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

There are 12 parquet files for analysis. I combined all 12 files taking sample of ‘0.008’ to achieve total data frame count to 2,50,000 to 3,00,000. With this approach I settled for 303427 rows.

2. Data Cleaning

2.1. Fixing Columns

2.1.1. Fix the index

Drop below colums as their values are constant, giving very less to no room for meaningfull analysis

dataframe = dataframe.drop('improvement_surcharge', axis = 1) # improvement_surcharge has 1 for: 100% of total data.

dataframe = dataframe.drop('tolls_amount', axis = 1) #tolls_amount is 0 for: 92% of total data.

dataframe = dataframe.drop('extra', axis = 1) # extra is 0 for: 40% of total data.

dataframe = dataframe.drop('mta_tax', axis = 1) # mta_extra is 0.5 for: 99% of total data.

dataframe = dataframe.drop('Unnamed: 0', axis = 1) # after concat of parquet files this column was added, which does not have any title.

2.1.2. Combine the two airport_fee columns

2.2. Handling Missing Values

Below is the percentage of missing values from dataset
dataframe.isnull().sum() * 100 / len(dataframe)
VendorID 0.000000
tpep_pickup_datetime 0.000000
tpep_dropoff_datetime 0.000000
passenger_count 3.400488
trip_distance 0.000000
RatecodeID 3.400488
store_and_fwd_flag 3.400488
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 3.400488
Airport_fee 11.232026
airport_fee 92.168462

2.2.1. Find the proportion of missing values in each column

print('passenger_count: ', ((dataframe.passenger_count.isnull().sum() / 303427) * 100)) # 3% 293109
print('RatecodeID: ', (dataframe.RatecodeID.isnull().sum() / 303427) * 100) # 3% 293109
print('store_and_fwd_flag: ',(dataframe.store_and_fwd_flag.isnull().sum() / 303427) * 100) # 3% store_and_fwd_flag 293109
print('Airport_fee: ', (dataframe.Airport_fee.isnull().sum() / 303427) * 100) # 3% Airport_fee 293109

2.2.2. Handling missing values in passenger_count

There were 4618 rows with missing passenget_count, imputed the missing values with mode of passenget_count which is 1.
dataframe['passenger_count'].apply(lambda x: 1.0 if x == 0.0 else x)

2.2.3. Handle missing values in RatecodeID

There were 10318 rows with missing value for RatecodeID. Imputed them with mode of RatecodeID which is 1.0.

2.2.4. Impute NaN in congestion_surcharge

There were 10318 rows with missing value for congestion_surcharge. Imputed them with mode of congestion_surcharge which 2.5

2.3. Handling Outliers and Standardising Values

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

a. dataframe[dataframe.passenger_count > 6]

There were 2 rows with passenger count more than 7 dropped those rows.

b. dataframe[(dataframe.trip_distance == 0) & (dataframe.fare_amount > 300)]

There were 6 rows with trip distance between 0 and more than 300 miles. Dropped those rows.

c. dataframe[(dataframe.trip_distance == 0) & (dataframe.fare_amount == 0) & (dataframe.PULocationID != dataframe.DOLocationID)]

There were 6 rows with distance = 0, fare amount = 0 and pickup and drop off location id was different, dropped those rows.

d. dataframe[(dataframe.trip_distance > 250)]

There were 3 rows with trip distance more than 250 miles, dropped those rows.

e. dataframe[dataframe.payment_type == 0]

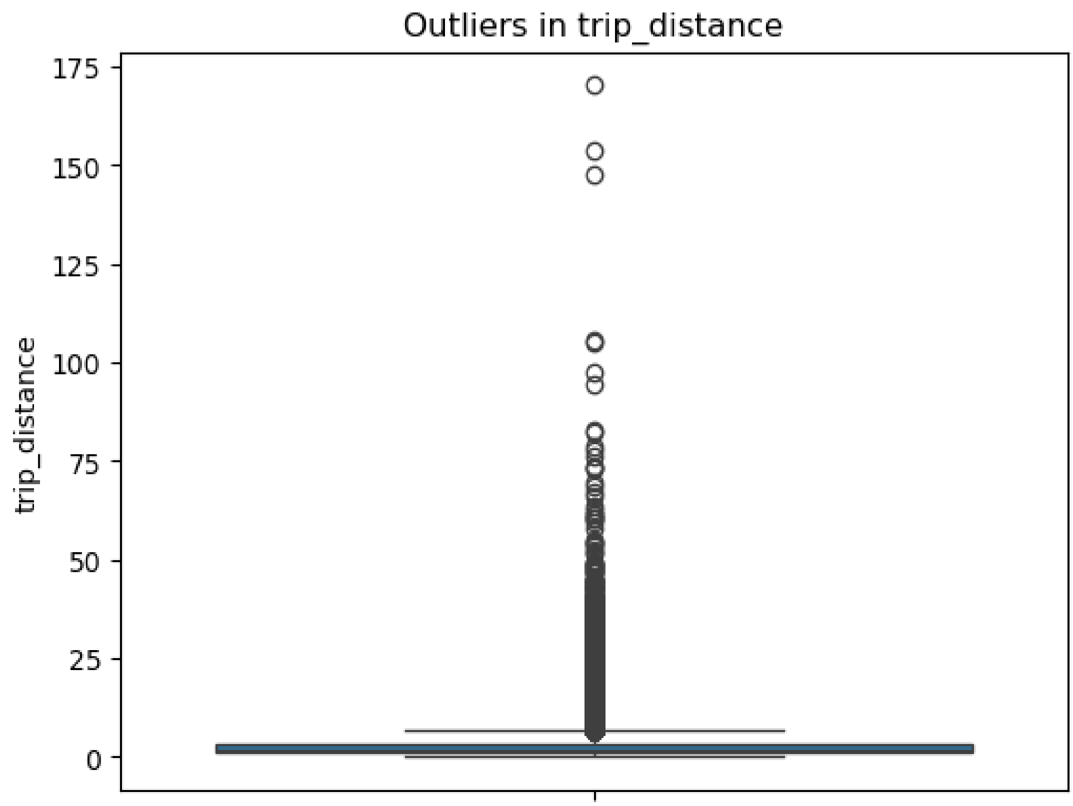
There were 10315 rows with payment type as 0, which is not present in data dictionary hence dropped those 10315 rows.

f. dataframe[(dataframe['RatecodeID'] == 99)]

There were 1672 rows with RatecodeID as 99, which is not present in data dictionary.

g. dataframe[dataframe.trip_distance > 75].index

There were 12 rows with trip distance more than 75 miles which is clear outlier in comparison with rest of the data, hence dropped them.



3. Exploratory Data Analysis

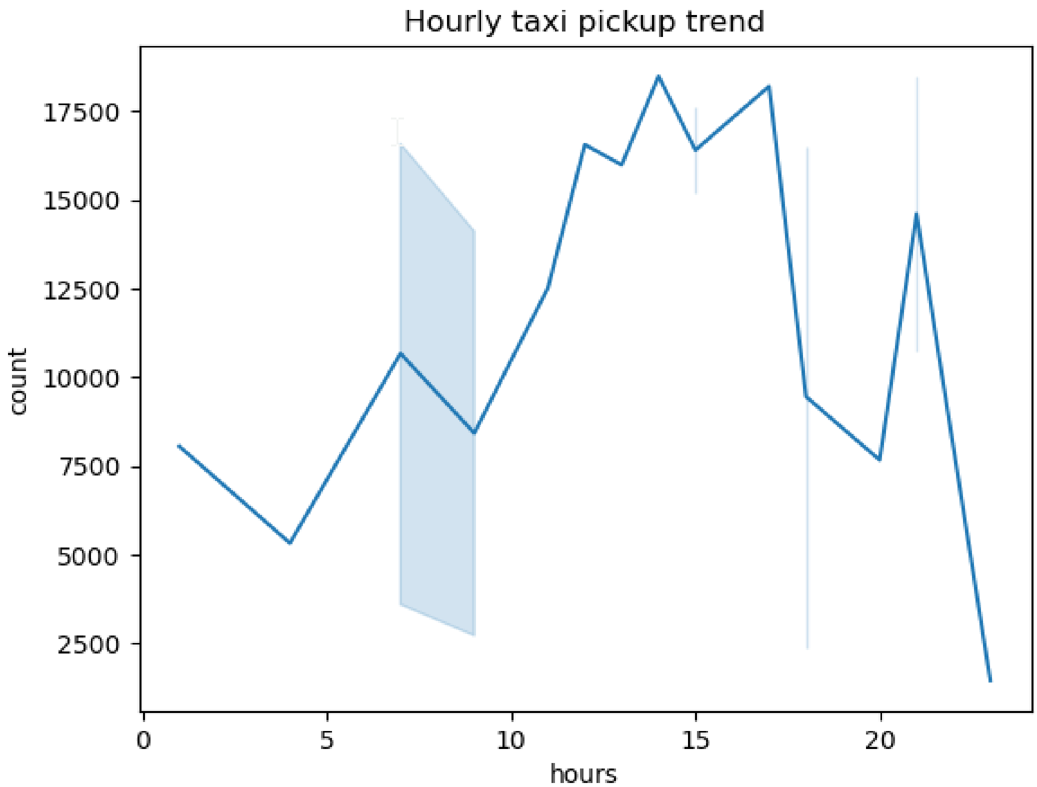
3.1. General EDA: Finding Patterns and Trends

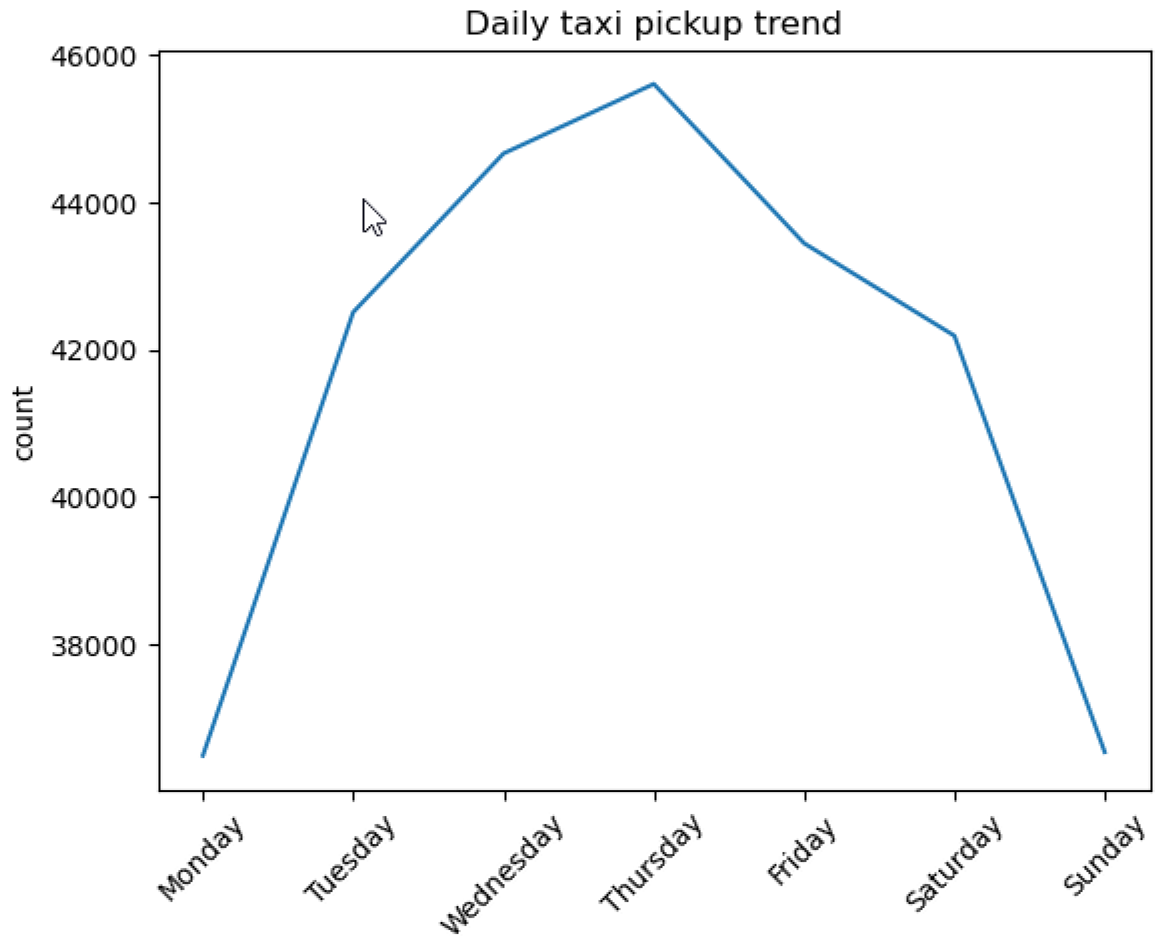
3.1.1. Classify variables into categorical and numerical

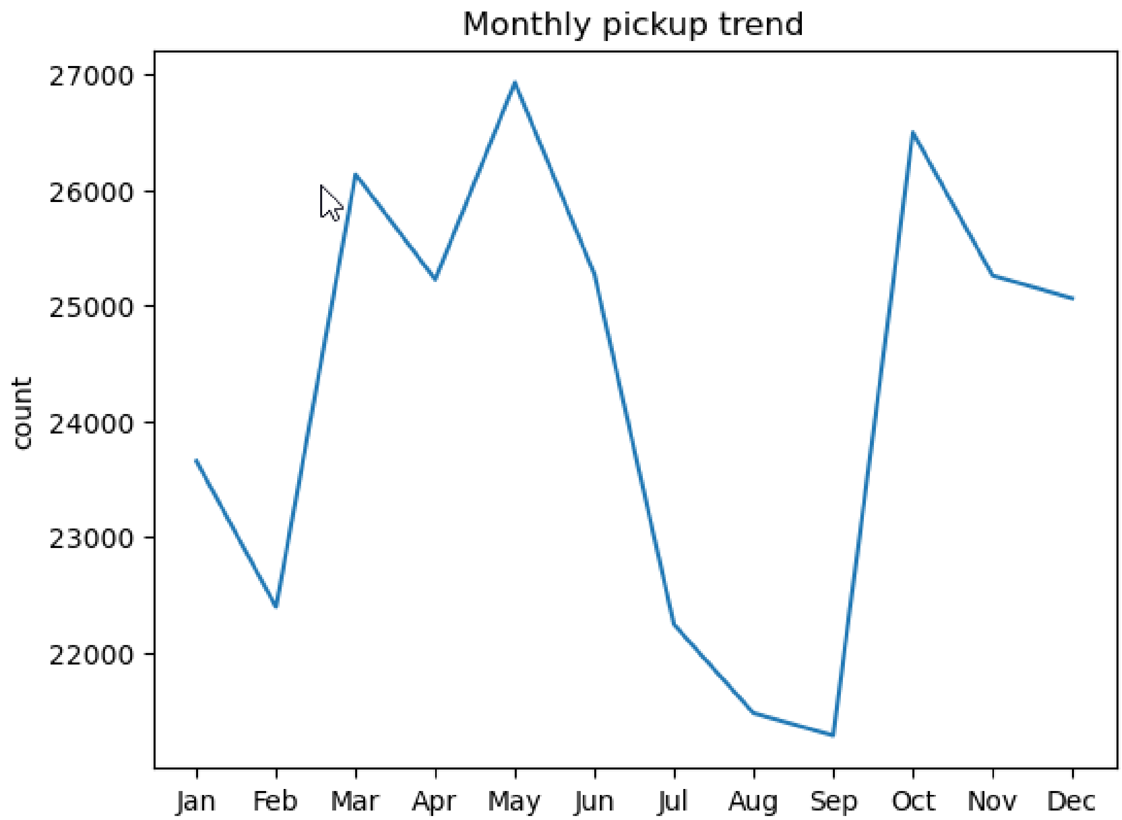
Categorical columns	**Numerical columns**
tpep_pickup_datetime	passenger_count
tpep_dropoff_datetime	trip_distance
RatecodeID	pickup_hour
PULocationID	trip_duration
DOLocationID	fare_amount
payment_type	extra
	mta_tax

	tip_amount tolls_amount improvement_surcharge total_amount congestion_surcharge airport_fee

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




Daily taxi pickup trend
count
46000
44000
42000
40000
38000
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday



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

a. dataframe[dataframe['fare_amount'] == 0]['fare_amount'].count()

There were 78 rows with fare_amount less than 0 dropped them.

index_zero_fare_amount = dataframe[dataframe['fare_amount'] <= 0].index

dataframe.drop(index = index_zero_fare_amount, inplace=True)

b. dataframe[dataframe['total_amount'] <= 0]['total_amount'].count())

There were 46 entries with total_amount <= 0, dropped those rows, which are already counted in the fare_amount == 0 count

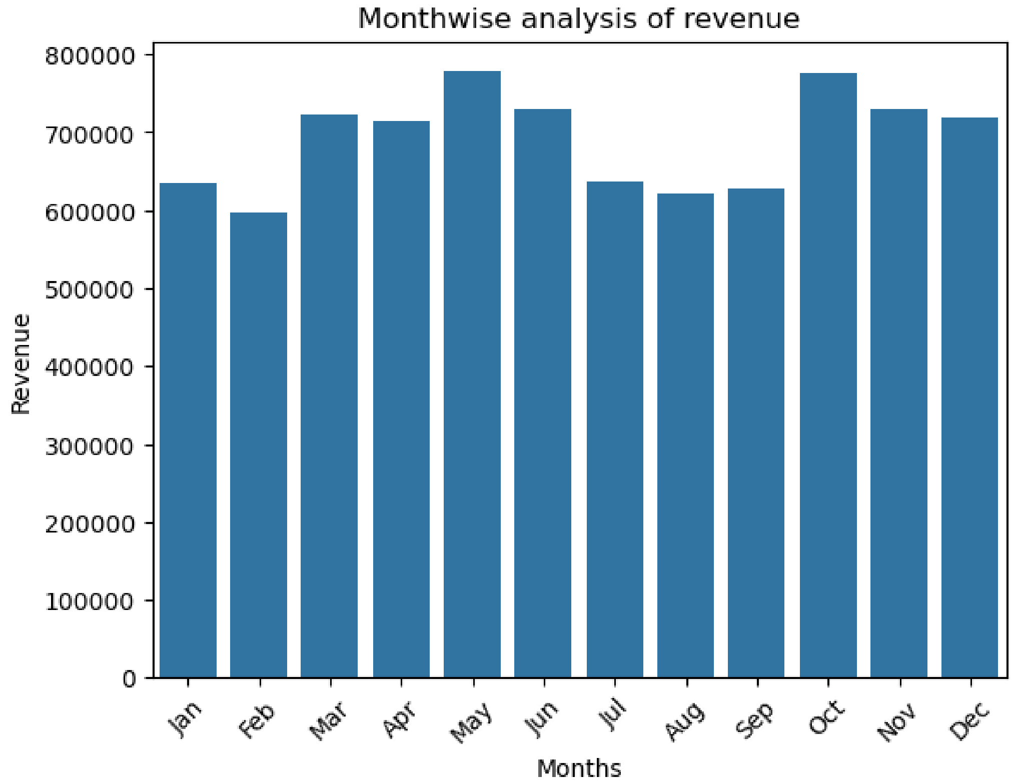
c. dataframe[dataframe['trip_distance'] <= 0]['trip_distance'].count()

There were 3367 entries with trip distance less than or eqaul to 0, dropped those rows.

index_zero_trip_distance = dataframe[dataframe['trip_distance'] == 0].index

```
dataframe.drop(index = index_zero_trip_distance, inplace=True) #288044
```

3.1.4. Analyze the monthly revenue trends



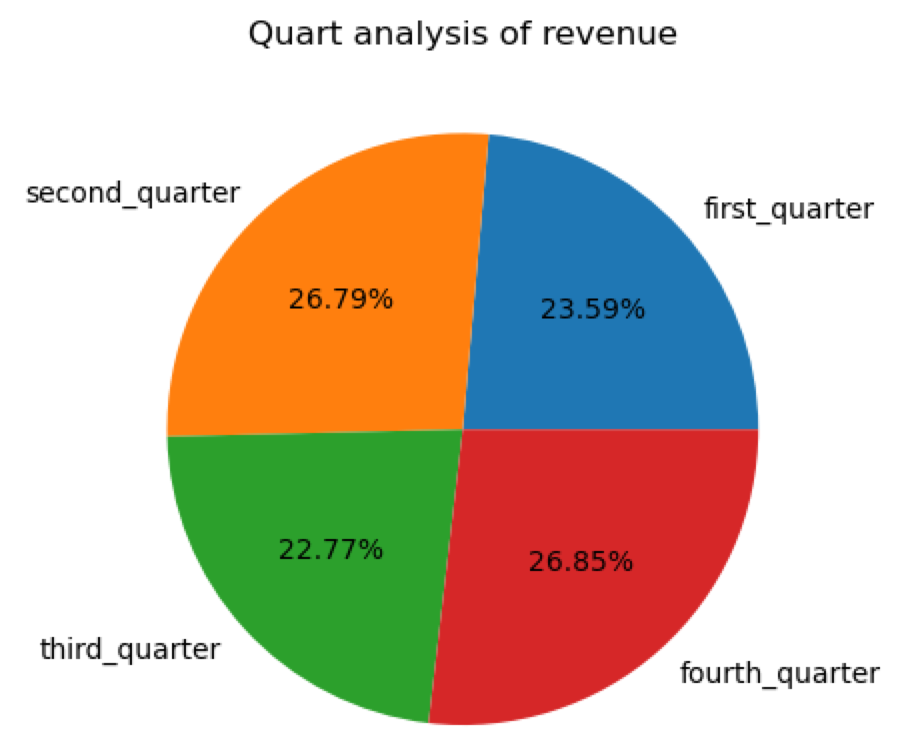
As of 2023 Mexican population in NYC was about 30% (28.4%).

Mexicans celebrate All Saints Day which was on Nov1, Halloween on Oct 31 and All Souls' Day on Nov2, which contributed the hike in Oct, Nov and Dec which year end and Christmas.

Indian festivals like Dasara and Diwali also fall in the months of Oct-Nov, contributing to hike in taxi bookings.

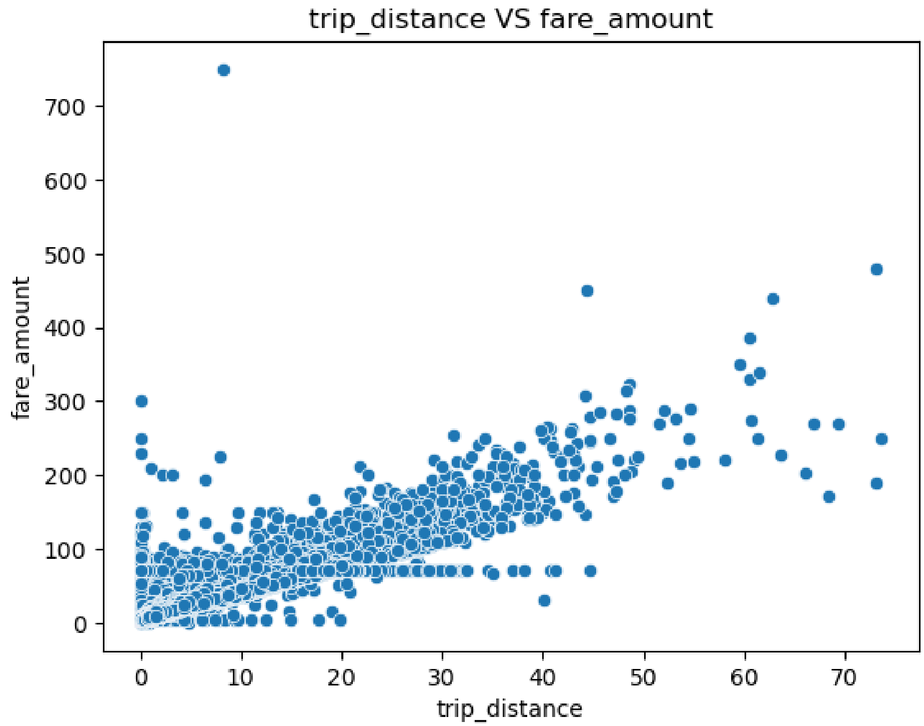
May and Jun see spike because Memorial Day: Celebrated from May 25–31, Memorial Day is a floating Monday.

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



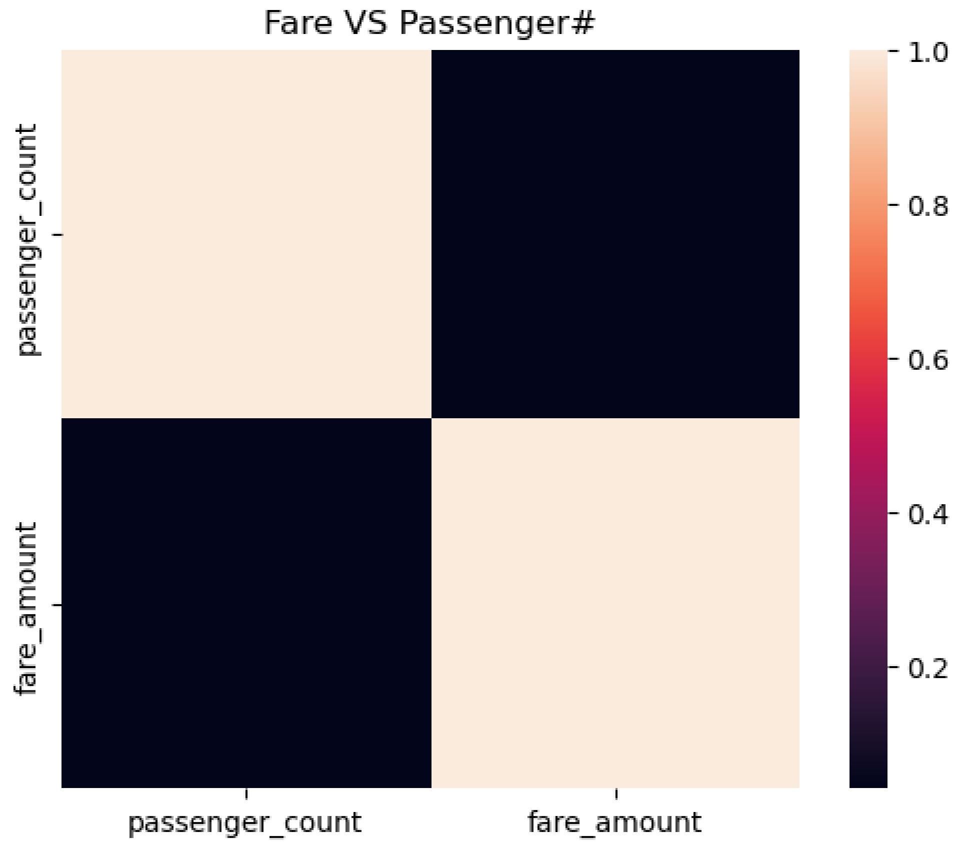
The fourth quarter is full of festivals and holidays. Which contributes to the high number of taxi bookings.

3.1.6. Analyze and visualize the relationship between distance and fare amount



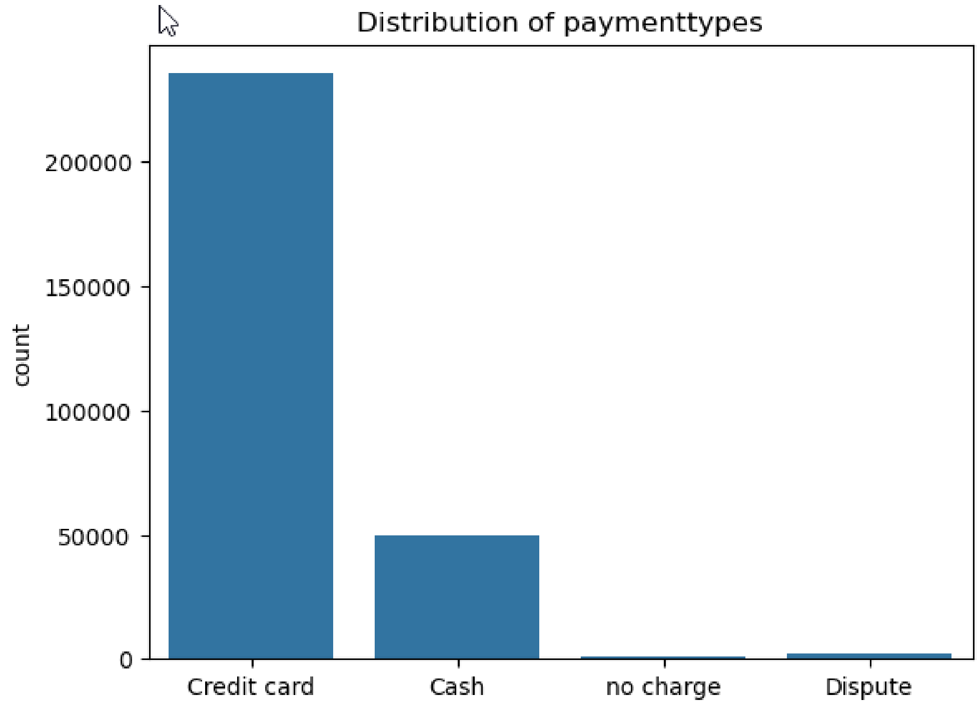
As the distance increases fare also increases. Fare and distance are co-related at 0.95.

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



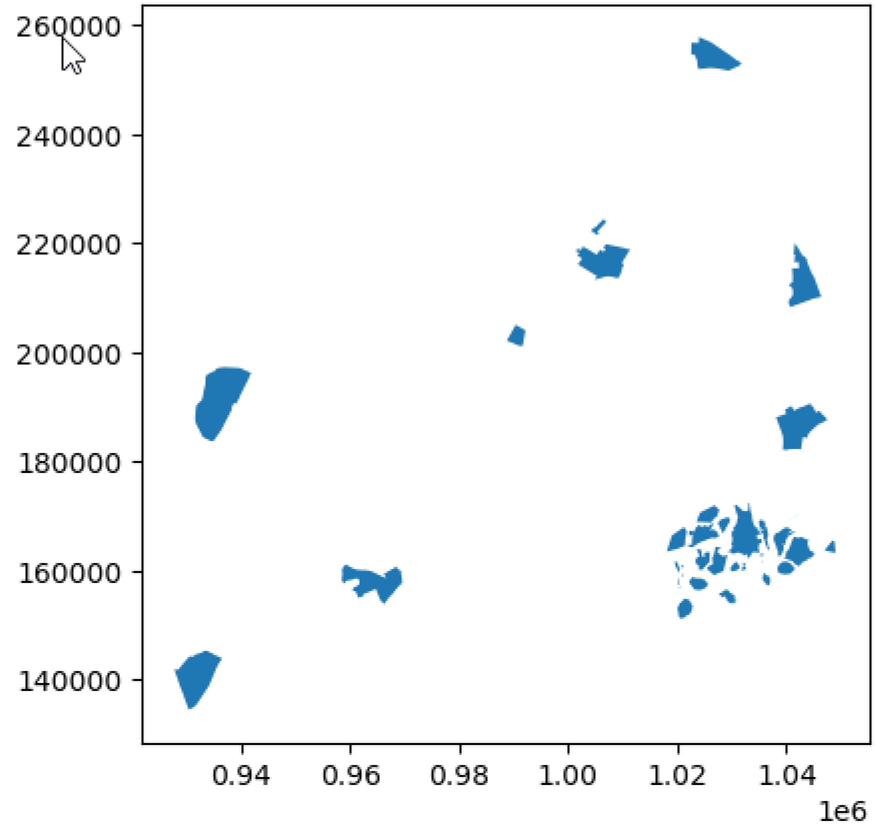
Passenger count and fare are correlated at only 0.04%. Which means fare amount is not affected by passenger count.

3.1.8. Analyze the distribution of different payment types



The highest payment is done using Credit card followed by cash.

3.1.9.



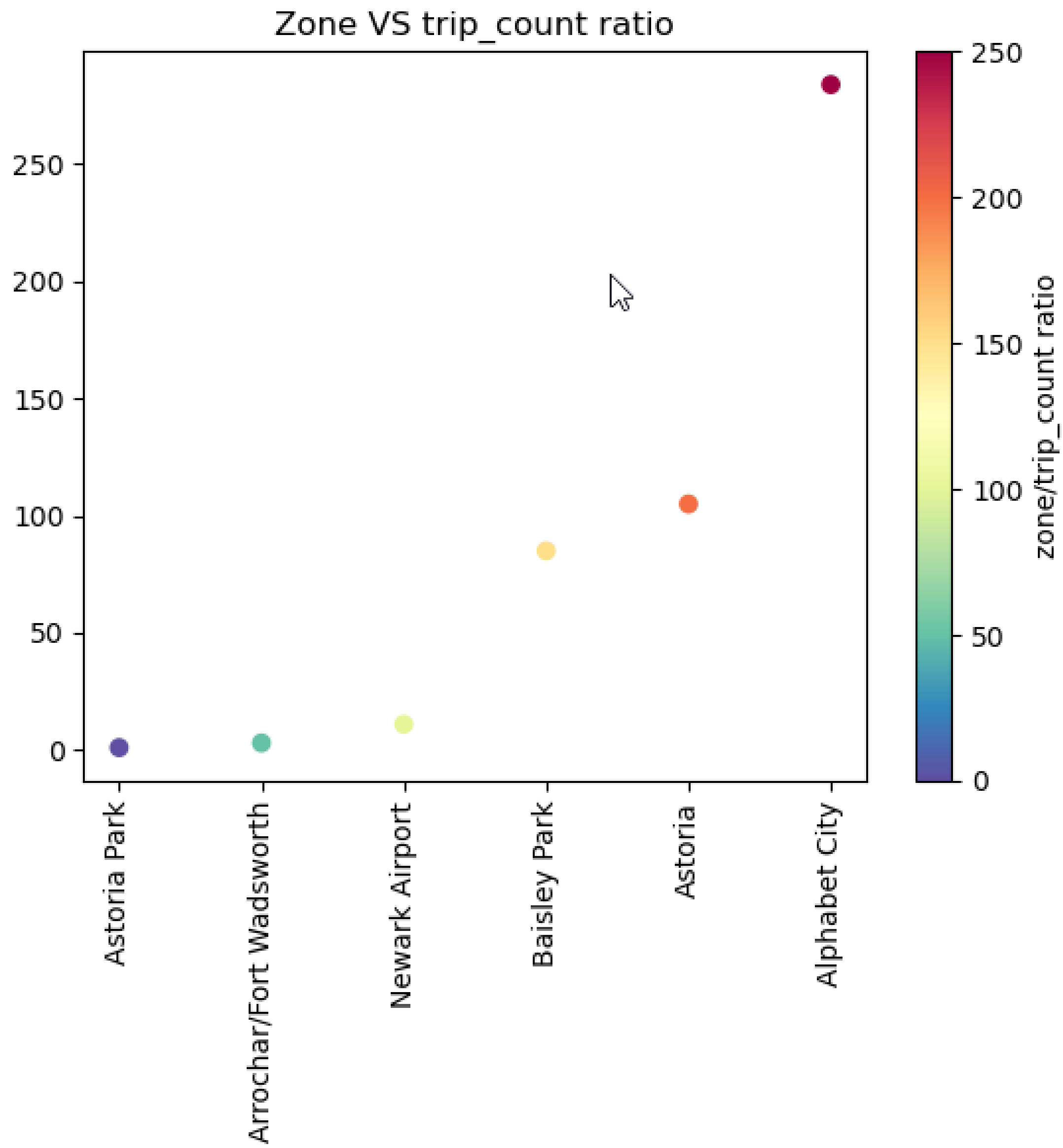
3.1.10. Merge the zone data with trips data

```
# Merge zones and trip records using locationID and PULocationID

zones['PULocationID'] = zones['LocationID']

zones = zones.drop('LocationID', axis = 1)

merged_df = pd.merge(dataframe, zones, on='PULocationID', how='inner')

merged_df.to_csv(r'D:\Savitri\EDA\Assignment\merged_df.csv')

merged_df.head()
```

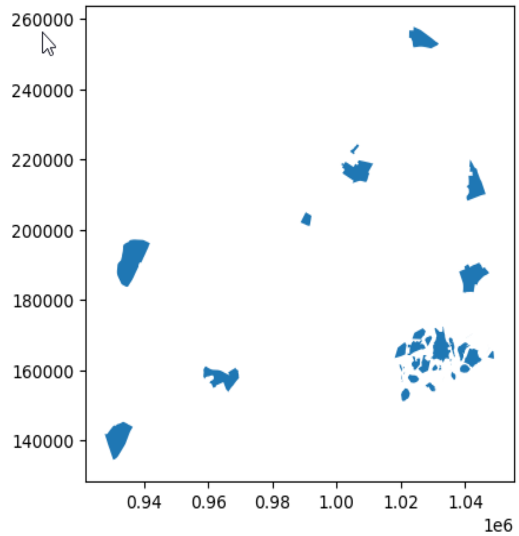
VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	...	pickuptime	droptime	trip_duration	miliseconds	OBJECTID	Shape_Leng	Shape_Area	zone	borough	geome
2	2023-09-14 00:51:33	2023-09-14 01:03:12	1.0	2.23	1.0	N	4	90	1	...	2023-09-14 00:51:33	2023-09-14 01:03:12	0 days 00:11:39	699000	4	0.043567	0.000112	Alphabet City	Manhattan	POLYG ((992073. 203714.(992068.
2	2023-09-19 18:28:04	2023-09-19 19:16:27	1.0	14.47	2.0	N	10	186	1	...	2023-09-19 18:28:04	2023-09-19 19:16:27	0 days 00:48:23	2903000	10	0.099839	0.000436	Baisley Park	Queens	POLYG ((1044355. 190734.: 1044612.
2	2023-09-02 13:46:41	2023-09-02 14:01:48	1.0	1.26	1.0	N	7	179	2	...	2023-09-02 13:46:41	2023-09-02 14:01:48	0 days 00:15:07	907000	7	0.107417	0.000390	Astoria	Queens	POLYG ((1010804. 218919.(1011049.
2	2023-09-15 12:50:02	2023-09-15 12:52:37	1.0	0.29	1.0	N	10	10	2	...	2023-09-15 12:50:02	2023-09-15 12:52:37	0 days 00:02:35	155000	10	0.099839	0.000436	Baisley Park	Queens	POLYG ((1044355. 190734.: 1044612.
2	2023-09-18 16:01:05	2023-09-18 17:14:28	1.0	14.20	2.0	N	10	161	1	...	2023-09-18 16:01:05	2023-09-18 17:14:28	0 days 01:13:23	4403000	10	0.099839	0.000436	Baisley Park	Queens	POLYG ((1044355. 190734.: 1044612.

ws × 28 columns

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

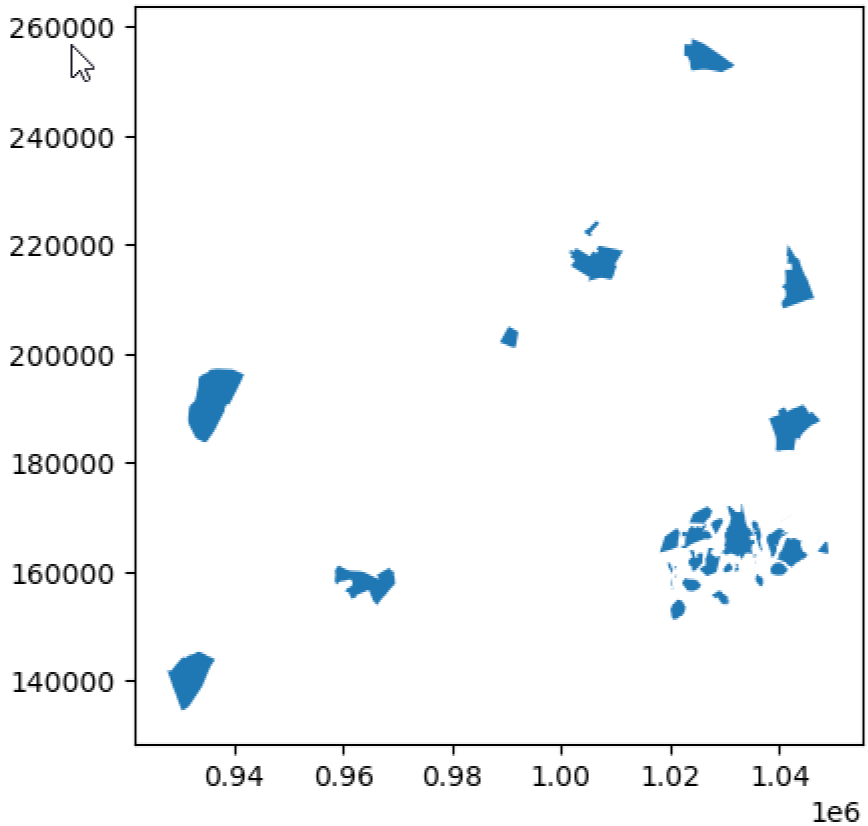


3.1.12.



	OBJECTID	Shape_Leng	Shape_Area	zone	borough	geometry	PULocationID	trip_counts
8	9	0.099784	0.000338	Auburndale	Queens	POLYGON ((1043803.993 216615.925, 1043849.708 ...	9	1.0
6	7	0.107417	0.000390	Astoria	Queens	POLYGON ((1010804.218 218919.641, 1011049.165 ...	7	3.0
1	2	0.433470	0.004866	Jamaica Bay	Queens	MULTIPOLYGON (((1033269.244 172126.008, 103343...	2	11.0
7	8	0.027591	0.000027	Astoria Park	Queens	POLYGON ((1005482.276 221686.466, 1005304.898 ...	8	105.0
4	5	0.092146	0.000498	Arden Heights	Staten Island	POLYGON ((935843.310 144283.336, 936046.565 14...	5	284.0

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



3.1.14. Conclude with results

Arden Heights having the highest taxi bookings with 284 trip followed by Astoria Park with 105, Jamaica Bay with 11, Astoria with 3 and last stands the Auburndale with only 1 booking.

3.2. Detailed EDA: Insights and Strategies

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

merged_df['trip_speed'] = merged_df['trip_distance'] / ((merged_df['miliseconds']/1000)/60)

merged_df.sort_values('trip_speed', axis = 0,ascending=True, inplace=True, na_position='last')

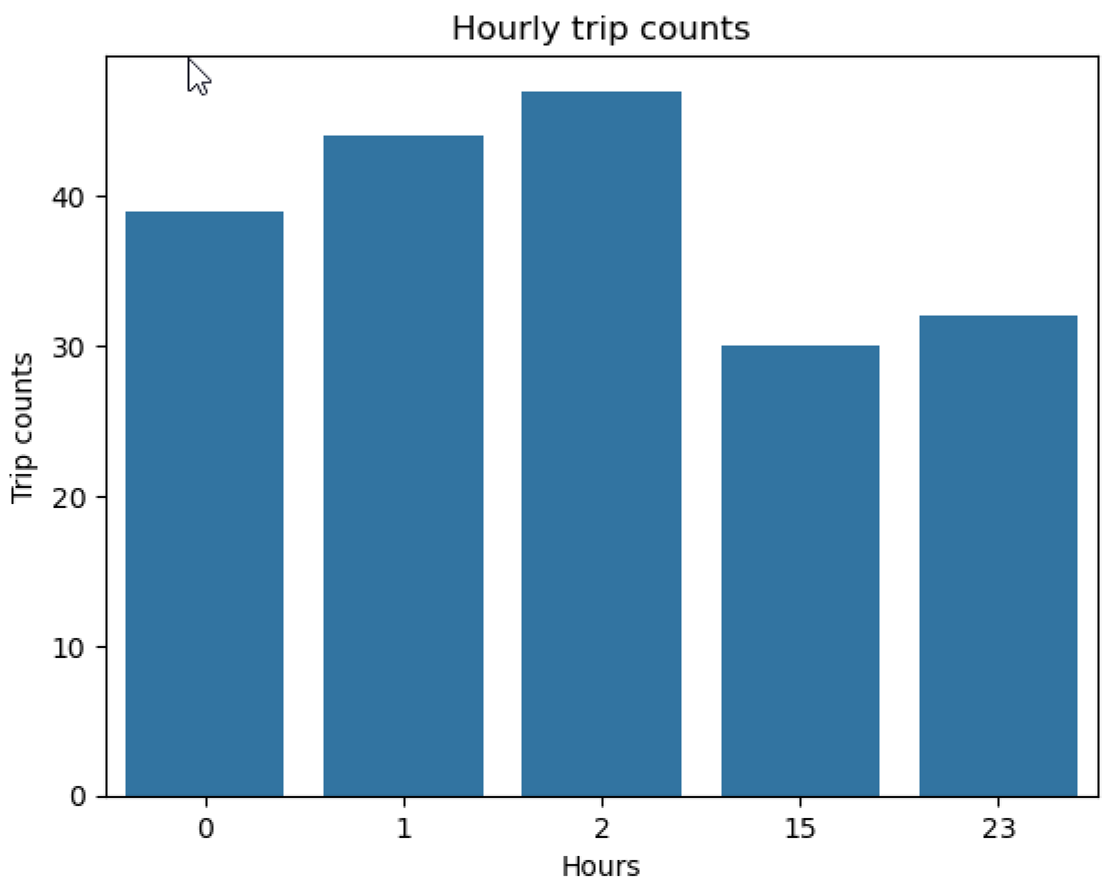
merged_df.head()

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocationID	payment_type	...	droptime	trip_duration	miliseconds	OBJECTID	Shape_Leng	Shape_Area	zone	borough	geometry	trip_speed
356	2	2023-11-05 01:56:20	2023-11-05 01:08:42	2.0	3.45	1.0	N	4	88	1	...	2023-11-05 01:08:42	-1 days +23:12:22	83542000	4	0.043567	0.000112	Alphabet City	Manhattan	POLYGON ((992073.467 203714.076, 992068.667 20...	0.002478
354	2	2023-11-22 15:21:16	2023-11-22 15:29:14	1.0	0.08	1.0	N	7	179	1	...	2023-11-22 15:29:14	0 days 00:07:58	478000	7	0.107417	0.000390	Astoria	Queens	POLYGON ((1010804.218 218919.641, 1011049.165 ...	0.010042
426	2	2023-10-08 01:51:41	2023-10-08 02:52:39	1.0	1.58	1.0	N	4	232	1	...	2023-10-08 02:52:39	0 days 01:00:58	3658000	4	0.043567	0.000112	Alphabet City	Manhattan	POLYGON ((992073.467 203714.076, 992068.667 20...	0.025916
281	2	2023-03-12 15:29:01	2023-03-12 15:29:23	2.0	0.01	5.0	N	1	1	1	...	2023-03-12 15:29:23	0 days 00:00:22	22000	1	0.116357	0.000782	Newark Airport	EWR	POLYGON ((933100.918 192536.086, 933091.011 19...	0.027273
380	1	2023-11-19 08:28:15	2023-11-19 08:32:06	2.0	0.20	1.0	Y	7	7	1	...	2023-11-19 08:32:06	0 days 00:03:51	231000	7	0.107417	0.000390	Astoria	Queens	POLYGON ((1010804.218 218919.641, 1011049.165 ...	0.051948

5 rows × 29 columns

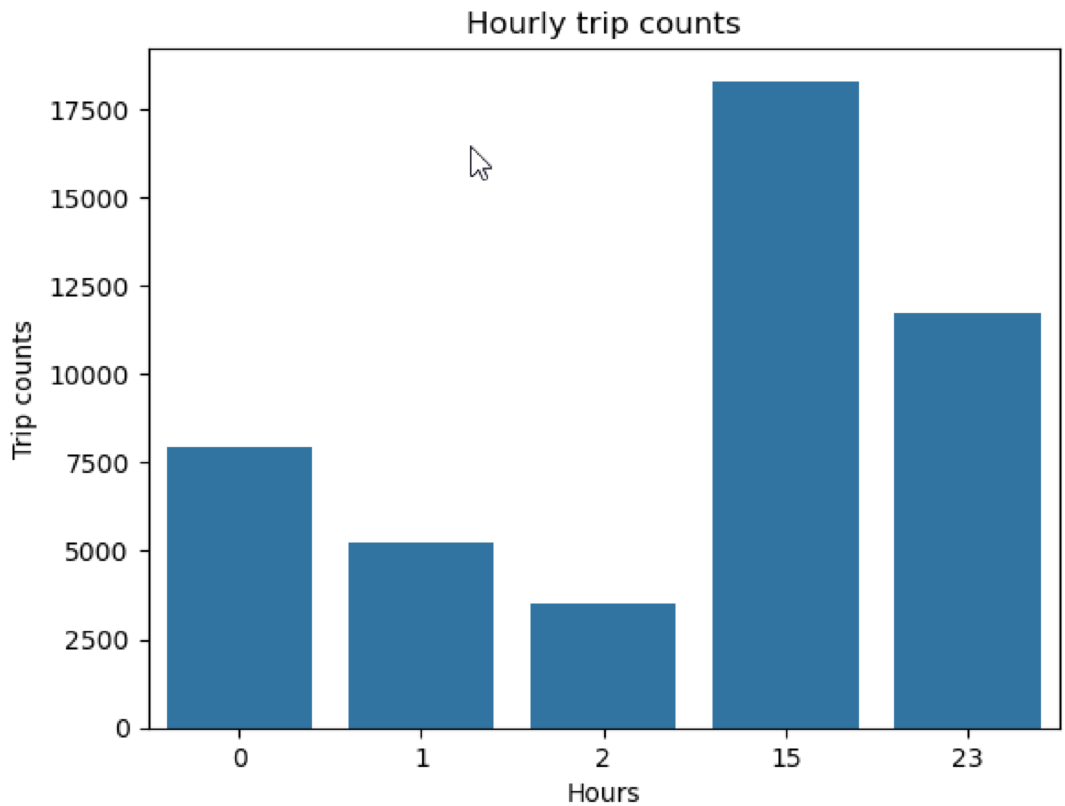
The slowest zone is Alphabet City.

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



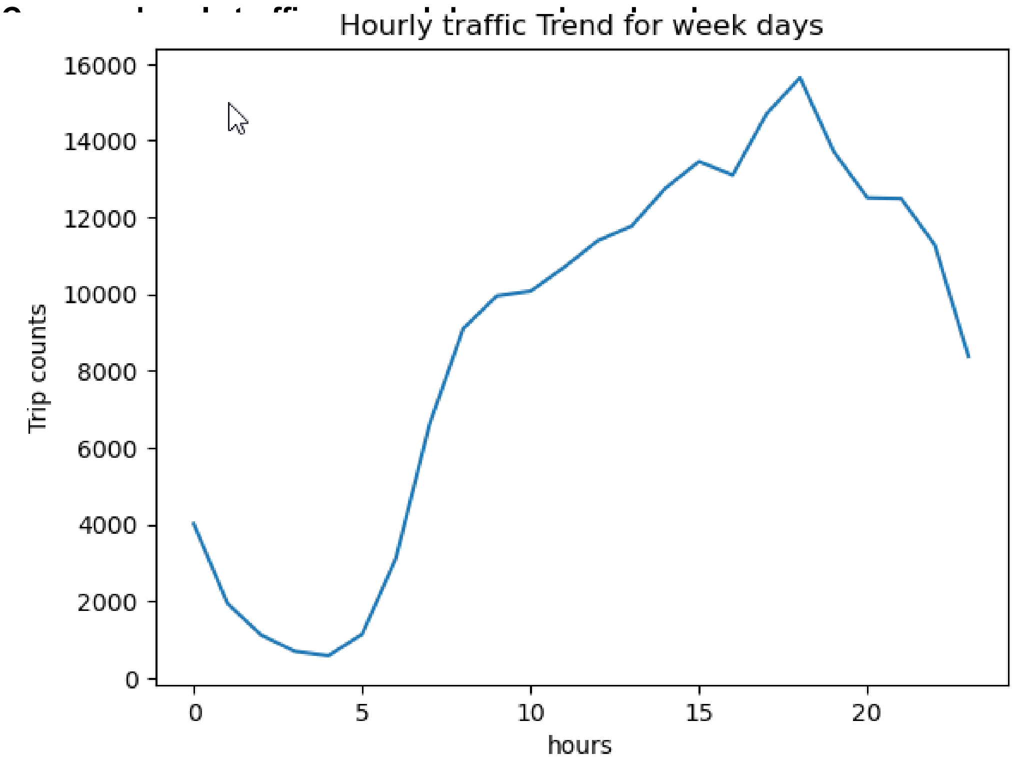
Busiest hours of the days are from are at 1PM and 2PM.

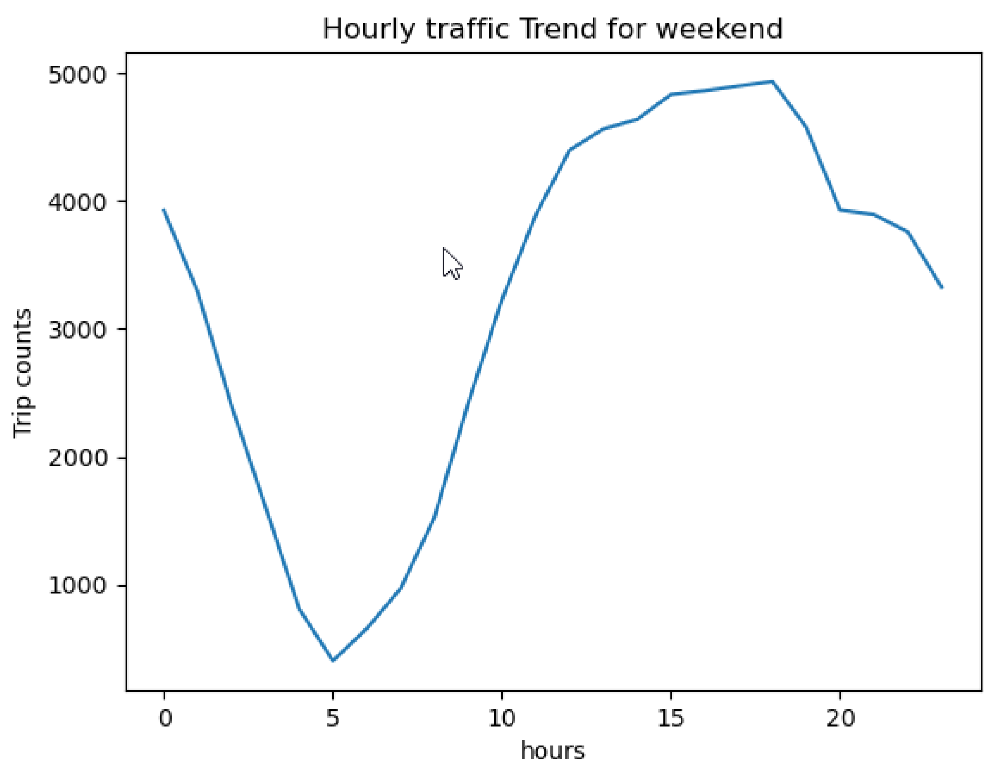
3.2.3. Scale up the number of trips from above to find the actual number of trips



Busiest hours of the day is 3PM when scaled up for huge dataset.

3.2.4.


Hourly traffic Trend for week days
Trip counts
hours
0
2000
4000
6000
8000
10000
12000
14000
16000
0
5
10
15
20

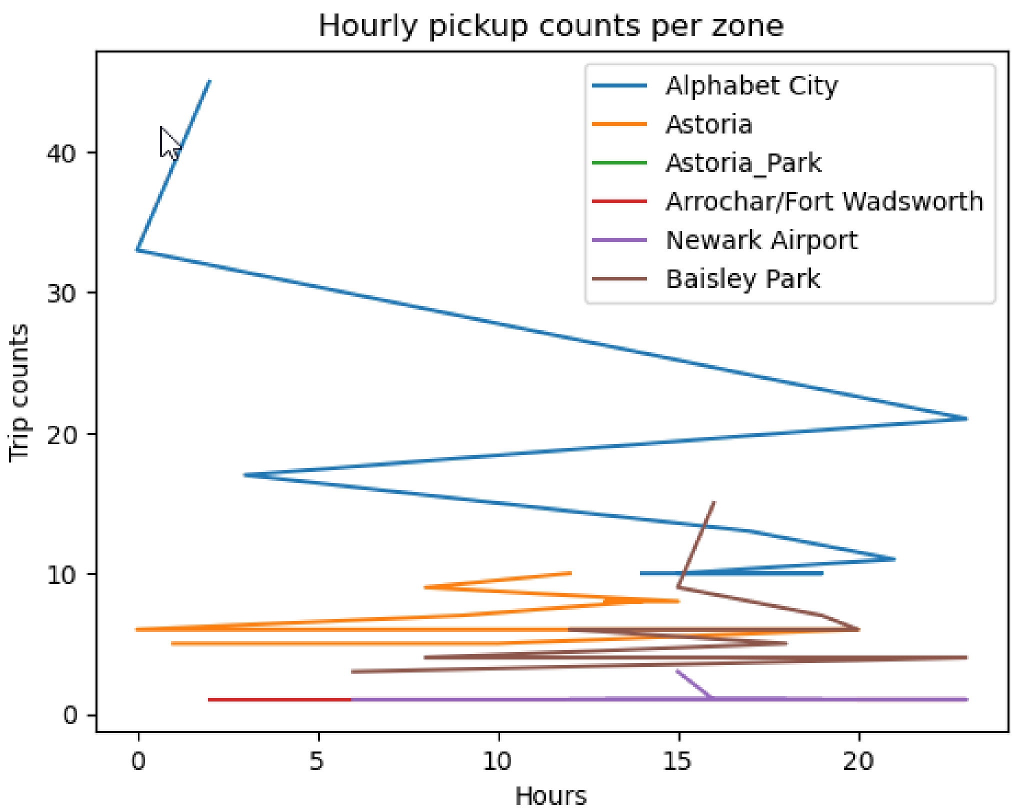

Hourly traffic Trend for weekend
Trip counts
hours
1000
2000
3000
4000
5000
0
5
10
15
20

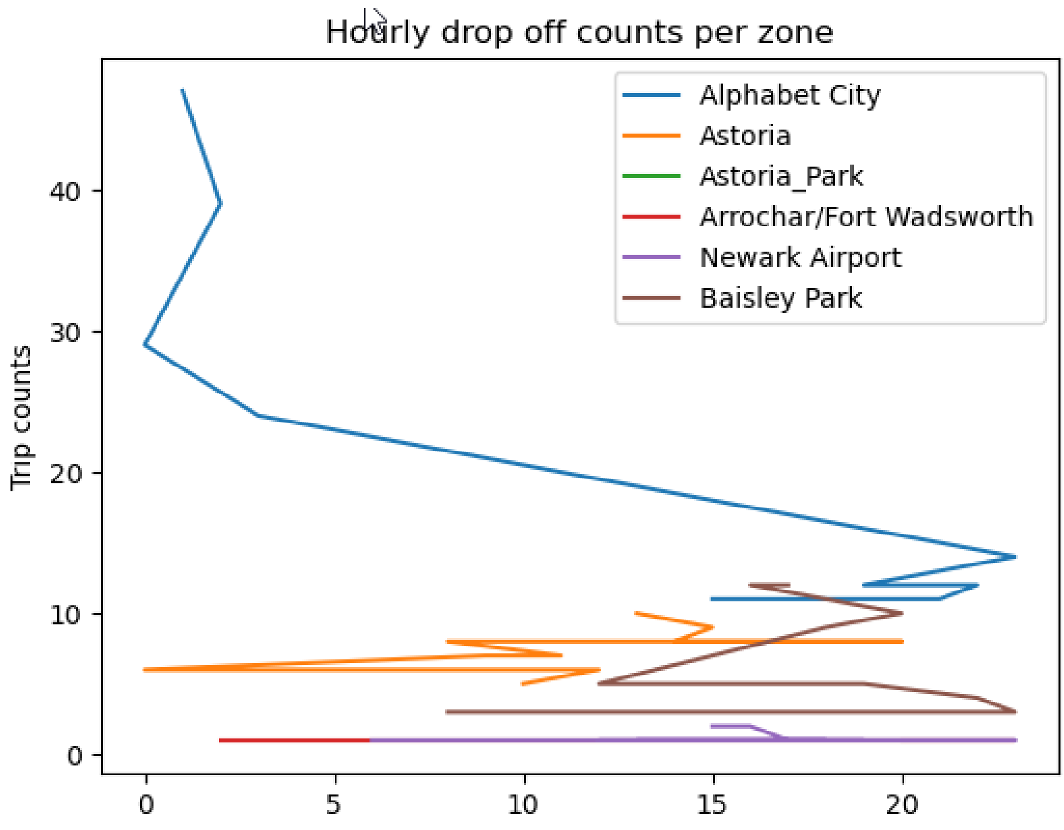
3.2.5. Identify the top 10 zones with high hourly pickups and drops

Since there are only only 10 ones and on merging the zone dataframe with original dataframe 5 zones are selected for merge with respect to location id in common.

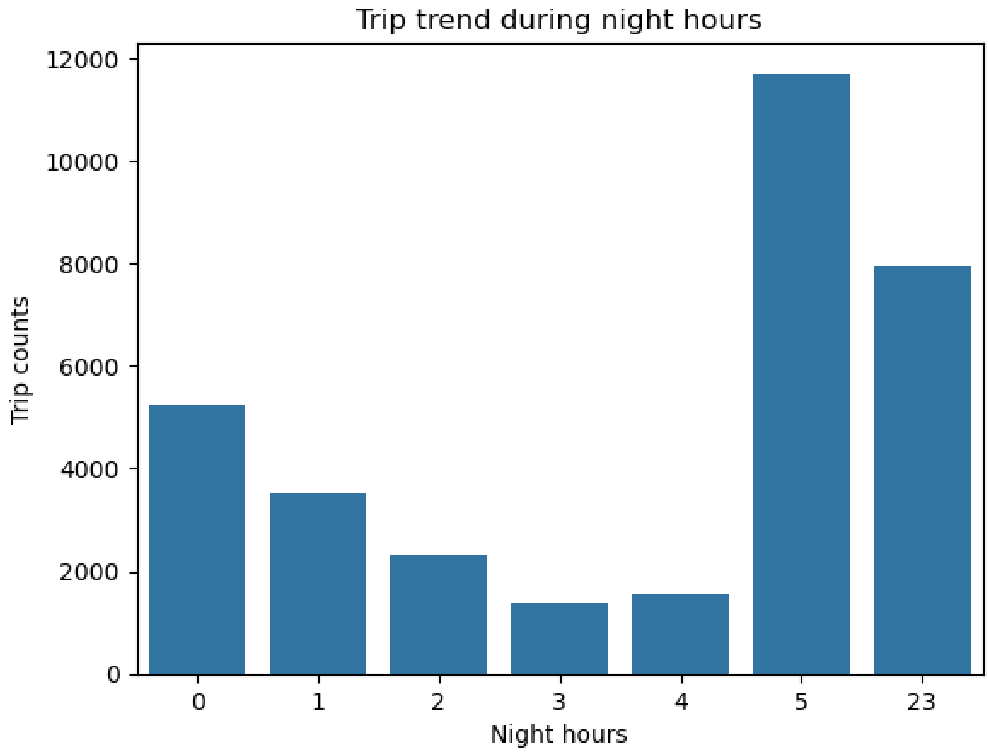
In the merged data frame high hourly pickups are in ‘Alphabet City’ with pickup count 250.

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

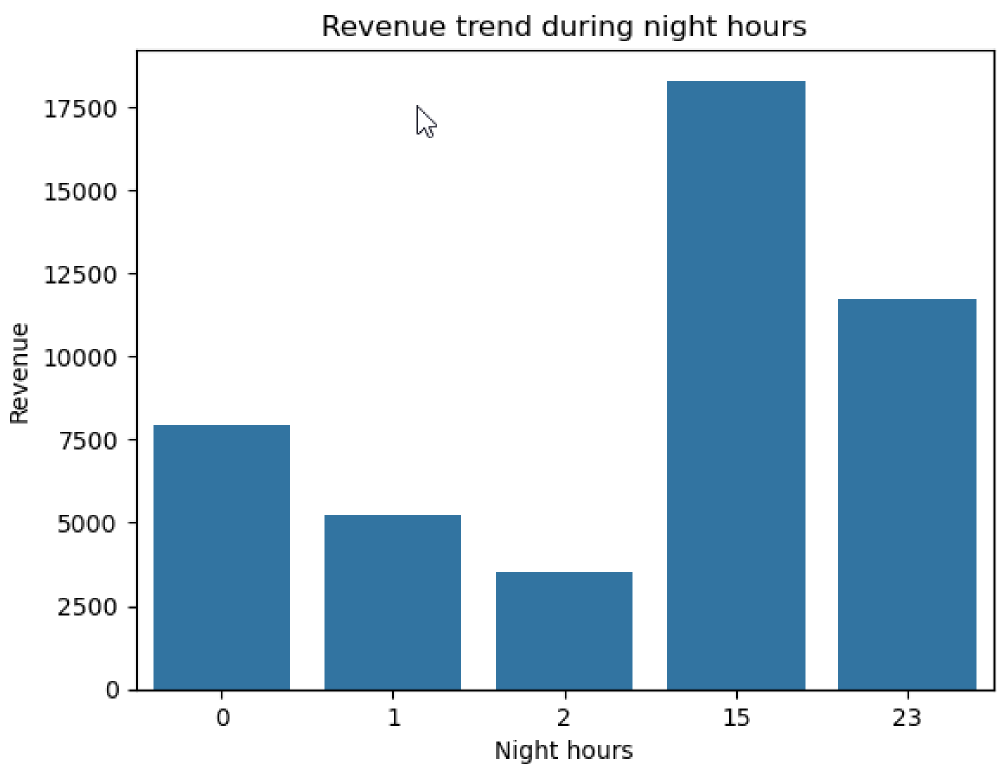



Hourly drop off counts per zone
Alphabet City
Astoria
Astoria_Park
Arrochar/Fort Wadsworth
Newark Airport
Baisley Park
Trip counts
0
10
20
30
40
0
5
10
15
20

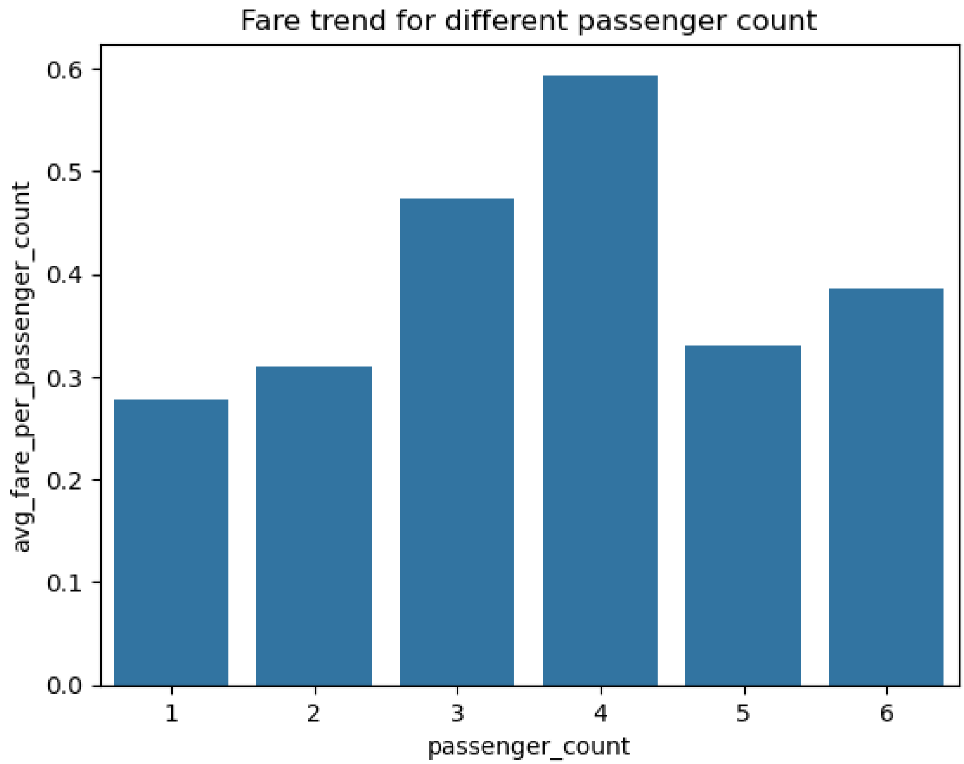
3.2.7. Identify the top zones with high traffic during night hours



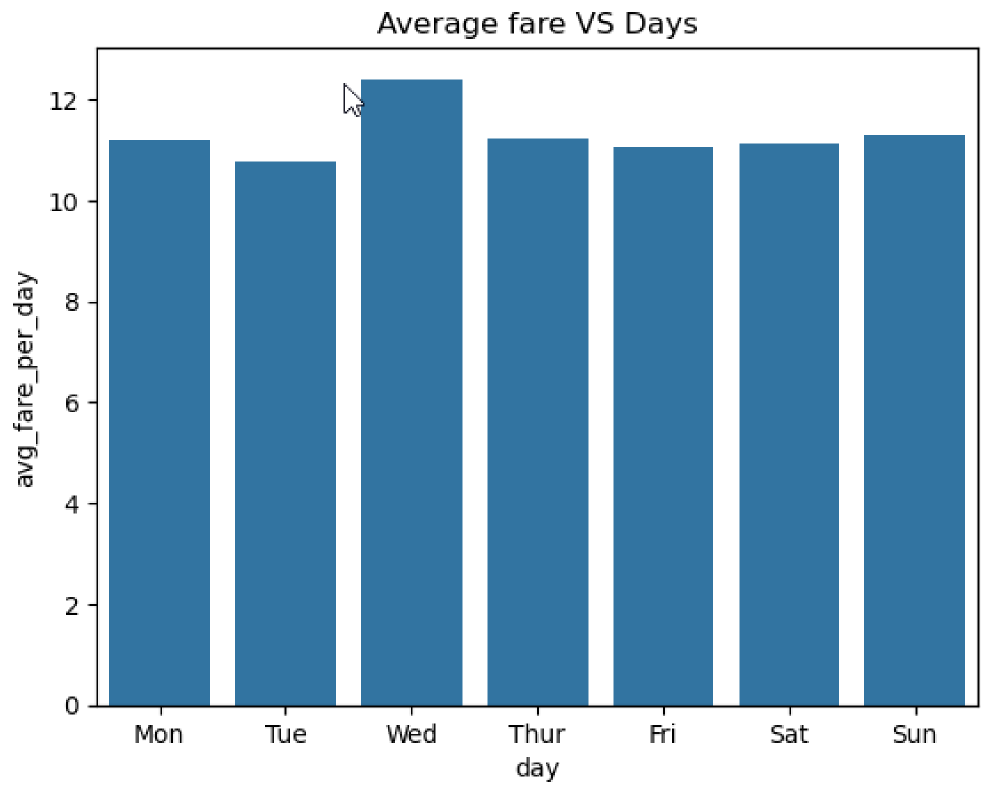
3.2.8. Find the revenue share for nighttime and daytime hours



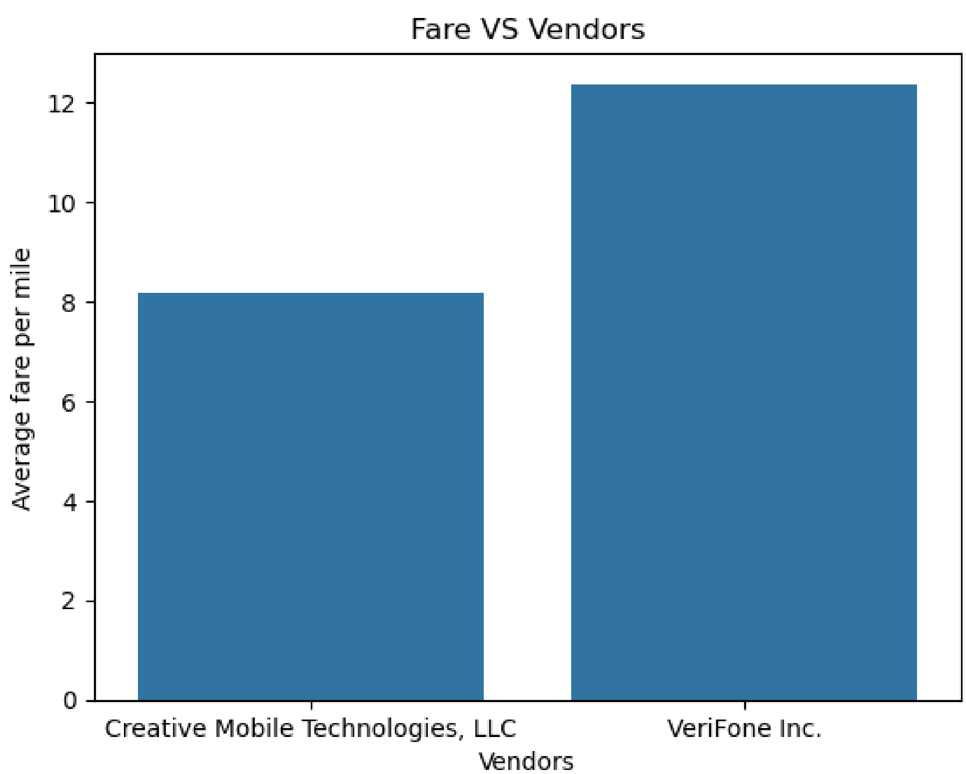
3.2.9. For the different passenger counts, find the average fare per mile per passenger



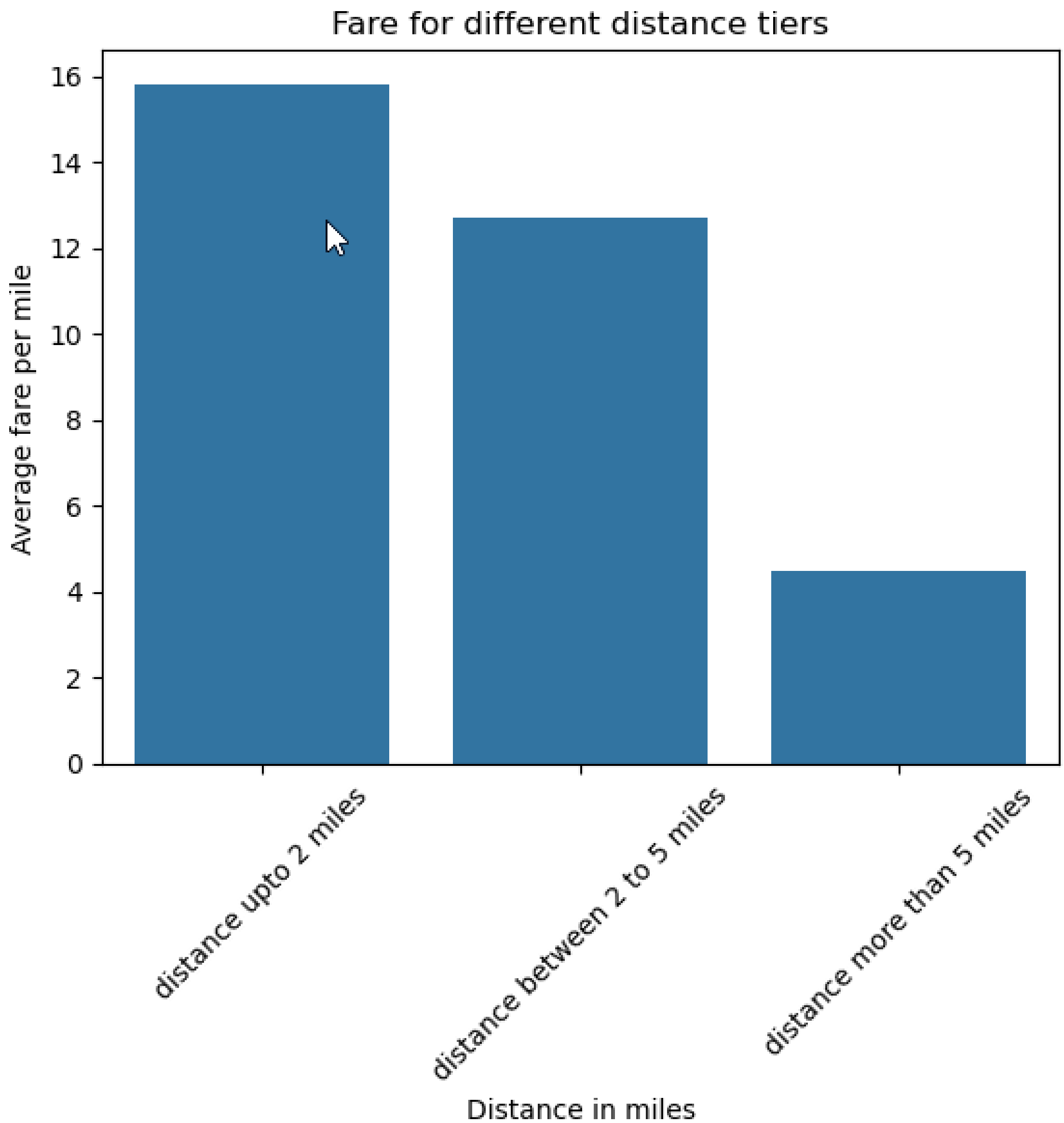
3.2.10. Find the average fare per mile by hours of the day and by days of the week



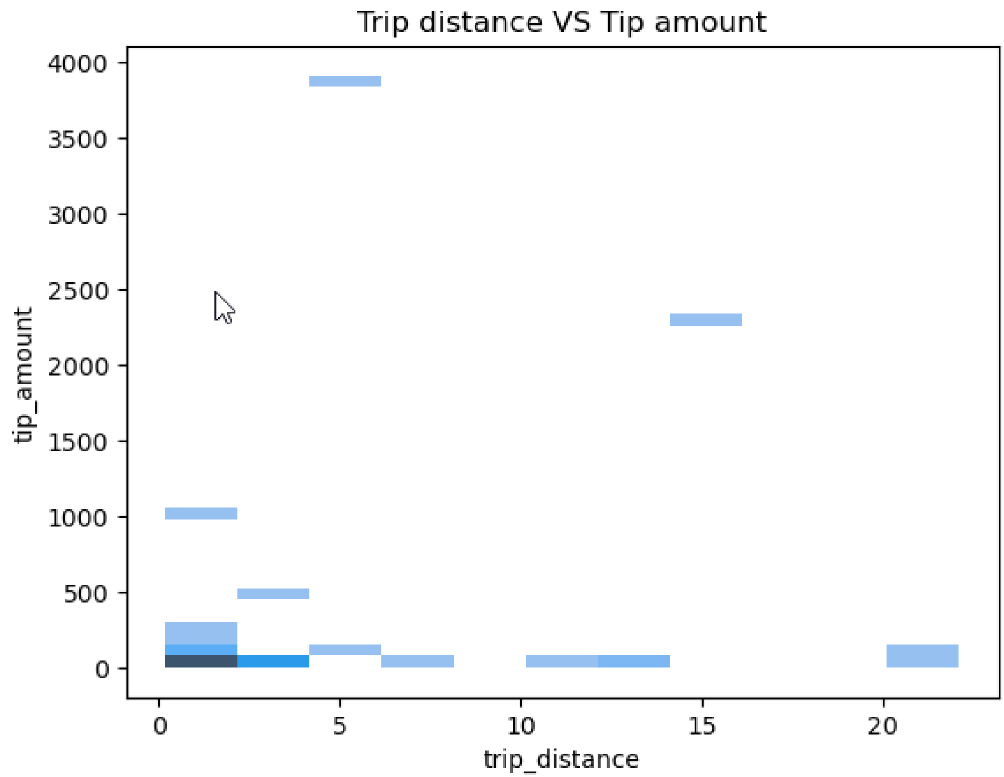
3.2.11. Analyze the average fare per mile for the different vendors



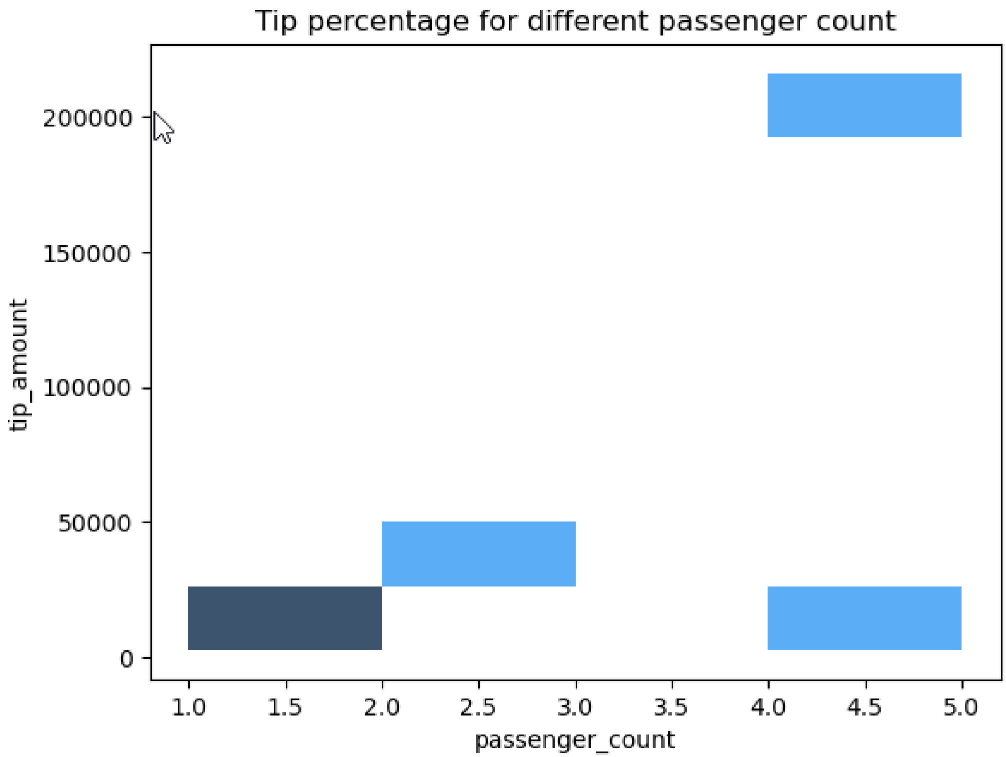
3.2.12. Compare the fare rates of different vendors in a distance-tiered fashion



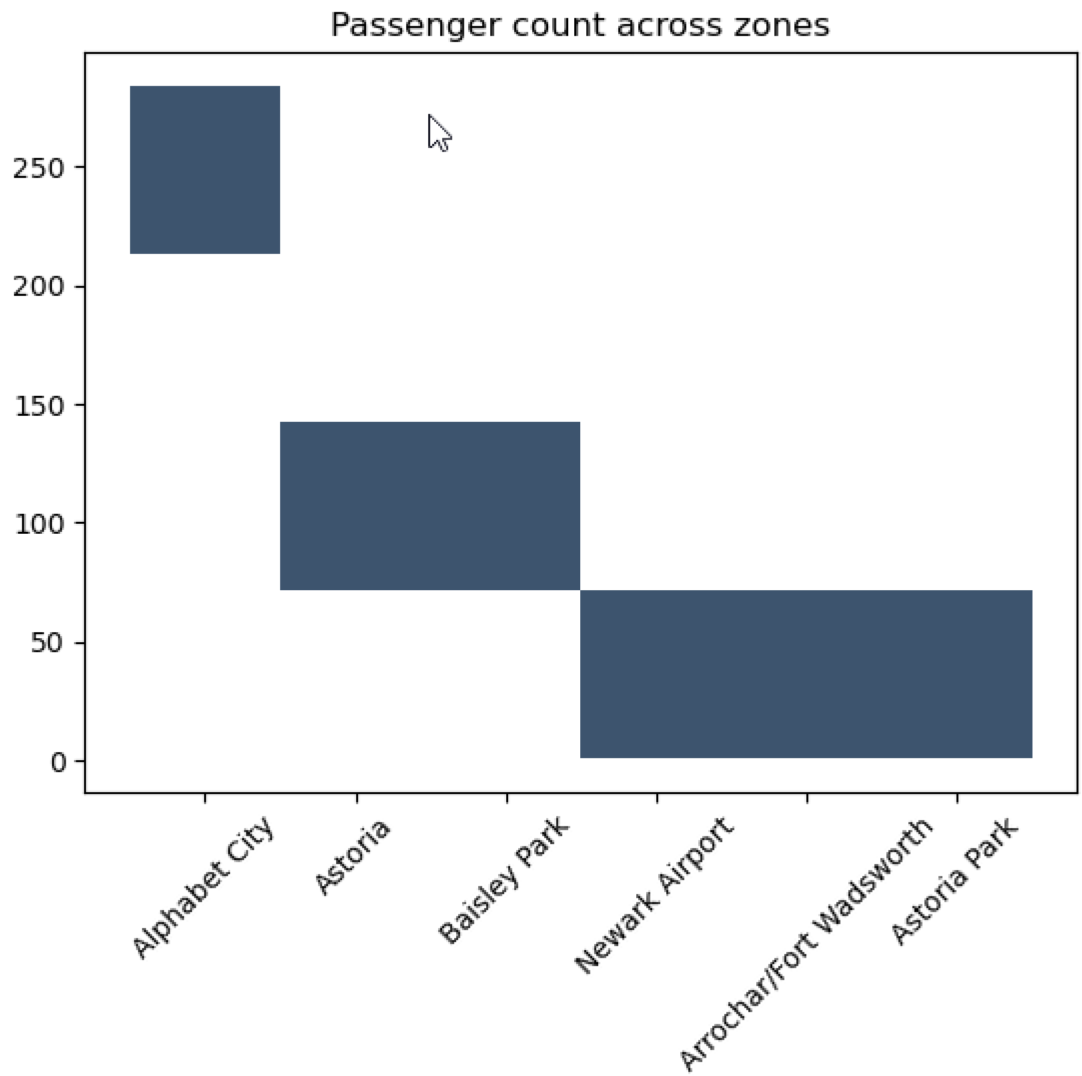
3.2.13. Analyze the tip percentages



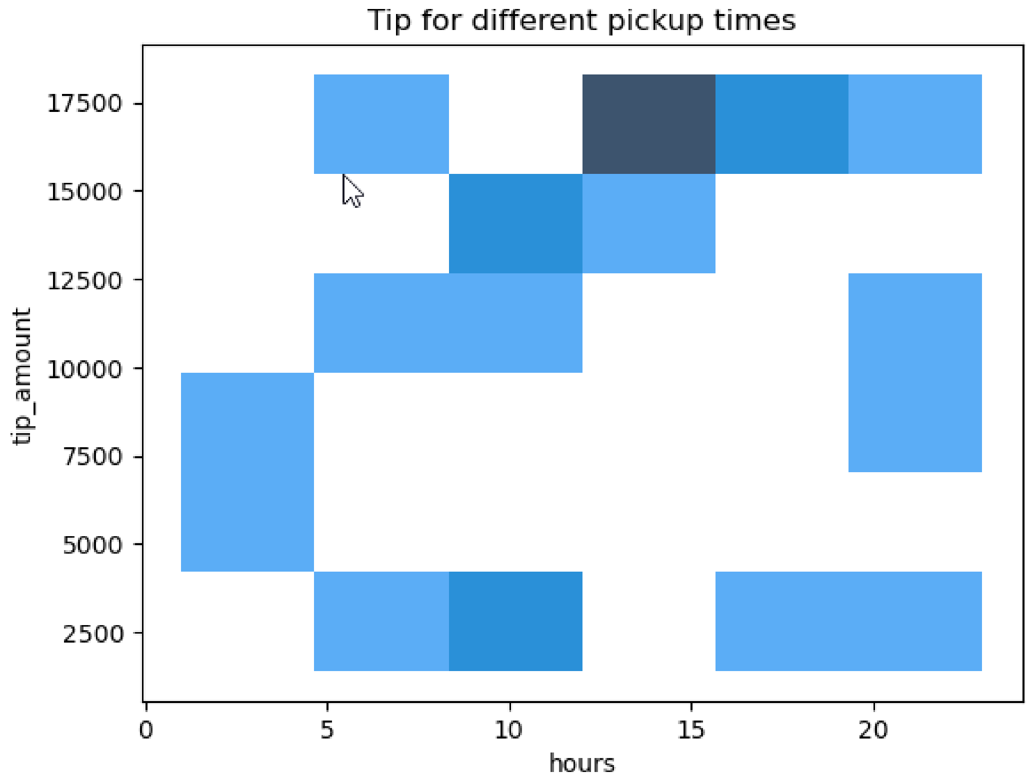
3.2.14. Analyze the trends in passenger count



3.2.15. Analyze the variation of passenger counts across zones



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



4. Conclusions

4.1. Final Insights and Recommendations

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

There are 4 factors to consider:

1. Fare amount with respect to distance
2. Days of the week
3. Time of the day
4. Month wise trend.

1. **Fare amount with respect to distance**: The correlation between distance and cost is 0.3. which means the cost increases only 30% with respect to distance. We can work on increasing the cost to match with the distance.
2. **Days of the week**: The highest bookings are on Wednesday and Thursday. The taxi bookings have a clear bell curve towards beginning and end of the week, which suggests to offer taxis at discounted price to promote more booking during weekends/week beginning,

3. **Time of the Day**: Highest bookings are at midday from 10AM to late evening around 6PM. Giving a hint the people book for office commute. These peek hours can be utilized to offer good service at a better price.

Month wise trend: As of 2023 Mexican population in NYC was about 30% (28.4%).

Mexicans celebrate All Saints Day which was on Nov1, Halloween on Oct 31 and All Souls' Day on Nov2, which contributed the hike in Oct, Nov and Dec which year end and Christmas.

Indian festivals like Dasara and Diwali also fall in the months of Oct-Nov, contributing to hike in taxi bookings.

May and Jun see spike because Memorial Day: Celebrated from May 25–31, Memorial Day is a floating Monday.

Basically other than office commute taxi booking is happening during the festive holidays, we can use these numbers to offer better service to encourage people to opt for more taxi bookings.

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

Zone wise analysis conclude that Alphabet City has outstood all other zones holding 284+ booking from merged data frame with count 404 which 70%, compared to other zones which are Astoria park 105, Jamaica Bay 11, and so on. We have study further to understand why there is huge gap and offer services tailored for those low booking zones.

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

There are only 2 vendors giving less room for competition. Customers have to adjust with the pricing given by either of one.
Creative Mobile Technologies, LLC has 72935 booking after all data clean up concluding to 72935 bookings which is just 25% of total booking
Verifone Incs has 215070 booking concluding to 75% total booking.
Clearly Verifone is winner and there is no healthy competition.