In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
data=pd.read_csv("OneDrive/Customer Churn (1).csv")
data
```

Out[1]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleL
0	7590-VHVEG	Female	0	Yes	No	1	No	No ph ser
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	No ph ser
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	No ph ser
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

7043 rows × 21 columns

In [3]:
```
data.head()
```

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service
1	5575-GNVDE	Male	0	No	No	34	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service
4	9237-HQITU	Female	0	No	No	2	Yes	No

5 rows × 21 columns

In [7]:
```
data.tail()
```

Out[7]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleL
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	No ph ser
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

5 rows × 21 columns

In [9]:
```
data.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 [11]:
```
data.count()                                                    #count
```

Out[11]:
```
customerID          7043
gender              7043
SeniorCitizen       7043
Partner             7043
Dependents          7043
tenure              7043
PhoneService        7043
MultipleLines       7043
InternetService     7043
OnlineSecurity      7043
OnlineBackup        7043
DeviceProtection    7043
TechSupport         7043
StreamingTV         7043
StreamingMovies     7043
Contract            7043
PaperlessBilling    7043
PaymentMethod       7043
MonthlyCharges      7043
TotalCharges        7043
Churn               7043
dtype: int64
```

In [13]:
```
data.isnull()
```

Out[13]:

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

7043 rows × 21 columns

In [15]:
```
data.isnull().sum()                                    #find all null values
```

Out[15]:
```
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 [17]:
```
data.describe()                                    #describe data
```

Out[17]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

In [19]:
```
data["TotalCharges"]=data["TotalCharges"].replace(" ","0")                # r
```

In [21]:
```
data
```

Out[21]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleL
0	7590-VHVEG	Female	0	Yes	No	1	No	No ph ser
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	No ph ser
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	No ph ser
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	
7042	3186-AJIEK	Male	0	No	No	66	Yes	

7043 rows × 21 columns

In [ ]:
```
# convert 0/1 values of senior citizens into yes/ no
```

In [29]:
```
def conv(value):
    if value==1:
        return "yes"
    else :
        return "No"
data["SeniorCitizen"]=data["SeniorCitizen"].apply(conv)
```

In [31]:
```
data.head(25)
```

Out[31]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLine
0	7590-VHVEG	Female	No	Yes	No	1	No	No phon servic
1	5575-GNVDE	Male	No	No	No	34	Yes	N
2	3668-QPYBK	Male	No	No	No	2	Yes	N
3	7795-CFOCW	Male	No	No	No	45	No	No phon servic
4	9237-HQITU	Female	No	No	No	2	Yes	N
5	9305-CDSKC	Female	No	No	No	8	Yes	Ye
6	1452-KIOVK	Male	No	No	Yes	22	Yes	Ye
7	6713-OKOMC	Female	No	No	No	10	No	No phon servic
8	7892-POOKP	Female	No	Yes	No	28	Yes	Ye
9	6388-TABGU	Male	No	No	Yes	62	Yes	N
10	9763-GRSKD	Male	No	Yes	Yes	13	Yes	N
11	7469-LKBCI	Male	No	No	No	16	Yes	N
12	8091-TTVAX	Male	No	Yes	No	58	Yes	Ye
13	0280-XJGEX	Male	No	No	No	49	Yes	Ye
14	5129-JLPIS	Male	No	No	No	25	Yes	N
15	3655-SNQYZ	Female	No	Yes	Yes	69	Yes	Ye
16	8191-XWSZG	Female	No	No	No	52	Yes	N
17	9959-WOFKT	Male	No	No	Yes	71	Yes	Ye
18	4190-MFLUW	Female	No	Yes	Yes	10	Yes	N
19	4183-MYFRB	Female	No	No	No	21	Yes	N
20	8779-QRDMV	Male	yes	No	No	1	No	No phon servic
21	1680-VDCWW	Male	No	Yes	No	12	Yes	N

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLine
22	1066-JKSGK	Male	No	No	No	1	Yes	N
23	3638-WEABW	Female	No	Yes	No	58	Yes	Ye
24	6322-HRPFA	Male	No	Yes	Yes	49	Yes	N

25 rows × 21 columns

In [33]:
```
data["customerID"].duplicated()                    #find duplicate from data
```

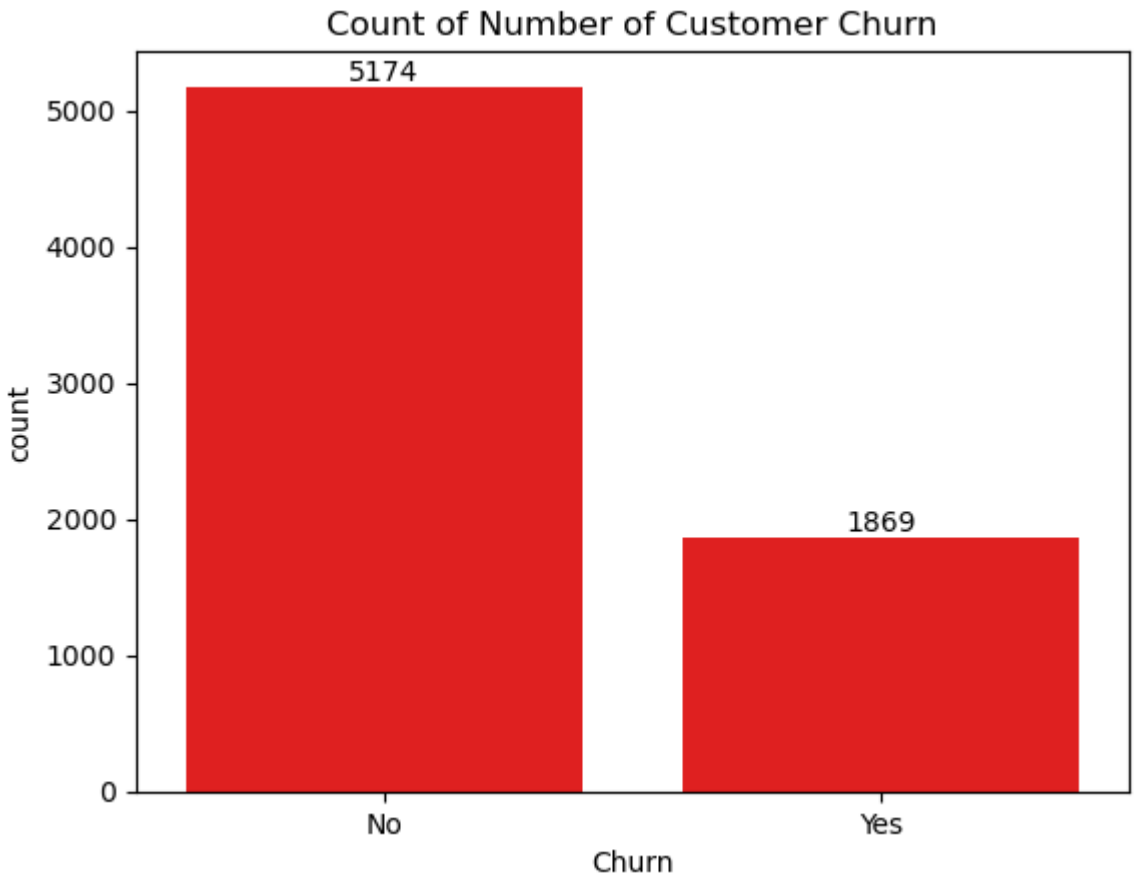
Out[33]:
```
0       False
1       False
2       False
3       False
4       False
        ...
7038    False
7039    False
7040    False
7041    False
7042    False
Name: customerID, Length: 7043, dtype: bool
```

In [35]:
```
data["customerID"].duplicated().sum()                    #find sum of dupli
```

Out[35]:
```
0
```

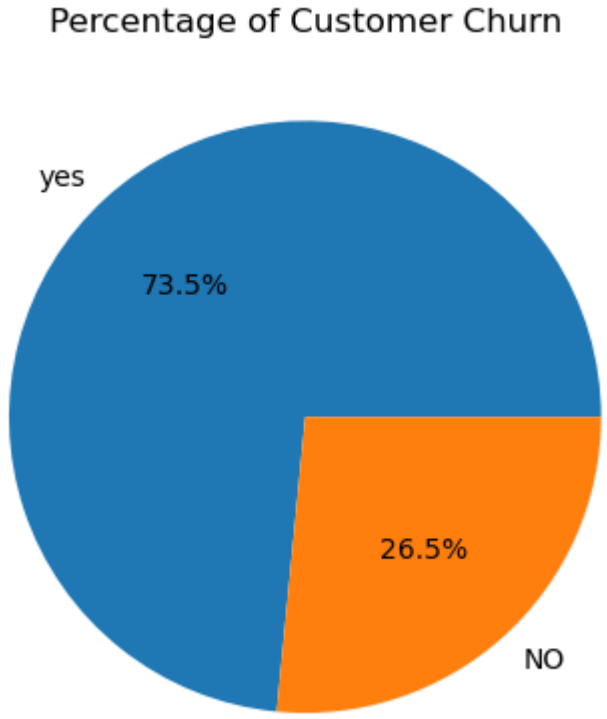
In [110…
```
ax=sns.countplot(x='Churn',data=data,color="red")                    #no.
ax.bar_label(ax.containers[0])
plt.title("Count of Number of Customer Churn")
#counting overall churn
```

Out[110…
```
Text(0.5, 1.0, 'Count of Number of Customer Churn')
```



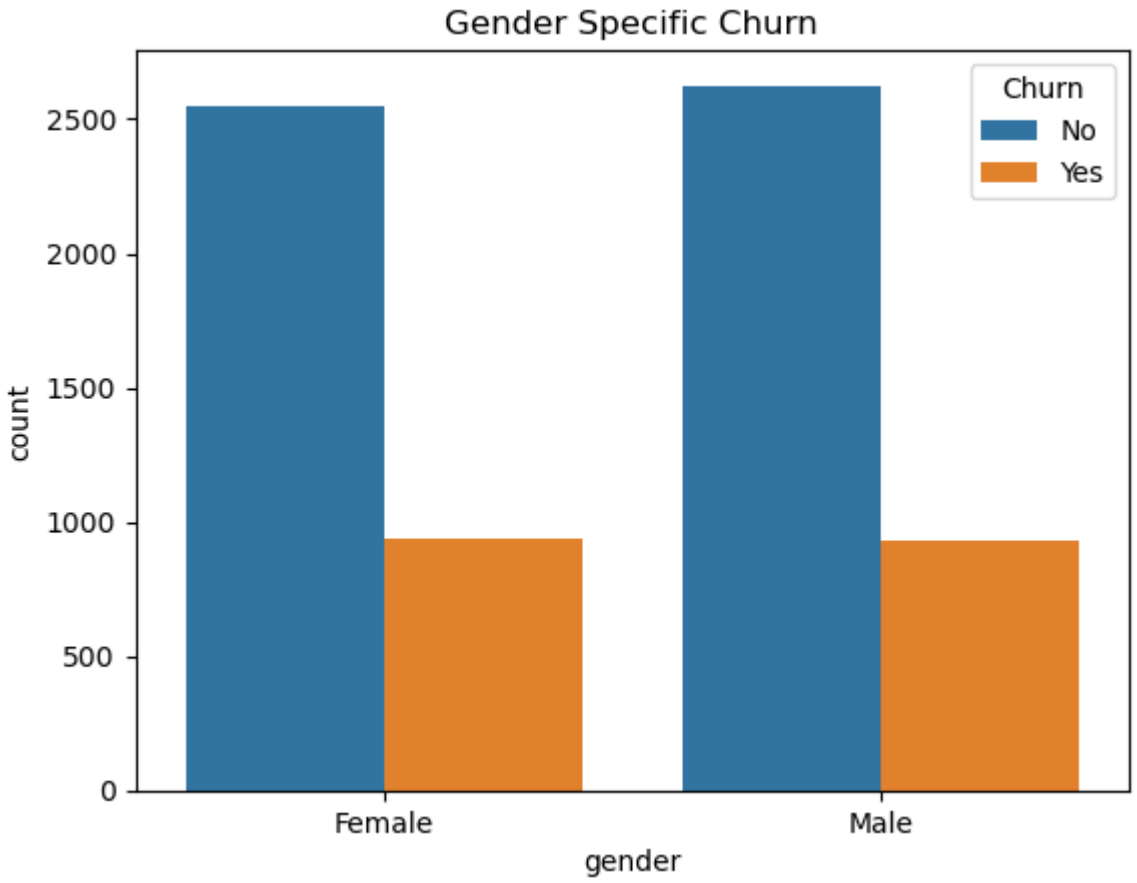
In [104…

```
gb=data.groupby("Churn").agg({"Churn":"count"})
gb
plt.pie(gb['Churn'],labels=['yes','NO'],autopct='%0.1f%%')
plt.title("Percentage of Customer Churn")
plt.show()
```



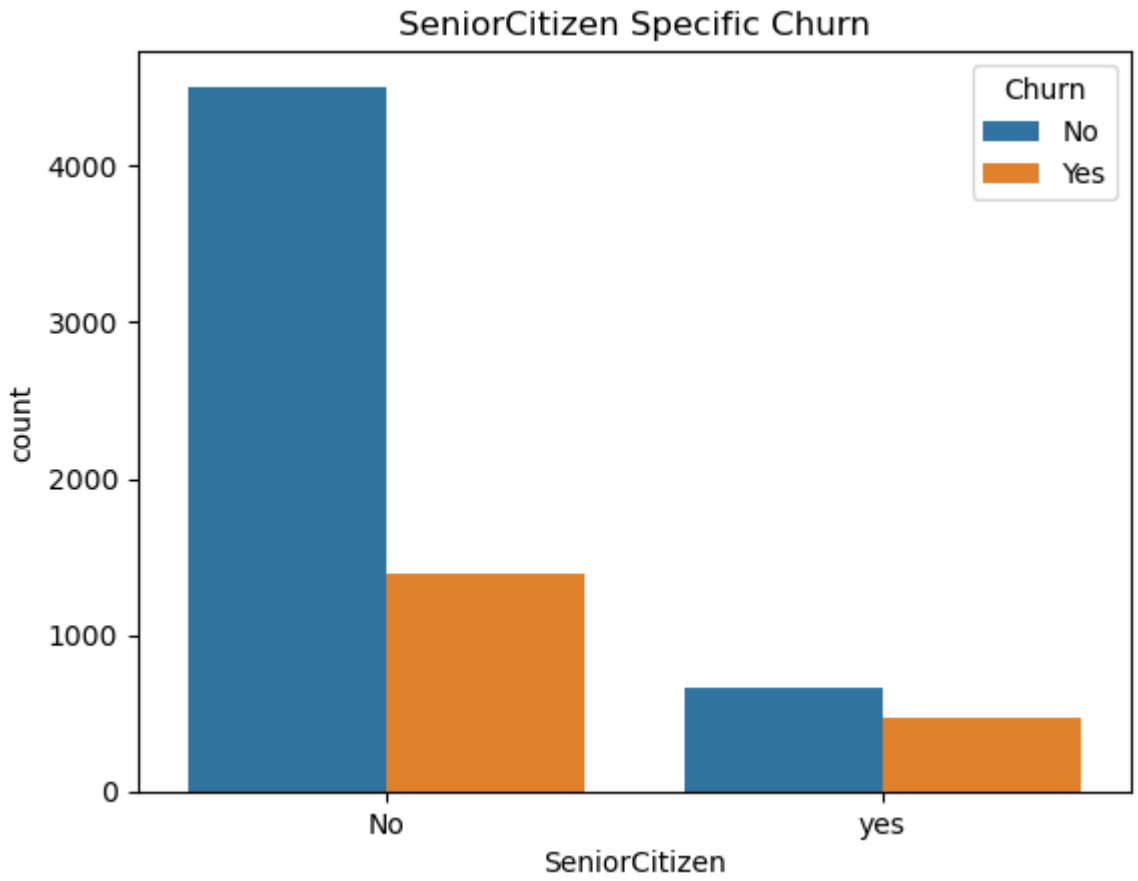
In [142…

```
sns.countplot(x="gender",data=data,hue="Churn")                    #no of male /f
plt.title("Gender Specific Churn")
plt.show()
```



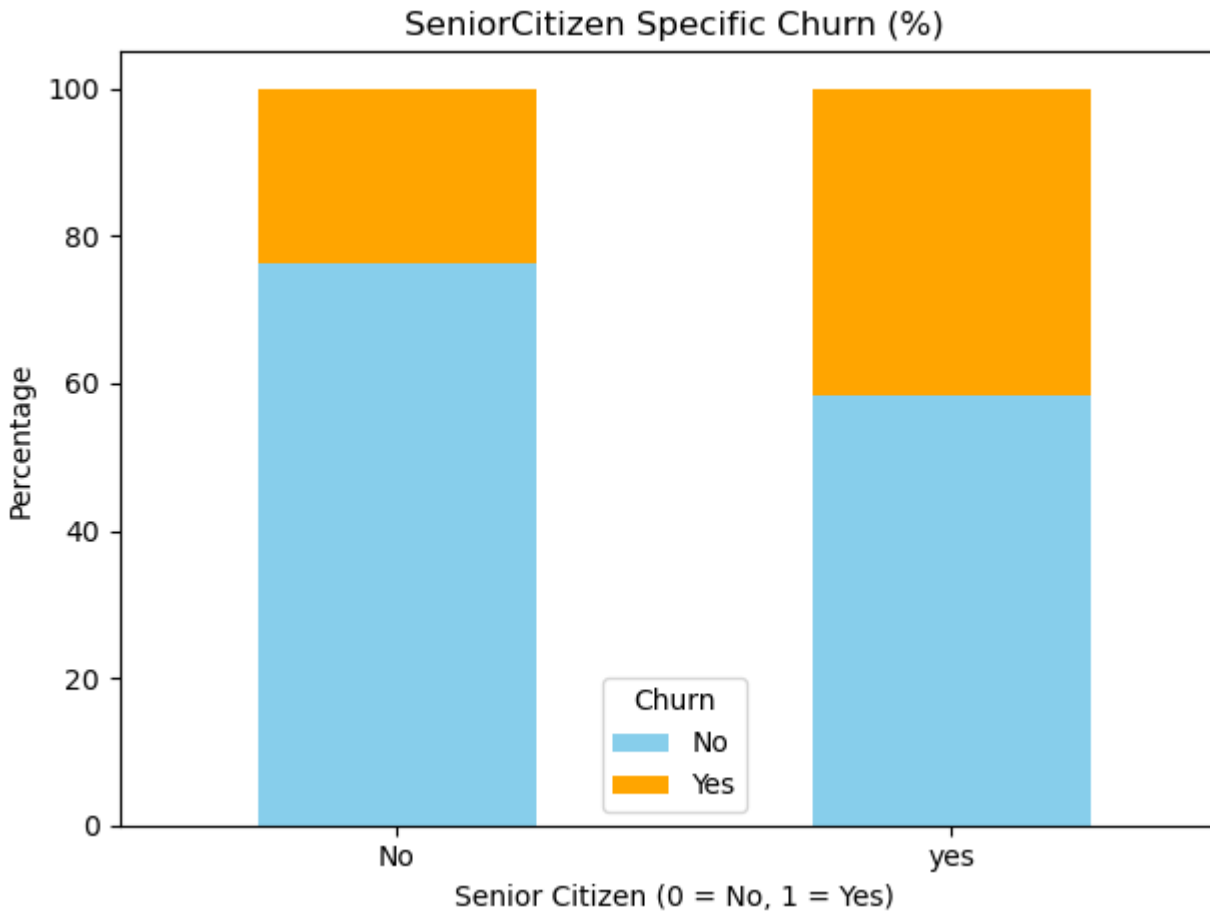
In [144…

```
sns.countplot(x="SeniorCitizen",data=data,hue="Churn")                    #no of
plt.title("SeniorCitizen Specific Churn")
plt.show()
```



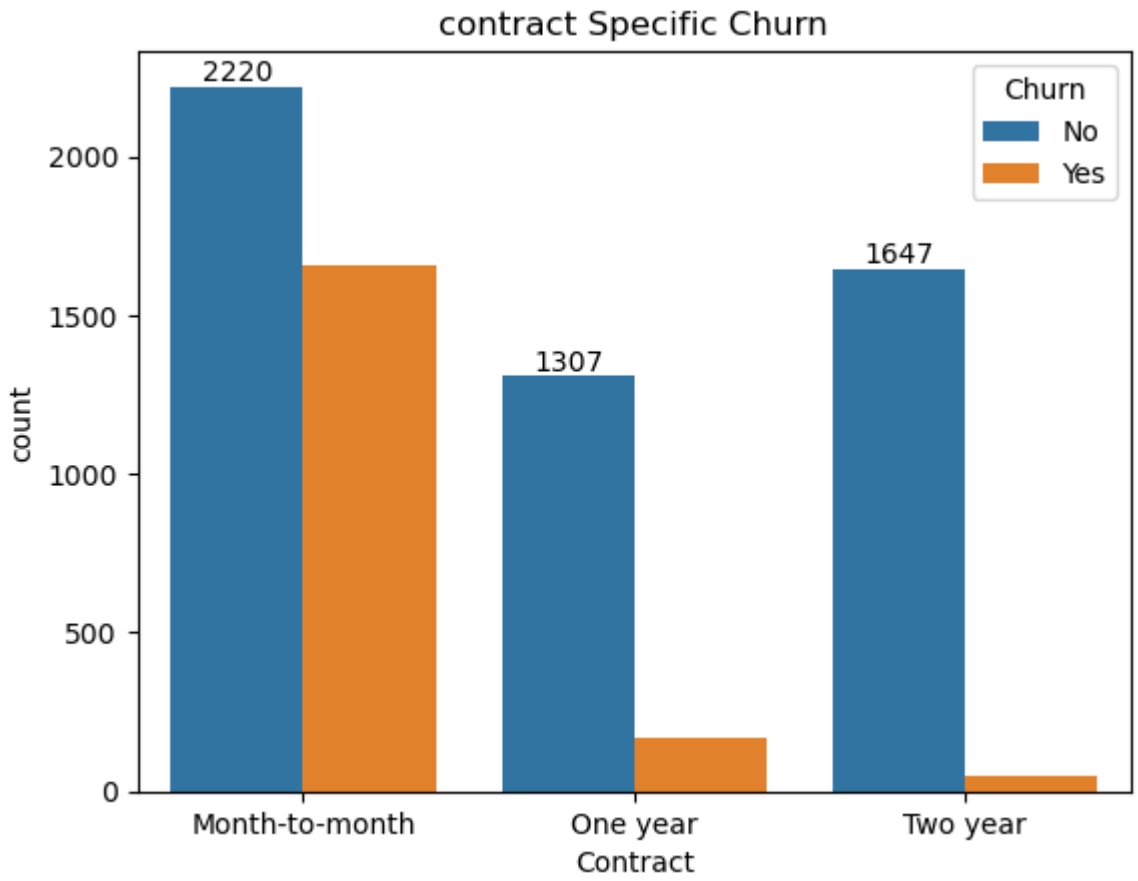
In [160…

```
# Step 1: Create percentage table
table = pd.crosstab(data['SeniorCitizen'], data['Churn'], normalize='index') * 1

# Step 2: Plot stacked bar chart
table.plot(kind='bar', stacked=True, color=['skyblue', 'orange'])

# Step 3: Chart formatting
plt.title("SeniorCitizen Specific Churn (%)")
plt.xlabel("Senior Citizen (0 = No, 1 = Yes)")
plt.ylabel("Percentage")
plt.legend(title='Churn')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



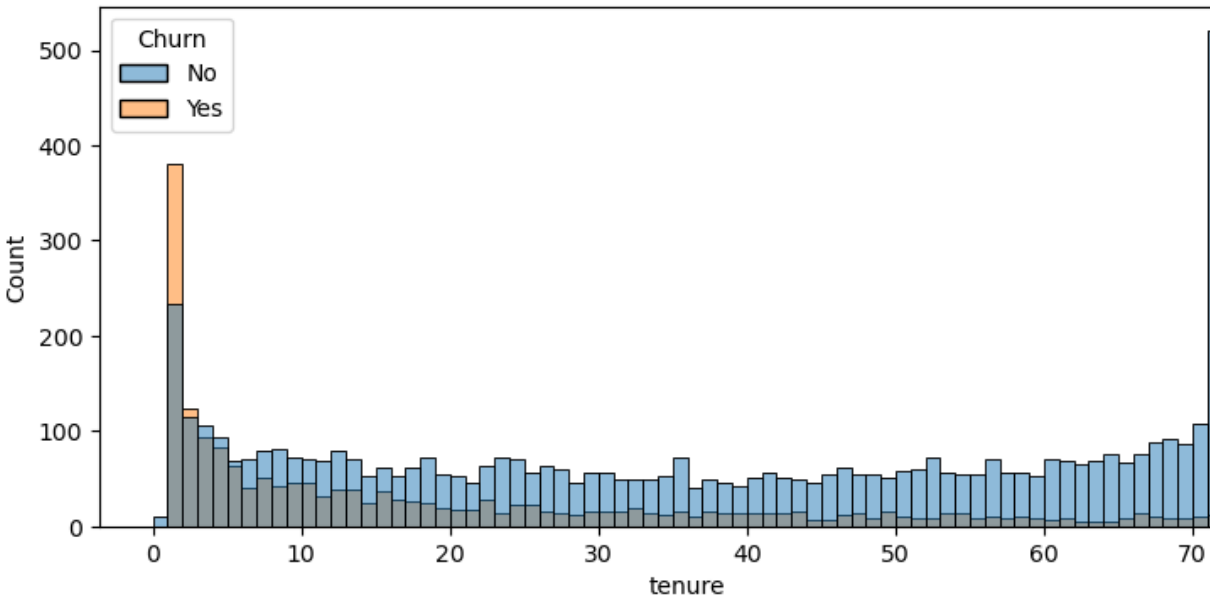
In [204…

```
ax=sns.countplot(x="Contract",data=data,hue="Churn")                    #contract
ax.bar_label(ax.containers[0])
plt.title("contract Specific Churn")
plt.show()
```



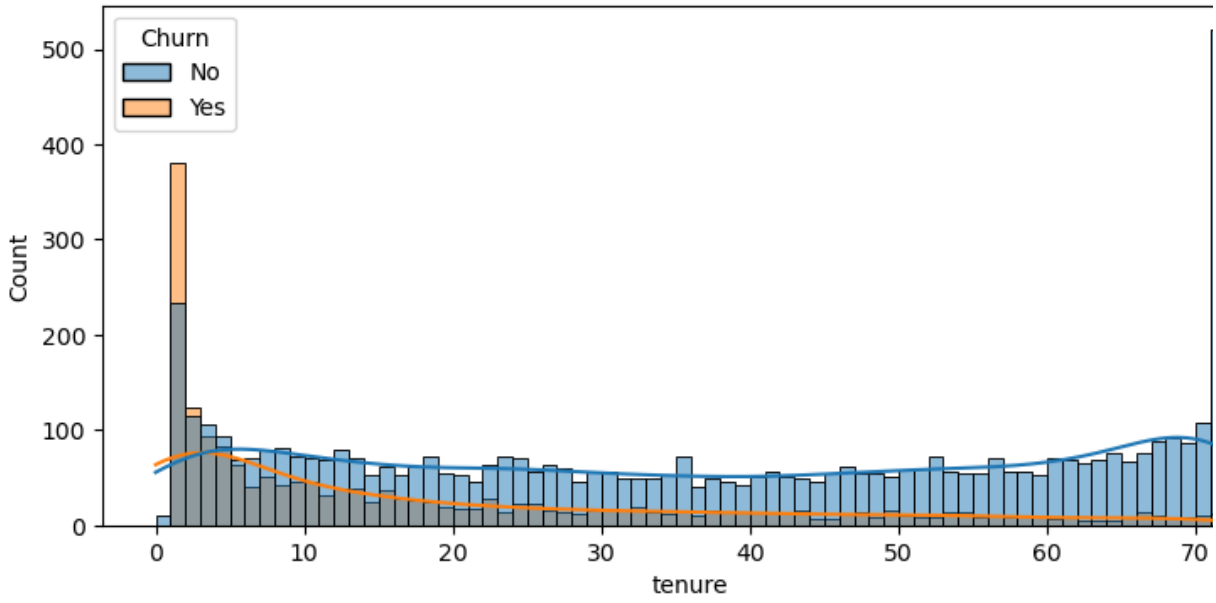
In [200…

```
plt.figure(figsize=(9,4))
sns.histplot(x="tenure",data=data,hue="Churn",bins=72)        #pepole who have lo
plt.show()
```



In [232…

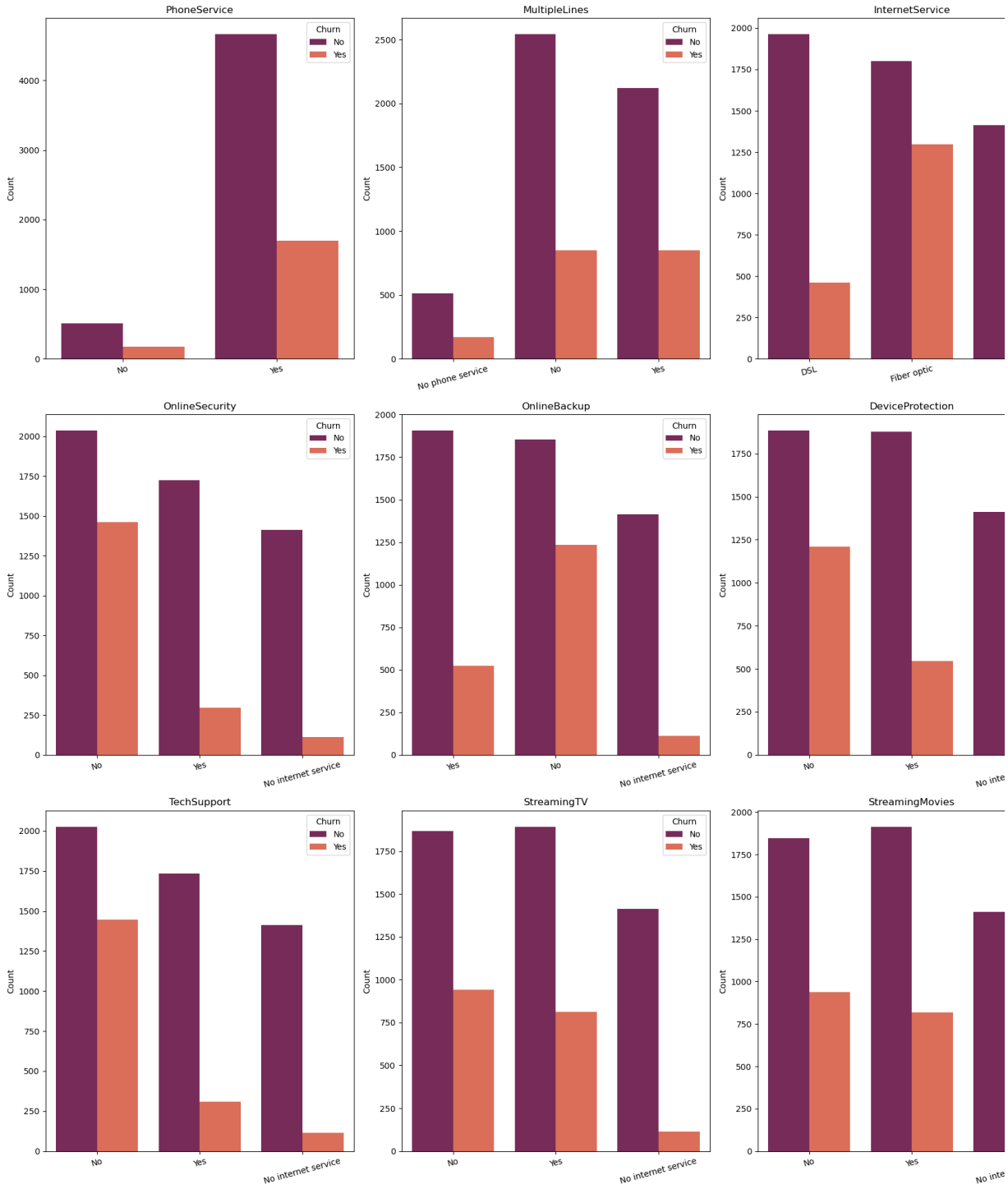
```
plt.figure(figsize=(9,4))
sns.histplot(x="tenure",data=data,hue="Churn",bins=72,kde=True)        #pepole wh
plt.show()
```



In [214…
```
data.columns.values
```

Out[214…
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

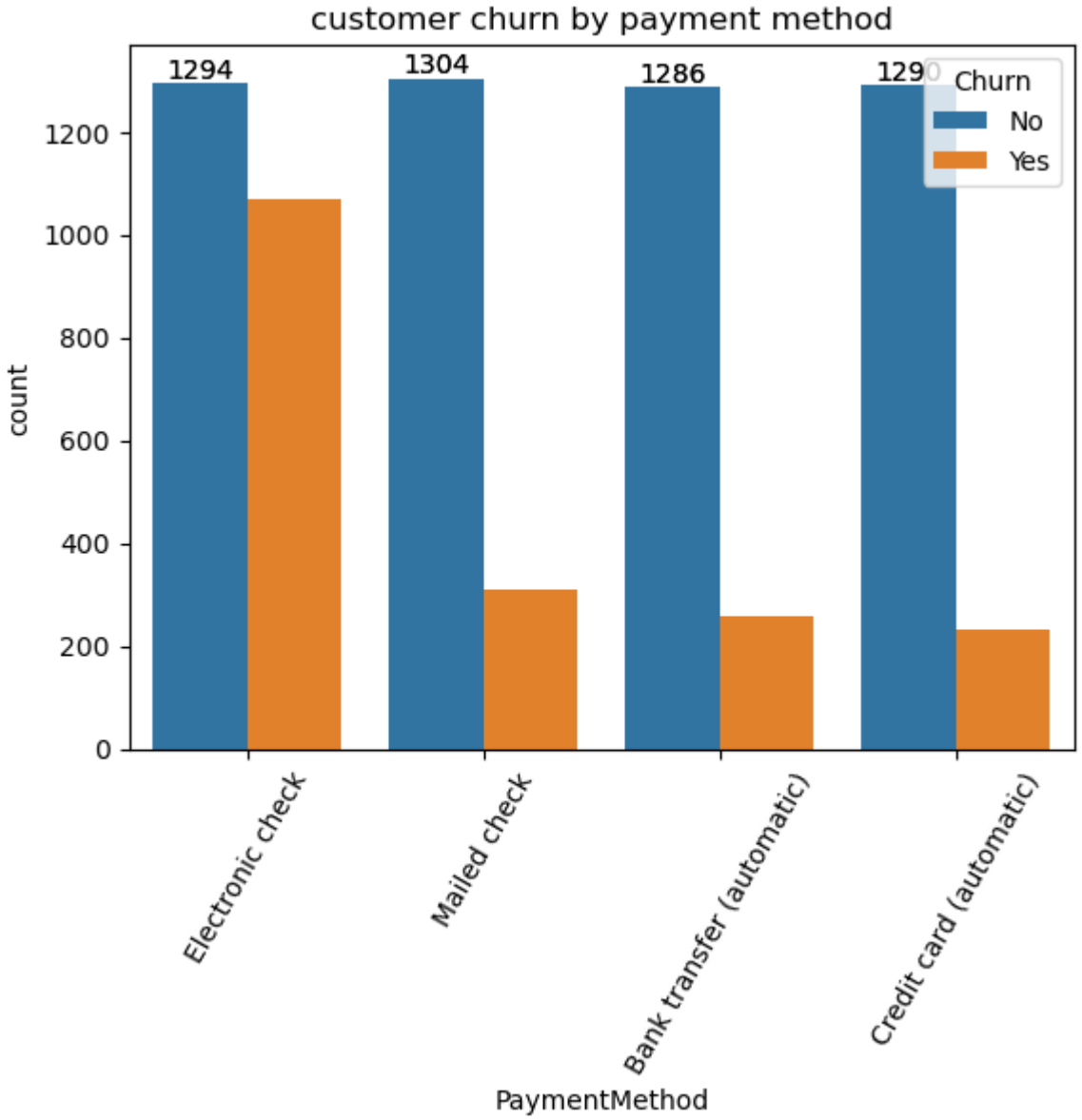
In [222…
```
# Example: Load your DataFrame (replace this with your actual data)
# data = pd.read_csv("your_file.csv")

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

# Setup subplots
plt.figure(figsize=(18, 20))
for i, col in enumerate(cols, 1):
    plt.subplot(3, 3, i)
    sns.countplot(data=data, x=col, palette="rocket",hue="Churn")
    plt.title(col)
    plt.xlabel('')
    plt.ylabel('Count')
    plt.xticks(rotation=15)

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



The subplots display countplots of various serv related features split by customer churn status. shows that customers with no additional servic (like OnlineSecurity, TechSupport, etc.) are mor likely to churn. Features such as StreamingTV, DeviceProtection, and OnlineBackup also show higher churn rates among non-users. This indicates that value-added services may help reduce customer churn.

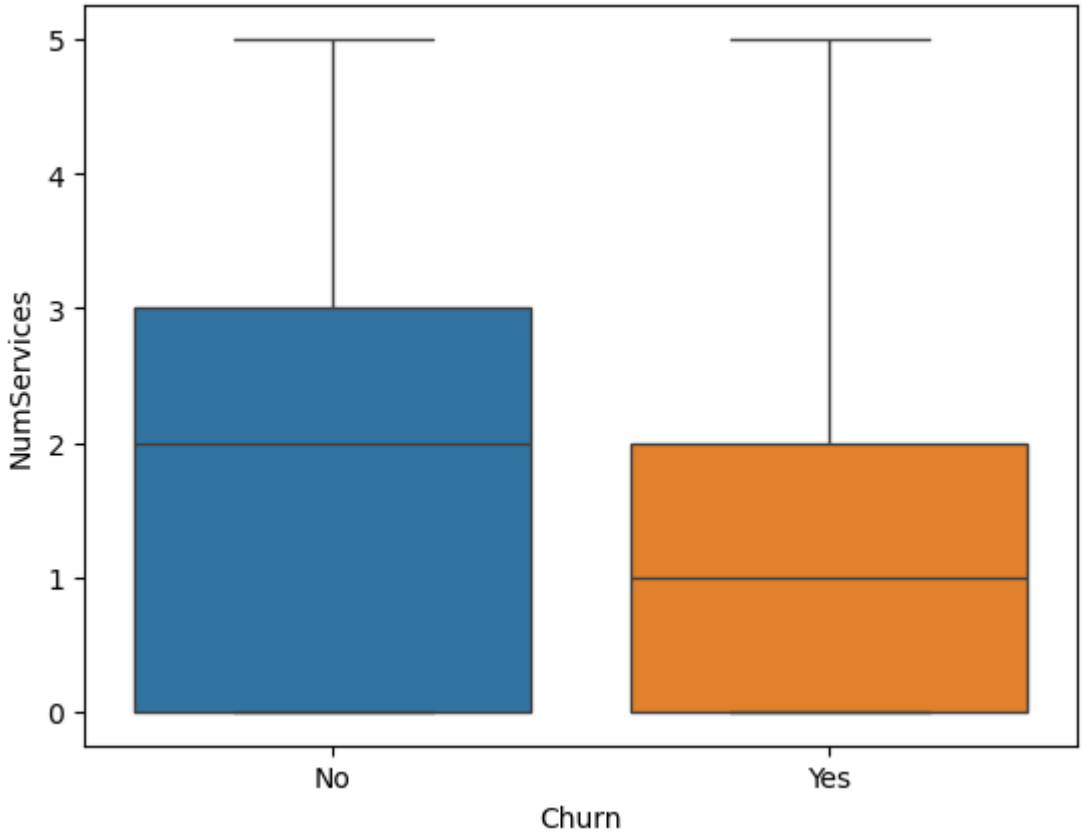
In [228…

```
ax=sns.countplot(x="PaymentMethod",data=data,hue="Churn")                    #cus
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[0])
plt.title("customer churn by payment method")
plt.xticks(rotation=60)
plt.show()
```



In [238…

```
services = ['OnlineSecurity', 'OnlineBackup', 'TechSupport', 'StreamingTV', 'Str
data['NumServices'] = data[services].apply(lambda x: (x == 'Yes').sum(), axis=1)
sns.boxplot(x='Churn', y='NumServices', data=data,hue="Churn")
plt.show()
```



In [ ]: