# CSP571 - DATA PREPARATION AND ANALYSIS
# FINAL PROJECT REPORT

## TIME SERIES ANALYSIS AND FORECASTING - NYC TAXI FARE

Yash Pradeep Gupte (A20472798)
Lakshmi Himaja Amrutham (A20474105)

## 1. ABSTRACT

When attempting to anticipate a number for the future, time series is an extrapolation problem. Regression problems are interpolation difficulties, in contrast to non-time series issues. In order to predict the data over the defined time intervals and to derive relevant statistics and other features from the data, time series analysis can be utilized. This aids in forecasting future values based on historical data. The demand for transportation has increased exponentially in metropolitan cities. New York City, one of the busiest cities in the US, has a high demand for taxi rides. Various factors like pickup and drop off locations and times affect the taxi fares directly. We seek to utilize the NYC taxi Fare dataset from the open source Kaggle platform and perform statistical analysis to determine future taxi fares based on pick up and drop off factors. This data collection includes 55M records with information related to location and time durations. We will be implementing Time Series Forecasting in R to predict taxi fares and Clustering Analysis to observe famous locations in NYC and their impact on taxi fare amount.

## 2. OVERVIEW

### 2.1 PROBLEM STATEMENT

Time series is an extrapolation problem where we want to predict a quantity in future.Where as non-time series problems like regression are interpolation problems. The dataset we're dealing with has potential features to determine future taxi fares along with pickup and drop off dynamics. The goal of this project is to explore time series and clustering techniques that could guide us towards a conclusion. Metropolitan cities demand a high amount of transport and analyze taxi fares. The location and time features have a direct impact on taxi fares, and utilization of data analysis and machine learning techniques can provide a better insight on prominent locations and times while booking a taxi leading to a higher profit.

In this study, we aim to address following questions regarding the NYC Taxi Fare Prediction Dataset,

- What are the features that affect the fare of the taxi?
- What are the locations where the taxi fare is relatively high and low(demand based on location)?
- What is the time when the fare of the taxi is high as well as low (demand based on time)?
- Does the dataset require additional features to determine the proposed outcome ?
- What features are statistically correlated with each other?

- How does stationarity and seasonality affect our time series analysis?

## 2.2 REVIEW OF LITERATURE

[1] Dama, Fatoumata and Sinoquet, Christine. “Time Series Analysis and Modeling to Forecast: a Survey”. 10.48550/ARXIV.2104.00164

The article takes a descriptive approach to time series analysis and forecasting, considering a red thread that leads the reader from time series preprocessing to forecasting by having them perform various tasks on the data, such as visualizing the data, finding patterns in the data, correlating the data, clustering the data, identifying outliers, forecasting, and simulating the results. To distinguish nonstationary effects from the remaining stochastic component, which is presumed to be stationary, time series decomposition is a crucial step in the preprocessing process[1].

[2] Antoniades, Ch., Delara Fadavi and Antoine Foba Amon. “Fare and Duration Prediction : A Study of New York City Taxi Rides.”

In this study, the authors seek to analyze and predict taxi fare based on time series data. Major factors attracting taxi rides including but not limited to traffic patterns, road blockage large scale events can be deduced from millions of rides taken each month by consumers. Predicting the cost and length of a trip can, for example, help drivers select which of two prospective rides will be most profitable or passengers decide when is the best time to start their journey.They have mainly utilized features like pickup and drop off coordinates, distance of the trip, start time, number of passengers and rate code to predict fares. Two models - Linear regression with Lasso and Random Forest models were used to predict duration and fare amount[3].

[3] A. Ian Mcleod and Hao Yu and Esam Mahdi. “ Time Series Analysis with R”

In this article, the authors give a quick introduction to the R statistical computing and programming environment and explain why R may be helpful for many time series researchers working on both applied and theoretical projects. Some topics include state space models, structural change, generalized linear models, threshold models, neural nets, co-integration, GARCH, wavelets, and stochastic differential equations. These topics are supported by R and are of intermediate and advanced level[2].

## 2.3 PROPOSED METHODOLOGY

We employ statistical and machine learning techniques to explore data, clean data, process transformations, generate and analyze features and build models to predict taxi fares in NYC. We have performed time series analysis and forecasting and cluster analysis as the dataset contains pickup date time stamps and pickup dropoff coordinates. Feature extraction is done to observe the impact of individual features on fare prices. We have performed descriptive analysis using k-means clustering, ACF and PACF plots, time series plots to identify stationarity and seasonality in the dataset. Performance metrics such as MLE, RMSE, MAPE functions have been set up to evaluate models such as Linear Regression, ARIMA and Seasonal ARIMA.

## 3. DATA PROCESSING

### 3.1 PIPELINE DETAILS

Data Structure : Two csv files - one for train and one for test set. Train contains 55M records and Test contains 10k records
Dataset size : 5.5 GB
Dimensions : train (55423856, 8) and test (9914, 8)
Feature Description:

- key (char) - unique string identifying each row in both the training and test sets. Composed of **pickup_datetime** plus a unique integer, a unique ID field.
- pickup_datetime - timestamp value indicating when the taxi ride started.
- pickup_longitude - float for longitude coordinate of where the taxi ride started.
- pickup_latitude - float for latitude coordinate of where the taxi ride started.
- dropoff_longitude - float for longitude coordinate of where the taxi ride ended.
- dropoff_latitude - float for latitude coordinate of where the taxi ride ended.
- passenger_count - integer indicating the number of passengers in the taxi ride.
- fare_amount - float indicating the taxi fare amount

### 3.2 DATA ISSUES

NYC taxi fare dataset contains information about pick up and drop off timestamps, locations , passenger count and the fare amount incurred. The actual size of this dataset is 5.5GB with 55M train observations and 10K test observations with 8 features.

To process this huge chunk of data, we consider only a subset of this dataset - about 2M observations from the train set and proceed for further preprocessing and analysis.

### 3.3 ASSUMPTIONS / ADJUSTMENTS

For clustering and performing the linear model on the dataset. Initially Omitting the NaN values present in the dataset. We could see that in the passengers count column there are 0's present omitting those rows.

For time series analysis and forecasting in R, we need to convert our dataset into a time series understandable objects which can be handled efficiently. First we loaded the dataset in the form of a dataframe and then performed data preprocessing. Initially the datetime for pickup in dataframe are of the following format:

| | key | fare_amount | pickup_datetime | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|---|---|---|---|---|---|---|---|---|
| 1 | 2009-06-15 17:26:21 | 4.50 | 2009-06-15 17:26:21 | -73.84431 | 40.72132 | -73.84161 | 40.71228 | 1 |
| 2 | 2010-01-05 16:52:16 | 16.90 | 2010-01-05 16:52:16 | -74.01605 | 40.71130 | -73.97927 | 40.78200 | 1 |
| 3 | 2011-08-18 00:35:00 | 5.70 | 2011-08-18 00:35:00 | -73.98274 | 40.76127 | -73.99124 | 40.75056 | 2 |
| 4 | 2012-04-21 04:30:42 | 7.70 | 2012-04-21 04:30:42 | -73.98713 | 40.73314 | -73.99157 | 40.75809 | 1 |
| 5 | 2010-03-09 07:51:00 | 5.30 | 2010-03-09 07:51:00 | -73.96810 | 40.76801 | -73.95665 | 40.78376 | 1 |

The key column is read as a character in R while loading the dataset. For Time series Analysis, we had to convert into a date time format. There are various date time formats in R like datetime, POSIXlt

and POSIXct. Utilizing the lubridate() package in R, we were able to convert the key and pickup_datetime feature into POSIXct format.

## 4. DATA ANALYSIS

### 4.1 SUMMARY STATISTICS

4.1.1 Cluster Analysis Statistics

By using clustering, we may determine which observations come into one category or similar and it is useful for classifying them. The simplest and most popular clustering technique for splitting the dataset into a set of k groups is known as "K-means clustering."

K-means algorithm as follows:

1. Specifying the desired number of clusters (K) to be created. Initially the k values are considered as the initial cluster centers.
2. Based on the Euclidean distance between the item and the centroid, each observation is assigned to its nearest centroid.
3. Update the cluster centroid for each of the k clusters by finding the updated means of all the cluster's data points. The means of all the variables for the observations in the Kth cluster are contained in a vector of length p that is the centroid of the Kth cluster; p is the number of variables.
4. Reduce the total within the sum of squares iteratively. Repeat steps 3 and 4 until the cluster assignments finish changing or the allotted number of iterations has been used.

Computing k-means clustering in R

The data will be divided into two clusters here (centers = 2).Adding nstart = 25, for instance, will produce 25 initial configurations.

The output of k means is a list with several bits of information.

- cluster: A vector of integers to which each point is assigned..
- centers: A matrix of cluster centers.
- totss: The total sum of squares.
- withinss: Vector of within-cluster sum of squares, one component per cluster.
- tot.withinss: Total within-cluster sum of squares.
- betweenss: The between-cluster sum of squares.
- size: The number of points in each cluster.

4.1.2 Time Series Analysis Statistics

Initial thoughts on the dataset : features are not encoded efficiently. Summarizing the dataset provided us statistic which are noted below

- The fare amount is in the range of -62.00 to 1273.31, which looks unrealistic initially. Removing the negative fare amounts was the first step. Next, a trip with a fare amount of $1273.31 might be an outlier. People won’t spend such huge amounts of money on a taxi or the distance between

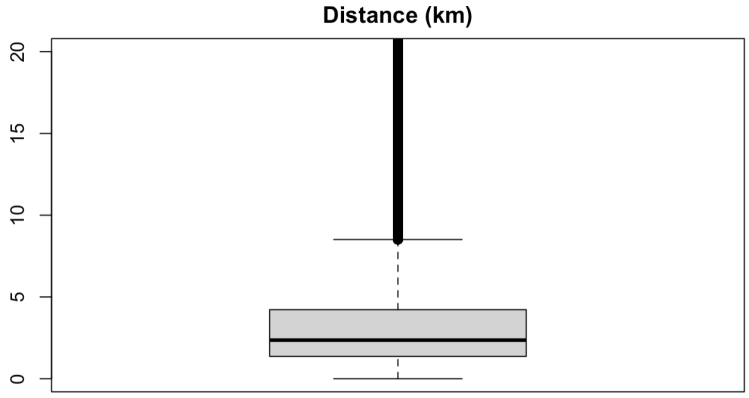
pickup and drop off should be very long(within NYC) to produce this kind fare amount. Thus we limit the fare amount between the range $1 to $500 and remove the rest records.

- The passenger count feature contains a maximum number of 208 passengers in one Taxi. This again is an outlier. Hence we limit this number to be between 1 to 6 passengers , if we consider a taxi to be an SUV vehicle.
- Generally the range of latitude is between -90 to 90 degrees and that for longitude is -180 to 180 degrees. We removed any coordinate values outside this range.
- There were 28 NaN values present in the dataset which were decided to be removed on the fact that these 28 values do not present any importance when compared to a total of 2M records.

**4.2 FEATURE EXTRACTION AND VISUALIZATION**

4.2.1 Clustering

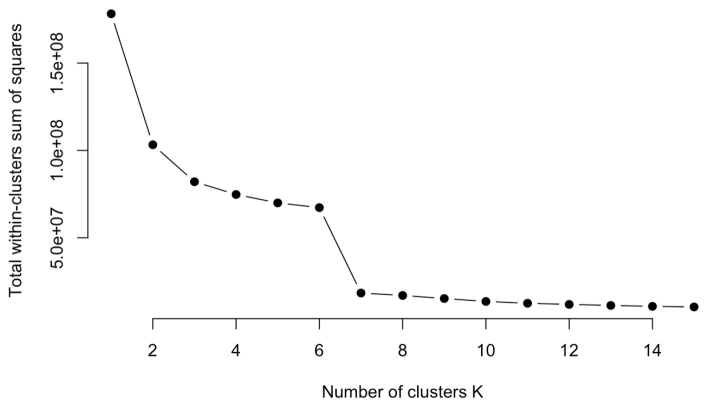
Examining the nan values. removing the dataset's nan values. I ran a passenger count check. I removed the rows that made no sense, such as those where passenger count was 0. The distance is calculated for each journey and included as a feature to the test and training datasets. The pick-up time and date data are added to get the pick-up hour and the month the journey occurred in. examining the dataframe's outliers that are present. removing the dataset's outliers with distances of 0 and 20 kilometers.



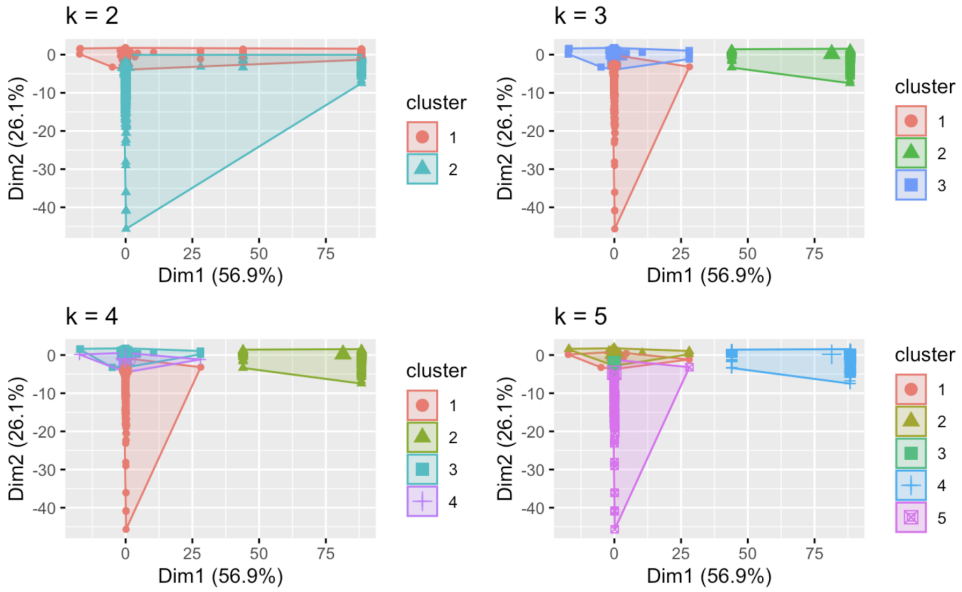

Using a sample of data from the train dataset for both pickup and dropoff locations, we created a Leaflet graph (a map view graph). By tying the pickup latitude and longitude, two additional features were added to the test and training datasets. Applying the same procedure to the longitude and latitude of the drop off.

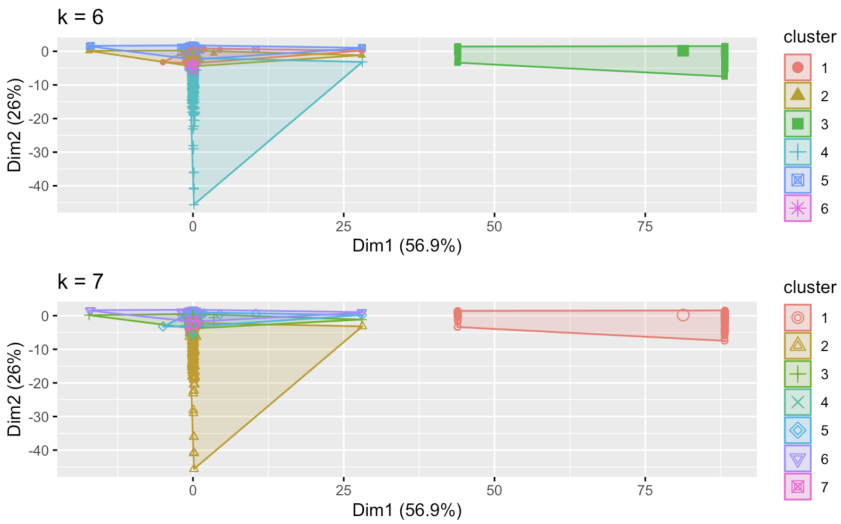
Elbow Method

1. Calculate the clustering algorithm for various k values, such as the k-means clustering. As an illustration, changing k from 1 to 10 clusters
2. We determine the total within-cluster sum of squares (wss) for each k and plot the curve of wss based on the k-fold clustering.
3. The optimal value of k is determined by where the plot bends.

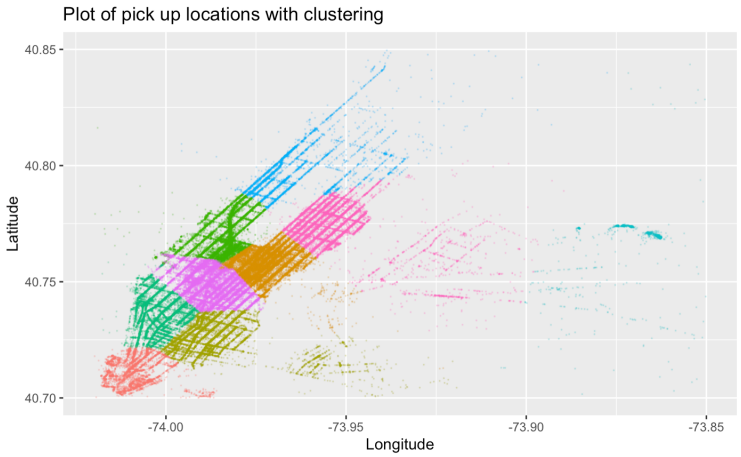


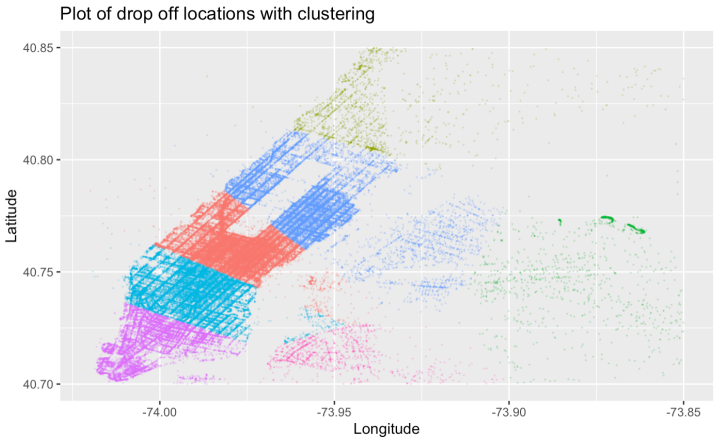

From the above plot, the optimal k values are 2 and 7. Perform k means clustering on the entire dataset.



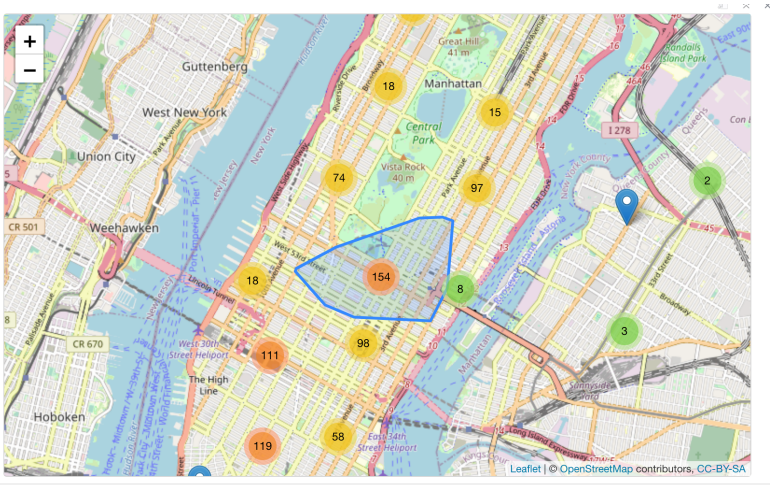



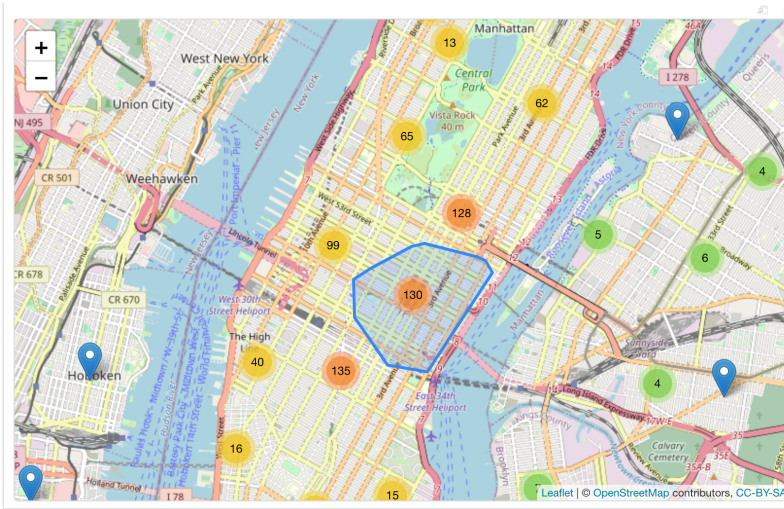
Plotting the pick up and dropoff  locations with clustering




Using the leaflet clustering algorithm, we can identify the prime pickup and dropoff locations in New York city where the demand for taxis is higher [8], [9].

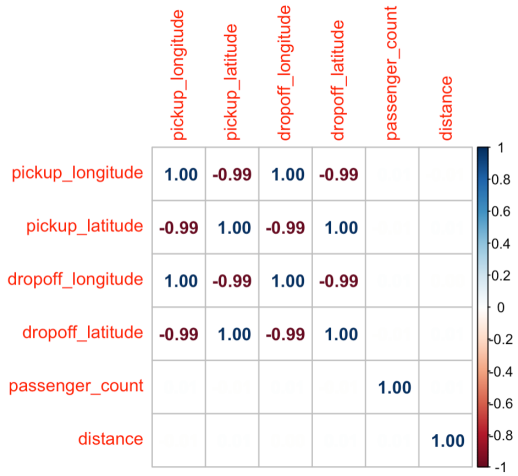



Pick Up Locations

Drop off locations

Taking a subset of the dataset(numeric columns). Performing correlation plot on the dataset between all the variables.




From the above we can say that the correlation between variables is either positive or negative.

4.2.1 Time series analysis

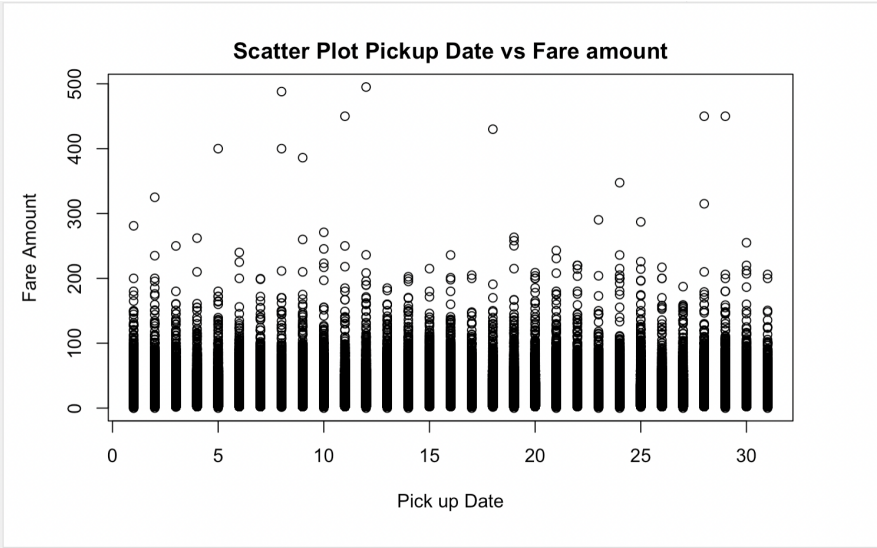
The pickup datetime feature contains various date and time components which can be used as individual features to observe impact on fare amount. We conduct feature extraction techniques and add features to existing data. Additional features include pickup year, month, day, hour and day of the week. Let's consider each of these new features and report statistical significance. To begin with, we raise concerns about fare amount such as what factors (features) of the data directly or indirectly impact the fare amount. We addressed three questions primarily for time series analysis.

(A) Does pick up date and time affect the fare?

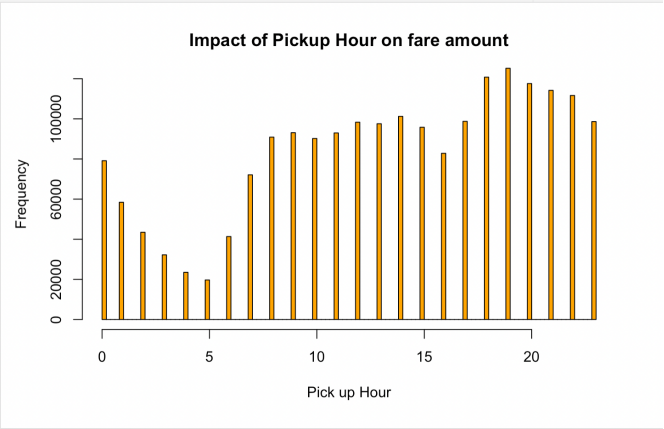
(B) Does day of the week affect the fare?

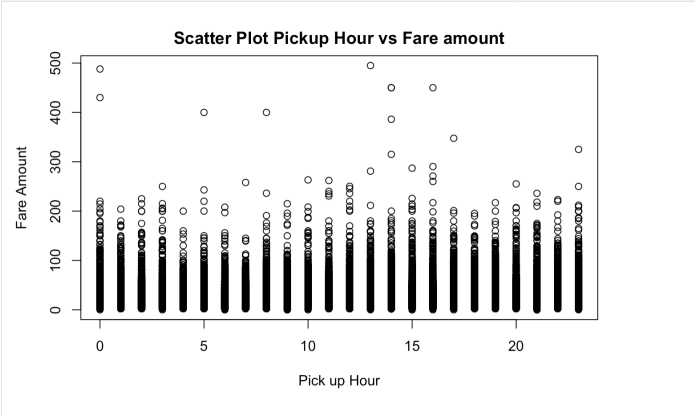
(C) Does the number of passengers affect the fare?

Consider (A) - As per the date of pickup in one month , we draw a scatter plot



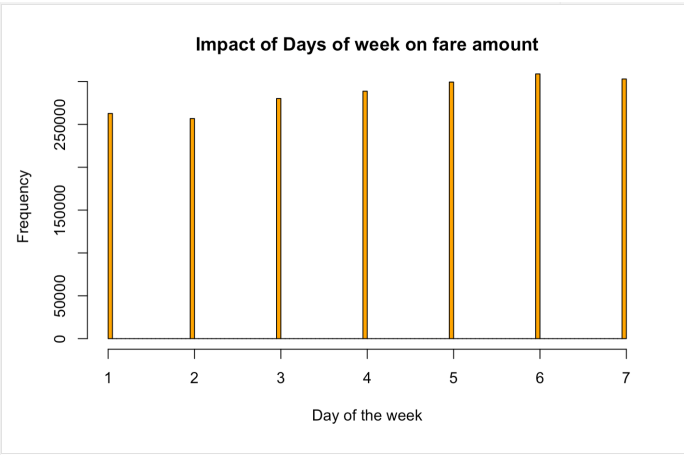
We can see that the fare amount is uniform throughout all the dates in a month , the highest fare is captured on date 12 in a month. Next, let's have a look at time (hour)

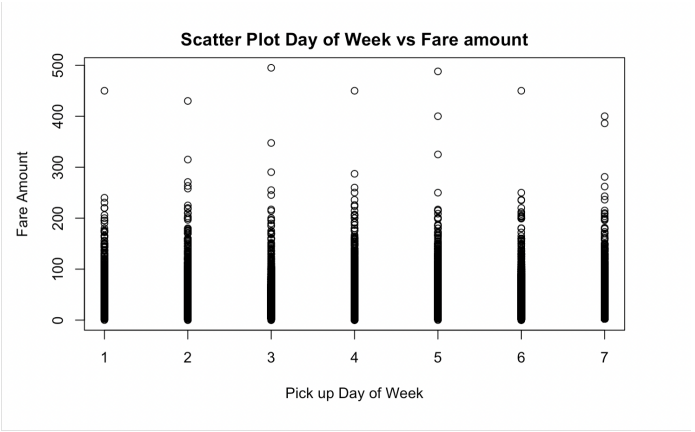



Time does play a crucial role while examining its impact on fare amount. We can see that the frequency is lowest at 5 AM and highest at 7 PM which is true as more people take taxis while leaving from work at 7 PM. However, we can also observe that the fare is higher between 5 AM to 10 AM and 2 PM to 4 PM. This might be due to people staying far away might be leaving early in the morning to avoid traffic delays.

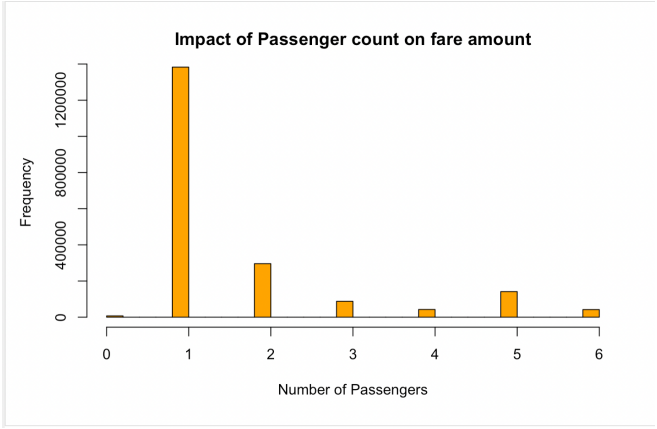
Now let's address question (B), plotting day of week against fare amount we observe that,

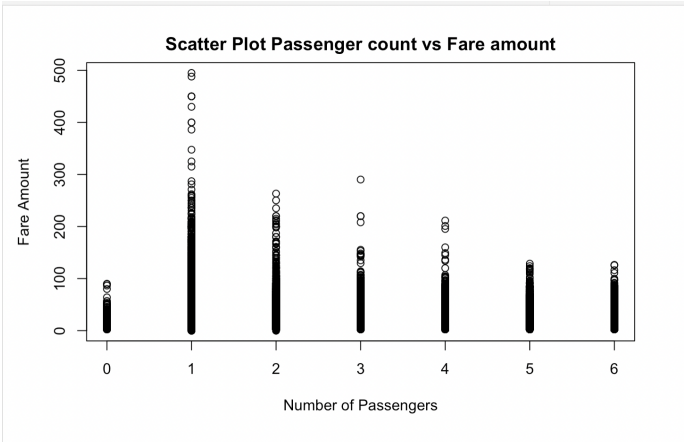



First, we can observe a similar behavior on all the days of the week. Next, the highest fares are incurred on Sunday and Monday and lowest on Wednesday and Friday. Perhaps people make longer trips (to visit relatives or friends) on Sundays and return on Monday. After a busy workweek, spend Friday at home.

Lastly, let's consider (C), passenger count impact on fare amount.



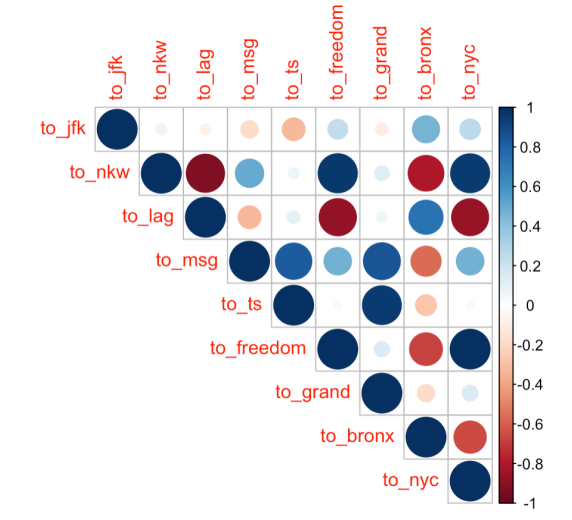

Based on the above plots, we can deduce that single passengers have a higher frequency of traveling and also contribute towards the highest fare amount for cab rides.

## 5. MODEL TRAINING

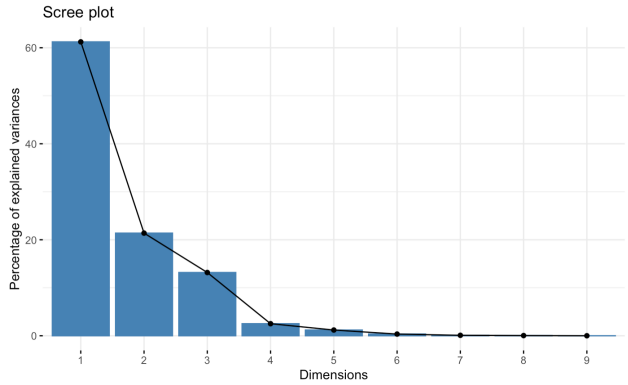
### 5.1 FEATURE ENGINEERING

5.1.1 Feature Engineering for clustering

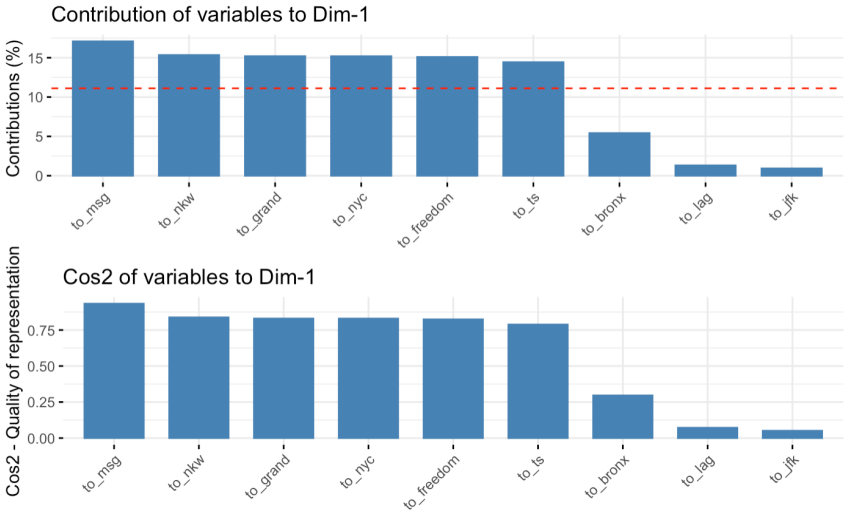
Let's add a few popular neighborhoods in the New York City area. Airports will be the most frequent sites, and we also added a few additional areas along with it. dividing the training data once more into a training set and a validation set The training set was the only data source used for the exploratory phase of my investigation. Find the haversine distance between the points and use Spearman's correlation to look up the correlation between the different variables. We can see that the answer appears to be substantially associated with the distance to Time Square, Grand Central Station, and Madison Square. Distance to Laguardia is the airport factor with the lowest correlation. Times Square and Grand Central Station seem to have a direct correlation. The distance to Newark Airport has a significant correlation with several predictors.



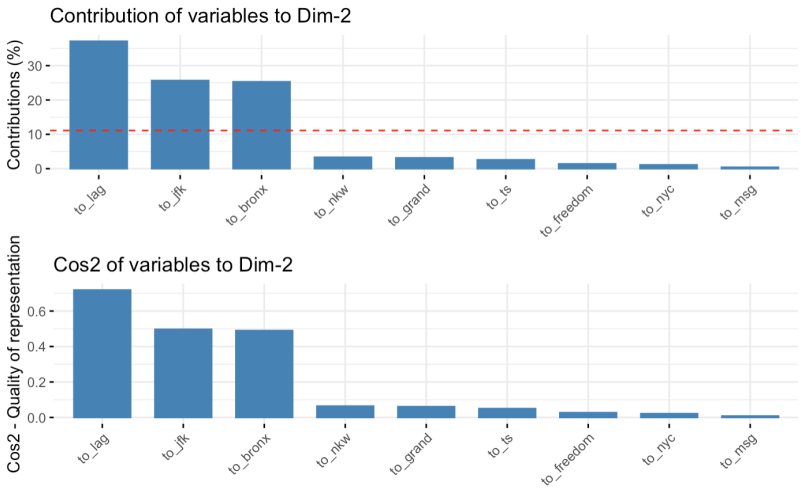
Performing Principal Component Analysis understanding the variability. Employing the package factoextra to visualize the PCA. Using the plot we examine how much variance is accounted for by the primary components. The first main component appears to explain more than 60% of the variance! By the second and third, it had dropped to almost 15%



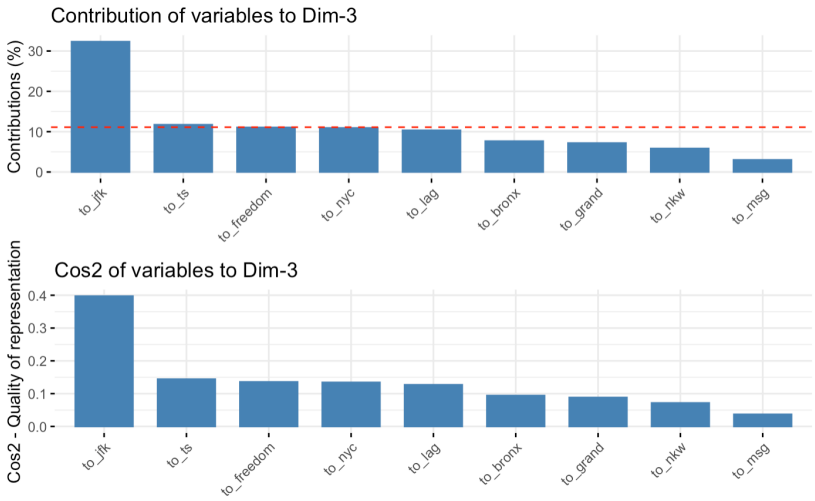

Considering the first principal component. It is primarily made up of Madison Square Garden, Newark Airport,Grand Central Station,Center of New York City, Freedom Tower and Time Square locations which are above the red line in the below graph. From the analysis we can see that more than 80% of the Madison Square Garden location is represented by the first principal component.



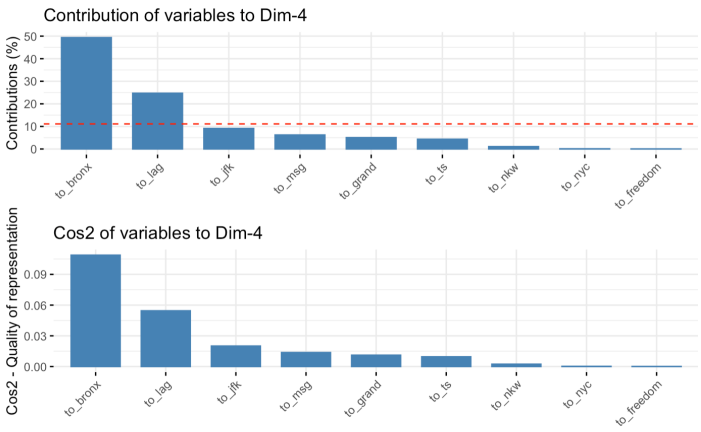

Considering the second principal component. It is primarily made up of Laguardia, JFK Airport and Bronx locations which are above the red line in the below graph which did not contribute in the first principal component. The Laguardia has the highest quality of representation of about 70% information retained.




Taking into account the third principal component.The first two are what make it up. Given how much variety the first two described, this seems logical. Using the first three principal components looks like a wise choice if we were interested in dimension reduction. We lose about 40% of the information from JFK and 15% from Time Square if we simply use the first two.




Considering the fourth principal component, It appears about 12% of information from the Bronx would be lost and 5% for Laguardia.




5.1.2 Feature Engineering for Time Series Analysis

In time series forecasting, our observations collected are dependent on the date time they are captured. For the NYC Taxi Fare dataset, we mainly consider two features - the key or pickup date time and fare amount. We start by selecting these two columns from the dataframe and convert the fare_amount (response) into a time series object in R[2].

We present two versions of time series analysis , one for year month fare amount where we group dates by month and sum the fare amount and another for hourly fare amount where we group dates and timestamps by hour and aggregate the fare amount with its sum.

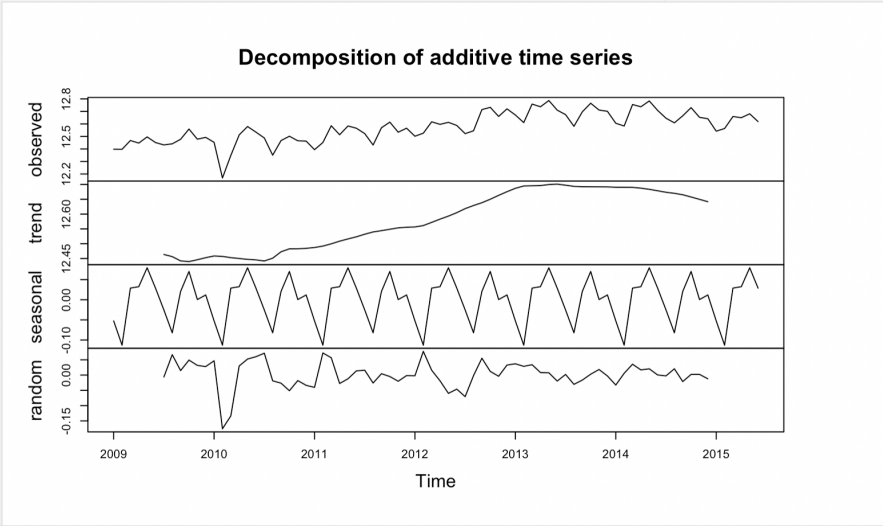
| | date_ym | fare_amount |
|---|---|---|
| 1 | 2009-01-01 | 242303.5 |
| 2 | 2009-02-01 | 242141.0 |
| 3 | 2009-03-01 | 259715.2 |
| 4 | 2009-04-01 | 254463.8 |
| 5 | 2009-05-01 | 267341.0 |

Fig. year_month dataframe

| | date_hr | fare_amount |
|---|---|---|
| 1 | 2009-01-01 00:00:00 | 266.30 |
| 2 | 2009-01-01 01:00:00 | 216.30 |
| 3 | 2009-01-01 02:00:00 | 176.30 |
| 4 | 2009-01-01 03:00:00 | 182.20 |
| 5 | 2009-01-01 04:00:00 | 186.90 |

Fig. hourly dataframe

I ] First, let's analyze the year_month data. We start by converting this data frame into a time series object with the start date as 2009-01-01 and end date as 2015-06-01 with a frequency set to be 12. Frequency is the number of observations per cycle, in our case the cycle is of one year so we set the frequency as 12. Now we have 78 observations in our time series object for monthly total taxi fares. Next, lets decompose the time series,




The above graph speaks a lot about monthly time series. Clearly there's an upward trend and we can also observe a seasonal pattern which is repeating every year. We can use these conclusions later in our model building section.

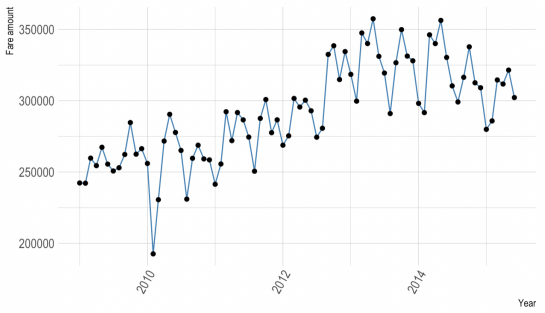



Fig. Monthly time series

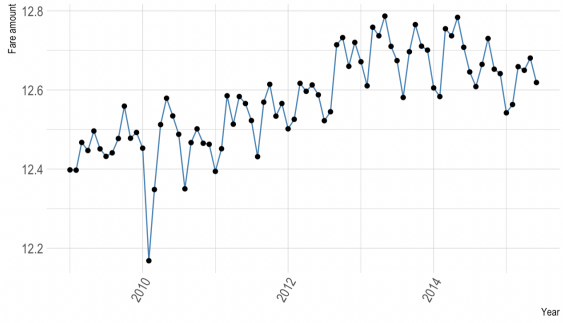



Fig. Log transformed Monthly time series

While working with the ARIMA model, we satisfy a few constraints.The time series needs to be stationary and without any seasonal component. A stationary time series is the one which has constant mean and constant variance or standard deviation. From the above graph on the left (Fig. Monthly time series) , we note two things - (a) there's an upward trend in the data (seasonal component) and , (b) overtime, the variability in the data is increasing. We need to remove both of these patterns to stationarize our time series. As the data shows changing variance over time, we take the log transform and the resulting series will be a linear time series. We will address seasonality in later sections.

The ARIMA model contains three components. The Auto Regressive (AR) component , Integration (I) component and Moving Average(MA) component. Let's see what these components are individually.

AR - Auto Regressive models consider lags , meaning we are trying to predict something for today based on its value on previous days. AR Models capture a pattern and predict the future values.

I - Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

MA - Moving Average or Rolling Mean model considers time period t impacted by unexpected external factors in previous time slots. These impacts are called as Errors or residuals and the MA model predicts the future values by considering these residuals from the past data.

As the ARIMA model requires the time series to be stationary, we utilize the ADF test of Augmented Dickey Fuller test which states a null hypothesis that the time series is not stationary and alternate otherwise. The inbuilt ADF test in R provides statistical results with p-value, and if this p-value is less than significance level 5% then our time series is stationary. The monthly time series is examined on the ADF test and we get a p-value of 0.72 which is greater than 5%. Now to make the time series stationary we introduce a differencing method (I term in ARIMA). We take the first order difference of the time series and again perform the ADF test which results in a p-value of 0.01 less than 5%. We're ready to proceed with analyzing time series with ACF and PACF plots.

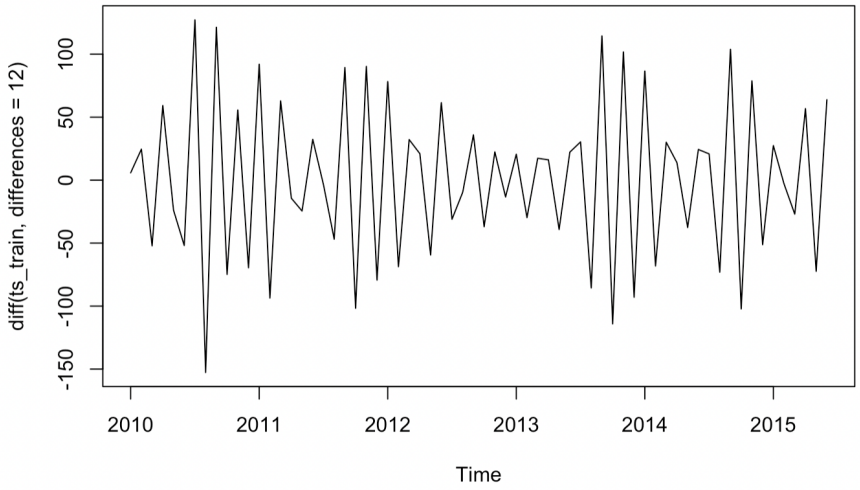



Fig. Stationary Monthly Time Series for Fare Amount

To decide the p,d and q values for the ARIMA model we visualize ACF and PACF plots. Now the big concept in time series is that the measurement of some value at a time period depends on the measurement of that value in the previous time period, at the time period before that and so on in the past. Here we're trying to find some kind of correlation between measurement of some value in current and previous timestamps. Auto Correlation Function (ACF) finds relation / correlation between these two or more timestamps. Auto Correlation Function finds correlation between multiple timestamps in an indirect and direct manner. For Partial Auto Correlation Function (PACF), it only considers direct effects between two timestamps. In our case we can ask what the fare amount was some number of periods ago and the fare amount today. The PACF plot gives us the AR component or p in ARIMA and the ACF plot gives us the MA component or q in ARIMA.

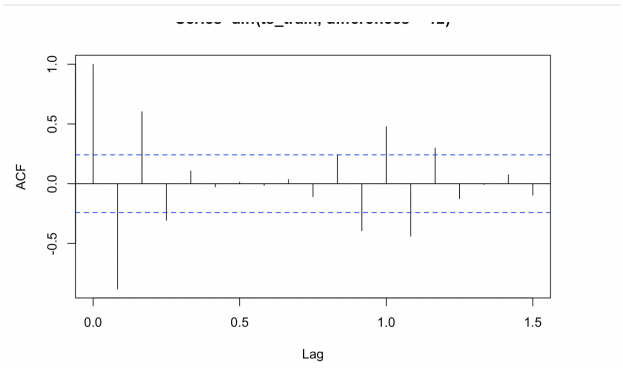



Fig. ACF plot for Monthly TS

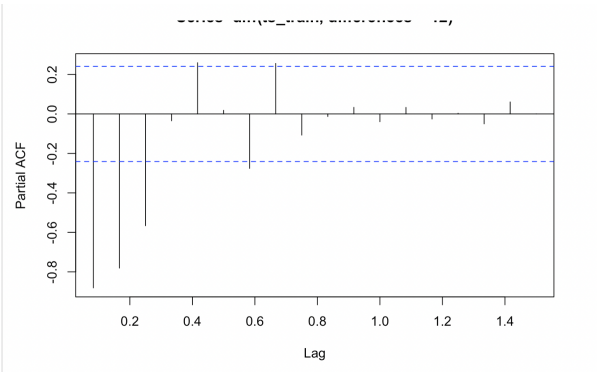
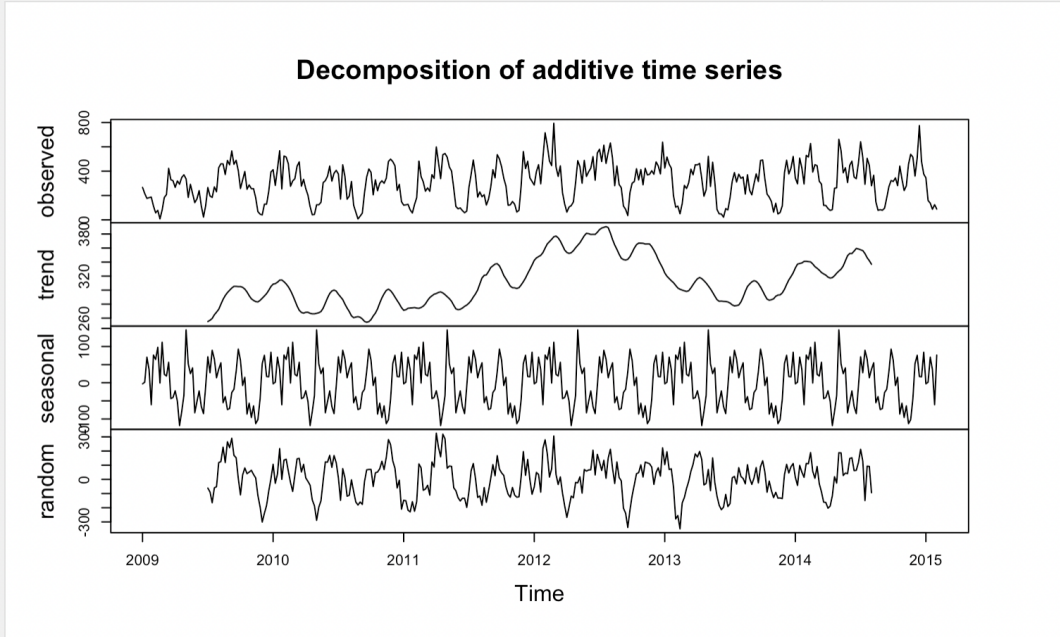



Fig. PACF plot for Monthly TS

II ] HOURLY TOTAL FARE AMOUNT

Now, let's analyze and follow the same steps for hourly data for fare amount. We convert the hourly data frame into a time series object with a frequency of 60 (hourly). There are total of 366 observations which we decompose as follows:




Clearly there is a trend and seasonal pattern in the hourly time series as well. Moving on to check stationarity, we plot the time series.

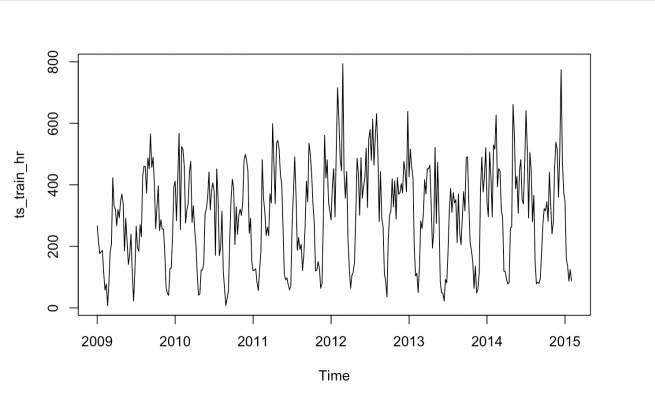



Fig. Hourly time series

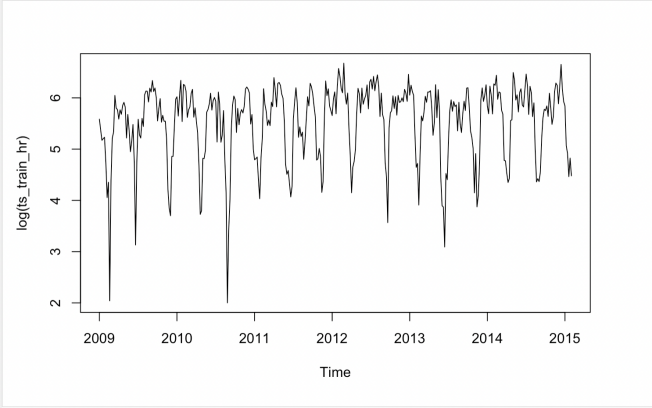



Fig. Log Transformed hourly time series

Looking at the hourly time series we can observe a seasonal trend and we take the log transformed version of this series to make it stationary. We perform the ADF test and get a p-value of 0.1 with first order differencing. Let's have a look at the ACF and PACF plots.

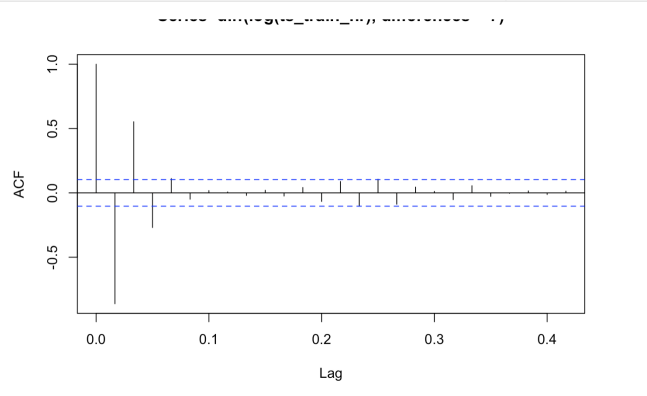



Fig. ACF plot for Hourly TS

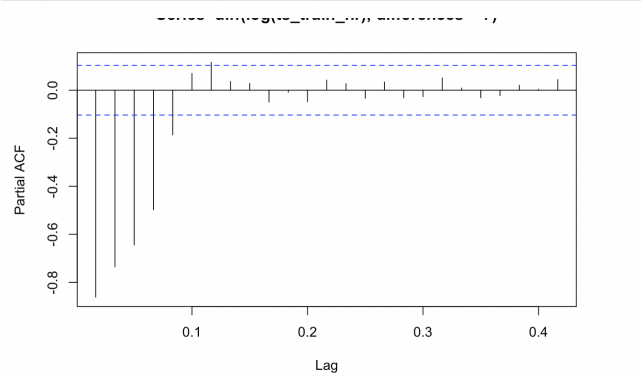



Fig. PACF plot for Hourly TS

The ACF and PACF plots will help us decide the p and q values in ARIMA. We will also have a look at the Seasonal ARIMA - SARIMA model in the section 5.3 Model Selection. We have employed the Auto Arima technique which will provide us optimal p,d,q values for ARIMA and SARIMA models[6].

**5.2 EVALUATION METRICS**

5.2.1 Time Series Evaluation Metrics

There are four evaluation criteria we're testing our models on

(i) MLE - Maximum likelihood : By identifying the parameter values that maximize the chance of making the observations given the parameters, the MLE approach estimates the parameters of a statistical model given observations. In time series forecasting, the higher the MLE the better the model

(ii) AIC - Akaike's Information Criterion : The AIC provides information on how well our model fits the time series without overfitting it. The lower the AIC score, the better is the model.

(iii) Root Mean Square Estimate - RMSE:For time series forecasting, we should aim to reduce RMSE as low as possible to get the best model.

(iv) Mean Absolute Percentage Error - MAPE: MAPE is a loss function which is used to normalize errors by inverse of true observation value. We aim to select a model with a lower MAPE score[7].

**5.3 MODEL SELECTION**

5.3.1 Time Series Model Selection

Model selection for time series forecasting depends on various factors such as the ACF and PACF plots, can the fare amount be predicted just by AR or MA model or do we need to integrate them both into ARIMA and do we consider Seasonal pattern using SARIMA model. In the ACF and PACF plots the region inside the blue dotted lines is the non-significant region meaning those lags are not statistically significant and these plots converge to 0.

I] MONTHLY FARE AMOUNT FORECASTING

First let's have a look at ACF and PACF plots for monthly time series for fare amount. Deciding the values for p and q is important. From Fig. PACF for monthly TS, we can see that lag 1 and lag 2 are significant and then it starts decaying to 0. We create the first AR model with order p= 1, d=1 and q=0 and get a MLE of 85.18 and AIC score of -166.35. Next we proceed to the MA model , observing Fig.ACF for Monthly TS, we observe the same pattern with lag 1 and 2 being significant. Training time series on the MA model with p=0,d=1 and q=1 yields a MLE of 88.4 and AIC of -172.79. Based on AIC, the MA model is better than AR. Next, we go for the ARIMA model with p=1,d=1 and q=1 which results in MLE of 92.84 and AIC = -179.68.

Moving on to the next set of parameters with p=2 , d=1 and q=2 , results in an ARIMA model with MLE 99.1 and AIC of -188.2. This seems to be a good model. Let's forecast values by splitting the time series into train and test sets with 85% and 15% train test distribution . We use the forecast() package in R to forecast train samples and use predict() to predict test samples. Calculating the evaluation metrics we get Train RMSE = 0.068, MAPE= 0.0039 and Test RMSE = 0.80 , MAPE =0.0052.

We have also implemented the auto arima functionality to provide us with optimal parameters for the SARIMA model. After executing auto arima , we get order of (0,1,2) and seasonal order of (2,1,1). Let's see how our time series fit on a SARIMA model. Executing SARIMA, we get the train RMSE = 0.042, MAPE =0.0022 and test RMSE= 0.108, MAPE=0.0078. This SARIMA model is our final optimal model with prediction graphs mentioned in the next section(6 MODEL VALIDATION).

II] HOURLY FARE AMOUNT FORECASTING

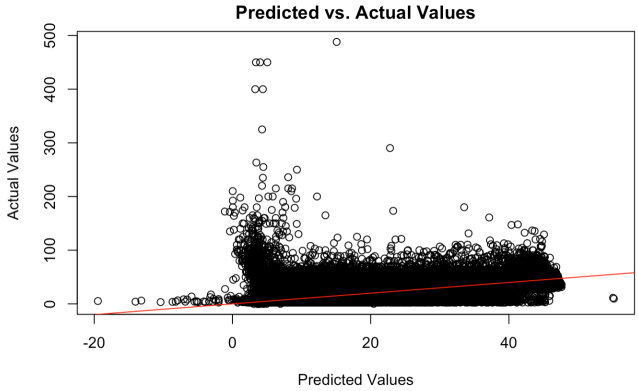
Consider Fig. PACF plot for hourly TS the lags starts decaying after lag 3, so we can consider lag1 and lag 2 for AR components. Now consider Fig. ACF Plot for hourly TS, lag 1 and lag 2 seem to be significant which relates to our MA components. We do not require differencing for the hourly time series as it is stationary after the logarithmic transformation, hence d = 0. We set p = 1, d =0 and q=1 for the ARIMA model and it generates a MLE of -252.12 and an AIC score of 512.25. Next we have implemented the auto arima model and it produced optimal parameters with order of (2,0,1) and seasonal order of (2,0,0). This model had an MLE score of -240.72 and AIC score of 494.44. We can use this

model to forecast hourly fare amounts. For this optimal model we achieved Train RMSE= 0.480, MAPE=0.0729 and Test RMSE= 0.5976, MAPE = 0.0947.

## 6. MODEL VALIDATION

6.1 Linear Regression Model

Building the linear model on the entire dataset, and predicting the fare amount based on the train dataset. The RMSE score for the linear model is 4.2, MSE score is 17.8 and R-square is 0.68 [10].




6.2 Time series Forecasting

I] MONTHLY FARE AMOUNT FORECASTING RESULTS

Initially we constructed a ARIMA model (Model 1) with parameters (2,1,2) we achieved the following results,

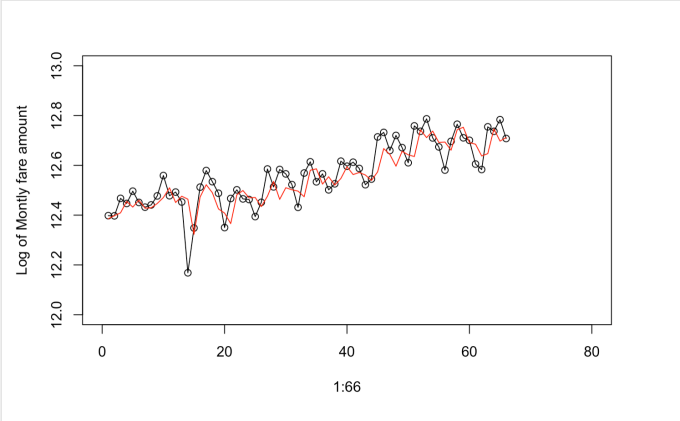



Fig. Train forecasts (red) for Monthly Fare Amount - Model1

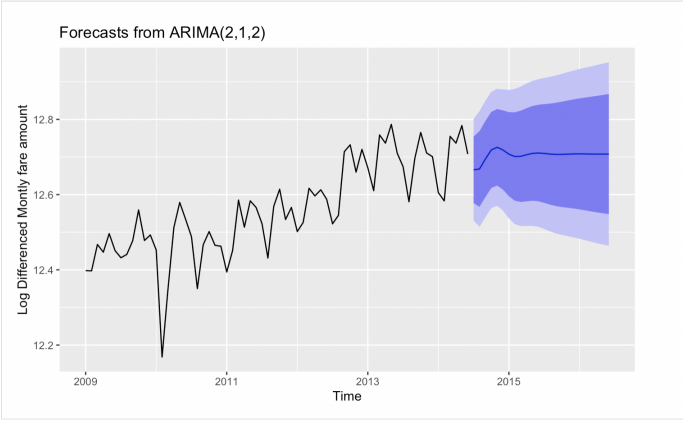



Fig.Test forecasts(blue) for Monthly fare amount - Model 1

Based on the above graphs, we can deduce that the model was able to fit train forecasts but looking at the test predictions, it is not accurate as we have not added the seasonal component to it. The test predictions follow the similar variance but the trend is not present. Let's have a look at our optimal SARIMA model (Model 2) forecasts.

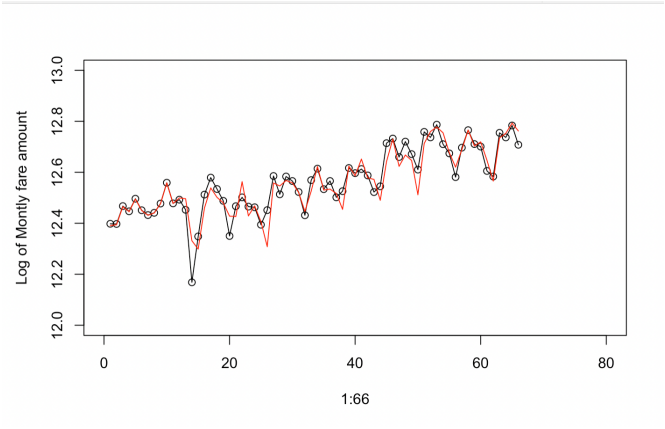



Fig.Train forecast(red) for Monthly fares - Model 2

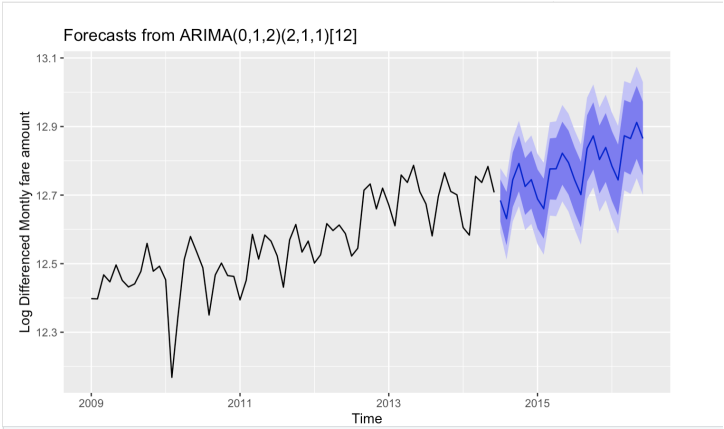



Fig.Test forecast for Monthly fares - Model 2

From the above plots, we can infer that Model 2 - SARIMA was able to fit the time series appropriately. Considering the Test predictions, we can observe that the SARIMA model captures trends and patterns in the time series with a lower level of variance.

II] HOURLY FARE AMOUNT FORECASTING RESULTS

For hourly fare amounts, we ran the auto arima model (Model 3) with order(2,0,1) and seasonal order (2,0,0). The following results were obtained,

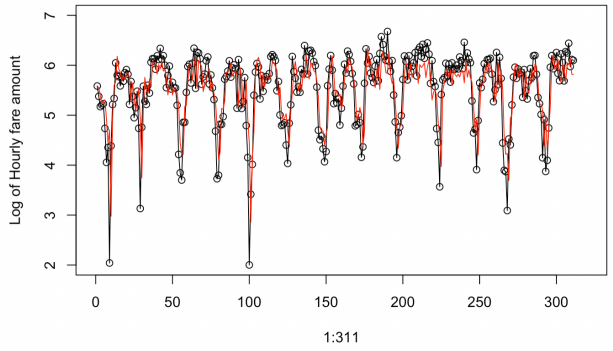



Fig.Train forecast(red) for Hourly fares - Model 3

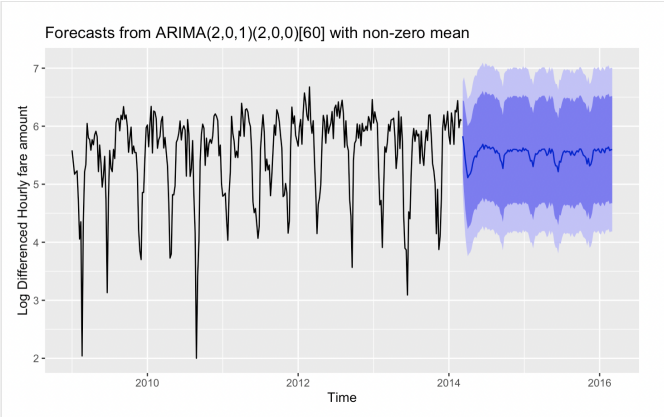



Fig.Test forecast(blue) for Hourly fares - Model 3

We can observe that this model fits the train data really well. For the test forecast, it is able to capture the trend and predict the test estimates.

## 7. CONCLUSION

As we performed predictive analysis, we realized that time series models are better at forecasting future taxi fares when compared to linear models. There's definitely an upward trend present in the dataset which comes from the fact that taxi prices are shooting up on a yearly basis. There's a high amount of taxi fare ratio for single passengers. The taxi fares are highest during early mornings and evenings as people travel to work and airports or famous locations during this time. We observed crucial

factors impacting taxi fares during month of year where people tend to travel more during vacations and holiday seasons and the demand for taxis increases. The time series model we built was able to predict monthly fare amounts accurately. The pickup and drop off location features added for cluster analysis, such as famous locations in NYC - airports, parks and tourist attractions, also contribute towards the hike in fare amounts. Outliers related to distance features were identified and removed for further analysis.

For the future scope of the project, we would like to explore the correlation between location and timestamps collectively. Considering these features, we would be able to draw conclusions for optimal routes with optimal taxi fares. We would also like to explore the impact on traditional taxis by considering taxi pool services like Uber and Lyft. We would extend the time series forecasting model to predict the exact or approximate taxi fare for a given date and time in the future.

## 8. DATA SOURCES

Our source dataset for this project is available on Kaggle Open Source ML Dataset Repository:
https://www.kaggle.com/c/new-york-city-taxi-fare-prediction

## 9. SOURCE CODE

Our source code is available on github , following the link :
https://github.com/yashgupte21/csp571_dpa_final_project.git

## 10. BIBLIOGRAPHY


[1] Dama, Fatoumata and Sinoquet, Christine. “Time Series Analysis and Modeling to Forecast: a Survey”. 10.48550/ARXIV.2104.00164

[2] A. Ian Mcleod and Hao Yu and Esam Mahdi. “ Time Series Analysis with R”

[3] Antoniades, Ch., Delara Fadavi and Antoine Foba Amon. “Fare and Duration Prediction : A Study of New York City Taxi Rides.”

[4] Y. Zhou, Y. Wu, J. Wu, L. Chen and J. Li, "Refined Taxi Demand Prediction with ST-Vec," 2018 26th International Conference on Geoinformatics, 2018, pp. 1-6, doi: 10.1109/GEOINFORMATICS.2018.8557158.

[5] Analyzing 1.1 Billion NYC Taxi and Uber Trips, with a Vengeance. Available at : https://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

[6] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. (2nd ed.) OTexts. https://otexts.org/fpp2/

[7] Cerqueira, V., Torgo, L. & Mozetič, I. Evaluating time series forecasting models: an empirical study on performance estimation methods. Mach Learn 109, 1997–2028 (2020). https://doi.org/10.1007/s10994-020-05910-7

[8] Leaflet.heat. Available online: https://github.com/Leaflet/Leaflet.heat (accessed on 30 June 2019).

[9] Leaflet—A JavaScript Library for Interactive Maps. Available online: http://leafletjs.com/ (accessed on 5 February 2019).

[10] Yan, X., and X. G. Su. Linear Regression Analysis: Theory and Computing. World Scientific Publishing Company, 1st edition, 2009.