STAT 405: Final Report

By Ella Gruen, Anthony Pagas, John Chumlea, Shien Zhu and Samuel Negus

Introduction:

This project aims to determine which taxi vendor in New York City is the most time-efficient to take on any day of the week. Using a large dataset of NYC taxi trip records, we analyzed trip efficiency by calculating minutes per mile traveled across two different vendors. Through data cleaning, aggregation, and parallel computing, we aimed to uncover which vendor consistently offers the fastest rides across the city.

Data:

The dataset was sourced from Kaggle at https://www.kaggle.com/datasets/chilam/nyctaxis and includes 28.85 GB of trip records. Each record contains information such as pickup and dropoff times, trip distance, passenger count, and the vendor ID.

Key variables used in our analysis include pickup_datetime (timestamp of trip start), trip_distance (distance traveled in miles), vendor_id (identifier of taxi vendor), passenger_count, and dropoff_datetime (timestamp of trip end).

GitHub:

git clone https://github.com/jellylemonfish/stat405group4.git

Analysis:

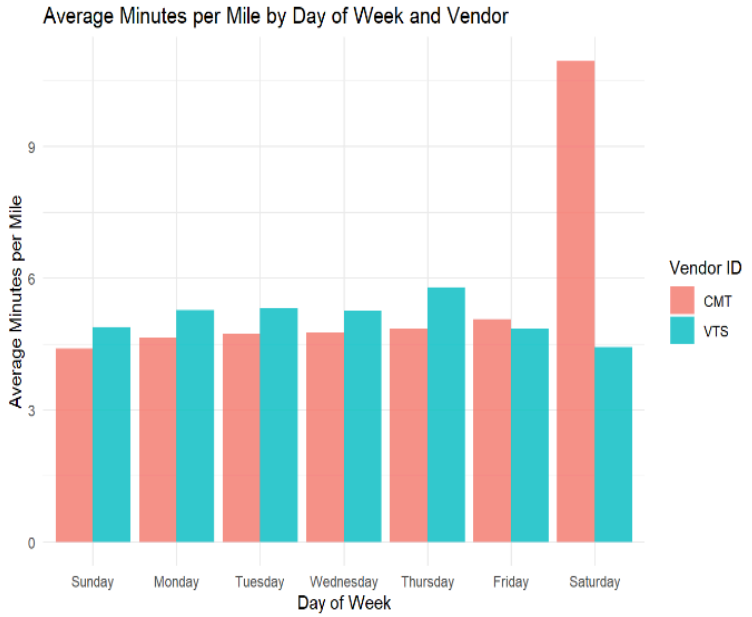
Before analysis, we cleaned the dataset to ensure accuracy. We dropped all entries with invalid, missing, or non-numeric trip distances. Then, we converted the pickup_datetime column into a proper datetime object, removing any entries with corrupt time formats. Finally, we removed trips with zero or negative trip distance or trip time, as they represent incorrect or incomplete records.

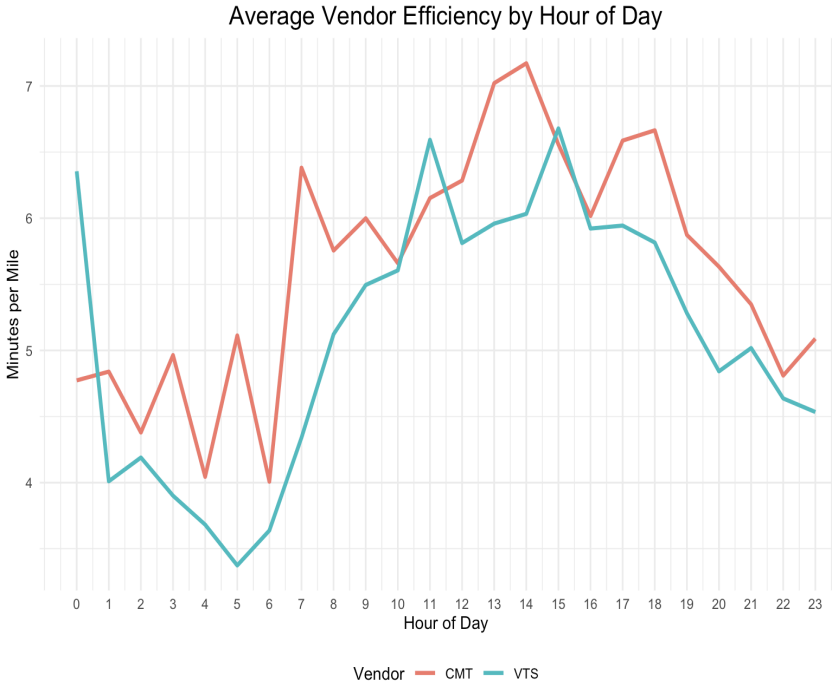
Statistical Computation Method:

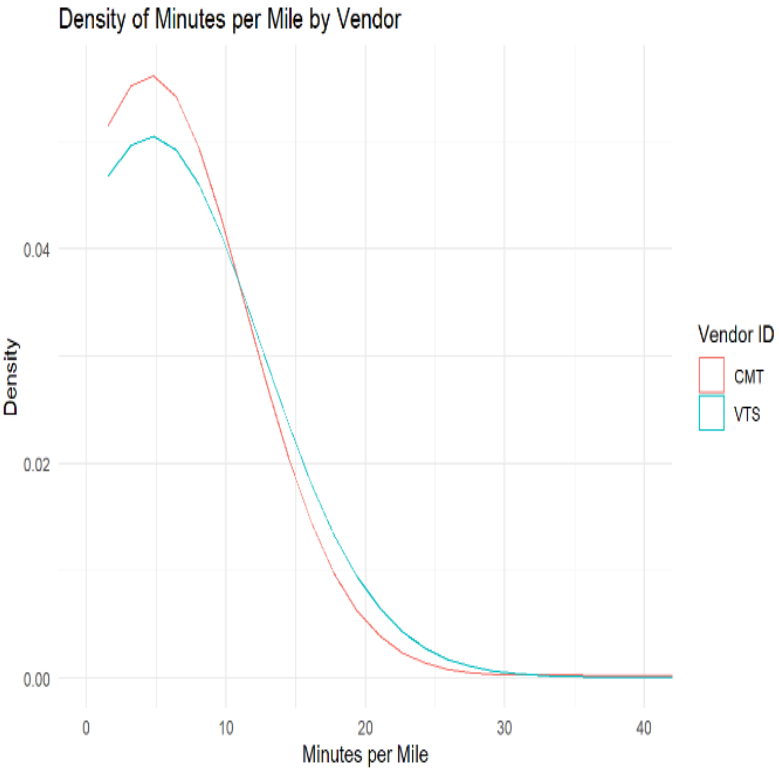
We performed group-wise aggregation to compute mean efficiency. To start, we grouped trip data by vendor ID and day of the week. Then, we calculated the mean minutes per mile within each group. (For example, we found that on a Tuesday around 12 AM, CMT taxis traveling between 0–1 mile averaged about 5.39 minutes per mile). We also used distance binning to prevent bias from low-sample size data points (e.g., if a vendor had very few trips at a given time/distance). Distance bins included 0–1 mile, 1–3 miles, 3–6 miles, 6-10 miles, and 10+ miles. This helped ensure fair comparisons by accounting for trip length.

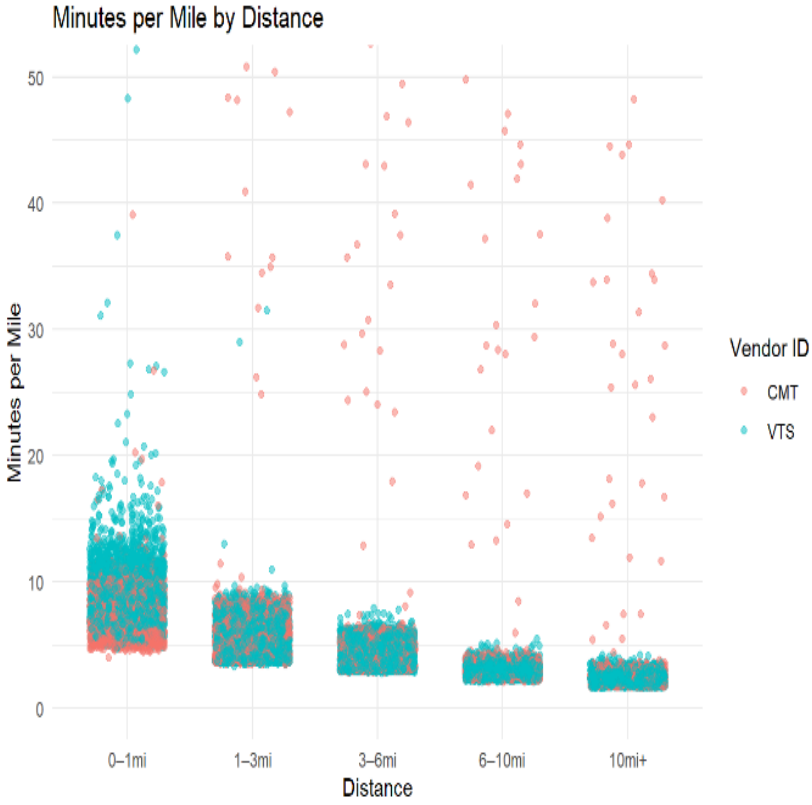
Parallel Computing:

To process this large dataset efficiently, we divided the computation into 12 parallel jobs. Each job requested 2 GB of disk and memory and 1 CPU, and job run times ranged from 10 to 20 minutes. This parallelization was critical due to the size of the dataset and frequent disk space constraints; individual files were still very large even after being split. Despite efficient job management, we faced challenges with disk s[illegible]









Conclusion:

Based on our analysis, VTS was the most time-efficient overall, averaging about 5 minutes and 6 seconds per mile. Interestingly, while CMT taxis had a lower mean minutes per mile than VTS on 5 out of 7 days when analyzed by day of the week, VTS consistently outperformed CMT across almost all distance bins except for trips between 0–1 mile. When looking at minutes per mile over each hour of the day, it is observed that VTS is faster than CMT at most times of the day.

This suggests that although daily averages may occasionally favor CMT, across a wider range of trip distances and times, VTS taxis provide a faster ride overall. For future work, expanding the analysis to account for traffic patterns by time of day or borough-specific performance could provide more in-depth insights into vendor efficiency.

Contributions:

Everyone: Met outside of class in person and virtually, maintained communication, and presented our findings.

Ella Gruen: Created presentation outline, debugged Condor job submissions, wrote report.

Anthony Pagas: Found the data set, made the R script, cleaned data, and made the foundation of our final presentation.

John Chumlea: Wrote code and attempted Condor jobs for original data set, created 3/4ths of the visualizations.

Shien Zhu: Loaded the data, assisted in coding, wrote the Condor parallel job scripts, got and combined the final result, created and managed GitHub version control.

Samuel Negus: Converted proposal to R markdown file for submission, assisted in loading data and parallel processing, debugging, wrote code and created visualizations.