

CIS 5200 Term Project Tutorial

Group 6

Authors: Anupam Sahay, Krithy, Nitesh Kamboj, Sohong Chakraborty, Zeeshan Khan

Date: 12/04/2017

Lab Tutorial

Flight Delay Data Analysis from 1987 to 2008 in Apache Pig using IBM BigInsights

Objective:

In this tutorial you will fetch, analyse and visualize Flight Delay Data. Thus,

- You will learn how to download data from <http://stat-computing.org/dataexpo/2009/the-data.html> (Statistical Computing Statistical Graphics)
- Then you will learn how to upload it to HDFS.
- You will figure out how to manipulate and analyze Flight Delay Data in HDFS using Apache Pig.
- You will also learn how to visualize the result in Tableau.

Introduction:

With the ever-expanding field of aviation, it has become imperative to maintain a record of the flight delays of commercial airlines. Airline flight delays have come under increased scrutiny lately in the popular press, with the Federal Aviation Administration data revealing that airline on-time performance was at its worst level in 21 years in 2007. Flight delays have been attributed to several causes such as weather conditions, airport congestion, airspace congestion, use of smaller aircraft & by airlines, etc. In this lab, you are going to examine a dataset provided by the United States Department of Transportation, Bureau of Transportation Statistics, containing data from the years (1987-2008). You will learn:

- Analyze data to determine which Airline Carrier was the most popular in a given year.
- Analyze data to determine outbound flights from top 20 airports on departure basis.
- Analyze data to determine total flights from top 20 airports on monthly traffic basis.
- Analyze data to determine total flight originating from Los Angeles, LAX to other airports.

- Analyze data to determine Carrier specific average delay.
- Analyze data to determine longest flight between two airports by Air Time.
- Visualize in Tableau

Pre-requisites:

- Tableau should be installed on your system
- IBM Bluemix account

Outline:

- Download the data
- Upload the data files into HDFS
- Further reading: Pig

Download the data:

Download the driver data file using the following shell command at your BigInsights terminal

```
$ wget http://stat-computing.org/dataexpo/2009/1987.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1988.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1989.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1990.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1991.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1992.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1993.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1994.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1995.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1996.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1997.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1998.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/1999.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2000.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2001.csv.bz2
```

```
$ wget http://stat-computing.org/dataexpo/2009/2002.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2003.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2004.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2005.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2006.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2007.csv.bz2
$ wget http://stat-computing.org/dataexpo/2009/2008.csv.bz2
```

Here what your output should look like

```
bi-hadoop-prod-4021.bi.services.us-south.bluemix.net - PuTTY
100%[=====] 112,450,321 31.9M/s  in 4.3s
2017-12-05 07:59:40 (24.8 MB/s) - "2005.csv.bz2" saved [112450321/112450321]
-bash-4.1$ 
-bash-4.1$ wget http://stat-computing.org/dataexpo/2009/2006.csv.bz2
--2017-12-05 07:59:40-- http://stat-computing.org/dataexpo/2009/2006.csv.bz2
Resolving stat-computing.org... 52.218.193.11
Connecting to stat-computing.org|52.218.193.11|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 115019195 (110M) [application/x-bzip2]
Saving to: "2006.csv.bz2"

100%[=====] 115,019,195 9.32M/s  in 21s
2017-12-05 08:00:01 (5.33 MB/s) - "2006.csv.bz2" saved [115019195/115019195]
-bash-4.1$ 
-bash-4.1$ wget http://stat-computing.org/dataexpo/2009/2007.csv.bz2
--2017-12-05 08:00:01-- http://stat-computing.org/dataexpo/2009/2007.csv.bz2
Resolving stat-computing.org... 52.218.208.99
Connecting to stat-computing.org|52.218.208.99|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113249243 (110M) [application/x-bzip2]
Saving to: "2007.csv.bz2"

100%[=====] 113,249,243 18.7M/s  in 7.9s
2017-12-05 08:00:09 (14.7 MB/s) - "2007.csv.bz2" saved [113249243/113249243]
-bash-4.1$ 
-bash-4.1$ wget http://stat-computing.org/dataexpo/2009/2008.csv.bz2
--2017-12-05 08:00:09-- http://stat-computing.org/dataexpo/2009/2008.csv.bz2
Resolving stat-computing.org... 52.218.200.211
Connecting to stat-computing.org|52.218.200.211|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113753229 (108M) [application/x-bzip2]
Saving to: "2008.csv.bz2"

100%[=====] 113,753,229 15.5M/s  in 19s  =====> 108,702,778 14.2M/s eta 1s
2017-12-05 08:00:28 (5.61 MB/s) - "2008.csv.bz2" saved [113753229/113753229]
-bash-4.1$ 
```

Upload the data into HDFS

Run the following shell commands to upload the data

```
$ hdfs dfs -mkdir flight_delay
$ hdfs dfs -put 1987.csv.bz2 flight_delay
$ hdfs dfs -put 1988.csv.bz2 flight_delay
$ hdfs dfs -put 1989.csv.bz2 flight_delay
$ hdfs dfs -put 1990.csv.bz2 flight_delay
$ hdfs dfs -put 1991.csv.bz2 flight_delay
```

```

$ hdfs dfs -put 1992.csv.bz2 flight_delay
$ hdfs dfs -put 1993.csv.bz2 flight_delay
$ hdfs dfs -put 1994.csv.bz2 flight_delay
$ hdfs dfs -put 1995.csv.bz2 flight_delay
$ hdfs dfs -put 1996.csv.bz2 flight_delay
$ hdfs dfs -put 1997.csv.bz2 flight_delay
$ hdfs dfs -put 1998.csv.bz2 flight_delay
$ hdfs dfs -put 1999.csv.bz2 flight_delay
$ hdfs dfs -put 2000.csv.bz2 flight_delay
$ hdfs dfs -put 2001.csv.bz2 flight_delay
$ hdfs dfs -put 2002.csv.bz2 flight_delay
$ hdfs dfs -put 2003.csv.bz2 flight_delay
$ hdfs dfs -put 2004.csv.bz2 flight_delay
$ hdfs dfs -put 2005.csv.bz2 flight_delay
$ hdfs dfs -put 2006.csv.bz2 flight_delay
$ hdfs dfs -put 2007.csv.bz2 flight_delay
$ hdfs dfs -put 2008.csv.bz2 flight_delay

```

Navigate to flight_delay to make sure if it has the files uploaded

```
$ hdfs dfs -ls flight_delay
```

```

$ hdfs dfs -ls flight_delay
Found 22 items
-rw-r--r-- 3 schakrakr.hdfs 12652442 2017-12-05 08:31 flight_delay/1987.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 49499025 2017-12-05 08:31 flight_delay/1988.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 49303815 2017-12-05 08:31 flight_delay/1989.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 49203322 2017-12-05 08:31 flight_delay/1990.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 49877448 2017-12-05 08:31 flight_delay/1991.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 50040946 2017-12-05 08:31 flight_delay/1992.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 50111774 2017-12-05 08:31 flight_delay/1993.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 51139887 2017-12-05 08:31 flight_delay/1994.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 74801752 2017-12-05 08:31 flight_delay/1995.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 75887707 2017-12-05 08:31 flight_delay/1996.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 76705687 2017-12-05 08:31 flight_delay/1997.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 77623574 2017-12-05 08:31 flight_delay/1998.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 79449438 2017-12-05 08:31 flight_delay/1999.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 82537924 2017-12-05 08:31 flight_delay/2000.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 83478700 2017-12-05 08:31 flight_delay/2001.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 75907218 2017-12-05 08:31 flight_delay/2002.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 95326001 2017-12-05 08:31 flight_delay/2003.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 110825331 2017-12-05 08:32 flight_delay/2004.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 112450381 2017-12-05 08:33 flight_delay/2005.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 115019186 2017-12-05 08:33 flight_delay/2006.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 131049249 2017-12-05 08:32 flight_delay/2007.csv.bz2
-rw-r--r-- 3 schakrakr.hdfs 133763229 2017-12-05 08:28 flight_delay/2008.csv.bz2
$ hdfs dfs -ls flight_delay

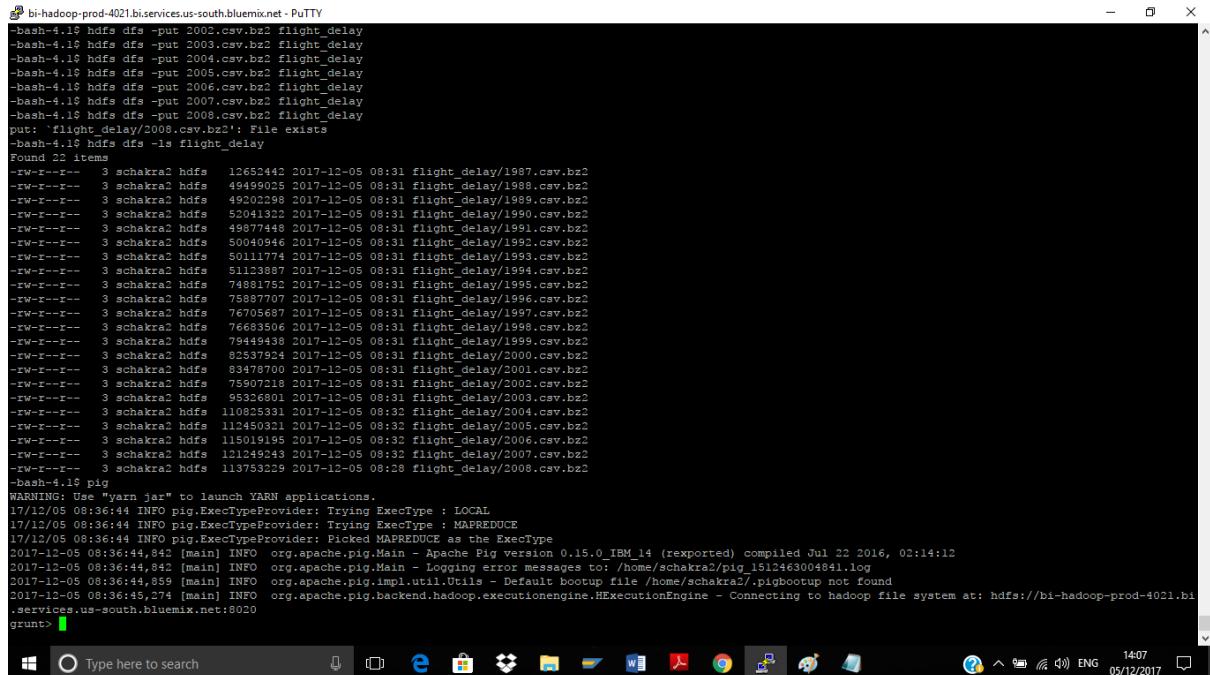
```

Create Tables for the Data Using Pig

Open the Pig interface in your terminal

Run the following command

```
$ pig
```



```
bin-hadoop-prod-4021.bi.services.us-south.bluemix.net - PuTTY
-bash-4.1$ hdfs dfs -put 2002.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2003.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2004.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2005.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2006.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2007.csv.bz2 flight_delay
-bash-4.1$ hdfs dfs -put 2008.csv.bz2 flight_delay
put: `flight_delay/2008.csv.bz2': File exists
-bash-4.1$ hdfs dfs -ls flight_delay
Found 22 items
-rw-r--r-- 3 schakra2 hdfs 12652442 2017-12-05 08:31 flight_delay/1987.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 49499025 2017-12-05 08:31 flight_delay/1988.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 49702298 2017-12-05 08:31 flight_delay/1989.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 52041322 2017-12-05 08:31 flight_delay/1990.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 49877448 2017-12-05 08:31 flight_delay/1991.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 50040946 2017-12-05 08:31 flight_delay/1992.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 50111774 2017-12-05 08:31 flight_delay/1993.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 51123887 2017-12-05 08:31 flight_delay/1994.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 74481752 2017-12-05 08:31 flight_delay/1995.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 75897707 2017-12-05 08:31 flight_delay/1996.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 76705697 2017-12-05 08:31 flight_delay/1997.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 76683506 2017-12-05 08:31 flight_delay/1998.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 79449438 2017-12-05 08:31 flight_delay/1999.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 82537824 2017-12-05 08:31 flight_delay/2000.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 83478700 2017-12-05 08:31 flight_delay/2001.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 75907218 2017-12-05 08:31 flight_delay/2002.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 95326801 2017-12-05 08:31 flight_delay/2003.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 110925331 2017-12-05 08:32 flight_delay/2004.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 112450321 2017-12-05 08:32 flight_delay/2005.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 115019195 2017-12-05 08:32 flight_delay/2006.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 121249243 2017-12-05 08:32 flight_delay/2007.csv.bz2
-rw-r--r-- 3 schakra2 hdfs 113753229 2017-12-05 08:32 flight_delay/2008.csv.bz2
-bash-4.1$ pig
WARNING: Use "yarn jar" to launch YARN applications.
17/12/05 08:36+44 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
17/12/05 08:36+44 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
17/12/05 08:36+44 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2017-12-05 08:36+44,842 [main] INFO org.apache.pig.Main - Apache Pig version 0.15.0 IBM_14 (reported) compiled Jul 22 2016, 02:14:12
2017-12-05 08:36+44,842 [main] INFO org.apache.pig.Main - Logging error messages to: /home/schakra2/pig_1512463004841.log
2017-12-05 08:36+44,859 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/schakra2/.pigbootup not found
2017-12-05 08:36+45,274 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://bi-hadoop-prod-4021.bi.services.us-south.bluemix.net:8020
grunt>[1]
```

We're now going to create a table from our CSV using a Pig query. Copy and paste the following query to run the command and create the table.

```
grunt> RAW_DATA = LOAD '/user/schakra2/flight_delay/2008.csv.bz2' USING PigStorage(',') AS
(year: int, month: int, day: int, dow: int,
dtime: int, sdtime: int, arftime: int, satime: int,
carrier: chararray, fn: int, tn: chararray,
etime: int, setime: int, airtime: int,
adelay: int, ddelay: int,
scode: chararray, dcode: chararray, dist: int,
tintime: int, touttime: int,
cancel: chararray, cancelcode: chararray, diverted: int,
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_7 = LOAD '/user/schakra2/flight_delay/2007.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_6 = LOAD '/user/schakra2/flight_delay/2006.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_6 = LOAD '/user/schakra2/flight_delay/2006.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,
```

```
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_5 = LOAD '/user/schakra2/flight_delay/2005.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_4 = LOAD '/user/schakra2/flight_delay/2004.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_3 = LOAD '/user/schakra2/flight_delay/2003.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_2 = LOAD '/user/schakra2/flight_delay/2002.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_1 = LOAD '/user/schakra2/flight_delay/2001.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,
```

```
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_0 = LOAD '/user/schakra2/flight_delay/2000.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_99 = LOAD '/user/schakra2/flight_delay/1999.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_98 = LOAD '/user/schakra2/flight_delay/1998.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_97 = LOAD '/user/schakra2/flight_delay/1997.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_96 = LOAD '/user/schakra2/flight_delay/1996.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,
```

```
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_95 = LOAD '/user/schakra2/flight_delay/1995.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_94 = LOAD '/user/schakra2/flight_delay/1994.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_93 = LOAD '/user/schakra2/flight_delay/1993.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_92 = LOAD '/user/schakra2/flight_delay/1992.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_91 = LOAD '/user/schakra2/flight_delay/1991.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,
```

```
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_90 = LOAD '/user/schakra2/flight_delay/1990.csv.bz2' USING  
PigStorage(',') AS
```

```
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

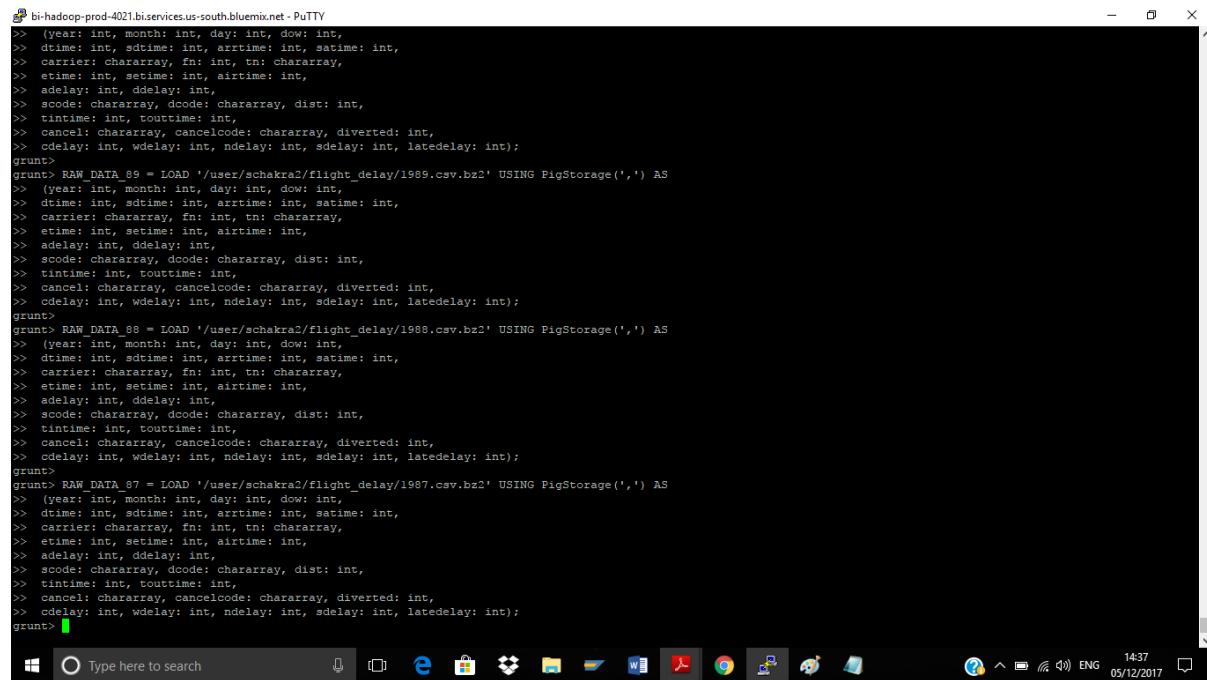
```
grunt> RAW_DATA_89 = LOAD '/user/schakra2/flight_delay/1989.csv.bz2' USING  
PigStorage(',') AS
```

```
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_88 = LOAD '/user/schakra2/flight_delay/1988.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

```
grunt> RAW_DATA_87 = LOAD '/user/schakra2/flight_delay/1987.csv.bz2' USING  
PigStorage(',') AS  
(year: int, month: int, day: int, dow: int,  
dtime: int, sdtime: int, arrtime: int, satime: int,  
carrier: chararray, fn: int, tn: chararray,  
etime: int, setime: int, airtime: int,  
adelay: int, ddelay: int,  
scode: chararray, dcode: chararray, dist: int,  
tintime: int, touttime: int,  
cancel: chararray, cancelcode: chararray, diverted: int,  
cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
```

The output should look like this



```
>> (year: int, month: int, day: int, dow: int,
>> dtme: int, sdtme: int, arftime: int, satime: int,
>> carrier: chararray, fn: int, tn: chararray,
>> etime: int, setime: int, airtime: int,
>> adelay: int, ddelay: int,
>> scode: chararray, dcode: chararray, dist: int,
>> tintime: int, touttime: int,
>> cancel: chararray, cancelcode: chararray, diverted: int,
>> cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
grunt> RAW_DATA_89 = LOAD '/user/schakra2/flight_delay/1989.csv.bz2' USING PigStorage(',') AS
>> (year: int, month: int, day: int, dow: int,
>> dtme: int, sdtme: int, arftime: int, satime: int,
>> carrier: chararray, fn: int, tn: chararray,
>> etime: int, setime: int, airtime: int,
>> adelay: int, ddelay: int,
>> scode: chararray, dcode: chararray, dist: int,
>> tintime: int, touttime: int,
>> cancel: chararray, cancelcode: chararray, diverted: int,
>> cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
grunt> RAW_DATA_88 = LOAD '/user/schakra2/flight_delay/1988.csv.bz2' USING PigStorage(',') AS
>> (year: int, month: int, day: int, dow: int,
>> dtme: int, sdtme: int, arftime: int, satime: int,
>> carrier: chararray, fn: int, tn: chararray,
>> etime: int, setime: int, airtime: int,
>> adelay: int, ddelay: int,
>> scode: chararray, dcode: chararray, dist: int,
>> tintime: int, touttime: int,
>> cancel: chararray, cancelcode: chararray, diverted: int,
>> cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
grunt> RAW_DATA_87 = LOAD '/user/schakra2/flight_delay/1987.csv.bz2' USING PigStorage(',') AS
>> (year: int, month: int, day: int, dow: int,
>> dtme: int, sdtme: int, arftime: int, satime: int,
>> carrier: chararray, fn: int, tn: chararray,
>> etime: int, setime: int, airtime: int,
>> adelay: int, ddelay: int,
>> scode: chararray, dcode: chararray, dist: int,
>> tintime: int, touttime: int,
>> cancel: chararray, cancelcode: chararray, diverted: int,
>> cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);
grunt>
```

You will need to join the data using the following shell command:

```
grunt> all_joined = UNION RAW_DATA, RAW_DATA_7, RAW_DATA_6, RAW_DATA_5,
RAW_DATA_4, RAW_DATA_3, RAW_DATA_2, RAW_DATA_1, RAW_DATA_0,
RAW_DATA_99, RAW_DATA_98, RAW_DATA_97, RAW_DATA_96, RAW_DATA_95,
RAW_DATA_94, RAW_DATA_93, RAW_DATA_92, RAW_DATA_91, RAW_DATA_90,
RAW_DATA_89, RAW_DATA_88, RAW_DATA_87;
```

Analyze the Data:

In this tutorial we are going to analyse the data set that we have just joined and find out some unique insights. The following insights are going to be worked upon:

- **Most Popular Airport**
- **Top monthly outbound from LAX**
- **Arrival and departure – LAX to other airports**

- Average Delay of airline carriers
- Longest flight by airtime

Note: Don't forget to change the user name before you type in the query

The following are the queries for the analysis that we are going to do:

Most Popular Airport

Copy and paste the following query

```
CARRIER_DATA = FOREACH all_joined GENERATE month AS m, carrier AS cname;
```

```
GROUP_CARRIERS = GROUP CARRIER_DATA BY (m,cname);
```

```
COUNT_CARRIERS = FOREACH GROUP_CARRIERS GENERATE FLATTEN(group),  
LOG10(COUNT(CARRIER_DATA)) AS popularity;
```

```
dump COUNT_CARRIERS-- we must save the result instead of dumping
```

```
STORE COUNT_CARRIERS INTO  
'/user/schakra2/output/final/COUNT_CARRIERS' USING PigStorage(',');
```

Top monthly outbound

Copy and paste the following query

```
OUTBOUND = FOREACH all_joined GENERATE month AS m, scode AS s;
```

```
GROUP_OUTBOUND = GROUP OUTBOUND BY (m,s);
```

```
COUNT_OUTBOUND = FOREACH GROUP_OUTBOUND GENERATE  
FLATTEN(group), COUNT(OUTBOUND) AS count;
```

```
GROUP_COUNT_OUTBOUND = GROUP COUNT_OUTBOUND BY m;
```

```
topMonthlyOutbound = FOREACH GROUP_COUNT_OUTBOUND {  
    result = TOP(20, 2, COUNT_OUTBOUND);  
    GENERATE FLATTEN(result);  
}
```

```
STORE topMonthlyOutbound INTO  
'/user/schakra2/output/final/OUTBOUND-TOP' USING PigStorage(',')
```

Monthly Traffic

```
UNION_TRAFFIC = UNION COUNT_INBOUND, COUNT_OUTBOUND;
```

```
GROUP_UNION_TRAFFIC = GROUP UNION_TRAFFIC BY (m,d);
```

```
TOTAL_TRAFFIC = FOREACH GROUP_UNION_TRAFFIC GENERATE FLATTEN(group) AS (m,code),  
SUM(UNION_TRAFFIC.count) AS total;
```

```
TOTAL_MONTHLY = GROUP TOTAL_TRAFFIC BY m;
```

```
topMonthlyTraffic = FOREACH TOTAL_MONTHLY {  
    result = TOP(20, 2, TOTAL_TRAFFIC);  
    GENERATE FLATTEN(result) AS (month, iata, traffic);  
}
```

```
STORE topMonthlyTraffic INTO '/user/schakra2/output/final/OUTBOUND-TOP'  
USING PigStorage(',');
```

Arrival and departure – LAX to other airports

Copy and paste the following query

```
A = FOREACH all_joined GENERATE scode AS s, dcode AS d;
```

```
B = GROUP A by (s,d);
```

```
COUNT = FOREACH B GENERATE group, COUNT(A);
```

DUMP CONT ---- we must save the result instead of dumping

```
STORE COUNT INTO '/user/schakra2/output/final/COUNT' USING  
PigStorage(',')
```

Average Delay

Copy and paste the following query

```
X= FOREACH all_joined GENERATE carrier, scode AS s, dcode AS d,  
float(adelay-ddelay) AS y;
```

```
Z = GROUP X BY carrier;
```

```
AVG_DELAY = FOREACH Z {  
    FILTER X BY (y >= 15);  
    GENERATE carrier, AVG(X.y); }
```

```
DUMP AVG_DELAY;
```

```
STORE AVG_DELAY INTO '/user/schakra2/output/final/COUNT2' USING  
PigStorage(',');
```

Longest flight by airtime

Copy and paste the following query

```
A = FOREACH all_joined GENERATE scode AS s, dcode AS d, arrtime AS x;
```

```
B = GROUP A BY (s,d,x);
```

```
LONGEST = FOREACH B GENERATE group, COUNT(x);
```

```
DUMP LONGEST;
```

```
STORE LONGEST INTO '/user/schakra2/output/final/COUNT' USING  
PigStorage(',');
```

Download all the files from the count folder in your ambari. Fetch these .csv file in excel using ODBC and come to delimeter.

Open Tableau on your local computer.

Tableau to open data file directly from Tableau and Visualization

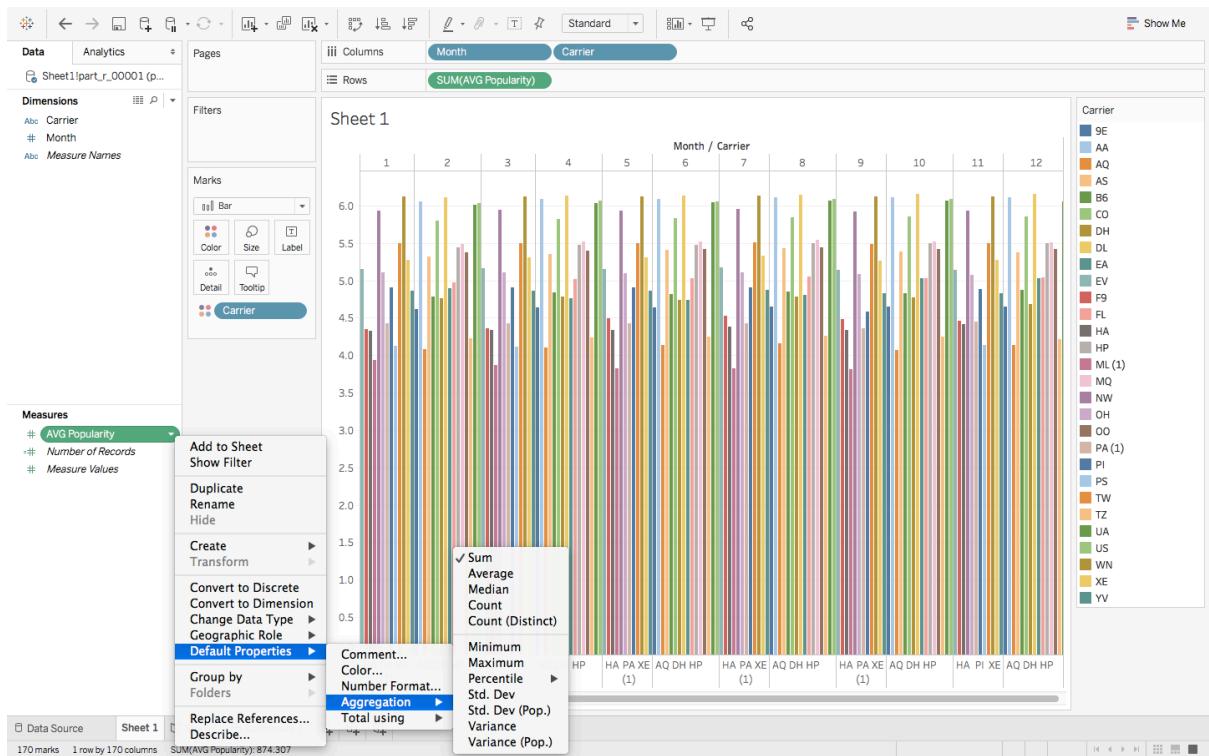
Open Tableau, and open the file according to the following order.

1. Average popularity of flight.

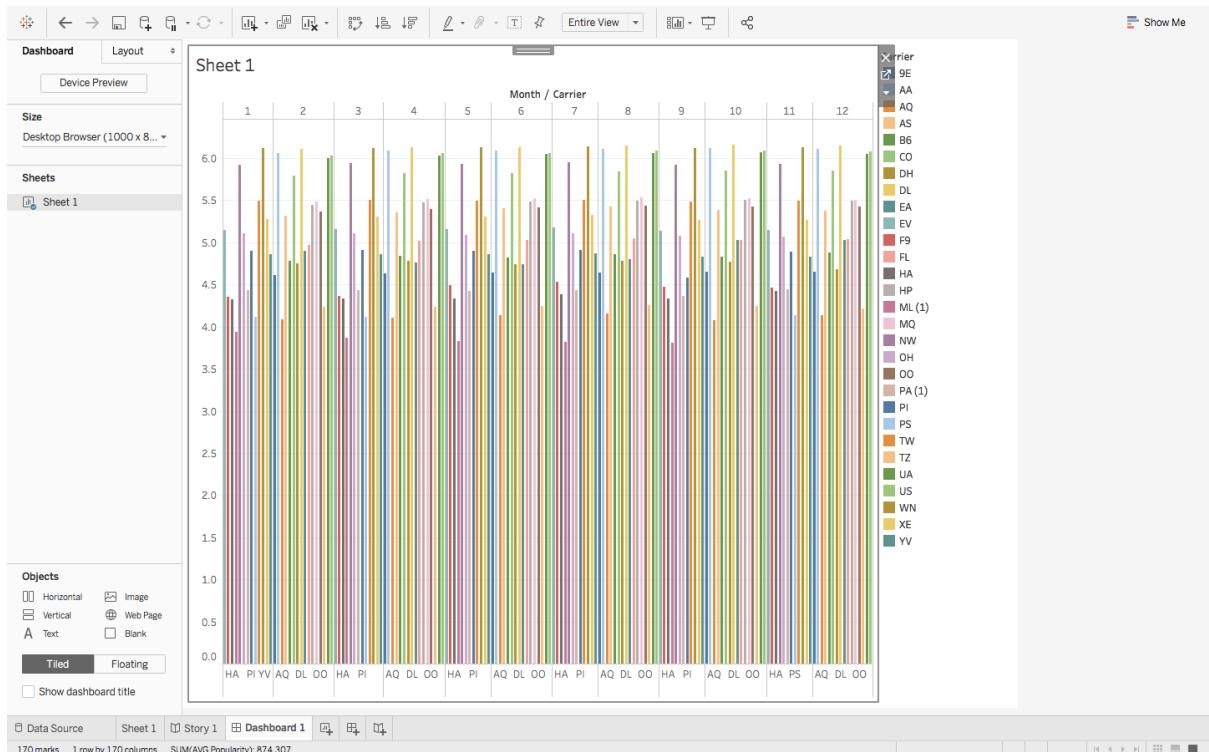
The screenshot shows the Tableau Data Source interface. On the left, under 'Connections', there is one entry: 'part01' (Excel). Under 'Sheets', there are two entries: 'Sheet1' and 'Sheet1 part_r_00001'. A preview of the 'Sheet1 part_r_00001' data is displayed in the main area. The data table has four columns: '#', 'Month', 'Carrier', and 'AVG Popularity'. The data rows are:

#	Month	Carrier	AVG Popularity
1	EV		5.14555
1	F9		4.35017
1	HA		4.32732
1	NW		5.92568
1	OH		5.10818
1	PI		4.90384
1	PS		4.11955
1	TW		5.49250
1	WN		6.11592
1	XE		5.27216
1	YV		4.86452
1	ML (1)		3.93460
1	PA (1)		4.43017

Select Sheet 1 next to Data Source, and drag AVG popularity to Rows and month and carriers to Columns. Right click on Popularity and keep its property Aggregation as SUM:



Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard. Then click on entire view .



2. Top monthly Outbound of flights

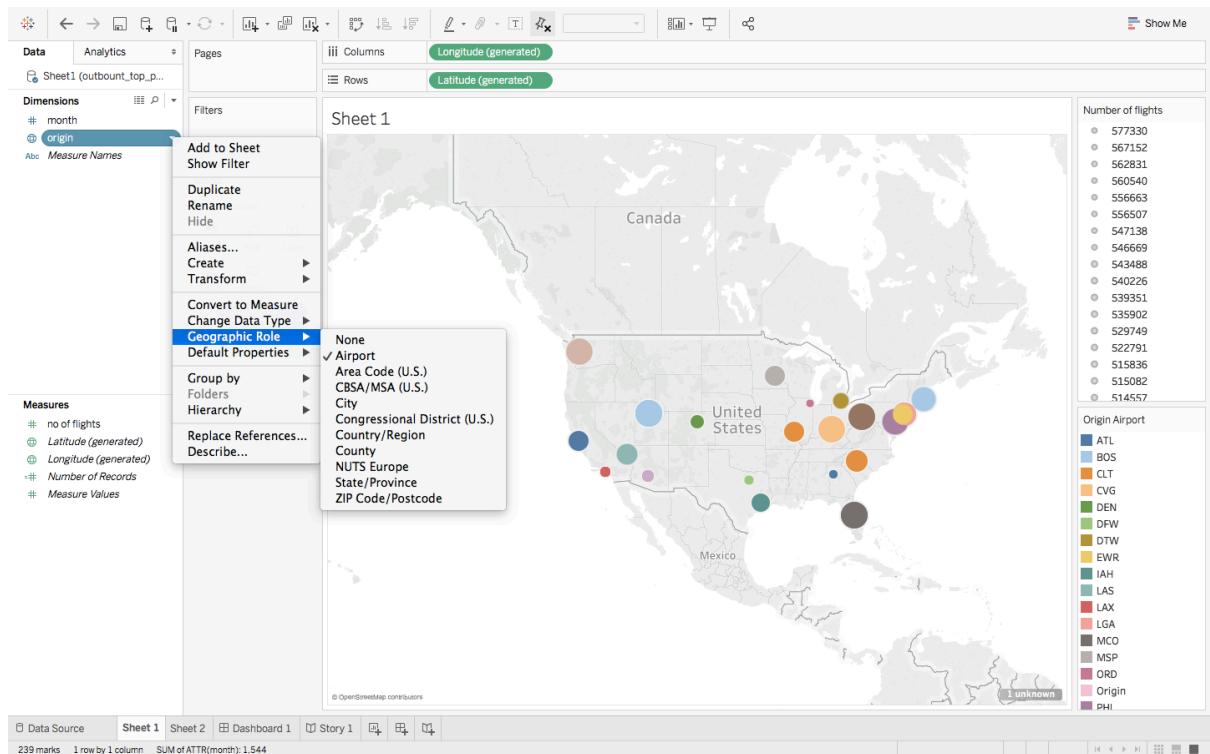
Fetch the data. Select Sheet 1 next to Data Source,

The screenshot shows a data visualization interface with the following details:

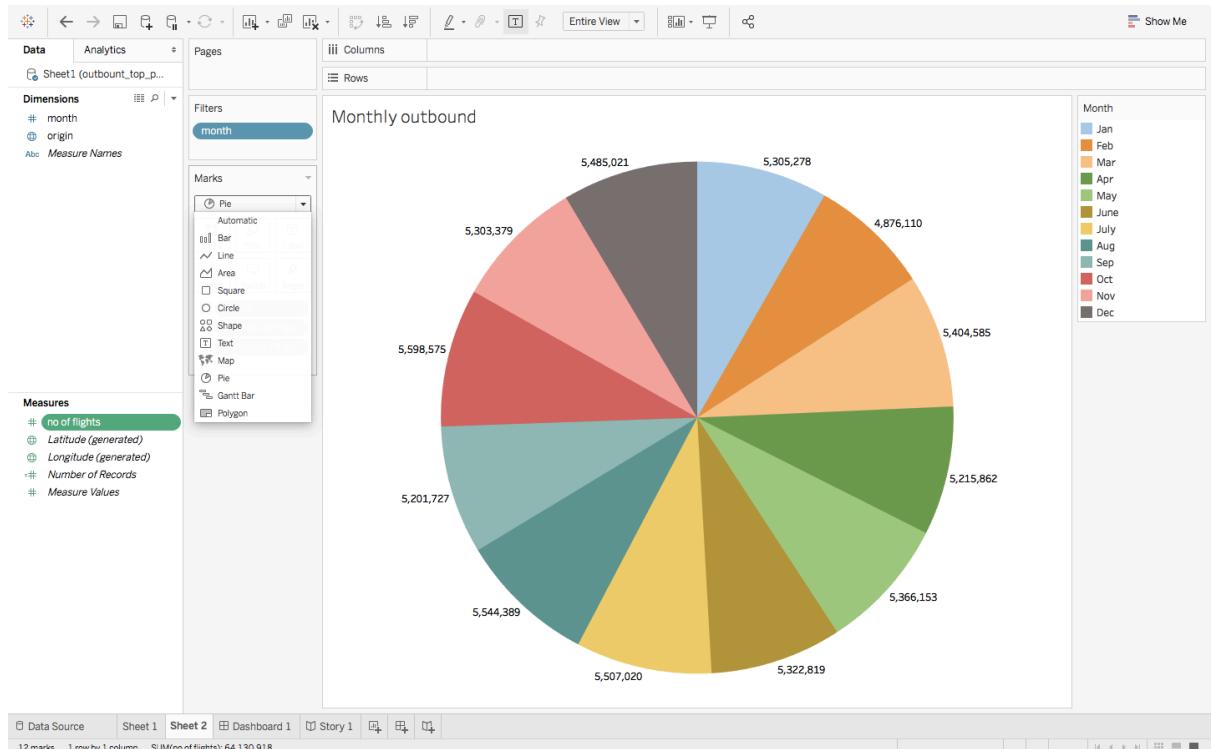
- Connections:** outbound_top_part00 (Excel)
- Sheets:** Sheet1 (selected), Sheet1 part_r_00000, New Union
- Data Interpreter:** Use Data Interpreter is checked.
- Table Headers:** #, Sheet1 month, Sheet1 origin, #, Sheet1 no of flights
- Table Data:**

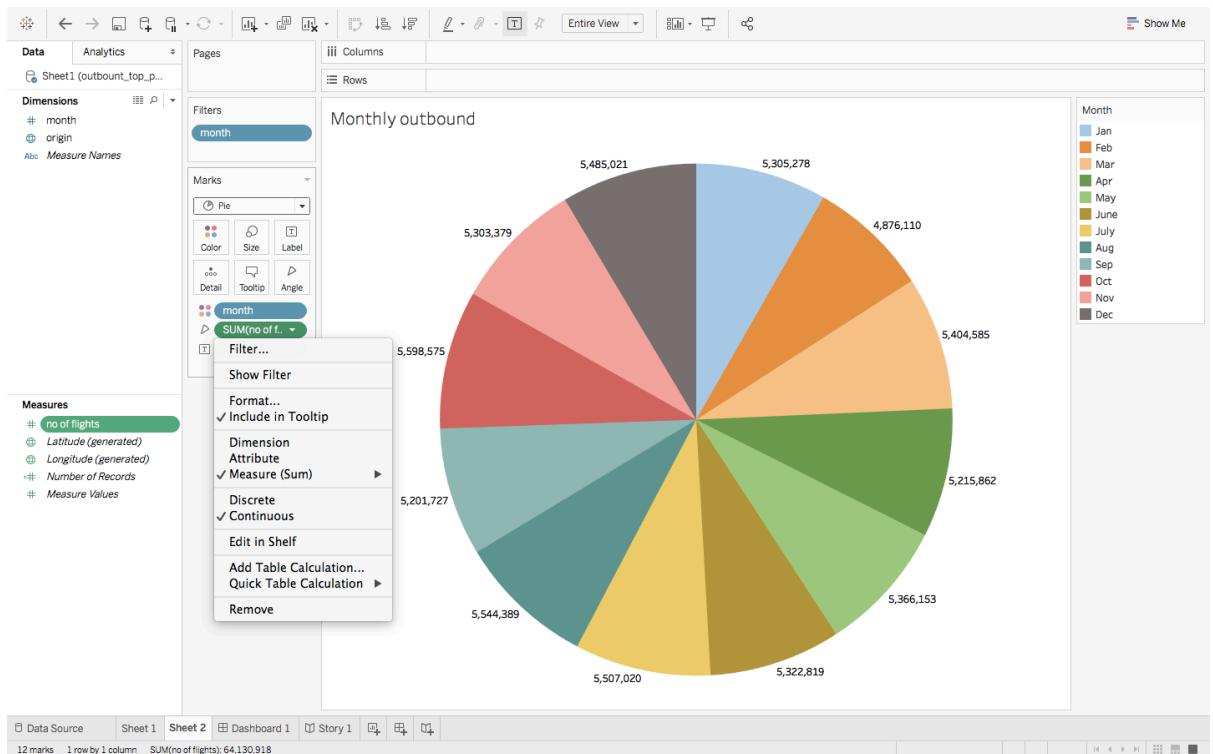
#	Sheet1 month	Sheet1 origin	#	Sheet1 no of flights
1	null	Origin	0	
2	Jan	CVG		164,836
3	Jan	SLC		165,982
4	Jan	PIT		174,945
5	Jan	LGA		193,243
6	Jan	PHL		180,215
7	Jan	LAS		215,690
8	Jan	SFO		222,988
9	Jan	EWR		224,000
10	Jan	CLT		211,650
11	Jan	BOS		190,762
12	Jan	DFW		478,239
13	Jan	DEN		268,743
- Toolbars and Buttons:** Data Source, Sheet 1, Sheet 2, Dashboard 1, Story 1, various icons for filtering, sorting, and saving.

Change the geographic role of Origin as Airport. Drag Longitude(generated) to Columns, Latitude(generated) to Rows. Select Show me, and select Geo Map:

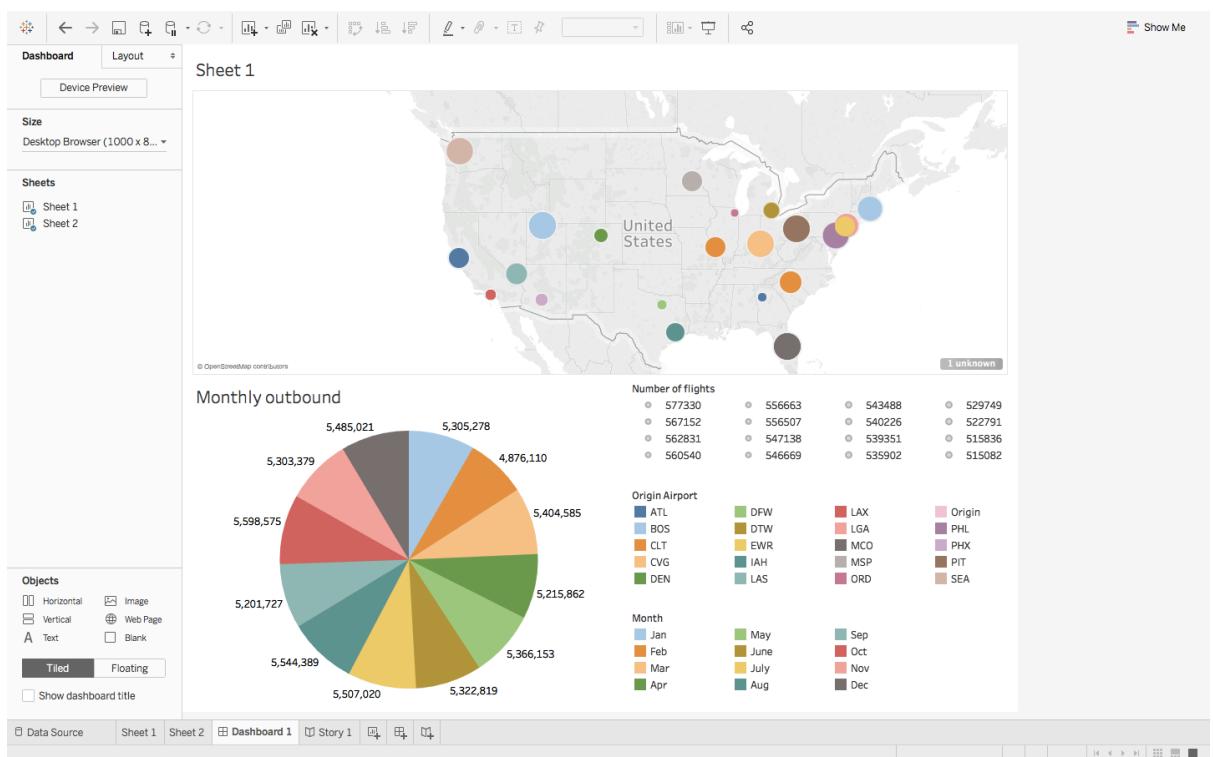


Click on Sheet 2 and in the mark functionality select pie diagram. Select months and no of flights and Fliter month





Open a dashboard by clicking next to sheet 2 and drag sheet 1 and Sheet 2 to the dashboard.



3. Monthly Traffic on Top 20 Airport.

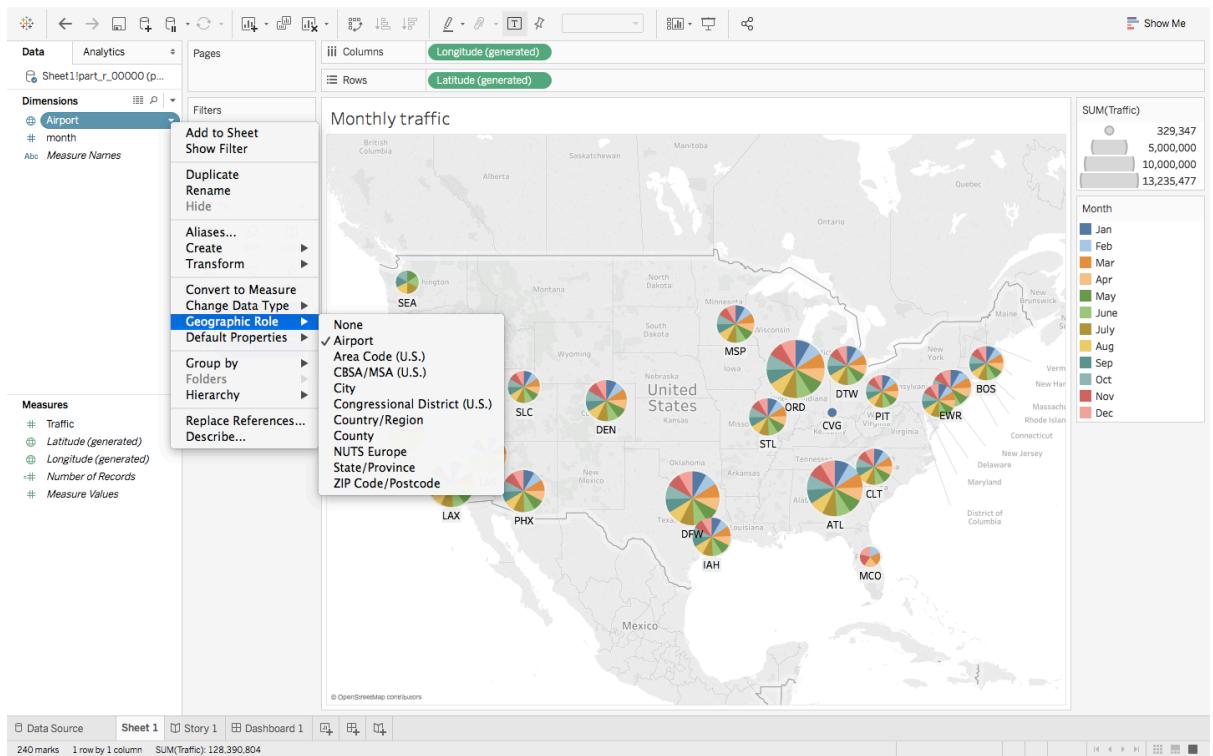
The screenshot shows a data visualization interface with the following details:

- Connections:** part00 (Excel)
- Sheets:** Sheet1, Sheet1 part_r_00000
- Data Interpreter:** A checkbox is checked with the note: "Data interpreter might be able to clean your Excel workbook."
- Table Headers:** #, month, Airport, Traffic
- Table Data:**

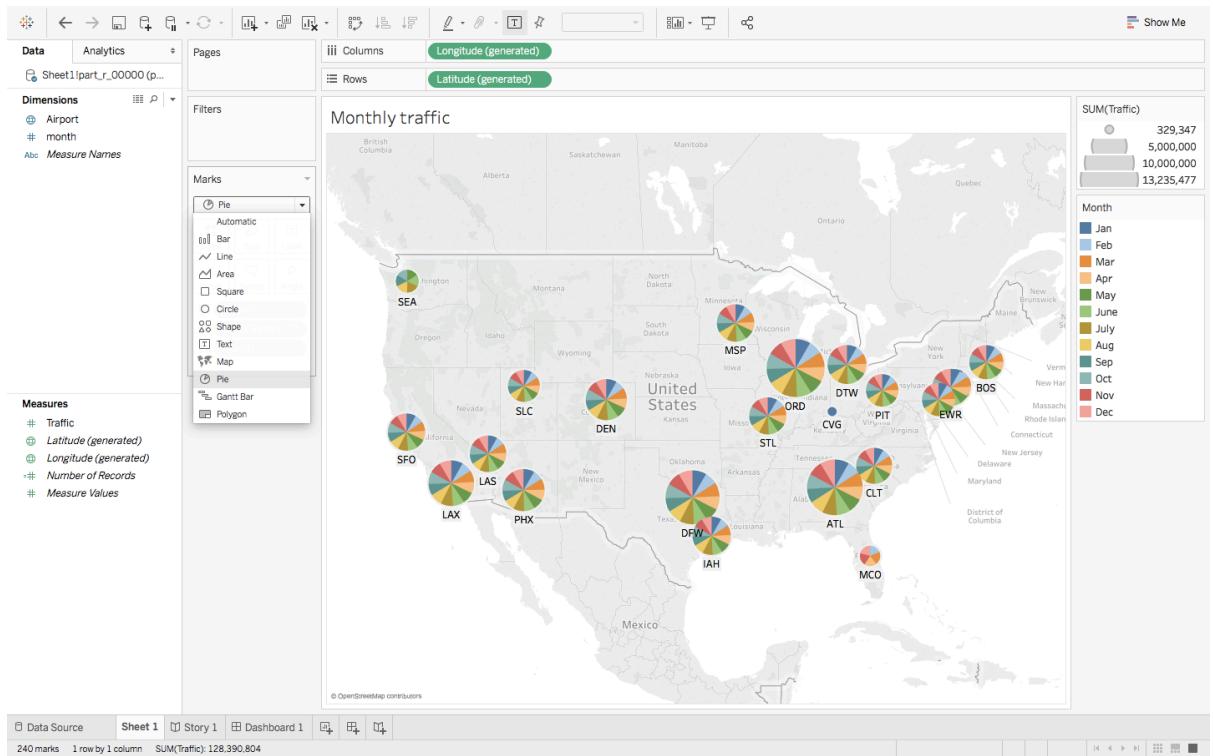
#	month	Airport	Traffic
	Jan	CVG	329,347
	Jan	SLC	331,980
	Jan	PHL	360,343
	Jan	LGA	386,296
	Jan	PIT	350,440
	Jan	EWR	448,267
	Jan	BOS	381,318
	Jan	SFO	445,337
	Jan	LAS	431,083
	Jan	STL	451,251
	Jan	DTW	490,905
	Jan	ATL	1,009,921
	Jan	DFW	959,904
- Toolbars and Buttons:** Data Source, Sheet 1, Story 1, Dashboard 1, and various navigation and filter icons.

Select Sheet 1 next to Data Source, and change Airports geographic role to Airport

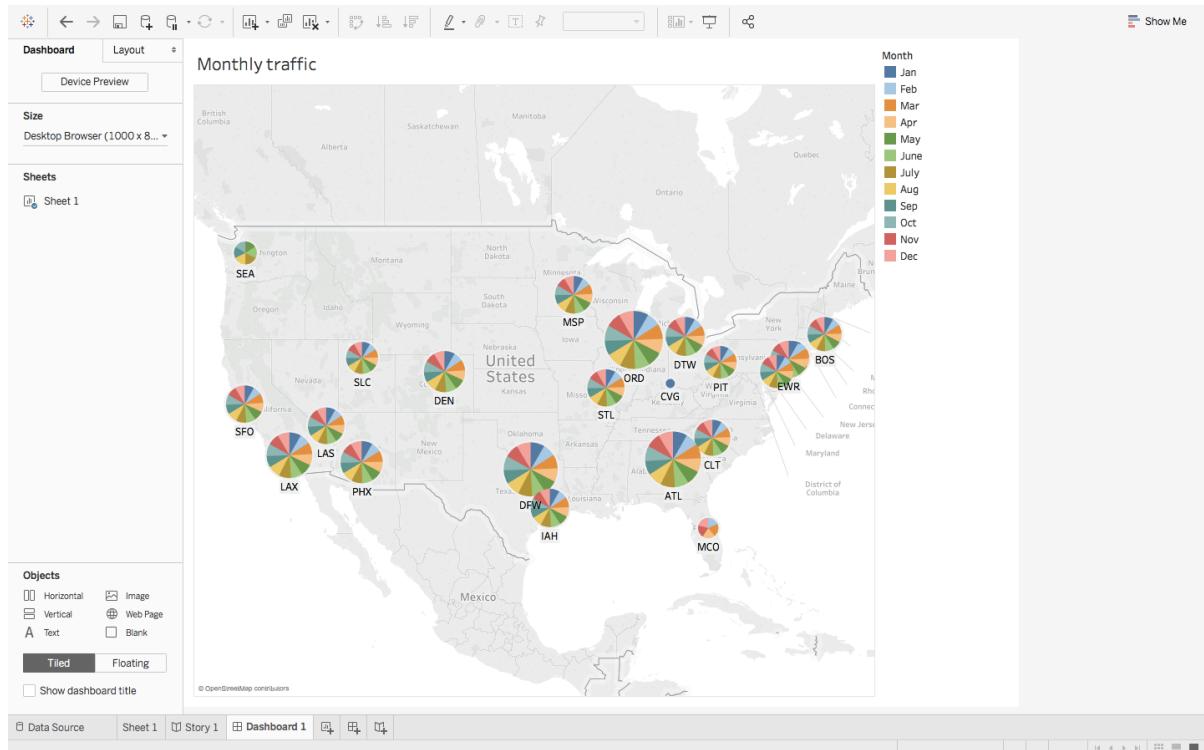
Drag Longitude(generated) to Columns, Latitude(generated) to Rows.



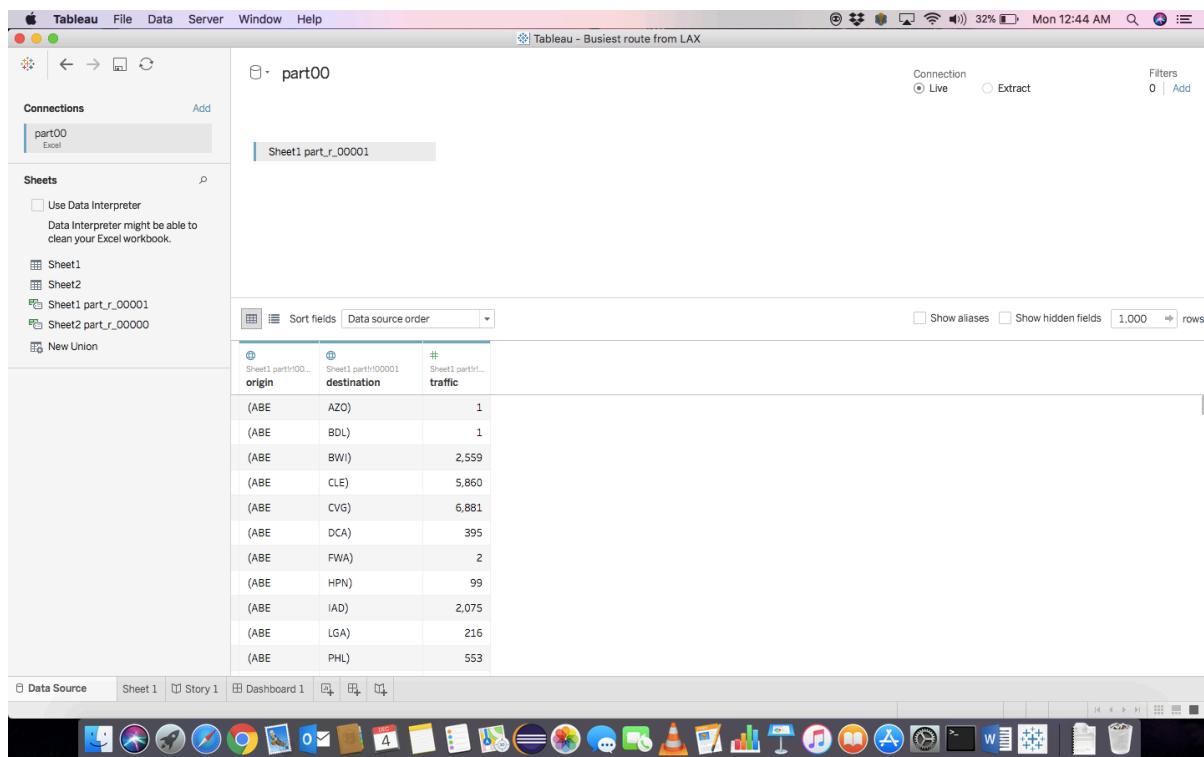
Create a new Worksheet by selecting the icon next to the Sheet 1. Drag Airport to marks and click on drop down menu to select pie chart, you will get this:



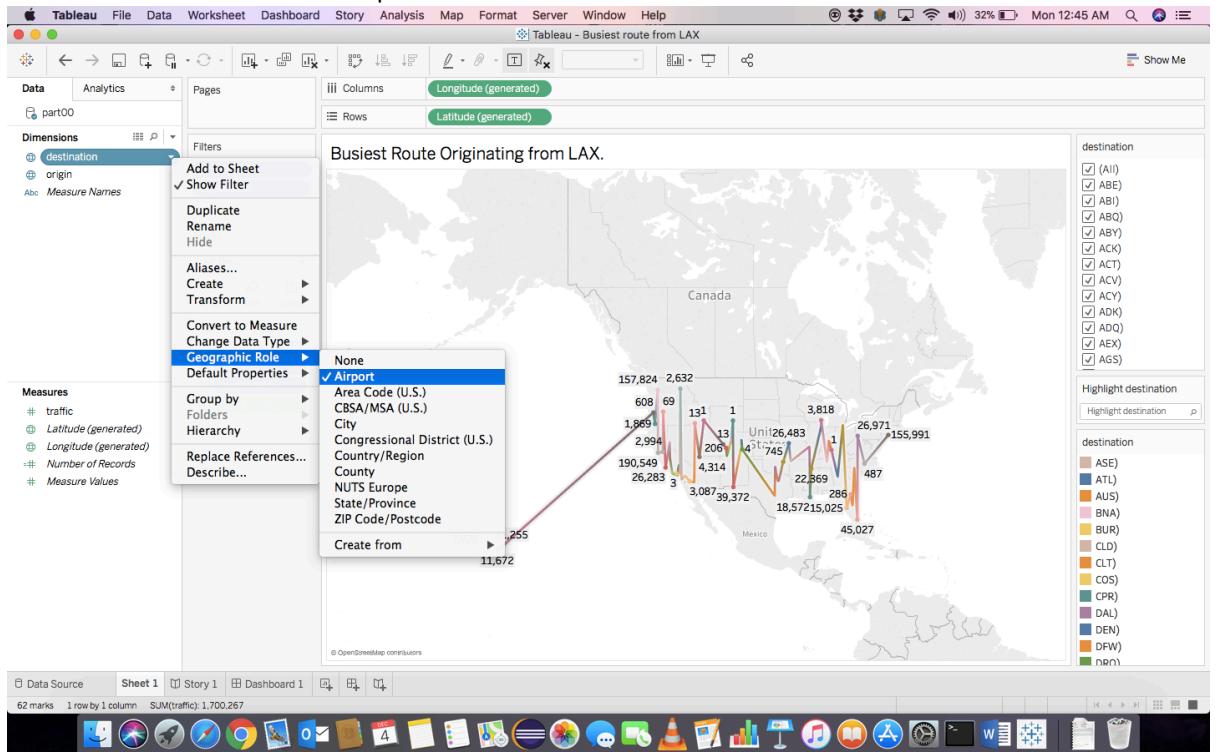
Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard.



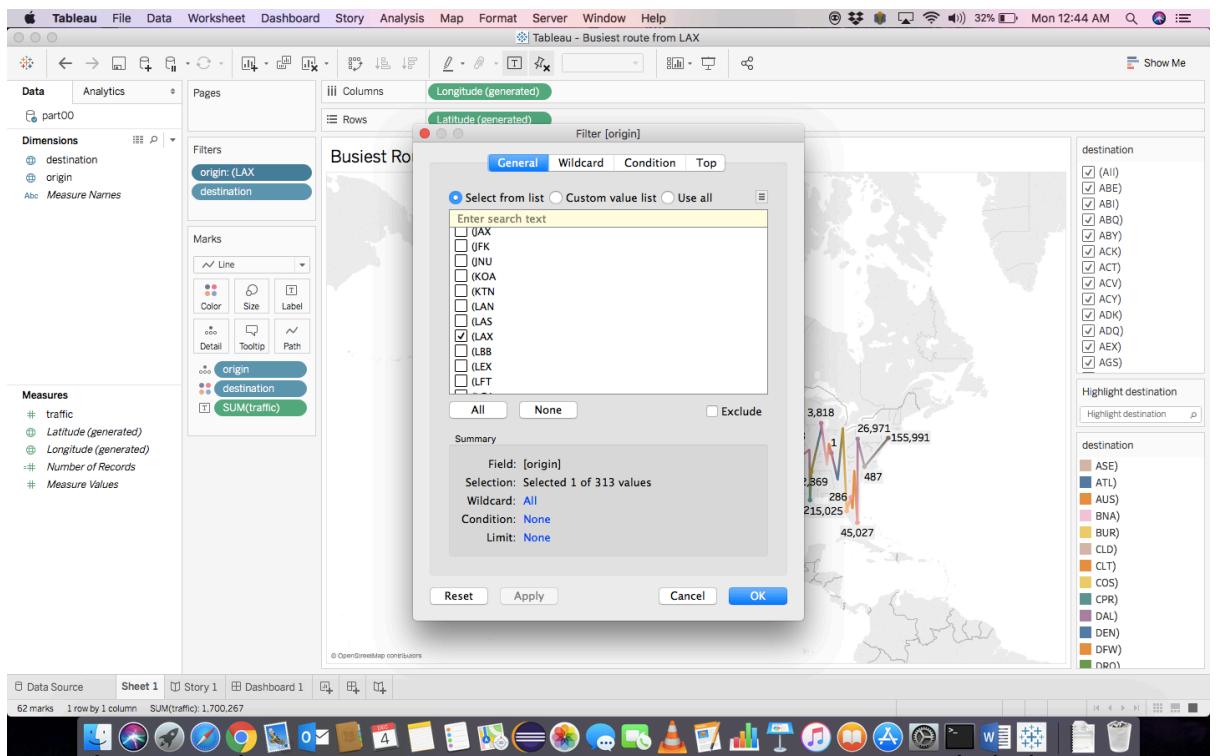
4. Arrival and departure from LAX airport.



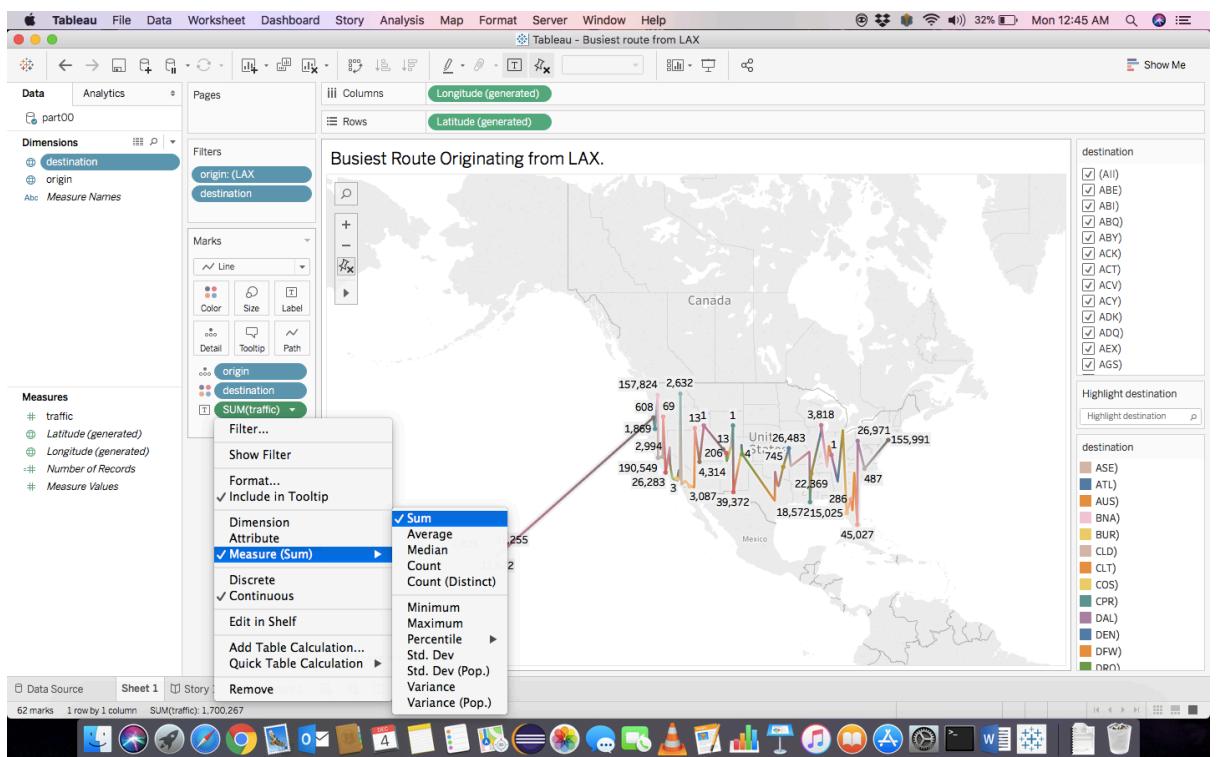
Select Sheet 1 next to Data Source, change State's geographical role of origin and destination to Airport. Drag Longitude to Columns, Latitude to Rows and select Geo Map.



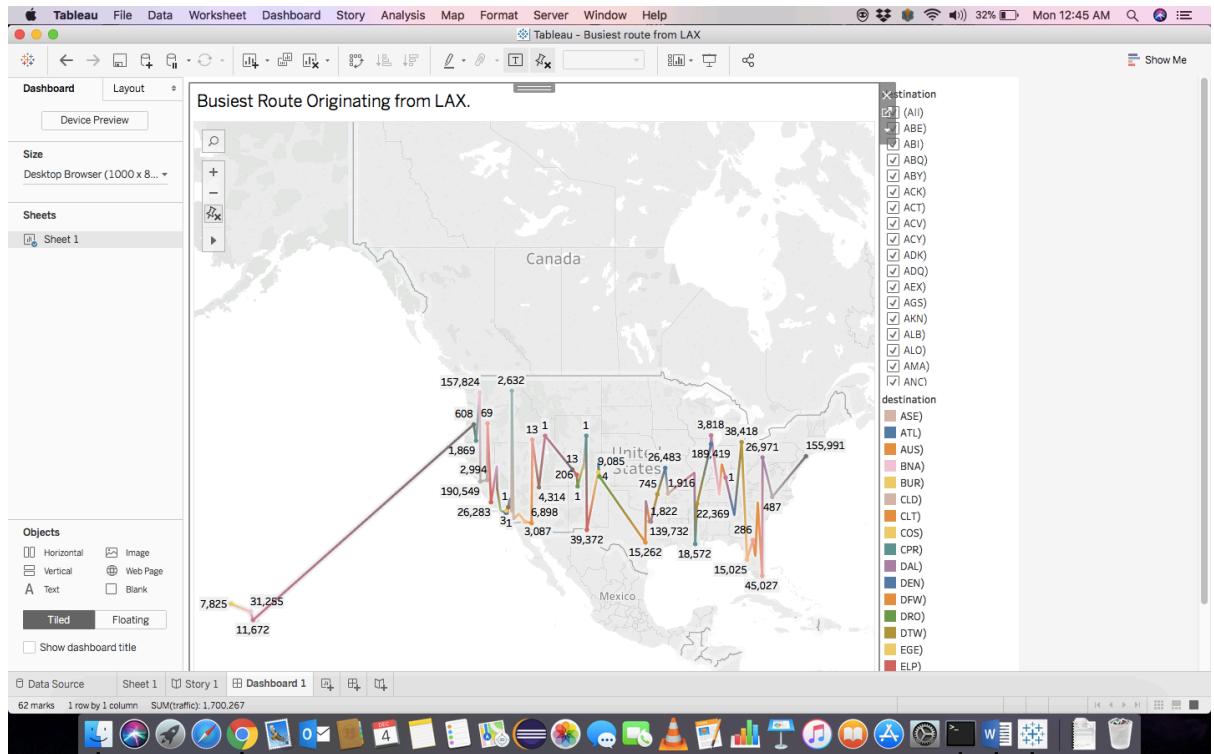
Create a new Worksheet by selecting the icon next to the Sheet 1. Use origin in filter and select LAX, you can use specific destination also in filter as shown. But in this visualization we have taken LAX as origin.



Change the measure of traffic to Sum as follow

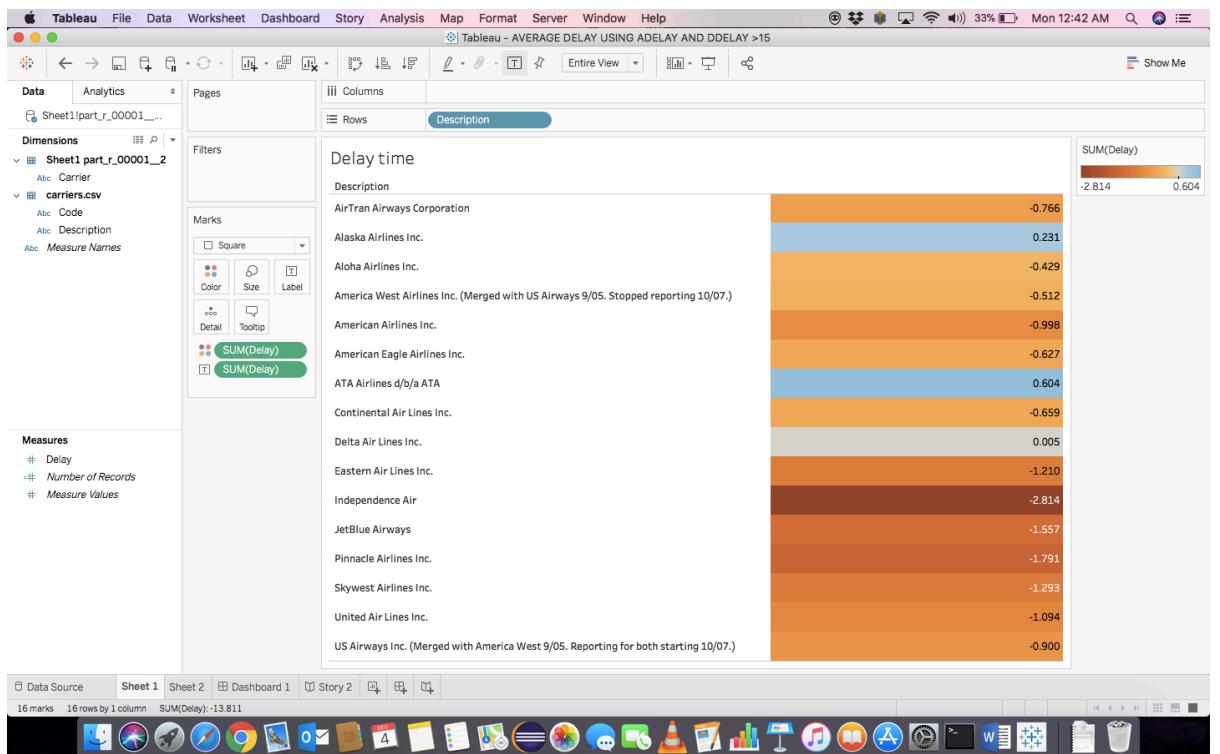


Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard.

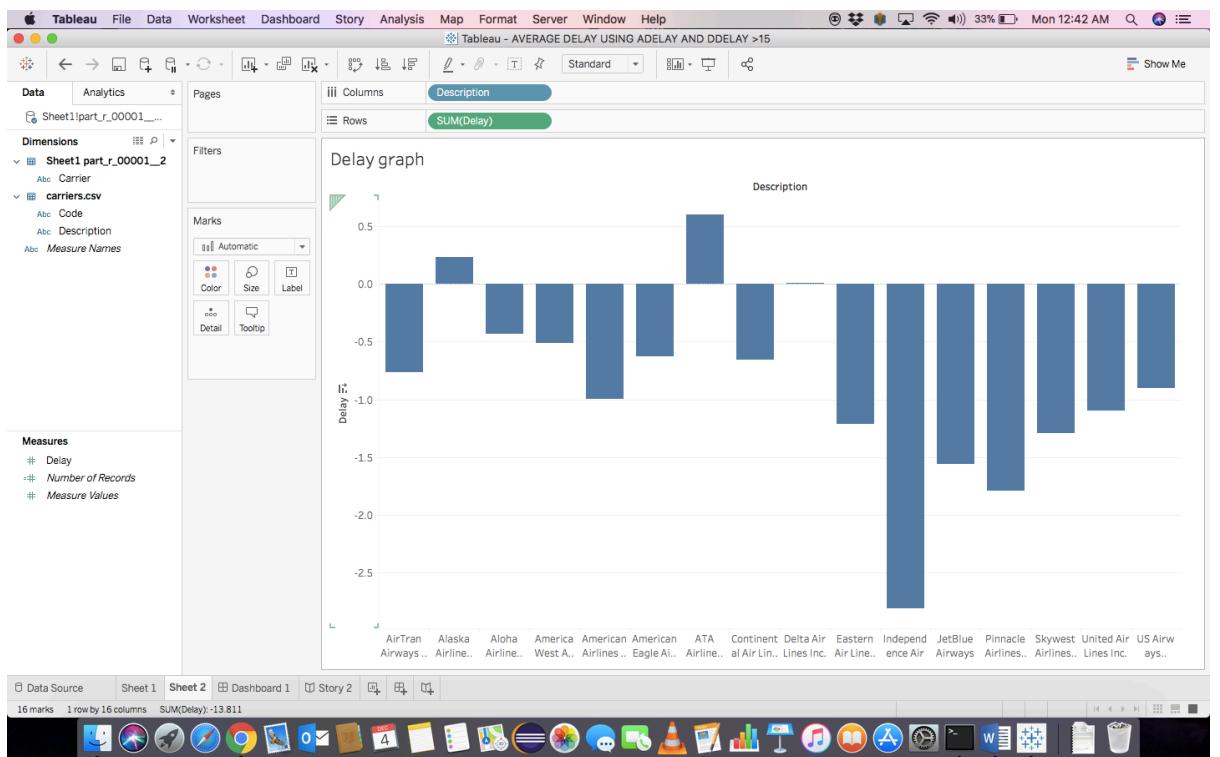


5. Average Delay

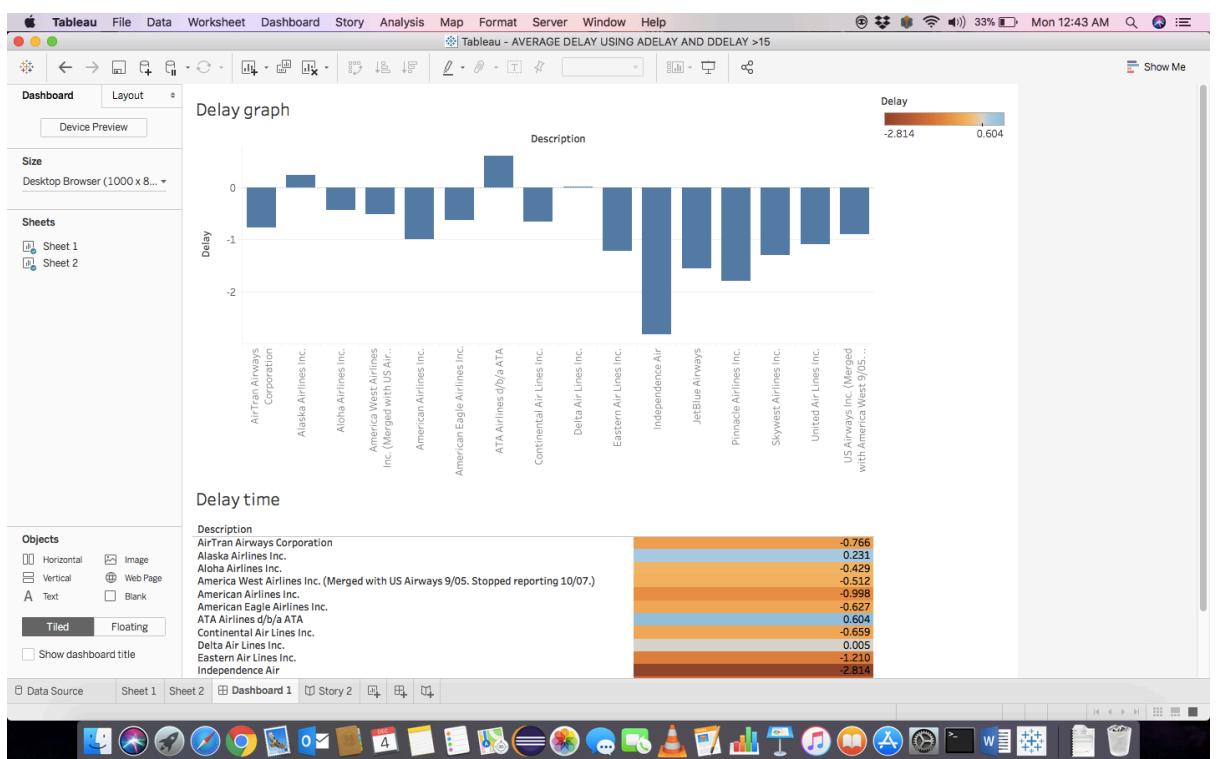
Select Sheet 1 next to Data Source, Delays to Color and again Delays to Text.



Create a new Worksheet by selecting the icon next to the Sheet 1. Drag Description to column and Delay to Rows.



Open a dashboard by clicking next to sheet 2 and drag sheet 1 and sheet 2 to the dashboard.



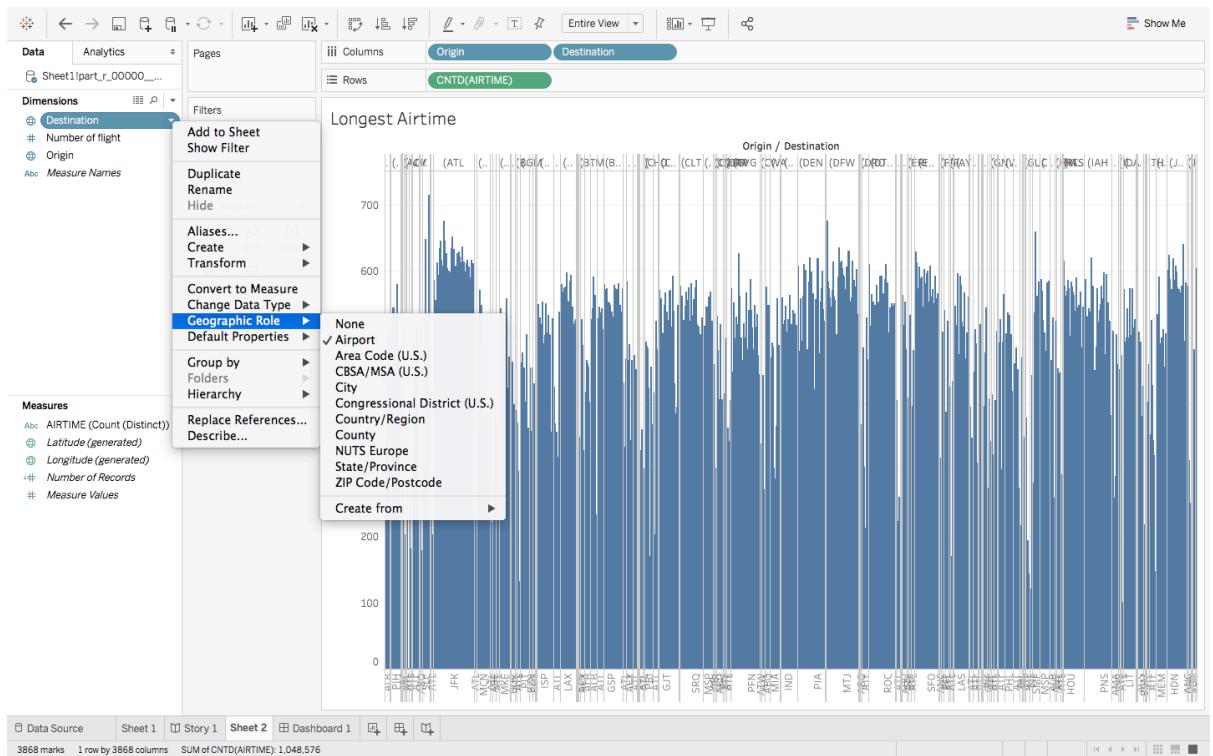
6. Longest flight by airtime

The screenshot shows a data analysis interface with the following details:

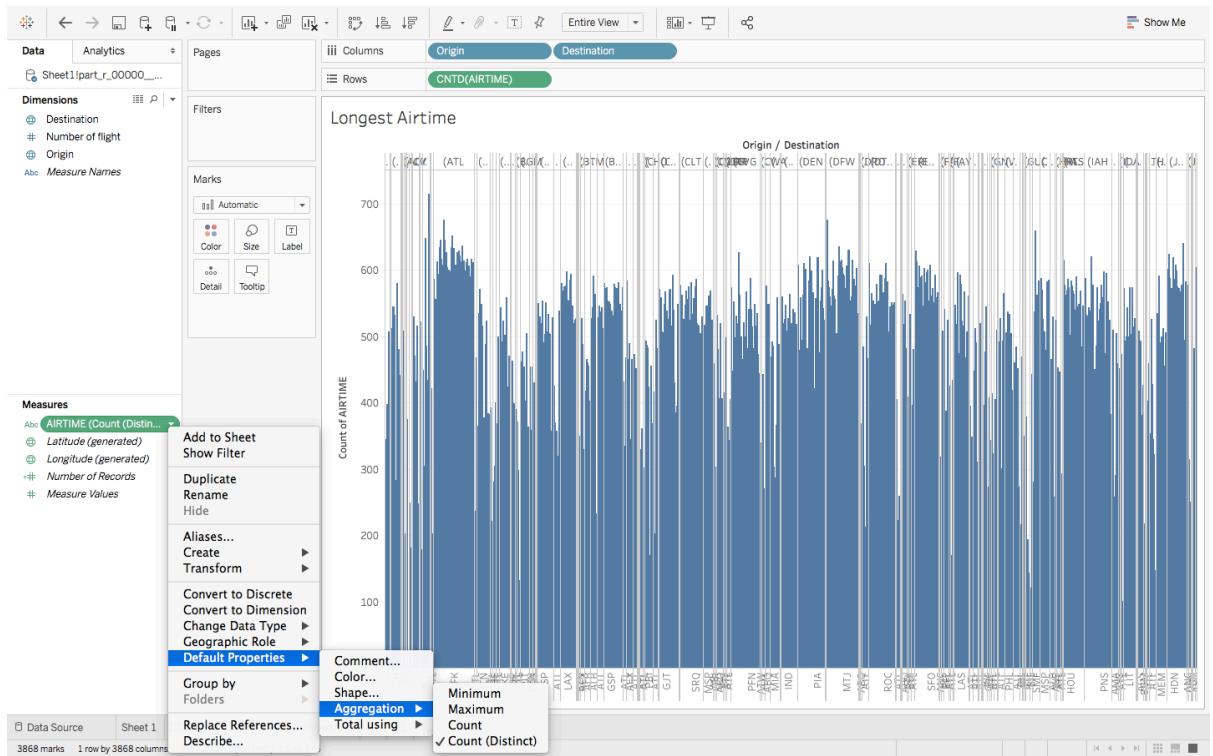
- Connections:** Longests (Excel) is selected.
- Sheets:** Sheet1 part_r_00000_3 is selected.
- Table Headers:** Origin, Destination, AIRTIME, Number of flight.
- Data Rows:** The table contains 14 rows of flight data, mostly originating from ABE and destination ALB or ATL, with varying airtimes and flight counts.
- Bottom Navigation:** Data Source, Sheet 1, Story 1, Sheet 2, Dashboard 1, and various navigation icons.

Origin	Destination	AIRTIME	Number of flight
(ABE	ALB	1744)	1
(ABE	ALB	2354)	1
(ABE	ATL	130)	1
(ABE	ATL	134)	1
(ABE	ATL	200)	1
(ABE	ATL	234)	1
(ABE	ATL	732)	1
(ABE	ATL	736)	1
(ABE	ATL	738)	4
(ABE	ATL	740)	7
(ABE	ATL	742)	12
(ABE	ATL	744)	7
(ABE	ATL	746)	10

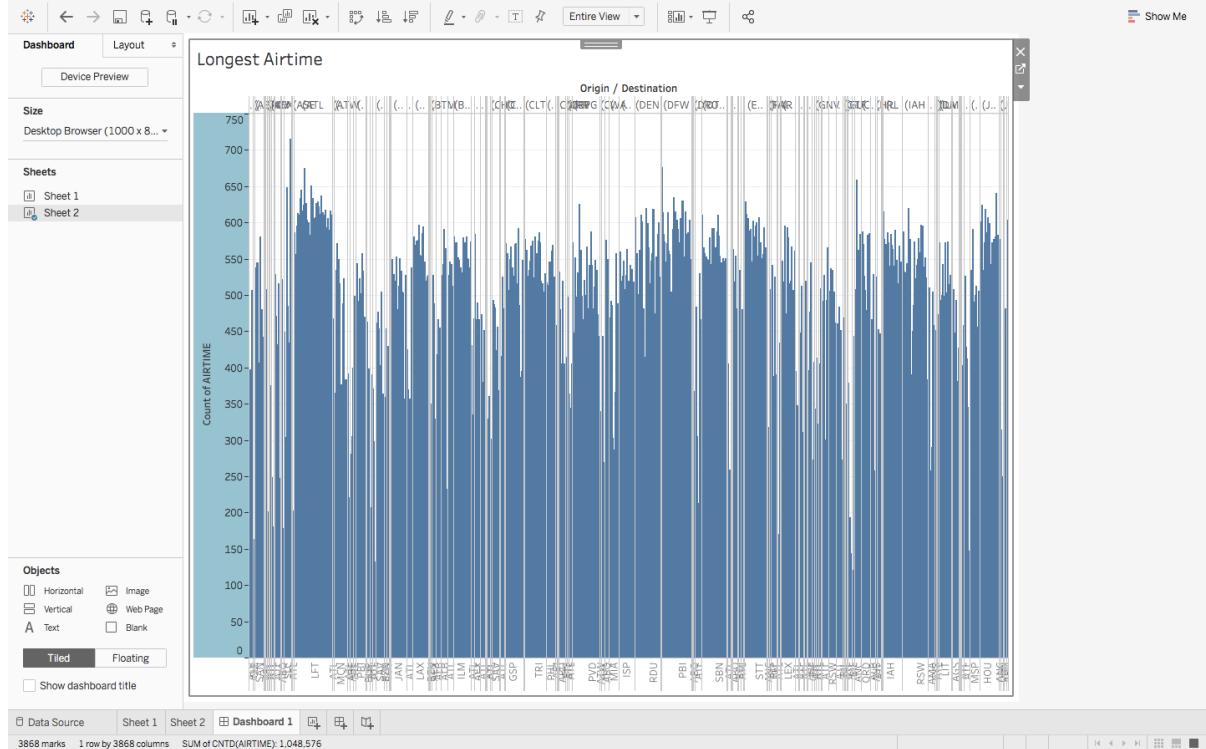
Select Sheet 1 next to Data Source, Change the geographic roles of origin and destination to Airport. Drag and drop origin and destination to column and airtime to rows and click on geo map.



Change the property of airtime in aggregation as to count.



Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard.



References

- T.A. Jones, "Writing a good paper," IEEE Trans. on *General Writing*, Vol. 1, no. 2, pp.1-10, May 2002.
- K. Hwang, *Computer Arithmetic*, John Wiley, 1997.
- Nillohit Bhattacharya and Jongwook Woo, "Airline Data Set Analysis using Big Data in Cloud Computing" in The 2017 Korea Society of Management Information Systems Spring Annual Conference (KMIS 2017), Chonnam University, Korea, June 6 - 9 2017
- <http://hadooptutorial.info/tableau-integration-with-hadoop/>
- <http://hortonworks.com/blog/how-to-integrate-tableau-and-hadoop-with-hortonworks-data-platform/>
- Apache Pig

