**Aviation data analysis**

Problem Statement 1

**Find out the top 5 most visited destinations.**

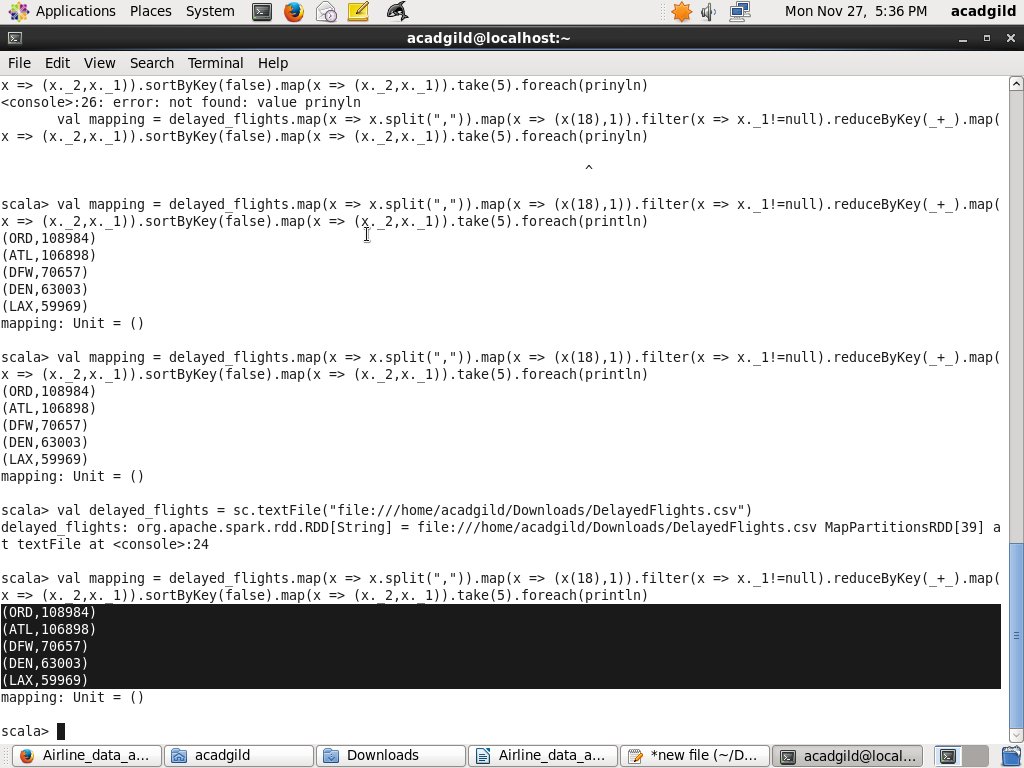
**Sol:**

1. val delayed\_flights = sc.textFile("file:///home/acadgild/Downloads/DelayedFlights.csv")

2. 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).foreach(println)

Output:



Problem Statement 2

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

**Sol:**

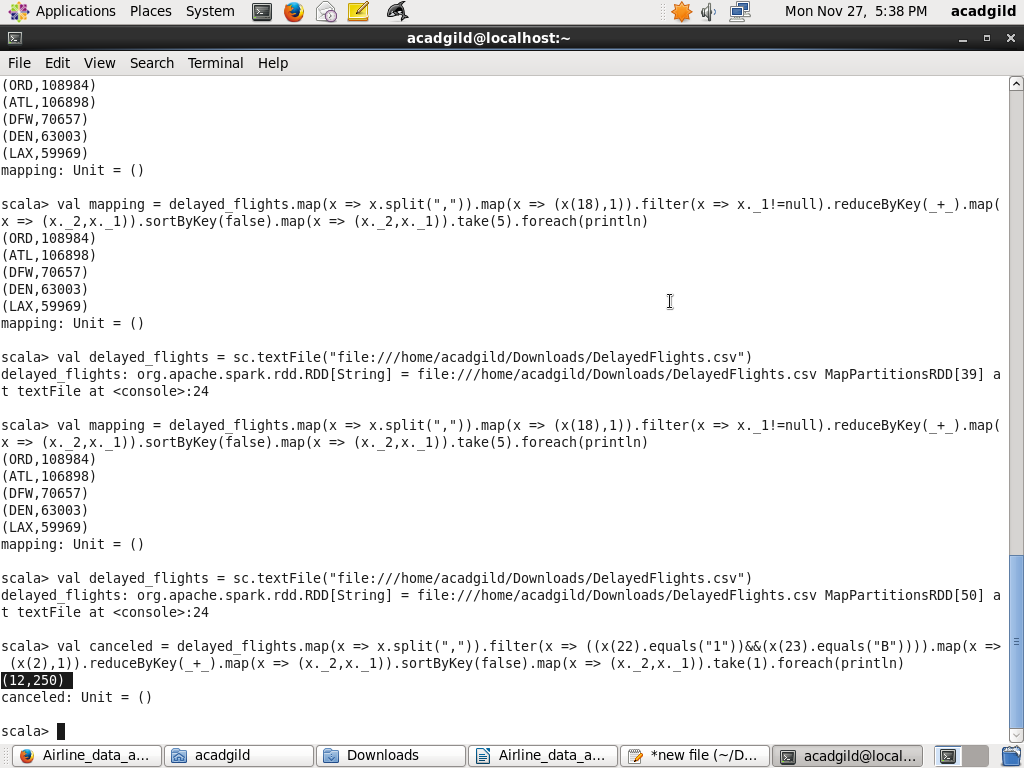
1.val delayed\_flights = sc.textFile("file:///home/acadgild/Downloads/DelayedFlights.csv")

2. 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).foreach(prntln)

Output:



Problem Statement 4

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

**Sol:**

1. val delayed\_flights = sc.textFile("file:///home/acadgild/Downloads/DelayedFlights.csv")

2.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)

Output:

