databrickstest

Spark Assignment 1

Fubang ZHAO

```
from pyspark import SparkContext, SparkConf


sc = SparkContext.getOrCreate()
rdd = sc.wholeTextFiles("FileStore/tables/kl5f55dz1509040574264")
f1 = rdd.map(lambda kv: (kv[0].split("/")[-1].split(".txt")[0], kv[1]))
f2 = f1.flatMapValues(lambda v: v.split("\r\n"))
data = f2.map(lambda kv: (kv[0].split("_")[0], kv[0], kv[1].split(" ")[0],
kv[1].split(" ")[1]))# (city, city&num, month, revenue)
```

The First question:

Average monthly income of the shop in France (on 1 year data)

```
f3 = data.map(lambda kv: (kv[2], int(kv[3]))) #(month, income)
Avg_month = f3.groupByKey().mapValues(lambda x: sum(x) / len(x))
Avg_month.collect()
```

```
Out[104]:
[(u'FEB', 19),
 (u'AUG', 23),
 (u'APR', 20),
 (u'JUN', 27),
 (u'JUL', 21),
 (u'JAN', 20),
 (u'MAY', 22),
 (u'NOV', 24),
 (u'MAR', 17),
 (u'DEC', 29),
 (u'OCT', 26),
 (u'SEP', 25)]
```

The second question:

Total revenue per city per year

```
f4 = data.map(lambda kv: (kv[0], int(kv[3]))) #(city, income)
Total_city = f4.groupByKey().mapValues(lambda x: sum(x))
Total_city.collect()
```

```
Out[101]:
[(u'anger', 166),
 (u'paris', 1568),
 (u'lyon', 193),
 (u'troyes', 214),
 (u'toulouse', 177),
 (u'orlean', 196),
 (u'rennes', 180),
 (u'nice', 203),
 (u'nantes', 207),
 (u'marseilles', 515)]
```

The third question:

Average monthly income of the shop in each city

```
f5 = data.map(lambda kv: (kv[0] + "_" + kv[2], int(kv[3]))) #(city_month,
income)
Avg_month_city = f5.groupByKey().mapValues(lambda x: sum(x) / len(x))
print Avg_month_city.take(200)
```

```
[(u'troyes_NOV', 11), (u'nantes_SEP', 13), (u'toulouse_AUG', 11), (u'troyes_J
AN', 21), (u'nice_DEC', 29), (u'paris_NOV', 48), (u'anger_APR', 15), (u'lyon_
MAY', 12), (u'nantes_DEC', 24), (u'toulouse_JAN', 12), (u'nantes_MAR', 20),
 (u'rennes_MAR', 10), (u'lyon_APR', 15), (u'rennes_DEC', 20), (u'nice_MAR', 2
0), (u'anger_JUL', 19), (u'troyes_JUN', 25), (u'anger_JUN', 15), (u'orlean_MA
R', 14), (u'troyes_MAY', 15), (u'lyon_JAN', 13), (u'toulouse_APR', 11), (u'an
ger_NOV', 14), (u'paris_APR', 38), (u'toulouse_MAY', 11), (u'troyes_APR', 1
7), (u'marseilles_OCT', 28), (u'paris_FEB', 33), (u'paris_MAY', 50), (u'toulo
use_NOV', 12), (u'anger_JAN', 13), (u'nantes_OCT', 14), (u'nice_OCT', 18),
 (u'lyon_JUL', 19), (u'anger_AUG', 15), (u'orlean_OCT', 8), (u'anger_FEB', 1
2), (u'lyon_FEB', 12), (u'lyon_AUG', 25), (u'toulouse_FEB', 13), (u'troyes_FE
B', 21), (u'marseilles_MAR', 16), (u'lyon_JUN', 15), (u'lyon_NOV', 22), (u'pa
ris_AUG', 41), (u'marseilles_DEC', 26), (u'orlean_DEC', 26), (u'paris_JUL', 3
3), (u'anger_MAY', 12), (u'toulouse_JUL', 19), (u'paris_JAN', 38), (u'toulous
e_JUN', 18), (u'troyes_JUL', 11), (u'nice_SEP', 23), (u'rennes_OCT', 18),
 (u'troyes_AUG', 22), (u'rennes_SEP', 23), (u'marseilles_SEP', 23), (u'orlean
_SEP', 13), (u'paris_JUN', 55), (u'toulouse_MAR', 14), (u'toulouse_OCT', 14),
(u'nice_JAN', 16), (u'lyon_MAR', 14), (u'nantes_FEB', 15), (u'nantes_NOV', 1
4), (u'lyon_OCT', 11), (u'anger_OCT', 8), (u'nantes_APR', 12), (u'nice_AUG',
 11), (u'nice_NOV', 14), (u'troyes_MAR', 11), (u'marseilles_FEB', 16), (u'lyo
n_DEC', 22), (u'rennes_AUG', 11), (u'paris_OCT', 56), (u'nantes_JUN', 28),
 (u'nantes_JUL', 19), (u'nantes_MAY', 21), (u'marseilles_NOV', 24), (u'paris_
DEC', 52), (u'marseilles_JAN', 16), (u'rennes_FEB', 18), (u'orlean_JAN', 13),
(u'anger_SEP', 13), (u'rennes_JUN', 13), (u'rennes_MAY', 11), (u'nice_JUN', 1
8), (u'orlean_MAY', 12), (u'orlean_APR', 15), (u'toulouse_DEC', 19), (u'troye
s_OCT', 28), (u'rennes_JAN', 19), (u'orlean_NOV', 24), (u'nice_APR', 9), (u'r
ennes_APR', 9), (u'toulouse_SEP', 23), (u'troyes_DEC', 11), (u'marseilles_JU
N', 25), (u'marseilles_JUL', 21), (u'orlean_AUG', 25), (u'anger_DEC', 16),
 (u'nice_MAY', 11), (u'paris_SEP', 48), (u'anger_MAR', 14), (u'marseilles_AU
G', 22), (u'troyes_SEP', 21), (u'nice_JUL', 19), (u'nantes_AUG', 11), (u'nant
```

```
es_JAN', 16), (u'orlean_FEB', 12), (u'rennes_JUL', 14), (u'paris_MAR', 26),
 (u'lyon_SEP', 13), (u'marseilles_APR', 22), (u'nice_FEB', 15), (u'rennes_NO
V', 14), (u'orlean_JUN', 15), (u'orlean_JUL', 19), (u'marseilles_MAY', 18)]
```

The 4th question:

Total revenue per store per year

```
f6 = data.map(lambda kv: (kv[1], int(kv[3]))) #(store, income)
Total_store = f6.groupByKey().mapValues(lambda x: sum(x))
Total_store.collect()
```

```
Out[102]:
[(u'troyes', 214),
 (u'lyon', 193),
 (u'toulouse', 177),
 (u'marseilles_2', 231),
 (u'anger', 166),
 (u'paris_3', 330),
 (u'paris_1', 596),
 (u'orlean', 196),
 (u'marseilles_1', 284),
 (u'rennes', 180),
 (u'nantes', 207),
 (u'paris_2', 642),
 (u'nice', 203)]
```

The 5th question:

The store that achieves the best performance in each month

```
f7 = data.map(lambda kv: (kv[2], kv[1], int(kv[3]))) #(month, store, income)
best_store = (f7.map(lambda x: (x[0], x)).reduceByKey(lambda x1, x2: max(x1,
x2, key=lambda x: x[-1])) .values()) # Drop keys
best_store.collect()
```

```
Out[110]:
[(u'FEB', u'paris_2', 42),
 (u'AUG', u'paris_2', 45),
 (u'APR', u'paris_1', 57),
 (u'JUN', u'paris_2', 85),
 (u'JUL', u'paris_1', 61),
 (u'JAN', u'paris_1', 51),
 (u'MAY', u'paris_2', 72),
 (u'NOV', u'paris_2', 64),
 (u'MAR', u'paris_2', 44),
 (u'DEC', u'paris_1', 71),
 (u'OCT', u'paris_1', 68),
 (u'SEP', u'paris_2', 63)]
```