Q.1

JAVA CODE:

```
import java.io.*;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;


public class AllTimeHigh {

      public static class MapClass extends
Mapper<LongWritable,Text,Text,DoubleWritable>
         {
             private Text stock_id = new Text();
             private DoubleWritable High = new DoubleWritable();

            public void map(LongWritable key, Text value, Context context)
            {

               try{
                  String[] str = value.toString().split(",");
                  double high = Double.parseDouble(str[4]);
                  stock_id.set(str[1]);
                  High.set(high);

                  //context.write(new Text(str[1]),new LongWritable(vol));
                  context.write(stock_id, High);
               }
               catch(Exception e)
               {
                  System.out.println(e.getMessage());
               }
            }
         }

        public static class ReduceClass extends
Reducer<Text,DoubleWritable,Text,DoubleWritable>
         {
                private DoubleWritable result = new DoubleWritable();

                public void reduce(Text key, Iterable<DoubleWritable>
values,Context context) throws IOException, InterruptedException {
                        double maxValue=0;
```

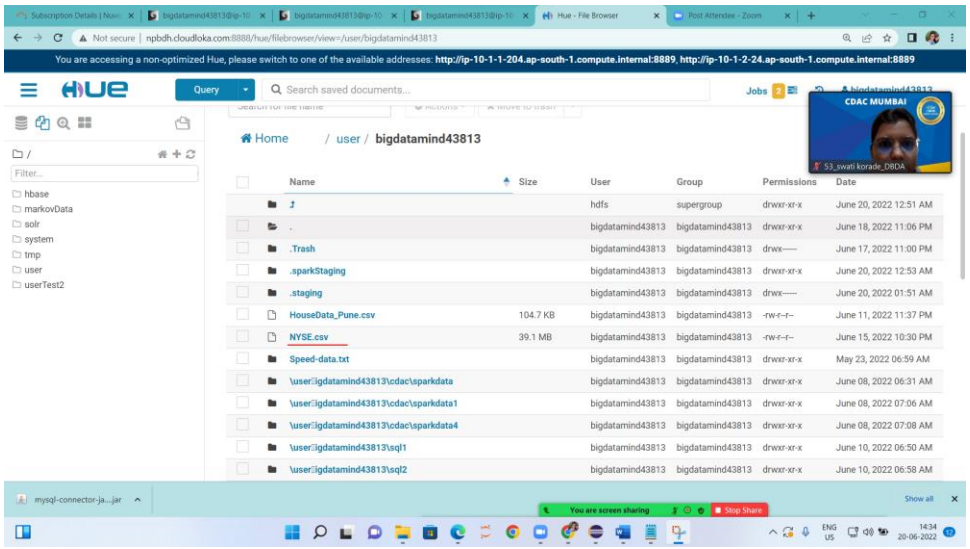
```
                        double temp_val=0;

                        for (DoubleWritable value : values) {
                              temp_val = value.get();
                              if (temp_val > maxValue) {
                                    maxValue = temp_val;
                              }
                        }
                        result.set(maxValue);

                  context.write(key, result);
                  //context.write(key, new LongWritable(sum));

                }
          }
        public static void main(String[] args) throws Exception {
                Configuration conf = new Configuration();
                //conf.set("name", "value")
                //conf.set("mapreduce.input.fileinputformat.split.minsize",
"134217728");
                Job job = Job.getInstance(conf, "Highest Price for each
stock");
                job.setJarByClass(AllTimeHigh.class);
                job.setMapperClass(MapClass.class);
                //job.setCombinerClass(ReduceClass.class);
                job.setReducerClass(ReduceClass.class);
                job.setNumReduceTasks(1);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(DoubleWritable.class);
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                System.exit(job.waitForCompletion(true) ? 0 : 1);
              }
}
```

Upload jar and file on FTP

MAP-REDUCE

FTP

```
[bigdatamind43813@ip-10-1-1-204 ~]$hadoop fs -put NYSE.csv
```

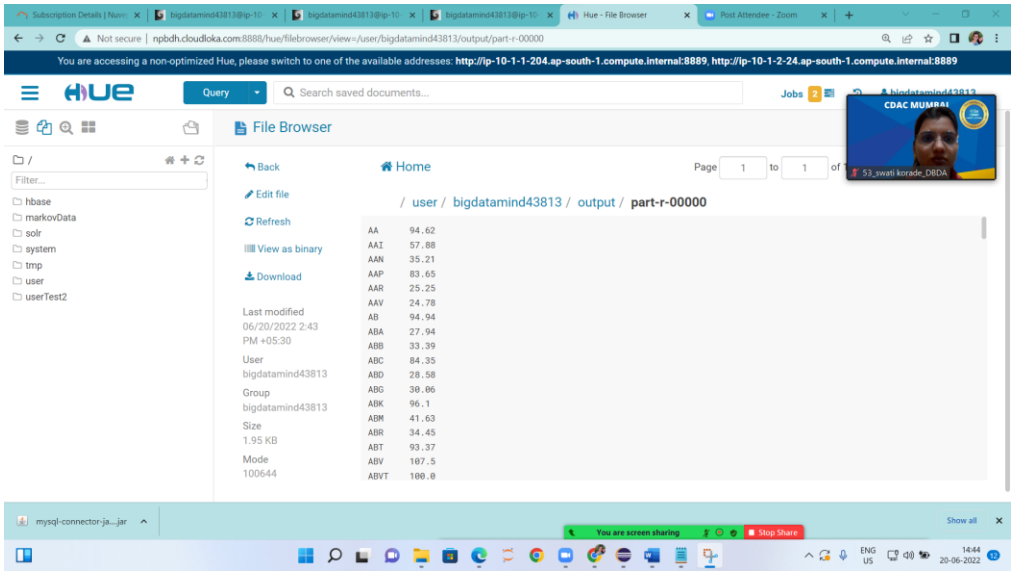


To check jar in hadoop

```
[bigdatamind43813@ip-10-1-1-204 ~]$ jar tvf myjar.jar
```

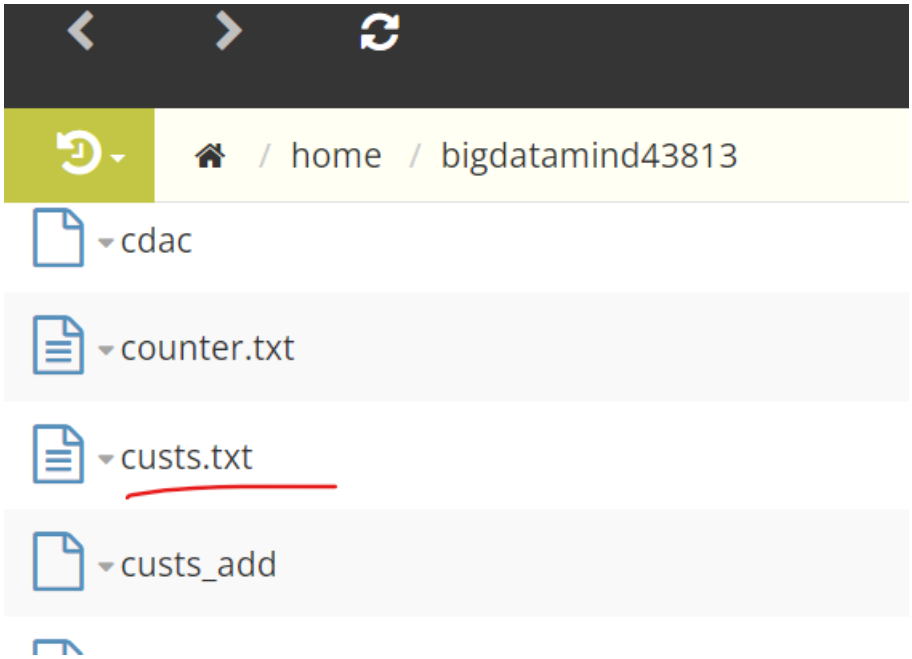
```
[bigdatamind43813@ip-10-1-1-204 ~]$ hadoop jar myjar.jar AllTimeHigh cdac/NYSE.csv output
WARNING: Use "yarn jar" to launch YARN applications.
22/06/20 09:13:04 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1
22/06/20 09:13:05 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the To
lication with ToolRunner to remedy this.
22/06/20 09:13:05 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/bigdatamind43813/.staging
22/06/20 09:13:05 INFO input.FileInputFormat: Total input files to process : 1
22/06/20 09:13:05 INFO mapreduce.JobSubmitter: number of splits:1
22/06/20 09:13:05 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. I
ublisher.enabled
22/06/20 09:13:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1654490426372_5616
22/06/20 09:13:05 INFO mapreduce.JobSubmitter: Executing with tokens: []
22/06/20 09:13:05 INFO conf.Configuration: resource-types.xml not found
22/06/20 09:13:05 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/06/20 09:13:05 INFO impl.YarnClientImpl: Submitted application application_1654490426372_5616
22/06/20 09:13:05 INFO mapreduce.Job: The url to track the job: http://ip-10-1-1-204.ap-south-1.compute.internal:6066/pr
6/
```

OUTPUT created on HUE



2.HIVE

1.Put data on FTP:



2. PUT data on hadoop:

hadoop fs -put cust.txt

3. create table,load data

```
hive> create table cust_table(custid INT,firstname STRING,lastname STRING,age INT,profession STRING)
    >
    > row format delimited
    >
    > fields terminated by ','
    >
    > stored as textfile;
OK
Time taken: 0.718 seconds
hive> load data local inpath 'custs.txt' overwrite into table cust_table;
Loading data to table default.cust_table
OK
Time taken: 1.398 seconds
```

4. QUERY

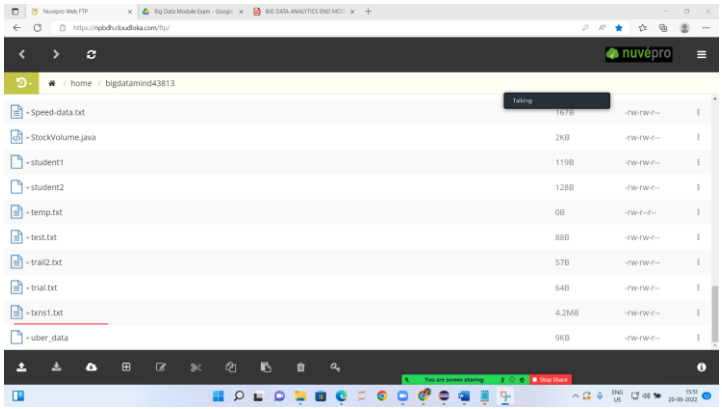
```
hive> select profession ,count(*) as count_prof from cust_table group by profession order by count_prof;
Query ID = bigdatamind43813_20220620085035_9d7f4eac-8a00-4ddf-a6e2-826a9cf08811
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
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>
22/06/20 08:50:36 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/06/20 08:50:37 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1654490426372_5558, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1654490426372_5558/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1654490426372_5558
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-06-20 08:50:49,958 Stage-1 map = 0%,  reduce = 0%
2022-06-20 08:51:00,546 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.79 sec
2022-06-20 08:51:06,735 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.29 sec
MapReduce Total cumulative CPU time: 5 seconds 290 msec
Ended Job = job_1654490426372_5558
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
```

OUTPUT

```
Total MapReduce CPU Time Spent: 9 seconds 440 msec
OK
Social Worker   1
Writer  101
Artist  175
Environmental scientist 176
Carpenter       181
Dancer  185
Therapist       187
Economist       189
Real estate agent       191
Electrical engineer     192
Nurse   192
Civil engineer  193
Automotive mechanic     193
Psychologist    194
Electrician     194
Agricultural and food scientist 195
Athlete 196
Statistician    196
Judge   196
Doctor  197
Financial analyst       198
Accountant      199
Reporter        200
Secretary       200
Coach   201
Physicist       201
Farmer  201
Actor   202
Architect       203
Computer hardware engineer      204
Teacher 204
```

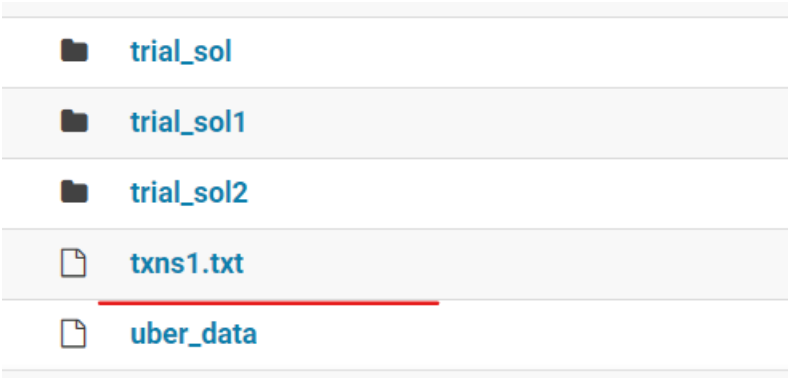
Please find Sales data sets

Put file on FTP



To put file on hadoop

hadoop fs -put txns1.txt




```
Time taken: 0.029 seconds
)ive> reate table trans_table1(txnno INT,txndate STRING,custno INT,amount DOUBLE, category STRING,product STRING,city STRING,state STRING,spendby STRING
    >
    > row format delimited
    >
    > fields terminated by ','
    >
    > stored as textfile;
OK
Time taken: 0.093 seconds
hive> load data local inpath 'txns1.txt' overwrite into table trans_table1;
Loading data to table training053.trans_table1
OK
Time taken: 0.683 seconds
hive> select product,sum(amount) as sale from trans_table1 group by product ordre by sale desc limit 10;
FAILED: ParseException line 1:70 missing EOF at 'ordre' near 'product'
hive> select product,sum(amount) as sale from trans_table1 group by product order by sale desc limit 10;
Query ID = bigdatamind43813_20220620101544_062cf380-21ae-43fb-a33b-6349bdb7e35b
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
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>
```

Table is created on hive warehouse:

	test1_hive	bigdata
	tester_hive	bigdata
	testing	bigdata
	training053__customer_prof_index__	bigdata
	trans_table1	bigdata
	trial_header	bigdata
	trial_part	bigdata
	trial_table	bigdata
	trial_table_header	bigdata
	txn_bucket	bigdata
	txnrecsbycat	bigdata
	txnrecsbycat2	bigdata

2) Write a program to find the top 10 products sales wise

```
Total MapReduce CPU Time Spent: 11 seconds 800 msec
OK
Yoga & Pilates  47804.93999999993
Swing Sets      47204.13999999999
Lawn Games      46828.44
Golf    46577.67999999999
Cardio Machine Accessories      46485.540000000045
Exercise Balls  45143.84
Weightlifting Belts     45111.67999999996
Mahjong 44995.19999999999
Basketball      44954.68000000004
Beach Volleyball        44890.67000000005
Time taken: 102.521 seconds, Fetched: 10 row(s)
```

3) Write a program to create partioned table on category.

TABLE CRERATION WITH PARTITION ,LOAD DATA,

```
hive> create table txnByCat(txnno INT, txndate STRING, custno INT, amount DOUBLE,
    >
    > product STRING, city STRING, state STRING, spendby STRING)
    >
    > partitioned by (category STRING)
    >
    > row format delimited
    >
    > fields terminated by ','
    >
    > stored as textfile;
OK
Time taken: 0.08 seconds
hive> et hive.exec.dynamic.partition.mode=nonstrict;
hive>
    > set hive.exec.dynamic.partition=true;
hive>
    >
    >
    > set hive.enforce.bucketing=true;
hive> INSERT OVERWRITE TABLE txnByCat PARTITION(category) select txn.txnno, txn.txndate,txn.custno, txn.amount,txn.product,txn.city,txn.state, txn.spendb
y, txn.category from trans_table1 txn DISTRIBUTE By category;
Query ID = bigdatamind43813_20220620104054_a4f4f9f1-4e74-484b-8831-dac7bd5ec044
Total jobs = 1
Launching Job 1 out of 1
```

PARTION TABLE IN VIEW

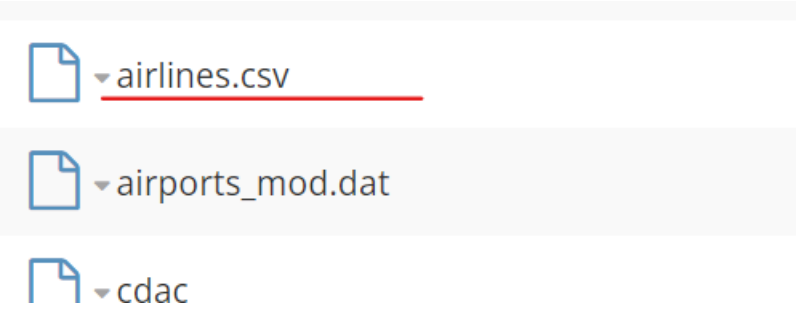
Home / user / hive / warehouse / training053.db / **txnbycat** Trash

	Name	Size	User	Group	Permissions	Date
	↑		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:39 AM
	.		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Air Sports		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Combat Sports		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Dancing		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Exercise & Fitness		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Games		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Gymnastics		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Indoor Games		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Jumping		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Outdoor Play Equipment		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Outdoor Recreation		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM
	category=Puzzles		bigdatamind43813	hive	drwxrwxrwxt	June 20, 2022 03:42 AM

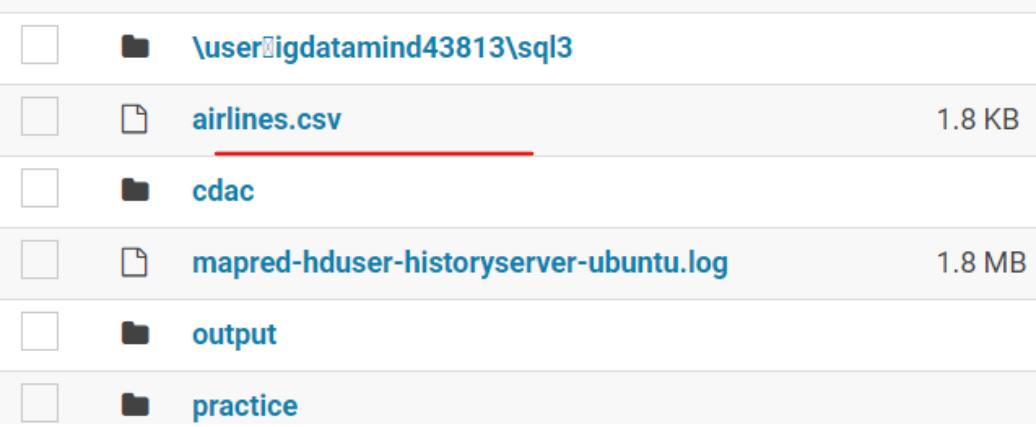
Q.3

.PYSPARK

PUT ON FTP



hadoop fs -put airlines.csv



RDD creation

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.2.1
      /_/

Using Python version 2.7.5 (default, Nov 16 2020 22:23:17)
SparkSession available as 'spark'.
>>> RDD1=sc.textFile("/user/bigdatamind43813/airlines.csv")
>>> RDD2=RDD1.map(lambda a: a.encode)("ascii","ignore"))
  File "<stdin>", line 1
    RDD2=RDD1.map(lambda a: a.encode)("ascii","ignore"))
                                                       ^
SyntaxError: invalid syntax
>>> RDD2=RDD1.map(lambda a: a.encode("ascii","ignore"))
>>> head=RDD1.first()
>>> head=RDD2.first()
>>> RDD3=RDD2.filter(lambda a:a!=head)
>>> for i in RDD3.take(5):
...     print(i)
...
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
>>>
```

```
>>> RDD4=RDD3.map(lambda a:a.split(','))
>>> for i in RDD4.take(5):
...     print(i)
...
['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']
>>>
```

1) What was the highest number of people travelled in which year?

```
>>> RDD5=RDD4.map(lambda a: (a[0],int(a[3])))
>>> hightravl=RDD5.reduceByKey(lambda a,b :a+b)
>>> highsort=hightravl.sortBy(lambda a: -a[1])
>>> for i in highsort.take(1):
...     print(i)
...
('2007', 176299)
```

2) Identifying the highest revenue generation for which year

```
>>> revenueRDD=RDD4.map(lambda a : (a[0],float(a[2])*int(a[3])))
>>> revenueRDD1=revenueRDD.reduceByKey(lambda a,b : a+b)
>>> revenuesort=revenueRDD1.sortBy(lambda a:-a[1])
>>> for i in revenuesort.take(1):
...     print(i)
...
('2013', 66363208.71)
```

3) Identifying the highest revenue generation for which year and quarter (Common group)

```
>>> quaterRDD=RDD4.map(lambda a:(a[0]+" "+a[1],float(a[2]) *int(a[3])))
>>> quaterRDD1=quarterRDD.reduceByKey(lambda a,b:a+b)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'quarterRDD' is not defined
>>> quaterRDD1=quaterRDD.reduceByKey(lambda a,b:a+b)
>>> sortquarter=quaterRDD1.sortBy(lambda a:-a[1])
>>> for i in sortquarter.take(1):
...     print(i)
...
('2014 4', 18819408.48)
>>>
```