CS450 - Final exam

Katya Griffiths-Julien, Jerry Lau

# Problem I: MapReduce implementation using Python and mrjob

1. Total Incomes

The totalIncome function mapper splits the lines into individual items and emits a key-value pair with a null key and the income as the value. Reducer then calculates the total sum of incomes.



1. Mean

The meanIncomes function calculates the average income using mapper which extracts individual incomes from each line, emitting key-value pairs with a null key and the income as the value. The reducer then computes the mean by summing up all incomes and dividing by the total number of incomes.



1. Generalized Mean

For the generalizedMeanIncomes function, mapper extracts individual incomes, emitting key-value pairs with a null key and the income as the value. Reducer then calculates the generalized mean using the given order. This demonstrates the flexibility of MapReduce in handling generalized mean calculations for different orders. We set p = 3 since a p = 2 would give the test dataset NaN since you cannot do a square root of a negative number.



1. Maximum

The maxIncome function mapper extracts individual incomes, emitting key-value pairs with a null key and the income as the value. Reducer then determines the maximum income from the received values and outputs the result. MapReduce efficiently handled finding maximum income in the large dataset.



1. Minimum

The minIncome mapper extracts individual incomes, emitting key-value pairs with a null key and the income as the value. Reducer determines the minimum income from the received values.



1. Standard deviation

The stdDevIncomes function mapper extracts individual incomes, emitting key-value pairs with a null key and the income as the value. Reducer calculates the mean and then computes the variance and standard deviation.



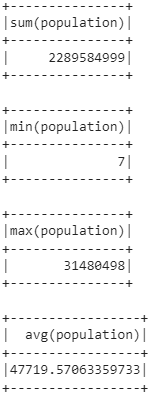
# Problem II: Sparkifying world cities

1. Simple cleaning worldcitiespop

This task required us to remove all entries where the population was null. Using the filter() function, we kept the values that are not null and discarded the rest.

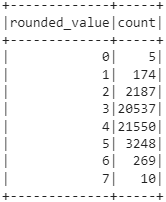
1. Statistics

This is the output of the cities and their population



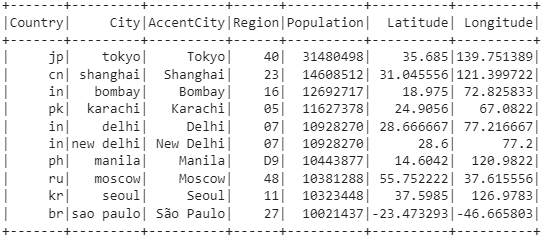
1. Histograms

In this task, we did a log and then floored the values then called a count to group them for the result



1. TopK

These were the top 10 cities in the dataset



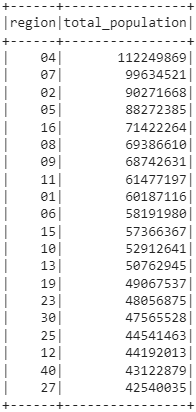
1. Total population of top K cities

These top 10 cities with New Delhi in duplicate accounts for ~5% of the dataset population



1. Total population of each region

Region 04 has the most population. The provided is the top 20 regions with the most population



1. Sum of population in countries where there is are cities with over 10 million population

The dataset returned with a total of 41 countries with the requirement

1. Average population of the for cities above certain threshold

We set the threshold to be at 1 million, the dataset returned with a total of 89 countries

1. Cities that are higher than avg country population

We set the threshold to be at 1 million, the data set returns with a total of 277 cities.

1. Re-cleaning

We removed the duplicate values, mainly Dehli and New Dehli, the new dataset now contains 47736 items compared with 47980. A reduction of 244 items. This is the new top 20 cities by population

