Assignment 21.2

Aviation data analysis

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights appears in DOT's monthly Air Travel Consumer Report, published about 30 days after the month's end, as well as in summary tables posted on this website. Summary statistics and raw data are made available to the public at the time the Air Travel Consumer Report is released.

Downloaded the dataset of DelayedFlights.csv and uploaded in the local file system **/home/acadgild/Sumona**

The dataset has 29 columns as follows :

Name	Description
1	Year 1987-2008
2	Month 1-12
3	DayofMonth 1-31
4	DayOfWeek 1 (Monday) - 7 (Sunday)
5	DepTime actual departure time (local, hhmm)
6	CRSDepTime scheduled departure time (local, hhmm)
7	ArrTime actual arrival time (local, hhmm)
8	CRSArrTime scheduled arrival time (local, hhmm)
9	UniqueCarrier unique carrier code
10	FlightNum flight number
11	TailNum plane tail number
12	ActualElapsedTime in minutes
13	CRSElapsedTime in minutes
14	AirTime in minutes
15	ArrDelay arrival delay, in minutes

```
16      DepDelay        departure delay, in minutes

17      Origin  origin IATA airport code

18      Dest    destination IATA airport code

19      Distance        in miles

20      TaxiIn  taxi in time, in minutes

21      TaxiOut taxi out time in minutes

22      Cancelled       was the flight cancelled?

23      CancellationCode        reason for cancellation (A = carrier, B =
weather, C = NAS, D = security)

24      Diverted        1 = yes, 0 = no

25      CarrierDelay    in minutes

26      WeatherDelay    in minutes

27      NASDelay        in minutes

28      SecurityDelay   in minutes

29      LateAircraftDelay       in minutes
```

Problem Statement 1:

Find out the top 5 most visited destinations.

First we will read the DelayedFlight.csv file and then split the columns and take the column 18.

After splitting, we will combine the occurrence of the value and take the top 5 values.

```
val delayed_flights = sc.textFile("/home/acadgild/sumona/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)
```

Output:

```
scala> val delayed_flights = sc.textFile("/home/acadgild/sumona/DelayedFlights.csv")
delayed_flights: org.apache.spark.rdd.RDD[String] = /home/acadgild/sumona/DelayedFlights.csv MapPartitionsRDD[57] at textFile at <console>:24

scala>

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).m
ap(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?

Here we will split the columns to get the columns 22 and 23 and combine the occurrence values and get the output.

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)

Output:

```
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

We will split the RDD and select the column 16 and 17, by using the reduceByKey we will combine the occurrence of the value and using the case, we are getting the average value.

val avg = delayed_flights.map(x => x.split(",")(17),x.split(",")(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)

Output:

```
scala> 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)).mapV
alues{ case (sum, count) => (1.0 * sum)/count}.map(x => (x._2,x._1)).sortByKey(false).map(x => (x._2,x._1)).take(10)
avg: Array[(String, Double)] = Array((CMX,154.95238095238096), (PLN,106.83333333333333), (SPI,86.05932203389831), (MOT,79.98571428571428), (ACY,79.3103448275862)
, (MQT,78.9776119402985), (HHH,75.55319148936171), (MBS,74.82413793103449), (ABI,74.80188679245283), (ACK,74.38461538461539))

scala>
```

Problem Statement 4

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

We will split the RDD and select the column 24 and by using the reduceBykey we will combine the occurrence of the value and the add the numbers.

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 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>
```