

# 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

#### 1.1.1 Sample the data and combine the files

A sample of 500,000 records was initially taken from each monthly Parquet file. The samples were then refined to create a combined DataFrame with approximately 1.89 million rows, ensuring both efficiency and balanced representation across time.

## 2. Data Cleaning

### 2.1 Fixing Columns


To ensure consistency, column names were standardized by removing extra spaces and applying uniform formatting

#### 2.1.1 Combine the two airport\_fee columns

The dataset had two similar columns named **airport\_fee** and **Airport\_fee**, likely due to inconsistent naming across files. To fix this, I created a new column called **combined\_airport\_fee** that takes the higher value from the two for each row, ensuring no data was lost. Then, I removed the original columns to avoid duplication

### 2.2 Handling Missing Values

#### 2.2.1 Find the proportion of missing values in each column



	0
VendorID	0.000000
tpep_pickup_datetime	0.000000
tpep_dropoff_datetime	0.000000
passenger_count	3.420903
trip_distance	0.000000
RatecodeID	3.420903
PULocationID	0.000000
DOLocationID	0.000000
payment_type	0.000000
fare_amount	0.000000
extra	0.000000
mta_tax	0.000000
tip_amount	0.000000
tolls_amount	0.000000
improvement_surcharge	0.000000
total_amount	0.000000
congestion_surcharge	3.420903
combined_airport_fee	3.420903

dtype: float64

## 2.2.2 Handling missing values in passenger\_count

For the **passenger\_count** column, I filled in the missing values using the **mode**, which is the value that appears most often. Since **passenger\_count** is a categorical number and most trips usually have just one passenger, this method makes sense. It helps keep the data realistic without changing the overall pattern.

## 2.2.3 Handle missing values in RatecodeID

Missing values in the **RatecodeID** column were filled using the **mode**, since it's a categorical field. This method keeps the most common category and avoids the impact of rare or unusual values, helping to maintain the overall pattern in the data.

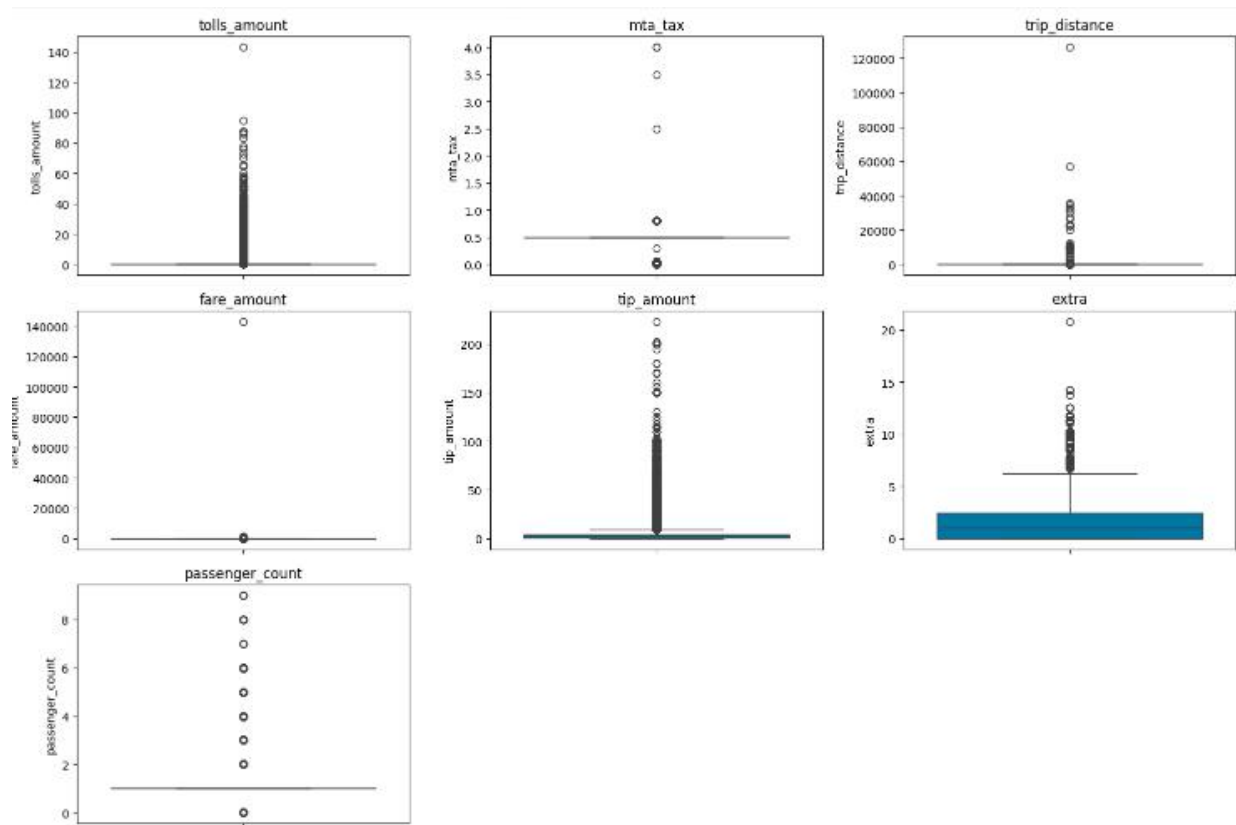
## 2.2.4 Impute NaN in congestion\_surcharge

I filled the missing values in the **congestion\_surcharge** column using the **median**. This helps avoid the effect of unusually high or low values and keeps the overall data pattern more accurate.

## 2.3 Handling Outliers and Standardising Values

To improve data quality and ensure meaningful analysis, outliers were identified and removed based on domain knowledge and logical constraints.

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



## 3. Exploratory Data Analysis

### 3.1 General EDA: Finding Patterns and Trends

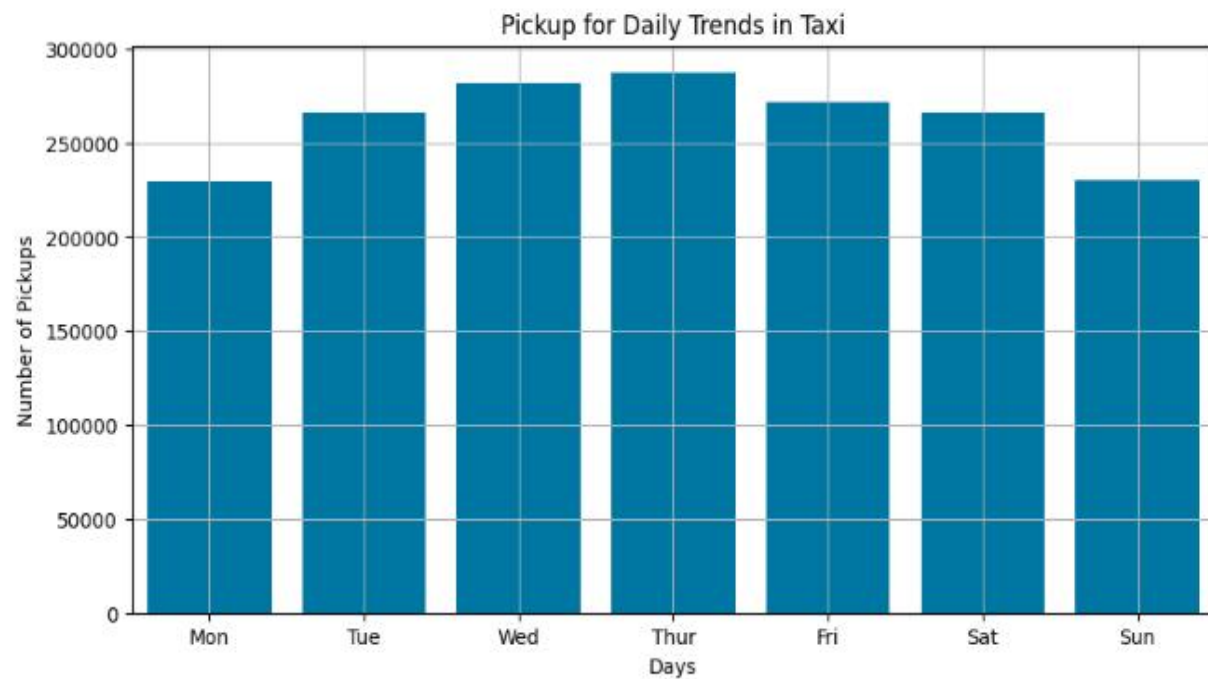
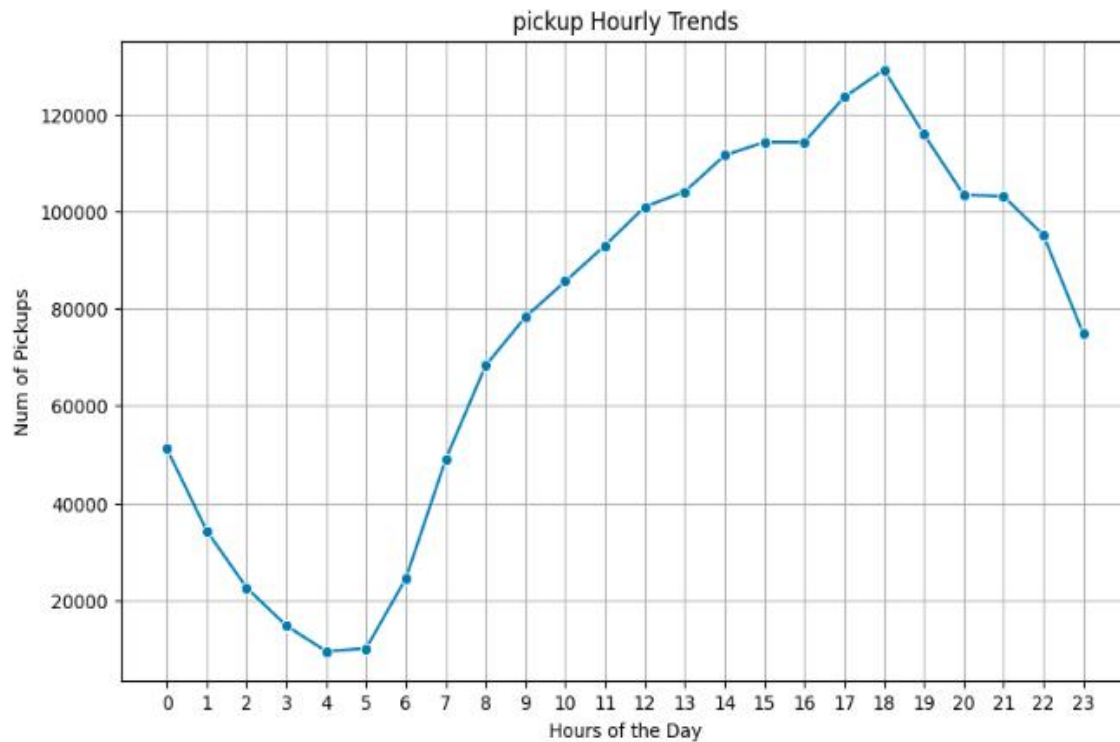
### 3.1.1 Classify variables into categorical and numerical

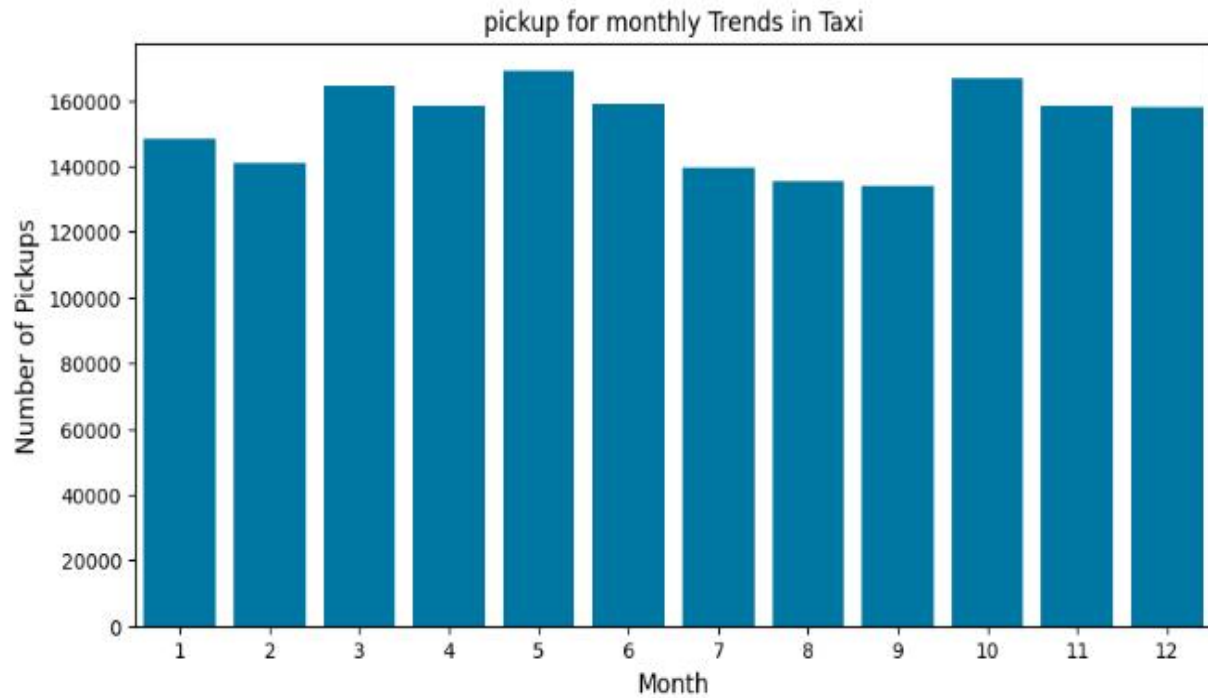
- VendorID:
- tpep\_pickup\_datetime:
- tpep\_dropoff\_datetime:
- passenger\_count:
- trip\_distance:
- RatecodeID:
- PULocationID:
- DOLocationID:
- payment\_type:
- pickup\_hour:
- trip\_duration:

The following monetary parameters belong in the same category, is it categorical or numerical?

- fare\_amount
- extra
- mta\_tax
- tip\_amount
- tolls\_amount
- improvement\_surcharge
- total\_amount
- congestion\_surcharge
- airport\_fee

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

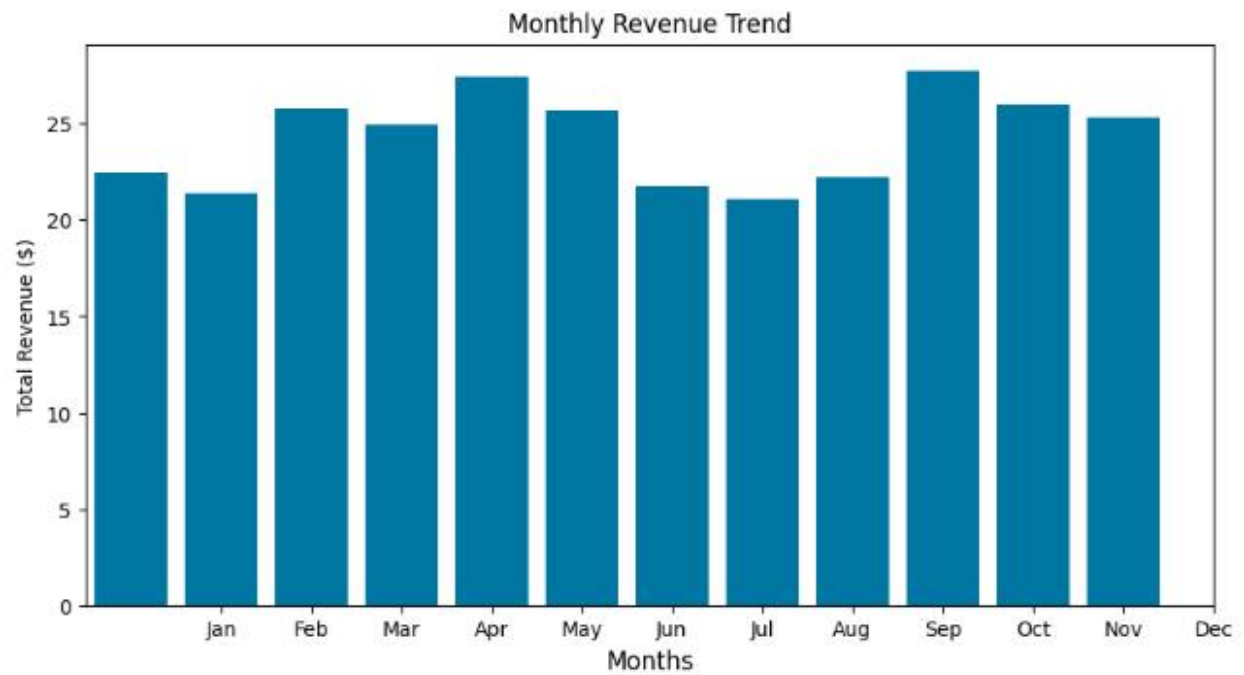




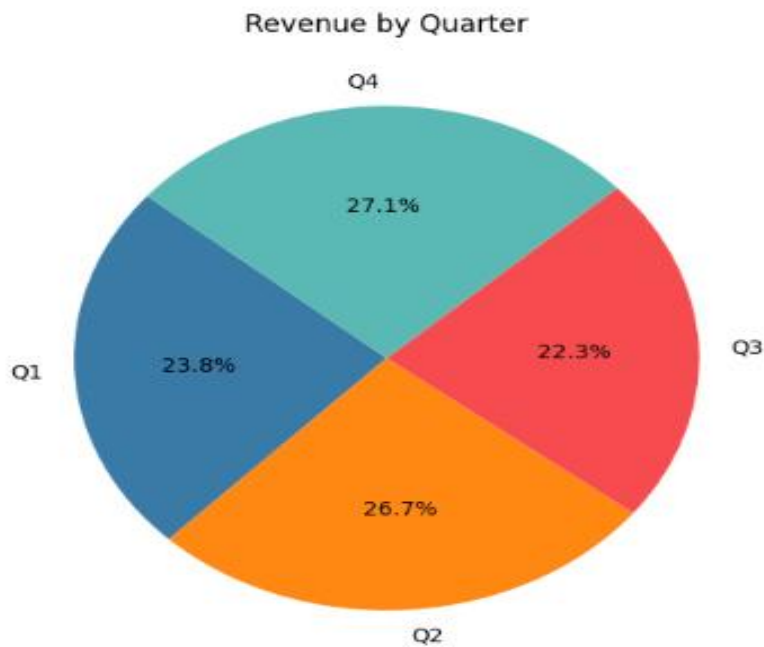
### 3.1.3 Filter out the zero/negative values in fares, distance and tips.

To keep the dataset clean and reliable, I removed records that didn't make sense. Trips with a **fare\_amount** or **total\_amount** of zero were dropped, as they likely represented canceled or incorrect entries. I also filtered out trips where the **trip\_distance** was zero but the pickup and dropoff locations were different, since that's not realistic. On the other hand, rows with **zero tip\_amount** were kept, because tipping is optional, and many valid trips didn't include a tip but still had proper fare and total values.

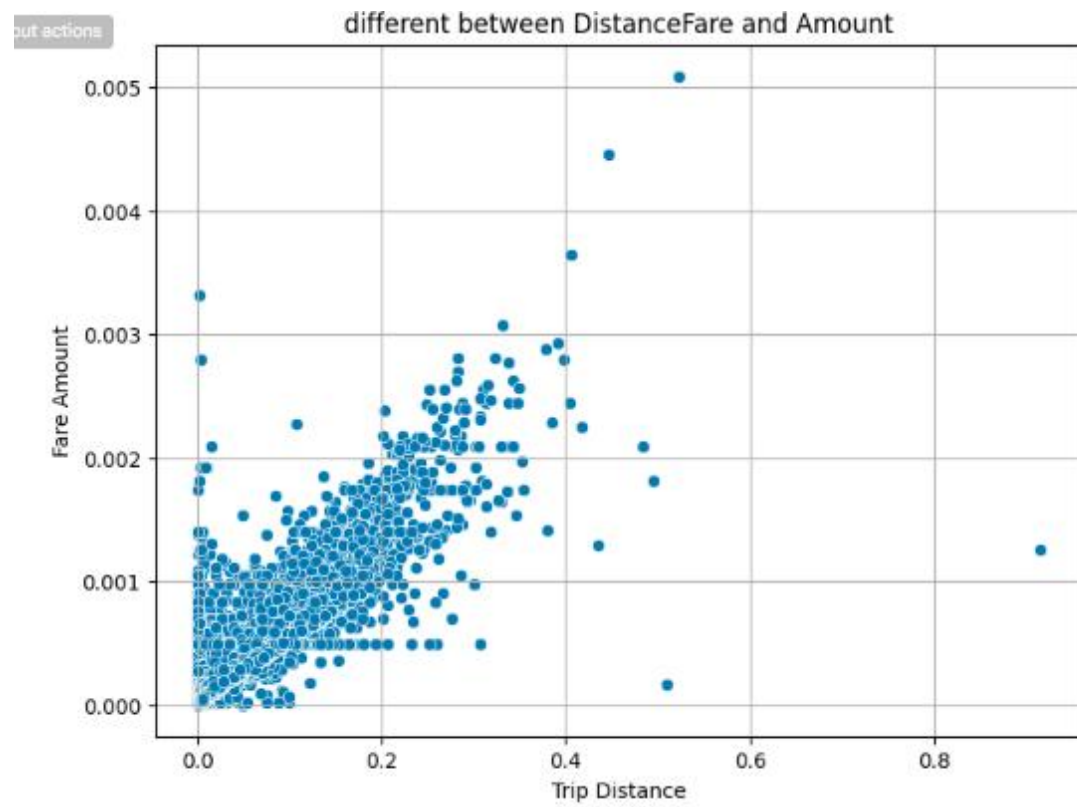
### 3.1.4 Analyse the monthly revenue trends



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

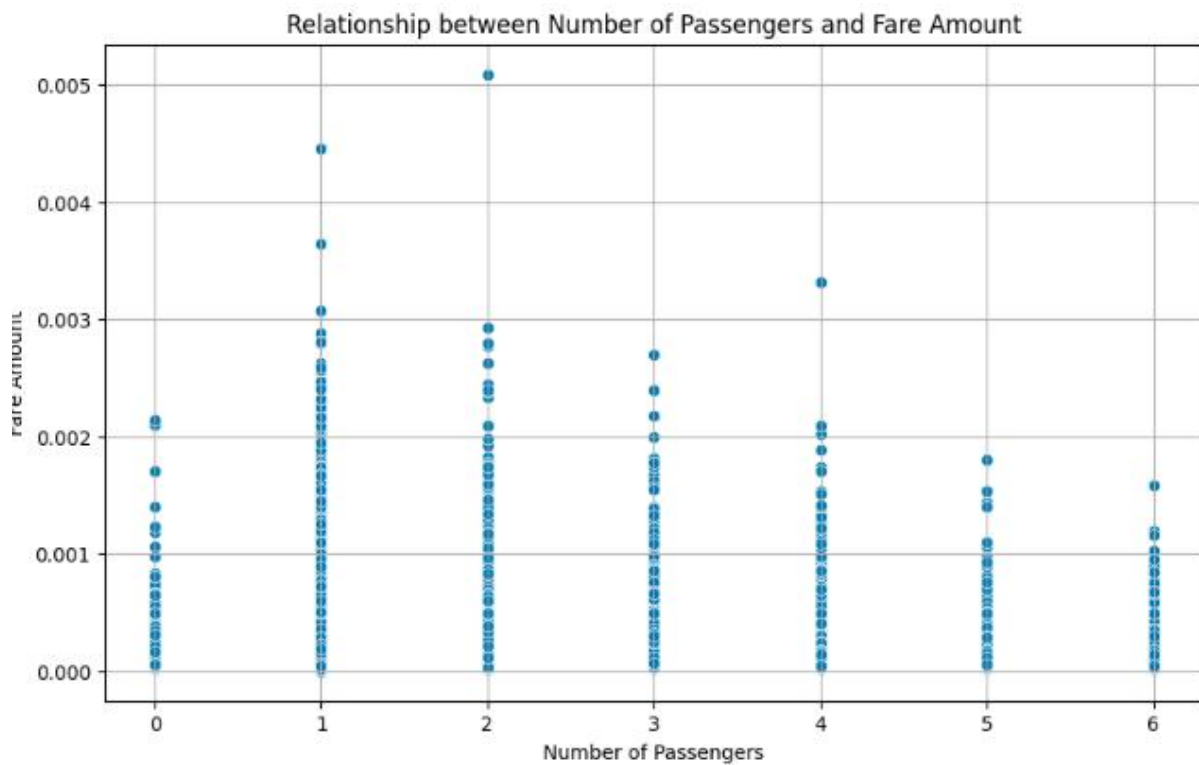
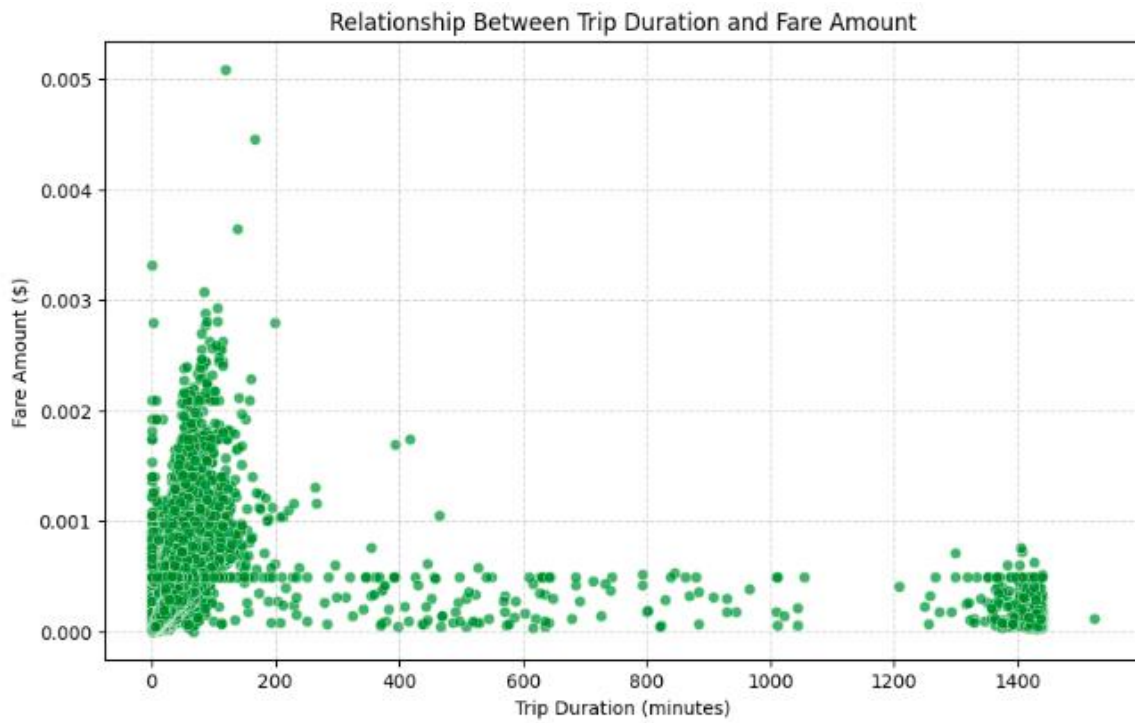


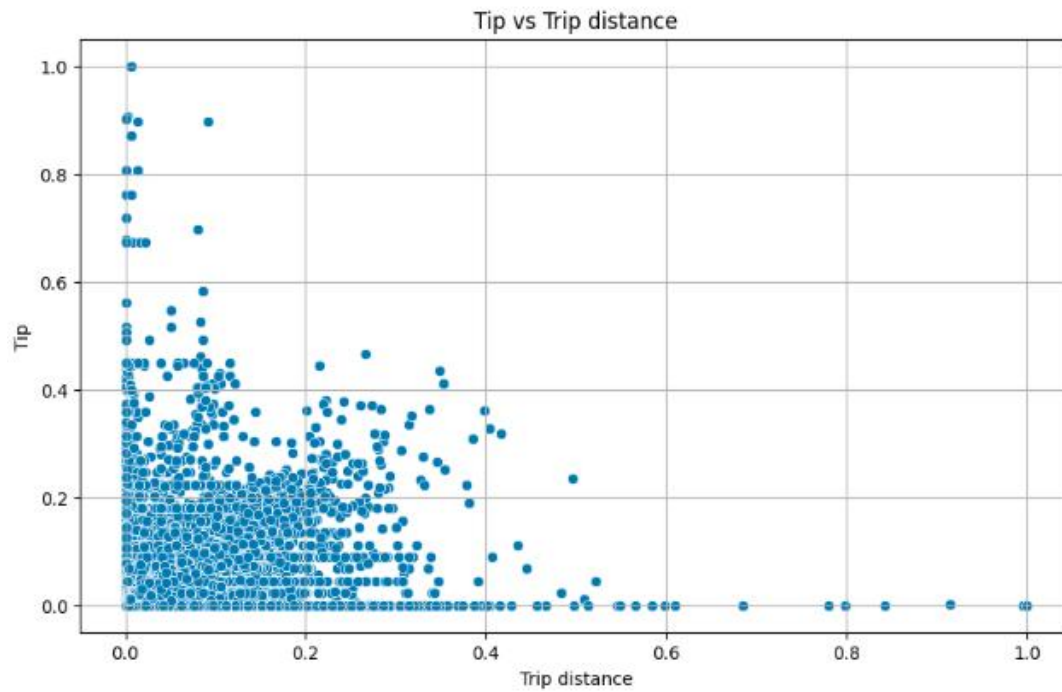
### 3.1.6 Analyse and visualise the relationship between distance and fare amount



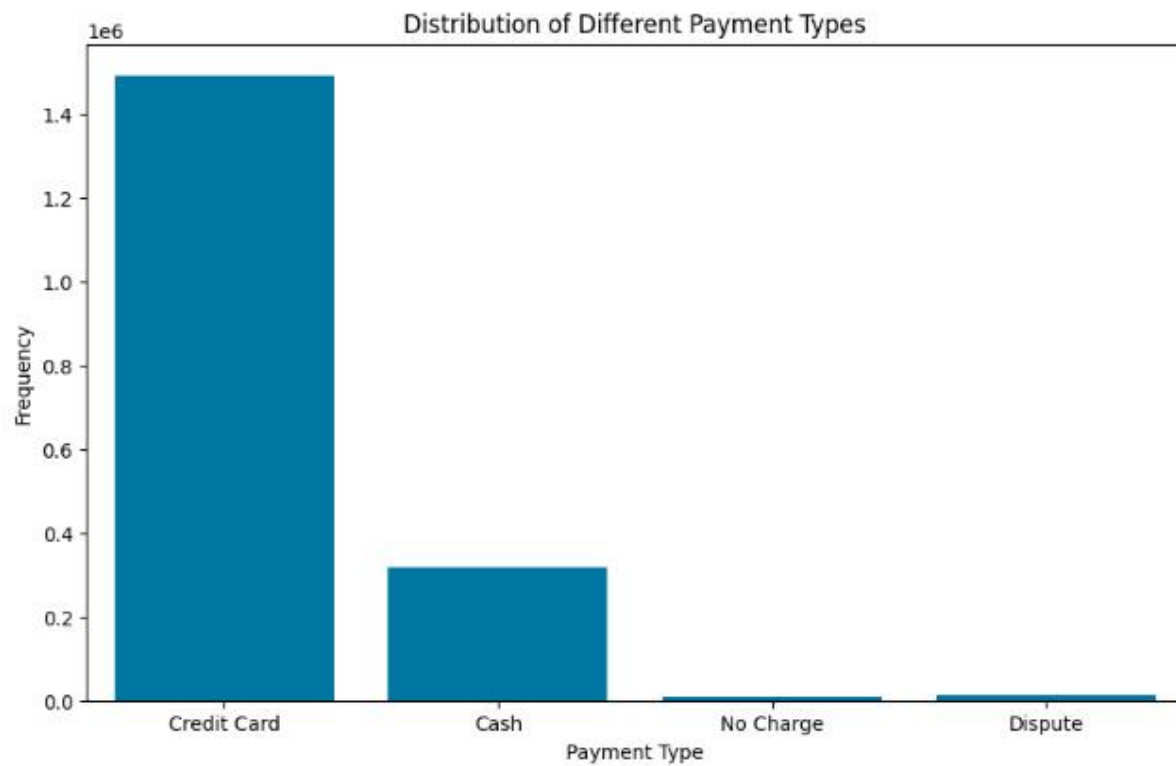


### 3.1.7 Analyse the relationship between fare/tips and trips/passengers





### 3.1.8 Analyse the distribution of different payment types

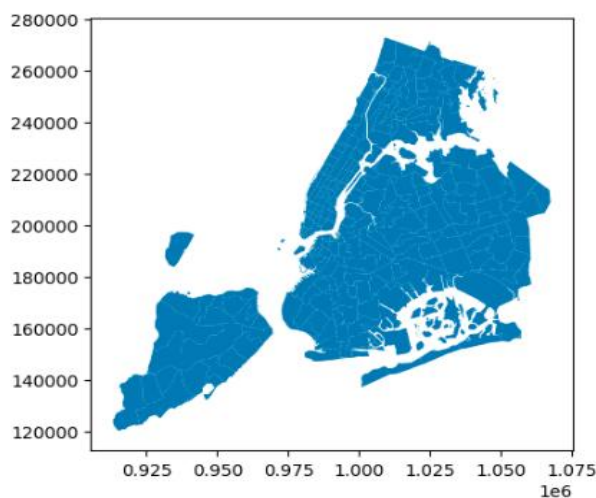


### 3.1.9 Load the taxi zones shapefile and display it

	OBJECTID	Shape_Leng	Shape_Area	zone	LocationID	borough	geometry
0	1	0.116357	0.000782	Newark Airport	1	EWB	POLYGON(((933100.918 192536.086, 933091.011 19...
1	2	0.433470	0.004866	Jamaica Bay	2	Queens	MULTIPOLYGON(((1033269.244 172126.008, 103343...
2	3	0.084341	0.000314	Allerton/Pelham Gardens	3	Bronx	POLYGON(((1026308.77 256767.698, 1026495.593 2...
3	4	0.043567	0.000112	Alphabet City	4	Manhattan	POLYGON(((992073.467 203714.076, 992068.667 20...
4	5	0.092146	0.000498	Arden Heights	5	Staten Island	POLYGON(((935843.31 144283.336, 936046.565 144...

Next steps: [Generate code with zones](#) [View recommended plots](#) [New interactive sheet](#)


```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 263 entries, 0 to 262
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   OBJECTID    263 non-null   int32
1   Shape_Leng  263 non-null   float64
2   Shape_Area  263 non-null   float64
3   zone        263 non-null   object
4   LocationID  263 non-null   int32
5   borough     263 non-null   object
6   geometry    263 non-null   geometry
dtypes: float64(2), geometry(1), int32(2), object(2)
memory usage: 12.5+ KB
None
```





### 3.1.10 Merge the zone data with trips data

The zones dataset was merged into the trip per loC using the locationID from the zones data and the PULocationID from the trip data as the key columns.

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



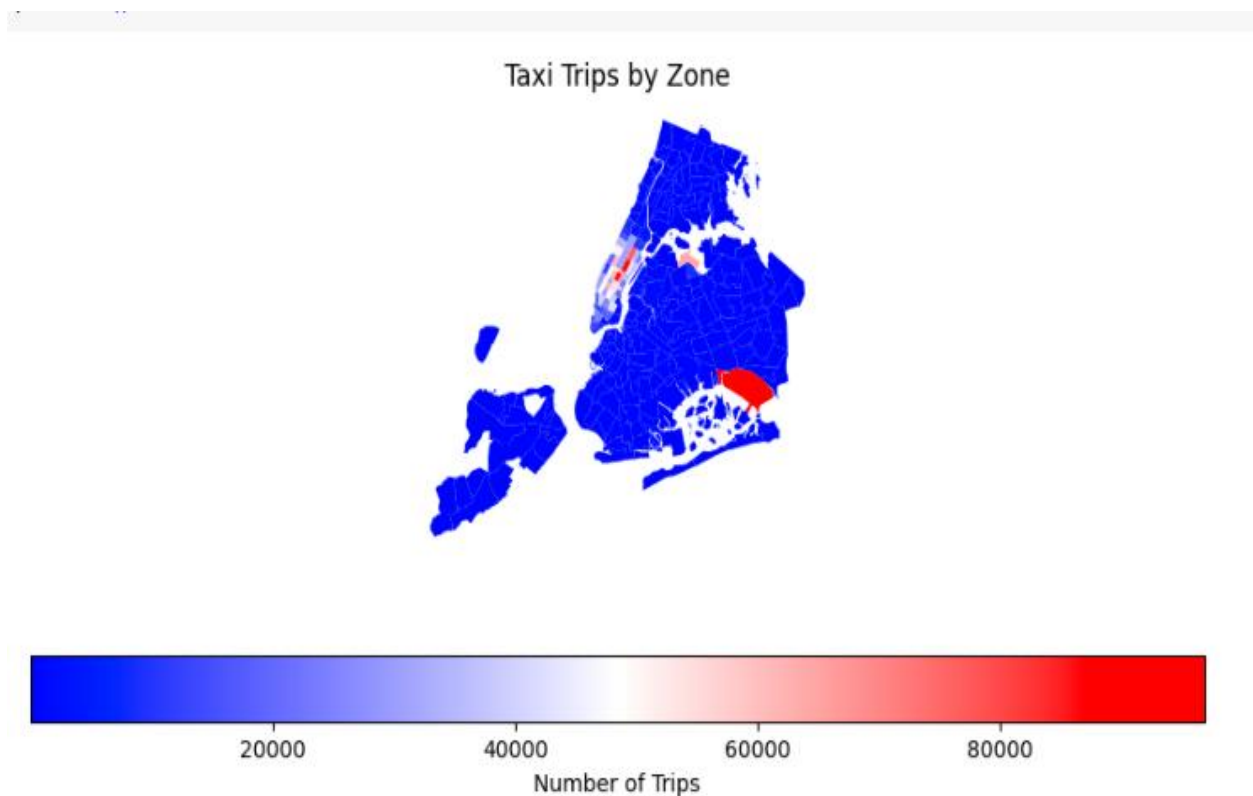
	PULocationID	numoftrips
0	1	214
1	2	2
2	3	40
3	4	1861
4	5	13



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

OBJECTID	Shape_Leng	Shape_Area	zone	LocationID	borough	geometry	PULocationID	numoftrips	
0	1	0.116357	0.000782	Newark Airport	1	EWB	POLYGON ((933100.918 192536.086, 933091.011 19...	1.0	214.0
1	2	0.433470	0.004866	Jamaica Bay	2	Queens	MULTIPOLYGON (((1033269.244 172126.008, 103343...	2.0	2.0
2	3	0.084341	0.000314	Allerton/Pelham Gardens	3	Bronx	POLYGON ((1026308.77 256767.698, 1026495.593 2...	3.0	40.0
3	4	0.043567	0.000112	Alphabet City	4	Manhattan	POLYGON ((992073.467 203714.076, 992068.667 20...	4.0	1861.0
4	5	0.092146	0.000498	Arden Heights	5	Staten Island	POLYGON ((935843.31 144283.336, 936046.565 144...	5.0	13.0

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



### 3.1.14 Conclude with results

- There is a clear link between **trip distance** and **fare amount**, showing that longer trips generally cost more.
- **Weekdays** saw peak activity during morning and evening rush hours, while **weekends** had more late-night trips.
- **Midtown Manhattan** and **airport areas** stood out as the busiest zones for both pickups and drop-offs.
- Most rides had **1 to 2 passengers**, and **credit cards** were the most frequently used payment method.
- A noticeable seasonal trend was found, with **July to September (Q3)** being the most active period for taxi trips.

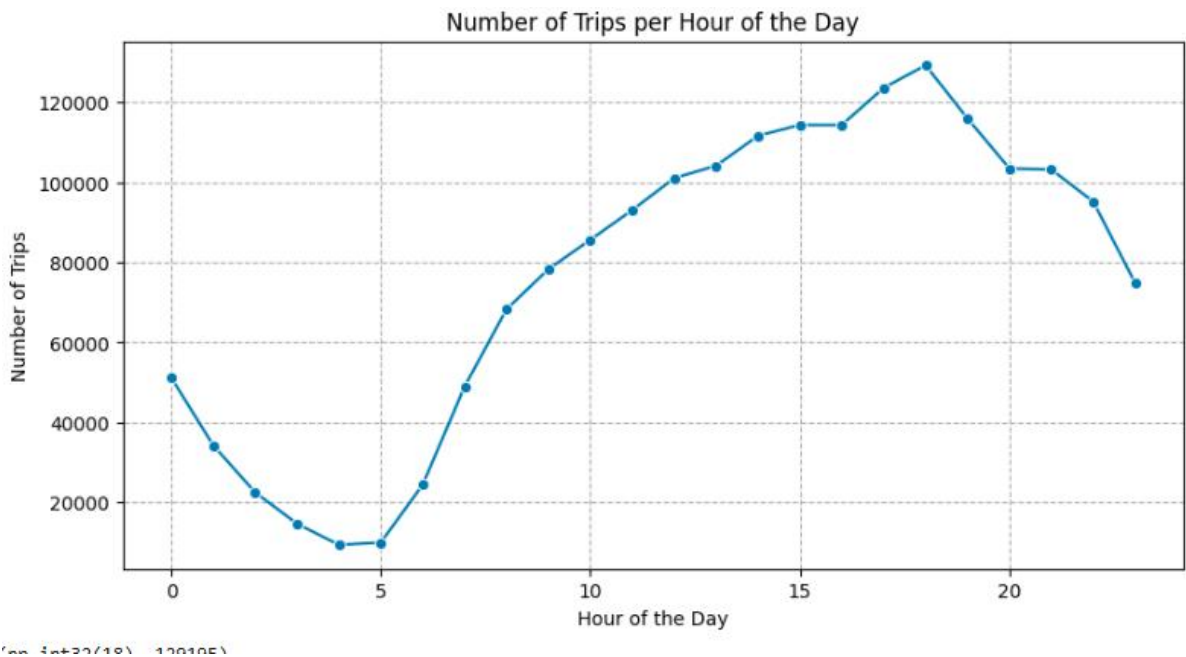
- Careful data cleaning helped remove errors and standardize key numeric columns, improving the overall quality of the analysis.

### 3.2 Detailed EDA: Insights and Strategies

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

	PULocationID	DOLocationID	pickup_hour	avg_duration	total_distance	avg_speed
102294	232	65	13	92.040556	0.002392	0.000026
114929	243	264	17	23.159167	0.000879	0.000038
61252	142	142	5	23.559167	0.002734	0.000116
120428	258	258	1	0.762500	0.000098	0.000128
33393	100	7	8	5.573889	0.001074	0.000193
6451	40	65	21	23.907222	0.005467	0.000229
39490	113	235	22	5.820556	0.001367	0.000235
89226	194	194	16	0.204444	0.000049	0.000239
95261	226	145	18	30.179352	0.007631	0.000253
9705	45	45	10	0.840556	0.000244	0.000290

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

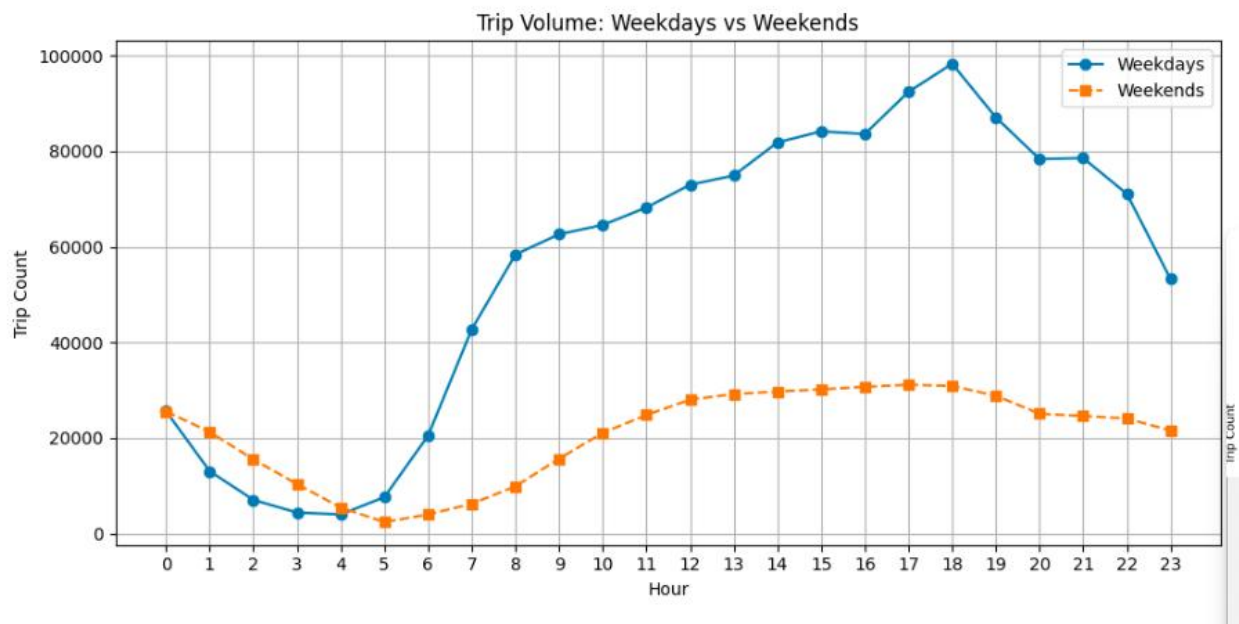




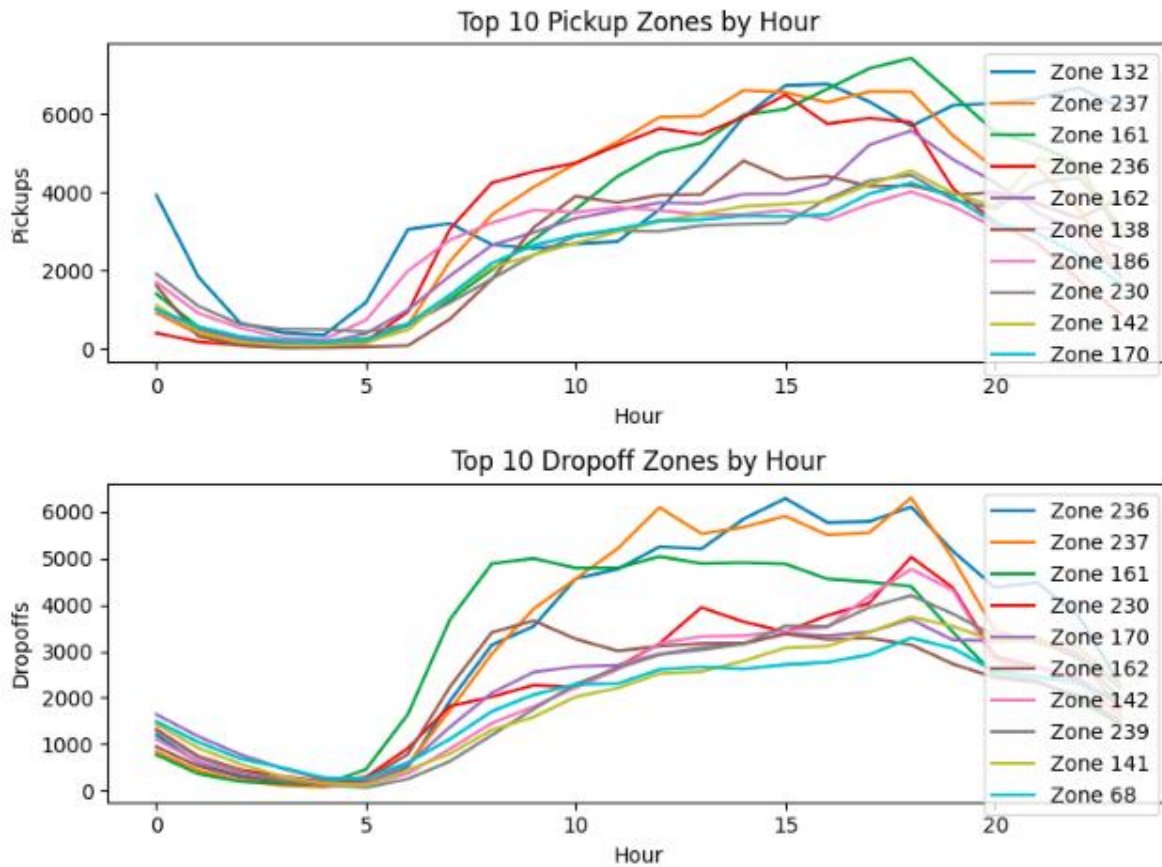
### 3.2.3 Scale up the number of trips from above to find the actual number of trips

pickup_hour	
18	129195
17	123569
19	115922
15	114311
16	114302

### 3.2.4 Compare hourly traffic on weekdays and weekends



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



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

Top 10 Zones with Highest Pickup/Dropoff Ratios:

	pickups	dropoffs	ratio
70	8354.0	1004.0	8.320717
132	96827.0	20970.0	4.617406
138	64177.0	22249.0	2.884489
199	2.0	1.0	2.000000
186	63471.0	40117.0	1.582147
43	30752.0	22369.0	1.374760
114	24113.0	17540.0	1.374743
249	40406.0	30468.0	1.326178
162	65634.0	52250.0	1.256153
161	85948.0	71648.0	1.199587

Bottom 10 Zones with Lowest Pickup/Dropoff Ratios

	pickups	dropoffs	ratio
30	0.0	18.0	0.000000
245	0.0	30.0	0.000000
176	0.0	12.0	0.000000
99	0.0	3.0	0.000000
27	1.0	39.0	0.025641
221	1.0	34.0	0.029412
257	29.0	758.0	0.038259
1	214.0	5319.0	0.040233
115	1.0	23.0	0.043478
198	52.0	990.0	0.052525



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

```
Top 10 Pickup Zones (11PM to 5AM):
pickup_zone
East Village          15339
JFK Airport           13399
West Village          12352
Clinton East          9797
Lower East Side       9535
Greenwich Village South 8720
Times Sq/Theatre District 7776
Penn Station/Madison Sq West 6233
Midtown South         5962
LaGuardia Airport     5947
Name: count, dtype: int64

Top 10 Dropoff Zones (11PM to 5AM):
dropoff_zone
East Village          8239
Clinton East          6641
Murray Hill           6085
Gramercy              5627
East Chelsea          5551
Lenox Hill West       5122
West Village          4896
Yorkville West        4878
Lower East Side       4321
Times Sq/Theatre District 4297
Name: count, dtype: int64
```

### 3.2.8 Find the revenue share for nighttime and daytime hours

```
Revenue Share Night time (11 PM - 5 AM): 12.06%
Revenue Share Daytime(6 AM - 10 PM): 87.94%
```

actions

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

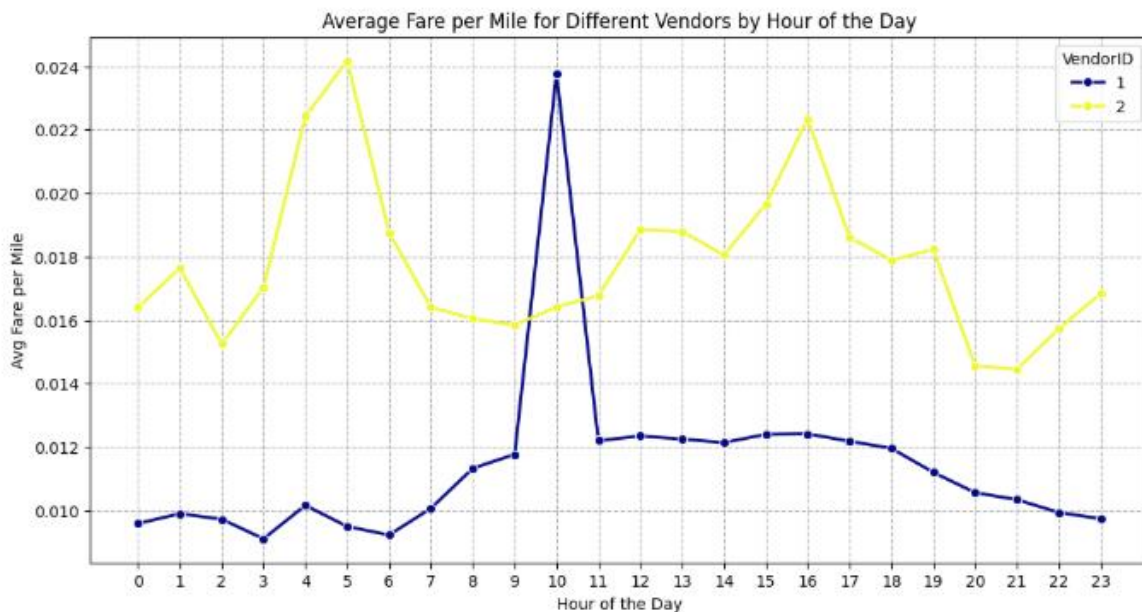
	passenger_count	fare_per_passenger_mile
0	1	0.024175
1	2	0.013309
2	3	0.008308
3	4	0.008498
4	5	0.003936
5	6	0.003173

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

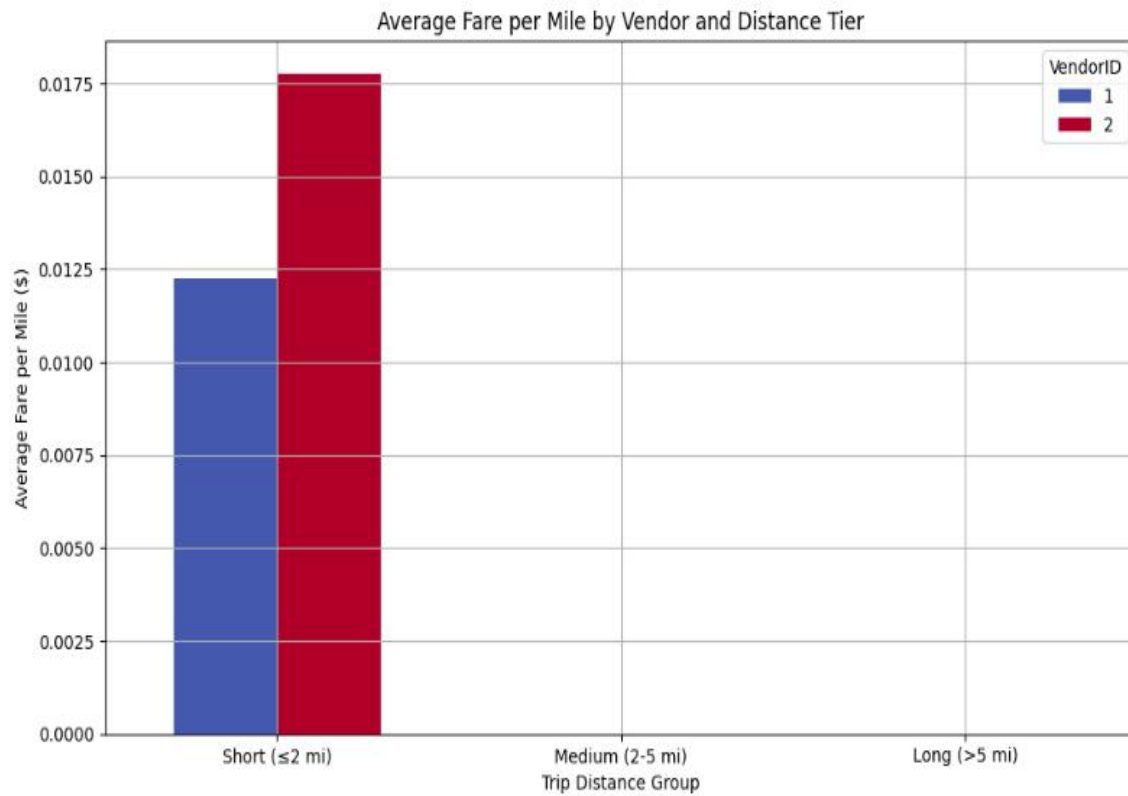
```
Average Fare per Mile by Day of Week:  
day_of_week  
Monday      0.02  
Tuesday      0.02  
Wednesday    0.02  
Thursday     0.02  
Friday       0.02  
Saturday     0.02  
Sunday       0.02  
Name: fare_per_mile, dtype: float64
```

```
Average Fare per Mile by Hour of Day:  
hour  
0      0.01  
1      0.02  
2      0.01  
3      0.02  
4      0.02  
5      0.02  
6      0.02  
7      0.01  
8      0.01  
9      0.01  
10     0.02  
11     0.02  
12     0.02  
13     0.02  
14     0.02  
15     0.02  
16     0.02  
17     0.02  
18     0.02  
19     0.02  
20     0.01  
21     0.01  
22     0.01  
23     0.02  
Name: fare_per_mile, dtype: float64
```

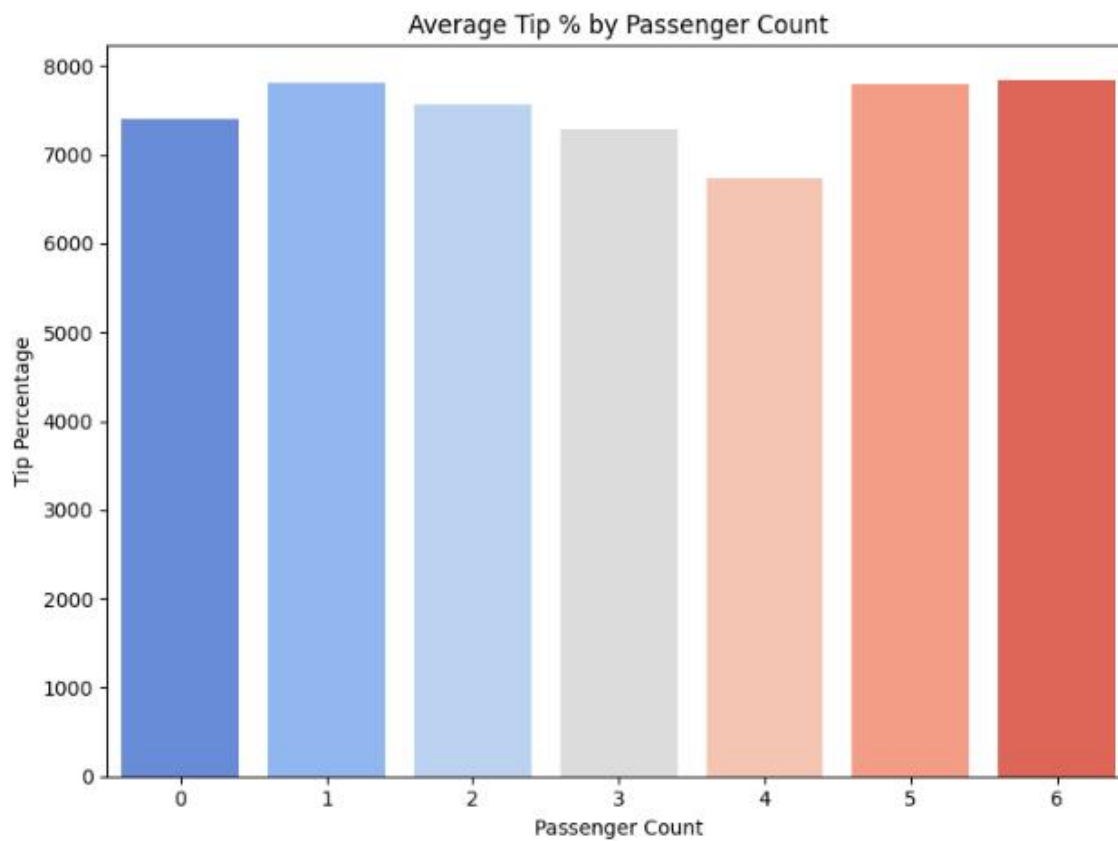
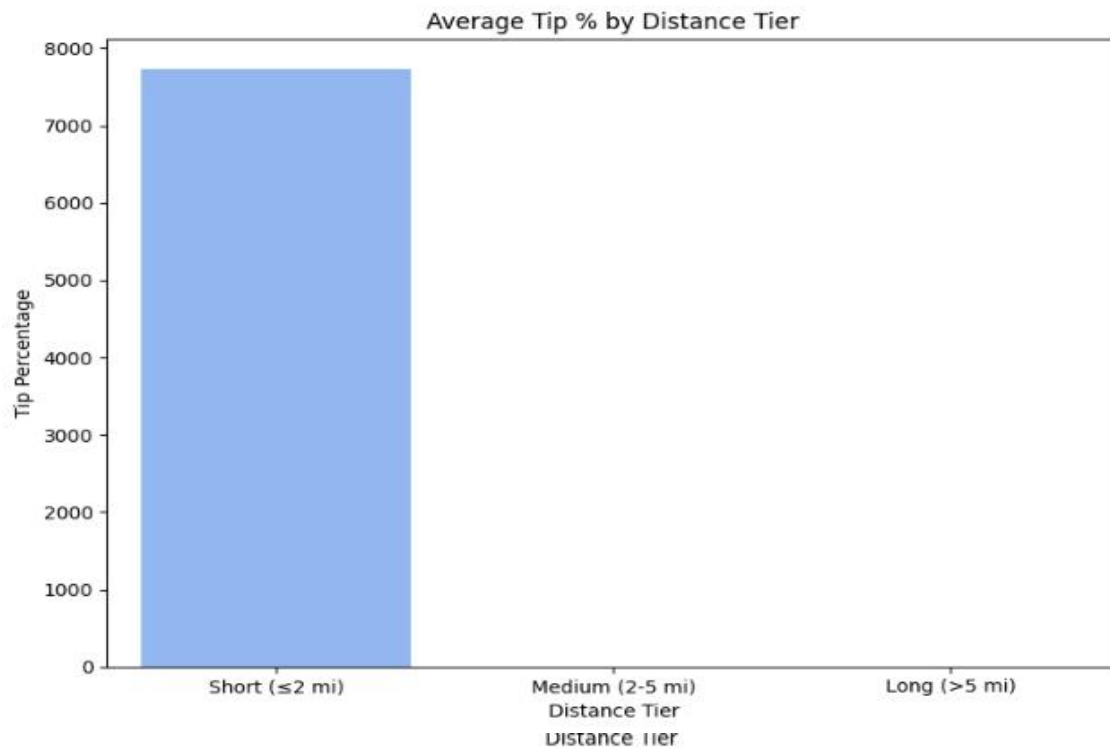
### 3.2.11 Analyse the average fare per mile for the different vendors

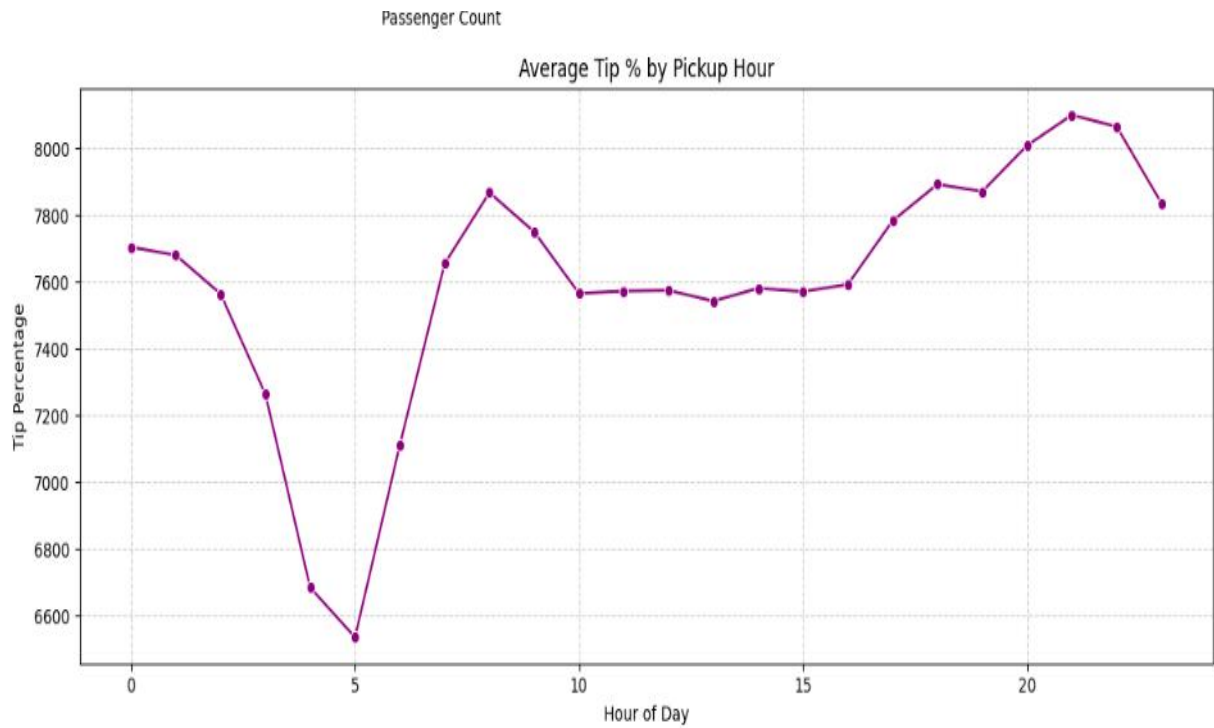


### 3.2.12 Compare the fare rates of different vendors in a distance-tiered fashion

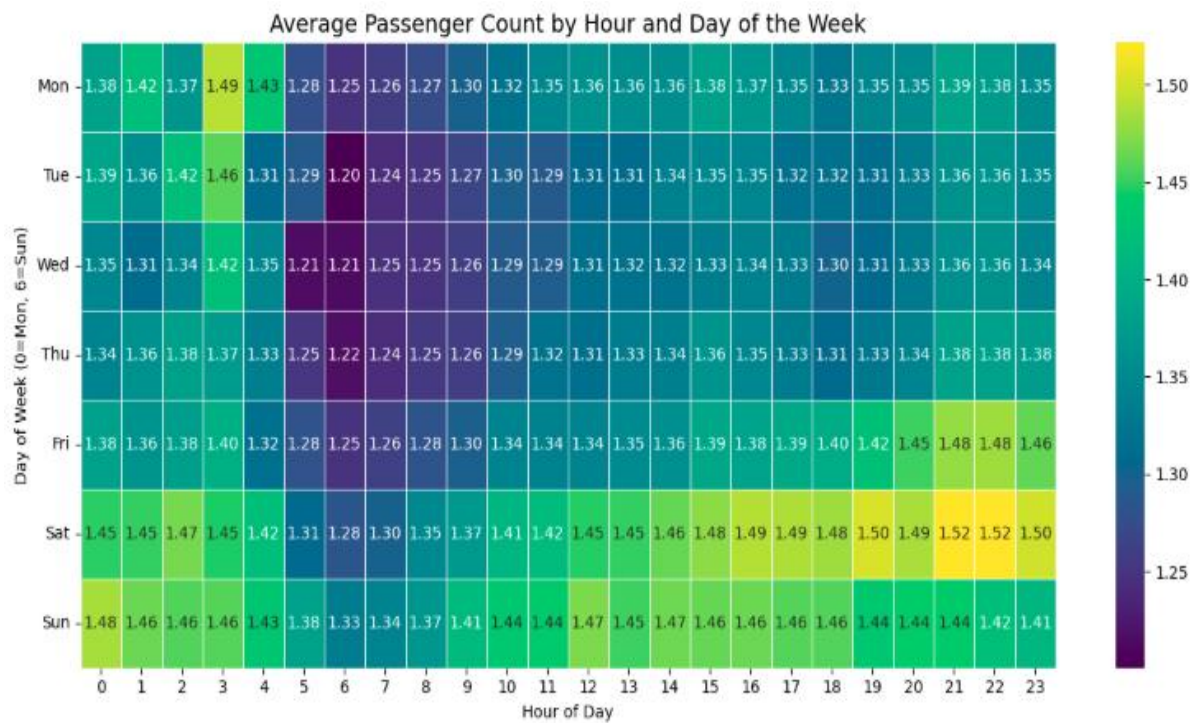


### 3.2.13 Analyse the tip percentages

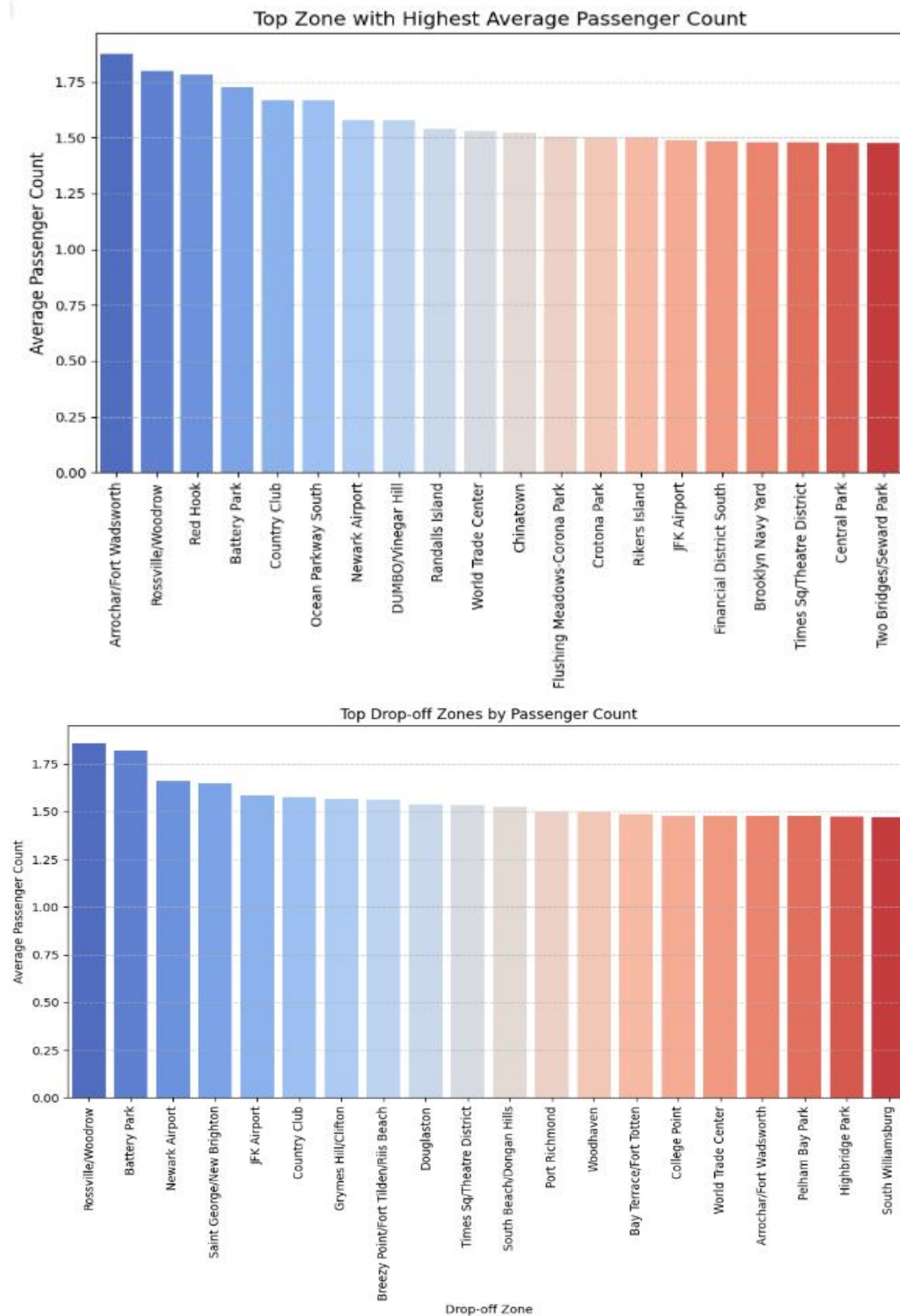




3.2.14 Analyse the trends in passenger count.

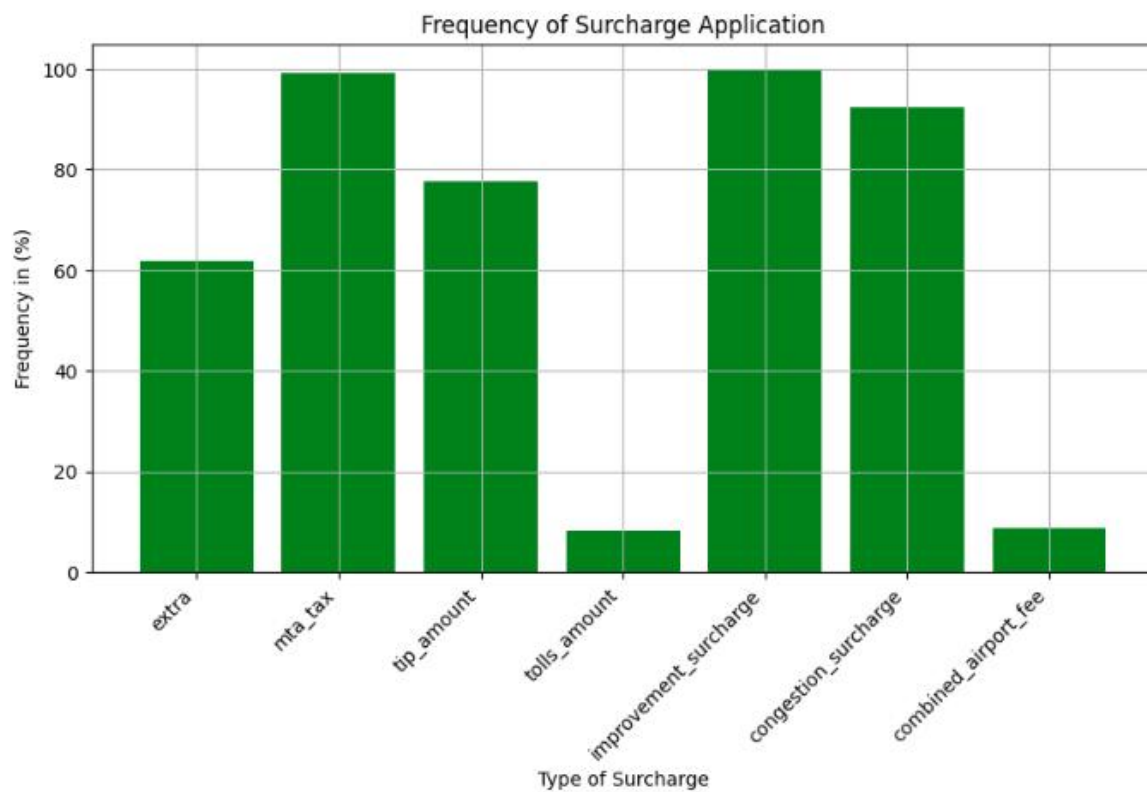


### 3.2.15 Analyse the variation of passenger counts across zones.



3.2.16 Analyse the pickup/dropoff zones or times when extra charges are applied more frequently.

```
Frequency of Surcharge Application (%):  
extra          61.942644  
mta_tax        99.066884  
tip_amount     77.599868  
tolls_amount   8.116399  
improvement_surcharge 99.959757  
congestion_surcharge 92.309561  
combined_airport_fee 8.787583  
dtype: float64
```



## 4. Conclusions

### 4.1 Final Insights and Recommendations

#### 4.1.1 Recommendations to optimize routing and dispatching based on demand patterns and operational inefficiencies.

**Targeted Nighttime Coverage:** Increase cab presence in nightlife-dense areas such as East Village, JFK Airport, and West Village between 11:00 PM and 2:00 AM to meet elevated late-night demand. Simultaneously, scale down deployments in consistently low-activity zones like the Bronx and Staten Island during these hours to improve fleet utilization.

**Strategic Airport Operations:** Airports such as JFK and LaGuardia generate sustained traffic and incur additional fare surcharges, making them ideal for continuous short-haul shuttle loops. Ensure dedicated cab availability throughout both day and night shifts to support airport access without service gaps.

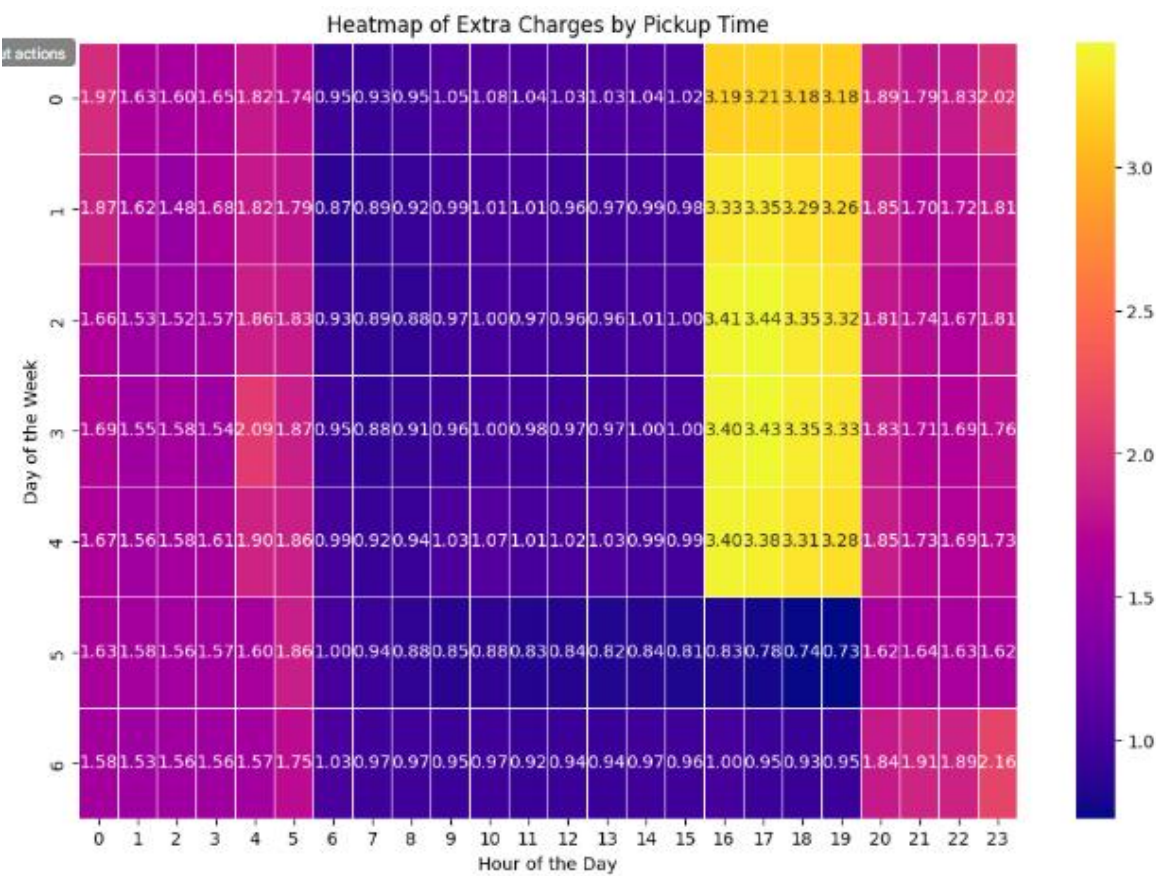
**Efficiency-Driven Zoning:** Minimize dispatch frequency to areas characterized by a high volume of drop-offs but consistently low pickup activity—especially purely residential neighborhoods—unless data forecasts a strong likelihood of return-trip demand.

**Real-Time Resource Reallocation:** Integrate live traffic and passenger flow analytics to proactively shift idle vehicles from oversupplied locations to emerging high-demand areas. Hourly trend tracking should drive these redistributions to maintain service balance.

**Smart Trip Pairing:** In regions where single-passenger rides are common but request frequency remains high, implement app-based features that support ride-pooling or sequential trip dispatching. This approach maximizes vehicle efficiency and reduces passenger wait times.



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



1. Time-Aware Deployment:

Adjust fleet availability to mirror temporal demand cycles—boost coverage during peak commute hours (morning and evening), increase presence in nightlife zones during late hours, and scale back during midday lulls. Incorporate seasonal shifts by analyzing monthly trends to proactively manage surges or slowdowns.

2. Day-Specific Focus:

Concentrate services around commercial and financial districts on weekdays to cater to office-goers. On weekends, redirect resources toward residential neighborhoods, parks, and entertainment venues. Stay responsive to demand spikes during public holidays, festivals, and major city events.

### 3. Zone-Level Prioritization:

Enhance coverage in high-frequency pickup zones by analyzing historical demand data. Use drop-off trends to detect and correct geographic imbalances. Expand visibility in areas that maintain strong late-night activity to meet under-addressed demand.

### 4. Data-Driven Deployment:

Leverage live traffic feeds and real-time booking activity to inform tactical vehicle placements. Augment this with predictive models to forecast high-demand windows, enabling preemptive positioning. Integrate with ride-hailing platforms to align cab availability with customer needs dynamically.

### 5. Coordinated Operations:

Foster ongoing communication between dispatch centers and drivers to provide live updates on demand hotspots and suggested routes. Collaborate with city transport departments to address urban mobility challenges and align with broader transportation goals.

### 6. Technology Integration:

Utilize GPS tracking, demand heatmaps, and interactive analytics dashboards to gain visibility into rider behavior and cluster activity. These tools enable smarter resource allocation and agile operational adjustments.

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

### 1. Tiered Distance-Based Fare Structuring

Implement graduated pricing models that offer modest discounts for medium-to-long-distance trips (e.g., those exceeding 5 miles). This encourages extended rides, which tend to have a lower per-mile cost, thereby increasing overall utilization and ride frequency.

### 2. Smart Surge Pricing Mechanism

Leverage real-time and historical data on hourly and location-specific demand to enable adaptive pricing. Introduce surge multipliers in high-traffic zones such as Times Square, Midtown Manhattan, and major airports (LGA/JFK) during peak hours, weekends, and event-driven surges.

### 3. Incentivized Ride-Sharing Models

Promote cab-pooling by offering lower fare-per-mile rates for trips with multiple passengers. This not only enhances vehicle occupancy rates but also increases revenue per trip while mitigating the impact of low individual passenger counts.

### 4. Transparent Airport Fare Bundles

Offer fixed-price airport packages that bundle base fare, tolls, and mandated surcharges for JFK and LGA trips. Flat-fee options improve cost predictability and can enhance rider trust—especially for first-time or occasional travelers.

#### 5. Competitive Vendor Pricing Insights

Maintain transparency in pricing differentials between vendors. If Vendor 2 continues to charge higher rates for short-distance trips, highlight the comparative cost advantage of Vendor 1 within the app or interface to appeal to price-sensitive riders.