**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

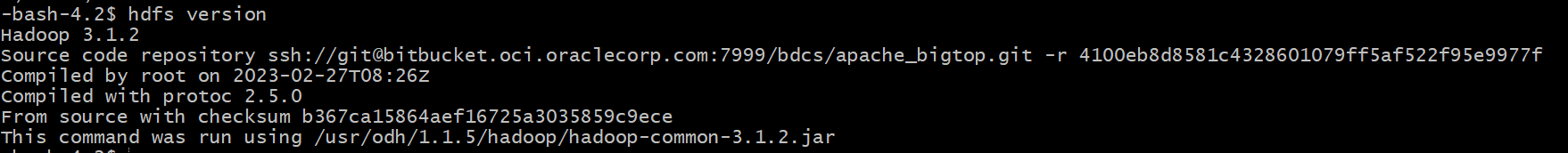
1. **Objectives**

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

1. **Platform Specifications**

hdfs version



lscpu

Text

Description automatically generated

hdfs dfsadmin -report

**Text

Description automatically generated**

* 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

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

**Note: We realized the steps in this section are not necessary in this project. We could have downloaded the zip file directly from Kaggle then unzip the file on a local computer. You can move to section 4.**

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.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

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')

Graphical user interface, text, website

Description automatically generated

• Now upload the kaggle.json file

from google.colab import files

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

Text

Description automatically generated

• make sure kaggle.json file is present

!ls -lha kaggle.json

Graphical user interface, website

Description automatically generated

• Install kaggle API client

!pip install -q kaggle

Text

Description automatically generated with medium confidence

• kaggle API client expects the file to be in ~/.kaggle

• so move it there

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/

Text

Description automatically generated

• we need to set permissions

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

Logo

Description automatically generated

• check your directory before downloading the datasets

!pwd

Graphical user interface, application

Description automatically generated

• list all available datasets

!kaggle datasets list

Text

Description automatically generated

• 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>)

Graphical user interface, text, application, chat or text message

Description automatically generated

!kaggle datasets download -d dilwong/flightprices

Graphical user interface, text, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated

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

Text

Description automatically generated with medium confidence

* Copy the zip file to Google Drive

!cp flightprices.zip /content/gdrive/MyDrive

Text

Description automatically generated with medium confidence

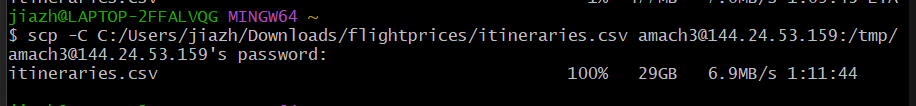
Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Secure copy csv files from my local computer to Linux server**

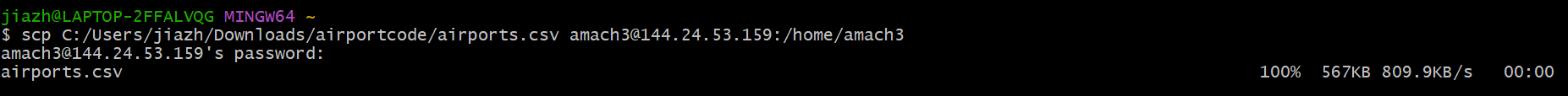
* Download the flightprices.zip file from Google Drive then unzip it in Download folder, then secure copy it to tmp folder on Linux server.

scp -C C:/Users/jiazh/Downloads/flightprices/itineraries.csv [amach3@144.24.53.159:/tmp](mailto:amach3@144.24.53.159:/tmp)



* 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



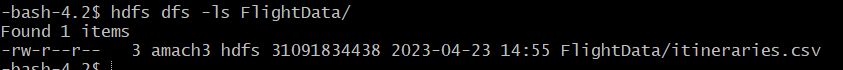
1. **Copy files from linux server to hdfs**

* 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/



* 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/

Text

Description automatically generated

* make the Airport directory public

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

1. **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;

A screenshot of a computer

Description automatically generated with medium confidence

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,

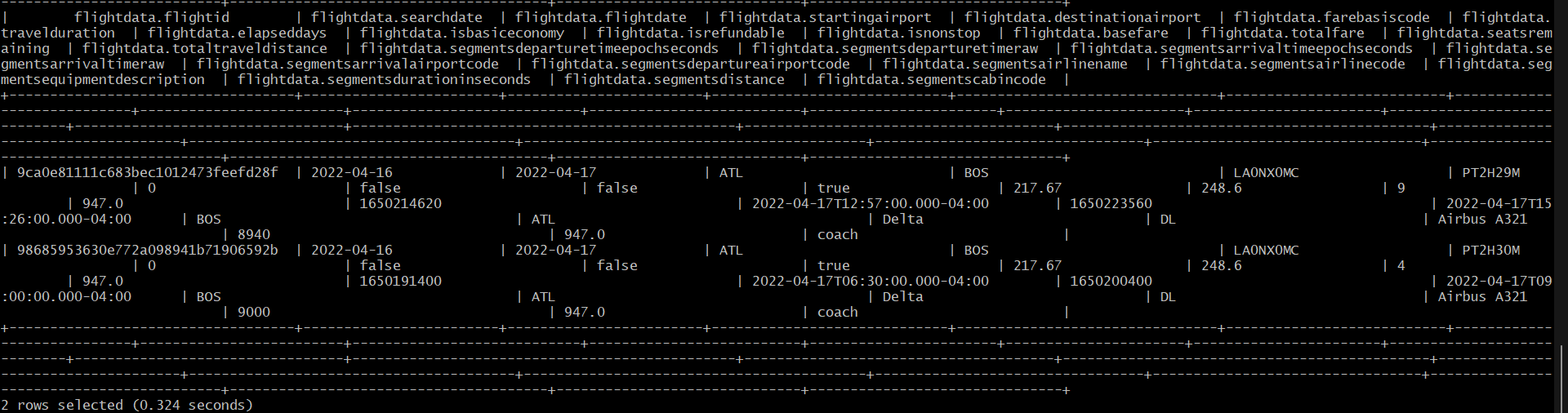
SegmentsCabinCode string)

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;

Graphical user interface, application

Description automatically generated

1. **Data Engineering**
2. 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;

Graphical user interface

Description automatically generated with medium confidence

1. 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;

A picture containing text

Description automatically generated

1. Check null values

SELECT COUNT(\*) FROM FlightData2 where TotalTravelDistance IS NULL;

A picture containing text

Description automatically generated

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

A picture containing text

Description automatically generated

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

A picture containing graphical user interface

Description automatically generated

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

A picture containing text

Description automatically generated

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

A picture containing text

Description automatically generated

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

A picture containing text

Description automatically generated

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

A picture containing graphical user interface

Description automatically generated

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

A picture containing text

Description automatically generated

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

A picture containing graphical user interface

Description automatically generated

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

Text

Description automatically generated

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

Text

Description automatically generated

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

A picture containing text

Description automatically generated

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;

A picture containing graphical user interface

Description automatically generated

1. 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;

A picture containing shape

Description automatically generated

1. Add a FlightRoute 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;

A picture containing diagram

Description automatically generated

1. **Write analysis queries, export result to HDFS, secure copy text files to local computer, and create visualizations.**
2. 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

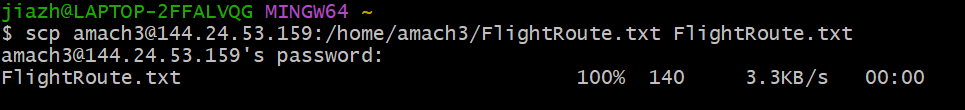
LIMIT 10;

A picture containing diagram

Description automatically generated

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

scp [amach3@144.24.53.159:/home/amach3/FlightRoute.txt FlightRoute.txt](mailto:amach3@144.24.53.159:/home/amach3/FlightRoute.txt%20FlightRoute.txt)



Graphical user interface, application

Description automatically generated

A picture containing bar chart

Description automatically generated

1. 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

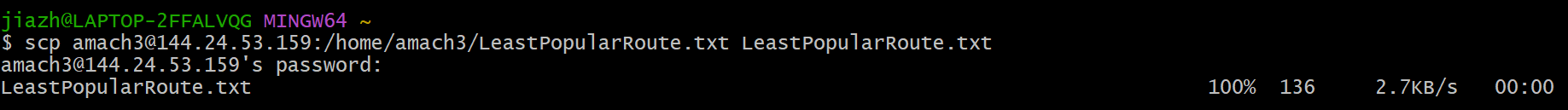
LIMIT 10;

Text

Description automatically generated

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

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



Chart

Description automatically generated

Chart, bar chart

Description automatically generated

1. 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

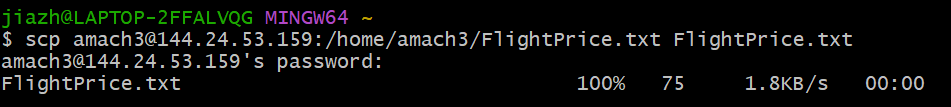
ORDER BY FlightPrice DESC LIMIT 15;

A picture containing diagram

Description automatically generated

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

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



Chart, treemap chart

Description automatically generated

Chart, treemap chart

Description automatically generated

1. 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;

A picture containing text

Description automatically generated

hdfs dfs -ls /user/amach3/FlightData2/PriceDistanceOverTime

hdfs dfs -get /user/amach3/FlightData2/PriceDistanceOverTime/000000\_0

hdfs dfs -get /user/amach3/FlightData2/PriceDistanceOverTime/000001\_0

hdfs dfs -get /user/amach3/FlightData2/PriceDistanceOverTime/000002\_0

hdfs dfs -get /user/amach3/FlightData2/PriceDistanceOverTime/000003\_0

cat 000000\_0 000002\_0 000003\_0 000001\_0 > PriceDistanceOverTime.txt

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

Text

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

1. Which are the most popular destinations over the months?

INSERT OVERWRITE DIRECTORY '/user/amach3/FlightData2/Destination/'

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

SELECT DestinationAirport, date\_format(FlightDate,'MM-01-yyyy') AS formatted\_date,Latitude, Longitude, COUNT(FlightID) AS FlightCount

FROM FlightData2 F

JOIN Airport A

ON F.DestinationAirport = A.IATA

GROUP BY DestinationAirport,date\_format(FlightDate,'MM-01-yyyy'), Latitude, Longitude;

A screen shot of a computer screen

Description automatically generated with low confidence

A screen shot of a computer screen

Description automatically generated with low confidence

hdfs dfs -get /user/amach3/FlightData2/Destination/000000\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000001\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000002\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000003\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000004\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000005\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000006\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000007\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000008\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000009\_0

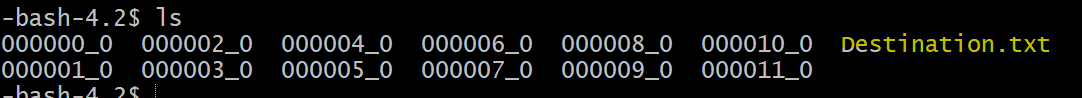
hdfs dfs -get /user/amach3/FlightData2/Destination/000010\_0

hdfs dfs -get /user/amach3/FlightData2/Destination/000011\_0

Text

Description automatically generated with medium confidence

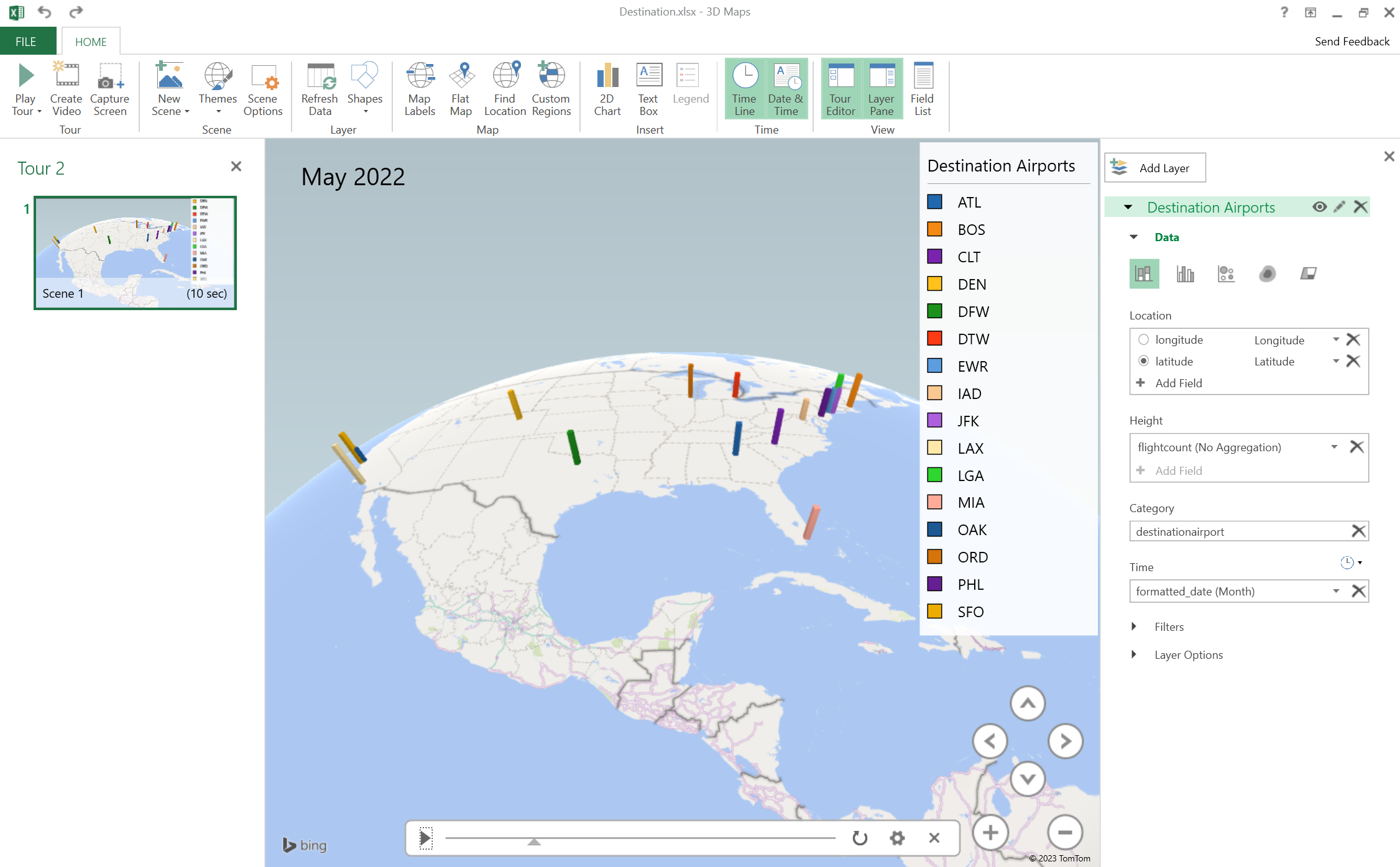
cat 000000\_0 000002\_0 000003\_0 000001\_0 000004\_0 000005\_0 000006\_0 000007\_0 000008\_0 000009\_0 000010\_0 000011\_0 > Destination.txt



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

Text

Description automatically generated with medium confidence



A screenshot of a computer screen

Description automatically generated with low confidence

1. 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;

A picture containing diagram

Description automatically generated

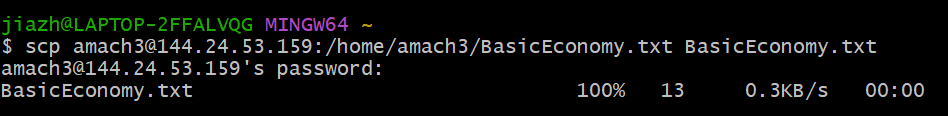
hdfs dfs -ls /user/amach3/FlightData2/BasicEconomy

hdfs dfs -get /user/amach3/FlightData2/BasicEconomy/000000\_0

hdfs dfs -get /user/amach3/FlightData2/BasicEconomy/000001\_0

cat 000000\_0 000001\_0 > BasicEconomy.txt

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



Graphical user interface, chart, application, pie chart

Description automatically generated

Chart, pie chart

Description automatically generated