main

July 20, 2021

Importing libraries

The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

```
Requirement already satisfied: ppscore in /usr/local/lib/python3.7/dist-packages
(1.2.0)
Requirement already satisfied: scikit-learn<1.0.0,>=0.20.2 in
/usr/local/lib/python3.7/dist-packages (from ppscore) (0.22.2.post1)
Requirement already satisfied: pandas<2.0.0,>=1.0.0 in
/usr/local/lib/python3.7/dist-packages (from ppscore) (1.1.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/dist-packages (from pandas<2.0.0,>=1.0.0->ppscore)
(2.8.1)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.7/dist-
packages (from pandas<2.0.0,>=1.0.0->ppscore) (1.19.5)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-
packages (from pandas<2.0.0,>=1.0.0->ppscore) (2018.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-
packages (from python-dateutil>=2.7.3->pandas<2.0.0,>=1.0.0->ppscore) (1.15.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn<1.0.0,>=0.20.2->ppscore) (1.0.1)
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn<1.0.0,>=0.20.2->ppscore) (1.4.1)
```

Data Analysis

[67]:

	customerID	gender	SeniorCitizen	…	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	Female	0	…	29.85	29.85	No
1	5575-GNVDE	Male	0	…	56.95	1889.5	No
2	3668-QPYBK	Male	0	…	53.85	108.15	Yes
3	7795-CFOCW	Male	0	…	42.30	1840.75	No
4	9237-HQITU	Female	0	…	70.70	151.65	Yes

```
[5 rows x 21 columns]
```

Description of a few features: * gender - Whether the customer is a male or a female * SeniorCitizen - Whether the customer is a senior citizen or not (1, 0) * Partner - Whether the customer has a partner or not (Yes, No) * Dependents - Whether the customer has dependents or not (Yes, No) * tenure - Number of months the customer has stayed with the company * PhoneService - Whether the customer has a phone service or not (Yes, No) * MultipleLines - Whether the customer has multiple lines or not, that is capable of holding some calls (Yes, No, No phone service) * InternetService - Customer's internet service provider (DSL, Fiber optic, No) * OnlineSecurity - Whether the customer has online security or not (Yes, No, No internet service)

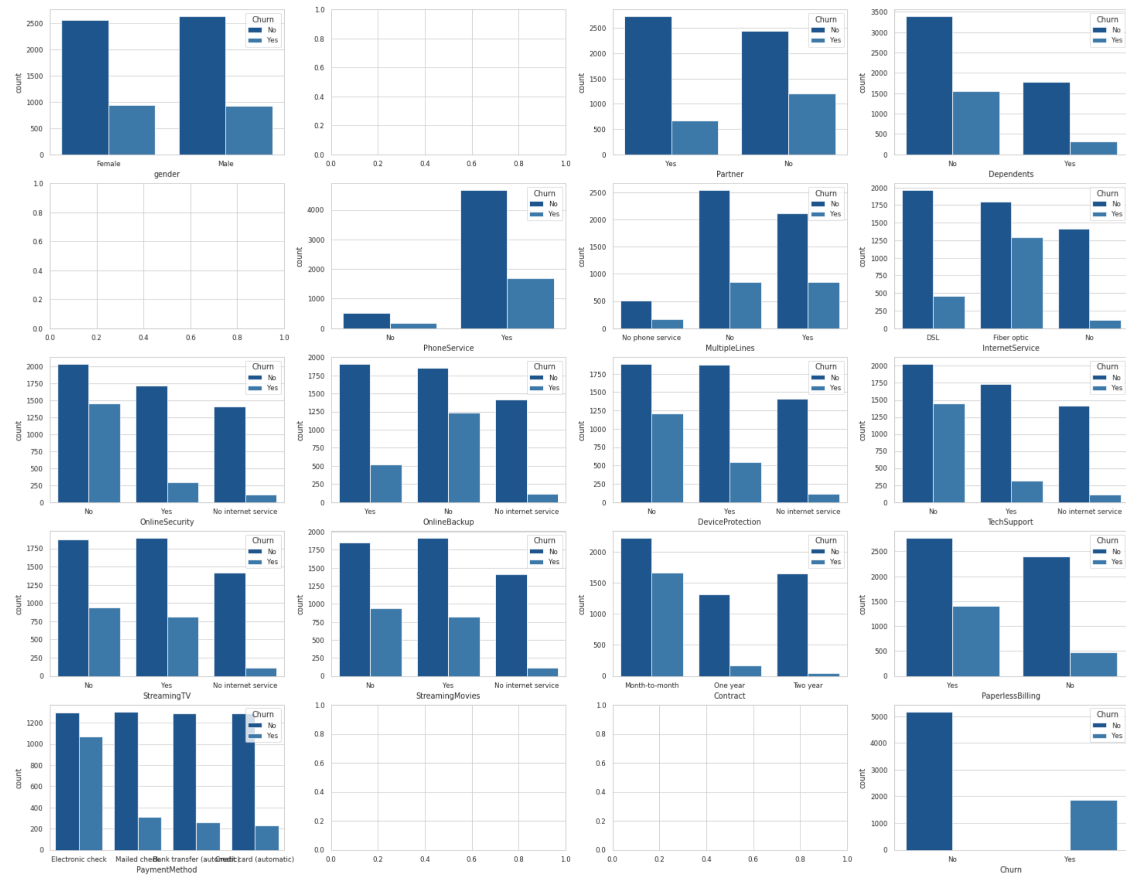
[68]:
```
(7043, 21)
```

```
<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
```

[70]:
```
array(['29.85', '1889.5', '108.15', …, '346.45', '306.6', '6844.5'],
      dtype=object)
```

Most of the data is categorical (might be ordinal or nominal). Number of features = 21 (20, if we ignore Customer ID) There are no missing or null values

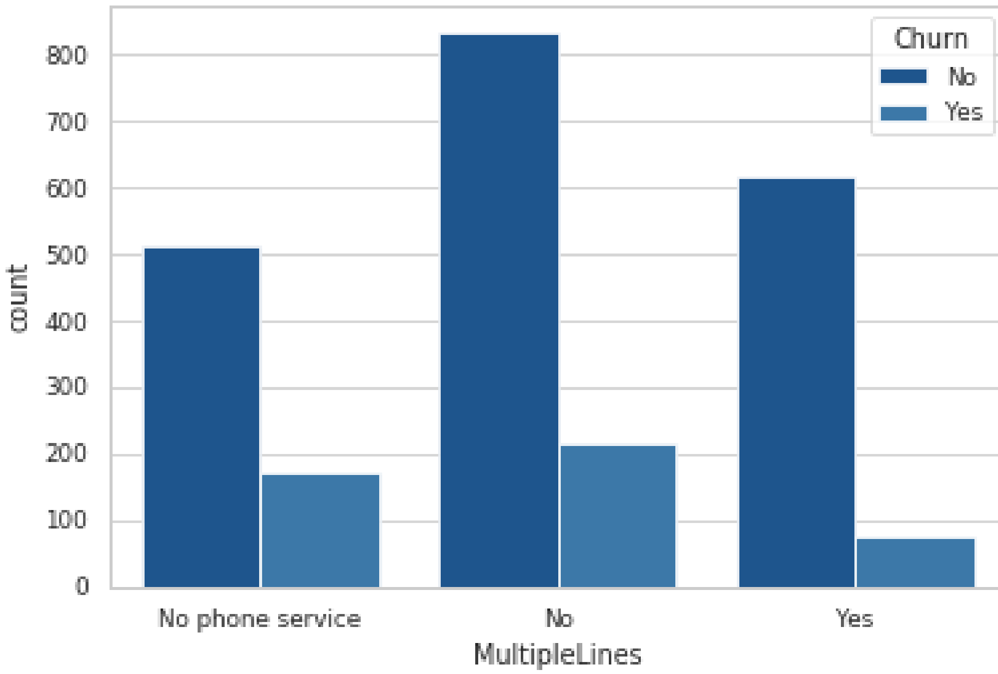
<Figure size 432x288 with 0 Axes>

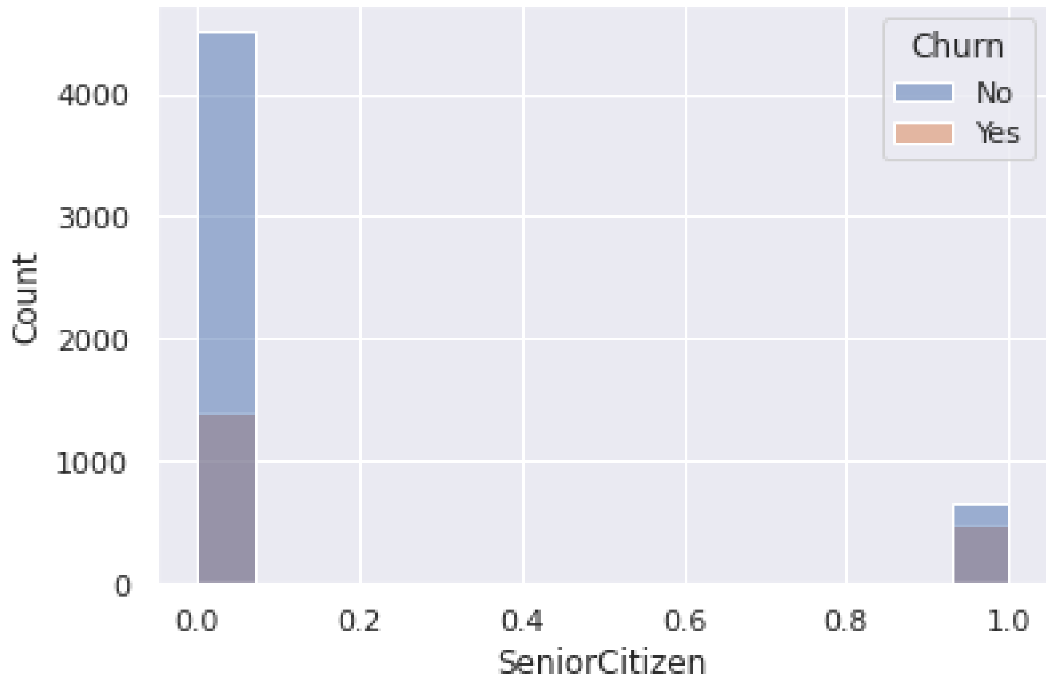


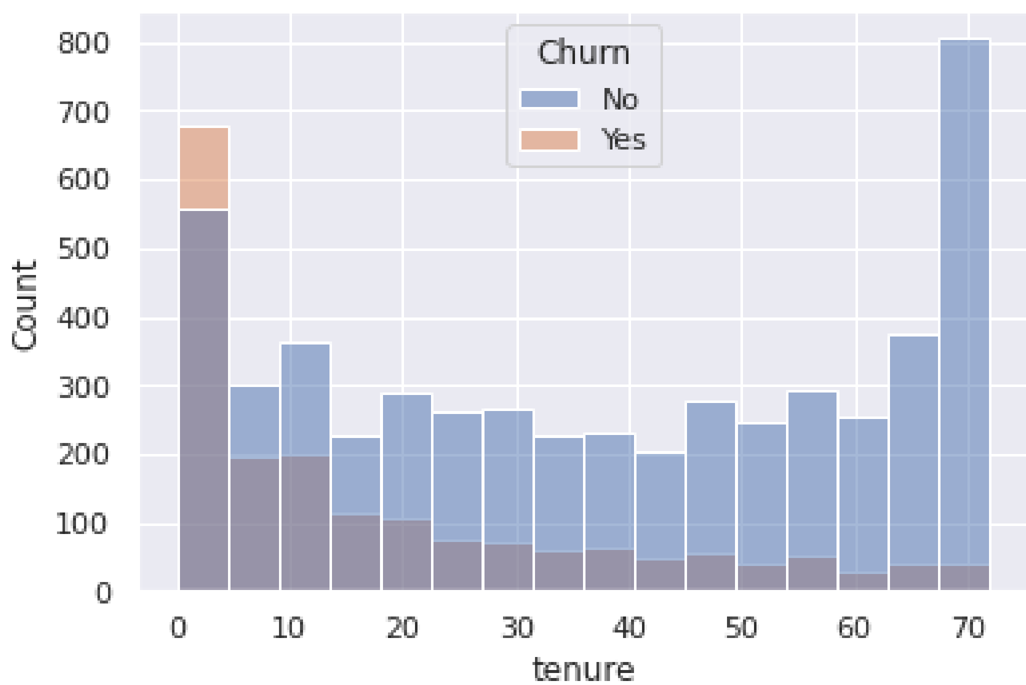
Observation from the above plot: * In the given data, around 71% people **didn't churn** * Churn distribution is almost similar for both genders * Electronic payment method has more churn rate as compared to other payment methods * Month-to-Month contract has higher churn rate than other contracts. It is logical too, cause once you have bought the service for an year or two, people generally do not prefer changing the service due to the hassle it might cause. * Customer with no Internet service (wherever this value is present) have a very low churn rate. (Maybe because they are not used to the modern technology, and do not prefer changing services) * People with device protection and Tech-Support (Yes) have less churn rate than people with no device protection or Tech-Support (No), indicating that people are satisfied with the services provided * Around 650 people didn't take phone service, number is relatively small compared to the total size of the dataset and won't be analyzed. * People with dependencies or partners have comparatively less churn rate (Around 12.5% for dependencies and 69.5% for partners), than people with no partnes or dependencies. * Customers with no internet services are generally from rural area or want to use their service for just calling or other purpose, thus leaving a very small margin for dissatisfaction and changing the service. * Another interesting pattern in internet-services is customer with DSL have relatievely very less churn rate as compared to Fiber-optics, indicating dissatisfaction with the latter.

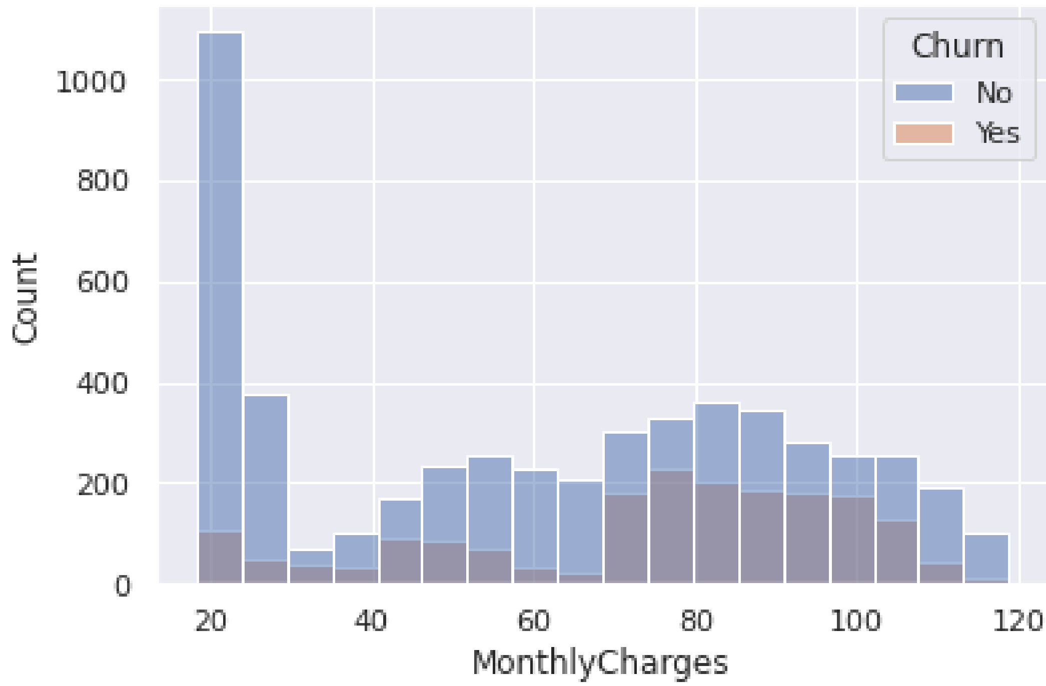
- A general patter can be observed: Customers who didn't take services like Online-Security, Online-Backup, Device-Support, Tech-Support, have higher churn rate.

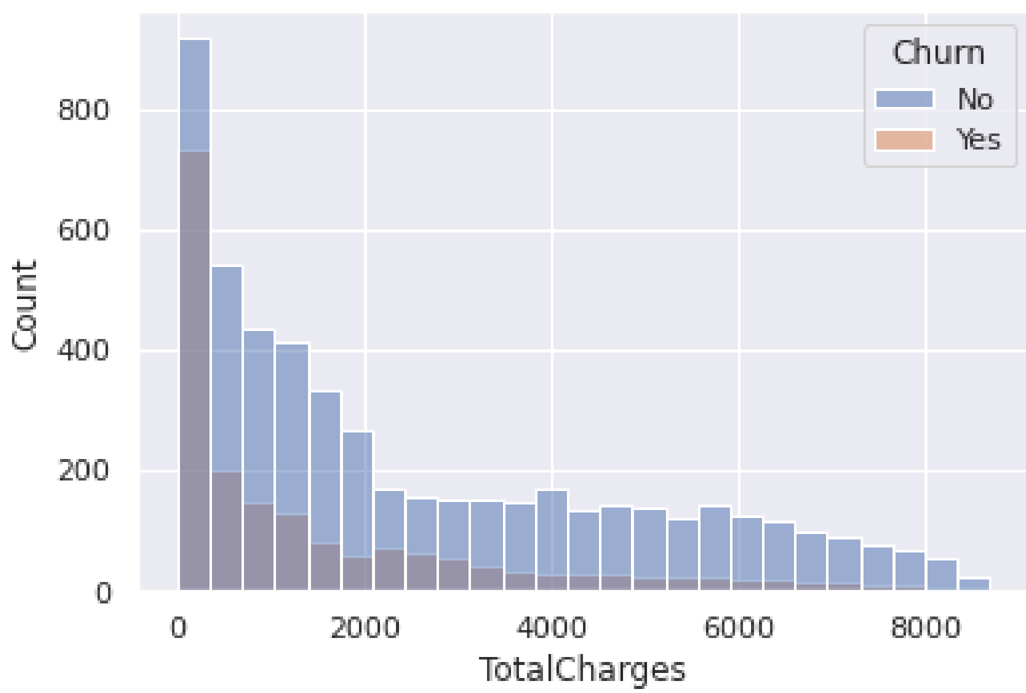
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```




Churn
No
Yes
4000
3000
2000
1000
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
SeniorCitizen


Churn
No
Yes
800
700
600
500
400
300
200
100
0
Count
0
10
20
30
40
50
60
70
tenure



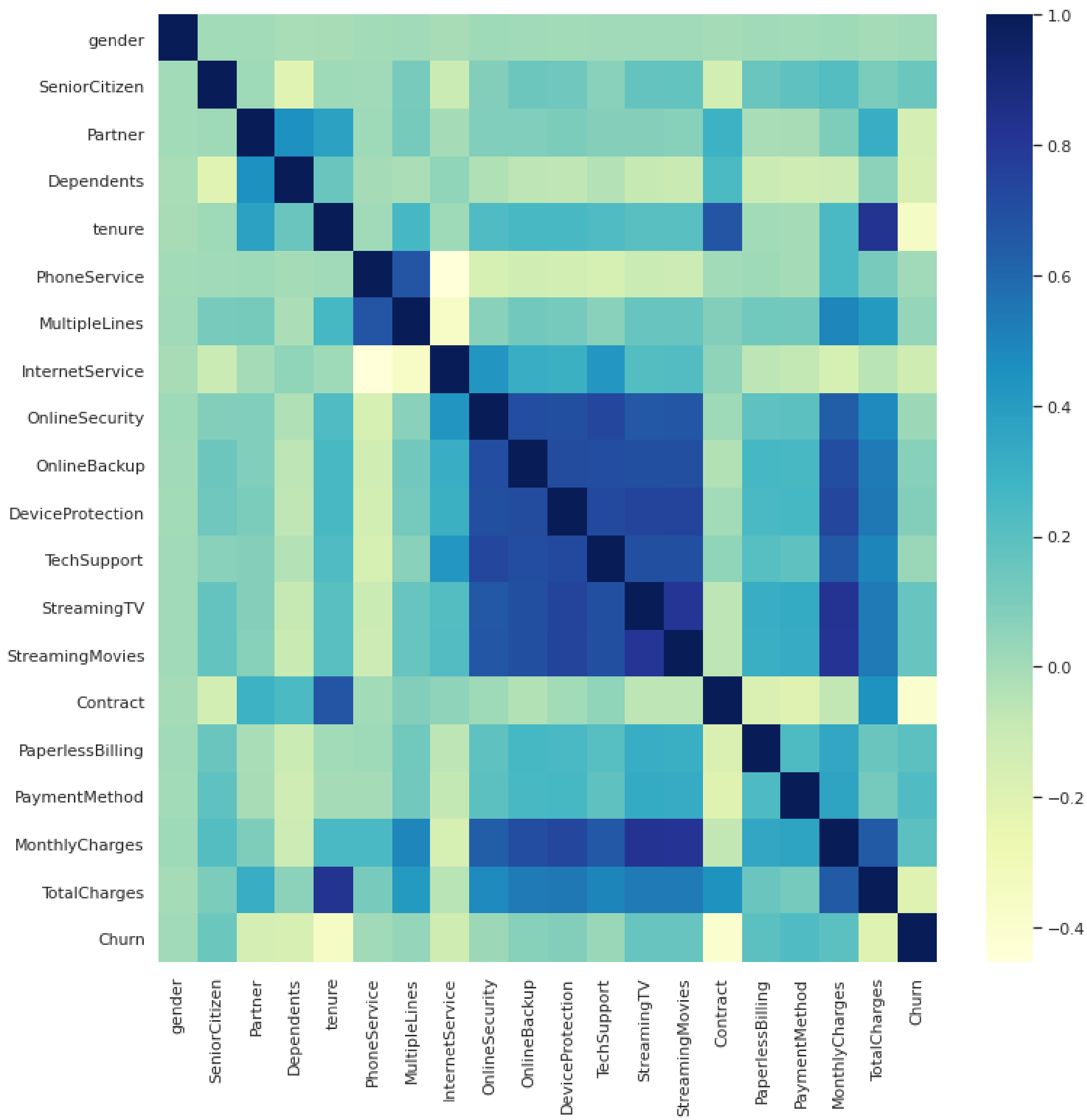


Observations from the above plots: * People with less value of tenure are more likely to churn. * Customer's whose monthly bill is less (<= 70) or among the greatest (>=110) are less likely to churn, than customers in the middle range. * Senior citizens are more likely to churn

Creating a copy of Dataframe, and replacing categorical values with numerical, for ease in further analysis.

[64]: array([3, 0, 1, 2])

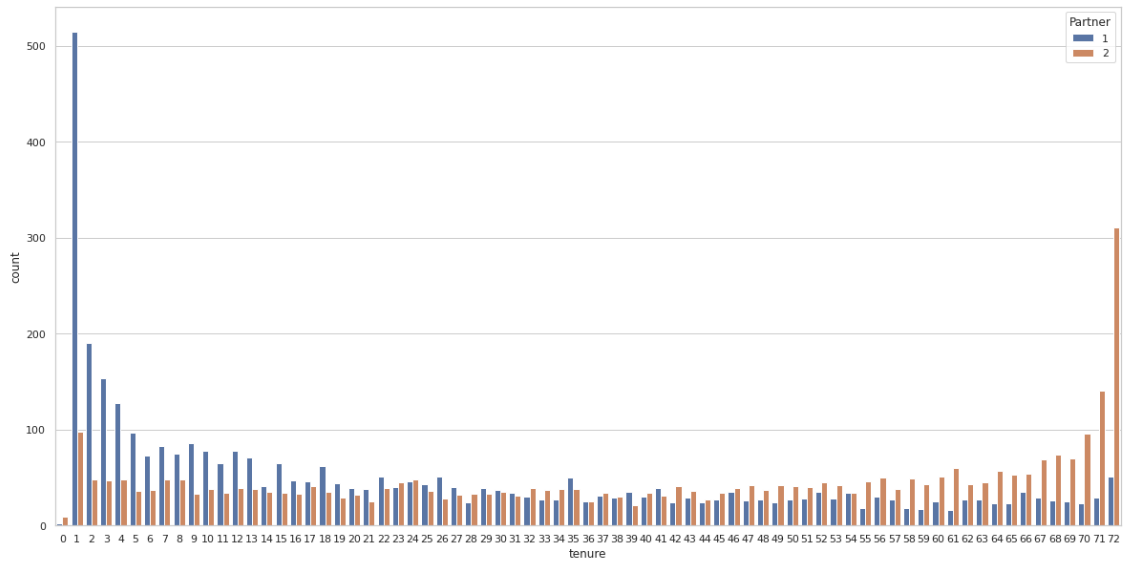
[19]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe818d17f10>



Observations from the above heatmap: * Contract,Total Charges and Tenure are highly correlated * Monthly charges increases when customers bought other services (Tech-Support, Onine backup, Streaming, etc.) * Most Senior citizens have no dependents * All the offered services are correlated, we can just replace them with one attribute Services * Tenure and Churn are negatively correlated as expected

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

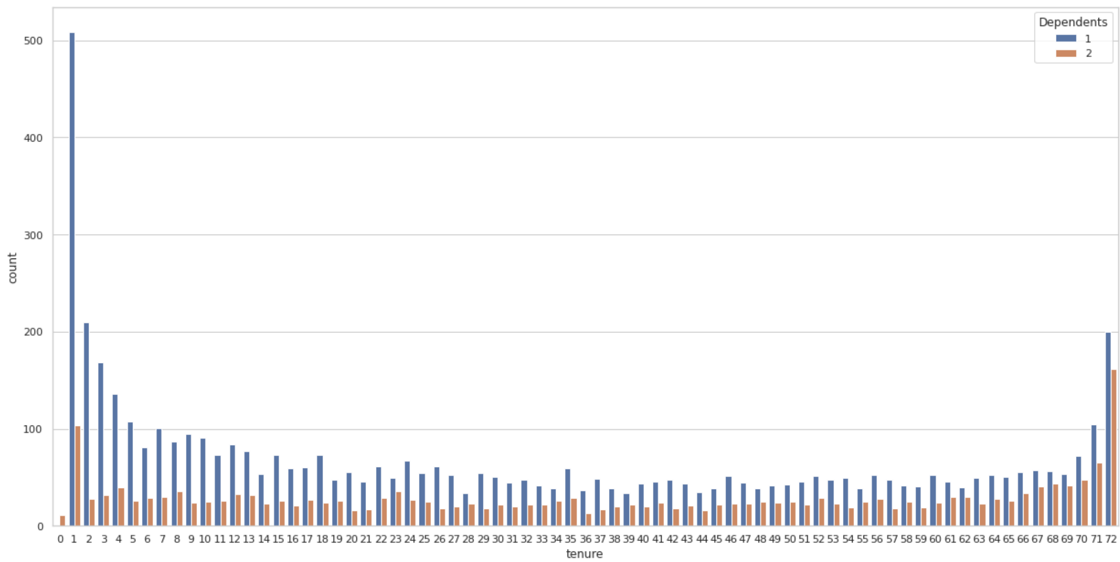
[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe807db8790>



If customers have partner, then their tenure is more likely to be greater as compared to the ones without a partner

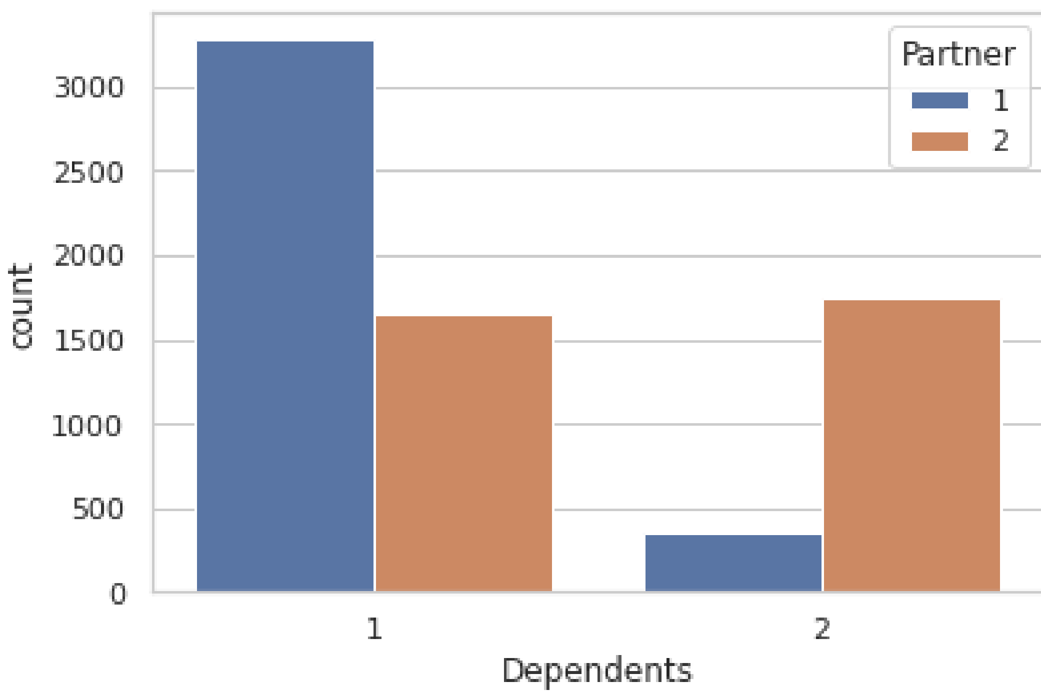
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7bb78c4d0>



Customers with no dependent are likely to have less tenure.

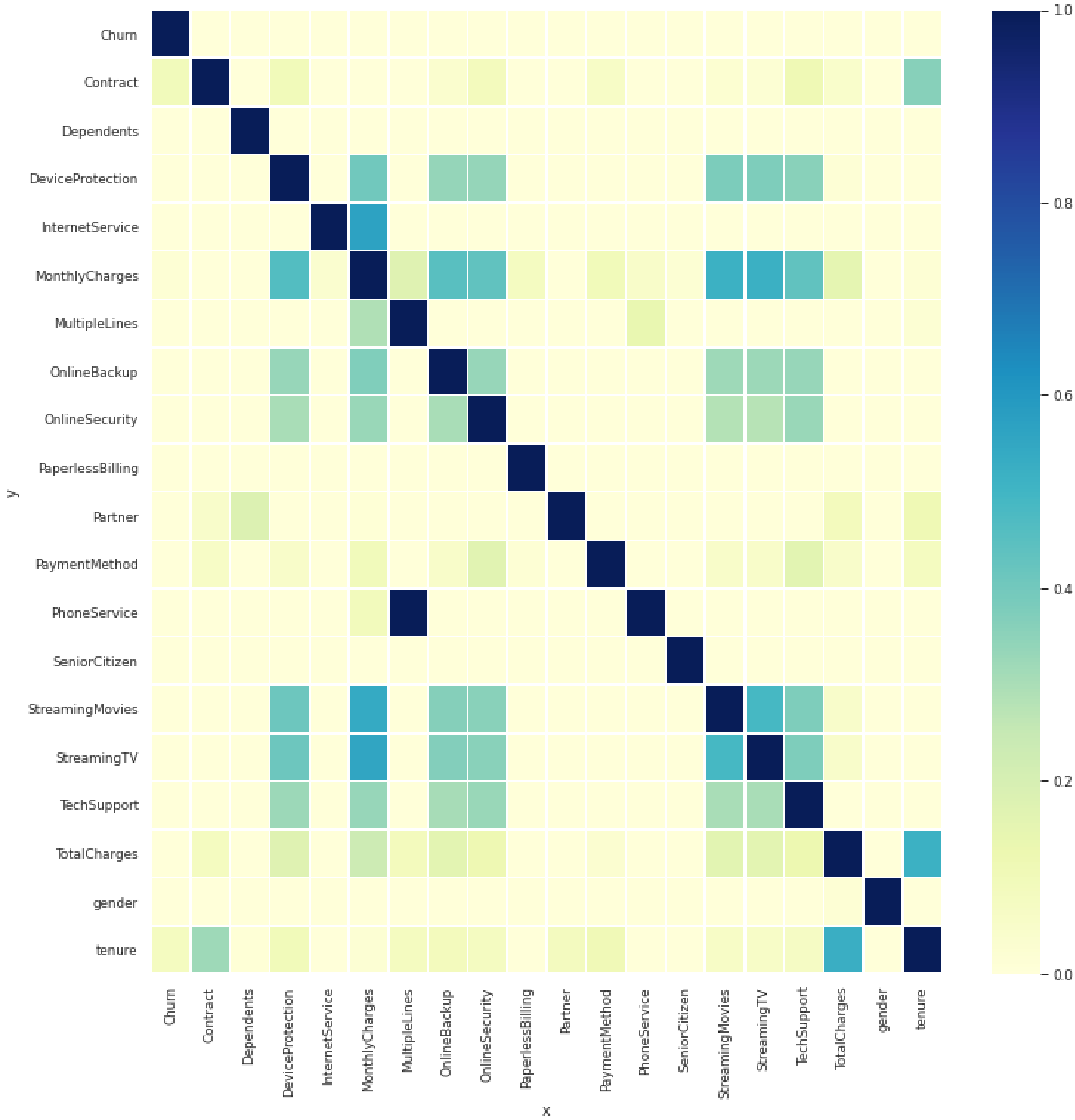
```
[46]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7bae07390>
```



We can club Dependents and Partner into one attribute

Plotting Heatmap of Predictive Power Score

[75]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7ba68a390>



- Services are good predictor of monthly charges,
- total charges and tenure are good predictors of each other
- Dependents can predict Partners , but the opposite is not True
- most relation observed are among services provided, charges (monthly and total) and tenure.