Phase-2

Data Pre-processing

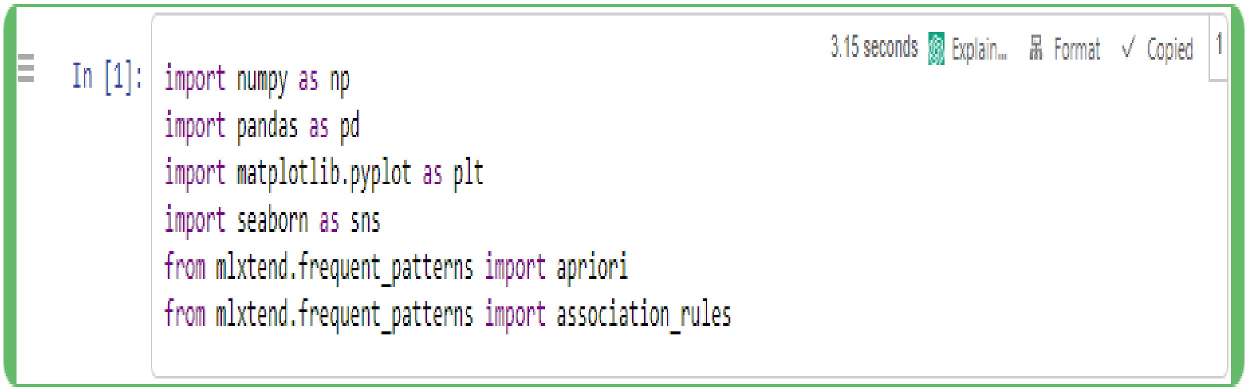
Date	8 October 2023
Team ID	Proj-212168-Team-2
Project Name	Market Basket Insights
Maximum marks	

Data pre-processing refers to the process of cleaning, transforming, and organizing raw data into a format that is suitable for machine learning algorithms and models. Data pre-processing aims to make the data more understandable and valuable for the AI model by addressing issues such as noise, missing values, outliers, and inconsistencies. It is an important step in the data mining process.

Program:

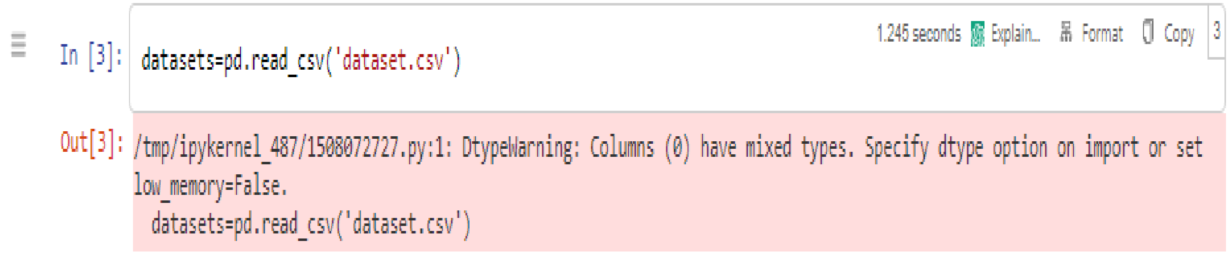
#import packages:

- Numpy :(import numpy as np) a library for mathematical operations and handling arrays.
- Pandas :(import pandas as pd) a library for data manipulation and analysis.
- matplotlib.pyplot: (import as plt) a library for creating visualization.
- Seaborn :as a library for creating additional data visualization.
- mlxtend.frequent_patterns: a module for performing frequent itemset mining and association rule learning.



```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

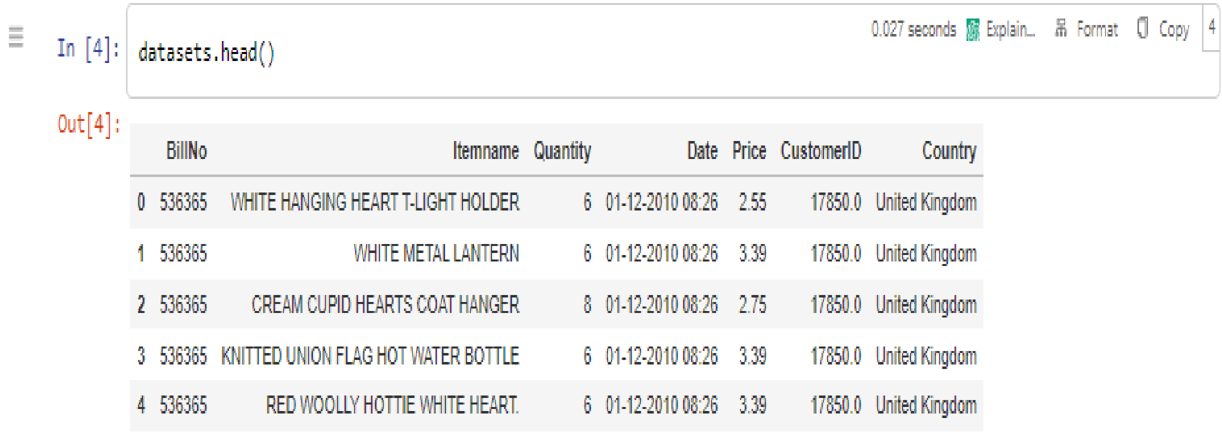
#Load the dataset:



```python
datasets=pd.read_csv('dataset.csv')
```

```
/tmp/ipykernel_487/1508072727.py:1: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set
low_memory=False.
  datasets=pd.read_csv('dataset.csv')
```

This code reads contents of a csv file called "dataset.csv" and saves it a variable called "datasets".

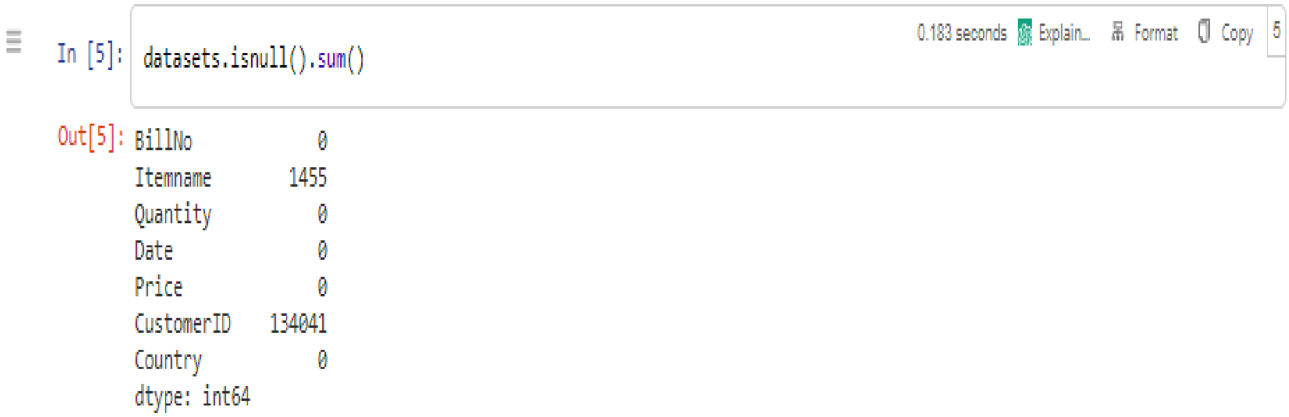


```python
datasets.head()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom

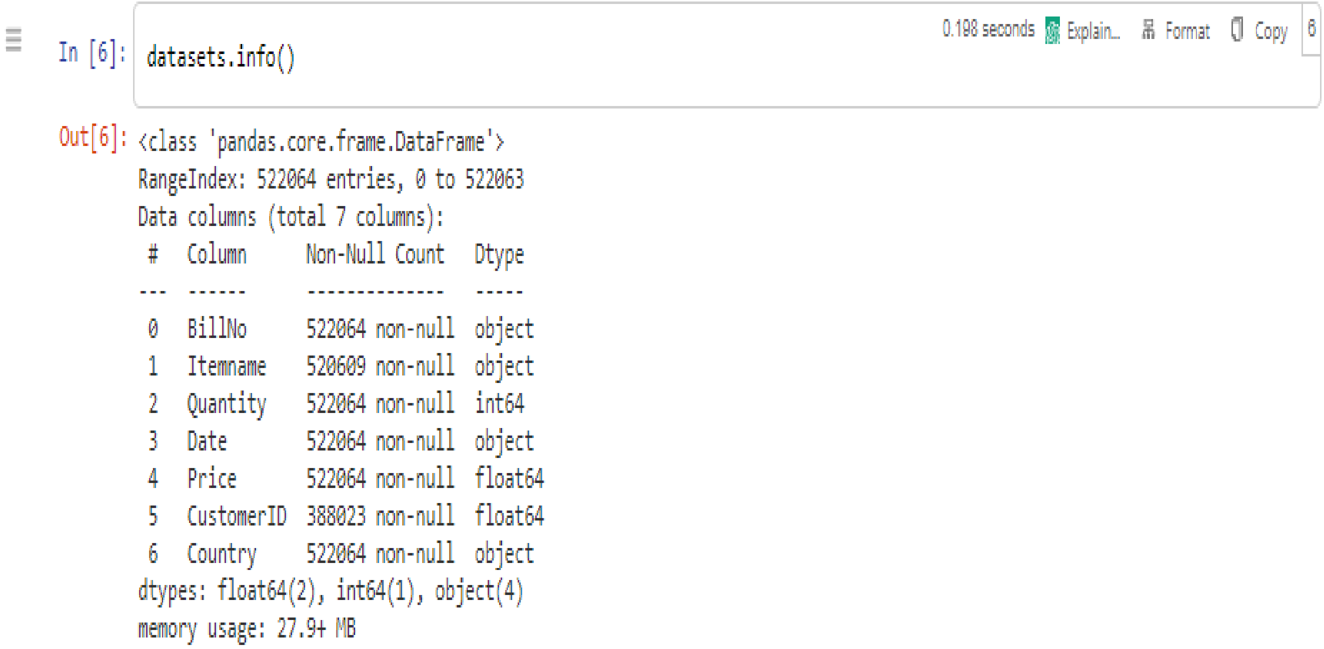
The code datasets.head() is calling the head() function on the dataset

is used display first few rows of a data set.



```
In [5]: datasets.isnull().sum()

Out[5]: BillNo            0
        Itemname       1455
        Quantity          0
        Date              0
        Price             0
        CustomerID   134041
        Country           0
        dtype: int64
```

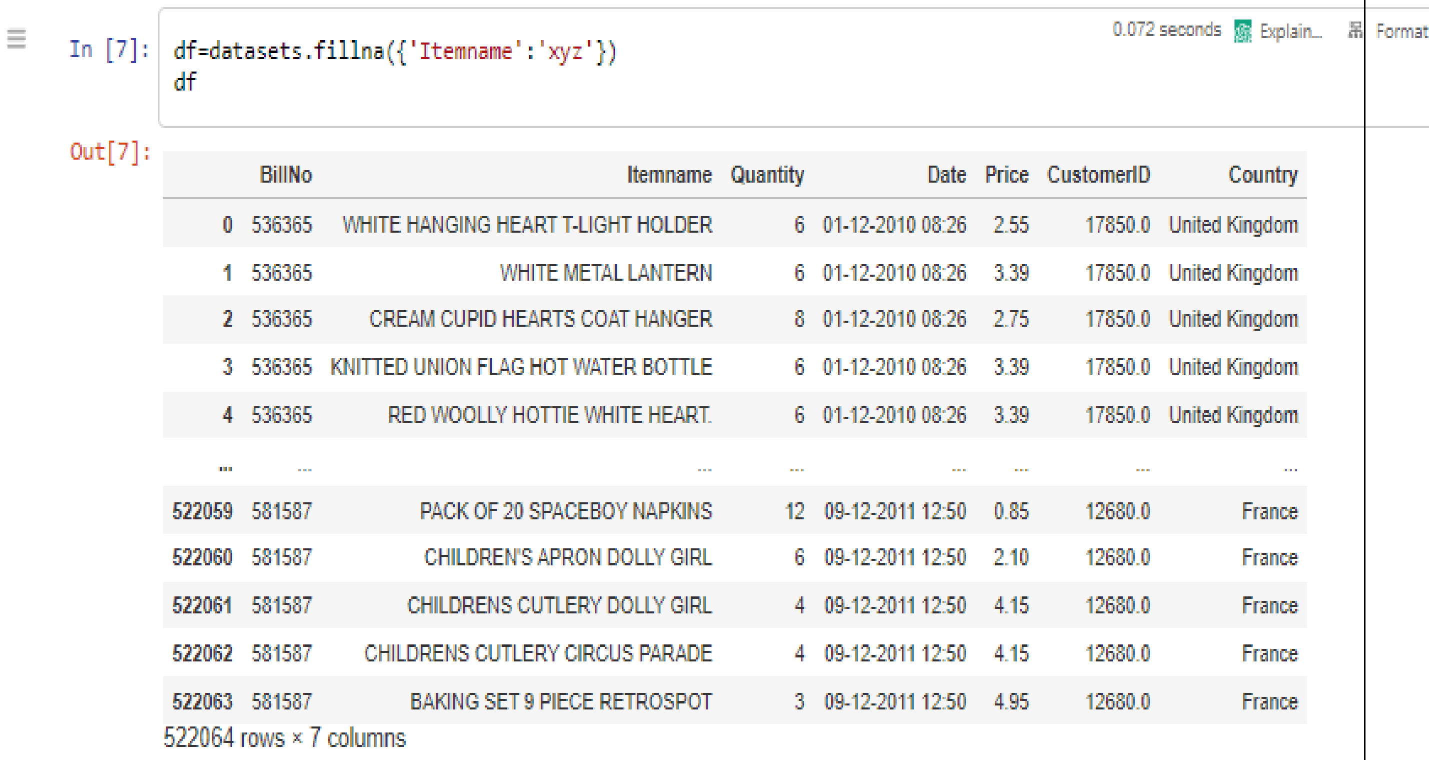
The isnull() function is used to find the number of missing values in column of a dataset. The sum() function is count the number of missing values.



```
In [6]: datasets.info()

Out[6]: <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 522064 entries, 0 to 522063
        Data columns (total 7 columns):
         #   Column      Non-Null Count   Dtype
        ---  ------      --------------   -----
         0   BillNo      522064 non-null  object
         1   Itemname    520609 non-null  object
         2   Quantity    522064 non-null  int64
         3   Date        522064 non-null  object
         4   Price       522064 non-null  float64
         5   CustomerID  388023 non-null  float64
         6   Country     522064 non-null  object
        dtypes: float64(2), int64(1), object(4)
        memory usage: 27.9+ MB
```

The code datasets.info() is a method call in python to display the

information about dataset. The info() method provides such as number of columns and rows datatypes of columns and memory usage of the dataset.



In [7]:

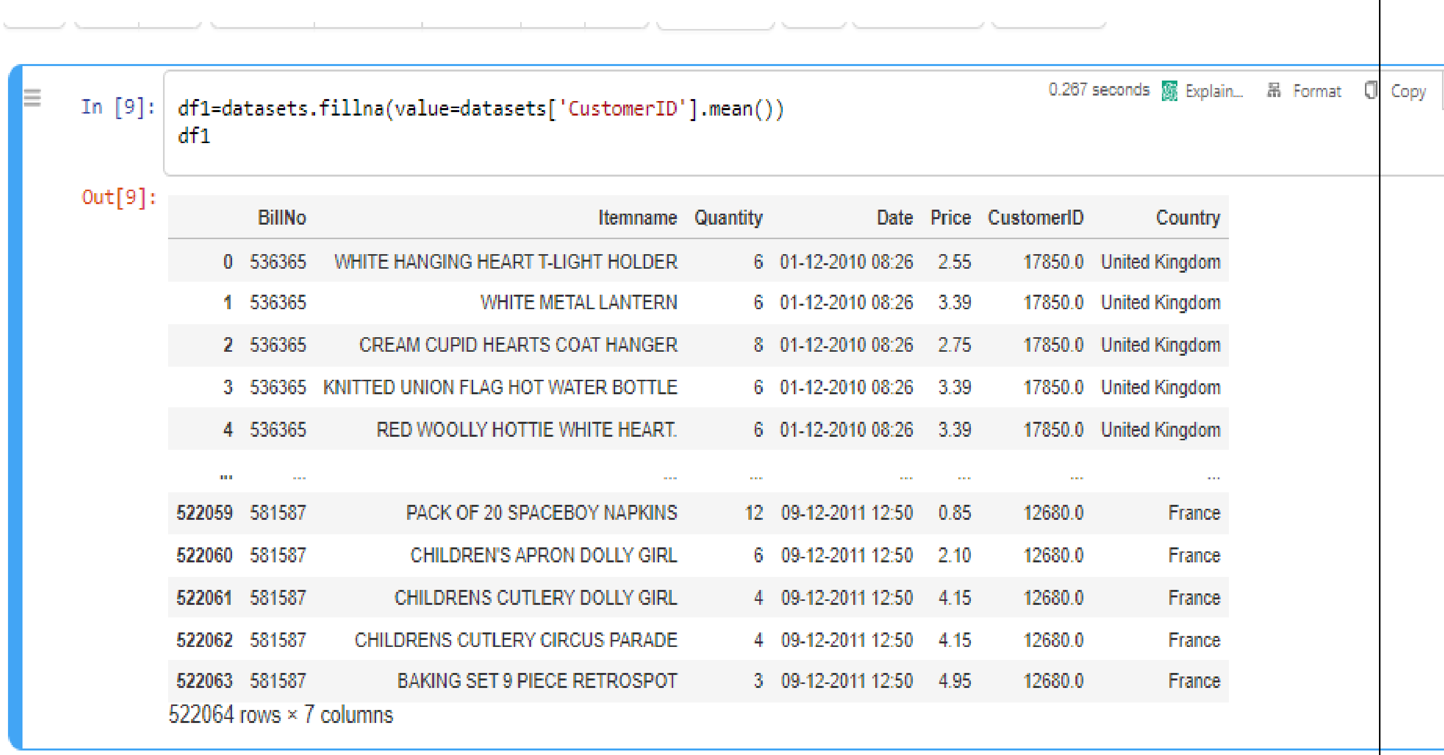
```
df=datasets.fillna({'Itemname':'xyz'})
df
```

Out[7]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	09-12-2011 12:50	0.85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	09-12-2011 12:50	2.10	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	09-12-2011 12:50	4.15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	09-12-2011 12:50	4.15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	09-12-2011 12:50	4.95	12680.0	France

522064 rows × 7 columns

The fillna() is used to filling the missing values in the columns "Itemname" of the data frame "datasets" with the value "xyz". The filled data frame is then displayed.



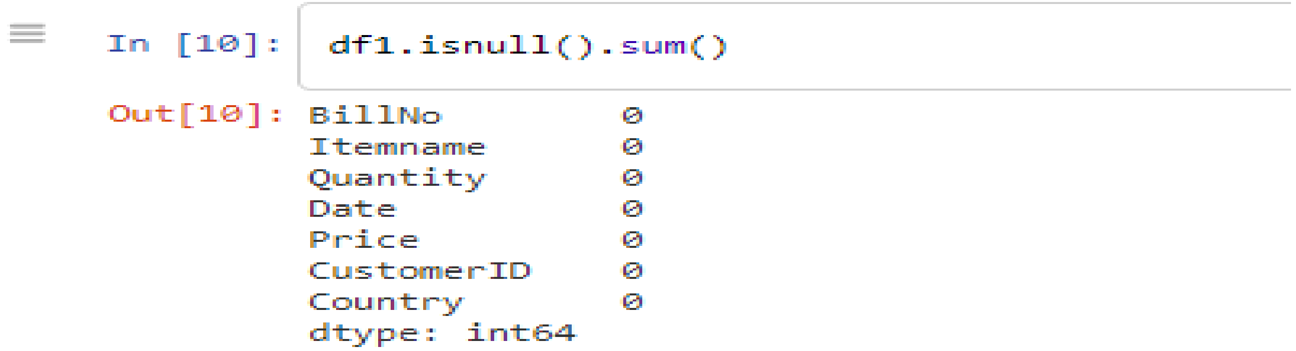
In [9]:
```
df1=datasets.fillna(value=datasets['CustomerID'].mean())
df1
```

Out[9]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:26	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:26	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:26	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	09-12-2011 12:50	0.85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	09-12-2011 12:50	2.10	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	09-12-2011 12:50	4.15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	09-12-2011 12:50	4.15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	09-12-2011 12:50	4.95	12680.0	France

522064 rows × 7 columns

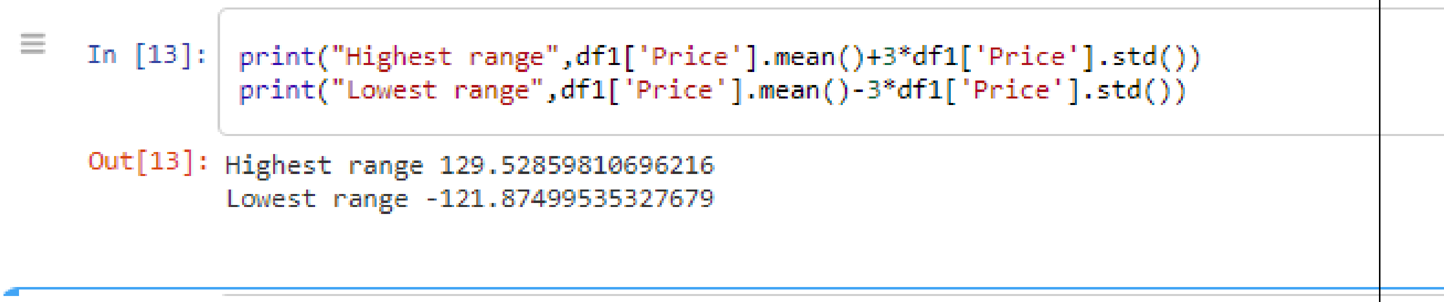
This code is fills the missing values in a data frame called dataset, using the mean of the "CustomerID" column. The filled data frame than assigned variable df1and displayed.



In [10]:
```
df1.isnull().sum()
```

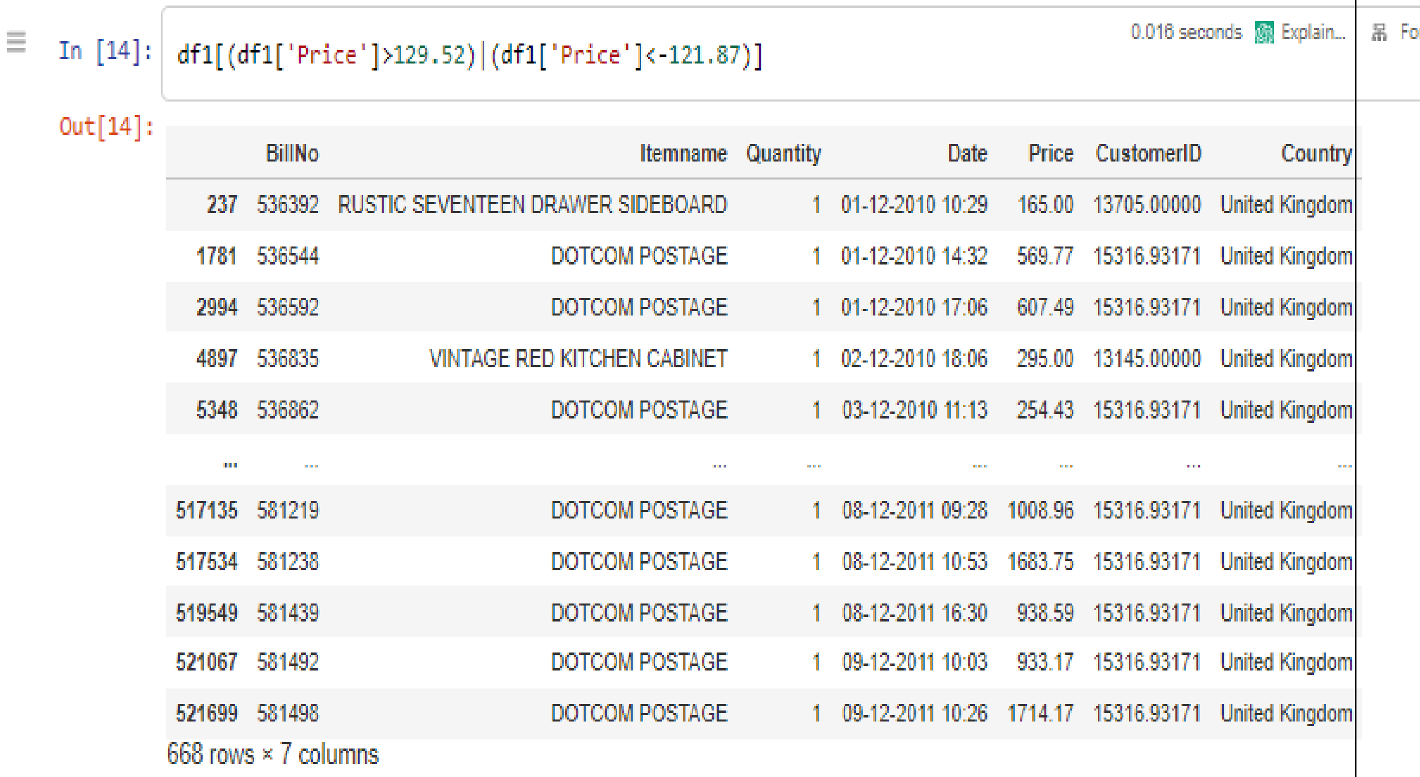
Out[10]:
```
BillNo        0
Itemname      0
Quantity      0
Date          0
Price         0
CustomerID    0
Country       0
dtype: int64
```

The isnull() function is used to find the number of missing values in column of a dataset. The sum() function is count the number of missing values.



```
In [13]: print("Highest range",df1['Price'].mean()+3*df1['Price'].std())
         print("Lowest range",df1['Price'].mean()-3*df1['Price'].std())

Out[13]: Highest range 129.52859810696216
         Lowest range -121.87499535327679
```

This code is printing the highest and lowest range based on statistical calculation. It calculates the mean and standard deviation of column called "Price" in data frame called df1.



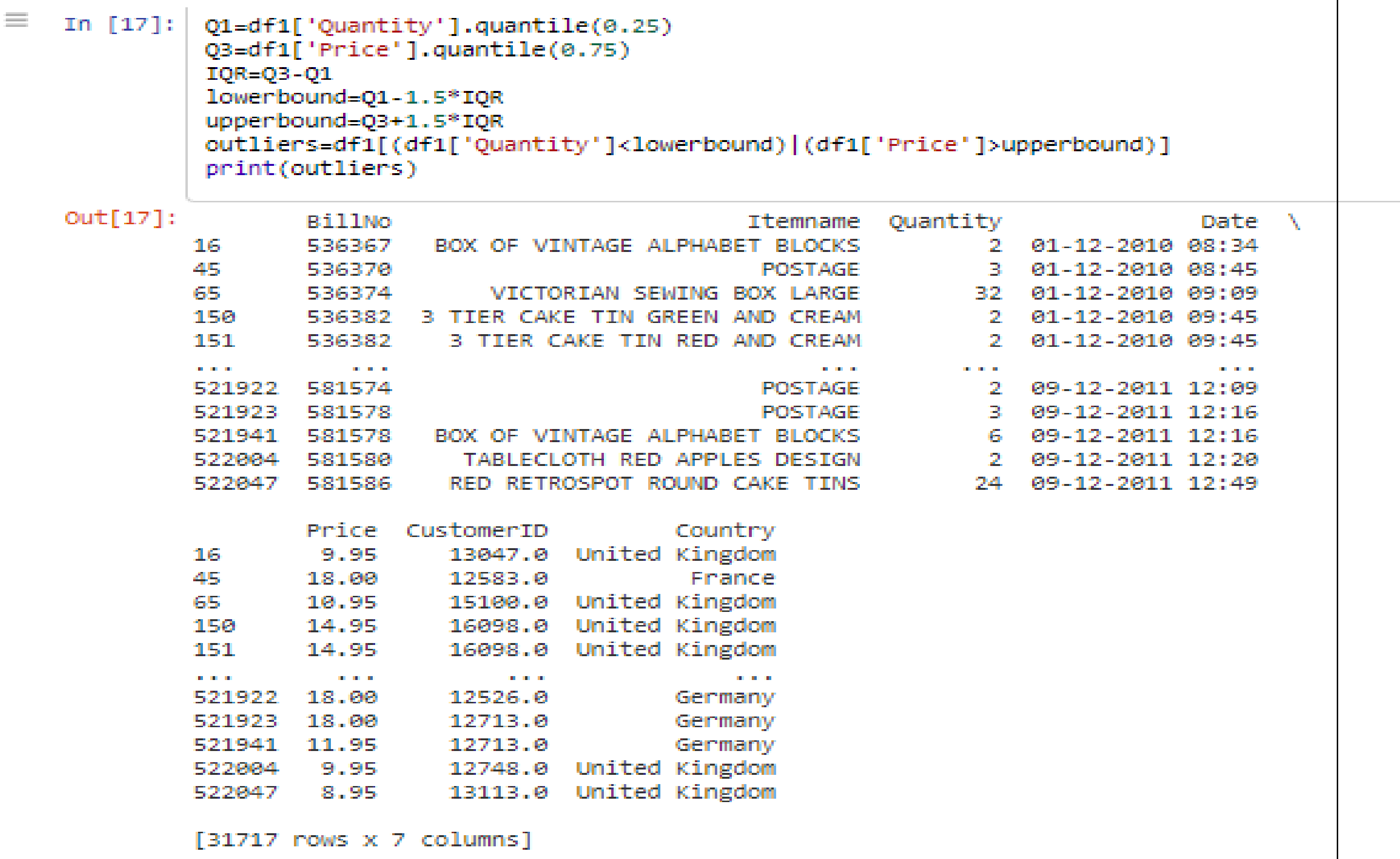
```
In [14]: df1[(df1['Price']>129.52)|(df1['Price']<-121.87)]
```

Out[14]:

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
237	536392	RUSTIC SEVENTEEN DRAWER SIDEBOARD	1	01-12-2010 10:29	165.00	13705.00000	United Kingdom
1781	536544	DOTCOM POSTAGE	1	01-12-2010 14:32	569.77	15316.93171	United Kingdom
2994	536592	DOTCOM POSTAGE	1	01-12-2010 17:06	607.49	15316.93171	United Kingdom
4897	536835	VINTAGE RED KITCHEN CABINET	1	02-12-2010 18:06	295.00	13145.00000	United Kingdom
5348	536862	DOTCOM POSTAGE	1	03-12-2010 11:13	254.43	15316.93171	United Kingdom
...	...	...	...	...	...	...	...
517135	581219	DOTCOM POSTAGE	1	08-12-2011 09:28	1008.96	15316.93171	United Kingdom
517534	581238	DOTCOM POSTAGE	1	08-12-2011 10:53	1683.75	15316.93171	United Kingdom
519549	581439	DOTCOM POSTAGE	1	08-12-2011 16:30	938.59	15316.93171	United Kingdom
521067	581492	DOTCOM POSTAGE	1	09-12-2011 10:03	933.17	15316.93171	United Kingdom
521699	581498	DOTCOM POSTAGE	1	09-12-2011 10:26	1714.17	15316.93171	United Kingdom

668 rows × 7 columns

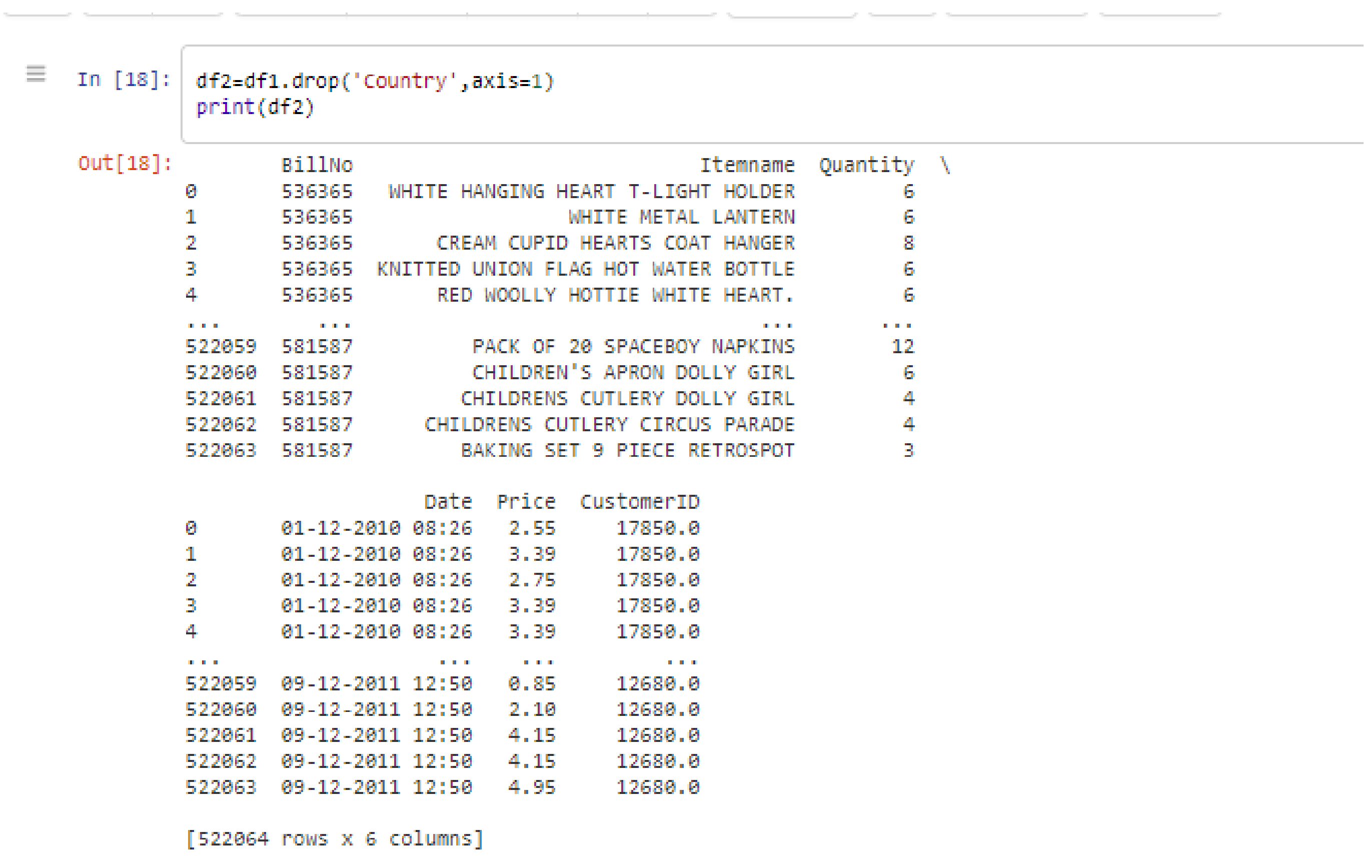
This code is used to filtering the data frame df1 based on the given condition.



```
In [17]:  Q1=df1['Quantity'].quantile(0.25)
          Q3=df1['Price'].quantile(0.75)
          IQR=Q3-Q1
          lowerbound=Q1-1.5*IQR
          upperbound=Q3+1.5*IQR
          outliers=df1[(df1['Quantity']<lowerbound)|(df1['Price']>upperbound)]
          print(outliers)
```

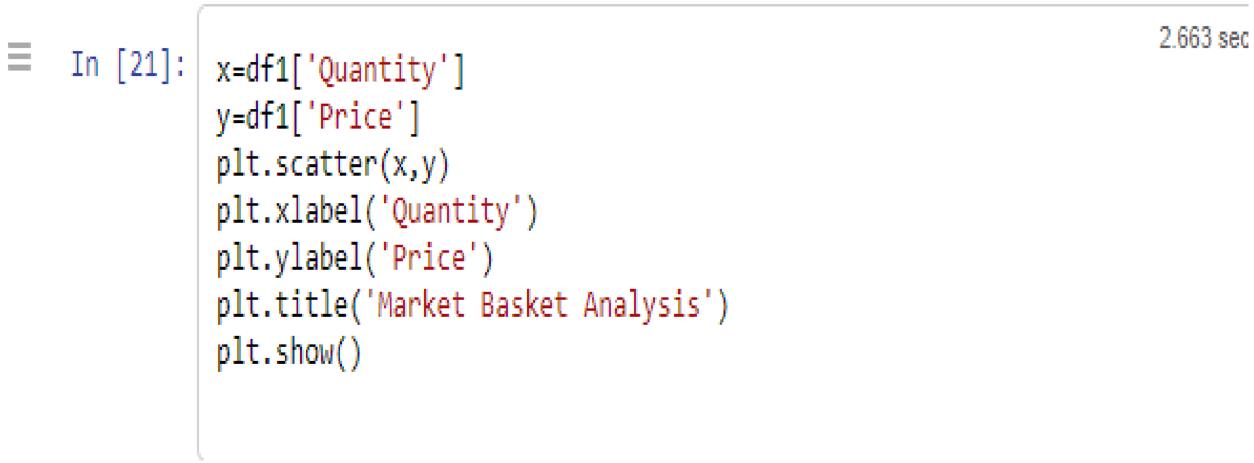
```
Out[17]:          BillNo                            Itemname  Quantity             Date  \
          16      536367    BOX OF VINTAGE ALPHABET BLOCKS         2  01-12-2010 08:34
          45      536370                           POSTAGE         3  01-12-2010 08:45
          65      536374        VICTORIAN SEWING BOX LARGE        32  01-12-2010 09:09
          150     536382  3 TIER CAKE TIN GREEN AND CREAM         2  01-12-2010 09:45
          151     536382    3 TIER CAKE TIN RED AND CREAM         2  01-12-2010 09:45
          ...        ...                               ...       ...              ...
          521922  581574                           POSTAGE         2  09-12-2011 12:09
          521923  581578                           POSTAGE         3  09-12-2011 12:16
          521941  581578    BOX OF VINTAGE ALPHABET BLOCKS         6  09-12-2011 12:16
          522004  581580     TABLECLOTH RED APPLES DESIGN          2  09-12-2011 12:20
          522047  581586    RED RETROSPOT ROUND CAKE TINS         24  09-12-2011 12:49

                  Price  CustomerID         Country
          16       9.95     13047.0  United Kingdom
          45      18.00     12583.0          France
          65      10.95     15100.0  United Kingdom
          150     14.95     16098.0  United Kingdom
          151     14.95     16098.0  United Kingdom
          ...       ...         ...             ...
          521922  18.00     12526.0         Germany
          521923  18.00     12713.0         Germany
          521941  11.95     12713.0         Germany
          522004   9.95     12748.0  United Kingdom
          522047   8.95     13113.0  United Kingdom

          [31717 rows x 7 columns]
```

- Q1 and Q3 are the first and third quartiles of the 'Quantity' and 'Price' columns, respectively.
- IQR is the interquartile range, calculated as the difference between Q3 and Q1.
- lowerbound and upperbound are the lower and upper bounds, respectively, for identifying outliers. They are calculated as Q1 - 1.5 * IQR and Q3 + 1.5 * IQR.
- Outliers is a Data Frame containing the rows from df1 where either the 'Quantity' is less than lowerbound or the 'Price' is greater than upperbound.
- Finally, the code prints out the outliers Data Frame.

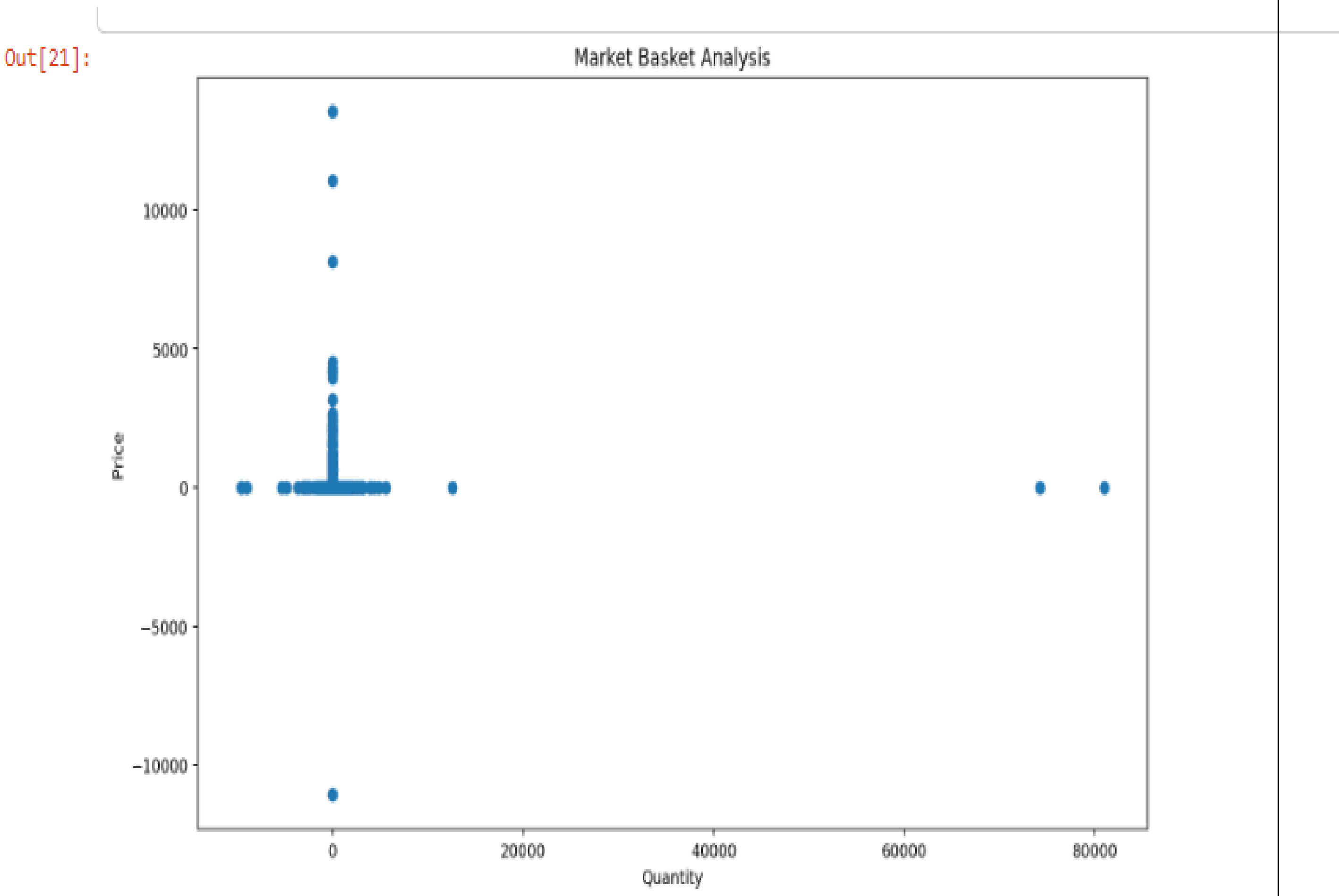


```
In [18]: df2=df1.drop('Country',axis=1)
         print(df2)

Out[18]:         BillNo                            Itemname  Quantity  \
         0       536365   WHITE HANGING HEART T-LIGHT HOLDER         6
         1       536365                 WHITE METAL LANTERN         6
         2       536365      CREAM CUPID HEARTS COAT HANGER         8
         3       536365 KNITTED UNION FLAG HOT WATER BOTTLE         6
         4       536365      RED WOOLLY HOTTIE WHITE HEART.         6
         ...        ...                                 ...       ...
         522059  581587         PACK OF 20 SPACEBOY NAPKINS        12
         522060  581587         CHILDREN'S APRON DOLLY GIRL         6
         522061  581587        CHILDRENS CUTLERY DOLLY GIRL         4
         522062  581587     CHILDRENS CUTLERY CIRCUS PARADE         4
         522063  581587        BAKING SET 9 PIECE RETROSPOT         3

                             Date  Price  CustomerID
         0       01-12-2010 08:26   2.55     17850.0
         1       01-12-2010 08:26   3.39     17850.0
         2       01-12-2010 08:26   2.75     17850.0
         3       01-12-2010 08:26   3.39     17850.0
         4       01-12-2010 08:26   3.39     17850.0
         ...                  ...    ...         ...
         522059  09-12-2011 12:50   0.85     12680.0
         522060  09-12-2011 12:50   2.10     12680.0
         522061  09-12-2011 12:50   4.15     12680.0
         522062  09-12-2011 12:50   4.15     12680.0
         522063  09-12-2011 12:50   4.95     12680.0

         [522064 rows x 6 columns]
```

This code using the pandas library in Python to drop the 'Country' column from a Data Frame called df1. The 'axis=1' parameter specifies that the column is being dropped.



```
In [21]: x=df1['Quantity']
         y=df1['Price']
         plt.scatter(x,y)
         plt.xlabel('Quantity')
         plt.ylabel('Price')
         plt.title('Market Basket Analysis')
         plt.show()
```



```
Out[21]:
```

This code takes two column values from a data frame and assign them to the variables x and y, plots them as a scatterplot using the scatter() function from the pyplot module of the matplotlib library, adds labels to the x-axis using the xlabel() and y-axis using the ylabel(), sets a title to the plot using the title() and displays the plot using the show() .