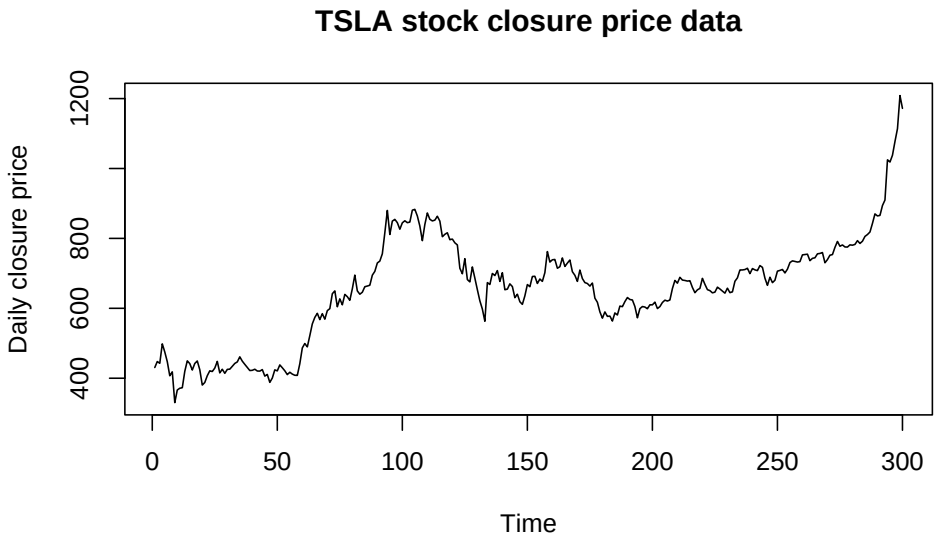
Stats 153 Project - Zoey

2. Exploratory Data Analysis

Here we explore the data. Naturally, the first plot you should make is the data itself. Point out any visible features, e.g. heteroscedasticity, seasonality, trend. To observe the recent pattern better, we only use the last 300 data points from the time series. Each data point represents daily closure price for TSLA stock. From observing the original data, it seems that the variability of the time series data set appears to be non-constant, which is heterscedasticity. Therefore we transform original data through $f(Y_t) = log(Y_t)$.

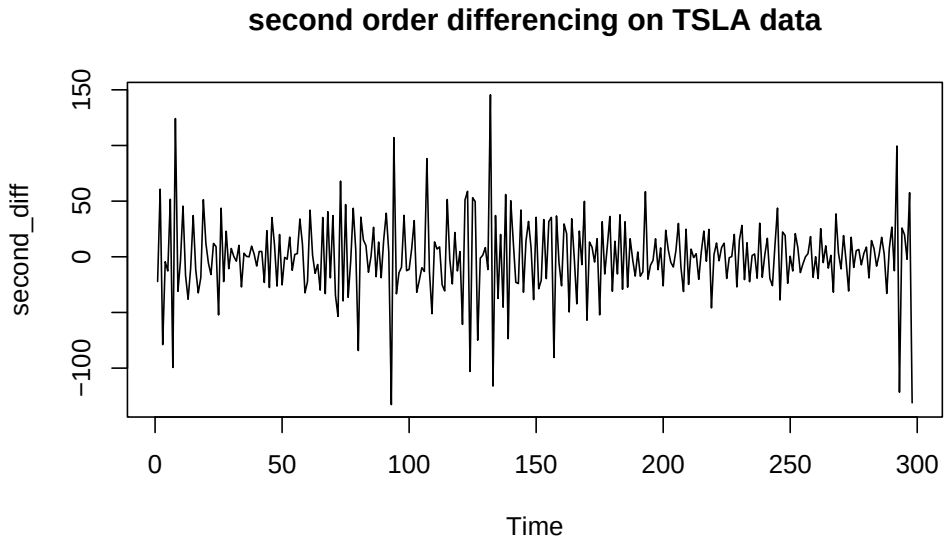
```
t = 1:300
data <- read.csv("data/TSLA.csv")
open <- data$Open
close <- data$Adj.Close
close_data = tail(close, 300)
plot.ts(close_data, main="TSLA stock closure price data", ylab="Daily closure price")
```



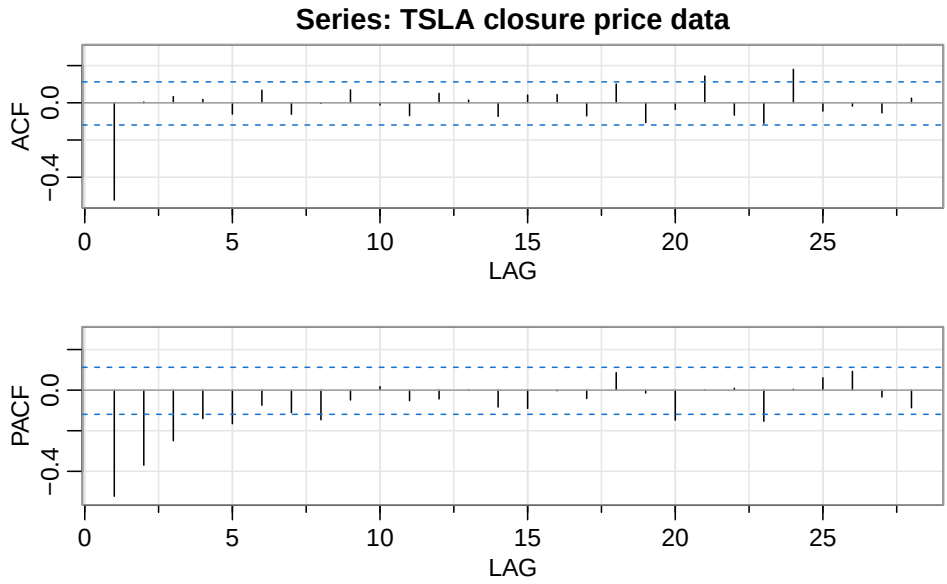
Models considered: Second Order Differencing + ARMA

Second order differencing, observe the data, the data seems stationary without obvioud trend or seasonality.

```
diff = diff(close_data, lag=1)
second_diff = diff(diff, lag = 1)
plot.ts(second_diff, main="second order differencing on TSLA data")
```




```
acf2(second_diff, main="Series: TSLA closure price data")
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.52  0.00  0.03  0.02 -0.06  0.07 -0.06  0.00  0.07 -0.01 -0.07  0.05
## PACF -0.52 -0.37 -0.25 -0.14 -0.17 -0.07 -0.11 -0.15 -0.05  0.02 -0.05 -0.04
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01 -0.07  0.04  0.04 -0.07  0.10 -0.11 -0.04  0.14 -0.07 -0.11  0.18
## PACF  0.00 -0.08 -0.09  0.00 -0.04  0.09 -0.01 -0.15  0.00  0.01 -0.15  0.00
##      [,25] [,26] [,27] [,28]
## ACF  -0.04 -0.02 -0.05  0.02
## PACF  0.06  0.09 -0.03 -0.09
```

Evaluation Evaluating AIC, BIC, AICc, and time-series cross validation. Evaluate model based on how well they predict future values. Choose model: (1, 2, 1)(0, 0, 0)[0]. This is because observing ACF plot on data after second order differencing, there is a cut off on approximately lag=1, and Recall that for MA(q) model, the sample ACF will contain zero values for lag $|h| > q$. Therefore we try MA(1) model. Observing partial ACF plot, there is an exponential decay, indicating we should use an ARMA(p, q) model. We use the function auto.arima to build intuition on what p, q value to choose.

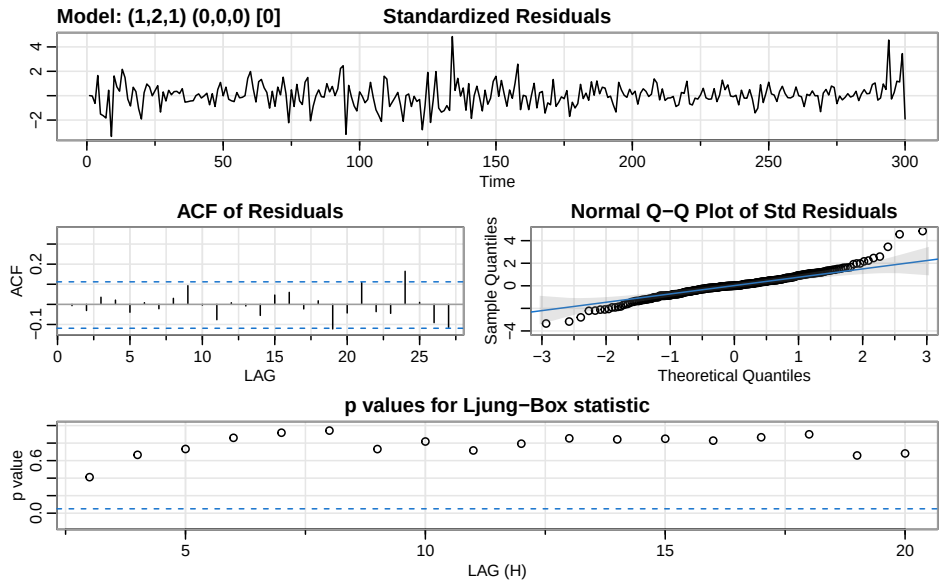
```
auto.arima(second_diff)
```

```
## Series: second_diff
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##           ar1      ma1
##       -0.1087  -0.9488
## s.e.   0.0635   0.0276
##
## sigma^2 estimated as 582.9:  log likelihood=-1371.93
## AIC=2749.87   AICc=2749.95   BIC=2760.96
```

The model looks well on Ljung-Box statistic with all p values above 0.05. Also the ACF of residuals shows no ACF go beyond two blue band.

```
model1 <- sarima(close_data, p=1, d=2, q=1, P=0, D=0, Q=0, S=0)
```

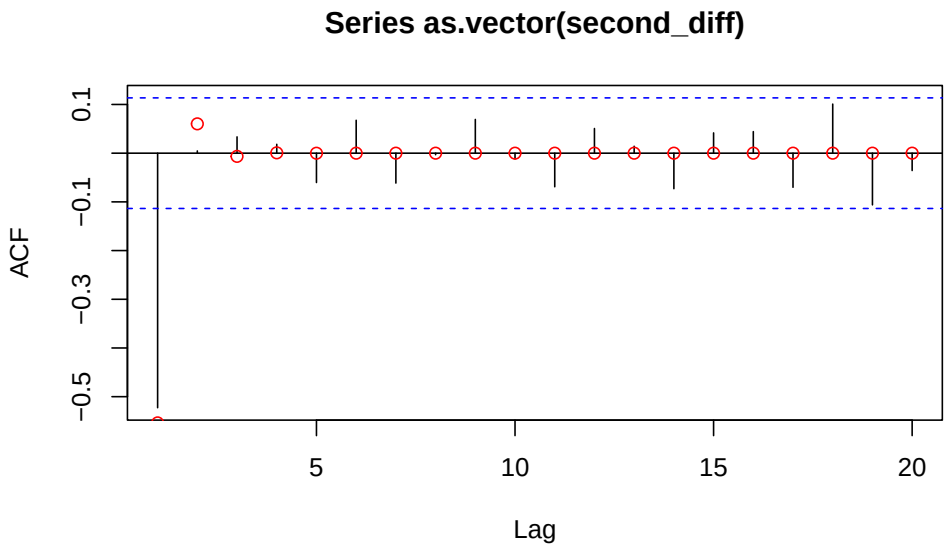
```
## initial  value 3.556748
## iter   2 value 3.293231
## iter   3 value 3.236037
## iter   4 value 3.208883
## iter   5 value 3.193975
## iter   6 value 3.187240
## iter   7 value 3.185019
## iter   8 value 3.182390
## iter   9 value 3.181715
## iter  10 value 3.181599
## iter  11 value 3.181571
## iter  12 value 3.181571
## iter  12 value 3.181571
## iter  12 value 3.181571
## final  value 3.181571
## converged
## initial  value 3.184899
## iter   2 value 3.184872
## iter   3 value 3.184869
## iter   4 value 3.184869
## iter   4 value 3.184869
## iter   4 value 3.184869
## final  value 3.184869
## converged
```




```
model1$ttable
```

```
##     Estimate     SE  t.value p.value
## ar1  -0.1087 0.0635  -1.7118   0.088
## ma1  -0.9488 0.0276 -34.3423   0.000
```

```
model1.acf=ARMAacf(ar=c(-0.1087), ma=c(-0.9488), lag.max = 20, pacf=FALSE)
acf(as.vector(second_diff), lag.max=20)
points(0:20, model1.acf, col='red')
```




```
cat("AIC", model1$AIC)
```

```
## AIC 9.227749
```

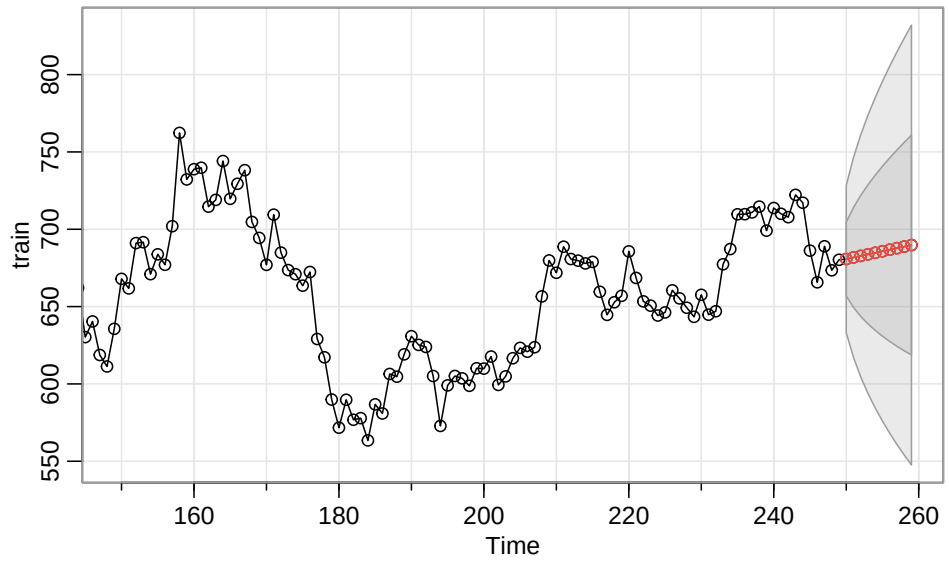
```
cat("BIC", model1$BIC)
```

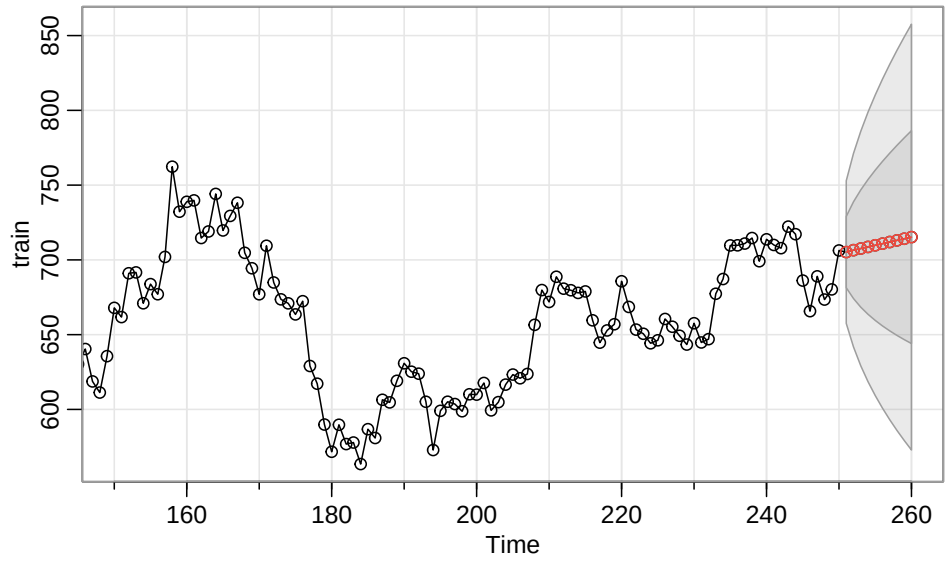
```
## BIC 9.264968
```

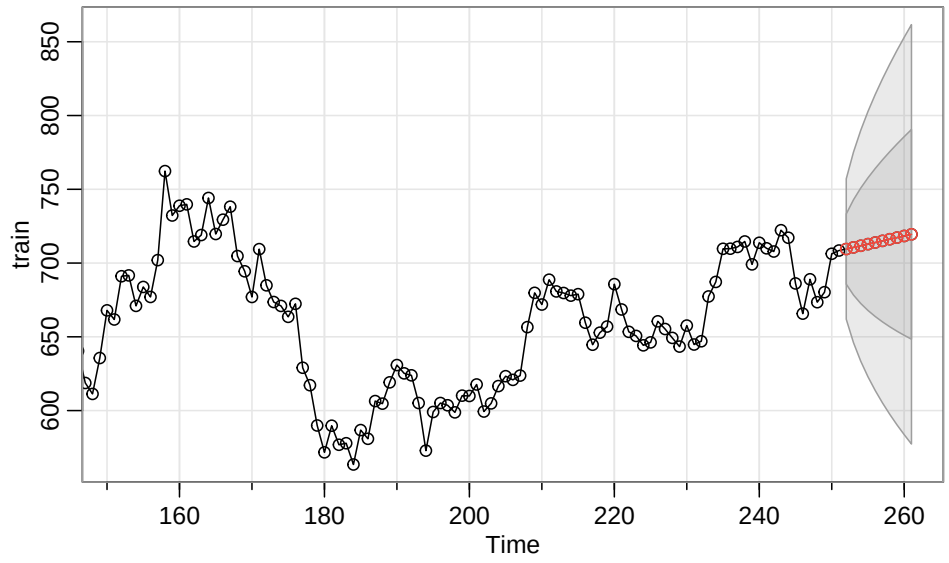
```
cat("AICc", model1$AICc)
```

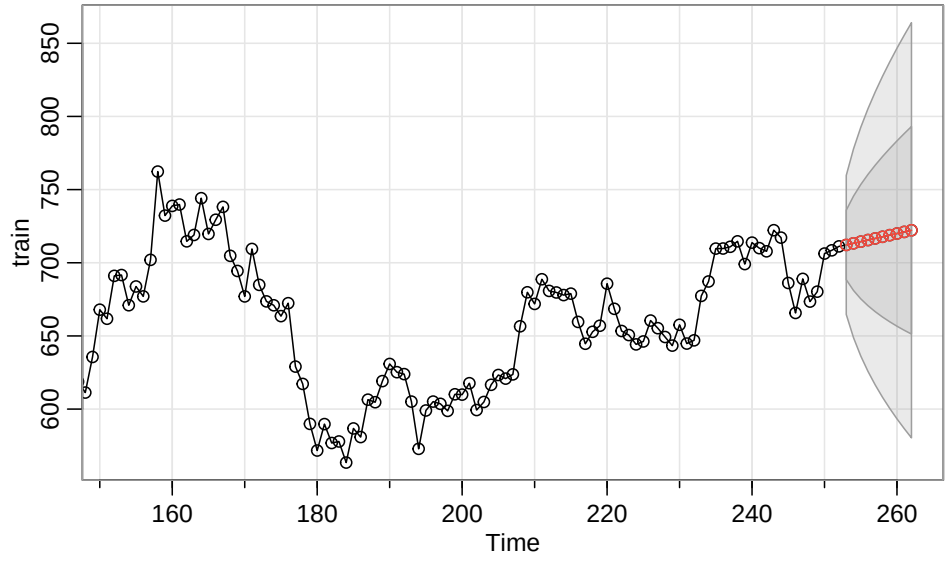
```
## AICc 9.227885
```

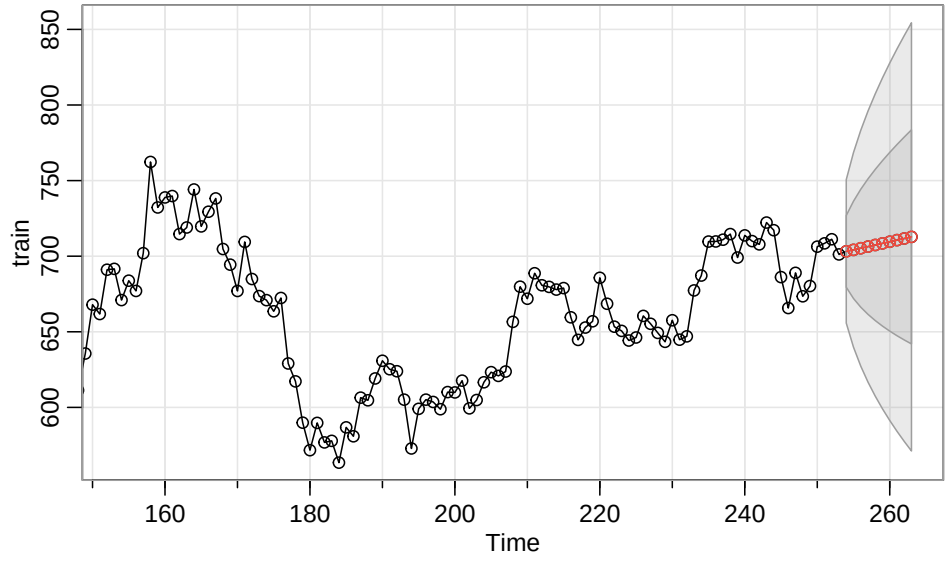
```r
# cross validation
start_day <- 250
end_day <- 290
jump = 10
error = 0
forward_time = 10
# we are performing n-fold validation
n = (end_day - start_day) / jump

for (k in start_day:end_day) {
  train <- window(close_data, end=k-0.001)
  test <- window(close_data, start=k, end=k + forward_time - 0.01)
  forecast <- sarima.for(train, n.ahead = forward_time, p=1, d=2, q=1, P=0, D=0, Q=0, S=0)
  error = error + sum((forecast$pred - test)^2)
  k = k + 10
}
```

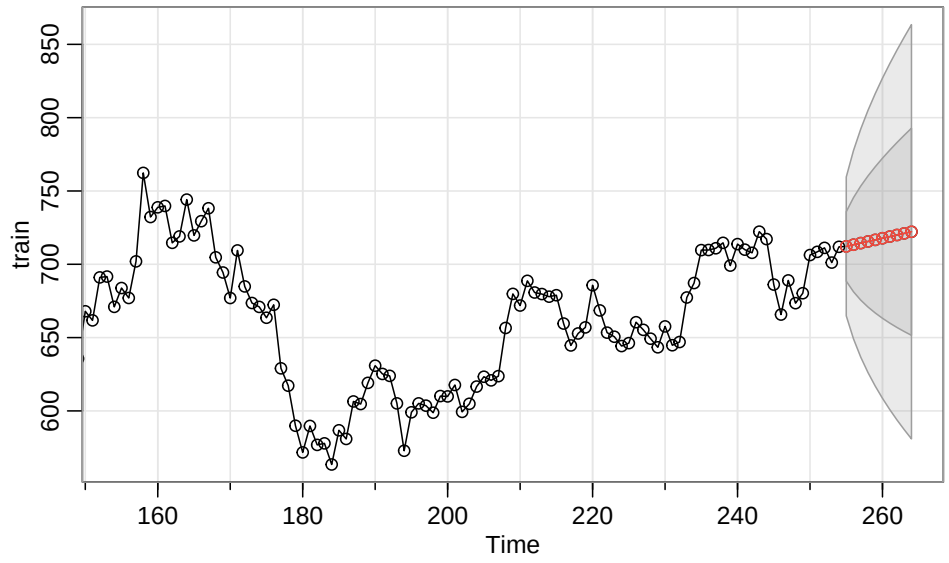


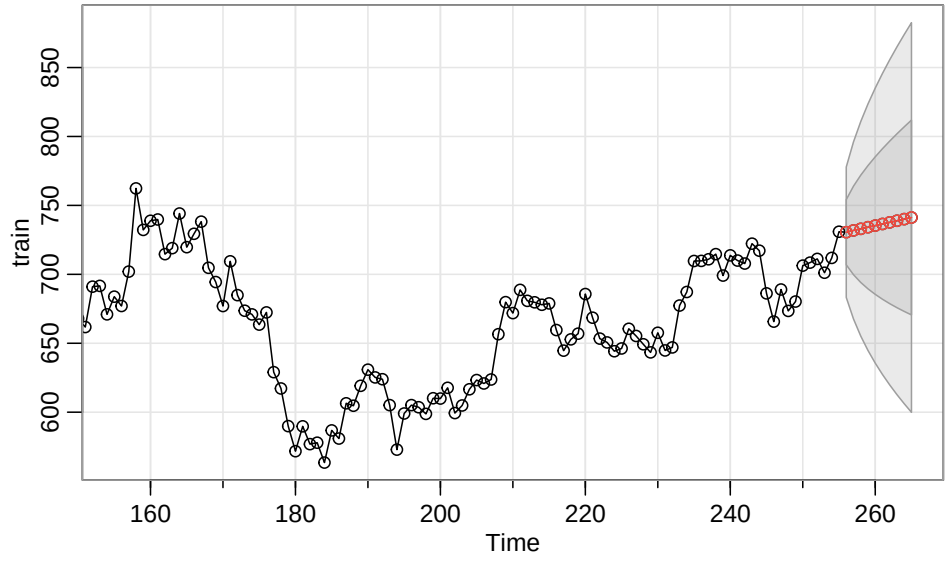


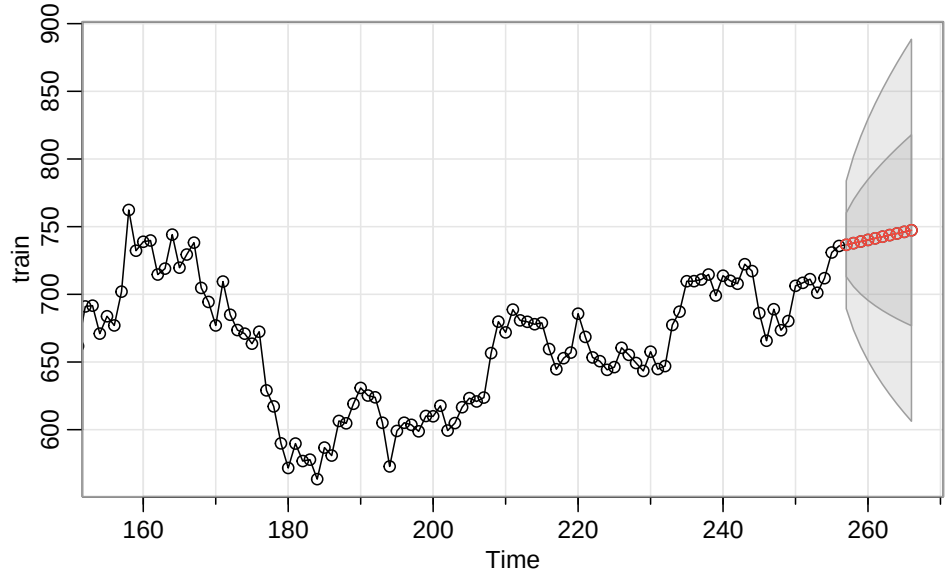

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

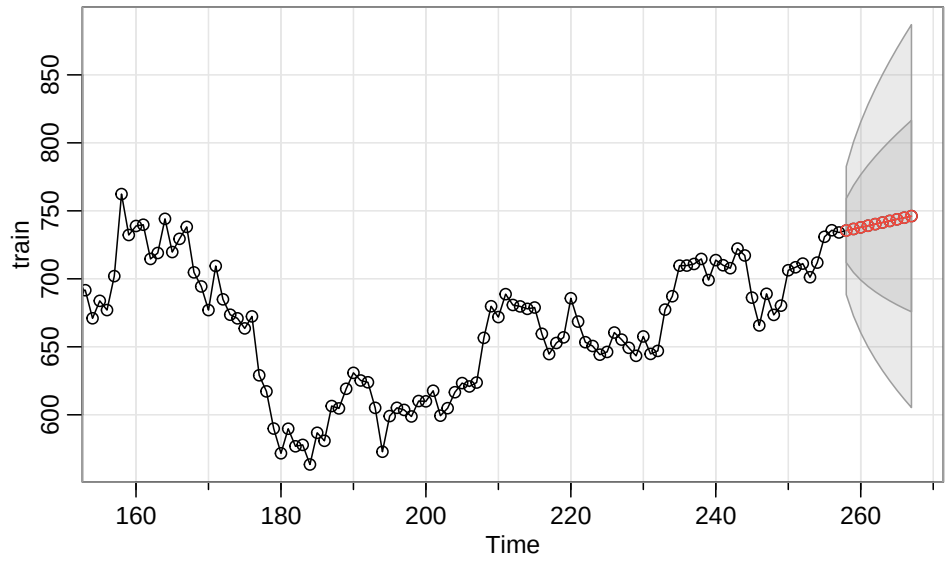

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

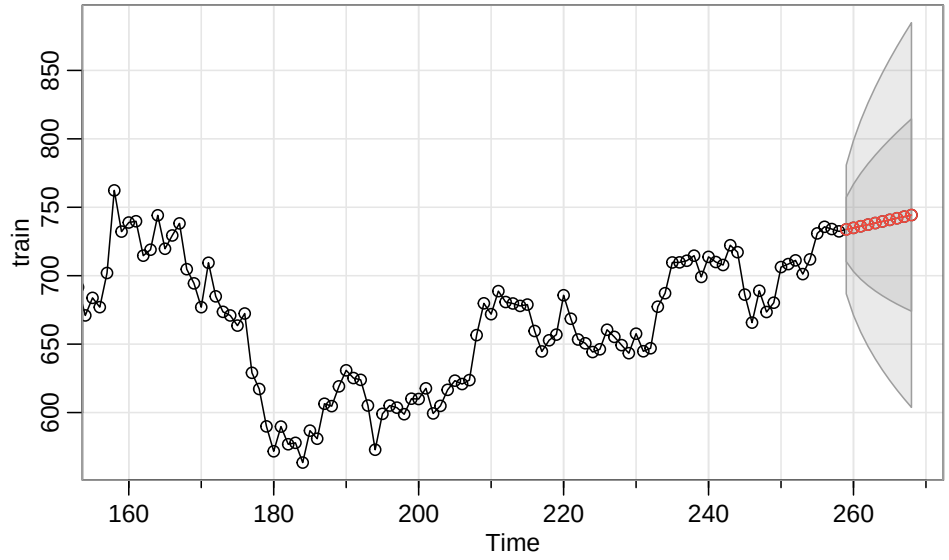

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

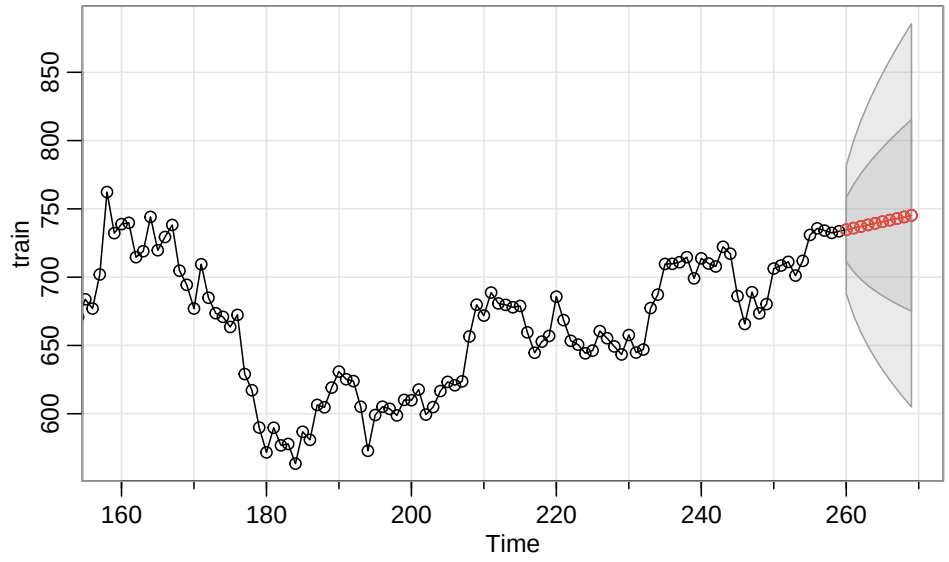

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

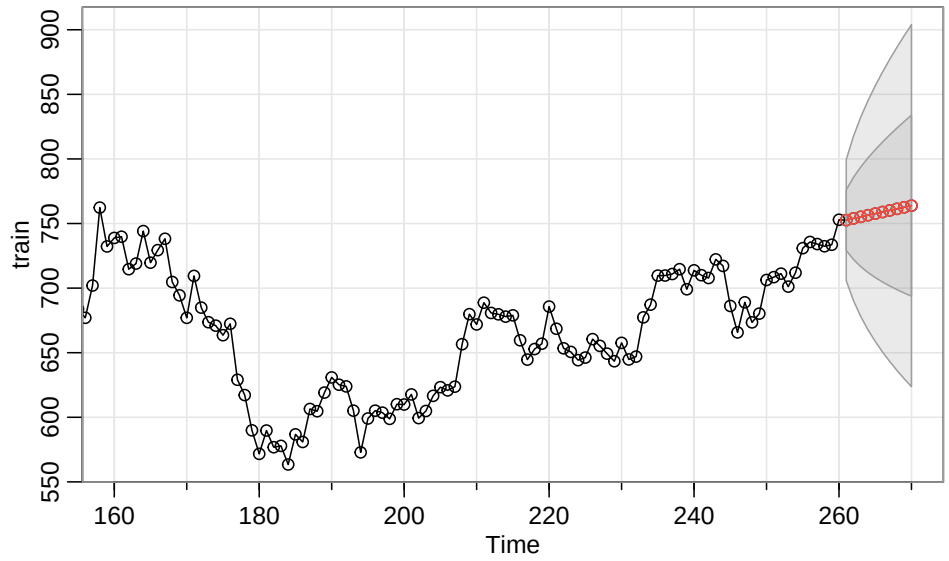

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

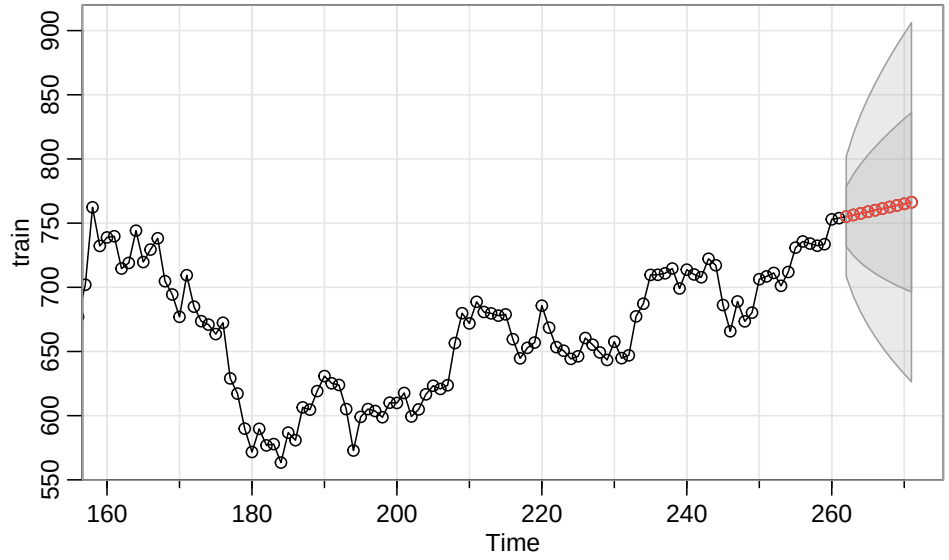

train
600
650
700
750
800
850
900
160
180
200
220
240
260
Time

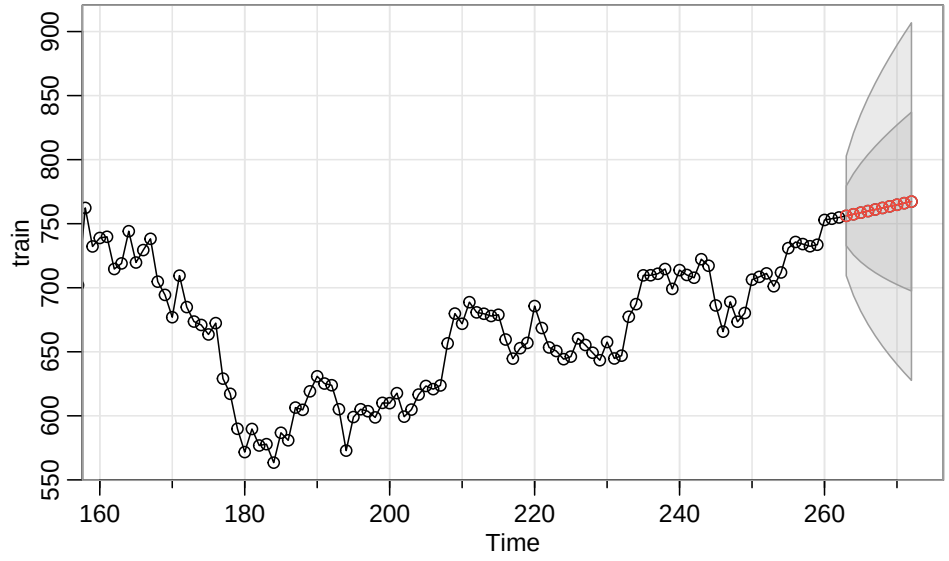

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

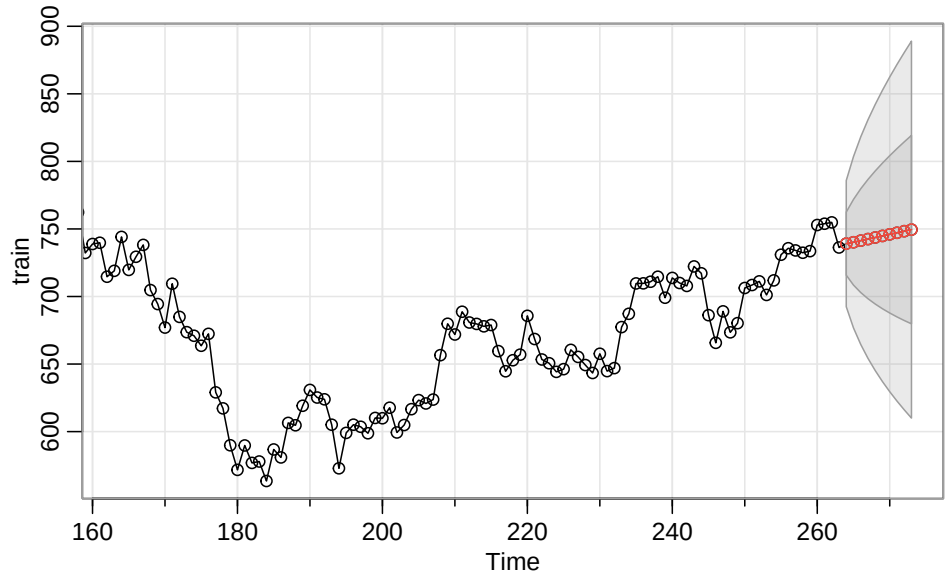

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

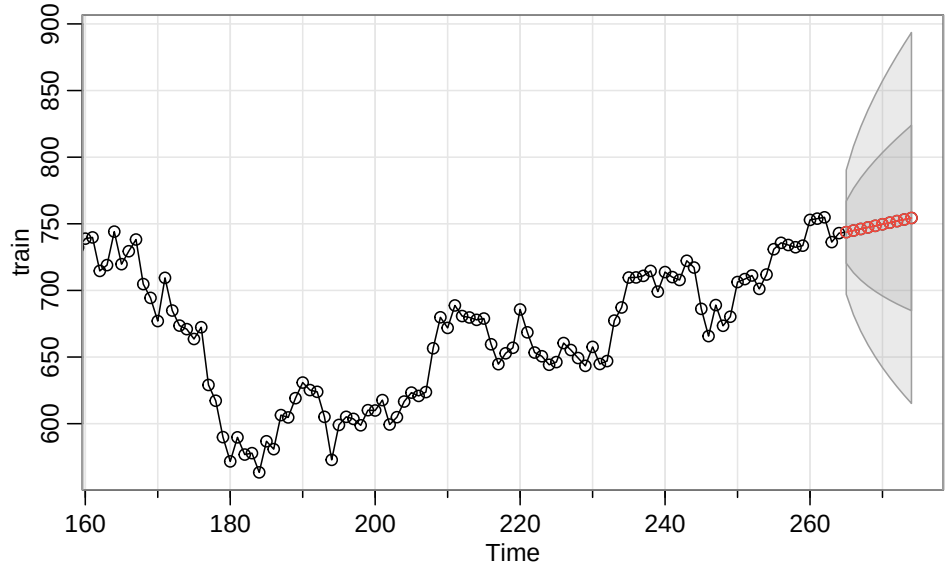

train
600
650
700
750
800
850
160
180
200
220
240
260
Time

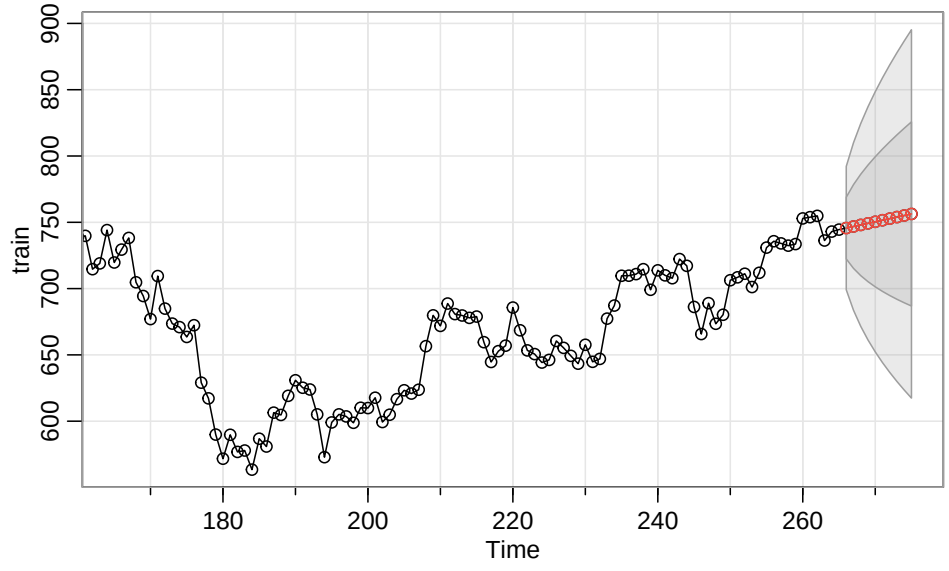

train
900
850
800
750
700
650
600
550
160
180
200
220
240
260
Time

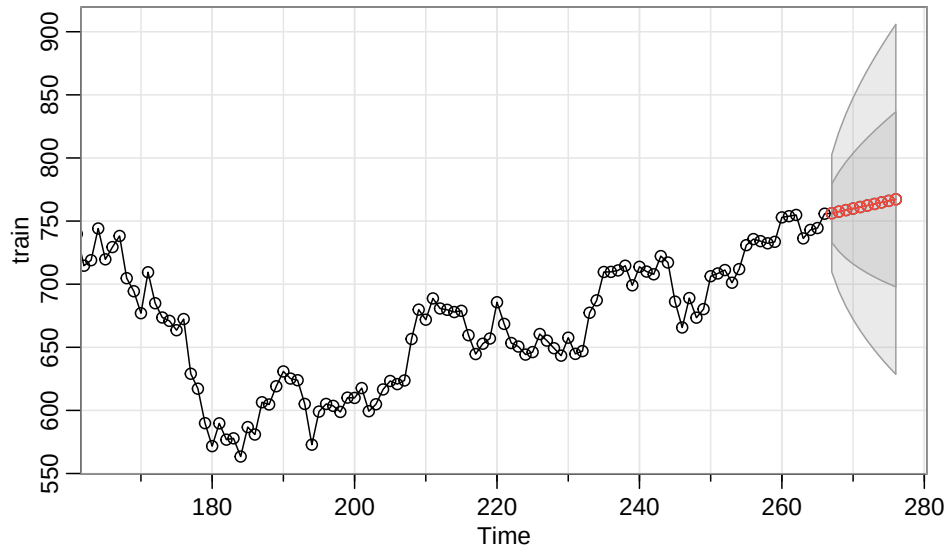

train
900
850
800
750
700
650
600
550
160
180
200
220
240
260
Time

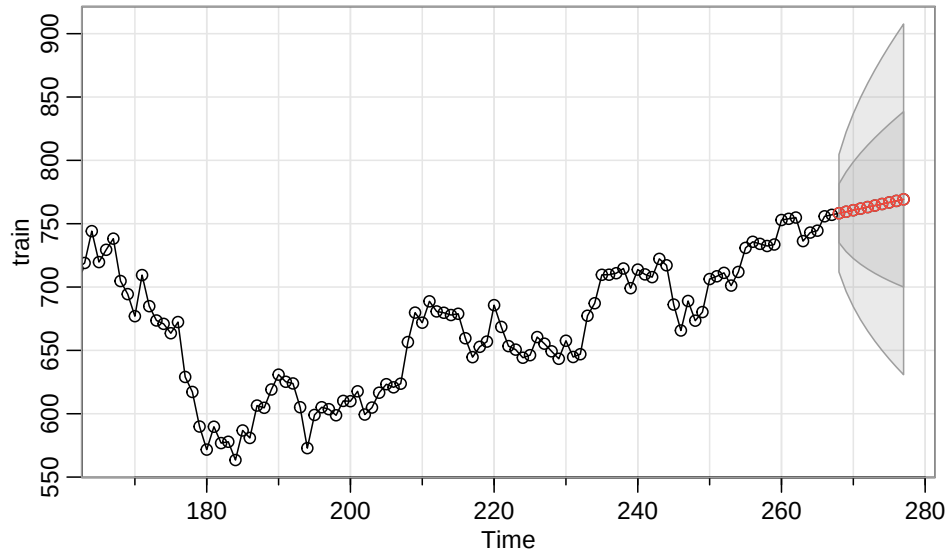

train
900
850
800
750
700
650
600
550
160
180
200
220
240
260
Time

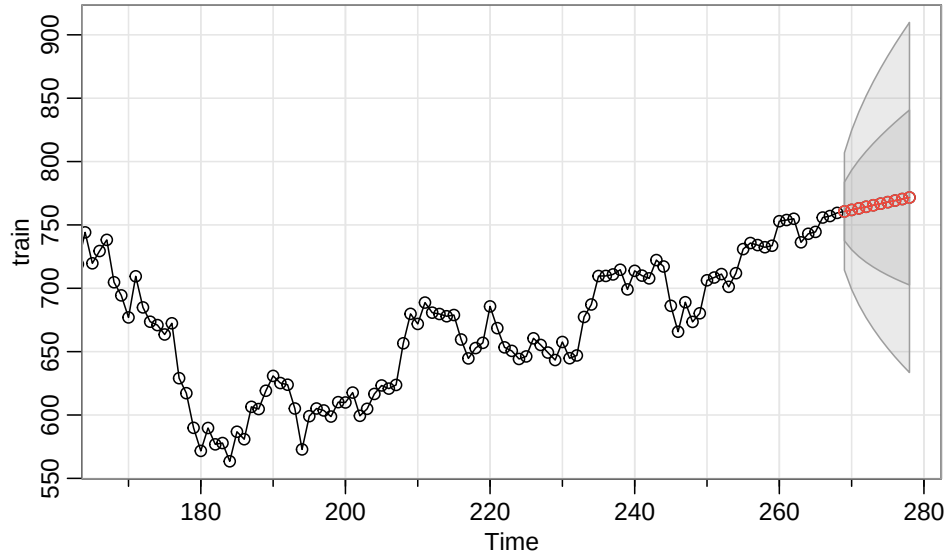

train
900
850
800
750
700
650
600
160
180
200
220
240
260
Time

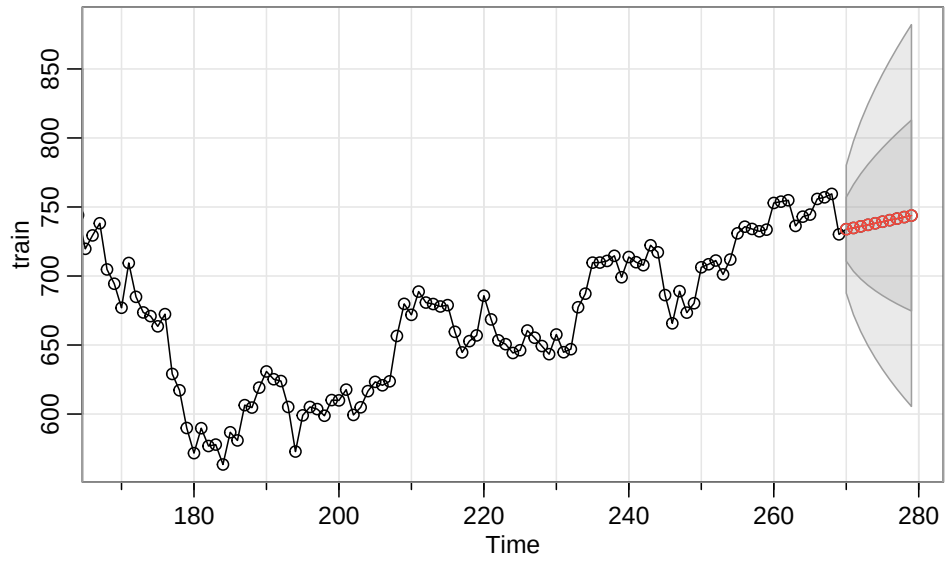

train
900
850
800
750
700
650
600
160
180
200
220
240
260
Time

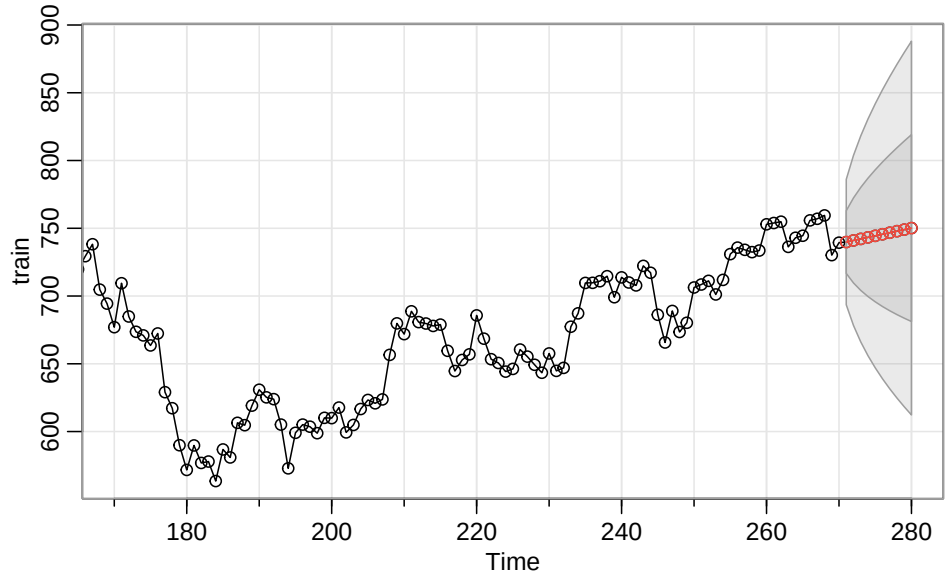

train
900
850
800
750
700
650
600
180
200
220
240
260
Time

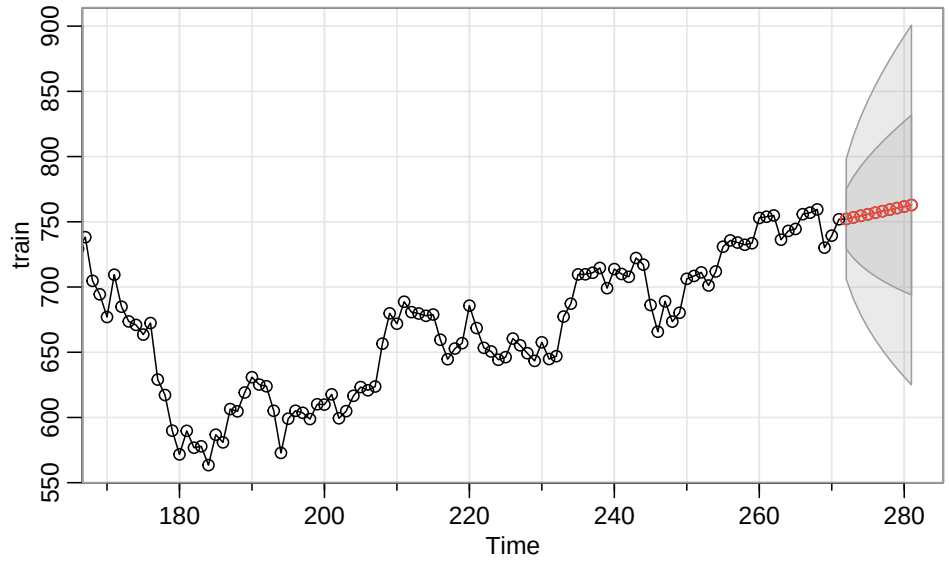

train
550
600
650
700
750
800
850
900
180
200
220
240
260
280
Time

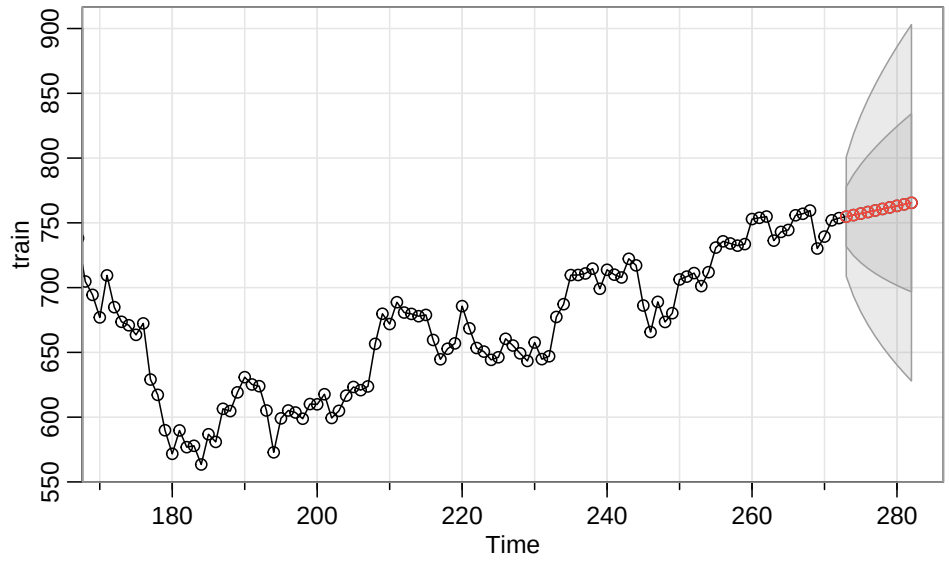

train
550
600
650
700
750
800
850
900
180
200
220
240
260
280
Time

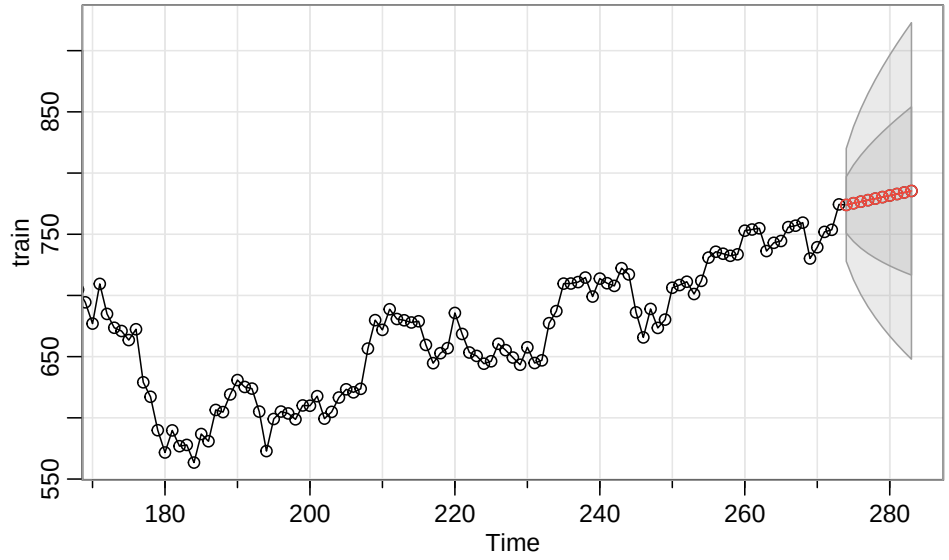

train
550
600
650
700
750
800
850
900
180
200
220
240
260
280
Time

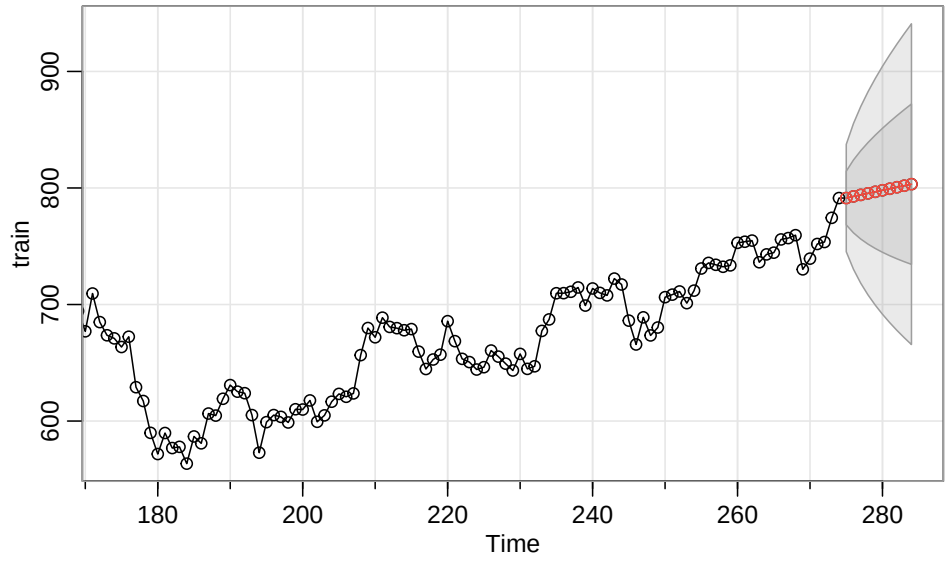

train
600
650
700
750
800
850
180
200
220
240
260
280
Time

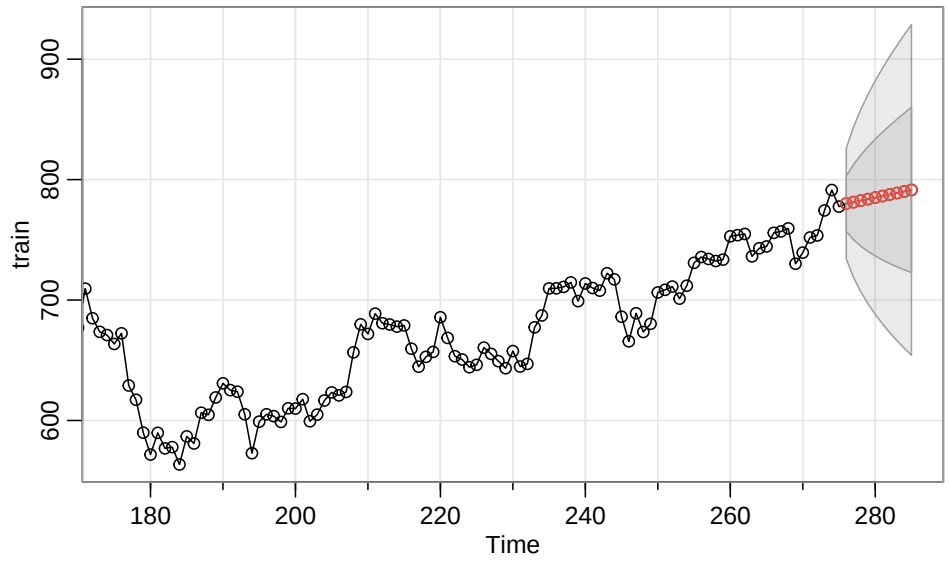

train
600
650
700
750
800
850
900
180
200
220
240
260
280
Time

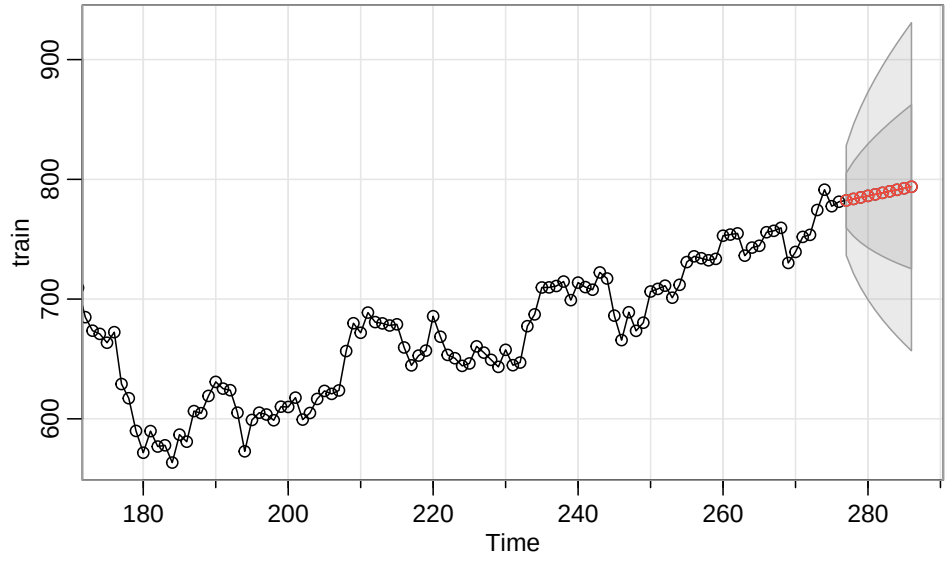

train
550
600
650
700
750
800
850
900
180
200
220
240
260
280
Time

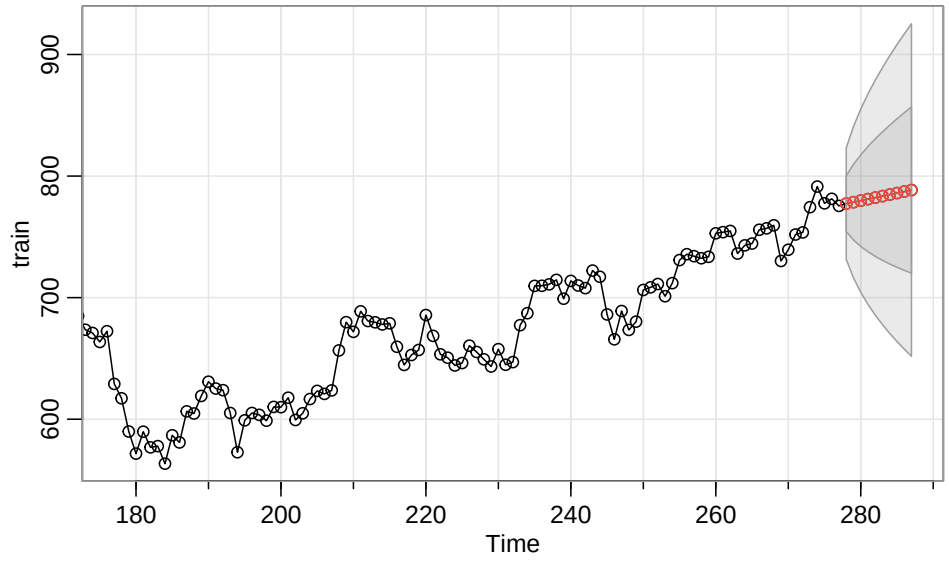

train
900
850
800
750
700
650
600
550
180
200
220
240
260
280
Time

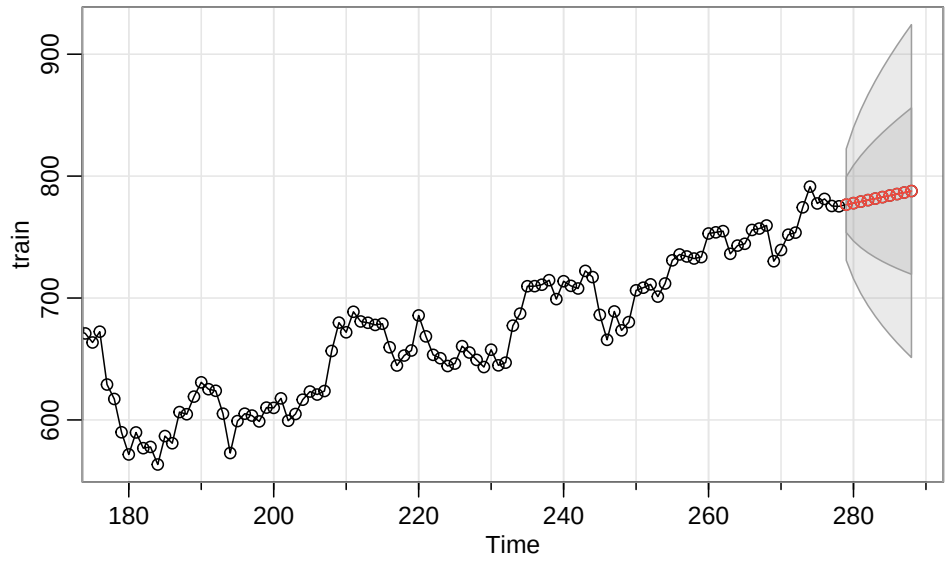

train
850
750
650
550
180
200
220
240
260
280
Time

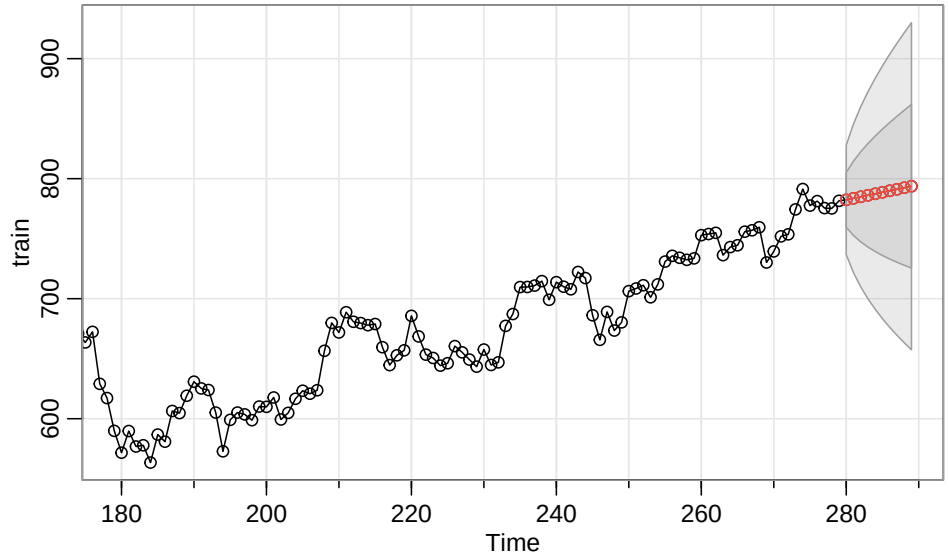

train
900
800
700
600
180
200
220
240
260
280
Time

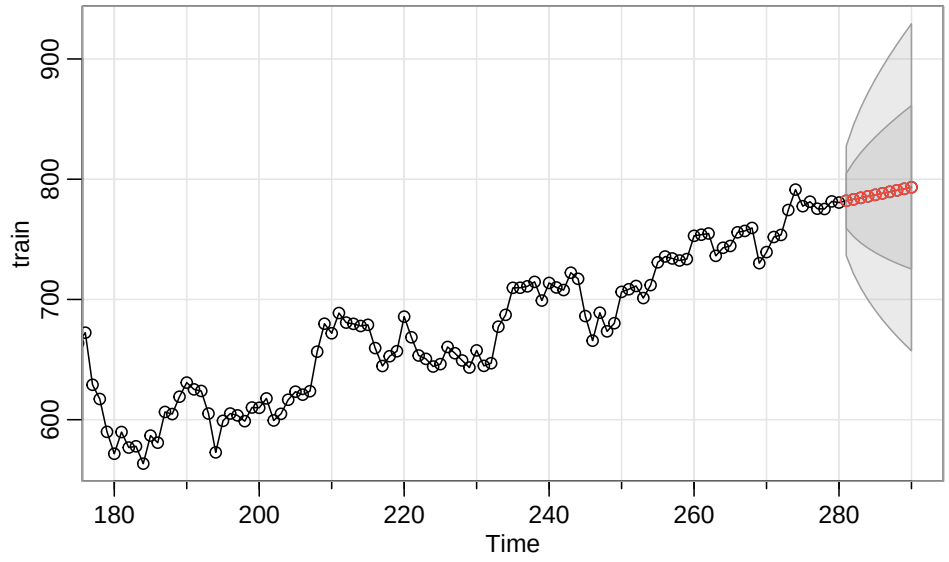

train
900
800
700
600
180
200
220
240
260
280
Time

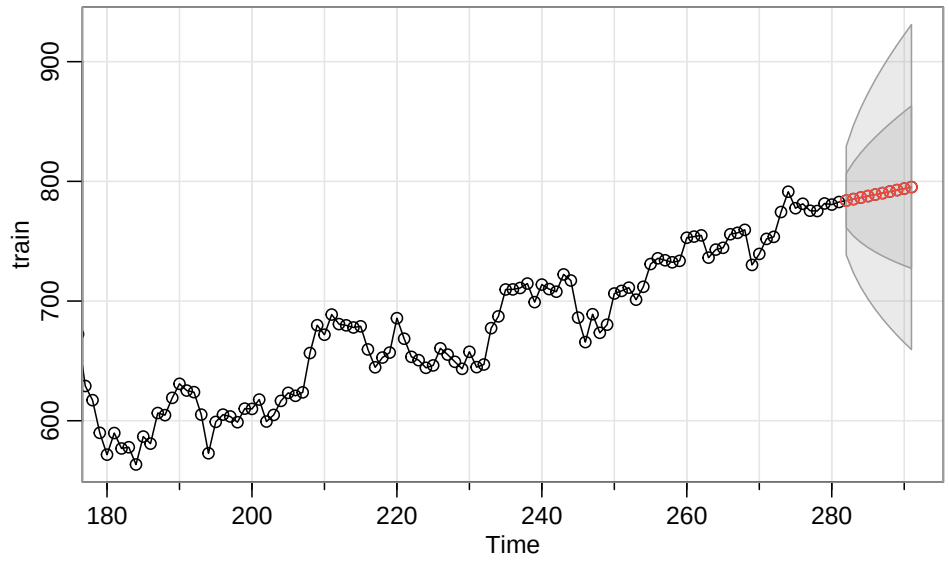

train
900
800
700
600
180
200
220
240
260
280
Time

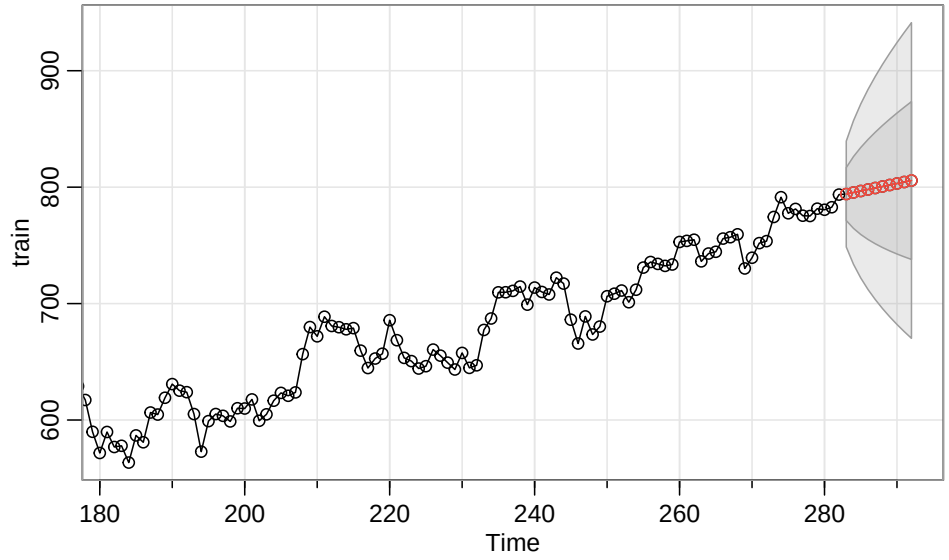

train
900
800
700
600
180
200
220
240
260
280
Time

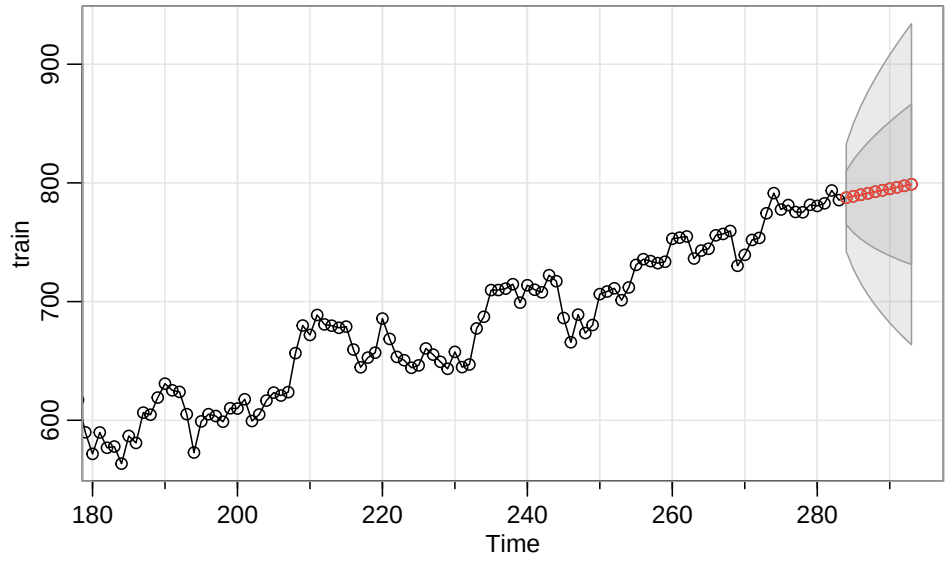

train
900
800
700
600
180
200
220
240
260
280
Time

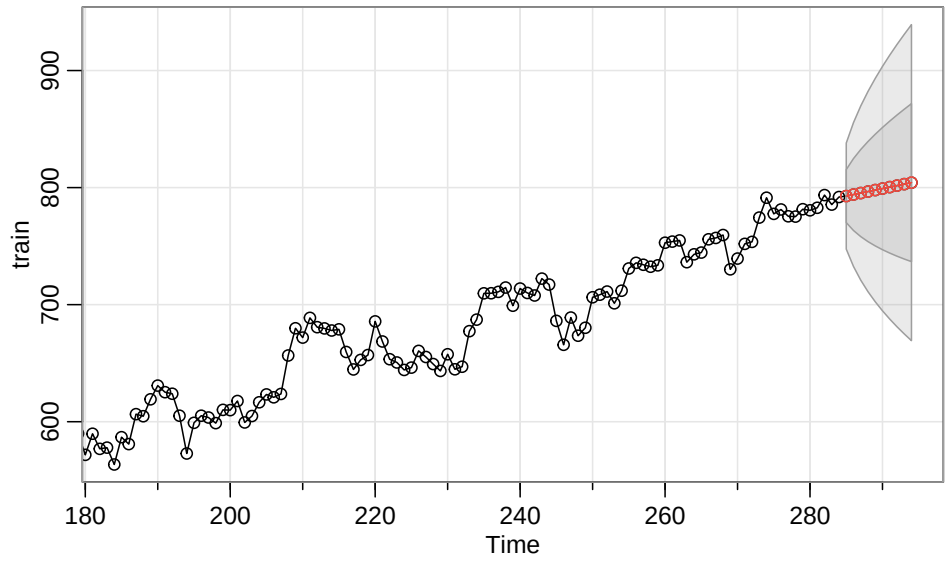

train
900
800
700
600
180
200
220
240
260
280
Time

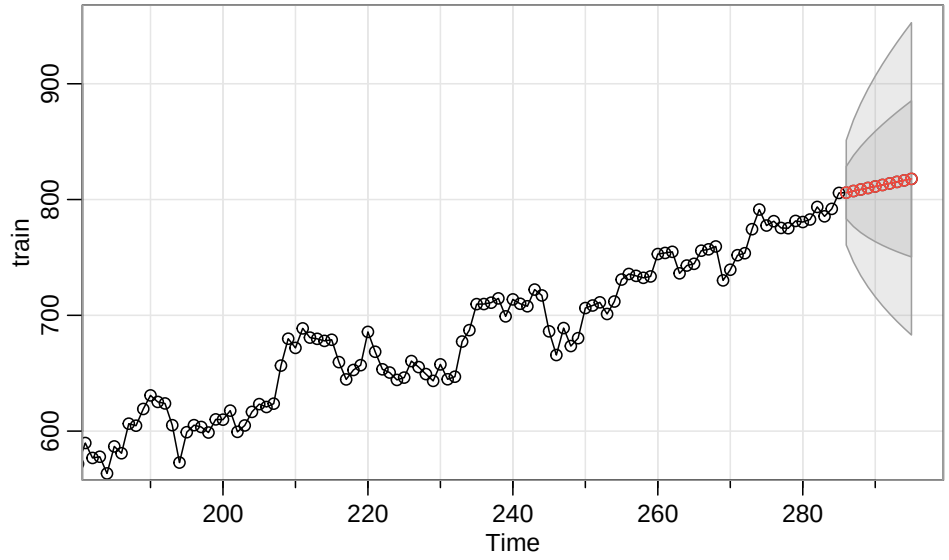

train
900
800
700
600
180
200
220
240
260
280
Time

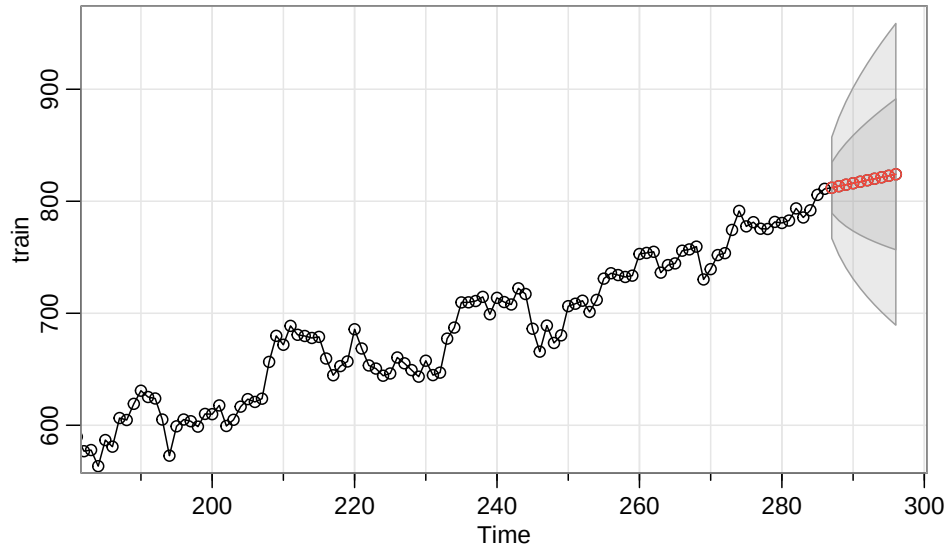

train
900
800
700
600
180
200
220
240
260
280
Time

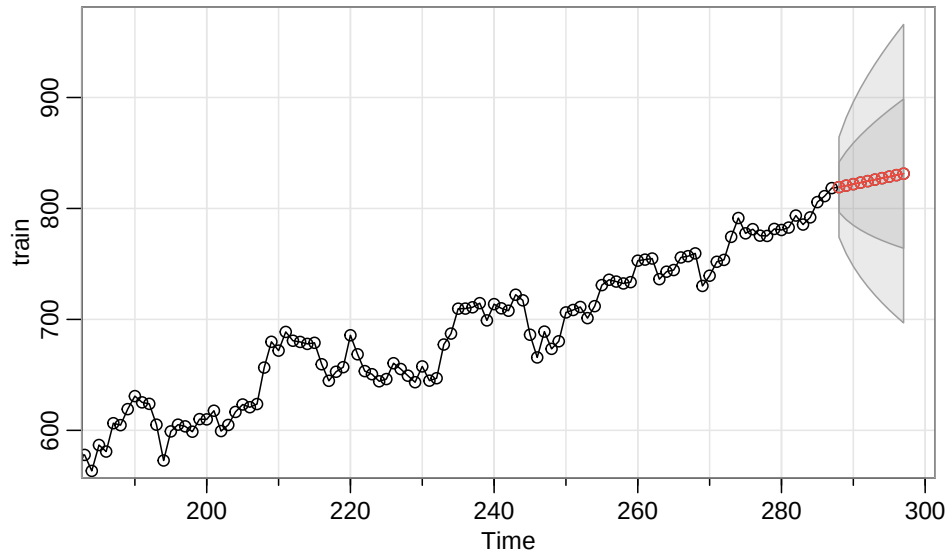

train
900
800
700
600
180
200
220
240
260
280
Time

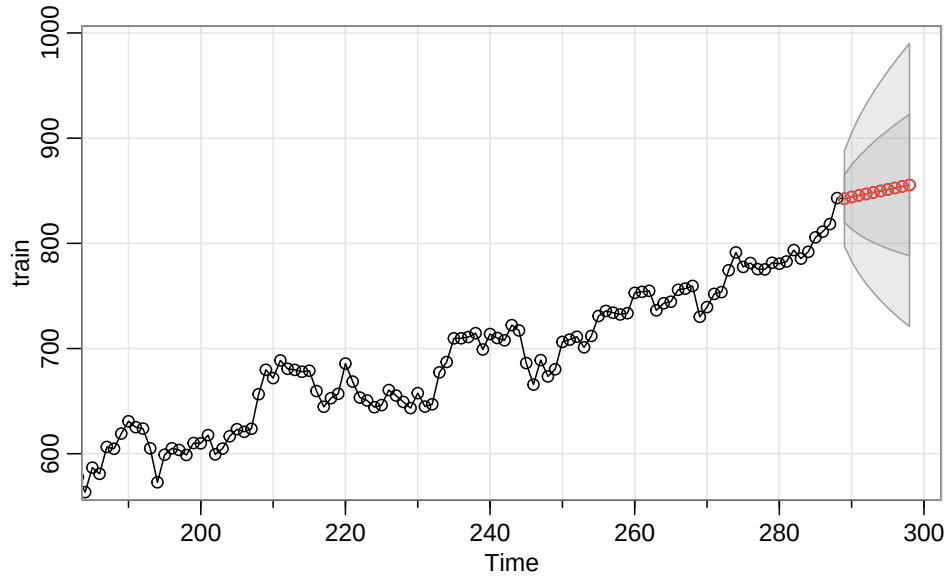

train
900
800
700
600
180
200
220
240
260
280
Time

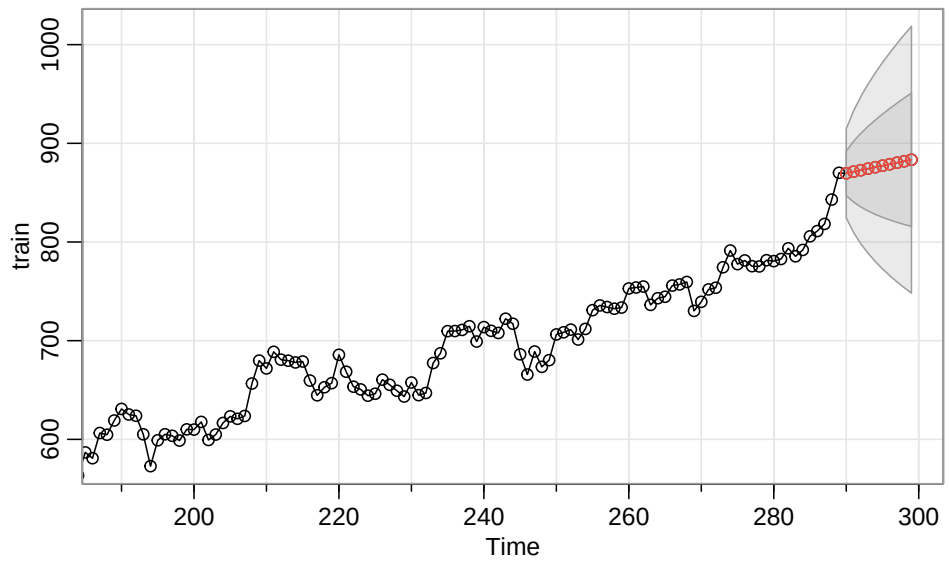

train
900
800
700
600
180
200
220
240
260
280
Time


train
900
800
700
600
200
220
240
260
280
Time


train
900
800
700
600
200
220
240
260
280
300
Time


train
600
700
800
900
200
220
240
260
280
300
Time


train
600
700
800
900
1000
200
220
240
260
280
300
Time


train
600
700
800
900
1000
200
220
240
260
280
300
Time

```
error = error / n
cat("cross validation error", error)
```

```
## cross validation error 311970.1
```

5. Results

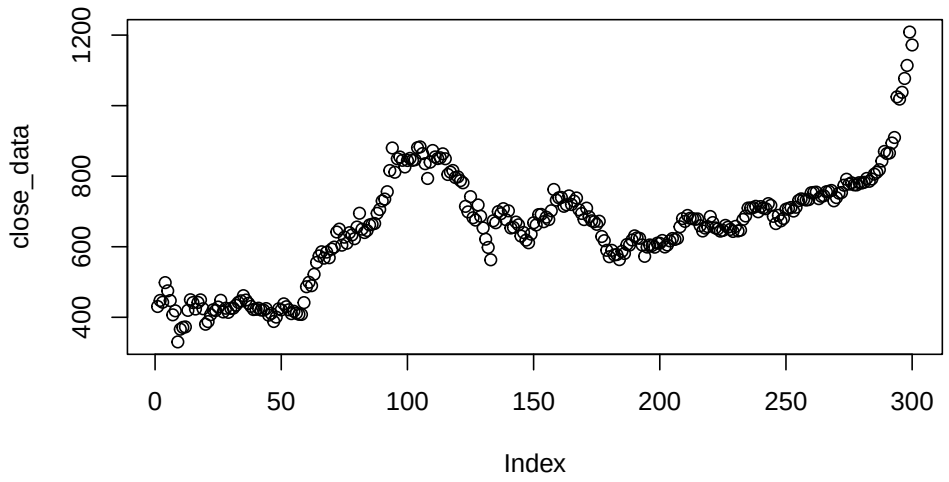
We choose model 3: 2nd-order differencing + ARMA 2A model. First, write out model mathematically. Let original time series stock data denoted by $ $\{X_t\}$ and second order differenced time series data denoted by $\{Z_t\}$. Then $(1-\phi_1 B)\nabla^2 X_t = (1-\theta_1 B)W_t$ or equivalently $(1-\phi_1 B)Z_t = (1-\theta_1 B)W_t$. Second, estimate the parameters of your chosen model, probably in a table.

```
model1$ttable
```

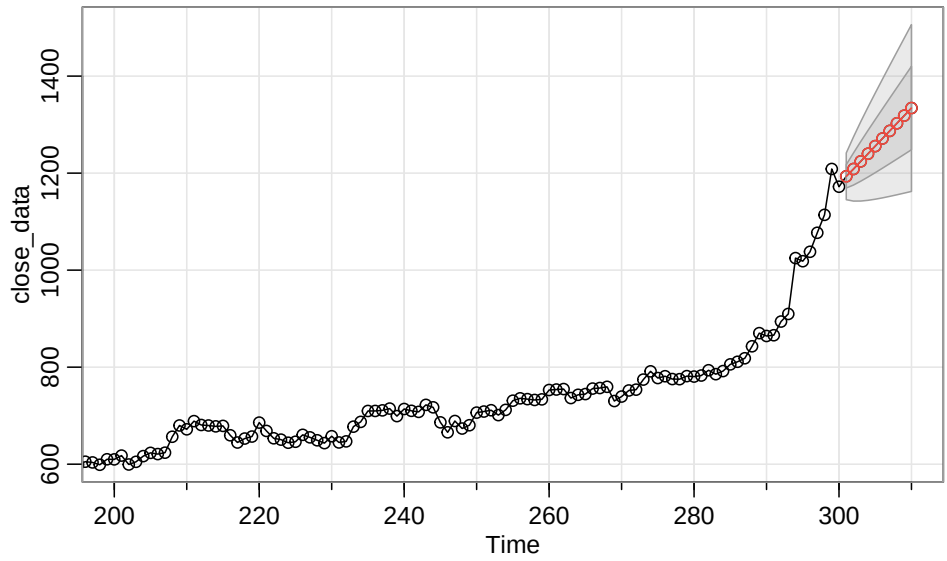
```
##     Estimate     SE  t.value p.value
## ar1  -0.1087 0.0635  -1.7118   0.088
## ma1  -0.9488 0.0276 -34.3423   0.000
```

Third, forecast appropriately and include a plot of your forecasted values appended to the end of your time series. Here I choose to forecast TSLA closure price for next 10 days. TODO: address uncertainty about prediction?

```
plot(close_data)
```




```
forecast <- sarima.for(close_data, n.ahead = 10, p=1, d=2, q=1, P=0, D=0, Q=0, S=0)
```




```
forecast$pred
```

```
## Time Series:
## Start = 301
## End = 310
## Frequency = 1
##  [1] 1193.389 1208.478 1224.251 1239.950 1255.657 1271.364 1287.070 1302.776
##  [9] 1318.482 1334.189
```

5.1 Estimation of model parameters