Forecasting Uber Trips in NYC Using a SARIMA Model

Cole M. Morgan

4/18/2019

Background and Key Questions

Uber is a global ride sharing service that handles tens of thousands to hundreds of thousands of trips every day just a single city. For Uber forecasting the number of pickups and drop offs is extremely important because it helps them plan and prepare for variable demand of cars and drivers which if done accurately can yield them large economic benefits. I will look at the number of Uber rides per day in New York City from a five month period (January 2015 to June 2015). In the process of forecasting this time series I will investigate whether Uber trips by day have an underlying signal that contains components such as seasonality and/or trend. After exploring the data I will fit SARIMA based models on a training set of data. The evaluation metric I will compare the models by will be their respective Akaike information criterion scores. Next I will test the best model's ability to predict the number of trips in the future by comparing a one month out forecast with one month of hold out test data. I will evaluate the error using the mean absolute error calculation.

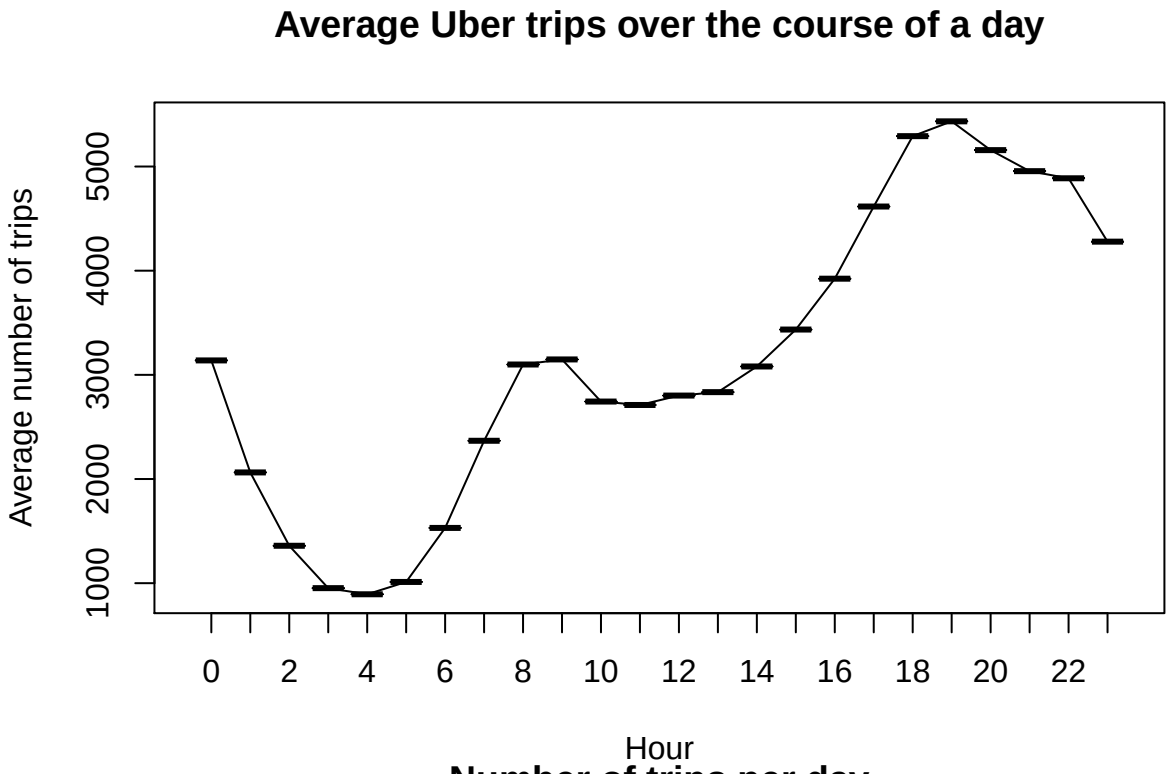
The two key questions I will investigate are:

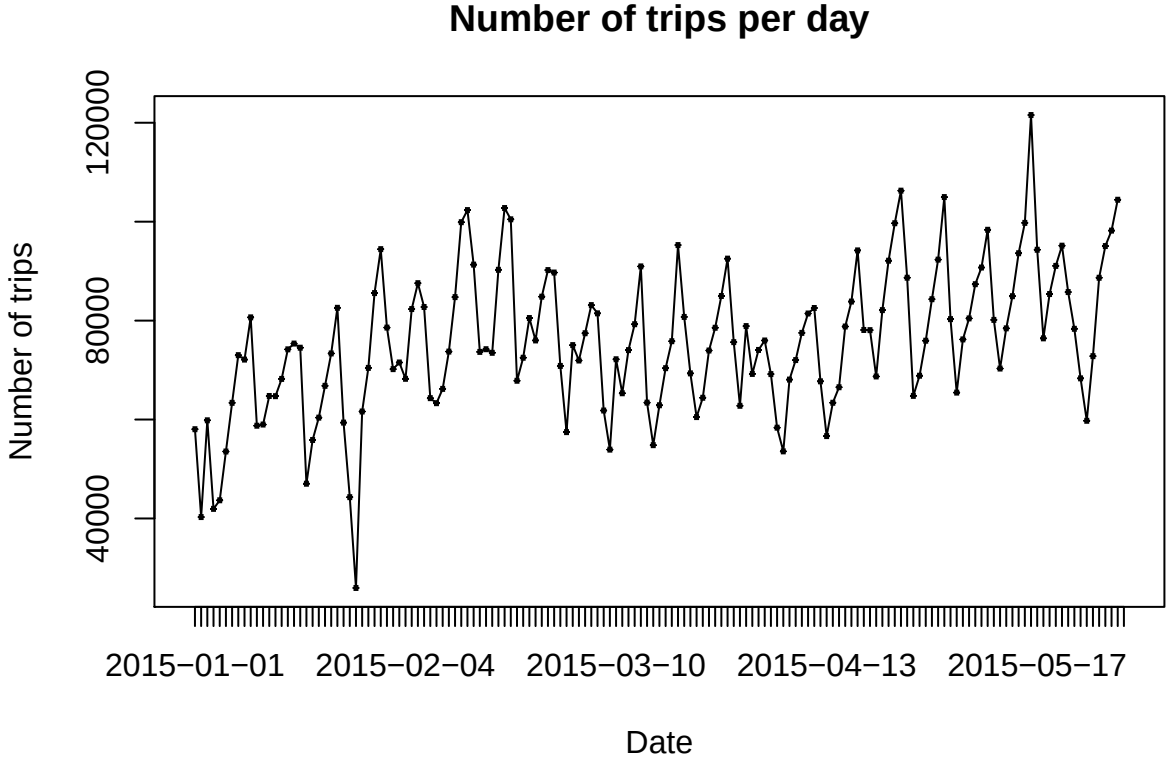
- Is it possible to accurately predict the number of Uber trips per day using a SARIMA based model?
- What is the accuracy of the forecasts made by the SARIMA model?

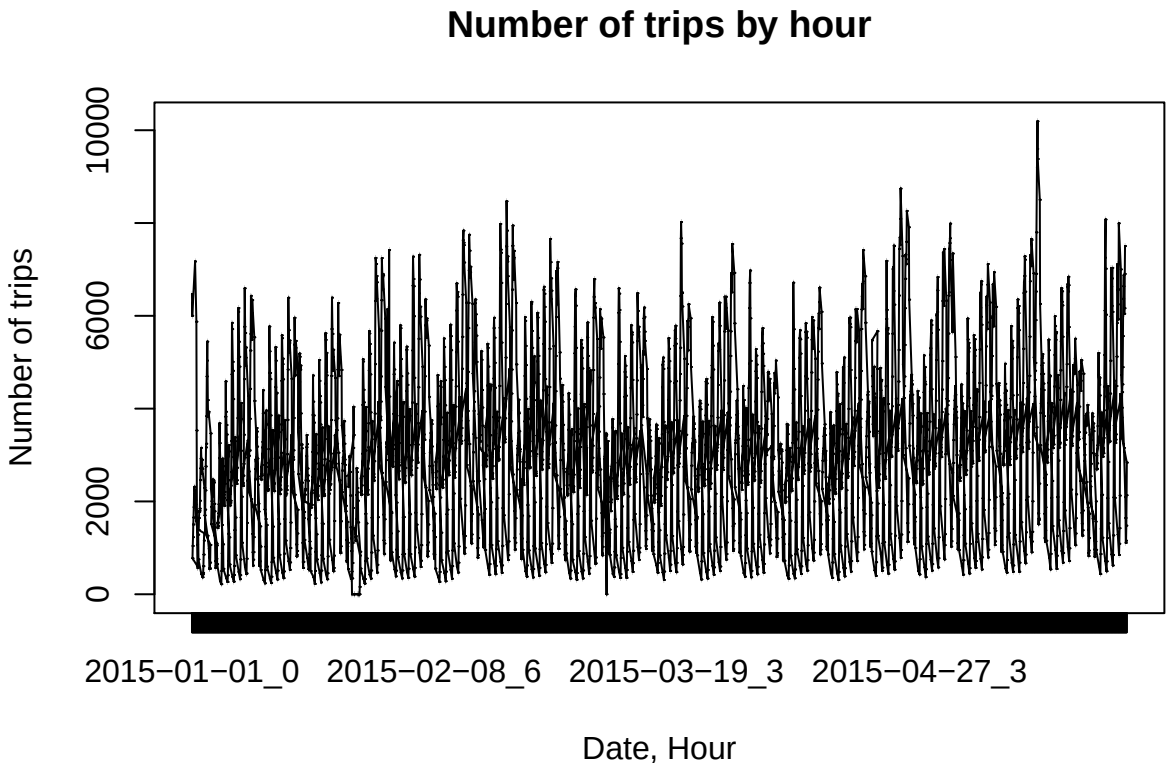
To answer these questions I will first visual examine the time series at different level of aggregation to get a sense of the data. After that I will try to remove any trend that I find and then investigate for seasonality, AR, or MA components. After both of these first two steps I will then model it using an appropriately tuned arima or sarima model using my findings from the previous step. After fitting the model on the train data I will evaluate its true performance by predicting on a hold out set of test data.

Exploritory Data Analysis

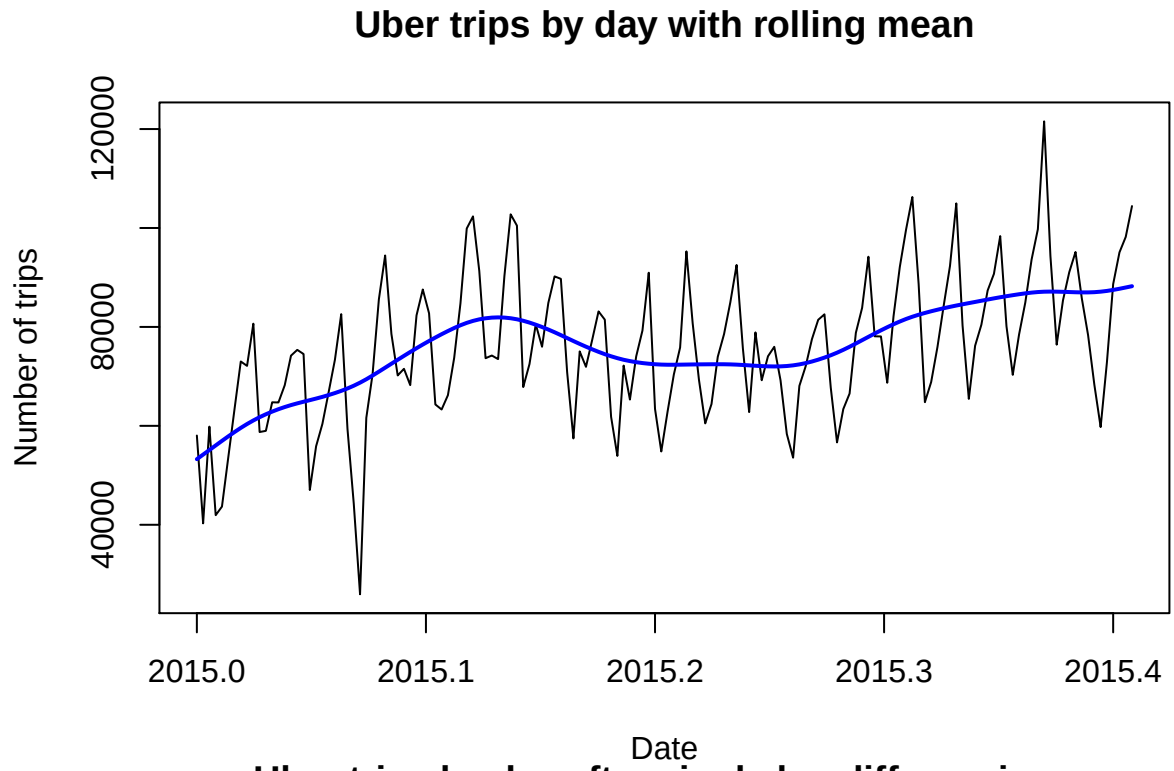
The data contains individual Uber trips including their date and time. The training data is from New York City from January 2105 through May 2015, and the test data is just June 2015. I will be primarily focused on forecasting the trip count at an aggregate by day level. I received the data from Professor Devika Submaranian, she created this data set for a graduate research program.

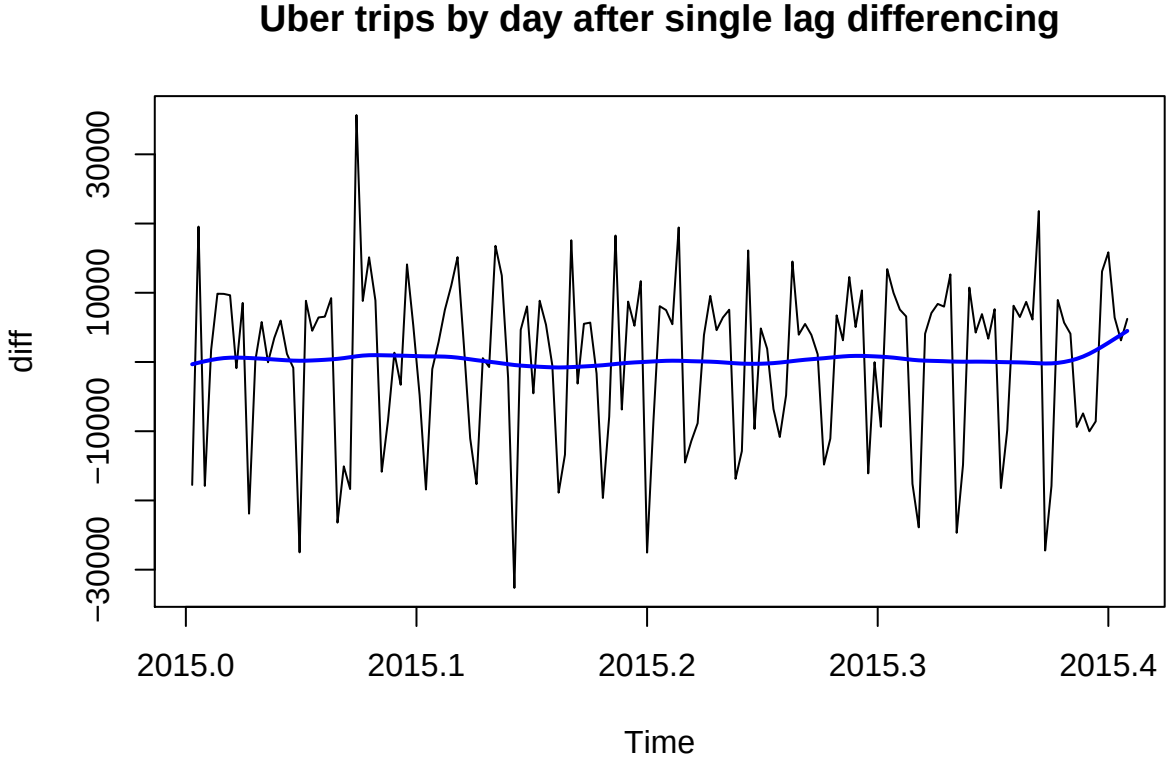

Average Uber trips over the course of a day
Average number of trips
1000
2000
3000
4000
5000
0
2
4
6
8
10
12
14
16
18
20
22
Hour


Number of trips per day
Number of trips
40000
80000
120000
2015−01−01
2015−02−04
2015−03−10
2015−04−13
2015−05−17
Date

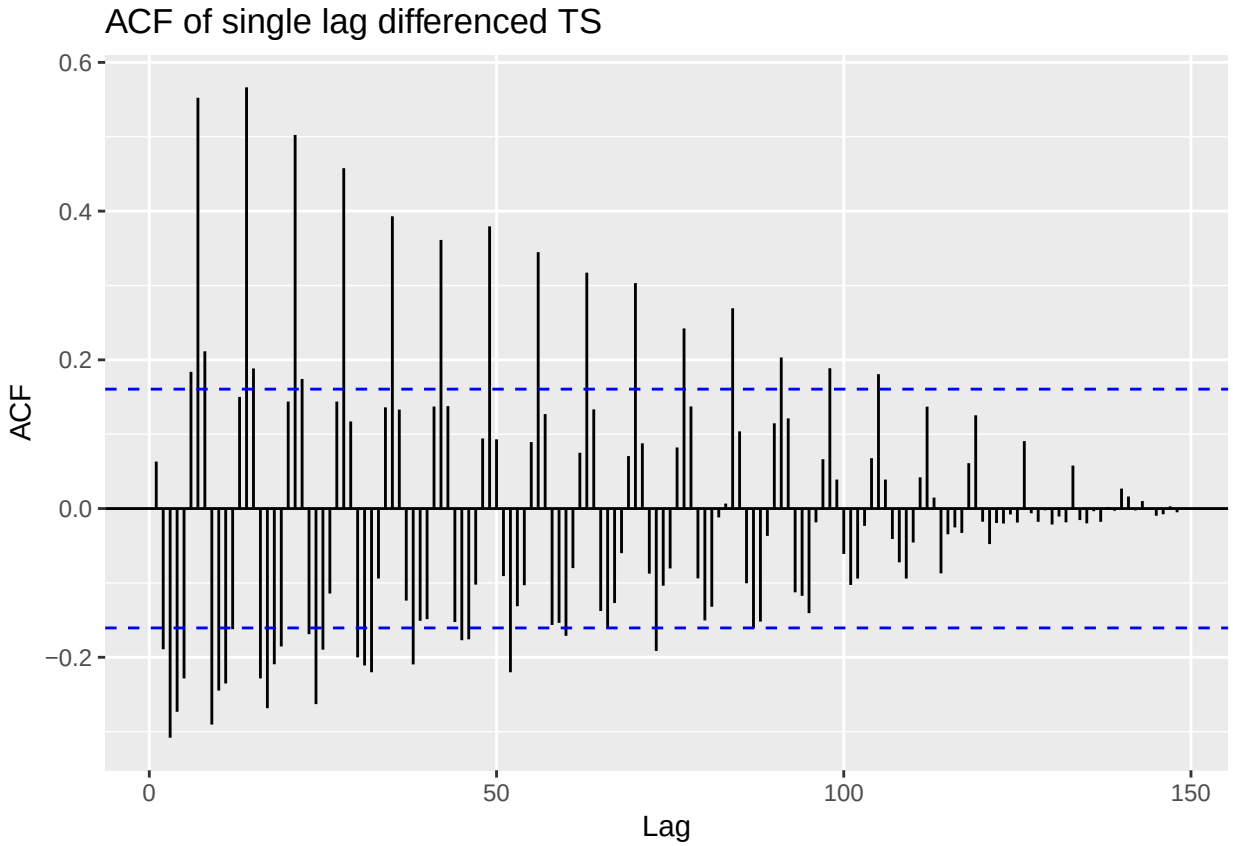


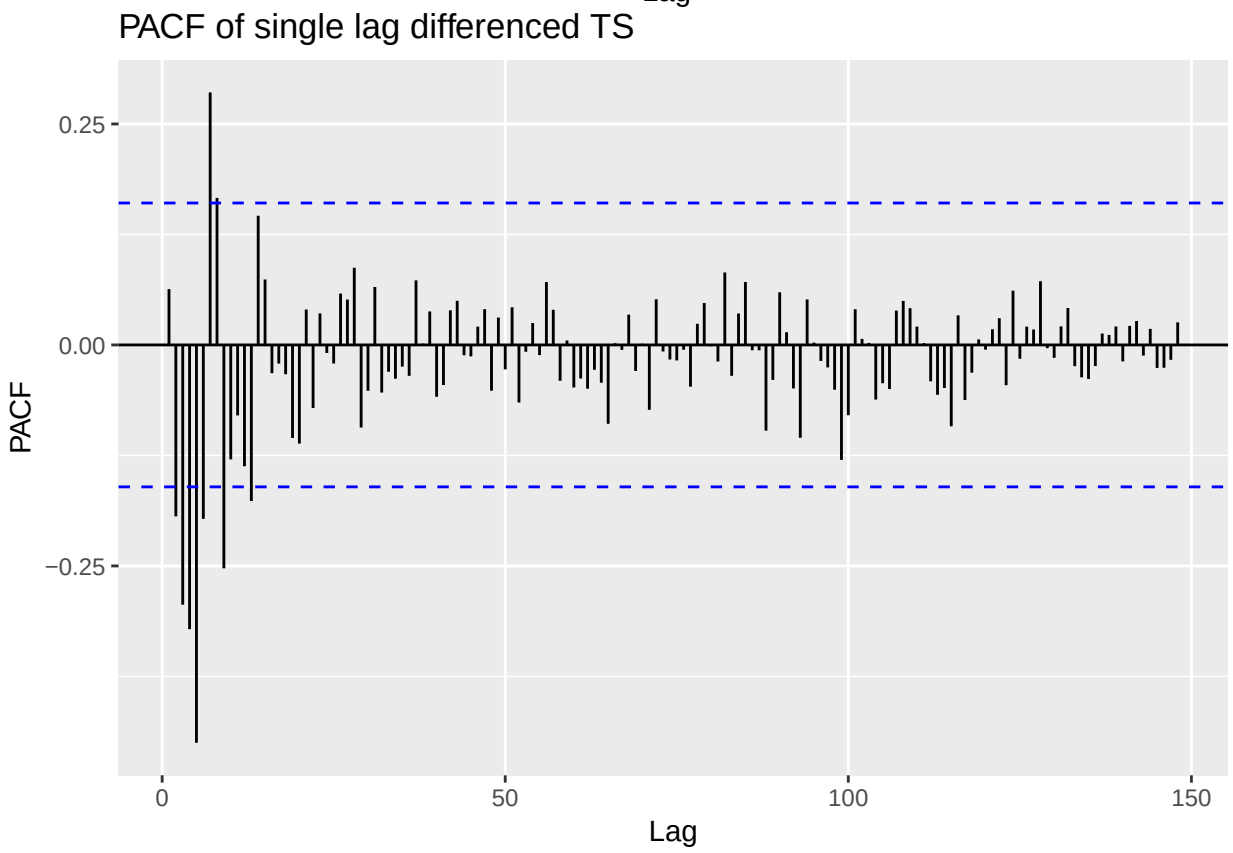
Plotting the average number of trips per hour for one day shows us that there is a 24 hour cyclic pattern, but since I am aggregating to the by day level this will not matter. This same pattern is observed repeating itself every 24 hours in the number of trips by hour plot. The Number of trips per day plot shows us that there is a general upward trend to the data and that there appears to be a 7 day or weekly cycle in the data. To investigate further I will first look at whether the time series of by day truly has a trend. I then will look at the ACF and PACF of the detrended time series.





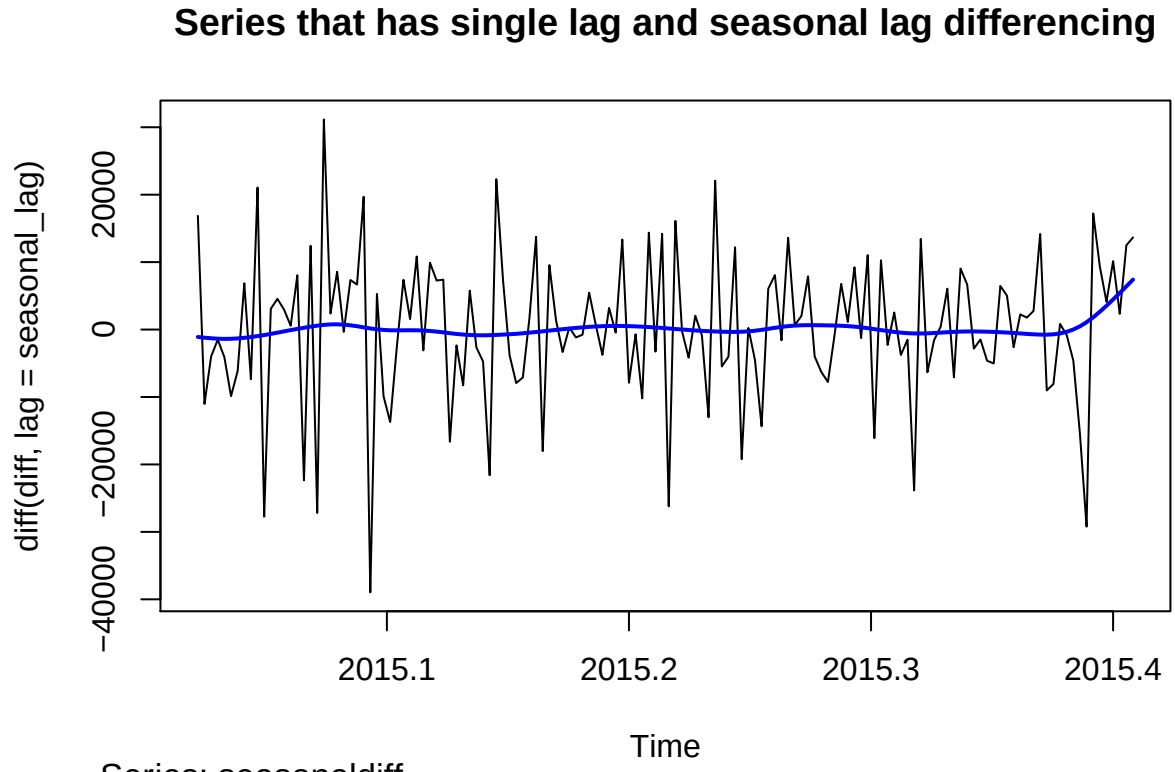
To first determine if the series contained a general trend I plotted the data along with its spline smoothed mean. From this plot I saw that the series had general upward drift, meaning it is not trend stationary. To handle this I decided to difference the series by a single lag. The single lag differencing resulted in a plot that appears to be trend stationary. The next logical step is to investigate the ACF and PACF of the single lag differenced time series.

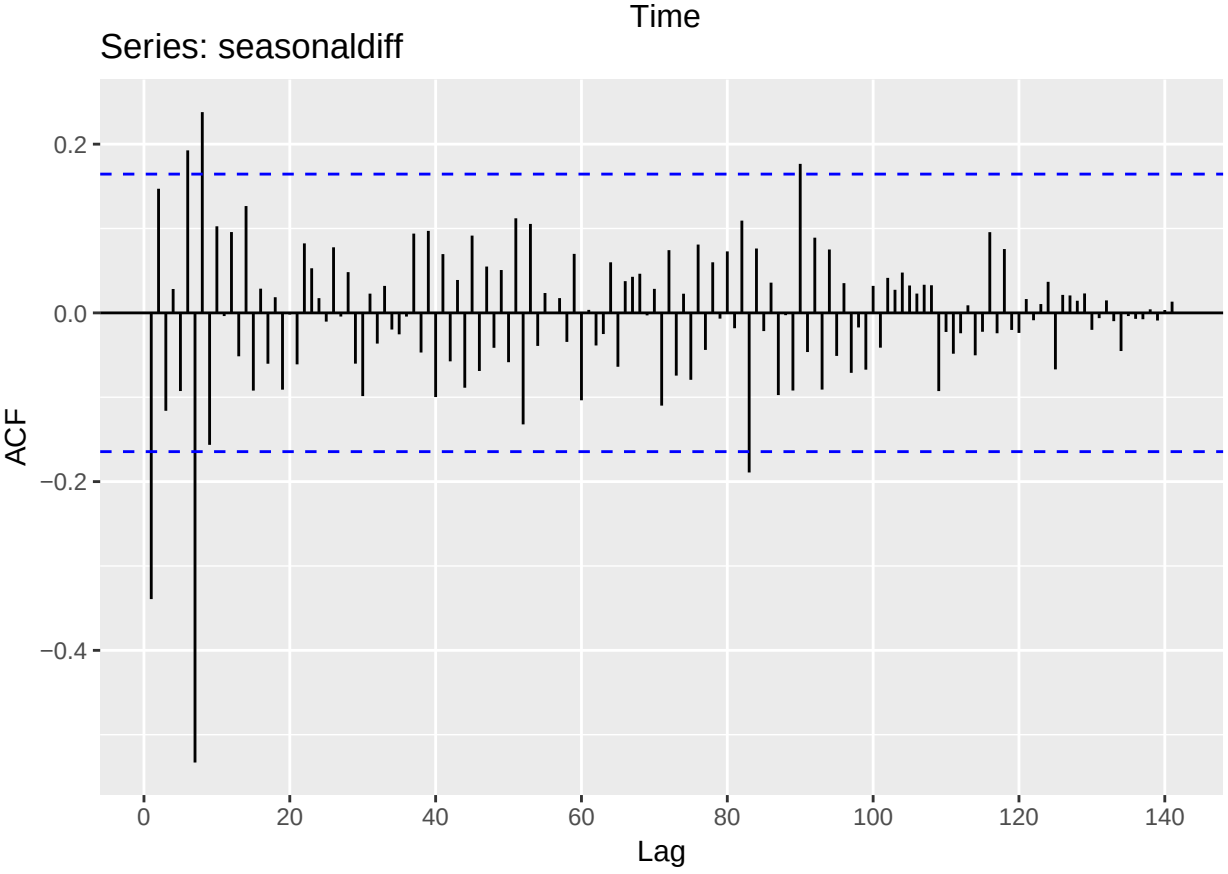


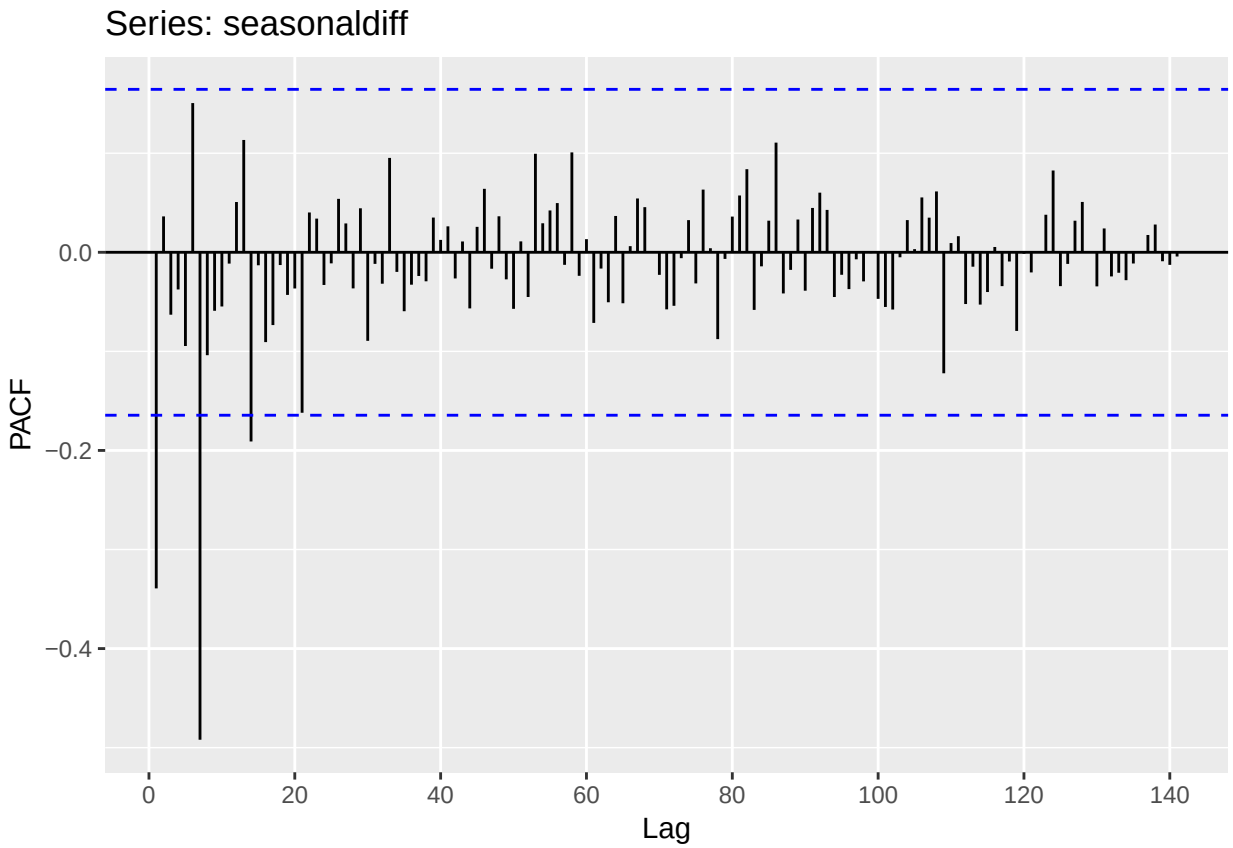


Looking at the ACF and PACF I attempted to determine the AR and MA order of the single lag differenced

time series. From the ACF I saw that there was a strong seasonal component still in the data which had a cycle of 7 lags (one week). Since the ACF and PACF are hard to interpret for the AR and MA order while this seasonal component is present I decided to do another difference of 7 lags.







Differencing by 7 lags yielded me a time series that was still trend stationary. I once again plotted the ACF and PACF of the twice differenced series. The ACF shows a sharp cut off after lag 7. It also has two lags that are significant: lag 1 and lag 7. The PACF shows a exponentially decreasing in magnitude pattern after lag 7 and it also only has two lags that are significant lag 1 and 7 again. From this evaluation the most likely model must have differencing for the seasonal and non seasonal component and be an MA order 1 for both the seasonal and non seasonal. This logic yields: $ARIMA(0,1,1) \times ARIMA(0,1,1)[7]$.

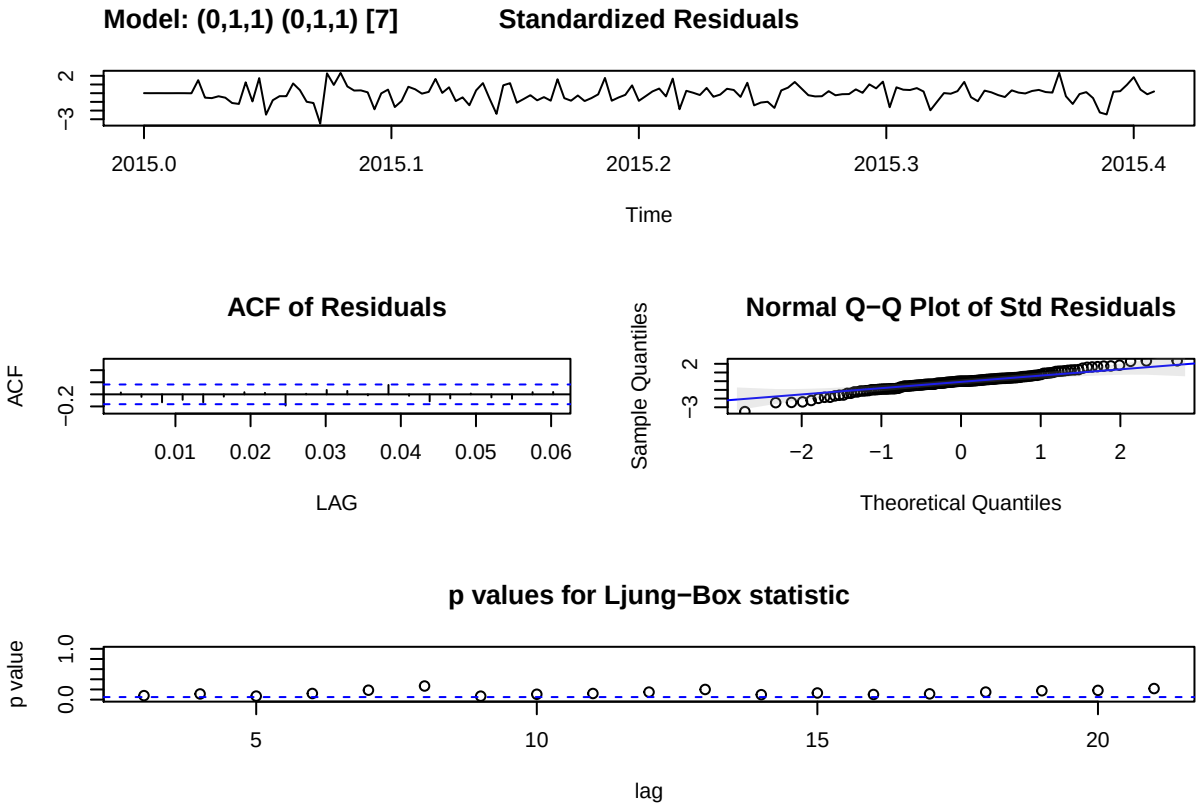
Modeling Dailly Uber Trips

I begin the modeling process with the starting model arrived to in the previous section:$ARIMA(0,1,1) \times ARIMA(0,1,1)[7]$.

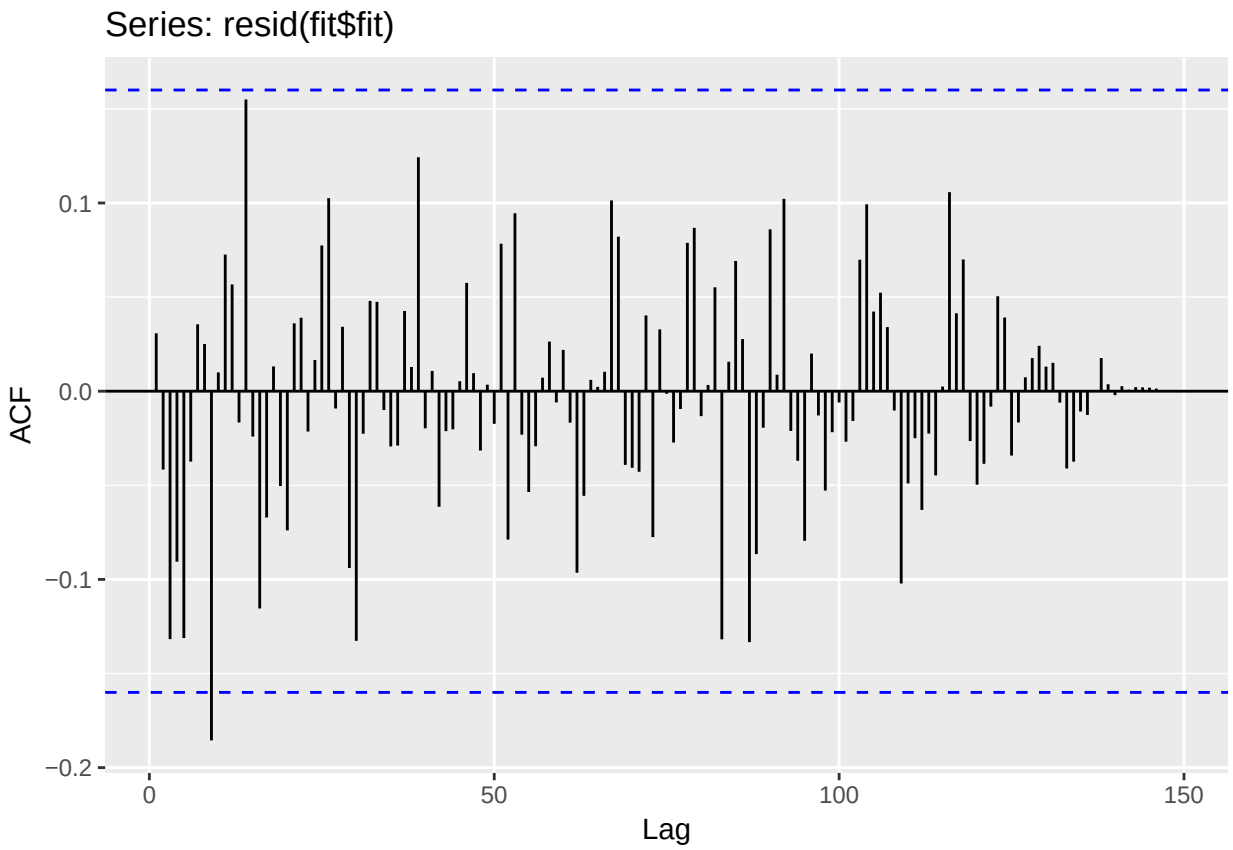
```
fit <- astsa::sarima(byday, 0, 1, 1, P = 0, D = 1, Q = 1, S = 7,
       details = TRUE, xreg=NULL, Model=TRUE,
       tol = sqrt(.Machine$double.eps),
       no.constant = FALSE)
```

```
## initial  value 9.305182
## iter   2 value 9.059140
## iter   3 value 9.057091
## iter   4 value 9.048166
## iter   5 value 9.047230
## iter   6 value 9.047143
## iter   7 value 9.047143
## iter   8 value 9.047142
## iter   8 value 9.047142
## iter   8 value 9.047142
```

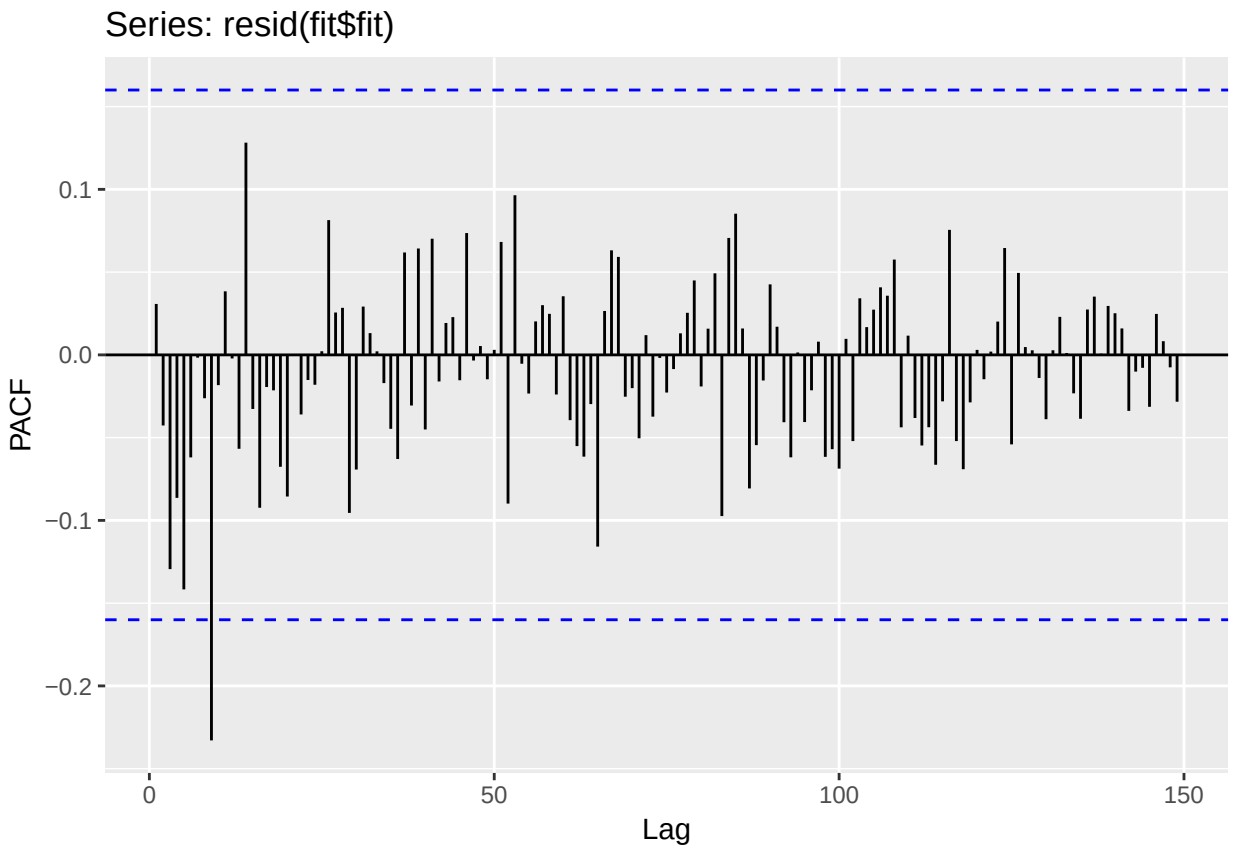
```
## final  value 9.047142
## converged
## initial  value 9.026012
## iter   2 value 9.003687
## iter   3 value 8.996437
## iter   4 value 8.993669
## iter   5 value 8.993645
## iter   6 value 8.993643
## iter   6 value 8.993643
## final  value 8.993643
## converged
```




```
ggAcf(resid(fit$fit))
```




```
ggPacf(resid(fit$fit))
```




```
fit$AIC
```

```
## [1] 18.86218
```

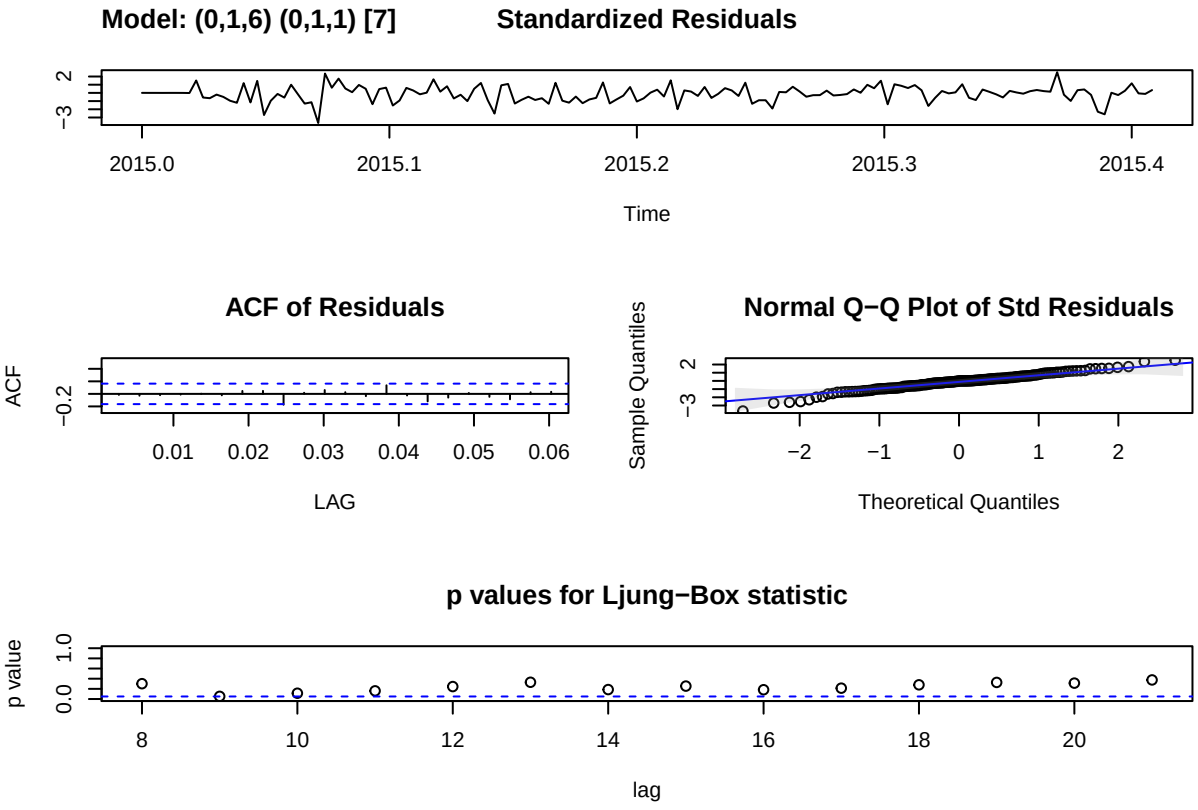
The model seems to be fitting quite well since the AIC is relatively low. The ACF and PACF of the residuals do not have any worrying significant features. But I still think that the models parameter tuning could be better. To find a better fit I swept through similar models and compare them using AIC.

From my parameter sweep I found this $ARIMA(0,1,6) \times ARIMA(0,1,1)[7]$ to be the best in terms of AIC.

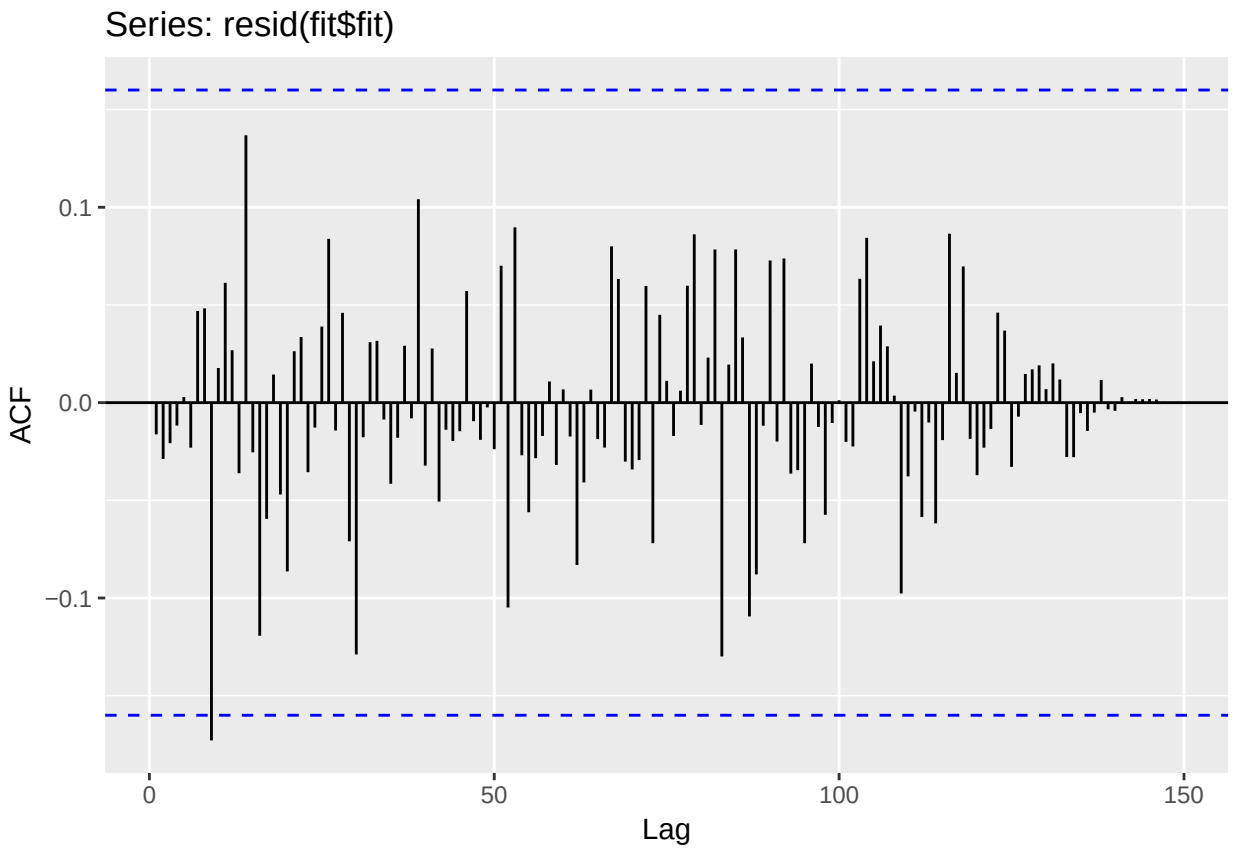
```
fit <- astsa::sarima(byday, 0, 1, 6, P = 0, D = 1, Q = 1, S = 7,
       details = TRUE, xreg=NULL, Model=TRUE,
       tol = sqrt(.Machine$double.eps),
       no.constant = FALSE)
```

```
## initial  value 9.305182
## iter   2 value 9.084310
## iter   3 value 9.051318
## iter   4 value 9.030851
## iter   5 value 9.024794
## iter   6 value 9.018711
## iter   7 value 9.015624
## iter   8 value 9.015378
## iter   9 value 9.015361
## iter  10 value 9.015359
## iter  11 value 9.015359
## iter  11 value 9.015359
## iter  11 value 9.015359
## final  value 9.015359
```

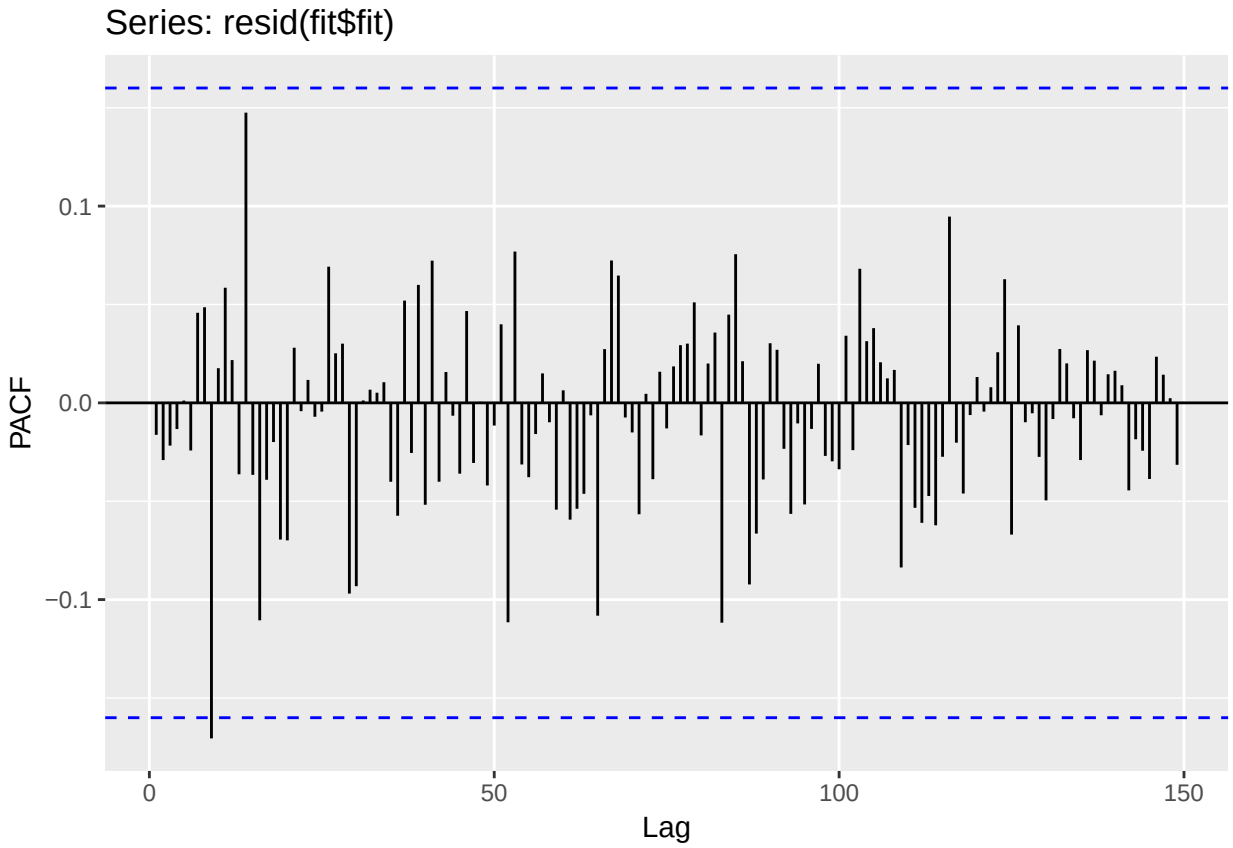
```
## converged
## initial  value 8.995238
## iter   2 value 8.982354
## iter   3 value 8.981127
## iter   4 value 8.970635
## iter   5 value 8.967446
## iter   6 value 8.963457
## iter   7 value 8.963159
## iter   8 value 8.962864
## iter   9 value 8.962781
## iter  10 value 8.962756
## iter  11 value 8.962753
## iter  11 value 8.962753
## iter  11 value 8.962753
## final  value 8.962753
## converged
```




```
ggAcf(resid(fit$fit))
```




```
ggPacf(resid(fit$fit))
```




```
fit
```

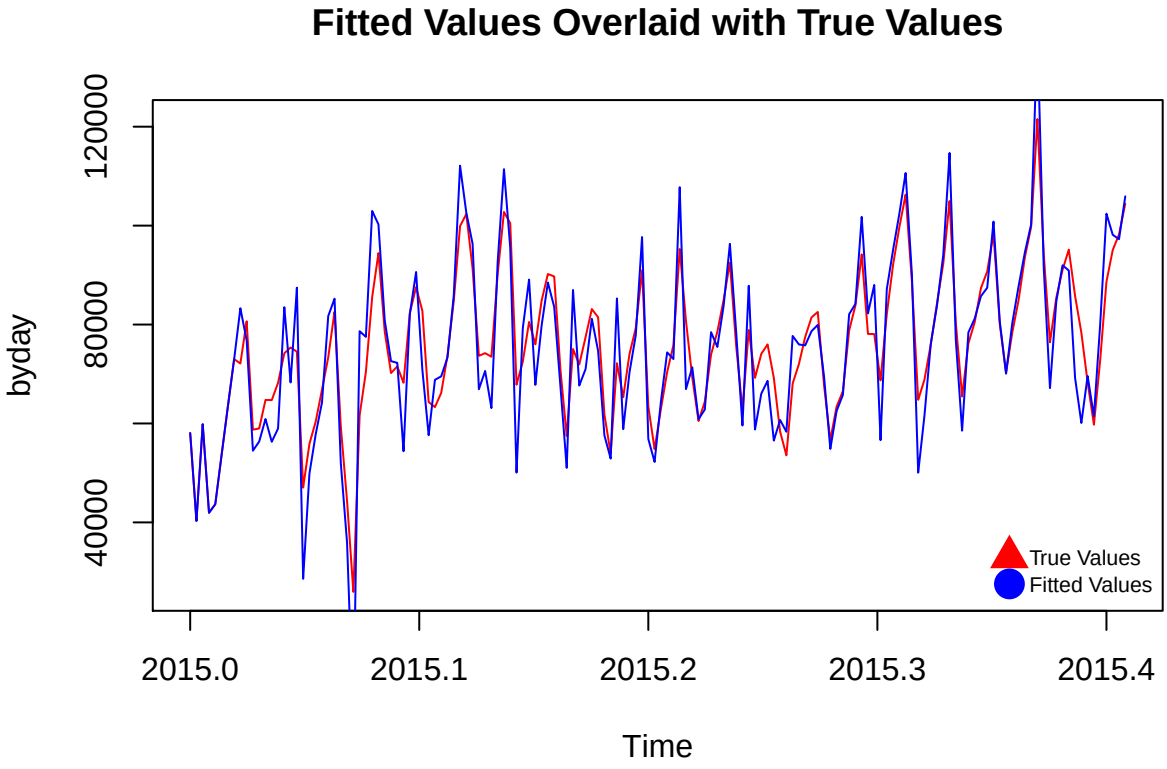
```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1      ma2      ma3      ma4      ma5     ma6     sma1
##       -0.3866  -0.0685  -0.1288  -0.0747  -0.1263  0.0369  -1.0000
## s.e.   0.0850   0.0956   0.0957   0.0967   0.0872  0.0981   0.1033
##
## sigma^2 estimated as 51881537:  log likelihood = -1474.2,  aic = 2964.4
##
## $degrees_of_freedom
## [1] 135
##
## $ttable
##      Estimate     SE t.value p.value
## ma1   -0.3866 0.0850 -4.5479  0.0000
## ma2   -0.0685 0.0956 -0.7159  0.4753
## ma3   -0.1288 0.0957 -1.3458  0.1806
## ma4   -0.0747 0.0967 -0.7730  0.4409
## ma5   -0.1263 0.0872 -1.4497  0.1495
## ma6    0.0369 0.0981  0.3759  0.7076
```

```
## sma1  -1.0000 0.1033 -9.6813  0.0000
##
## $AIC
## [1] 18.85781
##
## $AICc
## [1] 18.87795
##
## $BIC
## [1] 17.9983
```

```
fit$AIC
```

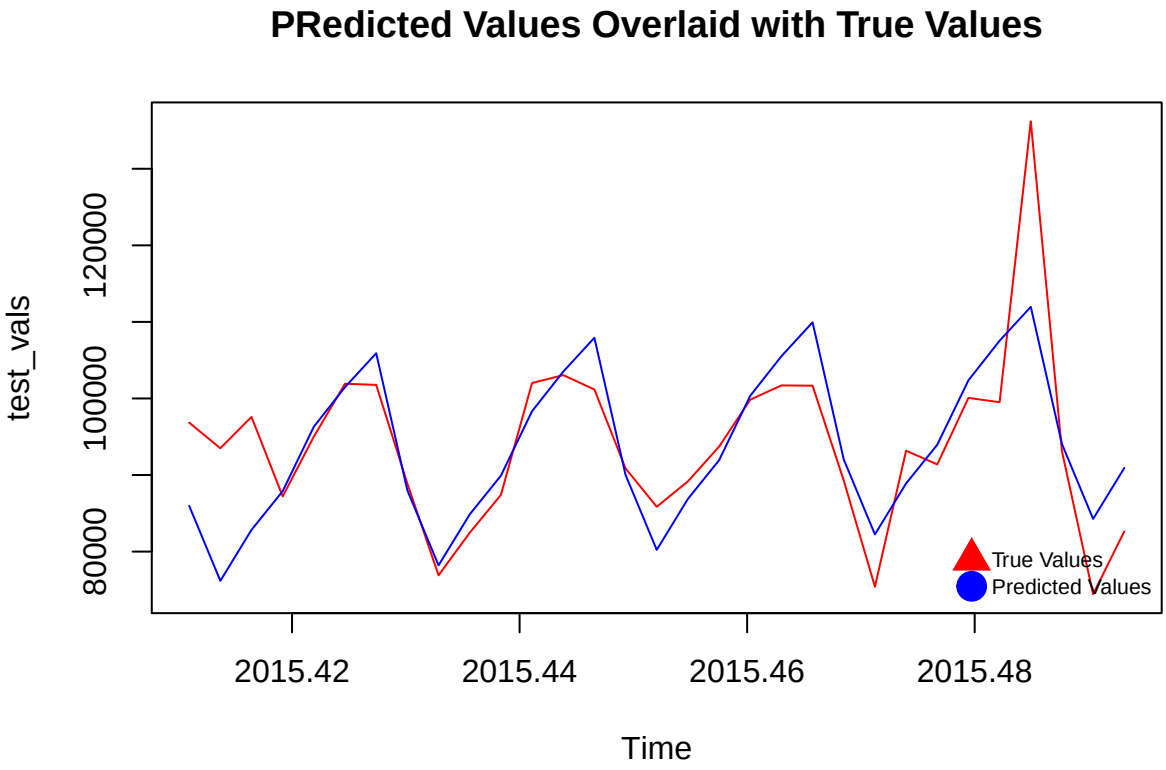
```
## [1] 18.85781
```

The model has the lowest AIC of 18.85781 and has no ACF or PACF values that are significant. However, upon looking at the pvalues in the ttable for the added MA coefficients non of them are significant meaning the original simpler model was actually the best. The simpler model was this $ARIMA(0,1,1)\times ARIMA(0,1,1)[7]$ model. Next I look at the fit of $ARIMA(0,1,1)\times ARIMA(0,1,1)[7]$ against the true data that it trained on to get a visual sense of how well the model is actually fitting.



Forecasting and Testing

Using the best model in terms of AIC trained on 4 months of data I forecasted one month out to compare on one month of hold out data. The plot of the true values one month out and the predicted values is below




```
## [1] 5182.988
```

The $ARIMA(0,1,1) \times ARIMA(0,1,1)[7]$ model seems to be generalizing quite well even while trained on such a small sample compared to the forecast window. The mean absolute error was 5455 over the one month forecast period. For values that are in the range of roughly 80 thousand to 120 thousand being off on average by 5 thousand is pretty good.

Conclusions and Future Questions

Using SARIMA based models can in fact help us fit and predict dialy Uber trip numbers in New York City. The model performed will on the train and test set and was able to predict out over quite a large horizon compared to its training set. The Uber data was quite interesting since it required regular and seasonal differencing in order to obtain its true AR and MA order. Hopefully Uber uses similar forecasting methods to help predict and then plan for future variable demand.

My next step will be to try to model the hourly series of this Uber data. Although I have already tried once the high lag order of the weekly seasonal component, 168, means that R's optimization function fails while training on this data. The hourly Uber data is interesting to me since it has two concurrent seasonal effects the weekly seasonal effect of period 168 and a daily seasonal effect of period 24. If I have the time my next inquiry will be into modeling this granularity of the data.