Lab 1

Ali Etminan, Ahmed Alhasan

2020-05-22

Q1

```
## Q1
## Lowest and highest temperatures measured each year for the period 1950-2014,
# in descending order with respect to the maximum temperature

from pyspark import SparkContext

sc = SparkContext(appName = "Lab_1")
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
# (station, year-month-day, time, temperature, quality)
lines = temperature_file.map(lambda line: line.split(";"))

# (year,temperature)
year_temperature = lines.map(lambda x: (x[1][0:4], float(x[3])))
# Readings during 1950-2014
year_temperature = year_temperature.filter(lambda x: int(x[0])>=1950 and int(x[0])<=2014)


# take min => take max => join
min_temperatures = year_temperature.reduceByKey(min)
max_temperatures = year_temperature.reduceByKey(max)
# (year, (max_temp, min_temp))
min_max = max_temperatures.join(min_temperatures)

# sort => combine in 1 file
min_max = min_max.sortBy(ascending = False, keyfunc=lambda k: k[1][0])
min_max = min_max.repartition(1)

min_max.saveAsTextFile("BDA/output/Lab_1/A1")
```

(u'2014', (34.4, -42.5))
(u'2010', (34.4, -41.7))
(u'1989', (33.9, -38.2))
(u'1982', (33.8, -42.2))
(u'1968', (33.7, -42.0))
(u'2003', (32.2, -41.5))
(u'1953', (32.2, -38.4))
(u'1955', (32.2, -41.2))
(u'2007', (32.2, -40.7))

(u'1973', (32.2, -39.3))

Q2

```
## Q2_1
# Monthly readings higher than 10 degrees

from pyspark import SparkContext
sc = SparkContext(appName = "Lab_1")

temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
# (station, year-month-day, time, temperature, quality)
lines = temperature_file.map(lambda line: line.split(";"))

# (year-month, (station, temperature))
monthly_temp = lines.map(lambda x: (x[1][0:7], (x[0], float(x[3]))))
# Readings during 1950-2014 & > 10
filtered_mon_temp = monthly_temp.filter(lambda x: int(x[0][0:4]) >= 1950
and int(x[0][0:4]) <=2014 and float(x[1][1]) > 10)

# ((year-month), 1) => ((year-month), count) => combine => sort
counter = filtered_mon_temp.map(lambda x: (x[0], 1))
m_count = counter.reduceByKey(lambda x,y: x+y)
m_count = m_count.repartition(1)
m_count = m_count.sortBy(ascending = False, keyfunc=lambda k: k[1])

m_count.saveAsTextFile("BDA/output/Lab_1/A2_1")
```

(u'2014-07', 147681)
(u'2011-07', 146656)
(u'2010-07', 143419)
(u'2012-07', 137477)
(u'2013-07', 133657)
(u'2009-07', 133008)
(u'2011-08', 132734)
(u'2009-08', 128349)
(u'2013-08', 128235)
(u'2003-07', 128133)

```
## Q2_2
# Distinct readings by station per month higher than 10 degrees

# ((year-month),(station,1)) and only taking one reading per month per station
unique_counter = monthly_temp.map(lambda x: (x[0],(x[1][0],1))).distinct()
# ((year-month), (station, count))
unique_count = unique_counter.reduceByKey(lambda x,y: (x[0], (x[1] + y[1])))

# map to ((year-month), (count)) => combine => sort
unique_count = unique_count.map(lambda x: (x[0], x[1][1]))
unique_count = unique_count.repartition(1)
unique_count = unique_count.sortBy(ascending = False, keyfunc=lambda k: k[1])
```

```
unique_count.saveAsTextFile("BDA/output/Lab_1/A2_2")
```

(u'1972-03', 381)
(u'1972-11', 380)
(u'1972-04', 380)
(u'1972-10', 380)
(u'1972-12', 380)
(u'1972-02', 379)
(u'1973-01', 379)
(u'1972-01', 379)
(u'1973-04', 379)
(u'1971-11', 379)

Q3

```
## Q3
# Average monthly temperature

from pyspark import SparkContext

sc = SparkContext(appName = "Lab_1")
# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
# (station, year-month-day, time, temperature, quality)
lines = temperature_file.map(lambda line: line.split(";"))

# ((year, month, day, station), (temperature))
station_temp = lines.map(lambda x: ((int(x[1][0:4]), int(x[1][5:7]),
int(x[1][8:10]), x[0]), (float(x[3]))))
# Readings during 1960-2014
filtered_temp = station_temp.filter(lambda x: x[0][0] >= 1960 and x[0][0] <=2014)

min_temp = filtered_temp.reduceByKey(min)
max_temp = filtered_temp.reduceByKey(max)
# ((year, month, day, station), (min, max))
min_max  = min_temp.join(max_temp)

# ((year, month, station), 1)
counter = min_max.map(lambda x: ((x[0][0], x[0][1], x[0][3]), 1))
# ((year, month, station), count)
count   = counter.reduceByKey(lambda a,b: (a+b))

# ((year, month, station), (min, max))
daily_min_max = min_max.map(lambda x: ((x[0][0], x[0][1], x[0][3]), (x[1])))
# ((year, month, station), (min_sum, max_sum))
min_max_sum   = daily_min_max.reduceByKey(lambda a,b: ((a[0]+b[0]), (a[1]+b[1])))

# ((year-month, station), ((min_sum, max_sum), count))
joint_RDD = min_max_sum.join(count)

# ((year, month, station), average) where average taken as (min_sum + max_sum / count * 2)
```

```
avg_temp = joint_RDD.map(lambda x: (x[0], ((x[1][0][0]+x[1][0][1])/(x[1][1]*2))))
avg_temp = avg_temp.sortBy(ascending = False, keyfunc=lambda k: k[1]).repartition(1)

avg_temp.saveAsTextFile("BDA/output/Lab_1/A3")
```

((2014, 7, u'96000'), 26.3)
((1994, 7, u'96550'), 23.071052631578944)
((1983, 8, u'54550'), 23.0)
((1994, 7, u'78140'), 22.970967741935482)
((1994, 7, u'85280'), 22.872580645161293)
((1994, 7, u'75120'), 22.85806451612903)
((1994, 7, u'65450'), 22.856451612903225)
((1994, 7, u'96000'), 22.80806451612903)
((1994, 7, u'95160'), 22.764516129032256)
((1994, 7, u'86200'), 22.711290322580645)

Q4

```
## Q4
# Stations with 25-30 degrees maximum temperature and 100-200mm maximum percipitation

from pyspark import SparkContext

sc = SparkContext(appName = "Lab 1")
# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))

# (key, value) = (year,temperature)
station_temp = lines.map(lambda x: (x[0], float(x[3])))
max_temp = station_temp.reduceByKey(max)
filtered_max_temp = max_temp.filter(lambda x: x[1]>=25 and x[1]<=30)

percipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
# (station, year-month-day, time, percipitation, quality)
perc_lines = percipitation_file.map(lambda line: line.split(";"))

# (station, percipitation)
station_perc = perc_lines.map(lambda x: ((x[0], x[1]), (float(x[3]))))
perc_sum = station_perc.reduceByKey(lambda a,b: (a+b))
# Maximum percipitation between 100 and 200mm
filtered_max_perc = perc_sum.filter(lambda x: x[1]>=100 and x[1]<=200).map(lambda x:
(x[0][0], x[1]))

result = filtered_max_temp.join(filtered_max_perc)
result = result.repartition(1)

result.saveAsTextFile("BDA/output/Lab_1/A4")
```

- There is no stations with 25-30 degrees maximum temperature and 100-200mm maximum percipitation

Q5

```
## Q5
# Average monthly precipitation for the Östergotland region for the period 1993-2016

from pyspark import SparkContext
sc = SparkContext(appName = "Lab_1")

stations_file = sc.textFile("BDA/input/stations-Ostergotland.csv")
station_lines = stations_file.map(lambda line: line.split(";"))
# (stations)
stations = station_lines.map(lambda x: x[0]).collect()

percipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
# (station, year-month-day, time, percipitation, quality)
perc_lines = percipitation_file.map(lambda line: line.split(";"))

# ((station, year-month), (percipitation))
prec_rdd = perc_lines.map(lambda x: ((x[0], x[1][0:7]), (float(x[3]))))
# Filtering satations that exist only in the Osterjotland-Stations file
# and in the period 1993-2016
precByStation = prec_rdd.filter(lambda x: x[0][0] in stations
and int(x[0][1][0:4])>=1993 and int(x[0][1][0:4])<=2016)

# ((year-month), (percipitation, 1))
precByStation = precByStation.map(lambda x: ((x[0][1]), (x[1], 1)))
# ((year-month), (perc_sum, count))
precByStation = precByStation.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))
# (year-month, average)
precByStation = precByStation.map(lambda x: (x[0], x[1][0]/x[1][1])).repartition(1)

precByStation.saveAsTextFile("BDA/output/Lab_1/A5")
```

(u'1996-11', 0.10184623166413778)
(u'2008-03', 0.05687331536388128)
(u'2008-10', 0.08107985480943733)
(u'2014-05', 0.08136068735753121)
(u'2001-11', 0.040600153885611674)
(u'2011-05', 0.051183231913455)
(u'2002-06', 0.13926221804511302)
(u'2010-02', 0.07920420420420393)
(u'2013-08', 0.07700249199003215)
(u'2010-09', 0.06052446733785992)