Project 2

Camryn Burley, Anna Haikl, and Joseph Keogh

12-4-19

Abstract

On our honor as students, we have neither given nor received aid on this assignment.

Load Data and Impute Missing Values

```
setwd(datadir)

airquality = read.csv('AirQualityUCI.csv')

# replace -200 with NA
airquality[airquality == -200] <- NA

# convert integer type to numeric
intcols = c(4,5,7,8,9,10,11,12)
for(i in 1:length(intcols)){
  airquality[,intcols[i]] <- as.numeric(airquality[,intcols[i]])
}

setwd(sourcedir)

# create new data frame with just CO and NO2
AQdata = airquality[,c(3,10)]

# impute missing air quality data
f <- ~ CO.GT. + NO2.GT.
t <- c(seq(1,dim(AQdata)[1],1))
i <- mnimput(f, AQdata, eps=1e-3, ts=TRUE, method='gam',
             ga.control=list(formula=paste(names(AQdata)[c(1:3)],'~ns(t,2)')))

# set airquality to imputed data
AQdata <- i$filled.dataset

# aggregate to daily maxima for model building
dailyAQ <- aggregate(AQdata, by=list(as.Date(airquality[,1],"%m/%d/%Y")), FUN=max)
```

Create Testing and Training Data

```
# the index to split the data on
separationIndex <- nrow(dailyAQ)-6

# split the data
air.train <- dailyAQ[1:separationIndex-1,]
```

```
air.test <- dailyAQ[separationIndex:nrow(dailyAQ),]

# making sure no data was duplicated or lost
a <- nrow(dailyAQ)
b <- nrow(air.train)
c <- nrow(air.test)

# a
# b
# c
```

Data Summary

```
summary(air.train)
```

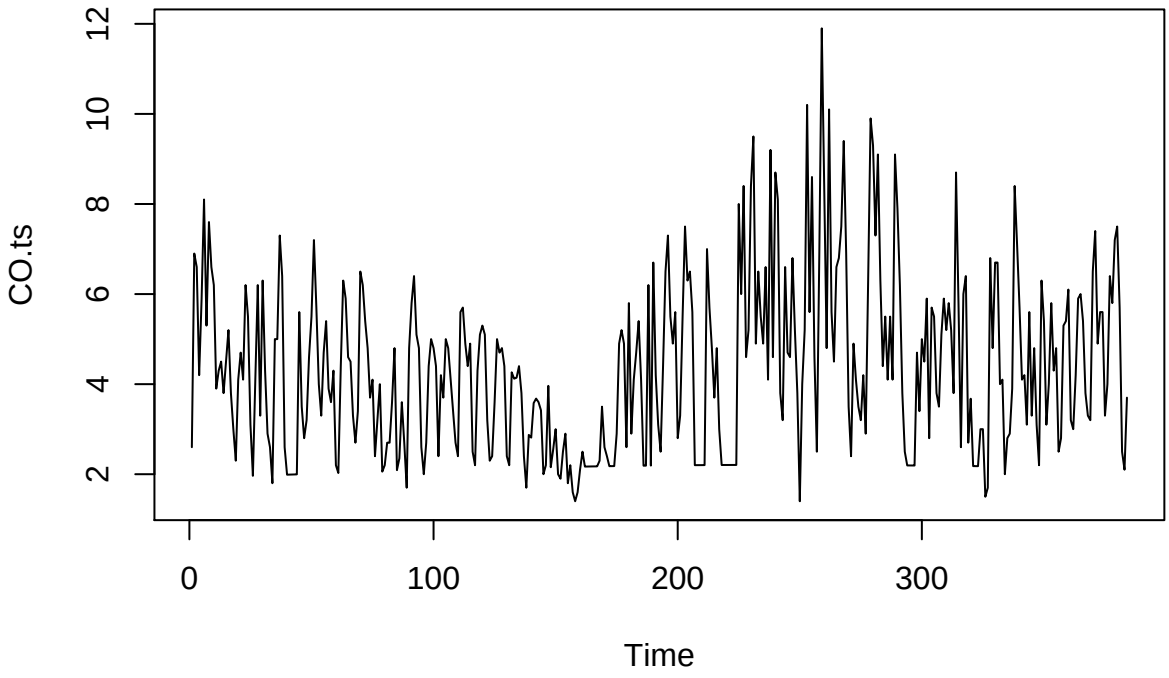
```
##     Group.1               CO.GT.          NO2.GT.
##  Min.   :2004-03-10   Min.   : 1.400   Min.   : 69.0
##  1st Qu.:2004-06-13   1st Qu.: 2.700   1st Qu.:124.6
##  Median :2004-09-17   Median : 4.132   Median :160.0
##  Mean   :2004-09-17   Mean   : 4.365   Mean   :164.5
##  3rd Qu.:2004-12-22   3rd Qu.: 5.600   3rd Qu.:196.2
##  Max.   :2005-03-28   Max.   :11.900   Max.   :340.0
```

The data is from March 2004 through April 2005. We are modeling the ambient daily maximum carbon monoxide (CO) and nitrogen dioxide (NO2) concentrations.

Univariate Time Series for CO

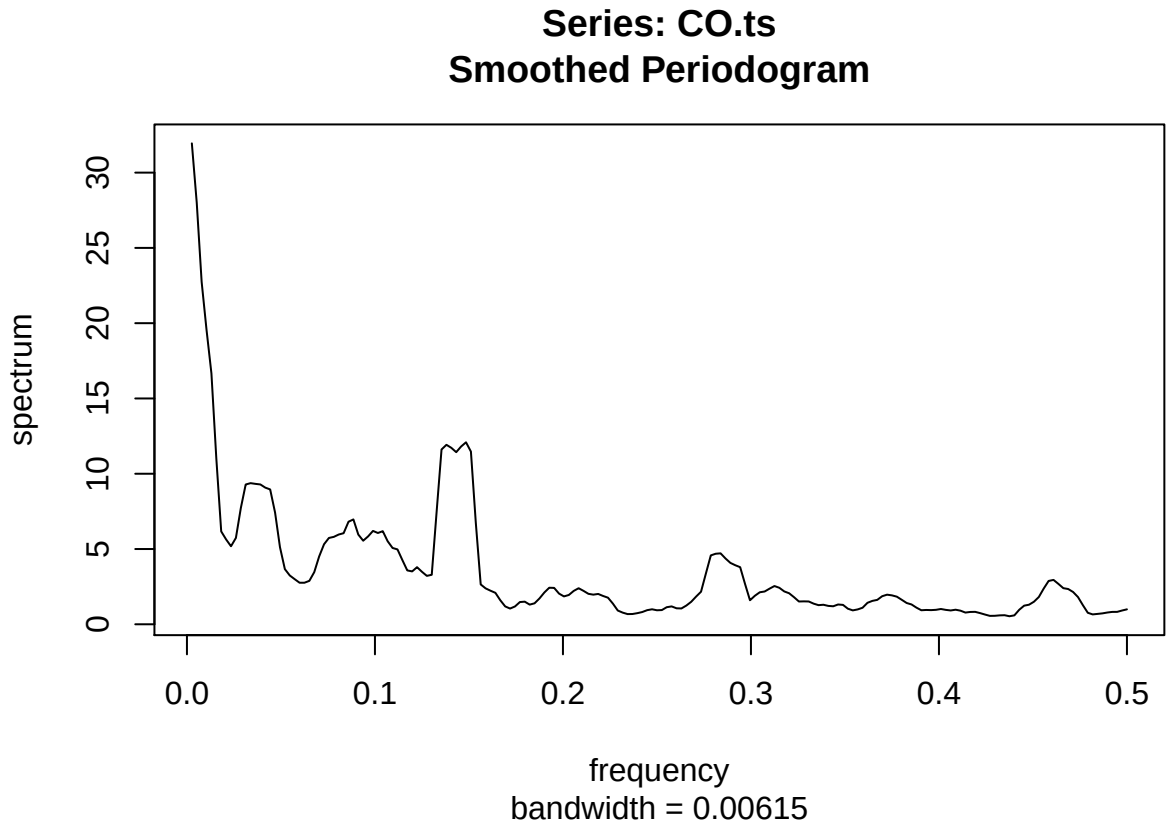
Visualize the Data

```
# create time series
CO.ts <- ts(air.train$CO.GT.)

# visualize raw data
plot(CO.ts)
```



There appears to be some trend from looking at the plot of the time series (i.e. the mean is not constant for the whole time series). There may be some seasonality, but we are unsure as of now if this fluctuation is from seasonality or if it is random.

```
# periodogram
pg.CO <- spec.pgram(CO.ts,spans=9,demean=T,log='no')
```



Based on the periodogram, there are multiple possibilities for the period of a seasonal component. There are a few spikes, each of which indicate a possible frequency/period to explain the seasonality of the data. We predict that if seasonality is significant in the model, it will be based on a complex wave. This is because there are many options for the period of potential seasons.

Seasonality

Finding Potential Periods

```
# sort the frequencies based on influence
sorted.spec <- sort(pg.CO$spec, decreasing=T, index.return=T)

# convert to periods
sorted.omegas <- pg.CO$freq[sorted.spec$ix]
sorted.Ts <- 1/pg.CO$freq[sorted.spec$ix]

# the cutoff for influential
CO.pg.cutoff <- 5

# the top periods
print('top periods')
```

```
## [1] "top periods"
```

```
sorted.Ts[1:CO.pg.cutoff]
```

```
## [1] 384.0 192.0 128.0  96.0  76.8
```

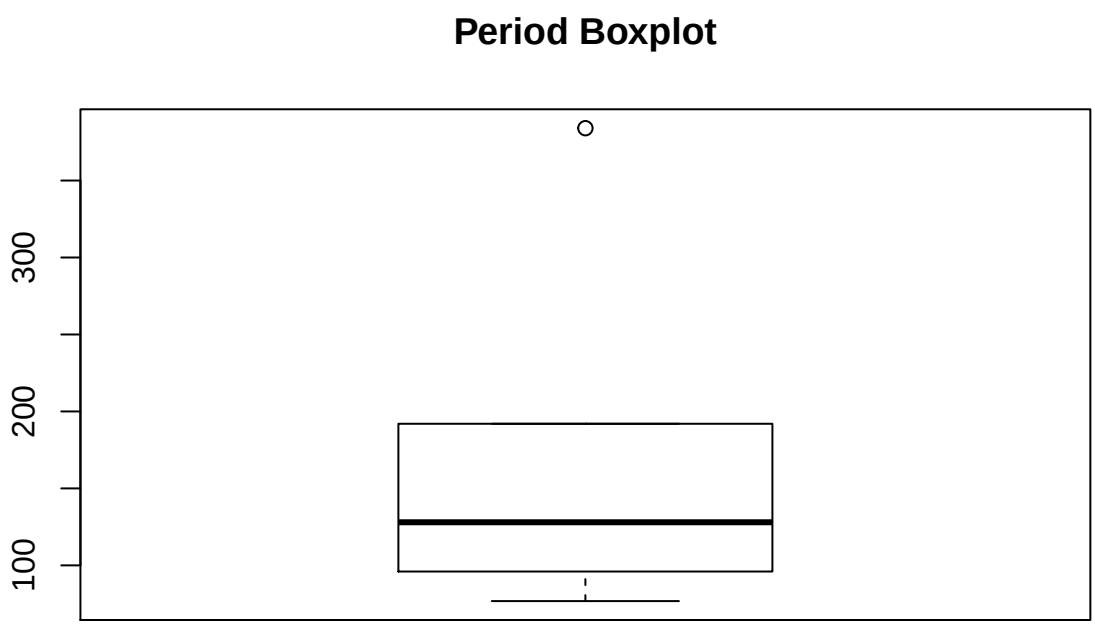
```
# top frequencies
## to double check that this makes sense based on periodogram
print('top frequencies')
```

```
## [1] "top frequencies"
```

```
sorted.omegas[1:CO.pg.cutoff]
```

```
## [1] 0.002604167 0.005208333 0.007812500 0.010416667 0.013020833
```

```
# visual
CO.pg.box <- boxplot(sorted.Ts[1:CO.pg.cutoff], main="Period Boxplot")
```



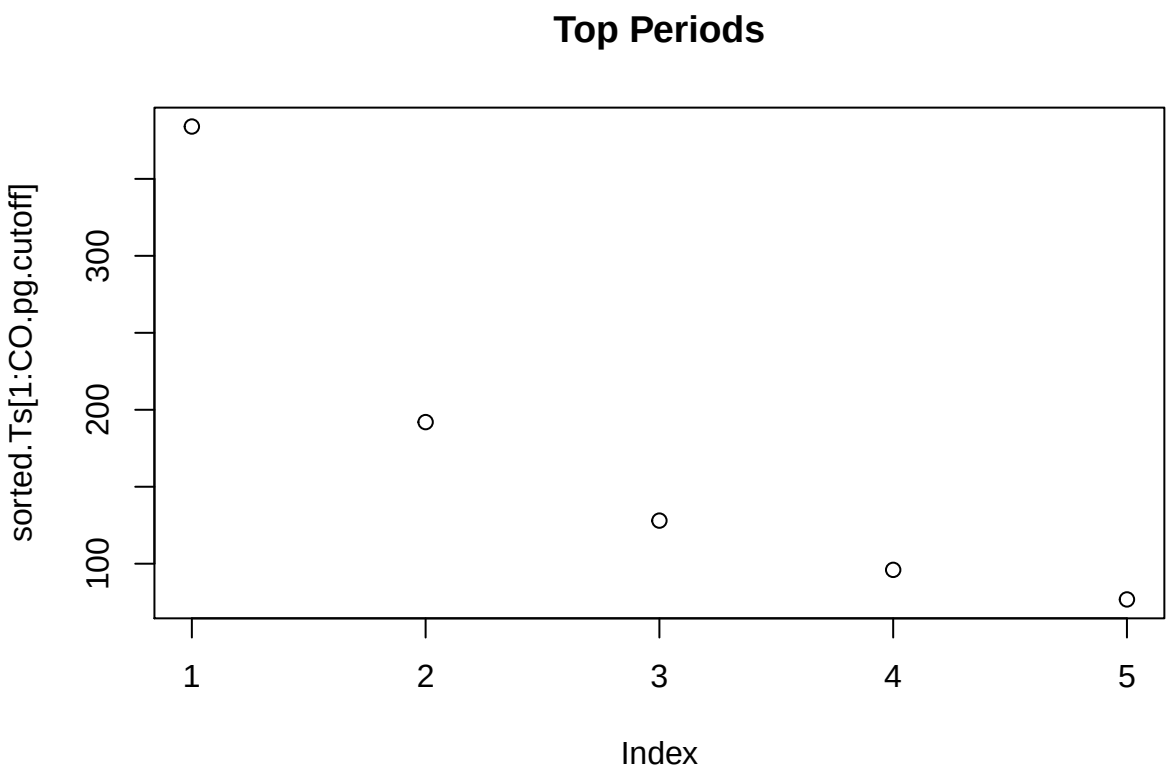

```
# the average influential period
print('mean of top periods')
```

```
## [1] "mean of top periods"
```

```
CO.pg.box.mean <- CO.pg.box$stats[3]
print(CO.pg.box.mean)
```

```
## [1] 128
```

```
# plot top periods
plot(sorted.Ts[1:CO.pg.cutoff], main = "Top Periods")
```



We found the frequencies of the largest spikes in the periodogram graph and converted them to periods by taking 1/the frequency. The five largest spikes correspond to periods of 384, 192, 128, 96, and 76.8 days. We considered these our "top" choices for periods to explain seasonality.
We are concerned that the reason we are seeing a period of 384 days is because this is close to the length of the data set and not because there is actual correlation between data values collected 384 days apart. We will investigate whether or not to use this period in the following sections.

Create Model with Potential Periods

To begin, we made a model without the 384 day period but that included the other four periods.

```
# assign potential periods to variables
CO.p1 <- sorted.Ts[1]
CO.p2 <- sorted.Ts[2]
CO.p3 <- sorted.Ts[3]
CO.p4 <- sorted.Ts[4]
```

```r
CO.p5 <- sorted.Ts[5]

# CO.p1
# CO.p2
# CO.p3
# CO.p4
# CO.p5

# create time variable
time.CO<-c(1:length(CO.ts))

# model
CO.lm.top4 <- lm(CO.ts ~ sin(2*pi*time.CO/CO.p2) +
                         cos(2*pi*time.CO/CO.p2) +
                         sin(2*pi*time.CO/CO.p3) +
                         cos(2*pi*time.CO/CO.p3) +
                         sin(2*pi*time.CO/CO.p4) +
                         cos(2*pi*time.CO/CO.p4)+
                         sin(2*pi*time.CO/CO.p5) +
                         cos(2*pi*time.CO/CO.p5))

# model summary
summary(CO.lm.top4)
```

```
##
## Call:
## lm(formula = CO.ts ~ sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2) + sin(2 * pi * time.CO/CO.p3) + cos(2 * pi *
##     time.CO/CO.p3) + sin(2 * pi * time.CO/CO.p4) + cos(2 * pi *
##     time.CO/CO.p4) + sin(2 * pi * time.CO/CO.p5) + cos(2 * pi *
##     time.CO/CO.p5))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9162 -1.3323 -0.2028  1.1502  6.4368
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.36457    0.09288  46.994  < 2e-16 ***
## sin(2 * pi * time.CO/CO.p2)  0.61552    0.13135   4.686  3.9e-06 ***
## cos(2 * pi * time.CO/CO.p2) -0.14310    0.13135  -1.089 0.276645
## sin(2 * pi * time.CO/CO.p3) -0.23858    0.13135  -1.816 0.070099 .
## cos(2 * pi * time.CO/CO.p3)  0.43888    0.13135   3.341 0.000917 ***
## sin(2 * pi * time.CO/CO.p4) -0.05762    0.13135  -0.439 0.661117
## cos(2 * pi * time.CO/CO.p4)  0.23999    0.13135   1.827 0.068470 .
## sin(2 * pi * time.CO/CO.p5) -0.01826    0.13135  -0.139 0.889533
## cos(2 * pi * time.CO/CO.p5) -0.21791    0.13135  -1.659 0.097949 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.82 on 375 degrees of freedom
## Multiple R-squared:  0.1048, Adjusted R-squared:  0.08573
## F-statistic: 5.489 on 8 and 375 DF,  p-value: 1.441e-06
```

Based on the model utility test, this model is significant at the 0.05 level, so we will continue to analyze its performance in comparison to other models. We will compare it to a model that only uses the first "important" period (i.e. the period associated with the largest spike in the periodogram, with the exception of the one corresponding to 384 days).

Compare Larger Model with Model with Only First Important Period

```
# actual model
CO.lm.top1 <- lm(CO.ts ~ sin(2*pi*time.CO/CO.p2) + cos(2*pi*time.CO/CO.p2))

# model summary
summary(CO.lm.top1)
```

```
##
## Call:
## lm(formula = CO.ts ~ sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5934 -1.5526 -0.1842  1.2392  6.9518
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.36457    0.09466  46.109  < 2e-16 ***
## sin(2 * pi * time.CO/CO.p2)  0.61552    0.13387   4.598 5.81e-06 ***
## cos(2 * pi * time.CO/CO.p2) -0.14310    0.13387  -1.069    0.286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.855 on 381 degrees of freedom
## Multiple R-squared:  0.05526,    Adjusted R-squared:  0.0503
## F-statistic: 11.14 on 2 and 381 DF,  p-value: 1.983e-05
```

```
# anova
anova(CO.lm.top1, CO.lm.top4)
```

```
## Analysis of Variance Table
##
## Model 1: CO.ts ~ sin(2 * pi * time.CO/CO.p2) + cos(2 * pi * time.CO/CO.p2)
## Model 2: CO.ts ~ sin(2 * pi * time.CO/CO.p2) + cos(2 * pi * time.CO/CO.p2) +
##     sin(2 * pi * time.CO/CO.p3) + cos(2 * pi * time.CO/CO.p3) +
##     sin(2 * pi * time.CO/CO.p4) + cos(2 * pi * time.CO/CO.p4) +
##     sin(2 * pi * time.CO/CO.p5) + cos(2 * pi * time.CO/CO.p5)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    381 1310.9
## 2    375 1242.1  6    68.788 3.4612 0.002429 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the partial F test is 0.0024, which is significant at the 0.05 level. We reject the null hypothesis, which means that the larger model contains at least one coefficient, not shared with the smaller model, that

is significant. As a result of this test, we have concluded that the larger model is better at explaining variability.

Create Model with All Identified Periods

```
# actual model
CO.lm.top5 <- lm(CO.ts ~ sin(2*pi*time.CO/CO.p1) +
                         cos(2*pi*time.CO/CO.p1) +
                         sin(2*pi*time.CO/CO.p2) +
                         cos(2*pi*time.CO/CO.p2) +
                         sin(2*pi*time.CO/CO.p3) +
                         cos(2*pi*time.CO/CO.p3) +
                         sin(2*pi*time.CO/CO.p4) +
                         cos(2*pi*time.CO/CO.p4)+
                         sin(2*pi*time.CO/CO.p5) +
                         cos(2*pi*time.CO/CO.p5))

# model summary
summary(CO.lm.top5)
```

```
##
## Call:
## lm(formula = CO.ts ~ sin(2 * pi * time.CO/CO.p1) + cos(2 * pi *
##     time.CO/CO.p1) + sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2) + sin(2 * pi * time.CO/CO.p3) + cos(2 * pi *
##     time.CO/CO.p3) + sin(2 * pi * time.CO/CO.p4) + cos(2 * pi *
##     time.CO/CO.p4) + sin(2 * pi * time.CO/CO.p5) + cos(2 * pi *
##     time.CO/CO.p5))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.482 -1.302 -0.115  1.082  5.783
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.36457    0.08767  49.785  < 2e-16 ***
## sin(2 * pi * time.CO/CO.p1) -0.83539    0.12398  -6.738 6.11e-11 ***
## cos(2 * pi * time.CO/CO.p1)  0.19483    0.12398   1.571 0.116926
## sin(2 * pi * time.CO/CO.p2)  0.61552    0.12398   4.965 1.05e-06 ***
## cos(2 * pi * time.CO/CO.p2) -0.14310    0.12398  -1.154 0.249164
## sin(2 * pi * time.CO/CO.p3) -0.23858    0.12398  -1.924 0.055072 .
## cos(2 * pi * time.CO/CO.p3)  0.43888    0.12398   3.540 0.000451 ***
## sin(2 * pi * time.CO/CO.p4) -0.05762    0.12398  -0.465 0.642361
## cos(2 * pi * time.CO/CO.p4)  0.23999    0.12398   1.936 0.053661 .
## sin(2 * pi * time.CO/CO.p5) -0.01826    0.12398  -0.147 0.883018
## cos(2 * pi * time.CO/CO.p5) -0.21791    0.12398  -1.758 0.079644 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.718 on 373 degrees of freedom
## Multiple R-squared:  0.2066, Adjusted R-squared:  0.1854
## F-statistic: 9.716 on 10 and 373 DF,  p-value: 1.876e-14
```

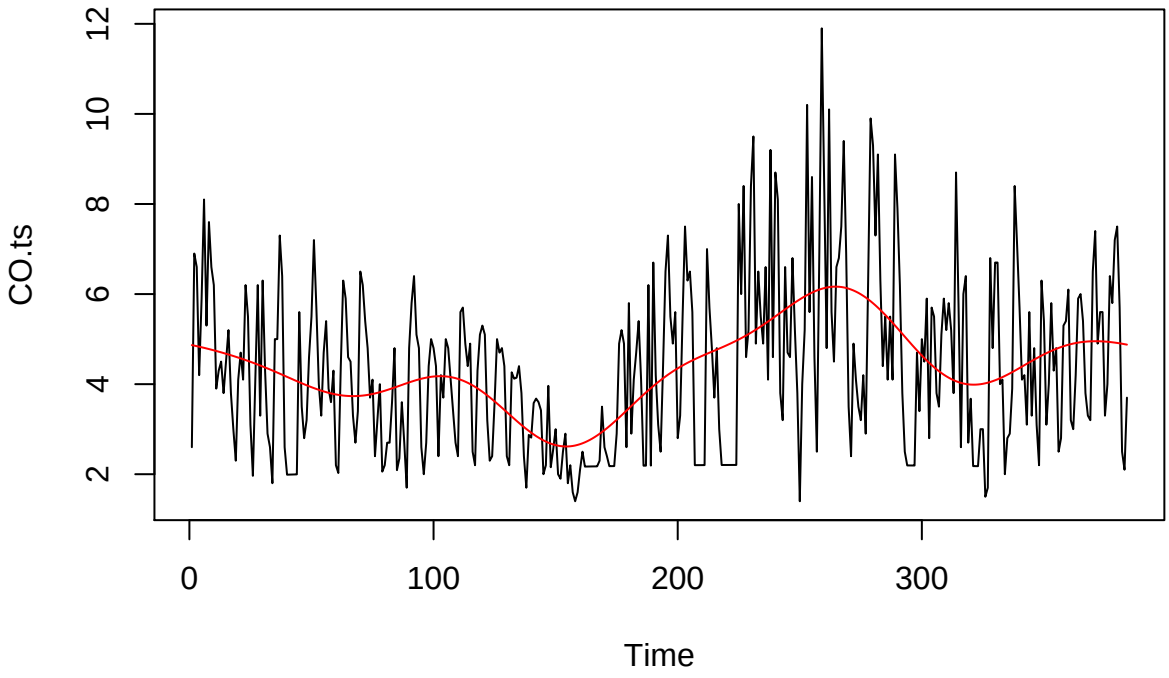
```r
# compare with smaller model
anova(CO.lm.top4, CO.lm.top5)
```

```
## Analysis of Variance Table
##
## Model 1: CO.ts ~ sin(2 * pi * time.CO/CO.p2) + cos(2 * pi * time.CO/CO.p2) +
##     sin(2 * pi * time.CO/CO.p3) + cos(2 * pi * time.CO/CO.p3) +
##     sin(2 * pi * time.CO/CO.p4) + cos(2 * pi * time.CO/CO.p4) +
##     sin(2 * pi * time.CO/CO.p5) + cos(2 * pi * time.CO/CO.p5)
## Model 2: CO.ts ~ sin(2 * pi * time.CO/CO.p1) + cos(2 * pi * time.CO/CO.p1) +
##     sin(2 * pi * time.CO/CO.p2) + cos(2 * pi * time.CO/CO.p2) +
##     sin(2 * pi * time.CO/CO.p3) + cos(2 * pi * time.CO/CO.p3) +
##     sin(2 * pi * time.CO/CO.p4) + cos(2 * pi * time.CO/CO.p4) +
##     sin(2 * pi * time.CO/CO.p5) + cos(2 * pi * time.CO/CO.p5)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    375 1242.1
## 2    373 1100.8  2    141.28 23.935 1.66e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

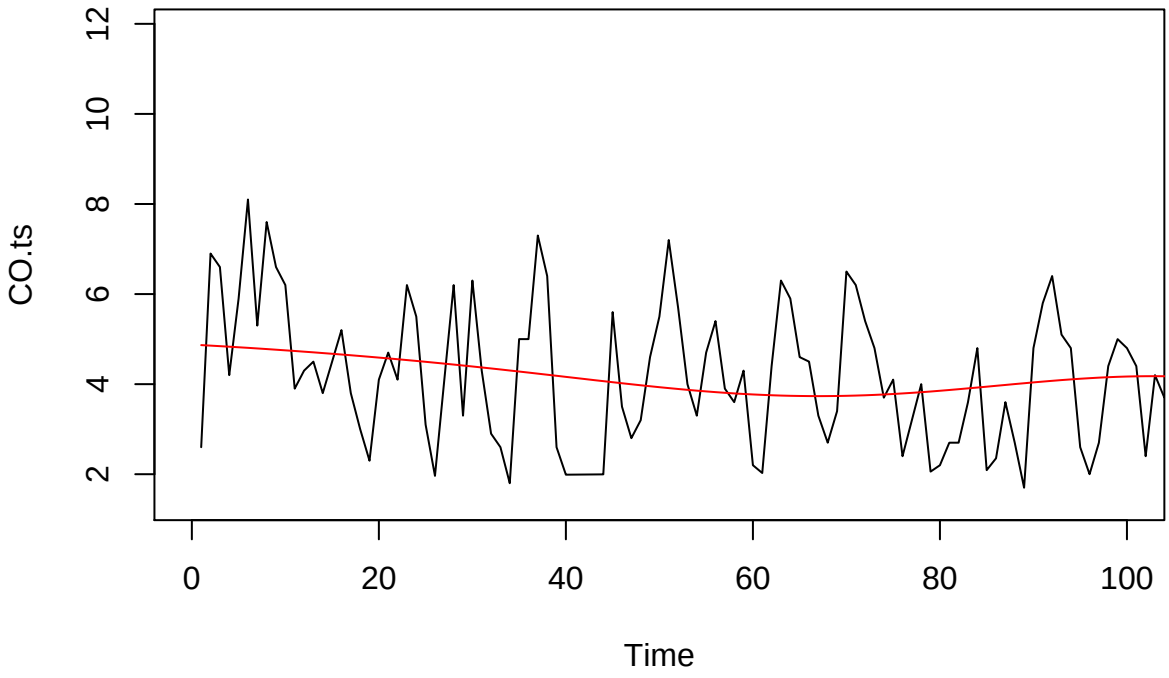
We created a model that used all five periods identified from the periodogram. We compared it to the model without the 384-day period using a partial F test. The test is significant at the 0.05 level, which means there is significant evidence to use the larger model. We will use the model with all five periods to explain seasonality in further analysis.

Visual Inspection of Model

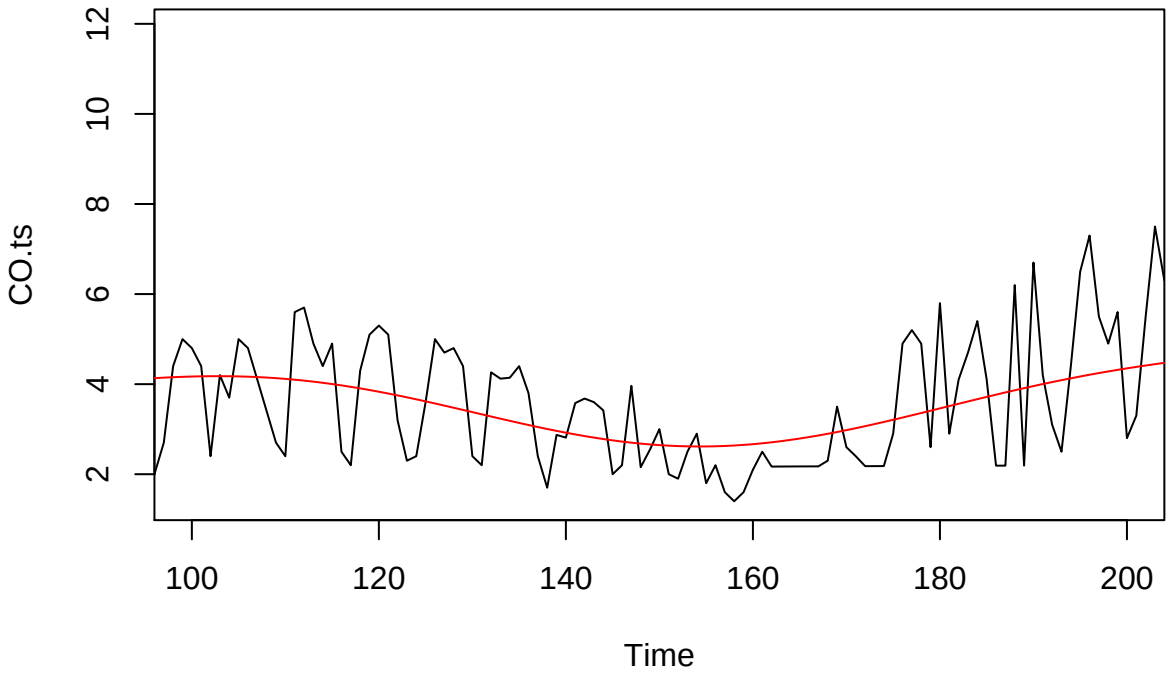
```r
plot(CO.ts)
lines(CO.lm.top5$fitted.values, col = "red")
```




```
plot(CO.ts, xlim=c(0,100))
lines(CO.lm.top5$fitted.values, col = "red")
```



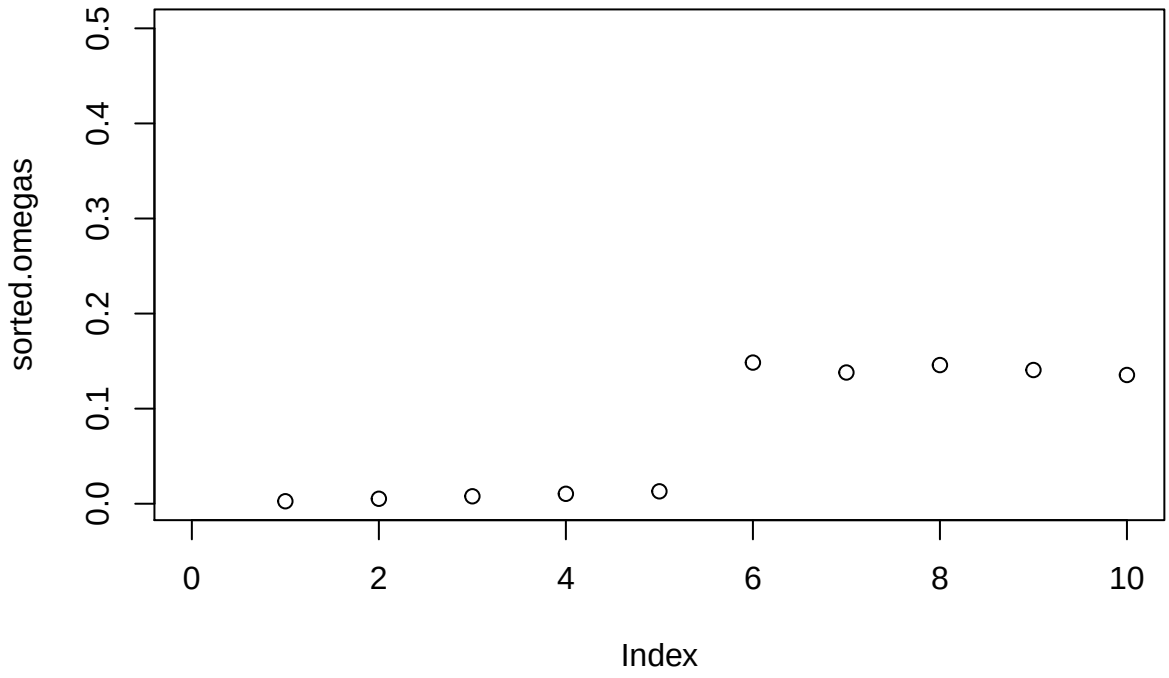

```
plot(CO.ts, xlim=c(100,200))
lines(CO.lm.top5$fitted.values, col = "red")
```



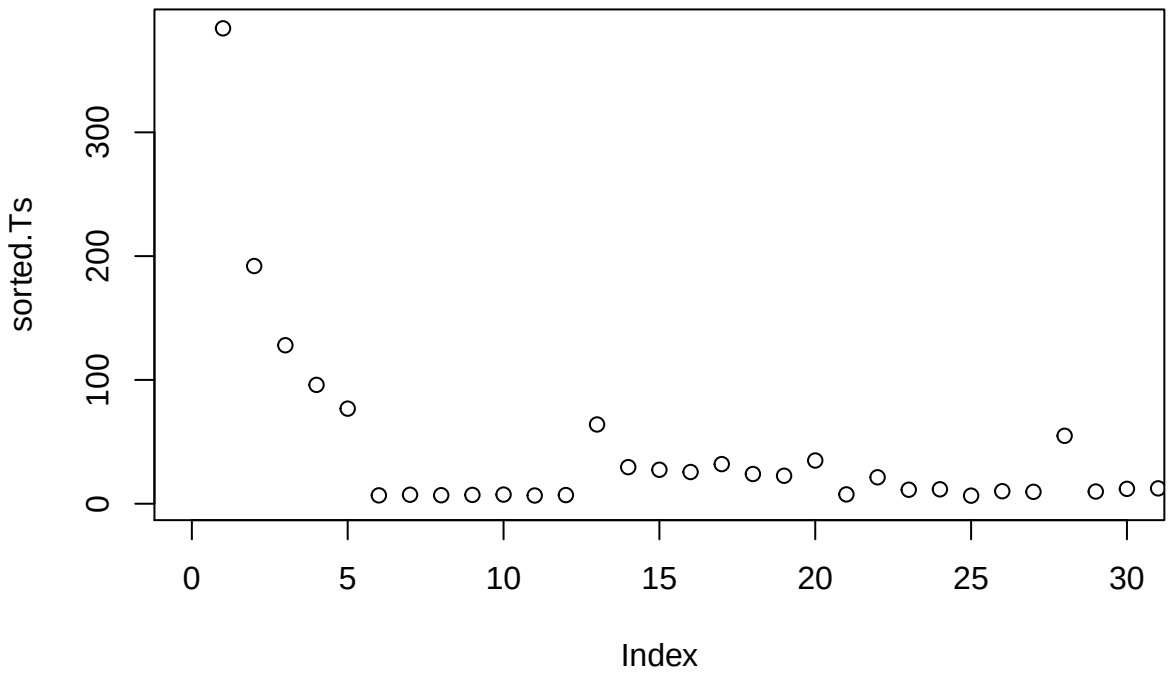
We plotted the fitted values of the model in red over the graph of the time series data. We plotted the entire time series and also zoomed in to see t = [0, 100] and t = [100, 200]. The model follows the general trend of the data, but there are no small fluctuations, of which the data has many. We may need to add a term with a smaller period in order to capture these smaller fluctuations.

Explore Adding Higher Frequency Components

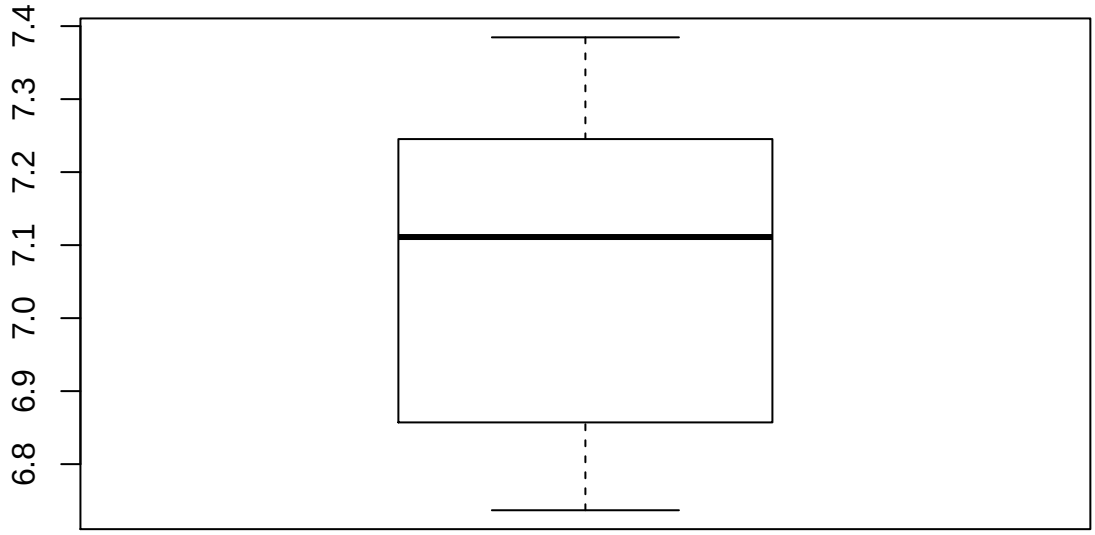
```
# visual for frequency
plot(sorted.omegas, xlim=c(0,10))
```




```
# visual for periods
plot(sorted.Ts, xlim=c(0,30))
```




```
# look at the 6th through 10th most influencial periods
next.low.period <- boxplot(sorted.Ts[6:10])$stats[3]
```




```
next.low.period
```

```
## [1] 7.111111
```

The average of the periods associated with the next five highest spikes of the periodogram is about 7 days (7.11). A period of around 7 days makes sense to us, as CO production could vary weekly from people working and commuting.

Add New Period to Model

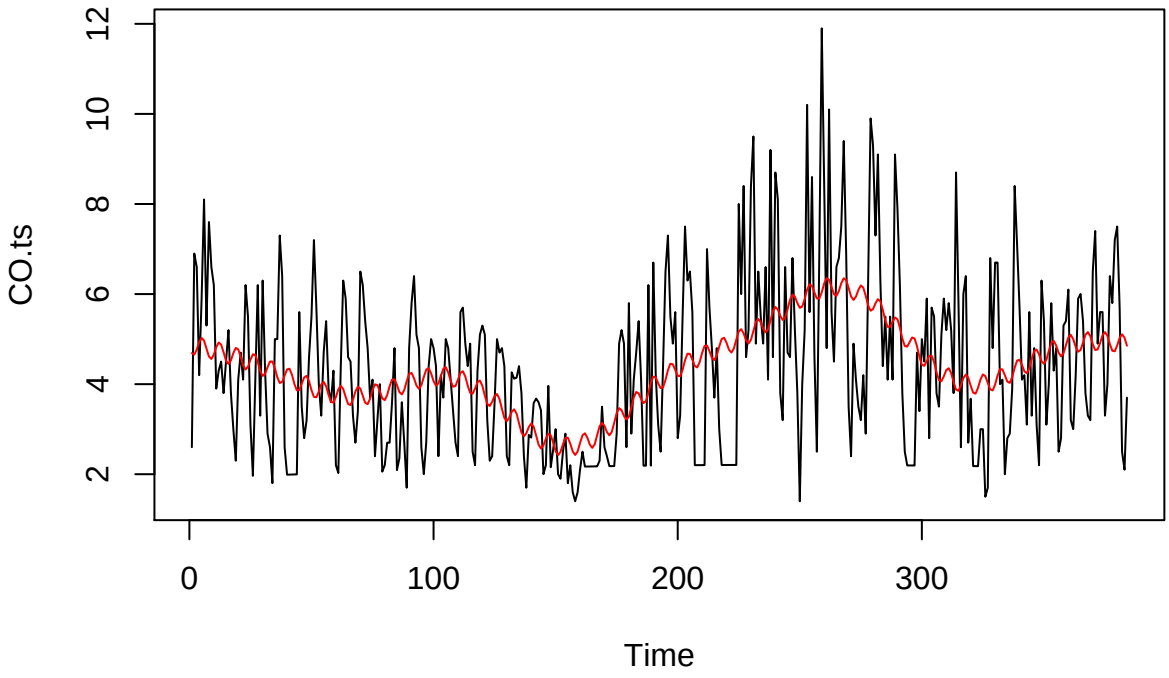
We made a new model with all five periods in our previous model with the addition of the new period of around 7 days.

```
CO.p6 <- next.low.period

# model
CO.lm.top6 <- lm(CO.ts ~ sin(2*pi*time.CO/CO.p1) +
                         cos(2*pi*time.CO/CO.p1) +
                         sin(2*pi*time.CO/CO.p2) +
                         cos(2*pi*time.CO/CO.p2) +
                         sin(2*pi*time.CO/CO.p3) +
                         cos(2*pi*time.CO/CO.p3) +
                         sin(2*pi*time.CO/CO.p4) +
                         cos(2*pi*time.CO/CO.p4) +
```

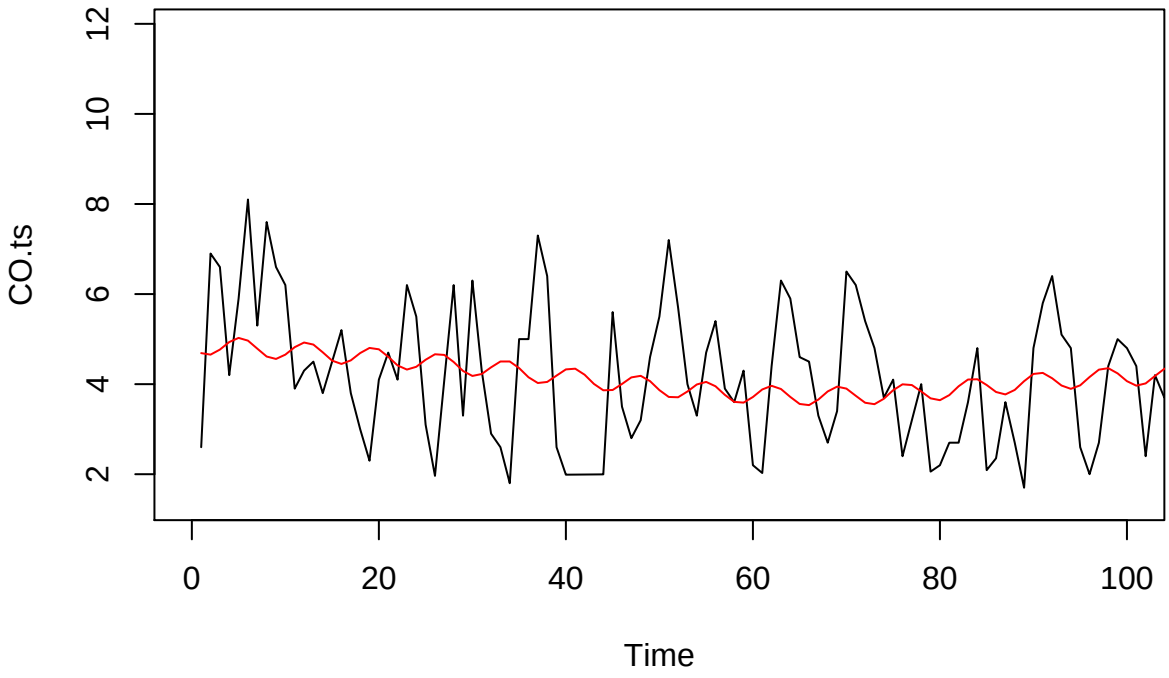
```
                        sin(2*pi*time.CO/CO.p5) +
                        cos(2*pi*time.CO/CO.p5) +
                        sin(2*pi*time.CO/CO.p6) +
                        cos(2*pi*time.CO/CO.p6))

# model summary
summary(CO.lm.top6)
```

```
##
## Call:
## lm(formula = CO.ts ~ sin(2 * pi * time.CO/CO.p1) + cos(2 * pi *
##     time.CO/CO.p1) + sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2) + sin(2 * pi * time.CO/CO.p3) + cos(2 * pi *
##     time.CO/CO.p3) + sin(2 * pi * time.CO/CO.p4) + cos(2 * pi *
##     time.CO/CO.p4) + sin(2 * pi * time.CO/CO.p5) + cos(2 * pi *
##     time.CO/CO.p5) + sin(2 * pi * time.CO/CO.p6) + cos(2 * pi *
##     time.CO/CO.p6))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2939 -1.2539 -0.1561  1.0615  5.8593
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.36457    0.08756  49.845  < 2e-16 ***
## sin(2 * pi * time.CO/CO.p1) -0.83539    0.12383  -6.746 5.85e-11 ***
## cos(2 * pi * time.CO/CO.p1)  0.19483    0.12383   1.573 0.116493
## sin(2 * pi * time.CO/CO.p2)  0.61552    0.12383   4.971 1.02e-06 ***
## cos(2 * pi * time.CO/CO.p2) -0.14310    0.12383  -1.156 0.248602
## sin(2 * pi * time.CO/CO.p3) -0.23858    0.12383  -1.927 0.054786 .
## cos(2 * pi * time.CO/CO.p3)  0.43888    0.12383   3.544 0.000444 ***
## sin(2 * pi * time.CO/CO.p4) -0.05762    0.12383  -0.465 0.641963
## cos(2 * pi * time.CO/CO.p4)  0.23999    0.12383   1.938 0.053380 .
## sin(2 * pi * time.CO/CO.p5) -0.01826    0.12383  -0.147 0.882879
## cos(2 * pi * time.CO/CO.p5) -0.21791    0.12383  -1.760 0.079289 .
## sin(2 * pi * time.CO/CO.p6) -0.20923    0.12383  -1.690 0.091946 .
## cos(2 * pi * time.CO/CO.p6) -0.02502    0.12383  -0.202 0.839997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.716 on 371 degrees of freedom
## Multiple R-squared:  0.2128, Adjusted R-squared:  0.1873
## F-statistic: 8.357 on 12 and 371 DF,  p-value: 5.083e-14
```

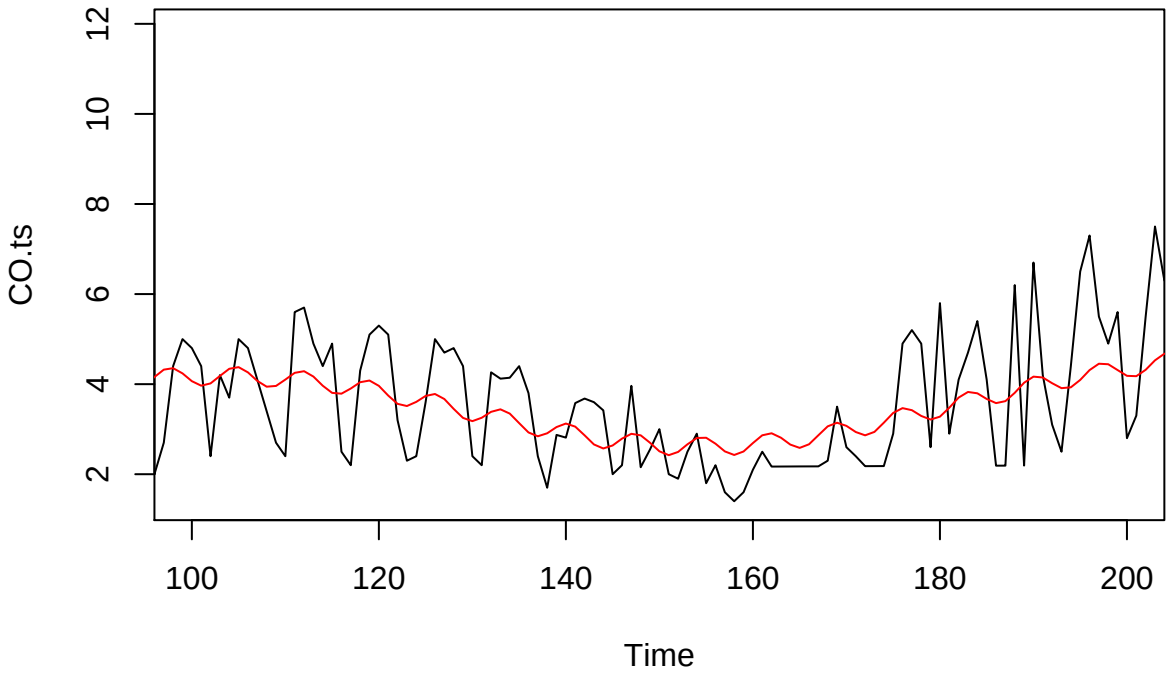
```
# visual inspection
plot(CO.ts)
lines(CO.lm.top6$fitted.values, col = "red")
```




```
plot(CO.ts, xlim=c(0,100))
lines(CO.lm.top6$fitted.values, col = "red")
```




```
plot(CO.ts, xlim=c(100,200))
lines(CO.lm.top6$fitted.values, col = "red")
```



Upon visually inspecting this new model, we see smaller fluctuations, which we wanted. However, we do see that the peaks of the smaller waves do not consistently match up with where the data has a peak, based on the zoomed in graphs.
After determining if there is a significant trend, we will compare the potential trend and seasonality models with five and six periods to determine which one will be the best selection for our final linear model.

Trend

Model Time Series Based on Time

```
# model
CO.lm.trend <- lm(CO.ts ~ time.CO)

# summary analysis
summary(CO.lm.trend)
```

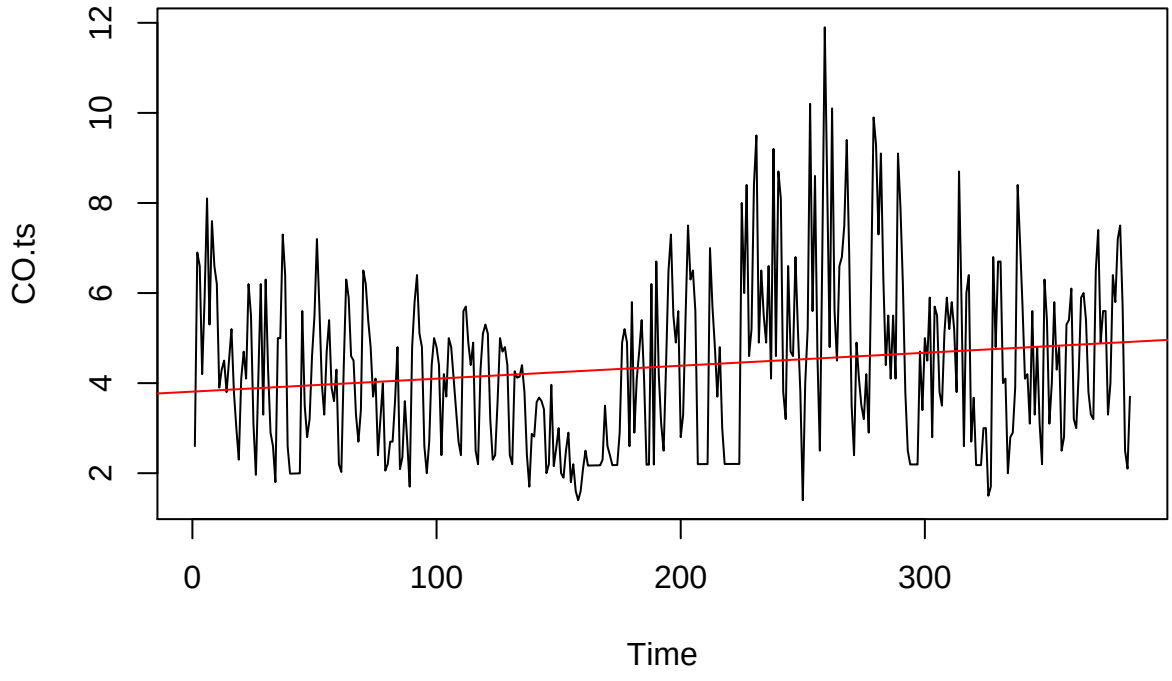
```
##
## Call:
## lm(formula = CO.ts ~ time.CO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2485 -1.6980 -0.0525  1.0863  7.3442
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.810929   0.192140  19.834  < 2e-16 ***
## time.CO     0.002876   0.000865   3.325  0.00097 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.879 on 382 degrees of freedom
## Multiple R-squared:  0.02813,    Adjusted R-squared:  0.02558
## F-statistic: 11.06 on 1 and 382 DF,  p-value: 0.0009695
```

The p-value is 0.00097, so there is a significant trend in the data.

Visualize Trend

```
plot(CO.ts)
abline(CO.lm.trend, col='red')
```



This graph shows the trendline plotted in red over the time series.

Model Comparison: Trend and Seasonality Together

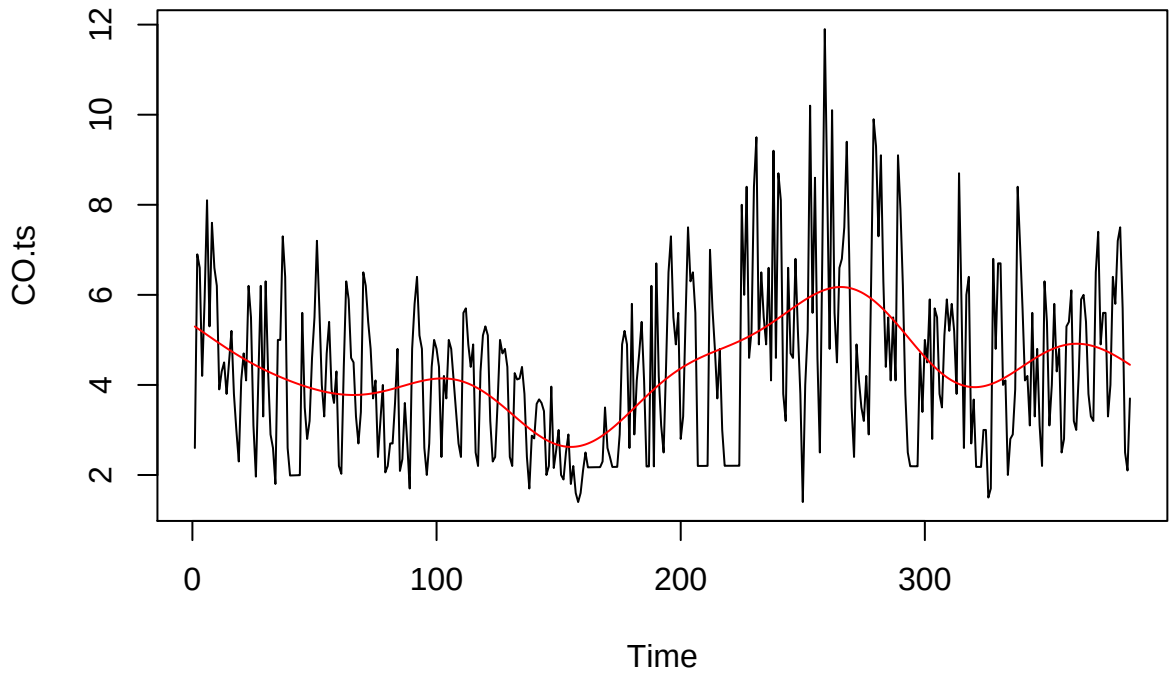
```r
# model with trend and 5 periods
CO.seasonal5.trend <- lm(CO.ts ~ time.CO + sin(2*pi*time.CO/CO.p1) +
                           cos(2*pi*time.CO/CO.p1) +
                           sin(2*pi*time.CO/CO.p2) +
                           cos(2*pi*time.CO/CO.p2) +
                           sin(2*pi*time.CO/CO.p3) +
                           cos(2*pi*time.CO/CO.p3) +
                           sin(2*pi*time.CO/CO.p4) +
                           cos(2*pi*time.CO/CO.p4)+
                           sin(2*pi*time.CO/CO.p5) +
                           cos(2*pi*time.CO/CO.p5))

# model with trend and 6 periods
CO.seasonal6.trend <- lm(CO.ts ~ time.CO + sin(2*pi*time.CO/CO.p1) +
                           cos(2*pi*time.CO/CO.p1) +
                           sin(2*pi*time.CO/CO.p2) +
                           cos(2*pi*time.CO/CO.p2) +
                           sin(2*pi*time.CO/CO.p3) +
                           cos(2*pi*time.CO/CO.p3) +
                           sin(2*pi*time.CO/CO.p4) +
                           cos(2*pi*time.CO/CO.p4) +
                           sin(2*pi*time.CO/CO.p5) +
                           cos(2*pi*time.CO/CO.p5) +
                           sin(2*pi*time.CO/CO.p6) +
                           cos(2*pi*time.CO/CO.p6))

anova(CO.seasonal5.trend, CO.seasonal6.trend)
```

```
## Analysis of Variance Table
##
## Model 1: CO.ts ~ time.CO + sin(2 * pi * time.CO/CO.p1) + cos(2 * pi *
##     time.CO/CO.p1) + sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2) + sin(2 * pi * time.CO/CO.p3) + cos(2 * pi *
##     time.CO/CO.p3) + sin(2 * pi * time.CO/CO.p4) + cos(2 * pi *
##     time.CO/CO.p4) + sin(2 * pi * time.CO/CO.p5) + cos(2 * pi *
##     time.CO/CO.p5)
## Model 2: CO.ts ~ time.CO + sin(2 * pi * time.CO/CO.p1) + cos(2 * pi *
##     time.CO/CO.p1) + sin(2 * pi * time.CO/CO.p2) + cos(2 * pi *
##     time.CO/CO.p2) + sin(2 * pi * time.CO/CO.p3) + cos(2 * pi *
##     time.CO/CO.p3) + sin(2 * pi * time.CO/CO.p4) + cos(2 * pi *
##     time.CO/CO.p4) + sin(2 * pi * time.CO/CO.p5) + cos(2 * pi *
##     time.CO/CO.p5) + sin(2 * pi * time.CO/CO.p6) + cos(2 * pi *
##     time.CO/CO.p6)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    372 1098.1
## 2    370 1089.2  2     8.914 1.5141 0.2214
```

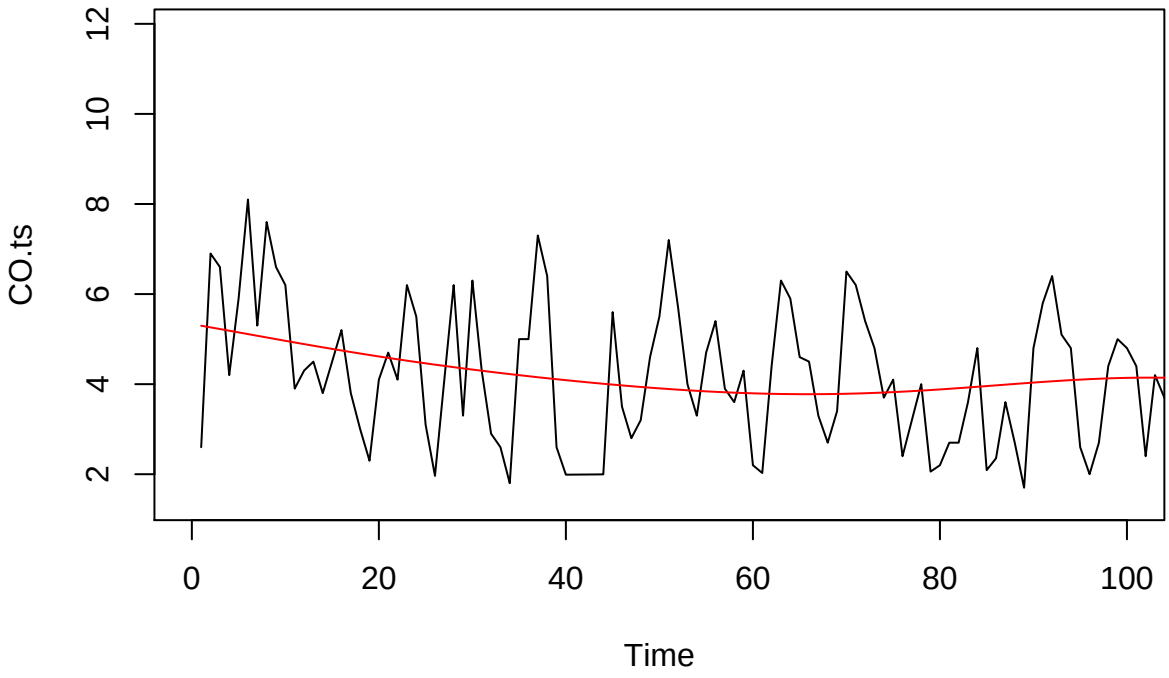
The p-value of the partial F test is large (not significant at the 0.05 level), which means that the smaller model is preferred. Based on this test, we would choose the model with trend and five periods.

Visual Comparison

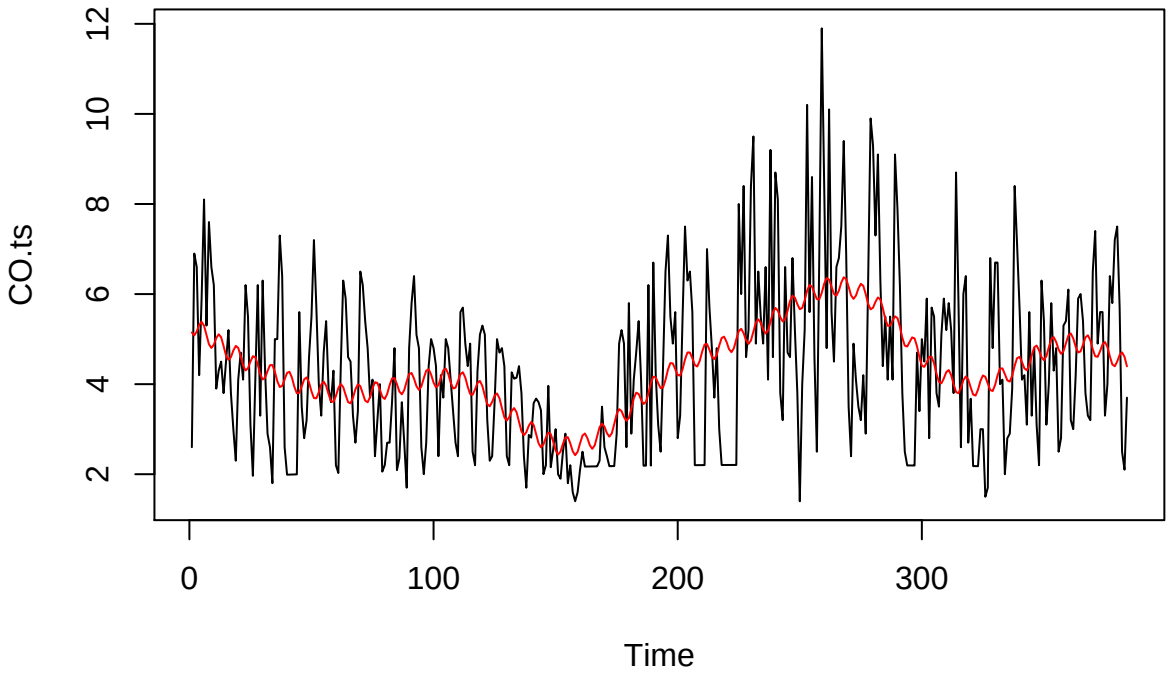
```
model1 <- CO.seasonal5.trend
plot(CO.ts)
lines(model1$fitted.values, col = "red")
```



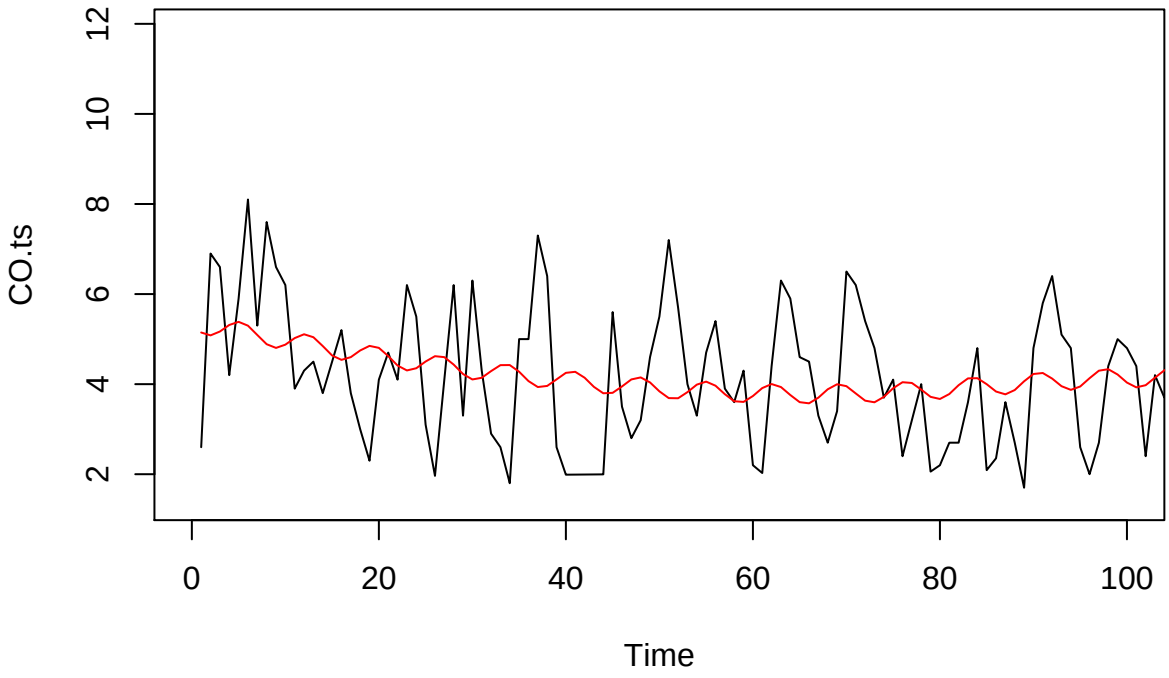

```
plot(CO.ts, xlim=c(0,100))
lines(model1$fitted.values, col = "red")
```




```
model2 <- CO.seasonal6.trend
plot(CO.ts)
lines(model2$fitted.values, col = "red")
```




```
plot(CO.ts, xlim=c(0,100))
lines(model2$fitted.values, col = "red")
```



Based on visual inspection, both models capture the overall trend of the data well. The model with trend and six periods has smaller fluctuations, instead of being a smooth curve like the other model, which visually appears to look more similar to the time series. Again, we see that the peaks of the model do not always occur at the same time value as the peaks of the data.

Comparison Using Metrics

```
# Adjusted R^2: model with trend and 5 periods
summary(CO.seasonal5.trend)$adj.r.squared
```

```
## [1] 0.1852438
```

```
# AIC: model with trend and 5 periods
AIC(CO.seasonal5.trend)
```

```
## [1] 1519.205
```

```
# Adjusted R^2: model with trend and 6 periods
summary(CO.seasonal6.trend)$adj.r.squared
```

```
## [1] 0.1874895
```

```
# AIC: model with trend and 6 periods
AIC(CO.seasonal6.trend)
```

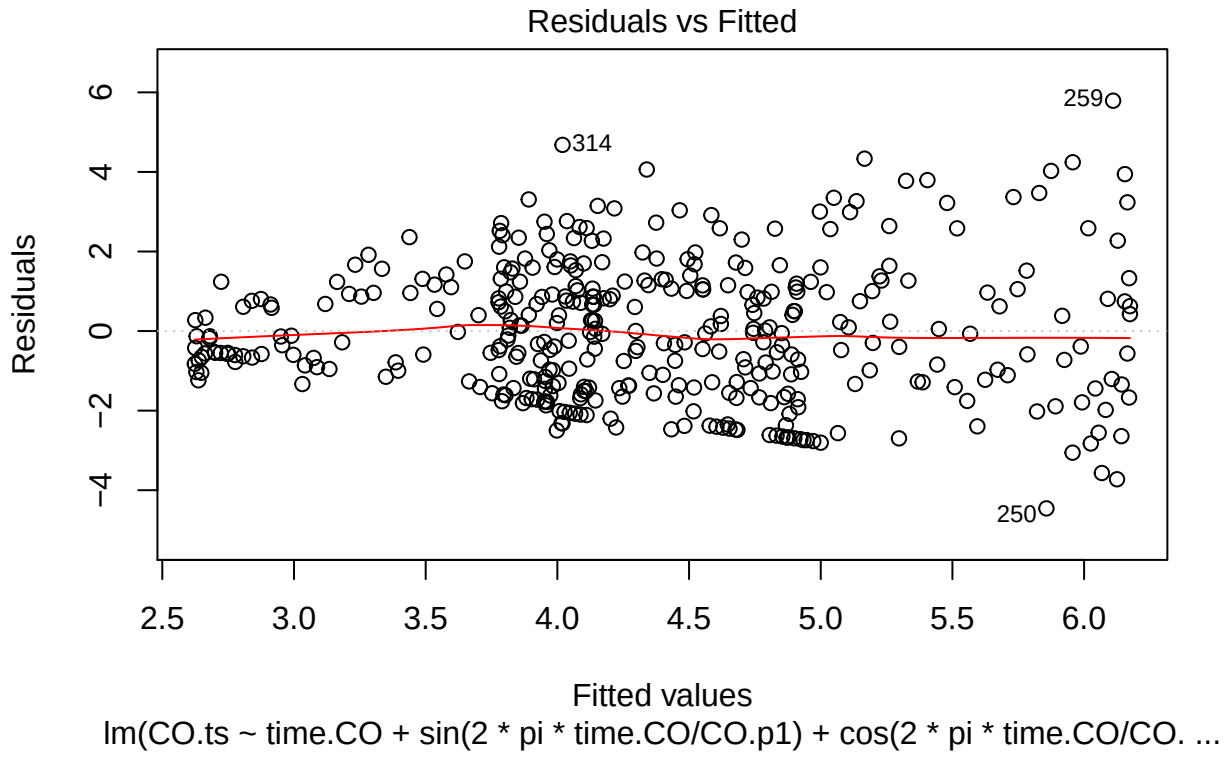
```
## [1] 1520.075
```

The adjusted R^2 of the model that includes the trend and 5 periods is 0.185. The adjusted R^2 of the model that includes the trend and 6 periods is 0.187. Based on adjusted R^2, we would select the model with 6 periods, though they are very close based on this metric and neither value is particularly close to 1. The AIC of the model with 5 periods is 1519.205. The AIC of the model with 6 periods is 1520.075. Based on AIC, we would choose the model with 5 periods, because it has the smaller AIC value. Again, the values are very similar, though.

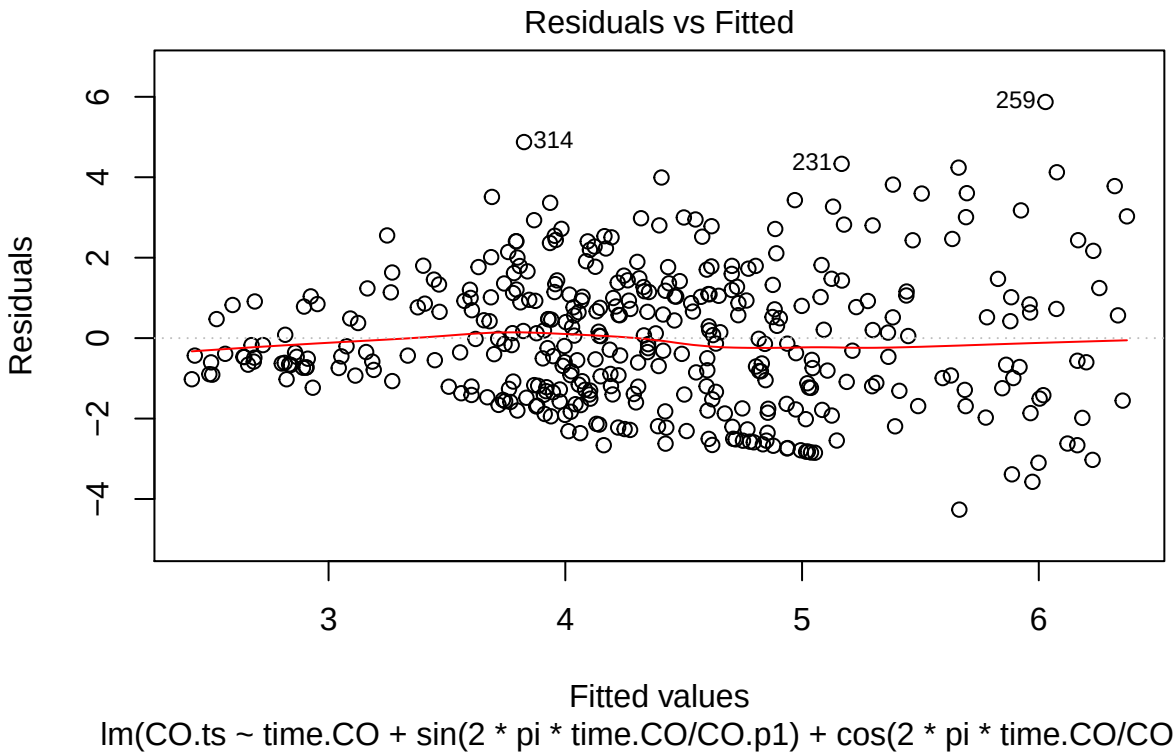
Diagnostics

Residuals vs. Fitted

```
plot(CO.seasonal5.trend, which = 1)
```



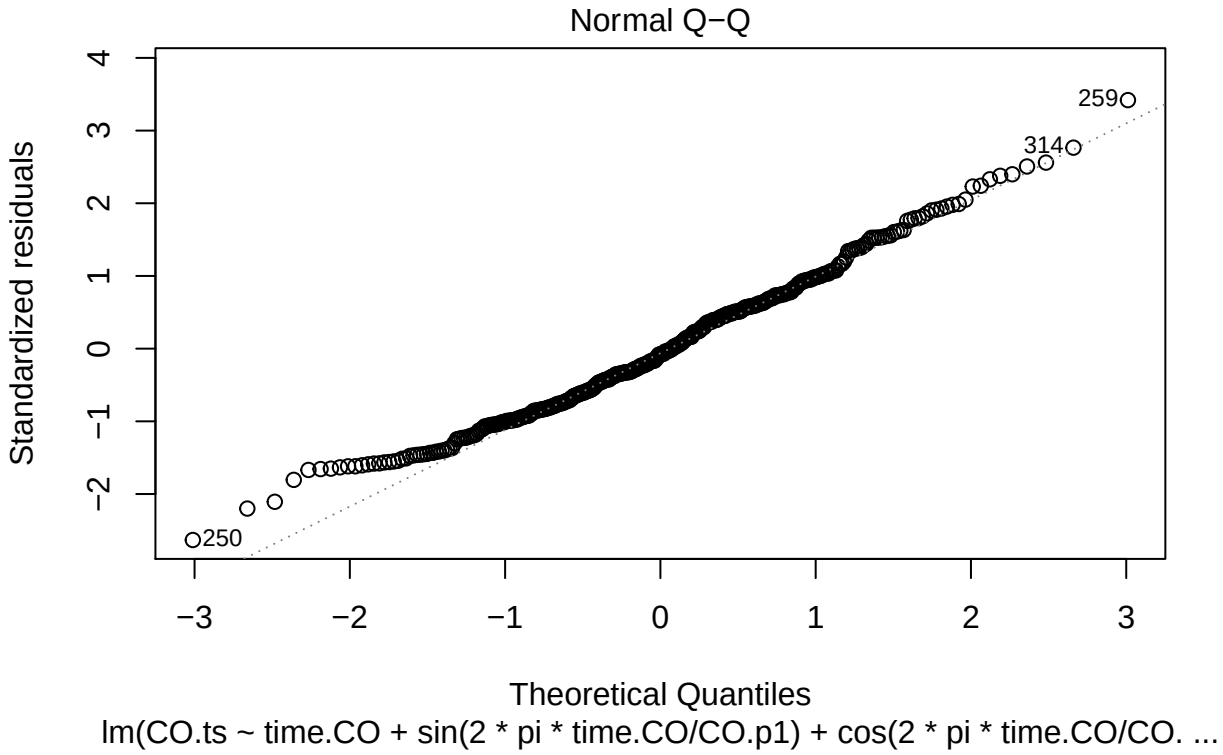

```
plot(CO.seasonal6.trend, which = 1)
```



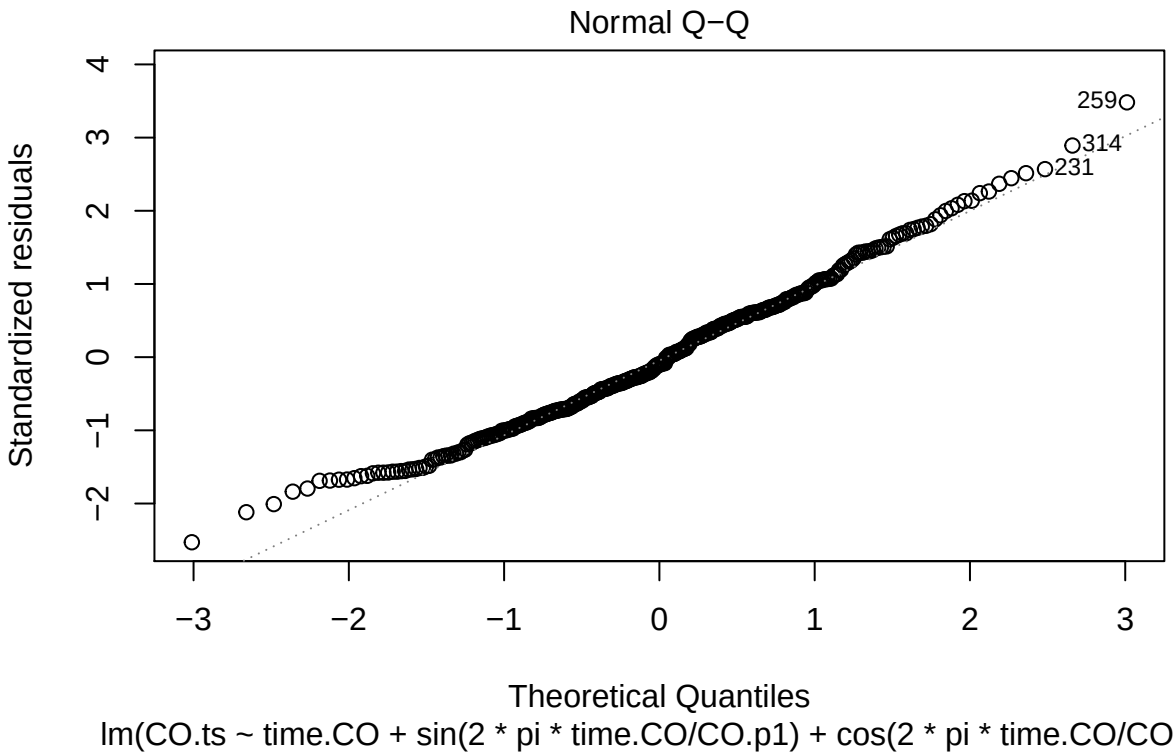
The mean of the residuals is approximately 0, but the variance is not constant for either model. There is less variance (less spread above and below the x-axis) on the left-hand side of the plot, and the variance increases to the right. The relationship evident in the plots indicates a lack of fit.

QQ Plot

```
plot(CO.seasonal5.trend, which = 2)
```



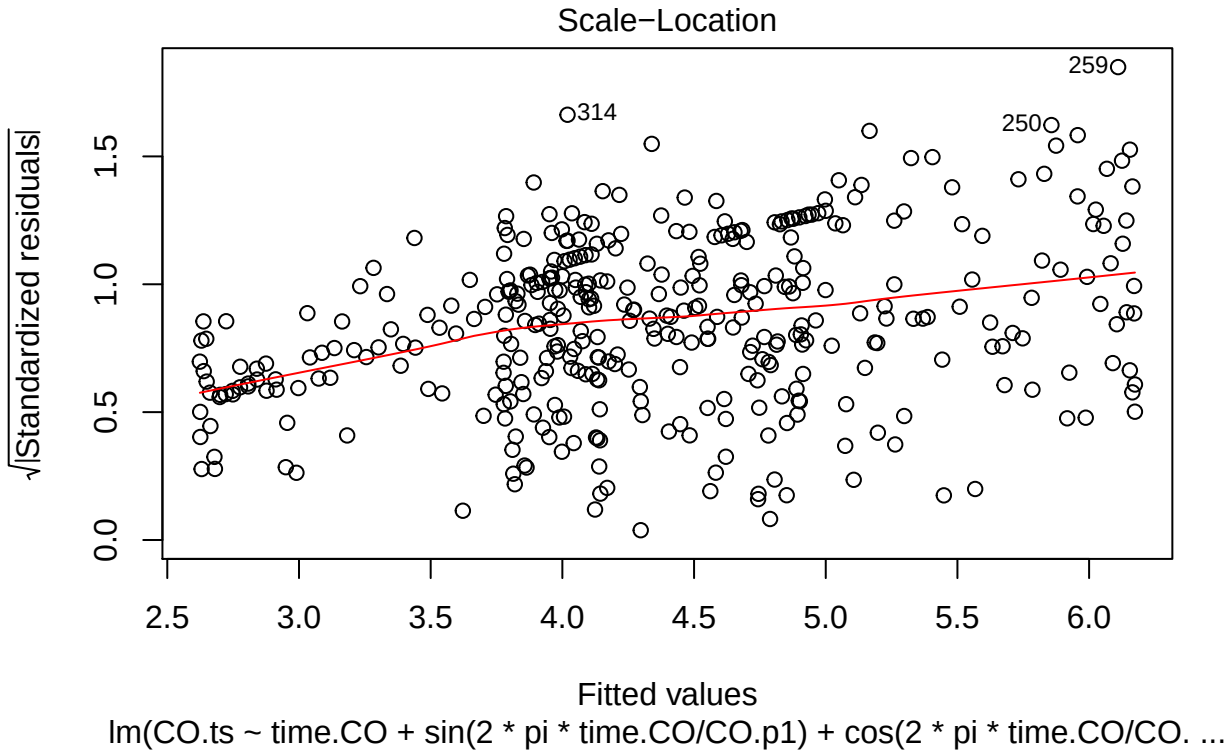

```
plot(CO.seasonal6.trend, which = 2)
```



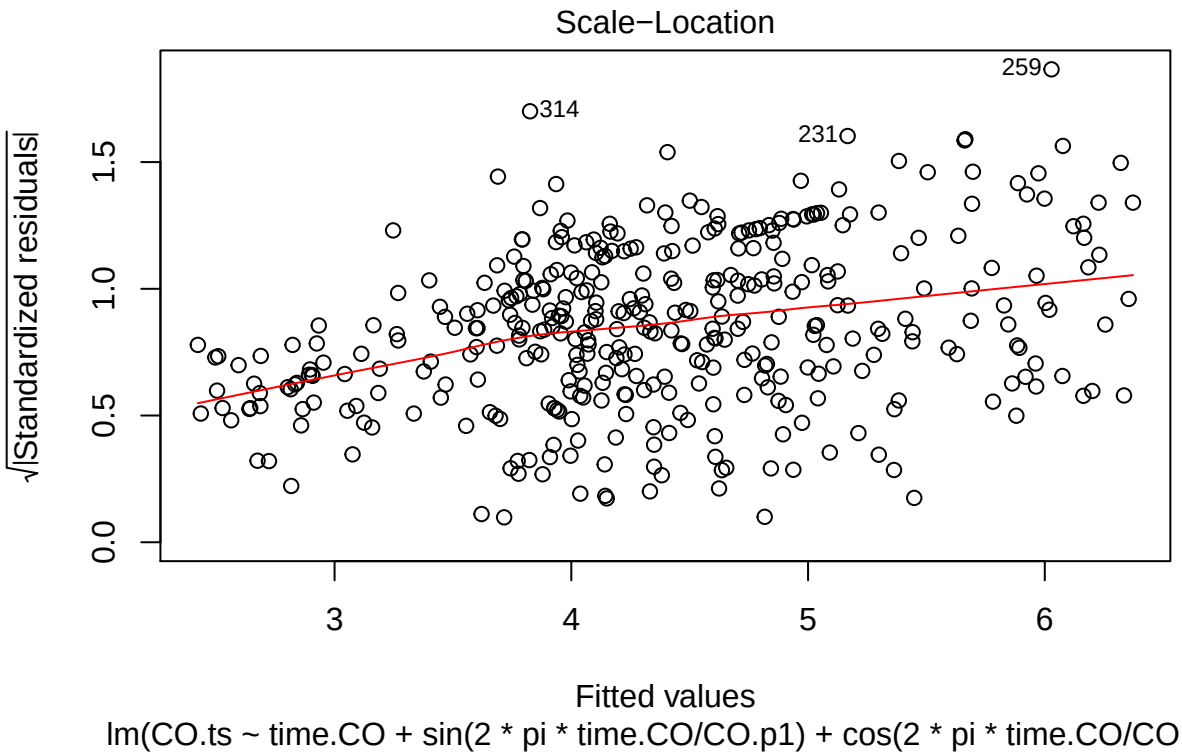
The residuals are approximately normal for both models, though the tails, and the lower tails especially, show some deviation from normality.

Scale-Location Plot

```
plot(CO.seasonal5.trend, which = 3)
```



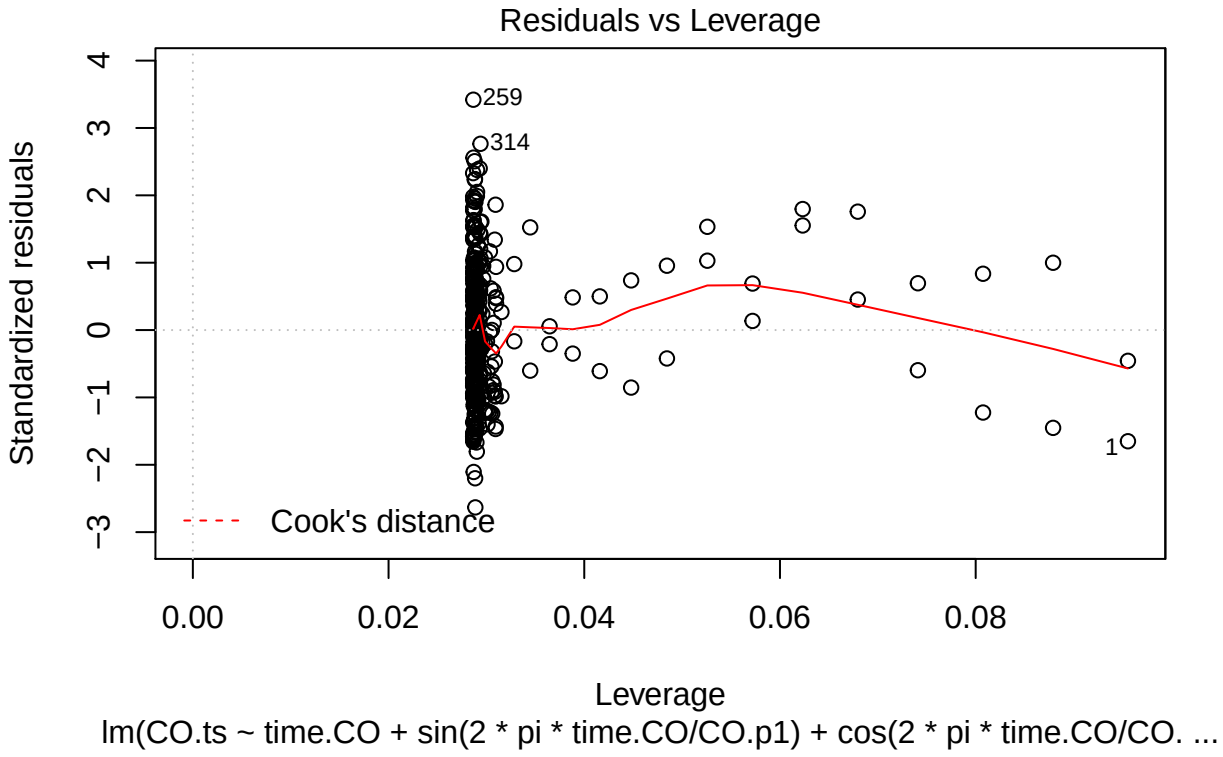

```
plot(CO.seasonal6.trend, which = 3)
```



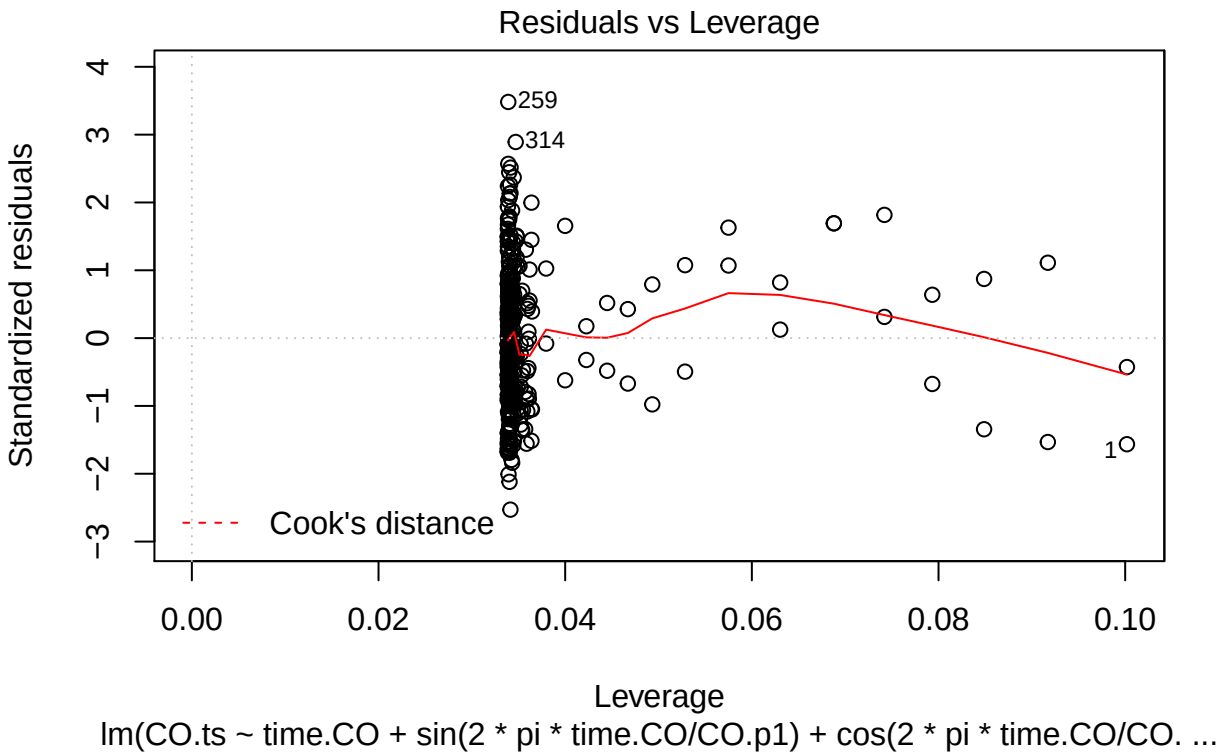
The mean is not centered at 0 and there may be a slight pattern to the scatter in both plots.

Residuals vs. Leverage

```
plot(CO.seasonal5.trend, which = 5)
```

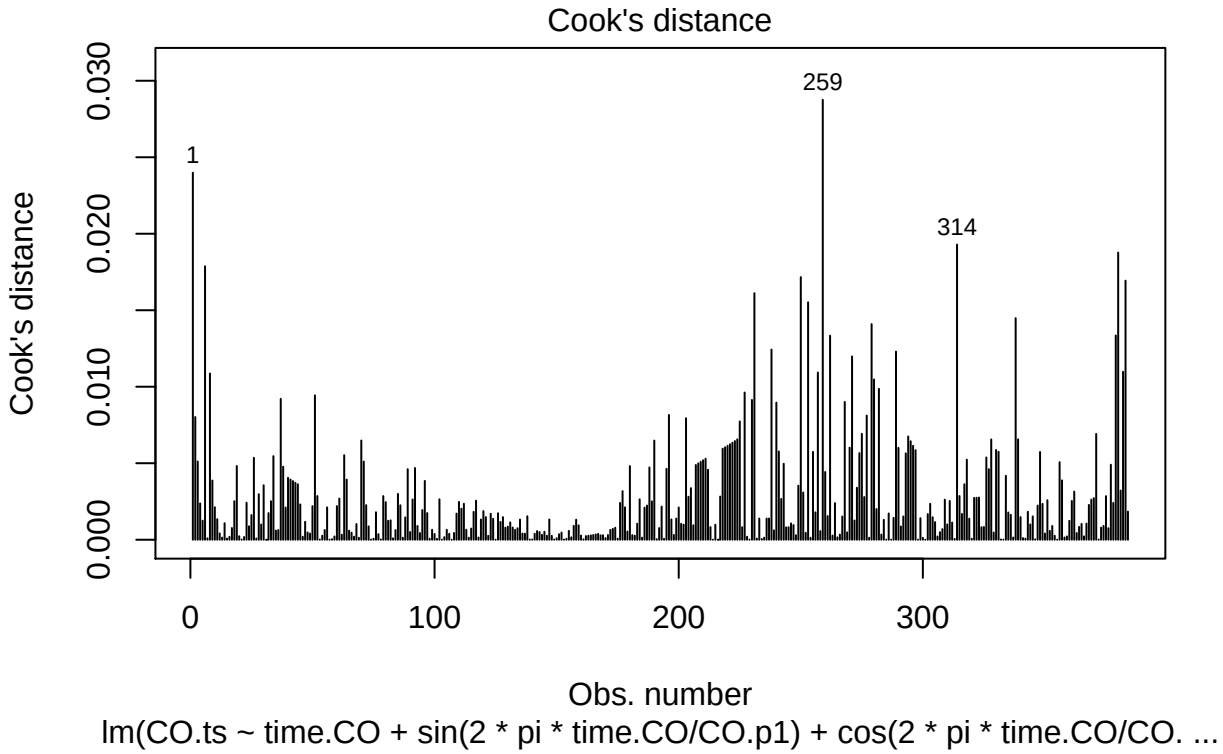



```
plot(CO.seasonal6.trend, which = 5)
```

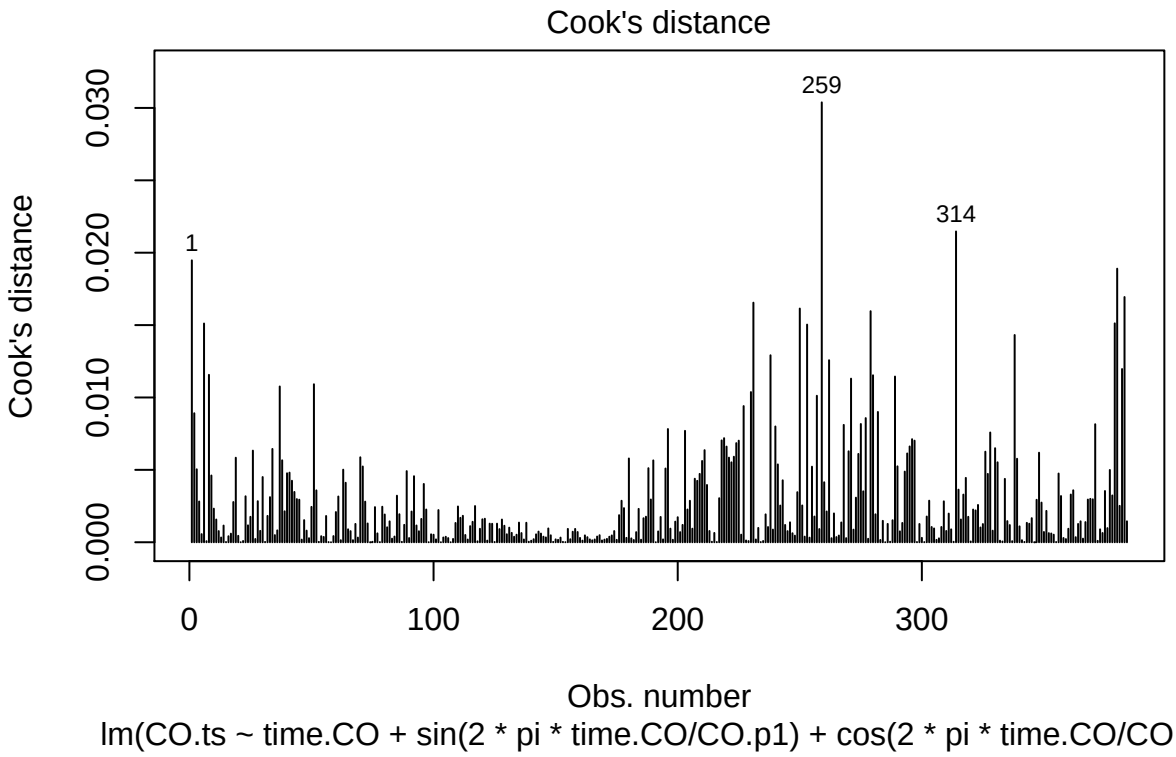


Neither plot has any points with Cook's distances greater than 0.5.

```
# plot Cook's distances
plot(CO.seasonal5.trend,labels.id = NULL, which = 4)
```




```
plot(CO.seasonal6.trend,labels.id = NULL, which = 4)
```



These plots show that all of the points have small Cook's distances for both models.

Both models perform very similarly in the diagnostics. We do not feel that either model needs a log or Box Cox transformation to improve the diagnostics.

Choose Trend + Seasonal Model

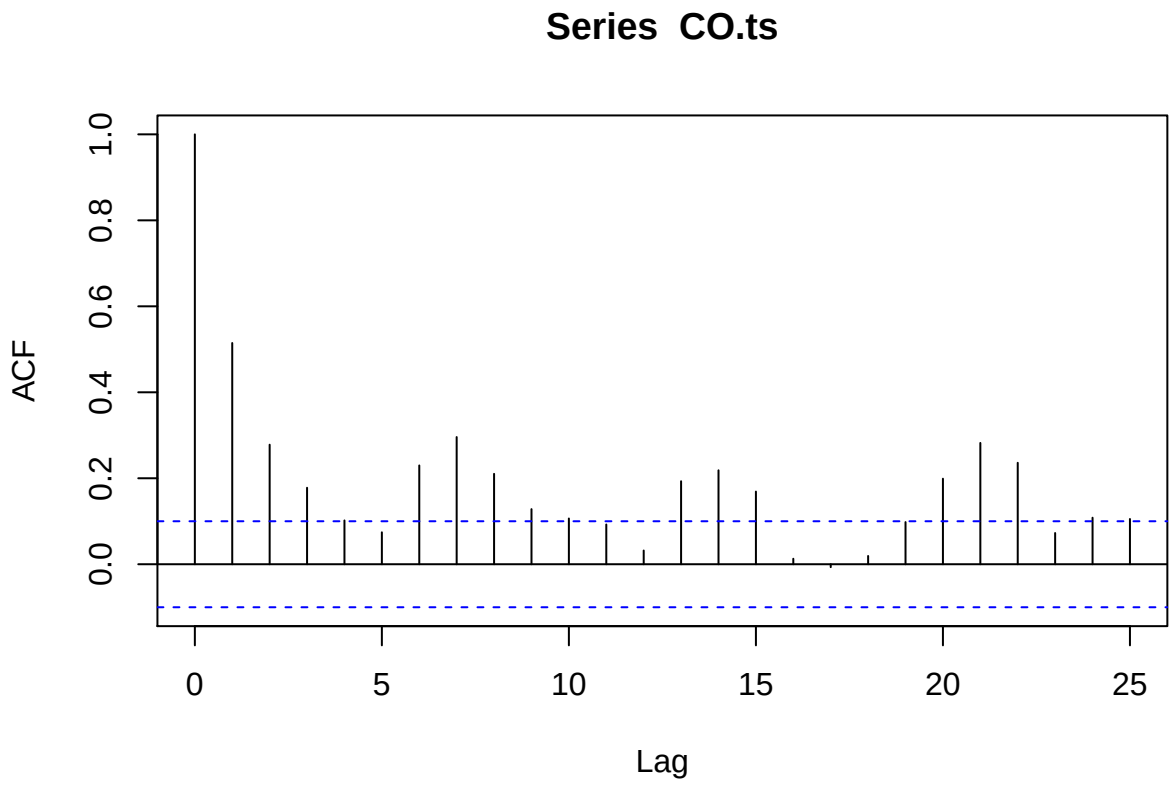
```
CO.lm <- CO.seasonal5.trend
```

Since both models performed similarly in metrics and diagnostics, we have chosen to use the model with the trend and top five identified periods to explain seasonality. It was preferred from the partial F test, and we feel that it is better to choose the simpler model since the two models have such similar performance in metrics and diagnostics.

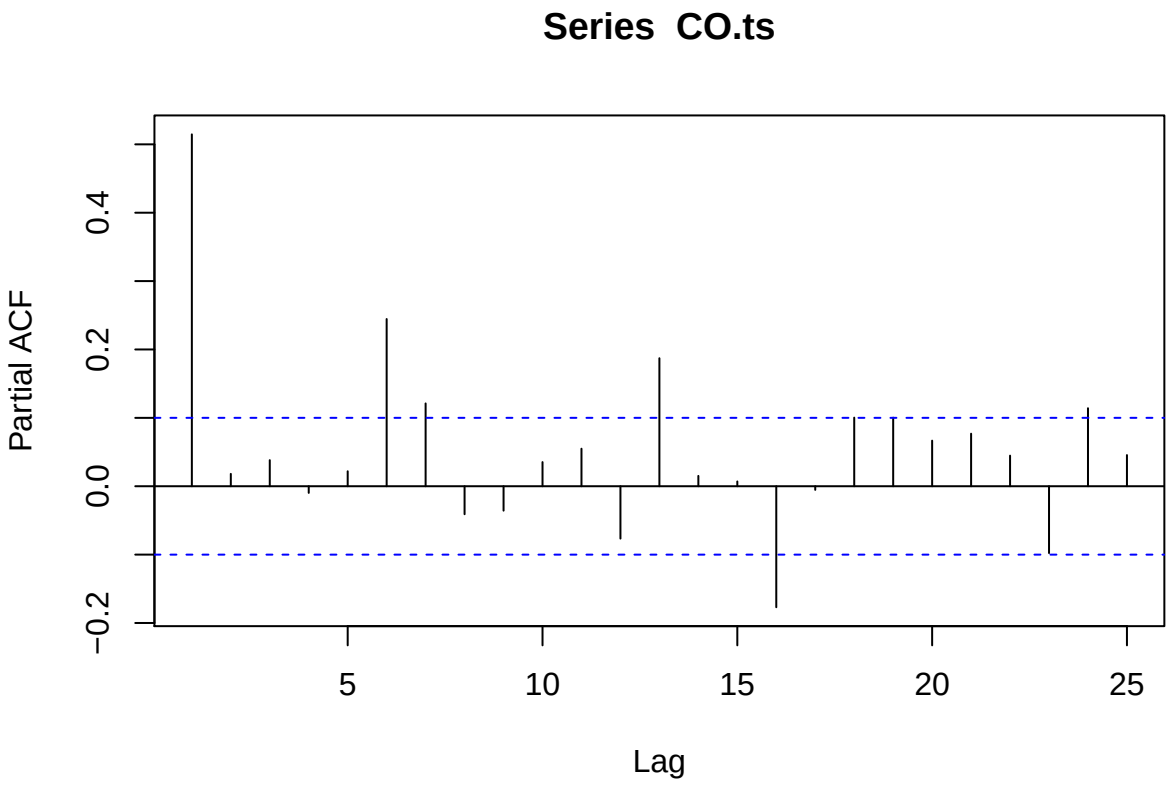
Model Residuals

ACF and PACF

```
# ACF and PACF
acf(CO.ts)
```




```
pacf(CO.ts)
```



The ACF is approximately sinusoidal. The PACF does not cut off after a certain number of lags and has some sinusoidal behavior. We predict that there will be autoregressive and moving average terms in the model of the residuals.
Based on the PACF, a potential value for p for the autoregressive portion of the model is 1. Since the first cutoff for the PACF is 1, we will also test a value of 2 for p, to see if that is better suited to modeling the residuals.
Based on the ACF, a potential value for q for the moving average portion of the model is 3. We will also test values of 1 and 2 for q, since they are the other significant lags before the first cutoff in the ACF.
Additionally, we are unsure of the stationarity of the data, as we know there is a trend and the ACF appears to have some linear decay at the very beginning. This is a key assumption that must be met, so we will test models using the time series itself (d = 0) and the first differences (d = 1).

ARIMA Models

We tested the possible combinations of our p, d, and q values.

```
# get residuals
e.ts.CO <-ts(CO.lm$residuals)

# 1, 0, 1
CO.arima101 <- arima(e.ts.CO, order = c(1,0,1), include.mean = FALSE)
summary(CO.arima101)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 0, 1), include.mean = FALSE)
```

```
##
## Coefficients:
##          ar1     ma1
##       0.2558  0.1556
## s.e.  0.1179  0.1194
##
## sigma^2 estimated as 2.422:  log likelihood = -714.84,  aic = 1435.68
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 9.857412e-06 1.556433 1.243264 -6.435751 234.0501 0.8925046
##                      ACF1
## Training set -0.001073604
```

```
# AIC = 1435.68

# 1, 0, 2
CO.arima102 <- arima(e.ts.CO, order = c(1,0,2), include.mean = FALSE)
summary(CO.arima102)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1     ma1     ma2
##       0.1424  0.2695  0.0515
## s.e.  0.3303  0.3303  0.1246
##
## sigma^2 estimated as 2.421:  log likelihood = -714.74,  aic = 1437.48
##
## Training set error measures:
##                       ME     RMSE     MAE       MPE     MAPE     MASE
## Training set 9.263487e-05 1.556042 1.24356 -3.628518 231.1227 0.892717
##                      ACF1
## Training set -0.002331555
```

```
# AIC = 1437.48

# 1, 0, 3
CO.arima103 <- arima(e.ts.CO, order = c(1,0,3), include.mean = FALSE)
summary(CO.arima103)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 0, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1     ma1     ma2     ma3
##       0.0256  0.3860  0.1080  0.0277
## s.e.  0.4518  0.4489  0.1919  0.0831
##
## sigma^2 estimated as 2.421:  log likelihood = -714.7,  aic = 1439.4
```

```
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 5.062446e-05 1.555868 1.242231 -2.990769 231.7542 0.8917627
##                      ACF1
## Training set -0.002410357
```

```
# AIC = 1439.4

# 1, 1, 1
CO.arima111 <- arima(e.ts.CO, order = c(1,1,1), include.mean = FALSE)
summary(CO.arima111)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 1, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1
##       0.3902  -1.0000
## s.e.  0.0473   0.0066
##
## sigma^2 estimated as 2.44:  log likelihood = -716.8,  aic = 1439.59
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.01294525 1.559858 1.239055 4.189649 222.4352 0.8894826
##                     ACF1
## Training set 0.03149704
```

```
# AIC = 1439.59

# 1, 1, 2
CO.arima112 <- arima(e.ts.CO, order = c(1,1,2), include.mean = FALSE)
summary(CO.arima112)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 1, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.2633  -0.8490  -0.1510
## s.e.  0.1178   0.1195   0.1193
##
## sigma^2 estimated as 2.429:  log likelihood = -716.01,  aic = 1440.02
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE    MAPE      MASE
## Training set 0.01281589 1.556437 1.236953 -8.176718 235.899 0.8879739
##                     ACF1
## Training set 0.005020125
```

```r
# AIC = 1440.02

# 1, 1, 3
CO.arima113 <- arima(e.ts.CO, order = c(1,1,3), include.mean = FALSE)
summary(CO.arima113)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(1, 1, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1      ma2      ma3
##       -0.5876  -0.0016  -0.6897  -0.3087
## s.e.   0.2798   0.2784   0.1803   0.1081
##
## sigma^2 estimated as 2.428:  log likelihood = -716.04,  aic = 1442.08
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.01187188 1.556257 1.238727 -14.01503 233.6882 0.8892473
##                    ACF1
## Training set 0.011253
```

```r
# AIC = 1442.08

# 2, 0, 1
CO.arima201 <- arima(e.ts.CO, order = c(2,0,1), include.mean = FALSE)
summary(CO.arima201)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 0, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2     ma1
##       1.3420  -0.4281  -1.000
## s.e.  0.0461   0.0461   0.007
##
## sigma^2 estimated as 2.259:  log likelihood = -703.16,  aic = 1414.33
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE    MASE
## Training set -0.04132272 1.503041 1.208502 -6.014657 253.0251 0.86755
##                       ACF1
## Training set -0.001386667
```

```r
# AIC = 1414.33
# lowest

# 2, 0, 2
CO.arima202 <- arima(e.ts.CO, order = c(2,0,2), include.mean = FALSE)
summary(CO.arima202)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.1614  -0.8786  -0.9495  0.7901
## s.e.  0.0502   0.0741   0.0967  0.1033
##
## sigma^2 estimated as 2.364:  log likelihood = -710.4,  aic = 1430.8
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE     MASE
## Training set 0.002406279 1.537392 1.212608 29.85034 205.0663 0.870497
##                    ACF1
## Training set 0.1493435
```

```
# AIC = 1430.8

# 2, 0, 3
CO.arima203 <- arima(e.ts.CO, order = c(2,0,3), include.mean = FALSE)
summary(CO.arima203)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 0, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2     ma1     ma2     ma3
##       0.0989  -0.6959  0.2846  0.8057  0.3595
## s.e.  0.1266   0.0823  0.1197  0.0615  0.0503
##
## sigma^2 estimated as 2.332:  log likelihood = -707.66,  aic = 1427.32
##
## Training set error measures:
##                        ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.0003645137 1.526964 1.209168 17.83528 228.9103 0.8680278
##                     ACF1
## Training set 0.01529922
```

```
# AIC = 1427.32

# 2, 1, 1
CO.arima211 <- arima(e.ts.CO, order = c(2,1,1), include.mean = FALSE)
summary(CO.arima211)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 1, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.4178  -0.0714  -1.0000
```

```
## s.e.  0.0512   0.0514   0.0067
##
## sigma^2 estimated as 2.426:  log likelihood = -715.83,  aic = 1439.67
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.01284961 1.555644 1.236936 -7.890044 235.9243 0.8879618
##                       ACF1
## Training set -0.0001517824
```

```
# AIC = 1439.67

# 2, 1, 2
CO.arima212 <- arima(e.ts.CO, order = c(2,1,2), include.mean = FALSE)
summary(CO.arima212)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 1, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.3439  -0.4301  -1.9987  0.9992
## s.e.  0.0468   0.0462   0.0136  0.0136
##
## sigma^2 estimated as 2.284:  log likelihood = -706.88,  aic = 1423.76
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.01900397 1.509176 1.210507 -5.440829 248.3164 0.8689891
##                     ACF1
## Training set 0.01292897
```

```
# AIC = 1423.76
# second lowest

# 2, 1, 3
CO.arima213 <- arima(e.ts.CO, order = c(2,1,3), include.mean = FALSE)
summary(CO.arima213)
```

```
##
## Call:
## arima(x = e.ts.CO, order = c(2, 1, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2      ma3
##       1.1599  -0.8756  -1.9457  1.7325  -0.7868
## s.e.  0.0517   0.0775   0.1008  0.2039   0.1079
##
## sigma^2 estimated as 2.37:  log likelihood = -711.87,  aic = 1435.73
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
```

```
## Training set 0.007537325 1.537423 1.210064 29.66972 205.7083 0.8686709
##                    ACF1
## Training set 0.1552788
```

```
# AIC = 1435.73
```

Based on AIC, the two best models are the ARIMA(2,0,1) and ARIMA(2,1,2) models.

After testing the possible values of p, d, and q that we identified, we also used the auto.arima function to generate another model.

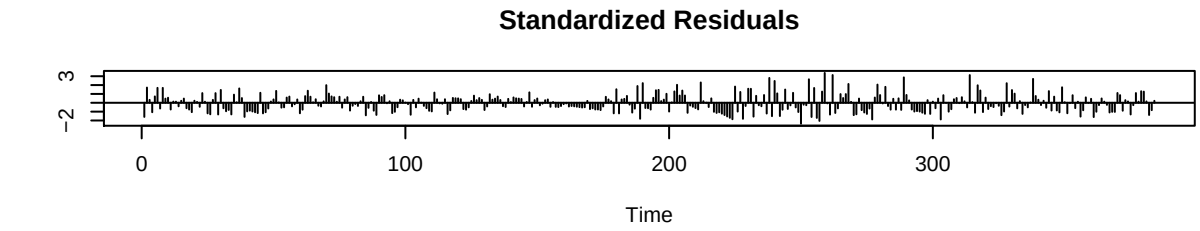
```
# auto.arima model
CO.residuals.auto <- auto.arima(e.ts.CO, approximation = FALSE)

# summary
summary(CO.residuals.auto)
```

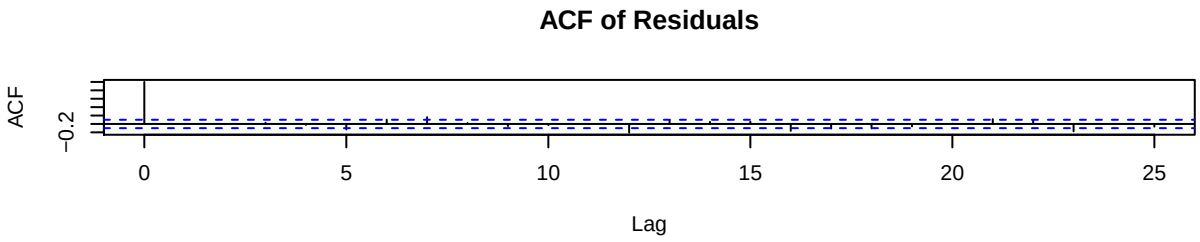
```
## Series: e.ts.CO
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##       0.4154  -0.0738
## s.e.  0.0511   0.0513
##
## sigma^2 estimated as 2.433:  log likelihood=-714.64
## AIC=1435.28   AICc=1435.35   BIC=1447.14
##
## Training set error measures:
##                         ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.0001387431 1.555635 1.243131 -6.270543 234.2375 0.892409
##                       ACF1
## Training set -0.006696604
```

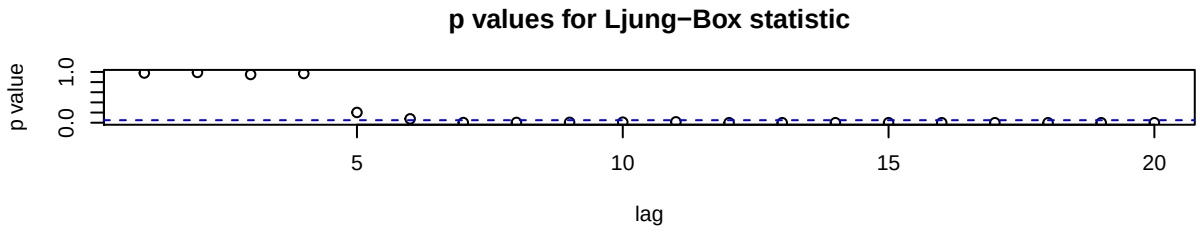
The ARIMA model is a 2,0,0 model. This means that there are autoregressive and moving average terms, where p = 2 and q = 2. Also, d = 0, meaning that the first differences were not needed to ensure stationarity, which is a key assumption to modeling the residuals. The AIC for this model is 1435.28.

Diagnostics

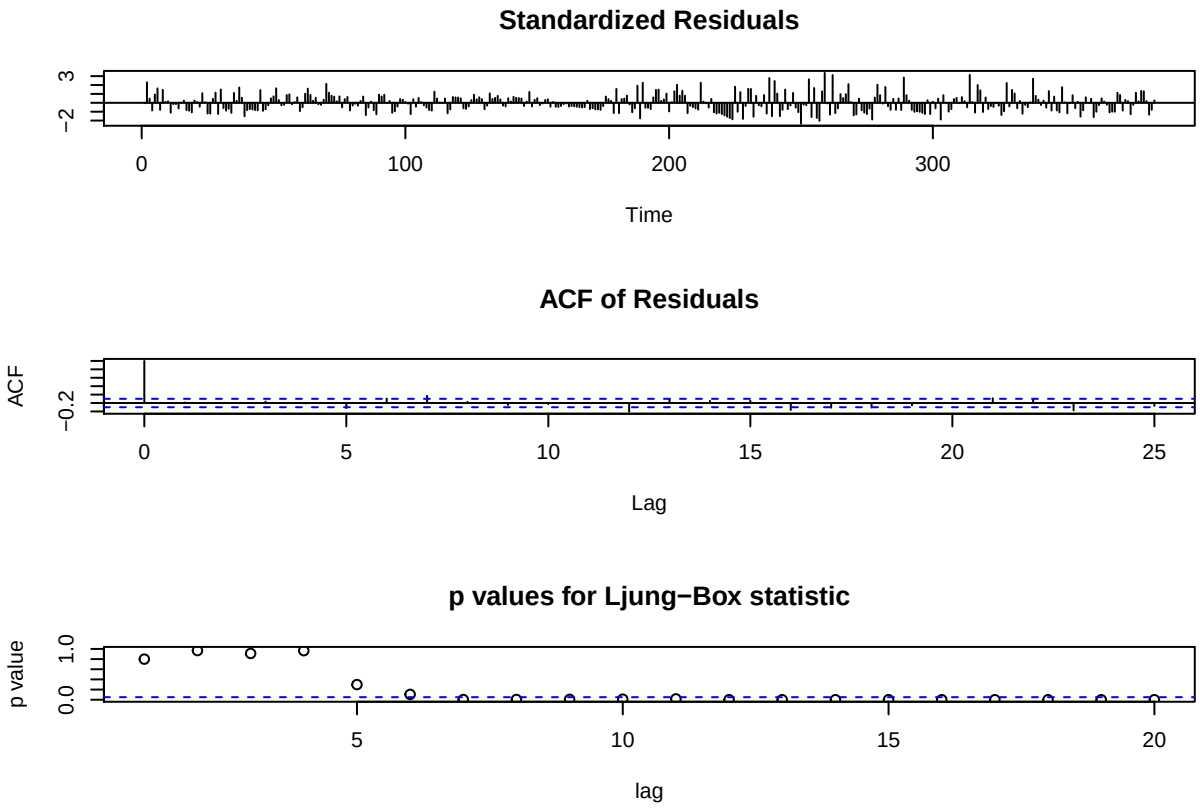
```
tsdiag(CO.arima201, gof.lag = 20)
```



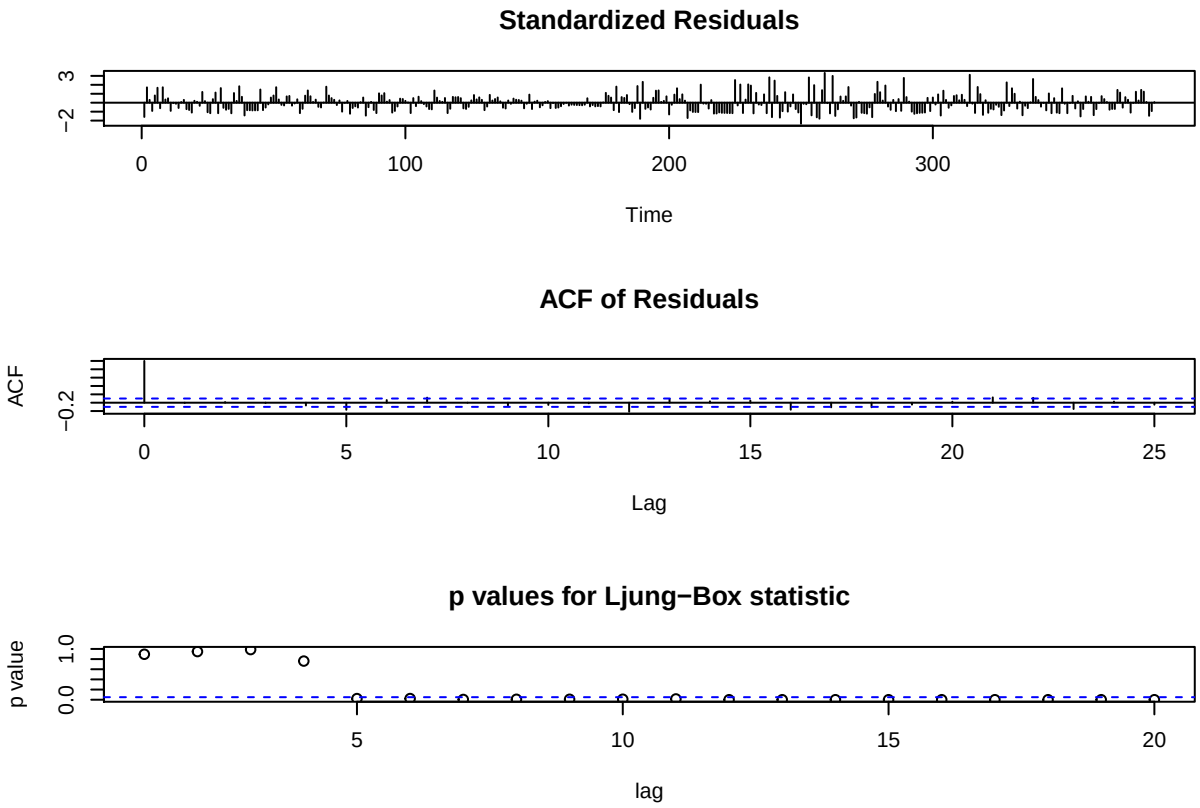





```
tsdiag(CO.arima212, gof.lag = 20)
```




```
tsdiag(CO.residuals.auto, gof.lag = 20)
```



Both the (2,0,1) and (2,1,2) models perform similarly in the diagnostic plot, though the (2,1,2) model has a slightly higher p-value for the sixth lag than the other model. The auto-generated (2,0,0) model is only adequate for four lags, as the p-value of the Ljung-Box statistic is not different from 0 (above the dashed line) for only the first four points.

Choose Model for Residuals

```
CO.residuals <- CO.arima201
```

The ARIMA(2,0,1) model has the smallest AIC of all of the models considered. It performs better than the auto-generated model, as it is adequate for more lags. Since it performs similarly to the ARIMA(2,1,2) model in the diagnostic plot but has a lower AIC, we decided to choose the ARIMA(2,0,1) model for the residuals.

Final Model

The final model includes a trend and seasonality, using 5 different periods (384, 192, 128, 96, and 76.8 days). The ARIMA model of the residuals is a 2,0,1 model.

Diagnostics

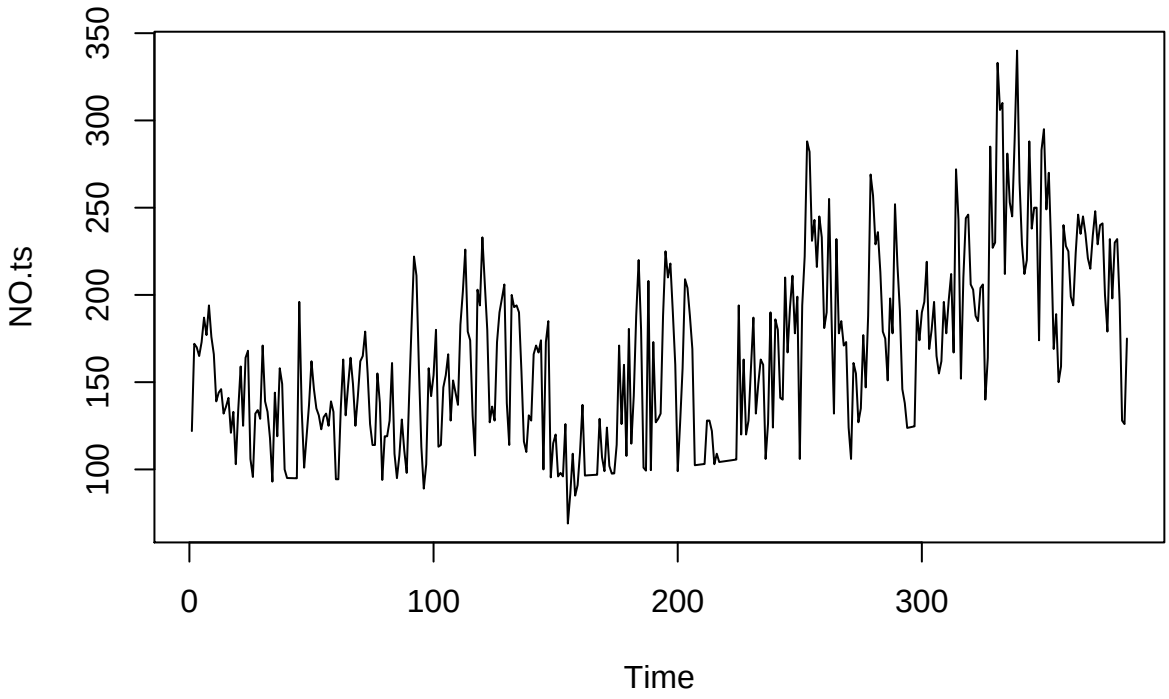
The residuals of the trend and seasonality linear model appear relatively normal, excluding the lower tail, from the QQ plot; however, problems remain with the residuals vs. fitted plot, indicating lack of fit. In

the future, models should address problems with the normality in the lower tail and the lack of fit. The diagnostic plot for the model of the residuals shows that it is adequate for five or six lags. Future work could focus on increasing the number of lags for which the model is considered adequate.

Univariate Time Series for NO2

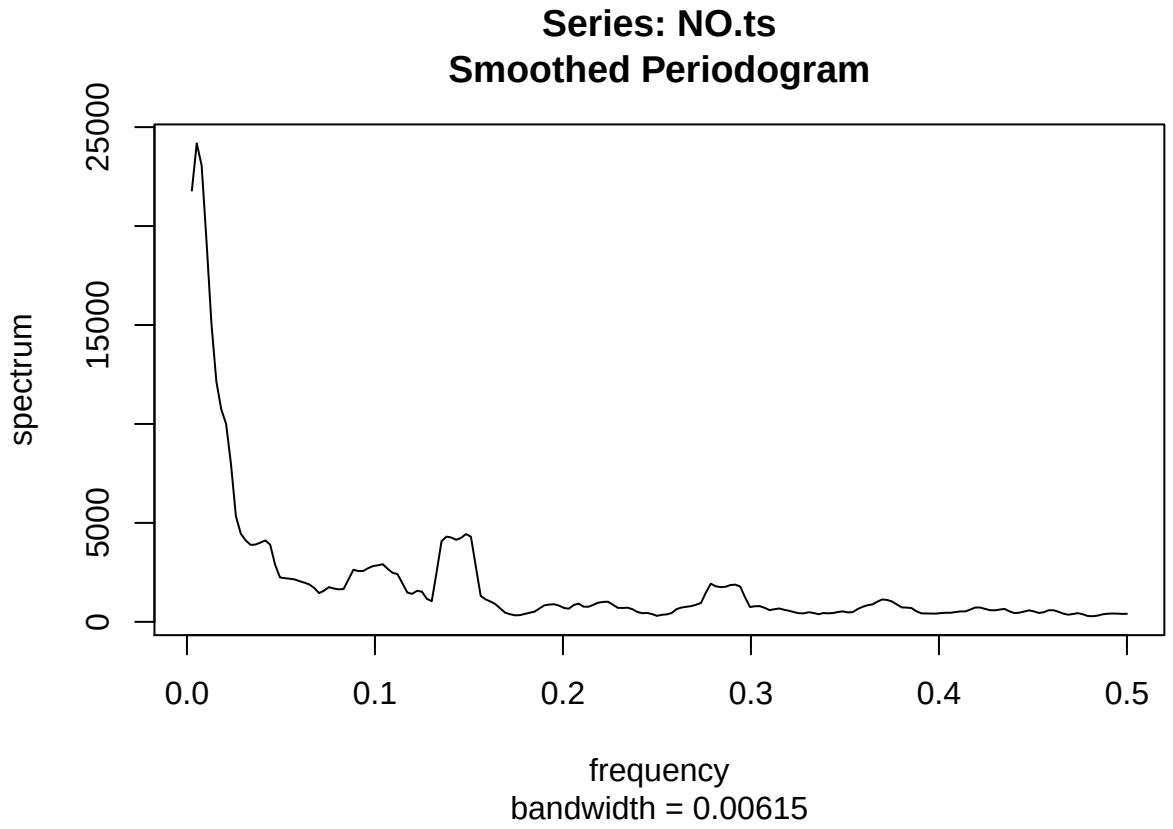
Visualize the Data

```
# create time series
NO.ts <- ts(air.train$NO2.GT.)

# visualize raw data
plot(NO.ts)
```



There appears to be an increasing trend, especially after about t = 150. There is also the potential for seasonality, but this fluctuation could instead be random.

```
# periodogram
pg.NO <- spec.pgram(NO.ts, spans = 9, demean = T, log = 'no')
```

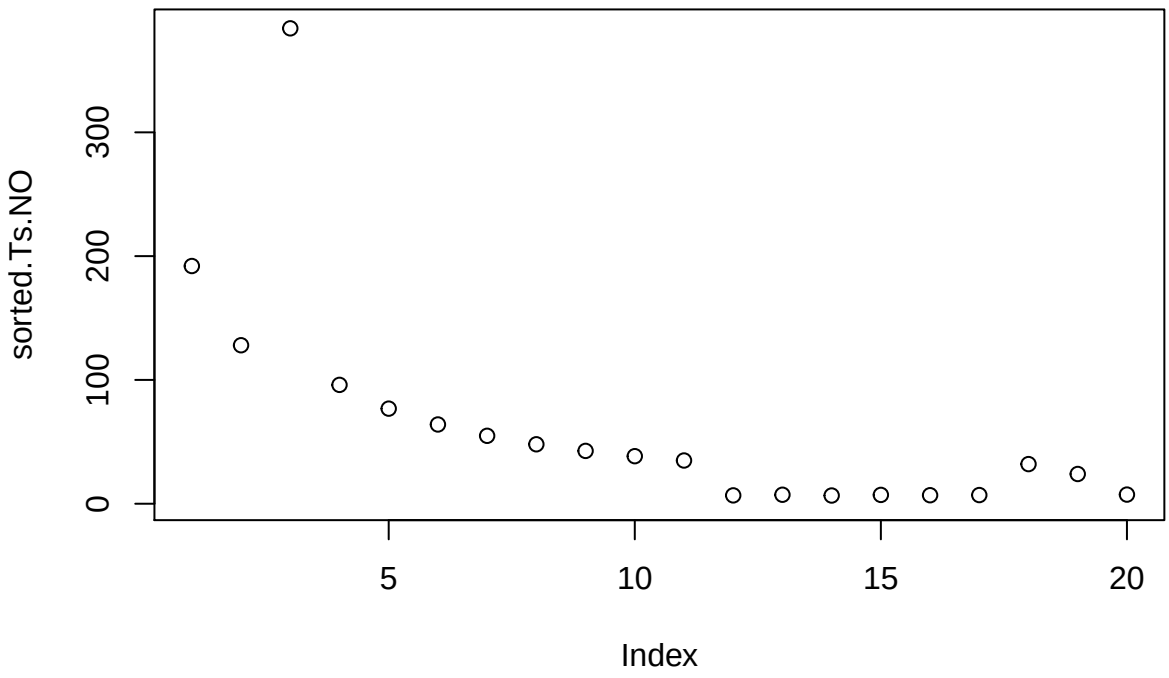


Based on the periodogram, there are multiple possibilities for the period of a seasonal component. There are fewer spikes than the periodogram for CO. There are still spikes, though, each of which indicate a possible frequency/period to explain the seasonality of the data. We predict that if seasonality is significant in the model, it will be based on a complex wave.

Seasonality

Finding Potential Periods

```
# sort the frequencies based on influence
sorted.spec.NO <- sort(pg.NO$spec, decreasing = T, index.return=T)

# convert to periods
sorted.omegas.NO <- pg.NO$freq[sorted.spec.NO$ix]
sorted.Ts.NO <- 1/pg.NO$freq[sorted.spec.NO$ix]

plot(sorted.Ts.NO, xlim = c(1,20))
```




```
# the cutoff for influential
NO.pg.cutoff <- 15

# the top periods
print('top periods')
```

```
## [1] "top periods"
```

```
sorted.Ts.NO[1:NO.pg.cutoff]
```

```
##  [1] 192.000000 128.000000 384.000000  96.000000  76.800000  64.000000
##  [7]  54.857143  48.000000  42.666667  38.400000  34.909091   6.736842
## [13]   7.245283   6.620690   7.111111
```

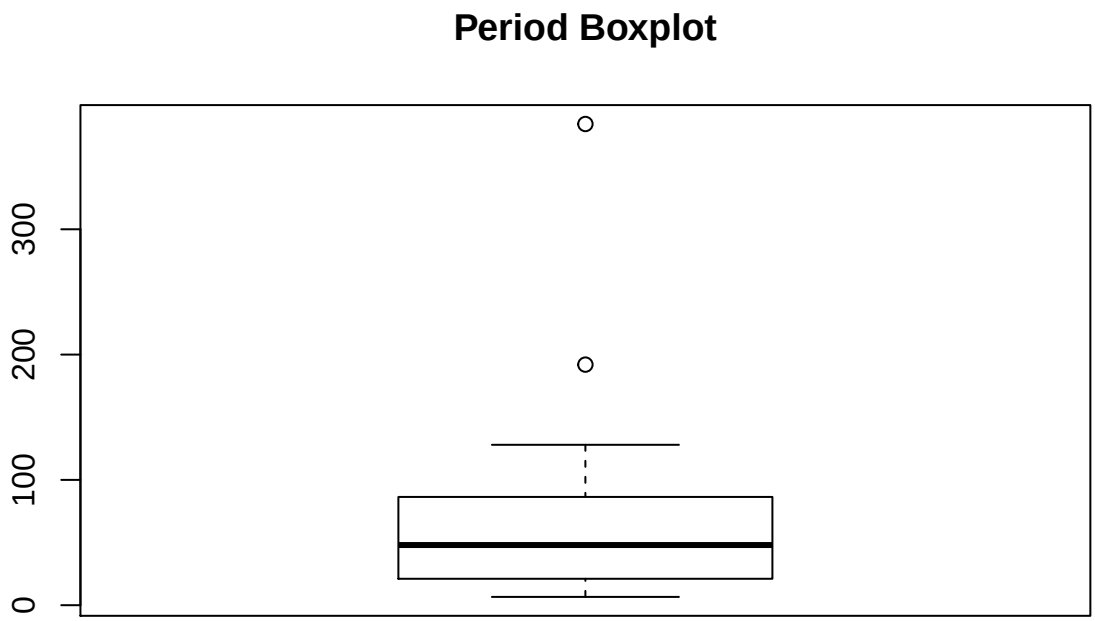
```
# top frequencies
## to double check that this makes sense based on periodogram
print('top frequencies')
```

```
## [1] "top frequencies"
```

```
sorted.omegas.NO[1:NO.pg.cutoff]
```

```
##  [1] 0.005208333 0.007812500 0.002604167 0.010416667 0.013020833
##  [6] 0.015625000 0.018229167 0.020833333 0.023437500 0.026041667
## [11] 0.028645833 0.148437500 0.138020833 0.151041667 0.140625000
```

```r
# visual
NO.pg.box <- boxplot(sorted.Ts.NO[1:NO.pg.cutoff], main="Period Boxplot")
```




```r
# the average influential period
print('mean of top periods')
```

```
## [1] "mean of top periods"
```

```r
NO.pg.box.mean <- NO.pg.box$stats[3]
print(NO.pg.box.mean)
```

```
## [1] 48
```

We found the frequencies of the largest spikes in the periodogram graph and converted them to periods by taking 1/the frequency. The 15 largest spikes corresponded to periods ranging from 6.62 to 384 days. We considered these our "top" choices for periods to explain seasonality.
The average influential period can be interpreted as on average, the seasons have a period of 48 days.

Create Models with Potential Periods

We created models with the top 3, 5, 10, and 15 periods, to determine which was best to explain the seasonality of the data. We did this because we predicted that the seasonality would be based on a complex wave, and we wanted to know how many different periods would be advantageous (i.e. explain more variance)

in a model.
We also created a model that used a period that was the average of several other periods that were close together. We noticed that top periods 8 through 11 were in the range of 34 to 48. We noticed that top periods 12 through 15 were all close to 7. We thought that since these sets of periods were clustered together, potentially their average would be better at explaining the variance in the model instead of adding separate predictors for each similar period.

```
# assign top periods to variables
NO.p1 <- sorted.Ts.NO[1]
NO.p2 <- sorted.Ts.NO[2]
NO.p3 <- sorted.Ts.NO[3]
NO.p4 <- sorted.Ts.NO[4]
NO.p5 <- sorted.Ts.NO[5]
NO.p6 <- sorted.Ts.NO[6]
NO.p7 <- sorted.Ts.NO[7]
NO.p8 <- sorted.Ts.NO[8]
NO.p9 <- sorted.Ts.NO[9]
NO.p10 <- sorted.Ts.NO[10]
NO.p11 <- sorted.Ts.NO[11]
NO.p12 <- sorted.Ts.NO[12]
NO.p13 <- sorted.Ts.NO[13]
NO.p14 <- sorted.Ts.NO[14]
NO.p15 <- sorted.Ts.NO[15]

NO.pavg1 <- mean(c(sorted.Ts.NO[8],sorted.Ts.NO[9], sorted.Ts.NO[10], sorted.Ts.NO[11]))
NO.pavg2 <- mean(c(sorted.Ts.NO[12],sorted.Ts.NO[13], sorted.Ts.NO[14], sorted.Ts.NO[15]))

# create time variable
time.NO<-c(1:length(NO.ts))
```

Model with Top 3 Periods

```
# model
NO.lm.top3 <- lm(NO.ts ~ sin(2*pi*time.NO/NO.p1) +
                         cos(2*pi*time.NO/NO.p1) +
                         sin(2*pi*time.NO/NO.p2) +
                         cos(2*pi*time.NO/NO.p2) +
                         sin(2*pi*time.NO/NO.p3) +
                         cos(2*pi*time.NO/NO.p3))
```

Model with Top 5 Periods

```
# model
NO.lm.top5 <- lm(NO.ts ~ sin(2*pi*time.NO/NO.p1) +
                         cos(2*pi*time.NO/NO.p1) +
                         sin(2*pi*time.NO/NO.p2) +
                         cos(2*pi*time.NO/NO.p2) +
                         sin(2*pi*time.NO/NO.p3) +
                         cos(2*pi*time.NO/NO.p3) +
                         sin(2*pi*time.NO/NO.p4) +
                         cos(2*pi*time.NO/NO.p4) +
                         sin(2*pi*time.NO/NO.p5) +
                         cos(2*pi*time.NO/NO.p5))
```

Model with Top 10 Periods

```
# model
NO.lm.top10 <- lm(NO.ts ~ sin(2*pi*time.NO/NO.p1) +
                          cos(2*pi*time.NO/NO.p1) +
                          sin(2*pi*time.NO/NO.p2) +
                          cos(2*pi*time.NO/NO.p2) +
                          sin(2*pi*time.NO/NO.p3) +
                          cos(2*pi*time.NO/NO.p3) +
                          sin(2*pi*time.NO/NO.p4) +
                          cos(2*pi*time.NO/NO.p4) +
                          sin(2*pi*time.NO/NO.p5) +
                          cos(2*pi*time.NO/NO.p5) +
                          sin(2*pi*time.NO/NO.p6) +
                          cos(2*pi*time.NO/NO.p6) +
                          sin(2*pi*time.NO/NO.p7) +
                          cos(2*pi*time.NO/NO.p7) +
                          sin(2*pi*time.NO/NO.p8) +
                          cos(2*pi*time.NO/NO.p8) +
                          sin(2*pi*time.NO/NO.p9) +
                          cos(2*pi*time.NO/NO.p9) +
                          sin(2*pi*time.NO/NO.p10) +
                          cos(2*pi*time.NO/NO.p10))
```

Model with Top 15 Periods

```
# model
NO.lm.top15 <- lm(NO.ts ~ sin(2*pi*time.NO/NO.p1) +
                          cos(2*pi*time.NO/NO.p1) +
                          sin(2*pi*time.NO/NO.p2) +
                          cos(2*pi*time.NO/NO.p2) +
                          sin(2*pi*time.NO/NO.p3) +
                          cos(2*pi*time.NO/NO.p3) +
                          sin(2*pi*time.NO/NO.p4) +
                          cos(2*pi*time.NO/NO.p4) +
                          sin(2*pi*time.NO/NO.p5) +
                          cos(2*pi*time.NO/NO.p5) +
                          sin(2*pi*time.NO/NO.p6) +
                          cos(2*pi*time.NO/NO.p6) +
                          sin(2*pi*time.NO/NO.p7) +
                          cos(2*pi*time.NO/NO.p7) +
                          sin(2*pi*time.NO/NO.p8) +
                          cos(2*pi*time.NO/NO.p8) +
                          sin(2*pi*time.NO/NO.p9) +
                          cos(2*pi*time.NO/NO.p9) +
                          sin(2*pi*time.NO/NO.p10) +
                          cos(2*pi*time.NO/NO.p10) +
                          sin(2*pi*time.NO/NO.p11) +
                          cos(2*pi*time.NO/NO.p11) +
                          sin(2*pi*time.NO/NO.p12) +
                          cos(2*pi*time.NO/NO.p12) +
                          sin(2*pi*time.NO/NO.p13) +
                          cos(2*pi*time.NO/NO.p13) +
                          sin(2*pi*time.NO/NO.p14) +
```

```
                          cos(2*pi*time.NO/NO.p14) +
                          sin(2*pi*time.NO/NO.p15) +
                          cos(2*pi*time.NO/NO.p15))
```

Model with Averaged Periods

```
NO.lm.combined <- lm(NO.ts ~ sin(2*pi*time.NO/NO.p1) +
                          cos(2*pi*time.NO/NO.p1) +
                          sin(2*pi*time.NO/NO.p2) +
                          cos(2*pi*time.NO/NO.p2) +
                          sin(2*pi*time.NO/NO.p3) +
                          cos(2*pi*time.NO/NO.p3) +
                          sin(2*pi*time.NO/NO.p4) +
                          cos(2*pi*time.NO/NO.p4) +
                          sin(2*pi*time.NO/NO.p5) +
                          cos(2*pi*time.NO/NO.p5) +
                          sin(2*pi*time.NO/NO.p6) +
                          cos(2*pi*time.NO/NO.p6) +
                          sin(2*pi*time.NO/NO.p7) +
                          cos(2*pi*time.NO/NO.p7) +
                          sin(2*pi*time.NO/NO.pavg1) +
                          cos(2*pi*time.NO/NO.pavg1) +
                          sin(2*pi*time.NO/NO.pavg2) +
                          cos(2*pi*time.NO/NO.pavg2))
```

Model Comparison

```
anova(NO.lm.top3, NO.lm.top5)
```

```
## Analysis of Variance Table
##
## Model 1: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3)
## Model 2: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    377 553038
## 2    373 497754  4     55284 10.357 5.789e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# small p-value suggests that the larger model performs better than the smaller model

anova(NO.lm.top5, NO.lm.top10)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5)
## Model 2: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5) +
##     sin(2 * pi * time.NO/NO.p6) + cos(2 * pi * time.NO/NO.p6) +
##     sin(2 * pi * time.NO/NO.p7) + cos(2 * pi * time.NO/NO.p7) +
##     sin(2 * pi * time.NO/NO.p8) + cos(2 * pi * time.NO/NO.p8) +
##     sin(2 * pi * time.NO/NO.p9) + cos(2 * pi * time.NO/NO.p9) +
##     sin(2 * pi * time.NO/NO.p10) + cos(2 * pi * time.NO/NO.p10)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    373 497754
## 2    363 421534 10     76220 6.5636 2.357e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# small p-value suggests that the larger model performs better than the smaller model

anova(NO.lm.top10, NO.lm.top15)
```

```
## Analysis of Variance Table
##
## Model 1: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5) +
##     sin(2 * pi * time.NO/NO.p6) + cos(2 * pi * time.NO/NO.p6) +
##     sin(2 * pi * time.NO/NO.p7) + cos(2 * pi * time.NO/NO.p7) +
##     sin(2 * pi * time.NO/NO.p8) + cos(2 * pi * time.NO/NO.p8) +
##     sin(2 * pi * time.NO/NO.p9) + cos(2 * pi * time.NO/NO.p9) +
##     sin(2 * pi * time.NO/NO.p10) + cos(2 * pi * time.NO/NO.p10)
## Model 2: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5) +
##     sin(2 * pi * time.NO/NO.p6) + cos(2 * pi * time.NO/NO.p6) +
##     sin(2 * pi * time.NO/NO.p7) + cos(2 * pi * time.NO/NO.p7) +
##     sin(2 * pi * time.NO/NO.p8) + cos(2 * pi * time.NO/NO.p8) +
##     sin(2 * pi * time.NO/NO.p9) + cos(2 * pi * time.NO/NO.p9) +
##     sin(2 * pi * time.NO/NO.p10) + cos(2 * pi * time.NO/NO.p10) +
##     sin(2 * pi * time.NO/NO.p11) + cos(2 * pi * time.NO/NO.p11) +
##     sin(2 * pi * time.NO/NO.p12) + cos(2 * pi * time.NO/NO.p12) +
##     sin(2 * pi * time.NO/NO.p13) + cos(2 * pi * time.NO/NO.p13) +
##     sin(2 * pi * time.NO/NO.p14) + cos(2 * pi * time.NO/NO.p14) +
##     sin(2 * pi * time.NO/NO.p15) + cos(2 * pi * time.NO/NO.p15)
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
```

```
## 1    363 421534
## 2    353 411452 10     10082 0.865 0.5664
```

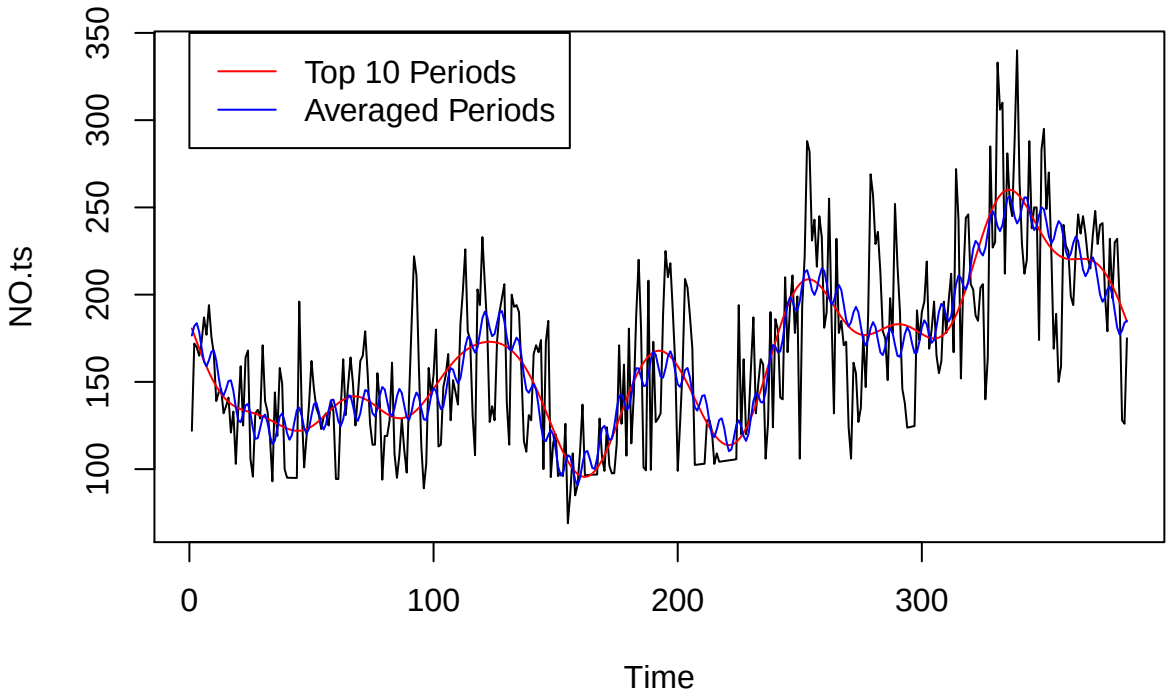
```
# large p-value suggests that the smaller model performs better

anova(NO.lm.top10, NO.lm.combined)
```

```
## Analysis of Variance Table
##
## Model 1: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5) +
##     sin(2 * pi * time.NO/NO.p6) + cos(2 * pi * time.NO/NO.p6) +
##     sin(2 * pi * time.NO/NO.p7) + cos(2 * pi * time.NO/NO.p7) +
##     sin(2 * pi * time.NO/NO.p8) + cos(2 * pi * time.NO/NO.p8) +
##     sin(2 * pi * time.NO/NO.p9) + cos(2 * pi * time.NO/NO.p9) +
##     sin(2 * pi * time.NO/NO.p10) + cos(2 * pi * time.NO/NO.p10)
## Model 2: NO.ts ~ sin(2 * pi * time.NO/NO.p1) + cos(2 * pi * time.NO/NO.p1) +
##     sin(2 * pi * time.NO/NO.p2) + cos(2 * pi * time.NO/NO.p2) +
##     sin(2 * pi * time.NO/NO.p3) + cos(2 * pi * time.NO/NO.p3) +
##     sin(2 * pi * time.NO/NO.p4) + cos(2 * pi * time.NO/NO.p4) +
##     sin(2 * pi * time.NO/NO.p5) + cos(2 * pi * time.NO/NO.p5) +
##     sin(2 * pi * time.NO/NO.p6) + cos(2 * pi * time.NO/NO.p6) +
##     sin(2 * pi * time.NO/NO.p7) + cos(2 * pi * time.NO/NO.p7) +
##     sin(2 * pi * time.NO/NO.pavg1) + cos(2 * pi * time.NO/NO.pavg1) +
##     sin(2 * pi * time.NO/NO.pavg2) + cos(2 * pi * time.NO/NO.pavg2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    363 421534
## 2    365 424611 -2   -3077.2 1.3249 0.2671
```

```
# large p-value suggests that the smaller model performs better
```

The partial F tests suggest that the model that includes the top 10 periods is better than the models with 3, 5, and 15 periods. The partial F test also suggests that the model with the averaged periods is better than the model with the top 10 periods. Both the model with 10 periods and the model with averaged periods will be further compared through metrics and diagnostics later, after we determine if there is a significant trend to be included.

```
# visualize
plot(NO.ts)
lines(NO.lm.top10$fitted.values, col = "red")
lines(NO.lm.combined$fitted.values, col = "blue")

legend(0, 350, legend = c("Top 10 Periods", "Averaged Periods"), col = c("red", "blue"), lwd = 1)
```



Upon visual inspection, both models get the basic shape of the time series correct, though the model with the averaged periods has smaller fluctuations that may better match the actual data.

Trend

Model Time Series Based on Time

```
# trend model
NO.lm.trend <- lm(NO.ts ~ time.NO)

# summary analysis
summary(NO.lm.trend)
```

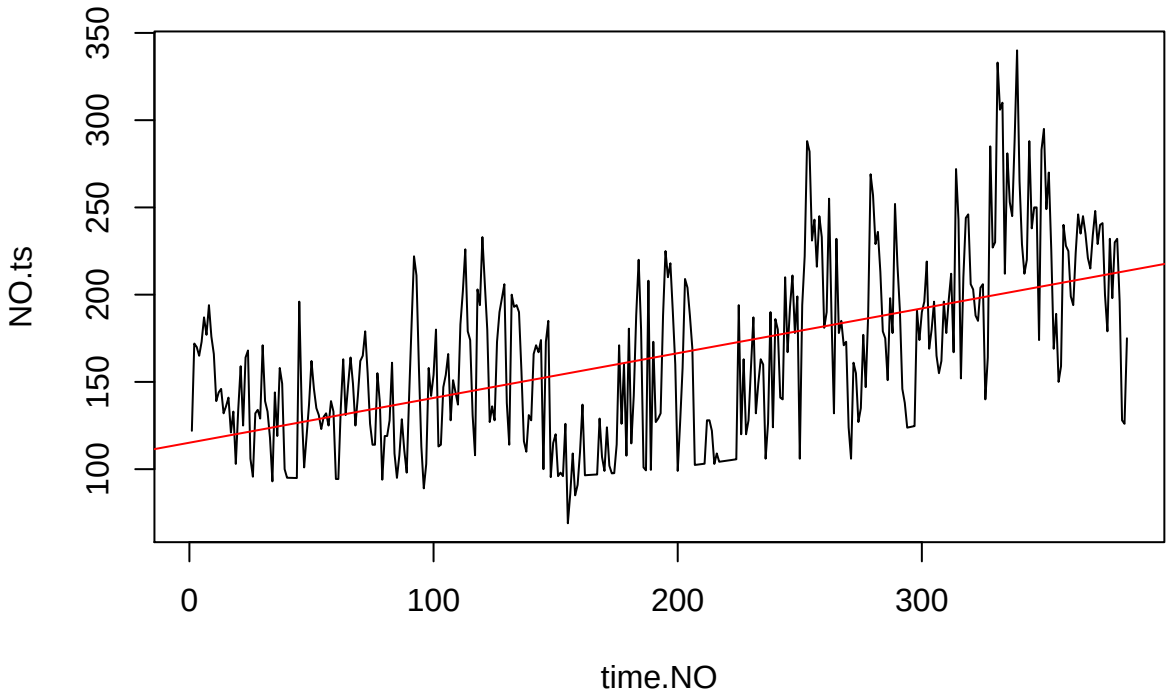
```
##
## Call:
## lm(formula = NO.ts ~ time.NO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -87.389 -34.365   2.159  27.847 137.895
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115.16473    4.40111   26.17   <2e-16 ***
```

```
## time.NO       0.25646    0.01981   12.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.04 on 382 degrees of freedom
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3031
## F-statistic: 167.6 on 1 and 382 DF,  p-value: < 2.2e-16
```

The p-value is significant at the 0.05 level, which means that the trend is significant.

Visualize Trend

```
plot(time.NO, NO.ts, type = "l")
abline(NO.lm.trend, col = "red")
```



As we noticed earlier from looking at the time series, there is an upward trend to the data.

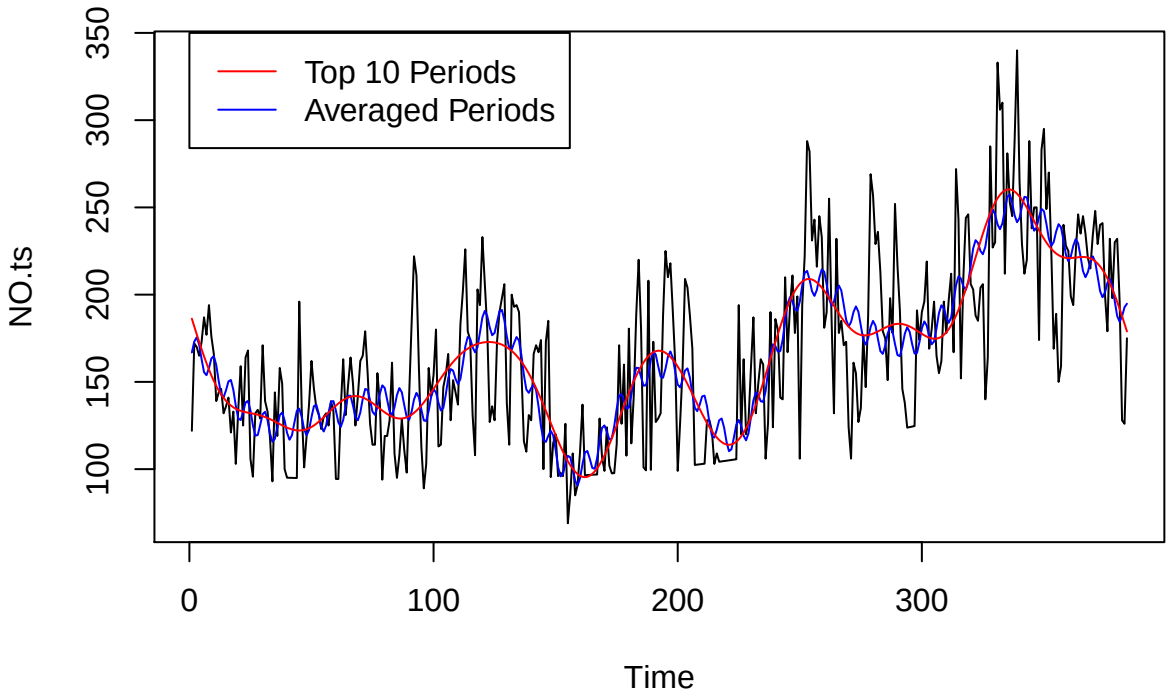
Model Comparison: Trend and Seasonality Together

```
# model with trend + averaged periods
NO.seasonalavg.trend <- lm(NO.ts ~ time.NO + sin(2*pi*time.NO/NO.p1) +
                           cos(2*pi*time.NO/NO.p1) +
                           sin(2*pi*time.NO/NO.p2) +
```

```
                          cos(2*pi*time.NO/NO.p2) +
                          sin(2*pi*time.NO/NO.p3) +
                          cos(2*pi*time.NO/NO.p3) +
                          sin(2*pi*time.NO/NO.p4) +
                          cos(2*pi*time.NO/NO.p4) +
                          sin(2*pi*time.NO/NO.p5) +
                          cos(2*pi*time.NO/NO.p5) +
                          sin(2*pi*time.NO/NO.p6) +
                          cos(2*pi*time.NO/NO.p6) +
                          sin(2*pi*time.NO/NO.p7) +
                          cos(2*pi*time.NO/NO.p7) +
                          sin(2*pi*time.NO/NO.pavg1) +
                          cos(2*pi*time.NO/NO.pavg1) +
                          sin(2*pi*time.NO/NO.pavg2) +
                          cos(2*pi*time.NO/NO.pavg2))

# model with trend + 10 periods
NO.seasonal10.trend <- lm(NO.ts ~ time.NO + sin(2*pi*time.NO/NO.p1) +
                          cos(2*pi*time.NO/NO.p1) +
                          sin(2*pi*time.NO/NO.p2) +
                          cos(2*pi*time.NO/NO.p2) +
                          sin(2*pi*time.NO/NO.p3) +
                          cos(2*pi*time.NO/NO.p3) +
                          sin(2*pi*time.NO/NO.p4) +
                          cos(2*pi*time.NO/NO.p4) +
                          sin(2*pi*time.NO/NO.p5) +
                          cos(2*pi*time.NO/NO.p5) +
                          sin(2*pi*time.NO/NO.p6) +
                          cos(2*pi*time.NO/NO.p6) +
                          sin(2*pi*time.NO/NO.p7) +
                          cos(2*pi*time.NO/NO.p7) +
                          sin(2*pi*time.NO/NO.p8) +
                          cos(2*pi*time.NO/NO.p8) +
                          sin(2*pi*time.NO/NO.p9) +
                          cos(2*pi*time.NO/NO.p9) +
                          sin(2*pi*time.NO/NO.p10) +
                          cos(2*pi*time.NO/NO.p10))
```

Visual Comparison

```
plot(NO.ts)
lines(NO.seasonalavg.trend$fitted.values, col = "blue")
lines(NO.seasonal10.trend$fitted.values, col = "red")
legend(0, 350, legend = c("Top 10 Periods", "Averaged Periods"), col = c("red", "blue"), lwd = 1)
```



Based on visual inspection, the model with trend and averaged periods appears to better capture the fluctuations of the data, though both models match the general trend of the data.

Comparison Using Metrics

```
# Adjusted R^2: model with trend and averaged periods
summary(NO.seasonalavg.trend)$adj.r.squared
```

```
## [1] 0.5622059
```

```
# AIC: model with trend and averaged periods
AIC(NO.seasonalavg.trend)
```

```
## [1] 3821.945
```

```
# Adjusted R^2: model with trend and 10 periods
summary(NO.seasonal10.trend)$adj.r.squared
```

```
## [1] 0.5621247
```

```
# AIC: model with trend and 10 periods
AIC(NO.seasonal10.trend)
```

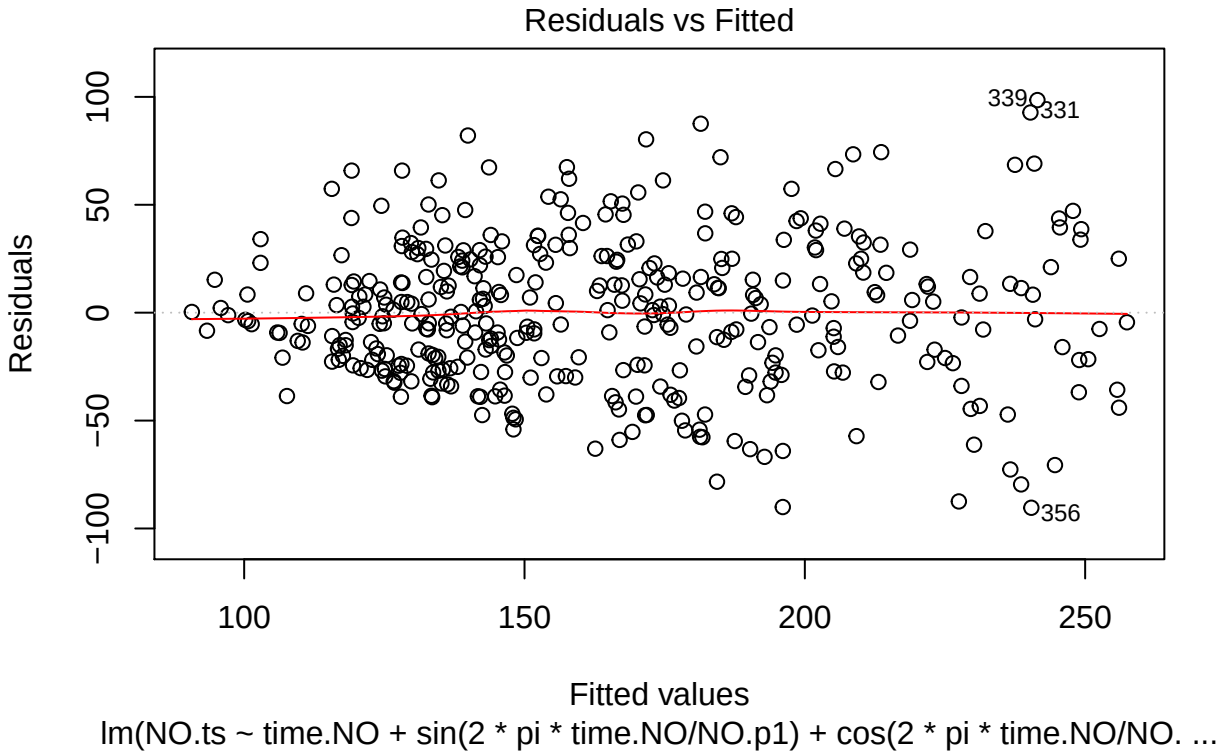
```
## [1] 3823.9
```

The adjusted R^2 of the model that includes the trend and averaged periods is 0.5622. The adjusted R^2 of the model that includes the trend and 10 periods is 0.5621. Based on adjusted R^2, the models are essentially equivalent.
The AIC of the model with averaged periods is 3821.945. The AIC of the model with 10 periods is 3823.9. Based on AIC, we would choose the model with averaged periods, because it has the smaller AIC value. The values are very similar, though.

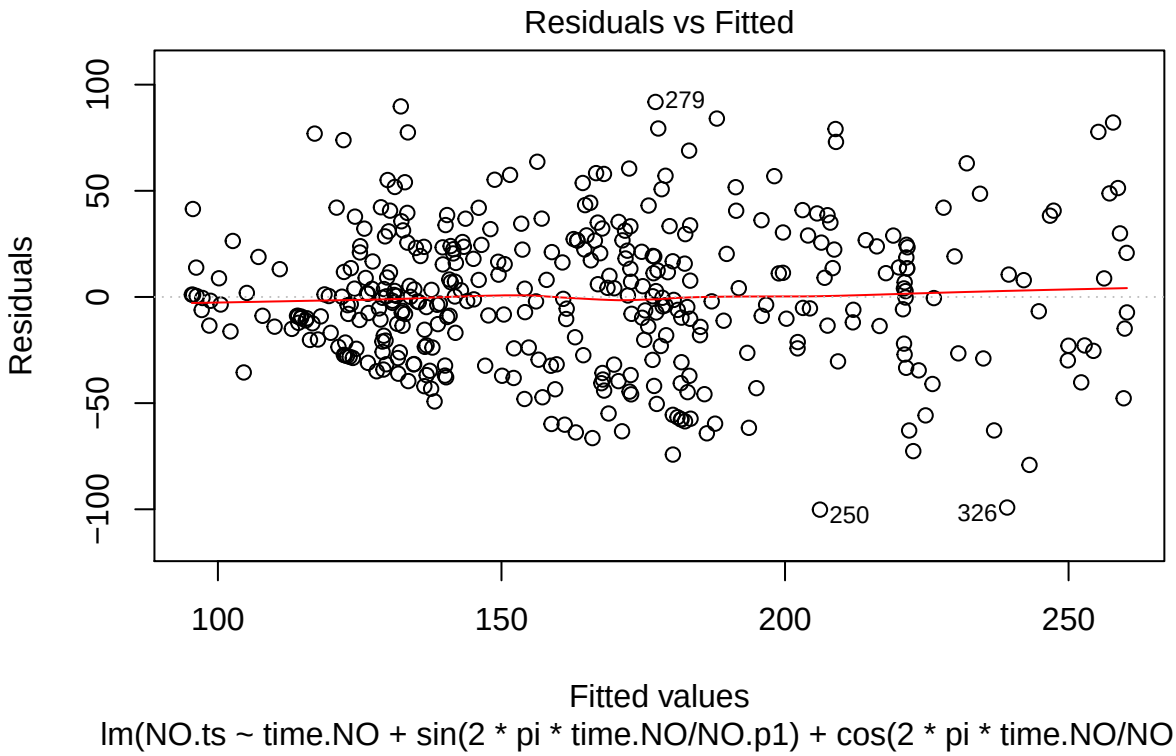
Diagnostics

Residuals vs. Fitted

```
plot(NO.seasonalavg.trend, which = 1)
```



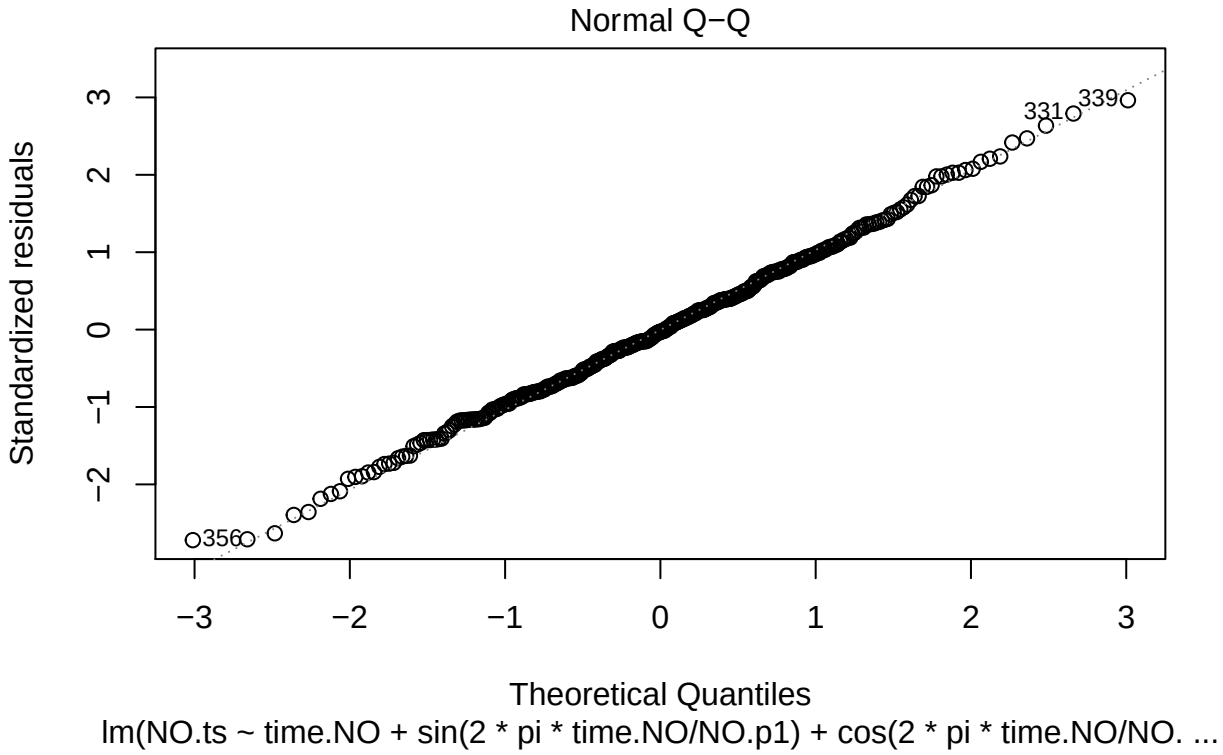

```
plot(NO.seasonal10.trend, which = 1)
```



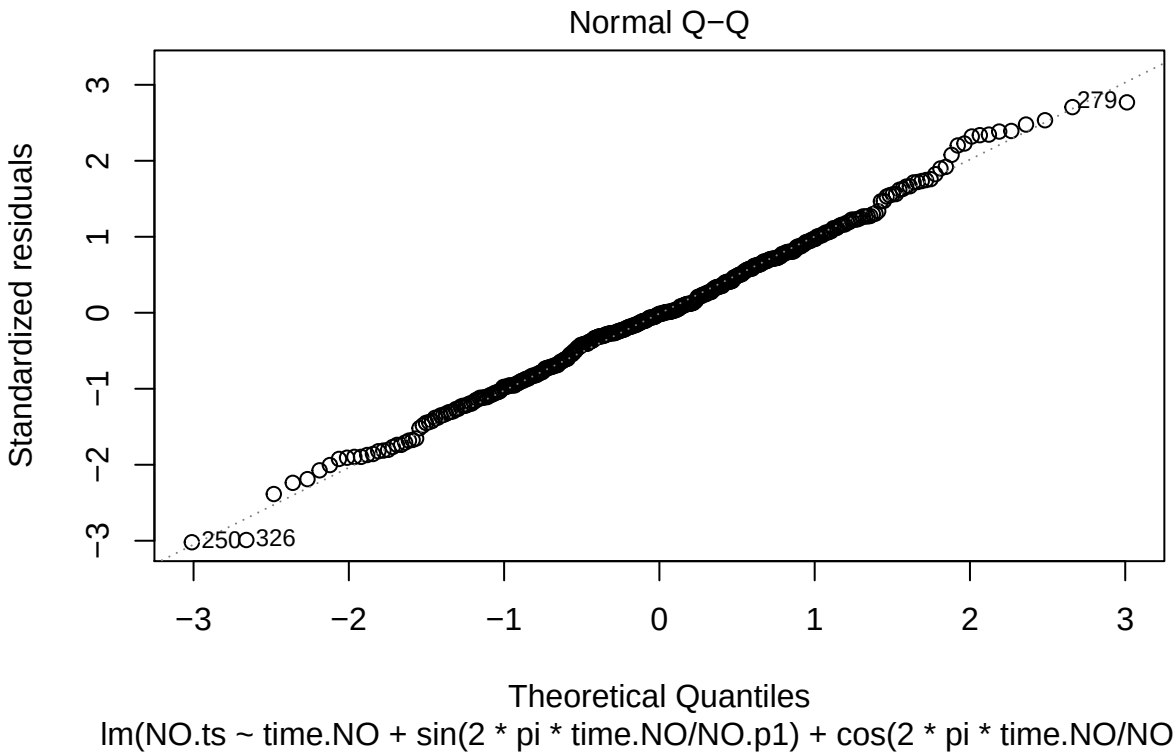
The mean of the residuals is approximately 0 and has fairly even spread above and below the x-axis for both models. The variance appears to have a slight trend, where there is less spread above and below the x-axis to the left and slightly more on the right.

QQ Plot

```
plot(NO.seasonalavg.trend, which = 2)
```



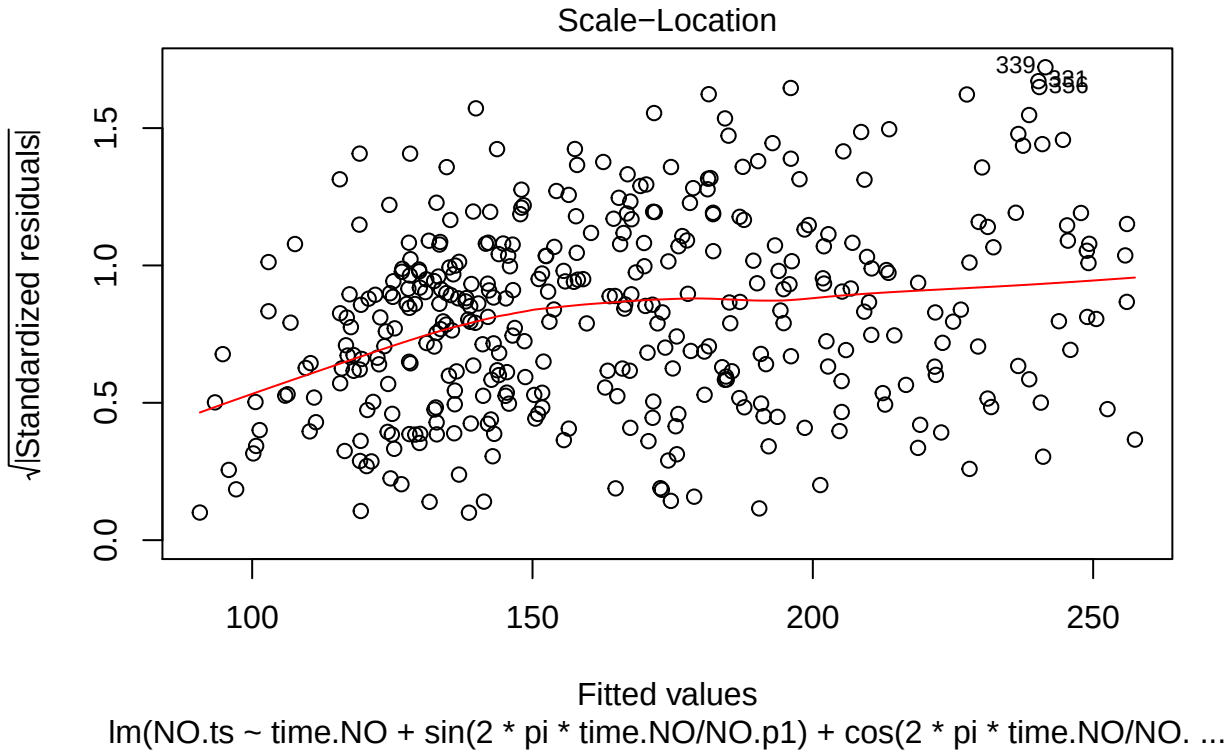

```
plot(NO.seasonal10.trend, which = 2)
```



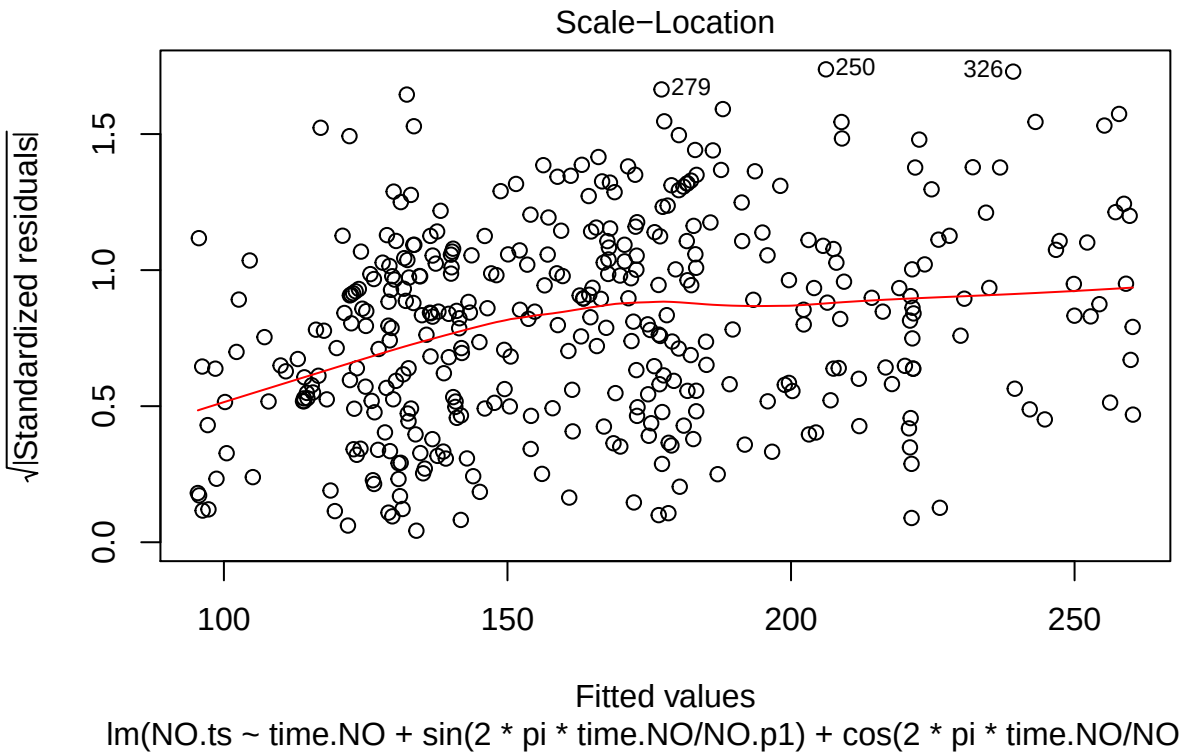
The residuals are approximately normal for both models, though the tails show some deviation from normality. The model with averaged periods appears to have slightly more normal residuals at the tails.

Scale-Location Plot

```
plot(NO.seasonalavg.trend, which = 3)
```



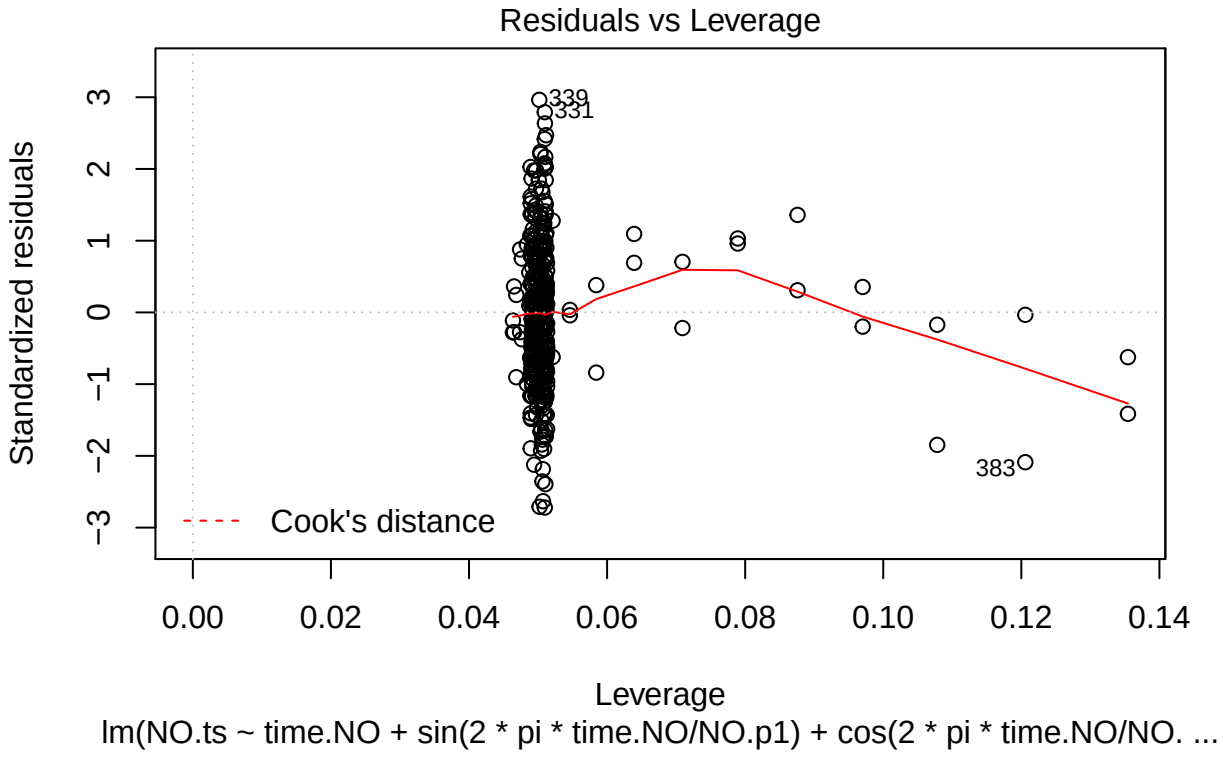

```
plot(NO.seasonal10.trend, which = 3)
```



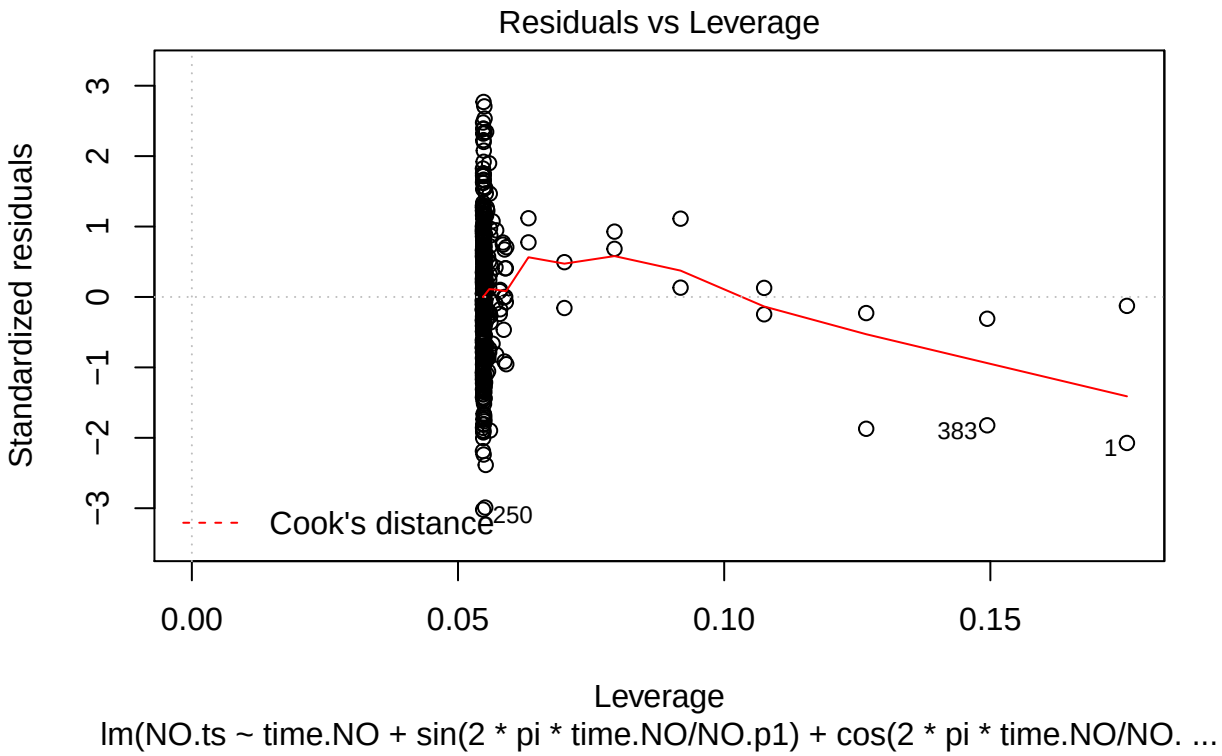
While the mean is not centered at 0 for either plot, there appears to be relatively good spread in both plots.

Residuals vs. Leverage

```
plot(NO.seasonalavg.trend, which = 5)
```

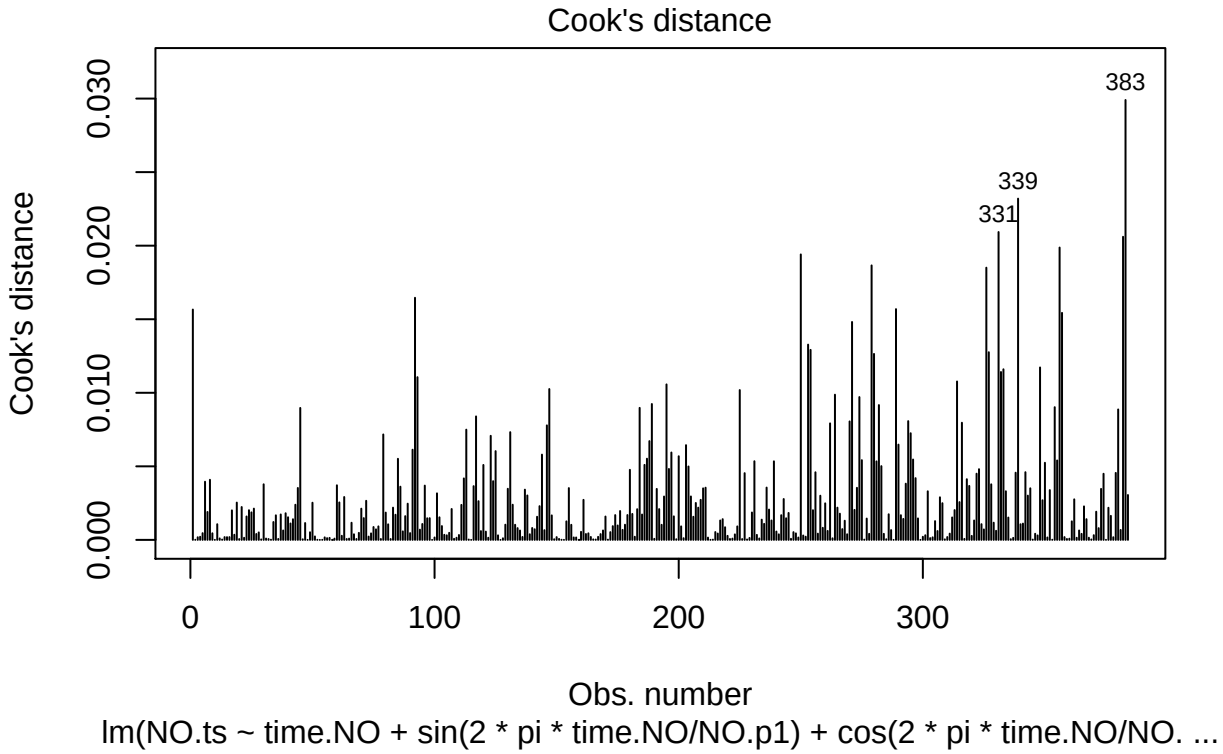



```
plot(NO.seasonal10.trend, which = 5)
```

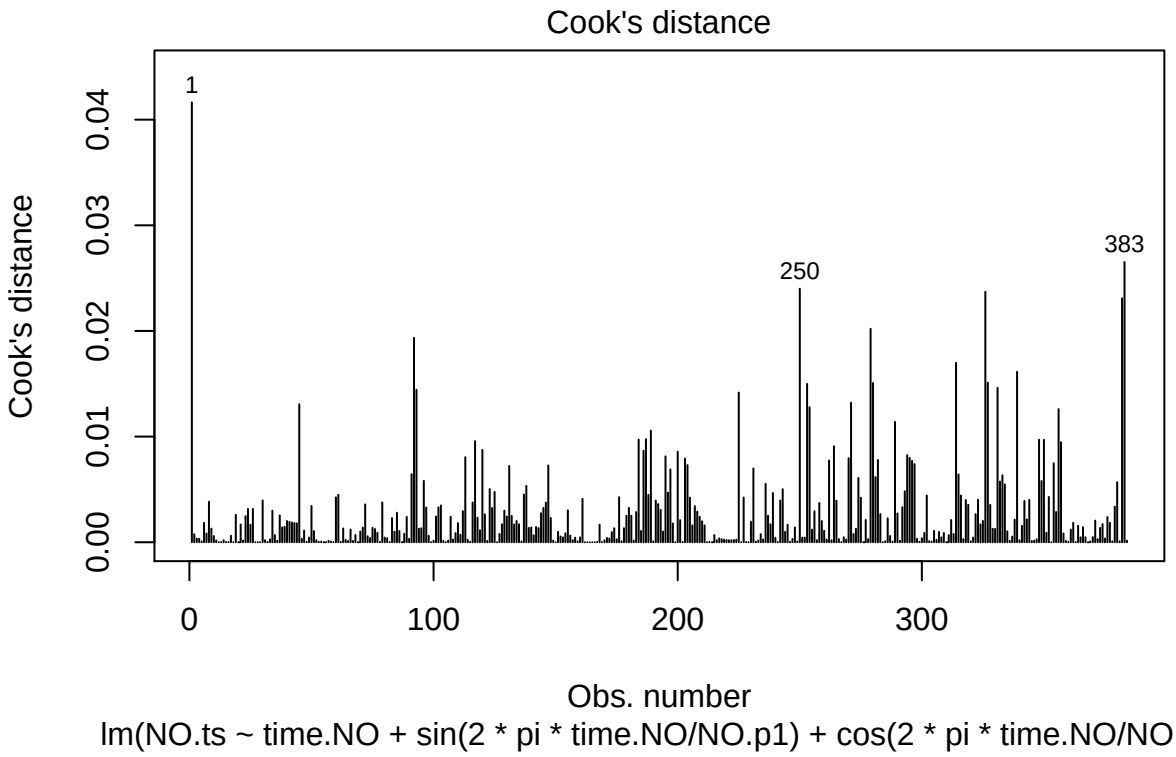


Neither plot has any points with Cook's distances greater than 0.5.

```r
# plot Cook's distances
plot(NO.seasonalavg.trend,labels.id = NULL, which = 4)
```




```r
plot(NO.seasonal10.trend,labels.id = NULL, which = 4)
```



Both models have low Cook's distances for all points.

Both models perform similarly in the diagnostics. We do not feel that either model needs a log or Box Cox transformation to improve the diagnostics.

Choose Trend + Seasonal Model

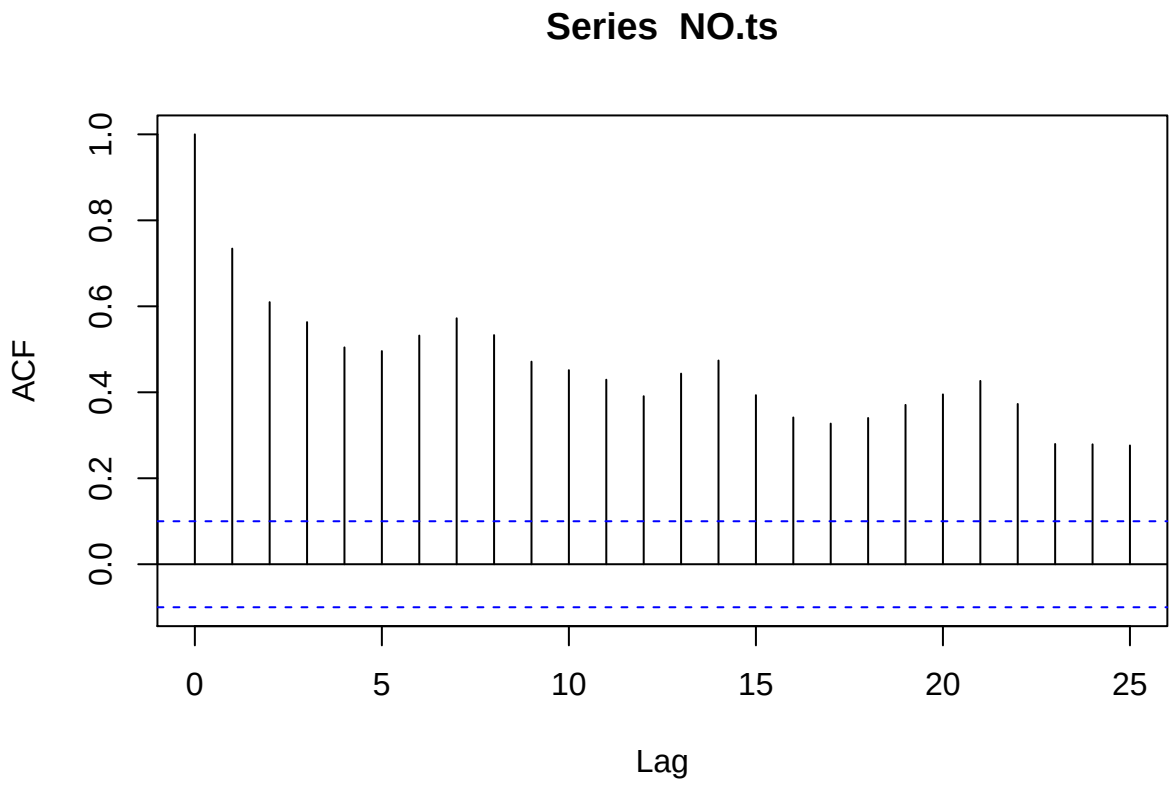
```
NO.lm <- NO.seasonalavg.trend
```

We have chosen the model with the trend and averaged periods to explain seasonality, as it performed slightly better in AIC than the model with trend and 10 periods, as well as being preferred after the partial F test. Since the diagnostics for both models were similar, AIC and the partial F test were used to make the decision. Though not a deciding factor in our choice, this model also has the advantage of having somewhat more normal residuals in the tails of the QQ plot.

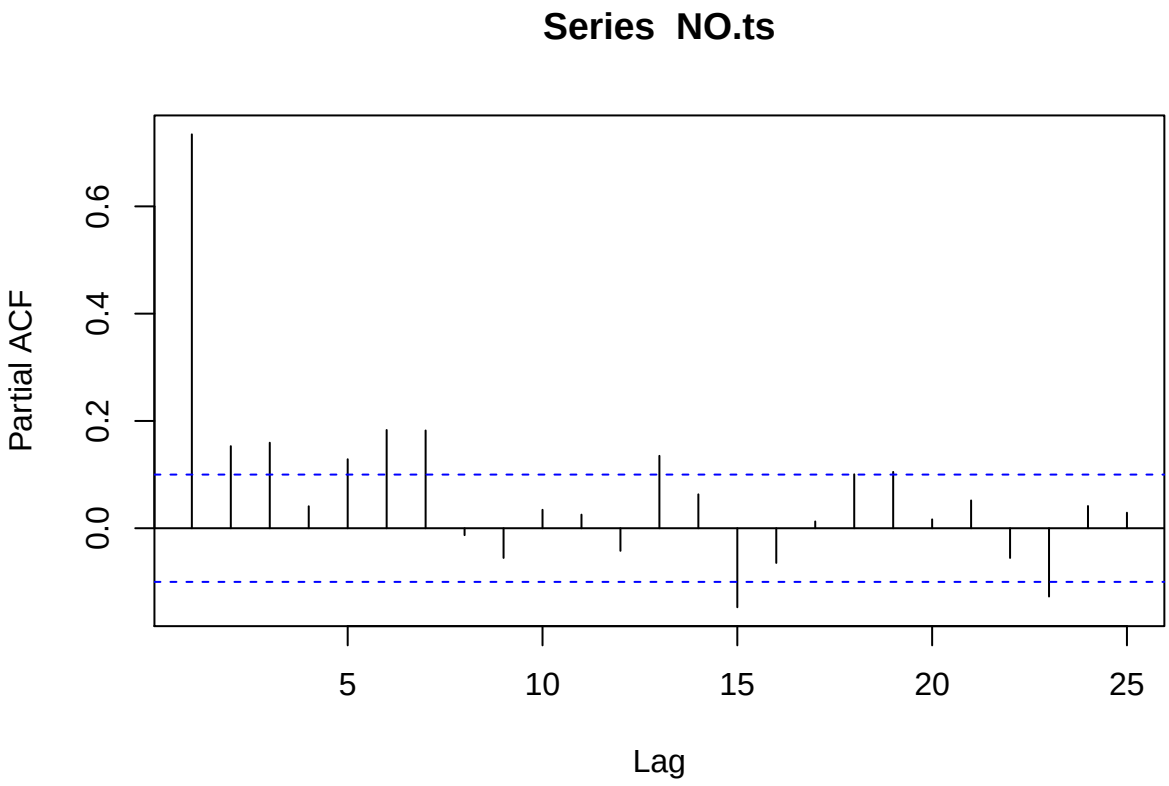
Model Residuals

ACF and PACF

```
# ACF and PACF
acf(NO.ts)
```




```
pacf(NO.ts)
```



The ACF has somewhat sinusoidal behavior and is significant for all 25 lags in view on the plot. The first few lags could be seen as having linear decay, suggesting that the time series may not be stationary and the first differences may need to be taken to meet the stationarity assumption. Values for d of 0 and 1 will be tested in potential models.

The PACF does not cut off after a certain number of lags and also has somewhat sinusoidal behavior. We predict that there will be autoregressive and moving average terms in the model of the residuals.
Based on the PACF, a potential value for p for the autoregressive portion of the model is 3. We will also test values of 1 and 2 for p, since they are the other significant lags before the first cutoff in the PACF.
Based on the ACF, a potential value for q for the moving average portion of the model is 4. This is the point where the ACF stops decreasing, before increasing again, for the first time. Values of 1, 2, and 3 will also be tested for q, as these lags are also significant before the lag of 4.

ARIMA Models

We tested the possible combinations of our p, d, and q values.

```
# get residuals
e.ts.NO <-ts(NO.lm$residuals)

# 1, 0, 1
NO.arima101 <- arima(e.ts.NO, order = c(1,0,1), include.mean = FALSE)
summary(NO.arima101)
```

```
##
## Call:
```

```
## arima(x = e.ts.NO, order = c(1, 0, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1     ma1
##       0.2752  0.1179
## s.e.  0.1392  0.1456
##
## sigma^2 estimated as 945:  log likelihood = -1860.39,  aic = 3726.77
##
## Training set error measures:
##                        ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.004790035 30.74157 24.23745 86.79421 137.8758 0.8724626
##                      ACF1
## Training set -0.001568528
```

```
# AIC = 3726.77

# 1, 0, 2
NO.arima102 <- arima(e.ts.NO, order = c(1,0,2), include.mean = FALSE)
summary(NO.arima102)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1     ma1      ma2
##       0.3184  0.0755  -0.0207
## s.e.  0.2502  0.2505   0.1061
##
## sigma^2 estimated as 944.9:  log likelihood = -1860.37,  aic = 3728.74
##
## Training set error measures:
##                        ME  RMSE      MAE     MPE     MAPE      MASE
## Training set -0.005851891 30.74 24.23397 87.2972 137.6968 0.8723372
##                      ACF1
## Training set -0.001778526
```

```
# AIC = 3728.74

# 1, 0, 3
NO.arima103 <- arima(e.ts.NO, order = c(1,0,3), include.mean = FALSE)
summary(NO.arima103)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 0, 3), include.mean = FALSE)
##
## Coefficients:
##           ar1     ma1     ma2     ma3
##       -0.0635  0.4581  0.1542  0.1201
## s.e.   0.2860  0.2807  0.1122  0.0636
##
```

```
## sigma^2 estimated as 938.7:  log likelihood = -1859.11,  aic = 3728.22
##
## Training set error measures:
##                       ME     RMSE      MAE     MPE     MAPE      MASE
## Training set -0.01218995 30.63822 24.02608 87.8098 130.2569 0.8648541
##                      ACF1
## Training set -0.001581292
```

```
# AIC = 3728.22

# 1, 0, 4
NO.arima104 <- arima(e.ts.NO, order = c(1,0,4), include.mean = FALSE)
summary(NO.arima104)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 0, 4), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1      ma2      ma3      ma4
##       0.7857  -0.5115  -0.2365  -0.0367  -0.2153
## s.e.  0.0417   0.0581   0.0585   0.0571   0.0629
##
## sigma^2 estimated as 850.7:  log likelihood = -1842.06,  aic = 3696.11
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.03387313 29.16726 22.70924 78.75416 129.5468 0.8174523
##                     ACF1
## Training set 0.01653262
```

```
# AIC = 3696.11
# lowest

# 1, 1, 1
NO.arima111 <- arima(e.ts.NO, order = c(1,1,1), include.mean = FALSE)
summary(NO.arima111)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 1, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1
##       0.3799  -1.0000
## s.e.  0.0475   0.0066
##
## sigma^2 estimated as 949.2:  log likelihood = -1858.88,  aic = 3723.77
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3978709 30.76934 24.25538 79.26134 144.9515 0.8731077
##                     ACF1
## Training set 0.01577263
```

```r
# AIC = 3723.77

# 1, 1, 2
NO.arima112 <- arima(e.ts.NO, order = c(1,1,2), include.mean = FALSE)
summary(NO.arima112)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 1, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.2846  -0.8890  -0.111
## s.e.  0.1385   0.1451   0.145
##
## sigma^2 estimated as 947.5:  log likelihood = -1858.58,  aic = 3725.16
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3978857 30.74171 24.21762 79.47537 145.2347 0.8717486
##                       ACF1
## Training set 0.0003740974
```

```r
# AIC = 3725.16

# 1, 1, 3
NO.arima113 <- arima(e.ts.NO, order = c(1,1,3), include.mean = FALSE)
summary(NO.arima113)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 1, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1      ma2     ma3
##       0.3258  -0.9291  -0.0914  0.0205
## s.e.  0.2453   0.2454   0.1696  0.1044
##
## sigma^2 estimated as 947.4:  log likelihood = -1858.56,  aic = 3727.12
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3977028 30.74014 24.21454 79.99985 145.0683 0.8716378
##                       ACF1
## Training set 5.095235e-05
```

```r
# AIC = 3727.12

# 1, 1, 4
NO.arima114 <- arima(e.ts.NO, order = c(1,1,4), include.mean = FALSE)
summary(NO.arima114)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(1, 1, 4), include.mean = FALSE)
##
## Coefficients:
##          ar1      ma1      ma2      ma3      ma4
##       -0.0492  -0.5528  -0.2935  -0.0314  -0.1223
## s.e.   0.2780   0.2728   0.1846   0.0959   0.0639
##
## sigma^2 estimated as 941.2:  log likelihood = -1857.25,  aic = 3726.51
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE    MAPE      MASE
## Training set 0.3940524 30.63907 24.02045 80.69848 137.941 0.8646512
##                       ACF1
## Training set -0.0001311285
```

```
# AIC = 3726.51

# 2, 0, 1
NO.arima201 <- arima(e.ts.NO, order = c(2,0,1), include.mean = FALSE)
summary(NO.arima201)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 0, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1     ar2     ma1
##       -0.3978  0.2559  0.7936
## s.e.   0.2394  0.1164  0.2277
##
## sigma^2 estimated as 942.4:  log likelihood = -1859.85,  aic = 3727.69
##
## Training set error measures:
##                       ME     RMSE     MAE      MPE     MAPE      MASE
## Training set -0.01030572 30.69773 24.1311 88.55227 132.7114 0.8686343
##                      ACF1
## Training set -0.001055449
```

```
# AIC = 3727.69

# 2, 0, 2
NO.arima202 <- arima(e.ts.NO, order = c(2,0,2), include.mean = FALSE)
summary(NO.arima202)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1     ar2     ma1     ma2
##       -0.4103  0.2571  0.8061  0.0038
```

```
## s.e.   0.3640  0.1231  0.3638  0.1513
##
## sigma^2 estimated as 942.4:  log likelihood = -1859.85,  aic = 3729.69
##
## Training set error measures:
##                       ME     RMSE      MAE    MPE     MAPE      MASE
## Training set -0.01016021 30.69779 24.13135 88.509 132.7257 0.8686431
##                      ACF1
## Training set -0.001098245
```

```
# AIC = 3729.69

# 2, 0, 3
NO.arima203 <- arima(e.ts.NO, order = c(2,0,3), include.mean = FALSE)
summary(NO.arima203)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 0, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3
##       0.9586  -0.8645  -0.5849  0.6460  0.3138
## s.e.  0.0412   0.0610   0.0631  0.0579  0.0596
##
## sigma^2 estimated as 894.9:  log likelihood = -1850.4,  aic = 3712.8
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.007454374 29.91516 23.39951 75.33624 137.8154 0.8422995
##                      ACF1
## Training set -0.006028335
```

```
# AIC = 3712.8

# 2, 0, 4
NO.arima204 <- arima(e.ts.NO, order = c(2,0,4), include.mean = FALSE)
summary(NO.arima204)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 0, 4), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3      ma4
##       0.9682  -0.8511  -0.5996  0.6185  0.3201  -0.0310
## s.e.  0.0569   0.0606   0.0765  0.0853  0.0632   0.0702
##
## sigma^2 estimated as 894.7:  log likelihood = -1850.3,  aic = 3714.6
##
## Training set error measures:
##                       ME    RMSE      MAE      MPE     MAPE     MASE
## Training set 0.009300056 29.9122 23.38371 76.97562 138.1145 0.841731
```

```
##                       ACF1
## Training set -0.0007867555
```

```
# AIC = 3714.6

# 2, 1, 1
NO.arima211 <- arima(e.ts.NO, order = c(2,1,1), include.mean = FALSE)
summary(NO.arima211)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 1, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.3945  -0.0391  -1.0000
## s.e.  0.0512   0.0514   0.0066
##
## sigma^2 estimated as 947.6:  log likelihood = -1858.6,  aic = 3725.19
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE     MASE
## Training set 0.3980231 30.74277 24.21871 79.07055 145.3024 0.871788
##                       ACF1
## Training set 0.0009342009
```

```
# AIC = 3725.19

# 2, 1, 2
NO.arima212 <- arima(e.ts.NO, order = c(2,1,2), include.mean = FALSE)
summary(NO.arima212)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 1, 2), include.mean = FALSE)
##
## Coefficients:
##           ar1     ar2      ma1      ma2
##       -0.3923  0.2582  -0.2085  -0.7915
## s.e.   0.2435  0.1183   0.2323   0.2322
##
## sigma^2 estimated as 944.8:  log likelihood = -1858.02,  aic = 3726.05
##
## Training set error measures:
##                      ME     RMSE      MAE     MPE     MAPE      MASE
## Training set 0.3979693 30.69783 24.11821 81.2856 140.3993 0.8681702
##                       ACF1
## Training set 0.0001632674
```

```
# AIC = 3726.05

# 2, 1, 3
```

```r
NO.arima213 <- arima(e.ts.NO, order = c(2,1,3), include.mean = FALSE)
summary(NO.arima213)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 1, 3), include.mean = FALSE)
##
## Coefficients:
##           ar1     ar2      ma1      ma2     ma3
##       -0.3691  0.2596  -0.2322  -0.7780  0.0102
## s.e.   0.3888  0.1300   0.3884   0.2915  0.1464
##
## sigma^2 estimated as 944.8:  log likelihood = -1858.02,  aic = 3728.05
##
## Training set error measures:
##                      ME   RMSE      MAE      MPE     MAPE     MASE
## Training set 0.3978561 30.698 24.11909 81.39144 140.3215 0.868202
##                       ACF1
## Training set 0.0007934061
```

```r
# AIC = 3728.05

# 2, 1, 4
NO.arima214 <- arima(e.ts.NO, order = c(2,1,4), include.mean = FALSE)
summary(NO.arima214)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(2, 1, 4), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2      ma3      ma4
##       0.9577  -0.8616  -1.5818  1.2247  -0.3279  -0.3150
## s.e.  0.0420   0.0592   0.0638  0.1174   0.1021   0.0586
##
## sigma^2 estimated as 897.3:  log likelihood = -1848.64,  aic = 3711.28
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3663267 29.91608 23.40343 68.38442 144.2727 0.8424406
##                       ACF1
## Training set -0.003584311
```

```r
# AIC = 3711.28

# 3, 0, 1
NO.arima301 <- arima(e.ts.NO, order = c(3,0,1), include.mean = FALSE)
summary(NO.arima301)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 0, 1), include.mean = FALSE)
```

```
##
## Coefficients:
##           ar1     ar2     ar3     ma1
##       -0.3955  0.2550  0.0003  0.7913
## s.e.   0.2705  0.1231  0.0591  0.2655
##
## sigma^2 estimated as 942.4:  log likelihood = -1859.85,  aic = 3729.69
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.0103448 30.69773 24.13108 88.56441 132.7086 0.8686336
##                      ACF1
## Training set -0.00104043
```

```
# AIC = 3729.69

# 3, 0, 2
NO.arima302 <- arima(e.ts.NO, order = c(3,0,2), include.mean = FALSE)
summary(NO.arima302)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2
##       0.4568  0.6835  -0.3570  -0.1307  -0.8693
## s.e.  0.1900  0.2549   0.1021   0.1789   0.1788
##
## sigma^2 estimated as 855.1:  log likelihood = -1843.08,  aic = 3698.16
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.02415789 29.24239 22.79553 77.20828 136.9156 0.8205584
##                      ACF1
## Training set -0.03285108
```

```
# AIC = 3698.16
# second lowest

# 3, 0, 3
NO.arima303 <- arima(e.ts.NO, order = c(3,0,3), include.mean = FALSE)
summary(NO.arima303)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 0, 3), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     ma3
##       0.9975  -0.9405  0.0275  -0.6353  0.7040  0.3123
## s.e.  0.1777   0.1818  0.1691   0.1721  0.1641  0.1716
##
```

```
## sigma^2 estimated as 882.3:  log likelihood = -1849.93,  aic = 3713.87
##
## Training set error measures:
##                       ME     RMSE    MAE      MPE    MAPE      MASE
## Training set 0.008406046 29.70326 23.227 69.83109 139.318 0.8360898
##                      ACF1
## Training set -0.001326911
```

```
# AIC = 3713.87

# 3, 0, 4
NO.arima304 <- arima(e.ts.NO, order = c(3,0,4), include.mean = FALSE)
summary(NO.arima304)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 0, 4), include.mean = FALSE)
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##          ar1      ar2      ar3      ma1     ma2     ma3     ma4
##       0.8624  -0.8088  -0.0963  -0.5012  0.6177  0.4078  0.0408
## s.e.     NaN      NaN      NaN      NaN     NaN     NaN     NaN
##
## sigma^2 estimated as 882.3:  log likelihood = -1849.94,  aic = 3715.89
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.008694892 29.70326 23.22665 69.93911 139.1647 0.8360774
##                      ACF1
## Training set -0.0001923338
```

```
# NaNs produced

# 3, 1, 1
NO.arima311 <- arima(e.ts.NO, order = c(3,1,1), include.mean = FALSE)
summary(NO.arima311)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 1, 1), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ar3      ma1
##       0.3945  -0.0383  -0.0021  -1.0000
## s.e.  0.0512   0.0550   0.0515   0.0066
##
## sigma^2 estimated as 947.6:  log likelihood = -1858.59,  aic = 3727.19
##
## Training set error measures:
```

```
##                      ME     RMSE      MAE     MPE     MAPE      MASE
## Training set 0.3980312 30.74254 24.21785 78.8734 145.3082 0.8717569
##                       ACF1
## Training set 0.0007485001
```

```
# AIC = 3727.19

# 3, 1, 2
NO.arima312 <- arima(e.ts.NO, order = c(3,1,2), include.mean = FALSE)
summary(NO.arima312)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 1, 2), include.mean = FALSE)
##
## Coefficients:
##           ar1     ar2     ar3      ma1      ma2
##       -0.3860  0.2566  0.0033  -0.2150  -0.7850
## s.e.   0.2674  0.1223  0.0586   0.2626   0.2625
##
## sigma^2 estimated as 944.8:  log likelihood = -1858.02,  aic = 3728.04
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3979659 30.69808 24.11754 81.38899 140.3052 0.8681461
##                       ACF1
## Training set 0.0004978209
```

```
# AIC = 3728.04

# 3, 1, 3
NO.arima313 <- arima(e.ts.NO, order = c(3,1,3), include.mean = FALSE)
summary(NO.arima313)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 1, 3), include.mean = FALSE)
##
## Coefficients:
##           ar1     ar2     ar3      ma1      ma2      ma3
##       -0.5623  0.1695  0.0641  -0.0408  -0.8008  -0.1584
## s.e.   1.3920  0.6288  0.4168   1.3857   0.3407   1.1632
##
## sigma^2 estimated as 944.8:  log likelihood = -1858.01,  aic = 3730.01
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.3975231 30.69712 24.11692 81.50642 139.8093 0.8681238
##                       ACF1
## Training set 0.002792214
```

```r
# AIC = 3730.01

# 3, 1, 4
NO.arima314 <- arima(e.ts.NO, order = c(3,1,4), include.mean = FALSE)
summary(NO.arima314)
```

```
##
## Call:
## arima(x = e.ts.NO, order = c(3, 1, 4), include.mean = FALSE)
##
## Coefficients:
##           ar1      ar2     ar3      ma1     ma2      ma3      ma4
##       -0.0521  -0.8240  0.3936  -0.5359  0.5471  -0.9850  -0.0262
## s.e.   0.1110   0.0497  0.1111   0.1211  0.0643   0.0632   0.1192
##
## sigma^2 estimated as 876.5:  log likelihood = -1846.37,  aic = 3708.73
##
## Training set error measures:
##                      ME     RMSE     MAE      MPE    MAPE      MASE
## Training set 0.3412935 29.56792 23.2143 77.55653 149.145 0.8356326
##                      ACF1
## Training set 0.001925498
```

```r
# AIC = 3708.73
```

Based on AIC, the two best models are the ARIMA(1,0,4) and ARIMA(3,0,2) models.

After testing the possible values of p, d, and q that we identified, we also used the auto.arima function to generate another model.

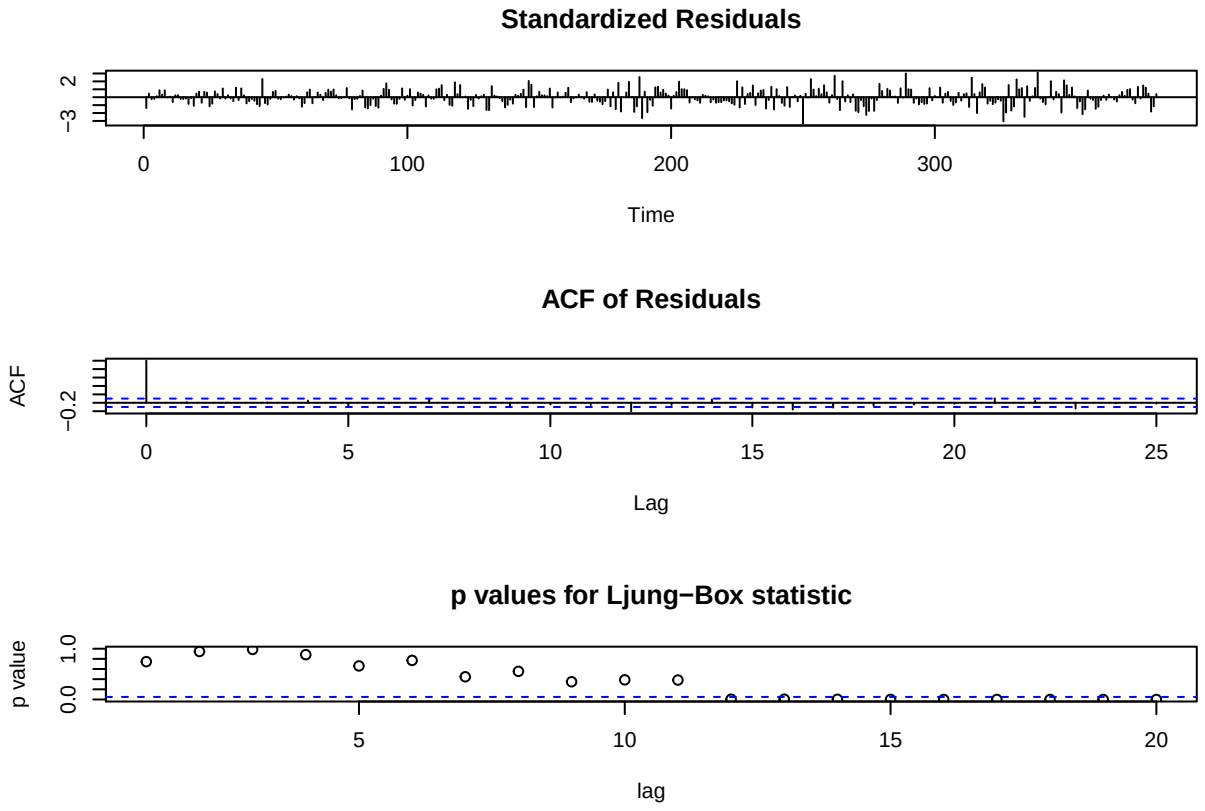
```r
# auto.arima model
NO.residuals.auto <- auto.arima(e.ts.NO, approximation = FALSE)

# summary
summary(NO.residuals.auto)
```

```
## Series: e.ts.NO
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.3764
## s.e.  0.0473
##
## sigma^2 estimated as 949.2:  log likelihood=-1860.73
## AIC=3725.46   AICc=3725.49   BIC=3733.36
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.01081192 30.76914 24.26731 86.49446 137.0982 0.8735374
##                    ACF1
## Training set 0.0145668
```

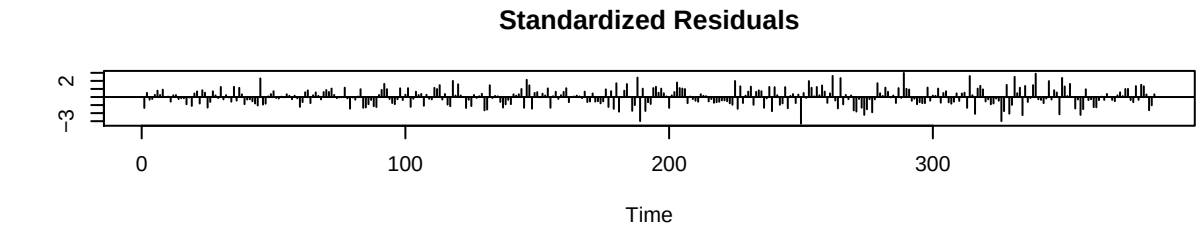
The ARIMA model is a 1,0,0 model. This means that there are autoregressive terms, where p = 1, but no moving average terms, since q = 0. Also, d = 0, meaning that the time series is stationary, which is a key assumption to modeling the residuals. The AIC for this model is 3725.46.

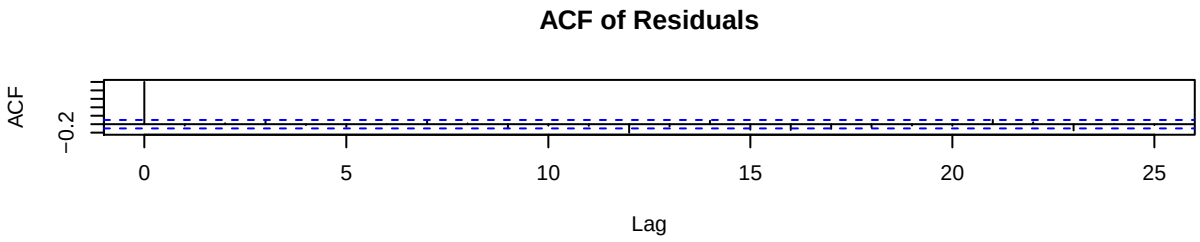
Diagnostics

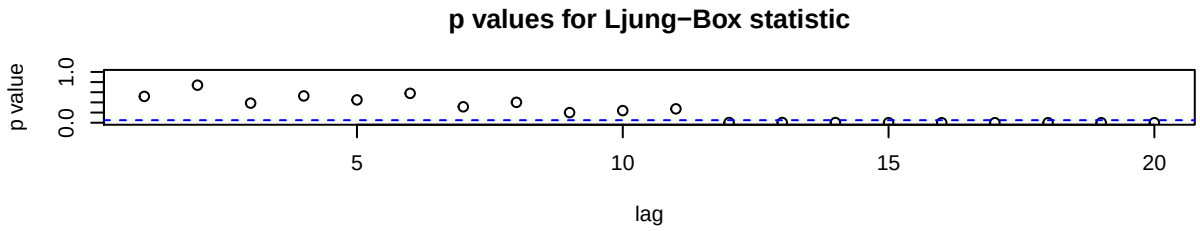
```
tsdiag(NO.arima104, gof.lag = 20)
```



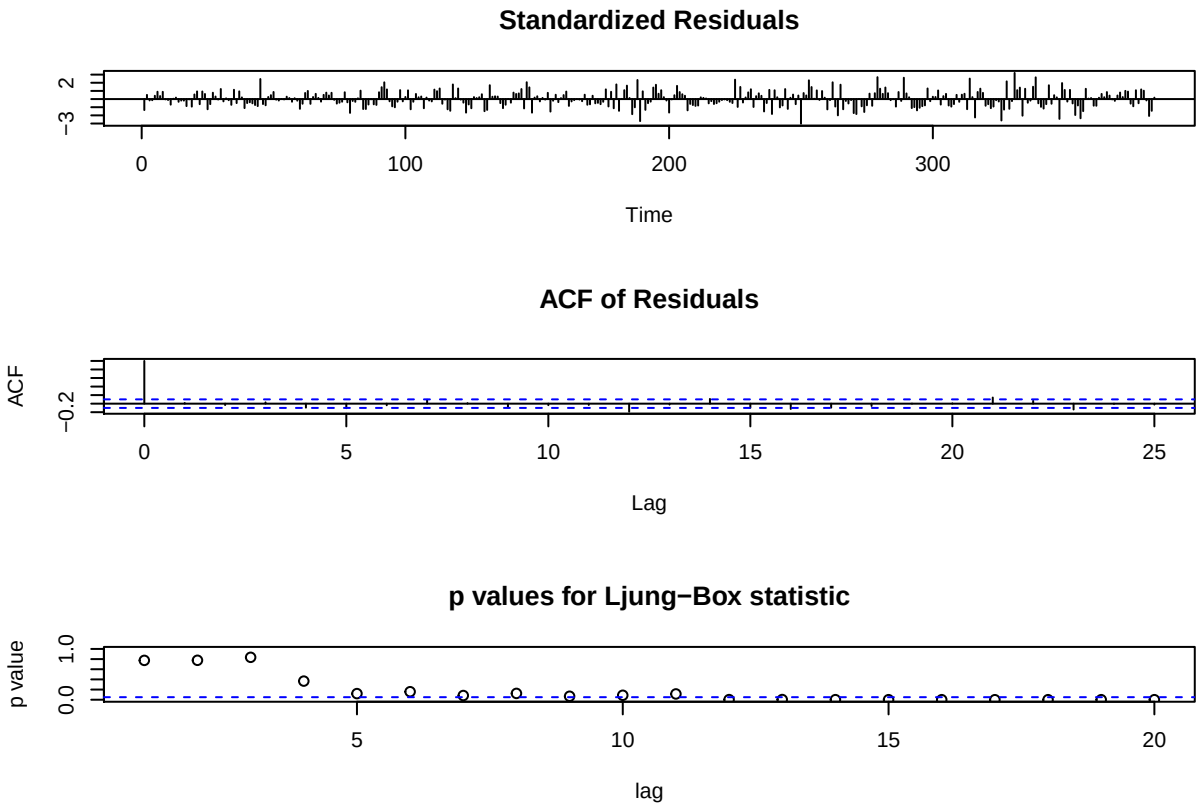

```
tsdiag(NO.arima302, gof.lag = 20)
```








```
tsdiag(NO.residuals.auto, gof.lag = 20)
```



The model that performs the best in diagnostics is the ARIMA(1,0,4) model. It has the same number of points with p-values above the dashed line as the ARIMA(3,0,2) model, but the points are higher. Both models performed significantly better than the auto-generated model, meaning that that model was adequate up to fewer lags. The two models we chose the parameters for are adequate for up to 11 lags, but the auto-generated model is adequate for about 6 lags.

Choose Model for Residuals

```
NO.residuals <- NO.arima104
```

This model has the lowest AIC and also the best diagnostic plot of the models considered.

Final Model

The final model includes a trend and seasonality, using 9 different periods, two of which are the averages of 4 similar periods each. The ARIMA model of the residuals is a 1,0,4 model.

Diagnostics

The residuals of the trend and seasonality linear model appear relatively normal, from the QQ plot. There are no points with high Cook's distances. The residuals vs. fitted and scale-location plots have minor problems; they could have more even spread, but overall they look good and indicate acceptable fit. In the diagnostic plot for the model of the residuals, the p-value for the Ljung-Box stastic is above the dashed line for 11 of

the 20 lags, meaning that the model of the residuals is adequate for up to 11 lags. A future model could work to extend the model adequacy to all 20 lags of the plot.

Multivariate Time Series

Multivariate Time Series

Seasonality and Trends

We used the same linear models for the seasonality and trends that we discovered in our analysis of the univariate time series for CO and NO2. See above for how we modeled seasonal components and trends to produce these linear models.
The ARMA models for the residuals of both time series use d = 0, meaning that both meet the stationarity assumption without taking the first differences.

Autoregressive and Moving Average Terms

In our univariate models, the residuals for CO were modeled with an ARIMA(2,0,1) model and the residuals for NO2 were modeled with an ARIMA(1,0,4) model. There were autoregressive and moving average terms in both cases. Since there were different p and q values used to model the residuals for each time series, it is difficult to use the ACF and PACF to determine values for p and q for the multivariate model. Because of this, we will test models with p from 1 to 2 and q from 0 to 4 to determine the best VARMA model.

Models

We did not include the chunk that actually ran the code to test the VARMA models, so that it would not print the output for every model. The commented code below is the code we ran.

```
# # all residuals together
# allResiduals <- data.frame(CO.lm$residuals, NO.lm$residuals)
#
# # Build VARMA models
#
# # p from 1 to 2, q from 0 to 4
# AICmatrix <- matrix(NA, 2, 5)
# for(p in 1:2) {
#   for(q in 0:4) {
#     varma.model <- VARMACpp(allResiduals, p=p, q=q, include.mean=F)
#     AICmatrix[p,q+1] <- varma.model$aic
#   }
# }
```

Model Comparison

Metrics

```
# get AIC matrix
AICmatrix
```

```
##          [,1]     [,2]     [,3]     [,4]      [,5]
## [1,] 7.103681 7.959438 7.108223 7.130918  8.696485
## [2,] 7.314305 8.141910 7.098223 7.124299 11.406676
```

```
# AICmatrix.sorted <- sort(AICmatrix, decreasing = F, index.return = T)
# AICmatrix.sorted
```

The two models with the lowest AIC values are VARMA(2,2) and VARMA(1,0) with an AIC of 7.098223 and 7.103681, respectively.

Diagnostics

```
# model with lowest AIC
varma.model1 <- VARMACpp(allResiduals, p=2, q=2, include.mean=F)
```

```
## Number of parameters:  16
## initial estimates:  0.8914 -0.0191 -0.4729 0.0071 12.5696 0.3034 -7.8712 0.1275 -0.475 0.0182 0.2559
## Par. lower-bounds:  0.3535 -0.0533 -0.8276 -0.0147 1.817 -0.3801 -14.9618 -0.3082 -1.0328 -0.0168 -0
## Par. upper-bounds:  1.4294 0.0151 -0.1181 0.0288 23.3222 0.9869 -0.7806 0.5632 0.0828 0.0532 0.5781
## Final   Estimates:  0.8914231 -0.01912298 -0.4728765 0.007050431 12.5696 0.3034098 -7.871199 0.12749
##
## Coefficient(s):
##                    Estimate  Std. Error  t value Pr(>|t|)
## CO.lm.residuals    0.891423          NA       NA       NA
## NO.lm.residuals   -0.019123          NA       NA       NA
## CO.lm.residuals   -0.472876          NA       NA       NA
## NO.lm.residuals    0.007050          NA       NA       NA
## CO.lm.residuals   12.569595          NA       NA       NA
## NO.lm.residuals    0.303410          NA       NA       NA
## CO.lm.residuals   -7.871199          NA       NA       NA
## NO.lm.residuals    0.127490          NA       NA       NA
##                   -0.474976          NA       NA       NA
##                    0.018219          NA       NA       NA
##                    0.255891          NA       NA       NA
##                    0.001505          NA       NA       NA
##                  -12.292959          NA       NA       NA
##                    0.057655          NA       NA       NA
##                    1.511878          NA       NA       NA
##                   -0.082400          NA       NA       NA
## ---
## Estimates in matrix form:
## AR coefficient matrix
## AR( 1 )-matrix
##         [,1]    [,2]
## [1,]  0.891 -0.0191
## [2,] 12.570  0.3034
## AR( 2 )-matrix
##         [,1]    [,2]
## [1,] -0.473 0.00705
## [2,] -7.871 0.12749
## MA coefficient matrix
## MA( 1 )-matrix
##         [,1]    [,2]
## [1,]  0.475 -0.0182
## [2,] 12.293 -0.0577
```

```
## MA( 2 )-matrix
##         [,1]     [,2]
## [1,] -0.256 -0.00151
## [2,] -1.512  0.08240
##
## Residuals cov-matrix:
##          [,1]      [,2]
## [1,]  2.38051  32.93033
## [2,] 32.93033 923.11764
## ----
## aic=  7.098223
## bic=  7.262833
```

```
# model with second lowest AIC
varma.model2 <- VARMACpp(allResiduals, p=1, q=0, include.mean=F)
```

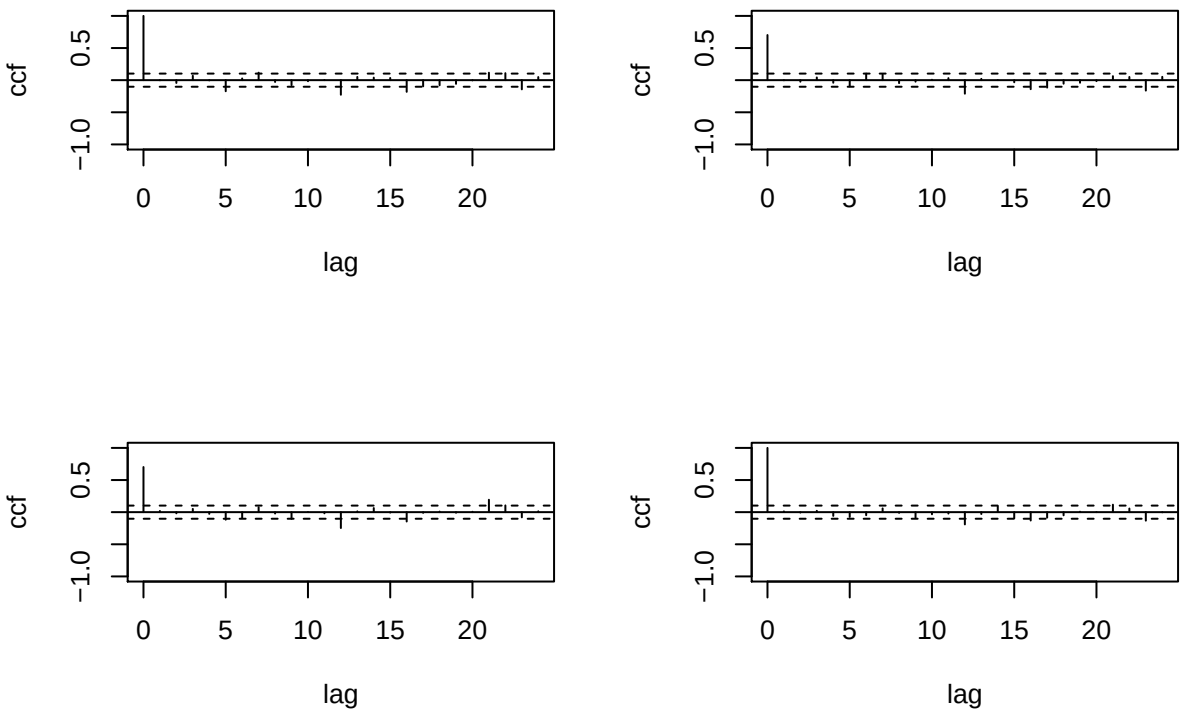
```
## Number of parameters:  4
## initial estimates:  0.4124 -0.0013 0.4313 0.3605
## Par. lower-bounds:  0.2788 -0.0081 -2.2295 0.2259
## Par. upper-bounds:  0.546 0.0055 3.0921 0.4951
## Final   Estimates:  0.3875942 0.005454857 -0.0007040024 0.3754043
##
## Coefficient(s):
##                  Estimate  Std. Error  t value Pr(>|t|)
## CO.lm.residuals  0.387594    0.059522    6.512 7.43e-11 ***
## NO.lm.residuals  0.005455    1.024486    0.005    0.996
## CO.lm.residuals -0.000704    0.003206   -0.220    0.826
## NO.lm.residuals  0.375404    0.059820    6.276 3.48e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ---
## Estimates in matrix form:
## AR coefficient matrix
## AR( 1 )-matrix
##           [,1]    [,2]
## [1,]  0.387594 0.00545
## [2,] -0.000704 0.37540
##
## Residuals cov-matrix:
##           [,1]     [,2]
## [1,]  2.464721  33.7206
## [2,] 33.720604 944.7054
## ----
## aic=  7.103681
## bic=  7.144833
```

```
# diagnostics for model1
MTSdiag(varma.model1)
```

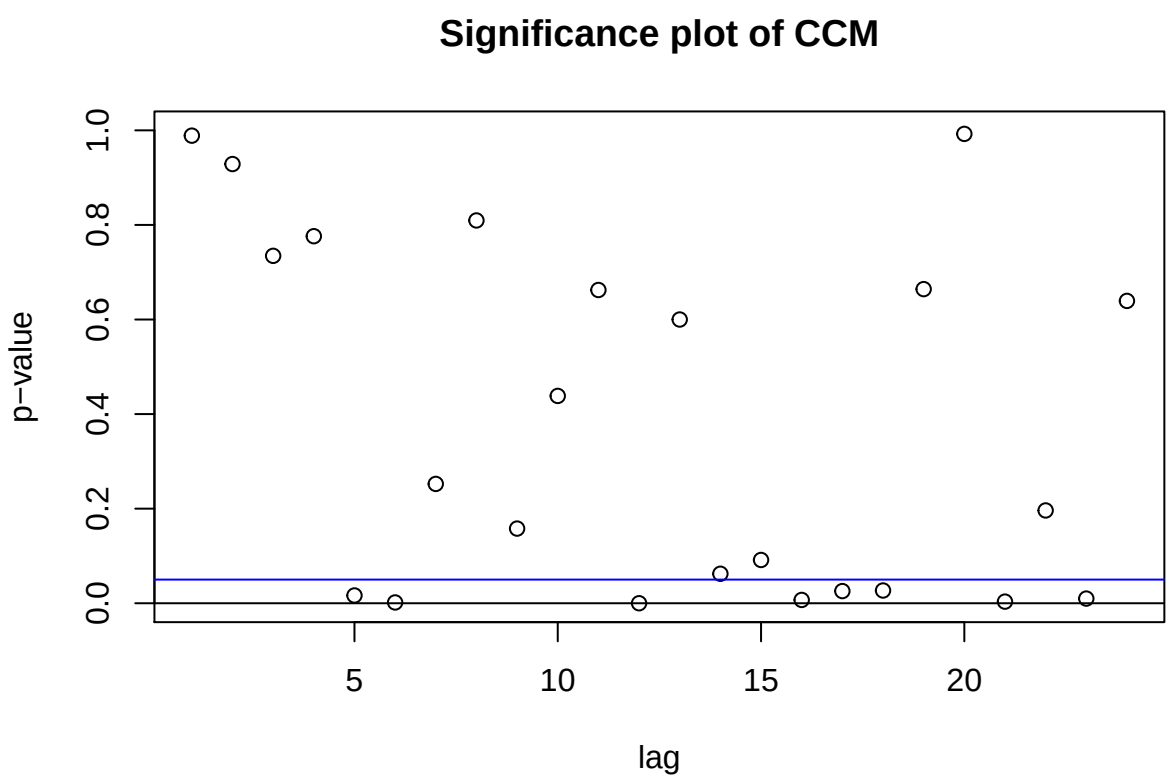
```
## [1] "Covariance matrix:"
##                 CO.lm.residuals NO.lm.residuals
## CO.lm.residuals            2.39              33
## NO.lm.residuals           33.02             926
```

```
## CCM at lag:  0
##       [,1]  [,2]
## [1,] 1.000 0.702
## [2,] 0.702 1.000
## Simplified matrix:
## CCM at lag:  1
## . .
## . .
## CCM at lag:  2
## . .
## . .
## CCM at lag:  3
## . .
## . .
## CCM at lag:  4
## . .
## . .
## CCM at lag:  5
## - .
## - .
## CCM at lag:  6
## . .
## . .
## CCM at lag:  7
## + .
## . .
## CCM at lag:  8
## . .
## . .
## CCM at lag:  9
## . .
## - .
## CCM at lag:  10
## . .
## . .
## CCM at lag:  11
## . .
## . .
## CCM at lag:  12
## - -
## - -
## CCM at lag:  13
## . .
## . .
## CCM at lag:  14
## . .
## . .
## CCM at lag:  15
## . .
## . .
## CCM at lag:  16
## - -
## - -
## CCM at lag:  17
```

```
## . -
## . .
## CCM at lag:  18
## . .
## . .
## CCM at lag:  19
## . .
## . .
## CCM at lag:  20
## . .
## . .
## CCM at lag:  21
## + .
## + +
## CCM at lag:  22
## + .
## + .
## CCM at lag:  23
## - -
## . -
## CCM at lag:  24
## . .
## . .
```

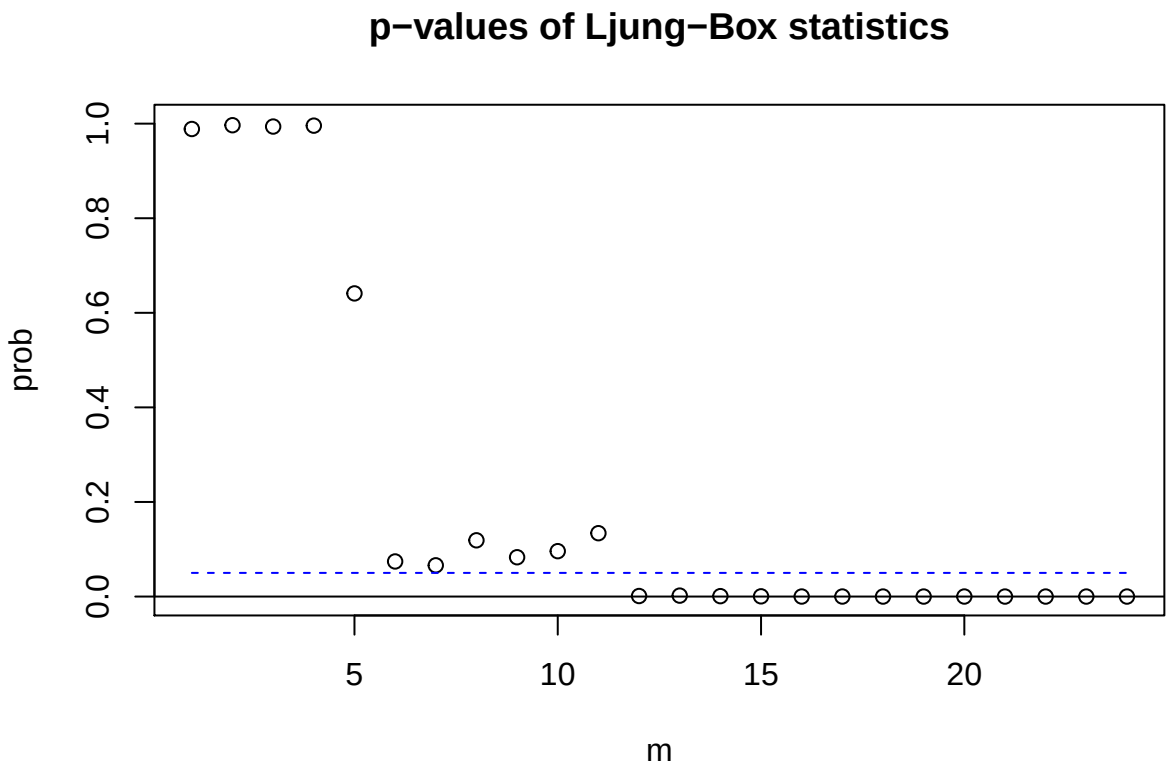



```
## Hit Enter for p-value plot of individual ccm:
```

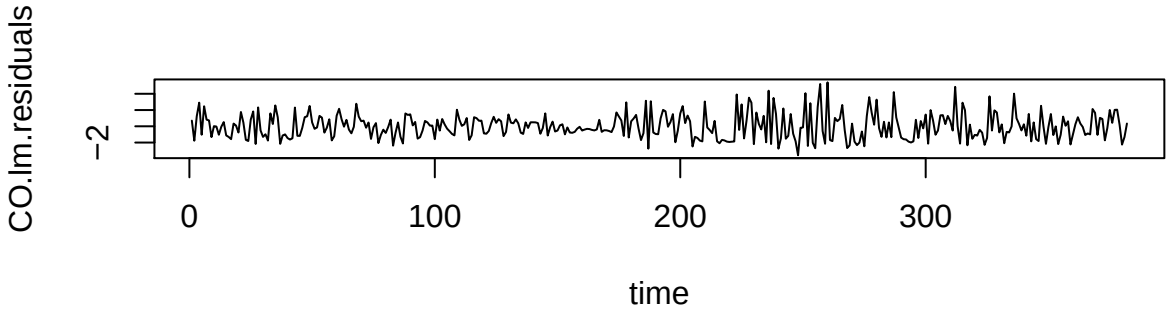


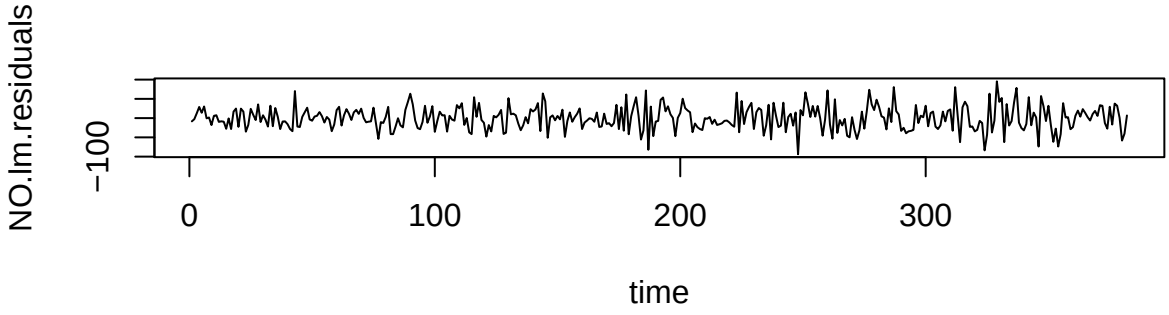

```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##           m       Q(m)     df    p-value
##  [1,]   1.000     0.317   4.000     0.99
##  [2,]   2.000     1.192   8.000     1.00
##  [3,]   3.000     3.209  12.000     0.99
##  [4,]   4.000     4.997  16.000     1.00
##  [5,]   5.000    17.183  20.000     0.64
##  [6,]   6.000    34.623  24.000     0.07
##  [7,]   7.000    40.010  28.000     0.07
##  [8,]   8.000    41.613  32.000     0.12
##  [9,]   9.000    48.265  36.000     0.08
## [10,]  10.000    52.054  40.000     0.10
## [11,]  11.000    54.469  44.000     0.13
## [12,]  12.000    83.534  48.000     0.00
## [13,]  13.000    86.300  52.000     0.00
## [14,]  14.000    95.297  56.000     0.00
## [15,]  15.000   103.338  60.000     0.00
## [16,]  16.000   117.516  64.000     0.00
## [17,]  17.000   128.654  68.000     0.00
## [18,]  18.000   139.678  72.000     0.00
## [19,]  19.000   142.081  76.000     0.00
## [20,]  20.000   142.337  80.000     0.00
## [21,]  21.000   158.142  84.000     0.00
## [22,]  22.000   164.210  88.000     0.00
```

```
## [23,]  23.000   177.591  92.000     0.00
## [24,]  24.000   180.135  96.000     0.00
```




```
## Hit Enter to obtain residual plots:
```



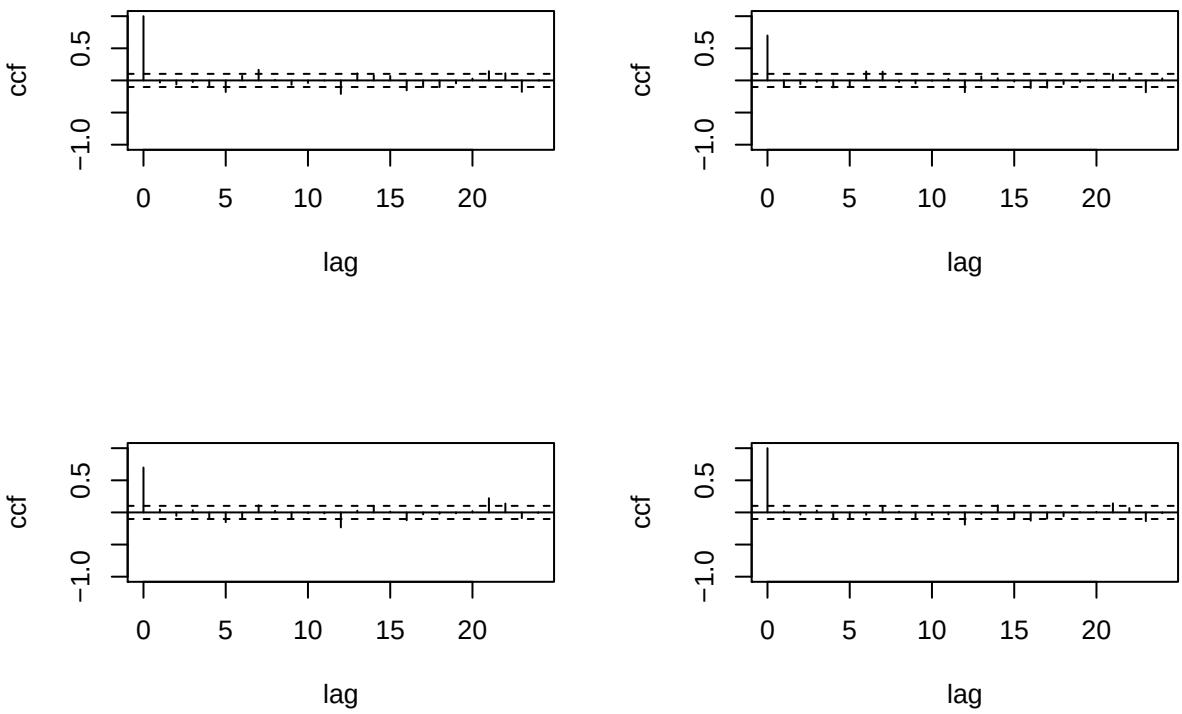



```
# diagnostics for model2
MTSdiag(varma.model2)
```

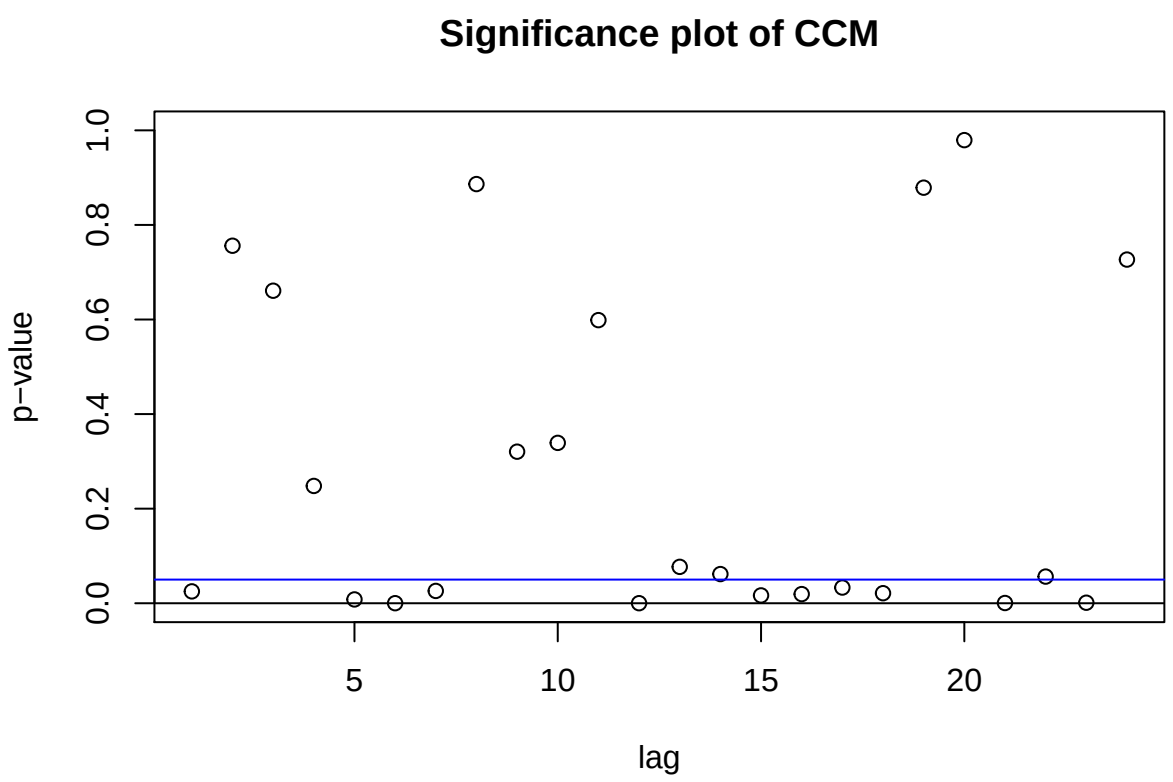
```
## [1] "Covariance matrix:"
##                  CO.lm.residuals NO.lm.residuals
## CO.lm.residuals            2.47            33.8
## NO.lm.residuals           33.81           947.2
## CCM at lag:  0
##       [,1]  [,2]
## [1,] 1.000 0.699
## [2,] 0.699 1.000
## Simplified matrix:
## CCM at lag:  1
## . -
## . .
## CCM at lag:  2
## . .
## . .
## CCM at lag:  3
## . .
## . .
## CCM at lag:  4
## . -
## . .
## CCM at lag:  5
## - -
```

```
## - -
## CCM at lag:  6
## . +
## . .
## CCM at lag:  7
## + +
## + .
## CCM at lag:  8
## . .
## . .
## CCM at lag:  9
## . .
## . .
## CCM at lag:  10
## . .
## . .
## CCM at lag:  11
## . .
## . .
## CCM at lag:  12
## - -
## - -
## CCM at lag:  13
## + .
## . .
## CCM at lag:  14
## . .
## + +
## CCM at lag:  15
## . .
## . .
## CCM at lag:  16
## - -
## - -
## CCM at lag:  17
## - -
## . .
## CCM at lag:  18
## - .
## . .
## CCM at lag:  19
## . .
## . .
## CCM at lag:  20
## . .
## . .
## CCM at lag:  21
## + .
## + +
## CCM at lag:  22
## + .
## + .
## CCM at lag:  23
## - -
```

```
## . -
## CCM at lag:  24
## . .
## . .
```

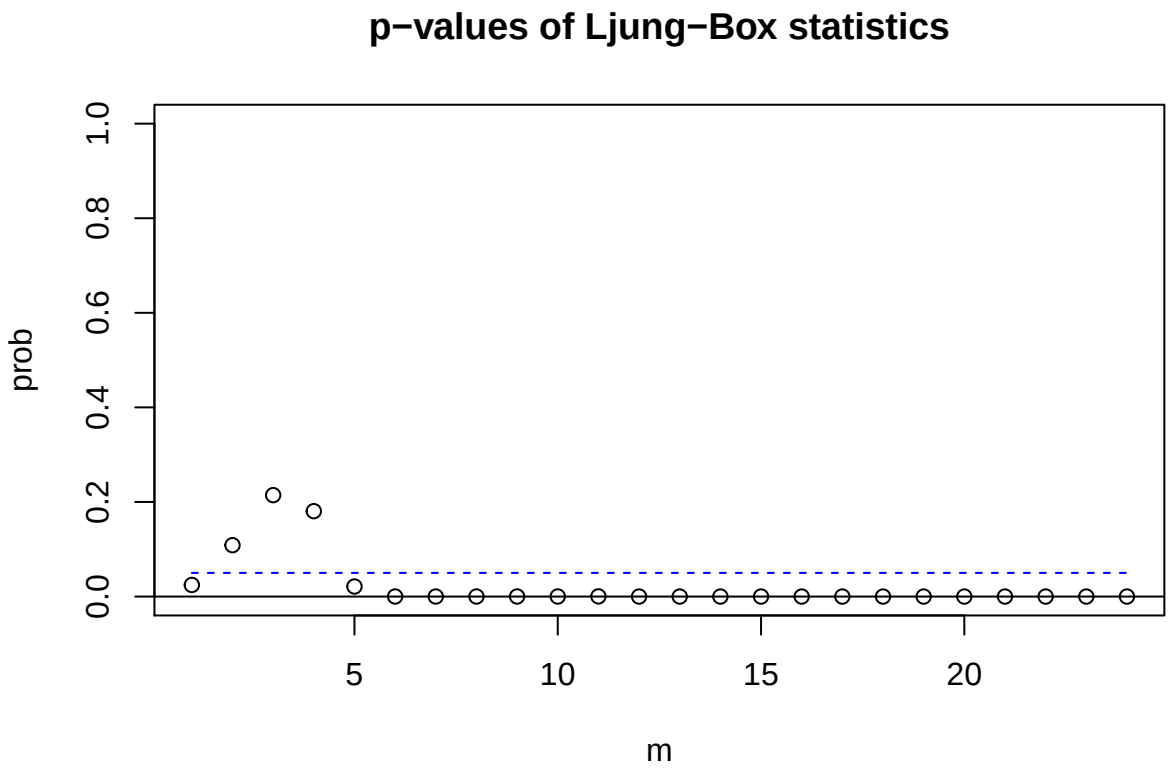



```
## Hit Enter for p-value plot of individual ccm:
```

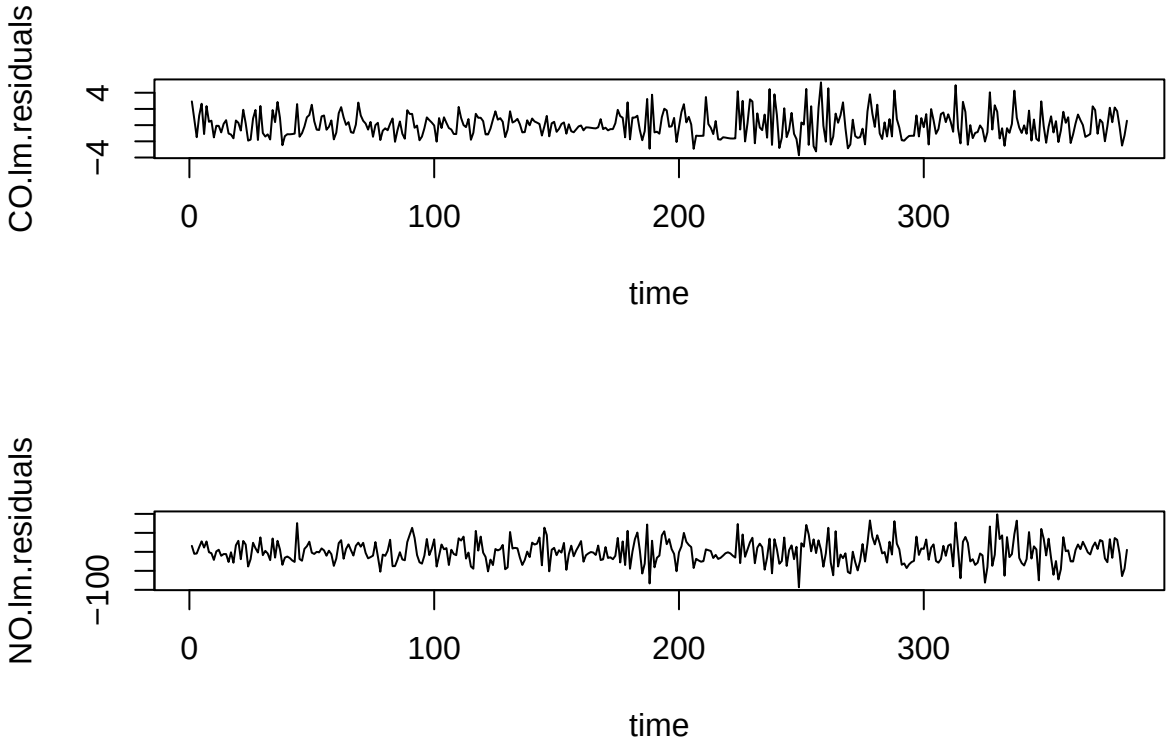



```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##          m       Q(m)     df    p-value
##  [1,]   1.0      11.2     4.0     0.02
##  [2,]   2.0      13.1     8.0     0.11
##  [3,]   3.0      15.5    12.0     0.21
##  [4,]   4.0      20.9    16.0     0.18
##  [5,]   5.0      34.8    20.0     0.02
##  [6,]   6.0      58.7    24.0     0.00
##  [7,]   7.0      69.8    28.0     0.00
##  [8,]   8.0      71.0    32.0     0.00
##  [9,]   9.0      75.7    36.0     0.00
## [10,]  10.0      80.2    40.0     0.00
## [11,]  11.0      83.0    44.0     0.00
## [12,]  12.0     107.3    48.0     0.00
## [13,]  13.0     115.8    52.0     0.00
## [14,]  14.0     124.8    56.0     0.00
## [15,]  15.0     136.9    60.0     0.00
## [16,]  16.0     148.8    64.0     0.00
## [17,]  17.0     159.3    68.0     0.00
## [18,]  18.0     170.9    72.0     0.00
## [19,]  19.0     172.1    76.0     0.00
## [20,]  20.0     172.5    80.0     0.00
## [21,]  21.0     192.7    84.0     0.00
## [22,]  22.0     202.0    88.0     0.00
```

```
## [23,]  23.0     220.3    92.0     0.00
## [24,]  24.0     222.4    96.0     0.00
```




```
## Hit Enter to obtain residual plots:
```



The VARMA(2,2) model performs best in the diagnostics. The model is adequate for 11 lags, as the p-values for the Ljung-Box statistic are above the dashed line for these points. The second model we tested had three Ljung-Box statistics at lags 2, 3, and 4 with a p-value above the dashed line.

Final Model

We did not include the chunk that actually ran the code to create the final model, so that it would not print the output. The commented code below is the code we ran.

```
# varma.model <- VARMACpp(allResiduals, p=2, q=2, include.mean=F)
```

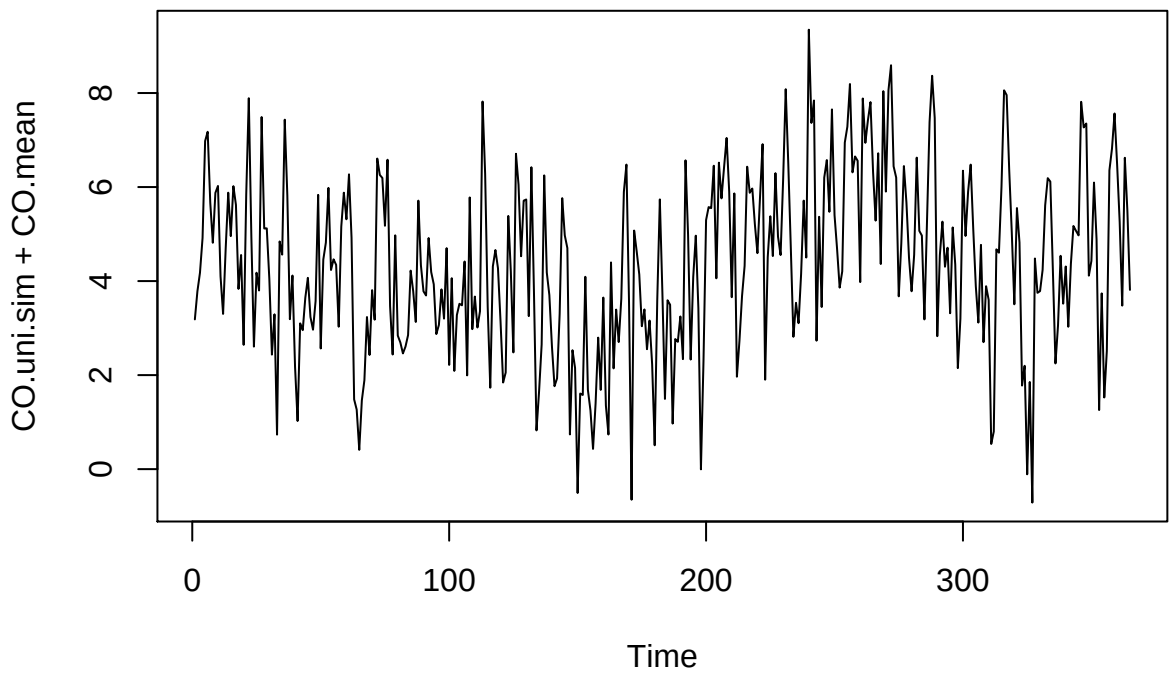
The final model was chosen as the VARMA(2,2) model, because it has the lowest AIC and performed the best in the diagnostic plot.

Diagnostics

The model is adequate for up to 11 lags, based on the diagnostic plot. Future work could attempt to produce a model adequate for more lags, but we feel that this is acceptable performance from the model we have chosen.

Simulation from Univariate CO Model

```r
# CO ARIMA model is 2,0,1

# simulate residuals
set.seed(1)
CO.uni.sim <- arima.sim(n = 365, list(ar = c(CO.residuals$coef[1],
                                             CO.residuals$coef[2]),
                                      ma = c(CO.residuals$coef[3])),
                                      sd = sqrt(CO.residuals$sigma2))

# the next time variable- simulate the next 365 days
next.time.time <- c(1:(365))
next.time <- data.frame(time.CO = next.time.time)

CO.mean <- predict(CO.lm, newdata = next.time)
plot(CO.uni.sim + CO.mean)
```

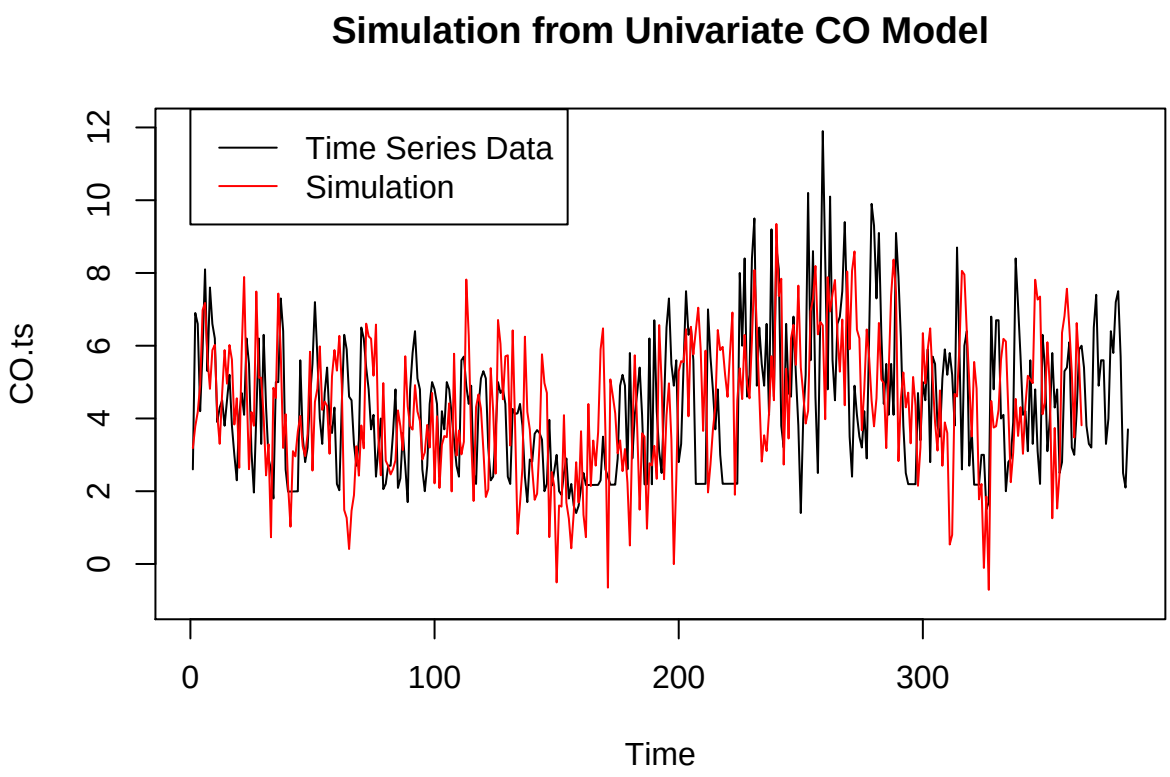



```r
# make time series
CO.uni.sim.ts <- ts(CO.uni.sim + CO.mean)
```

This is the plot of the simulation.

Visualization

```
# plot simulated values with observations
plot(CO.ts, main = "Simulation from Univariate CO Model", col = "black", ylim = c(-1,12))
lines(CO.uni.sim.ts, col = "red")
legend(0, 12.5, legend = c("Time Series Data", "Simulation"), col = c("black", "red"), lwd = 1)
```



The simulation values appear to be similar to the time series data, though they are slightly larger than the time series before t = 200.

Trend

```
# linear model for simulation
CO.sim.lm <- lm(CO.uni.sim.ts ~ next.time.time)
summary(CO.sim.lm)
```

```
##
## Call:
## lm(formula = CO.uni.sim.ts ~ next.time.time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.513 -1.171   0.006   1.368   4.819
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.7706155  0.1933586  19.501  < 2e-16 ***
## next.time.time 0.0031546  0.0009157   3.445 0.000638 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.843 on 363 degrees of freedom
## Multiple R-squared:  0.03166,    Adjusted R-squared:  0.02899
## F-statistic: 11.87 on 1 and 363 DF,  p-value: 0.0006376
```

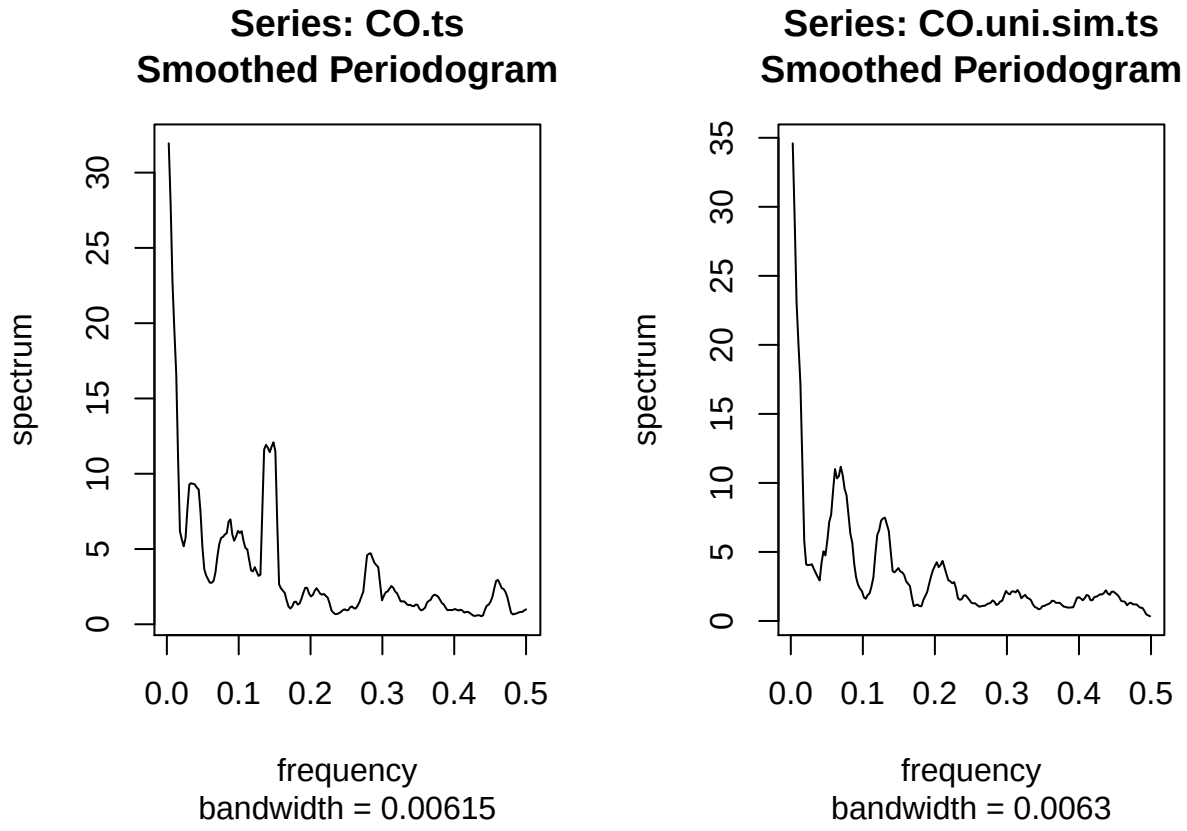
```
summary(CO.lm.trend)
```

```
##
## Call:
## lm(formula = CO.ts ~ time.CO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2485 -1.6980 -0.0525  1.0863  7.3442
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.810929   0.192140  19.834  < 2e-16 ***
## time.CO     0.002876   0.000865   3.325  0.00097 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.879 on 382 degrees of freedom
## Multiple R-squared:  0.02813,    Adjusted R-squared:  0.02558
## F-statistic: 11.06 on 1 and 382 DF,  p-value: 0.0009695
```

We made a linear model for the simulation with time as a predictor to determine if there was a trend to the data. The model was significant, meaning that there is a trend to the simulation. The estimated coefficient value was 0.0032 for the simulation and 0.0029 for the original time series. These coefficients are very close, so the simulation was able to accurately reproduce the trend of the original time series.

Seasonality

```
# compare periodogram of observations and periodogram of simulation
par(mfrow = c(1,2))
pg.CO <- spec.pgram(CO.ts,spans=9,demean=T,log='no')
pg.CO.uni.sim <- spec.pgram(CO.uni.sim.ts, spans = 9, demean = T, log = 'no')
```




```
par(mfrow = c(1,1))
```

The periodograms of the original observations and simulation look similar. They have similar peak locations and magnitudes, especially at the lower end of the frequencies. The simulation was able to closely reproduce the seasonality of the time series.

Mean and Variance

```
# mean of observations
mean(CO.ts)
```

```
## [1] 4.364574
```

```
# mean of simulation
mean(CO.uni.sim.ts)
```

```
## [1] 4.347904
```

```
# variance of observations
var(CO.ts)
```

```
## [1] 3.622965
```

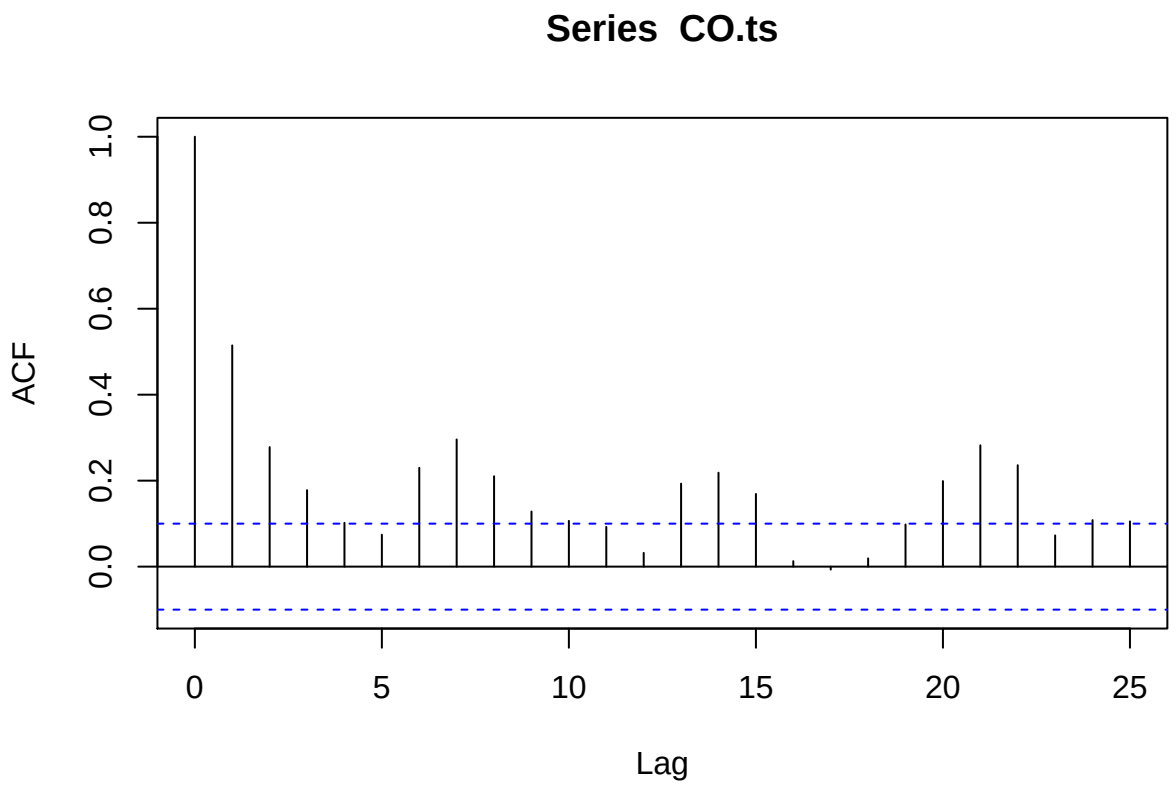
```r
# variance of simulation
var(CO.uni.sim.ts)
```

```
## [1] 3.499064
```

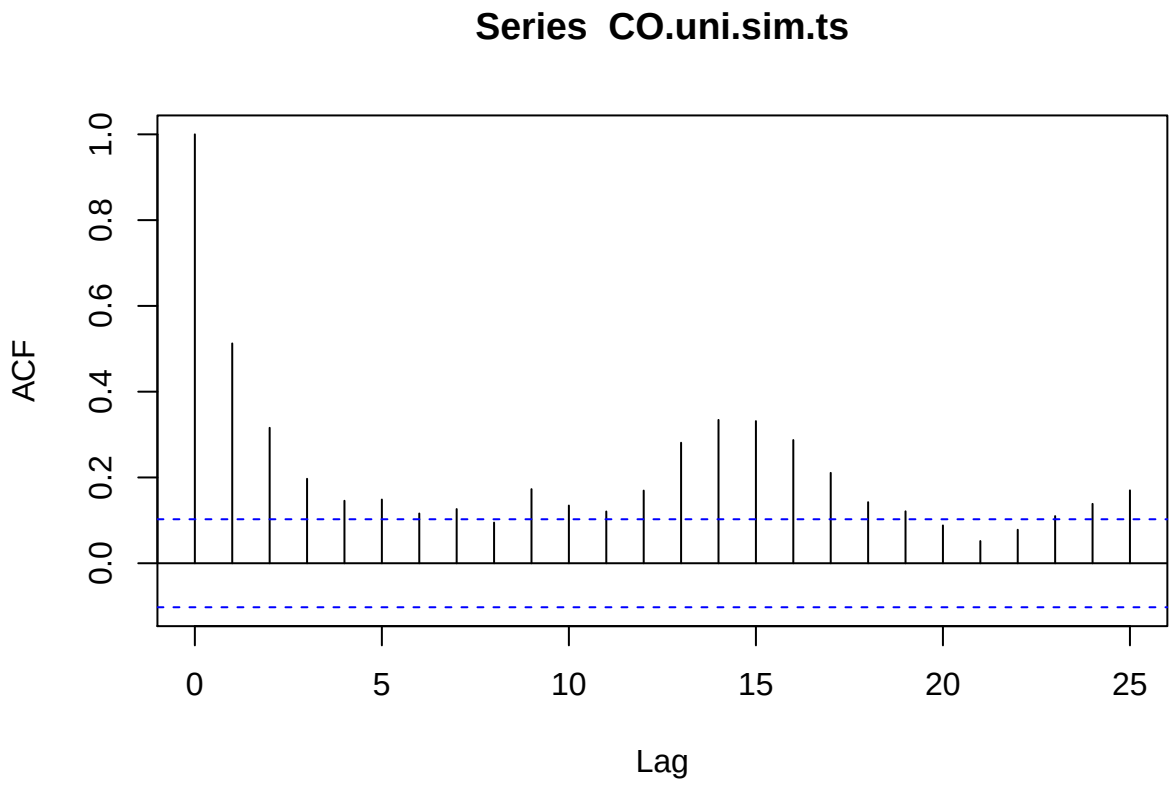
The mean of the original time series is 4.36, and the mean of the simulation is 4.34. The means are very close together, which was apparent from the plot of both the time series and simulation together.
The variance of the original time series is 3.62, and the variance of the simulation is 3.499. The variances are also similar.

Auto-Correlation

```r
# ACF of observations
acf(CO.ts)
```

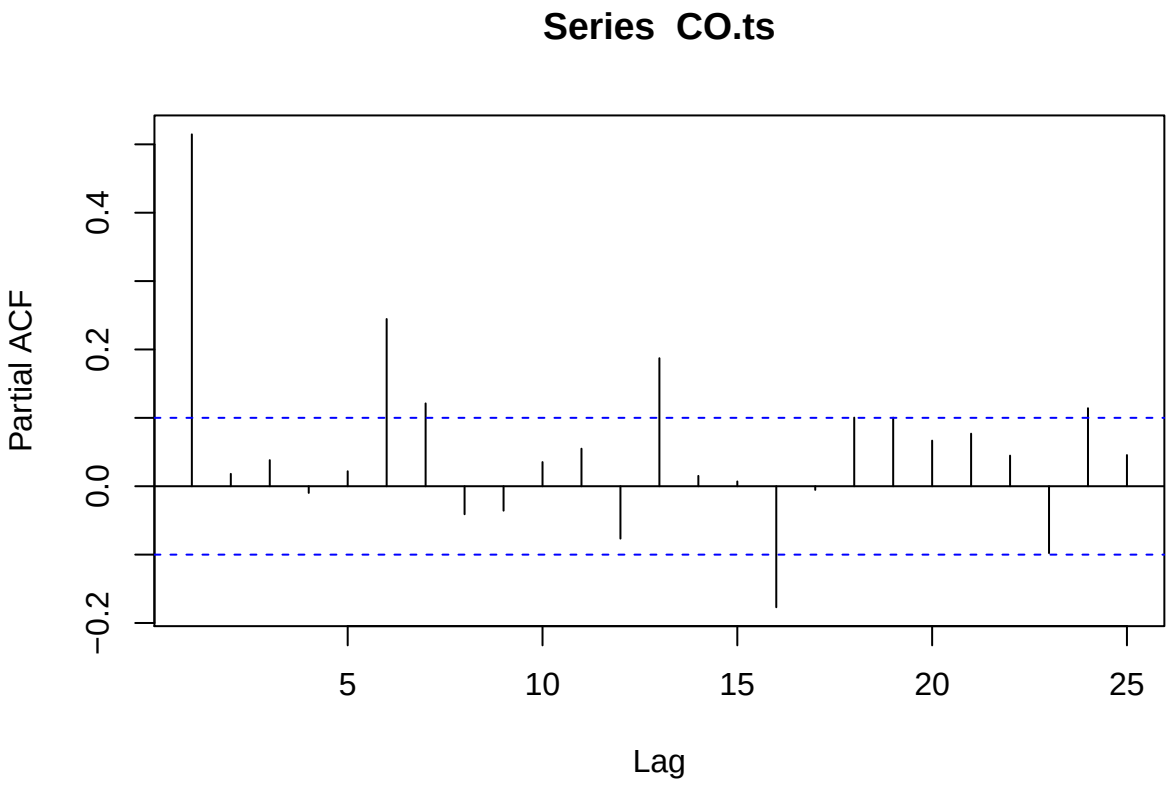



```r
# ACF of simulation
acf(CO.uni.sim.ts)
```

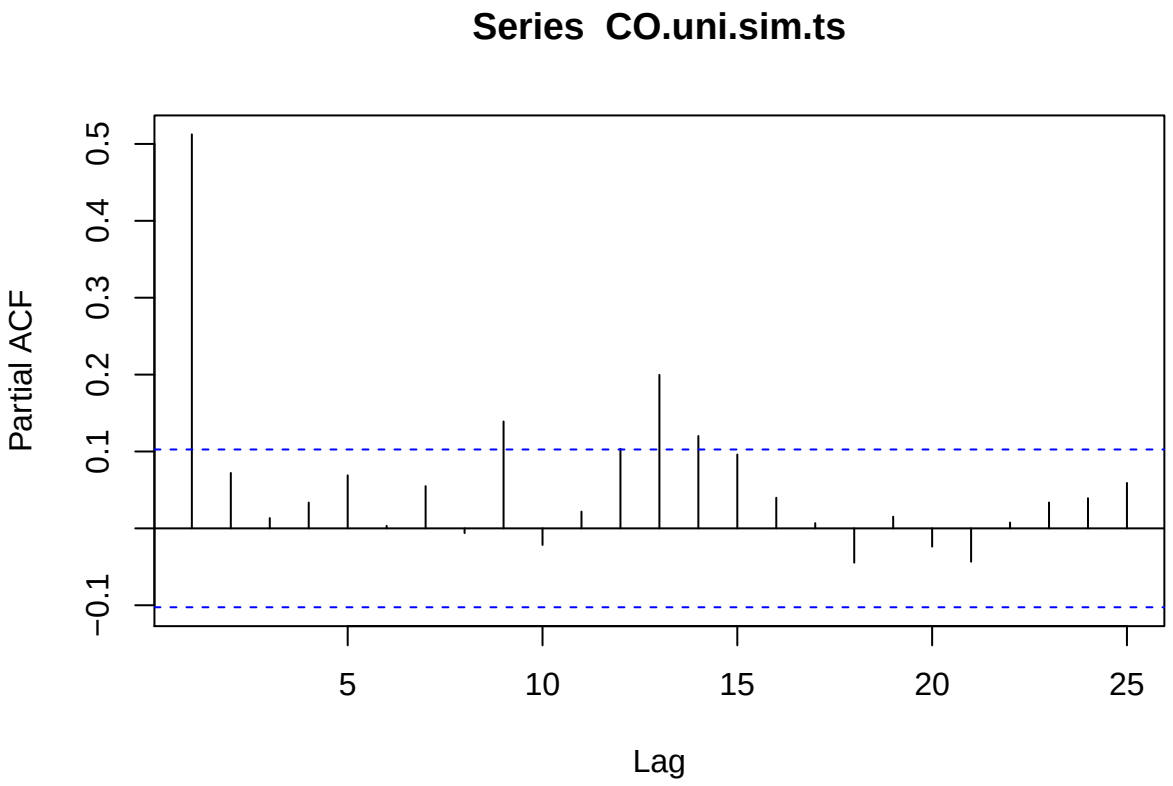


The simulation got close to reproducing the ACF of the time series. They both cutoff and become significant again and show somewhat sinusoidal behavior.

```
# PACF of observations
pacf(CO.ts)
```




```
# PACF of simulation
pacf(CO.uni.sim.ts)
```

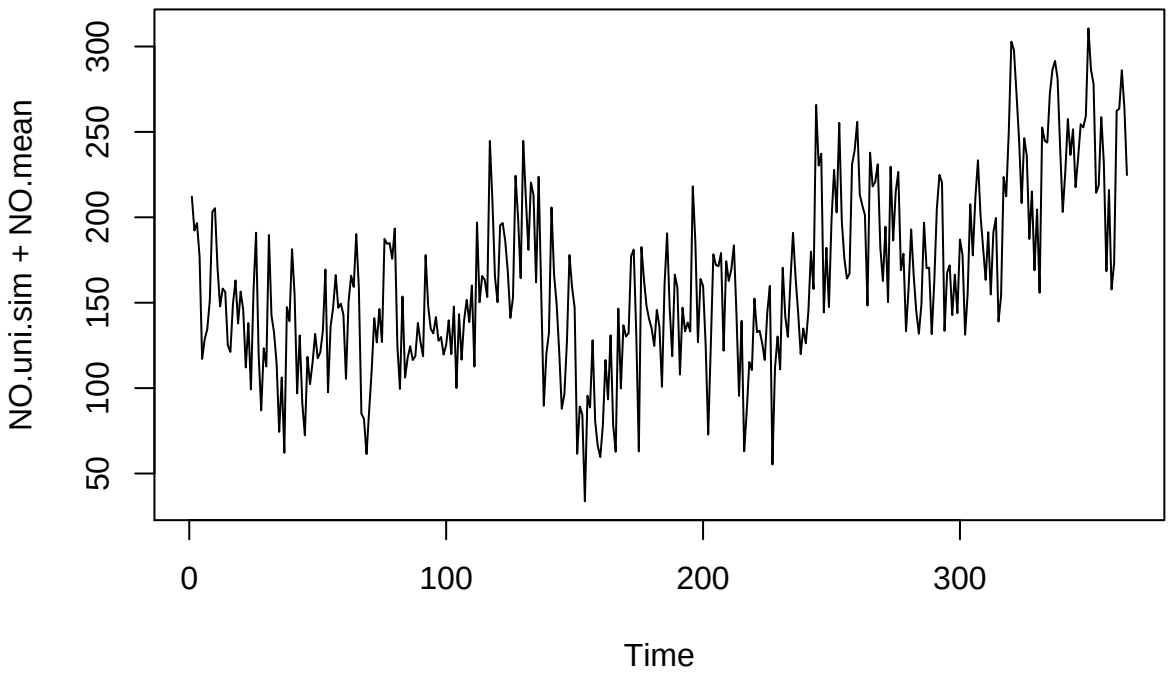


The ACFs look more similar, though the PACF of the time series and simulation are close as well. Both cutoff and become significant again, and both are significant for the first lag but not for the second.

From comparing the ACF and PACF of the time series and simulation, the simulation was able to reproduce the autocorrelation of the time series.

Simulation from Univariate NO2 Model

```
# NO2 ARIMA model is 1,0,4

# simulate residuals
set.seed(1)
NO.uni.sim <- arima.sim(n = 365, list(ar = c(NO.residuals$coef[1]),
                                      ma = c(NO.residuals$coef[2],
                                             NO.residuals$coef[3],
                                             NO.residuals$coef[4],
                                             NO.residuals$coef[5])),
                                      sd = sqrt(NO.residuals$sigma2))

# the next time variable- simulate the next 365 days
next.time.NO <- data.frame(time.NO = next.time.time)

NO.mean <- predict(NO.lm, newdata = next.time.NO)
plot(NO.uni.sim + NO.mean)
```

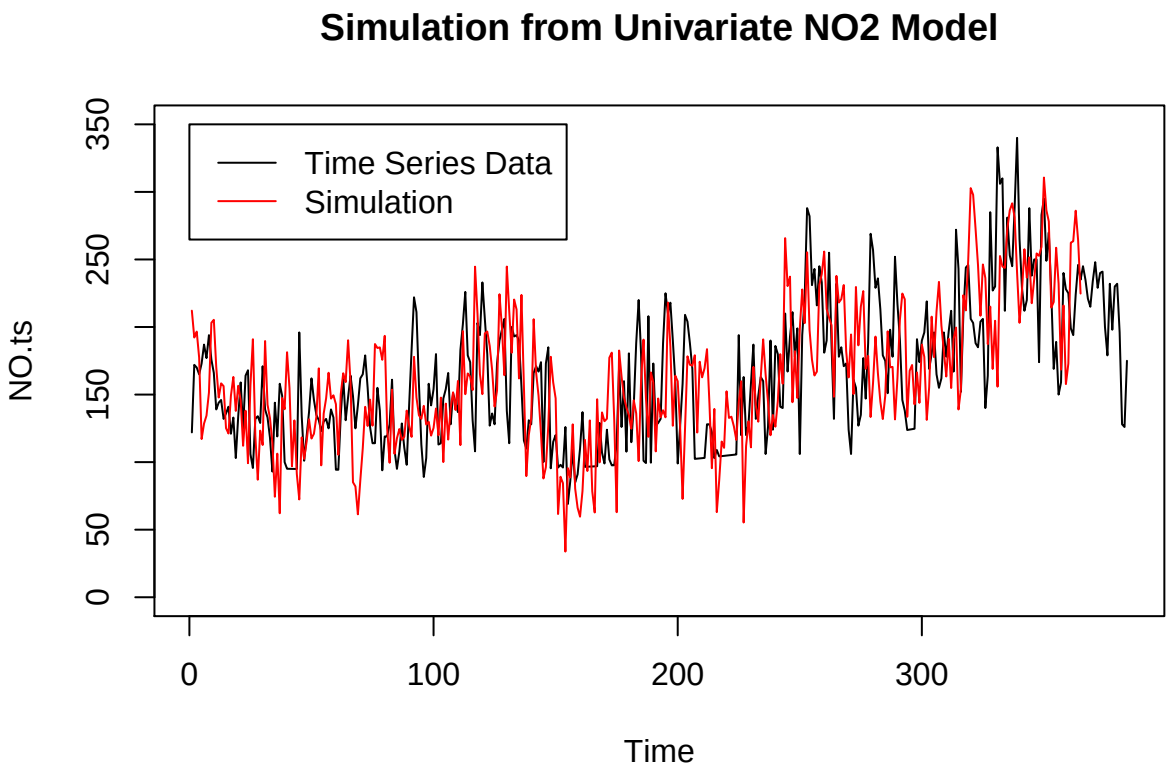



```
# make time series
NO.uni.sim.ts <- ts(NO.uni.sim + NO.mean)
```

This is the plot of the simulation.

Visualization

```
# plot simulated values with observations
plot(NO.ts, main = "Simulation from Univariate NO2 Model", col = "black", ylim = c(0,350))
lines(NO.uni.sim.ts, col = "red")
legend(0, 350, legend = c("Time Series Data", "Simulation"), col = c("black", "red"), lwd = 1)
```



The simulation values appear similar to the time series values, especially in the trend of the data.

Trend

```
# linear model for simulation
NO.sim.lm <- lm(NO.uni.sim.ts ~ next.time.time)
summary(NO.sim.lm)
```

```
##
## Call:
## lm(formula = NO.uni.sim.ts ~ next.time.time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.395  -29.517    0.047   29.131  104.051
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    114.75310    4.52157   25.38   <2e-16 ***
## next.time.time   0.26246    0.02141   12.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.1 on 363 degrees of freedom
## Multiple R-squared:  0.2927, Adjusted R-squared:  0.2908
```

```
## F-statistic: 150.2 on 1 and 363 DF,  p-value: < 2.2e-16
```

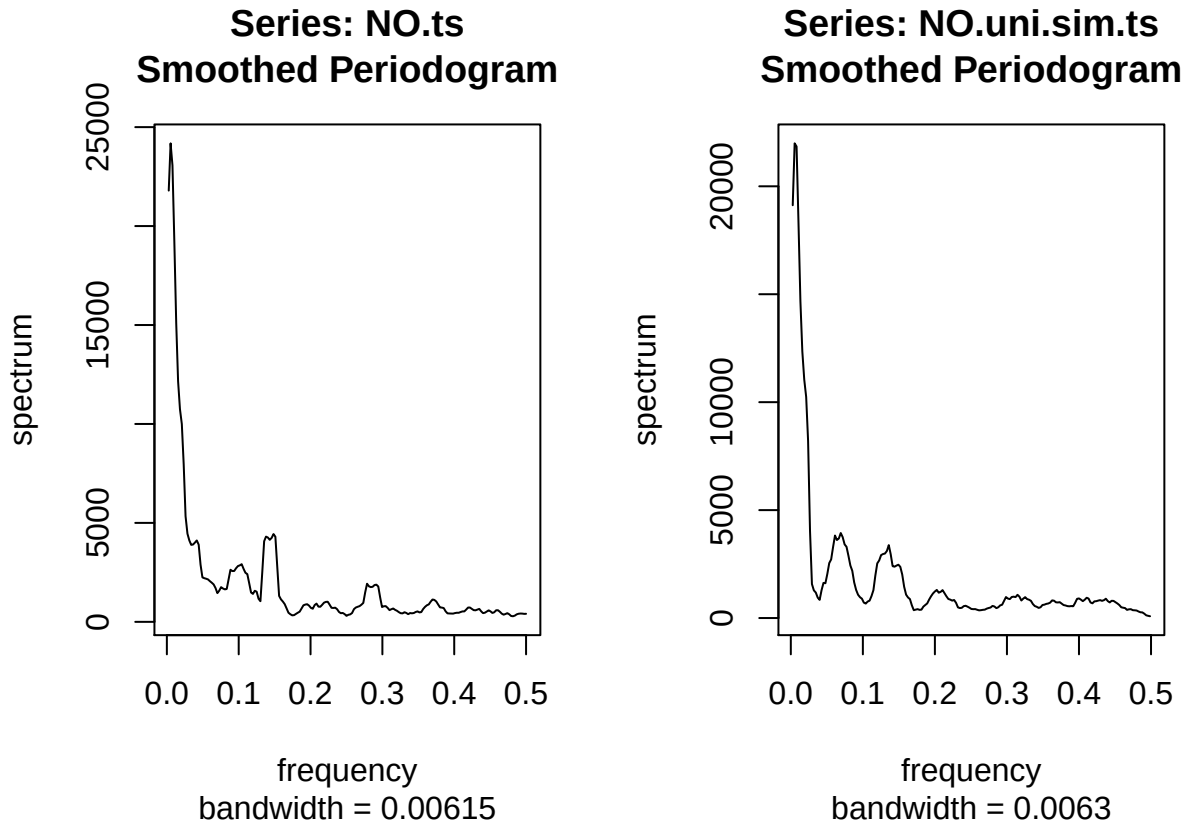
```
summary(NO.lm.trend)
```

```
##
## Call:
## lm(formula = NO.ts ~ time.NO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -87.389 -34.365   2.159  27.847 137.895
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115.16473    4.40111   26.17   <2e-16 ***
## time.NO       0.25646    0.01981   12.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.04 on 382 degrees of freedom
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3031
## F-statistic: 167.6 on 1 and 382 DF,  p-value: < 2.2e-16
```

We made a linear model for the simulation with time as a predictor to determine if there was a trend to the data. The model was significant, meaning that there is a trend to the simulation. The estimated coefficient value was 0.262 for the simulation and 0.256 for the original time series. These coefficients are very close, so the simulation was able to accurately reproduce the trend of the original time series.

Seasonality

```
# compare periodogram of observations and periodogram of simulation
par(mfrow = c(1,2))
pg.NO <- spec.pgram(NO.ts,spans=9, demean=T, log='no')
pg.NO.uni.sim <- spec.pgram(NO.uni.sim.ts, spans = 9, demean = T, log = 'no')
```




```
par(mfrow = c(1,1))
```

The periodograms of the original observations and simulation look similar, but the periodogram for the simulation appears to have different peaks in the frequency range of 0.05 to 0.2. The simulation was able to somewhat closely reproduce the seasonality of the time series.

Mean and Variance

```
# mean of observations
mean(NO.ts)
```

```
## [1] 164.5335
```

```
# mean of simulation
mean(NO.uni.sim.ts)
```

```
## [1] 162.7834
```

```
# variance of observations
var(NO.ts)
```

```
## [1] 2657.722
```

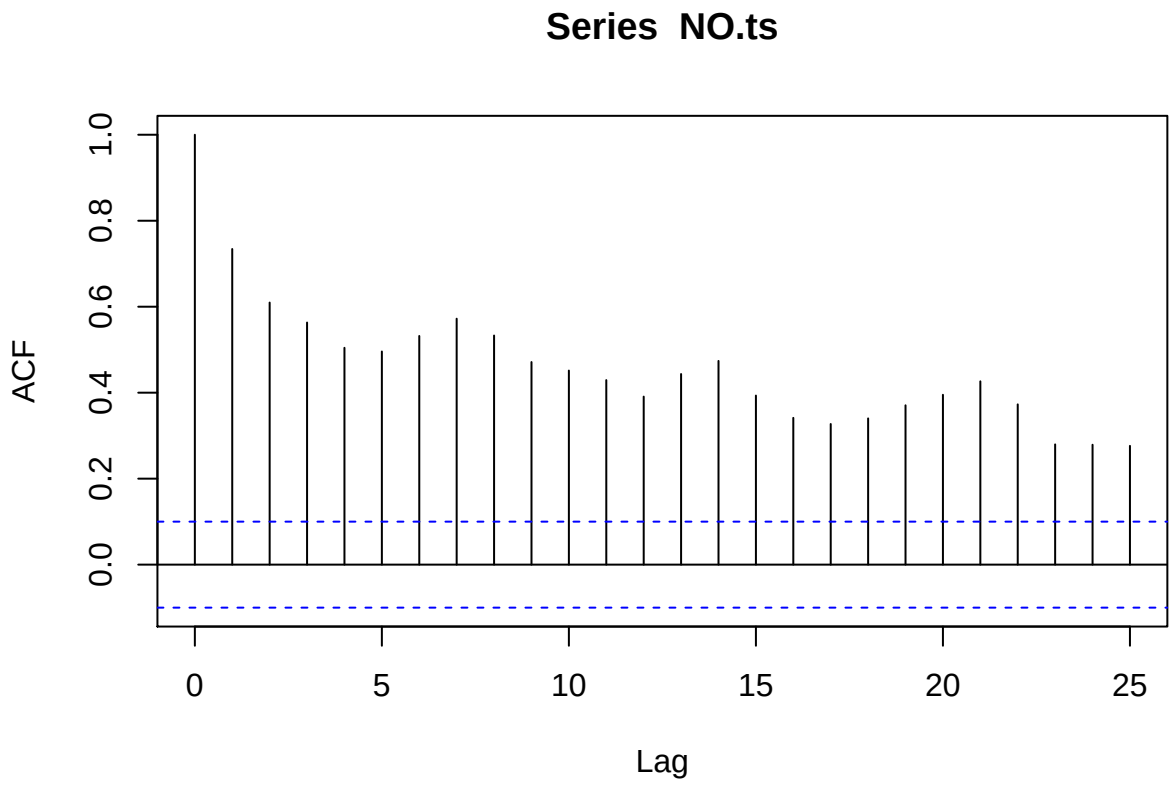
```r
# variance of simulation
var(NO.uni.sim.ts)
```

```
## [1] 2619.679
```

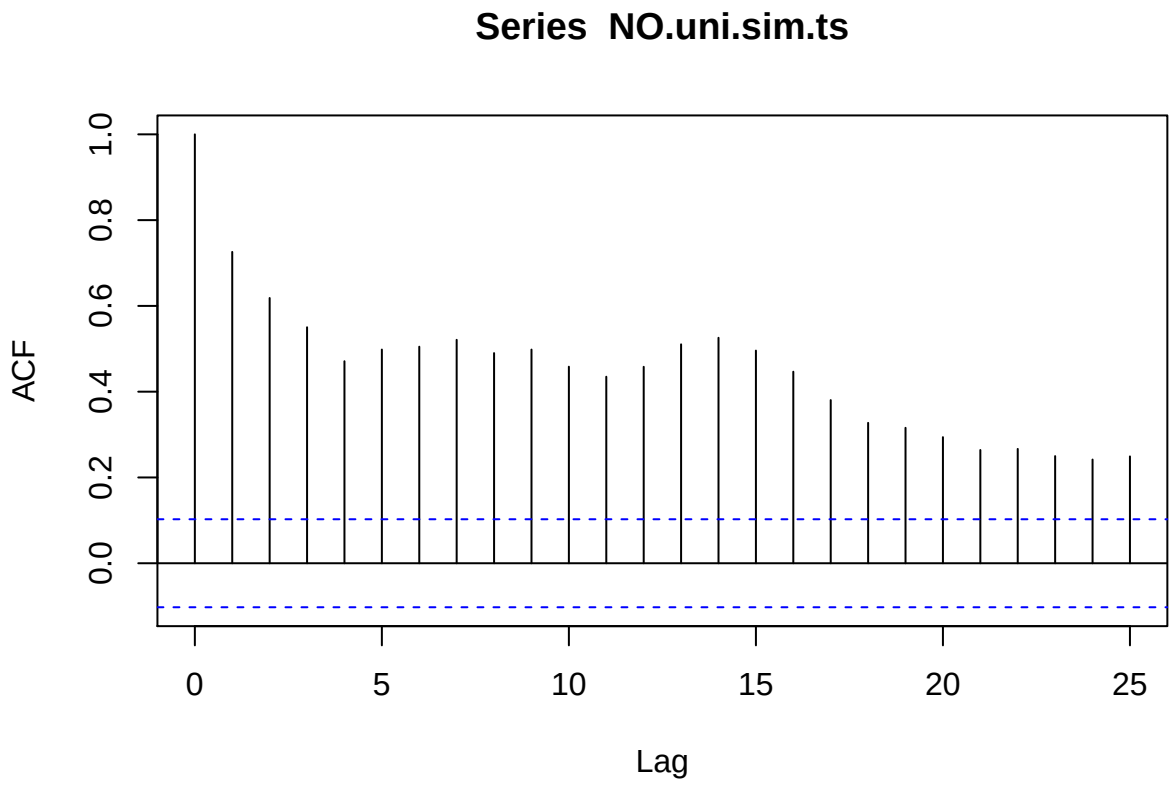
The mean of the original time series is 164.53, and the mean of the simulation is 162.78. The means are very close together, which was apparent from the plot of both the time series and simulation together.
The variance of the original time series is 2657.722, and the variance of the simulation is 2619.679. The variances are also similar, with the simulation being a little lower.

Auto-Correlation

```r
# ACF of observations
acf(NO.ts)
```

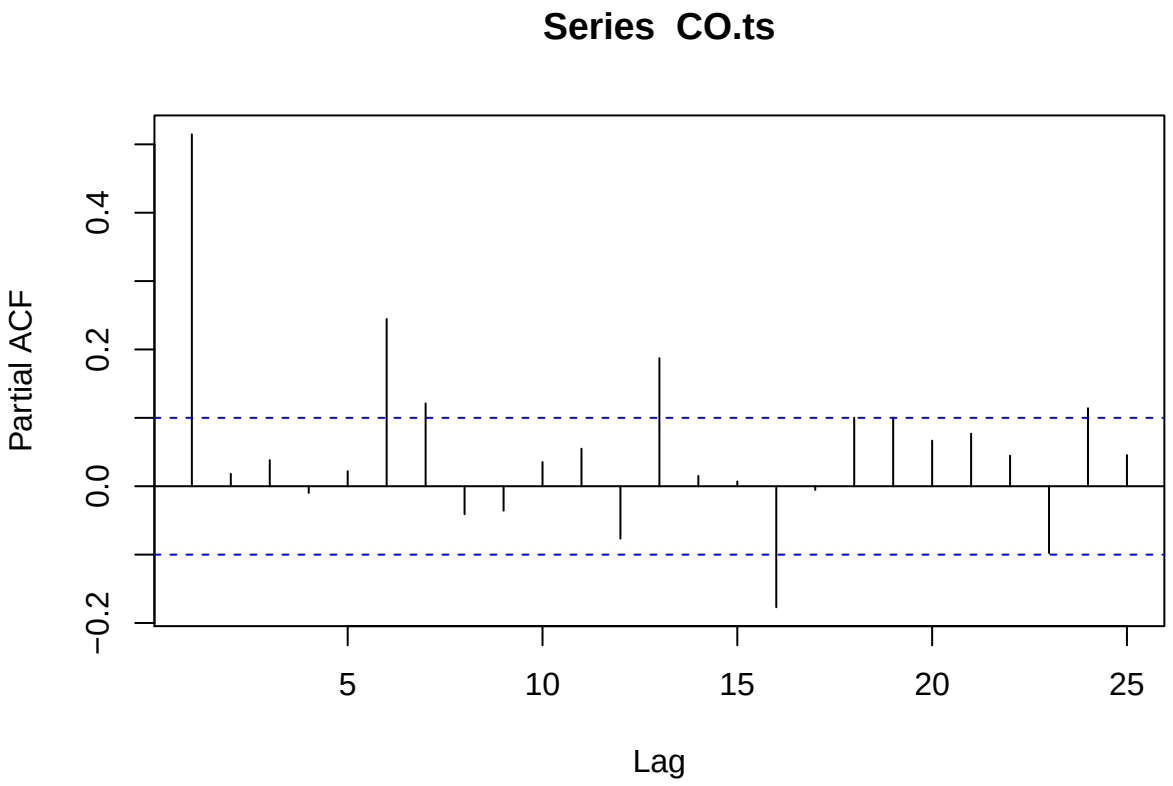



```r
# ACF of simulation
acf(NO.uni.sim.ts)
```

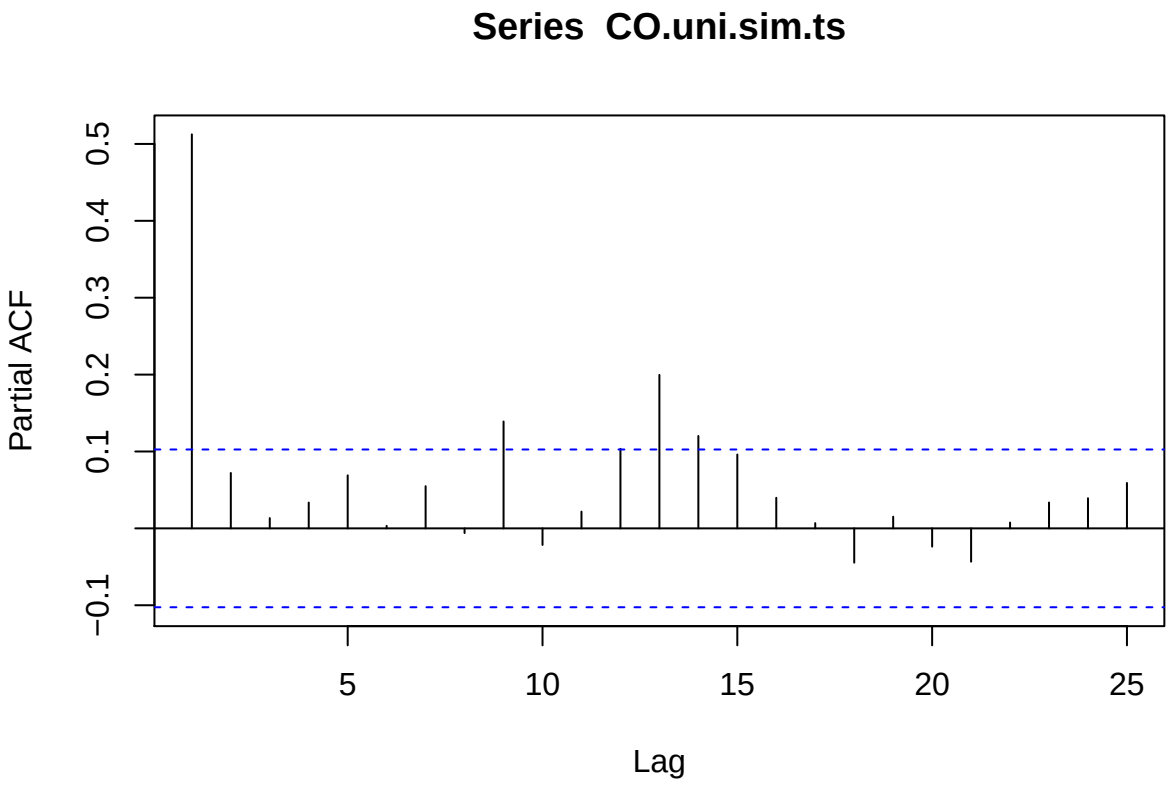


The ACFs look very similar; they are both significant for all 25 lags in view on the plot and have some sinusoidal behavior.

```
# PACF of observations
pacf(CO.ts)
```




```
# PACF of simulation
pacf(CO.uni.sim.ts)
```

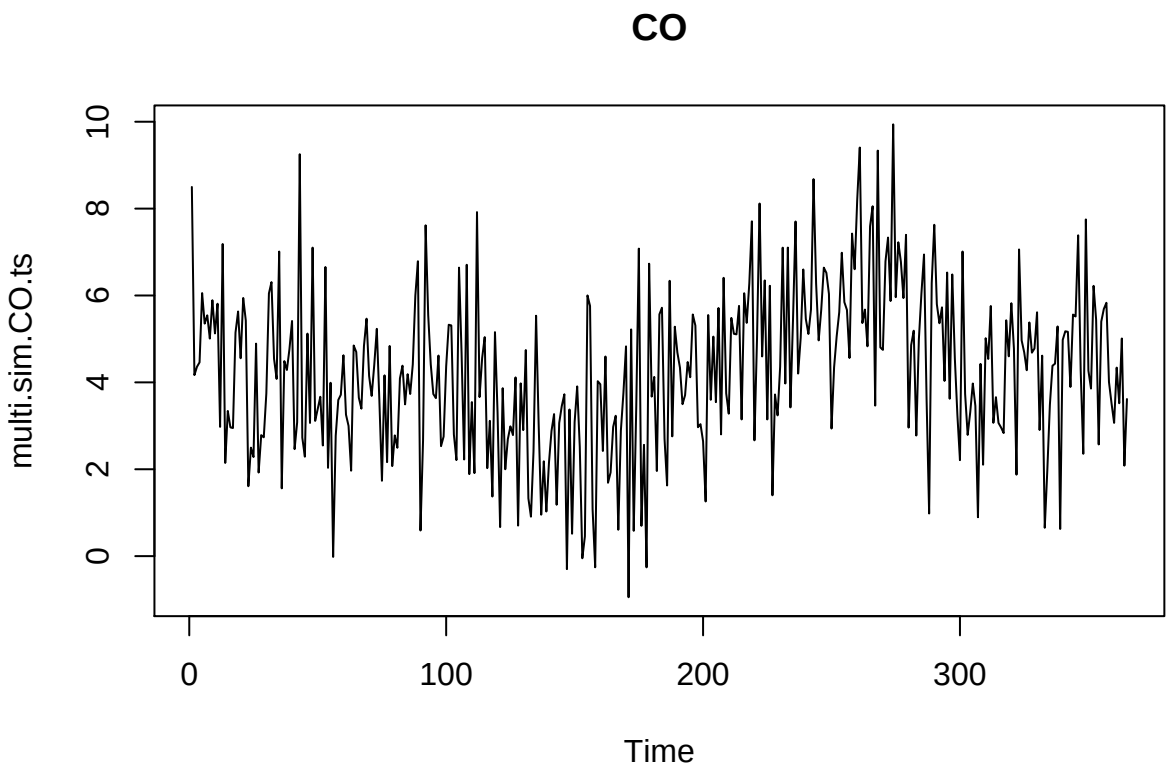


The ACFs look more similar, though the PACF of the time series and simulation are close as well. Both cutoff and become significant again, and both are significant for the first lag but not for the second.

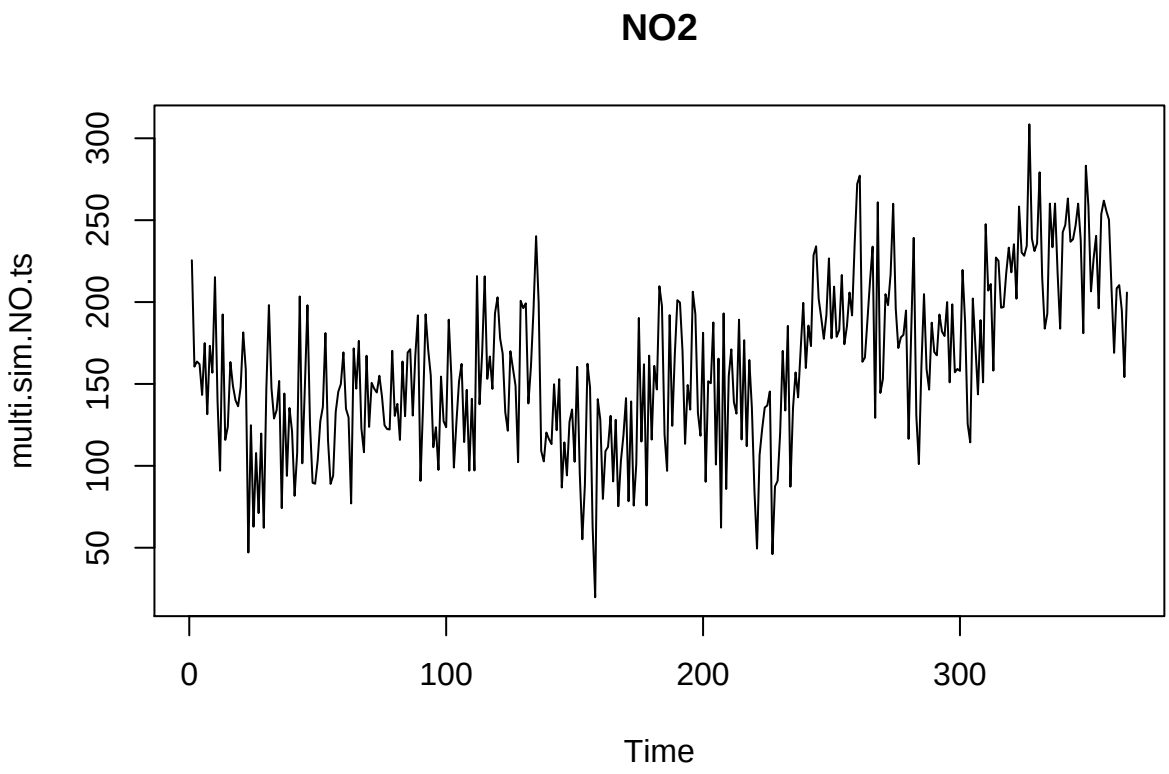
From comparing the ACF and PACF of the time series and simulation, the simulation was able to reproduce the autocorrelation of the time series.

Simulation from Multivariate Model

```
# VARMA model is 2,2

# simulate residuals
set.seed(1)
multi.sim <- VARMAsim(365, phi = varma.model$Phi,
                           theta = varma.model$Theta,
                           sigma = varma.model$Sigma)

# make time series
multi.sim.CO.ts <- ts(multi.sim$series[,1] + CO.mean)
multi.sim.NO.ts <- ts(multi.sim$series[,2] + NO.mean)
plot(multi.sim.CO.ts, main = "CO")
```



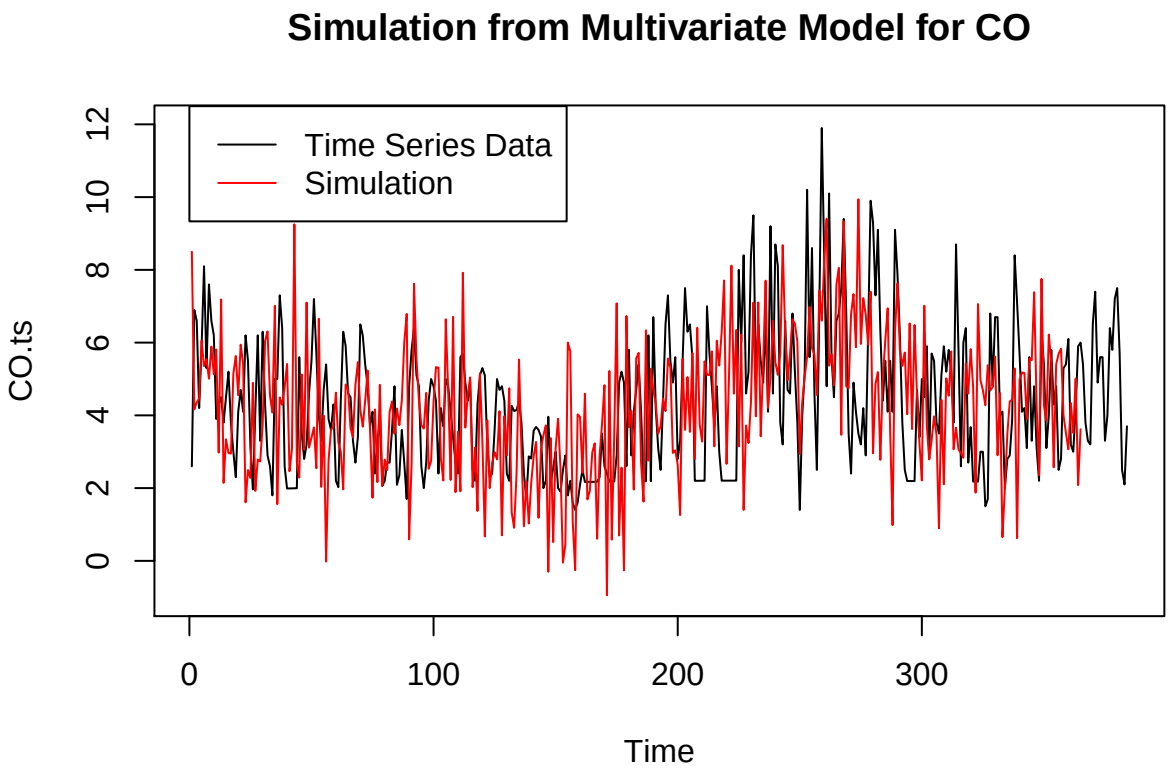

```
plot(multi.sim.NO.ts, main = "NO2")
```



These are the plots of the simulations.

Visualization- CO

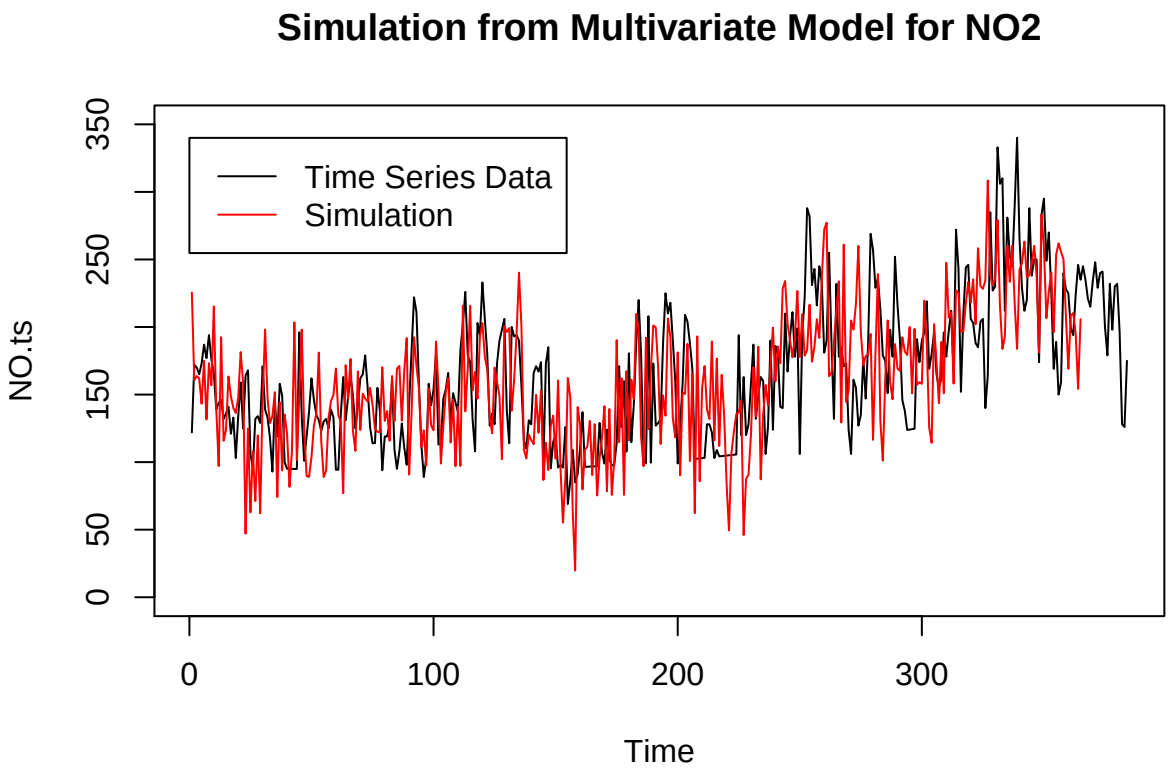
```
# plot simulated values with observations
plot(CO.ts, main = "Simulation from Multivariate Model for CO", col = "black", ylim = c(-1,12))
lines(multi.sim.CO.ts, col = "red")
legend(0, 12.5, legend = c("Time Series Data", "Simulation"), col = c("black", "red"), lwd = 1)
```



The simulation values appear to be similar to the time series data.

Visualization- NO2

```
# plot simulated values with observations
plot(NO.ts, main = "Simulation from Multivariate Model for NO2", col = "black", ylim = c(0,350))
lines(multi.sim.NO.ts, col = "red")
legend(0, 340, legend = c("Time Series Data", "Simulation"), col = c("black", "red"), lwd = 1)
```



The simulation values appear to be similar to the time series data.

Trend- CO

```
# linear model for simulation
CO.multi.sim.lm <- lm(multi.sim.CO.ts ~ next.time.time)
summary(CO.multi.sim.lm)
```

```
##
## Call:
## lm(formula = multi.sim.CO.ts ~ next.time.time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1458 -1.2297 -0.0621  1.2077  5.4711
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6405403  0.1957349  18.599  < 2e-16 ***
## next.time.time 0.0032704  0.0009269   3.528 0.000472 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.866 on 363 degrees of freedom
## Multiple R-squared:  0.03316,    Adjusted R-squared:  0.03049
```

```
## F-statistic: 12.45 on 1 and 363 DF,  p-value: 0.0004721
```

```
summary(CO.lm.trend)
```

```
##
## Call:
## lm(formula = CO.ts ~ time.CO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2485 -1.6980 -0.0525  1.0863  7.3442
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.810929   0.192140  19.834  < 2e-16 ***
## time.CO     0.002876   0.000865   3.325  0.00097 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.879 on 382 degrees of freedom
## Multiple R-squared:  0.02813,    Adjusted R-squared:  0.02558
## F-statistic: 11.06 on 1 and 382 DF,  p-value: 0.0009695
```

We made a linear model for the simulation with time as a predictor to determine if there was a trend to the data. The model was significant, meaning that there is a trend to the simulation. The estimated coefficient value was 0.0033 for the simulation and 0.0029 for the original time series. These coefficients are very close, so the simulation was able to accurately reproduce the trend of the original time series.

Trend- NO2

```
# linear model for simulation
NO.multi.sim.lm <- lm(multi.sim.NO.ts ~ next.time.time)
summary(NO.multi.sim.lm)
```

```
##
## Call:
## lm(formula = multi.sim.NO.ts ~ next.time.time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -134.501  -29.136    2.347   30.415  111.289
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    113.98562    4.42632   25.75   <2e-16 ***
## next.time.time   0.25496    0.02096   12.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.2 on 363 degrees of freedom
## Multiple R-squared:  0.2896, Adjusted R-squared:  0.2876
## F-statistic:   148 on 1 and 363 DF,  p-value: < 2.2e-16
```

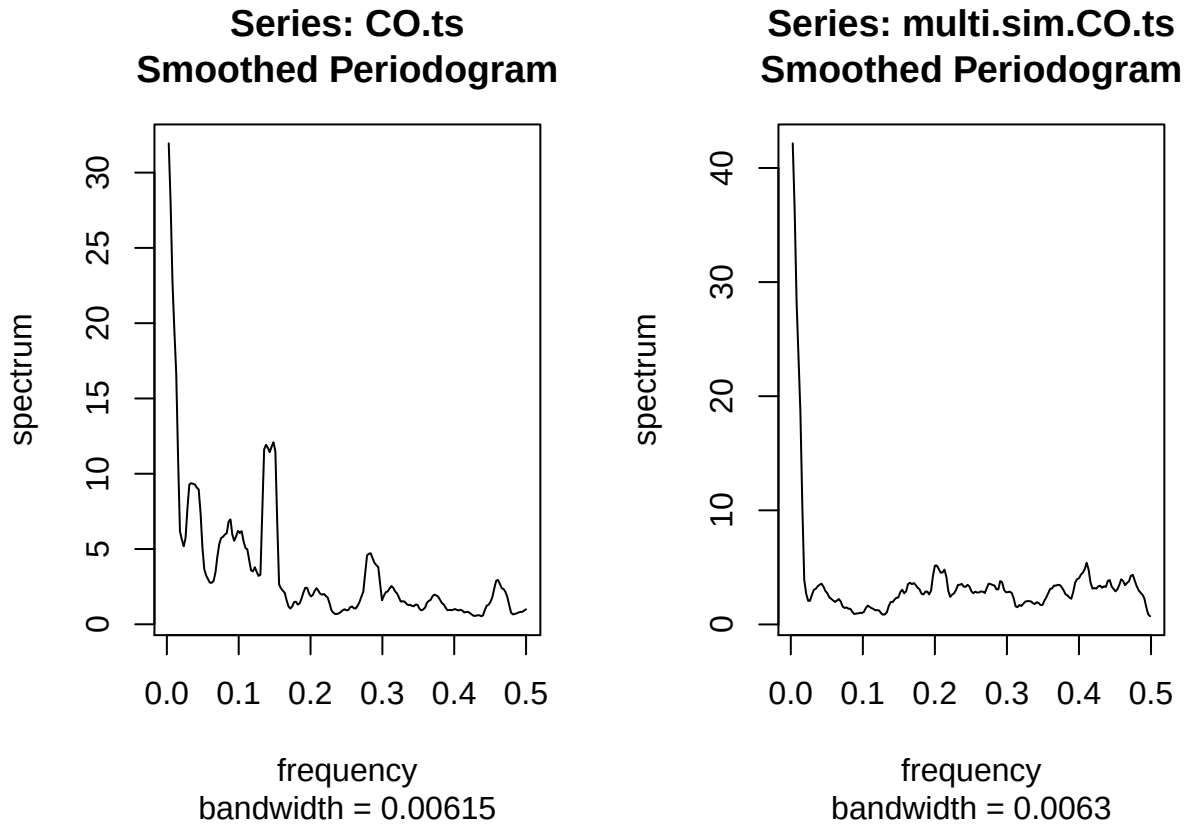
```
summary(NO.lm.trend)
```

```
##
## Call:
## lm(formula = NO.ts ~ time.NO)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -87.389 -34.365   2.159  27.847 137.895
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115.16473    4.40111   26.17   <2e-16 ***
## time.NO       0.25646    0.01981   12.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.04 on 382 degrees of freedom
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3031
## F-statistic: 167.6 on 1 and 382 DF,  p-value: < 2.2e-16
```

We made a linear model for the simulation with time as a predictor to determine if there was a trend to the data. The model was significant, meaning that there is a trend to the simulation. The estimated coefficient value was 0.255 for the simulation and 0.256 for the original time series. These coefficients are very close, so the simulation was able to accurately reproduce the trend of the original time series.

Seasonality- CO

```
# compare periodogram of observations and periodogram of simulation
par(mfrow = c(1,2))
pg.CO <- spec.pgram(CO.ts,spans=9,demean=T,log='no')
pg.CO.multi.sim <- spec.pgram(multi.sim.CO.ts, spans = 9, demean = T, log = 'no')
```

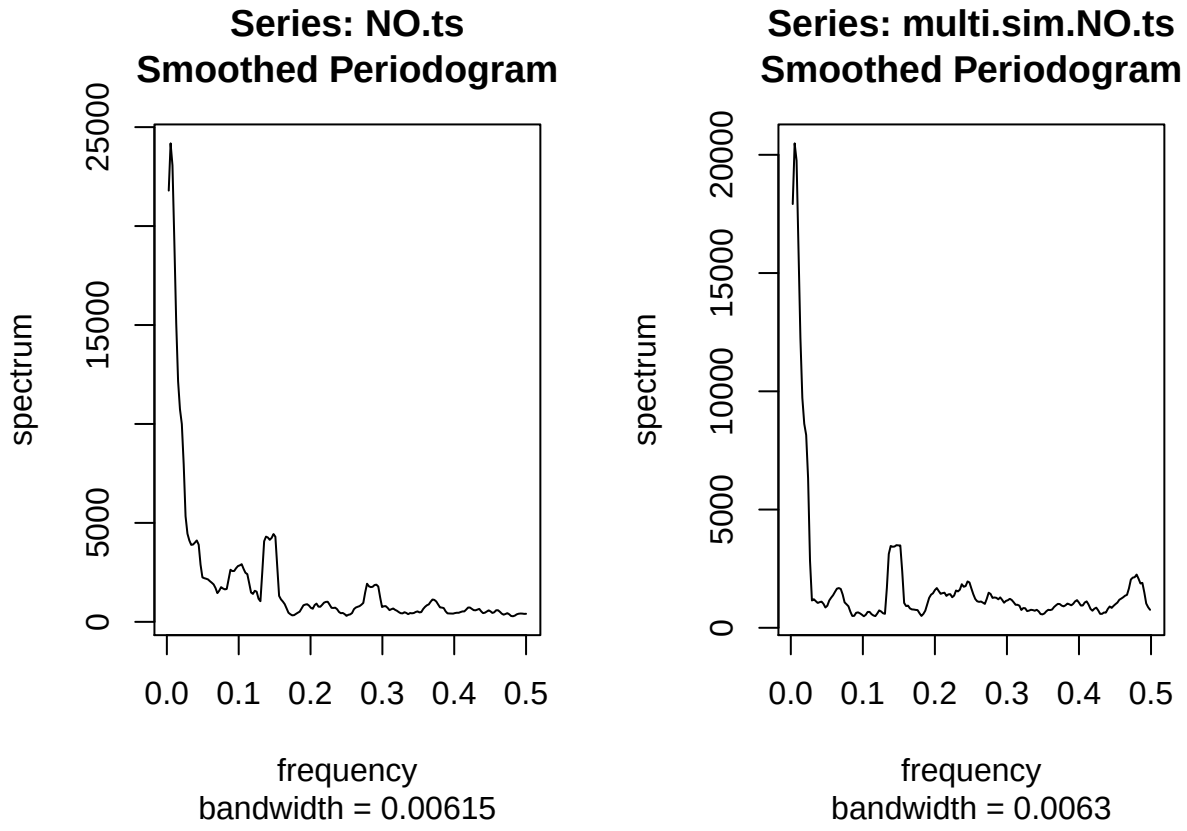



```
par(mfrow = c(1,1))
```

The periodograms of the original observations and simulation are not very similar. There are fewer peaks in the simulation periodogram.

Seasonality- NO2

```
# compare periodogram of observations and periodogram of simulation
par(mfrow = c(1,2))
pg.NO <- spec.pgram(NO.ts,spans=9,demean=T,log='no')
pg.NO.multi.sim <- spec.pgram(multi.sim.NO.ts, spans = 9, demean = T, log = 'no')
```




```
par(mfrow = c(1,1))
```

The periodograms of the original observations and simulation are similar. There are similar peaks in similar locations in both periodograms.

Mean and Variance

```
# mean of observations- CO
mean(CO.ts)
```

```
## [1] 4.364574
```

```
# mean of simulation- CO
mean(multi.sim.CO.ts)
```

```
## [1] 4.239021
```

```
# mean of observations- NO2
mean(NO.ts)
```

```
## [1] 164.5335
```

```r
# mean of simulation- NO2
mean(multi.sim.NO.ts)
```

```
## [1] 160.6439
```

```r
# variance of observations- CO
var(CO.ts)
```

```
## [1] 3.622965
```

```r
# variance of simulation- CO
var(multi.sim.CO.ts)
```

```
## [1] 3.59114
```

```r
# variance of observations- NO2
var(NO.ts)
```

```
## [1] 2657.722
```

```r
# variance of simulation- NO2
var(multi.sim.NO.ts)
```

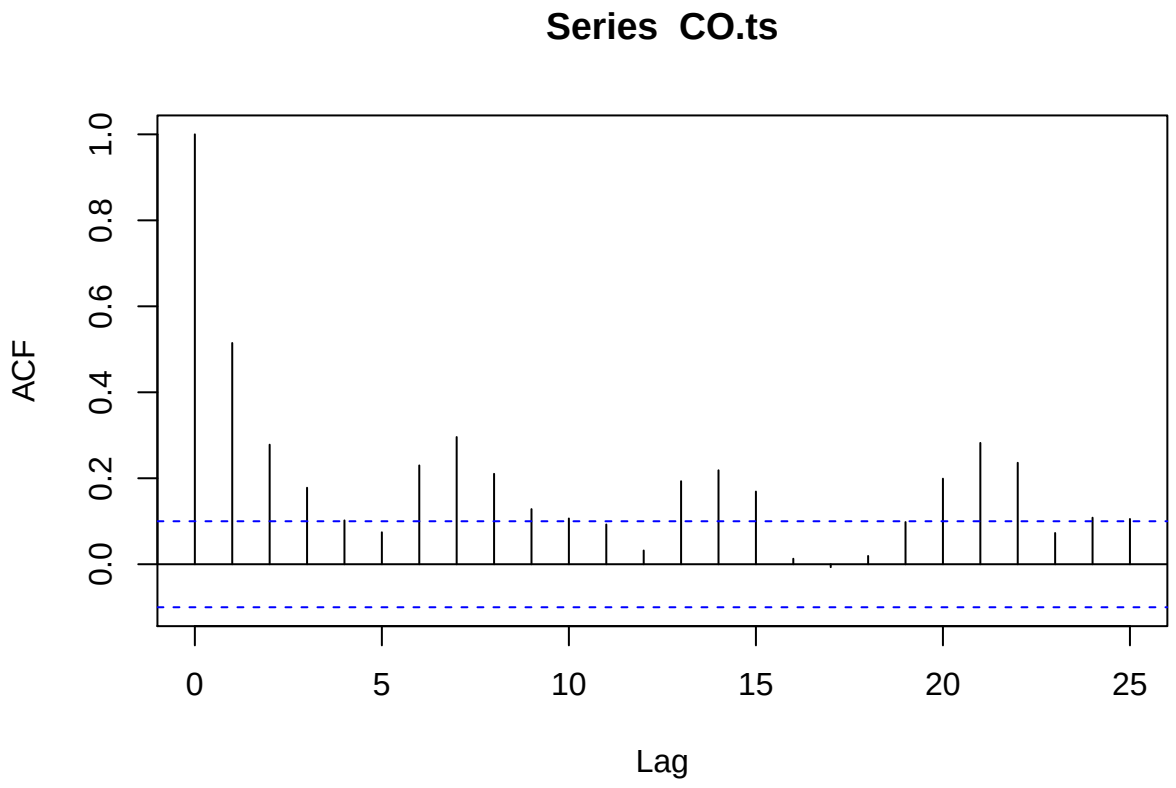
```
## [1] 2499.248
```

CO: The mean of the original time series is 4.36, and the mean of the simulation is 4.24. The means are very close together, which was apparent from the plot of both the time series and simulation together.
The variance of the original time series is 3.62, and the variance of the simulation is 3.59. The variances are also similar.

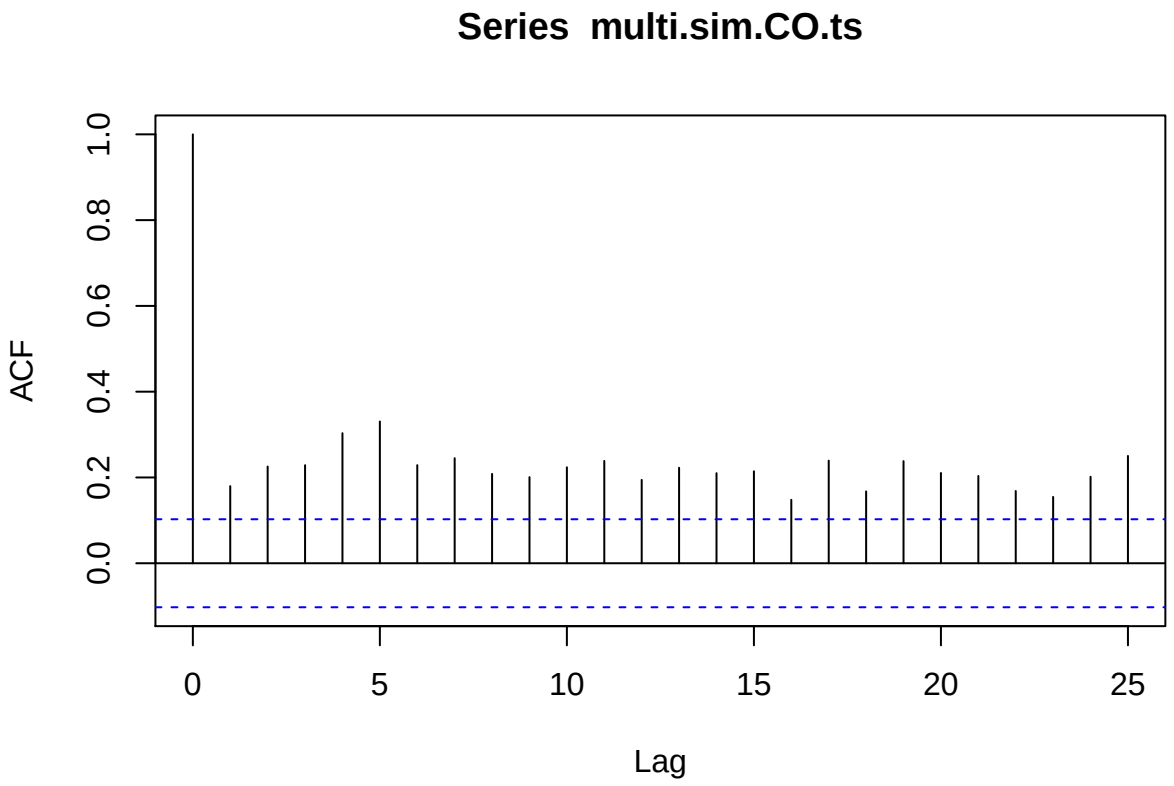
NO2: The mean of the original time series is 164.53, and the mean of the simulation is 160.64. The means are close together, which was apparent from the plot of both the time series and simulation together.
The variance of the original time series is 2657.722, and the variance of the simulation is 2499.248. The variances are also similar.

Auto-Correlation- CO

```r
# ACF of observations
acf(CO.ts)
```

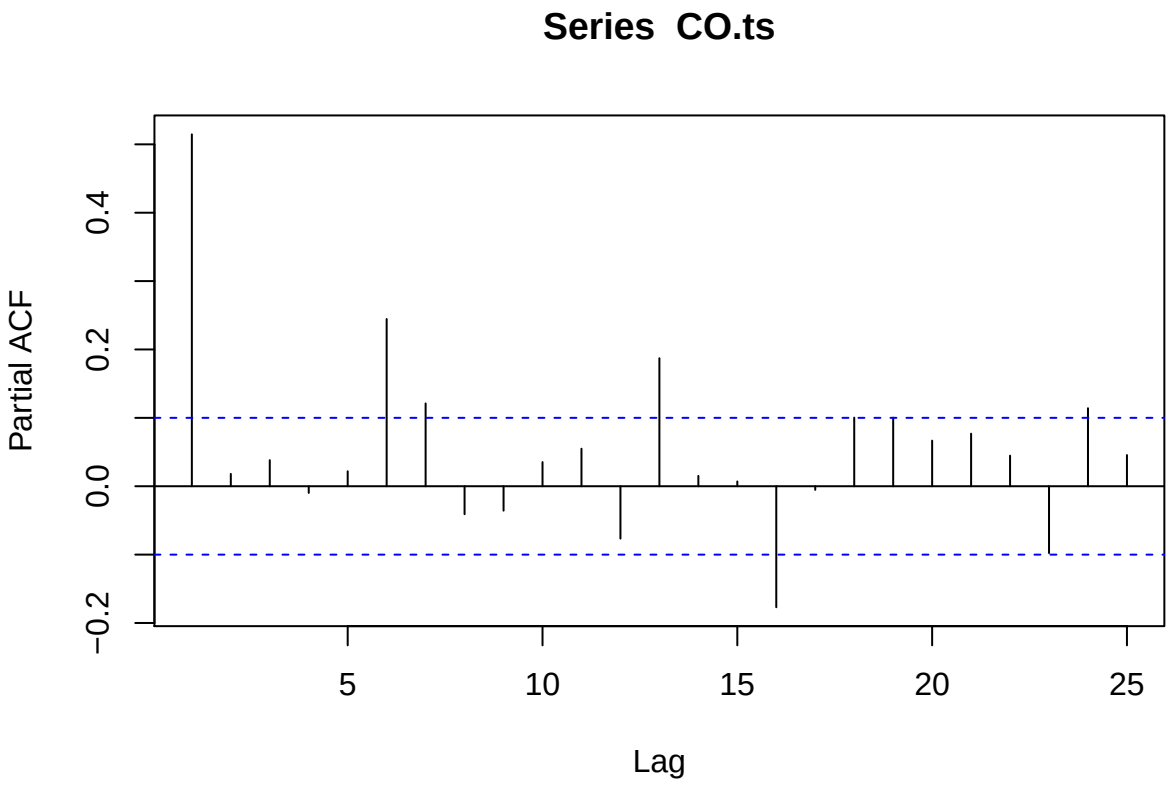



```
# ACF of simulation
acf(multi.sim.CO.ts)
```

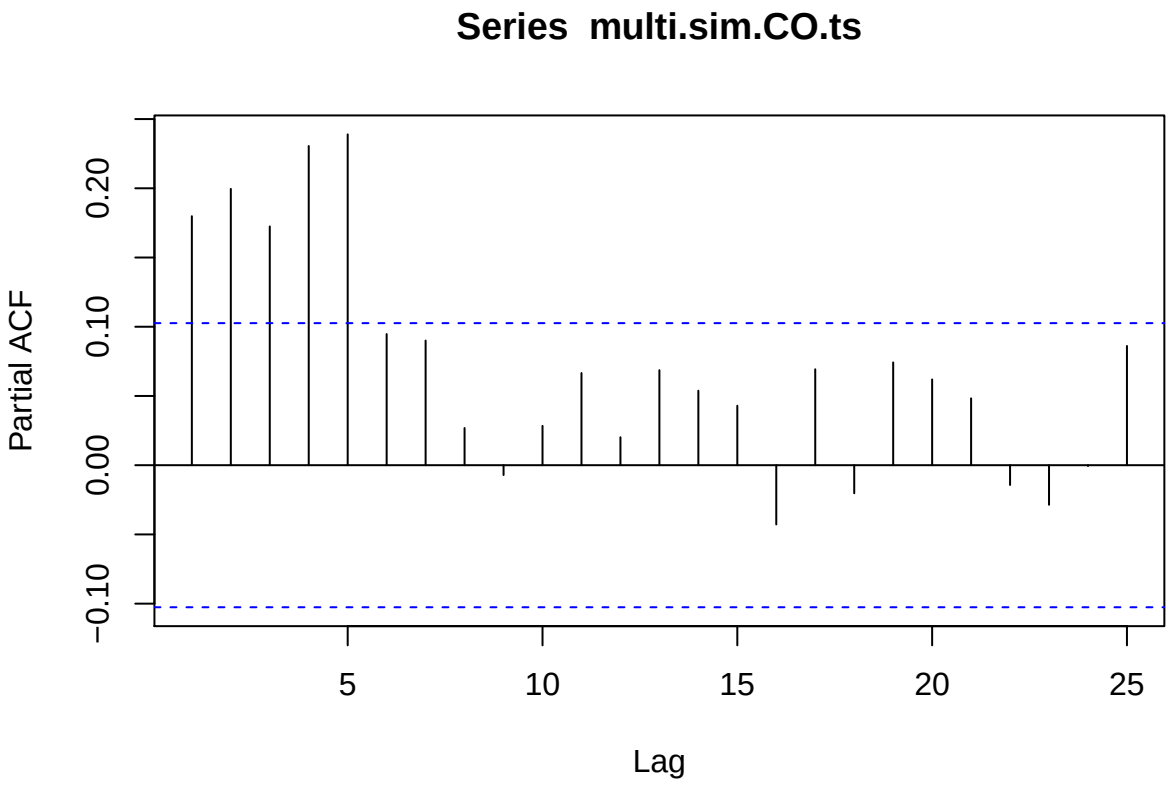


The simulation got close to reproducing the ACF of the time series, though the ACF of the simulation is significant for all of the lags in view and the ACF of the time series cuts off and becomes significant again. Both have some sinusoidal behavior.

```
# PACF of observations
pacf(CO.ts)
```




```
# PACF of simulation
pacf(multi.sim.CO.ts)
```

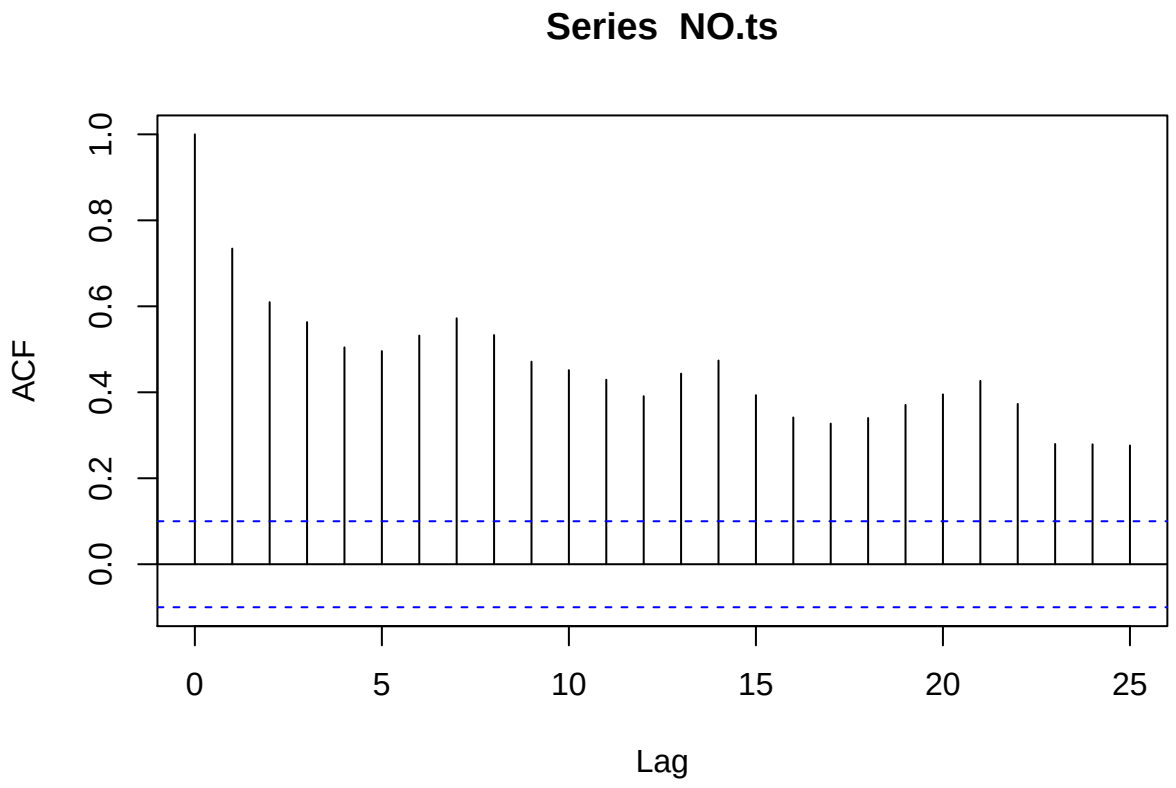


The PACF of the simulation does not match the PACF of the time series. The PACF of the simulation cuts off after 5 lags, but the PACF of the original time series cuts off after 1 lag and then becomes significant again.

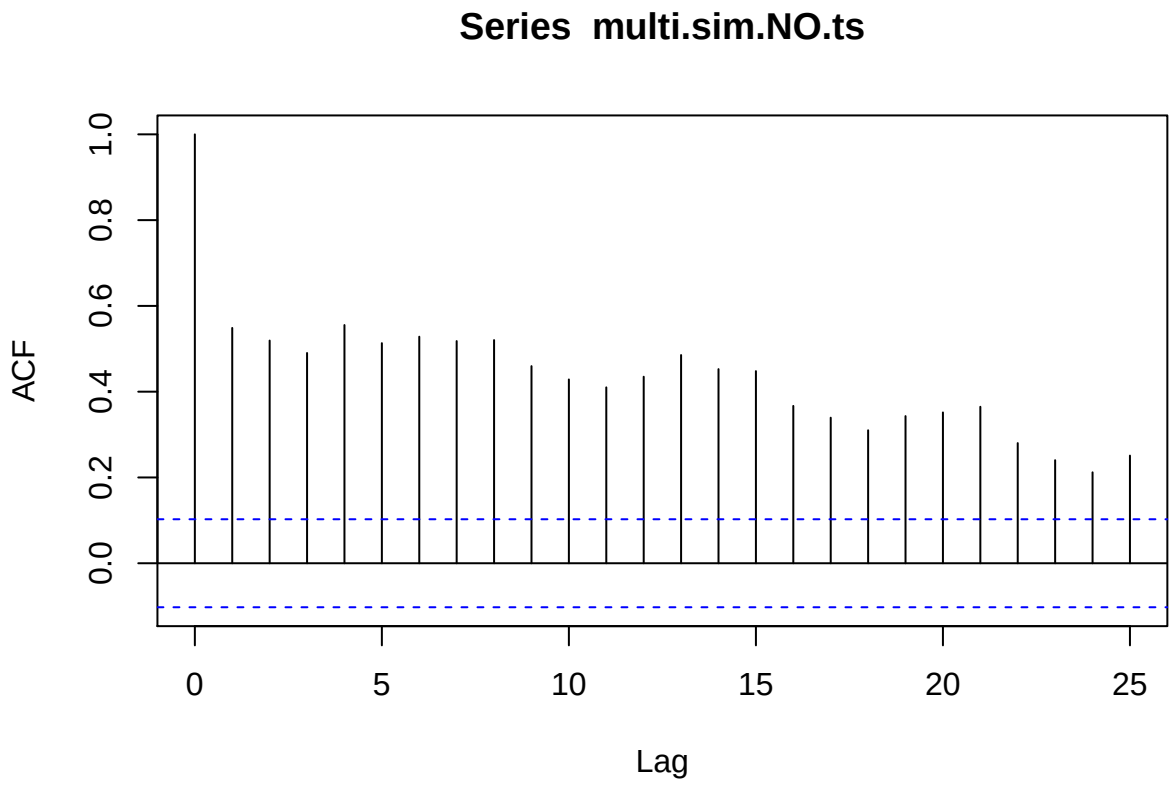
From comparing the ACF and PACF of the time series and simulation, the simulation was somewhat able to reproduce the autocorrelation of the time series.

Auto-Correlation- NO2

```
# ACF of observations
acf(NO.ts)
```

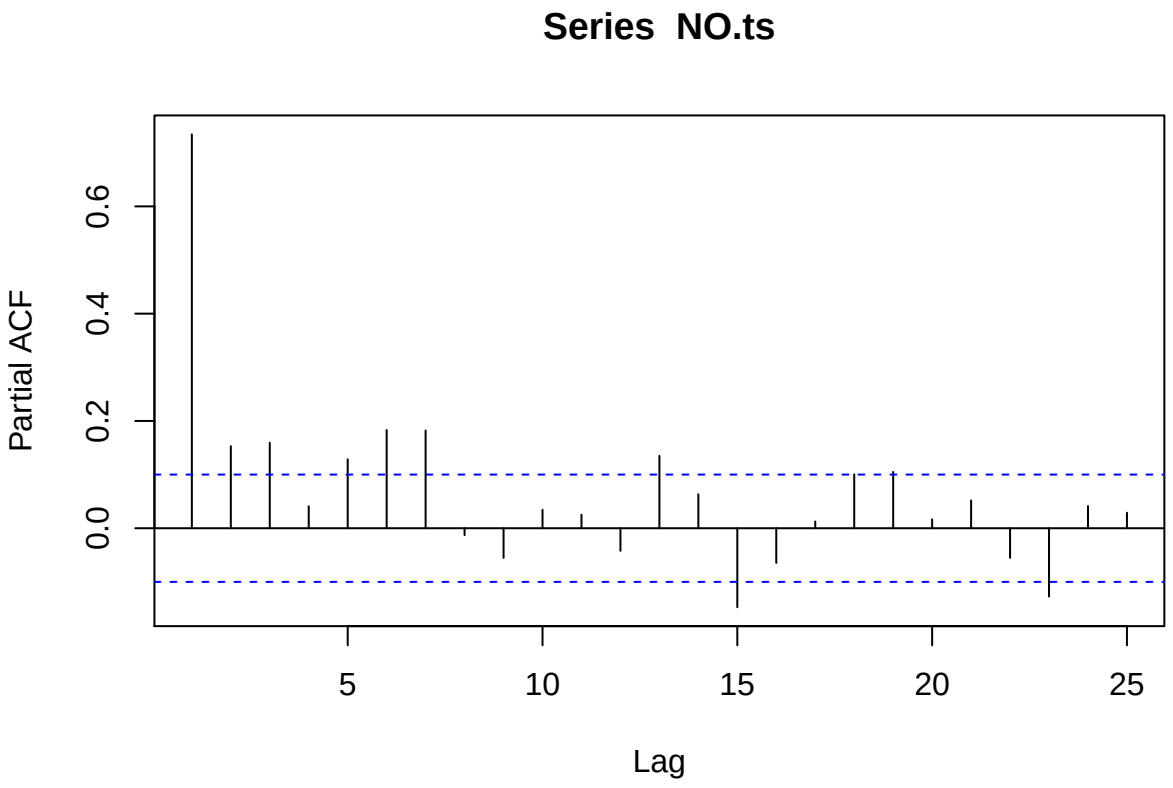



```
# ACF of simulation
acf(multi.sim.NO.ts)
```

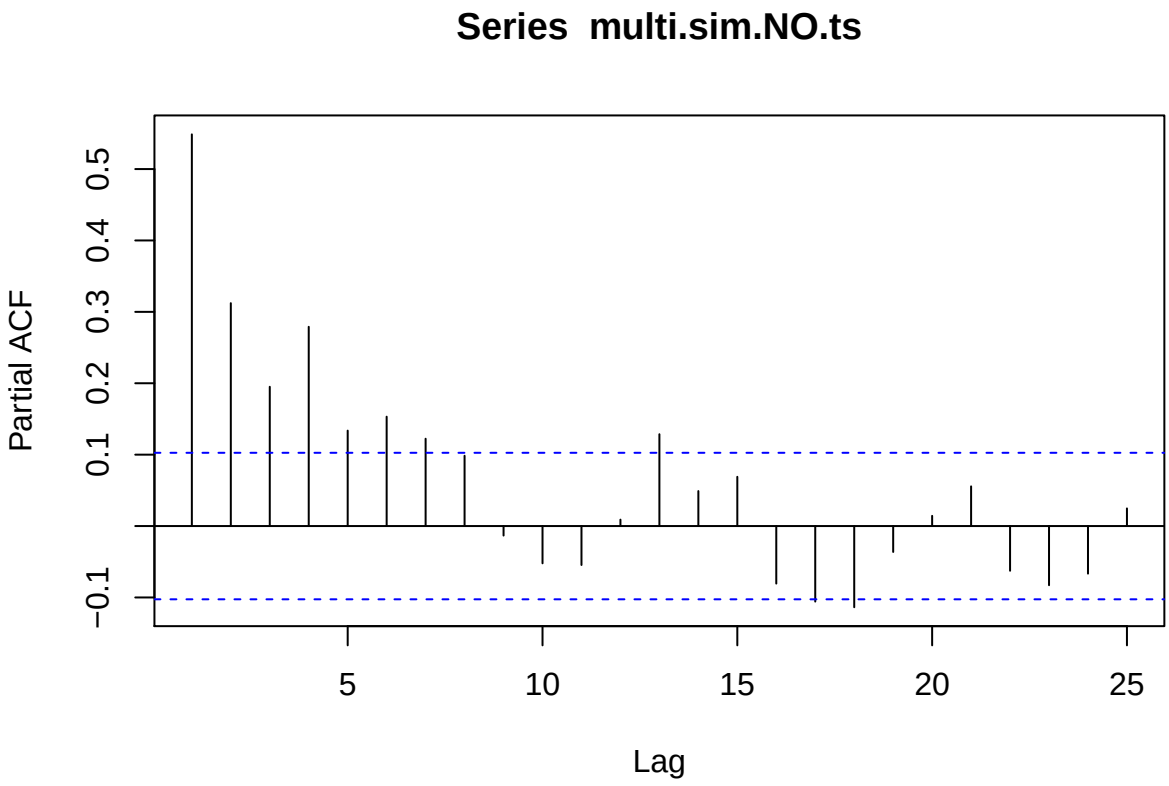


The simulation got very close to reproducing the ACF of the time series. They both are significant for all lags in view on the plot and display some sinusoidal behavior.

```
# PACF of observations
pacf(NO.ts)
```




```
# PACF of simulation
pacf(multi.sim.NO.ts)
```



The ACFs look more similar. The PACF shows more significance at the smaller lags than the original time series.

From comparing the ACF and PACF of the time series and simulation, the simulation was somewhat able to reproduce the autocorrelation of the time series.

Cross-Correlation

Cross-Correlation of Time Series

```
cor(CO.ts, NO.ts)
```

```
## [1] 0.6076964
```

```
#correlation = 0.608
```

Cross-Correlation of Univariate Models

```
cor(CO.uni.sim.ts, NO.uni.sim.ts)
```

```
## [1] 0.1255311
```

```
#correlation = 0.126
```

Cross-Correlation of Multivariate Models

```
cor(multi.sim.CO.ts, multi.sim.NO.ts)
```

```
## [1] 0.5604455
```

```
#correlation = 0.560
```

The cross-correlation for the original time series is 0.608. The cross-correlation for the univariate models is much lower, at 0.126. The cross-correlation for the multivariate models is 0.560, which is closer to the original value. Using a multivariate model better preserved the correlation between the time series.

Bonus

Create Forecasts

Forecast for CO

```
nextweek.time.time <- c(1:(7))
nextweek.time.CO <- data.frame(time.CO = nextweek.time.time)

E_Y.pred.CO <- predict(CO.lm, newdata = nextweek.time.CO)
e_t.pred.CO <- forecast(CO.residuals, h=7)
CO.forecast <- E_Y.pred.CO + e_t.pred.CO$mean
```

Forecast for NO2

```
nextweek.time.NO <- data.frame(time.NO = nextweek.time.time)
E_Y.pred.NO <- predict(NO.lm, newdata = nextweek.time.NO)
e_t.pred.NO <- forecast(NO.residuals, h=7)
NO.forecast <- E_Y.pred.NO + e_t.pred.NO$mean
```

Forecast for CO and NO2 (Multivariate Model)

```
CO.NO.forecast <- VARMApred(varma.model, h=7)
```

```
## Predictions at origin  384
##  CO.lm.residuals NO.lm.residuals
##        -0.064582         -7.5854
##         0.397990          0.1858
##         0.328284          4.6002
```

```
##          0.017780          2.4132
##         -0.153102         -1.0418
##         -0.107949         -2.0728
##          0.008463         -0.9135
## Standard errors of predictions
##       [,1]  [,2]
## [1,] 1.543 30.38
## [2,] 1.664 32.41
## [3,] 1.683 32.59
## [4,] 1.683 32.67
## [5,] 1.689 32.78
## [6,] 1.690 32.86
## [7,] 1.691 32.86
```

```
e_t.pred.CO.multi <- CO.NO.forecast$pred[,1]
e_t.pred.CO.lower <- CO.NO.forecast$pred[,1] - 1.96*CO.NO.forecast$se.err[,1]
e_t.pred.CO.upper <- CO.NO.forecast$pred[,1] + 1.96*CO.NO.forecast$se.err[,1]
CO.multi.forecast<- E_Y.pred.CO + e_t.pred.CO.multi

e_t.pred.NO.multi <- CO.NO.forecast$pred[,2]
e_t.pred.NO.lower <- CO.NO.forecast$pred[,2] - 1.96*CO.NO.forecast$se.err[,2]
e_t.pred.NO.upper <- CO.NO.forecast$pred[,2] + 1.96*CO.NO.forecast$se.err[,2]
NO.multi.forecast <- E_Y.pred.NO + e_t.pred.NO.multi
```

Comparison based on MSE

```
#CO MSE
mean((CO.forecast-air.test$CO.GT.)^2)
```

```
## [1] 4.441717
```

```
# MSE = 4.71

#NO2 MSE
mean((NO.forecast-air.test$NO2.GT.)^2)
```

```
## [1] 542.948
```

```
# MSE = 542.95

#CO and NO2 MSE - Multivariate
mean((CO.multi.forecast-air.test$CO.GT.)^2)
```

```
## [1] 4.40135
```

```
# MSE = 4.40
mean((NO.multi.forecast-air.test$NO2.GT.)^2)
```

```
## [1] 543.2495
```

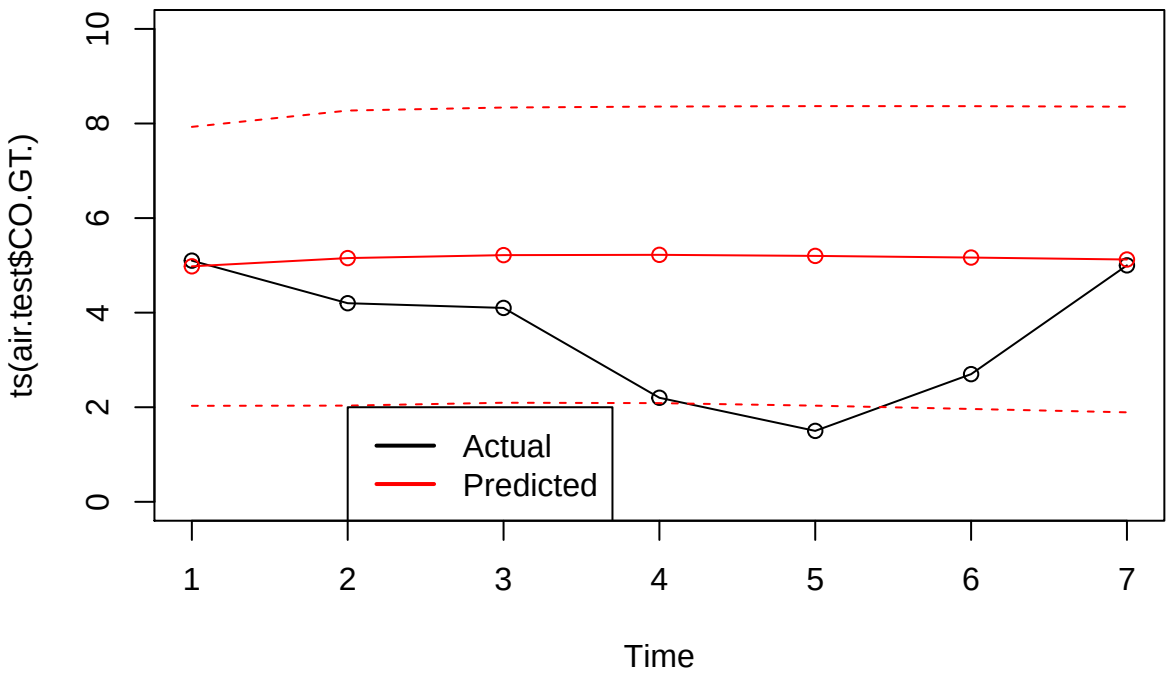
```
# MSE = 543.25
```

The MSE for the CO forecast from the univariate model is higher than that from the multivariate model which implies that the multivariate model models the residuals of CO better. Conversely, the MSE for the univariate model for NO2 is smaller than that from the multivariate model which implies that the univariate model does a better job of modeling the residuals of NO2. Despite all this, the MSE between the univariate and multivariate models are very close.

Visual Comparison

Forecast from CO Univariate Model

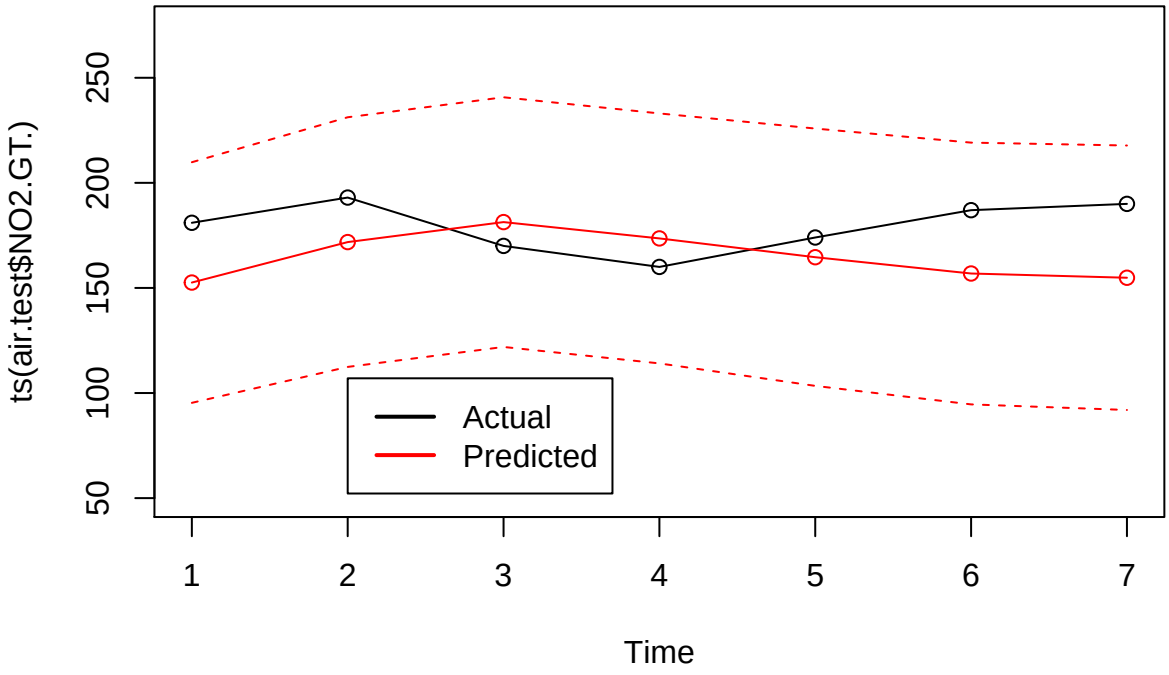
```
# Plot actual values and predicted values
plot(ts(air.test$CO.GT.),type='o',ylim=c(0,10))
lines(ts(CO.forecast),col='red',type='o')
lines(1:7, E_Y.pred.CO + e_t.pred.CO$lower[,2], col = "red", lty = "dashed")
lines(1:7, E_Y.pred.CO + e_t.pred.CO$upper[,2], col = "red", lty = "dashed")
legend(2,2, legend = c("Actual", "Predicted"), lwd = 2, col = c("black", "red"))
```



The predicted values are higher than the actual values for all but one point. There are two actual values that lie outside of the confidence interval of the prediction.

Forecast from NO2 Univariate Model

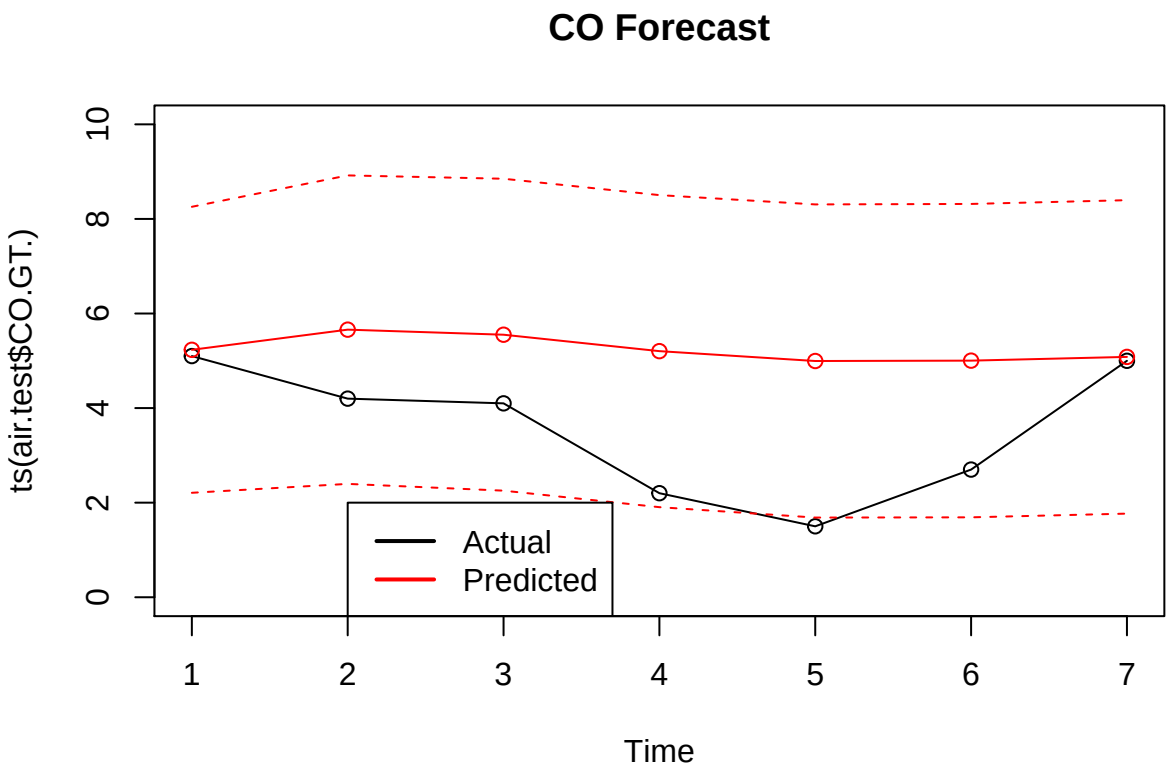
```
# Plot actual values and predicted values
plot(ts(air.test$NO2.GT.),type='o',ylim=c(50,275))
lines(ts(NO.forecast),col='red',type='o')
lines(1:7, E_Y.pred.NO + e_t.pred.NO$lower[,2], col = "red", lty = "dashed")
lines(1:7, E_Y.pred.NO + e_t.pred.NO$upper[,2], col = "red", lty = "dashed")
legend(2,107, legend = c("Actual", "Predicted"), lwd = 2, col = c("black", "red"))
```



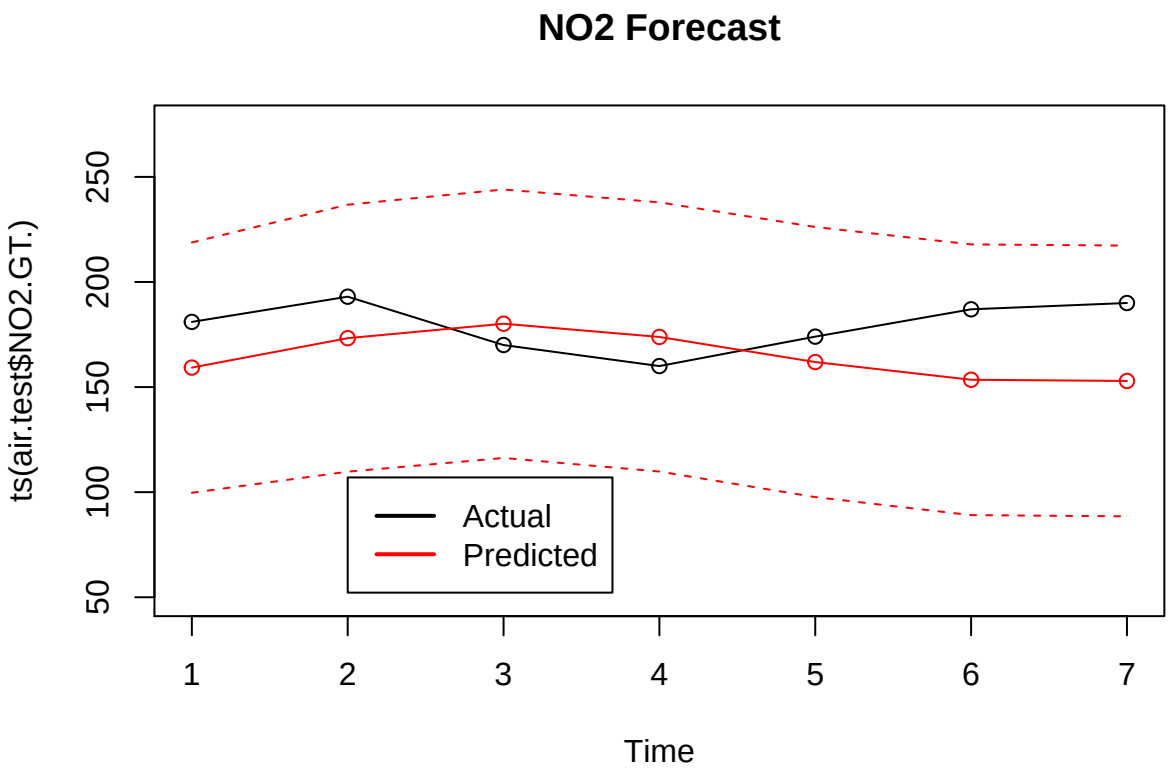
All of the actual values are within the bounds of the confidence interval of the predictions. The actual valyes are higher than the predictions for 5 out of the 7 points.

Forecast from Multivariate Model

```
# CO forecasts
plot(ts(air.test$CO.GT.),type='o', ylim = c(0,10), main = "CO Forecast")
lines(ts(CO.multi.forecast),col='red',type='o')
lines(1:7, E_Y.pred.CO + e_t.pred.CO.lower, col = "red", lty = "dashed")
lines(1:7, E_Y.pred.CO + e_t.pred.CO.upper, col = "red", lty = "dashed")
legend(2,2, legend = c("Actual", "Predicted"), lwd = 2, col = c("black", "red"))
```




```
#NO forecasts
plot(ts(air.test$NO2.GT.),type='o',ylim = c(50,275),main = "NO2 Forecast")
lines(ts(NO.multi.forecast),col='red',type='o')
lines(1:7, E_Y.pred.NO + e_t.pred.NO.lower, col = "red", lty = "dashed")
lines(1:7, E_Y.pred.NO + e_t.pred.NO.upper, col = "red", lty = "dashed")
legend(2,107, legend = c("Actual", "Predicted"), lwd = 2, col = c("black", "red"))
```



Similar to the univariate model's forecast, the multivariate model's forecast for CO predicts that all points are higher than their actual values. There is only one point that is outside the confidence interval and it appears as if it is only just barely outside the interval.

Much like the univariate model, the actual values for NO2 all lie within the multivariate model's confidence interval. 2 predicted points are less than their predicted values and 5 points are greater than their predicted values.