BDA1 - 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.

File: temperature-readings.csv

Code

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

Running comand

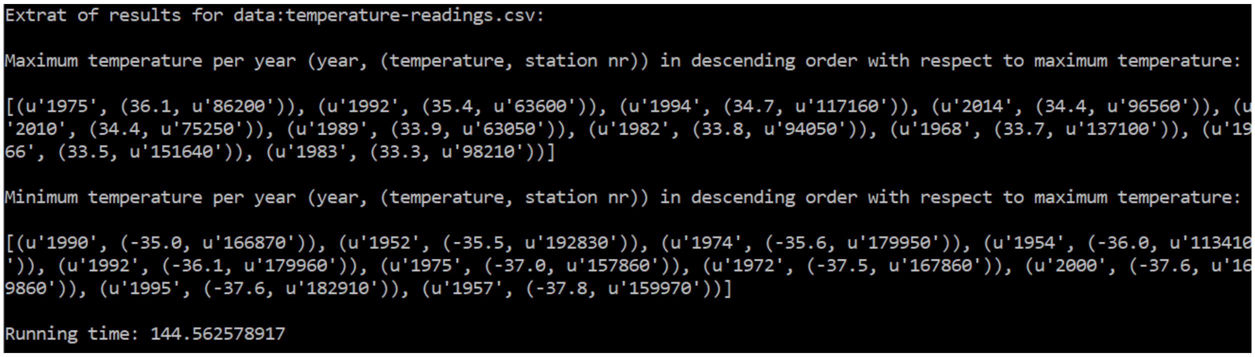
../runYarn-withHistory.sh 1a_temperature-readings.py

Results

As required, the data is ordered as follows:

year, station with the max, maxValue ORDER BY maxValue DESC

An extract of ten readings are shwon:



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

Maximum temperature per year (year, (temperature, station nr)) in descending order with respect to maximum temperature:

[(u'1975', (36.1, u'86200')), (u'1992', (35.4, u'63600')), (u'1994', (34.7, u'117160')), (u'2014', (34.4, u'96560')), (u
'2010', (34.4, u'75250')), (u'1989', (33.9, u'63050')), (u'1982', (33.8, u'94050')), (u'1968', (33.7, u'137100')), (u'19
66', (33.5, u'151640')), (u'1983', (33.3, u'98210'))]

Minimum temperature per year (year, (temperature, station nr)) in descending order with respect to maximum temperature:

[(u'1990', (-35.0, u'166870')), (u'1952', (-35.5, u'192830')), (u'1974', (-35.6, u'179950')), (u'1954', (-36.0, u'113410
')), (u'1992', (-36.1, u'179960')), (u'1975', (-37.0, u'157860')), (u'1972', (-37.5, u'167860')), (u'2000', (-37.6, u'16
9860')), (u'1995', (-37.6, u'182910')), (u'1957', (-37.8, u'159970'))]

Running time: 144.562578917
```

File: temperatures-big.csv

Code

See attached file *1a_temperatures-big.py*.

Running comand

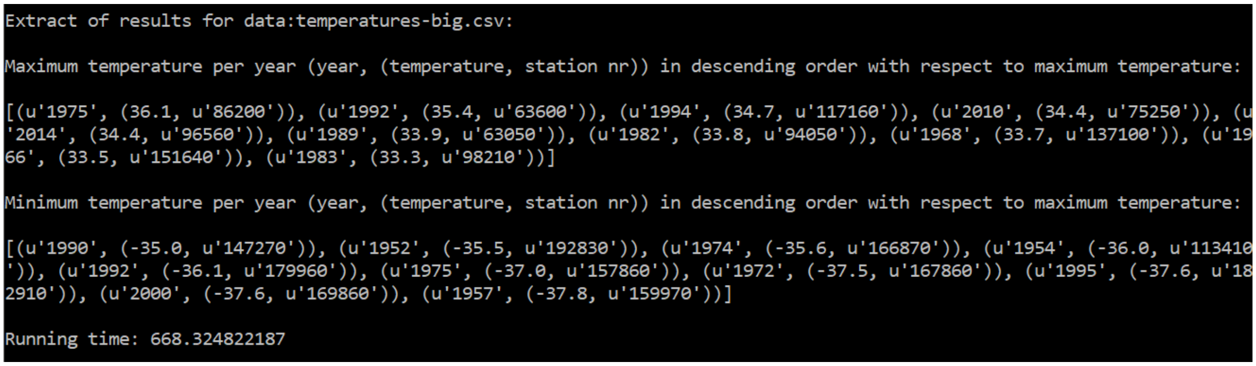
../runYarn-withHistory.sh 1a_temperatures-big.py

Results

As required, the data is ordered as follows:

year, station with the max, maxValue ORDER BY maxValue DESC

An extract of ten readings are shwon:



```
Extract of results for data:temperatures-big.csv:

Maximum temperature per year (year, (temperature, station nr)) in descending order with respect to maximum temperature:

[(u'1975', (36.1, u'86200')), (u'1992', (35.4, u'63600')), (u'1994', (34.7, u'117160')), (u'2010', (34.4, u'75250')), (u
'2014', (34.4, u'96560')), (u'1989', (33.9, u'63050')), (u'1982', (33.8, u'94050')), (u'1968', (33.7, u'137100')), (u'19
66', (33.5, u'151640')), (u'1983', (33.3, u'98210'))]

Minimum temperature per year (year, (temperature, station nr)) in descending order with respect to maximum temperature:

[(u'1990', (-35.0, u'147270')), (u'1952', (-35.5, u'192830')), (u'1974', (-35.6, u'166870')), (u'1954', (-36.0, u'113410
')), (u'1992', (-36.1, u'179960')), (u'1975', (-37.0, u'157860')), (u'1972', (-37.5, u'167860')), (u'1995', (-37.6, u'18
2910')), (u'2000', (-37.6, u'169860')), (u'1957', (-37.8, u'159970'))]

Running time: 668.324822187
```

Assignment 1b

Write the non-parallelized program in Python to find the maximum temperatures for each year without using Spark. In this case you will run the program using: python script.py This program will read the local file (not from HDFS). How does the runtime compare to the Spark version? Use logging (add the –conf spark.eventLog.enabled=true flag) to check the execution of the Spark program. Repeat the exercise, this time using temperatures-big.csv file available on hdfs. Explain the differences and try to reason why such runtimes were observed.

File: temperature-readings.csv

Code

See attached file *1b.py*.

Running comand

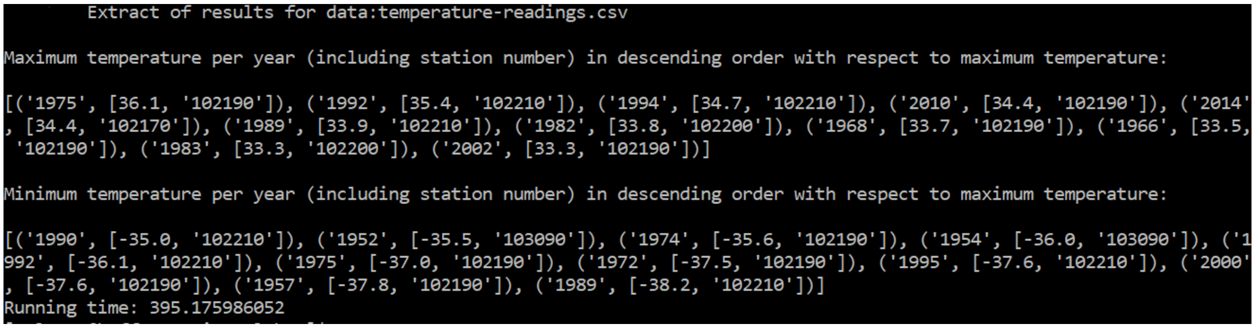
python 1b.py temperature-readings.csv

Results

As required, the data is ordered as follows:

year, station with the max, maxValue ORDER BY maxValue DESC

An extract of ten readings are shwon:



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

Maximum temperature per year (including station number) in descending order with respect to maximum temperature:

[('1975', [36.1, '102190']), ('1992', [35.4, '102210']), ('1994', [34.7, '102210']), ('2010', [34.4, '102190']), ('2014'
, [34.4, '102170']), ('1989', [33.9, '102210']), ('1982', [33.8, '102200']), ('1968', [33.7, '102190']), ('1966', [33.5,
 '102190']), ('1983', [33.3, '102200']), ('2002', [33.3, '102190'])]

Minimum temperature per year (including station number) in descending order with respect to maximum temperature:

[('1990', [-35.0, '102210']), ('1952', [-35.5, '103090']), ('1974', [-35.6, '102190']), ('1954', [-36.0, '103090']), ('1
992', [-36.1, '102210']), ('1975', [-37.0, '102190']), ('1972', [-37.5, '102190']), ('1995', [-37.6, '102210']), ('2000'
, [-37.6, '102190']), ('1957', [-37.8, '102190']), ('1989', [-38.2, '102210'])]
Running time: 395.175986052
```

File: temperatures-big.csv

Code

See attached file *1b.py*.

Running comand

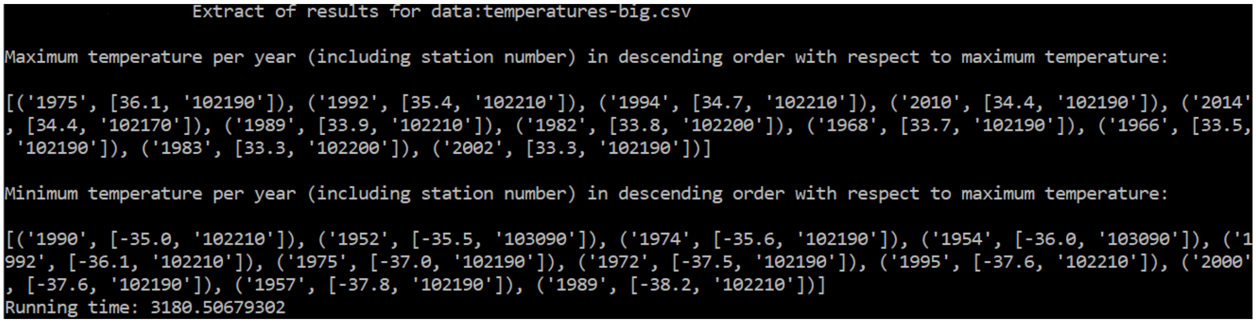
python 1b.py temperatures-big.csv

Results

As required, the data is ordered as follows:

year, station with the max, maxValue ORDER BY maxValue DESC

An extract of ten readings are shwon:



```
                 Extract of results for data:temperatures-big.csv

Maximum temperature per year (including station number) in descending order with respect to maximum temperature:

[('1975', [36.1, '102190']), ('1992', [35.4, '102210']), ('1994', [34.7, '102210']), ('2010', [34.4, '102190']), ('2014'
, [34.4, '102170']), ('1989', [33.9, '102210']), ('1982', [33.8, '102200']), ('1968', [33.7, '102190']), ('1966', [33.5,
 '102190']), ('1983', [33.3, '102200']), ('2002', [33.3, '102190'])]

Minimum temperature per year (including station number) in descending order with respect to maximum temperature:

[('1990', [-35.0, '102210']), ('1952', [-35.5, '103090']), ('1974', [-35.6, '102190']), ('1954', [-36.0, '103090']), ('1
992', [-36.1, '102210']), ('1975', [-37.0, '102190']), ('1972', [-37.5, '102190']), ('1995', [-37.6, '102210']), ('2000'
, [-37.6, '102190']), ('1957', [-37.8, '102190']), ('1989', [-38.2, '102210'])]
Running time: 3180.50679302
```

Comparing the runtime between the spark and the non-parallelized version, it is obvious for both files that the Spark version is much faster. Within every output in assignment 1, the runtime is printed. While for the file *temperature-readings.csv*, the runtime difference is 144 seconds versus 395 seconds, the runtime for file *temperatures-big.csv* differs even more (668 seconds versus 3180 seconds). This makes totally sense since for the non-parallelized, the file is not distributed in HDFS and therefore it is not possible to run procedures parallelly on several nodes which of course leads to longer runtimes.

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*.

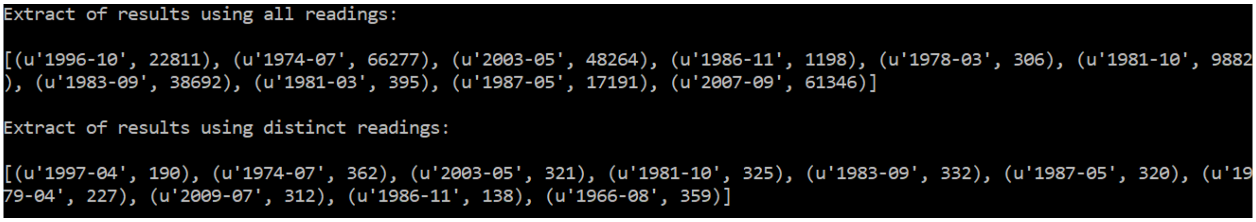
Running comand

../runYarn-withHistory.sh 2.py

Results

In contrast to BDA2, no specific ordering of the data has been required.

Two extracts of ten readings are shwon:



```
Extract of results using all readings:

[(u'1996-10', 22811), (u'1974-07', 66277), (u'2003-05', 48264), (u'1986-11', 1198), (u'1978-03', 306), (u'1981-10', 9882
), (u'1983-09', 38692), (u'1981-03', 395), (u'1987-05', 17191), (u'2007-09', 61346)]

Extract of results using distinct readings:

[(u'1997-04', 190), (u'1974-07', 362), (u'2003-05', 321), (u'1981-10', 325), (u'1983-09', 332), (u'1987-05', 320), (u'19
79-04', 227), (u'2009-07', 312), (u'1986-11', 138), (u'1966-08', 359)]
```

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*.

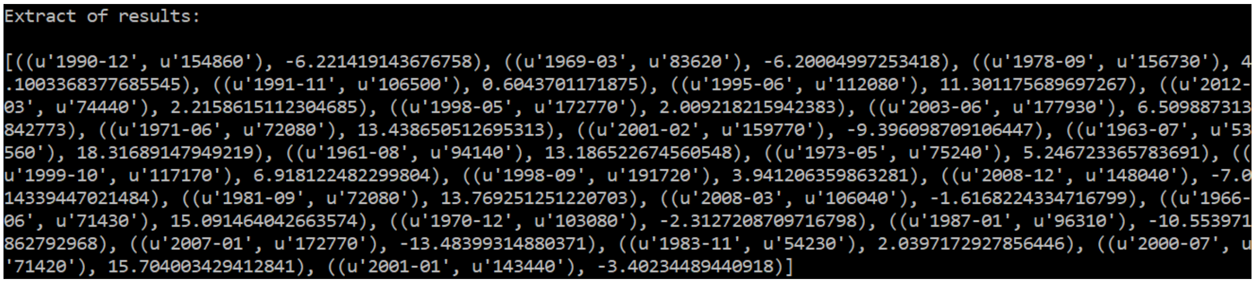
Running comand

../runYarn-withHistory.sh 3.py

Results

In contrast to BDA2, no specific ordering of the data has been required.

An extract of 25 readings are shwon:



```
Extract of results:

[((u'1990-12', u'154860'), -6.221419143676758), ((u'1969-03', u'83620'), -6.20004997253418), ((u'1978-09', u'156730'), 4
.1003368377685545), ((u'1991-11', u'106500'), 0.6043701171875), ((u'1995-06', u'112080'), 11.301175689697267), ((u'2012-
03', u'74440'), 2.2158615112304685), ((u'1998-05', u'172770'), 2.009218215942383), ((u'2003-06', u'177930'), 6.509887313
842773), ((u'1971-06', u'72080'), 13.438650512695313), ((u'2001-02', u'159770'), -9.396098709106447), ((u'1963-07', u'53
560'), 18.31689147949219), ((u'1961-08', u'94140'), 13.186522674560548), ((u'1973-05', u'75240'), 5.246723365783691), ((
u'1999-10', u'117170'), 6.918122482299804), ((u'1998-09', u'191720'), 3.941206359863281), ((u'2008-12', u'148040'), -7.0
14339447021484), ((u'1981-09', u'72080'), 13.769251251220703), ((u'2008-03', u'106040'), -1.6168224334716799), ((u'1966-
06', u'71430'), 15.091464042663574), ((u'1970-12', u'103080'), -2.3127208709716798), ((u'1987-01', u'96310'), -10.553971
862792968), ((u'2007-01', u'172770'), -13.48399314880371), ((u'1983-11', u'54230'), 2.0397172927856446), ((u'2000-07', u
'71420'), 15.704003429412841), ((u'2001-01', u'143440'), -3.40234489440918)]
```

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

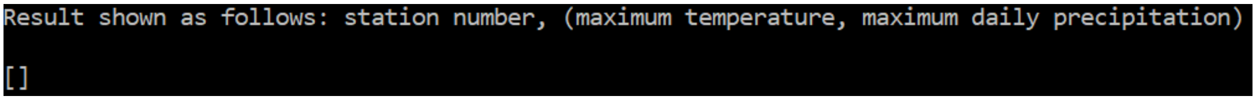
See attached file *4.py*.

Running comand

../runYarn-withHistory.sh 4.py

Results

In contrast to BDA2, no specific ordering of the data has been required.



```
Result shown as follows: station number, (maximum temperature, maximum daily precipitation)

[]
```

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*.

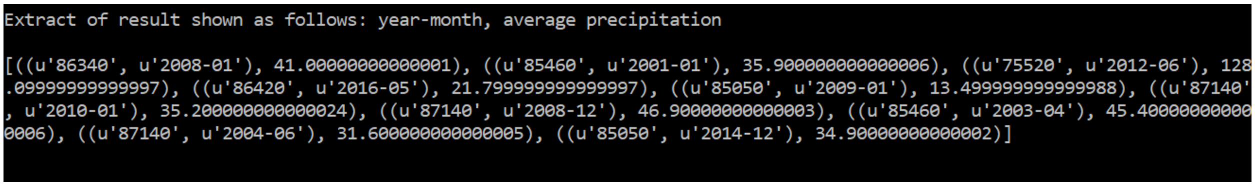
Running comand

../runYarn-withHistory.sh 5.py

Results

In contrast to BDA2, no specific ordering of the data has been required.

An extract of ten readings are shwon:



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

[((u'86340', u'2008-01'), 41.00000000000001), ((u'85460', u'2001-01'), 35.900000000000006), ((u'75520', u'2012-06'), 128
.09999999999997), ((u'86420', u'2016-05'), 21.799999999999997), ((u'85050', u'2009-01'), 13.499999999999988), ((u'87140'
, u'2010-01'), 35.200000000000024), ((u'87140', u'2008-12'), 46.90000000000003), ((u'85460', u'2003-04'), 45.40000000000
0006), ((u'87140', u'2004-06'), 31.600000000000005), ((u'85050', u'2014-12'), 34.90000000000002)]
```

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*.

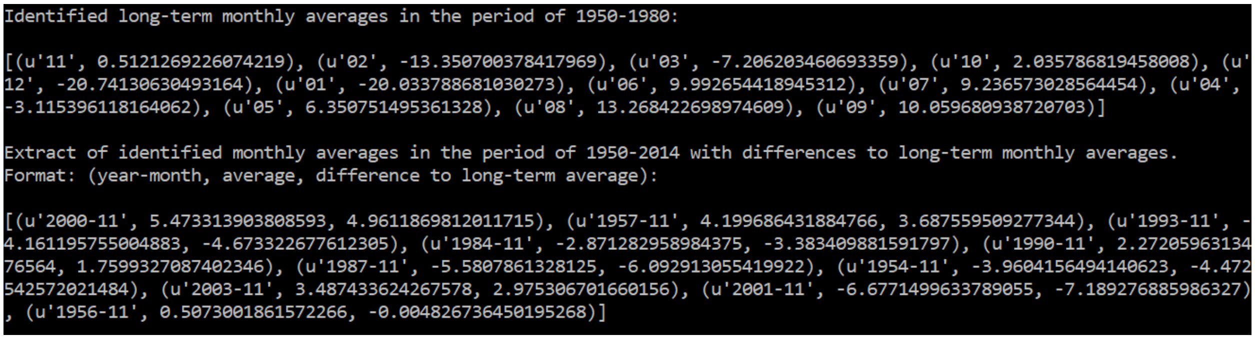
Running comand

../runYarn-withHistory.sh 6.py

Results

In contrast to BDA2, no specific ordering of the data has been required.

An extract of ten readings are shwon:



```
Identified long-term monthly averages in the period of 1950-1980:

[(u'11', 0.5121269226074219), (u'02', -13.350700378417969), (u'03', -7.206203460693359), (u'10', 2.035786819458008), (u'
12', -20.74130630493164), (u'01', -20.033788681030273), (u'06', 9.992654418945312), (u'07', 9.236573028564454), (u'04',
-3.115396118164062), (u'05', 6.350751495361328), (u'08', 13.268422698974609), (u'09', 10.059680938720703)]

Extract of identified monthly averages in the period of 1950-2014 with differences to long-term monthly averages.
Format: (year-month, average, difference to long-term average):

[(u'2000-11', 5.473313903808593, 4.9611869812011715), (u'1957-11', 4.199686431884766, 3.687559509277344), (u'1993-11', -
4.161195755004883, -4.673322677612305), (u'1984-11', -2.871282958984375, -3.383409881591797), (u'1990-11', 2.27205963134
76564, 1.7599327087402346), (u'1987-11', -5.5807861328125, -6.092913055419922), (u'1954-11', -3.9604156494140623, -4.472
542572021484), (u'2003-11', 3.487433624267578, 2.975306701660156), (u'2001-11', -6.6771499633789055, -7.189276885986327)
, (u'1956-11', 0.5073001861572266, -0.004826736450195268)]
```

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:

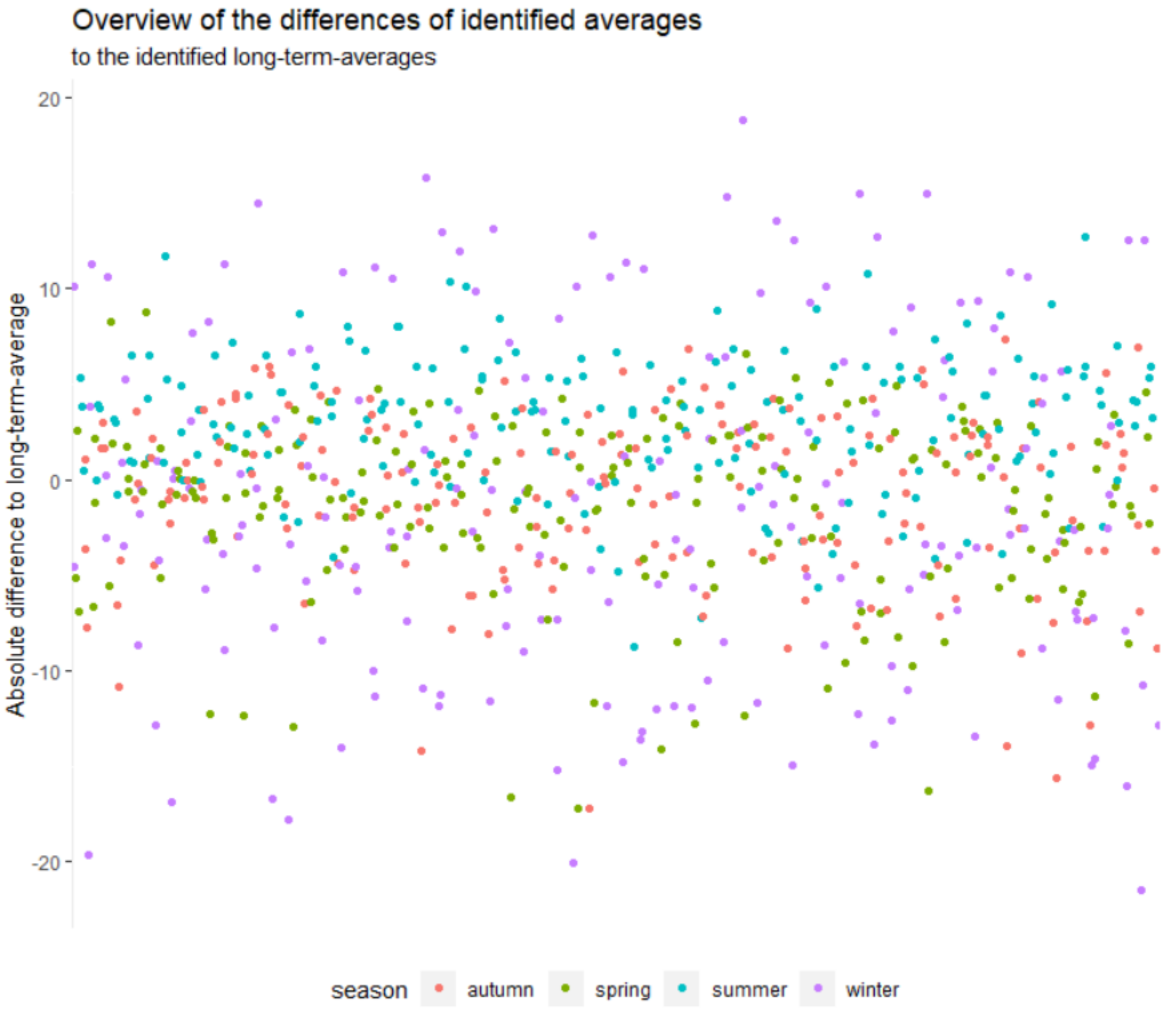
hdfs dfs -copyToLocal result_assignment6/

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:

```r
# Reading data.
data = read.delim("6.txt", header = FALSE, sep = ",", dec = ".")
# Adjusting column names.
colnames(data) = c("year_month", "average_temperature", "difference_to_longTerm")
# Preparing data.
  # year_month
  data$year_month = as.character(data$year_month)
  data$year_month = substr(data$year_month, 4, 10)
  data$year_month = as.factor(data$year_month)
  # difference_to_longTerm
  data$difference_to_longTerm = as.character(data$difference_to_longTerm)
  data$difference_to_longTerm = gsub(x = data$difference_to_longTerm,
                                     pattern = ")",
                                     replacement = "")
  data$difference_to_longTerm = as.numeric(data$difference_to_longTerm)
# Adding season variable
data$season = 1
for (i in 1:nrow(data)){
  if (substr(x = data$year_month[i], start = 6, stop = 7) %in% c("12","01","02")) {
    data[i, "season"] = "winter"
  } else if (substr(x = data$year_month[i], start = 6, stop = 7) %in% c("03","04","05")) {
    data[i, "season"] = "spring"
  } else if (substr(x = data$year_month[i], start = 6, stop = 7) %in% c("06","07","08")) {
    data[i, "season"] = "summer"
  } else {
    data[i, "season"] = "autumn"
  }
}
# Sorting data.
data = data[order(data$year_month),]
# Plotting.
library(ggplot2)
ggplot(data = data) +
  geom_point(aes(x = year_month,
                 y = difference_to_longTerm,
                 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.