Exercises with Spark SQL

This is a set of exercises to help you practice with Spark SQL. Like the previous RDD exercises, these use the data from the Wikipedia project pagecounts dataset (https://dumps.wikimedia.org/other/pagecounts-raw/). Please take a look at the previous document for more information on how to use these data.

While doing these exercises, place the dataset files in a local or HDFS directory that is accessible from your Spark Context. Write Spark code, in python, to perform the following tasks:

1. Load the input data into a DataFrame. This data frame must have the following fields:
    - project_name (String)
    - page_title (String)
    - num_requests (Long)
    - content_size (Long)

    Print the DataFrame schema and its 15 first rows.

2. From the DataFrame created in the previous exercise, calculate the following values. Write both pure SQL and DataFrame API code for each of them (remember to check that both alternatives produce the same result).
    - Total number of elements.
    - Complete list of project names (no repetitions).
    - Total content size of project “en” (Wikipedia in English).
    - Top 5 most visited pages of project “en”, and the number of visits for each.

3. Create a Spark application. This application has to be executed with spark-submit. When executed, the application should load the Wikimedia pagecounts data into a DataFrame, like in exercise 1. Then, it should process the data and produce the following output:
    - project_summary: A table with one row per wikimedia project. Each row must have the following fields:
        - project_name (String)
        - num_pages (Long): Total number of pages in the project.
        - content_size (long): Total content size in the project.
        - mean_requests (Double): Mean number of requests per page in the project.
    - most_visited: A table containing the most visited pages. To consider a page among the most visited, its number of requests has to be above the project mean. This table must have the same structure as the input (see exercise 1), but contain only the pages that fit into the criteria described.

4. Use the nice_guys input data and add a column with the international phone prefix using DataFrames and UDF.