Assignment 2

Spark interactive query and batch processing

Deadline: 23:59, Tuesday 1st of December.

Submission: Submit to the *Assignments* section in the Scio page, specifically in *Assignment 2*. A submission template is given under a folder with the name *"_template_"*. Please change its name using the following format (in lower case):

<first name>-<last name>

Where first and last names correspond to your own e.g., *elio-ventocilla*. Compress your submission using either zip or a tar ball.

Defense: You will defend the reasoning of your code the day after submission: Friday, 2nd of December.

The following assignment has two (2) parts, each with a different data set.

In PART I you will:

- Demonstrate interactive querying through the REPL.
- Demonstrate the use of RDDs.

In PART II you will:

- Demonstrate the use of standalone applications.
- Demonstrate the use of Data frames.

PART I

Data: Weather 2012.
Source: National Oceanic and Atmospheric Administration (NOAA).
Format: Compressed plain text files.
Folder: /home/big-data/datasets/weather
Attributes:

Attribute	Substring	Missing value	Comments
Date	15-23		Format: YYYYMMDD
Latitude	28-34	+99999	Includes negative or positive sign.
Longitude	34-41	+999999	Includes negative or positive sign.
Elevation	46-51	+9999	Includes negative or positive sign.
Wind speed	65-69	9999	
Wind quality	69-70		1 = good quality
Temperature	87-92	+9999	Includes negative or positive sign.
Temperature quality	92-93		1 = good quality
Pressure	99-104	99999	
Pressure quality	104-105		1 = good quality

More info here: ftp://ftp.ncdc.noaa.gov/pub/data/noaa/ish-format-document.pdf

Description:

The following exercise should be done with interactive queries on the Spark REPL and only RDDs. Queries for your solutions should be documented in the text file called *weather.scala* inside folder *part I* of the submission template. Previous to your answers you may have a section with preprocessing code.

Take into account:

- It must be possible to replicate the results by copying and pasting the code in the given order.
- Records with missing values or quality codes different than one (1) should be filtered.

Tasks:

Given that Sweden falls within the following coordinates:

- Latitude between +55382 and +69047.
- Longitude between +11391 and +24034.

Answer the following:

1. How many measures (records) are there?
2. Which is the highest elevation? Give latitude and longitude.
3. Which is the average wind speed in the country?
4. Which is the average temperature per month? Sort from highest to lowest.

Finally, transform the RDD into a data frame and save it as a parquet with the name *2010-parquet,* in the same folder of your answers.

PART II

Data: 2010 Reddit comments.
Source: Reddit.
Format: Compressed parquets.
Folder: /home/big-data/datasets/reddit
Attributes:

Attribute	Comment
author	Author of the comment.
name	ID of the comment.
created_utc	Unix timestamp
body	The actual comment.
ups	Ups given to a comment.
subreddit	Subreddit category it belongs to.
parent_id	Parent of the comment. It could be the ID (name) of another comment.

Description:

The following exercise should be done as a standalone application and using data frames. Use the app "shell" called *reddit* inside the folder *part II* of the submission template. Take the following into account:

- Comment your code.
- For each question, the results from the execution of your code should be saved in individual files inside folder *reddit/results*. The name of the files and output format are stated below.
- Your code will be executed and the resulting files will be checked.

Tasks:

Task	Output fields	Output name	Format
Which were the total amount of comments done per hour of the day?	- Hour (24 format). - Count (amount of comments). E.g.: 16, 8432	*commentcount*	CSV
Which were average amount of ups per subreddit? Order by highest average to lowest.	- Subreddit. - Ups average. E.g.: AskReddit, 21.6	*upsaverage*	CSV
Which was the comment with the highest ups for each week of the year?	- Week (1 - 52). - Comment ups. - Comment (body). E.g.: 46, 210, Comment body…	*weekheights*	Parquet
Do a word count for all comments (body), filtering out words which are found in file */res/stopwords.txt*. Save only the first 200 most frequently used words.	- Word. - Count. E.g.: blabla, 7564598	*wordcount*	CSV