Big Data

Worksheet 3

Name: Prakash Dahal

Student ID: 1828421

Remainder: 2

Making a separate worksheet3 folder through putty inside my directory (**home/1828421**) in the server.

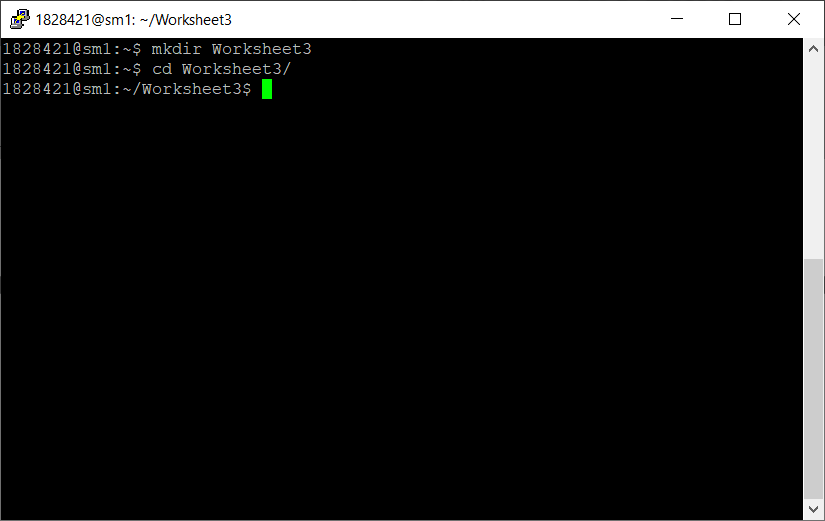


Figure : Worksheet folder

Copying Population.java file into this directory.

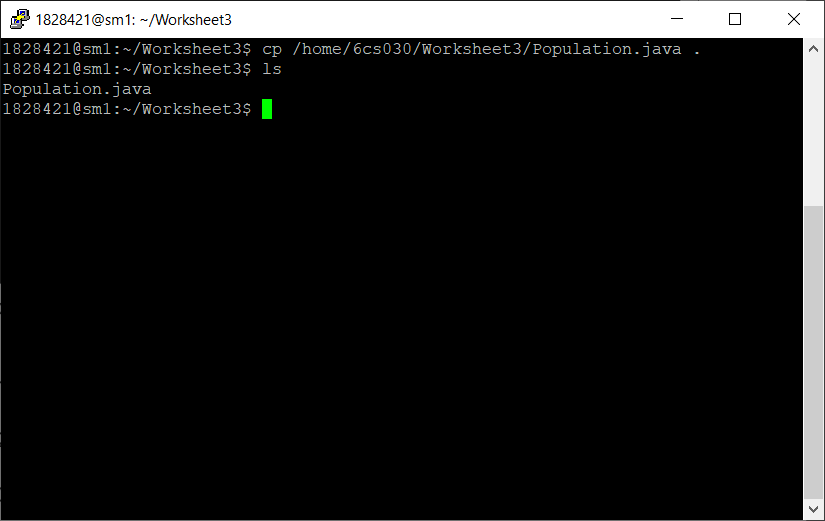


Figure : Copying java file and displaying

Now, making required changes in the copied file.

# Java and Hadoop

Population.java file has to be edited. Therefore, it is downloaded using filezilla and stored in my local folder to edit.

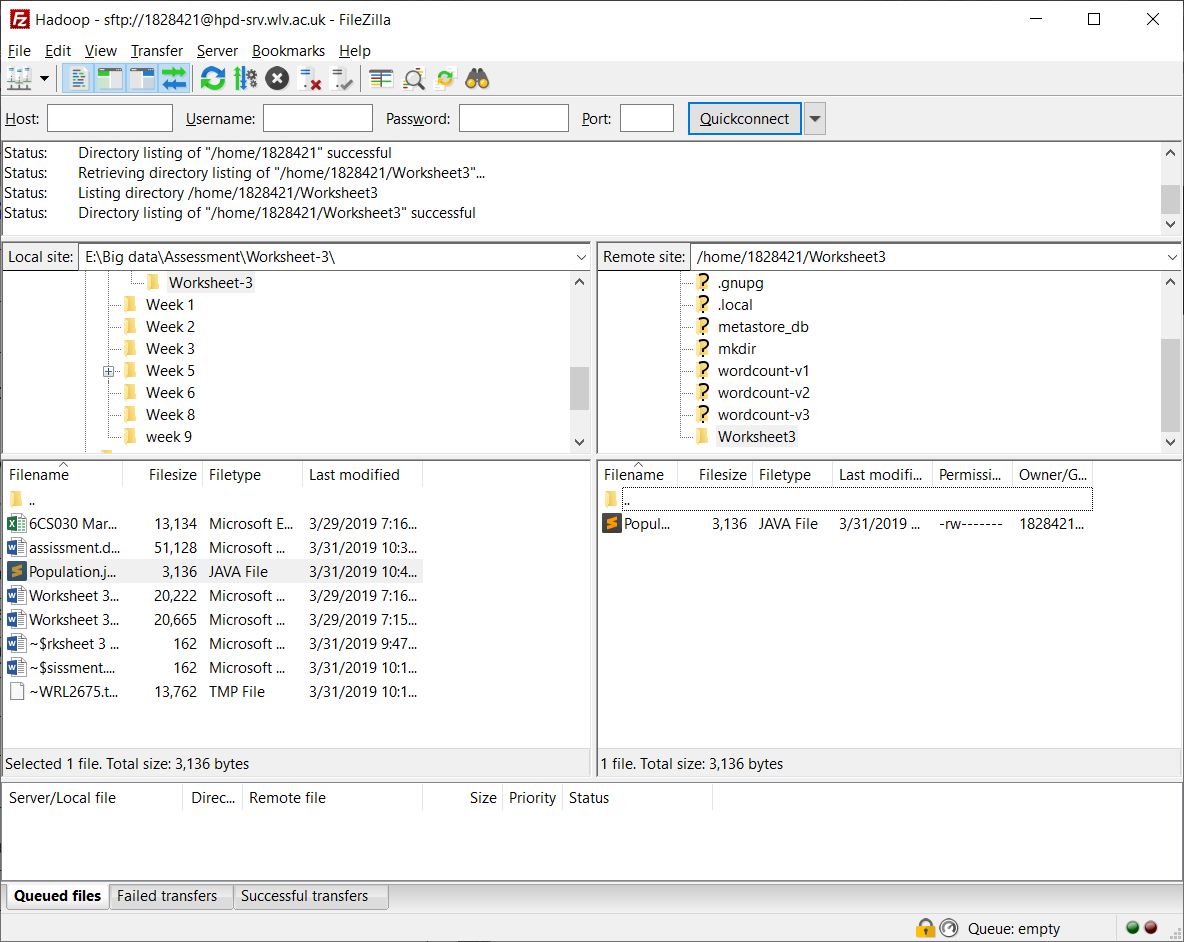


Figure : Saving file locally

Opening the locally saved file with sublime.

Changing the main Class name as **NoQuals** as per the given class name. Changing Mapper class as **PDMapper** and Reducer class as **PDReducer** where first two alphabets (**PD**) are my initials. Finally fixing the changed classes name in the job section.

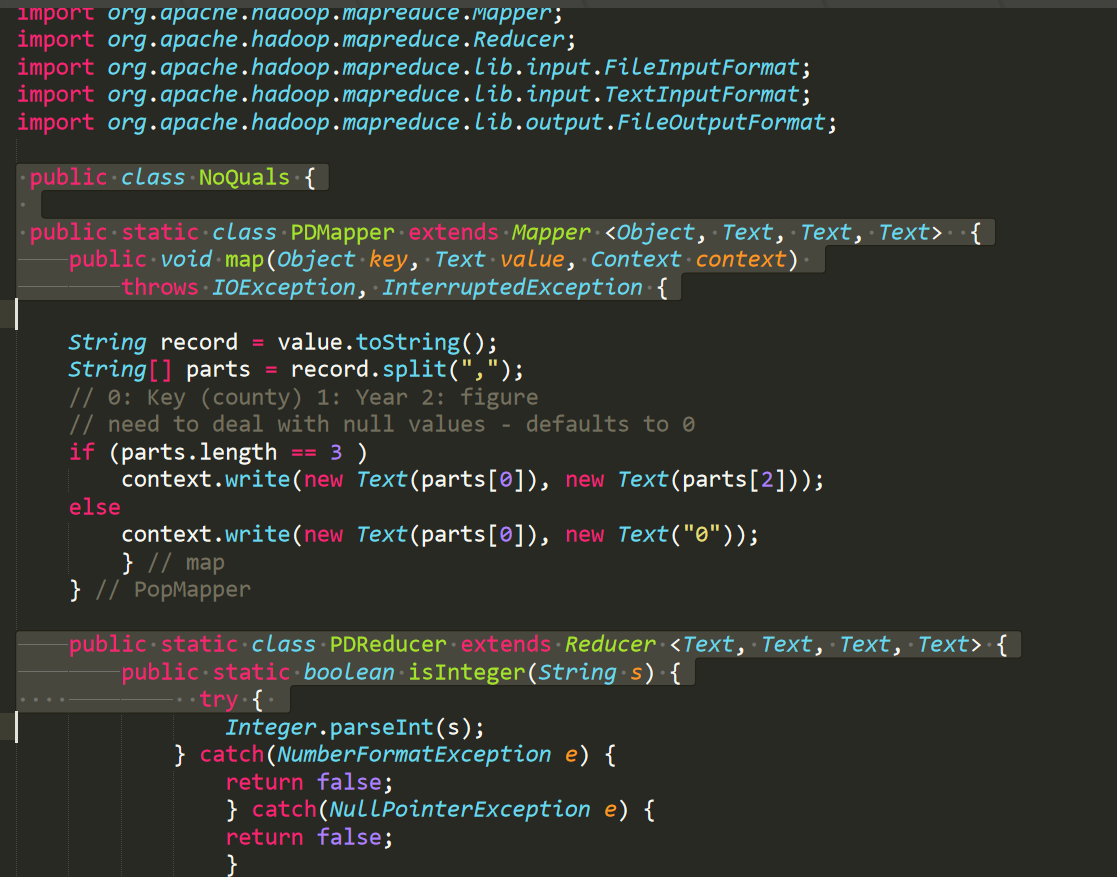


Figure : Updated class, mapper and reducer name

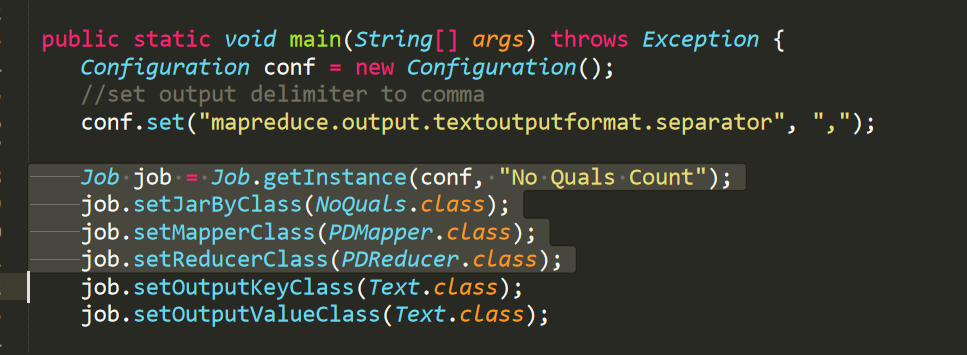


Figure : Fixing classes names in Job

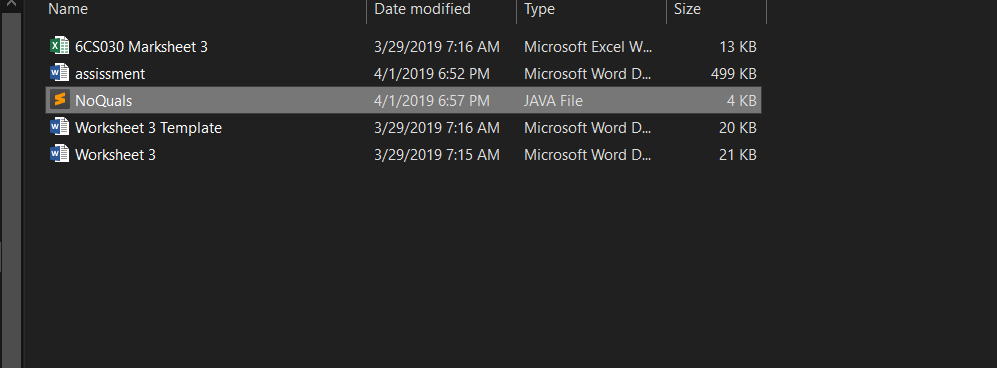
Now renaming the file name as the main class name since java class name and file name has to be same.

Figure : Renaming Java file name as NoQuals

Now uploading this updated file into the server (**Worksheet3**) folder through filezilla.

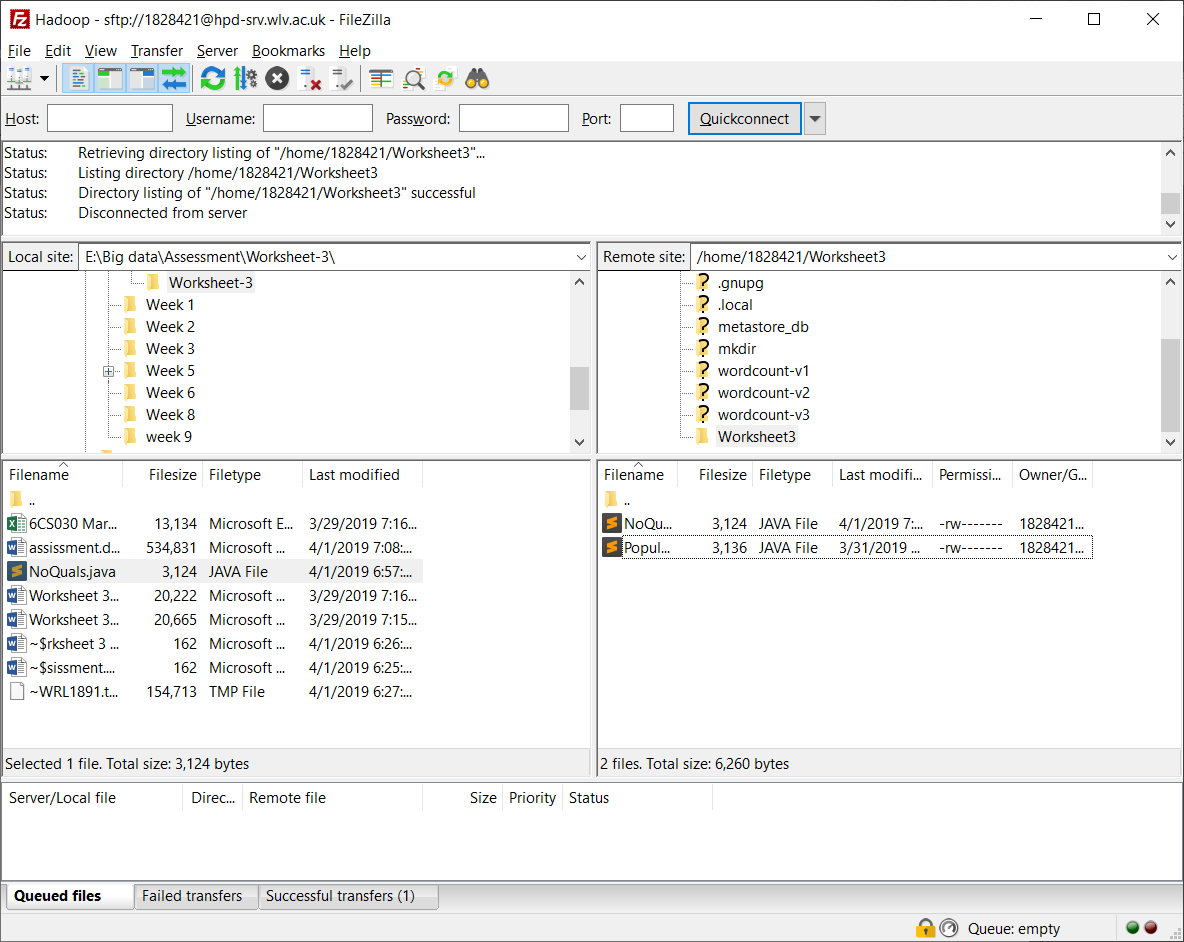


Figure : Uploading edited file

# Running the code

While running the code, at first java file is compiled and jar file is created then the file is stored into input and output folders and then running the program.

1. Compiling the NoQuals.java

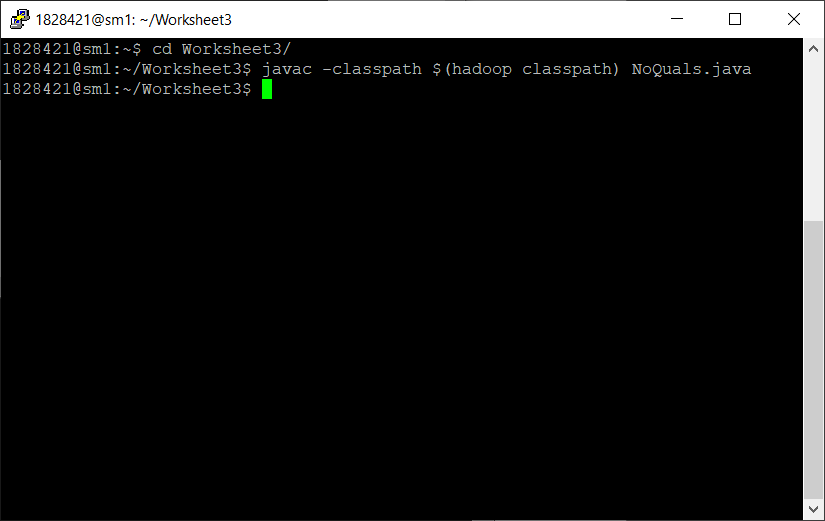


Figure : Compiling NoQuals.java

1. Creating jar file

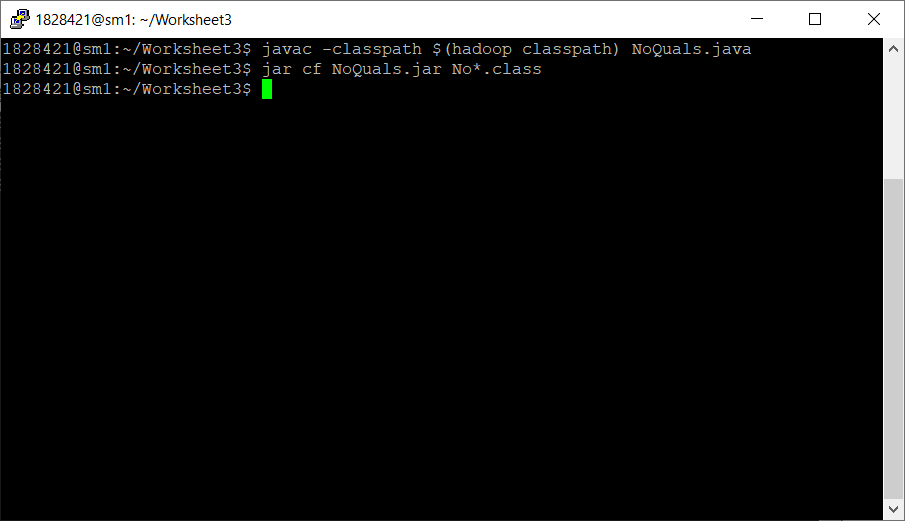


Figure : Creating jar file

1. Creating input and output directories and storing files

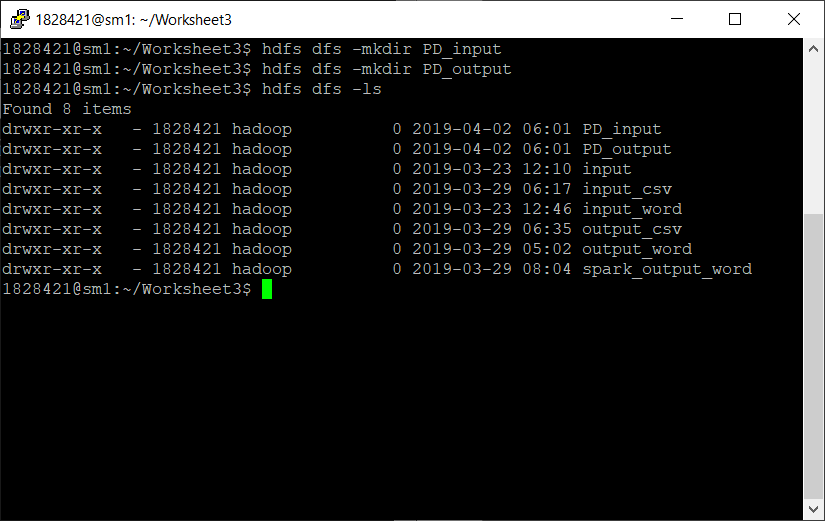


Figure : Making directories

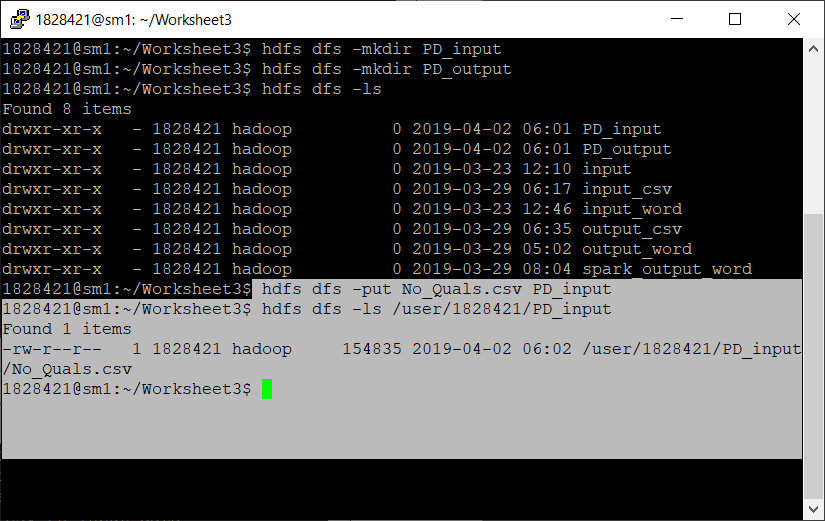


Figure : Putting file into directory

1. Running the program

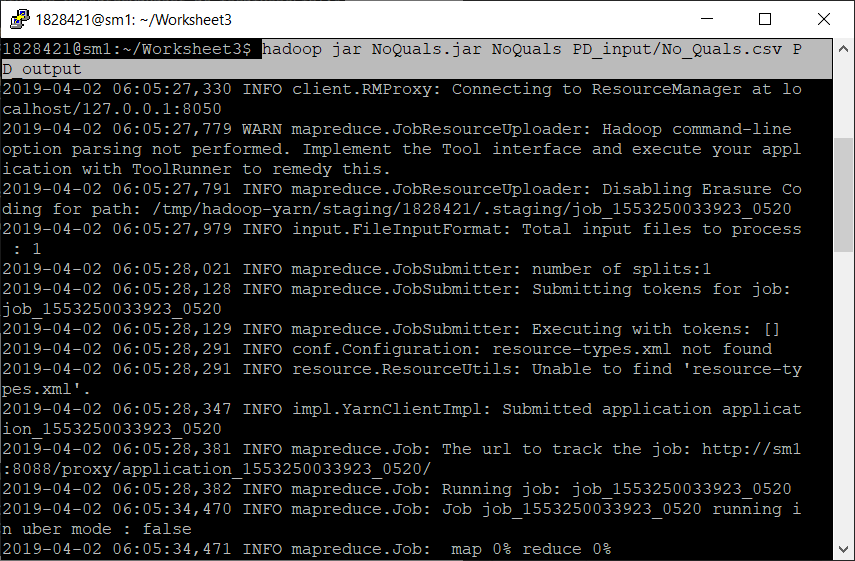


Figure : Running jar file

1. Output directory files:

Displaying files inside the output folders.

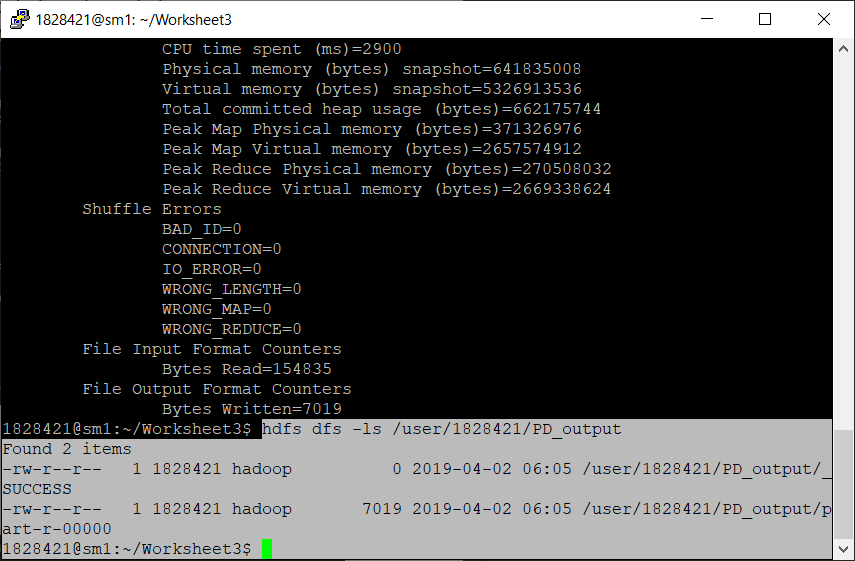


Figure : files in output folders

1. Displaying the result

To display the result more or -cat query can be used but here cat is used to display the result.

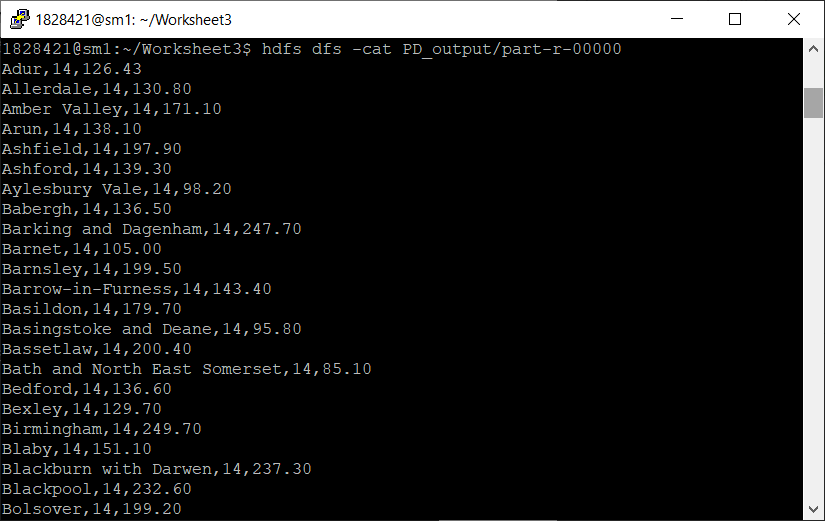


Figure : Result display

# Apache Spark

Running apache spark. To run apache spark, **pyspark** command is used.

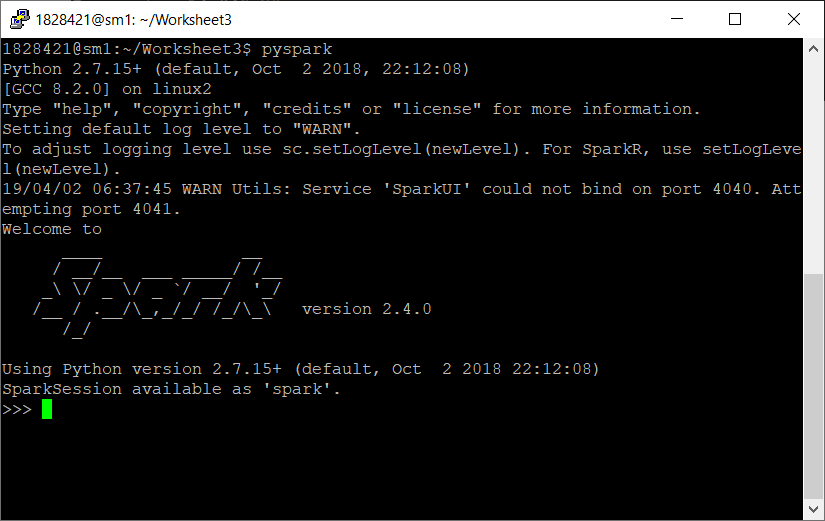


Figure : Running spark

1. Loading CSV file

Loading csv file from the hdfs and displaying the result.

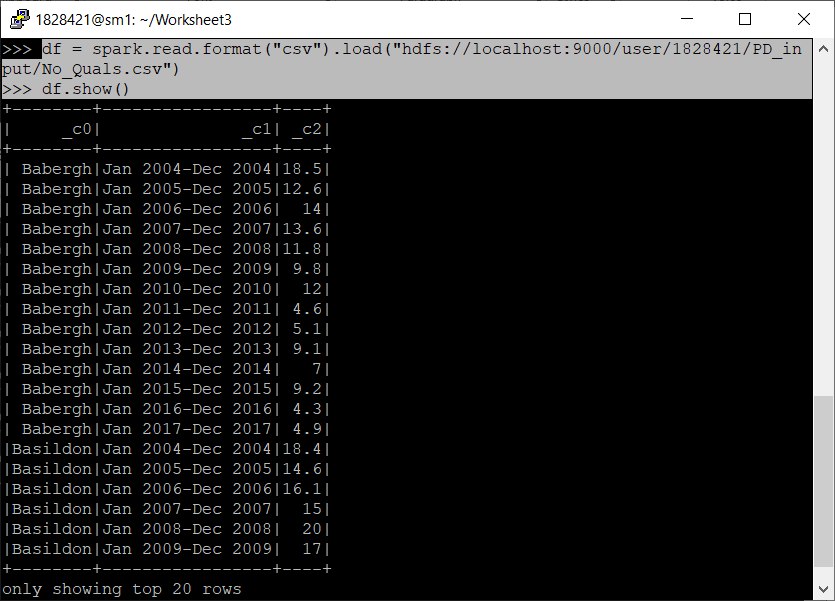


Figure : CSV load

1. Data manipulation query

Manipulation of data is shown below of csv as well as sql.

1. CSV:

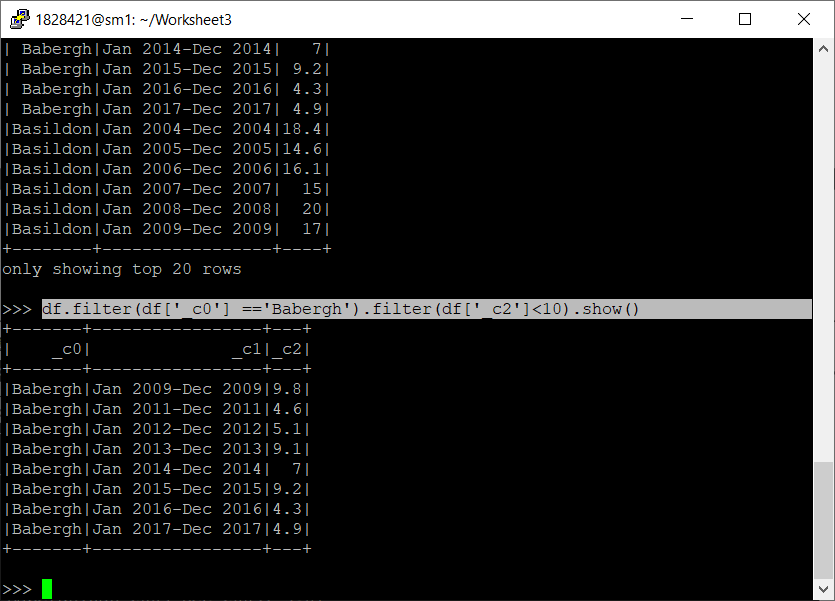


Figure : CSV filter

1. SQL:

Here again fresh file is loaded from the hdfs so that the one manipulation may not affect another, it is good to have fresh and separate file. To load sql, at first it has to be converted into view. Then SQL query is implemented.

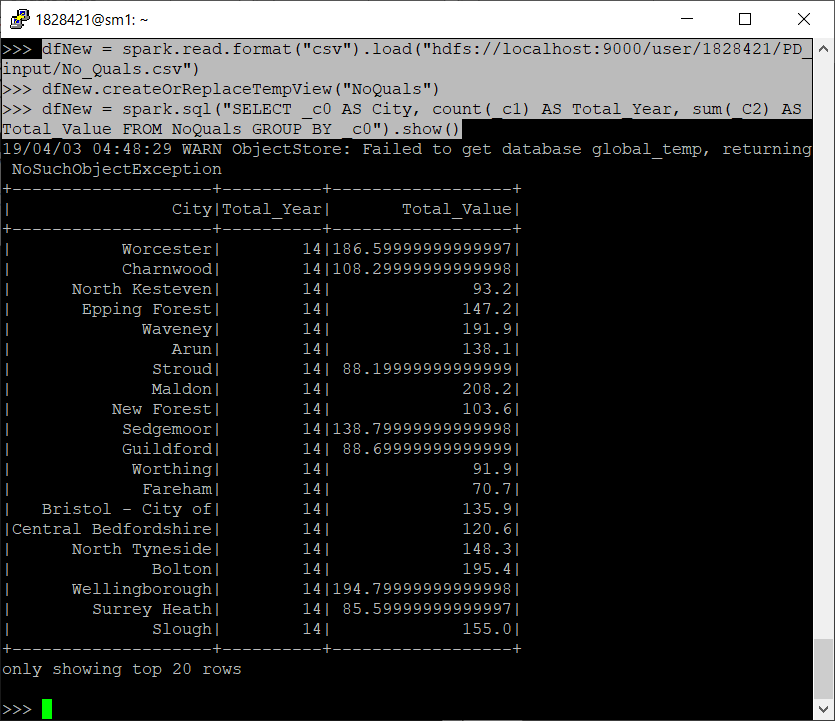


Figure : SQL filter

# Advantage and Disadvantage

1. **Advantage:**

*Fault tolerance:*

Hadoop is used for processing large number of data or files. While processing sometimes some issue can arise and node may fail to process. So, in this case the whole operation will be stopped, and the task will be incomplete. Therefore, in order to maintain this, Hadoop uses technique of Fault tolerance which resolve this problem by neglecting or overcoming that module. There are many techniques for this like check-point and recovery, heartbeat message and so on but mainly **Data Replication** and or **Shredding** is used to maintain fault tolerance.

In data replication, the copy of same data is stored in multiple places where in shredding, data are split and stored in different places either row or column wise (Mugunthan, 2015).

1. **Disadvantage:**

*Slow and not suitable for small data:*

Hadoop is built for processing large data or files. So, it goes from mapper to reducer and gives the output. It is suitable for large number of data but not suitable for small data since it has to pass through mapper and reducer which make the performance very slow. On the other hand, Hadoop is slower than Apache Spark as well which is one of the major limitations of Hadoop.