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

To effectively analyze **NYC taxi operations**, the dataset is first loaded, comprising multiple **Parquet files** containing trip details such as **pick-up and drop-off dates/times, locations, trip distances, itemized fares, rate types, payment types, driver-reported passenger counts,** and **additional charges**.

Since the dataset is extensive, **5% sampling** is performed for each **hour of each day of each month** to facilitate preliminary analysis before working with the full dataset.

Loading the dataset, sampling the data, and combining the files:

```
# Iterate through the list of files and sample one by one
for file_name in file_list:
    try:
        # File path for the current file
        file_path = os.path.join(directory_path, file_name)

        # Reading the current file
        file_data = pd.read_parquet(file_path)

        # Convert the date column to datetime if it's not already
        file_data['tpep_pickup_datetime'] = pd.to_datetime(file_data['tpep_pickup_datetime'])

        # Filter data to only include dates within the year 2023
        file_data = file_data[file_data['tpep_pickup_datetime'].between(start_datetime, end_datetime, inclusive='both')]

        # Iterate through each month and concatenate the data
        for month in range(1, 13):
            monthly_temp_data = file_data[file_data['tpep_pickup_datetime'].dt.month == month]
            monthly_data[month] = pd.concat([monthly_data[month], monthly_temp_data])

    except Exception as e:
        print(f"Error reading file {file_name}: {e}")

# Iterate through each month to sample data by hour
for month in range(1, 13):
    final_monthly_data = monthly_data[month]

    # Group by date and hour, then sample
    grouped = final_monthly_data.groupby([final_monthly_data['tpep_pickup_datetime'].dt.date,
                                          final_monthly_data['tpep_pickup_datetime'].dt.hour])
    for (day, hour), hourly_data in grouped:

        # Sample 5% of the hourly data randomly
        hourly_sampled = hourly_data.sample(frac=0.05, random_state=1)
        final_sampled_data = pd.concat([final_sampled_data, hourly_sampled])
```

This results in **1.8 million (18 lakh)** entries, which are then downsampled to maintain a total of **250,000 to 300,000 (2.5 to 3 lakh)** entries, as noted in the **Jupyter notebook comments**.

```
# Proportional Downsampling if Exceeding 299000 (<300000)
TARGET_MAX = 299000

if len(final_sampled_data) > TARGET_MAX:
    print("Downsampling to maintain proper date-hour proportions...")

    grouped = final_sampled_data.groupby([final_sampled_data['tpep_pickup_datetime'].dt.date,
                                          final_sampled_data['tpep_pickup_datetime'].dt.hour])
    # Calculate proportional sample size for each group
    group_sizes = grouped.size()
    total_rows = group_sizes.sum()
    downsample_frac = TARGET_MAX / total_rows  # Fraction to retain per group
    print("downsample fraction :::::", downsample_frac)
    final_sampled_data = grouped.apply(lambda x: x.sample(frac=min(1, downsample_frac), random_state=42)).reset_index(drop=True)
    print()
    print(f"After Downsampling, Final sampled dataset contains {len(final_sampled_data)} rows.", "\n")
```

2. Data Cleaning

2.1. Fixing Columns

2.1.1. Fix the index

To prepare the dataset for analysis, several **data-cleaning operations** are performed, including **fixing the index, removing unnecessary columns, handling missing values,** and **correcting incorrect data**.

- **Fixing the Index:** The dataset's index is reset using **reset_index(drop=True)** to ensure a continuous index after cleaning operations.
- **Removing Unnecessary Columns:** The **store_and_fwd_flag** column is dropped as it is not required for analysis.

2.1.2. Combine the two airport_fee columns

The dataset contains two different airport fee columns: **Airport_fee** and **airport_fee**. These are merged into a single column, **combined_airport_fee**, to ensure consistency in the dataset. After merging, both **Airport_fee** and **airport_fee** are dropped.

Identifying Columns with Negative Values

After cleaning, an additional check is performed to identify columns that still contain **negative values**, including **extra, mta_tax, improvement_surcharge, total_amount, congestion_surcharge, combined_airport_fee,** and **trip_duration**.

Negative Fare Amounts

1. An initial check is performed to identify rows where fare_amount is negative.
2. The analysis confirms that no negative values exist in the fare_amount column, so no further action is needed.

Negative total amount

1. Disputed Transactions: A total of 17 trips had negative total amounts, including 1 disputed trip (payment_type = 4)and 16 non-disputed trips.
2. Since this represents a small fraction of the dataset, all records where total_amount < 0 are removed.
3. After removing entries where total_amount < 0, all entries with extra, mta_tax, improvement_surcharge, congestion_surcharge, and combined_airport_fee < 0 are also removed. Therefore, no further actions are needed for these fields.

Negative Trip Duration

1. The trip_duration column is created to calculate the total travel time in minutes.
2. A negative trip duration indicates an issue with timestamps (e.g., incorrect ordering of pickup and drop-off times).
3. These invalid records are removed to maintain data integrity.

2.2. Handling Missing Values

2.2.1. Find the proportion of missing values in each column

Proportion of missing values in each column:

```
VendorID               0.000000
tpep_pickup_datetime   0.000000
tpep_dropoff_datetime  0.000000
passenger_count        3.174396
trip_distance          0.000000
RatecodeID             3.174396
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    3.174396
combined_airport_fee    3.174396
trip_duration           0.000000
pickup_hour             0.000000
pickup_day_of_week      0.000000
dtype: float64
```

According to the statistics, missing values are found in **passenger_count, RatecodeID, congestion_surcharge,** and **combined_airport_fee**, all at approximately **3.17%**. A pattern is observed where these missing values are associated with **payment_type = 0**, which is invalid.

2.2.2. Handling missing values in passenger_count

Removed rows where **passenger_count** was **NaN**, as they also had missing **RatecodeID, congestion_surcharge,** and **combined_airport_fee**.

Note: Found **1.58%** of rows with **passenger_count = 0**, which were removed since a trip should always have at least one passenger.

2.2.3. Handle missing values in RatecodeID

While removing entries where **passenger_count** is **null**, the missing values in **RatecodeID** are also removed. Additionally, an analysis of the **RatecodeID** column reveals the following distribution:

RatecodeID Distribution (Frequency Count)

```
RatecodeID
1.0    268949
2.0     11234
99.0     1634
5.0      1577
3.0       996
4.0       575
```

It is observed that **0.58%** of the rows have **RatecodeID = 99**, which is a small proportion of the entire dataset and an invalid category. To ensure data accuracy, these entries **are removed**.

2.2.4. Impute NaN in congestion_surcharge

While removing entries where **passenger_count** was null, the missing values in **congestion_surcharge** are also removed.

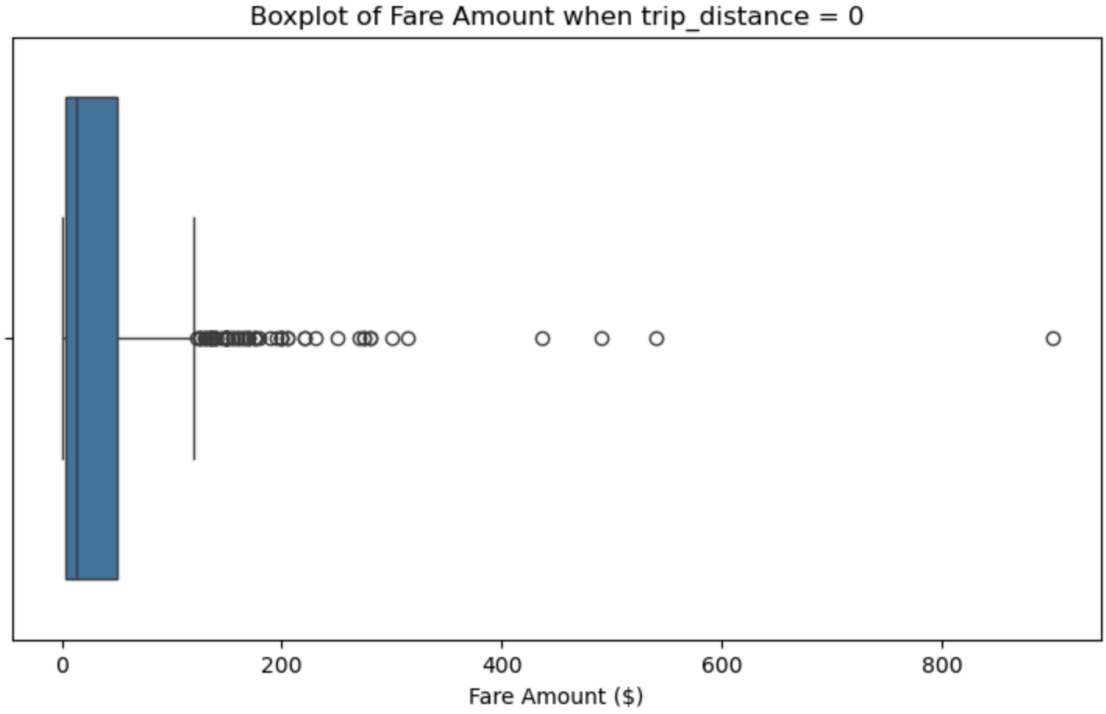
2.3. Handling Outliers and Standardising Values

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

1. **Remove passenger_count > 6:** Based on domain knowledge, records where passenger_count exceeded 6 are filtered out.

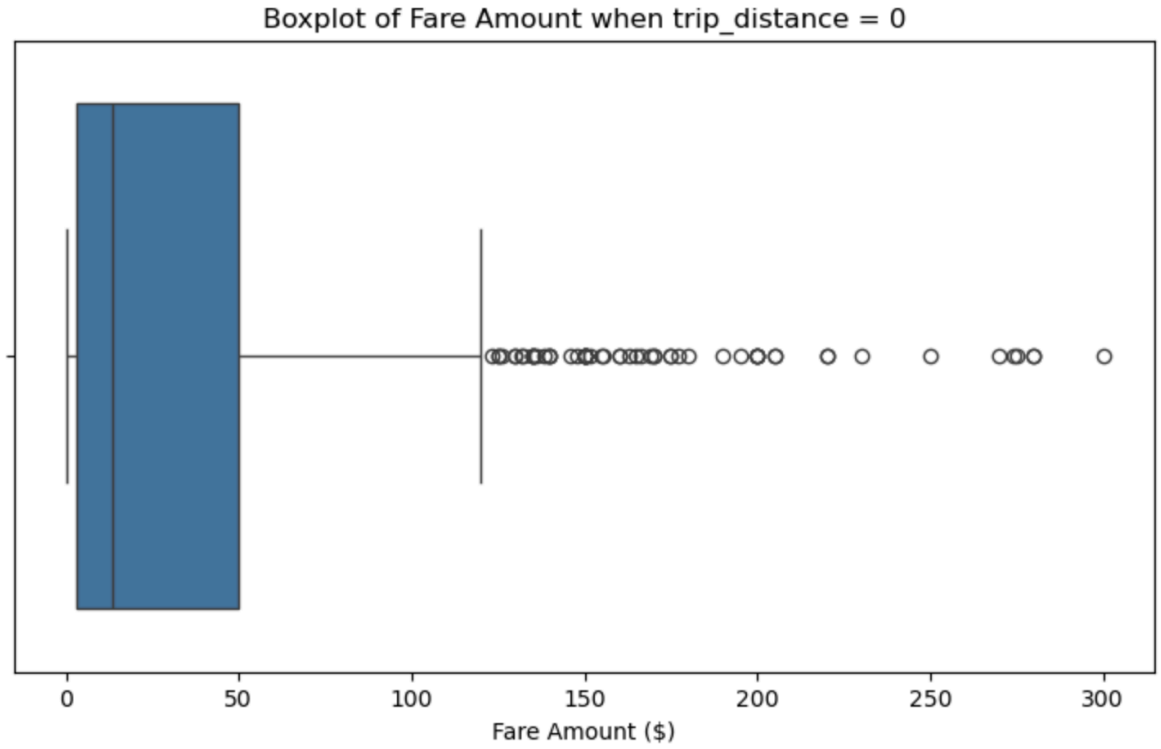
2. **Handling outliers where the trip_distance is nearly 0 and the fare_amount exceeds 300**
    - Identified outliers where **trip_distance = 0** and **fare_amount > 300**, representing erroneous or extreme cases.
    - Plotted a boxplot to visualize the distribution of **fare_amount** for these outliers.

Box Plot of Fare Amount for Trips with Zero Distance (Before Handling Outliers)



- Filtered out these extreme cases by removing rows where **trip_distance = 0** and **fare_amount > 300** to clean the data..
- After filtering out the outliers, the boxplot is plotted again to verify that the extreme values have been removed, ensuring data consistency.

. Box Plot of Fare Amount for Trips with Zero Distance (Before Handling Outliers)



3. **Handling outliers where trip_distance and fare_amount are 0 but the pickup and dropoff zones are different (both distance and fare should not be zero for different zones)**

Identified and removed **14 entries** where both **trip_distance** and **fare_amount** were **0**, but the **pickup and dropoff locations (PULocationID and DOLocationID)** were different.

This situation is inconsistent, as logically, a trip with **no distance or fare** should either have **identical pickup and dropoff locations** or should not be recorded at all.

4. **Handling outliers where `trip_distance` is more than 250 miles.**

Performed an initial analysis of the trip_distance column to understand its distribution and identify outliers with trip_distance > 250 miles.

Before removing entries with trip_distance > 250:

Summary Statistics for trip_distance:

Count: 283,310 records

Mean: 3.51 miles

Standard Deviation (std): 26.10 miles

Min: 0.00 miles

25th Percentile (25%): 1.06 miles

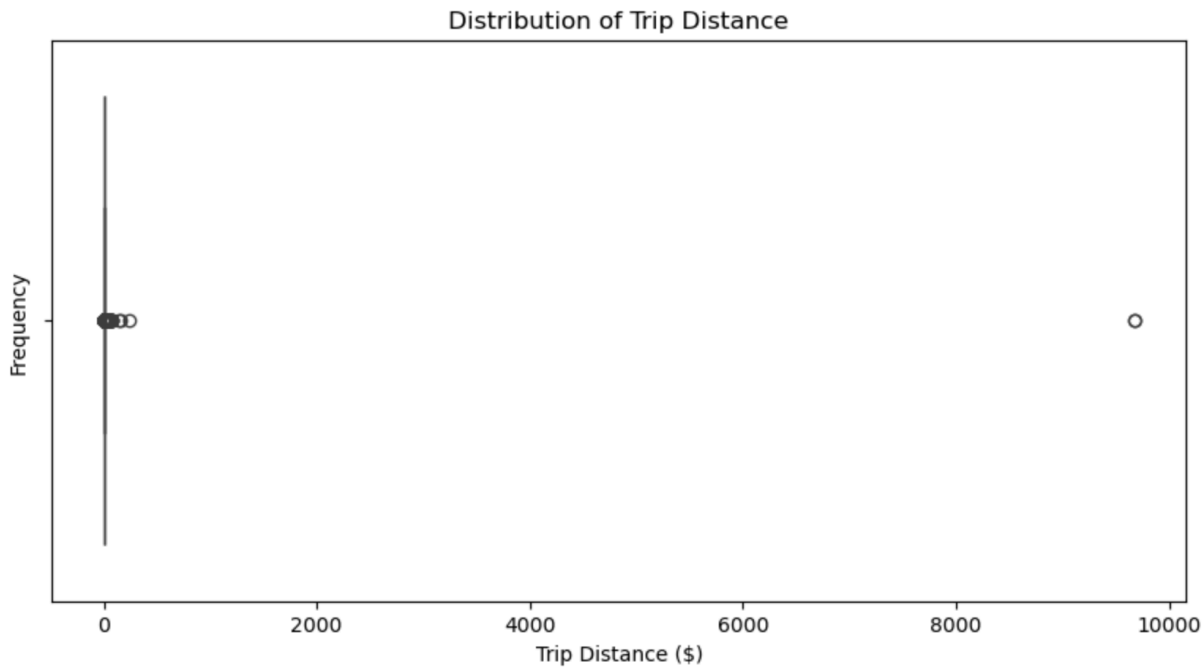
Median (50%): 1.77 miles

75th Percentile (75%): 3.34 miles

Max: 9676.91 miles

Median: 1.77 miles

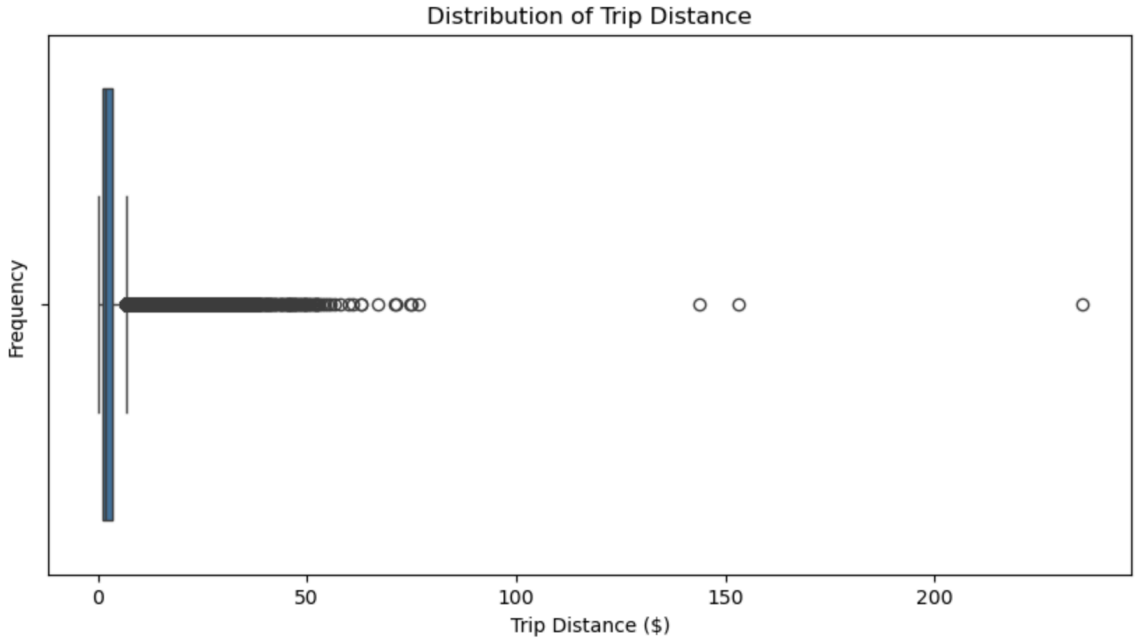
Distribution of Trip Distance (Before Removing Outliers > 250 miles):



Based on the summary statistics and visual analysis, the **trip_distance** column contains a few extreme outliers above **250 miles**, with one value as high as **9,676.91 miles**. The majority of the data, with a **median of 1.77 miles** and a **75th percentile of 3.34 miles**, indicates that these high values are not representative of typical trip distances.

After filtering out rows where **trip_distance > 250 miles**, the data is ensured to be **consistent and reliable** for further analysis or modeling.

Distribution of Trip Distance (After Removing Outliers > 250 miles):



5. **Handling outliers where payment_type is 0 (there is no payment_type 0 defined in the data dictionary)**

Upon checking the entries where **payment_type** is **0**, it is found that no such entries exist. Therefore, no further action is required to handle this case.

6. **Outlier Handling for Trip Distance, Fare per Mile, and Fare Amount**

To improve data quality and mitigate the influence of extreme values, **outliers are handled** for **trip_distance, fare_per_mile, and fare_amount** using a **99th percentile capping** approach. The process includes **recalculating derived features** to ensure consistency after handling outliers.

- **Trip Distance Outlier Handling**

The following statistics provide insights into the distribution of **trip distance** before handling outliers:

Statistic	Value	Description
Total Entries (Count)	283,308	Total number of recorded trips.
Mean	3.44 miles	The average trip distance, indicating most trips are relatively short.
Standard Deviation (Std)	4.56 miles	A high variation suggests some trips deviate significantly from the average distance.
Minimum (Min)	0.00 miles	The shortest recorded trip, possibly due to data entry errors or canceled trips.
25th Percentile (Q1)	1.06 miles	25% of trips have a distance shorter than this value.

Median (50th Percentile, Q2)	1.77 miles	The midpoint distance; half of the trips are shorter than this value.
75th Percentile (Q3)	3.34 miles	75% of trips have a distance shorter than this value.
Maximum (Before Capping)	235.36 miles	The longest recorded trip, an extreme outlier.
Maximum (After Capping)	20.27 miles	The upper limit after outlier capping to ensure data consistency.

After applying **99th percentile capping**, extreme values above **20.27 miles** are capped to prevent skewing the analysis.

- **Fare per Mile Outlier Handling**

  Recalculated **fare_per_mile** using **fare_amount / trip_distance**.

  Removed rows with **null (NaN)** values in the **fare_per_mile** column.

  Replaced **infinite values (np.inf, -np.inf)** and **NaN values** with **0** to ensure data consistency.

  These steps ensure that the **fare_per_mile** column is accurate and clean, making it suitable for further analysis or modeling.

  After reviewing the **fare_per_mile** statistics, a decision is made to **cap outliers beyond the 99th percentile (22.35)**. This adjustment reduces the influence of extreme values, making the data more representative of typical **fare per mile** values. As a result, unusually high values no longer distort the analysis, ensuring a more reliable dataset.

  Summary Statistics for Fare Per Mile

Statistic	Value	Description
Count	283,278	Total number of fare-per-mile entries.
Mean	11.45	The average fare per mile, influenced by extreme values before capping.
Standard Deviation	136.90	High variation indicating significant fluctuations.
Minimum	0.00	Possibly due to missing or incorrect fare data.
25th Percentile (Q1)	5.62	25% of trips had a fare per mile below this value.
Median (Q2, 50%)	7.14	Half of the trips had a fare per mile below this value.
75th Percentile (Q3)	9.03	75% of trips had a fare per mile below this value.
Maximum (Before Capping)	12,515.00	Extreme outlier, possibly due to short trips with high fares.

Maximum (After Capping)	22.35	Capped at the 99th percentile to reduce distortion.

After handling outliers and cleaning the **fare_per_mile** data, the next logical step is to **recalculate the fare_amount**using the updated **fare_per_mile** and **trip_distance** values. This ensures that **fare_amount** remains consistent with the corrected data and accurately reflects the cleaned **fare_per_mile** values.

Formula to calculate fare_amount :

```
new_df["fare_amount"] = new_df["fare_per_mile"] * new_df["trip_distance"]
```

After recalculating **fare_amount**, its statistics are analyzed, revealing the presence of outliers.

Statistic	Value	Description
Count	283,278	Total number of fare amount entries.
Mean	19.27	The average fare amount across all trips in the dataset.
Standard Deviation	17.80	A high standard deviation indicates significant variability in fare amounts.
Minimum	0.00	The minimum fare amount is 0, possibly due to missing or incorrect fare data.
25th Percentile (Q1)	9.30	25% of the trips had a fare amount below this value.
Median (50th Percentile)	13.50	The median fare amount is 13.50, indicating the middle value of all fare amounts.
75th Percentile (Q3)	21.20	75% of the trips had a fare amount below this value.
Maximum (Before Capping)	453.09	The highest fare amount before outlier handling. Extreme outliers were capped to reduce their impact.
Maximum (After Capping)	73.70	The maximum fare amount after capping at the 99th percentile, ensuring that extreme outliers do not distort the data.

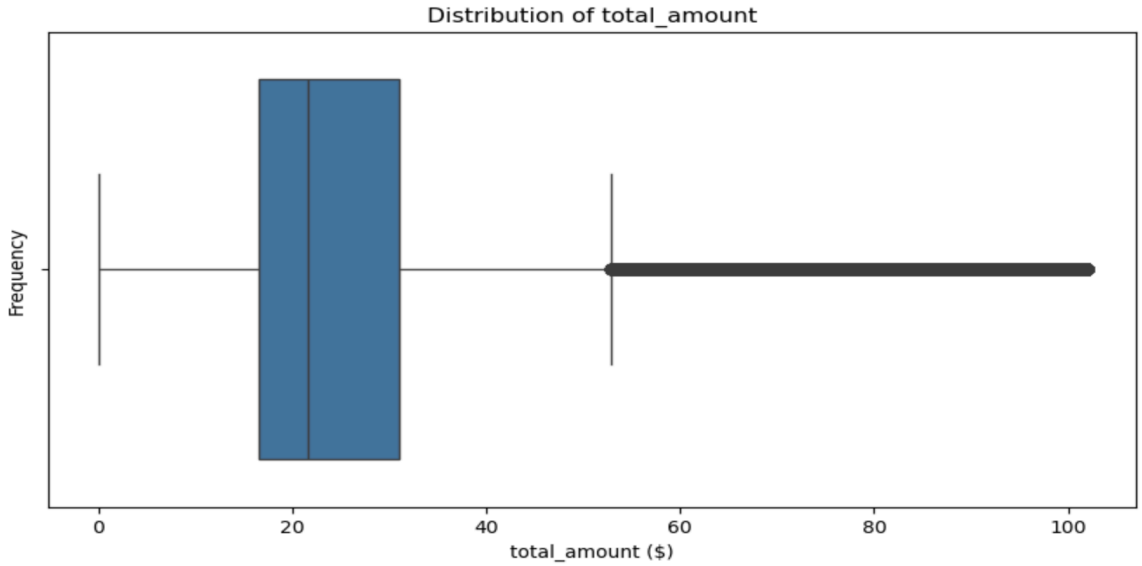
To ensure data consistency and minimize the impact of extreme values, **capping** is applied to the **fare_amount** column, with outliers above **$73.70** (99th percentile) capped at this threshold.

Similarly, outliers in **tolls_amount, tip_amount, and extra** are handled by applying capping at the **99th percentile** to maintain data consistency.

After addressing outliers in individual components (**fare_amount, tip_amount, tolls_amount, extra, etc.**), the **total_amount** column is recalculated as the sum of all relevant charges to prevent data inconsistencies.

The **total_amount** statistics are then analyzed, revealing the presence of outliers, which are handled using the **99th percentile capping method**. This ensures that extreme values are adjusted while preserving the overall data distribution.

Blox plot for total_amount (After handling Outliers)



Standardization

To enhance data clarity and interpretability, categorical columns with numerical representations are standardized by mapping them to their respective descriptive labels.

- **payment_type Standardization:**
  The **payment_type** column originally contained **numerical values (1-6)**, which are replaced with meaningful labels:

1 → Credit Card 2 → Cash 3 → No Charge 4 → Dispute 5 → Unknown 6 → Voided Trip

- **RatecodeID Standardization:**
  The **RatecodeID** column, which represents different rate codes, is converted into descriptive labels:

  1 → Standard Rate

  2 → JFK

  3 → Newark

  4 → Nassau or Westchester

  5 → Negotiated Fare

  6 → Group Ride

3. Exploratory Data Analysis

3.1. General EDA: Finding Patterns and Trends

3.1.1. Classify variables into categorical and numerical

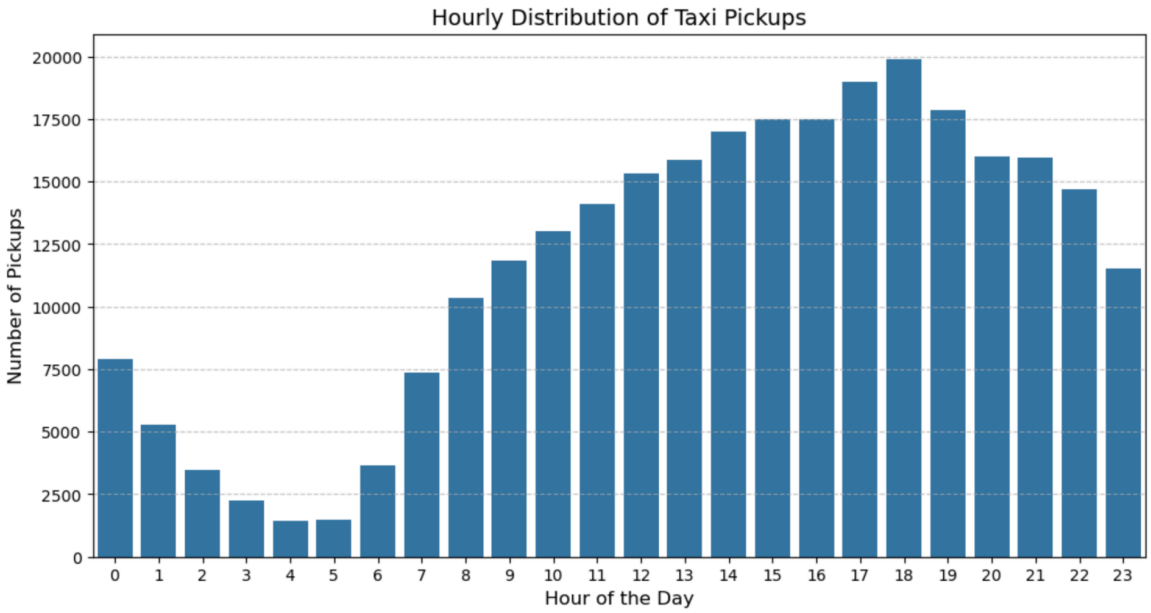
Categorization of Variables

Category	Variable	Type	Description
Categorical Variables	VendorID	Categorical	Identifier for the taxi vendor
	RatecodeID	Categorical	Rate code applied to the trip
	PULocationID	Categorical	Pickup location ID
	DOLocationID	Categorical	Drop-off location ID
	payment_type	Categorical	Type of payment used (Credit Card, Cash, etc.)
Numerical Variables	tpep_pickup_datetime	Numerical	Timestamp when the trip started
	tpep_dropoff_datetime	Numerical	Timestamp when the trip ended
	passenger_count	Numerical	Number of passengers in the trip

	trip_distance	Numerical	Distance of the trip in miles
	pickup_hour	Numerical	Hour of pickup time
	trip_duration	Numerical	Duration of the trip in minutes
Monetary Parameters (Numerical)	fare_amount	Numerical	Cost of the trip fare
	extra	Numerical	Additional charges (e.g., peak surcharge)
	mta_tax	Numerical	MTA tax applied to the fare
	tip_amount	Numerical	Tip given by the passenger
	tolls_amount	Numerical	Toll charges for the trip
	improvement_surcharge	Numerical	Improvement surcharge applied to the fare
	total_amount	Numerical	Total amount charged for the trip
	congestion_surcharge	Numerical	Congestion surcharge applied to the trip
	airport_fee	Numerical	Airport-related charges

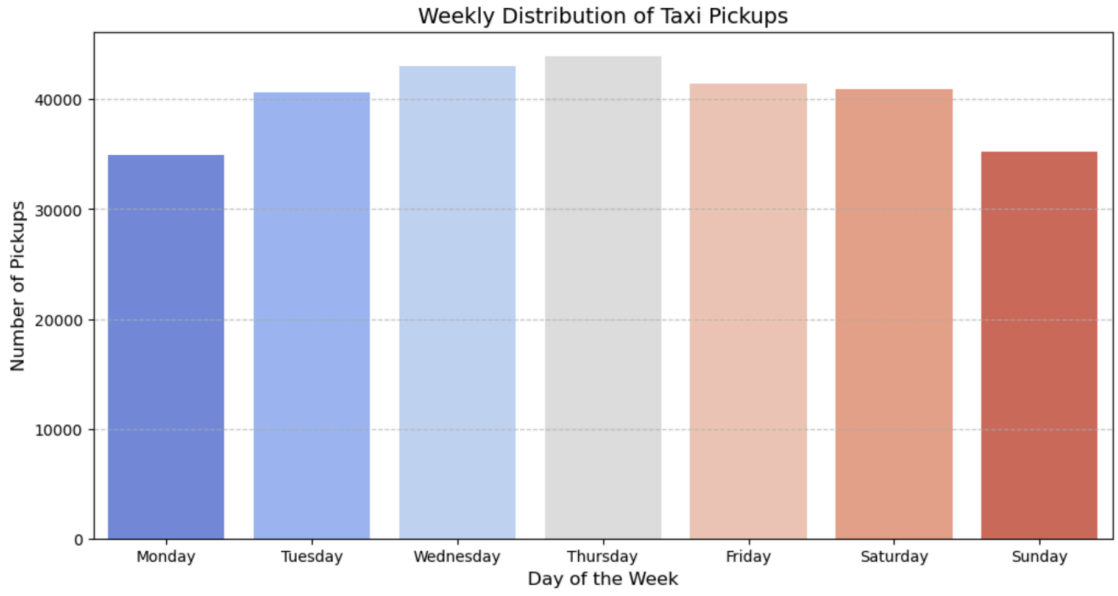
3.1.2. Analyse the distribution of taxi pickups by hours, days of the week, and months

Hourly Distribution of Taxi Pickups



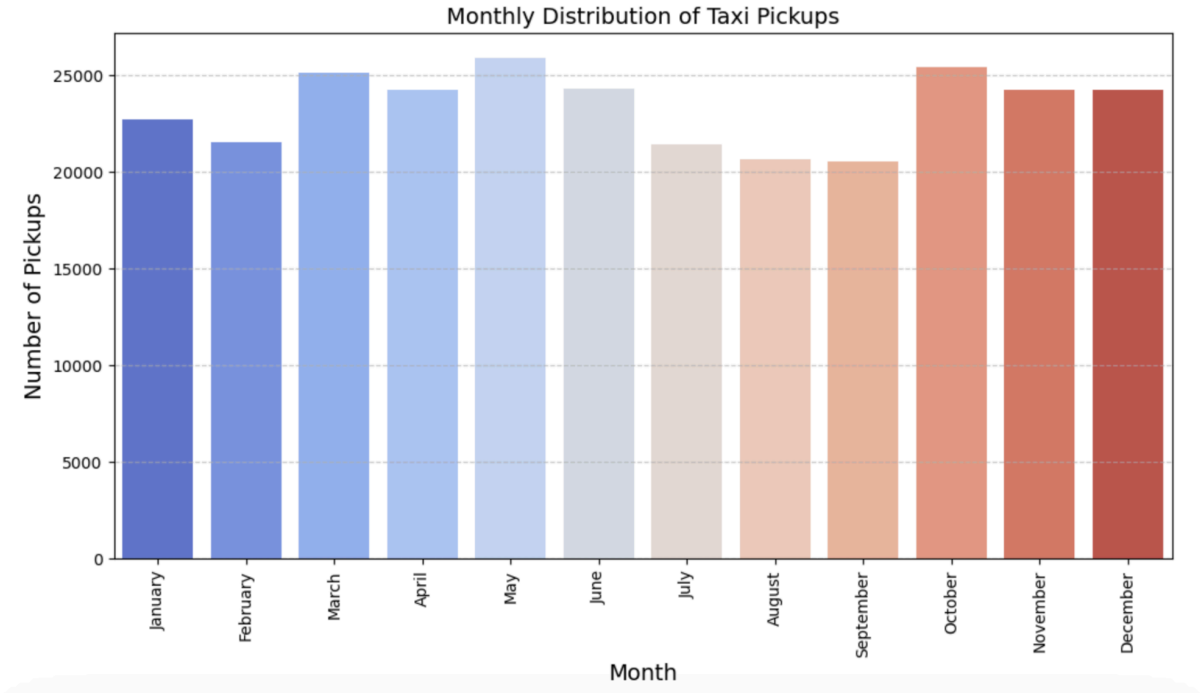
1. The highest number of taxi pickups occur between 3 PM and 8 PM, with a peak around 6 PM to 7 PM.
2. The lowest number of pickups happen between 3 AM and 6 AM, likely due to reduced demand during early morning hours.
3. The demand starts increasing after 6 AM, possibly due to the morning rush hour and continues rising through the day.

Weekly Distribution of Taxi Pickups



1. The busiest days appear to be Wednesday, Thursday, and Friday, with a peak around Thursday.
2. A slight decline in pickups is observed on Saturday and Sunday, though the difference is not drastic.
3. Mondays show fewer pickups, likely due to fewer outings after the weekend.

Monthly Distribution of Taxi Pickups



1. The highest taxi pickups occur in May and October, indicating seasonal trends.
2. February has the lowest pickup numbers, likely due to fewer days in the month.
3. Most months show relatively stable demand, except for small fluctuations.

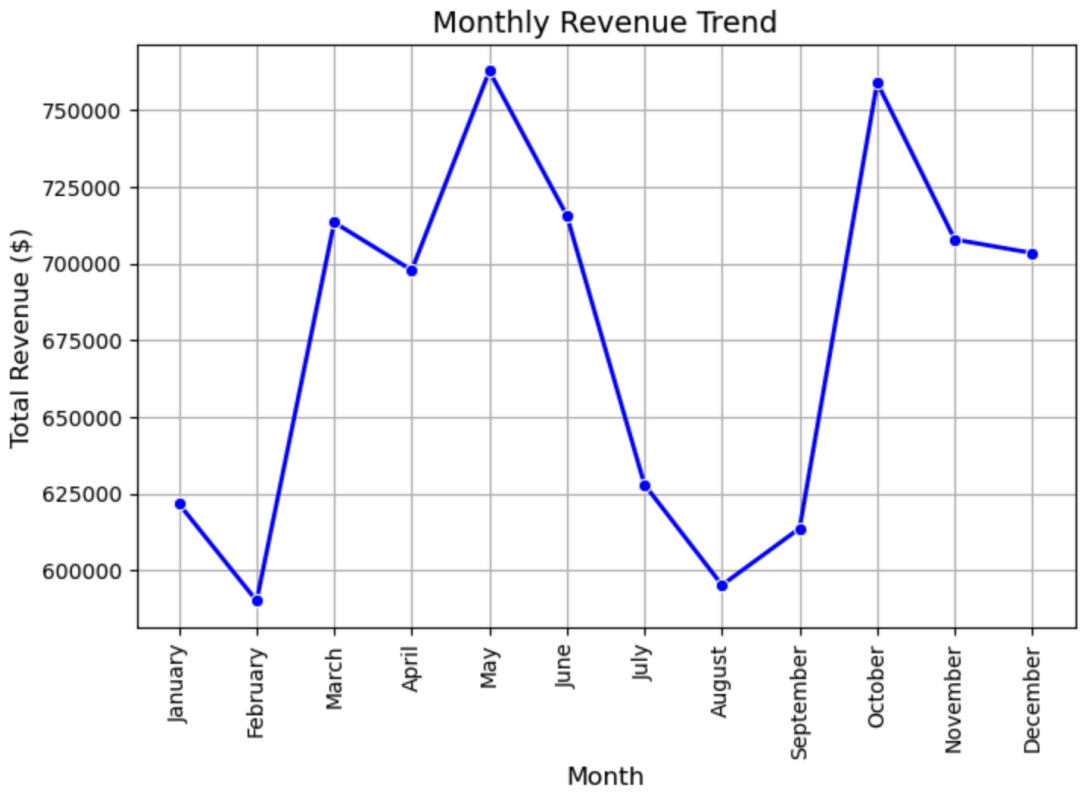
3.1.3. Filter out the zero/negative values in fares, distance and tips

Fare Amount: No missing or negative values, and only 0.01% of entries have zero fares, likely due to canceled trips or promotions.
Tip Amount: No missing or negative values, but 21.45% of trips recorded zero tips, possibly due to non-tipping behavior or unrecorded cash tips.
Total Amount: Clean data with no missing, negative, or zero values.
Trip Distance: No missing, negative, or zero values, ensuring accurate distance tracking.

However, creating a filtered copy of the DataFrame, excluding rows with zero values, can be useful for cleaner analysis. Whether to remove them completely depends on the context.

Valid zero values: A **tip_amount**, **fare_amount**, and **total_amount** of 0 may be valid, while a **trip_distance** of 0 might be valid if the trip occurred within the same zone (e.g., a short shuttle ride).

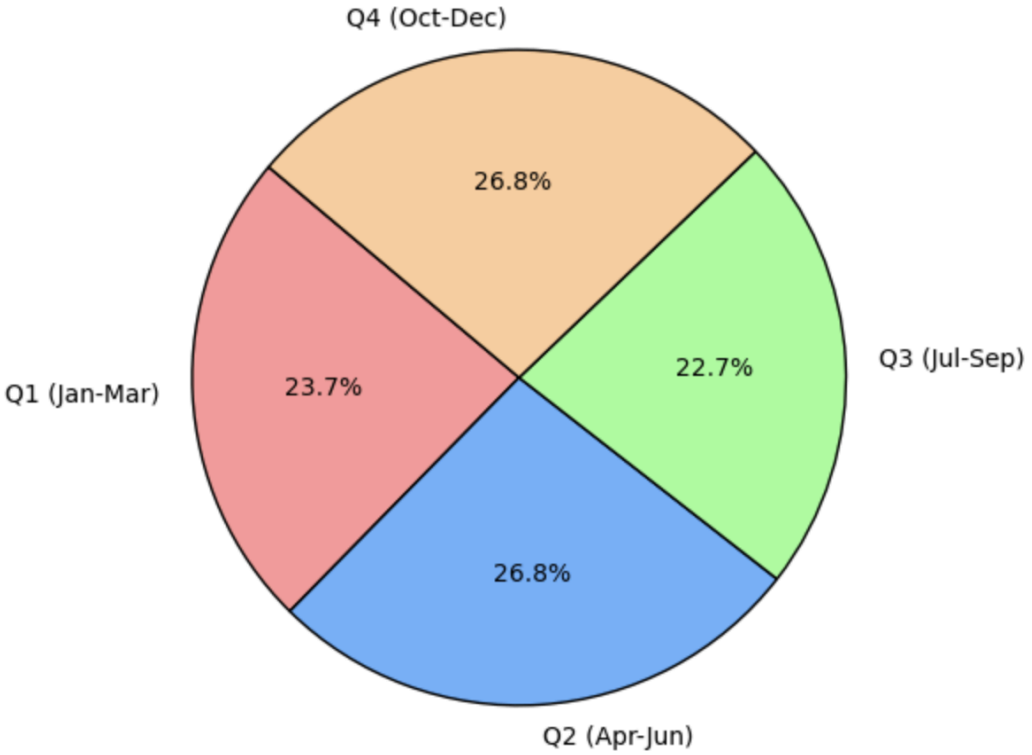
3.1.4. Analyse the monthly revenue trends



Overall Revenue Trends:

1. The average total revenue across all months is $675,633.89.
2. Revenue fluctuates significantly throughout the year, indicating seasonal or operational variations.
3. The highest revenue month is May ($762,334.60), followed closely by October ($758,968.47).
4. The lowest revenue month is February ($590,037.40).

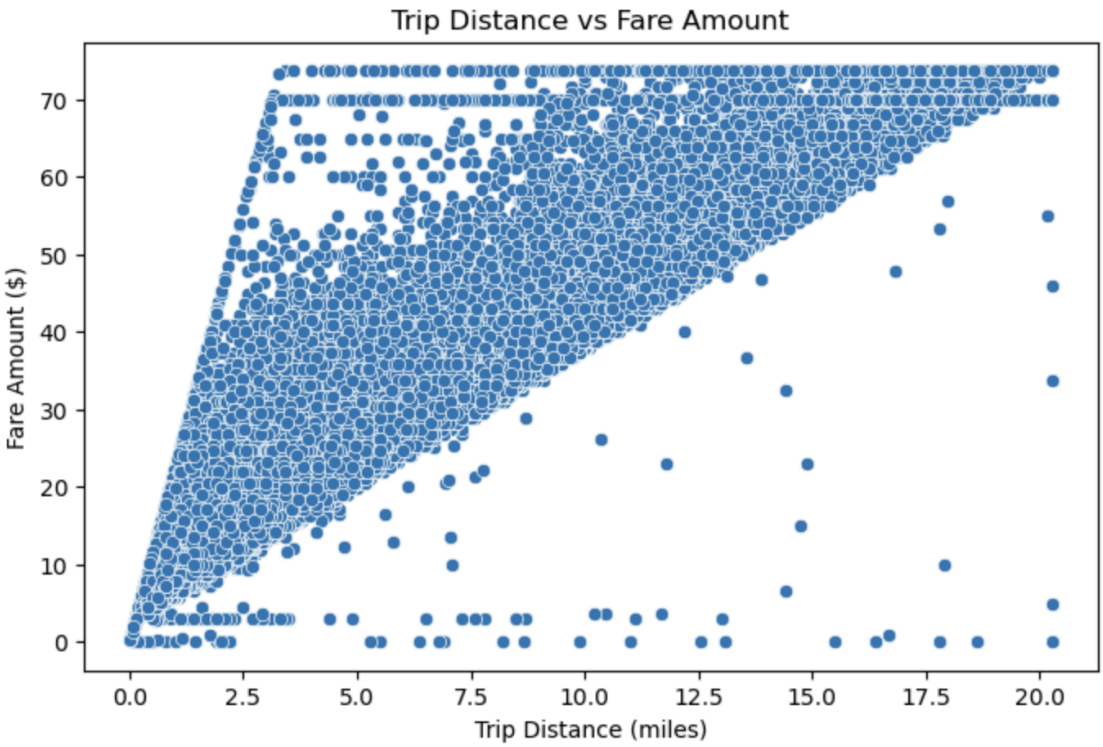
3.1.5. Find the proportion of each quarter's revenue in the yearly revenue



Quarterly Revenue Analysis & Insights

1. **Top Performing Quarters:** Q2 (Apr-Jun) and Q4 (Oct-Dec) lead with 26.8% of total revenue each. These quarters likely benefit from seasonal trends, promotions, or peak demand periods.
2. **Weakest Quarter:** Q3 (Jul-Sep) had the lowest share (22.7%), indicating a mid-year slowdown. A decline in consumer spending or business activity might contribute to this drop.
3. **Q1 Performance:** At 23.7%, Q1 (Jan-Mar) performs better than Q3 but lags behind Q2 and Q4. Post-holiday effects may slow down early-year revenue, with a recovery seen towards the quarter's end.

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

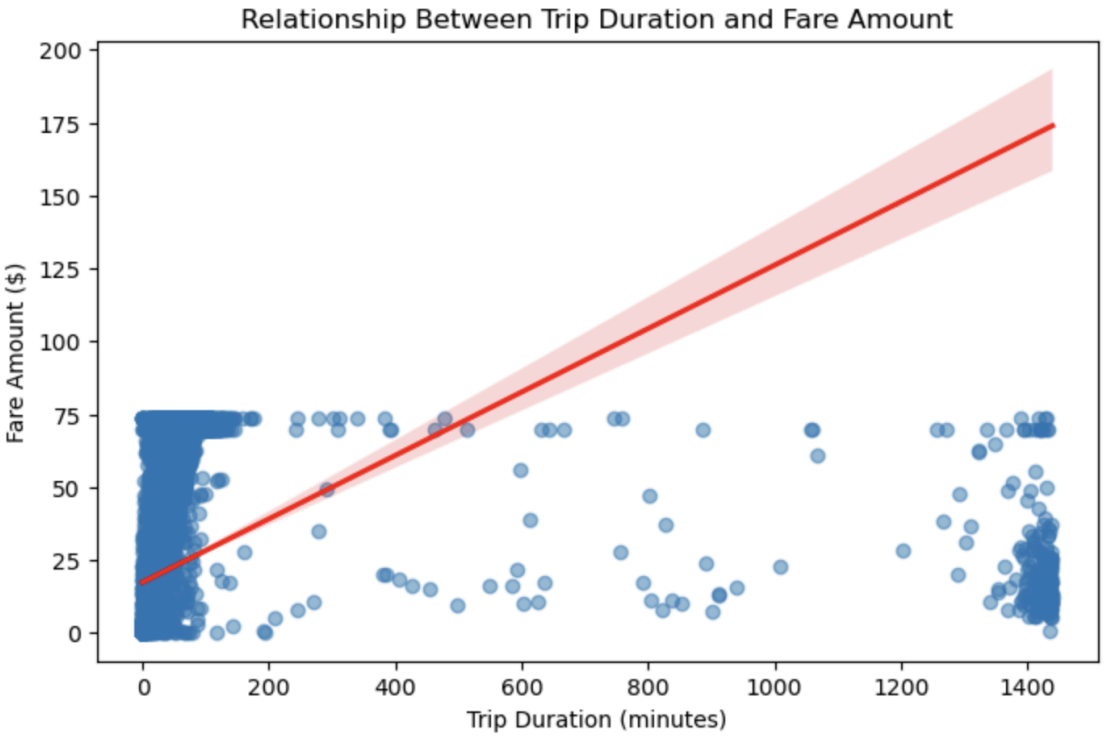


Analysis:

1. A strong positive correlation (0.97) exists between trip distance and fare amount, confirming that longer trips generally cost more. However, fare inconsistencies suggest surge pricing or fixed-rate fares.
2. The scatter plot forms a triangular pattern, showing proportional fare increases with some anomalies.
3. Outliers are present, so values are capped at the 99th percentile for future analysis.

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

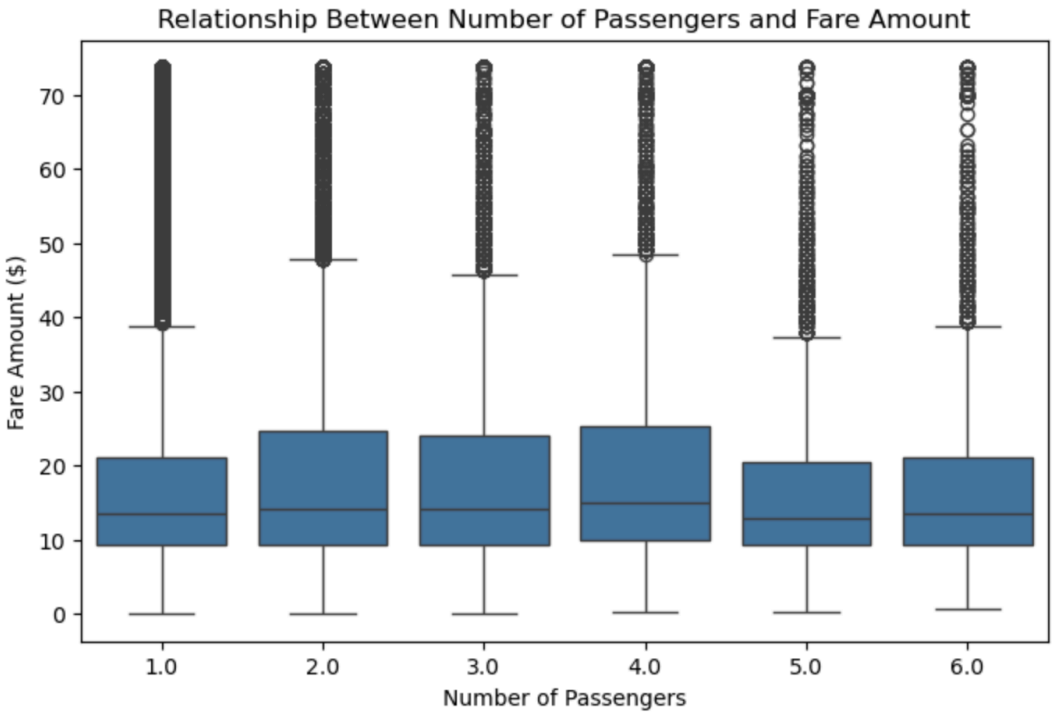
Relationship between fare and trip duration



Analysis

1. The correlation (0.28) between trip duration and fare amount is weak, indicating that fare is primarily influenced by distance rather than time, except in cases of congestion or waiting charges.
2. The scatter plot reveals significant variation in fares for similar durations, with some long trips having low fares, likely due to fixed pricing or discounts.

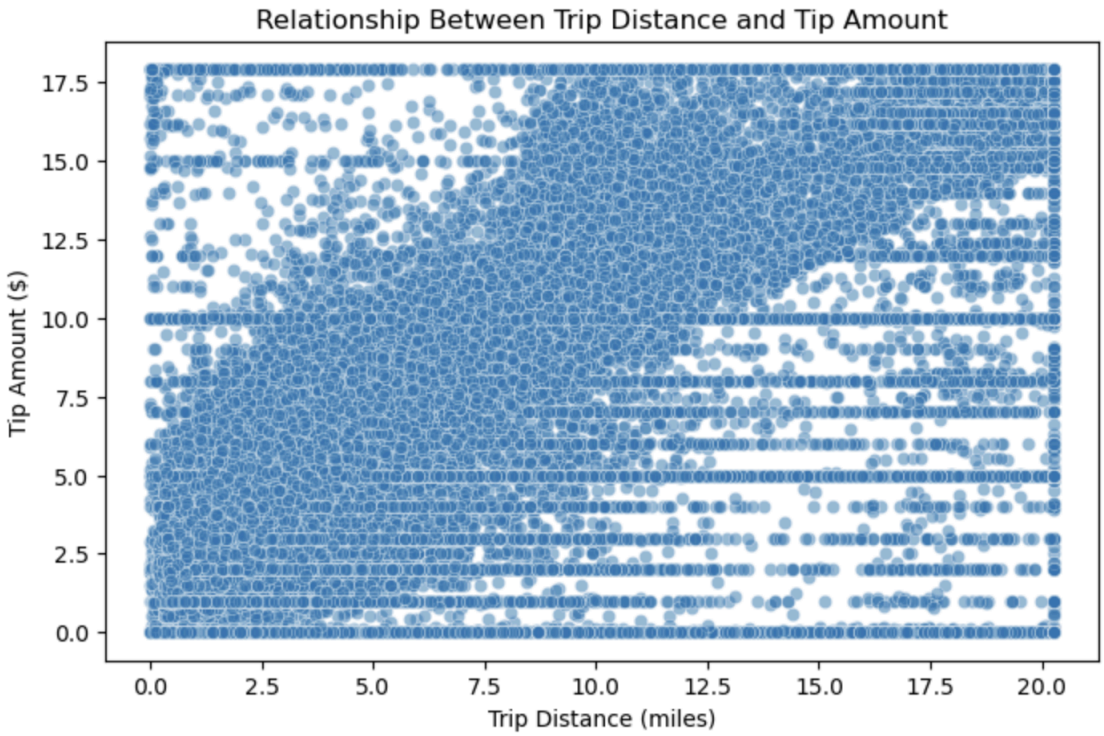
Relationship between fare and number of passengers



Analysis of Passenger Count vs. Fare Amount

1. Correlation: **0.05 (Very Weak Relationship)**
2. The number of passengers has little to no impact on the fare amount. This suggests that fare pricing is not significantly influenced by the number of passengers, likely due to a fixed fare structure based on distance and time rather than occupancy.
3. The box plots show similar median fares across different passenger counts, reinforcing the weak correlation.
4. Outliers are present, indicating some high fares, but they are capped at the 0.99 percentile for better distribution control while keeping them for future analysis.

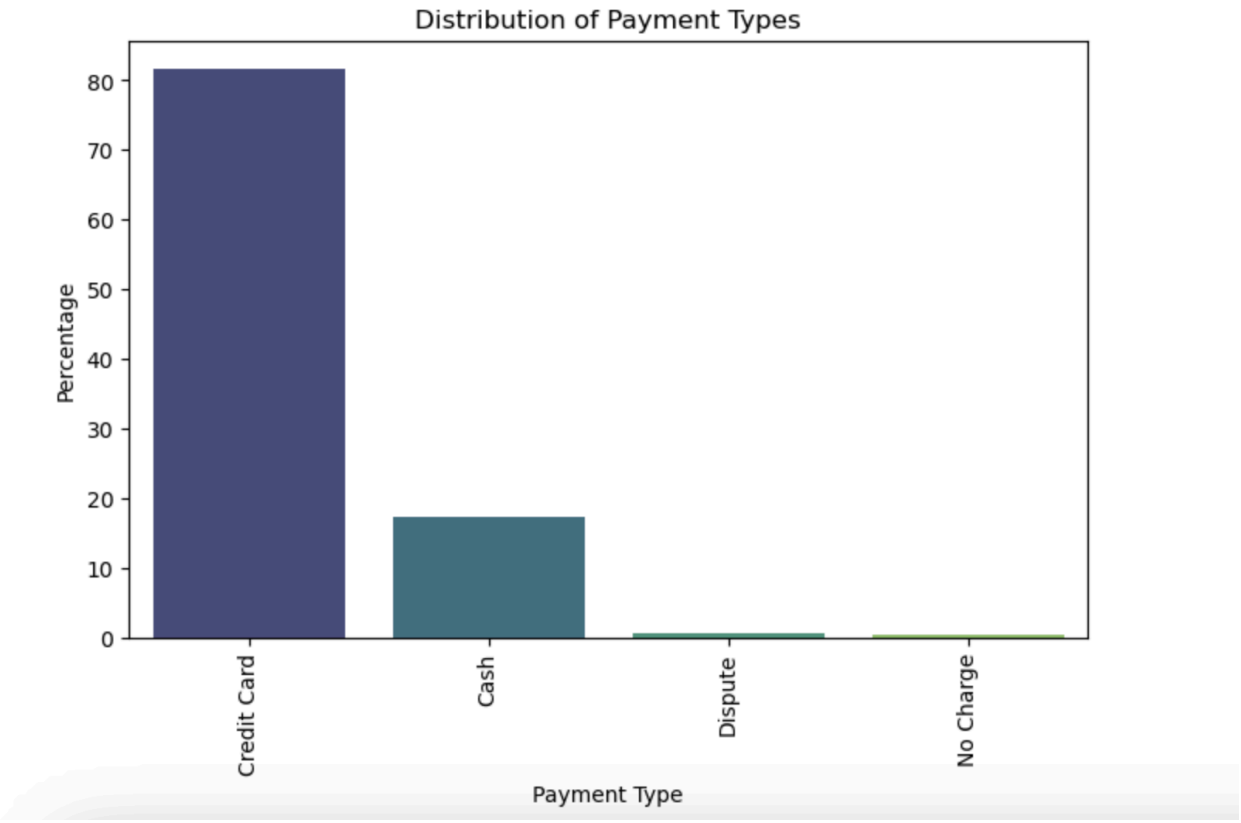
Relationship between tip amount and trip distance



Analysis of Trip Distance vs. Tip Amount

1. Correlation: 0.62 (Moderate to Strong Positive Relationship)
2. As trip distance increases, tip amounts generally increase, suggesting that passengers are more likely to tip higher for longer trips.
3. The scatter plot shows a clear upward trend, with denser points at lower tip amounts, indicating that many short trips still receive minimal or no tips.
4. Structured tipping behavior is visible (horizontal bands), likely due to rounding preferences or preset tipping options.
5. Outliers are present, indicating some high tips, but they are capped at the 0.99 percentile for better distribution control while keeping them for future analysis.

3.1.8. Analyse the distribution of different payment types



Credit card payments dominate (80%+), while cash transactions account for 15-20%. Other payment types, such as disputes and no-charge rides, are minimal. The trend highlights a preference for digital payments, though cash remains relevant.

3.1.9. Load the taxi zones shapefile and display it

To load and display a shapefile using GeoPandas, the following code can be used.

```
import geopandas as gpd

# Read the shapefile using geopandas
shape_file_path = os.path.join("taxi_zones/", 'taxi_zones.shp')
zones = gpd.read_file(shape_file_path)
zones.head()
```

Result:

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	0.00486 6	Jamaica Bay	2	Queens	MULTIPOLYG ON (((1033269.244 172126.008, 103343...
2	3	0.084341	0.00031 4	Allerton/Pelh am Gardens	3	Bronx	POLYGON ((1026308.77 256767.698, 1026495.593 2...
3	4	0.043567	0.00011 2	Alphabet City	4	Manhatta n	POLYGON ((992073.467 203714.076, 992068.667 20...
4	5	0.092146	0.00049 8	Arden Heights	5	Staten Island	POLYGON ((935843.31 144283.336, 936046.565 144...

3.1.10. Merge the zone data with trips data

To merge zones and trip records using locationID and PULocationID, the following code can be used.

```
# Merge zones and trip records using locationID and PULocationID
merged_df = new_df.merge(zones, left_on="PULocationID", right_on="LocationID", how="left")
merged_df.head(3)
```

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

No of Trips for each location ID:

```
    PULocationID  total_trips
89           132        14739
171          237        13422
111          161        13103
170          236        12034
```

94	138	10044
112	162	9997
131	186	9936
165	230	9494
97	142	9425
120	170	8540

Based on the grouped data by location ID, where each location ID represents a unique taxi zone (likely the "pickup location" for trips).

Analysis:

Top 10 High-Demand Taxi Pickup Zones

1. JFK Airport (132) has the highest trip count (14,739), reinforcing its role as a key transportation hub.
2. Upper East Side South (237) and Midtown Center (161) also see high demand with 13,422 and 13,103 trips, reflecting strong activity in these commercial and residential areas.
3. LaGuardia Airport (138), Penn Station (186), and Times Square (230) further highlight the importance of transit hubs for taxi services.
4. The majority of high-traffic zones are in Manhattan, indicating heavy taxi reliance in business and tourist areas.

.

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

To add the number of trips for each zone to the zones DataFrame, the following code can be used.

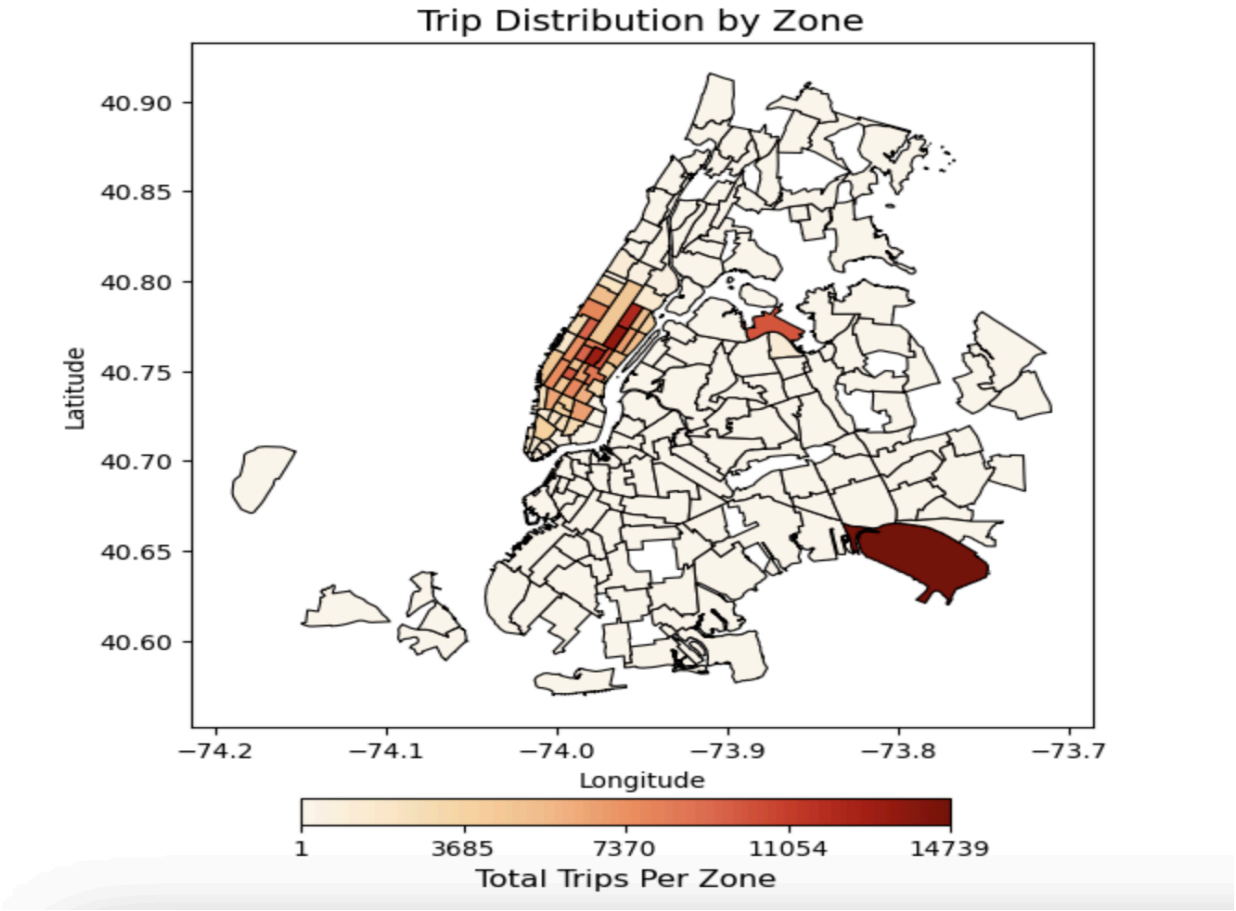
```
# Group data by location and calculate the number of trips
trip_counts = merged_df.groupby("PULocationID").size().reset_index(name="total_trips")
trip_counts = trip_counts.sort_values(by="total_trips", ascending=False)
print(trip_counts.head(10))
```

```
# Merge trip counts back to the zones GeoDataFrame
zones_with_trips = zones.merge(trip_counts, left_on="LocationID", right_on="PULocationID", how="left")
zones_with_trips.drop(columns=['PULocationID'], inplace=True)  # Remove extra column
zones_with_trips.head()
```

Result:

	OBJECTID	Shape_Leng	Shape_Area	zone	LocationID	borough	geometry	total_trips
0	1	0.116357	0.000782	Newark Airport	1	EWR	POLYGON ((933100.918 192536.086, 933091.011 19...	14.0
1	2	0.433470	0.004866	Jamaica Bay	2	Queens	MULTIPOLYGON (((1033269.244 172126.008, 103343...	NaN
2	3	0.084341	0.000314	Allerton/Pelham Gardens	3	Bronx	POLYGON ((1026308.77 256767.698, 1026495.593 2...	NaN
3	4	0.043567	0.000112	Alphabet City	4	Manhattan	POLYGON ((992073.467 203714.076, 992068.667 20...	284.0
4	5	0.092146	0.000498	Arden Heights	5	Staten Island	POLYGON ((935843.31 144283.336, 936046.565 144...	NaN

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



The trip distribution map highlights high-density zones, with the most trips originating from central Manhattan and key transit hubs. Areas like airports and commercial districts see significantly higher trip volumes, indicating high passenger demand in these regions. Lower-density zones correspond to residential or less-frequented areas.

3.1.14. Conclude with results

1. **Airports and Transit Hubs Drive Taxi Demand:** JFK Airport (14,739 trips) and LaGuardia Airport (10,044 trips) are among the busiest pickup locations, highlighting their role in taxi traffic.
2. **Manhattan Dominates High-Demand Zones:** Areas like Midtown Center, Upper East Side, Times Square, and Penn Station consistently record high trip volumes, indicating strong reliance on taxis in business and tourist-heavy districts.
3. **Strategic Insights:** Taxi services can optimize fleet distribution by prioritizing high-demand zones, especially around airports, transit stations, and business districts, ensuring better service efficiency and reduced wait times.

3.2. Detailed EDA: Insights and Strategies

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

To find the routes with the slowest speeds at different times of the day, we can analyze the provided data by sorting the routes based on their speed and identifying patterns of slow speeds across various time slots (**pickup hours**).

Result:

	PickupZone	DropoffZone	pickup_hour	speed
1192	Bloomingdale	Bloomingdale	9	0.000837
42999	Queensbridge/Ravenswood	Queensbridge/Ravenswood	12	0.030431
31778	Lower East Side	Lower East Side	22	0.053151
8333	East Harlem North	Central Park	17	0.053818
8378	East Harlem North	East Harlem South	20	0.063123
3209	Chinatown	Battery Park City	20	0.077371
3275	Chinatown	Financial District North	23	0.081240
2242	Central Harlem North	Manhattanville	18	0.085076
46551	Times Sq/Theatre District	Kips Bay	3	0.085294
43088	Saint Michaels Cemetery/Woodside	Saint Michaels Cemetery/Woodside	5	0.092545

Analysis:

Slowest Routes:

1. The route from Bloomingdale to Bloomingdale (Pickup Hour: 9) has the slowest speed (0.000837), indicating extremely low taxi speeds in this area at this time.
2. Queensbridge/Ravenswood to Queensbridge/Ravenswood (Pickup Hour: 12) also has a slow speed of 0.030431, suggesting potential congestion or low demand.

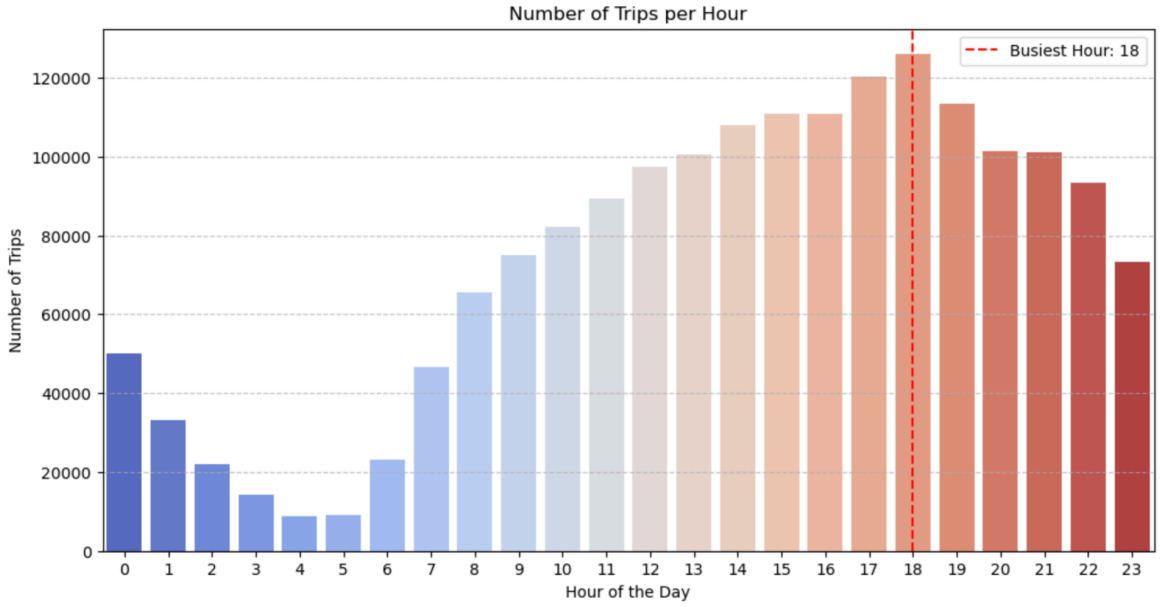
Time-of-Day Patterns:

1. From the dataset, early morning hours (e.g., Pickup Hour: 3 for Times Square) seem to have relatively slower speeds (0.085294), possibly due to early morning traffic or other factors like traffic signal timing, fewer taxis operating, or lower demand.
2. Mid-day hours (12 PM) and evening hours (20-23 PM) also show slow speeds, likely due to heavier commute traffic or rush-hour congestion.

Traffic Congestion Insights:

1. Routes in **high-density areas** like **Times Square** and **Chinatown** tend to have **slower speeds** (even during non-peak hours). These areas likely suffer from **congestion** due to both pedestrian and vehicle traffic.
2. **Areas like Central Harlem and East Harlem** also show slower speeds, suggesting traffic density, but this might also be linked to **taxis serving the areas during busy hours**.

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



Analysis.

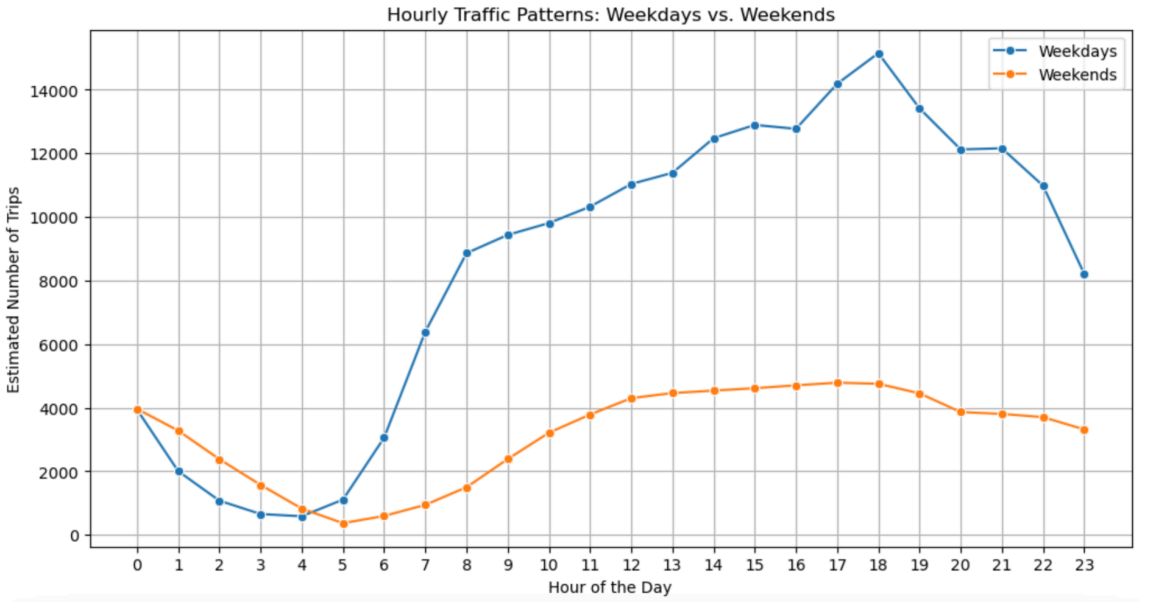
1. **Busiest Hour:** 6 PM (18:00) with 126,104 trips.
2. **Trend:** Trips increase from early morning, peak at 6 PM, then decline.

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

Sampled and Actual Number of Trips

	pickup_hour	sampled_num_trips	actual_num_trips
18	18	126104	2522080.0
17	17	120362	2407240.0
19	19	113308	2266160.0
15	15	110886	2217720.0
16	16	110861	2217220.0

3.2.4. Compare hourly traffic on weekdays and weekends

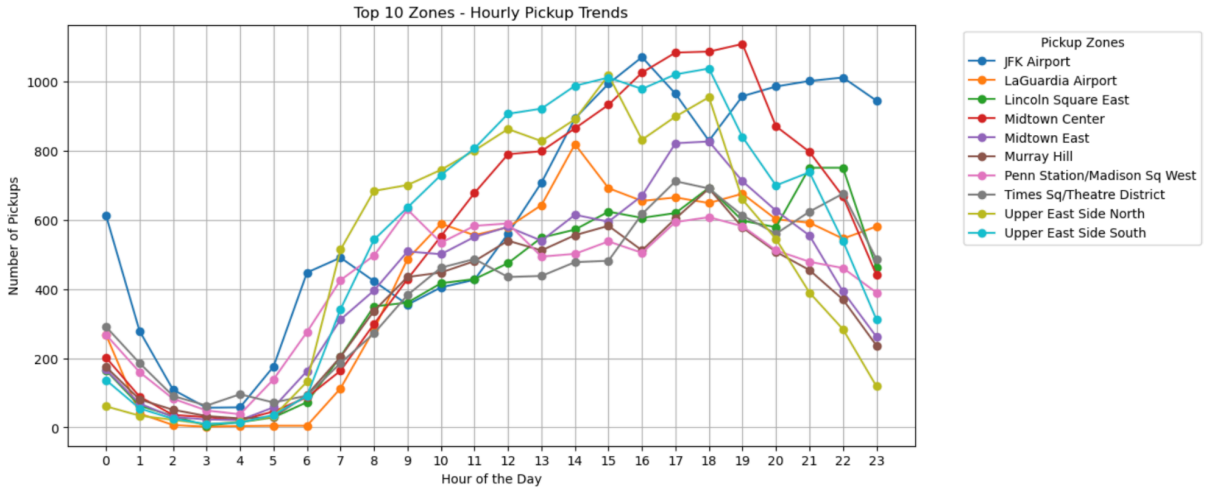


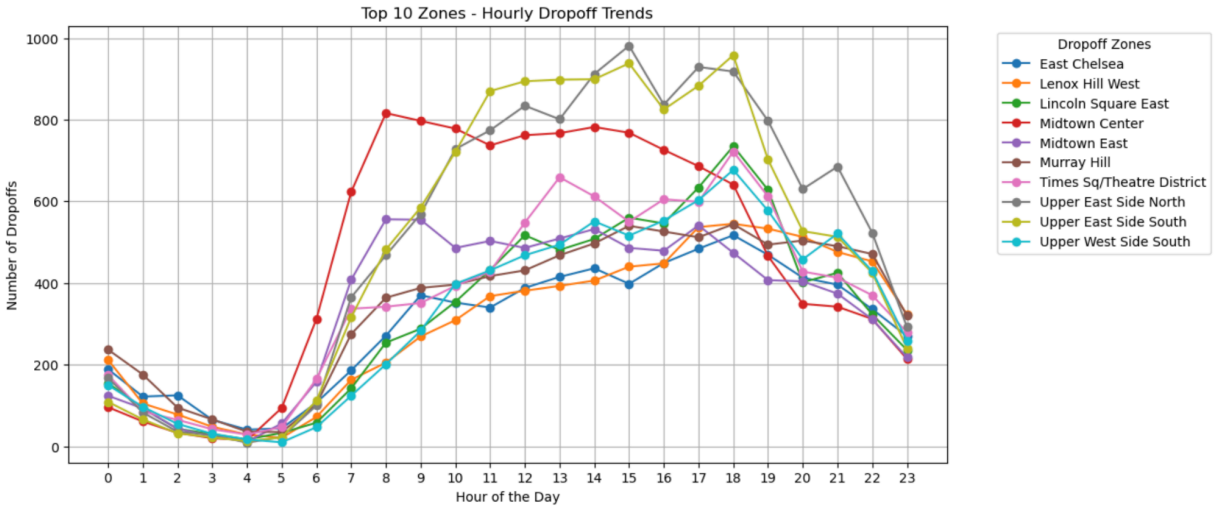
Time of Day	Weekdays Traffic Pattern	Weekends Traffic Pattern
12 AM - 5 AM	Low traffic, gradual decline after midnight	Slightly higher than weekdays, then gradually decreases
6 AM - 9 AM	Sharp increase, peaking at 8-9 AM due to morning commute	Low traffic, gradual rise starts around 8 AM
10 AM - 12 PM	Steady increase, small midday peak due to lunch breaks	Traffic gradually increases but remains stable
1 PM - 4 PM	Moderate, stable traffic, no major fluctuations	Traffic remains consistent, no sharp peaks
5 PM - 7 PM	Evening peak at 6-7 PM due to work commute	Gradual increase, but no sharp peaks
8 PM - 10 PM	Decline in traffic after 8 PM	Traffic stabilizes and declines gradually
11 PM - 12 AM	Sharp drop, significantly lower trips	Slightly higher than weekdays, indicating leisure/social activities

Key Differences:

1. Weekdays: Two major peaks (morning & evening rush hours).
2. Weekends: More even distribution, with gradual increases and decreases.
3. Morning traffic is lower on weekends, while late-night traffic is slightly higher.

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





Rank	Top 10 Pickup Zones	Top 10 Dropoff Zones
1	JFK Airport	East Chelsea
2	LaGuardia Airport	Lenox Hill West
3	Lincoln Square East	Lincoln Square East
4	Midtown Center	Midtown Center
5	Midtown East	Midtown East
6	Murray Hill	Murray Hill
7	Penn Station/Madison Sq West	Times Sq/Theatre District
8	Times Sq/Theatre District	Upper East Side North
9	Upper East Side North	Upper East Side South
10	Upper East Side South	Upper West Side South

Insights:

1. Midtown areas are the busiest, as they are central to offices, hotels, and tourist destinations.
2. Times Square/Theatre District has continuous traffic due to Broadway shows, events, and tourism.
3. Airports (JFK, LaGuardia) have high pickups but are not top drop-off zones—likely due to outbound flights.
4. Residential neighborhoods (Upper East Side, Upper West Side) see more drop-offs, indicating commuters returning home.

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

Top 10 Zones with Highest Pickup/Dropoff Ratio

Rank	Zone	Pickup/Dropoff Ratio
1	East Elmhurst	8.295302
2	JFK Airport	4.910210
3	LaGuardia Airport	2.868796
4	Saint Michaels Cemetery/Woodside	2.000000
5	Penn Station/Madison Sq West	1.601354
6	Greenwich Village South	1.402957
7	West Village	1.375725
8	Central Park	1.356671
9	Midtown East	1.213497
10	Garment District	1.211856

Top 10 Zones with Lowest Pickup/Dropoff Ratio

Rank	Zone	Pickup/Dropoff Ratio
1	Allerton/Pelham Gardens	0.000000
2	Astoria Park	0.000000
3	Bath Beach	0.000000
4	Bay Terrace/Fort Totten	0.000000
5	Bayside	0.000000
6	Belmont	0.000000
7	Bensonhurst East	0.000000
8	Bloomfield/Emerson Hill	0.000000
9	Breezy Point/Fort Tilden/Riis Beach	0.000000
10	Brighton Beach	0.000000

Analysis of Pickup/Dropoff Ratios:

Highest Ratios:

1. East Elmhurst (8.29) has the highest pickup/dropoff ratio, likely due to its proximity to LaGuardia Airport.
2. JFK Airport (4.91) and LaGuardia Airport (2.87) also have high ratios, indicating more trips start from these locations than end there.
3. Other high-ratio zones, such as Penn Station and Midtown East, are major transit hubs with frequent taxi pickups.

Lowest Ratios:

1. Several zones, like Allerton/Pelham Gardens, Astoria Park, and Bath Beach (0.0), have no recorded pickups, indicating they are primarily dropoff locations.
2. These areas are likely residential neighborhoods or low-traffic zones where passengers mostly arrive rather than depart.

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

Top 10 Pickup Zones at Night (11PM-5AM)

Rank	PULocationID	Pickup_Count	PickupZone
1	79	2431	East Village
2	132	2237	JFK Airport
3	249	1984	West Village
4	48	1547	Clinton East
5	148	1513	Lower East Side
6	114	1375	Greenwich Village South
7	230	1286	Times Sq/Theatre District
8	186	1127	Penn Station/Madison Sq West
9	164	960	Midtown South
10	68	928	East Chelsea

Top 10 Dropoff Zones at Night (11PM-5AM)

Rank	DOLocationID	Dropoff_Count	DropoffZone
1	79	1232	East Village
2	48	1101	Clinton East
3	170	965	Murray Hill
4	107	883	Gramercy
5	68	856	East Chelsea
6	141	816	Lenox Hill West

7	263	778	Yorkville West
8	249	749	West Village
9	230	722	Times Sq/Theatre District
10	229	714	Sutton Place/Turtle Bay North

Nighttime Taxi Activity Analysis (11PM-5AM)

Top Pickup Zones at Night

1. East Village (79) sees the highest number of pickups (2,431), indicating it is a hotspot for nighttime activity.
2. JFK Airport (132) ranks second (2,237 pickups), highlighting frequent late-night airport arrivals.
3. Popular nightlife and entertainment areas like West Village (249), Clinton East (48), and Lower East Side (148)also have high pickups, suggesting strong demand from bars, restaurants, and clubs.

Top Dropoff Zones at Night

1. East Village (79) and Clinton East (48) are the most common dropoff locations, reflecting their appeal as nightlife destinations.
2. Murray Hill (170), Gramercy (107), and East Chelsea (68) have high dropoff counts, indicating late-night residential returns.
3. Sutton Place/Turtle Bay North (229) appears in the top dropoffs but not in pickups, suggesting it is more of a residential area rather than a nightlife hub.

.

3.2.8. Find the revenue share for nighttime and daytime hours

Revenue Share Analysis: Nighttime vs. Daytime

Time Period	Revenue Share (%)
Nighttime (11PM - 5AM)	12.02%
Daytime (6AM - 10PM)	87.98%

Revenue Share Analysis: Nighttime vs. Daytime

1. Daytime (6AM - 10PM) dominates with 87.98% of total revenue, indicating most taxi earnings come from regular commuting hours, business travel, and airport trips.
2. Nighttime (11PM - 5AM) contributes only 12.02%, reflecting lower demand but possibly higher fares due to surcharges and nightlife-related rides.

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

Average Fare Per Mile Per Passenger Count:

Passenger Count	Avg Fare Per Mile ($)	Fare Per Mile Per Passenger ($)
1	5.674	5.674
2	5.305	2.652
3	5.427	1.809
4	5.360	1.340
5	5.690	1.138
6	5.739	0.956

Analysis

1. Solo passengers pay the highest fare per mile ($5.674) since the cost isn't shared.
2. As the number of passengers increases, the per-person fare drops significantly—for instance, with 2 passengers, it reduces to $2.652, and for 6 passengers, it's just $0.956.
3. Fares per mile remain fairly stable overall, but cost efficiency improves for larger groups, making shared rides more economical.

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

Average Fare Per Mile Analysis

By Hour of the Day:

Hour	Avg Fare Per Mile ($)
0	6.54
1	6.52
2	6.54
3	6.37
4	6.10
5	5.91
6	6.24
7	6.93
8	7.68
9	8.02
10	8.18

11	8.43
12	8.60
13	8.48
14	8.37
15	8.47
16	8.39
17	8.30
18	8.16
19	7.78
20	7.25
21	7.03
22	6.89
23	6.62

By Day of the Week:

Day	Avg Fare Per Mile ($)
Monday	7.46
Tuesday	8.04
Wednesday	8.15
Thursday	8.12
Friday	7.83
Saturday	7.67
Sunday	7.07

Analysis

1. Peak Fare Hours (10 AM - 5 PM): Fares per mile peak between 10 AM and 5 PM, with the highest at noon ($8.60 per mile), likely due to demand.
2. Cheaper Late Nights & Early Mornings: Lowest fares are observed between 3 AM - 6 AM ($5.91 - $6.54 per mile).
3. Weekday vs. Weekend Trends: Tuesdays, Wednesdays, and Thursdays have the highest fares (~$8.15 per mile), possibly due to business travel.
4. Sunday has the Lowest Fare ($7.07 per mile): Lower demand and lesser congestion could contribute to cheaper fares.

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

Average Fare Per Mile by Vendor and Hour of the Day

VendorID	Hour	Avg Fare Per Mile ($)

1	0	6.61
1	1	6.55
1	2	6.56
1	3	6.42
1	4	5.78
1	5	5.85
1	6	6.39
1	7	6.97
1	8	7.75
1	9	8.07
1	10	8.27
1	11	8.54
1	12	8.56
1	13	8.49
1	14	8.40
1	15	8.54
1	16	8.40
1	17	8.29
1	18	8.21
1	19	7.71
1	20	7.37
1	21	7.06
1	22	6.88
1	23	6.65
2	0	6.52
2	1	6.52
2	2	6.53
2	3	6.36
2	4	6.19
2	5	5.94
2	6	6.18
2	7	6.92
2	8	7.66
2	9	8.01
2	10	8.15
2	11	8.40
2	12	8.61
2	13	8.48
2	14	8.36
2	15	8.44
2	16	8.39

2	17	8.31
2	18	8.14
2	19	7.79
2	20	7.22
2	21	7.02
2	22	6.89
2	23	6.62

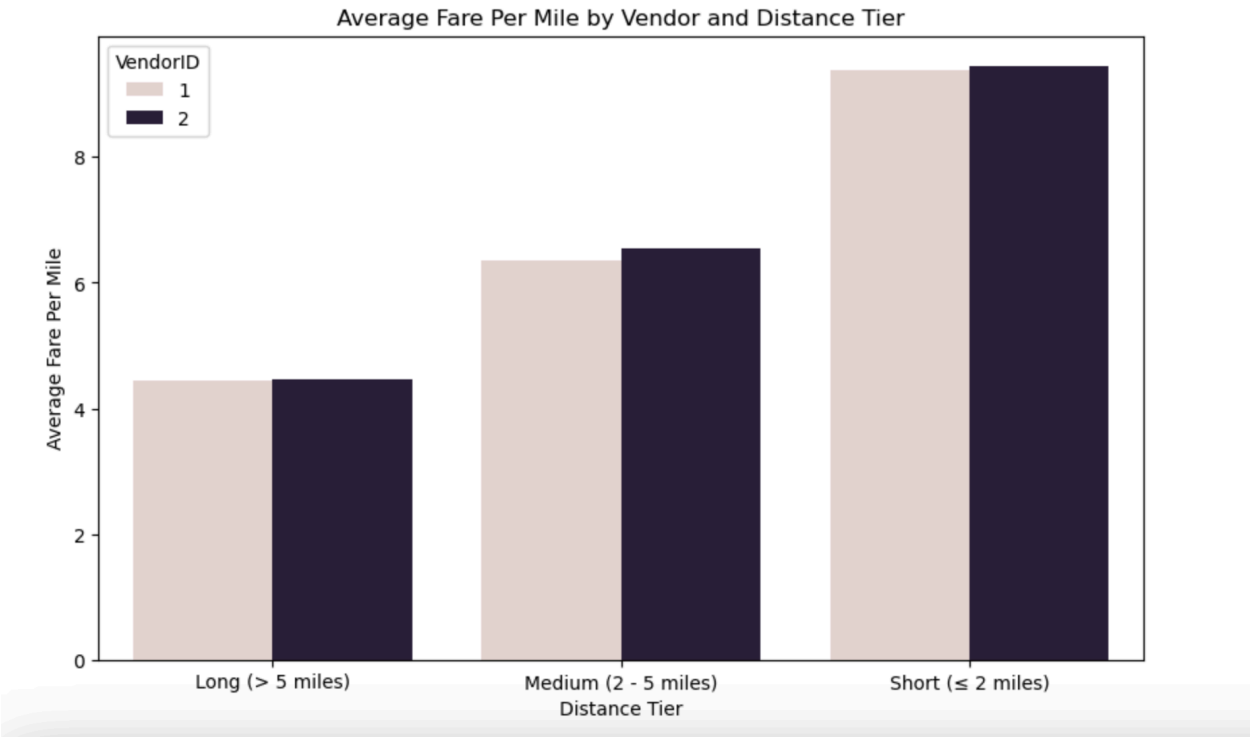
Analysis:

Peak Pricing Hours: Both vendors have the highest fare per mile during mid-morning (8 AM - 12 PM), peaking around 11 AM - 1 PM. The fares start increasing from early morning (6 AM) and decline after 6 PM.

Nighttime Trends: Fares remain relatively lower between 12 AM - 5 AM, with Vendor 1 having slightly higher fares than Vendor 2 during these hours.

Vendor Comparison: Vendor 1 has slightly higher fares than Vendor 2 across most hours, especially in morning and afternoon hours. Both vendors follow similar pricing trends throughout the day.
Evening & Late Night Decline: Fares start decreasing after 6 PM, with the lowest values seen around midnight to early morning. This suggests lower demand or competition affecting prices at night.

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



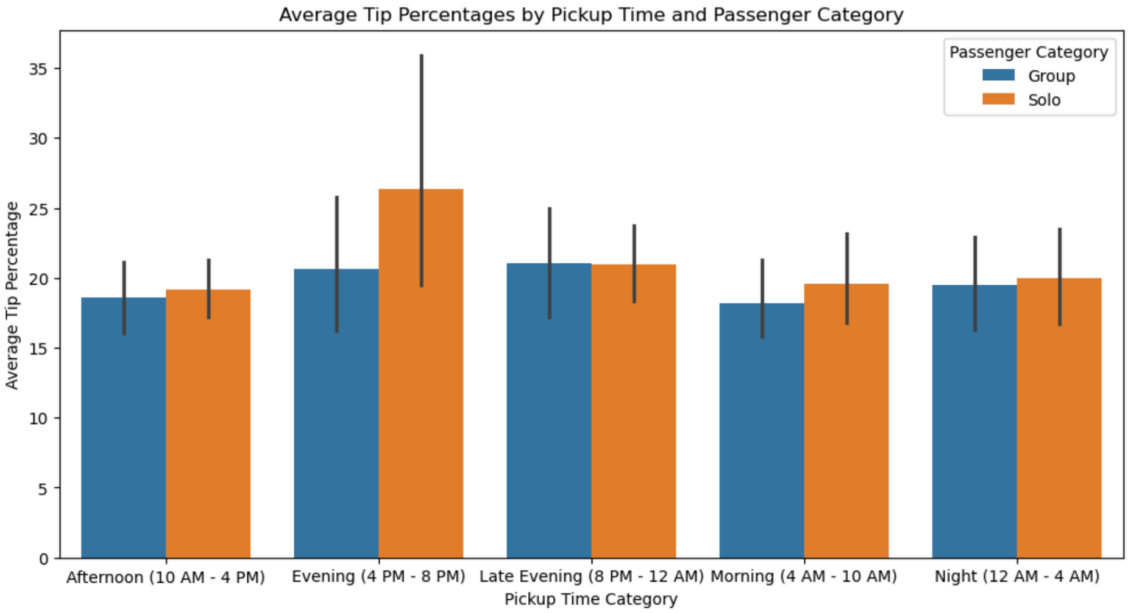
Average Fare Per Mile by Vendor and Distance Category

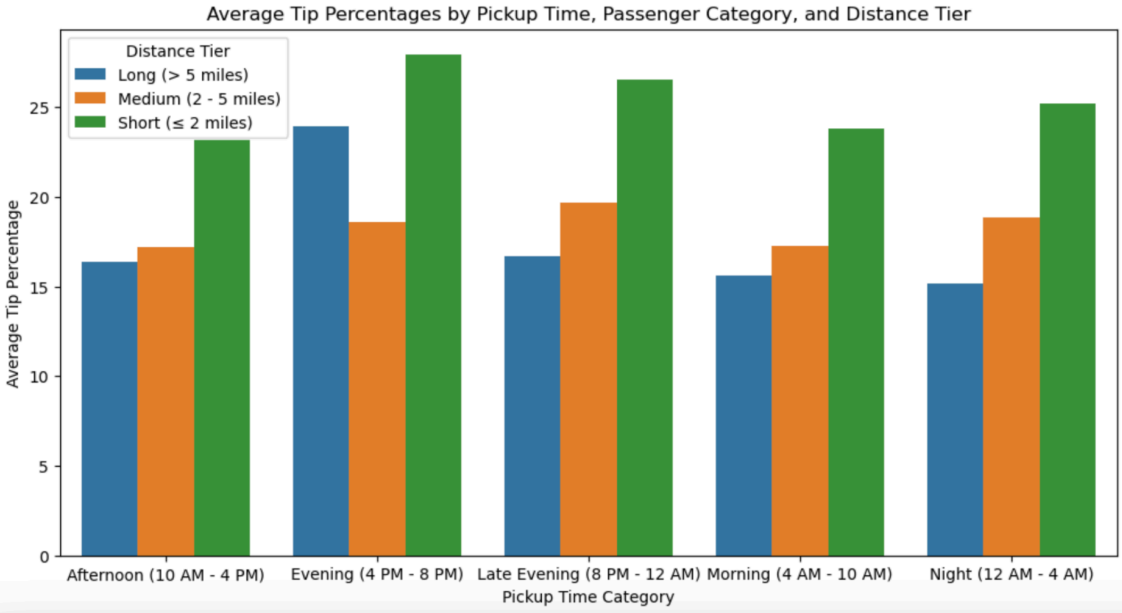
VendorID	Distance Tier	Avg Fare Per Mile
1	Long (> 5 miles)	4.433017
1	Medium (2 - 5 miles)	6.351526
1	Short (≤ 2 miles)	9.383724
2	Long (> 5 miles)	4.469460
2	Medium (2 - 5 miles)	6.547305
2	Short (≤ 2 miles)	9.451976

Analysis:

1. Short trips cost the most per mile – fares are highest for trips ≤ 2 miles (~$9.45/mile).
2. Medium trips (2-5 miles) have moderate rates – around $6.35 - $6.55/mile.
3. Long trips (>5 miles) are cheapest per mile – around $4.43 - $4.46/mile.
4. Vendor 2 is slightly more expensive than Vendor 1 across all distance tiers.
5. Fare per mile decreases as trip distance increases, making longer trips more cost-effective.

3.2.13. Analyse the tip percentages





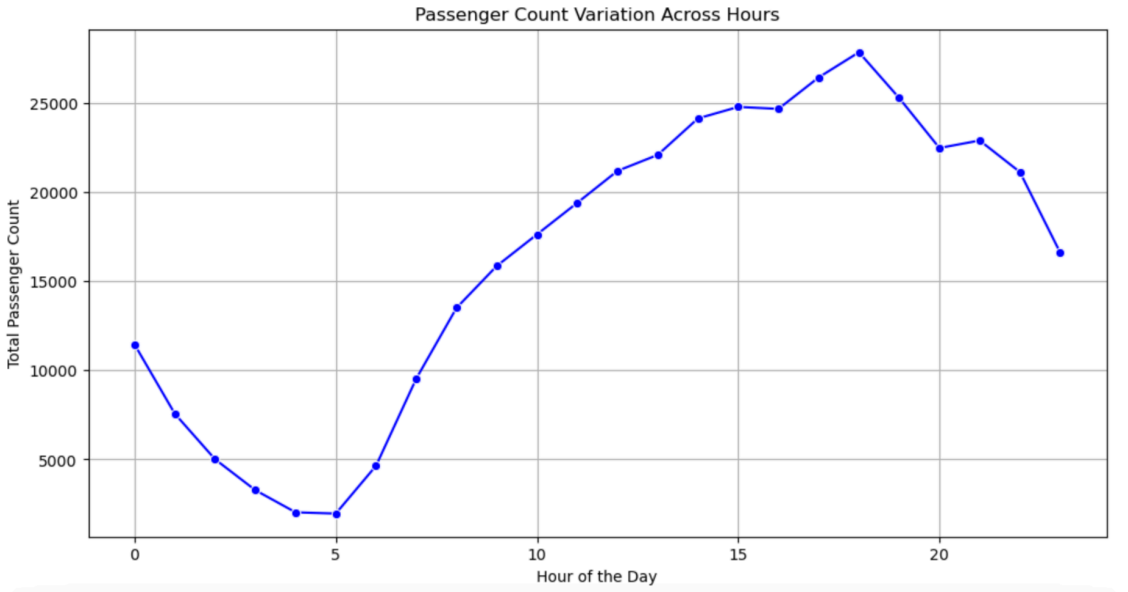
Analysis

1. Short trips (≤2 miles) have the highest tip percentages, especially during Evening (4 PM - 8 PM) and Late Evening (8 PM - 12 AM).
2. Solo passengers generally tip more than groups, likely due to more personal transactions.
3. Night (12 AM - 4 AM) and Early Morning (4 AM - 10 AM) trips have the lowest tip percentages, possibly due to fatigue or hurried travel.
4. Long-distance trips (>5 miles) receive the lowest average tips, possibly because of higher fare amounts leading to lower tipping ratios.

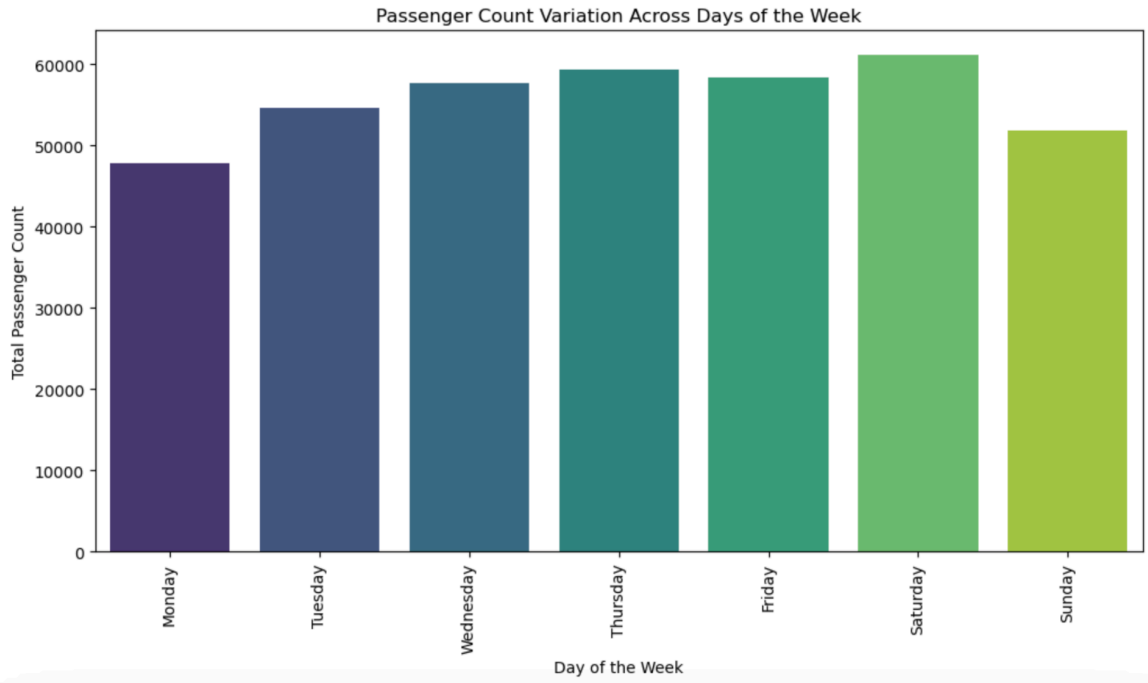
5. Tipping is highest in premium time slots (evening and late evening), potentially due to social outings or leisure trips.
6. Low tips are observed during early morning and late-night hours, possibly due to airport rides or work commutes where tipping is less customary.

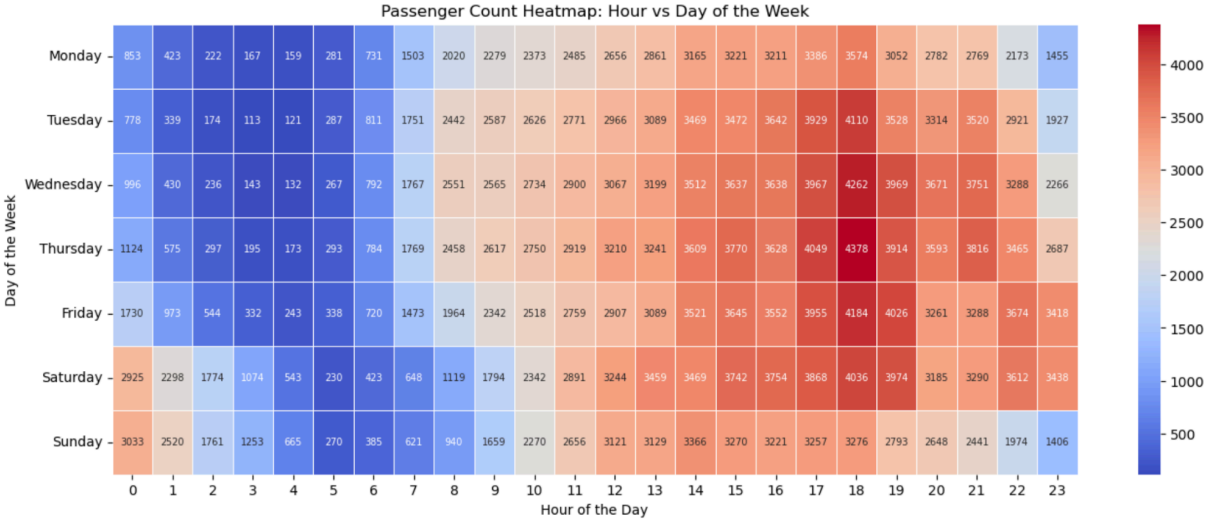
3.2.14. Analyse the trends in passenger count

Passenger Count Variation Across Hours



Passenger counts vary across the days of the week





Passenger Count Analysis Across Hours & Days

Peak Hours:

Weekdays: High passenger count during 6–9 AM (morning rush) & 4–7 PM (evening rush).
Weekends: Peak shifts to late morning (10 AM–1 PM) and afternoon (3–6 PM).

Day-wise Variations:

Monday–Friday: Consistent travel patterns with clear office commute trends.
Saturday: Highest passenger count, likely due to shopping, leisure, and weekend activities.

Sunday: Lowest overall, with a delayed peak around noon, indicating relaxed travel patterns.

Off-Peak Hours:
1–5 AM: Minimal passengers across all days, indicating very low demand during late-night hours.
Post 10 PM: Gradual decline in passenger count as services slow down.

3.2.15. Analyse the variation of passenger counts across zones

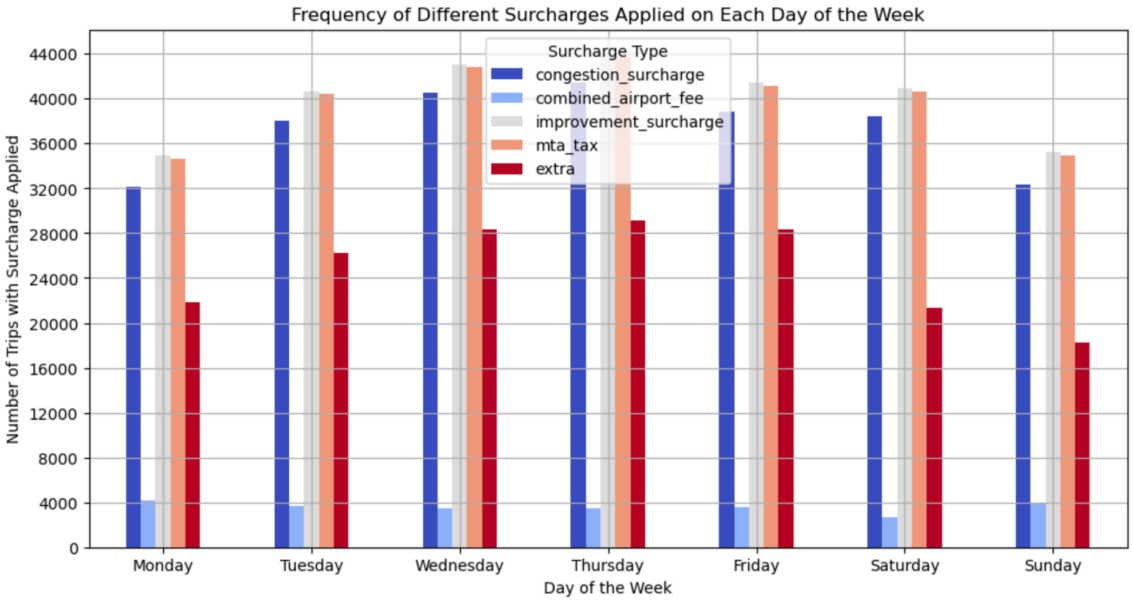
Top 10 Zones by Total Passenger Count:

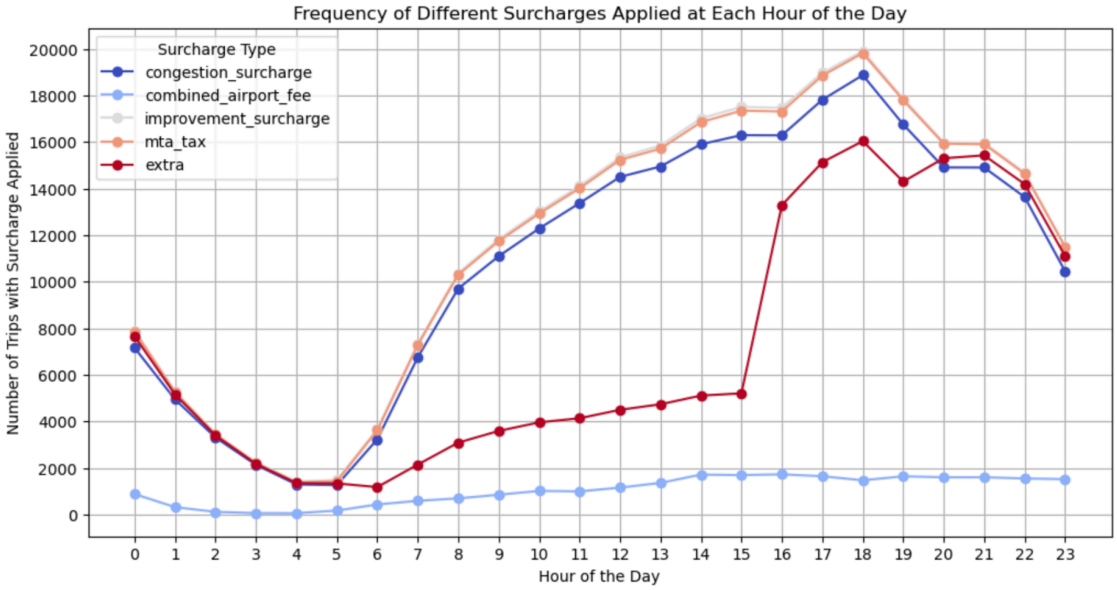
Rank	Zone	Total Passenger Count	Avg Passenger Count	Location ID
1	JFK Airport	22,250	1.508	132
2	Upper East Side South	17,907	1.334	237
3	Midtown Center	18,327	1.399	161
4	Upper East Side North	16,311	1.355	236
5	LaGuardia Airport	13,834	1.376	138
6	Midtown East	13,630	1.363	162
7	Penn Station/Madison Sq West	13,514	1.360	186
8	Times Sq/Theatre District	14,509	1.528	230
9	Lincoln Square East	13,206	1.401	142
10	Murray Hill	11,516	1.348	170

Passenger Count Variation Across Zones

1. JFK Airport has the highest total passenger count (22,250), reflecting its status as a major transportation hub.
2. Manhattan zones dominate the top 10, with high-traffic areas like Midtown, Penn Station, and Times Square attracting many passengers.
3. Airports (JFK & LaGuardia) see higher average passenger counts per trip, likely due to group travel.
4. Midtown Center and Lincoln Square East have moderate total passenger counts but slightly higher average passengers per trip (~1.4).
5. Overall, high-demand zones show a similar average passenger count per trip (1.3–1.5), suggesting mostly solo or small-group travel.

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





Insights on Extra Charges Based on Time and Location:

Day of the Week Analysis (First Chart):

1. Extra charges (red bars) are consistently applied throughout the week.
2. The frequency is highest from Tuesday to Friday and relatively lower on Sundays.

Hour of the Day Analysis (Second Chart):

1. Extra charges peak in the late afternoon and evening (16:00 - 22:00), aligning with peak traffic hours.
2. There is a smaller peak around midnight, possibly due to late-night ride demand.
3. The lowest frequency occurs in the early morning (3:00 - 6:00).

4. Conclusions

4.1. Final Insights and Recommendations

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

1. **AI-Driven Dispatching:** Use real-time demand forecasting for efficient taxi allocation.
2. **Reduce Inefficiencies:** Minimize deadhead miles, optimize routes, and encourage ride pooling.
3. **Demand-Based Fleet Allocation:** Deploy more cabs near business hubs during peak hours and nightlife zones at night.
4. **Smart Route Optimization:** Use real-time traffic data and GPS to avoid congestion and reduce costs.
5. **Event & Weather-Based Adjustments:** Increase cab supply near transit hubs during bad weather and major events.

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. **Zone-Based Allocation:** Increase cab availability in high-demand areas like business districts, airports, and entertainment hubs during peak times.
2. **Dynamic Heatmap Deployment:** Use real-time trip data to reposition cabs to emerging high-demand areas.
3. **Time-Based Redistribution:** Focus on residential areas in the morning, business districts in the afternoon, and nightlife zones in the evening.
4. **Driver Incentives:** Encourage movement to underserved areas through bonuses and geofencing alerts.
5. **Event & Weather-Based Adjustments:** Boost cab presence near stadiums, transit hubs, and key locations during special events and bad weather.

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

1. **Dynamic Pricing:** Adjust fares based on demand surges, peak hours, and trip distances to maximize revenue while staying competitive.
2. **Competitive Benchmarking:** Regularly compare pricing with competitors and adjust rates accordingly to maintain market share.
3. **Off-Peak Discounts:** Offer lower fares or ride incentives during slow hours to boost demand.
4. **Event-Based Pricing:** Implement temporary fare increases around major events, holidays, and weather-related demand spikes.
5. **Subscription & Loyalty Programs:** Introduce ride passes and discounts for frequent riders to encourage repeat business and customer retention.