Flight Data Analysis from 1987 to 2008 using Hadoop Ecosystem

Presented By:

Maitri Shah
Siddhi Udani
California State University, Los Angeles

Guided By: Prof Arun Aryal

Date Published: 2nd August,2019

Project Tutorial

Hadoop: Apache Hadoop is an open source software framework used for is an open-source software for distributed storage and processing of dataset of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

Apache PIG: Apache Pig is a high-level platform for creating programs that runs on Apache Hadoop. The language for this platform is called Pig Latin. Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark. Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to SQL for relational database management systems. Pig Latin can be extended using user-defined functions (UDFs) which the user can write in Java, Python, JavaScript, Ruby or Groovy and then call directly from the language.

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
- Basic knowledge about Hadoop ecosystem and Pig commands
- 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

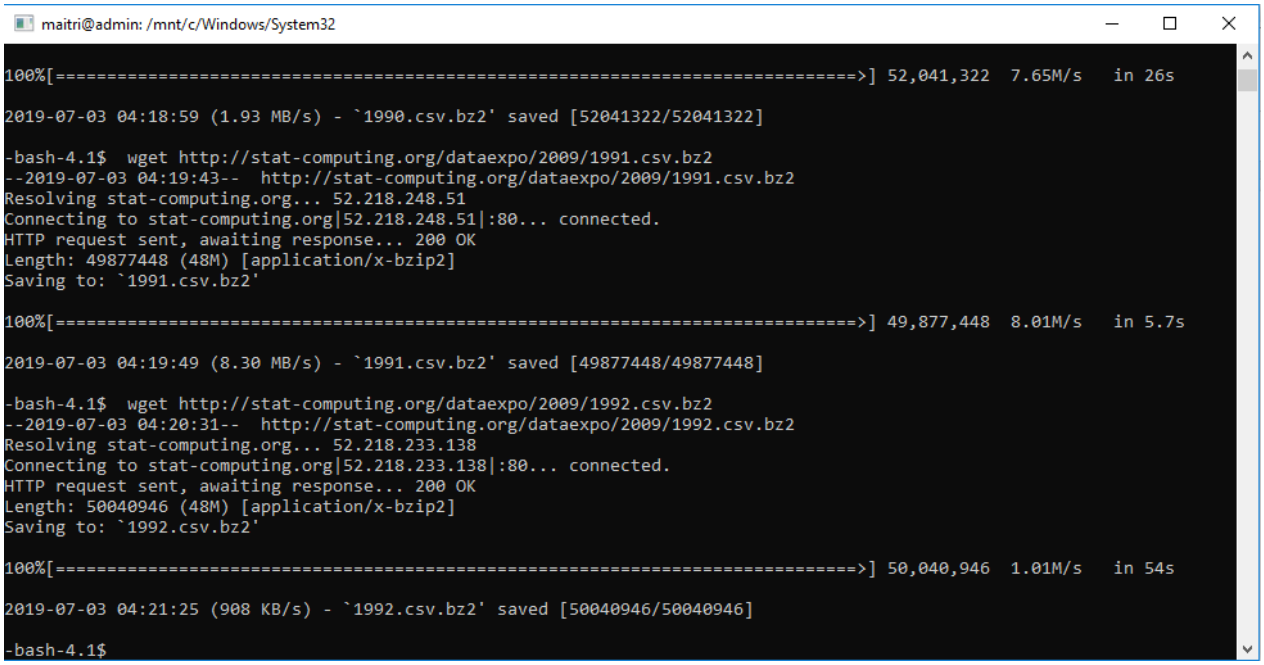
$ wget http://stat-computing.org/dataexpo/2009/2005.csv.bz2

$ 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



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

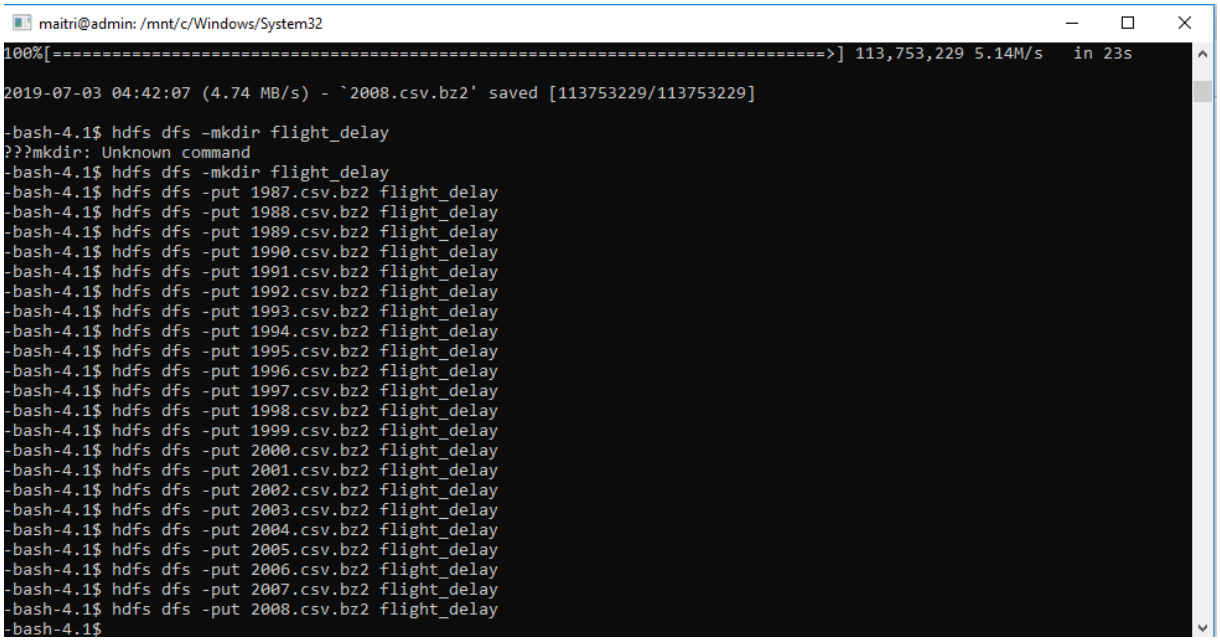
$ hdfs dfs -put 1990.csv.bz2 flight_delay

$ hdfs dfs -put 1991.csv.bz2 flight_delay

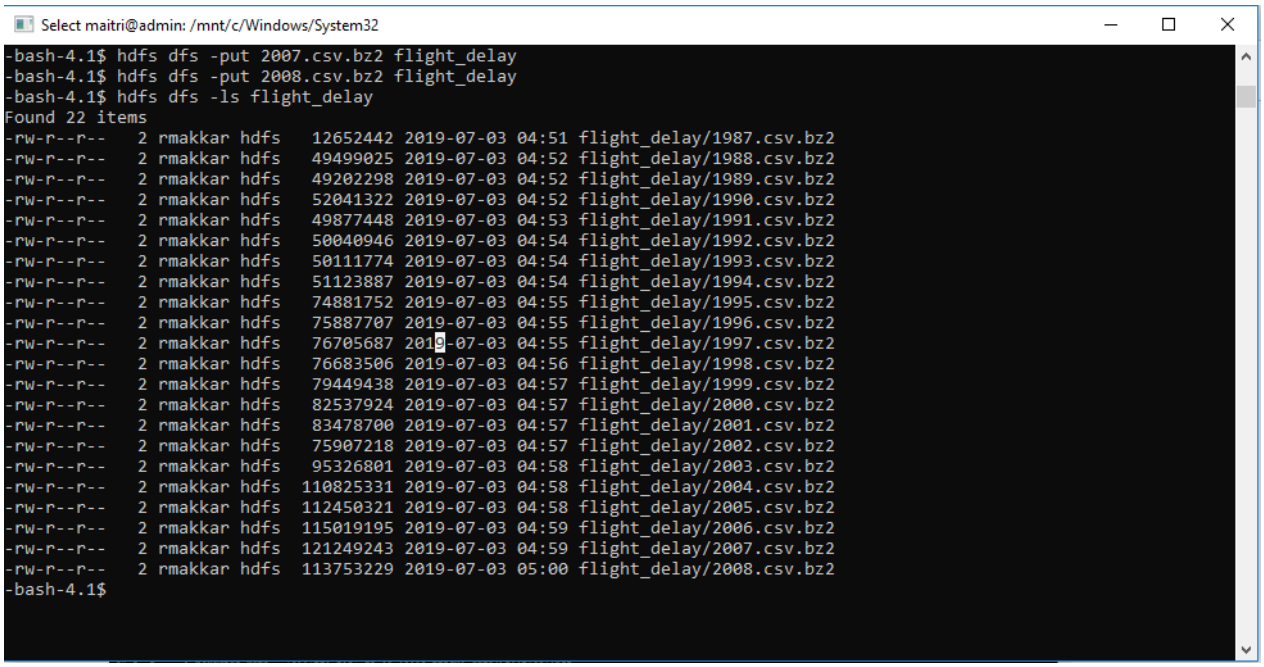
```
$ hdfs dfs -put 1992.csv.bz2 flight_delay
$ hdfs dfs -put 1993.csv.bz2 flight_delay
$ hdfs dfs -put 1994.csv.bz2 flight_delay
$ hdfs dfs -put 1995.csv.bz2 flight_delay
$ hdfs dfs -put 1996.csv.bz2 flight_delay
$ hdfs dfs -put 1997.csv.bz2 flight_delay
$ hdfs dfs -put 1998.csv.bz2 flight_delay
$ hdfs 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 -mkdir flight_delay



$ hdfs dfs -ls flight_delay

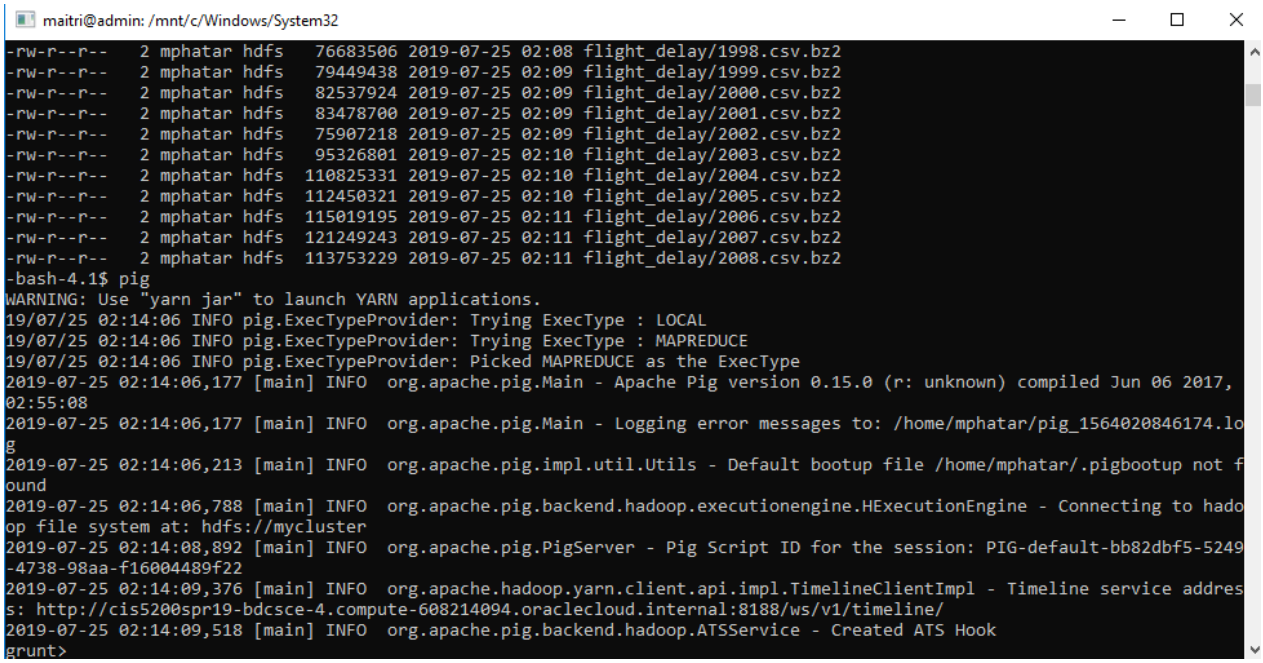


Create Tables for the Data Using Pig

Open the Pig interface in your terminal

Run the following command

$ pig



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/mshah3/flight_delay/2008.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_7 = LOAD '/user/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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,

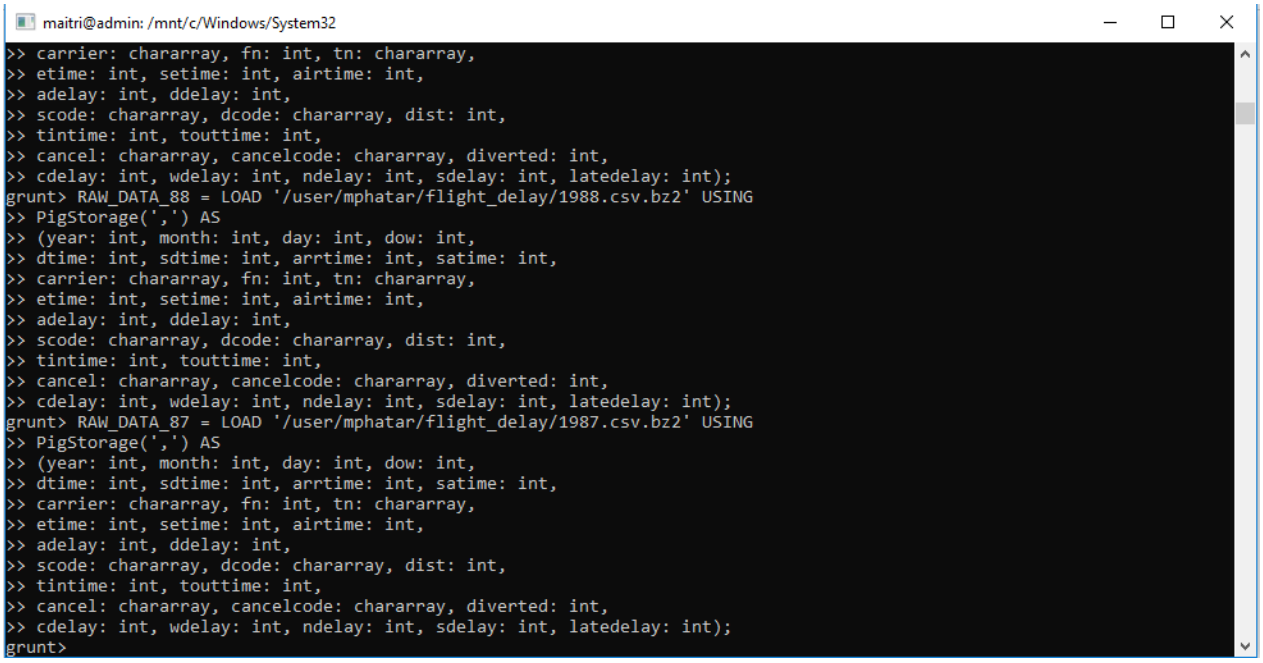
tintime: int, touttime: int,

cancel: chararray, cancelcode: chararray, diverted: int,

cdelay: int, wdelay: int, ndelay: int, sdelay: int, latedelay: int);

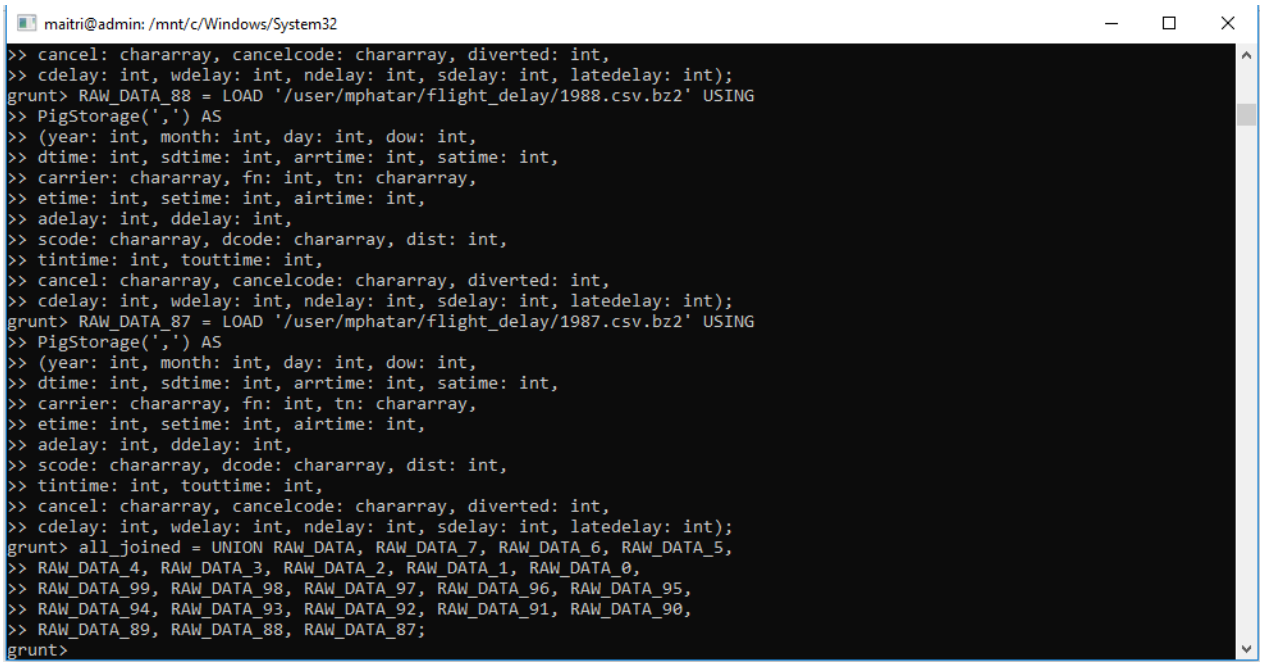
```
grunt> RAW_DATA_88 = LOAD '/user/mshah3/flight_delay/1988.csv.bz2'
USING PigStorage(',') AS
 (year: int, month: int, day: int, dow: int,
 dtime: int, sdtime: int, arrtime: int, satime: int,
 carrier: chararray, fn: int, tn: chararray,
 etime: int, setime: int, airtime: int,
 adelay: int, ddelay: 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/mshah3/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



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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/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/mshah3/output/final/COUNT' USING PigStorage(',');

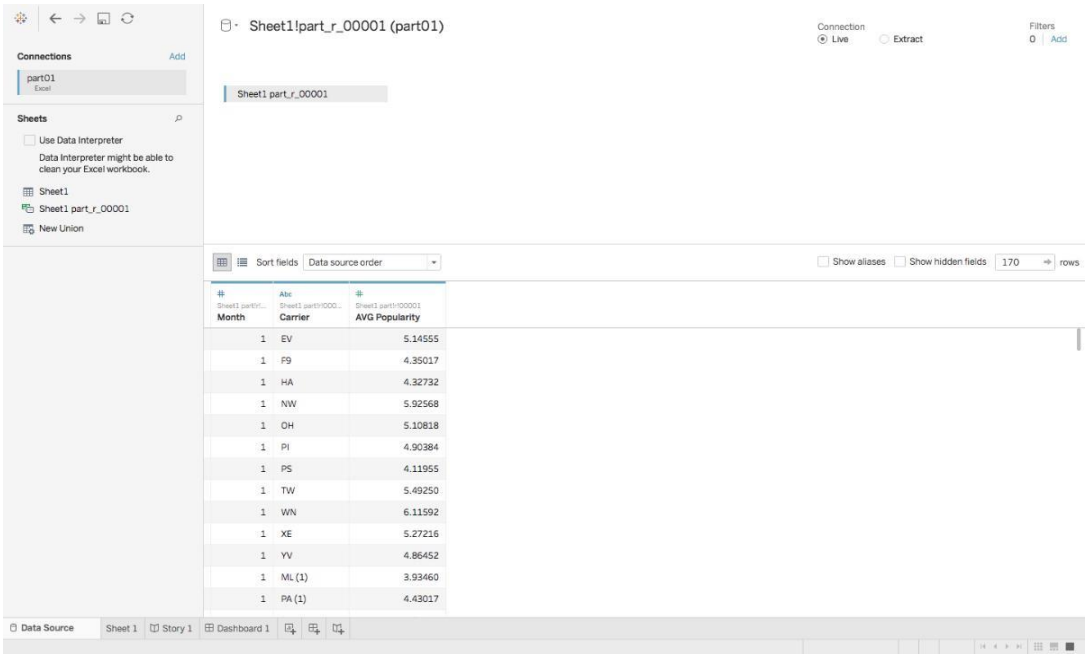
Download all the files from the count folder in your ambari. Fetch these .csv file in excel using ODBC and comer delimmeter.

Open Tableau on your local computer.

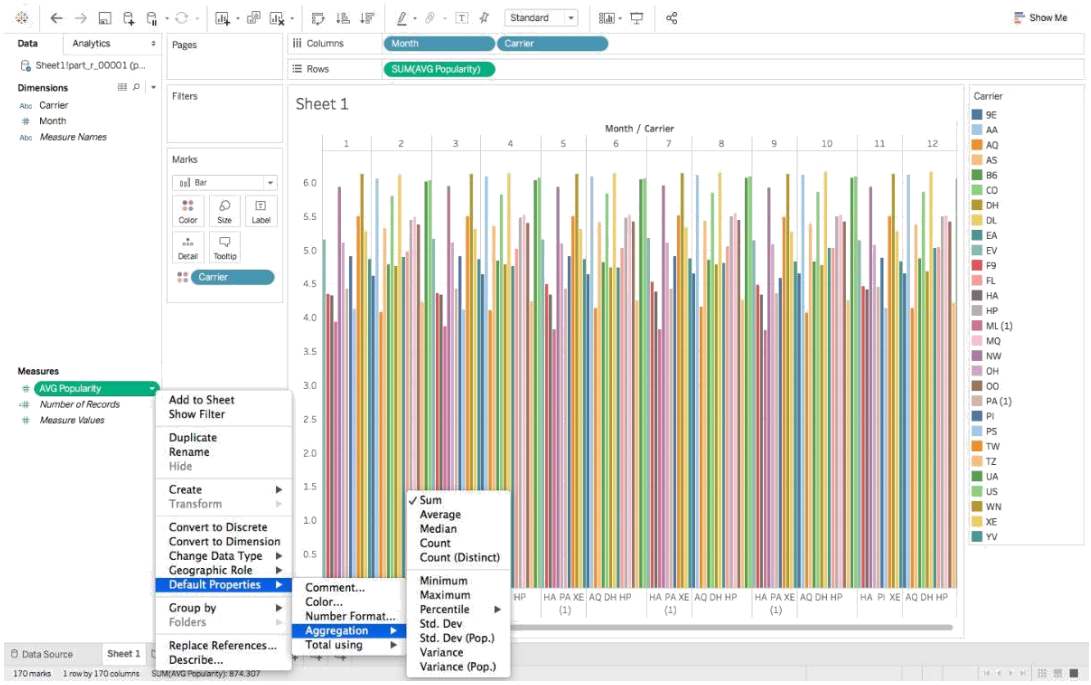
Tableau to open data file directly from Tableau and Visualization

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

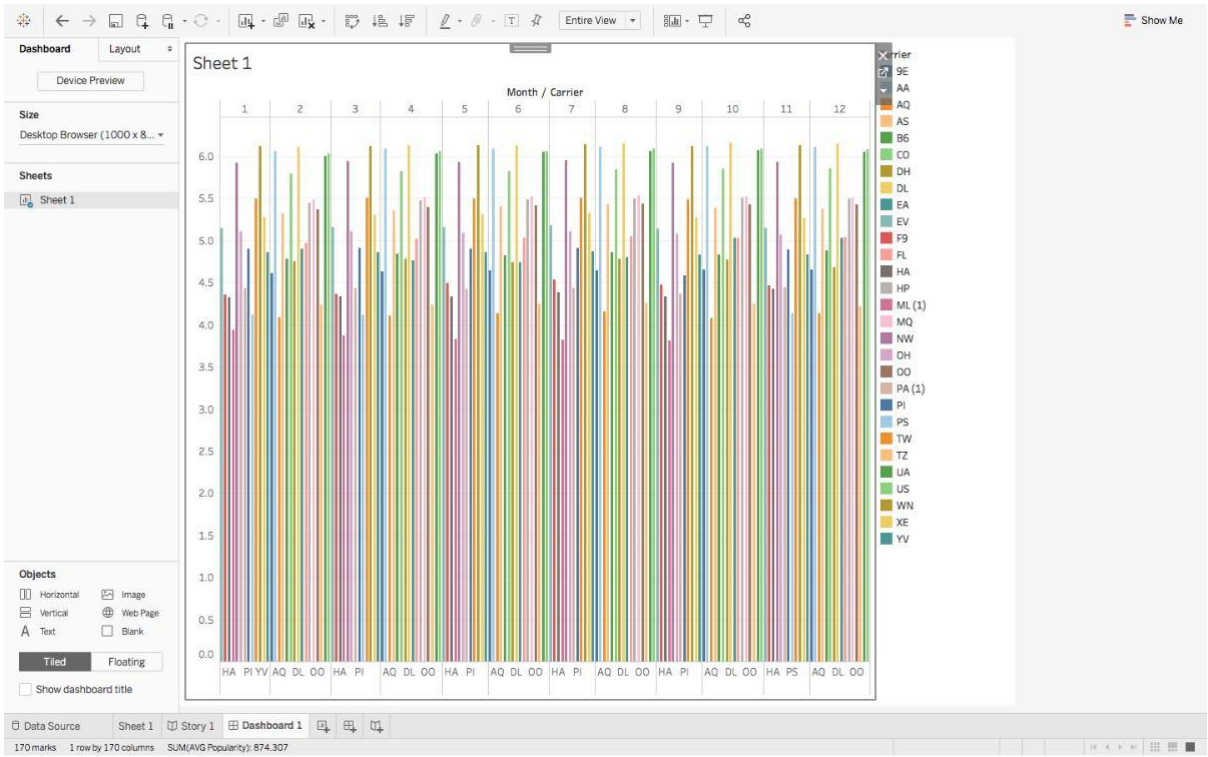
1. **Average popularity of flight.**



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:

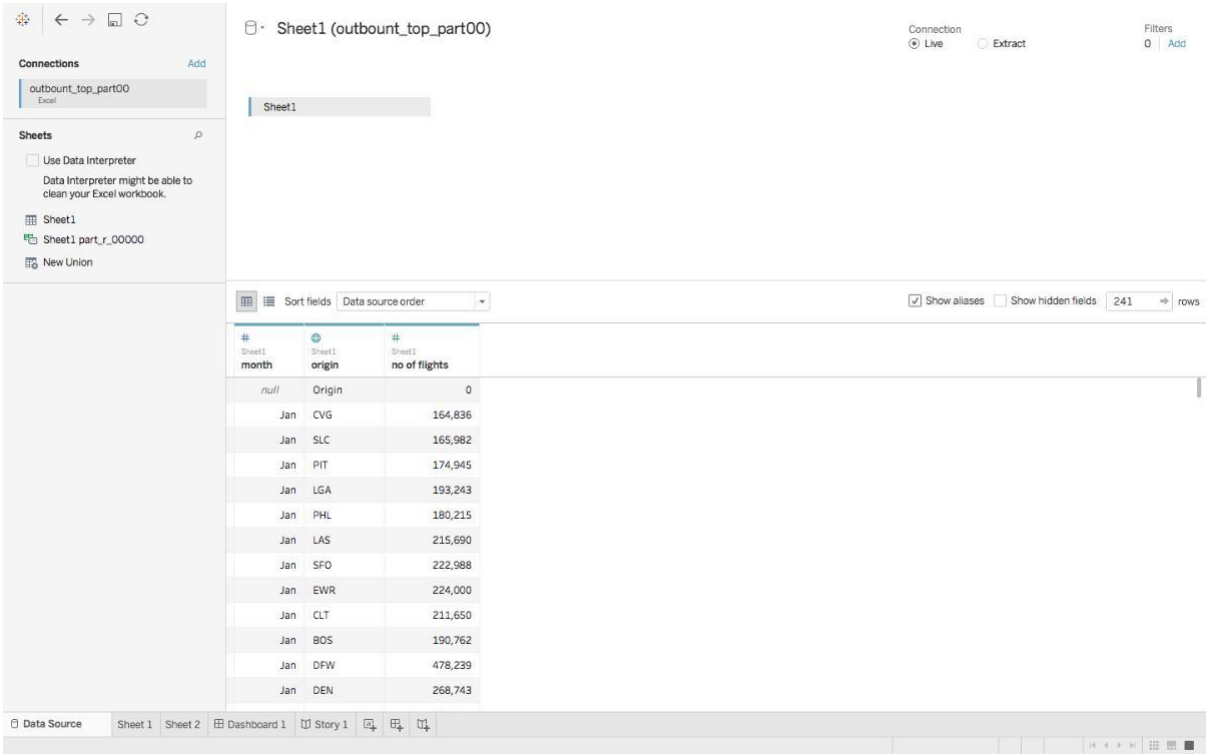


Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard. Then click on entire view.

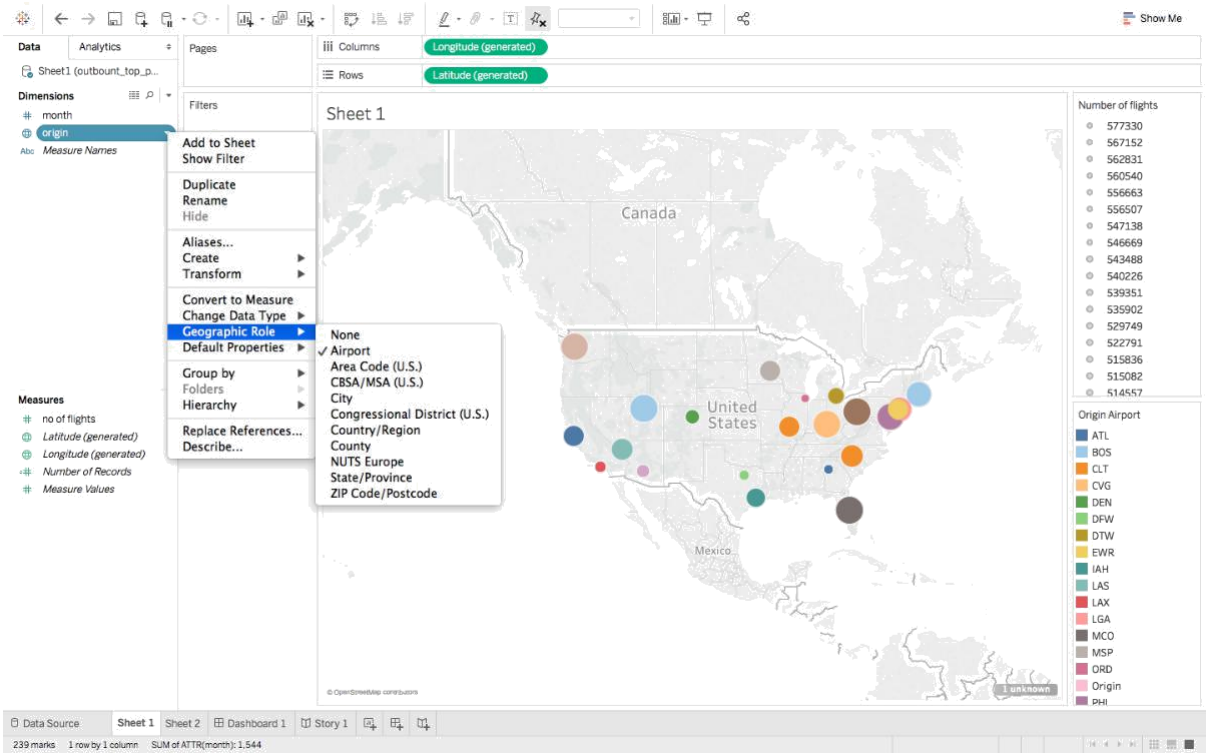


2. Top monthly Outbound of flights

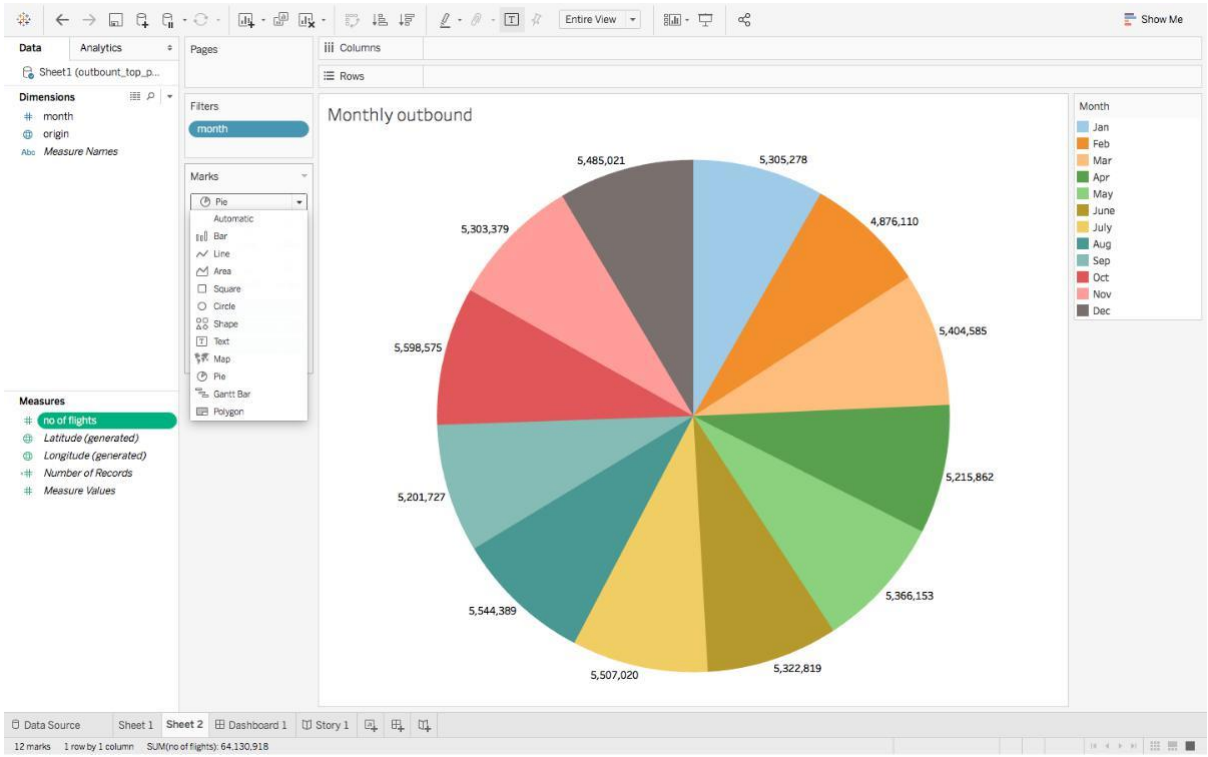
Fetch the data. Select Sheet 1 next to Data Source,

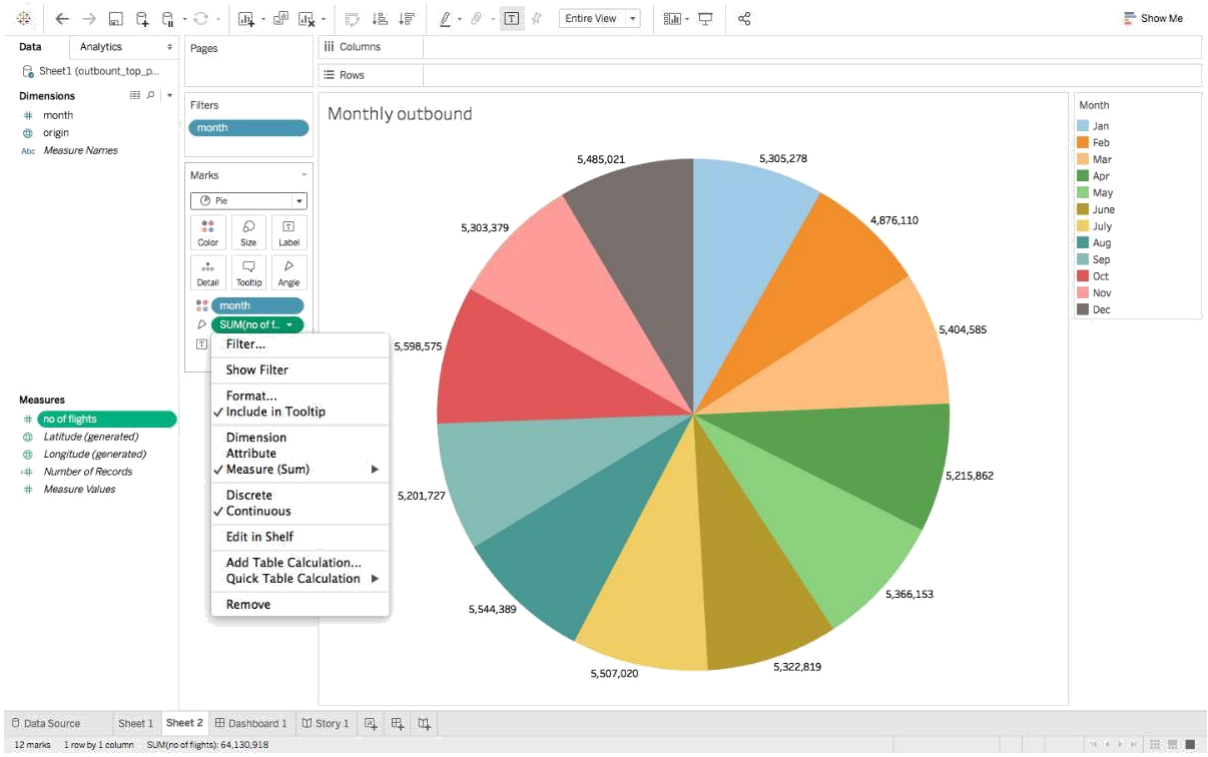


Change the geographic role of Origin as Airport. Drag Longitude(generated) to Columns, Latitude(generated) to Rows. Select Show me, and select Geo Map:

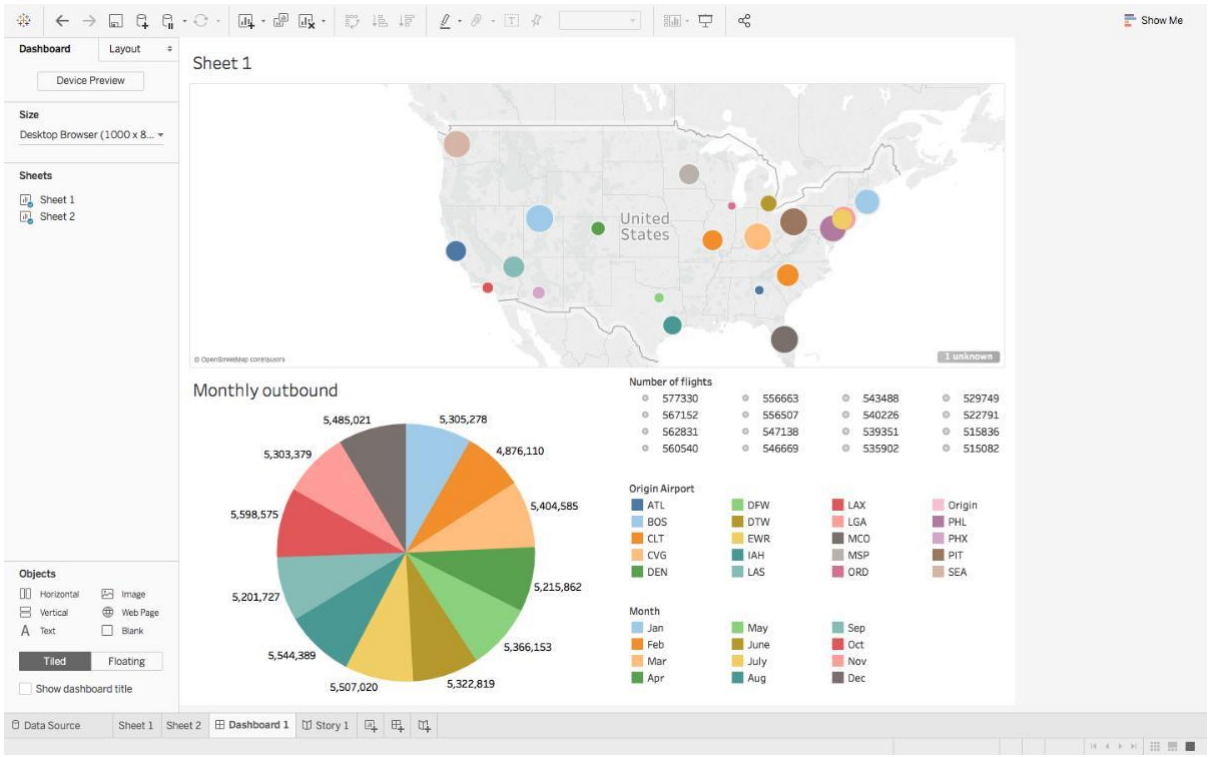


Click on Sheet 2 and in the mark functionality select pie diagram. Select months and no of flights and Filter month

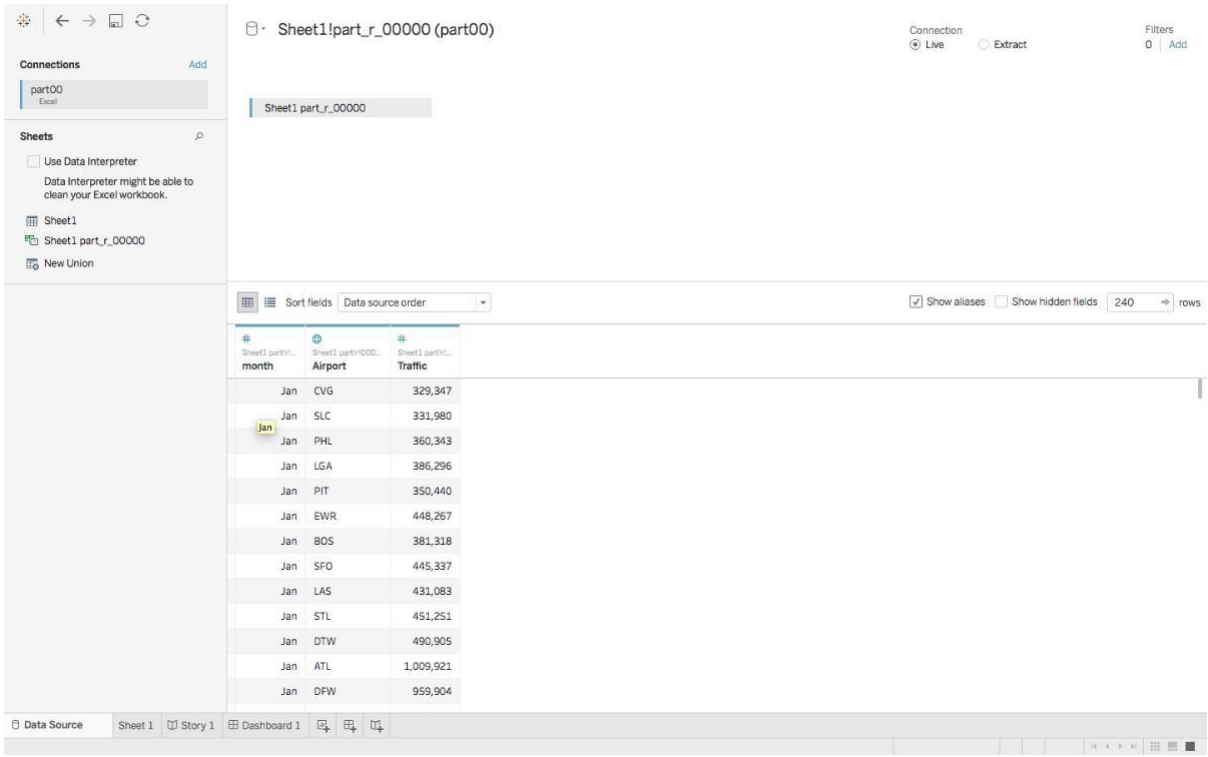




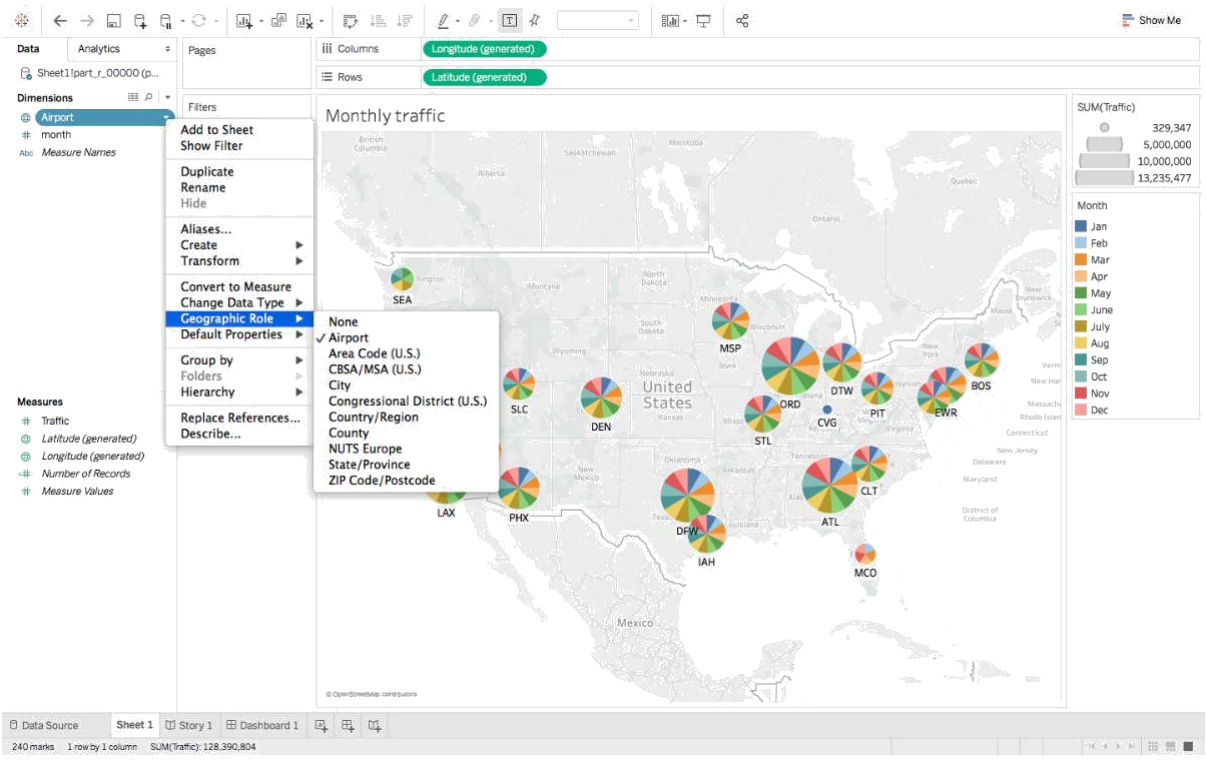
Open a dashboard by clicking next to sheet 2 and drag sheet 1 and Sheet 2 to the dashboard.



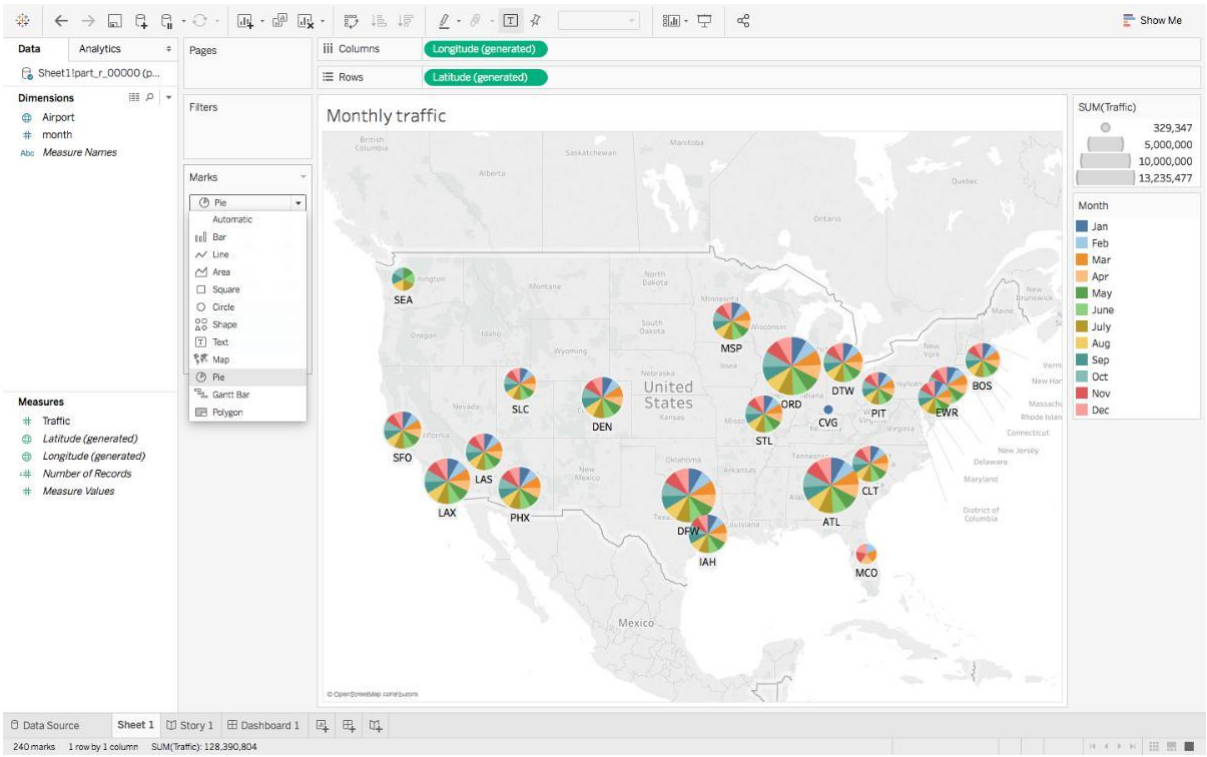
3. Monthly Traffic on Top 20 Airport.



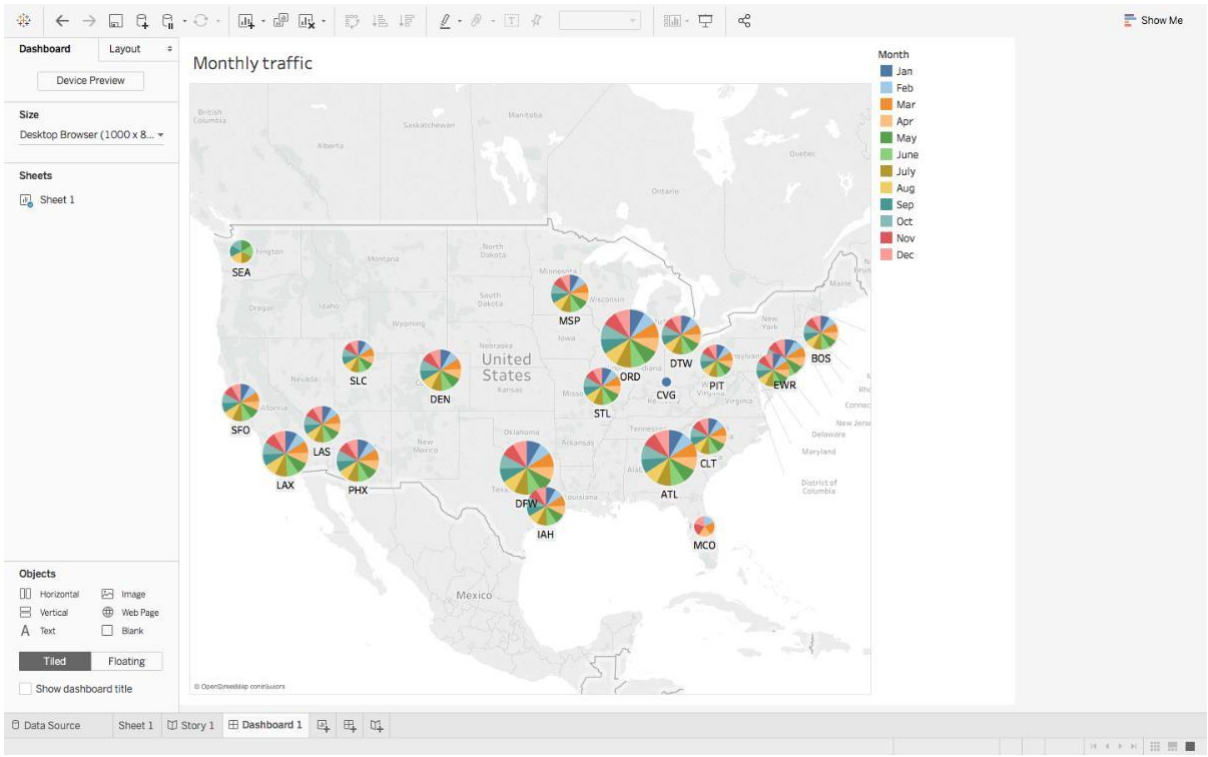
Select Sheet 1 next to Data Source, and change Airports geographic role to Airport
Drag Longitude(generated) to Columns, Latitude(generated) to Rows.



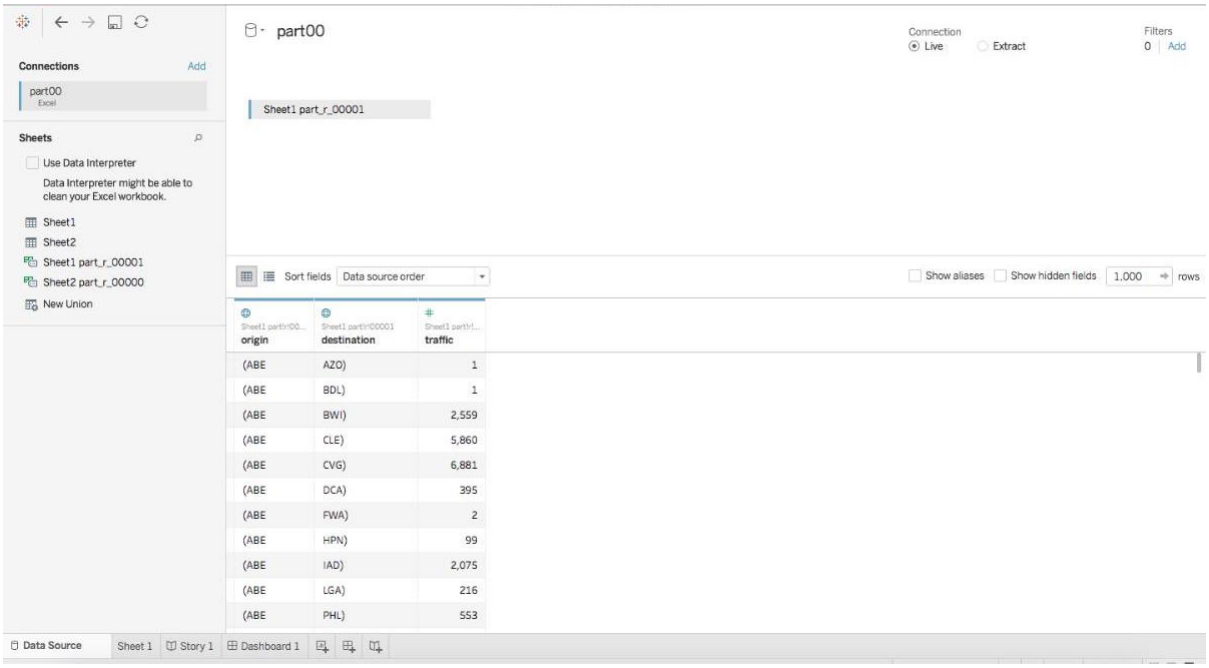
Create a new Worksheet by selecting the icon next to the Sheet 1. Drag Airport to marks and click on drop down menu to select pie chart, you will get this:



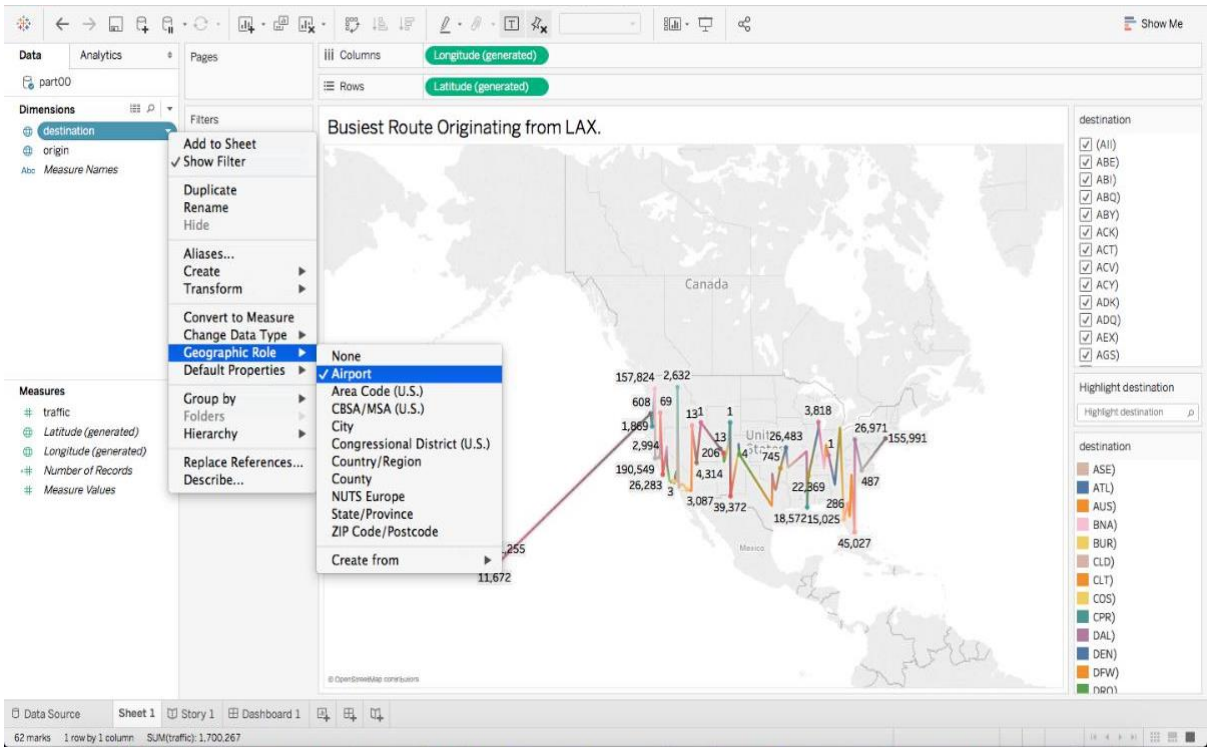
Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard.



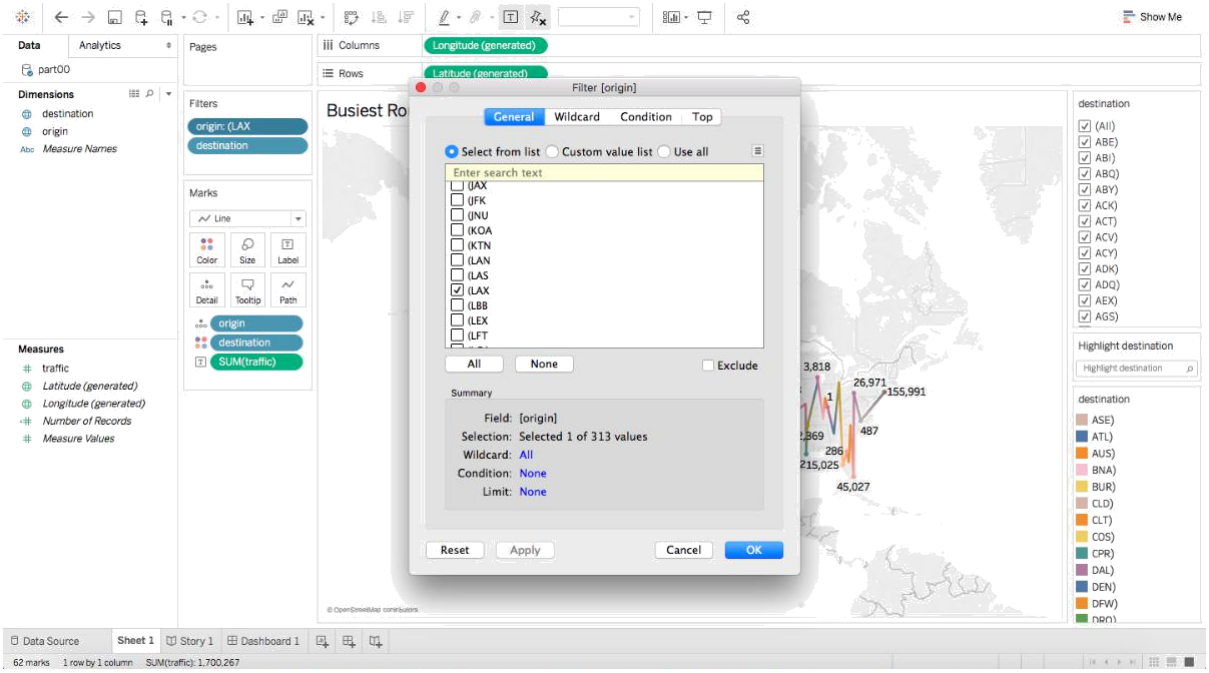
4. Arrival and departure from LAX airport.



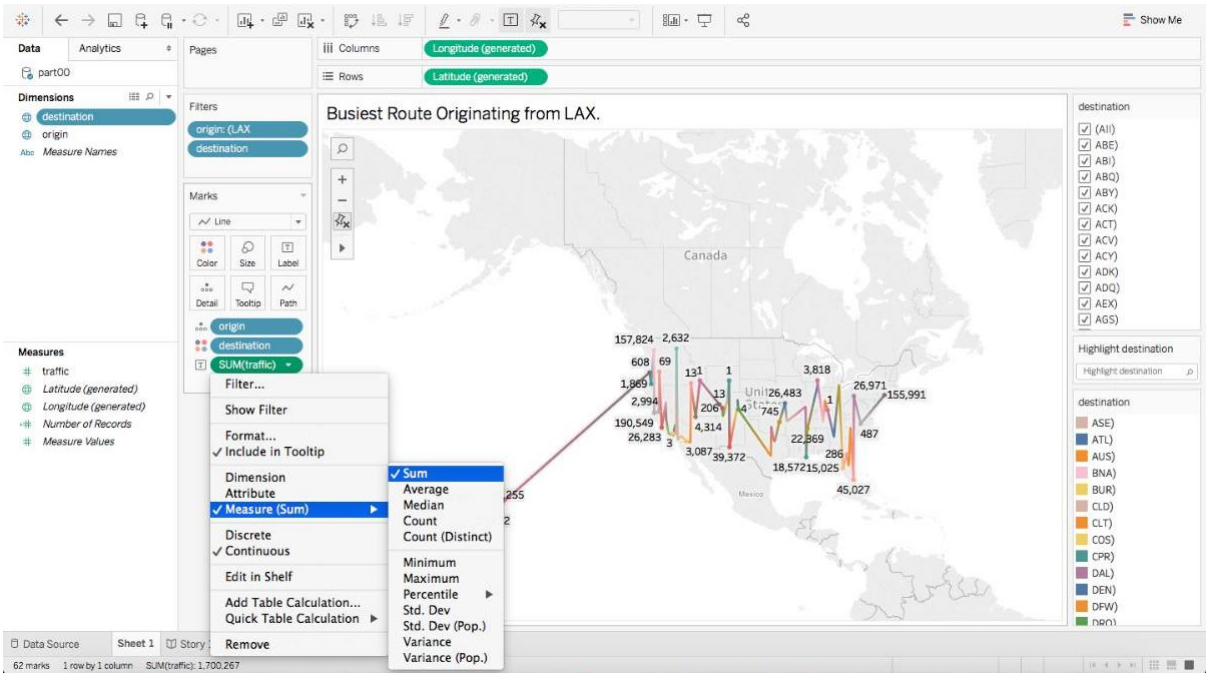
Select Sheet 1 next to Data Source, change State's geographical role of origin and destination to Airport. Drag Longitude to Columns, Latitude to Rows and select Geo Map



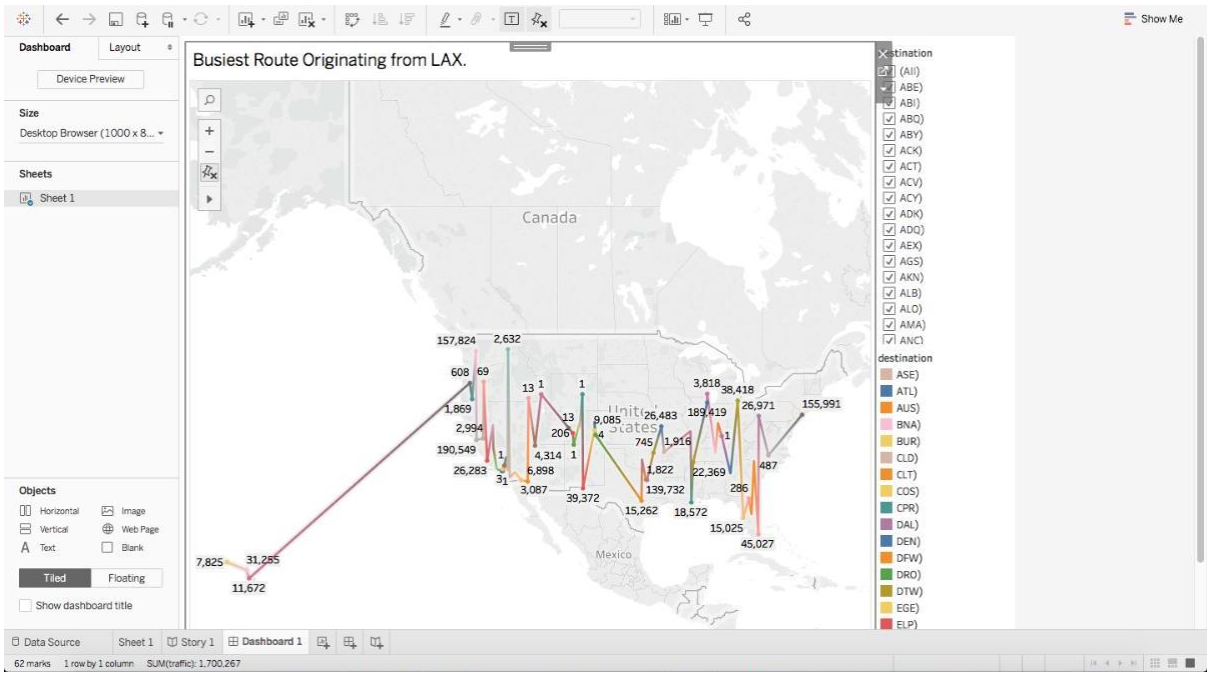
Create a new Worksheet by selecting the icon next to the Sheet 1. Use origin in filter and select LAX, you can use specific destination also in fliter as shown. But in this visualization, we have taken LAX as origin.



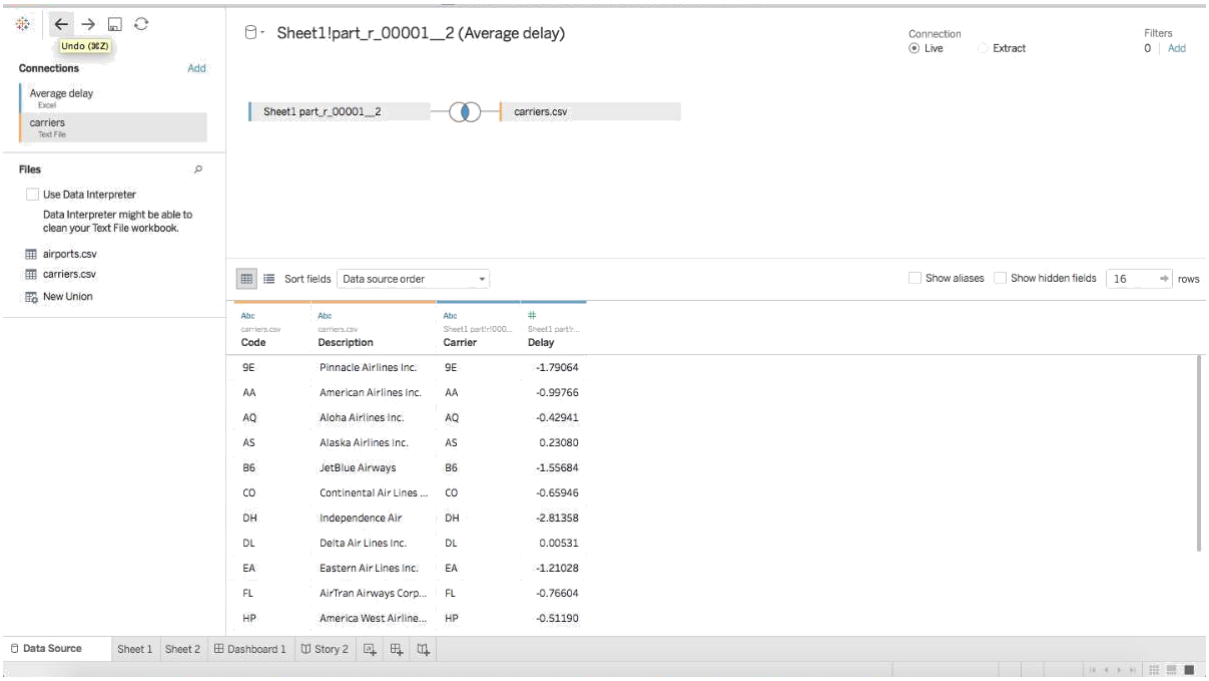
Change the measure of traffic to Sum as follow



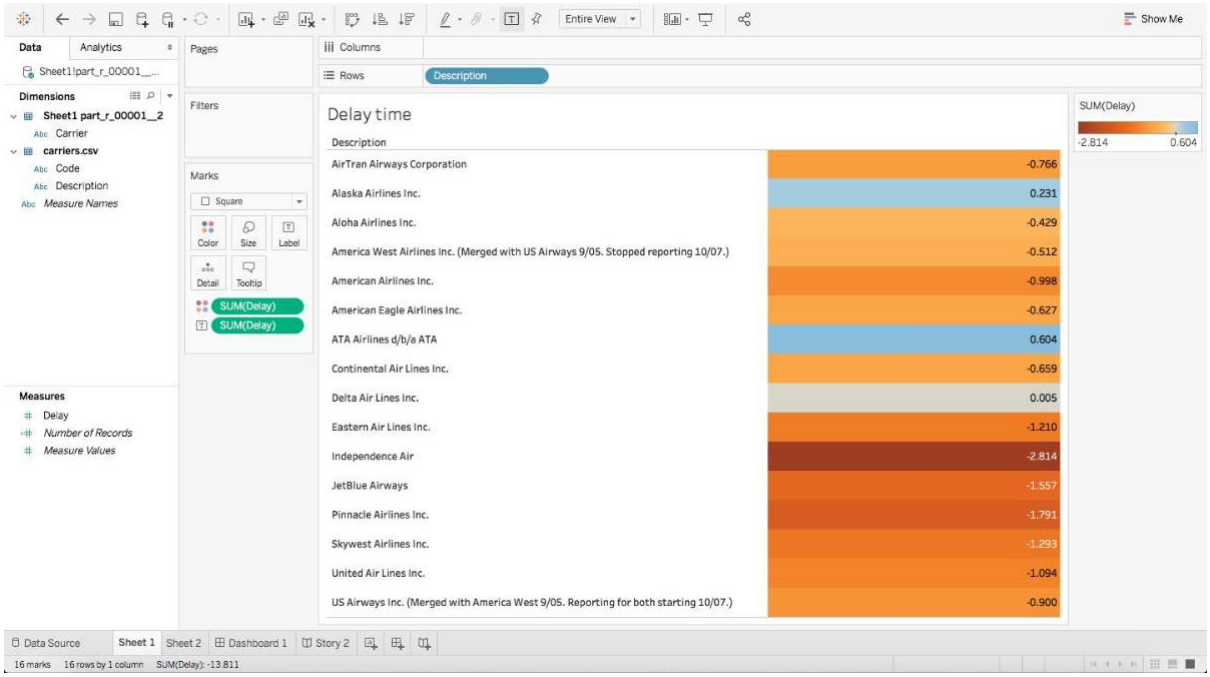
Open a dashboard by clicking next to sheet 2 and drag sheet 1 to the dashboard.



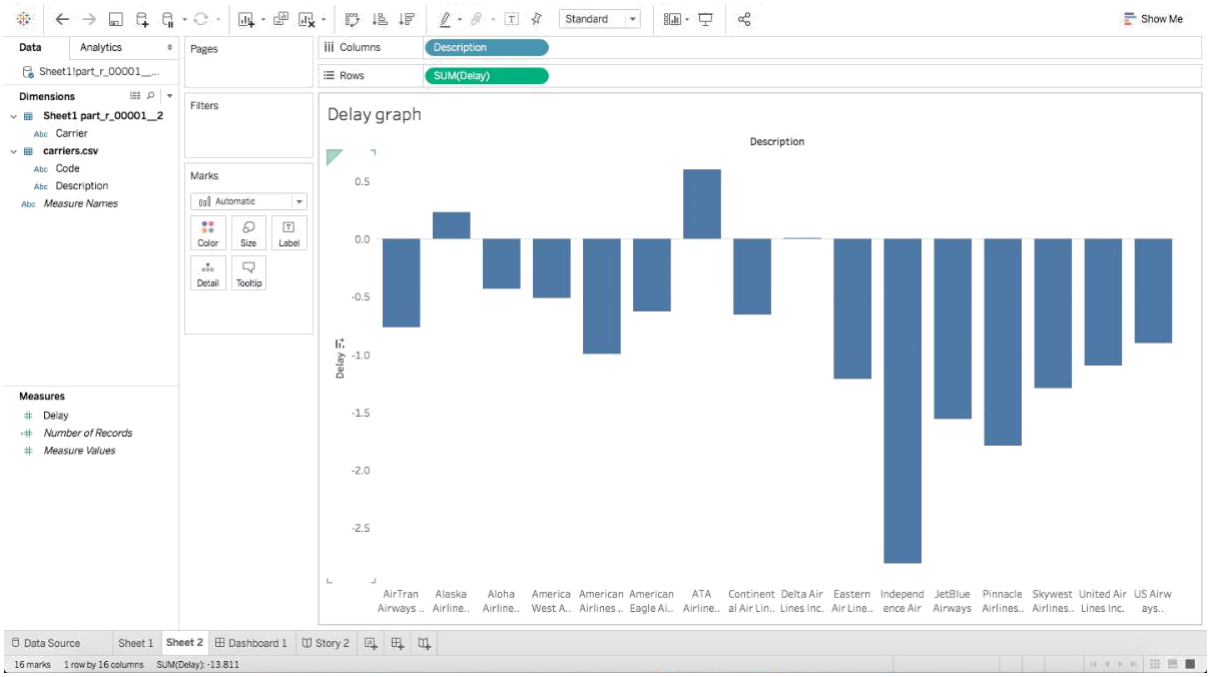
5. Average Delay



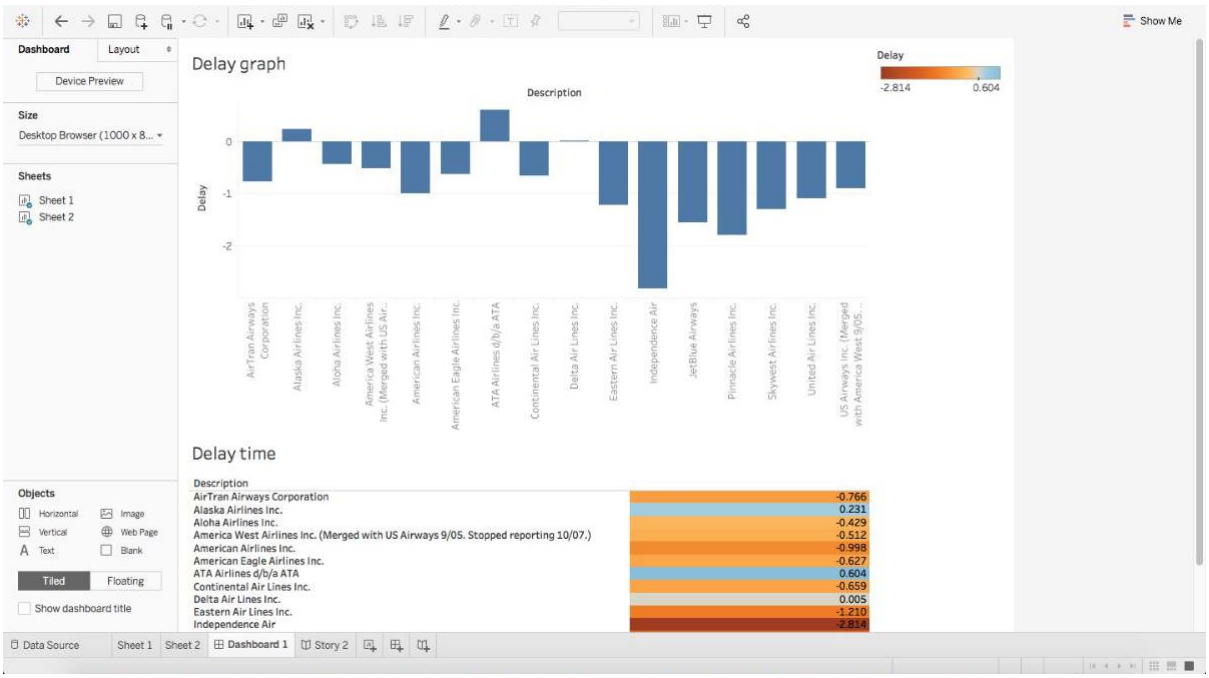
Select Sheet 1 next to Data Source, Delays to Color and again Delays to Text.



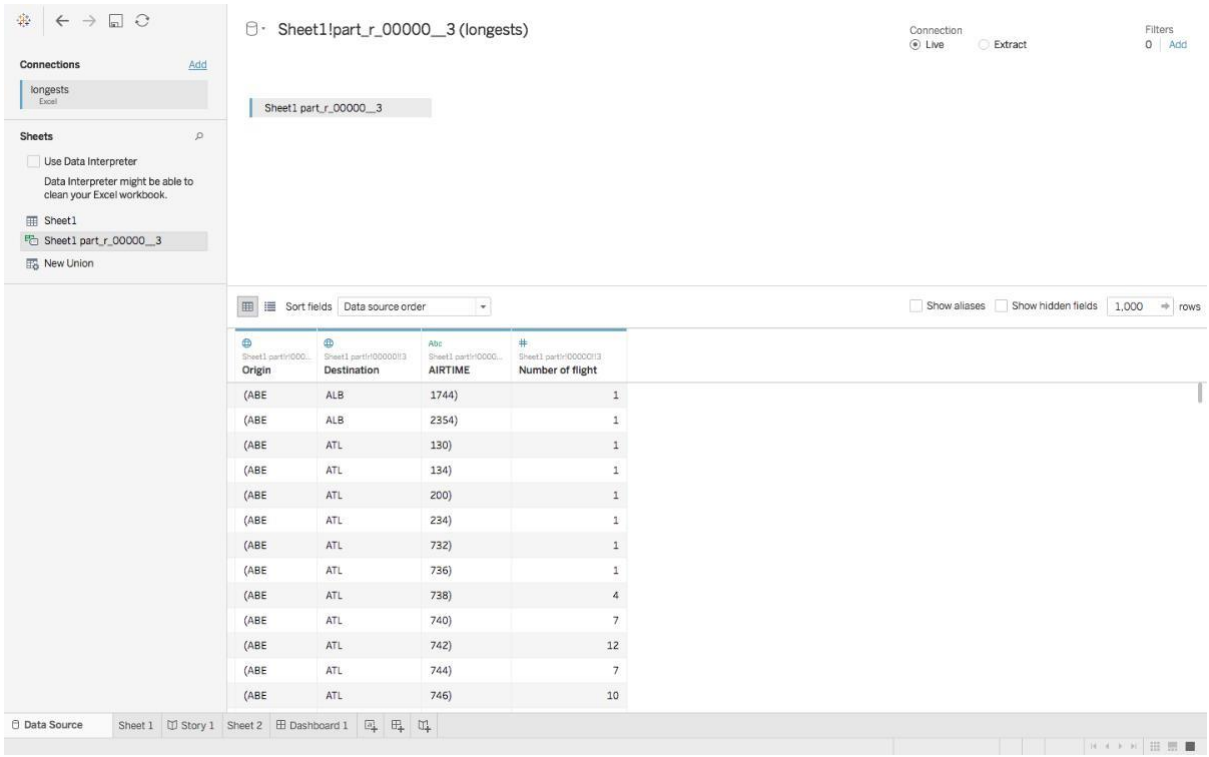
Create a new Worksheet by selecting the icon next to the Sheet 1. Drag Description to column and Delay to Rows.



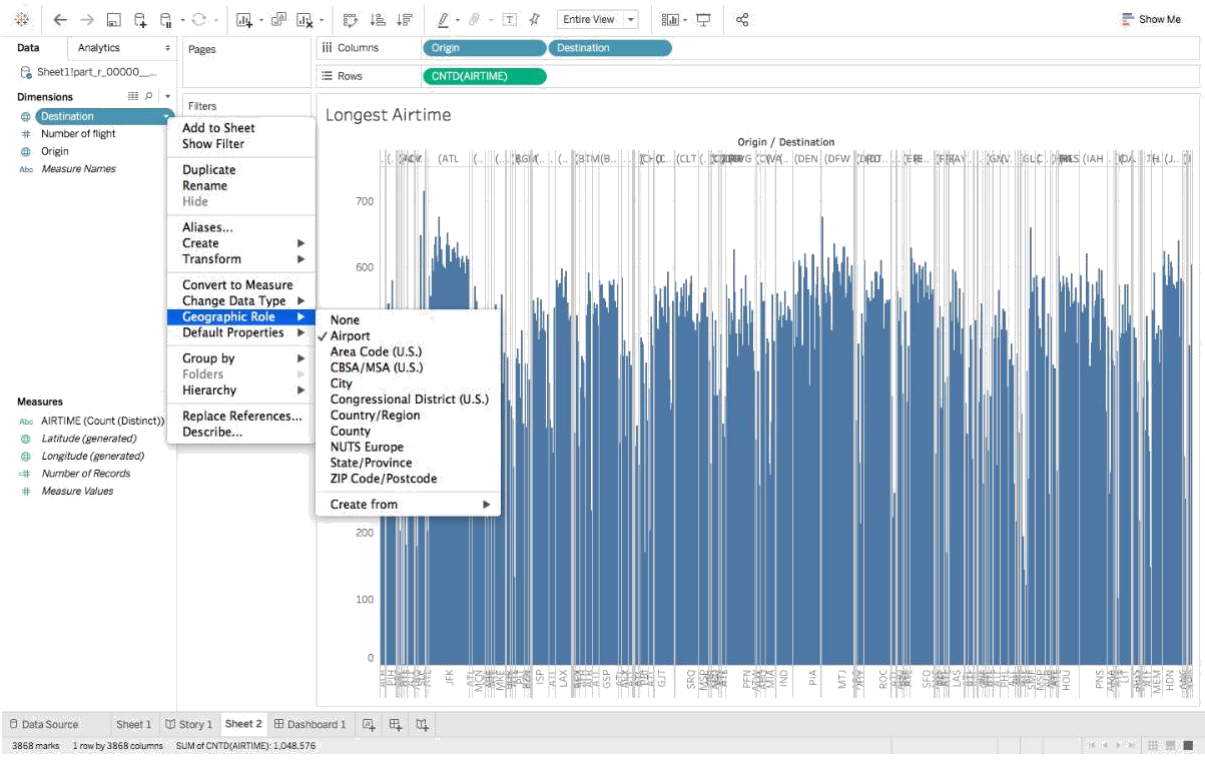
Open a dashboard by clicking next to sheet 2 and drag sheet 1 and sheet 2 to the dashboard.



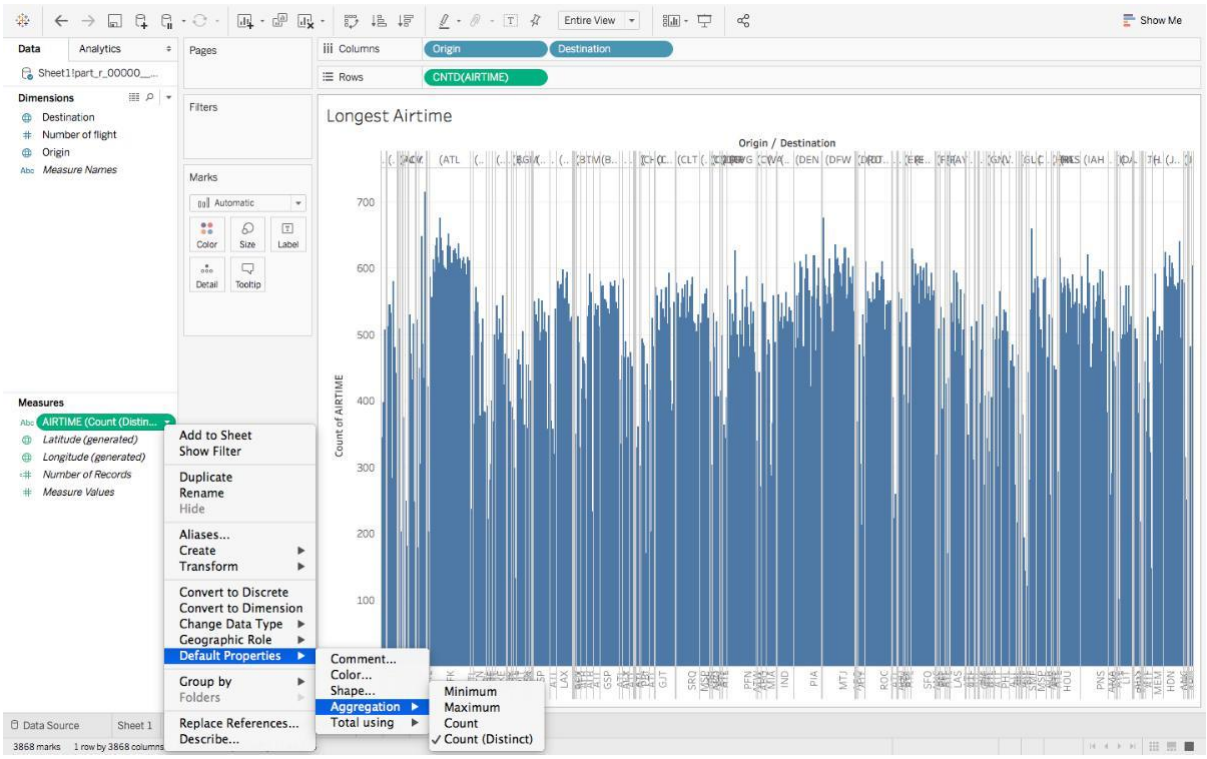
6. Longest flight by airtime



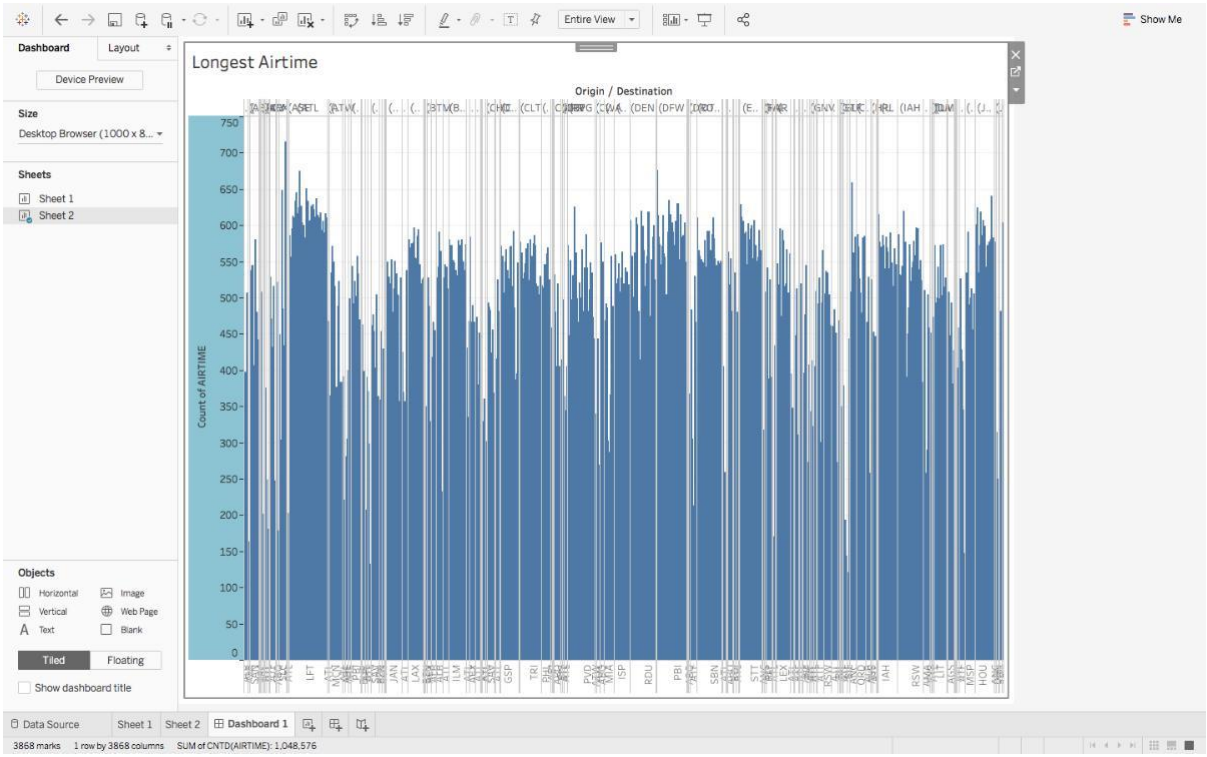
Select Sheet 1 next to Data Source, Change the geographic roles of origin and destination to Airport. Drag and drop origin and destination to column and airtime to rows and click on geo map.



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.



Conclusion

From the above experimental results, we can see that interesting sets of trends and patterns exists in large data sets which helps us to get a better understanding of the data. Airline Carrier which was the most popular in a given Year, By looking at this insight the airline specific marketing companies can narrow their products/services to specific carriers. Outbound flights from Top 20 airports on departure basis, airport authority of USA can narrow their traffic management resources at specific airports. Total flights from Top 20 airports on monthly traffic basis, the marketing companies can divide their funds for marketing of product/services on monthly basis when as the insight give the info about monthly traffic. Total flights originating from LAX to other airports, local LAX authorities have the info about the flights to other airports and can rework on the terminal allocation of flights on the basis of outbound traffic to certain airports. Carrier specific average delay, this insight will help the carriers to look at the competitions and users will have more knowledge about the delay timings of the carriers. Longest flight between two airports, the marketing companies can choose to showcase their product/services advertisement multiple times on AV devices due to longevity of the flight.

References

- 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/
- T.A. Jones, “Writing a good paper,” IEEE Trans. on *General Writing, Vol. 1, no. 2, pp.1-10, May 2002*.