ADS 506 Final Project: Time Sereis Analysis of U.S. Road Traffic Injuries from 2002-2010

Team 3: Anusia Edward & Harini Lakshmanan

GitHub Link: https://github.com/edwardam5/506_Final_Project.git

Appendix A

Preprocessing/EDA

```
#loading in the dataset
traffic_1 = read.csv("~/Desktop/road-traffic-injuries-02-10.csv")
```

```
#skim(traffic_1)
```

```
# dimensionality reduction
keep <- c("reportyear", "county_name", "region_name", "mode",
          "totalpop", "poprate", "severity","injuries")
traffic_1 = traffic_1[keep]
head(traffic_1)
```

```
##   reportyear county_name         region_name       mode totalpop poprate
## 1       2002      Orange Southern California  All modes  2914663    6.62
## 2       2002      Orange Southern California  All modes  2914663   24.74
## 3       2002      Orange Southern California  Bicyclist  2914663    0.24
## 4       2002      Orange Southern California  Bicyclist  2914663    1.68
## 5       2002      Orange Southern California        Bus  2914663    0.03
## 6       2002      Orange Southern California Car/Pickup  2914663    4.46
##        severity injuries
## 1        Killed      193
## 2 Severe Injury      721
## 3        Killed        7
## 4 Severe Injury       49
## 5 Severe Injury        1
## 6        Killed      130
```

```
# removing redundant rows
traffic_1 <- traffic_1[!is.na(traffic_1$totalpop), ]
head(traffic_1)
```

```
##   reportyear county_name         region_name       mode totalpop poprate
```

```
## 1       2002      Orange Southern California  All modes  2914663    6.62
## 2       2002      Orange Southern California  All modes  2914663   24.74
## 3       2002      Orange Southern California  Bicyclist  2914663    0.24
## 4       2002      Orange Southern California  Bicyclist  2914663    1.68
## 5       2002      Orange Southern California        Bus  2914663    0.03
## 6       2002      Orange Southern California Car/Pickup  2914663    4.46
##        severity injuries
## 1        Killed      193
## 2 Severe Injury      721
## 3        Killed        7
## 4 Severe Injury       49
## 5 Severe Injury        1
## 6        Killed      130
```

```
# summary/descriptive stats of data
summary(traffic_1)
```

```
##    reportyear   county_name        region_name            mode
##  Min.   :2002   Length:37828       Length:37828       Length:37828
##  1st Qu.:2004   Class :character   Class :character   Class :character
##  Median :2006   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2006
##  3rd Qu.:2008
##  Max.   :2010
##
##     totalpop           poprate           severity            injuries
##  Min.   :       0   Min.   :    0.01   Length:37828       Min.   :    1.0
##  1st Qu.:   26218   1st Qu.:    2.96   Class :character   1st Qu.:    1.0
##  Median :   61773   Median :    7.34   Mode  :character   Median :    4.0
##  Mean   :  516464   Mean   :   24.77                      Mean   :   41.2
##  3rd Qu.:  145438   3rd Qu.:   18.65                      3rd Qu.:   12.0
##  Max.   :37253956   Max.   :10679.61                      Max.   :13578.0
##                     NA's   :573                           NA's   :469
```

```
# formatting adjustments
traffic_2 <- traffic_1
traffic_2$county_name <- sub(" ", "_", traffic_1$county_name)
traffic_2$severity <- sub(" ", "_", traffic_1$severity)
traffic_2$mode <- sub(" ", "_", traffic_1$mode)
traffic_2$mode <- sub("/", "_", traffic_1$mode)
traffic_2$region_name <- sub("/", "_", traffic_1$region_name)
traffic_2$region_name <- sub("/", "_", traffic_1$region_name)
```

```
# focusing the study on southern California
traffic_2s <- subset(traffic_2,
                          region_name == 'Southern California')
```

```
# dropping region category as its all soCAL
traffic_2s <- subset(traffic_2s, select = -(region_name))
```

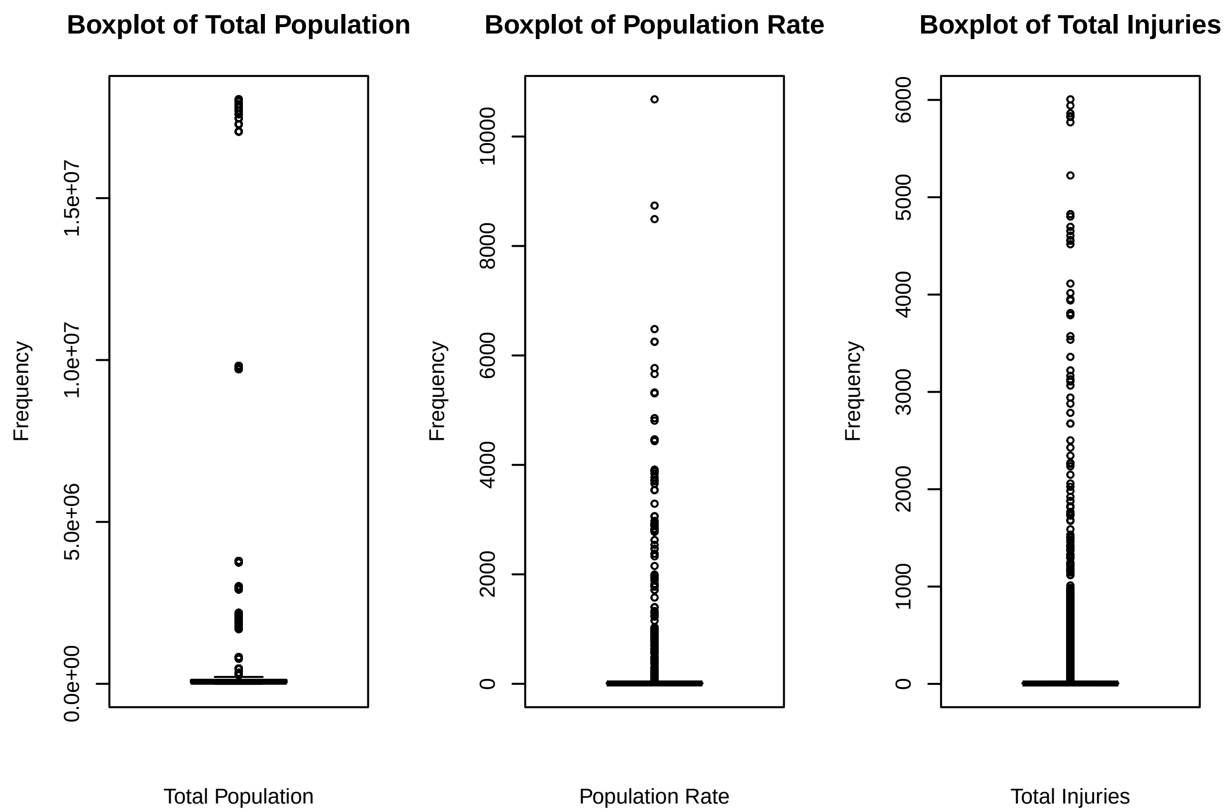
```r
# filling missing variables for poprate and injuries using knn imputation
traffic_fill <- kNN(traffic_2s, variable = c("injuries", "poprate"), k=5)
```

```r
# double checking that all NAs were handled
summary(traffic_fill)
```

```
##   reportyear   county_name            mode              totalpop
##  Min.   :2002   Length:14606       Length:14606       Min.   :       0
##  1st Qu.:2004   Class :character   Class :character   1st Qu.:   34119
##  Median :2006   Mode  :character   Mode  :character   Median :   63166
##  Mean   :2006                                         Mean   :  444042
##  3rd Qu.:2008                                         3rd Qu.:  117275
##  Max.   :2010                                         Max.   :18051534
##     poprate            severity             injuries      injuries_imp
##  Min.   :    0.01   Length:14606       Min.   :   1.00   Mode :logical
##  1st Qu.:    2.79   Class :character   1st Qu.:   2.00   FALSE:14520
##  Median :    6.35   Mode  :character   Median :   4.00   TRUE :86
##  Mean   :   35.54                      Mean   :  33.84
##  3rd Qu.:   16.23                      3rd Qu.:  10.00
##  Max.   :10679.61                      Max.   :6006.00
##  poprate_imp
##  Mode :logical
##  FALSE:14425
##  TRUE :181
##
##
##
```

```r
# removing poprate_imp and injuries_imp
traffic_fill <- subset(traffic_fill, select = reportyear:injuries)
```

```r
# boxplots of numeric variables to check for outliers:totalpop,poprate,injuries
par(mfrow=c(1,3))
boxplot(traffic_fill$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Boxplot of Total Population")
boxplot(traffic_fill$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Boxplot of Population Rate")
boxplot(traffic_fill$injuries, xlab="Total Injuries", ylab = "Frequency",
     main = "Boxplot of Total Injuries")
```

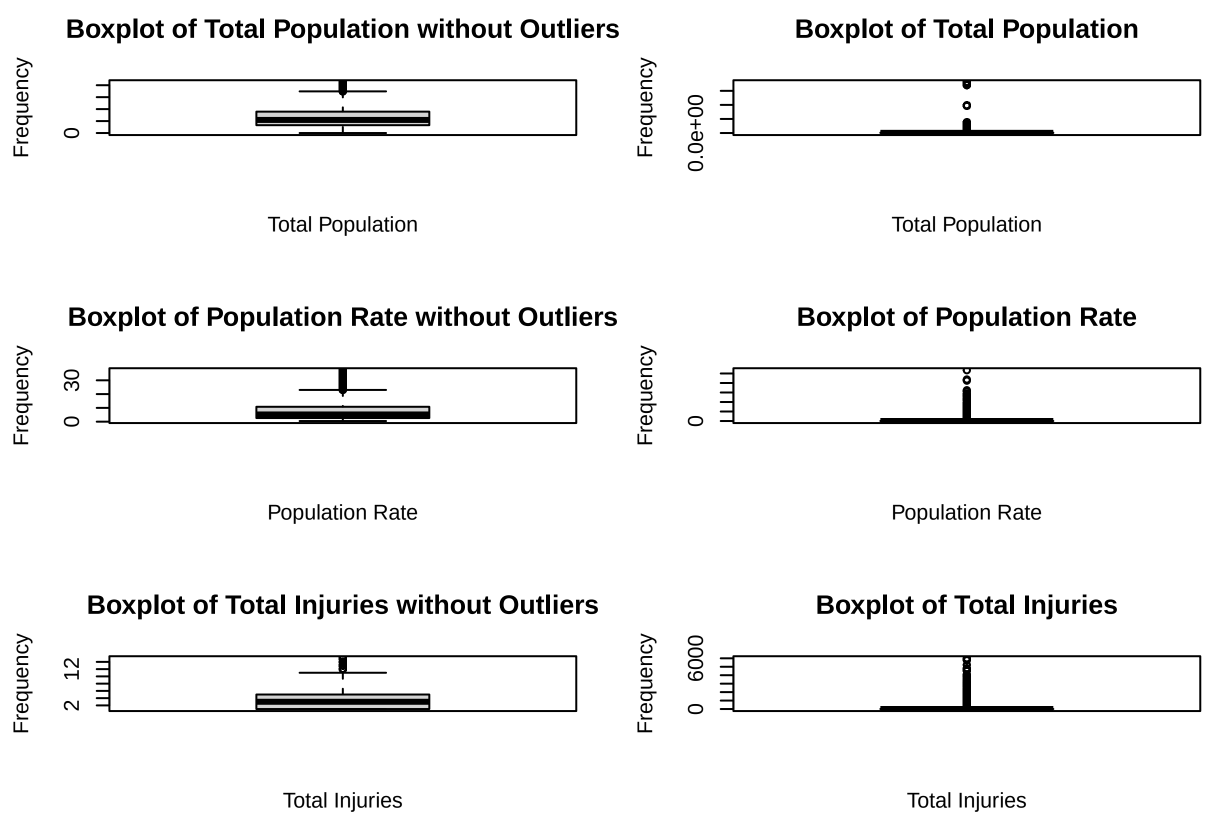



```r
# handling outliers using IQR for totalpop
Q1_totalpop <- quantile(traffic_fill$totalpop, .25)
Q3_totalpop <- quantile(traffic_fill$totalpop, .75)
IQR_totalpop <- IQR(traffic_fill$totalpop)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
traffic_out12 <- subset(traffic_fill, traffic_fill$totalpop> (Q1_totalpop- 1.5*IQR_totalpop) & traffic_
```

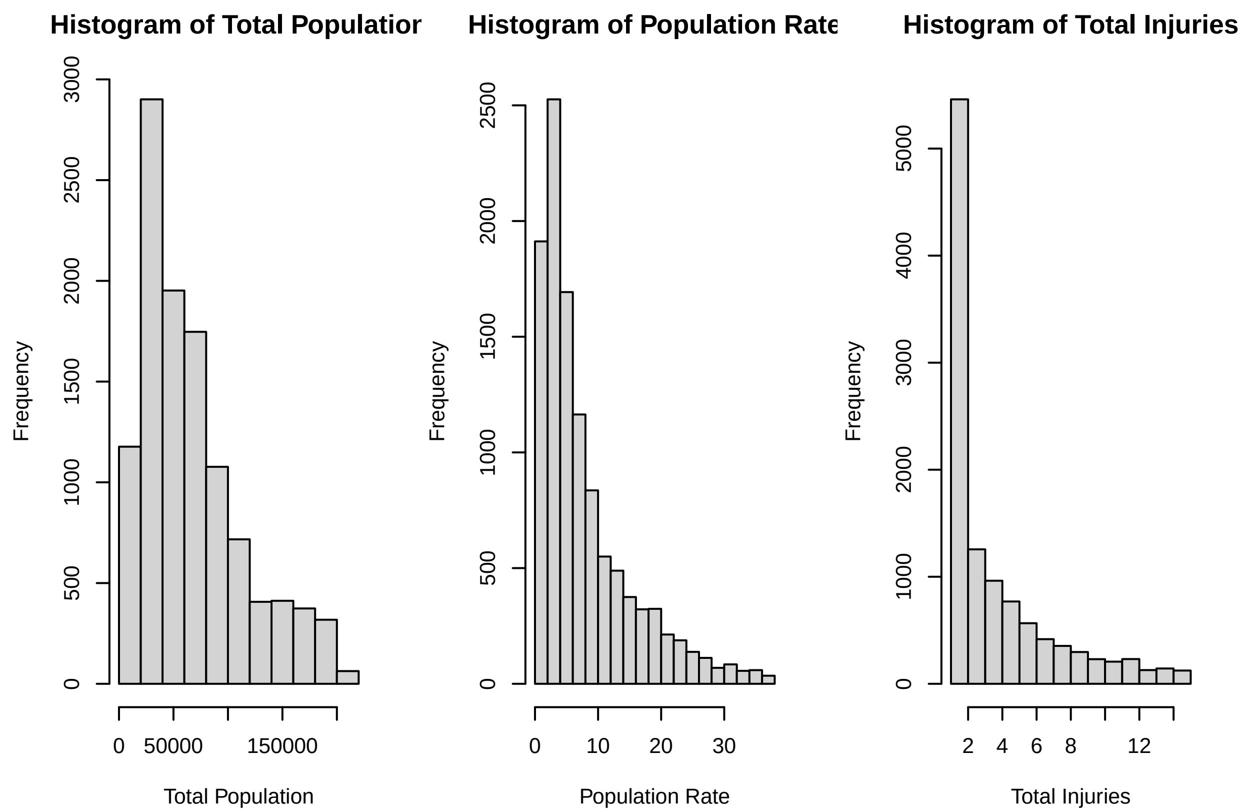
```r
# handling outliers using IQR for poprate injuries
Q1_poprate <- quantile(traffic_out12$poprate, .25)
Q3_poprate <- quantile(traffic_out12$poprate, .75)
IQR_poprate <- IQR(traffic_out12$poprate)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
traffic_out1 <- subset(traffic_out12, traffic_out12$poprate> (Q1_poprate- 1.5*IQR_poprate) & traffic_ou
```

```r
# handling outliers using IQR for injuries
Q1_injuries <- quantile(traffic_out1$injuries, .25)
Q3_injuries <- quantile(traffic_out1$injuries, .75)
IQR_injuries <- IQR(traffic_out1$injuries)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
traffic_out <- subset(traffic_out1, traffic_out1$injuries> (Q1_injuries- 1.5*IQR_injuries) & traffic_ou
```

```r
# boxplots after removal of outliers
par(mfrow=c(3,2))
boxplot(traffic_out$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Boxplot of Total Population without Outliers")
boxplot(traffic_fill$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Boxplot of Total Population")
boxplot(traffic_out$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Boxplot of Population Rate without Outliers")
boxplot(traffic_fill$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Boxplot of Population Rate")
boxplot(traffic_out$injuries, xlab="Total Injuries", ylab = "Frequency",
     main = "Boxplot of Total Injuries without Outliers")
boxplot(traffic_fill$injuries, xlab="Total Injuries", ylab = "Frequency",
     main = "Boxplot of Total Injuries")
```



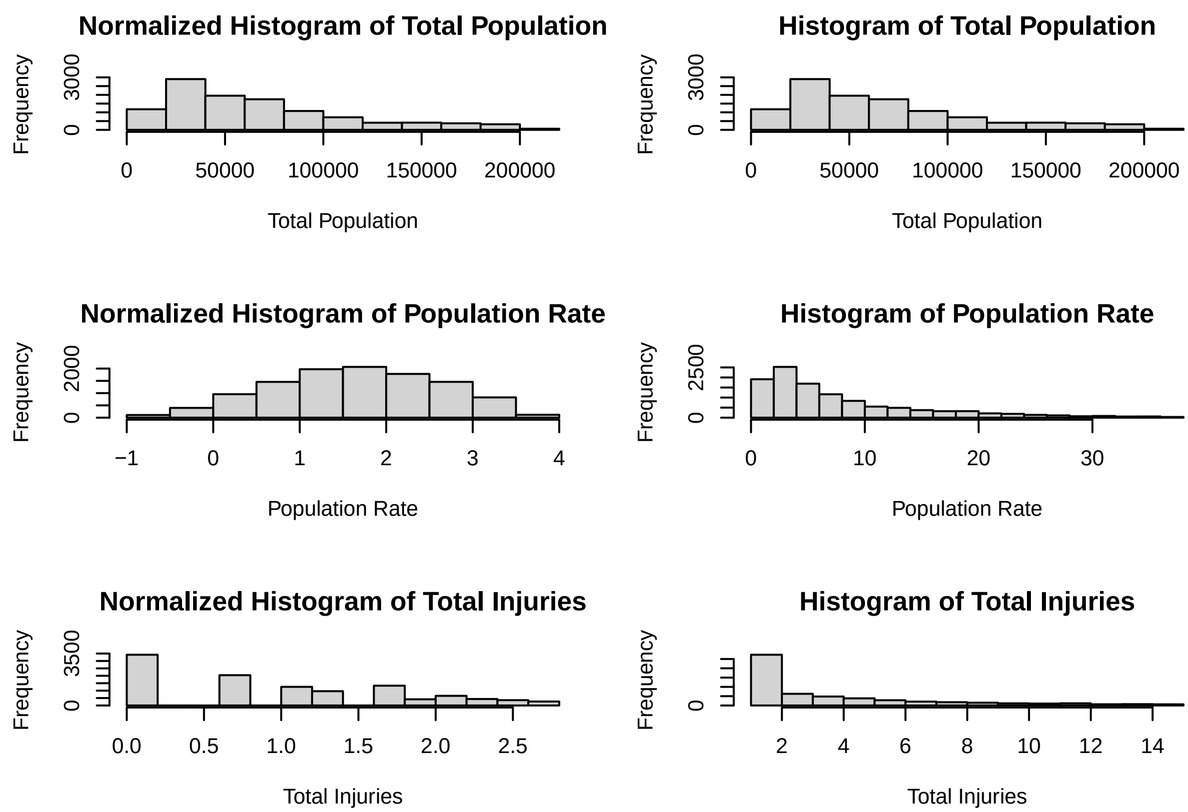

```r
# histograms of numeric variables to check for skewness
par(mfrow=c(1,3))
hist(traffic_out$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Histogram of Total Population")
hist(traffic_out$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Histogram of Population Rate")
hist(traffic_out$injuries, xlab="Total Injuries", ylab = "Frequency",
     main = "Histogram of Total Injuries")
```




```r
# handling skewness using box-cox transformation
traffic_norm <- traffic_out

bct_totalpop <- BoxCoxTrans(traffic_norm$totalpop)
hcv_totalpop <- predict(bct_totalpop, traffic_norm$totalpop)
traffic_norm$totalpop <- hcv_totalpop

bct_poprate <- BoxCoxTrans(traffic_norm$poprate)
hcv_poprate <- predict(bct_poprate, traffic_norm$poprate)
traffic_norm$poprate <- hcv_poprate

bct_injuries <- BoxCoxTrans(traffic_norm$injuries)
hcv_injuries<- predict(bct_injuries, traffic_norm$injuries)
traffic_norm$injuries <- hcv_injuries
```

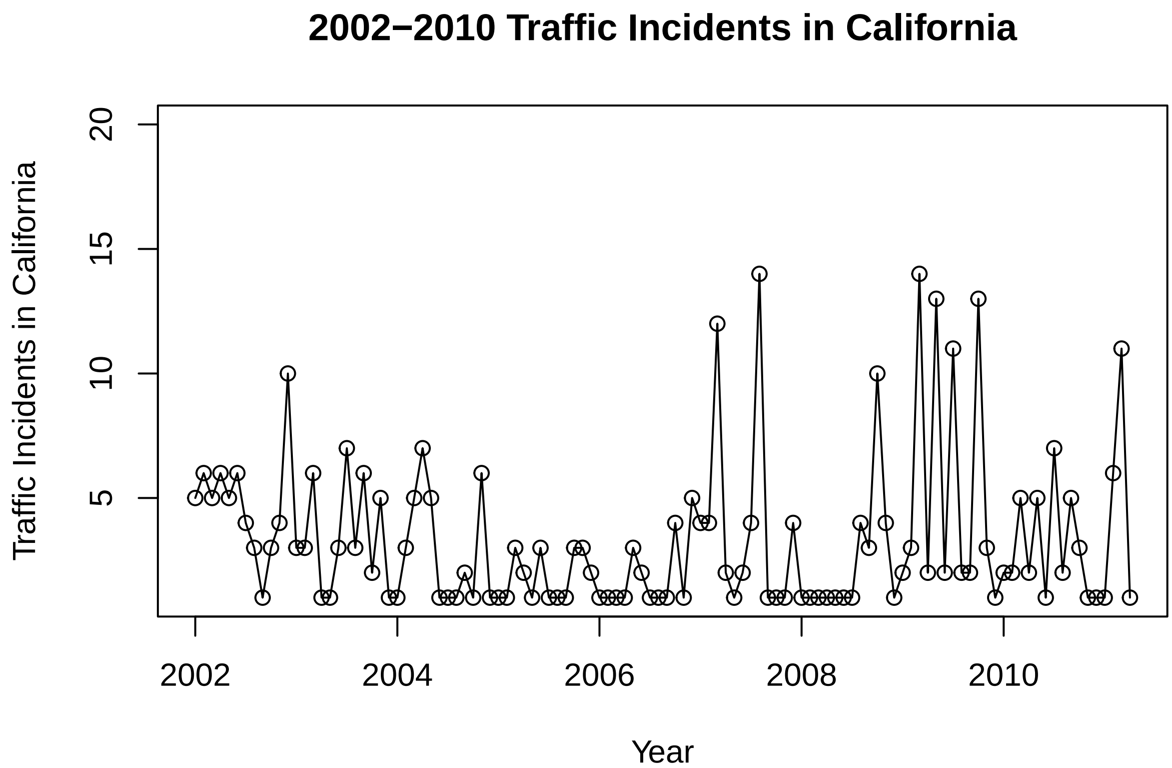
```r
# histograms after normalizations
par(mfrow=c(3,2))
hist(traffic_norm$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Normalized Histogram of Total Population")
hist(traffic_out$totalpop, xlab="Total Population", ylab = "Frequency",
     main = "Histogram of Total Population")
hist(traffic_norm$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Normalized Histogram of Population Rate")
hist(traffic_out$poprate, xlab="Population Rate", ylab = "Frequency",
     main = "Histogram of Population Rate")
hist(traffic_norm$injuries, xlab="Total Injuries", ylab = "Frequency",
```

```
     main = "Normalized Histogram of Total Injuries")
hist(traffic_out$injuries, xlab="Total Injuries", ylab = "Frequency",
     main = "Histogram of Total Injuries")
```

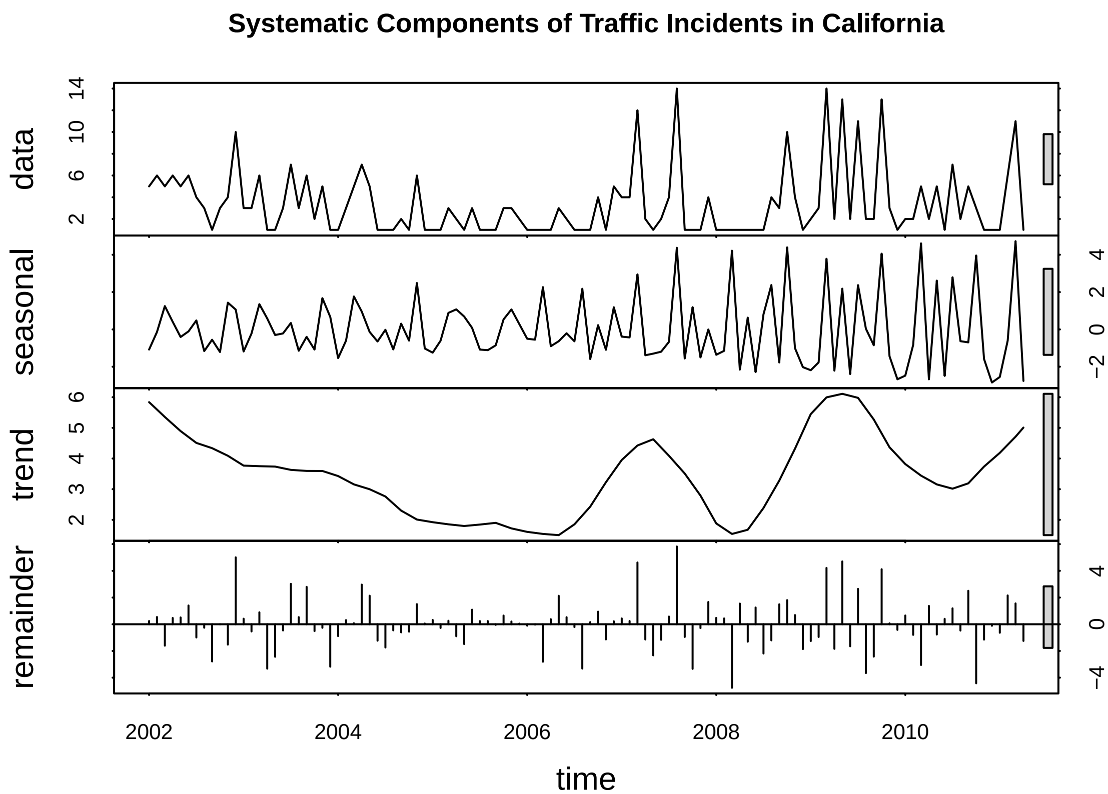



```
# dummy coding categorical variables:county_name, severity,and mode
traffic_dum <- dummy_cols(traffic_norm, select_columns = c('county_name',
                                                         'mode', 'severity'),
           remove_selected_columns = TRUE)
traffic_dum <- subset(traffic_dum, select = -c(mode_ND, severity_ND))
```

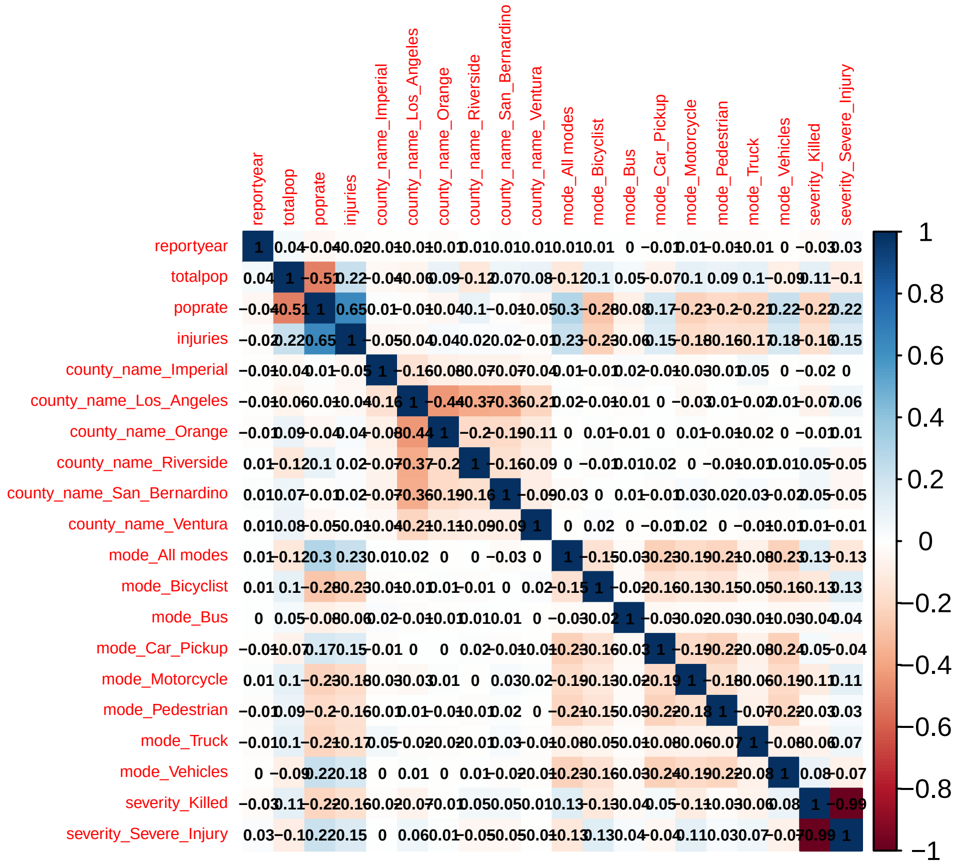
```
# time plots
# creating time plots
traffic.ts <- ts(traffic_out$injuries, start = c(2002, 1),
                         end =c(2011, 4), frequency = 12)
plot(traffic.ts, type = "o", xlab = "Year",
     ylab = "Traffic Incidents in California", ylim = c(1, 20),
     main = "2002-2010 Traffic Incidents in California")
```




```
traffic_decomp <- stl(traffic.ts, s.window = 5,  t.window=15)
plot(traffic_decomp,
     main = "Systematic Components of Traffic Incidents in California")
```




```
# correlation plots
corrplot(cor(traffic_dum), method="color", addCoef.col = 1, number.cex = 0.5,
         tl.cex = 0.5)
```




```r
# splitting into train and test
nValid_f <- 9
nTrain_f <- length(traffic.ts) - nValid_f
train_f.ts <- window(traffic.ts, start = c(2002, 1), end = c(2002, nTrain_f))
valid_f.ts <- window(traffic.ts, start = c(2002, nTrain_f + 1),
                     end = c(2002, nTrain_f + nValid_f))
```

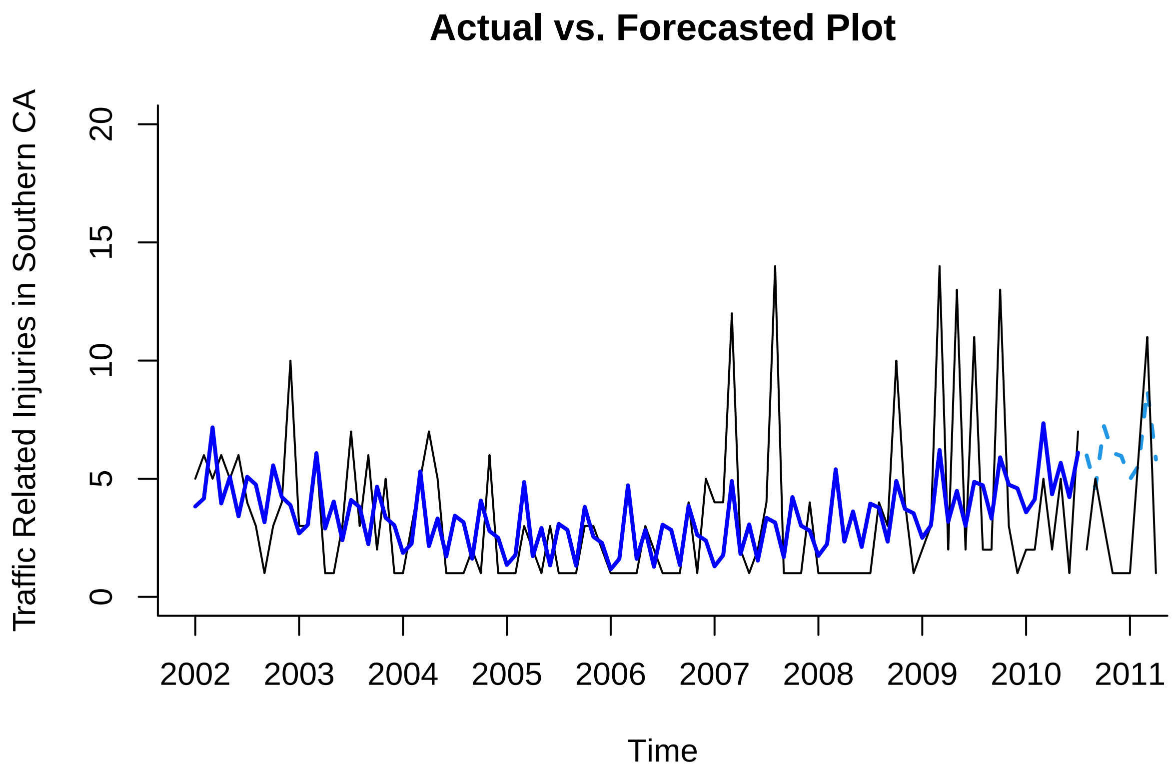
Models + Model Evaluations

```r
# Linear Regression Model
train_f.lm.trend.season <- tslm(train_f.ts ~ trend + I(trend^2) + season)
summary(train_f.lm.trend.season)
```

```
##
## Call:
## tslm(formula = train_f.ts ~ trend + I(trend^2) + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3987 -1.9034 -0.6092  0.6928 10.8576
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  3.9357731  1.2817705   3.071  0.00283 **
## trend       -0.1105296  0.0401694  -2.752  0.00719 **
## I(trend^2)   0.0011016  0.0003744   2.943  0.00415 **
## season2      0.4459179  1.4044455   0.318  0.75160
## season3      3.5562994  1.4045519   2.532  0.01310 *
## season4      0.4422555  1.4047269   0.315  0.75362
## season5      1.6593417  1.4049698   1.181  0.24073
## season6      0.0964470  1.4052812   0.069  0.94544
## season7      1.8646825  1.4056628   1.327  0.18805
## season8      1.6289730  1.4497785   1.124  0.26420
## season9      0.1282433  1.4499408   0.088  0.92972
## season10     2.6253105  1.4501300   1.810  0.07361 .
## season11     1.3701745  1.4503456   0.945  0.34736
## season12     1.1128354  1.4505881   0.767  0.44502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.979 on 89 degrees of freedom
## Multiple R-squared:  0.1974, Adjusted R-squared:  0.08018
## F-statistic: 1.684 on 13 and 89 DF,  p-value: 0.07815
```

```r
# predictions
train_f.lm.trend.season.pred <- forecast(train_f.lm.trend.season, h = nValid_f,
                                         level = 0)
# actual vs. forecast plot
plot(train_f.lm.trend.season.pred, ylim = c(0, 20),
     ylab = "Traffic Related Injuries in Southern CA",xlab = "Time", bty = "l",
     xaxt = "n", xlim = c(2002,2011), main = "Actual vs. Forecasted Plot",
     flty = 2)
axis(1, at = seq(2002, 2011, 1), labels = format(seq(2002, 2011, 1)))
lines(train_f.lm.trend.season.pred$fitted, lwd = 2, col = "blue")
lines(valid_f.ts)
```



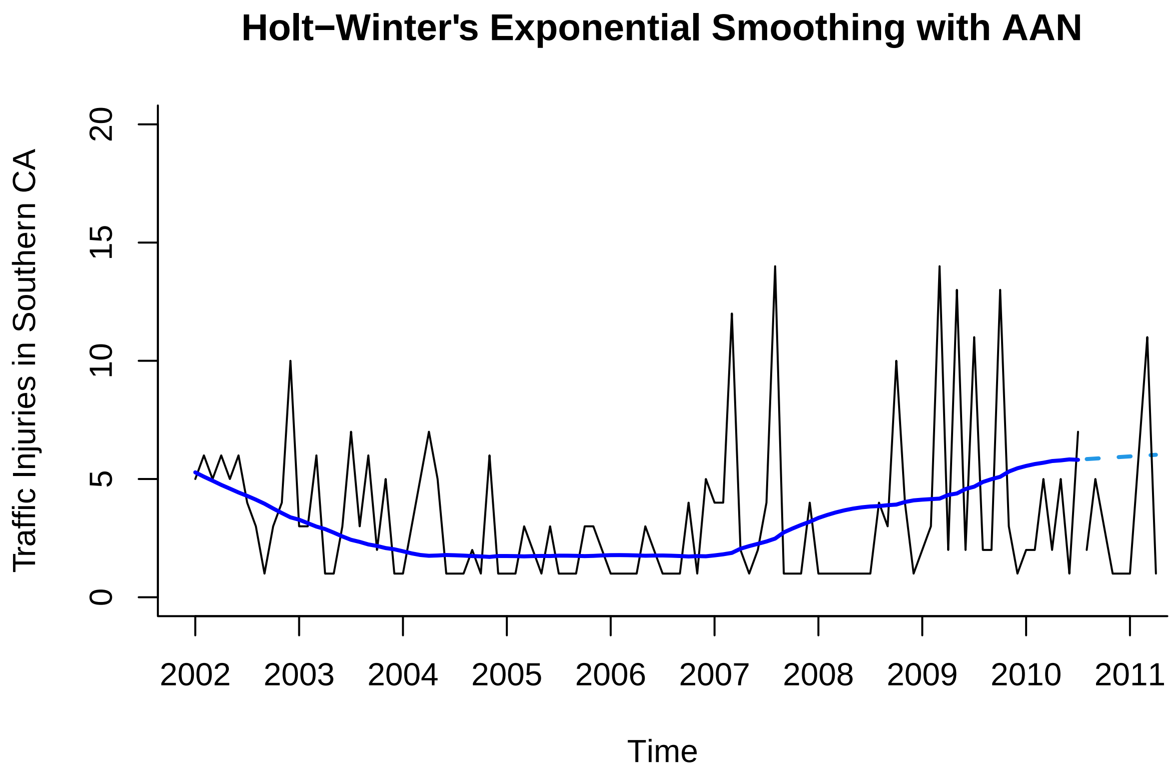

```
accuracy(train_f.lm.trend.season.pred$mean, valid_f.ts)
```

```
##                   ME   RMSE      MAE       MPE   MAPE       ACF1 Theil's U
## Test set -2.666936 3.7779 3.344353 -243.2166 251.09 0.06443753  1.231781
```

```
# Holt-Winter's Exponential Smoothing w/additive error+trend no seasonality
hwin_AANf <- ets(train_f.ts, model = "AAN")
hwin_AANf.pred <- forecast(hwin_AANf, h = nValid_f, level = 0)
accuracy(hwin_AANf.pred$mean, valid_f.ts)
```

```
##                   ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's U
## Test set -2.489284 4.043031 3.604656 -249.1688 259.3437 0.04041304  1.343364
```

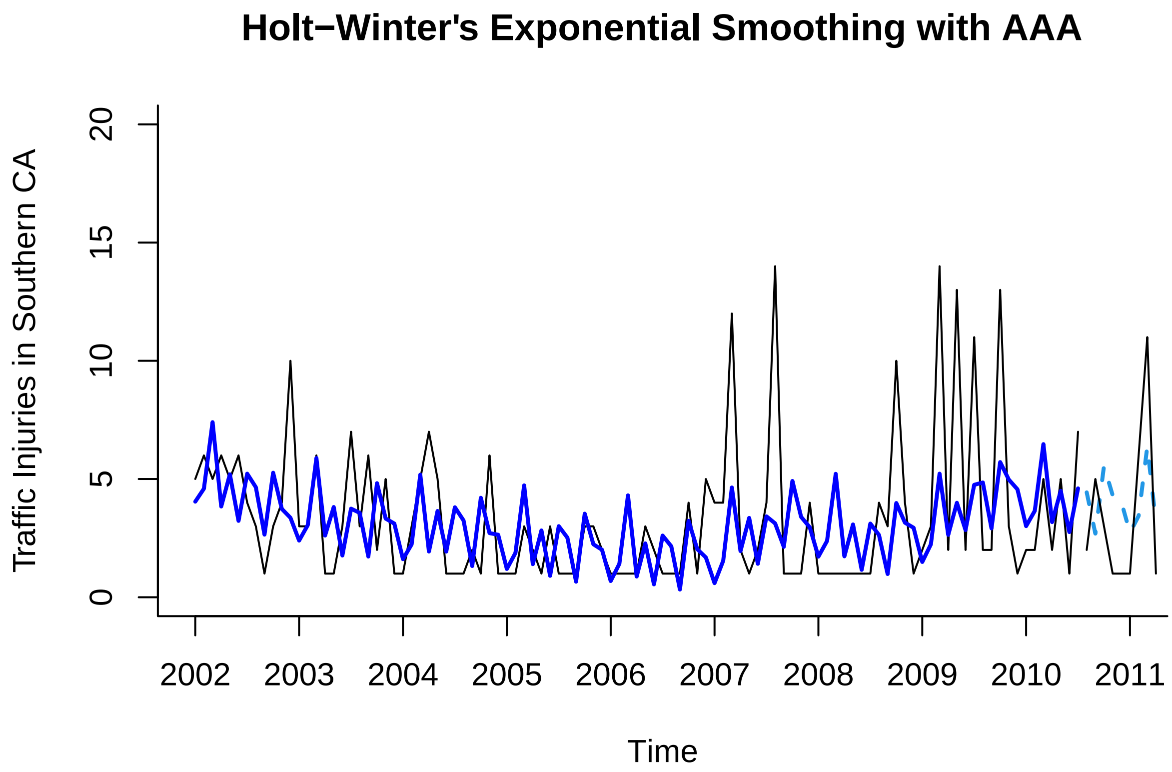
```
plot(hwin_AANf.pred, ylim = c(0, 20),
     ylab = "Traffic Injuries in Southern CA",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(2002,2011),
     main = "Holt-Winter's Exponential Smoothing with AAN",
     flty = 2)
axis(1, at = seq(2002, 2011, 1), labels = format(seq(2002, 2011, 1)))
lines(hwin_AANf.pred$fitted, lwd = 2, col = "blue")
lines(valid_f.ts)
```



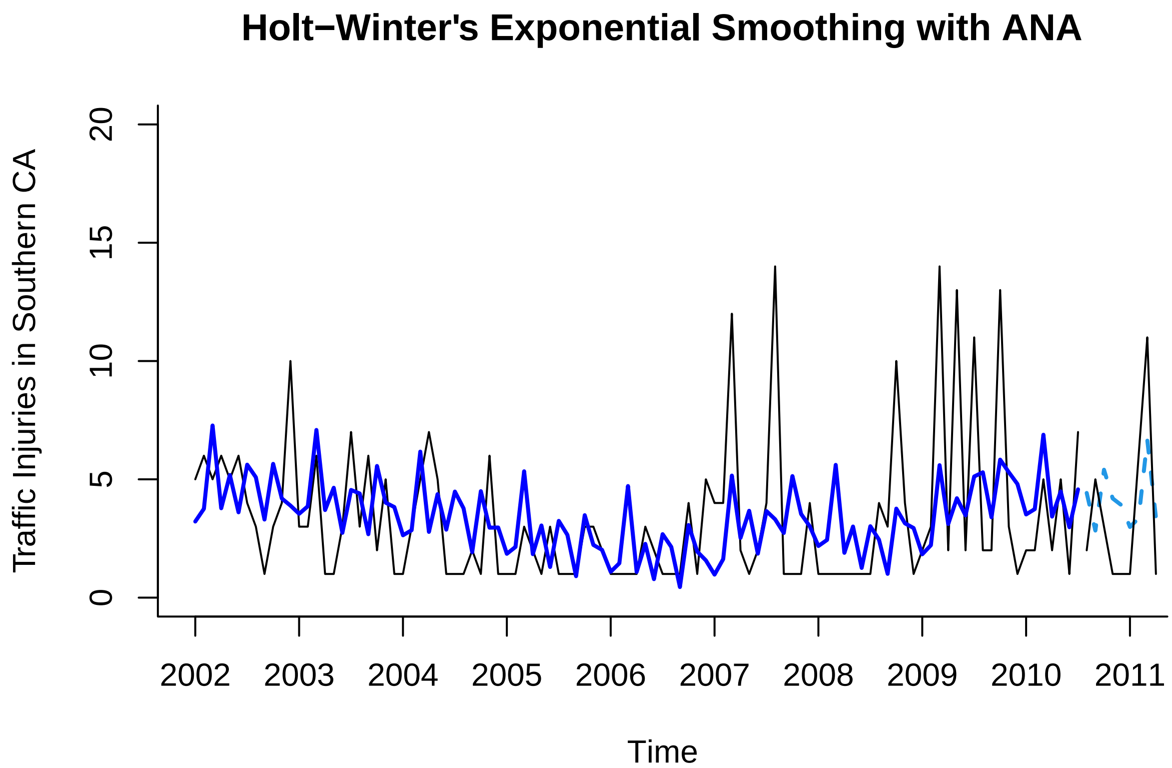

```r
# Holt-Winter's Exponential Smoothing w/additive error, additive trend,
# and additive seasonality
hwin_AAAf <- ets(train_f.ts, model = "AAA")
hwin_AAAf.pred <- forecast(hwin_AAAf, h = nValid_f, level = 0)
accuracy(hwin_AAAf.pred$mean, valid_f.ts)
```

```
##                   ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.6383865 2.860662 2.748507 -122.3321 151.4223 0.1207304 0.8691627
```

```r
plot(hwin_AAAf.pred, ylim = c(0, 20),
     ylab = "Traffic Injuries in Southern CA",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(2002,2011),
     main = "Holt-Winter's Exponential Smoothing with AAA",
     flty = 2)
axis(1, at = seq(2002, 2011, 1), labels = format(seq(2002, 2011, 1)))
lines(hwin_AAAf.pred$fitted, lwd = 2, col = "blue")
lines(valid_f.ts)
```



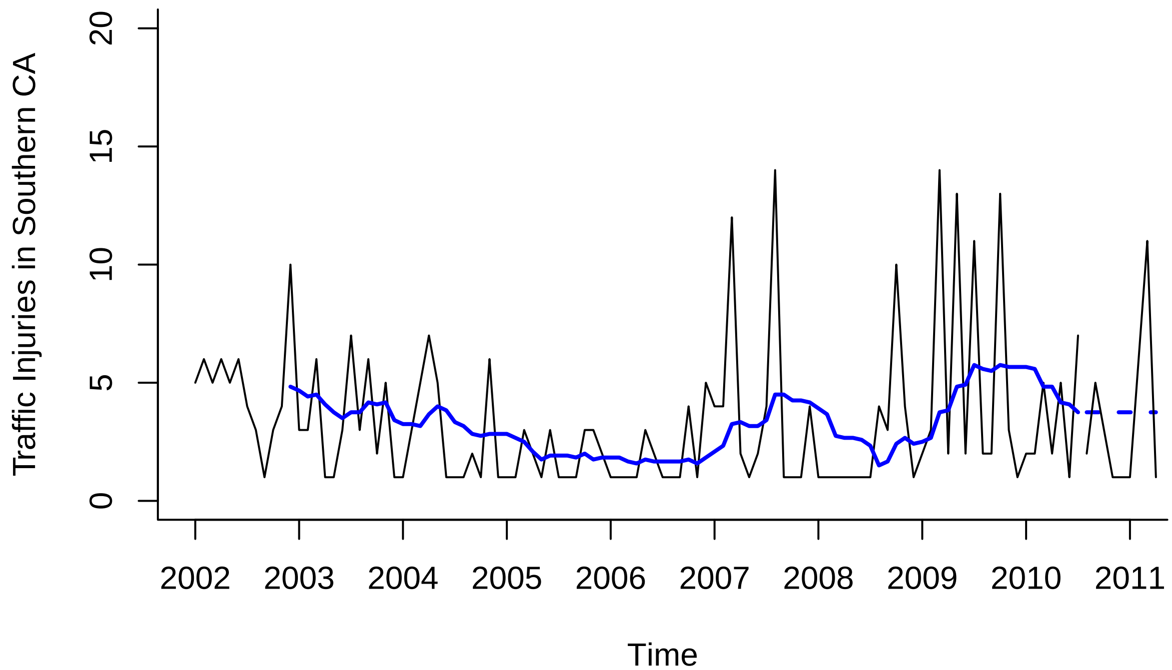

```r
# Holt-Winter's Exponential Smoothing w/additive error, no trend,
# and additive seasonality
hwin_ANAf <- ets(train_f.ts, model = "ANA")
hwin_ANAf.pred <- forecast(hwin_ANAf, h = nValid_f, level = 0)
plot(hwin_ANAf.pred, ylim = c(0, 20),
     ylab = "Traffic Injuries in Southern CA",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(2002,2011),
     main = "Holt-Winter's Exponential Smoothing with ANA",
     flty = 2)
axis(1, at = seq(2002, 2011, 1), labels = format(seq(2002, 2011, 1)))
lines(hwin_ANAf.pred$fitted, lwd = 2, col = "blue")
lines(valid_f.ts)
```




```
accuracy(hwin_ANAf.pred$mean, valid_f.ts)
```

```
##                  ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.683908 2.797637 2.716476 -124.6425 152.8209 0.1026866 0.8755918
```

```
# trailing moving average model
ma_f.trailing <- rollmean(train_f.ts, k = 12, align = "right")
last_f.ma <- tail(ma_f.trailing, 1)
ma_f.trailing.pred <- ts(rep(last_f.ma, nValid_f),
                         start = c(2002, nTrain_f + 1),
                         end = c(2002, nTrain_f + nValid_f), freq = 12)
plot(train_f.ts, ylim = c(0, 20),  ylab = "Traffic Injuries in Southern CA",
     xlab = "Time", bty = "l", xaxt = "n",xlim = c(2002,2011), main = "")
axis(1, at = seq(2002, 2011, 1), labels = format(seq(2002, 2011, 1)))
lines(ma_f.trailing, lwd = 2, col = "blue")
lines(ma_f.trailing.pred, lwd = 2, col = "blue", lty = 2)
lines(valid_f.ts)
```




```r
accuracy(ma_f.trailing.pred, valid_f.ts)
```

```
##                   ME    RMSE      MAE       MPE     MAPE       ACF1 Theil's U
## Test set -0.3055556 3.21563 2.694444 -120.4545 148.9899 0.05087015 0.8841529
```

```r
# Model Evaluations
accuracy(train_f.lm.trend.season.pred$mean, valid_f.ts)
```

```
##                  ME   RMSE      MAE       MPE   MAPE       ACF1 Theil's U
## Test set -2.666936 3.7779 3.344353 -243.2166 251.09 0.06443753  1.231781
```

```r
accuracy(hwin_AANf.pred$mean, valid_f.ts)
```

```
##                  ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's U
## Test set -2.489284 4.043031 3.604656 -249.1688 259.3437 0.04041304  1.343364
```

```r
accuracy(hwin_AAAf.pred$mean, valid_f.ts)
```

```
##                   ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.6383865 2.860662 2.748507 -122.3321 151.4223 0.1207304 0.8691627
```

```
accuracy(hwin_ANAf.pred$mean, valid_f.ts)
```

```
##                  ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.683908 2.797637 2.716476 -124.6425 152.8209 0.1026866 0.8755918
```

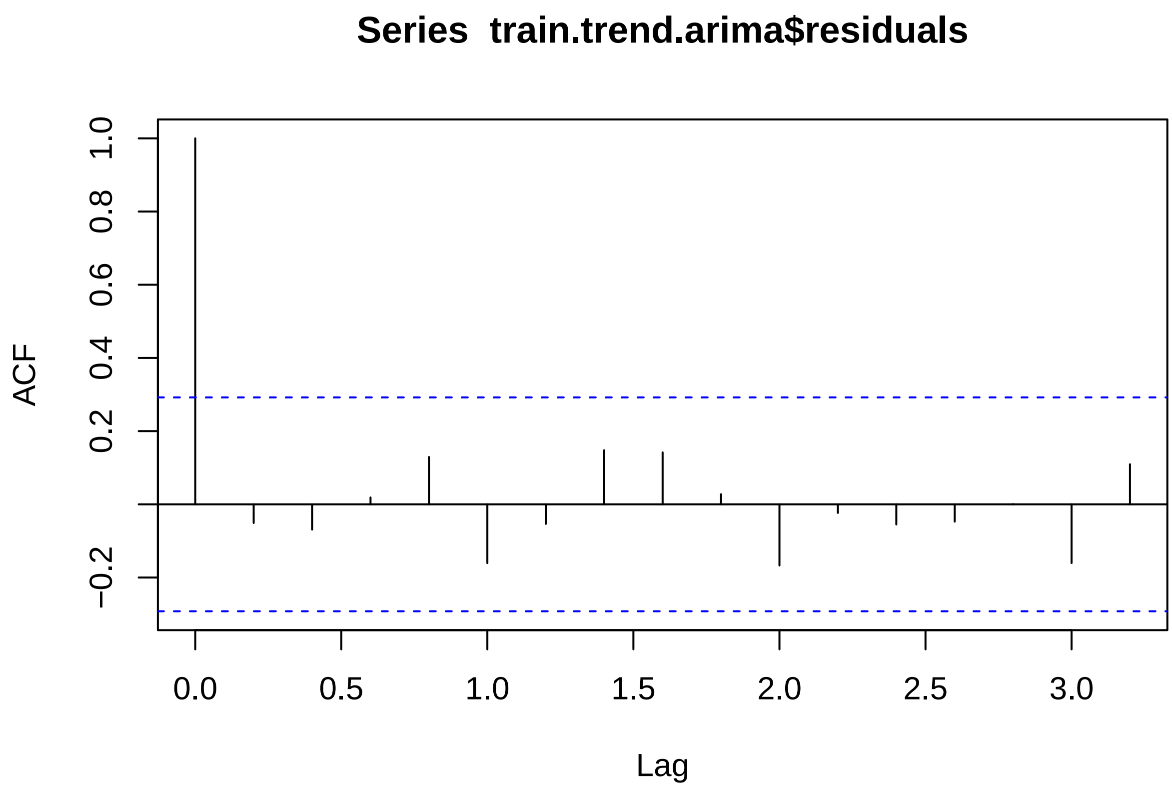
```
accuracy(ma_f.trailing.pred, valid_f.ts)
```

```
##                   ME    RMSE      MAE       MPE     MAPE       ACF1 Theil's U
## Test set -0.3055556 3.21563 2.694444 -120.4545 148.9899 0.05087015 0.8841529
```

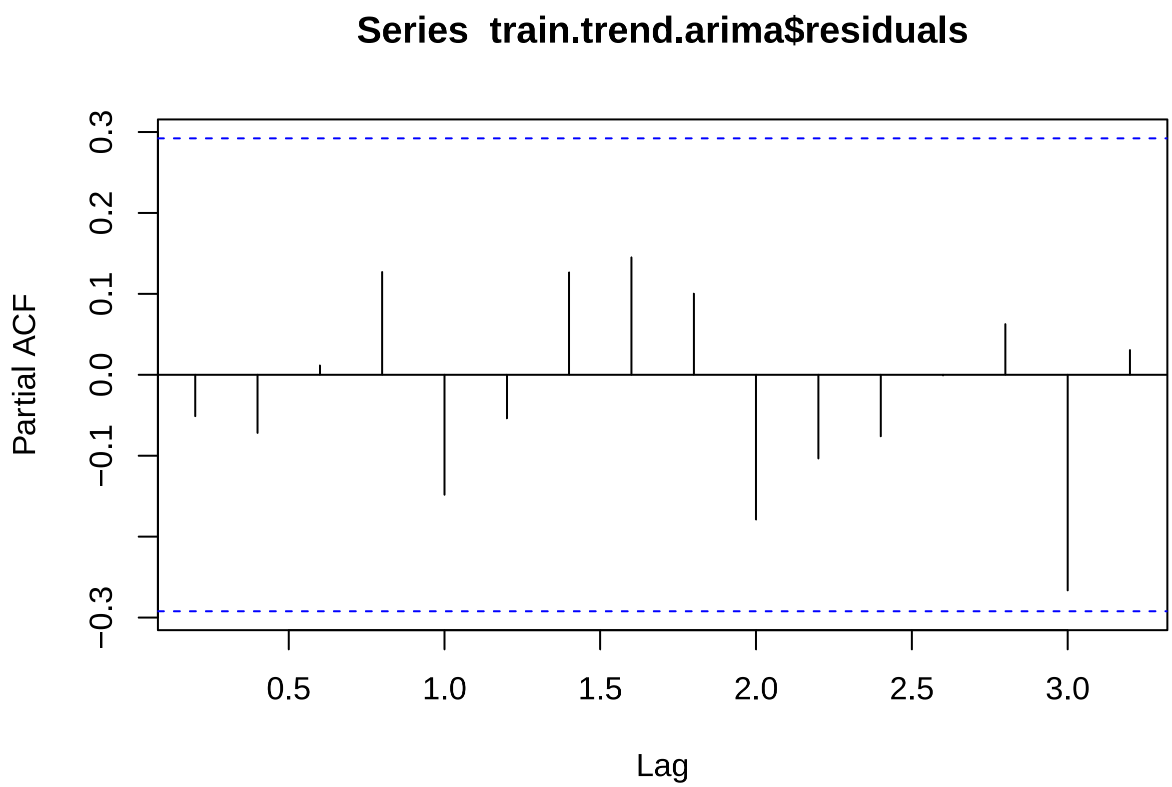
```
##ARIMA
traffic_dum.ts=ts(traffic_dum$injuries, start = c(2001,1), end = c(2010,5),
                  freq=5)
n=floor(length(traffic_dum.ts)/10)
train.ts <- window(traffic_dum.ts, start = c(2001, 1), end = c(2009, n))
valid.ts <- window(traffic_dum.ts, start = c(2010, 1), end = c(2010, n))

library(forecast)
train.trend <- tslm(train.ts ~ trend + I(trend^2) + season)
train.trend.arima <- Arima(train.trend$residuals, order = c(3,2,3))
train.trend.arima.pred <- forecast(train.trend.arima, h = n)
summary(train.trend.arima)
```

```
## Series: train.trend$residuals
## ARIMA(3,2,3)
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2      ma3
##       0.6060  -0.1452  -0.2918  -2.6524  2.3089  -0.6556
## s.e.  0.2242   0.1669   0.1544   0.5592  0.9929   0.4384
##
## sigma^2 = 0.3582:  log likelihood = -43.69
## AIC=101.37   AICc=104.57   BIC=113.7
##
## Training set error measures:
##                      ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.04595489 0.5427023 0.4581148 94.35942 187.4523 0.5262178
##                     ACF1
## Training set -0.05106203
```

```
acf(train.trend.arima$residuals)
```



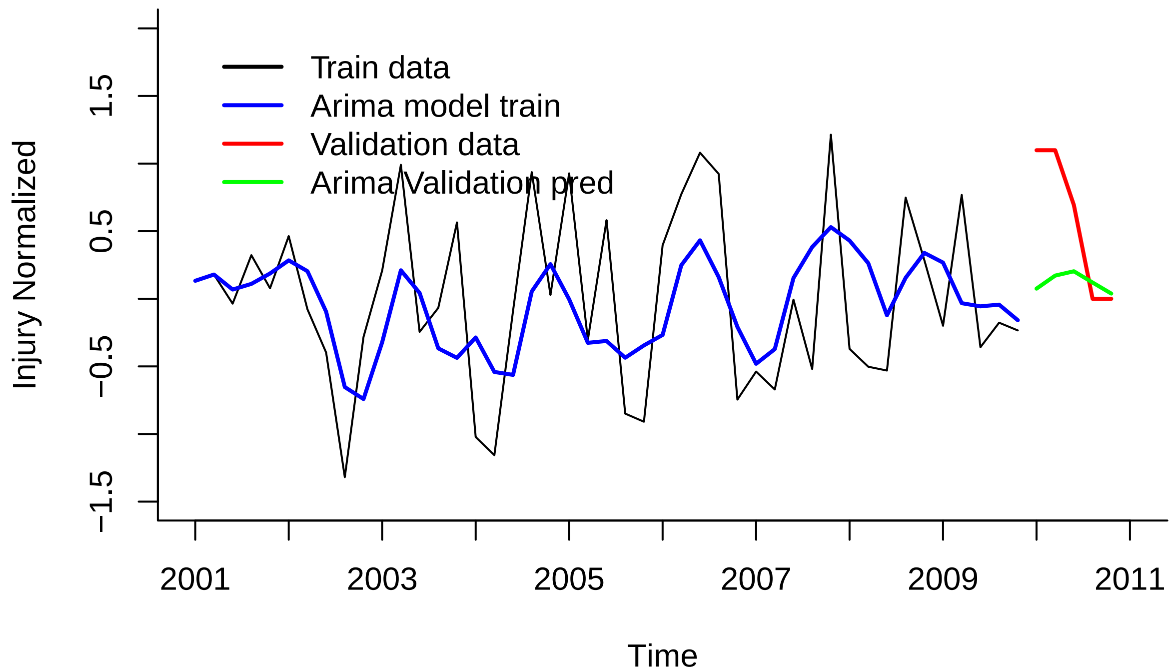

```
pacf(train.trend.arima$residuals)
```




```
train.trend.arima.pred
```

```
##         Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 2010.00     0.07518628 -0.7190385 0.8694111 -1.139476 1.289848
## 2010.20     0.17165516 -0.6216040 0.9649143 -1.041530 1.384840
## 2010.40     0.20353546 -0.5928253 0.9998963 -1.014393 1.421464
## 2010.60     0.12023097 -0.7167992 0.9572611 -1.159896 1.400358
## 2010.80     0.03857271 -0.8076417 0.8847871 -1.255601 1.332746
```

```
plot(train.trend$residuals, ylab = "Injury Normalized",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(2001,2011), ylim=c(-1.5,2),
     main = "")
axis(1, at = seq(2001, 2011, 1), labels = format(seq(2001, 2011, 1)))
lines(train.trend.arima.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts, col = 'red', lwd=2,)
lines(train.trend.arima.pred$mean, lwd = 2, col = "green")
legend(2001,2,c("Train data","Arima model train", "Validation data",
                "Arima Validation pred"), lty=c(1,1,1,1),
       lwd=c(2,2,2,2), bty = "n", col =c("black","blue","red","green"))
```



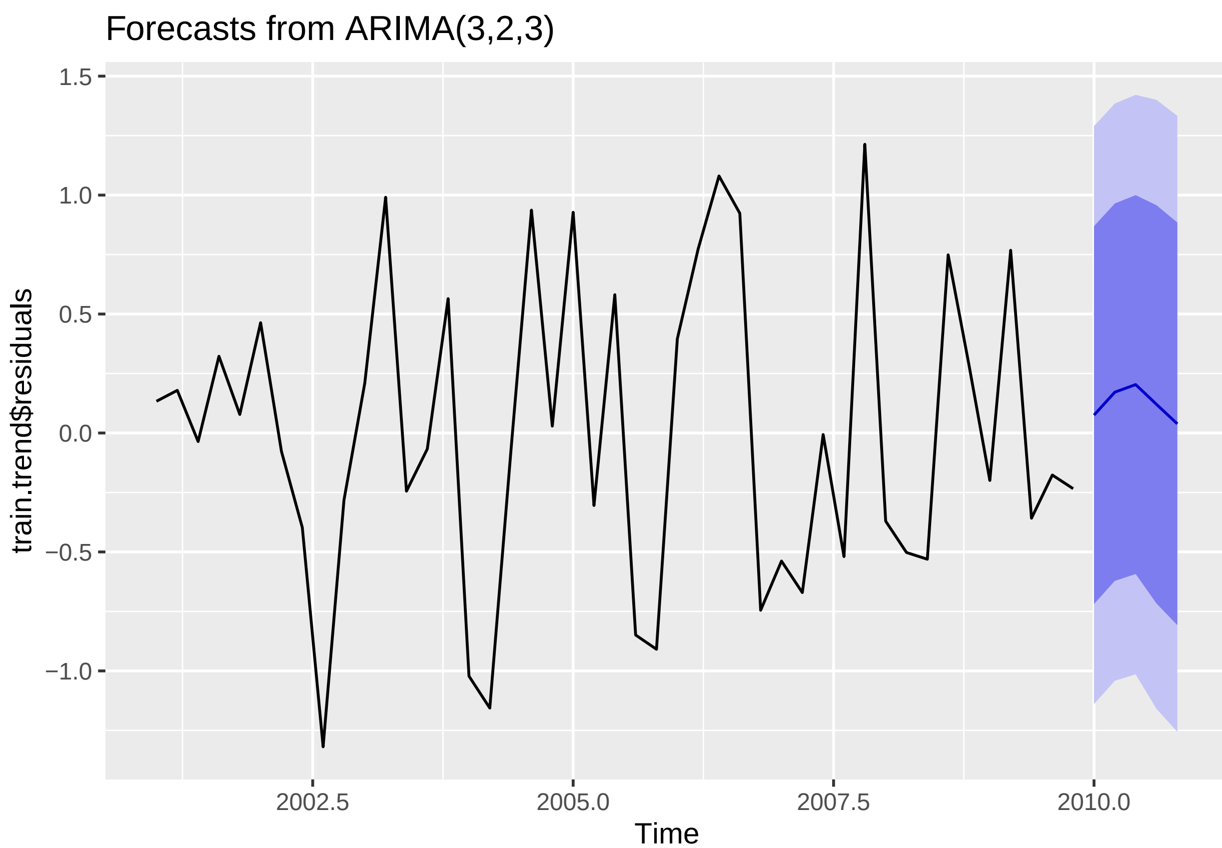

```r
valid.trend.arima.pred <- forecast(train.trend.arima, newdata=vaild.ts)
```

```
## Warning in forecast.forecast_ARIMA(train.trend.arima, newdata = vaild.ts): The
## non-existent newdata arguments will be ignored.
```

```r
valid.trend.arima.pred
```

```
##         Point Forecast       Lo 80     Hi 80     Lo 95    Hi 95
## 2010.00    0.075186275 -0.7190385 0.8694111 -1.139476 1.289848
## 2010.20    0.171655159 -0.6216040 0.9649143 -1.041530 1.384840
## 2010.40    0.203535464 -0.5928253 0.9998963 -1.014393 1.421464
## 2010.60    0.120230968 -0.7167992 0.9572611 -1.159896 1.400358
## 2010.80    0.038572713 -0.8076417 0.8847871 -1.255601 1.332746
## 2011.00   -0.006517952 -0.8526842 0.8396483 -1.300618 1.287582
## 2011.20    0.003922381 -0.8517020 0.8595468 -1.304642 1.312487
## 2011.40    0.042224911 -0.8245906 0.9090404 -1.283455 1.367905
## 2011.60    0.078679303 -0.7925898 0.9499484 -1.253812 1.411170
## 2011.80    0.093764918 -0.7774051 0.9649349 -1.238575 1.426104
```

```r
## Forecast from ARIMA
autoplot(train.trend.arima.pred)
```




```r
## Neural Network
traffic_dum.ts=ts(traffic_dum$injuries, start = c(2001,1), end = c(2010,5),
                  freq=5)
n=floor(length(traffic_dum.ts)/10)
train.ts <- window(traffic_dum.ts, start = c(2001, 1), end = c(2009, n))
valid.ts <- window(traffic_dum.ts, start = c(2010, 1), end = c(2010, n))

library(forecast)
train.trend <- tslm(train.ts ~ trend + I(trend^2) + season)
train.trend.nnet <- nnetar(train.trend$residuals, repeats=20, P=1, size=7)
train.trend.nnet.pred <- forecast(train.trend.nnet, h = n)
summary(train.trend.nnet.pred)
```

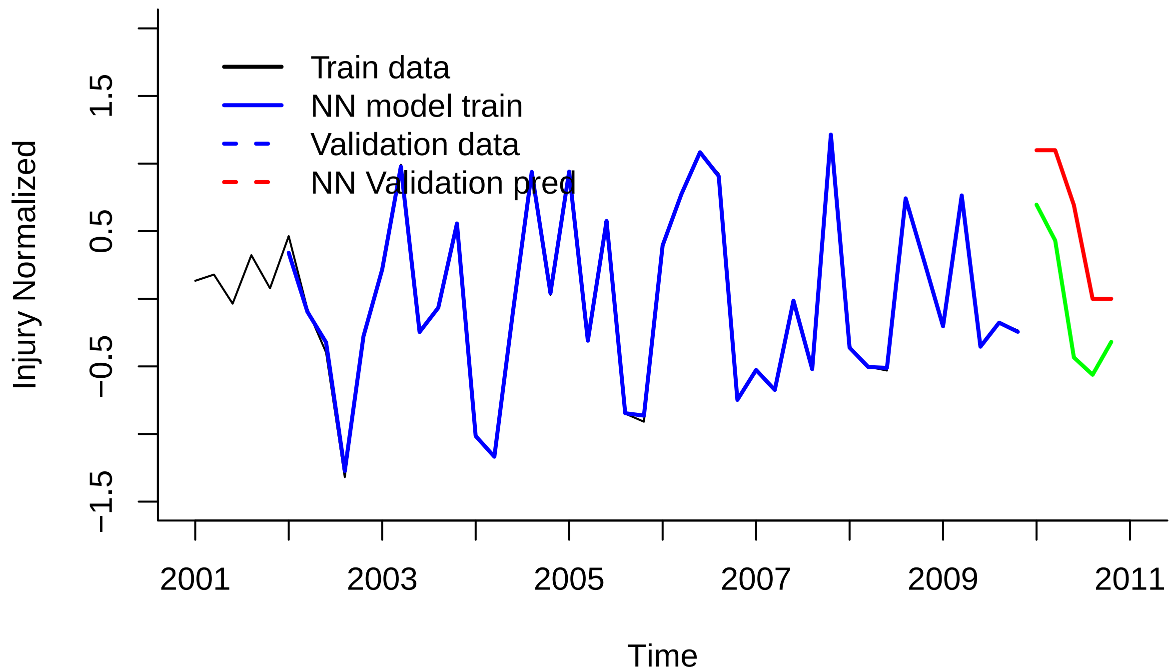
```
##
## Forecast method: NNAR(5,1,7)[5]
##
## Model Information:
##
## Average of 20 networks, each of which is
## a 5-7-1 network with 50 weights
## options were - linear output units
##
## Error measures:
##                          ME       RMSE        MAE      MPE     MAPE       MASE
## Training set -0.0002047017 0.02624207 0.01314175 -2.95566 6.946195 0.01509539
```

```
##                    ACF1
## Training set 0.1644987
##
## Forecasts:
##          Point Forecast
## 2010.00       0.6962091
## 2010.20       0.4307866
## 2010.40      -0.4336651
## 2010.60      -0.5616935
## 2010.80      -0.3190286
```

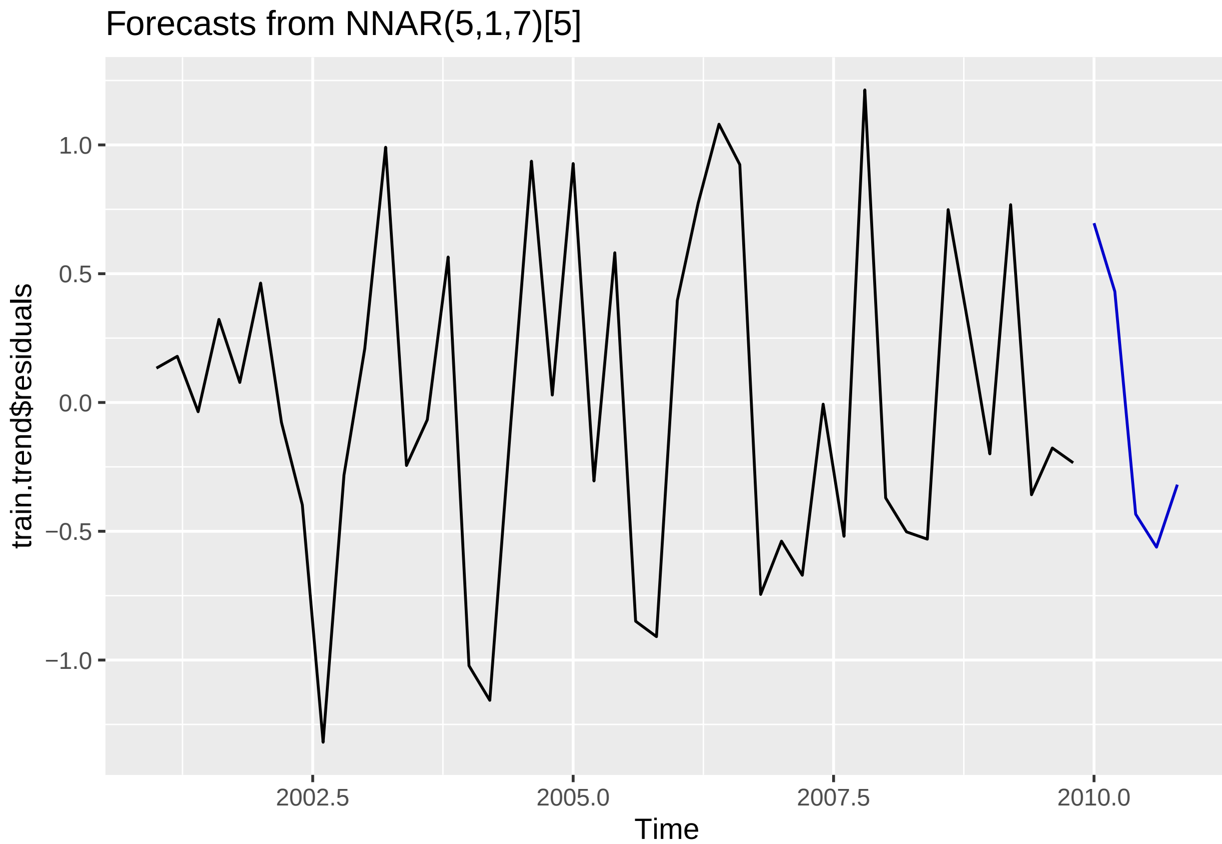
```
train.trend.nnet.pred
```

```
##          Point Forecast
## 2010.00       0.6962091
## 2010.20       0.4307866
## 2010.40      -0.4336651
## 2010.60      -0.5616935
## 2010.80      -0.3190286
```

```
plot(train.trend$residuals, ylab = "Injury Normalized",
     xlab = "Time", bty = "l", xaxt = "n", xlim = c(2001,2011),
     ylim=c(-1.5,2), main = "")
axis(1, at = seq(2001, 2011, 1), labels = format(seq(2001, 2011, 1)))
lines(train.trend.nnet.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts, col = 'red', lwd=2,)
lines(train.trend.nnet.pred$mean, lwd = 2, col = "green")
legend(2001,2,c("Train data","NN model train",
                "Validation data", "NN Validation pred"), lty=c(1,1,2,2),
       lwd=c(2,2,2,2), bty = "n", col =c("black","blue","blue","red"))
```




```
## Forecast from NN
autoplot(train.trend.nnet.pred)
```


Forecasts from NNAR(5,1,7)[5]
1.0
0.5
0.0
−0.5
−1.0
train.trend$residuals
2002.5
2005.0
2007.5
2010.0
Time