Project 2

Steven Pham

Weds Dec 8th

Load data and impute missing values

Comment: I changed the column names to NO2 and time & created a time series model NO2.ts

```
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 NO2 and impute missing values
AQdata = airquality["NO2.GT."]
AQdata = na_interpolation(AQdata)

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

colnames(dailyAQ) <- c("time" , "NO2")


# create time series of NO2
NO2.ts <- ts(dailyAQ[,2])
```

Modeling Seasonality

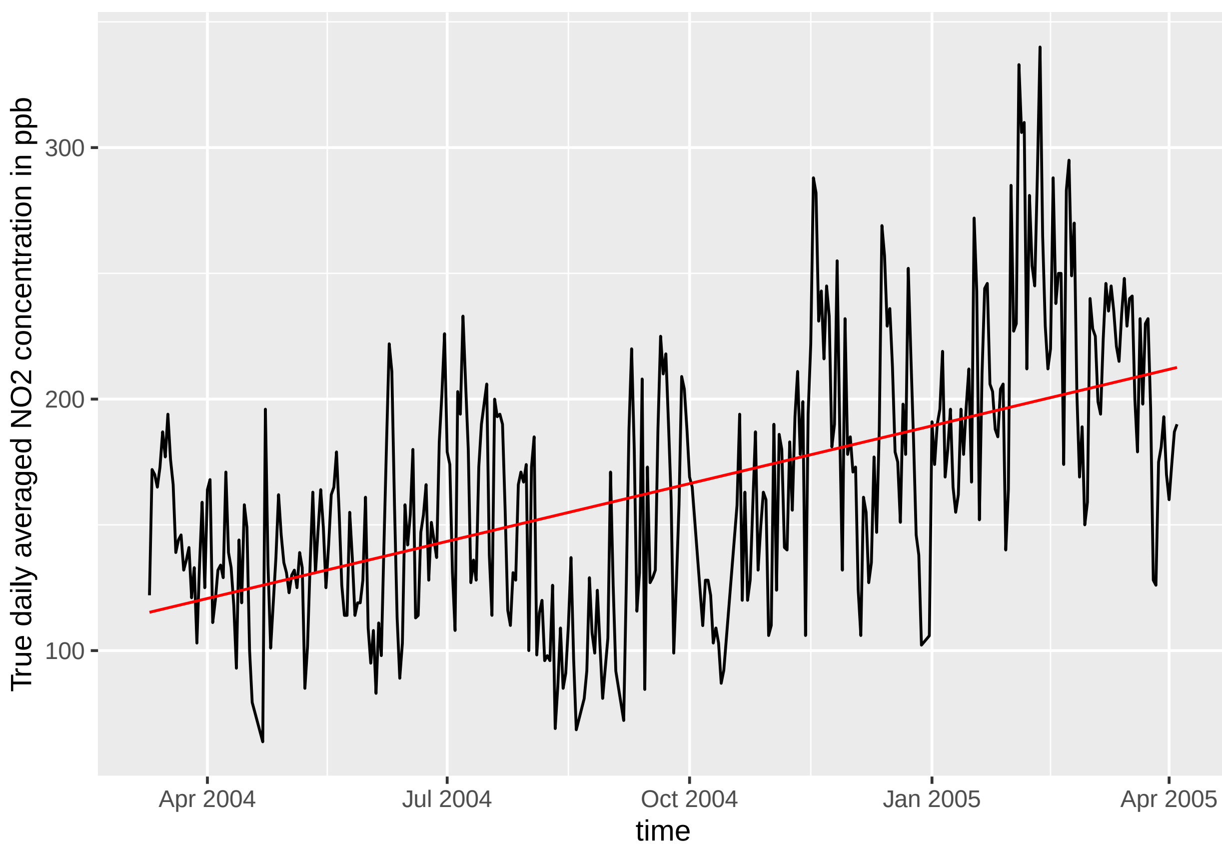
Create a model NO2.trendseason and plot the NO2 fitted values to actual values

###Comment: I created a linear model NO2.trendseason, plotted the fitted values of the linear model as a trendline.

```r
NO2.trendseason <- lm(NO2 ~ time, data = dailyAQ)
summary(NO2.trendseason)
```

```
##
## Call:
## lm(formula = NO2 ~ time, data = dailyAQ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -87.960 -33.175   2.301  28.907 140.401
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.001e+03  2.500e+02  -12.01   <2e-16 ***
## time         2.496e-01  1.971e-02   12.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.99 on 389 degrees of freedom
## Multiple R-squared:  0.2919, Adjusted R-squared:   0.29
## F-statistic: 160.3 on 1 and 389 DF,  p-value: < 2.2e-16
```

```r
ggplot(dailyAQ , aes(x = time , y=NO2)) + geom_line() + ylab("True daily averaged NO2 concentration in p
```

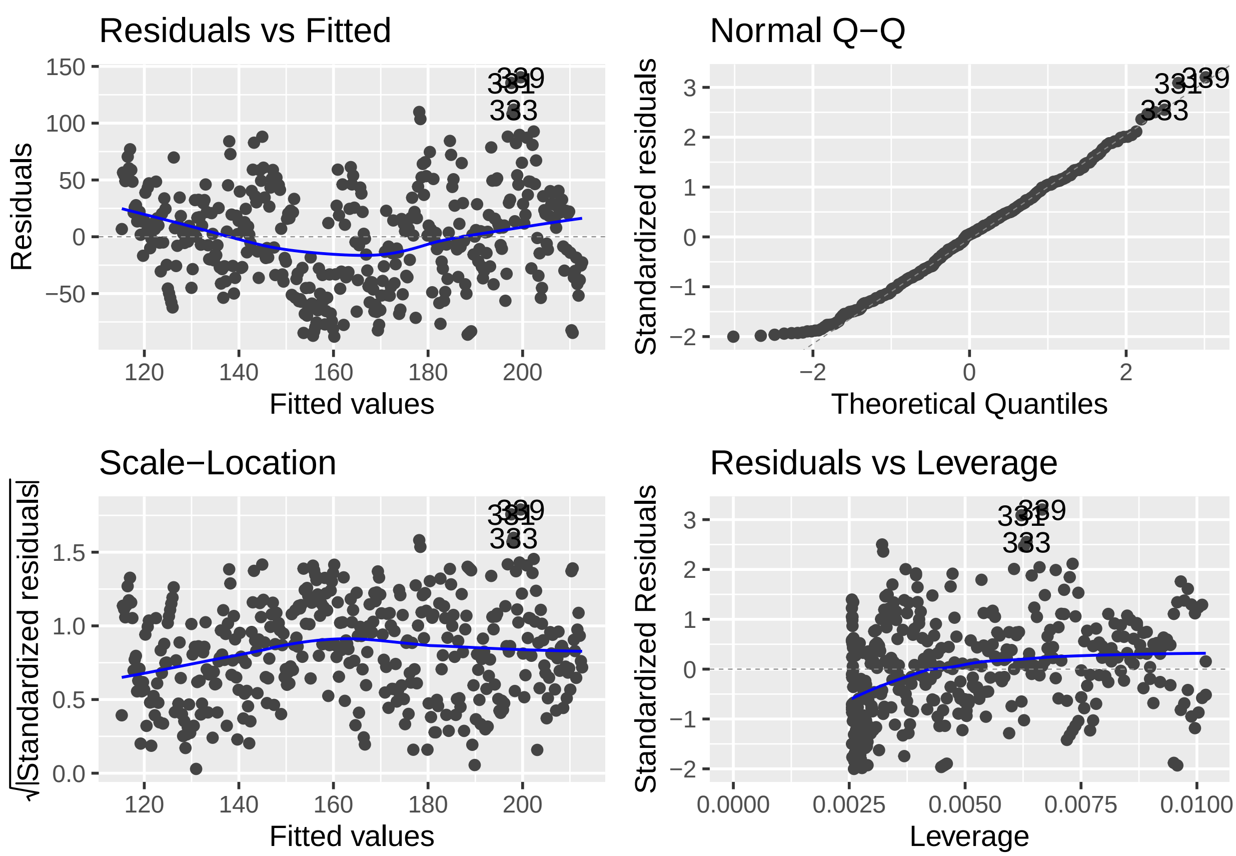


Takeaway: Linear Model NO2.trendseason is a significant model and time is significant to the NO2

concentration. The graph shows a linear increase which suggest that the model is non-stationary. There are two choices, build a residual model or a first difference model.

Create diagnostic plots for NO2.trendseason

```
autoplot(NO2.trendseason, labels.id = NULL)
```

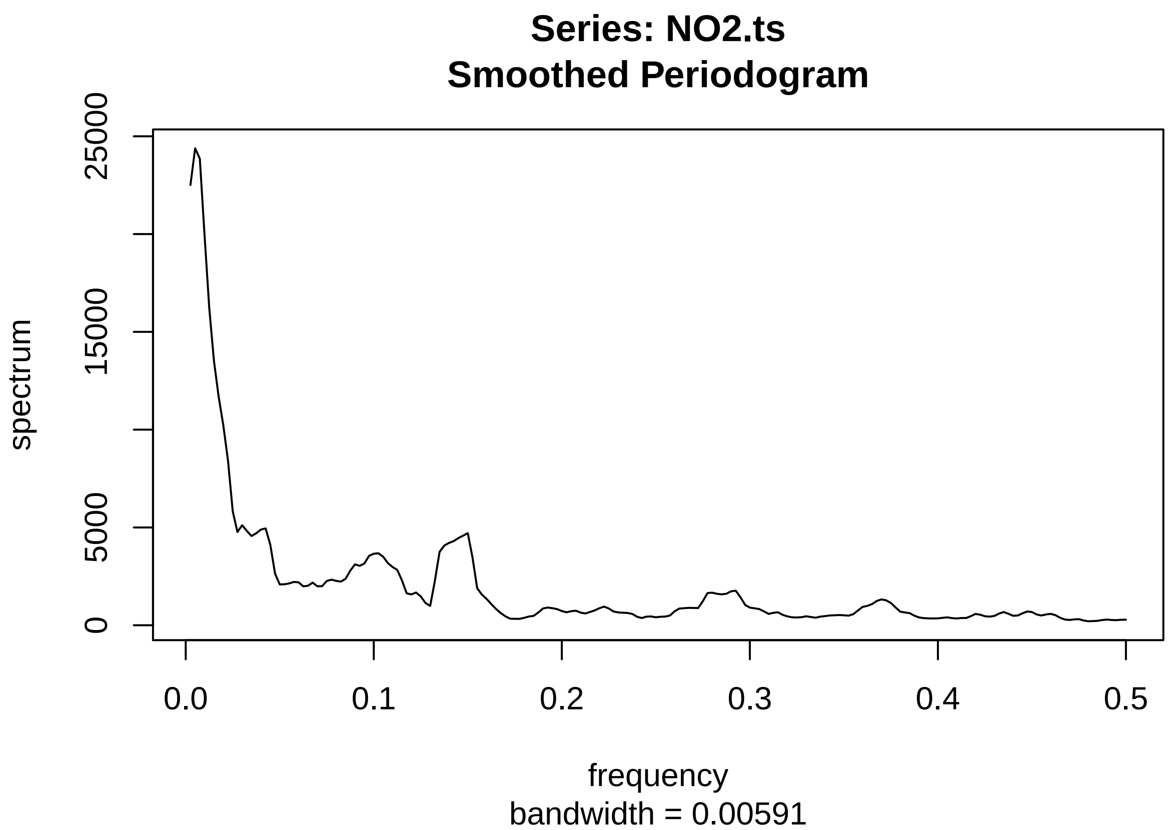


Takeaway: Here I confirm that the linear model is valid before continuing with the time series analysis. Residual vs. fitted graph suggests the mean is not constant, QQ plot looks good with a couple of outlier points.

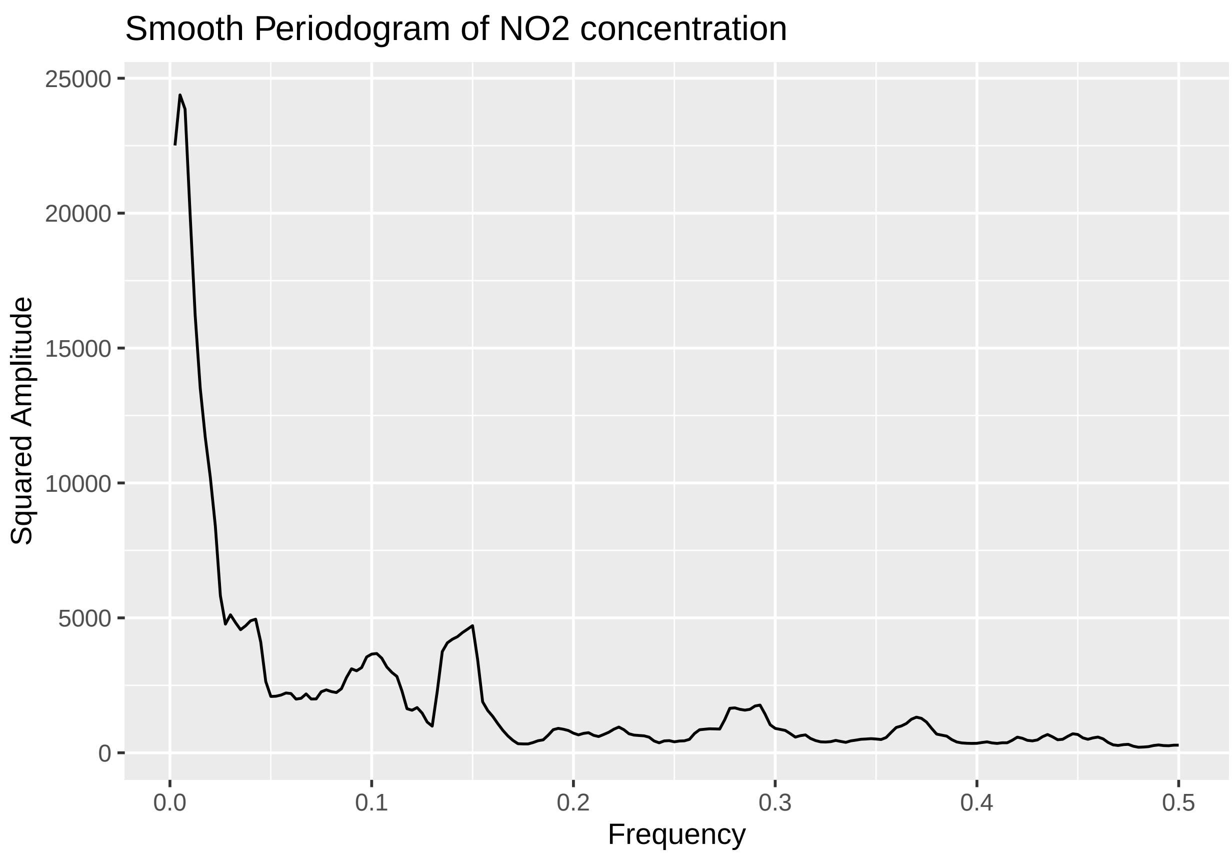
Spectral Analysis

Comment: I created a periodogram, found the max omega and period

```
NO2.pg <- spec.pgram(NO2.ts , spans=9 , demean=T , log="no")
```




```
NO2.spec <- data.frame(freq=NO2.pg$freq , spec=NO2.pg$spec)
ggplot(NO2.spec) + geom_line(aes(x=freq , y=spec)) + ggtitle("Smooth Periodogram of NO2 concentration ")
```




```
max.omega <- NO2.pg$freq[which(NO2.pg$spec==max(NO2.pg$spec))]
max.omega
```

```
## [1] 0.005
```

```
1/max.omega
```

```
## [1] 200
```

```
sorted.spec <- sort(NO2.pg$spec, decreasing=T, index.return=T)
sorted.omegas <- NO2.pg$freq[sorted.spec$ix]
sorted.Ts <- 1/NO2.pg$freq[sorted.spec$ix]
sorted.omegas[1:20]
```

```
##  [1] 0.0050 0.0075 0.0025 0.0100 0.0125 0.0150 0.0175 0.0200 0.0225 0.0250
## [11] 0.0300 0.0425 0.0400 0.0325 0.0275 0.1500 0.0375 0.1475 0.0350 0.1450
```

```
sorted.Ts[1:20]
```

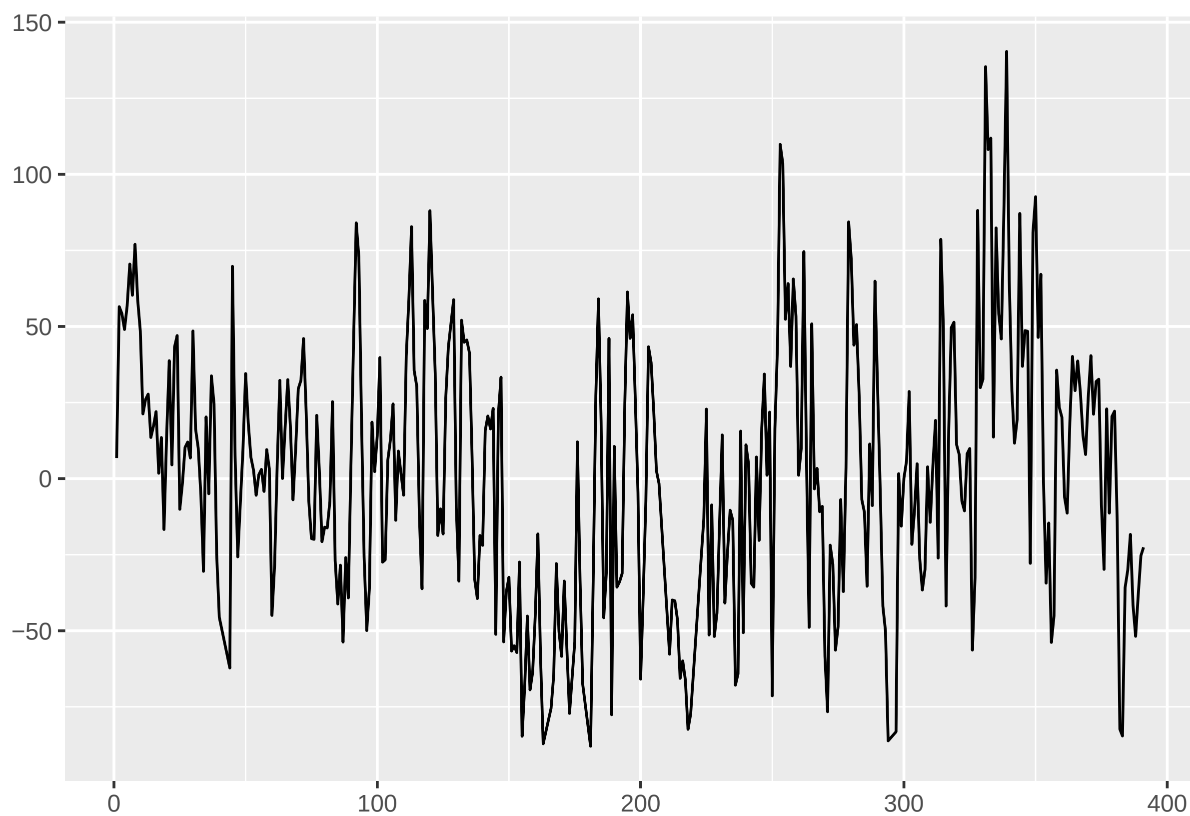
```
##  [1] 200.000000 133.333333 400.000000 100.000000  80.000000  66.666667
##  [7]  57.142857  50.000000  44.444444  40.000000  33.333333  23.529412
## [13]  25.000000  30.769231  36.363636   6.666667  26.666667   6.779661
## [19]  28.571429   6.896552
```

Takeaway: There is definitely noticeable peaks in the smaller frequency that can indicate seasonality. What's odd is that the peak frequency is 0.005 which makes the corresponding period 200 days. Once we sort the peak from greatest to smallest, one thing to notice is that the strong peaks have periods of at least 2 months (60+ days) which could be due to red noise. It is likely that the model will have autoregressive components

Choosing the Ideal Model

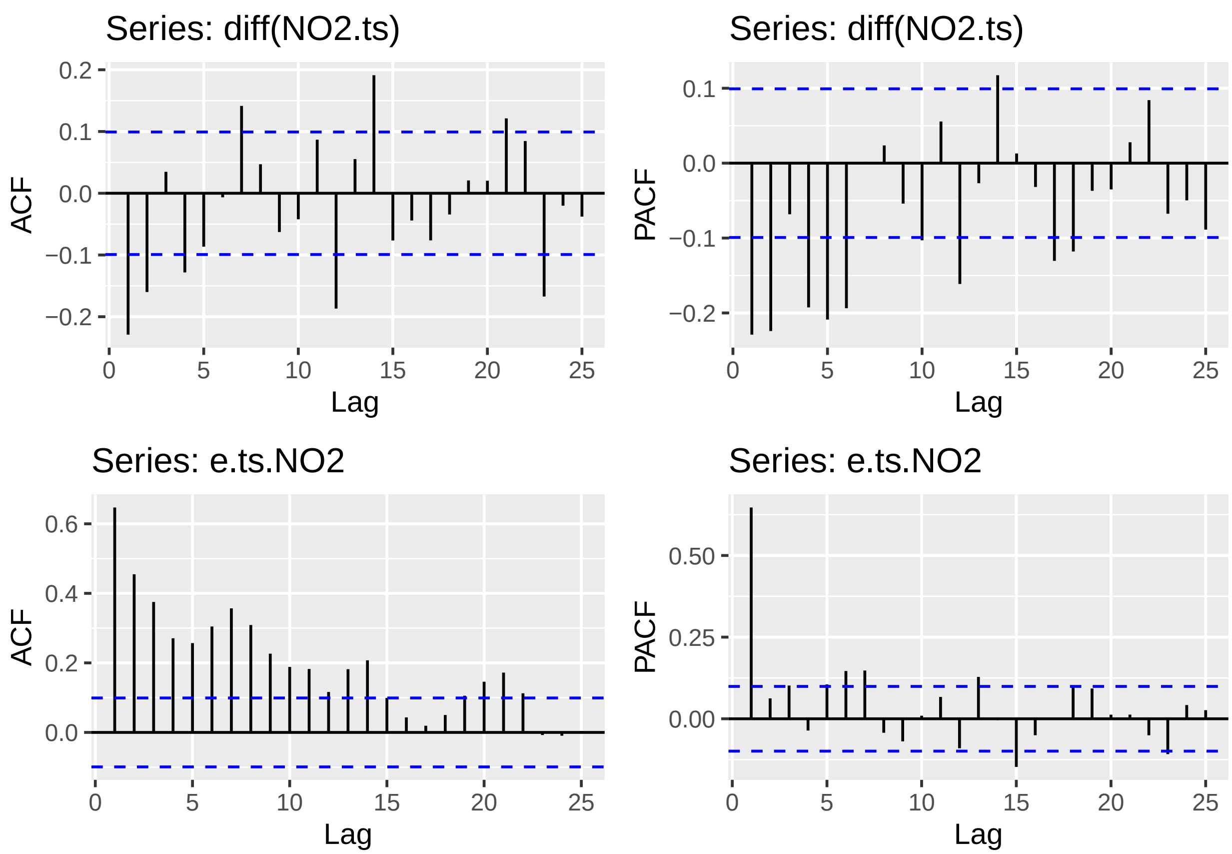
##Residual or first difference Model?

```
e.ts.NO2 <- ts(NO2.trendseason$residuals)
autoplot(e.ts.NO2)
```




```
NO2.trendseason.acf <- ggAcf(diff(NO2.ts))
NO2.trendseason.pacf <- ggPacf(diff(NO2.ts))


NO2.acf <- ggAcf(e.ts.NO2)
NO2.pacf <- ggPacf(e.ts.NO2)
ggarrange(NO2.trendseason.acf , NO2.trendseason.pacf ,NO2.acf,NO2.pacf,nrow=2,ncol=2)
```



Takeaway: The first different model is very different and is not what we are looking for. With regards to the residual model, ACF shows sinusoidal decay , PACF cuts off after 1 lag, not significant. Based on the ACF and PACF alone, I would try a ARMA(3,0). The acf and pacf is an indicator that the residual model is the ideal one.

Auto.arima model

```
auto = auto.arima(e.ts.NO2,approximation = FALSE)
summary(auto)
```

```
## Series: e.ts.NO2
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1
##       1.3961  -0.4298  -0.8332
## s.e.  0.0906   0.0754   0.0697
##
## sigma^2 estimated as 1098:  log likelihood=-1922.3
## AIC=3852.6   AICc=3852.71   BIC=3868.48
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.2042815 33.00289 25.72095 -4.749865 292.7781 0.9327885
```

```
##                      ACF1
## Training set 0.01805261
```

```r
auto2 = auto.arima(NO2.ts,approximation = FALSE)
summary(auto2)
```

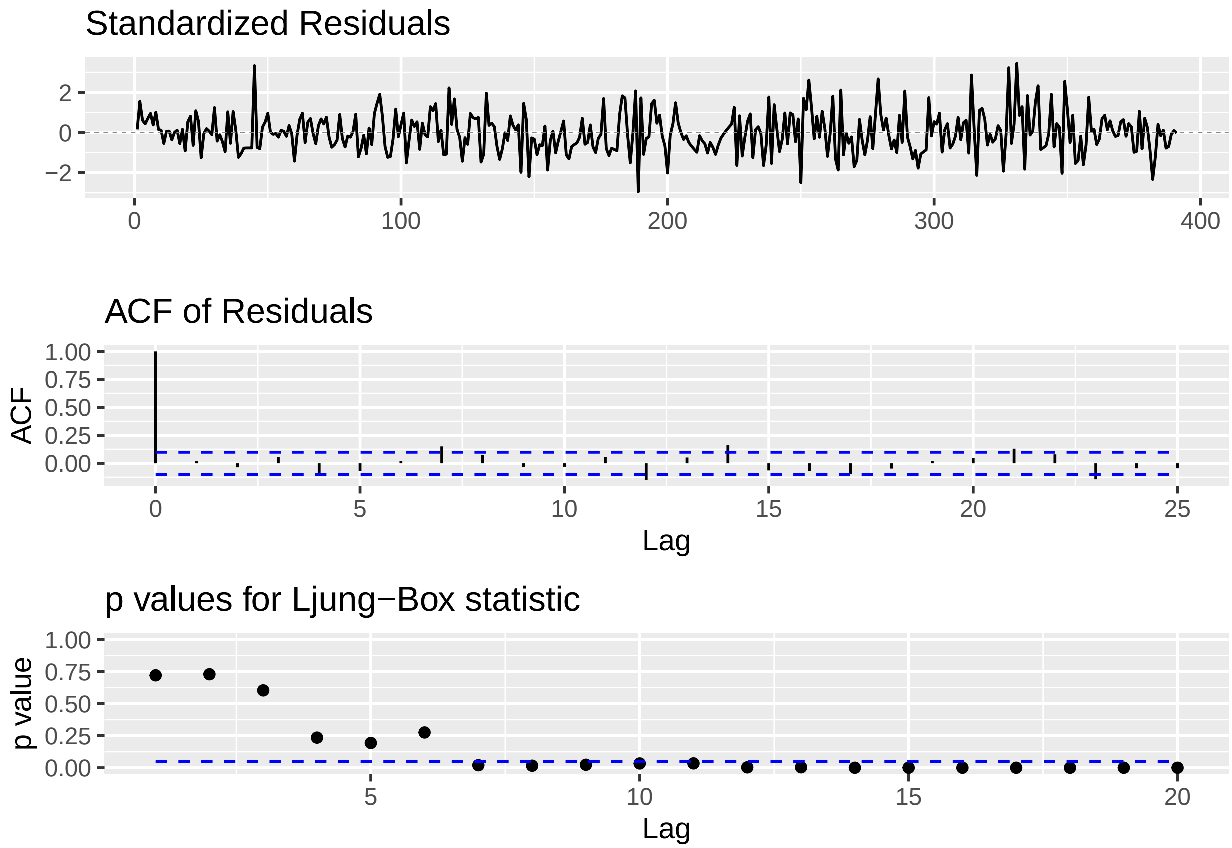
```
## Series: NO2.ts
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.4926  -0.9043
## s.e.  0.0660   0.0362
##
## sigma^2 estimated as 1118:  log likelihood=-1921.47
## AIC=3848.95   AICc=3849.01   BIC=3860.85
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
## Training set 0.6030949 33.30169 25.59834 -3.730363 16.86521 0.928342 0.01087294
```

```r
auto3 = auto.arima(diff(NO2.ts) , approximation = FALSE)
summary(auto3)
```

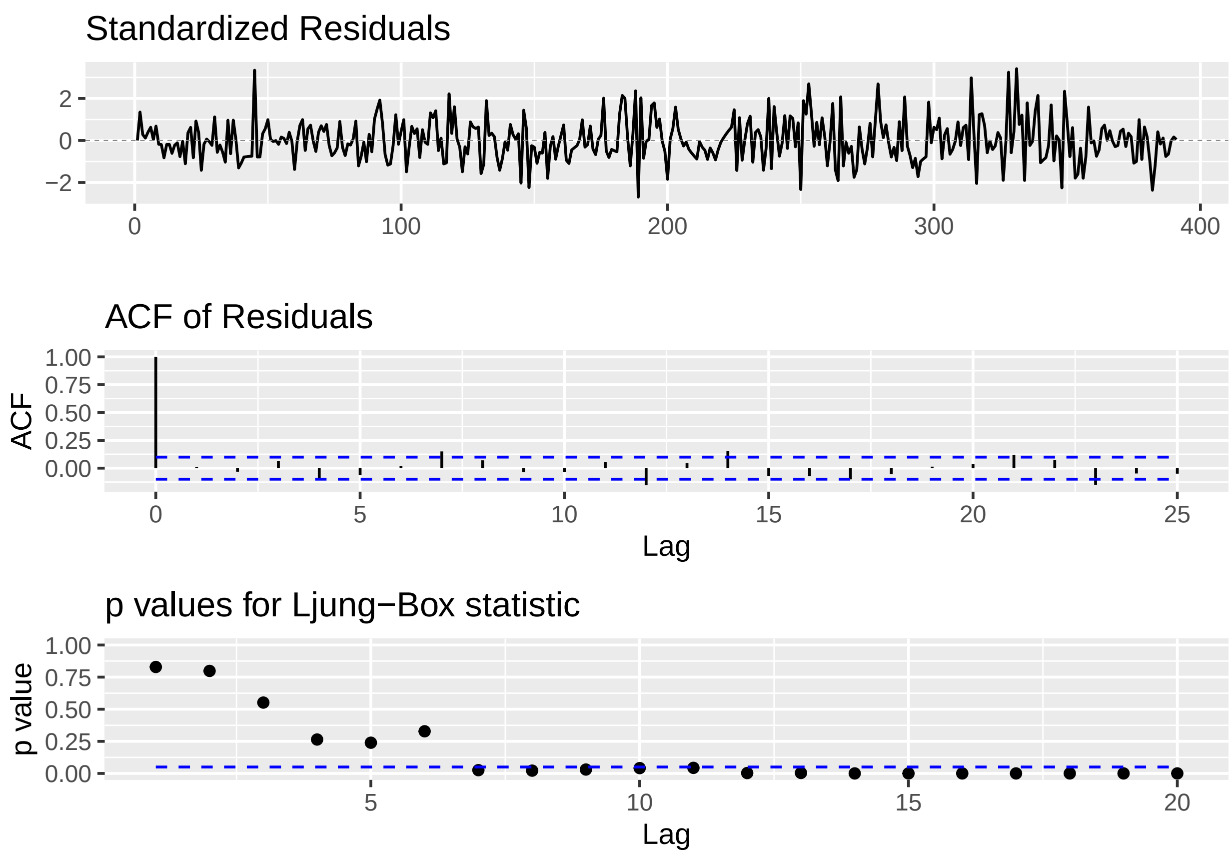
```
## Series: diff(NO2.ts)
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##          ar1      ma1
##       0.4926  -0.9043
## s.e.  0.0660   0.0362
##
## sigma^2 estimated as 1118:  log likelihood=-1921.47
## AIC=3848.95   AICc=3849.01   BIC=3860.85
##
## Training set error measures:
##                     ME     RMSE      MAE MPE MAPE      MASE       ACF1
## Training set 0.6043278 33.34436 25.66366 NaN  Inf 0.6022576 0.01092258
```

Takeaway: Auto.arima shows ARIMA(2,0,1) is best for the residual model, ARIMA(1,1,1) is best for regular model. And for fun, ARIMA(1,0,1) is best for first difference model. The models with the lowest AIC are ARIMA(1,1,1) and ARIMA(1,0,1), despite the residual model being the ideal stationary model.

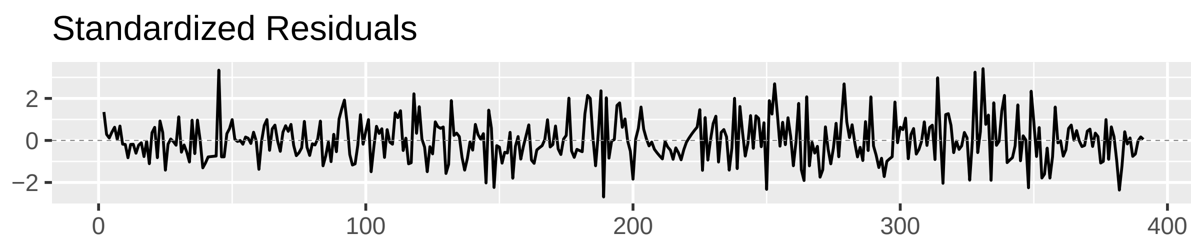
```r
ggtsdiag(auto,gof.lag=20)
```

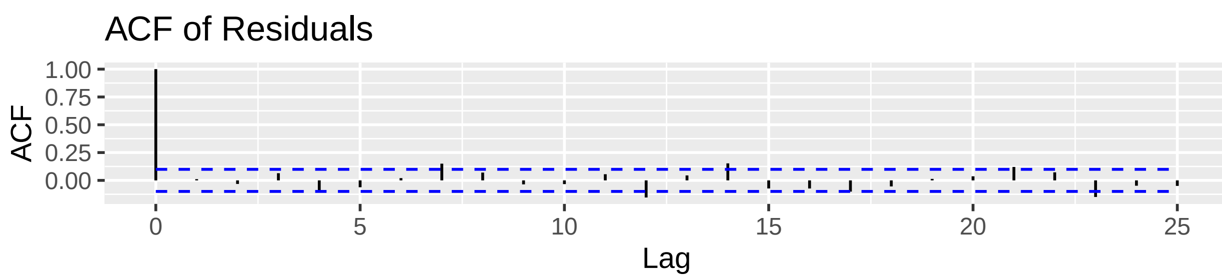


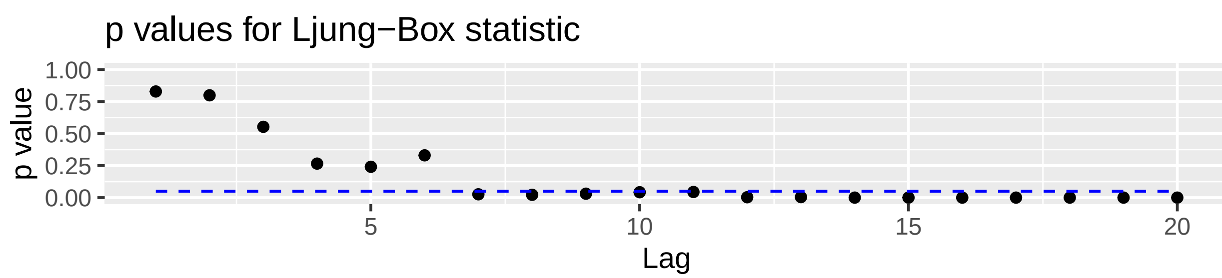

```
ggtsdiag(auto2,gof.lag=20)
```




```
ggtsdiag(auto3,gof.lag=20)
```







Takeaway: The dignostic plot is the same for each model and good for about 6 lags. At this point I am more confident in using the residual model despite the AIC is slightly higher, given what we know about the data and its non stationary nature, using the residual model accounts for the non stationary. This concludes part 1 of the analysis.

Part 2: Forecast and Prediction

```
#residual
next.365days <- c((length(e.ts.NO2)-364):(length(e.ts.NO2)))
next.365 <- data.frame(time.temp = next.365days, temp = e.ts.NO2[next.365days])
next.365.ts <- e.ts.NO2[next.365days]

time.temp <- c(1:(length(e.ts.NO2)-365))
NO2.lm <- lm(e.ts.NO2[time.temp]~time.temp)
summary(NO2.lm)
```

```
##
## Call:
## lm(formula = e.ts.NO2[time.temp] ~ time.temp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -48.687 -10.310  -3.907  15.189  36.926
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.4191     8.6606   6.630  7.4e-07 ***
## time.temp    -1.9719     0.5608  -3.516  0.00177 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.45 on 24 degrees of freedom
## Multiple R-squared:   0.34,  Adjusted R-squared:  0.3125
## F-statistic: 12.36 on 1 and 24 DF,  p-value: 0.00177
```

```r
E_Y.pred <- predict(NO2.lm, newdata=next.365)
e_t.pred <- forecast(auto, h=365)
next.365days.prediction <- E_Y.pred + e_t.pred$mean

mean((next.365days.prediction-next.365$temp)^2)
```

```
## [1] 171152.6
```

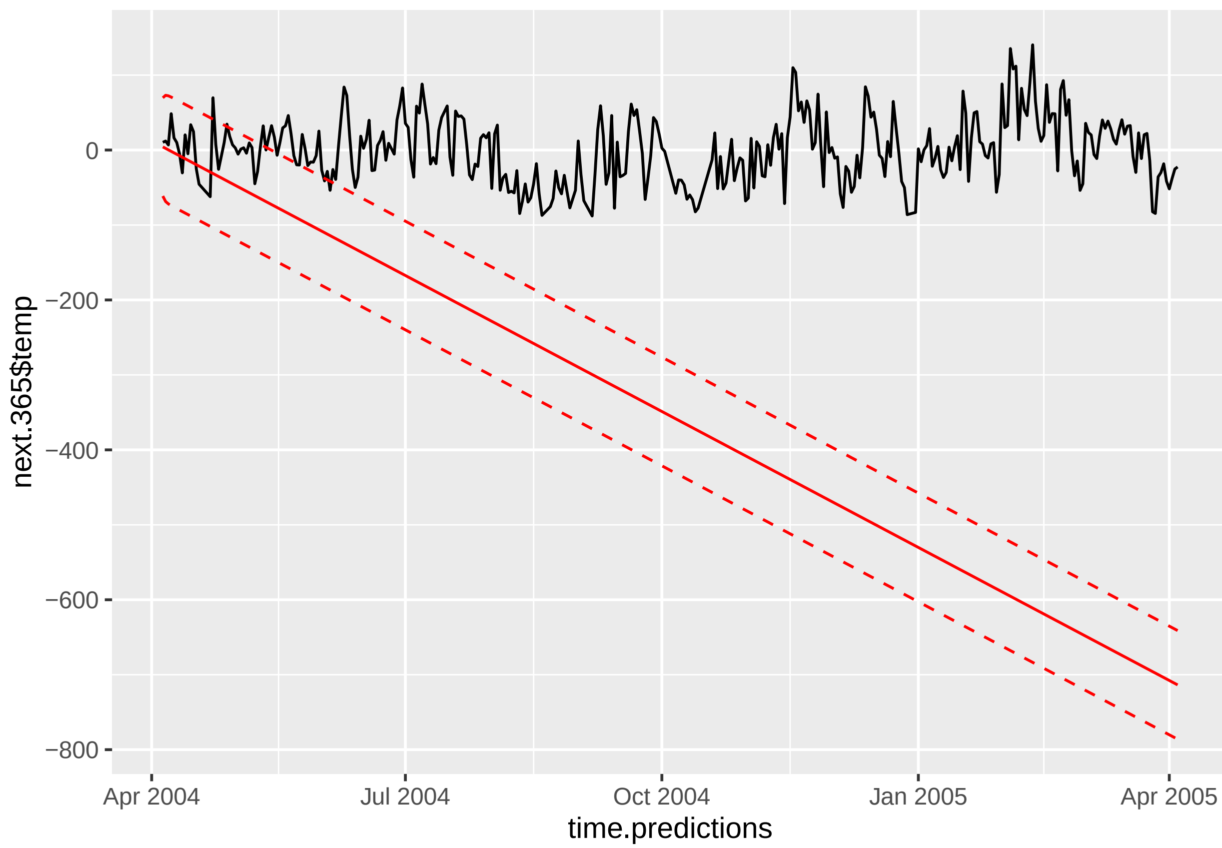
```r
#first diff
E_Y.pred <- predict(NO2.lm, newdata=next.365)
e_t.pred <- forecast(auto3, h=365)
next.365days.prediction <- E_Y.pred + e_t.pred$mean

mean((next.365days.prediction-next.365$temp)^2)
```

```
## [1] 171099.9
```

```r
time.predictions <- dailyAQ$time[(length(time.temp)+1) : (length(time.temp)+365)]


ggplot() + geom_line(aes(x=time.predictions,y=next.365$temp),color="black") +
  geom_line(aes(x=time.predictions,y=next.365days.prediction),color="red") +
  geom_line(aes(x=time.predictions,y=E_Y.pred + e_t.pred$lower[,2]),
            color="red",linetype="dashed") +
  geom_line(aes(x=time.predictions,y=E_Y.pred + e_t.pred$upper[,2]),
            color="red",linetype="dashed")
```



Takeaway: The MSE of the residual model is 171,152.6, a very high number which is given since we are predicting 365 points. Compared to the MSE of first difference model 171,099.9, slightly less. I think the problem with my graph

```
set.seed(5)
auto.sim <- arima.sim(n=365, list(ar=c(auto$coef[1],auto$coef[2]),
                                   ma=c(auto$coef[3])),
                        sd=sqrt(auto$sigma2))
next.year <- c(1:365)
next.yr <- data.frame(time.temp = next.year)

next.yr.pred <- predict(NO2.lm , newdata = next.yr)

autoplot(ts(next.yr.pred + auto.sim))
```


```r
mean(ts(next.yr.pred + auto.sim))
```

```
## [1] -302.7068
```

```r
var(ts(next.yr.pred + auto.sim))
```

```
## [1] 45027.53
```

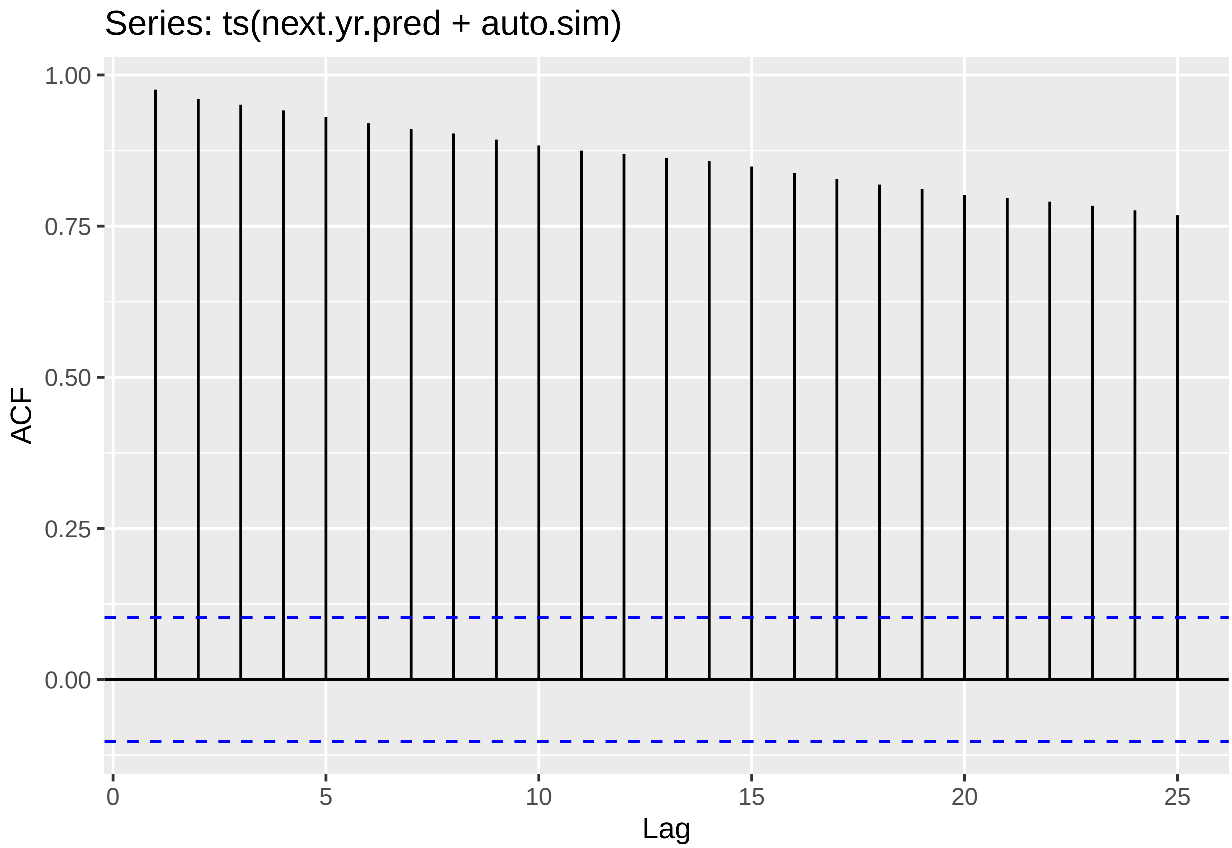
```r
mean(e.ts.NO2)
```

```
## [1] -2.441639e-15
```

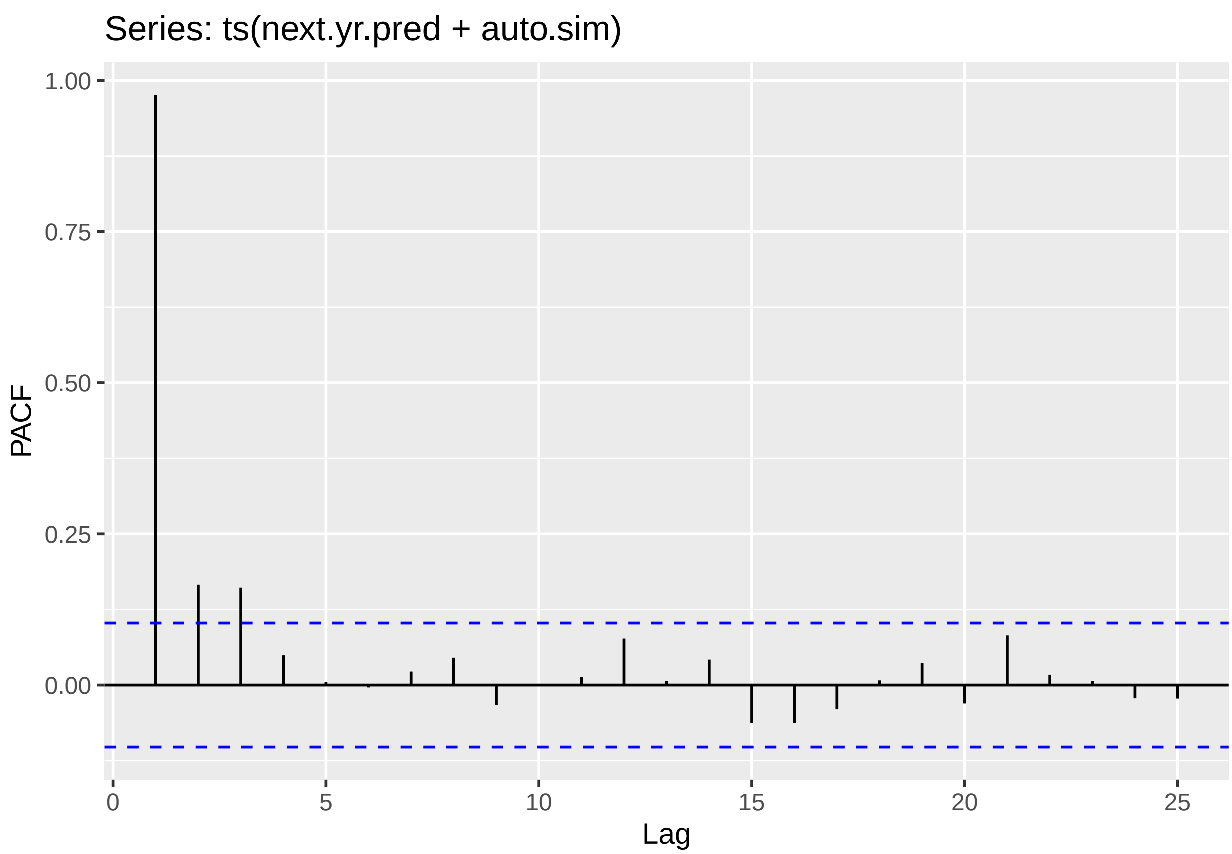
```r
var(e.ts.NO2)
```

```
## [1] 1930.46
```

```r
ggAcf(ts(next.yr.pred + auto.sim))
```




```
ggPacf(ts(next.yr.pred + auto.sim))
```




```r
next365.lm <- lm(ts(next.yr.pred + auto.sim)~next.year)
summary(next365.lm)
```

```
##
## Call:
## lm(formula = ts(next.yr.pred + auto.sim) ~ next.year)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -149.380  -27.352    0.809   29.611  120.476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 57.01595    4.71170    12.1   <2e-16 ***
## next.year   -1.96570    0.02231   -88.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.92 on 363 degrees of freedom
## Multiple R-squared:  0.9553, Adjusted R-squared:  0.9552
## F-statistic:  7761 on 1 and 363 DF,  p-value: < 2.2e-16
```

Takeaway: The model is significant and its prediction is the NO2 level will decrease. Oddly enough, the mean and variance of the prediction model is very different from the residual model and the ACF is linearly decaying.