

Assignment19:

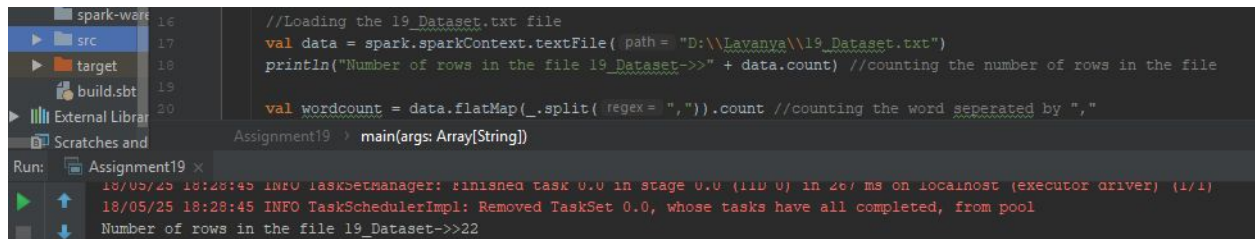
Task 1 :

1. Write a program to read a text file and print the number of rows of data in the document.

Query used:

```
//Loading the 19_Dataset.txt file
val data = spark.sparkContext.textFile("D:\\Lavanya\\19_Dataset.txt")
println("Number of rows in the file 19_Dataset->>" + data.count) //counting the number
of rows in the file
```

Output:

A screenshot of a Spark IDE interface. The left sidebar shows a project structure with folders 'src' and 'target', and files 'build.sbt' and 'External Librar'. The main editor area displays Scala code for loading a text file and counting its rows. The output console at the bottom shows the execution of the code, with a message indicating the number of rows in the file is 22.

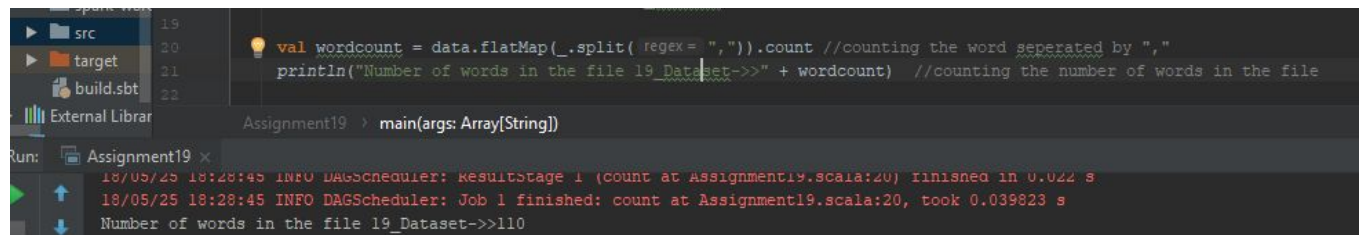
```
spark-wor 16
  17 //Loading the 19_Dataset.txt file
  18 val data = spark.sparkContext.textFile("D:\\Lavanya\\19_Dataset.txt")
  19 println("Number of rows in the file 19_Dataset->>" + data.count) //counting the number
  20 of rows in the file
Assignment19 > main(args: Array[String])
Run: Assignment19 x
18/05/25 18:28:45 INFO TaskSetManager: finished task 0.0 in stage 0.0 (110 U) in 26/ ms on localhost (executor driver) (1/1)
18/05/25 18:28:45 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
Number of rows in the file 19_Dataset->>22
```

2. Write a program to read a text file and print the number of words in the document.

Query used:

```
val wordcount = data.flatMap(_.split(",")).count //counting the word seperated by ","
println("Number of words in the file 19_Dataset->>" + wordcount) //counting the
number of words in the file
```

Output:



```
19
20 val wordcount = data.flatMap(_.split(regex = ",")).count //counting the word seperated by ","
21 println("Number of words in the file 19_Dataset->>" + wordcount) //counting the number of words in the file
22
Assignment19 > main(args: Array[String])

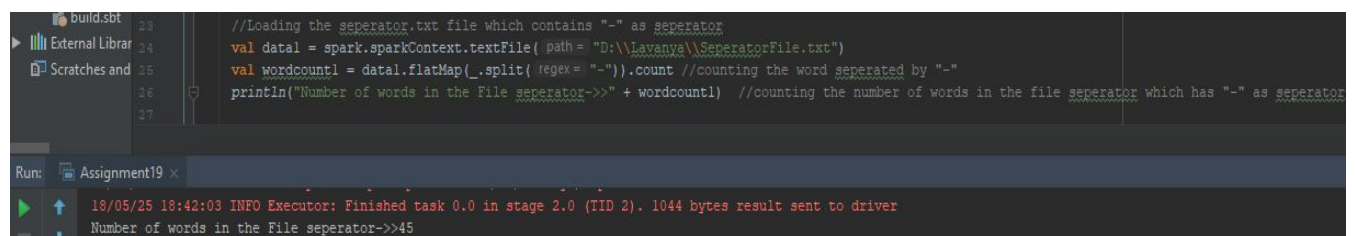
Run: Assignment19 x
18/05/25 18:28:45 INFO DAGScheduler: ResultStage 1 (count at Assignment19.scala:20) finished in 0.022 s
18/05/25 18:28:45 INFO DAGScheduler: Job 1 finished: count at Assignment19.scala:20, took 0.039823 s
Number of words in the file 19_Dataset->>110
```

3. We have a document where the word separator is -, instead of space.
Write a spark code, to obtain the count of the total number of words present in the document.

Query used:

```
//Loading the separator.txt file which contains "-" as separator
val data1 = spark.sparkContext.textFile("D:\\Lavanya\\SeperatorFile.txt")
val wordcount1 = data1.flatMap(_.split(regex = "-")).count //counting the word seperated by "-"
println("Number of words in the File separator->>" + wordcount1) //counting the number of words in the file separator which has "-" as separator.
```

Output:



```
23 //Loading the separator.txt file which contains "-" as separator
24 val data1 = spark.sparkContext.textFile(path = "D:\\Lavanya\\SeperatorFile.txt")
25 val wordcount1 = data1.flatMap(_.split(regex = "-")).count //counting the word seperated by "-"
26 println("Number of words in the File separator->>" + wordcount1) //counting the number of words in the file separator which has "-" as separator
27

Run: Assignment19 x
18/05/25 18:42:03 INFO Executor: Finished task 0.0 in stage 2.0 (TID 2). 1044 bytes result sent to driver
Number of words in the File separator->>45
```

Task 2:

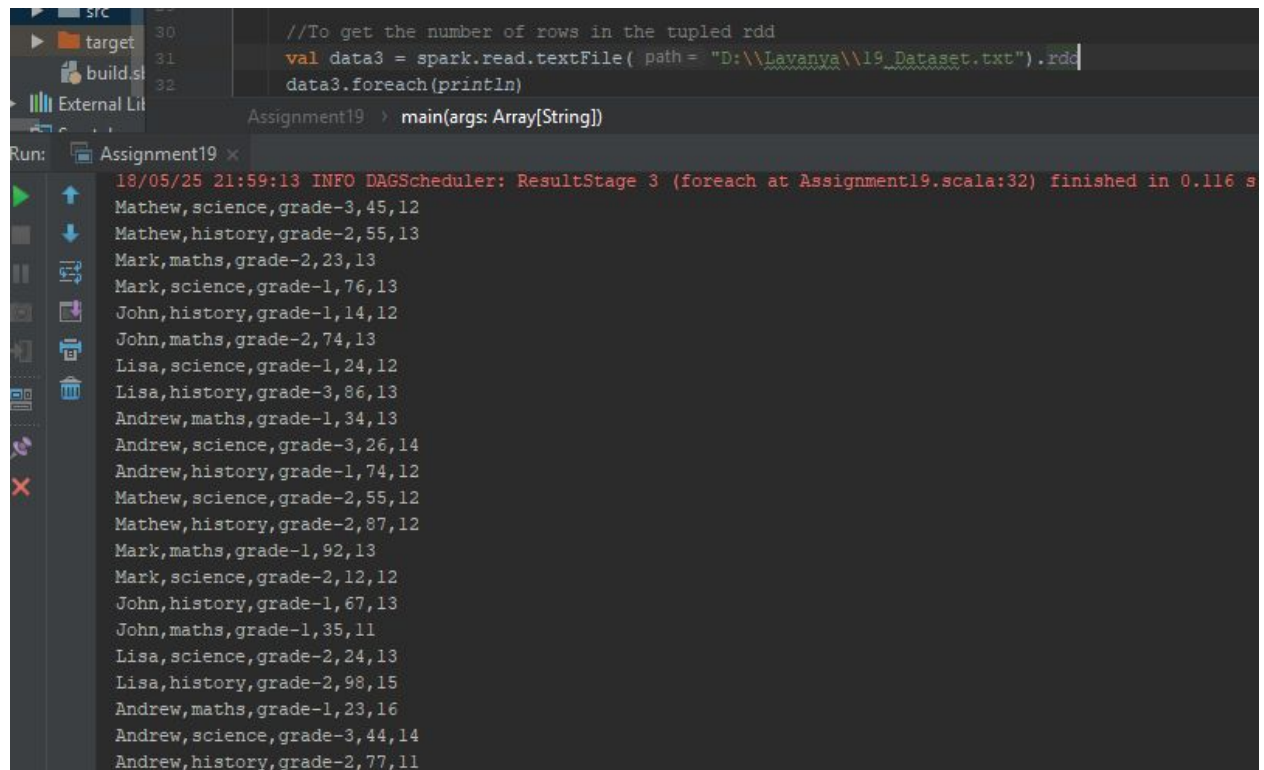
Problem Statement 1:

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

Query used:

```
-----  
//To get the number of rows in the tupled rdd  
val data3 = spark.read.textFile("D:\\Lavanya\\19_Dataset.txt").rdd  
data3.foreach(println)
```

Output:



```
Run: Assignment19 x  
18/05/25 21:59:13 INFO DAGScheduler: ResultStage 3 (foreach at Assignment19.scala:32) finished in 0.116 s  
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
```

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

Query used:

```
-----  
//To get the number of rows in the tupled rdd  
val data3 = spark.read.textFile("D:\\Lavanya\\19_Dataset.txt").rdd  
data3.foreach(println)  
println("The number of rows in the rdd tuple data3 is " + data3.count())
```

Output:

```
build.sbt 30 //To get the number of rows in the tupled rdd
31 val data3 = spark.read.textFile( path = "D:\\Lavanva\\l19_Dataset.txt").rdd
32 data3.foreach(println)
33 println("The number of rows in the rdd tuple data3 is " + data3.count())
34

Assignment19 > main(args: Array[String])

Run: Assignment19 x
18/05/25 21:59:14 INFO DAGScheduler: Job 4 finished: count at Assignment19.scala:33, took 0.029276 s
The number of rows in the rdd tuple data3 is 22
```

```
project 37 val data4 = data3.map(x=>x.split( regex=",")).map(x => Students_cls(x(0),x(1),x(2),x(3).toInt,x(4).toInt)).toDF()
spark-v 38 data4.show()
src 39 data4.registerTempTable( tableName = "StudentsMark") //Registering as temporary table HVAC.
target 40 println("Dataframe Registered as table !")
41

Assignment19 > main(args: Array[String])

Run: Assignment19 x

+-----+-----+-----+-----+-----+
| name|Subject| grade|mark| Id|
+-----+-----+-----+-----+-----+
|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|
```

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

Query used:

```
-----
//To get the number of Distinct Subjects
val DistinctSubject = spark.sql("select distinct(Subject) from StudentsMark").count()
println("The number of Distinct Subjects are :" + DistinctSubject)
```

Output:

```
41 //To get the number of Distinct Subjects
42
43 val DistinctSubject = spark.sql( sqlText = "select distinct(Subject) from StudentsMark").count()
44 println("The number of Distinct Subjects are :" + DistinctSubject)
45
Assignment19 > main(args: Array[String])

Run: Assignment19 x
18/05/25 21:48:43 INFO CodeGenerator: Code generated in 9.392921 ms
The number of Distinct Subjects are :3
```

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

Query used:

```
-----
//To count the number of students having name as mathew and mark as 55
val StudCount = spark.sql("select * from StudentsMark where name = 'Mathew' and mark = 55").count()
println("The count of number of students having name as mathew and mark as 55 :" + StudCount)
```

Output:

```
Scratches: 46 //To count the number of students having name as mathew and mark as 55
47 val StudCount = spark.sql( sqlText = "select * from StudentsMark where name = 'Mathew' and mark = 55").count()
48 println("The count of number of students having name as mathew and mark as 55 :" + StudCount)
Assignment19 > main(args: Array[String])

Run: Assignment19 x
18/05/25 21:48:43 INFO TaskSetManager: Starting task 0.0 in stage 8.0 (110 207, localhost, executor driver, partition 0, ANY, 5885 bytes)
The count of number of students having name as mathew and mark as 55 :2
```

Problem Statement 2:

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

Query used:

```
-----
//To count of students per grade in the school
val AvgCountpergrade = spark.sql(sqlText= "select count(name), grade from StudentsMark group by grade")
AvgCountpergrade.show()
```

Output:

```
Scratches:
50 //To count of students per grade in the school
51 val AvgCountpergrade = spark.sql(sqlText= "select count(name), grade from StudentsMark group by grade")
52 AvgCountpergrade.show()
53
Assignment19 > main(args: Array[String])

Run: Assignment19 x
18/05/25 22:38:56 INFO DAGScheduler: Job 12 finished: show at Assignment19.scala:52, took 0.467438 s
+-----+
|count(name)| grade|
+-----+
|          4|grade-3|
|          9|grade-1|
|          9|grade-2|
+-----+
```

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

Query used:

```
//To count of students per grade in the school
val AvgStudentsgrade = spark.sql(sqlText= "select name,avg(mark), grade from StudentsMark group by grade, name")
AvgStudentsgrade.show()
```

Output:

```
project
├── spark-y
├── src
├── target
└── build.sbt

Assignment19 > main(args: Array[String])

Run: Assignment19 x
+-----+
| name|      avg(mark)| grade|
+-----+
| Andrew|      35.0|grade-3|
| John|38.666666666666664|grade-1|
| Lisa|      24.0|grade-1|
| Lisa|      86.0|grade-3|
| Lisa|      61.0|grade-2|
| Mathew|      45.0|grade-3|
| John|      74.0|grade-2|
| Mark|      84.0|grade-1|
| Andrew|43.666666666666664|grade-1|
| Andrew|      77.0|grade-2|
| Mathew|65.666666666666667|grade-2|
| Mark|      17.5|grade-2|
+-----+
```

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

Query used:

```
-----  
//To get average score of students in each subject across all grades  
val AvgScoreperSubject = spark.sql(sqlText= "select name,avg(mark), subject from  
StudentsMark group by subject, name")  
AvgScoreperSubject.show()
```

Output:

```
-----  
57  
58 //To get average score of students in each subject across all grades  
59 val AvgScoreperSubject = spark.sql(sqlText= "select name,avg(mark), subject from StudentsMark group by subject, name")  
60 AvgScoreperSubject.show()  
-----  
Assignment19 > main(args: Array[String])  
Assignment19 x  
18/05/26 18:10:07 INFO DAGScheduler: ResultStage 40 (show at Assignment19.scala:60) finished in 0.373 s  
18/05/26 18:10:07 INFO DAGScheduler: Job 22 finished: show at Assignment19.scala:60, took 0.379625 s  
+-----+-----+  
| name|avg(mark)|subject|  
+-----+-----+  
| Mathew| 71.0|history|  
| Mathew| 45.0|science|  
| Mathew| 55.0|science|  
| Lisa| 24.0|science|  
| Mark| 44.0|science|  
| Lisa| 92.0|history|  
| Mark| 57.5| maths|  
| John| 40.5|history|  
| Andrew| 35.0|science|  
| Andrew| 75.5|history|  
| Andrew| 28.5| maths|  
| John| 54.5| maths|  
+-----+-----+
```

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

Query used:

```
-----  
//To get What is the average score of students in each subject per grade?  
val AvgScoreinSubjectpergrade = spark.sql(sqlText= "select name,avg(mark),subject,  
grade from StudentsMark group by grade, subject, name")  
AvgScoreinSubjectpergrade.show()
```


Output:

```
-----
project [src] 62 //To get What is the average score of students in each subject per grade?
spark-ware 63 val AvgScoreinSubjectpergrade = spark.sql(sqlText= "select name,avg(mark),subject, grade from StudentsMark group by subject, name, grade")
src 64 AvgScoreinSubjectpergrade.show()
main
Run: Assignment19 x main(args: Array[String])
18/05/26 18:24:13 INFO SparkContext: Invoking stop() from shutdown hook
+-----+
| name|avg(mark)|subject| grade|
+-----+
| Mathew| 45.0|science|grade-3|
| John| 74.0| maths|grade-2|
| Mark| 23.0| maths|grade-2|
| Lisa| 24.0|science|grade-2|
| Andrew| 35.0|science|grade-3|
| Mathew| 71.0|history|grade-2|
| Lisa| 86.0|history|grade-3|
| Andrew| 28.5| maths|grade-1|
| Mark| 12.0|science|grade-2|
| John| 40.5|history|grade-1|
| Mark| 92.0| maths|grade-1|
| Andrew| 74.0|history|grade-1|
| Andrew| 77.0|history|grade-2|
| Mathew| 55.0|science|grade-2|
| John| 35.0| maths|grade-1|
| Lisa| 98.0|history|grade-2|
| Lisa| 24.0|science|grade-1|
| Mark| 76.0|science|grade-1|
+-----+
```

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

Query used:

```
-----
//To get how many have average score greater than 50 in grade-2?
val AvgScoregreaterthan50 = spark.sql(sqlText= "select name ,avg(mark), grade from
StudentsMark group by name, grade having grade = 'grade-2' and avg(mark) > 50 ")
AvgScoregreaterthan50.show()
```

Output:

```
-----
src 66
main 67 //To get how many have average score greater than 50 in grade-2?
scala 68 val AvgScoregreaterthan50 = spark.sql(sqlText= "select name ,avg(mark), grade from StudentsMark group by name, grade having grade = 'grade-2' and
Assignment12.sc 69 AvgScoregreaterthan50.show()
Assignment15.sc 70
Run: Assignment19 x main(args: Array[String])
18/05/26 18:31:23 INFO SparkContext: Started 0 remote RDDs in 0 ms
+-----+
| name| avg(mark)| grade|
+-----+
| John| 74.0|grade-2|
| Lisa| 61.0|grade-2|
| Mathew| 65.66666666666667|grade-2|
| Andrew| 77.0|grade-2|
+-----+
```


Problem Statement 3:

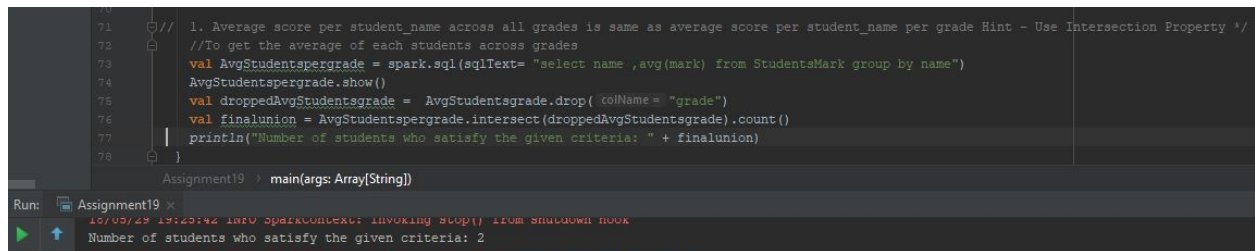
Are there any students in the college that satisfy the below criteria:

1. Average score per student_name across all grades is same as average score per student_name per grade Hint - Use Intersection Property

Query used:

```
// 1. Average score per student_name across all grades is same as average score per
student_name per grade Hint - Use Intersection Property */
//To get the average of each students across grades
val AvgStudentspergrade = spark.sql(sqlText= "select name ,avg(mark) from
StudentsMark group by name")
AvgStudentspergrade.show()
val droppedAvgStudentsgrade = AvgStudentsgrade.drop("grade")
val finalunion = AvgStudentspergrade.intersect(droppedAvgStudentsgrade).count()
println("Number of students who satisfy the given criteria: " + finalunion)
```

Output:



The screenshot shows a code editor with the following code:

```
71 // 1. Average score per student_name across all grades is same as average score per student_name per grade Hint - Use Intersection Property */
72 //To get the average of each students across grades
73 val AvgStudentspergrade = spark.sql(sqlText= "select name ,avg(mark) from StudentsMark group by name")
74 AvgStudentspergrade.show()
75 val droppedAvgStudentsgrade = AvgStudentsgrade.drop( colName = "grade")
76 val finalunion = AvgStudentspergrade.intersect(droppedAvgStudentsgrade).count()
77 | println("Number of students who satisfy the given criteria: " + finalunion)
78 }
```

Below the code editor, the output of the program is displayed:

```
Run: Assignment19 x
15/09/23 15:23:42 INFO SparkContext: Invoking stop() from shutdown hook
Number of students who satisfy the given criteria: 2
```