In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.ticker as mtick
import matplotlib.pyplot as plt
%matplotlib inline
```

In [18]:
```
telco_base_data = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

In [19]:
```
telco_base_data.head()
```

Out[19]:

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

5 rows × 21 columns

In [20]:
```
telco_base_data.shape
```

Out[20]:
```
(7043, 21)
```

In [21]:
```
telco_base_data.columns.values
```

Out[21]:
```
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 [22]:
```
# Checking the data types of all the columns
telco_base_data.dtypes
```

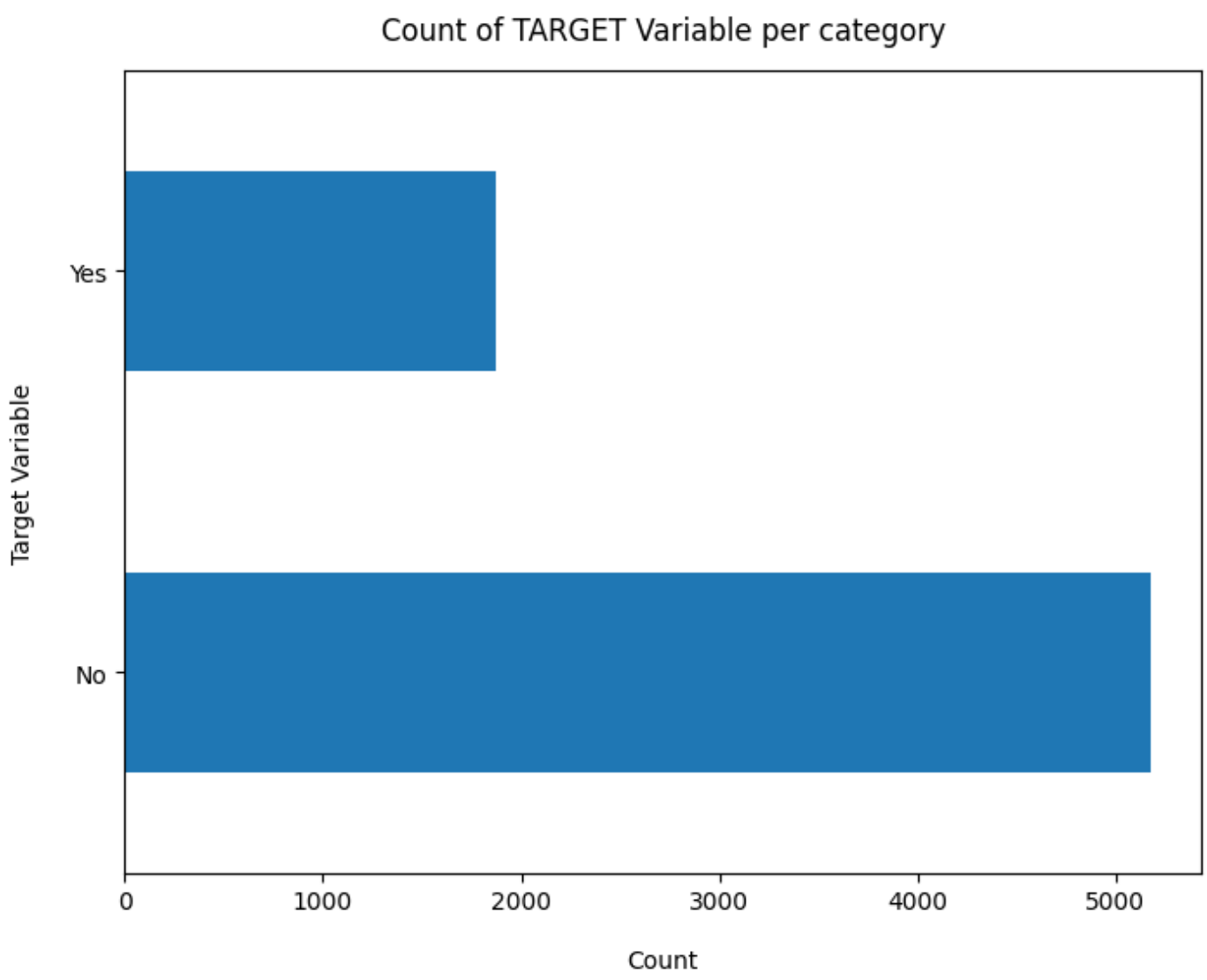
Out[22]:
```
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges         object
Churn                object
dtype: object
```

In [23]:
```
# Check the descriptive statistics of numeric variables
telco_base_data.describe()
```

Out[23]:

	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 [24]:
```
telco_base_data['Churn'].value_counts().plot(kind='barh', figsize=(8, 6))
plt.xlabel("Count", labelpad=14)
plt.ylabel("Target Variable", labelpad=14)
plt.title("Count of TARGET Variable per category", y=1.02);
```



In [25]:
```
100*telco_base_data['Churn'].value_counts()/len(telco_base_data['Churn'])
```

Out[25]:
```
Churn
No     73.463013
Yes    26.536987
Name: count, dtype: float64
```

In [26]:
```
telco_base_data['Churn'].value_counts()
```

Out[26]:
```
Churn
No     5174
Yes    1869
Name: count, dtype: int64
```

In [27]:
```
# Concise Summary of the dataframe, as we have too many columns, we are using the verbose = True
telco_base_data.info(verbose = True)
```

```
<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 [29]:
```
telco_data = telco_base_data.copy()
```

In [30]:
```
telco_data.TotalCharges = pd.to_numeric(telco_data.TotalCharges, errors='coerce')
telco_data.isnull().sum()
```

Out[30]:
```
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       11
Churn               0
dtype: int64
```

In [31]:
```
telco_data.loc[telco_data ['TotalCharges'].isnull() == True]
```

Out[31]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetServ
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	D
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	D
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	D
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	D
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	D

11 rows × 21 columns

In [32]:
```
#Removing missing values
telco_data.dropna(how = 'any', inplace = True)
```

In [33]:
```
print(telco_data['tenure'].max()) #72
```

```
72
```

In [34]:
```
# Group the tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

telco_data['tenure_group'] = pd.cut(telco_data.tenure, range(1, 80, 12), right=False, labels=labe
```

In [35]:
```
telco_data['tenure_group'].value_counts()
```

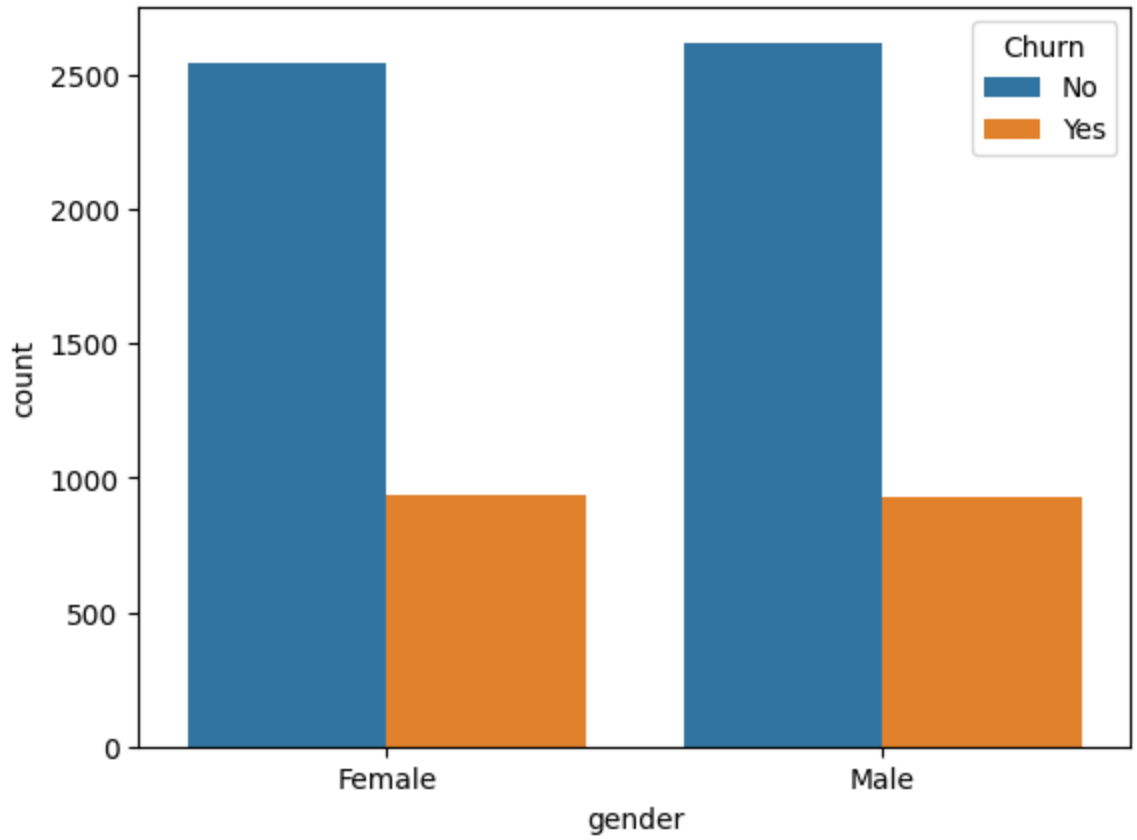
Out[35]:
```
tenure_group
1 - 12     2175
61 - 72    1407
13 - 24    1024
25 - 36     832
49 - 60     832
37 - 48     762
Name: count, dtype: int64
```

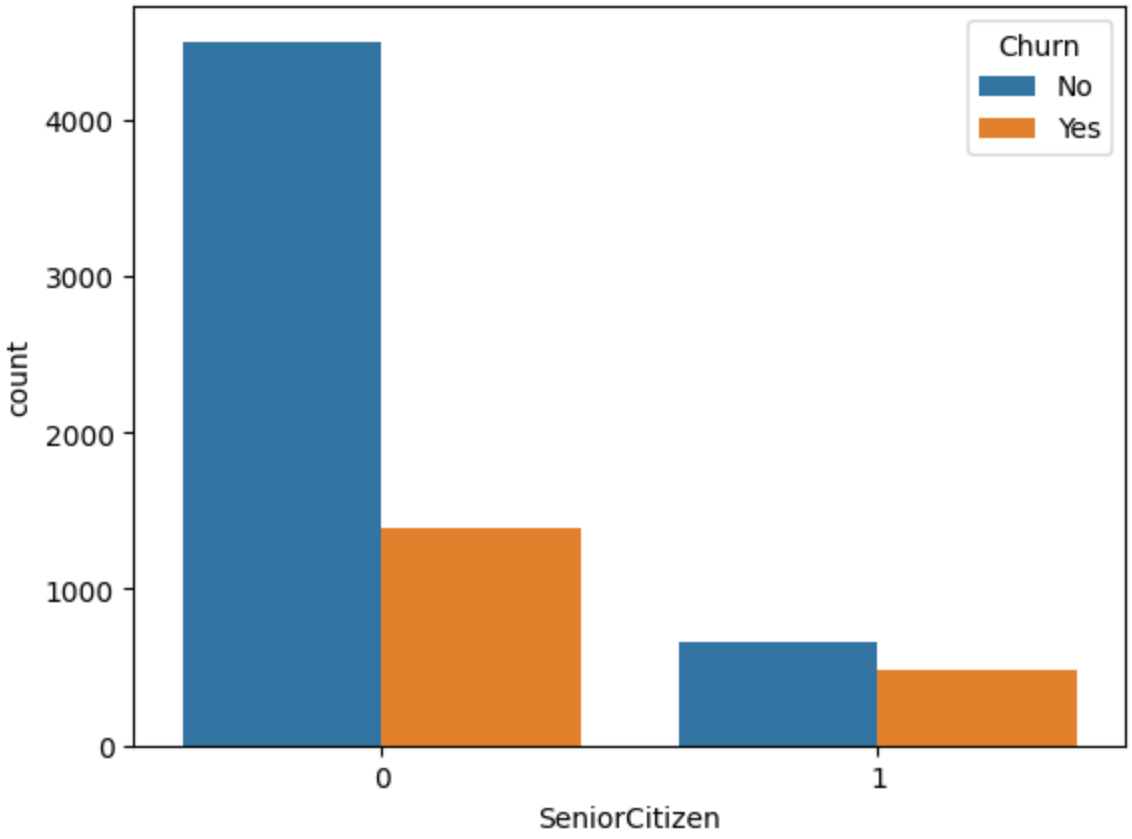
In [36]:
```
#drop column customerID and tenure
telco_data.drop(columns= ['customerID','tenure'], axis=1, inplace=True)
telco_data.head()
```

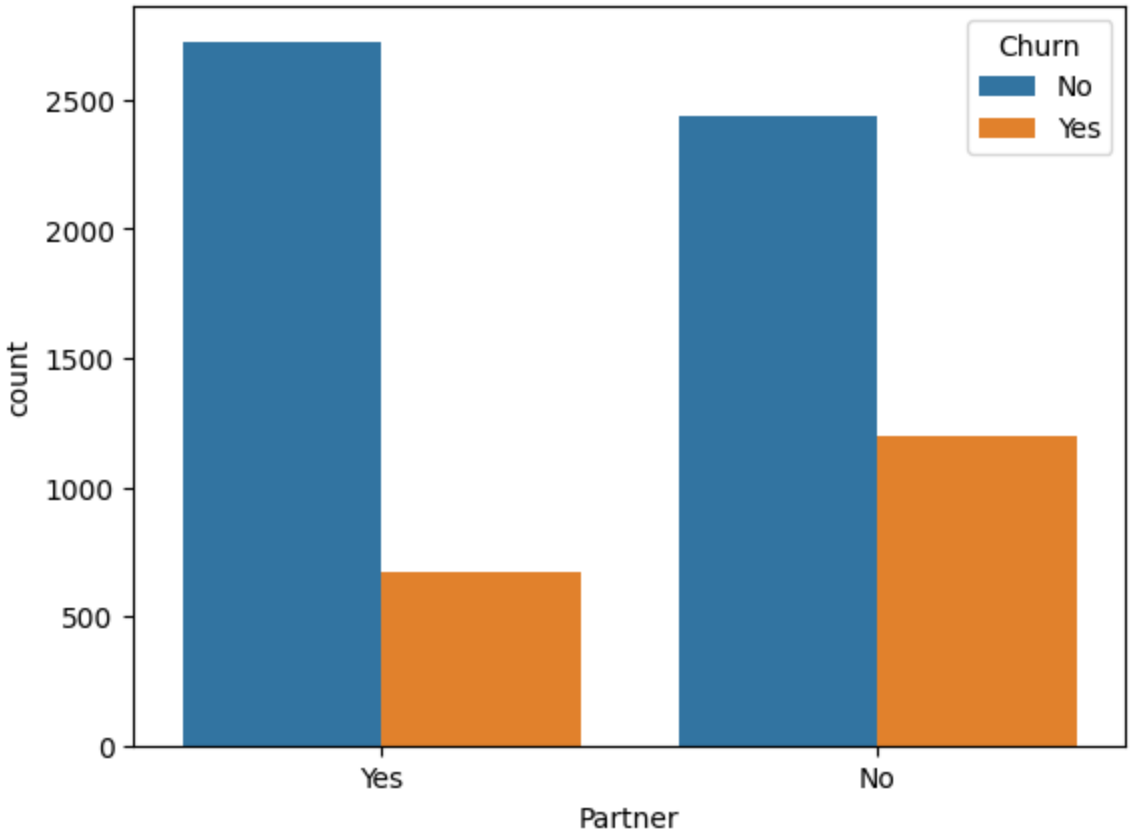
Out[36]:

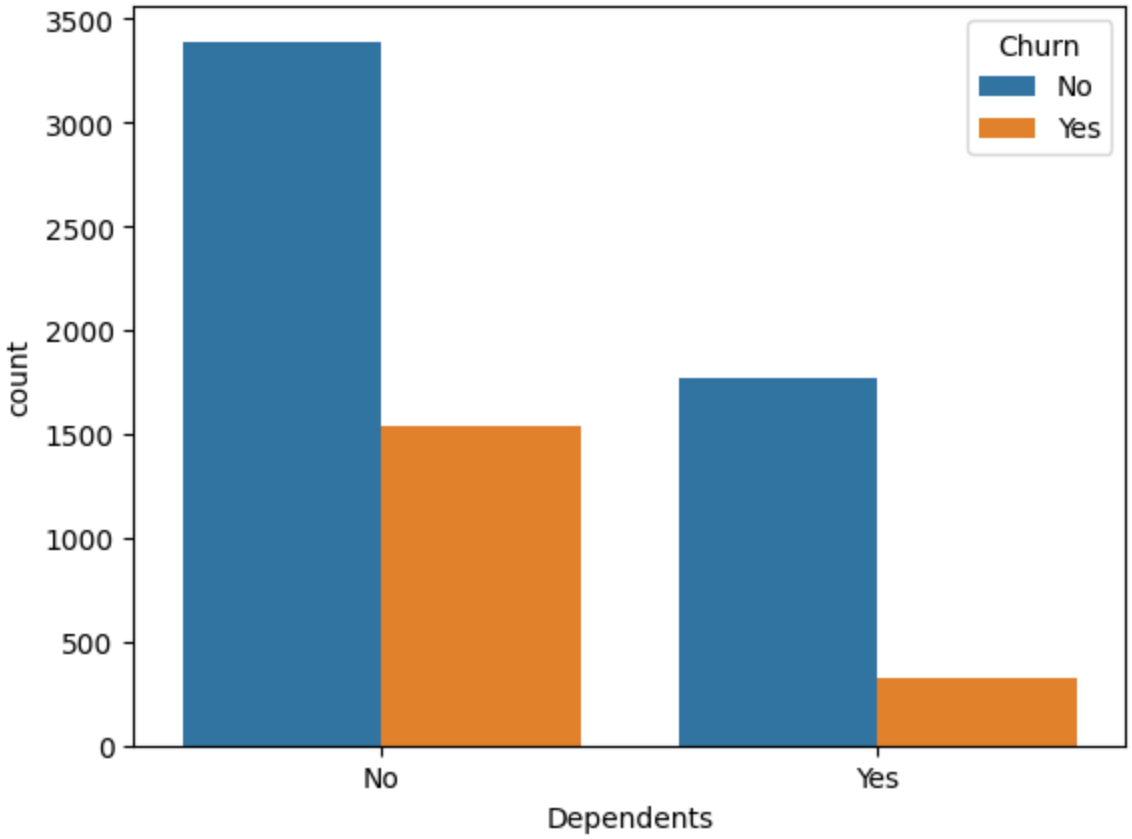
	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	Onli
0	Female	0	Yes	No	No	No phone service	DSL	No	
1	Male	0	No	No	Yes	No	DSL	Yes	
2	Male	0	No	No	Yes	No	DSL	Yes	
3	Male	0	No	No	No	No phone service	DSL	Yes	
4	Female	0	No	No	Yes	No	Fiber optic	No	

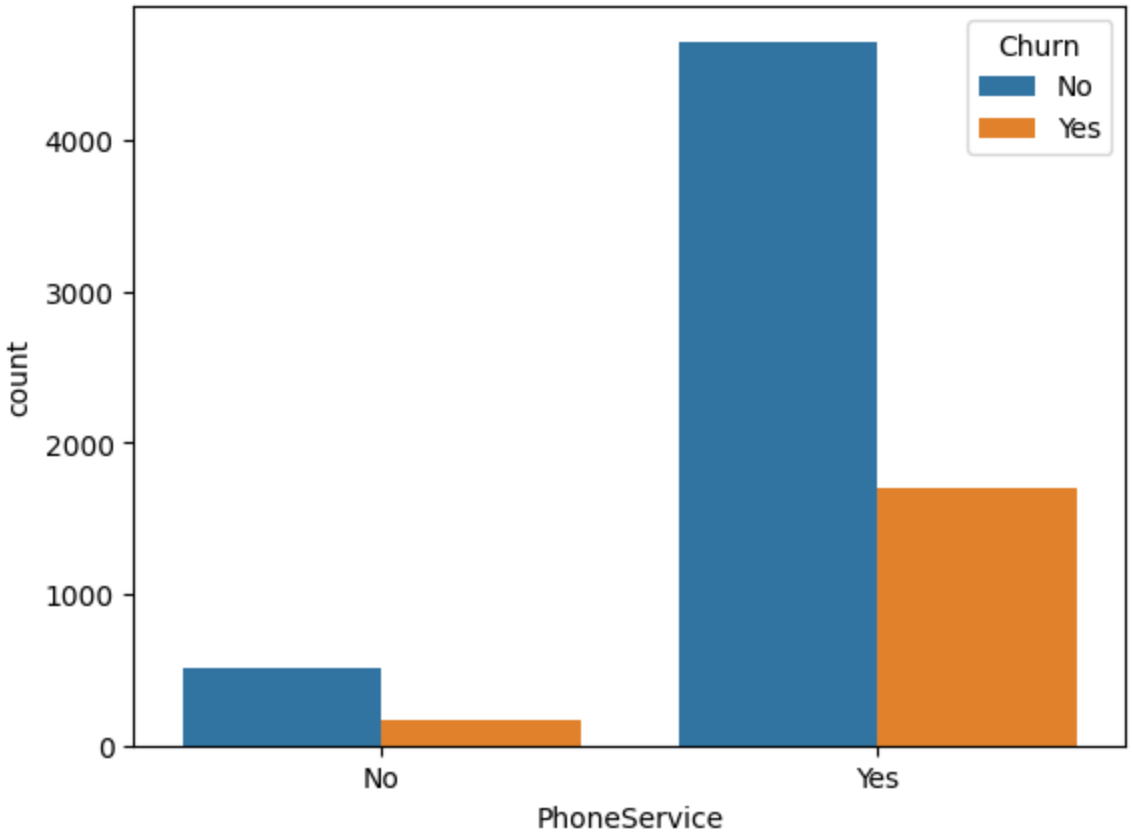
In [37]:
```
for i, predictor in enumerate(telco_data.drop(columns=['Churn', 'TotalCharges', 'MonthlyCharges'
    plt.figure(i)
    sns.countplot(data=telco_data, x=predictor, hue='Churn')
```

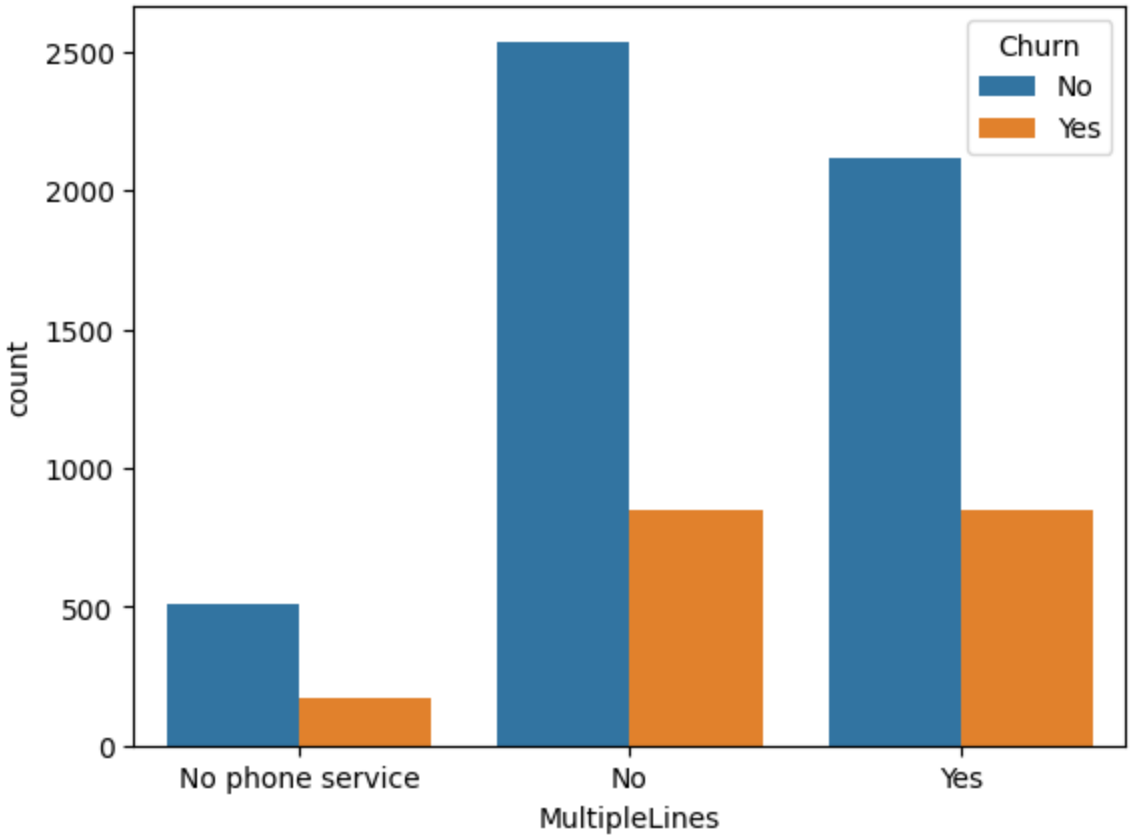


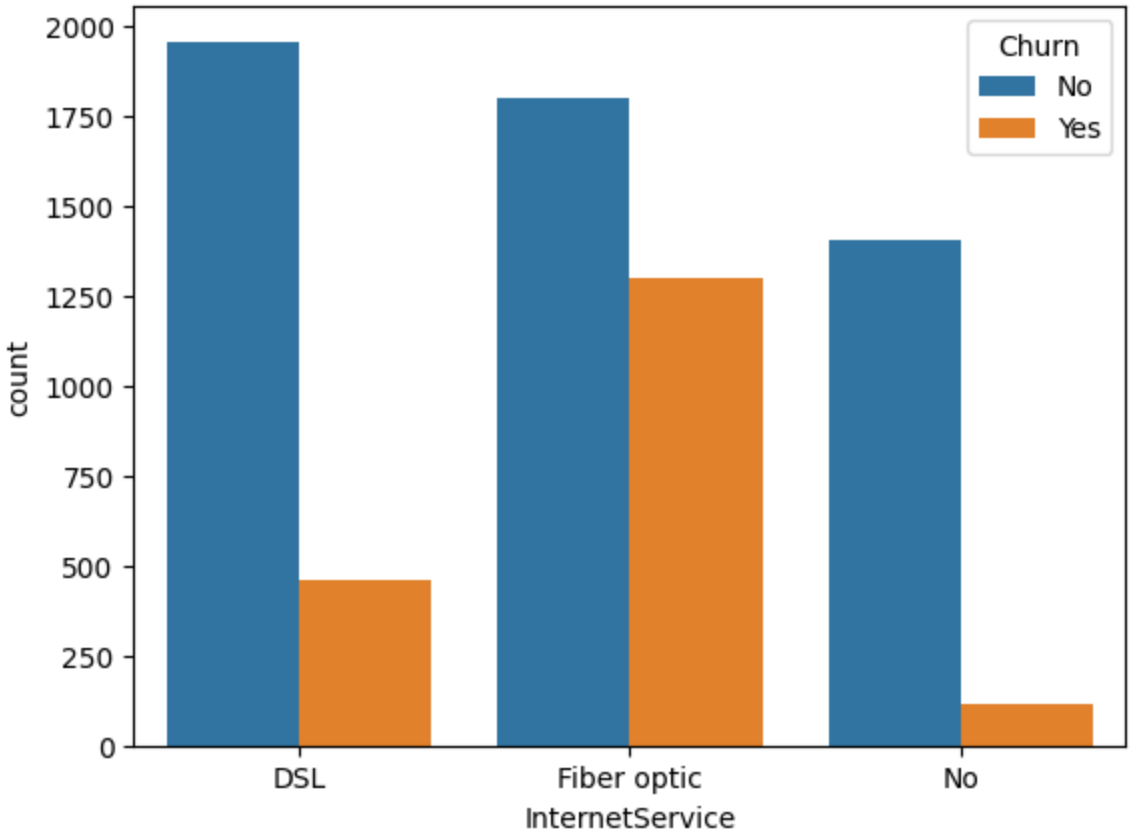

Churn
No
Yes
4000
3000
2000
1000
0
count
0
1
SeniorCitizen

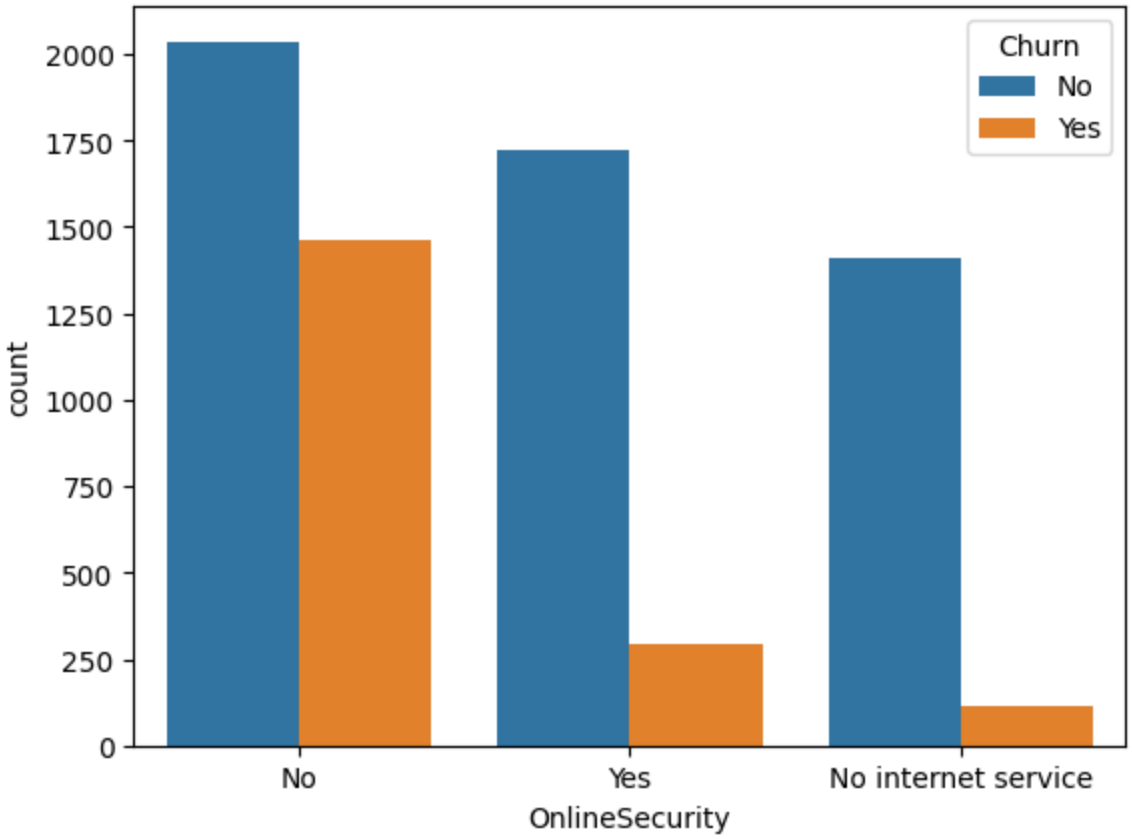

Churn
No
Yes
2500
2000
1500
1000
500
0
count
Yes
No
Partner

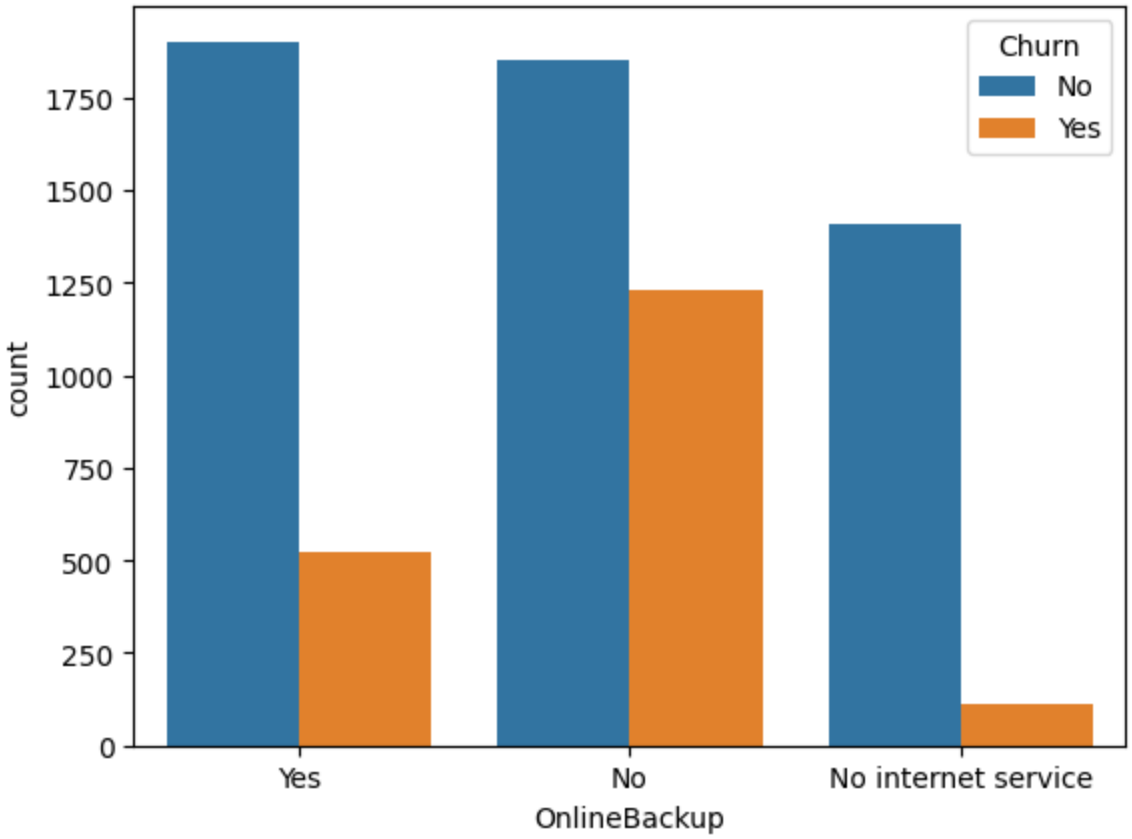

Churn
No
Yes
3500
3000
2500
2000
1500
1000
500
0
count
No
Yes
Dependents

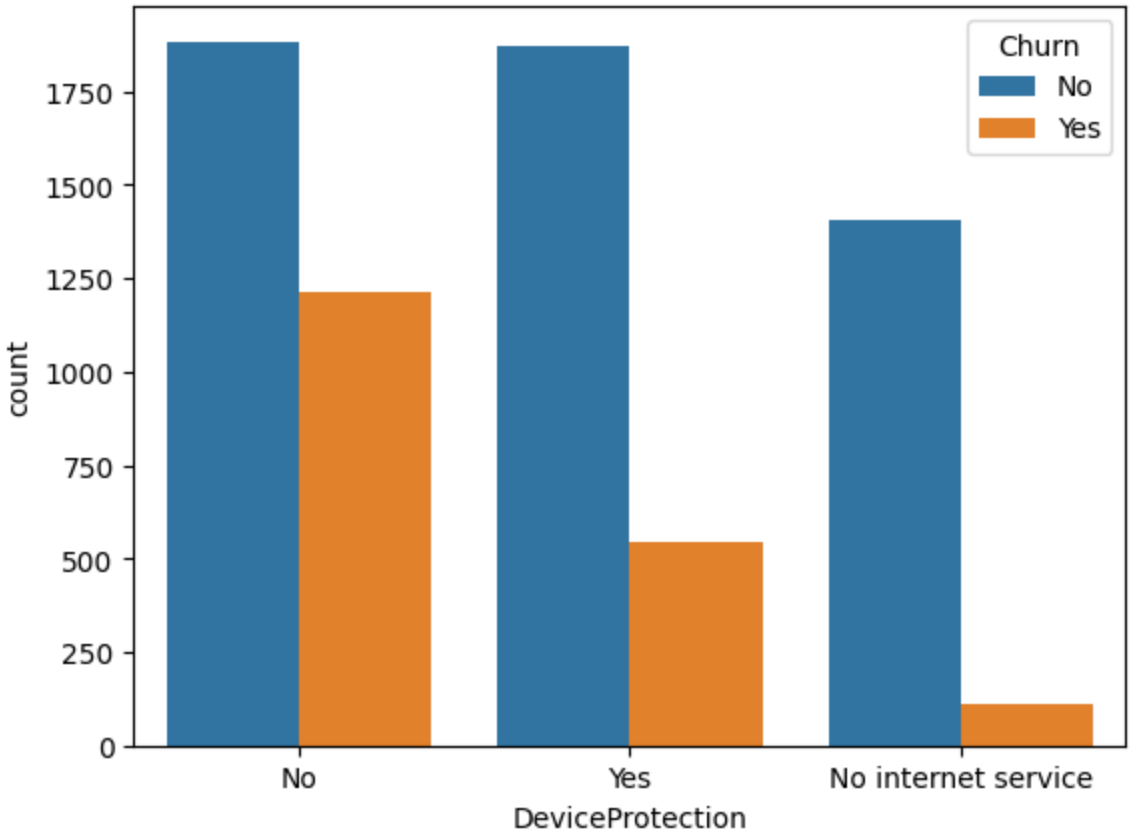

Churn
No
Yes
4000
3000
2000
1000
0
count
No
Yes
PhoneService

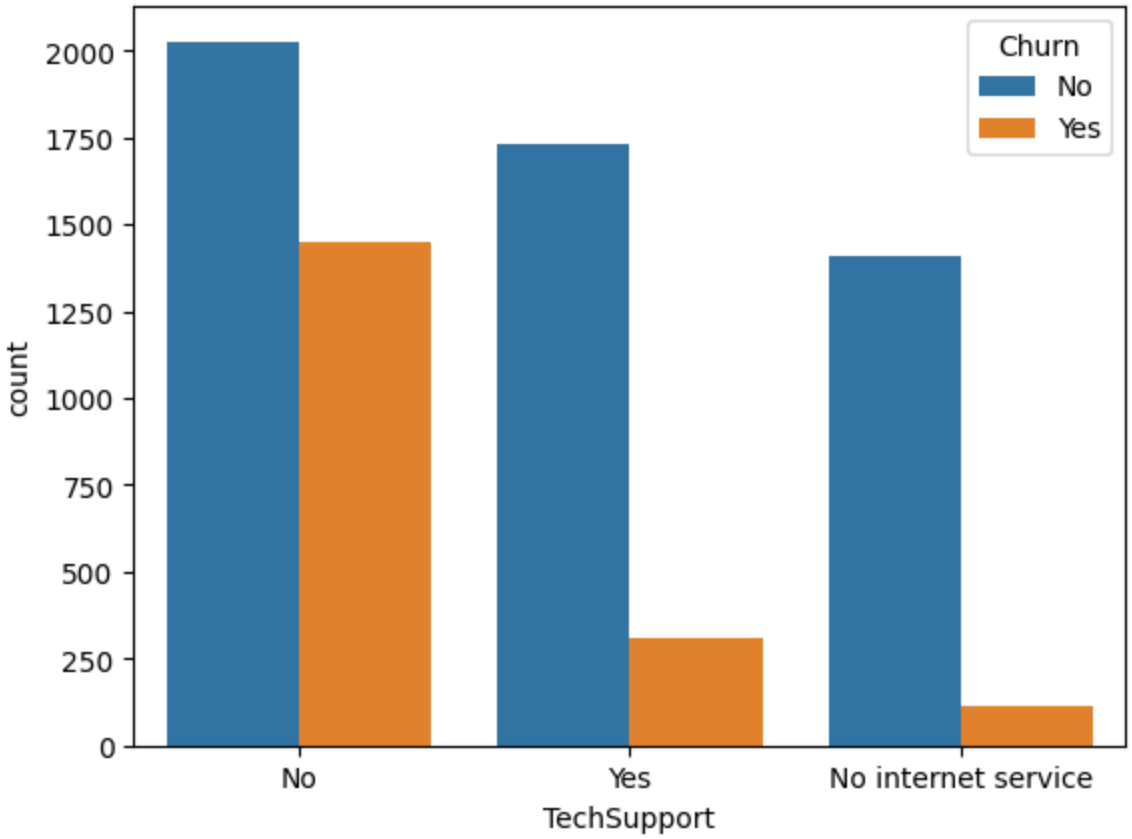

Churn
No
Yes
2500
2000
1500
1000
500
0
count
No phone service
No
Yes
MultipleLines

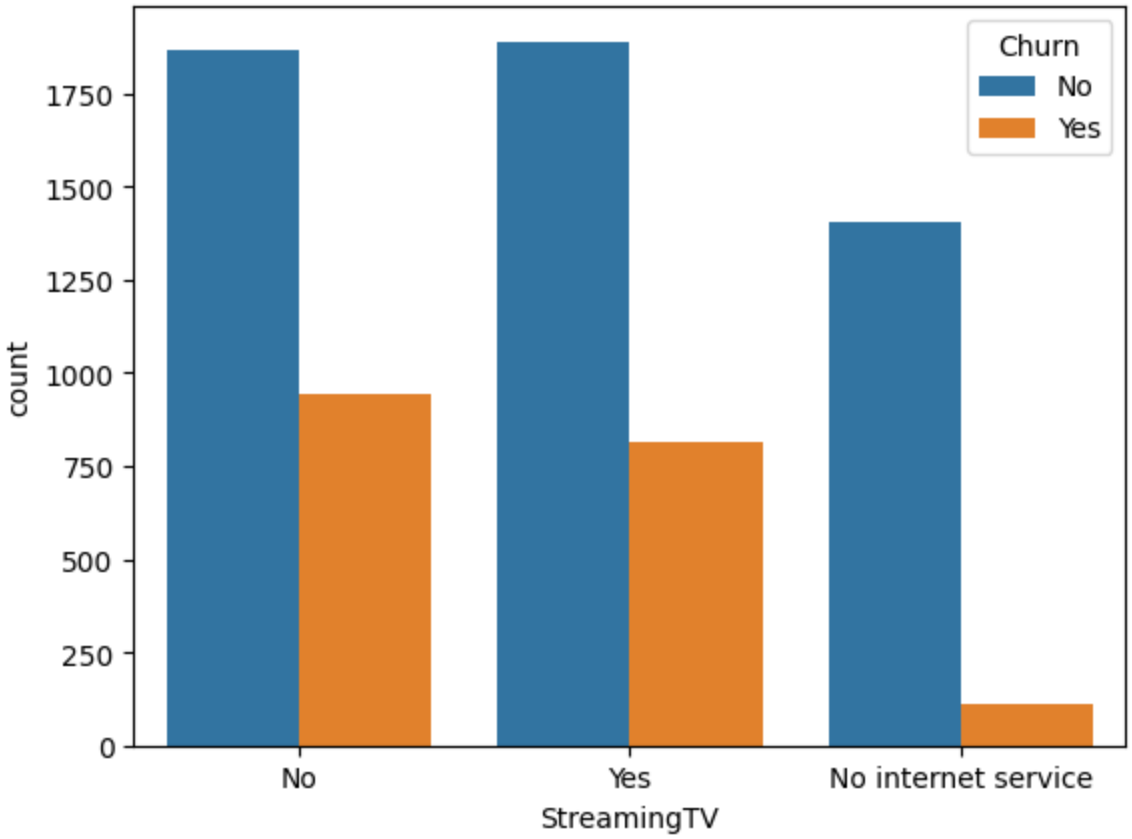

Churn
No
Yes
2000
1750
1500
1250
1000
750
500
250
0
count
DSL
Fiber optic
No
InternetService

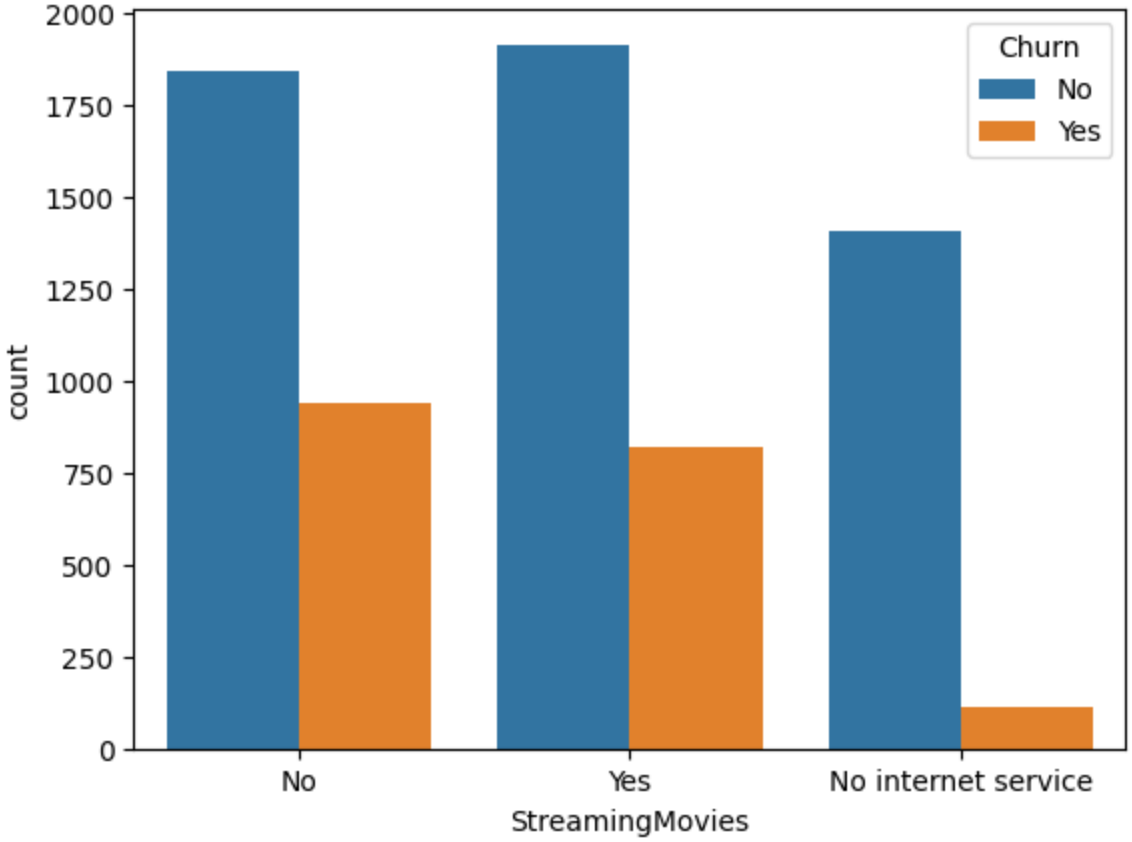

Churn
No
Yes
2000
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
OnlineSecurity

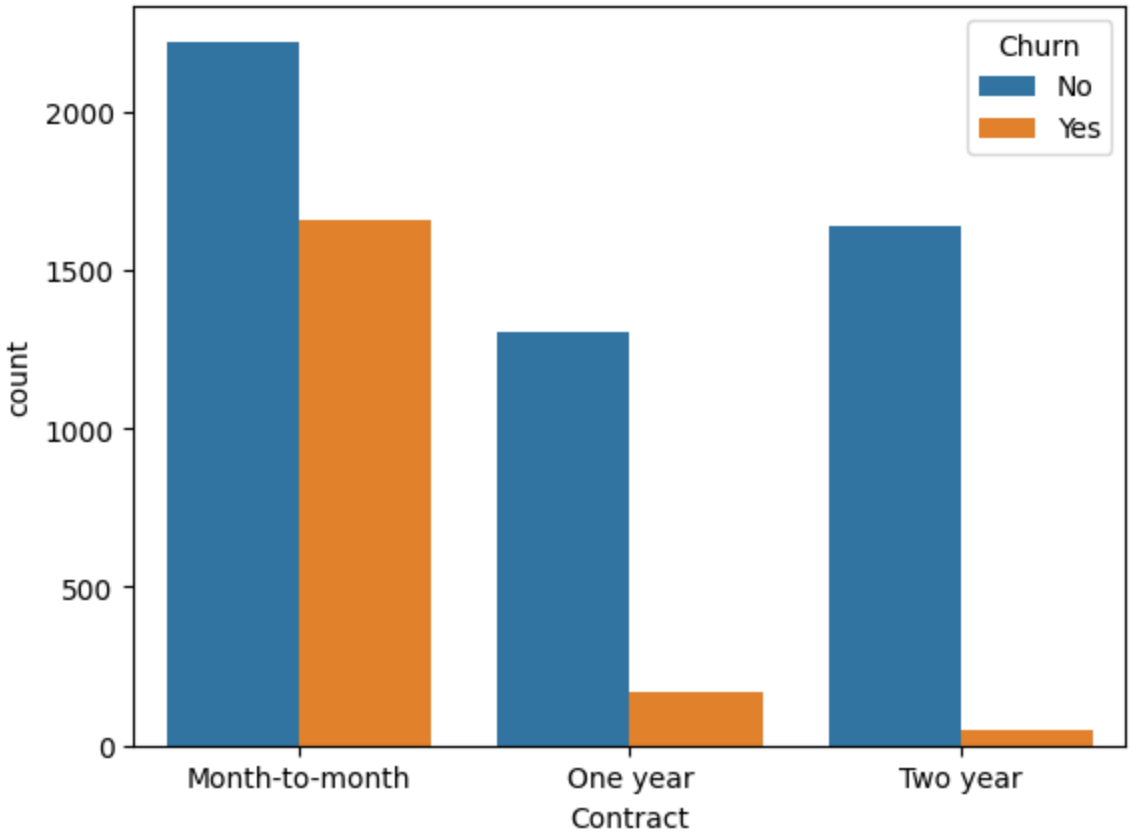

Churn
No
Yes
1750
1500
1250
1000
750
500
250
0
count
Yes
No
No internet service
OnlineBackup

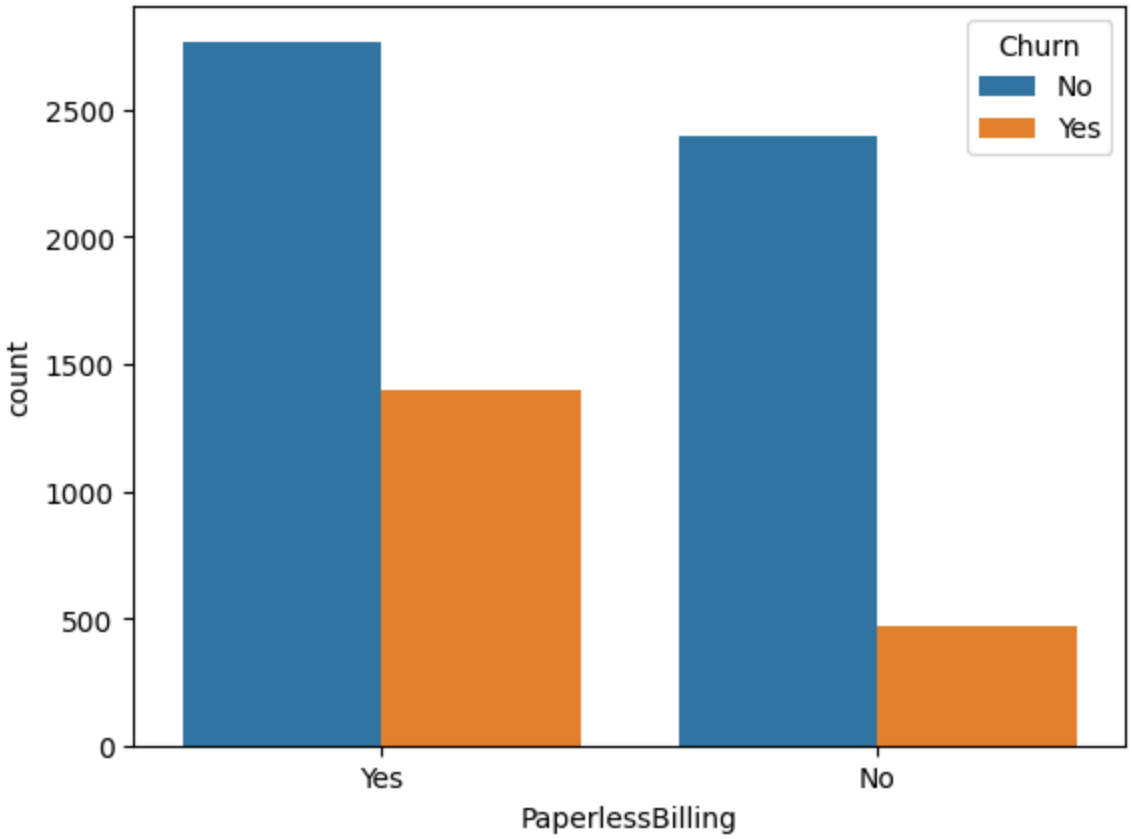

Churn
No
Yes
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
DeviceProtection

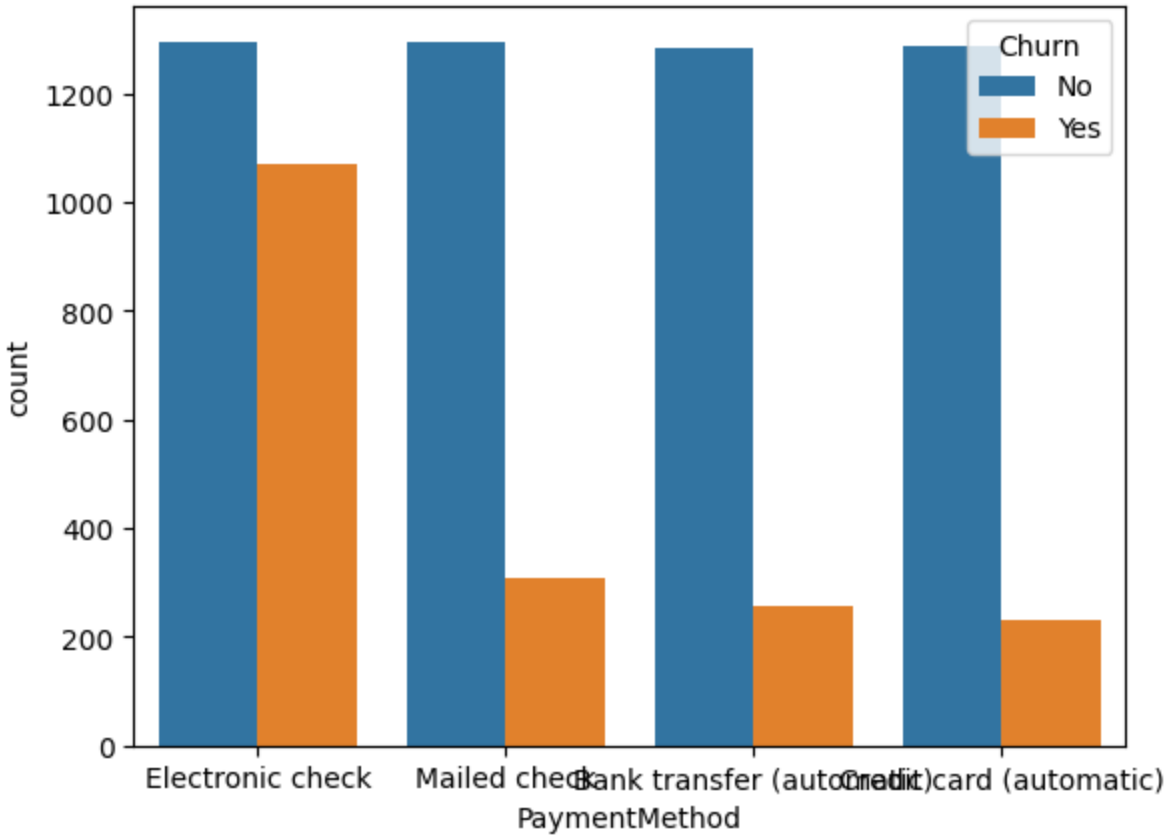

Churn
No
Yes
2000
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
TechSupport

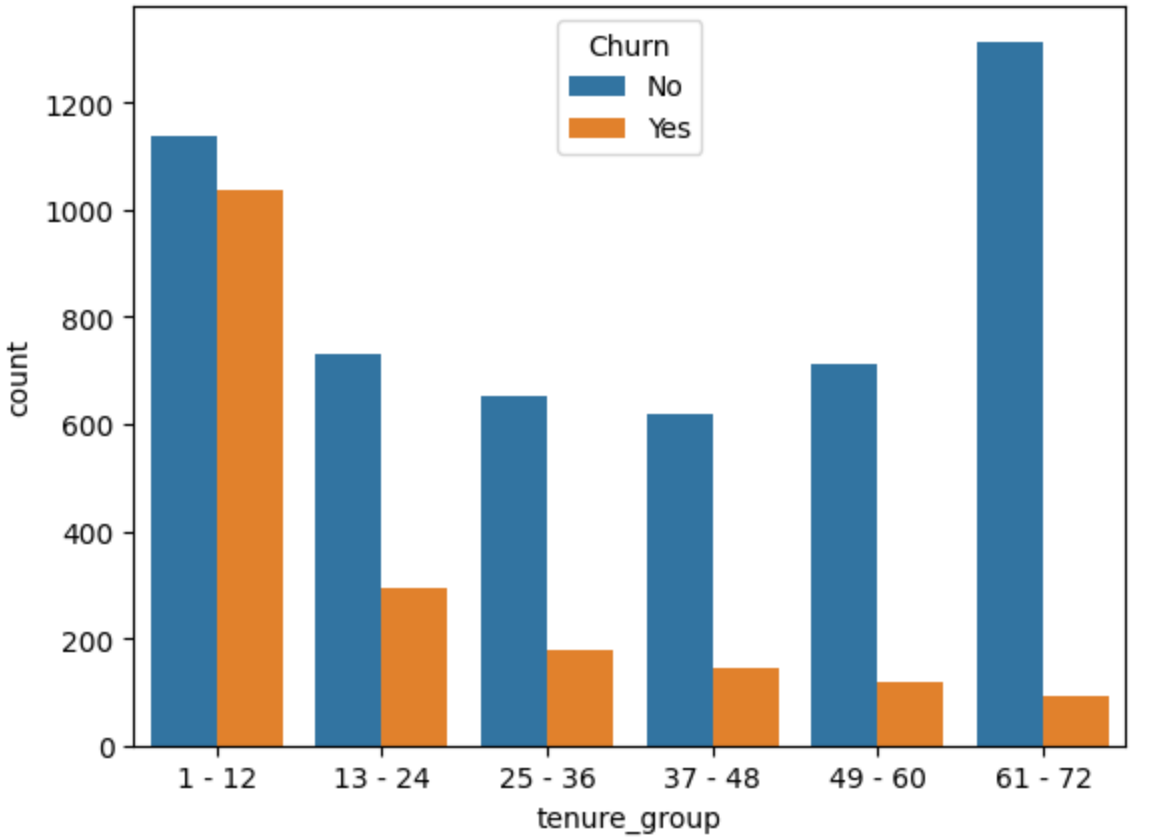

Churn
No
Yes
count
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
StreamingTV


Churn
No
Yes
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
StreamingMovies


Churn
No
Yes
2000
1500
1000
500
0
count
Month-to-month
One year
Two year
Contract


Churn
No
Yes
2500
2000
1500
1000
500
0
count
Yes
No
PaperlessBilling





In [38]:
```
telco_data['Churn'] = np.where(telco_data.Churn == 'Yes',1,0)
```

In [39]:
```
telco_data.head()
```

Out[39]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	Onli
0	Female	0	Yes	No	No	No phone service	DSL	No	
1	Male	0	No	No	Yes	No	DSL	Yes	
2	Male	0	No	No	Yes	No	DSL	Yes	
3	Male	0	No	No	No	No phone service	DSL	Yes	
4	Female	0	No	No	Yes	No	Fiber optic	No	

In [40]:
```
telco_data_dummies = pd.get_dummies(telco_data)
telco_data_dummies.head()
```

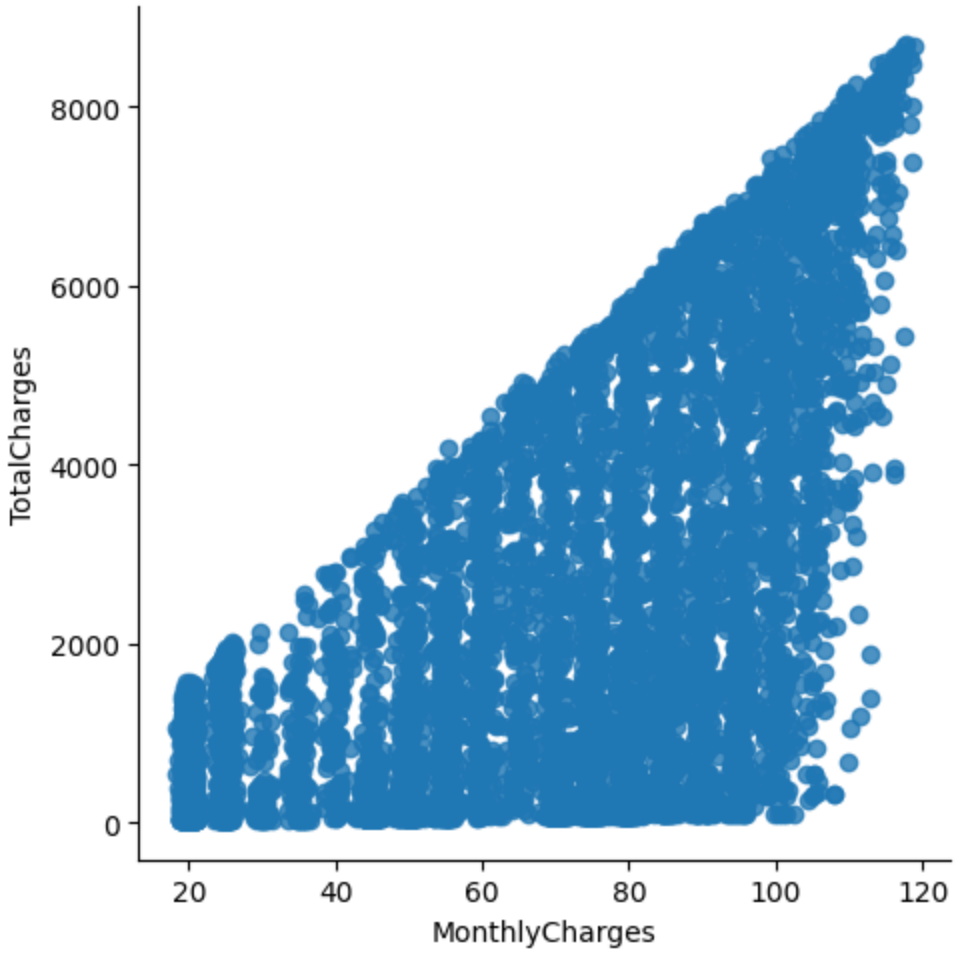
Out[40]:

	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	D
0	0	29.85	29.85	0	True	False	False	True	
1	0	56.95	1889.50	0	False	True	True	False	
2	0	53.85	108.15	1	False	True	True	False	
3	0	42.30	1840.75	0	False	True	True	False	
4	0	70.70	151.65	1	True	False	True	False	

5 rows × 51 columns

In [41]:
```
sns.lmplot(data=telco_data_dummies, x='MonthlyCharges', y='TotalCharges', fit_reg=False)
```

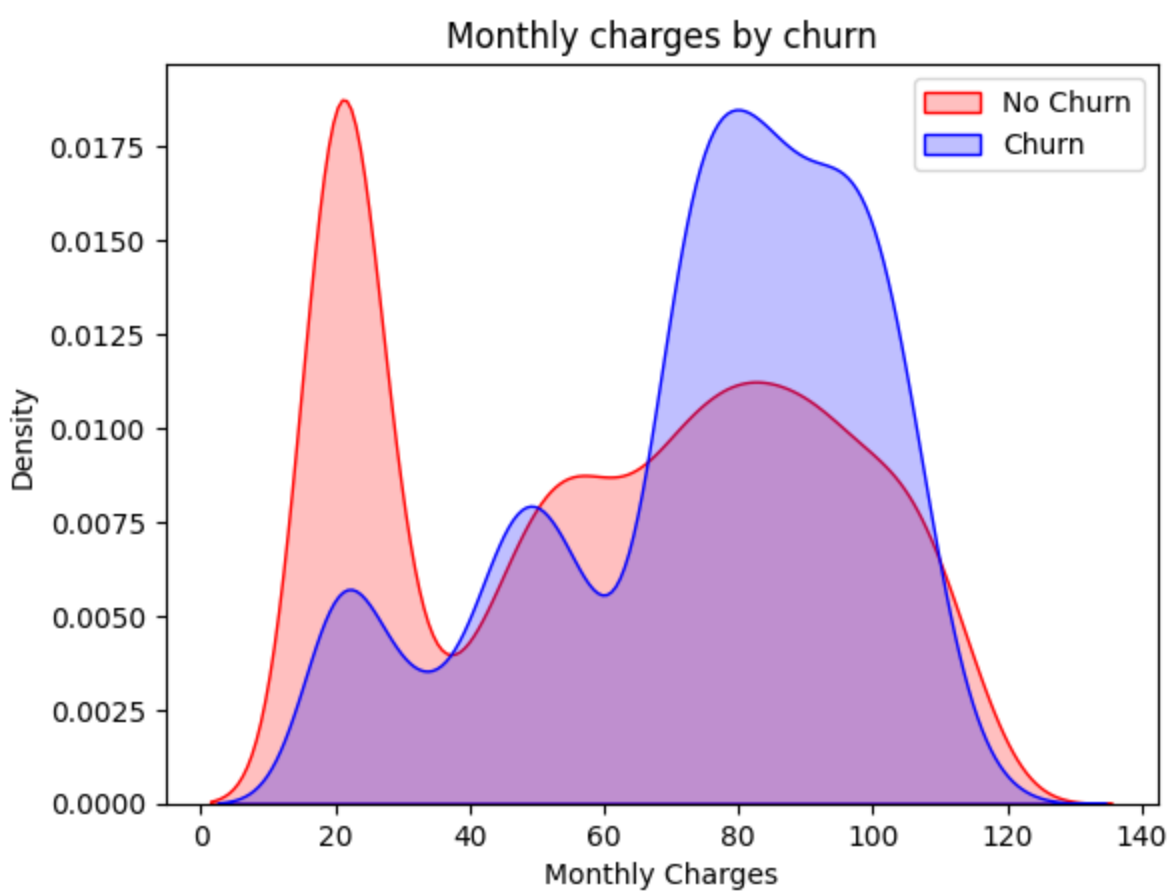
Out[41]:
```
<seaborn.axisgrid.FacetGrid at 0x1df6c914550>
```



In [42]:
```
Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 0) ],
                color="Red", shade = True)
Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 1) ],
                ax =Mth, color="Blue", shade= True)
Mth.legend(["No Churn","Churn"],loc='upper right')
Mth.set_ylabel('Density')
Mth.set_xlabel('Monthly Charges')
Mth.set_title('Monthly charges by churn')
```

```
C:\Users\mohda\AppData\Local\Temp\ipykernel_24292\722082952.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 0) ],
C:\Users\mohda\AppData\Local\Temp\ipykernel_24292\722082952.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  Mth = sns.kdeplot(telco_data_dummies.MonthlyCharges[(telco_data_dummies["Churn"] == 1) ],
```

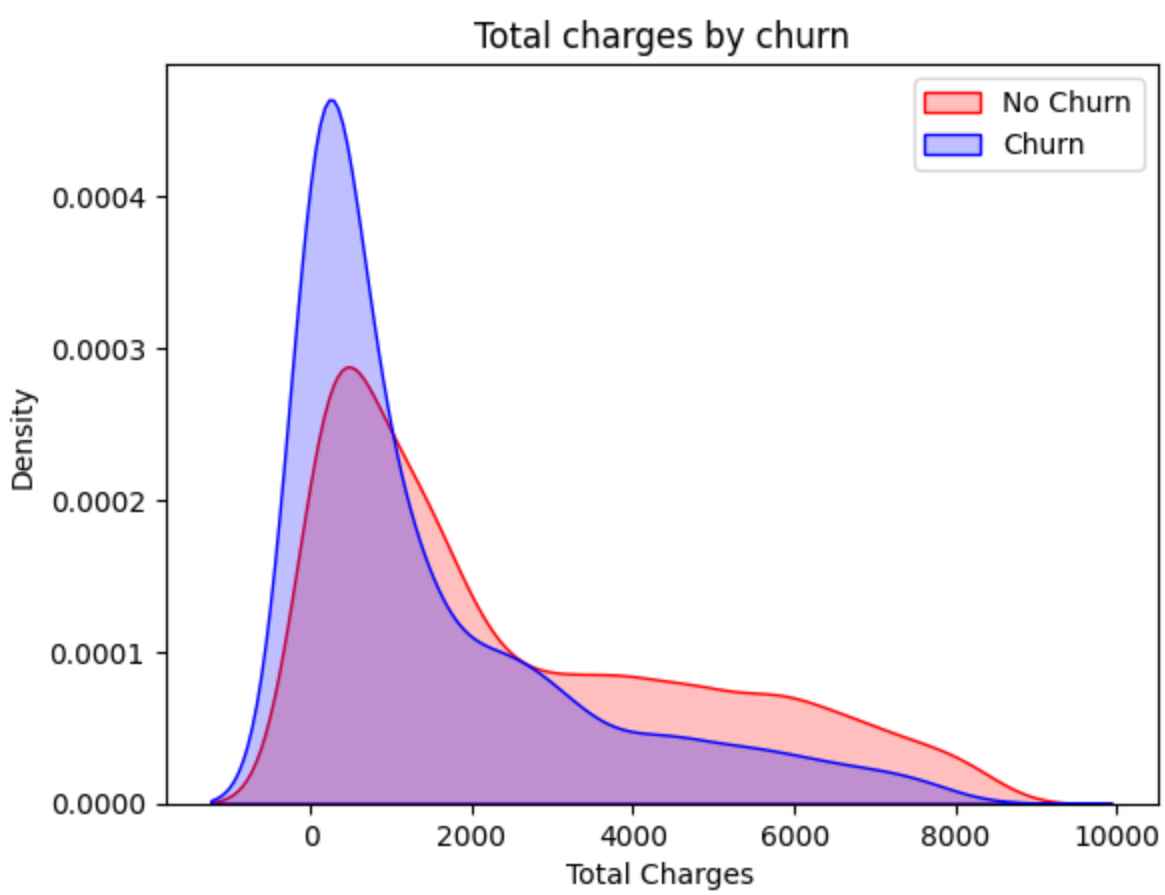
Out[42]: Text(0.5, 1.0, 'Monthly charges by churn')



In [43]:
```
Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 0) ],
                  color="Red", shade = True)
Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 1) ],
                  ax =Tot, color="Blue", shade= True)
Tot.legend(["No Churn","Churn"],loc='upper right')
Tot.set_ylabel('Density')
Tot.set_xlabel('Total Charges')
Tot.set_title('Total charges by churn')
```

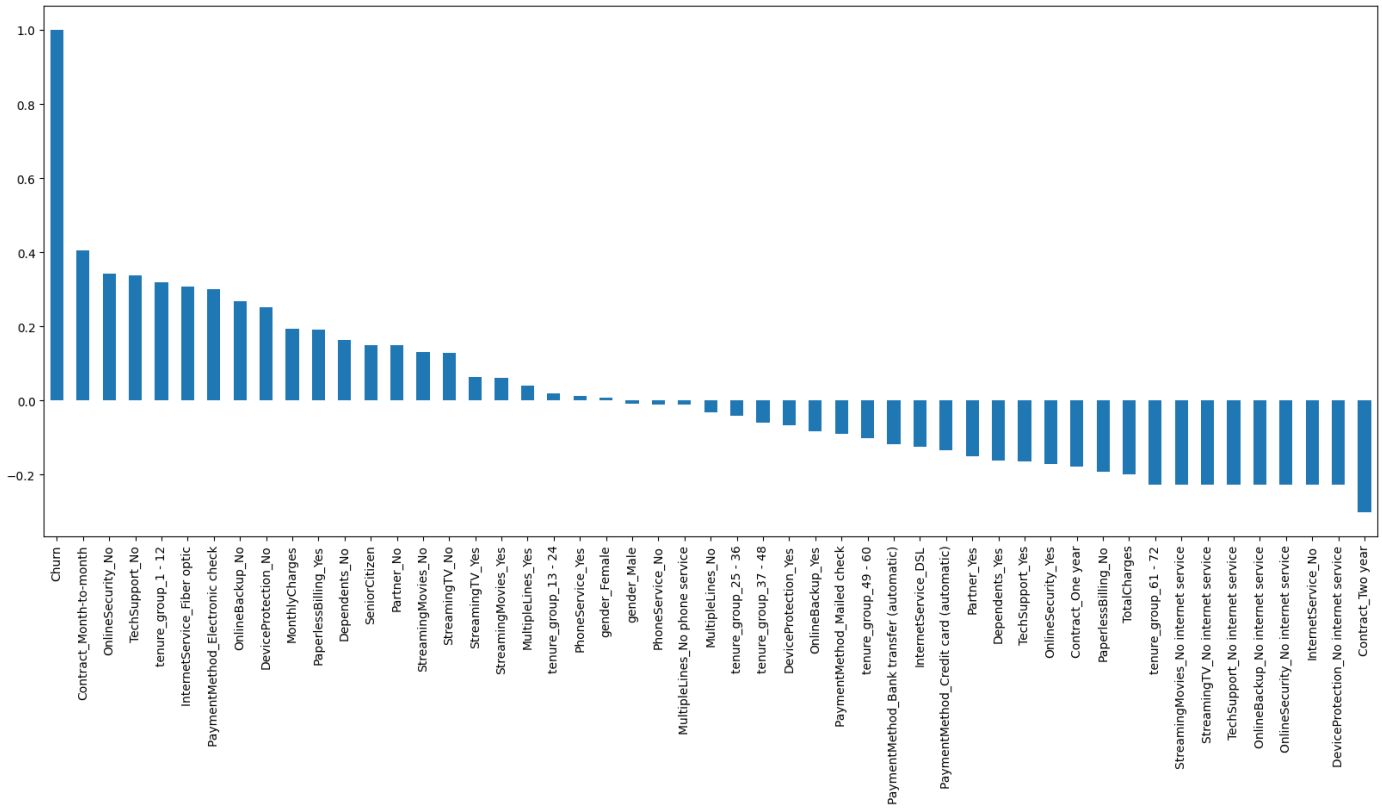
```
C:\Users\mohda\AppData\Local\Temp\ipykernel_24292\4019118049.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 0) ],
C:\Users\mohda\AppData\Local\Temp\ipykernel_24292\4019118049.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  Tot = sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Churn"] == 1) ],
```

Out[43]:
```
Text(0.5, 1.0, 'Total charges by churn')
```



In [44]:
```
plt.figure(figsize=(20,8))
telco_data_dummies.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
```

Out[44]: <Axes: >



In [45]:
```
plt.figure(figsize=(12,12))
sns.heatmap(telco_data_dummies.corr(), cmap="Paired")
```

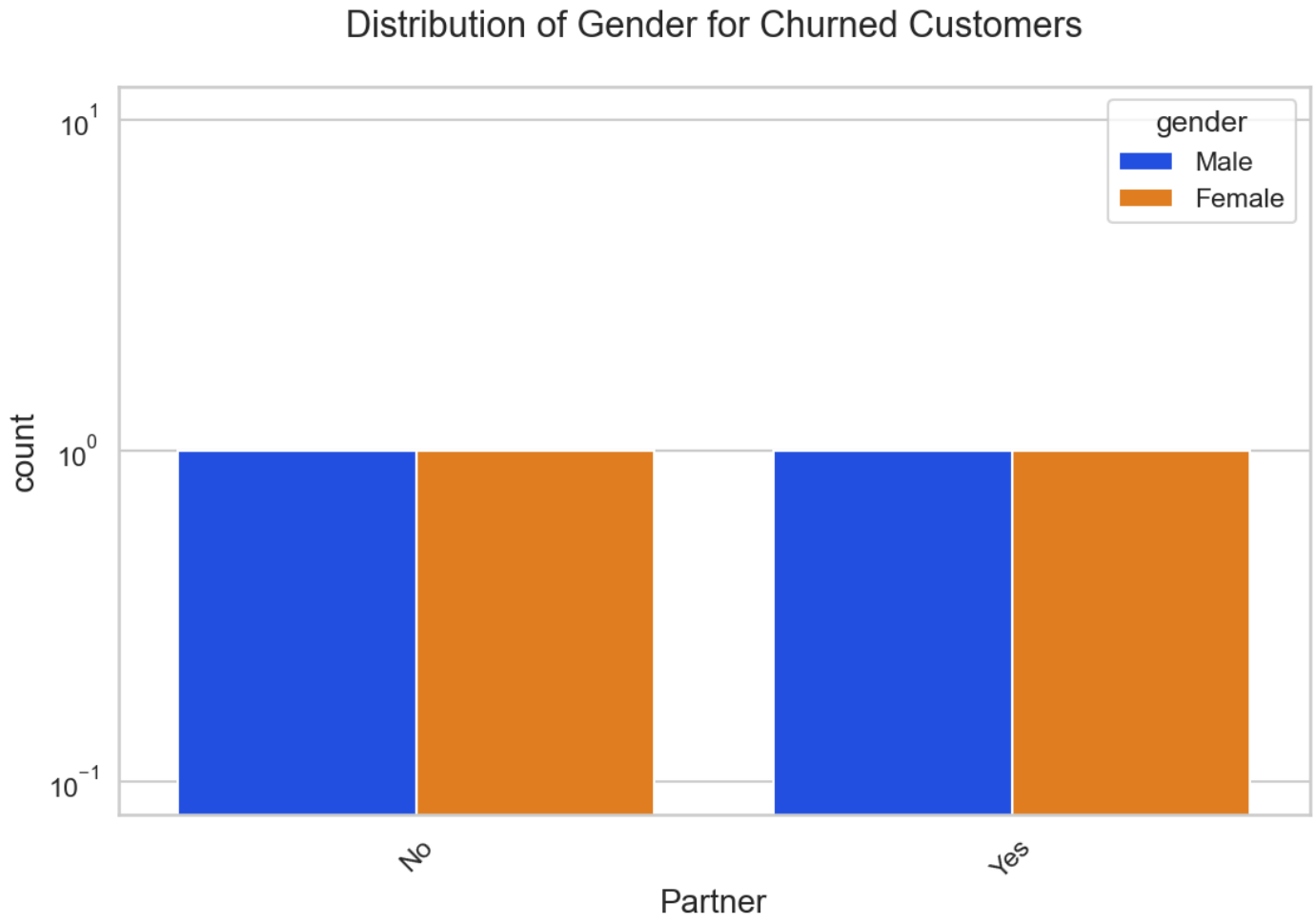
Out[45]: <Axes: >

In [46]:
```
new_df1_target0=telco_data.loc[telco_data["Churn"]==0]
new_df1_target1=telco_data.loc[telco_data["Churn"]==1]
```

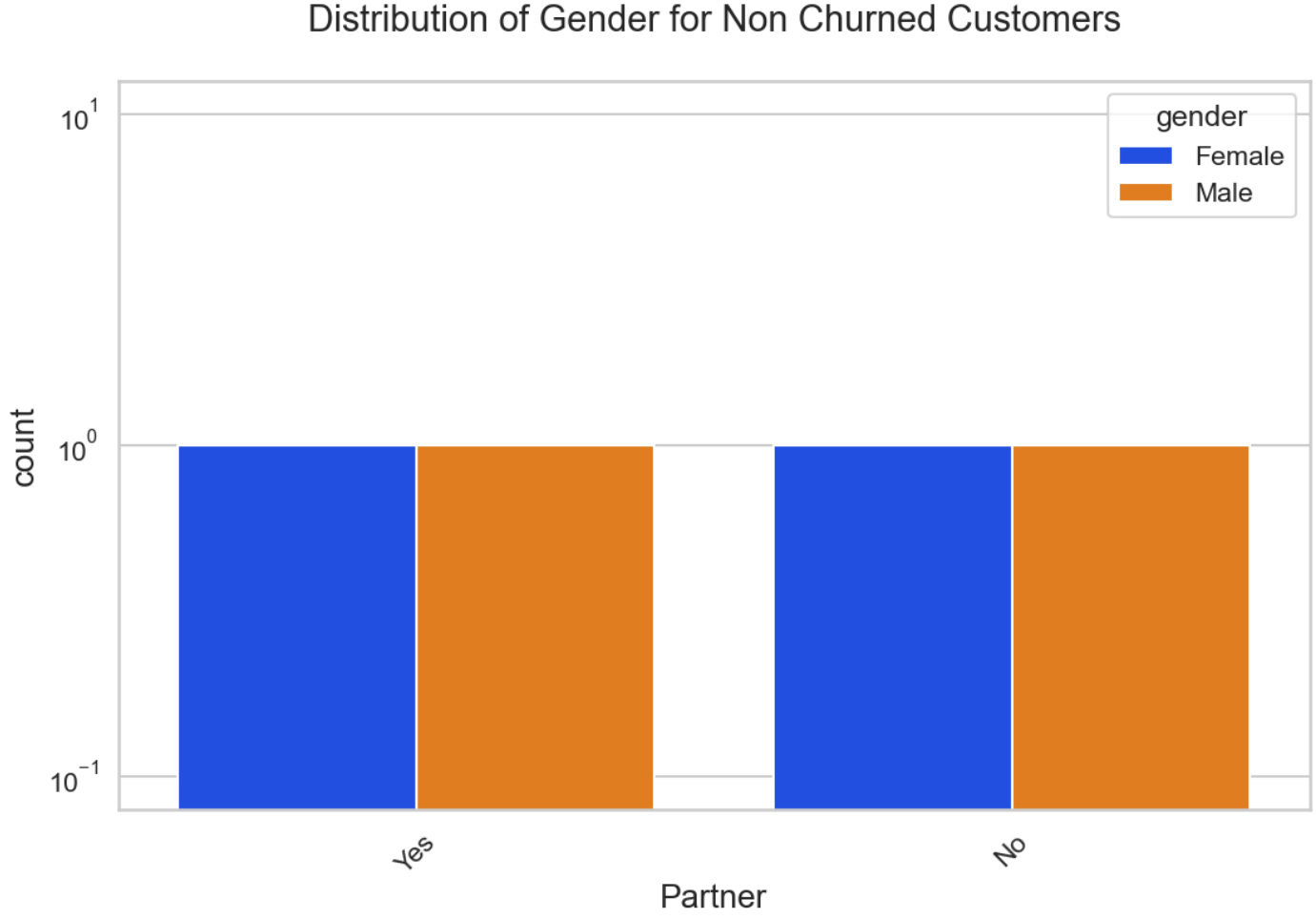
In [47]:
```
def uniplot(df,col,title,hue =None):

    sns.set_style('whitegrid')
    sns.set_context('talk')
    plt.rcParams["axes.labelsize"] = 20
    plt.rcParams['axes.titlesize'] = 22
    plt.rcParams['axes.titlepad'] = 30


    temp = pd.Series(data = hue)
    fig, ax = plt.subplots()
    width = len(df[col].unique()) + 7 + 4*len(temp.unique())
    fig.set_size_inches(width , 8)
    plt.xticks(rotation=45)
    plt.yscale('log')
```

```
    plt.title(title)
    ax = sns.countplot(data = df, x= col, order=df[col].value_counts().index,hue = hue,palette='
    plt.show()
```

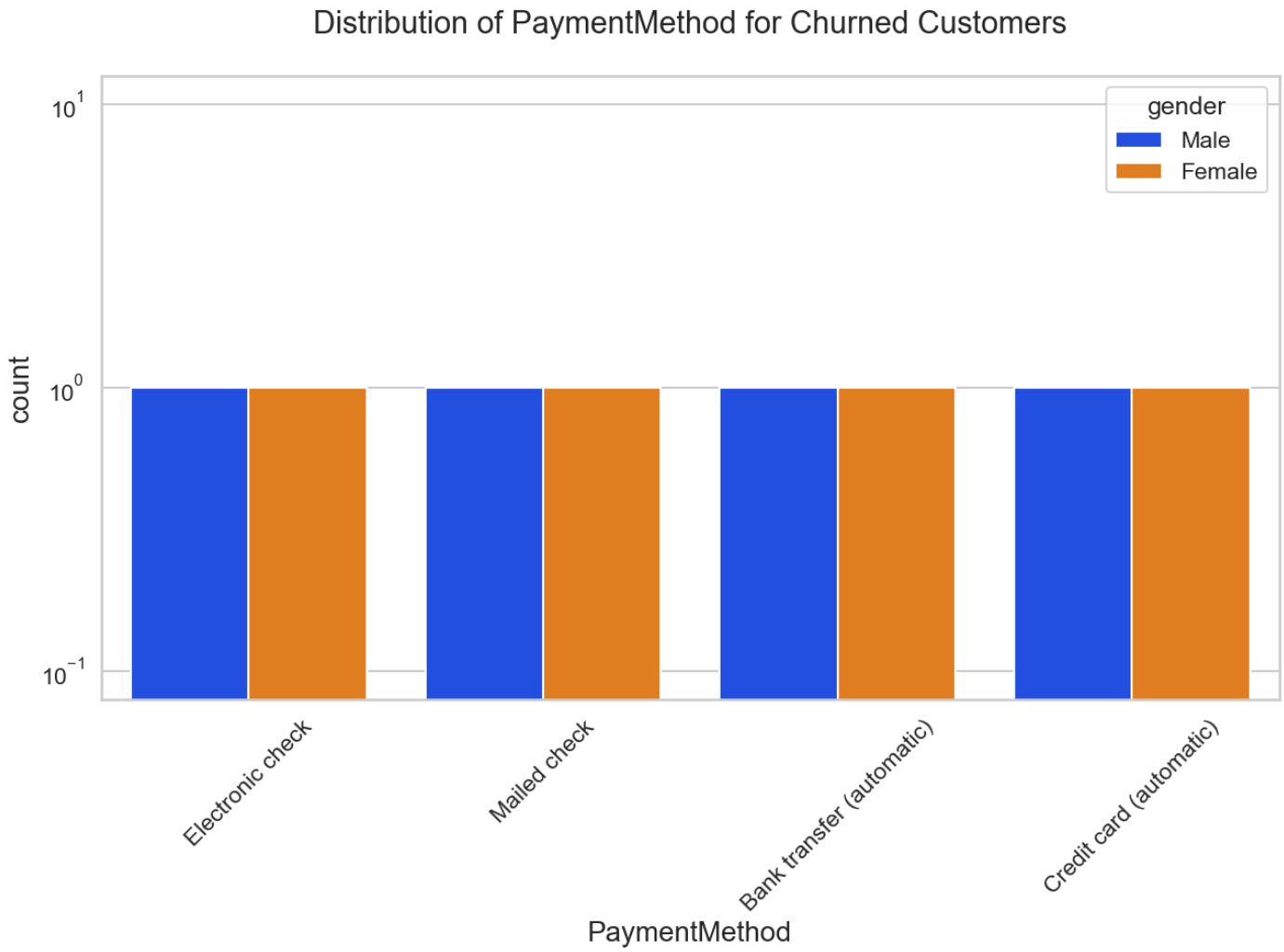
In [48]:
```
uniplot(new_df1_target1,col='Partner',title='Distribution of Gender for Churned Customers',hue='
```



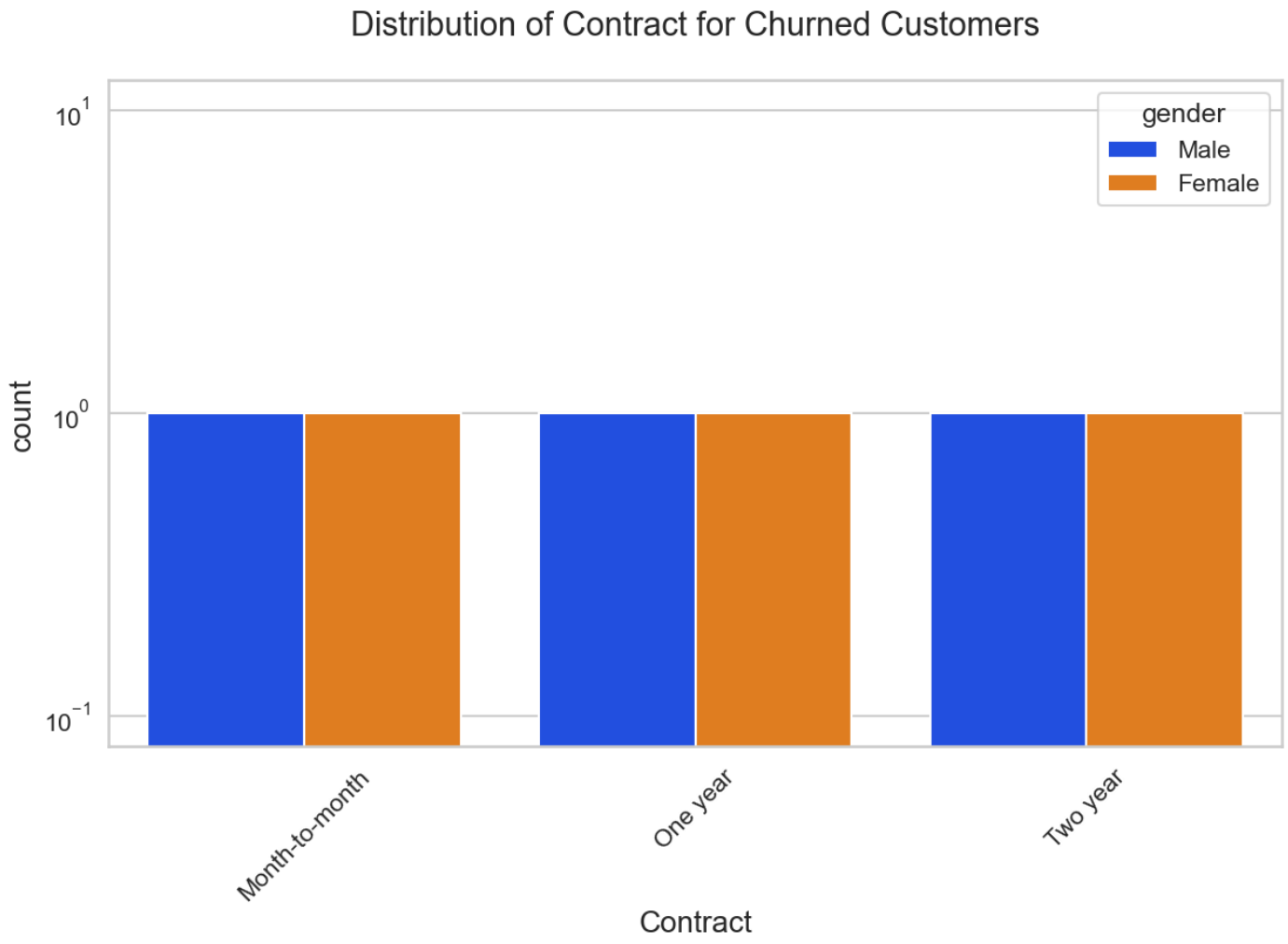
In [49]:
```
uniplot(new_df1_target0,col='Partner',title='Distribution of Gender for Non Churned Customers',hu
```



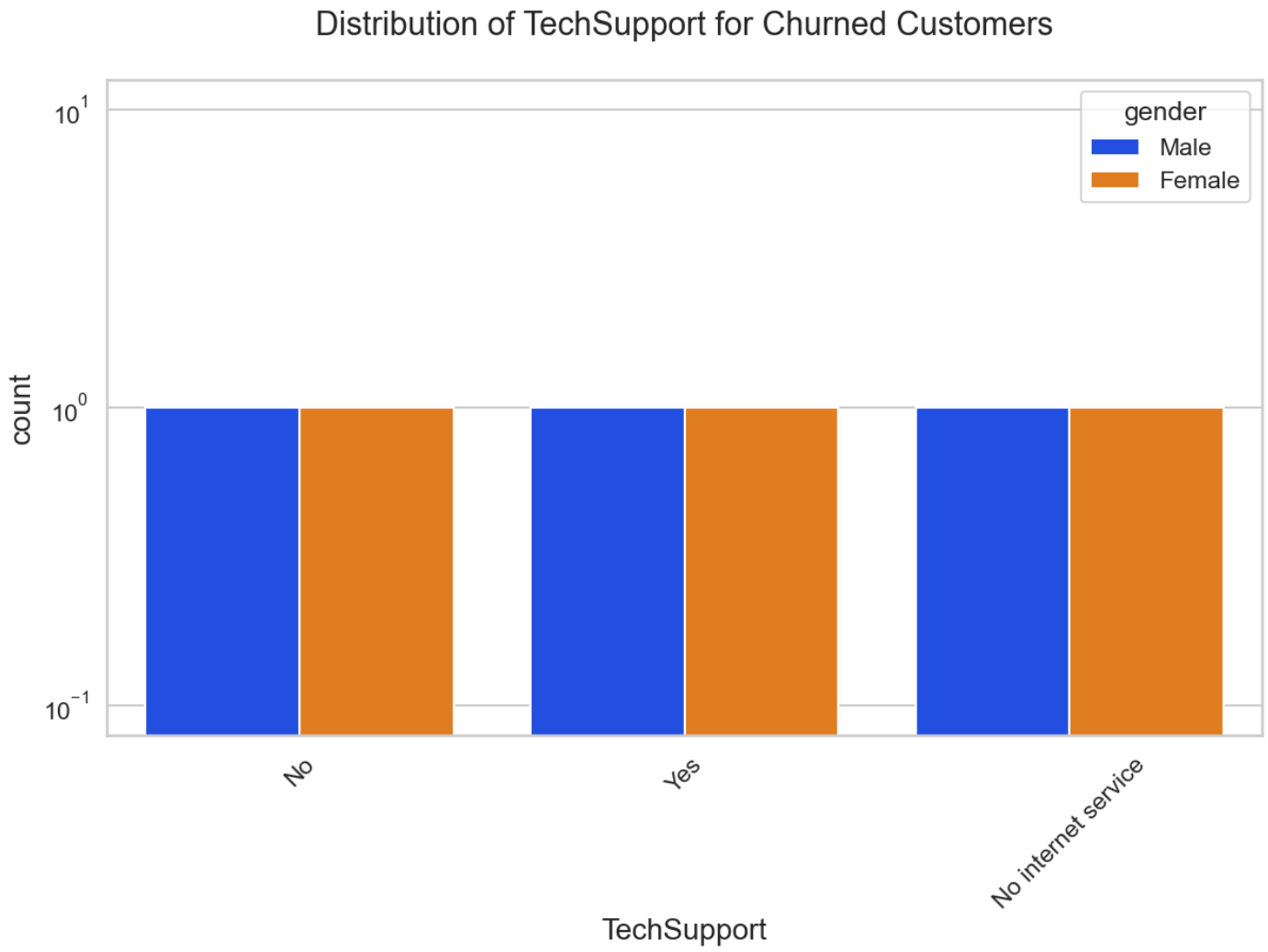
In [50]:
```
uniplot(new_df1_target1,col='PaymentMethod',title='Distribution of PaymentMethod for Churned Cust
```



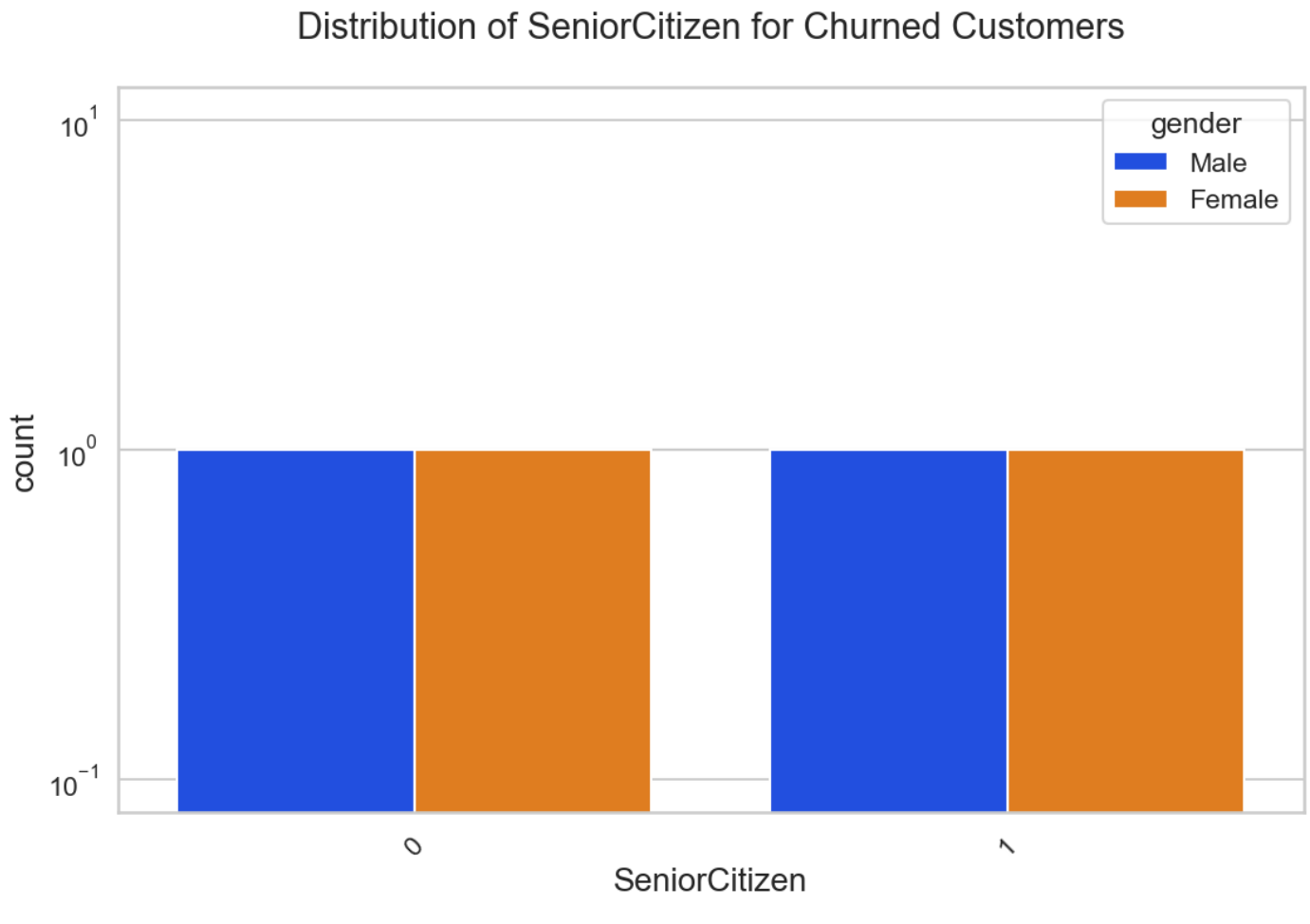
In [51]:
```
uniplot(new_df1_target1,col='Contract',title='Distribution of Contract for Churned Customers',hu
```



In [52]:
```
uniplot(new_df1_target1,col='TechSupport',title='Distribution of TechSupport for Churned Customer
```



In [53]:
```
uniplot(new_df1_target1,col='SeniorCitizen',title='Distribution of SeniorCitizen for Churned Cust
```



In [56]:
```
telco_data_dummies.to_csv('tel_churn.csv')
```

In [ ]:
```

```