Ass14_telecom_users

June 21, 2025

[126]:

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

[128]:

```
df = pd.read_csv('telecom_users.csv')
df.head()
```

[128]:

```
   Unnamed: 0  customerID  gender  SeniorCitizen Partner Dependents  tenure  \
0        1869  7010-BRBUU    Male              0     Yes        Yes      72
1        4528  9688-YGXVR  Female              0      No         No      44
2        6344  9286-DOJGF  Female              1     Yes         No      38
3        6739  6994-KERXL    Male              0      No         No       4
4         432  2181-UAESM    Male              0      No         No       2

  PhoneService MultipleLines InternetService  …          StreamingTV  \
0          Yes           Yes              No  …  No internet service
1          Yes            No     Fiber optic  …                  Yes
2          Yes           Yes     Fiber optic  …                   No
3          Yes            No             DSL  …                   No
4          Yes            No             DSL  …                   No

       StreamingMovies        Contract PaperlessBilling  \
0  No internet service        Two year               No
1                   No  Month-to-month              Yes
2                   No  Month-to-month              Yes
3                  Yes  Month-to-month              Yes
4                   No  Month-to-month               No

               PaymentMethod MonthlyCharges TotalCharges Churn Provider  Age
0    Credit card (automatic)          24.10      1734.65    No      NaN  NaN
1    Credit card (automatic)          88.15      3973.20    No      NaN  NaN
2  Bank transfer (automatic)          74.95      2869.85   Yes      NaN  NaN
3           Electronic check          55.90       238.50    No      NaN  NaN
4           Electronic check          53.45       119.50    No      NaN  NaN

[5 rows x 24 columns]
```

[130]:

```
# Get information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6050 entries, 0 to 6049
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Unnamed: 0        6050 non-null   int64
 1   customerID        6050 non-null   object
 2   gender            6050 non-null   object
 3   SeniorCitizen     6050 non-null   int64
 4   Partner           6050 non-null   object
 5   Dependents        6050 non-null   object
 6   tenure            6050 non-null   int64
 7   PhoneService      6050 non-null   object
 8   MultipleLines     6050 non-null   object
 9   InternetService   6050 non-null   object
 10  OnlineSecurity    6050 non-null   object
 11  OnlineBackup      6050 non-null   object
 12  DeviceProtection  6050 non-null   object
 13  TechSupport       6050 non-null   object
 14  StreamingTV       6050 non-null   object
 15  StreamingMovies   6050 non-null   object
 16  Contract          6050 non-null   object
 17  PaperlessBilling  6050 non-null   object
 18  PaymentMethod     6050 non-null   object
 19  MonthlyCharges    6050 non-null   float64
 20  TotalCharges      6040 non-null   float64
 21  Churn             6050 non-null   object
 22  Provider          0 non-null      float64
 23  Age               0 non-null      float64
dtypes: float64(4), int64(3), object(17)
memory usage: 1.1+ MB
```

[132]:

```
# Remove unwanted fields

df.drop(columns=['Unnamed: 0', 'Provider', 'Age'], inplace=True)
df.head()
```

[132]:

```
     customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7010-BRBUU    Male              0     Yes        Yes      72          Yes
1  9688-YGXVR  Female              0      No         No      44          Yes
2  9286-DOJGF  Female              1     Yes         No      38          Yes
3  6994-KERXL    Male              0      No         No       4          Yes
4  2181-UAESM    Male              0      No         No       2          Yes

  MultipleLines InternetService       OnlineSecurity  …  \
0           Yes              No  No internet service  …
1            No     Fiber optic                   No  …
```

```
2           Yes     Fiber optic                   No  ...
3            No             DSL                   No  ...
4            No             DSL                  Yes  ...

      DeviceProtection          TechSupport          StreamingTV  \
0  No internet service  No internet service  No internet service
1                  Yes                   No                  Yes
2                   No                   No                   No
3                   No                   No                   No
4                  Yes                   No                   No

       StreamingMovies        Contract PaperlessBilling  \
0  No internet service        Two year               No
1                   No  Month-to-month              Yes
2                   No  Month-to-month              Yes
3                  Yes  Month-to-month              Yes
4                   No  Month-to-month               No

               PaymentMethod MonthlyCharges  TotalCharges  Churn
0    Credit card (automatic)          24.10       1734.65     No
1    Credit card (automatic)          88.15       3973.20     No
2  Bank transfer (automatic)          74.95       2869.85    Yes
3           Electronic check          55.90        238.50     No
4           Electronic check          53.45        119.50     No

[5 rows x 21 columns]
```

[134]:

```
# Remove duplicates

df.drop_duplicates(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5986 entries, 0 to 6049
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        5986 non-null   object
 1   gender            5986 non-null   object
 2   SeniorCitizen     5986 non-null   int64
 3   Partner           5986 non-null   object
 4   Dependents        5986 non-null   object
 5   tenure            5986 non-null   int64
 6   PhoneService      5986 non-null   object
 7   MultipleLines     5986 non-null   object
 8   InternetService   5986 non-null   object
 9   OnlineSecurity    5986 non-null   object
 10  OnlineBackup      5986 non-null   object
 11  DeviceProtection  5986 non-null   object
 12  TechSupport       5986 non-null   object
 13  StreamingTV       5986 non-null   object
 14  StreamingMovies   5986 non-null   object
 15  Contract          5986 non-null   object
 16  PaperlessBilling  5986 non-null   object
 17  PaymentMethod     5986 non-null   object
 18  MonthlyCharges    5986 non-null   float64
 19  TotalCharges      5976 non-null   float64
 20  Churn             5986 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.0+ MB
```

[136]:

```
# Checking the customerID unique values. ID should be unique and no duplicates

df['customerID'].nunique()
```

[136]:

```
5986
```

[138]:

```
# Checking gender field. There should be only 2 - Male and Female

df.gender.value_counts(dropna=False)
```

[138]:

```
gender
Male      3050
Female    2936
Name: count, dtype: int64
```

[15]:

```
# Checking SeniorCitizen field. Here there are 2 values - 0 for Not a senior␣
  ↪citizen and the 1 for a senior citizen
df.SeniorCitizen.value_counts()

#df.SeniorCitizen.nunique()
```

[15]:

```
SeniorCitizen
0    5020
1     966
Name: count, dtype: int64
```

[17]:

```
# Checking the Partner field. We have No and Yes

df.Partner.value_counts(dropna=False)
```

[17]:

```
Partner
No     3082
Yes    2904
Name: count, dtype: int64
```

[19]:
```
# Checking the Dependents Field, values are either Yes or No

df.Dependents.value_counts(dropna=False)
```

[19]:
```
Dependents
No     4195
Yes    1791
Name: count, dtype: int64
```

[21]:
```
# Checking the tenure field. This should have numbers for different durations␣
 ↪for the subscription

df.tenure.value_counts(dropna=False)
```

[21]:
```
tenure
1     510
72    308
2     194
3     169
4     154
     …
45     47
44     47
39     44
36     43
0      10
Name: count, Length: 73, dtype: int64
```

[23]:
```
# Checking the PhoneService field. EitherYes or No. This is complete.

df.PhoneService.value_counts(dropna=False)
```

[23]:
```
PhoneService
Yes    5396
No      590
Name: count, dtype: int64
```

[25]:
```
# Checking the MultipleLines field. Here, there are three values

df.MultipleLines.value_counts(dropna=False)
```

[25]:
```
MultipleLines
No                  2848
Yes                 2548
No phone service     590
Name: count, dtype: int64
```

[27]:

```
# Checking the InternetService field. There are three values

df.InternetService.value_counts(dropna=False)
```

[27]:

```
InternetService
Fiber optic    2627
DSL            2068
No             1291
Name: count, dtype: int64
```

[29]:

```
# Online security field. This has three values


df.OnlineSecurity.value_counts()
```

[29]:

```
OnlineSecurity
No                     2982
Yes                    1713
No internet service    1291
Name: count, dtype: int64
```

[31]:

```
# Checking the online backup field. There are three values


df.OnlineBackup.value_counts(dropna=False)
```

[31]:

```
OnlineBackup
No                     2605
Yes                    2090
No internet service    1291
Name: count, dtype: int64
```

[33]:

```
# Checking the Device protection field. There are three values


df.DeviceProtection.value_counts(dropna=False)
```

[33]:

```
DeviceProtection
No                     2640
Yes                    2055
No internet service    1291
Name: count, dtype: int64
```

[35]:

```
# Checking the TechSupport field. This has three values, either No, Yes, or No␣
 ↪internet service
df.TechSupport.value_counts(dropna=False)
```

[35]:
```
TechSupport
No                     2960
Yes                    1735
No internet service    1291
Name: count, dtype: int64
```

[37]:
```
# Checking the StreamingTv. There are three values also


df.StreamingTV.value_counts(dropna=False)
```

[37]:
```
StreamingTV
No                     2389
Yes                    2306
No internet service    1291
Name: count, dtype: int64
```

[39]:
```
# Checking the StreamingMovies field. Here again, three values


df.StreamingMovies.value_counts(dropna=False)
```

[39]:
```
StreamingMovies
No                     2356
Yes                    2339
No internet service    1291
Name: count, dtype: int64
```

[41]:
```
# Checking the Contract field. There are three contract types


df.Contract.value_counts(dropna=False)
```

[41]:
```
Contract
Month-to-month    3269
Two year          1441
One year          1276
Name: count, dtype: int64
```

[43]:
```
# Checking the PaperlessBilling field. It is either Yes or No


df.PaperlessBilling.value_counts(dropna=False)
```

[43]:
```
PaperlessBilling
Yes    3528
No     2458
Name: count, dtype: int64
```

[45]:
```
# Checking the PaymentMethod field. There are four options


df.PaymentMethod.value_counts(dropna=False)
```

[45]:
```
PaymentMethod
Electronic check             2006
Mailed check                 1369
Bank transfer (automatic)    1308
Credit card (automatic)      1303
Name: count, dtype: int64
```

[47]:
```
# Checking Monthly Charges field. This should be float values showing money␣
 ↪values


df.MonthlyCharges.value_counts(dropna=False)
```

[47]:
```
MonthlyCharges
20.05    50
19.85    40
19.70    39
19.55    37
19.95    36
         ..
73.70     1
52.00     1
42.40     1
96.70     1
21.15     1
Name: count, Length: 1526, dtype: int64
```

[49]:
```
# Checking the TotalCharges field. This should be float showing the total␣
 ↪charges

df.TotalCharges.value_counts(dropna=False)
```

[49]:
```
TotalCharges
NaN        10
20.20      10
19.75       8
19.55       7
20.05       6
           ..
4138.90     1
4428.60     1
6991.60     1
```

```
7040.85      1
457.30       1
Name: count, Length: 5611, dtype: int64
```

[51]:

```
# We have 10 rows with NaN. A close observation


df[df['TotalCharges'].isna()][['tenure', 'MonthlyCharges', 'TotalCharges']]
```

[51]:

```
      tenure  MonthlyCharges  TotalCharges
380        0           61.90           NaN
658        0           56.05           NaN
2814       0           25.35           NaN
3129       0           19.70           NaN
3298       0           19.85           NaN
4377       0           80.85           NaN
5439       0           20.25           NaN
5446       0           20.00           NaN
5759       0           52.55           NaN
6015       0           25.75           NaN
```

[53]:

```
# This reveals that the tenure for this missing TotalCharges rows is 0. This␣
 ↪makes sense because they have not charged yet.
# Instead of leaving them as NaN, we can multiply tenure by MonthlyCharges to␣
 ↪have the value in the TotalCharges for these rows

df.loc[df['tenure'] == 0, 'TotalCharges'] = df.loc[df['tenure'] == 0, 'tenure']␣
 ↪* df.loc[df['tenure'] == 0, 'MonthlyCharges']
df.TotalCharges.value_counts(dropna=False)
```

[53]:

```
TotalCharges
0.00        10
20.20       10
19.75        8
19.55        7
20.05        6
            ..
4138.90      1
4428.60      1
6991.60      1
7040.85      1
457.30       1
Name: count, Length: 5611, dtype: int64
```

[55]:

```
# Checking the TotalCharges field again. There is no missing value

df.TotalCharges.isna().sum()
```

[55]:
```
0
```

[57]:
```python
# Checking the Churn field. Yes indicates churn, No indicates no customer churn


df.Churn.value_counts(dropna=False)
```

[57]:
```
Churn
No     4399
Yes    1587
Name: count, dtype: int64
```

[59]:
```python
# Looking at the dataset info again shows that everything is good


df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5986 entries, 0 to 6049
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        5986 non-null   object
 1   gender            5986 non-null   object
 2   SeniorCitizen     5986 non-null   int64
 3   Partner           5986 non-null   object
 4   Dependents        5986 non-null   object
 5   tenure            5986 non-null   int64
 6   PhoneService      5986 non-null   object
 7   MultipleLines     5986 non-null   object
 8   InternetService   5986 non-null   object
 9   OnlineSecurity    5986 non-null   object
 10  OnlineBackup      5986 non-null   object
 11  DeviceProtection  5986 non-null   object
 12  TechSupport       5986 non-null   object
 13  StreamingTV       5986 non-null   object
 14  StreamingMovies   5986 non-null   object
 15  Contract          5986 non-null   object
 16  PaperlessBilling  5986 non-null   object
 17  PaymentMethod     5986 non-null   object
 18  MonthlyCharges    5986 non-null   float64
 19  TotalCharges      5986 non-null   float64
 20  Churn             5986 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.0+ MB
```

[61]:
```python
# Everything looks alright with the right data types. Save the cleaned dataframe into a file
```

```
df.to_excel("cleaned_telecom_users.xlsx", index=False)
```

1 ANALYSIS

1.0.1 LOAD THE CLEANED DATASET

[148]:
```
tel_df = pd.read_excel('cleaned_telecom_users.xlsx')
tel_df.head()
```

[148]:
```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7010-BRBUU    Male              0     Yes        Yes      72          Yes
1  9688-YGXVR  Female              0      No         No      44          Yes
2  9286-DOJGF  Female              1     Yes         No      38          Yes
3  6994-KERXL    Male              0      No         No       4          Yes
4  2181-UAESM    Male              0      No         No       2          Yes

  MultipleLines InternetService       OnlineSecurity  …  \
0           Yes              No  No internet service  …
1            No     Fiber optic                   No  …
2           Yes     Fiber optic                   No  …
3            No             DSL                   No  …
4            No             DSL                  Yes  …

      DeviceProtection          TechSupport          StreamingTV  \
0  No internet service  No internet service  No internet service
1                  Yes                   No                  Yes
2                   No                   No                   No
3                   No                   No                   No
4                  Yes                   No                   No

       StreamingMovies        Contract PaperlessBilling  \
0  No internet service        Two year               No
1                   No  Month-to-month              Yes
2                   No  Month-to-month              Yes
3                  Yes  Month-to-month              Yes
4                   No  Month-to-month               No

               PaymentMethod MonthlyCharges  TotalCharges  Churn
0    Credit card (automatic)          24.10       1734.65     No
1    Credit card (automatic)          88.15       3973.20     No
2  Bank transfer (automatic)          74.95       2869.85    Yes
3           Electronic check          55.90        238.50     No
4           Electronic check          53.45        119.50     No

[5 rows x 21 columns]
```

1.0.2 Display the dataset information

[68]:
```
tel_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5986 entries, 0 to 5985
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        5986 non-null   object
 1   gender            5986 non-null   object
 2   SeniorCitizen     5986 non-null   int64
 3   Partner           5986 non-null   object
 4   Dependents        5986 non-null   object
 5   tenure            5986 non-null   int64
 6   PhoneService      5986 non-null   object
 7   MultipleLines     5986 non-null   object
 8   InternetService   5986 non-null   object
 9   OnlineSecurity    5986 non-null   object
 10  OnlineBackup      5986 non-null   object
 11  DeviceProtection  5986 non-null   object
 12  TechSupport       5986 non-null   object
 13  StreamingTV       5986 non-null   object
 14  StreamingMovies   5986 non-null   object
 15  Contract          5986 non-null   object
 16  PaperlessBilling  5986 non-null   object
 17  PaymentMethod     5986 non-null   object
 18  MonthlyCharges    5986 non-null   float64
 19  TotalCharges      5986 non-null   float64
 20  Churn             5986 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 982.2+ KB
```

1.0.3 1. Total Male Subscribers (not Senior Citizens)

1.0.4 a. with Phone Service

[71]:
```
total_male_not_seniors_phone_service = tel_df[(tel_df['gender'] == "Male") &␣
  ↪(tel_df['SeniorCitizen'] == 0) & (tel_df['PhoneService'] == "Yes")]
print("Total male subscribers who are not senior citizens and have a phone␣
  ↪service: ", len(total_male_not_seniors_phone_service))
```

```
Total male subscribers who are not senior citizens and have a phone service:
2313
```

1.0.5 b. with Internet Service

[74]:
```
total_male_not_seniors_internet_service = tel_df[(tel_df['gender'] == "Male") &␣
 ↪(tel_df['SeniorCitizen'] == 0) & (tel_df['InternetService'] != "No")]
print("Total male subscribers who are not senior citizens and have an internet␣
 ↪service: ", len(total_male_not_seniors_internet_service))
```

```
Total male subscribers who are not senior citizens and have an internet service:
1928
```

1.0.6 c. with Device Protection

[77]:
```
total_male_not_seniors_device_protection = tel_df[(tel_df['gender'] == "Male")␣
 ↪& (tel_df['SeniorCitizen'] == 0) & (tel_df['DeviceProtection'] == "Yes")]
print("Total male subscribers who are not senior citizens and have a device␣
 ↪protection: ", len(total_male_not_seniors_device_protection))
```

```
Total male subscribers who are not senior citizens and have a device protection:
849
```

1.0.7 d. with Streaming TV

[80]:
```
total_male_not_seniors_streamingTV = tel_df[(tel_df['gender'] == "Male") &␣
 ↪(tel_df['SeniorCitizen'] == 0) & (tel_df['StreamingTV'] == "Yes")]
print("Total male subscribers who are not senior citizens and have a␣
 ↪streamingTV: ", len(total_male_not_seniors_streamingTV))
```

```
Total male subscribers who are not senior citizens and have a streamingTV:  925
```

1.0.8 e. with paperless billing

[83]:
```
total_male_not_seniors_paperless_billing = tel_df[(tel_df['gender'] == "Male")␣
 ↪& (tel_df['SeniorCitizen'] == 0) & (tel_df['PaperlessBilling'] == "Yes")]
print("Total male subscribers who are not senior citizens and have a paperless␣
 ↪Billing: ", len(total_male_not_seniors_paperless_billing))
```

```
Total male subscribers who are not senior citizens and have a paperless Billing:
1408
```

[ ]:
```

```

1.0.9 2. Total Female Subscribers (not Senior Citizens)

1.0.10 a. with Phone Serice

[87]:
```
total_female_not_seniors_phone_service = tel_df[(tel_df['gender'] == "Female")␣
 ↪& (tel_df['SeniorCitizen'] == 0) & (tel_df['PhoneService'] == "Yes")]
print("Total female subscribers who are not senior citizens and have a phone␣
 ↪service: ", len(total_female_not_seniors_phone_service))
```

```
Total female subscribers who are not senior citizens and have a phone service:
2206
```

1.0.11 b. with Internet Service

[90]:

```
total_female_not_seniors_internet_service = tel_df[(tel_df['gender'] ==␣
 ↪"Female") & (tel_df['SeniorCitizen'] == 0) & (tel_df['InternetService'] !=␣
 ↪"No")]
print("Total female subscribers who are not senior citizens and have an␣
 ↪internet service: ", len(total_female_not_seniors_internet_service))
```

```
Total female subscribers who are not senior citizens and have an internet
service:  1842
```

1.0.12 c. with Device Protection

[93]:

```
total_female_not_seniors_device_protection = tel_df[(tel_df['gender'] ==␣
 ↪"Female") & (tel_df['SeniorCitizen'] == 0) & (tel_df['DeviceProtection'] ==␣
 ↪"Yes")]
print("Total female subscribers who are not senior citizens and have a device␣
 ↪protection: ", len(total_female_not_seniors_device_protection))
```

```
Total female subscribers who are not senior citizens and have a device
protection:  810
```

1.0.13 d. with Streaming TV

[96]:

```
total_female_not_seniors_streamingTV = tel_df[(tel_df['gender'] == "Female") &␣
 ↪(tel_df['SeniorCitizen'] == 0) & (tel_df['StreamingTV'] == "Yes")]
print("Total female subscribers who are not senior citizens and have a␣
 ↪streamingTV: ", len(total_female_not_seniors_streamingTV))
```

```
Total female subscribers who are not senior citizens and have a streamingTV:
899
```

1.0.14 e. with paperless billing

[99]:

```
total_female_not_seniors_paperless_billing = tel_df[(tel_df['gender'] ==␣
 ↪"Female") & (tel_df['SeniorCitizen'] == 0) & (tel_df['PaperlessBilling'] ==␣
 ↪"Yes")]
print("Total female subscribers who are not senior citizens and have a␣
 ↪paperless Billing: ", len(total_female_not_seniors_paperless_billing))
```

```
Total female subscribers who are not senior citizens and have a paperless
Billing:  1382
```

[ ]:

```

```

1.0.15 Additional Summaries/Analysis

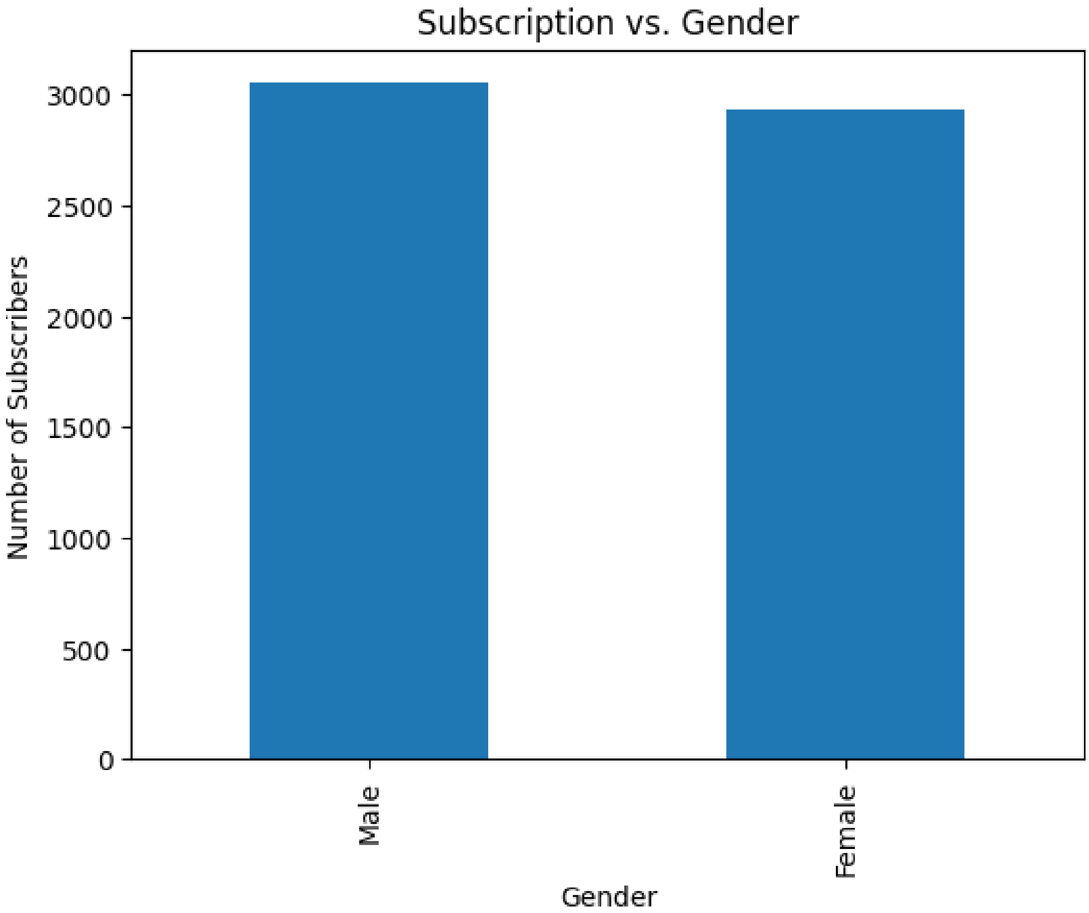
1.0.16 1. Find the number of subscriptions based on gender

[104]:

```
import matplotlib.pyplot as plt


tel_df['gender'].value_counts().plot(kind='bar')
plt.title('Subscription vs. Gender')
plt.ylabel('Number of Subscribers')
plt.xlabel('Gender')
```

[104]:

```
Text(0.5, 0, 'Gender')
```



This graphs shows that there are more men subscribers than there are of female subscibers. There are over three thousand male subscribers and close to three thousand female subscribers. But the difference is not that much.

[ ]:

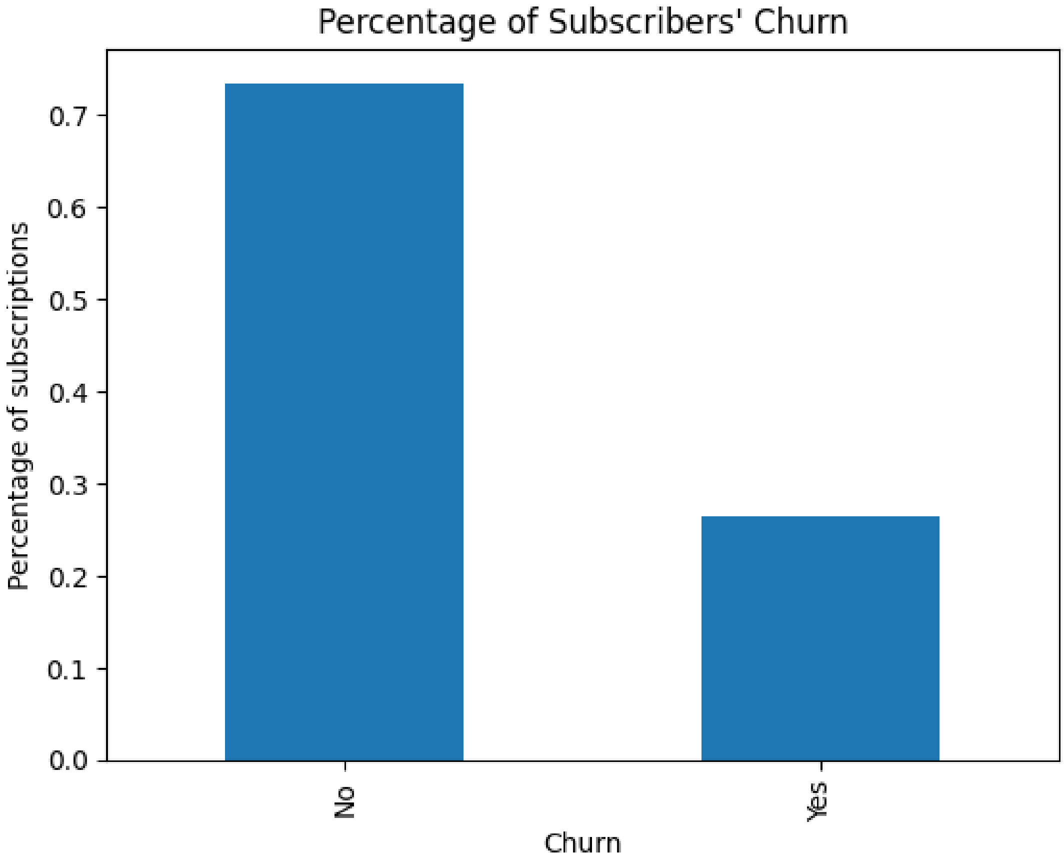
```

```

1.0.17 2. What percentage of subscribers are churning?

[109]:

```
print(tel_df.groupby('Churn')['customerID'].count() / tel_df.shape[0])

(tel_df.groupby('Churn')['customerID'].count() / tel_df.shape[0]).
 ↪plot(kind='bar')
plt.title("Percentage of Subscribers' Churn")
plt.xlabel('Churn')
plt.ylabel('Percentage of subscriptions')
plt.show()
```

```
Churn
No     0.734881
Yes    0.265119
Name: customerID, dtype: float64
```



The graph shows about 73% of the subscribers are staying and about 26% are churning. The business should try to reduce this number.

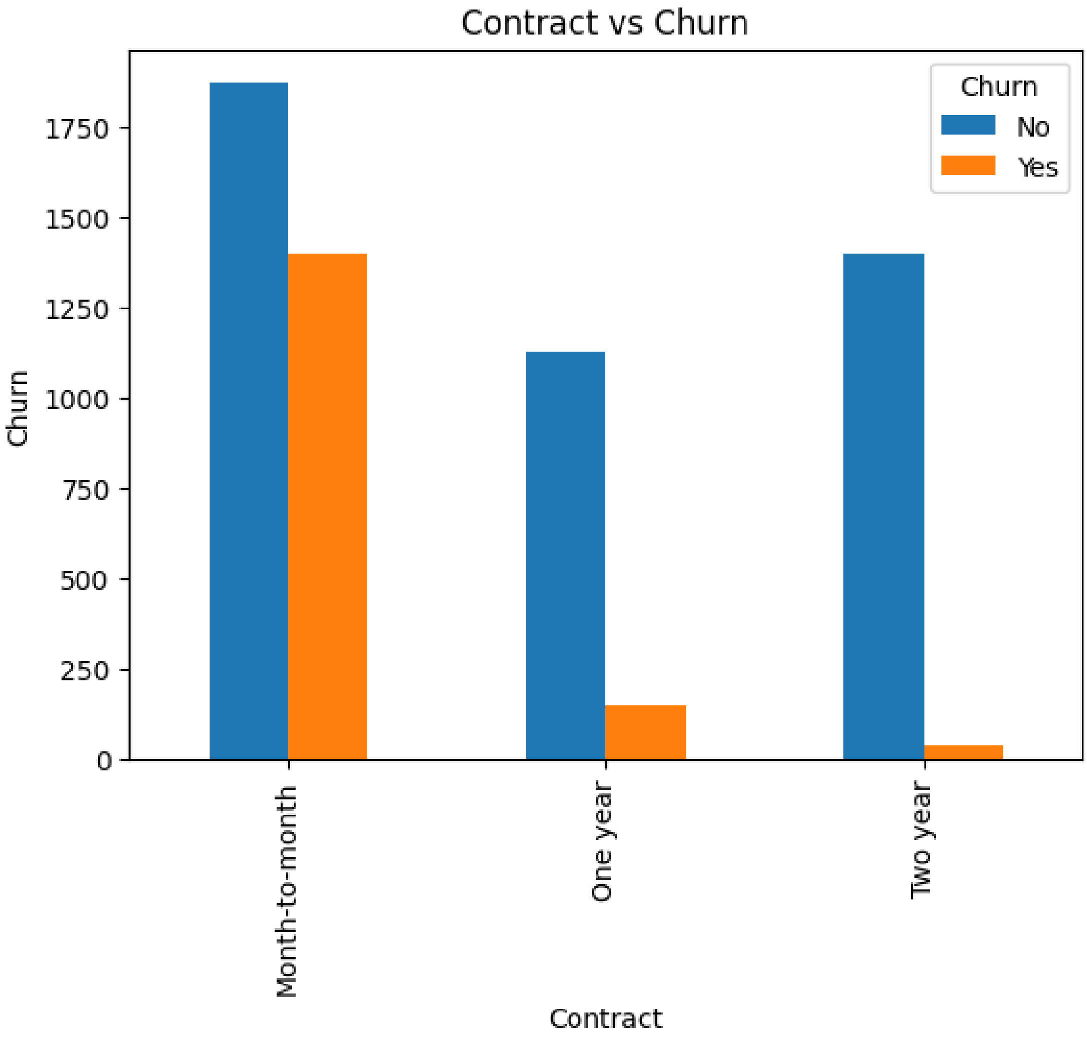
[ ]:

```
```

1.0.18 3. Contract Type vs. Churn. Does the contract type impact so much the churn?

[114]:
```
print(tel_df.groupby('Contract')['Churn'].value_counts().unstack())

(tel_df.groupby('Contract')['Churn'].value_counts().unstack()).plot(kind='bar')
plt.title("Contract vs Churn")
plt.xlabel('Contract')
plt.ylabel('Churn')
plt.show()
```

```
Churn             No   Yes
Contract
Month-to-month  1871  1398
One year        1128   148
Two year        1400    41
```



The graph shows that in the three contracts, we still have more remaining than the number leaving. But there should be a worry in the monthly contract, there are high numbers of subscribers who are churning. As the duration of the contract increases, less and less customers are churning. The same goes with one-year and two-year contracts, the more the year, the more the number that remain.

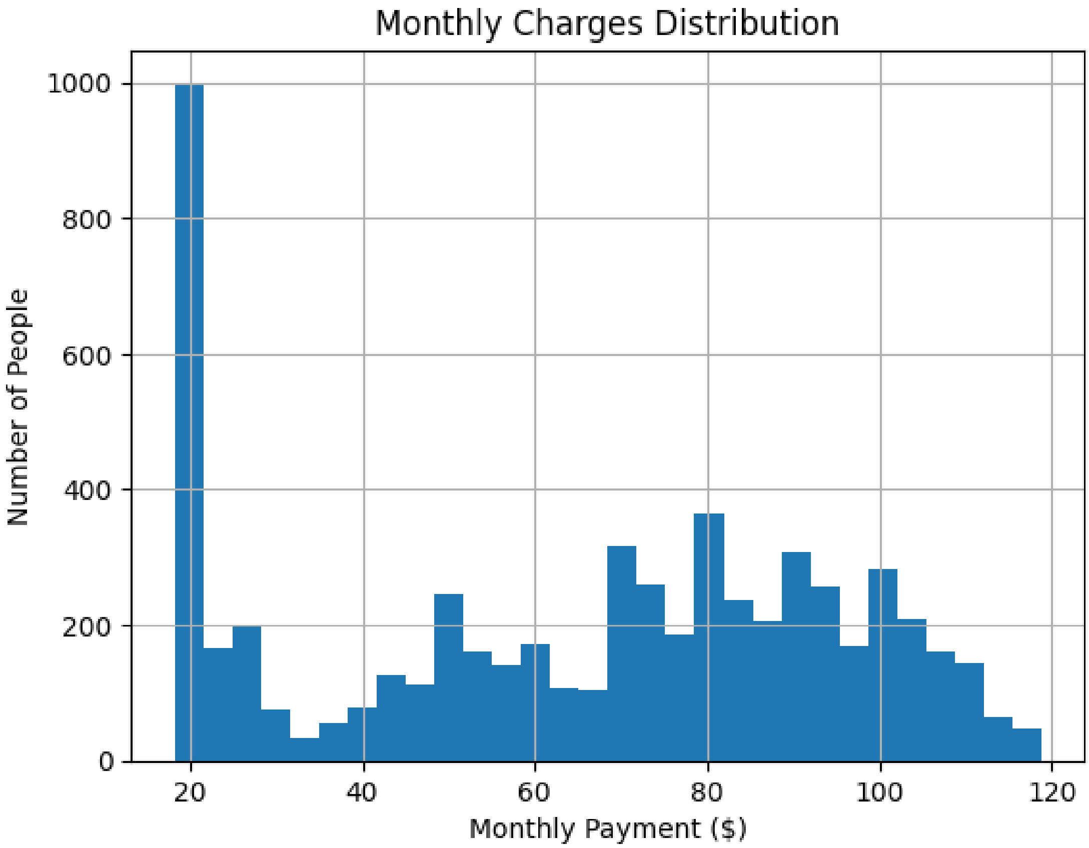
[ ]:

```
```

1.0.19 4. Monthly Charges Distribution. What are the monthly charges payment segments?

[157]:

```
tel_df['MonthlyCharges'].hist(bins=30)
plt.title("Monthly Charges Distribution")
plt.xlabel('Monthly Payment ($)')
plt.ylabel('Number of People')
plt.show()
```



The graph is that of a multi-modal histogram, telling that the distribution is multi-segmented. In terms of monthly charges or payments, the customers can be segmented

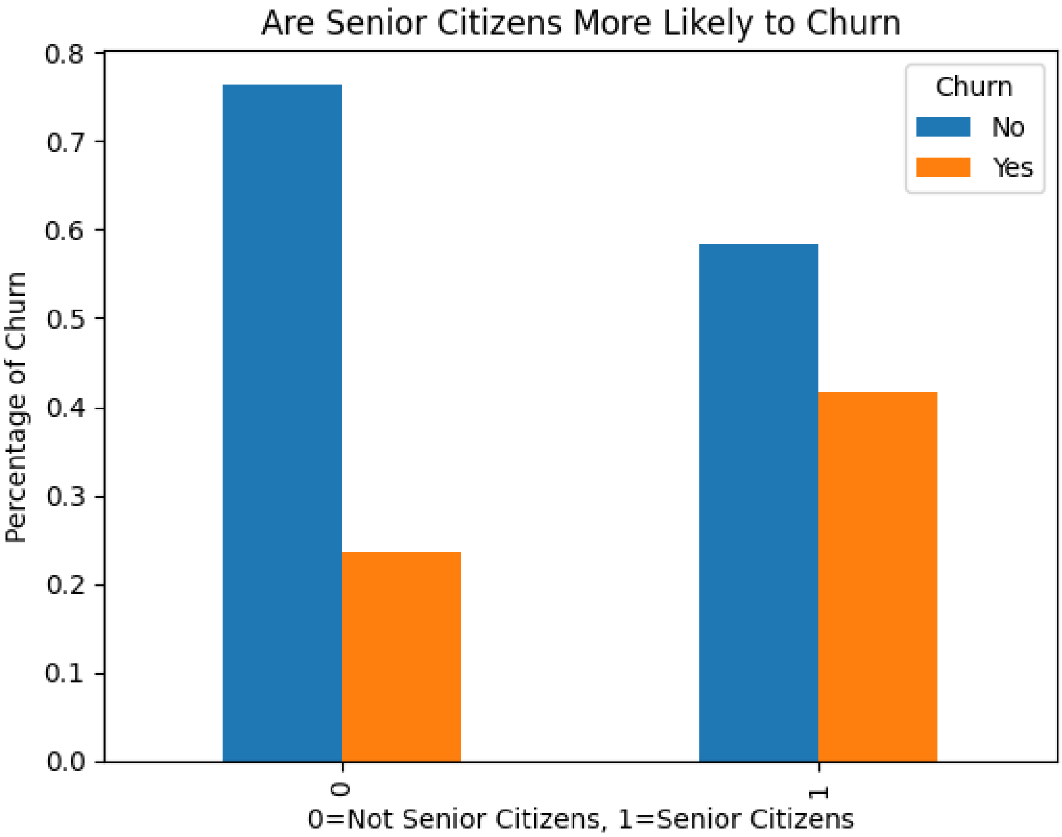
into three payment segments. The first segment has a range 18 eur - 32 eur, a mode at 20 eur at 1000 people mark. The second payment segment has a range of 32 eur - 70 eur, a mode of 50 eur peaking at about 250 people mark. The third segment has a range 70 eur - 118 eur, a mode at 80 eur at 380 people mark. This is the interpretation. The first segment represents basic customers who pay for basic services like phone only, no internet. Here you have more people in this segment (app. 1000). The second segment represents customers with basic internet plans (DSL). And the third segment is premium users, fiber optics and streaming and other services.

```
[ ]:
```

1.0.20 5. Are Senior citizens more likely to churn?

```
[180]: print(pd.crosstab(tel_df['SeniorCitizen'], tel_df['Churn'], normalize='index'))

       (pd.crosstab(tel_df['SeniorCitizen'], tel_df['Churn'], normalize='index')).
        ↪plot(kind='bar')
       plt.title("Are Senior Citizens More Likely to Churn")
       plt.xlabel('0=Not Senior Citizens, 1=Senior Citizens')
       plt.ylabel('Percentage of Churn')
       plt.show()
```

```
Churn                No       Yes
SeniorCitizen
0              0.763944  0.236056
1              0.583851  0.416149
```



From the graph, we can see that the pink bar (Yes) for 1 (Senior Citizen) is higher than the one for 0 (Not Senior Citizens). This confirms that senior citizens are more likely to churn. So, 41.6% of the senior citizens against 23.6% of younger citizens.

[ ]:

```

```

1.0.21 6. What is the preferable (most popular) method of payment?

[201]:

```
print(tel_df.groupby(['PaymentMethod'])['customerID'].count())
```

```
PaymentMethod
Bank transfer (automatic)    1308
Credit card (automatic)      1303
Electronic check             2006
Mailed check                 1369
Name: customerID, dtype: int64
```

Electronic check is the most popular method of payment with 2006, followed by Mailed check with 1369.

[ ]:

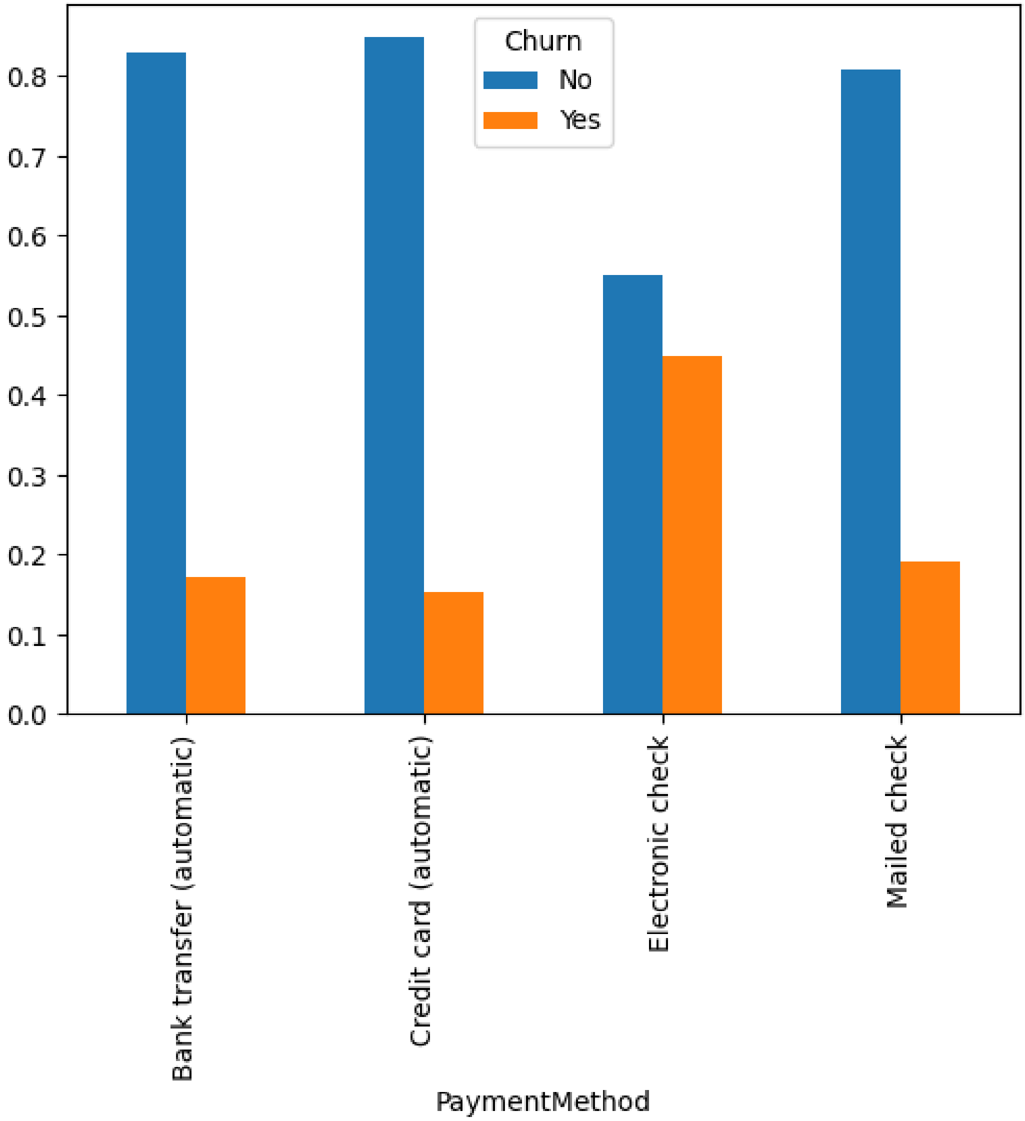
```

```

1.0.22 7. What method of payment is the most linked to churn?

```
[208]: print(pd.crosstab(tel_df['PaymentMethod'], tel_df['Churn'], normalize='index'))
       (pd.crosstab(tel_df['PaymentMethod'], tel_df['Churn'], normalize='index')).
        ↪plot(kind='bar')
```

```
Churn                            No       Yes
PaymentMethod
Bank transfer (automatic)  0.828746  0.171254
Credit card (automatic)    0.848043  0.151957
Electronic check           0.550349  0.449651
Mailed check               0.807889  0.192111
```

```
[208]: <Axes: xlabel='PaymentMethod'>
```



Even though Electronic check is the most popular method of payment, from the graph it is the most linked to churn, 44.9%, followed by Mailed check, 19.2%.

```
[ ]:
```

1.0.23 8. How likely do senior citizens use phone and internet?

```
[217]: print(pd.crosstab(tel_df['SeniorCitizen'], tel_df['PhoneService']))
       print()
       pd.crosstab(tel_df['SeniorCitizen'], tel_df['InternetService'])
```

```
PhoneService     No   Yes
SeniorCitizen
0               501  4519
1                89   877
```

```
[217]: InternetService   DSL  Fiber optic    No
       SeniorCitizen
       0                1847         1923  1250
       1                 221          704    41
```

```
[219]: ##### Senior citizens use phone service but not so much (877), DSL (221), and␣
         ↪Fiber optic (704). They use more of phone service and fiber optic.
```

```
[ ]:
```

1.0.24 9. Do Monthly Charges, TotalCharges, and tenure correlate?

```
[223]: tel_df[['MonthlyCharges', 'TotalCharges', 'tenure']].corr()
```

```
[223]:                 MonthlyCharges  TotalCharges    tenure
       MonthlyCharges        1.000000      0.656762  0.256983
       TotalCharges          0.656762      1.000000  0.827756
       tenure                0.256983      0.827756  1.000000
```

Monthly charges and Total charges correlate positively, (0.65), Monthly charges and tenure correlate positively but weakly (0.25), but total charges and tenure are positively and strongly correlated (0.82)

```
[ ]:
```

2 CONCLUSIONS

- 73% of subscribers stay while 26% of subscribers churn. This is a big number, the management should pay attention to reasons they are leaving.

- Subscriber who churn are from the month-to-month contact. The management should try to incentify this contract category. For that subscribers do not leave but continue to longer contracts.
- More senior citizens are likely to leave, 41.6% of senior citizens will leave. The management should target this group with more incentives.
- Electonic check and mailed check are the most popular payment among subscribers, but they are also linked to high churn.

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```