Session 23 Assignment 2

Problem Statement 1

Find out the top 5 most visited destinations.

val delayed_flights = sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFlights.csv")

val mapping = delayed_flights.map(x => x.split(",")).map(x => (x(18),1)).filter(x => x._1!=null).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(5)

```
scala> val delayed_flights=sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFl
ights.csv")
delayed_flights: org.apache.spark.rdd.RDD[String] = file:///home/bigdata/deepak/docs/Acadgild/Session23Assignmen
t2/DelayedFlights.csv MapPartitionsRDD[18] at textFile at <console>:24

scala> val mapping = delayed_flights.map(x => x.split(",")).map(x => (x(18),1)).filter(x =>
     | x._1!=null).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(5)
mapping: Array[(String, Int)] = Array((ORD,108984), (ATL,106898), (DFW,70657), (DEN,63003), (LAX,59969))

scala>
```

Problem Statement 2

Which month has seen the most number of cancellations due to bad weather?

val delayed_flights = sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFlights.csv")

val canceled = delayed_flights.map(x => x.split(",")).filter(x => ((x(22).equals("1"))&& (x(23).equals("B")))).map(x => (x(2),1)).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(1)

```
scala> val delayed_flights=sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFl
ights.csv")
delayed_flights: org.apache.spark.rdd.RDD[String] = file:///home/bigdata/deepak/docs/Acadgild/Session23Assignmen
t2/DelayedFlights.csv MapPartitionsRDD[29] at textFile at <console>:24

scala> val canceled = delayed_flights.map(x => x.split(",")).filter(x => ((x(22).equals("1"))&&(x(23).equals("B"
)))).map(x => (x(2),1)).reduceByKey(_+_).map(x =>(x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(1)
canceled: Array[(String, Int)] = Array((12,250))

scala>
```

Problem Statement 3

Top ten origins with the highest AVG departure delay

val delayed_flights = sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFlights.csv")

val avg = delayed_flights.map(x => x.split(",")).map(x => (x(17),x(16).toDouble)).mapValues((_, 1)).reduceByKey((x, y) => (x._1 + y._1, x._2 + y._2)).mapValues{ case (sum, count) => (1.0 * sum)/count}.map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(10)

Problem Statement 4

Which route (origin & destination) has seen the maximum diversion?

val delayed_flights = sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFlights.csv")

val diversion = delayed_flights.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x => ((x(17)+","+x(18)),1)).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(10).foreach(println)

```
scala> val delayed_flights=sc.textFile("file:///home/bigdata/deepak/docs/Acadgild/Session23Assignment2/DelayedFl
ights.csv")
delayed_flights: org.apache.spark.rdd.RDD[String] = file:///home/bigdata/deepak/docs/Acadgild/Session23Assignmen
t2/DelayedFlights.csv MapPartitionsRDD[93] at textFile at <console>:24

scala> val diversion = delayed_flights.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x =>
     | ((x(17)+","+x(18)),1)).reduceByKey(_+_).map(x => (x._2,x._1)).sortByKey(false).map(x =>
     | (x._2,x._1)).take(10).foreach(println)
(ORD,LGA,39)
(DAL,HOU,35)
(DFW,LGA,33)
(ATL,LGA,32)
(SLC,SUN,31)
(ORD,SNA,31)
(MIA,LGA,31)
(BUR,JFK,29)
(HRL,HOU,28)
(BUR,DFW,25)
diversion: Unit = ()

scala>
```