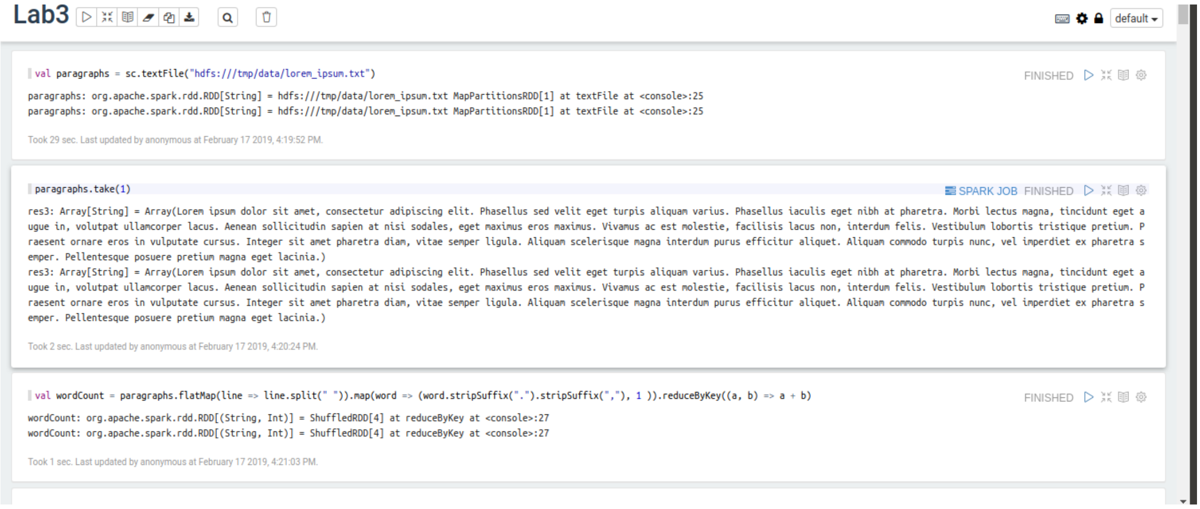
CS4830: Big Data Laboratory

Lab 3

Divya K Raman, EE15B085

Answer 1

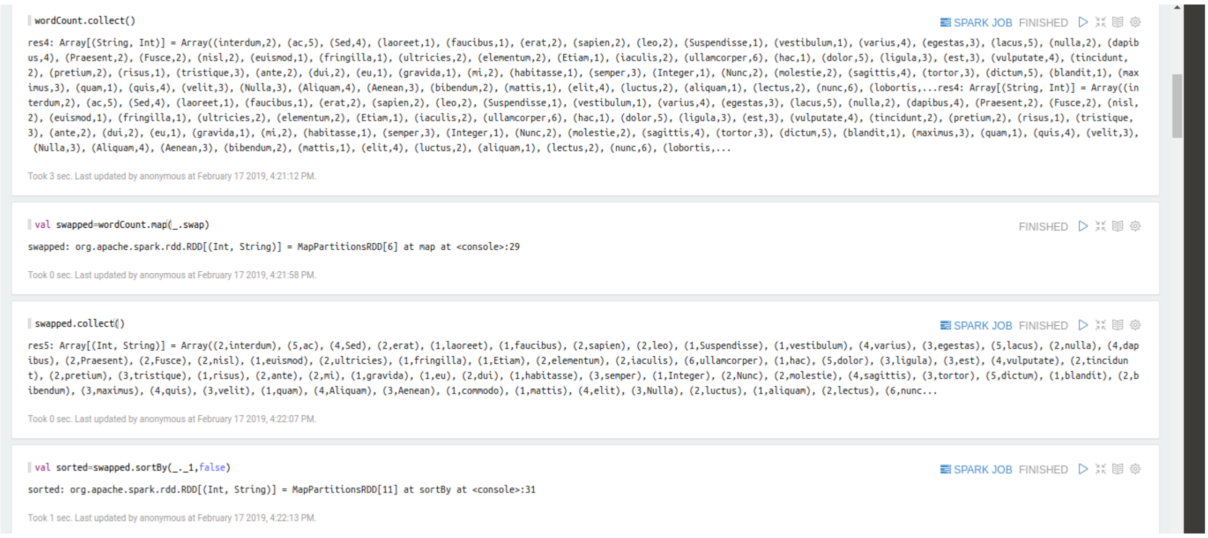
California state has the highest number of purchases and los angeles city has the highest number of purchases.



```
val paragraphs = sc.textFile("hdfs:///tmp/data/lorem_ipsum.txt")

paragraphs: org.apache.spark.rdd.RDD[String] = hdfs:///tmp/data/lorem_ipsum.txt MapPartitionsRDD[1] at textFile at <console>:25
paragraphs: org.apache.spark.rdd.RDD[String] = hdfs:///tmp/data/lorem_ipsum.txt MapPartitionsRDD[1] at textFile at <console>:25

Took 29 sec. Last updated by anonymous at February 17 2019, 4:19:52 PM.
```

```
paragraphs.take(1)

res3: Array[String] = Array(Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sed velit eget turpis aliquam varius. Phasellus iaculis eget nibh at pharetra. Morbi lectus magna, tincidunt eget a
ugue in, volutpat ullamcorper lacus. Aenean sollicitudin sapien at nisi sodales, eget maximus eros maximus. Vivamus ac est molestie, facilisis lacus non, interdum felis. Vestibulum lobortis tristique pretium. P
raesent ornare eros in vulputate cursus. Integer sit amet pharetra diam, vitae semper ligula. Aliquam scelerisque magna interdum purus efficitur aliquet. Aliquam commodo turpis nunc, vel imperdiet ex pharetra s
emper. Pellentesque posuere pretium magna eget lacinia.)
res3: Array[String] = Array(Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sed velit eget turpis aliquam varius. Phasellus iaculis eget nibh at pharetra. Morbi lectus magna, tincidunt eget a
ugue in, volutpat ullamcorper lacus. Aenean sollicitudin sapien at nisi sodales, eget maximus eros maximus. Vivamus ac est molestie, facilisis lacus non, interdum felis. Vestibulum lobortis tristique pretium. P
raesent ornare eros in vulputate cursus. Integer sit amet pharetra diam, vitae semper ligula. Aliquam scelerisque magna interdum purus efficitur aliquet. Aliquam commodo turpis nunc, vel imperdiet ex pharetra s
emper. Pellentesque posuere pretium magna eget lacinia.)

Took 2 sec. Last updated by anonymous at February 17 2019, 4:20:24 PM.
```

```
val wordCount = paragraphs.flatMap(line => line.split(" ")).map(word => (word.stripSuffix(".").stripSuffix(","), 1 )).reduceByKey((a, b) => a + b)

wordCount: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:27
wordCount: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:27

Took 1 sec. Last updated by anonymous at February 17 2019, 4:21:03 PM.
```

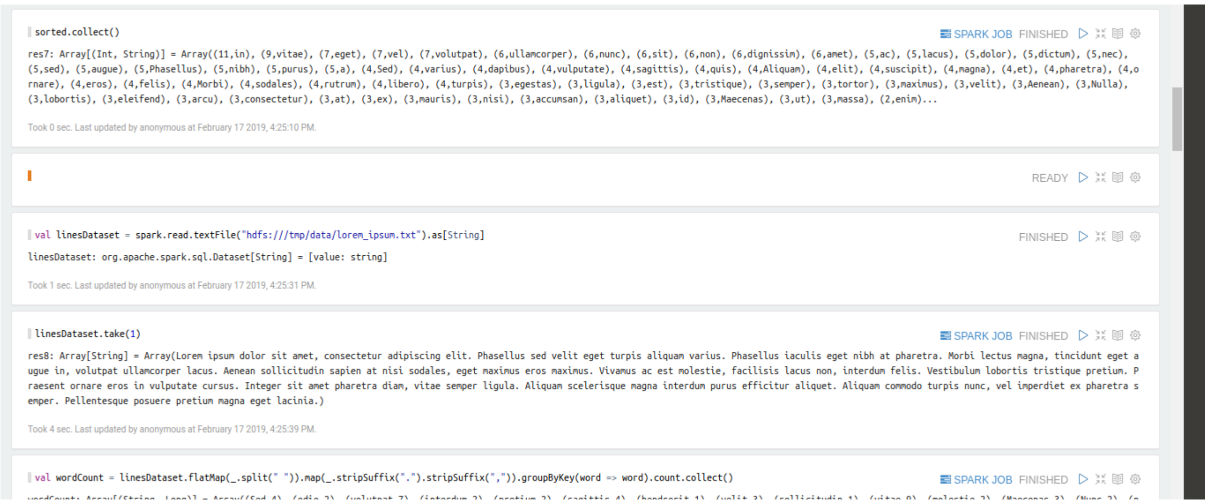


```
wordCount.collect()

res4: Array[(String, Int)] = Array((interdum,2), (ac,5), (Sed,4), (laoreet,1), (faucibus,1), (erat,2), (sapien,2), (leo,2), (Suspendisse,1), (vestibulum,1), (varius,4), (egestas,3), (lacus,5), (nulla,2), (dapib
us,4), (Praesent,2), (Fusce,2), (nisl,2), (euismod,1), (fringilla,1), (ultricies,2), (elementum,2), (Etiam,1), (iaculis,2), (ullamcorper,6), (hac,1), (dolor,5), (ligula,3), (est,3), (vulputate,4), (tincidunt,
2), (pretium,2), (risus,1), (tristique,3), (ante,2), (dui,2), (eu,1), (gravida,1), (mi,2), (habitasse,1), (semper,3), (Integer,1), (Nunc,2), (molestie,2), (sagittis,4), (tortor,3), (dictum,5), (blandit,1), (max
imus,3), (quam,1), (quis,4), (velit,3), (Nulla,3), (Aliquam,4), (Aenean,3), (bibendum,2), (mattis,1), (elit,4), (luctus,2), (aliquam,1), (lectus,2), (nunc,6), (lobortis,...res4: Array[(String, Int)] = Array((in
terdum,2), (ac,5), (Sed,4), (laoreet,1), (faucibus,1), (erat,2), (sapien,2), (leo,2), (Suspendisse,1), (vestibulum,1), (varius,4), (egestas,3), (lacus,5), (nulla,2), (dapibus,4), (Praesent,2), (Fusce,2), (nisl,
2), (euismod,1), (fringilla,1), (ultricies,2), (elementum,2), (Etiam,1), (iaculis,2), (ullamcorper,6), (hac,1), (dolor,5), (ligula,3), (est,3), (vulputate,4), (tincidunt,2), (pretium,2), (risus,1), (tristique,
3), (ante,2), (dui,2), (eu,1), (gravida,1), (mi,2), (habitasse,1), (semper,3), (Integer,1), (Nunc,2), (molestie,2), (sagittis,4), (tortor,3), (dictum,5), (blandit,1), (maximus,3), (quam,1), (quis,4), (velit,3),
 (Nulla,3), (Aliquam,4), (Aenean,3), (bibendum,2), (mattis,1), (elit,4), (luctus,2), (aliquam,1), (lectus,2), (nunc,6), (lobortis,...

Took 3 sec. Last updated by anonymous at February 17 2019, 4:21:12 PM.
```

```
val swapped=wordCount.map(_.swap)

swapped: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[6] at map at <console>:29

Took 0 sec. Last updated by anonymous at February 17 2019, 4:21:58 PM.
```

```
swapped.collect()

res5: Array[(Int, String)] = Array((2,interdum), (5,ac), (4,Sed), (2,erat), (1,laoreet), (1,faucibus), (2,sapien), (2,leo), (1,Suspendisse), (1,vestibulum), (4,varius), (3,egestas), (5,lacus), (2,nulla), (4,dap
ibus), (2,Praesent), (2,Fusce), (2,nisl), (1,euismod), (2,ultricies), (1,fringilla), (1,Etiam), (2,elementum), (2,iaculis), (6,ullamcorper), (1,hac), (5,dolor), (3,ligula), (3,est), (4,vulputate), (2,tincidun
t), (2,pretium), (3,tristique), (1,risus), (2,ante), (2,mi), (1,gravida), (1,eu), (2,dui), (1,habitasse), (3,semper), (1,Integer), (2,Nunc), (2,molestie), (4,sagittis), (3,tortor), (5,dictum), (1,blandit), (2,b
ibendum), (3,maximus), (4,quis), (3,velit), (1,quam), (4,Aliquam), (3,Aenean), (1,commodo), (1,mattis), (4,elit), (3,Nulla), (2,luctus), (1,aliquam), (2,lectus), (6,nunc...

Took 0 sec. Last updated by anonymous at February 17 2019, 4:22:07 PM.
```

```
val sorted=swapped.sortBy(_._1,false)

sorted: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[11] at sortBy at <console>:31

Took 1 sec. Last updated by anonymous at February 17 2019, 4:22:13 PM.
```



```
sorted.collect()
```

```
res7: Array[(Int, String)] = Array((11,in), (9,vitae), (7,eget), (7,vel), (7,volutpat), (6,ullamcorper), (6,nunc), (6,sit), (6,non), (6,dignissim), (6,amet), (5,ac), (5,lacus), (5,dolor), (5,dictum), (5,nec),
(5,sed), (5,augue), (5,Phasellus), (5,nibh), (5,purus), (5,a), (4,Sed), (4,varius), (4,dapibus), (4,vulputate), (4,sagittis), (4,quis), (4,Aliquam), (4,elit), (4,suscipit), (4,magna), (4,et), (4,pharetra), (4,o
rnare), (4,eros), (4,felis), (4,Morbi), (4,sodales), (4,rutrum), (4,libero), (4,turpis), (3,egestas), (3,ligula), (3,est), (3,tristique), (3,semper), (3,tortor), (3,maximus), (3,velit), (3,Aenean), (3,Nulla),
(3,lobortis), (3,eleifend), (3,arcu), (3,consectetur), (3,at), (3,ex), (3,mauris), (3,nisi), (3,accumsan), (3,aliquet), (3,id), (3,Maecenas), (3,ut), (3,massa), (2,enim)...
```

Took 0 sec. Last updated by anonymous at February 17 2019, 4:25:10 PM.

```
val linesDataset = spark.read.textFile("hdfs:///tmp/data/lorem_ipsum.txt").as[String]
```

```
linesDataset: org.apache.spark.sql.Dataset[String] = [value: string]
```

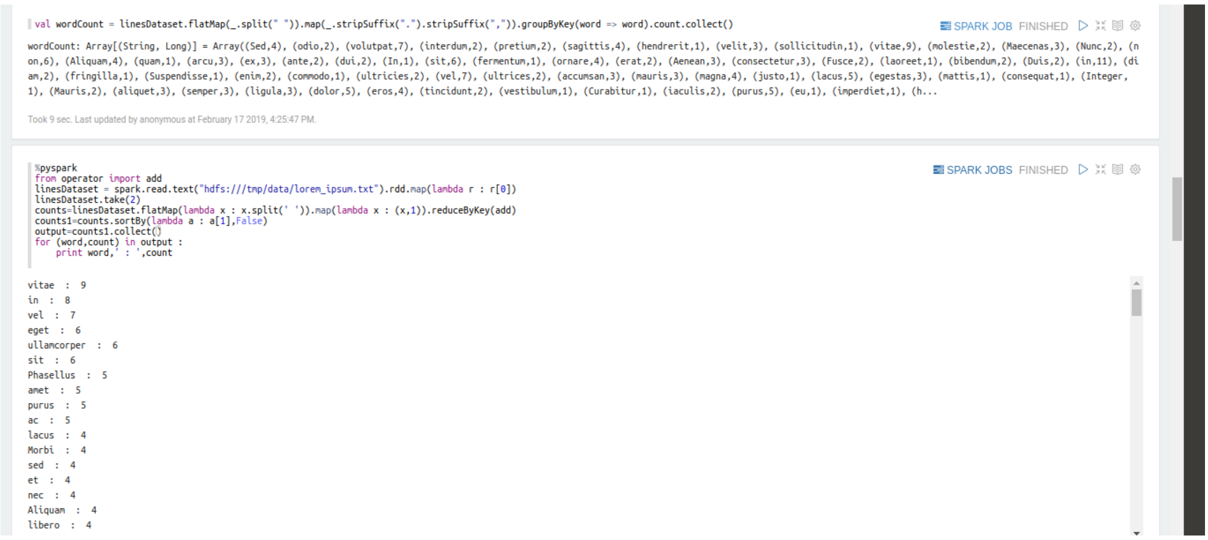
Took 1 sec. Last updated by anonymous at February 17 2019, 4:25:31 PM.

```
linesDataset.take(1)
```

```
res8: Array[String] = Array(Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sed velit eget turpis aliquam varius. Phasellus iaculis eget nibh at pharetra. Morbi lectus magna, tincidunt eget a
ugue in, volutpat ullamcorper lacus. Aenean sollicitudin sapien at nisi sodales, eget maximus eros maximus. Vivamus ac est molestie, facilisis lacus non, interdum felis. Vestibulum lobortis tristique pretium. P
raesent ornare eros in vulputate cursus. Integer sit amet pharetra diam, vitae semper ligula. Aliquam scelerisque magna interdum purus efficitur aliquet. Aliquam commodo turpis nunc, vel imperdiet ex pharetra s
emper. Pellentesque posuere pretium magna eget lacinia.)
```

Took 4 sec. Last updated by anonymous at February 17 2019, 4:25:39 PM.

```
val wordCount = linesDataset.flatMap(_.split(" ")).map(_.stripSuffix(".").stripSuffix(",")).groupByKey(word => word).count.collect()
```



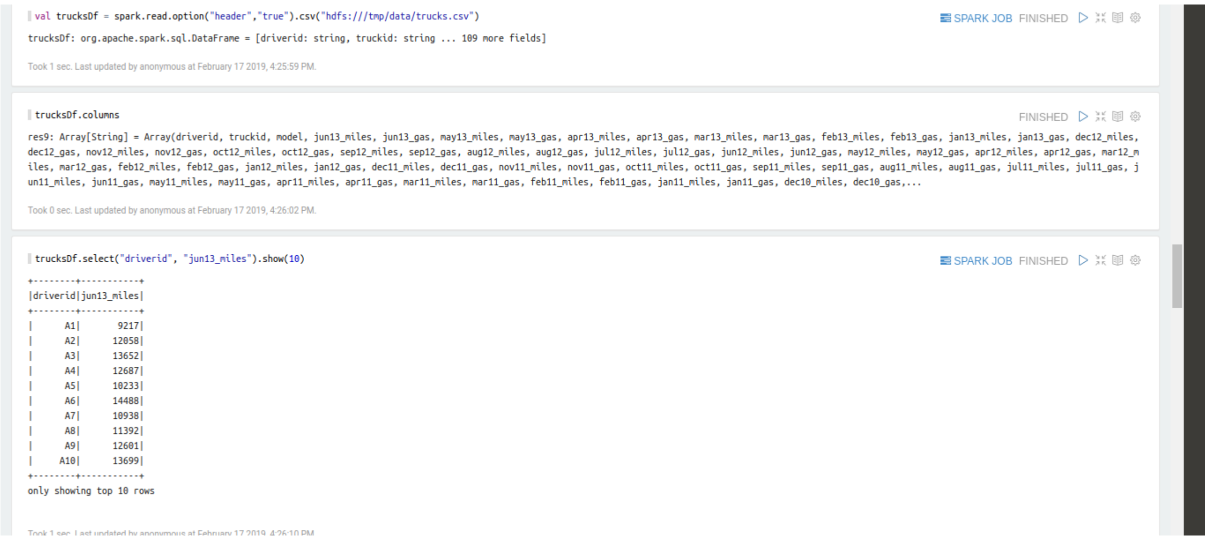
```
val wordCount = linesDataset.flatMap(_.split(" ")).map(_.stripSuffix(".").stripSuffix(",")).groupByKey(word => word).count.collect()
```

```
wordCount: Array[(String, Long)] = Array((Sed,4), (odio,2), (volutpat,7), (interdum,2), (pretium,2), (sagittis,4), (hendrerit,1), (velit,3), (sollicitudin,1), (vitae,9), (molestie,2), (Maecenas,3), (Nunc,2), (n
on,6), (Aliquam,4), (quam,1), (arcu,3), (ex,3), (ante,2), (dui,2), (In,1), (sit,6), (fermentum,1), (ornare,4), (erat,2), (Aenean,3), (consectetur,3), (Fusce,2), (laoreet,1), (bibendum,2), (Duis,2), (in,11), (di
am,2), (fringilla,1), (Suspendisse,1), (enim,2), (commodo,1), (ultricies,2), (vel,7), (ultrices,2), (accumsan,3), (mauris,3), (magna,4), (justo,1), (lacus,5), (egestas,3), (mattis,1), (consequat,1), (Integer,
1), (Mauris,2), (aliquet,3), (semper,3), (ligula,3), (dolor,5), (eros,4), (tincidunt,2), (vestibulum,1), (Curabitur,1), (iaculis,2), (purus,5), (eu,1), (imperdiet,1), (h...
```

Took 9 sec. Last updated by anonymous at February 17 2019, 4:25:47 PM.

```
%pyspark
from operator import add
linesDataset = spark.read.text("hdfs:///tmp/data/lorem_ipsum.txt").rdd.map(lambda r : r[0])
linesDataset.take(2)
counts=linesDataset.flatMap(lambda x : x.split(' ')).map(lambda x : (x,1)).reduceByKey(add)
counts1=counts.sortBy(lambda a : a[1],False)
output=counts1.collect()
for (word,count) in output :
    print word,' : ',count
```

```
vitae  :  9
in  :  8
vel  :  7
eget  :  6
ullamcorper  :  6
sit  :  6
Phasellus  :  5
amet  :  5
purus  :  5
ac  :  5
lacus  :  4
Morbi  :  4
sed  :  4
et  :  4
nec  :  4
Aliquam  :  4
libero  :  4
```



```
val trucksDf = spark.read.option("header","true").csv("hdfs:///tmp/data/trucks.csv")
```

```
trucksDf: org.apache.spark.sql.DataFrame = [driverid: string, truckid: string ... 109 more fields]
```

Took 1 sec. Last updated by anonymous at February 17 2019, 4:25:59 PM.

```
trucksDf.columns
```

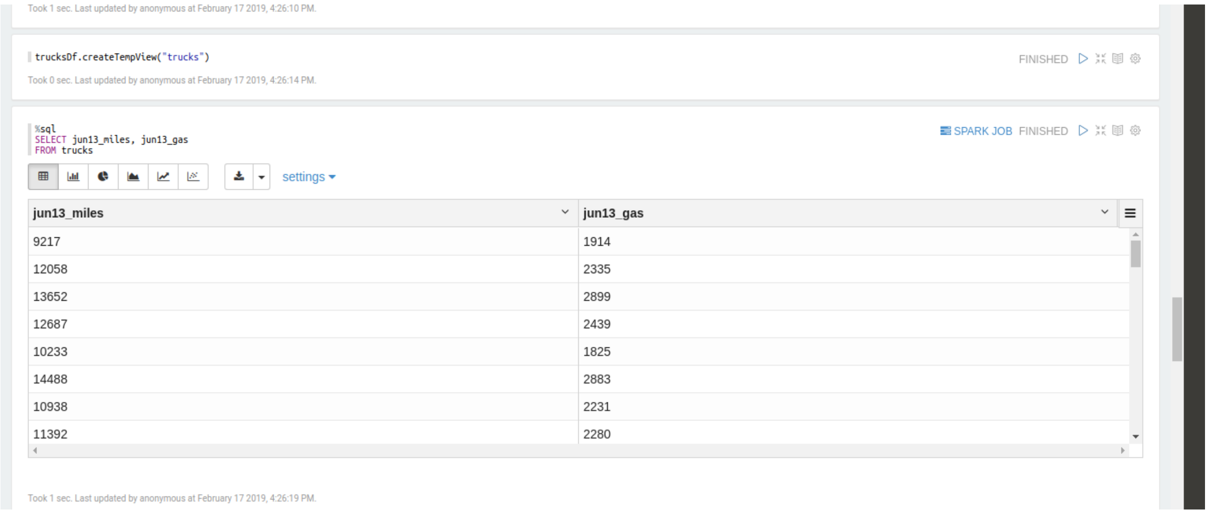
```
res9: Array[String] = Array(driverid, truckid, model, jun13_miles, jun13_gas, may13_miles, may13_gas, apr13_miles, apr13_gas, mar13_miles, mar13_gas, feb13_miles, feb13_gas, jan13_miles, jan13_gas, dec12_miles,
dec12_gas, nov12_miles, nov12_gas, oct12_miles, oct12_gas, sep12_miles, sep12_gas, aug12_miles, aug12_gas, jul12_miles, jul12_gas, jun12_miles, jun12_gas, may12_miles, may12_gas, apr12_miles, apr12_gas, mar12_m
iles, mar12_gas, feb12_miles, feb12_gas, jan12_miles, jan12_gas, dec11_miles, dec11_gas, nov11_miles, nov11_gas, oct11_miles, oct11_gas, sep11_miles, sep11_gas, aug11_miles, aug11_gas, jul11_miles, jul11_gas, j
un11_miles, jun11_gas, may11_miles, may11_gas, apr11_miles, apr11_gas, mar11_miles, mar11_gas, feb11_miles, feb11_gas, jan11_miles, jan11_gas, dec10_miles, dec10_gas,...
```

Took 0 sec. Last updated by anonymous at February 17 2019, 4:26:02 PM.

```
trucksDf.select("driverid", "jun13_miles").show(10)
```

```
+--------+-----------+
|driverid|jun13_miles|
+--------+-----------+
|      A1|       9217|
|      A2|      12058|
|      A3|      13652|
|      A4|      12687|
|      A5|      10233|
|      A6|      14488|
|      A7|      10938|
|      A8|      11392|
|      A9|      12601|
|     A10|      13699|
+--------+-----------+
only showing top 10 rows
```

Took 1 sec. Last updated by anonymous at February 17 2019, 4:26:10 PM.



Took 1 sec. Last updated by anonymous at February 17 2019, 4:26:10 PM.

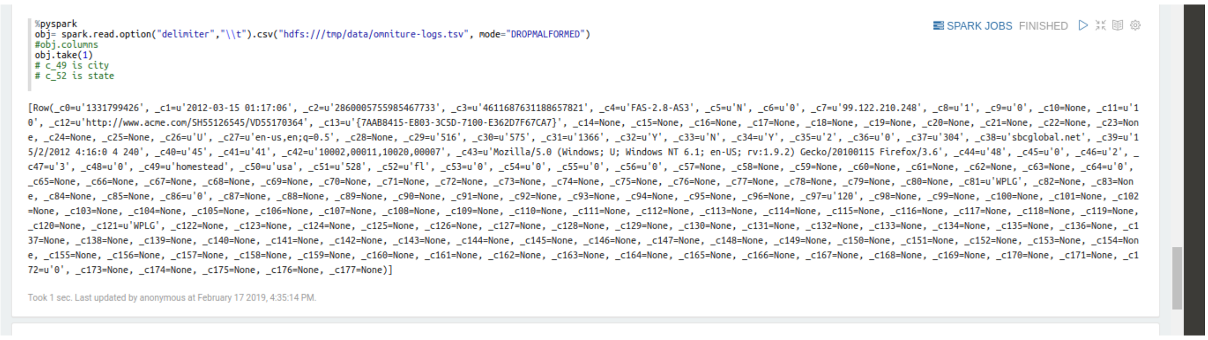
```
truckDf.createTempView("trucks")
```

Took 0 sec. Last updated by anonymous at February 17 2019, 4:26:14 PM.

```
%sql
SELECT jun13_miles, jun13_gas
FROM trucks
```

jun13_miles	jun13_gas
9217	1914
12058	2335
13652	2899
12687	2439
10233	1825
14488	2883
10938	2231
11392	2280

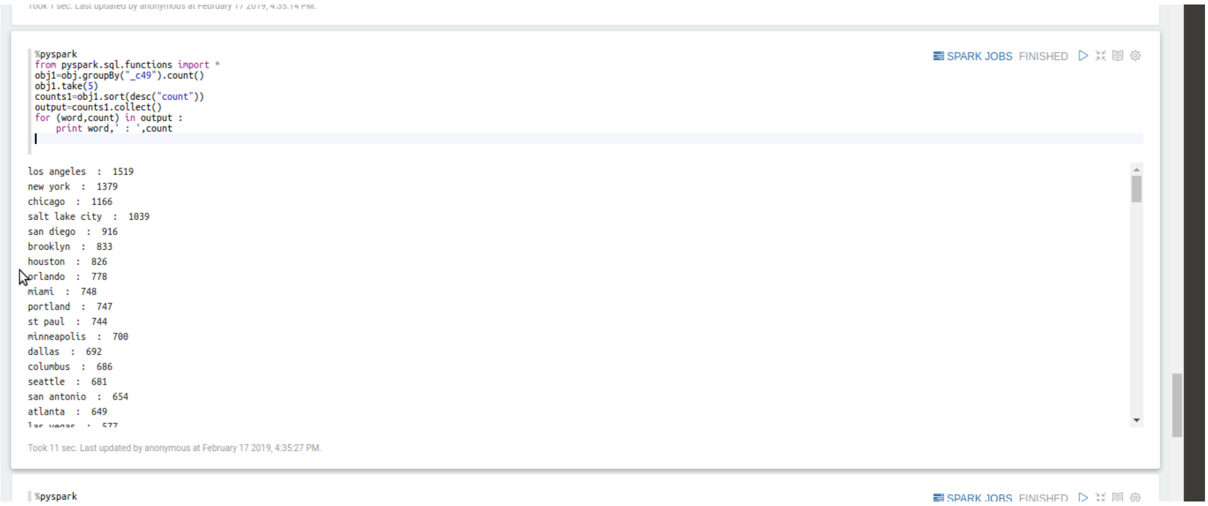
Took 1 sec. Last updated by anonymous at February 17 2019, 4:26:19 PM.



```
%pyspark
obj= spark.read.option("delimiter","\\t").csv("hdfs:///tmp/data/omniture-logs.tsv", mode="DROPMALFORMED")
#obj.columns
obj.take(1)
# c_49 is city
# c_52 is state
```

```
[Row(_c0=u'1331799426', _c1=u'2012-03-15 01:17:06', _c2=u'2860005755985467733', _c3=u'4611687631188657821', _c4=u'FAS-2.8-AS3', _c5=u'N', _c6=u'0', _c7=u'99.122.210.248', _c8=u'1', _c9=u'0', _c10=None, _c11=u'1
0', _c12=u'http://www.acme.com/SH55126545/VD55170364', _c13=u'{7AAB8415-E803-3C5D-7100-E362D7F67CA7}', _c14=None, _c15=None, _c16=None, _c17=None, _c18=None, _c19=None, _c20=None, _c21=None, _c22=None, _c23=Non
e, _c24=None, _c25=None, _c26=u'U', _c27=u'en-us,en;q=0.5', _c28=None, _c29=u'516', _c30=u'575', _c31=u'1366', _c32=u'Y', _c33=u'N', _c34=u'Y', _c35=u'2', _c36=u'0', _c37=u'304', _c38=u'sbcglobal.net', _c39=u'1
5/2/2012 4:16:0 4 240', _c40=u'45', _c41=u'41', _c42=u'10002,00011,10020,00007', _c43=u'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6', _c44=u'48', _c45=u'0', _c46=u'2', _
c47=u'3', _c48=u'0', _c49=u'homestead', _c50=u'usa', _c51=u'528', _c52=u'fl', _c53=u'0', _c54=u'0', _c55=u'0', _c56=u'0', _c57=None, _c58=None, _c59=None, _c60=None, _c61=None, _c62=None, _c63=None, _c64=u'0',
_c65=None, _c66=None, _c67=None, _c68=None, _c69=None, _c70=None, _c71=None, _c72=None, _c73=None, _c74=None, _c75=None, _c76=None, _c77=None, _c78=None, _c79=None, _c80=None, _c81=u'WPLG', _c82=None, _c83=Non
e, _c84=None, _c85=None, _c86=u'0', _c87=None, _c88=None, _c89=None, _c90=None, _c91=None, _c92=None, _c93=None, _c94=None, _c95=None, _c96=None, _c97=u'120', _c98=None, _c99=None, _c100=None, _c101=None, _c102
=None, _c103=None, _c104=None, _c105=None, _c106=None, _c107=None, _c108=None, _c109=None, _c110=None, _c111=None, _c112=None, _c113=None, _c114=None, _c115=None, _c116=None, _c117=None, _c118=None, _c119=None,
_c120=None, _c121=u'WPLG', _c122=None, _c123=None, _c124=None, _c125=None, _c126=None, _c127=None, _c128=None, _c129=None, _c130=None, _c131=None, _c132=None, _c133=None, _c134=None, _c135=None, _c136=None, _c1
37=None, _c138=None, _c139=None, _c140=None, _c141=None, _c142=None, _c143=None, _c144=None, _c145=None, _c146=None, _c147=None, _c148=None, _c149=None, _c150=None, _c151=None, _c152=None, _c153=None, _c154=Non
e, _c155=None, _c156=None, _c157=None, _c158=None, _c159=None, _c160=None, _c161=None, _c162=None, _c163=None, _c164=None, _c165=None, _c166=None, _c167=None, _c168=None, _c169=None, _c170=None, _c171=None, _c1
72=u'0', _c173=None, _c174=None, _c175=None, _c176=None, _c177=None)]
```

Took 1 sec. Last updated by anonymous at February 17 2019, 4:35:14 PM.



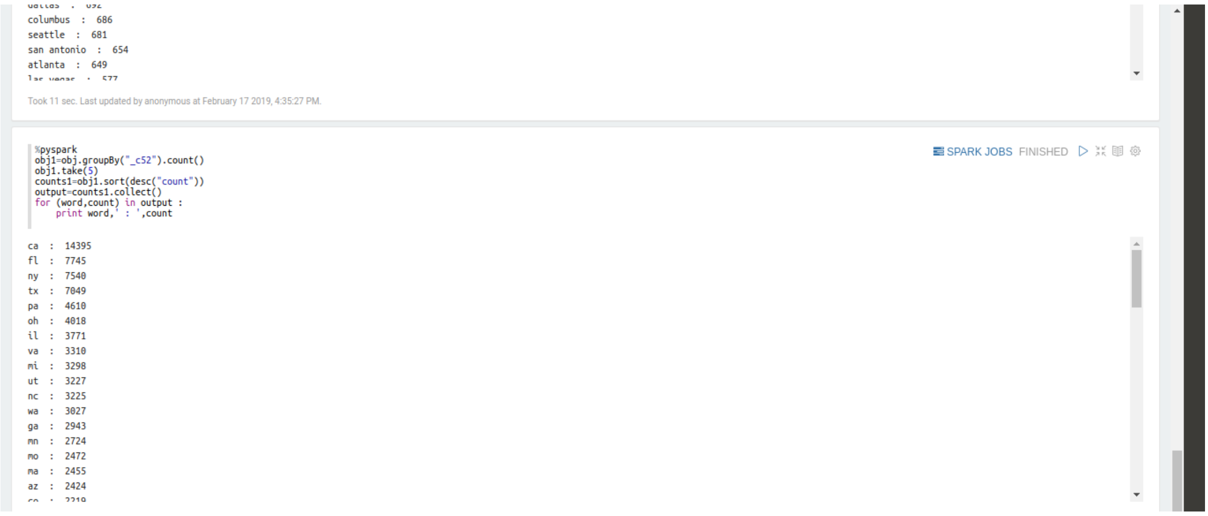
Took 1 sec. Last updated by anonymous at February 17 2019, 4:35:14 PM.

```
%pyspark
from pyspark.sql.functions import *
obj1=obj.groupBy("_c49").count()
obj1.take(5)
counts1=obj1.sort(desc("count"))
output=counts1.collect()
for (word,count) in output :
    print word,' : ',count
```

```
los angeles  :  1519
new york  :  1379
chicago  :  1166
salt lake city  :  1039
san diego  :  916
brooklyn  :  833
houston  :  826
orlando  :  778
miami  :  748
portland  :  747
st paul  :  744
minneapolis  :  700
dallas  :  692
columbus  :  686
seattle  :  681
san antonio  :  654
atlanta  :  649
las vegas  :  577
```

Took 11 sec. Last updated by anonymous at February 17 2019, 4:35:27 PM.

```
%pyspark
```



```
dallas  :  692
columbus  :  686
seattle  :  681
san antonio  :  654
atlanta  :  649
las vegas  :  577
```

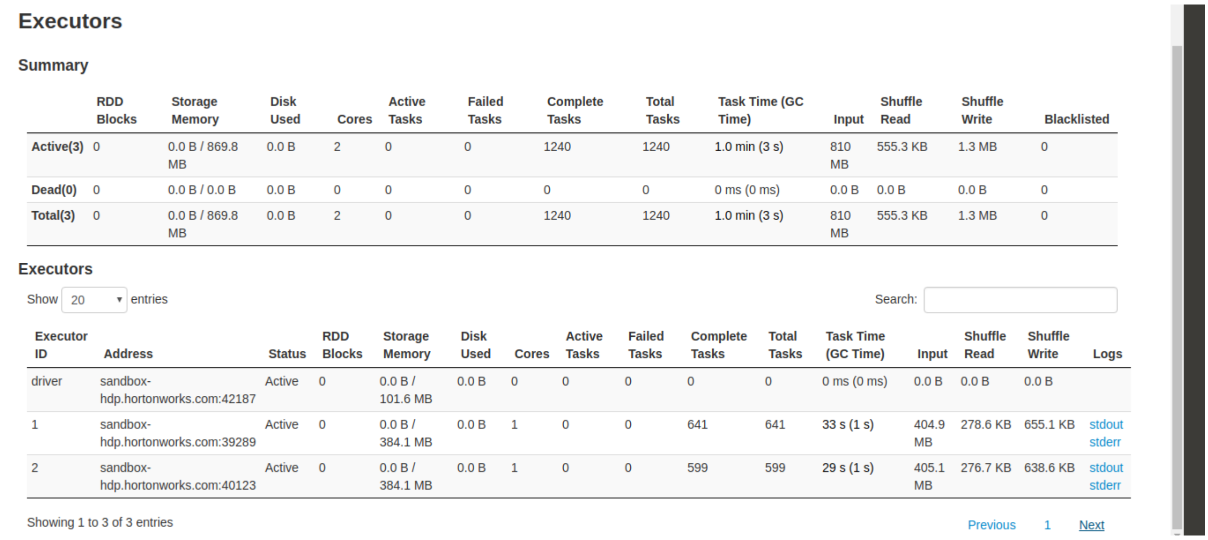
Took 11 sec. Last updated by anonymous at February 17 2019, 4:35:27 PM.

```
%pyspark
obj1=obj.groupBy("_c52").count()
obj1.take(5)
counts1=obj1.sort(desc("count"))
output=counts1.collect()
for (word,count) in output :
    print word,' : ',count
```

```
ca  :  14395
fl  :  7745
ny  :  7540
tx  :  7049
pa  :  4610
oh  :  4018
il  :  3771
va  :  3310
mi  :  3298
ut  :  3227
nc  :  3225
wa  :  3027
ga  :  2943
mn  :  2724
mo  :  2472
ma  :  2455
az  :  2424
co  :  2219
```

Answer 2

There are two active cores. The tasks are split asymmetrically between the two cores.



Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(3)	0	0.0 B / 869.8 MB	0.0 B	2	0	0	1240	1240	1.0 min (3 s)	810 MB	555.3 KB	1.3 MB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(3)	0	0.0 B / 869.8 MB	0.0 B	2	0	0	1240	1240	1.0 min (3 s)	810 MB	555.3 KB	1.3 MB	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	sandbox-hdp.hortonworks.com:42187	Active	0	0.0 B / 101.6 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	
1	sandbox-hdp.hortonworks.com:39289	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	641	641	33 s (1 s)	404.9 MB	278.6 KB	655.1 KB	stdout stderr
2	sandbox-hdp.hortonworks.com:40123	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	599	599	29 s (1 s)	405.1 MB	276.7 KB	638.6 KB	stdout stderr

Showing 1 to 3 of 3 entries