Session 18

Assignment 3

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 18 – Assignment 3 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 18 – Assignment 3** |
|  |  |
| **Start Date:** | 18/10/2017 |
|  |  |
|  |  |



# 

# Contents

[Contents 2](#_Toc496186126)

[Change History 3](#_Toc496186127)

[1. Problem Statement 4](#_Toc496186128)

[2. Datasets 4](#_Toc496186129)

[3. Solution 5](#_Toc496186130)

[4. Results 7](#_Toc496186131)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 19/10/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

* Considering age groups of < 20 , 20-35, 35 > ,Which age group spends the most amount of money travelling.
* What is the amount spent by each age-group, every year in travelling?

Using the datasets below:

# Datasets

**S18\_Dataset\_Holidays.txt**

1,CHN,IND,airplane,200,1990

2,IND,CHN,airplane,200,1991

3,IND,CHN,airplane,200,1992

4,RUS,IND,airplane,200,1990

5,CHN,RUS,airplane,200,1992

6,AUS,PAK,airplane,200,1991

7,RUS,AUS,airplane,200,1990

8,IND,RUS,airplane,200,1991

9,CHN,RUS,airplane,200,1992

10,AUS,CHN,airplane,200,1993

1,AUS,CHN,airplane,200,1993

2,CHN,IND,airplane,200,1993

3,CHN,IND,airplane,200,1993

4,IND,AUS,airplane,200,1991

5,AUS,IND,airplane,200,1992

6,RUS,CHN,airplane,200,1993

7,CHN,RUS,airplane,200,1990

8,AUS,CHN,airplane,200,1990

9,IND,AUS,airplane,200,1991

10,RUS,CHN,airplane,200,1992

1,PAK,IND,airplane,200,1993

2,IND,RUS,airplane,200,1991

3,CHN,PAK,airplane,200,1991

4,CHN,PAK,airplane,200,1990

5,IND,PAK,airplane,200,1991

6,PAK,RUS,airplane,200,1991

7,CHN,IND,airplane,200,1990

8,RUS,IND,airplane,200,1992

9,RUS,IND,airplane,200,1992

10,CHN,AUS,airplane,200,1990

1,PAK,AUS,airplane,200,1993

5,CHN,PAK,airplane,200,1994

**S18\_Dataset\_User\_Details.txt**

1,mark,15

2,john,16

3,luke,17

4,lisa,27

5,mark,25

6,peter,22

7,james,21

8,andrew,55

9,thomas,46

10,annie,44

**S18\_Dataset**\_Transport.txt

airplane,170

car,140

train,120

ship,200

# Solution

I have deliberately commented and printed out all steps for enhanced learning

**Code written**

**package** com.duncb.spark

**import** org.apache.spark.\_

**import** org.apache.spark.SparkContext.\_

**import** org.apache.log4j.\_

**object** holiday3 {

/\*\* A function that splits a line of input into (userID,src,dest,travelMode,distance,yearOfTravel) tuples. \*/

**def** travelInfo(line: *String*) = {

// Split by commas

**val** fields = line.split(",")

**val** userID = fields(0).toInt

**val** src = fields(1).toString

**val** dest = fields (2).toString

**val** travelMode = fields (3).toString

**val** distance = fields (4).toInt

**val** yearOfTravel =fields(5).toInt

// Create a tuple that is our result.

(userID,src,dest,travelMode,distance,yearOfTravel)

}

/\*\* A function that splits a line of input into (userID,name and age) tuples. \*/

**def** userInfo(line: *String*) = {

// Split by commas

**val** fields = line.split(",")

**val** userID = fields(0).toInt

**val** userName = fields(1).toString

**val** age = fields (2).toInt

// Create a tuple that is our result.

(userID,userName,age)

}

/\*\* A function that splits a line of input into (transport,cost) tuples. \*/

**def** transportInfo(line: *String*) = {

// Split by commas

**val** fields = line.split(",")

**val** transport = fields(0).toString

**val** cost = fields (1).toInt

// Create a tuple that is our result.

(transport,cost)

}

**def** main(args: Array[*String*]) {

// Set the log level to only print errors

Logger.getLogger("org").setLevel(Level.ERROR)

// Create a SparkContext using every core of the local machine

**val** sc = **new** SparkContext("local[1]", "holidayData")

// Load each line of the source data into an holidaysRDD

**val** linesH = sc.textFile("file:///N:/Datasets/S18\_Dataset\_Holidays.txt")

**val** holidaysRDD = linesH.map(travelInfo)

// Load each line of the source data into a usersRDD

**val** linesU = sc.textFile("file:///N:/Datasets/S18\_Dataset\_User\_Details.txt")

**val** usersRDD = linesU.map(userInfo)

// Load each line of the source data into a transportRDD

**val** linesT = sc.textFile("file:///N:/Datasets/S18\_Dataset\_Transport.txt")

**val** transportRDD = linesT.map(transportInfo)

**val** ut1 = holidaysRDD.map(x=> (x.\_1,x.\_4))

**val** ua1 = usersRDD.map(x=>(x.\_1,x.\_3))

// Create RDD with the 2 columns required

println(" Assignment 18.3 Problem 1")

//join RDD's on columns required

**val** ags1 = ua1.join(ut1).map(x=>(x.\_2.\_2,x.\_2.\_1)).join(transportRDD).map(x=> (x.\_2.\_1,x.\_2.\_2))

ags1.foreach(println)

println ("-----------------------------------------------------")

**val** ags2 = ags1.groupByKey().map(x=>(x.\_1,x.\_2.sum))

ags2.foreach(println)

println ("-----------------------------------------------------")

//Find total spend for the age groups

**val** ags3 =ags2.map(x=> **if**(x.\_1<20) x.\_2.toInt).filter(x => x!=()).map(x=> x.asInstanceOf[Int]).sum

**val** ags4 =ags2.map(x=> **if**(x.\_1>20 && x.\_1<=35) x.\_2.toInt).filter(x => x!=()).map(x=> x.asInstanceOf[Int]).sum

**val** ags5 =ags2.map(x=> **if**(x.\_1>35) x.\_2.toInt).filter(x => x!=()).map(x=> x.asInstanceOf[Int]).sum

println (ags3)

println (ags4)

println (ags5)

**val** ags6 = List((" Less than 20 years old ",ags3),(" Between 20 and 35 years old ",ags4),(" Older than 55 years old",ags5)).sortBy(x=> -x.\_2).take(1)

println ("-----------------------------------------------------")

println ("The age group that spends the most is " + ags6)

println(" Assignment 18.3 Problem 2")

//obtain columns required and join

**val** ut2 = holidaysRDD.map(x=>(x.\_1,(x.\_4,x.\_6)))

**val** ua2 = usersRDD.map(x=> (x.\_1,x.\_3))

**val** sagy1=ut2.join(ua2).map(x=>(x.\_2.\_1.\_1,(x.\_2.\_1.\_2,x.\_2.\_2))).join(transportRDD).map(x=>x.\_2)

//total spend for age and year

**val** sagy2=sagy1.groupByKey.map(x=>((x.\_1.\_2,x.\_1.\_1),x.\_2.sum))

sagy2.foreach(println)

//filter age ranges and obtain total sum per year

**val** sagy3 = sagy2.filter(x=> x.\_1.\_1 <20).map(x=>(x.\_1.\_2,x.\_2)).groupByKey.map(x=>(x.\_1,x.\_2.sum)).sortBy(x=> x.\_1)

**val** sagy4 = sagy2.filter(x=> x.\_1.\_1 >20 && x.\_1.\_1<=35).map(x=>(x.\_1.\_2,x.\_2)).groupByKey.map(x=>(x.\_1,x.\_2.sum)).sortBy(x=> x.\_1)

**val** sagy5 = sagy2.filter(x=> x.\_1.\_1 >35).map(x=>(x.\_1.\_2,x.\_2)).groupByKey.map(x=>(x.\_1,x.\_2.sum)).sortBy(x=> x.\_1)

println ("-----------------------------------------------------")

//Display the results

println("Amount spent by people younger than 20 years old's in the years shown")

sagy3.foreach(println)

println("Amount spent by people between 20 and 35 years old in the years shown")

sagy4.foreach(println)

println("Amount spent by people older than 55 years old in the years shown")

sagy5.foreach(println)

}

}

# Results

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Assignment 18.3 Problem 1

(27,170)

(27,170)

(27,170)

(15,170)

(15,170)

(15,170)

(15,170)

(22,170)

(22,170)

(22,170)

(17,170)

(17,170)

(17,170)

(21,170)

(21,170)

(21,170)

(46,170)

(46,170)

(46,170)

(55,170)

(55,170)

(55,170)

(44,170)

(44,170)

(44,170)

(25,170)

(25,170)

(25,170)

(25,170)

(16,170)

(16,170)

(16,170)

-----------------------------------------------------

(15,680)

(21,510)

(16,510)

(55,510)

(22,510)

(25,680)

(46,510)

(27,510)

(17,510)

(44,510)

-----------------------------------------------------

1700.0

2210.0

1530.0

-----------------------------------------------------

The age group that spends the most is List(( Between 20 and 35 years old ,2210.0))

Assignment 18.3 Problem 2

((17,1993),170)

((55,1990),170)

((44,1993),170)

((16,1993),170)

((22,1993),170)

((46,1991),170)

((27,1990),340)

((17,1992),170)

((22,1991),340)

((25,1991),170)

((46,1992),340)

((17,1991),170)

((55,1991),170)

((55,1992),170)

((44,1990),170)

((25,1992),340)

((27,1991),170)

((15,1990),170)

((44,1992),170)

((21,1990),510)

((16,1991),340)

((15,1993),510)

((25,1994),170)

-----------------------------------------------------

Amount spent by people younger than 20 years old's in the years shown

(1990,170)

(1991,510)

(1992,170)

(1993,850)

Amount spent by people between 20 and 35 years old in the years shown

(1990,850)

(1991,680)

(1992,340)

(1993,170)

(1994,170)

Amount spent by people older than 55 years old in the years shown

(1990,340)

(1991,340)

(1992,680)

(1993,170)