Big Data Analytics
732A54
Lab 4

Karolina Ziomek, Rasmus Holm

December 10, 2016

Exercise 1

Output

Question

Minimums Row(year=u'1990', temp=-35.0) Row(year=u'1952', temp=-35.5) Row(year=u'1974', temp=-35.6) Row(year=u'1954', temp=-36.0) Row(year=u'1992', temp=-36.1) Row(year=u'1975', temp=-37.0) Row(year=u'1972', temp=-37.5) Row(year=u'2000', temp=-37.6) Row(year=u'1995', temp=-37.6) Row(year=u'1957', temp=-37.8) Row(year=u'1983', temp=-38.2) Row(year=u'1989', temp=-38.2) Row(year=u'1953', temp=-38.4) Row(year=u'2009', temp=-38.5) Row(year=u'1993', temp=-39.0)

Maximums Row(year=u'1975', temp=36.1) Row(year=u'1992', temp=35.4) Row(year=u'1994', temp=34.7) Row(year=u'2014', temp=34.4) Row(year=u'2010', temp=34.4) Row(year=u'1989', temp=33.9) Row(year=u'1982', temp=33.8) Row(year=u'1968', temp=33.7) Row(year=u'1966', temp=33.5) Row(year=u'1983', temp=33.3) Row(year=u'2002', temp=33.3) Row(year=u'1986', temp=33.2) Row(year=u'1970', temp=33.2) Row(year=u'1956', temp=33.0) Row(year=u'2000', temp=33.0)

a)

Minimums Row(year=u'1990', station=166870, temp=-35.0) Row(year=u'1952', station=192830, temp=-35.5) Row(year=u'1974', station=166870, temp=-35.6) Row(year=u'1954', station=113410, temp=-36.0) Row(year=u'1992', station=179960, temp=-36.1) Row(year=u'1975', station=157860, temp=-37.0) Row(year=u'1972', station=167860, temp=-37.5) Row(year=u'1995', station=182910, temp=-37.6) Row(year=u'2000', station=169860, temp=-37.6) Row(year=u'1957', station=159970, temp=-37.8) Row(year=u'1983', station=191900, temp=-38.2) Row(year=u'1989', station=166870, temp=-38.2) Row(year=u'1953', station=183760, temp=-38.4) Row(year=u'2009', station=179960, temp=-38.5) Row(year=u'1993', station=191900, temp=-39.0)

Maximums Row(year=u'1975', station=86200, temp=36.1) Row(year=u'1992', station=63600, temp=35.4) Row(year=u'1994', station=117160, temp=34.7) Row(year=u'2010', station=75250, temp=34.4) Row(year=u'2014', station=96560, temp=34.4) Row(year=u'1989', station=63050, temp=33.9) Row(year=u'1982', station=94050, temp=33.8) Row(year=u'1968', station=137100, temp=33.7) Row(year=u'1966', station=151640, temp=33.5) Row(year=u'2002', station=78290, temp=33.3) Row(year=u'1983', station=98210, temp=33.3) Row(year=u'1970', station=103080, temp=33.2) Row(year=u'1986', station=76470, temp=33.2) Row(year=u'2000', station=62400, temp=33.0) Row(year=u'1956', station=145340, temp=33.0)

Code

Initialization

```
def exercise01():
data = sc.textFile("/user/x_rahol/data/temperature-readings.csv")

observations = data.map(lambda line: line.split(";")) \
.filter(lambda obs:
(int(obs[1][:4]) >= 1950 and
    int(obs[1][:4]) <= 2014)) \
.map(lambda obs: Row(station=int(obs[0]),
                                year=obs[1].split("-")[0],
                                temp=float(obs[3])))

schema_temp_readings = sqlContext.createDataFrame(observations)
schema_temp_readings.registerTempTable("temp_readings")

exercise01question()
exercise01a()
```

Question

```
def exercise01question():
year_min_temp = sqlContext.sql(
"""
        SELECT year, MIN(temp) AS temp
        FROM temp_readings
        GROUP BY year
        ORDER BY temp DESC
        """
)

year_max_temp = sqlContext.sql(
"""
        SELECT year, MAX(temp) AS temp
        FROM
        temp_readings
        GROUP BY year
        ORDER BY temp DESC
        """
)

year_min_temp.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (year, temp): temp) \
.saveAsTextFile("sql_result/1_qa")
year_max_temp.rdd.repartition(1) \
```

```
.sortBy(ascending=False, keyfunc=lambda (year, temp): temp) \
.saveAsTextFile("sql_result/1_qb")
```

a)

```
def exercise01a():
year_min_temp = sqlContext.sql(
"""
        SELECT DISTINCT(tr.year) AS year, FIRST(tr.station) AS station,
               FIRST(temp) AS temp
        FROM temp_readings AS tr
        INNER JOIN
        (
        SELECT year, MIN(temp) AS min_temp
        FROM temp_readings
        GROUP BY year
        ) AS tbl
        ON tr.year = tbl.year
        WHERE tr.temp = tbl.min_temp
        GROUP BY tr.year
        ORDER BY temp DESC
        """
)

year_max_temp = sqlContext.sql(
"""
        SELECT DISTINCT(tr.year) AS year, FIRST(tr.station) AS station,
               FIRST(temp) AS temp
        FROM temp_readings AS tr
        INNER JOIN
        (
        SELECT year, MAX(temp) AS max_temp
        FROM temp_readings
        GROUP BY year
        ) AS tbl
        ON tr.year = tbl.year
        WHERE tr.temp = tbl.max_temp
        GROUP BY tr.year
        ORDER BY temp DESC
        """
)

year_min_temp.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (year, station, temp): temp) \
.saveAsTextFile("sql_result/1_aa")
year_max_temp.rdd.repartition(1) \
```

```
.sortBy(ascending=False, keyfunc=lambda (year, station, temp): temp) \
.saveAsTextFile("sql_result/1_ab")
```

Exercise 2

Output

a)

Row(month=u'2014-07', count=147681) Row(month=u'2011-07', count=146656)
Row(month=u'2010-07', count=143419) Row(month=u'2012-07', count=137477)
Row(month=u'2013-07', count=133657) Row(month=u'2009-07', count=133008)
Row(month=u'2011-08', count=132734) Row(month=u'2009-08', count=128349)
Row(month=u'2013-08', count=128235) Row(month=u'2003-07', count=128133)
Row(month=u'2002-07', count=127956) Row(month=u'2006-08', count=127622)
Row(month=u'2008-07', count=126973) Row(month=u'2002-08', count=126073)
Row(month=u'2005-07', count=125294)

b)

Row(month=u'1972-10', count=378) Row(month=u'1973-06', count=377) Row(month=u'1973-05', count=377) Row(month=u'1972-08', count=376) Row(month=u'1973-09', count=376) Row(month=u'1972-06', count=375) Row(month=u'1972-05', count=375) Row(month=u'1971-08', count=375) Row(month=u'1972-09', count=375) Row(month=u'1971-06', count=374) Row(month=u'1972-07', count=374) Row(month=u'1971-09', count=374) Row(month=u'1973-08', count=373) Row(month=u'1971-05', count=373) Row(month=u'1974-08', count=372)

Code

Initialization

```
def exercise02():
data = sc.textFile("/user/x_rahol/data/temperature-readings.csv")

observations = data.map(lambda line: line.split(";")) \
.filter(lambda obs:
(int(obs[1][:4]) >= 1950 and
    int(obs[1][:4]) <= 2014)) \
.map(lambda obs: Row(station=int(obs[0]),
                                month=obs[1][:7],
                                temp=float(obs[3])))

schema_temp_readings = sqlContext.createDataFrame(observations)
schema_temp_readings.registerTempTable("temp_readings")

# exercise02a()
exercise02aAPI(schema_temp_readings)
# exercise02b()
exercise02bAPI(schema_temp_readings)
```

a)

```
def exercise02aAPI(table):
month_count = table.filter(table["temp"] > 10) \
.groupBy("month") \
.agg(F.count("*").alias("count")) \
.orderBy(F.count("*").desc())

month_count.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (month, count): count) \
.saveAsTextFile("sql_result/2_a")
```

b)

```
def exercise02bAPI(table):
month_distinct_count = table.filter(table["temp"] > 10) \
.groupBy("month") \
.agg(F.countDistinct("station").alias("count"))

month_distinct_count.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (month, count): count) \
.saveAsTextFile("sql_result/2_b")
```

Exercise 3

Output

Row(month=u'2014-07', station=96000, avg_{temp}=26.3) Row(month=u'1994-07', station=96550, avg_{temp}=23.07105263157895) Row(month=u'1983-08', station=54550, avg_{temp}=23.0) Row(month=u'1994-07', station=78140, avg_{temp}=22.97096774193548) Row(month=u'1994-07', station=85280, avg_{temp}=22.872580645161296) Row(month=u'1994-07', station=75120, avg_{temp}=22.85806451612903) Row(month=u'1994-07', station=65450, avg_{temp}=22.85645161290323) Row(month=u'1994-07', station=96000, avg_{temp}=22.80806451612904) Row(month=u'1994-07', station=95160, avg_{temp}=22.76451612903225) Row(month=u'1994-07', station=86200, avg_{temp}=22.71129032258064) Row(month=u'2002-08', station=78140, avg_{temp}=22.7) Row(month=u'1994-07', station=76000, avg_{temp}=22.698387096774198) Row(month=u'1997-08', station=78140, avg_{temp}=22.666129032258066) Row(month=u'1994-07', station=105260, avg_{temp}=22.659677419354843) Row(month=u'1975-08', station=54550, avg_{temp}=22.642857142857142)

Code

```
def exercise03():
data = sc.textFile("/user/x_rahol/data/temperature-readings.csv")

observations = data.map(lambda line: line.split(";")) \
.filter(lambda obs:
(int(obs[1][:4]) >= 1960 and
    int(obs[1][:4]) <= 2014)) \
.map(lambda obs: Row(station=int(obs[0]),
                                day=obs[1],
                                month=obs[1][:7],
                                temp=float(obs[3])))

schema_temp_readings = sqlContext.createDataFrame(observations)
schema_temp_readings.registerTempTable("temp_readings")

station_month_avg_temps = sqlContext.sql(
"""
        SELECT mytbl.month, mytbl.station,
               AVG(mytbl.max_temp + mytbl.min_temp) / 2 AS avg_temp
        FROM
        (
        SELECT month, station, MIN(temp) AS min_temp, MAX(temp) AS max_temp
        FROM temp_readings
        GROUP BY day, month, station
        ) AS mytbl
        GROUP BY mytbl.month, mytbl.station
        ORDER BY AVG(mytbl.max_temp + mytbl.min_temp) / 2 DESC
```

```
        """
)

station_month_avg_temps.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (month, station, temp): temp) \
.saveAsTextFile("sql_result/3")
```

Exercise 4

Output

Row(station=97510, $\max_{temp}$=30.0, $\max_{precip}$=103.99999999999999) Row(station=75250, $\max_{temp}$=30.0, $\max_{precip}$=101.8) Row(station=71420, $\max_{temp}$=30.0, $\max_{precip}$=106.3) Row(station=52350, $\max_{temp}$=30.0, $\max_{precip}$=101.6)

Code

```
def exercise04():
temperature_data = sc.textFile("/user/x_rahol/data/temperature-readings.csv")
precipitation_data = sc.textFile("/user/x_rahol/data/precipitation-readings.csv")

temperature_obs = temperature_data.map(lambda line: line.split(";")) \
.map(lambda obs: Row(station=int(obs[0]),
                                temp=float(obs[3])))

precipitation_obs = precipitation_data.map(lambda line: line.split(";")) \
.map(lambda obs: Row(station=int(obs[0]),
                                day=obs[1],
                                precip=float(obs[3])))

schema_temp_readings = sqlContext.createDataFrame(temperature_obs)
schema_temp_readings.registerTempTable("temp_readings")

schema_precip_readings = sqlContext.createDataFrame(precipitation_obs)
schema_precip_readings.registerTempTable("precip_readings")

combined = sqlContext.sql(
"""
        SELECT tr.station, MAX(temp) AS max_temp, MAX(precip) AS max_precip
        FROM
        temp_readings AS tr
        INNER JOIN
        (
        SELECT station, SUM(precip) AS precip
        FROM precip_readings
        GROUP BY day, station
        ) AS pr
        ON tr.station = pr.station
        WHERE temp >= 25 AND temp <= 30
        AND precip >= 100 AND precip <= 200
        GROUP BY tr.station
        ORDER BY tr.station DESC
```

```
        """
)

combined.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (station, temp, precip): station) \
.saveAsTextFile("sql_result/4")
```

Exercise 5

Output

Row(month=u'2016-07', avg_{precip}=0.0) Row(month=u'2016-06', avg_{precip}=47.6625) Row(month=u'2016-05', avg_{precip}=29.250000000000004) Row(month=u'2016-04', avg_{precip}=26.9) Row(month=u'2016-03', avg_{precip}=19.9625) Row(month=u'2016-02', avg_{precip}=21.5625) Row(month=u'2016-01', avg_{precip}=22.325) Row(month=u'2015-12', avg_{precip}=28.925) Row(month=u'2015-11', avg_{precip}=63.887499999999996) Row(month=u'2015-10', avg_{precip}=2.2625) Row(month=u'2015-09', avg_{precip}=101.3) Row(month=u'2015-08', avg_{precip}=26.987499999999997) Row(month=u'2015-07', avg_{precip}=119.10000000000001) Row(month=u'2015-06', avg_{precip}=78.66250000000001) Row(month=u'2015-05', avg_{precip}=93.22500000000002)

Code

```
def exercise05():
station_data = sc.textFile("/user/x_rahol/data/stations-Ostergotland.csv")

stations = station_data.map(lambda line: line.split(";")) \
.map(lambda obs: int(obs[0])) \
.distinct().collect()
stations = sc.broadcast(stations)
stations = {station: True for station in stations.value}

precipitation_data = sc.textFile("/user/x_rahol/data/precipitation-readings.csv")

precipitation_obs = precipitation_data.map(lambda line: line.split(";")) \
.filter(lambda obs: stations.get(int(obs[0]), False)) \
.map(lambda obs: Row(day=obs[1],
                             month=obs[1][:7],
                             station=int(obs[0]),
                             precip=float(obs[3])))

schema_precip_readings = sqlContext.createDataFrame(precipitation_obs)
schema_precip_readings.registerTempTable("precip_readings")

precipitation_avg_month = sqlContext.sql(
"""
        SELECT mytbl2.month, AVG(mytbl2.precip) AS avg_precip
        FROM
        (
        SELECT mytbl1.month, mytbl1.station, SUM(mytbl1.precip) AS precip
        FROM
        (
        SELECT month, station, SUM(precip) AS precip
```

```
        FROM precip_readings
        GROUP BY day, month, station
        ) AS mytbl1
        GROUP BY mytbl1.month, mytbl1.station
        ) AS mytbl2
        GROUP BY mytbl2.month
        ORDER BY mytbl2.month DESC
        """
)

precipitation_avg_month.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (month, precip): month) \
.saveAsTextFile("sql_result/5")
```

Exercise 6

Output

Row(month=u'2014-12', temp=-0.793851783409785) Row(month=u'2014-11', temp=2.063539672692896) Row(month=u'2014-10', temp=1.5219574906179707) Row(month=u'2014-09', temp=0.06105818643722216) Row(month=u'2014-08', temp=-0.6426470719706945) Row(month=u'2014-07', temp=2.1059218387139786) Row(month=u'2014-06', temp=-1.8073686197315162) Row(month=u'2014-05', temp=0.26719065014070154) Row(month=u'2014-04', temp=2.0661931589915445) Row(month=u'2014-03', temp=3.176498950234617) Row(month=u'2014-02', temp=-2.2292398859946143) Row(month=u'2014-01', temp=-0.9325880207201744) Row(month=u'2013-12', temp=1.9232603493728853) Row(month=u'2013-11', temp=0.9342517939050214) Row(month=u'2013-10', temp=0.7523093967763295)

Code

Data

```
def exercise06():
station_data = sc.textFile("/user/x_rahol/data/stations-Ostergotland.csv")

stations = station_data.map(lambda line: line.split(";")) \
.map(lambda obs: int(obs[0])) \
.distinct().collect()
stations = sc.broadcast(stations)
stations = {station: True for station in stations.value}

temperature_data = sc.textFile("/user/x_rahol/data/temperature-readings.csv")

temperature_data_filtered = temperature_data.map(lambda line: line.split(";")) \
.filter(lambda obs:
(stations.get(int(obs[0]), False) and
             int(obs[1][:4]) >= 1950 and
             int(obs[1][:4]) <= 2014)) \
.map(lambda obs: Row(station=int(obs[0]),
                                day=obs[1],
                                month=obs[1][:7],
                                temp=float(obs[3])))

schema_temp_readings = sqlContext.createDataFrame(temperature_data_filtered)
schema_temp_readings.registerTempTable("temp_readings")

month_avg_temp = sqlContext.sql(
"""
        SELECT mytbl.month, AVG(mytbl.max_temp + mytbl.min_temp) / 2 AS avg_temp
        FROM
```

```
        (
        SELECT month, station, MIN(temp) AS min_temp, MAX(temp) AS max_temp
        FROM temp_readings
        GROUP BY day, month, station
        ) AS mytbl
        GROUP BY mytbl.month
        """
)

longterm_avg_temp = month_avg_temp.filter(F.substring(month_avg_temp["month"],
1, 4) <= 1980) \
.groupBy(F.substring(month_avg_temp["month"], 6, 7).alias("month")) \
.agg(F.avg(month_avg_temp["avg_temp"]).alias("longterm_avg_temp"))

result = month_avg_temp.join(longterm_avg_temp,
(F.substring(month_avg_temp["month"], 6, 7) ==
                    longterm_avg_temp["month"]), "inner") \
.select(month_avg_temp["month"],
(F.abs(month_avg_temp["avg_temp"]) -
              F.abs(longterm_avg_temp["longterm_avg_temp"])).alias("temp")) \
.orderBy(month_avg_temp["month"].desc())

result.rdd.repartition(1) \
.sortBy(ascending=False, keyfunc=lambda (month, temp): month) \
.saveAsTextFile("sql_result/6")
```