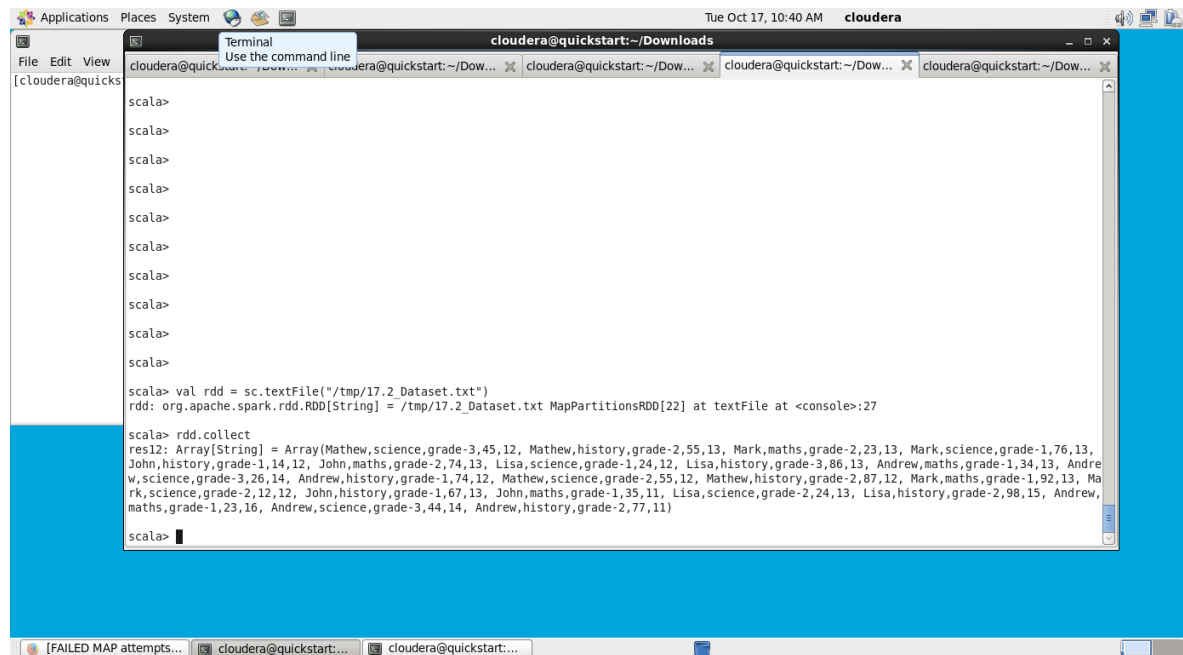


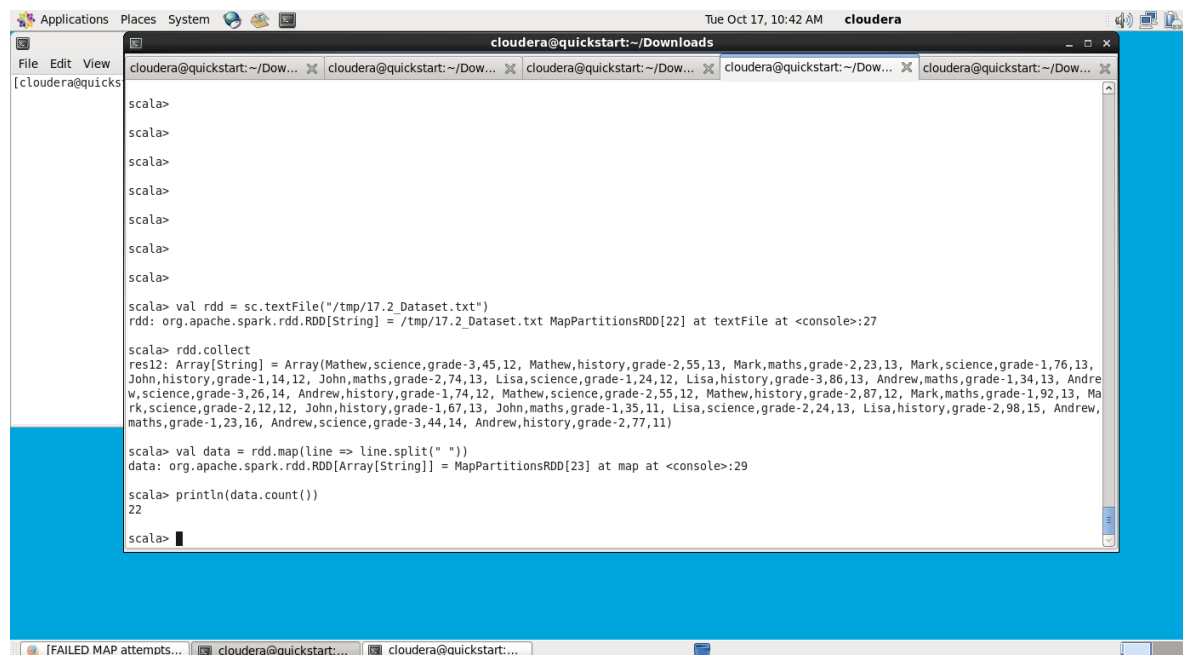
## Problem Statement 1:

1. Read the text file, and create a tupled rdd.



```
cloudera@quickstart:~/Downloads
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[22] at textFile at <console>:27
scala> rdd.collect
res12: Array[String] = Array(Mathew,science,grade-3,45,12, Mathew,history,grade-2,55,13, Mark,maths,grade-2,23,13, Mark,science,grade-1,76,13,
John,history,grade-1,14,12, John,maths,grade-2,74,13, Lisa,science,grade-1,24,12, Lisa,history,grade-3,86,13, Andrew,maths,grade-1,34,13, Andre
w,science,grade-3,26,14, Andrew,history,grade-1,74,12, Mathew,science,grade-2,55,12, Mathew,history,grade-2,87,12, Mark,maths,grade-1,92,13, Ma
rk,science,grade-2,12,12, John,history,grade-1,67,13, John,maths,grade-1,35,11, Lisa,science,grade-2,24,13, Lisa,history,grade-2,98,15, Andrew,
maths,grade-1,23,16, Andrew,science,grade-3,44,14, Andrew,history,grade-2,77,11)
scala>
```

2. Find the count of total number of rows present.



```
cloudera@quickstart:~/Downloads
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[22] at textFile at <console>:27
scala> rdd.collect
res12: Array[String] = Array(Mathew,science,grade-3,45,12, Mathew,history,grade-2,55,13, Mark,maths,grade-2,23,13, Mark,science,grade-1,76,13,
John,history,grade-1,14,12, John,maths,grade-2,74,13, Lisa,science,grade-1,24,12, Lisa,history,grade-3,86,13, Andrew,maths,grade-1,34,13, Andre
w,science,grade-3,26,14, Andrew,history,grade-1,74,12, Mathew,science,grade-2,55,12, Mathew,history,grade-2,87,12, Mark,maths,grade-1,92,13, Ma
rk,science,grade-2,12,12, John,history,grade-1,67,13, John,maths,grade-1,35,11, Lisa,science,grade-2,24,13, Lisa,history,grade-2,98,15, Andrew,
maths,grade-1,23,16, Andrew,science,grade-3,44,14, Andrew,history,grade-2,77,11)
scala> val data = rdd.map(line => line.split(" "))
data: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[23] at map at <console>:29
scala> println(data.count())
22
scala>
```

3. What is the distinct number of subjects present in the entire school



```
cloudera@quickstart:~/Downloads
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[32] at textFile at <console>:27
scala> val rows = rdd.map(line => line.split(",")).map(line => (line(0).toString, line(2).toString))
rows: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[34] at map at <console>:29
scala> rows.countByValue()
res17: scala.collection.Map[(String, String),Long] = Map((Mark,grade-1) -> 2, (Andrew,grade-1) -> 3, (Lisa,grade-2) -> 2, (John,grade-1) -> 3, (Lisa,grade-3) -> 1, (Mathew,grade-3) -> 1, (Andrew,grade-2) -> 1, (Andrew,grade-3) -> 2, (Lisa,grade-1) -> 1, (Mathew,grade-2) -> 3, (John,grade-2) -> 1, (Mark,grade-2) -> 2)
scala>
```

2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

```
cloudera@quickstart:~/Downloads
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[39] at textFile at <console>:27
scala> val rows = rdd.map(line => line.split(",")).map(line => (line(0).toString, line(2).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[41] at map at <console>:29
scala> val groups = rows.groupBy(x => (x._1,x._2))
groups: org.apache.spark.rdd.RDD[(String, String), Iterable[(String, String, Int)]] = ShuffledRDD[43] at groupBy at <console>:31
scala> val avg = groups.mapValues(iter => iter.map(_._3).map(iter => iter._2.sum/iter._2.size))
avg: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[45] at map at <console>:33
scala> avg.collect
res18: Array[Int] = Array(24, 17, 61, 45, 77, 43, 86, 38, 74, 84, 35, 65)
scala>
```

3. What is the average score of students in each subject across all grades?

```
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[87] at textFile at <console>:27

scala> val rows = rdd.map(line => line.split(",")).map(line => (line(1).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[89] at map at <console>:29

scala> rows.collect()
res144: Array[(String, Int)] = Array((science,45), (history,55), (maths,23), (science,76), (history,14), (maths,74), (science,24), (history,86), (maths,34), (science,26), (history,74), (science,55), (history,87), (maths,92), (science,12), (history,67), (maths,35), (science,24), (history,98), (maths,23), (science,44), (history,77))

scala> val groups = rows.groupBy(x => (x._1))
groups: org.apache.spark.rdd.RDD[(String, Iterable[(String, Int)])] = ShuffledRDD[91] at groupBy at <console>:31

scala> groups.collect()
res145: Array[(String, Iterable[(String, Int)])] = Array((maths,CompactBuffer((maths,23), (maths,74), (maths,34), (maths,92), (maths,35), (maths,23))), (history,CompactBuffer((history,55), (history,14), (history,86), (history,74), (history,87), (history,67), (history,98), (history,77))), (science,CompactBuffer((science,45), (science,76), (science,24), (science,26), (science,55), (science,12), (science,24), (science,44))))

scala> val avg = groups.mapValues(iter => iter.map(_._2).map(iter => iter._1.sum/iter._1.size))
avg: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[93] at map at <console>:33

scala> avg.collect()
res146: Array[Int] = Array(108, 112, 104)

scala>
```

4. What is the average score of students in each subject per grade?

```
scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[95] at textFile at <console>:27

scala> val rows = rdd.map(line => line.split(",")).map(line => (line(1).toString, line(2).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[97] at map at <console>:29

scala> rows.collect()
res147: Array[(String, String, Int)] = Array((science,grade-3,45), (history,grade-2,55), (maths,grade-2,23), (science,grade-1,76), (history,grade-1,14), (maths,grade-2,74), (science,grade-1,24), (history,grade-3,86), (maths,grade-1,34), (science,grade-3,26), (history,grade-1,74), (science,grade-2,55), (history,grade-2,87), (maths,grade-1,92), (science,grade-2,12), (history,grade-1,67), (maths,grade-1,35), (science,grade-2,24), (history,grade-2,98), (maths,grade-1,23), (science,grade-3,44), (history,grade-2,77))

scala> val groups = rows.groupBy(x => (x._1,x._2))
groups: org.apache.spark.rdd.RDD[(String, String), Iterable[(String, String, Int)]] = ShuffledRDD[99] at groupBy at <console>:31

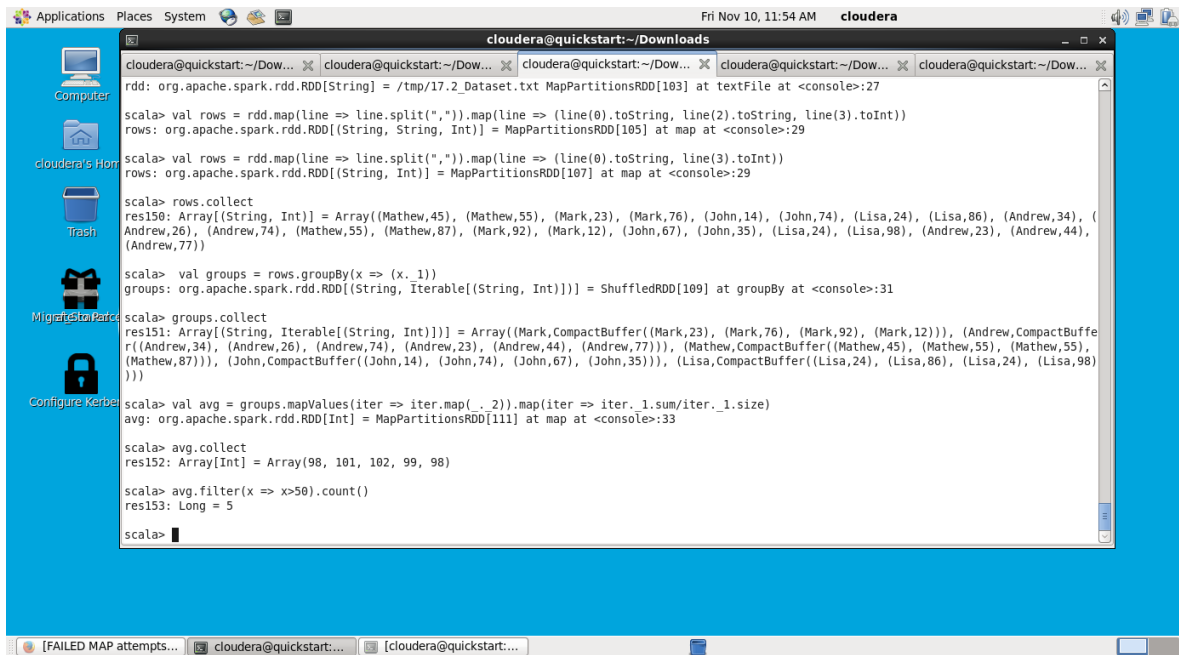
scala> groups.collect()
res148: Array[(String, String), Iterable[(String, String, Int)]] = Array((history,grade-2,CompactBuffer((history,grade-2,55), (history,grade-2,87), (history,grade-2,98), (history,grade-2,77))), ((history,grade-3,CompactBuffer((history,grade-3,86))), ((maths,grade-1,CompactBuffer((maths,grade-1,34), (maths,grade-1,92), (maths,grade-1,35), (maths,grade-1,23))), ((science,grade-3,CompactBuffer((science,grade-3,45), (science,grade-3,26), (science,grade-3,44))), ((science,grade-1,CompactBuffer((science,grade-1,76), (science,grade-1,24))), ((science,grade-2,CompactBuffer((science,grade-2,55), (science,grade-2,12), (science,grade-2,24))), ((history,grade-1,CompactBuffer((history,grade-1,14), (history,grade-1,74), (history,grade-1,67))), ((maths,grade-2,CompactBuffer((maths,grade-2,23), (maths,grade-2,74))))

scala> val avg = groups.mapValues(iter => iter.map(_._3).map(iter => iter._2.sum/iter._2.size))
avg: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[101] at map at <console>:33

scala> avg.collect()
res149: Array[Int] = Array(79, 86, 46, 38, 50, 30, 51, 48)

scala>
```

5. For all students in grade-2, how many have average score greater than 50?



```
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads

rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[103] at textFile at <console>:27

scala> val rows = rdd.map(line => line.split(",")).map(line => (line(0).toString, line(2).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[105] at map at <console>:29

scala> val rows = rdd.map(line => line.split(",")).map(line => (line(0).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[107] at map at <console>:29

scala> rows.collect
res150: Array[(String, Int)] = Array((Mathew,45), (Mathew,55), (Mark,23), (Mark,76), (John,14), (John,74), (Lisa,24), (Lisa,86), (Andrew,34), (Andrew,26), (Andrew,74), (Mathew,55), (Mathew,87), (Mark,92), (Mark,12), (John,67), (John,35), (Lisa,24), (Lisa,98), (Andrew,23), (Andrew,44), (Andrew,77))

scala> val groups = rows.groupBy(x => (x._1))
groups: org.apache.spark.rdd.RDD[(String, Iterable[(String, Int)])] = ShuffledRDD[109] at groupBy at <console>:31

scala> groups.collect
res151: Array[(String, Iterable[(String, Int)])] = Array((Mark,CompactBuffer((Mark,23), (Mark,76), (Mark,92), (Mark,12)))), (Andrew,CompactBuffer((Andrew,34), (Andrew,26), (Andrew,74), (Andrew,23), (Andrew,44), (Andrew,77)))), (Mathew,CompactBuffer((Mathew,45), (Mathew,55), (Mathew,55), (Mathew,87))), (John,CompactBuffer((John,14), (John,74), (John,67), (John,35))), (Lisa,CompactBuffer((Lisa,24), (Lisa,86), (Lisa,24), (Lisa,98))))

scala> val avg = groups.mapValues(iter => iter.map(_._2).map(iter => iter._1.sum/iter._1.size))
avg: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[111] at map at <console>:33

scala> avg.collect
res152: Array[Int] = Array(98, 101, 102, 99, 98)

scala> avg.filter(x => x>50).count()
res153: Long = 5

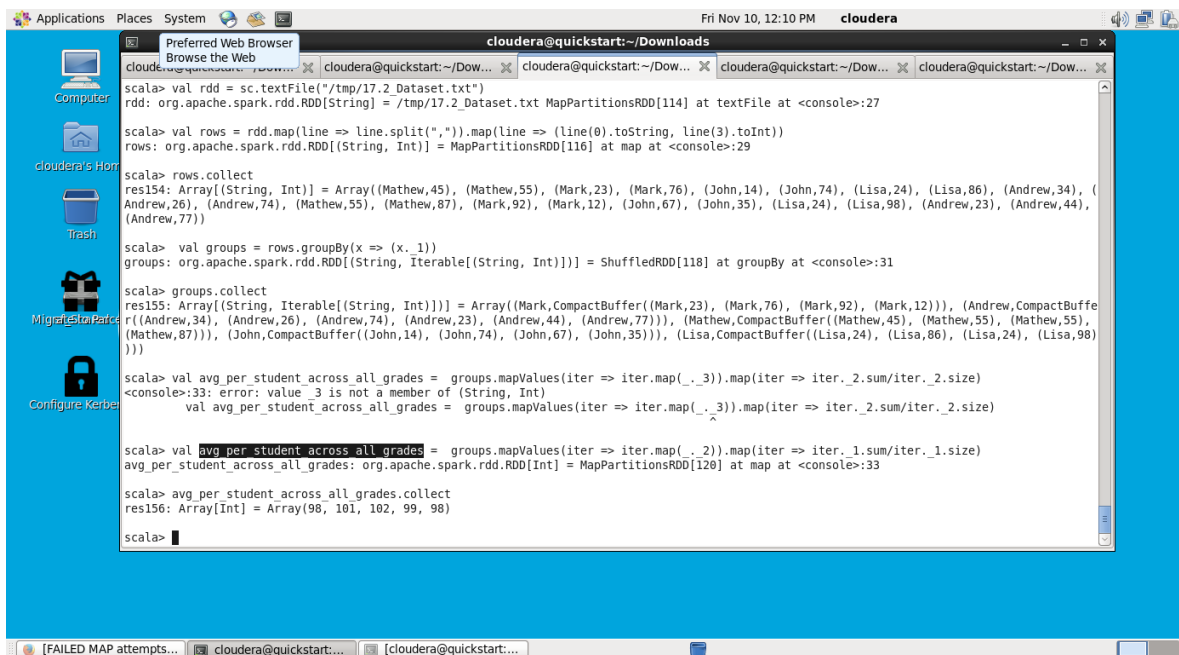
scala>
```

### Problem Statement 3:

Are there any students in the college that satisfy the below criteria :

1. Average score per student\_name across all grades is same as average score per student\_name per grade

Hint - Use Intersection Property.



```
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads
cloudera@quickstart:~/Downloads

scala> val rdd = sc.textFile("/tmp/17.2_Dataset.txt")
rdd: org.apache.spark.rdd.RDD[String] = /tmp/17.2_Dataset.txt MapPartitionsRDD[114] at textFile at <console>:27

scala> val rows = rdd.map(line => line.split(",")).map(line => (line(0).toString, line(3).toInt))
rows: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[116] at map at <console>:29

scala> rows.collect
res154: Array[(String, Int)] = Array((Mathew,45), (Mathew,55), (Mark,23), (Mark,76), (John,14), (John,74), (Lisa,24), (Lisa,86), (Andrew,34), (Andrew,26), (Andrew,74), (Mathew,55), (Mathew,87), (Mark,92), (Mark,12), (John,67), (John,35), (Lisa,24), (Lisa,98), (Andrew,23), (Andrew,44), (Andrew,77))

scala> val groups = rows.groupBy(x => (x._1))
groups: org.apache.spark.rdd.RDD[(String, Iterable[(String, Int)])] = ShuffledRDD[118] at groupBy at <console>:31

scala> groups.collect
res155: Array[(String, Iterable[(String, Int)])] = Array((Mark,CompactBuffer((Mark,23), (Mark,76), (Mark,92), (Mark,12)))), (Andrew,CompactBuffer((Andrew,34), (Andrew,26), (Andrew,74), (Andrew,23), (Andrew,44), (Andrew,77)))), (Mathew,CompactBuffer((Mathew,45), (Mathew,55), (Mathew,55), (Mathew,87))), (John,CompactBuffer((John,14), (John,74), (John,67), (John,35))), (Lisa,CompactBuffer((Lisa,24), (Lisa,86), (Lisa,24), (Lisa,98))))

scala> val avg_per_student_across_all_grades = groups.mapValues(iter => iter.map(_._2).map(iter => iter._2.sum/iter._2.size))
<console>:33: error: value _3 is not a member of (String, Int)
val avg_per_student_across_all_grades = groups.mapValues(iter => iter.map(_._3).map(iter => iter._2.sum/iter._2.size))

scala> val avg_per_student_across_all_grades = groups.mapValues(iter => iter.map(_._2).map(iter => iter._1.sum/iter._1.size))
avg_per_student_across_all_grades: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[120] at map at <console>:33

scala> avg_per_student_across_all_grades.collect
res156: Array[Int] = Array(98, 101, 102, 99, 98)

scala>
```

