Assignment 21.2:

Problem Statement:

Implement the below blog at your end and send the complete documentation.

https://drive.google.com/file/d/0B_Qjau8wv1KoUThzZ24tT1NsZGs/view?usp=sharing

Solution:

The dataset is downloaded and placed at below location:

```
[acadgild@localhost spark]$ pwd
/home/acadgild/spark
[acadgild@localhost spark]$ head -10 DelayedFlights.csv
,Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,A
irTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,Secur
ityDelay,LateAircraftDelay
0,2008,1,3,4,2003.0,1955,2211.0,2225,WN,335,N712SW,128.0,150.0,116.0,-14.0,8.0,IAD,TPA,810,4.0,8.0,0,N,0,,,,,
1,2008,1,3,4,754.0,735,1002.0,1000,WN,3231,N772SW,128.0,145.0,113.0,2.0,19.0,IAD,TPA,810,5.0,10.0,0,N,0,,,,,
2,2008,1,3,4,628.0,620,804.0,750,WN,448,N428WN,96.0,90.0,76.0,14.0,8.0,IND,BWI,515,3.0,17.0,0,N,0,,,,,
4,2008,1,3,4,1829.0,1755,1959.0,1925,WN,3920,N464WN,90.0,90.0,77.0,34.0,34.0,IND,BWI,515,3.0,10.0,0,N,0,2.0,0.0,0.0,0.0,32.0
5,2008,1,3,4,1940.0,1915,2121.0,2110,WN,378,N726SW,101.0,115.0,87.0,11.0,25.0,IND,JAX,688,4.0,10.0,0,N,0,,,,,
6,2008,1,3,4,1937.0,1830,2037.0,1940,WN,509,N763SW,240.0,250.0,230.0,57.0,67.0,IND,LAS,1591,3.0,7.0,0,N,0,10.0,0.0,0.0,0.0,47.0
10,2008,1,3,4,706.0,700,916.0,915,WN,100,N690SW,130.0,135.0,106.0,1.0,6.0,IND,MCO,828,5.0,19.0,0,N,0,,,,,
11,2008,1,3,4,1644.0,1510,1845.0,1725,WN,1333,N334SW,121.0,135.0,107.0,80.0,94.0,IND,MCO,828,6.0,8.0,0,N,0,8.0,0.0,0.0,0.0,72.0
15,2008,1,3,4,1029.0,1020,1021.0,1010,WN,2272,N263WN,52.0,50.0,37.0,11.0,9.0,IND,MDW,162,6.0,9.0,0,N,0,,,,,
[acadgild@localhost spark]$
```

Problem Statement 1

Find out the top 5 most visited destinations.

val delayed_flights = sc.textFile("/home/acadgild/spark/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("/home/acadgild/spark/DelayedFlights.csv")
delayed_flights: org.apache.spark.rdd.RDD[String] = /home/acadgild/spark/DelayedFlights.csv MapPartitionsRDD[1] 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 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 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 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)

```
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?

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>
```