CIS 5200 Term Project Tutorial

Flight Prices Analysis

Authors: Ragi Dave, An Mach, Ankita Hasmukhbhai Savaliya, Bhumika Suvagia

Instructor: Dr. Jongwook Woo

Date: 05/17/2023

Objectives

In this hands-on lab, you will learn how to:

- Upload the data into HDFS using -put command.
- Create Hive tables and perform data engineering in Hive using HiveQL
- Query data using HiveQL
- Secure copy tables created with HiveQL from Hadoop to local computers
- Create visualizations

Platform Specifications

hdfs version

```
-bash-4.2$ hdfs version
Hadoop 3.1.2
Source code repository ssh://git@bitbucket.oci.oraclecorp.com:7999/bdcs/apache_bigtop.git -r 4100eb8d8581c4328601079ff5af522f95e9977f
Compiled by root on 2023-02-27T08:26Z
Compiled with protoc 2.5.0
From source with checksum b367ca15864aef16725a3035859c9ece
This command was run using /usr/odh/1.1.5/hadoop/hadoop-common-3.1.2.jar
```

Lscpu

```
-bash-4.2$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GHz
Stepping:              4
CPU MHz:               1995.312
BogoMIPS:              3990.62
Virtualization:        VT-x
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
L3 cache:              16384K
NUMA node0 CPU(s):     0-7
```

hdfs dfsadmin -report

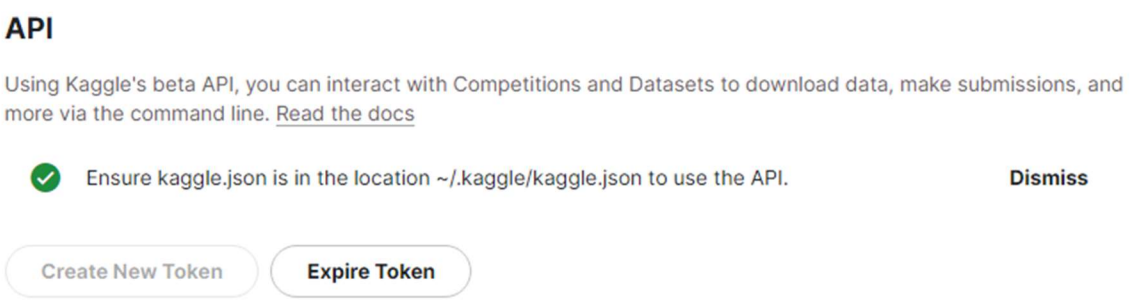
```
-bash-4.2$ hdfs dfsadmin -report
Configured Capacity: 419520548352 (390.71 GB)
Present Capacity: 417859894557 (389.16 GB)
DFS Remaining: 42330897693 (39.42 GB)
DFS Used: 375528996864 (349.74 GB)
DFS Used%: 89.87%
Replicated Blocks:
        Under replicated blocks: 0
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
report: Access denied for user amach3. Superuser privilege is required
```

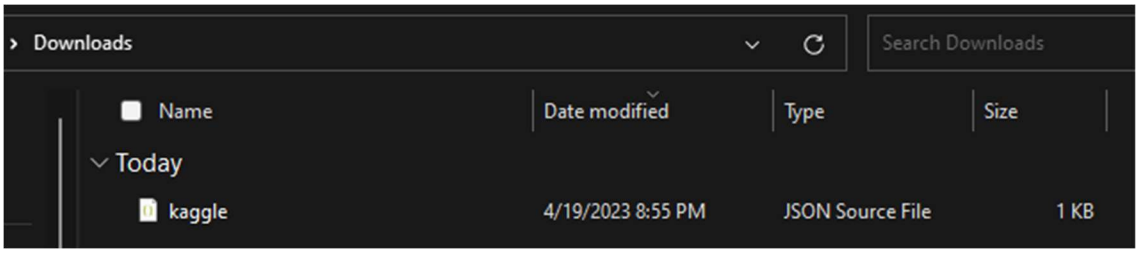
- Cluster Version: Hadoop 3.1.2
- CPU Speed: 1995.312 MHz
- Number of CPU cores: 8 cores * 5 nodes = 40 cores
- Number of nodes: 5 (2 Master and 3 Worker)
- Total Memory Size: 390.71 GB

Steps to download Kaggle datasets in Google Colab
(https://www.kaggle.com/general/156610)

1. Go to your kaggle account, Scroll to API section and Click Expire API Token to remove previous tokens

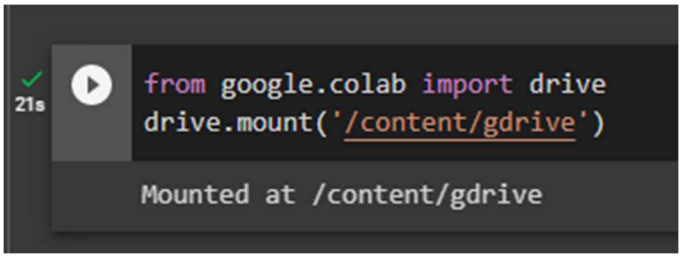
2. Click on Create New API Token - It will download kaggle.json file on your machine.





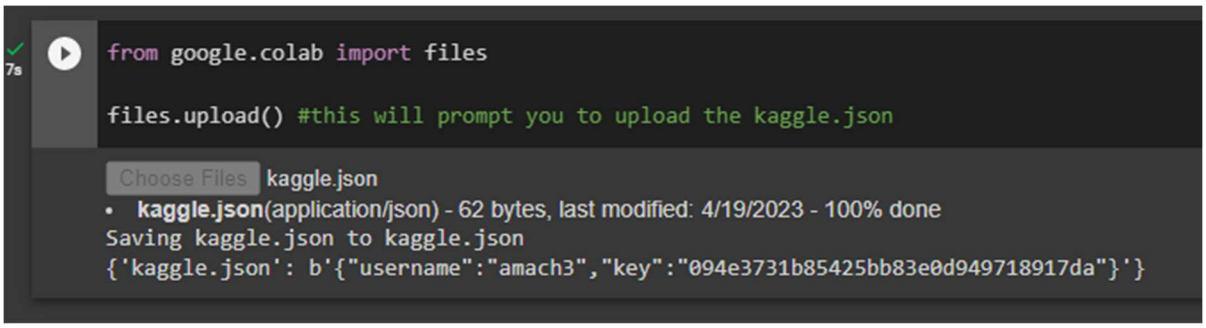
3. Go to your Google Colab project file and run the following commands to mount your Google Drive files Following code make mount your google drive

```
from google.colab import drive

drive.mount('/content/gdrive')
```



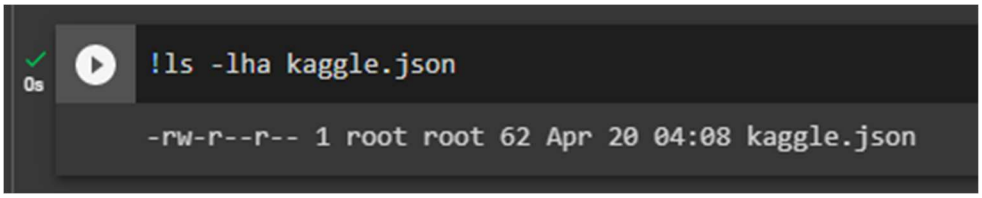
- Now upload the kaggle.json file

```
from google.colab import files

files.upload() #this will prompt you to upload the kaggle.json
```



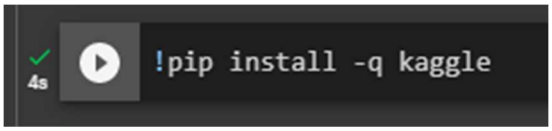
- make sure kaggle.json file is present

```
!ls -lha kaggle.json
```



- Install kaggle API client

```
!pip install -q kaggle
```

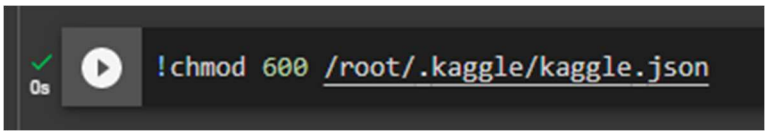


- kaggle API client expects the file to be in ~/.kaggle
- so move it there

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

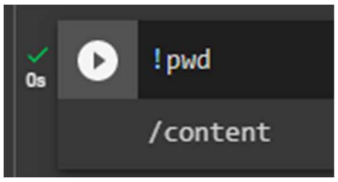
- we need to set permissions

```
!chmod 600 /root/.kaggle/kaggle.json
```



- check your directory before downloading the datasets

```
!pwd
```



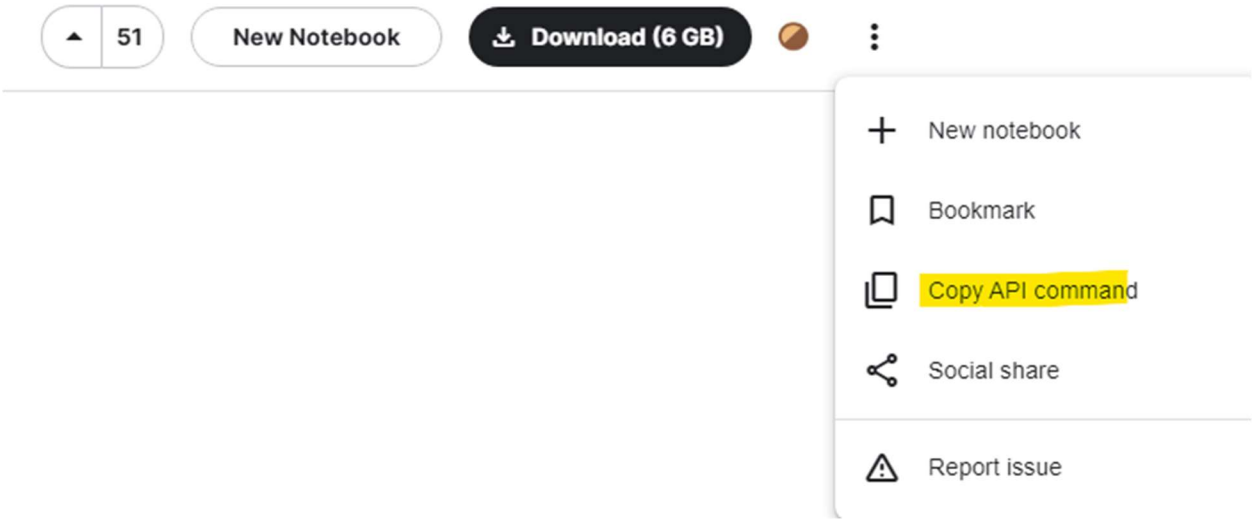
- list all available datasets

```
!kaggle datasets list
```

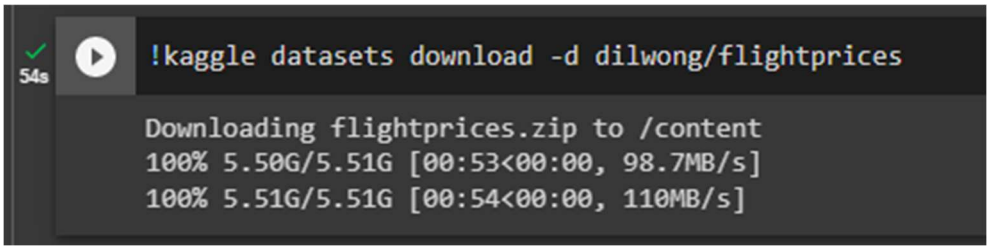
```
!kaggle datasets list
```

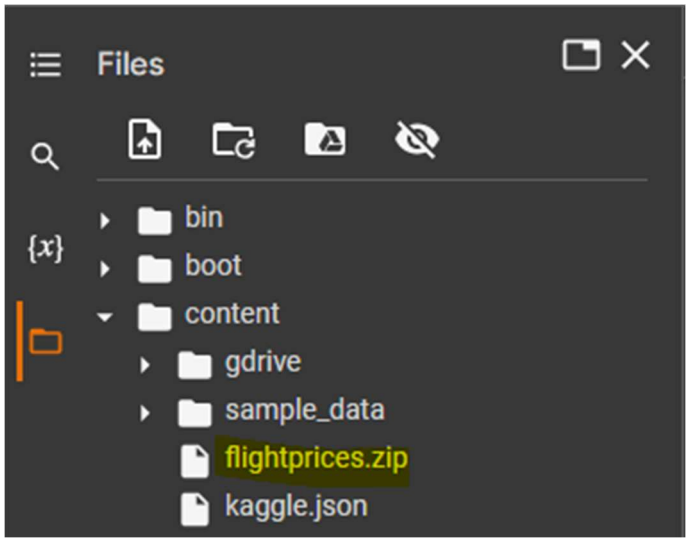
ref	title	size	lastUpdated	downloadCount	voteCount	usabilityRating
salvatorerastelli/spotify-and-youtube	Spotify and Youtube	9MB	2023-03-20 15:43:25	8181	301	1.0
ppb00x/country-gdp	Country_GDP	7KB	2023-04-07 06:47:36	1090	34	1.0
erdemtaha/cancer-data	Cancer Data	49KB	2023-03-22 07:57:00	3554	82	1.0
omartorres25/honda-data	Honda Cars Data	184KB	2023-03-28 04:19:11	1254	31	1.0
lokeshparab/amazon-products-dataset	Amazon Products Sales Dataset 2023	80MB	2023-03-26 10:45:19	3981	83	1.0
ulrikthygepedersen/fastfood-nutrition	Fastfood Nutrition	12KB	2023-03-21 10:02:41	3285	65	1.0
rkiattisak/student-performance-in-mathematics	Student performance prediction	9KB	2023-03-12 04:32:56	8949	188	1.0
ppb00x/credit-risk-customers	credit_risk_customers	18KB	2023-04-12 08:28:28	1147	34	1.0
arnabchaki/data-science-salaries-2023	Data Science Salaries 2023	25KB	2023-04-13 09:55:16	1615	42	1.0
kapturovalexander/nvidia-amd-intel-asus-msi-share-prices	NVIDIA, AMD, Intel, ASUS, MSI share prices (GPU)	902KB	2023-04-13 12:15:18	475	33	1.0
ashishraut64/internet-users	Global Internet users	163KB	2023-03-29 12:25:13	2229	56	1.0
rkiattisak/smart-watch-prices	Smart Watch prices	5KB	2023-04-12 06:04:23	609	23	1.0
arnabchaki/popular-video-games-1980-2023	Popular Video Games 1980 - 2023	1MB	2023-03-23 16:16:51	3847	107	1.0
ashishraut64/global-methane-emissions	Global Emissions.	31KB	2023-03-27 09:02:51	2787	57	1.0
tayyarhussain/best-selling-game-consoles-of-all-time	Best-Selling Gaming Consoles Dataset	1KB	2023-04-01 10:59:00	1011	34	1.0
shrutiambekar/smartphone-specifications-and-prices-in-india	Smartphone Specifications and Prices in India	36KB	2023-04-13 06:13:13	453	22	1.0
kapturovalexander/bitcoin-and-ethereum-prices-from-start-to-2023	Bitcoin & Ethereum prices (from start to 2023)	149KB	2023-04-09 06:07:57	824	30	1.0
r1chardson/the-world-university-rankings-2011-2023	THE World University Rankings 2011-2023	1MB	2023-04-03 12:43:37	1954	45	1.0
dgoenrique/netflix-movies-and-tv-shows	Netflix Movies and TV Shows	2MB	2023-03-13 18:49:00	3940	97	1.0
dansbecker/melbourne-housing-snapshot	Melbourne Housing Snapshot	451KB	2018-06-05 12:52:24	113094	1247	0.7058824

- download the required dataset from kaggle by copying the API command from the dataset page (https://www.kaggle.com/datasets/dilwong/flightprices?select=itineraries.csv)

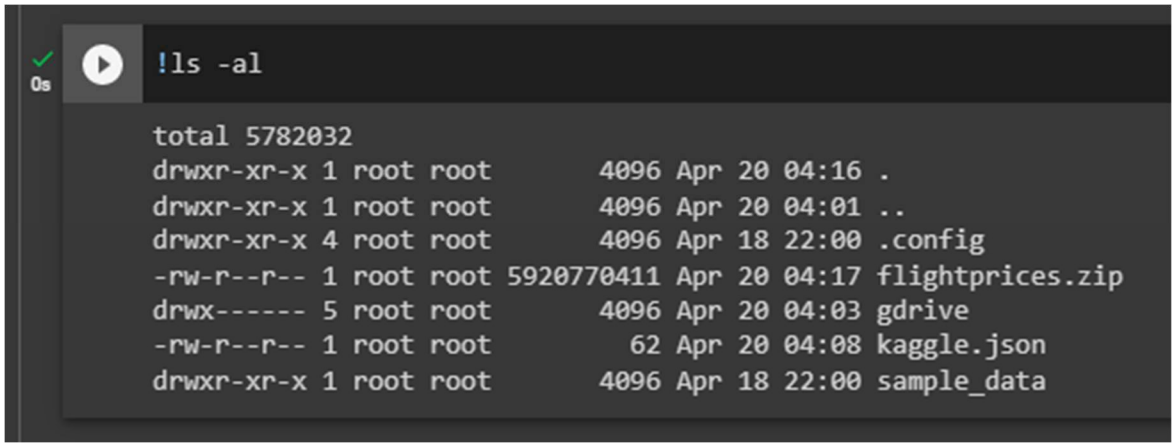



```
!kaggle datasets download -d dilwong/flightprices
```



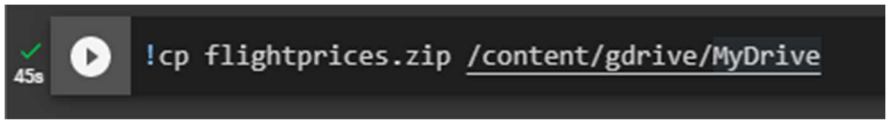


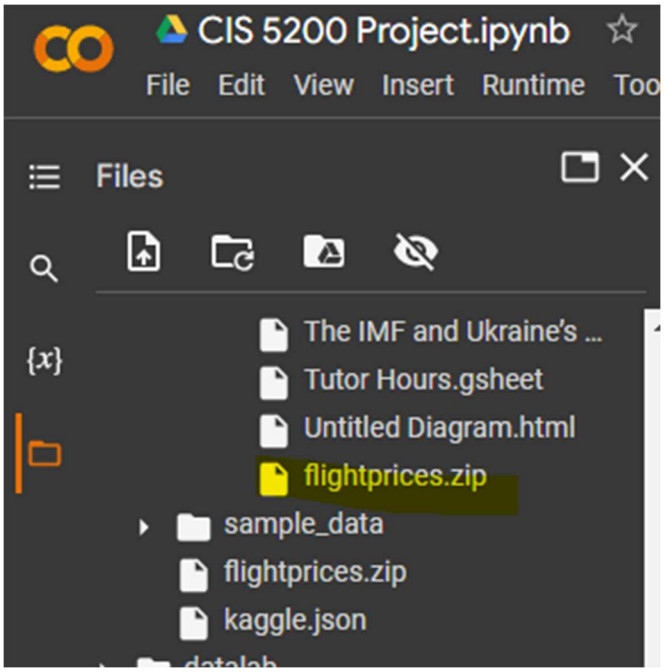
- check if the zip file is in the current working directory



- Copy the zip file to Google Drive

```
!cp flightprices.zip /content/gdrive/MyDrive
```





Note: We realized the above steps are not necessary in this project. We could have downloaded the zip file directly from Kaggle then unzip the file on a local computer.

Secure copy csv files from my local computer to Linux server

- Download the flightprices.zip file from Google Drive then unzip it in Download folder, then secure copy it to tmp folder on Linux server.

```
scp -C C:/Users/jiazh/Downloads/flightprices/itineraries.csv amach3@144.24.53.159:/tmp
```

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp -C C:/Users/jiazh/Downloads/flightprices/itineraries.csv amach3@144.24.53.159:/tmp/
amach3@144.24.53.159's password:
itineraries.csv                                100%   29GB   6.9MB/s 1:11:44
```

- Download the airport.csv file from Kaggle and unzip it in Download folder. Then secure copy it to linux.

```
scp C:/Users/jiazh/Downloads/airportcode/airports.csv amach3@144.24.53.159:/home/amach3
```

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp C:/Users/jiazh/Downloads/airportcode/airports.csv amach3@144.24.53.159:/home/amach3
amach3@144.24.53.159's password:
airports.csv                                   100%  567KB 809.9KB/s   00:00
```

Copy files from linux server to hdfs

- Make FlightData directory and put itineraries.csv file here

```
hdfs dfs -mkdir FlightData
hdfs dfs -put /tmp/itineraries.csv /user/amach3/FlightData/
hdfs dfs -ls FlightData/
```

```
-bash-4.2$ hdfs dfs -ls FlightData/
Found 1 items
-rw-r--r--   3 amach3 hdfs 31091834438 2023-04-23 14:55 FlightData/itineraries.csv
```

- Make the FlightData directory public

```
hdfs dfs -chmod -R go+rx /user/amach3/FlightData
```

- Make Airport directory and put Airport.csv file here

```
hdfs dfs -mkdir Airport
hdfs dfs -put airports.csv Airport/
hdfs dfs -ls Airport/
```

```
-bash-4.2$ hdfs dfs -mkdir Airport
hdfs dfs -ls Airport/
-bash-4.2$ hdfs dfs -put airports.csv Airport/
-bash-4.2$ hdfs dfs -ls Airport/
Found 1 items
-rw-r--r--   3 amach3 hdfs     580468 2023-04-21 03:49 Airport/airports.csv
-bash-4.2$
```

- make the Airport directory public

```
hdfs dfs -chmod -R go+rx /user/amach3/Airport
```

Create tables in Hive

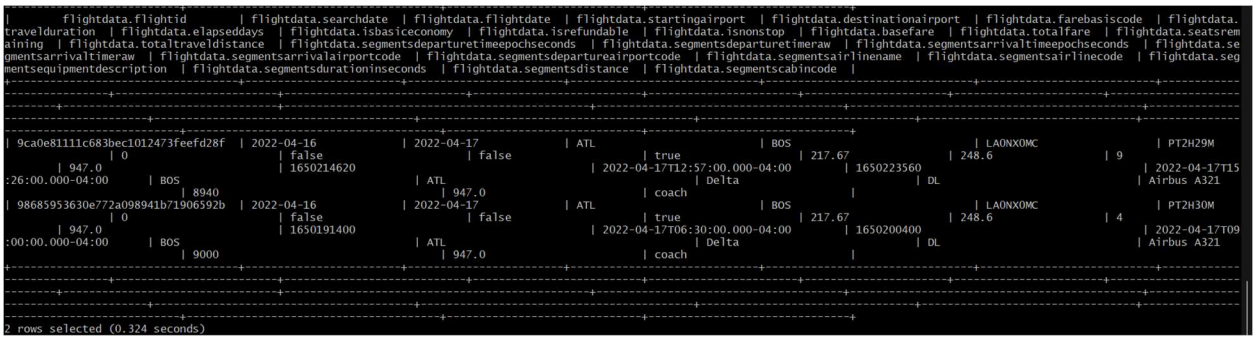
```
Beeline
Use amach3;


DROP TABLE IF EXISTS Airport;


CREATE EXTERNAL TABLE IF NOT EXISTS Airport
(Name string, City string, Country string, IATA string, ICAO string,
Latitude double, Longitude double)
```

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/amach3/Airport/'
TBLPROPERTIES ('skip.header.line.count'='1');

SELECT * FROM Airport LIMIT 10;
```

airport.name	airport.city	airport.country	airport.iata	airport.icao	airport.latitude	airport.longitude
Goroka Airport	Goroka	Papua New Guinea	GKA	AYGA	-6.081689835	145.3919983
Madang Airport	Madang	Papua New Guinea	MAG	AYMD	-5.207079887	145.7890015
Mount Hagen Kagamuga Airport	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.826789856	144.2960052
Nadzab Airport	Nadzab	Papua New Guinea	LAE	AYNZ	-6.569803	146.725977
Port Moresby Jacksons International Airport	Port Moresby	Papua New Guinea	POM	AYPY	-9.443380356	147.2200012
Wewak International Airport	Wewak	Papua New Guinea	WWK	AYWK	-3.583830118	143.6690063
Narsarsuaq Airport	Narssarssuaq	Greenland	UAK	BGBW	61.16049957	-45.42599869
Godthaab / Nuuk Airport	Godthaab	Greenland	GOH	BGGH	64.19090271	-51.67810059
Kangerlussuaq Airport	Sondrestrom	Greenland	SFJ	BGSF	67.0122219	-50.71160316
Thule Air Base	Thule	Greenland	THU	BGTL	76.53119659	-68.70320129

```
DROP TABLE IF EXISTS FlightData;

CREATE EXTERNAL TABLE IF NOT EXISTS FlightData
(FlightID string, SearchDate date, FlightDate date, StartingAirport string,
DestinationAirport string, FareBasisCode string, TravelDuration string,
ElapsedDays int, IsBasicEconomy boolean, IsRefundable boolean,
IsNonStop boolean, BaseFare double, TotalFare double, SeatsRemaining int,
TotalTravelDistance double, SegmentsDepartureTimeEpochSeconds string,
SegmentsDepartureTimeRaw string, SegmentsArrivalTimeEpochSeconds string,
SegmentsArrivalTimeRaw string, SegmentsArrivalAirportCode string,
SegmentsDepartureAirportCode string, SegmentsAirlineName string,
SegmentsAirlineCode string, SegmentsEquipmentDescription string,
SegmentsDurationInSeconds string, SegmentsDistance double,
SegmentsCabinCode string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/amach3/FlightData/'
TBLPROPERTIES ('skip.header.line.count'='1');
```

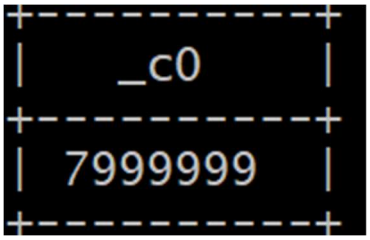
```
SELECT * FROM FlightData LIMIT 2;
```



Since the dataset is too large, Dr. Woo advised us to reduce our data size to 2 – 3 GB by sampling our data.

```
select count(*) from FlightData; --82138753 rows

--10% is about 8,000,000 rows


CREATE EXTERNAL TABLE IF NOT EXISTS FlightData2

(FlightID string, SearchDate date, FlightDate date, StartingAirport string,

DestinationAirport string, FareBasisCode string, TravelDuration string,

ElapsedDays int, IsBasicEconomy boolean, IsRefundable boolean,

IsNonStop boolean, BaseFare double, TotalFare double, SeatsRemaining int,

TotalTravelDistance int, SegmentsDepartureTimeEpochSeconds string,

SegmentsDepartureTimeRaw string, SegmentsArrivalTimeEpochSeconds string,

SegmentsArrivalTimeRaw string, SegmentsArrivalAirportCode string,

SegmentsDepartureAirportCode string, SegmentsAirlineName string,

SegmentsAirlineCode string, SegmentsEquipmentDescription string,

SegmentsDurationInSeconds string, SegmentsDistance int,

SegmentsCabinCode string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/amach3/FlightData2/'

TBLPROPERTIES ('skip.header.line.count'='1');


INSERT OVERWRITE TABLE FlightData2
```

```
select * from FlightData
where rand() <= 0.1
distribute by rand()
sort by rand()
limit 8000000;

SELECT COUNT(*) from FlightData2;
```

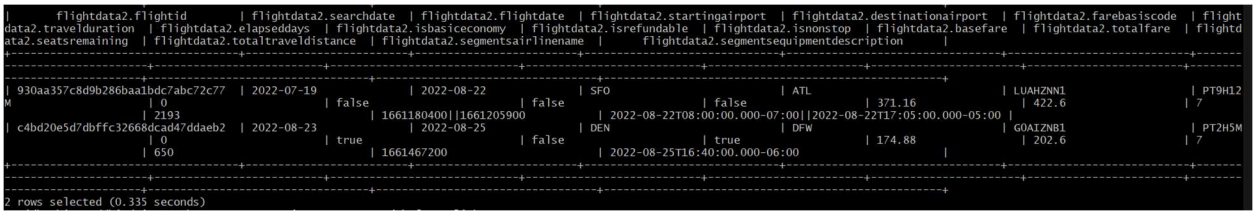


_c0
7999999

Data Engineering

1. Drop unusable columns

```
ALTER TABLE FlightData2 REPLACE COLUMNS (FlightID string, SearchDate date, FlightDate date, StartingAirport string,
DestinationAirport string, FareBasisCode string, TravelDuration string,
ElapsedDays int, IsBasicEconomy boolean, IsRefundable boolean,
IsNonStop boolean, BaseFare double, TotalFare double, SeatsRemaining int,
TotalTravelDistance int, SegmentsAirlineName string, SegmentsEquipmentDescription string);
SELECT * FROM FlightData2 LIMIT 2;
```



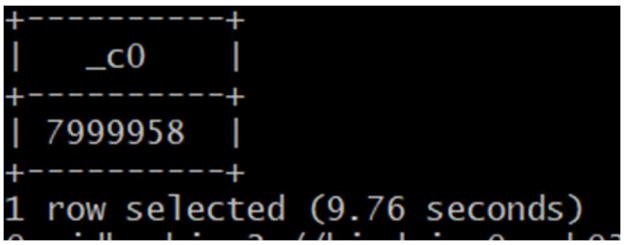
flightdata2.flightid	flightdata2.searchdate	flightdata2.flightdate	flightdata2.startingairport	flightdata2.destinationairport	flightdata2.farebasiscode	flightdata2.travelduration	flightdata2.elapseddays	flightdata2.isbasiceconomy	flightdata2.isrefundable	flightdata2.isnonstop	flightdata2.basefare	flightdata2.totalfare	flightdata2.seatsremaining	flightdata2.totaltraveldistance	flightdata2.segmentsairlinename	flightdata2.segmentsequipmentdescription
930aa357c8d9b286baa1bdc7abc72c77	2022-07-19	2022-08-22	SFO	ATL	LUAHZNN1	PT9H12M	0	false	false	false	371.16	422.6	7	2193	1661180400||1661205900	2022-08-22T08:00:00.000-07:00||2022-08-22T17:05:00.000-05:00
c4bd20e5d7dbffc32668dcad47ddaeb2	2022-08-23	2022-08-25	DEN	DFW	G0AIZNB1	PT2H5M	0	true	false	true	174.88	202.6	7	650	1661467200	2022-08-25T16:40:00.000-06:00

2 rows selected (0.335 seconds)

2. Remove any duplicate rows from FlightData2 table

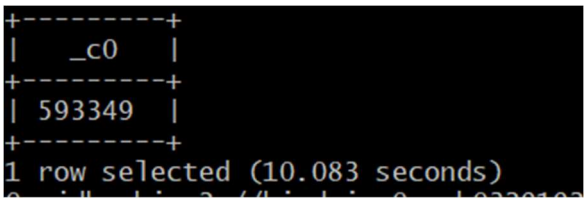
```
INSERT OVERWRITE TABLE  FlightData2
```

```
SELECT DISTINCT * FROM FlightData2;
-- Check number of rows after removing duplicates
SELECT COUNT(*) from FlightData2;
```

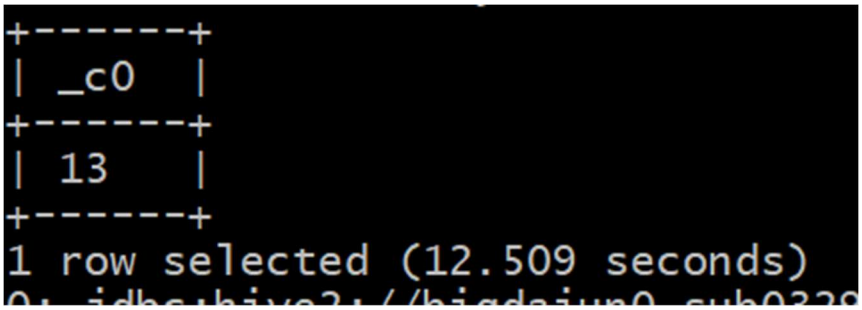


3. Check null values

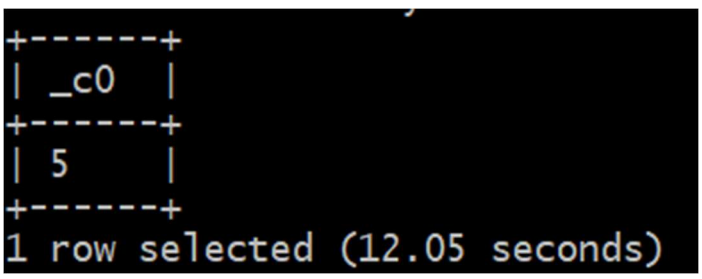
```
SELECT count(*) from FlightData2 where TotalTravelDistance IS NULL;
```



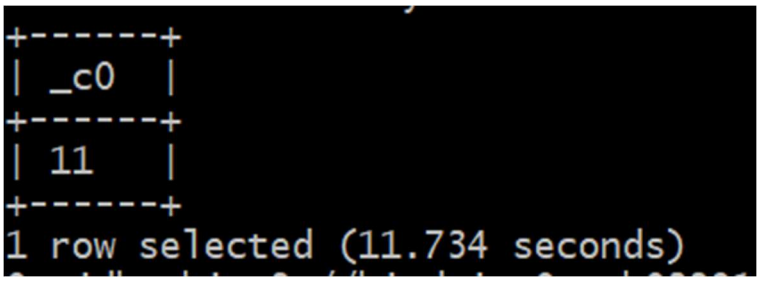

```
SELECT count(*) from FlightData2 where FlightDate IS NULL;
```



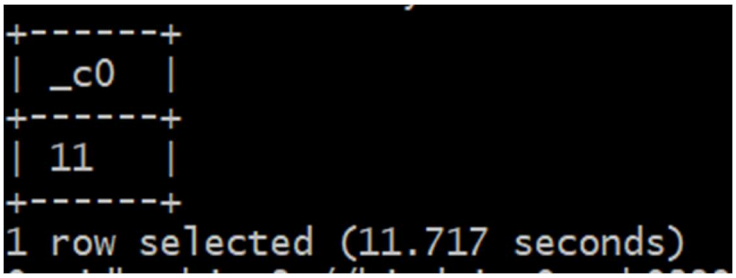

```
SELECT count(*) from FlightData2 where StartingAirport IS NULL;
```



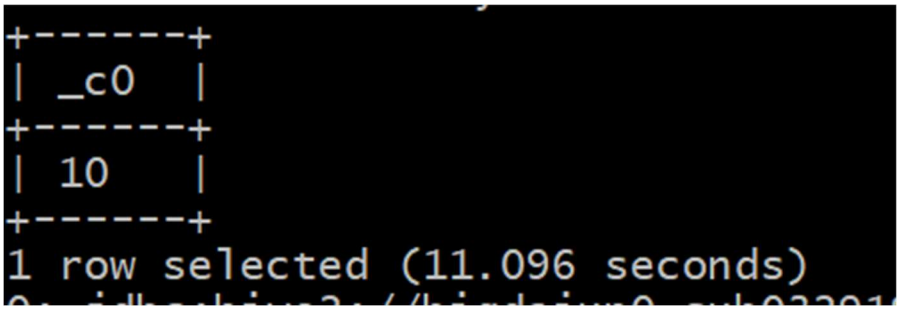
SELECT count(*) from FlightData2 where DestinationAirport IS NULL;



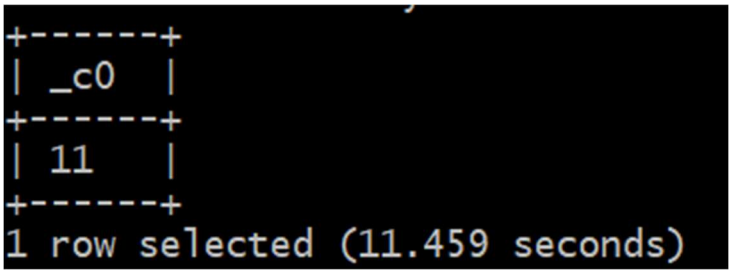
SELECT count(*) from FlightData2 where TravelDuration IS NULL;



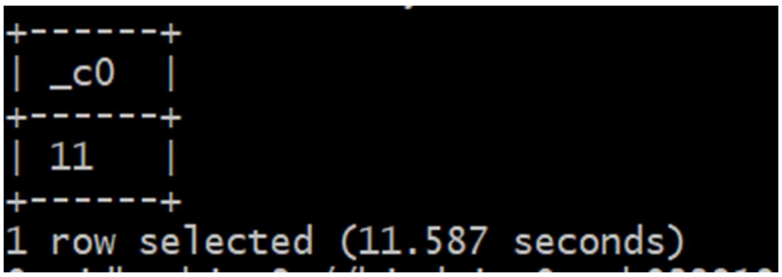
SELECT count(*) from FlightData2 where IsBasicEconomy IS NULL;



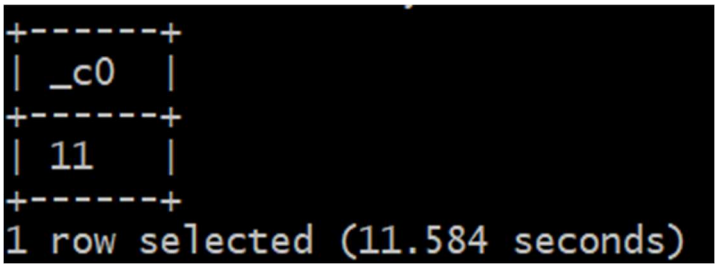
SELECT count(*) from FlightData2 where IsNonStop IS NULL;



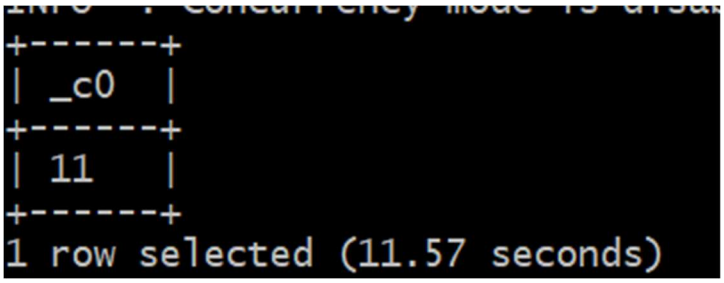
SELECT count(*) from FlightData2 where BaseFare IS NULL;



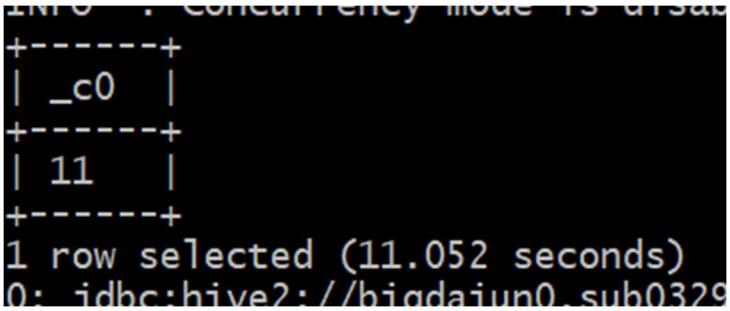
SELECT count(*) from FlightData2 where TotalFare IS NULL;



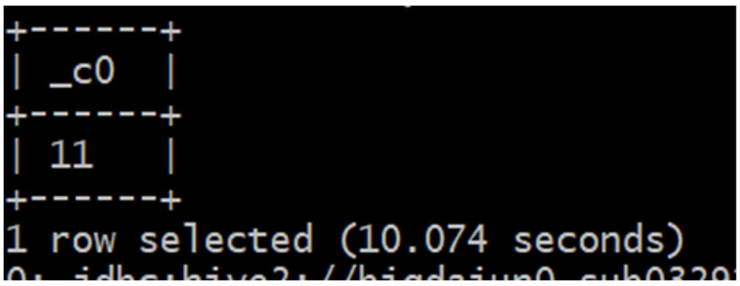
SELECT count(*) from FlightData2 where SeatsRemaining IS NULL;



SELECT count(*) from FlightData2 where SegmentsAirlineName IS NULL;



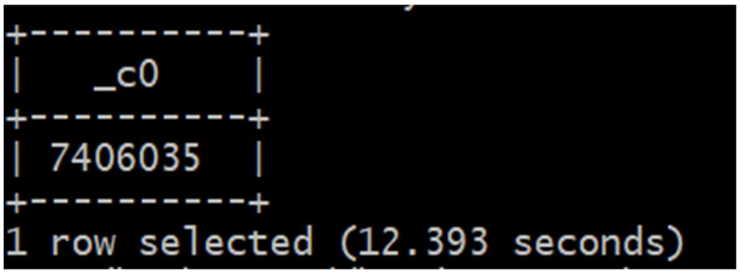
```
SELECT count(*) from FlightData2 where SegmentsEquipmentDescription IS NULL;
```



Since our dataset is huge (almost 8 million rows of data), we decided to remove rows with null values detected above.

```
INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE TotalTravelDistance IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE FlightDate IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE StartingAirport IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE DestinationAirport IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE TravelDuration IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE IsBasicEconomy IS NOT NULL;
```

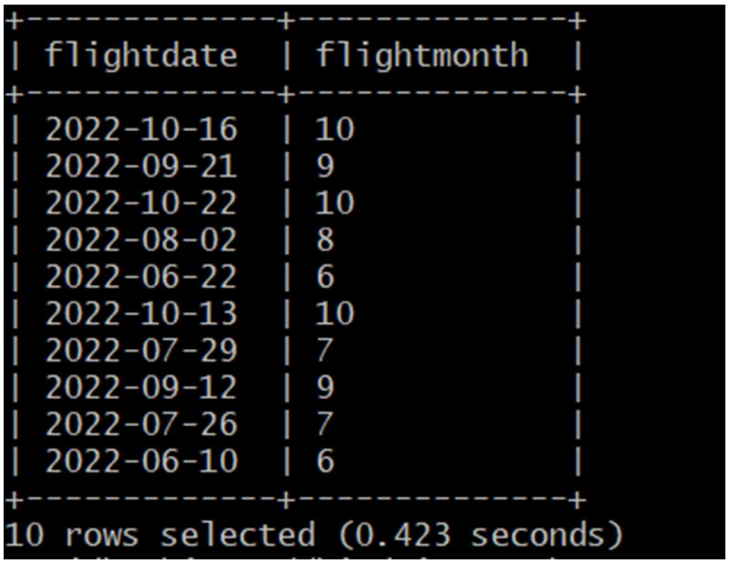
```
INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE IsNonStop IS NOT NULL;
INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE BaseFare IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE TotalFare IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE SeatsRemaining IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE SegmentsAirlineName IS NOT NULL;

INSERT OVERWRITE TABLE FlightData2
SELECT * FROM FlightData2 WHERE SegmentsEquipmentDescription IS NOT NULL;

SELECT COUNT(*) from FlightData2;
```



4. Add FlightMonth column by getting the month from FlightDate column

```
ALTER TABLE FlightData2 ADD COLUMNS (FlightMonth int);

INSERT OVERWRITE TABLE FlightData2
```

```
SELECT FlightID, SearchDate, FlightDate, StartingAirport,
DestinationAirport, FareBasisCode string, TravelDuration,
ElapsedDays, IsBasicEconomy, IsRefundable,
IsNonStop, BaseFare, TotalFare, SeatsRemaining,
TotalTravelDistance, SegmentsAirlineName , SegmentsEquipmentDescription,
MONTH(FlightDate) AS FlightMonth FROM FlightData2;


Select FlightDate, FlightMonth from FlightData2 LIMIT 10;
```



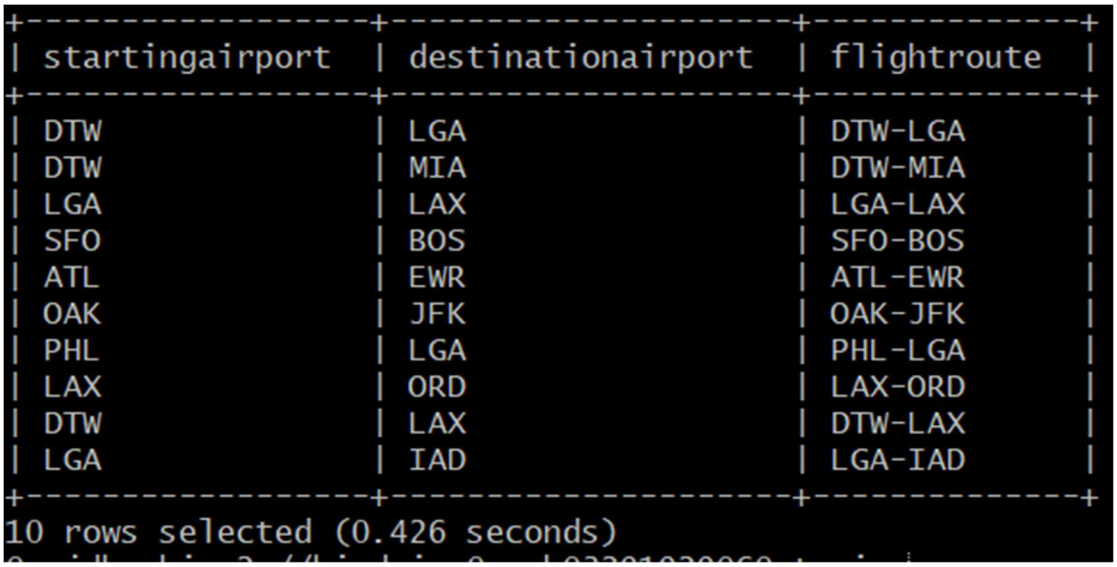
flightdate	flightmonth
2022-10-16	10
2022-09-21	9
2022-10-22	10
2022-08-02	8
2022-06-22	6
2022-10-13	10
2022-07-29	7
2022-09-12	9
2022-07-26	7
2022-06-10	6

10 rows selected (0.423 seconds)

5. Add a Flight Route column that concatenate startingAirport with destinationAirport columns

```
ALTER TABLE FlightData2 ADD COLUMNS (FlightRoute string);


INSERT OVERWRITE TABLE FlightData2
SELECT FlightID, SearchDate, FlightDate, StartingAirport,
DestinationAirport, FareBasisCode string, TravelDuration,
ElapsedDays, IsBasicEconomy, IsRefundable,
IsNonStop, BaseFare, TotalFare, SeatsRemaining,
TotalTravelDistance, SegmentsAirlineName , SegmentsEquipmentDescription, FlightMonth,
```

```
CONCAT(StartingAirport, '-', DestinationAirport) AS FlightRoute
FROM FlightData2;


Select StartingAirport, DestinationAirport, FlightRoute From FlightData2 limit 10;
```



startingairport	destinationairport	flightroute
DTW	LGA	DTW-LGA
DTW	MIA	DTW-MIA
LGA	LAX	LGA-LAX
SFO	BOS	SFO-BOS
ATL	EWR	ATL-EWR
OAK	JFK	OAK-JFK
PHL	LGA	PHL-LGA
LAX	ORD	LAX-ORD
DTW	LAX	DTW-LAX
LGA	IAD	LGA-IAD

10 rows selected (0.426 seconds)

Write analysis queries, export result to HDFS, secure copy text files to local computer, and create visualizations

1. Top 10 most popular flight routes

```
INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/FlightRoute/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT FlightRoute, COUNT(FlightID) AS FlightCount
FROM FlightData2
GROUP BY FlightRoute
ORDER BY FlightCount DESC
3.LIMIT 10;
```

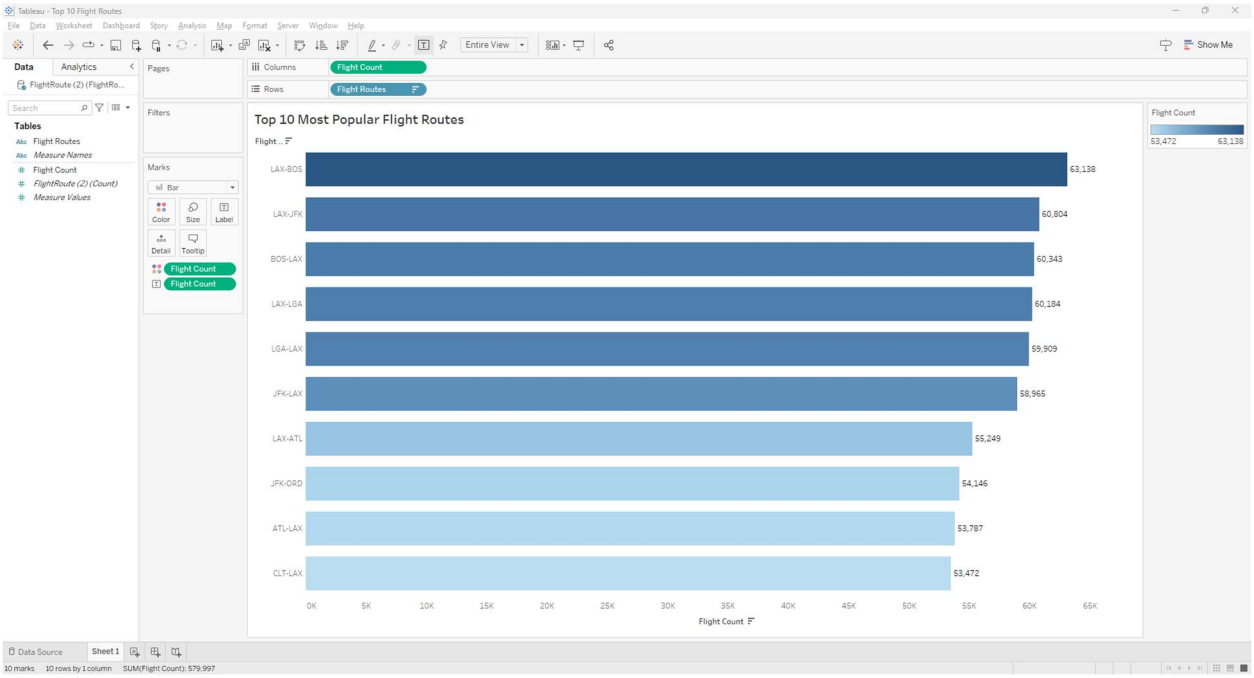
flightroute	flightcount
LAX-BOS	63138
LAX-JFK	60804
BOS-LAX	60343
LAX-LGA	60184
LGA-LAX	59909
JFK-LAX	58965
LAX-ATL	55249
JFK-ORD	54146
ATL-LAX	53787
CLT-LAX	53472

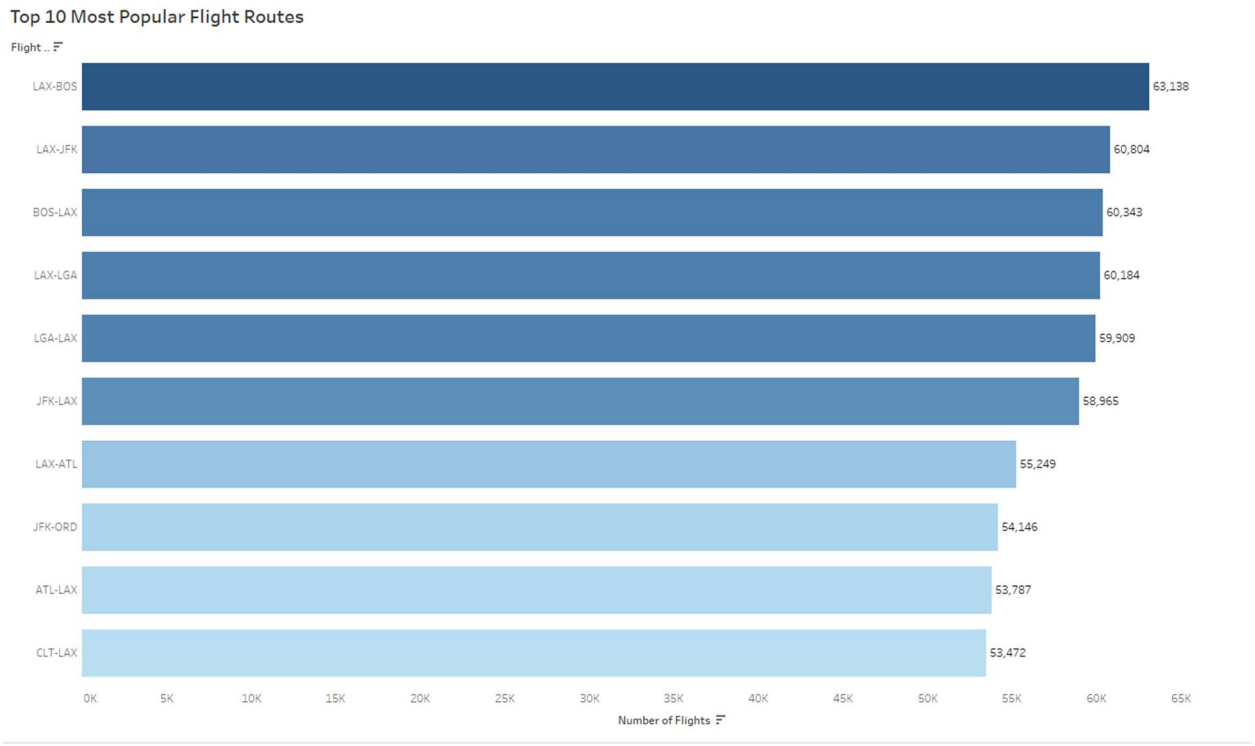
10 rows selected (10.983 seconds)

hdfs dfs -get /user/amach3/FlightData2/FlightRoute/000000_0 FlightRoute.txt

scp amach3@144.24.53.159:/home/amach3/FlightRoute.txt FlightRoute.txt

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/FlightRoute.txt FlightRoute.txt
amach3@144.24.53.159's password:
FlightRoute.txt                               100%  140     3.3KB/s   00:00
```





2. Least 10 popular flight routes

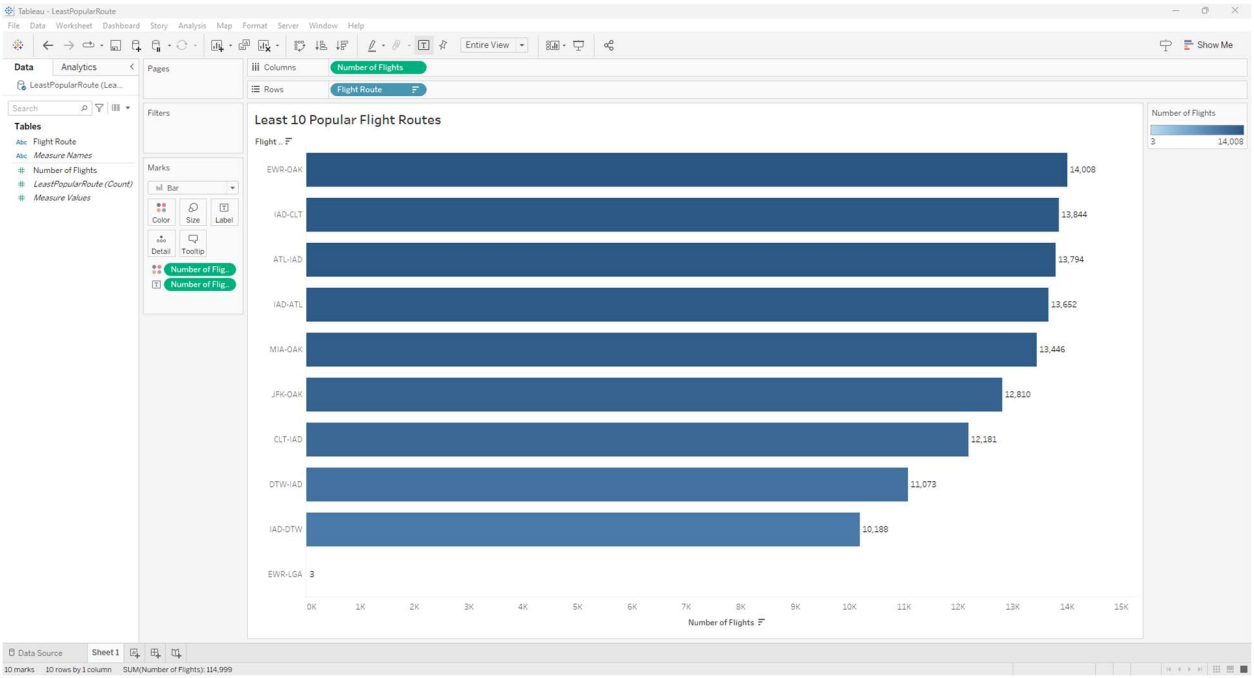
```
INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/LeastPopularRoute/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT FlightRoute, COUNT(FlightID) AS FlightCount
FROM FlightData2
GROUP BY FlightRoute
ORDER BY FlightCount ASC
LIMIT 10;
```

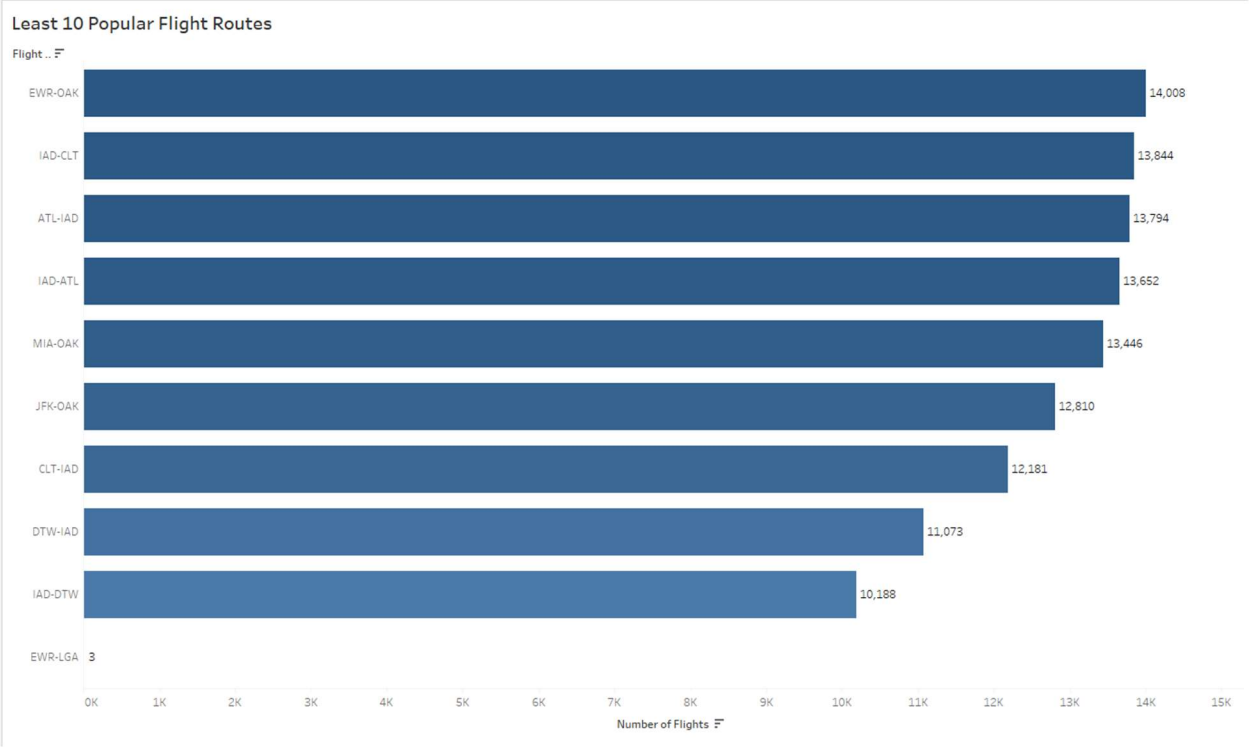
flightroute	flightcount
EWR-LGA	3
IAD-DTW	10188
DTW-IAD	11073
CLT-IAD	12181
JFK-OAK	12810
MIA-OAK	13446
IAD-ATL	13652
ATL-IAD	13794
IAD-CLT	13844
EWR-OAK	14008

hdfs dfs -get /user/amach3/FlightData2/LeastPopularRoute/000000_0 LeastPopularRoute.txt

scp amach3@144.24.53.159:/home/amach3/LeastPopularRoute.txt LeastPopularRoute.txt

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/LeastPopularRoute.txt LeastPopularRoute.txt
amach3@144.24.53.159's password:
LeastPopularRoute.txt                                    100%  136     2.7KB/s   00:00
```





3. Top 15 most expensive routes

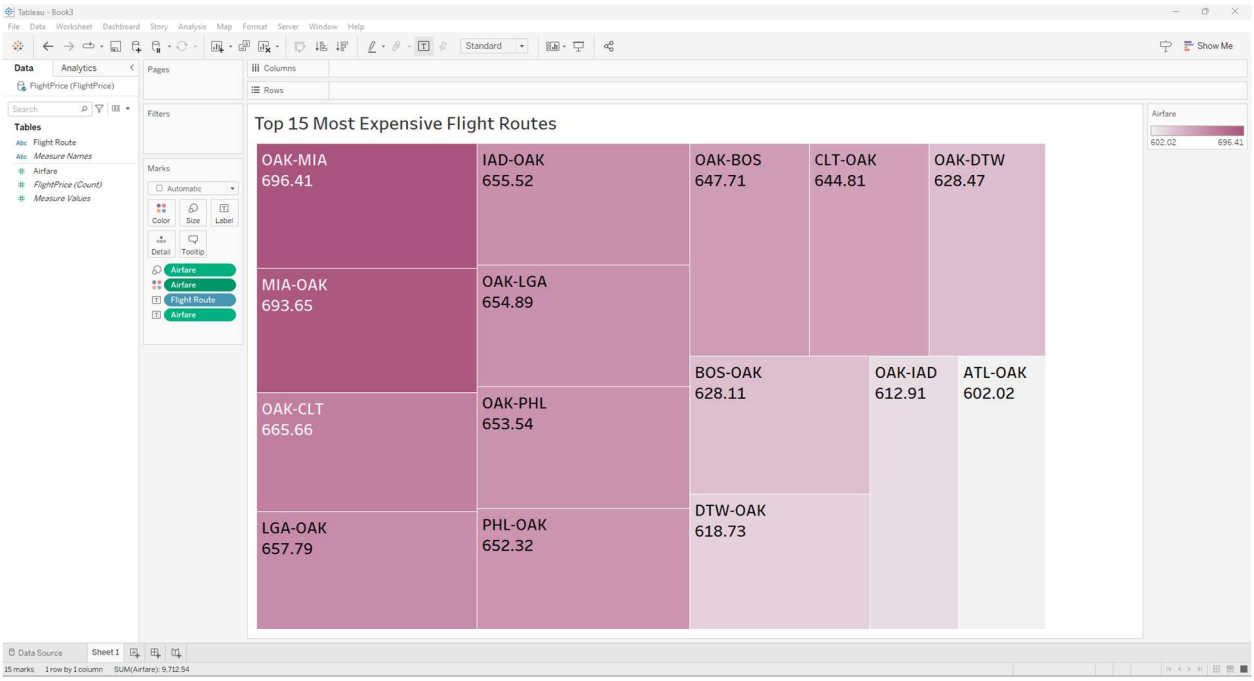
```
INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/FlightPrice/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT FlightRoute, ROUND(AVG(TotalFare),2) AS FlightPrice
FROM FlightData2
GROUP BY FlightRoute
ORDER BY FlightPrice DESC LIMIT 15;
```

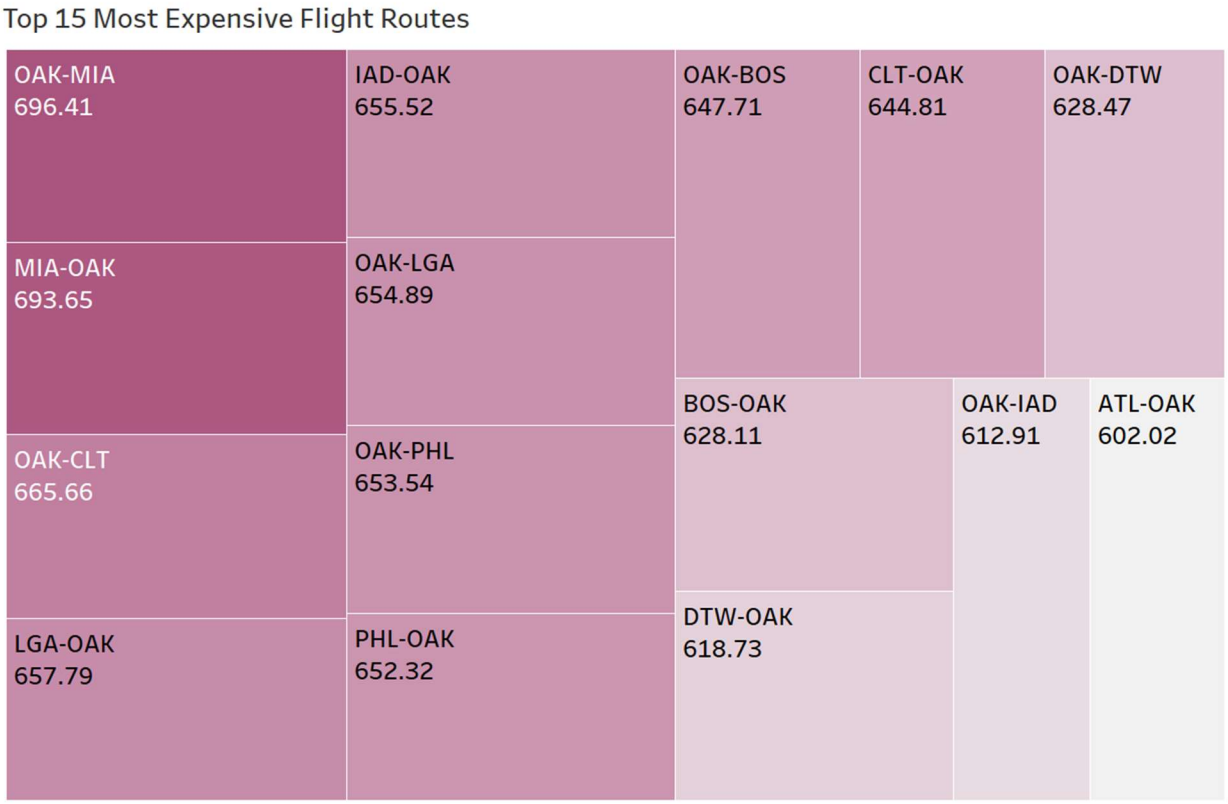
flightroute	flightprice
OAK-MIA	696.41
MIA-OAK	693.65
OAK-CLT	665.66
LGA-OAK	657.79
IAD-OAK	655.52
OAK-LGA	654.89
OAK-PHL	653.54
PHL-OAK	652.32
OAK-BOS	647.71
CLT-OAK	644.81
OAK-DTW	628.47
BOS-OAK	628.11
DTW-OAK	618.73
OAK-IAD	612.91
ATL-OAK	602.02

hdfs dfs -get /user/amach3/FlightData2/FlightPrice/000000_0 FlightPrice.txt

scp amach3@144.24.53.159:/home/amach3/FlightPrice.txt FlightPrice.txt

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/FlightPrice.txt FlightPrice.txt
amach3@144.24.53.159's password:
FlightPrice.txt                               100%   75     1.8KB/s   00:00
```





3. Airfare and Travel Distance over time

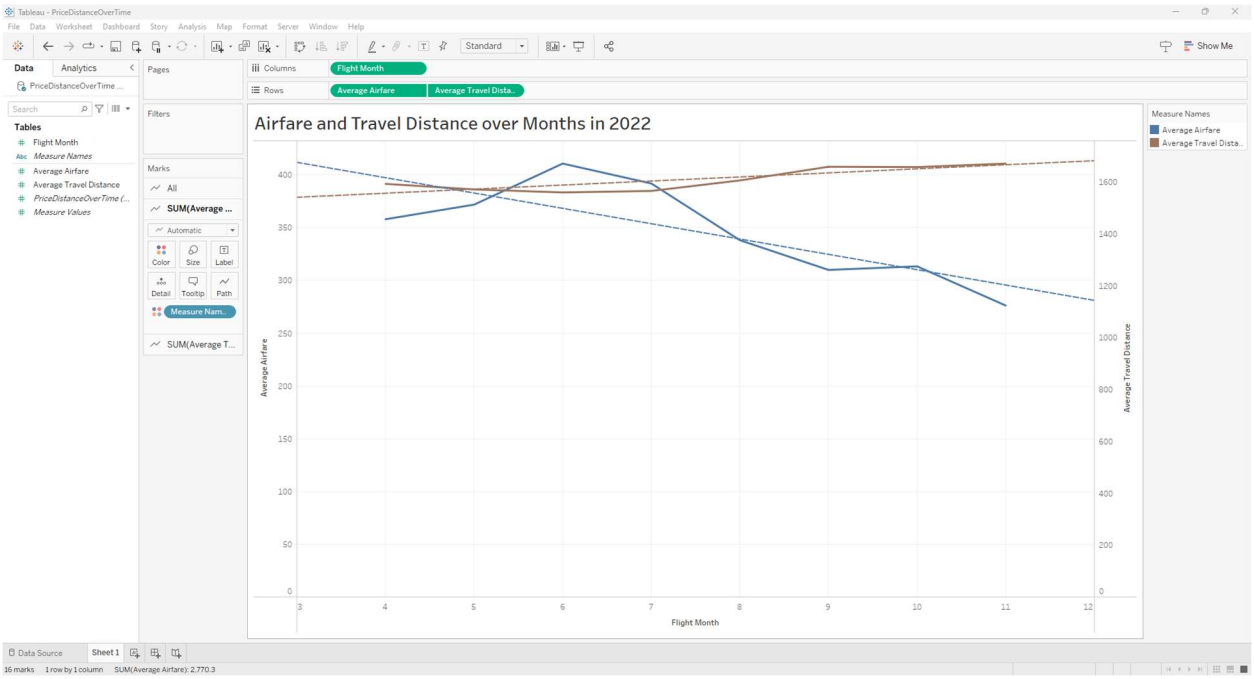
```
INSERT OVERWRITE DIRECTORY
'/user/amach3/FlightData2/PriceDistanceOverTime/'
```

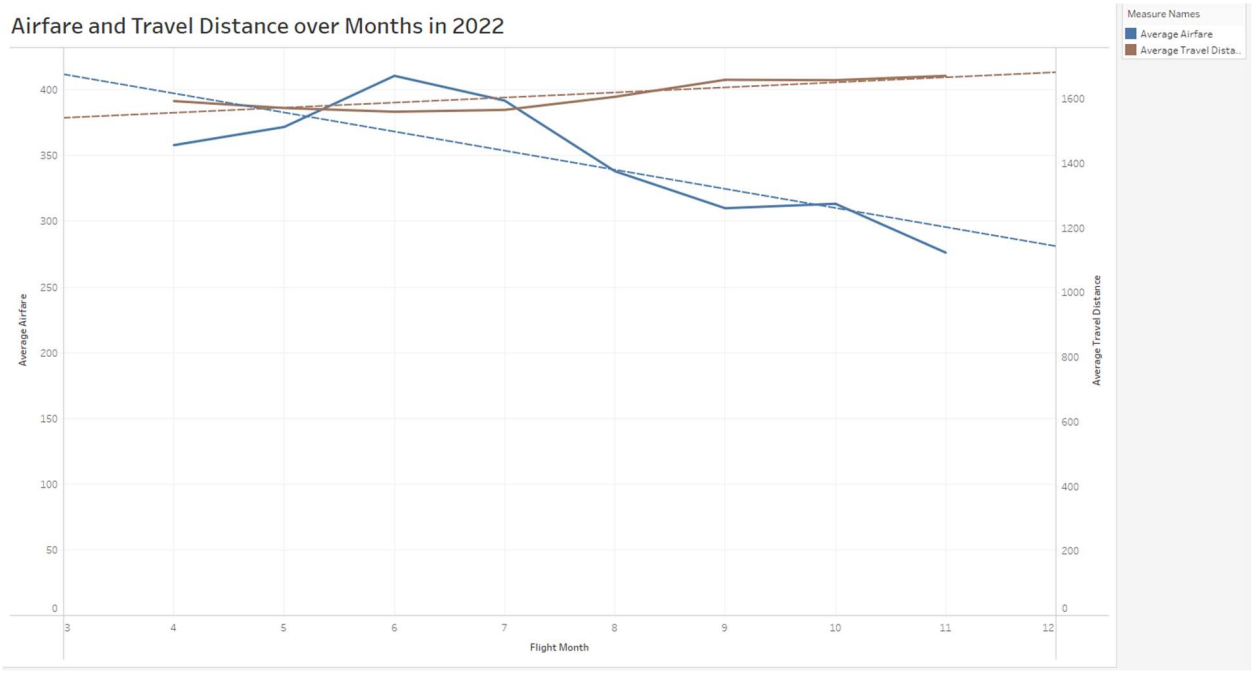
```
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT FlightMonth, ROUND(AVG(TotalFare),2) AS AverageAirfare,
ROUND(AVG(TotalTravelDistance),2) AS AverageTravelDistance
FROM FlightData2
GROUP BY FlightMonth;
```

```
+--------------+-----------------+------------------------+
| flightmonth  | averageairfare  | averagetraveldistance  |
+--------------+-----------------+------------------------+
| 8            | 338.14          | 1606.34                |
| 4            | 358.01          | 1593.09                |
| 6            | 410.76          | 1560.08                |
| 7            | 391.91          | 1565.84                |
| 10           | 313.39          | 1657.82                |
| 11           | 276.23          | 1671.29                |
| 5            | 371.85          | 1571.6                 |
| 9            | 309.98          | 1659.03                |
+--------------+-----------------+------------------------+
8 rows selected (13.431 seconds)
```

```
hdfs dfs -get /user/amach3/FlightData2/PriceDistanceOverTime/000000_0
PriceDistanceOverTime.txt


scp amach3@144.24.53.159:/home/amach3/PriceDistanceOverTime.txt
PriceDistanceOverTime.txt
```

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/PriceDistanceOverTime.txt PriceDistanceOve
rTime.txt
amach3@144.24.53.159's password:
PriceDistanceOverTime.txt                     100%   17     0.3KB/s   00:00
```





4. Which are the most popular destinations?

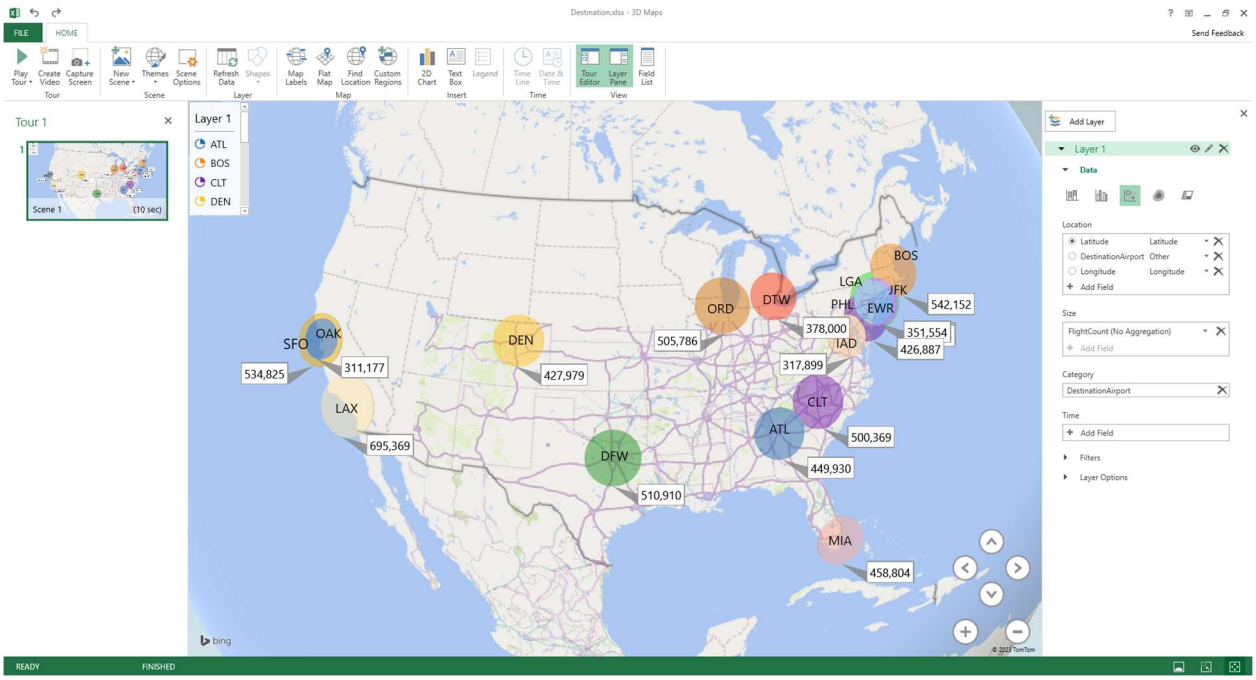
```
INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/Destination/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT DestinationAirport, Latitude, Longitude, COUNT(FlightID) AS FlightCount
FROM FlightData2 F
JOIN Airport A
ON F.DestinationAirport = A.IATA
```

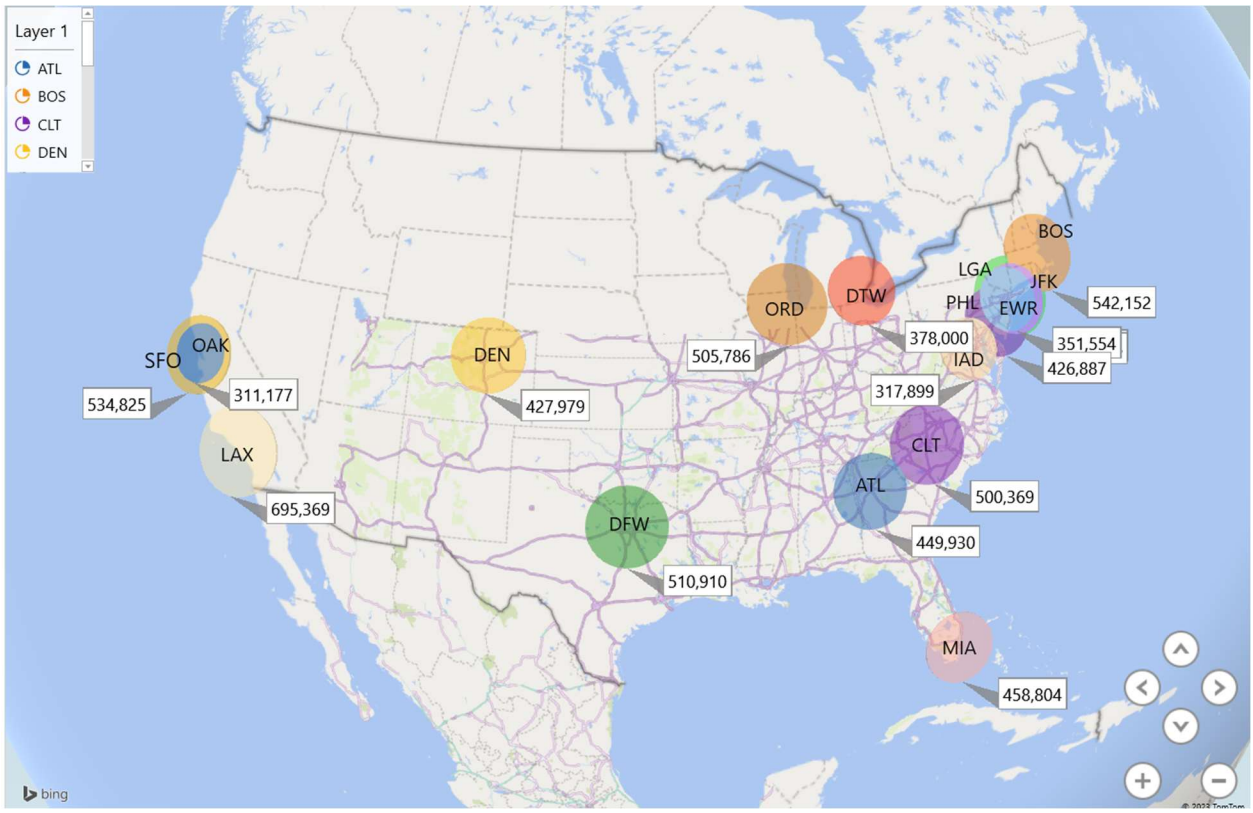
```
GROUP BY DestinationAirport, Latitude, Longitude;
```

```
+---------------------+-------------+---------------+-------------+
| destinationairport  |   latitude  |   longitude   | flightcount |
+---------------------+-------------+---------------+-------------+
| DEN                 | 39.86169815 | -104.6729965  | 427980      |
| DFW                 | 32.896801   | -97.038002    | 510912      |
| EWR                 | 40.69250107 | -74.16870117  | 351559      |
| CLT                 | 35.2140007  | -80.94309998  | 500372      |
| BOS                 | 42.36429977 | -71.00520325  | 542153      |
| DTW                 | 42.21239853 | -83.35340118  | 378003      |
| LGA                 | 40.77719879 | -73.87259674  | 569748      |
| JFK                 | 40.63980103 | -73.77890015  | 425023      |
| ORD                 | 41.9786     | -87.9048      | 505789      |
| MIA                 | 25.79319954 | -80.29060364  | 458808      |
| IAD                 | 38.94449997 | -77.45580292  | 317899      |
| LAX                 | 33.94250107 | -118.4079971  | 695371      |
| OAK                 | 37.721298   | -122.221001   | 311179      |
| PHL                 | 39.87189865 | -75.2410965   | 426892      |
| SFO                 | 37.61899948 | -122.375      | 534826      |
| ATL                 | 33.6367     | -84.428101    | 449931      |
+---------------------+-------------+---------------+-------------+
16 rows selected (28.689 seconds)
```

```
hdfs dfs -get /user/amach3/FlightData2/Destination/000000_0 Destination.txt


scp amach3@144.24.53.159:/home/amach3/Destination.txt Destination.txt
```

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/Destination.txt Destination.txt
amach3@144.24.53.159's password:
Destination.txt                               100%   36     0.9KB/s   00:00
```





5. Distribution of basic economy tickets

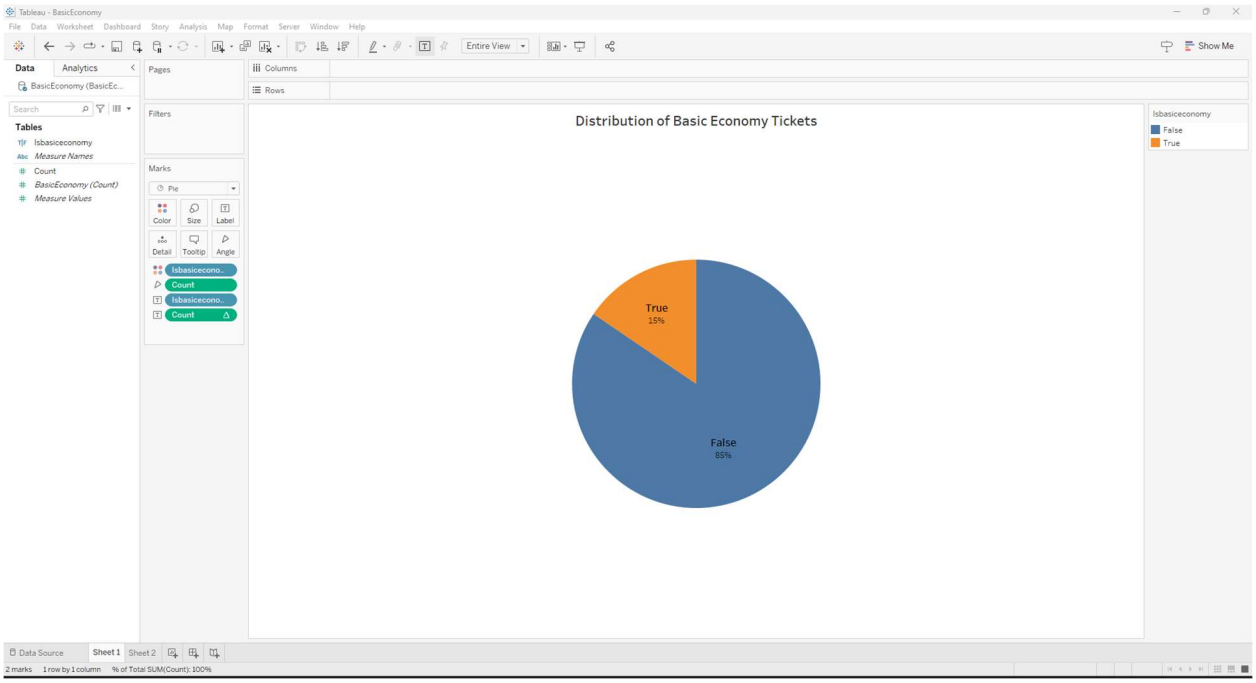
```
INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/BasicEconomy/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
```

```
SELECT isBasicEconomy, COUNT(*) as count
FROM FlightData2
GROUP BY isBasicEconomy;
```

```
+-----------------+---------+
| isbasiceconomy  |  count  |
+-----------------+---------+
| true            | 1146599 |
| false           | 6259805 |
+-----------------+---------+
2 rows selected (12.57 seconds)
```

```
hdfs dfs -get /user/amach3/FlightData2/BasicEconomy/000000_0 BasicEconomy.txt


scp amach3@144.24.53.159:/home/amach3/BasicEconomy.txt BasicEconomy.txt
```

```
jiazh@LAPTOP-2FFALVQG MINGW64 ~
$ scp amach3@144.24.53.159:/home/amach3/BasicEconomy.txt BasicEconomy.txt
amach3@144.24.53.159's password:
BasicEconomy.txt                              100%   13     0.3KB/s   00:00
```



Distribution of Basic Economy Tickets