BDA Spark Lab 2

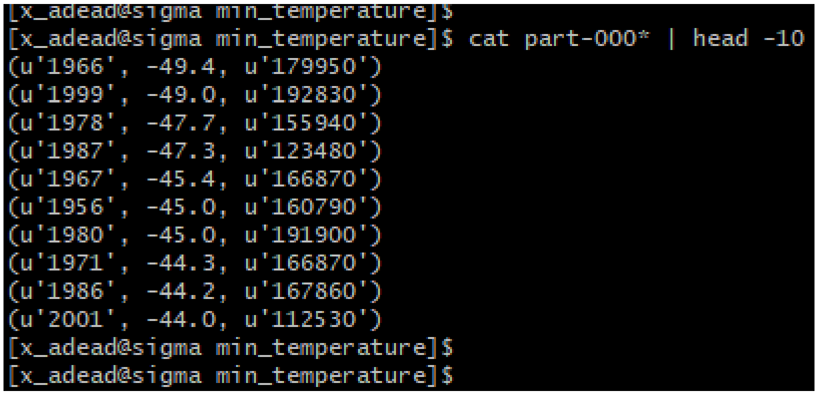
Mohammed Ali{mohal954} and Adesijibomi Aderinto{adead268}

5/2/2022

Question 1

Min Temperature

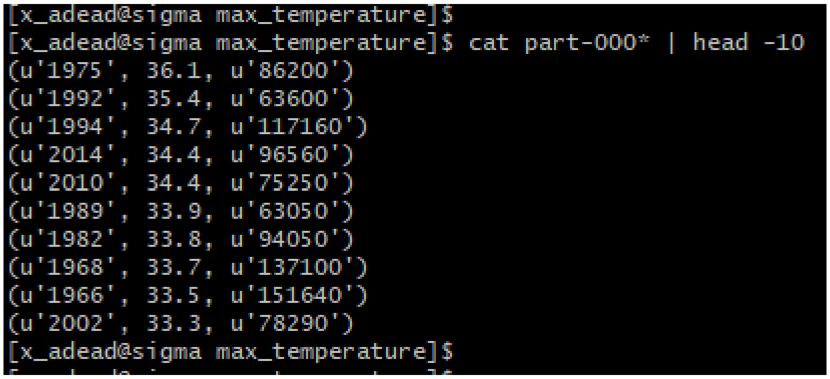
```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 1")
# This path is to the file on hdfs
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
lines = temperature_file.map(lambda line: line.split(";"))

#Maps
# (key, value) = (year,temperature)
year_temperature = lines.map(lambda x: (x[1][0:4],(float(x[3]),x[0])))

#filters
year_temperature = year_temperature.filter(lambda x: int(x[0]) >= 1950 and int(x[0])<= 2014)

#Transformations
min_temperatures = year_temperature.reduceByKey(lambda x,y :x if x<=y else y)
min_temperatures1 = min_temperatures.sortBy(ascending = True, keyfunc=lambda k: k[1][0])
min_temperatures2 = min_temperatures1.map(lambda x: (x[0], float(x[1][0]),x[1][1]))

#Actions
# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
min_temperatures2.saveAsTextFile("BDA/output/min_temperature")
```



```
[x_adead@sigma min_temperature]$
[x_adead@sigma min_temperature]$ cat part-000* | head -10
(u'1966', -49.4, u'179950')
(u'1999', -49.0, u'192830')
(u'1978', -47.7, u'155940')
(u'1987', -47.3, u'123480')
(u'1967', -45.4, u'166870')
(u'1956', -45.0, u'160790')
(u'1980', -45.0, u'191900')
(u'1971', -44.3, u'166870')
(u'1986', -44.2, u'167860')
(u'2001', -44.0, u'112530')
[x_adead@sigma min_temperature]$
[x_adead@sigma min_temperature]$
```

Max Temperature

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 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(";"))

#Maps
# (key, value) = (year,temperature)
year_temperature = lines.map(lambda x: (x[1][0:4],(float(x[3]),x[0])))

#filters
year_temperature = year_temperature.filter(lambda x: int(x[0]) >= 1950 and int(x[0])<= 2014)

#Transformations
max_temperatures = year_temperature.reduceByKey(lambda x,y :x if x>=y else y)
max_temperatures1 = max_temperatures.sortBy(ascending = False, keyfunc=lambda k: k[1][0])
max_temperatures2 = max_temperatures1.map(lambda x: (x[0], float(x[1][0]),x[1][1]))

#Actions
# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
max_temperatures2.saveAsTextFile("BDA/output/max_temperature")
```

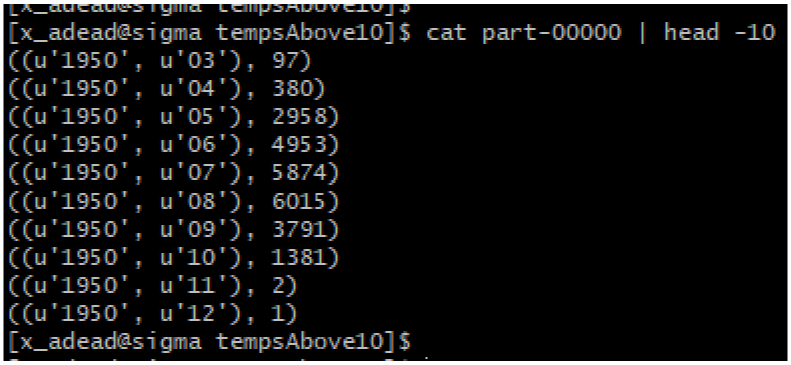


```
[x_adead@sigma max_temperature]$
[x_adead@sigma max_temperature]$ cat part-000* | head -10
(u'1975', 36.1, u'86200')
(u'1992', 35.4, u'63600')
(u'1994', 34.7, u'117160')
(u'2014', 34.4, u'96560')
(u'2010', 34.4, u'75250')
(u'1989', 33.9, u'63050')
(u'1982', 33.8, u'94050')
(u'1968', 33.7, u'137100')
(u'1966', 33.5, u'151640')
(u'2002', 33.3, u'78290')
[x_adead@sigma max_temperature]$
```

Question 2

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 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)
year_temLess10 = lines.map(lambda x: ((x[1][0:4], x[1][5:7]),float(x[3])))

#filters
year_temLess10 = year_temLess10.filter(lambda x: int(x[0][0]) >= 1950 and int(x[0][0]) <= 2014)
tempsAbove10 = year_temLess10.filter(lambda x: x[1] >= 10)

#Transformations
```

```
tempsAbove10counts = tempsAbove10.map(lambda x:((x[0][0], x[0][1]),1))
tempsAbove10 = tempsAbove10counts.reduceByKey(lambda x,y : x+y).sortByKey()

#Actions
# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
tempsAbove10.saveAsTextFile("BDA/output/tempsAbove10")
```



```
[x_adead@sigma tempsAbove10]$
[x_adead@sigma tempsAbove10]$ cat part-00000 | head -10
((u'1950', u'03'), 97)
((u'1950', u'04'), 380)
((u'1950', u'05'), 2958)
((u'1950', u'06'), 4953)
((u'1950', u'07'), 5874)
((u'1950', u'08'), 6015)
((u'1950', u'09'), 3791)
((u'1950', u'10'), 1381)
((u'1950', u'11'), 2)
((u'1950', u'12'), 1)
[x_adead@sigma tempsAbove10]$
```

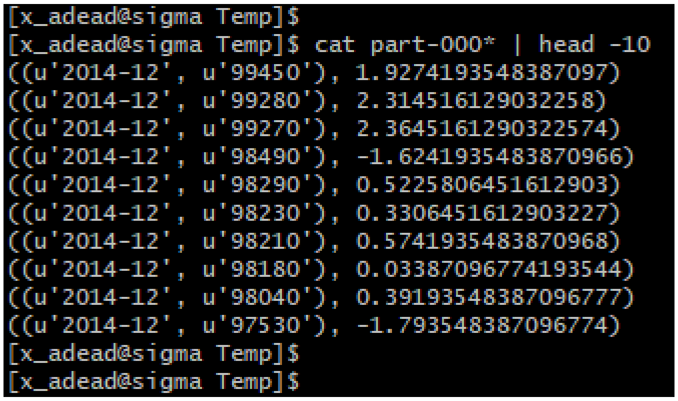
Distinct Temp above 10 degrees per station

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 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(";"))

#Maps
#(key, value) = (year,temperature)
year_temLess10Distinct = lines.map(lambda x: (x[1][0:7], (x[0], float(x[3]))))

#filters
year_temLess10Distinct = year_temLess10Distinct.filter(lambda x: int(x[0][0:4])>=1950 and int(x[0][0:4]
year_temLess10Distinct = year_temLess10Distinct.filter(lambda x: float(x[1][1]) >= 10)

#Transformations
month = year_temLess10Distinct.map(lambda x: (x[0],x[1][0])).distinct()
month_unique = month.map(lambda x: x[0])
month_count = month_unique.map(lambda s : (s, 1))
count = month_count.reduceByKey(lambda a,b : a + b)


#Actions
#Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
count.saveAsTextFile("BDA/output/DistictTempAbove10")
```

Question 3

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 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)
rdd = lines.map(lambda x: ((x[1][0:11],x[0]),float(x[3])))

#filters
average = rdd.filter(lambda x: int(x[0][0][0:4]) >= 1960 and int(x[0][0][0:4]) <= 2014)

max_temperatures = average.reduceByKey(lambda x,y :x if x>=y else y)
min_temperatures = average.reduceByKey(lambda x,y :x if x<=y else y)

temperature_average = max_temperatures.join(min_temperatures)

rdd = temperature_average.map(lambda x: (x[0], x[1][0] + x[1][1]))

daily_temp_rdd = rdd.map(lambda x: (x[0],x[1]/2))

monthly_temp_rdd = daily_temp_rdd.map(lambda x: ((x[0][0][0:7],x[0][1]),x[1]))
.mapValues(lambda x: (x, 1)).reduceByKey(lambda x, y: (x[0]+y[0], x[1]+y[1]))
.mapValues(lambda x: x[0]/x[1]).sortByKey(ascending=False)

monthly_temp_rdd.saveAsTextFile("BDA/output/Temp")
```

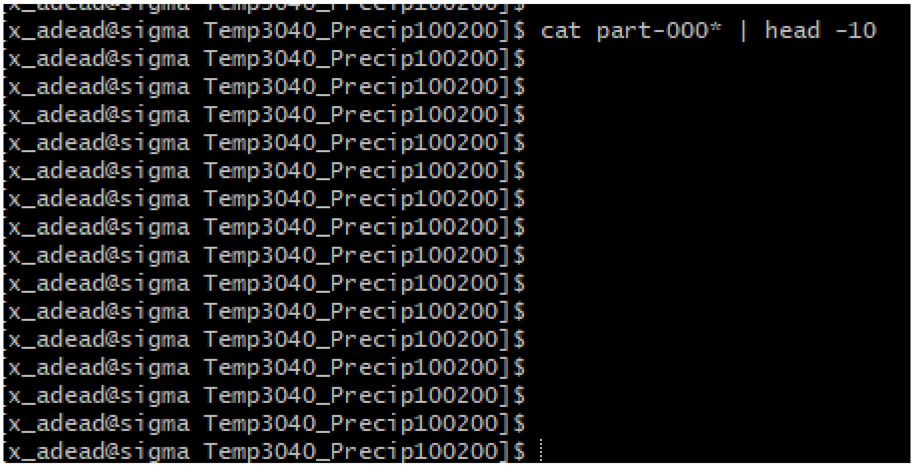


```
[x_adead@sigma Temp]$
[x_adead@sigma Temp]$ cat part-000* | head -10
((u'2014-12', u'99450'), 1.9274193548387097)
((u'2014-12', u'99280'), 2.314516129032258)
((u'2014-12', u'99270'), 2.3645161290322574)
((u'2014-12', u'98490'), -1.6241935483870966)
((u'2014-12', u'98290'), 0.5225806451612903)
((u'2014-12', u'98230'), 0.3306451612903227)
((u'2014-12', u'98210'), 0.5741935483870968)
((u'2014-12', u'98180'), 0.03387096774193544)
((u'2014-12', u'98040'), 0.39193548387096777)
((u'2014-12', u'97530'), -1.793548387096774)
[x_adead@sigma Temp]$
[x_adead@sigma Temp]$
```

Question 4

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 1")
# This path is to the file on hdfs

temp_data = sc.textFile("BDA/input/temperature-readings.csv")
precip_data = sc.textFile("BDA/input/precipitation-readings.csv")

#(key, value) = (year,temperature)
lines_temp = temp_data.map(lambda lines: lines.split(";"))
lines_precip = precip_data.map(lambda lines: lines.split(";"))
```

```
#filter
temp = lines_temp.map(lambda x: (int(x[0]), float(x[3])))
temp_max_station = temp.reduceByKey(lambda a,b: a if a >= b else b)
temp_max_station_boundary = temp_max_station.filter(lambda x: x[1] >= 25 and x[1] <= 30)

precip = lines_precip.map(lambda x: ((int(x[0]), x[1][0:4], x[1][5:7], x[1][8:10]), float(x[3])))
daily_precip = precip.reduceByKey(lambda a,b: a + b)
daily_precip = daily_precip.map(lambda x: (x[0][0], x[1]))
precip_max_station = daily_precip.reduceByKey(lambda a,b: a if a >= b else b)
precip_max_station_boundary = precip_max_station.filter(lambda x: x[1] >= 100 and x[1] <= 200)

joins = temp_max_station_boundary.join(precip_max_station_boundary)

# Following code will save the result into /user/ACCOUNT_NAME/BDA/output folder
joins.saveAsTextFile("BDA/output/Temp3040_Precip100200")
#data_precip_max_station_restrictions.saveAsTextFile("BDA/output/A1")
#data_temp_max_station_restrictions.saveAsTextFile("BDA/output/A2")
```

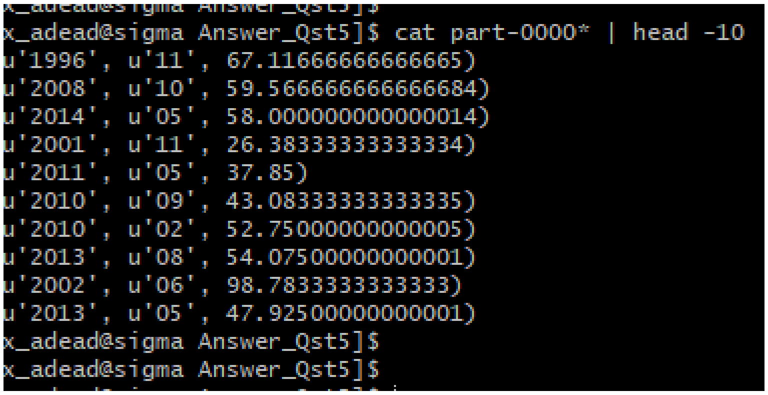


```
x_adead@sigma Temp3040_Precip100200]$ cat part-000* | head -10
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
x_adead@sigma Temp3040_Precip100200]$
```

Question 5

```
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 1")
# This path is to the file on hdfs

station_data = sc.textFile("BDA/input/stations-Ostergotland.csv")
precip_data = sc.textFile("BDA/input/precipitation-readings.csv")

stations = station_data.map(lambda line: line.split(";")[0]).collect()

precip = precip_data.map(lambda line: line.split(";"))

precip_years = precip.filter(lambda x: int(x[1][0:4])>=1993 and int(x[1][0:4])<=2016)
counts = precip_years.filter(lambda x: x[0] in stations)

counts = counts.map(lambda x: ((x[1][0:7], x[0]), float(x[3])))
```

```
preci_station = counts.reduceByKey(lambda a,b: a+b)

avg_preci = preci_station.map(lambda x: (x[0][0],  (x[1], 1)))

preci_monthly = avg_preci.reduceByKey(lambda a,b: (a[0]+b[0],a[1]+b[1]))
preci_monthly = preci_monthly.map(lambda x: (x[0][0:4], x[0][5:7], x[1][0]/x[1][1]))

preci_monthly.saveAsTextFile("BDA/output/Answer_Qst5")
```



```
x_adead@sigma Answer_Qst5]$
x_adead@sigma Answer_Qst5]$ cat part-0000* | head -10
u'1996', u'11', 67.11666666666665)
u'2008', u'10', 59.566666666666684)
u'2014', u'05', 58.000000000000014)
u'2001', u'11', 26.383333333333334)
u'2011', u'05', 37.85)
u'2010', u'09', 43.083333333333335)
u'2010', u'02', 52.75000000000005)
u'2013', u'08', 54.07500000000001)
u'2002', u'06', 98.78333333333333)
u'2013', u'05', 47.92500000000001)
x_adead@sigma Answer_Qst5]$
x_adead@sigma Answer_Qst5]$
```