Taxi Data Analytics

Team Members: Katrina Ying, Jin (Tina) Yu, Zhen (Zander) Gong

Team Member Introduction

Katrina Ying
MSIS ‘25
Co-leader

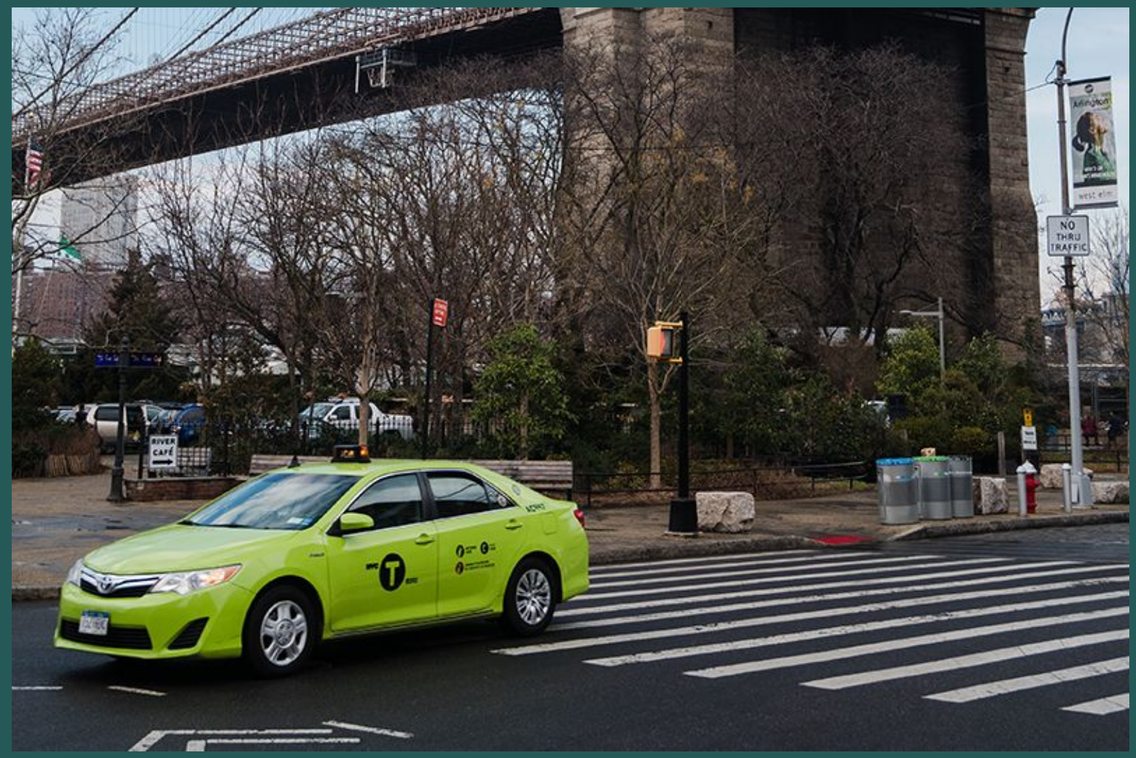
Jin(Tina) Yu
MSIS ‘25
Co-leader

Zhen(Zander) Gong
MSIS ‘25
Co-leader

Project Description & Scope

Topic: Revenue Tracking

Objective: Monitor revenue trends over time and by location to inform pricing strategies.



Abstract: This project conducts a multi-angle revenue analysis using green taxi trip records from January 2020 to October 2024

Final Goals

- Uncover temporal trends and operational patterns of New York City's green taxi fleet
- Assess changes in demand for service
- Identify peak usage periods
- Understand how external events affect traffic behavior over a five-year period

Software Used

- Python
- DuckDB
- Mysql
- Tableau
- Lucid chart

Data Description

- **New York Taxi Data - Green Taxi**
  - ➔ **Data Type** : Parquet format
  - ➔ **Features** :
    - Pick-up and drop-off dates/times & locations
    - Trip distances & payment types
    - Itemized fares
    - Rate types
- **taxi_zones**
  - ➔ **Data Type** : csv/shp format
  - ➔ **Features** :
    - mapping LocationID

ETL Process - Extraction

Data Source Identification & Access

- Identify file type: Parquet
- Identify data timeline: 2020-2024
- Access data from Trip Record Data source
- Key Information: pick-up and drop-off dates/times, trip distances, itemized fares, rate types, payment types, etc.

File Ingestion and Parsing

- Load parquet files into Python using pandas
- Read and parse files to extract key information

ETL Process - Transformations

Data Cleaning & Standardization

- Drop null values and duplicates

Normalization

- Rate types
- Payment types

Schema Mapping

- Identify fact & dimension tables
- Validate foreign key relationships
- Ensure referential integrity

Rate Types	
1	Standard
2	JFK
3	Newark
4	Nassau or Westchester
5	Negotiated Fare
6	Group Rides

Payment Types	
1	Credit Card
2	Cash
3	No Charge
4	Dispute
5	Unknown
6	Voided Trip

ETL Process - Loading

Load Table Data Into Database

- Create database
- Create tables based on star schema
- Insert transformed records into dimension tables
- Load fact table with reference to dimension tables
- Verify that foreign keys match existing dimension records

Incremental update of the Star Schema

- Identify the New Data
- Process the Data (Transformation)
- Load the Data

Database & Star Schema

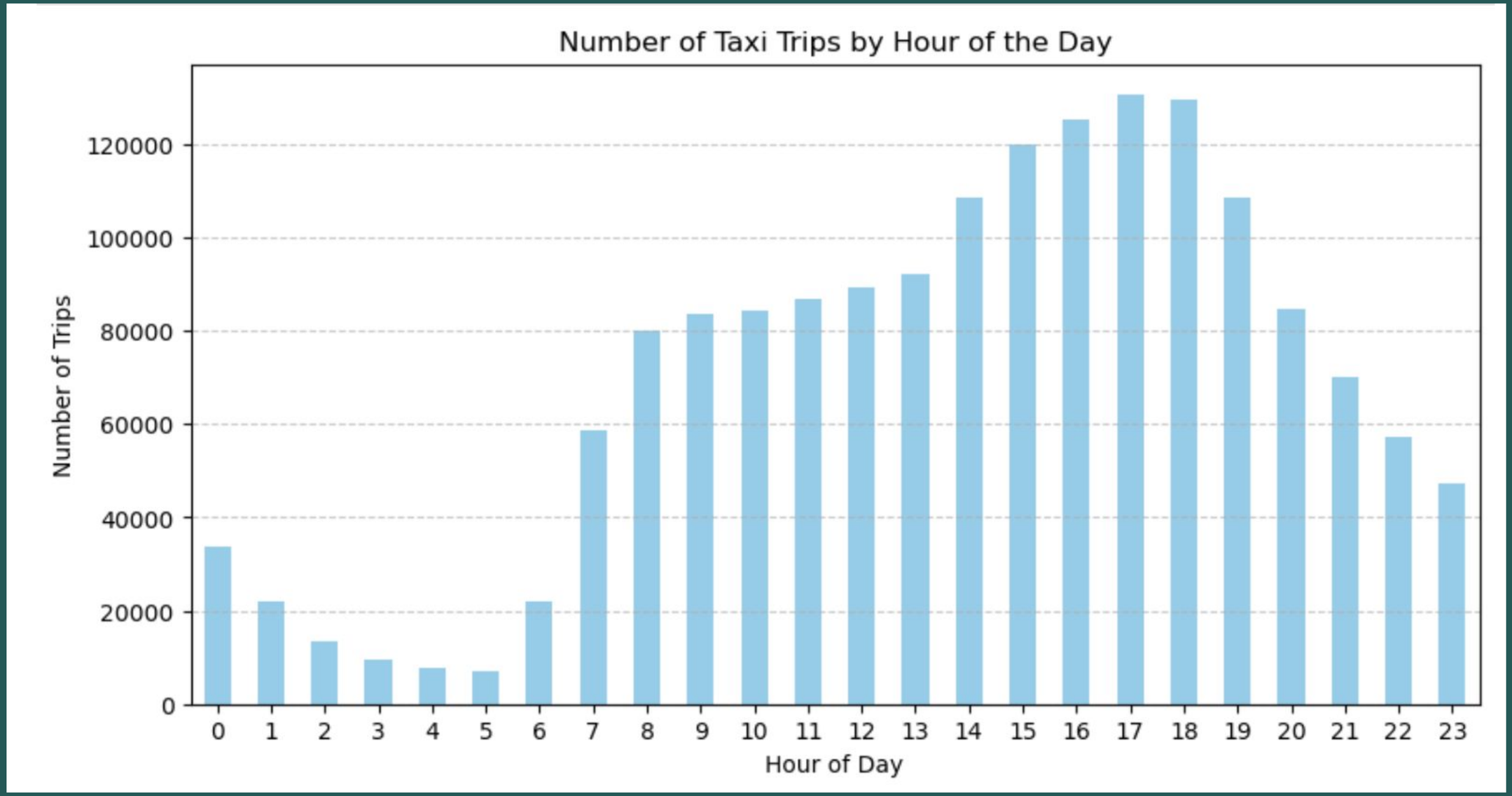
FACT TABLE

Column Name	Description
`trip_id`	Primary Key
`lpep_pickup_date_id`	FK to `dim_date`
`lpep_dropoff_date_id`	FK to `dim_date`
‘pulocationid’	
`dolocationid’	
`payment_type_id`	FK to `dim_payment_type`
`ratecode_id`	FK to `dim_ratecode`
`fare_amount`	Fare price ($)
`total_amount`	Total fare paid ($)
`trip_distance`	Distance in miles
`trip_duration`	Duration in minutes
`tip_amount`	
`extra`	
`fare_per_mile`	Calculated field

DIMENSION TABLES

dim_date

Column Name	Column Type
date_id	BIGINT
full_date	TIMESTAMP
year	INTEGER
month	INTEGER
day	INTEGER
day_of_week	VARCHAR
hour_of_day	INTEGER
am_pm	VARCHAR
is_peak_hour	BOOLEAN
minute	INTEGER
second	INTEGER


Number of Taxi Trips by Hour of the Day
Number of Trips
0
20000
40000
60000
80000
100000
120000
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day

Database & Star Schema cont.

DIMENSION TABLES CONTINUED

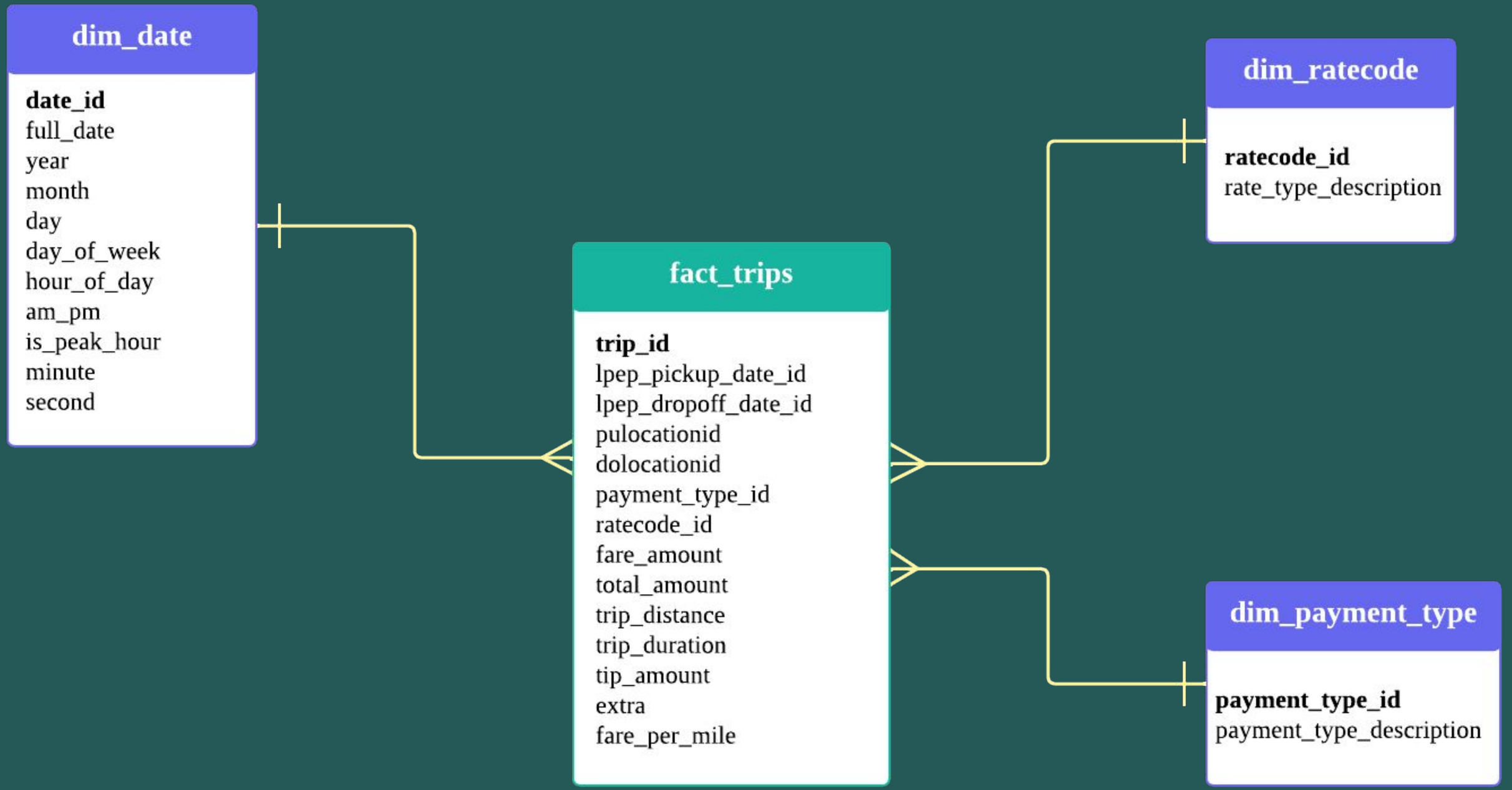
dim_payment_type

Column Name	Column Type
payment_type_id (PK)	INTEGER
payment_type_description	VARCHAR

dim_ratecode

Column Name	Column Type
ratecode_id (PK)	INTEGER
rate_type_description	VARCHAR

Star Schema Structure: 3 Dimension Tables, 1 Fact Table


dim_date
date_id
full_date
year
month
day
day_of_week
hour_of_day
am_pm
is_peak_hour
minute
second
fact_trips
trip_id
lpep_pickup_date_id
lpep_dropoff_date_id
pulocationid
dolocationid
payment_type_id
ratecode_id
fare_amount
total_amount
trip_distance
trip_duration
tip_amount
extra
fare_per_mile
dim_ratecode
ratecode_id
rate_type_description
dim_payment_type
payment_type_id
payment_type_description

Incremental update of DW

Monthly Incremental Update Process

- Schedule the Update (Monthly Basis)
- Incremental Data Identification

Scripts(Sample SQL queries)

- Extraction & Transformation
- Loading
- Scheduling: Create a scheduled job in DW system (or an external scheduler like cron or Airflow) to execute the stored procedure monthly.

```sql
INSERT INTO dim_time (trip_date, year, month, day)
SELECT DISTINCT CAST(pickup_datetime AS DATE) AS trip_date,
       EXTRACT(YEAR FROM pickup_datetime),
       EXTRACT(MONTH FROM pickup_datetime),
       EXTRACT(DAY FROM pickup_datetime)
FROM staging_green_taxi
WHERE pickup_datetime > (SELECT last_watermark
                         FROM etl_metadata
                         WHERE source = 'green_taxi')
ON DUPLICATE KEY UPDATE
    year  = VALUES(year),
    month = VALUES(month),
    day   = VALUES(day);

WITH NewGreenTaxiData AS (
    SELECT *
    FROM staging_green_taxi
    WHERE pickup_datetime > (
        SELECT last_watermark
        FROM etl_metadata
        WHERE source = 'green_taxi'
    )
)
SELECT DISTINCT CAST(pickup_datetime AS DATE) AS trip_date
FROM NewGreenTaxiData;

UPDATE etl_metadata
SET last_watermark = (SELECT MAX(pickup_datetime) FROM NewGreenTaxiData)
WHERE source = 'green_taxi';
```

Business Intelligence

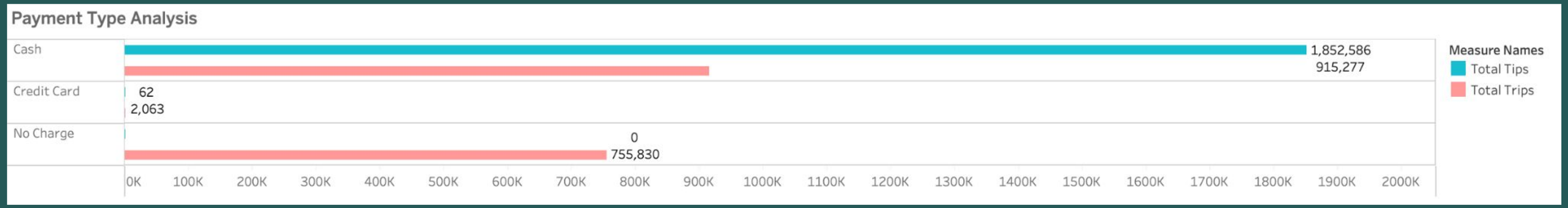
End users of DW : Business Analysts/Data Scientists/Management and Executive Teams/Finance and Accounting Teams

Experience improvement

- Promote Payment Methods with Higher Tips
- Optimize Trip Length-Based Marketing
- Enhance Driver Deployment Based on Peak Hours and Location
- Improve Customer Experience in High-Tip and High-Volume Locations
- Promote Off-Peak Travel with Incentives
- Optimize Extra Charges for Fair Pricing
- Implement Data-Driven Dynamic Pricing

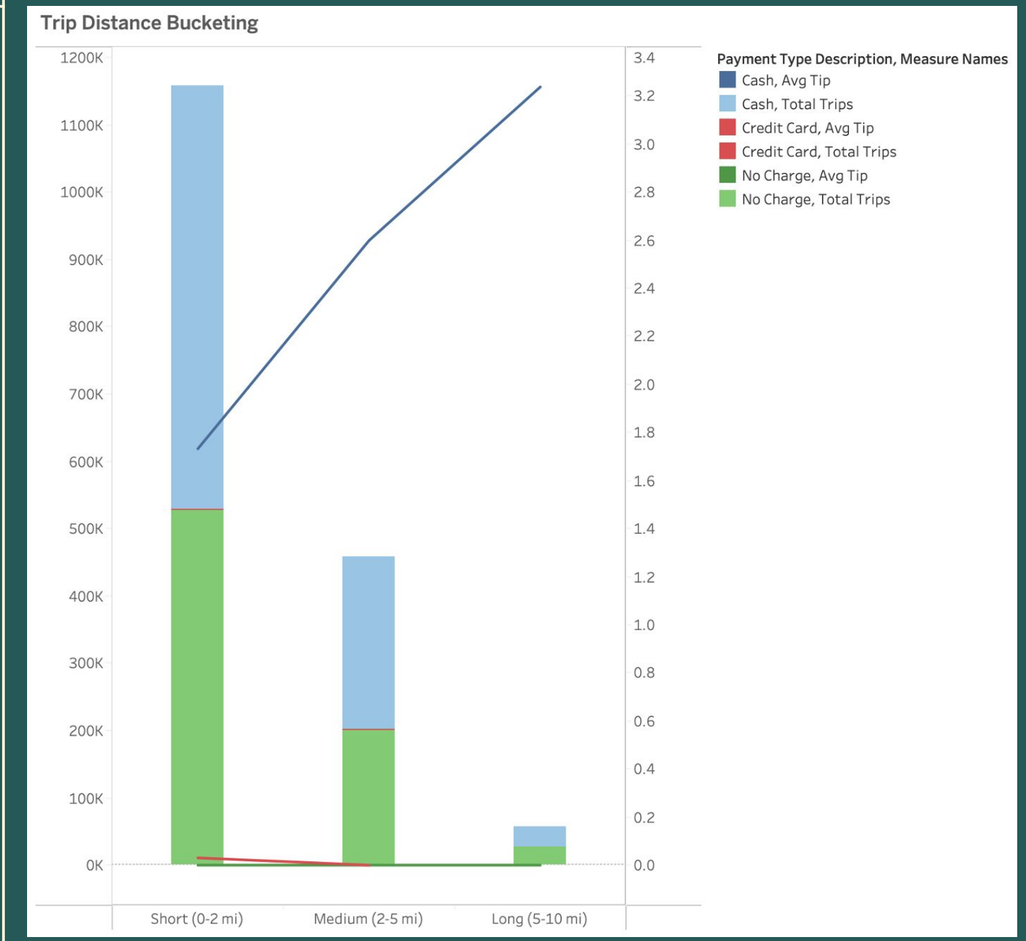
Query 1 - Payment Type Analysis

Query Description	OLAP Function	Query Code
Tip amount distribution by payment type	Roll Up	SELECT d.payment_type_description, COUNT(t.trip_id) AS total_trips, AVG(t.tip_amount) AS avg_tip, SUM(t.tip_amount) AS total_tips FROM fact_trips t JOIN dim_payment_type d ON t.payment_type_id = d.payment_type_id GROUP BY d.payment_type_description ORDER BY total_tips DESC;



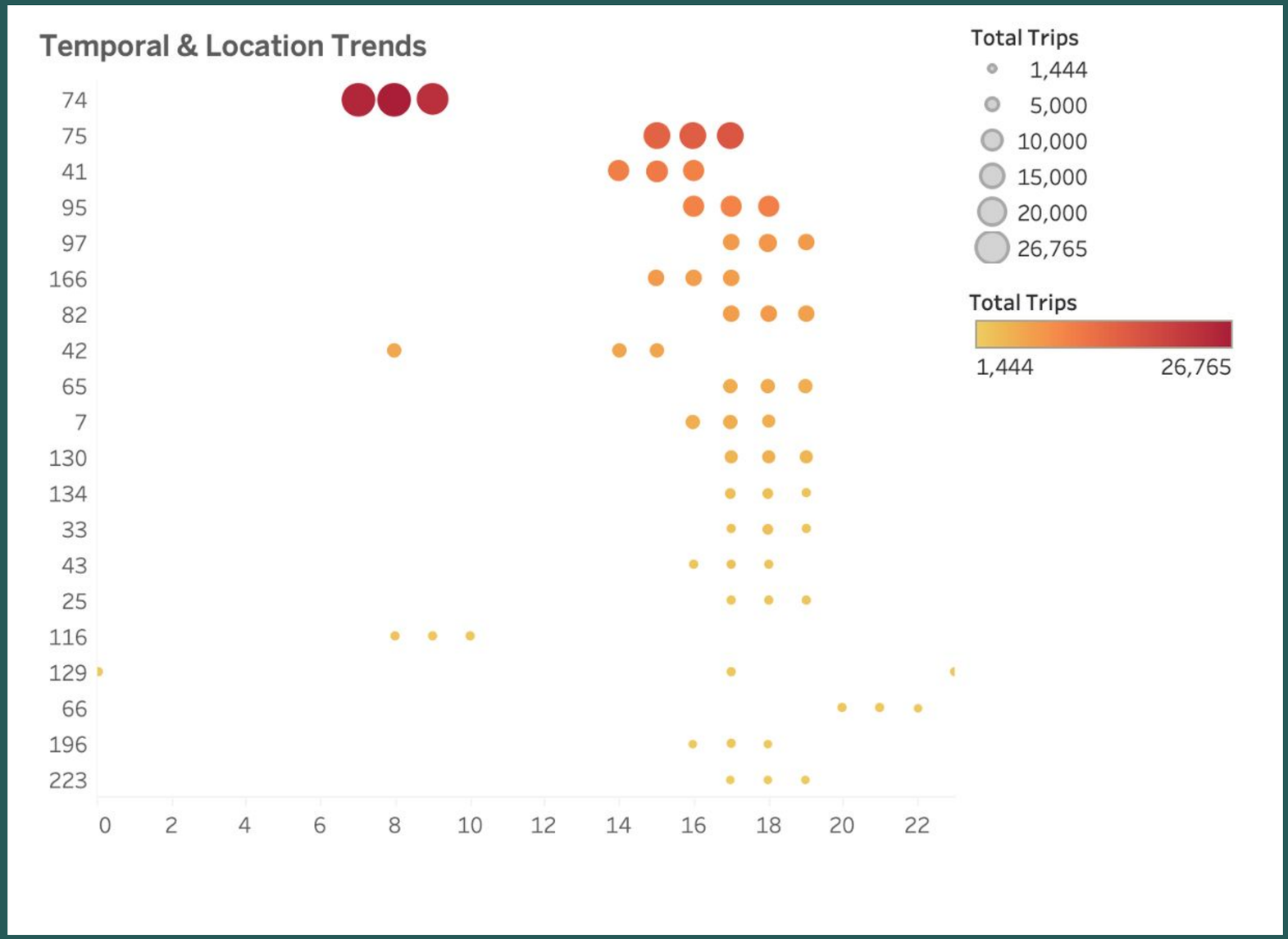
Query 2 - Trip Distance Bucketing

Query Description	OLAP Function	Query Code
Analyze tip amount distribution and influencing factors like ride distance	Dice	SELECT CASE WHEN t.trip_distance < 2 THEN 'Short (0-2 miles)' WHEN t.trip_distance BETWEEN 2 AND 5 THEN 'Medium (2-5 miles)' WHEN t.trip_distance BETWEEN 5 AND 10 THEN 'Long (5-10 miles)' ELSE 'Very Long' END AS distance_category, d.payment_type_description, AVG(t.tip_amount) AS avg_tip, COUNT(t.trip_id) AS total_trips FROM fact_trips t JOIN dim_payment_type d ON t.payment_type_id = d.payment_type_id GROUP BY distance_category, d.payment_type_description ORDER BY avg_tip DESC;



Query 3 - Temporal & Location Trends

Query Description	OLAP Function	Query Code
Identify passenger demand peaks by time and location	Roll Down	WITH RankedTrips AS (SELECT d.hour_of_day AS pickup_hour, f.pulocationid COUNT(f.trip_id) AS total_trips, RANK() OVER (PARTITION BY f.pulocationid ORDER BY COUNT(f.trip_id) DESC) AS rank_num FROM fact_trips f JOIN dim_date d ON f.lpep_pickup_date_id = d.date_id GROUP BY d.hour_of_day, f.pulocationid HAVING COUNT(f.trip_id) > 50 -- Filter out trips where total_trips <= 50) SELECT pickup_hour, pulocationid, total_trips FROM RankedTrips WHERE rank_num <= 3 ORDER BY pulocationid, rank_num;

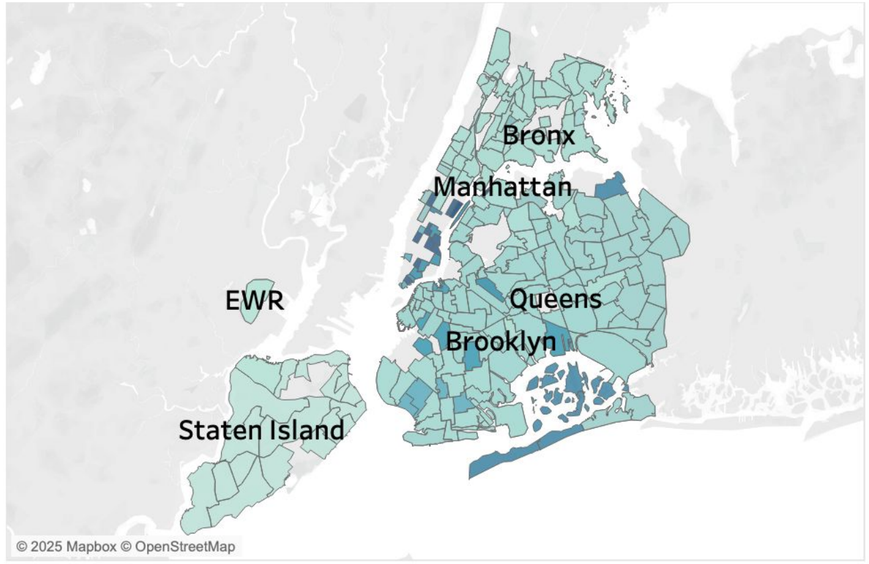


New Choropleth Pick-up L Drop-off Locations

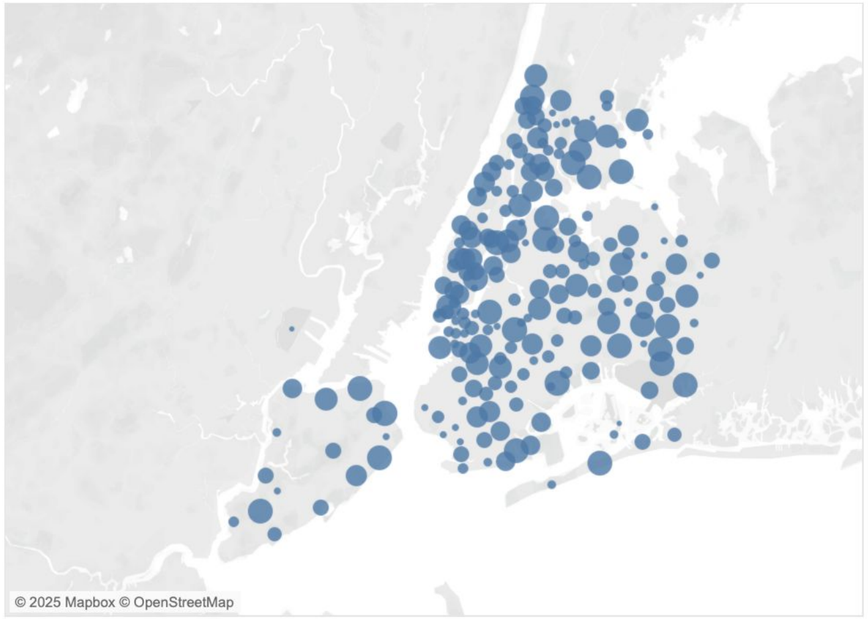
Map Analysis

New York City Boroughs

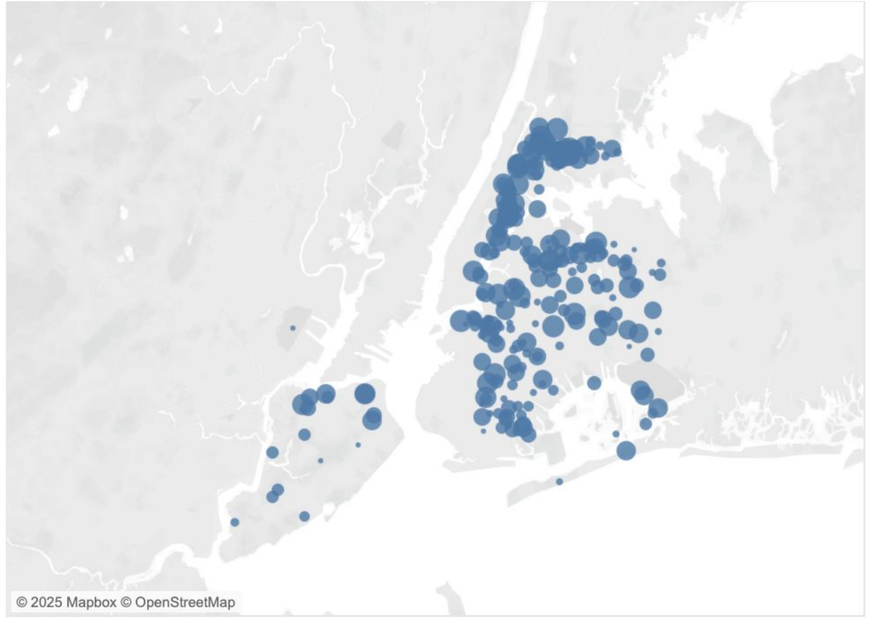
Choropleth of Fares by Boroughs



Pick-up Locations

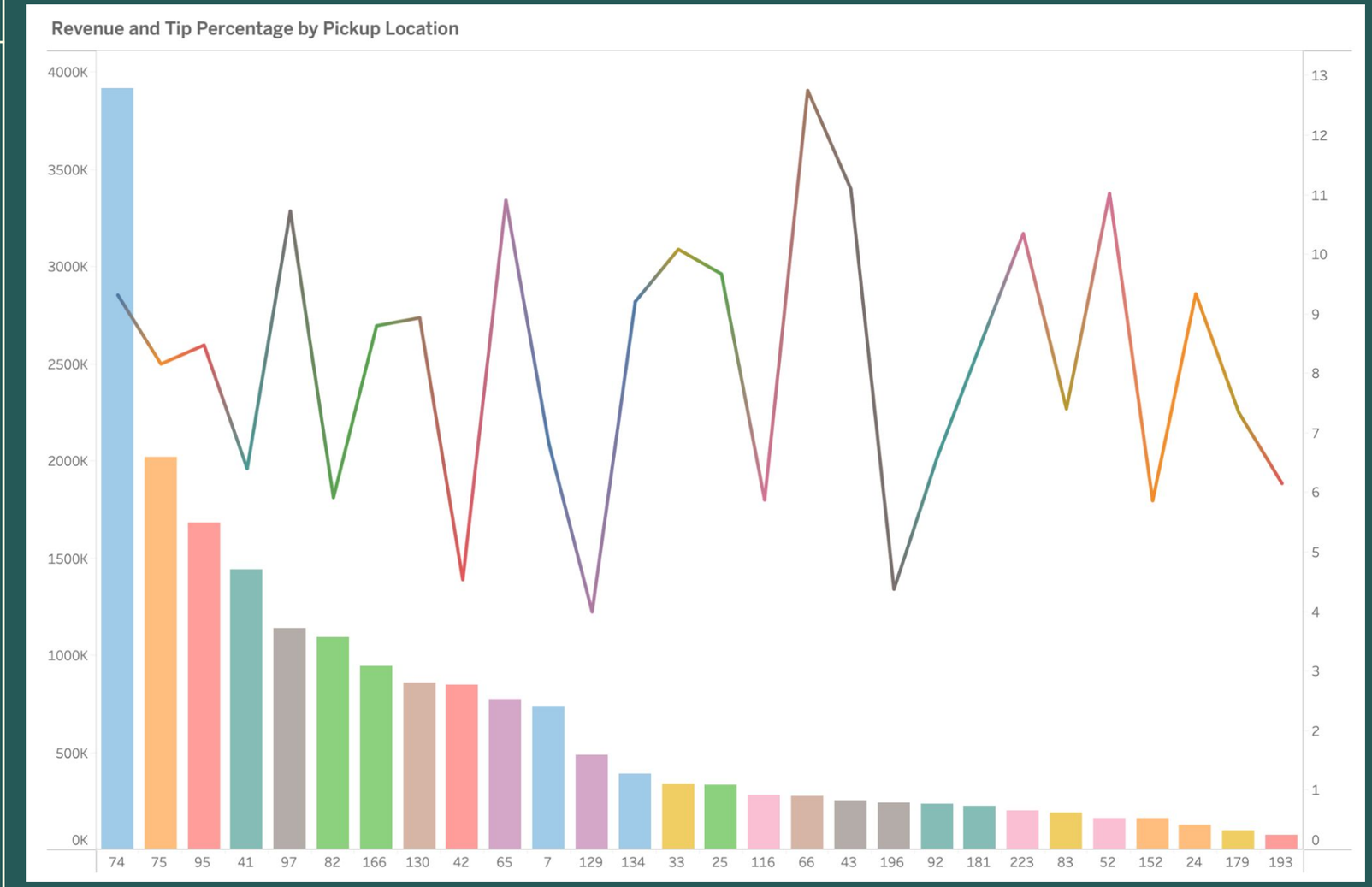


Drop-off Locations



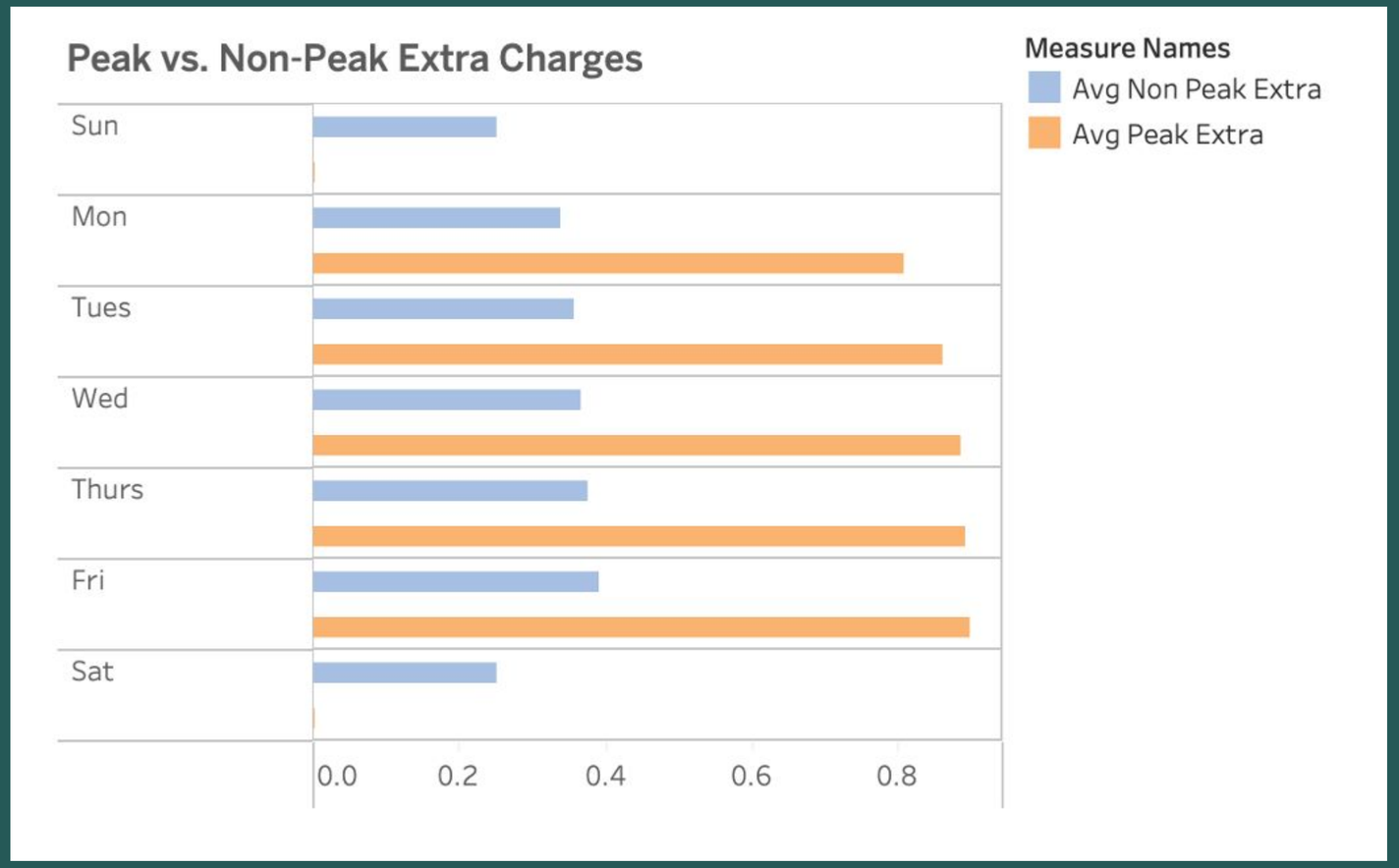
Query 4 - Revenue and Tip Percentage by Pickup Location

Query Description	OLAP Function	Query Code
Revenue & Tip Percentage for High-Trip Locations	Slice	WITH PU_TripCounts AS (SELECT pulocationid, COUNT(trip_id) AS total_trips FROM fact_trips GROUP BY pulocationid,), AverageTrips AS (SELECT AVG(total_trips) AS avg_trips FROM PU_TripCounts) SELECT f.pulocationid, SUM(total_amount) AS total_revenue, AVG(f.tip_amount * 100.0 / f.total_amount) AS avg_tip_percentage FROM fact_trips f JOIN PU_TripCounts pu ON f.pulocationid = pu.pulocationid JOIN AverageTrips a ON pu.total_trips > a.avg_trips GROUP BY f.pulocationid ORDER BY avg_tip_percentage DESC;



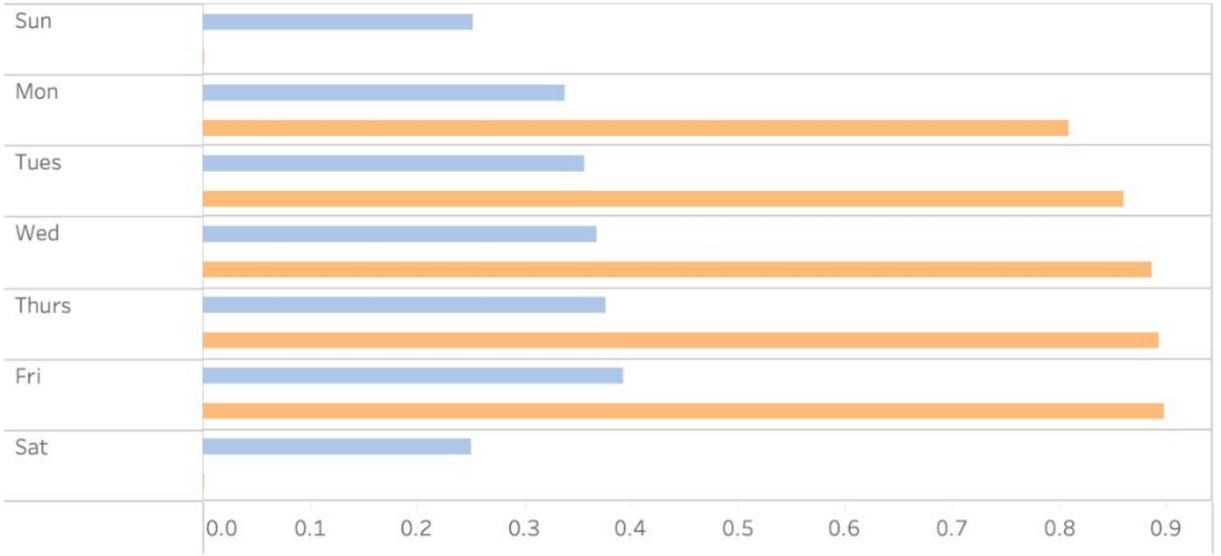
Query 5 - Peak vs. Non-Peak Extra Charges

Query Description	OLAP Function	Query Code
Pivot Extra Charges by Peak vs. Non-Peak Hours and Day of Week	Pivot	SELECT d.day_of_week, AVG(CASE WHEN d.is_peak_hour = 1 THEN f.extra END) AS avg_peak_extra, AVG(CASE WHEN d.is_peak_hour = 0 THEN f.extra END) AS avg_non_peak_extra FROM fact_trips f JOIN dim_date d ON f.lpep_pickup_date_id = d.date_id GROUP BY d.day_of_week;



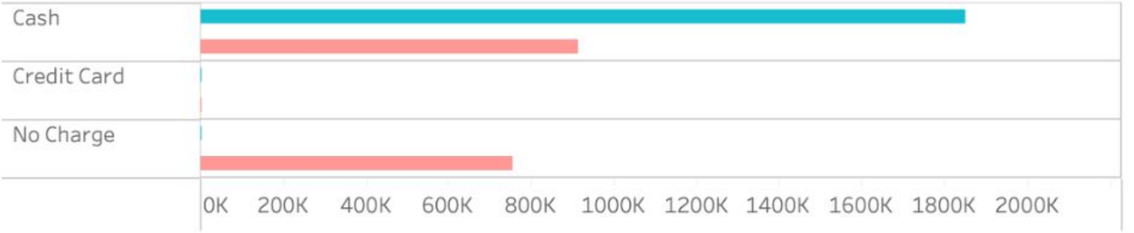
GREEN TAXI ANALYTICS

Peak vs. Non-Peak Extra Charges

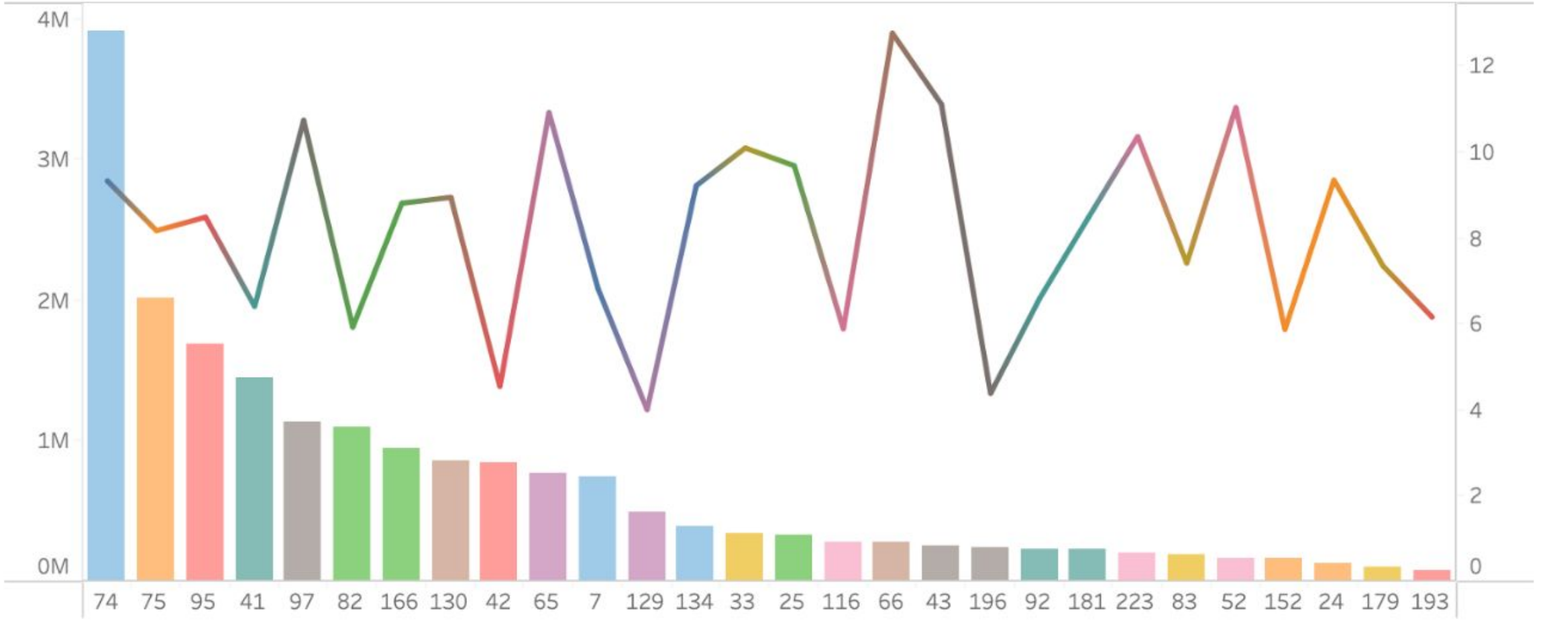


Temporal & Location Trends

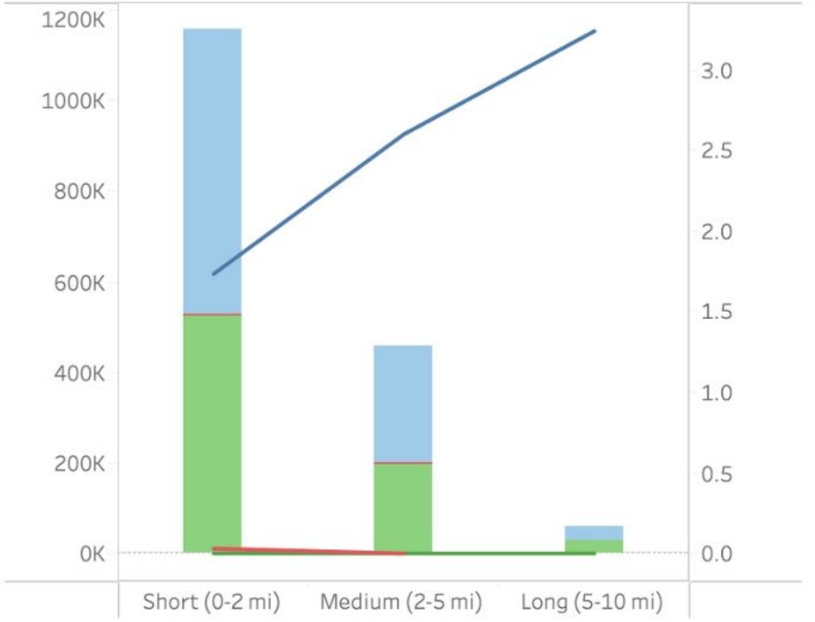
Payment Type Analysis



Revenue and Tip Percentage by Pickup Location



Trip Distance Bucketing



Summary

Challenges	Key Learning	Improvements
1. Understanding the ETL process 2. Using python and MySQL to build the database and star schema 3. Creating action oriented visualizations with Tableau	1. How to extract key insights from data 2. How to use visualizations to communicate to the audience 3. How to apply business intelligence concepts to our project	1. Create additional dimension tables for analysis 2. Incorporate more data for analysis (weather, special events, demographic data) 3. Cross-city comparison (analyze taxi data with similar datasets)

THANKS

Any Questions?