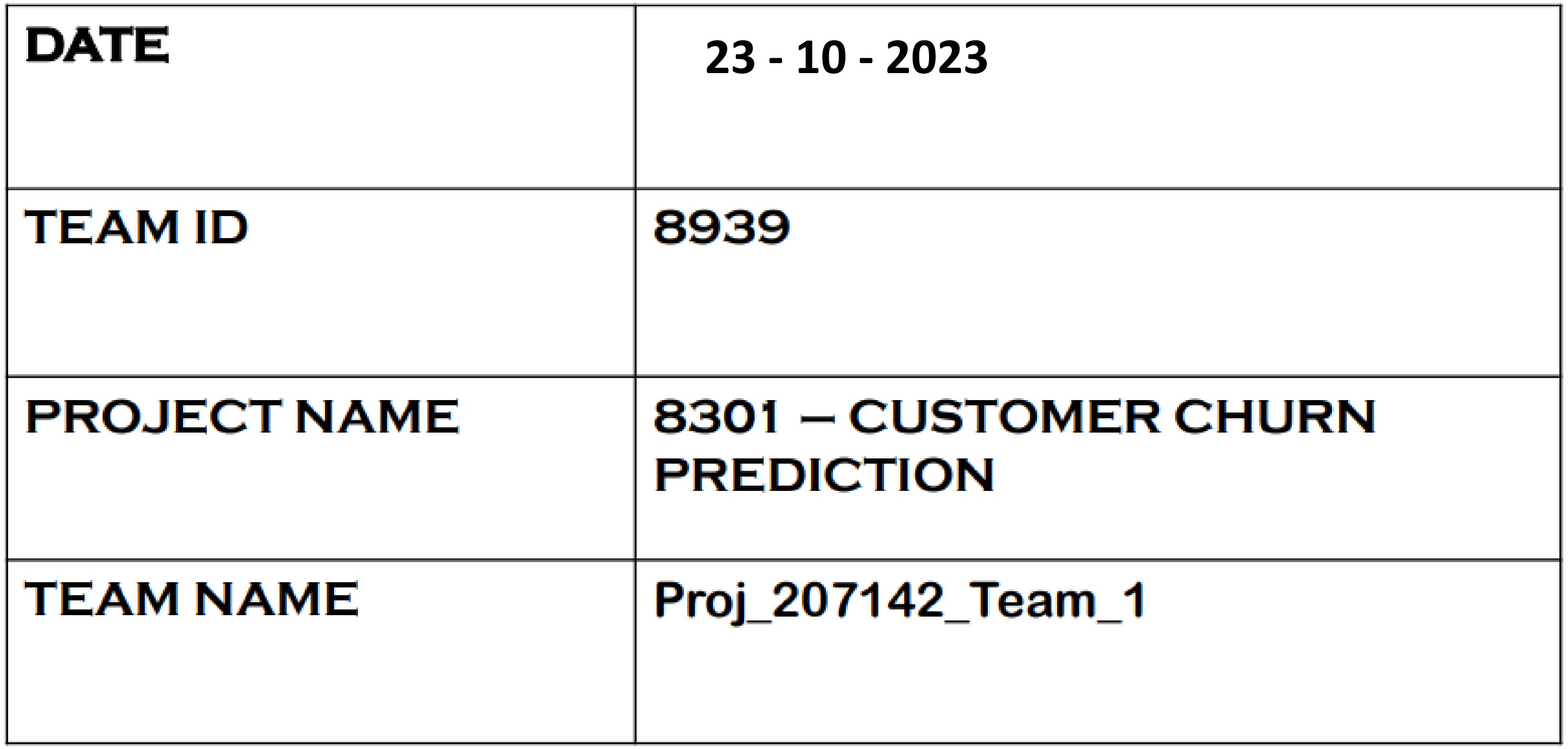
PHASE - 3

Development Part 1



DATE	23 - 10 - 2023
TEAM ID	8939
PROJECT NAME	8301 – CUSTOMER CHURN PREDICTION
TEAM NAME	Proj_207142_Team_1

ANALYTICS OBJECTIVES

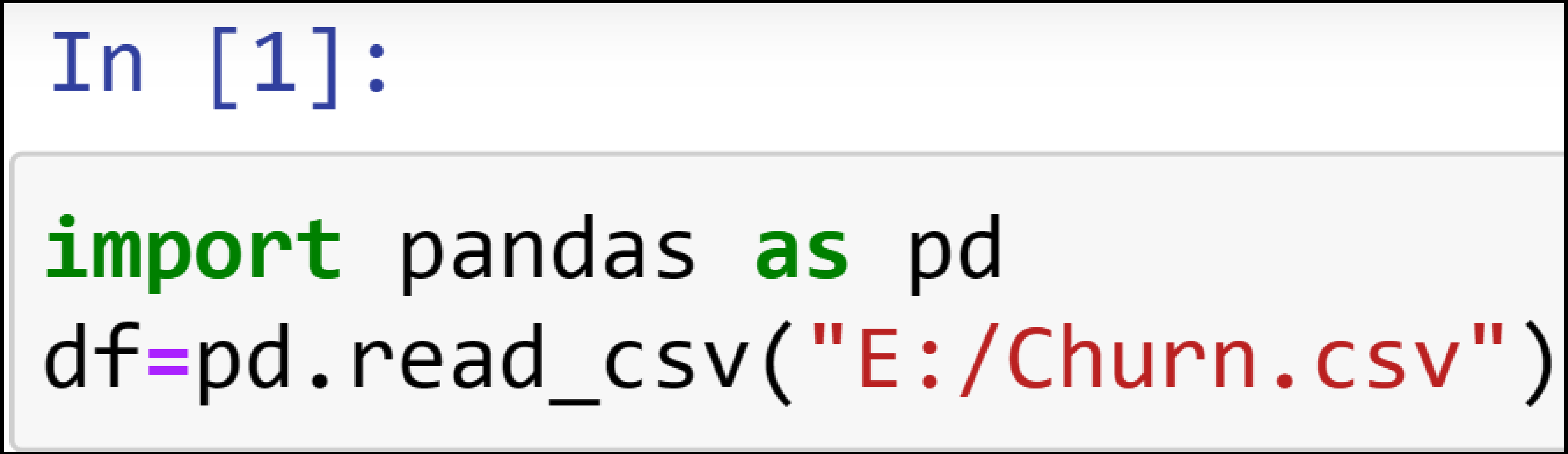
Data Preprocessing :

1.Data Inspection
2.Data Cleaning
3.Data Transformation
4.Data Splitting
5.Data Normalization
6.Data Validation
7.Data Visualization
8.Data Collection

1.Loading Data :

Use pandas.read_csv() to load data from a CSV file.

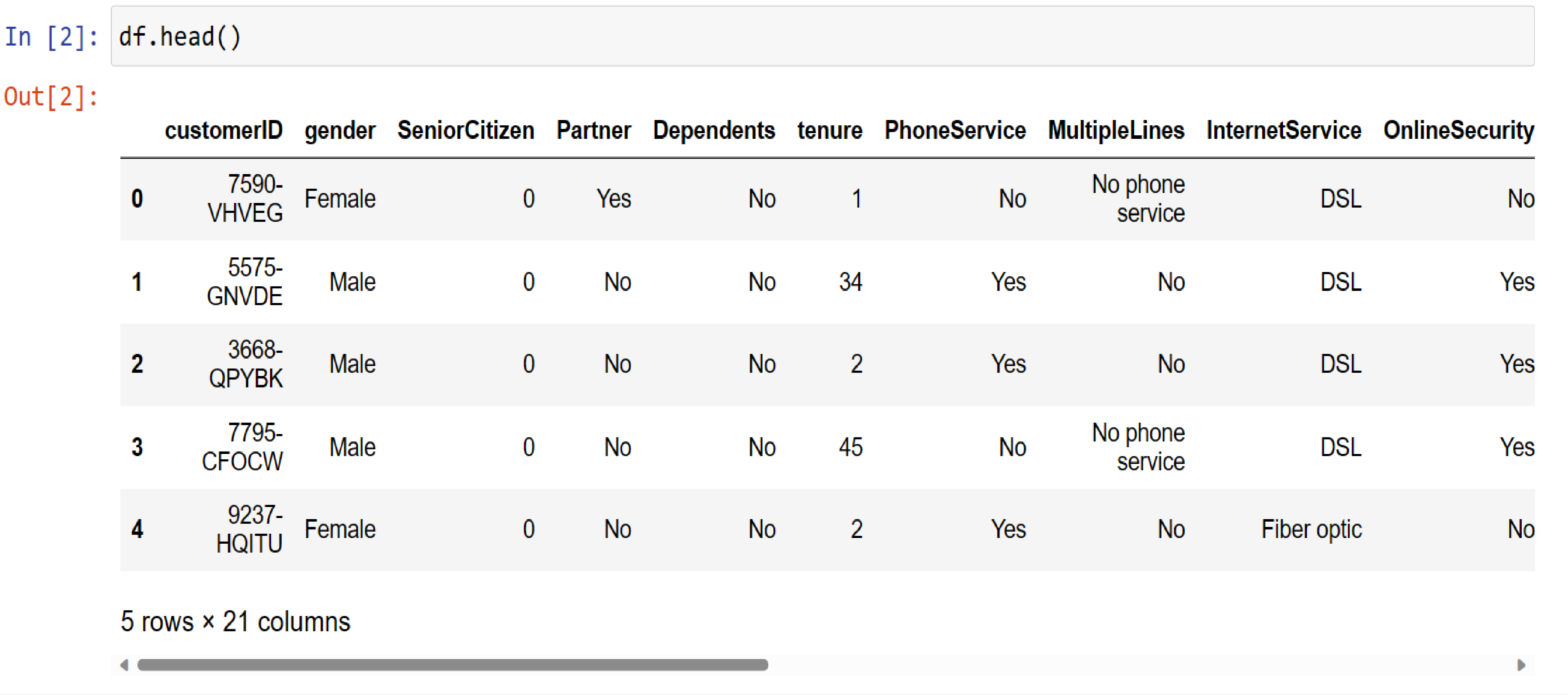
Use pandas.read_excel() for Excel files.



```
In [1]:
import pandas as pd
df=pd.read_csv("E:/Churn.csv")
```

2. Exploring Data:

Use df.head() to view the first few rows of the dataset.



```
In [2]: df.head()
```

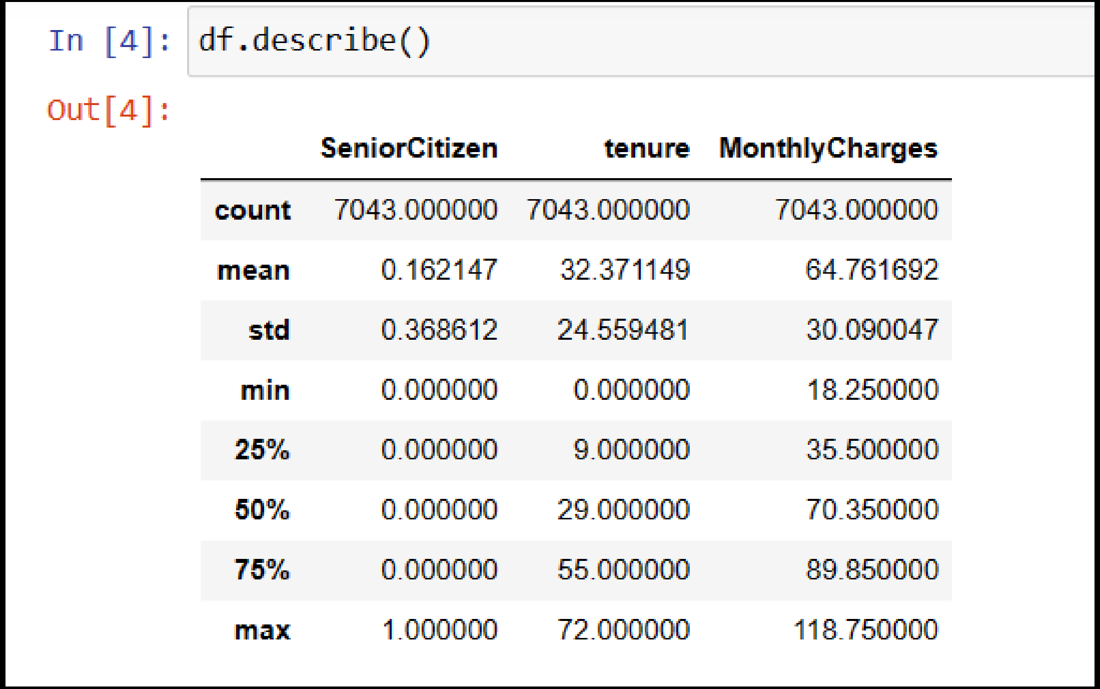
Out[2]:

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

5 rows × 21 columns

Use df.info() to get information about data types and missing values.

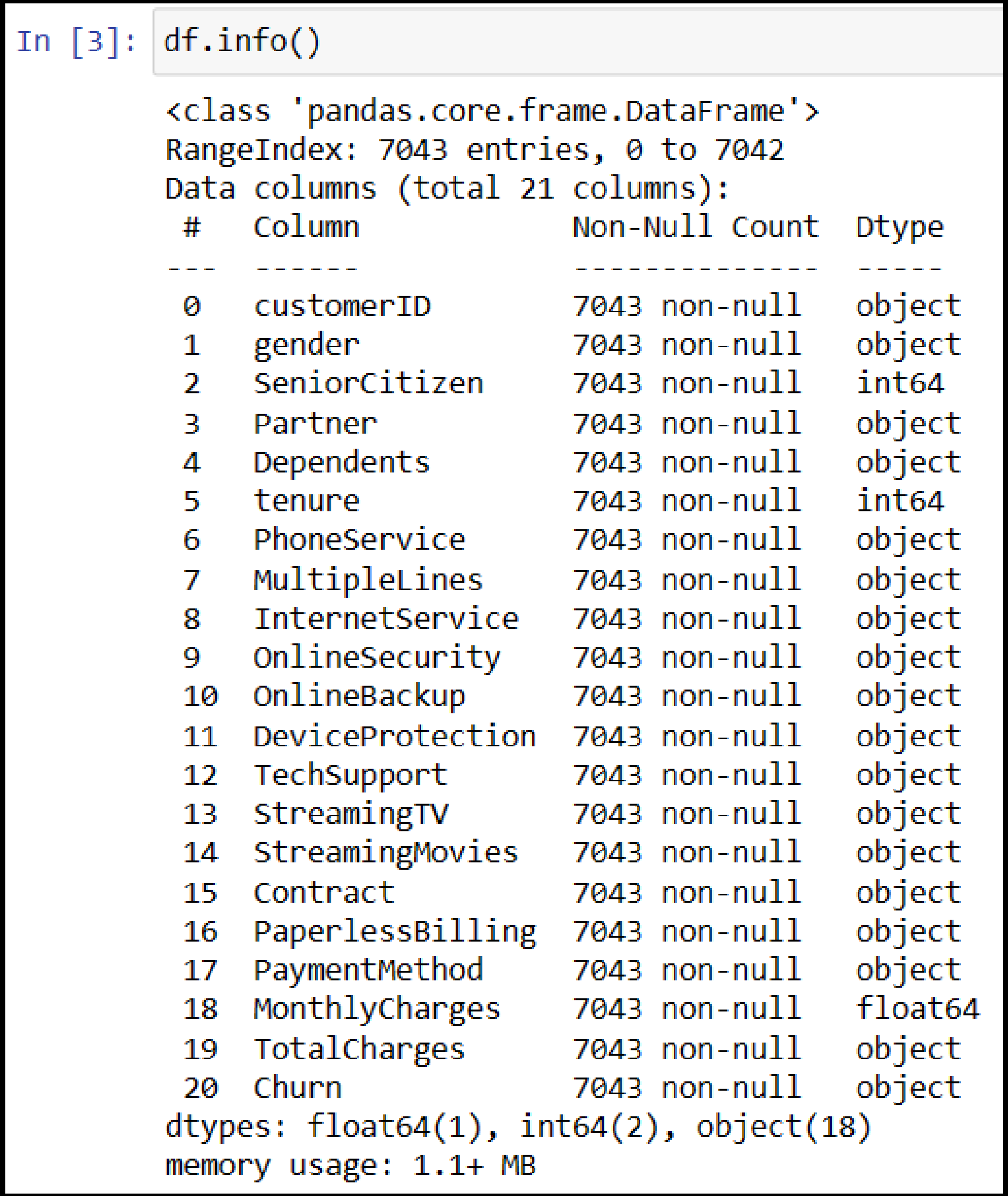
Use df.describe() for summary statistics.python



```
In [4]: df.describe()
```

Out[4]:

	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



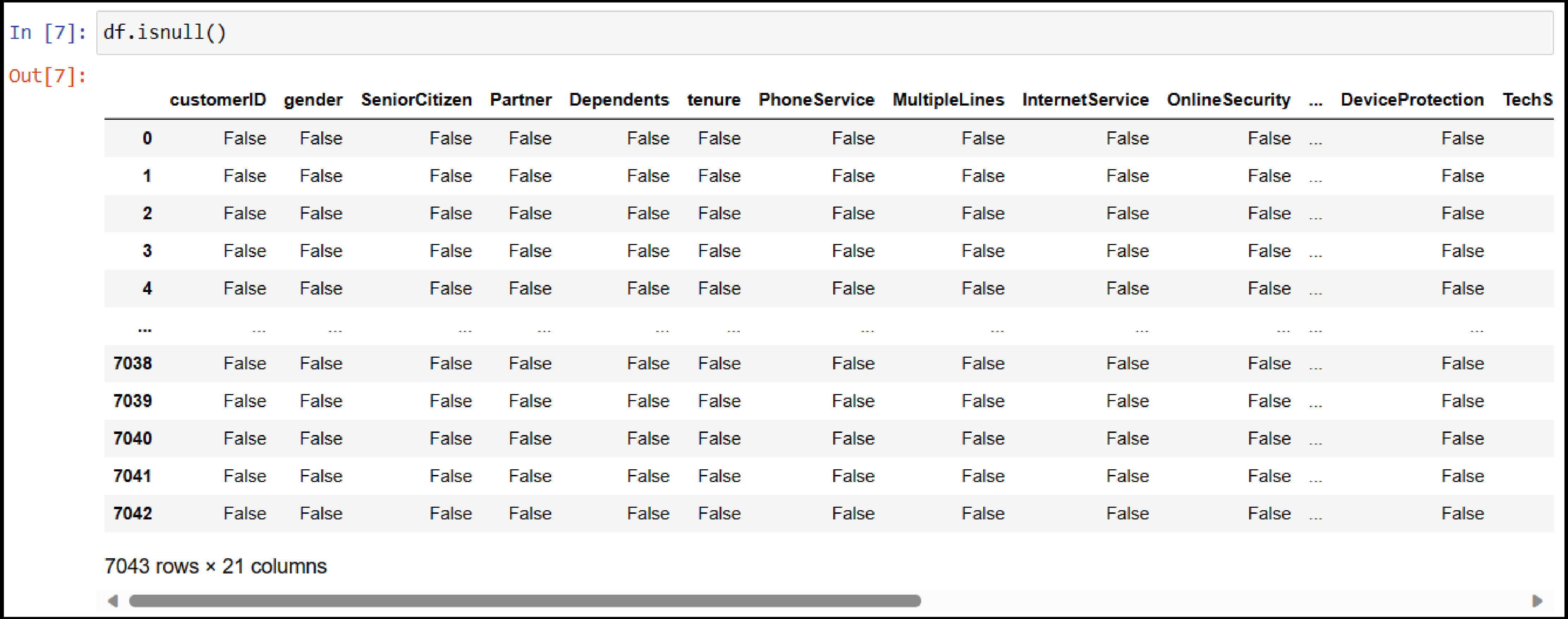
```
In [3]: df.info()

<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     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

3.Handling Missing Values:

Use df.isnull() to identify missing values.

Use df.fillna() or df.dropna() to handle missing values.



In [7]:
```
df.isnull()
```

Out[7]:

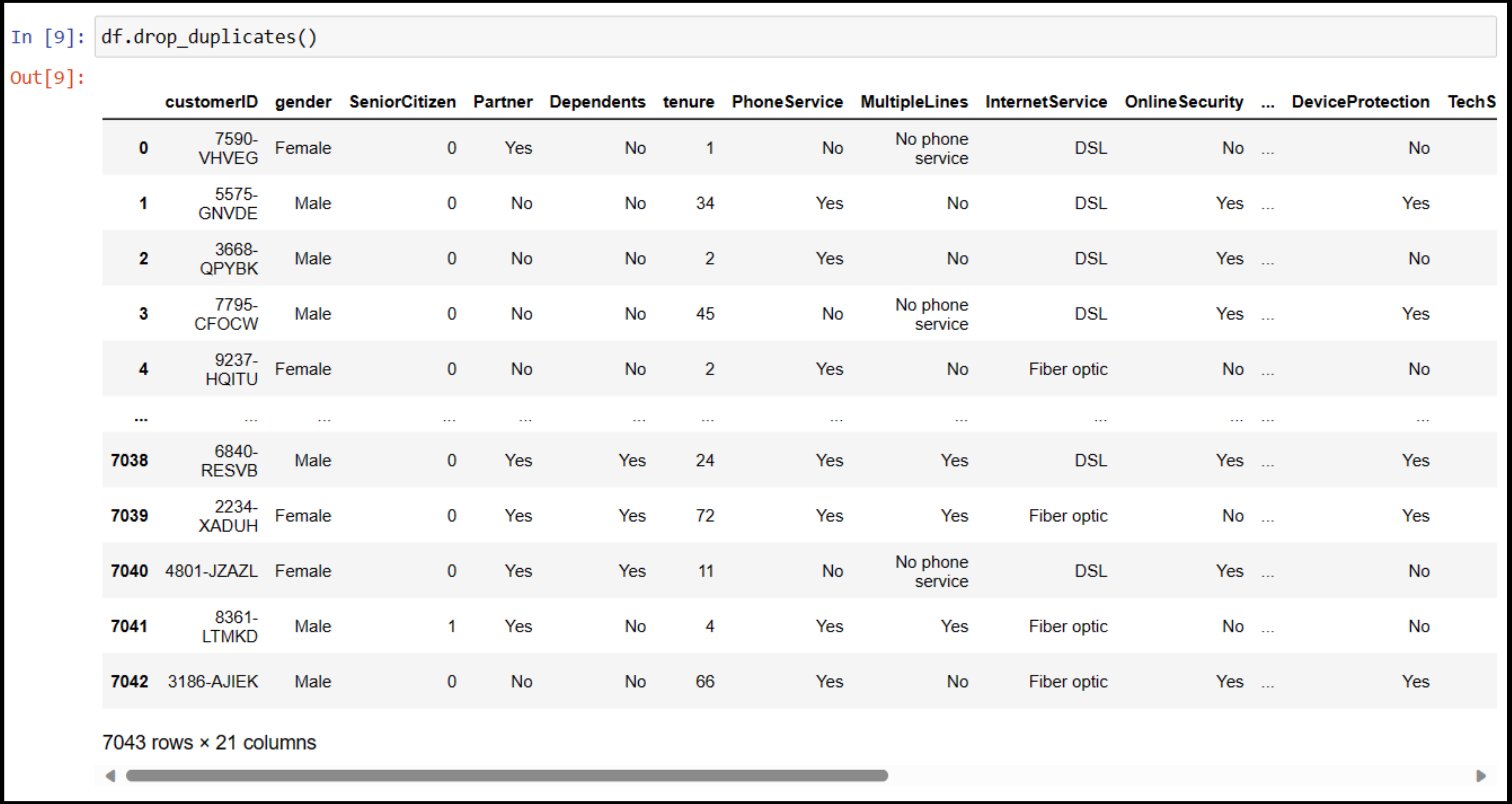
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
0	False	False	False	False	False	False	False	False	False	False	...	False	
1	False	False	False	False	False	False	False	False	False	False	...	False	
2	False	False	False	False	False	False	False	False	False	False	...	False	
3	False	False	False	False	False	False	False	False	False	False	...	False	
4	False	False	False	False	False	False	False	False	False	False	...	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	False	False	False	False	False	False	False	False	False	False	...	False	
7039	False	False	False	False	False	False	False	False	False	False	...	False	
7040	False	False	False	False	False	False	False	False	False	False	...	False	
7041	False	False	False	False	False	False	False	False	False	False	...	False	
7042	False	False	False	False	False	False	False	False	False	False	...	False	

7043 rows × 21 columns

4.Data Cleaning:

Remove duplicate rows with df.drop_duplicates().

Rename columns using df.rename() if necessary.Convert data types with df.astype().



```
In [9]: df.drop_duplicates()
```

Out[9]:

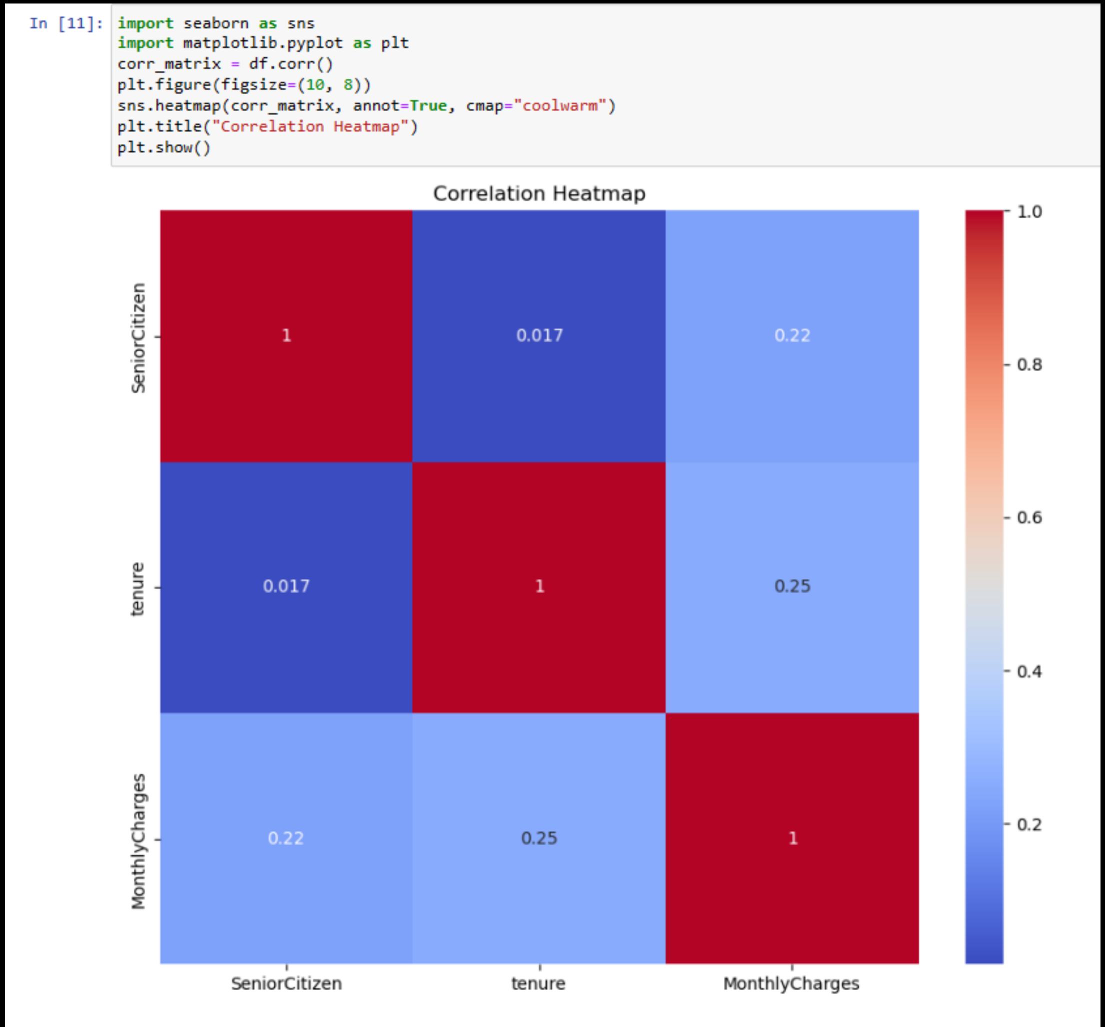
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	

7043 rows × 21 columns

5.Handling Outliers:

Detect and deal with outliers using statistical methods or visualization.

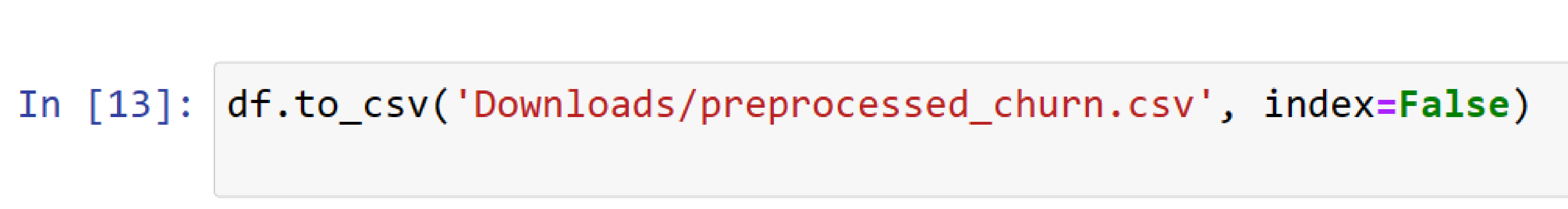
You can use techniques like z-scores or IQR (Interquartile Range).



```
In [11]: import seaborn as sns
         import matplotlib.pyplot as plt
         corr_matrix = df.corr()
         plt.figure(figsize=(10, 8))
         sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
         plt.title("Correlation Heatmap")
         plt.show()
```

6.Saving Data:

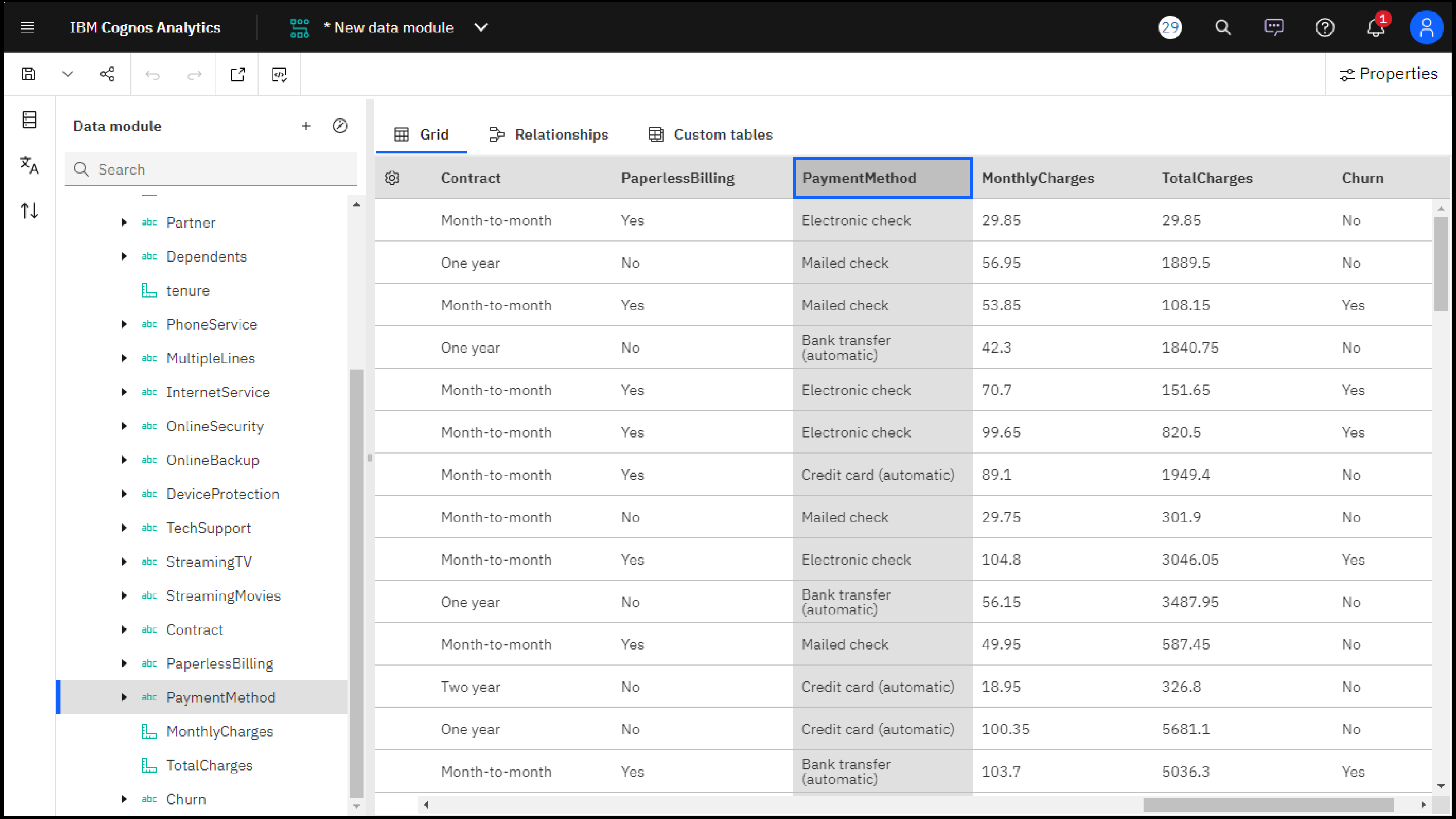
Save the preprocessed data back to a file if needed.



```
In [13]: df.to_csv('Downloads/preprocessed_churn.csv', index=False)
```

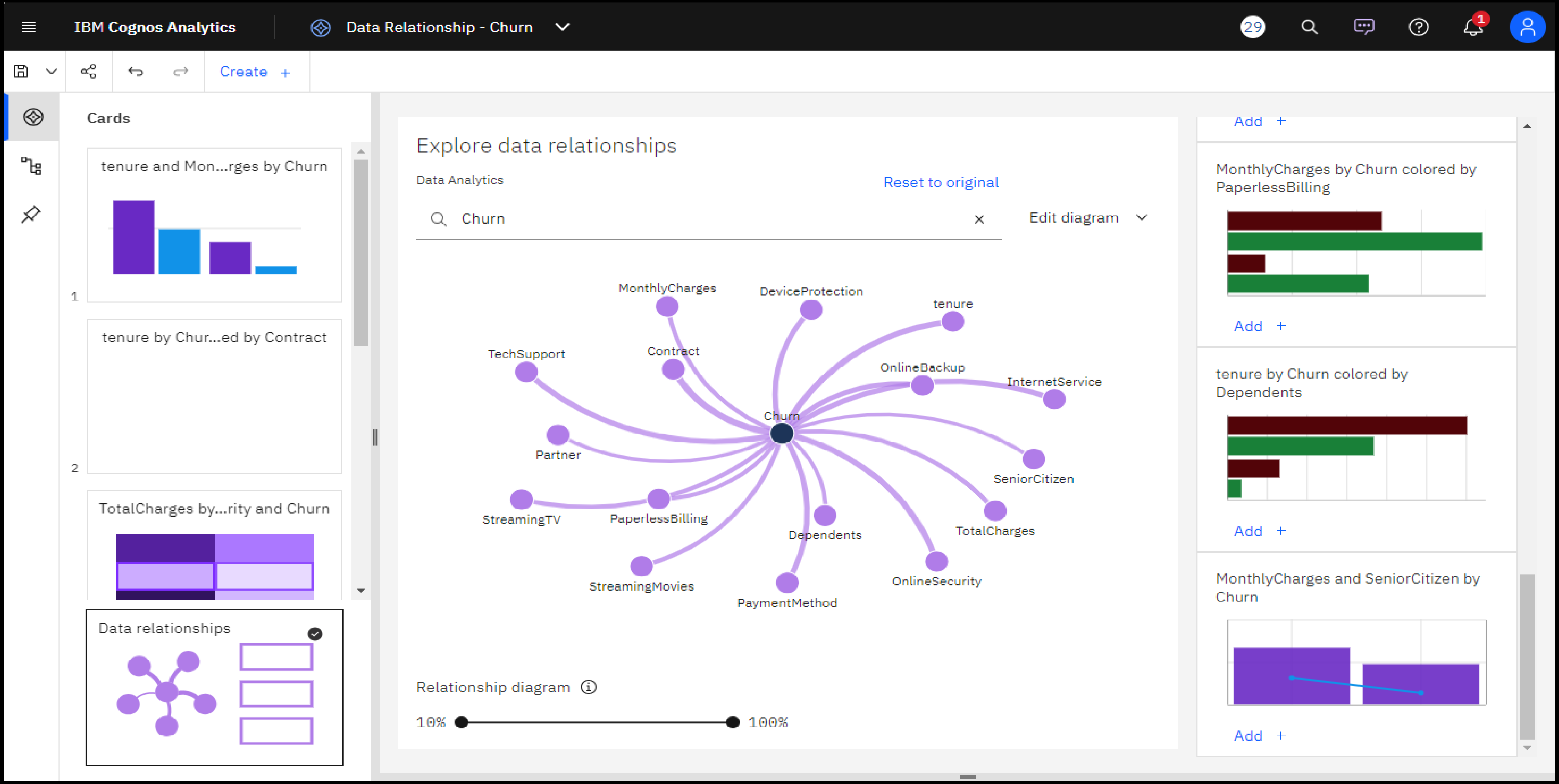
IBM COGNOS ANALYTICS

IN COGNOS - DATA LOADING



Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
Month-to-month	Yes	Electronic check	29.85	29.85	No
One year	No	Mailed check	56.95	1889.5	No
Month-to-month	Yes	Mailed check	53.85	108.15	Yes
One year	No	Bank transfer (automatic)	42.3	1840.75	No
Month-to-month	Yes	Electronic check	70.7	151.65	Yes
Month-to-month	Yes	Electronic check	99.65	820.5	Yes
Month-to-month	Yes	Credit card (automatic)	89.1	1949.4	No
Month-to-month	No	Mailed check	29.75	301.9	No
Month-to-month	Yes	Electronic check	104.8	3046.05	Yes
One year	No	Bank transfer (automatic)	56.15	3487.95	No
Month-to-month	Yes	Mailed check	49.95	587.45	No
Two year	No	Credit card (automatic)	18.95	326.8	No
One year	No	Credit card (automatic)	100.35	5681.1	No
Month-to-month	Yes	Bank transfer (automatic)	103.7	5036.3	Yes

IN COGNOS - DATA RELATIONSHIP



IBM Cognos Analytics

Data Relationship - Churn

Create +

Cards

1 tenure and Mon...rges by Churn

2 tenure by Chur...ed by Contract

TotalCharges by...rity and Churn

Data relationships

Explore data relationships

Data Analytics

Reset to original

Churn

Edit diagram

Relationship diagram

10% 100%

Add +

MonthlyCharges by Churn colored by PaperlessBilling

Add +

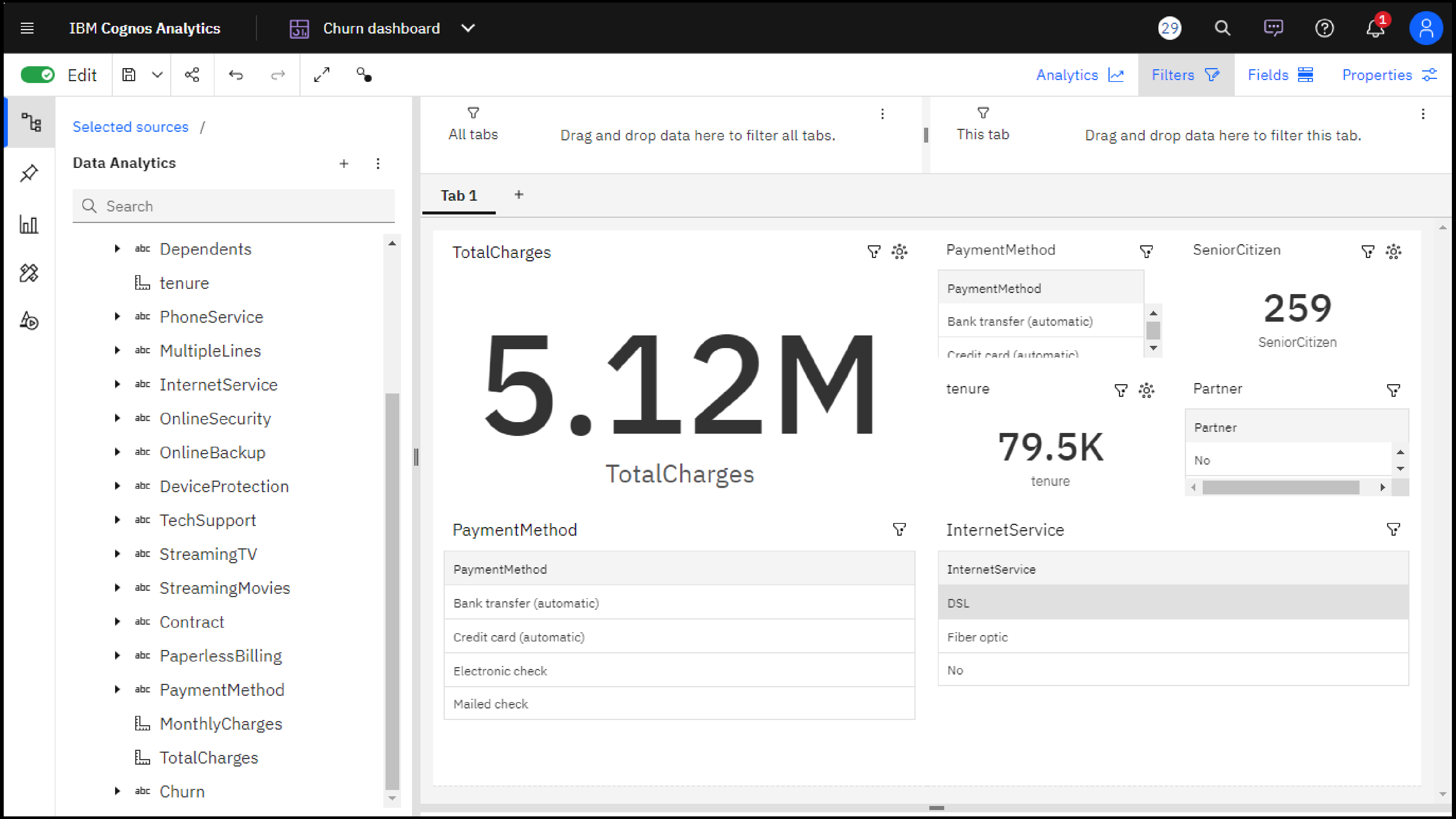
tenure by Churn colored by Dependents

Add +

MonthlyCharges and SeniorCitizen by Churn

Add +

IN COGNOS - CHURN DASHBOARD



All tabs — Drag and drop data here to filter all tabs.

This tab — Drag and drop data here to filter this tab.

PaymentMethod

PaymentMethod
Bank transfer (automatic)
Credit card (automatic)

Partner

Partner
No

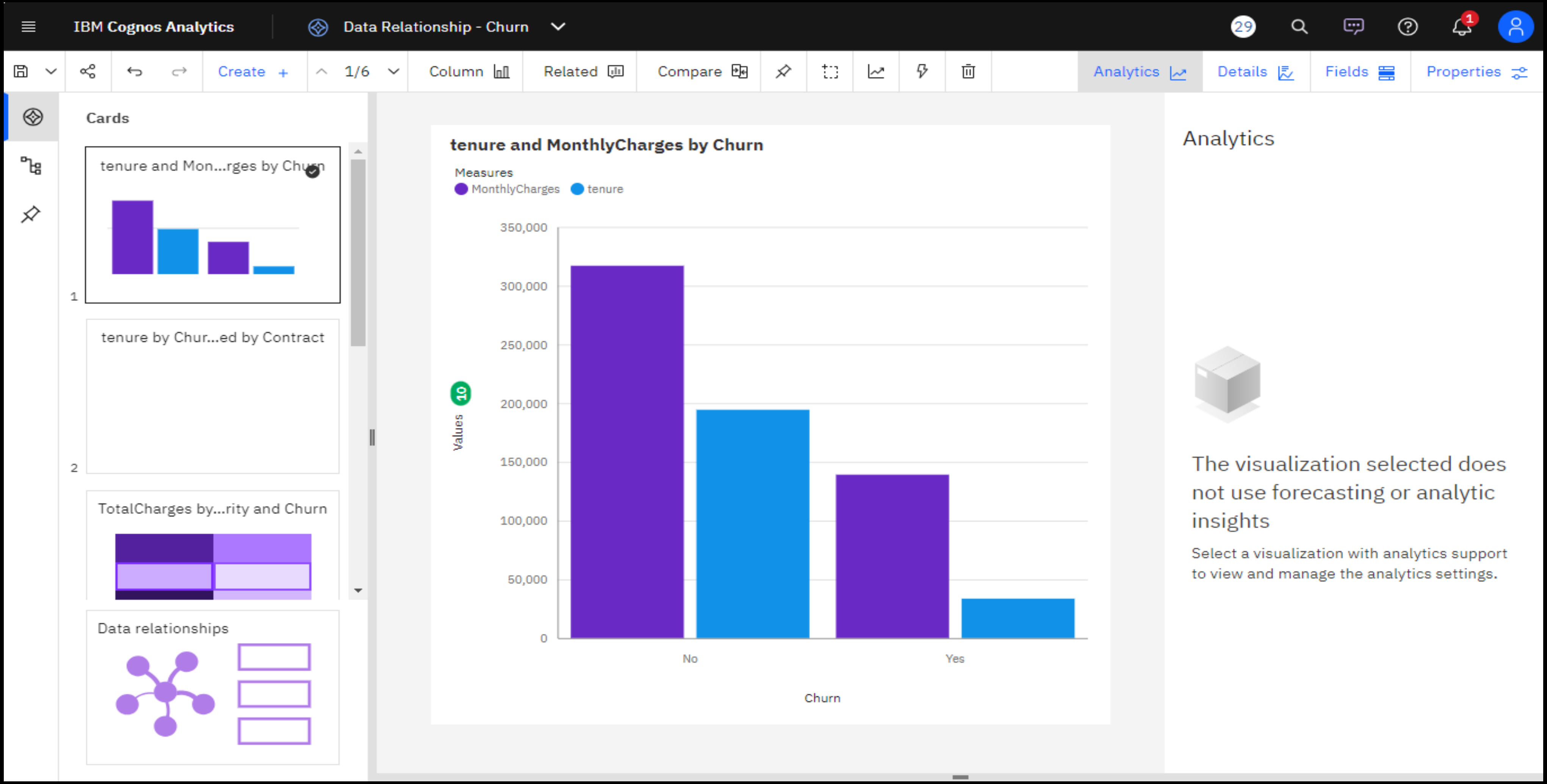
PaymentMethod

PaymentMethod
Bank transfer (automatic)
Credit card (automatic)
Electronic check
Mailed check

InternetService

InternetService
DSL
Fiber optic
No

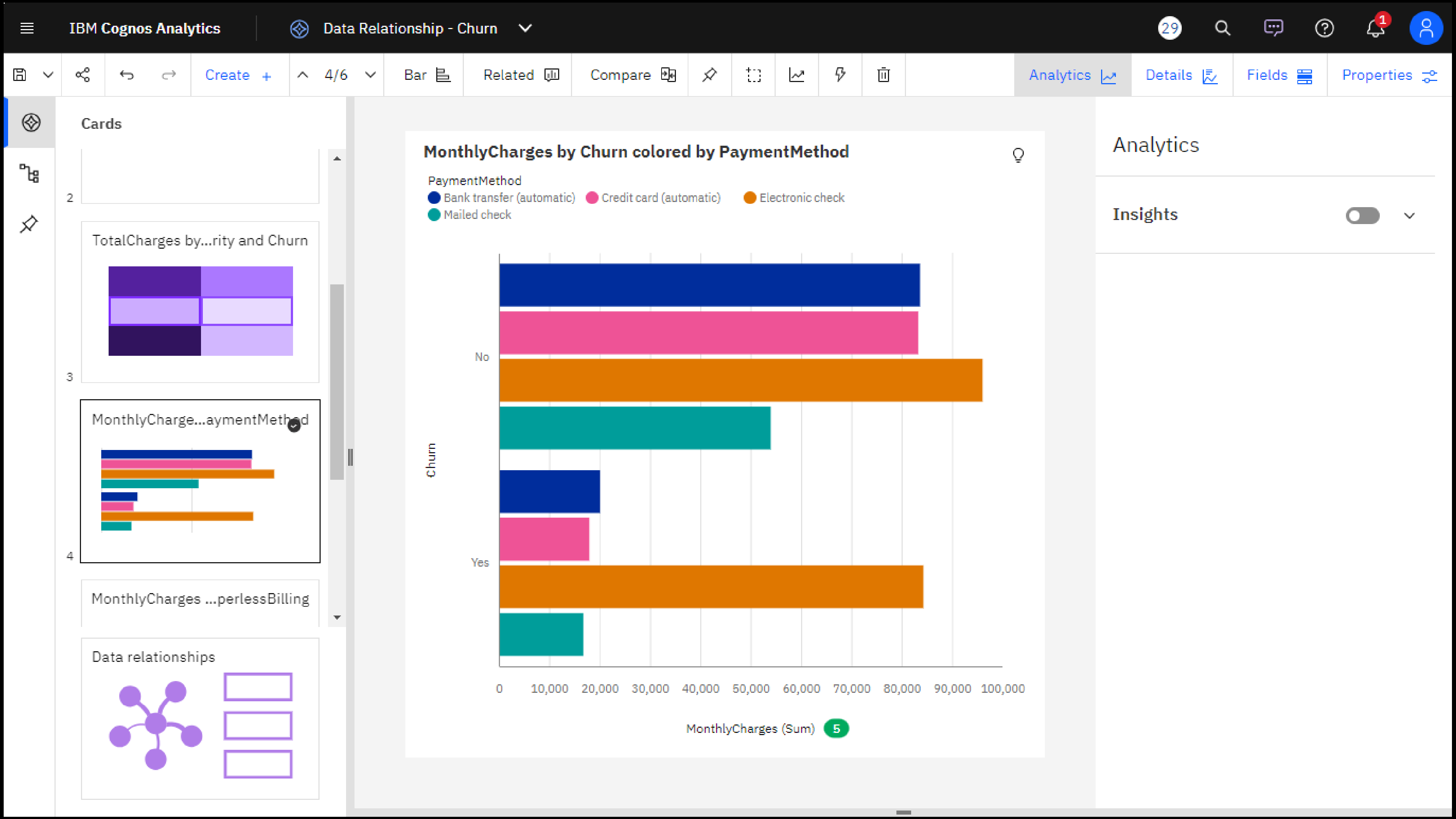
IN COGNOS - TENURE AND MONTHLY CHARGES



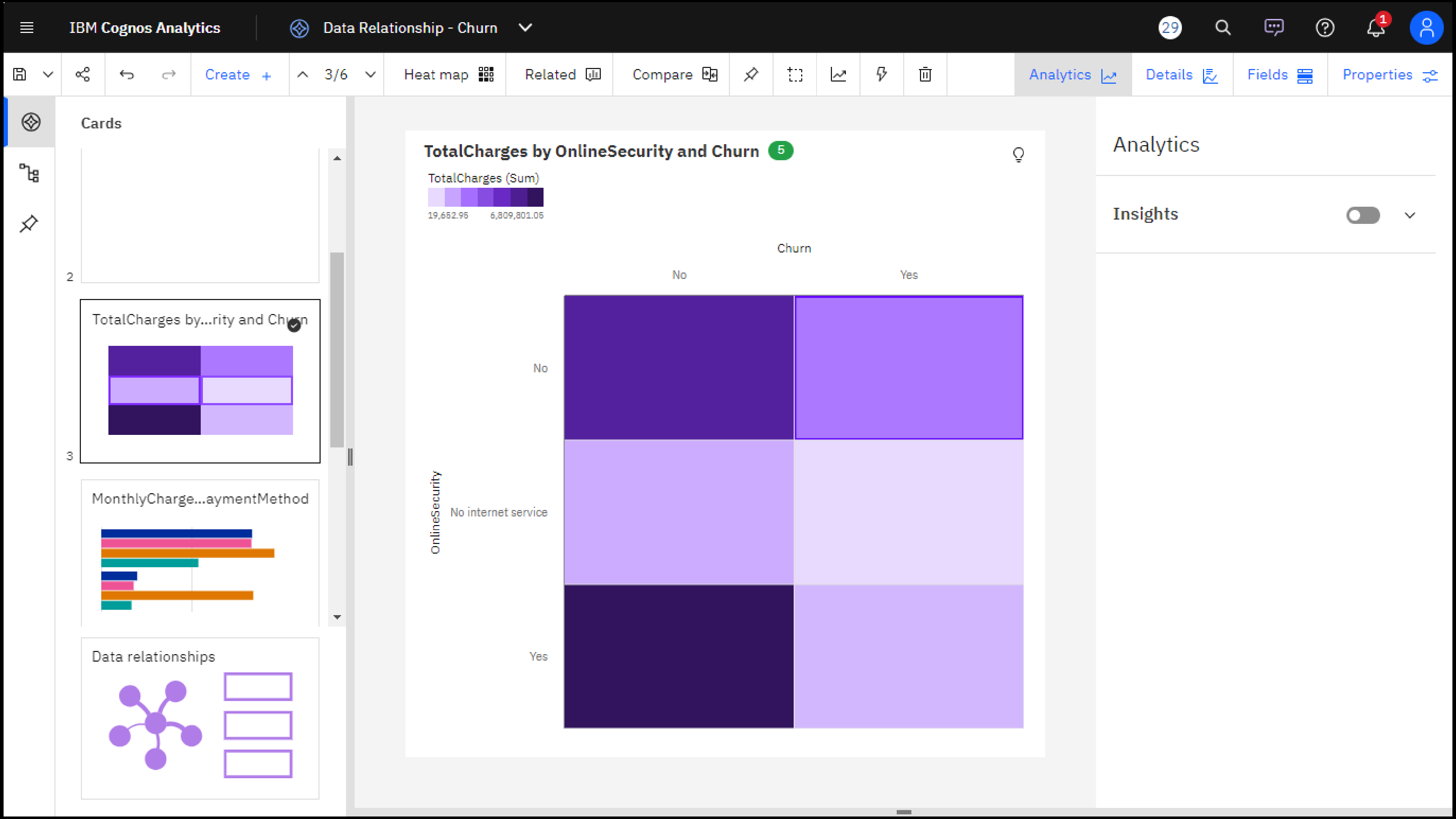
The visualization selected does not use forecasting or analytic insights

Select a visualization with analytics support to view and manage the analytics settings.

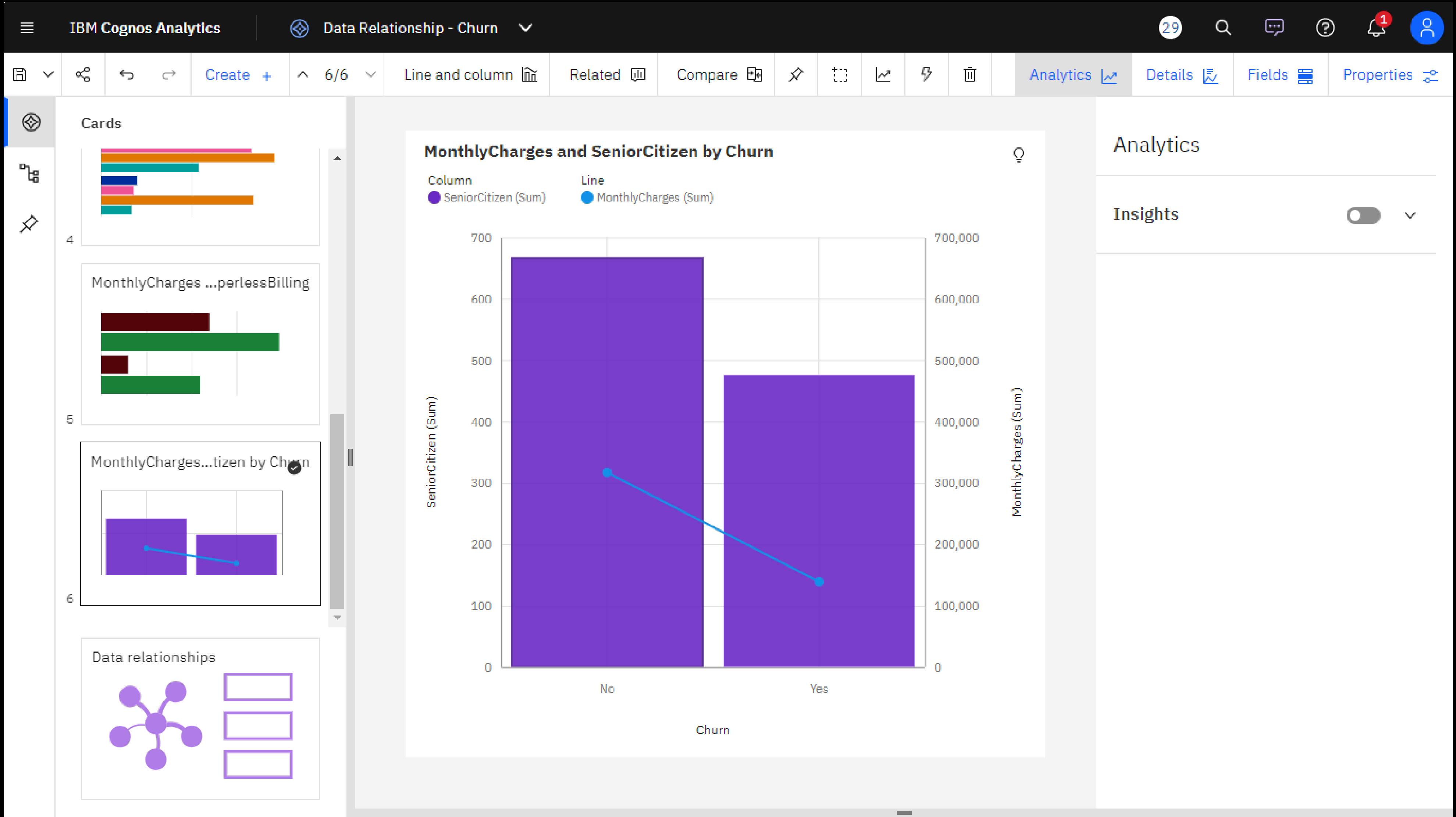
MONTHLY CHARGES BY CHURN COLORED BY PAYMENT METHOD



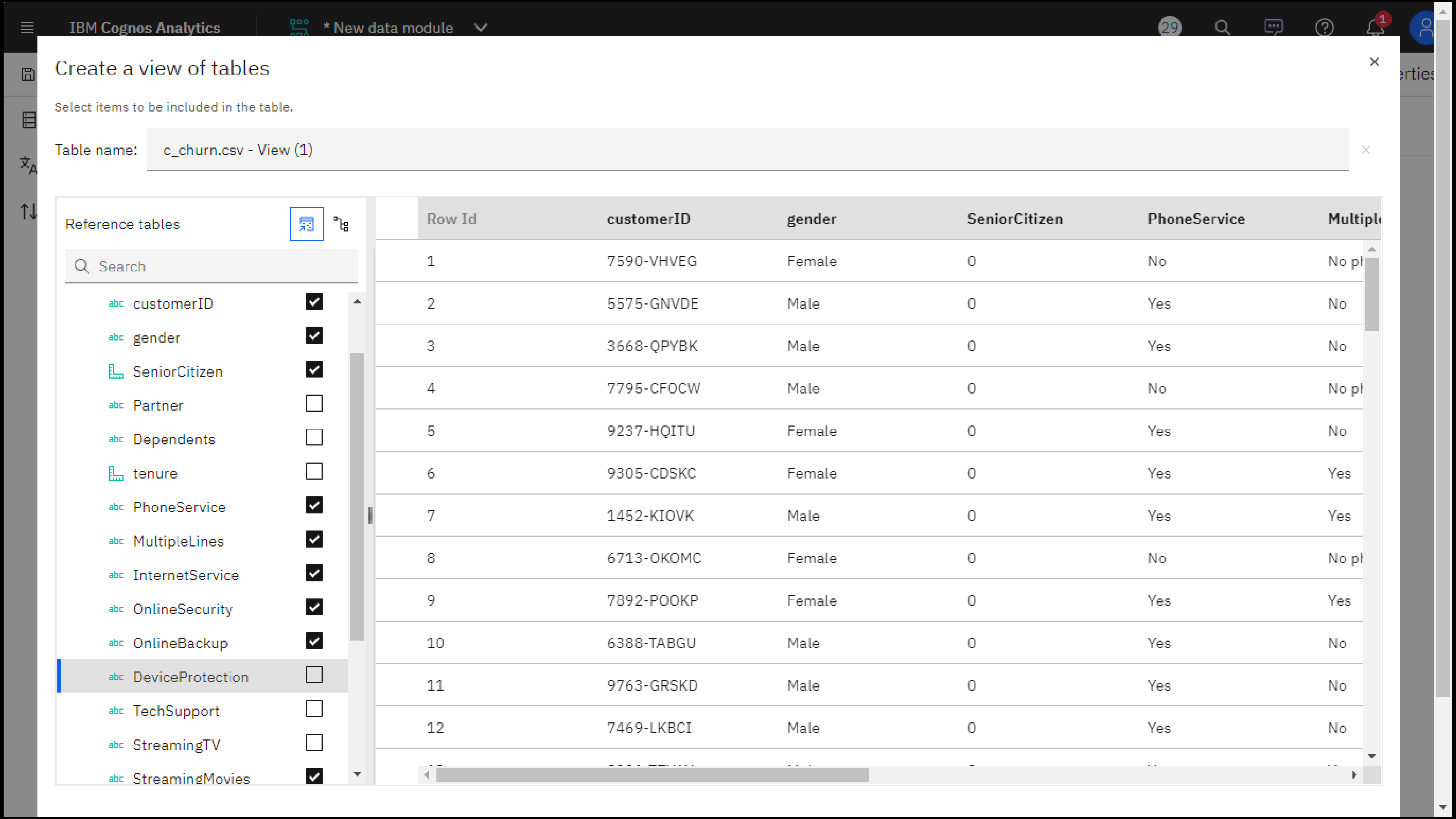
TOTAL CHARGES BY ONLINE SECURITY



SENIOR CITIZER BY CHURN



VIEW OF TABLES



Create a view of tables

Select items to be included in the table.

Table name: c_churn.csv - View (1)

Reference tables

Search

- ☑ customerID
- ☑ gender
- ☑ SeniorCitizen
- ☐ Partner
- ☐ Dependents
- ☐ tenure
- ☑ PhoneService
- ☑ MultipleLines
- ☑ InternetService
- ☑ OnlineSecurity
- ☑ OnlineBackup
- ☐ DeviceProtection
- ☐ TechSupport
- ☐ StreamingTV
- ☑ StreamingMovies

Row Id	customerID	gender	SeniorCitizen	PhoneService	Multipl
1	7590-VHVEG	Female	0	No	No ph
2	5575-GNVDE	Male	0	Yes	No
3	3668-QPYBK	Male	0	Yes	No
4	7795-CFOCW	Male	0	No	No ph
5	9237-HQITU	Female	0	Yes	No
6	9305-CDSKC	Female	0	Yes	Yes
7	1452-KIOVK	Male	0	Yes	Yes
8	6713-OKOMC	Female	0	No	No ph
9	7892-POOKP	Female	0	Yes	Yes
10	6388-TABGU	Male	0	Yes	No
11	9763-GRSKD	Male	0	Yes	No
12	7469-LKBCI	Male	0	Yes	No

CONCLUSION

Data preprocessing is a crucial step in preparing data for analysis and machine learning. It involves collecting, inspecting, cleaning, transforming, and organizing data. The main steps include data collection, inspection, cleaning, transformation, splitting, normalization, and validation. It ensures data is ready for analysis and model training, improving the success of data-related projects.