Alex Dekhtyar **CSC 369: Distributed Computing** **Winter 2019**

Lab 7: Work with Resilient Distributed Datasets in PySpark

Due: Monday, March 4, 2019 (midnight)

Lab Preparation.

This is an individual lab. It is expected that you complete most of the lab during the lab period on March 4, finishing up the preparation of the lab deliverables after the lab period is over.

The lab asks you to complete a single task that is broken into smaller steps.

Assignment.

You will be working with `/data/winequality-red-fixed.csv` file available on HDFS.

This file contains information about different wine characteristics, and their effect on the quality of wine. The dataset is stored in a CSV file, all values are numeric, first row contains column names:

fixed acidity,volatile acidity,citric acid,residual sugar,chlorides,free sulfur dioxide,total sulfur dioxide,density,pH,sulphates,alcohol,quality

The `quality` column is the quality of the wine (values from 3 to 8, higher is better). All other columns represent specific measurable characteristics of wine.

Your goal is to create a sequence of Spark computations that analyze this dataset in the following ways.

Note: as the assignment proceeds, you will be creating a sequence of RDDs. Each RDD you create will need to be placed into a separate variable. At the end you will collect, or materialize in other ways the contents of most of the RDDs you create.

Step 1. Load the dataset into an Data Frame `wine`. Represent it as an RDD named `wineRDD`. For subsequent operations you may need to use either the `wine` or the `wineRDD` object as a starting point. Make sure that all your values are numeric (if needed, convert the data you read into appropriate data type).

Step 2. Collect some simple information about the dataset. Specifically, for each quality value, compute the number of times it shows in the dataset. Create an RDD named `qualityHistogram` to store this information. You can use any RDD transformations and actions to achieve that goal, and create any helper functions you need.

Step 3. For wines of quality 5,6, and 7, compute the averages and the standard deviations for the following features: `alcohol, residual sugar, fixed acidity`. Create and RDD named `wineAverages` to store this information. You can use any RDD transformations and actions to achieve this goal, and you can create any helper functions you need.

Step 4. For wines of each quality find the top 10 sweetest wines (i.e., wines or wines with the highest concentration of residual sugar) and place them into the `sweetest` RDD. Similarly, find 50 least acidic wines (by `fixed acidity`) and place them into the `leastAcidic` RDD. Both RDDs should have the same format as `wineRDD.`

Step 5. Find if there are any intersections in `sweetest` and `leastAcidic RDDs.` Create an RDD called `finalRDD` and place any wines common to both of the RDDs above in it.

Step 6. Output the contents of the following RDDs:

```
qualityHistorgam
wineAverages
sweetest
leastAcidic
finalRDD
```

Delivarables. You can test all of this functionality in pyspark shell. But in order to submit, please create, and debug, a Python program `lab7.py` that performs all operations in Steps 1 through 6. The program shall provide the correct result when submitted to spark using `spark-submit.`

`Submission.` Use handin:

```
$handin dekhtyar lab07 lab7.py
```

Make sure your name is in the header comment.