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?

Dataset

https://drive.google.com/drive/folders/0B_P3pWagdIrrVThBaUdVSUtzbms

Dataset-Holidays:
The dataset is of holiday details of travelers with columns: **user_id**, **source**, **destination**, **travel_mode**, **distance**, **year_of_travel**:

```
[acadgild@localhost ~]$ cat /home/acadgild/Downloads/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
[acadgild@localhost ~]$
```

Dataset-Transport:
The dataset is of transport details with columns: **travel_mode**, **cost_per_unit**:

```
[acadgild@localhost ~]$ cat /home/acadgild/Downloads/Transport.txt
airplane,170
car,140
train,120
ship,200[acadgild@localhost ~]$
```

Dataset-User_details:
The dataset is of user details of travelers with columns: **user_id**, **name,age**:

```
ship,200[acadgild@localhost ~]$ cat /home/acadgild/Downloads/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[acadgild@localhost ~]$
```

Intialization Spark-Shell:

```
Acadgild_SPARK_FINAL_64bit [Running] - Oracle VM VirtualBox
Help
Applications  Places  System                                   Sat Dec 16, 12:49 PM   acadgil
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ spark-shell
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Using Spark's repl log4j profile: org/apache/spark/log4j-defaults-repl.properties
To adjust logging level use sc.setLogLevel("INFO")
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_65)
Type in expressions to have them evaluated.
Type :help for more information.
17/12/16 12:35:35 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 10.0.2.15
 instead (on interface eth6)
17/12/16 12:35:35 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Spark context available as sc.
17/12/16 12:35:53 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/16 12:35:56 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/16 12:36:14 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not ena
bled so recording the schema version 1.2.0
17/12/16 12:36:15 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
17/12/16 12:36:25 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/16 12:36:26 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
SQL context available as sqlContext.
```

- Creating tupleRdd travelRDD from dataset Holidays.txt

```
scala> val travel = sc.textFile("/home/acadgild/Downloads/Holidays.txt")
travel: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[67] at textFile at <console>:27

scala> val travelRDD  = travel.map(x=> {
     |  val w = x.split(",")
     |  val user_id = w(0)
     |  val src = w(1)
     |  val dest = w(2)
     |  val travel_mode = w(3)
     |  val distance = w(4).toInt
     |  val year_of_travel = w(5).toInt
     |  (user_id,src,dest,travel_mode,distance,year_of_travel)
     | })
travelRDD: org.apache.spark.rdd.RDD[(String, String, String, String, Int, Int)] = MapPartitionsRDD[68] at map at <console>:29
```

- Displaying all data tupleRdd travelRDD

```
scala> travelRDD.foreach(println)
(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)

scala>
```

- Creating tupleRdd transportRdd from dataset Transport.txt

```
scala> val transport = sc.textFile("/home/acadgild/Downloads/Transport.txt")
transport: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[70] at textFile at <console>:27

scala> val transportRDD = transport.map(x=> {
 val w = x.split(",")
 val travel_mode = w(0)
 val cost_per_unit = w(1).toInt
 (travel_mode,cost_per_unit)
})
transportRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[71] at map at <console>:29
```

- Displaying all data tupleRdd transportRDD

```
scala> transportRDD.foreach(println)
(airplane,170)
(car,140)
(train,120)
(ship,200)

scala>
```

- Creating tupleRdd userRDD from dataset User_details.txt

```
scala> val user = sc.textFile("/home/acadgild/Downloads/User_details.txt")
user: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[73] at textFile at <console>:27

scala> val userRDD = user.map(x=> {
     |
     |  val w = x.split(",")
     |
     |  val user_id = w(0)
     |
     |  val name = w(1)
     |
     |  val age = w(2).toInt
     |
     |  (user_id,name,age)
     |
     | })
userRDD: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[74] at map at <console>:29
```

- Displaying all data tupleRdd userRDD

```
scala> userRDD.foreach(println)
(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)
```

- Considering age groups of < 20 , 20-35, 35 > ,Which age group spends the most amount of money travelling.

Grouping Age Groups of <20,20-35,35>

```
scala> val filterAgeGroup = userRDD.map(x => x._1 -> {
     |
     | if(x._3 < 20)
     |
     | "<20"
     | else if (x._3 > 35)
     |
     | ">35"
     |
     | else "20-35"
     |
     | })
filterAgeGroup: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[75] at map at <console>:31
```

Code:

Finding age group spends the most amount of money travelling.

```
scala> val f1 = travelRDD.map(x => x._4 -> (x._1, x._5, x._6))
f1: org.apache.spark.rdd.RDD[(String, (String, Int, Int))] = MapPartitionsRDD[108] at map at <console>:31

scala>

scala> val f2 = transportRDD.map(x => x._1 -> x._2)
f2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[109] at map at <console>:31

scala>

scala> val jtwo = f1.join(f2)
jtwo: org.apache.spark.rdd.RDD[(String, ((String, Int, Int), Int))] = MapPartitionsRDD[112] at join at <console>:39

scala>

scala> val trans = jtwo.map(x => (x._2._1._1, x._2._1._3) -> (x._2._1._2 * x._2._2))
trans: org.apache.spark.rdd.RDD[((String, Int), Int)] = MapPartitionsRDD[113] at map at <console>:41

scala>

scala> val transMap = trans.map(x => (x._1._1) -> x._2)
transMap: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[114] at map at <console>:43

scala>

scala> val j = filterAgeGroup.join(transMap)
j: org.apache.spark.rdd.RDD[(String, (String, Int))] = MapPartitionsRDD[117] at join at <console>:51

scala>

scala> val finalTrans = j.map(x => (x._2._1) -> (x._2._2))
finalTrans: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[118] at map at <console>:53
```

```
scala>

scala> val grp = finalTrans.groupByKey()
grp: org.apache.spark.rdd.RDD[(String, Iterable[Int])] = ShuffledRDD[119] at groupByKey at <console>:55

scala>

scala> val output= grp.map(x => x._1 -> x._2.sum)
output: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[120] at map at <console>:57

scala>
```

Output:

Displaying age group spends the most amount of money travelling.

```
scala> val result = output.sortBy(x => -x._2).first()
result: (String, Int) = (20-35,442000)
```

- What is the amount spent by each age-group, every year in travelling?

Code:

Finding amount spent by each age-group, every year in travelling

```
scala> val f1 = travelRDD.map(x => x._4 -> (x._1, x._5, x._6))
f1: org.apache.spark.rdd.RDD[(String, (String, Int, Int))] = MapPartitionsRDD[134] at map at <console>:31

scala>
```

```
scala> val f2 = transportRDD.map(x => x._1 -> x._2)
f2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[135] at map at <console>:31

scala>
```

```
scala> val jtwo = f1.join(f2)
jtwo: org.apache.spark.rdd.RDD[(String, ((String, Int, Int), Int))] = MapPartitionsRDD[138] at join at <console>:39

scala>
```

```
scala> val trans = jtwo.map(x => (x._2._1._1, x._2._1._3) -> (x._2._1._2 * x._2._2))
trans: org.apache.spark.rdd.RDD[((String, Int), Int)] = MapPartitionsRDD[139] at map at <console>:41

scala>
```

```
scala> val transMap = trans.map(x => (x._1._1) -> x._2)
transMap: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[140] at map at <console>:43

scala>
```

```
scala> val j = filterAgeGroup.join(transMap)
j: org.apache.spark.rdd.RDD[(String, (String, Int))] = MapPartitionsRDD[143] at join at <console>:51

scala>
```

```
scala> val finalTrans = j.map(x => (x._2._1) -> (x._2._2))
finalTrans: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[144] at map at <console>:53

scala>
```

```
scala> val grp = finalTrans.groupByKey()
grp: org.apache.spark.rdd.RDD[(String, Iterable[Int])] = ShuffledRDD[145] at groupByKey at <console>:55

scala>
```

Output:

Displaying amount spent by each age-group, every year in travelling

```
scala> val output= grp.map(x => x._1 -> x._2.sum).collect
output: Array[(String, Int)] = Array((20-35,442000), (>35,306000), (<20,340000))

scala>
```