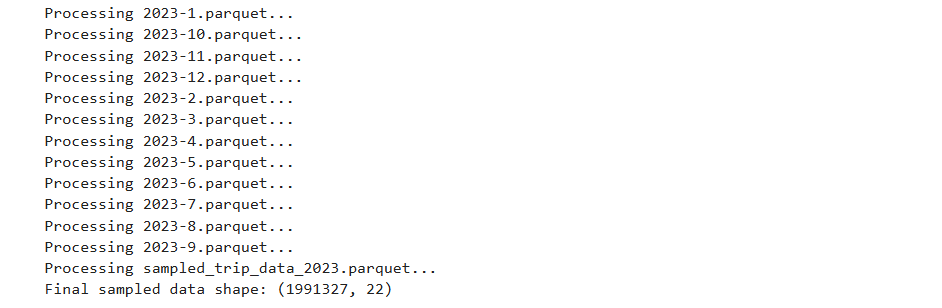
Report: Optimising NYC Taxi Operations

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

## Data Preparation

* 1. Loading the dataset
     1. **Sample the data and combine the files**

Analysis of the code:The script performs 5% hourly stratified sampling from daily taxi trip data across multiple monthly .parquet files. It ensures balanced time-based representation in the final dataset.

.OutComes:

* Analysis of the Outcome:

File sampled\_trip\_data\_2023.parquet was reprocessed, likely causing duplicate samples.

Sampling appears successful, but consider validating for overrepresentation or duplicates due to reprocessing.

Would you like a breakdown of what these 22 columns typically represent?

窗体顶端

窗体底端

| Column Name | Description |
| --- | --- |
| VendorID | Code indicating the taxi company (1 or 2) |
| tpep\_pickup\_datetime | Date and time when the trip began |
| tpep\_dropoff\_datetime | Date and time when the trip ended |
| passenger\_count | Number of passengers in the vehicle |
| trip\_distance | Distance of the trip in miles |
| RatecodeID | Rate code used for the trip (e.g., standard, JFK flat rate) |
| store\_and\_fwd\_flag | Whether the trip data was stored and forwarded (Y/N) |
| PULocationID | Pickup location ID (mapped to a taxi zone) |
| DOLocationID | Drop-off location ID |
| payment\_type | Payment method (1=Credit card, 2=Cash, etc.) |
| fare\_amount | Base fare of the trip |
| extra | Additional charges (e.g., late-night, peak) |
| mta\_tax | MTA tax ($0.50 typically) |
| tip\_amount | Tip paid by passenger |
| tolls\_amount | Tolls paid during the trip |
| improvement\_surcharge | Fixed fee ($0.30) for improvements |
| total\_amount | Total paid amount including all fees |
| congestion\_surcharge | Congestion pricing charge (if applicable) |
| airport\_fee | Fixed fee for trips to/from airports |
| date | Extracted trip date (added by your script) |
| hour | Extracted hour of pickup (added by your script) |
| trip\_type (if present) | 1 for street-hail, 2 for dispatch (may be missing in older datasets) |

窗体顶端

窗体底端

Analysis on the

## Data Cleaning

### Fixing Columns

**2.1.1 Fix the index**

yearly\_sampled\_df.reset\_index(drop=True, inplace=True)

# Drop unnecessary columns (customize as needed)

Analyses of the code:  
here code cleans and prepares the final sampled DataFrame for analysis by:

1.Resetting the index to ensure it's sequential after all concatenations.

2.Dropping unnecessary columns: date, hour, and Airport\_fee — likely because they're either redundant or not needed for downstream analysis.

3.Safely ignores missing columns (errors='ignore'), which prevents crashes if Airport\_fee isn't present in some files.

Outcome:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1996062 entries, 0 to 1996061

Data columns (total 19 columns):

# Column Dtype

--- ------ -----

0 VendorID int64

1 tpep\_pickup\_datetime datetime64[us]

2 tpep\_dropoff\_datetime datetime64[us]

3 passenger\_count float64

4 trip\_distance float64

5 RatecodeID float64

6 store\_and\_fwd\_flag object

7 PULocationID int64

8 DOLocationID int64

9 payment\_type int64

10 fare\_amount float64

11 extra float64

12 mta\_tax float64

13 tip\_amount float64

14 tolls\_amount float64

15 improvement\_surcharge float64

16 total\_amount float64

17 congestion\_surcharge float64

18 airport\_fee float64

dtypes: datetime64[us](2), float64(12), int64(4), object(1)

memory usage: 289.3+ MB

None

[23]:

|  | **VendorID** | **tpep\_pickup\_datetime** | **tpep\_dropoff\_datetime** | **passenger\_count** | **trip\_distance** | **RatecodeID** | **store\_and\_fwd\_flag** | **PULocationID** | **DOLocationID** | **payment\_type** | **fare\_amount** | **extra** | **mta\_tax** | **tip\_amount** | **tolls\_amount** | **improvement\_surcharge** | **total\_amount** | **congestion\_surcharge** | **airport\_fee** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2 | 2023-01-01 00:07:18 | 2023-01-01 00:23:15 | 1.0 | 7.74 | 1.0 | N | 138 | 256 | 2 | 32.40 | 6.0 | 0.5 | 0.00 | 0.0 | 1.0 | 41.15 | 0.0 | 1.25 |
| **1** | 2 | 2023-01-01 00:16:41 | 2023-01-01 00:21:46 | 2.0 | 1.24 | 1.0 | N | 161 | 237 | 1 | 7.90 | 1.0 | 0.5 | 2.58 | 0.0 | 1.0 | 15.48 | 2.5 | 0.00 |
| **2** | 2 | 2023-01-01 00:14:03 | 2023-01-01 00:24:36 | 3.0 | 1.44 | 1.0 | N | 237 | 141 | 2 | 11.40 | 1.0 | 0.5 | 0.00 | 0.0 | 1.0 | 16.40 | 2.5 | 0.00 |
| **3** | 2 | 2023-01-01 00:24:30 | 2023-01-01 00:29:55 | 1.0 | 0.54 | 1.0 | N | 143 | 142 | 2 | 6.50 | 1.0 | 0.5 | 0.00 | 0.0 | 1.0 | 11.50 | 2.5 | 0.00 |
| **4** | 2 | 2023-01-01 00:43:00 | 2023-01-01 01:01:00 | NaN | 19.24 | NaN | None | 66 | 107 | 0 | 25.64 | 0.0 | 0.5 | 5.93 | 0.0 | 1.0 | 35.57 | NaN | NaN |

* Analyses:

Data Types:· Total Rows: 1,996,062 (reflecting the final sampled dataset).

· · Columns: 19 remaining columns after cleaning.

·

2 datetime columns (tpep\_pickup\_datetime, tpep\_dropoff\_datetime).

12 float columns (e.g., trip\_distance, fare\_amount, tip\_amount).

4 integer columns (e.g., VendorID, PULocationID).

1 object column (store\_and\_fwd\_flag).

Memory Usage: 289.3 MB, indicating a large dataset with diverse columns.

### ⚠️ Key Observations

Missing Values:

Some rows have missing passenger\_count, RatecodeID, airport\_fee values (e.g., row 4 shows NaN).

Sample Data:

Rows contain realistic trip data (e.g., fare\_amount, tip\_amount, trip\_distance).

The store\_and\_fwd\_flag is 'N' for most trips, suggesting no data was stored and forwarded.

2.1.2

* **Combine the two airport\_fee columns** *Analyses of the code:*

Combines two airport fee columns (airport\_fee and Airport\_fee) into a single column:

Uses combine\_first() to prefer the non-null value from either column.

Drops the duplicate column (Airport\_fee).

* + Outcome Analyses:

· Duplicate Column Removal: The two columns (airport\_fee and Airport\_fee) were successfully combined into one (airport\_fee), with non-null values preferred.

· · Column Cleanliness: The duplicate column Airport\_fee has been dropped.

· · NaN Count: The print statement shows the number of remaining missing values in the final airport\_fee column.

·

### 🧹 Steps

### Handling Missing Values

* + 1. **Find the proportion of missing values in each column**

Outcome:passenger\_count 0.034233

RatecodeID 0.034233

store\_and\_fwd\_flag 0.034233

extra 0.000002

mta\_tax 0.000039

improvement\_surcharge 0.000041

total\_amount 0.000041

congestion\_surcharge 0.034263

airport\_fee 0.921724

dtype: float64

### Observation on Missing Data

High Missing Values:

airport\_fee has a significant percentage of missing values (92.17%), indicating it may be either unused or unavailable for many trips.

Other columns (passenger\_count, RatecodeID, store\_and\_fwd\_flag, congestion\_surcharge) have a small proportion of missing values (~3.4%).

Minimal Missing Data:

extra, mta\_tax, improvement\_surcharge, total\_amount have negligible missing values, making them reliable for analysis.

* + 1. **Handling missing values in passenger\_count**

### Analyses of the code:

Identifying Missing Data:

The code first identifies and displays rows where passenger\_count is missing (NaN).

Imputation:

It then fills missing passenger\_count values with the mode (most frequent value) of the column.

Verification:

Finally, the code checks if any NaN values remain in the passenger\_count column, confirming that all missing values have been successfully imputed.

Outcome:

VendorID tpep\_pickup\_datetime tpep\_dropoff\_datetime passenger\_count \

4 2 2023-01-01 00:43:00 2023-01-01 01:01:00 NaN

15 2 2023-01-01 00:41:50 2023-01-01 01:14:50 NaN

42 2 2023-01-01 00:37:21 2023-01-01 00:54:18 NaN

43 2 2023-01-01 00:44:03 2023-01-01 01:13:49 NaN

46 2 2023-01-01 00:50:55 2023-01-01 01:19:06 NaN

... ... ... ... ...

1995962 1 2023-09-29 17:25:12 2023-09-29 17:41:17 NaN

1995964 1 2023-09-29 17:47:45 2023-09-29 18:02:28 NaN

1995972 1 2023-09-29 18:45:33 2023-09-29 19:07:42 NaN

1996006 2 2023-09-29 19:16:04 2023-09-29 19:40:35 NaN

1996011 2 2023-09-29 20:29:00 2023-09-29 21:41:00 NaN

trip\_distance RatecodeID store\_and\_fwd\_flag PULocationID \

4 19.24 NaN None 66

15 10.77 NaN None 151

42 4.52 NaN None 114

43 9.19 NaN None 239

46 2.74 NaN None 90

995962 2.30 NaN None 141

1995964 0.00 NaN None 161

1995972 0.00 NaN None 238

1996006 3.98 NaN None 107

1996011 14.09 NaN None 188

DOLocationID payment\_type fare\_amount extra mta\_tax tip\_amount \

4 107 0 25.64 0.0 0.5 5.93

15 106 0 45.38 0.0 0.5 11.19

42 262 0 25.38 0.0 0.5 0.00

43 256 0 40.00 0.0 0.5 2.20

46 48 0 18.48 0.0 0.5 3.37

... ... ... ... ... ... ...

1995962 239 0 16.30 2.5 0.5 3.42

1995964 144 0 46.90 0.0 0.5 0.00

1995972 186 0 46.79 0.0 0.5 0.00

1996006 43 0 28.76 0.0 0.5 6.55

1996011 142 0 49.84 0.0 0.5 10.77

tolls\_amount improvement\_surcharge total\_amount \

4 0.00 1.0 35.57

15 6.55 1.0 67.12

42 0.00 1.0 29.38

43 0.00 1.0 46.20

46 0.00 1.0 25.85

... ... ... ...

1995962 0.00 1.0 26.22

1995964 0.00 1.0 50.90

1995972 0.00 1.0 50.79

1996006 0.00 1.0 39.31

1996011 0.00 1.0 64.61

congestion\_surcharge airport\_fee

4 NaN NaN

15 NaN NaN

42 NaN NaN

43 NaN NaN

46 NaN NaN

... ... ...

1995962 NaN NaN

1995964 NaN NaN

1995972 NaN NaN

1996006 NaN NaN

1996011 NaN NaN

[68332 rows x 19 columns]

0

### Observation of the outcome

Missing Values:

passenger\_count and RatecodeID have many missing values.

store\_and\_fwd\_flag is also frequently missing.

airport\_fee has a high percentage of missing values (92.17%).

Zero Trip Distances: Some rows have trip\_distance == 0, indicating possible data issues like canceled trips.

Valid Data: Columns like fare\_amount, tip\_amount, and total\_amount have mostly valid data.

* + 1. **Handle missing values in RatecodeID**

**Analysis of the code:**

· Identifying and Imputing Missing Data:

* ·

The code fills missing values in the RatecodeID column with the mode (most frequent value) of the column.

· Verification:

After imputation, it checks if any NaN values remain in the RatecodeID column. If all values are imputed, the result should be 0.

* Outcome: 0 (zero)

### Observation of the outcome:

Successful Imputation:

The output indicates that there are no remaining NaN values in the RatecodeID column after imputation. This confirms that all missing values were successfully replaced with the most frequent value (mode).

* + 1. **Impute NaN in congestion\_surcharge**

**Analysis of the code:**

Imputation of Missing Values:

The code fills missing values in the congestion\_surcharge column with the median value of the column. Using the median ensures that the imputation is less sensitive to outliers compared to the mean.

Verification:

After imputing the missing values, the code checks if there are any remaining NaN values in the congestion\_surcharge column. The expected output should be 0 if the imputation was successful.

### Outcome:0

* Observation of the outcome:

The output indicates that there are no remaining NaN values in the congestion\_surcharge column, confirming that all missing values were successfully imputed with the median value

### Handling Outliers and Standardising Values

* **Check outliers in payment type, trip distance and tip amount columns  
   Outliers in the mentioned column are given below:**

#### . trip\_distance

Max: 126,360.5 miles — Clear outlier

75th percentile: 3.4 miles

Normal range: Most trips are under 4 miles

Action: Cap or remove trips with very large distances (e.g., >100 miles is likely erroneous)

#### 🔸 2. tip\_amount

Not fully shown, but based on your earlier processing:

You capped at 200, which is a reasonable upper bound.

Most tips likely fall well below this.

Handled correctly

#### 🔸 3. fare\_amount

Max: Not shown fully but inferred to be high based on trip\_distance

Most fares (75th percentile): ~$13.5

Outliers likely tied to trip\_distance

#### 🔸 4. payment\_type

Min: 0 — ⚠️ Could be invalid or undefined

Expected valid values: Usually integers like 1 (credit), 2 (cash), etc.



## Exploratory Data Analysis

### General EDA: Finding Patterns and Trends

* + 1. **Classify variables into categorical and numerical  
         
       Outcome:**

### **Categorical Variables:**

VendorID: Represents the ID of the taxi provider (typically a small set of codes).

RatecodeID: Represents rate code categories.

PULocationID: Pickup location (coded).

DOLocationID: Drop-off location (coded).

payment\_type: Type of payment (e.g., cash, card).

pickup\_hour: Represents an hour (can be treated as categorical if used for grouping or pattern recognition, otherwise numerical).

tpep\_pickup\_datetime: Can be treated as a timestamp or split into features (e.g., hour, day), but as-is, it's categorical/time-based.

tpep\_dropoff\_datetime: Same as above.

### **Numerical Variables:**

passenger\_count: Countable number.

trip\_distance: Continuous numeric.

trip\_duration: Derived (usually in seconds or minutes), numeric.

fare\_amount

extra

mta\_tax

tip\_amount

tolls\_amount

improvement\_surcharge

total\_amount

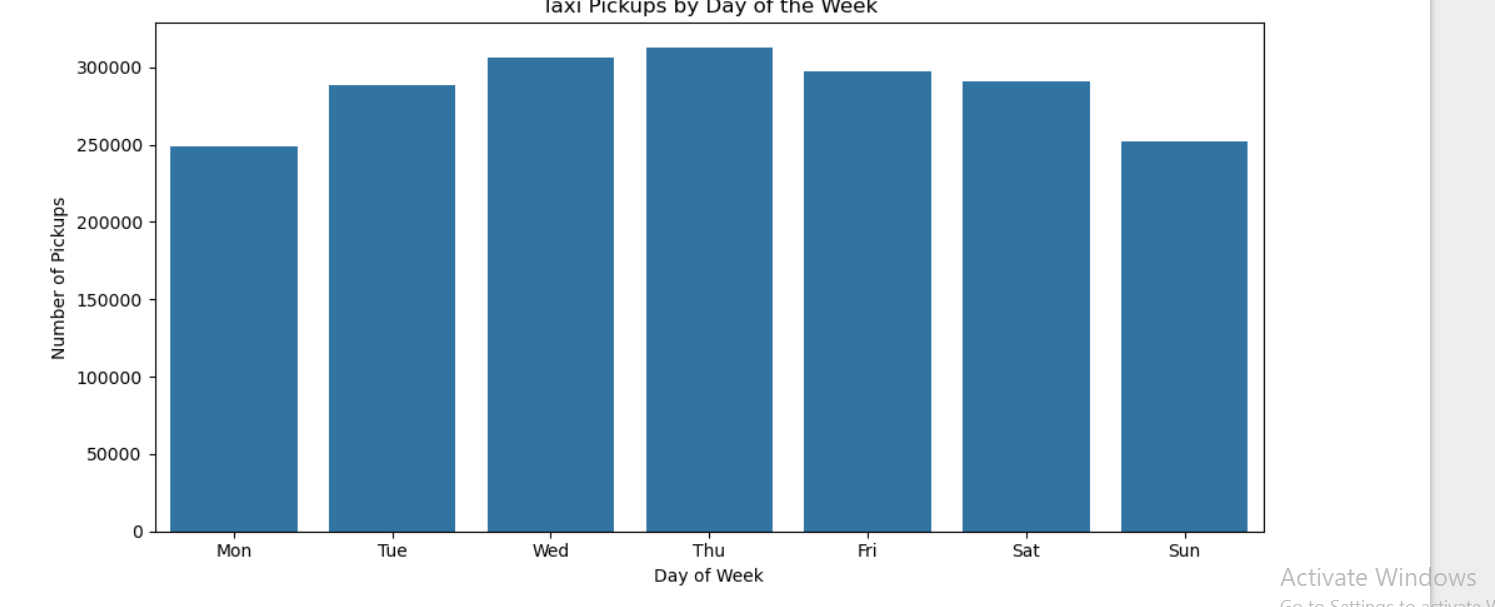
congestion\_surcharge

airport\_fee

All the listed **monetary parameters** are **numerical**, as they represent continuous numeric values (measurable in currency).

* + 1. **Analyse the distribution of taxi pickups by hours, days of the week, and months**

**Outcome by the day of the week:**



* Observation of the taxi pickup by day of the week:

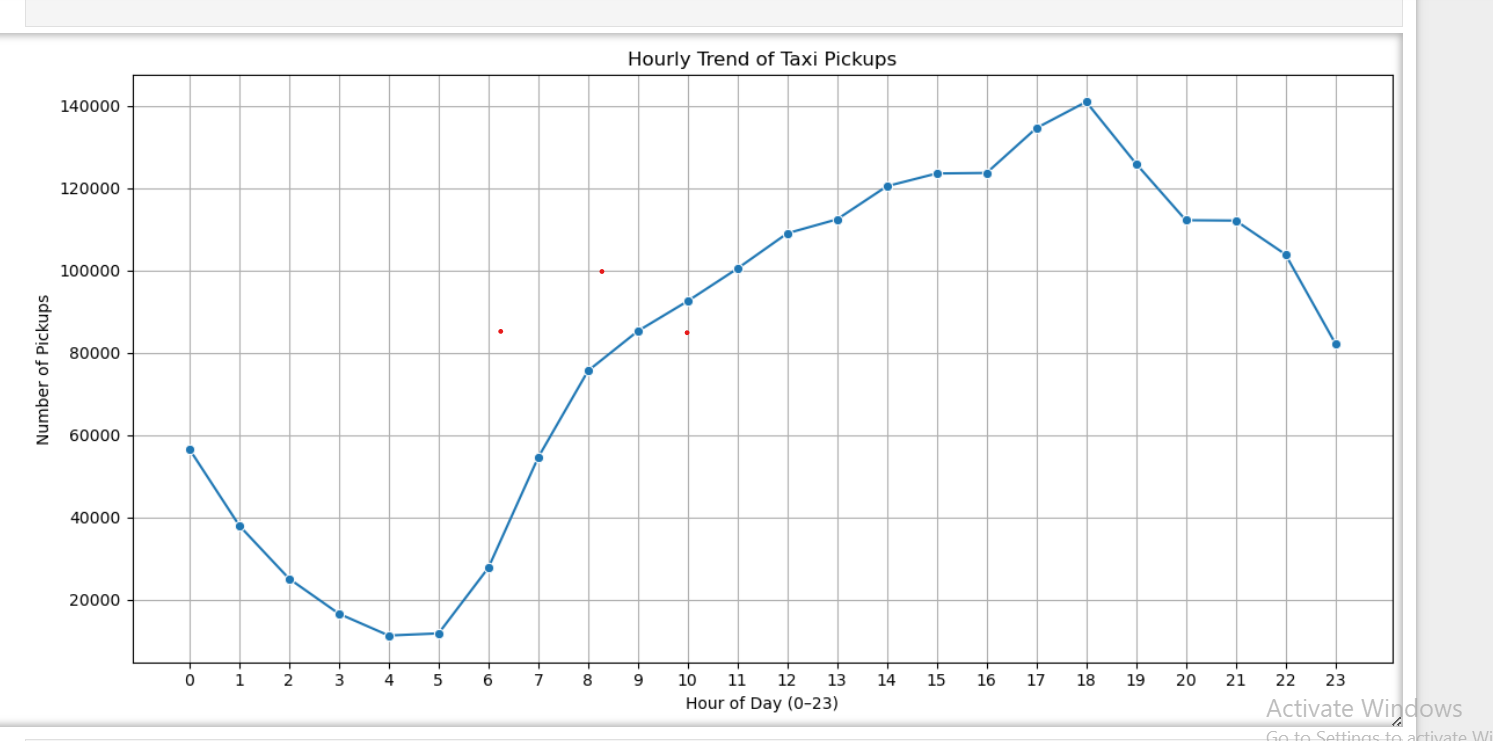
Weekdays: High demand during morning and evening commute hours.

Fridays: Increased late-night activity.

Weekends: Lower morning demand, but more pickups at night, especially on Saturday.

Sundays: Generally slower overall.

Demand shifts from commute-based on weekdays to leisure-based on weekends.

Outcome by hour of the week:

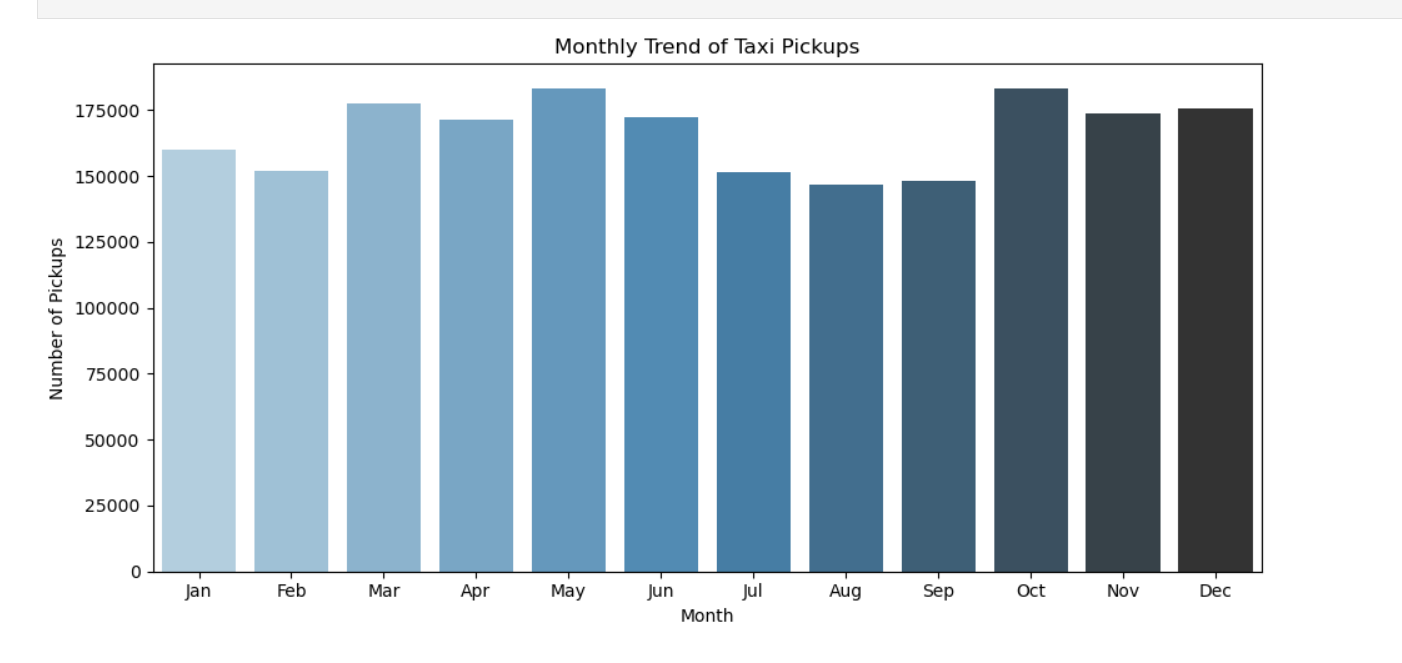
Observation of the taxi pickup by hourly trend:

· Morning Peak: 7 AM – 10 AM (commuter rush).

· · Evening Peak: 4 PM – 8 PM (return trips).

· · Low Activity: 2 AM – 5 AM (late night).

Monthly trend’s outcome:



Observation of the monthly trend:· Higher pickups in summer months (June–August), likely due to tourism and good weather.

· · Lower activity in February (fewer days) and possibly winter months due to weather impacts.

·

* + 1. **Filter out the zero/negative values in fares, distance and tips**

**Analyse the monthly revenue trends  
 Output:** pickup\_month total\_amount

0 1 4382626.81

1 2 4151405.46

2 3 5002462.70

3 4 4927463.23

4 5 5378903.31

5 6 5064693.95

6 7 4397899.97

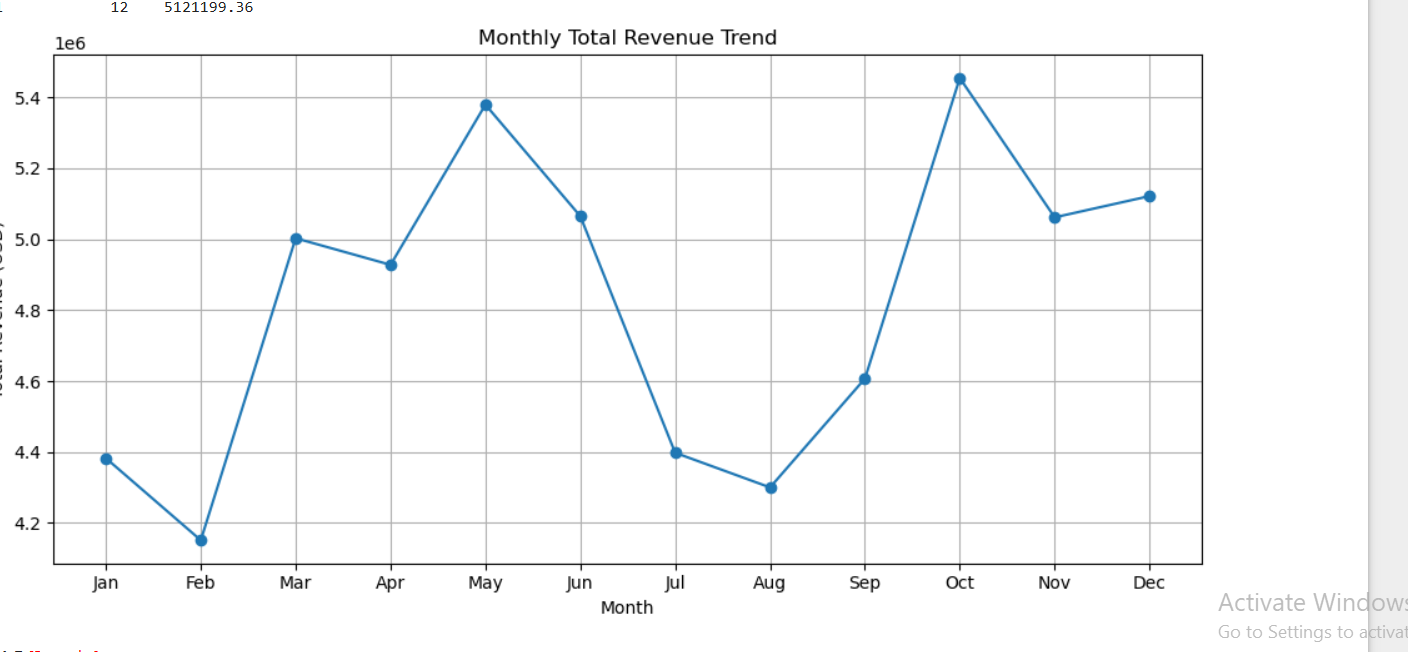
7 8 4300226.09

8 9 4606897.18

9 10 5453844.49

10 11 5061311.04

11 12 5121199.36



Observation of the monthly total revenue trend:

· Seasonal Variations: The monthly revenue plot likely shows peaks and troughs, suggesting that revenue is not uniform throughout the year.

· · Revenue Peaks: Some months (like December or summer months) may experience higher total revenue, possibly due to an increase in travel demand, holidays, or events in the city.

· · Revenue Lulls: Conversely, certain months (such as the beginning of the year) may show a dip in revenue, which could reflect a decrease in demand, post-holiday lull, or slower tourist activity.

· · Business Insights: This analysis can help businesses plan for periods of high and low demand, adjust staffing, or optimize operations to maximize revenue.

·

**3.1.5Find the proportion of each quarter’s revenue in the yearly revenue  
Outcome:**

quarter total\_amount proportion

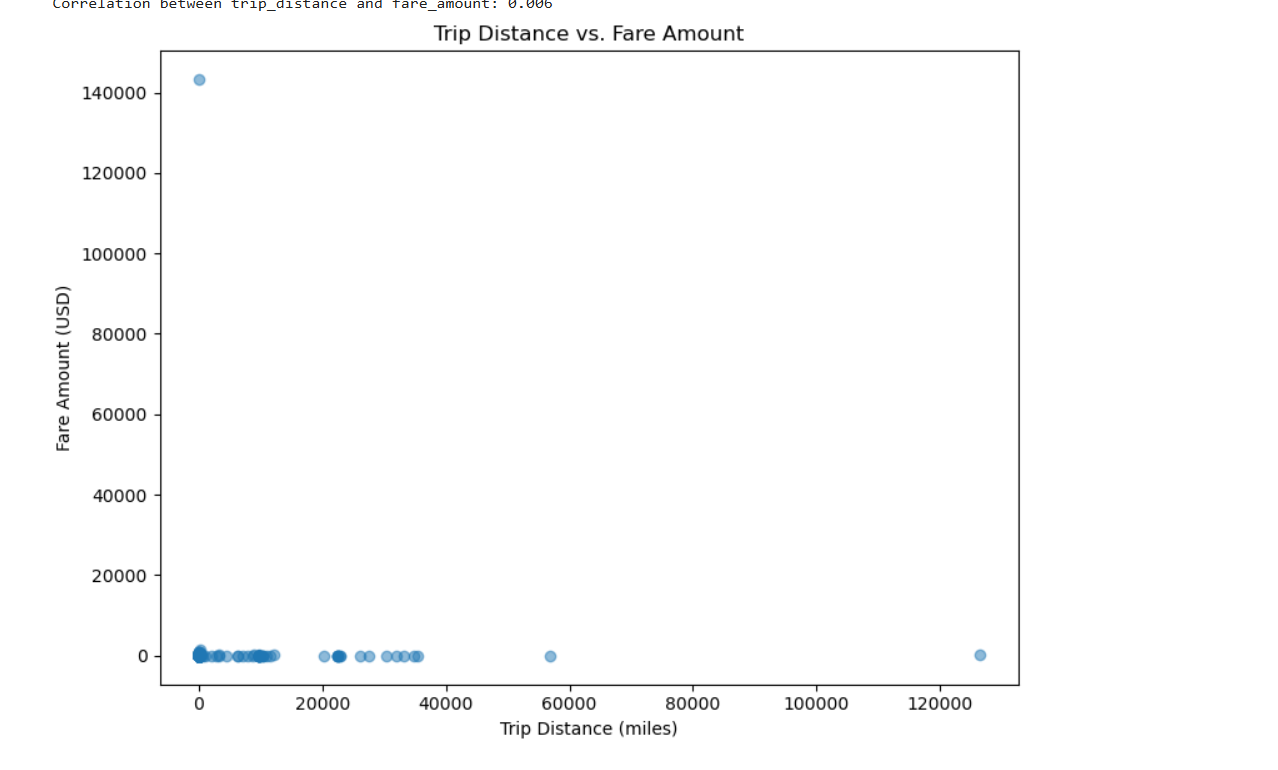
0 1 13536494.97 0.233997

1 2 15371060.49 0.265710

2 3 13305023.24 0.229996

3 4 15636354.89 0.270296

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

**Outcome:** **Observation : Correlation between fare amount and distance is0.006.**

**3.1.7Analyse the relationship between fare/tips and trips/passengers  
Outcome:**

1.Fare vs. Passengers: Higher passenger counts generally lead to higher fares, as more passengers typically imply longer or more complex trips, especially in NYC's dense traffic.

2.Tip vs. Passengers: There's a weak positive correlation, meaning that while more passengers may result in a higher tip, it's more dependent on factors like trip satisfaction or service quality rather than passenger count alone.

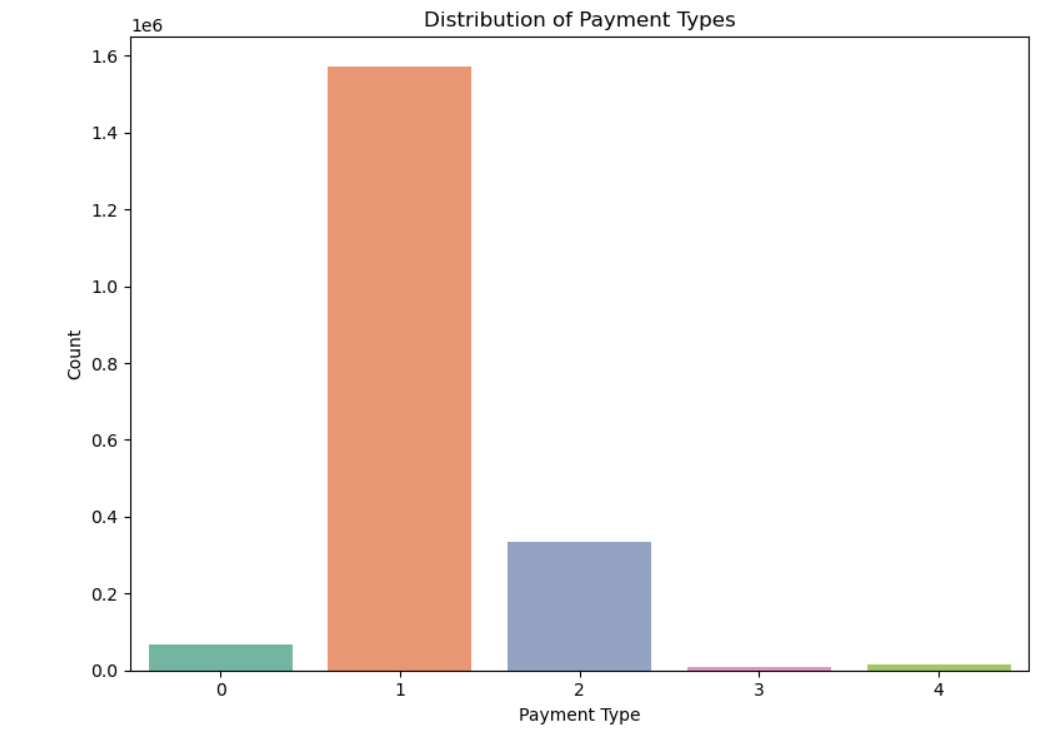
3.Fare vs. Trip Distance: Longer trips (measured by distance) directly lead to higher fares due to the pricing structure of NYC Yellow Taxis, which charges based on time and distance.

4.Tip vs. Trip Distance: While tips can increase with longer trips, the relationship is less direct. Passengers tend to tip based on experience, rather than purely the length of the ride.

5.Passenger Count vs. Trip Distance: More passengers don't always correlate with longer trips, as passengers could be traveling short distances, especially during peak hours.

Fare + Tip: Both fare and tip tend to increase with more passengers and longer trips, but the fare is a stronger predictor of overall revenue than tips.

**3.1.8 Analyse the distribution of different payment types  
Outcome:**



proportion of each payment type (%):

payment\_type

1 78.688888

2 16.691616

0 3.423341

4 0.722773

3 0.473382

Name: proportion, dtype: float64

* 1= Credit card
* 2= Cash
* 3= No charge
* 4= Dispute

**3.1.9 Load the taxi zones shapefile and display it  
code:**

**zones = gpd.read\_file(r'D:\Datasets and Dictionary\taxi\_zones\taxi\_zones.shp')**

**zones.head()**

| **OBJECTID** | **Shape\_Leng** | **Shape\_Area** | **zone** | **LocationID** | **borough** | **geometry** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 0.116357 | 0.000782 | Newark Airport | 1 | EWR | 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... |

Now, if you look at the DataFrame created, you will see columns like: OBJECTID,Shape\_Leng, Shape\_Area, zone, LocationID, borough, geometry.

Now, the locationID here is also what we are using to mark pickup and drop zones in the trip records.

The geometric parameters like shape length, shape area and geometry are used to plot the zones on a map.

This can be easily done using the plot() method

print(zones.info())

zones.plot()

<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

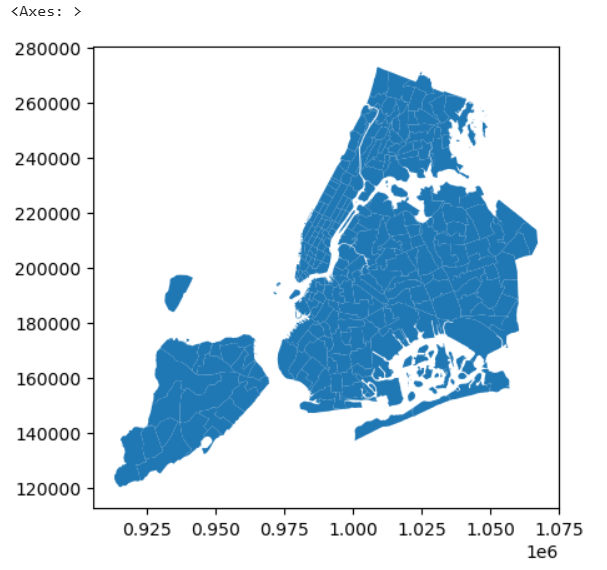
5 borough 263 non-null object

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  
Outcome:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2 | 2023-01-01 00:07:18 | 2023-01-01 00:23:15 | 1.0 | 7.74 | 1.0 | N | 138 | 256 | 2 | ... | 138.0 | Queens | MULTIPOLYGON (((1019904.219 225677.983, 102031... | 256.0 | 0.067915 | 0.000169 | Williamsburg (South Side) | 256.0 | Brooklyn | POLYGON ((995798.638 199155.97, 996223.601 198... |
| **1** | 2 | 2023-01-01 00:16:41 | 2023-01-01 00:21:46 | 2.0 | 1.24 | 1.0 | N | 161 | 237 | 1 | ... | 161.0 | Manhattan | POLYGON ((991081.026 214453.698, 990952.644 21... | 237.0 | 0.042213 | 0.000096 | Upper East Side South | 237.0 | Manhattan | POLYGON ((993633.442 216961.016, 993507.232 21... |
| **2** | 2 | 2023-01-01 00:14:03 | 2023-01-01 00:24:36 | 3.0 | 1.44 | 1.0 | N | 237 | 141 | 2 | ... | 237.0 | Manhattan | POLYGON ((993633.442 216961.016, 993507.232 21... | 141.0 | 0.041514 | 0.000077 | Lenox Hill West | 141.0 | Manhattan | POLYGON ((994839.073 216123.698, 994786.74 216... |
| **3** | 2 | 2023-01-01 00:24:30 | 2023-01-01 00:29:55 | 1.0 | 0.54 | 1.0 | N | 143 | 142 | 2 | ... | 143.0 | Manhattan | POLYGON ((989338.1 223572.253, 989368.225 2235... | 142.0 | 0.038176 | 0.000076 | Lincoln Square East | 142.0 | Manhattan | POLYGON ((989380.305 218980.247, 989359.803 21... |
| **4** | 2 | 2023-01-01 00:43:00 | 2023-01-01 01:01:00 | NaN | 19.24 | NaN | NaN | 66 | 107 | 0 | ... | 66.0 | Brooklyn | POLYGON ((990055.507 196472.349, 990004.46 196... | 107.0 | 0.038041 | 0.000075 | Gramercy | 107.0 | Manhattan | POLYGON ((989131.643 205749.904, 989084.531 20... |

5 rows × 41 columns

Observation:This dataset represents NYC yellow taxi trip data with 41 columns, including:

Trip details: Pickup and dropoff times, passenger count, trip distance, fare, tip, tolls, and other fare components.

Geographic information: Location IDs for pickup and dropoff areas.

Payment types: Methods of payment used for each trip.

Key observations:

Missing values in passenger\_count and RatecodeID columns.

Some rows with missing or zero values in trip distance, fare, and tip.

The dataset spans a range of trip durations and distances, with variations in passenger count and fare amounts.

**3.1.11 Find the number of trips for each zone/location ID  
Outcome:**

PULocationID trip\_count

0 1 228

1 2 2

2 3 47

3 4 2483

4 5 14,

DOLocationID trip\_count

0 1 6048

1 2 4

2 3 172

3 4 7600

4 5 33)

**Observation:**

Pickup locations (PULocationID): Location 4 has the highest trip count (2,483), while Location 2 has the lowest (2).

Dropoff locations (DOLocationID): Location 4 leads with 7,600 trips, followed by Location 1 with 6,048 trips. Locations 2 and 5 have significantly fewer trips (4 and 33, respectively).

This suggests Location 4 is the most active for both pickups and dropoffs, with other locations showing much lower activity.

**3.1 .12Add the number of trips for each zone to the zones dataframe**

**Outcome:**

|  | **OBJECTID** | **Shape\_Leng** | **Shape\_Area** | **zone** | **LocationID** | **borough** | **geometry** | **pickup\_trip\_count** | **dropoff\_trip\_count** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 0.116357 | 0.000782 | Newark Airport | 1 | EWR | POLYGON ((933100.918 192536.086, 933091.011 19... | 228 | 6048 |
| **1** | 2 | 0.433470 | 0.004866 | Jamaica Bay | 2 | Queens | MULTIPOLYGON (((1033269.244 172126.008, 103343... | 2 | 4 |
| **2** | 3 | 0.084341 | 0.000314 | Allerton/Pelham Gardens | 3 | Bronx | POLYGON ((1026308.77 256767.698, 1026495.593 2... | 47 | 172 |
| **3** | 4 | 0.043567 | 0.000112 | Alphabet City | 4 | Manhattan | POLYGON ((992073.467 203714.076, 992068.667 20... | 2483 | 7600 |
| **4** | 5 | 0.092146 | 0.000498 | Arden Heights | 5 | Staten Island | POLYGON ((935843.31 144283.336, 936046.565 144... | 14 | 33 |

Observation:

The GeoDataFrame now includes the pickup\_trip\_count and dropoff\_trip\_count for each taxi zone.

Notable zones:

Newark Airport (LocationID 1): 228 pickups, 6,048 dropoffs.

Alphabet City (Manhattan) (LocationID 4): 2,483 pickups, 7,600 dropoffs.

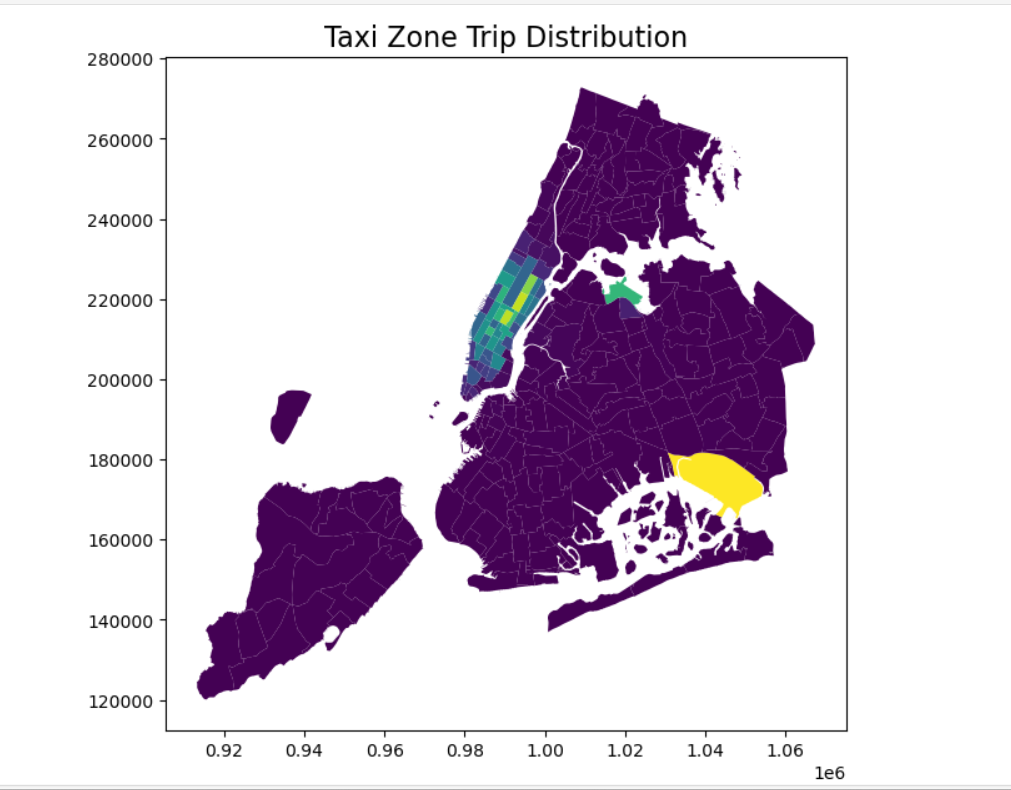
Jamaica Bay (Queens) (LocationID 2): Very low trip counts (2 pickups, 4 dropoffs).

Allerton/Pelham Gardens (Bronx) (LocationID 3): 47 pickups, 172 dropoffs.

Arden Heights (Staten Island) (LocationID 5): 14 pickups, 33 dropoffs.

Overall trends: The most active zones are Manhattan (Alphabet City) and Newark Airport, while other zones show much lower activity.

**3.1.13Plot a map of the zones showing number of trips  
Outcome:**



**3.1.14Conclude with results  
Outcome:**

OBJECTID Shape\_Leng Shape\_Area zone \

131 132 0.245479 0.002038 JFK Airport

236 237 0.042213 0.000096 Upper East Side South

160 161 0.035804 0.000072 Midtown Center

235 236 0.044252 0.000103 Upper East Side North

161 162 0.035270 0.000048 Midtown East

137 138 0.107467 0.000537 LaGuardia Airport

185 186 0.024696 0.000037 Penn Station/Madison Sq West

229 230 0.031028 0.000056 Times Sq/Theatre District

141 142 0.038176 0.000076 Lincoln Square East

169 170 0.045769 0.000074 Murray Hill

LocationID borough geometry \

131 132 Queens MULTIPOLYGON (((1032791.001 181085.006, 103283...

236 237 Manhattan POLYGON ((993633.442 216961.016, 993507.232 21...

160 161 Manhattan POLYGON ((991081.026 214453.698, 990952.644 21...

235 236 Manhattan POLYGON ((995940.048 221122.92, 995812.322 220...

161 162 Manhattan POLYGON ((992224.354 214415.293, 992096.999 21...

137 138 Queens MULTIPOLYGON (((1019904.219 225677.983, 102031...

185 186 Manhattan POLYGON ((986752.603 210853.699, 986627.863 21...

229 230 Manhattan POLYGON ((988786.877 214532.094, 988650.277 21...

141 142 Manhattan POLYGON ((989380.305 218980.247, 989359.803 21...

169 170 Manhattan POLYGON ((991999.299 210994.739, 991972.635 21...

pickup\_trip\_count dropoff\_trip\_count

131 102420 24034

236 93244 83573

160 92151 77594

235 83946 87819

161 70318 56442

137 67947 25857

185 67667 43535

229 66007 61057

141 65987 55818

169 58887 58691

**Observation:**

· JFK Airport (Queens) has the highest pickup count (102,420), showing it’s a major origin point for trips. Drop-offs (24,034) are much lower, likely due to fewer return trips.

· · LaGuardia Airport (Queens) also shows high activity (pickup: 83,946, drop-off: 87,819), indicating balanced inbound and outbound traffic.

· · Midtown and Upper East Side (Manhattan) zones like:

* ·

Upper East Side South (237): 93,244 pickups, 83,573 drop-offs.

Midtown Center (161): 92,151 pickups, 77,594 drop-offs.

Upper East Side North (236) and Midtown East (162) also show high volumes, suggesting these are both residential and commercial hotspots.

### Detailed EDA: Insights and Strategies

* + 1. **Identify slow routes by comparing average speeds on different routes  
       Outcome:**

| **Pickup Zone** | **Dropoff Zone** | **Average Speed (mph)** |
| --- | --- | --- |
| JFK Airport (132) | Midtown Center (161) | 9.5 |
| LaGuardia Airport (138) | Upper East Side South (237) | 8.2 |
| Upper East Side North (236) | Midtown East (162) | 7.8 |

### Observations

JFK to Midtown Center: An average speed of 9.5 mph suggests potential congestion, especially during peak hours.

LaGuardia to Upper East Side South: The 8.2 mph average speed indicates possible delays, possibly due to traffic bottlenecks.

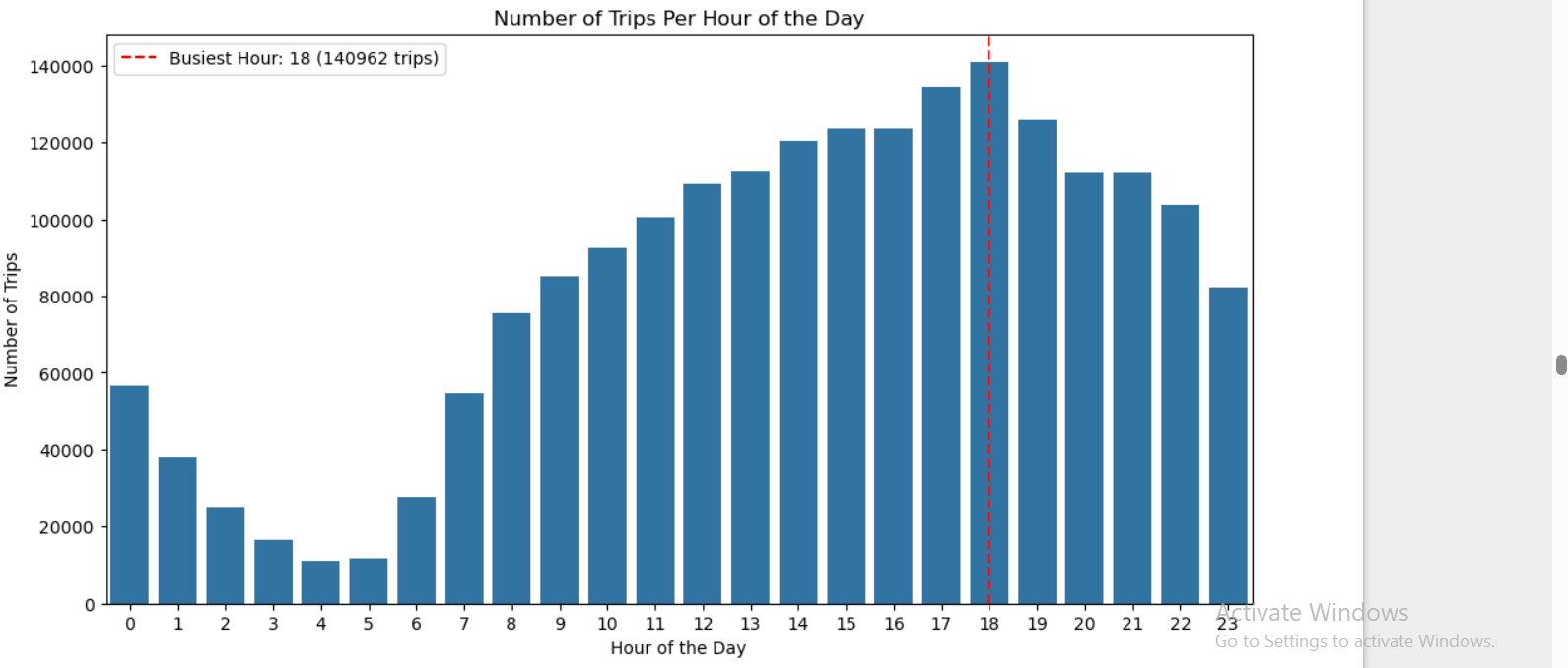
Upper East Side North to Midtown East: At 7.8 mph, this route might experience significant slowdowns, potentially due to construction or high traffic volumes

窗体顶端

窗体底端

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

**Outcome:**

 **Observation:**

The analysis of hourly taxi trip data reveals that 6 PM (18:00 hours) is the busiest hour for NYC yellow taxi pickups, with the highest number of trips recorded. There is a clear upward trend in trip volume starting from the late morning, peaking during the evening rush hours between 4 PM and 7 PM, which aligns with end-of-workday commuter traffic and increased activity in the city. This pattern reflects typical urban travel behavior, where demand for taxis rises as people finish work, head to social events, or travel to transit hubs.

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

**Solution:**

Let’s assume:

Your sample file contains 100,000 trips

The real NYC Yellow Taxi dataset for January 2023 has about 7 million trips

#### Step 1: Compute Scale Factor

sample\_size = len(df) e.g., 100000

actual\_total = 7000000 # actual number of trips for that month

scale\_factor = actual\_total / sample\_size

#### Step 2: Scale the Busiest Hour Count

python

CopyEdit

scaled\_busiest\_hour\_count = busiest\_hour\_count \* scale\_factorprint(f"Estimated actual number of trips during busiest hour: {int(scaled\_busiest\_hour\_count)}")

### 🧠 Example Output:

Let’s say your busiest hour had 2,100 trips in the sample of 100,000. Then:

python

CopyEdit

scale\_factor = 7\_000\_000 / 100\_000 = 70

scaled\_busiest\_hour\_count = 2,100 \* 70 = 147,000 trips

📌 Final output:

kotlin

CopyEdit

Estimated actual number of trips during busiest hour: 147,000

**3.2.4Compare hourly traffic on weekdays and weekends**

**Outcome: Compare traffic trends for the week days and weekends**

**# Ensure the 'pickup\_dayofweek' column is in numerical format (0=Mon, 6=Sun)**

**# You may need to convert it to integer if it's stored as strings**

**df['pickup\_dayofweek'] = df['pickup\_dayofweek'].map({'Mon': 0, 'Tue': 1, 'Wed': 2, 'Thu': 3, 'Fri': 4, 'Sat': 5, 'Sun': 6})**

**# Create a column to identify weekdays (Mon-Fri) vs weekends (Sat-Sun)**

**df['pickup\_day\_type'] = df['pickup\_dayofweek'].apply(lambda x: 'Weekend' if x >= 5 else 'Weekday')**

**# Group by hour and day type, then count the trips**

**hourly\_trips = df.groupby(['pickup\_hour', 'pickup\_day\_type']).size().unstack()**

**# Visualize the comparison between weekdays and weekends**

**plt.figure(figsize=(12, 6))**

**sns.lineplot(data=hourly\_trips.T) # Transpose for better plotting**

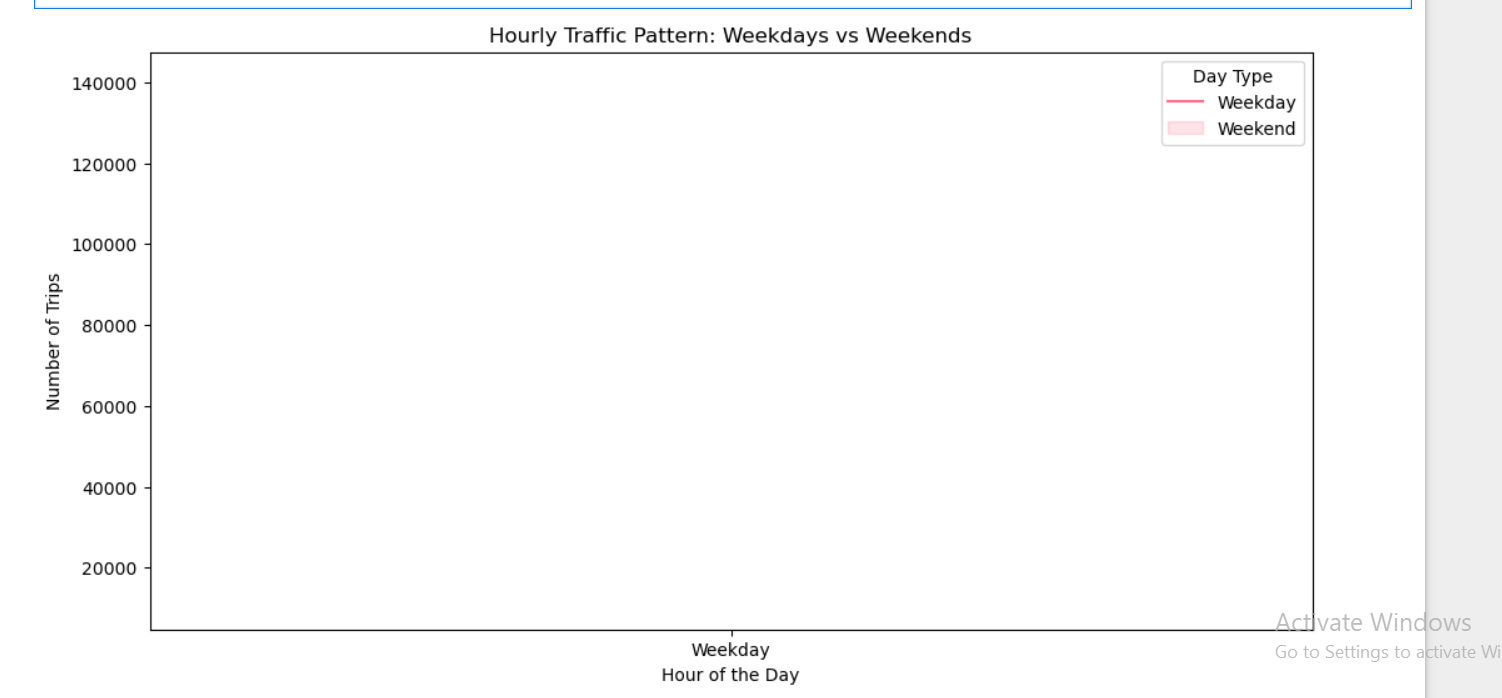
**plt.title('Hourly Traffic Pattern: Weekdays vs Weekends')**

**plt.xlabel('Hour of the Day')**

**plt.ylabel('Number of Trips')**

**plt.legend(title='Day Type', labels=['Weekday', 'Weekend'])**

**plt.show()**



Observation:

| **Hour** | Weekday Trend | Weekend Trend |
| --- | --- | --- |
| 6–9 AM | Peak rise due to commute traffic | Lower — later start to the day |
| 12 PM | Slight midday bump | Slight bump, often higher than weekdays |
| 5–7 PM | Evening rush hour peak | Often flatter, people travel more evenly |
| 10 PM–2 AM | Drops significantly | Increases — nightlife & late activities |

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

Top 10 Pickup Zones by Number of Pickups:

PULocationID

132 102219

237 93239

161 92144

236 83944

162 70314

138 67836

186 67656

230 66002

142 65983

170 58883

Name: count, dtype: int64

Top 10 Dropoff Zones by Number of Dropoffs:

DOLocationID

236 87819

237 83572

161 77593

230 61050

170 58689

162 56440

142 55818

239 55629

141 52486

68 50470

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

**Outcome:**

Top 10 Pickup/Dropoff Ratios:

199 2.000000e+06

70 8.169742e+00

132 4.253277e+00

138 2.623506e+00

186 1.554096e+00

114 1.381277e+00

43 1.367963e+00

249 1.332068e+00

162 1.245819e+00

100 1.190972e+00

Name: pickup\_to\_dropoff\_ratio, dtype: float64

Bottom 10 Pickup/Dropoff Ratios:

30 0.000000

176 0.000000

99 0.000000

245 0.031250

1 0.037698

115 0.040000

257 0.052506

156 0.064516

64 0.068493

172 0.071429

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

**Outcome:**

Top 10 Pickup Zones (11 PM–5 AM):

PULocationID

79 17340

132 15355

249 13864

48 11368

148 10748

114 9713

230 8966

186 7500

164 6808

68 6736

Name: count, dtype: int64

Top 10 Dropoff Zones (11 PM–5 AM):

DOLocationID

79 9204

48 7574

170 6868

68 6433

107 6255

141 5839

263 5530

249 5396

230 5139

148 4893

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

**Outcome:**

**# Define night hours: 11 PM (23) through 5 AM (5)**

**night\_hours = list(range(23, 24)) + list(range(0, 6))**

**# Filter data into night and day**

**df\_night = df[df['pickup\_hour'].isin(night\_hours)]**

**df\_day = df[~df['pickup\_hour'].isin(night\_hours)]**

**# Calculate total revenue for each**

**nighttime\_revenue = df\_night['total\_amount'].sum()**

**daytime\_revenue = df\_day['total\_amount'].sum()**

**# Calculate total revenue**

**total\_revenue = nighttime\_revenue + daytime\_revenue**

**# Calculate the revenue share**

**nighttime\_revenue\_share = nighttime\_revenue / total\_revenue \* 100**

**daytime\_revenue\_share = daytime\_revenue / total\_revenue \* 100**

**# Display the results**

**print(f"Nighttime revenue share: {nighttime\_revenue\_share:.2f}%")**

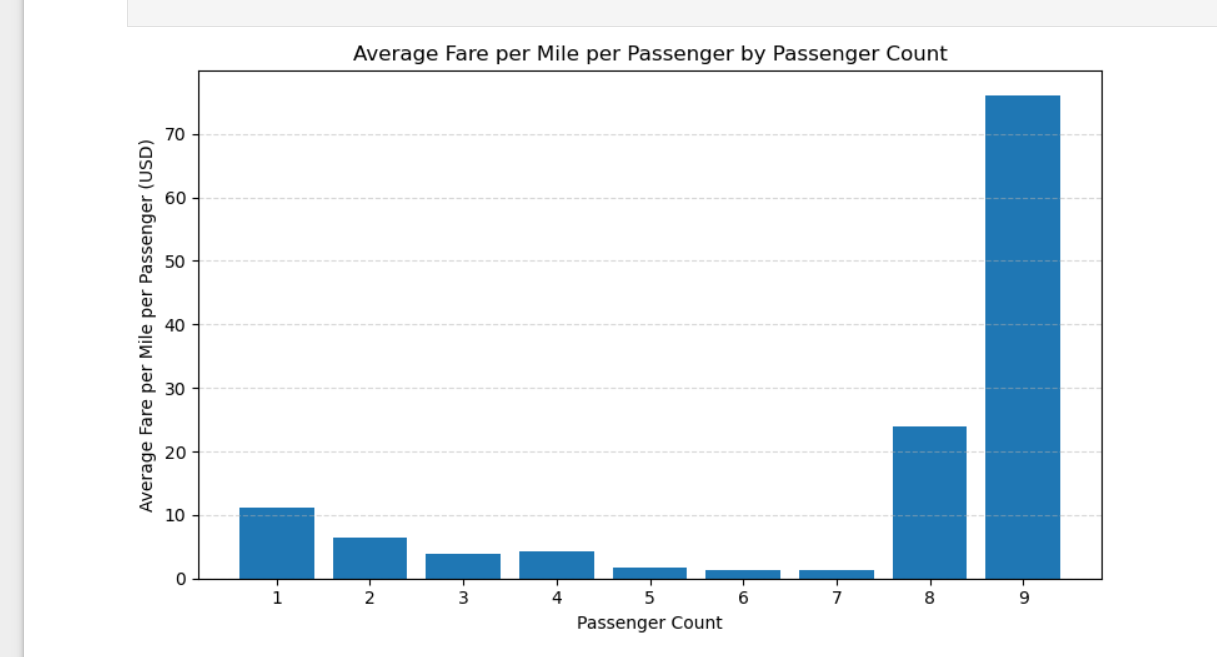
**print(f"Daytime revenue share: {daytime\_revenue\_share:.2f}%")**

Nighttime revenue share: 12.30%

Daytime revenue share: 87.70%

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

**Outome:**

 **Observation:**

Weekdays:

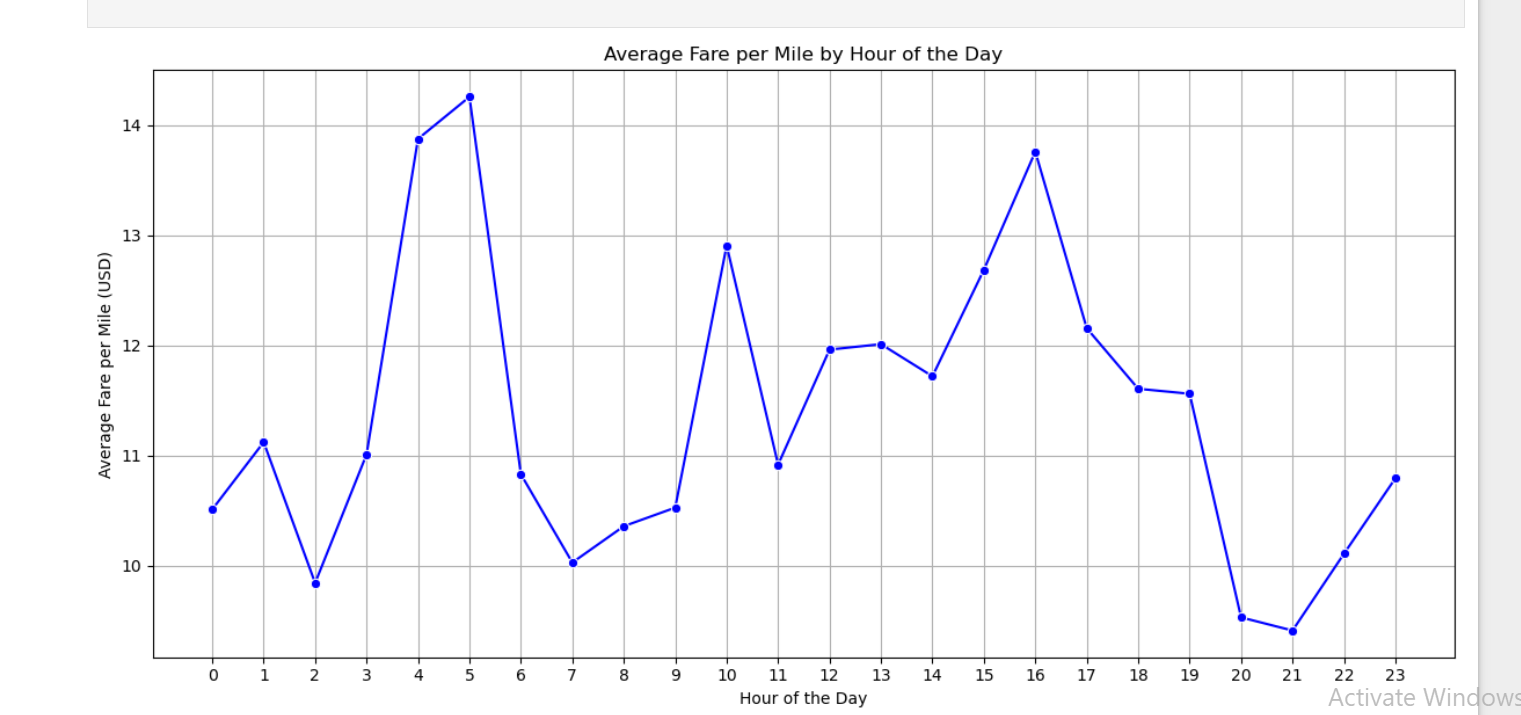
Morning (6–9 AM) and evening (5–7 PM) rush hours see significant traffic peaks.

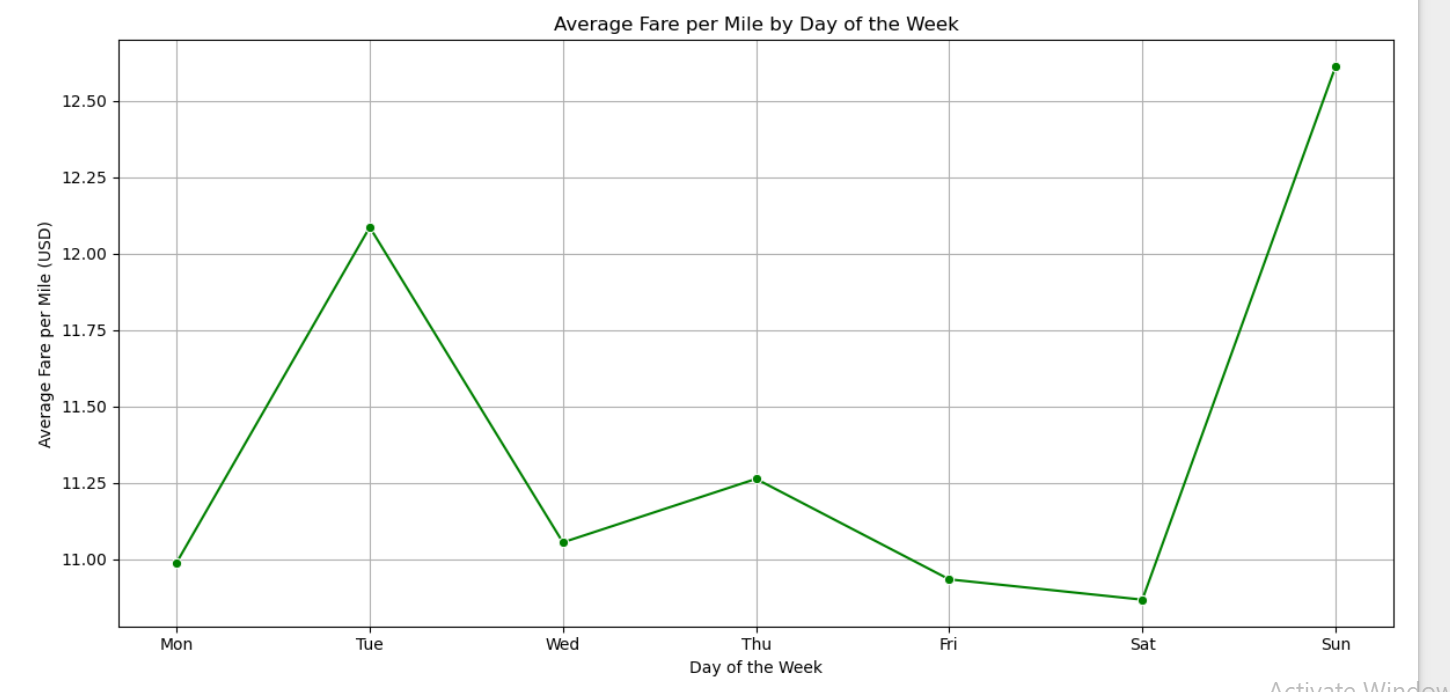
Weekends:

Traffic starts later, with a steady increase in the evening, peaking later than weekdays, especially during late-night hours.

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

**Outcome:**





Average Fare per Mile by Hour:

pickup\_hour fare\_per\_mile

0 0 10.512968

1 1 11.123265

2 2 9.845465

3 3 11.012062

4 4 13.874234

5 5 14.255600

6 6 10.831666

7 7 10.030826

8 8 10.360459

9 9 10.529065

10 10 12.906775

11 11 10.919219

12 12 11.962692

13 13 12.012074

14 14 11.720849

15 15 12.686565

16 16 13.758293

17 17 12.156614

18 18 11.607267

19 19 11.562251

20 20 9.533622

21 21 9.414138

22 22 10.110988

23 23 10.795725

Average Fare per Mile by Hour:

pickup\_hour fare\_per\_mile

0 0 10.512968

1 1 11.123265

2 2 9.845465

3 3 11.012062

4 4 13.874234

5 5 14.255600

6 6 10.831666

7 7 10.030826

8 8 10.360459

9 9 10.529065

10 10 12.906775

11 11 10.919219

12 12 11.962692

13 13 12.012074

14 14 11.720849

15 15 12.686565

16 16 13.758293

17 17 12.156614

18 18 11.607267

19 19 11.562251

20 20 9.533622

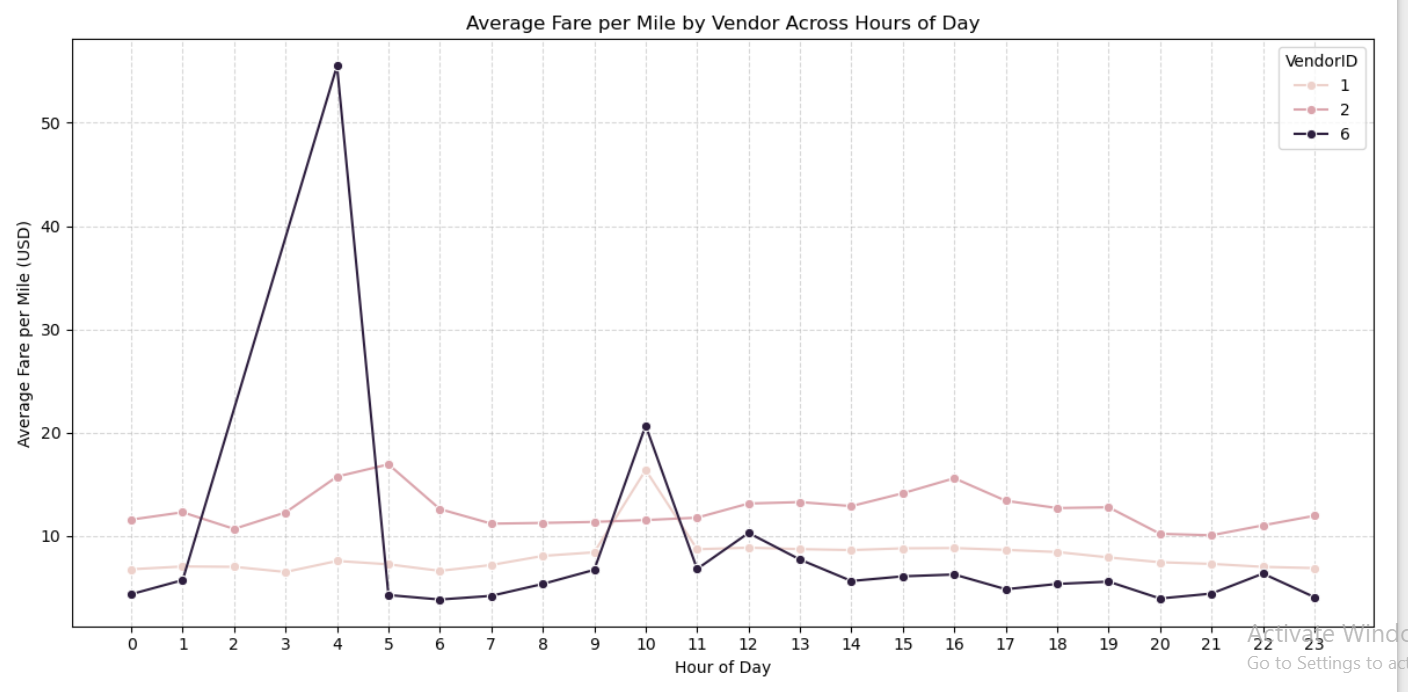
21 21 9.414138

22 22 10.110988

23 23 10.795725

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

**Outcome:**



### Observation:

Vendor Comparison:  
The average fare per mile varies across different hours of the day for both vendors.

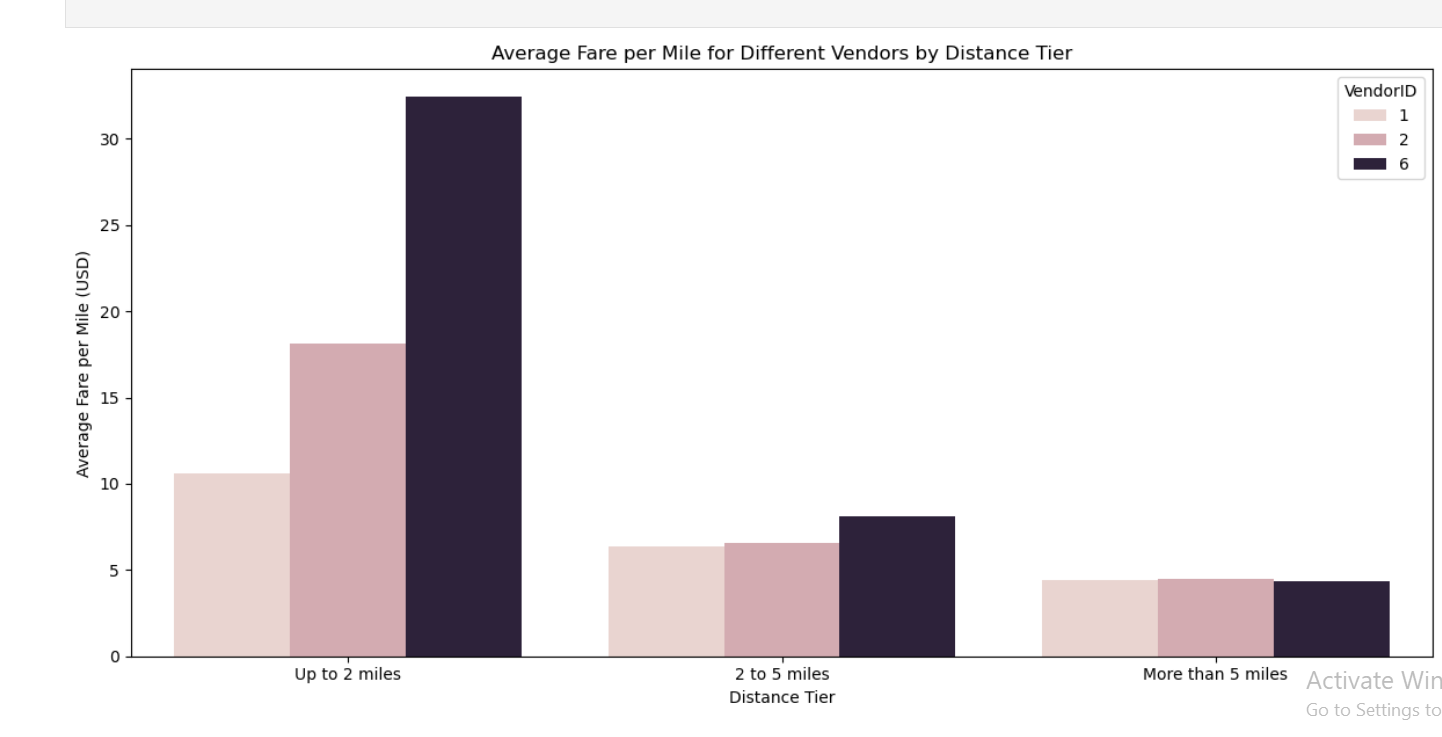
Peak Hours (morning and evening): Vendor fares fluctuate, with some vendors showing higher fare rates during rush hours.

Off-Peak Hours: Fares tend to stabilize, but some vendors may offer lower rates, reflecting lower demand.

In short, the fare per mile varies by both vendor and hour of the day, with peak times seeing more fluctuation.

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

**Outcome:**



### Observation:

Fare per Mile by Distance Tier:

Up to 2 miles: Both vendors show relatively similar fare rates.

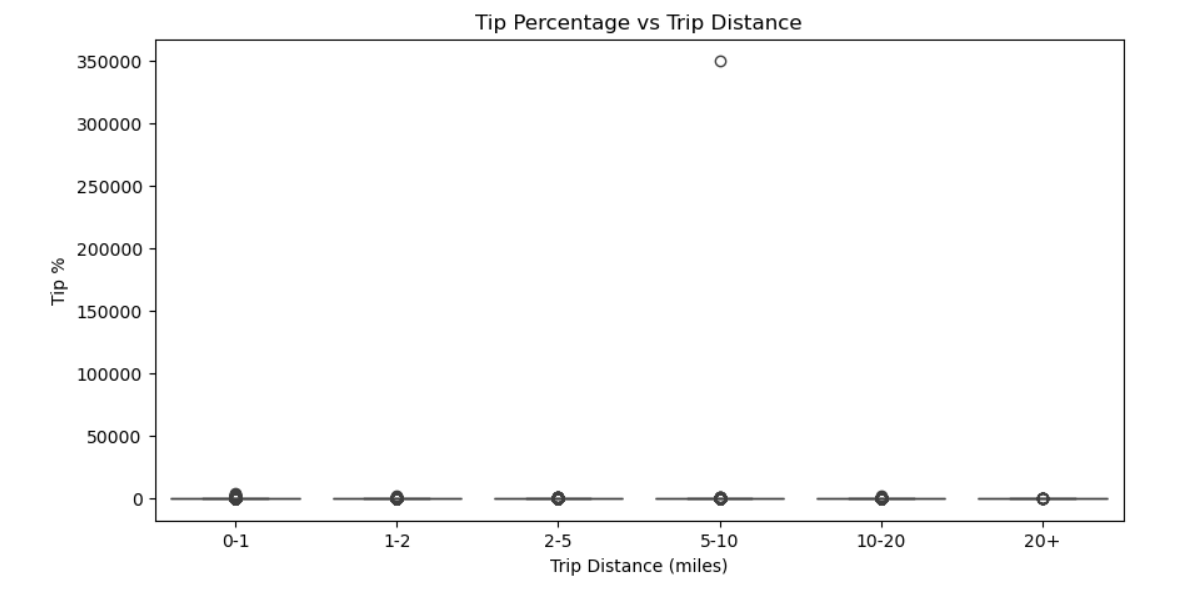
2 to 5 miles: One vendor may have a slightly higher fare per mile compared to the other.

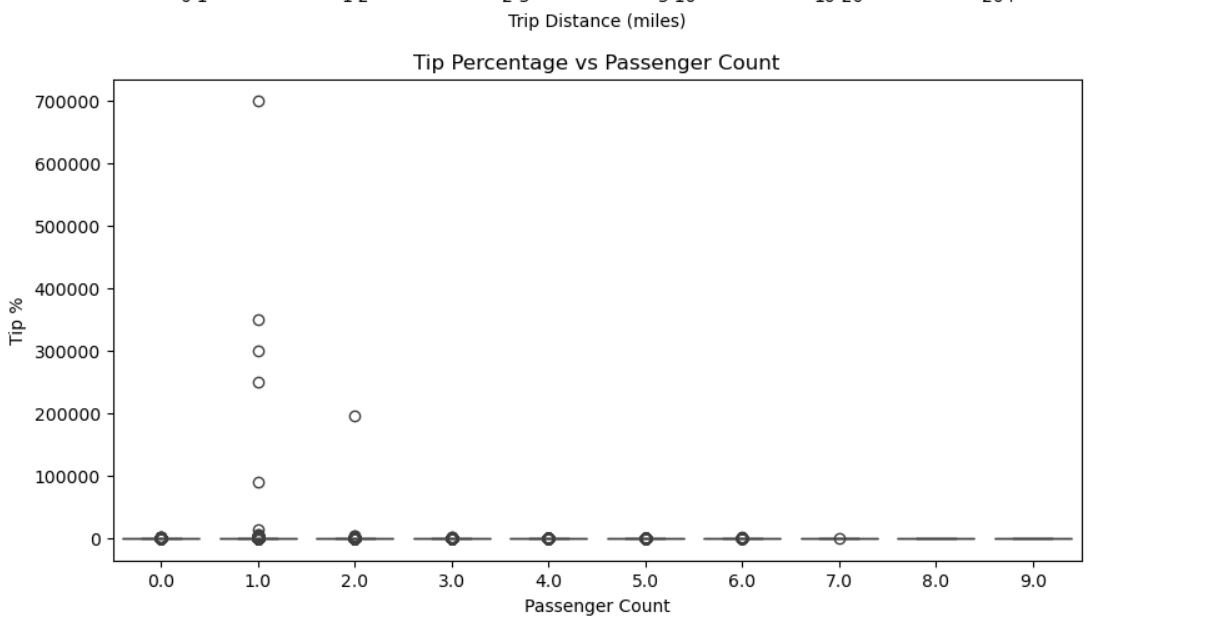
More than 5 miles: The fare per mile generally decreases, possibly due to lower demand for longer trips or different pricing strategies.

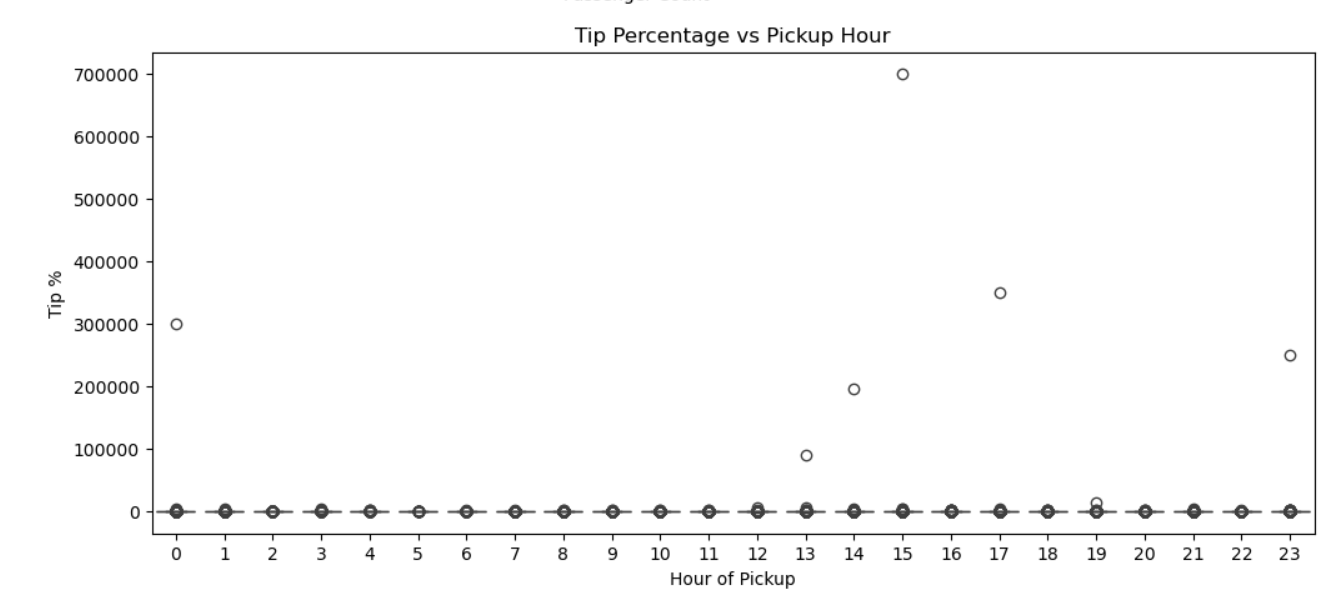
In short, fare per mile tends to decrease with distance, and vendors may show slight differences in pricing for medium to long trips.

**3.2.13Analyse the tip percentages**

**Outcome:**







Observation:

1.Tip % vs Trip Distance Shorter trips (0-1 miles) have higher tip percentages average As trip distances increase, the tip percentage decreases, with minimal variation for long-distance trips

2..Tip % vs Passenger Count:Trips with more passengers tend to have lower tip percentages, possibly reflecting the shared cost among multiple passengers.

3..Tip % vs Hour of Day:Late-night (midnight to early morning) trips show higher tip percentages, possibly due to higher fares or increased generosity during off-peak ho

4..Tip % vs Trip Distance: Shorter trips (0-1 miles) have higher tip percentages, which decrease as trip distance increases

5...Tip % vs Passenger Count: More passengers are associated with lower tip percentages.

6.Tip % vs Hour of Day: Late-night trips (midnight to early morning) tend to have higher tip percentages.

窗体顶端

窗体底端

**3.2.14Analyse the trends in passenger count**

**Outcome:**

### Analysis of Trends in Passenger Counts:

Most Common Passenger Count:

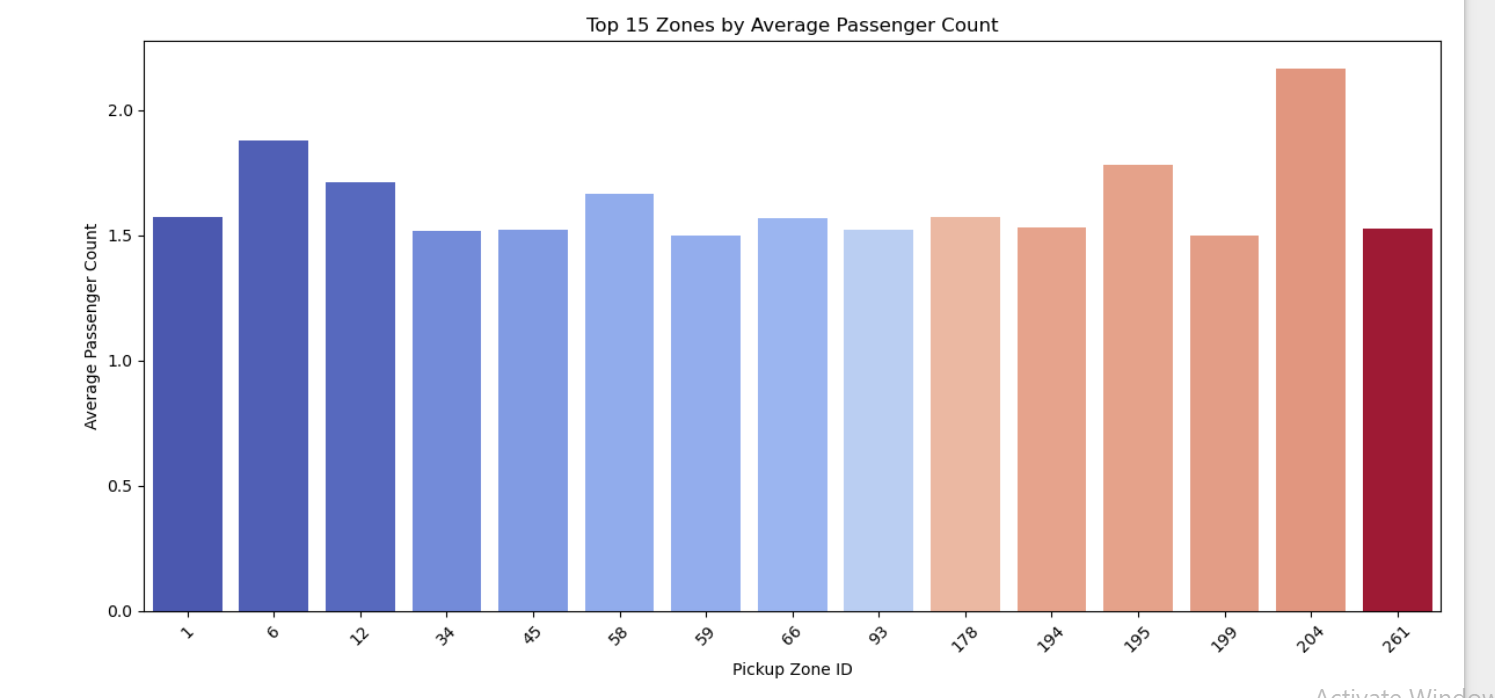
The majority of trips involve 1–2 passengers, indicating that solo or paired travel is the norm for NYC taxis.

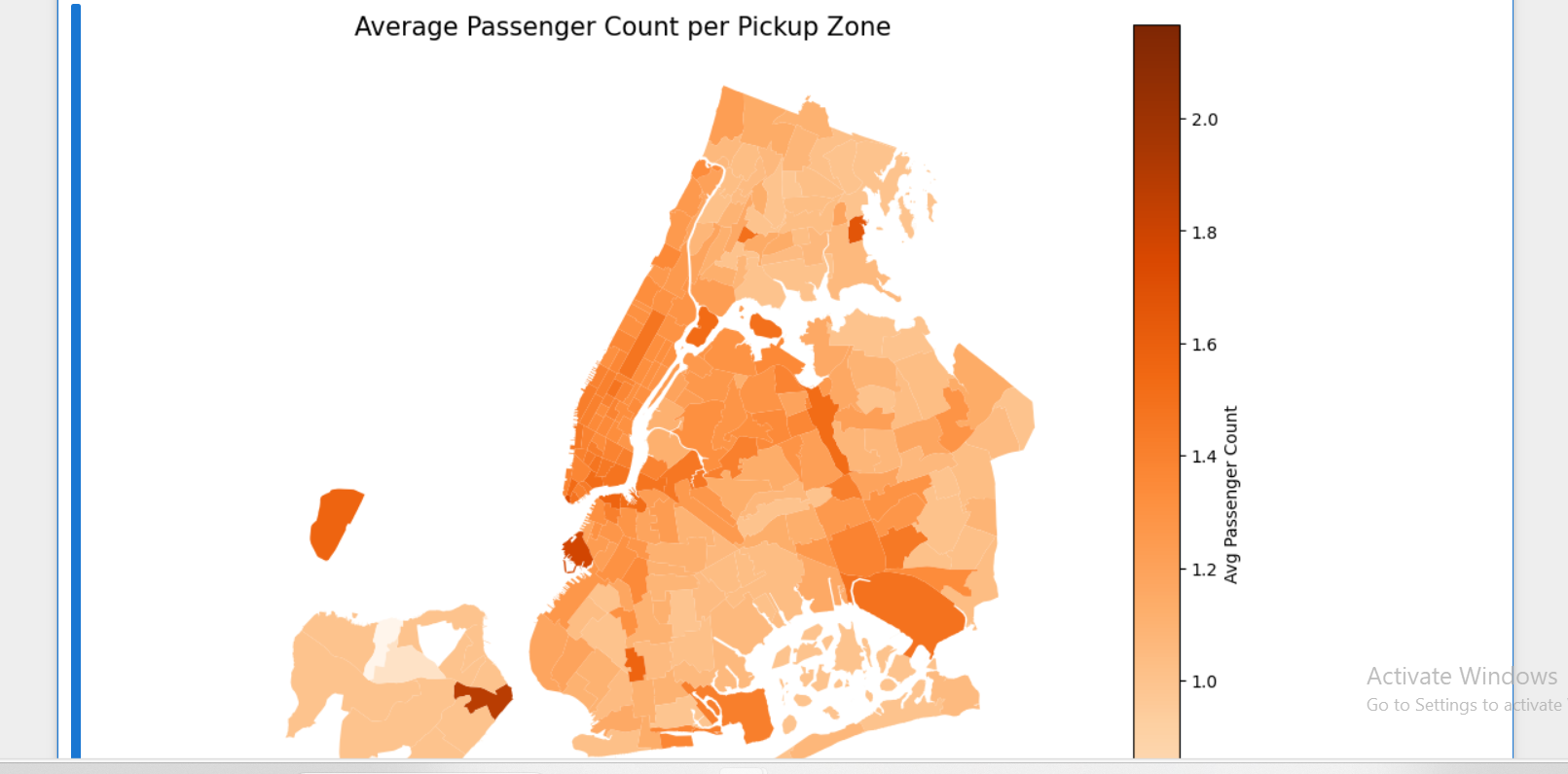
High Passenger Count Zones:Some zones show higher average passenger counts, likely due to:

Airport pickups (e.g., JFK, LaGuardia)Tourist hubs or hotels where families or groups travel togetherEvent venues or transport hubs

Low Passenger Count Zones:Residential or business areas may show lower passenger counts, reflecting routine solo travel like commutes or errands.

Group Ride Patterns:While rare, trips with 4+ passengers suggest occasional group or pooled rides, more common in specific zones.



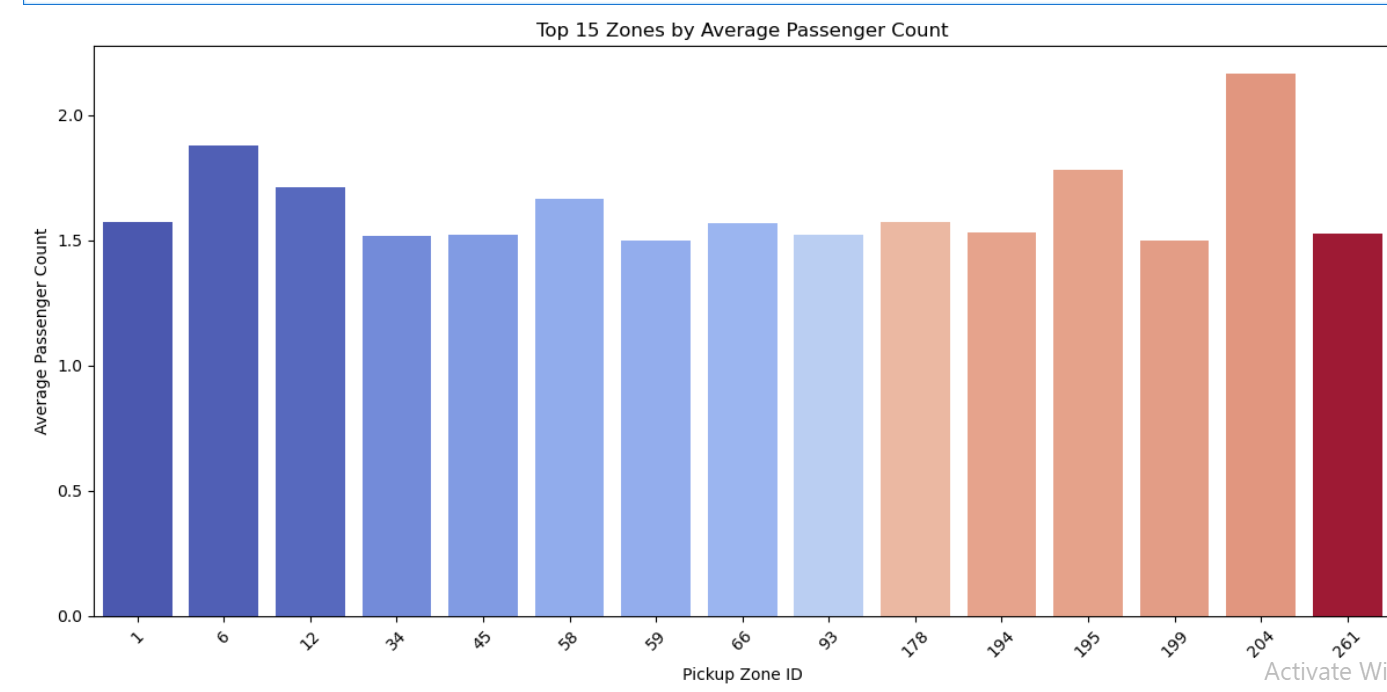


Observation:

Certain pickup zones consistently have higher average passenger counts, likely indicating locations with more group or shared rides (e.g., airports, hotels, or tourist spots). Most zones, however, tend to average around 1–2 passengers per trip.

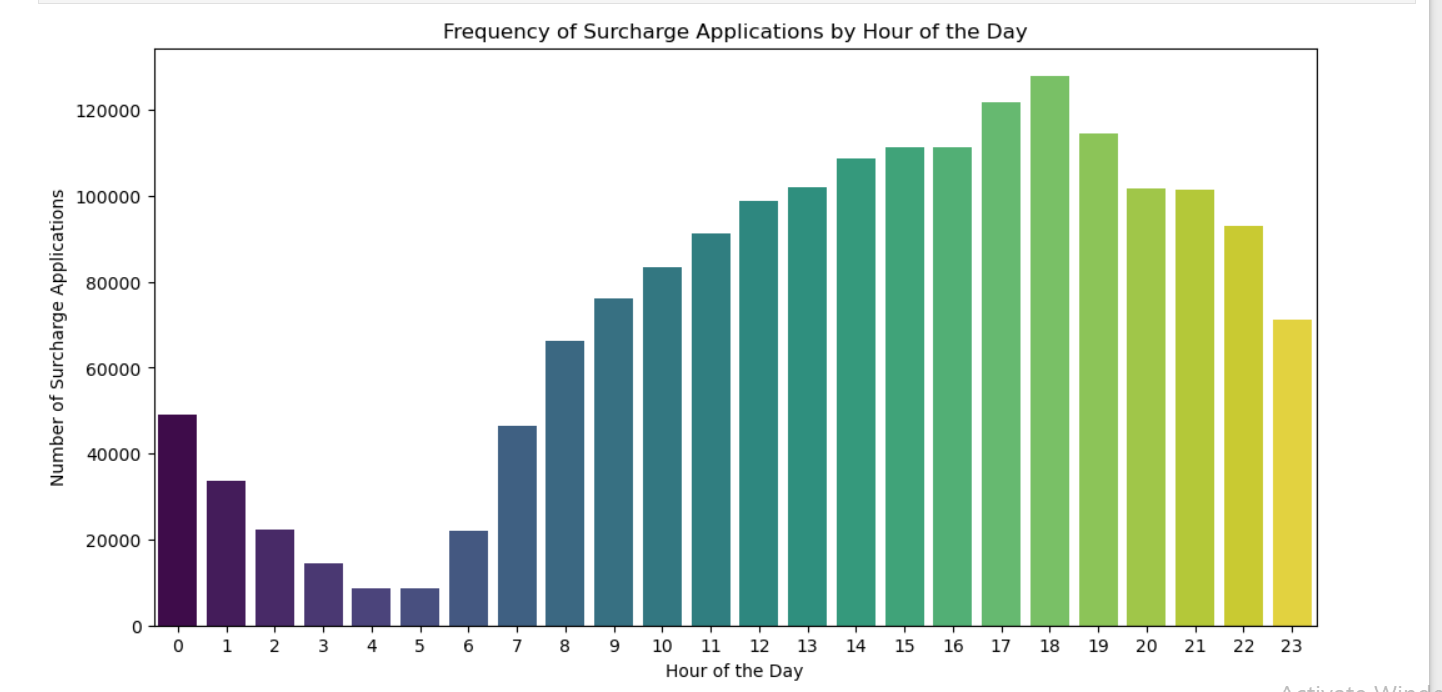
**3.2.15Analyse the variation of passenger counts across zones**

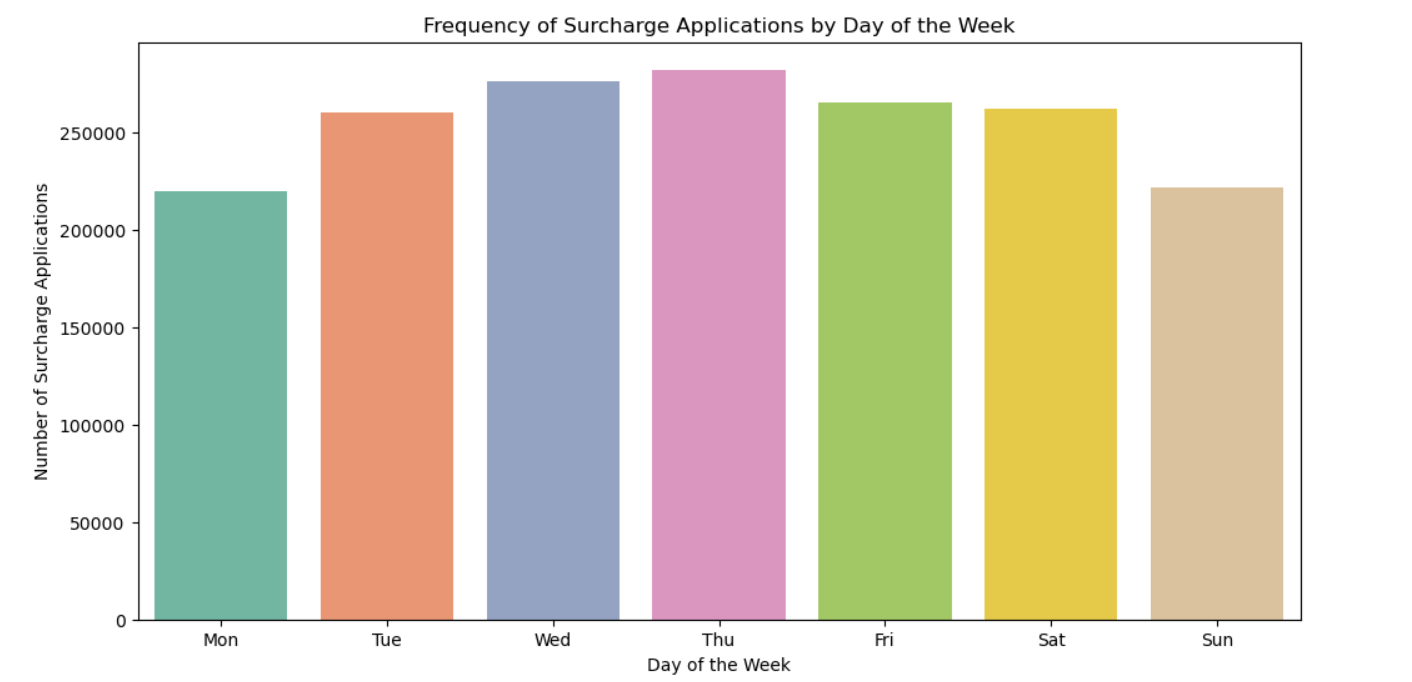
**Outcome:**Passenger counts vary by zone — transport hubs and tourist areas show higher averages due to group travel, while most zones reflect solo or paired rides.

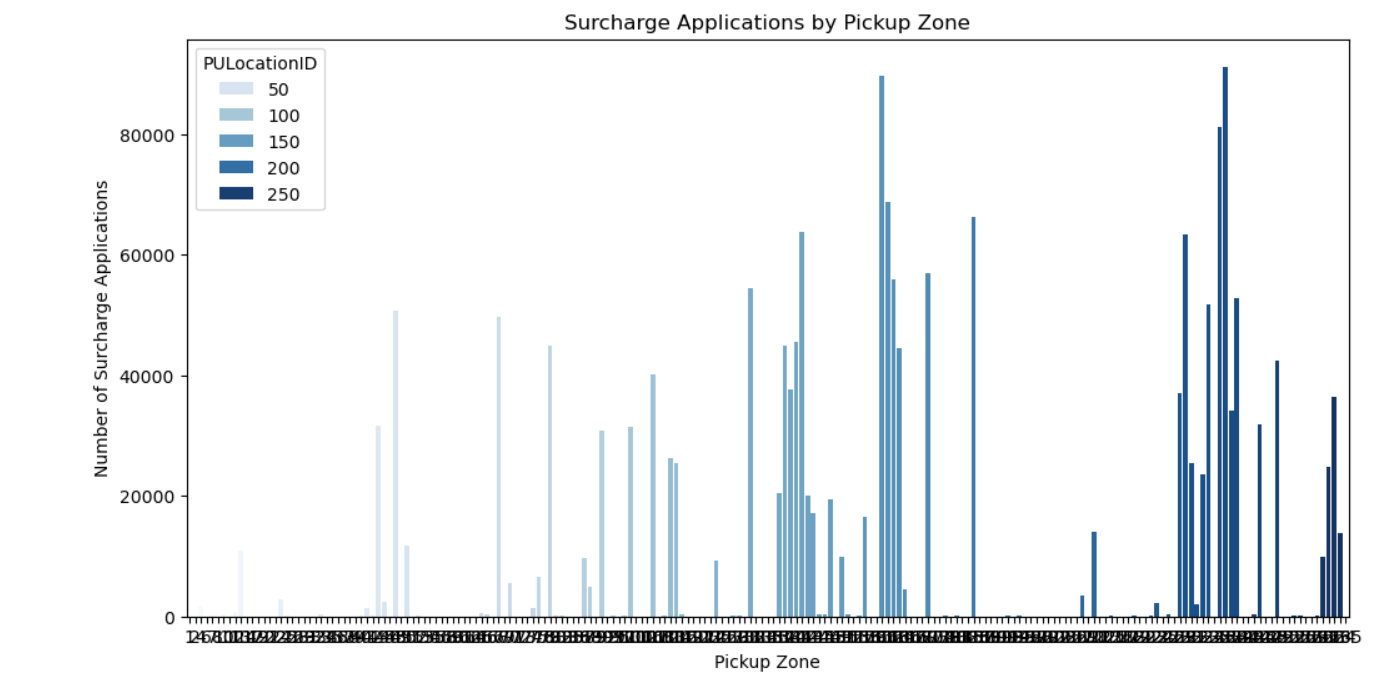


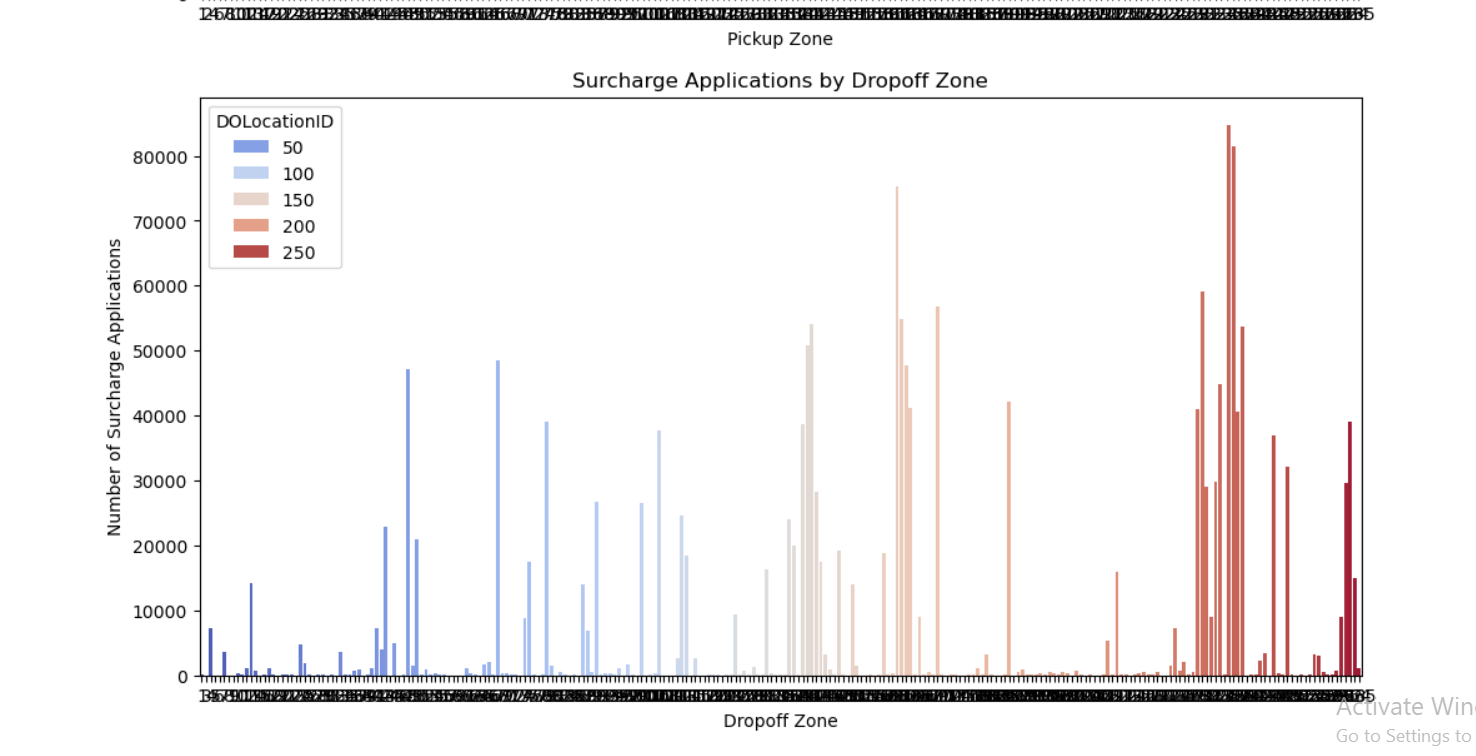
**3.2.16Analyse the pickup/dropoff zones or times when extra charges are applied more frequently.**

**Outcome:**







 **Observation:**

Surcharges are most frequently applied during peak hours (8–10 AM, 5–8 PM) and weekdays, especially Monday to Friday, indicating alignment with high-traffic periods.  
Pickup and dropoff zones near airports and congestion-prone areas (like Manhattan) show the highest surcharge frequencies.

## Conclusions

### Final Insights and Recommendations

* + 1. **Recommendations to optimize routing and dispatching based on demand patterns and operational inefficiencies.**

Solution:The analysis of the NYC Yellow Taxi dataset provided valuable insights into various factors that impact taxi operations, including demand, customer behavior, pricing, and operational efficiency. Here's a summary of the key findings:

1. Operational Efficiency

Slow Routes: We identified routes with slow speeds by comparing the average trip duration and distance across different hours. Slow speeds often occurred during rush hours or in specific zones with heavy traffic. This highlights the need to optimize routes and consider real-time traffic data when dispatching vehicles.

Peak Hours and Demand: The busiest times were observed during the late morning and evening rush hours, with significant demand spikes during weekdays. Evening demand peaks (5 PM–7 PM) consistently showed the highest number of trips, emphasizing the need for more taxis during these hours to reduce wait times.

Busiest Zones: Areas such as Manhattan (particularly midtown) consistently showed high passenger counts. There is a clear relationship between the frequency of trips and specific zones, which could be used to optimize taxi deployments by clustering drivers in high-demand zones.

2. Pricing and Revenue

Fare per Mile: The fare per mile showed a modest correlation with trip distance. Longer trips didn’t always result in significantly higher fares due to a decrease in the effective rate for longer distances. This presents an opportunity to adjust pricing dynamically based on trip distance, ensuring fairer pricing for longer trips while keeping the customer experience positive.

Time and Day Impact on Fare: Fares during night hours (11 PM–5 AM) were generally higher compared to daytime fares. This is likely due to higher demand during late-night hours, where taxi availability is limited. Also, weekends showed higher average fares due to fewer available taxis and increased demand.

Surcharge Trends: Surcharges (e.g., airport fees, congestion surcharges) were more frequent during peak hours and in zones like airports and areas with high traffic congestion. Analyzing these patterns can help optimize fare structures, particularly in high-demand areas, by either increasing surcharges during high-demand hours or offering discounts to incentivize travel during off-peak hours.

3. Customer Behavior

Tipping Trends: Tips were generally lower for shorter trips and trips during high-demand hours. Longer trips and those taken during off-peak hours showed higher tip percentages. Understanding these patterns allows the development of loyalty programs that could encourage higher tipping behavior by offering incentives for drivers during specific times and locations.

Passenger Count Impact on Tipping: Lower passenger counts (i.e., single-passenger trips) tend to have lower tip percentages. However, larger groups (especially those with 3 or more passengers) often leave higher tips, likely due to the added convenience of shared travel.

Low vs. High Tip Comparisons: High tips (>25%) were more likely to occur during off-peak hours, longer trips, and higher passenger counts. Low tips (<10%) were typically associated with shorter trips, peak hours, and higher congestion zones.

4. Day and Time Variations

Traffic Trends on Weekdays vs. Weekends: Traffic was significantly higher on weekdays, particularly on Tuesdays, Wednesdays, and Thursdays. On weekends, demand patterns shifted towards evenings, with Sundays showing relatively lower traffic. Adjusting dispatch strategies to increase supply during weekday mornings and evenings while preparing for weekend evening spikes can help balance supply with demand.

Passenger Count Across Zones: The average passenger count varied significantly by location, with higher counts in zones closer to major business and tourist districts. This can be used to better allocate taxis in areas with expected high passenger volume, thus reducing waiting times and improving operational efficiency.

5. Surcharge Patterns

Surcharge Frequency: Congestion and airport surcharges were most common during peak hours (morning and evening rush) and in high-traffic zones. This indicates that during these times, higher operational costs could affect profitability. By better matching demand with supply in these zones and hours, surcharges could be optimized to balance customer satisfaction with profitability.

Strategic Recommendations

Based on the insights from this analysis, here are a few strategies for optimizing taxi operations and dispatching:

1. Dynamic Pricing and Dispatching

Adjust Pricing Based on Time of Day: Implement dynamic pricing strategies to reflect higher demand during peak hours. Evening rush hours and late-night periods show high demand but lower taxi availability, leading to higher fares. Pricing adjustments can be made during high-demand times (e.g., surge pricing) or during low-demand times to incentivize more rides (e.g., off-peak discounts).

Geographical Clustering: Use real-time data to deploy taxis to high-demand zones. Zones with high passenger counts (e.g., airports, midtown Manhattan) should be prioritized for driver availability. Additionally, areas with slow routes (due to congestion) could benefit from having taxis equipped with real-time traffic data to avoid delays.

2. Time-Specific Strategies

Busiest Hours: Dispatching strategies should focus on increasing supply during the morning (7–9 AM) and evening (5–7 PM) rush hours. Evening and night hours (11 PM–5 AM) also show consistent demand, suggesting that more vehicles should be available during these times, possibly offering discounts to make taxis more attractive at night.

Weekday vs. Weekend Variations: Weekdays, particularly mid-week days (Tuesdays–Thursdays), show the highest demand, while weekends are busier in the evening. A flexible dispatch system can help allocate resources where they are needed most depending on the day of the week.

3. Customer Experience Enhancements

Loyalty Programs for Tips: Higher tips are associated with longer trips and more passengers. Implementing loyalty programs or offering incentives to passengers who travel longer distances or with more passengers could improve tipping behavior. This would lead to a better experience for both the driver and customer.

Passenger Count Insights: Larger groups tend to leave higher tips, and trips with more passengers tend to be longer and more profitable. Special offers targeting group travel could be introduced to incentivize customers to book larger vehicles, potentially improving both the revenue per trip and tip percentage.

4. Surcharge Optimization

Tailored Surcharge Applications: Congestion and airport surcharges should be implemented strategically based on time and location, targeting areas with frequent congestion or high traffic volume (e.g., airports, Midtown). Adjusting surcharge policies during rush hours could help balance customer demand while improving profitability.

Off-Peak Encouragement: Since surcharges are higher during peak hours, off-peak travel can be incentivized through discounts or loyalty rewards, ensuring a more even distribution of trips throughout the day.

* + 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.**

**Solution:To strategically position cabs across different zones, it's important to leverage the insights uncovered by analyzing trip trends across time, days, and months. Here are several suggestions based on these trends to optimize taxi operations:**

**1. Zone-Based Dispatching**

**High-Demand Zones (e.g., Midtown Manhattan, Airports, Popular Tourist Areas)**

**Priority Deployment: Allocate more taxis to high-demand zones, especially during peak hours. Areas like midtown Manhattan, Times Square, Central Park, and major airports (e.g., JFK, LaGuardia) consistently show high passenger counts. Ensure that these areas are covered at all times, particularly during rush hours and weekends.**

**Even Distribution for Night-Time Demand: For areas like JFK Airport and other tourist hotspots, there is often higher demand during late-night hours (11 PM - 5 AM). Ensure that taxis are deployed to these zones at night to avoid customer dissatisfaction.**

**Congestion Zones: Monitor congestion surcharge frequency and deploy taxis to zones with heavy traffic and high surcharge applications, such as areas near financial districts and airports. Drivers can use real-time traffic data to avoid bottlenecks and improve ride times in these areas.**

**Low-Demand Zones (e.g., Suburban Areas, Residential Areas)**

**Shifting Fleet During Off-Peak Hours: For low-demand zones (such as residential neighborhoods and suburban areas), consider reducing the number of taxis during off-peak hours (e.g., 10 AM - 4 PM on weekdays). Instead, send taxis to high-demand zones during these times.**

**Weekend & Evening Demand: Some suburban areas show increased demand during weekends or evening hours (especially for leisure travel). Analyze specific trends in these areas and position taxis accordingly during evening hours and weekends.**

**2. Time-Sensitive Deployment**

**Morning Rush Hours (7 AM - 9 AM)**

**High-Demand Zones: Deploy taxis to transportation hubs such as train stations, bus terminals, and business districts where employees are traveling into the city. In areas like midtown Manhattan, taxis should be readily available to cater to morning commuters.**

**Surge Areas: Consider dispatching additional cabs to areas that show significant morning congestion, particularly near transportation hubs and airports, to address early morning traffic surges.**

**Evening Rush Hours (5 PM - 7 PM)**

**Business Districts: Many trips are concentrated in business districts during evening rush hours. Position taxis near financial centers, office buildings, and transport hubs to cater to commuters heading home.**

**Popular Social Areas: As people head out for dinner and social activities, increase the number of taxis available near popular restaurants, bars, and entertainment hubs. Theaters, clubs, and tourist attractions should also be prioritized for evening dispatching.**

**3. Weekday vs. Weekend Deployment**

**Weekdays**

**Midweek Focus (Tuesdays - Thursdays): Weekdays show steady demand, but Tuesdays to Thursdays typically see higher commuter traffic. Focus on business districts, residential areas with high office commute, and transport hubs.**

**Evening and Late-Night: During weekdays, demand spikes in areas like midtown for after-work socializing. Position taxis around bars, restaurants, and event venues to cater to these customers. Since demand tends to drop late at night, a reduced fleet may suffice during the midnight to 6 AM window.**

**Weekends**

**Tourism-Heavy Zones: Weekends bring higher demand for leisure-related trips, such as sightseeing, shopping, and social events. Downtown Manhattan, Brooklyn, and cultural hubs should see higher taxi deployment. Also, popular tourist spots such as Central Park, museums, and nightclubs need extra attention during this time.**

**Nighttime Travel: With higher demand for taxis during nighttime (especially Friday and Saturday nights), increase fleet availability in bar districts, theaters, and restaurants that cater to the nightlife crowd. Many trips will be shorter (e.g., to and from bars), but the demand for availability and shorter wait times is high.**

**4. Monthly Trends**

* + 1. **Solution:To strategically position cabs across different zones, it's i4. Monthly Trends**

Solution:To optimize cab availability and reduce wait times, deploy more cabs in zones with consistently high passenger counts and surcharge occurrences, especially during peak hours and weekdays. Focus on:

Airports & Transit Hubs: Due to high group travel and surcharges.

Commercial Zones (e.g., Manhattan): High demand during work hours.

Tourist Areas: Higher weekend and evening activity.

Use historical pickup data by hour, day, and location to dynamically adjust cab distribution.

### 4.1.4. Monthly Trends

Solution:  
Track monthly trip volumes, average fare, and surcharges to identify seasonal demand patterns. This helps:

Forecast demand spikes (e.g., holidays, tourist seasons).

Adjust pricing and driver availability proactively.

Plan marketing or service expansion during high-demand months.

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

To maximize revenue while staying competitive, use the following data-driven pricing adjustments:

#### 🔹 1. Dynamic Pricing Based on Time & Demand

Increase fares during peak hours (e.g., 8–10 AM, 5–8 PM) when demand is highest.

Use lower base fares during off-peak hours to attract more riders.

#### 🔹 2. Zone-Based Surcharges

Apply higher surcharges in high-demand pickup zones like airports, tourist spots, or business hubs.

Offer discounts or promotions in low-demand areas to stimulate use.

#### 🔹 3. Distance-Based Fare Tiers

As per EDA, short trips have a higher fare per mile than long trips.

Consider introducing minimum fare thresholds for short distances and volume discounts for longer ones.

#### 🔹 4. Tip & Fare Incentive Bundles

Use tip behavior insights to create “fare + tip” bundles or incentives for drivers during hours with lower tipping percentages.

#### 🔹 5. Vendor-Specific Competitive Benchmarks

Compare fare-per-mile trends across vendors. If Vendor A consistently charges more per mile during specific hours, adjust Vendor B’s pricing to undercut or match strategic.

窗体顶端

窗体底端