Transit Demand Forecasting for Urban Development

ADSP 31012 IP03 Data Engineering Platforms for Analytics
Final Project
Group 3 - Aadya Nair, Amulya Jayanti, John Melel, Nico Posner, Nidhi Pareddy

Executive Summary

We analyzed publicly available data from the New York City Taxi and Limousine Commission (TLC) to gain valuable insights into urban mobility trends within New York City. This dataset, encompassing for-hire vehicles, rideshare apps, and both green and yellow taxis, provided a comprehensive view of the city's transportation ecosystem.

By examining patterns in transportation demand, accessibility, and rider preferences, we identified key trends, such as peak travel times, popular routes, and variations in ride choices across different demographics and geographies.

These findings have the potential to drive impactful changes in urban planning by identifying bottlenecks, optimizing traffic management, and improving transit accessibility. For transportation service providers, this analysis can inform strategies to enhance customer experiences, allocate resources effectively, and expand service coverage. Overall, our work aims to contribute to a more efficient, equitable, and sustainable transportation network in New York City.

Data Sources

1. New York City Taxi and Limousine Commission, with data and dictionaries for taxi, rideshare, and for-hire vehicle trips https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

2. NYC TLC Zone data Lookup File: https://d37ci6vzurychx.cloudfront.net/misc/

3. NYC Census: https://www.nyc.gov/site/planning/planning-level/nyc-population/2020-census.page

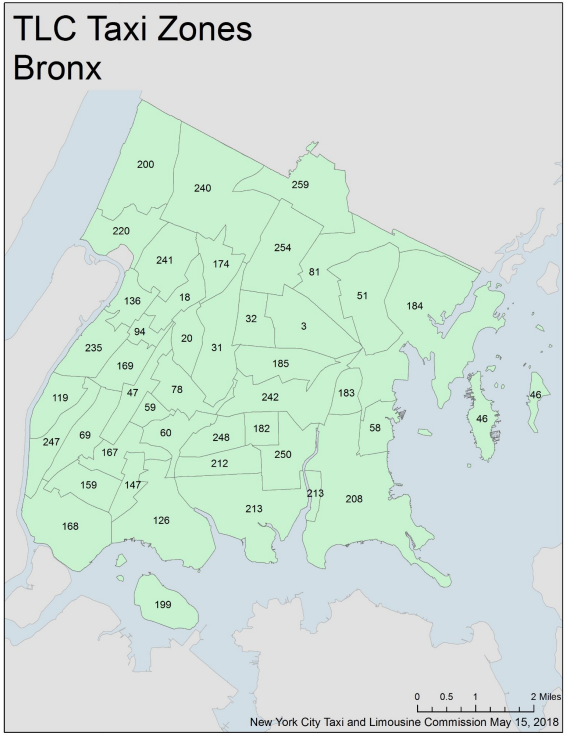


Fig. Bronx TLC Taxi Zone map

Business Use Case

The key business use case of this taxi/rideshare data is as a reliable indicator of desire paths: where do people tend to go, from where, at what times and on what days? This data, capturing all non-personal/commercial automobile traffic in the city, could be used to guide urban planning to guide the creation of new bus routes or measures to relieve congestion.

Alternately, since this traffic is composed disproportionately of tourists, it can help event planners and local businesses plan around common hubs and transportation logistics.

The analysis of New York taxi data aims to evaluate ride accessibility, identify underserved areas, and uncover patterns in ride demand. Insights from this analysis can guide efforts to enhance service accessibility, optimize operations, and inform strategies for promoting equitable and efficient transportation in the future.

Further, it could be useful to taxi organizations to highlight differences in use patterns between themselves and rideshare apps: when/where do people disproportionately go for one over the other, and what opportunities does it highlight for cabs?

Data Preprocessing

Initial Processing:

- Identified and removed fully null rows.
- Converted numerical columns to appropriate types for computation.
- Validated ranges for numerical columns.
- Applied sanity checks on categorical values to ensure valid entries.

Original Loading:

- Normalized tables into 3NF.
- Created ER diagrams to visualize relationships and dependencies.
- Transformed raw Parquet files into CSV for easier access and compatibility with various tools.

Further Processing:

- Designed a dimensional (star/snowflake) schema (sometimes by denormalizing 3NF tables).
- Created Fact Tables to store transactional data and Dimension Tables for descriptive attributes.
- Loaded processed data into a local MySQL server for testing and development.
- Migrated data to a cloud-based MySQL server for scalability and accessibility across teams.

1. Processing for Taxi Data

Initial data table excerpt (contains null values and 0's as well as irrelevant columns)

	Unnamed: 0	VendorID	pickup_datetime	dropoff_datetime	store_and_fwd_flag	RatecodeID	PULocationID	DOLocationID	passenger_count	trip_distance	...	extra	mta_tax	tip_amount
0	0	2	2024-01-01 00:46:55	2024-01-01 00:58:25	N	1.0	236	239	1.0	1.98	...	1.0	0.5	3.61
1	1	2	2024-01-01 00:31:42	2024-01-01 00:52:34	N	1.0	65	170	5.0	6.54	...	1.0	0.5	7.11
2	2	2	2024-01-01 00:30:21	2024-01-01 00:49:23	N	1.0	74	262	1.0	3.08	...	1.0	0.5	3.00
3	3	1	2024-01-01 00:30:20	2024-01-01 00:42:12	N	1.0	74	116	1.0	2.40	...	1.0	1.5	0.00
4	4	2	2024-01-01 00:32:38	2024-01-01 00:43:37	N	1.0	74	243	1.0	5.14	...	1.0	0.5	6.28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
56546	56546	2	2024-01-31 20:46:00	2024-01-31 20:55:00	NaN	NaN	33	25	NaN	0.00	...	0.0	0.0	3.14
56547	56547	2	2024-01-31 21:06:00	2024-01-31 21:11:00	NaN	NaN	72	72	NaN	0.49	...	0.0	0.0	0.00
56548	56548	2	2024-01-31 21:36:00	2024-01-31 21:40:00	NaN	NaN	72	72	NaN	0.52	...	0.0	0.0	2.52
56549	56549	2	2024-01-31 22:45:00	2024-01-31 22:51:00	NaN	NaN	41	42	NaN	1.17	...	0.0	0.0	0.00
56550	56550	2	2024-01-31 22:28:00	2024-01-31 22:59:00	NaN	NaN	33	91	NaN	9.27	...	0.0	0.0	4.56

56551 rows × 21 columns

Final data table excerpted (merged yellow and green taxi datasets, removed unnecessary rows as well as null and all 0 values)

ride_id	VendorID	pickup_datetime	dropoff_datetime	PULocationID	DOLocationID	passenger_count	trip_distance	fare_amount	total_amount	payment_type	congestion_surcharge	t
0	2	2024-01-01 00:46:55	2024-01-01 00:58:25	236	239	1.0	1.98	12.80	21.66	1.0	2.75	
1	2	2024-01-01 00:31:42	2024-01-01 00:52:34	65	170	5.0	6.54	30.30	42.66	1.0	2.75	
2	2	2024-01-01 00:30:21	2024-01-01 00:49:23	74	262	1.0	3.08	19.80	28.05	1.0	2.75	
3	1	2024-01-01 00:30:20	2024-01-01 00:42:12	74	116	1.0	2.40	14.20	16.70	2.0	0.00	
4	2	2024-01-01 00:32:38	2024-01-01 00:43:37	74	243	1.0	5.14	22.60	31.38	1.0	0.00	
...	...	...	...	...	...	...	...	...	...	...	...	
1692310	1	2024-01-20 00:23:09	2024-01-20 00:36:25	79	140	1.0	3.40	16.30	25.55	1.0	2.50	
2871723	1	2024-01-12 23:45:04	2024-01-12 23:52:07	162	140	NaN	0.00	10.83	14.83	0.0	NaN	
50320	1	2024-01-01 16:48:59	2024-01-01 16:55:30	239	151	2.0	1.30	9.30	15.95	1.0	2.50	
293937	1	2024-01-04 18:06:23	2024-01-04 18:16:04	234	234	1.0	0.80	9.30	20.55	1.0	2.50	
1796731	2	2024-01-21 01:16:41	2024-01-21 01:21:20	79	107	1.0	0.95	7.20	13.42	1.0	2.50	

2. Processing for Rideshare & FHV (For Hire Vehicle) Data

	hvfhs_license_num	dispatching_base_num	originating_base_num	request_datetime	on_scene_datetime	pickup_datetime	dropoff_date
0	HV0003	B03404	B03404	2024-01-06T22:25:04.000000000	2024-01-06T22:25:13.000000000	2024-01-06T22:26:46.000000000	202-06T22:42:02.00000
1	HV0005	B03406	NaN	2024-01-05T20:14:37.000000000	NaN	2024-01-05T20:21:34.000000000	202-05T20:36:53.00000
2	HV0003	B03404	B03404	2024-01-22T17:53:38.000000000	2024-01-22T17:54:35.000000000	2024-01-22T17:56:35.000000000	202-22T18:02:26.00000
3	HV0003	B03404	B03404	2024-01-19T21:30:01.000000000	2024-01-19T21:31:17.000000000	2024-01-19T21:33:14.000000000	202-19T21:54:08.00000
4	HV0003	B03404	B03404	2024-01-06T16:56:53.000000000	2024-01-06T17:00:02.000000000	2024-01-06T17:01:46.000000000	202-06T17:15:48.00000

5 rows × 24 columns

	Unnamed: 0	dispatching_base_num	pickup_datetime	dropOff_datetime	PUlocationID	DOlocationID	SR_Flag	Affiliated_base_number
0	582504	B00856	2024-01-15 18:55:10	2024-01-15 19:16:23	NaN	76.0	NaN	B03283
1	154119	B01667	2024-01-05 03:55:58	2024-01-05 04:06:21	NaN	129.0	NaN	B03147
2	792907	B02550	2024-01-20 11:14:18	2024-01-20 11:25:25	NaN	119.0	NaN	B02902
3	574229	B03116	2024-01-15 13:30:05	2024-01-15 13:58:58	NaN	248.0	NaN	B03116
4	304354	B00149	2024-01-09 08:27:41	2024-01-09 08:36:50	NaN	17.0	NaN	B00149

Both datasets contained several null entries with missing pickup and drop-off locations, as well as columns that were irrelevant to the scope of our analysis, which were removed.

3. Consolidated Hire Vehicle Data

	hvfhs_license_num	dispatching_base_num	pickup_datetime	dropoff_datetime	PULocationID	DOLocationID	trip_miles	trip_time	base_pa
0	HV0003	B03404	2024-01-06T22:26:46.000000000	2024-01-06T22:42:02.000000000	234.0	265.0	4.310	916.0	
1	HV0005	B03406	2024-01-05T20:21:34.000000000	2024-01-05T20:36:53.000000000	220.0	174.0	1.867	919.0	
2	HV0003	B03404	2024-01-22T17:56:35.000000000	2024-01-22T18:02:26.000000000	193.0	202.0	0.800	351.0	
3	HV0003	B03404	2024-01-19T21:33:14.000000000	2024-01-19T21:54:08.000000000	216.0	205.0	4.570	1254.0	
4	HV0003	B03404	2024-01-06T17:01:46.000000000	2024-01-06T17:15:48.000000000	226.0	83.0	2.320	842.0	
...	...	...	...	...	...	...	...	...	
308470	NaN	B01176	2024-01-31 01:51:31	2024-01-31 02:09:26	83.0	173.0	NaN	NaN	
308471	NaN	B01626	2024-01-29 08:32:49	2024-01-29 08:42:55	NaN	121.0	NaN	NaN	
308472	NaN	B00647	2024-01-19 23:08:38	2024-01-19 23:26:06	NaN	254.0	NaN	NaN	
308473	NaN	B02254	2024-01-27 13:26:58	2024-01-27 13:40:42	29.0	55.0	NaN	NaN	
308474	NaN	B01546	2024-01-25 16:49:59	2024-01-25 17:42:11	181.0	76.0	NaN	NaN	

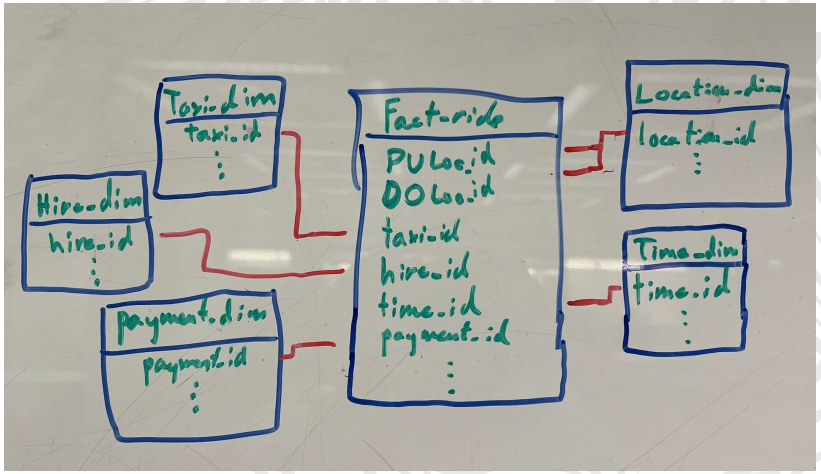
308475 rows × 21 columns

After cleaning the datasets, we merged them into a single dataset and added an identifier column to distinguish between FHV and Rideshare.

Design of Conceptual and Logical Model (1)

The database is organized as a **dimensional model,** with all fields feeding into a central fact_ride table.

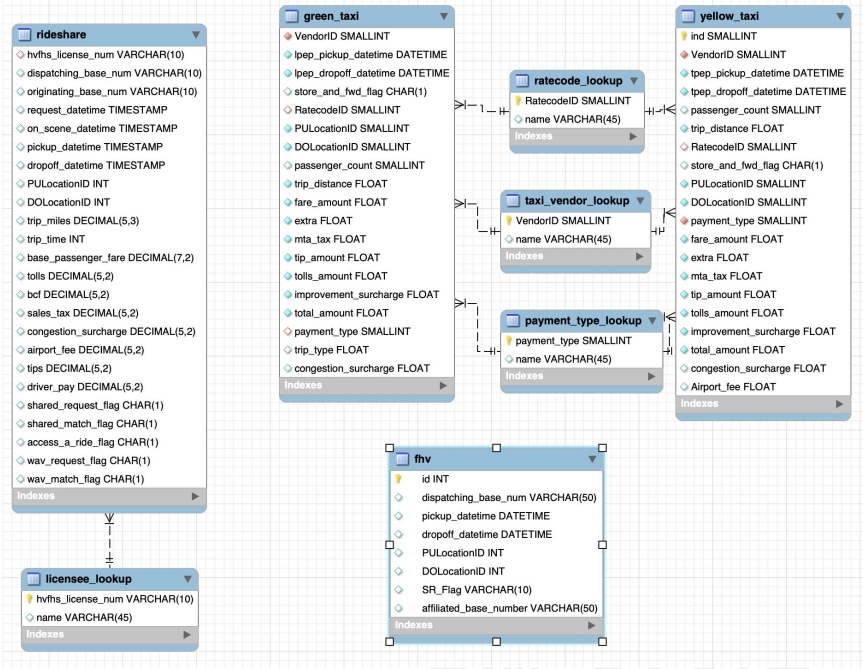
We chose this form in order to focus on the main element of interest, the rides themselves and their patterns over space, time, and service.



Design of Conceptual and Logical Model (2)

When filling out the logical model, our greatest obstacle was the **disconnected structure** of the data, followed by the presence of nulls across categories depending on the source: not all the listed services closely track the time or location of pickups and dropoffs.

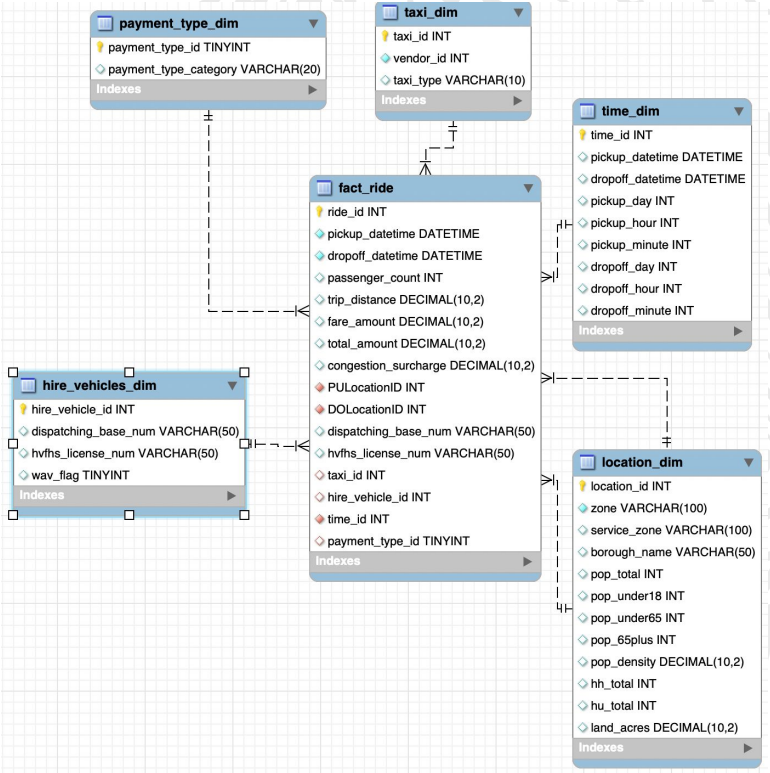
Therefore, we added our census data into the location dimension table, creating a condensed version of our model.



Physical Model (EER)

The final implementation of the physical model on GCS was constructed by staging our processed datasets before transforming them into a dimensional model and dropping the redundant staging tables.

All dimensions are organized around the central **fact_ride** table, with supplementary details of time, location, vehicle identification, and payment arrayed around it.



Database Creation and Loading

After producing the physical model, we exported it into a SQL dump in order to upload and recreate it on a remote server, from which we ran our queries and constructed our visualizations.

We used Google Cloud Platform (GCP) to host our common MySQL server to allow for simultaneous use with the MySQL application on personal computers.

The tables were created first using DDL statements, data loading done through DML statements, and the loaded data tables were modified and additional tables were made to create the final data model.

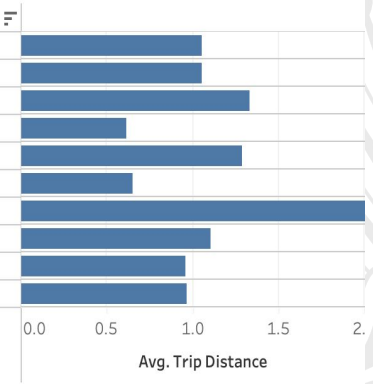
A few tables used as intermediary tables for data staging were dropped at the end of this step.

Insights and Analytics: Popular Routes

Findings:
The 10 most commonly traveled routes, are overwhelmingly short, either within individual zones or between adjacent zones, mainly in Manhattan. These routes have very short average rides, usually under 2 miles, compared to an average of 7 miles for all rides.

Popular Routes with Average Distance

Route	Pickup	Dropoff	Count of ..
237 -> 236	Upper East Side South	Upper East Side North	46,055
236 -> 237	Upper East Side North	Upper East Side South	40,927
74 -> 75	East Harlem North	East Harlem South	36,665
236 -> 236	Upper East Side North	Upper East Side North	34,515
75 -> 74	East Harlem South	East Harlem North	32,966
237 -> 237	Upper East Side South	Upper East Side South	30,170
74 -> 236	East Harlem North	Upper East Side North	24,216
161 -> 237	Midtown Center	Upper East Side South	23,392
95 -> 95	Forest Hills	Forest Hills	21,450
75 -> 236	East Harlem South	Upper East Side North	19,482



Business Case:
The high volume of short rides along common routes suggests that public transport, particularly buses, may be underperforming in these areas. Enhancing surface transport infrastructure should prioritize increasing capacity and efficiency on these routes.

Insights and Analytics: Popular Routes

Business Case: Understanding the most popular ride routes

```sql
#Popular routes(Zone wise):
WITH ride_counts AS (
    SELECT
        PULocationID,
        DOLocationID,
        COUNT(*) AS ride_count
    FROM fact_ride
    GROUP BY PULocationID, DOLocationID
),
pickup_zones AS (
    SELECT
        rc.PULocationID,
        rc.DOLocationID,
        rc.ride_count,
        pl.zone AS pickup_zone
    FROM ride_counts rc
    JOIN location_dim pl ON rc.PULocationID = pl.location_id
),
pickup_and_dropoff_zones AS (
    SELECT
        pz.PULocationID,
        pz.pickup_zone,
        pz.DOLocationID,
        dl.zone AS dropoff_zone,
        pz.ride_count
    FROM pickup_zones pz
    JOIN location_dim dl ON pz.DOLocationID = dl.location_id
)
SELECT
    PULocationID,
    pickup_zone,
    DOLocationID,
    dropoff_zone,
    ride_count
FROM pickup_and_dropoff_zones
ORDER BY ride_count DESC
LIMIT 10;
```

Code → Output

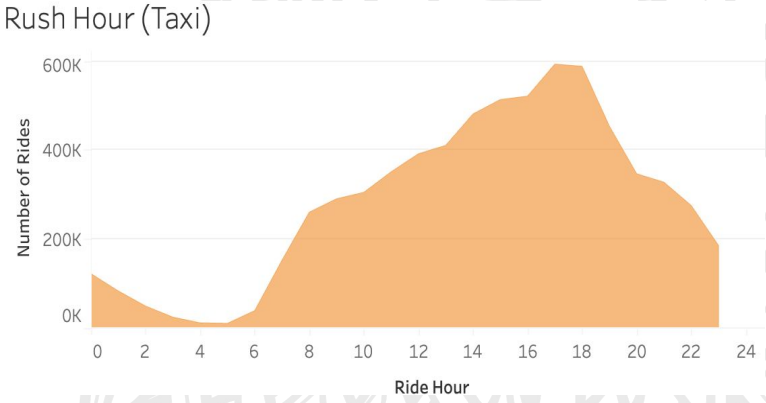
PULocationID	pickup_zone	DOLocationID	dropoff_zone	ride_count
237	Upper East Side South	236	Upper East Side North	46055
236	Upper East Side North	237	Upper East Side South	40927
74	East Harlem North	75	East Harlem South	36665
236	Upper East Side North	236	Upper East Side North	34515
75	East Harlem South	74	East Harlem North	32966
237	Upper East Side South	237	Upper East Side South	30170
74	East Harlem North	236	Upper East Side North	24216
161	Midtown Center	237	Upper East Side South	23392
95	Forest Hills	95	Forest Hills	21450
75	East Harlem South	236	Upper East Side North	19482

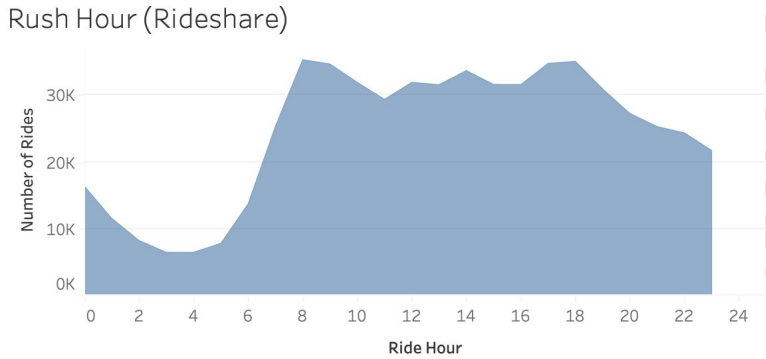
Insights and Analytics: Rush Hours

Findings: An analysis of usage patterns reveals:

- **Taxi Usage**: Peaks at **5 PM and 6 PM**, suggesting strong demand during evening commutes and leisure travel.
- **Rideshare/FHV Usage**: Peaks at **8 AM**, remains elevated throughout the day, tapering off after 6 PM. This indicates a broader coverage of morning commutes, business travel, and daily errands.

Business Case: Understanding these disparate usage patterns enables us to fine-tune operations, optimize driver availability, and offer differentiated services ultimately enhancing customer satisfaction, improve urban mobility, and increase profitability.





Insights and Analytics: Rush Hours

Business Case: Understanding the peak/rush and the non-rush hours of transport through taxis and ride shares.

```sql
#Peak Hours (Rush vs Non-Rush):

WITH hourly_ride_counts AS (
    SELECT
        fr.time_id,
        td.pickup_hour,
        COUNT(*) AS ride_count
    FROM fact_ride fr
    JOIN time_dim td ON fr.time_id = td.time_id
    GROUP BY fr.time_id, td.pickup_hour
),
time_period_classification AS (
    SELECT
        pickup_hour,
        SUM(ride_count) AS total_ride_count,
        CASE
            WHEN pickup_hour BETWEEN 7 AND 9 THEN 'Morning Rush'
            WHEN pickup_hour BETWEEN 17 AND 19 THEN 'Evening Rush'
            ELSE 'Non-Rush'
        END AS time_period
    FROM hourly_ride_counts
    GROUP BY pickup_hour, time_period
)
SELECT
    pickup_hour,
    total_ride_count AS ride_count,
    time_period
FROM time_period_classification
ORDER BY ride_count DESC;
```

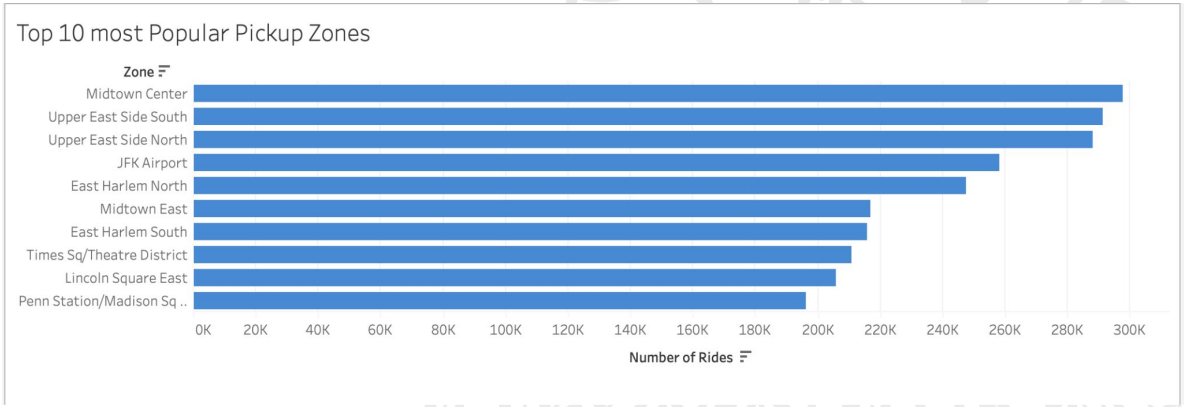
Code | Output

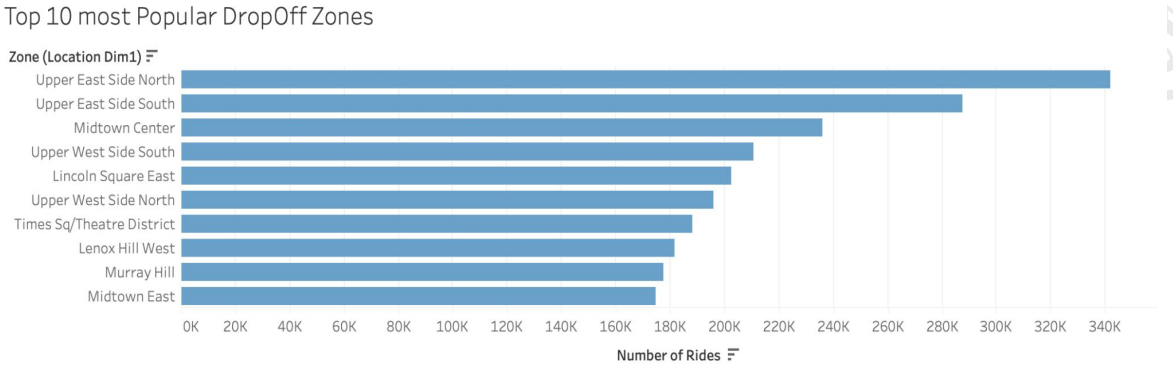
pickup_ho...	ride_count	time_period
17	627373	Evening Rush
18	622833	Evening Rush
16	552325	Non-Rush
15	544577	Non-Rush
14	514288	Non-Rush
19	484332	Evening Rush
13	441665	Non-Rush
12	423181	Non-Rush
11	380125	Non-Rush
20	373633	Non-Rush
21	352700	Non-Rush
10	336552	Non-Rush
9	324824	Morning Rush
22	299401	Non-Rush
8	295389	Morning Rush
23	207081	Non-Rush
7	177454	Morning Rush
0	138232	Non-Rush
1	94673	Non-Rush
2	57381	Non-Rush
6	52610	Non-Rush
3	30779	Non-Rush
5	18138	Non-Rush
4	17962	Non-Rush

Insights and Analytics: Popular Zones

Findings:
There is some overlap between popular pickup and dropoff zones, with the Upper East Side and Midtown being extremely popular start and end points.

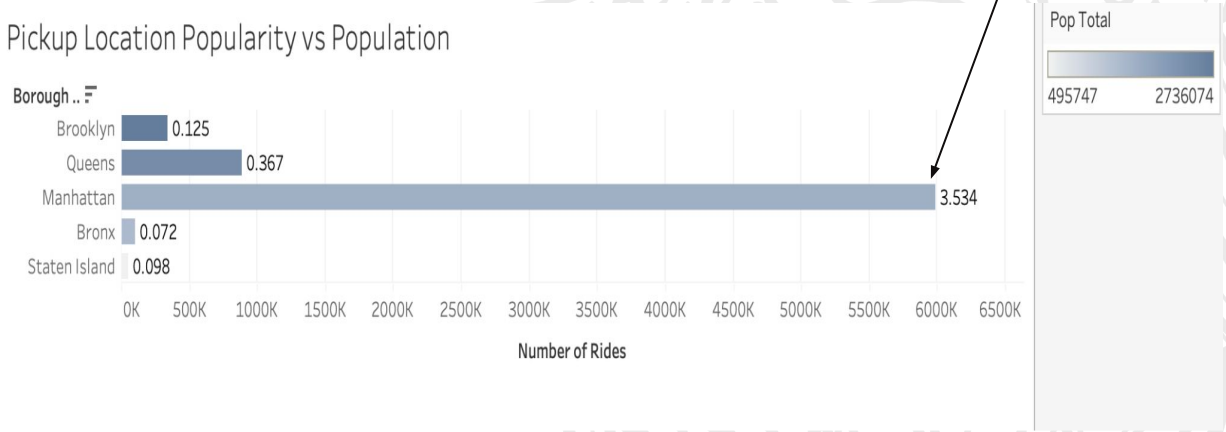
Business Case:
The overlap in popular pickup and dropoff zones, particularly in areas like the Upper East Side and Midtown, highlights high-demand corridors. This suggests opportunities for enhancing public transportation infrastructure to alleviate congestion and improve accessibility, ultimately optimizing urban mobility.





Insights and Analytics: Popularity/Population

Findings:
Different boroughs have greatly varying ratios of population to ride popularity, with pickups per capita in Manhattan being far higher than in Brooklyn or the Bronx.



Business Case:
Taxi and rideshare usage in Manhattan is disproportionately higher relative to its population, despite its walkability and extensive subway system. This suggests these services may compete more with private car ownership, which is more prevalent in other boroughs, rather than with walking or public transit.

Insights and Analytics: Popularity vs Population (Census)

Business Case: Census - Popularity vs Population

To understand the total population(segregated at borough level) vs. the rides taken

```sql
#Census – Popularity vs Population: Total population in boroughs vs. the rides taken
WITH location_rides AS (
    SELECT
        PULocationID,
        COUNT(ride_id) AS ride_count
    FROM fact_ride
    GROUP BY PULocationID
),

borough_rides AS (
    SELECT
        lr.PULocationID,
        l.borough_name,
        l.pop_total,
        lr.ride_count
    FROM location_rides lr
    JOIN location_dim l ON lr.PULocationID = l.location_id
)

SELECT
    borough_name,
    pop_total,
    SUM(ride_count) AS total_rides
FROM borough_rides
GROUP BY borough_name, pop_total
ORDER BY total_rides DESC;
```

Code Output

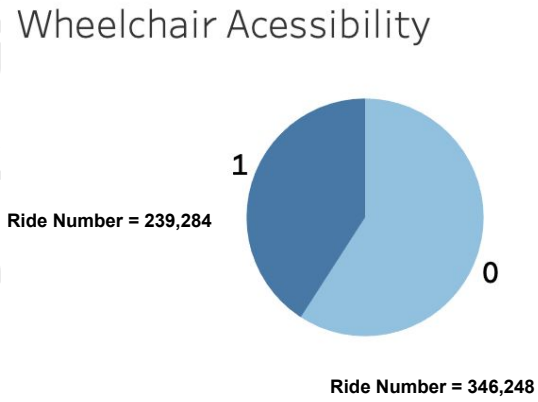
borough_name	pop_total	total_rides
Manhattan	1694251	5987586
Queens	2405464	883857
Brooklyn	2736074	341495
Bronx	1472654	106159
Staten Island	495747	48411

Insights and Analytics: Wheelchair Accessibility

Findings:
We find that around 1/3rd of rideshares have accommodations for wheelchairs.

Business Case:
Given that approximately 1 in 80 New Yorkers use a wheelchair, this fairly wide coverage in rideshares is good. But this should be expanded to taxis(green/yellow), to provide robust accessibility.

Key:
1 = Rides with wheelchair accessibility
0 = Rides without wheelchair accessibility



Insights and Analytics: Wheelchair Accessibility(1)

Business Case: Understanding how many rides are accessible or inaccessible using Wheelchair accessibility(wav flag), to ultimately promote accessible transport

```sql
#How many rides are accessible or inaccessible -> Wheelchair accessibility(wav flag):

WITH vehicle_rides AS (
    SELECT
        hire_vehicle_id,
        COUNT(*) AS ride_count
    FROM fact_ride
    GROUP BY hire_vehicle_id
),

wav_rides AS (
    SELECT
        v.hire_vehicle_id,
        h.wav_flag,
        v.ride_count
    FROM vehicle_rides v
    JOIN hire_vehicles_dim h ON v.hire_vehicle_id = h.hire_vehicle_id
)

SELECT
    wav_flag,
    SUM(ride_count) AS ride_count
FROM wav_rides
GROUP BY wav_flag
ORDER BY ride_count DESC;
```

Code | Output

wav_flag	ride_count
0	346248
1	239284

Key:
1 = Rides with wheelchair accessibility
0 = Rides without wheelchair accessibility

Insights and Analytics: Wheelchair Accessibility(2)

Business Case: Routes which have accessible rides, to ultimately promote accessible transport

```sql
# All routes with wheelchair accessible rides (wheelchair accessibility- wav flag=1):

-- Aggregate ride counts by hire_vehicle_id, PULocationID, and DOLocationID
WITH vehicle_rides AS (
    SELECT
        f.hire_vehicle_id,
        f.PULocationID,
        f.DOLocationID,
        COUNT(*) AS ride_count
    FROM fact_ride f
    GROUP BY f.hire_vehicle_id, f.PULocationID, f.DOLocationID
),

-- Join the aggregated data with hire_vehicles_dim to include WAV flag
wav_rides AS (
    SELECT
        vr.hire_vehicle_id,
        vr.PULocationID,
        vr.DOLocationID,
        vr.ride_count,
        h.wav_flag
    FROM vehicle_rides vr
    JOIN hire_vehicles_dim h ON vr.hire_vehicle_id = h.hire_vehicle_id
),

-- Add zone names for PULocationID and DOLocationID
zone_routes AS (
    SELECT
        wr.PULocationID,
        pl.zone AS pickup_zone,
        wr.DOLocationID,
        dl.zone AS dropoff_zone,
        wr.ride_count,
        wr.wav_flag
    FROM wav_rides wr
    JOIN location_dim pl ON wr.PULocationID = pl.location_id
    JOIN location_dim dl ON wr.DOLocationID = dl.location_id
-- Aggregate rides by route and WAV flag
route_analysis AS (
    SELECT
        pickup_zone,
        dropoff_zone,
        wav_flag,
        SUM(ride_count) AS total_rides
    FROM zone_routes
    GROUP BY pickup_zone, dropoff_zone, wav_flag
)

-- Filter and display results for accessible rides (wav_flag = 1)
SELECT
    pickup_zone,
    dropoff_zone,
    total_rides AS accessible_rides
FROM route_analysis
WHERE wav_flag = 1
ORDER BY accessible_rides DESC;
```

Code | Output

pickup_zone	dropoff_zone	accessible_rides
East New York	East New York	866
Borough Park	Borough Park	638
Canarsie	Canarsie	514
Crown Heights North	Crown Heights North	424
Jackson Heights	Jackson Heights	386
Bay Ridge	Bay Ridge	372
Forest Hills	Forest Hills	326
Astoria	Astoria	325
South Ozone Park	JFK Airport	322
South Ozone Park	South Ozone Park	286
Park Slope	Park Slope	282
Bushwick North	Bushwick South	280
Bushwick South	Bushwick South	267
Williamsburg (North…	Greenpoint	257
Jamaica	Jamaica	244
Long Island City/Hu…	Long Island City/Hu…	243

Finding: High number of rides with wheelchair accessibility in East New York and Borough Park

Insights and Analytics: Choice of Ride

Business Case: Understanding the popular choice of Ride (Taxi vs rideshare) by aggregating rides by taxi_id and dispatching_base_num

```sql
#Choice of Ride (Taxi vs Rideshare):
-- Aggregate rides by taxi_id and dispatching_base_num
WITH aggregated_rides AS (
    SELECT
        f.taxi_id,
        f.dispatching_base_num,
        COUNT(*) AS ride_count
    FROM fact_ride f
    GROUP BY f.taxi_id, f.dispatching_base_num
),

-- Classify rides as Taxi or Rideshare with appropriate identifiers
ride_classification AS (
    SELECT
        CASE
            WHEN ar.taxi_id IS NOT NULL THEN ar.taxi_id
            ELSE ar.dispatching_base_num
        END AS identifier,
        CASE
            WHEN ar.taxi_id IS NOT NULL THEN 'Taxi'
            ELSE 'Rideshare'
        END AS ride_type,
        ar.ride_count
    FROM aggregated_rides ar
)

SELECT
    identifier,
    ride_type,
    SUM(ride_count) AS ride_count
FROM ride_classification
GROUP BY identifier, ride_type
ORDER BY ride_count DESC;
```

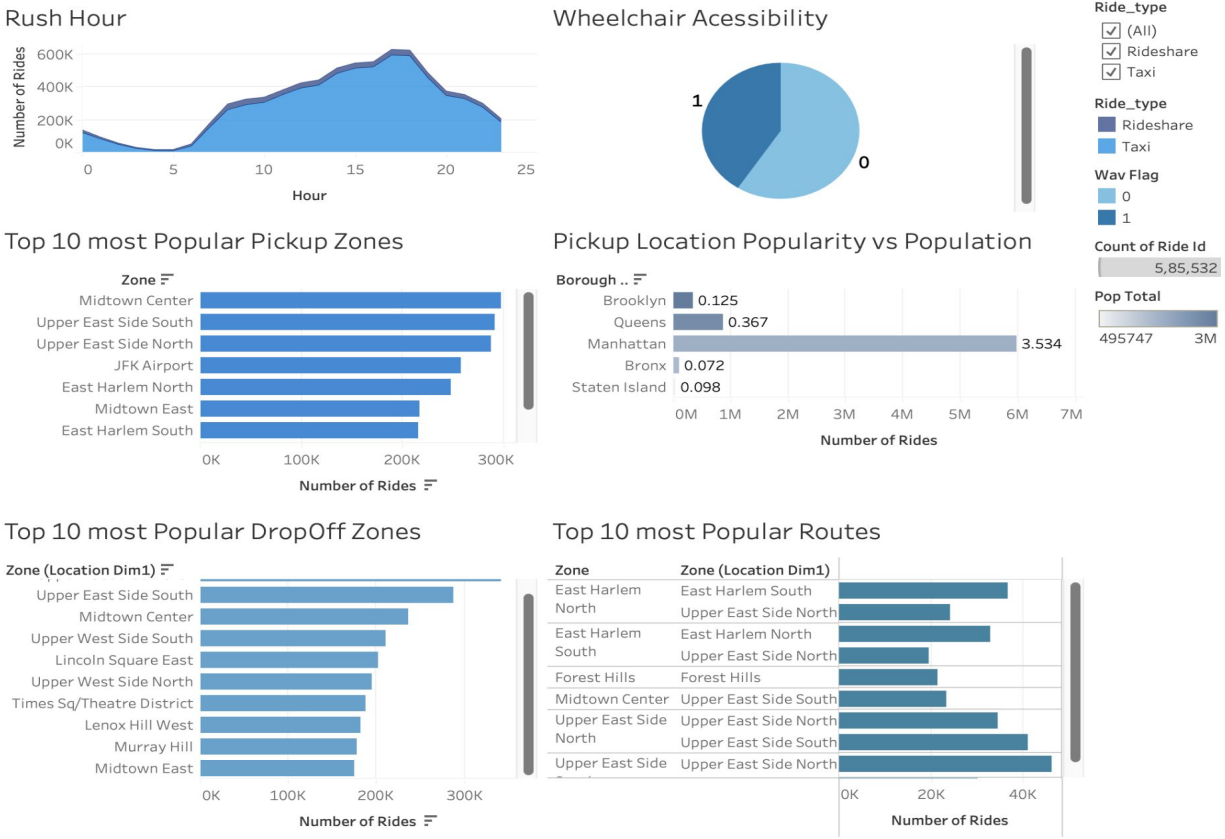
Code | Output

Finding: Taxis are used at a higher frequency compared to rideshares

identifier	ride_type	ride_count
3	Taxi	2626937
1	Taxi	2626937
4	Taxi	764051
2	Taxi	764051
B03404	Rideshare	348750
B03406	Rideshare	129818
B01288	Rideshare	27504
B00965	Rideshare	21823
B03160	Rideshare	8476
B03046	Rideshare	7682
B01268	Rideshare	5846
B02661	Rideshare	3573
B03494	Rideshare	3353
B01546	Rideshare	2917
B03252	Rideshare	2325
B03455	Rideshare	2005
B01710	Rideshare	1651
B02795	Rideshare	1617
B03284	Rideshare	1370
B02301	Rideshare	1368
B01176	Rideshare	1288
B00706	Rideshare	1054
B03285	Rideshare	1015
B03266	Rideshare	952
B03320	Rideshare	729
B01087	Rideshare	701

Dashboard

This dashboard lends an intuitive visualization of all the insights in an easily engageable way. The main filter on ride type (taxi or rideshare) allows for different explorations based on business needs.



Recommendations: Corrective Measures

1. **Optimize Data Loading via GCP:**
   - *Current Issue:* Data ingestion was suboptimal, potentially leading to higher latency and resource consumption.
   - *Improvement:* Utilize **Google Cloud Storage (GCS)** for staging data and leverage **BigQuery** for direct querying of large datasets, avoiding intermediate transformations. Enable **parallelized loading** through GCP's Dataflow or Apache Beam to improve ETL efficiency.
2. **Database Selection - NoSQL Integration:**
   - *Current Issue:* Sole reliance on relational databases limits flexibility in handling unstructured or semi-structured data.
   - *Improvement:* In future, introduce **MongoDB** for handling dynamic datasets, such as driver feedback or ride metadata. NoSQL databases would offer better scalability for high-volume data scenarios like real-time trip data analysis.
3. **Ensure Data Completeness and Quality:**
   - Regularly audit data for missing values and anomalies, particularly with fields like wav_flag or location data.
   - Implement automated checks and **data deduplication scripts** in pipelines to ensure consistent and reliable analytics.

Recommendations: Scope for improvement

Expand business use cases:

1. Revenue Optimization: Analyze fare structures to identify optimal pricing strategies during rush hours and low-demand periods.
2. Sustainability Insights: Explore emissions data to compare the carbon footprint of traditional taxis versus rideshare, and propose greener routes.

Geospatial and Demographic Insights:

1. Overlay geospatial data with borough demographics to correlate ride popularity with income levels, housing density, or public transport availability.
2. Analyze cross-borough ride trends to identify potential partnerships or service expansions.