Data Cleaning and Analysis Report

NYC TLC Taxi Trips 2019
Normalize & Aggregate the Trips by Time and Area

AIT2006-4-1.1

I. Introduction

This document presents the final project of group **AIT2006-4-1.1** for the course Programming for Data Processing.

The objective of our group is to transform, normalize, and aggregate the raw dataset of yellow taxi trips in New York City. Through this process, we aim to visualize key metrics and evaluate relevant KPIs, enabling us to develop well-supported insights and predictive models for future trends in taxi trip data.

This project is completed by the following members of Group **AIT2006-4-1.1**:
- **Tạ Mạnh Cường** - Student ID: 24022278
- **Nguyễn Minh Huy** - Student ID: 24022356
- **Nguyễn Viết Hùng** - Student ID: 24022344

Date of Last Commit: 19 December 2025
Commit Hash Used for Assessment: *6f2b5efb16455b42bf13616c3c7d298b18e68c22*
Git Tag: *final-submission*

II. Outline

The data analysis process in this project consists of 5 steps:
1. **Data Acquisition:** Collecting raw trip data from TLC and organizing folder structure for systematic storage.
2. **Data Processing:** Removing outliers, handling missing values, standardizing formats.
3. **Data Aggregation and KPI:** Combining relevant variables and computing key performance indicators.
4. **Data Visualization:** Visualizing the data to identify patterns, trends and anomalies.
5. **Modelling:** Developing predictive models to forecast daily trip demand using historical data.

In this project, we primarily use Python to clean data and prepare for EDA. Additionally, we will be applying basic statistics to find the relationship between data and utilize Matplotlib for better visualization.

III. Data Acquisition

The project investigates trip records in 2019 from New York City Yellow Taxis, sourced from the **Taxi & Limousine Commission (TLC) Website**.

These publicly available datasets contain detailed information about taxi operations, including pick-up and drop-off times and location ID, trip distances, fares, tips and passenger counts, etc.

Because we are working with large-scale datasets, we prefer using Parquet files instead of CSV format. This enables both quicker data access and more optimized data storage.

Given that loading the entire year of data into memory is impractical due to its size (around 10 GB), an iterative, chunk-based processing approach is required, handling one monthly parquet file at a time.

IV. Data Preprocessing

Once the correct dataset has been acquired, the next stage is data preprocessing, a phase that includes cleaning, integrating, transforming, and reducing the data.

4.1. Memory Optimization and Data Normalization

To optimize memory, numeric columns were downcast from float64 to float32 to reduce RAM usage, and a garbage collector (*gc.collect()*) was utilized after every iteration to prevent memory leaks.

We also standardize the date time format for *tpep_pickup_datetime* and *tpep_dropoff_datetime* columns to make them easier to work with.

4.2. Calculating Additional Metrics

To support further analysis and visualization, we derived two additional columns from the dataset:

- *trip_duration*: computed as the difference between *tpep_dropoff_datetime* and *tpep_pickup_datetime.*
- *avg_speed*: computed as *trip_distance* divided by *trip_duration.* (This leads to the appearance of more missing values since *trip_duration* can be zero).

4.3. Handling Missing Values

Missing data occurred very commonly in large-scale datasets due to factors such as signal loss, or manual entry errors by humans. In this particular dataset, missing values appear in: *passenger_count, RatecodeID, store_and_fwd_flag, congestion_surcharge, airport_fee,* and *avg_speed*.

The “dropping all rows with NaNs” was not feasible, as it would have discarded many valid trips that were merely missing minor fields (e.g., a missing surcharge). Instead, we decided to fill these missing values with reasonable defaults to preserve data integrity.

The following mapping outlines how each missing value was handled:

Column	Fill Value	Rationale
passenger_count	1	Missing passenger counts are treated as single-passenger trips.
RatecodeID	1	According to the TLC data dictionary, ‘1’ represents the standard rate.
store_and_fwd_flag	N	‘N’ indicates that the trip was not a store-and-forward trip.
congestion_surcharge	0	Missing financial data is treated as ‘0’, assuming the charge/fee was not applied.
airport_fee	0	
avg_speed	0	Defaulting to ‘0’ for inf or NaN values.

4.4. Data Cleaning Logic

We applied a strict set of **Quality Assurance (QA)** rules. Trips failing one or more of the following conditions were flagged as *is_valid_trip = False* and removed from further analysis:

Rule	Rationale
pickup_datetime ≥ current_month AND dropoff_datetime < next_month	Ensures the trip belongs to the correct month and year.
trip_duration > 0 AND < 600	Removes trips with impossible durations, including drop-off-before-pickup errors or durations longer than 10 hours.
(trip_distance > 0) OR (trip_distance = 0 AND total_amount > 0)	Excludes trips with negative distances but retains legitimate zero-distance trips (e.g., flat-fare scenarios).
avg_speed ≥ 0 AND ≤ 70	Assumes taxis operate with realistic speeds (within 70 miles per hour).

payment_type ≥ 0 AND ≤ 4	Validates payment types according to the TLC data dictionary (0–4).
total_amount > 0 AND < 1000	Remove negative fares and extreme outliers
tip_amount ≤ total_amount	Ensures tips are less than 100% of the total bill
fare_amount ≥ 0 AND extra ≥ 0 AND mta_tax ≥ 0 AND improvement_surcharge ≥ 0 AND tolls_amount ≥ 0 AND congestion_surcharge ≥ 0 AND airport_fee ≥ 0	Every monetary component must be non-negative values
PULocationID > 0 AND ≤ 263	Valid pick-up locations in New York City based on the taxi_zone_lookup table
DOLocationID > 0 AND ≤ 265	Valid drop-off locations based on the taxi_zone_lookup table
RatecodeID > 0 AND ≤ 6	RatecodeID must fall within 1-6 according to the data dictionary
passenger_count ≥ 0 AND ≤ 9	Valid passenger counts (zero values are retained for possible delivery purposes or application/human errors).

4.5. QA Results Summary

After applying the QA rules and removing all duplicate rows, the final data metrics are as follow:

- **Total raw rows:** ~84,600,000
- **Total clean rows:** ~82,950,000
- **Total rows dropped:** ~1,650,000
- **Overall drop rate:** ~2%

The resulting drop percentage is acceptable and allows us to proceed to the next step, which is Data Aggregation.

V. Data Aggregation

The objective of this phase is to transform this detailed data into higher level strategic metrics **(KPIs)**. These metrics enable us to develop insights into seasonal trends, revenue performance, and operational efficiency on hourly, daily, weekly, and monthly scales.

The following Key Performance Indicators were calculated using the Pandas *.agg()* method and made consistent across different time scales:

Name	Definition	Formula	Unit
trips	Number of trips	('VendorID', 'count')	trip
duration_p50	Median trip duration	('trip_duration', 'median')	minute
duration_p95	95th percentile of trip duration	('trip_duration', lambda x: x.quantile(0.95))	minute
duration_mean	Mean trip duration	('trip_duration', 'mean')	mph
speed_50	Median average speed	('avg_speed', 'median')	mph
speed_mean	Mean average speed	('avg_speed', 'mean')	mph
total_money	Total revenue generated	('total_amount', 'sum')	dollar
passenger_mean	Mean passenger count per trip	('passenger_count', 'mean')	person
passenger_sum	Total number of passengers	('passenger_count', 'sum')	person
distance_sum	Total distance traveled	('trip_distance', 'sum')	mile
distance_p50	Median trip distance	('trip_distance', 'median')	mile
distance_mean	Mean trip distance	('trip_distance', 'mean')	mile

Because the data are processed on a per-month basis, additional aggregation is required for the hourly and weekly datasets. Weekly periods may span across two calendar months, while hourly aggregation is performed by day of the week, which can introduce duplicate entries when monthly aggregates are concatenated.

As a result, a secondary aggregation step is applied to consolidate these overlaps. This approach may introduce minor approximation errors; however, such deviations are considered acceptable given the scope of this project.

The aggregation process produces seven aggregated datasets:

- Four datasets summarize the KPIs at hourly, daily, weekly, and monthly temporal scales.
- In addition, three separate monthly-level datasets are generated to capture distributions by payment type, pickup location, and drop-off location.

VI. Data Visualization

After data cleaning and aggregation, the workflow proceeds to **data visualization**. Common Python visualization libraries, including **Matplotlib** and **Seaborn**, are used to generate the plots.

In this study, more than **five plots** are generated to support exploratory analysis and interpretation. Therefore, visualization is conducted using **Jupyter Notebook**, with each plot implemented in a separate notebook cell to ensure clarity and reproducibility.

6.1. Daily Revenue and Number of Trips

First, we examine the daily revenue fluctuations throughout the year:

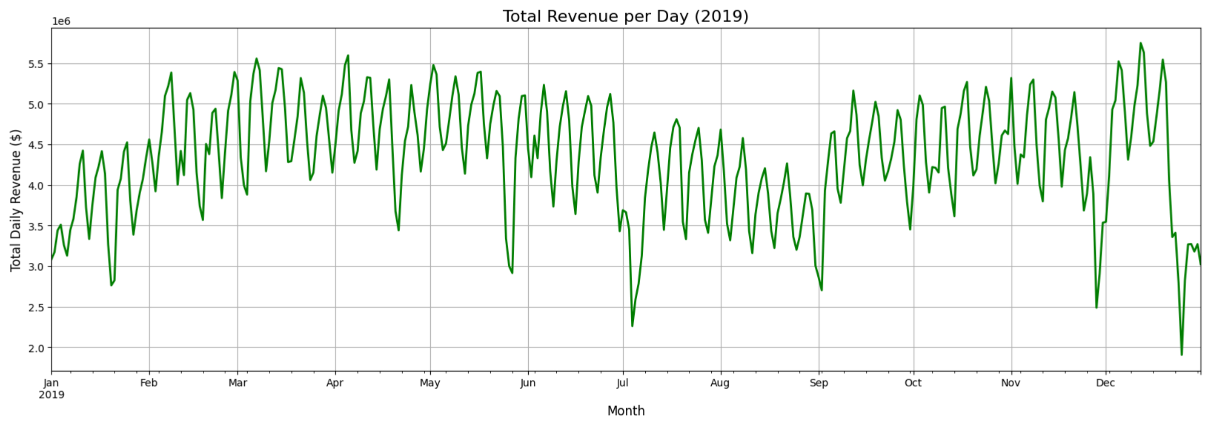

Figure 1. Daily Total Revenue in 2019

Figure 1 displays a time series of daily revenue for the entire year of 2019. The y-axis represents revenue in millions of dollars, while the x-axis represents time by month.

The most prominent feature of the graph is the consistent, high-frequency **"sawtooth"** pattern. This indicates a strong weekly cycle where revenue peaks on business days of the week and noticeably dips on weekends.

The line graph also shows sharp declines during major US holidays, most significantly on **Independence Day (July 4th)**, **Thanksgiving (November 28th)**, and **Christmas (December 25th)**.
Among these, **December** exhibits the highest volatility. A substantial surge occurs near the end of the year, reaching the **annual maximum** of $5.7 million (likely a pre-holiday rush), followed immediately by a steep drop to the **lowest point of the year** (below $2.0 million), corresponding to Christmas Day.

Next, we analyze the corresponding daily trip volume to assess whether similar temporal patterns are observed:

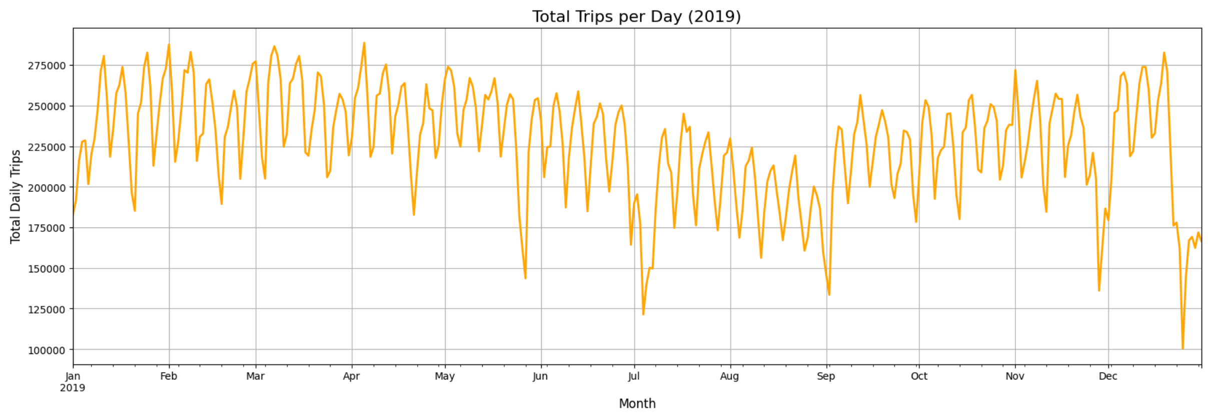

Figure 2. Daily Total Trips in 2019

Figure 2 tracks the volume of daily taxi trips throughout 2019. The y-axis shows volume ranging from roughly 100,000 to 290,000 trips per day.

We can observe that daily revenue and trip volume track closely throughout the year, exhibiting a **strong correlation**. This relationship suggests that overall revenue fluctuations are primarily driven by changes in demand rather than variations in pricing or trip characteristics.

The alignment in figure 3 between these two metrics further reinforces the presence of stable underlying travel patterns across the year:

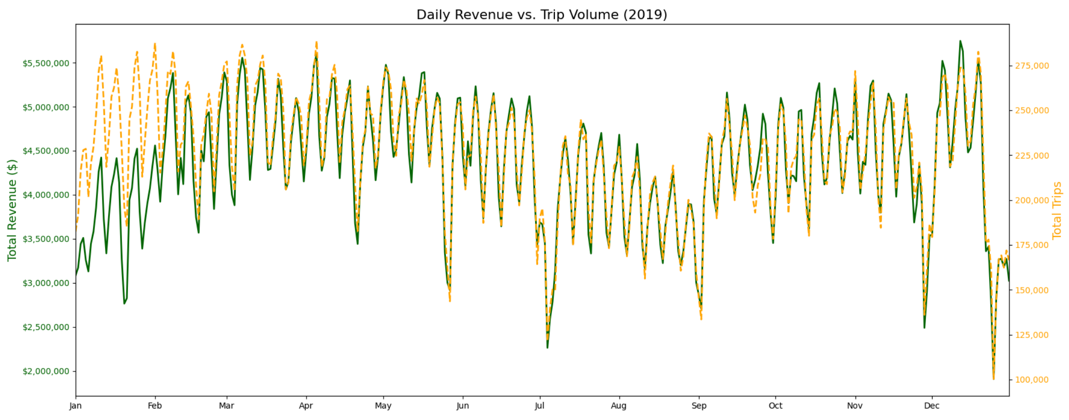

Figure 3. Daily Revenue vs. Trip Volume in 2019

However, a notable divergence occurs in **January**, where revenue is disproportionately low compared to trip volume. This suggests a lower average transaction value (fare) during the post-holiday season.
In contrast, **December** exhibits the opposite behavior. The system generated more money despite handling fewer passengers than in the Spring.

By visually comparing the peaks, we can estimate the surge:

- **Spring Peak:** ~290k Trips & ~$5.5M Revenue → ~$19.00 per trip
- **Holiday Peak:** ~275k Trips & ~$5.7M Revenue → ~$20.70 per trip

In summary, daily revenue and trip volume in 2019 exhibit strong and consistent temporal patterns, including a clear weekly cycle and sharp declines during major U.S. holidays. The two metrics closely track each other, indicating that revenue fluctuations are largely driven by changes in travel demand.

Seasonal differences are also evident. January shows lower revenue relative to trip volume, suggesting reduced average fares, while December generates higher revenue despite fewer trips, reflecting increased transaction values during the holiday period.

6.2. Trip Volume by Payment Type

Figure 4 illustrates payment method usage across 2019. The y-axis shows the number of trips in the **Logarithmic Scale** (powers of 10), while the x-axis represents time by month.

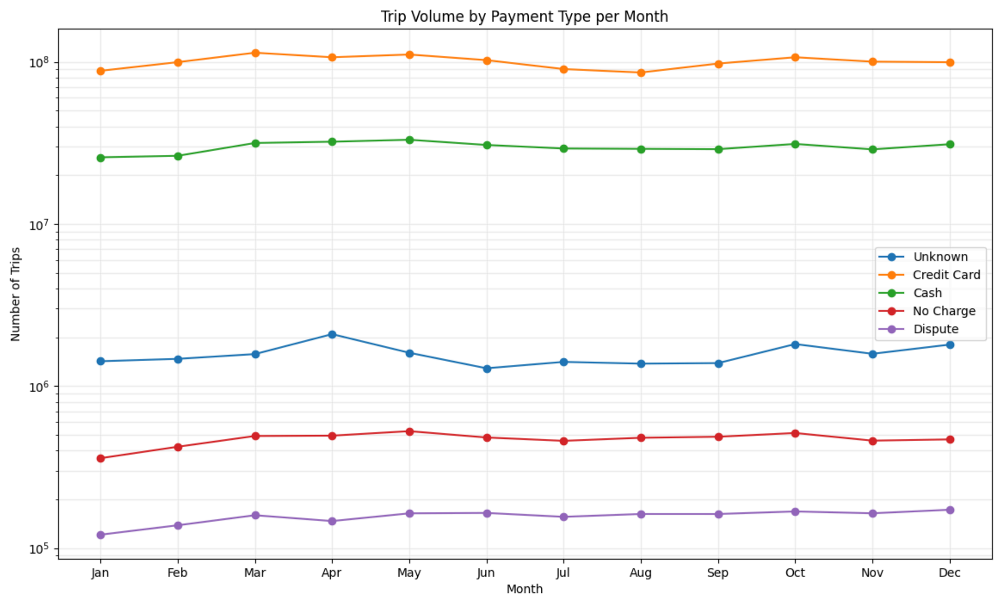

Figure 4. Trip Volume by Payment Type in 2019

The graph clearly shows a stable hierarchy of how passengers pay for their rides:

- **Credit Card:** This is by far the most dominant payment method. It sits in the 10^8 mark (100 million trips per month), indicating it is the standard for the vast majority of trips.

- **Cash:** Cash is the second most common option, but it is significantly lower than credit card.
- **Unknown, No Charge & Dispute:** Only a small fraction of trips have these payment types.

The most important insight from this graph is how **flat** the lines are. There is no shift in payment preference throughout the whole year.

Figure 5 highlights the substantial gap in usage between **credit card** payments and **cash**, as well as other payment types:

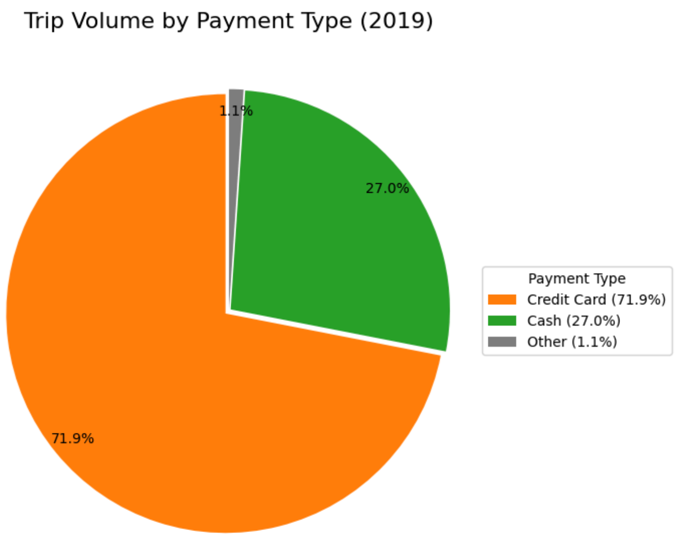

Figure 5. Trip Volume by Payment Type in 2019 (Pie Chart)

Credit card payments account for approximately three-quarters of all trips, while **cash** payments make up the remaining one-quarter, making cash the secondary payment method. **Other** payment types represent only a negligible share of total trips and have minimal impact on overall payment patterns.

6.3. Top Pick-up and Drop-off Zones by Trip Volume

To explore where trips are distributed, we analyze the relative frequency of **pick-up and drop-off zones** in the entire year. A bar chart is used to rank locations by trip volume, accompanied by a line indicating the cumulative percentage across zones.

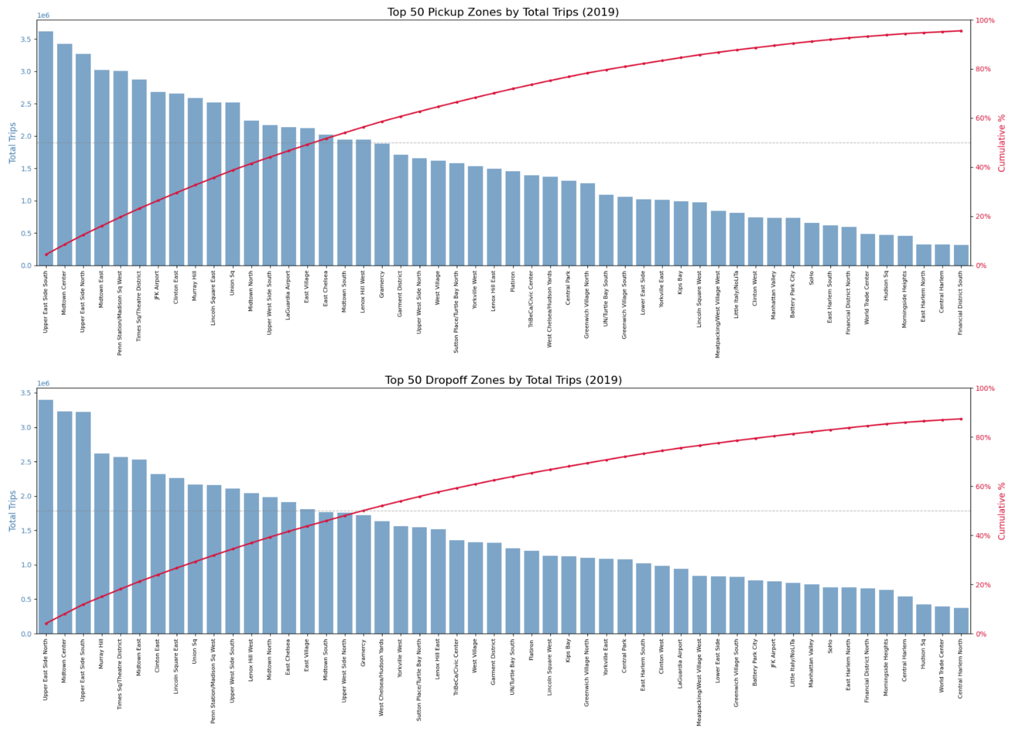

Figure 6. Top 50 Pickup and Dropoff Zones by Trip Volume (2019)

The tall bars on the left represent a small number of zones that generate the vast majority of all trips, while the right ones represent zones in the outer boroughs (Bronx, far Brooklyn/Queens) that generate almost negligible traffic, closely resembling a **Pareto** distribution.

- **Top Zones: Upper East Side (South & North), Times Square** and **Midtown Center/ East** consistently rank among the highest volume locations for both pick-ups and drop-offs.
- **Cumulative Impact:** As shown by the cumulative percentage line (red), the **top ~20 zones** account for **50% of all trips**, while the remaining hundreds of zones share the other half. This highlights the extreme density of taxi operations in Manhattan.

6.4. Congestion Analysis

Figure 7 visualizes the **average speed** of vehicles in NYC over a 24-hour cycle.

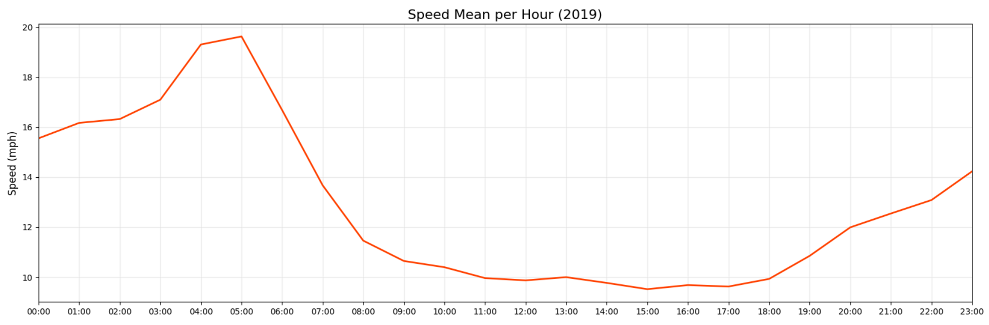

Figure 7. Speed Mean per Hour (2019)

The y-axis shows speed in miles per hour and the x-axis is time of day.

The graph displays an inverse relationship between the time of day and traffic speed, correlating directly with typical human activity and congestion patterns.

From 00:00 to 05:00, traffic speeds are at their **highest**, with the maximum of approximately 20 mph at 5 AM.

As the city wakes up, there is a drastic and rapid decline in speed. **Between 5 AM and 9 AM**, the average speed nearly halves, dropping from ~19.5 mph to roughly 10.5 mph. This indicates the **start of heavy congestion** during the morning commute.

During **typical business hours**, the average speed remains **consistently low**, fluctuating slightly between 9.5 mph and 10.5 mph. The slowest point of the day appears to be around 15:00 (3 PM), where speeds dip to approximately 9.5 mph.

Starting at 18:00 (6 PM), speeds begin to **slowly recover** as the workday ends and rush hour concludes. The recovery is more gradual than the morning drop. By 23:00, speeds had risen back to roughly 14 mph.

6.5. Temporal Revenue Efficiency

To answer the question **“When is the most profitable time of day?”**, we plotted a heatmap describing the relationship between Revenue per Trip and Number of Trips by hour of the week.

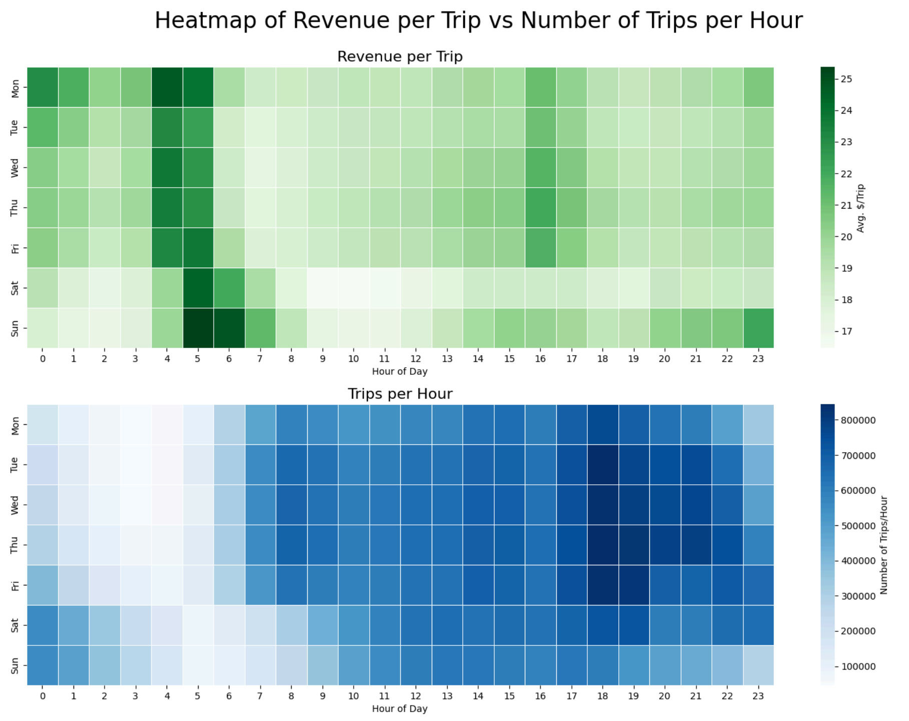

Figure 8. Heatmap of Revenue per Trip vs. Number of Trips per Hour (2019)

Figure 8 reveals an **inverse relationship** between **trip volume** and **revenue per trip**. During the early morning hours, **lower demand** is associated with **higher average revenue per trip**. This pattern suggests that trips occurring during these hours tend to be longer or more costly, despite reduced overall activity.

- **From 4 AM to 6 AM:** During these hours, the average fare spikes to $23–$25+, compared to the daytime average of $17–$19.

This strongly correlates with early morning airport runs (JFK/LaGuardia). At this hour, local commuter traffic is non-existent, so most passengers are likely travelers heading to airports for morning flights. These are long-distance, high-value trips.

- **From 6 PM to 8 PM:** This is the standard evening rush hour.

While the number of trips is massive (exceeding 700,000 trips), the revenue per trip (from the green chart) is relatively average. This implies lots of short trips as people commute home or go to dinner.

When considered alongside previous results, these findings indicate that **5 AM** is the **most efficient time** for drivers. They drive faster and make more money per trip.

VII. Modelling

7. 1. Predictive Models

1. Objective

This study focuses on predicting **daily NYC Yellow Taxi demand** for **January and February 2020.**
Several forecasting approaches are evaluated, ranging from simple baselines to more advanced time-series models. Model performance is assessed using Mean Absolute Percentage Error (MAPE) to ensure comparability across methods.

2. Baseline

The baseline model serves as a benchmark by assuming that **future demand mirrors historical patterns**. To generate the 60-day forecast, observed data from the previous year are repeated such that each forecasted date corresponds to the same calendar day in the prior year (e.g., January 1, 2020 is mapped to January 1, 2019).

- **Strength:** Performs reasonably well under stable, repetitive seasonal patterns.
- **Limitation:** Unable to capture trends, structural changes, or shifting holiday effects.

3. Linear Regression

We used the pre-built **Scikit-learn Linear Regression Model** for this approach.

- **Features:** Calendar variables (day of week, month), a binary holiday indicator, and a 7-day lag feature.
- **Strategy:** Forecasts are generated recursively, where each predicted value is fed back into the model as input for subsequent predictions.
- **Limitation:** Error accumulation over time and an inability to capture nonlinear seasonal trends.

4. ARIMA Model

The **SARIMAX Model** provided by **Statsmodels** is used for this approach.

- **Transformation:** A logarithmic transformation is applied to stabilize variance.
- **Seasonality:** Weekly patterns are modeled using a 7-day seasonal component.
- **Exogenous Variable:** Federal holidays are incorporated to account for demand disruptions.
- **Post-processing:** Predictions are transformed back to the original scale with a bias correction to approximate the expected mean.

5. Results

Several forecasting approaches are evaluated, ranging from simple baselines to more advanced time

To evaluate model performance, we visualized and compared the forecasts from each model against the observed demand during the first two months of 2020:

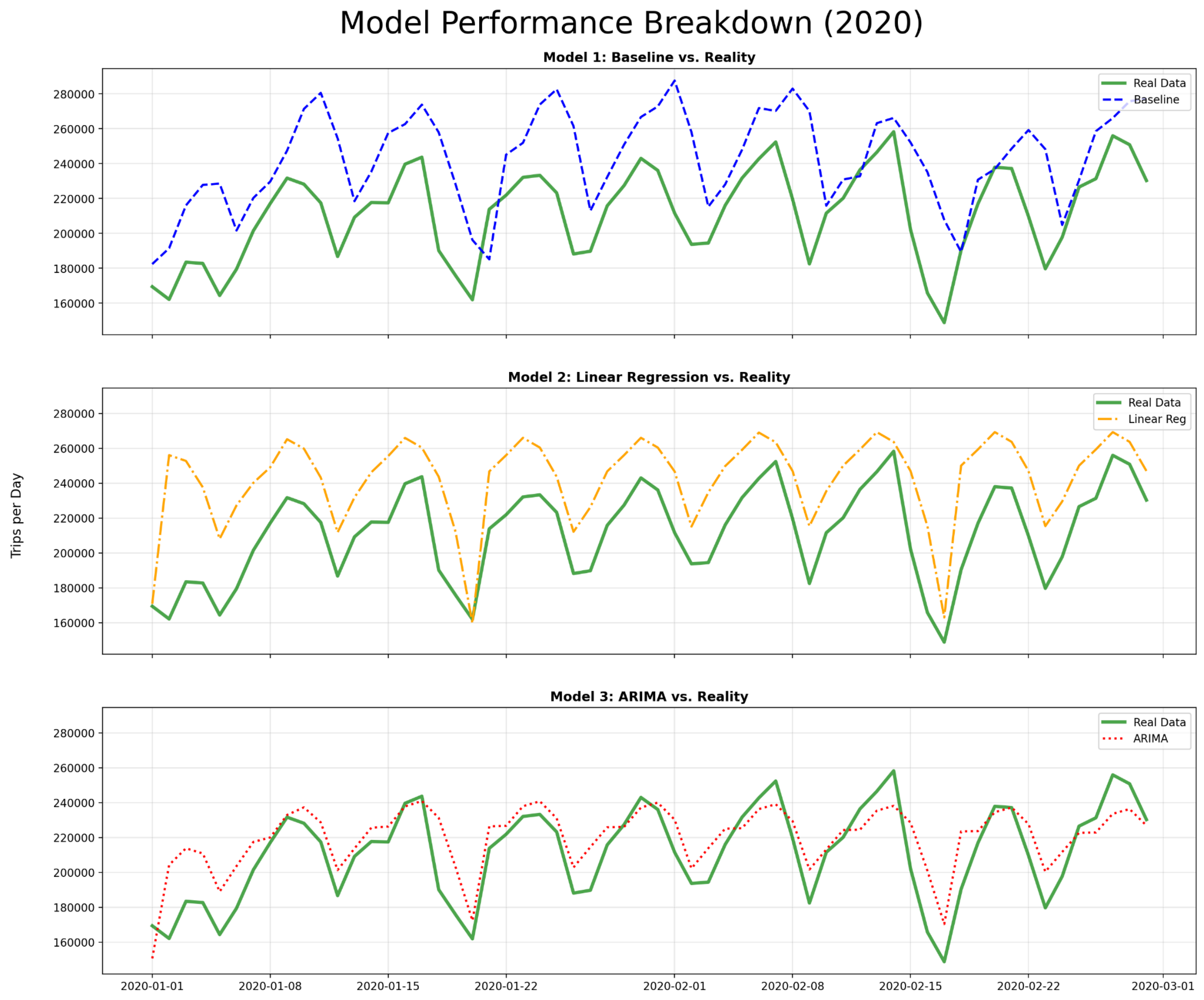

Figure 9. Model Performance Comparison for January and February 2020

Although all approaches successfully captured the overall trend of the dataset, **ARIMA** (red) clearly outperformed its counterparts. Both the **Baseline** (blue) and **Linear Regression** (orange) models suffer from the same major flaw: **systematic overestimation of demand.**

Error evaluation confirms that the **ARIMA model** delivers the strongest performance, achieving a Mean Absolute Percentage Error (MAPE) of ~7%, compared to ~15% for both Baseline and Linear Regression.

7. 2. Anomaly Detection

1. Objective

The objective of this section is to automatically detect outlier events in NYC taxi operations during 2019 in order to:

- **Identify impactful events:** Detect days or hours affected by external factors such as public holidays, snowstorms, major events, or traffic disruptions.
- **Monitor operational performance:** Identify time periods with poor operational efficiency (extremely low speeds) or abnormal revenue behavior (unusually high or low pricing).

2. Methodology

A **Contextual Z-score** statistical method is applied:

- Data is grouped by [Day of Week, Hour of Day]

(e.g., comparing this Monday at 8 AM with all other Mondays at 8 AM).

- **Z-score formula:**

$$Z = \frac{x - \mu}{\sigma}$$

where:

- ❖ x is the observed value
- ❖ μ is the historical mean for that specific time context
- ❖ σ is the standard deviation

3. Evaluation Criteria

A data point is classified as anomalous if the absolute Z-score exceeds a predefined threshold:

- Threshold: $|Z| > 3$

(Under the normal distribution, this occurs with probability < 0.3%, making it extremely rare.)

- Anomaly categories:
  - ❖ Low Demand: $Z(trips) < -3$
  - ❖ High Demand: $Z(trips) > 3$
  - ❖ Traffic Jam: $Z(speed) < -3$
  - ❖ High Price Rate: $Z(revenue\_per\_mile) > 3$

4. Representative Anomaly Cases

a. Case 1: Holiday Effect – Low Demand

- **Detection:** December 25, 2019 (Wednesday) at 08:00 AM
- **Metric:** Z(trips) ≈ −4.9
- **Interpretation:** The model flags this as anomalous because Wednesdays are typically high-demand days. In reality, this is Christmas Day, when most people stay home.
- **Conclusion:** The model successfully identifies holiday patterns without requiring a predefined holiday calendar.

b. Case 2: Weather or Incident Effect – Traffic Jam

- **Detection:** An evening hour between 20:00–21:00 in December
- **Metric:** Z(speed) ≈ −3.2
- **Interpretation:** Vehicle speeds dropped to an unusually low level, indicating severe congestion likely caused by an accident, roadwork, or a snowstorm.
- **Conclusion:** The system effectively flags periods of extreme operational inefficiency, signaling drivers to avoid these times or locations.

c. Case 3: Pricing Anomaly

- **Detection:** Several early morning periods between 04:00–05:00 AM
- **Metric:** Z(revenue_per_mile) ≈ 3.7
- **Interpretation:** These trips are predominantly airport runs (e.g., JFK), often involving fixed fares and night surcharges, combined with light traffic and shorter travel times. Alternatively, very short trips with minimum fare charges may inflate revenue per mile.
- **Conclusion:** The model correctly identifies structural pricing anomalies associated with specific travel patterns.

VIII. Conclusion

8. 1. Key Findings:

This study successfully transformed and analyzed large-scale NYC Yellow Taxi trip data from 2019, producing several important insights across operational, financial, and temporal fields.

- The data cleaning process was effective, removing only about 2% of invalid records while preserving data quality.
- Clear weekly and seasonal patterns were observed in both trip volume and revenue, with sharp declines during major U.S. holidays.
- Credit cards dominate payment methods, accounting for approximately 75% of all trips, with no significant change throughout the year.
- Taxi activity is highly concentrated in Manhattan, where a small number of zones generate the majority of trips.
- Congestion analysis shows that early morning hours offer the highest operational efficiency, combining faster speeds with higher revenue per trip.

8. 2. Next Steps:

Going forward, we plan to improve forecast accuracy by incorporating additional explanatory variables and using more advanced models, such as LSTM or Prophet.

IX. References

1. New York City Taxi & Limousine Commission Trip Record Data

2. Mô hình ARIMA trong Time Series by Pham Dinh Khanh

3. Python Libraries:
- matplotlib
- numpy
- pandas
- pyarrow
- seaborn
- jupyter notebook
- sklearn
- statsmodels