Report: Optimising NYC Taxi Operations

Include your visualisations, analysis, results, insights, and outcomes. Explain your methodology and approach to the tasks. Add your conclusions to the sections.

1. Data Preparation

1.1. Loading the dataset

Multiple NYC Yellow Taxi CSV files were loaded using pandas. While reading, all timestamp columns (tpep_pickup_datetime, tpep_dropoff_datetime) were parsed as datetime objects to enable time-based feature extraction. Numeric fields such as fares, surcharges, trip distance, and passenger counts were loaded as numeric types where possible.

1.1.1. Sample the data and combine the files

Since the yearly dataset is very large, each monthly CSV was sampled using a fixed frac value and a fixed random_state. All sampled DataFrames were concatenated into a single DataFrame using pd.concat(...).
This produced a representative but computationally manageable dataset for EDA.

2. Data Cleaning

2.1. Fixing Columns

Column names were standardised to maintain consistency across months.
Datatypes were corrected (string → numeric, string → datetime).
Taxi zone-related fields, rate codes, and payment types were ensured to be integers or categories.

2.1.1. Fix the index

After concatenation and filtering, the DataFrame index was no longer sequential. We reset the index using reset_index(drop=True) to maintain a clean structure for merging later.operations was dropped.

2.1.2. Combine the two airport_fee columns

Some files contained both airport_fee and Airport_fee. To unify them:

1. They were combined into a single airport_fee column.
2. The duplicate column was dropped.

A later inspection showed the values were consistently zero or missing, so the final airport_fee column was removed as it had no analytic value.

2.2. Handling Missing Values

2.2.1. Find the proportion of missing values in each column

We computed the percentage of missing values using df.isna().mean(). This helped identify key fields with missing data (e.g., passenger_count, surcharges).

2.2.2. Handling missing values in passenger_count

Missing or zero passenger counts were filled with 1 based on the assumption that most trips are single-passenger unless otherwise recorded. This decision reduced noise while preventing division-by-zero in passenger-based calculations.

2.2.3. Handle missing values in RatecodeID

Missing RatecodeID values were imputed using the mode of the column, ensuring categorical consistency.

2.2.4. Impute NaN in congestion_surcharge

Missing congestion surcharge entries were treated as zero, reflecting the assumption that a missing surcharge indicates it was not applied.

2.3. Handling Outliers and Standardising Values

2.3.1. Check outliers in payment type, trip distance and tip amount columns

We explored outliers using:

- Distribution checks (describe())
- Value counts for categorical fields

- Visual checks for extreme values

Outlier handling actions:

- Removed rows with negative values in critical numerical fields (distance, fare, tips, total amount).
- Converted negative surcharges and taxes to absolute values (common data entry issue).
- Removed impossible trip distances and durations (e.g., zero distance with large fare).

This ensured downstream analytics would not be distorted.

3. Exploratory Data Analysis

3.1. General EDA: Finding Patterns and Trends

3.1.1. Classify variables into categorical and numerical

Categorical variables: VendorID, RatecodeID, payment_type, pickup/dropoff zones, store_and_fwd_flag.
Numerical variables: trip distance, fare components, total amount, tips, surcharges.

This classification guided the type of plots and aggregations used.

3.1.2. Analyse the distribution of taxi pickups by hours, days of the week, and months

Using datetime-derived columns (hour, day_of_week, month), we grouped and plotted trip counts.
Findings show:

- Morning and evening peaks (commute hours)
- Higher late-night activity on weekends
- Month-to-month variations in demand

3.1.3. Filter out the zero/negative values in fares, distance and tips

We created a filtered dataset (df_1) containing only rows with valid (positive) amounts and distances, used for correlation- and fare-based analyses.

3.1.4. Analyse the monthly revenue trends

Revenue per month was calculated using sum of total_amount. Seasonal patterns and strong months were identified.

3.1.5. Find the proportion of each quarter's revenue in the yearly revenue

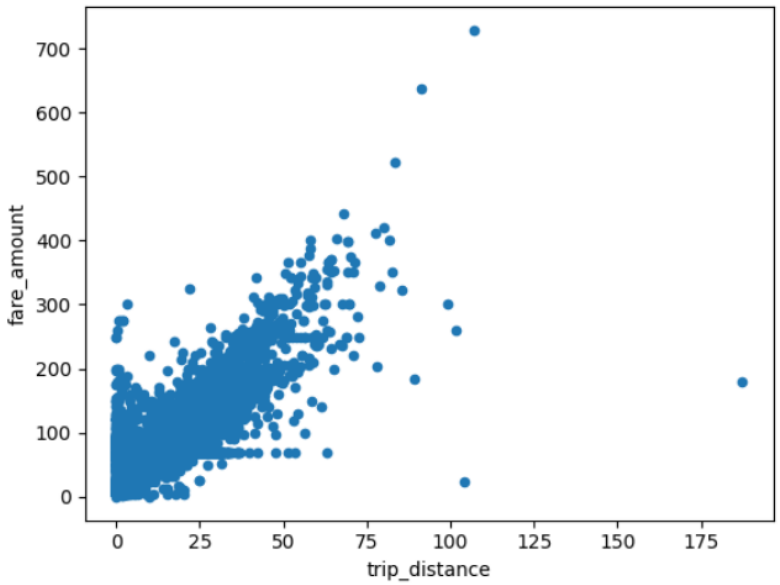
Quarterly revenue was calculated and compared against annual revenue. This highlighted the strongest and weakest quarters**.**

3.1.6. Analyse and visualise the relationship between distance and fare amount

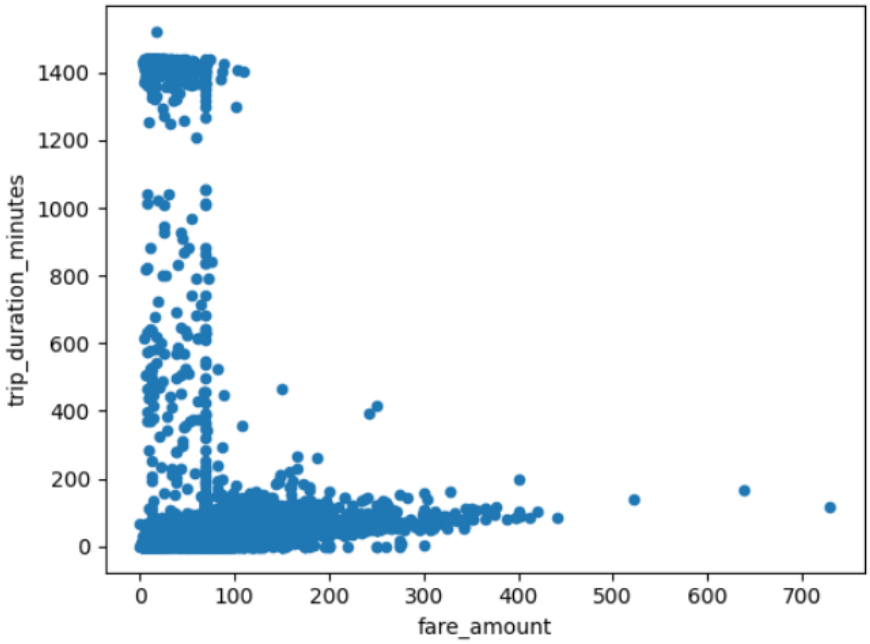
Scatter plots and correlations showed a clear positive trend, with variance introduced by surcharges, traffic, and rate codes.

Scatterplot for:

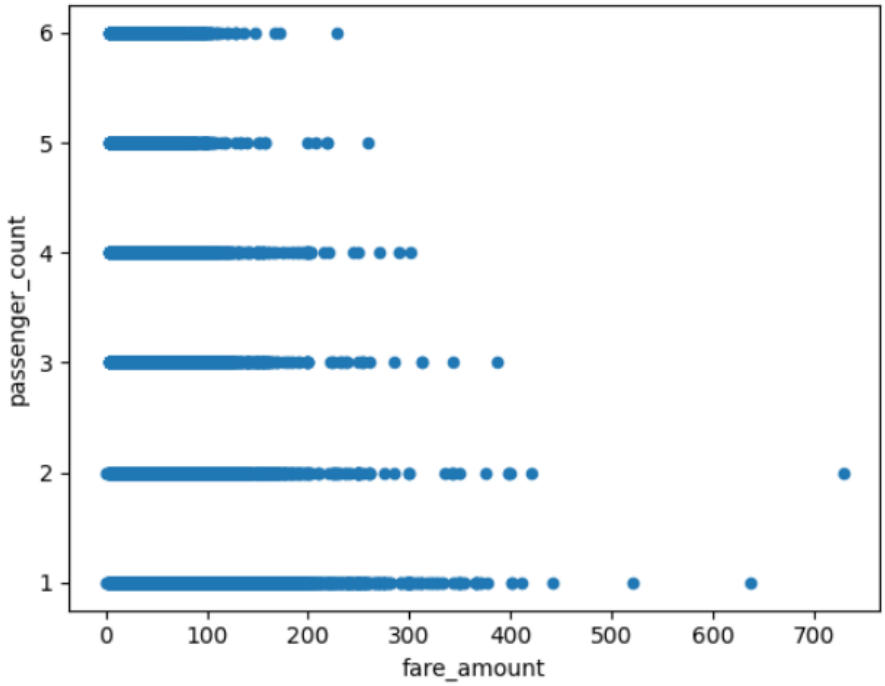
- Fare amount and trip distance: Shows positive correlation



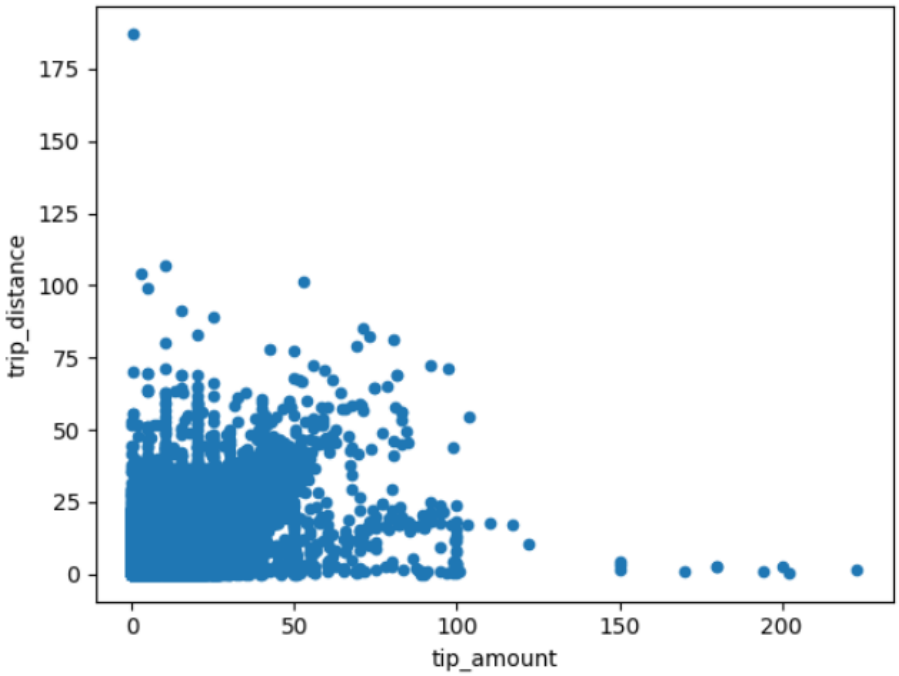
- Trip duration and fare: No correlation



- Passengers count and fare: No correlation



- Trip distance and tip: Positive correlation



3.1.7. Analyse the relationship between fare/tips and trips/passengers

Grouping by passenger_count, we computed:

- Average fare
- Average tip
- Total revenue contribution

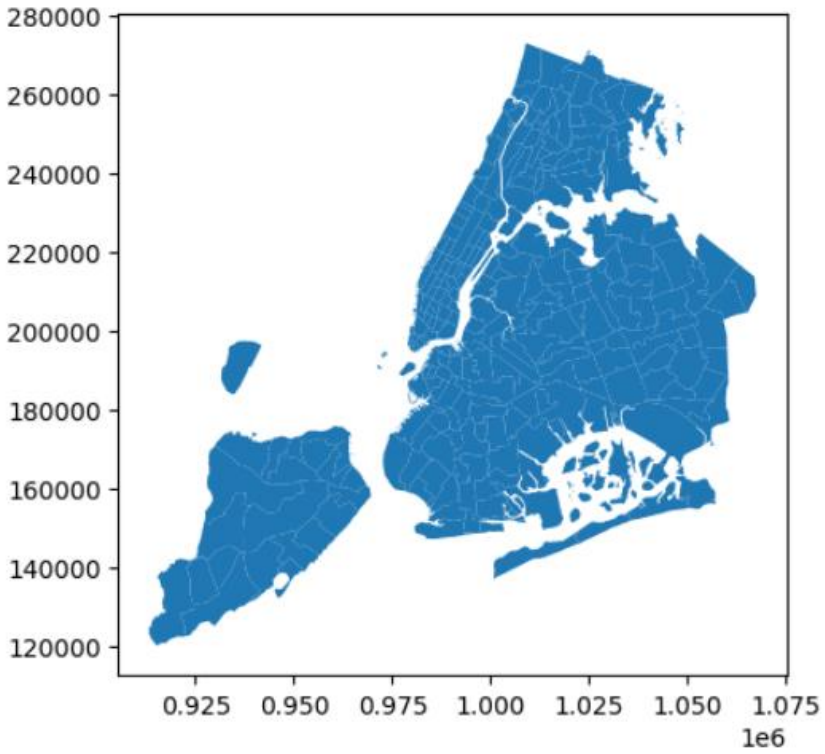
Multi-passenger trips show slightly different patterns compared to solo trips.

3.1.8. Analyse the distribution of different payment types

Credit card payments dominated both trip count and revenue.
Cash payments have lower tip percentages.

3.1.9. Load the taxi zones shapefile and display it

GeoPandas was used to load the official NYC Taxi Zone shapefile.
A simple boundary map was plotted to show geographic distribution.



3.1.10. Merge the zone data with trips data

Trip counts per zone were computed and merged back into the GeoDataFrame using LocationID.
This enabled spatial trip density visualisation.

3.1.11. Find the number of trips for each zone/location ID

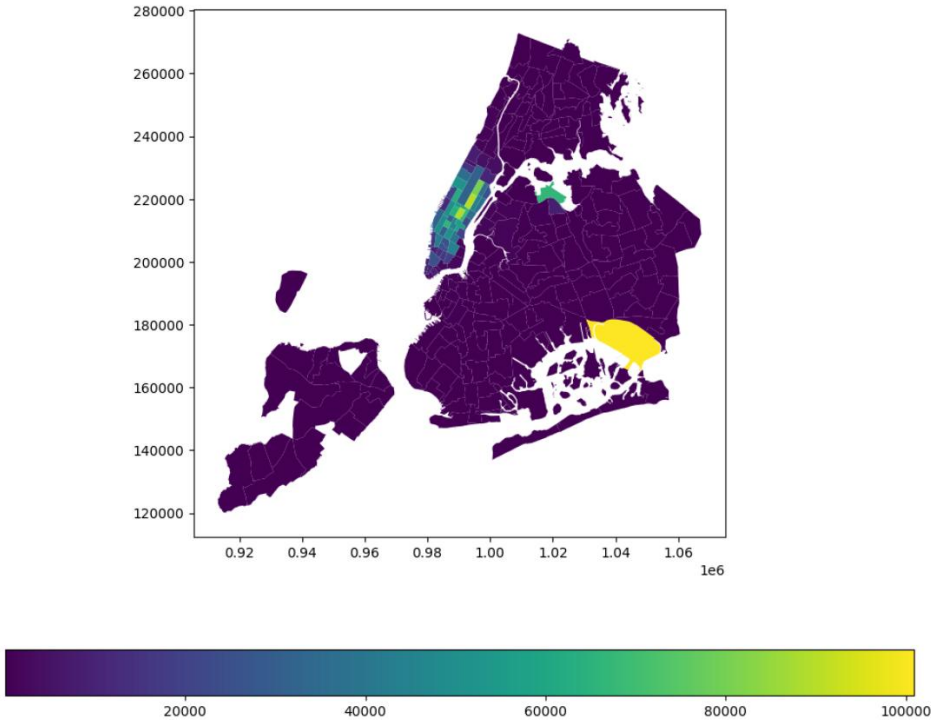
Trip counts were calculated separately for pickup and dropoff zones.
Spatial hotspots such as Midtown and major transit hubs were prominent.

3.1.12. Add the number of trips for each zone to the zones dataframe

Trip counts were merged into zones as new columns, enabling mapping and deeper zone-wise analysis.

3.1.13. Plot a map of the zones showing number of trips

A choropleth map was generated, revealing high-traffic zones geographically.



3.1.14. Conclude with results

Findings reinforced:

- High variability of demand across time
- High concentration of trips in specific zones
- Clear relationships between trip characteristics, timing, and revenues

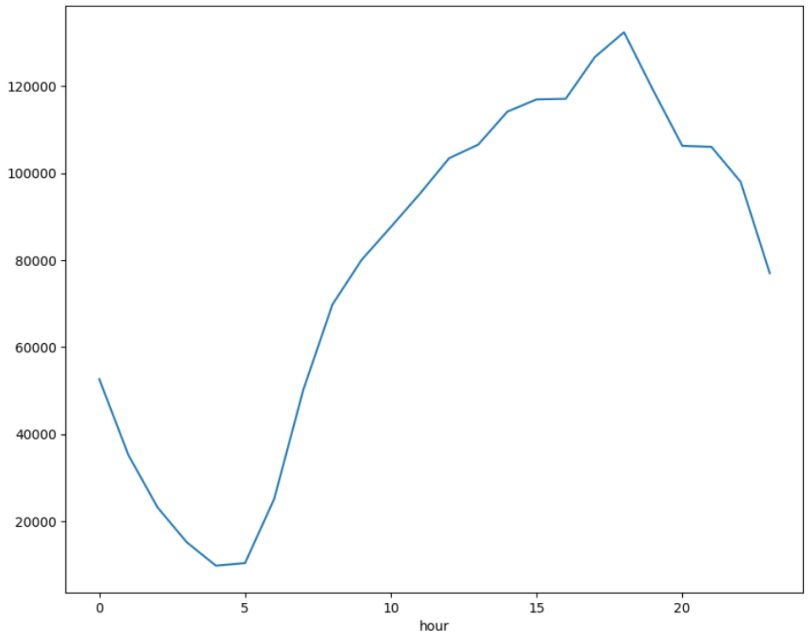
3.2. Detailed EDA: Insights and Strategies

3.2.1. Identify slow routes by comparing average speeds on different routes

Speed = trip_distance / trip_duration
Grouping by origin–destination pair revealed consistently slow routes (high congestion areas).

3.2.2. Calculate the hourly number of trips and identify the busy hours

Hourly counts identified clear peak hours.
Scaling based on sampling fraction gave estimated true hourly trip volumes.



18 p.m. is the busiest hour of the day.

3.2.3. Scale up the number of trips from above to find the actual number of trips

We scaled up no. of trips by dividing at by sampling rate of 0.5.

3.2.4. Compare hourly traffic on weekdays and weekends

Weekdays show commute-based peaks; weekends show stronger late-night demand.

3.2.5. Identify the top 10 zones with high hourly pickups and drops

Grouping by (zone, hour) identified the busiest operational hotspots.

3.2.6. Find the ratio of pickups and dropoffs in each zone

The pickup-to-dropoff ratio highlighted zones requiring rebalancing:

- Pickup-heavy (source zones)
- Dropoff-heavy (sink zones)

3.2.7. Identify the top zones with high traffic during night hours

Trips between 23:00–05:00 were analysed.
Night hotspots aligned with nightlife, airports, and transportation hubs.

3.2.8. Find the revenue share for nighttime and daytime hours

Night hours contributed a meaningful share of total revenue, further supporting night-focused routing strategies.

3.2.9. For the different passenger counts, find the average fare per mile per passenger

fare_per_mile_per_passenger showed how cost per person varies across group sizes.

3.2.10. Find the average fare per mile by hours of the day and by days of the week

Patterns showed time-of-day influence on fare intensity, likely reflecting congestion or rate-code differences.

3.2.11. Analyse the average fare per mile for the different vendors

Vendor-level comparisons found small pricing variations and differences in short vs long trip pricing patterns.

3.2.12. Compare the fare rates of different vendors in a distance-tiered fashion

Distance buckets (0–2, 2–5, >5 miles) helped compare vendor behaviour for different trip lengths.

3.2.13. Analyse the tip percentages

Tip percentages were analysed by:

- Trip distance
- Passenger count
- Hour of day

Meaningful differences emerged between low-tip (<10%) and high-tip (>25%) trips.

3.2.14. Analyse the trends in passenger count

Average passengers varied across hours and days, with slight increases during commute and event-heavy times.

3.2.15. Analyse the variation of passenger counts across zones

Average passenger count mapped by zone showed transport hubs and event-heavy areas having slightly higher passenger numbers.

4. Conclusions

4.1. Final Insights and Recommendations

- **Recommendations to optimize routing and dispatching based on demand patterns and operational inefficiencies.**

  Taxi demand in NYC follows clear patterns, and adjusting routing based on these trends can make operations more efficient. More taxis should be available during peak hours and in high-demand zones, while quieter periods can be managed with a smaller fleet. Routes that consistently show slow speeds may need alternative paths or different timing. Zones that regularly receive more dropoffs than pickups should be refilled by moving idle vehicles from nearby areas. Night-time demand, especially around airports, major transit hubs, and nightlife areas, is also significant and should be supported with steady cab availability.

4.1.1. Suggestions on strategically positioning cabs across different zones to make best use of insights uncovered by analysing trip trends across time, days and months.

Certain zones—like Midtown, airports, and busy stations—act as major pickup and dropoff hotspots and should always have enough taxis. Medium-demand zones work well with a rotational approach where

vehicles shift in based on recent trip activity. Events such as concerts or sports games create short-term spikes, so placing cabs nearby ahead of time can reduce wait times. Low-demand outer areas still need coverage, and providing small incentives during off-peak hours can help keep service reliable there.

4.1.2. Propose data-driven adjustments to the pricing strategy to maximize revenue while maintaining competitive rates with other vendors.

A tiered pricing approach can better match trip patterns. Short trips can keep a slightly higher base fare, while medium and longer trips can benefit from a lower per-mile rate. Small surcharges during busy hours or in congested central zones are reasonable, as long as they remain transparent. Ensuring pricing works similarly across vendors helps avoid confusion for riders. Since tipping patterns vary by time and trip type, keeping fares reasonable and encouraging tipping through reminders can help drivers earn more without increasing base prices too much.