Assignment_TS

Weijie Gao

10/26/2017

```
# load related packages
library(timeSeries)

## Loading required package: timeDate

library(forecast)
library(tseries)
library(TSA)

## Loading required package: leaps

## Loading required package: locfit

## locfit 1.5-9.1    2013-03-22

## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':
##
##     getResponse

## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

Data Preparation: For the convenience of time series analysis, the traffic counts in column I80E 1EXIT was extracted from each .xls files and combined into a csv file called Traffic_Flow_2013.csv. The new dataset has three variables: date, time, num. This dataset records an hourly count of the number of vehicles at I80E 1EXIT.

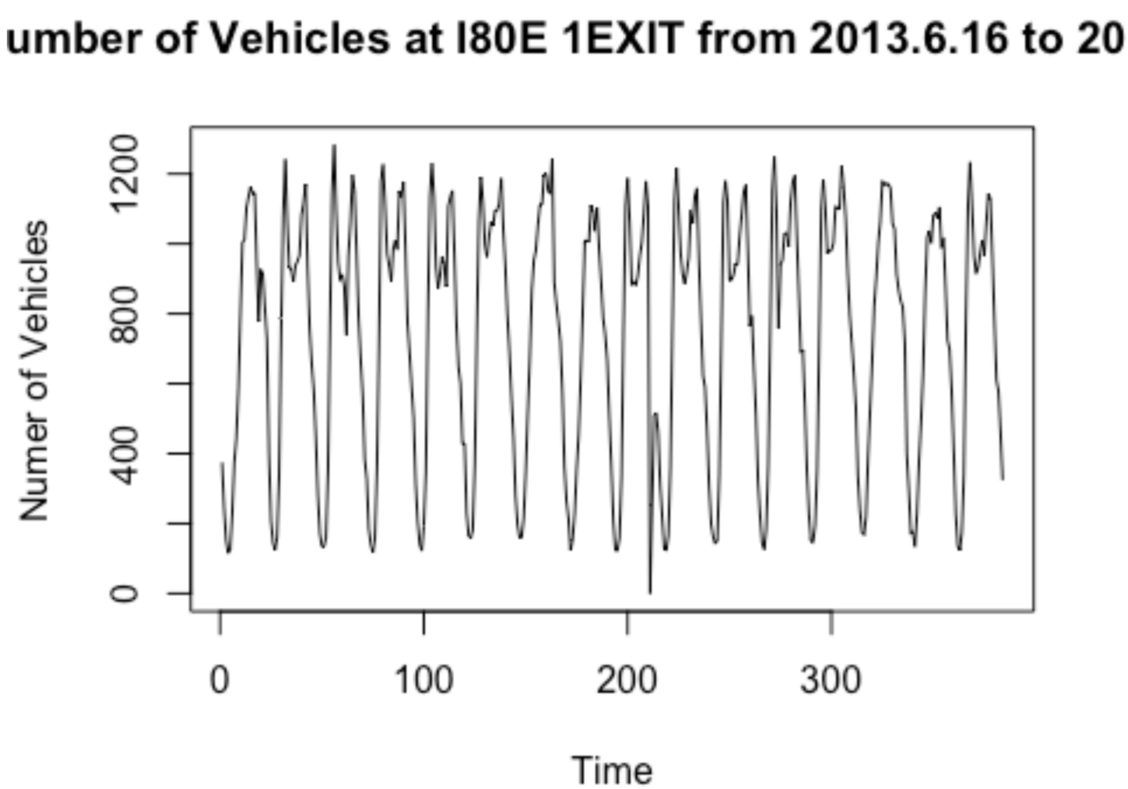
```
dataPath <- "/Users/gaoweijie/Google Drive/2017 Fall/Time Series/Week4"
traffic <-
read.csv(paste(dataPath,"Traffic_Flow_2013.csv",sep='/'),header=TRUE)
head(traffic)
```

```
##        Date  Time Counts
## 1  6/16/13 01:00    375
## 2  6/16/13 02:00    244
## 3  6/16/13 03:00    152
## 4  6/16/13 04:00    115
## 5  6/16/13 05:00    126
## 6  6/16/13 06:00    228
```

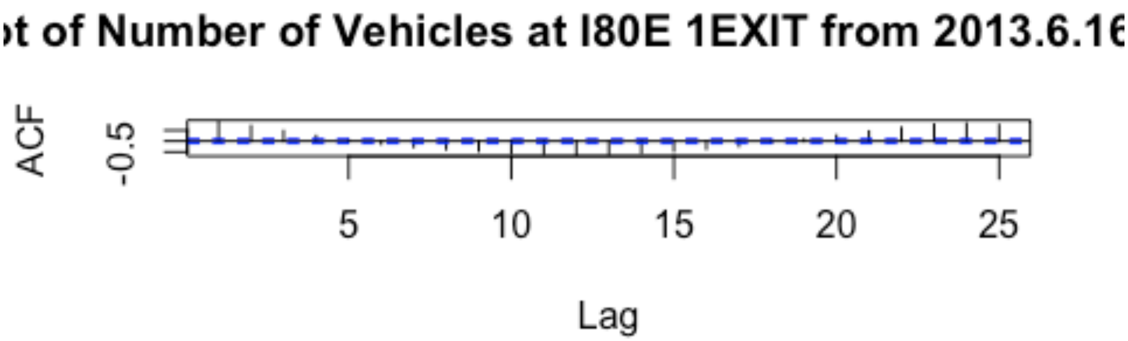
```
dim(traffic)
```

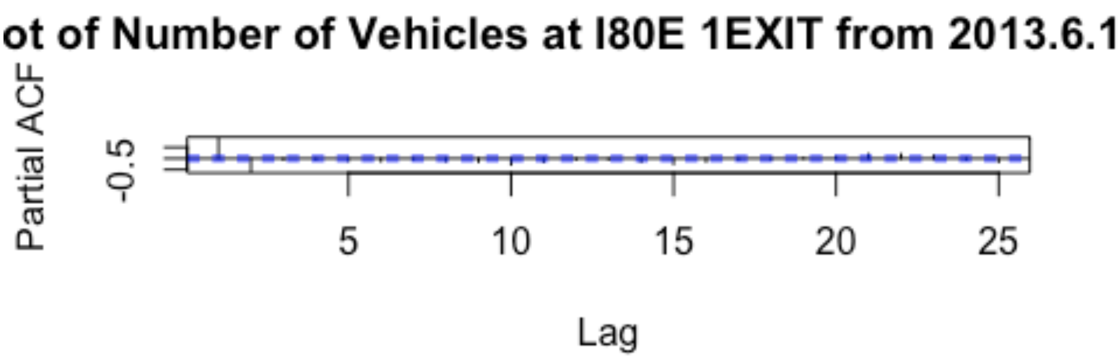
```
## [1] 384   3
```

```
plot(traffic[,3],type="l", xlab="Time", ylab="Numer of Vehicles", main =
"Number of Vehicles at I80E 1EXIT from 2013.6.16 to 2013.7.1")
```

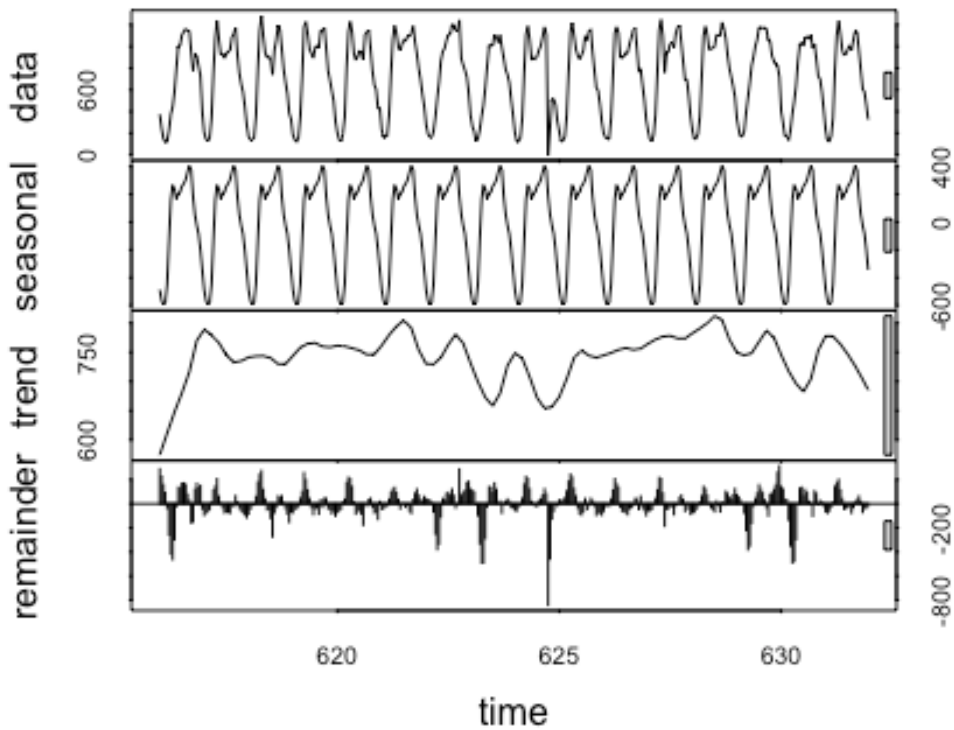


```
par(mfrow=c(2,1))
acf(traffic[,3],main="ACF plot of Number of Vehicles at I80E 1EXIT from 2013.6.16 to 2013.7.1")
pacf(traffic[,3],main="PACF plot of Number of Vehicles at I80E 1EXIT from 2013.6.16 to 2013.7.1")
```






```
par(mfrow=c(1,1))
traffic_ts <- ts(traffic[,3],start=616,freq=24)
plot(stl(traffic_ts,s.window="periodic"))
```



The above analysis shows that there is a clear seasonality in the data. And instead of having an obviously decreasing/increasing sign, the trend changes over time.

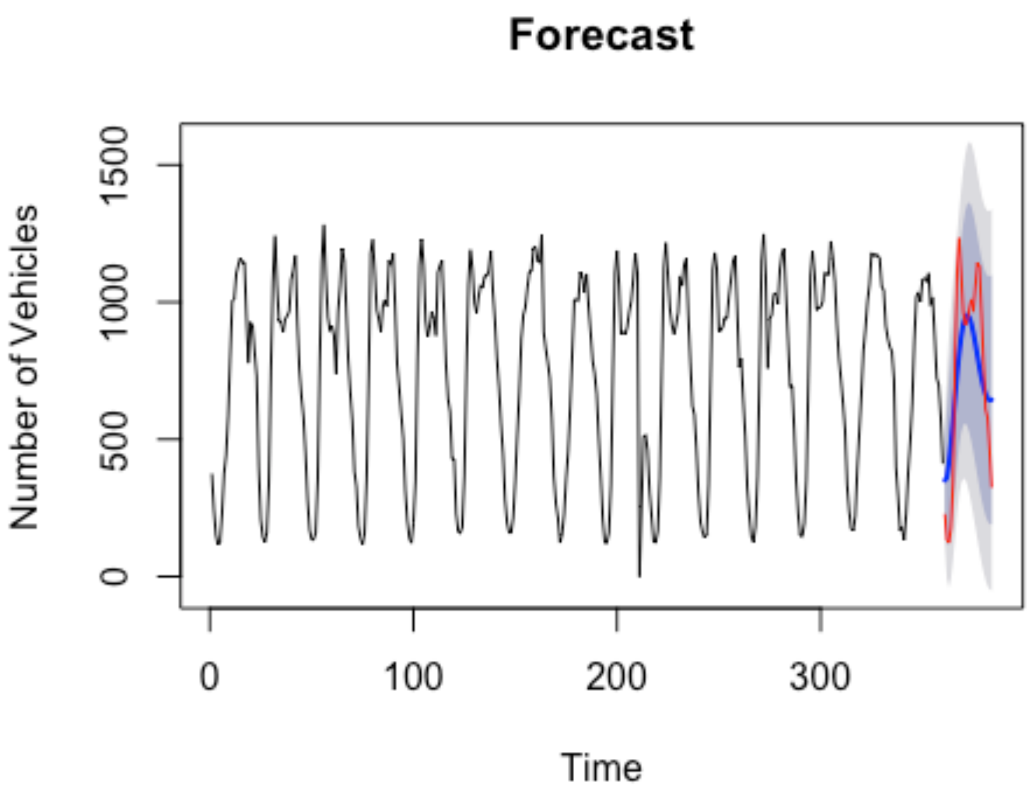
Part 1

```
# Train data: row 1-360 (Data for last 2 weeks of June 2013)
# Test data: row 361-384 (Data for July 1 2013)
train_data <- traffic[1:360,]
test_data <- traffic[361:384,]

fit1 <- auto.arima(train_data[,3], stepwise = FALSE, approximation = FALSE)
summary(fit1)
```

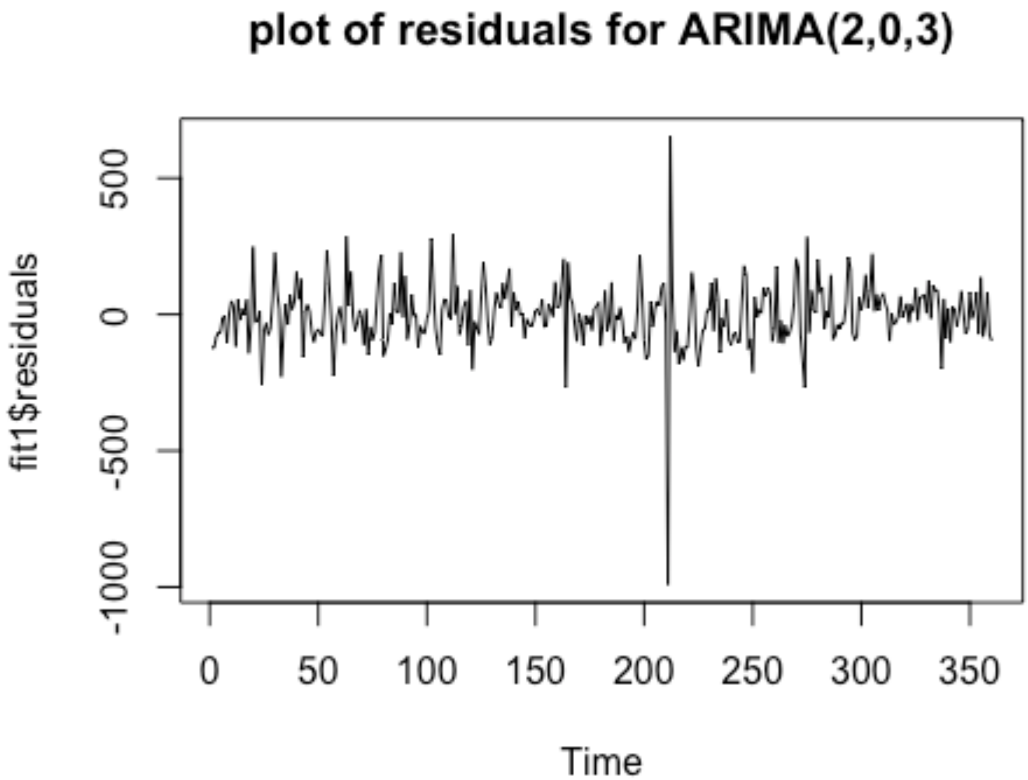
```
## Series: train_data[, 3]
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3      mean
##       1.8088  -0.8853  -0.5348  -0.2671  -0.1157  746.3181
## s.e.  0.0288   0.0287   0.0600   0.0596   0.0654    6.8586
##
## sigma^2 estimated as 13443:  log likelihood=-2220.78
## AIC=4455.56   AICc=4455.88   BIC=4482.77
##
```

```
## Training set error measures:
##                     ME     RMSE    MAE  MPE MAPE      MASE         ACF1
## Training set -1.390098 114.9732 79.019 -Inf  Inf 0.7027304 -0.003018285
```

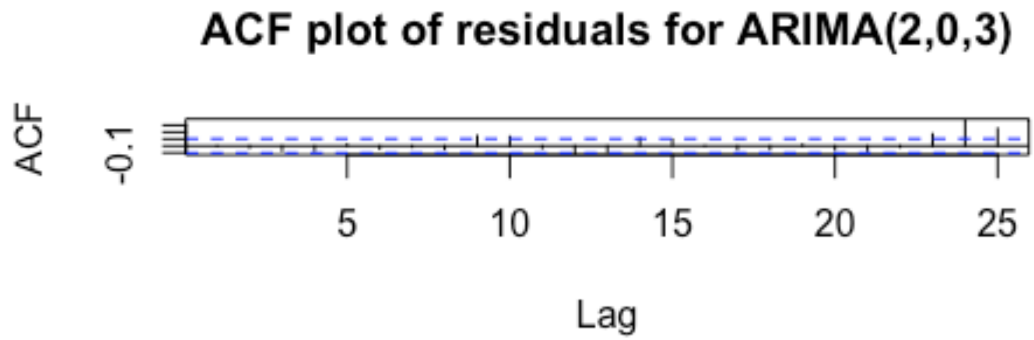
```
plot(forecast(fit1, 24), xlab="Time", ylab="Number of
Vehicles",main="Forecast")
lines(x=c(361:384), y =test_data[,3], col="red")
```

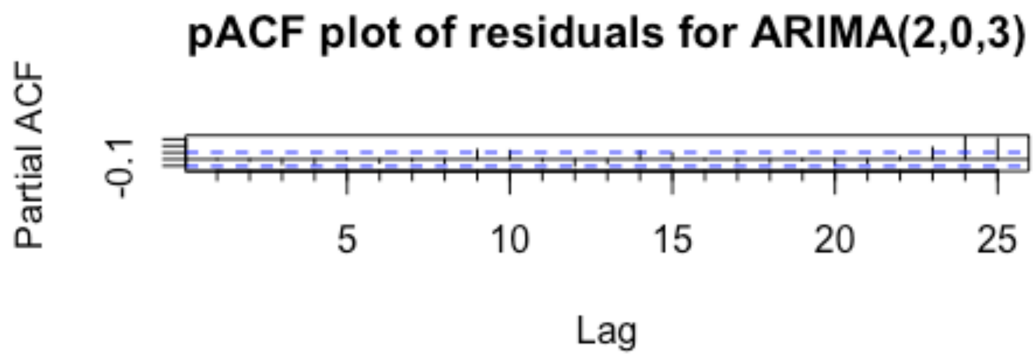



```
plot(fit1$residuals, main="plot of residuals for ARIMA(2,0,3)")
```




```r
par(mfrow=c(2,1))
acf(fit1$residuals, main="ACF plot of residuals for ARIMA(2,0,3)")
Pacf(fit1$residuals, main="pACF plot of residuals for ARIMA(2,0,3)")
```





The auto.arima() function returns a model of ARIMA(2,0,3) with AICc = 4455.88 and BIC = 4482.77. In the forecast plot, red line is the actual number of vehicles and blue line is the forecast line, and as we could seen that blue line does not match closely with the red line. Also in the residual plot, there is a huge spike around the middle time, and these suggest that our model might not be a good fit.

```
AICc_min <- 5000
AICc_min_p <- 0
AICc_min_q <- 0
for (p in 1:5){
  for (q in 1:5){
        fit11 <- Arima(train_data[,3], order = c(p,0,q))
        AICc <- fit11$aicc
        BIC <-fit11$bic
        if(AICc < AICc_min){
        AICc_min <- AICc
        AICc_min_p <- p
        AICc_min_q <- q}
  }
}
cbind(AICc_min=AICc_min, AICc_min_p=AICc_min_p,AICc_min_q=AICc_min_q)
```

```
##      AICc_min AICc_min_p AICc_min_q
## [1,] 4409.439          4          3
```

```r
BIC_min <- 5000
BIC_min_p <- 0
BIC_min_q <- 0
for (p in 1:5){
    for (q in 1:5){
        fit11 <- Arima(train_data[,3], order = c(p,0,q))
        AICc <- fit11$aicc
        BIC <-fit11$bic
        if(BIC < BIC_min){
            BIC_min <- BIC
            BIC_min_p <- p
            BIC_min_q <- q}
  }
}
cbind(BIC_min=BIC_min,BIC_min_p=BIC_min_p,BIC_min_q= BIC_min_q)
```

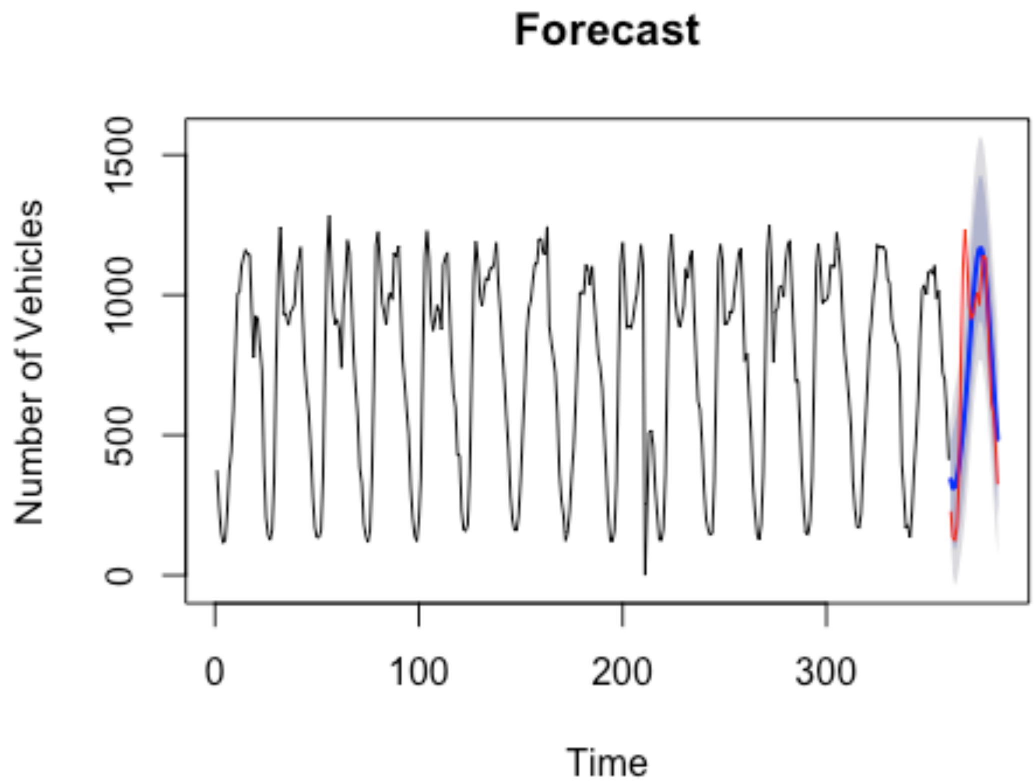
```
##      BIC_min BIC_min_p BIC_min_q
## [1,]  4443.9         4         3
```

Both AICc and BIC select the same model as the best model: ARIMA(4,0,3) with AICc=4409.439 and BIC=4443.9.

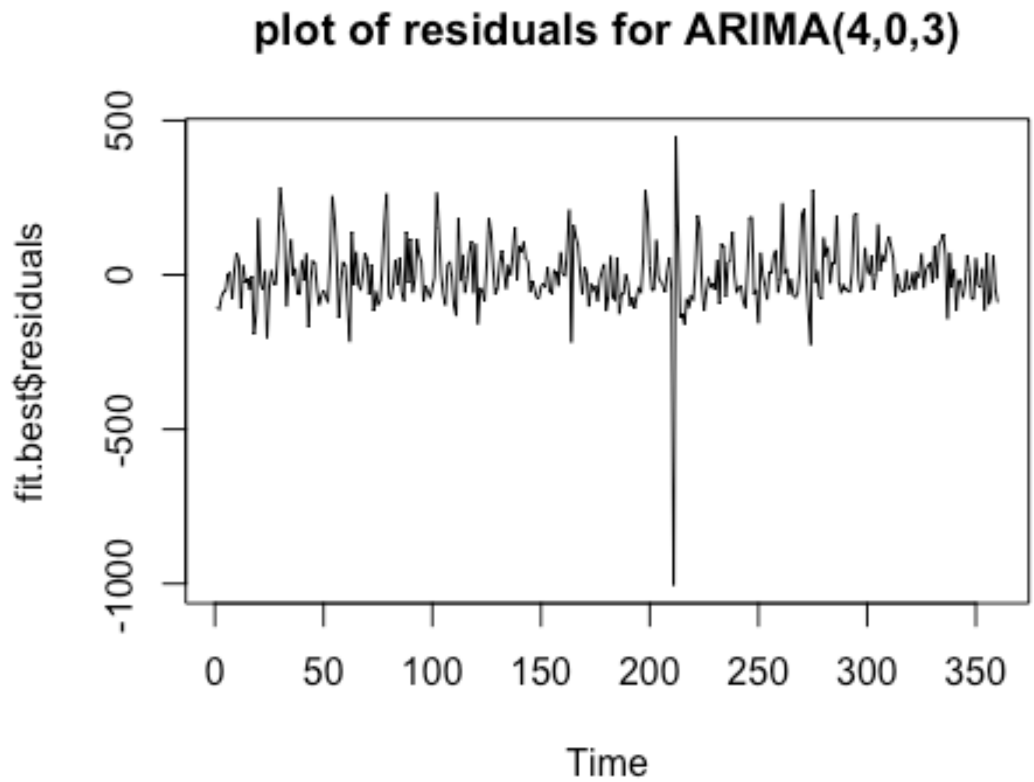
```r
fit.best <- Arima(train_data[,3], order=c(4,0,3))
fit.best
```

```
## Series: train_data[, 3]
## ARIMA(4,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1     ma2      ma3      mean
##       3.4089  -4.6362  2.9890  -0.7824  -2.3607  1.8739  -0.4776  743.2774
## s.e.  0.1767   0.4837  0.4521   0.1429   0.3015  0.6102   0.3283    9.8655
##
## sigma^2 estimated as 11649:  log likelihood=-2195.46
## AIC=4408.92   AICc=4409.44   BIC=4443.9
```

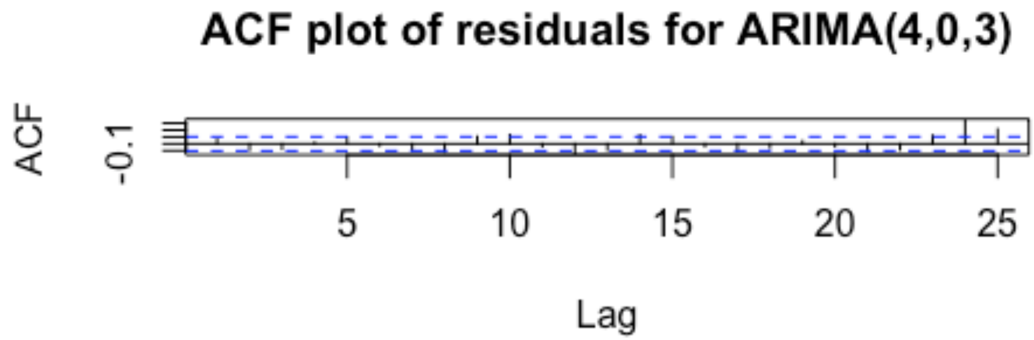
```r
plot(forecast(fit.best, 24), xlab="Time", ylab="Number of
Vehicles",main="Forecast")
lines(x=c(361:384), y =test_data[,3], col="red")
```

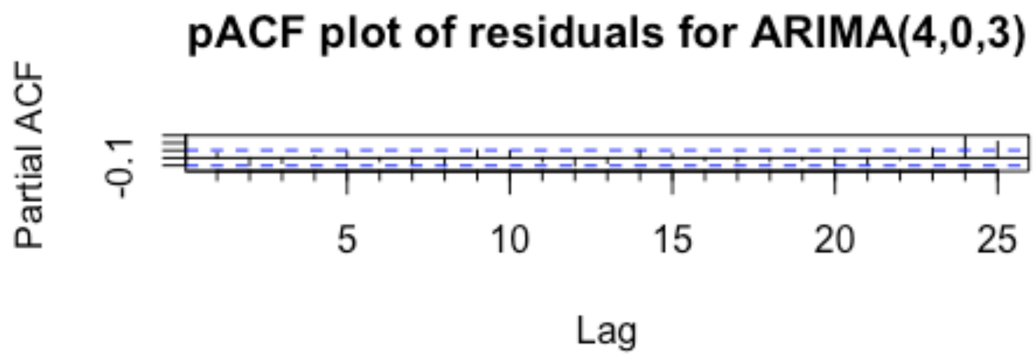



```
plot(fit.best$residuals, main="plot of residuals for ARIMA(4,0,3)")
```



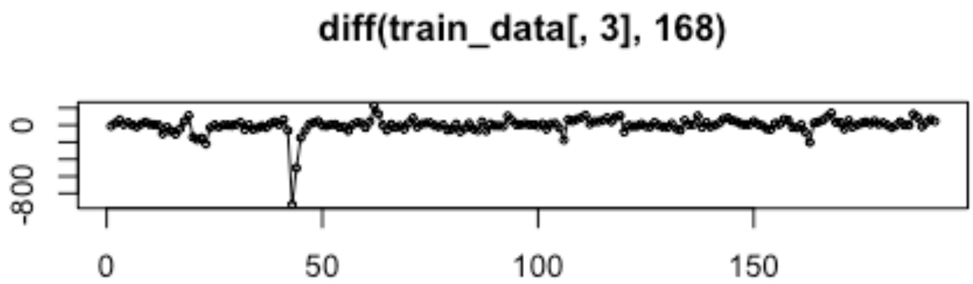

```
par(mfrow=c(2,1))
acf(fit.best$residuals, main="ACF plot of residuals for ARIMA(4,0,3)")
Pacf(fit.best$residuals, main="pACF plot of residuals for ARIMA(4,0,3)")
```

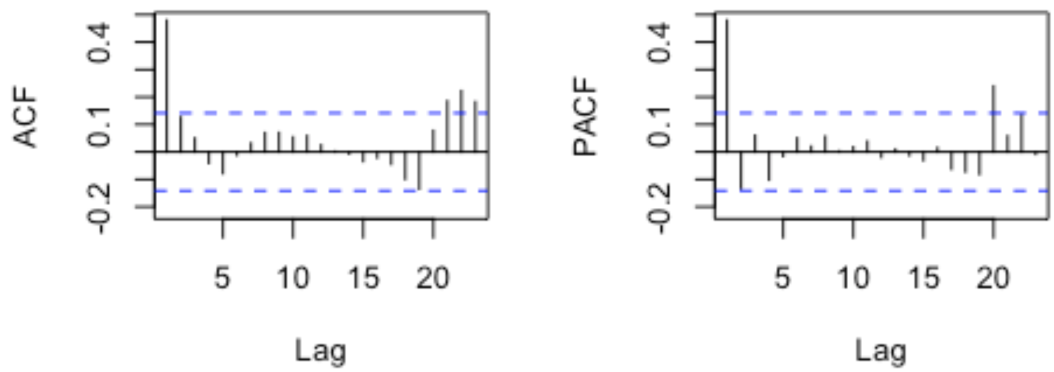




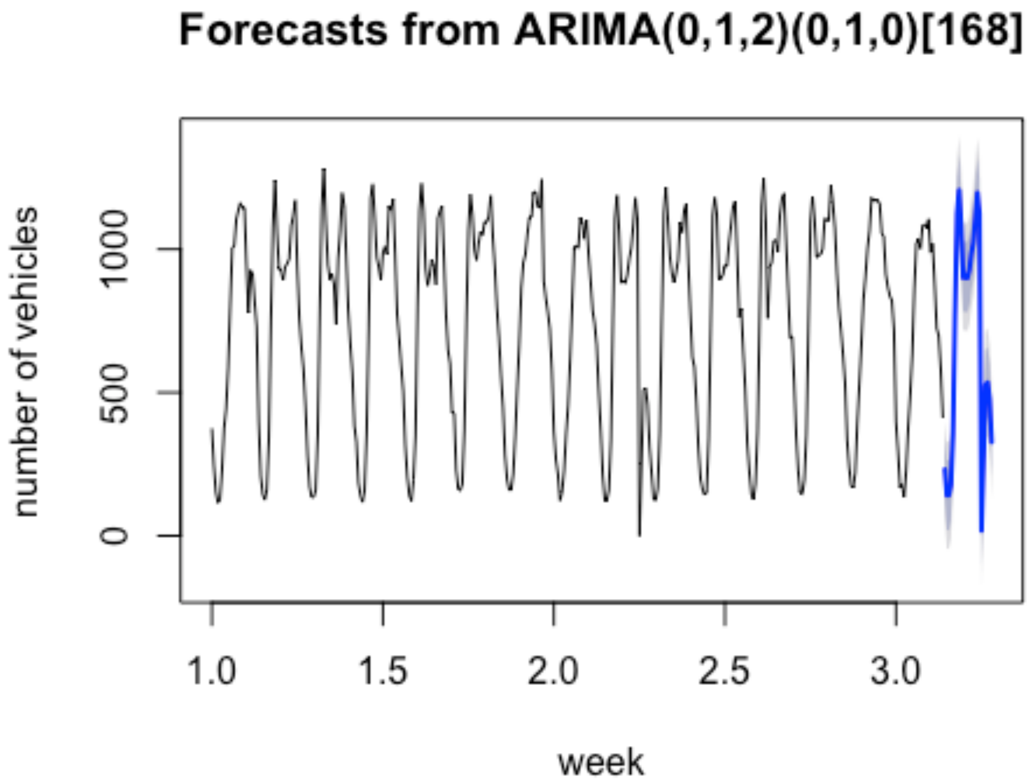
The best model is ARIMA(4,0,3) with AICc = 4409.439 and BIC = 4443.89. Both AICc and BIC are lower than that from AIC(2,0,3), suggesting our model of ARIMA(4,0,3) is better. In the forecast plot, the blue line matches the actual red line's better. However, in the residual plot, there is still a spike around the middle time, and this might suggests that our model could be further improved. Generally, ARIMA(4,0,3) is better than ARIMA(2,0,3).

```
# use day of the week: s=24*7=168
tsdisplay(diff(train_data[,3],168))
```

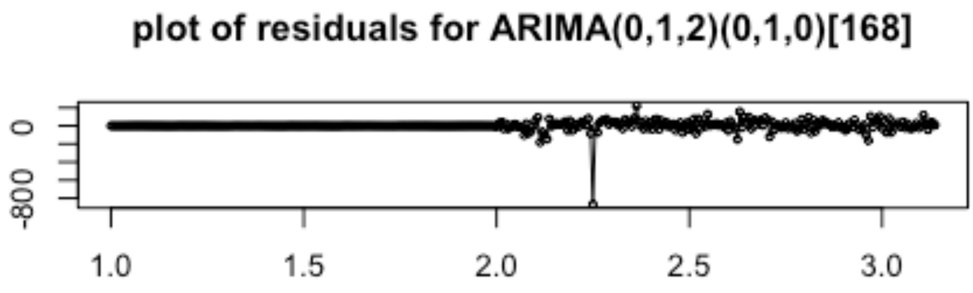


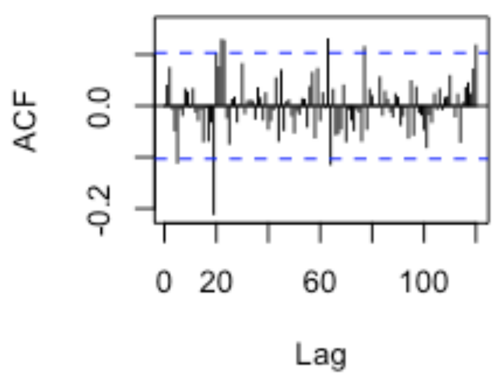


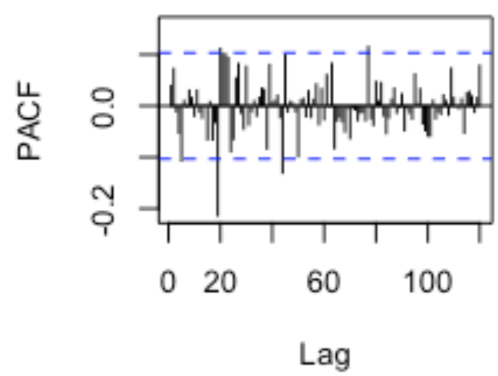

```r
# use day of the week: s=24*7=168
fit2 <- auto.arima(ts(train_data[,3], frequency=168))
fit2

## Series: ts(train_data[, 3], frequency = 168)
## ARIMA(0,1,2)(0,1,0)[168]
##
## Coefficients:
##           ma1      ma2
##       -0.4741  -0.4853
## s.e.   0.0593   0.0586
##
## sigma^2 estimated as 7081:  log likelihood=-1121.66
## AIC=2249.31   AICc=2249.44   BIC=2259.07

# forecast for July 1
fit2.forecast.July1 <- forecast(fit2,24)
fit2.predict <- data.frame(forecast(fit2,24))[,1]

(rmse.arima <- sqrt(mean((test_data[,3] - fit2.predict)^2)))

## [1] 221.8351

plot(fit2.forecast.July1, xlab="week", ylab="number of vehicles")
```




```
tsdisplay(fit2$residuals, main = "plot of residuals for
ARIMA(0,1,2)(0,1,0)[168]" )
```





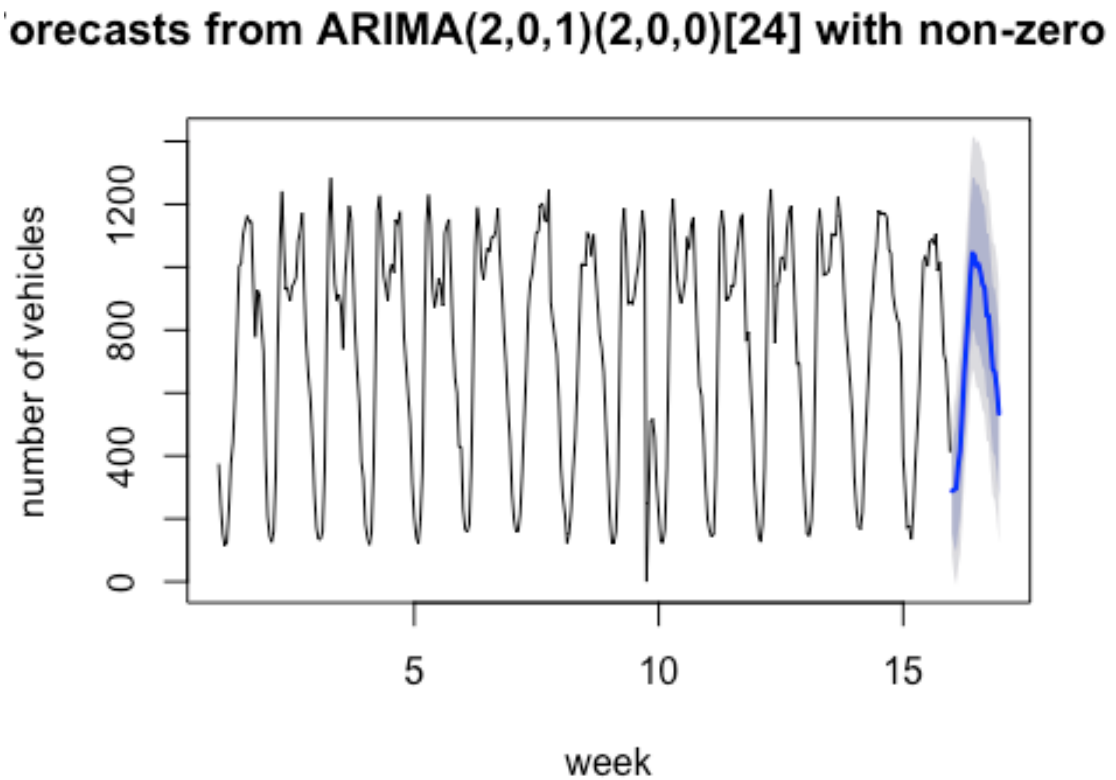


Use day of the week, I fit a seasonal ARIMA(0,1,2)(0,1,0) model with AICc = 2249.44 and BIC = 2259.07. In the residual plot, it looks like no pattern for most of time except an outlier data near the middle time. Also both ACF and PACF plot have fewer spikes exceeding bounds than before.

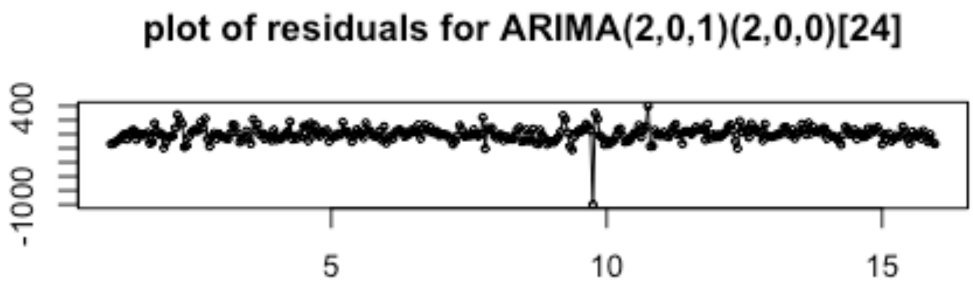
Part 3

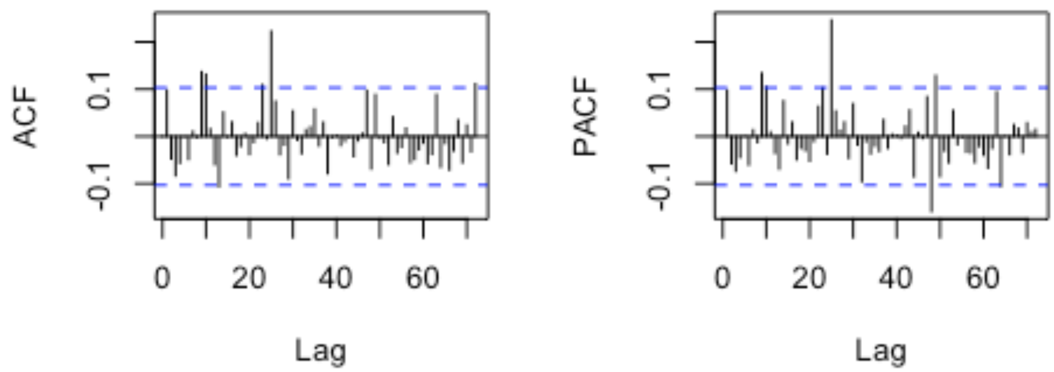
```
# use hour of the day: s=24
fit3 <- auto.arima(ts(train_data[,3], frequency=24))
fit3

## Series: ts(train_data[, 3], frequency = 24)
## ARIMA(2,0,1)(2,0,0)[24] with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1    sar1    sar2      mean
##       1.7922  -0.8685  -0.9146  0.4866  0.1010  743.7286
## s.e.  0.0299   0.0291   0.0257  0.0555  0.0557   13.6793
##
## sigma^2 estimated as 10737:  log likelihood=-2184.12
## AIC=4382.23   AICc=4382.55   BIC=4409.43
```

```r
# forecast for July 1
fit3.forecast.July1 <- forecast(fit3,24)
plot(fit3.forecast.July1, xlab="week", ylab="number of vehicles")
```




```r
tsdisplay(fit3$residuals, main = "plot of residuals for
ARIMA(2,0,1)(2,0,0)[24]" )
```





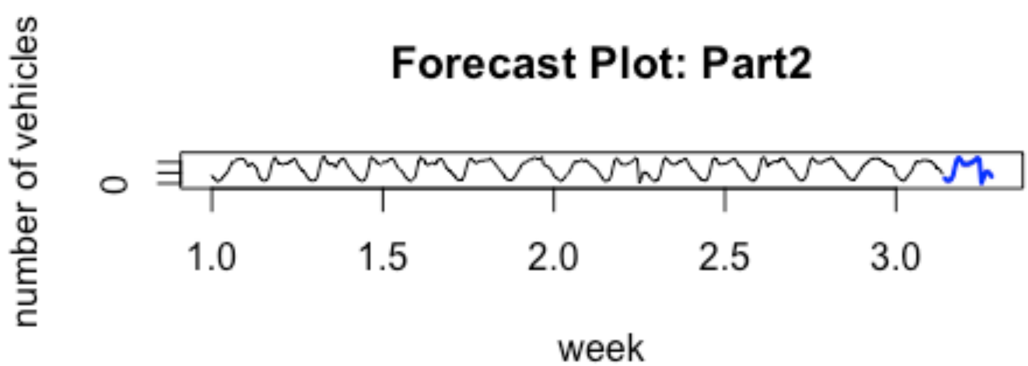
Use hour of the day, I fit a seasonal ARIMA(2,0,1)(2,0,0)[24] model with AICc=4382.55 and BIC=4409.43. In the residual plot, it looks like no pattern for most of time except an outlier data near the middle time. Both ACF and PACF plot have fewer spikes exceeding bounds than before, which is a good sign. But in the forecast plot, the shape of the blue line seems does not match actual data as well as the blue line in Part 2.

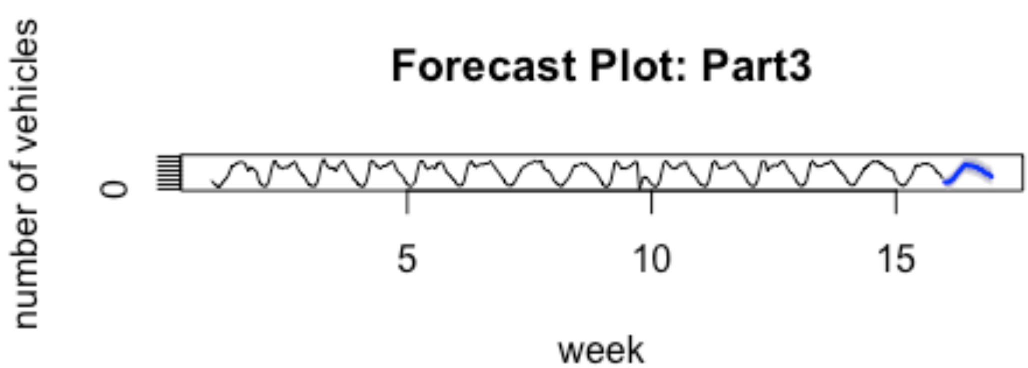
```
#forecast for hour 8:00, 9:00, 17:00, 18:00 on July 1
hour <- c(8,9,17,18)
fit3.forecast.July1.hour <- fit3.forecast.July1$mean[hour]
fit3.forecast.July1.hour

## [1] 756.5516 854.0998 933.5026 846.3402
```

Part 4

```
# Sum of Squared Error (SSE) for model in part 2
fit2.forecast.July1 <- forecast(fit2,24)
(fit2.forecast.July1.hour <- fit2.forecast.July1$mean[hour])

## [1] 1205.979 1080.979 1196.979 1125.979

# root mean square eror
(rmse.2 <- sqrt(mean((test_data[hour,3] - fit2.forecast.July1.hour)^2)))

## [1] 33.92703
```

```r
(fit3.forecast.July1.hour <- fit3.forecast.July1$mean[hour])

## [1] 756.5516 854.0998 933.5026 846.3402

(rmse.3 <- sqrt(mean((test_data[hour,3] - fit3.forecast.July1.hour)^2)))

## [1] 322.4344

cbind(rMSE_Part2=rmse.2, rMSE_Part3=rmse.3)

##      rMSE_Part2 rMSE_Part3
## [1,]   33.92703   322.4344

par(mfrow=c(2,1))
#---- Forecast Plot ----#
plot(fit2.forecast.July1, xlab="week", ylab="number of vehicles",main="Forecast Plot: Part2")
plot(fit3.forecast.July1, xlab="week", ylab="number of vehicles",main="Forecast Plot: Part3")
```






```r
par(mfrow=c(1,1))
#---- AICc ----#
cbind(AICc.Part2 = 2249.44, AICc.Part3=4382.55)
```

```
##      AICc.Part2 AICc.Part3
## [1,]    2249.44    4382.55
```

```r
#---- BIC ----#
cbind(BIC.Part2=2259.07, BIC.Part3=4409.43)
```

```
##      BIC.Part2 BIC.Part3
## [1,]   2259.07   4409.43
```

As we can see, the day of the week model in Part 2 has both lower sum of squared error and root mean squared error, thus doing a beeter job than the hour of the day model in Part 3. In addition, both AICc and BIC from Part 2 are lower, thus model in Part 2 is better. Also, from the forecast plot, we could see that the forcast of vehicle numbers from ARIMA model in Part 2 is closer to the actual vehicle numbers in July 1. Even the prediction interval in Part 2 is narrower. All those evidences suggests that the model from Part 2 might be better.

Part 5

```r
# Holt-Winters exponential smoothing with trend and additive seasonal component.
fit4 <- HoltWinters(ts(train_data[,3], frequency=168), seasonal = "additive")
fit4
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts(train_data[, 3], frequency = 168), seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.05902247
##  beta : 0
##  gamma: 0.4146915
##
## Coefficients:
##               [,1]
## a     759.22105251
## b      -0.04478252
## s1   -524.19340468
## s2   -610.40219795
## s3   -609.70920212
## s4   -574.90255042
## s5   -398.97811828
## s6     23.88813111
## s7    373.72720465
## s8    453.47888450
## s9    328.24404531
## s10   146.98606348
## s11   156.59419470
## s12   145.50704042
```

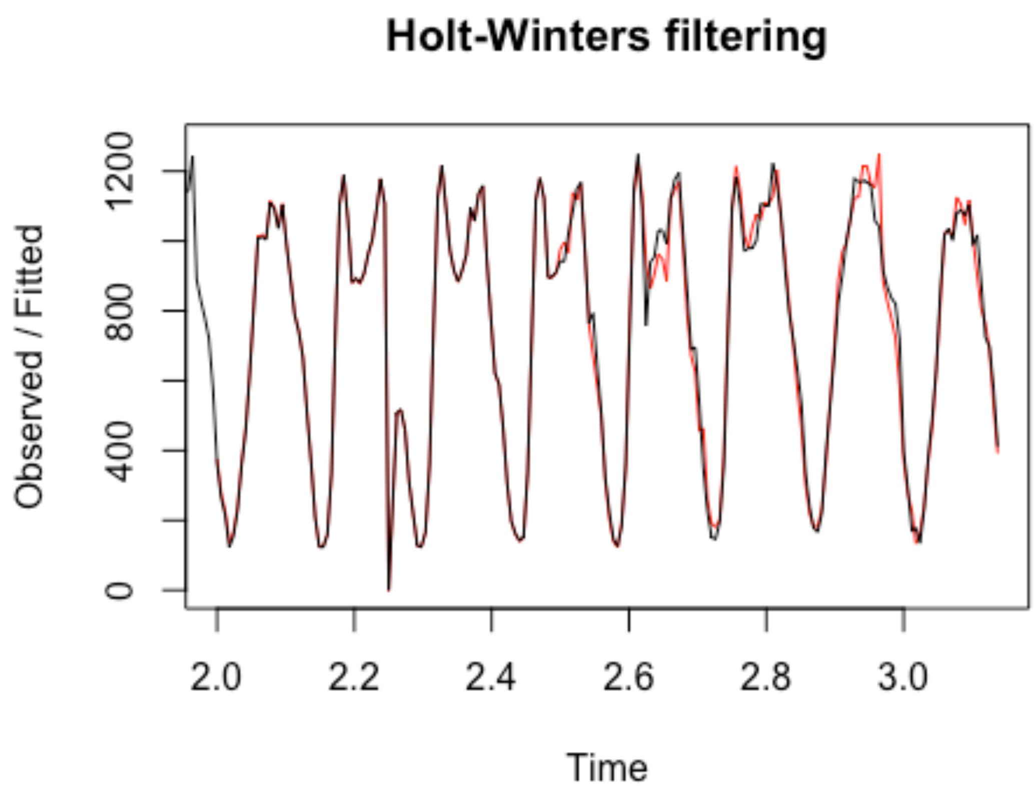
```
## s13   173.62846639
## s14   229.63787765
## s15   269.66277486
## s16   351.63995441
## s17   443.62175922
## s18   372.66634235
## s19  -735.36740099
## s20  -482.42649270
## s21  -227.39937613
## s22  -218.33533795
## s23  -276.32940605
## s24  -426.29338160
## s25  -518.11734004
## s26  -607.11016169
## s27  -611.21902950
## s28  -569.24144529
## s29  -375.44776868
## s30    27.39648621
## s31   390.37199324
## s32   480.33536375
## s33   362.35867784
## s34   237.30626686
## s35   179.11878693
## s36   147.81661885
## s37   181.53754923
## s38   225.32693269
## s39   358.18266227
## s40   322.10041355
## s41   399.03548829
## s42   421.90496258
## s43   213.76609836
## s44    43.71668627
## s45  -114.23432578
## s46  -147.09976903
## s47  -272.07115219
## s48  -432.14965948
## s49  -539.31609069
## s50  -575.56824361
## s51  -595.65670144
## s52  -585.58731746
## s53  -399.57190551
## s54   -11.45518623
## s55   381.04764621
## s56   443.37339822
## s57   386.27744577
## s58   159.11836656
## s59   160.84163506
## s60   174.39810915
## s61   226.35000290
## s62   241.94482325
```

```
## s63   258.00961057
## s64   382.87498824
## s65   399.30595696
## s66   431.42102884
## s67   242.33558791
## s68    29.67424888
## s69   -12.76793264
## s70  -131.63395498
## s71  -240.53485784
## s72  -428.73807042
## s73  -534.44688928
## s74  -603.32427864
## s75  -622.65391084
## s76  -557.26341274
## s77  -392.17086715
## s78    33.90516521
## s79   396.55785172
## s80   491.20666408
## s81   374.25578534
## s82   121.44975868
## s83   159.40771920
## s84   186.75432481
## s85   248.25187804
## s86   235.73679694
## s87   177.80763018
## s88   365.35980467
## s89   402.70372341
## s90   421.05020093
## s91   246.65848046
## s92    57.73875267
## s93   -86.66122242
## s94  -113.89292829
## s95  -282.49342391
## s96  -356.40194479
## s97  -523.42589726
## s98  -591.15540391
## s99  -596.55057414
## s100 -562.66645863
## s101 -387.54930394
## s102  -14.09361530
## s103  332.49581415
## s104  444.38397808
## s105  370.96659780
## s106  247.18364069
## s107  229.76735786
## s108  266.13077217
## s109  301.36668289
## s110  336.89179101
## s111  361.98972802
## s112  359.59358732
```

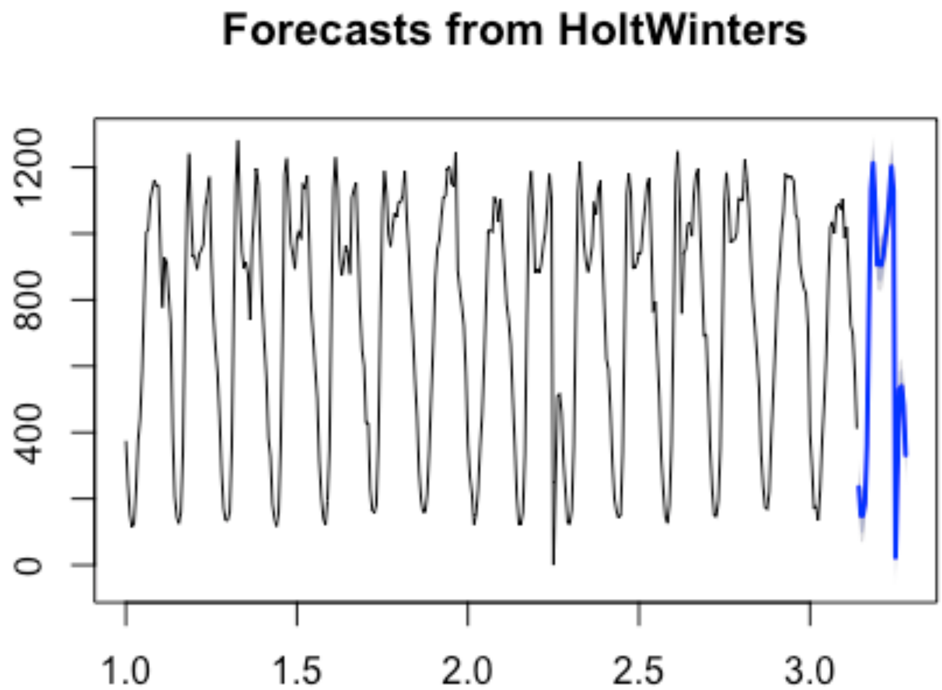
```
## s113  420.76116100
## s114  438.13717671
## s115  317.48054963
## s116  191.77851196
## s117   60.35695593
## s118  -34.15784900
## s119 -148.74126112
## s120 -256.36785862
## s121 -437.61749892
## s122 -528.51705315
## s123 -576.12917525
## s124 -579.29696383
## s125 -524.22557947
## s126 -376.94908172
## s127 -220.72067683
## s128  -72.61973310
## s129  108.67748560
## s130  194.44169922
## s131  252.09735123
## s132  320.29987656
## s133  400.59215714
## s134  394.56556110
## s135  446.55348350
## s136  449.57736251
## s137  415.50918547
## s138  371.76020798
## s139  429.44616103
## s140  168.63306120
## s141  114.09373711
## s142   70.69120345
## s143   26.97802316
## s144 -102.00920045
## s145 -363.76420085
## s146 -465.72923361
## s147 -545.34317653
## s148 -595.74827692
## s149 -594.79675124
## s150 -512.27196551
## s151 -374.67126311
## s152 -270.29743150
## s153 -114.40543859
## s154   90.40729901
## s155  270.23950718
## s156  279.91319728
## s157  262.24284640
## s158  357.81314332
## s159  351.08492124
## s160  311.95415669
## s161  364.38143947
## s162  248.97785245
```

```
## s163  201.33534602
## s164   81.65951223
## s165   -8.68579722
## s166  -63.16155892
## s167 -205.37919870
## s168 -358.66816058
```

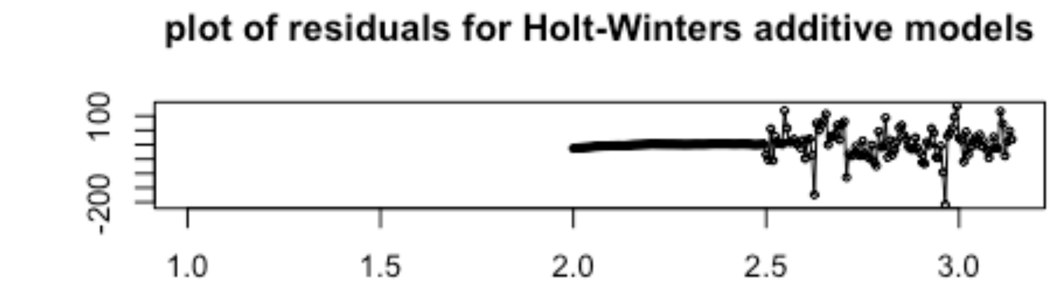
```r
plot(fit4)
```

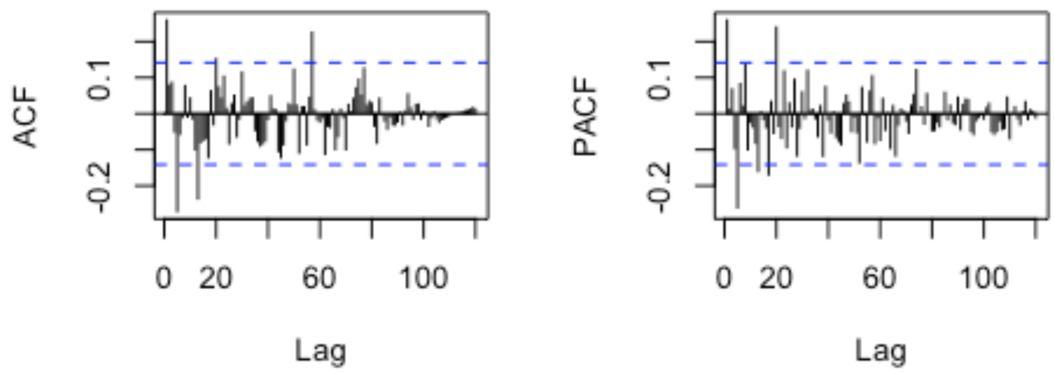



```r
fit4.forecast.July1 <- forecast(fit4, h=24)
plot(fit4.forecast.July1)
```




```
fit4.predict.July1 <- as.data.frame(fit4.forecast.July1)[,1]
(rmse.4 <- sqrt(mean((test_data[,3] - fit4.predict.July1)^2)))

## [1] 220.4259

tsdisplay(fit4.forecast.July1$residuals, main = "plot of residuals for Holt-
Winters additive models" )
```






```
Box.test(fit4.forecast.July1$residuals, lag=20, type="Ljung-Box")

##
##  Box-Ljung test
##
## data:  fit4.forecast.July1$residuals
## X-squared = 59.333, df = 20, p-value = 9.029e-06

data <- ts(train_data[,3], frequency=168)
data <- data[data!=0]
#fit5 <- HoltWinters(data, seasonal = "multiplicative")
#fit5
```

It seem that our data is not suitable to fit a Holt-Winters multiplicative as the seasonal variation is clearly not multiplicative, and it also shows the error message that "time series has no or less than 2 periods", so I didn't build a multiplicative model in this case.

```
cbind(rMSE_Part2=rmse.arima, rMSE_Part5=rmse.4)

##      rMSE_Part2 rMSE_Part5
## [1,]   221.8351   220.4259
```

Based on the root mean square error, we could see that the Holt-Winters additive seasonality model is slightly better than ARIMA model in part2. However, the correlogram shows that the autocorrelations for the in-sample forecast errors exceed the significance bounds for lags 1-20. Furthermore, the p-value for Ljung-Box test is small, indicating that there is strong evidence of non-zero autocorrelations at lags 1-20, hence the Holt winters model still have room to improve.