Assignment 20

Task 1

Dataset used
S20_Dataset_Holidays.txt

```
[acadgild@localhost ~]$ hadoop fs -put S20_Dataset_Holidays.txt
18/04/07 10:25:42 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -cat S20_Dataset_Holidays.txt
18/04/07 10:26:14 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
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
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

Dataset used,

S20_Dataset_Transport.txt

acadgild@localhost:~

```
[acadgild@localhost ~]$ hadoop fs -cat S20_Dataset_Transport.txt
18/04/08 11:26:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library
airplane,170
car,140
train,120
ship,200
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ █
```

Dataset used,

S20_Dataset_User_details.txt

acadgild@localhost:~

```
[acadgild@localhost ~]$ hadoop fs -cat S20_Dataset_User_details.txt
18/04/08 11:27:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library
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
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ █
```

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

```
val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
val baseRDD1 = baseRDD.map(x=>(x.split(",")(5).toInt,1))
val no_air_travelers =
baseRDD1.reduceByKey((x,y)=>(x+y)).foreach(println)
```

acadgild@localhost:~

```
scala> val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.
txt MapPartitionsRDD[6] at textFile at <console>:25

scala> val baseRDD1 = baseRDD.map(x=>(x.split(",")(5).toInt,1))
baseRDD1: org.apache.spark.rdd.RDD[(Int, Int)] = MapPartitionsRDD[7] at map at <
console>:27

scala> val no_air_travelers = baseRDD1.reduceByKey((x,y)=>(x+y)).foreach(println
)
(1994,1)
(1992,7)
(1990,8)
(1991,9)
(1993,7)
no_air_travelers: Unit = ()

scala>
```

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

```
val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
val baseRDD1 = baseRDD.map(x =>
((x.split(",")(0),x.split(",")(5)),x.split(",")(4).toInt))
val distance_user = baseRDD1.reduceByKey((x,y) => (x +
y)).foreach(println)
```

```
scala> val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.
txt MapPartitionsRDD[26] at textFile at <console>:25

scala> val baseRDD1 = baseRDD.map(x => ((x.split(",")(0),x.split(",")(5)),x.spli
t(",")(4).toInt))
baseRDD1: org.apache.spark.rdd.RDD[((String, String), Int)] = MapPartitionsRDD[2
7] at map at <console>:27

scala> val distance_user = baseRDD1.reduceByKey((x,y) => (x + y)).foreach(printl
n)
((3,1992),200)
((3,1993),200)
((5,1991),200)
((6,1991),400)
((10,1993),200)
((5,1992),400)
((8,1991),200)
((8,1990),200)
((1,1993),600)
((5,1994),200)
((2,1993),200)
((2,1991),400)
((4,1990),400)
((10,1992),200)
((3,1991),200)
((1,1990),200)
((10,1990),200)
((6,1993),200)
((9,1992),400)
((8,1992),200)
((7,1990),600)
((9,1991),200)
((4,1991),200)
distance_user: Unit = ()

scala>
```

3) Which user has travelled the largest distance till date

val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")

val baseRDD1 = baseRDD.map(x => (x.split(",")(0),x.split(",")(4).toInt))

val largest_dist = baseRDD1.reduceByKey((x,y)=>(x+y)).takeOrdered(1)

```
scala> val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.txt MapPartitionsRDD[30] at textFile at <console>:25

scala> val baseRDD1 = baseRDD.map(x => (x.split(",")(0),x.split(",")(4).toInt))
baseRDD1: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[31] at map at <console>:27

scala> val largest_dist = baseRDD1.reduceByKey((x,y)=>(x+y)).takeOrdered(1)
largest_dist: Array[(String, Int)] = Array((1,800))

scala>
```

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

val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
val baseRDD1 = baseRDD.map(x => (x.split(",")(2),1))
val dest = baseRDD1.reduceByKey((x,y)=>(x+y))
dest.foreach(println)
val dest =
baseRDD1.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].rev
erse.on(_._2))

```
scala> val baseRDD = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.txt MapPartitionsRDD[72] at textFile at <console>:25

scala> val baseRDD1 = baseRDD.map(x => (x.split(",")(2),1))
baseRDD1: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[73] at map at <console>:27

scala> val dest = baseRDD1.reduceByKey((x,y)=>(x+y))
dest: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[74] at reduceByKey at <console>:29

scala> dest.foreach(println)
(CHN,7)
(IND,9)
(PAK,5)
(RUS,6)
(AUS,5)

scala> val dest = baseRDD1.reduceByKey((x,y)=>(x+y)).takeOrdered(1)(Ordering[Int].reverse.on(_._2))
dest: Array[(String, Int)] = Array((IND,9))

scala>
```

5) Which route is generating the most revenue per year

we are loading the dataset into the spark context,
val baseRDD1 =
sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
val baseRDD2 =
sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")
val baseRDD3 =
sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")

We are loading the dataset's in the name of holidays, transport and user RDD's.

val holidays =
baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.spl
it(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))
val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))
val user =
baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toIn
t))

```
scala> val baseRDD1 = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD1: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.txt MapPartitionsRDD[15] at textFile at <console>:24

scala> val baseRDD2 = sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")
baseRDD2: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Transport.txt MapPartitionsRDD[17] at textFile at <console>:24

scala> val baseRDD3 = sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")
baseRDD3: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_User_details.txt MapPartitionsRDD[19] at textFile at <console>:24

scala> val holidays =
     | baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))
holidays: org.apache.spark.rdd.RDD[(Int, String, String, String, Int, Int)] = MapPartitionsRDD[20] at map at <console>:27

scala> val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))
transport: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[21] at map at <console>:26

scala> val user = baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))
user: org.apache.spark.rdd.RDD[(Int, String, Int)] = MapPartitionsRDD[22] at map at <console>:26

scala>

scala>
```

val holidaysmap = holidays.map(x=>x._4->(x._2,x._5,x._6))
val transportmap = transport.map(x=>x._1->x._2)
val join1 = holidaysmap.join(transportmap)

```
val route = join1.map(x=>(x._2._1._1->x._2._1._3)  ->
(x._2._1._2*x._2._2))
val revenue = route.groupByKey().map(x=>x._2.sum->x._1)
val routemostrevenue = revenue.sortByKey(false).first()
```

```
scala> val holidaysmap = holidays.map(x=>x._4->(x._2,x._5,x._6))
holidaysmap: org.apache.spark.rdd.RDD[(String, (String, Int, Int))] = MapPartitionsRDD[42] at map at <console>:28

scala> val transportmap = transport.map(x=>x._1->x._2)
transportmap: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[43] at map at <console>:28

scala> val join1 = holidaysmap.join(transportmap)
join1: org.apache.spark.rdd.RDD[(String, ((String, Int, Int), Int))] = MapPartitionsRDD[46] at join at <console>:36

scala> val route = join1.map(x=>(x._2._1._1->x._2._1._3)->(x._2._1._2*x._2._2))
route: org.apache.spark.rdd.RDD[((String, Int), Int)] = MapPartitionsRDD[47] at map at <console>:38

scala> val revenue = route.groupByKey().map(x=>x._2.sum->x._1)
revenue: org.apache.spark.rdd.RDD[(Int, (String, Int))] = MapPartitionsRDD[49] at map at <console>:40

scala> val routemostrevenue = revenue.sortByKey(false).first()
routemostrevenue: (Int, (String, Int)) = (204000,(IND,1991))

scala>
```

6) **What is the total amount spent by every user on air-travel per year**

we are loading the dataset into the spark context,
val baseRDD1 =
sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
val baseRDD2 =
sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")
val baseRDD3 =
sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")

We are loading the dataset's in the name of holidays, transport and user RDD's.

val holidays =
baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))
val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))

val user =
baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))

```
acadgild@localhost:~                                                                           –

scala> val baseRDD1 = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD1: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.txt MapPartitionsRDD[15] at textFile at <console>:24

scala> val baseRDD2 = sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")
baseRDD2: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Transport.txt MapPartitionsRDD[17] at textFile at <console>:24

scala> val baseRDD3 = sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")
baseRDD3: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_User_details.txt MapPartitionsRDD[19] at textFile at <console>:24

scala> val holidays =
     | baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))
holidays: org.apache.spark.rdd.RDD[(Int, String, String, String, Int, Int)] = MapPartitionsRDD[20] at map at <console>:27

scala> val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))
transport: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[21] at map at <console>:26

scala> val user = baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))
user: org.apache.spark.rdd.RDD[(Int, String, Int)] = MapPartitionsRDD[22] at map at <console>:26

scala>

scala>
```

val userMap = holidays.map(x => x._4 -> (x._1,x._5,x._6))
val amount = userMap.join(transportmap)
val spend = amount.map(x => (x._2._1._1, x._2._1._3) -> (x._2._1._2 * x._2._2))
val total = spend.groupByKey().map(x => x._1 -> x._2.sum)

```
acadgild@localhost:~

scala> val userMap = holidays.map(x => x._4 -> (x._1,x._5,x._6))
userMap: org.apache.spark.rdd.RDD[(String, (Int, Int, Int))] = MapPartitionsRDD[66] at map at <console>:28

scala> val amount = userMap.join(transportmap)
amount: org.apache.spark.rdd.RDD[(String, ((Int, Int, Int), Int))] = MapPartitionsRDD[69] at join at <console>:36

scala> val spend = amount.map(x => (x._2._1._1, x._2._1._3) -> (x._2._1._2 * x._2._2))
spend: org.apache.spark.rdd.RDD[((Int, Int), Int)] = MapPartitionsRDD[70] at map at <console>:38

scala> val total = spend.groupByKey().map(x => x._1 -> x._2.sum)
total: org.apache.spark.rdd.RDD[((Int, Int), Int)] = MapPartitionsRDD[72] at map at <console>:40

scala> total.foreach(println)
((2,1993),34000)
((6,1993),34000)
((10,1993),34000)
((10,1992),34000)
((2,1991),68000)
((4,1990),68000)
((10,1990),34000)
((5,1992),68000)
((4,1991),34000)
((1,1993),102000)
((9,1992),68000)
((5,1991),34000)
((3,1993),34000)
((1,1990),34000)
((8,1990),34000)
((7,1990),102000)
((6,1991),68000)
((5,1994),34000)
((3,1991),34000)
((9,1991),34000)
((3,1992),34000)
((8,1991),34000)
((8,1992),34000)

scala>
```

7) Considering age groups of < 20 , 20-35, 35 > ,Which age group is travelling the most every year.

we are loading the dataset into the spark context,
val baseRDD1 = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")

val baseRDD2 = sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")

val baseRDD3 = sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")

We are loading the dataset's in the name of holidays, transport and user RDD's.

val holidays =

baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))

val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))

val user =
baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))

```
scala> val baseRDD1 = sc.textFile("/user/acadgild/S20_Dataset_Holidays.txt")
baseRDD1: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Holidays.txt MapPartitionsRDD[15] at textFile at <console>:24

scala> val baseRDD2 = sc.textFile("/user/acadgild/S20_Dataset_Transport.txt")
baseRDD2: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_Transport.txt MapPartitionsRDD[17] at textFile at <console>:24

scala> val baseRDD3 = sc.textFile("/user/acadgild/S20_Dataset_User_details.txt")
baseRDD3: org.apache.spark.rdd.RDD[String] = /user/acadgild/S20_Dataset_User_details.txt MapPartitionsRDD[19] at textFile at <console>:24

scala> val holidays =
     | baseRDD1.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2),x.split(",")(3),x.split(",")(4).toInt,x.split(",")(5).toInt))
holidays: org.apache.spark.rdd.RDD[(Int, String, String, String, Int, Int)] = MapPartitionsRDD[20] at map at <console>:27

scala> val transport = baseRDD2.map(x=> (x.split(",")(0),x.split(",")(1).toInt))
transport: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[21] at map at <console>:26

scala> val user = baseRDD3.map(x=>(x.split(",")(0).toInt,x.split(",")(1),x.split(",")(2).toInt))
user: org.apache.spark.rdd.RDD[(Int, String, Int)] = MapPartitionsRDD[22] at map at <console>:26

scala>

scala>
```

val AgeMap = user.map(x=>x._1->

{

if(x._3<20)

 "20"

else if(x._3>35)

 "35"

 else "20-35"

 })

val UserID = holidays.map(x => x._1 -> 1)

val joinMap1 = AgeMap.join(UserID)

val joinMap2 = joinMap1.map(x => x._2._1 -> x._2._2)

val groupKey = joinMap2.groupByKey.map(x => x._1 -> x._2.sum)

val mostGroup = groupKey.sortBy(x => -x._2).first()

```
scala> val AgeMap = user.map(x=>x._1->
     | {
     | if(x._3<20)
     | "20"
     | else if(x._3>35)
     | "35"
     | else "20-35"
     | })
AgeMap: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[104] at map at <console>:28

scala> val UserID = holidays.map(x => x._1 -> 1)
UserID: org.apache.spark.rdd.RDD[(Int, Int)] = MapPartitionsRDD[105] at map at <console>:28

scala> val joinMap1 = AgeMap.join(UserID)
joinMap1: org.apache.spark.rdd.RDD[(Int, (String, Int))] = MapPartitionsRDD[108] at join at <console>:36

scala> val joinMap2 = joinMap1.map(x => x._2._1 -> x._2._2)
joinMap2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[109] at map at <console>:38

scala> val groupKey = joinMap2.groupByKey.map(x => x._1 -> x._2.sum)
groupKey: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[111] at map at <console>:40

scala> val mostGroup = groupKey.sortBy(x => -x._2).first()
mostGroup: (String, Int) = (20-35,13)

scala>
```