BDA2

Rabnawaz jansher & Saman Zahid

5/21/2018

BDA2 - Spark Sql - Exercises

Question 1

Part A

```
#sql spark imports
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
#create a spark objec
sc = SparkContext(appName = "Lab2 Q1")
#create a sql context
sqlContext = SQLContext(sc)
#read a temperatue data from file
temperatureFile = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
lines = temperatureFile.map(lambda line: line.split(";"))
#create temperature dataframe
tempReadingsRows = lines.map(lambda x:
(x[0],x[1],int(x[1][0:4]),float(x[3])))
dataFrameString = ["station","date","year","temp"]
df = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
df.registerTempTable("tempReadingTable")
#group by year and find max temperature
df_filter = df.groupBy('year').agg(F.max('temp').alias('max_temp'))
output = df.join(df_filter,df_filter.year == df.year
,'inner').select('station' , df_filter.year,df_filter.max_temp , df.temp)
#filter data by year
output = output.where('temp =
max_temp').select('station','year','max_temp').where( 'year >= 1950 and year
<= 2014' )
#order dataframe in descending order
output = output.orderBy('max_temp', ascending=False)
#convert dataframe into Rdd
output = output.rdd
#repartion result and save into file
output = output.coalesce(1)
output.saveAsTextFile("lab2_1a")
```

```
Row(station=u'86200', year=1975, max_temp=36.1)
Row(station=u'63600', year=1992, max_temp=35.4)
Row(station=u'117160', year=1994, max_temp=34.7)
Row(station=u'96560', year=2014, max_temp=34.4)
Row(station=u'75250', year=2010, max_temp=34.4)
Row(station=u'63050', year=1989, max_temp=33.9)
Row(station=u'94050', year=1982, max_temp=33.8)
Row(station=u'137100', year=1968, max_temp=33.7)
Row(station=u'151640', year=1966, max_temp=33.5)
Row(station=u'78290', year=2002, max_temp=33.3)
Row(station=u'78290', year=2002, max_temp=33.3)
Row(station=u'98210', year=1983, max_temp=33.3)
Row(station=u'76470', year=1986, max_temp=33.2)
Row(station=u'103080', year=1970, max_temp=33.2)
Row(station=u'62400', year=2000, max_temp=33.0)
Row(station=u'145340', year=1956, max_temp=33.0)
Row(station=u'65160', year=1959, max_temp=32.8)
Row(station=u'137040', year=1991, max_temp=32.7)
Row(station=u'75240', year=2006, max_temp=32.7)
Row(station=u'102540', year=1988, max_temp=32.6)
Row(station=u'172770', year=2011, max_temp=32.5)
Row(station=u'98210', year=1999, max_temp=32.4)
Row(station=u'86420', year=2007, max_temp=32.2)
Row(station=u'97260', year=1955, max_temp=32.2)
Row(station=u'136420', year=2003, max_temp=32.2)
Row(station=u'71470', year=1973, max_temp=32.2)
Row(station=u'95130', year=2008, max_temp=32.2)
Row(station=u'82090', year=2008, max_temp=32.2)
Row(station=u'82090', year=2008, max_temp=32.2)
Row(station=u'102390', year=2008, max_temp=32.2)
Row(station=u'65160', year=1953, max_temp=32.2)
Row(station=u'107140', year=2005, max_temp=32.1)
Row(station=u'63600', year=1979, max_temp=32.0)
Row(station=u'71470', year=1969, max_temp=32.0)
Row(station=u'97260', year=1969, max_temp=32.0)
Row(station=u'62400', year=2001, max_temp=31.9)
Row(station=u'74180', year=1997, max_temp=31.8)
Row(station=u'76420', year=1997, max_temp=31.8)
Row(station=u'94180', year=1977, max_temp=31.8)
```

Part B

```
#import spark libraries
from pyspark import SparkContext
from operator import add
#spark sql context
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
from pyspark.sql.functions import broadcast

#spark context object
sc = SparkContext(appName = "Lab2 Q5")
#create a sql context
```

```
sqlContext = SQLContext(sc)
#temperature dataframe
#read temperature data
temperature_file = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
temp_lines = temperature_file.map(lambda line: line.split(";"))
#temperature dataframe
tempRows = temp_lines.map(lambda x: (x[0],
x[1][0:10],int(x[1][0:4]),int(x[1][5:7]),int(x[1][8:10]), float(x[3]) ))
tempDataString = ["station", "date", "year","month","day","temp"]
#register temperature table
dfTemp = sqlContext.createDataFrame(tempRows,tempDataString)
dfTemp.registerTempTable("tempReadingTable")
#filter stations by using broadcast Join
dfTemp_filter = dfTemp.where('year >= 1960 and year <= 2014')
maxTemp =
dfTemp_filter.groupBy('date','station').agg(F.max('temp').alias('max_temp'),F
.min('temp').alias('min_temp'))
output = maxTemp.rdd
#for now
#output = output.coalesce(1)
output.saveAsTextFile("ppp")
```

```
Row(station=u'147270', year=1990, min_temp=-35.0)
Row(station=u'166870', year=1990, min_temp=-35.0)
Row(station=u'192830', year=1952, min_temp=-35.5)
Row(station=u'166870', year=1974, min_temp=-35.6)
Row(station=u'179950', year=1974, min_temp=-35.6)
Row(station=u'113410', year=1954, min_temp=-36.0)
Row(station=u'179960', year=1992, min_temp=-36.1)
Row(station=u'157860', year=1975, min_temp=-37.0)
Row(station=u'167860', year=1972, min_temp=-37.5)
Row(station=u'169860', year=2000, min_temp=-37.6)
Row(station=u'182910', year=1995, min_temp=-37.6)
Row(station=u'159970', year=1957, min_temp=-37.8)
Row(station=u'166870', year=1989, min_temp=-38.2)
Row(station=u'191900', year=1983, min_temp=-38.2)
Row(station=u'183760', year=1953, min_temp=-38.4)
Row(station=u'179960', year=2009, min_temp=-38.5)
Row(station=u'191900', year=1993, min_temp=-39.0)
Row(station=u'191900', year=1984, min_temp=-39.2)
Row(station=u'123480', year=1984, min_temp=-39.2)
Row(station=u'166870', year=1973, min_temp=-39.3)
Row(station=u'179960', year=1991, min_temp=-39.3)
Row(station=u'179960', year=2008, min_temp=-39.3)
Row(station=u'179960', year=2008, min_temp=-39.3)
Row(station=u'155790', year=2005, min_temp=-39.4)
Row(station=u'181900', year=1961, min_temp=-39.5)
Row(station=u'166810', year=1964, min_temp=-39.5)
Row(station=u'179950', year=1970, min_temp=-39.6)
Row(station=u'166940', year=2004, min_temp=-39.7)
Row(station=u'170790', year=1988, min_temp=-39.9)
Row(station=u'155910', year=1960, min_temp=-40.0)
Row(station=u'160790', year=1960, min_temp=-40.0)
Row(station=u'167710', year=1960, min_temp=-40.0)
Row(station=u'179960', year=1997, min_temp=-40.2)
Row(station=u'179960', year=1994, min_temp=-40.5)
Row(station=u'169860', year=2006, min_temp=-40.6)
Row(station=u'169860', year=2007, min_temp=-40.7)
Row(station=u'169860', year=2007, min_temp=-40.7)
Row(station=u'179960', year=2013, min_temp=-40.7)
Row(station=u'181900', year=1963, min_temp=-41.0)
```

Quesrion 2

Part A

```
#import spark libraries
from pyspark import SparkContext
from operator import add
#spark sql context
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
#spark context object
```

```
sc = SparkContext(appName = "Lab1 Q2-count-records")
#create a sql context
sqlContext = SQLContext(sc)
#read temperature data from file
temperature_file = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))
#create a temperature datafreame
tempReadingsRows = lines.map(lambda p: (p[0], p[1], int(p[1].split("-")[0]),
int(p[1].split("-")[1]), float(p[3]),1))
dataFrameString = ["station", "date", "year", "month", "temp","counter"]
df = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
df.registerTempTable("tempReadingTable")
#select and filter data on year
df_select = df.select('year','month','counter','year','temp').where( 'year >=
1950 and year <= 2014 and temp > 10' )
#count records
output =
df_select.groupBy('year','month').agg(F.count('counter').alias('count')).orde
rBy('count', ascending=False)
#convert dataframe into rdd
output = output.rdd
#repartion data and save
output = output.coalesce(1)
output.saveAsTextFile("lab2_2a")
```

```
Row(year=2014, month=7, count=147681)
Row(year=2011, month=7, count=146656)
Row(year=2010, month=7, count=143419)
Row(year=2012, month=7, count=137477)
Row(year=2013, month=7, count=133657)
Row(year=2009, month=7, count=133008)
Row(year=2011, month=8, count=132734)
Row(year=2009, month=8, count=128349)
Row(year=2013, month=8, count=128235)
Row(year=2003, month=7, count=128133)
Row(year=2002, month=7, count=127956)
Row(year=2006, month=8, count=127622)
Row(year=2008, month=7, count=126973)
Row(year=2002, month=8, count=126073)
Row(year=2005, month=7, count=125294)
Row(year=2011, month=6, count=125193)
Row(year=2012, month=8, count=125037)
Row(year=2006, month=7, count=124794)
Row(year=2010, month=8, count=124417)
Row(year=2014, month=8, count=124045)
Row(year=1997, month=7, count=123496)
Row(year=2007, month=7, count=123218)
Row(year=2013, month=6, count=122181)
Row(year=1997, month=8, count=121154)
Row(year=2001, month=7, count=120529)
Row(year=1998, month=7, count=120230)
Row(year=2000, month=7, count=119769)
Row(year=2004, month=7, count=119536)
Row(year=1999, month=7, count=116385)
Row(year=2008, month=8, count=114272)
Row(year=2004, month=8, count=114168)
Row(year=2002, month=6, count=114034)
Row(year=2005, month=8, count=113950)
Row(year=2001, month=8, count=113937)
Row(year=2007, month=8, count=110428)
Row(year=2000, month=8, count=109201)
Row(year=2003, month=8, count=108501)
Row(year=1996, month=8, count=107758)
Row(year=1997, month=6, count=104696)
```

Part B

```
#import spark libraries
from pyspark import SparkContext
from operator import add
#spark sql context
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
#spark context object
sc = SparkContext(appName = "Lab1 Q2b-count-distinct-records")
#create a sql context
sqlContext = SQLContext(sc)
```

```
#read temperature from file
temperature_file = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))
#create a data frame
tempReadingsRows = lines.map(lambda p: (p[0], p[1], int(p[1].split("-")[0]),
int(p[1].split("-")[1]), float(p[3])))
dataFrameString = ["station", "date", "year", "month", "temp"]
df = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
df.registerTempTable("tempReadingTable")
#filter data year wise
output = df.where(df.year >= 1950)\
                .where(df.year <= 2014)\
                .where(df.temp >= 10)
#groupby data count distinct count
output = output.groupBy(output.year,
output.month).agg(F.countDistinct(output.station).alias('count_record'))\
         .orderBy('count_record', ascending = False)
#convert dataframe into rdd
output = output.rdd
#repartion data and save
output = output.coalesce(1)
output.saveAsTextFile("lab2_2b")
```

```
Row(year=1972, month=10, count_record=378)
Row(year=1973, month=6, count_record=377)
Row(year=1973, month=5, count_record=377)
Row(year=1972, month=8, count_record=376)
Row(year=1973, month=9, count_record=376)
Row(year=1972, month=5, count_record=376)
Row(year=1972, month=6, count_record=375)
Row(year=1972, month=9, count_record=375)
Row(year=1971, month=8, count_record=375)
Row(year=1972, month=7, count_record=374)
Row(year=1971, month=6, count_record=374)
Row(year=1971, month=9, count_record=374)
Row(year=1973, month=8, count_record=373)
Row(year=1971, month=5, count_record=373)
Row(year=1974, month=6, count_record=372)
Row(year=1974, month=8, count_record=372)
Row(year=1973, month=7, count_record=370)
Row(year=1974, month=9, count_record=370)
Row(year=1970, month=8, count_record=370)
Row(year=1974, month=5, count_record=370)
Row(year=1971, month=7, count_record=370)
Row(year=1975, month=9, count_record=369)
Row(year=1976, month=5, count_record=369)
Row(year=1970, month=6, count_record=369)
Row(year=1970, month=9, count_record=369)
Row(year=1976, month=6, count_record=368)
Row(year=1975, month=6, count_record=368)
Row(year=1975, month=8, count_record=367)
Row(year=1975, month=5, count_record=367)
Row(year=1970, month=5, count_record=366)
Row(year=1976, month=9, count_record=365)
Row(year=1977, month=6, count_record=364)
Row(year=1967, month=5, count_record=363)
Row(year=1976, month=8, count_record=363)
Row(year=1974, month=7, count_record=362)
Row(year=1970, month=7, count_record=362)
Row(year=1967, month=9, count_record=361)
Row(year=1966, month=9, count_record=361)
Row(year=1966, month=6, count_record=360)
```

Quesrion 3

```
#sql spark imports
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
sc = SparkContext(appName = "Lab2 Q3")
# create a sql context
sqlContext = SQLContext(sc)
# reading data
temperatureFile = sc.textFile("/user/x_samza/data/temperature-readings.csv")
```

```
lines = temperatureFile.map(lambda line: line.split(";"))
# creating dataframe of temperatures data
tempReadingsRows = lines.map(lambda x: (x[0], int(x[1][8:10]) ,int(x[1][0:4])
,int(x[1][5:7]) , float(x[3]) ))
dataFrameString = ["station","date","year","month","temp"]
df = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
df.registerTempTable("tempReadingTable")
# filtering data within the period 1950 and 2014
df = df.where('year >= 1950 and year <= 2014')
# finding max temperature
df_groupby_df =
df.groupBy('year','month','date','station').agg(F.max('temp').alias('max_temp
erature'))
# finding min temperature
df_groupby =
df.groupBy('year','month','date','station').agg(F.min('temp').alias('min_temp
erature'))
# joining dataframes containing min and max temperature and selecting
required attributes
df_join = df_groupby_df.join(df_groupby,(df_groupby_df.year ==
df_groupby.year) & (df_groupby_df.month == df_groupby.month) &
                        (df_groupby_df.date == df_groupby.date) &
(df_groupby_df.station == df_groupby.station))\

.select(df_groupby.year,df_groupby.month,df_groupby.date,df_groupby.station,

df_groupby.min_temperature,df_groupby_df.max_temperature)
# determining daily avg i-e (daily_max + daily_min)/2
daily_average = df_join.withColumn('sum_min_max', (df_join.min_temperature +
df_join.max_temperature)/2)
# determining monthly avg by avg daily avg over stations
monthly_avg =
daily_average.groupBy('year','month','station').agg(F.avg(daily_average.sum_m
in_max).alias('average'))
# sorting in descending order
monthly_avg = monthly_avg.orderBy('average',ascending = False)
output = monthly_avg.rdd
output = output.coalesce(1)
# saving output
output.saveAsTextFile("lab_q2")
```

```
Row(year=2014, month=7, station=u'96000', average=26.3)
Row(year=1994, month=7, station=u'96550', average=
23.071052631578947)
Row(year=1983, month=8, station=u'54550', average=23.0)
Row(year=1994, month=7, station=u'78140', average=
22.970967741935485)
Row(year=1994, month=7, station=u'85280', average=
22.872580645161293)
Row(year=1994, month=7, station=u'75120', average=
22.858064516129037)
Row(year=1994, month=7, station=u'65450', average=
22.85645161290323)
Row(year=1994, month=7, station=u'96000', average=
22.80806451612904)
Row(year=1994, month=7, station=u'95160', average=
22.764516129032256)
Row(year=1994, month=7, station=u'86200', average=
22.711290322580645)
Row(year=2002, month=8, station=u'78140', average=
22.700000000000003)
Row(year=1994, month=7, station=u'76000', average=
22.698387096774198)
Row(year=1997, month=8, station=u'78140', average=
22.666129032258063)
Row(year=1994, month=7, station=u'105260', average=
22.659677419354843)
Row(year=1975, month=8, station=u'54550', average=
22.642857142857142)
Row(year=2006, month=7, station=u'76530', average=
22.598387096774193)
Row(year=1994, month=7, station=u'86330', average=
22.548387096774192)
Row(year=2006, month=7, station=u'75120', average=
22.527419354838706)
Row(year=1994, month=7, station=u'54300', average=
22.46935483870968)
Row(year=2006, month=7, station=u'78140', average=
22.458064516129035)
Row(year=2001, month=7, station=u'96550', average=
```

Quesrion 4

```python
#import spark libraries
from pyspark import SparkContext
from operator import add
#spark sql context
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F

#spark context object
sc = SparkContext(appName = "Lab1 Q4")
#create a sql context
```

```
sqlContext = SQLContext(sc)
#read preciptate data
precipitation_file = sc.textFile("/user/x_rabsh/data/precipitation-
readings.csv")
precipt_lines = precipitation_file.map(lambda line: line.split(";"))
#preciptiate dataframe
preciptRows = precipt_lines.map(lambda x: (x[0], x[1][0:10], float(x[3]) ))
preciptDataString = ["station", "date", "value"]
#register temperature table
dfPrecipt = sqlContext.createDataFrame(preciptRows,preciptDataString)
dfPrecipt.registerTempTable("tempReadingTable")
#sum preciptate
precipt_filter = dfPrecipt.groupBy('station','date') \
                          .agg(F.sum('value').alias('pvalue'))
output_2 = precipt_filter.groupBy(precipt_filter.station)\

.agg(F.max(precipt_filter.pvalue).alias('max_precipt'))
max_precipt_r = output_2.where('max_precipt >= 100 and max_precipt <= 200')
#temperature file reading
temperature_file = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
temp_lines = temperature_file.map(lambda line: line.split(";"))
#temperature dataframe
tempReadingsRows = temp_lines.map(lambda p: (p[0], p[1], int(p[1].split("-
")[0]), int(p[1].split("-")[1]), float(p[3])))
dataFrameString = ["station", "date", "year", "month", "temp"]
#register temperature table
dfTemperature = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
dfTemperature.registerTempTable("tempReadingTable")
#max temperature
df_filter =
dfTemperature.groupBy(dfTemperature.station).agg(F.max('temp').alias('max_tem
p'))
output = dfTemperature.join(df_filter,df_filter.station ==
dfTemperature.station ,'inner')\
                      .select(dfTemperature.station , df_filter.max_temp ,
dfTemperature.temp)
output = output.where('temp = max_temp')
output = output.where('max_temp >= 20 and max_temp <= 30')
combine_result = max_precipt_r.join(output, output.station ==
max_precipt_r.station)\

.select(output.station,output.max_temp,max_precipt_r.max_precipt)
output = combine_result.orderBy('station', ascending=False)
output = output.rdd
#for now
output = output.coalesce(1)
output.saveAsTextFile("lab2_q4")
```

Because there is no matching between 2 Rdd, so result will be null.

Quesrion 5

```
#import spark libraries
from pyspark import SparkContext
from operator import add
#spark sql context
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
from pyspark.sql.functions import broadcast

#spark context object
sc = SparkContext(appName = "Lab2 Q5")
#create a sql context
sqlContext = SQLContext(sc)
#reading data
ostergotland_file = sc.textFile("/user/x_rabsh/data/stations-
Ostergotland.csv")
#partition data
stations = ostergotland_file.map(lambda line: line.split(";"))
#station data frame
stationRow = stations.map(lambda x: (x[0],x[1]) )
stationDataFrameString = ["station","name"]
stations = sqlContext.createDataFrame(stationRow,stationDataFrameString)
stations.registerTempTable("tempReadingTable")
#boradcast stations
#stations = stations.distinct().collect()
#Os_stations = sc.broadcast(stations)
#preciptiate dataframe
#read preciptate data
precipitation_file = sc.textFile("/user/x_rabsh/data/precipitation-
readings.csv")
precipt_lines = precipitation_file.map(lambda line: line.split(";"))
#preciptiate dataframe
preciptRows = precipt_lines.map(lambda x: (x[0],
x[1][0:10],int(x[1][0:4]),int(x[1][5:7]),int(x[1][8:10]), float(x[3]) ))
preciptDataString = ["station", "date", "year","month","day","value"]
#register temperature table
dfPrecipt = sqlContext.createDataFrame(preciptRows,preciptDataString)
dfPrecipt.registerTempTable("tempReadingTable")
#filter stations by using broadcast Join
filter_stations = stations.join(dfPrecipt , dfPrecipt.station ==
stations.station)\

.select(stations.station,dfPrecipt.value,dfPrecipt.date,dfPrecipt.year,dfPrec
ipt.month,dfPrecipt.day)
#groupby year,month and day and add precpitate values
filter_stations = filter_stations.groupBy('year','month','day') \
                          .agg(F.sum('value').alias('pvalue'))
filter_stations =
filter_stations.groupBy(filter_stations.year,filter_stations.month).agg(F.avg
```

```
('pvalue').alias('avgvalue'))
#sort data by year and month in Descending order
filter_stations = filter_stations.orderBy(['year', 'month'], ascending=False)
#convert datafream into Rdd
output = filter_stations.rdd
#parartion data and save
output = output.coalesce(1)
output.saveAsTextFile("lab2_q5")
```

```
Row(year=2016, month=7, avgvalue=0.0)
Row(year=2016, month=6, avgvalue=12.7100000000000003)
Row(year=2016, month=5, avgvalue=7.548387096774194)
Row(year=2016, month=4, avgvalue=7.173333333333335)
Row(year=2016, month=3, avgvalue=5.151612903225806)
Row(year=2016, month=2, avgvalue=5.9482758620689635)
Row(year=2016, month=1, avgvalue=5.7612903225806456)
Row(year=2015, month=12, avgvalue=7.4645161290322575)
Row(year=2015, month=11, avgvalue=17.036666666666665)
Row(year=2015, month=10, avgvalue=0.5838709677419354)
Row(year=2015, month=9, avgvalue=27.013333333333335)
Row(year=2015, month=8, avgvalue=6.964516129032257)
Row(year=2015, month=7, avgvalue=30.73548387096774)
Row(year=2015, month=6, avgvalue=20.976666666666667)
Row(year=2015, month=5, avgvalue=24.05806451612903)
Row(year=2015, month=4, avgvalue=4.09)
Row(year=2015, month=3, avgvalue=10.996774193548388)
Row(year=2015, month=2, avgvalue=7.0928571428571425)
Row(year=2015, month=1, avgvalue=15.254838709677419)
Row(year=2014, month=12, avgvalue=9.151612903225807)
Row(year=2014, month=11, avgvalue=13.979999999999999)
Row(year=2014, month=10, avgvalue=18.616129032258065)
Row(year=2014, month=9, avgvalue=12.919999999999998)
Row(year=2014, month=8, avgvalue=23.435483870967737)
Row(year=2014, month=7, avgvalue=5.9322580645161285)
Row(year=2014, month=6, avgvalue=20.036666666666665)
Row(year=2014, month=5, avgvalue=14.967741935483867)
Row(year=2014, month=4, avgvalue=8.469999999999999)
Row(year=2014, month=3, avgvalue=9.435483870967744)
Row(year=2014, month=2, avgvalue=12.489285714285716)
Row(year=2014, month=1, avgvalue=16.148387096774194)
Row(year=2013, month=12, avgvalue=10.906451612903224)
Row(year=2013, month=11, avgvalue=12.366666666666667)
Row(year=2013, month=10, avgvalue=13.903225806451614)
Row(year=2013, month=9, avgvalue=6.983333333333333)
Row(year=2013, month=8, avgvalue=13.95483870967742)
Row(year=2013, month=7, avgvalue=14.080645161290324)
Row(year=2013, month=6, avgvalue=16.353333333333335)
Row(year=2013, month=5, avgvalue=12.367741935483872)
```

Quesrion 6

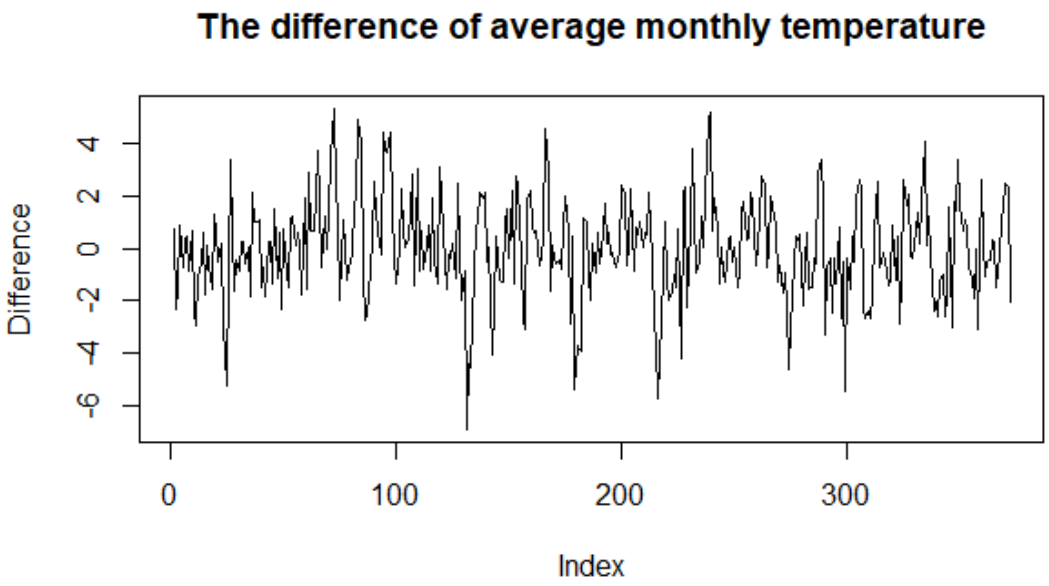
```
#sql spark imports
from pyspark import SparkContext
from pyspark.sql import SQLContext, Row
from pyspark.sql import functions as F
#spark object
sc = SparkContext(appName = "Lab2 Q6")
#create a sql context
sqlContext = SQLContext(sc)
#reading station data
ostergotland_file = sc.textFile("/user/x_rabsh/data/stations-
Ostergotland.csv")
#partition data
stations = ostergotland_file.map(lambda line: line.split(";"))
#station data frame
stationRow = stations.map(lambda x: (x[0],x[1]) )
stationDataFrameString = ["station","name"]
stations = sqlContext.createDataFrame(stationRow,stationDataFrameString)
stations.registerTempTable("stationReadingTable")
# reading temperature data from file
temperatureFile = sc.textFile("/user/x_rabsh/data/temperature-readings.csv")
lines = temperatureFile.map(lambda line: line.split(";"))
#create a temperature dataframe
tempReadingsRows = lines.map(lambda x: (x[0], int(x[1][8:10]) ,int(x[1][0:4])
,int(x[1][5:7]) , float(x[3]) ))
dataFrameString = ["station","date","year","month","temp"]
df = sqlContext.createDataFrame(tempReadingsRows,dataFrameString)
df.registerTempTable("tempReadingTable")
#filter stations from temeperature dataframe
station_filter = df.join(stations, df.station == stations.station)\
                   .select(stations.station,df.date,df.month,df.year,df.temp)
#monthly average temperature
max_temp =
station_filter.groupBy('year','month','date','station').agg(F.max('temp').ali
as('max_temperature'))
min_temp =
station_filter.groupBy('year','month','date','station').agg(F.min('temp').ali
as('min_temperature'))
df_join = max_temp.join(min_temp,(min_temp.year == max_temp.year) &
(min_temp.month == max_temp.month) & (min_temp.date == max_temp.date) &
(min_temp.station == max_temp.station))\

.select(min_temp.year,min_temp.month,min_temp.date,min_temp.station,min_temp.
min_temperature,max_temp.max_temperature)
daily_average = df_join.withColumn('sum_min_max', (df_join.min_temperature +
df_join.max_temperature)/2)
monthly_avg =
daily_average.groupBy('year','month','station').agg(F.avg(daily_average.sum_m
in_max).alias('average'))
```

```
#average by year
year_average_r =
monthly_avg.groupBy('year','month').agg(F.avg(monthly_avg.average).alias('yea
r_average'))
#filter year
year_average_r = year_average_r.where('year >= 1950 and year <= 1980')
#long term average
long_term_average_r =
year_average_r.groupBy(year_average_r.month).agg(F.avg(year_average_r.year_av
erage).alias('long_term_average'))
long_term_average_r = year_average_r.join(long_term_average_r,
long_term_average_r.month == year_average_r.month,'left_outer')\

.select(year_average_r.year,year_average_r.year_average,year_average_r.month,
long_term_average_r.long_term_average)
#find a difference in temperature
difference_temp = long_term_average_r.withColumn('difference_temp',
(long_term_average_r.year_average - long_term_average_r.long_term_average))
difference_temp =
difference_temp.select('year','month','difference_temp').orderBy(['year',
'month'], ascending=False)
output = difference_temp.rdd
output = output.coalesce(1)
output.saveAsTextFile("lab2_q6")
```

```
Row(year=1980, month=12, difference_temp=0.7257955121534725)
Row(year=1980, month=11, difference_temp=-2.3136269939737675)
Row(year=1980, month=10, difference_temp=-1.6017851919152637)
Row(year=1980, month=9, difference_temp=0.8665127318917598)
Row(year=1980, month=8, difference_temp=-0.7276910602404936)
Row(year=1980, month=7, difference_temp=0.3366527113769564)
Row(year=1980, month=6, difference_temp=0.4496010772381691)
Row(year=1980, month=5, difference_temp=-0.8851700536716152)
Row(year=1980, month=4, difference_temp=0.6892537650521522)
Row(year=1980, month=3, difference_temp=-2.3499720865329605)
Row(year=1980, month=2, difference_temp=-2.921488767535764)
Row(year=1980, month=1, difference_temp=-1.0766055276669215)
Row(year=1979, month=12, difference_temp=-0.565333520104592)
Row(year=1979, month=11, difference_temp=0.6004007838040102)
Row(year=1979, month=10, difference_temp=-1.7698692583864277)
Row(year=1979, month=9, difference_temp=0.12817939855843008)
Row(year=1979, month=8, difference_temp=-0.6591548921075479)
Row(year=1979, month=7, difference_temp=-1.5392881488380983)
Row(year=1979, month=6, difference_temp=1.309601077238172)
Row(year=1979, month=5, difference_temp=0.7474057039041426)
Row(year=1979, month=4, difference_temp=-0.527576071777685)
Row(year=1979, month=3, difference_temp=0.1646887902246904)
Row(year=1979, month=2, difference_temp=-2.9874496744449477)
Row(year=1979, month=1, difference_temp=-4.055389646773621)
Row(year=1978, month=12, difference_temp=-5.252864537474319)
Row(year=1978, month=11, difference_temp=3.38009095474418)
Row(year=1978, month=10, difference_temp=0.5014293354928121)
Row(year=1978, month=9, difference_temp=-1.590794960415927)
Row(year=1978, month=8, difference_temp=-0.4170849996344348)
Row(year=1978, month=7, difference_temp=-0.8478902993757309)
Row(year=1978, month=6, difference_temp=0.2673788550159504)
Row(year=1978, month=5, difference_temp=0.2214648436890876)
Row(year=1978, month=4, difference_temp=-0.552063251264864)
Row(year=1978, month=3, difference_temp=0.046574646304095024)
Row(year=1978, month=2, difference_temp=-1.8084661579614307)
Row(year=1978, month=1, difference_temp=2.134560725434815)
Row(year=1977, month=12, difference_temp=1.0280804583900318)
Row(year=1977, month=11, difference_temp=1.017345228248455)
Row(year=1977, month=10, difference_temp=1.1084787377035035)
```


The difference of average monthly temperature
Difference
-6
-4
-2
0
2
4
0
100
200
300
Index