NYC Green Taxi Analysis for Sept. 2015
A hobby project with SAS

Parag Guruji, Purdue University, West Lafayette, IN, USA

Email: pguruji@purdue.edu Cell: +1-765-775-8727

Step 1: Programmatically download and load into your favorite analytical tool the trip data for September 2015

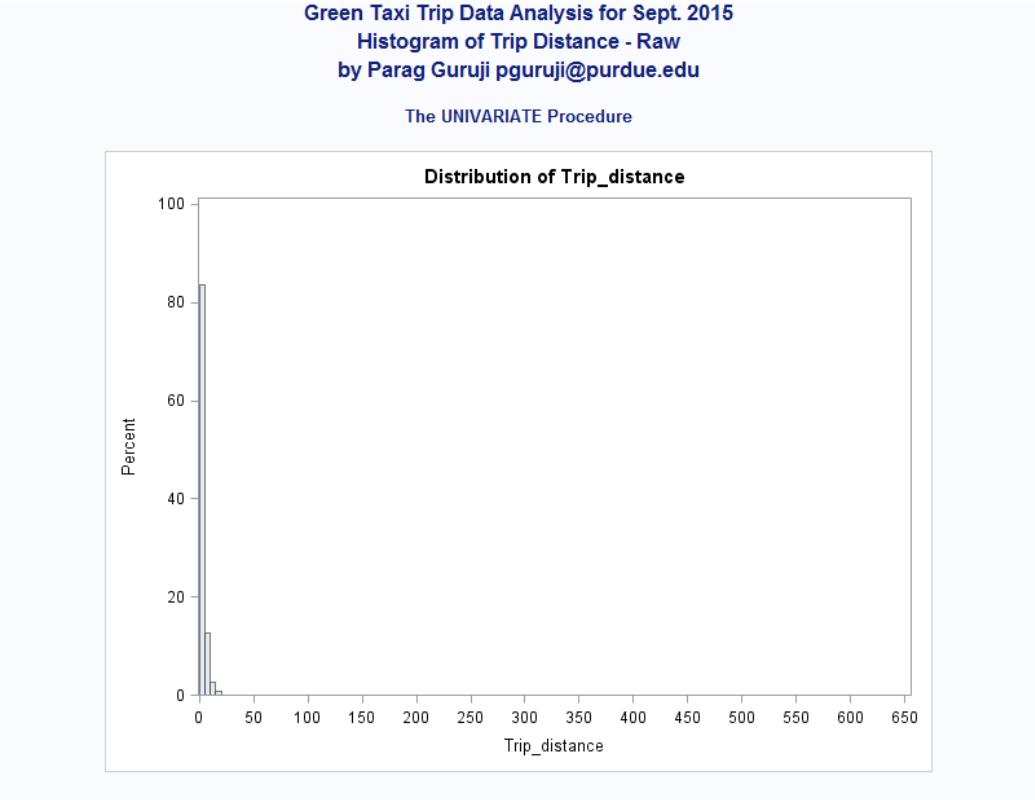
URL: https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata_2015-09.csv

SAS Log: NOTE: The data set WORK.RAW_DATASET has 1494926 observations and 21 variables.

The Dataset loaded from the source URL has 1494926 rows and 21 columns.

A look into Trip Distances:

Histograms: 1. Raw Data simply plot: Due to high skew and outliers, this plot is of little help in understanding any trend in the trip distances. Hence we plot further 2 histograms:



2. Here's a histogram that kind of zooms in into the previous one but ignores the extreme 1% of the data. It plots 99% quantiles of trip distances from 0 to 15 miles.

We observe from the histogram and the numerical summary of Trip_distance that:

1. The distribution is heavily right skewed

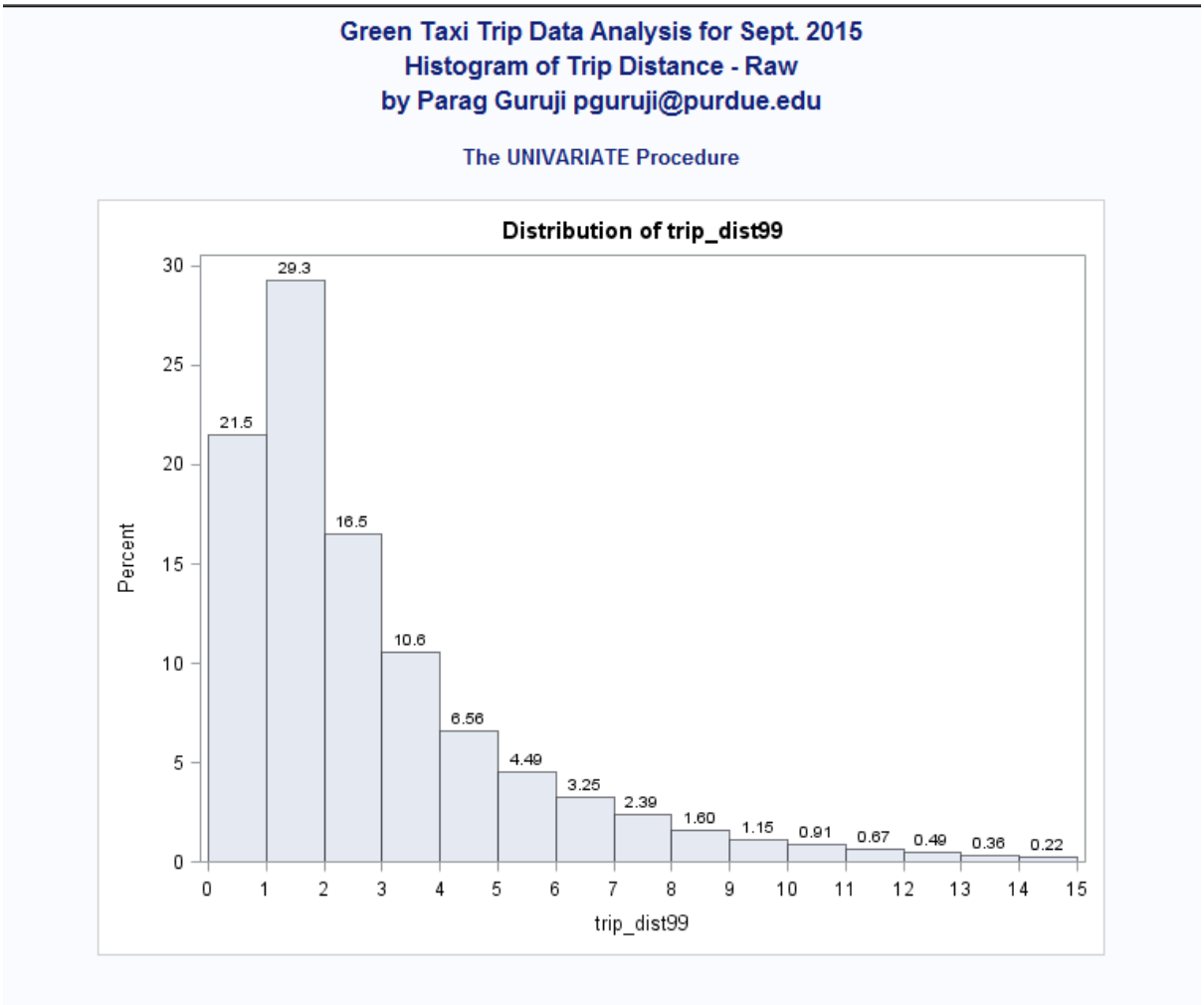
2. The first 99% Quantiles are in range 0 to 14.77 and last 1% quantiles in 14.77 to 603.10

3. by the AIRPORT TRIPS ANALYSIS + 1.5(IQR) measure, all observations above 7.7 can be termed as outliers which is more than 10% of the data

For the practical reason of facilitating a closer look at the shape of distribution of the maximum possible chunk of the data (99% quantiles),

we eliminate the extreme right 1% quantiles to get a derived dataset for plotting a histogram.

Assumption: The distance record of 0 is assumed to be a distance that was too small to be recorded but not exactly 0. Hence not ignoring it so as to see the effect of very small trips.



Observation: A clear pattern of logarithmic decay in the percentage (and hence in the number) of trips along with linear increase in the trip distance is visible.

We can see that half the trips are shorter than 2 miles while three fourth are shorter than 4 miles.

To visualize this better, we plot our next histogram on transformed data:

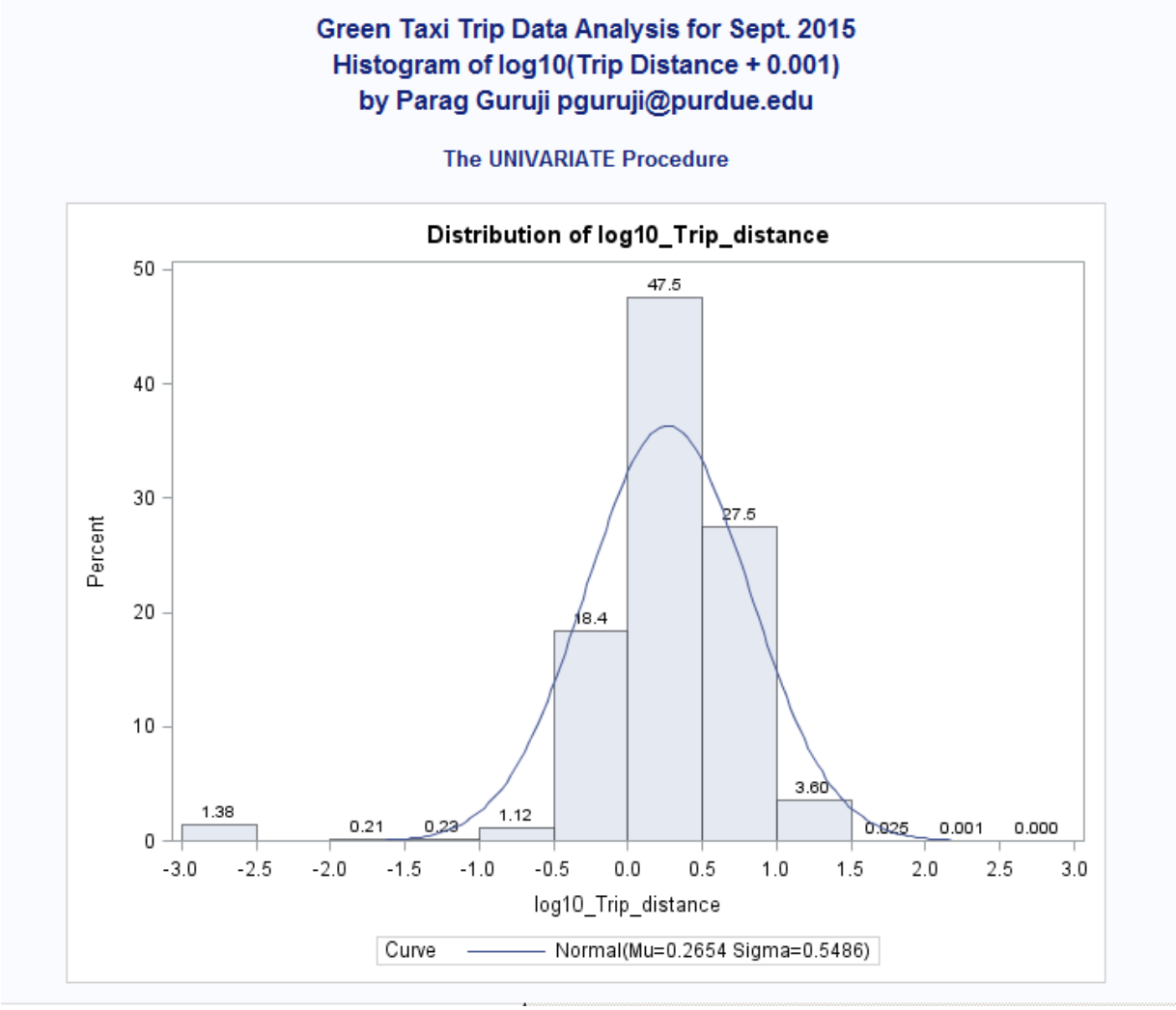
3. Log-transformed Distance:

We observed that the Trip_distance variable takes values over a range of multi-order magnitudes from min. 0.01 to max. 603.1 (= 60310*min)

The histogram of subset comprizing of first 99% quantiles also shows logarithmic fall in percent frequency over Trip_distance

Thus, it is of interest to log-transform the data and explore the resultant shape.

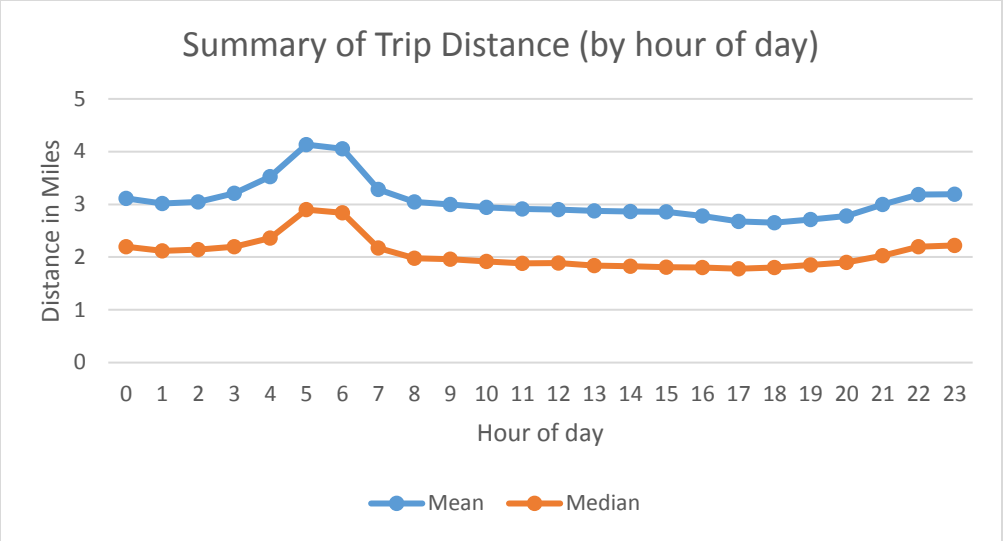
Hence, we create a new dataset 'logified_trip_distance' where 'Trip_distance' is log transformed as 'log10_Trip_distance' = log10(Trip_distance + 0.001)



Does the trip distance vary at different hour of day?

Summary of hourly mean trip distance in miles

Obs	hour_of_day	hourly_mean	hourly_median
1	0	3.1152760654	2.2
2	1	3.0173471817	2.12
3	2	3.0461755996	2.14
4	3	3.2129453224	2.2
5	4	3.5265550257	2.36
6	5	4.1334742515	2.9
7	6	4.0551488949	2.84
8	7	3.2843944447	2.17
9	8	3.0484495887	1.98
10	9	2.9991052284	1.96
11	10	2.9444823206	1.92
12	11	2.9120154602	1.88
13	12	2.9030647783	1.89
14	13	2.8782944482	1.84
15	14	2.8643042722	1.83
16	15	2.8570399989	1.81
17	16	2.7798515608	1.8
18	17	2.6791138579	1.78
19	18	2.653222068	1.8
20	19	2.7155968837	1.85
21	20	2.7770517156	1.9
22	21	2.9991886114	2.03
23	22	3.1853935423	2.2
24	23	3.191537941	2.22



There are 3 airports in NYC area. In sept 2015, (UNGA session & Pope's visit!) how were the trips to these airports look like?

Count of airport trips: 25293

Intent: To identify a trip as an airport trip and derive following pieces of information-

1. Is airport trip? – **Assumption: The trip which either starts or ends within the radius of 1 Km from the location point of either of the 3 NYC area airports.**

   The location point is specified by their geo-coordinates found on Wikipedia

2. Which airport/(s): JFK (**JFKA**), Newark Library (**NLA**) or La Guardia (**LGA**)
3. In-bound and out-bound locations as one of the 4 options:

   JFKA, **NLA**, **LGA,** and **OTHER**

   Using OTHER keyword keeps the work (both code as well as data) reusable for analysis with more locations in future by encoding them and replacing corresponding entries of OTHER with new code

For this, we create 3 varables:

1. airport: {Y, N} - is this an airport trip? Y=yes, N=no
2. pickup: {JFKA, LGA, NLIA, OTHER} - specifies pickup location
3. dropoff: {JFKA, LGA, NLIA, OTHER} - specifies pickup location

Opinion: To identify peculiar characteristics of the airport trips, it is important to view them in contrast to the non-airport trips.

Following is a summary of few interesting statistics of airport trips vs non-airport trips.

The data was found to have many inconsistencies w.r.t. various variables. However, we have not cleaned the data except w.r.t. the variables directly used in this part of the analysis. Especially for the Tip variable – which is not recorded reliably anywhere except the trips with payment type 'Credit Card' (Code 1), the results shouldn't be interpreted without investigating more into reasons for inconsistencies and separation of erroneously recorded data and the outliers.

Green Taxi Trip Data Analysis for Sept. 2015
Summary of Characteristics: Airport Trips VS Other Trips
by Parag Guruji pguruji@purdue.edu

The MEANS Procedure

airport	N Obs	Variable	Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Median	Mode
N	570916	Fare_amount	15.2398456	15.2149917	15.2646994	12.5000000	7.0000000
		Trip_distance	3.8203719	3.8124015	3.8283424	2.8600000	1.0000000
		Tip_amount	2.7731721	2.7663828	2.7799615	2.2600000	0
		Fare_per_distance	4.5107905	4.5075560	4.5140250	4.3689320	5.0000000
		Tip_as_percent_fare	18.7552932	18.6896664	18.8209201	21.0000000	21.0000000
Y	16312	Fare_amount	30.1587365	29.9095056	30.4079675	28.0000000	52.0000000
		Trip_distance	9.6030266	9.5143150	9.6917382	8.4600000	2.8000000
		Tip_amount	5.9042502	5.8381448	5.9703557	5.2700000	0
		Fare_per_distance	3.2967496	3.2843298	3.3091693	3.1620553	3.3333333
		Tip_as_percent_fare	19.6382418	19.4494708	19.8270128	20.0000000	20.0000000

Now, as we observe the values in the above table, the stark differences are evident. However, if they are statistically significant or not can be checked by the two-sample t-test.

In this test, we compare the characteristic of interest – in our case – say, the average fare amount of two independent samples – here, the airport trips and the non-airport trips:

The results are as shown below:

Two sample t-test on mean Fare amount Airport Trips vs Other Trips

The TTEST Procedure

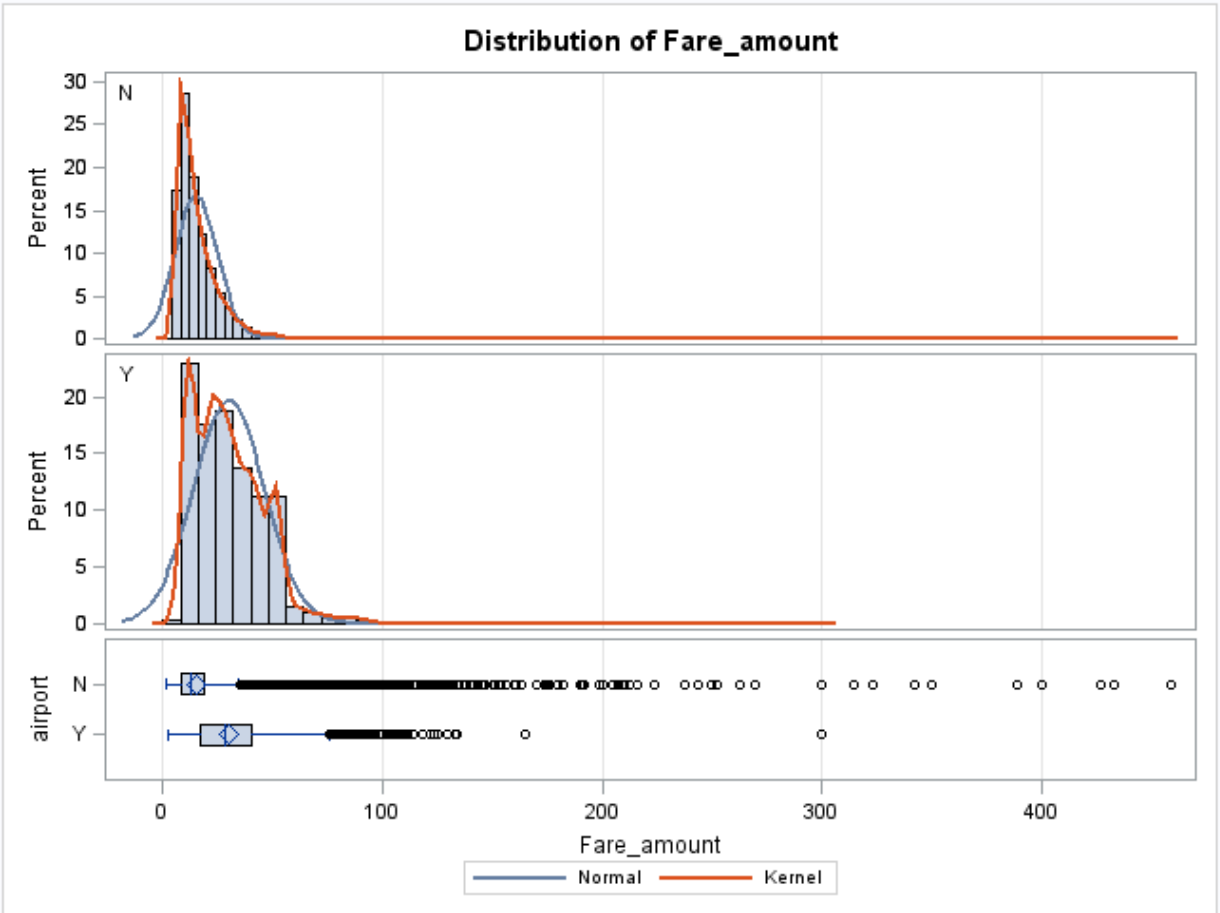
Variable: Fare_amount

airport	N	Mean	Std Dev	Std Err	Minimum	Maximum
N	570916	15.2398	9.5815	0.0127	1.0000	459.0
Y	16312	30.1587	16.2396	0.1272	2.5000	300.0
Diff (1-2)		-14.9189	9.8275	0.0780		

airport	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
N		15.2398	15.2150	15.2647	9.5815	9.5639	9.5991
Y		30.1587	29.9095	30.4080	16.2396	16.0653	16.4178
Diff (1-2)	Pooled	-14.9189	-15.0718	-14.7659	9.8275	9.8097	9.8453
Diff (1-2)	Satterthwaite	-14.9189	-15.1694	-14.6684			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	587226	-191.17	<.0001
Satterthwaite	Unequal	16637	-116.75	<.0001

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	16311	570915	2.87	<.0001



Here, we are actually trying to check that what is the probability that the apparent difference in the mean characteristic of our sample is in fact just occurred by chance and in fact is not significant when thinking about the whole population. The p-values (Pr> |t|) tell us this probability.

The variances of the samples are clearly significantly different ((Pr>F) < .0001).

The Mean fare amounts are also clearly significantly different (Pr> |t|) < .0001

Thus, we can claim so far with 95% confidence (α = 0.05) that the fare amount is significantly larger for airport trips than other trips (Pr>|t|)/2 << α

But, on a closer look, the airport trips are by far longer than other trips. Hence, is the difference in fare is just a reflection of difference in distance, or is it really an effect of these trip being 'airport trips'?

Hence, we constructed a derived variable **Fare_per_distance** which actually appears to have lower value for the airport trips than other trips. On performing the same t-test for the **Fare_per_distance**

Two sample t-test on mean Fare/distance Airport Trips vs Other Trips

The TTEST Procedure

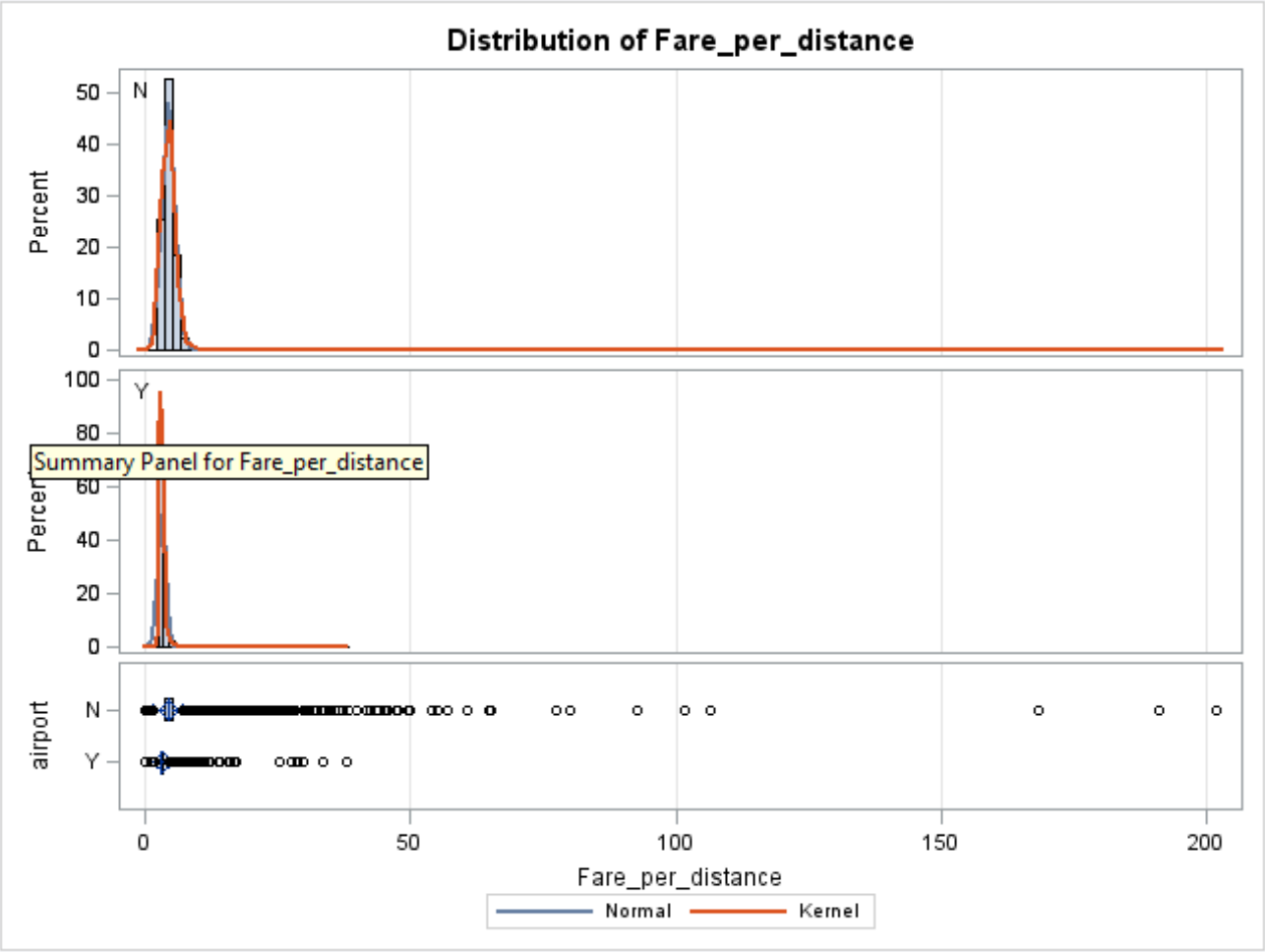
Variable: Fare_per_distance

airport	N	Mean	Std Dev	Std Err	Minimum	Maximum
N	570916	4.5108	1.2469	0.00165	0.1025	201.6
Y	16312	3.2967	0.8093	0.00634	0.1429	38.0000
Diff (1-2)		1.2140	1.2369	0.00982		

airport	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
N		4.5108	4.5076	4.5140	1.2469	1.2447	1.2492
Y		3.2967	3.2843	3.3092	0.8093	0.8006	0.8181
Diff (1-2)	Pooled	1.2140	1.1948	1.2333	1.2369	1.2347	1.2391
Diff (1-2)	Satterthwaite	1.2140	1.2012	1.2269			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	587226	123.61	<.0001
Satterthwaite	Unequal	18597	185.42	<.0001

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	570915	16311	2.37	<.0001



As it is clearly evident from the test outcomes and the distribution, here,

We can claim with 95% confidence that the average Fare per mile happens to be significantly greater for non-airport trips as opposed to the airport trips in the NYC.

This is interesting finding because although airport trips are more expensive in total fare, they don't result in higher fare per mile and are rather cheaper compared to other trips!!

Further investigation in the causality of this finding will need more data w.r.t. the zones through which the trip passes, traffic density, etc.

Also, given more time, I'd like to break this data down for each airport and analyze them separately – especially for international vs. domestic airports. I'd also want to see the variation w.r.t. weekdays vs weekends and holidays.

Let's try designing a derived feature for tips given and then identify a factor to predict that variable based on the factor identified!

1. Feature design

Definition of total fare:

Total Fare = Fare + Extra + Taxes + Surcharge + Fee + Tolls = Total amount - Tip amount

Definition of derived variable: Tip_as_percent_total_fare = 100 * Tip_amount / Total_fare

2. Predictive modeling:

Intent:

Identify the locality of origination of a trip and based on that, predict how much percent of the fare amount is likely to be awarded.

Reasoning / Assumptions for choice of area as the predictor:

There are certain areas with higher likelihood of getting higher tips as can be seen in the figure above. Whereas there are some other areas with likelihood of low percentage of fare to be awarded as tip.

Assumption:

In large and diverse cities like NYC, the residences and regular visiting places of the people are geographically clustered in correlation with their affluence – which further is correlated with the percent of fare they may be willing to award.

Thought:

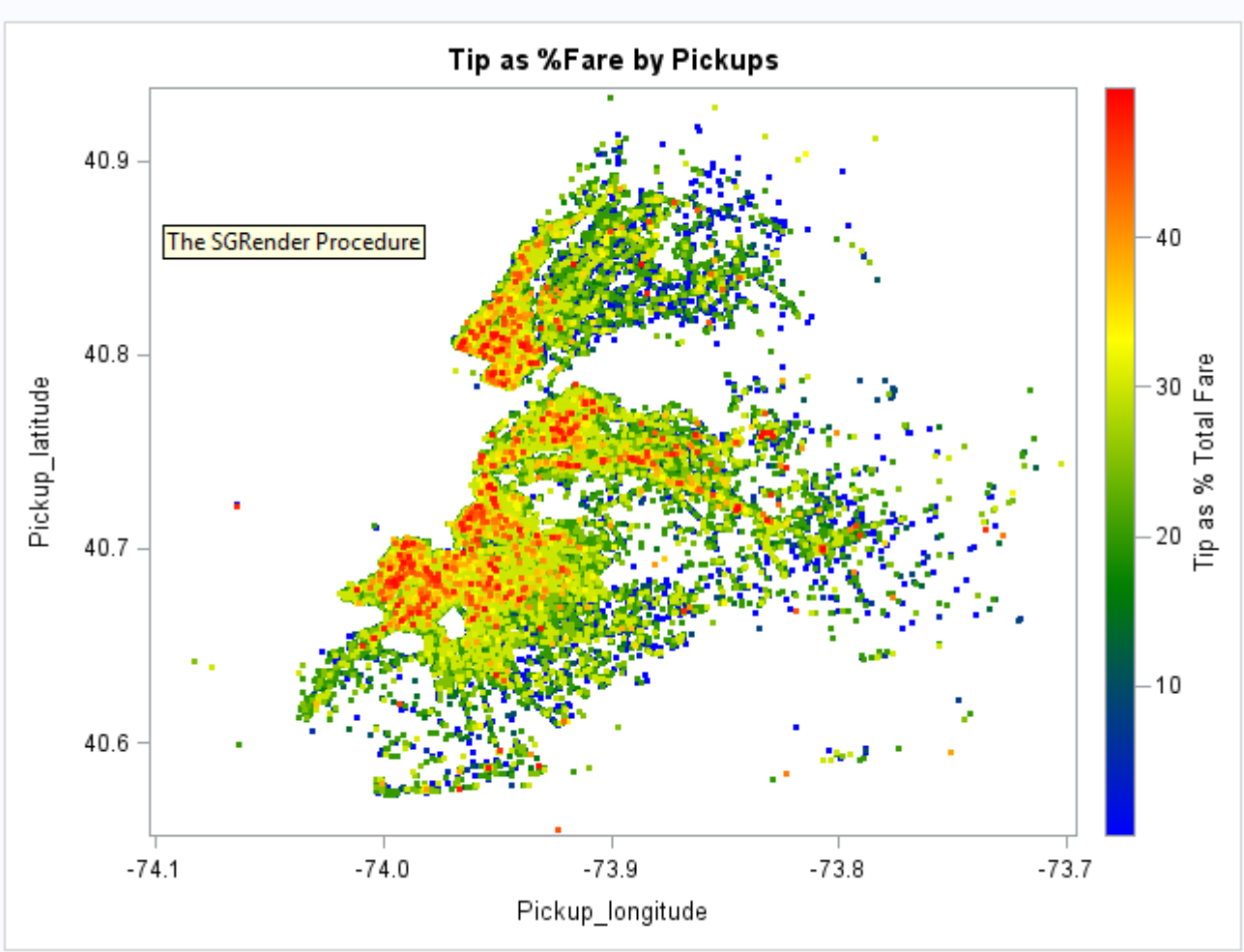
Although there are many other factors such as the fare amount itself **my intention here was to explore the variation in the tip irrespective of peculiarities of that particular trip-transaction.** To that end, the possible features I'd like to incorporate in my model given further time and resources are – **the specialty of the region of pickup and drop off – e.g.: How the tip percentage varies for Hotel trips, Office trips, Home trips, trips to the Entertainment zones such as clubs, theatres and exhibitions, shopping mall trips, day (working, weekend, etc.), time (morning, evening, night) of trip etc**. One such feature is developed in previous exercise – the Airport trips. But for the fear of diminishing the geographical scope of problem and since we have around only 25000 data-points for airport trips, I kept that idea aside for now.

Exploratory Data Analysis with Scatterplots

The scatterplots for the considered potential predictors other than the proposed one are given in Appendix-1 at the end of the document.

The Area map of training data color-coded as per Tip_as_percent_total_fare is shown below.

Green Taxi Trip Data Analysis for Sept. 2015
Location Map of Tip as Percent of Fare as per Pickup Points
by Parag Guruji pguruji@purdue.edu
CAUTION: the data is sorted by Tip_as_percent_total_fare. Hence, the overlaping points showing higher-end color may contain data for lower-end color underneath



Failed Approach: Looking at the area map of training data, it is tempting to predict the tip as percent of fare based on the geographical Euclidian distance from the urban center of Manhattan since the data pattern seem to be circling around the that urban center (Blank in this graph because the Green taxies are not allowed to Pick-up there.) with the radii increasing with decreasing tip percent.

However, it is important to note that due to high overlap of this graph, and sorting of the plotting data by tip-percent, the low value points are covered by high value ones and thus this 'Euclidian distance from Manhattan' measure can be a good predictor of "MAXIMUM

POSSIBLE Percent of the Fare – Offered as Tip" but not for the variable of our interest which is just the "Percent of Fare – Offered as Tip".

Proposed Approach:

Learning from the failure of this approach, my proposed approach is as following:

1. Divide the area of city in equal sized squares of certain granularity. (Here approx. 1000ft. X 1000ft) and assign unique codes to them. – Form an area matrix.
2. In preprocessing the data, assign each valid trip to the square that contains the geolocation of that trip's origin/end point. (Here: Origin)
3. Compute the global expected value of the variable of interest.
4. Compute local expected value (mean) for the variable of interest for each cell in the area matrix.
5. Principle: Play safe when running low on data i.e. Don't rely on the local mean for areas with little number of training data-points. Instead – use the global mean.
6. Preprocess the test data in the same way as training data – assign the area code.
7. Predict the local mean of that area code as the predicted value for that test datapoint
8. The baseline considered is the local mean itself.
9. Ideas for future implementation:
    a. In case of insufficient datapoints for an area, instead of directly moving to global from local, use the local mean of the neighboring cells – with gradually increasing neighborhood radius and minimum threshold of data-points.
    b. Compute the similar value for Drop-off area. And predict the weighted sum of Pickup-area local mean and Dropoff area local mean where weights are the counts of datapoints available for those area codes.
    c. Factor in more variables on the explanatory side of the model – such as those mentioned in the thought section above.

Model Evaluation: The performance metric for the model used here is the Mean Squared Errors MSE which is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\widehat{Y}_i - Y_i)^2$$

Where, N is sample size, Y-hat is predicted value and Y is the observed value.

Data:

The data source is the same as used in LOADING DATA in this document. From it, A sample of **286571** was used as training data and a sample of size **287571** was used for testing. Both samples were generated by **SRS**. The samples were chosen from cleaned data to ensure the number of valid training and testing records available.

Cleaning and Preprocessing:

Observations and assumptions:

1. Most of the data with all payment types except credit cards (code 1) is either erroneous, inconsistent, missing or extreme. e.g. 0 or negative values in trip distances, fare amount, total amount etc. may represent variety of scenarios, including but not limited to: passenger-disputes, availing of special offers/discounts, missing/erroneous data, etc.
   Hence, only payment type 1 is considered valid for this analysis.

2. To keep visualizations meaningfully large enough, observations in far-away outskirts of the city which are in very tiny number are ignored.

3. Observations which show logical inconsistencies such as total amount < fare amount are ignored.

4. Outliers and extreme observations in terms of Tip_as_percent_total_fare beyond 50% are removed after careful observation, since 99+ quantiles of observations lie within 0 to 40.

5. Trips with time recorded as more than 200 minutes are ignored - few have more than 20 hours of trip time which doesn't help (and rather harms by influencing) analysis about routine taxi trips - such cases may be analyzed separately.

6. Average trip speed records were found to have illogical extremes such as speeds very close to zero and above 1000mph. The NYC taxi speed guideline is of 25mph. We assume the speed limit in between 5 and 40 mph.

7. The area under consideration is delineated in between latitudes 40.55 and 40.95 and longitudes -74.1 and -73.7 and is divided into 100X100 matrix, who's each cell gets a code which is an integer representing the serial number of that cell when measured from left to right and bottom to top, staring at bottom-left. Each cell represents approximately 1000ft.X1000ft. area on ground

31	32	33	34	35	36
25	26	27	28	29	30
19	20	21	22	23	24
13	14	15	16	17	18
07	08	09	10	11	12
01	02	03	04	05	06

Example of an area matrix of 6X6 along with area codes

8. Each trip is assigned a pickup and a dropoff area code in which the respective coordinates of that trip will fall.

Derived features generated:

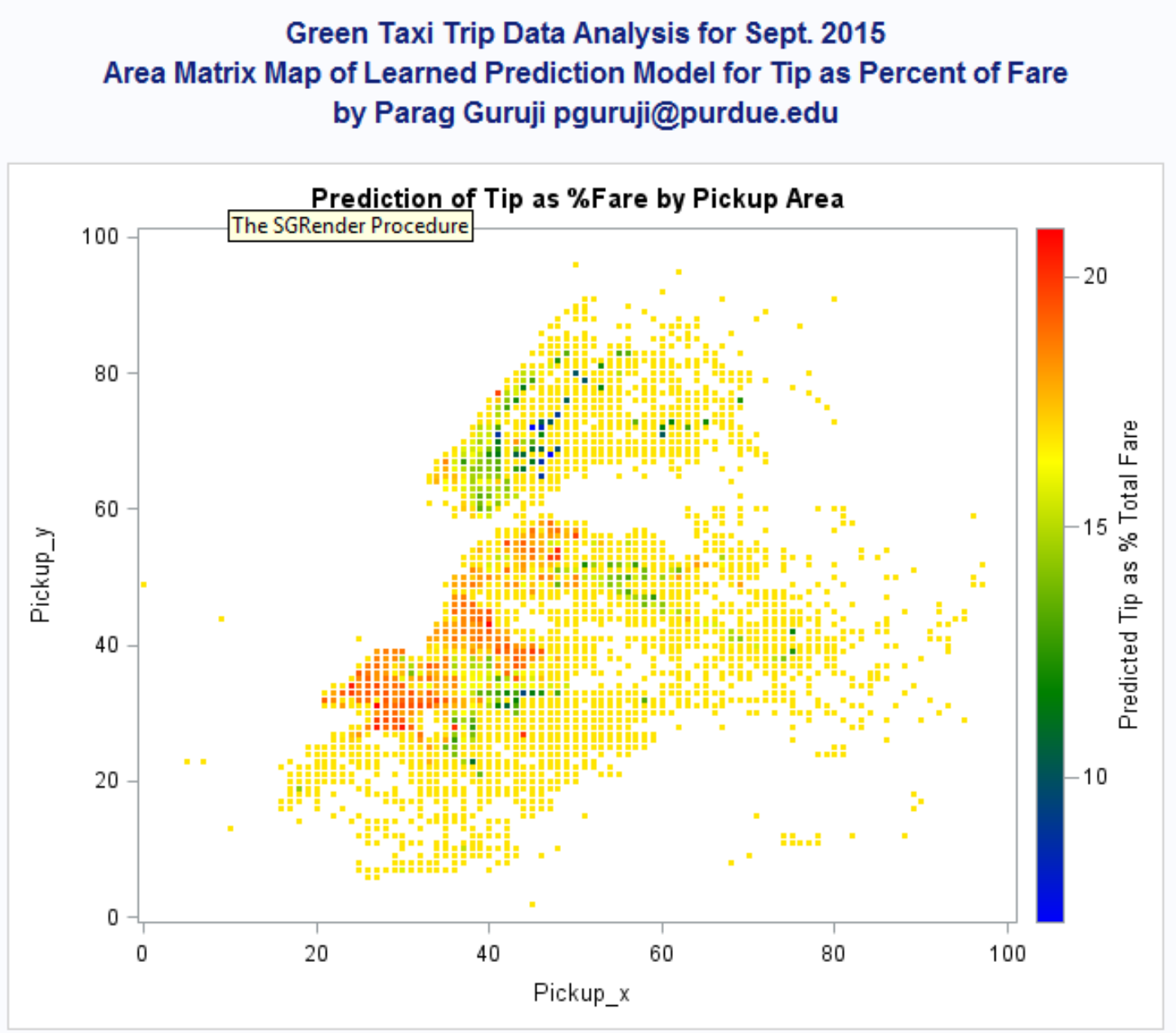
1. Trip_id: unique number for the record
2. Total_fare: as give in assumption above
3. Tip_as_percent_total_fare: (Tip_amount/Total_fare)*100;
4. Trip_minutes: Time spent in taxi (in min.)
5. Trip_speed: average speed over the course of the trip
6. Hour_of_day: {0, ..., 23} the hourly timeslot of pickup time
7. Pickup_area_code: area code for pickup location as explained above
8. Dropoff_area_code: area code for dropoff location as explained above

Instructions to run the code in SAS.

- The Source Code can be found in Appendix-2 and is also submitted in a separate .sas file (text file) - ParagGurujiCapitalOneDataScienceChallenge.sas
- Most of the relevant documentation and the logic is written in the comments for quick reference.
- Please verify the file paths mentioned at the top of the code are valid for your usage.
- If using URL instead of file, use the keyword 'url' before the quote mark of the file path as used in first example of dataurl.
- Code regarding each part of analysis begins with the comment "P<part_no.>"
- Always execute the LOADING DATA section of the code to read the source data to begin with.
- For : The code blocks are numbered and surrounded with comments "STEP<step_no> begin" and "STEP<step_no> end". Please execute them in that order to ensure the smooth process of all the code for Predictive Modeling.
- When using precooked data in intermediate states corresponding steps before that part can be skipped. E.g. while using training data that is already cleaned in some previous iteration of the code, no need to run the step 2 which corresponds to cleaning of raw data.
- To run any code block from the code open in the text editor of SAS, select all the lines intended to be executed by cursor and hit Run icon in the toolbar at the top of the editor.
- Keep an eye on log tab to ensure successful execution.

Model:

The Area matrix of trained model - color-coded as per Tip_as_percent_total_fare is visualized as shown below:



Following is the performance evaluation of my model. It outperforms the rather naïve baseline of using global mean of response variable by significant margin.

Baseline Performance Evaluation

The MEANS Procedure

Analysis Variable : error_sqaured				
N	Mean	Std Dev	Minimum	Maximum
286571	72.4187144	102.7213085	3.887158E-7	1099.59

Model Performance Evaluation

The MEANS Procedure

Analysis Variable : error_sqaured				
N	Mean	Std Dev	Minimum	Maximum
286571	69.3965226	101.0559297	4.7783823E-9	1445.53

The Mean columns in above table show the means of variable error_squared.

error_squared is computed as square of the error_term which is difference in predicted and observed values.

Thus, the proposed model outperforms baseline model by the metric MSE.

Analysis of Trip Speed – example of how conclusions from visual appearance may mislead!

Derived variable representing the average speed in MpH over the course of a trip:

Trip_speed = Trip_distance / (Trip_minutes / 60)

From , Trip minutes = minutes(Timestamp_of_dropoff – Timestamp_of_pickup)

Average trip speeds in all weeks of September:

Assumption: No datapoints satisfying either of these conditions are considered in this analysis for the suspicion of inconsistent/erroneous/extreme data recording.

1. Lpep_dropoff_datetime = 0
2. Lpep_pickup_datetime = 0
3. Trip_distance <= 0
4. Trip_minutes < 1 OR Trip_minutes > 200
5. Trip_speed > 100

The Means summary:

Green Taxi Trip Data Analysis for Sept. 2015
Mean Trip-Speed for all Weeks of September 2015

by Parag Guruji pguruji@purdue.edu

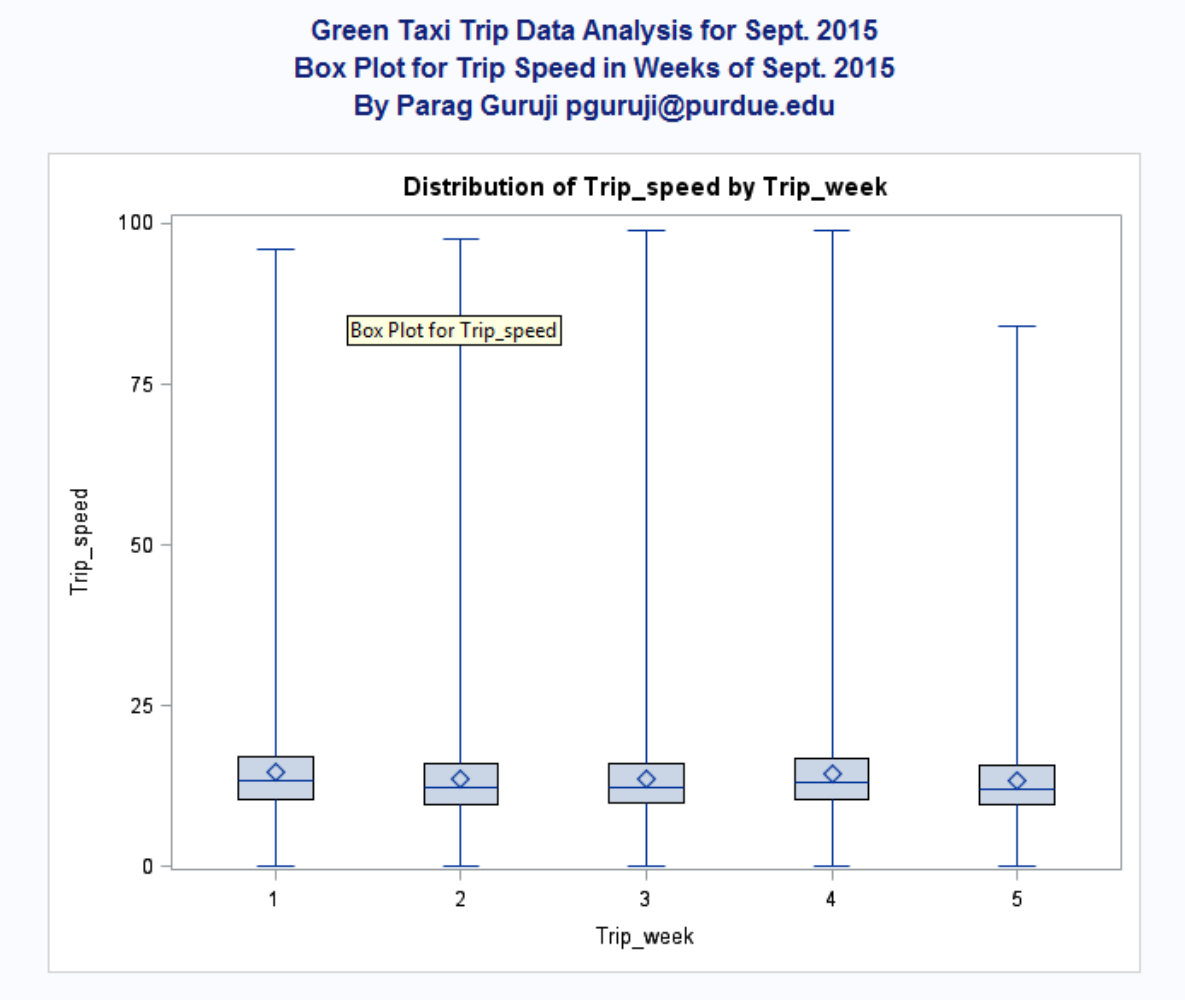
The MEANS Procedure

Analysis Variable : Trip_speed						
Trip_week	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	333226	333226	14.5765387	6.2641773	0.0181818	96.0000000
2	351246	351246	13.5600834	6.0135024	0.0035088	97.7142857
3	352994	352994	13.6284333	5.9455487	0.0057143	99.0000000
4	329004	329004	14.3208640	6.1523081	0.0056075	99.0000000
5	87782	87782	13.2968767	5.8275923	0.0461538	84.0000000

All the means appear to be in the range of 13 to 14.6 MpH

The Standard Deviations are almost same i.e. approximately 6

Graphical Summary:



From observation, all the weeks may appear to have almost the same mean approximately. However, such conclusions can be misleading especially given the huge sample size. Hence, we perform a statistical test to check if the mean trip speed is same for all the weeks.

We test if the small apparent differences in the mean speed for all 5 weeks is statistically insignificant or not by using ANOVA.

Let the null hypothesis be that

The mean speeds of all the weeks are materially the same

$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$

Let the alternative hypothesis be that Not All weeks of September have same mean speed

$H_a: \exists\ \mu_i \neq \mu_j$ for some, $i, j \in \{1, 2, 3, 4, 5\}$

ANOVA:

Although the distribution of the trip speed is highly right skewed, counting on the huge sample size, we choose to perform the ANOVA with Alpha=0.05

Green Taxi Trip Data Analysis for Sept. 2015
ANOVA for Trip Speed in Weeks of Sept. 2015
By Parag Guruji pguruji@purdue.edu

The GLM Procedure

Class Level Information		
Class	Levels	Values
Trip_week	5	1 2 3 4 5

Number of Observations Read	1454252
Number of Observations Used	1454252

Green Taxi Trip Data Analysis for Sept. 2015
ANOVA for Trip Speed in Weeks of Sept. 2015
By Parag Guruji pguruji@purdue.edu

The GLM Procedure

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	303044.01	75761.00	2052.07	<.0001
Error	1.45E6	53689834.66	36.92		
Corrected Total	1.45E6	53992878.67			

R-Square	Coeff Var	Root MSE	Trip_speed Mean
0.005613	43.50716	6.076128	13.96581

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Trip_week	4	303044.0068	75761.0017	2052.07	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Trip_week	4	303044.0068	75761.0017	2052.07	<.0001

From the ANOVA output, we see that the p-value is $<0.0001 << \alpha(0.05)$.

Hence, the data provides enough evidence that NOT all weeks of September 2015 have same average speed of Green Taxi trips in NYC.

Hypothesized REASONING for difference in mean traffic speeds for weeks of September 2015:

1. United Nations General Assembly's 70th session opens in NYC on the beginning of 3rd week i.e. 15th September. – traffic regulations and closures due to presence of foreign delegations may have causes much rerouting and congestion in NYC traffic – resulting in shift in the average traffic speed.
2. Pope Francis visited NYC on 25th September 2015 which further caused similar traffic issues in the 4th & 5th week.

Hypothesis of average trip speed as a function of time of day:

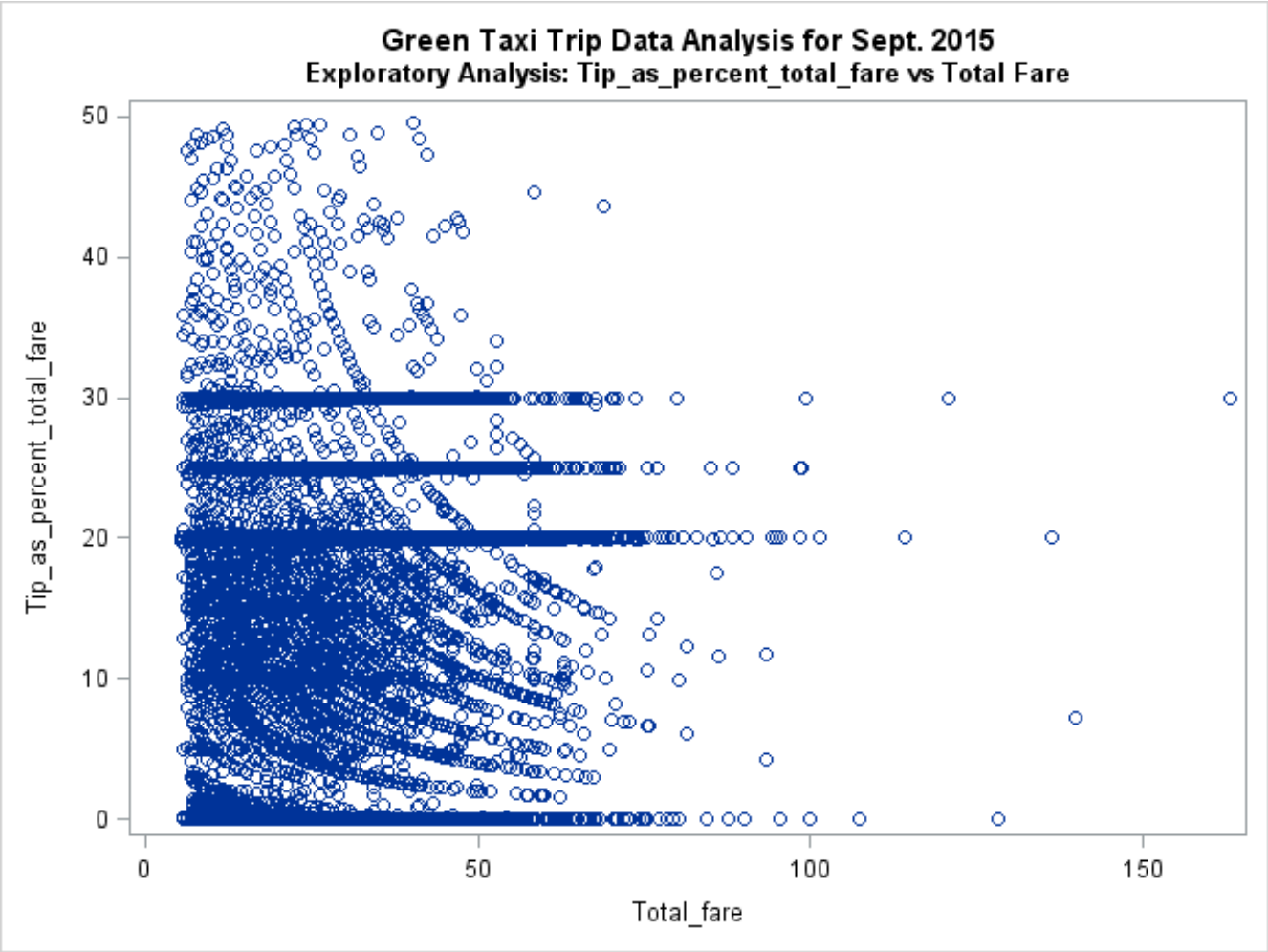
i.e. Trip_speed = f(Time_of_day)

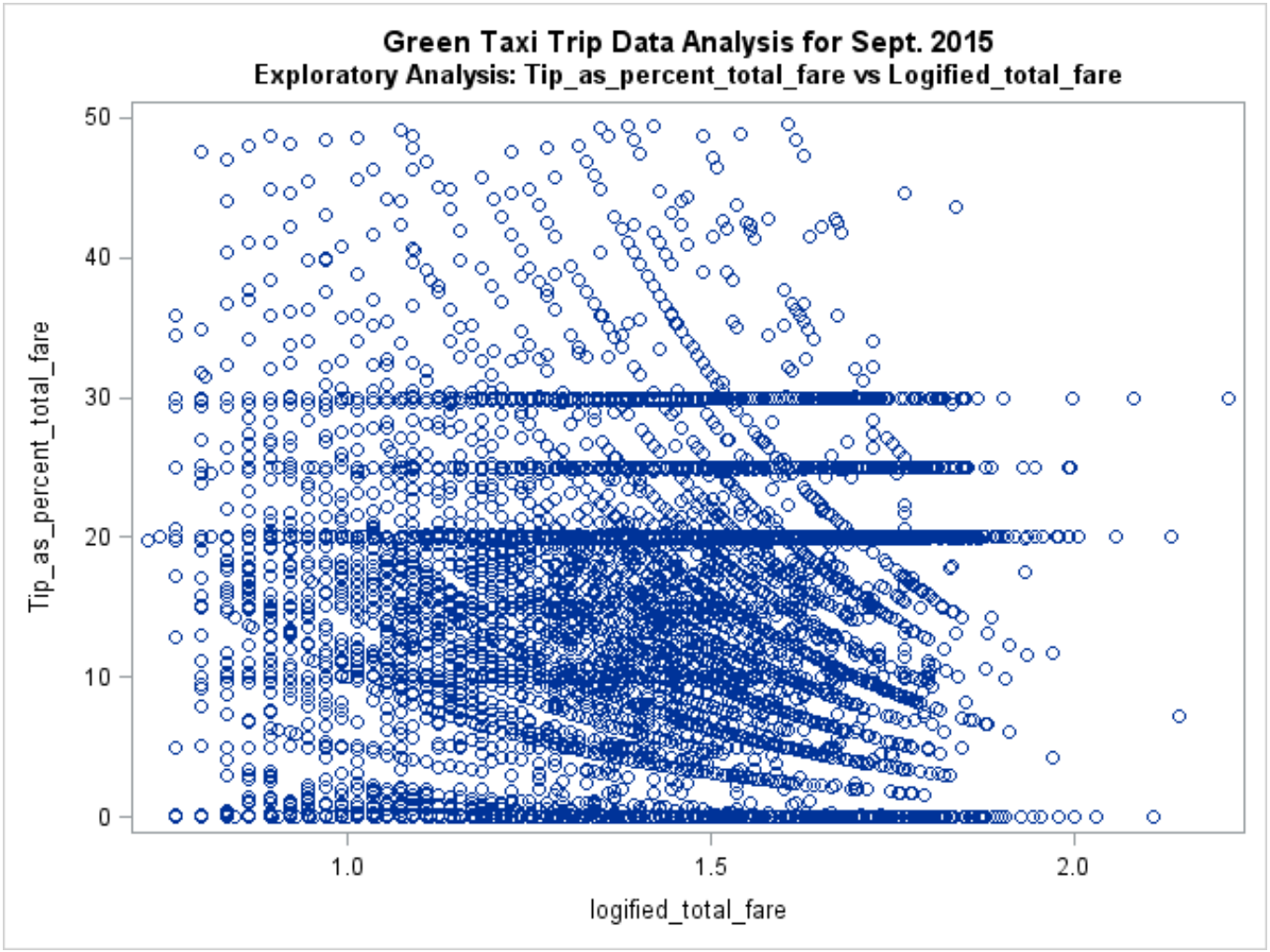
H_0: Trip_speed is independent of Time_of_day

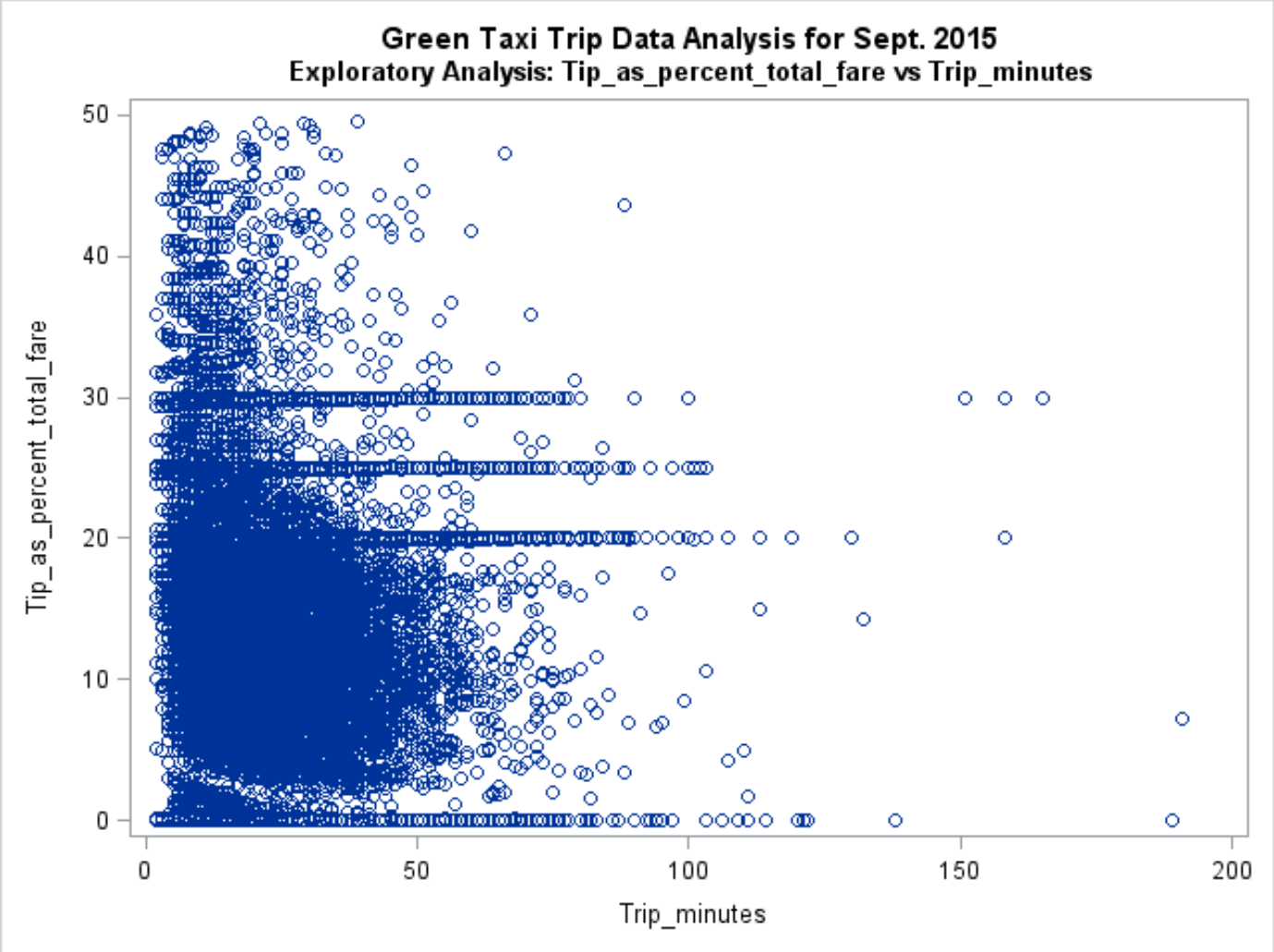
H_a: Trip_speed = f (Time_of_day)

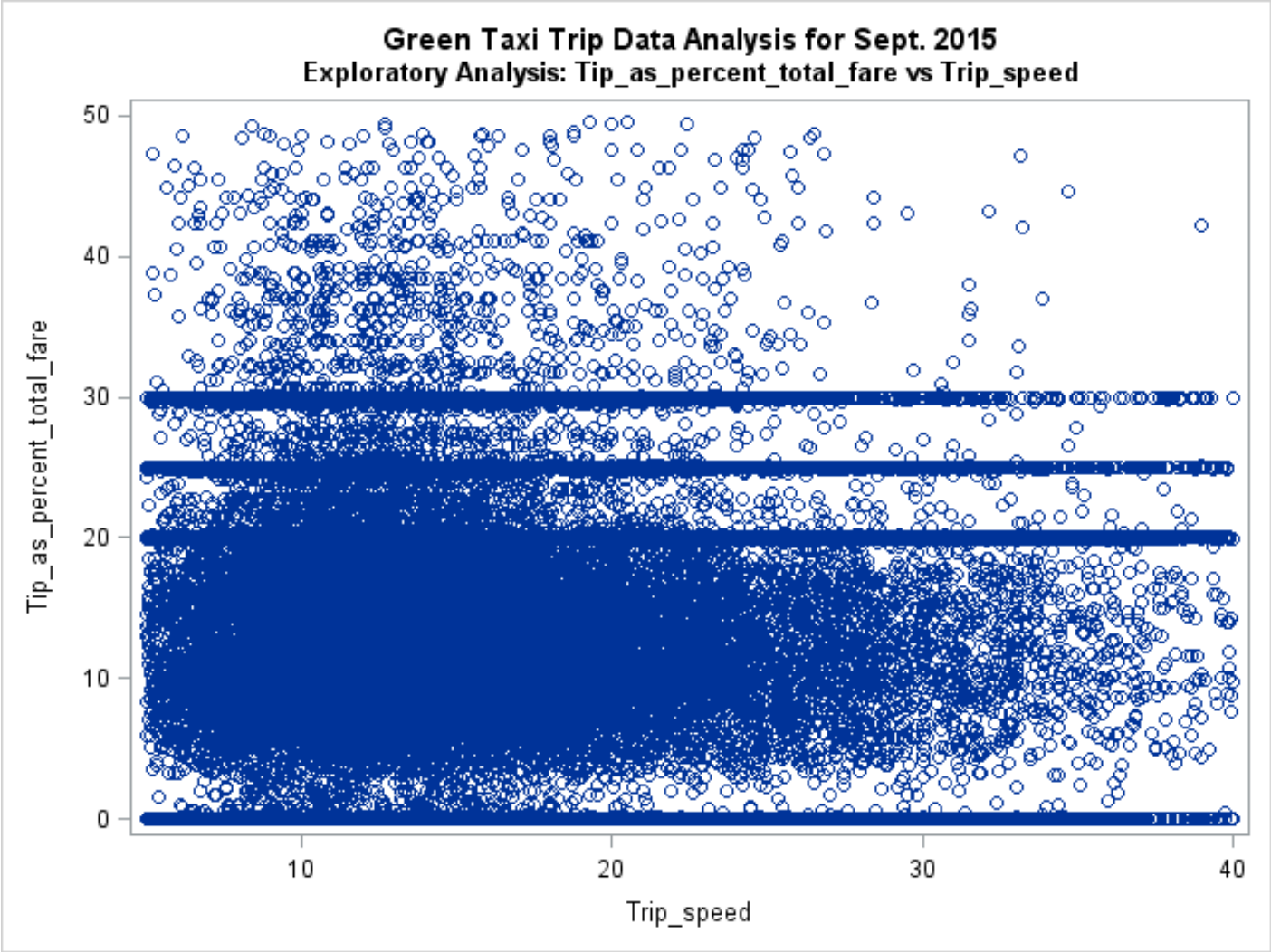
Appendix 1

Scatterplots from the Exploratory Analysis of Predictive Modeling




Green Taxi Trip Data Analysis for Sept. 2015
Exploratory Analysis: Tip_as_percent_total_fare vs Logified_total_fare
Tip_as_percent_total_fare
50
40
30
20
10
0
1.0
1.5
2.0
logified_total_fare


Green Taxi Trip Data Analysis for Sept. 2015
Exploratory Analysis: Tip_as_percent_total_fare vs Trip_minutes
Tip_as_percent_total_fare
50
40
30
20
10
0
0
50
100
150
200
Trip_minutes


Green Taxi Trip Data Analysis for Sept. 2015
Exploratory Analysis: Tip_as_percent_total_fare vs Trip_speed
Tip_as_percent_total_fare
50
40
30
20
10
0
10
20
30
40
Trip_speed

Appendix 2

SAS Source Code – Also submitted as a separate text file
ParagGurujiCapitalOneDataScienceChallenge.sas

```
/*Source Code for Capital One Data Science Challenge - Summer Internship
2017*/
/*
Author: Parag Guruji
Affiliation: Purdue University, West Lafayette, IN, USA
Email: pguruji@purdue.edu
Cell: +1-765-775-8727
*/

/*Set dataurl filename to the source URL for raw data*/
filename dataurl url 'https://s3.amazonaws.com/nyc-
tlc/trip+data/green_tripdata_2015-09.csv';
/*This dataset is assumed to always have the same format as of the given raw
dataset*/

/*Filenames for Predictive Modeling*/
/*Set filename for cleaned training dataset*/
filename ClTrDt 'W:\tip_prediction\cleaned_training_data.csv';
/*Predictive_Modeling_Cleaned_Training_Data: This dataset is cleaned and
processed training dataset*/

/*Set filename for inputting raw test data*/
filename RwTsDt 'W:\tip_prediction\raw_test_data.csv';
/*Predictive_Modeling_Raw_Test_Data: This dataset is assumed to always have
the same format as of the given raw dataset*/

/*Set filename for cleaned and preprocessed test dataset*/
filename ClTsDt 'W:\tip_prediction\cleaned_test_data.csv';
/*Predictive_Modeling_Cleaned_Test_Data: If your data is in the same form as
the dataurl input data, use the RwTsDt instead.*/

/*Set filename for saving the trained model*/
filename TrdMdl 'W:\tip_prediction\model_trained_on_pickup.csv';
/*Predictive_Modeling_Trained_Model:*/

/*Set filename for output data generated by running the proposed model
(usually referred to as 'model')*/
filename MdlOp 'W:\tip_prediction\output_of_model.csv';
/*Predictive_Modeling_Model_Output:*/

/*Set filename for output data generated by running the baseline model
(usually referred to as 'baseline')*/
filename BslOp 'W:\tip_prediction\output_of_baseline.csv';
/*Predictive_Modeling_Baseline_Output:*/

/*Set filename for baseline performance evaluation results dataset*/
filename BslPrf 'W:\tip_prediction\baseline_performance.csv';
/*Predictive_Modeling_Baseline_Performance:*/
```

```
/*Set filename for model performance evaluation results dataset*/
filename MdlPrf 'W:\tip_prediction\model_performance.csv';
/*Predictive_Modeling_Model_Performance:*/


/*LOADING DATA*/
/*Import CSV file from source specified by DATAURL into the dataset
WORK.raw_dataset */
proc import file=dataurl
                  DBMS=CSV
                  out=raw_dataset
                  replace;
      delimiter=',';
      getnames=yes;
run;

/*Debugging*/
/*Visually check if the data loaded as expected by printing first 10
observations*/
/*proc print data=raw_dataset(obs=10);
run;
*/



/*HISTOGRAM ANALYSIS FOR TRIP DISTANCE*/
/*Plot Histogram of Trip_distance*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Histogram of Trip Distance - Raw';
title3 'by Parag Guruji pguruji@purdue.edu';

proc univariate data=raw_dataset /*noprint*/;
      var Trip_distance;
      histogram Trip_distance / endpoints =(0 to 605 by 5);
run;

/*
We observe from the histogram and the numerical summary of Trip_distance
that:
1. The distribution is heavily right skewed
2. The first 99% Quantiles are in range 0 to 14.77 and last 1% quantiles in
14.77 to 603.10
3. by the AIRPORT TRIPS ANALYSIS + 1.5(IQR) measure, all observations above
7.7 can be termed as outliers which is more than 10% of the data
For the practical reason of facilitating a closer look at the shape of
distribution of the maximum possible chunk of the data (99% quantiles),
we eliminate the extreme right 1% quantiles to get a derived dataset for
ploting a histogram.
*/
data data99quantile(keep=Trip_distance trip_dist99); set raw_dataset;
      trip_dist99 = Trip_distance;
      if trip_dist99 > 14.77 then trip_dist99= .;
run;
```

```
/*Plot Histogram of trip_dist99, i.e. the Trip_distance variable in the 99%
quantile subset of the original data*/
proc univariate data=data99quantile /*noprint*/;
      var trip_dist99;
      histogram trip_dist99 / endpoints =(0.0 to 15 by 1) barlabels=percent;
run;


/*
From the data, we observe that the Trip_distance variable takes values over a
range of multi-order magnitudes from min. 0.01 to max. 603.1 (= 60310*min.)
The histogram of subset comprizing of first 99% quantiles also shows
logarithmic fall in percent frequency over Trip_distance
Thus, it is of interest to log-transform the data and explore the resultant
shape.

Hence, we create a new dataset 'logified_trip_distance' where 'Trip_distance'
is log transformed as 'log10_Trip_distance'
log10_Trip_distance = log10(Trip_distance + 0.001)
*/
data logified_trip_distance(keep=Trip_distance log10_Trip_distance); set
raw_dataset;
      log10_Trip_distance = log10(Trip_distance + 0.001);
run;

title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Histogram of log-transformed Trip_distance i.e. log10(Trip Distance +
0.001)';
title3 'by Parag Guruji pguruji@purdue.edu';

/*Plot Histogram of log10_Trip_distance*/
proc univariate data=logified_trip_distance noprint;
      var log10_Trip_distance;
      histogram log10_Trip_distance / endpoints =(-3 to +3 by 0.5) /*normal*/
barlabel=percent;
run;


/*AIRPORT TRIPS ANALYSIS*/
/*1. Mean Trip_distance grouped by hour of day*/
/*Compute variable 'hour of day' from the pickup timestamp of trip-
transaction*/
data trips_by_hour(keep=lpep_pickup_datetime Trip_distance hour_of_day); set
raw_dataset;
      hour_of_day=HOUR(lpep_pickup_datetime);
run;

/*Debugging*/
/*
proc print data=trips_by_hour(obs=10);
run;
*/
```

```
/*Compute the means and medians for trip_distances grouped by their Pickup
hour*/
proc means data=trips_by_hour MEAN MEDIAN noprint;
      class hour_of_day;
      var Trip_distance;
      output out=means_output mean=hourly_mean median=hourly_median;
run;

data hourly_mean_trip_dist; set means_output;
      if _TYPE_ = 0 then delete;
      drop _FREQ_ _TYPE_;
run;

proc print data=hourly_mean_trip_dist;
      title1 'Green Taxi Trip Data Analysis for Sept. 2015';
      title2 'Summary of hourly mean trip distance';
      title3 'by Parag Guruji pguruji@purdue.edu';
run;



/*AIRPORT TRIPS ANALYSIS. 2. Analysis of Airport Trips*/
/*Identify the airport trips and derive following pieces of information-
      1. Is airport trip? - The trip which either starts or ends within a
radius of 1 Km
            from the location point either of the 3 NYC airports as specified
by their geo-coordinates
      2. Which airport/(s)
      3. In-bound or out-bound or both

      For this, we create 3 varables:
      1. airport: {Y, N} - is this an airport trip? Y=yes, N=no
      2. pickup: {JFKA, LGA, NLIA, OTHER} - specifies pickup location as
either of 3 airports or OTHER
      3. dropoff: {JFKA, LGA, NLIA, OTHER} - specifies pickup location as
either of 3 airports or OTHER
*/
```

```
data airport_trips;
      set raw_dataset;
      trip_id = _n_;
      pickup = 'OTHER';
      dropoff = 'OTHER';
      airport = 'Y';
      /*Pickup within 1 km of the coordinates of LaGuardia Airport*/
      if    geodist(Pickup_latitude, Pickup_longitude, 40.77725, -73.872611,
'K') <= 1 then do;
            pickup = 'LGA';
            end;
      else do;
            /*Pickup within 1 km of the coordinates of John F. Kennedy
International Airport*/
            if    geodist(Pickup_latitude, Pickup_longitude, 40.639722, -
73.778889, 'K') <= 1 then do;
                  pickup = 'JFKIA';
                  end;
            else do;
                  /*Pickup within 1 km of the coordinates of Newark Liberty
International Airport*/
                  if    geodist(Pickup_latitude, Pickup_longitude, 40.6925, -
74.168611, 'K') <= 1 then do;
                        pickup = 'NLIA';
                        end;
                  end;
            end;
      /*Dropoff within 1 km of the coordinates of LaGuardia Airport*/
      if    geodist(Dropoff_latitude, Dropoff_longitude, 40.77725, -
73.872611, 'K') <= 1 then do;
            dropoff = 'LGA';
            end;
      else do;
            /*Dropoff within 1 km of the coordinates of John F. Kennedy
International Airport*/
            if    geodist(Dropoff_latitude, Dropoff_longitude, 40.639722, -
73.778889, 'K') <= 1 then do;
                  dropoff = 'JFKIA';
                  end;
            else do;
                  /*Dropoff within 1 km of the coordinates of Newark Liberty
International Airport*/
                  if    geodist(Dropoff_latitude, Dropoff_longitude, 40.6925,
-74.168611, 'K') <= 1 then do;
                        dropoff = 'NLIA';
                        end;
                  end;
            end;
      /*split the dataset in two datasets - airport_trips and other_trips*/
      if pickup = 'OTHER' and dropoff = 'OTHER' then do; airport='N'; end;
run;

/*Debugging*/
proc print data=cleaned_airport_trips(obs=20);
run;
*/
```

```
proc sort data=cleaned_airport_trips;
      by airport;
run;



title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Count of trips that are either in-bound or out-bound to any of 3 NYC
area airports';
title3 'by Parag Guruji pguruji@purdue.edu';

/*Count no. of airport trips*/
proc sql;
  select count(*) as Airport_trips_count
  from airport_trips
  where airport='Y';
quit;

/*Cleaning Airport Trips data for summary analysis*/
data cleaned_airport_trips; set airport_trips;
      if Fare_amount < 1 OR Trip_distance < 1 OR Payment_type NE 1
            OR Dropoff_latitude = 0 OR Dropoff_longitude = 0 OR
Pickup_latitude = 0 OR Pickup_longitude = 0 then delete;
      Fare_per_distance = Fare_amount/Trip_distance;
      Tip_as_percent_fare = FLOOR(100*Tip_amount/Fare_amount);
run;

/*Generate Summary (mean & median along with 95% confidence intervals) for
airport trips
      w.r.t. their: Fare amount, Trip distance and Tip amount*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Summary of Characteristics: Airport Trips VS Other Trips';
title3 'by Parag Guruji pguruji@purdue.edu';

proc means data=cleaned_airport_trips MEAN CLM MEDIAN MODE;
      class airport;
      var Fare_amount Trip_distance Tip_amount Fare_per_distance
Tip_as_percent_fare;
run;
```

```
/*
Compare if - on an average, the characteristics of airport trips are
significantly different from those of other trips.
      characteristics to be compared:
      1. Fare amount
      2. Trip distance
      3. Tip_amount
      4. Fare_per_distance
      5. Tip_as_percent_fare
*/
proc ttest data=cleaned_airport_trips sides=2 alpha=0.05 H0=0;
      title "Two sample t-test on  mean Fare amount Airport Trips vs Other
Trips";
      class airport;
      var Fare_amount Trip_distance Tip_amount Fare_per_distance
Tip_as_percent_fare;
run;



/**/
/*1. Feature design, 2. Predictive Modeling*/


/*
STEP1 begin
*/
/*create a GTL template that displays a scatter plot with highly transperant
markers
colored according to values of given continuous variable [Blue to Red
increasingly]*/
proc template;
  define statgraph gradientplot;
  dynamic _X _Y _Z _T;
  mvar LEGENDTITLE "optional title for legend";
    begingraph;
      entrytitle _T;
      layout overlay;
        scatterplot x=_X y=_Y /
          markercolorgradient=_Z colormodel=(BLUE GREEN YELLOW RED)
          markerattrs=(symbol=SquareFilled size=3) transparency=0.98
name="scatter";
        continuouslegend "scatter" / title=LEGENDTITLE;
      endlayout;
    endgraph;
  end;
run;
/*Template ends*/
/*
STEP1 end
*/
```

```
/*
STEP2 begin
*/
/*Assumption for total fare:
      Total Fare = Fare + Extra + Taxes + Surcharge + Fee + Tolls = Total
amount - Tip amount*/
/*Observations and assumptions in data cleaning:
      1. Most of the data with all payment types except credit cards (1) is
either erroneous, inconsistent, missing or extreme.
            e.g. 0 or negative values in trip distances, fare amount, total
amount etc. may represent variety of scenarios:
            including but not limited to: passanger-disputes, availing of
special offers/discounts, missing/erroneous data, etc.
            Hence, only payment type 1 is considered valid for this analysis.
      2. To keep visualizations meaningfully large enough, obervations in
far-away outskirts of the city which are in very tiny number are ignored
      3. Observations which show logical inconsistencies such as total amount
< fare amount are ignored.
      4. Outliers and extreme observations in terms of
Tip_as_percent_total_fare beyond 50% are removed after careful observation
            since 99.5 quantiles of observations lie within 0 to 40
      5. Trips with time recorded as more than 200 minutes are ignored - few
have more than 20hours of trip time
            which doesn't help (rather harms by influencing) analysis about
routine taxi trips - such cases may be analysed separately.
      6. Average trip speed records were found to have illogical extremes
such as speeds very close to zero and above 1000mph.
            The NYC taxi speed guideline is of 25mph. We assume the speed
limit in between 5 and 40 mph.
      7. The area under consideration is deliniated in between latitudes
40.55 and 40.95 and longitudes -74.1 and -73.7
            and is divided into 100X100 matrix, whose each cell gets a code
which is an integer representing the serial number of that cell
            when measured from left to right and bottom to top, staring at
bottom-left. Each cell represents approximately 1000ft.X1000ft. area on
ground
      8. Each trip is assigned a pickup and a dropoff area code in which the
respective coordinates of that trip will fall.

      Derived features generated:
      1. Trip id: unique number for the record
      2. Total_fare: as give in assumption above
      3. Tip_as_percent_total_fare:  (Tip_amount/Total_fare)*100;
      4. Trip_minutes: Time spent in taxi (in min.)
      5. Trip_speed: average speed over the course of the trip
      6. Hour_of_day: {0, ..., 23} the hourly timeslot of pickup time
      7. Pickup_area_code: area code for pickup location as explained above
      8. Dropoff_area_code: area code for dropoff location as explained above
*/
```

```
/*Clean and preprocess data
*/
data cleaned_training_data;
      set raw_dataset;
      Trip_id = _n_;
      if (Payment_type NE 1) OR (Fare_amount <= 0) OR (Total_amount <= 0) OR
Trip_distance < 1
            OR Pickup_latitude < 40.55 OR Pickup_latitude > 40.95 OR
Pickup_longitude < -74.1 OR Pickup_longitude > -73.7
            OR Dropoff_latitude < 40.55 OR Dropoff_latitude > 40.95 OR
Dropoff_longitude < -74.1 OR Dropoff_longitude > -73.7
            OR Extra < 0 OR MTA_tax < 0 OR Tip_amount < 0 OR Tolls_amount < 0
OR improvement_surcharge < 0
      then delete;
      else do;
            Total_fare = Total_amount - Tip_amount;
            if Total_fare <= 0 OR Total_fare > 200 then do; delete; end;
            else; do;
                  Tip_as_percent_total_fare = 100*Tip_amount/Total_fare;
                  if Tip_as_percent_total_fare > 50 then do; delete; end;
                  else; do;
                        Trip_minutes = FLOOR( (Lpep_dropoff_datetime -
Lpep_pickup_datetime)/60 );
                        if Trip_minutes < 1 OR Trip_minutes > 200 then do;
delete; end;
                        else; do;
                              Trip_speed = ( Trip_distance /
(Trip_minutes/60) );
                              if Trip_speed < 5 OR Trip_speed > 40 then do;
delete; end;
                              else; do;
                                    Hour_of_day=HOUR(Lpep_pickup_datetime);
                                    Pickup_area_code = (100 *
FLOOR((Pickup_latitude - 40.55)/0.004))
                                                                        +
CEIL((74.1 + Pickup_longitude)/0.004);
                                    Dropoff_area_code = (100 *
FLOOR((Dropoff_latitude - 40.55)/0.004))
                                                                        +
CEIL((74.1 + Dropoff_longitude)/0.004);
                              end;
                        end;
                  end;
            end;
      end;
run;

/*USE FOR TAINING DATA: Save cleaned training dataset to the specified file-
path given by ClTrDt*/
proc export data=cleaned_training_data
    outfile=ClTrDt /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP2 end
*/
```

```
/*Code for Exploratary Analysis*/
/*Scatterplots for relationship of Tip_as_percent_total_fare with different
potential predictor variables*/
proc surveyselect data=cleaned_training_data method=srs n=100000
out=plotting_sample noprint;
run;
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title3 'By Parag Guruji pguruji@purdue.edu';

/*Tip_as_percent_total_fare vs Total Fare*/
title2 'Exploratory Analysis: Tip_as_percent_total_fare vs Total Fare';
proc sgplot data=plotting_sample;
      scatter y=Tip_as_percent_total_fare x=Total_fare;
run;


/*Tip_as_percent_total_fare vs logified_total_fare*/
title2 'Exploratory Analysis: Tip_as_percent_total_fare vs
Logified_total_fare';
data logified_x_var; set plotting_sample;
      logified_total_fare = log10(Total_fare);
run;

proc sgplot data=logified_x_var;
      scatter y=Tip_as_percent_total_fare x=logified_total_fare;
run;


/*Tip_as_percent_total_fare vs Trip_minutes i.e. time spent in the taxi -
with the driver*/
title2 'Exploratory Analysis: Tip_as_percent_total_fare vs Trip_minutes';
proc sgplot data=plotting_sample;
      scatter y=Tip_as_percent_total_fare x=Trip_minutes;
run;


/*Tip_as_percent_total_fare vs Trip_speed*/
title2 'Exploratory Analysis: Tip_as_percent_total_fare vs Trip_speed';
proc sgplot data=plotting_sample;
      scatter y=Tip_as_percent_total_fare x=Trip_speed;
run;
/*Exploratary Analysis Code Ends*/
```

```
/*
STEP3 begin
*/

/*Visualization of Training Data:
      Plot Training Data on the area-map with marker color changing from blue
to red with increase in our response variable
      i.e. the tip as percentage of total fare*/
proc import file=ClTrDt
                  DBMS=CSV
                  out=cleaned_training_data
                  replace;
      delimiter=',';
      getnames=yes;
run;

proc sort data=cleaned_training_data;
      by Tip_as_percent_total_fare;
run;

title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Location Map of Tip as Percent of Fare as per Pickup Points';
title3 'by Parag Guruji pguruji@purdue.edu';
title4 'CAUTION: the data is sorted by Tip_as_percent_total_fare.
            Hence, the overlaping points showing higher-end color may contain
data for lower-end color underneath';

%let LegendTitle = "Tip as % Total Fare";
proc sgrender data=cleaned_training_data template=gradientplot;
   dynamic _X='Pickup_longitude' _Y='Pickup_latitude'
_Z='Tip_as_percent_total_fare' _T='Tip as %Fare by Pickups';
run;
/*Plotting ends*/
/*
STEP3 end
*/


/*Model Building Process*/
/*
STEP4 begin
*/
/*1. Generate Global Mean*/
proc univariate data=cleaned_training_data noprint;
      var Tip_as_percent_total_fare;
      output out=univar_op mean=global_mean;
run;

/* Debugging */
/*
proc print data=univar_op;
run;
*/

/*
STEP4 end
*/
```

```
/*
STEP5 begin
*/
/*2. Generate area-wise mean for prediction variable - tip as %age of total
fare*/
proc summary nway data=cleaned_training_data;
      class Pickup_area_code;
      var Tip_as_percent_total_fare;
      output out=intermediate_output_pickup n=Pickup_count mean=area_mean;
run;

/*
proc print data=intermediate_output_pickup(obs=20);
      title "Intermediate Output from Pickup Area";
run;
*/

/*
STEP5 end
*/


/*
STEP6 begin
*/
/*3. Set the predicted value to the Area-wise mean generated in (2) if we
have enough(>50) datapoints for that area,
      otherwise set it to the global mean.*/
data trained_on_pickup;
      if _n_=1 then set univar_op;
      set intermediate_output_pickup;
      if Pickup_count < 50
      then pred_tip_by_pickup = global_mean;
      else pred_tip_by_pickup = area_mean;
      Pickup_x = mod(Pickup_area_code, 100);
      Pickup_y = ceil(Pickup_area_code/100);
      drop _FREQ_ _TYPE_;
run;


/*Debugging*/
/*
proc print data=trained_on_pickup(obs=20);
      title "Model Trained on Pickup Area";
run;
*/


proc sort data=trained_on_pickup;
      by pred_tip_by_pickup;
run;
```

```
/*Save trained model to the specified file-path given by TrdMdl*/
proc export data=trained_on_pickup
    outfile=TrdMdl /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP6 end
*/


/*
STEP7 begin
*/
/*Import existing trained model*/
proc import file=TrdMdl
                  DBMS=CSV
                  out=trained_on_pickup
                  replace;
      delimiter=',';
      getnames=yes;
run;


/*Plot the area-wise gradient map to visualize our prediction*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Area Matrix Map of Learned Prediction Model for Tip as Percent of
Fare';
title3 'by Parag Guruji pguruji@purdue.edu';

%let LegendTitle = "Predicted Tip as % Total Fare";
proc sgrender data=trained_on_pickup template=gradientplot;
   dynamic _X='Pickup_x' _Y='Pickup_y' _Z='pred_tip_by_pickup' _T='Prediction
of Tip as %Fare by Pickup Area';
run;
/*
STEP7 end
*/
/*Model Building Process Ends*/



/*Test Dataset Preparation*/
/*
STEP8 begin
*/
/*Import the raw Test data*/
proc import file=RwTsDt
                  DBMS=CSV
                  out=raw_test_data
                  replace;
      delimiter=',';
      getnames=yes;
run;
/*
STEP8 end
*/
```

```
/*
STEP9 begin
*/
/*Clean and preprocess the test data in same way as done for training data*/
data cleaned_test_data;
      set raw_test_data;
      Trip_id = _n_;
      if (Payment_type NE 1) OR (Fare_amount <= 0) OR (Total_amount <= 0) OR
Trip_distance < 1
            OR Pickup_latitude < 40.55 OR Pickup_latitude > 40.95 OR
Pickup_longitude < -74.1 OR Pickup_longitude > -73.7
            OR Dropoff_latitude < 40.55 OR Dropoff_latitude > 40.95 OR
Dropoff_longitude < -74.1 OR Dropoff_longitude > -73.7
            OR Extra < 0 OR MTA_tax < 0 OR Tip_amount < 0 OR Tolls_amount < 0
OR improvement_surcharge < 0
      then delete;
      else do;
            Total_fare = Total_amount - Tip_amount;
            if Total_fare <= 0 OR Total_fare > 200 then do; delete; end;
            else; do;
                  Tip_as_percent_total_fare = 100*Tip_amount/Total_fare;
                  if Tip_as_percent_total_fare > 50 then do; delete; end;
                  else; do;
                        Trip_minutes = FLOOR( (Lpep_dropoff_datetime -
Lpep_pickup_datetime)/60 );
                        if Trip_minutes < 1 OR Trip_minutes > 200 then do;
delete; end;
                        else; do;
                              Trip_speed = ( Trip_distance /
(Trip_minutes/60) );
                              if Trip_speed < 5 OR Trip_speed > 40 then do;
delete; end;
                              else; do;
                                    Hour_of_day=HOUR(Lpep_pickup_datetime);
                                    Pickup_area_code = (100 *
FLOOR((Pickup_latitude - 40.55)/0.004))
                                                                        +
CEIL((74.1 + Pickup_longitude)/0.004);
                                    Dropoff_area_code = (100 *
FLOOR((Dropoff_latitude - 40.55)/0.004))
                                                                        +
CEIL((74.1 + Dropoff_longitude)/0.004);
                              end;
                        end;
                  end;
            end;
      end;
run;

proc sort data=cleaned_test_data;
      by Pickup_area_code;
run;
```

```
/*Save cleaned dataset to the specified file-path given by ClTsDt*/
proc export data=cleaned_test_data
    outfile=ClTsDt /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP9 end
*/
/*Test Dataset Preparation Ends*/



/*Running the baseline model on test data*/
/*
STEP10 begin
*/
/*Import the existing cleaned and preprocessed Test data*/
proc import file=ClTsDt
                DBMS=CSV
                out=cleaned_test_data
                replace;
      delimiter=',';
      getnames=yes;
run;

proc sort data=cleaned_test_data;
      by Pickup_area_code;
run;

/*
STEP10 end
*/

/*
STEP11 begin
*/
/*Baseline model generation*/
data baseline_global_mean;
      if _n_=1 then set univar_op;
      set intermediate_output_pickup;
      pred_tip_by_pickup = global_mean;
      Pickup_x = mod(Pickup_area_code, 100);
      Pickup_y = ceil(Pickup_area_code/100);
      drop _FREQ_ _TYPE_;
run;

/*Debugging*/
/*
proc print data=baseline_global_mean;
      title "Baseline Predictions";
run;
*/
/*
STEP11 end
*/
/*Baseline model generation ends*/
```

```
/*
STEP12 begin
*/
/*Baseline Evaluation Process*/
proc sort data=baseline_global_mean;
      by Pickup_area_code;
run;

/*Run the baseline model of global mean on test data to generate the output*/
data baseline_evaluation;
      merge baseline_global_mean cleaned_test_data;
      by Pickup_area_code;
      error_term = pred_tip_by_pickup - Tip_as_percent_total_fare;
      error_sqaured = error_term * error_term;
run;


/*Save the baseline output for test data in BslOp file*/
proc export data=baseline_evaluation
    outfile=BslOp /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP12 end
*/

/*
STEP13 begin
*/
/*Import the existing baseline output for Test data*/
proc import file=BslOp
                  DBMS=CSV
                  out=baseline_evaluation
                  replace;
      delimiter=',';
      getnames=yes;
run;

/*Compute evaluation metric Mean Squared Error MSE for baseline model*/
proc means data=baseline_evaluation;
      var error_sqaured;
      Title "Baseline Performance Evaluation";
      output out=baseline_performance mean=MSE;
run;

/*Save the baseline performance metric MSE in BslPrf*/
proc export data=baseline_performance
    outfile=BslPrf /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP13 end
*/
```

```
/*Baseline Evaluation Process Ends*/


/*Model Evaluation Process*/
/*
STEP14 begin
*/
/*Import existing trained model*/
proc import file=TrdMdl
                  DBMS=CSV
                  out=trained_on_pickup
                  replace;
      delimiter=',';
      getnames=yes;
run;

proc sort data=trained_on_pickup;
      by Pickup_area_code;
run;
/*
STEP14 end
*/


/*
STEP15 begin
*/
/*Import the existing cleaned and preprocessed Test data*/
proc import file=ClTsDt
                  DBMS=CSV
                  out=cleaned_test_data
                  replace;
      delimiter=',';
      getnames=yes;
run;

proc sort data=cleaned_test_data;
      by Pickup_area_code;
run;
/*
STEP15 end
*/
```

```
/*
STEP16 begin
*/
/*Running the model (trained_on_pickup) on test data*/
data model_evaluation;
      merge trained_on_pickup cleaned_test_data;
      by Pickup_area_code;
      error_term = pred_tip_by_pickup - Tip_as_percent_total_fare; /*Error =
Predicted - Observed*/
      error_sqaured = error_term * error_term; /*error squared*/
run;


/*Save the output of running the model on test data in _output_data file*/
proc export data=model_evaluation
    outfile=MdlOp /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP16 end
*/


/*
STEP17 begin
*/
/*Import the existing output of running the model on test data*/
proc import file=MdlOp
                  DBMS=CSV
                  out=model_evaluation
                  replace;
      delimiter=',';
      getnames=yes;
run;


/*Compute Model's Mean Squared Error MSE*/
proc means data=model_evaluation;
      var error_sqaured;
      Title "Model Performance Evaluation";
      output out=model_performance mean=MSE;
run;


/*Save the model performance metric MSE in MdlPrf*/
proc export data=model_performance
    outfile=MdlPrf /*Set the appropriate file path at the top*/
    dbms=csv
    replace;
run;
/*
STEP17 end
*/
/*Model Evaluation Process Ends*/
```

```
/*WEEKLY VARIATION IN AVERAGE TRIP SPEED*/
/*Prepare a lightly cleaned - i.e. cleaned w.r.t. only the variables involved
in this point of analysis

Create variables:
1. Trip_minutes: Duration of trip in minutes
2. Trip_speed: Average speed over the course of trip - in MpH
      Trip_speed = ( Trip_distance / (Trip_minutes/60) )
3. Trip_week: week of Sept. {1, 2, 3, 4, 5} in which trip occured.
*/
data speed_by_week; set raw_dataset;
      if Lpep_dropoff_datetime = 0 OR Lpep_pickup_datetime = 0 OR
Trip_distance <= 0 then do; delete; end;
      else; do;
            Trip_minutes = FLOOR( (Lpep_dropoff_datetime -
Lpep_pickup_datetime)/60 );
            if Trip_minutes < 1 OR Trip_minutes > 200 then do; delete; end;
            else; do;
                  Trip_speed = ( Trip_distance / (Trip_minutes/60) );
                  if trip_speed > 100 then do; delete; end;
                  else; do;
                        Trip_week =
ceil(day(datepart(lpep_pickup_datetime))/7);
                  end;
            end;
      end;
      keep Trip_week Trip_id Trip_speed;
run;

proc sort data = speed_by_week;
      by Trip_week;
run;

/*Compute the average speeds for grouped by week number */
proc means data=speed_by_week;
      class Trip_week;
      var Trip_speed;
run;

/*Side by Side Boxplots*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Side by Side Boxplots for Trip Speed in Weeks of Sept. 2015';
title3 'By Parag Guruji pguruji@purdue.edu';
proc boxplot data=speed_by_week;
      plot Trip_speed*Trip_week;
run;
```

```
/*ANOVA*/
```

```
/*Check the normality*/
/*QQplot*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'Side by Side Boxplots for Trip Speed in Weeks of Sept. 2015';
title3 'By Parag Guruji pguruji@purdue.edu';
symbol value = circle i=none;
proc univariate data=speed_by_week noprint;
  var Trip_speed;
  by Trip_week;
  qqplot;
run;

/*Generate the ANOVA table*/
/*Side by Side Boxplots*/
title1 'Green Taxi Trip Data Analysis for Sept. 2015';
title2 'ANOVA for Trip Speed in Weeks of Sept. 2015';
title3 'By Parag Guruji pguruji@purdue.edu';
proc glm data=speed_by_week;
  class Trip_week;
  model Trip_speed = Trip_week;
  means Trip_week / bon lines;
run;


/*Auxilary code for splitting up the Given data set into training and testing
in Predictive Modeling*/
/*
proc import file='W:\tip_prediction\randomized total clean data.csv'
                  DBMS=CSV
                  out=total_data
                  replace;
      delimiter=',';
      getnames=yes;
run;

data final_training_data; set total_data;
      if _n_ > 287043 then delete;
run;

proc export data=final_training_data
    outfile='W:\tip_prediction\final_training_data.csv'
    dbms=csv
    replace;
run;


data final_testing_data; set total_data;
      if _n_ <= 287043 then delete;
run;

proc export data=final_testing_data
    outfile='W:\tip_prediction\final_testing_data.csv'
    dbms=csv
    replace;
run;
*/
```