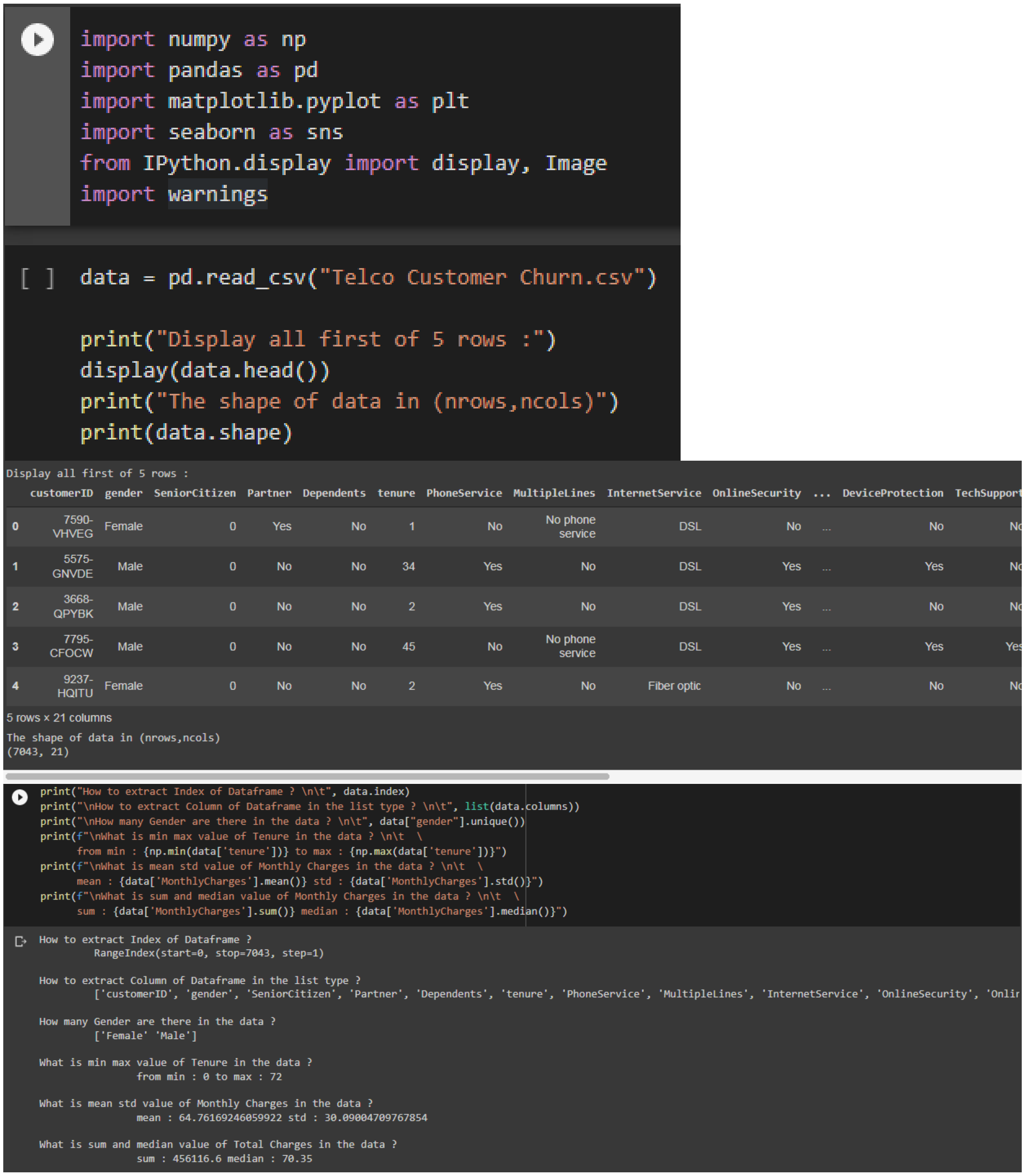


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display, Image
import warnings
```

```python
[ ]  data = pd.read_csv("Telco Customer Churn.csv")

     print("Display all first of 5 rows :")
     display(data.head())
     print("The shape of data in (nrows,ncols)")
     print(data.shape)
```

```
Display all first of 5 rows :
```

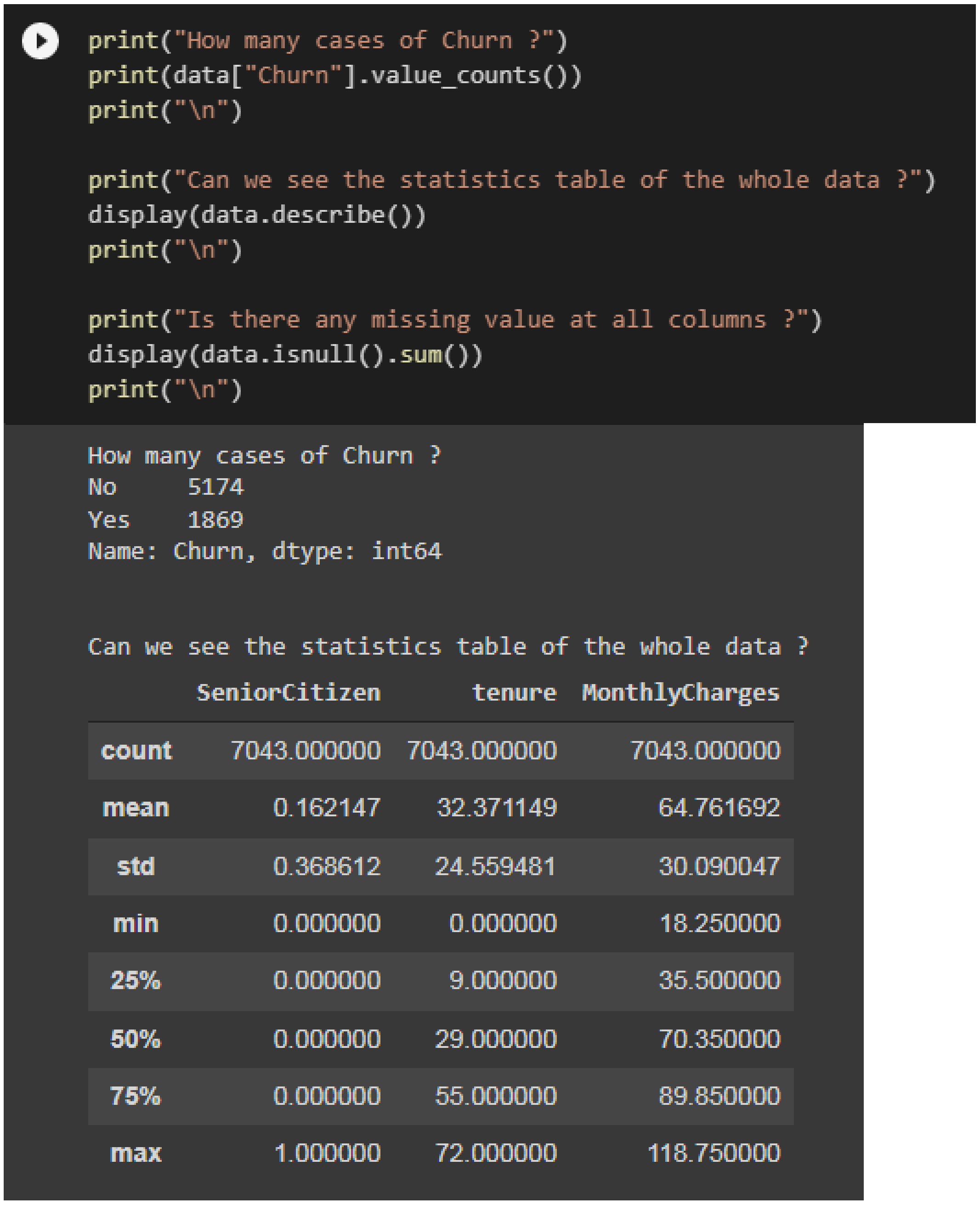
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	N
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	N
2	3668-QPYBK	Male	0	No	No	2	Yes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

5 rows × 21 columns

```
The shape of data in (nrows,ncols)
(7043, 21)
```

```python
print("How to extract Index of Dataframe ? \n\t", data.index)
print("\nHow to extract Column of Dataframe in the list type ? \n\t", list(data.columns))
print("\nHow many Gender are there in the data ? \n\t", data["gender"].unique())
print(f"\nWhat is min max value of Tenure in the data ? \n\t  \
      from min : {np.min(data['tenure'])} to max : {np.max(data['tenure'])}")
print(f"\nWhat is mean std value of Monthly Charges in the data ? \n\t  \
      mean : {data['MonthlyCharges'].mean()} std : {data['MonthlyCharges'].std()}")
print(f"\nWhat is sum and median value of Monthly Charges in the data ? \n\t  \
      sum : {data['MonthlyCharges'].sum()} median : {data['MonthlyCharges'].median()}")
```

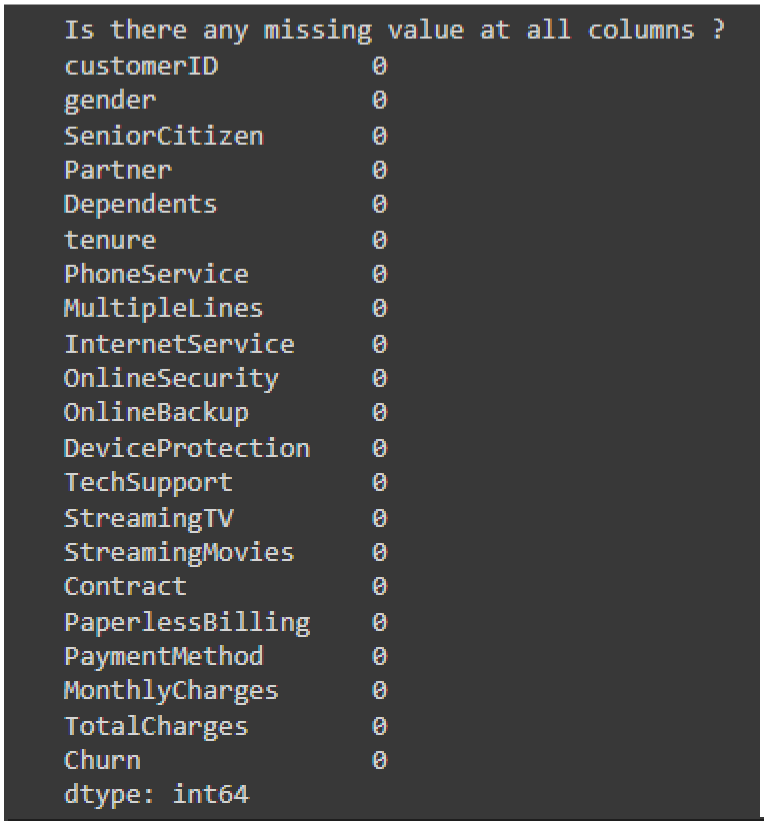
```
How to extract Index of Dataframe ?
         RangeIndex(start=0, stop=7043, step=1)

How to extract Column of Dataframe in the list type ?
         ['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'Onlir

How many Gender are there in the data ?
         ['Female' 'Male']

What is min max value of Tenure in the data ?
                from min : 0 to max : 72

What is mean std value of Monthly Charges in the data ?
                mean : 64.76169246059922 std : 30.09004709767854

What is sum and median value of Total Charges in the data ?
                sum : 456116.6 median : 70.35
```



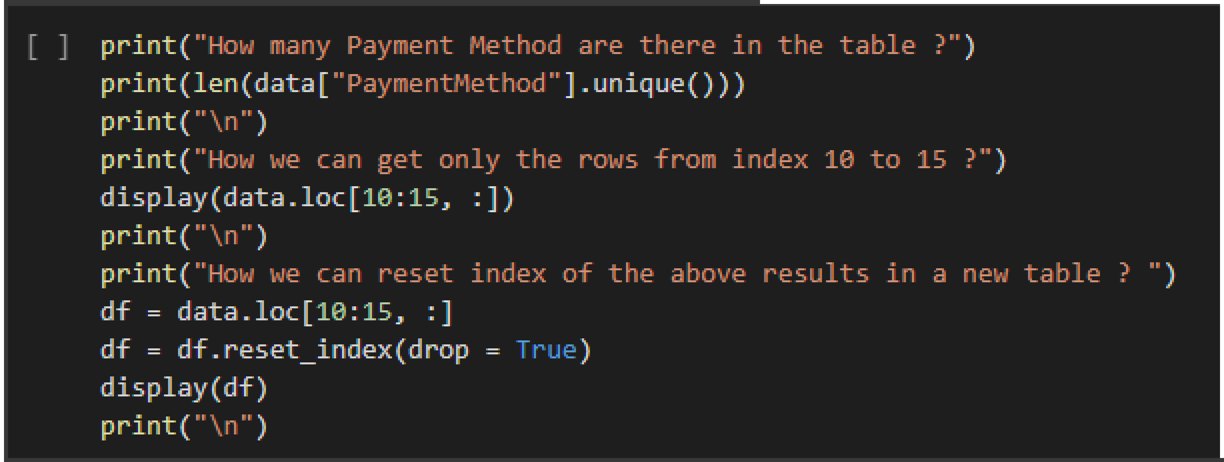
```
print("How many cases of Churn ?")
print(data["Churn"].value_counts())
print("\n")

print("Can we see the statistics table of the whole data ?")
display(data.describe())
print("\n")

print("Is there any missing value at all columns ?")
display(data.isnull().sum())
print("\n")
```

```
How many cases of Churn ?
No     5174
Yes    1869
Name: Churn, dtype: int64


Can we see the statistics table of the whole data ?
```

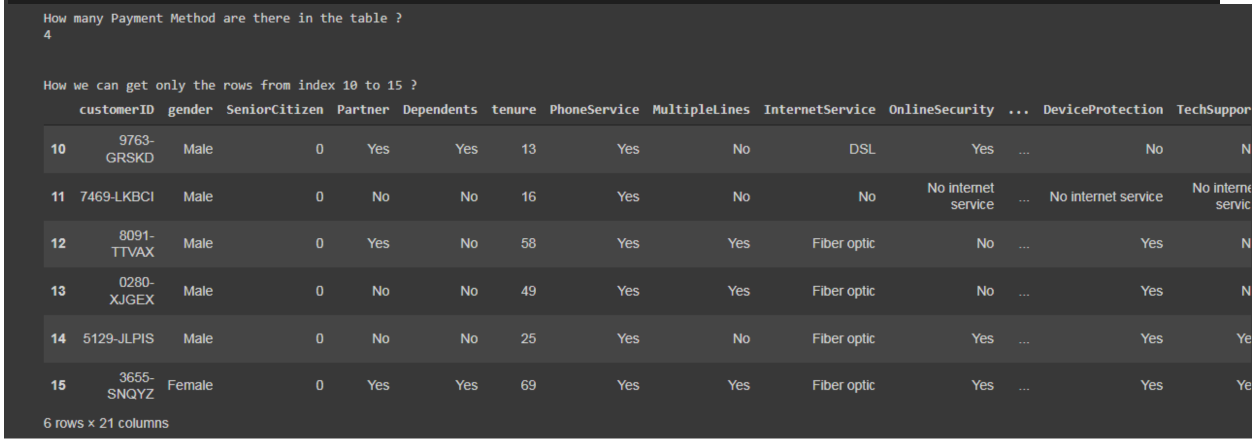
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000



```
Is there any missing value at all columns ?
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```



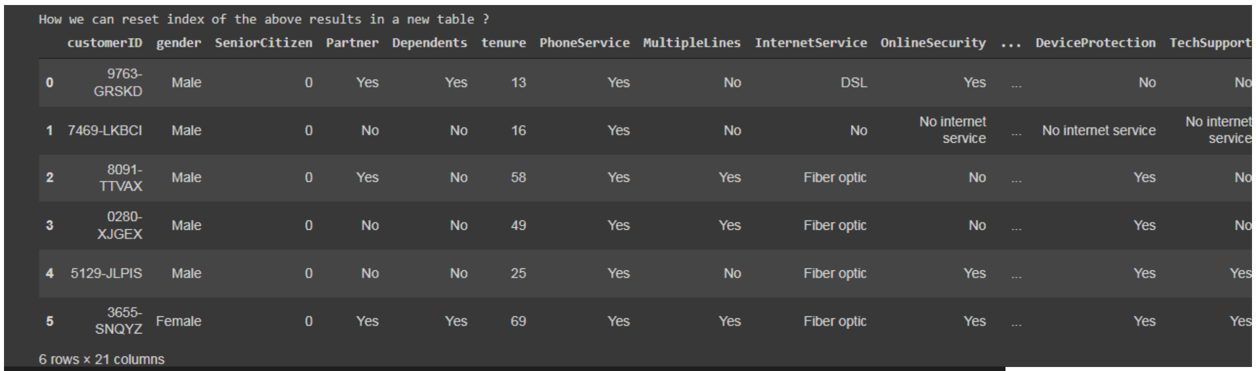
```
[ ]  print("How many Payment Method are there in the table ?")
     print(len(data["PaymentMethod"].unique()))
     print("\n")
     print("How we can get only the rows from index 10 to 15 ?")
     display(data.loc[10:15, :])
     print("\n")
     print("How we can reset index of the above results in a new table ? ")
     df = data.loc[10:15, :]
     df = df.reset_index(drop = True)
     display(df)
     print("\n")
```



```
How many Payment Method are there in the table ?
4


How we can get only the rows from index 10 to 15 ?
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSuppor
10	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	...	No	N
11	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	...	No internet service	No interne servic
12	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	...	Yes	N
13	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	...	Yes	N
14	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	...	Yes	Ye
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes	Ye

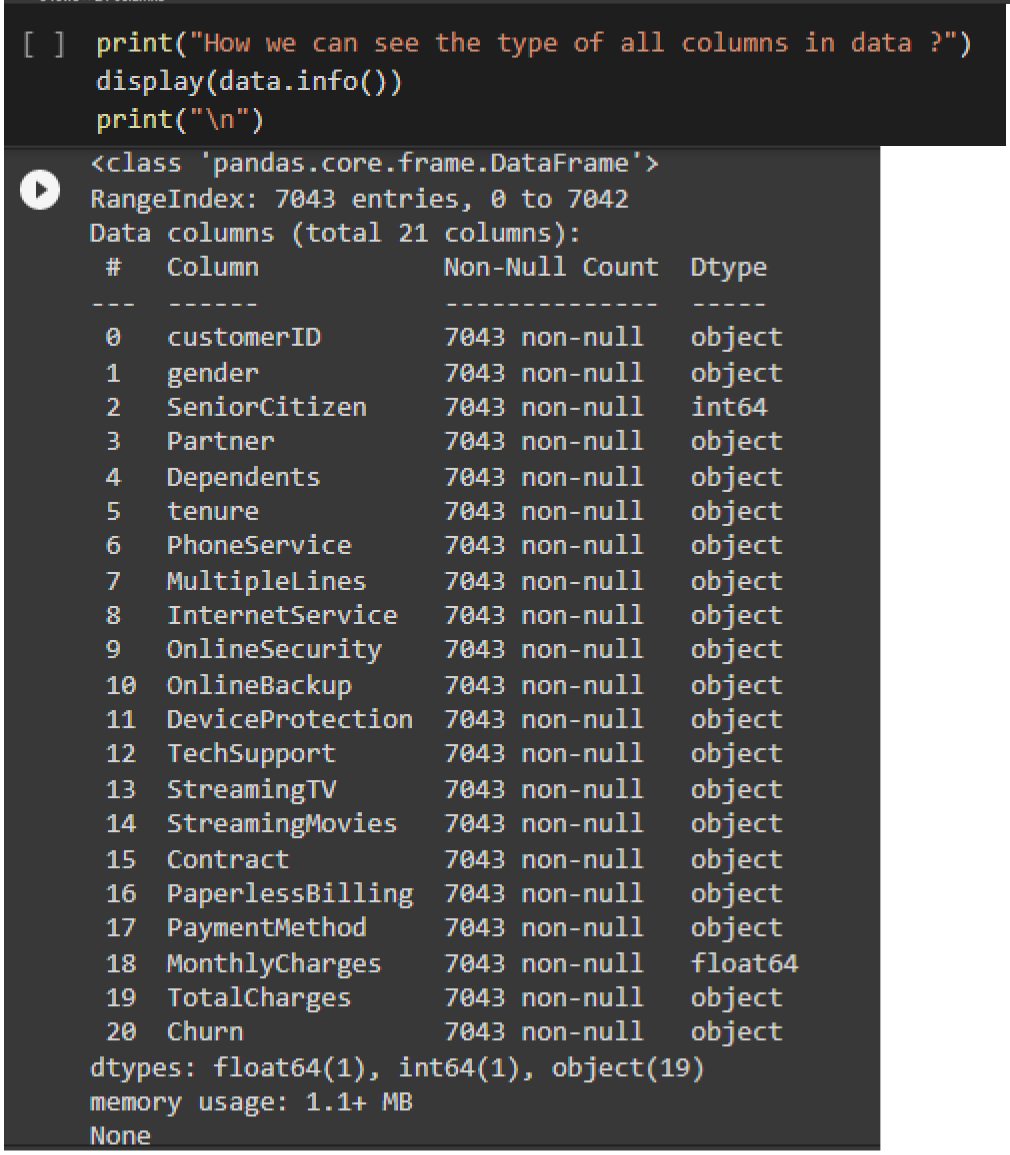
6 rows × 21 columns



```
How we can reset index of the above results in a new table ?
```

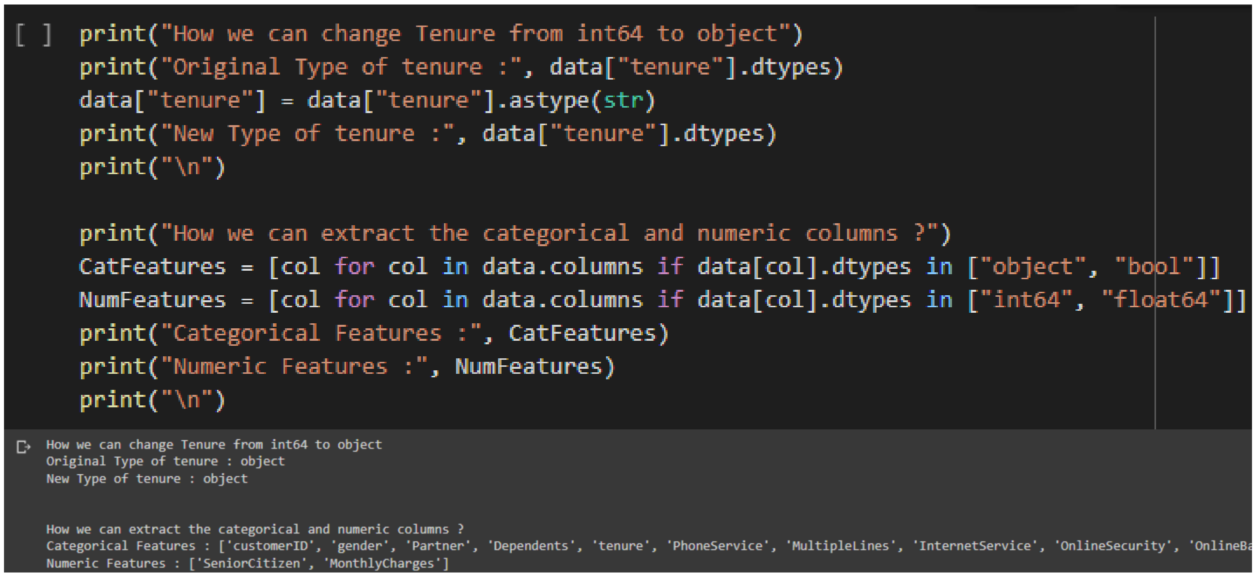
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	...	No	No
1	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	...	No internet service	No internet service
2	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	...	Yes	No
3	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	...	Yes	No
4	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	...	Yes	Yes
5	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes	Yes

6 rows × 21 columns



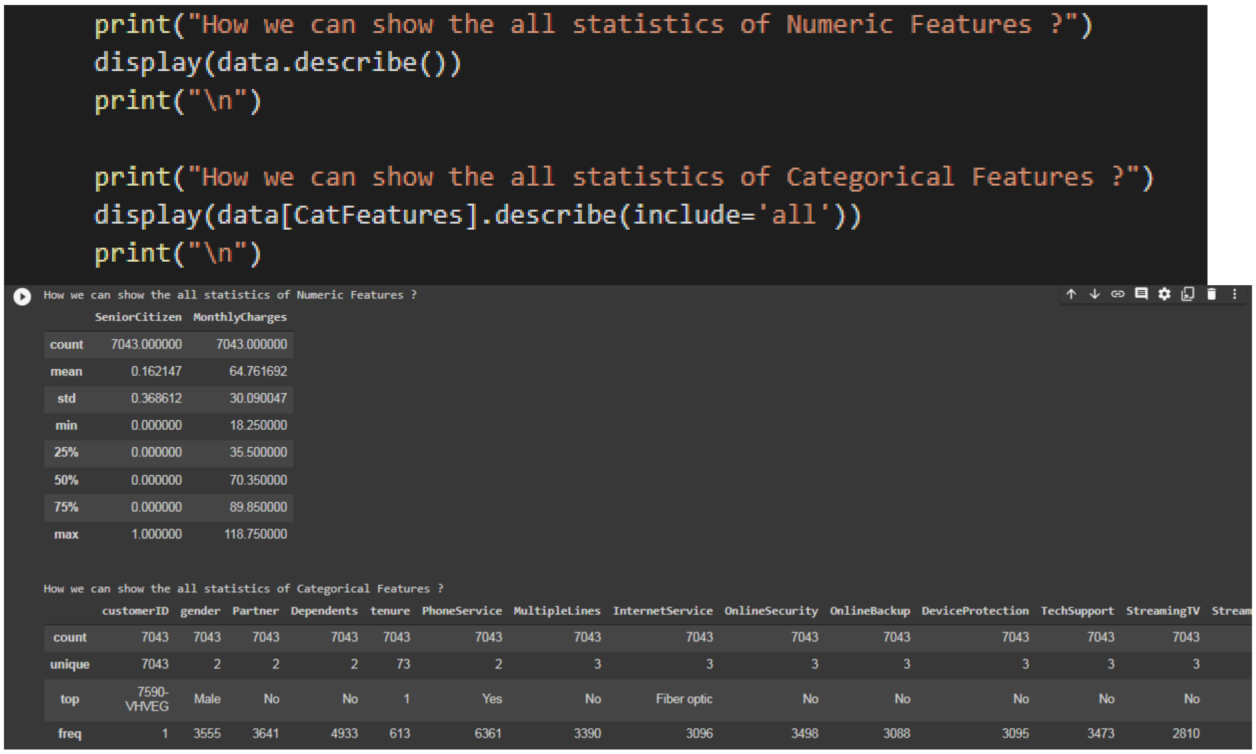
```
[ ]  print("How we can see the type of all columns in data ?")
     display(data.info())
     print("\n")
```

```
<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   object 
 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(1), object(19)
memory usage: 1.1+ MB
None
```



```
[ ]  print("How we can change Tenure from int64 to object")
     print("Original Type of tenure :", data["tenure"].dtypes)
     data["tenure"] = data["tenure"].astype(str)
     print("New Type of tenure :", data["tenure"].dtypes)
     print("\n")

     print("How we can extract the categorical and numeric columns ?")
     CatFeatures = [col for col in data.columns if data[col].dtypes in ["object", "bool"]]
     NumFeatures = [col for col in data.columns if data[col].dtypes in ["int64", "float64"]]
     print("Categorical Features :", CatFeatures)
     print("Numeric Features :", NumFeatures)
     print("\n")
```

```
How we can change Tenure from int64 to object
Original Type of tenure : object
New Type of tenure : object


How we can extract the categorical and numeric columns ?
Categorical Features : ['customerID', 'gender', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBa
Numeric Features : ['SeniorCitizen', 'MonthlyCharges']
```



```
print("How we can show the all statistics of Numeric Features ?")
display(data.describe())
print("\n")

print("How we can show the all statistics of Categorical Features ?")
display(data[CatFeatures].describe(include='all'))
print("\n")
```

```
How we can show the all statistics of Numeric Features ?
```

	SeniorCitizen	MonthlyCharges
count	7043.000000	7043.000000
mean	0.162147	64.761692
std	0.368612	30.090047
min	0.000000	18.250000
25%	0.000000	35.500000
50%	0.000000	70.350000
75%	0.000000	89.850000
max	1.000000	118.750000

```
How we can show the all statistics of Categorical Features ?
```

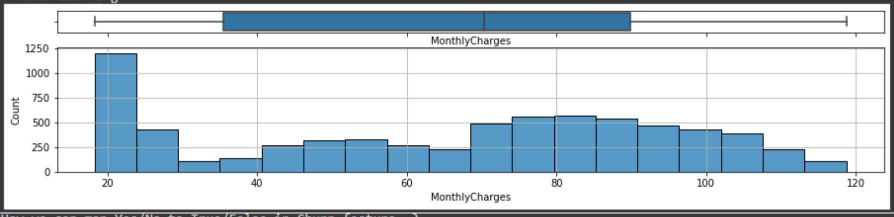
	customerID	gender	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Stream
count	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043	
unique	7043	2	2	2	73	2	3	3	3	3	3	3	3	
top	7590-VHVEG	Male	No	No	1	Yes	No	Fiber optic	No	No	No	No	No	
freq	1	3555	3641	4933	613	6361	3390	3096	3498	3088	3095	3473	2810	

```python
print("How we can get data from describe table ?")
NumStats = data[NumFeatures].describe(include='all')
CatStats = data[CatFeatures].describe(include='all')
MonthlyCharges_50 = NumStats.loc["50%", "MonthlyCharges"]
Churn_top_freq = CatStats.loc[["top", "freq"], "Churn"]
print("MonthlyCharges at 50 %(median) : \n", MonthlyCharges_50)
print("Top and Frequency of Top in Churn : \n", Churn_top_freq)
```

```
How we can get data from describe table ?
MonthlyCharges at 50 %(median) :
 70.35
Top and Frequency of Top in Churn :
 top       No
freq    5174
Name: Churn, dtype: object
```

```python
print("How we can draw chart for a numeric features ?")
feature = "MonthlyCharges"
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15, .85)})
f.set_figheight(3)
f.set_figwidth(15)
sns.boxplot(data[feature], ax=ax_box)
sns.histplot(data=data, x=feature, ax=ax_hist)
plt.grid()
plt.show()

print("How we can map Yes/No to True/False in Churn feature  ?")
MapDict = {"Yes" : True, "No" : False}
data["Churn_Or_Not"] = data["Churn"].map(MapDict)
display(data.head())
```

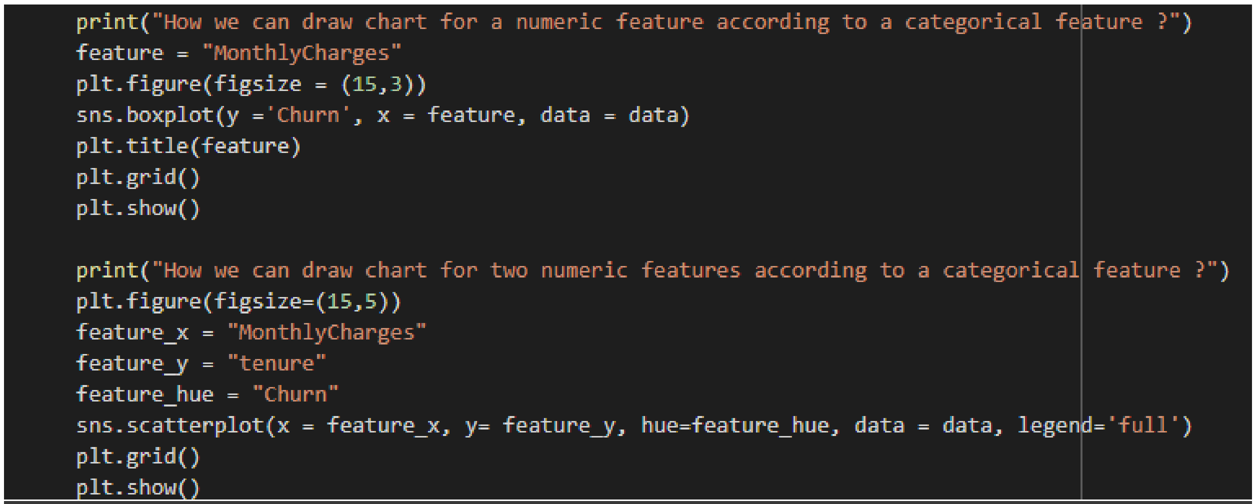
```
How we can draw chart for a numeric features ?
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only v
  FutureWarning
```



```
How we can map Yes/No to True/False in Churn feature  ?
```

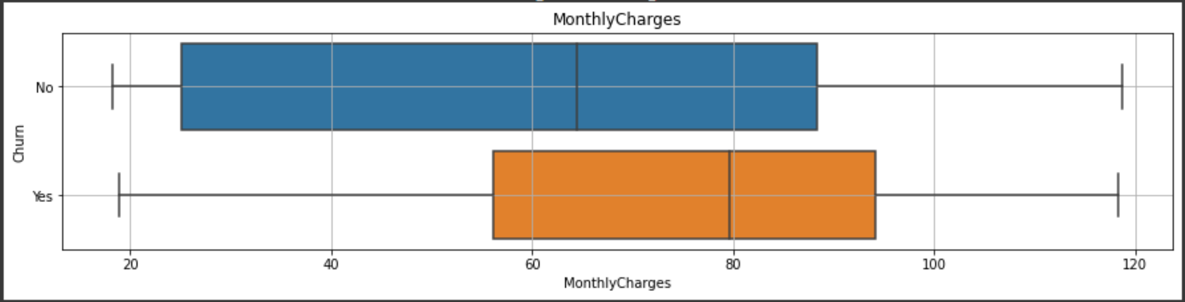
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	No	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No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

5 rows × 22 columns

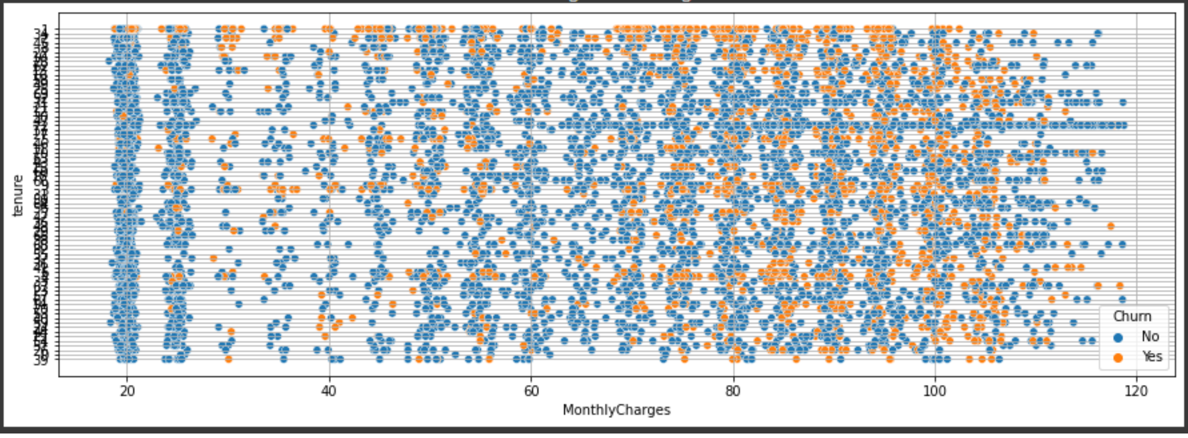


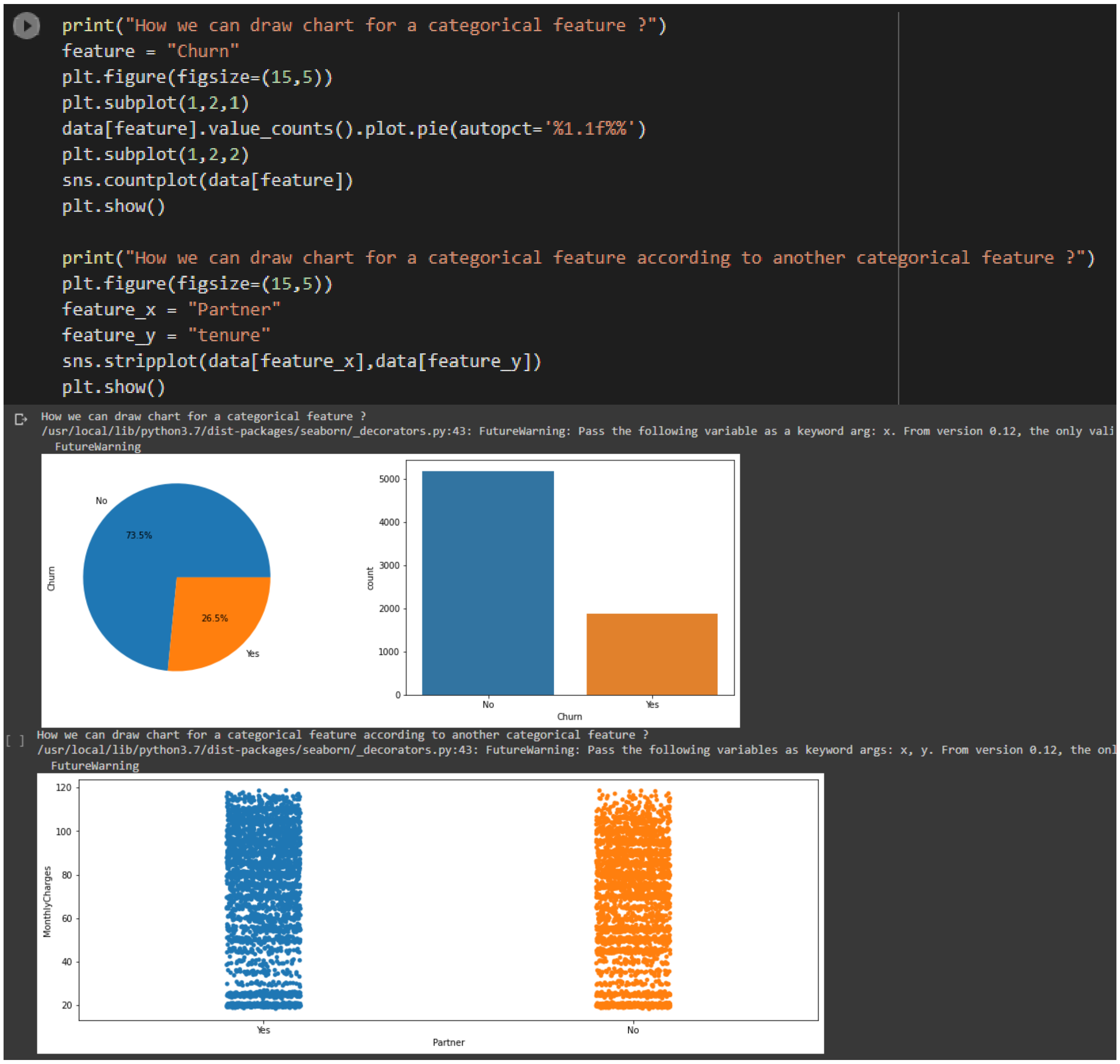
```
print("How we can draw chart for a numeric feature according to a categorical feature ?")
feature = "MonthlyCharges"
plt.figure(figsize = (15,3))
sns.boxplot(y ='Churn', x = feature, data = data)
plt.title(feature)
plt.grid()
plt.show()

print("How we can draw chart for two numeric features according to a categorical feature ?")
plt.figure(figsize=(15,5))
feature_x = "MonthlyCharges"
feature_y = "tenure"
feature_hue = "Churn"
sns.scatterplot(x = feature_x, y= feature_y, hue=feature_hue, data = data, legend='full')
plt.grid()
plt.show()
```

```
How we can draw chart for a numeric feature according to a categorical feature ?
```



```
How we can draw chart for two numeric features according to a categorical feature ?
```

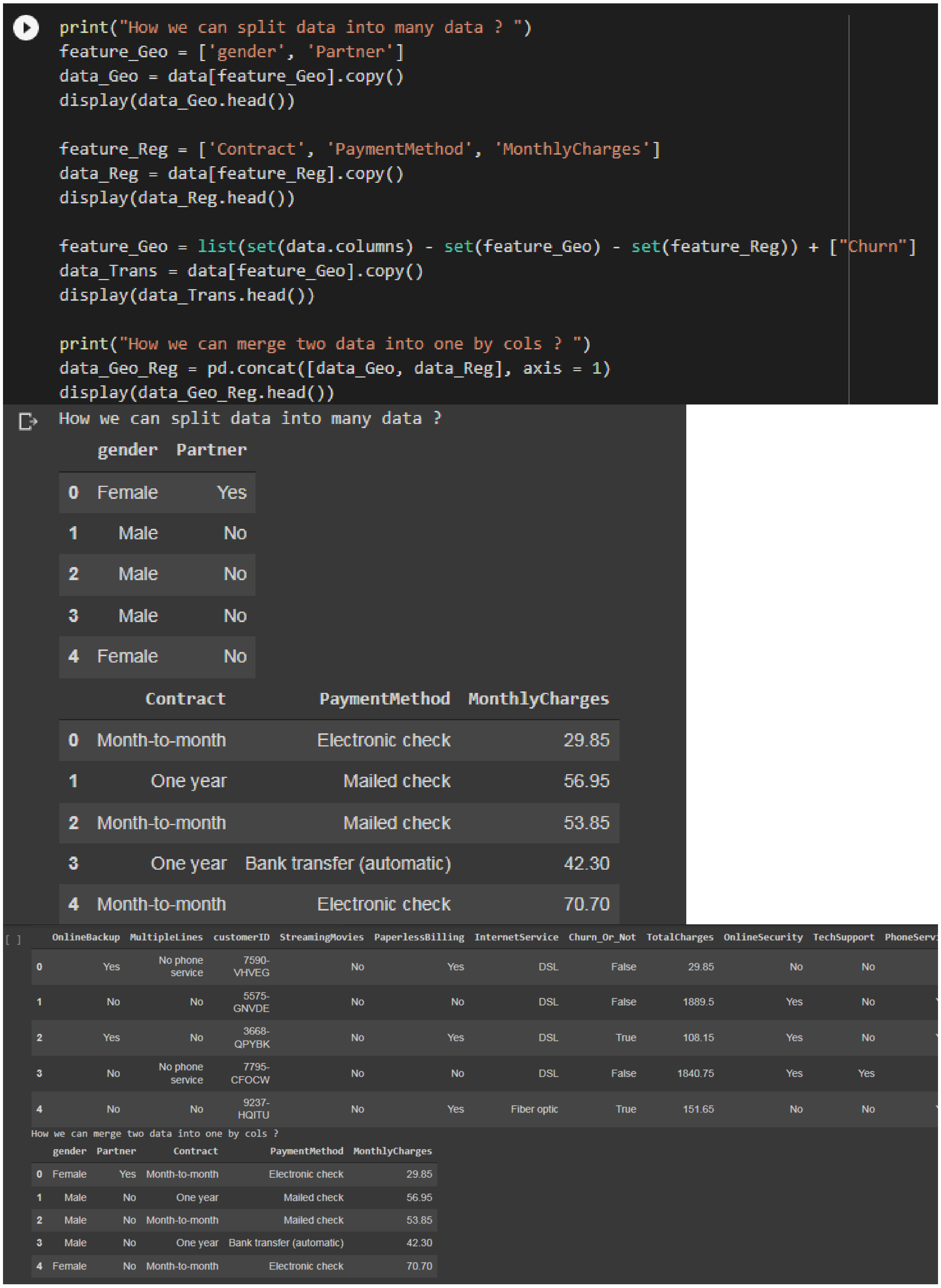




```python
print("How we can draw chart for a categorical feature ?")
feature = "Churn"
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
data[feature].value_counts().plot.pie(autopct='%1.1f%%')
plt.subplot(1,2,2)
sns.countplot(data[feature])
plt.show()

print("How we can draw chart for a categorical feature according to another categorical feature ?")
plt.figure(figsize=(15,5))
feature_x = "Partner"
feature_y = "tenure"
sns.stripplot(data[feature_x],data[feature_y])
plt.show()
```

```
How we can draw chart for a categorical feature ?
/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 vali
  FutureWarning
```

```
How we can draw chart for a categorical feature according to another categorical feature ?
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the onl
  FutureWarning
```



```python
print("How we can split data into many data ? ")
feature_Geo = ['gender', 'Partner']
data_Geo = data[feature_Geo].copy()
display(data_Geo.head())

feature_Reg = ['Contract', 'PaymentMethod', 'MonthlyCharges']
data_Reg = data[feature_Reg].copy()
display(data_Reg.head())

feature_Geo = list(set(data.columns) - set(feature_Geo) - set(feature_Reg)) + ["Churn"]
data_Trans = data[feature_Geo].copy()
display(data_Trans.head())

print("How we can merge two data into one by cols ? ")
data_Geo_Reg = pd.concat([data_Geo, data_Reg], axis = 1)
display(data_Geo_Reg.head())
```

```
How we can split data into many data ?
```

	gender	Partner
0	Female	Yes
1	Male	No
2	Male	No
3	Male	No
4	Female	No

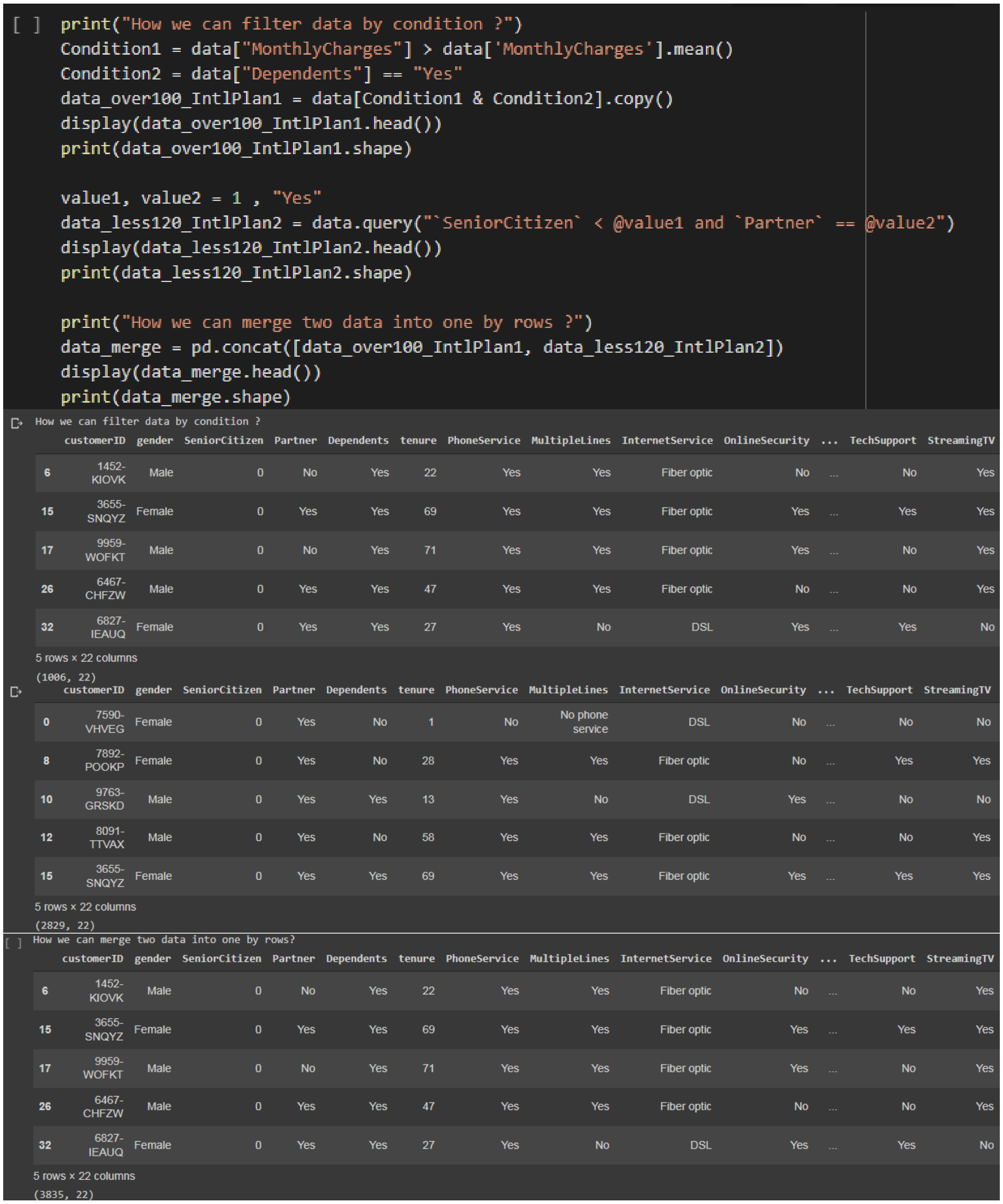
	Contract	PaymentMethod	MonthlyCharges
0	Month-to-month	Electronic check	29.85
1	One year	Mailed check	56.95
2	Month-to-month	Mailed check	53.85
3	One year	Bank transfer (automatic)	42.30
4	Month-to-month	Electronic check	70.70

[ ]

	OnlineBackup	MultipleLines	customerID	StreamingMovies	PaperlessBilling	InternetService	Churn_Or_Not	TotalCharges	OnlineSecurity	TechSupport	PhoneServi
0	Yes	No phone service	7590-VHVEG	No	Yes	DSL	False	29.85	No	No	
1	No	No	5575-GNVDE	No	No	DSL	False	1889.5	Yes	No	
2	Yes	No	3668-QPYBK	No	Yes	DSL	True	108.15	Yes	No	
3	No	No phone service	7795-CFOCW	No	No	DSL	False	1840.75	Yes	Yes	
4	No	No	9237-HQITU	No	Yes	Fiber optic	True	151.65	No	No	

```
How we can merge two data into one by cols ?
```

	gender	Partner	Contract	PaymentMethod	MonthlyCharges
0	Female	Yes	Month-to-month	Electronic check	29.85
1	Male	No	One year	Mailed check	56.95
2	Male	No	Month-to-month	Mailed check	53.85
3	Male	No	One year	Bank transfer (automatic)	42.30
4	Female	No	Month-to-month	Electronic check	70.70



```
[ ]  print("How we can filter data by condition ?")
     Condition1 = data["MonthlyCharges"] > data['MonthlyCharges'].mean()
     Condition2 = data["Dependents"] == "Yes"
     data_over100_IntlPlan1 = data[Condition1 & Condition2].copy()
     display(data_over100_IntlPlan1.head())
     print(data_over100_IntlPlan1.shape)

     value1, value2 = 1 , "Yes"
     data_less120_IntlPlan2 = data.query("`SeniorCitizen` < @value1 and `Partner` == @value2")
     display(data_less120_IntlPlan2.head())
     print(data_less120_IntlPlan2.shape)

     print("How we can merge two data into one by rows ?")
     data_merge = pd.concat([data_over100_IntlPlan1, data_less120_IntlPlan2])
     display(data_merge.head())
     print(data_merge.shape)
```

```
How we can filter data by condition ?
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	...	No	Yes
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes	Yes
17	9959-WOFKT	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	...	No	Yes
26	6467-CHFZW	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	No	...	No	Yes
32	6827-IEAUQ	Female	0	Yes	Yes	27	Yes	No	DSL	Yes	...	Yes	No

5 rows × 22 columns

```
(1006, 22)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No
8	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	...	Yes	Yes
10	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	...	No	No
12	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	...	No	Yes
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes	Yes

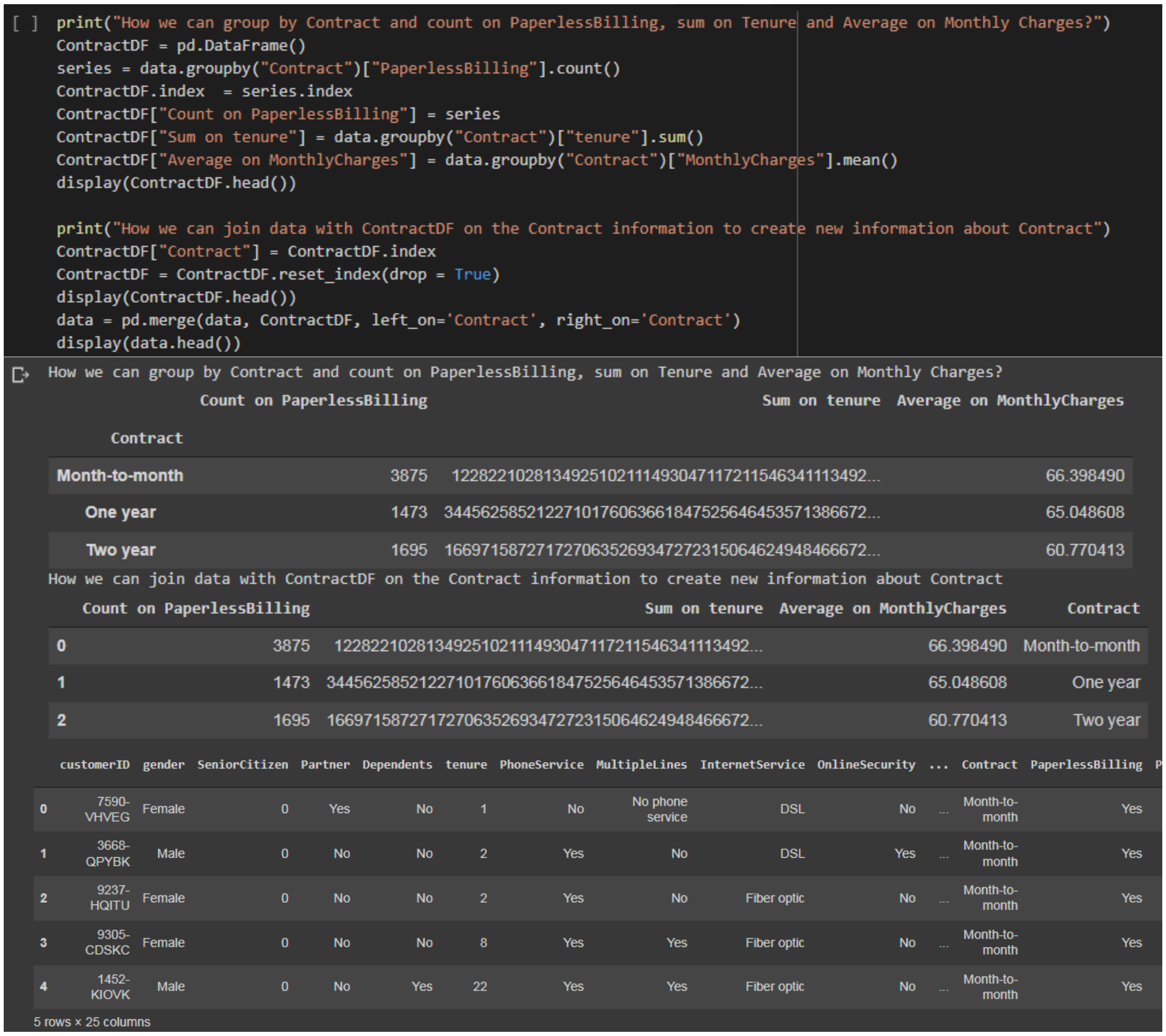
5 rows × 22 columns

```
(2829, 22)
How we can merge two data into one by rows?
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	...	No	Yes
15	3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...	Yes	Yes
17	9959-WOFKT	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	...	No	Yes
26	6467-CHFZW	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	No	...	No	Yes
32	6827-IEAUQ	Female	0	Yes	Yes	27	Yes	No	DSL	Yes	...	Yes	No

5 rows × 22 columns

```
(3835, 22)
```



```
[ ] print("How we can group by Contract and count on PaperlessBilling, sum on Tenure and Average on Monthly Charges?")
    ContractDF = pd.DataFrame()
    series = data.groupby("Contract")["PaperlessBilling"].count()
    ContractDF.index  = series.index
    ContractDF["Count on PaperlessBilling"] = series
    ContractDF["Sum on tenure"] = data.groupby("Contract")["tenure"].sum()
    ContractDF["Average on MonthlyCharges"] = data.groupby("Contract")["MonthlyCharges"].mean()
    display(ContractDF.head())

    print("How we can join data with ContractDF on the Contract information to create new information about Contract")
    ContractDF["Contract"] = ContractDF.index
    ContractDF = ContractDF.reset_index(drop = True)
    display(ContractDF.head())
    data = pd.merge(data, ContractDF, left_on='Contract', right_on='Contract')
    display(data.head())
```

```
How we can group by Contract and count on PaperlessBilling, sum on Tenure and Average on Monthly Charges?
```

	Count on PaperlessBilling	Sum on tenure	Average on MonthlyCharges
Contract			
Month-to-month	3875	1228221028134925102111493047117211546341113492...	66.398490
One year	1473	3445625852122710176063661847525646453571386672...	65.048608
Two year	1695	1669715872717270635269347272315064624948466672...	60.770413

```
How we can join data with ContractDF on the Contract information to create new information about Contract
```

	Count on PaperlessBilling	Sum on tenure	Average on MonthlyCharges	Contract
0	3875	1228221028134925102111493047117211546341113492...	66.398490	Month-to-month
1	1473	3445625852122710176063661847525646453571386672...	65.048608	One year
2	1695	1669715872717270635269347272315064624948466672...	60.770413	Two year

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Contract	PaperlessBilling
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	Month-to-month	Yes
1	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	Month-to-month	Yes
2	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	Month-to-month	Yes
3	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	Month-to-month	Yes
4	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	...	Month-to-month	Yes

5 rows × 25 columns