import pandas as pd import

numpy as np import

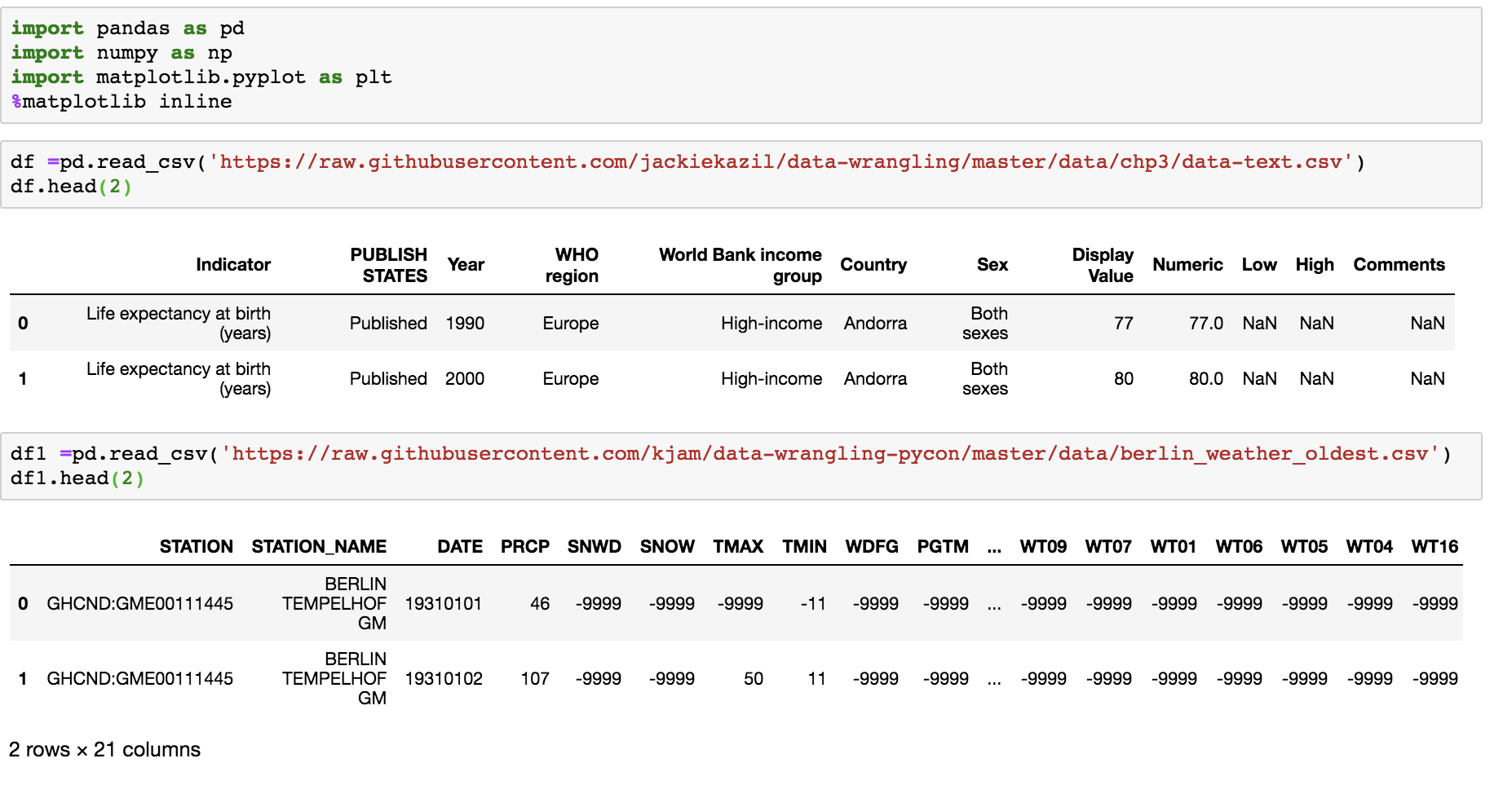
matplotlib.pyplot as plt

%matplotlib inline

df =pd.read\_csv('https://raw.githubusercontent.com/jackiekazil/data-wrangling/master/data/chp3/data-text.csv') df.head(2)

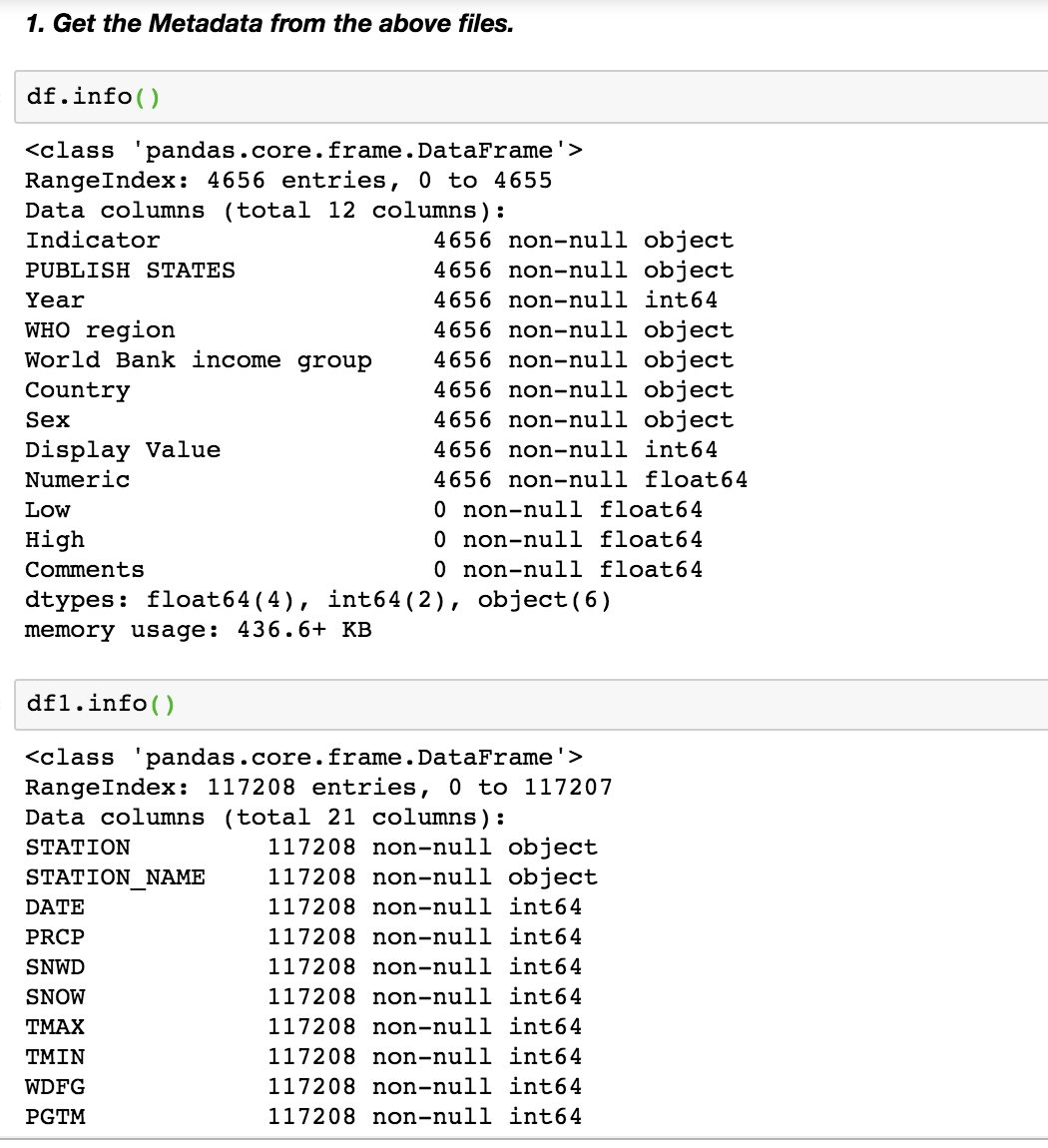
df1 =pd.read\_csv('https://raw.githubusercontent.com/kjam/data-wrangling-pycon/master/data/berlin\_weather\_oldest.csv') df1.head(2)

**SolutionScreenshot:-**

******

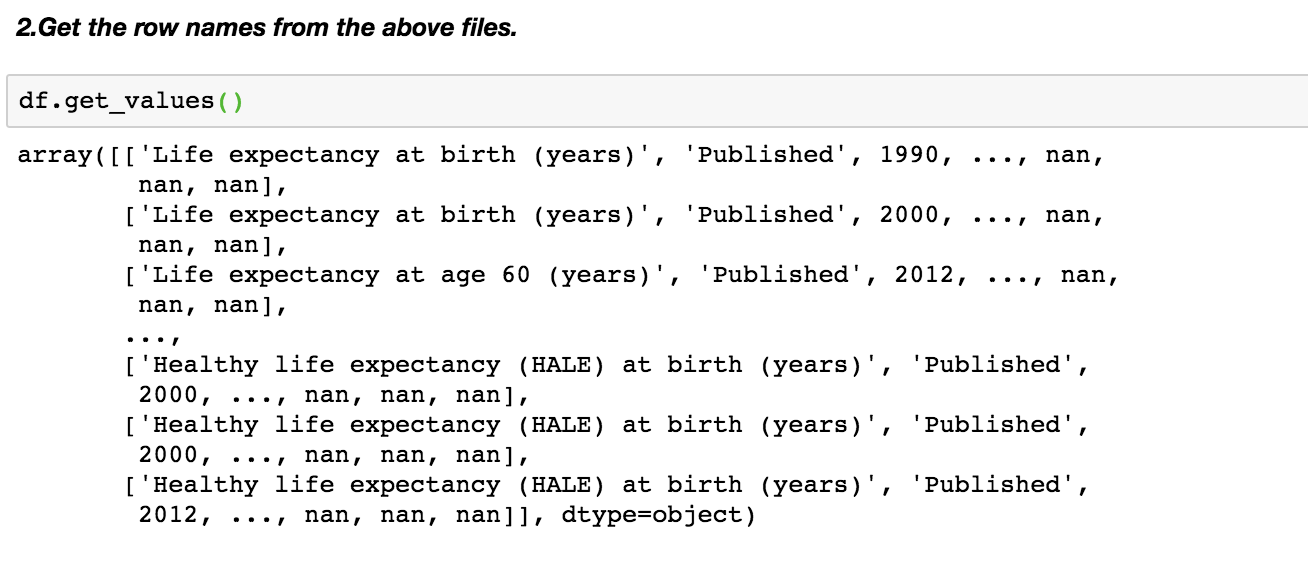
***1) Get the Metadata from the above files.***

**SolutionScreenshot:-**

****

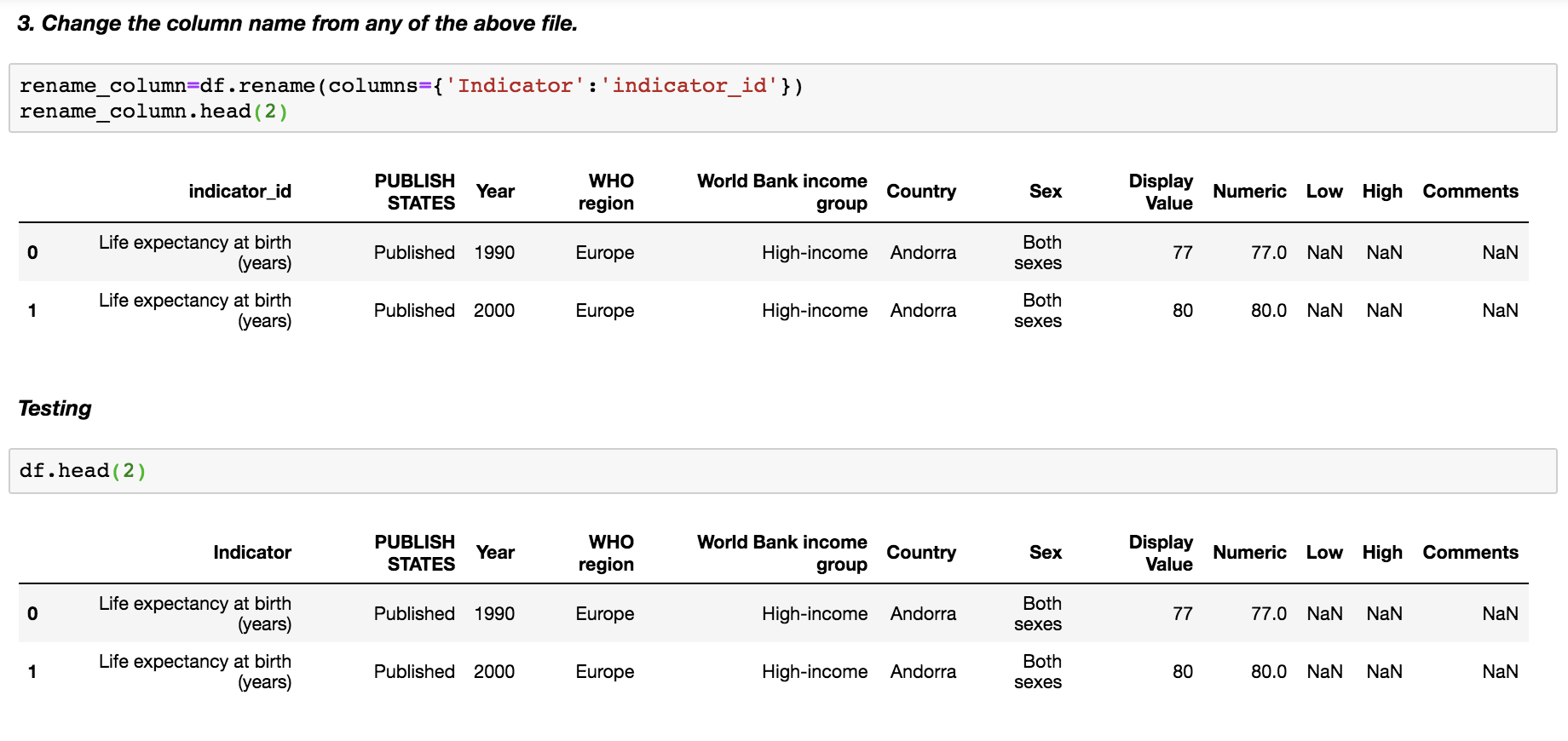
***2) Get the row names from the above files.***

**SolutionScreenshot:-**

******

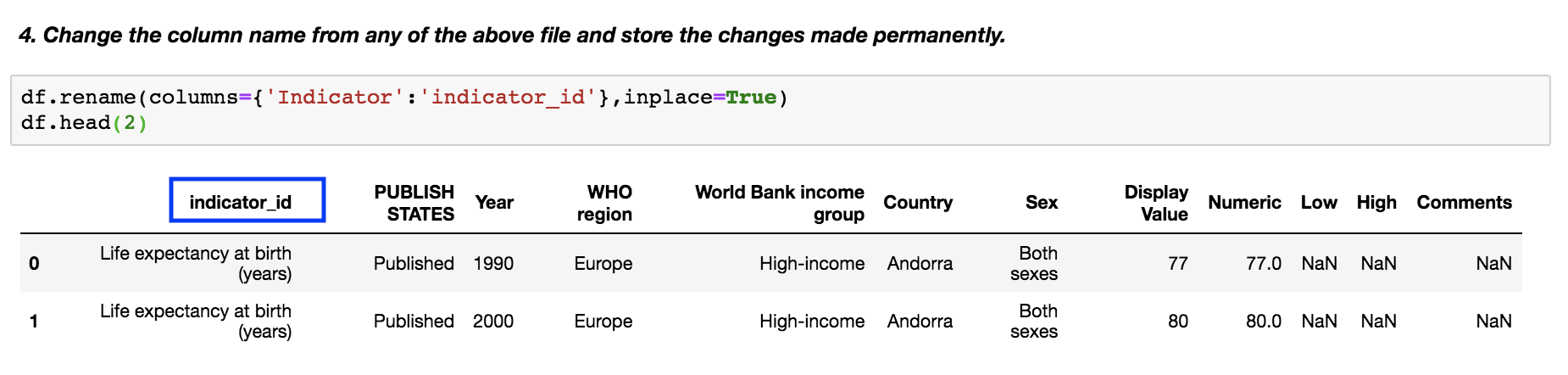
***3) Change the column name from any of the above file.***

**SolutionScreenshot:-**

****

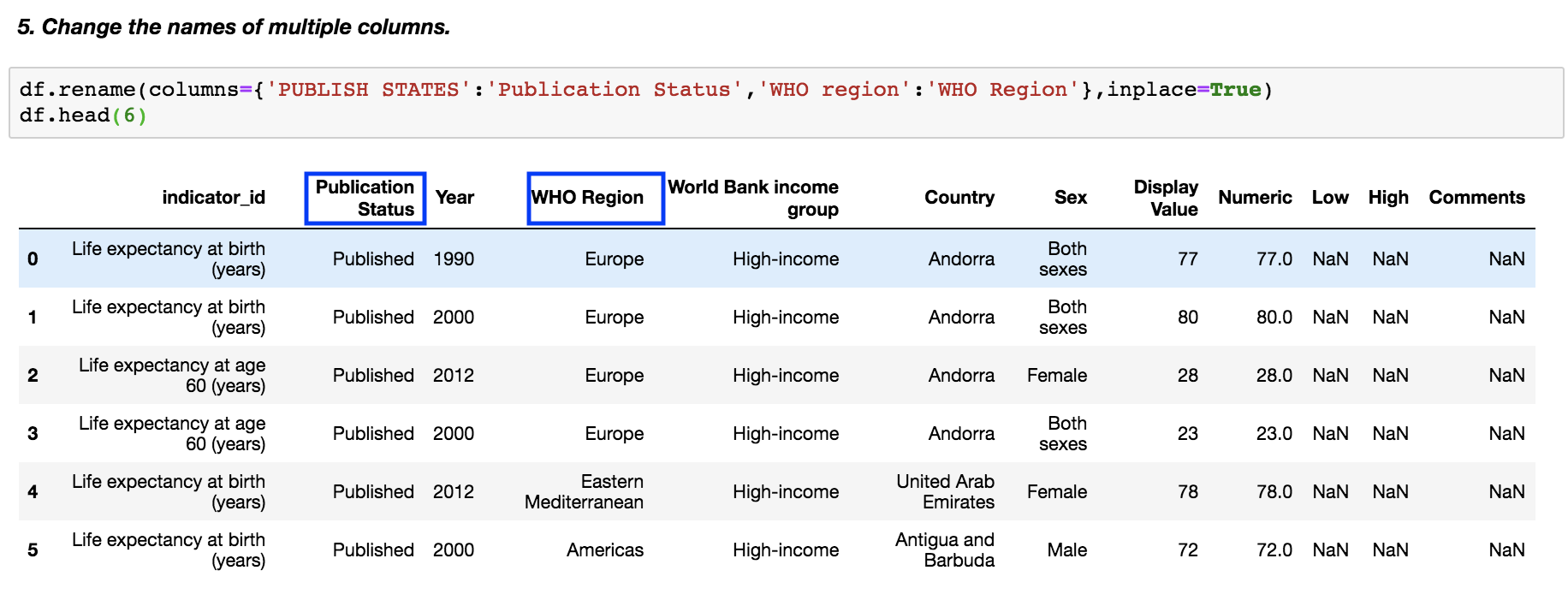
***4) Change the column name from any of the above file and store the changes made permanently***

**SolutionScreenshot:-**



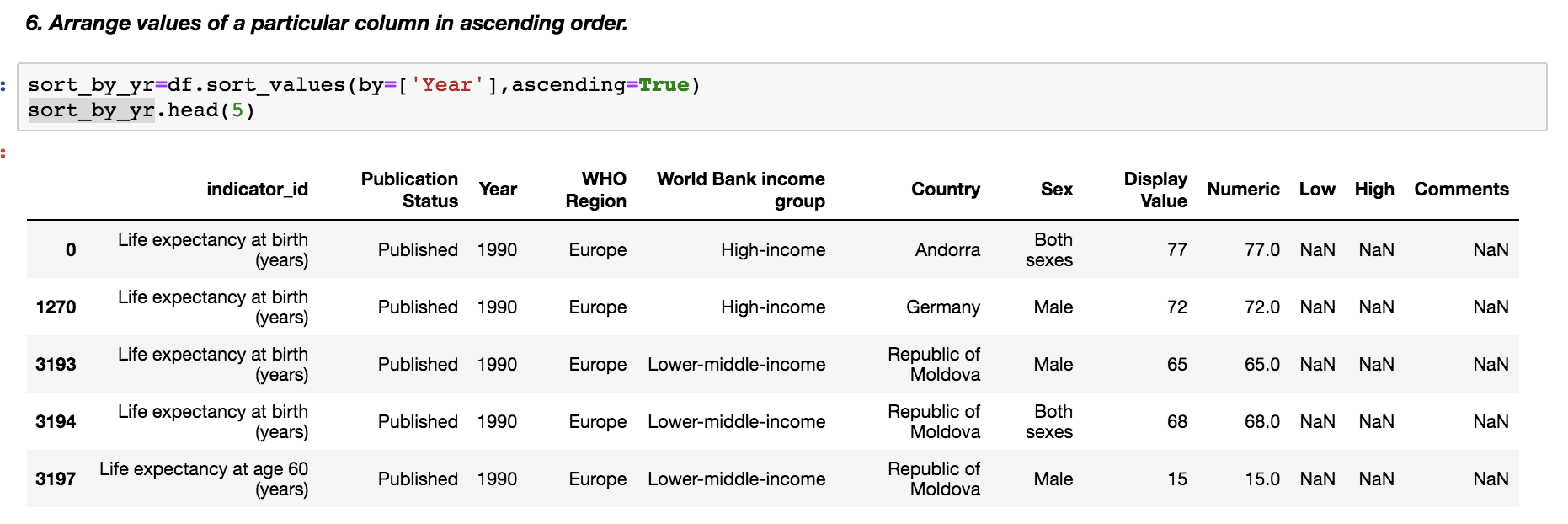
*5****) Change the names of multiple columns.***

**SolutionScreenshot:-**

****

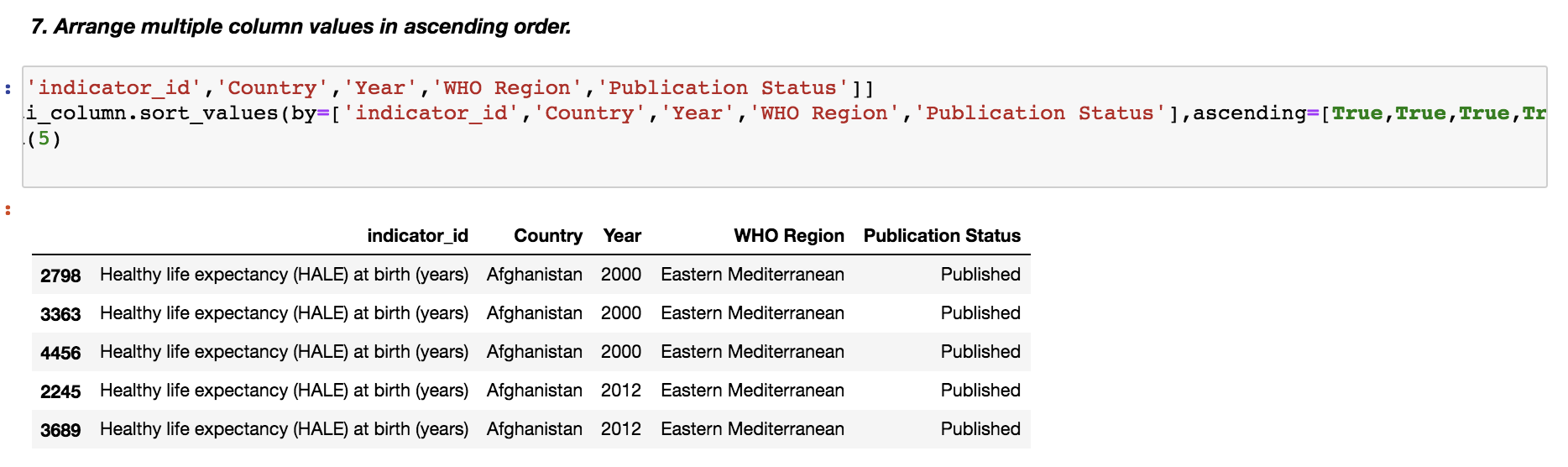
*6****) Arrange values of a particular column in ascending order..***

**SolutionScreenshot:-**

****

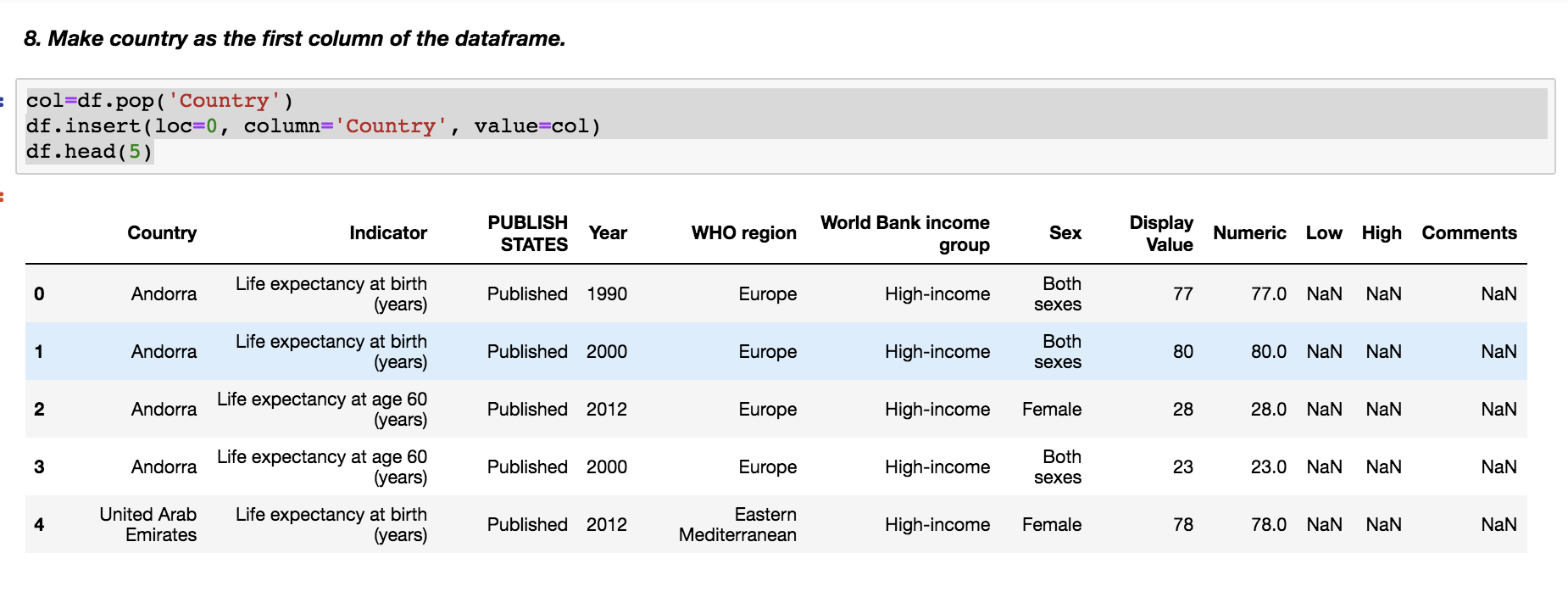
*7****) Arrange multiple column values in ascending order..***

**SolutionScreenshot:-**

****

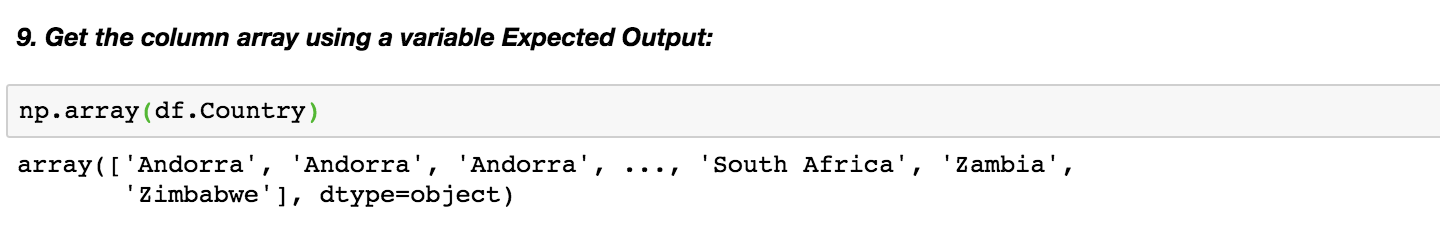
*8****) Make country as the first column of the dataframe..***

**SolutionScreenshot:-**

****

*9****) Get the column array using a variable.***

**SolutionScreenshot:-**

****

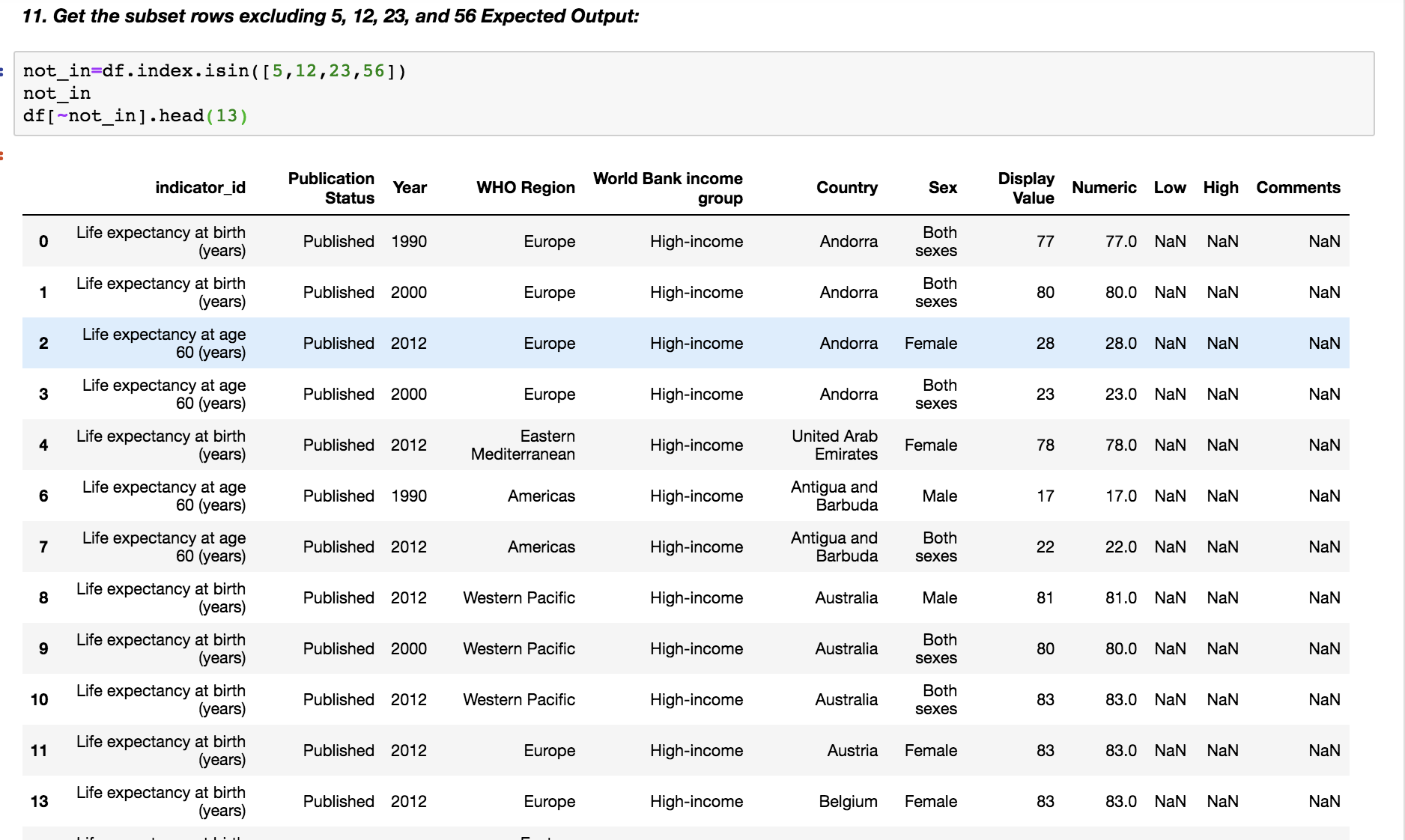
*10****) Get the subset rows 11, 24, 37.***

**SolutionScreenshot:-**

****

*11****) Get the subset rows excluding 5, 12, 23, and 56.***

**SolutionScreenshot:-**

****

Load datasets from CSV

Users= pd.read\_csv ('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data/users.csv' )

sessions =pd.read\_csv ('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data/sessions.csv)

products =pd.read\_csv ('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data/products.csv' )

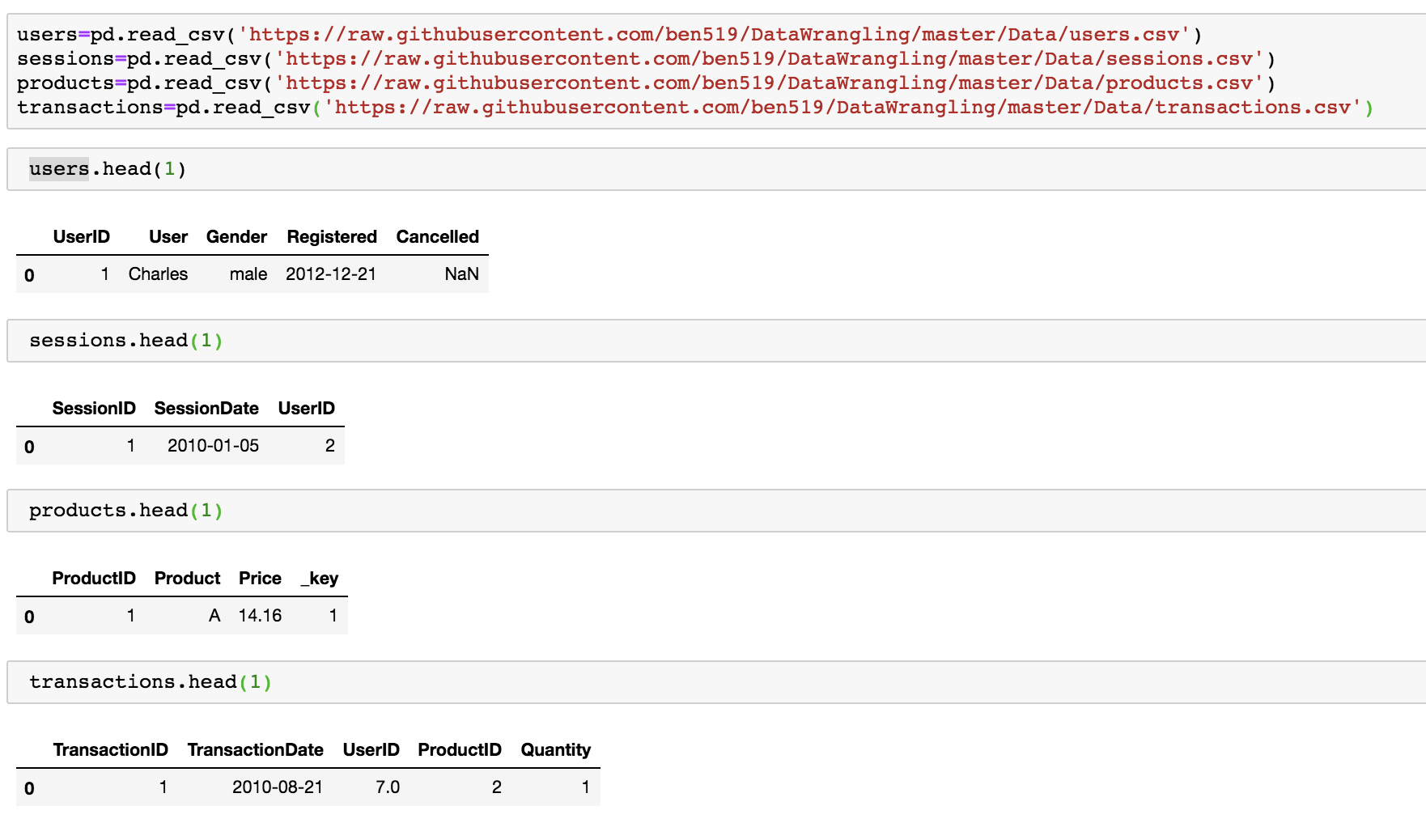
transactions =pd.read\_csv ('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data/transactions.csv)

users.head()

sessions.head()

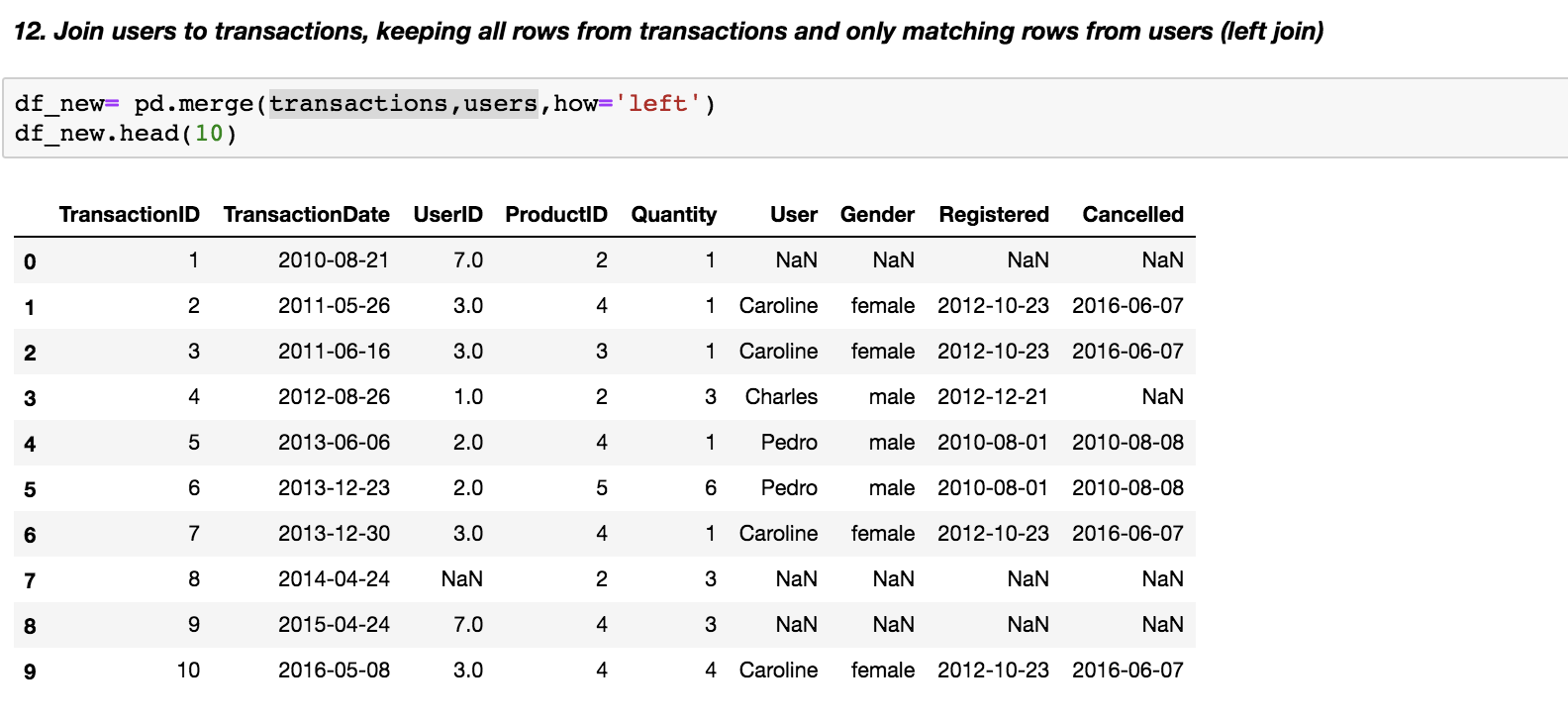
transactions.head()

**SolutionScreenshot:-**



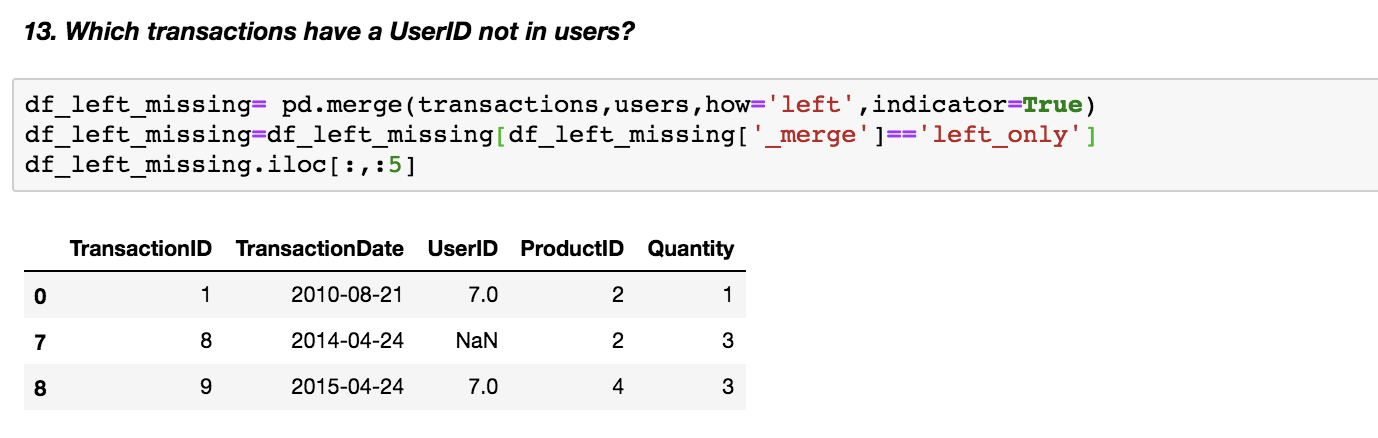
*12****)*** *Join users to transactions, keeping all rows from transactions and only matching rows from users (left join)*

**SolutionScreenshot:-**

****

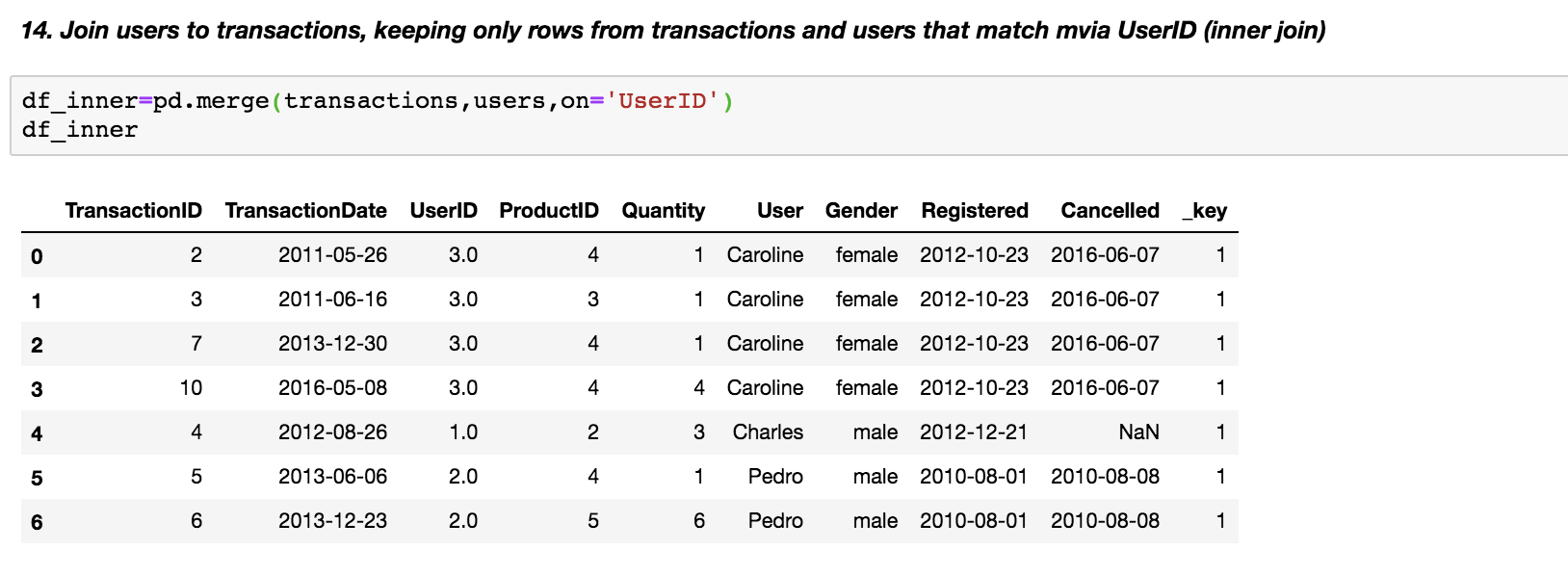
*13****) Which transactions have a UserID not in users?.***

**SolutionScreenshot:-**

****

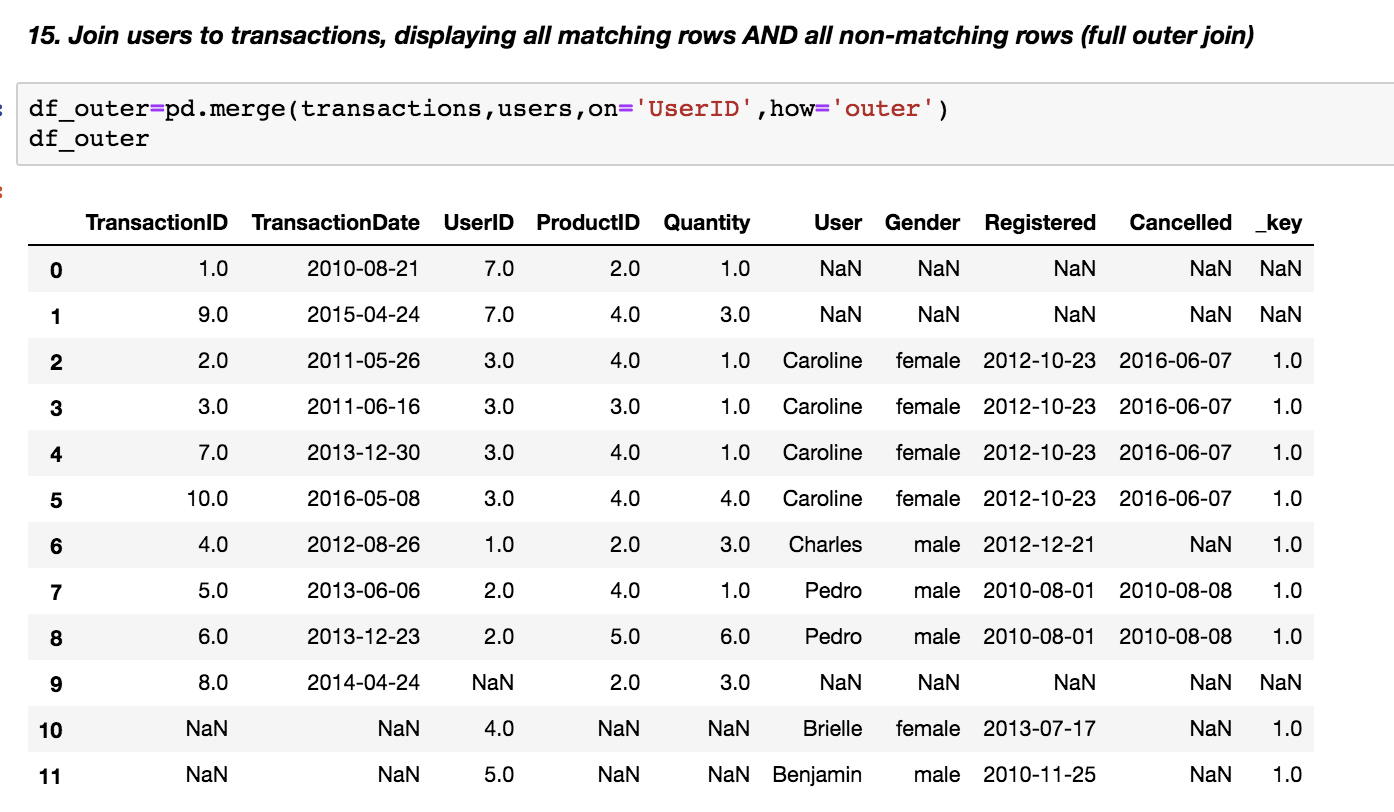
*14****) Join users to transactions, keeping only rows from transactions and users that match via UserID (inner join).***

**SolutionScreenshot:-**

****

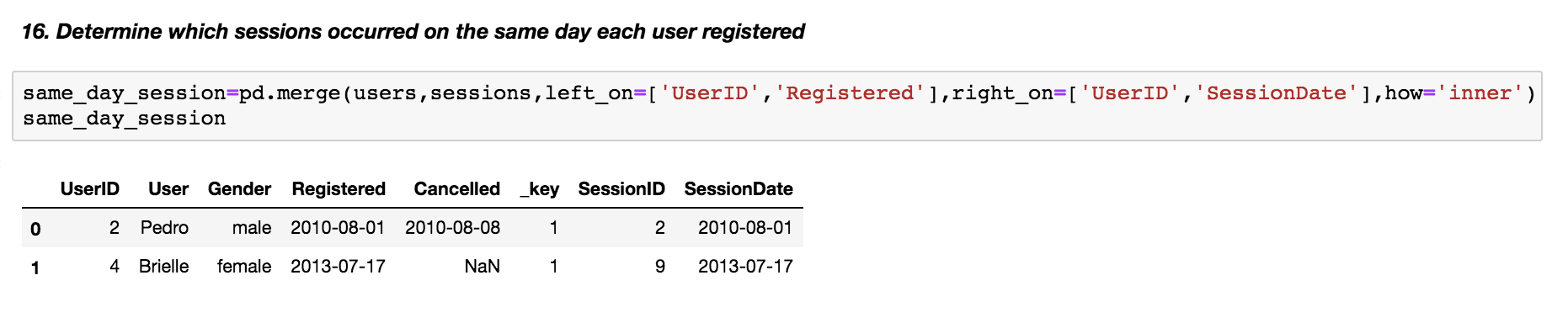
*15****) Join users to transactions, displaying all matching rows AND all non-matching rows (full outer join).***

**SolutionScreenshot:-**

****

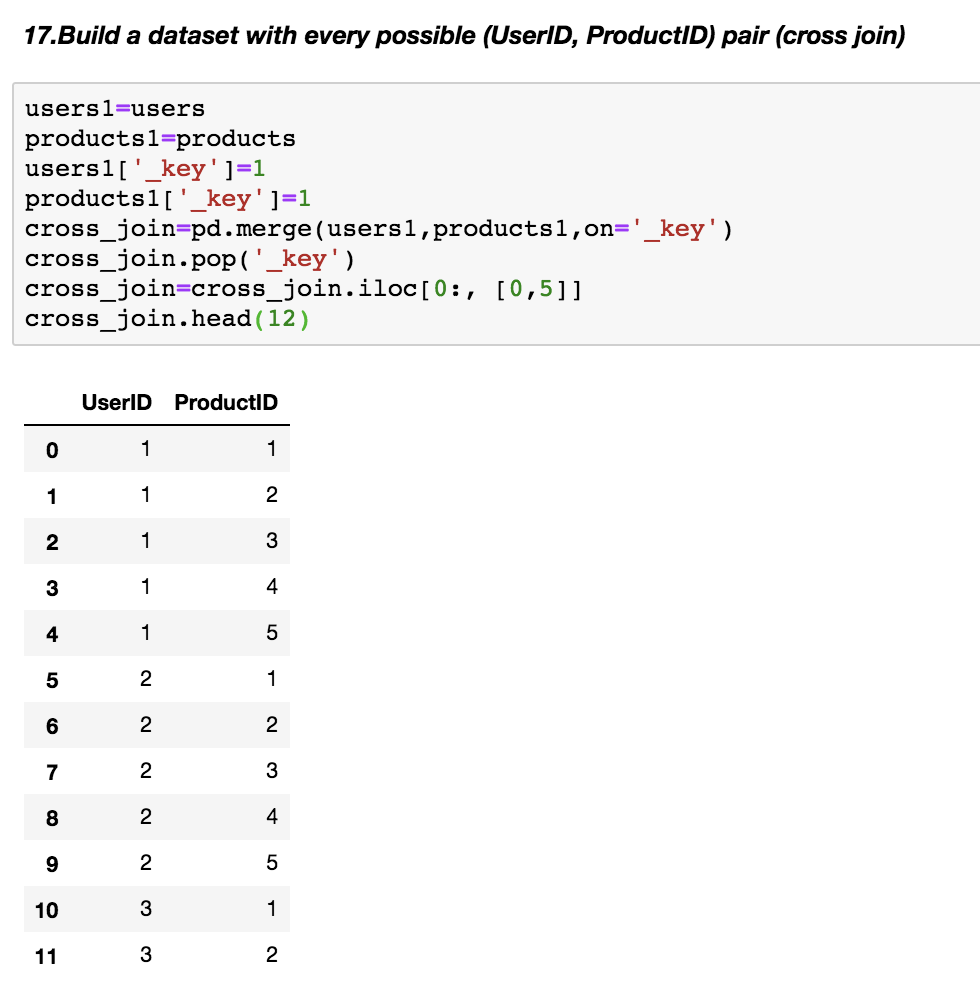
*16****)*** *Determine which sessions occurred on the same day each user registered.*

**SolutionScreenshot:-**

****

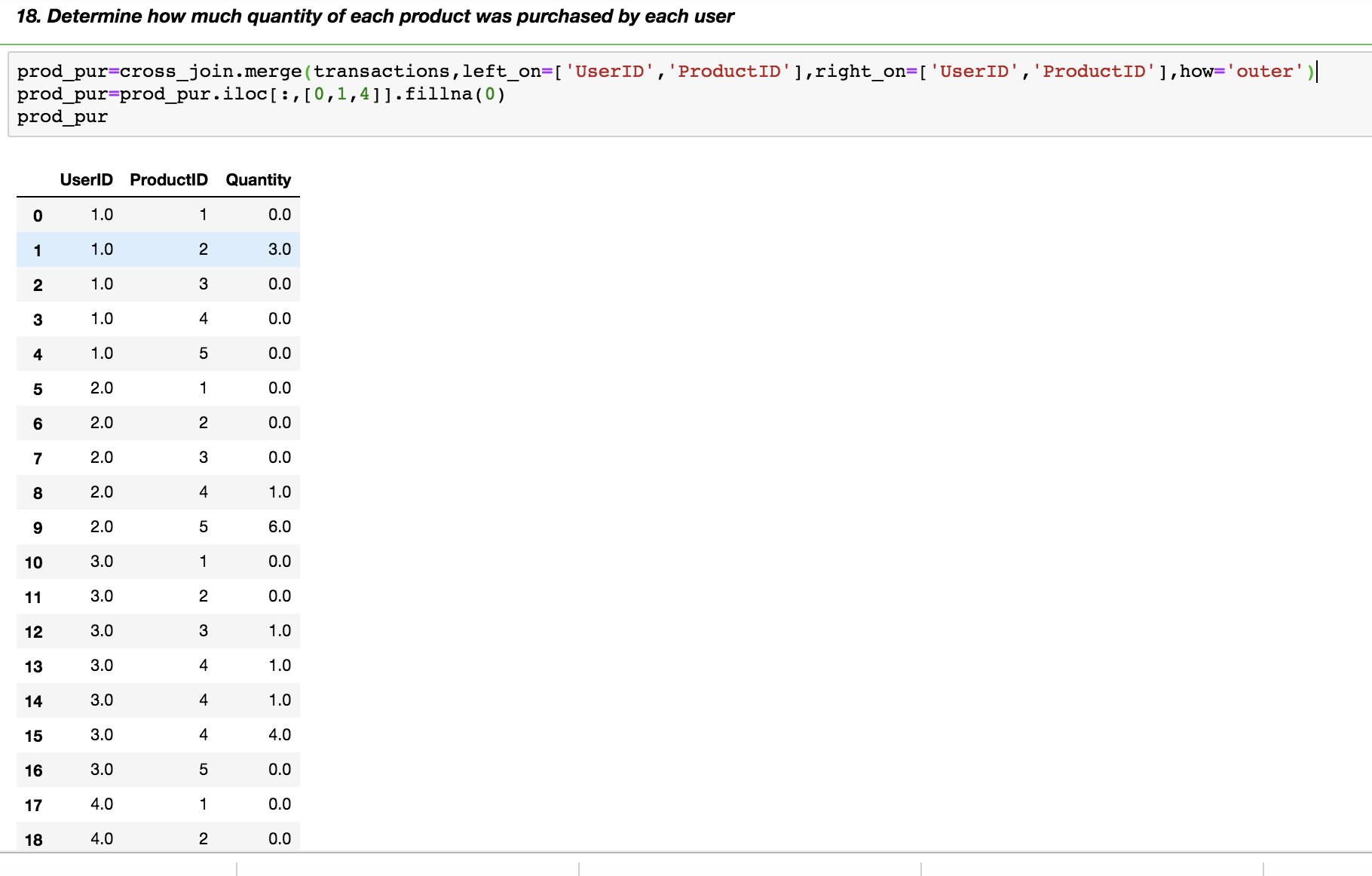
*17****)*** *Build a dataset with every possible (UserID, ProductID) pair (cross join).*

**SolutionScreenshot:-**

****

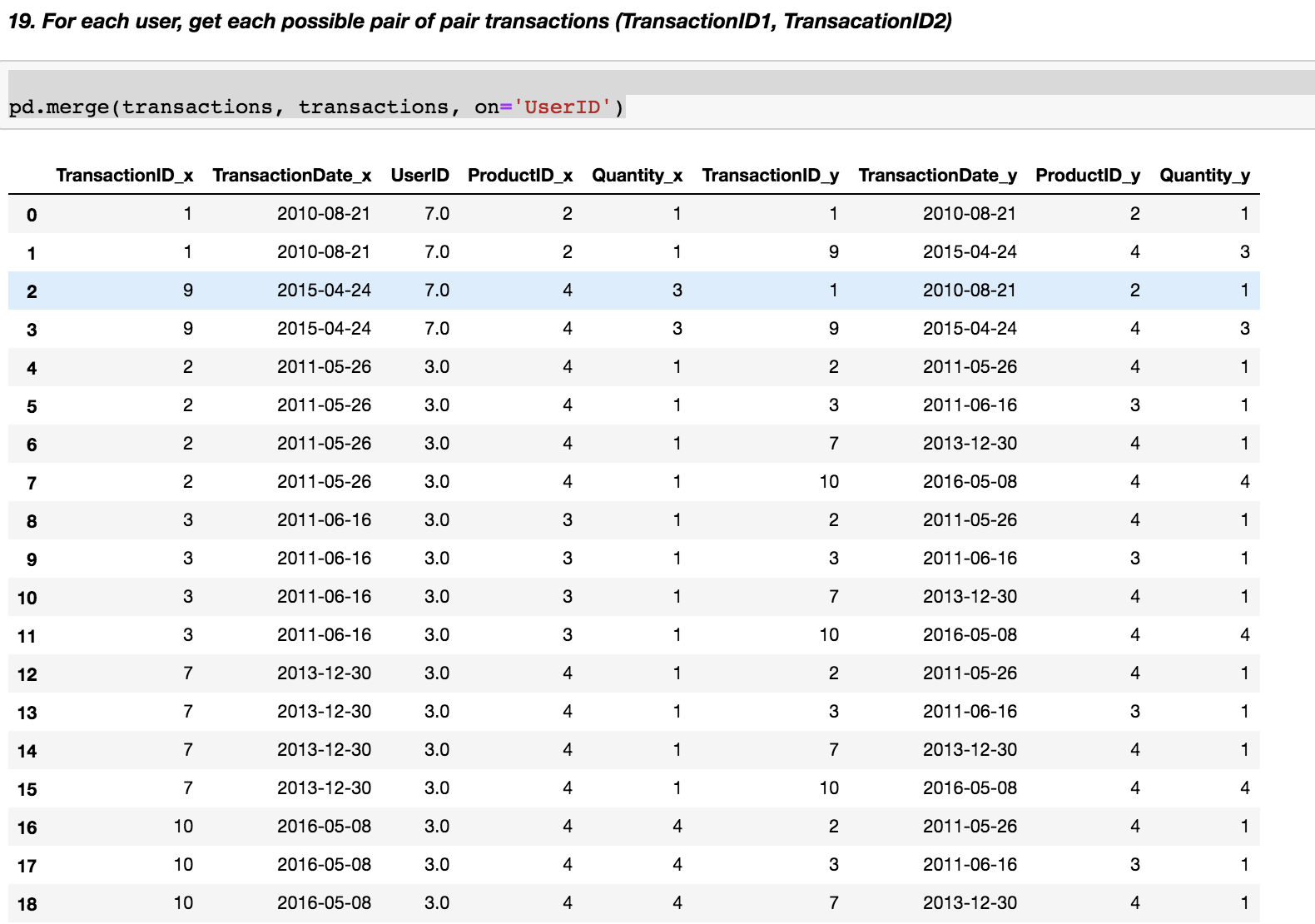
*18) Determine how much quantity of each product was purchased by each user.*

**SolutionScreenshot:-**

****

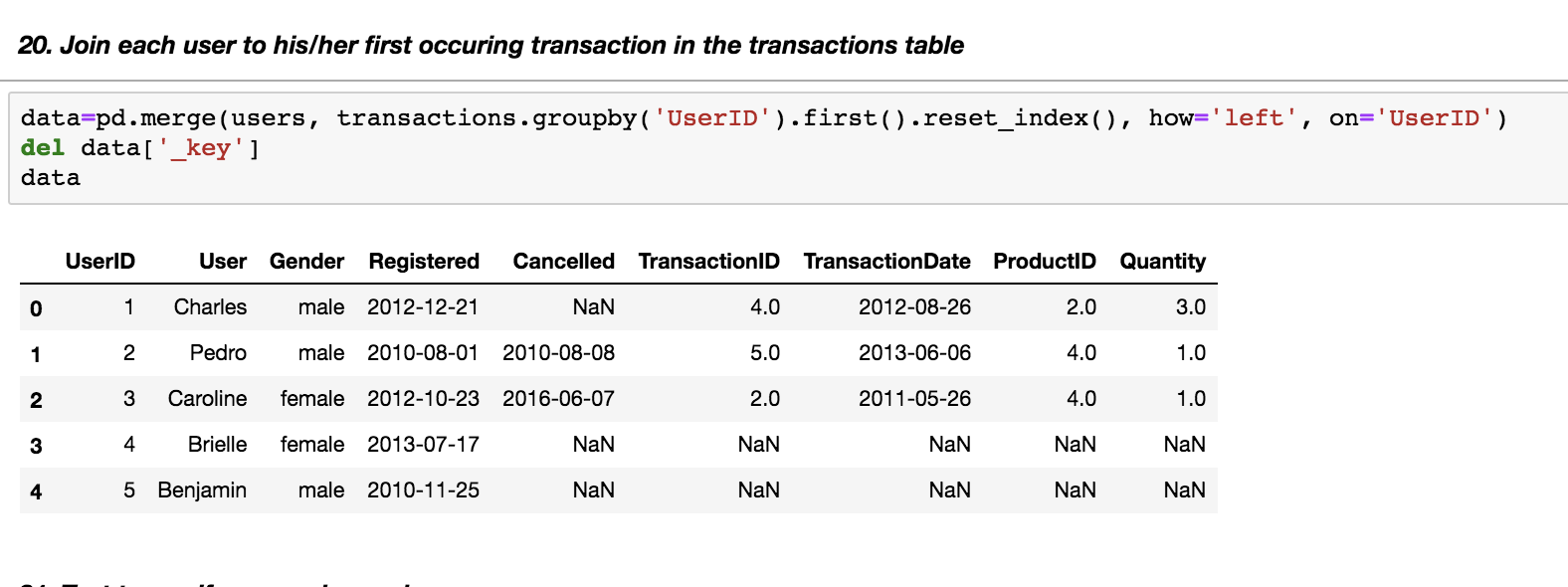
*19****)*** *For each user, get each possible pair of pair transactions (TransactionID1,TransacationID2).*

**SolutionScreenshot:-**

****

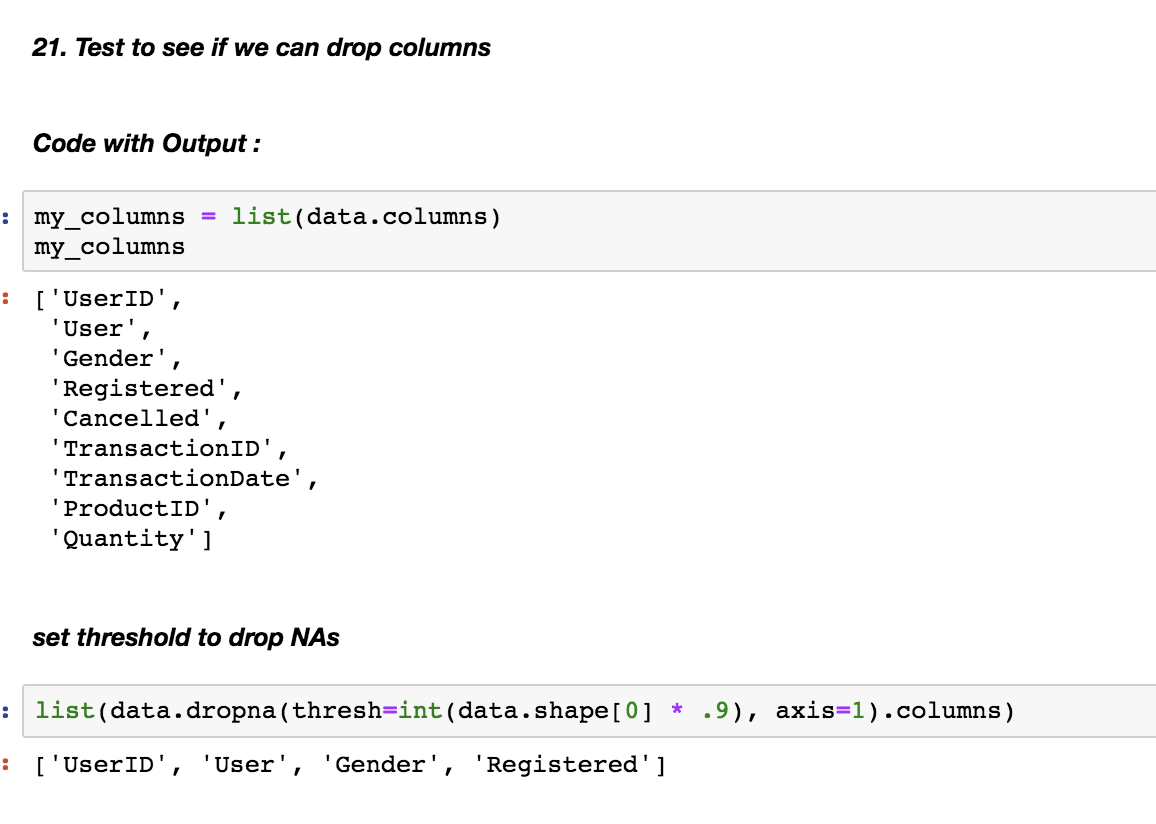
*20****)*** *Join each user to his/her first occuring transaction in the transactions table****.***

**SolutionScreenshot:-**

****

*21****)*** *Test to see if we can drop columns.*

**SolutionScreenshot:-**

****

****