Pyspark

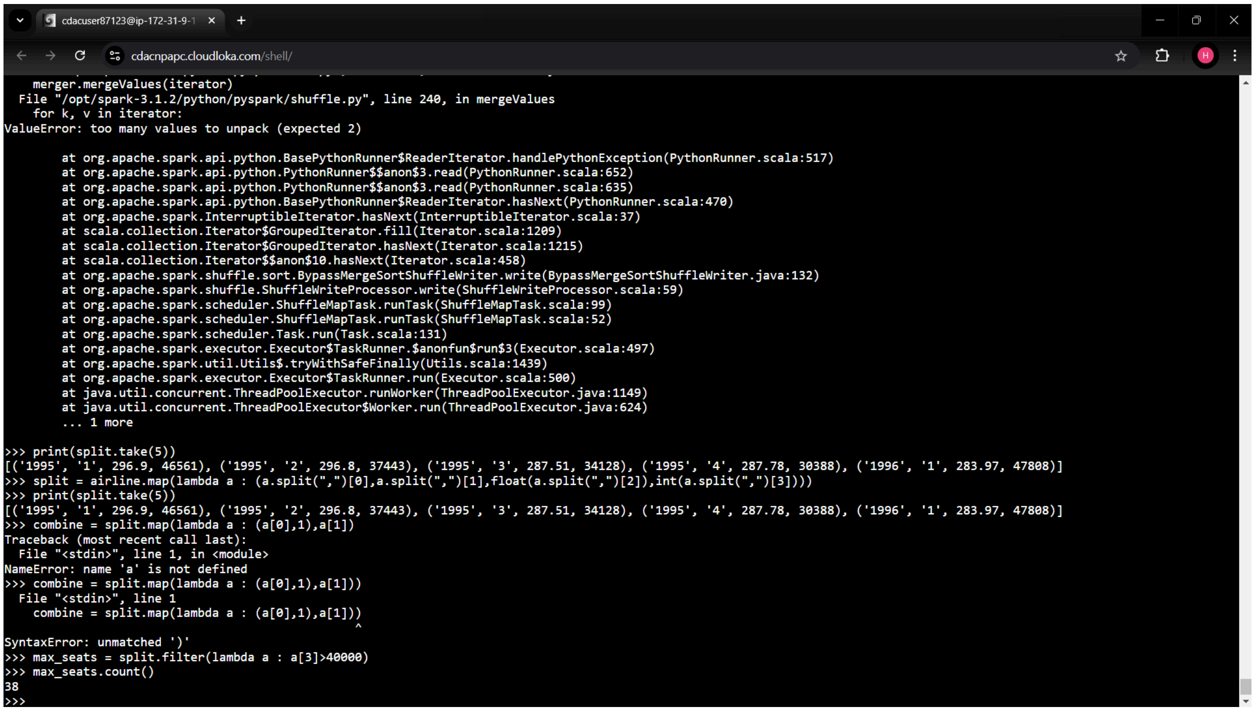
```
>>> airline =
sc.textFile("/user/cdacuser87123/airlines.csv")
>>> airline.count()
85
>>> header = airline.first()
>>> airline = airline.filter(lambda line:
line!=header)
>>> airline.count()
84
print(airline.take(5))
['1995,1,296.9,46561', '1995,2,296.8,37443',
'1995,3,287.51,34128', '1995,4,287.78,30388',
'1996,1,283.97,47808']
>>> split = airline.map(lambda a :
(a.split(",")[0],a.split(",")[1],float(a.split(",
")[2]),int(a.split(",")[3])))
>>> print(split.take(5))
[('1995', '1', 296.9, 46561), ('1995', '2',
296.8, 37443), ('1995', '3', 287.51, 34128),
('1995', '4', 287.78, 30388), ('1996', '1',
283.97, 47808)]
```

Q1

a

```
max_seats = split.filter(lambda a : a[3]>40000)
>>> max_seats.count()
```

Output



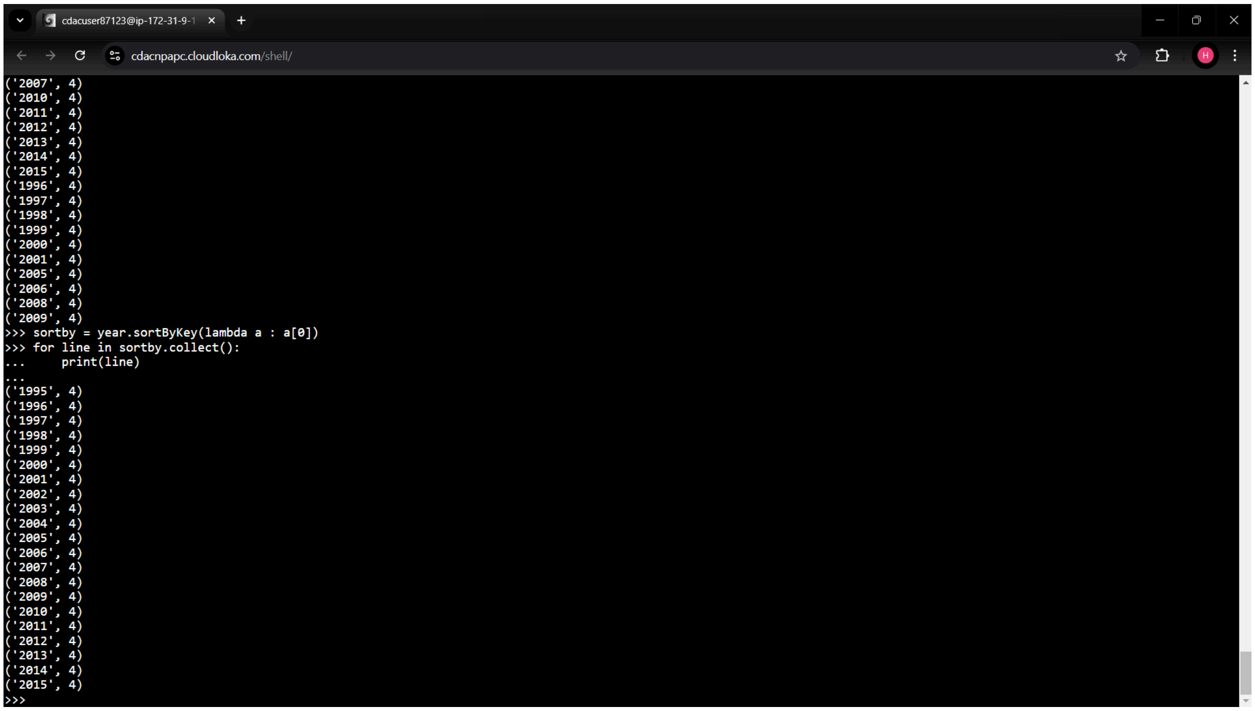
```
    merger.mergeValues(iterator)
  File "/opt/spark-3.1.2/python/pyspark/shuffle.py", line 240, in mergeValues
    for k, v in iterator:
ValueError: too many values to unpack (expected 2)

        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.handlePythonException(PythonRunner.scala:517)
        at org.apache.spark.api.python.PythonRunner$$anon$3.read(PythonRunner.scala:652)
        at org.apache.spark.api.python.PythonRunner$$anon$3.read(PythonRunner.scala:635)
        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.hasNext(PythonRunner.scala:470)
        at org.apache.spark.InterruptibleIterator.hasNext(InterruptibleIterator.scala:37)
        at scala.collection.Iterator$GroupedIterator.fill(Iterator.scala:1209)
        at scala.collection.Iterator$GroupedIterator.hasNext(Iterator.scala:1215)
        at scala.collection.Iterator$$anon$10.hasNext(Iterator.scala:458)
        at org.apache.spark.shuffle.sort.BypassMergeSortShuffleWriter.write(BypassMergeSortShuffleWriter.java:132)
        at org.apache.spark.shuffle.ShuffleWriteProcessor.write(ShuffleWriteProcessor.scala:59)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:99)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:52)
        at org.apache.spark.scheduler.Task.run(Task.scala:131)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Executor.scala:497)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> print(split.take(5))
[('1995', '1', 296.9, 46561), ('1995', '2', 296.8, 37443), ('1995', '3', 287.51, 34128), ('1995', '4', 287.78, 30388), ('1996', '1', 283.97, 47808)]
>>> split = airline.map(lambda a : (a.split(",")[0],a.split(",")[1],float(a.split(",")[2]),int(a.split(",")[3])))
>>> print(split.take(5))
[('1995', '1', 296.9, 46561), ('1995', '2', 296.8, 37443), ('1995', '3', 287.51, 34128), ('1995', '4', 287.78, 30388), ('1996', '1', 283.97, 47808)]
>>> combine = split.map(lambda a : (a[0],1),a[1])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>> combine = split.map(lambda a : (a[0],1),a[1]))
  File "<stdin>", line 1
    combine = split.map(lambda a : (a[0],1),a[1]))
                                                 ^
SyntaxError: unmatched ')'
>>> max_seats = split.filter(lambda a : a[3]>40000)
>>> max_seats.count()
38
>>>
```

b

```
>>> combine = split.map(lambda a : (a[0],1))
>>> year = combine.reduceByKey(lambda a,b: a+b)
```

```
sortby = year.sortByKey(lambda a : a[0])
>>> for line in sortby.collect():
...       print(line)
```

Output



```
('2007', 4)
('2010', 4)
('2011', 4)
('2012', 4)
('2013', 4)
('2014', 4)
('2015', 4)
('1996', 4)
('1997', 4)
('1998', 4)
('1999', 4)
('2000', 4)
('2001', 4)
('2005', 4)
('2006', 4)
('2008', 4)
('2009', 4)
>>> sortby = year.sortByKey(lambda a : a[0])
>>> for line in sortby.collect():
...     print(line)
...
('1995', 4)
('1996', 4)
('1997', 4)
('1998', 4)
('1999', 4)
('2000', 4)
('2001', 4)
('2002', 4)
('2003', 4)
('2004', 4)
('2005', 4)
('2006', 4)
('2007', 4)
('2008', 4)
('2009', 4)
('2010', 4)
('2011', 4)
('2012', 4)
('2013', 4)
('2014', 4)
('2015', 4)
>>>
```

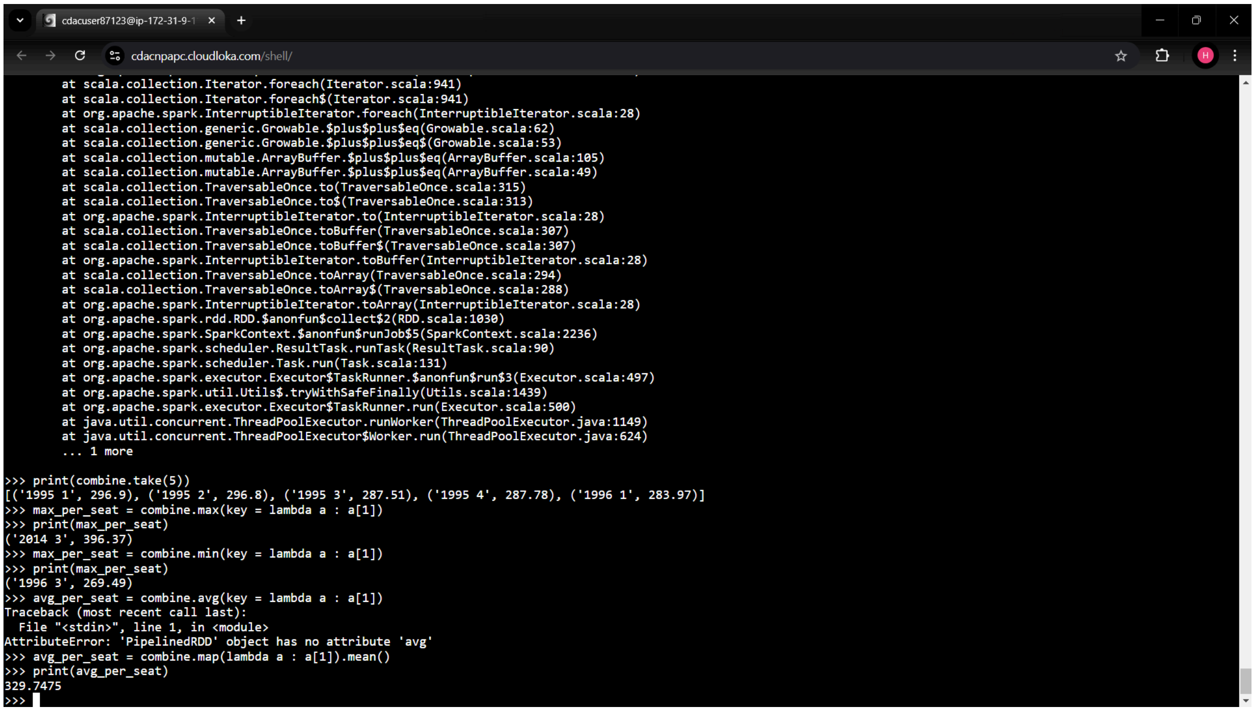
Q2

a)

```
combine = split.map(lambda a : (a[0],a[2]))
```

```
>>> max_per_seat = combine.max(key = lambda a : a[1])
max_per_seat = combine.min(key = lambda a : a[1])
>>> print(max_per_seat)
avg_per_seat = combine.map(lambda a : a[1]).mean()
>>> print(avg_per_seat)
```

Output:

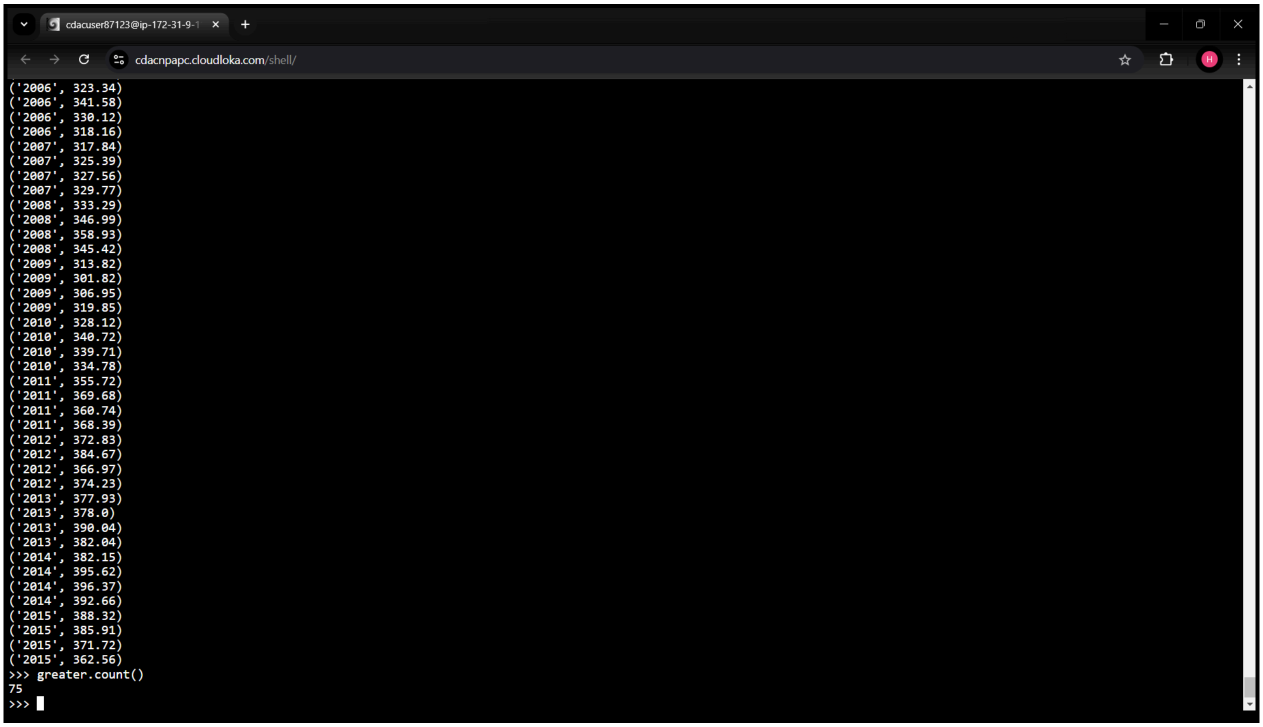


```
        at scala.collection.Iterator.foreach(Iterator.scala:941)
        at scala.collection.Iterator.foreach$(Iterator.scala:941)
        at org.apache.spark.InterruptibleIterator.foreach(InterruptibleIterator.scala:28)
        at scala.collection.generic.Growable.$plus$plus$eq(Growable.scala:62)
        at scala.collection.generic.Growable.$plus$plus$eq$(Growable.scala:53)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:105)
        at scala.collection.mutable.ArrayBuffer.$plus$plus$eq(ArrayBuffer.scala:49)
        at scala.collection.TraversableOnce.to(TraversableOnce.scala:315)
        at scala.collection.TraversableOnce.to$(TraversableOnce.scala:313)
        at org.apache.spark.InterruptibleIterator.to(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toBuffer(TraversableOnce.scala:307)
        at scala.collection.TraversableOnce.toBuffer$(TraversableOnce.scala:307)
        at org.apache.spark.InterruptibleIterator.toBuffer(InterruptibleIterator.scala:28)
        at scala.collection.TraversableOnce.toArray(TraversableOnce.scala:294)
        at scala.collection.TraversableOnce.toArray$(TraversableOnce.scala:288)
        at org.apache.spark.InterruptibleIterator.toArray(InterruptibleIterator.scala:28)
        at org.apache.spark.rdd.RDD.$anonfun$collect$2(RDD.scala:1030)
        at org.apache.spark.SparkContext.$anonfun$runJob$5(SparkContext.scala:2236)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:131)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Executor.scala:497)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1439)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:500)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        ... 1 more

>>> print(combine.take(5))
[('1995 1', 296.9), ('1995 2', 296.8), ('1995 3', 287.51), ('1995 4', 287.78), ('1996 1', 283.97)]
>>> max_per_seat = combine.max(key = lambda a : a[1])
>>> print(max_per_seat)
('2014 3', 396.37)
>>> max_per_seat = combine.min(key = lambda a : a[1])
>>> print(max_per_seat)
('1996 3', 269.49)
>>> avg_per_seat = combine.avg(key = lambda a : a[1])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'PipelinedRDD' object has no attribute 'avg'
>>> avg_per_seat = combine.map(lambda a : a[1]).mean()
>>> print(avg_per_seat)
329.7475
>>>
```

b)

```
combine = split.map(lambda a : (a[0],a[2]))
```

```
greater = combine.filter(lambda a : (a[1]>290))
greater.count()
```

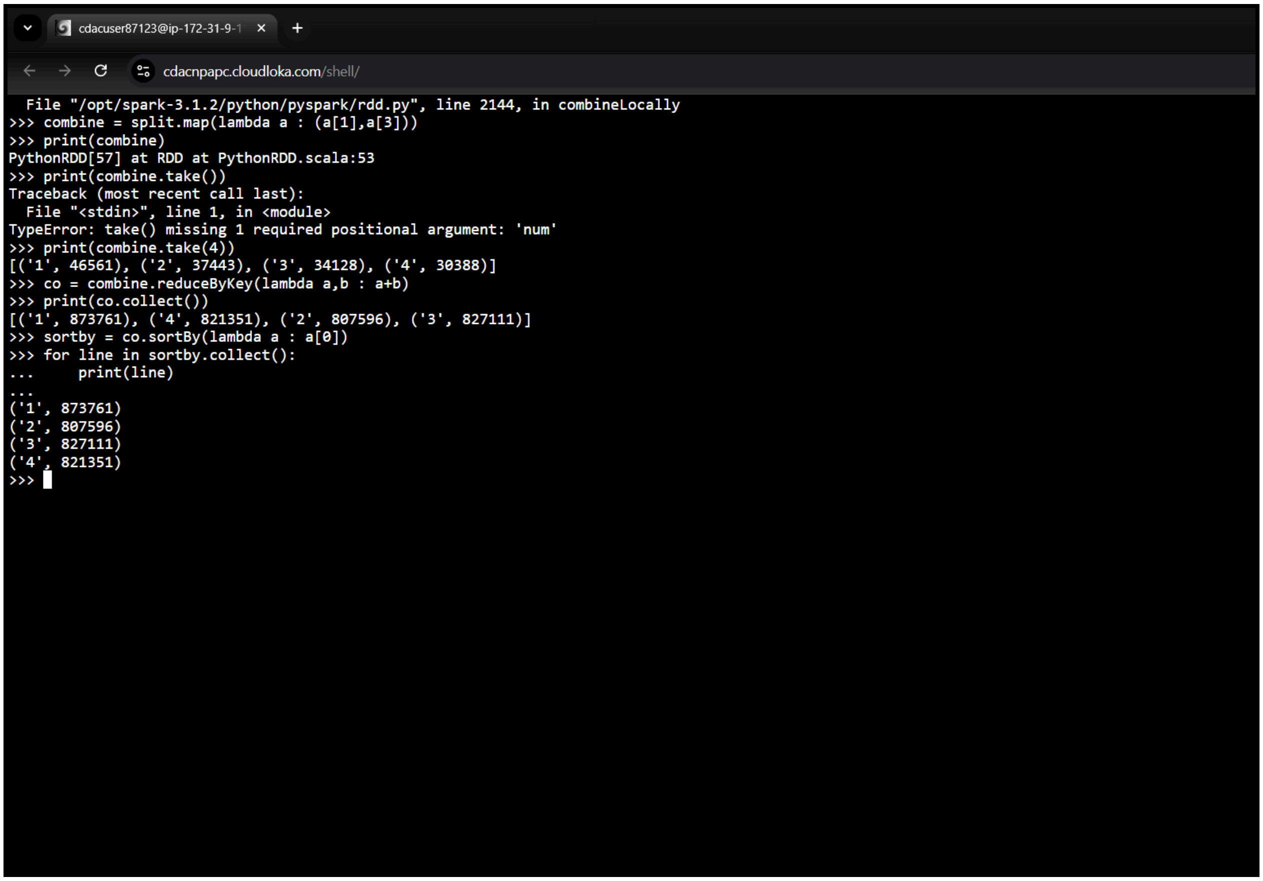


```
('2006', 323.34)
('2006', 341.58)
('2006', 330.12)
('2006', 318.16)
('2007', 317.84)
('2007', 325.39)
('2007', 327.56)
('2007', 329.77)
('2008', 333.29)
('2008', 346.99)
('2008', 358.93)
('2008', 345.42)
('2009', 313.82)
('2009', 301.82)
('2009', 306.95)
('2009', 319.85)
('2010', 328.12)
('2010', 340.72)
('2010', 339.71)
('2010', 334.78)
('2011', 355.72)
('2011', 369.68)
('2011', 360.74)
('2011', 368.39)
('2012', 372.83)
('2012', 384.67)
('2012', 366.97)
('2012', 374.23)
('2013', 377.93)
('2013', 378.0)
('2013', 390.04)
('2013', 382.04)
('2014', 382.15)
('2014', 395.62)
('2014', 396.37)
('2014', 392.66)
('2015', 388.32)
('2015', 385.91)
('2015', 371.72)
('2015', 362.56)
>>> greater.count()
75
>>> 
```

c)

```
combine = split.map(lambda a : (a[1],a[3]))
>>> print(combine)
PythonRDD[57] at RDD at PythonRDD.scala:53
>>> print(combine.take())
print(combine.take(4))
```

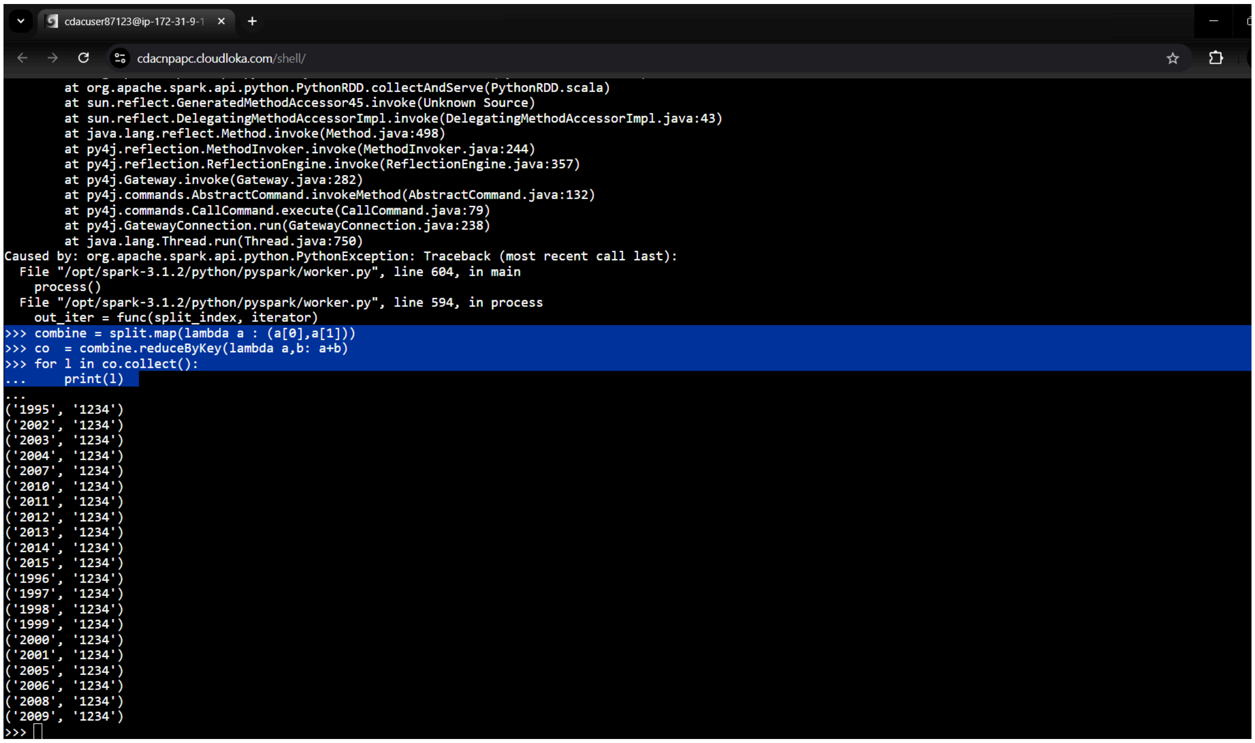
```
[('1', 46561), ('2', 37443), ('3', 34128), ('4',
30388)]
>>> co = combine.reduceByKey(lambda a,b : a+b)
>>> print(co.collect())
[('1', 873761), ('4', 821351), ('2', 807596),
('3', 827111)]
>>> sortby = co.sortBy(lambda a : a[0])
>>> for line in sortby.collect():
...        print(line)
```



```
  File "/opt/spark-3.1.2/python/pyspark/rdd.py", line 2144, in combineLocally
>>> combine = split.map(lambda a : (a[1],a[3]))
>>> print(combine)
PythonRDD[57] at RDD at PythonRDD.scala:53
>>> print(combine.take())
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: take() missing 1 required positional argument: 'num'
>>> print(combine.take(4))
[('1', 46561), ('2', 37443), ('3', 34128), ('4', 30388)]
>>> co = combine.reduceByKey(lambda a,b : a+b)
>>> print(co.collect())
[('1', 873761), ('4', 821351), ('2', 807596), ('3', 827111)]
>>> sortby = co.sortBy(lambda a : a[0])
>>> for line in sortby.collect():
...     print(line)
...
('1', 873761)
('2', 807596)
('3', 827111)
('4', 821351)
>>>
```

d)

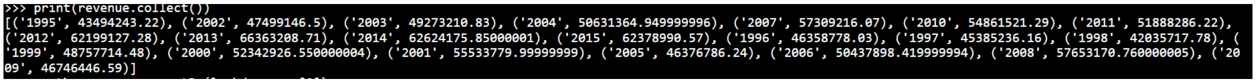
```
>>> combine = split.map(lambda a : (a[0],a[1]))
>>> co  = combine.reduceByKey(lambda a,b: a+b)
>>> for l in co.collect():
...      print(l)
```



```
        at org.apache.spark.api.python.PythonRDD.collectAndServe(PythonRDD.scala)
        at sun.reflect.GeneratedMethodAccessor45.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:750)
Caused by: org.apache.spark.api.python.PythonException: Traceback (most recent call last):
  File "/opt/spark-3.1.2/python/pyspark/worker.py", line 604, in main
    process()
  File "/opt/spark-3.1.2/python/pyspark/worker.py", line 594, in process
    out_iter = func(split_index, iterator)
>>> combine = split.map(lambda a : (a[0],a[1]))
>>> co  = combine.reduceByKey(lambda a,b: a+b)
>>> for l in co.collect():
...     print(l)
...
('1995', '1234')
('2002', '1234')
('2003', '1234')
('2004', '1234')
('2007', '1234')
('2010', '1234')
('2011', '1234')
('2012', '1234')
('2013', '1234')
('2014', '1234')
('2015', '1234')
('1996', '1234')
('1997', '1234')
('1998', '1234')
('1999', '1234')
('2000', '1234')
('2001', '1234')
('2005', '1234')
('2006', '1234')
('2008', '1234')
('2009', '1234')
>>>
```

e)

```
>>> combine = split(lambda a :
(a[0],(a[2]*a[3])))
Traceback (most recent call last):
 File "<stdin>", line 1, in <module>
TypeError: 'PipelinedRDD' object is not callable
>>> combine = split.map(lambda a :
(a[0],(a[2]*a[3])))
>>> print(combine.take(5))
[('1995', 13823960.899999999), ('1995',
11113082.4), ('1995', 9812141.28), ('1995',
8745058.639999999), ('1996', 13576037.760000002)]
>>> revenue = combine,reduceByKey(lambda a,b :
a+b)
Traceback (most recent call last):
 File "<stdin>", line 1, in <module>
NameError: name 'reduceByKey' is not defined
>>> revenue = combine.reduceByKey(lambda a,b :
a+b)
>>> print(revenue.collect())
```



```
>>> print(revenue.collect())
[('1995', 43494243.22), ('2002', 47499146.5), ('2003', 49273210.83), ('2004', 50631364.949999996), ('2007', 57309216.07), ('2010', 54861521.29), ('2011', 51888286.22),
('2012', 62199127.28), ('2013', 66363208.71), ('2014', 62624175.85000001), ('2015', 62378990.57), ('1996', 46358778.03), ('1997', 45385236.16), ('1998', 42035717.78), (
'1999', 48757714.48), ('2000', 52342926.550000004), ('2001', 55533779.99999999), ('2005', 46376786.24), ('2006', 50437898.419999994), ('2008', 57653170.760000005), ('20
09', 46746446.59)]
```

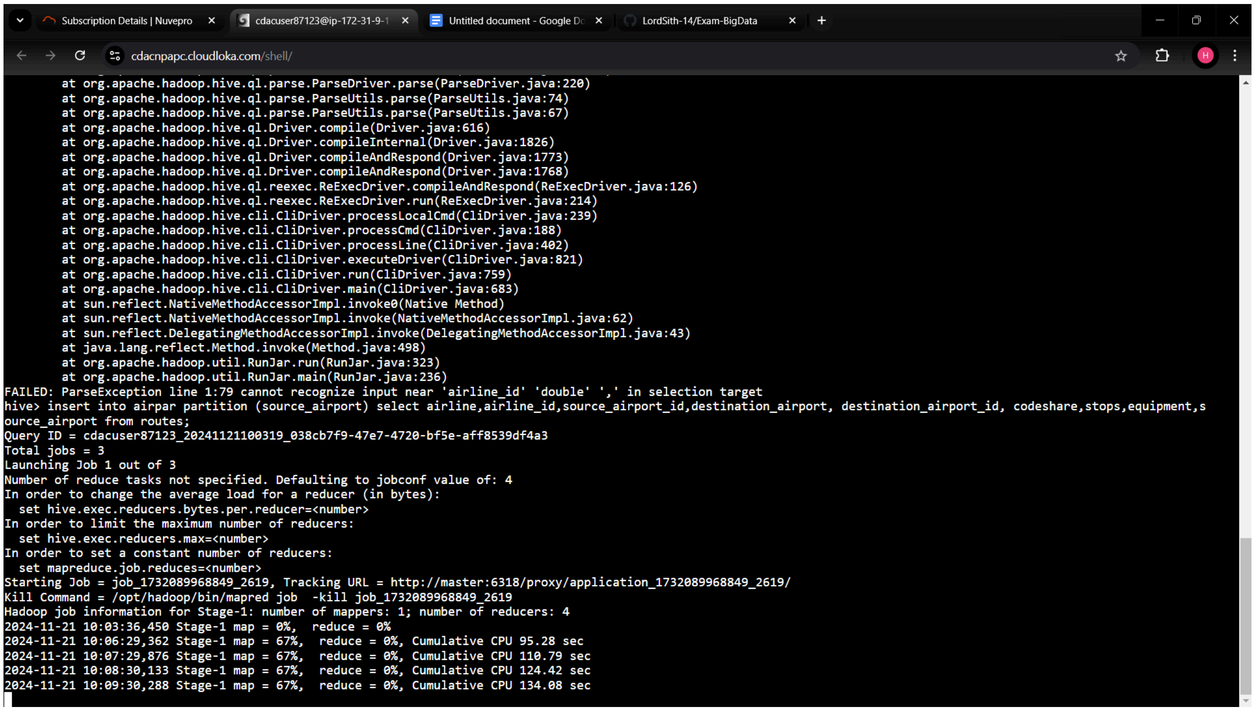
Hive

Q2)

A

create table airpar(airline string,airline_id double,source_airport_id double,destination_airport string, destination_airport_id double, codeshare string,stops in t,equipment) partitioned by (source_airport string) row format delimited fields terminated by ',' stored as textFile;

insert into airpar partition(source_airport) select airline ,airline_id ,source_airport_id ,destination_airport , destination_airport_id , codeshare,stops,equipment from routes


Subscription Details | Nuvepro
cdacuser87123@ip-172-31-9-1
Untitled document - Google Do
LordSith-14/Exam-BigData
cdacnpapc.cloudloka.com/shell/

```
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:220)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:74)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:67)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:616)
        at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1826)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1773)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1768)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.compileAndRespond(ReExecDriver.java:126)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.run(ReExecDriver.java:214)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:239)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:188)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:402)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:323)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:236)
FAILED: ParseException line 1:79 cannot recognize input near 'airline_id' 'double' ',' in selection target
hive> insert into airpar partition (source_airport) select airline,airline_id,source_airport_id,destination_airport, destination_airport_id, codeshare,stops,equipment,s
ource_airport from routes;
Query ID = cdacuser87123_20241121100319_038cb7f9-47e7-4720-bf5e-aff8539df4a3
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Defaulting to jobconf value of: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1732089968849_2619, Tracking URL = http://master:6318/proxy/application_1732089968849_2619/
Kill Command = /opt/hadoop/bin/mapred job  -kill job_1732089968849_2619
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 4
2024-11-21 10:03:36,450 Stage-1 map = 0%,  reduce = 0%
2024-11-21 10:06:29,362 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 95.28 sec
2024-11-21 10:07:29,876 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 110.79 sec
2024-11-21 10:08:30,133 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 124.42 sec
2024-11-21 10:09:30,288 Stage-1 map = 67%,  reduce = 0%, Cumulative CPU 134.08 sec
```

b)

```
insert overwrite airpar partition(source_airport) select airline
,airline_id ,source_airport_id ,destination_airport ,
destination_airport_id , codeshare,stops,equipment from routes
where source_airport = "JFK"
```

c)

```
select source_airport ,destination_airport from airpar  where
source_airport = "LAX"
```