duplicated().sum()
```

Out[70]:
```
0
```

In [71]:
```
#converted 0 and 1 values of SeniorCitizen  to yes/no to make it easier
```

In [72]:
```
def conv(value):
    if value==1:
        return"yes"
    else:
        return"no"
df['SeniorCitizen'] = df["SeniorCitizen"].apply(conv)
```

In [73]:
```
df.head(25)
```

Out[73]:

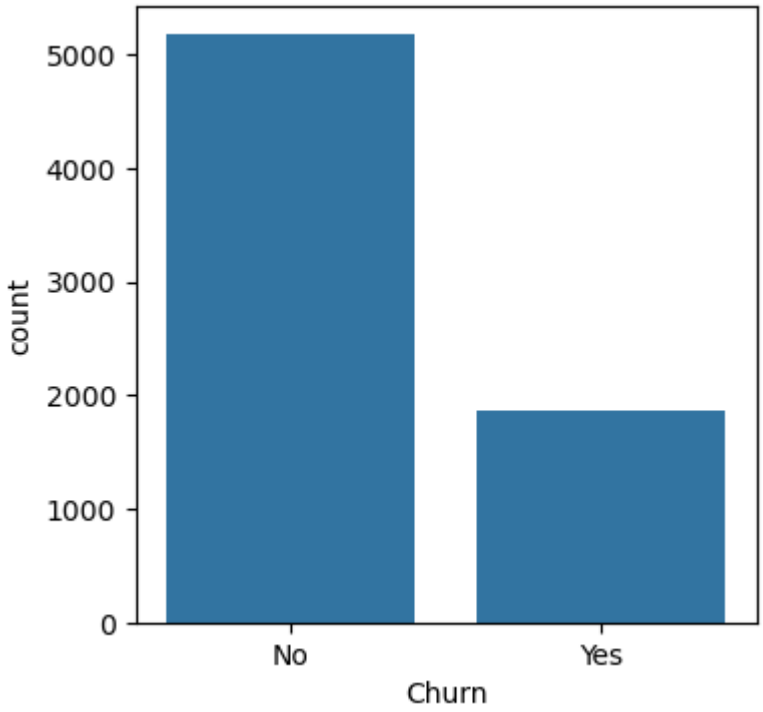
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mu
0	7590-VHVEG	Female	no	Yes	No	1	No	
1	5575-GNVDE	Male	no	No	No	34	Yes	
2	3668-QPYBK	Male	no	No	No	2	Yes	
3	7795-CFOCW	Male	no	No	No	45	No	
4	9237-HQITU	Female	no	No	No	2	Yes	
5	9305-CDSKC	Female	no	No	No	8	Yes	
6	1452-KIOVK	Male	no	No	Yes	22	Yes	
7	6713-OKOMC	Female	no	No	No	10	No	
8	7892-POOKP	Female	no	Yes	No	28	Yes	
9	6388-TABGU	Male	no	No	Yes	62	Yes	
10	9763-GRSKD	Male	no	Yes	Yes	13	Yes	
11	7469-LKBCI	Male	no	No	No	16	Yes	
12	8091-TTVAX	Male	no	Yes	No	58	Yes	
13	0280-XJGEX	Male	no	No	No	49	Yes	
14	5129-JLPIS	Male	no	No	No	25	Yes	
15	3655-SNQYZ	Female	no	Yes	Yes	69	Yes	
16	8191-XWSZG	Female	no	No	No	52	Yes	

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mu
17	9959-WOFKT	Male	no	No	Yes	71	Yes	
18	4190-MFLUW	Female	no	Yes	Yes	10	Yes	
19	4183-MYFRB	Female	no	No	No	21	Yes	
20	8779-QRDMV	Male	yes	No	No	1	No	
21	1680-VDCWW	Male	no	Yes	No	12	Yes	
22	1066-JKSGK	Male	no	No	No	1	Yes	
23	3638-WEABW	Female	no	Yes	No	58	Yes	
24	6322-HRPFA	Male	no	Yes	Yes	49	Yes	

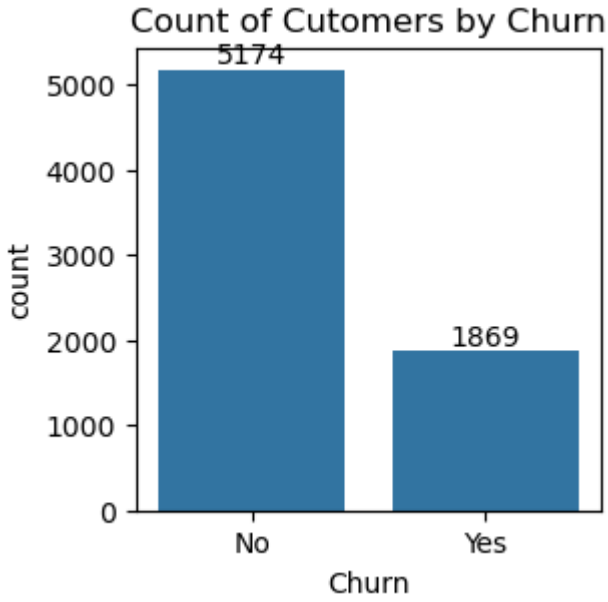
25 rows × 21 columns

In [74]:
```
# why customer stop using services
# why customer churnout
# characterstics of customer
```

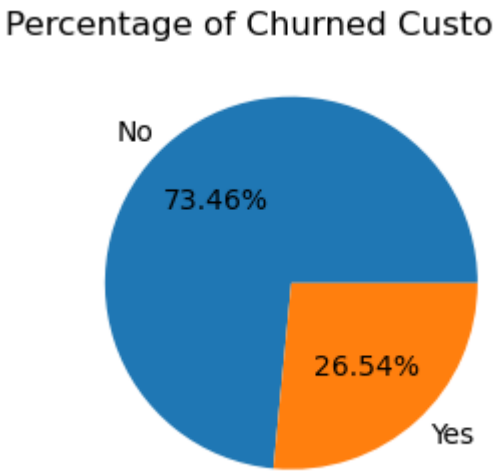
In [75]:
```
plt.figure(figsize=(4,4))
sns.countplot(x= 'Churn',data=df)
plt.show()
```



In [76]:
```
plt.figure(figsize=(3,3))
ax=sns.countplot(x= 'Churn',data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Cutomers by Churn")
plt.show()
```

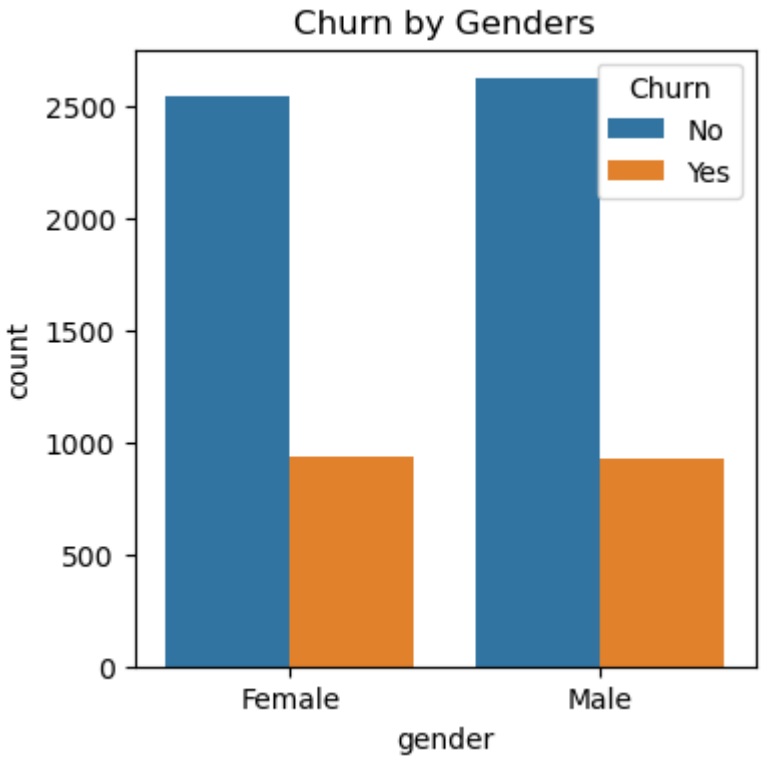


In [77]:
```
plt.figure(figsize =(3,4))
gb=df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'],labels=gb.index,autopct="%1.2f%%")
plt.title("Percentage of Churned Customers")
plt.show()
```

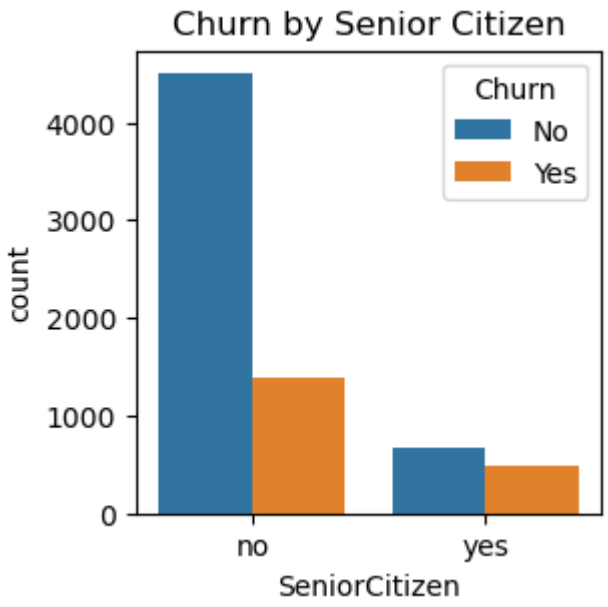


NOW LETS EXPLORE REASON BEHIND CHURN OUT

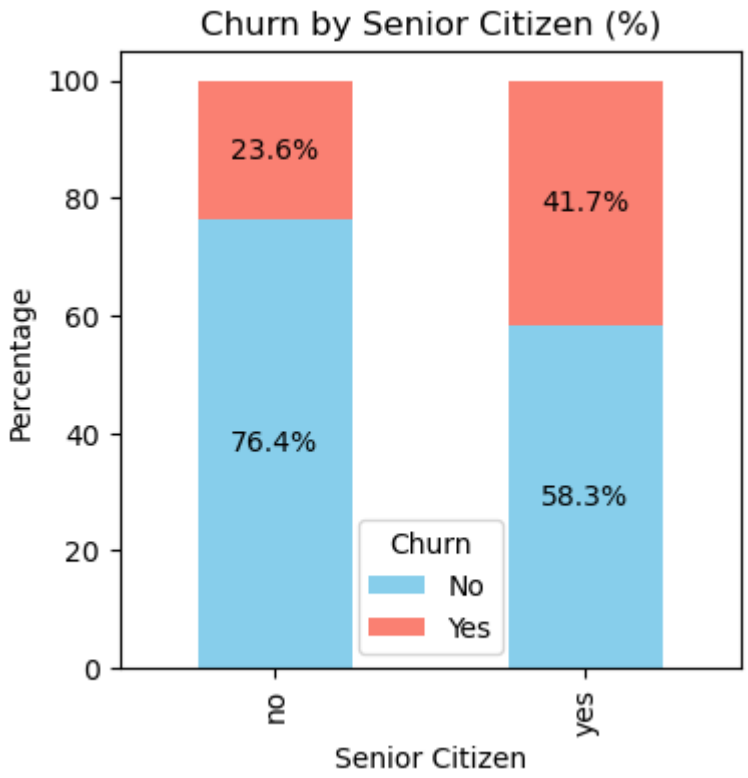
In [79]:
```
plt.figure(figsize=(4,4))
sns.countplot(x="gender",data=df,hue="Churn")
plt.title("Churn by Genders")
plt.show()
```



In [80]:
```
plt.figure(figsize=(3,3))
sns.countplot(x="SeniorCitizen",data=df,hue="Churn")
plt.title("Churn by Senior Citizen")
plt.show()
```

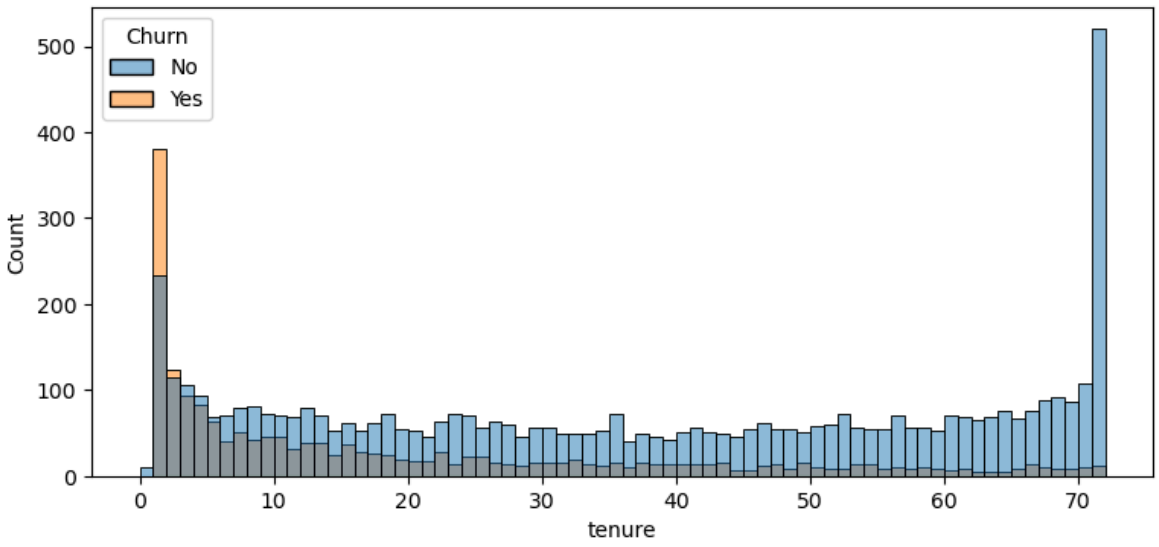


In [82]:
```
counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)

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

# Plot stacked bar chart
percentages.plot(kind='bar', stacked=True, figsize=(4,4), color=['skyblue', 'sal

# Add percentage labels
for i, (senior, row) in enumerate(percentages.iterrows()):
    bottom = 0
    for val in row:
        plt.text(i, bottom + val/2, f'{val:.1f}%', ha='center', va='center')
        bottom += val

plt.title('Churn by Senior Citizen (%)')
plt.ylabel('Percentage')
plt.xlabel('Senior Citizen')
plt.legend(title='Churn')
plt.show()
```



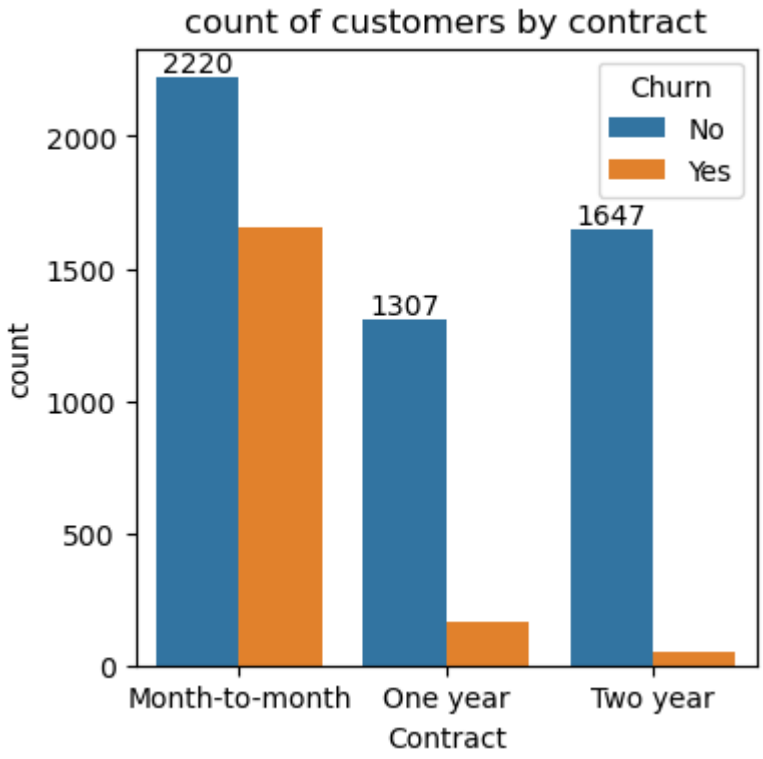
comparative a greated percentage of people in senior citixen category have churned

In [91]:
```
plt.figure(figsize=(9,4))
sns.histplot(x="tenure" ,data=df,bins=72, hue="Churn")
plt.show()
```



people who have month-to-month contract are likely to churn thrn from thode who have 1 or 2 year of contract

In [94]:
```
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()
```

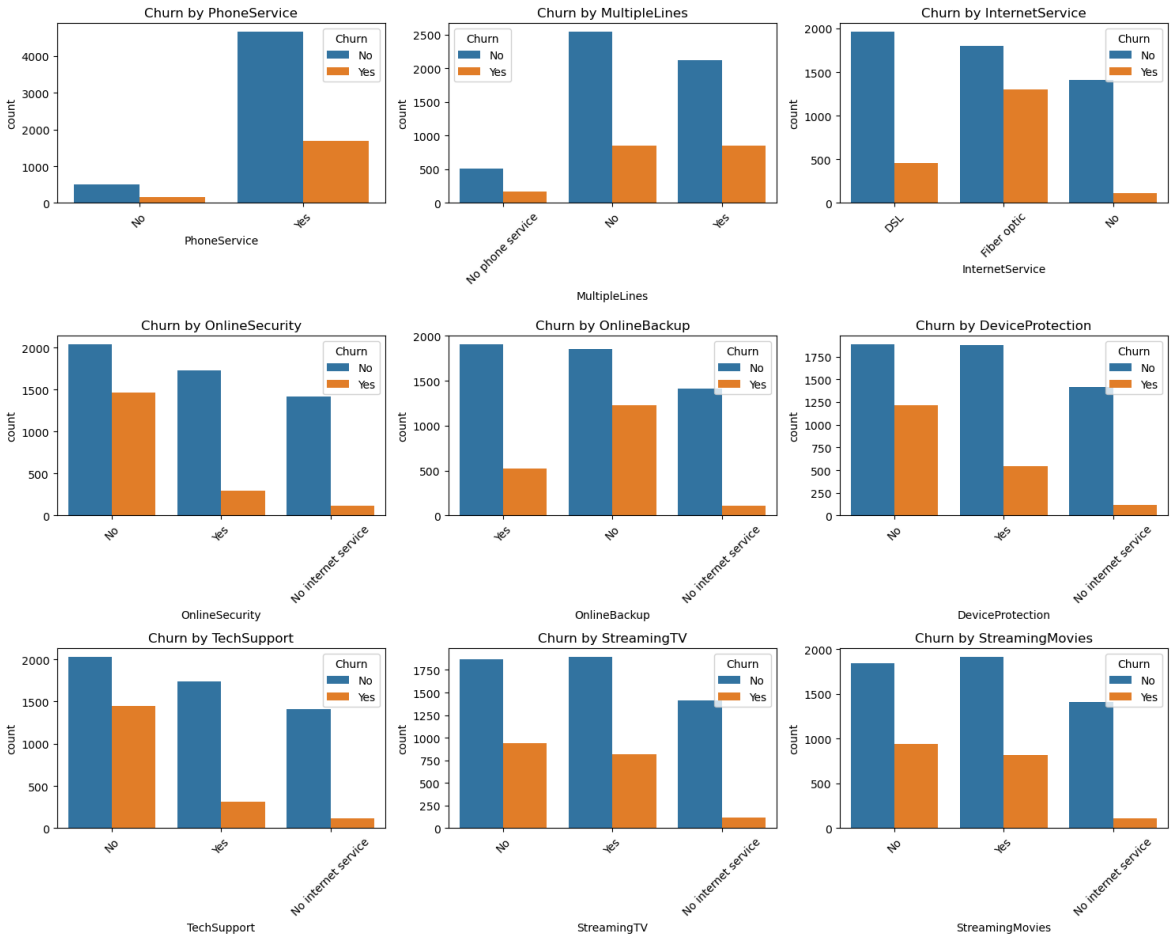


In [95]:
```
df.columns.values
```

Out[95]:
```
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 [99]:
```
cols = ['PhoneService', 'MultipleLines', 'InternetService',
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies']

# Define grid size
n_cols = 3  # Number of plots per row
n_rows = (len(cols) + n_cols - 1) // n_cols  # Compute rows needed

# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(n_cols*5, n_rows*4))
axes = axes.flatten()  # Flatten in case of multiple rows

for i, col in enumerate(cols):
    sns.countplot(x=col, hue='Churn', data=df, ax=axes[i])
    axes[i].set_title(f'Churn by {col}')
    axes[i].tick_params(axis='x', rotation=45)

# Remove any empty subplots
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])
```

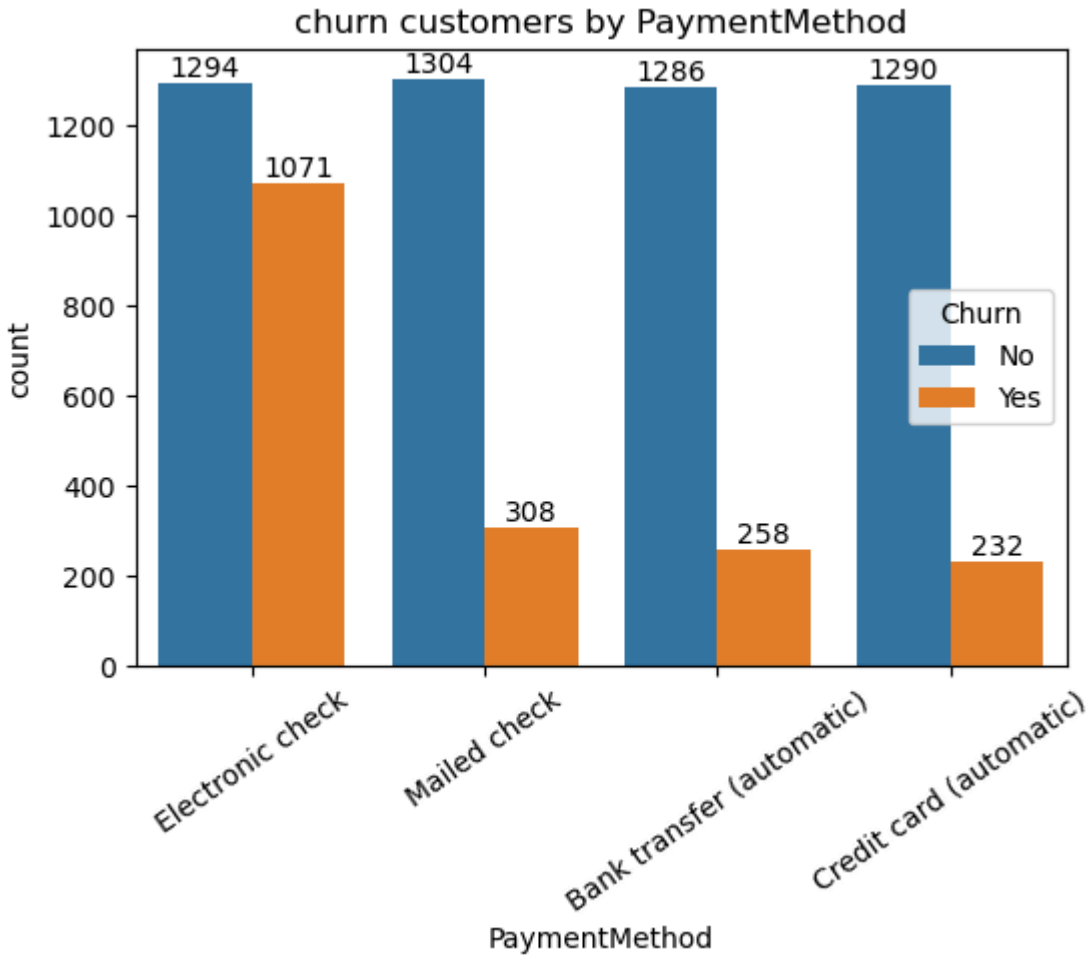
```
plt.tight_layout()
plt.show()
```



Customers with internet-related services (OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport) show higher churn rates when they don't subscribe to these features. PhoneService alone does not significantly impact churn. Streaming services (TV and Movies) show relatively balanced churn behavior.

In [105…

```
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("churn customers by PaymentMethod")
plt.xticks(rotation=35)
plt.show()
```


churn customers by PaymentMethod
1294
1071
1304
308
1286
258
1290
232
1200
1000
800
600
400
200
0
count
Churn
No
Yes
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod

In [ ]: