Project1 Data Mangling – Session14: Answers with screen shots

**2. Problem Statement**

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/dat

a/chp3/data-text.csv') **df**.head(2)

**df1** =

pd.read\_csv('https://raw.githubusercontent.com/kjam/data-wrangling-pycon/master/d

ata/berlin\_weather\_oldest.csv') **df1**.head(2)

1. Get the Metadata from the above files.

**Expected Output:**

2. Get the row names from the above files.

**Expected Output:**

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

**Expected Output:**

4. Change the column name from any of the above file and store the changes made

permanently.

**Expected Output:**

5. Change the names of multiple columns.

**Expected Output:**

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

**Expected Output:**

7. Arrange multiple column values in ascending order.

**Expected Output:**

8. Make **country** as the first column of the dataframe. **Expected Output:**

9. Get the column array using a variable **Expected Output:**

10. Get the subset rows 11, 24, 37 **Expected Output:**

11. Get the subset rows excluding 5, 12, 23, and 56 **Expected Output:**

**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:**

13. Which transactions have a UserID not in users?

**Expected Output:**

14. Join users to transactions, keeping only rows from transactions and users that match

via UserID (inner join) **Expected Output:**

15. Join users to transactions, displaying all matching rows AND all non-matching rows

(full outer join)

**Expected Output:**

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

**Output:**

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

**Output:**

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

**Output:**

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

TransacationID2)

**Expected Output:**

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

**Output:**

21. Test to see if we can drop columns

**Code with Output :**

my\_columns = list(data.columns) my\_columns

['UserID',

'User',

'Gender',

'Registered',

'Cancelled',

'TransactionID',

'TransactionDate',

'ProductID', 'Quantity'] list(data.dropna(thresh=int(data.shape[0] \* .9), axis=1).columns)

#set threshold to drop NAs

['UserID', 'User', 'Gender', 'Registered'] missing\_info

= list(data.columns[data.isnull().any()]) missing\_info

['Cancelled', 'TransactionID', 'TransactionDate', 'ProductID', 'Quantity']

**//for col in missing\_info:**

num\_missing = data[data[col].isnull() == True].shape[0] print('number

missing for column {}: {}'.format(col, num\_missing)) **Output: Count of**

**missing data**

number missing for column Cancelled: 3 number

missing for column TransactionID: 2 number

missing for column TransactionDate: 2 number

missing for column ProductID: 2 number missing

for column Quantity: 2

//**for col in missing\_info:**

num\_missing = data[data[col].isnull() == True].shape[0] print('number missing for

column {}: {}'.format(col, num\_missing)) #count of missing data

for col in missing\_info:

percent\_missing = data[data[col].isnull() == True].shape[0] /

data.shape[0] print('percent missing for column {}: {}'.format( col,

percent\_missing))

**Output of percentage missing data**

percent missing for column Cancelled: 0.6 percent

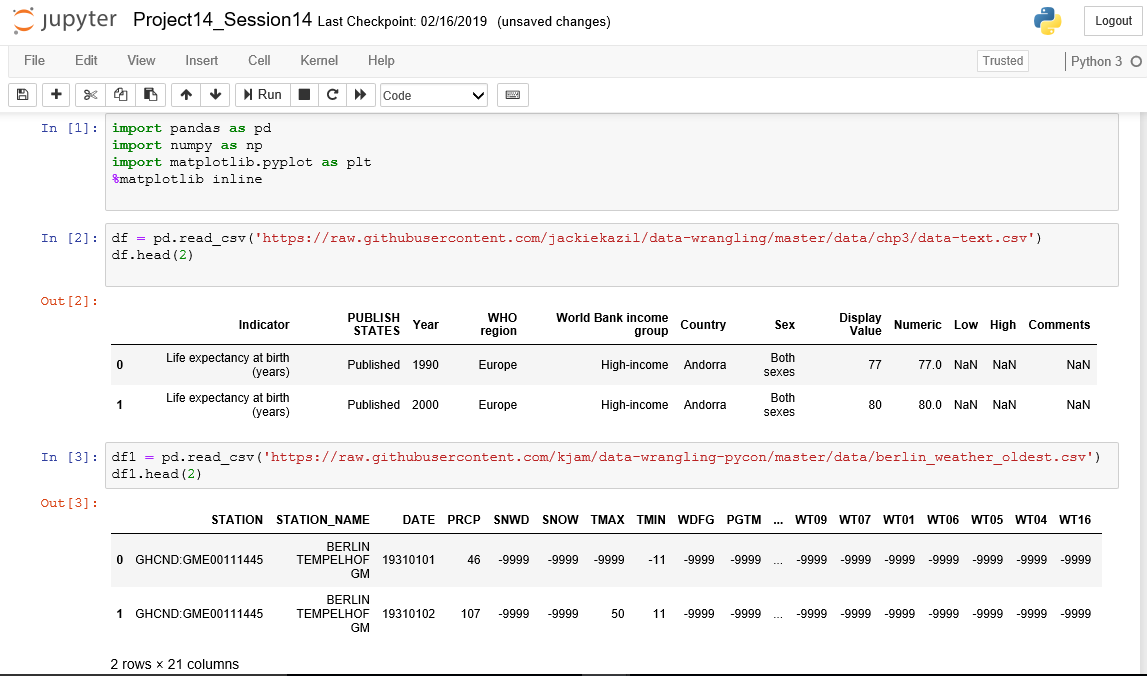
missing for column TransactionID: 0.4 percent

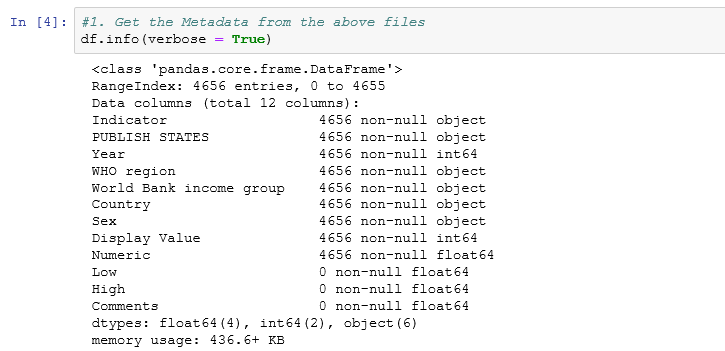
missing for column TransactionDate: 0.4 percent

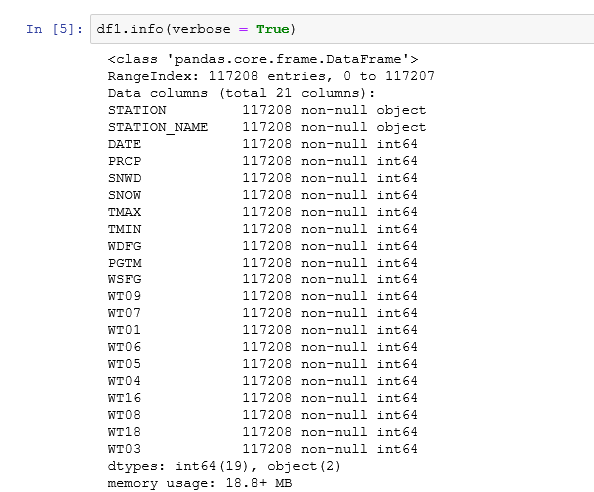
missing for column ProductID: 0.4 percent missing

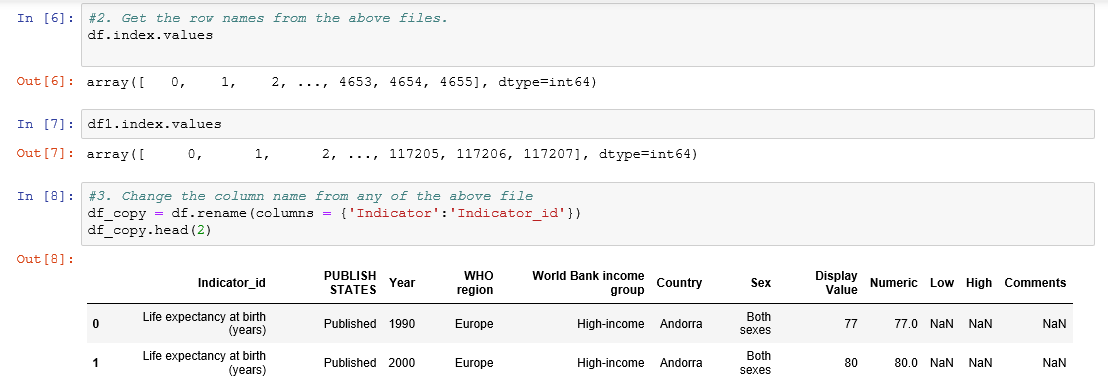
for column Quantity: 0.4

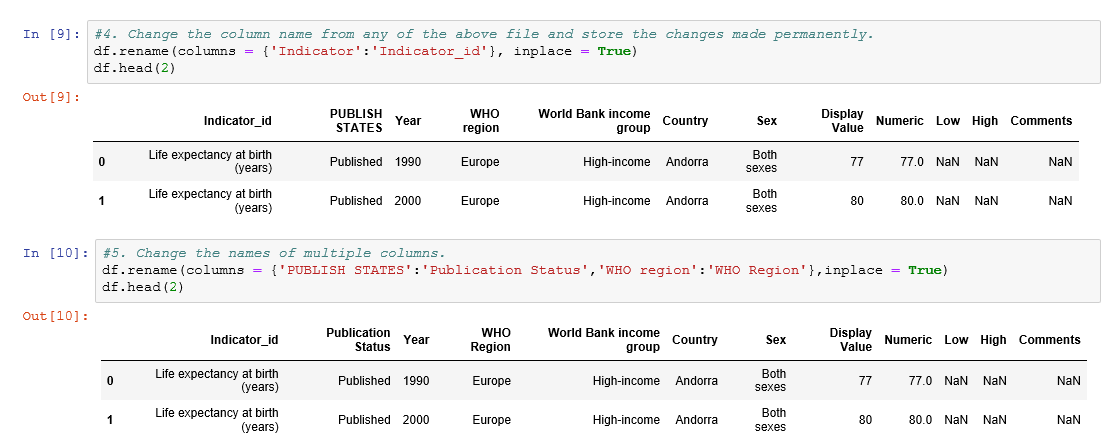
**3. Output**

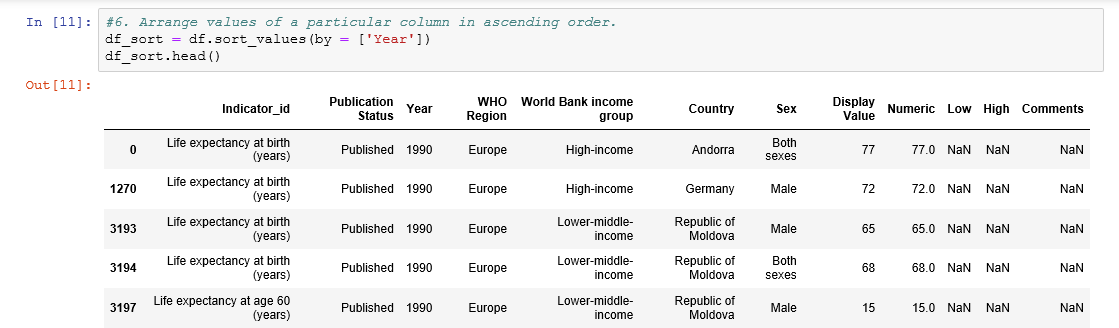
****

****

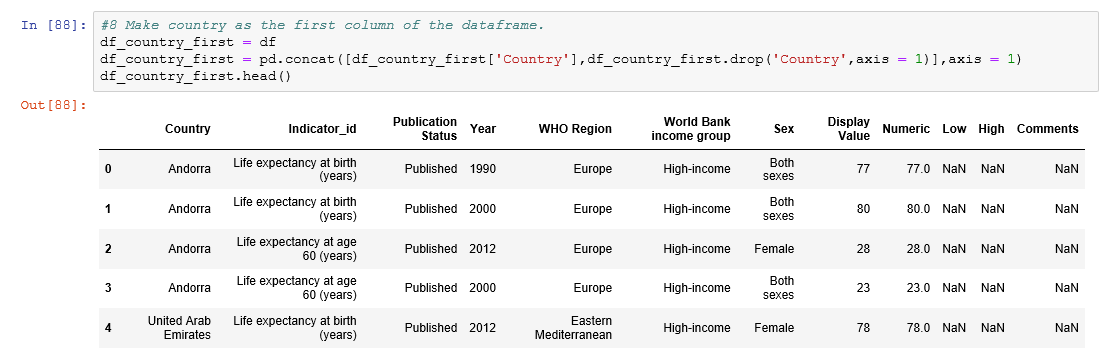
****

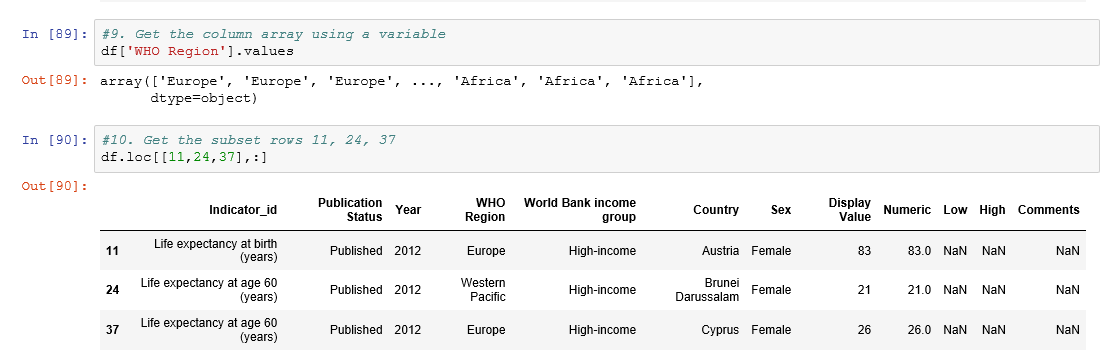
****

****

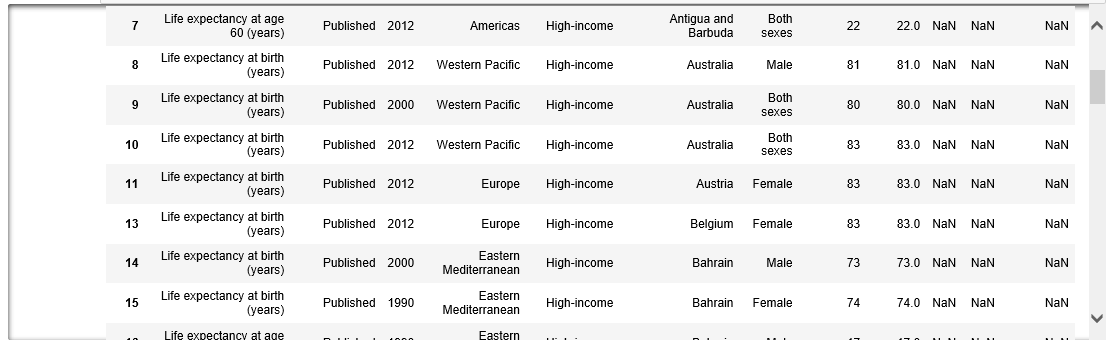
****

****

****

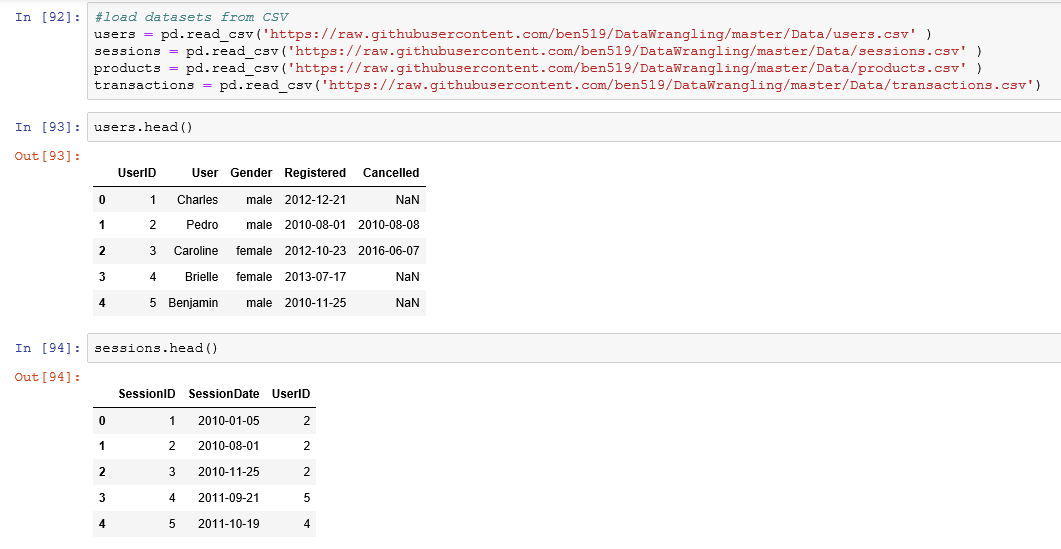
****

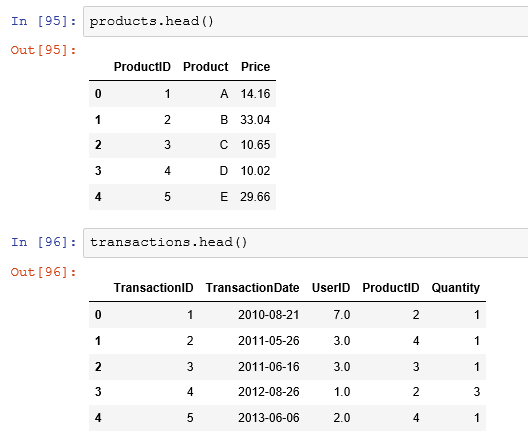
****

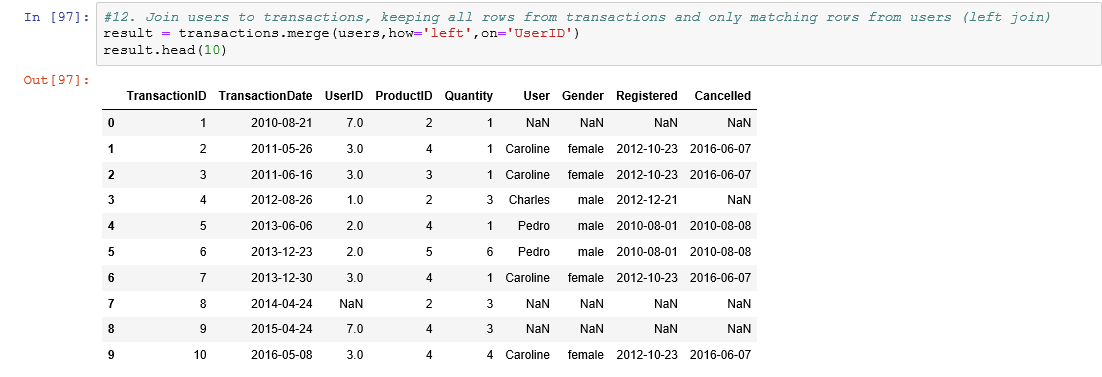
****

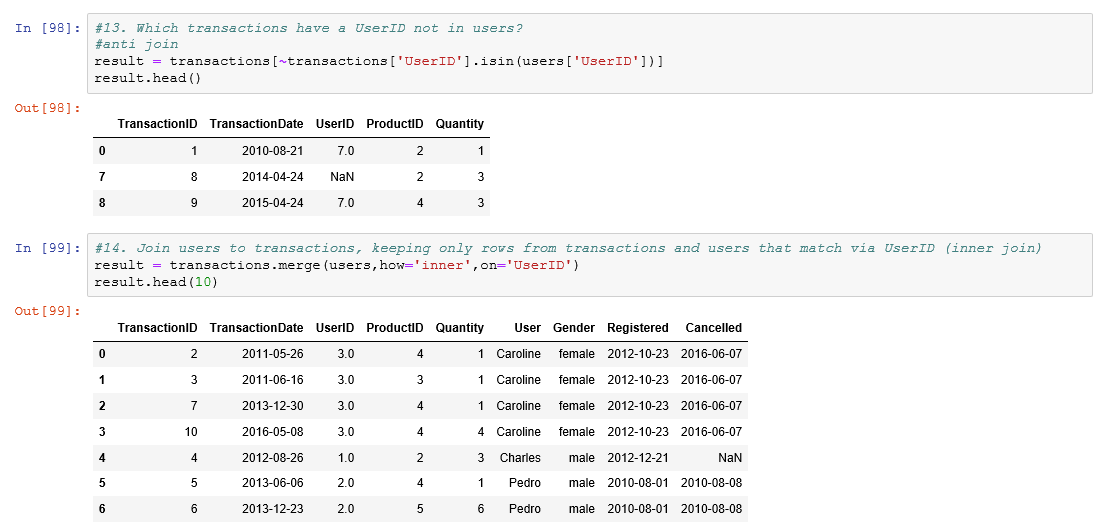
****

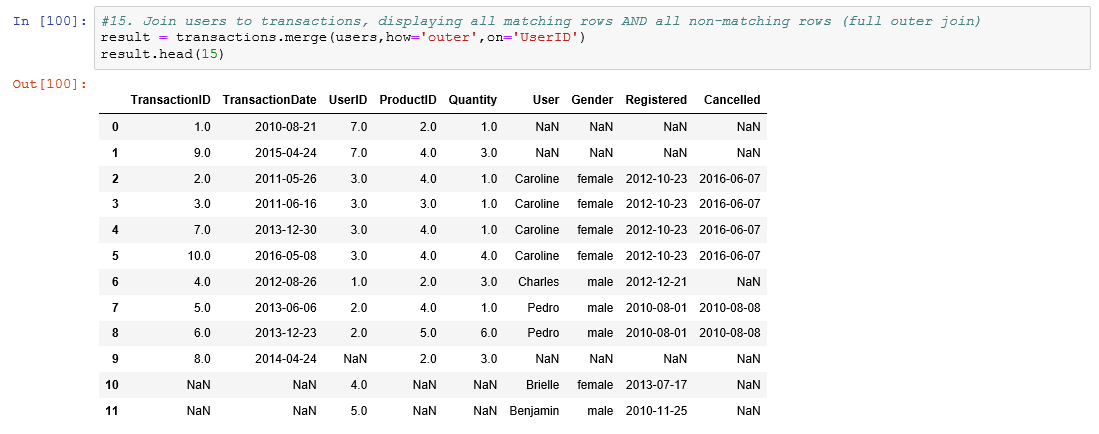
****

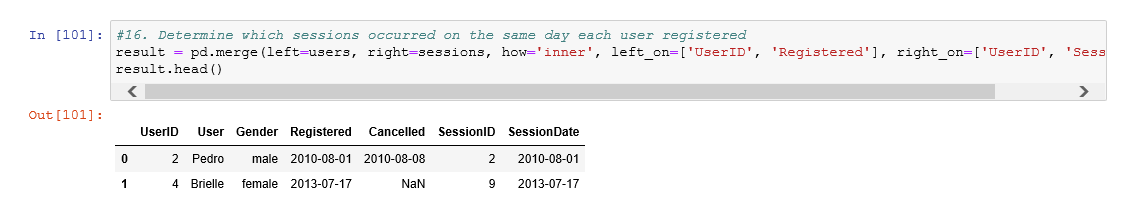
****

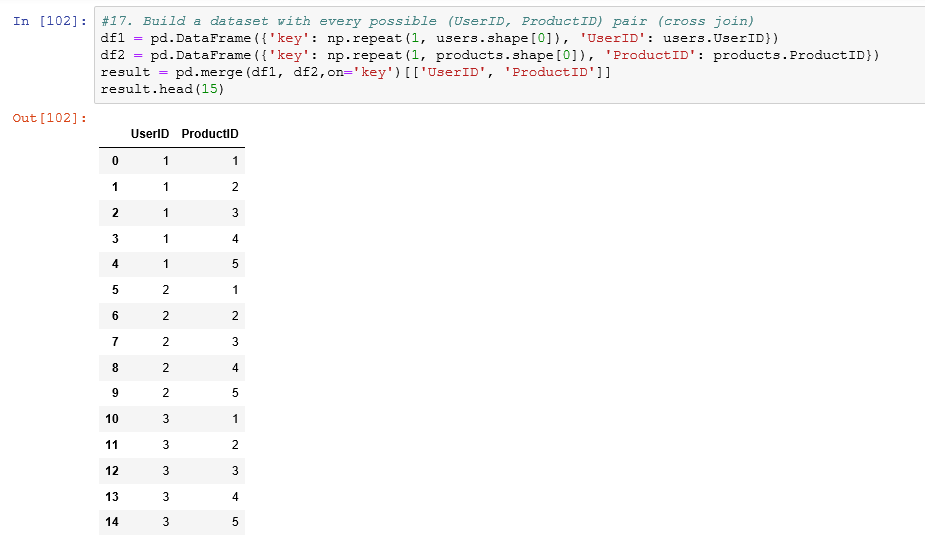
****

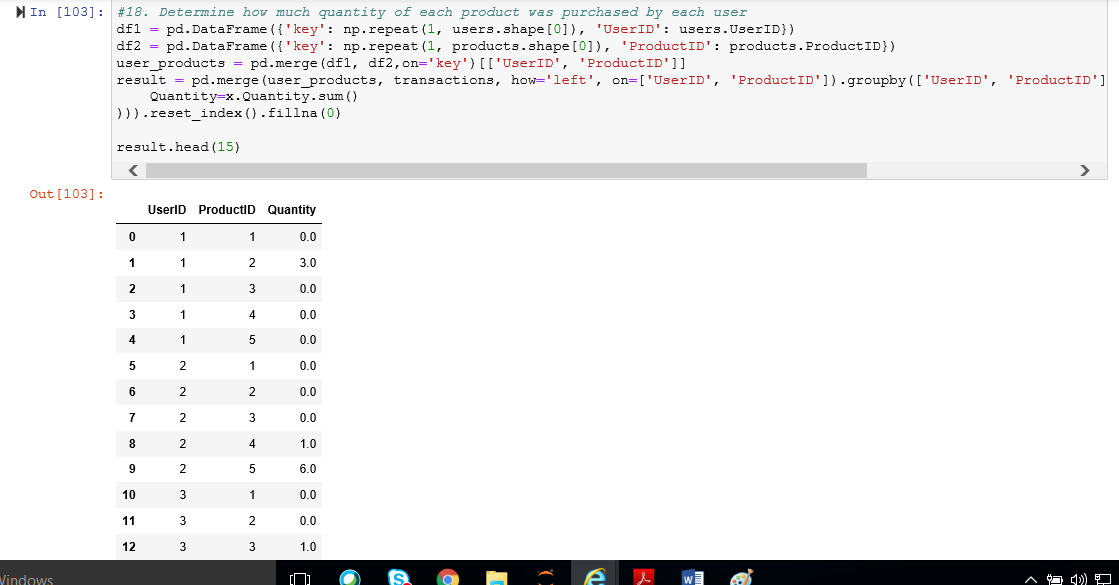
****

****

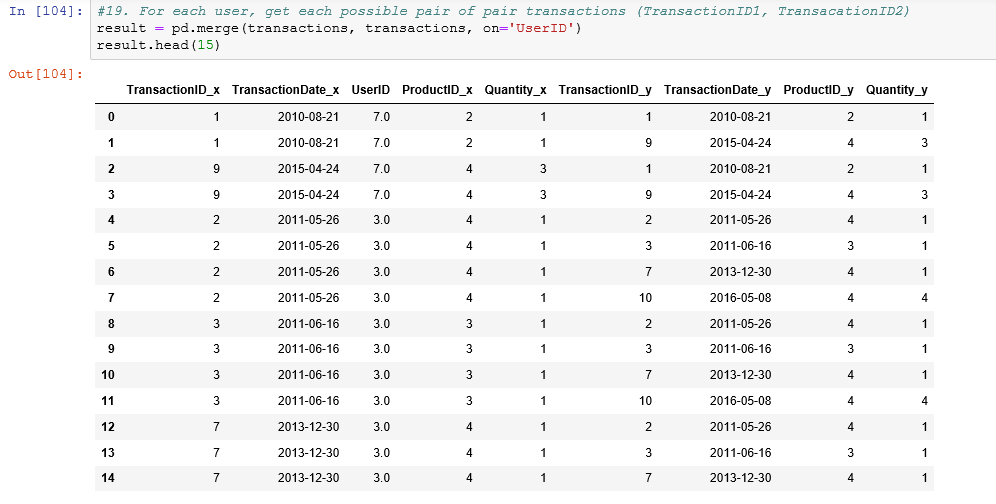
****

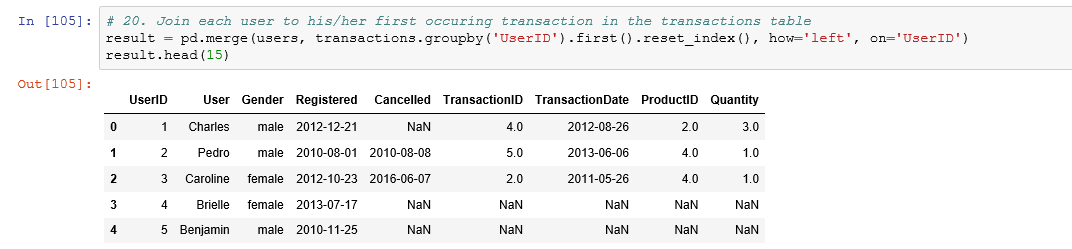
****

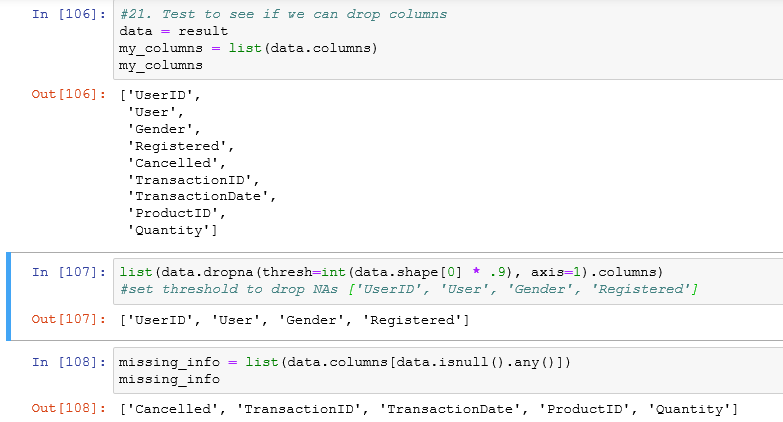
****

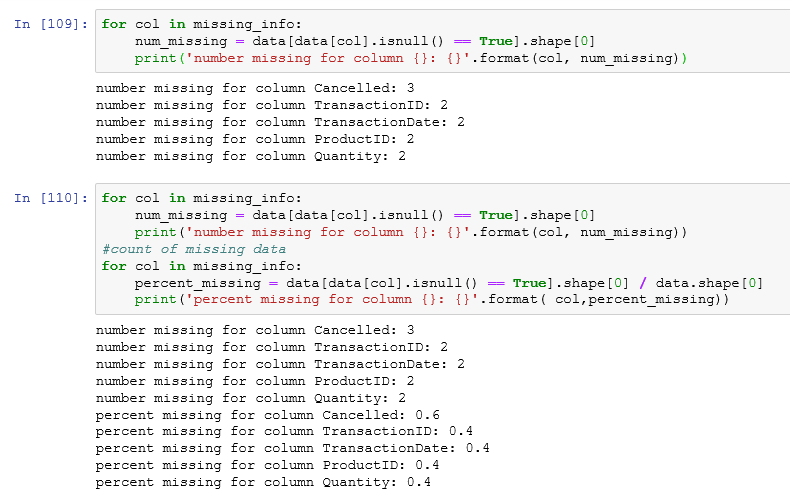
****

****

****

****

****

****