STSCI 5065 HW 4

(Assigned: 4/23/2019; Due: 5/3/2019 at 11:59PM)

What and how to turn in:

Submit electronically to the course website: an MS Word or PDF file named **LastName_FirstName _STSCI5065HW4**, containing all the steps you performed, including the code, and your answers to the questions, listed in the order of the questions.

In this homework, there are two problem sets. You will practice how to use Apache Pig to achieve the same results as you did in HW3 when you used Apache Hive. Write all your Pig scripts in the Ambari graphical interface. Unless otherwise directed, list all your code in a separate paragraph with **blue fonts** right below your homework questions, for example:

```
A = load "mydata.txt" using …;
Dump A;
```

All your screenshots of Pig script results must contain job IDs.

Problem Set 1

You are given a set of two text files of New York Stock Exchange (NYSE) data, NYSE_daily_prices.csv and NYSE_dividends.csv. You are required to create some Pig relations based on the text files, and then do some analysis of the data using Pig scripts. (60 points total)

A. (6 points) Download the two data files from the course website to a local OS directory. Create an HDFS directory, **HW4**, right below the CentOS root. Use the Files View in Ambari to upload the two data files into HW4. Display a screenshot (in Ambari) showing all these files in the /HW4 (your screenshot should show this directory name).

B. (14 points) Create two relations **stocks_prices** and **stock_dividends**. The **stocks_prices** relation contains columns in the following order: exchng, symbol, ymd, price_open, price_high, price_low, price_close, volume, and price_adj_close. The data types of exchng, symbol and ymd are chararray (same in the **stock_dividends** relation), volume's data type is int, and other columns are of double data type. The **stock_dividends** contains the columns in the following order: exchng, symbol, ymd, and dividend. The data type of dividend is double. Dump these two relations, and attach screenshots to show your results (first 10 rows only).

C. (7 points) Use the stock_prices relation to find out the number of entries of each stock symbol. You are only required to show two screenshots of the first 10 rows and the last 10 rows of your result (sorted by stock symbol).

D. (6 points) Calculate the average stock opening price of each stock symbol rounded to 4 decimal points (same below). Use an alias, ap, for the average price column. Attach a screenshot of the last ten rows of the output.

E. (10 points) Find out the stock symbols that have the highest and lowest average opening prices, respectively. You are required to use the relation created in step D. Attach the screenshots of the query results. Each of your final results should only contain two columns, i.e., the symbol and the highest or lowest average opening price.

F. (10 points) Create a relation called stock_div from NYSE_dividends.csv. Find out the stock symbol that has the highest dividends. What are the date and the opening price of that stock when that happened? Attach the screenshots of your results.

Problem Set 2

You are given a text file (flight12.csv) of flight delays and are required to do the following. (40 points total)

A. (8 points) Create a relation called **alldata** based on the flight12.csv. You use the following column names for the data fields in each record: YEAR, FL_DATE, UNIQUE_CARRIER, CARRIER, FL_NUM, ORIGIN_AIRPORT_ID, ORIGIN, ORIGIN_CITY_NAME, ORIGIN_STATE_ABR, DEST_AIRPORT_ID, DEST, DEST_CITY_NAME, DEST_STATE_ABR, DEP_DELAY_NEW, ARR_DELAY, ARR_DELAY_NEW, CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELA, and LATE_AIRCRAFT_DELAY. If a column name contains the word "DELAY," assign the float data type. Leave all other columns the default data type. Since the flight12.csv dataset contains a special data format (e.g., data values that contains a comma and that are enclosed in quotation marks), you will need to use a special loader function to load the data, which is org.apache.pig.piggybank.storage.CSVExcelStorage(). Give a screenshot of the first ten rows of the alldata relation.

B. (2 points) Create a relation, **wanted_data**, based on the **alldata** relation, which only contains the following columns: FL_DATE, FL_NUM, CARRIER_DELAY,

WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, and LATE_AIRCRAFT_DELAY.

C. (7 points) Find out the longest delay of each category of delay based on the wanted_data relation. Attach a screenshot of your results.

D. (8 points) What are the average delays (round to 2 decimal points) of the following delay categories: CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, and LATE_AIRCRAFT_DELAY?

E. (15 points) Use Python to write a UDF named, delay_udf.py, to identify the most common and most uncommon delay categories based on the average delays in the avg_result relation obtained in Step D. If the highest and lowest average delays are x minutes (correspondent to XX_DELAT) and y minutes (correspondent to YY_DELAY) respectively, your Pig script must output the following language: "The most common delay category is XX_DELAY, which has an average delay of x minutes, and the most uncommon delay category is YY_DELAY, which has an average delay of y minutes." Submit a screenshot of your delay_udf.py file in the vi editor and a screenshot of your Pig script using the UDF. (Hint: you should use a tuple of tuples as the argument for the UDF.)