MAPREDUCE LAB 2, DUE: SEPTEMBER 23, 11:59PM EST

1. Instructions

1.1. **What to look at.** The only references you may use for this lab are:

- The mrjob api reference: `https://mrjob.readthedocs.io/en/latest/`
- Any material on Canvas

In particular, do not use stackoverflow or chegg for two reasons, they often give bad advice.

1.2. **How to collaborate.**

- Write down (in your code) the names of people you collaborate with.
- You can look at and discuss code with your collaborators *only for the purposes of debugging.*
- You *cannot* copy/email/transmit code. Your code must be typed by you without looking at other people's code.
- You can discuss algorithmic strategies (pictures are fine, pseudocode is not)

1.3. **Getting the code and submitting the code.** Use update-starter to get the code and submit to gradescope. You can submit multiple times based on autograder responses **Make sure to only commit human-generated files, and no other files into your repository.**

After running update-starter, you should see an mrlab2 directory for this assignment and an examples directory that has example mrjob code.

1.4. **The Assignment.**

Question 0. When using **git add**, never use **git add .** or **git add *** (the penalty will be -1000000000000000 points). Always list the exact files you want to add:
git add supercode.py
git add mytextdata.txt
etc. In general, git should only be used to store manually generated files (ones that you type in yourself) and not computer generated files (no pycache, no code output).

Question 1. In this question, we are going to be working with the **retailtab** dataset, located in HDFS in **/datasets/retailtab**. This is a dataset about customer orders from a store. The retailtab dataset is:

- A text file, where each line represents an item from an order. The fields in a line are separated by tabs (represented as `"\t"` in code).
- The dataset has headers which explain what the different fields are
- The field **InvoiceNo** contains the order id (all lines belonging to the same order have the same InvoiceNo).

Your goal is to write mapreduce code using mrjob to do wordcount on the item descriptions. That is, if your code produces output

```
"Toyhouse"      11
```

then it means that if you counted the appearance of "Toyhouse" in the description field of every line, you will get 11 appearcance.

- The input files are CSV, separated by tab. There is nothing scary in them and you should not use the csv python library. **Autograder will check to make sure that "csv" does not appear anywhere in your code.** Each line is a string (as in normal text files) and has tab characters (write as `\t` in code) separating the different record items (you can split by tabs in python).

- Your test file should go into the file **mrlab2/q1/testdata.csv** and your expected output (generated **manually by typing**) should go into **mrlab2/q1/expectedresult.txt** in github. Remember that your test files should have similar properties to the real data. Make sure that the fields are separated by tabs instead of spaces (see discussion of how to spot tabs using vi in the canvas page on using the cluster).
- Remember that the input files have a header in the beginning.
  - The headers are not useful for the actual computation (they are not actual orders).
  - Mapreduce does not have a concept of a "first" line, so you should not be using line ordering to determine whether a line is a header or not. You need to think about this carefully.
- Your code should go into **mrlab2/q1/retailwords.py**.
- Your run script should go into **mrlab2/q1/run**

Question 2. In this question, we are going to be working with the **retailtab** dataset, located in HDFS in **/datasets/retailtab**. This is a dataset about customer orders from a store. The retailtab dataset is:

- A text file, where each line represents an item from an order. The fields in a line are separated by tabs (represented as `"\t"` in code).
- The dataset has headers which explain what the different fields are
- The field **InvoiceNo** contains the order id (all lines belonging to the same order have the same InvoiceNo).

Your goal is to write mapreduce code using mrjob to find, for each combination of country and month, the total amount spent (which depends on the quantity and price of items purchased in the country during that month) and the maximum price. So, in your output, lines should look something like:

```
["France", "11"]        [12345.67, 4.99]
```

Which means that for orders in France in November, the total amount spent was 12345.67 and the maximum price among all items ordered was 4.99. **Don't worry about rounding.**

- The input files are CSV, separated by tab. There is nothing scary in them and you should not use the csv python library. **Autograder will check to make sure that "csv" does not appear anywhere in your code.** Each line is a string (as in normal text files) and has tab characters (write as `\t` in code) separating the different record items (you can split by tabs in python).
- Your test file should go into the file **mrlab2/q2/testdata.csv** and your expected output (generated **manually by typing**) should go into **mrlab2/q2/expectedresult.txt** in github. Remember that your test files should have similar properties to the real data. Make sure that the fields are separated by tabs instead of spaces (see discussion of how to spot tabs using vi in the canvas page on using the cluster).
- Remember that the input files have a header in the beginning.
  - The headers are not useful for the actual computation (they are not actual orders).
  - Mapreduce does not have a concept of a "first" line, so you should not be using line ordering to determine whether a line is a header or not. You need to think about this carefully.
- Your code should go into **mrlab2/q2/retaildetail.py**.
- Your run script should go into **mrlab2/q2/run**