Citi Bike Rides

Isaac Rene Mollinedo Portilla

November 14, 2018

Abstract

The objective of this paper was to identify a time series model representative of the number of weekly Citi Bike rides in New York City – and then using such model to produce forecasts. The data analyzed had weekly frequency, and it spanned from the beginning of September 2013 to the end of October 2018 (n = 277 weeks). The analysis lead to six potential Multiplicative Seasonal ARIMA Models that could be used for the stated objective. The six models' diagnostics were evaluated; as a result of the evaluation, out of the six, an ARIMA$(1,1,1)\times(0,1,1)_{51}$ model and an ARIMA$(0,1,1)\times(1,1,1)_{51}$ model were selected. These two models were then compared using forecast cross validation testing. At the end, taking into consideration both the diagnostics and the sum of square errors obtained from the cross validation tests, the chosen model to represent the weekly Citi Bike rides was the ARIMA$(0,1,1) \times (1,1,1)_{51}$. The project concludes by using the latter model to forecast the next 26 weeks of Citi Bike rides.

1. Introduction

1.1. Motivation in the Selection of the Data

The selection of the Citi Bike rides data was motivated by the interesting challenges that utilizing this data entailed – starting with the challenges involved in the acquisition and preparation of the data, and continued in the analysis of the time series with the presence of seasonality.

The data was first considered for the subject of this project out of the curiosity of finding out what the series itself would look like, for it wasn't aggregately readily available. Furthermore, there was also curiosity as to whether this data could serve as part of a bigger research project; one in which this data could be used as a sort of proxy (a limited proxy) for the trends in the overall bicycle usage in the city of New York; something which would be interesting to compare with the government's spending on bicycle infrastructure throughout the city (if this data is also available).

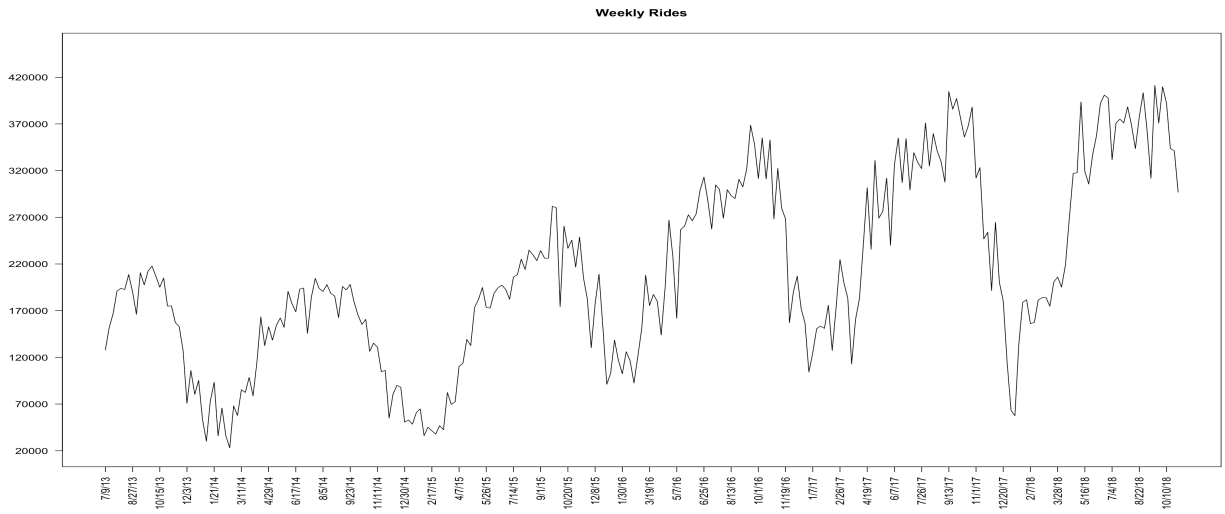

Figure 2.1. Values of the Weekly Citi Bike rides (September 2013 – October 2018, n = 277 weeks)

1.2. Data Sourcing, Scrapping, and Preparation

1.2.1. Raw Data

The data was obtained directly from the Citi Bike NYC website. The original data was in the form of csv files, with one file for each month since September 2013. Each file contained a row for each individual ride, and had been corrected to not include rides that were under 60 seconds.

1.2.2. Processing of the Data

Among all the rides, only those with durations greater than 90 seconds were kept. After this filtering, the remaining rides where grouped by day – and for each day it's respective number of rides was obtained. Finally, the data was aggregated by week; for this, every seven days, was considered a week, and in order to get only whole weeks, the first two days of the time series were eliminated. (Python Pandas was used for the processing)

1.2.3. Issues in the data

The following days were missing from the data:

1/23/16	1/24/16	1/25/16	1/26/16
2/9/17	3/14/17	3/15/17	3/16/17

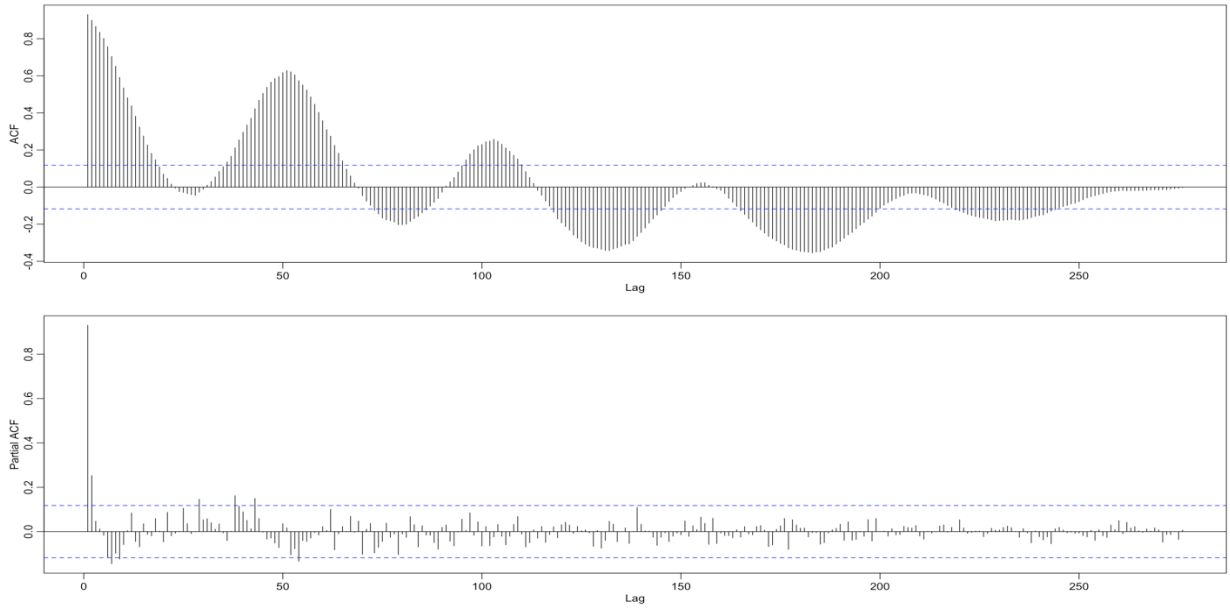

Figure 2.2. ACF and PACF of the Citi Bike series

2. Exploratory Analysis

2.1.1. Data Visualization and Preparation Requirements

We begin our time series analysis by plotting the data as show in Figure 2.1. From this plot we note a few things of interest:

i. The pattern is representative of seasonal data, which suggest we will have to use a Multiplicative Seasonal ARIMA Model.
ii. The data has a trend, which is indicative of nonstationary behavior.
iii. The variance does not appear to be constant, suggesting that it must be transformed.

Further evidence for point i and ii is obtained by graphing the ACF and PACF shown in Figure 2.2.

To advance the analysis, the issue of variance needed to be addressed first; for it the Box-Cox procedure was used. For the Box-Cox lambda we obtained a value of 0.4869, which indicates a square root transformation of the data. After the transformation we found that the behavior of the variance along the time series was more constant. Having fixed the issue of non-constant variance, to address the nonstationary behavior of the series, the first difference was taken.

The ACF and PACF of the transformed-differenced data are shown in Figure 2.3. From these graphs, the seasonality is made clear by the peaks at lags 51 and 102 (Although in a yearly data set we would usually expect the lag to be at 52 weeks, in this case 51 weeks appears to be more fitting, something that might be due to the missing days mentioned in section 1.2.3). This seasonality further suggests we take a seasonal difference with *s* equaling 51.

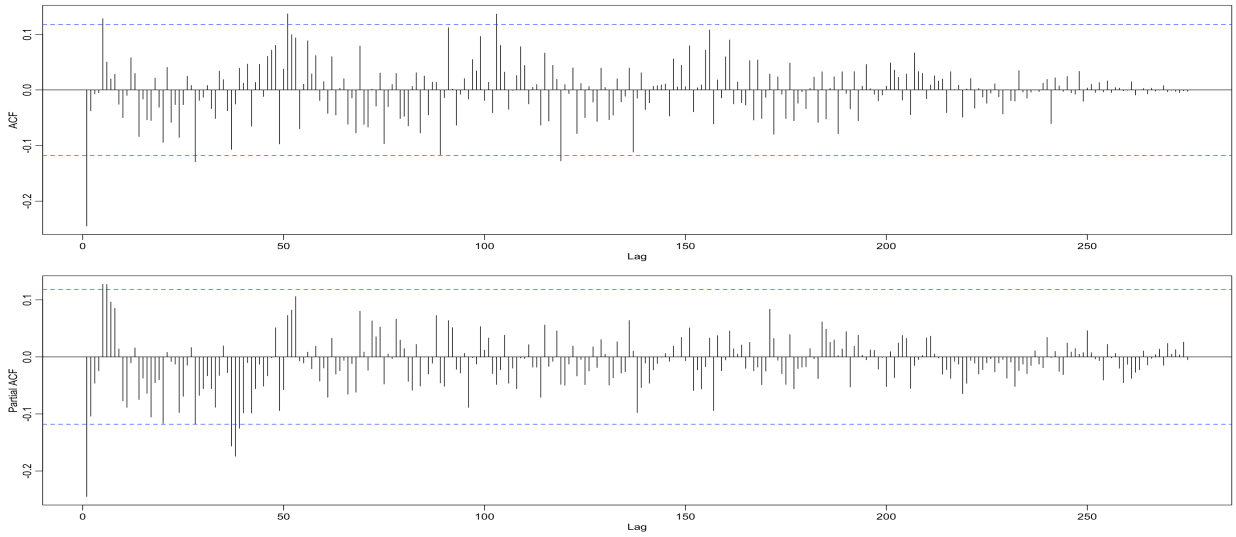

Figure 2.3. ACF and PACF of the square rooted and differenced Citi Bike series, $(1 - \mathrm{B})\mathrm{sqrt}(x_t)$

To recap this final step can be represented by the following equation:

$$\nabla_{51}\nabla \mathrm{sqrt}(x_t) = (1 - \mathrm{B}^{51})(1 - \mathrm{B})\mathrm{sqrt}(x_t) \quad (1)$$

Where $\mathrm{sqrt}(x_t)$ is the transformed Citi Bike Weekly rides.

2.1.2. Analysis of ACF and PACF, and Selection of Candidate Models

By analyzing the ACF and PACF shown in Figure 2.4 – which were calculated by using the data with the transformation depicted by equation (1) – we can get an idea of what models might be good candidates for representing the original data.

To select the potential candidates, we first focus on the behavior of the ACF and PACF at the seasonal lags 51, 102, 153, 204, 255. From this we see that it appears that either:

i. Both the ACF and the PACF tail off: Candidate model = SARMA(1,1) *
ii. ACF cuts off at seasonal lag 1 (lag 51) and PACF tails off: Candidate model = SMA(1)
iii. ACF tails off and PACF cuts off at seasonal lag 1: Candidate model = SAR(1)

*(with p = 1 because one seasonal spike in PACF and q = 1 because 1 spike in ACF)

Having selected these, we turn our attention to the within seasonalities sections of the ACF and PACF. From this we see that it appears that either:

a. Both the ACF and the PACF tail off: Candidate model = ARMA(1,1)
b. ACF cuts off at lag 1 and PACF tails off: Candidate model = MA(1)

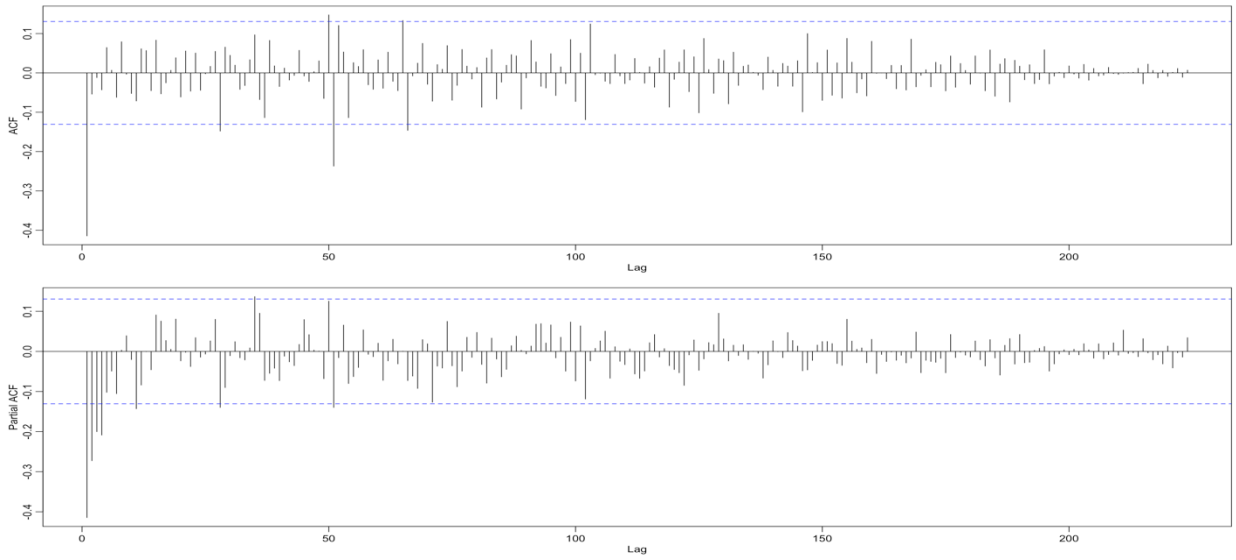

Figure 2.4. ACF and PACF of the square rooted and differenced Citi Bike series, $(1 - \mathrm{B}^{51})(1 - \mathrm{B})\mathrm{sqrt}(x_t)$

The six resulting full candidates are:

(a) x (i) = ARIMA(1,1,1) x $(1,1,1)_{51}$

(a) x (ii) = ARIMA(1,1,1) x $(0,1,1)_{51}$

(a) x (iii) = ARIMA(1,1,1) x $(1,1,0)_{51}$

(b) x (i) = ARIMA(0,1,1) x $(1,1,1)_{51}$

(b) x (ii) = ARIMA(0,1,1) x $(0,1,1)_{51}$

(b) x (iii) = ARIMA(0,1,1) x $(1,1,0)_{51}$

2.1.3. Diagnostics and Evaluation of Candidate Models

The diagnostics were analyzed for all the candidate models, and among them the three best were selected. The AIC, BIC and AICc for these three models are shown in the following table:

Model	AIC	BIC	AICc
ARIMA(1,1,1) x (0,1,1) s=51	8.420327	7.459576	8.428078
ARIMA(0,1,1) x (1,1,0) s=51	8.505964	7.53213	8.513502
ARIMA(0,1,1) x (1,1,1) s=51	8.261425	7.300675	8.269176

From the table we can see that the model ARIMA(0,1,1) x $(1,1,1)_{51}$ looks like the most promising one, for it has the lowest values across all the information criteria. However, on the graphs of the diagnostics of fit the model ARIMA(1,1,1) x $(0,1,1)_{51}$ shows better results. The diagnostics of fit for both models can be seen on Figure 2.5 and Figure 2.6.

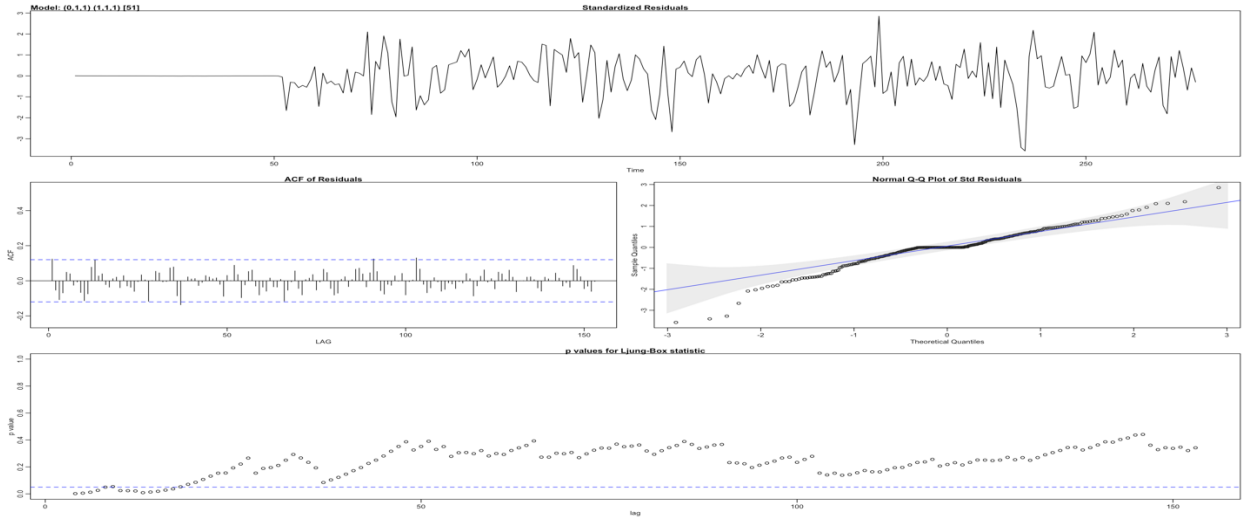

Figure 2.5. Diagnostics for model ARIMA(0,1,1) x $(1,1,1)_{51}$

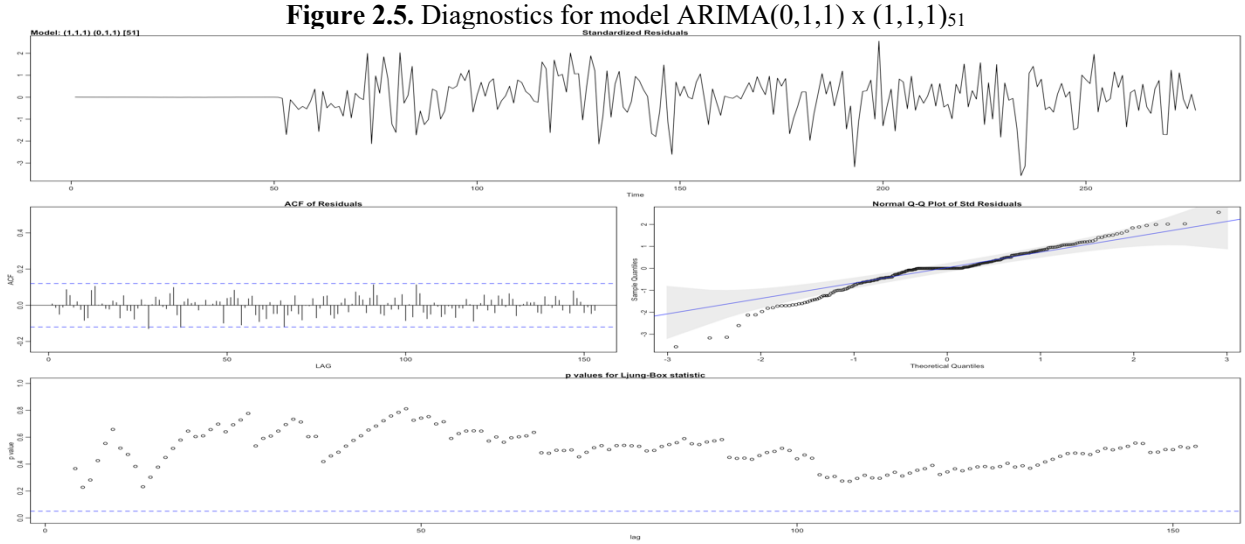

Figure 2.6. Diagnostics for model ARIMA(1,1,1) x $(0,1,1)_5$

Aside from the comparative performance, there are some concerns that show up for both models that are important to note. From the plots of the standardized residuals we can see that there are a few outliers in the series. From their normal Q-Q plot we can see they have heavy tails, slightly more so on the left side. And from the ACF of the residuals we can see that a small amount of autocorrelation still remains (as seen from some spikes that barely grace or surpass the limits). Aside from these concerns the models fit well, and given their relatively similar performance – in the following section we will use cross validation to evaluate them, and select the one that shows better results for the final forecasting.

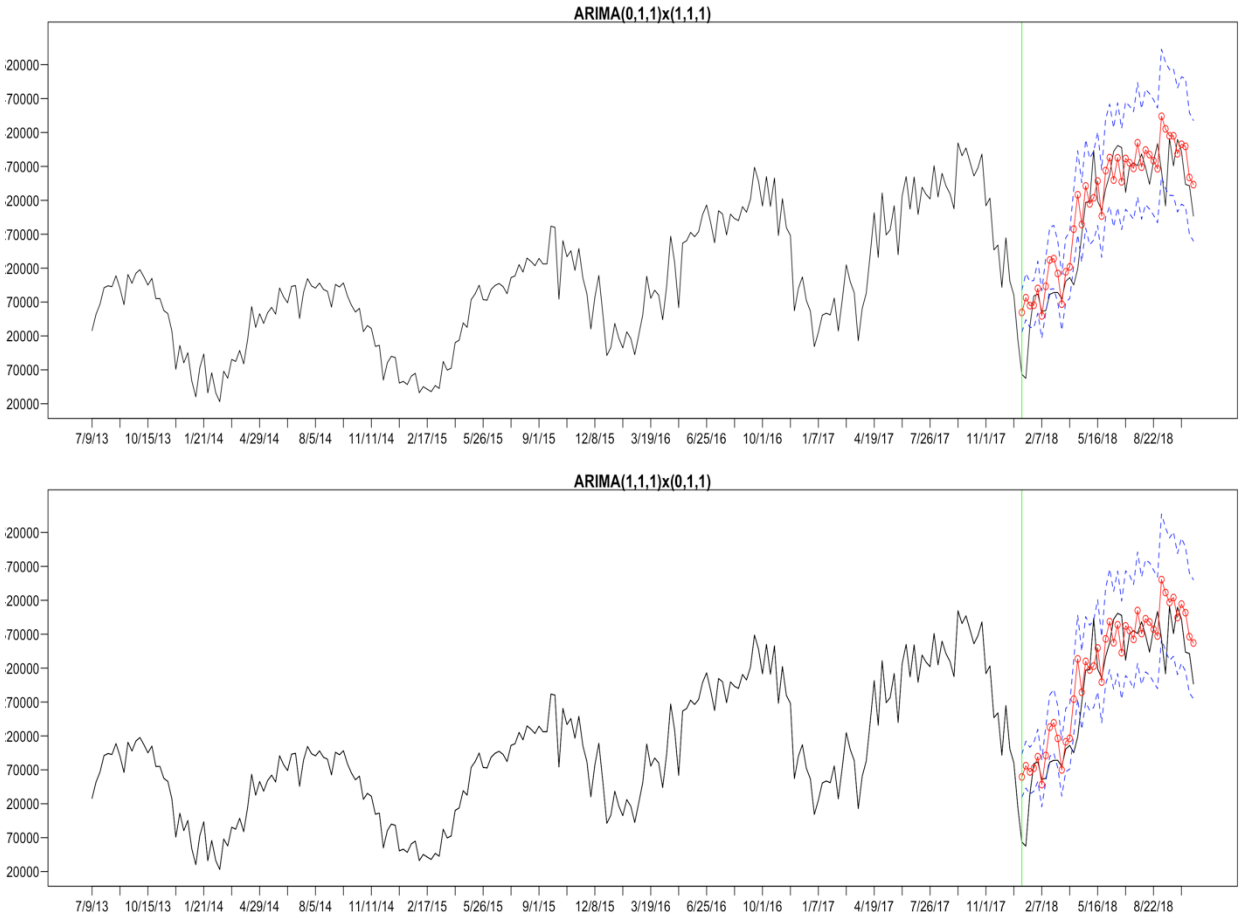

Figure 2.7. Forecasting for January 2018 to October 2018, Top ARIMA(0,1,1) x $(1,1,1)_{51}$, Bottom ARIMA(1,1,1) x $(0,1,1)_{51}$. The red line is the forecast, the blue lines are the std. errors, the green line marks the start of the forecasting.

2.1.4. Cross Validation and Final Model Selection

Given the relatively similar power of models, cross validation testing was used to further assess which model was better. For the CV test, the models were re-estimated while leaving out a total of 44 weeks that encompass the time frame from January 2018 to October 2018. After re-estimating the models on the reduced data set, they were then used to forecast the weeks that were left out (the forecast is shown in Figure 2.7) and their sum of square errors was calculated. The results for the sum of square errors were 98259945534 and 105956867623 for ARIMA(0,1,1) x $(1,1,1)_{51}$ and ARIMA(1,1,1) x $(0,1,1)_{51}$ respectively.

From the graphs we note that visually there is not a great difference between the forecasts, so we rely sum of square errors to select the final model. The SSE indicates the selection of the model ARIMA(0,1,1) x $(1,1,1)_{51}$.

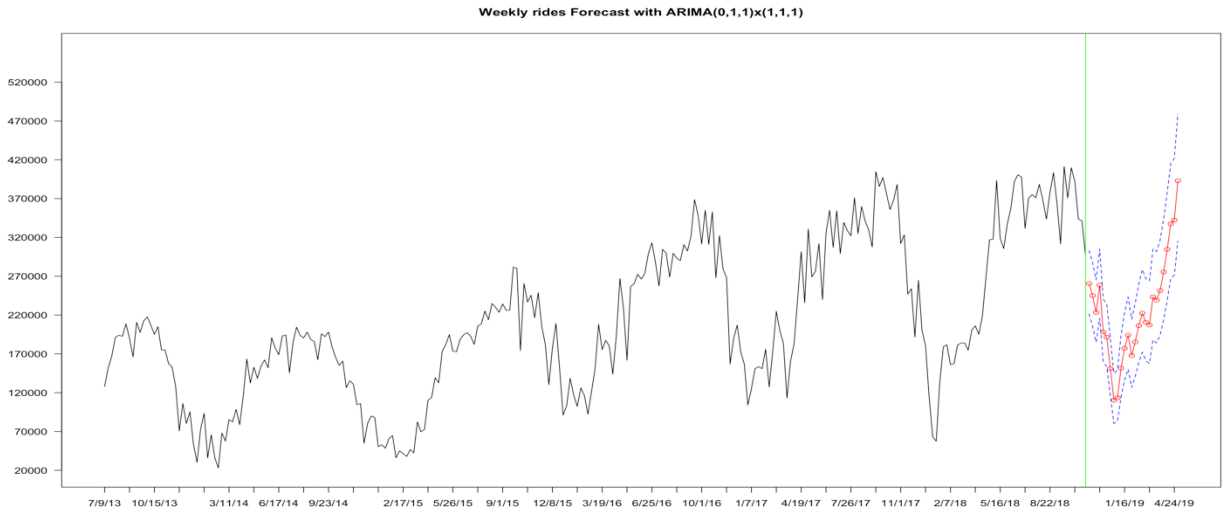

Figure 2.8. Forecasting from the beginning of November 2018 to the end of April 2019, using the model ARIMA(0,1,1) x $(1,1,1)_{51}$.

3. Forecast and Conclusion

For the final forecast, the model ARIMA(0,1,1) x $(1,1,1)_{51}$ was used to forecast 26 weeks ahead – which corresponds to roughly to 6 months, that is, from the beginning of November 2018 to the end of April 2019.

The first thing we note for the forecasts shown – in both section 2.1.4. and Figure 3.1 – is that they are reasonable. Moreover, we can appreciate that they both closely resemble the shape of the season immediately before; although in the case of this final forecast, we can observe a little bit of an increase when comparing the year to year growth. This increase is something we would expect if the upward trend shown in the data should continue. All of this indicates that the model is suitable, but in order to assess it absolute validity we would have to wait to see how the estimates fare with reality.

R CODE (it's a little messy)

```
rm(list = ls())    #delete objects
cat("\014")
library(ISLR)
library(astsa)
library(forecast)
library(TSA)
library(xts)

monthlyrides <- read.csv("/Users/root1/Documents/Python_Files/Outputs/monthlyrides.csv", header=TRUE)
weeklyrides <- read.csv("/Users/root1/Documents/Python_Files/Outputs/weeklyrides.csv", header=TRUE)

# x <- subset(weeklyrides, select=c("week_end", "rides"))
data1 <- weeklyrides[3]

# prepare the data
data1 <- ts(data1[,1])



#=============================================#
######### Exploratory Data Analysis #############
#=============================================#

##====================##
## First Plot the Data ##
##====================##

plot.ts(data1, main = "Weekly Rides")

par ( mfrow =c(2 ,1), mar=c(3 ,3 ,1.5 ,1.5) , mgp=c (1.6 ,.6 ,0) )
acf(data1,length(data1), main = "")
pacf(data1,length(data1), main = "")

#=============================================#
## Variance Stabilization by log transformation##
#=============================================#

BoxCox.lambda(data1) #= 0.4869091 Close to 0, therefore use Square Root Transformaation



sqrt.data1 = sqrt(data1)
```

```r
plot(sqrt.data1, main = "Square Rooted Weekly Rides")

##=================================##
## Estimate and Eliminate the Trend ##
##=================================##


#####          Diferenced Sqrt Data 1        ############
diff.sqrt.data1 = diff(sqrt.data1)

#####  Seasonal Difference diff(Sqrt Data 1) ############

lag.diff.sqrt.data1 = diff(diff.sqrt.data1,51)


#####  Plot Differences ############

par ( mfrow =c(3 ,1), mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )
plot(sqrt.data1,main="sqrt(Weekly Rides)")
plot(diff.sqrt.data1,main="Differenced sqrt(Weekly Rides)")
plot(lag.diff.sqrt.data1,main="lag Differenced sqrt(Weekly Rides)")


########       ACFs               #########

par ( mfrow=c(3 ,1) , mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )
acf(sqrt.data1           , lag.max = 275 , main = "ACF of sqrt(Weekly Rides)"             )
acf(diff.sqrt.data1      , lag.max = 275 , main = "ACF of differenced sqrt(Weekly Rides)" )
acf(lag.diff.sqrt.data1  , lag.max = 275 , main = "ACF of lag differenced sqrt(Weekly Rides)" )

########       PACFs             #########

par ( mfrow=c(3 ,1) , mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )
pacf(sqrt.data1           , lag.max = 275 , main = "ACF of sqrt(Weekly Rides)"             )
pacf(diff.sqrt.data1      , lag.max = 275 , main = "ACF of differenced sqrt(Weekly Rides)" )
pacf(lag.diff.sqrt.data1  , lag.max = 275 , main = "ACF of lag differenced sqrt(Weekly Rides)" )


##        Observations   & Candidate Models        ##
```

```
#The data appears to show signs of seasonality around the week 51 or 52 (Although in a yearly data set we would usually
# expect 52 weeks). The seasonality is made clear by the spikes in the ACF at lags~~ 51, 103
# using s = 51
# using s = 52
# We can see at seasonal lags:
# It appears that either,
#   (i)  ACF either tails off  and the PACF tails off: Candidate model = SARIMA(1,1,1) (p = 1 beaucuase one seasonal spike in
PACF and q = 1 because 1 spike in ACF)
#   (ii)  ACF cuts off at slag 1 and PACF tails off: : Candidate model = SARIMA(0,1,1)
#   (iii) ACF tails off and PACF cuts off at slag 1: : Candidate model = SARIMA(1,1,0)

# Within seasons candidates :
# It appears that either,
#   (a)   ACF either tails off  and the PACF tails off: Candidate model = ARIMA(1,1,1) (p = 1 beaucuase one seasonal spike in
PACF and q = 1 because 1 spike in ACF)
#   (b)  ACF cuts off at slag 1 and PACF tails off: : Candidate model = ARIMA(0,1,1)

#Full candidate models:
#  (a) x (i)   =  ARIMA(1,1,1) x (1,1,1) s=51
#  (a) x (ii)  =  ARIMA(1,1,1) x (0,1,1) s=51
#  (a) x (iii) =  ARIMA(1,1,1) x (1,1,0) s=51

#  (b) x (i)   =  ARIMA(0,1,1) x (1,1,1) s=51
#  (b) x (ii)  =  ARIMA(0,1,1) x (0,1,1) s=51
#  (b) x (iii) =  ARIMA(0,1,1) x (1,1,0) s=51


#Diagnostics
sarima.111.111  =  sarima(sqrt.data1, 1, 1, 1, 1, 1, 1, 51) # Good
sarima.111.011  =  sarima(sqrt.data1, 1, 1, 1, 0, 1, 1, 51) # Better
sarima.111.110  =  sarima(sqrt.data1, 1, 1, 1, 1, 1, 0, 51) # Almost as Good as first, but has errors

sarima.011.111  =  sarima(sqrt.data1, 0, 1, 1, 1, 1, 1, 51) # 1 Good with a few issues on the Ljung-Box statistic
sarima.011.011  =  sarima(sqrt.data1, 0, 1, 1, 0, 1, 1, 51) # 3
sarima.011.110  =  sarima(sqrt.data1, 0, 1, 1, 1, 1, 0, 51) # 2 Good but not as good

# Finalists
sarima.111.011
sarima.011.111


Table_AIC_BIC_Comparission = data.frame('Model'= c('ARIMA(1,1,1) x (0,1,1) s=51'  ,  'ARIMA(0,1,1) x (1,1,0) s=51'  ,
'ARIMA(0,1,1) x (1,1,1) s=51' ),
```

```
                                    'AIC'  = c(sarima.111.011$AIC   ,  sarima.011.110$AIC   ,  sarima.011.111$AIC )  ,
                                    'BIC'  = c(sarima.111.011$BIC   ,  sarima.011.110$BIC   ,  sarima.011.111$BIC  ) ,
                                    'AICc' = c(sarima.111.011$AICc  ,  sarima.011.110$AICc  ,  sarima.011.111$AICc ) )


Table_AIC_BIC_Comparission
## Forecasts ##
sarima.011.111.forcast  =  sarima.for(sqrt.data1, 26, 0, 1, 1, 1, 1, 1, 51) # 1 All p values success
sarima.111.011.forcast  =  sarima.for(sqrt.data1, 26, 1, 1, 1, 0, 1, 1, 51) # Good!



#             Comparisson of AIC, BIC, and AICc
Table_AIC_BIC_Comparission = data.frame('Model'= c('ARIMA(1,1,1) x (0,1,1) s=51'  ,  'ARIMA(0,1,1) x (1,1,0) s=51'  ,
'ARIMA(0,1,1) x (0,1,1) s=51' ),
                                    'AIC'  = c(sarima.111.011$AIC   ,  sarima.011.110$AIC   ,  sarima.011.011$AIC )  ,
                                    'BIC'  = c(sarima.111.011$BIC   ,  sarima.011.110$BIC   ,  sarima.011.011$BIC  ) ,
                                    'AICc' = c(sarima.111.011$AICc  ,  sarima.011.110$AICc  ,  sarima.011.011$AICc ) )
Table_AIC_BIC_Comparission
# Cross Validation # forcasting the year 2018 up until october (44 weeks)
#Measure of comparison sum of squre errors

sarima.011.111.forcast.cv  =  sarima.for(sqrt.data1[1:233], 44, 0, 1, 1, 1, 1, 1, 51) # 1 All p values success
sarima.111.011.forcast.cv  =  sarima.for(sqrt.data1[1:233], 44, 1, 1, 1, 0, 1, 1, 51) # Good!

#Plot the predictions against the real data ##

U_y_limit = max(data1)+150000
L_y_limit = min(data1)-10000

x_labels_for_plots = weeklyrides[,2][seq(from = 1, to = 280, by =7)]
y_labels_for_plots = seq(from = 20000, to = U_y_limit, by = 50000)

par ( mfrow=c(2 ,1) , mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )

plot.ts(data1, main = 'ARIMA(0,1,1)x(1,1,1)', axes = F ,ylim = c(20000,U_y_limit), xlab = "",ylab="")
lines(sarima.011.111.forcast.cv$pred^2, col="red", type="o")
lines((sarima.011.111.forcast.cv$pred + sarima.011.111.forcast.cv$se) ^ 2 , col="blue", lty="dashed")
lines((sarima.011.111.forcast.cv$pred - sarima.011.111.forcast.cv$se) ^ 2 , col="blue", lty="dashed")
lines(abline(v = 234, col="green"))
axis(1, labels= x_labels_for_plots, at=seq(from = 1, by=7, to=length(weeklyrides[,2])) )
axis(2, labels= y_labels_for_plots, at=seq(from = 20000, by=50000, to = U_y_limit) ,las=1)
box()
```

```
plot( data1, main = 'ARIMA(1,1,1)x(0,1,1)',lty ='solid', axes = F ,ylim = c(20000,U_y_limit), xlab = "",ylab="")
lines(data1)
lines(sarima.111.011.forcast.cv$pred^2, col="red", type="o")
lines((sarima.111.011.forcast.cv$pred + sarima.111.011.forcast.cv$se) ^ 2 , col="blue", lty="dashed")
lines((sarima.111.011.forcast.cv$pred - sarima.111.011.forcast.cv$se) ^ 2 , col="blue", lty="dashed")
lines(abline(v = 234, col = 'green'))
axis(1, labels= x_labels_for_plots, at=seq(from = 1, by=7, to=length(weeklyrides[,2])) )
axis(2, labels= y_labels_for_plots, at=seq(from = 20000, by=50000, to = U_y_limit) ,las = 1)
box()

##           Obtain the sum of square errors        ##

SSE_sarima.011.111.cv = sum((sarima.011.111.forcast.cv$pred^2 -  data1[-1:-233])^2)
SSE_sarima.111.011.cv = sum((sarima.111.011.forcast.cv$pred^2 -  data1[-1:-233])^2)


SSE_sarima.011.111.cv
SSE_sarima.111.011.cv

#More plots for paper


forecast_labels <- read.csv("/Users/root1/Documents/Python_Files/Outputs/forecast_labels.csv", header=TRUE)

x_labels_for_plots_forecast_26weeks = forecast_labels[,1][seq(from = 1, to = length(weeklyrides[,2])+26, by = 7)]




plot.ts(data1, main = 'Weekly rides Forecast with ARIMA(0,1,1)x(1,1,1)', axes = F ,ylim = c(20000,U_y_limit), xlim =
c(1,length(weeklyrides[,2])+26), xlab = "",ylab="")
lines(sarima.011.111.forcast$pred^2, col="red", type="o")
lines((sarima.011.111.forcast$pred + sarima.011.111.forcast$se) ^ 2 , col="blue", lty="dashed")
lines((sarima.011.111.forcast$pred - sarima.011.111.forcast$se) ^ 2 , col="blue", lty="dashed")
lines(abline(v = 277, col="green"))
axis(1, labels= x_labels_for_plots_forecast_26weeks, at=seq(from = 1, by=7, to=length(weeklyrides[,2])+26) )
axis(2, labels= y_labels_for_plots, at=seq(from = 20000, by=50000, to = U_y_limit),las=1 )
box()



plot( data1, main = 'ARIMA(1,1,1)x(0,1,1)',lty ='solid', axes = F , ylim = c(20000,U_y_limit), xlim =
c(1,length(weeklyrides[,2])+26))
```

```
lines(data1)
lines(sarima.111.011.forcast$pred^2, col="red", type="o")
lines((sarima.111.011.forcast$pred + sarima.011.110.forcast$se) ^ 2 , col="blue", lty="dashed")
lines((sarima.111.011.forcast$pred - sarima.011.110.forcast$se) ^ 2 , col="blue", lty="dashed")
lines(abline(v = 277), col="green")
axis(1, labels= x_labels_for_plots_forecast_26weeks, at=seq(from = 1, by=7, to=length(weeklyrides[,2])) )
axis(2, labels= y_labels_for_plots, at=seq(from = 20000, by=50000, to = U_y_limit) )
box()


plot.ts( data1, main = 'Weekly Rides',lty ='solid', axes = F , ylim = c(20000,450000), xlab = "", ylab = "")
axis(1, labels= x_labels_for_plots, at=seq(from = 1, by=7, to=length(weeklyrides[,2])),las = 3 )
axis(2, labels= y_labels_for_plots, at=seq(from = 20000, by=50000, to = U_y_limit) ,las=2)
box()



########       ACFs               #########

par ( mfrow=c(2 ,1) , mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )
acf(diff.sqrt.data1      , lag.max = 275 , main = "" )
pacf(diff.sqrt.data1      , lag.max = 275 , main = "" )

########       PACFs             #########

par ( mfrow=c(2 ,1) , mar=c(3 ,3 ,1 ,1) , mgp=c (1.6 ,.6 ,0) )
acf(lag.diff.sqrt.data1  , lag.max = 275 , main = "" )
pacf(lag.diff.sqrt.data1  , lag.max = 275 , main = "" )
```