>>> Where are the numbers? >>>

Unravelling New York City's Yellow Taxi Trip Business Insights To Improve Revenue

Advanced Data Analytics and Actionable Recommendations for SHU Consultancy

Sheffield Hallam University | Faculty of Science, Technology & Arts

The Team – Group 4

In this project, a cohesive team with interdependent roles collaborated to achieve success. Team members' previous work experience, expertise and area of interest were useful in the roles and responsibility allocation.

The project manager who has consistently demonstrated strong leadership skills was assigned based on his years of experience in project management.

The data engineer's experience with Linux made a useful impact in ensuring and maintaining the data infrastructure.

The Business analyst research skills were useful in conducting thorough background of the NYC Taxi industry and project use case.

The data visualization engineer with their experience with data visualization tools such as Tableau and Microsoft power BI ensures creative and informative data insight.

Throughout the project, all team members closely collaborated, fostering effective communication and ensuring the project's seamless completion.

Project Manager

Fatola, 'Joba - 32074392

- Oversees the entire data analytics project, from inception to completion, ensuring it stays on track with the objectives/deliverables and meets deadlines.
- His leadership and coordination skills are essential to ensure the project is delivered on time and meets the client's expectations.

Business Analyst

Arunachalam Rajendran, Senthilnathan - 32068795

- Conduct thorough data requirements gathering and analysis
- Perform data validation and quality checks to ensure data accuracy and integrity.
- His expertise in understanding business needs and translating them into data requirements is essential for creating effective data analytics solutions that provide tangible business value

Data Engineer

Hewa Devundarage, Lahiru Madumal - 32073073

- Design and develop data pipelines to extract, transform, and load (ETL) data from various sources into the data warehouse.
- Implement data integration processes, ensuring data consistency and integrity.
- Design and maintain the data warehouse architecture, including data modelling and schema design.

Data Analyst

Duraisamy, Madhankumar - 32070486

- Ensure necessary extraction of required datasets from different sources and transformation including cleaning and validation are done to NYC Yellow taxi trip records during staging

Data Visualization Engineer

Hansrod, Mohmed Samad – 32076141

- Develop interactive and visually appealing data dashboards and reports. Utilize data visualization tools and techniques to communicate complex insights effectively.
- His expertise in data visualization tools and techniques enables them to communicate complex data findings visually engagingly, helping stakeholders make informed decisions.

Project Overview

Introduction

In this project, we delve deep into the wealth of data collected by the Taxi & Limousine Commission (TLC) to uncover valuable insights and trends hidden within NYC's bustling streets.

Background

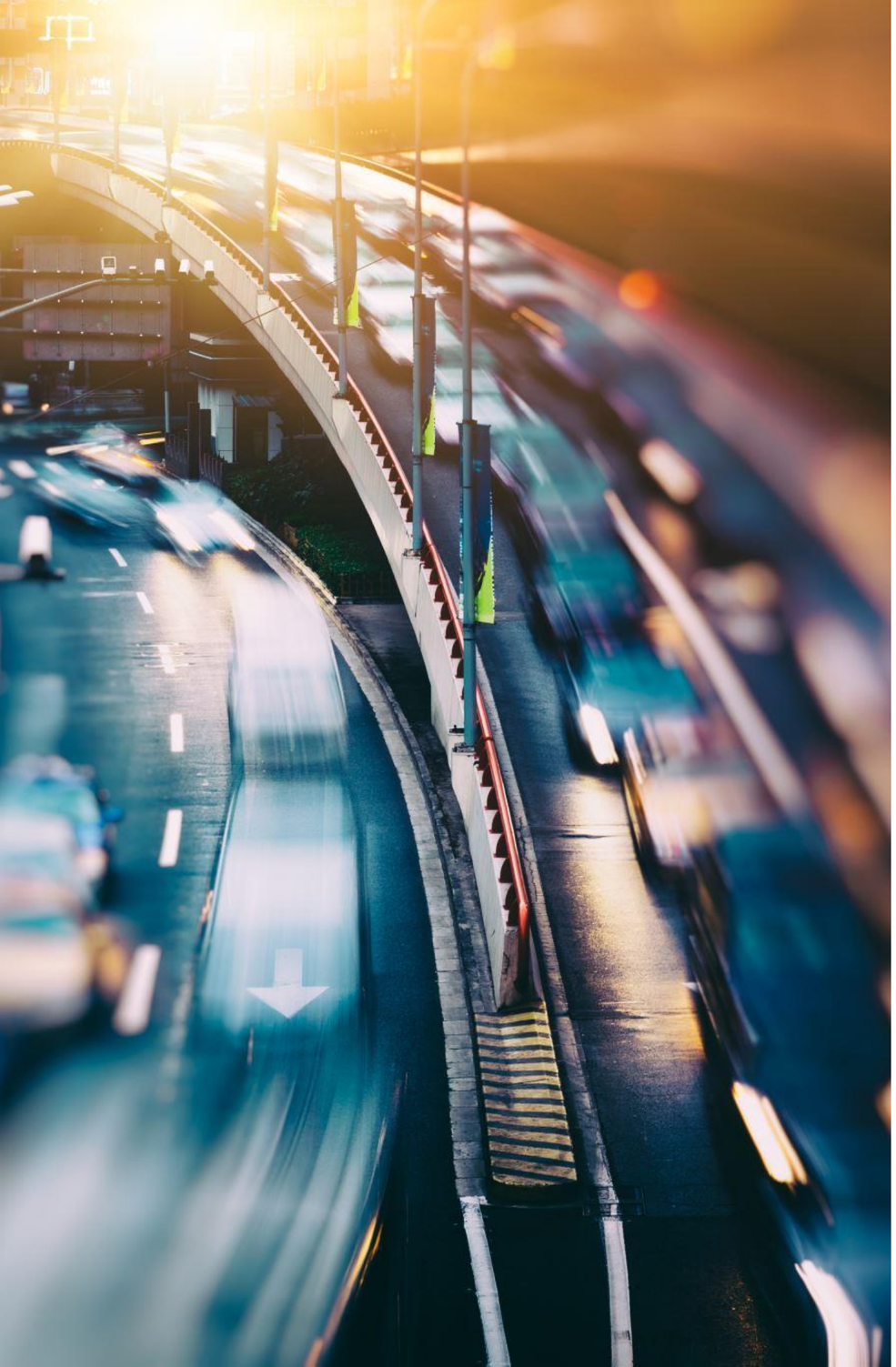
The New York City Yellow Taxi industry has been facing significant challenges in recent years, including the emergence of ride-hailing services, the medallion crisis, and the adverse impact of the COVID-19 pandemic. These issues have led to declining ridership, financial hardships for drivers, and regulatory complexities. According to Wu et al. (2021), the development and quick growth of ride-hailing services has had a negative impact on taxi drivers' income, with lower fares and overall profits because of increasing competition. Understanding characteristics such as passenger demand patterns and trip dispersion is critical in dealing with drivers' low salaries and economic insecurity, (Saia, 2021). Fare reform that considers driver expenses, operating costs, and the changing transportation scenario is also required. By delving into the data meticulously maintained by the Taxi & Limousine Commission (TLC), we aim to gain valuable insights into passenger demand patterns, trip trends, peak hours, and significant geographical locations. This background study sets the stage for leveraging data analytics to unlock hidden opportunities and improve the efficiency and effectiveness of Yellow Taxi services.

Purpose

By employing advanced data analytics techniques, we aim to transform raw data into actionable recommendations for SHU Consultancy. This project is aimed to explore how data insights can elevate taxi services, optimize operations, and ultimately enhance the passenger experience in the iconic Yellow Taxis of New York City. And, ultimately answer the question "Where and when are the numbers" for the SHU Consultancy management team.

Project Requirements Brief

- Collect and clean extensive Yellow Taxi trip records data from the Taxi & Limousine Commission (TLC) and NYC neighbourhood tabulation data.
- Utilize advanced data analytics techniques to identify passenger demand patterns, trip trends, and key performance indicators, including income from trips and trip duration.
- Create interactive data visualizations using Tableau to present actionable insights.
- Collaborate with SHU Consultancy and ensure data privacy and security standards adherence.
- Deliver a comprehensive report with clear and pragmatic recommendations and provide a summary presentation for SHU Consultancy's informed decision-making.



Project Benefits

Below are some of the benefits of this project:

- **Sustainable Revenue Growth:** Providing knowledge about peak times of the day and high-demand zones and leveraging on these times can help in boosting revenue
- **Enhancing Taxi Services and Efficiency**: The project can offer insight to allocate resources efficiently, reduce idle times and increase the overall efficiency of the NYC taxi commission.
- **Passenger Experience Enhancement**: Predictive analysis of demand patterns can help position drivers in high-demand areas, reducing passenger wait times and enhancing their experience
- **Data-Driven Decision making**: Government agencies and policymakers can use findings to implement evidence-based regulations and policies that improve the taxi industry's sustainability
- **Positive Economic Impact**: Enhancing the economic viability of taxi services can lead to improved job stability and income opportunities for drivers, contributing to a resilient workforce.
- **Safety and Security**: Knowledge of peak times and areas can enable better coordination with emergency services during high-demand periods, ensuring swift responses.

Business Questions

Provide a month-by-month overview of the total income from the NYC Yellow Taxi by borough from January 2013 – December 2022.

Justification

- This will help to understand the income distributions across different boroughs over 10 years and provide insight into revenue generation patterns by boroughs in NYC

Dimensions: Time, Borough
Granularity: Monthly
Metrics: Total Income

1

Provide an overview of the monthly total income by NYC Tazi Zone neighbourhood from January 2013 – December 2022

Justification

- Monitoring the monthly income generated by Yellow taxis in different neighbourhoods allows for revenue analysis and identifying lucrative areas for taxi services. This analysis can reveal financial trends and contribute to financial planning and forecasting.

Dimensions: Time, Neighbourhood
Granularity: Monthly
Metrics: Total Income

2

For the top 10 neighbourhoods with the highest income, provide a month-by-month comparison of the total number of trips, trip duration, and distances by TLC Yellow taxis by Taxi Zone neighbourhood from January 2013 - December 2022.

Justification

- This question delves deeper into the top-performing neighbourhoods, exploring their taxi service metrics at a more granular level helps to identify patterns and relationships between passenger demand, distances travelled, trip duration and income.

Dimensions: Time, Taxi Zone Neighbourhood
Metrics: Total number of trips, Total distance covered, Trip duration and Total Income

3

Business Questions *contd.*

For the top 10 taxi zone neighbourhoods with the highest income, what are the peak days of the week (weekdays or weekends) with the highest total income on a monthly basis from January 2013 - December 2022?

Justification

- Understanding peak times of the week for the top income-generating neighbourhoods aids in strategic planning, resource management, and service optimization. Understanding this will aid in optimizing taxi services for profitability.

Dimensions

- Time: Day of the Week (Weekday/Weekend)
- Taxi Zone Neighbourhood, : Top 10 high-income areas in New York City.

Granularity: Month

Metrics: Total Income

4

What time of the day (Morning, Afternoon, evening, and night) do we have the highest income from the top 10 Taxis Zone neighbourhood monthly from Jan 2013 - December 2022?

Justification

This question investigates the time periods during which specific neighbourhoods generate the highest income, helping to optimize taxi services for profitability. We gain a comprehensive understanding of when and where revenue potential is at its highest supporting efficient resource allocation.

Dimensions:

- Day Time: Morning, Afternoon, Evening, Night.
- Taxi Zone Neighbourhood: Different geographical areas in New York City.

Metrics: Total Income

5

Data Collection – Dataset 1

New York City Yellow Taxi Trip Records: The TLC shares detailed records of NYC taxi rides, submitted monthly by bases and stored in parquet format for each month. There are **19 columns in with millions of observations** in each monthly parquet file. **Data types** in this dataset include strings, doubles, and integers. *details in Table 1.*

Key Features: tpep_pickup_datetime, tpep_dropoff_datetime, trip_distance, PULocationID, DOLocationID, fare_amount, tip_amount, and total_amount

Unessential features: VendorID, RatecodeID, store_and_fwd_flag, payment_type, extra, mta_tax, tolls_amount, improvement_surcharge, congestion_surcharge, airport_fee, passenger_count,

Data Period: Jan 2013 – December 2022

Source hyperlink:

TLC Trip Record Data

(NYC Taxi & Limousine Commission)

Key Search Terms:

"NYC", "Yellow Taxi", "Data", "Transportation"

Common Identifier

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	airport_
2	01/01/2023 00:32	01/01/2023 00:40	1	0.97	1	N	161	141	2	9.3	1	0.5	0	0	1	14.3	2.5	
2	01/01/2023 00:55	01/01/2023 01:01	1	1.1	1	N	43	237	1	7.9	1	0.5	4	0	1	16.9	2.5	
2	01/01/2023 00:25	01/01/2023 00:37	1	2.51	1	N	48	238	1	14.9	1	0.5	15	0	1	34.9	2.5	
1	01/01/2023 00:03	01/01/2023 00:13	0	1.9	1	N	138	7	1	12.1	7.25	0.5	0	0	1	20.85	0	1.
2	01/01/2023 00:10	01/01/2023 00:21	1	1.43	1	N	107	79	1	11.4	1	0.5	3.28	0	1	19.68	2.5	
2	01/01/2023 00:50	01/01/2023 01:02	1	1.84	1	N	161	137	1	12.8	1	0.5	10	0	1	27.8	2.5	
2	01/01/2023 00:09	01/01/2023 00:19	1	1.66	1	N	239	143	1	12.1	1	0.5	3.42	0	1	20.52	2.5	
2	01/01/2023 00:27	01/01/2023 00:49	1	11.7	1	N	142	200	1	45.7	1	0.5	10.74	3	1	64.44	2.5	
2	01/01/2023 00:21	01/01/2023 00:36	1	2.95	1	N	164	236	1	17.7	1	0.5	5.68	0	1	28.38	2.5	
2	01/01/2023 00:39	01/01/2023 00:50	1	3.01	1	N	141	107	2	14.9	1	0.5	0	0	1	19.9	2.5	
2	01/01/2023 00:53	01/01/2023 01:01	1	1.8	1	N	234	68	1	11.4	1	0.5	3.28	0	1	19.68	2.5	
1	01/01/2023 00:43	01/01/2023 01:17	4	7.3	1	N	79	264	1	33.8	3.5	0.5	7.75	0	1	46.55	2.5	
2	01/01/2023 00:34	01/01/2023 01:04	1	3.23	1	N	164	143	1	26.1	1	0.5	6.22	0	1	37.32	2.5	
2	01/01/2023 00:09	01/01/2023 00:29	2	11.43	1	N	138	33	1	44.3	6	0.5	13.26	0	1	66.31	0	1.
2	01/01/2023 00:33	01/01/2023 00:49	1	2.95	1	N	33	61	1	17.7	1	0.5	4.04	0	1	24.24	0	
2	01/01/2023 00:13	01/01/2023 00:22	1	1.52	1	N	79	186	1	10	1	0.5	1.25	0	1	16.25	2.5	
2	01/01/2023 00:45	01/01/2023 01:07	1	2.23	1	N	90	48	1	19.8	1	0.5	4.96	0	1	29.76	2.5	
1	01/01/2023 00:04	01/01/2023 00:19	1	4.5	1	N	113	255	1	20.5	3.5	0.5	4	0	1	29.5	2.5	
1	01/01/2023 00:03	01/01/2023 00:09	3	1.2	1	N	237	239	2	8.6	3.5	0.5	0	0	1	13.6	2.5	
1	01/01/2023 00:15	01/01/2023 00:29	2	2.5	1	N	143	229	2	15.6	3.5	0.5	0	0	1	20.6	2.5	
1	01/01/2023 00:51	01/01/2023 00:58	1	1.4	1	N	137	79	1	9.3	3.5	0.5	2.85	0	1	17.15	2.5	
1	01/01/2023 00:13	01/01/2023 00:44	1	17.8	2	N	132	116	1	70	1.25	0.5	15.85	6.55	1	95.15	0	1.
1	01/01/2023 00:21	01/01/2023 00:29	4	0.8	1	N	163	161	4	8.6	3.5	0.5	0	0	1	13.6	2.5	
1	01/01/2023 00:52	01/01/2023 01:02	2	1.7	1	N	161	164	4	11.4	3.5	0.5	0	0	1	16.4	2.5	
2	01/01/2023 00:19	01/01/2023 00:38	1	5.7	1	N	161	87	1	26.8	1	0.5	6.36	0	1	38.16	2.5	
2	01/01/2023 00:31	01/01/2023 00:51	1	1.18	1	N	68	164	1	17	1	0.5	6.6	0	1	28.6	2.5	
2	03/01/2023 09:35	03/01/2023 09:53	1	2.45	1	N	43	100	2	17	0	0.5	0	0	1	21	2.5	
2	03/01/2023 09:22	03/01/2023 09:25	1	0.57	1	N	113	90	1	5.8	0	0.5	1.96	0	1	11.76	2.5	
2	03/01/2023 09:56	03/01/2023 10:08	1	1.83	1	N	237	239	1	13.5	0	0.5	4.38	0	1	21.88	2.5	
2	03/01/2023 09:05	03/01/2023 09:11	1	1.17	1	N	237	161	2	8.6	0	0.5	0	0	1	12.6	2.5	
2	03/01/2023 09:19	03/01/2023 09:31	1	1.51	1	N	163	236	1	12.8	0	0.5	3.36	0	1	20.16	2.5	
2	03/01/2023 09:43	03/01/2023 10:04	1	2.63	1	N	238	140	1	19.8	0	0.5	3.57	0	1	27.37	2.5	
2	03/01/2023 09:01	03/01/2023 09:09	1	1.19	1	N	234	114	1	9.3	0	0.5	2.66	0	1	15.96	2.5	
2	03/01/2023 09:13	03/01/2023 09:23	1	2.56	1	N	79	162	1	13.5	0	0.5	3.5	0	1	21	2.5	

Figure 1: *Preview of TLC Trip Record Data from https://www.nyc.gov/*

NB: *We have decided to keep trip records for the year 2020 even though we suspect there might be a huge impact from the Covid pandemic. Observations of this impact before, during and post covid shall be presented in the appendix.*

Data Collection – Dataset 2

NYC Neighbourhood: NYC Taxi Zones record corresponds to the pickup and drop-off zones, or LocationIDs, included in the Yellow TaxiTrip Records. The taxi zones are roughly based on the NYC Department of City Planning's Neighborhood Tabulation Areas (NTAs) and are meant to approximate neighbourhoods, so you can see which neighbourhood a passenger was picked up in, and which neighbourhood they were dropped off in.

The dataset consists of **263 rows , 7 columns and a total of 1,841 observations**.

Data types in this dataset include strings, and integers.

- **Key Features**: LocationID, borough, zone (same as a neighborhood), the_geom
- **Data Period:** Last updated on March 8, 2019. (Update frequency – as needed)
- **Source hyperlink:** NYC Neighbourhood (NYC Open Data Source)
- **Key Search Terms:** "taxi zones", "neighbourhoods", "taxi", "Transportation", "NYC"

Common Identifier

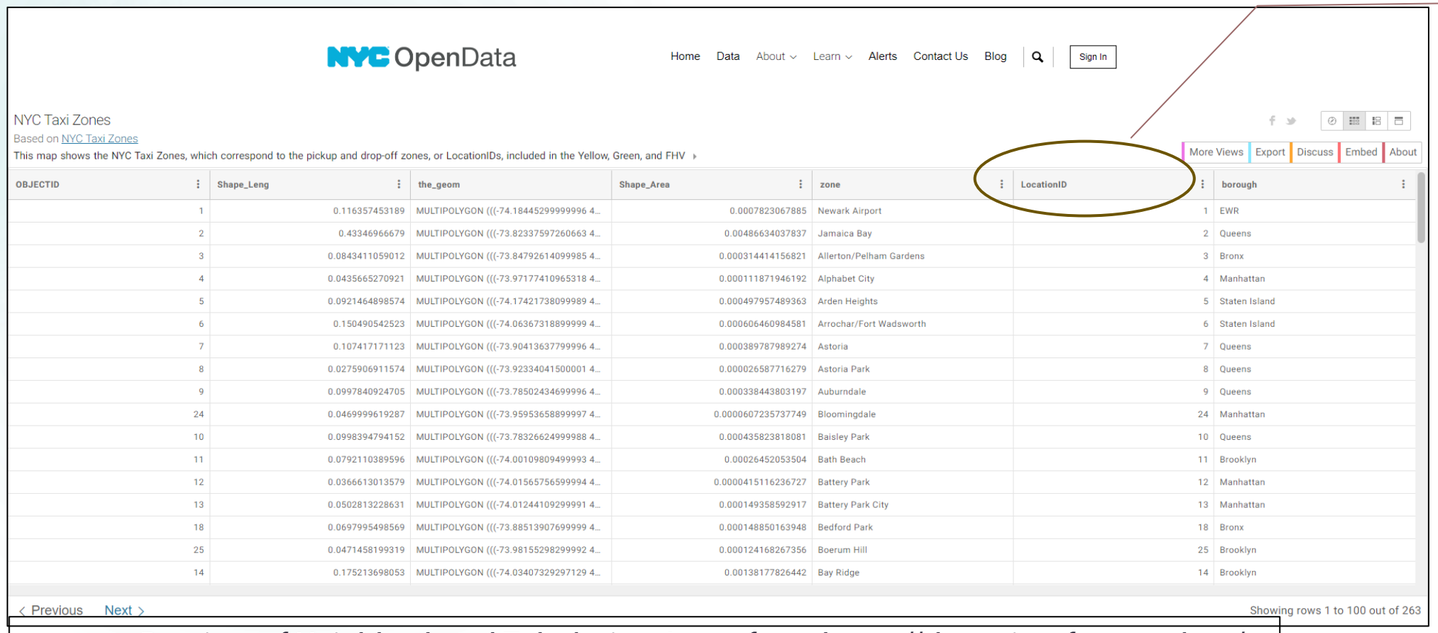


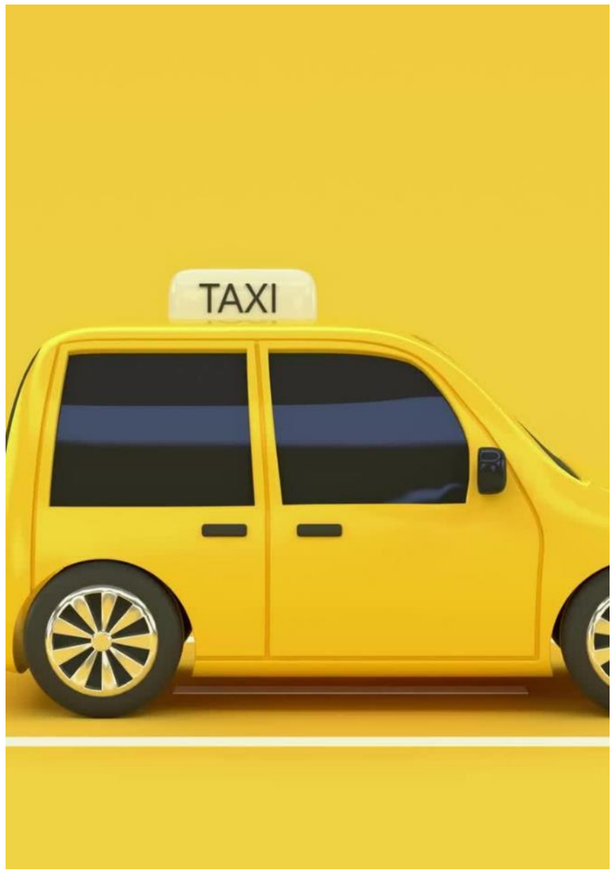
Figure 2: Preview of Neighborhood Tabulation Areas from https://data.cityofnewyork.us/

Data Dictionary - Column Information

Column Name	Column Description
borough	the Borough that the zone is located within
location_ID	the number of the zone, which can be joined to TLC trip record LocationIDs
objectID	the number of the zone, which can be joined to TLC trip record LocationIDs
shape_area	the area of the zone in map units
shape_leng	the length of the zone in map units
zone	The name of the taxi zone

Figure 3: Preview of NYC Taxi Zone data dictionary

Justification for the dataset



- **Integrity and Reliability of source:** The NYC yellow taxi trip records and NYC Taxi zone records were obtained from reliable and approved government agencies. The NYC Taxi and Limousine Commission is authorized under the Taxicab & Livery Passenger Enhancement Programs (TPEP/LPEP), (NYC.GOV).
- **Use Case:** A combination of the 2 datasets will provide useful information for this project use case. Combining trip records with the taxi zones table which provides further insight into the neighbourhood in which the trip happens will provide useful insight that supports this project's aims.
- **Compliance**: Both datasets used in this project are obtained from the government-approved open-source data which is recommended for research project works. Open-source government data are publicly accessible, reliable and promote compliance with GDPR.
- **Usability:** Open-source data also promotes the reusability of these datasets by other researchers for continuous learning and improving the body of knowledge

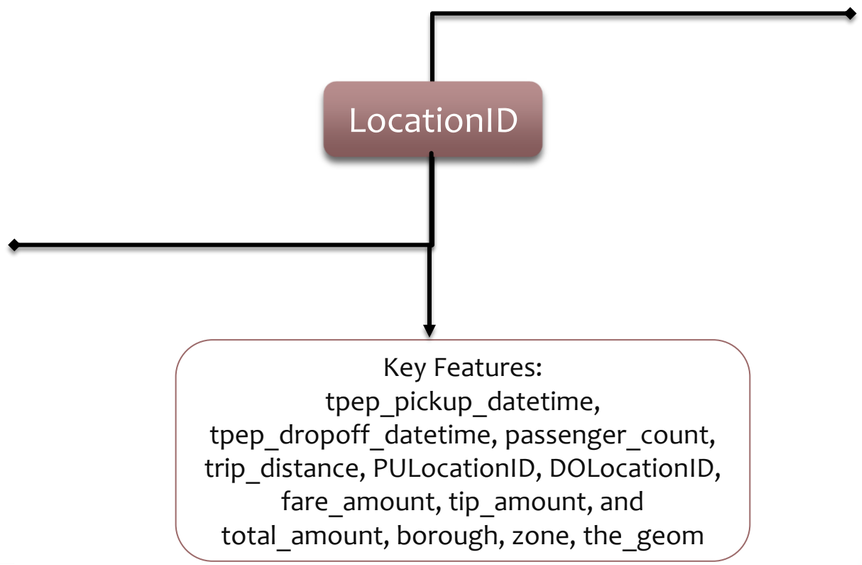
Data Integration

- **Dataset 1: New York City Yellow Taxi Trip Records (2013 -2022)**
  - As mentioned earlier, each monthly parquet file which contains daily trip records retrieved from the NYC.gov website is combined into a single monthly file after ensuring that the column names and number index match for all files. Details of the R programming script can be found in the appendix.
  - These steps left us with 10 yearly files versus 120 files retrieved from the source. Table 1 shows the summary of the combined monthly trip records.
- **Why is this important?**
  - This will allow the project team to have very good referential integrity, and ease and enhance the efficiency of the data warehouse process.
- **Data Join:**
- Data set 1 (**New York City Yellow Taxi Trip Records**) will be connected to data set 2 (NYC Taxi Zones) using the "LocationID" as a common identifier between these 2 data sets after transformation.

NYC Yellow Taxi Yearly Data Summary

Year	No of Rows	No of Column	Total number of observations
2013	171,816,340	19	3,264,510,441
2014	165,447,579	19	3,143,503,982
2015	146,039,231	19	2,774,745,370
2016	131,131,805	19	2,491,504,276
2017	113,500,327	19	2,156,506,194
2018	102,871,387	19	1,954,556,334
2019	84,598,444	19	1,607,370,417
2020	24,649,092	19	468,332,729
2021	30,904,308	19	587,181,833
2022	39,656,098	19	753,465,843
Total	**1,010,614,611**	**19**	**19,201,677,419**

Table 1: *Overview of combined monthly records of yellow taxi trips from 2013 - 2022*



Taxi Zone Dataset Summary

No of rows	264
No of Columns	7
Total Observations	**1,841**

Table 2: *Summary of Taxi Zone dataset*

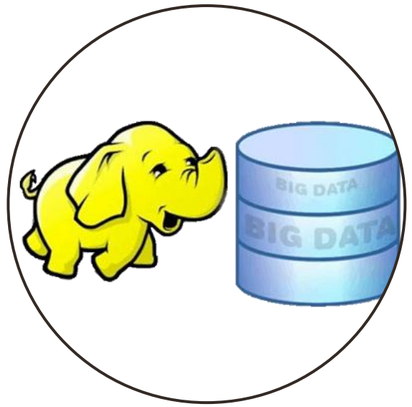
DI and BDDS

Data Integration and Big Data & Distributed Systems Tools

The tools chosen for this project have been thoughtfully selected to align with its specific requirements. The selection process took into account the volume(scale), variety(accuracy, quality, integrity and credibility), and variety (diversity) of the datasets involved, while also considering the potential insights (value) to be derived from this NYC analytics venture.

R-Studio (DI)

- **Use:** Extraction, data preprocessing(e.g. combining monthly parquet trip record files) including data quality checks, data cleaning, data visualization and EDA of the 2 datasets.
- **Justification:** Its integrated development open-source environment will empower us to efficiently manipulate and visualize datasets with maximum cost savings.
- Python on the other hand would have been useful if the project involves machine learning. R is preferred because of it specialized collection of libraries for data manipulation and analysis.



Hadoop Distributed File System (BDDS)

- **Use**: Primary storage system to store both structured and unstructured NYC trip and taxi zone records. Also, to manage datasets in blocks across multiple nodes in the Hadoop cluster.
- **Justification**: The HDFS provides fault tolerance, scalability, reliability and high throughput for storing and processing massive datasets in parallel (Ji et al., 2019). So, this is best suited for this project since the size of the NYC yellow taxi trip records is not able to fit our single machine, fault tolerance and scalability is an advantage especially when big data processing frameworks like Apache Spark or Apache Hadoop MapReduce for more complex computations. This will also help us integrate well with Apache Hive we will be using for our data warehousing. It will also support the storage of large datasets in a distributed manner

Apache Hive (BDDS)

- **Use:** Data warehousing, filtering, querying and aggregation, dimension table creation, data partitioning and optimization
- **Justification**: Hive translates SQL queries into MapReduce jobs, enabling data analysts and engineers to leverage their SQL skills to analyze and process big data (Sumalatha et al., 2021). Apache Hive complements HDFS by allowing SQL-like querying on big data. Choosing Hive facilitates data extraction and transformation using familiar SQL commands. This tool's optimization capabilities align with our project's need for efficient querying.

Tableau (Visualization)

- **Use:** Data storytelling, reporting, interactive dashboard
- **Justification:** For ease of direct connectivity to the this project data engineering and warehousing Tabluea is the preferred choice over another visualization tool like Microsoft power BI which will require steps such as extracting data from HDFS/Hive, transforming it into a format that Power BI can work with (e.g. CSV, Excel). Tableau's selection is based on its prowess in data visualization and reporting. Given the aim to communicate insights effectively, Tableau's interactive dashboards and user-friendly interface will empower the team to create engaging visualizations for SHU consultancy management team.

Project Solution Architecture

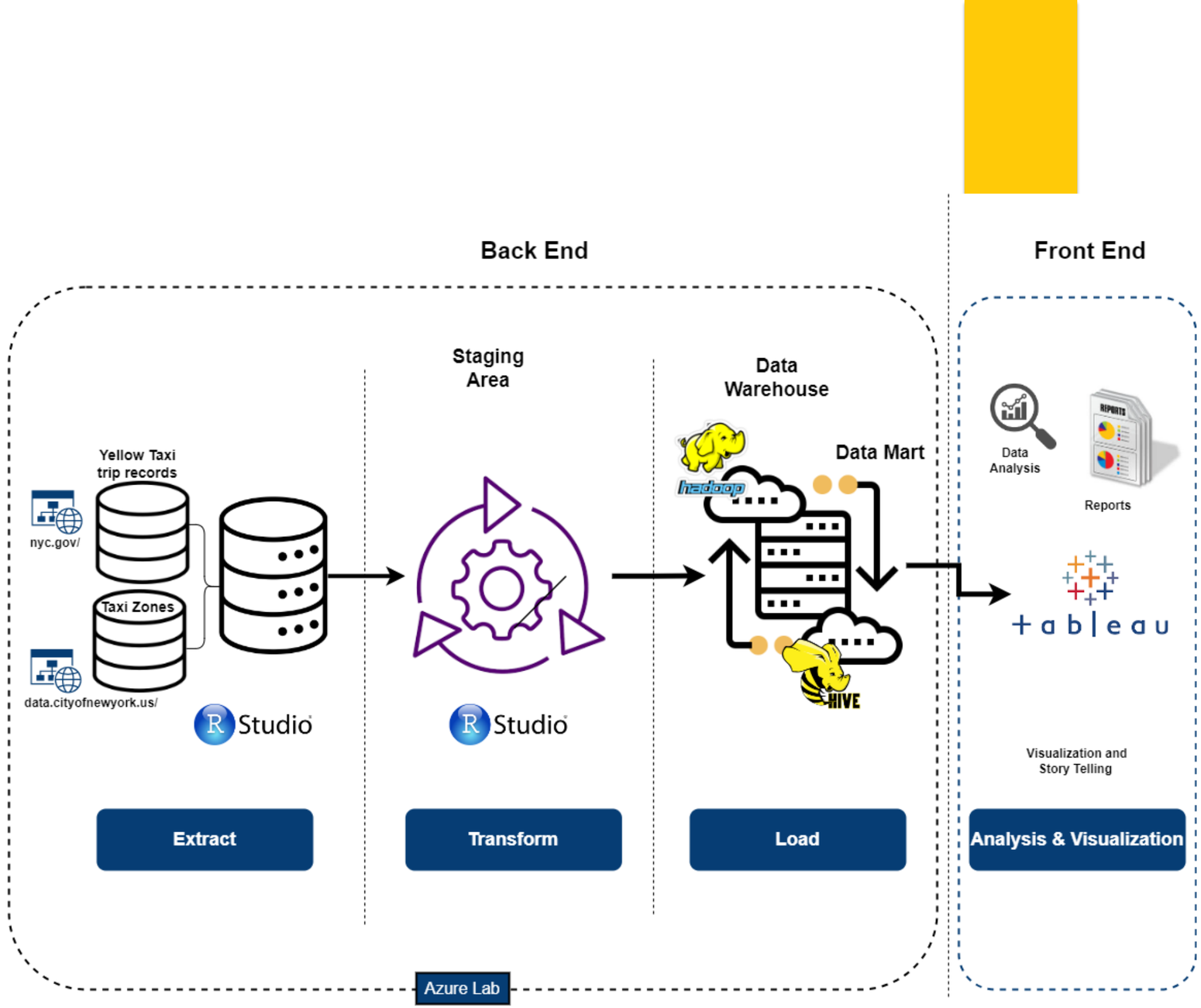


Figure 4: Overview of the solution process adopting the Kimball bottom-up methodology including the use of DI & BDDS tools selected for NYC Yellow Taxi designed by the group 4 team using draw.io

Data Integration Method Justification

The ETL process is the preferred choice for this Yellow Taxi data analysis project due to its capability to ensure data quality, apply business logic, handle historical data effectively, and optimise data integration and performance. These attributes align well with the project's goals of providing reliable insights and actionable recommendations for SHU Consultancy's decision-making.

ETL- Extract, Transform, Load

- **Data Quality and Cleansing:** Enables data cleansing and transformation before loading into data warehouse ensuring reliable and accurate insight
- **Scalability and Maintenance:** Modular structure facilitates easier scalability and maintenance as the project evolves or new data sources are added ensuring project can grow in line with business needs. Errors are easy to trace and corrected.
- **Data Governance and Security:** ETL provides transformation and encryption of sensitive data before loading into target system reducing the risk of GDPR issues.
- **Historical Data Handling:** Preserves historical context during transformation
- **Performance Optimization**: Allows for optimization during transformation

ELT – Extract, Load Transform

- **Data Quality and Cleansing:** Data cleansing might be limited and complicated after loading which can impact the reliability of insight
- **Scalability and Maintenance:** Modular structure facilitates easier scalability and maintenance as the project evolves or new data sources are added ensuring project can grow in line with business needs. Error troubleshooting and modification can be complex as the data is already in the system
- **Data Governance and Security:** ETL provides transformation and encryption of sensitive data before loading into target system reducing the risk of GDPR issues.
- **Historical Data Handling:** Handling historical context might be complex as the data is already in the target system
- **Performance Optimization**: Performance optimization might be limited after loading

Data Warehousing

The bottom-up Kimball approach, with its emphasis on dimensional modelling and providing quick access to business insights, aligns well with the analysis of historical data for the NYC Yellow Taxi project.

Table 2 shows some comparison how historical data compatibility supports the choice of Kimball over the Inmon approach:

Area	Kimball Data Warehousing Methodology	Inmon Approach
Business Focus	Business-centric and allow dimensional models for analysis which fits the project's requirement of providing actionable insights and recommendations for improving taxi services and earnings.	Enterprise-level, and requires normalized data for consistency
Data Modelling	Simpler relational models like star or snowflake schemas for ease of use. This aligns well with the project's goals of analysing data based on dimensions like time, location, and income.	It's relational model requires highly normalized structures for comprehensive data
Time performance	Optimized for time-based analysis and trends. Promotes building the data warehouse incrementally, allowing for agile development and quicker time-to-insight, which can be beneficial for getting results in a timely manner.	May require additional time for data normalization
Query performance	Pre-aggregation can enhance query speed	Complex queries due to normalized data structures
Data Integration	Can handle data integration from different sources, which is crucial for combining Yellow Taxi trip records, income data, and other related information. Integrates data before loading into the warehouse	Centralizes data and then integrates into the system
Data Governance	Allows for business-specific data rules	Centralized governance with a single source of truth
Cost	Lower cost due to simpler structure and processes	Higher cost due to complex integration and structure
End-user empowerment	Well-suited for end-users to create their own queries and reports, aligning with the project's aim to provide insights for SHU Consultancy's decision-making.	Might require IT involvement for data extraction
Business agility /Ease of use	Adaptable to changing business needs. Intuitive design for easier interaction	Less flexible due to centralized data architecture. Complexity may require training for end-users

Table 3: *Comparison of Kimball and Inmon Data Warehousing Methodology*

ETL Process Overview

Extract

- Extract monthly of NYC Yellow taxi trip records from 2013 to 2022 from NYC.gov
- Check for column consistency
- Append monthly files to obtain yearly record
- Keep only ten files for data transformation
- Obtain Taxi zone neighborhood data from cityofnewyork.us

Transform

- Check for missing or null values
- Select essential columns
- Data type mismatch
- Rename columns
- Incorrect data / Invalid
- Derive new fields (income & Time ID)
-

Load

Load cleaned data to Hive

Create dimension and fact tables

Data Extraction

Dataset 1- NYC Yellow Taxi

- Connecting to the website to download trip record monthly files file

```
for (y in 2013:2022) {

  year = y

  for (x in 1:12) {

      month = ""

      if(x < 10) {
        month = paste("0", x, sep="")
      } else {
        month = x
      }
    url <- paste("https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_",year ,"-", month, ".parquet",sep="")

    # Directory where you want to save the downloaded file
    download_dir <- "D:/trip-data/"

    # Create the full path to the downloaded file
    download_path <- paste("D:/trip-data/program/yellow-data/yellow_tripdata_",year ,"-", month, ".parquet",sep="")

    # Download the file
    download.file(url, destfile = download_path, method = "auto")

  }
}
```

```
+     download_path <- paste("D:/trip-data/program/yellow-data/yellow_tripdata_",year ,"-", month, ".parquet",sep="")
+
+     # Download the file
+     download.file(url, destfile = download_path, method = "auto")
+
+   }
+ }
trying URL 'https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2013-01.parquet'
Content type 'application/x-www-form-urlencoded; charset=utf-8' length 184736650 bytes (176.2 MB)
```

Figure 5: Screenshot of R code to extract monthly parquet files for NYC Yellow taxi for the Years 2013 -2022

Data Extraction

Dataset 1- NYC Yellow Taxi

- Monthly NYC Yellow Taxi Trip records parquet files from 2013 – 2014 were downloaded and merged yearly as shown in fig2.
- A quick column name consistency was done and confirmed okay before merging into yearly file records as shown in fig 1.

```
library(dplyr)
library(arrow)

# Create a list to store column names for each data frame
column_names_list <- list()

# Directory containing your data frames
data_frames_directory <- "D:/SHU Class/ADMP-Assignment/NYC Taxi - Datasets/Yellow Taxi Trip Records/all-data"

# List of data frame file names
data_frame_files <- list.files(path = data_frames_directory, pattern = "\\.parquet$", full.names = TRUE)  # Change pattern if needed

# Read each data frame and store its column names
for (file_path in data_frame_files) {
  df <- read_parquet(file_path)  # Change read function if your files are in a different format
  column_names_list[[file_path]] <- colnames(df)
}

# Check if all data frames have the same column names
print('check column names')
consistent_column_names <- length(unique(column_names_list)) == 1
print('received result')

# Print result
if (consistent_column_names) {
  print("All data frames have consistent column names.\n")
} else {
  print("Data frames have inconsistent column names.\n")
}
```

28:29 (Top Level)

Console | Terminal | Background Jobs

R 4.3.0 · ~/

```
> # Read each data frame and store its column names
> for (file_path in data_frame_files) {
+   df <- read_parquet(file_path)  # Change read function if your files are in a different format
+   column_names_list[[file_path]] <- colnames(df)
+ }
>
> # Check if all data frames have the same column names
> print('check column names')
[1] "check column names"
> consistent_column_names <- length(unique(column_names_list)) == 1
> print('received result')
[1] "received result"
>
> # Print result
> if (consistent_column_names) {
+   print("All data frames have consistent column names.\n")
+ } else {
+   print("Data frames have consistent column names.\n")
+ }
[1] "Data frames have consistent column names.\n"
>
```

Figure 6: Screenshot of R code to confirm column name consistency before appending into single yearly trip record files

```
install.packages("arrow")
install.packages("dplyr")

library(arrow)
library(dplyr)

setwd("D:/SHU Class/ADMP-Assignment/NYC Taxi - Datasets/Yellow Taxi Trip Records/2013")
file_list <- list.files(pattern = "\\.parquet$")
combined_data <- data.frame()

for (file in file_list) {
  # Read the data from the Parquet file
  data <- arrow::read_parquet(file)

  # Append the data to the combined_data data frame
  combined_data <- bind_rows(combined_data, data)
}

arrow::write_parquet(combined_data, "yellowtrip_2013_combined.parquet")
```

Figure 7: Sample screen shot of R code showing how monthly files were combined into yearly trip record file

Data Extraction

Dataset 1- NYC Yellow Taxi

```
install.packages("arrow")

library(arrow)

# extract and read data from trip record files
parquet_file <- "D:/SHU Class/ADMP-Assignment/NYC Taxi - Yellow_trip_Combined/yellowtrip_2013_combined.parquet"
df <- arrow::read_parquet(parquet_file)

dim(df)
head(df)
```

```
[1] "Data frames have consistent column names.\n
> # extract and read data from trip record files
> parquet_file <- "D:/SHU Class/ADMP-Assignment/NYC Taxi - Yellow_trip_Combined/yellowtrip_2013_combined.parquet"
> df <- arrow::read_parquet(parquet_file)
> dim(df)
[1] 171816340        19
> head(df)
  VendorID tpep_pickup_datetime tpep_dropoff_datetime passenger_count trip_distance RatecodeID store_and_fwd_flag PULocationID DOLocationID payment_type fare_amount extra
1        2  2013-01-01 00:39:00   2013-01-01 00:55:00               3          3.86          1              <NA>          238          116            2        15.0   0.5
2        2  2013-01-01 00:12:00   2013-01-01 00:16:00               5          0.00          1              <NA>          264          264            1         3.5   0.5
3        2  2013-01-01 00:02:00   2013-01-01 00:03:00               3          0.00          1              <NA>          264          264            1         2.5   0.5
4        2  2013-01-01 00:38:00   2013-01-01 00:38:00               2          0.00          1              <NA>          264          264            2         2.5   0.5
5        2  2013-01-01 00:03:00   2013-01-01 00:04:00               4          0.00          1              <NA>          264          264            1         3.0   0.5
6        2  2013-01-01 00:03:00   2013-01-01 00:04:00               3          0.00          1              <NA>          146          146            1         2.5   0.5
  mta_tax tip_amount tolls_amount improvement_surcharge total_amount congestion_surcharge airport_fee
1     0.5       0.00            0                     0        16.00                   NA          NA
2     0.5       0.12            0                     0         4.62                   NA          NA
3     0.5       0.25            0                     0         3.75                   NA          NA
4     0.5       0.00            0                     0         3.50                   NA          NA
5     0.5       0.07            0                     0         4.07                   NA          NA
6     0.5       0.25            0                     0         3.75                   NA          NA
>
```

Figure 8: Screen shot of R code extracting, reading and describing dimensions of NYC Yellow Taxi trip records for the year 2013

Data Extraction

Dataset 2- NYC Taxi Zone Neighborhood

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

number of columns.R* | check-column-consistancy.R | combining files _updated.R | col_names&selection.R | Untitled1* | Untitled2* | cleanining_data.R | sum_Inc_col.R | schemas chdck.R

Source on Save | Run | Source

```
1  install.packages("arrow")
2
3  library(arrow)
4  library(dplyr)
5
6  # set file path and read the file
7  file_path <- "D:/SHU Class/ADMP-Assignment/Nyx Taxi - Cleaned Data(Hive)/NYC_Taxi_Zones.xlsx"
8  nyc_taxi_zone <- read_excel(file_path)
9
10 dim(nyc_taxi_zone)
11 head(nyc_taxi_zone)
12
```

12:1 (Top Level) | R Script

Console | Terminal | Background Jobs

R 4.3.0 · ~/

```
> # set file path and read the file
> file_path <- "D:/SHU Class/ADMP-Assignment/Nyx Taxi - Cleaned Data(Hive)/NYC_Taxi_Zones.xlsx"
> nyc_taxi_zone <- read_excel(file_path)
>
> dim(nyc_taxi_zone)
[1] 263   7
> head(nyc_taxi_zone)
# A tibble: 6 × 7
  OBJECTID Shape_Leng the_geom                                                                Shape_Area zone  LocationID borough
     <dbl>      <dbl> <chr>                                                                       <dbl> <chr>      <dbl> <chr>
1        1      0.116 MULTIPOLYGON (((-74.18445299999996 40.694995999999904, -74.18448899999999 40.69509499999987, -74.18449799999996 40.… 0.000782 Newa…        1 EWR
2        2      0.433 MULTIPOLYGON (((-73.82337597260663 40.63898704717672, -73.82277105438692 40.63557691408512, -73.82265046764824 40.6… 0.00487  Jama…        2 Queens
3        3      0.084… MULTIPOLYGON (((-73.84792614099985 40.87134223399991, -73.84725144800001 40.870987051999926, -73.84699273699992 40.… 0.000314 Alle…        3 Bronx
4        4      0.043… MULTIPOLYGON (((-73.97177410965318 40.72582128133706, -73.97179142961485 40.72581422114911, -73.97181650587731 40.7… 0.000112 Alph…        4 Manhat…
5        5      0.092… MULTIPOLYGON (((-74.17421738099989 40.5625680859999, -74.17348507999998 40.56226748999995, -74.17225507699992 40.56… 0.000498 Arde…        5 Staten…
6        6      0.150  MULTIPOLYGON (((-74.06367318899999 40.60219816599994, -74.06350565699998 40.60215005399992, -74.06383152699986 40.6… 0.000606 Arro…        6 Staten…
>
```

Environment | History | Connections | Tutorial

Import Dataset | 522 MiB | List

R | Global Environment

Data

nyc_taxi_zo…	263 obs. of 7 variables

values

file_path	"D:/SHU Class/ADMP-Assignment/Nyx…

Files | Plots | Packages | Help | Viewer | Presentation

Install | Update

Name	Description	Version
User Library		
☑ arrow	Integration to 'Apache' 'Arrow'	12.0.1.1
☐ askpass	Safe Password Entry for R, Git, and SSH	1.1
☐ assertthat	Easy Pre and Post Assertions	0.2.1
☐ backports	Reimplementations of Functions Introduced Since R-3.0.0	1.4.1
☐ base64enc	Tools for base64 encoding	0.1-3
☐ bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.5
☐ bit64	A S3 Class for Vectors of 64bit Integers	4.0.5
☐ blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.4
☐ brew	Templating Framework for Report Generation	1.0-8
☐ brio	Basic R Input Output	1.1.3
☐ broom	Convert Statistical Objects into Tidy Tibbles	1.0.5
☐ bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'rmarkdown'	0.5.0
☐ cachem	Cache R Objects with Automatic Pruning	1.0.8
☐ callr	Call R from R	3.7.3
☐ cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
☐ class	Functions for Classification	7.3-22

Figure 9: Screenshot of R code extracting, reading and describing dimensions of Taxi Zones neighbourhood in data set 2

Data Extraction Summary

```r
for (x in 2013:2022) {
  year = x
  total_rows = 0
  total_size = 0
  total_cols = 0

  for (x in 1:12) {

    month = ""

    if(x < 10) {
      month = paste("0", x, sep="")
    } else {
      month = x
    }

    file_path = paste("D:/trip-data/program/yellow-data/yellow_tripdata_",year ,"-", month, ".parquet",sep="")
    df = read_parquet(file_path)
    file_info <- file.info(file_path)
    # Calculate file size in megabytes
    file_size_mb <- file_info$size / (1024 * 1024)
    rows = nrow(df)
    cols = ncol(df)
    total_rows = total_rows + rows
    total_size = total_size + file_size_mb
  }
  print(paste("rows, columns, size in MB for year", year, "is", total_rows, " " , cols, " " , total_size))
}
```

```
[1] "rows, columns, size in MB for year 2013 is 171816340   19   2044.59596157074"
[1] "rows, columns, size in MB for year 2014 is 165447579   19   2048.72796535492"
[1] "rows, columns, size in MB for year 2015 is 146039231   19   1933.15325927734"
[1] "rows, columns, size in MB for year 2016 is 131131805   19   1748.14667129517"
[1] "rows, columns, size in MB for year 2017 is 113500327   19   1520.8112487793"
[1] "rows, columns, size in MB for year 2018 is 102871387   19   1396.5766582489"
[1] "rows, columns, size in MB for year 2019 is 84598444   19   1185.92541790009"
[1] "rows, columns, size in MB for year 2020 is 24649092   19   357.246046066284"
[1] "rows, columns, size in MB for year 2021 is 30904308   19   457.363823890686"
[1] "rows, columns, size in MB for year 2022 is 39656098   19   586.60475063324"
```

Figure 10: Screen shot of R code describing data now of rows, columns, and file size for yearly trip files after appending.

NYC Yellow Taxi Yearly Data Summary

Year	No of Rows	No of Column	Total number of observations
2013	171,816,340	19	3,264,510,441
2014	165,447,579	19	3,143,503,982
2015	146,039,231	19	2,774,745,370
2016	131,131,805	19	2,491,504,276
2017	113,500,327	19	2,156,506,194
2018	102,871,387	19	1,954,556,334
2019	84,598,444	19	1,607,370,417
2020	24,649,092	19	468,332,729
2021	30,904,308	19	587,181,833
2022	39,656,098	19	753,465,843
Total	**1,010,614,611**	**19**	**19,201,677,419**

Table 4: *Overview of combined monthly records of yellow taxi trips from 2013 - 2022*

Taxi Zone Dataset Summary

No of rows	264
No of Columns	7
Total Observations	**1,841**

Table 5: *Summary of Taxi Zone neighborhood dataset*

Data Transformation

Column Selection

- **NYC Trip Records**: 7 columns were selected ("tpep_pickup_datetime", "tpep_dropoff_datetime","trip_distance", "PULocationID","fare_amount","extra","tip_amount“)

- **Taxi Zone Neighbourhood Dataset**: 4 columns were selected (LocationID, borough, zone, the_geom)

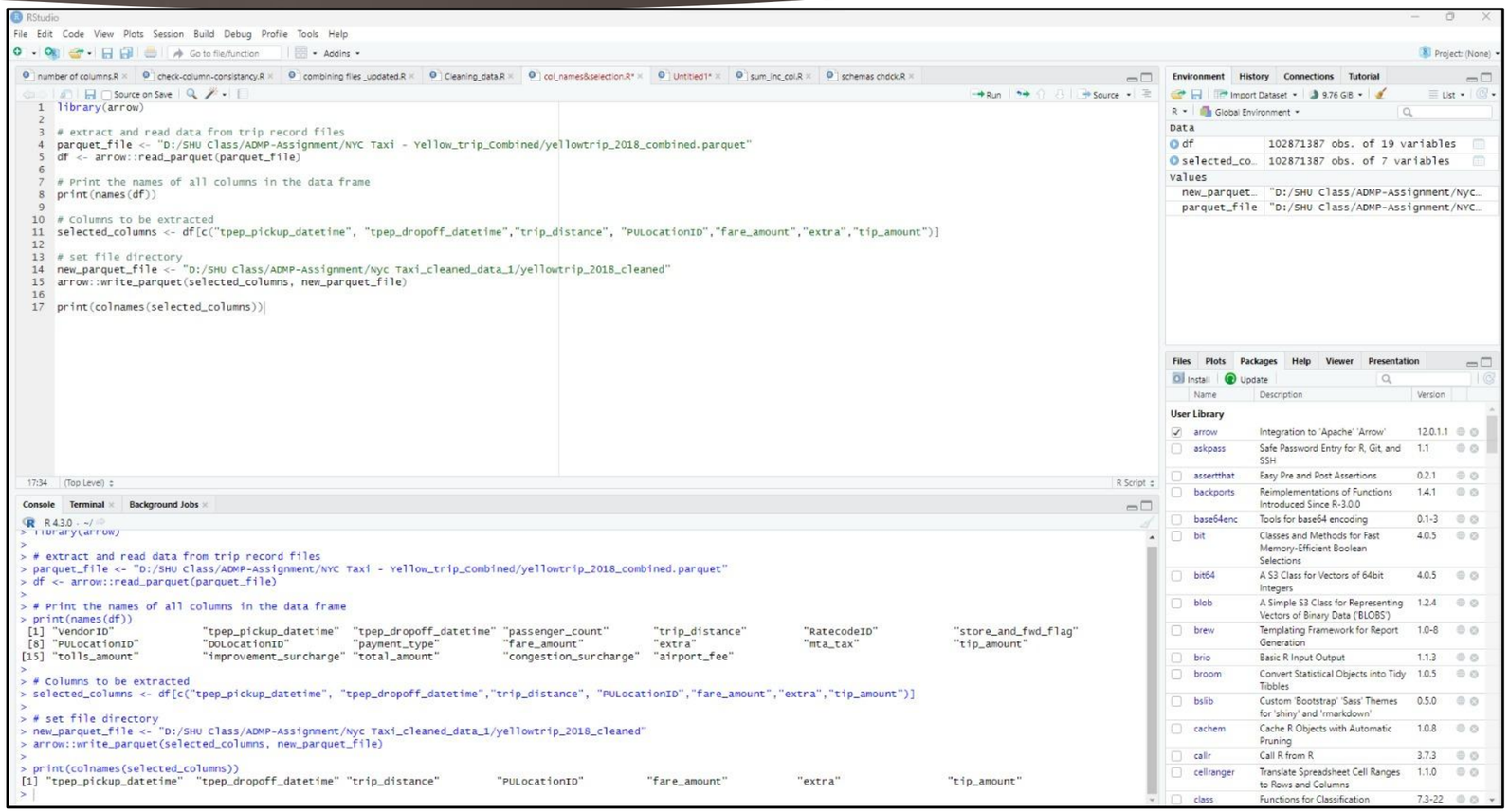


Figure 11: Screenshot of R code describing column selection from NYC trip records and Taxi Zone data set

Data Transformation

Derived Columns / Column Renaming

The following columns were derived

- **Income**:

  Income = fare amount + tip amount + extra

- **TimeID** = YYYYMMDD + DayID + SessionID
  - DayID is defined with an integer data type ranging from 1 -7 (Sun-Sat).
  - SessionID is defined as the time of the day in New York city divided into Morning, Afternoon, Evening and Night

  Morning: 12:00am to 12:00pm

  Afternoon: 12:00pm to 05:00pm

  Evening: 05:00pm to 08:00pm

  Night – 08:00pm to 12:00am

  Source: https://www.yourdictionary.com/articles/afternoon-evening-difference
- **PULocationID** column in the Yellow taxi trip records renamed to in the "LocationID". This will ensure the common column in both data set have same name.
- Column **Zone** in taxi zone dataset renamed to **Neighbourhood**

The derived columns will be helpful in providing answers to BQ4 and 5

```r
df$dateID <- as.integer(format(df$tpep_pickup_datetime, "%Y%m%d"))
df$date_month <- as.integer(format(df$tpep_pickup_datetime, "%Y%m"))
df$day_id <- as.integer(paste0(df$date_month, "", wday(df$dateID)))


# Create the 'time_id' column based on 'pickupdatetime'
df$time_id <- with(df, {
  hour_of_day <- hour(tpep_pickup_datetime)
  ifelse(hour_of_day >= 0 & hour_of_day < 12, day_id * 10 + 1,
         ifelse(hour_of_day >= 12 & hour_of_day < 17, day_id * 10 + 2,
                ifelse(hour_of_day >= 17 & hour_of_day < 20, day_id * 10 + 3,
                       day_id * 10 + 4)))
})
```

Figure 12: Screenshot of R code describing column derivation

```
Console  Terminal  Background Jobs
R 4.3.0 · ~/
> # extract and read data from trip record files
> parquet_file <- "D:/SHU Class/ADMP-Assignment/Nyc_Taxi_cleaned_Data(Hive)/NYC_Taxi_Zones.parquet"
> df <- arrow::read_parquet(parquet_file)
>
> # Print the names of all columns in the data frame
> print(names(df))
[1] "OBJECTID"   "Shape_Leng" "the_geom"   "Shape_Area" "zone"       "LocationID" "borough"
>
> # Columns to be extracted
> selected_columns <- df[c("the_geom", "zone","LocationID", "borough")]
>
> # Rename the "zone" column to "neighbourhood"
> selected_columns$neighbourhood <- selected_columns$zone
> selected_columns$zone <- NULL  #  remove the old "zone" column
>
>
> # set file directory
> new_parquet_file <- "D:/SHU Class/ADMP-Assignment/Nyc_Taxi_cleaned_Data(Hive)/NYC_Taxi_Zones_cleaned.parquet"
> arrow::write_parquet(selected_columns, new_parquet_file)
>
> # extract and read data from trip record files
> cleaned_parquet_file <- "D:/SHU Class/ADMP-Assignment/Nyc_Taxi_cleaned_Data(Hive)/NYC_Taxi_Zones_cleaned.parquet"
> new_df <- arrow::read_parquet(cleaned_parquet_file)
>
> # Print the names of all columns in the data frame
> print(names(new_df))
[1] "the_geom"     "LocationID"   "borough"      "neighbourhood"
>
```

Figure 13: Screenshot of R code showing the dimension of cleaned taxi zone dataset after dropping non-essential columns

Data Transformation

Invalid Data Type / Negative values

- It was observed that there are some **negative** values in the fare_trip column. All negative values were dropped to avoid biases in the analysis. These were dropped from the data set
- We have instances of single trips that span more than a week. This type of data will introduce unfair bias to the analysis, hence, was dropped. We are focused on trips not more than 24 hrs.
- We also have an instance of a pick-up date time higher than the drop-off date time.
- There are also instances of invalid date records. For instance, seeing a date of 2008 in the 2017 file. This was taken as an imputation error and was dropped.

```
51    df$tpep_pickup_datetime <- as.POSIXct(df$tpep_pickup_datetime)
52    df$tpep_dropoff_datetime <- as.POSIXct(df$tpep_dropoff_datetime)
53    #filter taxi rides beyond 24 hours
54    df <- df %>%
55      filter(abs(difftime(tpep_pickup_datetime, tpep_dropoff_datetime, units = "hours")) <= 24)
56
57    df <- df %>%
58      filter(year(tpep_pickup_datetime) == year, month(tpep_pickup_datetime) == x)
59     df <- df %>%
60      mutate(income = fare_amount + extra + tip_amount)
61
62    df$tpep_pickup_datetime <- as.POSIXct(df$tpep_pickup_datetime, format = "%Y-%m-%d %H:%M:%S")
63    # Create the 'dateID' column in the format 'yyyymmdd'
64    df$dateID <- as.integer(format(df$tpep_pickup_datetime, "%Y%m%d"))
65
```

Figure 14: Screen shot of R code use for dropping invalid records

tpep_pickup_datetime	tpep_dropoff_datetime
<dttm>	<dttm>
2009-01-01 00:19:39	2009-01-01 01:14:02
2008-12-31 23:34:04	2008-12-31 23:39:41
2009-01-01 00:22:33	2009-01-01 00:43:50
2009-01-01 02:04:37	2009-01-01 02:38:12

Figure 15: Invalid date record

tpep_dropoff_datetime	trip_distance
<dttm>	<dbl>
2018-01-30 11:42:09	189484.

Figure 16: Invalid distance

VendorID	tpep_pickup_datetime	tpep_dropoff_datetime
<int>	<dttm>	<dttm>
1	2013-04-01 13:53:05	2013-04-15 14:01:04
1	2013-04-04 20:34:27	2013-04-25 21:07:36
1	2013-04-09 14:49:49	2013-04-16 15:20:21

Figure 17: Invalid trip duration

DOLocationID	payment_type	fare_amount
Min. : 1.0	Min. :1.000	Min. : -350.0
1st Qu.:107.0	1st Qu.:1.000	1st Qu.: 6.5
Median :162.0	Median :1.000	Median : 9.0
Mean :161.8	Mean :1.338	Mean : 12.4
3rd Qu.:234.0	3rd Qu.:2.000	3rd Qu.: 13.5

Figure 18: Negative fare amount

trip_distance	extra	tip_amount	total_amount
<dbl>	<dbl>	<dbl>	<dbl>
1.4	1.5	1	10.5
7.7	1.5	0	25.5
5.8	2	1	22
18	12.5	0	100

Figure 19: Invalid extra charges

Data Transformation

Missing / Null values

- There is no evidence of missing or null values in the trip data records and the taxi zone records

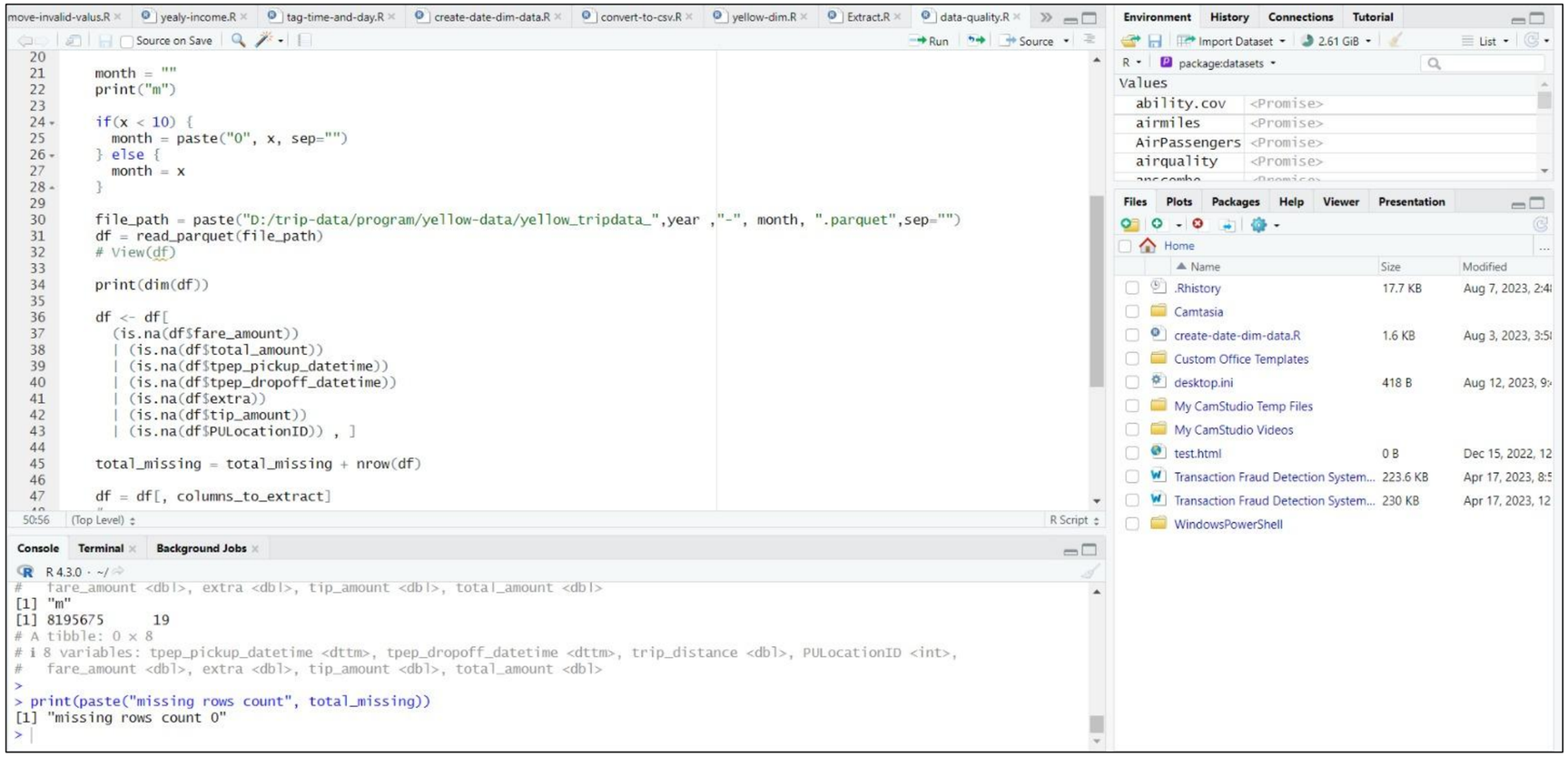


Figure 20: Screenshot of R code describing missing value check for NYC trip records and Taxi Zone data set

Data Transformation Summary

Dataset 1

Before Cleaning Yellow Taxi Trip Records

NYC Yellow Taxi Yearly Data Summary

Year	No of Rows	No of Column	Total number of observations	Dropped Rows after cleaning steps
2013	171,816,340	19	3,264,510,441	26,276,659.00
2014	165,447,579	19	3,143,503,982	24,893,627.00
2015	146,039,231	19	2,774,745,370	22,118,906.00
2016	131,131,805	19	2,491,504,276	21,116,719.00
2017	113,500,327	19	2,156,506,194	19,001,734.00
2018	102,871,387	19	1,954,556,334	17,900,567.00
2019	84,598,444	19	1,607,370,417	22,919,988.00
2020	24,649,092	19	468,332,729	6,635,006.00
2021	30,904,308	19	587,181,833	9,021,136.00
2022	39,656,098	19	753,465,843	11,243,116.00
Total	**1,010,614,611**	**19**	**19,201,677,419**	181,127,458.00

After Cleaning Yellow Taxi Trip Records

After Cleaning
NYC Yellow Taxi Yearly Data Summary

Year	No of Rows	No of Column	Total number of observations
2033	145,539,681.00	5	727,698,400
2014	140,553,952.00	5	702,769,755
2015	123,920,325.00	5	619,601,620
2016	110,015,086.00	5	550,075,425
2017	94,498,593.00	5	472,492,960
2018	84,970,820.00	5	424,854,095
2019	61,678,456.00	5	308,392,275
2020	18,014,086.00	5	90,070,425
2021	21,883,172.00	5	109,415,855
2022	28,412,982.00	5	142,064,905
Total	**829487153**	**5**	**4,147,435,715**

Table 6: Summary of NYC Yellow Taxi trip records data description before and after all cleaning process stages

Dataset 2

Before Cleaning Taxi Zone Dataset

Taxi Zone Dataset Summary

No of rows	263
No of Colums	7
Total Observations	1834

Dropped observations after cleaning steps

786

After Cleaning Taxi Zone Dataset

Taxi Zone Dataset Summary

No of rows	263
No of Colums	4
Total Observations	1048

Table 7: Summary of NYC Taxi Zone Neighbourhoods data description before and after all cleaning process stages

- 181,127,458 rows were dropped after performing all cleaning and transformation.
- 3 columns (fare amount, tip, and extra amount) from the 7 selected columns from the taxi trip datasets were dropped after deriving the income column. The cleaned dataset is left with 5 columns
- 3 unessential columns (ObjectID, shape length and shape area were dropped leaving only 4 columns in the cleaned taxi zone dataset

Data Load

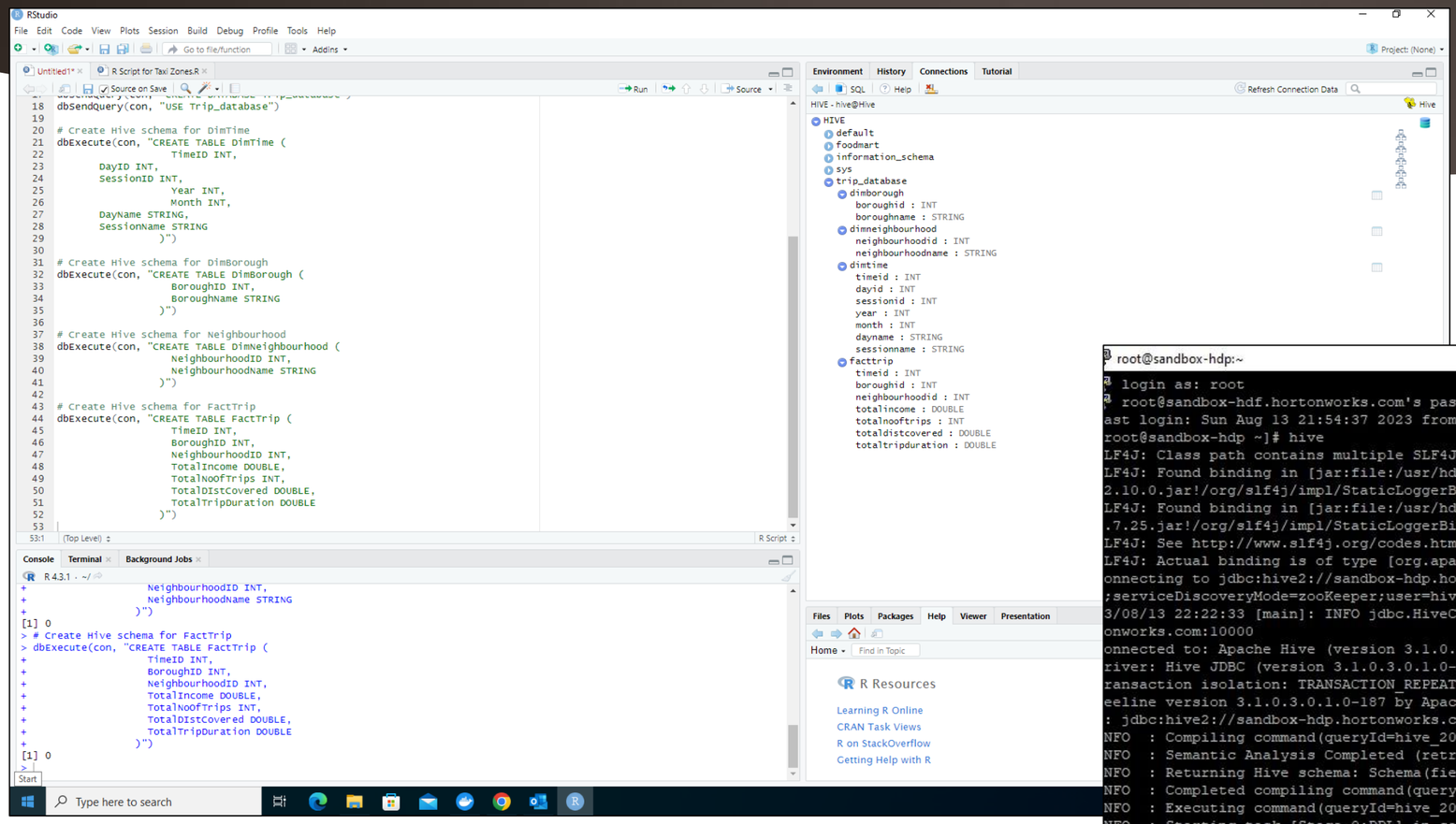


Fig.22a: Screenshot showing creation of creation of Hive Schema in R

Fig.22b: Shows database in Hive

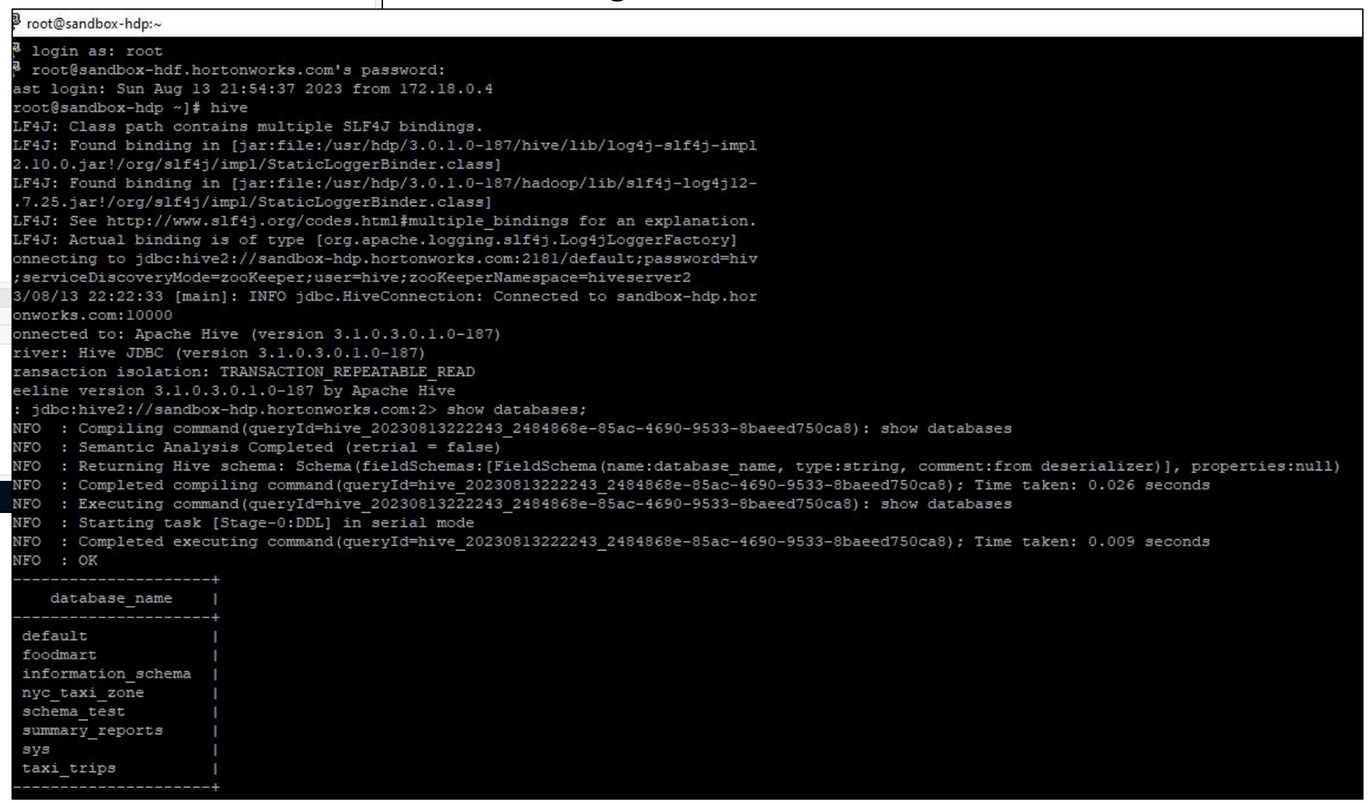
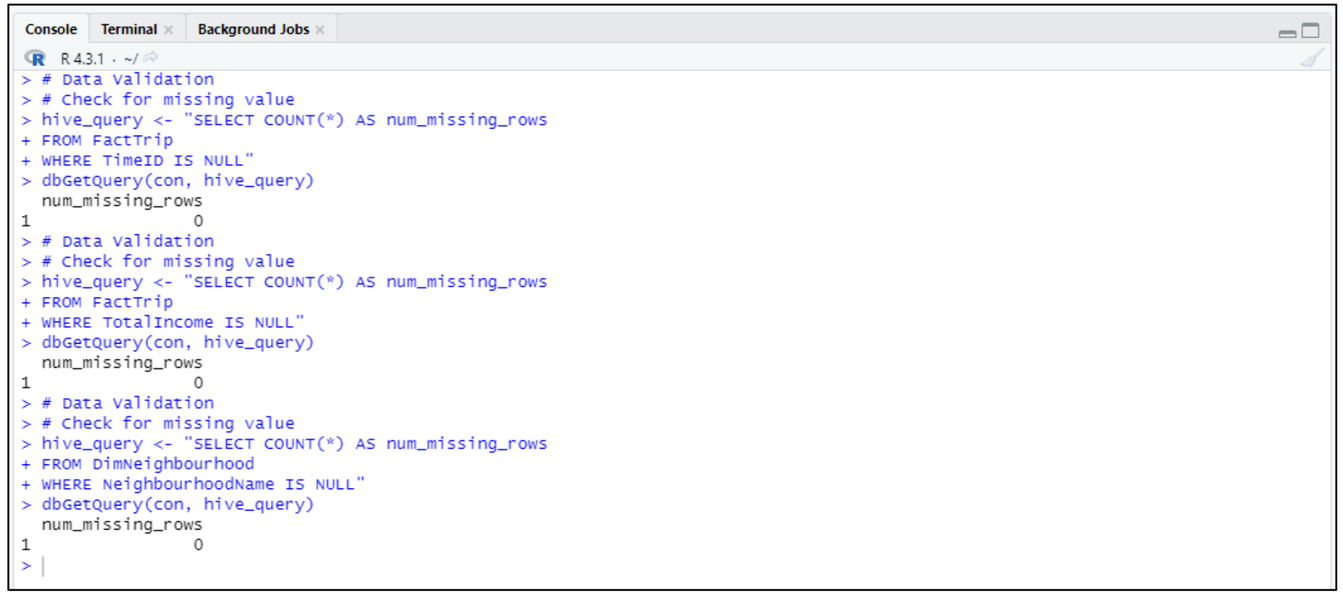


Fig.22c: Screenshot showing validity check after creation of schema



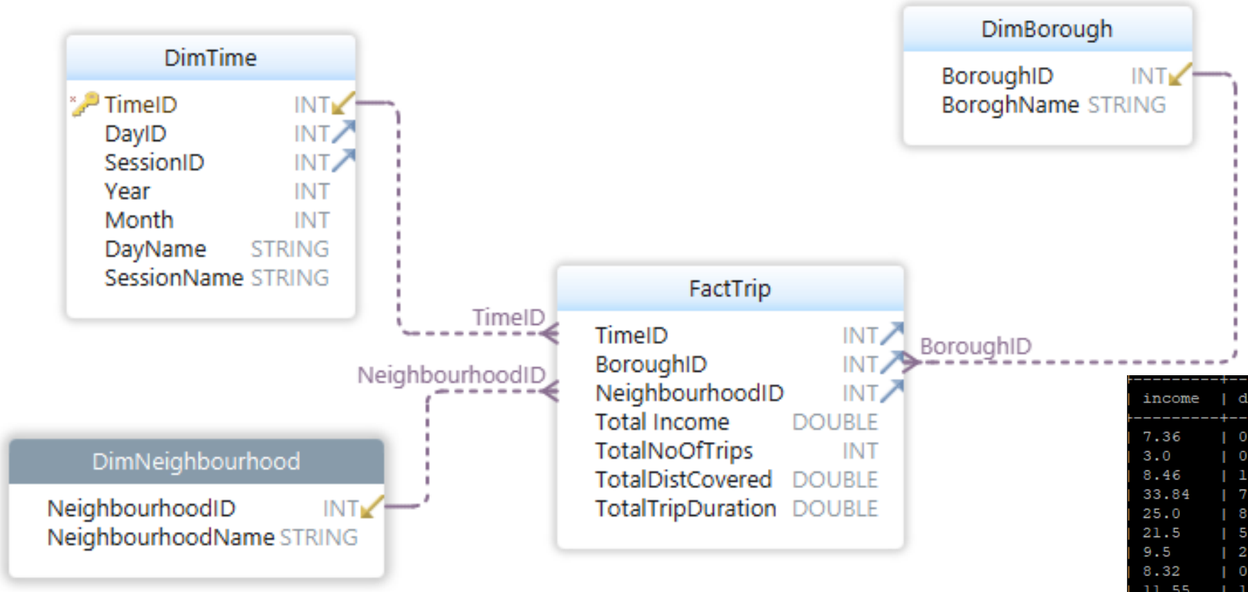
Data Validity Check

Data Validation

- No Missing values
- No duplicate values
- Data types confirmed

Data Warehouse Schema

Fig.23: Dataware Schema




```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> describe dimtime;
INFO  : Compiling command(queryId=hive_20230814040606_38299d5b-2d61-4910-84e8-ddd8df09587b): describe dimtime
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_ty
  type:string, comment:from deserializer), FieldSchema(name:comment, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230814040606_38299d5b-2d61-4910-84e8-ddd8df09587b); Time taken: 0.053 seconds
INFO  : Executing command(queryId=hive_20230814040606_38299d5b-2d61-4910-84e8-ddd8df09587b): describe dimtime
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20230814040606_38299d5b-2d61-4910-84e8-ddd8df09587b); Time taken: 0.016 seconds
INFO  : OK
+--------------+------------+----------+
|   col_name   | data_type  | comment  |
+--------------+------------+----------+
| timeid       | int        |          |
| dayid        | int        |          |
| sessionid    | int        |          |
| year         | int        |          |
| month        | int        |          |
| dayname      | string     |          |
| sessionname  | string     |          |
+--------------+------------+----------+
7 rows selected (0.127 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

Fig.24a: Output from SQL Query in hive showing trip fact table columns

income	distance	duration	n.neighbourhoodname	b.boroughname	d.dayname	d.sessionname	d.month	d.year
7.36	0.8	4.85	Upper West Side North	Manhattan	Wednesday	Morning	1	2020
3.0	0.03	0.8833333	Astoria	Queens	Wednesday	Morning	1	2020
8.46	1.07	5.6166667	Central Park	Manhattan	Wednesday	Morning	1	2020
33.84	7.76	37.3333333	Lincoln Square West	Manhattan	Wednesday	Morning	1	2020
25.0	8.45	17.1333333	LaGuardia Airport	Queens	Wednesday	Morning	1	2020
21.5	5.73	26.7	West Chelsea/Hudson Yards	Manhattan	Wednesday	Morning	1	2020
9.5	2.12	9.6833333	Bloomingdale	Manhattan	Wednesday	Morning	1	2020
8.32	0.93	5.85	Upper East Side North	Manhattan	Wednesday	Morning	1	2020
11.55	1.5	10.8833333	Central Harlem North	Manhattan	Wednesday	Morning	1	2020
6.0	0.81	5.4666667	Laurelton	Manhattan	Wednesday	Morning	1	2020

Fig.24b: Output from SQL Query in hive showing the efficiency of data warehouse schema

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> SELECT t.total_income AS income, t.totaldistcovered as distance, t.totaltripduration as duration , n.neighbourhoodname, b.boroughname, d.dayname, d.sessionname, d.month, d.year FROM facttrip
 t JOIN dimneighbourhood n ON t.neighbourhoodid = n.neighbourhoodid JOIN dimborough b ON t.boroughid = b.boroughid JOIN dimtime d ON t.timeid = d.timeid limit 10;
INFO  : Compiling command(queryId=hive_20230814041044_240b3709-2de6-4c68-aad9-d7149b884442): SELECT t.total_income AS income, t.totaldistcovered as distance, t.totaltripduration as duration , n.neighbourhoodname, b.boroughname, d.dayname
, d.sessionname, d.month, d.year FROM facttrip t JOIN dimneighbourhood n ON t.neighbourhoodid = n.neighbourhoodid JOIN dimborough b ON t.boroughid = b.boroughid JOIN dimtime d ON t.timeid = d.timeid limit 10
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:income, type:double, comment:null), FieldSchema(name:distance, type:double, comment:null), FieldSchema(name:duration, type:double, comment:null), FieldSchema(name:n.nei
ghbourhoodname, type:string, comment:null), FieldSchema(name:b.boroughname, type:string, comment:null), FieldSchema(name:d.dayname, type:string, comment:null), FieldSchema(name:d.sessionname, type:string, comment:null), FieldSchema(name:
d.month, type:int, comment:null), FieldSchema(name:d.year, type:int, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230814041044_240b3709-2de6-4c68-aad9-d7149b884442); Time taken: 0.696 seconds
INFO  : Executing command(queryId=hive_20230814041044_240b3709-2de6-4c68-aad9-d7149b884442): SELECT t.total_income AS income, t.totaldistcovered as distance, t.totaltripduration as duration , n.neighbourhoodname, b.boroughname, d.dayname
, d.sessionname, d.month, d.year FROM facttrip t JOIN dimneighbourhood n ON t.neighbourhoodid = n.neighbourhoodid JOIN dimborough b ON t.boroughid = b.boroughid JOIN dimtime d ON t.timeid = d.timeid limit 10
INFO  : Query ID = hive_20230814041044_240b3709-2de6-4c68-aad9-d7149b884442
```

Fig.23: SQL Query in hive to display few rows from the fact trip table

Dart mart Modelling 1

BQ1
Provide a month-by-month overview of the total income from the NYC Yellow Taxi by borough from January 2013 – December 2022.

Dimension Tables
- DimTime
- DimBorough

Lowest Level of Granularity for each dimension
- DimTime - Month
- DimBorough – BoroughName

Metric(s)
- Total Income

Dimension Table Attributes
- DimTime – TimeID, Year, Month
- DimBorough – BoroughID, BoroughName

Fact Tables (keys and Metrics)
- TimeID
- BoroughID
- Total Income

Table 8: Simple Schema Details for Business Question 1

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> describe dimborough;
INFO  : Compiling command(queryId=hive_20230814044609_d342dfdb-fbba-4377-977c-e2b9f9276cfe): describe dimborough
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string, comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer), FieldSchema(name:comment, type:string, comment:
rom deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230814044609_d342dfdb-fbba-4377-977c-e2b9f9276cfe); Time taken: 0.071 seconds
INFO  : Executing command(queryId=hive_20230814044609_d342dfdb-fbba-4377-977c-e2b9f9276cfe): describe dimborough
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20230814044609_d342dfdb-fbba-4377-977c-e2b9f9276cfe); Time taken: 0.022 seconds
INFO  : OK
+-------------+------------+----------+
|  col_name   | data_type  | comment  |
+-------------+------------+----------+
| boroughid   | int        |          |
| boroughname | string     |          |
+-------------+------------+----------+
```

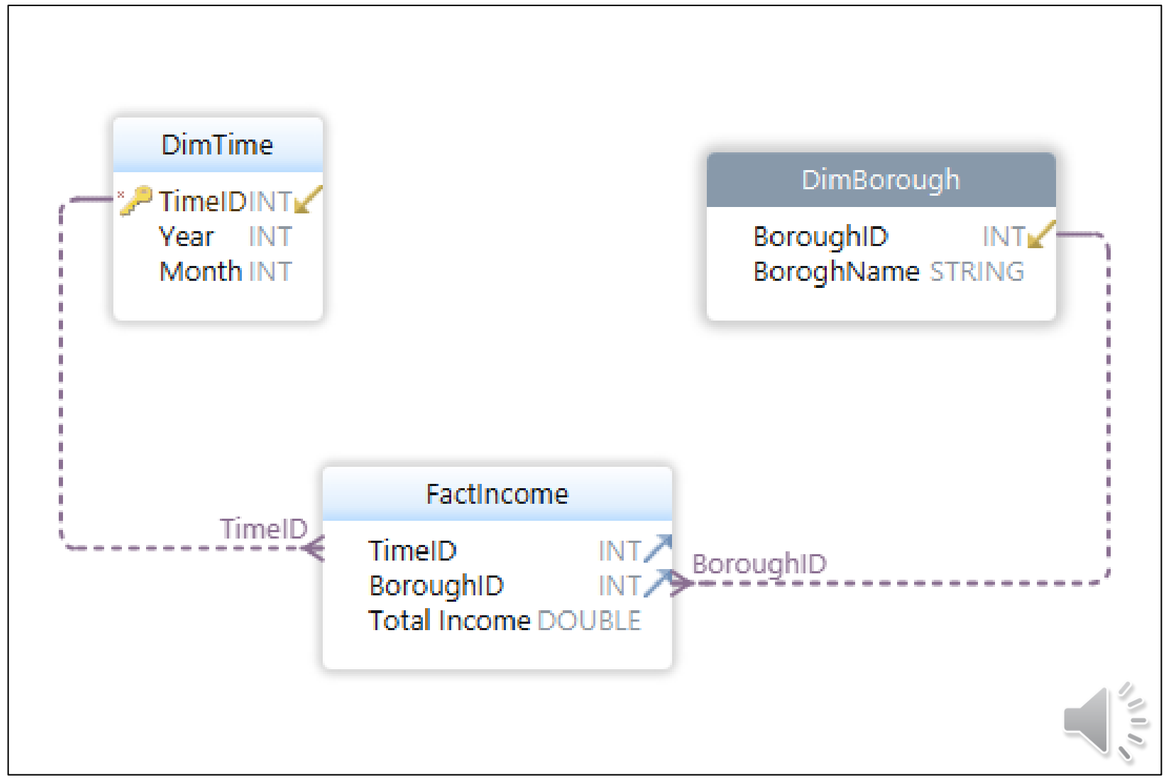


Figure 25: Screenshot of simple data schema for Business Question 1

Dart mart Modelling 2

BQ2 Provide an overview of the monthly total income by NYC Taxi Zone neighbourhood from January 2013 – December 2022.
Dimension Tables • DimTime • DimNeighbourhoods
Lowest Level of Granularity for each dimension • DimTime - Month • DimNeighbourhoods – NeighbourhoodName
Metric(s) • Total Income
Dimension Table Attributes • DimTime – TimeID, Year, Month • DimNeighbourhoods – NeighbourhoodID, NeighbourhoodName
Fact Tables (keys and Metrics) • TimeID • BoroughID • Total Income

Table 9: Simple Schema Details for Business Question 2

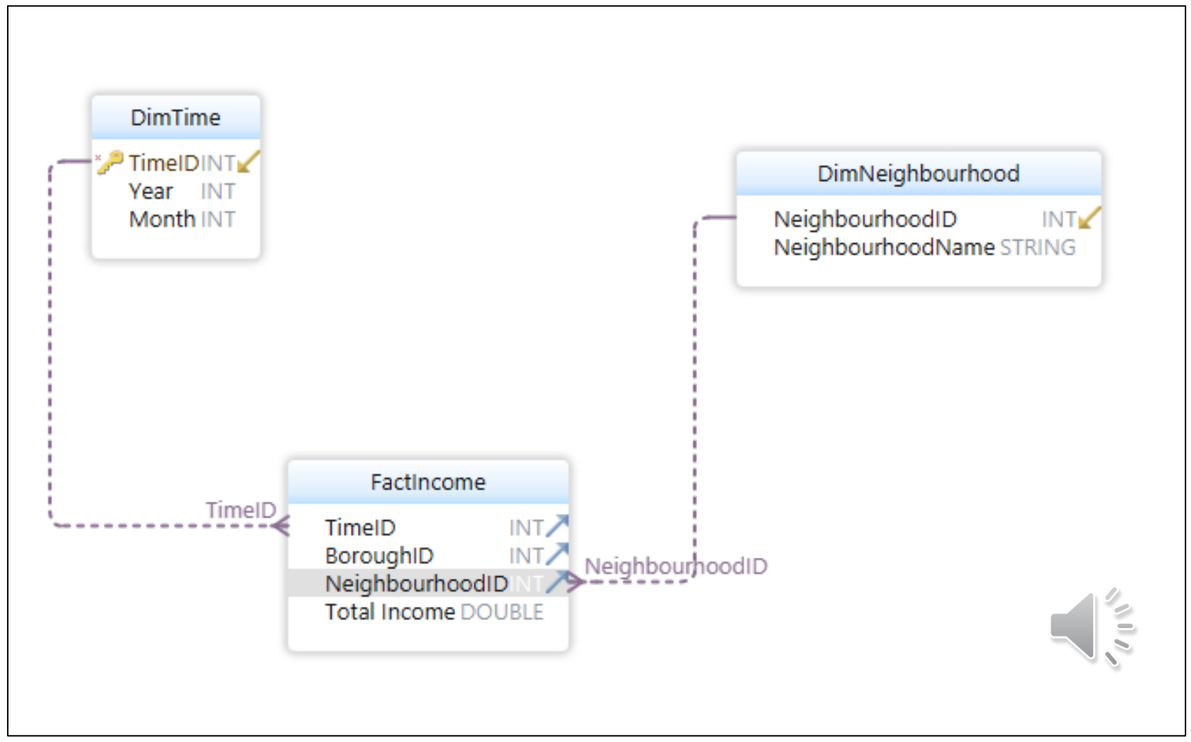


Figure 26: Screenshot of simple data schema for Business Question 2

Dart mart Modelling 3

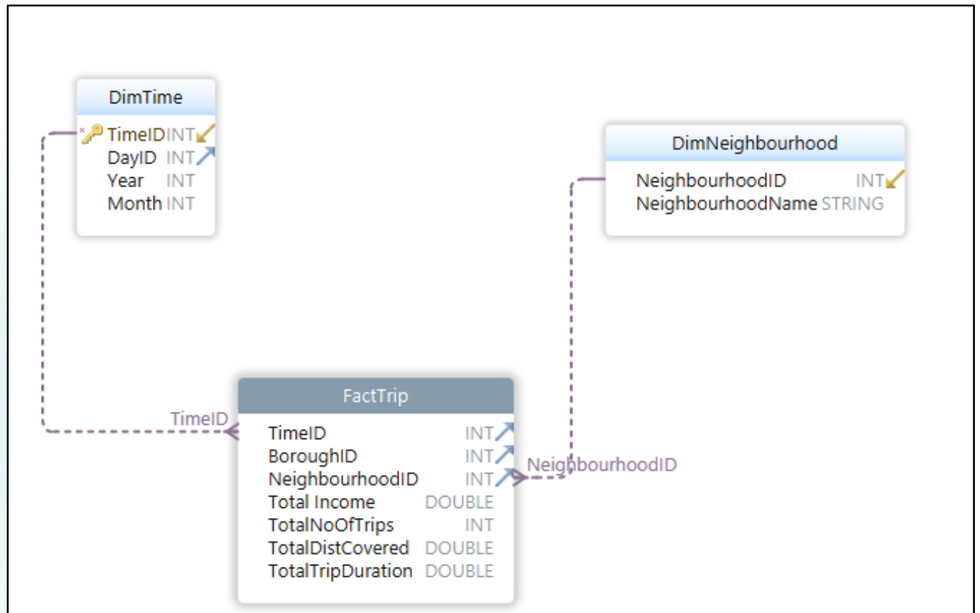


Figure 27: Screenshot of simple data schema for Business Question 3

BQ3 For the top 10 neighbourhoods with the highest income, provide a month-by-month comparison of the total number of trips, trip duration, and distances by TLC Yellow taxis by Taxi Zone neighbourhood from January 2013 - December 2022.
Dimension Tables • DimTime • DimBorough • DimNeighbourhoods
Lowest Level of Granularity for each dimension • DimTime - Month • NeighbourhoodID – Neighbourhood_Name • DimBorough – BoroughName
Metric(s) • Total number of trips • Total Distance Covered • Time Duration • Total Income
Dimension Table Attributes • DimTime – TimeID, Year, Month • NeighbourhoodID – NeighbourhoodID, NeighbourhoodName
Fact Tables (keys and Metrics) • TimeID • NeighbourhoodID • Total number of trips • Total Distance Covered • Trip Duration • Total Income

Table 10: Simple Schema Details for Business Question 3

Dart mart Modelling 4

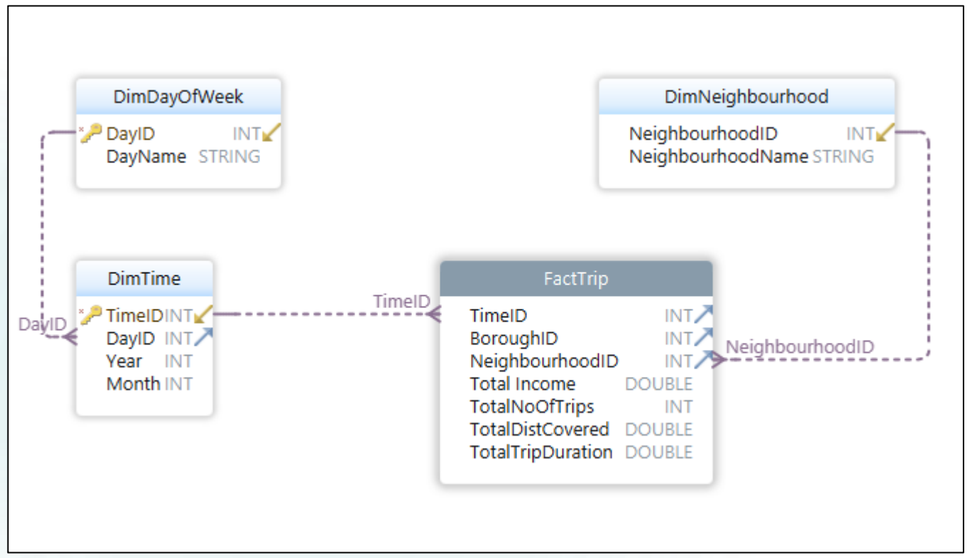


Figure 28: Screenshot of simple data schema for Business Question 4

BQ4
For the top 10 taxi zone neighbourhoods with the highest income, what are the peak days of the week (weekdays or weekends) with the highest total income on a monthly basis from January 2013 - December 2022?

Dimension Tables
- DimTime
- DimNeighbourhoods

Lowest Level of Granularity for each dimension
- DimTime – Month, Week
- DimNeighbourhoods – NeighbourhoodName
- DimDayofWeek – DayName

Metric(s)
- Total Income
- Shortest distance
- Total number of trips

Dimension Table Attributes
- DimTime – TimeID, Year, Month
- DimNeighbourhoods – NeighbourhoodID, NeighbourhoodName
- DimDayofWeek – DayID, DayName

Fact Tables (keys and Metrics)
- TimeID
- LocationID

- Total Income
- Total Distance Covered
- Total number of trips

Table 11: Simple Schema Details for Business Question 4

Dart mart Modelling 5

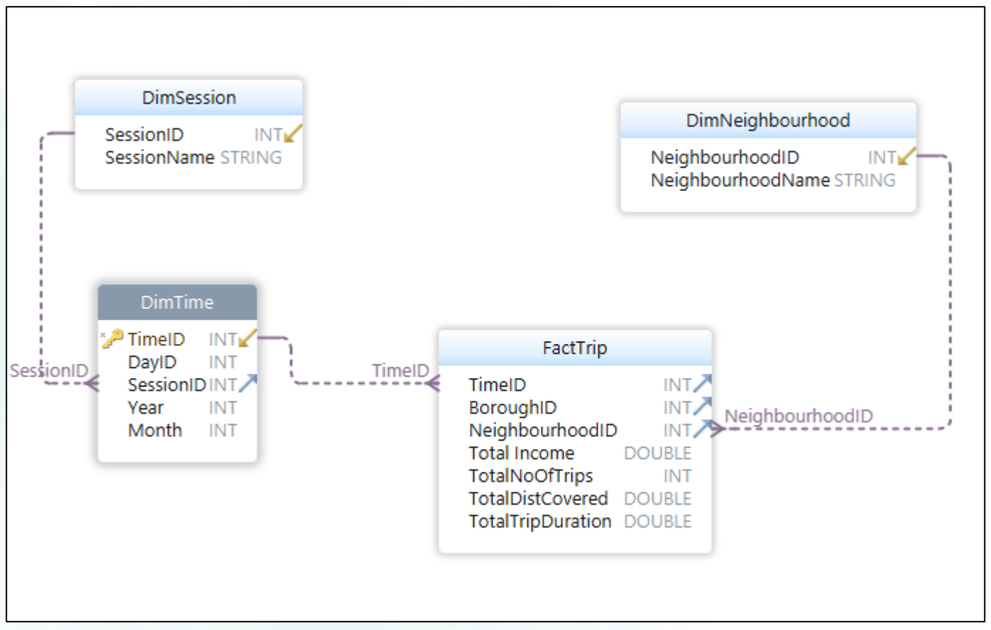
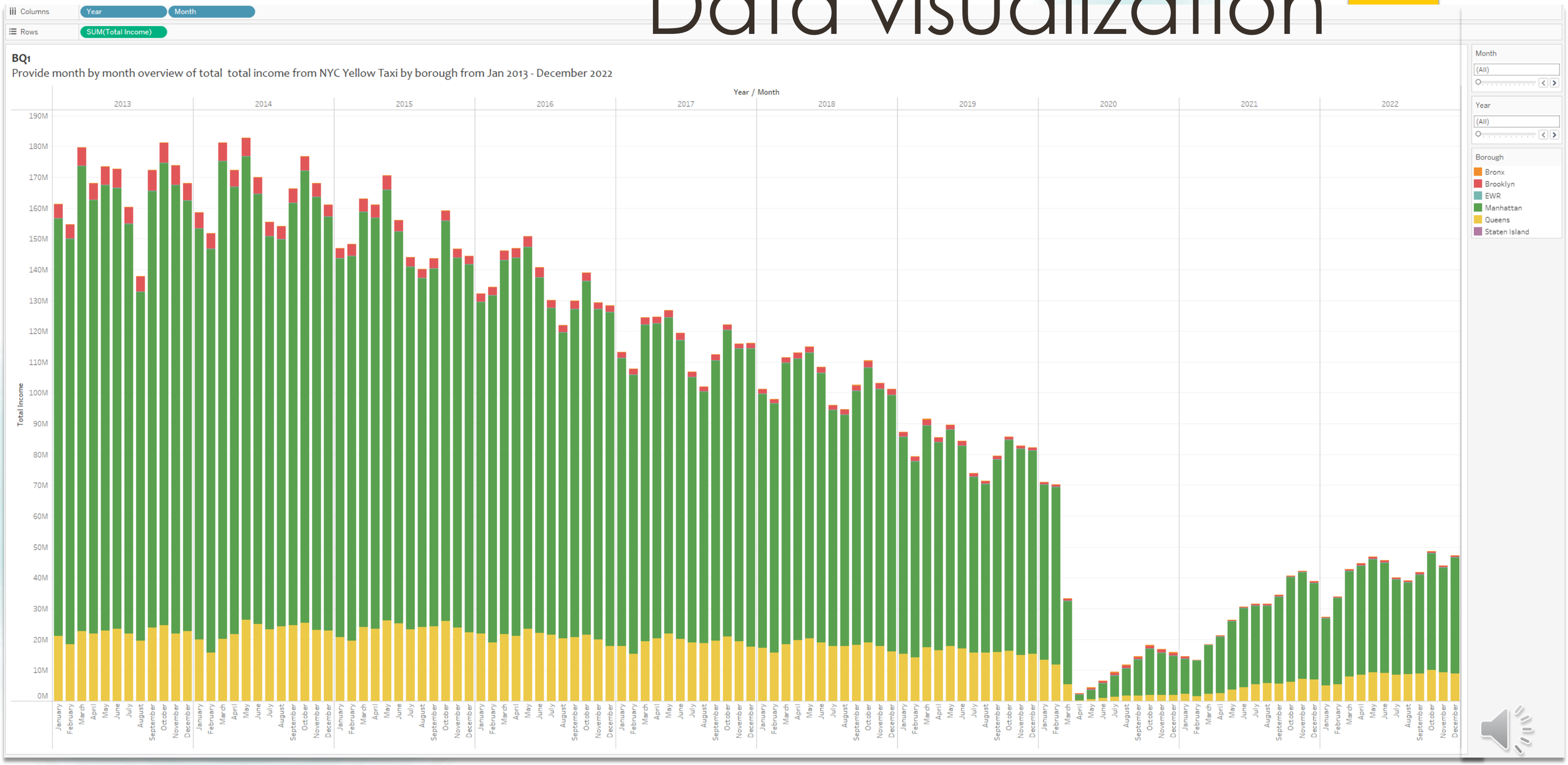


Figure 29: Screenshot of simple data schema for Business Question 5

BQ5 What time of the day (Morning, Afternoon, evening, and night) do we have the highest income from the top 10 Taxi Zone neighbourhoods monthly from Jan 2013 - December 2022?
Dimension Tables • DimTime • DimLocation • DimSeason
Lowest Level of Granularity for each dimension • DimTime – Time Session • DimLocation – NeighbourhoodName • DimSeason - Season
Metric(s) • Total Income
Dimension Table Attributes • DimTime – TimeID, Year, Month • DimLocation – LocationID, NeighbourhoodName • DimSeason – SeassonID, SeasonName
Fact Tables (keys and Metrics) • TimeID • ZoneID • SeasonID • Total Income

Table 12: Simple Schema Details for Business Question 5

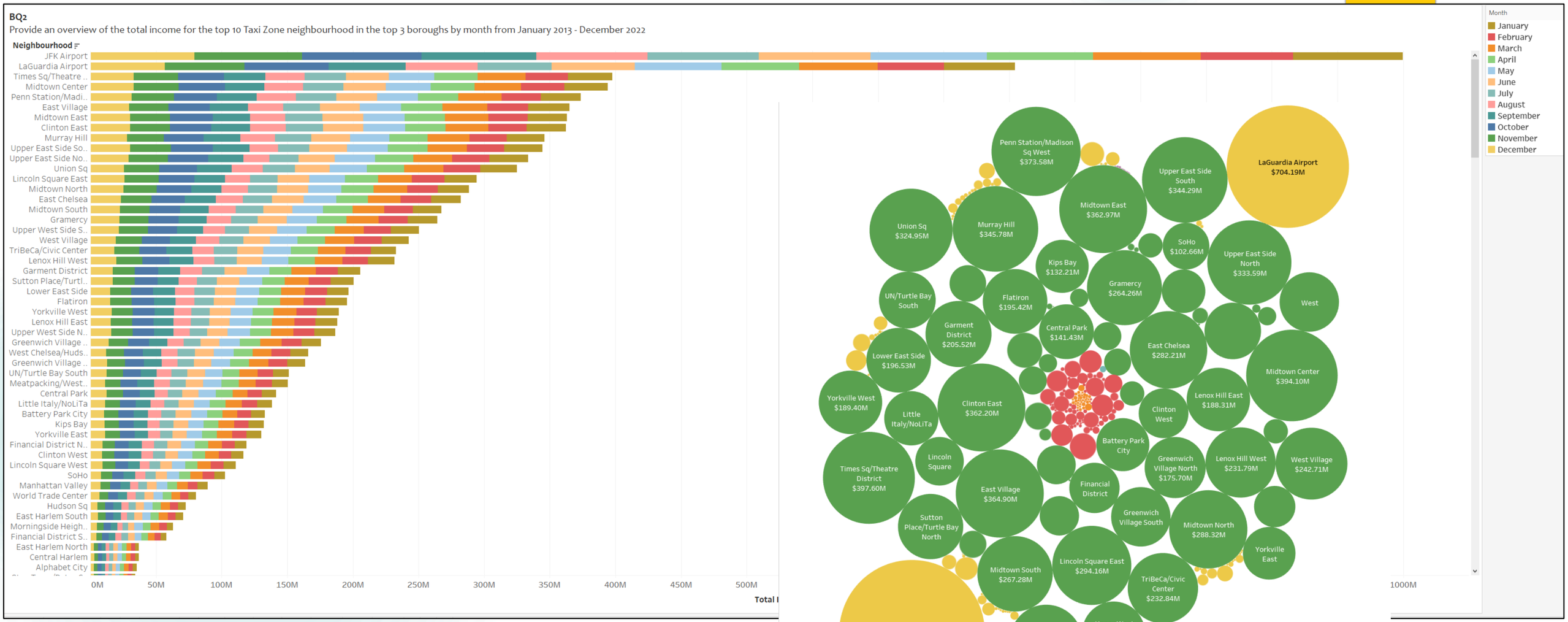
Data visualization



Most of the money was generated in Manhattan and Queens boroughs

Figure 30: Bar chart representing the distribution of income from Yellow taxi across the borough from 2013 to 2022

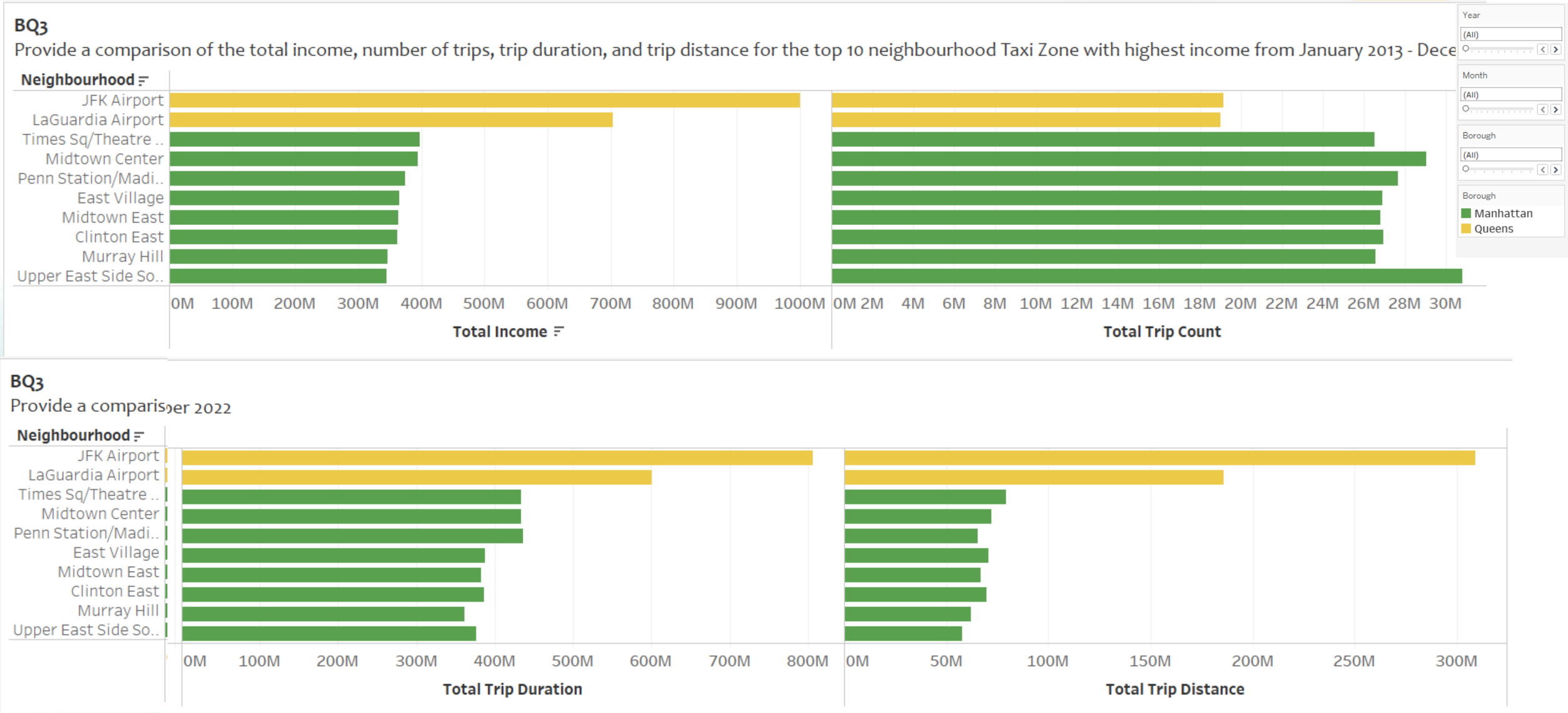
Data visualization



JKF airports, LaGuadia airport and Timesq/Theatre are the top 3 highest income neighbourhood

Figure 31: Bar chart and heat map representing the distribution of income from Yellow taxi across each neighbourhoods in from 2013 to 2022

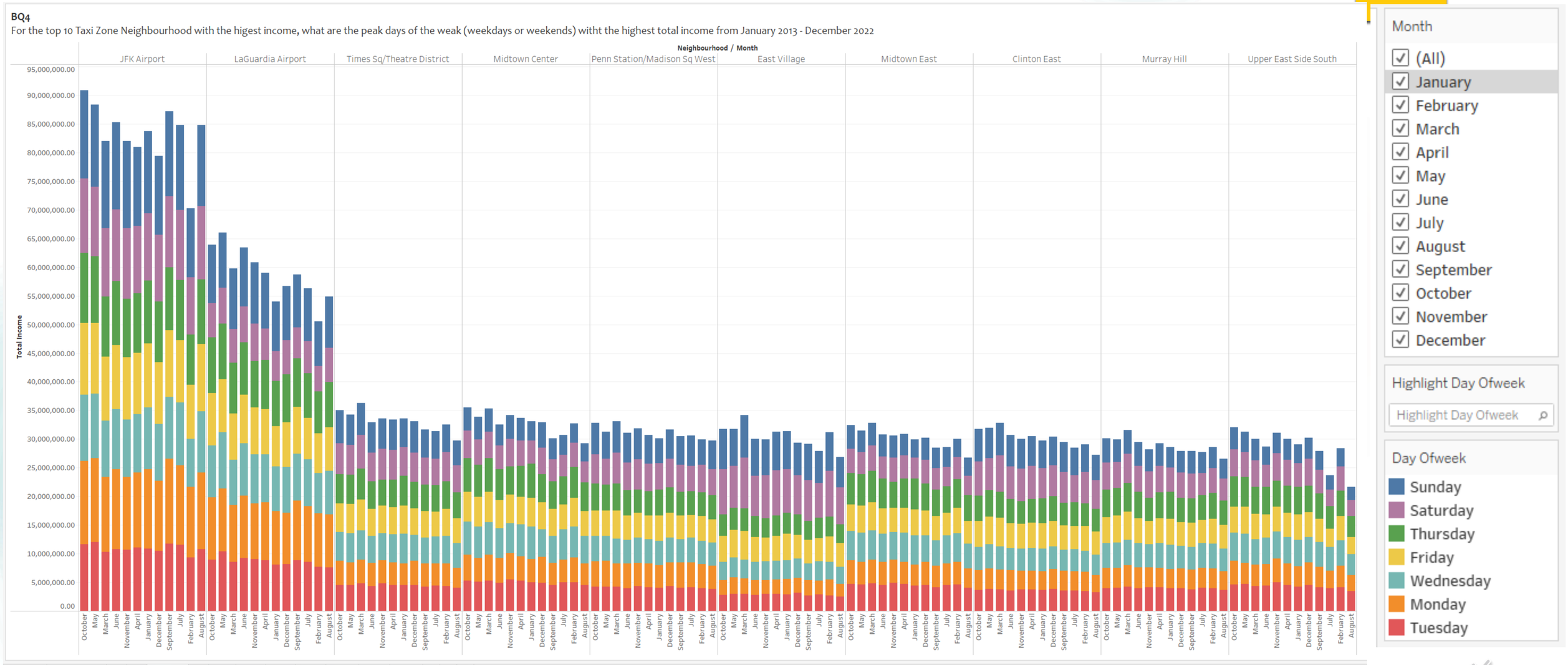
Data visualization



Area is a strong determinant of income as JKF airport with the lowest count generates more income than upper east side with the trip count.

Figure 32: Horizontal bar chart representing the distribution of income from Yellow taxi in the top 10 neighbourhood by Trip distance, income, count and income from 2013 to 2022

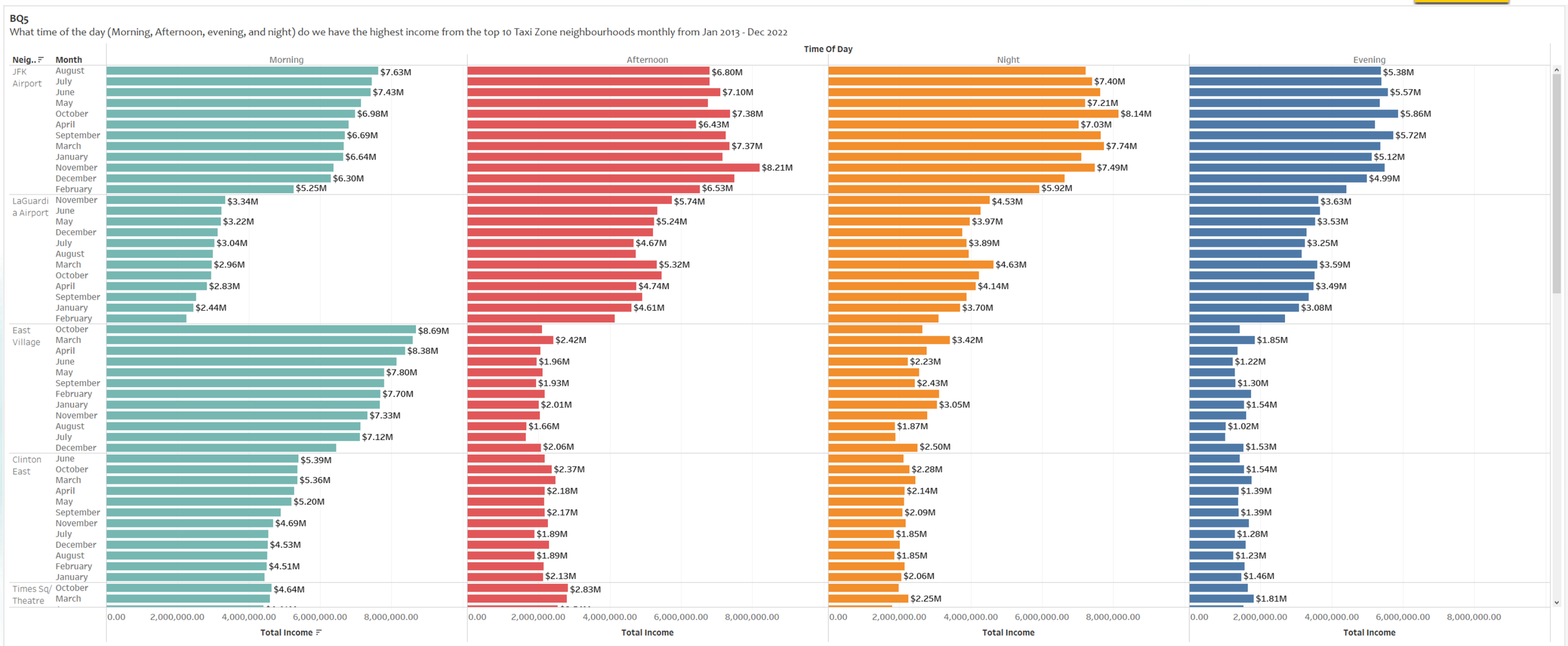
Data visualization



The highest income in the top ten boroughs is generated on Sundays and Saturdays. Thursdays and Fridays are the strongest generating income weekdays.

Figure 33: Vertical bar chart representing the distribution of income from Yellow taxi in the top 10 neighbourhood by days of the week

Data visualization



East village is the most profitable area in the morning time during weekends even thou it is ranked the 6th generating income neighborhood from the top 10.

Figure 34: Vertical bar chart representing the distribution of income from Yellow taxi in the top 10 neighbourhood by days of the week

Key Findings & Recommendation

Key Finding

- Significant decrease in income and no of trips in the year 2020 possibly due to the impact of covid pandemic
- Manthan and Queens boroughs are visible as the most profitable borough compared to other boroughs. This two borough contributes about 89% of the generated for the year under review
- Even though Manhattan generates the highest income, the 2 most profitable neighbourhoods are in Queens borough due to the presence of airports with 33% contribution to income from 263 neighbourhoods
- The number of trips does not necessarily mean higher income as evident in BQ3 where JKF airport has the lowest number of trips with the highest income compared to the upper east side south with the highest number of trips and yet lowest income.
- Even though the neighbourhood (East Village) that generates the highest income during weekends is not in the top 5, shows that days of the week are important during resource allocation and drivers' vacation management
- Also, most income during weekends is generated in the morning at East Village and more at JKF airports during other times of the day.

Recommendations

- Consider regular updating of the demographic data for the NYC neighbourhood as this can provide additional insight into the class of people and industry in the neighbourhood of each borough
- Regular monitoring and optimization of the developed ETL process to accommodate increasing data volume. Creating clusters and having replicas to handle node failures and prevent data looses
- Consider integrating real-time data feeds of the yellow taxi trip to enable real-time analytic insight and enhance quicker decisions.
- Regular updates and maintenance of the database system (Hadoop HDFs, Hive clusters) to ensure these resources are capable of handling growing datasets.
- Consider automated schedule monthly for validating trip records on website to prevent further capturing of invalid records especially those that involve store records in the file with the correct date.

Thank you

References

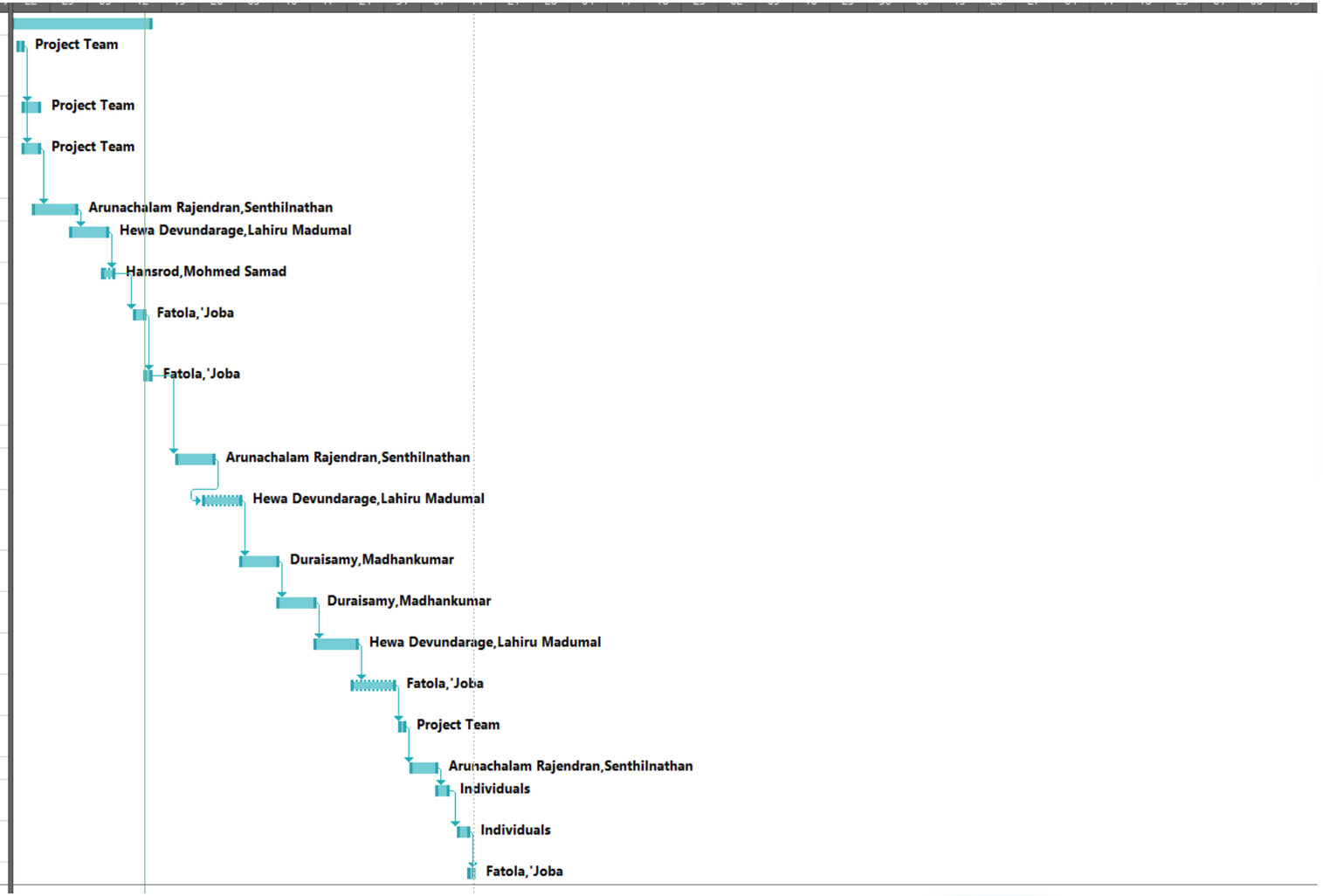
- Königsheim, C., Lukas, M., & Noeth, M. (2019). Should I Stop or Should I Go? New Evidence on the Labor Supply of Taxi Drivers in New York City. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3380113
- Wang, S., & Smart, M. (2020, April). The disruptive effect of ride sourcing services on for-hire vehicle drivers' income and employment. *Transport Policy*, *89*, 13–23. https://doi.org/10.1016/j.tranpol.2020.01.016
- Saia, A. (2021, July 8). Trouble Underground: Demand Shocks and the Labor Supply Behavior of New York City Taxi Drivers. *Italian Economic Journal*, *8*(1), 1–27. https://doi.org/10.1007/s40797-021-00162-3
- Dong, X., & Ryerson, M. S. (2020, November 26). Taxi Drops Off as Transit Grows amid Ride-Hailing's Impact on Airport Access in New York. *Transportation Research Record: Journal of the Transportation Research Board*, *2675*(2), 74–86. https://doi.org/10.1177/0361198120963116
- Saia, A. (2019). Trouble Underground: Demand Shocks and the Labor Supply Behavior of New York City Taxi Drivers. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3315200
- TLC Trip Record Data: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
- NYC Open Data: https://data.cityofnewyork.us/City-Government/2020-Neighborhood-Tabulation-Areas-NTAs-Tabular/9nt8-h7nd/data
- GeeksforGeeks. (2021). *Data Warehouse Architecture - GeeksforGeeks*. https://www.geeksforgeeks.org/data-warehouse-architecture/
- Apache Hadoop. (2019). Retrieved from Apache.org website: *https://hadoop.apache.org/*
- Apache SparkTM - Unified Analytics Engine for Big Data. (2019). Retrieved from Apache.org https://spark.apache.org/

References

- Business Intelligence Project Management. (n.d.). Retrieved from www.cmbi.com.au website: http://www.cmbi.com.au/6000_BIProjectManagement.html
- Betterteam. (2023). *Researcher Job Description.* https://www.betterteam.com/researcher-job-description
- Daivi. (2023). *Data Warehouse Engineer - A Complete Career Guide.* https://www.projectpro.io/article/data-warehouse-engineer/658#:~:text=A%20data%20warehouse%20engineer%20manages,purview%20of%20data%20warehouse%20engineers.
- Purdue University. (2023). *Project Manager Job Description.* https://www.purdue.edu/projectmanagementcertification/news/project-manager-job-description-career-outlook/
- Resource for Employer. (2023). *Data Engineer job description.* https://resources.workable.com/data-engineer-job-description#:~:text=Data%20engineers%20implement%20methods%20to,for%20predictive%20or%20prescriptive%20modeling.
- Simplilearn. (2023a). Business Analyst vs Data Analyst: Differences & Career Paths Explained . https://www.simplilearn.com/business-analyst-vs-data-analyst-article
- Simplilearn. (2023b). *Data Analyst Job Description: Responsibilities, Skills Required.* https://www.simplilearn.com/data-analyst-job-description-article
- Sumalatha, A., Vookanti, R., & Vannala, S. (2021). Study on Applications of SQL and Not only SQL Databases used for Big Data Analytics. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3817635
- Ji, Y., Chai, Y., Zhou, X., Ren, L., & Qin, Y. (2019, December 19). Smart Intra-query Fault Tolerance for Massive Parallel Processing Databases. *Data Science and Engineering*, *5*(1), 65–79. https://doi.org/10.1007/s41019-019-00114-z

Appendices

	Task Name	Duration	Start	Finish	Predecessors	Names
	PROPOSAL	20 days	Mon 22/05/2	Fri 16/06/23		
	Project team Meeting and Role Assingment	1 day	Tue 23/05/23	Tue 23/05/23		Project Team
	Planning and Project Scope Definition	3 days	Wed 24/05/23	Fri 26/05/23	3	Project Team
	Problem Identification/Literat Reviews	3 days	Wed 24/05/23	Fri 26/05/23	3	Project Team
	Data Gathering	6 days	Fri 26/05/23	Fri 02/06/23	5FS-1 day	Arunachala
	Research on DI and BDDS Tool	5 days	Fri 02/06/23	Thu 08/06/23	6FS-2 days	Hewa Devundarag
	Project proposal drafting	2 days	Thu 08/06/23	Fri 09/06/23	7	Hansrod, Mohmed
	Project Proposal Submission to Studiosity	2 days	Wed 14/06/23	Thu 15/06/23	8	Fatola, 'Joba
	Project Proposal Submission to Blackboard	1 day	Fri 16/06/23	Fri 16/06/23	9	Fatola, 'Joba
	IMPLEMENTATION					
	Data Identification and Definition	5 days	Thu 22/06/23	Wed 28/06/23	10	Arunachala Rajendran
	Data Acquisition, Filteration and Wrangling	5 days	Tue 27/06/23	Mon 03/07/23	12	Hewa Devundarag Lahiru
	Duraisamy, Madhankumar	5 days	Tue 04/07/23	Mon 10/07/23	13	Duraisamy, Madhankur
	Data Storage / Staging	5 days	Tue 11/07/23	Mon 17/07/23	14	Duraisamy, Madhankur
	Data Management Architecture	6 days	Tue 18/07/23	Tue 25/07/23	15	Hewa Devundarag
	Data Visualization and Story Telling	6 days	Tue 25/07/23	Tue 01/08/23	16	Fatola, 'Joba
	General Reflection Meeting	1 day	Thu 03/08/23	Thu 03/08/23	17	Project Team
	Slide Preparation	4 days	Sat 05/08/23	Wed 09/08/2	18	Arunachala
	Individual Written Assignment	2 days	Thu 10/08/23	Fri 11/08/23	19	Individuals
	Project Reflection (Individual)	2 days	Mon 14/08/23	Tue 15/08/23	20	Individuals
	Project Submission	1 day	Wed 16/08/2	Wed 16/08/2	21	Fatola,'Joba



Gaant Chart

Appendices

Project R Code

Scripts