Given a dataset of college students as a text file (name, subject, grade, marks) :

Problem Statement 1:

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

   val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

   val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))

   tupleRDD.collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[15] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[17] at map at <console>:26

scala> tupleRDD.collect
res10: Array[(String, String, String, Int, Int)] = 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), (Andrew,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), (Mark,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.

   val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

   val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))

   tupleRDD.count

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[19] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[21] at map at <console>:26

scala> tupleRDD.count
res12: Long = 22
```

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

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))

val subjectRDD = tupleRDD.map(x=>x._2)

subjectRDD.distinct.collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[41] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[43] at map at <console>:26

scala> val subjectRDD = tupleRDD.map(x=>x._2)
subjectRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[44] at map at <console>:28

scala> subjectRDD.distinct.collect
res23: Array[String] = Array(maths, history, science)
```

4. What is the count of the number of students in the school, whose name is Mathew and marks is 55.

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))

tupleRDD.map(x=>(x._1,x._4)).filter(x=>(x._1=="Mathew" && x._2==55)).count

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[64] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt,array(4).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, String, Int, Int)] = MapPartitionsRDD[66] at map at <console>:26

scala>

scala> tupleRDD.map(x=>(x._1,x._4)).filter(x=>(x._1=="Mathew" && x._2==55)).count
res34: Long = 2

scala>
```

Problem Statement 2:

1. What is the count of students per grade in the school?

   val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

   val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(2),1))

   tupleRDD.reduceByKey((x,y)=>(x+y)).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[87] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(2),1))
tupleRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[89] at map at <console>:26

scala> tupleRDD.reduceByKey((x,y)=>(x+y)).collect
res38: Array[(String, Int)] = Array((grade-3,4), (grade-1,9), (grade-2,9))

scala>
```

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

   val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

   val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0), array(2),array(3).toInt))

   val mapStudent_Grade = tupleRDD.map(x=>((x._1,x._2),1)).reduceByKey((x,y)=>(x+y))

   val mapStudent_Marks = tupleRDD.map(x=>((x._1,x._2),x._3))

   mapStudent_Marks.join(mapStudent_Grade).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[203] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>( array(0), array(2),array(3).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[205] at map at <console>:26

scala> val mapStudent_Grade = tupleRDD.map(x=>((x._1,x._2),1)).reduceByKey((x,y)=>(x+y))
mapStudent_Grade: org.apache.spark.rdd.RDD[((String, String), Int)] = ShuffledRDD[207] at reduceByKey at <console>:28

scala> val mapStudent_Marks = tupleRDD.map(x=>((x._1,x._2),x._3))
mapStudent_Marks: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[208] at map at <console>:28

scala> mapStudent_Marks.join(mapStudent_Grade).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect
res108: Array[((String, String), Float)] = Array(((Lisa,grade-1),24.0), ((Mark,grade-2),11.5), ((Mark,grade-2),6.0), ((Lisa,grade-2),12.0),
((Lisa,grade-2),49.0), ((Mathew,grade-3),45.0), ((Andrew,grade-2),77.0), ((Andrew,grade-1),11.333333), ((Andrew,grade-1),24.666666), ((Andre
w,grade-1),7.6666665), ((Lisa,grade-3),86.0), ((John,grade-1),4.6666665), ((John,grade-1),22.333334), ((John,grade-1),11.666667), ((John,gra
de-2),74.0), ((Mark,grade-1),38.0), ((Mark,grade-1),46.0), ((Andrew,grade-3),13.0), ((Andrew,grade-3),22.0), ((Mathew,grade-2),18.333334), (
(Mathew,grade-2),18.333334), ((Mathew,grade-2),29.0))
```

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

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(1), array(3).toInt))

val summ = tupleRDD.reduceByKey((x,y)=>(x+y))

val cnt =tupleRDD.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))

summ.join(cnt).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[127] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>( array(1), array(3).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[129] at map at <console>:26

scala> val summ = tupleRDD.reduceByKey((x,y)=>(x+y))
summ: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[130] at reduceByKey at <console>:28

scala> val cnt =tupleRDD.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))
cnt: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[132] at reduceByKey at <console>:28

scala> summ.join(cnt).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect
res73: Array[(String, Float)] = Array((maths,46.833332), (history,69.75), (science,38.25))

scala>
```

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

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt))

val mapStudent_Grade_Subjects = tupleRDD.map(x=>((x._1,x._2,x._3),1)).reduceByKey((x,y)=>(x+y))

val mapStudent_Grade_Marks = tupleRDD.map(x=>((x._1,x._2,x._3),x._4)).reduceByKey((x,y)=>(x+y))

mapStudent_Grade_Subjects.join(mapStudent_Grade_Marks).map(x=>(x._1,(x._2._2.toFloat/x._2._1.toFloat))).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[235] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(1),array(2),array(3).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, String, Int)] = MapPartitionsRDD[237] at map at <console>:26

scala> val mapStudent_Grade_Subjects = tupleRDD.map(x=>((x._1,x._2,x._3),1)).reduceByKey((x,y)=>(x+y))
mapStudent_Grade_Subjects: org.apache.spark.rdd.RDD[((String, String, String), Int)] = ShuffledRDD[239] at reduceByKey at <console>:28

scala> val mapStudent_Grade_Marks = tupleRDD.map(x=>((x._1,x._2,x._3),x._4)).reduceByKey((x,y)=>(x+y))
mapStudent_Grade_Marks: org.apache.spark.rdd.RDD[((String, String, String), Int)] = ShuffledRDD[241] at reduceByKey at <console>:28

scala> mapStudent_Grade_Subjects.join(mapStudent_Grade_Marks).map(x=>(x._1,(x._2._2.toFloat/x._2._1.toFloat))).collect
res116: Array[((String, String, String), Float)] = Array(((Lisa,history,grade-3),86.0), ((John,history,grade-1),40.5), ((Andrew,history,grad
e-2),77.0), ((John,maths,grade-2),74.0), ((Andrew,maths,grade-1),28.5), ((Mark,maths,grade-2),23.0), ((Mark,science,grade-2),12.0), ((Andrew
,science,grade-3),35.0), ((Mathew,science,grade-3),45.0), ((Mathew,history,grade-2),71.0), ((Andrew,history,grade-1),74.0), ((John,maths,gra
de-1),35.0), ((Mark,maths,grade-1),92.0), ((Mark,science,grade-1),76.0), ((Mathew,science,grade-2),55.0), ((Lisa,science,grade-2),24.0), ((L
isa,history,grade-2),98.0), ((Lisa,science,grade-1),24.0))
```

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

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))

val Student_Grade2 = tupleRDD.map(x=>(x._1,x._2,x._3,(x._2=="grade-2"))).filter(x=>(x._4==true))

val Student_Grd2_Total = Student_Grade2.map(x=>(x._1,x._3)).reduceByKey((x,y)=>(x+y))

val Student_Grd2_Cnt = Student_Grade2.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))

Student_Grd2_Total.join(Student_Grd2_Cnt).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[3] at map at <console>:26

scala> val Student_Grade2 = tupleRDD.map(x=>(x._1,x._2,x._3,(x._2=="grade-2"))).filter(x=>(x._4==true))
Student_Grade2: org.apache.spark.rdd.RDD[(String, String, Int, Boolean)] = MapPartitionsRDD[5] at filter at <console>:28

scala> val Student_Grd2_Total = Student_Grade2.map(x=>(x._1,x._3)).reduceByKey((x,y)=>(x+y))
Student_Grd2_Total: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[7] at reduceByKey at <console>:30

scala> val Student_Grd2_Cnt = Student_Grade2.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))
Student_Grd2_Cnt: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[9] at reduceByKey at <console>:30

scala> Student_Grd2_Total.join(Student_Grd2_Cnt).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat))).collect
res0: Array[(String, Float)] = Array((Mark,17.5), (Andrew,77.0), (Mathew,65.666664), (John,74.0), (Lisa,61.0))
```

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.

val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")

val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(3).toInt))

val Student_Total = tupleRDD.reduceByKey((x,y)=>(x+y))

val Student_Count = tupleRDD.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))

val Student_avg = Student_Total.join(Student_Count).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat)))

val tupleRDD1 = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))

val Student_Total1 = tupleRDD1.map(x=>((x._1,x._2),x._3)).reduceByKey((x,y)=>(x+y))

val Student_Count1 = tupleRDD1.map(x=>((x._1,x._2),1)).reduceByKey((x,y)=>(x+y))

val Student_avg1 = Student_Total1.join(Student_Count1).map(x=>(x._1._1,(x._2._1.toFloat/x._2._2.toFloat)))

Student_avg.intersection(Student_avg1).collect

```
scala> val studentRDD = sc.textFile("/home/acadgild/inputdir/student_dataset.txt")
studentRDD: org.apache.spark.rdd.RDD[String] = /home/acadgild/inputdir/student_dataset.txt MapPartitionsRDD[110] at textFile at <console>:24

scala> val tupleRDD = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(3).toInt))
tupleRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[112] at map at <console>:26

scala> val Student_Total = tupleRDD.reduceByKey((x,y)=>(x+y))
Student_Total: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[113] at reduceByKey at <console>:28

scala> val Student_Count = tupleRDD.map(x=>(x._1,1)).reduceByKey((x,y)=>(x+y))
Student_Count: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[115] at reduceByKey at <console>:28

scala> val Student_avg = Student_Total.join(Student_Count).map(x=>(x._1,(x._2._1.toFloat/x._2._2.toFloat)))
Student_avg: org.apache.spark.rdd.RDD[(String, Float)] = MapPartitionsRDD[119] at map at <console>:32

scala> Student_avg.collect
res34: Array[(String, Float)] = Array((Mark,50.75), (Andrew,46.333332), (Mathew,60.5), (John,47.5), (Lisa,58.0))

scala> val tupleRDD1 = studentRDD.map(line=>line.split(",")).map(array=>(array(0),array(2),array(3).toInt))
tupleRDD1: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[121] at map at <console>:26

scala> val Student_Total1 = tupleRDD1.map(x=>((x._1,x._2),x._3)).reduceByKey((x,y)=>(x+y))
Student_Total1: org.apache.spark.rdd.RDD[((String, String), Int)] = ShuffledRDD[123] at reduceByKey at <console>:28

scala> val Student_Count1 = tupleRDD1.map(x=>((x._1,x._2),1)).reduceByKey((x,y)=>(x+y))
Student_Count1: org.apache.spark.rdd.RDD[((String, String), Int)] = ShuffledRDD[125] at reduceByKey at <console>:28

scala> val Student_avg1 = Student_Total1.join(Student_Count1).map(x=>(x._1._1,(x._2._1.toFloat/x._2._2.toFloat)))
Student_avg1: org.apache.spark.rdd.RDD[(String, Float)] = MapPartitionsRDD[129] at map at <console>:32

scala> Student_avg1.collect
res35: Array[(String, Float)] = Array((Lisa,24.0), (Mark,17.5), (Lisa,61.0), (Mathew,45.0), (Andrew,77.0), (Andrew,43.666668), (Lisa,86.0),
(John,38.666668), (John,74.0), (Mark,84.0), (Andrew,35.0), (Mathew,65.666664))

scala> Student_avg.intersection(Student_avg1).collect
res36: Array[(String, Float)] = Array()
```

There are no students who satisfy the criteria.