Spark Exercicios

March 28, 2020

[2]:

```
import os
os.environ['PYSPARK_PYTHON'] = '/usr/bin/python3'

import pyspark
conf = pyspark.SparkConf()

conf.setMaster('spark://172.18.0.7:7077')
conf.set('spark.executor.memory', '1g')

sc = pyspark.SparkContext.getOrCreate()
sc.stop()
sc = pyspark.SparkContext(conf = conf)
```

[3]:

```
arquivoRDD = sc.textFile('hdfs://172.18.0.9:9000/ocorrencias_criminais_sample.
 ↪csv')
```

[6]:

```
arquivoRDD.take(5)
```

[6]:

```
['11;04;2011;051XX S INDIANA AVE;PUBLIC PEACE VIOLATION;RECKLESS
CONDUCT;ALLEY;41.801097014;-87.621002356',
 '07;05;2015;0000X N LONG AVE;OFFENSE INVOLVING CHILDREN;ENDANGERING LIFE/HEALTH
CHILD;APARTMENT;41.880913243;-87.760142556',
 '14;08;2011;005XX N LECLAIRE AVE;THEFT;$500 AND
UNDER;STREET;41.889968929;-87.752973368',
 '06;05;2004;024XX N BURLING ST;BURGLARY;UNLAWFUL ENTRY;RESIDENCE-
GARAGE;41.926218147;-87.647562173',
 '09;12;2010;062XX S EVANS AVE;WEAPONS VIOLATION;UNLAWFUL POSS OF
HANDGUN;STREET;41.781367064;-87.607207607']
```

[7]:

```
arquivoRDD.map(lambda linha:[linha.split(';')[2], 1])\
    .reduceByKey(lambda x, y: x+ y)\
    .sortBy(lambda x:x[1]).collect()
```

[7]:

```
[('2013', 789),
 ('2014', 55859),
 ('2009', 81834),
 ('2008', 88382),
 ('2007', 91903),
 ('2005', 93688),
 ('2006', 94071),
 ('2004', 97951),
 ('2003', 100106),
 ('2018', 115224),
 ('2002', 123451),
 ('2012', 143950),
 ('2015', 208264),
 ('2017', 209939),
 ('2016', 239556),
 ('2011', 335849),
 ('2010', 355655),
 ('2001', 465932)]
```

1 1 - Quantidade de crimes por ano

```
[44]: arquivoRDD.map(lambda x: [x.split(';')[2], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .sortBy(lambda x:x[0]).collect()
```

```
[44]: [('2001', 465932),
       ('2002', 123451),
       ('2003', 100106),
       ('2004', 97951),
       ('2005', 93688),
       ('2006', 94071),
       ('2007', 91903),
       ('2008', 88382),
       ('2009', 81834),
       ('2010', 355655),
       ('2011', 335849),
       ('2012', 143950),
       ('2013', 789),
       ('2014', 55859),
       ('2015', 208264),
       ('2016', 239556),
       ('2017', 209939),
       ('2018', 115224)]
```

2 2 - Quantidade de crimes por ano que sejam do tipo NARCOTICS

```
[43]: arquivoRDD.filter(lambda x: x.split(';')[4] == 'NARCOTICS')\
          .map(lambda x: [x.split(';')[2], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .sortBy(lambda x:x[0]).collect()
```

```
[43]: [('2001', 49515),
       ('2002', 13265),
       ('2003', 11687),
       ('2004', 12150),
       ('2005', 11867),
       ('2006', 11977),
       ('2007', 11531),
       ('2008', 9548),
       ('2009', 9025),
       ('2010', 42397),
       ('2011', 37775),
       ('2012', 15479),
       ('2013', 10),
       ('2014', 6111),
       ('2015', 16305),
       ('2016', 11572),
       ('2017', 9077),
       ('2018', 5673)]
```

3 3 - Quantidade de crimes por ano, que sejam do tipo NARCOTICS, e tenham ocorrido em dias pares

```
[42]: arquivoRDD.filter(lambda x: x.split(';')[4] == 'NARCOTICS')\
          .filter(lambda x: int(x.split(';')[0]) % 2 == 0)\
          .map(lambda x: [x.split(';')[2], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .sortBy(lambda x:x[0]).collect()
```

```
[42]: [('2001', 24131),
       ('2002', 6514),
       ('2003', 5742),
       ('2004', 5939),
       ('2005', 5650),
       ('2006', 5833),
       ('2007', 5738),
       ('2008', 4644),
       ('2009', 4443),
       ('2010', 20695),
       ('2011', 18435),
       ('2012', 7536),
       ('2013', 4),
       ('2014', 3065),
       ('2015', 7955),
       ('2016', 5735),
       ('2017', 4442),
       ('2018', 2782)]
```

4 4 - Mês com maior ocorrência de crimes

[22]:

```
arquivoRDD.map(lambda x: [x.split(';')[1], 1])\
     .reduceByKey(lambda x, y: x + y)\
     .max(lambda x: x[1])
```

[22]:

```
('05', 289160)
```

5 5 - Mês com menor ocorrência de crimes;

[12]:

```
arquivoRDD.map(lambda x: [x.split(';')[1], 1])\
     .reduceByKey(lambda x, y: x + y)\
     .min(lambda x: x[1])
```

[12]:

```
('02', 181320)
```

6 6 - Mês por ano com a maior ocorrência de crimes

[13]:

```
arquivoRDD.map(lambda x: [x.split(';')[1] + '-' + x.split(';')[2], 1])\
     .reduceByKey(lambda x, y: x + y)\
     .max(lambda x: x[1])
```

[13]:

```
('07-2001', 43757)
```

7 7 - Mês com a maior ocorrência de crimes do tipo “DECEPTIVE PRACTICE”

```
[27]: arquivoRDD.filter(lambda x: x.split(';')[4] == 'DECEPTIVE PRACTICE')\
          .map(lambda x: [x.split(';')[1], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .max(lambda x: x[1])
```

```
[27]: ('04', 11589)
```

8 8 - Dia do ano com a maior ocorrência de crimes;

```
[28]: arquivoRDD.map(lambda x: [x.split(';')[0] + '-' + x.split(';')[1], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .max(lambda x: x[1])
```

```
[28]: ('01-01', 11325)
```

9 9 - Quantidade de crimes por ano que sejam do tipo NARCOTICS, que ocorreram na localização descrita como STREET

```
[41]: arquivoRDD.filter(lambda x: x.split(';')[4] == 'NARCOTICS')\
          .filter(lambda x: x.split(';')[6] == 'STREET')\
          .map(lambda x: [x.split(';')[2], 1])\
          .reduceByKey(lambda x, y: x + y)\
          .sortBy(lambda x:x[0]).collect()
```

```
[41]: [('2001', 23178),
       ('2002', 6085),
       ('2003', 5437),
       ('2004', 4923),
       ('2005', 4536),
       ('2006', 4227),
       ('2007', 3959),
       ('2008', 3293),
       ('2009', 2861),
       ('2010', 12699),
       ('2011', 11356),
       ('2012', 4337),
       ('2013', 3),
       ('2014', 2066),
       ('2015', 5404),
       ('2016', 3626),
       ('2017', 2507),
       ('2018', 1838)]
```

10 10 - Quantidade de crimes por ano que sejam do tipo NARCOTICS, que ocorreram na localização descrita como STREET, no raio de tamanho 2 da latitude 41 e longitude -87

[40]:

```
arquivoRDD.filter(lambda x: x.split(';')[4] == 'NARCOTICS')\
    .filter(lambda x: x.split(';')[6] == 'STREET')\
    .filter(lambda x: float(x.split(';')[7])>= 39)\
    .filter(lambda x: float(x.split(';')[7])<= 43)\
    .filter(lambda x: float(x.split(';')[8])<= -85)\
    .filter(lambda x: float(x.split(';')[8])>= -89)\
    .map(lambda x: [x.split(';')[2], 1])\
    .reduceByKey(lambda x, y: x + y)\
    .sortBy(lambda x:x[0]).collect()
```

[40]:

```
[('2001', 23178),
 ('2002', 6085),
 ('2003', 5437),
 ('2004', 4923),
 ('2005', 4536),
 ('2006', 4227),
 ('2007', 3959),
 ('2008', 3293),
 ('2009', 2861),
 ('2010', 12698),
 ('2011', 11355),
 ('2012', 4336),
 ('2013', 2),
 ('2014', 2065),
 ('2015', 5404),
 ('2016', 3626),
 ('2017', 2507),
 ('2018', 1838)]
```

11 11 - Dia da semana com maior quantidade de ocorrências criminal

[33]:
```
arquivoRDD.take(5)
```

[33]:
```
['11;04;2011;051XX S INDIANA AVE;PUBLIC PEACE VIOLATION;RECKLESS
CONDUCT;ALLEY;41.801097014;-87.621002356',
 '07;05;2015;0000X N LONG AVE;OFFENSE INVOLVING CHILDREN;ENDANGERING LIFE/HEALTH
CHILD;APARTMENT;41.880913243;-87.760142556',
 '14;08;2011;005XX N LECLAIRE AVE;THEFT;$500 AND
UNDER;STREET;41.889968929;-87.752973368',
 '06;05;2004;024XX N BURLING ST;BURGLARY;UNLAWFUL ENTRY;RESIDENCE-
GARAGE;41.926218147;-87.647562173',
 '09;12;2010;062XX S EVANS AVE;WEAPONS VIOLATION;UNLAWFUL POSS OF
HANDGUN;STREET;41.781367064;-87.607207607']
```

[62]:
```
import datetime

def dayweek_map(x):
    dict_weekday = {
        0:'Segunda',
        1:'Terça',
        2:'Quarta',
        3:'Quinta',
        4:'Sexta',
        5:'Sabado',
        6:'Domingo'
    }
    day = datetime.datetime(int(x.split(';')[2]), int(x.split(';')[1]), int(x.
 ↪split(';')[0])).weekday()
    return [dict_weekday[day], 1]

arquivoRDD.map(dayweek_map)\
    .reduceByKey(lambda x, y: x + y)\
    .max(lambda x: x[1])
```

[62]:
```
('Sexta', 433240)
```

[ ]:
```

```