BDA2 - Spark - Exercises

Naveen Gabriel (navga709), Lennart Schilling (lensc874)

2019-05-10

Contents

Assignment 1a

What are the lowest and highest temperatures measured each year for the period 1950-2014. Provide the lists sorted in the descending order with respect to the maximum temperature. In this exercise you will use the temperature-readings.csv file. Extend the program to include the station number (not the station name) where the maximum/minimum temperature was measured.

Code

See attached file *1a_temperature-readings.py*.

Running comand

../runYarn-withHistory.sh 1a_temperature-readings.py

Results

Exactly as in BDA1, it was required to order the data as follows:

year, station with the min, minValue ORDER BY minValue DESC

An extract of ten readings are shwon:

```
Extrat of results for data:temperature-readings.csv:

Maximum temperature per year:

+----+-----------+----------+
|year|temperature|station_nr|
+----+-----------+----------+
|1975|       36.1|     86200|
|1992|       35.4|     63600|
|1994|       34.7|    117160|
|2010|       34.4|     75250|
|2014|       34.4|     96560|
|1989|       33.9|     63050|
|1982|       33.8|     94050|
|1968|       33.7|    137100|
|1966|       33.5|    151640|
|1983|       33.3|     98210|
+----+-----------+----------+
only showing top 10 rows

None
\Minimum temperature per year:

+----+-----------+----------+
|year|temperature|station_nr|
+----+-----------+----------+
|1990|      -35.0|    166870|
|1990|      -35.0|    147270|
|1952|      -35.5|    192830|
|1974|      -35.6|    166870|
|1974|      -35.6|    179950|
|1954|      -36.0|    113410|
|1992|      -36.1|    179960|
|1975|      -37.0|    157860|
|1972|      -37.5|    167860|
|1995|      -37.6|    182910|
+----+-----------+----------+
only showing top 10 rows
```

Assignment 2

Count the number of readings for each month in the period of 1950-2014 which are higher than 10 degrees. Repeat the exercise, this time taking only distinct readings from each station. That is, if a station reported a reading above 10 degrees in some month, then it appears only once in the count for that month. In this exercise you will use the temperature-readings.csv file. The output should contain the following information: Year, month, count

Code

See attached file *2.py*.

Running comand

../runYarn-withHistory.sh 2.py

Results

In contrast to BDA1, it was now required to order the data as follows:

year, month, value ORDER BY value DESC

Two extracts of ten readings are shwon:

```
Extract of results using all readings:

+----+-----+------+
|year|month| count|
+----+-----+------+
|2014|   07|147910|
|2011|   07|147060|
|2010|   07|143860|
|2012|   07|138166|
|2013|   07|134297|
|2009|   07|133570|
|2011|   08|133483|
|2009|   08|129007|
|2013|   08|128920|
|2003|   07|128360|
+----+-----+------+
only showing top 10 rows

None

Extract of results using distinct readings:

+----+-----+-----+
|year|month|count|
+----+-----+-----+
|1972|   10|  378|
|1973|   05|  377|
|1973|   06|  377|
|1972|   08|  376|
|1973|   09|  376|
|1972|   05|  376|
|1972|   09|  375|
|1971|   08|  375|
|1972|   06|  375|
|1971|   09|  374|
+----+-----+-----+
only showing top 10 rows
```

Within this assignment, we also used a SQL-like query as it has been required.

Assignment 3

Find the average monthly temperature for each available station in Sweden. Your result should include average temperature for each station for each month in the period of 1960- 2014. Bear in mind that not every station has the readings for each month in this timeframe. In this exercise you will use the temperature-readings.csv file. The output should contain the following information:
Year, month, station number, average monthly temperature

Code

See attached file *3.py*.

Running comand

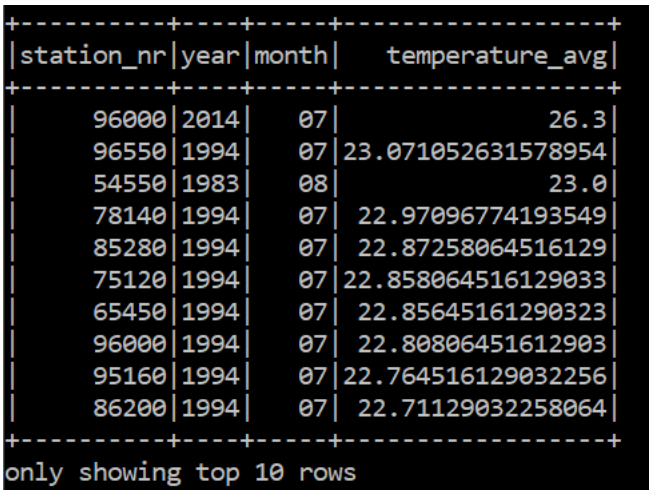
../runYarn-withHistory.sh 3.py

Results

An extract of 25 readings are shwon:

In contrast to BDA1, it was now required to order the data as follows:

year, month, station, avgMonthlyTemperature ORDER BY avgMonthlyTemperature DESC



station_nr	year	month	temperature_avg
96000	2014	07	26.3
96550	1994	07	23.071052631578954
54550	1983	08	23.0
78140	1994	07	22.97096774193549
85280	1994	07	22.87258064516129
75120	1994	07	22.858064516129033
65450	1994	07	22.85645161290323
96000	1994	07	22.80806451612903
95160	1994	07	22.764516129032256
86200	1994	07	22.71129032258064

only showing top 10 rows

Assignment 4

Provide a list of stations with their associated maximum measured temperatures and maximum measured daily precipitation. Show only those stations where the maximum temperature is between 25 and 30 degrees and maximum daily precipitation is between 100 mm and 200 mm. In this exercise you will use the temperature-readings.csv and precipitation-readings.csv files. The output should contain the following information:
Station number, maximum measured temperature, maximum daily precipitation

Code

See attached file *4.py*.

Running comand

../runYarn-withHistory.sh 4.py

Results

In contrast to BDA1, it was now required to order the data as follows:

station, maxTemp, maxDailyPrecipitation ORDER BY station DESC

```
Extract of the result:

+----------+---------------+-----------------+
|station_nr|temperature_max|precipitation_max|
+----------+---------------+-----------------+
+----------+---------------+-----------------+
```

It can be seen that there is no station with a maximum temperature between 25 and 30 degrees and maximum daily precipitation between 100 mm and 200 mm.

Assignment 5

Calculate the average monthly precipitation for the Ostergotland region (list of stations is provided in the separate file) for the period 1993-2016. In order to do this, you will first need to calculate the total monthly precipitation for each station before calculating the monthly average (by averaging over stations). In this exercise you will use the precipitation-readings.csv and stations-Ostergotland.csv files. HINT (not for the SparkSQL lab): Avoid using joins here! stations-Ostergotland.csv is small and if distributed will cause a number of unnecessary shuffles when joined with precipitation RDD. If you distribute precipitation-readings.csv then either repartition your stations RDD to 1 partition or make use of the collect to acquire a python list and broadcast function to broadcast the list to all nodes. The output should contain the following information:
Year, month, average monthly precipitation

Code

See attached file *5.py*.

Running comand

../runYarn-withHistory.sh 5.py

Results

An extract of ten readings are shwon:

In contrast to BDA1, it was now required to order the data as follows:

year, month, avgMonthlyPrecipitation ORDER BY year DESC, month DESC

```
Extract of result shown as follows: year-month, average precipitation

+----+-----+------------------+
|year|month| precipitation_avg|
+----+-----+------------------+
|2016|   07|               0.0|
|2016|   06|           47.6625|
|2016|   05|29.250000000000004|
|2016|   04|26.900000000000006|
|2016|   03|19.962500000000002|
|2016|   02|           21.5625|
|2016|   01|            22.325|
|2015|   12|28.924999999999997|
|2015|   11| 63.88750000000002|
|2015|   10|            2.2625|
+----+-----+------------------+
only showing top 10 rows
```

Assignment 6

Compare the average monthly temperature (find the difference) in the period 1950-2014 for all stations in Ostergotland with long-term monthly averages in the period of 1950-1980. Make a plot of your results. HINT: The first step is to find the monthly averages for each station. Then, you can average over all stations to acquire the average temperature for a specific year and month. This RDD/Data Frame can be used to compute the long-term average by averaging over all the years in the interval. The output should contain the following information:
Year, month, difference

Code

See attached file *6.py*.

Running comand

../runYarn-withHistory.sh 6.py

Results

An extract of ten readings are shwon:

In contrast to BDA1, it was now required to order the data as follows:

year, month, difference ORDER BY year DESC, month DESC

```
+----+-----+--------------------+
|year|month|          difference|
+----+-----+--------------------+
|2014|   12|  0.6110683429360741|
|2014|   11|  1.1205447618203657|
|2014|   10| 0.49473294313226024|
|2014|   09| 0.12224575499532087|
|2014|   08|-0.24716922491881732|
|2014|   07|  2.8730233662686686|
|2014|   06| -1.7731327396984735|
|2014|   05|-0.16487460483164007|
|2014|   04|   1.355649391229755|
|2014|   03|   3.452941056440987|
+----+-----+--------------------+
```

In addition to the printed results, the results has been stored in the HDFS. To access this data, we first copied it to Heffa using the following command:

hdfs dfs -copyToLocal result_assignment6/

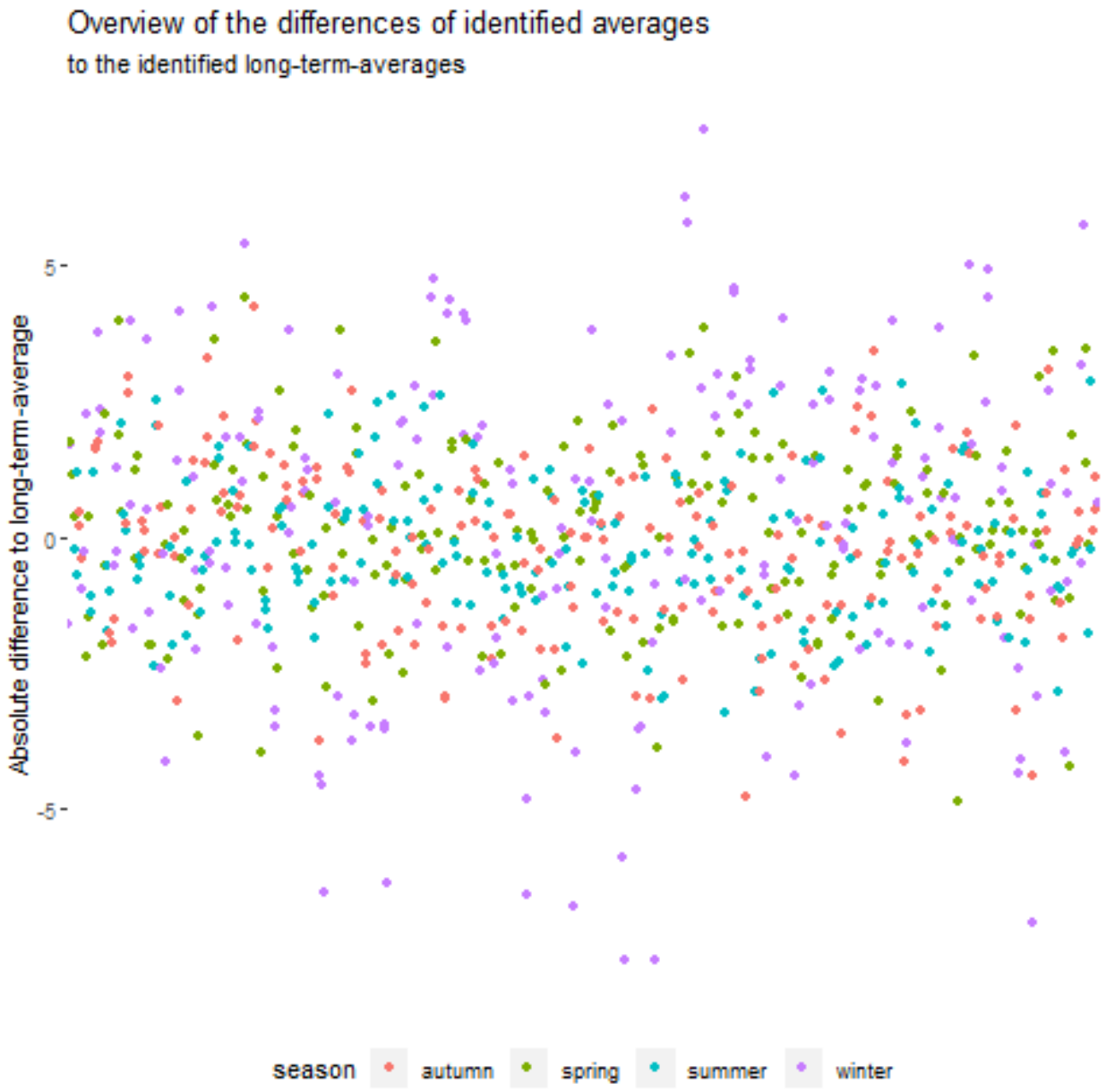
Afterwards, we were able to copy it to our local computer by usage of this command:

scp -r x_lensc@heffa.nsc.liu.se:result_assignment6 result_assignment6

Using this data, we were able to create following plot:

```
# Reading data.
data = read.delim("result_assignment6.txt", header = FALSE, sep = ",", dec = ".")
# Adjusting column names.
colnames(data) = c("year", "month", "difference")
# Preparing data.
# year_month
data$year = as.character(data$year)
```

```r
data$year = substr(data$year, 12, 15)
# month
data$month = as.character(data$month)
data$month = substr(data$month, 10, 11)
# difference
data$difference = substring(data$difference, first = 13)
data$difference = gsub(x = data$difference,
                       pattern = ")",
                       replacement = "")
data$difference = as.numeric(data$difference)
# Adding season variable
data$season = 1
for (i in 1:nrow(data)){
  if (data$month[i] %in% c("12","01","02")) {
    data[i, "season"] = "winter"
  } else if (data$month[i] %in% c("03","04","05")) {
    data[i, "season"] = "spring"
  } else if (data$month[i] %in% c("06","07","08")) {
    data[i, "season"] = "summer"
  } else {
    data[i, "season"] = "autumn"
  }
}
# Sorting data.
data$year_month = paste0(data$year, "-", data$month)
data = data[order(data$year_month),]
# Plotting.
library(ggplot2)
ggplot(data = data) +
  geom_point(aes(x = year_month,
                 y = difference,
                 color = season)) +
  theme(legend.position = "bottom") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  labs(x = "year_month",
       y = "Absolute difference to long-term-average",
       title = "Overview of the differences of identified averages",
       subtitle = "to the identified long-term-averages")
```



For every month over the whole time period (1950-2014), the difference of the average of this specific month in the specific year to the long-term average (calculated by usage of data from 1950-1980) of this month is shown. Based on too many different values, the labels of the year/month-values is not shown. However, the data is sorted related to this variable so that the plot can be read from the left (1950) to the right (2014). By usage of the season as a group variable, it can be seen that especially for winter months, a larger spread is visible.