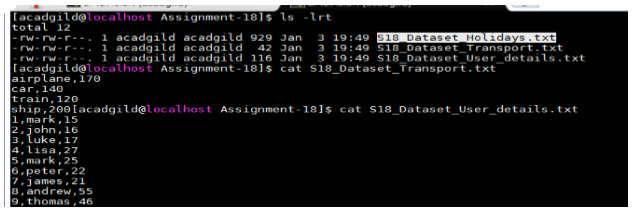
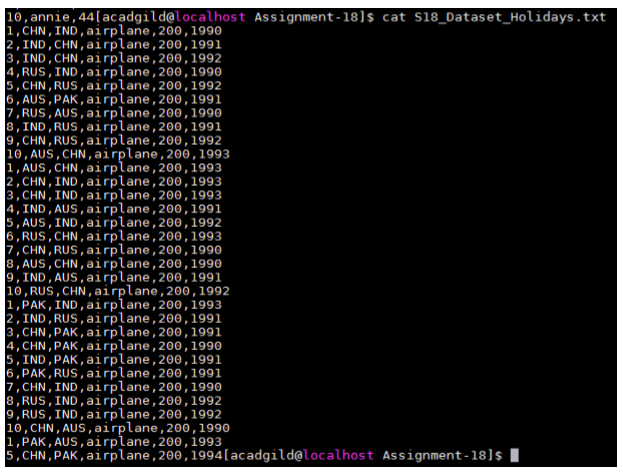
**Assignment 18.1**

Below is the dataset which we will be using for this Assignment in all problems. It has been kept in local file system:-

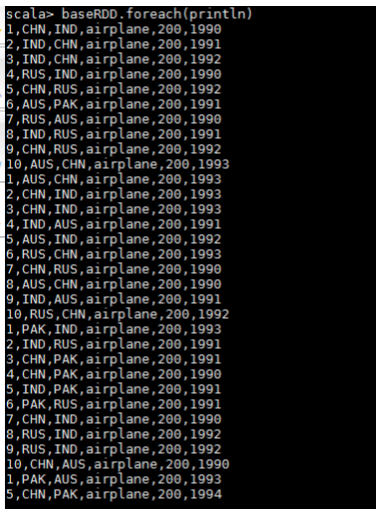




DataSet is uploaded in baseRDD:-

* val baseRDD = sc.textFile("/home/acadgild/Assignment18/S18\_Dataset\_Holidays.txt")
* import org.apache.spark.storage.StorageLevel
* baseRDD.persist(StorageLevel.MEMORY\_ONLY)





Problem Statement:-

1. What is the distribution of the total number of air-travelers per year

2. What is the total air distance covered by each user per year

3. Which user has travelled the largest distance till date

4. What is the most preferred destination for all users.

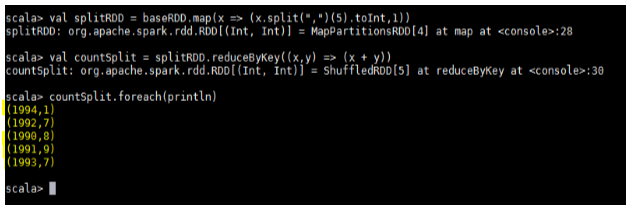
Solution:-

* The distribution of the total number of air-travelers per year

Below is the code used:-

* val splitRDD = baseRDD.map(x => (x.split(",")(5).toInt,1))
* val countSplit = splitRDD.reduceByKey((x,y) => (x + y))
* countSplit.foreach(println)

Output:-

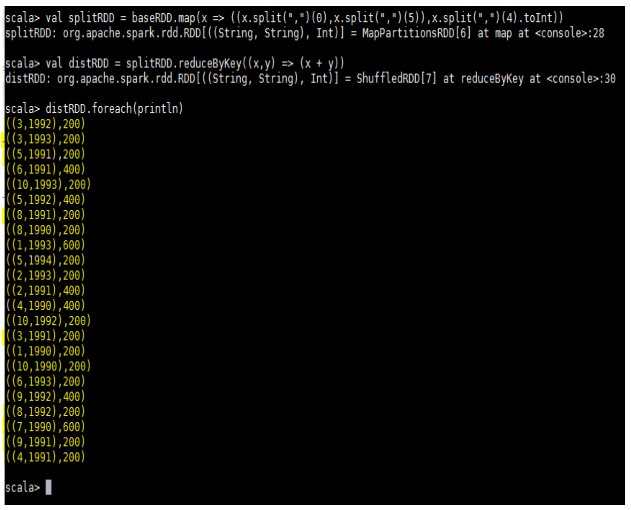


* The total air distance covered by each user per year

Below is the code used:-

* val splitRDD = baseRDD.map(x => ((x.split(",")(0),x.split(",")(5)),x.split(",")(4).toInt))
* val distRDD = splitRDD.reduceByKey((x,y) => (x + y))
* distRDD.foreach(println)

Output:-

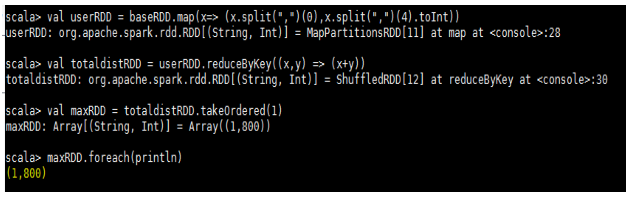


* User has travelled the largest distance till date

Below is the code used:-

* val userRDD = baseRDD.map(x=> (x.split(",")(0),x.split(",")(4).toInt))
* val totaldistRDD = userRDD.reduceByKey((x,y) => (x+y))
* val maxRDD = totaldistRDD.takeOrdered(1)
* maxRDD.foreach(println)

Output:-



This shows that Mark has travelled the largest distance till date.

* The most preferred destination for all users.

Below is the code used:-

* val destRDD = baseRDD.map(x => (x.split(",")(2),1))
* val destreduceRDD = destRDD.reduceByKey((x,y) => (x + y))
* val maxRDD = destreduceRDD.takeOrdered(1)(Ordering[Int].reverse.on(\_.\_2))
* maxRDD.foreach(println)

Output:-

