eda

November 5, 2021

1 Exploratory data analysis

In this notebook I will explore how do the data look like. I will calculate some basis statistics and visualise the dataset.

1.0.1 Load libraries

[1]:

```
import pandas as pd
import numpy as np
import srs

from dataprep import eda
```

[2]:

```
import seaborn as sns

sns.set()
```

[3]:

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

1.0.2 Load data

[4]:

```
data = pd.read_csv('../data/WA_Fn-UseC_-Telco-Customer-Churn.csv',␣
 ↪index_col='customerID')

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7043 entries, 7590-VHVEG to 3186-AJIEK
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7043 non-null   object
 1   SeniorCitizen     7043 non-null   int64
 2   Partner           7043 non-null   object
 3   Dependents        7043 non-null   object
 4   tenure            7043 non-null   int64
 5   PhoneService      7043 non-null   object
 6   MultipleLines     7043 non-null   object
 7   InternetService   7043 non-null   object
 8   OnlineSecurity    7043 non-null   object
 9   OnlineBackup      7043 non-null   object
 10  DeviceProtection  7043 non-null   object
 11  TechSupport       7043 non-null   object
 12  StreamingTV       7043 non-null   object
 13  StreamingMovies   7043 non-null   object
 14  Contract          7043 non-null   object
 15  PaperlessBilling  7043 non-null   object
 16  PaymentMethod     7043 non-null   object
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   object
 19  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```

Description of the features: * Gender: The customer's gender: Male, Female * Senior Citizen: Indicates if the customer is 65 or older: Yes, No * Partner: Indicates if the customer is a partner: Yes, No * Dependents: Indicates if the customer lives with any dependents: Yes, No. Dependents could be children, parents, grandparents, etc. * Tenure: How long they've been a customer (in months) * Phone Service: Indicates if the customer subscribes to home phone service with the company: Yes, No * Multiple Lines: Indicates if the customer subscribes to multiple telephone lines with the company: Yes, No * Internet Service: Indicates if the customer subscribes to Internet service with the company: No, DSL, Fiber Optic, Cable. * Online Security: Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No * Online Backup: Indicates if the customer subscribes to an additional online backup service provided by the company: Yes, No * Device Protection Plan: Indicates if the customer subscribes to an additional device protection plan for their Internet equipment provided by the company: Yes, No * Tech Support: Indicates if the customer subscribes to an additional technical support plan from the company with reduced wait times: Yes, No * Streaming TV: Indicates if the customer uses their Internet service to stream television programing from a third party provider: Yes, No. The company does not charge an additional fee for this service * Streaming Movies: Indicates if the customer uses their Internet service to stream movies from a third party provider: Yes, No. The company does not charge an additional fee for this service * Contract: Indicates the customer's current contract type: Month-to-Month, One Year, Two Year * Paperless Billing: Indicates if the customer has chosen paperless billing: Yes, No * Payment Method: Indicates how the customer pays their bill: Bank Withdrawal, Credit Card, Mailed Check * Monthly Charge: Indicates the customer's current total monthly charge for all their services from the company * Total Charges: Indicates the customer's total charges * Churn: Indicates if the customer have churned: Yes, No

[5]:
```
data.head()
```

[5]:
```
            gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
customerID
7590-VHVEG  Female              0     Yes         No       1           No
5575-GNVDE    Male              0      No         No      34          Yes
3668-QPYBK    Male              0      No         No       2          Yes
7795-CFOCW    Male              0      No         No      45           No
```

```
9237-HQITU  Female              0      No         No       2          Yes

               MultipleLines InternetService OnlineSecurity OnlineBackup  \
customerID
7590-VHVEG  No phone service             DSL             No          Yes
5575-GNVDE                No             DSL            Yes           No
3668-QPYBK                No             DSL            Yes          Yes
7795-CFOCW  No phone service             DSL            Yes           No
9237-HQITU                No     Fiber optic             No           No

           DeviceProtection TechSupport StreamingTV StreamingMovies  \
customerID
7590-VHVEG               No          No          No              No
5575-GNVDE              Yes          No          No              No
3668-QPYBK               No          No          No              No
7795-CFOCW              Yes         Yes          No              No
9237-HQITU               No          No          No              No

                  Contract PaperlessBilling              PaymentMethod  \
customerID
7590-VHVEG  Month-to-month              Yes           Electronic check
5575-GNVDE        One year               No               Mailed check
3668-QPYBK  Month-to-month              Yes               Mailed check
7795-CFOCW        One year               No  Bank transfer (automatic)
9237-HQITU  Month-to-month              Yes           Electronic check

            MonthlyCharges TotalCharges Churn
customerID
7590-VHVEG           29.85        29.85    No
5575-GNVDE           56.95       1889.5    No
3668-QPYBK           53.85       108.15   Yes
7795-CFOCW           42.30      1840.75    No
9237-HQITU           70.70       151.65   Yes
```

Issue: `SeniorCitizen` has values 0/1 instead of No/Yes

[6]:

```
data['SeniorCitizen'] = data['SeniorCitizen'].map({1: 'Yes', 0: 'No'})
```

Issue: There are some values in the `TotalCharges` column that prevent us from converting it to a numerical type.

[7]:

```
try:
    data['TotalCharges'].astype(float)
except ValueError as e:
    print(e)
```

```
could not convert string to float: ''
```

```
[8]: data[data['TotalCharges']==' '].groupby('tenure').
      ↪agg(occurance=('TotalCharges', 'count'))
```

```
[8]:         occurance
     tenure
     0              11
```

We see that these odd values appear only for customers which have their tenure equal to 0; meaning that they probably have not payed any bills yet. We will replace it with 0 then.

```
[9]: data['TotalCharges'] = data['TotalCharges'].str.replace(' ', '0').astype(float)
```

Clean column names.

```
[10]: data.columns = [col[0].upper() + col[1:] for col in data.columns]
```

1.0.3 Plot distribution summary

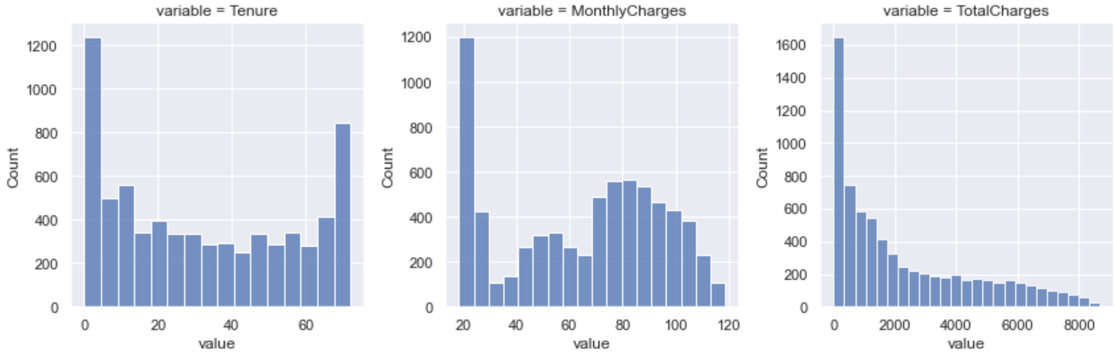
```
[11]: data.head()
```

```
[11]:             Gender SeniorCitizen Partner Dependents  Tenure PhoneService  \
      customerID
      7590-VHVEG  Female             No     Yes         No       1           No
      5575-GNVDE    Male             No      No         No      34          Yes
      3668-QPYBK    Male             No      No         No       2          Yes
      7795-CFOCW    Male             No      No         No      45           No
      9237-HQITU  Female             No      No         No       2          Yes

                     MultipleLines InternetService OnlineSecurity OnlineBackup  \
      customerID
      7590-VHVEG  No phone service             DSL             No          Yes
      5575-GNVDE                No             DSL            Yes           No
      3668-QPYBK                No             DSL            Yes          Yes
      7795-CFOCW  No phone service             DSL            Yes           No
      9237-HQITU                No     Fiber optic             No           No

                 DeviceProtection TechSupport StreamingTV StreamingMovies  \
      customerID
      7590-VHVEG               No          No          No              No
      5575-GNVDE              Yes          No          No              No
      3668-QPYBK               No          No          No              No
      7795-CFOCW              Yes         Yes          No              No
      9237-HQITU               No          No          No              No

                        Contract PaperlessBilling             PaymentMethod  \
      customerID
      7590-VHVEG  Month-to-month              Yes          Electronic check
      5575-GNVDE        One year               No              Mailed check
```

```
3668-QPYBK  Month-to-month              Yes              Mailed check
7795-CFOCW        One year               No  Bank transfer (automatic)
9237-HQITU  Month-to-month              Yes          Electronic check

            MonthlyCharges  TotalCharges Churn
customerID
7590-VHVEG           29.85         29.85    No
5575-GNVDE           56.95       1889.50    No
3668-QPYBK           53.85        108.15   Yes
7795-CFOCW           42.30       1840.75    No
9237-HQITU           70.70        151.65   Yes
```

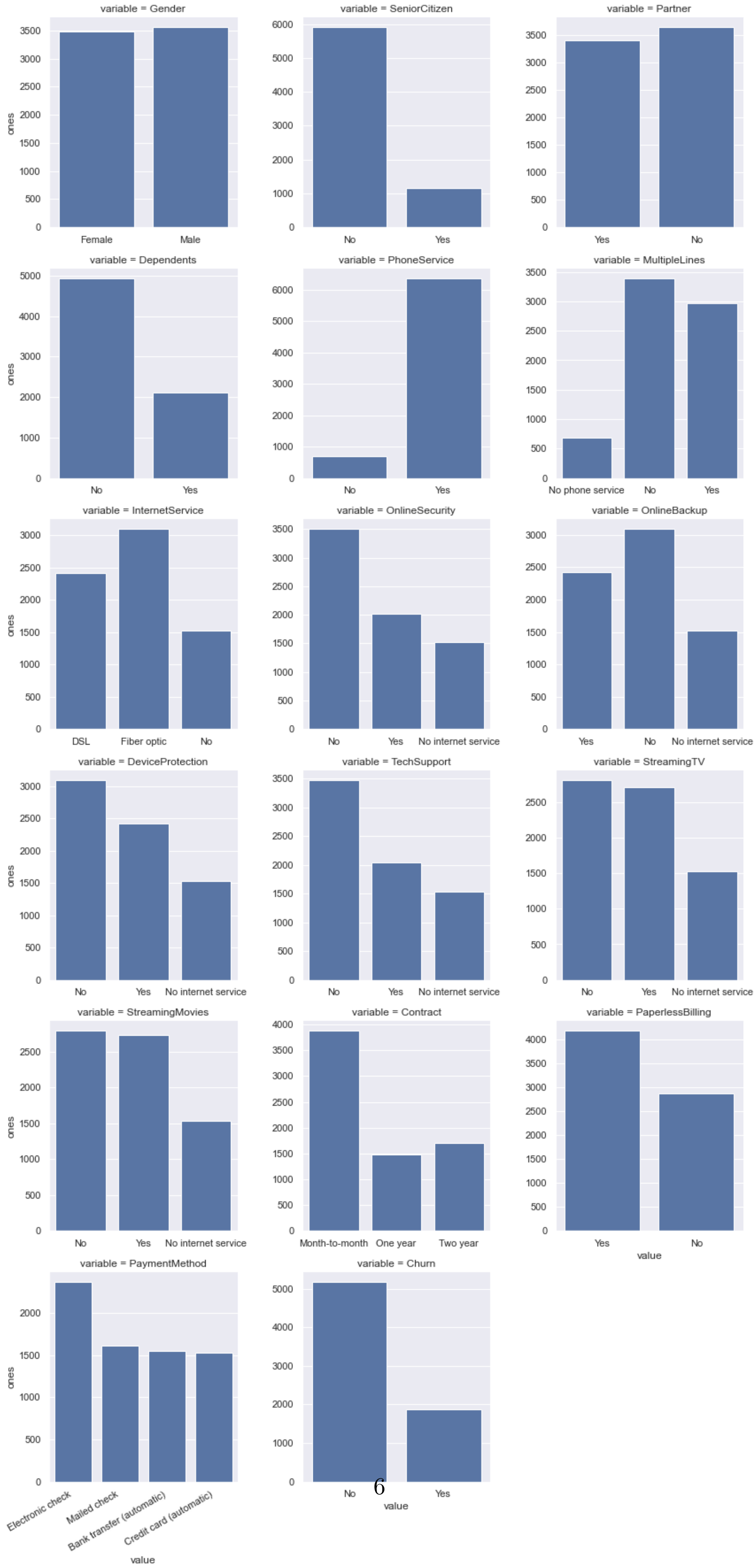
[12]:

```
srs.plot_distribution(data, columns_type='numerical');
```



[13]:

```
srs.plot_distribution(data, columns_type='objects');
```


variable = Gender
ones
0
500
1000
1500
2000
2500
3000
3500
Female
Male
variable = SeniorCitizen
0
1000
2000
3000
4000
5000
6000
No
Yes
variable = Partner
0
500
1000
1500
2000
2500
3000
3500
Yes
No
variable = Dependents
ones
0
1000
2000
3000
4000
5000
No
Yes
variable = PhoneService
0
1000
2000
3000
4000
5000
6000
No
Yes
variable = MultipleLines
0
500
1000
1500
2000
2500
3000
3500
No phone service
No
Yes
variable = InternetService
ones
0
500
1000
1500
2000
2500
3000
DSL
Fiber optic
No
variable = OnlineSecurity
0
500
1000
1500
2000
2500
3000
3500
No
Yes
No internet service
variable = OnlineBackup
0
500
1000
1500
2000
2500
3000
Yes
No
No internet service
variable = DeviceProtection
ones
0
500
1000
1500
2000
2500
3000
No
Yes
No internet service
variable = TechSupport
0
500
1000
1500
2000
2500
3000
3500
No
Yes
No internet service
variable = StreamingTV
0
500
1000
1500
2000
2500
No
Yes
No internet service
variable = StreamingMovies
ones
0
500
1000
1500
2000
2500
No
Yes
No internet service
variable = Contract
0
500
1000
1500
2000
2500
3000
3500
4000
Month-to-month
One year
Two year
variable = PaperlessBilling
0
500
1000
1500
2000
2500
3000
3500
4000
Yes
No
value
variable = PaymentMethod
ones
0
500
1000
1500
2000
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
value
variable = Churn
0
1000
2000
3000
4000
5000
No
Yes
value

[46]:
```
report = eda.create_report(data, title='EDA Summary')
report.save('../app/EDA-report')
```

```
  0%|                                                                                   ␣
↪                            | 0…
```

```
Report has been saved to /Users/monikakubek/Repositories/telco-customer-
churn/notebooks/../app/EDA-report.html!
```

This report will be later presented in our web app.

[14]:
```
data.describe(include='number')
```

[14]:
```
            Tenure  MonthlyCharges  TotalCharges
count  7043.000000     7043.000000   7043.000000
mean     32.371149       64.761692   2279.734304
std      24.559481       30.090047   2266.794470
min       0.000000       18.250000      0.000000
25%       9.000000       35.500000    398.550000
50%      29.000000       70.350000   1394.550000
75%      55.000000       89.850000   3786.600000
max      72.000000      118.750000   8684.800000
```

[15]:
```
data.describe(include='object')
```

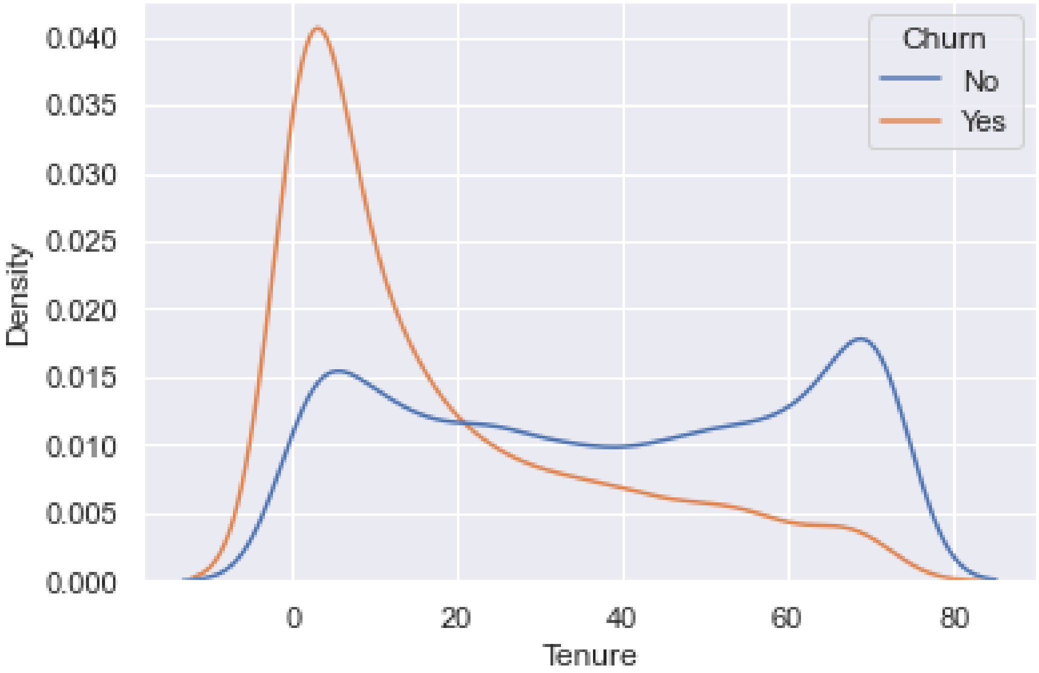
[15]:
```
       Gender SeniorCitizen Partner Dependents PhoneService MultipleLines  \
count    7043          7043    7043       7043         7043          7043
unique      2             2       2          2            2             3
top      Male            No      No         No          Yes            No
freq     3555          5901    3641       4933         6361          3390

       InternetService OnlineSecurity OnlineBackup DeviceProtection  \
count             7043           7043         7043             7043
unique               3              3            3                3
top        Fiber optic             No           No               No
freq              3096           3498         3088             3095

       TechSupport StreamingTV StreamingMovies        Contract  \
count         7043        7043            7043            7043
unique           3           3               3               3
top             No          No              No  Month-to-month
freq          3473        2810            2785            3875

       PaperlessBilling     PaymentMethod Churn
count              7043              7043  7043
unique                2                 4     2
```

```
top                    Yes  Electronic check    No
freq                  4171              2365  5174
```

1.0.4 Study feature importance

Tenure and charges

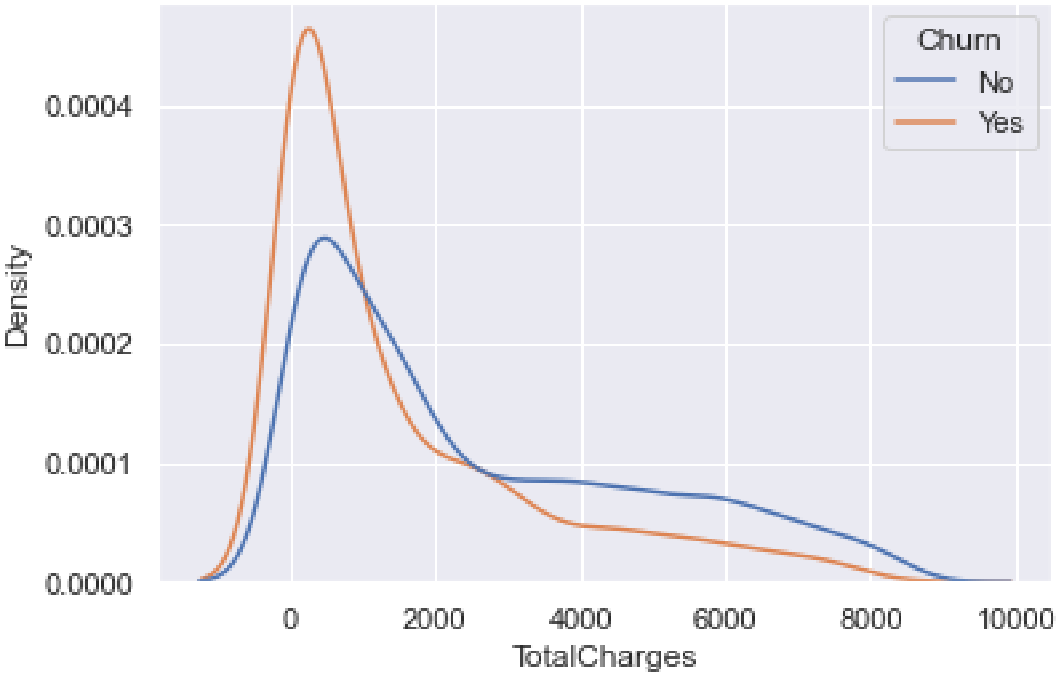
```
[16]: sns.kdeplot(data=data, x='Tenure', hue='Churn', common_norm=False);
```



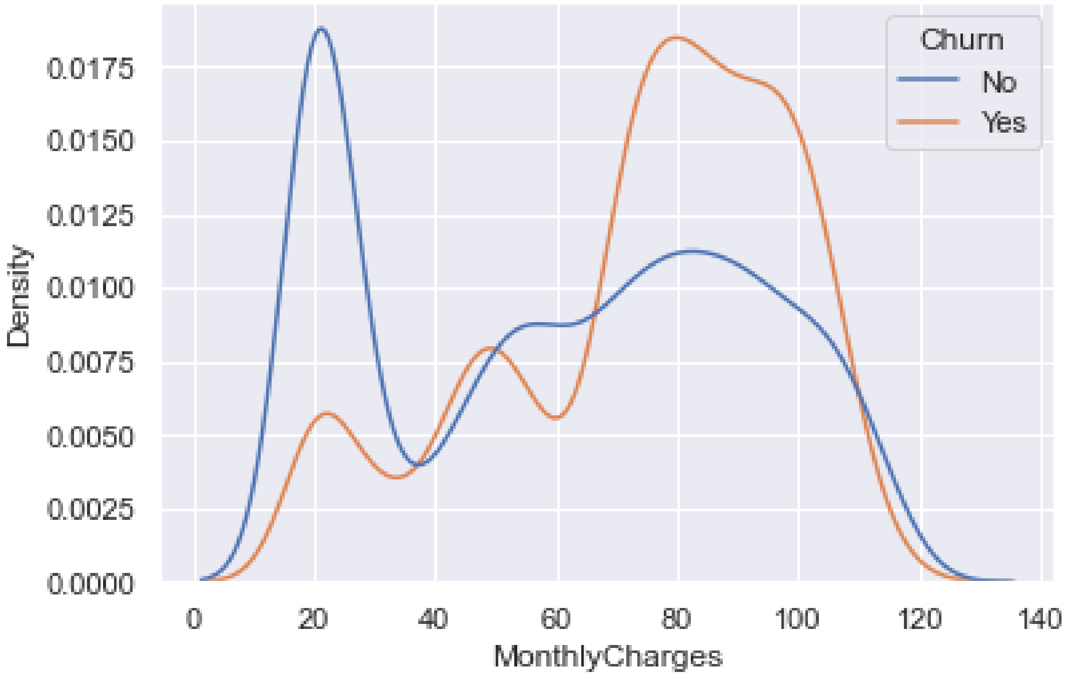
We can divide `Tenure` into three buckets: * 0-20: related to high churn * 21-50: related to medium churn * 50+: related to low churn

```
[17]: data['TenureBuckets'] = data['Tenure'].apply(srs.feature_tenure_bucket)
```

```
[18]: sns.kdeplot(data=data, x='TotalCharges', hue='Churn', common_norm=False);
```



[19]:
```
sns.kdeplot(data=data, x='MonthlyCharges', hue='Churn', common_norm=False);
```

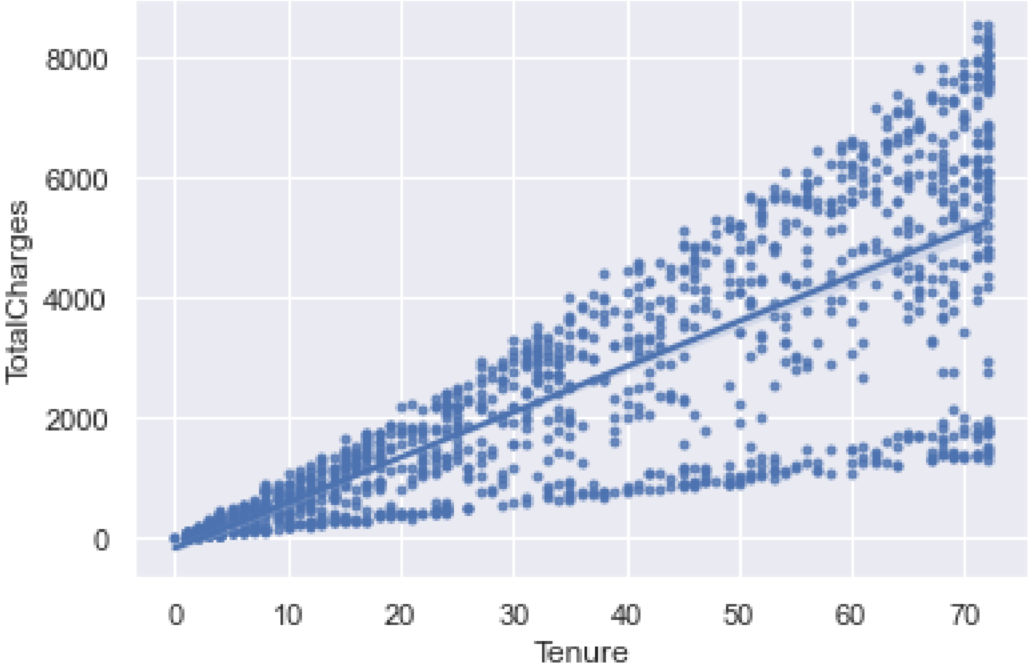


Here, we can also divide the MonthlyCharges values into three buckets: * 0-40: with low churn * 41-60: with medium churn * 60+: with high churn

[20]:
```
data['MonthlyChargesBuckets'] = data['MonthlyCharges'].apply(srs.
 ↪feature_monthlycharges_bucket)
```

Tenure must correlate with TotalChurges, let's investiage it.

[21]:
```
sns.regplot(data=data.sample(frac=0.2), x='Tenure', y='TotalCharges',␣
 ↪x_ci=None, marker='.');
```



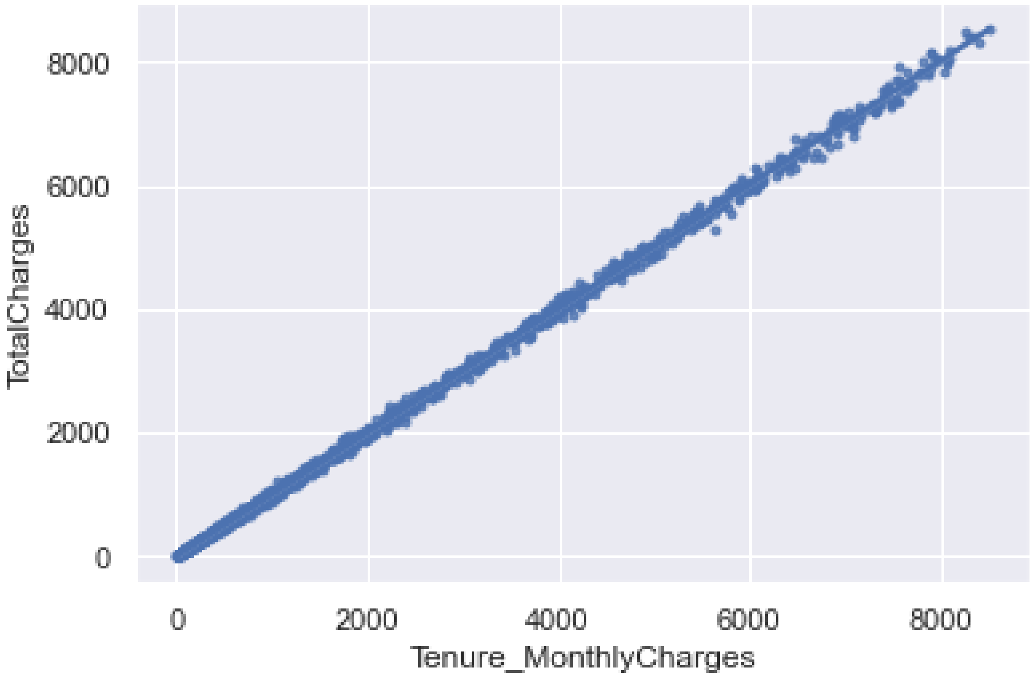
[22]:
```
from scipy.stats import kendalltau, pearsonr

kendalltau(data['Tenure'].values, data['TotalCharges'].values)
```

[22]:
```
KendalltauResult(correlation=0.7348547875506766, pvalue=0.0)
```

We can include also the information about monthly charges to check if it further improves the correlation.

[23]:
```
data['Tenure_MonthlyCharges'] = data['Tenure'] * data['MonthlyCharges']
```

[24]:
```
sns.regplot(data=data.sample(frac=0.2), x='Tenure_MonthlyCharges',␣
 ↪y='TotalCharges', x_ci=None, marker='.');
```



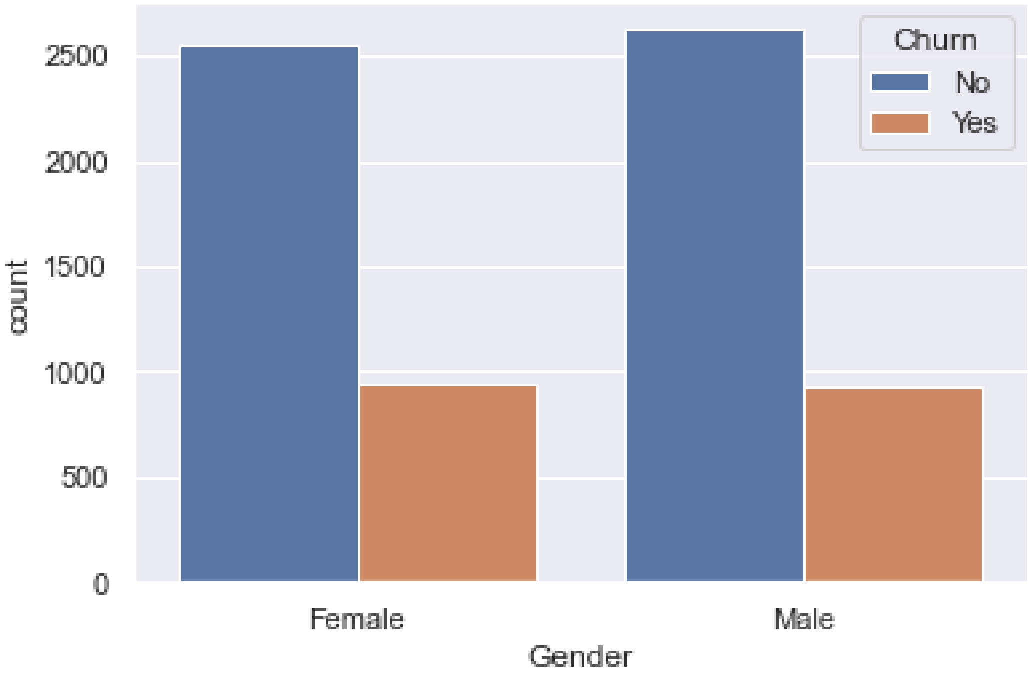
```
[25]: pearsonr(data['Tenure_MonthlyCharges'].values, data['TotalCharges'].values)
```

```
[25]: (0.9995605537972276, 0.0)
```

Here we can see almost perferct correlation. Having the same information from these two features, maybe the `TotalCharges` is a redundant one.

Gender

```
[26]: sns.countplot(data=data, x='Gender', hue='Churn');
```

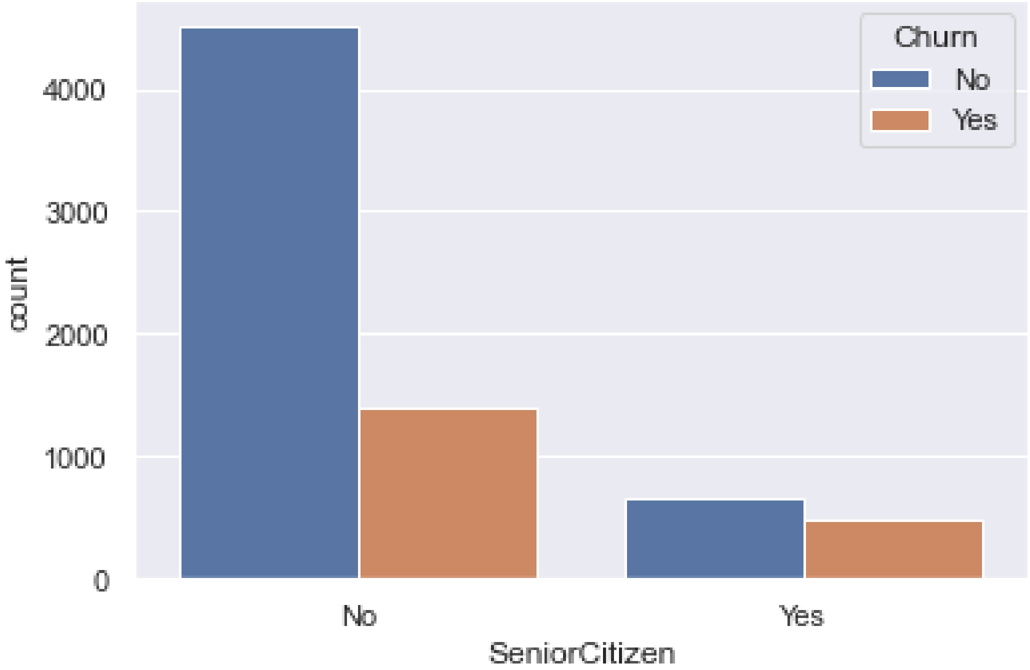


There is barely any difference in churn between the genders.

Senior citizen

[27]:

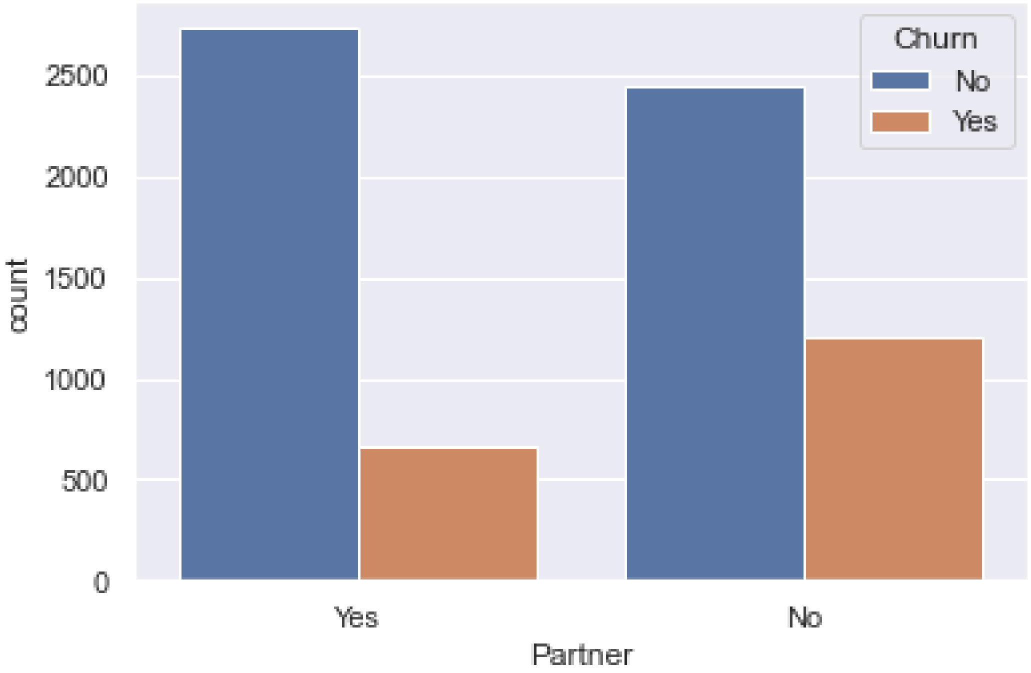
```
sns.countplot(data=data, x='SeniorCitizen', hue='Churn');
```



We see that senior customers are less likely to churn.

Partner

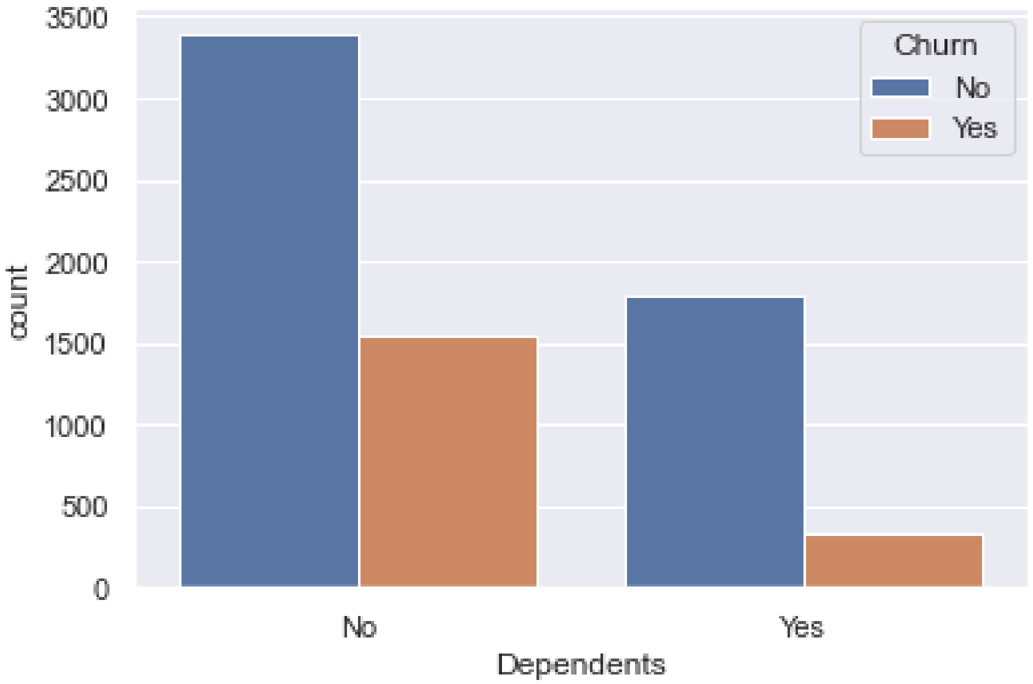
[28]:
```
sns.countplot(data=data, x='Partner', hue='Churn');
```



Customers without a partner are more likely to churn.

Dependents

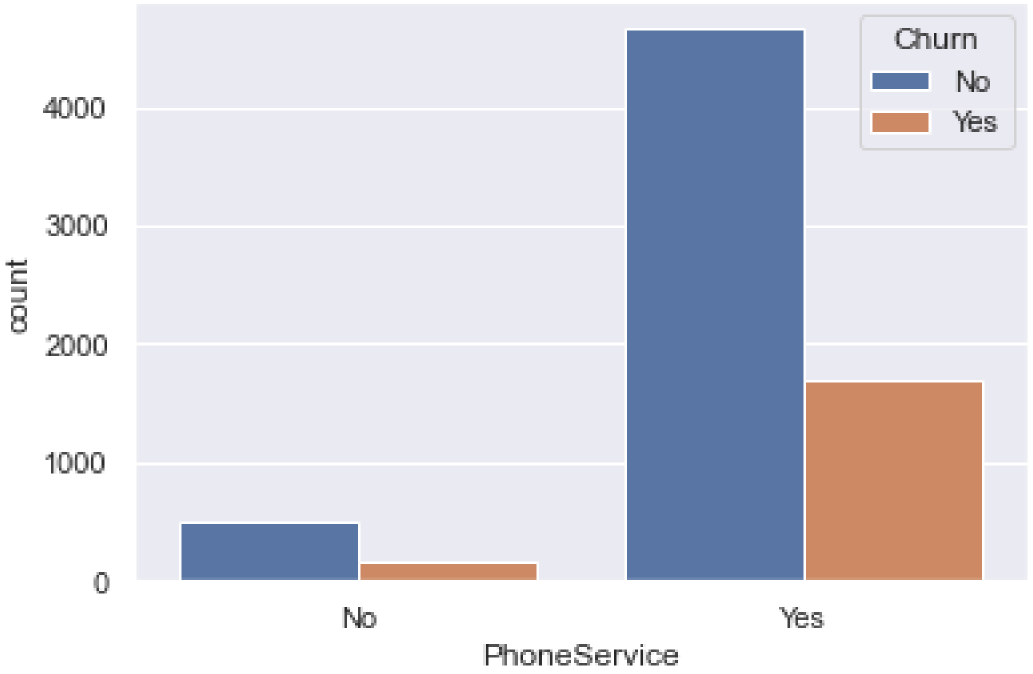
[29]:
```
sns.countplot(data=data, x='Dependents', hue='Churn');
```



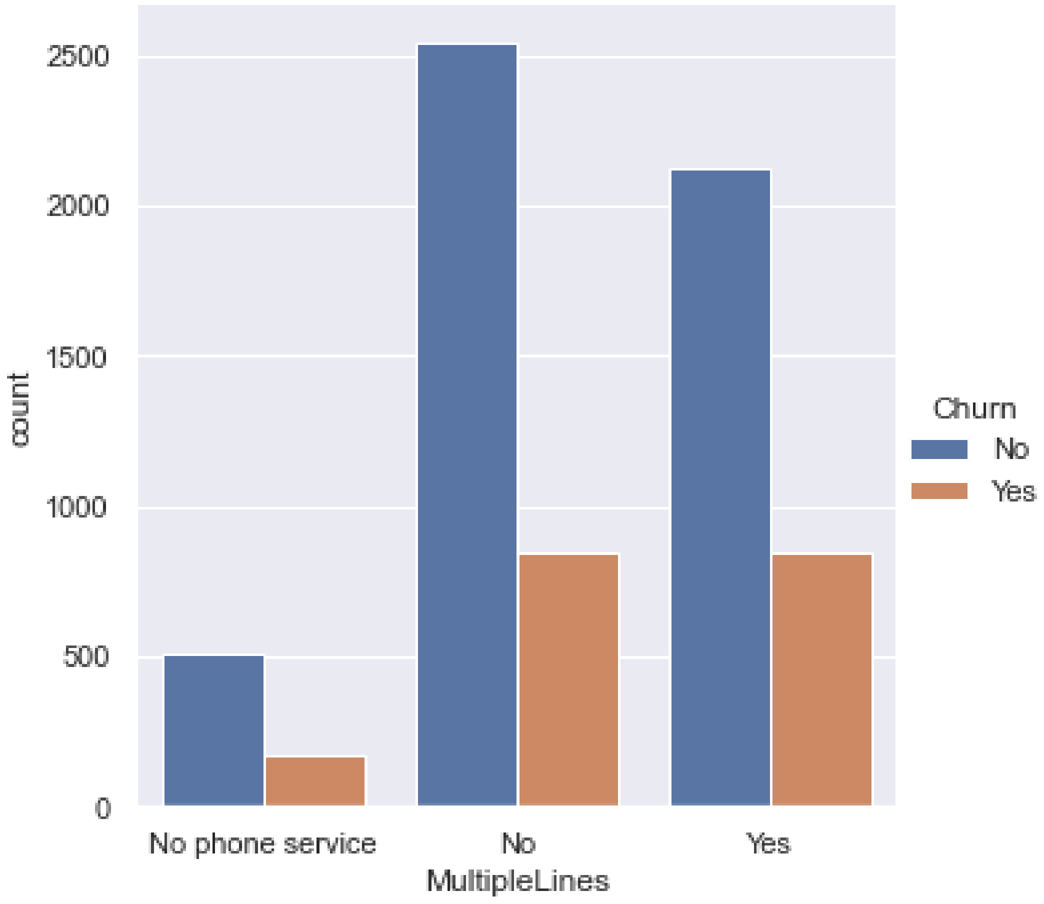
Customers without any dependents are more likely to churn,

Phone service

```
[30]: sns.countplot(data=data, x='PhoneService', hue='Churn');
```



```
[31]: sns.catplot(data=data, x='MultipleLines', hue='Churn', kind='count');
```



The results look similar for all the categories. Let's look at exact numbers.

```
[32]: srs.heatmap_churned_customers_share(data, columns='PhoneService')
```

```
[32]: <pandas.io.formats.style.Styler at 0x7fed418997c0>
```

```
[33]: srs.heatmap_churned_customers_share(data, columns='MultipleLines')
```

```
[33]: <pandas.io.formats.style.Styler at 0x7fed532f92e0>
```

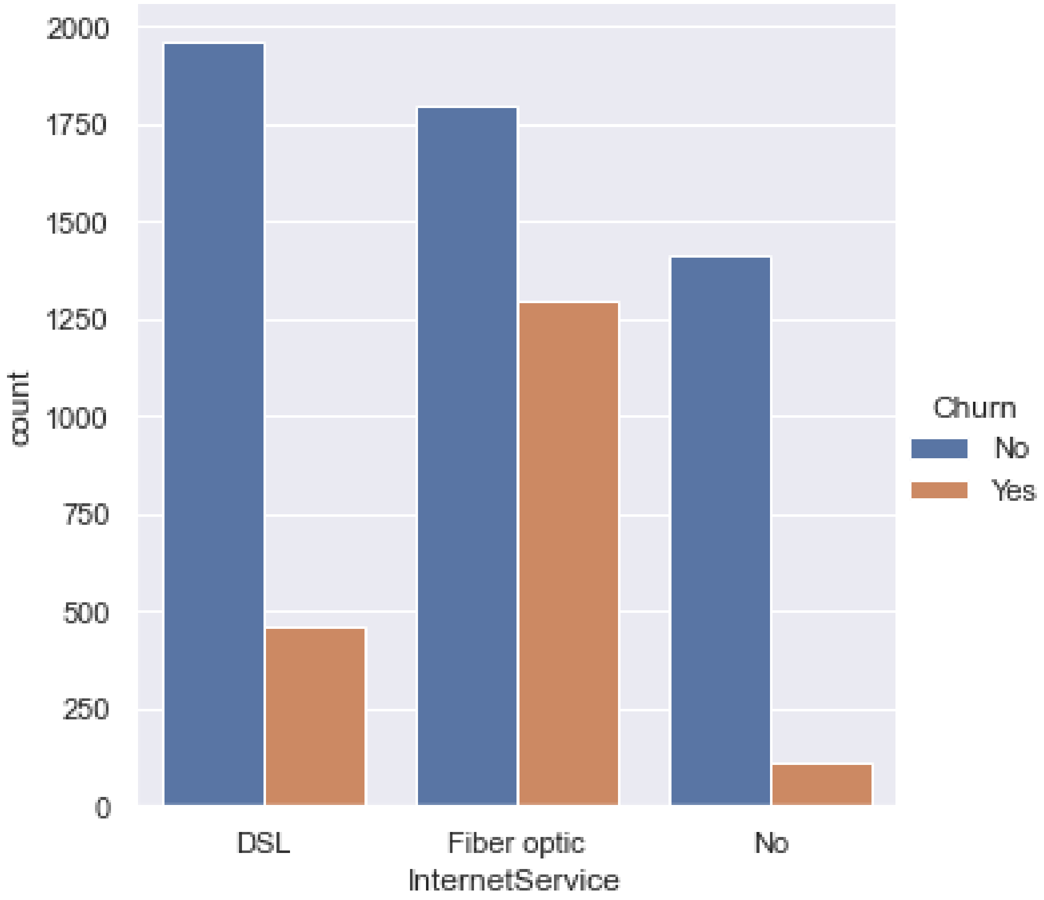
Actually, the category `MultipleLines` contains basically the same information as `PhoneService`, but with additional detail about the number of lines for customers who do use the phone service.

There is also no difference in churn in the group without phone service and the group with only one line. Hence, we may combine it into one for simplicity.

```
[34]: data['MultipleLinesBuckets'] = data['MultipleLines'].apply(srs.
       ↪feature_multiplelines_bucket)
```

Internet services

```
[35]: sns.catplot(data=data, x='InternetService', hue='Churn', kind='count');
```



We see highest churn among customers who use the fiber optic service. Let's check whether various additional services influence the probability of churn.

```
[36]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'OnlineSecurity'])
```

```
[36]: <pandas.io.formats.style.Styler at 0x7fed54661cd0>
```

```
[37]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'OnlineBackup'])
```

```
[37]: <pandas.io.formats.style.Styler at 0x7fed41a6d8e0>
```

```
[38]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'DeviceProtection'])
```

```
[38]: <pandas.io.formats.style.Styler at 0x7fed41a039a0>
```

```
[39]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'TechSupport'])
```

```
[39]: <pandas.io.formats.style.Styler at 0x7fed5463fa60>
```

```
[40]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'StreamingTV'])
```

```
[40]: <pandas.io.formats.style.Styler at 0x7fed41a59a00>
```

```
[41]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'StreamingMovies'])
```

```
[41]: <pandas.io.formats.style.Styler at 0x7fed310320a0>
```

If a customer has additional services enabled, then they are less likely to churn. Let's check now whether the number of additional services used also influences the probability of churn.

```
[42]: data['NumInternetlServices'] = srs.feature_numinternetservices(data)
```

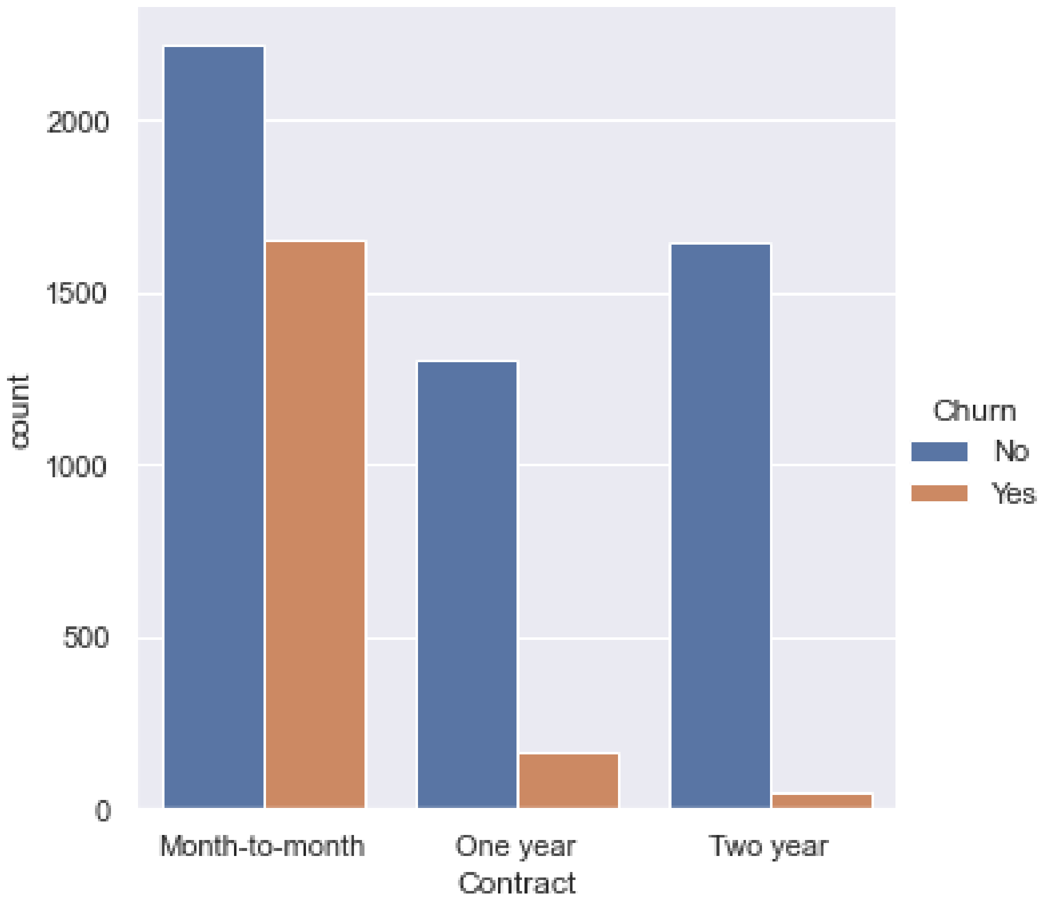
```
[43]: srs.heatmap_churned_customers_share(data, columns=['InternetService',␣
       ↪'NumInternetlServices'])
```

```
[43]: <pandas.io.formats.style.Styler at 0x7fed54666df0>
```

As expected, the more services one uses the less likely they are to churn.

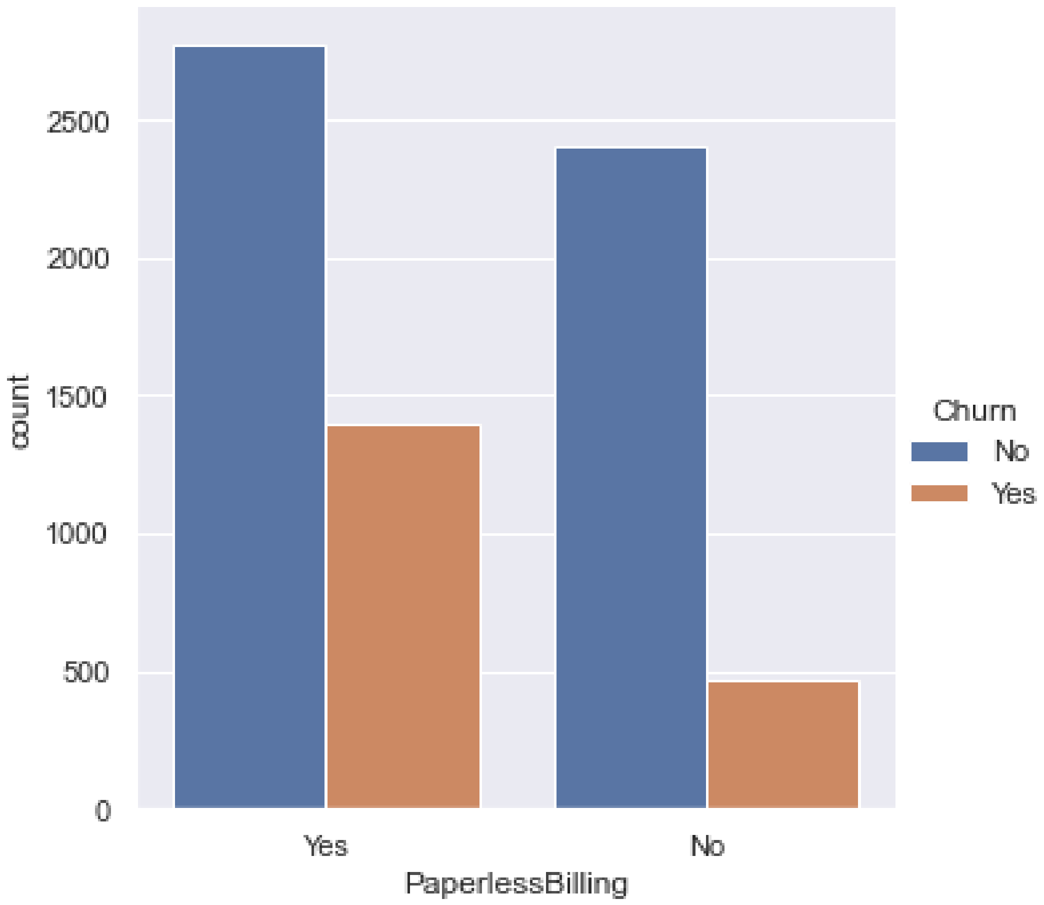
Contract and payment

```
[44]: sns.catplot(data=data, x='Contract', hue='Churn', kind='count');
```



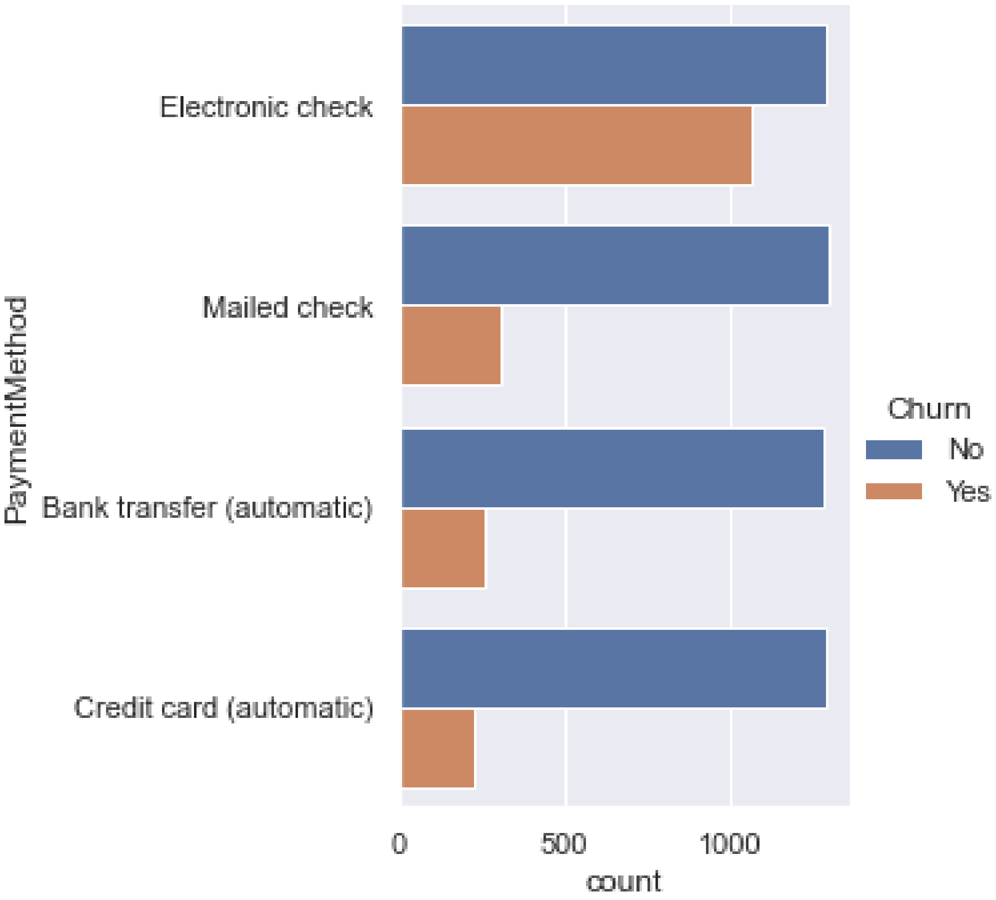
Customers with shorter contracts are more likely to churn.

```
[45]: sns.catplot(data=data, x='PaperlessBilling', hue='Churn', kind='count');
```



Customers with paperless billing are more likely to churn.

```
[46]: sns.catplot(data=data, y='PaymentMethod', hue='Churn', kind='count');
```



Customers who do not use automatic payment methods are more likely to churn.

1.0.5 Save the transformed data

[47]:

```
cols_for_model = [
    'Gender', 'SeniorCitizen', 'Partner', 'Dependents',
    'PhoneService', 'InternetService', 'OnlineSecurity',
    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
    'TenureBuckets', 'MonthlyChargesBuckets', 'MultipleLinesBuckets',
    'NumInternetlServices', 'Churn'
]
```

[48]:

```
data[cols_for_model].to_csv('../data/transformed.csv')
```

[ ]:

```
```