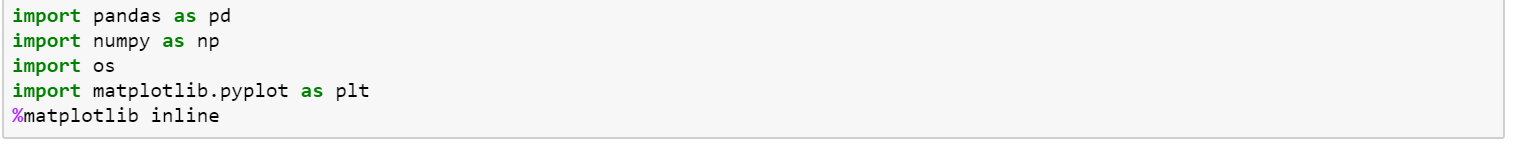
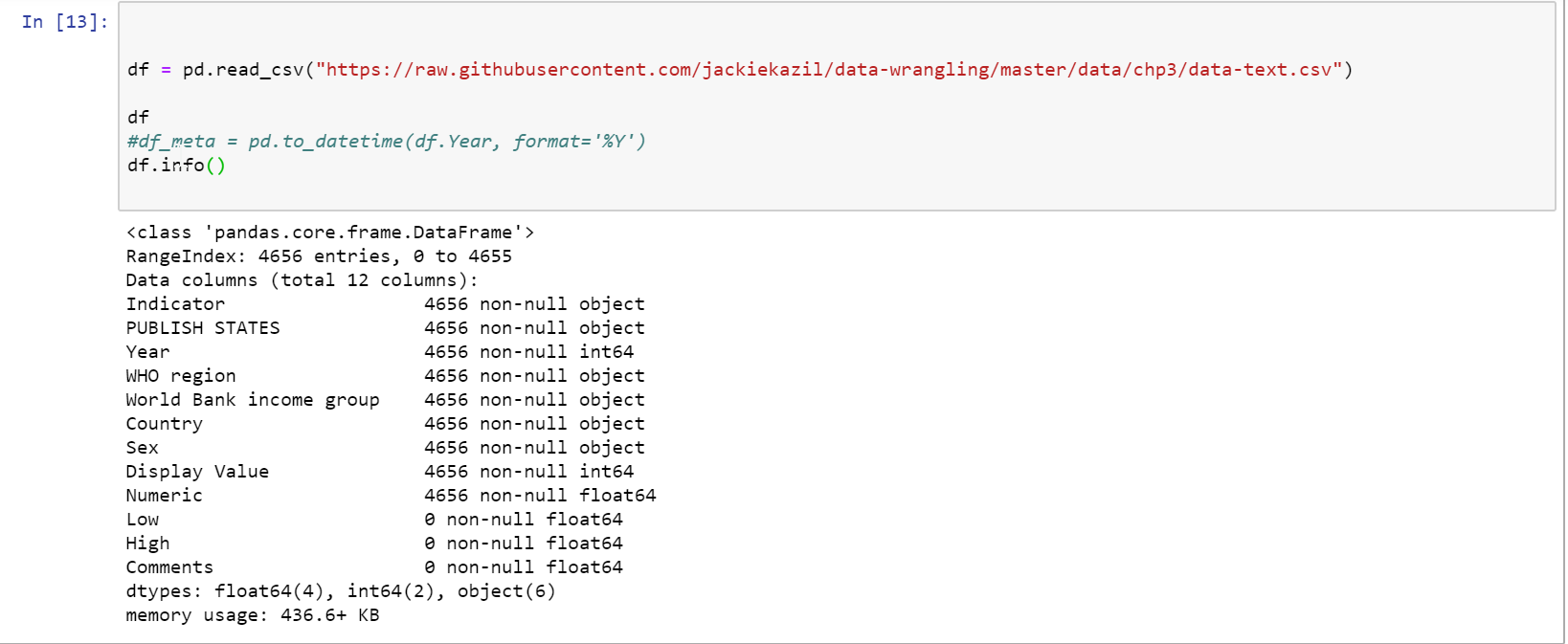
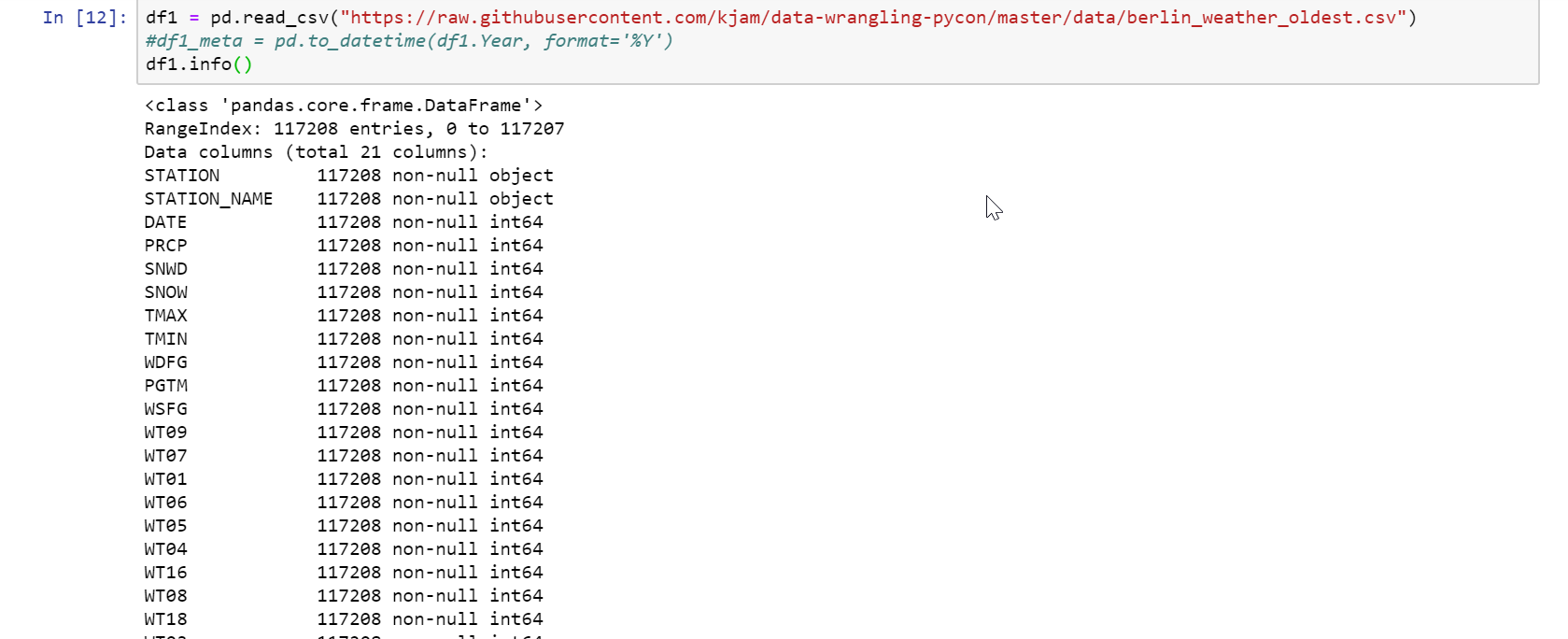
Problem 1

Get the meta data



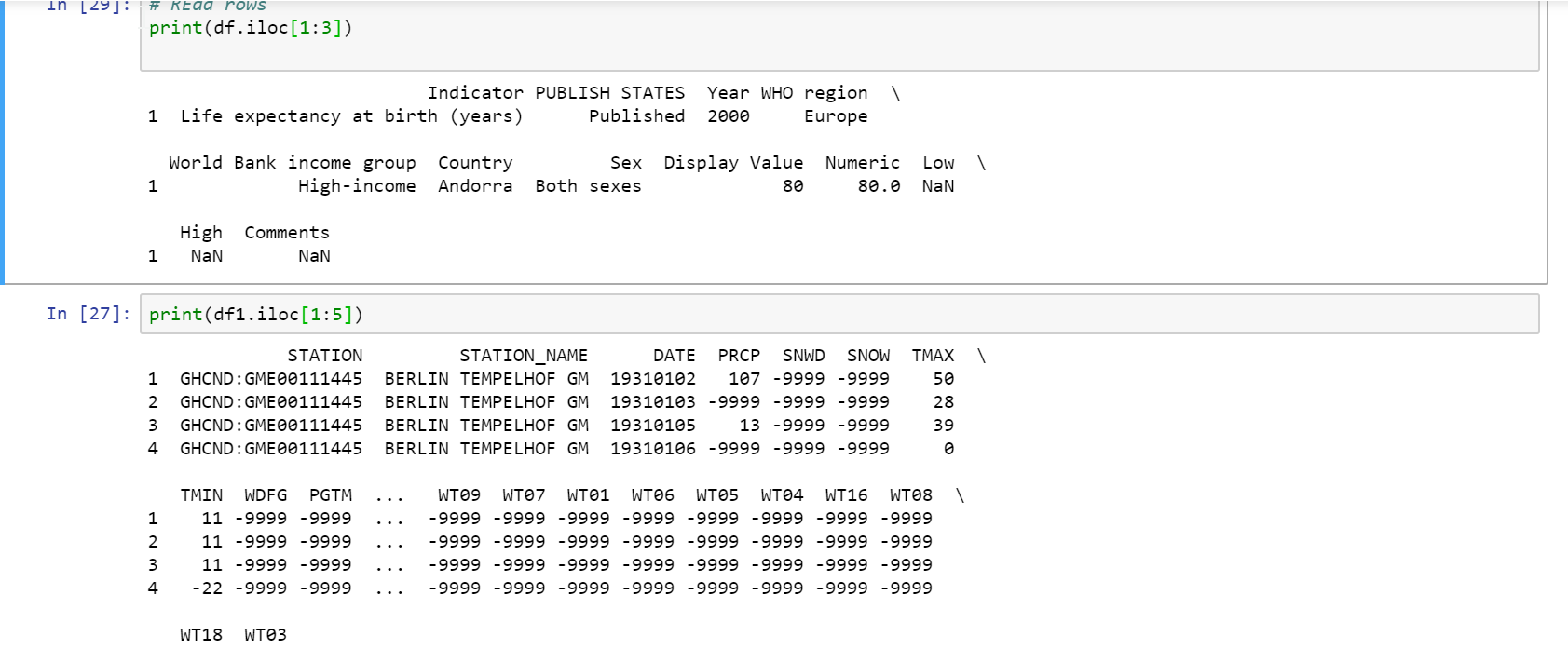






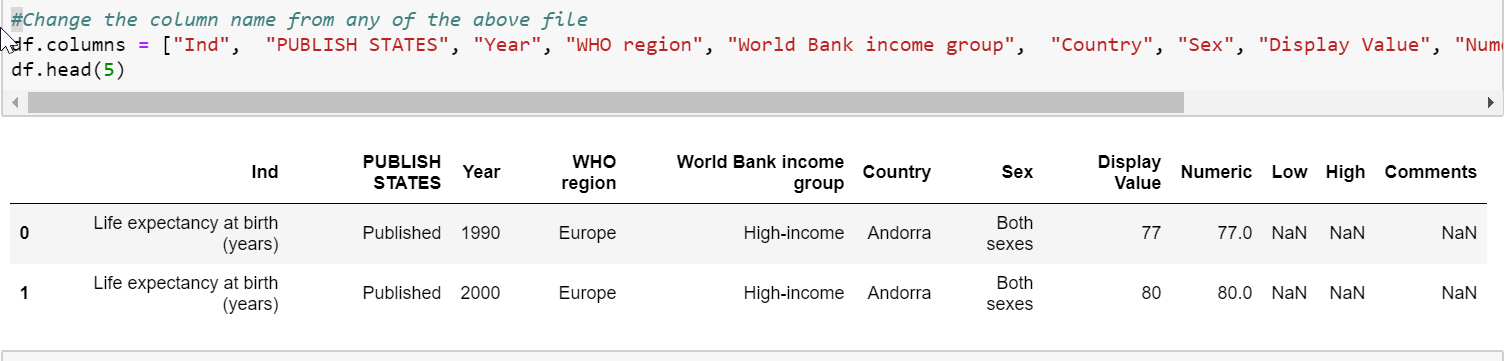
Problem 2

* Read rows

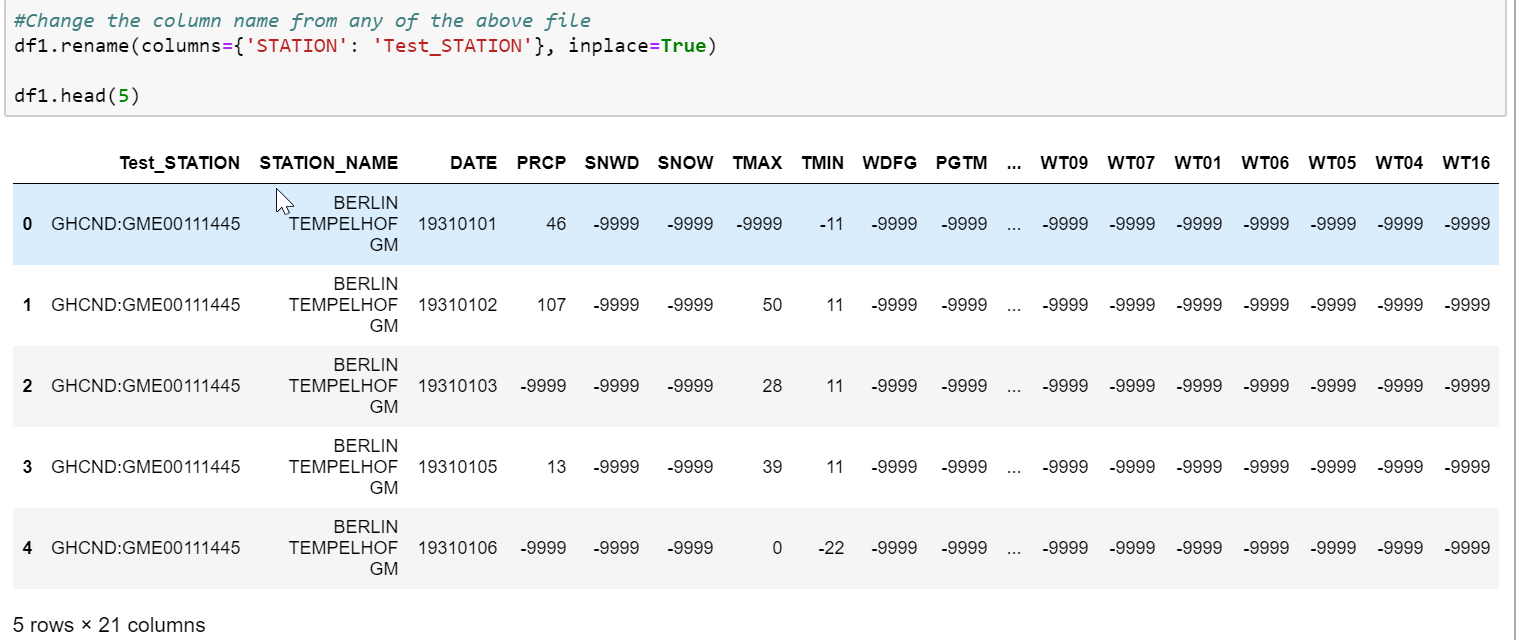


Problem 3:

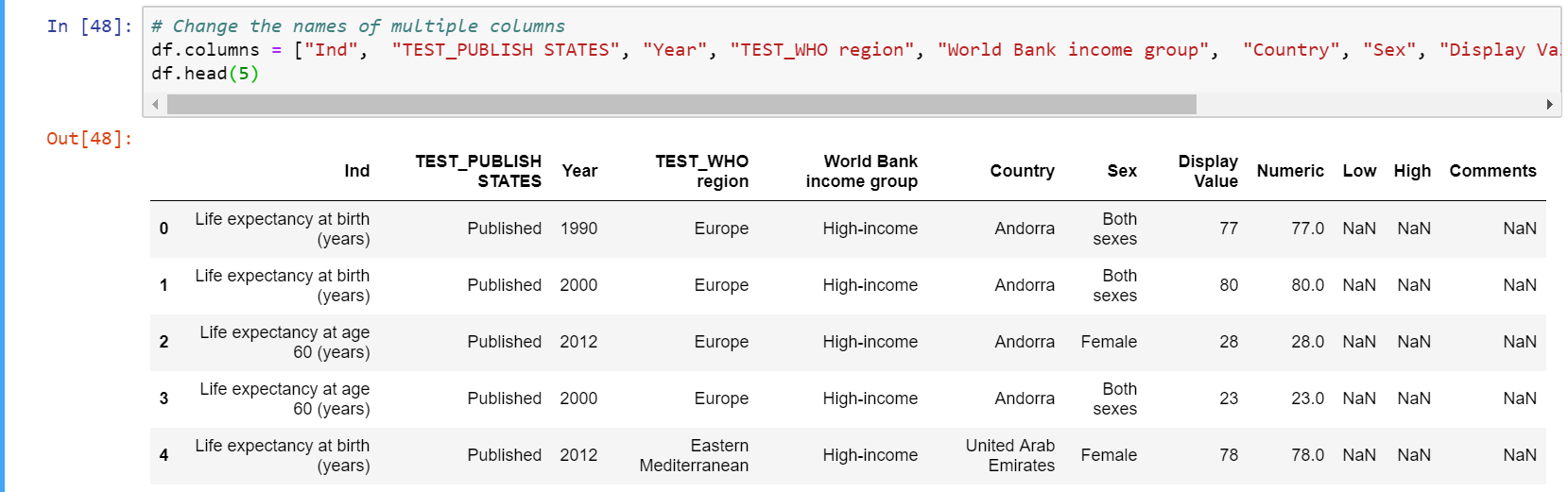
Change the column name for any file



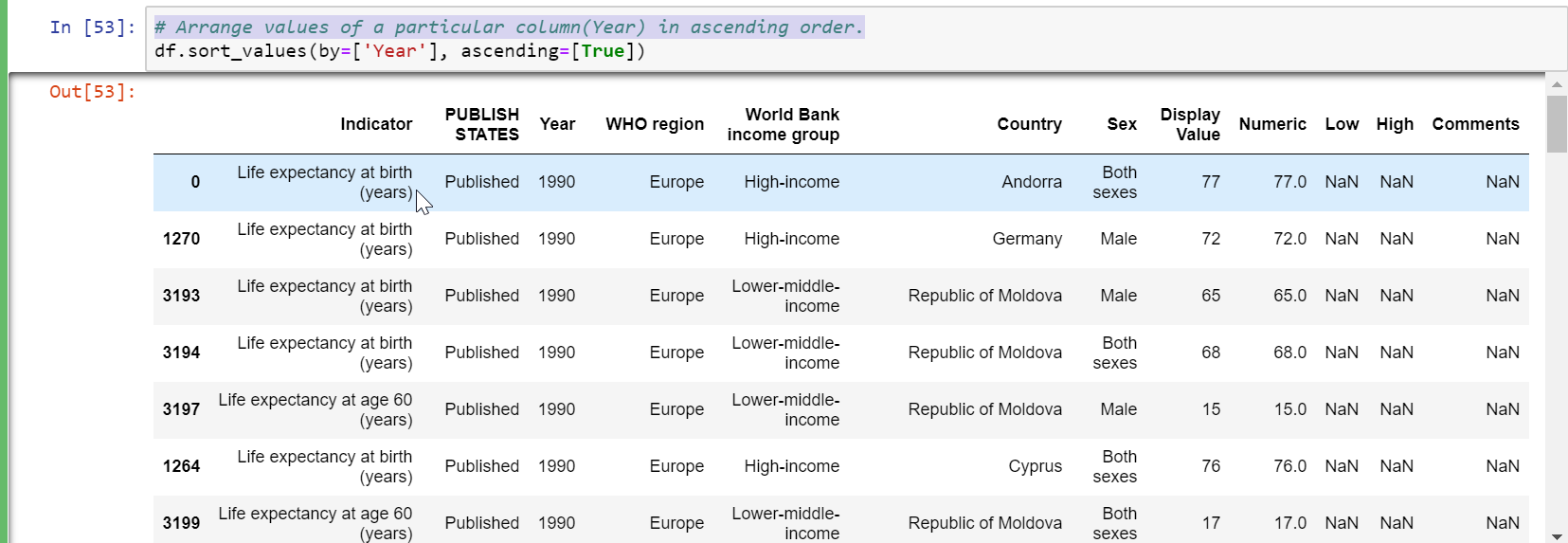
Change the column name perment basis for any file



Change multiple columns



# Arrange values of a particular column(Year) in ascending order.



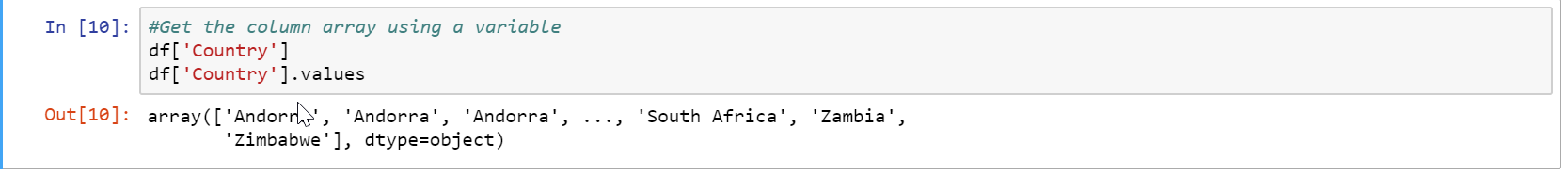
Arrange multiple column values in ascending order (Indicator, Country)



Make **country** as the first column of the dataframe



Get the column array using a variable



**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 =

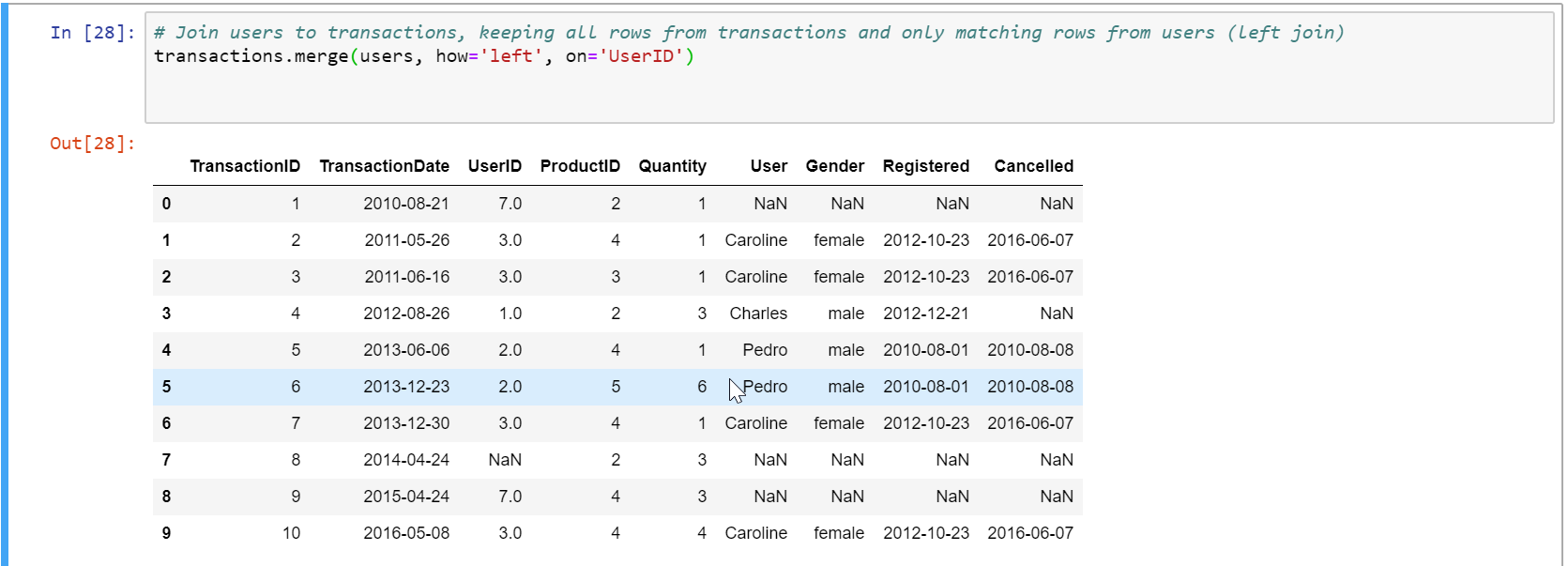
pd.read\_csv('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data / products.csv' )

transactions =

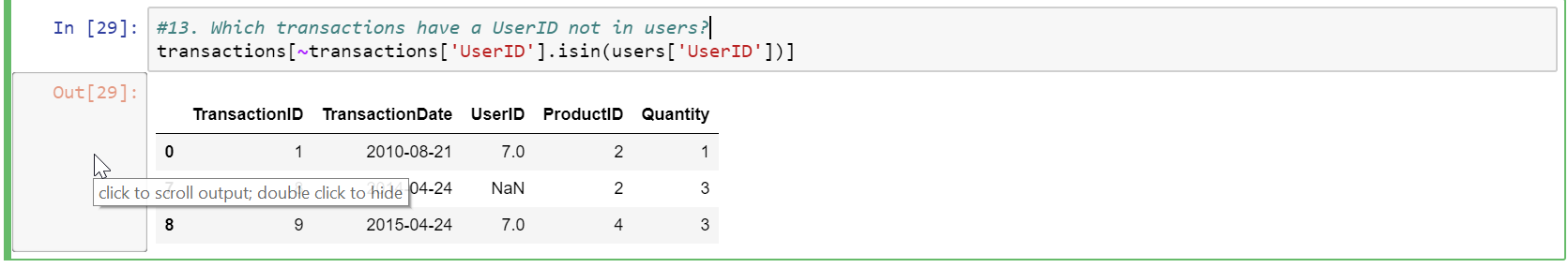
pd.read\_csv('https://raw.githubusercontent.com/ben519/DataWrangling/master/Data / transactions.csv') users.head() sessions.head() transactions.head()

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

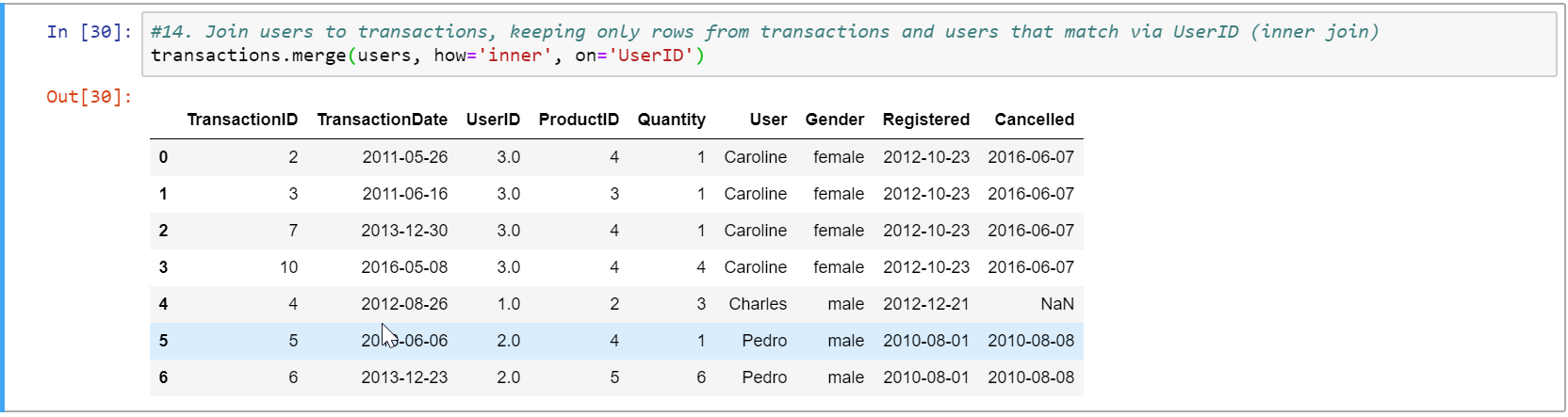
**Join users to transactions, keeping all rows from transactions and only matching rows from users (left join)**

****

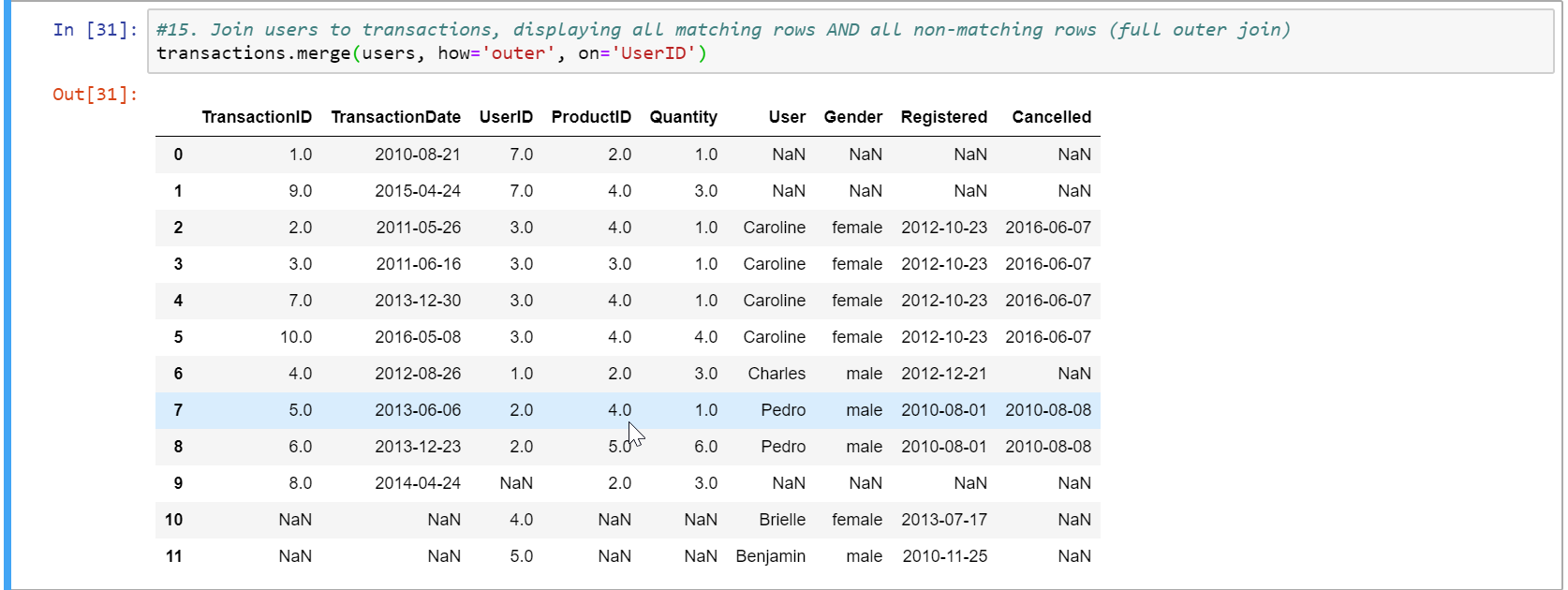
13. Which transactions have a UserID not in users?



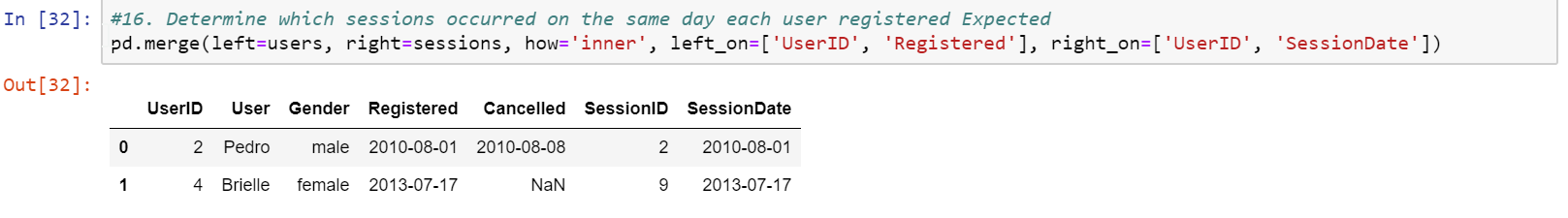
14. Join users to transactions, keeping only rows from transactions and users that match via UserID (inner join)



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



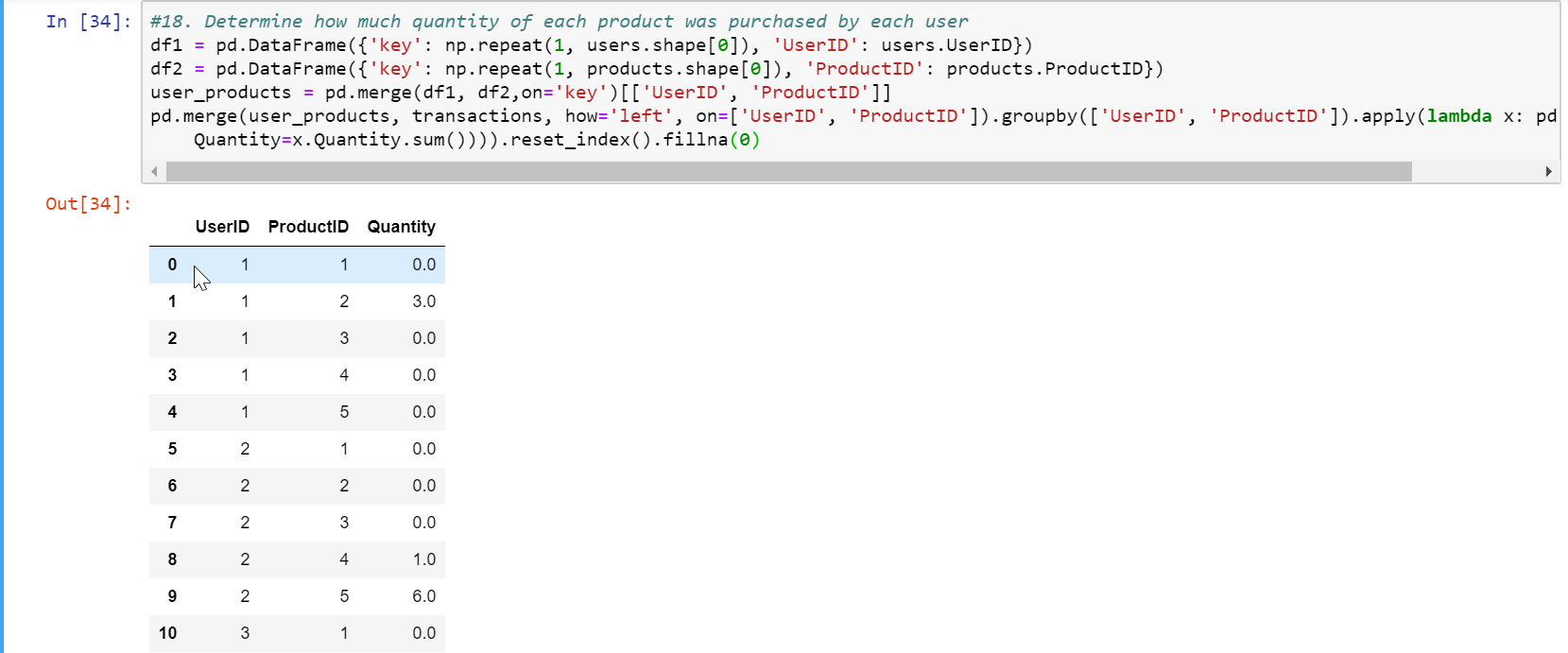
16. Determine which sessions occurred on the same day each user registered **Expected**

****

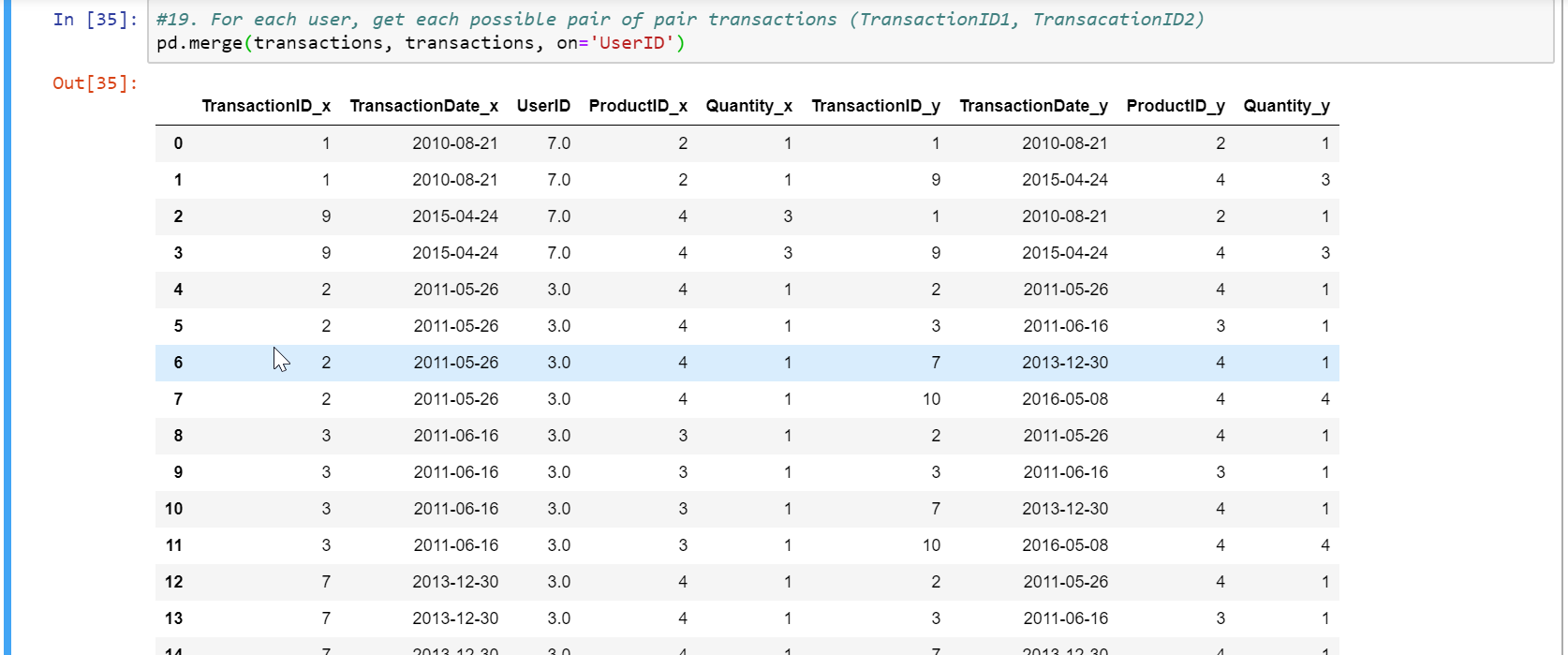
17. Build a dataset with every possible (UserID, ProductID) pair (cross join)



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



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



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

