PREDICTING TAXI DEMAND IN NYC

Charlotte Williams

This paper aims to predict the demand for taxis for different neighbourhoods in New York City (NYC) using a combination of taxi trip and weather data. To model this data, a Poisson regression model will be used to predict taxi demand in NYC. The model will be trained and tested on data drawn from 2017-2019. The NYC Taxi and Limousine Commission (TLC) is the agency responsible for licensing and regulating New York City's Taxi cabs and for-hire vehicles and provides open access to trip data used for this task. Weather data was drawn from Weather Underground's historical data and was used to supplement trip data to provide contextual information that will be explored for predictive value.

BACKGROUND

The problem at hand is to predict taxi demand with respect to contexts of time, weather and location. The demand is thought to be the number of trips that a group of customers in a location would take given an excess of taxis in the area. Unmet demand is unable to be estimated as there is no customer sided reporting, only documentation of the trips that actually occurred. Therefore, real trip occurrence is used as a surrogate marker for taxi demand. This paper also explores the effect of weather on taxi demand. Kamga, Camille & Yazici, M. Anil & Singhal, Abhishek (2013) found there was significant variation in taxi ridership for different weather conditions.

The value of predicting demand is particularly relevant to all levels of stakeholder in the taxi industry. Customers want the ability for their demand to be predicted so to reduce their pre-trip wait time. Taxi drivers want to predict areas of high demand to inform their route while searching for trips. Taxi companies want to ensure that the amount of taxis in an area saturates but does not exceed demand in order to minimize idle taxis but still optimize trip frequency.

The ability to predict taxi demand may also provide valuable insights to parties outside the taxi industry; vitally, urban planners. Urban planners attempt to monitor and optimize transport in a city. One of the means by which they do this is to reduce road congestion. The taxi industry is a large source of commercial motor vehicle traffic and as such, the distribution of its activity across the city can help inform strategies to minimize congestion. Moving beyond identifying road congestion and the factors at play, strategies to reduce congestion include public transport systems that connect key hubs in a city. To identity areas where communal transport will be most beneficial to the public, one means may be by isolating areas in which commercial transport is in particularly high demand - where individuals either lack or have opted not to use private vehicles yet still require a means of transport.

The taxi industry is one strongly regulated by the Taxi and Limousine Commission. To operate as a taxi, each cab must have a medallion. Medallions are auctioned by the City and are transferable by licensed brokers. The medallions may be held by individuals or private companies. The yellow taxis cater to passengers from any of the 5 boroughs in NYC and are able to pick up street hail passengers. Green taxis differ in that they may not be street hailed and operate on an on-call basis only. There are 13,587 yellow taxis and 3500 green taxis operating in NYC. The Taxi and Limousine Commission has collected the data on taxi since 2009. This analysis is an examination of the data generated from the yellow medallion taxicabs alone and this decision will be discussed below.

Weather Underground is a global weather information provider. Weather underground has 15 weather stations across NYC. Data measured includes that on temperature, wind and precipitation. Data for this project was collected from the LaGuardia weather station as the data best aligned with this project's aims as will be discussed later.

INTRODUCTION

In addressing the factors affecting trip demand, factors that informed on the nature of the trip's context were chosen. For this reason, the location, time and weather were deemed most predominant and most unilaterally accessible.

This analysis looks at the data generated by the yellow taxicabs alone for the task of predicting taxi demand. Green taxis are not able to be street hailed and are only ordered by the customer. This means that the taxis will only be picking up customers in an area if they are booked in advance. Hence, where the taxis are is less at the discretion of taxi companies or drivers and more so, in the hands of the customer. Intuitively, the demand therefore will always be perfectly saturated on a neighbourhood level by taxi supply. Rideshare data was also discarded as rideshare operates in a similar manner of online request. Whilst it differs slightly and there is room for predicting demand, the dramatic difference in operation presents a different problem than that being considered here.

A significant component of the problem addressed here is the capacity to make predictions of future circumstances using historical data. As a result, 2017 and 2018 were used as training data to predict 2019's trip demand. Whilst the predictive capacity of historical data is being evaluated, it is recognized that context does play a role and that increasingly distant data decreases in relevance. It was in this vein that earlier data than 2017 was not used.
Both 2017 and 2018 were used to predict 2019 as it was decided that the predictive model would benefit from seeing the seasonal cycles over two years to better determine broader temporal patterns.

Data from 2020 is currently available from the Taxi and Limousine Commission. However, despite being most recent it was not included in this analysis on two grounds. Most obvious is that it does not yet provide an entire year's data and therefore would skew temporal relationships within the data. Further, the data from 2020 was determined to be unreliable in analysis completed for Assignment 1 (An Analysis of the Effects of Covid-19 on the Taxi Industry, 2020). The analysis found that data generated from 2020 was dramatically affected by contextual events surrounding the COVID-19 pandemic. As that is an isolated contextual event, it was deemed to have little ostensible valuable in predicting events where the pandemic was not relevant context.

Weather data was chosen to supplement the taxi trip data as it contained the most fluctuant yet relevant data. Considering this analysis aims to examine trip demand with high temporal fidelity, weather data was selected as a supplement data that both varied on a similar time scale and was documented to an appropriate temporal fidelity.
A variety of weather measures were available from Weather Underground. Of these, temperature, wind and precipitation were selected as they were thought to be represent weather conditions that presented the greatest barrier to the taxi consumer. These metrics were thought to correlate most closely with overt classifications of weather, i.e. cold/hot days, windy conditions and rainfall. Other measures such as humidity and dewpoint were thought to correspond less with overt weather and instead more subtly represent conditions. LaGuardia Airport was selected as a centralized location that best represented weather conditions across all neighbourhoods of NYC.

The problem being considered by this paper is one of spatial optimization – where should the taxi car be? The answer to this has already had its relevance outlined to the various stakeholders in the taxi industry.
Taxi demand was first approached by grouping trip data on a per-location basis. This location was determined by the pick-up zone of each ride. While customers solely control the drop off location, they vote en masse as to where is best or most profitable for a taxi driver to be and therefore trip data needed to be aggregated across the entire pickup zone. Consequently too, drop off location was determined to be a less relevant feature for the prediction as it was subject to individuals and not a trend that would predict pickup activity. To quantify taxi demand, this exploration predicts the number of trips expected for a given hour on a given day in a given neighbourhood.

Trips per hour represents count data measuring the number of rides within a time interval of an hour for a given day for a given neighbourhood. The Poisson distribution is a distribution of how many times an event is likely to occur in a given time period. For predicting Trip demand, it is how many trips occur in an hour period for a given hour, of a given day. Since trips per hour is both discrete and non-negative, a linear model which assumes that the response variable is continuous, and an element of the real numbers is deemed inappropriate. Therefore, a Poisson regression model was chosen for the task.

DATA

The taxi data was collected from the Taxi and Limousine commission (TLC) of New York City (NYC). The data contains approximately 300 million taxi rides from the yellow cab taxis. This data is from 2017, 2018 and 2019. The data from 2017 and 2018 was combined to train the model and 2019 was used as testing. Each dataset from each year contained the same recorded attributes for each trip. Of these the following attributes were used in the predictive modelling:

> ***Payment_type:*** *A numeric code signifying how the passenger paid for the trip;*
> ***Passenger_count:*** *The number of passengers in the taxi;*
> ***Tpep_pickup_datetime:*** *The date and time when the meter was engaged;*
> ***PULocationID:*** *TLC Taxi Zone in which the taximeter was engaged;*
> ***Fare_amount:*** *The time and distance fare calculated by the meter;*
> ***Trip_distance:*** *The trip distance travelled in miles reported by the taxi meter*
> ***Tip_amount:*** *The amount the customer tipped the driver, this only includes tips if the customer paid by credit card.*

In order to assess the impact of weather on the trip demand in NYC, data was needed on the daily weather conditions from 2017, 2018 and 2019. This data was available from Weather Underground. This data consists of daily measures of temperature, dewpoint, humidity, windspeed, pressure and precipitation. As discussed above, of the available attributes, the below were selected and used in this analysis:

> ***Average daily temperature:*** *The average temperature for each day, measured in Fahrenheit;*
> ***Average daily wind speed:*** *The average wind speed for each day, measured in miles per hour;*
> ***Daily precipitation:*** *The daily recorded precipitation measured in inches.*

METHOD

Pre-processing
Before any pre-processing, the 2017 taxi dataset contained 113,496,874 instances, the 2018 taxi dataset contained 102,804,250 instances and the 2019 taxi dataset contained 84,399,019 instances.

An initial investigation of the taxi data revealed abnormal data entries. Table 1 shows the summary statistics of 2019 as an example. These statistics indicate the dataset contained both significant outliers in addition to otherwise invalid values.

Attribute	count	mean	std	min	25%	50%	75%	max
passenger_count	113496874	1.62528476	1.26513412	0	1	1	2	192
trip_distance	113496874	2.92865986	4.07391538	0	0.99	1.62	3.04	9496.98
fare_amount	113496874	13.047984	190.360385	-550	6.5	9.5	14.5	861604.49
tip_amount	113496874	1.83730726	2.62481736	-391	0	1.35	2.45	1600.22

TABLE 1: Summary statistics of 2017 dataset before pre-processing

These incorrect values were considered on an attribute by attribute basis. Fare amounts which were less than the $2.5 mandatory fee were removed. Passenger counts of less than 1 were considered erroneous and removed. Negative trip distances and negative tip amounts were both removed. Given the absence of recording for tips for cash payments, all trips paid in anything other than a credit card were removed. This is to ensure all trips had an accurate recording of the tip amount paid. Once the incorrect values were removed, the distribution of fare amount, tip amount and trip distance were visualized.

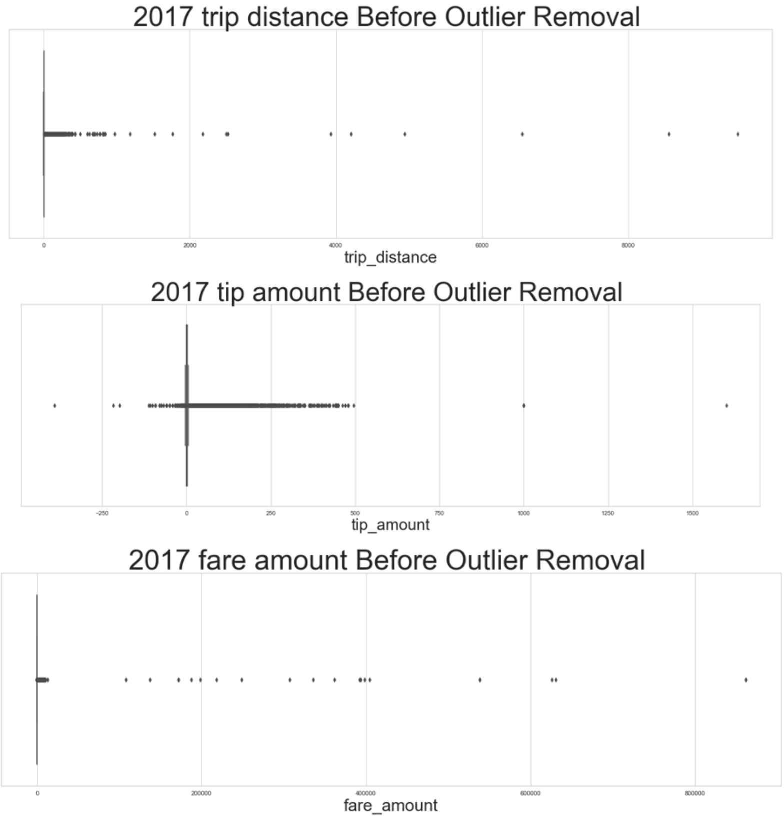
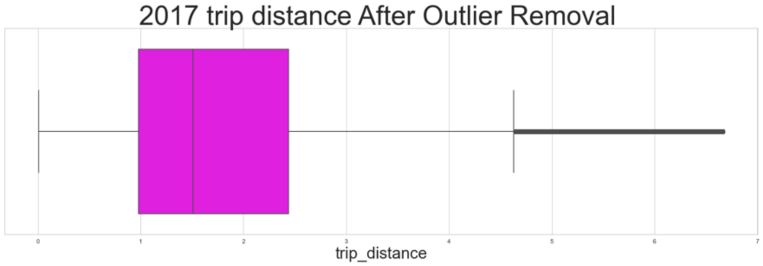


FIGURE 1 - Box plot of trip distance, tip amount and fare amount for 2017 before outlier removal

These visualisations reveal a large left skew of the data and show a large number of outliers. Outliers from each year of the taxi datasets. This is important for the predictive model as large outliers can have a large influence (leverage) on the fit of the model. This may result in less accurate predictions. In this analysis, an outlier was defined to be either greater than the third quantile by 1.5 times the interquartile range (IQR) or less than the first quantile by 1.5 times the IQR. The IQR is the difference between the first and the third quantiles.



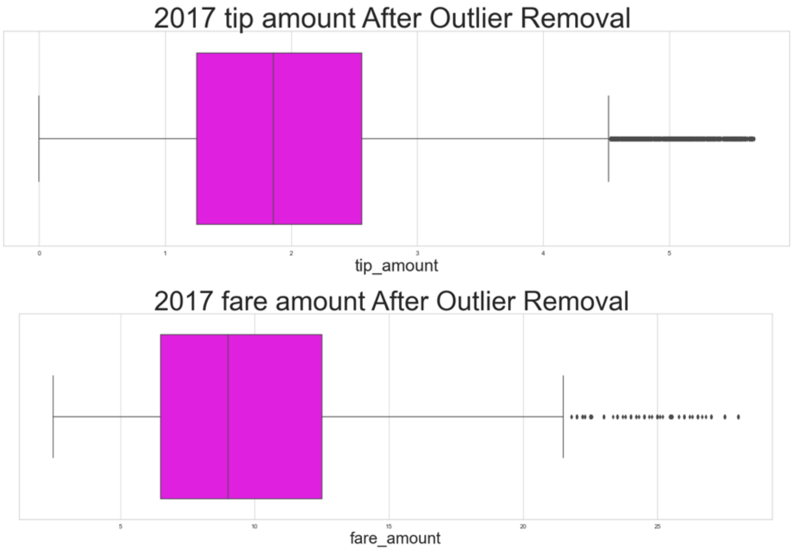


FIGURE 2 - Box plots of trip distance, tip amount and fare amount for 2017 after outlier removal.

Figure 2 shows a more even dispersion of the data with a much less skew.

Weather data was scrapped from Weather Underground's daily historical data recorded by the La Guardia weather station for 2017-2019. There were no missing values nor intuitively impossible instances. As such, no further pre-processing was performed.
The weather data was recorded daily. As a result, data could not be removed from this dataset without additionally removing thousands of taxi data instances which occur on the same day. Therefore, it was decided that no outlier removal was implemented on the weather datasets.

Feature Engineering
Following pre-processing, feature engineering was undertaken to produce the attributes of interest for the model. The first was to create the features pick-up month and pick-up hour. Pickup hour was found by extracting the hour the taxi ride fell into for each day using the pickup datetime attribute. Pick up month was found the same way, by extracting the month the taxi ride occurred in.

The response variable, expected trips for a given hour of a given day for a give n pickup location, was then was created and will be from here on out referred to as "trips per hour." This will be used to model trip demand. To engineer this variable each trip entry was grouped by location of pick up, the date of pick up and the hour of pick up. The trips that fell into each of these groups were counted and summed to produce the attribute trips per hour. The formation of this variable meant that fare amount, trip amount and trip distance became the average for each given pick up location on each day of a month for each hour. This was implemented for each data set and 2017 and 2018 were concatenated to form the training data while 2019 was kept separate as it is being used as the test data.

The produced training dataset contained 1,640,179 instances. The data for 2019 formed the test set and contained 806,767 instances.

Feature Selection
Predictor variables and the response variable, trips per hour, were then visualised against one another to identify any relationships that may exist.

The temporal dependency of the response variable was explored by plotting average trips per hour and the average trips per month.

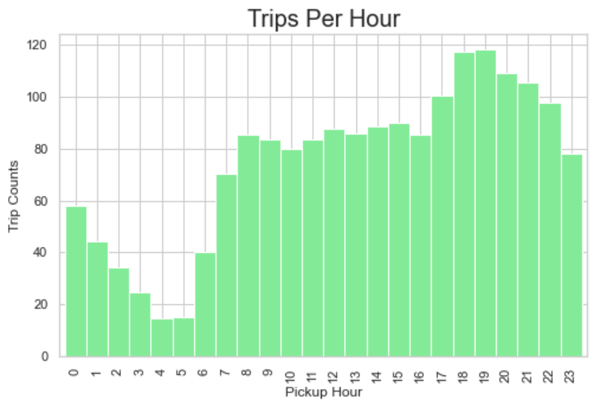


FIGURE 3: Trips per hour

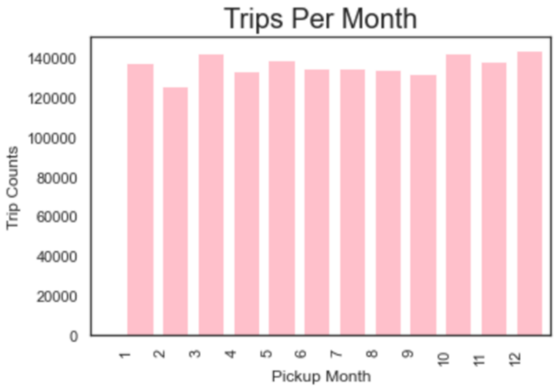


FIGURE 4: Trips per month

Figure 3 shows the average number of trips for each hour of the day. The number of trips peaks between 6-8pm every day and is at a minimum between 4-6am.

Figure 4 shows the differences in the number of trips per month through the years 2017 and 2018. It can be observed that there isn't too much variation of the total amount of trips each month however there is a slight increase in trips over the months November - January and a smaller number of trips over the months June - September.

The relationships between average fare amount, average tip amount and average trip distance and the response variable are represented in Figures 5, 6 and 7 below.

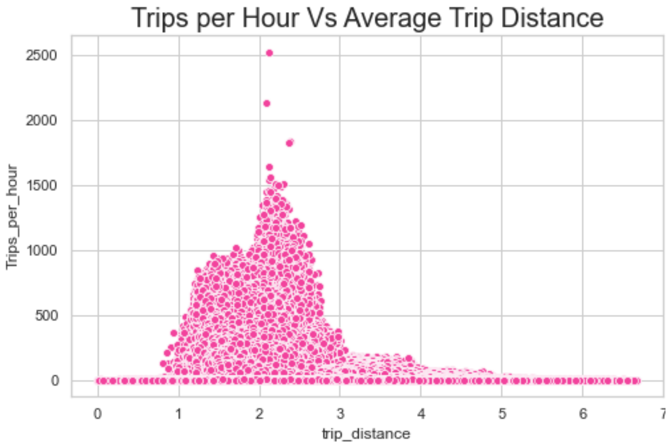


FIGURE 5: Trips per hour for different values of average trip distance

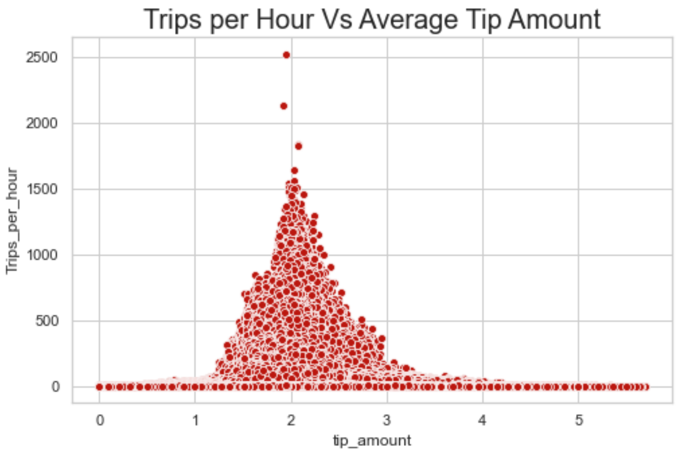


FIGURE 6: Trips per hour for different values of average tip amount

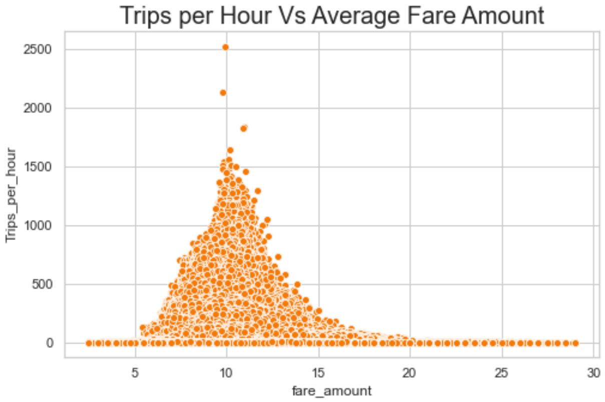


FIGURE 7: Trips per hour for different values of average fare amount

Figures 5, 6 and 7 have a similar shape – resembling a rough normal distribution with a peak in trips per hour occurring around the mean values of each attribute. This indicates that average trip distance, fare amount and tip amount all have a similar relationship with trips per hour. For larger tip and fare amount and for longer trip distances there are less trips. The strong relationship between these features and trips per hour suggests predictive potential of these attributes.

Finally, the relationship between weather and trips per hour is observed in Figures 8, 9 and 10.

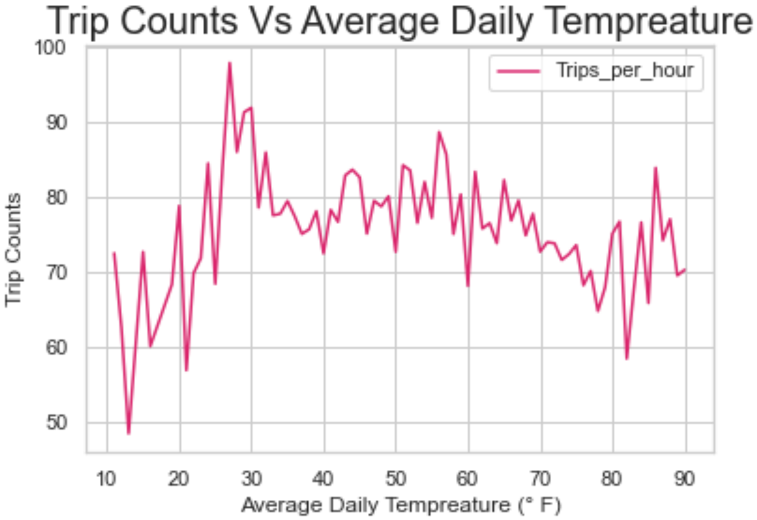


FIGURE 8: Average Daily Temperature vs Trips per Hour.

Figure 8 shows the relationship between trips per hour and average daily temperature. For temperatures between 25-35 degrees Fahrenheit (-4 to 2 degrees Celsius), there is a peak of trips per hour. Between 40-80 degrees Fahrenheit (5 to 26 degrees Celsius) there is a downward trend before increasing again 80-90 degrees Fahrenheit (26 to 33 degrees Celsius). This graph suggests a trimodal relationship between temperature and trip count that may indicate significant predictive value.

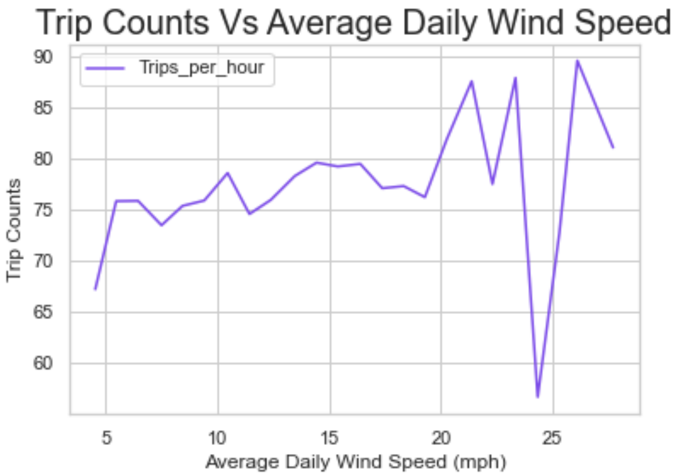


FIGURE 9: Average Daily Wind Speed vs Trips per Hour.

Figure 9 shows the relationship between trips per hour and average daily wind speed. There is a consistent upwards trend in trips for high values of wind speed. At approximately 24 mph there is the global minimum. This minimum could be explained by a lack of data for higher wind speeds and therefore increasing the variability of the trips counts for these wind speeds. From this plot it can be assumed that with higher wind speeds there are more trips per hour.

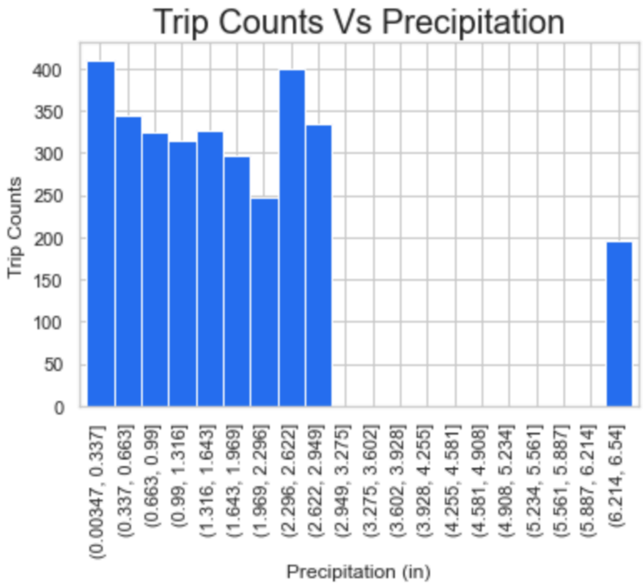


FIGURE 10: Daily Precipitation vs Trips per Hour.

Figure 10 shows the average trips per hour for different precipitation levels. There is a decrease in trips between 0.337 and 2.296 inches of precipitation but a spike once it reaches the 2.296-2.622 inches bin. There is no observed data for any precipitation levels between the quantities 2.929 and 6.214 inches of precipitation. There are no strong trends revealed in this visualisation and questions the efficacy of rainfall as a predictor for trips per hour.

Given these relationships with the response variable, the attributes used in predicting trips per hour are fare amount, tip amount, trip distance, average daily temperature, average daily wind speed, daily precipitation, pick up month, pick up hour and pick up location. Pick up month, pick up hour and pick up location are treated as categorical variables and the rest as continuous. Table 2 shows the summary statistics for the continuous attributes.

Attribute	count	mean	std	min	25%	50%	75%	max
Trip_distance	1640179	2.29793275	0.9853691	0.01	1.67689655	2.05862069	2.68571429	6.67
Fare_amount	1640179	10.5901488	3.12882531	2.5	8.8125	10	11.75	29
Tip_amount	1640179	1.92487201	0.78081695	0	1.677	1.9675	2.24705542	5.7
Trips_per_hour	1640179	76.3805749	119.823191	1	3	15	110	2523
Tempreature	1640179	56.3239122	17.3078373	11	42	56	72	90
Wind_speed	1640179	10.9163139	3.95614525	4	8.1	10.2	12.9	27.7
Precipitation	1640179	0.14591977	0.39923565	0	0	0	0.08	6.54

TABLE 2: Summary statistics for training data

DISCUSSION

To model trips per hour, a Poisson distribution will be used, with probability density function:

$$f(x,\lambda) = P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Equation 1: Pdf of Poisson distribution

Before fitting the model, the model's assumptions had to be checked to deem it appropriate for the task. The first assumption of the Poisson model is that each response variable must be a non-negative integer. This assumption is satisfied as trips per hour is a count of the number of trips in a certain neighbourhood for each hour for each day. The next assumption is that the distribution of these counts follows a Poisson distribution.

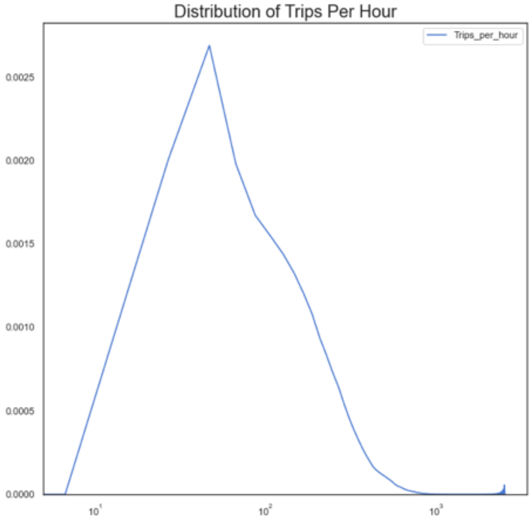


FIGURE 11: Distribution of Trips per hour

Figure 11 shows the distribution of trips per hour. A log scale has been used on the x-axis to visualise the distribution better. It exhibits a rough bell curve shape with a sharper maximum which is indicative of the Poisson distribution. It can therefore be said that this data doesn't show any violations of the Poisson distribution and can fit a Poisson distribution model.

The next assumption which must be satisfied is the independence of the predictors. To determine this Figure 12 shows a pairwise plot of the continuous attributes.

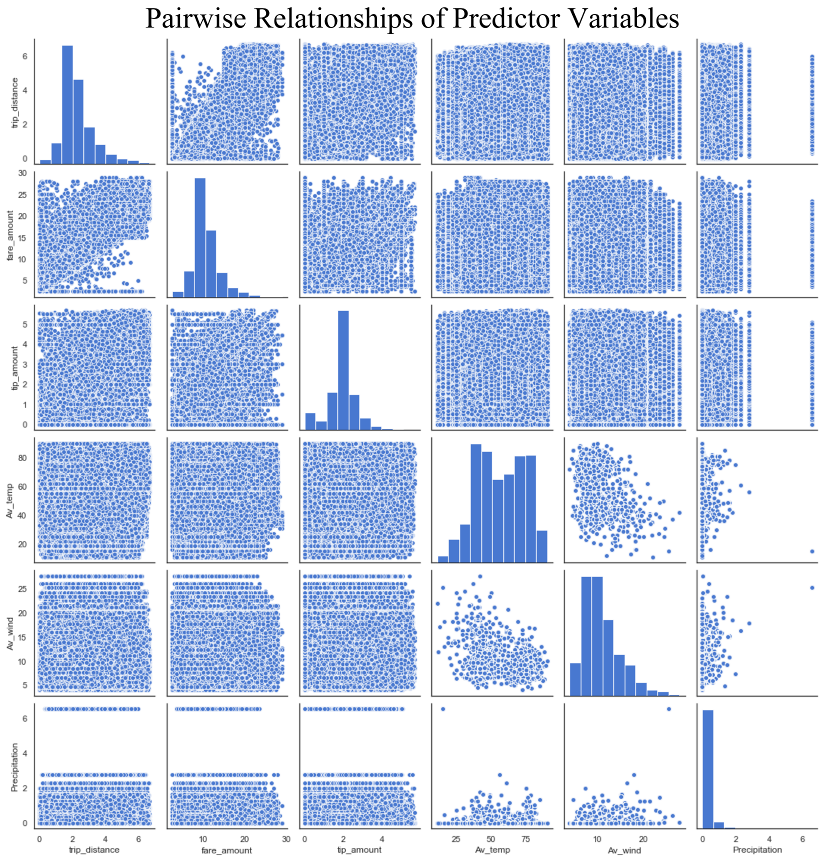


FIGURE 12: Pairwise plot of Predictor variables

Figure 12 highlights a linear relationship between trip distance and fare amount. All other predictors do not present any overt linear relationship with each other. To investigate this further, Figure 13 shows a correlation heatmap of all the predictor variables.

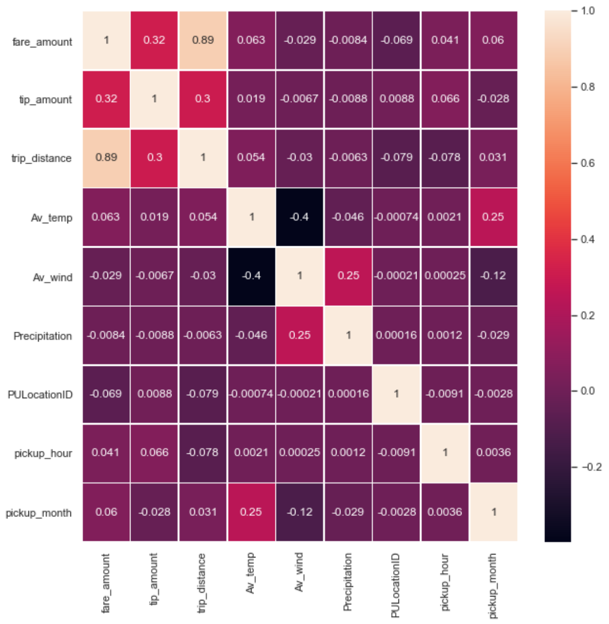


FIGURE 13: Correlation Heatmap of Predictor Variables

The correlation heatmap shows a 0.89 correlation between fare amount and trip distance indicating the two attributes are highly correlated. Therefore, trip distance will be removed as a possible predictor from the model to adhere to the model assumptions. All other variables have correlation coefficients close to 0 except for the pairing of fare amount and tip amount and wind speed and temperature. Since no noticeable relationships between these variables could be observed in the pairwise plot, they were deemed to be adequately unrelated and kept in the model as predictors.

The subsequent assumption to be checked is whether the instances are independent. In this data set, the instances are taxi rides, which are intuitively independent from one another. Therefore, the data satisfies this assumption.

The final assumption for a Poisson model is that the mean and variance are equal. The mean value of trips per hour is 76.38 and the variance is 14357.57. Evidently these are not equal. In an attempt to accommodate this, the log of trips per hour was taken and this minimised the difference between the two. However, this resulted in the distribution of trip per hour to no longer follow the shape of a Poisson distribution and it was thought this trade off in assumptions was not acceptable. Assumptions of other models such as a linear model have discussed previously, and it has been discussed that this data fails a greater proportion of those. Therefore while the data does conflict with this assumption it was decided that it fails on more assumptions of competing models and so the data will better fit to a Poisson distribution.

The data was fitted using a Poisson regression model.

Trips per hour ~ fare amount + tip amount + average temperature + wind speed + pickup hour + pickup month + pick up location

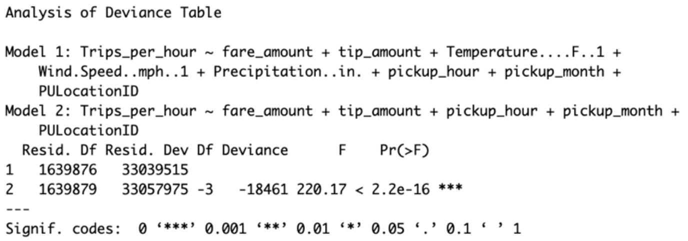
Initially this model is fitted with 303 prediction variables as pickup hour, pickup month and pickup location are treated as categorical variables. The AIC value is 40,718,939. The residual deviance is 33,039,515 this isn't too bad considering there is 1,639,876 degrees freedom. The R squared value a goodness of fit measure. It measures the percentage of the variance for the response variable that is explained by the predictor variables collectively.

$$R^2 = 1 - \frac{residual\ deviance}{null\ deviance}$$

FORMULA 2: R Squared Formula

The initial fitting of the model resulted in an R squared of 0.8577. This indicates that approximately 85.77% of trip per hour variation can be explained by the predictor variables. This emphasises that the model is a appropriately fits to the data.

The next step was to quantify the predictive value of the weather. To do this an ANOVA F-test was implemented to compare the models with and without weather. Figure 14 shows the output from the ANOVA function.



```
Analysis of Deviance Table

Model 1: Trips_per_hour ~ fare_amount + tip_amount + Temperature....F..1 +
    Wind.Speed..mph..1 + Precipitation..in. + pickup_hour + pickup_month +
    PULocationID
Model 2: Trips_per_hour ~ fare_amount + tip_amount + pickup_hour + pickup_month +
    PULocationID
  Resid. Df Resid. Dev Df Deviance      F    Pr(>F)
1   1639876   33039515
2   1639879   33057975 -3   -18461 220.17 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

FIGURE 14: ANOVA Output.

The ANOVA function shows a very small p-value (< 0.001), indicating significance. Therefore, weather should not be removed from the model.

To optimise this model further, stepwise selection using AIC was implemented to choose the variables which were the best predictors. However, after implementing stepwise selection, no attribute was removed. Removing any attribute increased the AIC value and therefore the model could not be improved any further by removing any predictor variables.

Since the variance of the response variable is larger than the mean, the overdispersion of the model was estimated. This is because a non-constant rate can be a cause of overdispersion. Overdispersion is what occurs when the variability of the data is more than expected under the assumed distribution. For the Poisson model, the dispersion parameter incorporated into the model was set at 1. Equation 3 shows how to estimate the dispersion parameter.

$$\phi = \frac{\sum residuals^2}{degrees\ freedom}$$

EQUATION 3: Dispersion Parameter

Using this equation, the true dispersion parameter was estimated to be 27.95, over 27 times than it was built into the Poisson model. Therefore, it can confidently be assumed that the data is over dispersed and a quasi-Poisson model would be a better fit of the data.

The quasi-Poisson model results in the same R squared value of 0.8577 as the Poisson model. However, it is still deemed a better fit to the data as it fits the model assumptions better than the Poisson model.

The quasi-Poisson model was then used to predict the test data. Figure 15 shows the predicted verses the actual values of trips per hour.

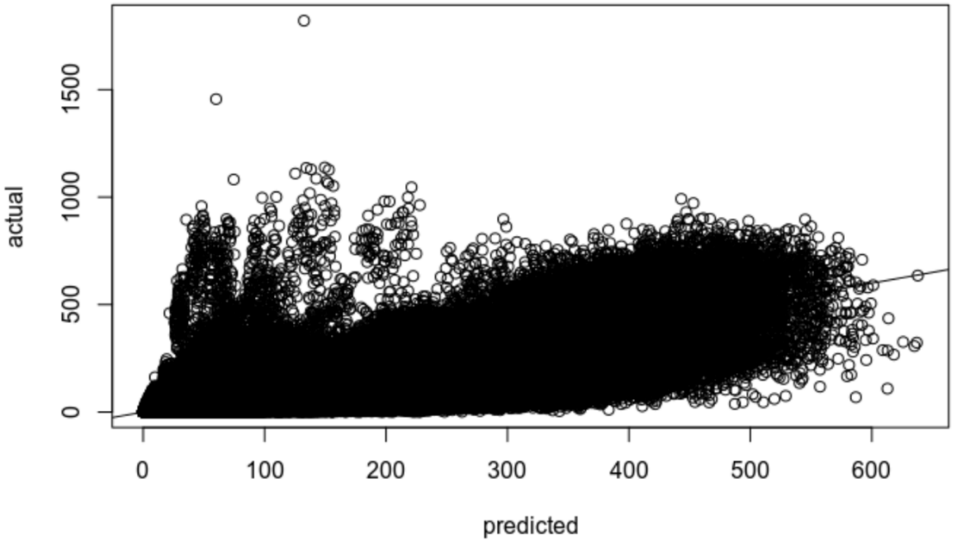


FIGURE 15: Predicted Vs Actual Values

Figure 15 shows that the predictions are following the regression line. There are more incorrect predictions for actual values of 500-1000 trips per hour. However, the model performs optimally on actual values of trips per hour between 0-500. This suggests that the model struggles particularly in predicting the high trip densities. This is going to be a very large source of error as not only did the model incorrectly predict more of these values, but it guessed smaller numbers for them, approximately between 100 – 200. This plot also shows a constant variation of predictions around the regression line. This reflects that the residuals are homoscedastic. It suggests that the regression coefficients are reliable measures of the relationship between the predictors and the response and the statistical significance of the predictor variables can be trusted. This plot demonstrates that the model is working very well and is well suited to the task.

To evaluate the model root, mean square error (RMSE) and mean absolute error (MAE) have been used. RMSE gives a larger penalty to larger prediction errors while MAE treats all errors as the same. The formulas to calculate these metrics are given in equations 3 and 4.

$$RMSE = \sqrt{\frac{\sum(predicted - acutal)^2}{N}} \quad MAE = \frac{\sum|predcited - actual|}{N}$$

EQUATION 3 and 4: Root Mean Squared Error (left) and Mean Absolute Error (right)

DATA	RMSE	MAE
Train	57.8	28.4
Test	54.5	27.3

TABLE 3: RMSE and MAE values for Train and Test

The RMSE and MAE for the training and tests sets are given in Table 3.

For both metrics, the testing and training values of RMSE and MAE are very close to each other. This indicates that the model is not overfitting much at all. An interesting point to note is that the train RMSE is larger than the test RMSE. An explanation for this is that since the training dataset is larger there is more room for error in predictions. The model is generalising very well for the testing RMSE to be less than the training RMSE. Since the MAE of the train set is larger than the MAE of the test set however, the errors of the training set prediction must be slightly larger than the test set predictions and hence being penalised more by the RMSE.

Figure 16 is a plot of the model parameters. Since there are over 303 independent variables used in the model, not all of attributes were plotted. Instead Figure 16 is a plot of all coefficients except the pickup location, whilst Figure 17 shows the largest 50 coefficients overall. Notably the largest 50 coefficients are all pickup locations.

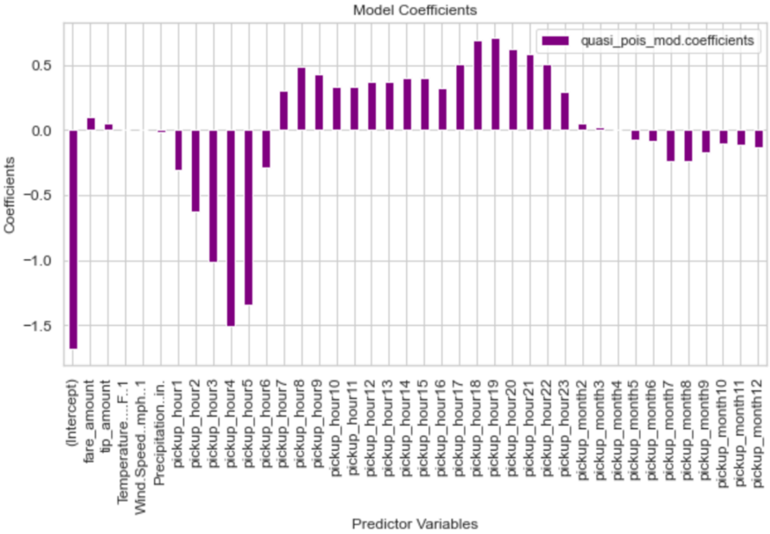


FIGURE 16: Plot of All Non-Location Model Parameters

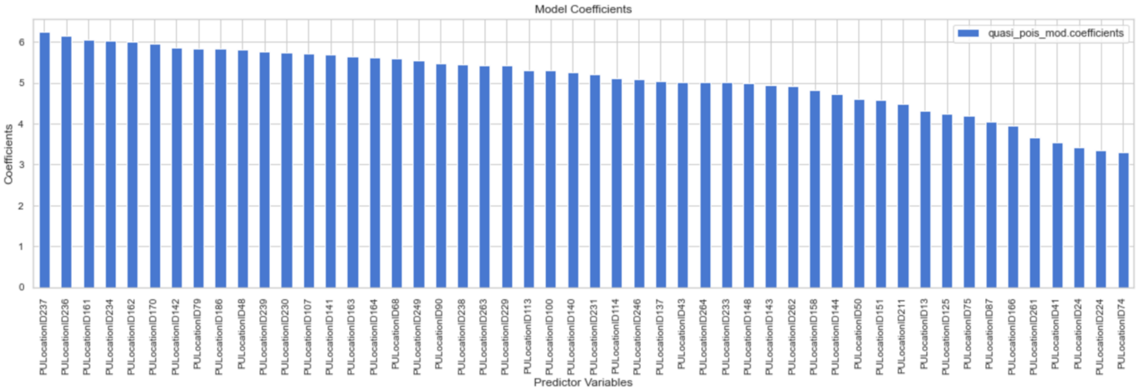


FIGURE 17: Plot of Largest 50 Model Parameters.

The model coefficients are used to describe the relationship between the response variable trips per hour and the predictor variables. The coefficients value represents the mean change in the response variable given a one unit increase in the predictor variable. The size of the parameter does not necessarily correlate to the importance of the variable in the prediction. Figure 16 shows that the weather variables have parameters very close to 0, suggesting change in the weather, either wind speed, temperature or precipitation does not largely affect trips per hour. This is interesting to observe as weather was not removed in stepwise selection. Also, when comparing the two nested models of 'with-' and 'without weather' the p-value returned identified all weather as very significant in the prediction of trips per hour. While weather might help the model predict, this visualisation highlights that the change in weather does not have a large effect on taxi demand.

Given the performance of the Poisson regression model, there is suggestion that high rates did not conform to the Poisson model. Upon retrospective analysis of the distribution of trips per hour, there is suggestion of a separate tail distribution for high trio per hour data. Thus, a potential means to improve this model further may be through the use of different Poisson distributions with different rates. The expectation maximisation algorithm would be implemented to estimate the rates for different Poisson distributions.

These results conclude that future trip demand in NYC can be predicted using purely past data. To use this model, having the retrospective data of fare amounts and tip amounts from the taxi data definitely aided the model in making its predictions. However, in practice, if predicting the future trip demand, these values will not be available as these rides have not yet happened to produce the values. As such, this work to build on that done here in transition this model to real world significance would do well to find predictors these values. One approach may include further regression models to predict the average fare amount and tip amount for each neighbourhood for a given hour and day. These predicted values could then be imputed into this model for the prediction of trip demand. This approach intuitively holds greater value than training this model a smaller subset of the data as a hierarchical model architecture allows for greater analysis and aggregation of contextual factors in making these predictions.

CONCLUSION:

The intent of this paper is to predict the 2019 taxi demand on a per neighbourhood basis in NYC using the yellow cab taxi data from 2017 and 2018. The purpose of this prediction is to better inform stakeholders of the taxi industry on where taxi demand is concentrated. There are many potential applications of this including employing trip demand prediction to reduce customer wait times and driver idle time. Taxi companies see the greatest utility in knowing the trip demand in advance. More informed decisions on the distribution of company taxi cabs throughout NYC given the weather and hour of the day may be enabled. This will increase the number of trips their cabs take and, in turn maximise the company's profit.
The last stakeholder mentioned was the urban planners who aim to minimise traffic congestion. This model will enable them to identify where communal transport areas will be most beneficial to the public and ensure that these high demand areas for shared transport are forecasted to continue in the future.

These benefits are not yet fully realised by this model. The strengths of this model have been demonstrated to lie in areas with lower trip density. This may be employed to good effect as these are generally the areas of greatest uncertainty. This analysis raises the potential of supplementing this model's predictions with that of another for areas of high trip density. However, further work is required before this work could transition to reality. Until imputation of several variables is performed, this model is restricted to retrospect.

Ultimately, this analysis shows the efficacy of this approach and demonstrates the value of all included data in make inferences of taxi activity. The analysis shows not only the benefit of using direct taxi data but shows also the value in broader contextual data such as weather.

REFERENCES

Chapter 4 Poisson Regression, (2020, Oct). Retrieved from https://bookdown.org/roback/bookdown-bysh/ch-poissonreg.html

Kamga, Camille & Yazici, M. Anil & Singhal, Abhishek. (2013). Hailing in the Rain: Temporal and Weather-Related Variations in Taxi Ridership and Taxi Demand-Supply Equilibrium.

New York City, NY Weather History. (2020, Oct). Retrieved from https://www.wunderground.com/history/daily/us/ny/new-york-city/KLGA

Rana, Singh. (2019), Taxi demand prediction in New York City. Retrieved from https://medium.com/@ranasinghiitkgp/taxi-demand-prediction-in-new-york-city-916cde6a3492

Appendix

October 9, 2020

[11]:

```
import pandas as pd
import os
import datetime
import seaborn as sns
```

1 Preprocess

[2]:

```
def preprocess_data(filename, arr,PATH):
    final_data = []

    for doc in arr:

        taxi = pd.read_csv(PATH + filename + doc)

        taxi_impossible_vals_removed = taxi[(taxi['tip_amount']>=0) &\
             (taxi['trip_distance']>0) & (taxi['trip_distance']<25) &\
             (taxi['fare_amount']>=2.5) & (taxi['passenger_count']>0) &\
             (taxi['payment_type']==1)& (taxi['extra']>=0)].copy()


        taxi_clean =␣
↪taxi_impossible_vals_removed[['tpep_pickup_datetime','trip_distance','PULocationID',\
                                ␣
↪'DOLocationID','fare_amount','tip_amount','total_amount','extra','mta_tax',\
                                'tolls_amount','improvement_surcharge']]

        columns = list(taxi_clean.columns)
        info = taxi_clean.describe().transpose()

        for index, col in info.iterrows():

            if index in ['VendorID',␣
↪'tpep_pickup_datetime','tpep_dropoff_datetime',␣
↪'PULocationID','DOLocationID']:
                continue

            IQR = col["75%"] - col["25%"]

            upper_lim = col["75%"] + (IQR * 1.5)
            lower_lim = col["25%"] - (IQR * 1.5)

            taxi_clean=taxi_clean.loc[taxi_clean[index]<= upper_lim]
            taxi_clean=taxi_clean.loc[taxi_clean[index]>= lower_lim]

        final_data.append(taxi_clean)

    final_taxi = pd.concat(final_data)

    return final_taxi
```

[7]:
```
def get_datetime(data):

    data_dates = data.copy()
    data_dates['tpep_pickup_datetime'] = pd.to_datetime(data.
 ↪tpep_pickup_datetime)# Pickups

    data_dates['pickup_date'] = data_dates['tpep_pickup_datetime'].dt.date #␣
 ↪Extract date
    data_dates['pickup_hour'] = data_dates['tpep_pickup_datetime'].dt.hour
    data_dates['pickup_month']=data_dates['tpep_pickup_datetime'].dt.month

    data_dates.sort_values(by = ['pickup_date'],inplace=True)

    data_dates.drop(['tpep_pickup_datetime'], axis = 1, inplace = True)

    return data_dates.reset_index()
```

2 Weather data

[5]:
```
WEATHER_PATH = '/Users/Work/Documents/2020/Sem 2/Applied_Data_Science/
 ↪Assignment2/Weather/'

import numpy as np
from bs4 import BeautifulSoup

def get_month_weather(year_month):

    with open(WEATHER_PATH+year_month+".html") as file:
        soup = BeautifulSoup(file.read(), "html.parser")
    tables = soup.find_all("table", "days ng-star-inserted")

    # Get data
    table_data = [
        [col.get_text() for col in row.find_all('td') if len(col.get_text())>10]
        for row in tables[0].find_all("tr")
    ]
    table_data = [data for data in table_data if len(data)>0]
    table_data[1] = [col.split("  ") for col in table_data[1]]

    table_data.append([np.asarray(col).reshape((-1, 3)) for col in␣
↪table_data[1][1:-1]])

    temp_table = [np.asarray(table_data[1][0])]

    for col in table_data[2]:
        temp_table.append(col)

    temp_table.append(np.asarray(table_data[1][-1]))

    final_table = [
        [temp_table[col][row] for col in range(len(temp_table))]
        for row in range(len(temp_table[1]))
    ]

    final_table = [
        [final_table[row][0]] +
        [item for sublist in final_table[row][1:-1] for item in sublist] +
        [final_table[row][-1]]
        for row in range(len(final_table))
    ]

    final_table.insert(0, ['Day'] +
                       [col_name for duplicate_cols in [
                           [item, item, item] for item in table_data[0][:-1]
                       ] for col_name in duplicate_cols] +
                       [table_data[0][-1]])

    return final_table
```

```
[3]: def weather_per_year(year):

         df_lst = []
         for i in range(1,13):

             month = str(year + i)

             tmp = get_month_weather(month)

             if i == 1:
```

```
            row = 1
        else:
            row = 2

        df = pd.DataFrame(tmp[row:-1], columns=tmp[0])


        df["Month"] = i
        df_lst.append(df)

    weather = pd.concat(df_lst).reset_index()


    return weather
```

[2]:

```
def create_date_time(year,weather_data):

    weather_data['Date'] = ""

    for index,col in weather_data.iterrows():

        if index == 0:
            continue

        weather_data.loc[index, 'Date'] = datetime.
 ↪date(year,col['Month'],int(col['Day']))

    weather_data = weather_data[['Temperature (° F)','Wind Speed␣
 ↪(mph)','Precipitation (in)','Date']]


    return weather_data
```

Plots

October 9, 2020

[1]:

```
import pandas as pd
import datetime
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

1 final preprocessing

[2]:

```
def final_preprocess(PATH,filename):

    data = pd.read_csv(PATH + filename)

    data.drop(['Unnamed: 0'], axis = 1, inplace = True)
    data.drop(['index'], axis = 1, inplace = True)

    return data
```

2 Create response variable

[3]:

```
def create_target(data):

    data_by_hour = data.groupby(['PULocationID','pickup_date','pickup_hour']).
 ↪mean().reset_index()
    trips_per_hour = data.groupby(['PULocationID','pickup_date','pickup_hour']).
 ↪size().reset_index()
    data_by_hour['Trips_per_hour'] = trips_per_hour[0]

    return data_by_hour
```

[318]:

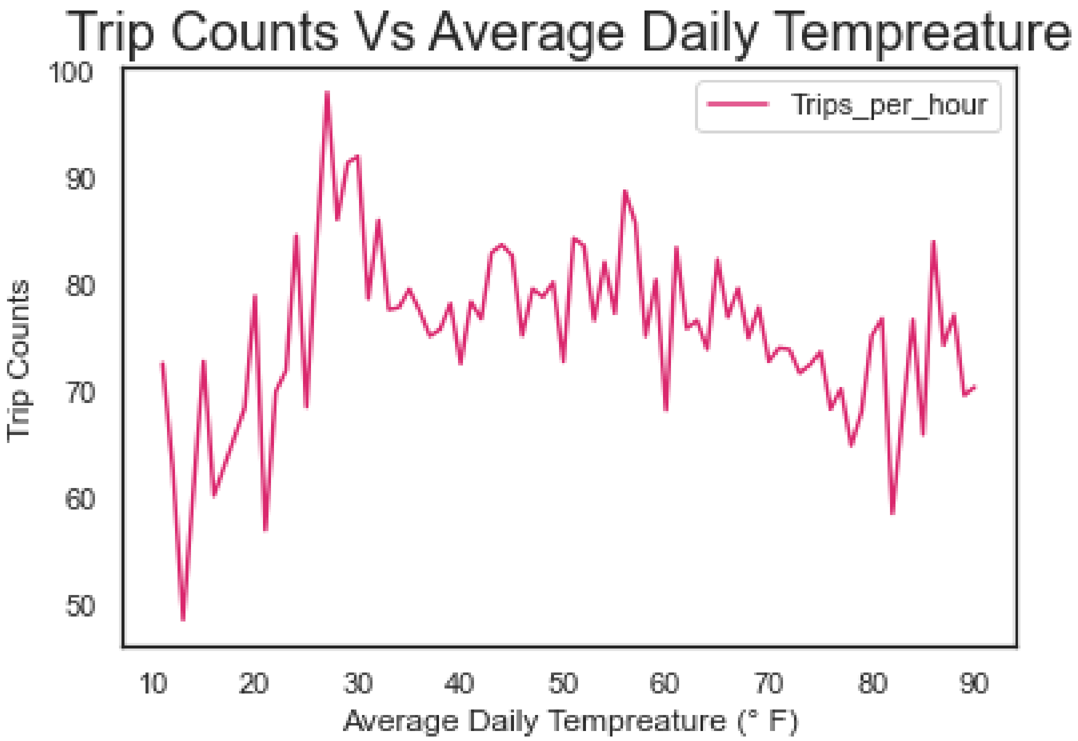
```
info = train_data.describe().transpose()
info.to_csv(PATH + 'train_sum_stats.csv')
```

3 Weather visualisations

3.0.1 Trempreature

[323]:

```
train_data['Av_temp'] = train_data['Av_temp'].astype(int)
temp_graph_data = train_data.groupby(['Av_temp']).mean().reset_index()

fig = temp_graph_data.plot(x='Av_temp',y= 'Trips_per_hour',color = '#DA276D',␣
 ↪xlabel = 'Average Daily Trempreature (° F)',ylabel = 'Trip Counts')

fig.axes.set_title("Trip Counts Vs Average Daily Trempreature",fontsize=20)

fig.figure.savefig('temp_vs_trips.png')
```

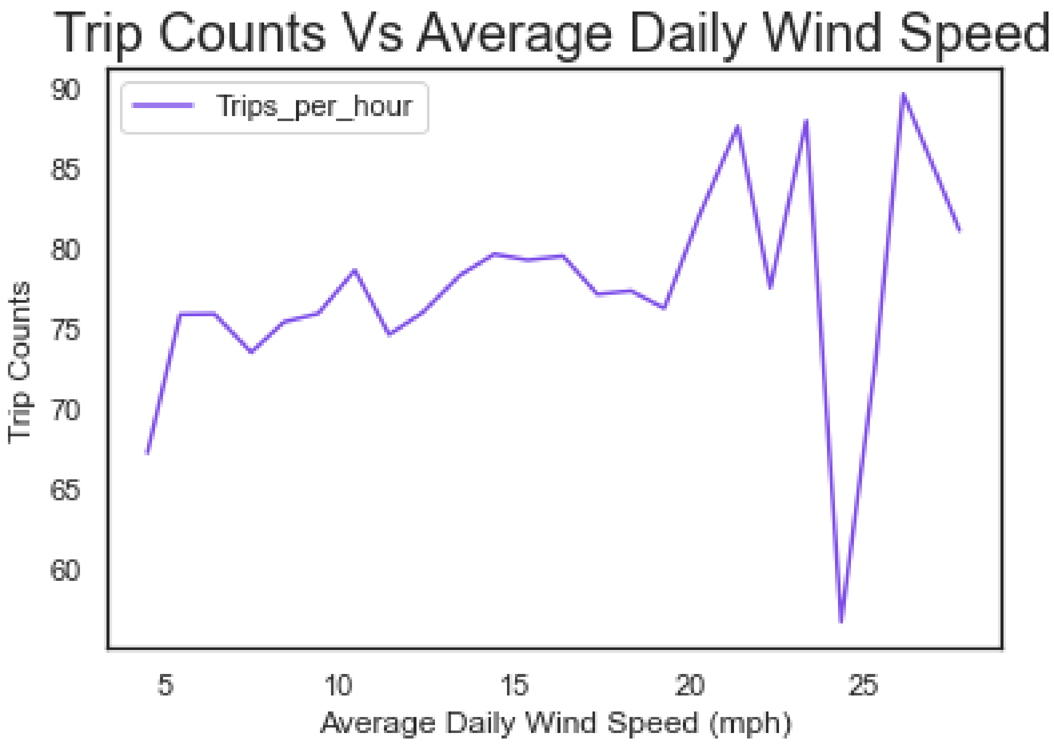


3.0.2 Wind

[324]:

```
train_data['Wind Speed (mph).1'] = train_data['Wind Speed (mph).1'].astype(int)

wind_graph_data = train_data.groupby(['Wind Speed (mph).1']).mean().
 ↪reset_index()

fig = wind_graph_data.plot(x='Av_wind',y= 'Trips_per_hour',color = '#7C4CE9',␣
 ↪xlabel = 'Average Daily Wind Speed (mph)',ylabel = 'Trip Counts')
```

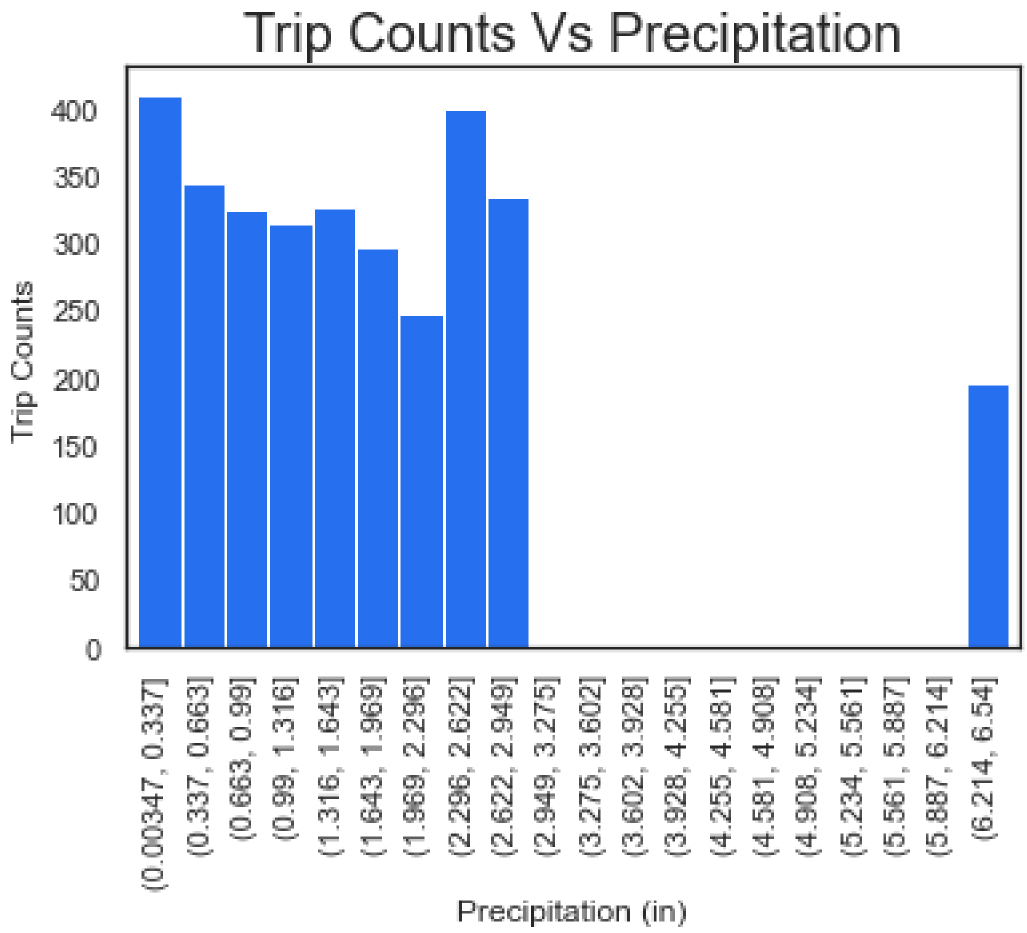
```
fig.axes.set_title("Trip Counts Vs Average Daily Wind Speed",fontsize=20)

fig.figure.savefig('wind_vs_trips.png')
```



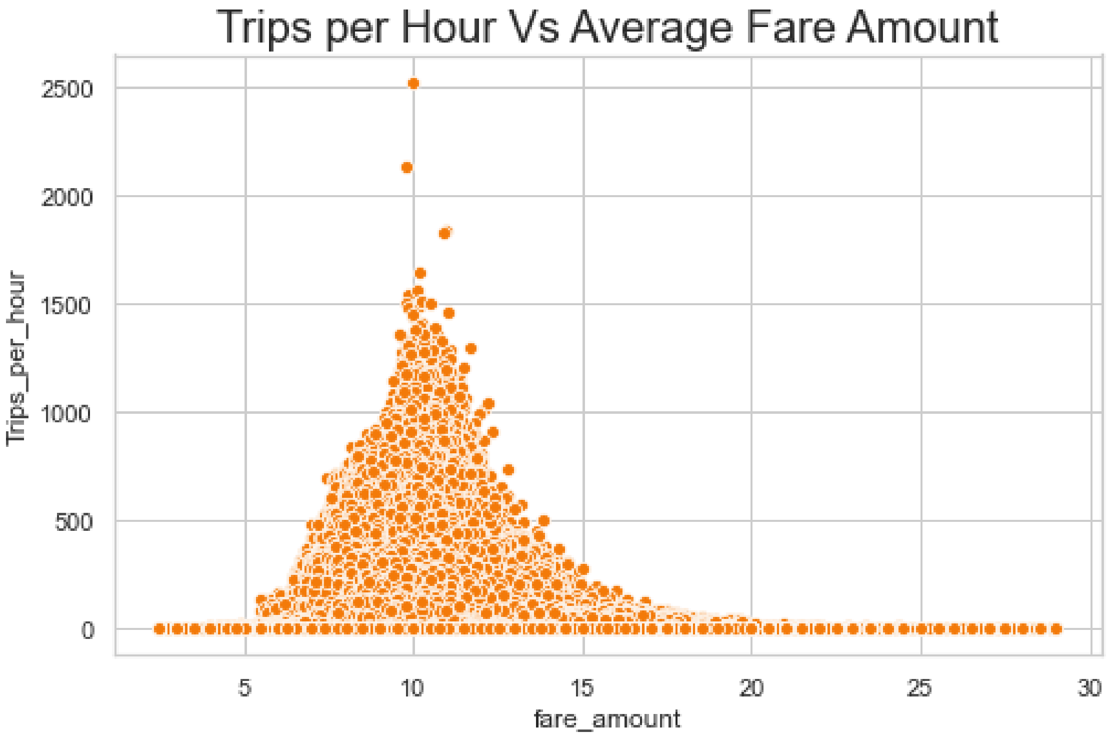
3.0.3 Precipitation

[325]:

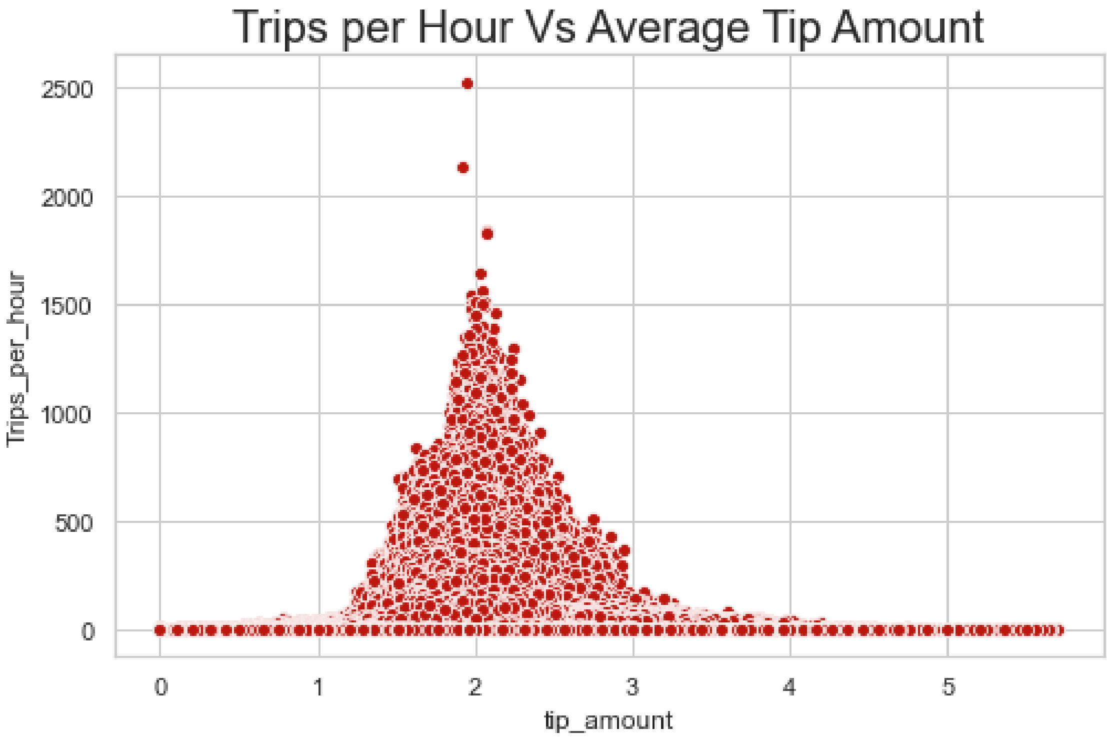
```
rain_graph_data = train_data.groupby(['Precipitation']).mean().reset_index()

rain_graph_data = rain_graph_data[rain_graph_data['Precipitation'] > 0]

rain_graph_data['Precipitation'] = pd.cut(rain_graph_data['Precipitation'],␣
 ↪bins = 20)

rain_graph_data = rain_graph_data.groupby(['Precipitation']).mean().
 ↪reset_index()

fig = rain_graph_data.plot(x = 'Precipitation',y =␣
 ↪'Trips_per_hour',kind='bar',color = '#2670F0',width = 1, xlabel =␣
 ↪'Precipitation (in)',ylabel = 'Trip Counts',legend=False)

fig.axes.set_title("Trip Counts Vs Precipitation",fontsize=20)
fig.figure.savefig('trips_Vs_rain.png')
```



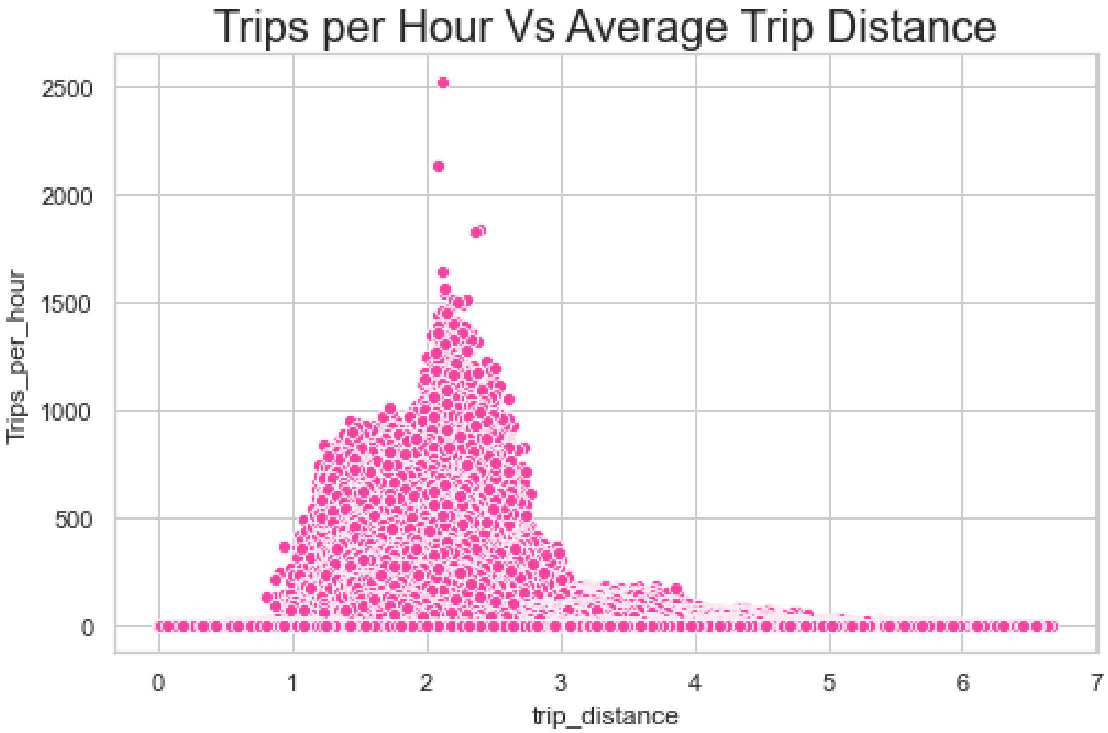
[249]:
```
f, axes = plt.subplots(1, 1, figsize=(8, 5), sharex=True)

fig = sns.
 ↪scatterplot(x=train_data['fare_amount'],y=train_data['Trips_per_hour'],color␣
 ↪='#F37A09' )

plt.title("Trips per Hour Vs Average Fare Amount",fontsize=20)

fig.figure.savefig('trips_vs_fare.png')
```



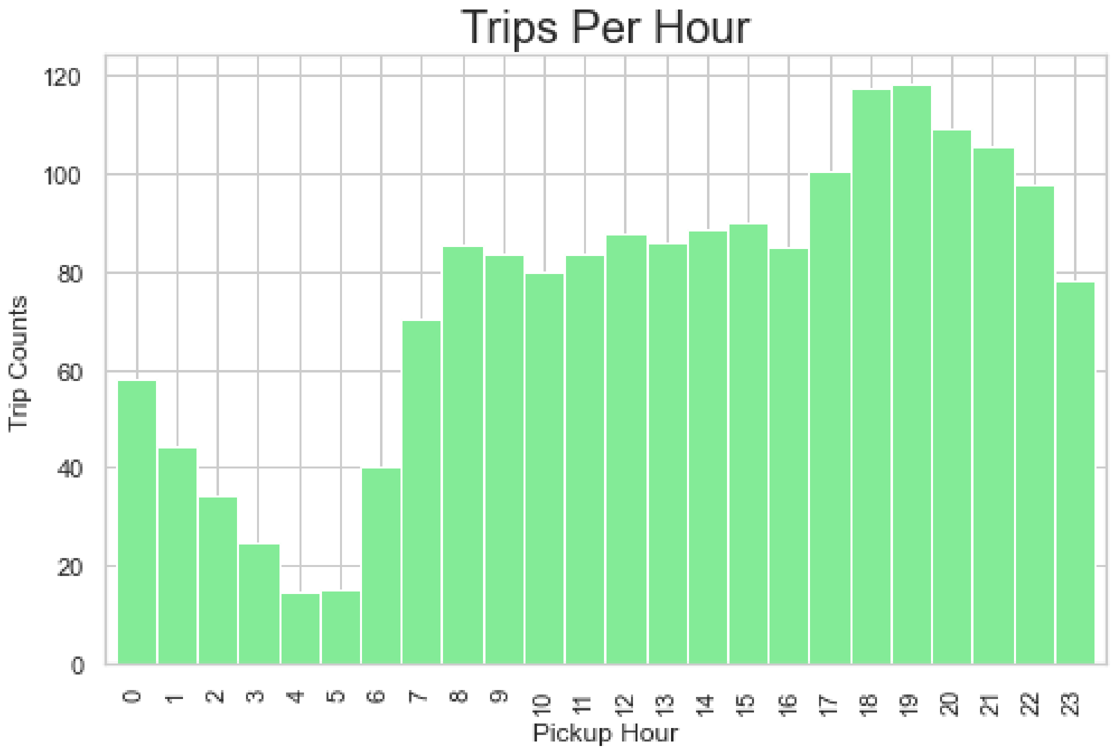
[250]:
```
f, axes = plt.subplots(1, 1, figsize=(8, 5), sharex=True)

fig = sns.
 ↪scatterplot(x=train_data['tip_amount'],y=train_data['Trips_per_hour'],color␣
 ↪='#BB170F' )

plt.title("Trips per Hour Vs Average Tip Amount",fontsize=20)

fig.figure.savefig('trips_vs_tip.png')
```



[251]:
```
f, axes = plt.subplots(1, 1, figsize=(8, 5), sharex=True)

fig = sns.
 ↪scatterplot(x=train_data['trip_distance'],y=train_data['Trips_per_hour'],color␣
 ↪='#F2439D' )

plt.title("Trips per Hour Vs Average Trip Distance",fontsize=20)

fig.figure.savefig('trips_vs_tripdist.png')
```

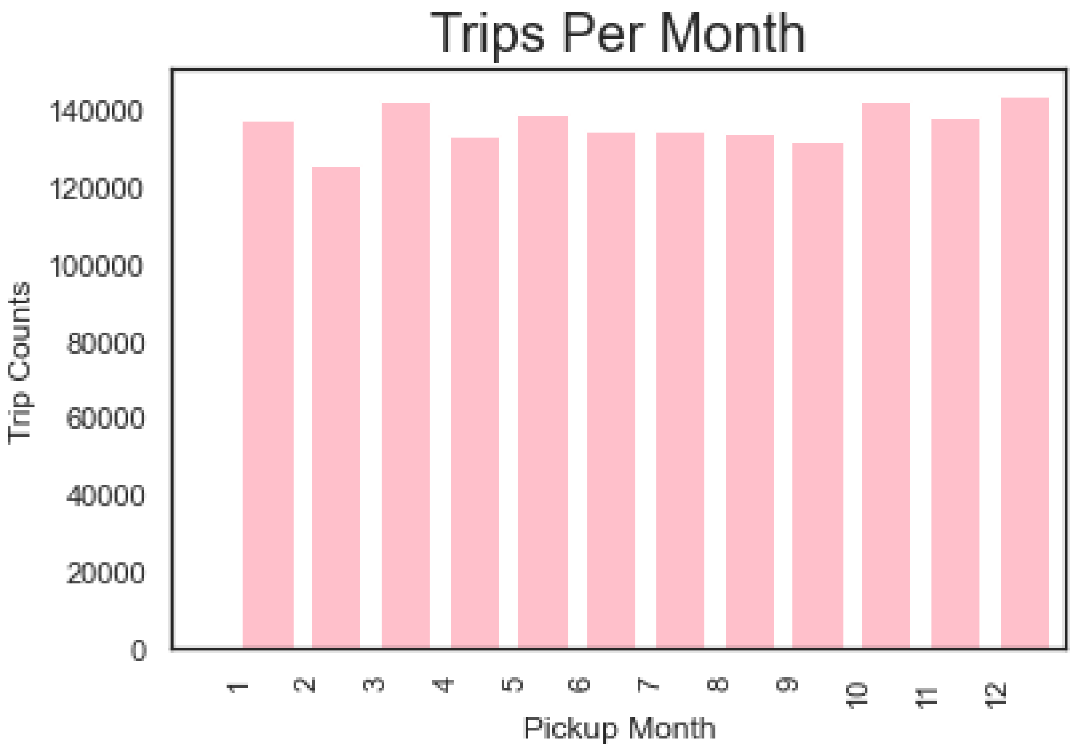


```
[220]: hour_graph_data = train_data.groupby(['pickup_hour']).mean().reset_index()
hour_graph_data.set_index('pickup_hour', drop=False, inplace=True)
hour_graph_data = hour_graph_data['Trips_per_hour']

f, axes = plt.subplots(1, 1, figsize=(8, 5), sharex=True)

fig = hour_graph_data.plot(kind='bar',color = '#83EB97',width = 1, xlabel =␣
 ↪'Pickup Hour',ylabel = 'Trip Counts')

fig.axes.set_title("Trips Per Hour",fontsize=20)

fig.figure.savefig('trips_per_hour.png')
```



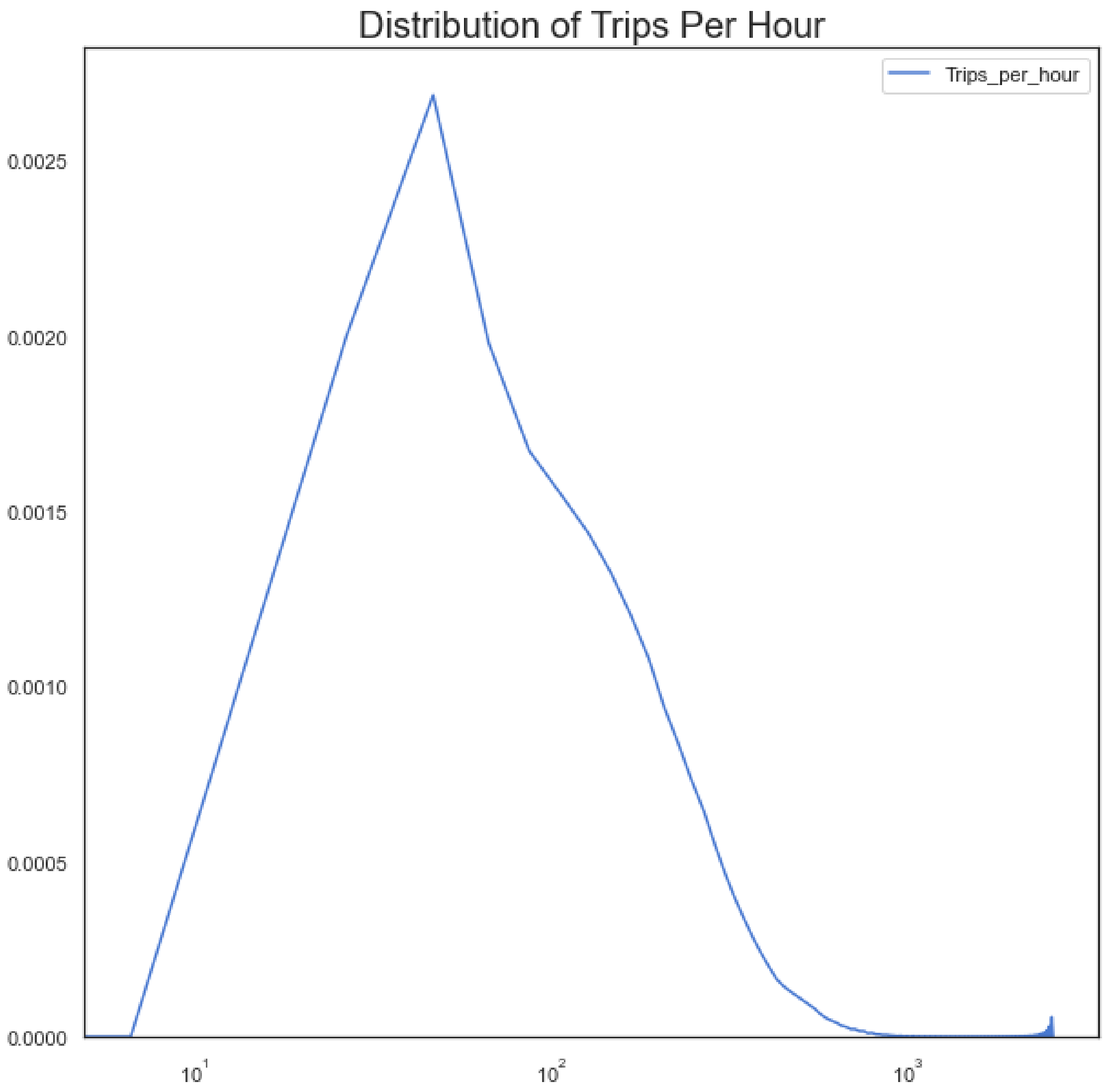
[326]:

```python
month_graph_data = train_data.groupby(['pickup_month']).size().reset_index()

month_graph_data['pickup_month'] = month_graph_data['pickup_month'].astype(int)

month_graph_data.set_index('pickup_month', drop=False, inplace=True)

fig = month_graph_data.loc[1:].plot(kind='bar',color = 'pink',width = 1.5,␣
 ↪xlabel = 'Pickup Month',ylabel = 'Trip Counts',legend=False)

fig.axes.set_title("Trips Per Month",fontsize=20)

fig.figure.savefig('trips_per_month.png')
```

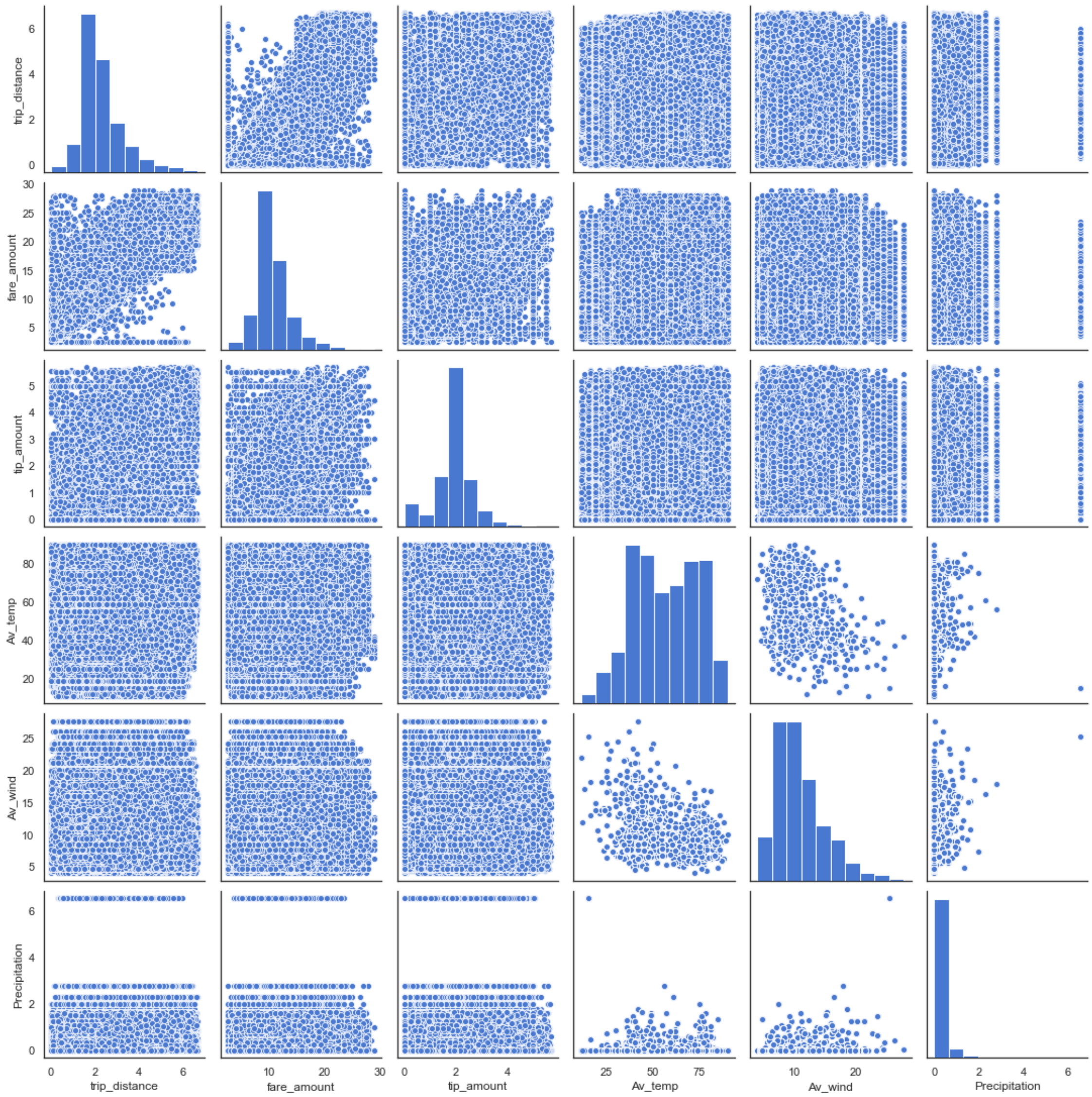


[309]:

```
plt.figure(figsize=(10,10))

fig = sns.kdeplot(train_data['Trips_per_hour'])
fig.set_title("Distribution of Trips Per Hour",fontsize = 20)

fig.set_xscale('log')


fig.figure.savefig('Trips_per_hour_distribution.png')
```

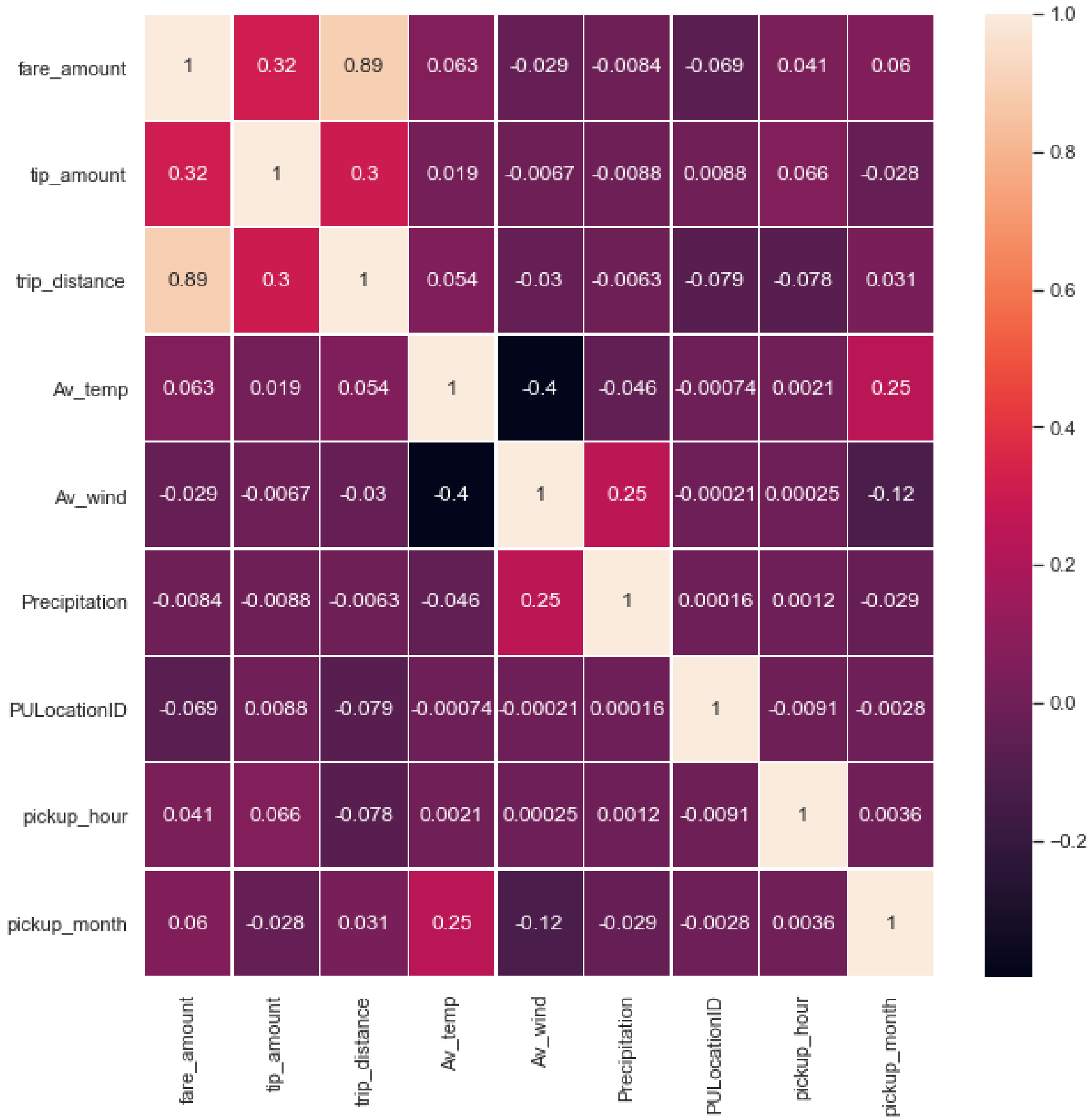


```
[322]: fig = sns.
        ↪pairplot(train_data[['trip_distance','fare_amount','tip_amount','Av_temp','Av_wind','Precip

       fig.savefig('Pairwise_relationships.png')
```



[307]:

```
fig, ax = plt.subplots(figsize=(10,10))
fig = sns.heatmap(corr_data.corr(),annot=True,lw=0.5,ax=ax)

fig.figure.savefig('correlation.png')
```


fare_amount
tip_amount
trip_distance
Av_temp
Av_wind
Precipitation
PULocationID
pickup_hour
pickup_month
1 0.32 0.89 0.063 -0.029 -0.0084 -0.069 0.041 0.06
0.32 1 0.3 0.019 -0.0067 -0.0088 0.0088 0.066 -0.028
0.89 0.3 1 0.054 -0.03 -0.0063 -0.079 -0.078 0.031
0.063 0.019 0.054 1 -0.4 -0.046 -0.00074 0.0021 0.25
-0.029 -0.0067 -0.03 -0.4 1 0.25 -0.00021 0.00025 -0.12
-0.0084 -0.0088 -0.0063 -0.046 0.25 1 0.00016 0.0012 -0.029
-0.069 0.0088 -0.079 -0.00074 -0.00021 0.00016 1 -0.0091 -0.0028
0.041 0.066 -0.078 0.0021 0.00025 0.0012 -0.0091 1 0.0036
0.06 -0.028 0.031 0.25 -0.12 -0.029 -0.0028 0.0036 1
fare_amount
tip_amount
trip_distance
Av_temp
Av_wind
Precipitation
PULocationID
pickup_hour
pickup_month
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Plots

October 9, 2020

[10]:

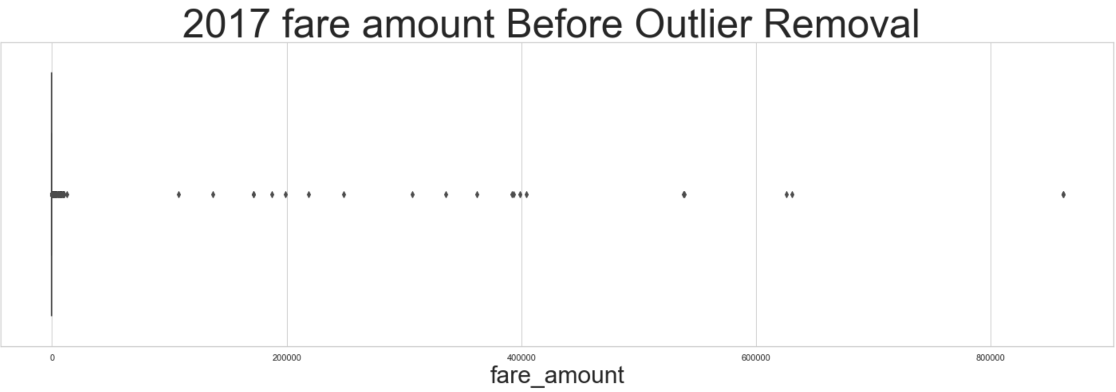
```
import pandas as pd
import matplotlib.pyplot as plt

PATH = '/Users/Work/Documents/2020/Sem 2/Applied_Data_Science/Assignment2/'

taxi_17_clean = pd.read_csv('taxi_17_clean.csv')
```

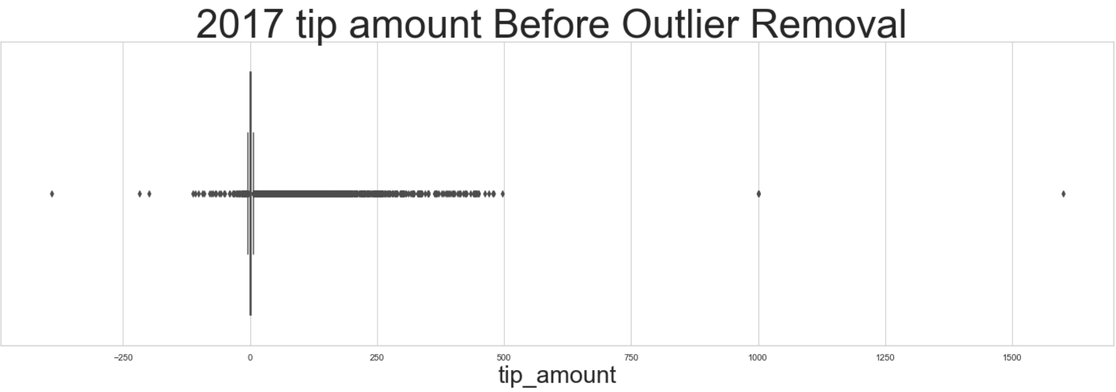
[19]:

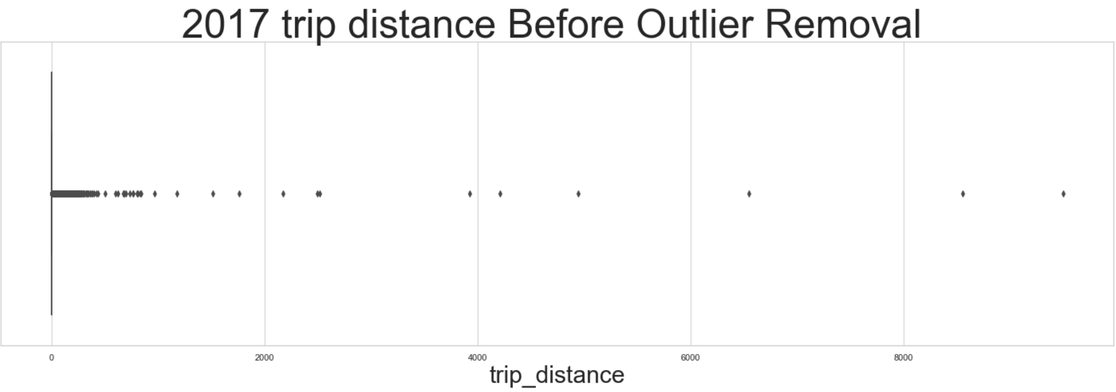
```
import seaborn as sns
import matplotlib.pyplot as plt

def pre_outlier_boxplot(data):

    for i in ['fare_amount','tip_amount','trip_distance']:

        sns.set(style="whitegrid", palette="muted", color_codes=True)
        f, axes = plt.subplots(1, 1, figsize=(25, 7), sharex=True)

        fig = sns.boxplot(x=data[i],color="magenta")

        fig.axes.set_title("2017 " + i.replace("_"," ") + " Before Outlier␣
 ↪Removal ",fontsize=50)

        fig.set_xlabel(i,fontsize=30)


        fig.figure.savefig(i + '17.png')

        plt.show()
```

[20]:

```
pre_outlier_boxplot(taxi_17)
```


2017 fare amount Before Outlier Removal


2017 tip amount Before Outlier Removal

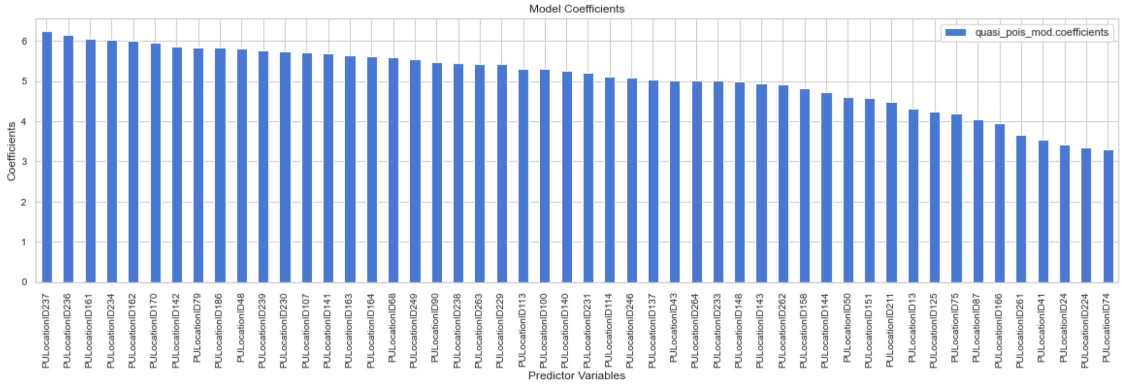

2017 trip distance Before Outlier Removal

```
[86]: params = pd.read_csv(PATH + 'parameters (1).csv')

      best = params.sort_values(by = 'quasi_pois_mod.coefficients', ascending = False)

      no_pickupID = params.iloc[:40]
```

```python
pickup_ID = params.iloc[40:]

no_pickupID = no_pickupID.set_index('names.quasi_pois_mod.coefficients.')
```

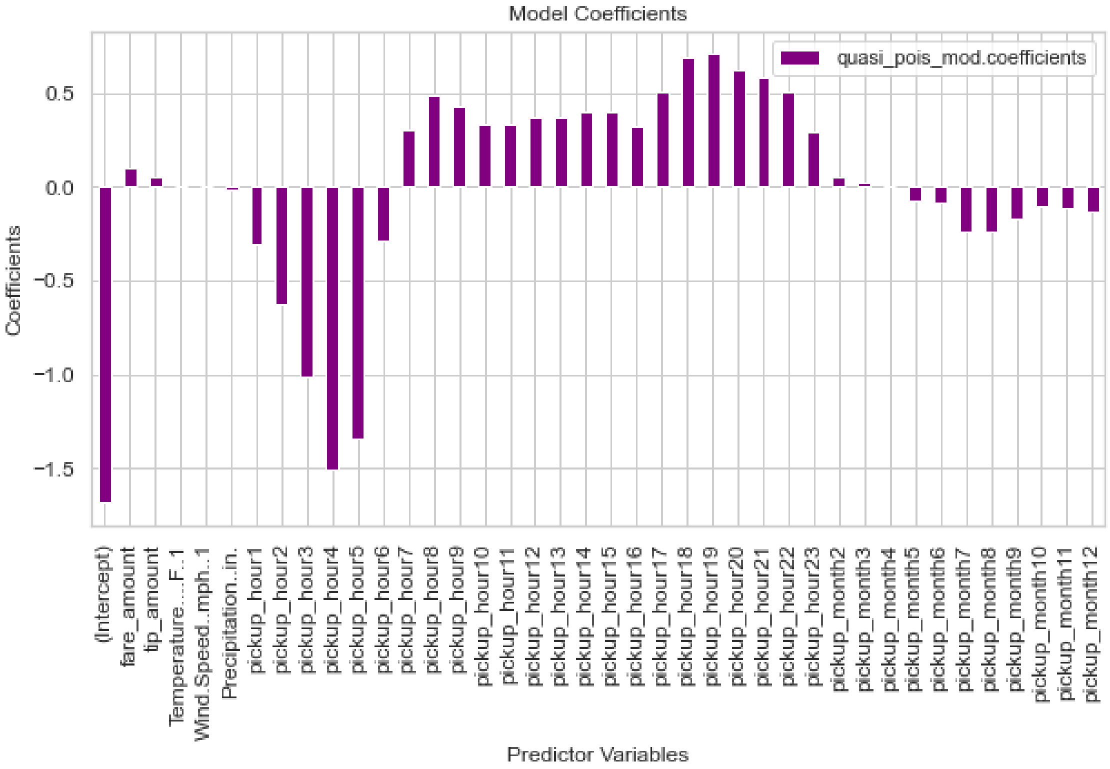
[84]:

```python
best = best.head(50)

best = best.set_index('names.quasi_pois_mod.coefficients.')

fig = best.plot(kind='bar', title='Model Coefficients',figsize =␣
↪(20,5),fontsize = 10,xlabel = 'Predictor Variables', ylabel = "Coefficients")

fig.figure.savefig(PATH + 'params_best.png')
```



[91]:

```python
fig = no_pickupID.plot(kind='bar', title='Model Coefficients',figsize =␣
↪(10,5),fontsize = 12,xlabel = 'Predictor Variables',ylabel = "Coefficients",␣
↪color = 'purple')

fig.figure.savefig(PATH + 'params_noID.png')
```

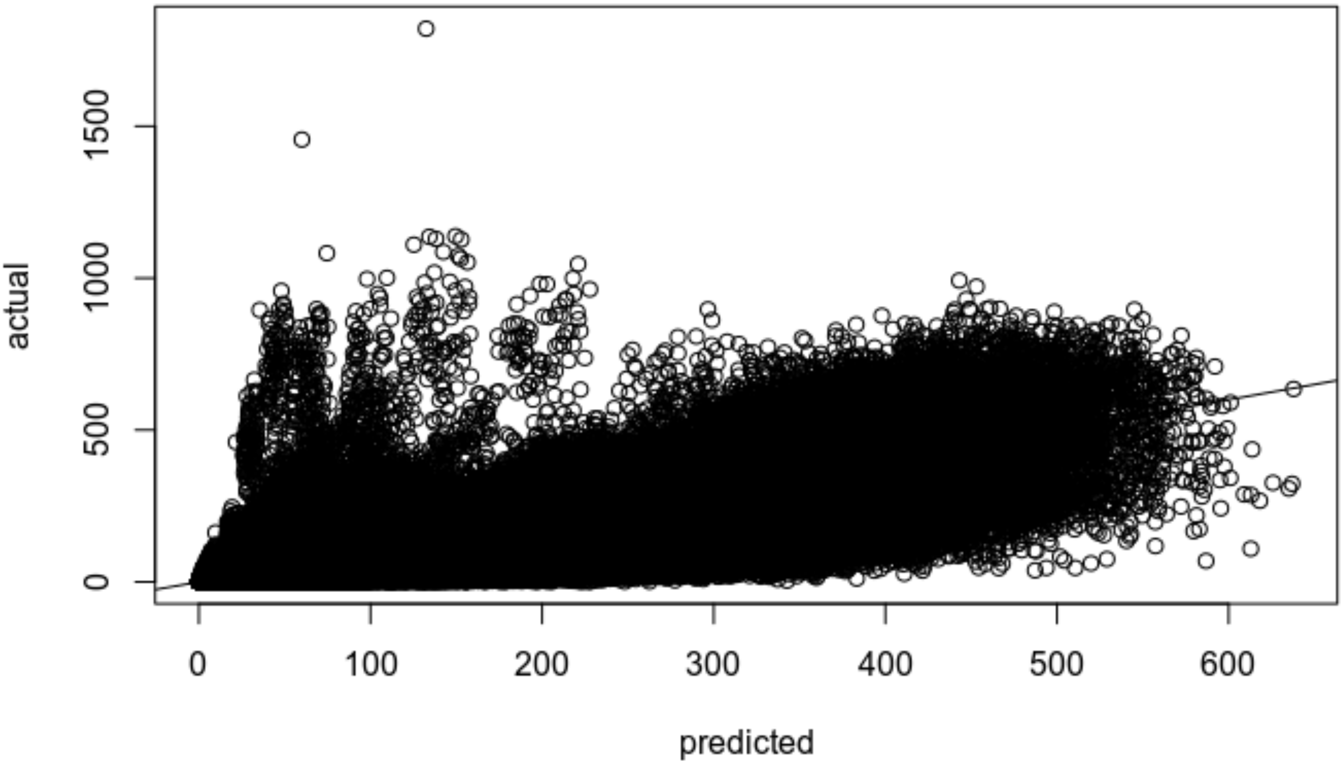

Model Coefficients
quasi_pois_mod.coefficients
0.5
0.0
−0.5
−1.0
−1.5
Coefficients
(Intercept)
fare_amount
tip_amount
Temperature....F..1
Wind.Speed..mph..1
Precipitation..in.
pickup_hour1
pickup_hour2
pickup_hour3
pickup_hour4
pickup_hour5
pickup_hour6
pickup_hour7
pickup_hour8
pickup_hour9
pickup_hour10
pickup_hour11
pickup_hour12
pickup_hour13
pickup_hour14
pickup_hour15
pickup_hour16
pickup_hour17
pickup_hour18
pickup_hour19
pickup_hour20
pickup_hour21
pickup_hour22
pickup_hour23
pickup_month2
pickup_month3
pickup_month4
pickup_month5
pickup_month6
pickup_month7
pickup_month8
pickup_month9
pickup_month10
pickup_month11
pickup_month12
Predictor Variables

Possion Model

Code ▾

Hide

```
# 2017 -2018 data

data <- read.csv('train_data.csv')

data$pickup_hour <- factor(data$pickup_hour)
data$PULocationID <- factor(data$PULocationID)
data$pickup_month <- factor(data$pickup_month)

# 2019 data
test <- read.csv('test_data.csv')

test$pickup_hour <- factor(test$pickup_hour)
test$PULocationID <- factor(test$PULocationID)
test$pickup_month <- factor(test$pickup_month)


predictors <- c("fare_amount","tip_amount","Temperature....F..1","Wind.Speed..mph..1","Precipitation..in.","PULocationID","pickup_hour","pickup_month")

# poisson model

#pois_mod <- glm(Trips_per_hour ~ fare_amount + tip_amount + Temperature....F..1+ Wind.Speed..mph..1 + Precipitation..in. + pickup_hour + pickup_month + PULocationID, data=data, family="poisson")

#quasi-poisson model

#pois_mod_quasipossion<- glm(Trips_per_hour ~ fare_amount + tip_amount + Temperature....F..1 + Wind.Speed..mph..1 + Precipitation..in. + pickup_hour + pickup_month + PULocationID,data=data, family=quasipoisson)

#pois_mod_no_weather <- glm(Trips_per_hour ~ fare_amount + tip_amount + pickup_hour + pickup_month + PULocationID,data=data, family="poisson")

#quasi_mod_no_weather <- glm(Trips_per_hour ~ fare_amount + tip_amount + pickup_hour + pickup_month + PULocationID,data=data, family=quasipoisson)


#model_selected <- step(pois_mod,scope=~.)
```



Hide

```
# over dispersion

(phi <- sum(residuals(pois_mod,type="pearson")^2/1639876))
```

```
[1] 27.94917
```

Evaluation

Hide

```
#R-squared

1 - quasi_pois_mod$deviance/quasi_pois_mod$null.deviance
```

```
[1] 0.8577216
```

Hide

```
predicted = predict(quasi_pois_mod,newdata = test,type= 'response')
predict_train <-  predict(quasi_pois_mod,newdata = train_eval,type= 'response')

# mean squared prediction error for test set

sqrt(sum((test_true - predicted)^2)/length(test_true[,1]))
```

```
[1] 54.55061
```

Hide

```
# mean squared prediction error for train set

sqrt(sum((train_labels - predict_train)^2)/length(train_labels[,1]))
```

```
[1] 57.86204
```

Hide

```
# mean absolute error

sum(abs((test_true - predicted)))/length(test_true[,1])
```

```
[1] 28.41104
```

Hide

```
sum(abs((train_labels -predict_train)))/length(train_labels[,1])
```

```
[1] 27.34586
```

Hide

```


# means weather is significant
anova(quasi_pois_mod,quasi_mod_no_weather,test = 'F')
```

```
Analysis of Deviance Table

Model 1: Trips_per_hour ~ fare_amount + tip_amount + Temperature....F..1 + 
    Wind.Speed..mph..1 + Precipitation..in. + pickup_hour + pickup_month + 
    PULocationID
Model 2: Trips_per_hour ~ fare_amount + tip_amount + pickup_hour + pickup_month + 
    PULocationID
  Resid. Df Resid. Dev Df Deviance      F    Pr(>F)    
1   1639876   33039515                                 
2   1639879   33057975 -3   -18461 220.17 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```