we will make analysis on each column

> Important Notes :

1. this NoteBook is explainig every step by comments and Markdowns in each cell
2. please do not jump below , read this Notebook cell by cell to understand what i am doing
3. read my comments carefully,i am using some tricks (every thing is explained with comments)
4. finally after reading all Notebook , you will be satisfied isa

In [108]:
```
#EDA Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

take over view on our data

In [213]:
```
df = pd.read_csv("customer .csv", sep = "," , encoding = "utf-8")
```

In [214]:
```
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecuri
0	7590-VHVEG	Female	NaN	Yes	No	NaN	No	No phone service	DSL	No
1	5575-GNVDE	Male	NaN	No	No	34.0	Yes	No	DSL	Yes
2	3668-QPYBK	Male	NaN	No	No	2.0	Yes	No	DSL	Yes
3	7795-CFOCW	Male	NaN	No	No	45.0	No	No phone service	DSL	Yes
4	9237-HQITU	Female	NaN	No	No	2.0	Yes	No	Fiber optic	No

5 rows × 21 columns

In [111]:
```
df.tail()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSec
7038	6840-RESVB	Male	0.0	Yes	Yes	24.0	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0.0	Yes	Yes	72.0	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0.0	Yes	Yes	11.0	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1.0	Yes	No	4.0	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0.0	No	No	66.0	Yes	No	Fiber optic	Yes

5 rows × 21 columns

In [112]:
```
df.columns
```

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

In [180]:
```
df.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     7000 non-null   float64
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            6896 non-null   float64
 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   float64
 20  Churn             7043 non-null   object 
dtypes: float64(4), object(17)
memory usage: 1.1+ MB
```

In [114]:
```
df.describe()
#total charge biased  to max
```

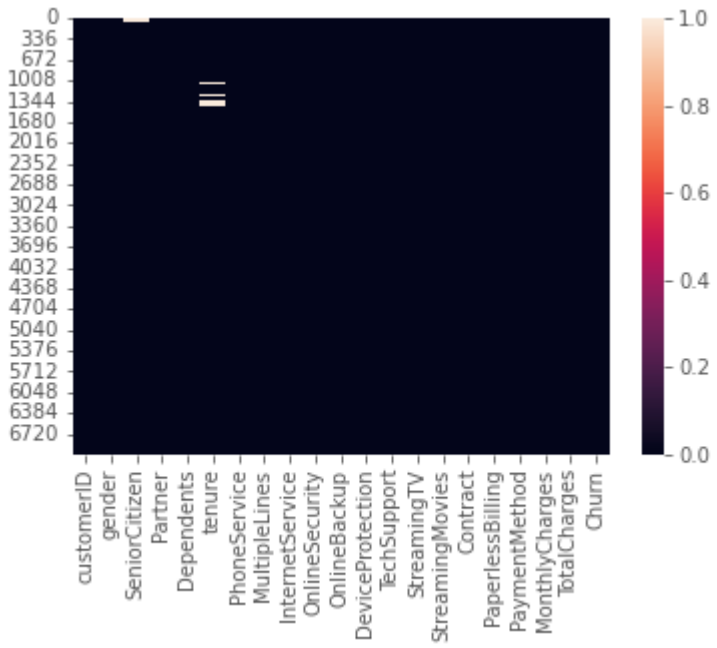
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7000.000000	6896.000000	7043.000000	7043.000000
mean	0.163143	33.041473	64.761692	2283.300440
std	0.369522	24.382260	30.090047	2265.000258
min	0.000000	1.000000	18.250000	18.800000
25%	0.000000	10.000000	35.500000	402.225000
50%	0.000000	30.000000	70.350000	1400.550000
75%	0.000000	56.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

In [115]:
```
df.isnull().sum()
```

```
customerID           0
gender               0
SeniorCitizen       43
Partner              0
Dependents           0
tenure             147
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
```

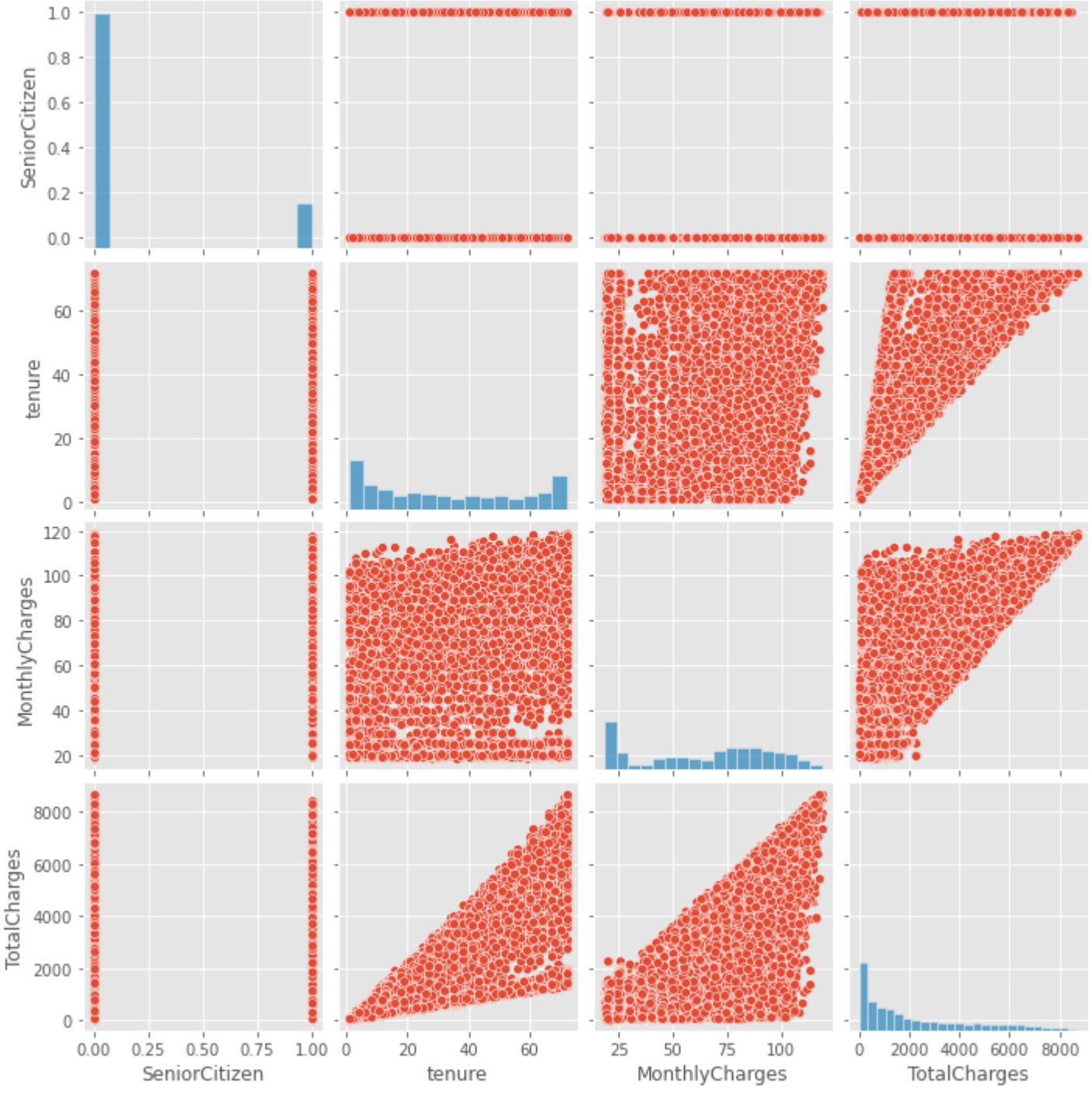
In [116]:
```
sns.heatmap(df.isnull())
```

```
<AxesSubplot:>
```



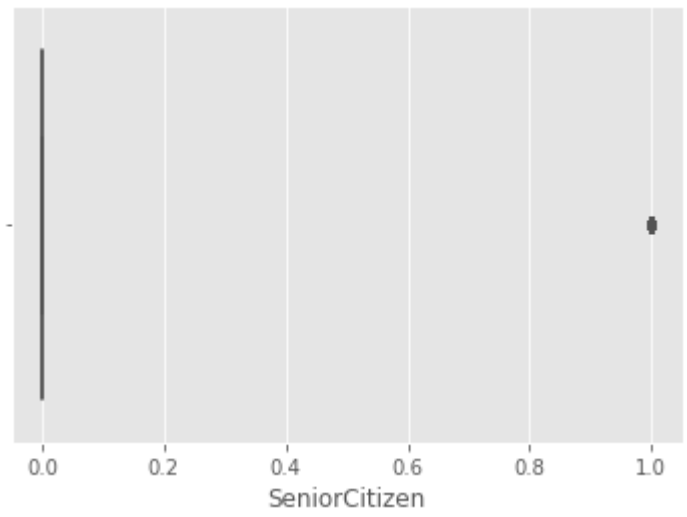
In [117]:
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x269c3c25f70>
```

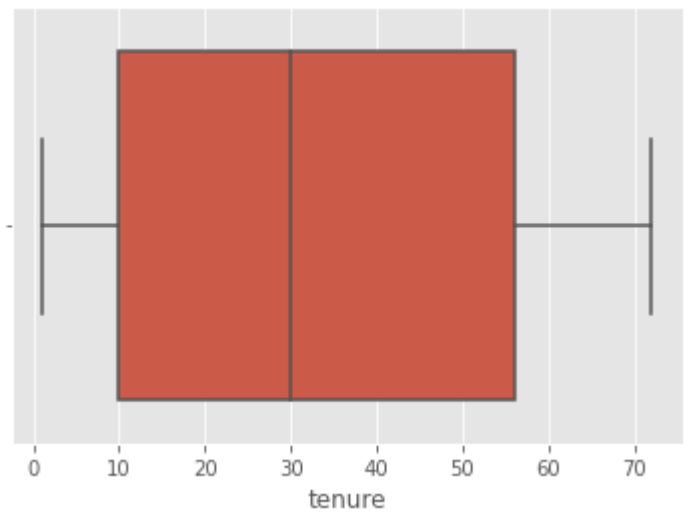


In [118]:
```
for i , coltype in df.dtypes.iteritems():
    if coltype != object:
        print(sns.boxplot(x=df[i]))
        plt.show()


# no outliers
```

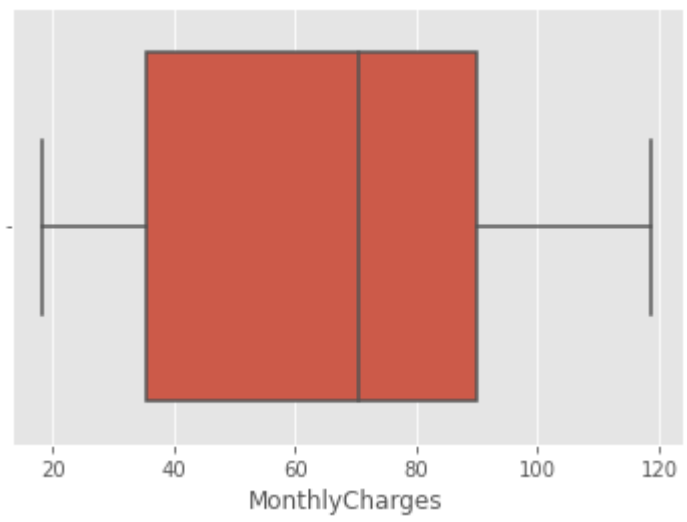
```
AxesSubplot(0.125,0.125;0.775x0.755)
```



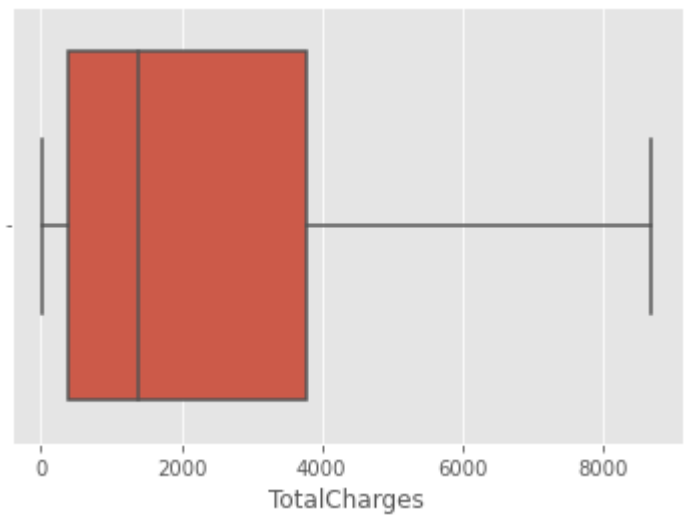

```
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
AxesSubplot(0.125,0.125;0.775x0.755)
```




```
AxesSubplot(0.125,0.125;0.775x0.755)
```

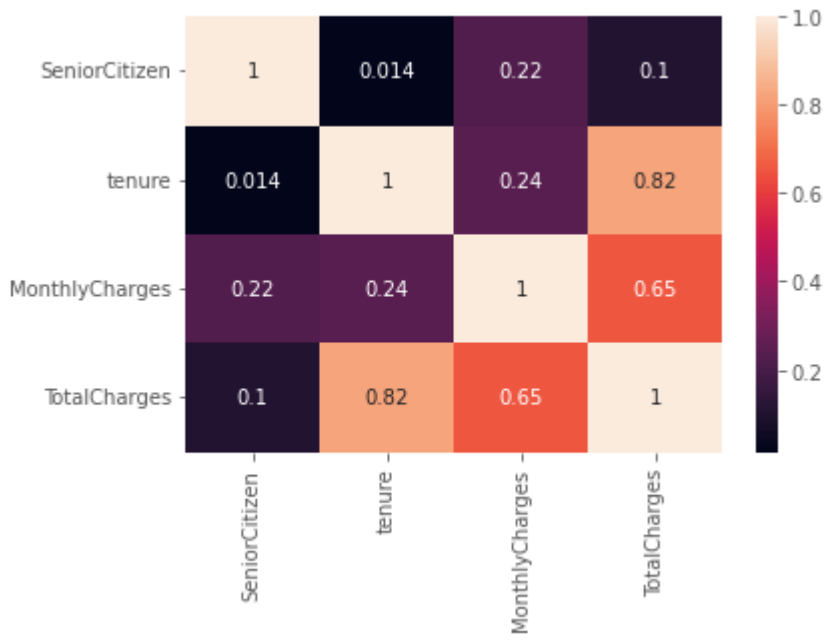


In [119]:
```
df.corr()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
SeniorCitizen	1.000000	0.013521	0.221101	0.102831
tenure	0.013521	1.000000	0.238635	0.822171
MonthlyCharges	0.221101	0.238635	1.000000	0.650468
TotalCharges	0.102831	0.822171	0.650468	1.000000

In [120]:
```
sns.heatmap(df.corr(), annot = True)
#we have correlation between (MonthlyCharges  and TotalCharges ) & (TotalCharges and tenure )
```

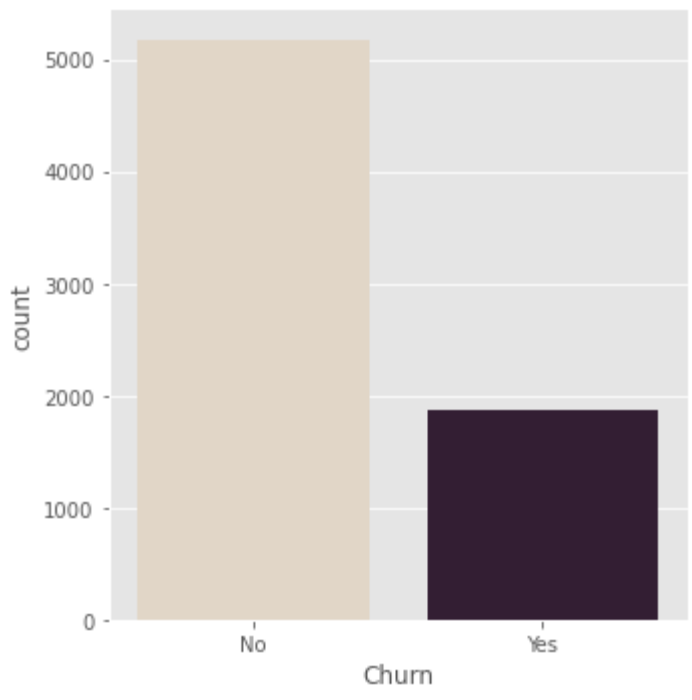
```
<AxesSubplot:>
```



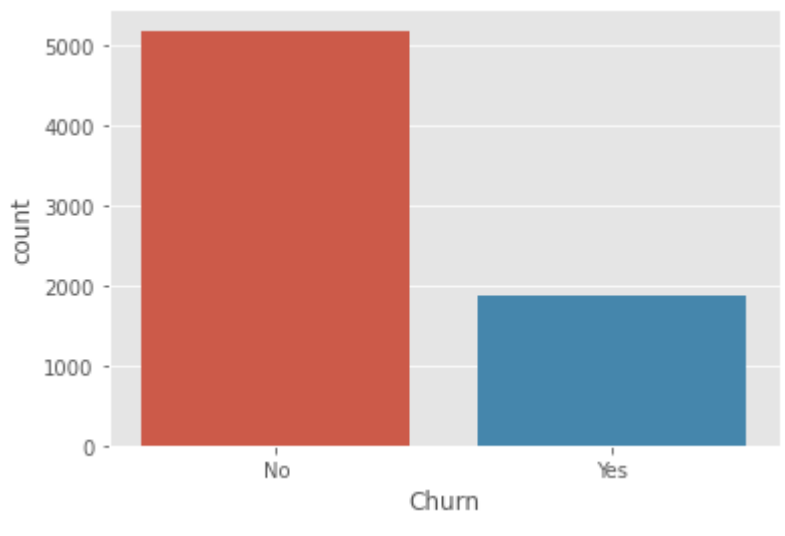
analysis on churn

In [121]:
```
sns.catplot(x="Churn", kind="count", palette="ch:.25", data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x269c7301520>
```



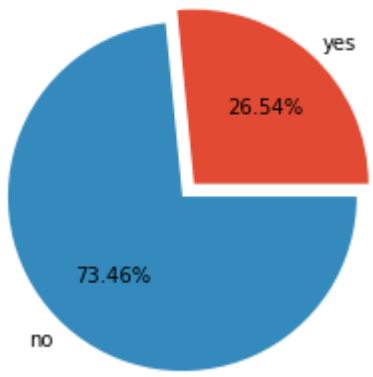
In [122]:
```
ax = sns.countplot(x="Churn", data=df)
```



In [123]:
```
df['Churn'].value_counts()
```

```
No     5174
Yes    1869
Name: Churn, dtype: int64
```

In [124]:
```
plt.style.use('ggplot')
data  = [1869,5174]
la = ["yes", "no"]
ex= [0.1,0]
plt.pie(data ,labels= la , explode = ex , autopct="%1.2f%%" )
plt.show()
```



we have 1869 customers churn

In [125]:
```
df_1 = df.groupby(["Churn" ,  "PaymentMethod"]).sum()
df_1.style.background_gradient(cmap = "PuBu")
```

Churn	PaymentMethod	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
No	Bank transfer (automatic)	180.000000	60618.000000	83653.550000	4167234.750000
	Credit card (automatic)	159.000000	59469.000000	83285.250000	4128616.850000
	Electronic check	277.000000	41566.000000	96056.250000	3377326.850000
	Mailed check	50.000000	32678.000000	53990.700000	1545179.650000
Yes	Bank transfer (automatic)	53.000000	6779.000000	20091.900000	585611.750000
	Credit card (automatic)	62.000000	6383.000000	17946.600000	545259.800000
	Electronic check	317.000000	17909.000000	84288.750000	1567576.400000
	Mailed check	44.000000	2452.000000	16803.600000	164478.950000

In [126]:
```
df
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSec
0	7590-VHVEG	Female	NaN	Yes	No	NaN	No	No phone service	DSL	No
1	5575-GNVDE	Male	NaN	No	No	34.0	Yes	No	DSL	Yes
2	3668-QPYBK	Male	NaN	No	No	2.0	Yes	No	DSL	Yes
3	7795-CFOCW	Male	NaN	No	No	45.0	No	No phone service	DSL	Yes
4	9237-HQITU	Female	NaN	No	No	2.0	Yes	No	Fiber optic	No
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0.0	Yes	Yes	24.0	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0.0	Yes	Yes	72.0	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0.0	Yes	Yes	11.0	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1.0	Yes	No	4.0	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0.0	No	No	66.0	Yes	No	Fiber optic	Yes

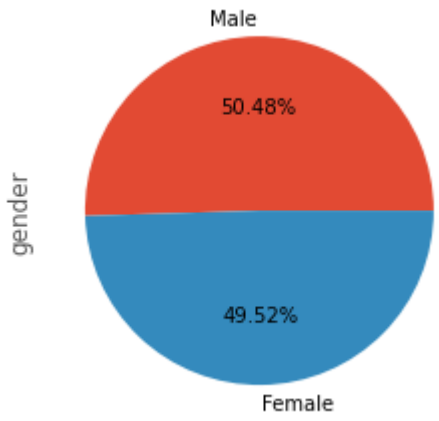
```
7043 rows × 21 columns
```

analysis on gender

In [127]:
```
#count


df["gender"].value_counts()
#almost the same ratio >> good
```

```
Male      3555
Female    3488
Name: gender, dtype: int64
```

In [128]:
```
round(df['gender'].value_counts()/df.shape[0]*100,2).plot.pie(autopct='%1.2f%%');
```



In [129]:
```
#relation between gender and churn
df.groupby([ 'gender', 'Churn']).sum()
```

		SeniorCitizen	tenure	MonthlyCharges	TotalCharges
gender	**Churn**				
Female	**No**	328.0	96479.0	157183.85	6610690.95
	Yes	240.0	15935.0	70248.55	1353079.75
Male	**No**	338.0	97852.0	159801.90	6607667.15
	Yes	236.0	17588.0	68882.30	1509847.15

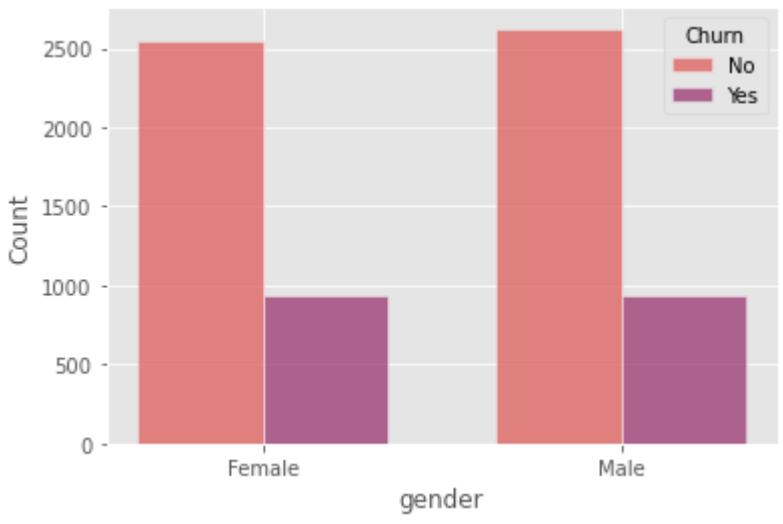
In [130]:
```
df.groupby([ 'gender']).sum()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
gender				
Female	568.0	112414.0	227432.4	7963770.7
Male	574.0	115440.0	228684.2	8117514.3

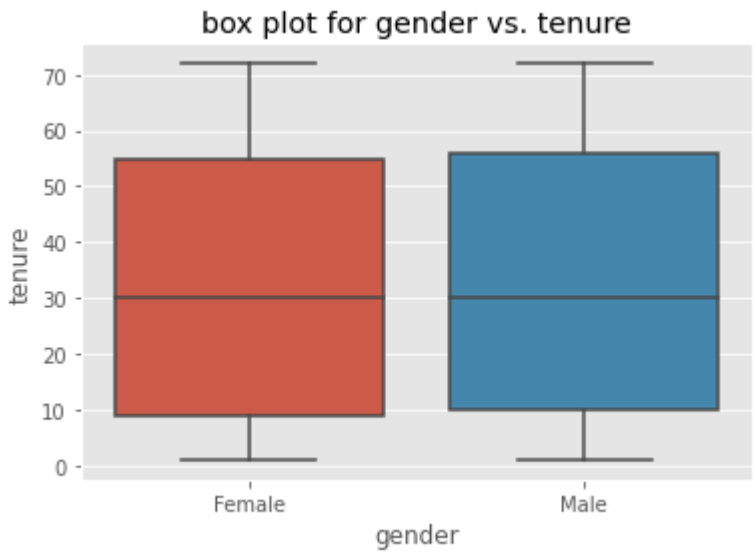
In [131]:
```
df.groupby([ 'gender', 'Churn']).MonthlyCharges.count()
```

```
gender  Churn
Female  No       2549
        Yes       939
Male    No       2625
        Yes       930
Name: MonthlyCharges, dtype: int64
```

In [132]:
```
sns.histplot(data=df, x="gender", hue="Churn", multiple="dodge", palette ='flare',shrink=.7);


# almost equal in churn
```



In [133]:
```
# show the boxplot of each gender to the tenure(months of services)
sns.boxplot(x='gender', y='tenure', data=df)
plt.title('box plot for gender vs. tenure');  # almost the same ratio >> good
```



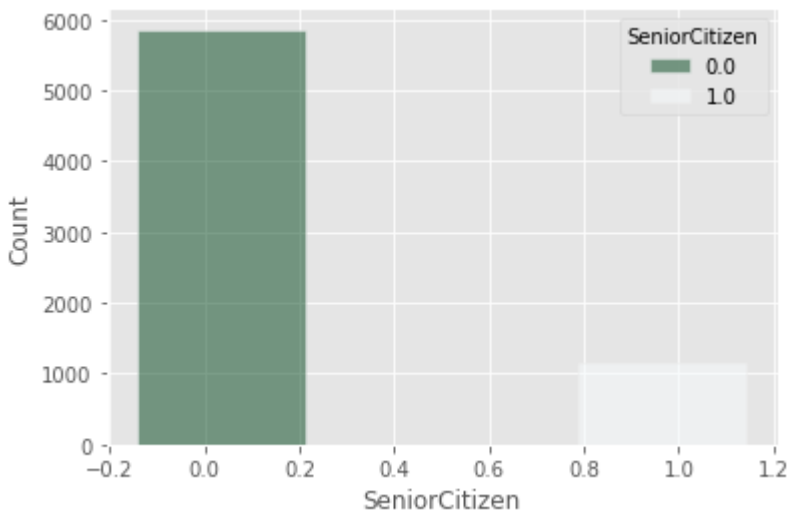
finally gender not important to us and not has effective on our data (not important feature)

analysis on SeniorCitizen

In [134]:
```
df["SeniorCitizen"].value_counts()
```

```
0.0    5858
1.0    1142
Name: SeniorCitizen, dtype: int64
```

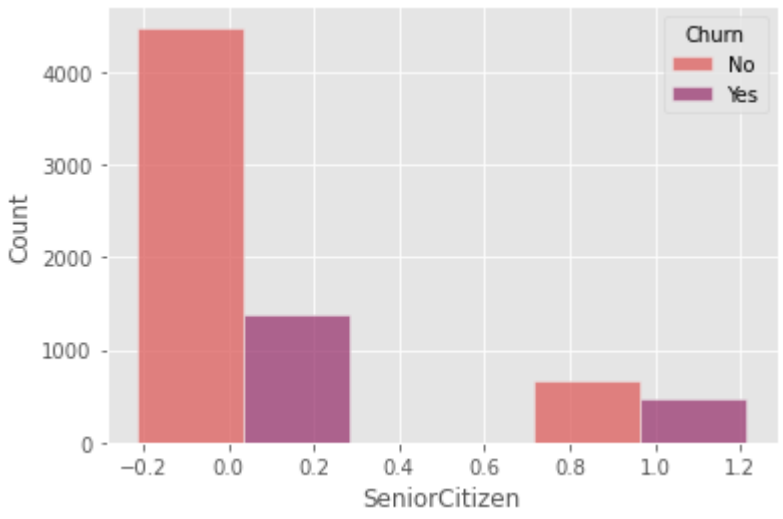
In [135]:
```
sns.histplot(x= "SeniorCitizen", data=df , shrink=5, palette ='BuGn_r',hue = "SeniorCitizen");
#most of our customers less than 65
```



In [136]:
```
df.groupby([ 'SeniorCitizen', 'Churn']).MonthlyCharges.count()
```

```
SeniorCitizen  Churn
0.0            No       4478
               Yes      1380
1.0            No        666
               Yes       476
Name: MonthlyCharges, dtype: int64
```

In [137]:
```
sns.histplot(data=df, x="SeniorCitizen", hue="Churn", multiple="dodge", palette ='flare',shrink=7);
```

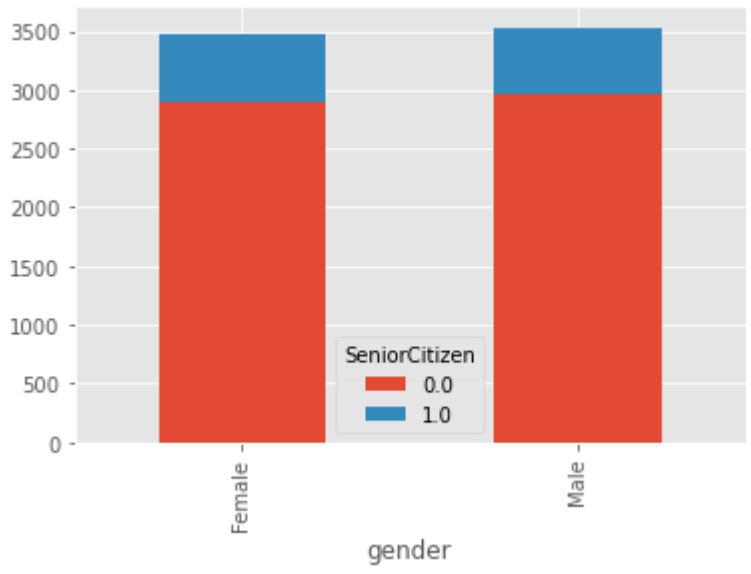


customer over 65 go out relative to their number

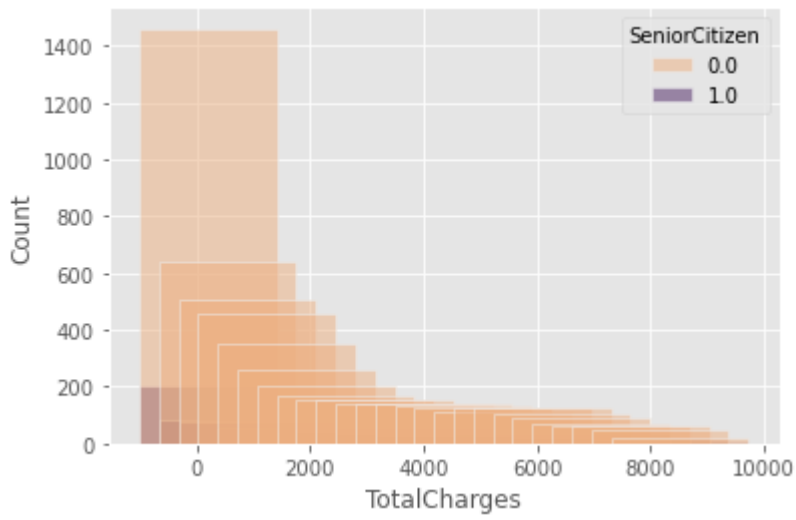
In [138]:
```
df.groupby(['SeniorCitizen']).sum()
```

SeniorCitizen	tenure	MonthlyCharges	TotalCharges
0.0	188546.0	362100.85	12771349.35
1.0	38007.0	91154.85	3209551.25

In [139]:
```
## seniorCitizen (0 , 1) for each gender
gender_seniorCitizen = df.groupby(['gender', 'SeniorCitizen']).size().unstack()

gender_seniorCitizen.plot(stacked=True, kind='bar');
#almoust equel
```



In [140]:
```
sns.histplot(data=df, x="TotalCharges", hue="SeniorCitizen", palette ='flare',shrink=7);
```



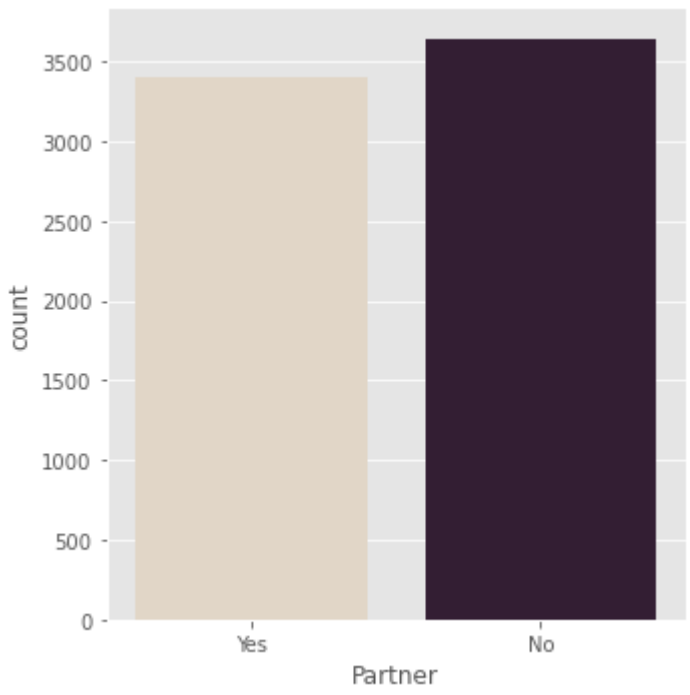
analysis on Partnership

In [141]:
```
df["Partner"].value_counts()
```

```
No     3641
Yes    3402
Name: Partner, dtype: int64
```

In [142]:
```
sns.catplot(x="Partner", kind="count", palette="ch:.25", data=df)
```

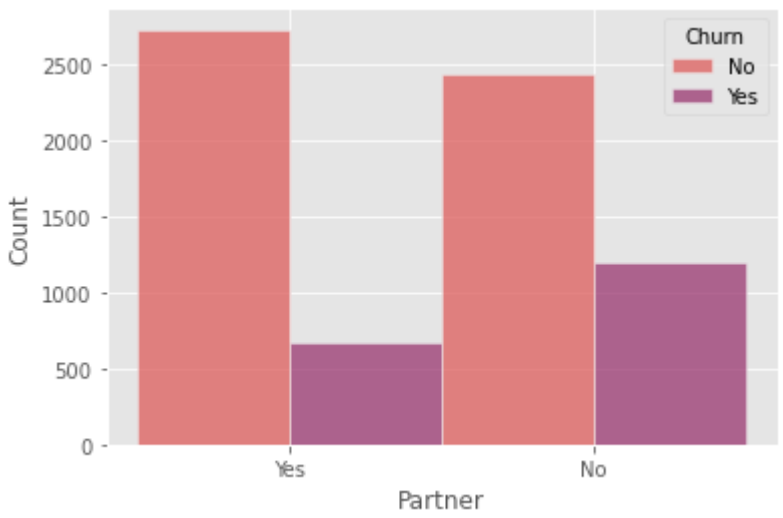
```
<seaborn.axisgrid.FacetGrid at 0x269c880e730>
```



In [143]:
```
df.groupby([ 'Partner', 'Churn']).MonthlyCharges.count()
```

```
Partner  Churn
No       No       2441
         Yes      1200
Yes      No       2733
         Yes       669
Name: MonthlyCharges, dtype: int64
```

In [144]:
```
sns.histplot(data=df, x="Partner", hue="Churn", multiple="dodge", palette ='flare');


#no partner hight churn
```



no partner hight churn

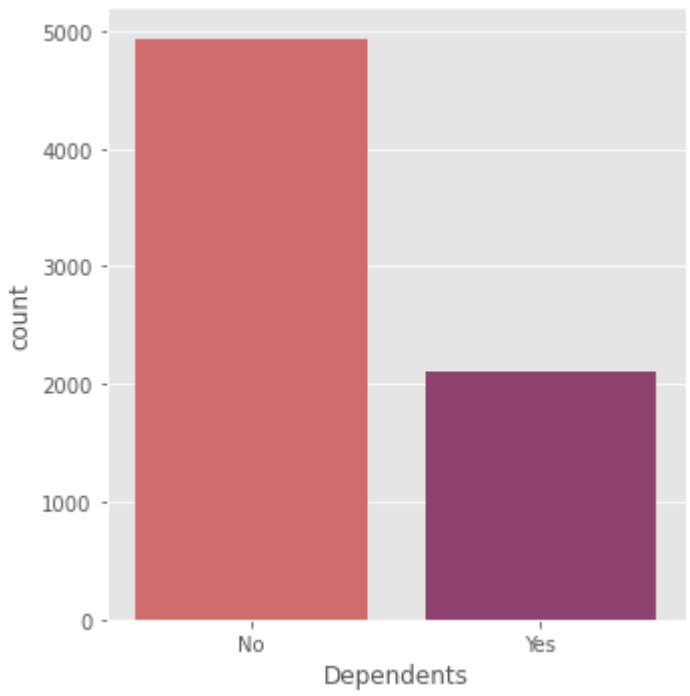
analysis on Dependents

In [145]:
```
df["Dependents"].value_counts()
```

```
No     4933
Yes    2110
Name: Dependents, dtype: int64
```

In [146]:
```
sns.catplot(x="Dependents", kind="count", palette="flare", data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x269c8b64550>
```

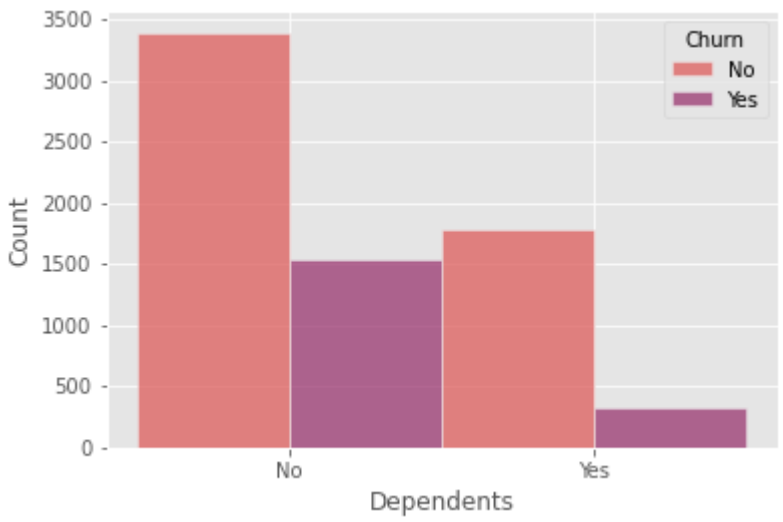


In [147]:
```
df.groupby([ 'Dependents', 'Churn']).MonthlyCharges.count()
```

```
Dependents  Churn
No          No       3390
            Yes      1543
Yes         No       1784
            Yes       326
Name: MonthlyCharges, dtype: int64
```

In [148]:
```
sns.histplot(data=df, x="Dependents", hue="Churn", multiple="dodge", palette ='flare');
```

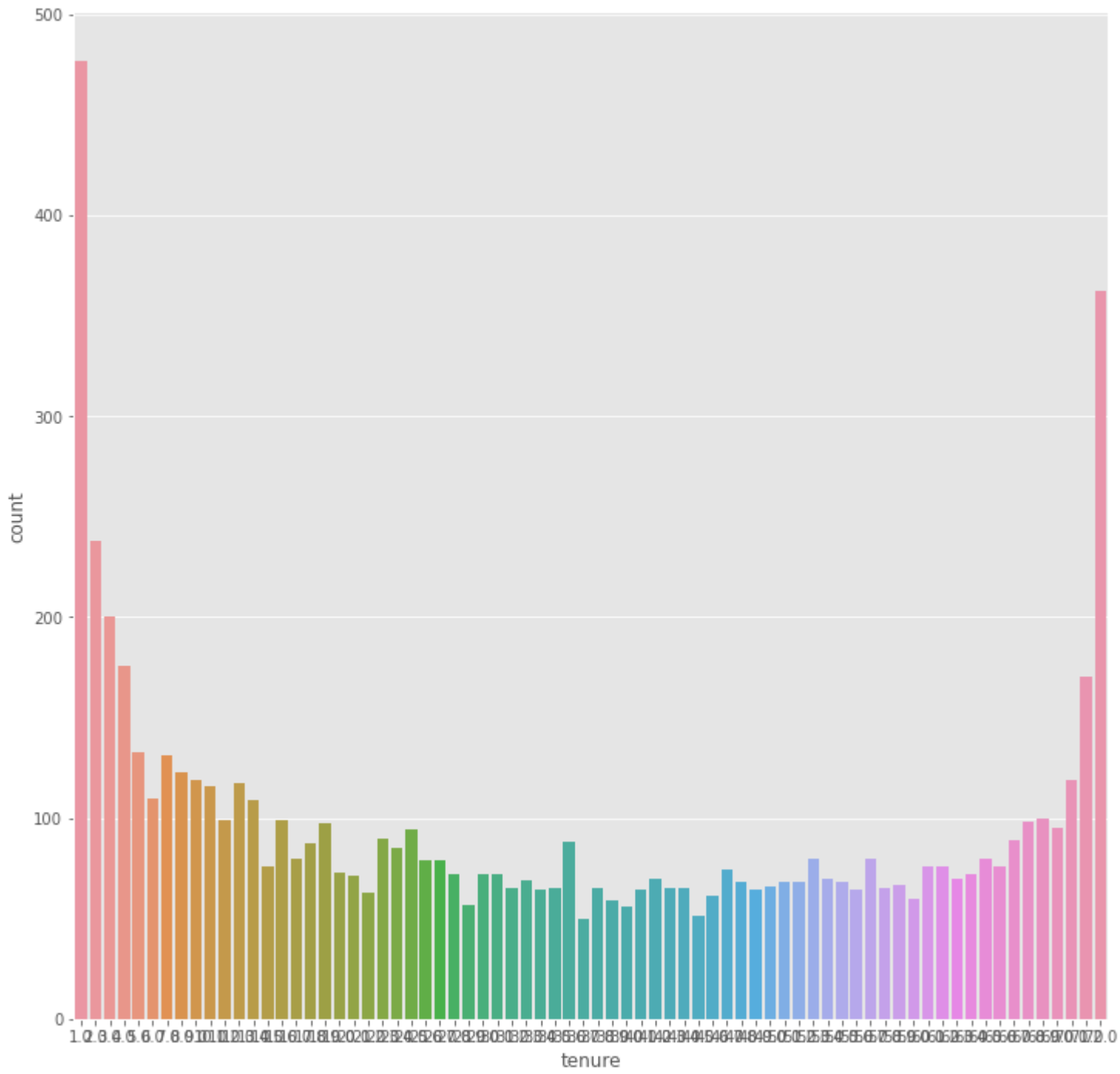
##no dependents hight churn



	##no dependents hight churn
	analysis on tenure

In [149]:
```
plt.figure(figsize=(12,12))
sns.countplot(x="tenure", data=df)
```

```
<AxesSubplot:xlabel='tenure', ylabel='count'>
```

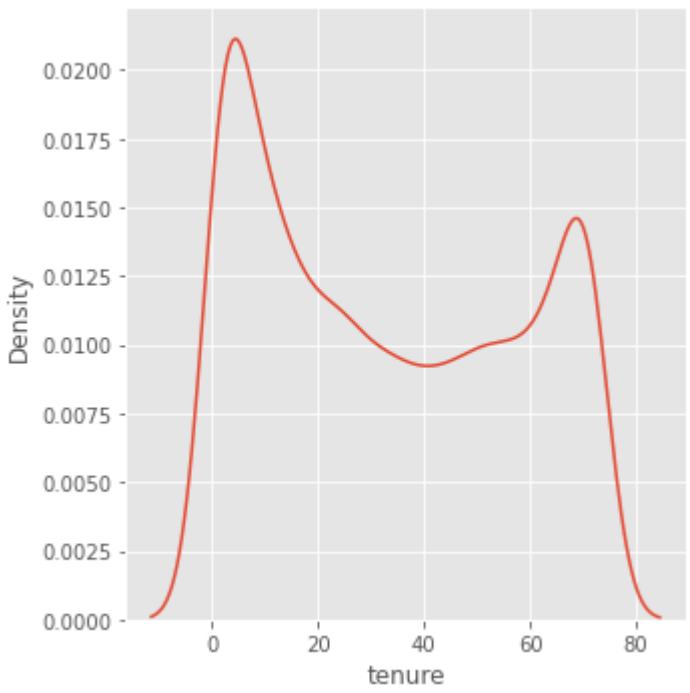


In [150]:
```
df["tenure"].value_counts()
```

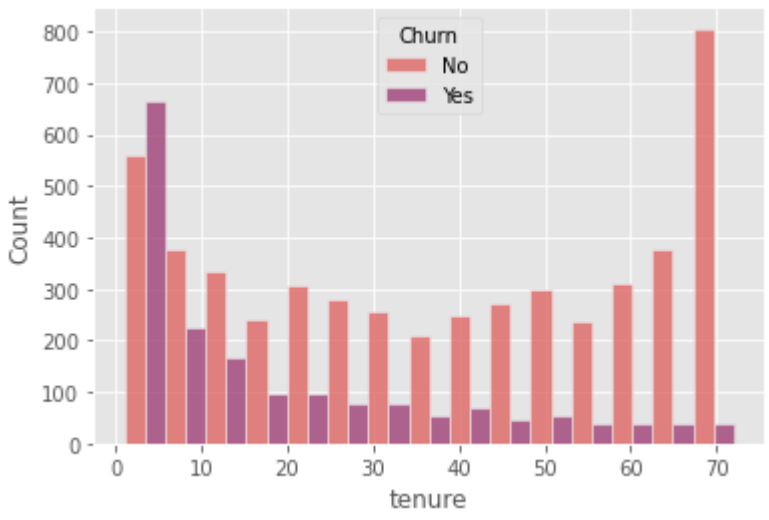
```
1.0     477
72.0    362
2.0     238
3.0     200
4.0     176
       ...
38.0     59
28.0     57
39.0     56
44.0     51
36.0     50
Name: tenure, Length: 72, dtype: int64
```

In [151]:
```
sns.displot(data=df, x="tenure", kind="kde")


#normal distribution
```

```
<seaborn.axisgrid.FacetGrid at 0x269c897d160>
```



In [152]:
```
sns.histplot(data=df, x="tenure", hue="Churn", multiple="dodge", palette ='flare');
```



New customers churn faster

In [153]:
```
df["tenure"].describe()


#normal distribution
```

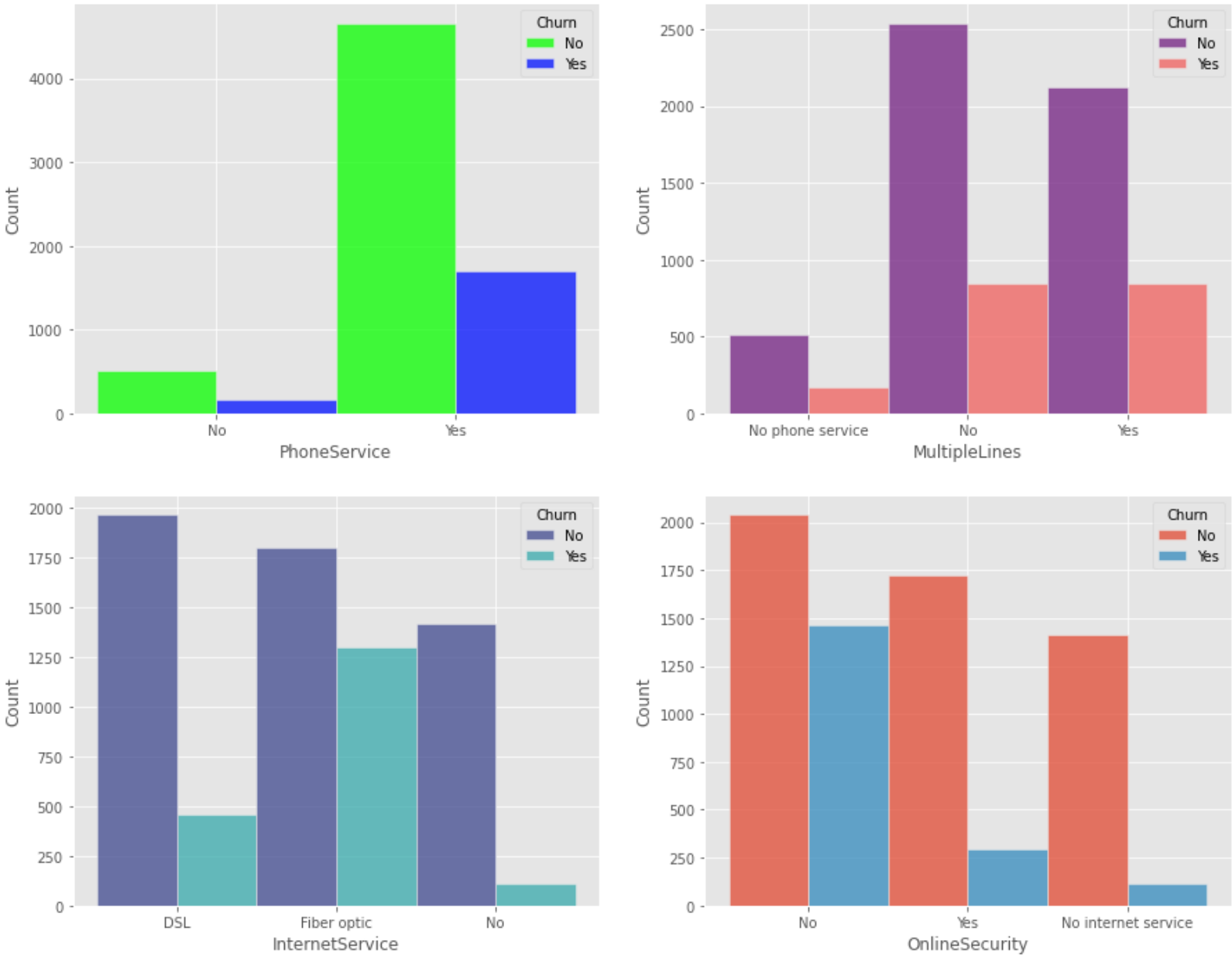
```
count    6896.000000
mean       33.041473
std        24.382260
min         1.000000
25%        10.000000
50%        30.000000
75%        56.000000
max        72.000000
Name: tenure, dtype: float64
```

Telecom services

In [154]:
```
print(df["PhoneService"].value_counts())
print(df["MultipleLines"].value_counts())
print(df["InternetService"].value_counts())
print(df["OnlineSecurity"].value_counts())
```

```
Yes    6361
No      682
Name: PhoneService, dtype: int64
No                  3390
Yes                 2971
No phone service     682
Name: MultipleLines, dtype: int64
Fiber optic    3096
DSL            2421
No             1526
Name: InternetService, dtype: int64
No                     3498
Yes                    2019
No internet service    1526
Name: OnlineSecurity, dtype: int64
```

In [155]:
```
fig, ax = plt.subplots(2, 2,figsize=(15,12))
sns.histplot(data=df,x="PhoneService",hue='Churn', palette='hsv' ,multiple="dodge",ax=ax[0,0])
sns.histplot(data=df,x="MultipleLines",hue='Churn', palette='magma',multiple="dodge",ax=ax[0,1])
sns.histplot(data=df,x="InternetService",hue='Churn', palette='mako',multiple="dodge",ax=ax[1,0])
sns.histplot(data=df,x='OnlineSecurity',hue='Churn',multiple="dodge",ax=ax[1,1]);
```



In [ ]:

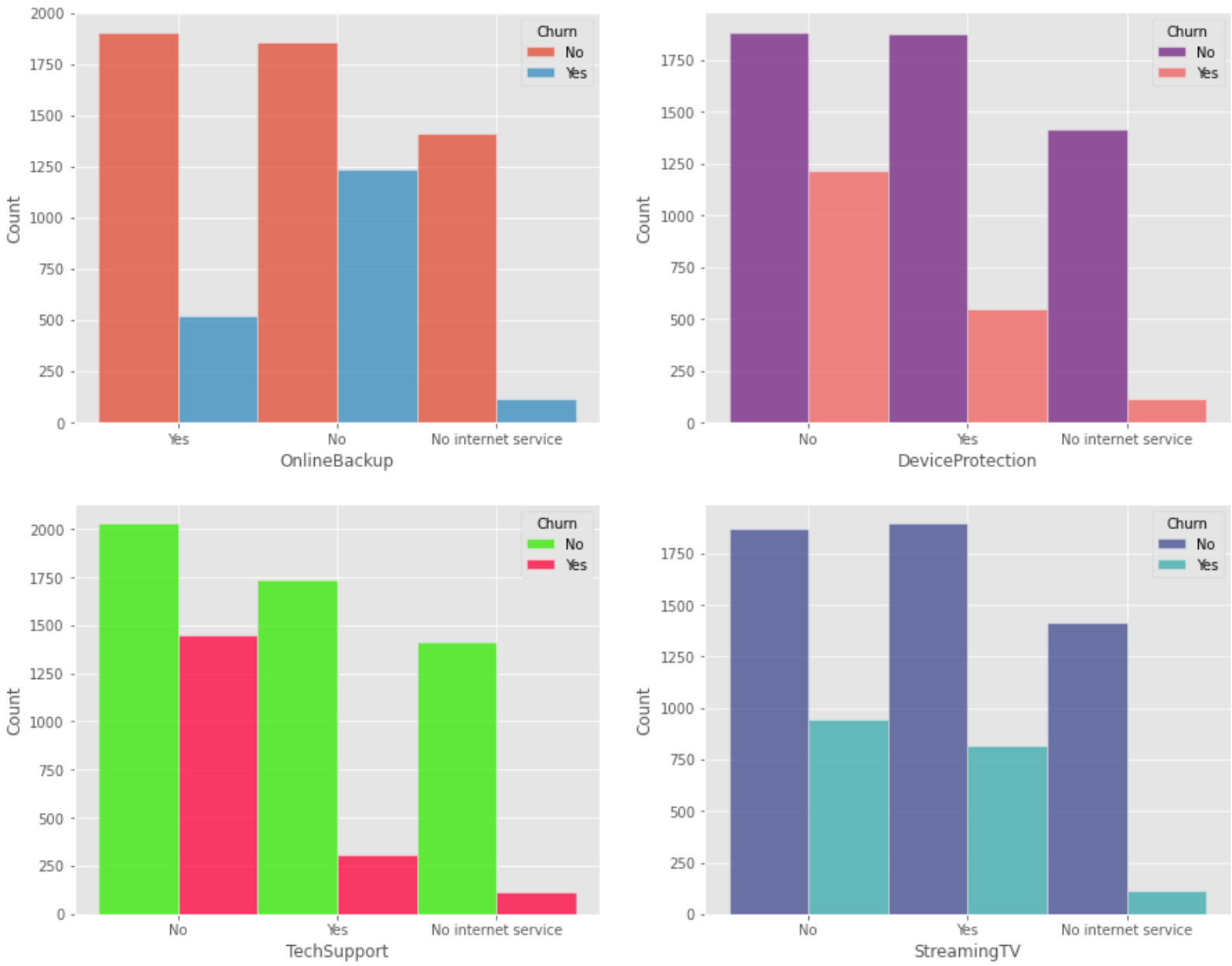
users fiber hight churn

Client who not use OnlineSecurity churn

In [156]:
```
print(df["OnlineBackup"].value_counts())
print(df["DeviceProtection"].value_counts())
print(df["TechSupport"].value_counts())
print(df["StreamingTV"].value_counts())
```

```
No                     3088
Yes                    2429
No internet service    1526
Name: OnlineBackup, dtype: int64
No                     3095
Yes                    2422
No internet service    1526
Name: DeviceProtection, dtype: int64
No                     3473
Yes                    2044
No internet service    1526
Name: TechSupport, dtype: int64
No                     2810
Yes                    2707
No internet service    1526
Name: StreamingTV, dtype: int64
```

In [157]:
```
fig, ax = plt.subplots(2,2,figsize=(15,12))
sns.histplot(data=df,x="OnlineBackup",hue='Churn',multiple="dodge",ax=ax[0,0])
sns.histplot(data=df,x="DeviceProtection",hue='Churn', palette='magma',multiple="dodge",ax=ax[0,1])
sns.histplot(data=df,x="TechSupport",hue='Churn', palette='prism',multiple="dodge",ax=ax[1,0])
sns.histplot(data=df,x='StreamingTV',hue='Churn', palette='mako',multiple="dodge",ax=ax[1,1]);
```



user streaming tv churn like user who not using it

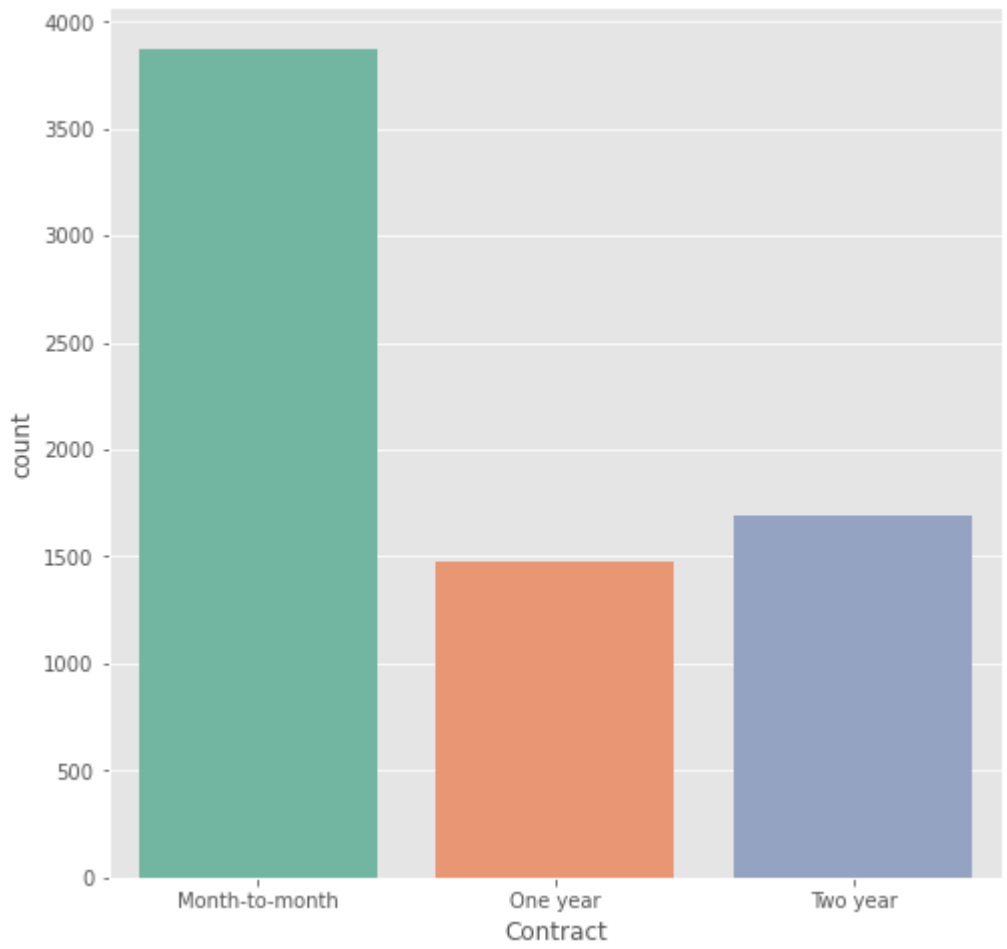
In [ ]:

contract

Contract: The contract term of the customer (Month-to-month, One year, Two year)

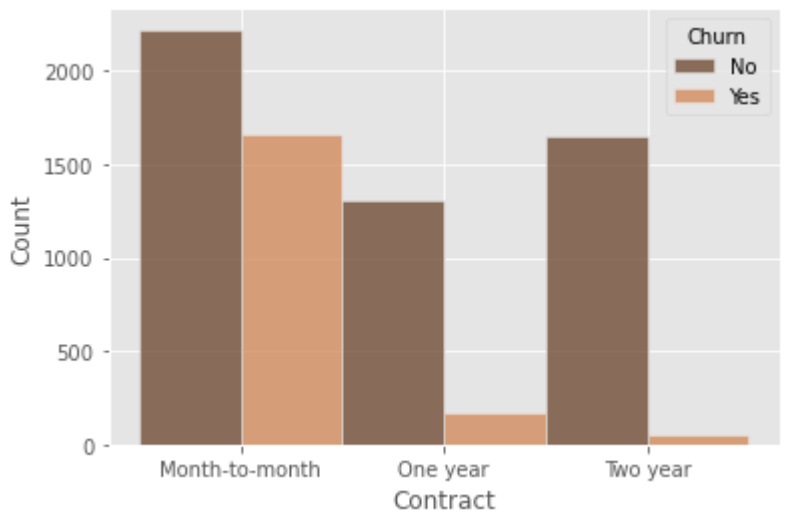
In [158]:
```
plt.figure(figsize=(8,8))
sns.countplot(x="Contract", palette='Set2' ,data=df)
```

```
<AxesSubplot:xlabel='Contract', ylabel='count'>
```



In [159]:
```
sns.histplot(data=df,x="Contract",hue='Churn', multiple="dodge" , palette='copper')
```

```
<AxesSubplot:xlabel='Contract', ylabel='Count'>
```



Month to month customer hightly churn

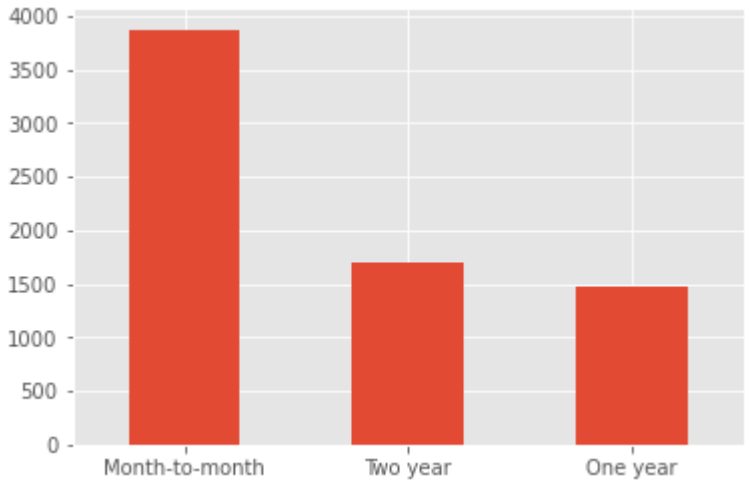
In [160]:
```
df["Contract"].value_counts()
```

```
Month-to-month    3875
Two year          1695
One year          1473
Name: Contract, dtype: int64
```

In [161]:
```
df.groupby([ 'Contract', 'Churn']).MonthlyCharges.count()
```

```
Contract        Churn
Month-to-month  No       2220
                Yes      1655
One year        No       1307
                Yes       166
Two year        No       1647
                Yes        48
Name: MonthlyCharges, dtype: int64
```

In [162]:
```
ax = df['Contract'].value_counts().plot(kind='bar', rot=0)
```



PaperlessBilling: Whether the customer has paperless billing or not (Yes, No)

In [163]:
```
df["PaperlessBilling"].value_counts()
```

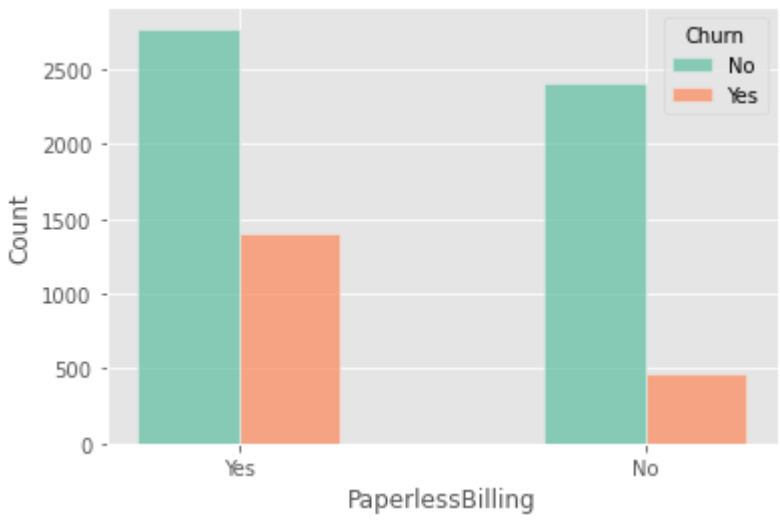
```
Yes    4171
No     2872
Name: PaperlessBilling, dtype: int64
```

In [164]:
```
df.groupby([ 'PaperlessBilling', 'Churn']).MonthlyCharges.count()
```

```
PaperlessBilling  Churn
No                No       2403
                  Yes       469
Yes               No       2771
                  Yes      1400
Name: MonthlyCharges, dtype: int64
```

In [165]:
```
sns.histplot(data=df,x="PaperlessBilling",hue='Churn',multiple="dodge", palette='Set2',shrink=.5)
```

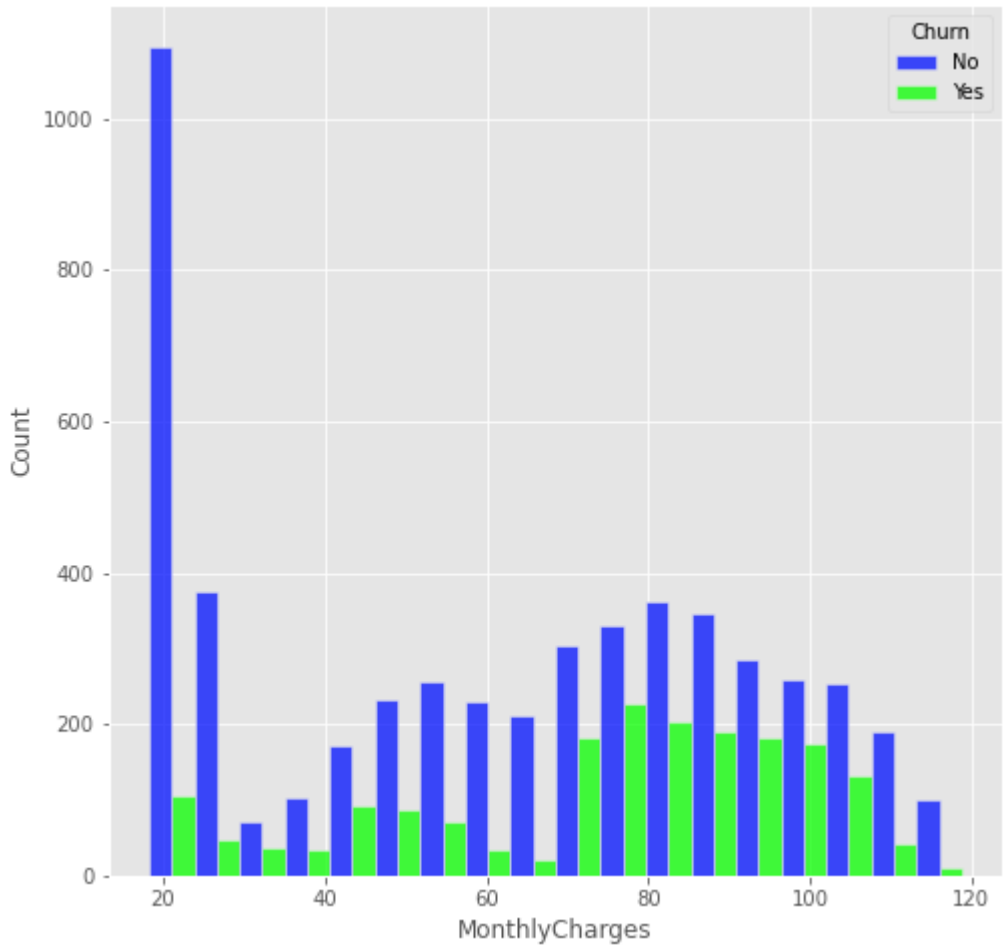
```
<AxesSubplot:xlabel='PaperlessBilling', ylabel='Count'>
```



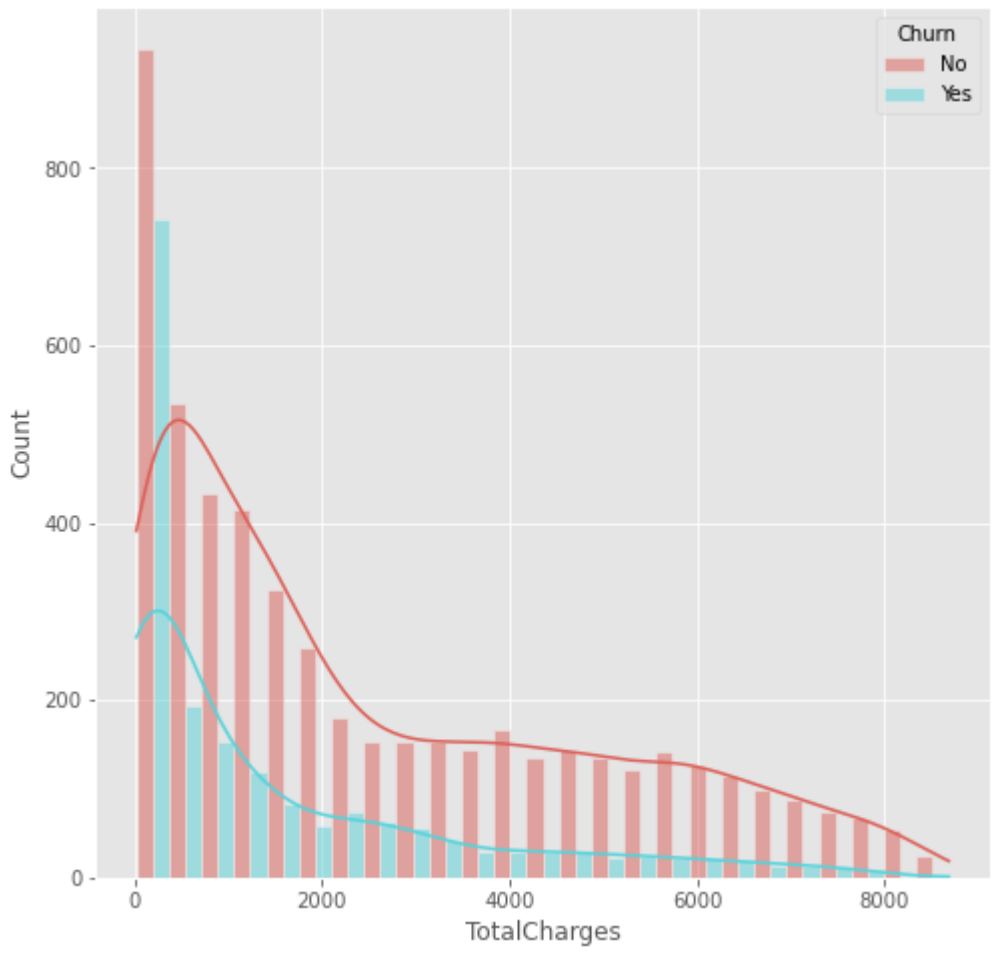
the customer has paperless billing hight churn

Charging

In [166]:
```
plt.figure(figsize=(8,8))

sns.histplot(data=df,x="MonthlyCharges", multiple="dodge",palette='hsv_r',hue ="Churn");
```



In [167]:
```
plt.figure(figsize=(8,8))
sns.histplot(data=df,kde=True,x="TotalCharges", multiple="dodge",palette='hls',hue ="Churn");
```



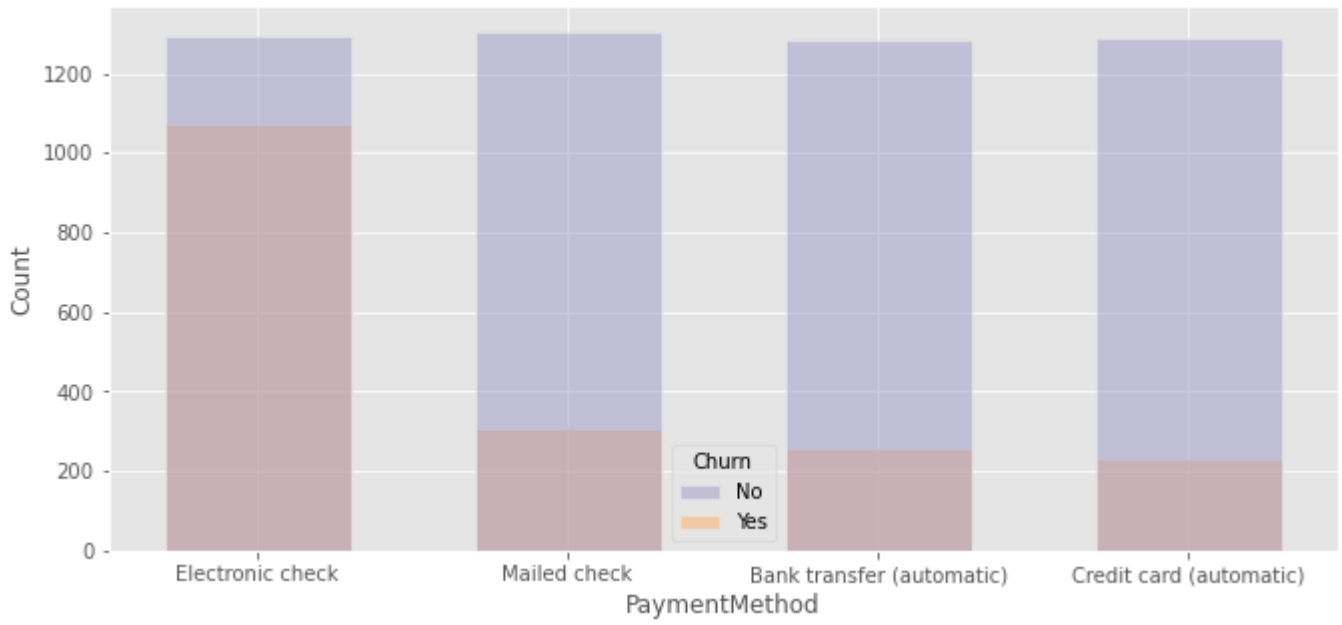
In [168]:
```
df["PaymentMethod"].value_counts()
```

```
Electronic check             2365
Mailed check                 1612
Bank transfer (automatic)    1544
Credit card (automatic)      1522
Name: PaymentMethod, dtype: int64
```

In [169]:
```
df.groupby([ 'PaymentMethod', 'Churn']).MonthlyCharges.count()
```

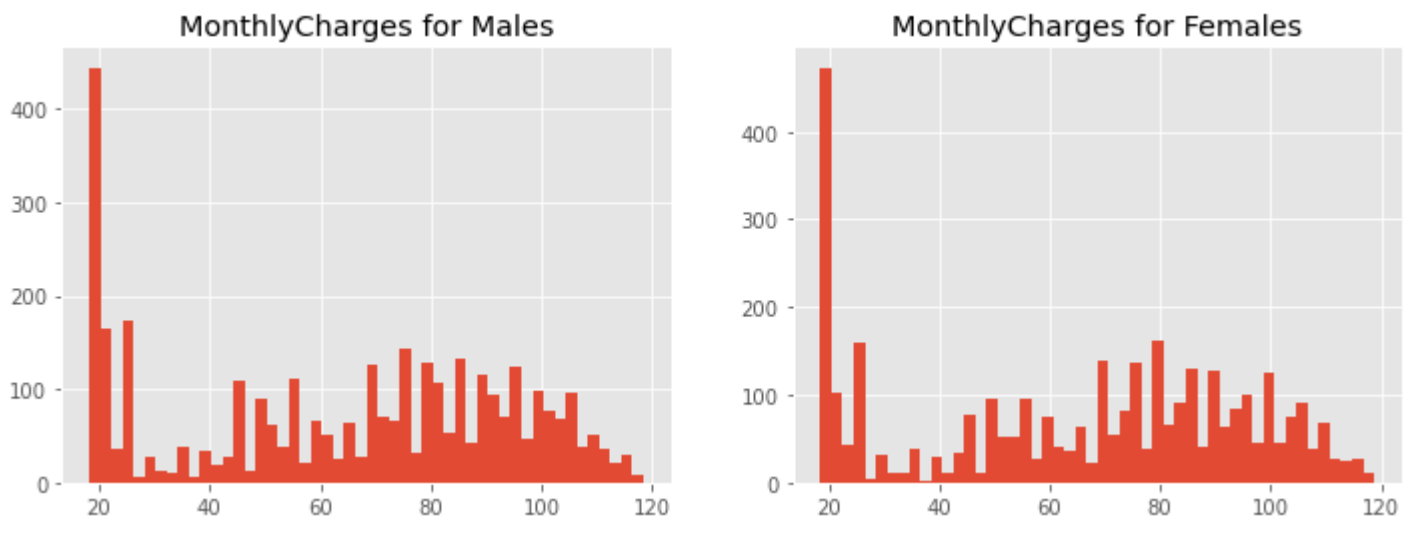
```
PaymentMethod              Churn
Bank transfer (automatic)  No       1286
                           Yes       258
Credit card (automatic)    No       1290
                           Yes       232
Electronic check           No       1294
                           Yes      1071
Mailed check               No       1304
                           Yes       308
Name: MonthlyCharges, dtype: int64
```

In [170]:
```
plt.figure(figsize=(11,5))
sns.histplot(data=df,x="PaymentMethod",hue='Churn', palette='tab20c_r',shrink =.6);


#Electronic check hight churn
```

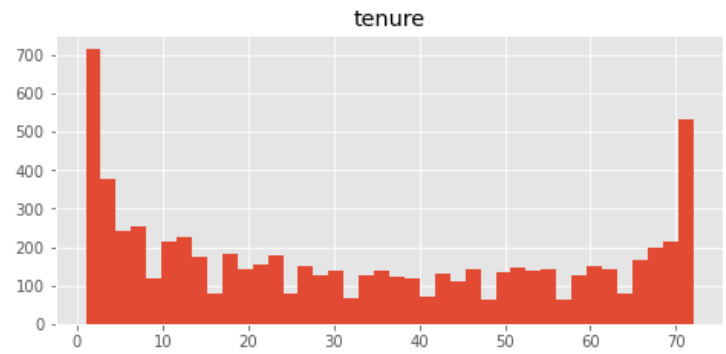


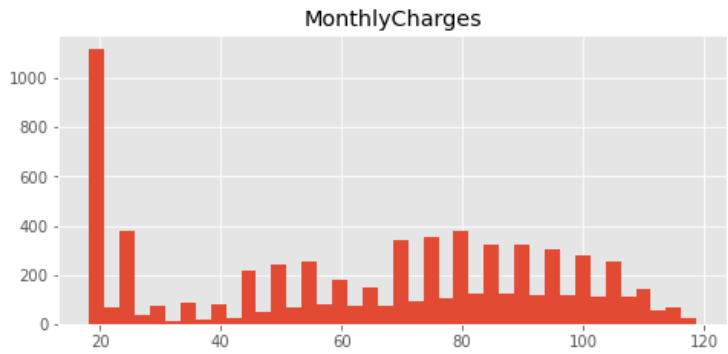
Electronic check hight churn

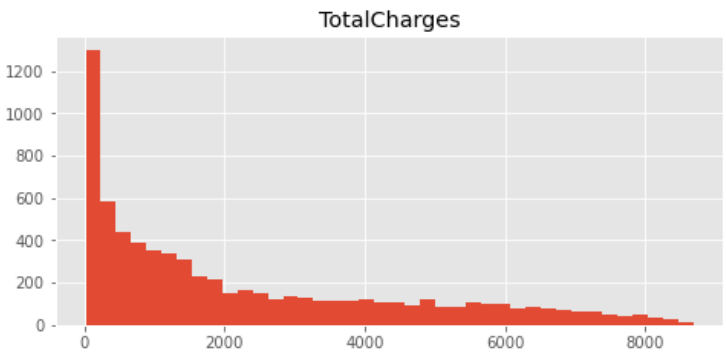
In [171]:
```
# get the distibution of MonthlyCharges for each gender alone
fig, ax = plt.subplots(1, 2, figsize=(12, 4))

plt.sca(ax[0])
df[df['gender']=='Male']['MonthlyCharges'].hist(bins=50)
plt.title('MonthlyCharges for Males')

plt.sca(ax[1])
df[df['gender']=='Female']['MonthlyCharges'].hist(bins=50)
plt.title('MonthlyCharges for Females');   ## Distribution is almost the same
```



In [172]:
```
# tenure and MonthlyCharges and TotalCharges historam ditribution
df[['tenure', 'MonthlyCharges', 'TotalCharges']].hist(bins=40, figsize=(18, 8));
```







finell report

-we have 26.54% churn
-gender not important feature
-customer over 65 go out relative to their number
-no partner hight churn
-no dependents hight churn
-New customers churn faster
-we have a problem on fiber optic service
-we can make offer on online security to make customers trying it
-Streaming TV service not good
-Month to month customer hightly churn
-customer has paperless billing hight churn
-why electronic check hight churn ?

In [ ]:

In [ ]:

session 2

In [216]:
```
data = df.copy()
data
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSec
0	7590-VHVEG	Female	NaN	Yes	No	NaN	No	No phone service	DSL	No
1	5575-GNVDE	Male	NaN	No	No	34.0	Yes	No	DSL	Yes
2	3668-QPYBK	Male	NaN	No	No	2.0	Yes	No	DSL	Yes
3	7795-CFOCW	Male	NaN	No	No	45.0	No	No phone service	DSL	Yes
4	9237-HQITU	Female	NaN	No	No	2.0	Yes	No	Fiber optic	No
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0.0	Yes	Yes	24.0	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0.0	Yes	Yes	72.0	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0.0	Yes	Yes	11.0	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1.0	Yes	No	4.0	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0.0	No	No	66.0	Yes	No	Fiber optic	Yes

7043 rows × 21 columns

In [217]:
```
data.dropna(inplace= True)
```

In [218]:
```
data
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSec
30	3841-NFECX	Female	1.0	Yes	No	71.0	Yes	Yes	Fiber optic	Yes
31	4929-XIHVW	Male	1.0	Yes	No	2.0	Yes	No	Fiber optic	No
47	7760-OYPDY	Female	0.0	No	No	2.0	Yes	No	Fiber optic	No
48	7639-LIAYI	Male	0.0	No	No	52.0	Yes	Yes	DSL	Yes
49	2954-PIBKO	Female	0.0	Yes	Yes	69.0	Yes	Yes	DSL	Yes
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0.0	Yes	Yes	24.0	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0.0	Yes	Yes	72.0	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0.0	Yes	Yes	11.0	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1.0	Yes	No	4.0	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0.0	No	No	66.0	Yes	No	Fiber optic	Yes

```
6857 rows × 21 columns
```

Business Questions

what is the ratio between males and females in our company?
what is the ratio between SeniorCitizens and others in our company?
what is the ratio between who has partners and not in our company?
what is the ratio between who has dependents and not in our company?
what is the ratio between who has MultipleLines and not in our company?
Depending on the correlation matrix, which variables have a strong relationship with each other?
From the previous question,What is the relationship between the Internet Services and the churn rate?
how many Internet Services we provide in our company? - list names and ratio please -
what is our Contract types we provide? - names and ratio please-
how many customers uses StreamingTV ?
what is the ratio between users who streaming movies to StreamingTV subscribers?
Is there a strong relationship between the monthly recharge rate and the dependents?
who is the the most important customer in the company according to Monthly and Total charges?
how many payment methods we provide? and what is the ratio between each others?
what is our churn rate?
what is the average monthly charge?
From the correlation matrix ,What is the relationship between the Senior Citizens and the monthly charging rate?

We want to give offers according to the monthly charge categories, can you explain that? Using Visulization

We want to present offers by g

In [185]:
```
#what is the ratio between males and females in our company?
round(data['gender'].value_counts()/data.shape[0]*100,2)
```

```
Male      50.3
Female    49.7
Name: gender, dtype: float64
```

In [186]:
```
#what is the ratio between SeniorCitizens and others in our company
round(data['SeniorCitizen'].value_counts()/data.shape[0]*100,2)
```

```
0.0    83.59
1.0    16.41
Name: SeniorCitizen, dtype: float64
```

In [201]:
```
#what is the ratio between who has partners and not in our company?
data.groupby([ 'Partner', 'Churn']).customerID.count()/data.shape[0]*100
```

```
Partner  Churn
No       No       34.665306
         Yes      16.377425
Yes      No       39.404988
         Yes       9.552282
Name: customerID, dtype: float64
```

In [202]:
```
data.groupby([ 'Partner', 'Churn']).customerID.count()['Yes']['Yes']/data.shape[0]*100
```

```
9.552282339215399
```

In [203]:
```
#what is the ratio between who has dependents and not in our company?
data.groupby([ 'Dependents', 'Churn']).customerID.count()/data.shape[0]*100
```

```
Dependents  Churn
No          No       48.563512
            Yes      21.379612
Yes         No       25.506781
            Yes       4.550095
Name: customerID, dtype: float64
```

In [204]:
```
data.groupby([ 'Dependents', 'Churn']).customerID.count()['Yes']['Yes']/data.shape[0]*100
```

```
4.550094793641534
```

In [205]:
```
#what is the ratio between who has MultipleLines and not in our company?


data.groupby([ 'MultipleLines', 'Churn']).customerID.count()/data.shape[0]*100
```

```
MultipleLines     Churn
No                No       36.065335
                  Yes      11.579408
No phone service  No        7.350153
                  Yes       2.289631
Yes               No       30.654805
                  Yes      12.060668
Name: customerID, dtype: float64
```

In [206]:
```
data.groupby([ 'MultipleLines', 'Churn']).customerID.count()['Yes']['Yes']/data.shape[0]*100
```

```
12.060667930581886
```

In [219]:
```
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, StandardScaler, OrdinalEncoder
def label_encoder(dataframe, binary_col):
    labelencoder = LabelEncoder()
    dataframe[binary_col] = labelencoder.fit_transform(dataframe[binary_col])
    return dataframe

binary_cols = [col for col in data.columns if data[col].dtype not in [int, float]
               and data[col].nunique() == 2 ]

for col in binary_cols:
    label_encoder(data, col)
```

In [220]:
```
def one_hot_encoder(dataframe, categorical_cols, drop_first=True):
    dataframe = pd.get_dummies(dataframe, columns=categorical_cols, drop_first=drop_first)
    return dataframe

ohe_cols = [col for col in data.columns if 10 >= data[col].nunique() and col not in binary_cols]

data = one_hot_encoder(data, ohe_cols)
data
```

	customerID	gender	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	Churr
30	3841-NFECX	0	1	0	71.0	1	1	96.35	6766.95	0
31	4929-XIHVW	1	1	0	2.0	1	1	95.50	181.65	0
47	7760-OYPDY	0	0	0	2.0	1	1	80.65	144.15	1
48	7639-LIAYI	1	0	0	52.0	1	1	79.75	4217.80	0
49	2954-PIBKO	0	1	1	69.0	1	1	64.15	4254.10	0
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	1	1	1	24.0	1	1	84.80	1990.50	0
7039	2234-XADUH	0	1	1	72.0	1	1	103.20	7362.90	0
7040	4801-JZAZL	0	1	1	11.0	0	1	29.60	346.45	0
7041	8361-LTMKD	1	1	0	4.0	1	1	74.40	306.60	1
7042	3186-AJIEK	1	0	0	66.0	1	1	105.65	6844.50	0

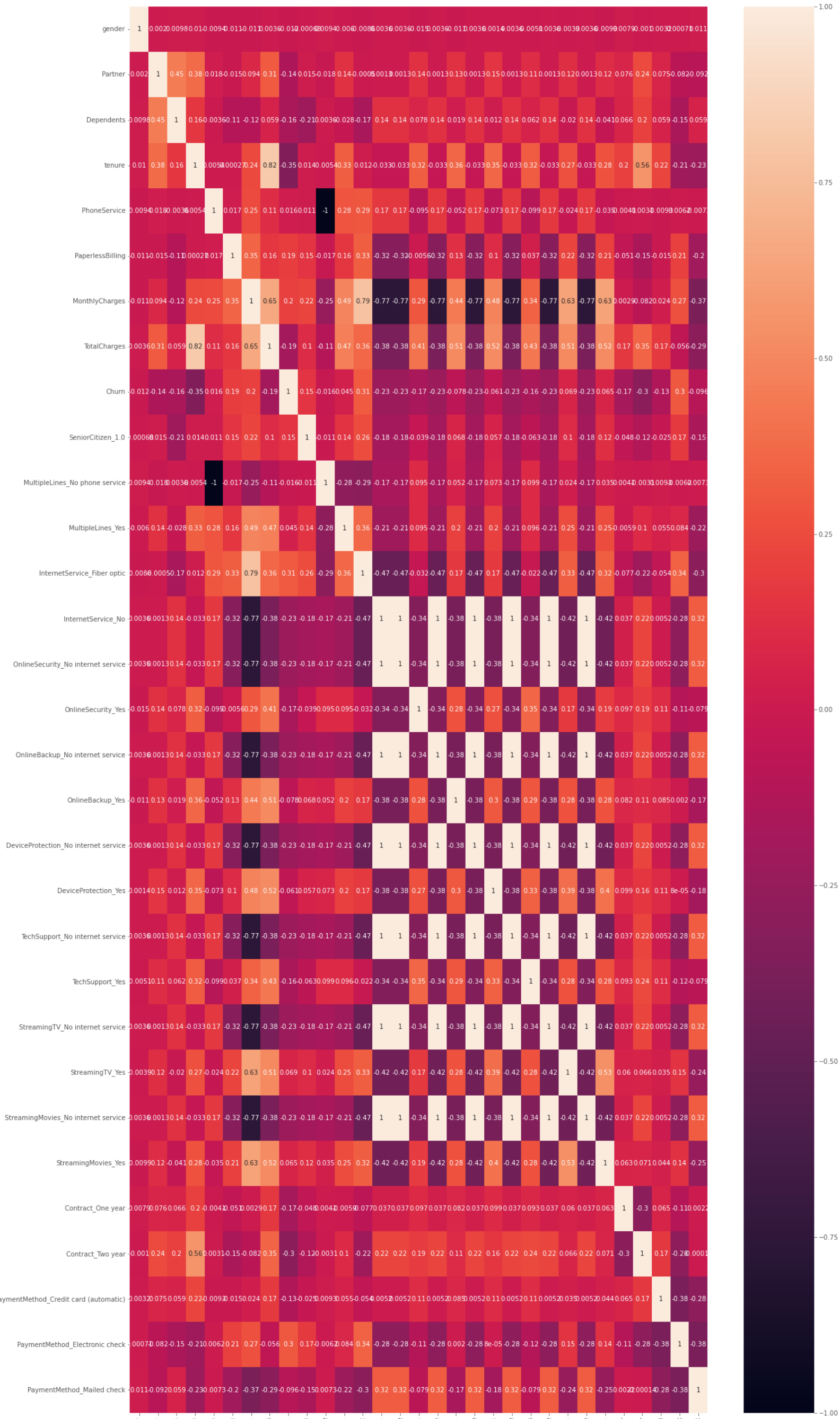
6857 rows × 32 columns

In [221]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6857 entries, 30 to 7042
Data columns (total 32 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   customerID                             6857 non-null   object
 1   gender                                 6857 non-null   int32
 2   Partner                                6857 non-null   int32
 3   Dependents                             6857 non-null   int32
 4   tenure                                 6857 non-null   float64
 5   PhoneService                           6857 non-null   int32
 6   PaperlessBilling                       6857 non-null   int32
 7   MonthlyCharges                         6857 non-null   float64
 8   TotalCharges                           6857 non-null   float64
 9   Churn                                  6857 non-null   int32
 10  SeniorCitizen_1.0                      6857 non-null   uint8
 11  MultipleLines_No phone service         6857 non-null   uint8
 12  MultipleLines_Yes                      6857 non-null   uint8
 13  InternetService_Fiber optic            6857 non-null   uint8
 14  InternetService_No                     6857 non-null   uint8
 15  OnlineSecurity_No internet service     6857 non-null   uint8
 16  OnlineSecurity_Yes                     6857 non-null   uint8
 17  OnlineBackup_No internet service       6857 non-null   uint8
 18  OnlineBackup_Yes                       6857 non-null   uint8
 19  DeviceProtection_No internet service   6857 non-null   uint8
 20  DeviceProtection_Yes                   6857 non-null   uint8
 21  TechSupport_No internet service        6857 non-null   uint8
 22  TechSupport_Yes                        6857 non-null   uint8
 23  StreamingTV_No internet service        6857 non-null   uint8
 24  StreamingTV_Yes                        6857 non-null   uint8
 25  StreamingMovies_No internet service    6857 non-null   uint8
 26  StreamingMovies_Yes                    6857 non-null   uint8
 27  Contract_One year                      6857 non-null   uint8
 28  Contract_Two year                      6857 non-null   uint8
 29  PaymentMethod_Credit card (automatic)  6857 non-null   uint8
 30  PaymentMethod_Electronic check         6857 non-null   uint8
 31  PaymentMethod_Mailed check             6857 non-null   uint8
dtypes: float64(3), int32(6), object(1), uint8(22)
memory usage: 575.9+ KB
```

In [222]:
```
plt.figure(figsize=(20, 40))
sns.heatmap(data.corr(),annot = True)
```

```
<AxesSubplot:>
```



In [227]:
```
#Depending on the correlation matrix, which variables have a strong relationship with each other?
corr = (data.corr()>=.3).sum()-1
corr
```

```
gender                                   0
Partner                                  3
Dependents                               1
tenure                                   8
PhoneService                             0
PaperlessBilling                         2
MonthlyCharges                           9
TotalCharges                            12
Churn                                    2
SeniorCitizen_1.0                        0
MultipleLines_No phone service           0
MultipleLines_Yes                        4
InternetService_Fiber optic              8
InternetService_No                       7
OnlineSecurity_No internet service       7
OnlineSecurity_Yes                       3
OnlineBackup_No internet service         7
OnlineBackup_Yes                         4
DeviceProtection_No internet service     7
DeviceProtection_Yes                     7
TechSupport_No internet service          7
TechSupport_Yes                          5
StreamingTV_No internet service          7
StreamingTV_Yes                          5
StreamingMovies_No internet service      7
StreamingMovies_Yes                      5
Contract_One year                        0
Contract_Two year                        2
PaymentMethod_Credit card (automatic)    0
PaymentMethod_Electronic check           2
PaymentMethod_Mailed check               7
dtype: int64
```

In [ ]:
```
#From the previous question,What is the relationship between the Internet Services and the churn rate?
```

In [228]:
```
data.corr()['Churn']
```

```
gender                                  -0.011524
Partner                                 -0.143429
Dependents                              -0.161420
tenure                                  -0.347047
PhoneService                             0.016231
PaperlessBilling                         0.190699
MonthlyCharges                           0.198017
TotalCharges                            -0.190510
Churn                                    1.000000
SeniorCitizen_1.0                        0.153017
MultipleLines_No phone service          -0.016231
MultipleLines_Yes                        0.045421
InternetService_Fiber optic              0.310888
InternetService_No                      -0.226327
OnlineSecurity_No internet service      -0.226327
OnlineSecurity_Yes                      -0.166200
OnlineBackup_No internet service        -0.226327
OnlineBackup_Yes                        -0.077842
DeviceProtection_No internet service    -0.226327
DeviceProtection_Yes                    -0.061086
TechSupport_No internet service         -0.226327
TechSupport_Yes                         -0.158977
StreamingTV_No internet service         -0.226327
StreamingTV_Yes                          0.068889
StreamingMovies_No internet service     -0.226327
StreamingMovies_Yes                      0.064815
Contract_One year                       -0.172629
Contract_Two year                       -0.299489
PaymentMethod_Credit card (automatic)   -0.129815
PaymentMethod_Electronic check           0.300294
PaymentMethod_Mailed check              -0.096431
Name: Churn, dtype: float64
```

In [230]:
```
#how many Internet Services we provide in our company? - list names and ratio please -
df["InternetService"].value_counts()/df.shape[0]*100
```

```
Fiber optic    43.958540
DSL            34.374556
No             21.666903
Name: InternetService, dtype: float64
```

In [231]:
```
#what is our Contract types we provide? - names and ratio please-
df["Contract"].value_counts()/df.shape[0]*100
```

```
Month-to-month    55.019168
Two year          24.066449
One year          20.914383
Name: Contract, dtype: float64
```

In [233]:
```
#how many customers uses StreamingTV ?
df["StreamingTV"].value_counts()["Yes"]
```

```
2707
```

In [234]:
```
#what is the ratio between users who streaming movies to StreamingTV subscribers?
df["StreamingMovies"].value_counts()["Yes"]/df["StreamingTV"].value_counts()["Yes"]
```

```
1.009235315847802
```

In [235]:
```
#Is there a strong relationship between the monthly recharge rate and the dependents?
data.corr()['Dependents']['MonthlyCharges']
#no
```

```
-0.1170513806195996
```

In [237]:
```
#who is the the most important customer in the company according to Monthly and Total charges?
x= df['MonthlyCharges'].idxmax()
df.iloc[x]
```

```
customerID                          7569-NMZYQ
gender                                  Female
SeniorCitizen                              0.0
Partner                                    Yes
Dependents                                 Yes
tenure                                    72.0
PhoneService                               Yes
MultipleLines                              Yes
InternetService                    Fiber optic
OnlineSecurity                             Yes
OnlineBackup                               Yes
DeviceProtection                           Yes
TechSupport                                Yes
StreamingTV                                Yes
StreamingMovies                            Yes
Contract                              Two year
PaperlessBilling                           Yes
PaymentMethod       Bank transfer (automatic)
MonthlyCharges                         118.75
TotalCharges                          8672.45
Churn                                      No
Name: 4586, dtype: object
```

In [238]:
```
#what is the average monthly charge?
df["MonthlyCharges"].mean()
```

```
64.76169246059922
```

In [241]:
```
#From the correlation matrix ,What is the relationship between the Senior Citizens and the monthly charging
data.corr()['SeniorCitizen_1.0']['MonthlyCharges']
```

```
0.22039659649256546
```

In [242]:
```
#We want to present offers by gender and the Senior Citizen, could you explain that?
def offer(row):
    if row['gender']=="Male" and row['SeniorCitizen']==1:
        return "Male_senior"
    elif row['gender']=="Male" and row['SeniorCitizen']==0:
        return "Male_junior"
    elif row['gender']=="Female" and row['SeniorCitizen']==1:
        return "Female_senior"
    elif row['gender']=="Female" and row['SeniorCitizen']==0:
        return "Female_junior"
    else :
        return "Other"
```

In [243]:
```
df["offer"]=df.apply(offer, axis =1)
```

In [244]:
```
df
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSec
0	7590-VHVEG	Female	NaN	Yes	No	NaN	No	No phone service	DSL	No
1	5575-GNVDE	Male	NaN	No	No	34.0	Yes	No	DSL	Yes
2	3668-QPYBK	Male	NaN	No	No	2.0	Yes	No	DSL	Yes
3	7795-CFOCW	Male	NaN	No	No	45.0	No	No phone service	DSL	Yes
4	9237-HQITU	Female	NaN	No	No	2.0	Yes	No	Fiber optic	No
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0.0	Yes	Yes	24.0	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0.0	Yes	Yes	72.0	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0.0	Yes	Yes	11.0	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1.0	Yes	No	4.0	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0.0	No	No	66.0	Yes	No	Fiber optic	Yes

7043 rows × 22 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: