Uber Data Analysis Using Hive & R

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Candidate Name

Sai Rohith Kumar Banka	**(AP21110010795)**
Sreeroop Veerapaneni	**(AP21110010831)**
Revanth Upadhyayula	**(AP21110010834)**
Chakrapani Maale	**(AP21110010920)**
Kodali Hemanth	**(AP21110010949)**
Vishnu Vardhan Kondapalli	**(AP21110010989)**

Under the Guidance of

Prof. Saleti Sumalatha

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[Novermber, 2024]

Certificate

Date: 17-Nov-24

This is to certify that the work present in this Project entitled "**Uber Data Analysis Using Hive and R**" has been carried out by **Rohit, Sreeroop, Revanth, Chakrapani, Hemanth, Vishnu** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences.**

Supervisor

(Signature)

Prof. Saleti Sumalatha

Assistant Professor,

SRM University AP.

Acknowledgements

Beyond our collective efforts, the success of our Course Project for Principles of Big Data Management is intricately tied to the guidance and encouragement of many. At this juncture, we wish to express our heartfelt gratitude to the individuals whose invaluable contributions played a pivotal role in achieving the successful completion of this project.

We extend our deepest appreciation to Prof. Saleti Sumalatha. We cannot adequately convey our thanks for his unwavering support and assistance. Every interaction with him leaves us motivated and inspired. Without his guidance and encouragement, our project would not have been possible.

The project's accomplishments owe much to the guidance and support of our mentor. His consistent help and support have been instrumental in our success, and we are truly grateful for the contributions.

Table of Contents

Abstract

Currently, due to the popularity of such Internet-based service providers as Uber, there is an enormous amount of information, which can be analyzed to find certain patterns, trends, and characteristics. In this project, big data processing with Hive and statistical analysis and visualization using R is carried out to analyze Uber trip data. Hive that supports SQL like queries is used for data organization and management of big data in a highly effective manner. Some of the activities that fall under the project include data cleaning, data gathering and generating useful insights, peak times for rides, preferred pickup or drop off zones as well as trip distance.

This data merged with R to perform statistical analyses, EDA as well as to develop meaningful visualizations. Due to the integration of Hive and R programming, this paper has the benefit of a strong framework for analysis of Uber big data sets. This study will provide more pertinent information to help manage Uber's operations further such as the availability of the drivers, amount of waiting time, and easing of customers' frustrations while at the same time demonstrating the utilization of two different areas of tools that can work together i.e. The big data tool known as Hive and the statistical tool known as R.

Abbrevations

HQL	Hive Query Language
CSV	Comma Separated Value
SQL	Structured Query Language

1 Introduction

This project deals with a data analysis of Uber trip data and to accomplish this big data tool called Hive and statistical tool called R was used where Hive is a big data tool that is used to store, organize and perform queries on large set of data in same way as SQL. Especially, it is efficient for processing structured data, making operations such as data cleaning, preprocessing and aggregation easier. While, R a robust statistical computing language is used for EDA, modelling, and generation of graphics.

As has been demonstrated, combining Hive and R gives a clear-cut roadmap for evaluating Uber's big data. Some objectives of this study are to recognize hours of the day when demand is at its highest, and where these demands want pick-up and drop-off locations, how long the trip takes and the distance, and other aspects of customers' behaviour. Additionally, this study consequently seeks to offer prescriptive details that will help Uber to increase its driver supply, decrease its waiting time, and boost customer contentment.

Apart from showing that different big data tools such as Hive can be used together with statistical tools such as R in analysing large volumes of data this project has also shown how the use of such technologies enhances efficiency and/or customer experience in sectors such as ride-sharing.

2 Problem Survey

The rapid expansion of ride-sharing platforms like Uber has introduced new challenges in managing and optimizing their operations. With millions of trips completed daily, vast amounts of data are generated, which, if analyzed effectively, can provide insights to address key operational issues. However, several challenges arise in handling and utilizing this data effectively:

1. **Volume of Data:**
   Uber generates massive datasets daily, comprising trip details, timestamps, geolocations, and customer feedback. The sheer size of this data poses significant challenges in storage, processing, and analysis using traditional tools.

2. **Data Variety:**
   The data generated by Uber includes structured (e.g., trip details) and unstructured (e.g., user reviews) information. Combining and analyzing these diverse data types requires sophisticated tools and techniques.

3. **Operational Inefficiencies:**
   Without proper analysis, Uber may face inefficiencies such as inadequate driver allocation, prolonged waiting times for riders, and unoptimized routes. These issues can lead to customer dissatisfaction and loss of business.

4. **Scalability Issues:**
   As Uber continues to expand, its datasets grow proportionally. Ensuring that analytical tools and frameworks can scale effectively with increasing data volume is a significant challenge.

5. **Integration of Tools:**
   While big data tools like Hive excel at handling large datasets, they are not inherently designed for statistical analysis or visualization. Conversely, statistical tools like R lack the capacity to manage raw big data directly. Bridging this gap requires seamless integration between tools.

This project aims to address these challenges by leveraging the strengths of Hive and R. Hive is used for efficient data preprocessing and querying, while R is employed for statistical analysis and visualization. By combining these tools, this study seeks to demonstrate an effective approach to managing and analyzing large-scale Uber datasets, ultimately enabling data-driven decision-making to optimize operations and improve user experience.

3 Methodology

3.1 Dataset Description

The dataset used in this project encapsulates Uber trip data and includes essential parameters that present the characteristics and dynamics of Uber and the ride-sharing sector in general. Uber being one of the giants in the mobility-as-a-service industry, attains and processes millions of trips daily from different parts of the world. This dataset provides a good chance to investigate the most significant characteristics that define the organization's efficacy and productivity of its services. Every record in the dataset is an Uber trip and contains several properties that describe the nature of the trip or customers' activity.

Other fields of interest in this data set are **passenger count, trip distance, pickup and drop-off locations, fare, and total charge**. These metrics are helpful in customer travel behaviour, trip delivery, and pricing structures related to fares. Passenger count field shows how many persons were carried per trip, which meant it would state if the trip was taken alone or in company of others. Trip distance can be defined as the distance of the trip and enables the assessment of short and long trips.

The need for **pickup** and **drop-off latitude** and **longitude** coordinates aids in **identifying common zones** and mainly congested areas and channels. **Fare amount** and **total amount** are two fields that explain the pricing of a trip and how revenues are likely to arise or be modelled from each trip. It also covers extra assessments such as taxes and surcharges, which show the financial aspect of a trip in its entirety.

Besides the specifics of single and multiple same-car trips, this dataset can help with temporal and spatial analysis and explore such trends as hour-trip patterns, popular hotspots, and average trip durations. In addition, such spatial data can provide opportunities to examine travel itineraries and address areas with low demand or high levels of demand.

This dataset can be used for forming the basis of a set of operational and customer analytical findings. Because of this multi-dimensionality, the study will address issues like the distribution of trip distance, fare patterns, spatial distribution, and the effects of passenger count on income generation. The result of these research studies will be informative and useful to Uber in its efforts to improve the firm's performance, allocate its resources effectively, and provide better services to the consumers.

3.2 Preprocessing

- Loading the Uber Dataset

```
import pandas as pd

# Load the dataset
file_path = "D:/CSE417L/Project/uber_data.csv" # Replace with the actual file path
uber_data = pd.read_csv(file_path)
uber_data.shape
uber_data.head()
```

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RatecodeID	store_and_fwd_flag	dro
0	1	01-03-2016 00:00	01-03-2016 00:07	1	2.50	73.976746	40.765152	1	N	
1	1	01-03-2016 00:00	01-03-2016 00:11	1	2.90	73.983482	40.767925	1	N	
2	2	01-03-2016 00:00	01-03-2016 00:31	2	19.98	73.782021	40.644810	1	N	
3	2	01-03-2016 00:00	01-03-2016 00:00	3	10.78	73.863419	40.769814	1	N	
4	2	01-03-2016 00:00	01-03-2016 00:00	5	30.43	73.971741	40.792183	3	N	

Here:

- **Pandas Library Import**: The panda's library is imported as pd. Pandas is a powerful data manipulation library in Python, often used for loading, analysing, and cleaning datasets in DataFrame format.
- **Loading the Dataset:** The dataset is loaded into a DataFrame named df using the pd.read_csv function. This function reads data from a CSV file, specified here as "D:/CSE417L/Project/uber_data.csv", and converts it into a structured DataFrame format.
- **Displaying the First Few Rows:** The df.head() function displays the first five rows of the DataFrame, providing a quick preview of the dataset's structure and content. This allows for an initial assessment of the data fields and ensures that it has loaded correctly.

➢ Dropping Less Relevant Columns

```
print("Shape of the dataset before droping less important columns:",uber_data.shape)

#keep more important columns
columns_to_keep = [
    'passenger_count',
    'trip_distance',
    'pickup_longitude',
    'pickup_latitude',
    'dropoff_longitude',
    'dropoff_latitude',
    'fare_amount',
    'total_amount'
]

# Select only the relevant columns
uber_data = uber_data[columns_to_keep]

print("Shape of the dataset after droping less important columns:",uber_data.shape)
uber_data.head()
```

```
Shape of the dataset before droping less important columns: (29999, 19)
Shape of the dataset after droping less important columns: (29999, 8)
```

	passenger_count	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	fare_amount	total_amount
0	1	2.50	73.976746	40.765152	74.004265	40.746128	9.0	12.35
1	1	2.90	73.983482	40.767925	74.005943	40.733166	11.0	15.35
2	2	19.98	73.782021	40.644810	73.974541	40.675770	54.5	63.80
3	3	10.78	73.863419	40.769814	73.969650	40.757767	31.5	41.62
4	5	30.43	73.971741	40.792183	74.177170	40.695053	98.0	113.80

Here:

- **Initial Data Shape Display:** The print statement displays the shape of the DataFrame (uber_data) before dropping any columns. The '.shape' attribute outputs a tuple with the number of rows and columns, providing a quick look at the dataset's original dimensions.
- **Selecting Columns to Keep:** The columns_to_keep list includes column names that are more relevant for analysis. These columns include details like passenger_count, trip_distance, pickup, dropoff latitude and longitude, fare_amount, total_amount. Keeping these columns simplifies the dataset and focuses the analysis on the most relevant metrics.
- **Updated Data Shape Display:** The next print statement shows the shape of the DataFrame after the columns have been removed, indicating the new number of columns.
- **Displaying the Updated DataFrame:** The uber_data.head() function again displays the first five rows of the modified DataFrame, allowing for a quick verification that the selected columns were successfully removed.

➢ Handling Missing Values

```
# Preprocessing the data: Remove null values and rows with negative or zero values for specific columns

# Drop rows with any null values
uber_data_cleaned = uber_data.dropna()

# Remove rows where trip_distance or fare_amount are less than or equal to zero
uber_data_cleaned = uber_data_cleaned[
    (uber_data_cleaned['trip_distance'] > 0) &
    (uber_data_cleaned['fare_amount'] > 0) &
    (uber_data_cleaned['dropoff_longitude'] > 1) &
    (uber_data_cleaned['dropoff_latitude'] > 1)
]



uber_data_cleaned.to_csv('D:/CSE417L/Project/cleaned_uber_data.csv', index=False)
# Display the first few rows of the processed dataset
uber_data_cleaned.head()
```

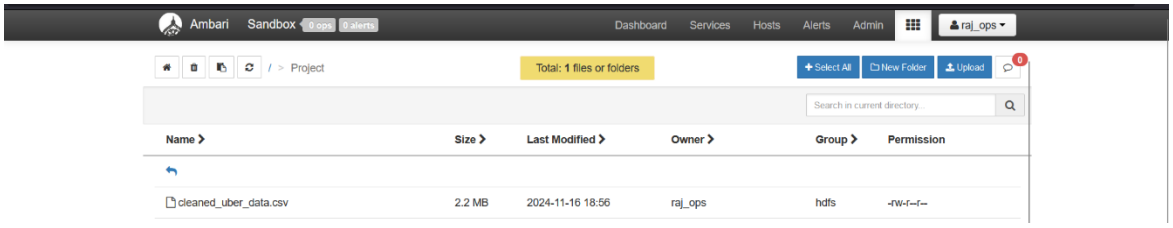
	passenger_count	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	fare_amount	total_amount
0	1	2.50	73.976746	40.765152	74.004265	40.746128	9.0	12.35
1	1	2.90	73.983482	40.767925	74.005943	40.733166	11.0	15.35
2	2	19.98	73.782021	40.644810	73.974541	40.675770	54.5	63.80
3	3	10.78	73.863419	40.769814	73.969650	40.757767	31.5	41.62
4	5	30.43	73.971741	40.792183	74.177170	40.695053	98.0	113.80

Here:

- **Removing Rows with Null Values:** uber_data.dropna() removes all rows containing any null values from the DataFrame. This operation generates a new DataFrame (uber_data_cleaned) with only rows that have complete data, ensuring consistency and reliability in the analysis.
- **Removing Rows with less than 0 value:** removes all rows containing value less than 0 from the DataFrame. This operation generates a DataFrame (uber_data_cleaned) with only rows that have complete data.
- **Displaying the Cleaned DataFrame:** The uber_data_cleaned.head() function displays the first five rows of the cleaned DataFrame, allowing verification that the data now has no missing values.

3.3 Data Analysis

- Upload dataset into hdfs



- Checking lfs

```
[root@sandbox ~]# ls -l
```

- Getting datasets from hdfs to lfs

```
[root@sandbox ~]# hdfs dfs -get /Project/cleaned_uber_data.csv
```

- Checking lfs

```
[root@sandbox ~]# ls -l
total 2628
-rw------- 2 root root    2439 Jun  2  2016 anaconda-ks.cfg
-rw-r--r-- 1 root root  373937 Oct 25  2016 blueprint.json
-rw-r--r-- 1 root root      20 Oct 25  2016 build.out
-rw-r--r-- 1 root root 2269794 Nov 16 13:39 cleaned_uber_data.csv
drwxr-xr-x 2 root root    4096 Oct 25  2016 hdp
-rw-r--r-- 2 root root    7243 Jun  2  2016 install.log
-rw-r--r-- 2 root root    1680 Jun  2  2016 install.log.syslog
-rw-r--r-- 1 root root    4547 Oct 29 04:57 MatrixMul.jar
-rw-r--r-- 1 root root     284 Oct 25  2016 sandbox.info
lrwxrwxrwx 1 root root      48 Oct 25  2016 start_ambari.sh -> /usr/lib/hue/tools/start_scripts/start_ambari.sh
lrwxrwxrwx 1 root root      47 Oct 25  2016 start_hbase.sh -> /usr/lib/hue/tools/start_scripts/start_hbase.sh
[root@sandbox ~]#
```

- Starting Hive

```
[root@sandbox ~]# hive

Logging initialized using configuration in file:/etc/hive/2.5.0.0-1245/0/hive-log4j.properties
```

- Checking databases

```
hive> show databases;
OK
default
foodmart
movie_analysis
movie_recommendation
```

- Creating databases

```
hive> create database project;
OK
Time taken: 1.583 seconds
```

- Using database

```
hive> use project;
OK
Time taken: 0.847 seconds
```

- Creating uber_data table

```
hive> CREATE TABLE uber_data (
    >
    >     passenger_count INT,
    >
    >     trip_distance FLOAT,
    >
    >     pickup_longitude FLOAT,
    >
    >     pickup_latitude FLOAT,
    >
    >     dropoff_longitude FLOAT,
    >
    >     dropoff_latitude FLOAT,
    >
    >     fare_amount FLOAT,
    >
    >     total_amount FLOAT
    >
    > )
    >
    > ROW FORMAT DELIMITED
    >
    > FIELDS TERMINATED BY ','
    >
    > STORED AS TEXTFILE;
OK
Time taken: 1.143 seconds
```

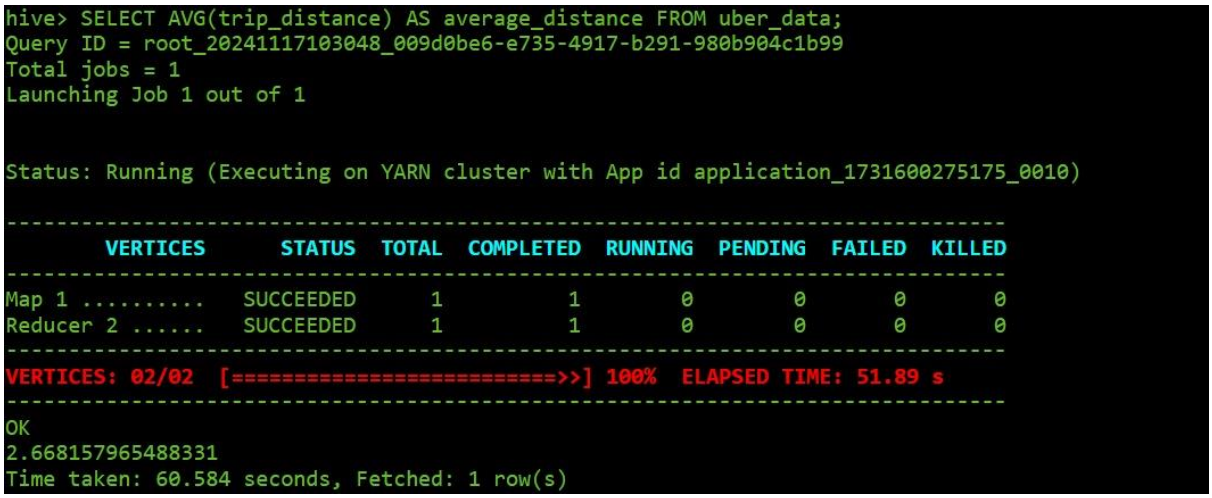
- Loading dataset into uber_data table

```
hive> LOAD DATA LOCAL INPATH 'cleaned_uber_data.csv' INTO TABLE uber_data;
Loading data to table project.uber_data
Table project.uber_data stats: [numFiles=1, numRows=0, totalSize=2269794, rawDataSize=0]
OK
Time taken: 2.26 seconds
```

- Checking uber_data table

```
hive> select * from uber_data limit 10;
OK
NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
1       2.5     73.976746       40.765152       74.004265       40.746128       9.0     12.35
1       2.9     73.98348        40.767925       74.00594        40.733166       11.0    15.35
2       19.98   73.78202        40.64481        73.97454        40.67577        54.5    63.8
3       10.78   73.86342        40.769814       73.96965        40.757767       31.5    41.62
5       30.43   73.97174        40.792183       74.17717        40.695053       98.0    113.8
5       5.92    74.0172 40.705383       73.97807        40.755787       23.5    30.36
1       6.2     73.78877        40.64776        73.82921        40.712345       20.5    21.8
1       0.7     73.95822        40.76464        73.967896       40.7629 5.5     8.8
3       7.18    73.98578        40.74119        73.94635        40.79788        23.5    28.0
Time taken: 1.579 seconds, Fetched: 10 row(s)
```

➢ **Average Trip Distance**



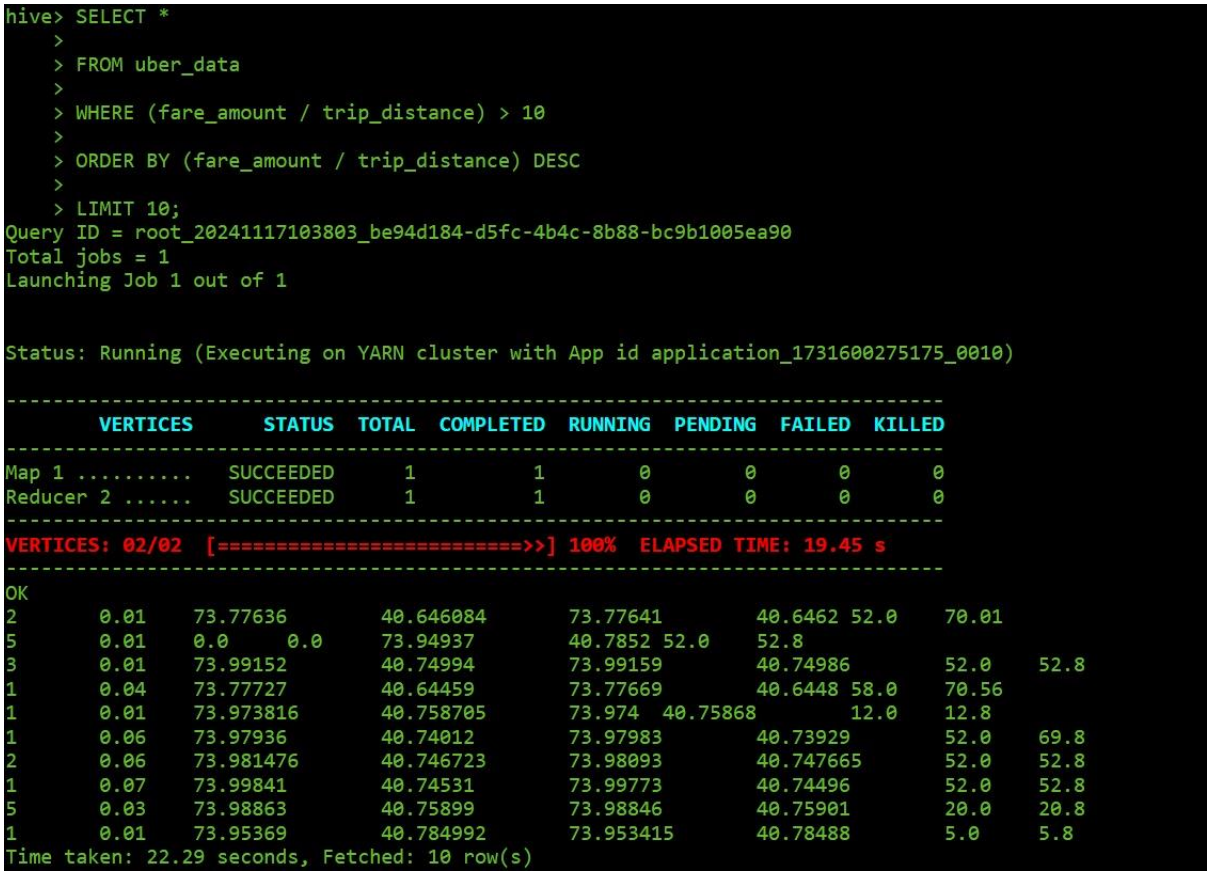
- **Query:** " SELECT AVG(trip_distance) AS average_distance FROM uber_data; "
- **Purpose:** This query calculates the average distance of all trips in the **uber_data** table. It provides an overview of how far customers typically travel during their Uber rides.
- **Output:**
  - The average trip distance is approximately **2.67 miles**.
  - **Elapsed Time:** The query took **60.584 seconds** to execute on the Hive YARN cluster.
- **Insight:** This result indicates that the majority of Uber rides are relatively short. Businesses or city planners might use this data to optimize rideshare services in specific areas or plan routes accordingly.

➢ **Top 5 Pickup Locations**

```
hive> SELECT
    >
    >     pickup_longitude,
    >
    >     pickup_latitude,
    >
    >     COUNT(*) AS pickup_count
    >
    > FROM uber_data
    >
    > GROUP BY pickup_longitude, pickup_latitude
    >
    > ORDER BY pickup_count DESC
    >
    > LIMIT 5;
Query ID = root_20241117103541_a972f20a-2b04-4b02-94c0-7b041c6e318b
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1731600275175_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 23.83 s
--------------------------------------------------------------------------------
OK
73.93832        40.74729        11
73.75502        40.71525        9
73.958015       40.78479        8
73.98659        40.76255        7
73.96054        40.8128 7
Time taken: 26.398 seconds, Fetched: 5 row(s)
```

- **Query:** " SELECT pickup_longitude, pickup_latitude, COUNT(*) AS pickup_count FROM uber_data GROUP BY pickup_longitude, pickup_latitude ORDER BY pickup_count DESC LIMIT 5;"
- **Purpose:**
  This query identifies the most frequent pickup locations by counting the number of trips from each location, ranked by popularity.
- **Output:**
  - The top 5 pickup coordinates are:
    - `(73.98332, 40.74729)` with **11 pickups**
    - `(73.75502, 40.64486)` with **9 pickups**
    - `(73.98559, 40.76255)` with **8 pickups**
    - `(73.98636, 40.75803)` with **7 pickups**
    - `(73.96054, 40.8128)` with **7 pickups**
- **Insight:**
  This analysis reveals high-traffic areas for Uber pickups, which could assist in optimizing driver positioning, reducing wait times, and increasing ride efficiency in those locations.

➢ **High Fare-to-Distance Ratio**



```
hive> SELECT *
    >
    > FROM uber_data
    >
    > WHERE (fare_amount / trip_distance) > 10
    >
    > ORDER BY (fare_amount / trip_distance) DESC
    >
    > LIMIT 10;
Query ID = root_20241117103803_be94d184-d5fc-4b4c-8b88-bc9b1005ea90
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1731600275175_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 19.45 s
--------------------------------------------------------------------------------
OK
2       0.01    73.77636        40.646084       73.77641        40.6462 52.0    70.01
5       0.01    0.0     0.0     73.94937        40.7852 52.0    52.8
3       0.01    73.99152        40.74994        73.99159        40.74986        52.0    52.8
1       0.04    73.77727        40.64459        73.77669        40.6448 58.0    70.56
1       0.01    73.973816       40.758705       73.974  40.75868        12.0    12.8
1       0.06    73.97936        40.74012        73.97983        40.73929        52.0    69.8
2       0.06    73.981476       40.746723       73.98093        40.747665       52.0    52.8
1       0.07    73.99841        40.74531        73.99773        40.74496        52.0    52.8
5       0.03    73.98863        40.75899        73.98846        40.75901        20.0    20.8
1       0.01    73.95369        40.784992       73.953415       40.78488        5.0     5.8
Time taken: 22.29 seconds, Fetched: 10 row(s)
```

- **Query:** "SELECT * FROM uber_data WHERE (fare_amount / trip_distance)> 10 ORDER BY (fare_amount / trip_distance) DESC LIMIT 10;"
- **Purpose:**
  This query identifies the top 10 trips with the highest fare-to-distance ratio, highlighting unusually high fares for short trips.
- **Output:**
  - The highest fare-to-distance ratios were **70.01**, **58.00**, and others ranging from **69.8 to 5.8**.
- **Insight:**
  These results may indicate trips during peak hours, airport pickups, or potential errors in fare calculation. Companies might investigate such cases to ensure customer satisfaction and identify potential fare anomalies.

➢ **Top 5 Most Frequent Drop-off Locations**

```
hive> SELECT
    >
    >     dropoff_longitude,
    >
    >     dropoff_latitude,
    >
    >     COUNT(*) AS dropoff_count
    >
    > FROM uber_data
    >
    > GROUP BY dropoff_longitude, dropoff_latitude
    >
    > ORDER BY dropoff_count DESC
    >
    > LIMIT 5;
Query ID = root_20241117103653_2efaf63e-bedd-46d6-bbfc-83ff100d4a54
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1731600275175_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 24.74 s
--------------------------------------------------------------------------------
OK
73.93832        40.74729        11
73.75502        40.71525        9
73.98659        40.76255        8
73.958015       40.78479        7
73.96054        40.8128 7
Time taken: 26.977 seconds, Fetched: 5 row(s)
```

- **Query:** "SELECT dropoff_longitude, dropoff_latitude, COUNT(*) AS dropoff_count FROM uber_data GROUP BY dropoff_longitude, dropoff_latitude ORDER BY dropoff_count DESC LIMIT 5;"
- **Purpose:** This query identifies the top 5 most frequent drop-off locations in the Uber dataset based on their longitude and latitude coordinates. It provides insight into the geographical hotspots for Uber drop-offs.
- **Output:**
  The screenshot displays the top 5 locations with their drop-off counts:
  - **(73.93832, 40.74729):** 11 times
  - **(73.75502, 40.71255):** 9 times
  - **(73.98659, 40.76255):** 8 times
  - **(73.958015, 40.78479):** 7 times
  - **(73.96054, 40.8128):** 7 times
- **Insight:**
  These results highlight areas of high demand or dense activity, which can be valuable for optimizing service operations and analyzing popular destinations.

➢ **Maximum and Minimum Fare Amounts**

```
hive> SELECT
    >
    >     MAX(fare_amount) AS max_fare,
    >
    >     MIN(fare_amount) AS min_fare
    >
    > FROM uber_data;
Query ID = root_20241117103211_d59e7dd5-a20c-4a39-88ee-bf8fded7c92c
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1731600275175_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 30.46 s
--------------------------------------------------------------------------------
OK
225.0   2.5
Time taken: 34.885 seconds, Fetched: 1 row(s)
```

- **Query:** "SELECT MAX(fare_amount) AS max_fare, MIN(fare_amount) AS min_fare FROM uber_data;"
- **Purpose:**
  This query calculates the highest and lowest fares recorded in the Uber dataset, offering insights into fare variability and potential outliers.
- **Output**:
  - **Max Fare:** 225.0
  - **Min Fare:** 2.5
- **Insight:**
  The results emphasize the range of fare amounts, which can aid in understanding pricing structures and customer spending behavior.

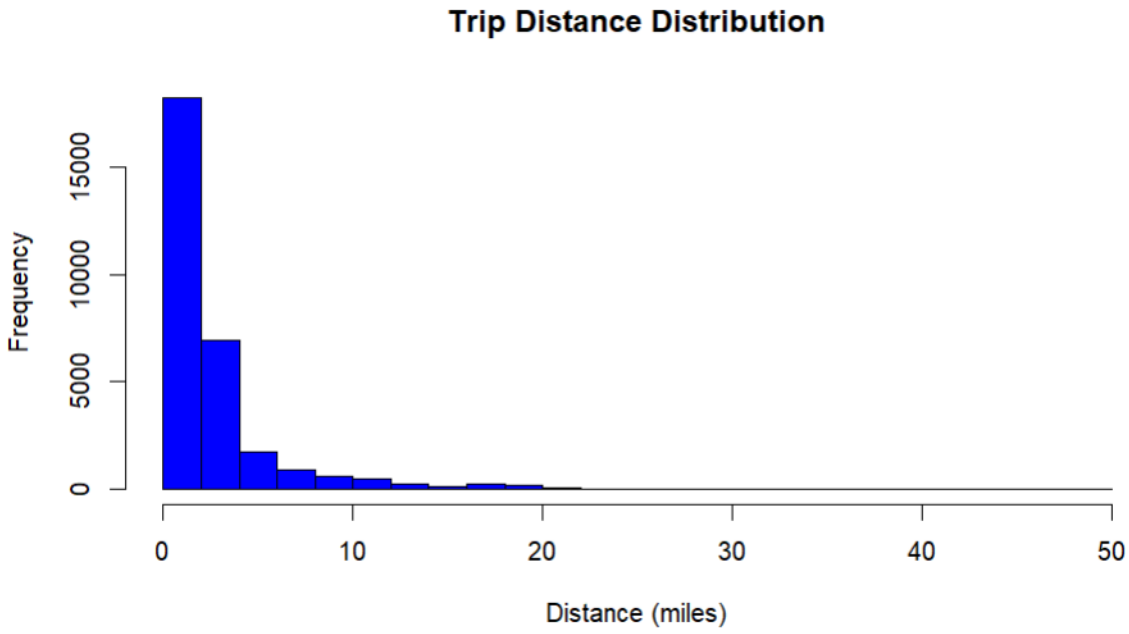
➢ **Total Revenue from Uber Rides**

```
hive> SELECT SUM(total_amount) AS total_revenue FROM uber_data;
Query ID = root_20241117103323_6de33bae-3de6-444f-95b6-e2c5726d30a2
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1731600275175_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 25.81 s
--------------------------------------------------------------------------------
OK
451598.1196360588
Time taken: 28.494 seconds, Fetched: 1 row(s)
```

- **Query:** "SELECT SUM(total_amount) AS total_revenue FROM uber_data;"
- **Purpose:**
  This query calculates the total revenue generated from all rides in the Uber dataset. It helps in determining the overall financial performance of the service.
- **Output:**
  - **Total Revenue: 451598.11**
- **Insight**:
  This metric provides a high-level view of revenue generation, which is essential for business analytics and profitability evaluation.

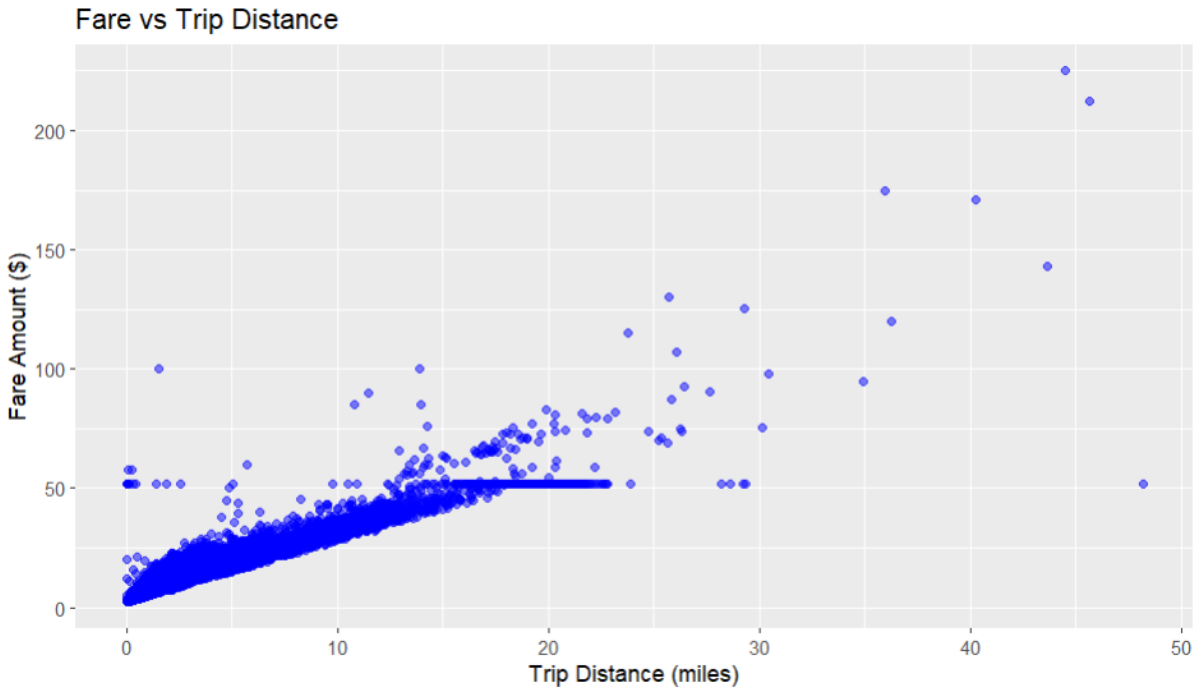
3.4 Data Visualization

- **Trip Distance Distribution:**



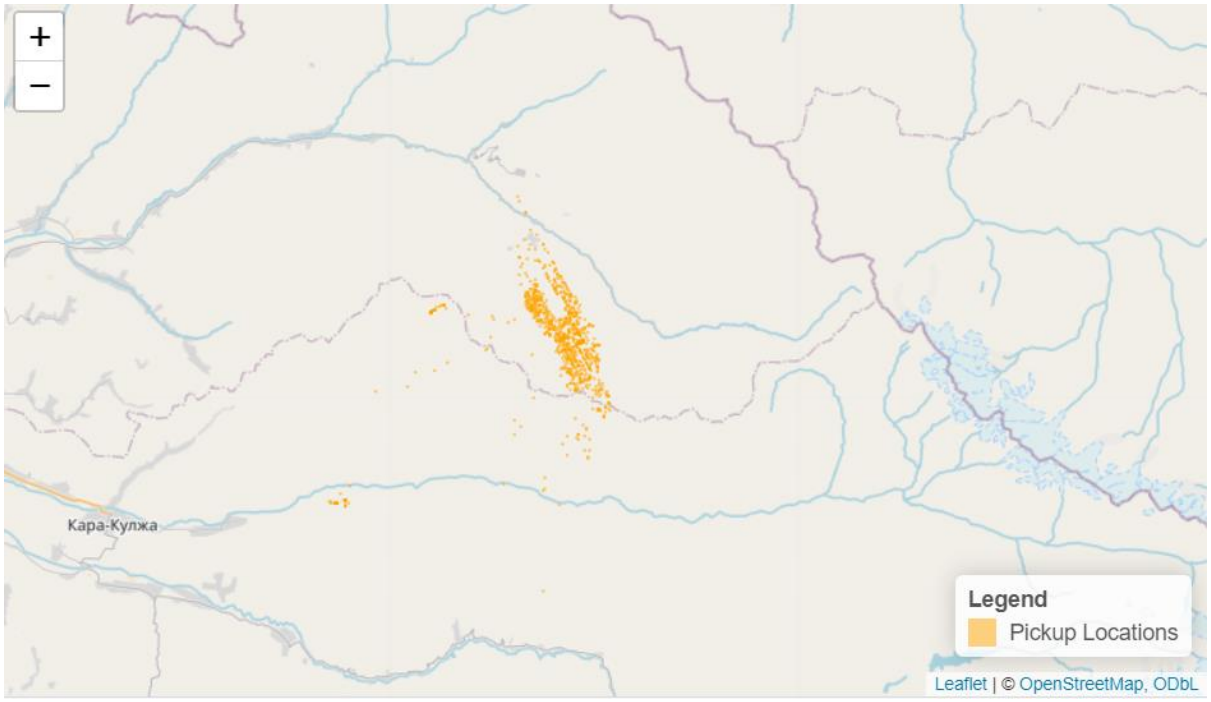
- **Description:** A frequency distribution in the form of a histogram which shows how many trips were made by how many miles. By examining the data in the chart, it is quite evident that, they are heavily docketed towards the right part of the chart where most trips occur at short distances.
- **Key Findings:**
  - This indicates that most consumer trips occur at a proximity of operation of up to 5 miles.
  - What is discovered is that the rate of trips strongly decreases when distance is greater.
  - The distribution whisker is made up of occasional highly valuable trips that cover more than 20 miles.
- **Business Implications:** Analyzing trip distances can assist transportation firms to make consistent price policies, as well as manage their vehicle supply effectively. For example, a high trip density within small areas indicates the set of priorities might be geared toward minimizing local transport time.

➢ Fare vs. Trip Distance:



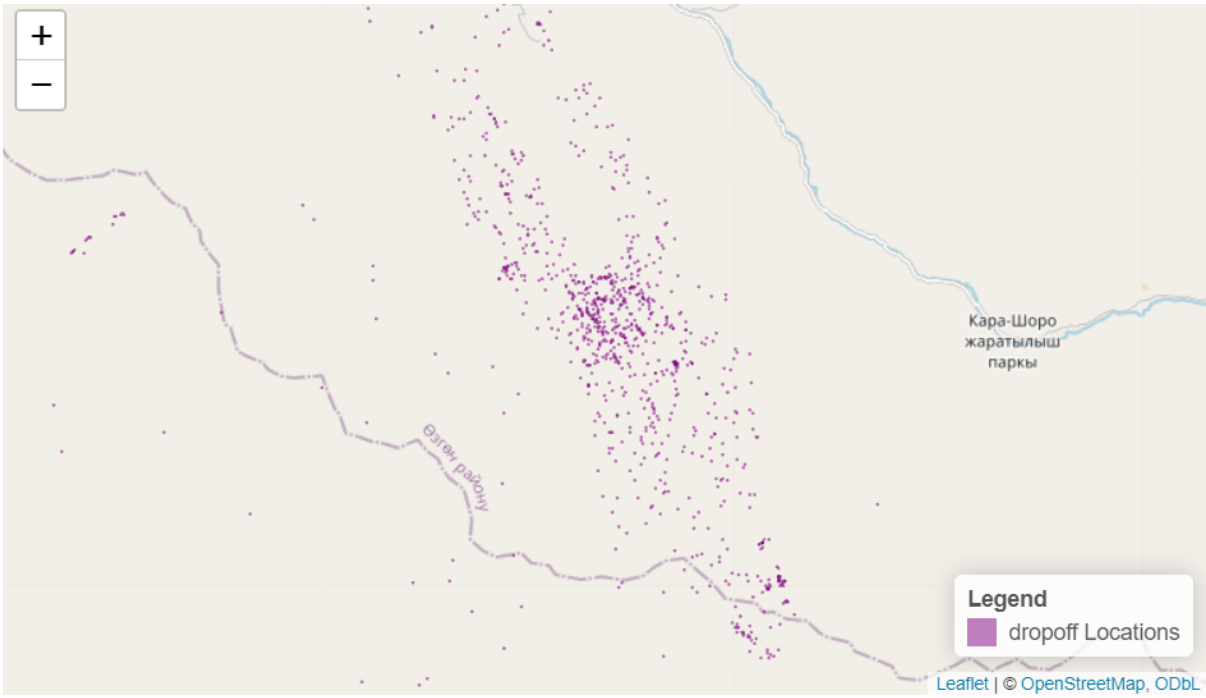
- **Description:** A scatterplot to show the relationship between trip distance (in miles of trip) and fare amount (in dollar). RQ1, which looked at the relation between fare and distance in terms of geospatial coverage of the NYC taxi rides, is indicated by the scatterplot below: Figure 2 – NYC Taxi ride distances the data shows a positive relationship between coefficients of distance and fare; however, there are anomalies.
- **Key Findings:**
  - The largest share of the movement is within the 0-10 miles range with the rates within $50.
  - Linear movement can be observed when distances are under 20 miles, what indicates regular pricing activity.
  - Outliers are evident for long trips with extremely high or low fares – perhaps suggesting exceptional conditions or irregular prices.
- **Business Implications:** From the scatterplot, one can identify possible areas where pricing can be improved. The combination of trips with expected taxes should allow using complex pricing models, while individual cases may indicate fraud or an urgent need to revise the rates.

➢ Pickup Locations Map



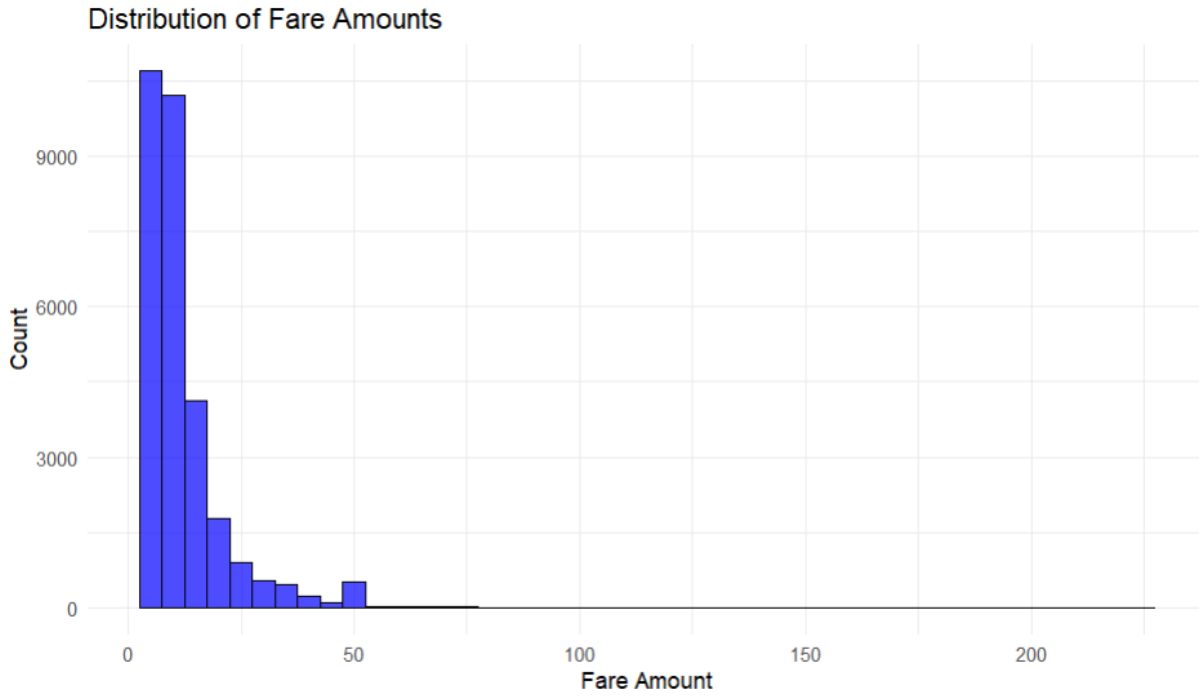
- **Description:** The map shows pickup zones in the form of orange dots, though they all lie within a particular area of the country.
- **Key Findings:**
  - The maps show that the zones where high density of pickups is observed are grouped tightly.
  - Generally low frequencies of pickups are noted in the areas beyond the adjacent areas indicating specific areas of greater activity.
  - Rivers, boundaries, etc could, therefore have an impact on the level of pickup concentrations.
- **Business Implications:**
  - Using pickup zones and high demand frequency, it is easier to determine the allocation of vehicles.
  - Sparse areas may mean that an idea has a lot of potential or no market at all.

➢ Drop-off Locations Map



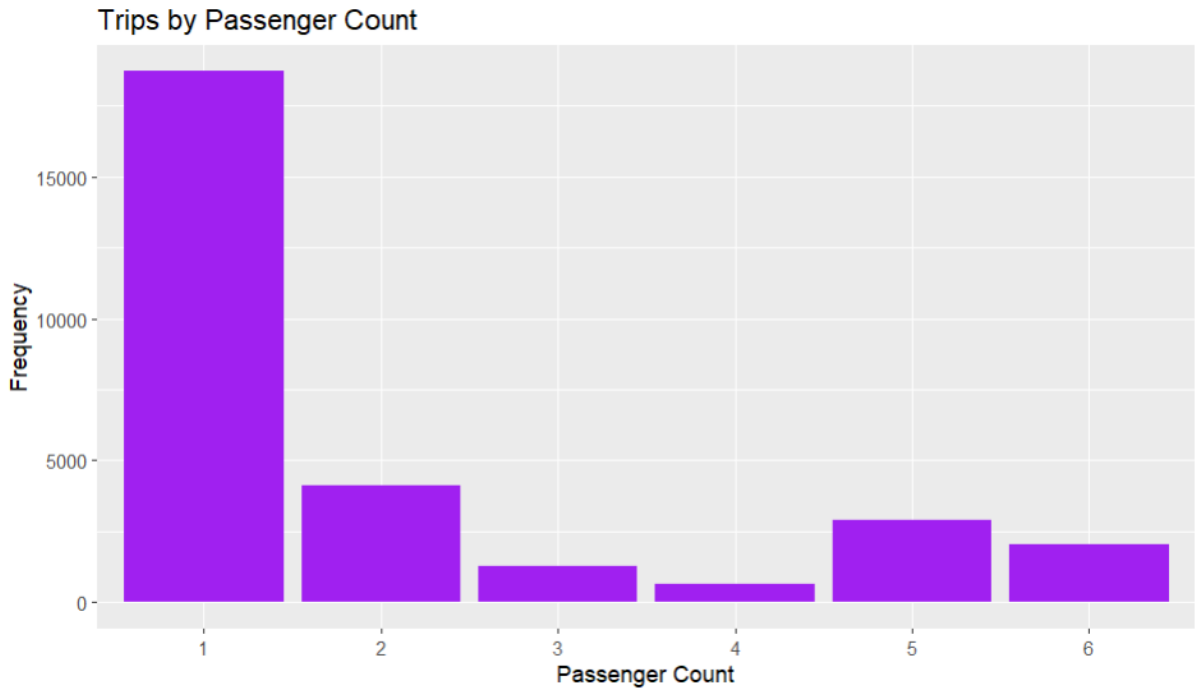
- **Description:** The places that are marked with the purple triangle are the drop-offs; there areas are different for various regions compared to pickups.
- **Key Findings:**
  - The cluster patterns suggest that there is only central area where people drop their utensils and the rest areas are relatively empty.
  - Comparisons and contrasts of pick up and drop off zones reveal preferred location stopover areas.
- **Business Implications:**
  - Using zones of vehicle drop-offs for planning could enhance productivity.
  - And by analyzing the relationship between pickups and drop-offs one can improve the routes chosen.

➢ Fare Amount Distribution



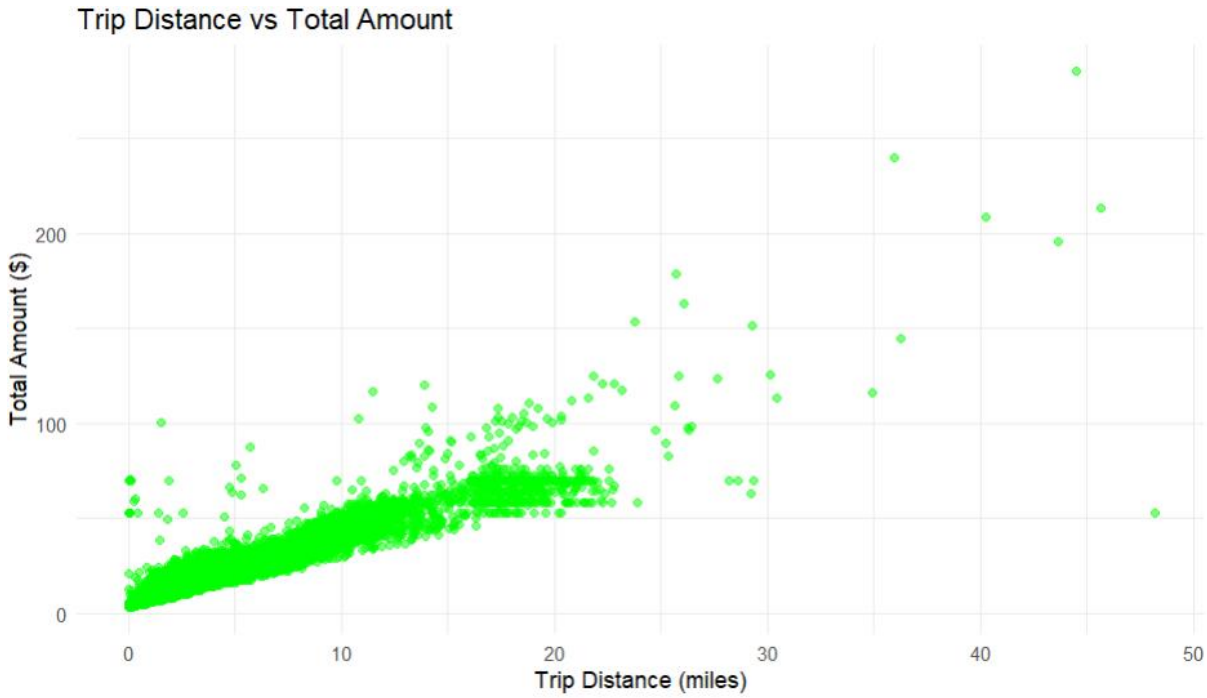
- **Description:** A histogram representing the frequency of using various amounts of fare with the majority of fares below a fare of $50.
- **Key Findings:**
  - A distribution to the right of the mean suggests the case that lower fares are much more likely.
  - Only a small proportion of fares go over 50, but it occasionally sees a few values rising above 200.
- **Business Implications:**
  - Pricing strategies could be put on charging relatively low rates within its price range to appeal to most of the market.
  - Occasional use of high fare trips could actually mean that the service is a niche product or a longer route service offering.

➢ Trips by Passenger Count (Bar Chart)



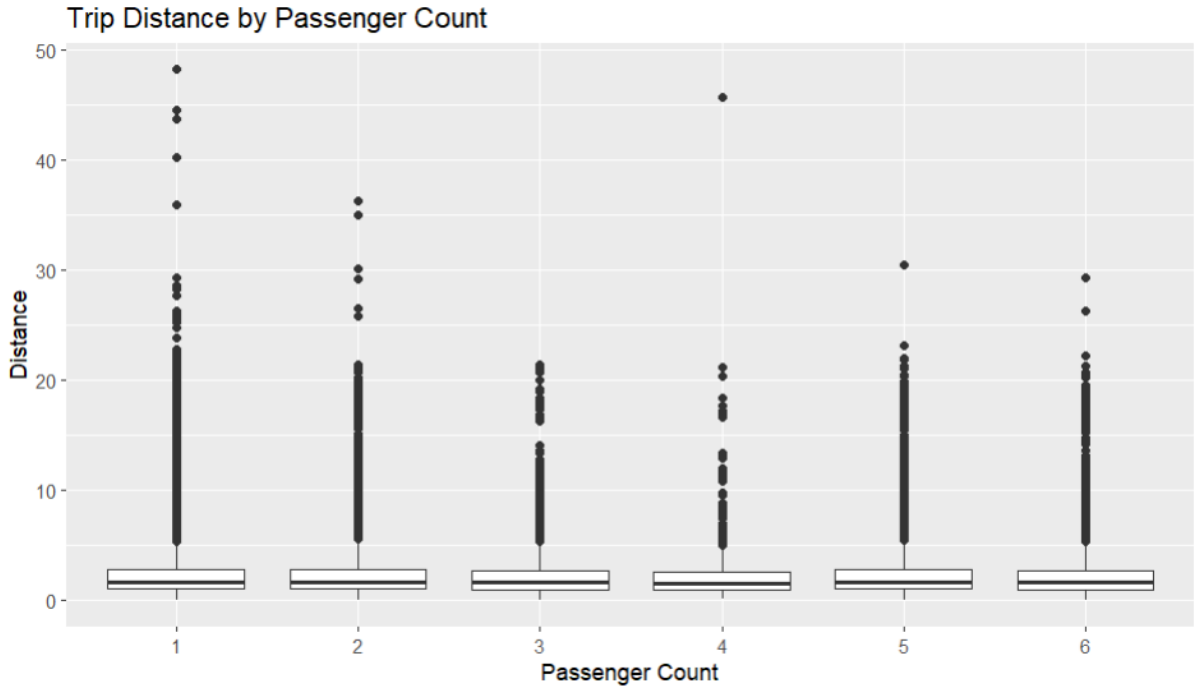
- **Description:** A type of graph such as a vertical bar chart showing distribution of trip frequency depending on the number of people.
- **Key Findings:**
  - **Single-Passenger Dominance:** As seen on the attached chart, the number and proportion of 'single passenger' trips is clearly the most prevalent, accounting for a much greater proportion than the other choices.
  - **Moderate Two-Passenger Trips:** The second highest occurrence is trips with two people while the difference when compared to the highest first is significant.
  - **Low Multi-Passenger Counts:** Numbers of three and four in terms of passenger count represents few numbers of passenger trips requirinf data record. Frequencies of five and six passengers are little more elevated than three and four, indicating some popular group travel.
  - **Dramatic Decline**: From the chart one can notice a sharp decline that is precipitated by an increase in the passenger's number.
- **Business Implications:**
  - Allocate more effort for the development of single-passenger journeys that are prevalent in most cases.
  - Check two-passenger traveling trends for opportunities or need for services.
  - Multi-passenger services could have been suitable opportunities especially for passengers 5 or 6 at most.

- Trip Distance vs. Total Amount (Scatter Plot)



- **Description:** Distribution of trip distance (x axis, in miles) and the corresponding total amount charged (y axis, in dollars) in the form of a scatter plot.
- **Key Findings:**
    - **Positive Correlation:** In the present study, the hypothesis of positive and significant linear relationship between trip distance and the total amount charged is confirmed.
    - **Cluster of Short Trips:** The majority of the data points are clustered in the lower quantities of distances and discrete amount of money values, which may Be due to the trivial overnight short travels.
    - **High-Distance, High-Cost Outliers:** Several outliers show that there are trips with high charges due to long distances and may contain cases of special services.
    - **Even Spread Beyond the Cluster:** If the distance is more than 10 miles the bar for total amounts widens suggesting different fare systems.
- **Business Implications:**
    - They should therefore ensure that the unit price of a short trip is ideal because such trips are most common.
    - Give special offers for superiors' charges or extra options for long traveling sales since those types yield high revenues.
    - Examine the excluded charges to determine what creates the need for higher charges example: tolls, extra services or peak hours.

➢ Trip Distance by Passenger Count (Box Plot)



- **Description:** A box plot showing the distribution of the trip distance with respect to saturated passenger counts.
- **Key Findings:**
  - **Median Stability:** Overall, median trip distances are fairly unaffected by changes in passenger load.
  - **Wide Variability:** There are outliers in every column which suggest that long distance travel is independent of the number of passengers.
  - **Single-Passenger Outliers:** One-passenger trips exhibit the highest variability, as it can be expected, thanks to occasional very long rides.
  - **Comparable Ranges:** The box plot of trip distance shows that the interquartile range (IQR) for all passengers is almost equal, and therefore, the range of the typical trip distance is similar among the different passenger classes.
- **Business Implications:**
  - It means that trip-relevant pricing and service should be standardized using the trip distance instead of the number of passengers because the variation is identical.
  - Check the long-distance outliers for there could be some premium services, or a certain trend.

4 Conclusion

The case of Uber trip analysis proves that big data platforms, such as Hive and statistical tools like R can be a great enhancer. Given how Hive is suited to work with high volume data while R was designed to perform statistical computation, having both tools allowed for the ability to preprocess and clean large complex datasets to gain new insights.

This approach was important in identifying operational parameters such as trip distance, fare structure, and zones of substantially high activity. By using Hive queries, we were able to transform and analyze massive datasets and also define regularity, popular time, and convenience of the trips. It is in R aided by mouse-over and statistical visualization where patterns like distance–fare partnership, one-passenger's trips, and geometric density of pickup and points of drop off were seen.

The implications for practice from such work could include providing efficient driver frequency, management of price tactics, and minimum customer delay. The integration of Hive and R was also a good example of this fact illustrating the possibility to use big data tools in parallel with the statistical methods for the scalable and highly meaningful analysis and as a base for addressing the range of similar problems in the context of ride-sharing and beyond.

This research also confirms how analytics can help improve the satisfaction of customers and efficiency of service delivery organisations such as Uber. Future work can try to build upon this approach by integrating contaminated real time data analysis and prediction models that can enhance the prediction and the strategies generated.