TAMIDS 2016 Data Science Competition

Nishant Barma, Josiah Coad, Erica Metheney, Savinay Narendra

1 Goals and Objectives

Since the introduction of ride sharing services, the Chicago taxi industry has seen a steady decline. The TAMIDS Data Science Competition has asked for visualizations and predictive models explaining how the Chicago taxi industry has changed in response to this new source of competition. This will be achieved using publicly available data provided by the city of Chicago which contains information about over 110 million taxi rides during the time period January 1, 2013 to July 31, 2017. Specifically we aim to build interpretable, predictive models for

- Weekly Median Revenue (Trip Total)
- Daily Median Revenue (Trip Total)
- Hourly Median Revenue (Trip Total)

For each model we will provide thorough explanations of the predictors in the model as well as the prediction error. We will also provide dynamic visualizations for

- Heatmap of weekly and hourly medians of Trip Total, Trip Miles and Trip Seconds (provided in the appendix)

as well as interactive visualizations of

- Concentration of Drop-offs for each Year

Finally the visualizations and predictive models will be combined to give a comprehensive explanation of the changes seen in the Chicago taxi industry from 2013 to 2017.

2 Data Exploration

When creating our visualizations and predictive models we will use a combination of the data provided by the city of Chicago, variables inferred from the provided data, and variables acquired from other sources.

2.1 Data Cleaning

We used the following method for cleaning and aggregating our interval variables. We took the sum of the variable for each taxi in a given time block (hour, day, week). This resulted in a table of Taxi ID by time block. Then we took the medians of all the taxis operating in that time block. If there was a taxi operating during that time period but for whatever reason, that taxi had all 0 values for that time block (this happened a lot for tolls), there would be a zero total sum for that taxi for that time block. If half or more of the taxis had 0s for that time block, then the median would also be zero (again, this happened a lot for tolls). If there were no entries for a taxi for a particular time block (say that taxi took the hour off), then their respective sum for that time block would be NA.

When calculating the sums of variables we decided to drop the NA values for that variable instead of imputing. We did this individually for each variable. We considered this justifiable because of the relatively small number of NAs existing in the data set, see Table 1. Further we are working with medians per week, which is a statistic robust to such a small number of NA values.

The code implementing data cleaning can be found at `https://github.com/savinay/TAMU-Data-Science-Competition/`.

Table 1: Number of NA Values per Year

	2013	2014	2015	2016	2017
Fare	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Tips	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Tolls	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Extras	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Taxi ID	>0.00%	>0.00%	>0.00%	0.01%	0.05%
Trip Total	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Trip Miles	>0.00%	>0.00%	>0.00%	>0.00%	>0.00%
Trip Seconds	4.23%	0.49%	>0.02%	0.01%	>0.00%
Trip Start Timestamp	0	0	0	0	0
Total Entries	26,870,288	31,021,727	27,400,745	19,878,277	7,688,951

2.2 Provided Data

The taxi ride data provided by the city of Chicago contains 23 variables describing taxi rides for the years 2013 – 2017. Appendix D contains the summary for the interval variables by year as well as correlation plots between the interval variables. Table 2 summarizes the variables.

Table 2

Nominal Variables	Trip ID, Taxi ID, Payment Type , Company, Pickup Census Tract, Drop off Census Tract
Interval Variables (Monetary)	Fare, Tips, Tolls, Extras, Trip Total
Interval Variables (Geographic)	Pickup Centroid Latitude/Longitude, Pickup Drop off Centroid Latitude/Longitude, Pickup/Drop off Centroid Location, Pickup/Drop off Community Area
Interval Variables (Time)	Trip Start Timestamp, Trip End Timestamp, Trip Seconds
Interval Variable (Distance)	Trip Miles

2.3 Additional Data

When deciding what additional variables to compute and obtain, we focused on those factors that directly impact taxi usage. We also considered variables found important in a previous

analysis performed by Todd Schneider [1].

2.3.1 External Variables

We have taken into account external variables which include

- Weather Data [2]
- Beginning of Lyft (May 11, 2013)
- Beginning of Uber (2011)
- Day Light Savings
- Holidays (Halloween, Christmas, New Year)
- Rush Hour (6 am - 10 am and 3 pm - 7 pm) [3]

2.4 Visualizations

We provide a visualization of the median Trip Total a driver is expected to make on each hour in a weekday for a particular year. We have calculated the sum of the Trip Totals a driver makes in each hour of the week. Then we computed the median of the sum of Trip Totals for each Taxi ID to obtain this visualization.

Figure 1: Median Trip Total 2013

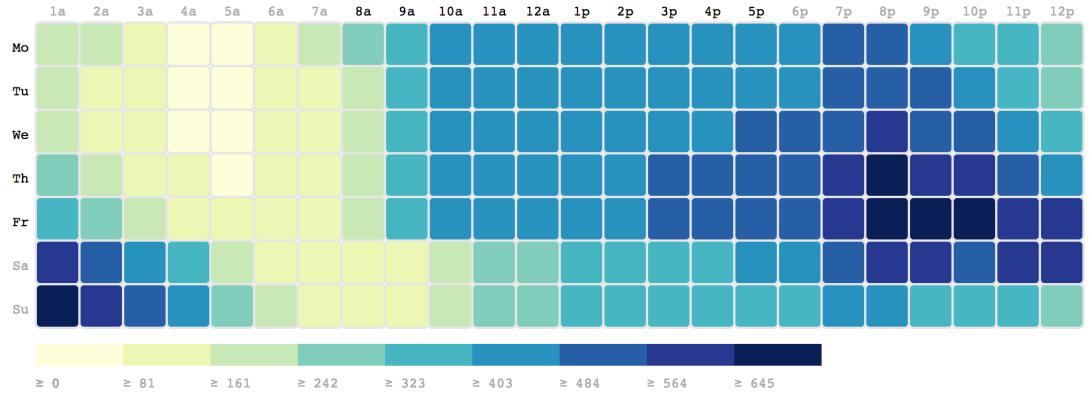


Figure 2: Median Trip Total 2014

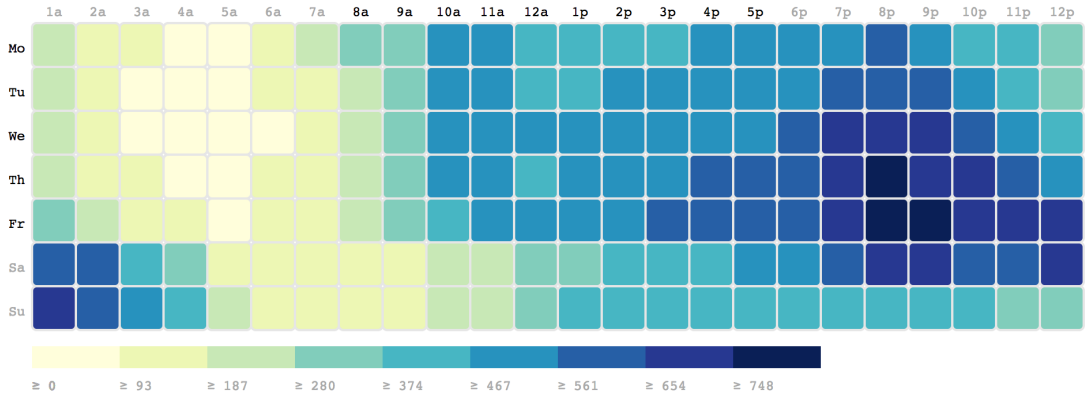
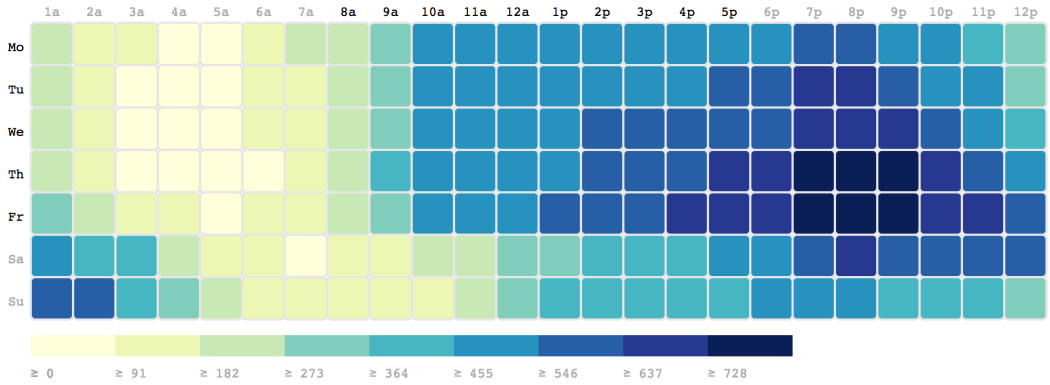


Figure 3: Median Trip Total 2015



We can see from the visualizations that Trip Totals from 7 PM – 9 PM are the highest as this is the time people return from offices. We also see a high number of Trip Totals on Friday and Saturday nights which provides us the information that people are returning home after celebrating their weekend nights. We observe the same trends for all the years while calculating Trip Totals. Even though the scale for each plot is different, we see that each one demonstrates almost the same behavior.

All the other visualizations for the remaining variables (Trip Miles and Trip Seconds) for each year are in Appendix E. We have also created an interactive heatmap which shows the concentration of taxi dropoffs in different locations for each year. Below is a snapshot of the interactive map.

Figure 4: Median Trip Total 2016

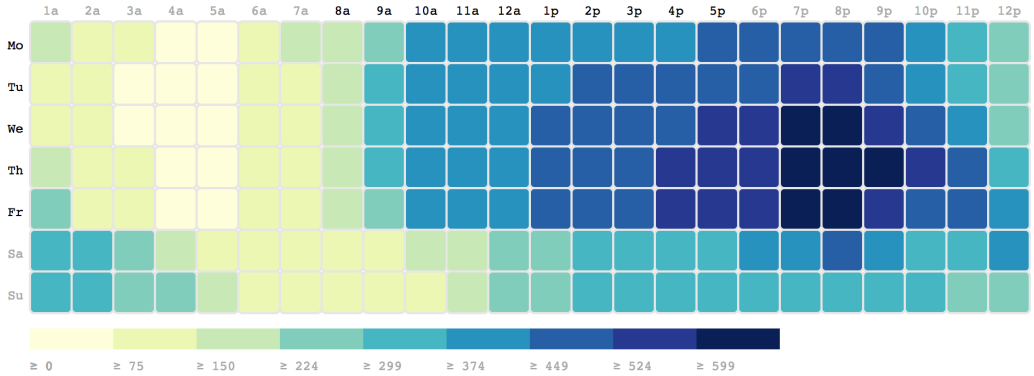
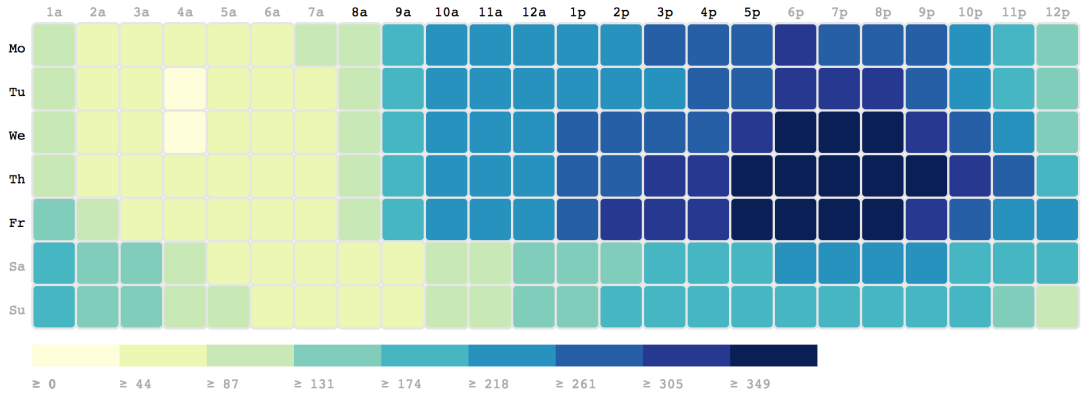
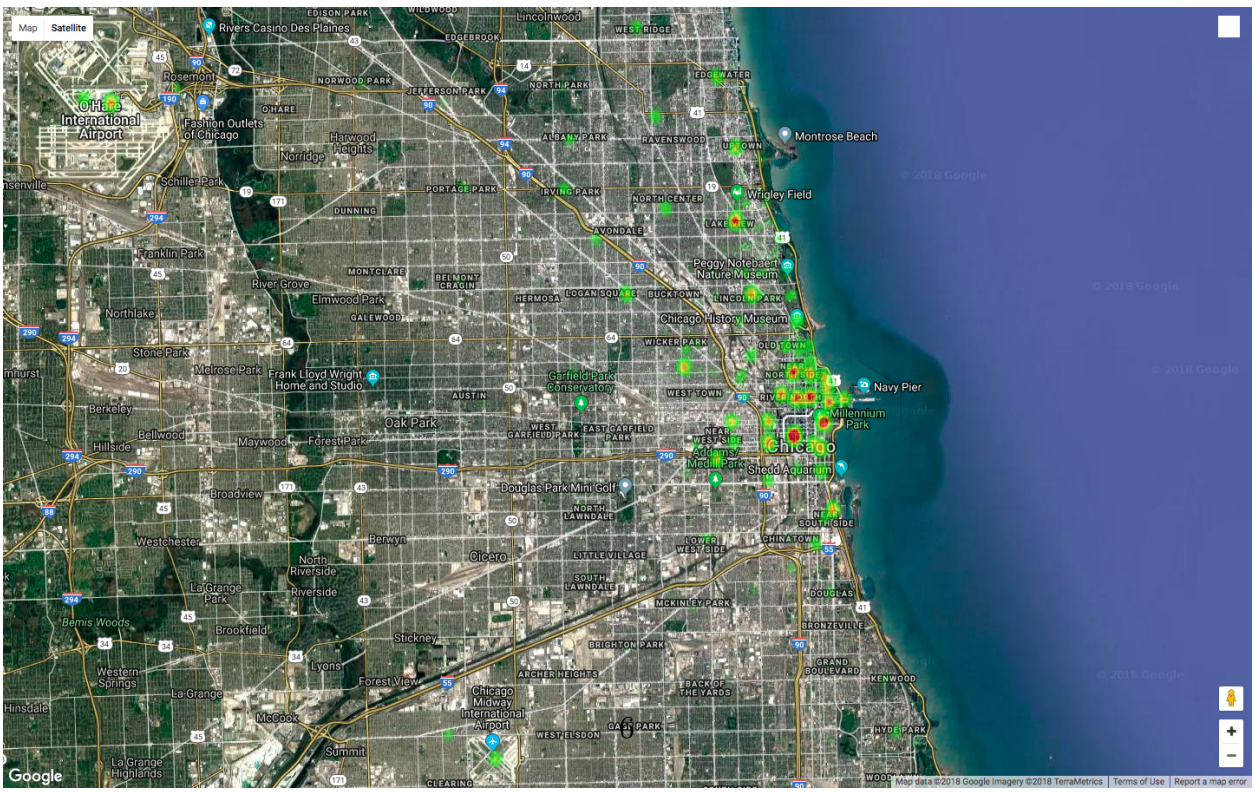


Figure 5: Median Trip Total 2017





Please visit `https://stark-dawn-75655.herokuapp.com/` if you would like to interact with the map on your own.

3 Methods for Analysis

Since the revenue data is a time series, we will be fitting a linear model with time series errors. Our analysis will consist of four stages: model building, interpretation, prediction, and accuracy measures. All models will be built on the data from January 1, 2013 to December 31, 2016. All predictions will be made on the data from January 1, 2017 to July 31, 2017. Below we explain each stage in detail.

3.1 Model Building

1. First plot the response variable for the time period of interest to determine whether there are any necessary external variables, eg Holidays, major events.
2. Fit a full linear model containing all candidate predictors.
3. Use an exhaustive search to find the three best models of each size.
4. Use BIC to select an initial model from those generated in Step 3.
5. Check the residuals for stationarity.
    - If not stationary, modify the linear model and return to Step 5.
    - If stationary, create ACF and PACF plots of the residuals.
6. Use an automated ARIMA order selector (or SARIMA order selector if the ACF/PACF indicate seasonality) to obtain an initial time series model for the residuals.
7. Check diagnostic plots to see if the time series model is valid.
    - If invalid, modify the time series model.
    - If valid, proceed to prediction stage.

3.2 Interpretation of the Model

For each coefficient in the linear portion of our model we will report the following:

- Value of the Coefficient
- Comments about Value of Coefficient

3.3 Accuracy Measures

In order to assess the accuracy of our predictions, we will provide graphs of our predictions overlaid with the true values. We will also provide the three measures of prediction accuracy described in Table 3. In the descriptions: y_i is the true value, $\hat{y}_i$ is the predicted value, and $\hat{e}_i$ is the difference between them.

Table 3: Prediction Accuracy Measures

MAE	Mean Absolute Error	$\text{mean}(\lvert\hat{e}_i\rvert)$
RMSE	Root Mean Squared Error	$\sqrt{\text{mean}(\hat{e}_i{}^2)}$
MAPE	Mean Absolute Percent Error	$\text{mean}\left(\left\lvert\frac{y_i-\hat{y}_i}{y_i}\right\rvert\right)$

3.4 Important Note

We would like to take a moment to address a very serious shortcoming to our methodology. We have first fit a linear model, extracted the residuals, and then fit a time series to those residuals. We then simply combine the two halves to create a final model. While this procedure does create a model with white noise errors, it does not provide a model with coefficients on which we can perform inference.

To avoid this issue one should use the time series model on the residuals to inform the order that should be used, then refit the entire model, estimating the linear and time series coefficients at the same time. For the sake of time we have omitted this step and as a result will not be reporting p-values, confidence intervals, or other types of inference.

4 Analysis

4.1 Weekly Median Revenue

We show the final model for weekly median revenue built from 212 weekly observations used to predict the first 31 weeks of 2017.

4.1.1 Added Variables

The weekly median revenue model required a number of additional variables. In addition to weather data, it turned out that 2013 had a very different behavior than the other years. For that reason we added an indicator for the year 2013 and included all significant interactions between i2013 and the other predictors. The other added variables can be found in Table 4.

Table 4

Variable Name	**Description**
iHoliday	Indicator representing the last 4 weeks of the year corresponding to the holiday season which had a lower value.
i29, i157	Indicator for week 29/157, each of which had a drastically low number which we could not explain.
lyft	Represents the number of weeks since lyft began operations

4.1.2 Final Model

The linear portion of the final weekly median revenue contained 14 predictors, listed below in Table 5. The time series model applied to the residuals was a SARIMA$(2,0,2)(2,0,2)_4$. The VIFs as well as the diagnostic plots for the time series can be found in Appendix C.

4.1.3 Predictions

Below is a plot of the actual 2017 median weekly revenues in black overlaid with our predictions in blue. We see that our model follows the general trend fairly well. However efforts

Table 5: Weekly Model Coefficients Summary

Predictor	Value	Comments
TripMiles	0.4902	
Tips	4.4989	
Extras	5.2294	
iDepart	-92.3274	Extreme temperature fluctuations decrease revenue
iSnowDepth	118.2571	Large amounts of snow increase revenue
i2013	378.7097	
lyft	-0.6489	Negative impact on revenue
iHoliday	-40.0712	Decrease in revenue the last 4 weeks of the year
i29	-381.9099	Dramatically lower for unknown reason
i157	-335.2704	Dramatically lower for unknown reason
i2013*lyft	-8.4967	
TripMiles*i2013	-1.3351	
Tips*i2013	2.9160	
Extras*i2013	-2.2070	

should be made to improve performance at weeks 5–10 and 15–20. From Table 6 we see that our average error is about 5% or equivalently $60.

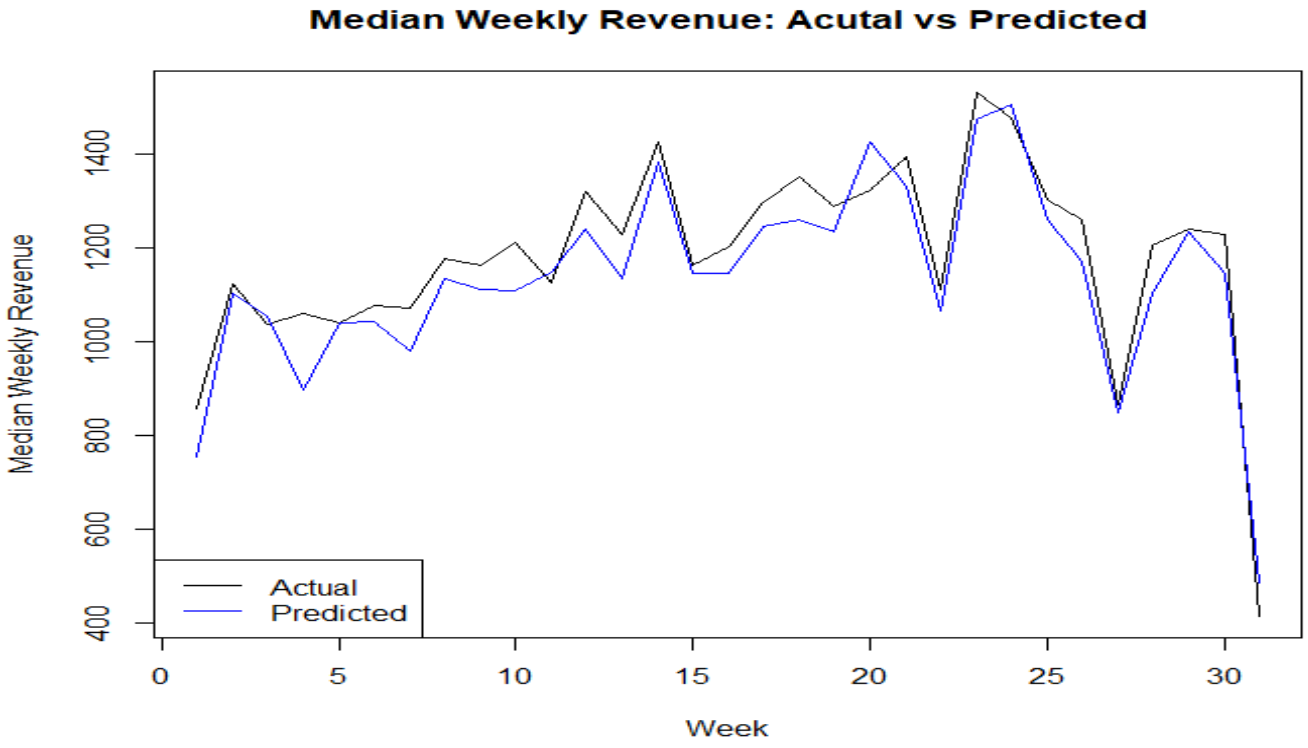


Table 6: Weekly Median Revenue Prediction Accuracy

MAE	59.769
RMSE	69.888
MAPE	0.053

4.2 Daily Median Revenue

4.2.1 Additional Variables

The model from daily median revenue required the creation of many additional variables to control issues such as a bimodal distribution of trip miles, holidays, effect of weekends, and a different behavior in median daily revenue after 2014. A full explanation of all added variables can be found in Table 7.

Table 7: Explanation of Additional Variables

lowTripMiles	Median daily miles if value is < 2.5 and 0 otherwise
highTripMiles	Median daily miles if value is $>= 2.5$ and 0 otherwise
iMilesLow	Indicator: 1 if TripMiles<2.5 and 0 if TripMiles >=2.5
xmas	Indictor for Christmas Day
stpat	Indicator for the Saturday St. Patrick's Day is celebrated
thanks	Indicator for Thanksgiving day
lyft	Indicator: 1 if Lyft was in operation and 0 otherwise
iWeekend	Indicator: 1 if the day is Saturday or Sunday and 0 otherwise
Time	Sequence of integers from 1 to 212
iAfter2014	Indictor: 1 if year is > 2014 and 0 otherwise

Many of the provided variables have been excluded from consideration. An explanation of those choices can be found in Table 12 in Appendix A.

4.2.2 Final Model

A summary of the predictors included in the linear part of the final model can be found in Table 8 below. The time series model applied to the residuals was a SARIMA$(2,0,2)(2,0,0)_7$. The VIFs as well as the diagnostic plots for the time series can be found in Appendix C.

Table 8: Daily Model Coefficients Summary

Predictor	Value	Comments
Day	0.0023	
Day^2	-0.00001	Helps capture the quadratic annual trend
TripSeconds	0.5118	
Tips	4.132	
iMilesLow	-4.317	As expected driving an extremely low number of miles per day decreases revenue
stpat	7.431	St. Patrick's day increases revenue (as seen in [1])
xmas	-0.410	Has a negative impact on revenue
lyft	-1.080	Has a negative impact on revenue
i2015	-0.9811	2015 had a generally lower daily revenue
iWeekend	1.094	Weekends have slightly higher revenue
Day:iMilesLow	0.0055	

Note that we have used a transformation of Trip Total, Trip Seconds, and Tips in our model. The power transformations are 0.6377, 0.4113, and 0.3880 respectively.

4.2.3 Predictions

Applying the final model, we obtain the following predictions. A summary of the models prediction accuracy can be found in Table 9.

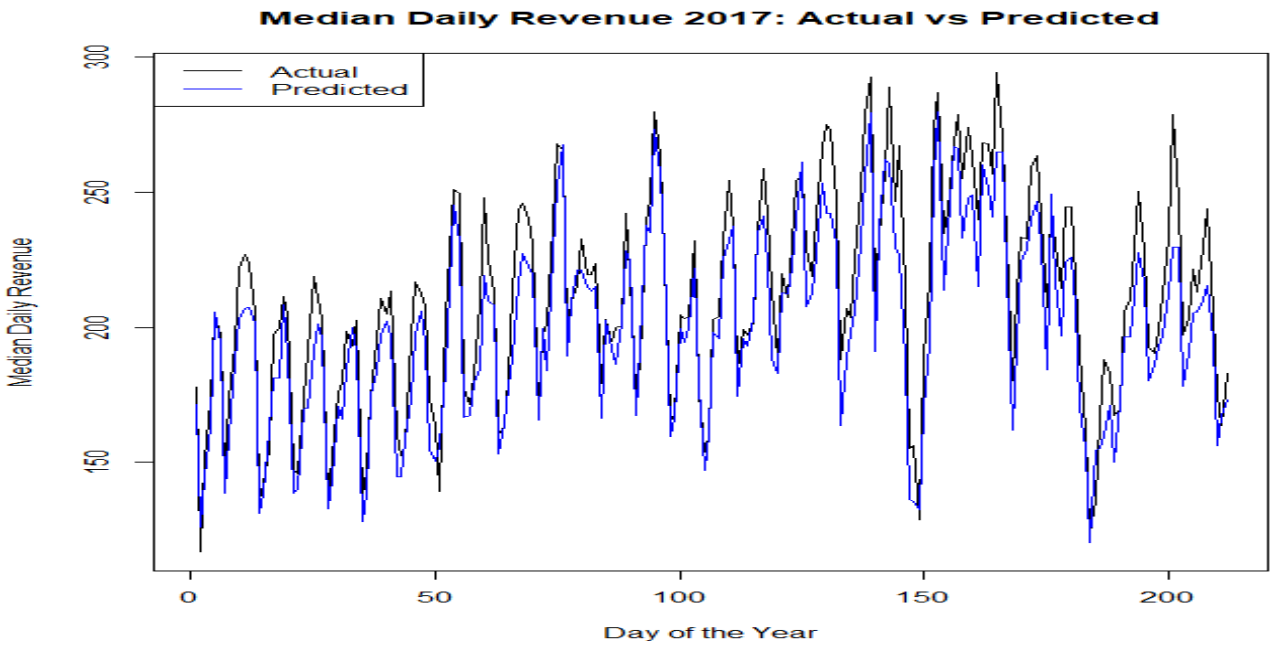


Table 9: Daily Median Revenue Prediction Accuracy

MAE	12.14
RMSE	14.68
MAPE	0.06

From the plot of the fitted values we see that our model does a good job following the overall trend of median daily revenue over time however it misses some of the peaks. The prediction accuracy measures show that our predictions are off by about 6% or equivalently $12.

4.3 Hourly Median Revenue

We present the final model for hourly median revenue built from 35060 hourly observations used to predict the first 5087 hours of 2017.

4.3.1 Added Variables

The hourly data provides the additional issue of Daylight Savings Time (DLS) that needs to be incorporated into the model. There are two consequences of DLS: one hour is lost in March and one hour is counted twice in November. To address this issue we have made sure to adjust any predictor based on hour of the day to account for the loss in March and have added an indictor variable $iDLS$. The definition of $iDLS$ is

$$iDLS = \begin{cases} 1 & \text{if the hour is 1AM on DLS in November} \\ 0 & \text{otherwise} \end{cases}.$$

We also added the following variables

4.3.2 Final Model

Table 10 shows the linear predictors included in the final model. We have also provided their estimates and some comments. The final time series model applied to the residuals was a

Hour of Day	Account for Daily changes
Hour of Year	Account for quadratic annual trend
iHalloween	It fell on DLS and magnified the effect
iRushHour	Indicator: 1 if(6-10a and 3-7p)
iSM	Indictor 6AM Monday morning
iSATSUN	Indicator for 1AM Sunday morning
iCUBS	Indicator for the last sequence of Cubs games before World Series
iEVENING	Indicator for Evening: 1 if (6-11p)
iLATE	Indicator for Late: 1 if (12-2a)
iExtra	Indicator for extra charge: 1 if Extras>0

SARIMA$(4,0,2)(3,0,0)_{24}$. The VIFs as well as the diagnostic plots for the time series can be found in Appendix C.

Table 10: Hourly Model Coefficients Summary

Predictor	**Value**	**Comments**
Trip Miles	2.717	
Trip Seconds	1.664	
Hour of Day	-0.2189	
iExtra	1.258	
iHalloween	4.414	Postive impact on hourly revenue
iLYFT	2.641	Positive sign seems counter intuitive but there is also an interaction with time which is negative
Hour of Year	0.0006	
$(\text{Hour of Year})^2$	-7.12e-8	
iDayLightSavings	3.545	Double counting the hour results in higher hourly total
iCubs	0.847	Positive impact on hourly total
iEvening	2.279	Positive impact on hourly total
iLate	-1.115	Negative impact on hourly total
iSaturdaySunday	-0.7915	Negative impact on hourly total
iSundayMonday	5.379	Positive impact on hourly total
iRushHour	-1.204	Negative impact on hourly total
iLyft*Time	-0.001	Overall negative impact of lyft on hourly total

4.3.3 Predictions

Due to the high number of predictions made we, we split the plot of the predicted versus real values into three separated plot below.

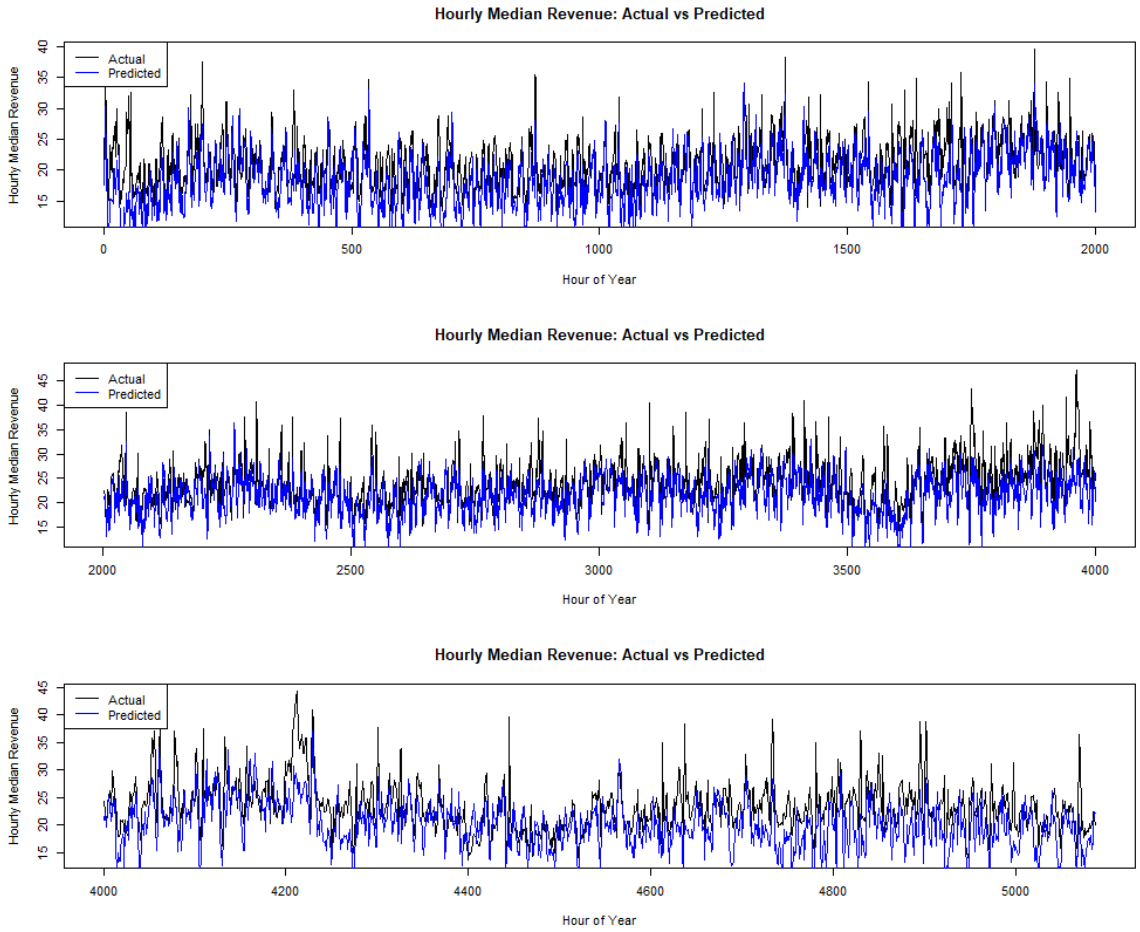


Table 11: Hourly Median Revenue Prediction Accuracy

MAE	3.39
RMSE	4.34
MAPE	0.146

The plot of the actual versus predicted values shows that our model consistently missing the high peaks in the series. Table 11 shows that our average error was about %14 or $3.39.

5 Summary

5.1 Further Work

During our analysis we came upon many surprising items that would be interesting for further study.

- Different years had unique behavior at different time frequencies (2013 - weekly, 2015 - hourly)
- There were 2 weeks with inexplicably low revenue. It would be interesting to see if this was legitimate or a data entry error.

It would also be of interest to fit the linear model and time series portions of the model simultaneously in order to make inference on the model parameters. Doing so one could make stronger claims about the effect of predictors.

5.2 Conclusion

By incorporating external variables to the data provided by the city of Chicago, were able to create predictions of 2017 median weekly revenue with the following accuracy

- Weekly - 5% Average Error
- Daily - 6% Average Error
- Hourly - 15% Average Error

We see that as the time scale gets finer it is more difficult to accurately predict median revenue. This is due to the median revenue being more sensitive to a greater number of factors. This is reflected in the increase in the number of external variables needed in the model as the time scale gets finer. Each model also showed that the introduction of Lyft as a source of competition had a negative impact on median revenue at every time scale.

References

[1] Chicago public taxi data :trends across neighborhoods, the cubs world series, and a good-faith effort to protect privacy. http://toddwschneider.com/posts/chicago-taxi-data/. Accessed: 2018-04-08.

[2] National weather service forecast office, chicago il. http://w2.weather.gov/climate/xmacis.php?wfo=lot. Accessed: 2018-04-07.

[3] Illinois vehicle auto insurance, traffic patterns in chicago. http://www.illinoisvehicle.com/about-us/blog/traffic-patterns-chicago/. Accessed: 2018-04-15.

Appendix A Variables

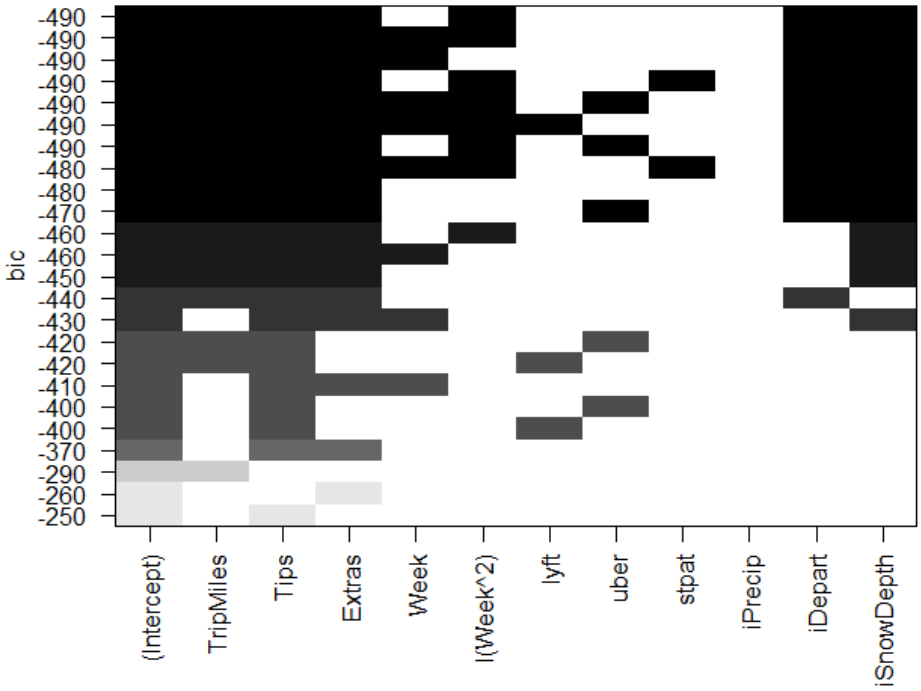
Table 12: Reasons for Not Considering Provided Variables

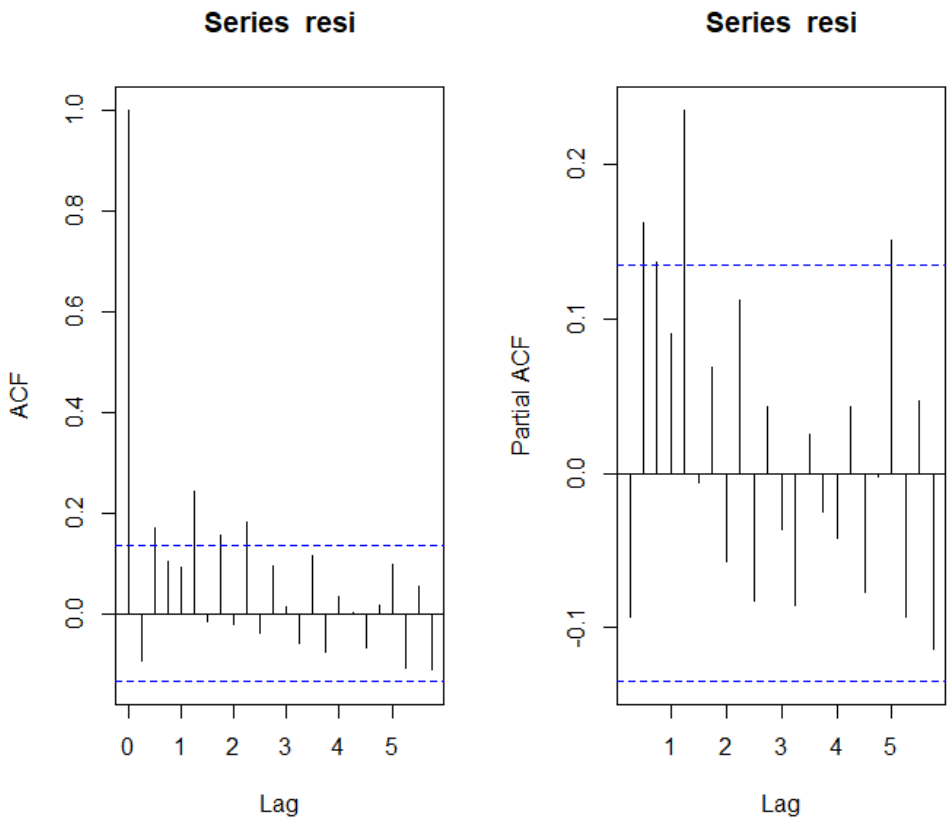
Fare	Median values is sometimes greater than Trip Total causing logical inconsistency
Tolls	Median daily value was always zero
TripMiles	Has a bimodal distribution.

Appendix B Variable Selection

B.1 Weekly

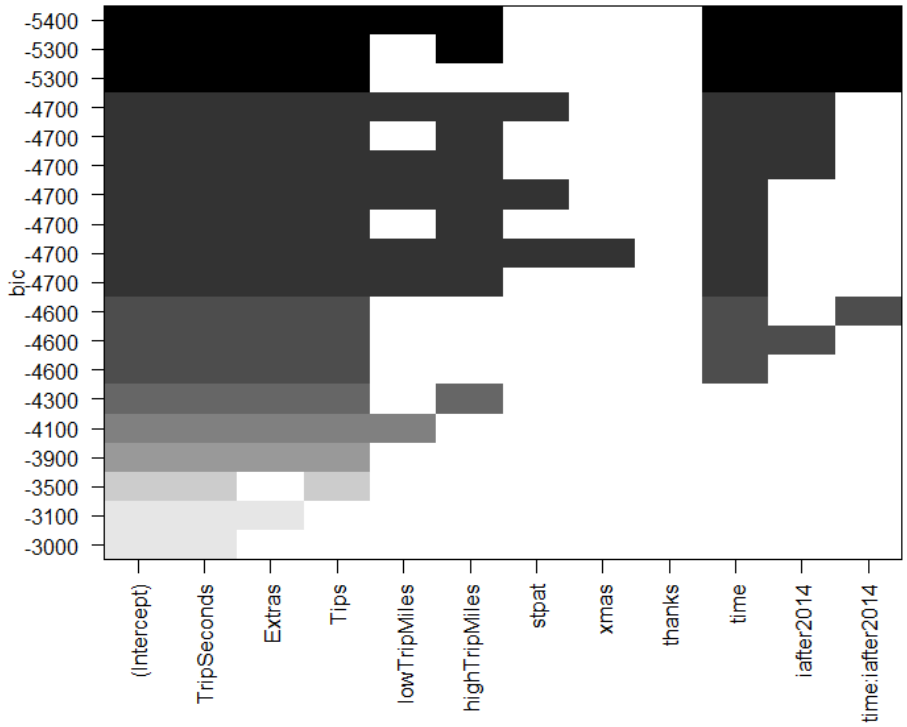
Figure 6: Weekly BIC

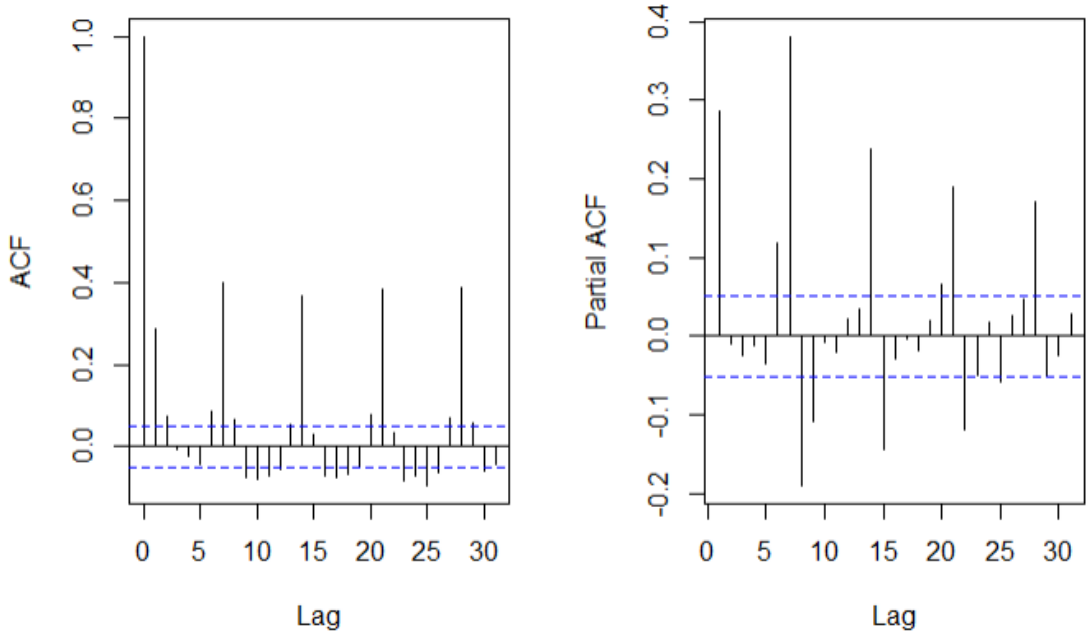



Series resi
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
Lag
Series resi
Partial ACF
0.2
0.1
0.0
-0.1
1
2
3
4
5
Lag

B.2 Daily

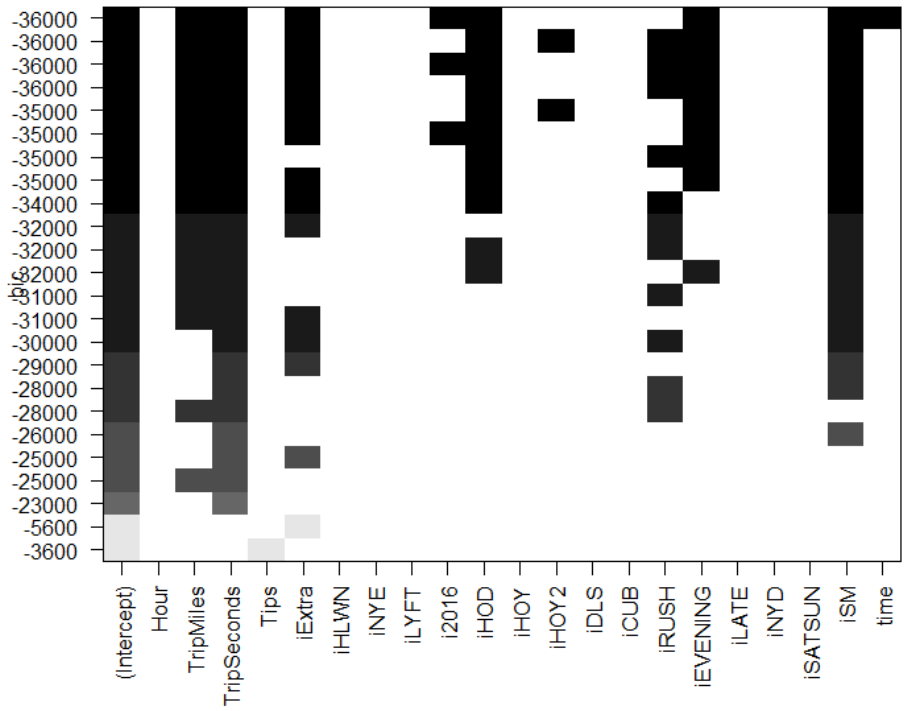
Figure 7: Daily BIC

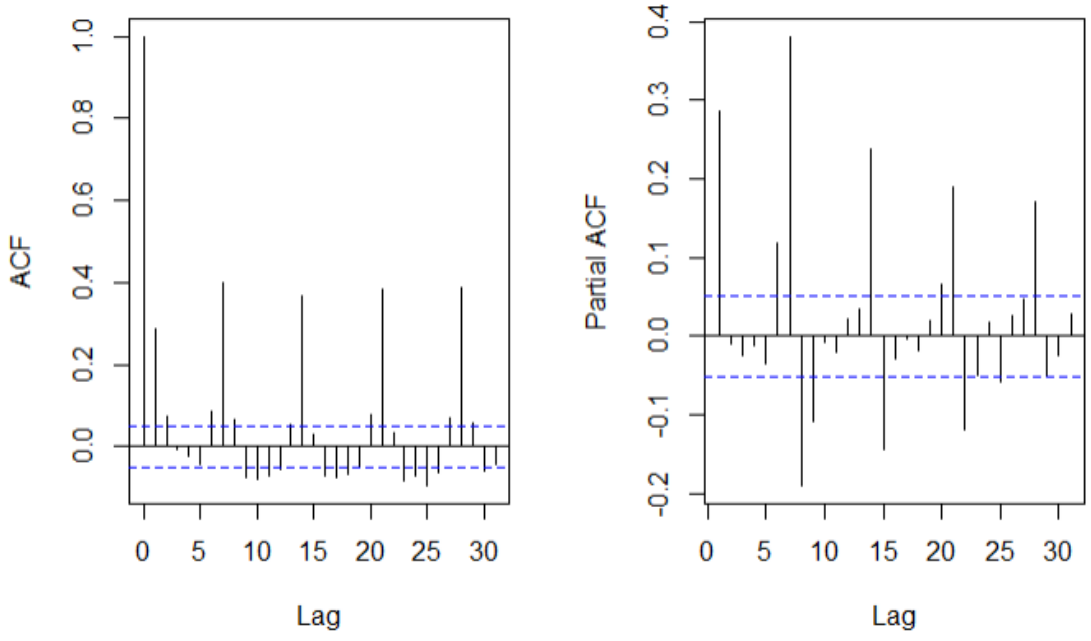




B.3 Hourly

Figure 8: Hourly BIC





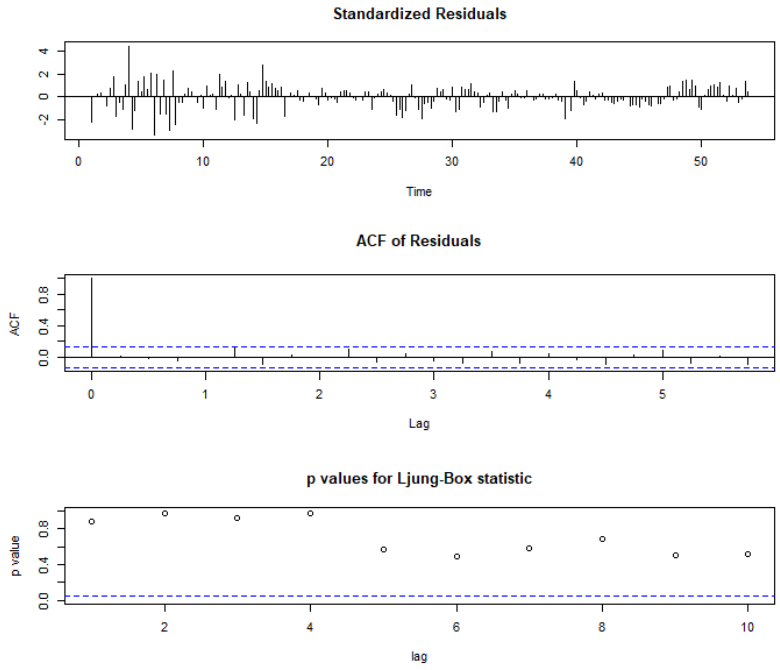
Appendix C Diagnostics

C.1 Weekly

Figure 9: Weekly Model VIFs

TripMiles	Tips	Extras
24.2560	14.4870	12.0870
iDepart	iSnowDepth	i2013
1.2489	1.5631	99.3450
lyft	iHol	i29
13.7500	1.9165	1.1627
i157	i2013:lyft	TripMiles:i2013
1.3501	6.7768	266.1500
Tips:i2013	Extras:i2013	
211.4000	222.6700	

Figure 10: Diagnostic Plots for Weekly Median Revenue Residual Time Series Model

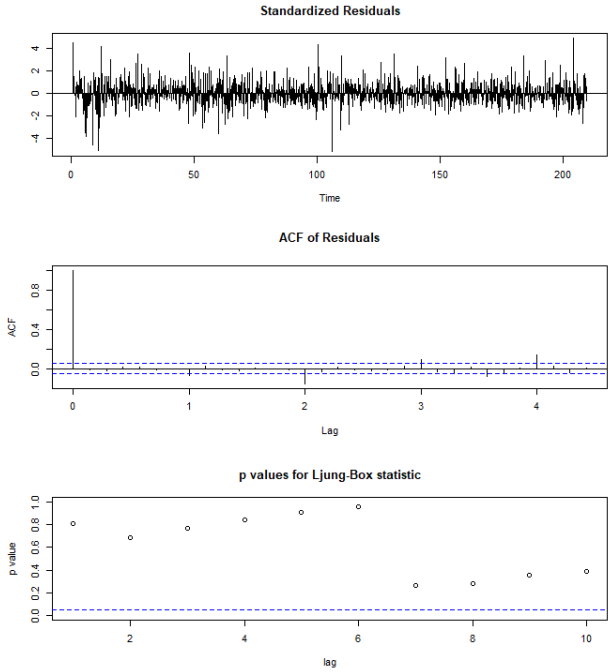


C.2 Daily

Figure 11: Daily Model VIFs

day	I(day^2)	TripSeconds	Tips
20.3770	20.1810	2.0529	2.7723
iMiles_low	stpat	xmas	lyft
5.8493	1.0913	1.1005	1.6959
i2015	iWeekend	day:iMiles_low	
1.1371	1.2619	5.2922	

Figure 12: Diagnostic Plots for Daily Median Revenue Residual Time Series Model

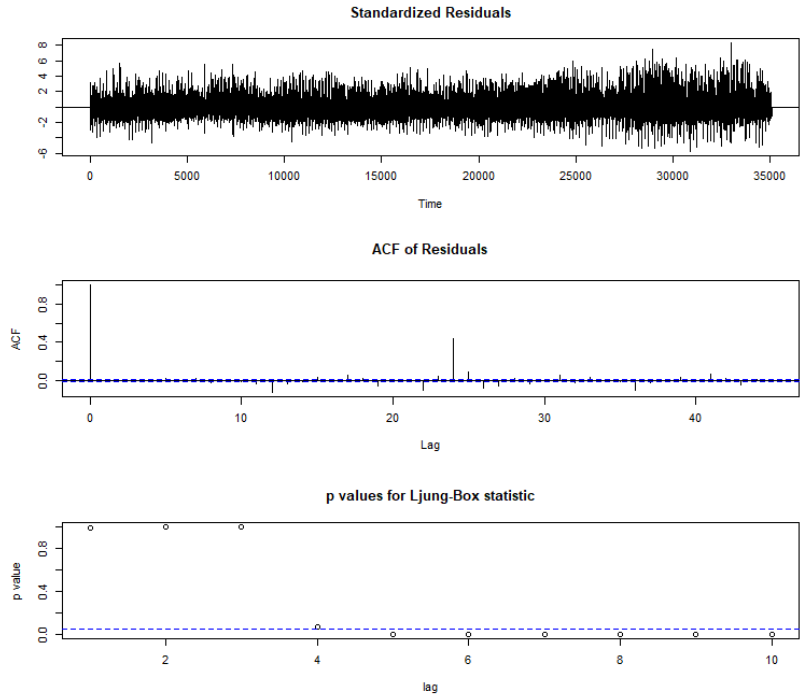


C.3 Hourly

Figure 13: Hourly Model VIFs

TripMiles	TripSeconds	iHOD	iExtra
3.1933	2.0969	4.7773	1.4799
iHLWN	iLYFT	iHOY	iHOY2
1.0017	5.5117	19.8270	18.6410
iDLS	iCUB	iEVENING	iLATE
1.0015	1.0208	2.9223	2.1783
iSATSUN	iSM	iRUSH	time
1.1455	1.0609	1.4539	1637.5000
i2016	iLYFT:time		
3.8446	1761.3000		

Figure 14: Diagnostic Plots for Hourly Median Revenue Residual Time Series Model

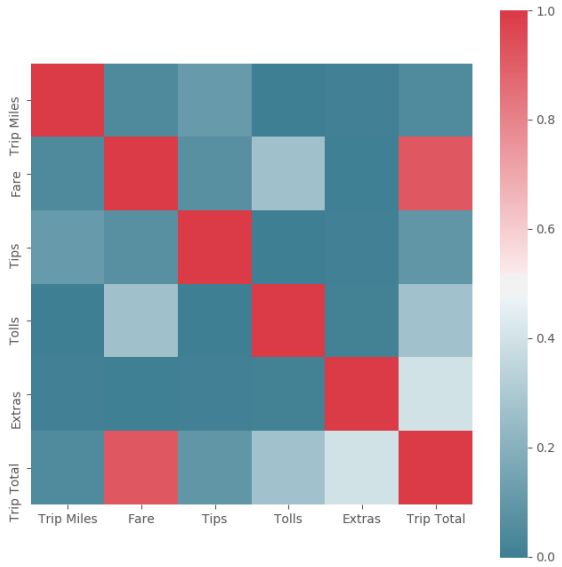


Appendix D Data Summary

D.1 Data Summary

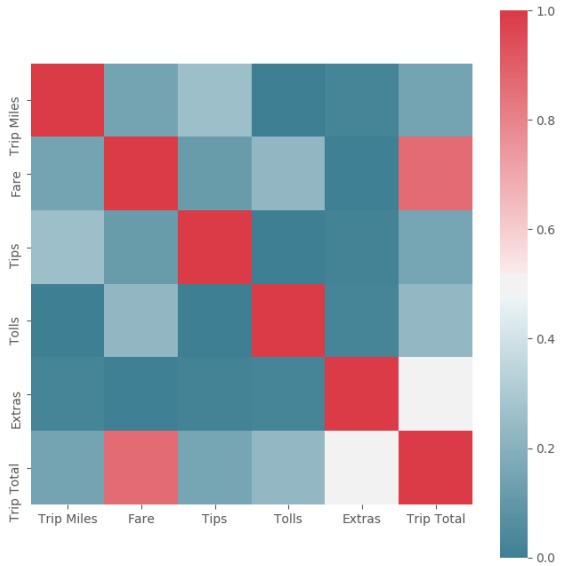
D.1.1 2013

	Trip Miles	Fare	Tips	Tolls	Extras	Trip Total
Count	26870079	26870016	26870016	26870016	26870016	26870016
Mean	2.16	12.45	0.966	0.015	0.85	14.29
Std Deviation	9.72	68.19	2.17	1.95	29.61	75.10
Minimum	0.00	0.00	0.00	0.00	0.00	0.00
25% quartile	0.0	5.85	0.0	0.0	0.0	6.65
50% quartile	0.17	7.85	0.0	0.0	0.0	9.05
75% quartile	1.7	12.45	1.25	0.0	1.0	14.05
Maximum	1998.1	9999.99	444.74	8099.94	9878.76	9999.99



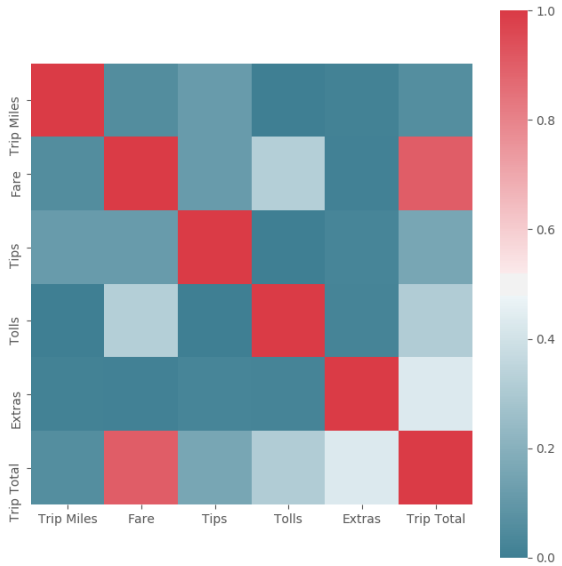
D.1.2 2014

	Trip Miles	Fare	Tips	Tolls	Extras	Trip Total
Count	31021674.0	31021498.0	31021498.0	31021498.0	31021498.0	31021498.0
Mean	2.58	12.28	1.15	0.0068	0.85	14.29
Std Deviation	5.36	43.37	2.31	1.39	25.19	50.90
Minimum	0.0	0.0	0.0	0.0	0.0	0.0
25% quartile	0.0	5.85	0.0	0.0	0.0	6.85
50% quartile	1.0	8.05	0.0	0.0	0.0	9.25
75% quartile	2.6	12.85	2.0	0.0	1.0	14.45
Maximum	1530.4	9929.39	500.0	2807.29	9989.05	9999.82



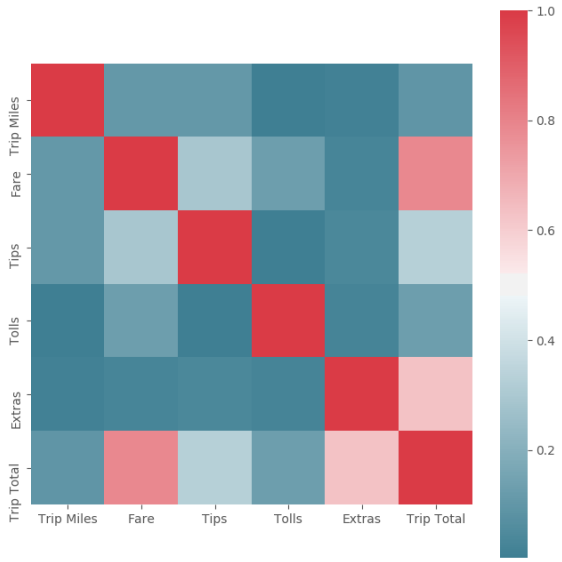
D.1.3 2015

	Trip Miles	Fare	Tips	Tolls	Extras	Trip Total
Count	27400692.0	27400571.0	27400571.0	27400571.0	27400571.0	27400571.0
Mean	3.04	12.73	1.40	0.006	0.89	15.03
Std Deviation	15.94	47.160	2.53	1.03	22.36	53.018
Minimum	0.0	0.0	0.0	0.0	0.0	0.0
25% quartile	0.05	6.05	0.0	0.0	0.0	7.05
50% quartile	1.1	8.05	0.0	0.0	0.0	9.45
75% quartile	2.8	13.25	2.0	0.0	1.0	15.05
Maximum	3460.0	9966.66	596.85	2103.75	9934.23	9966.66



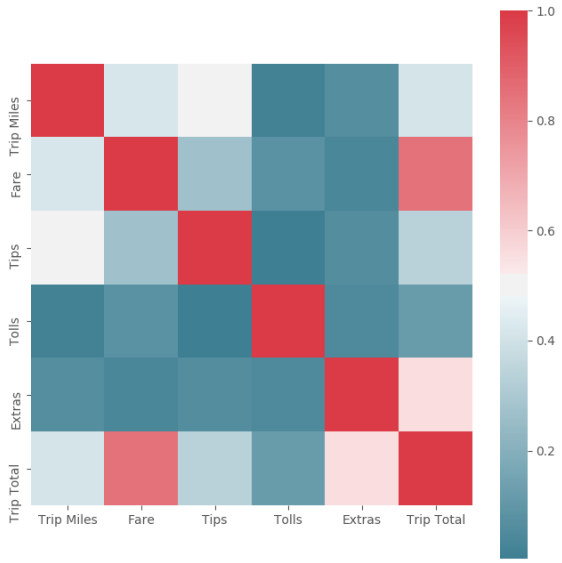
D.1.4 2016

	Trip Miles	Fare	Tips	Tolls	Extras	Trip Total
Count	19878044.0	19877975.0	19877975.0	19877975.0	19877975.0	19877975.0
Mean	3.39	13.89	1.64	0.003	1.03	16.62
Std Deviation	22.60	25.38	2.92	0.56	20.71	34.13
Minimum	0.0	0.0	0.0	0.0	0.0	0.0
25% quartile	0.1	6.25	0.0	0.0	0.0	7.4
50% quartile	1.1	8.5	0.0	0.0	0.0	10.0
75% quartile	2.7	14.25	2.0	0.0	1.0	16.5
Maximum	3353.1	9999.0	496.5	999.99	9993.41	9999.0

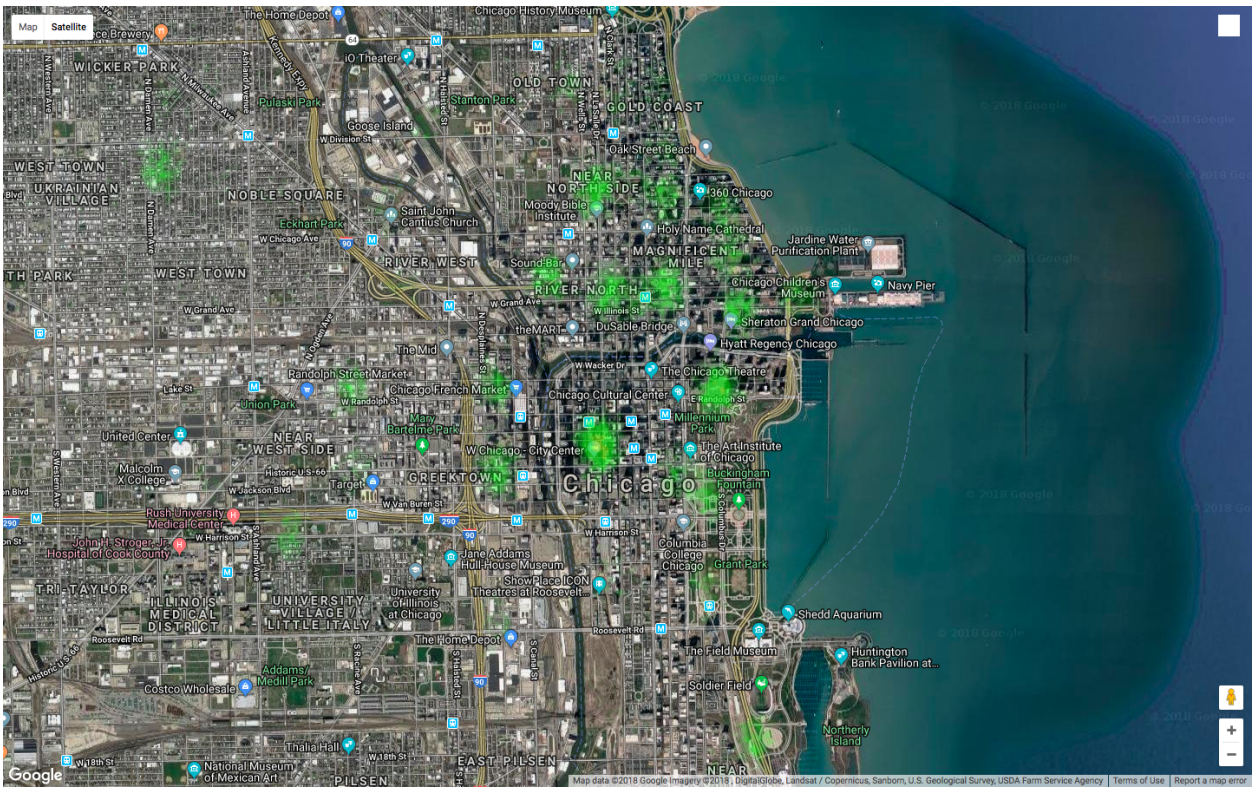


D.1.5 2017

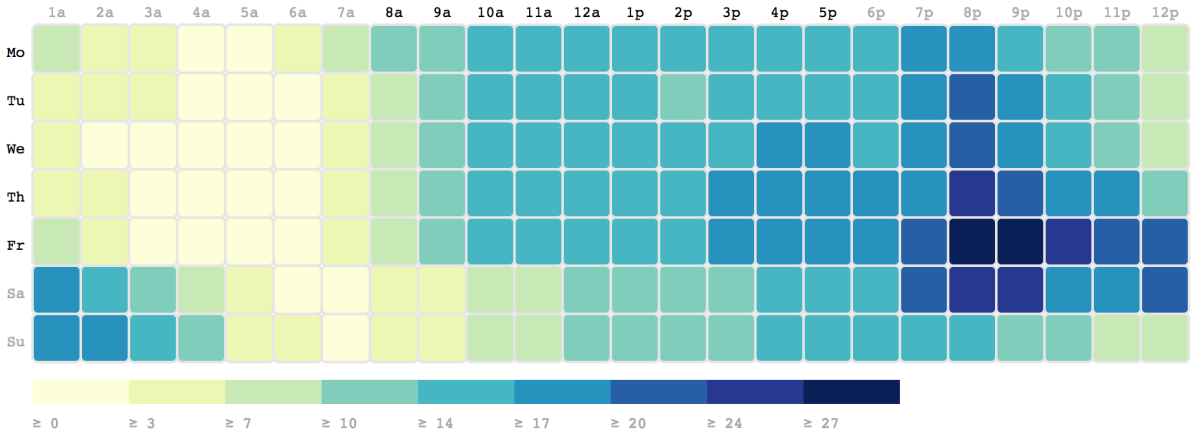
	Trip Miles	Fare	Tips	Tolls	Extras	Trip Total
Count	7688861.0	7688847.0	7688847.0	7688847.0	7688847.0	7688847.0
Mean	3.14	13.51	1.65	0.004	1.08	16.36
Std Deviation	5.44	27.04	2.91	1.02	17.683	33.82
Minimum	0.0	0.0	0.0	0.0	0.0	0.0
25% quartile	0.42	6.0	0.0	0.0	0.0	7.25
50% quartile	1.2	8.25	0.0	0.0	0.0	9.75
75% quartile	2.8	13.5	2.0	0.0	1.0	15.75
Maximum	1234.6	9600.58	800.0	999.99	9864.84	9872.09

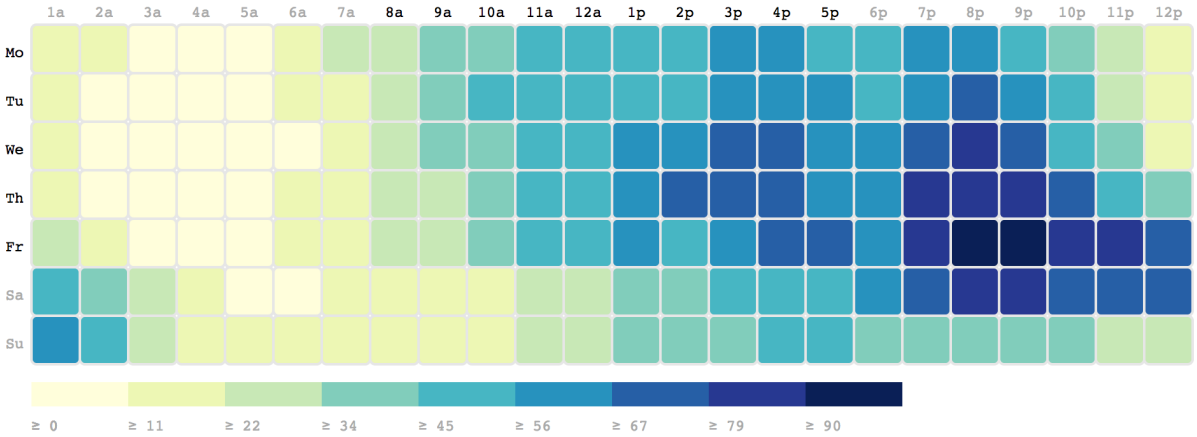


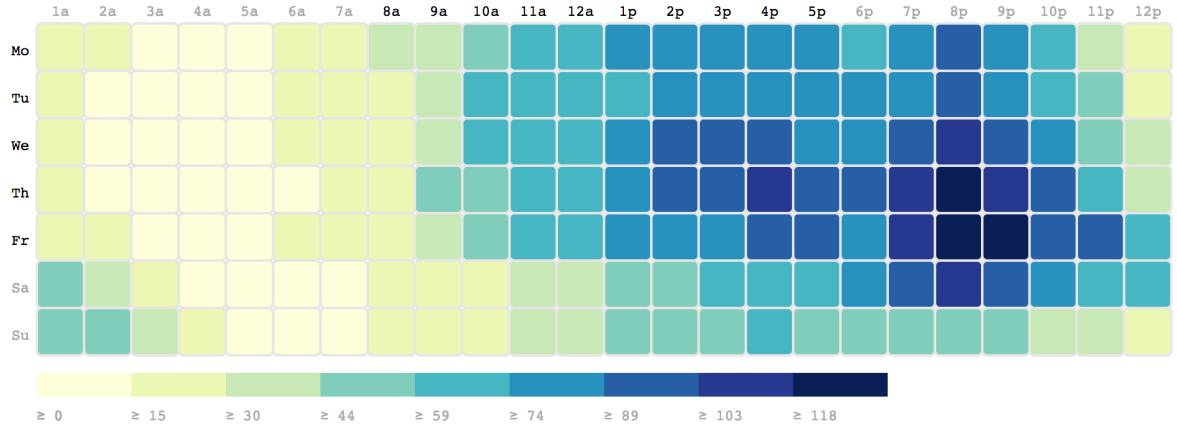
Appendix E Visualizations

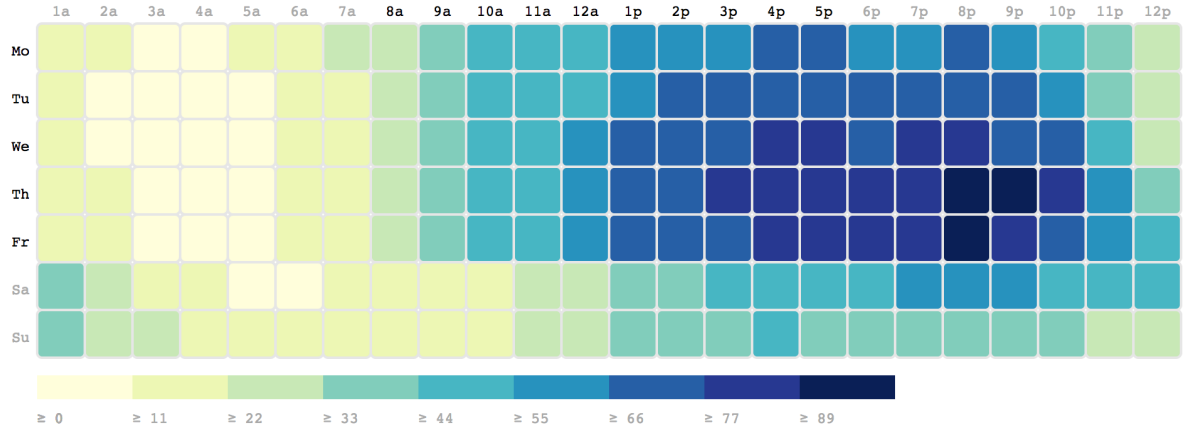


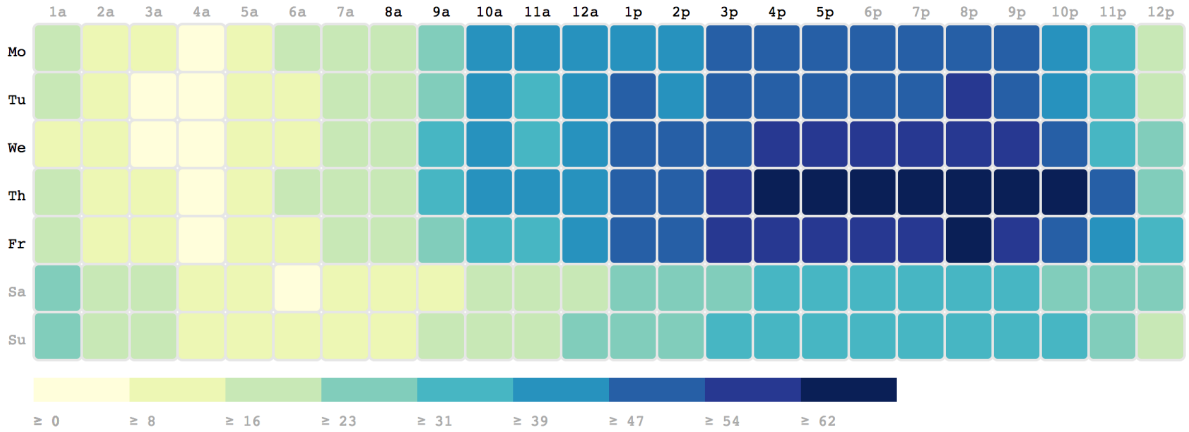
E.1 Trip Miles Visualization



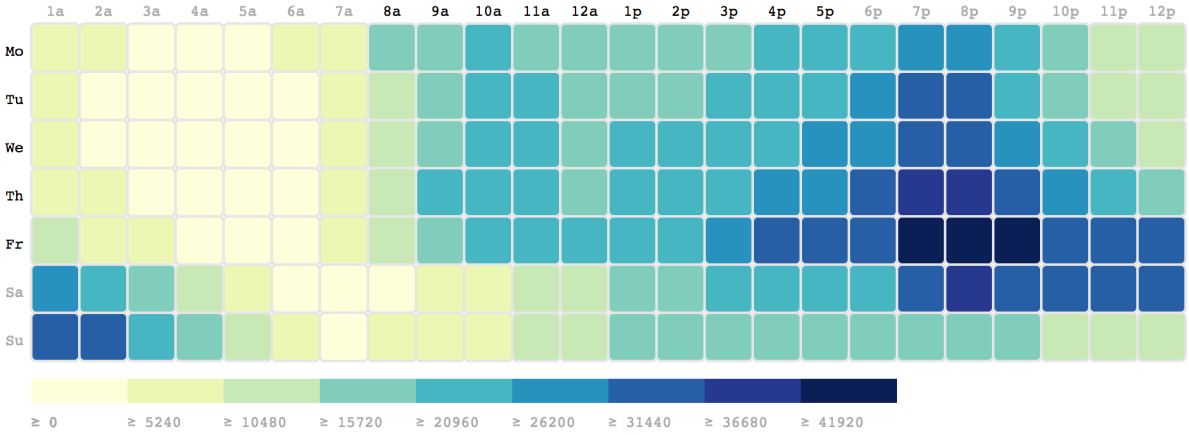

1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0 ≥ 11 ≥ 22 ≥ 34 ≥ 45 ≥ 56 ≥ 67 ≥ 79 ≥ 90

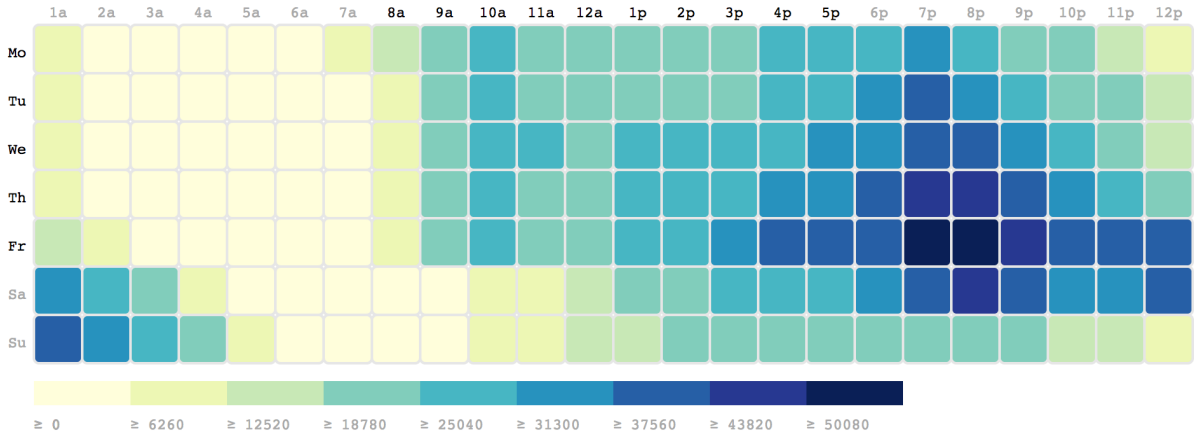

1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0 ≥ 15 ≥ 30 ≥ 44 ≥ 59 ≥ 74 ≥ 89 ≥ 103 ≥ 118

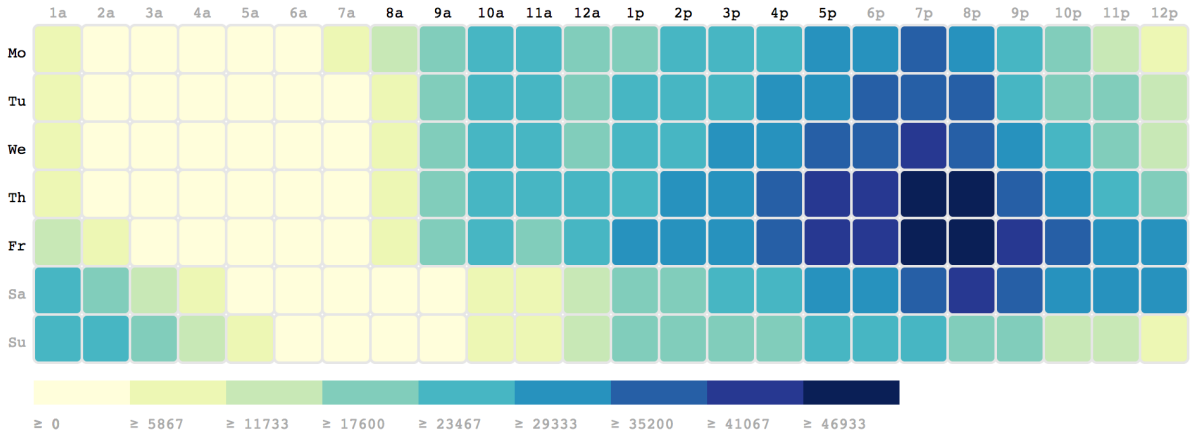

1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0 ≥ 11 ≥ 22 ≥ 33 ≥ 44 ≥ 55 ≥ 66 ≥ 77 ≥ 89

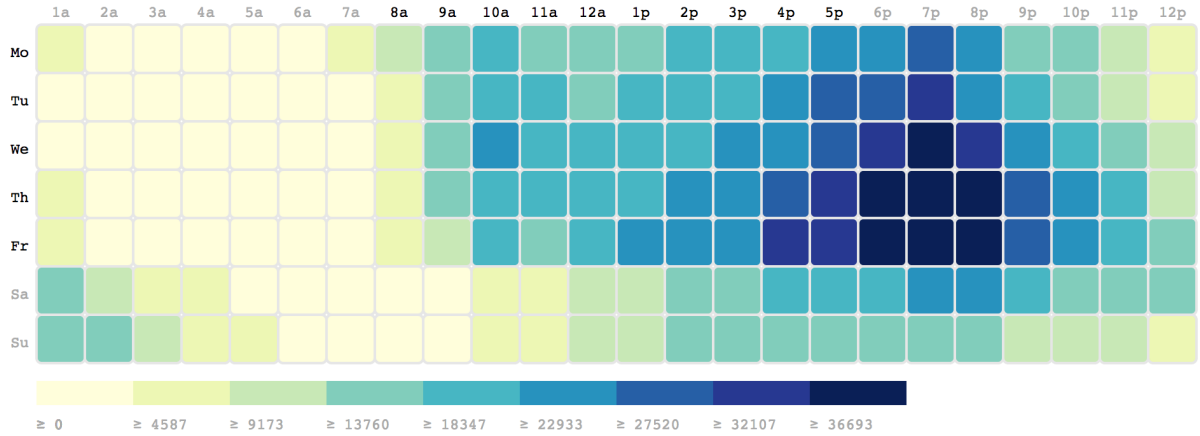


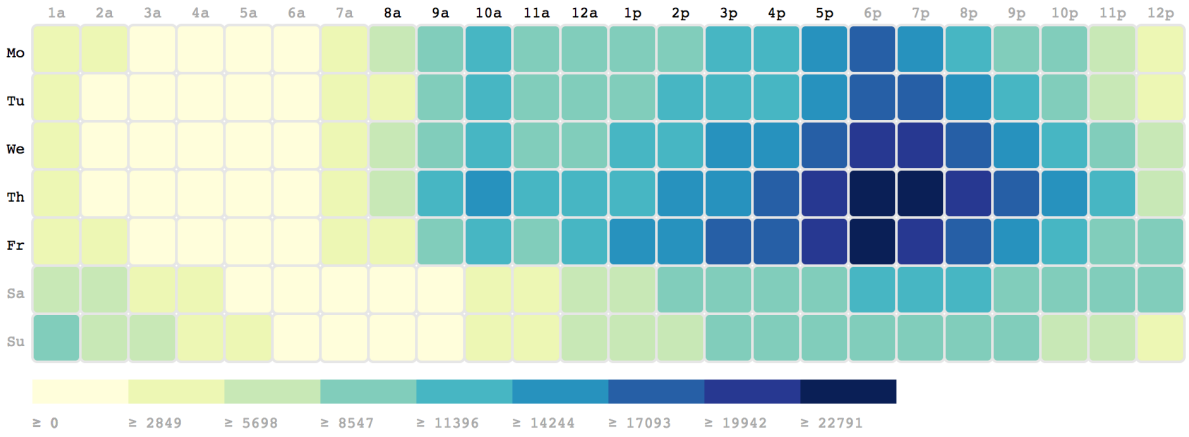
E.2 Trip Seconds Visualization






1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0
≥ 5867
≥ 11733
≥ 17600
≥ 23467
≥ 29333
≥ 35200
≥ 41067
≥ 46933


1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0
≥ 4587
≥ 9173
≥ 13760
≥ 18347
≥ 22933
≥ 27520
≥ 32107
≥ 36693


1a 2a 3a 4a 5a 6a 7a 8a 9a 10a 11a 12a 1p 2p 3p 4p 5p 6p 7p 8p 9p 10p 11p 12p
Mo
Tu
We
Th
Fr
Sa
Su
≥ 0
≥ 2849
≥ 5698
≥ 8547
≥ 11396
≥ 14244
≥ 17093
≥ 19942
≥ 22791

Appendix F Additional Considerations

We considered the pickups and drop-offs from and to Downtown Chicago, Suburbs, rides from suburbs to airport and from not airport to suburbs.

To calculate these indicators, we got the latitudes and longitudes of downtown locations and suburbs. The downtown of Chicago forms a closed polygon and we were able to calculate whether a pickup or a drop-off was from downtown or not by calculating if the point lied inside the downtown polygon or outside of it.

Similarly, for calculating whether a pickup or drop-off was from suburb, we checked if that point lied outside the main chicago city. The suburbs is the area mainly comprises of the outlying districts of the city.

However, these indicators when aggregated together did not have an impact on revenue. But these indicators can be useful to predict the individual rides.

Appendix G Code

G.1 Weekly

```
---
title: "Chicago Taxi Competition 2018: Data Exploration"
author: "The Outliers"
date: "April 10, 2018"
output: pdf_document
---
```

```
```{r part1, echo=FALSE,warning=FALSE}
#Load Data
library(readxl)
library(readr)
numTaxi <- read_excel("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/allYears_uniqueTaxi.csv.xlsx")
median2013 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Medians/2013_median.csv")
median2014 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Medians/2014_median.csv")
median2015 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Medians/2015_median.csv")
median2016 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Medians/2016_median.csv")


rownames = unlist(median2013[,1])
rownames = unname(rownames)
median2013 = as.matrix(median2013[,-1])
median2013 = t(median2013)
rownames[5:7] = c("TripTotal","TripSeconds","TripMiles")
median2013 = as.data.frame(median2013)
colnames(median2013) = rownames


rownames = unlist(median2014[,1])
rownames = unname(rownames)
median2014 = as.matrix(median2014[,-1])
median2014 = t(median2014)
rownames[2] = "TripSeconds"
rownames[6:7] = c("TripMiles","TripTotal")
median2014 = as.data.frame(median2014)
colnames(median2014) = rownames


rownames = unlist(median2015[,1])
rownames = unname(rownames)
median2015 = as.matrix(median2015[,-1])
median2015 = t(median2015)
rownames[3] = "TripSeconds"
rownames[6:7] = c("TripMiles","TripTotal")
median2015 = as.data.frame(median2015)
colnames(median2015) = rownames
```
```

```
rownames = unlist(median2016[,1])
rownames = unname(rownames)
median2016 = as.matrix(median2016[,-1])
median2016 = t(median2016)
rownames[1] = "TripSeconds"
rownames[5] = "TripMiles"
median2016 = as.data.frame(median2016)
colnames(median2016) = rownames

Tips = c(median2013$Tips,median2014$Tips,median2015$Tips, median2016$Tips)
Tolls = c(median2013$Tolls,median2014$Tolls,median2015$Tolls,median2016$Tolls)
Extras = c(median2013$Extras,median2014$Extras,median2015$Extras,median2016$Extras)
Fare = c(median2013$Fare,median2014$Fare,median2015$Fare,median2016$Fare)
TripMiles = c(median2013$TripMiles,median2014$TripMiles,median2015$TripMiles,median2016$TripMiles)
TripSeconds = c(median2013$TripSeconds,median2014$TripSeconds,median2015$TripSeconds,median2016$TripSecon
ds)
TripTotal = c(median2013$TripTotal,median2014$TripTotal,median2015$TripTotal,median2016$TripTotal)

dat = cbind(TripTotal,TripMiles,TripSeconds,Fare,Tips,Tolls,Extras)
Week = rep((1:53),4)
dat = cbind(dat,Week)
dat = as.data.frame(dat)

write.csv(TripTotal,"response_weekly.csv")

write.csv(dat,"basic_predictors_weekly.csv")
```

#Additional Variables

```{r code1}
#variable representing weeks since uber introduction to chicago
uber = seq(from = 65, to = 276, by = 1)

#variable representing weeks since lyft introduction to chicago
lyft = c(rep(0,17),seq(from=1,to=195,by=1))

#week of st pat celebration
sp2013 = rep(0,53)
sp2014 = rep(0,53)
sp2015 = rep(0,53)
sp2016 = rep(0,53)

sp2013[11] = 1
sp2014[11] = 1
sp2015[11] = 1
sp2016[11] = 1

stpat = c(sp2013,sp2014,sp2015,sp2016)
```

```
weather <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/weatherDat.csv")

plot(weather$Precip,dat$TripTotal)
plot(weather$Depart,dat$TripTotal)
plot(weather$SnowDepth,dat$TripTotal)

indexP = which(weather$Precip>5)
indexD = which(weather$Depart>40)
indexS = which(weather$SnowDepth>2)

iPrecip = rep(0,212)
iDepart = rep(0,212)
iSnowDepth = rep(0,212)

iPrecip[indexP] = 1
iDepart[indexD] = 1
iSnowDepth[indexS] = 1

iweather = cbind(iPrecip, iDepart, iSnowDepth)

iweather = as.data.frame(iweather)


iLast = rep(0,212)
iLast[c(53,106,159,212)] = 1

iHol = rep(0,212)
iHol[50:53] = 1
iHol[103:106] = 1
iHol[156:159] = 1
iHol[209:212] = 1

i29 = rep(0,212)
i29[29] = 1
i157 = rep(0,212)
i157[157] = 1

dat = cbind(dat,uber,lyft, stpat, iweather, iLast, iHol, i29, i157)


```

```{r plots2}
x = 1:212
week = rep((1:53),4)

plot(x,dat$TripTotal,type="l",main="Median Weekly Trip Total",xlab="Week Number",xaxt="n", ylab = "Trip Total")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
par(mfrow=c(3,2))
plot(x,dat$Tips,type="l",main="Median Weekly Tips",xlab="Week Number",xaxt="n", ylab = "Tips" )
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
plot(x,dat$TripMiles,type="l",main="Median Weekly Distance Traveled (Miles)",xlab="Week Number",xaxt="n",ylab="Distance Traveled (Miles)")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
plot(x,dat$TripSeconds,type="l",main="Median Weekly Time Traveld (seconds)",xlab="Week Number",xaxt="n", ylab = "Time Traveled (seconds)")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
plot(x,dat$Fare,type="l",main="Median Weekly Total Fare",xlab="Week Number",xaxt="n", ylab = "Total Fare")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
plot(x,dat$Tolls,type="l",main="Median Weekly Total Tolls",xlab="Week Number",xaxt="n", ylab = "Total Tolls")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
plot(x,dat$Extras,type="l",main="Median Weekly Total Extra Charges",xlab="Week Number",xaxt="n", ylab = "Total Extra Charges")
axis(1, at=1:212, labels=week)
abline(v=53,col="red")
abline(v=106,col="red")
abline(v=159,col="red")
```

```{r model1}
#Check for Needed Transformations
varlist = c("TripTotal","TripMiles","TripSeconds","Fare","Tips","Tolls","Extras","Week","uber","lyft","stpat","iPrecip","iDepart","iSnowDepth")

par(mfrow=c(4,4))
for(i in 1:14){
  hist(dat[,i],main = varlist[i])
}
par(mfrow=c(1,1))


#Compute Weights for Weighted Least Squares
wts = unlist(c(numTaxi[,2],numTaxi[,3],numTaxi[,4],numTaxi[,5]))
wts = 1/wts


par(mfrow=c(4,4))
for(i in 1:14){
  hist(dat[,i]*wts,main = varlist[i])
}
par(mfrow=c(1,1))


```

Now we can we our full weighted least squares model.

```{r model2}
#Full Model using Weighted Least Squares
fit = lm(TripTotal~TripMiles+TripSeconds+Tips+Tolls+Extras+ lyft +uber + Week+ I(Week^2) + stpat +iPrecip
+ iDepart + iSnowDepth, data=dat,weights = wts)

summary(fit)
```

```{r model3}

#Try to Use Stepwise
require(leaps)
leaps = regsubsets(TripTotal~TripMiles+Tips+Extras+Week+ I(Week^2)+ lyft + uber + stpat + iPrecip +
iDepart + iSnowDepth, data=dat,weights = wts,nbest = 3)


plot(leaps,scale="bic")

i2013 = c(rep(1,53),rep(0,159))
dat = cbind(dat, i2013)

fit2 = lm(TripTotal ~  TripMiles  + Tips + Extras + iDepart + iSnowDepth +i2013 +lyft +i2013*lyft+iHol
+i29 + i157 + TripMiles*i2013 +Tips*i2013 +Extras*i2013 , data=dat)


summary(fit2)
plot(fit2$residuals,type="l")
library(DAAG)
vif(fit2)
lin.coef = fit2$coefficients


#------------------------------------------------------------------------

resi = fit2$residuals
resi = ts(resi, frequency = 4)

aafit = auto.arima(resi,seasonal=TRUE)

tsdiag(aafit)

ts.coef = aafit$coef
```

```{r predict}

Wpred2017 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Predicting/predData2017.csv")


Wpred2017 = Wpred2017[,-1]


Wpred2017 = as.matrix(Wpred2017)

lin.coef = fit2$coefficients
lin.coef = as.matrix(lin.coef)

lin.part = Wpred2017%*%lin.coef

set.seed(1234)
ts.part = simulate(aafit,nsim=31)

pred2017 = lin.part[,1] + ts.part

y2017 = c(855.05, 1123.8, 1035.825, 1058.14,  1039.38,  1077.855, 1072.1, 1177.135, 1163.66,  1211.3,
1126.1, 1320.805, 1227.7, 1426.25,  1161.4, 1203.325, 1297.4, 1352.15,  1288.16,  1322.96,  1393.65,
1110.055, 1531.775, 1478.255, 1302.72,  1259.93,  863.88, 1204.735, 1240.965, 1227.65,  413.265)


plot(1:31,y2017, type="l", xlab = "Week", ylab = "Median Weekly Revenue", main = "Median Weekly Revenue:
Acutal vs Predicted")
lines(1:31,pred2017,col="blue")
legend("bottomleft",c("Actual","Predicted"),lty=c(1,1),col=c(1,4))

MSE = sum((pred2017-y2017)^2)
MAE = mean(abs(pred2017-y2017))
RMSE = sqrt(mean((pred2017-y2017)^2))
MAPE = mean(abs((y2017-pred2017)/(y2017)))
```

G.2 Daily

```
library(readr)
Dmedians_2013 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/Original_Data/Dm
edians_2013.csv")
Dmedians_2014 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/Original_Data/Dm
edians_2014.csv")
Dmedians_2015 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/Original_Data/Dm
edians_2015.csv")
Dmedians_2016 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/Original_Data/Dm
edians_2016.csv")
Dmedians_2017 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/Original_Data/Dm
edians_2017.csv")

fullData = rbind(Dmedians_2013, Dmedians_2014, Dmedians_2015, Dmedians_2016)

fullData = as.data.frame(fullData)
write.csv(fullData, "untran_daily_preds.csv")
```

```
# Transformations

```{r combine}
varlist = c("Day", "TripTotal","TripMiles","TripSeconds", "Fare", "Tolls", "Extras", "Tips")
for(i in 2:8){
  hist(fullData[,i],main = varlist[i])
}


library(forecast)
lambdas = rep(0,6)
for(i in 2:8){
  if(i != 6){
  lambdas[i] = BoxCox.lambda(fullData[,i])
  print(varlist[i])
  print(lambdas[i])
  }
}


for(i in 2:8){
  if(i != 6){
    if(i !=3){
    hist(fullData[,i]^lambdas[i], main = varlist[i])
    }
    if(i == 3){
      hist(log(fullData[,i]),main = varlist[i])
    }
  }
}
```
```

```
#Apply Transformations
for(i in 2:8){
  if(i != 6){
    if(i !=3){
    fullData[,i] = fullData[,i]^lambdas[i]
    }
    if(i == 3){
      fullData[,i] = log(fullData[,i])
    }
  }
}

```

Inferred Variables

```{r infer}
lowTripMiles = rep(0, 1461)
lowTripMiles[which(fullData$`Trip Miles`<2.5)] = log(fullData$`Trip Miles`[which(fullData$`Trip
Miles`<2.5)])

highTripMiles = rep(0, 1461)
highTripMiles[which(fullData$`Trip Miles`>2.5)] = log(fullData$`Trip
Miles`[which(fullData$`Trip Miles`>2.5)])

iMiles_low  = rep(0, 1461)
iMiles_low[which(fullData$`Trip Miles`<2.5)] = 1
```

```{r external}
stpat = rep(0, 1461)
xmas = rep(0, 1461)
thanks = rep(0, 1461)

stpat[c(75, 439, 803, 1167)] = 1

xmas[c(74,75,723,724,1088,1089,1453,1454)] = 1
#1819, 1820 for 2017

thanks[c(332, 696, 1060, 1424)] = 1
# 1789 for 2017

time = seq(1, 1461)

iafter2014 = c(rep(0,365*2),rep(1,365),rep(1,366))

lyft = rep(0, 1461)
lyft[131:1461] = 1
```

```
iweekend = c(rep(0,4),rep(c(1,1,0,0,0,0,0),208),1)


```

```{r combine}
fullDataHold = fullData[,-c(3,5,6)]

modeling_fullData = cbind(fullDataHold,lowTripMiles, highTripMiles, iMiles_low, stpat, xmas,
thanks, time, iafter2014, lyft,i2015,iweekend)


modeling_fullData = as.data.frame(modeling_fullData)

colnames(modeling_fullData) = c("day", "TripTotal","TripSeconds","Extras", "Tips",
"lowTripMiles","highTripMiles","iMiles_low","stpat","xmas","thanks","time","iafter2014","lyft",
"i2015","iweekend")
```

Linear Model

```{r dailyLinear}
fitfull = lm(TripTotal~TripSeconds + Extras + Tips + lowTripMiles + highTripMiles + iMiles_low
+ stpat + xmas + thanks + time + iafter2014 + time*iafter2014 + lyft + iWeekend, data =
modeling_fullData)
```

```{r selection}
require(leaps)

leaps1 = regsubsets(TripTotal~TripSeconds + Extras + Tips + lowTripMiles + highTripMiles +
stpat+ xmas + thanks + time + iafter2014 + time*iafter2014, data = modeling_fullData,
method="backward", nbest = 3)

plot(leaps1, scale = 'bic')


leaps2 = regsubsets(TripTotal~ TripSeconds + Extras + Tips + iMiles_low + stpat+ xmas + thanks
+ time + iafter2014 + time*iafter2014, data = modeling_fullData, method="exhaustive",nbest = 3)

plot(leaps2, scale = 'bic')
```

```
leaps2 = regsubsets(TripTotal~ TripSeconds + Extras + Tips + iMiles_low + stpat+ xmas + thanks
+ time + iafter2014 + time*iafter2014, data = modeling_fullData, method="exhaustive",nbest = 3)

plot(leaps2, scale = 'bic')

step(fitfull, direction= "backward")
#(Intercept)            day    TripSeconds         Extras
#    20.739584      -0.003345       0.415266     -26.389009
#         Tips  lowTripMiles highTripMiles    iMiles_low
#     4.271787     -0.402446     -1.912166     -0.907503
#        stpat
#    -0.497341


step(fitfull, direction = "forward")
#  (Intercept)            day    TripSeconds         Extras
#    20.739584      -0.003345       0.415266     -26.389009
#         Tips   lowTripMiles  highTripMiles     iMiles_low
#     4.271787      -0.402446      -1.912166      -0.907503
#        stpat
#    -0.497341

step(fitfull, direction = "both")
#  (Intercept)            day    TripSeconds         Extras
#    20.739584      -0.003345       0.415266     -26.389009
#         Tips   lowTripMiles highTripMiles    iMiles_low
#    4.271787     -0.402446     -1.912166     -0.907503
#        stpat
#   -0.497341
```

```
```{r reduced}
red1 = lm(TripTotal~ TripSeconds + Extras + Tips + iMiles_low + stpat+ xmas  + time +
iafter2014 + time*iafter2014, data = modeling_fullData)

red2 = lm(TripTotal~ TripSeconds + Extras + Tips + iMiles_low + stpat+ xmas  + time, data =
modeling_fullData)



red3 = lm(TripTotal~ day + I(day^2) + TripSeconds + Tips  +  iMiles_low + stpat+ xmas + lyft +
iMileslowday + i2015 + iWeekend, data = modeling_fullData)

#----------FINAL MODEL-----------------------------------------------------#
red4 = lm(TripTotal~ day + I(day^2) + TripSeconds + Tips  +  iMiles_low + stpat+ xmas + lyft  +
i2015 + iWeekend, data = modeling_fullData)
#---------FINAL MODEL-----------------------------------------------------#

anova(red4, red3)
```
```

```
anova(red4,red3)

summary(red1)
summary(red2)
summary(red3)
summary(red4)

library(DAAG)
vif(red1)
vif(red2)
vif(red3)
vif(red4)

plot(red1)
plot(red2)
plot(red3)

plot(red1$residuals, type = "l",main="Residuals for Initial Linear Model")
plot(red2$residuals, type="l")
plot(red3$residuals, type="l")
plot(red4$residuals, type="l",main="Residuals for Initial Linear Model + Day, Day^2, i2015")

par(mfrow=c(1,2))
acf(red4$residuals)
pacf(red4$residuals)

resi = red4$residuals


lin.coef = red4$coefficients

```

```{r timeseries}
day2 = modeling_fullData$day^2
iMiles_lowday = modeling_fullData$iMiles_low*modeling_fullData$day

finalData = cbind(modeling_fullData$day, day2, modeling_fullData$TripSeconds,
modeling_fullData$Tips, modeling_fullData$iMiles_low, modeling_fullData$stpat,
modeling_fullData$xmas,modeling_fullData$lyft,iMiles_lowday,modeling_fullData$i2015
,modeling_fullData$iweekend)

finalData = as.data.frame(finalData)

colnames(finalData) = c("Day","Day2","TripSeconds","Tips","iMiles_low","stpat","xmas","lyft",
"iMiles_lowday","i2015","iweekend")
```

```
y = red4$residuals

y = ts(y, frequency = 7)

armafit = arima(y,order=c(2,0,2),seasonal=c(2,0,0))

armafit$coef
armafit$sigma2

tsdiag(armafit)
```

```{r predictions}
library(readr)
dailyPredData <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Daily/dailyPredData.csv")

pred2017 = dailyPredData[,-1]
Intercept = rep(1,212)
pred2017 = cbind(Intercept,pred2017)
pred2017 = as.matrix(pred2017)

TripTotal_2017 = (Dmedians_2017$`Trip Total`)^(0.63771005)

lin.coef = as.matrix(lin.coef)

lin.part = pred2017%*%lin.coef

ts.part = simulate(armafit,nsim=212)

pred2017 = lin.part[,1] + ts.part


#Transform Predictions Back to Original Units
pred2017 = pred2017^(1/0.63771005)
TripTotal_2017 = TripTotal_2017^(1/0.63771005)

MSE = sum((pred2017-TripTotal_2017)^2)
MAE = mean(abs(pred2017-TripTotal_2017))
RMSE = sqrt(mean((pred2017-TripTotal_2017)^2))
MAPE = mean(abs((TripTotal_2017-pred2017)/(TripTotal_2017)))


plot(1:212,pred2017,xlab="Day of the Year",ylab="Median Daily Revenue",type="l",main="Median Daily Revenue 2017: Actual vs Predicted")
lines(1:212,TripTotal_2017,col="blue")
legend("topleft",c("Actual","Predicted"),lty=c(1,1),col=c(1,4))
```

G.3 Hourly

```
library(readr)
Hmedians_2013 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/OriginalData/Hmedians_2013.csv")
Hmedians_2014 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/OriginalData/Hmedians_2014.csv")
Hmedians_2015 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/OriginalData/Hmedians_2015.csv")
Hmedians_2016 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/OriginalData/Hmedians_2016.csv")

preds2013 = cbind(Hmedians_2013$hour,Hmedians_2013$`Trip Total`,Hmedians_2013$`Trip Miles`,Hmedians_2013$`Trip Seconds`,Hmedians_2013$Tolls, Hmedians_2013$Tips, Hmedians_2013$Extras)

preds2014 = cbind(Hmedians_2014$hour,Hmedians_2014$`Trip Total`,Hmedians_2014$`Trip Miles`,Hmedians_2014$`Trip Seconds`,Hmedians_2014$Tolls, Hmedians_2014$Tips, Hmedians_2014$Extras)

preds2015 = cbind(Hmedians_2015$hour,Hmedians_2015$`Trip Total`,Hmedians_2015$`Trip Miles`,Hmedians_2015$`Trip Seconds`,Hmedians_2015$Tolls, Hmedians_2015$Tips, Hmedians_2015$Extras)

preds2016 = cbind(Hmedians_2016$hour,Hmedians_2016$`Trip Total`,Hmedians_2016$`Trip Miles`,Hmedians_2016$`Trip Seconds`,Hmedians_2016$Tolls, Hmedians_2016$Tips, Hmedians_2016$Extras)


HourlyPreds = rbind(preds2013, preds2014, preds2015, preds2016)


HourlyPreds = as.data.frame(HourlyPreds)

colnames(HourlyPreds) = c("Hour","TripTotal","TripMiles","TripSeconds","Tolls","Tips","Extras")

write.csv(HourlyPreds, "HourlyPreds_untransformed.csv")
```

```
```{r external}
iHLWN = rep(0, 35060)
iHLWN[c(7297,16057,24815,33601)] = 1

iNYE = rep(0,35060)
iNYE[c(0,1,2,17519,17520,17521,26278,26279,26280)] = 1

iNYD = rep(0,35060)
iNYD[0:23] = 1
iNYD[8760:(23+8760)] = 1
iNYD[17519:(23+17519)] = 1
iNYD[26278:(23+26278)] = 1
```
```

```
iLYFT = c(rep(0,3120),rep(1,31940))

i2016 = rep(0, 35060)
i2016[26279:35060] = 1


hod2013 = c(rep(0:23,68),0,seq(2:23),rep(0:23,296))
hod2014 = c(rep(0:23,67),0,seq(2:23),rep(0:23,297))
hod2015 = c(rep(0:23,66),0,seq(2:23),rep(0:23,298))
hod2016 = c(rep(0:23,72),0,seq(2:23),rep(0:23,293))

iHOD = c(hod2013,hod2014,hod2015,hod2016) #Hour of Day

iHOY = c(rep(1:8759,3),1:8783)
iHOY2 = iHOY^2

iDLS = rep(0,35060)
iDLS[c(7346,16081,24816,33719)] = 1

iCUB = rep(0,35060)
iCUB[32426:32551] = 1

iRUSH = rep(0,35060)
iRUSH[which(iHOD == 6 | iHOD == 7 | iHOD == 8 | iHOD == 9 | iHOD == 10 | iHOD == 15 | iHOD ==
16 | iHOD == 17 | iHOD == 18 | iHOD == 19 )] = 1


iEVENING = rep(0,35060)
iEVENING[which(iHOD == 18 | iHOD == 19 | iHOD == 20 | iHOD == 21 | iHOD == 22 | iHOD == 23)] =
1


iLATE = rep(0,35060)
iLATE[which(iHOD == 0 | iHOD == 1 | iHOD == 2)] = 1


W2013 = c(rep(0,120),1,1,1,rep(0,21),rep(0,24))
W2014 = c(rep(0,96),1,1,1,rep(0,21),rep(0,48))
W2015 = c(rep(0,72),1,1,1,rep(0,21),rep(0,72))
W2016 = c(rep(0,48),1,1,1,rep(0,21),rep(0,96))



iSATSUN2013 = c(rep(W2013,9),rep(0,120),rep(0,23),rep(0,24),rep(W2013,42),rep(0,24))
iSATSUN2014 = c(rep(W2014,9),rep(0,96),rep(0,23),rep(0,48),rep(W2013,42),rep(0,24))
iSATSUN2015 = c(rep(W2015,9),rep(0,72),rep(0,23),rep(0,72),rep(W2013,42),rep(0,24))
iSATSUN2016 = c(rep(W2016,9),rep(0,48),rep(0,23),rep(0,96),rep(W2013,42),rep(0,48))

iSATSUN = c(iSATSUN2013,iSATSUN2014,iSATSUN2015,iSATSUN2016)
```

```
SM2013 = c(rep(0,144),rep(0,5),rep(1,3),rep(0,16))
SM2014 = c(rep(0,120),rep(0,5),rep(1,3),rep(0,16),rep(0,24))
SM2015 = c(rep(0,96),rep(0,5),rep(1,3),rep(0,16),rep(0,48))
SM2016 = c(rep(0,72),rep(0,5),rep(1,3),rep(0,16),rep(0,72))

iSM2013 = c(rep(SM2013,52),rep(0,24))
iSM2013 = iSM2013[-1633]
iSM2014 = c(rep(SM2014,52),rep(0,24))
iSM2014 = iSM2014[-1610]
iSM2015 = c(rep(SM2015,52),rep(0,24))
iSM2015 = iSM2015[-1586]
iSM2016 = c(rep(SM2016,52),rep(0,48))
iSM2016 = iSM2016[-1730]

iSM = c(iSM2013,iSM2014,iSM2015,iSM2016)

```


```{r combine}
allData = cbind(HourlyPreds[,-5],iHLWN,iNYE,iLYFT, i2016,iHOD,iHOY,iHOY2, iDLS, iCUB, iRUSH,
iEVENING, iLATE, iNYD,iSATSUN,iSM)

```

```{r transformations}
#-------Transformed TripMiles and TripSeconds----------------
#------Replace Extras with Indicator for Nonzero Extras-----

varlist = c("Hour","TripTOtal","TripMiles","TripSeconds","Tips","Extras","iHLWN","iNYE","iLYFT"
,"i2016","iHOD","iHOY","iHOY2","iDLS","iCUB")
for(i in 1:9){
  hist(allData[,i],main=varlist[i])
}
#Transform Columns 3,4,5
library(forecast)
lambdas = rep(0,3)
for(i in 3:5){
   lambdas[(i-2)] = BoxCox.lambda(allData[,i])
}
lambdas[1] = 0.5714286
for(i in 3:4){
  hist(allData[,i]^lambdas[(i-2)],main=varlist[i])
}

for(i in 3:4){
  allData[,i] = allData[,i]^lambdas[(i-2)]
}

iExtra = rep(0,35060)
iExtra[which(allData$Extras>0)] = 1
```

```
allData[,6] = iExtra

time = 1:35060

allData = cbind(allData,time)

colnames(allData) = c("Hour","TripTOtal","TripMiles","TripSeconds","Tips","iExtra","iHLWN","iNYE","iLYFT","i2016","iHOD","iHOY","iHOY2","iDLS","iCUB","iRUSH","iEVENING","iLATE","iNYD","iSATSUN","iSM","time")



```

```{r fullModel}
fitfull = lm(TripTOtal~ TripMiles + TripSeconds + Tips + iExtra + iHLWN + iNYE + iLYFT + i2016 + iHOD +iHOY+ iHOY2 +iDLS +iCUB +iRUSH + iEVENING + iLATE + iSATSUN + iNYD +iSM + time +time*lyft, data=allData)

summary(fitfull)
library(DAAG)
vif(fitfull)
```

```{r variableSelection}
library(leaps)
leaps1 = regsubsets(y = allData$TripTOtal,x = allData[,-2], method="exhaustive",nbest=3)

plot(leaps1,scale="bic")


step(fitfull,direction="backward")
```

```{r reduced}
#------------------------------------------------------------------------
red1 = lm(TripTOtal~ TripMiles  + TripSeconds+ iHOD + iExtra + iHLWN  + iLYFT +iHOY+ iHOY2 +iDLS +iCUB  +iEVENING +iLATE +iSATSUN +iSM +iRUSH + time +time*iLYFT +i2016, data=allData)
#------------------------------------------------------------------------

xpred = cbind(allData$TripMiles, allData$TripSeconds, allData$iHOD, allData$iExtra, allData$iHLWN, allData$iLYFT, allData$iHOY, allData$iHOY2, allData$iDLS, allData$iCUB, allData$iEVENING, allData$iLATE, allData$iSATSUN, allData$iSM, allData$iRUSH, allData$time, allData$time*allData$iLYFT, allData$i2016)

#red2 = lm(TripTOtal ~ Hour + I(Hour^2)+ TripMiles + TripSeconds + iExtra #+ i2016 + iHOY + iHOY2 + iDLS + iSATSUN +iSM, data = allData)

#anova(red1,red2)
```

```
summary(red1)
#summary(red2)

#plot(red2$residuals,type="l")
#resi2 = red2$residuals
#resi2 = ts(resi, frequency = 24)
#acf(resi2)
#pacf(resi2)


plot(red1$residuals,type="l")
resi1 = red1$residuals
resi1 = ts(resi1, frequency = 24)
par(mfrow=c(2,1))
acf(resi1)
pacf(resi1)
par(mfrow=c(1,1))

confint(red1)

#Save Linear Coefficients
lin.coef = red1$coefficients
```

```{r timeseries}
armafit = arima(resi1, order = c(4,0,2), seasonal =c(3,0,0))
ts.coef = armafit$coef
```

```{r predict}
HpredData2017 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/HpredData2017.csv")
Hmedians_2017 <- read_csv("//sf1/Users/grad/metheney/Desktop/ChicagoTaxi/Hourly/OriginalData/Hmedians_2017.csv")

iExtra = HpredData2017$iExtra2017
iExtra[which(is.na(HpredData2017$iExtra2017)==TRUE)] = 0

seconds2017 = Hmedians_2017$`Trip Seconds`^(0.3999674)


HpredData2017[,5] = iExtra
HpredData2017 = HpredData2017[,-1]
HpredData = HpredData2017[,-(16:19)]


miles2017 = HpredData$Miles2017^(0.5714286)

time2017 = seq(35061,by = 1, length.out = 5087)
timeLYFT2017 = time2017*HpredData$iLYFT2017
```

```
iHOD2017 = HpredData$iHOD2017

iRUSH2017 = rep(0,5087)
iRUSH2017[which(iHOD2017 == 6 | iHOD2017 == 7 | iHOD2017 == 8 | iHOD2017 == 9 | iHOD2017 == 10
| iHOD2017 == 15 | iHOD2017 == 16 | iHOD2017 == 17 | iHOD2017 == 18 | iHOD2017 == 19 )] = 1

HpredData = cbind(HpredData[,1],miles2017,seconds2017,HpredData[,3:6],HpredData[,8:15],iRUSH201
7,time2017,HpredData$i20162017,timeLYFT2017)

y2017 = Hmedians_2017$`Trip Total`



HpredData = as.matrix(HpredData)

lin.coef = as.matrix(lin.coef)

lin.part = HpredData%*%lin.coef

set.seed(1234)
ts.part = simulate(armafit,n=5087)

pred2017 = lin.part[,1] + ts.part

x1 = 1:2000
x2 = 2002:4000
x3 = 4001:5087

par(mfrow=c(3,1))
plot(x1, y2017[x1],type="l", main = "Hourly Median Revenue: Actual vs Predicted", xlab = "Hour
of Year", ylab="Hourly Median Revenue")
lines(x1, pred2017[x1],col="blue")
legend("topleft",c("Actual","Predicted"),lty=c(1,1),col=c(1,4))

plot(x2, y2017[x2],type="l", main = "Hourly Median Revenue: Actual vs Predicted", xlab = "Hour
of Year", ylab="Hourly Median Revenue")
lines(x2, pred2017[x2],col="blue")
legend("topleft",c("Actual","Predicted"),lty=c(1,1),col=c(1,4))

plot(x3, y2017[x3],type="l", main = "Hourly Median Revenue: Actual vs Predicted", xlab = "Hour
of Year", ylab="Hourly Median Revenue")
lines(x3, pred2017[x3],col="blue")
legend("topleft",c("Actual","Predicted"),lty=c(1,1),col=c(1,4))
par(mfrow=c(1,1))


MSE = sum((pred2017-y2017)^2) #95912.89
MAE = mean(abs(pred2017-y2017)) #3.3927
RMSE = sqrt(mean((pred2017-y2017)^2)) #4.34
MAPE = mean(abs((y2017-pred2017)/(y2017))) #0.1467
```