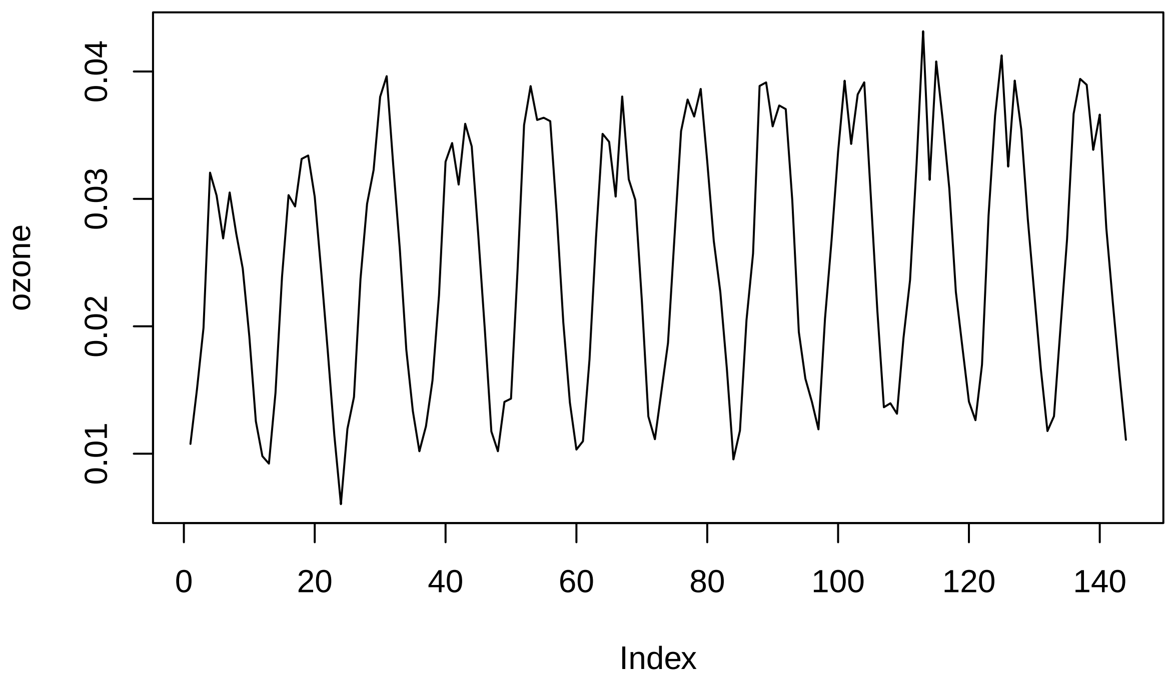
Non-parametric Model

2022-04-21

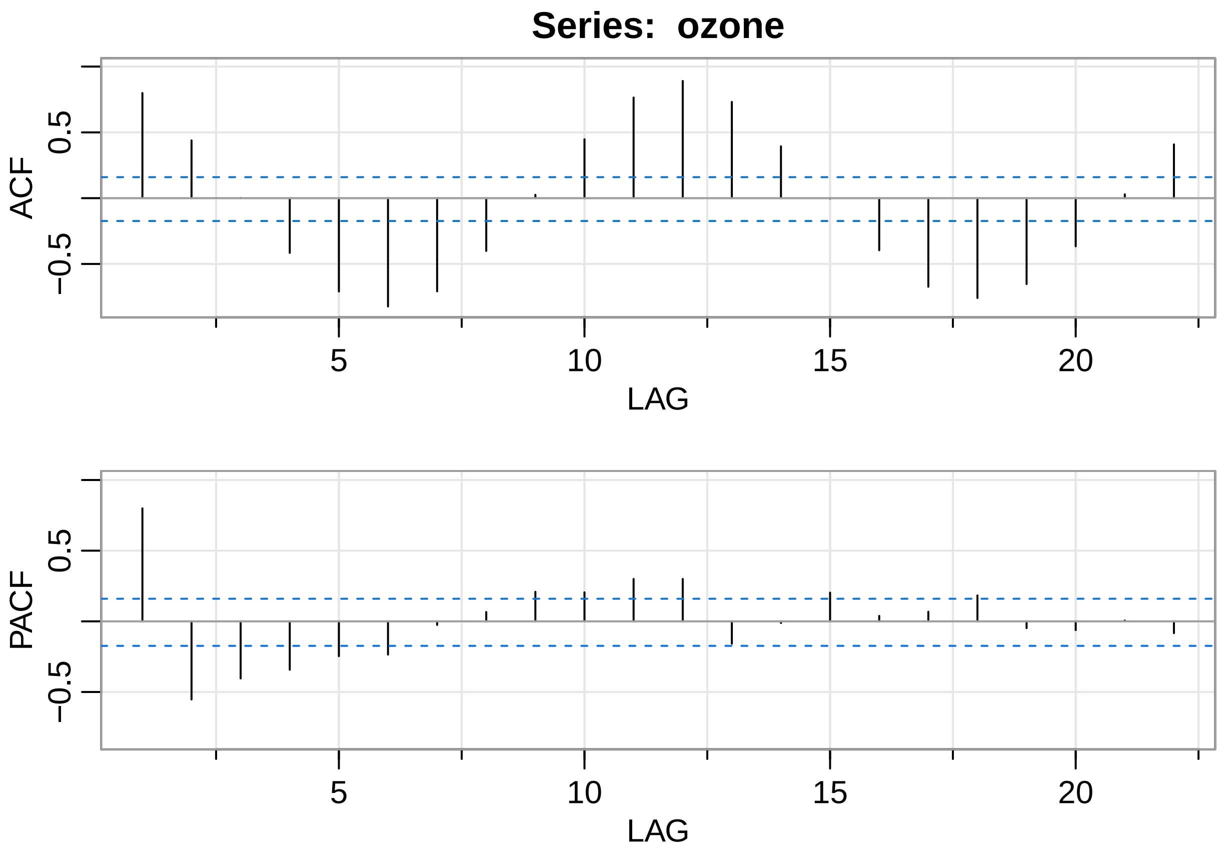
```
library(forecast)
library(astsa)
source('cleaning.R')
```

Deciding on signal model. Ignore all commented out code.

```
# raw
ozone <- phoenix %>%
  pull(o3)
plot(ozone, type = 'l')
```

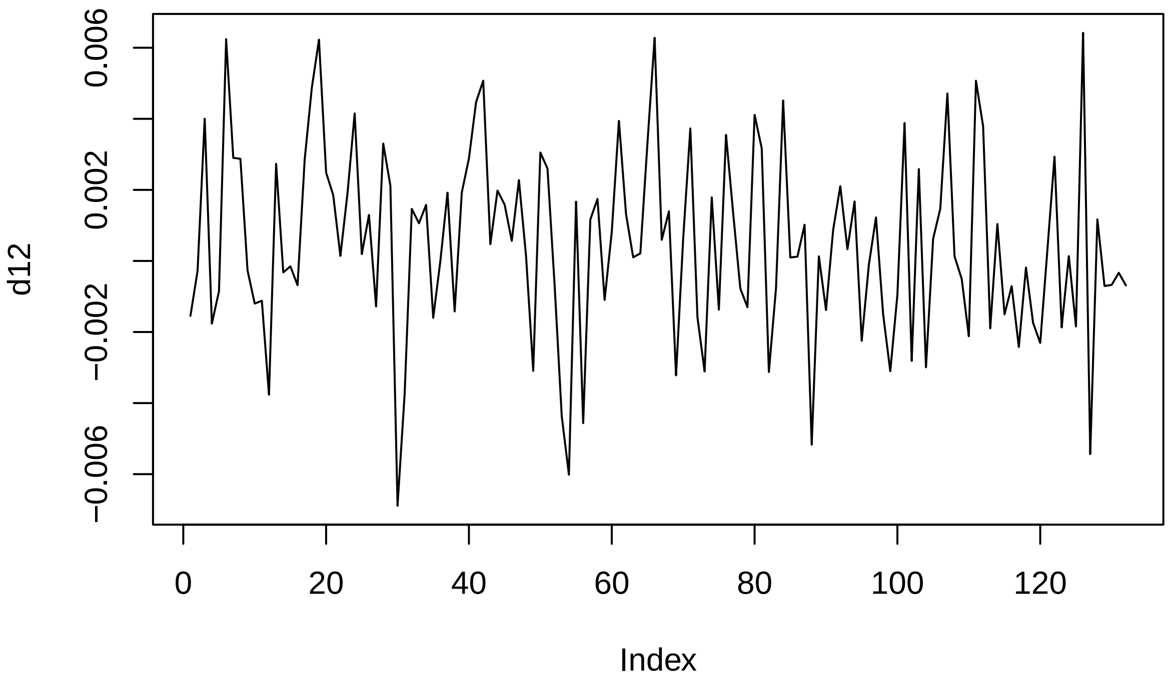



```
acf2(ozone)
```

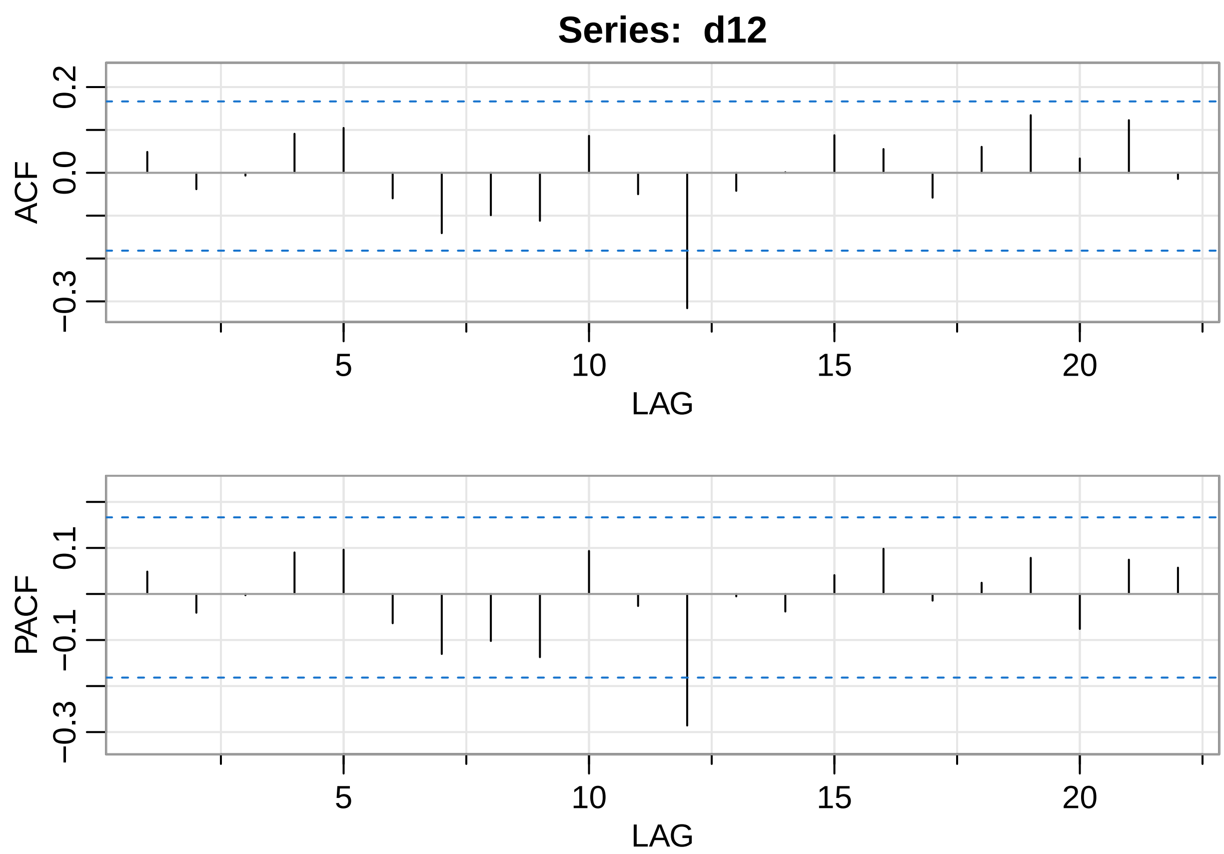



```
##      [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF   0.8  0.44  0.0 -0.42 -0.71 -0.82 -0.71 -0.40 0.03  0.45  0.76  0.89  0.73
## PACF  0.8 -0.55 -0.4 -0.34 -0.25 -0.24 -0.03  0.07 0.21  0.21  0.30  0.30 -0.16
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## ACF   0.39   0.0 -0.40 -0.67 -0.76 -0.65 -0.37  0.03  0.41
## PACF -0.01   0.2  0.04  0.07  0.18 -0.05 -0.06  0.01 -0.08
```

```
# yearly difference
d12 <- diff(ozone, lag = 12)
plot(d12, type = 'l')
```




```
acf2(d12)
```



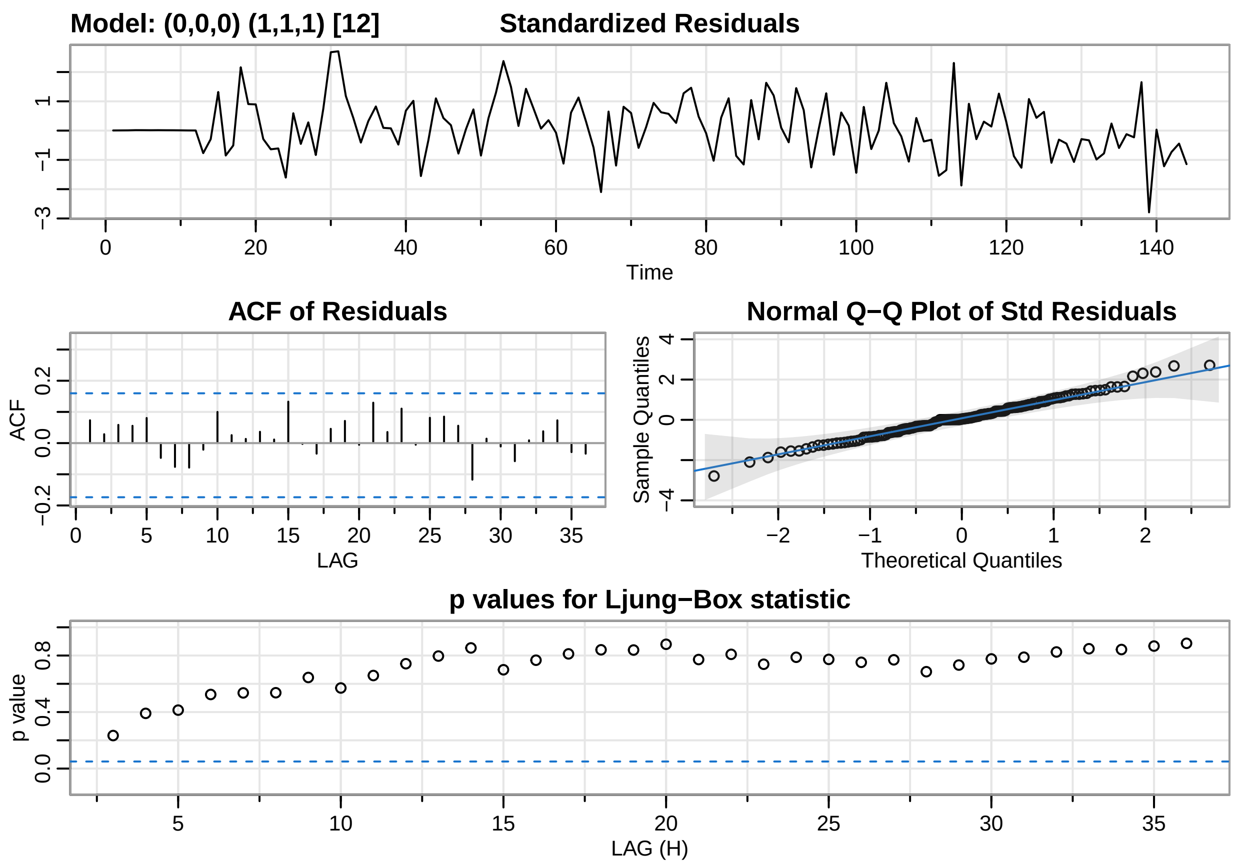

```
##      [,1]  [,2]  [,3] [,4] [,5]  [,6]  [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.05 -0.04 -0.01 0.09  0.1 -0.06 -0.14 -0.1 -0.11  0.09 -0.05 -0.32 -0.04
## PACF 0.05 -0.04  0.00 0.09  0.1 -0.06 -0.13 -0.1 -0.14  0.09 -0.03 -0.29 -0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## ACF   0.00  0.09  0.06 -0.06  0.06  0.13  0.03  0.12 -0.01
## PACF -0.04  0.04  0.10 -0.01  0.02  0.08 -0.08  0.07  0.06
```

```r
# yearly difference + weekly difference
#d7d365 <- diff(diff(ozone, lag = 365), lag = 7)
#plot(d7d365, type = 'l')
#acf2(d7d365)


# yearly diff + weekly diff + first diff
#d1d7d365 <- diff(diff(diff(ozone, lag = 365), lag = 7))
#plot(d1d7d365, type = 'l')
#acf2(d1d7d365)

# this is the best candidate!
```

Pursuing stationarity. These are the final non-parametric models. Once again ignored commented out code.

```r
# model 1: ARIMA(0,0,0)x(1,1,1)_12
model1 <- sarima(ozone, p = 0, d = 0, q = 0, P = 1, D = 1, Q = 1, S = 12)
```

```
## initial  value -5.947183
## iter   2 value -6.044153
## iter   3 value -6.065018
## iter   4 value -6.070275
## iter   5 value -6.077325
## iter   6 value -6.081873
## iter   7 value -6.083224
## iter   8 value -6.083306
## iter   9 value -6.083311
## iter  10 value -6.083312
## iter  11 value -6.083312
## iter  11 value -6.083312
## iter  11 value -6.083312
## final  value -6.083312
## converged
## initial  value -6.062969
## iter   2 value -6.067589
## iter   3 value -6.074012
## iter   4 value -6.077775
## iter   5 value -6.077906
## iter   6 value -6.077942
## iter   7 value -6.077954
## iter   8 value -6.077968
## iter   9 value -6.077968
## iter  10 value -6.077969
## iter  11 value -6.077969
## iter  12 value -6.077970
## iter  13 value -6.077971
## iter  14 value -6.077973
## iter  15 value -6.077973
## iter  15 value -6.077973
## final  value -6.077973
## converged
```




```
model1$AIC
```

```
## [1] -9.257464
```

```
model1$AICc
```

```
## [1] -9.256043
```

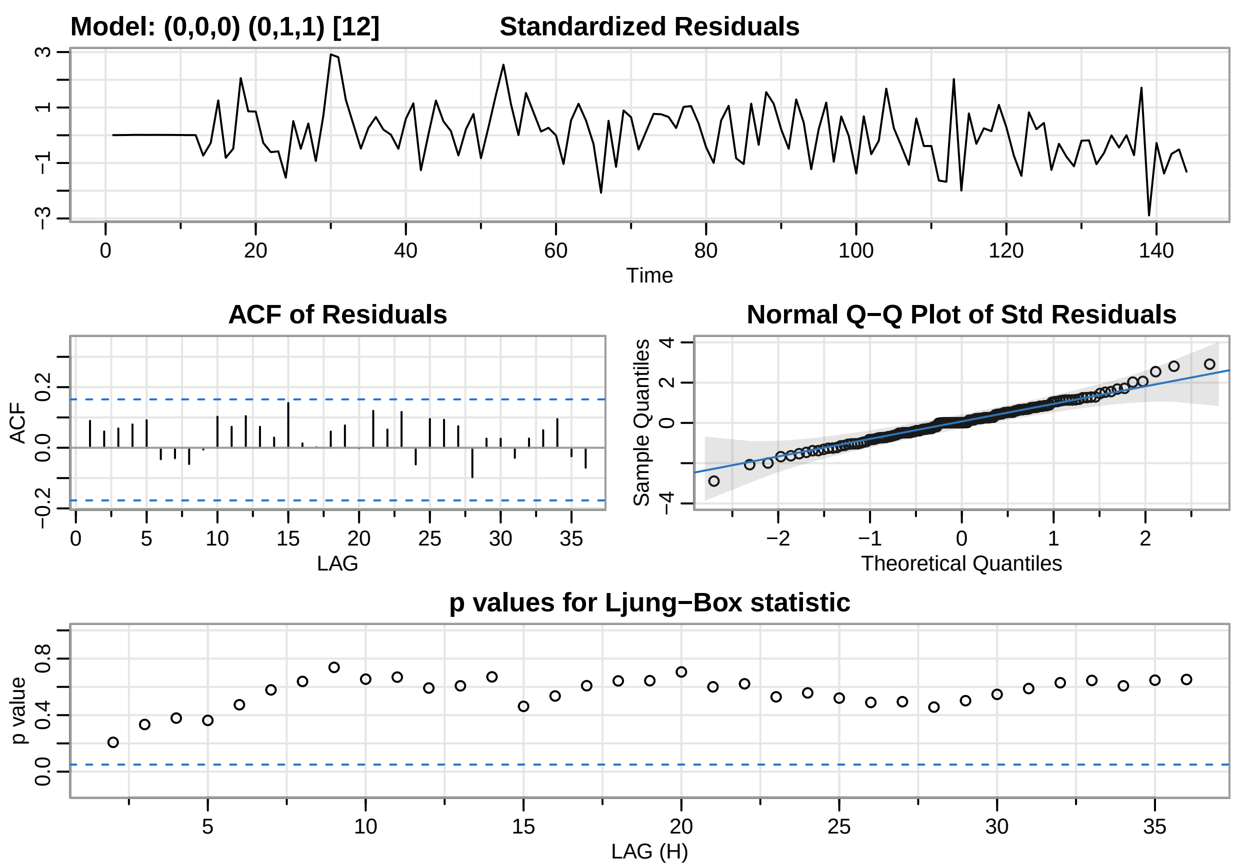
```
model1$BIC
```

```
## [1] -9.170106
```

```
# model 2: ARIMA(0,0,0)x(0,1,1)_12
model2 <- sarima(ozone, p = 0, d = 0, q = 0, P = 0, D = 1, Q = 1, S = 12)
```

```
## initial  value -5.941282
## iter   2 value -6.016678
## iter   3 value -6.038298
## iter   4 value -6.042036
## iter   5 value -6.043742
## iter   6 value -6.043900
## iter   7 value -6.043934
## iter   8 value -6.043938
```

```
## iter   9 value -6.043938
## iter   9 value -6.043938
## iter   9 value -6.043938
## final  value -6.043938
## converged
## initial  value -6.059838
## iter   2 value -6.063282
## iter   3 value -6.066532
## iter   4 value -6.067895
## iter   5 value -6.068211
## iter   6 value -6.068223
## iter   6 value -6.068223
## final  value -6.068223
## converged
```




```
model2$AIC
```

```
## [1] -9.253114
```

```
model2$AICc
```

```
## [1] -9.252409
```

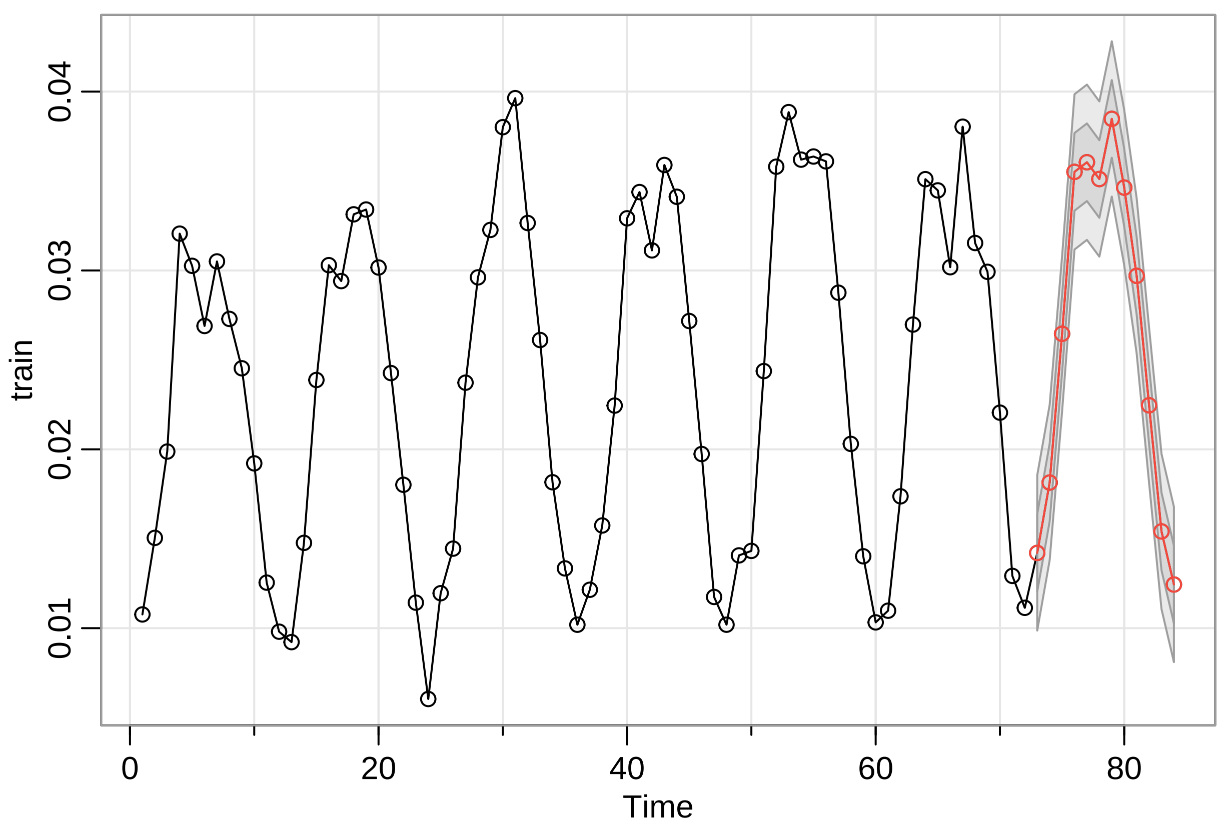
```r
model2$BIC
```

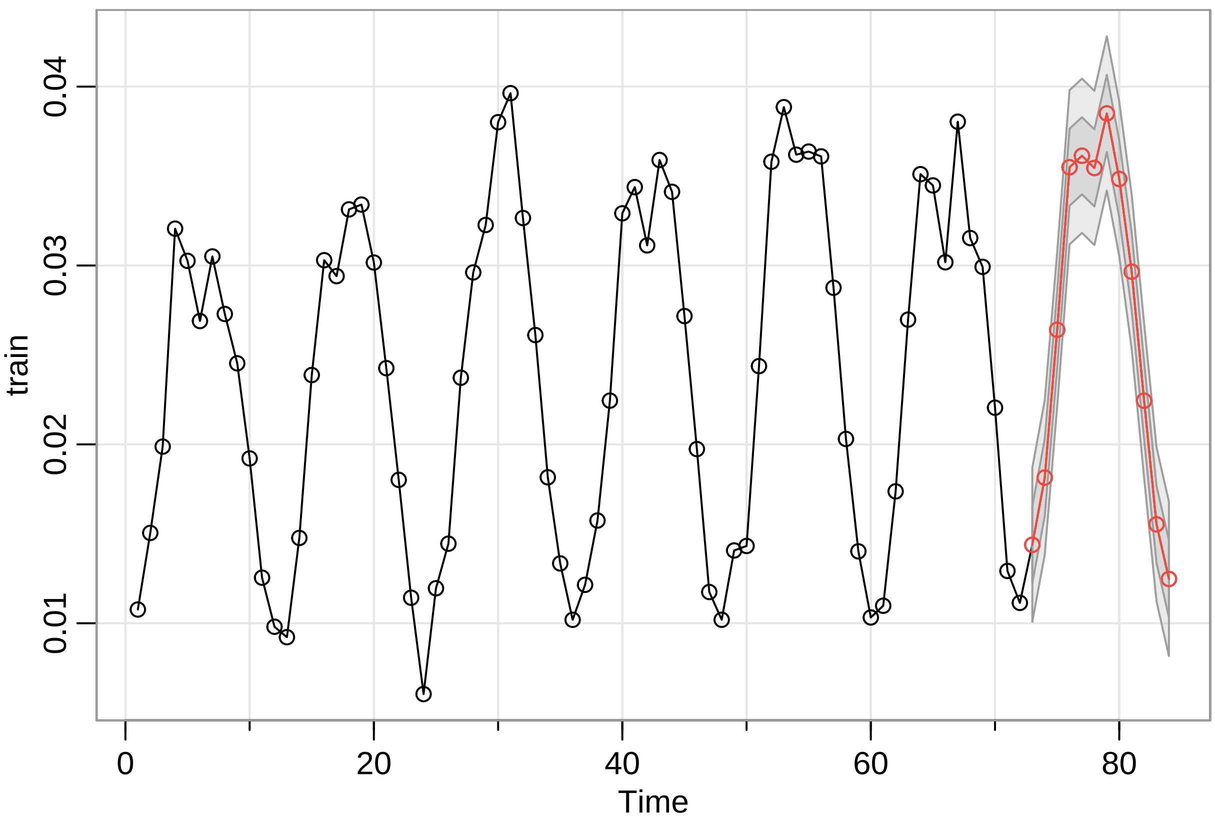
```
## [1] -9.187596
```

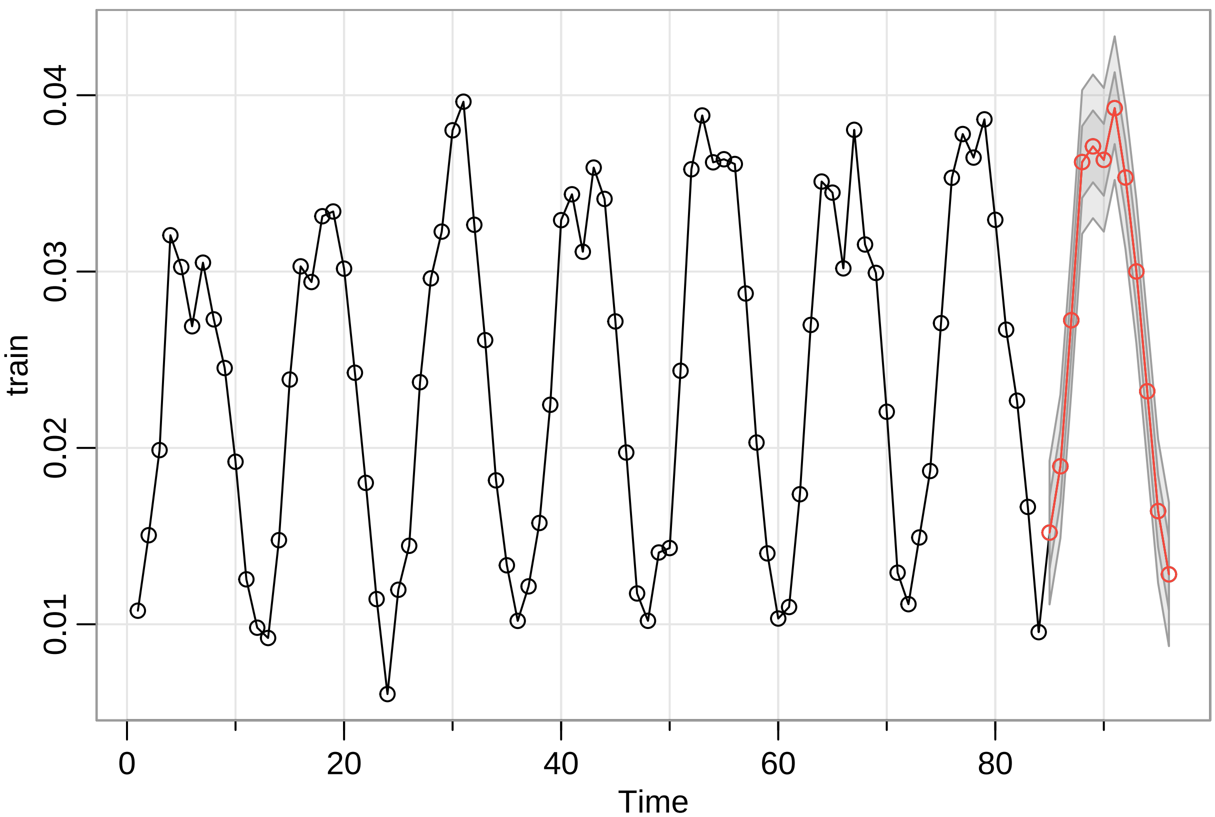
```r
# first SARIMA that passes Ljung-Box test!!
#sarima(d365, p = 3, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 7)


# final models
#model1 <- sarima(d365, p = 3, d = 0, q = 1, P = 1, D = 0, Q = 2, S = 7)
#model1$AIC
#model1$AICc
#model1$BIC

#model2 <- sarima(d365, p = 3, d = 0, q = 1, P = 1, D = 0, Q = 1, S = 7)
#model2$AIC
#model2$AICc
#model2$BIC

# auto.arima(d365) suggested model
#mtest <- auto.arima(d365, max.order = 7, stepwise = FALSE, approximation = FALSE, d = 0, D = 0)

#acf2(d365)
#sarima(d365, p = 3, d = 0, q = 1)


# others
#sarima(d365, p = 3, d = 0, q = 1, P = 1, D = 1, Q = 1, S = 7)
```

Cross-validation. When we decide on best model out of all 4, this step should be designed the same way as the CV for the parametric models.

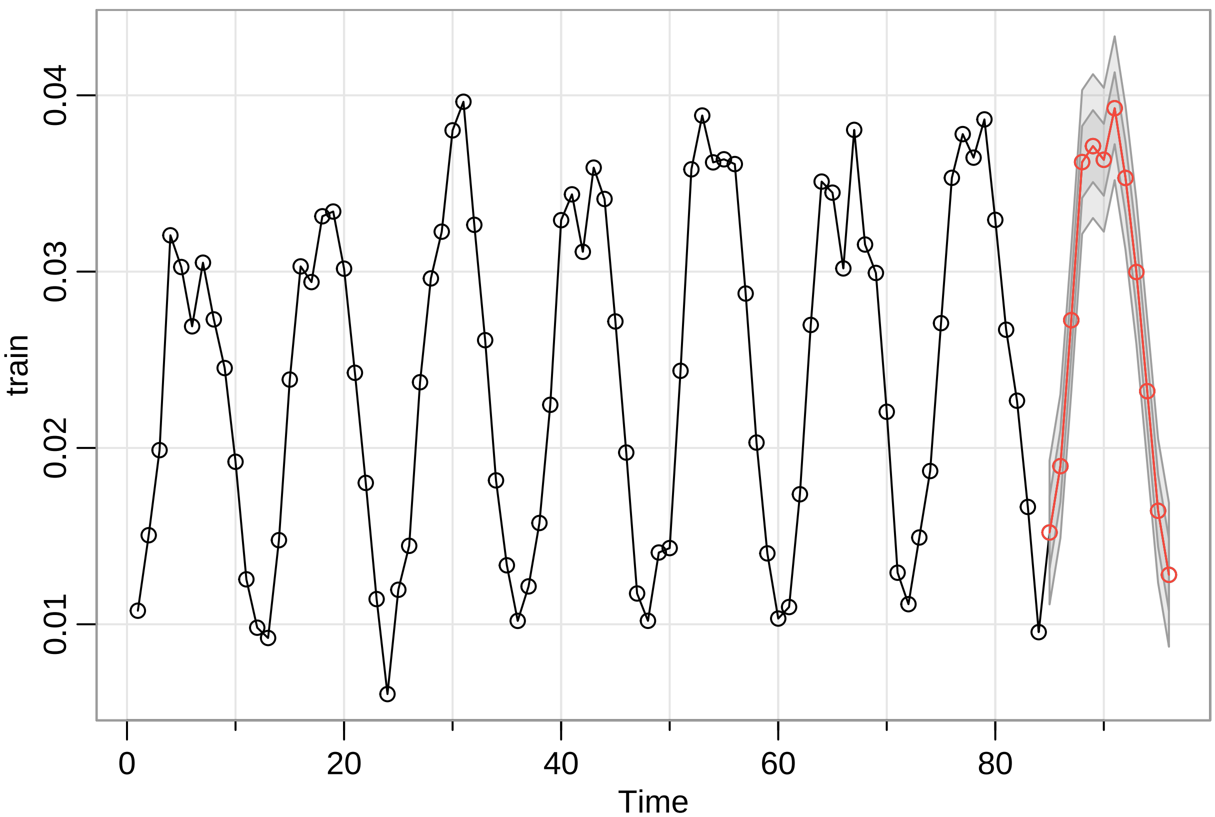
```
sse <- matrix(NA, nrow = 2, ncol = 6)  # 2 models, test on 2010, 2011, 2012, 2013, 2014, 2015

# train on 2004-2009, test on 2010-2015
for (i in 1:6) {
  train <- ozone[1:(12 * (i + 5))]
  test <- ozone[(12 * (i + 5) + 1):(12 * (i + 6))]

  m1 <- sarima.for(train, n.ahead = 12,  # 30
                   p = 0, d = 0, q = 0, P = 1, D = 1, Q = 1, S = 12)
  m2 <- sarima.for(train, n.ahead = length(test),
                   p = 0, d = 0, q = 0, P = 0, D = 1, Q = 1, S = 12)

  sse[1,i] <- sum((test - m1$pred)^2)
  sse[2,i] <- sum((test - m2$pred)^2)
}
```



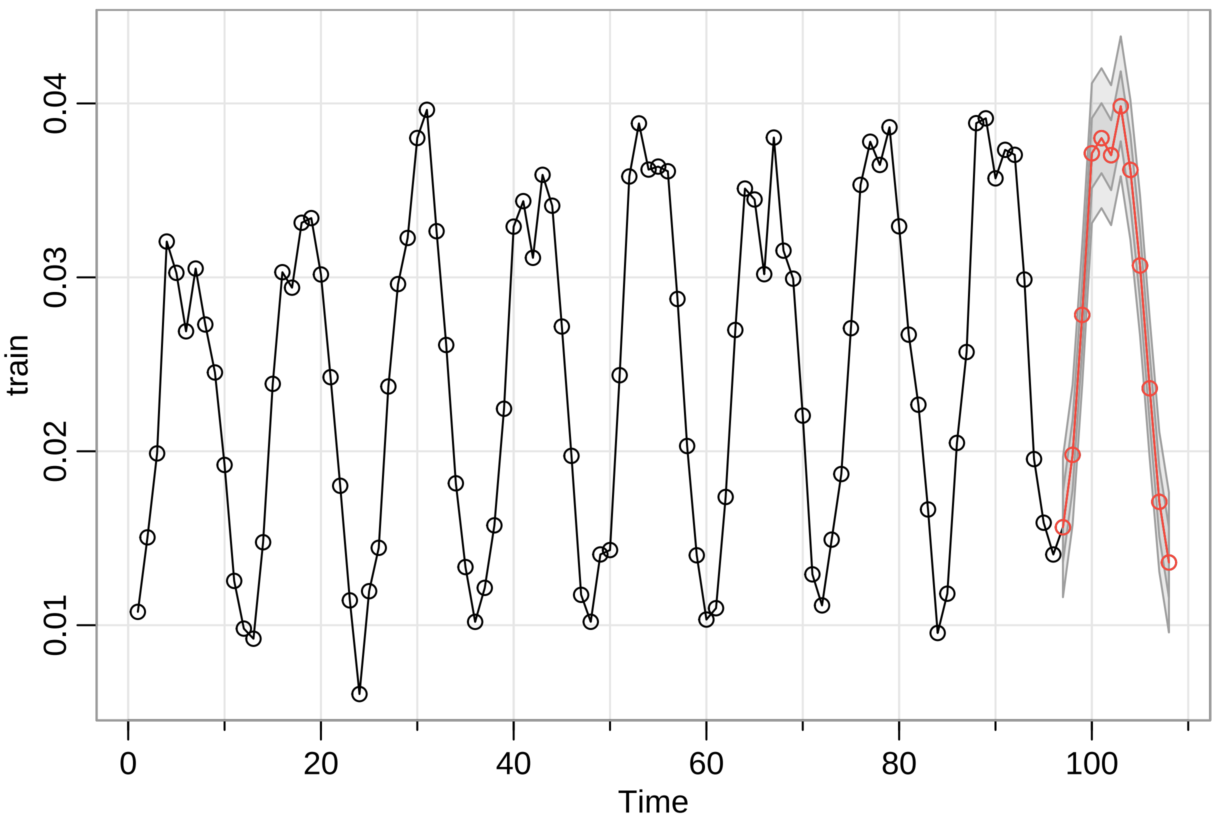

train
0.01
0.02
0.03
0.04
0
20
40
60
80
Time

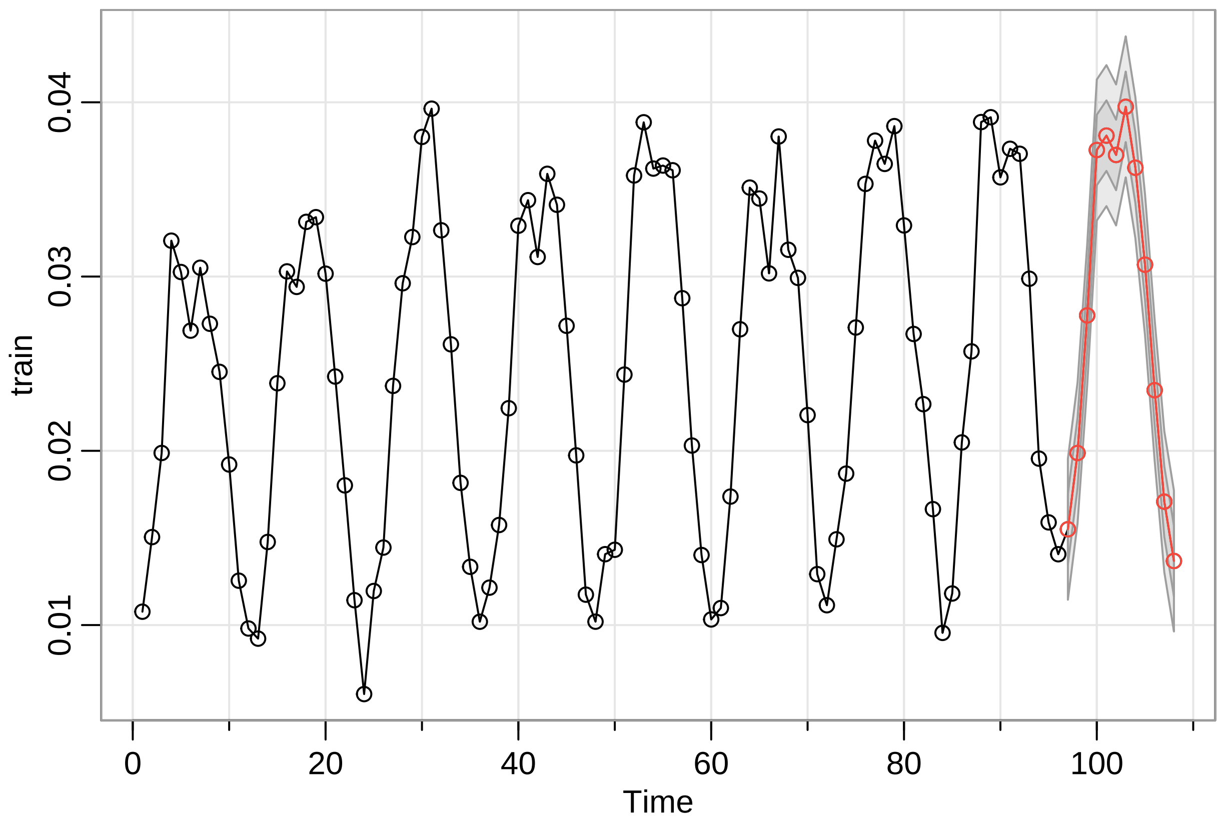


train
0.01
0.02
0.03
0.04
0
20
40
60
80
Time

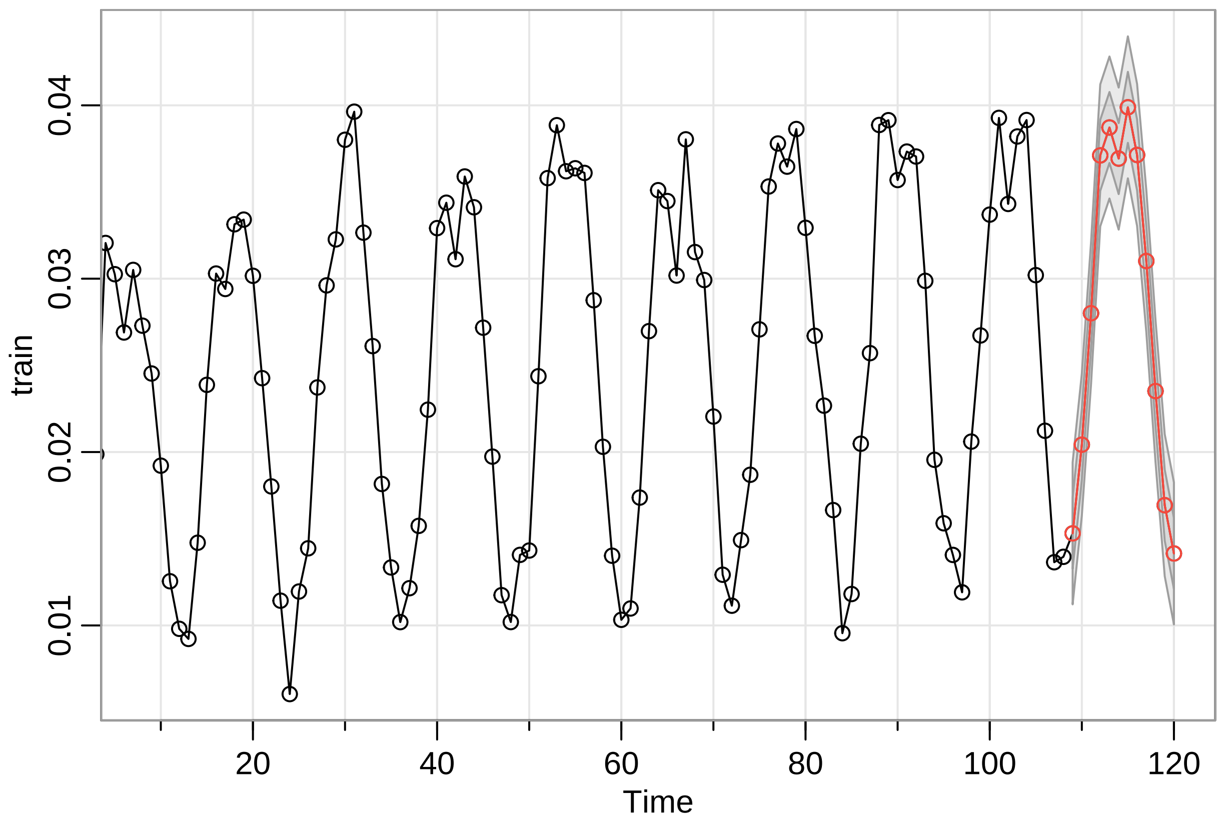


0.01
0.02
0.03
0.04
train
0
20
40
60
80
Time

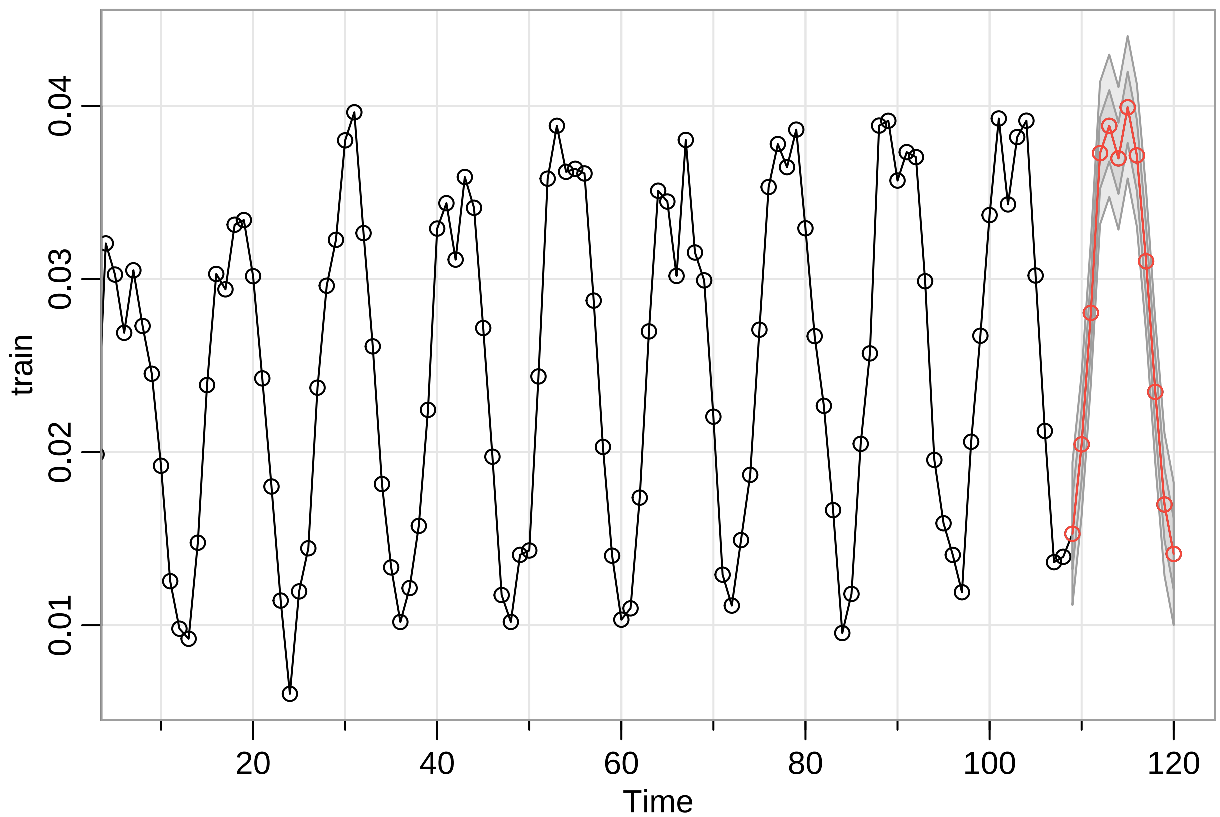

0.04
0.03
0.02
0.01
train
0
20
40
60
80
100
Time

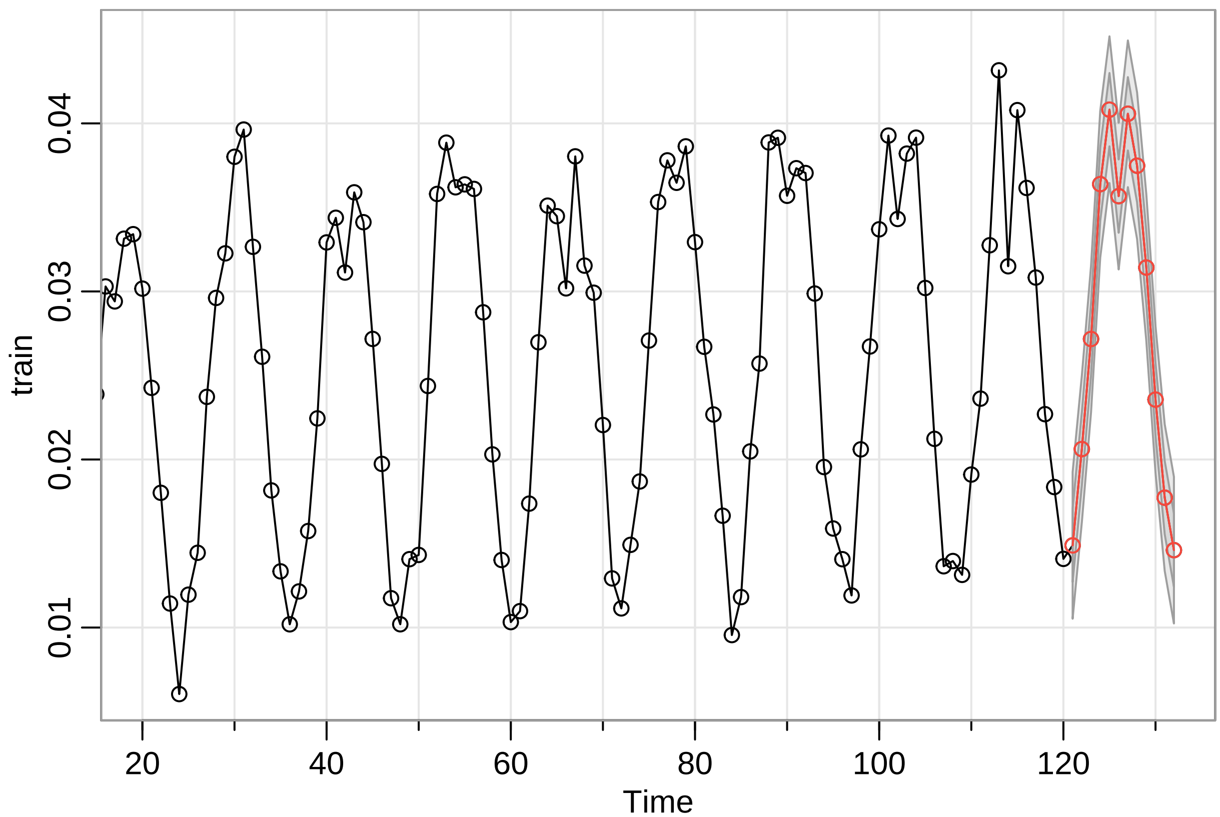


train
0.01
0.02
0.03
0.04
0
20
40
60
80
100
Time

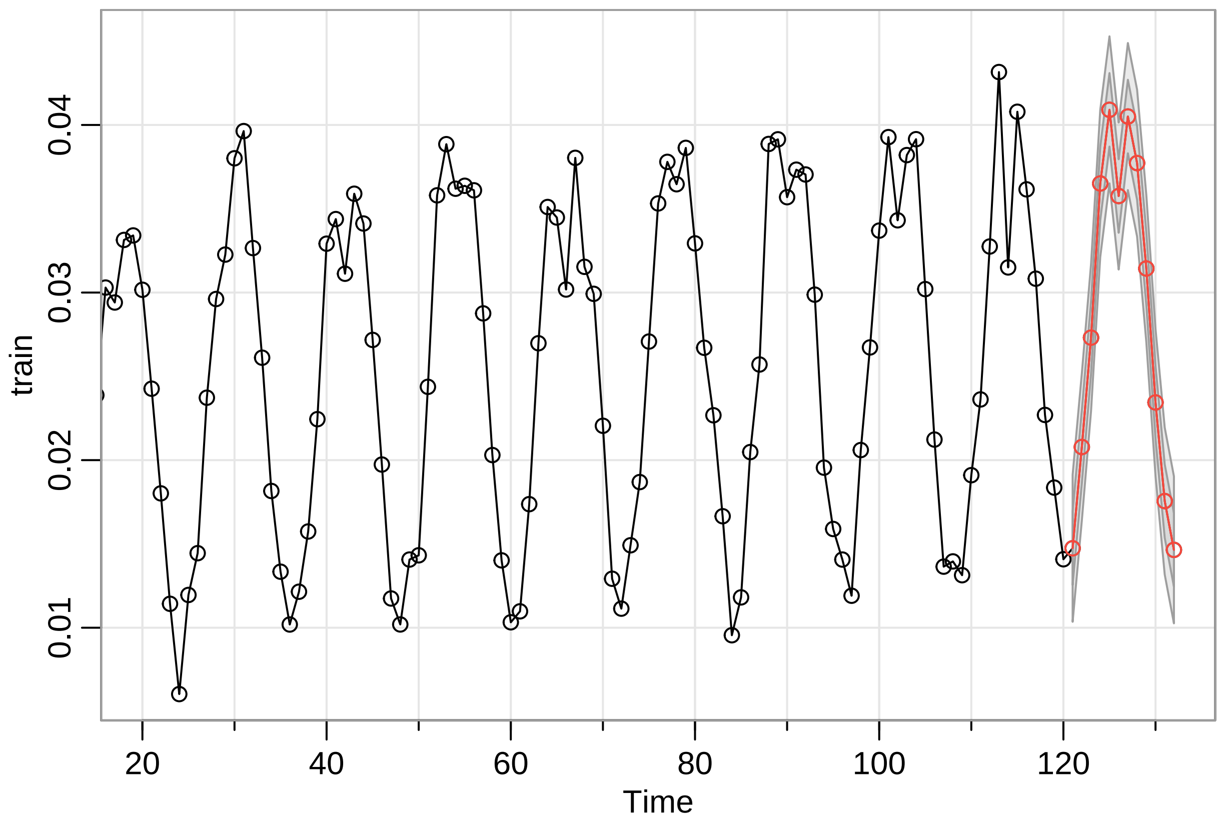


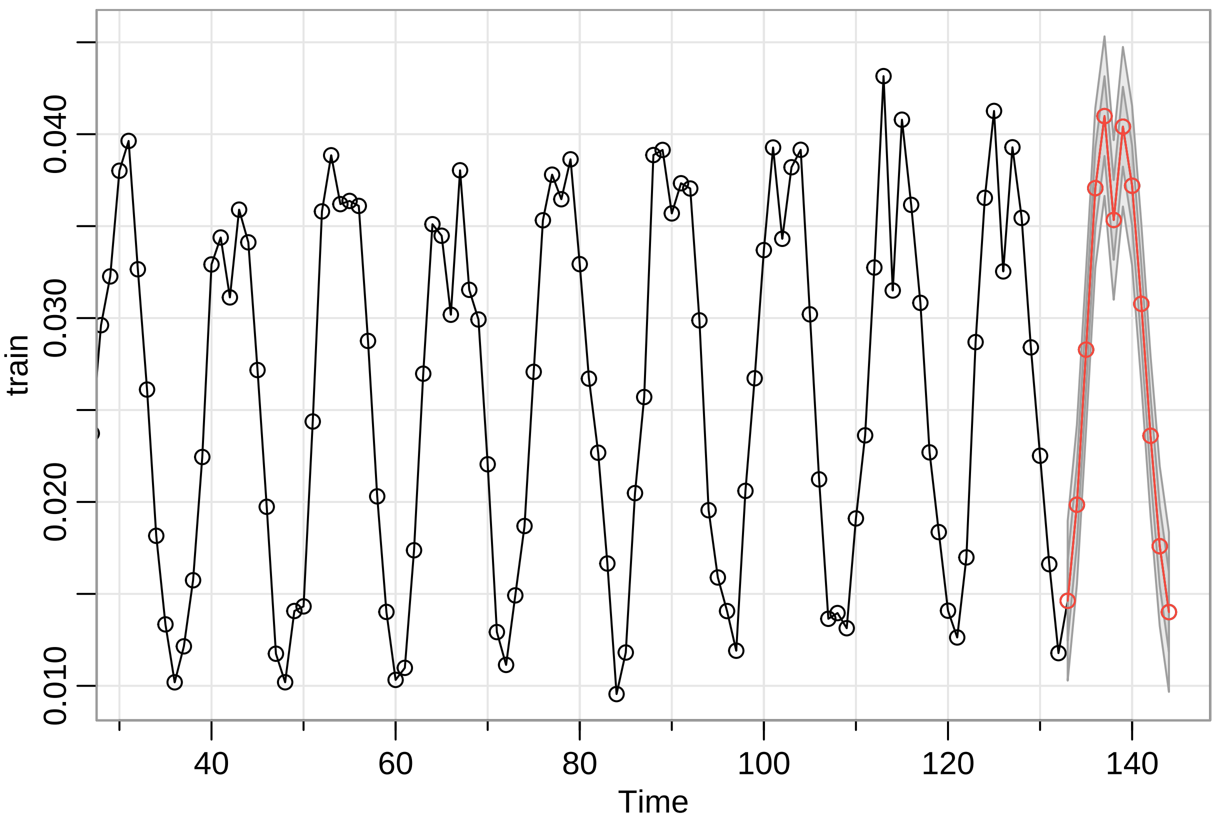
train
0.01
0.02
0.03
0.04
20
40
60
80
100
120
Time

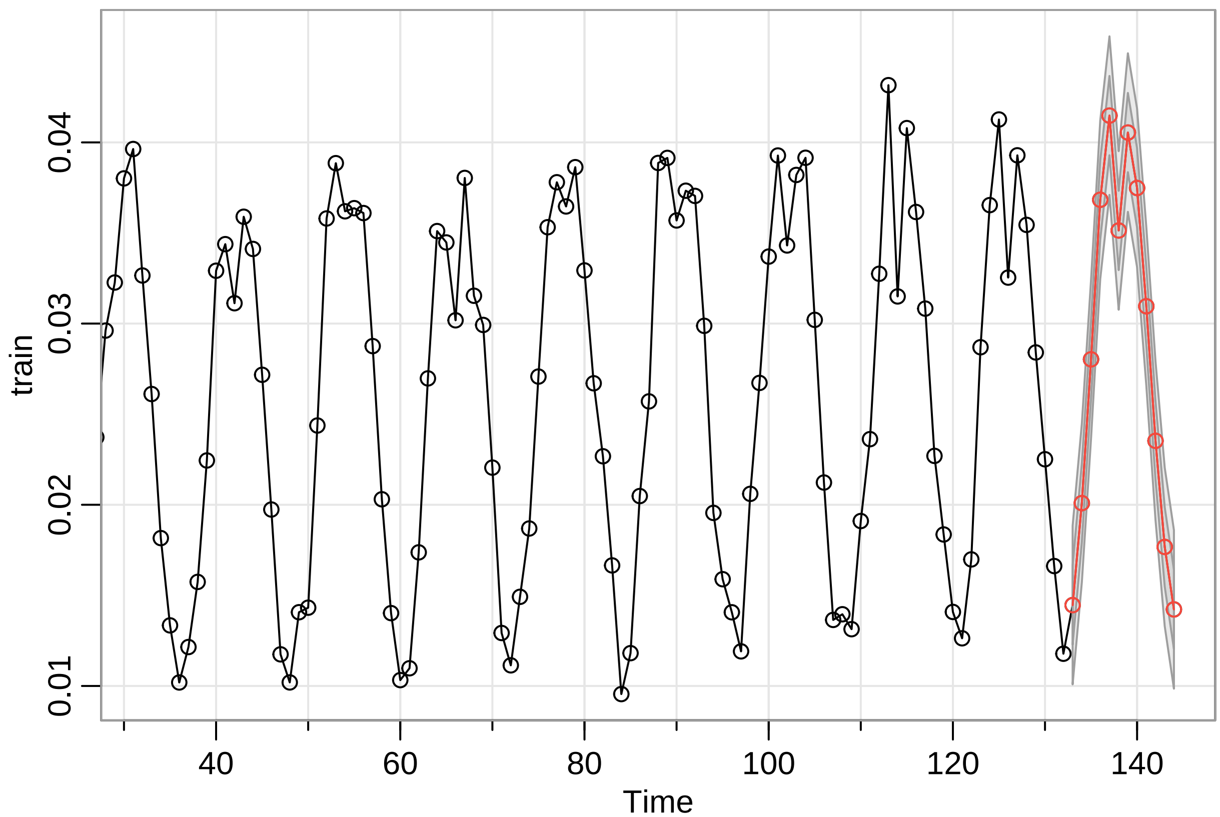

train
0.01
0.02
0.03
0.04
20
40
60
80
100
120
Time



train
0.01
0.02
0.03
0.04
20
40
60
80
100
120
Time


train
0.01
0.02
0.03
0.04
20
40
60
80
100
120
Time


train
0.010
0.020
0.030
0.040
40
60
80
100
120
140
Time




```r
# final SSEs
rowSums(sse, na.rm = TRUE)
```

```
## [1] 0.0003847380 0.0003905154
```

Notes from Tyler's OH (can ignore):

- look at acf of residuals and qq plot as well as box test
- x_t raw, v_t diff, v_t is the sarima
- auto.arima search parameters, eg. order