Econ 104 Project 2

Ishaan Shah, Tuhsar Malhotra, Austin Gigi, Avanish Jay Arun

4/29/2022

UIDs -
Ishaan Shah - 105336308
Tushar Malhotra - 205332418
Austin Gigi - 605393626
Avanish Jay Arun - 505304876

```
pollution_df <- read.csv('pollution.csv')
pollution_df[,"date_time"] <- as.Date(pollution_df[,"date_time"])
```

Question 1

```
dim(pollution_df)
```

```
## [1] 2193    9
```

There are 2193 observations with multiple predictors such as Humidity, precipMM, pressure, tempC, windspeedKmph, location, as well as two response variables in CO_Conc and NO_Conc. The variables were measured from 2015 to 2021 in California.

Description of each variable:

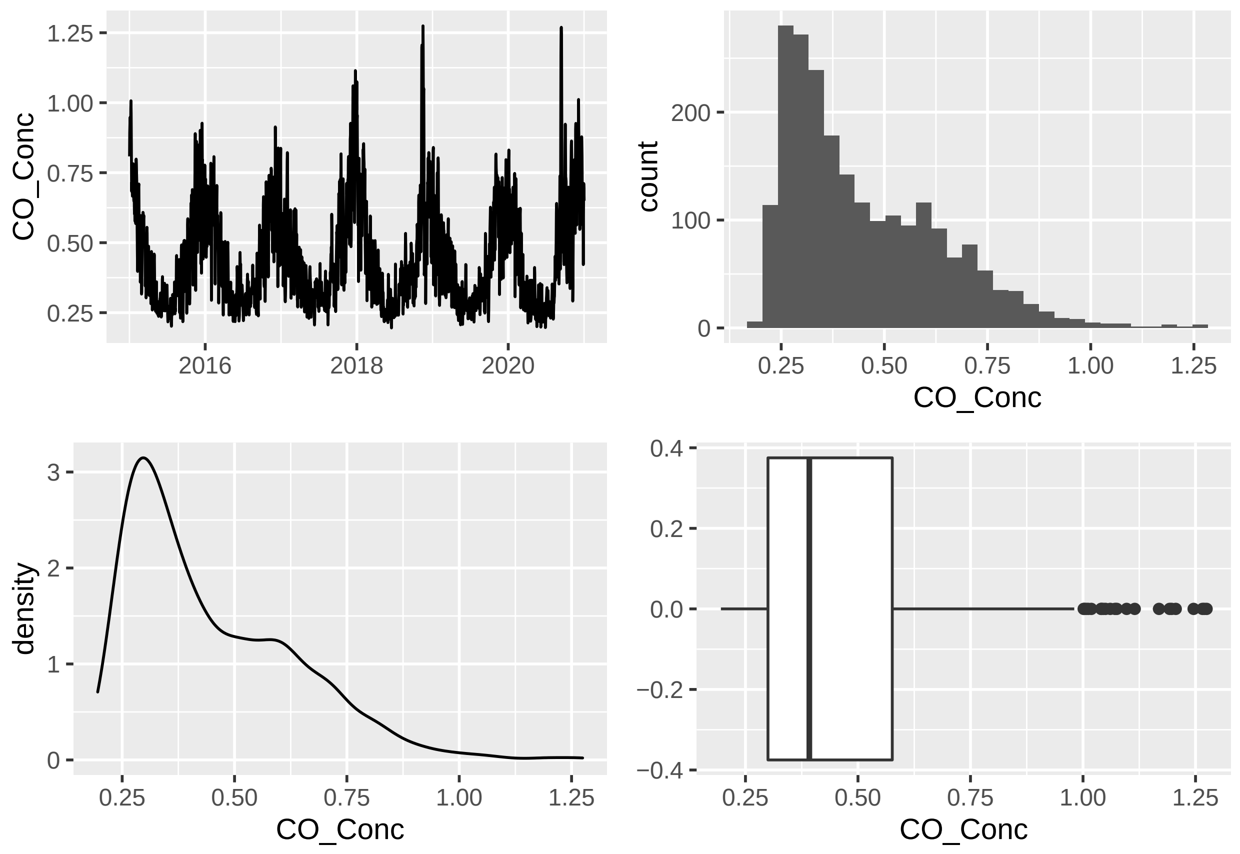
date_time - Date of observation
humidity - Humidity in California
precipMM - Precipitation in millimeters
pressure - Atmospheric pressure
tempC - Temperature in degrees Celsius
windspeedKmph - Windspeed in kilometres per hour
location - location of observation
CO_Conc - Carbon Monoxide concentration in California
NO2_Conc - Nitrogen Dioxide concentration in California
date - Date of Observation

```
library(ggplot2)
library(gridExtra)

# Carbon Monoxide
q1 <- ggplot(data = pollution_df, aes(x = date_time, y = CO_Conc)) + geom_line()+xlab("") + scale_x_date
q2 <- ggplot(data = pollution_df, aes(x = CO_Conc)) + geom_histogram()
```

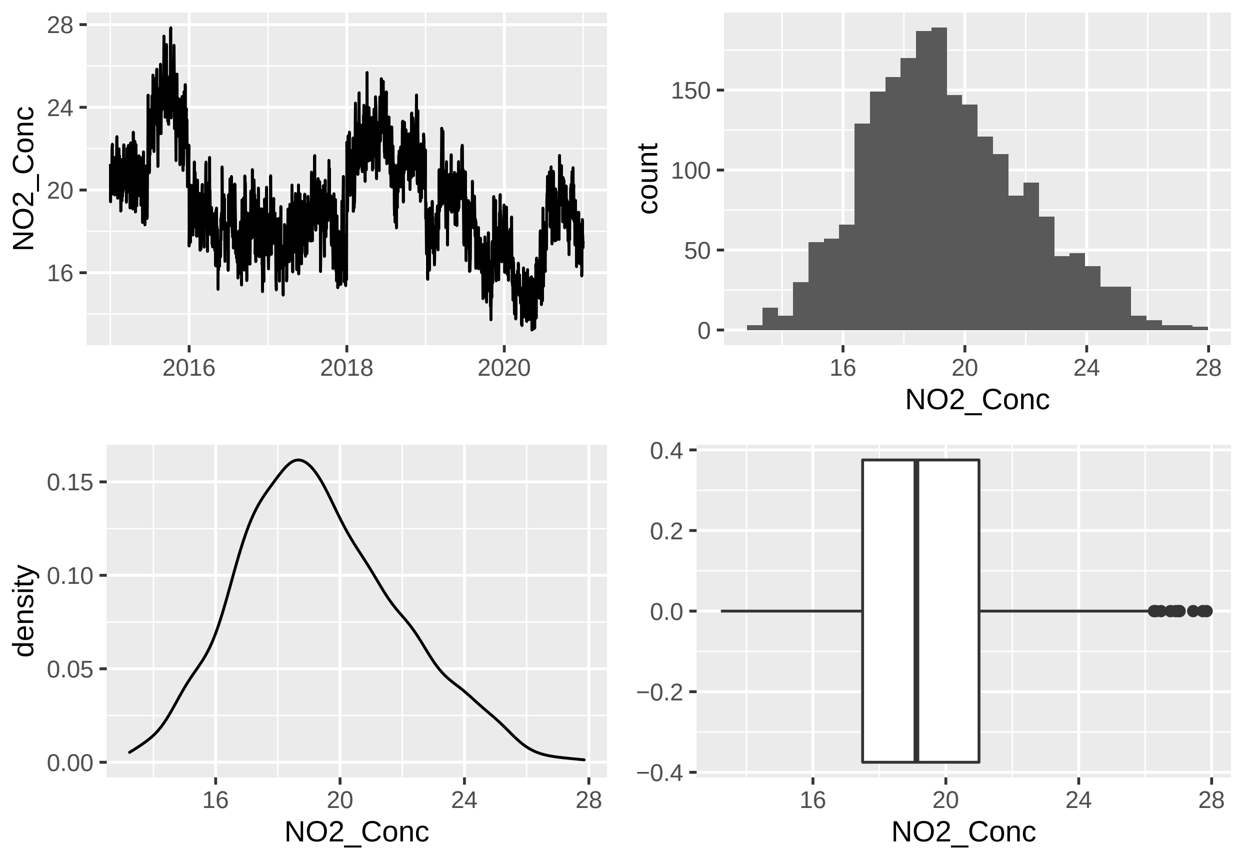
```r
q3 <- ggplot(data = pollution_df, aes(x = CO_Conc)) + geom_density()
q4 <- ggplot(data = pollution_df, aes(x = CO_Conc)) + geom_boxplot()
grid.arrange(q1, q2, q3, q4, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
# Nitrogen Dioxide
a1 <- ggplot(data = pollution_df, aes(x = date_time, y = NO2_Conc)) + geom_line()+ xlab("") + scale_x_da
a2 <- ggplot(data = pollution_df, aes(x = NO2_Conc)) + geom_histogram()
a3 <- ggplot(data = pollution_df, aes(x = NO2_Conc)) + geom_density()
a4 <- ggplot(data = pollution_df, aes(x = NO2_Conc)) + geom_boxplot()
grid.arrange(a1, a2, a3, a4, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Carbon Monoxide Distributions

i) Carbon monoxide levels have a cyclical trend with a slight dip in 2020 that could be attributed to COVID-19.

ii) The histogram above shows that the distribution for carbon monoxide levels are right-skewed.

iii) The density ploy for the Carbon monoxide concentration shows that the most occurring observations are between 0.25 and 0.50.

iv) The boxplot above shows that the median is 0.3919 and that the interquartile range is between [0.30, 0.5762].

Nitrogen Dioxide Distributions

i) The nitrogen dioxide levels follow an irregular pattern that at times is cyclical such as between 2018 and 2020.

ii) The histogram appears to follow a normal distribution given the bell-shaped curve. The most number of observations fall around the range of 18-20.

iii) The density for nitrogen dioxide concentration shows that the most occurring observations are between 18 - 20.

iv) The boxplot above shows that the median is 19.11 and that the interquartile range is between [17.50, 21.00].

```
summary(pollution_df$CO_Conc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.1955  0.3000  0.3919  0.4493  0.5762  1.2746
```

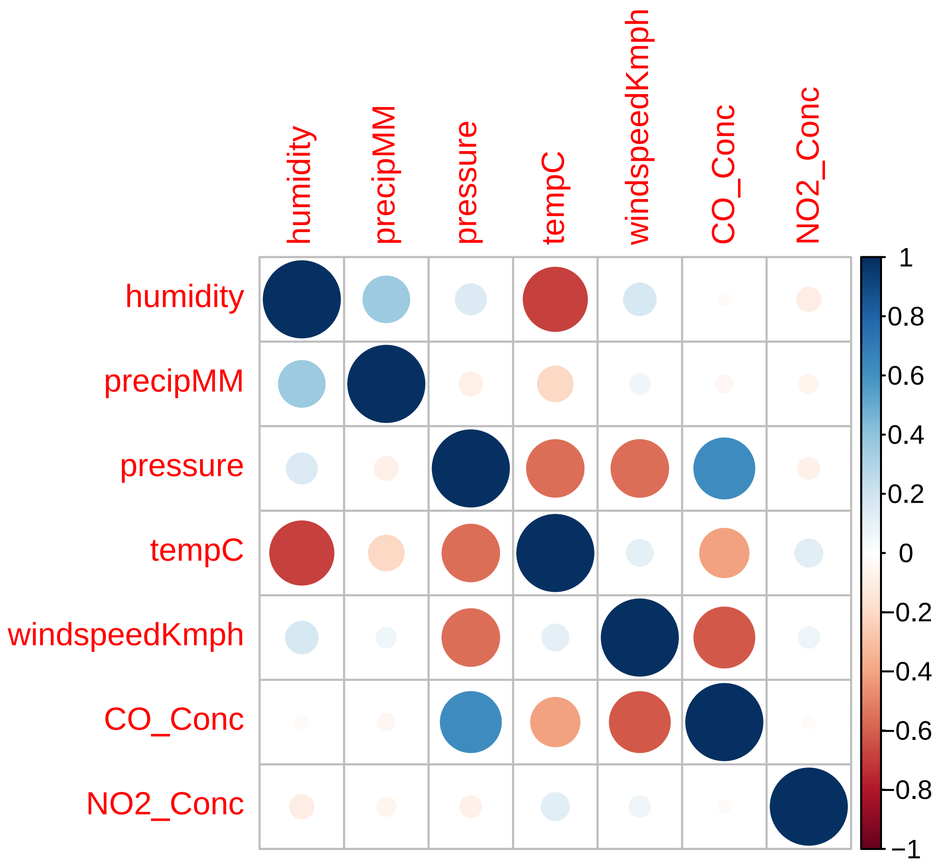
```
summary(pollution_df$NO2_Conc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   13.23   17.50   19.11   19.34   21.00   27.85
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
data_cor <- cor(pollution_df[,c(2,3,4,5,6,8,9)])
corrplot(data_cor)
```



The correlation plot above shows the correlation between each individual variable in the dataset. We are looking for a correlation between the exogenous variables and the concentration levels. The most correlated variables for carbon monoxide concentration from the correlation plot appear to be pressure which has a positive correlation, alongside windspeedKmph and tempC which have a negative correlation. The nitrogen dioxide concentration doesn't appear to be correlated with any other variable in the dataset. However, from the correlation plot we can see a lot of these variables are highly correlated with each other as they measure similar statistics. We will need to take this into account later to avoid multicollinearity.

Question 2

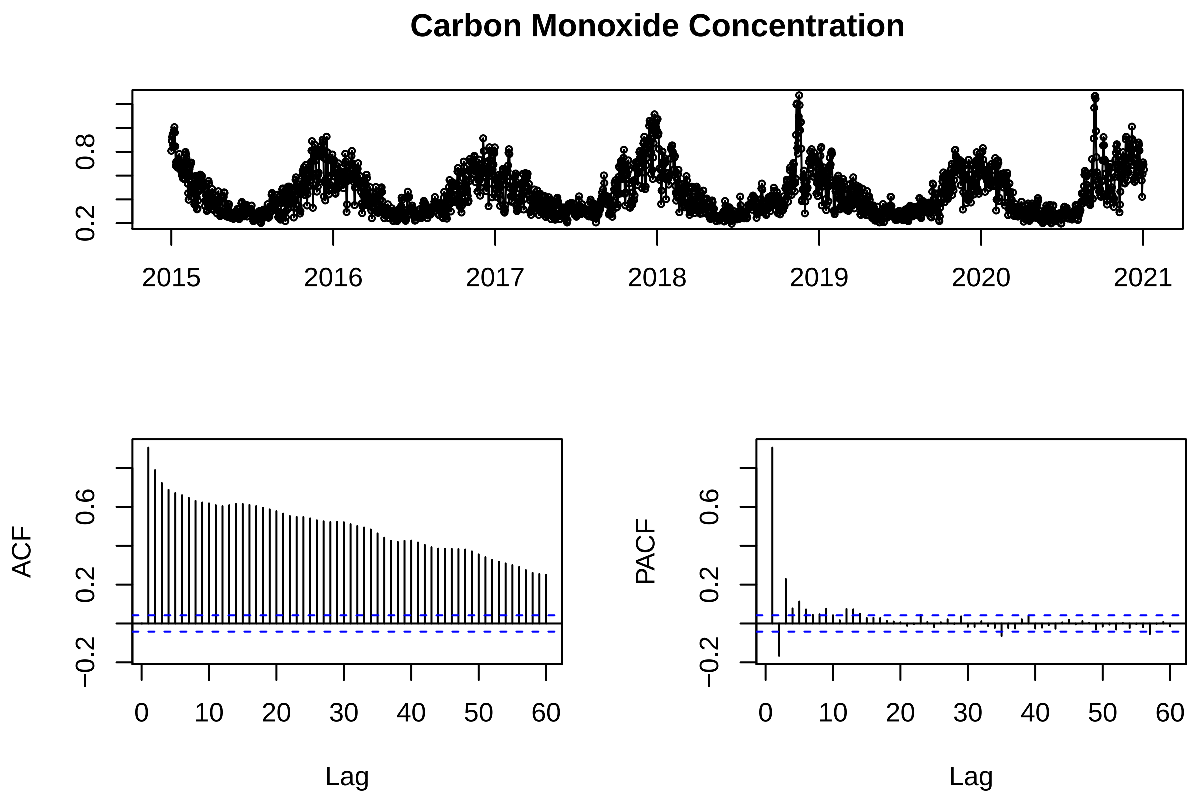
```r
CO <- ts(data = pollution_df$CO_Conc, start = c(2015,1), frequency = 365)
NO2 <- ts(data = pollution_df$NO2_Conc, start = c(2015,1), frequency = 365)
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5
```

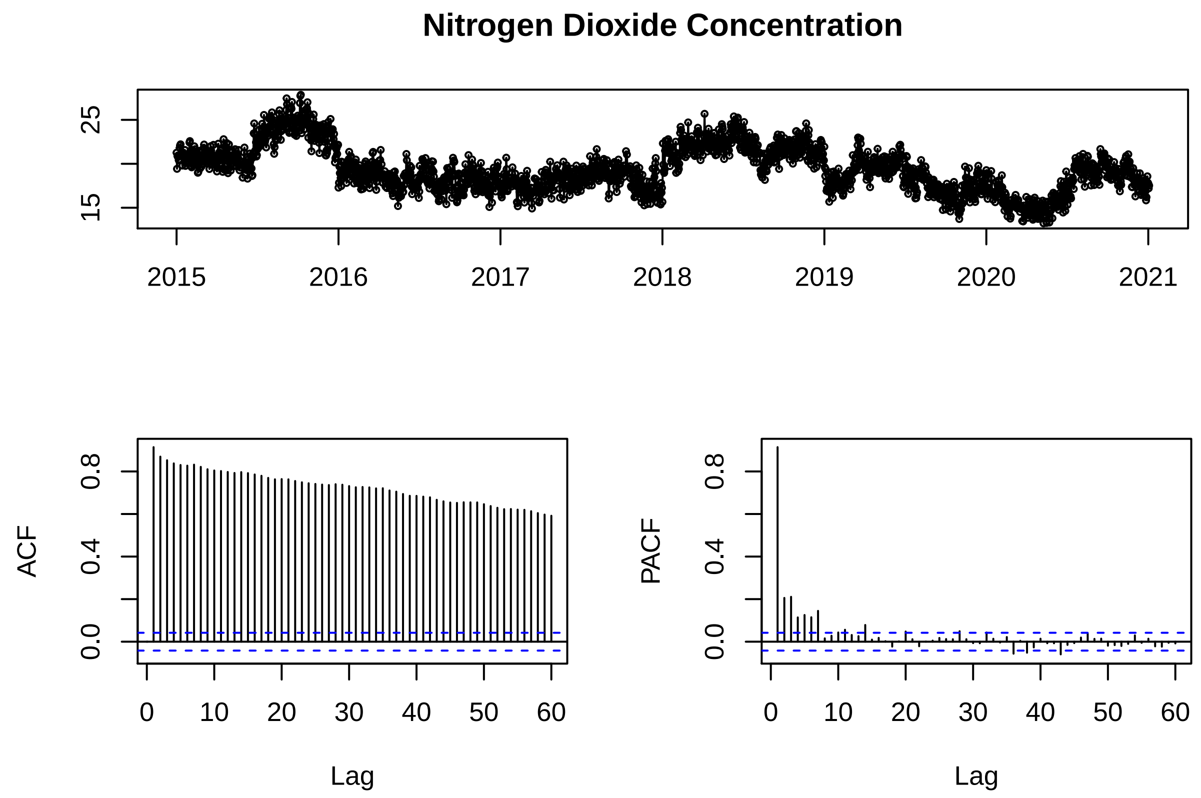
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
tsdisplay(CO, main = "Carbon Monoxide Concentration", lag.max = 60)
```



It is tough to judge stationarity from just the plot but Carbon Monoxide does look to be stationary as the mean or variance doesn't change much with respect to time. The ACF plot shows that every lag is significant, however the PACF plot shows a diffrent story. It looks like the 1st lag adds the most information, with subsequent lags till about the 15th lag adding more information as they are significant.

```r
tsdisplay(NO2, main = "Nitrogen Dioxide Concentration", lag.max = 60)
```



NO2 doesn't look stationary as it looks like the variance changes with respect to time, violating the stationarity laws. The ACF plot shows that every lag is significant, however the PACF plot shows a diffrent story. It looks like the 1st lag adds the most information, with subsequent lags till about the 9th lag adding more information as they are significant.

Question 3

Carbon Monoxide model, AR(3) and AR(5)

```
library(dynlm)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
co_model1 <- dynlm(CO ~ L(CO,1:3))
summary(co_model1)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(4), End = 2021(3)
##
## Call:
## dynlm(formula = CO ~ L(CO, 1:3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44424 -0.03859 -0.00330  0.04157  0.38613
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.038283   0.004432   8.637   <2e-16 ***
## L(CO, 1:3)1  1.091164   0.020797  52.466   <2e-16 ***
## L(CO, 1:3)2 -0.409196   0.030009 -13.636   <2e-16 ***
## L(CO, 1:3)3  0.232469   0.020778  11.188   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07558 on 2186 degrees of freedom
## Multiple R-squared:  0.8341, Adjusted R-squared:  0.8339
## F-statistic:  3665 on 3 and 2186 DF,  p-value: < 2.2e-16
```

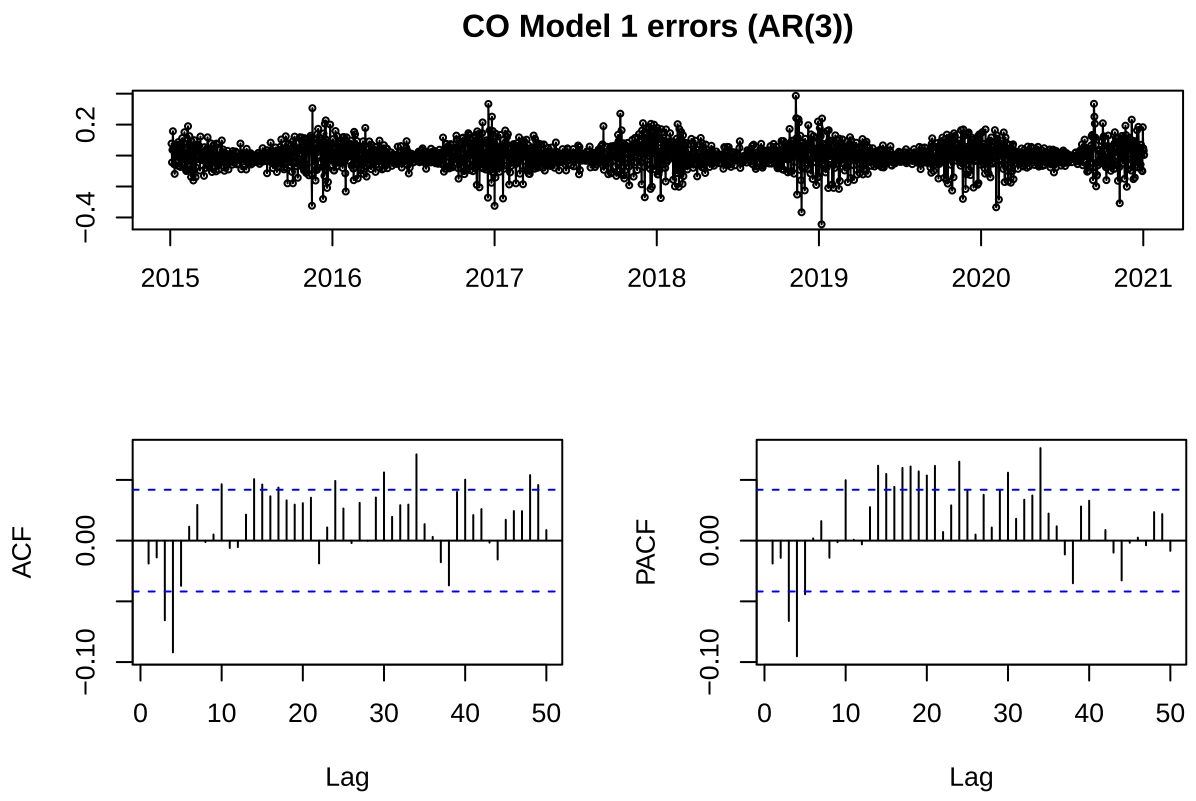
```
co_model2 <- dynlm(CO ~ L(CO,1:5))
summary(co_model2)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(6), End = 2021(3)
##
## Call:
## dynlm(formula = CO ~ L(CO, 1:5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41870 -0.03808 -0.00316  0.03989  0.36518
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.031500   0.004531   6.952 4.73e-12 ***
## L(CO, 1:5)1  1.063557   0.021270  50.003  < 2e-16 ***
## L(CO, 1:5)2 -0.393491   0.031137 -12.637  < 2e-16 ***
## L(CO, 1:5)3  0.188522   0.032003   5.891 4.44e-09 ***
## L(CO, 1:5)4 -0.041958   0.031148  -1.347    0.178
## L(CO, 1:5)5  0.112818   0.021267   5.305 1.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0749 on 2182 degrees of freedom
## Multiple R-squared:  0.8365, Adjusted R-squared:  0.8361
## F-statistic:  2232 on 5 and 2182 DF,  p-value: < 2.2e-16
```

We chose an AR(3) and AR(5) model based on the spikes seen in the PACF plots for Carbon Monoxide in question 2.

Comparing Residuals

```r
tsdisplay(co_model1$residuals, main =  "CO Model 1 errors (AR(3))", lag.max = 50)
```

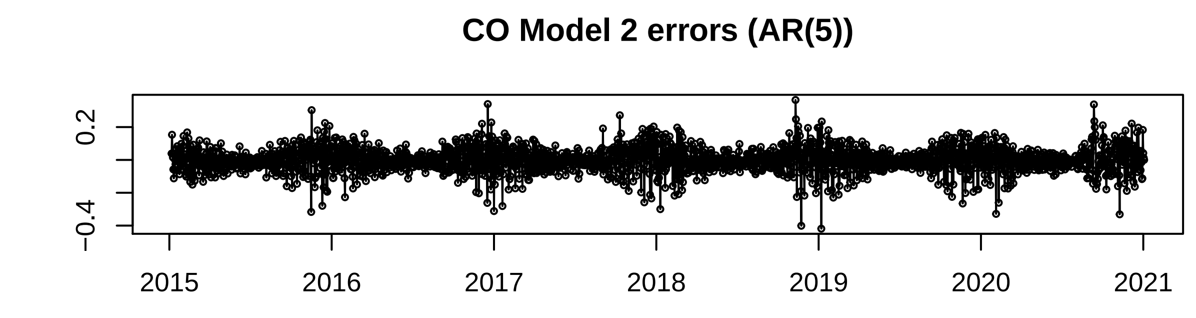


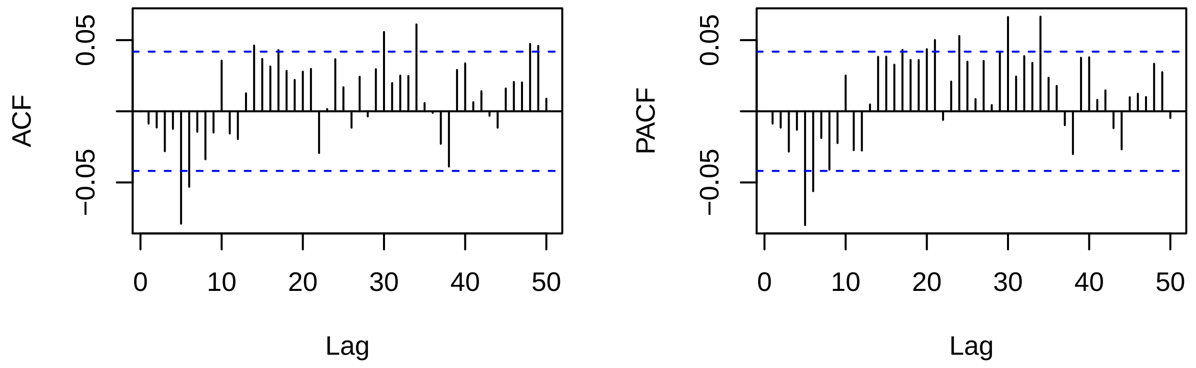
Looking at the ACF and PACF plots above for the AR(3) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the 35th lag, showing that there is high order serial correlation. This is confirmed by the low p-value of the test below.This implies that this model may not be the best as there are dynamics that are unaccounted for.

```r
library(lmtest)
bgtest(co_model1, order = 35)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 35
##
## data:  co_model1
## LM test = 150.66, df = 35, p-value = 2.672e-16
```

```r
tsdisplay(co_model2$residuals, main =  "CO Model 2 errors (AR(5))", lag.max = 50)
```





Looking at the ACF and PACF plots above for the AR(5) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the 35th lag, showing that there is high order serial correlation. This is confirmed by the low p-value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```
bgtest(co_model2, order = 35)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 35
##
## data:  co_model2
## LM test = 111.66, df = 35, p-value = 6.186e-10
```

Training MSE, Testing MSE, AIC

```
train_CO <- pollution_df[(1:((2/3)*nrow(pollution_df))),] #2/3rd split
test_CO <- pollution_df[-(1:((2/3)*nrow(pollution_df))),]
```

```
CO_train1 <- ts(data = train_CO$CO_Conc, frequency = 365)
CO_test1 <- ts(data = test_CO$CO_Conc, frequency = 365)
```

```
#Training model construction for AR(3)
co_train_model1 <- dynlm(CO_train1 ~ L(CO_train1,1:3))
co_train_model2 <- ar(CO_train1, FALSE, 3)
```

```
#used a different library for the same model to get OOS predictions
#Calculating OOS predictions
num_predictions <- dim(test_CO)[1]
test_preds <- forecast(co_train_model2, num_predictions )
#MSE calculations
cat("The Training MSE for AR(3) is:",
    sqrt(mean((train_CO[-(1:3),]$CO_Conc - predict(co_train_model1)) ^ 2)),
    ", while the Testing MSE for AR(3) is:",
    sqrt(mean((test_CO$CO_Conc - test_preds$mean) ^ 2)))
```

```
## The Training MSE for AR(3) is: 0.07436907 , while the Testing MSE for AR(3) is: 0.1865845
```

```
#Training model construction for AR(5)
co_train_model3 <- dynlm(CO_train1 ~ L(CO_train1,1:5))
co_train_model4 <- ar(CO_train1, FALSE, 5)
#Calculating OOS predcitions
num_predictions <- dim(test_CO)[1]
#MSE calculations
test_preds <- forecast(co_train_model4, num_predictions )
cat("The Training MSE for AR(5) is:",
    sqrt(mean((train_CO[-(1:5),]$CO_Conc - predict(co_train_model3)) ^ 2)),
    ", while the Testing MSE for AR(5) is:",
    sqrt(mean((test_CO$CO_Conc - test_preds$mean) ^ 2)))
```

```
## The Training MSE for AR(5) is: 0.07353411 , while the Testing MSE for AR(5) is: 0.186393
```

```
cat("The AIC for the AR(3) model is:", AIC(co_model1),
    ", while the AIC for the AR(5) model is:", AIC(co_model2))
```

```
## The AIC for the AR(3) model is: -5090.899 , while the AIC for the AR(5) model is: -5123.472
```

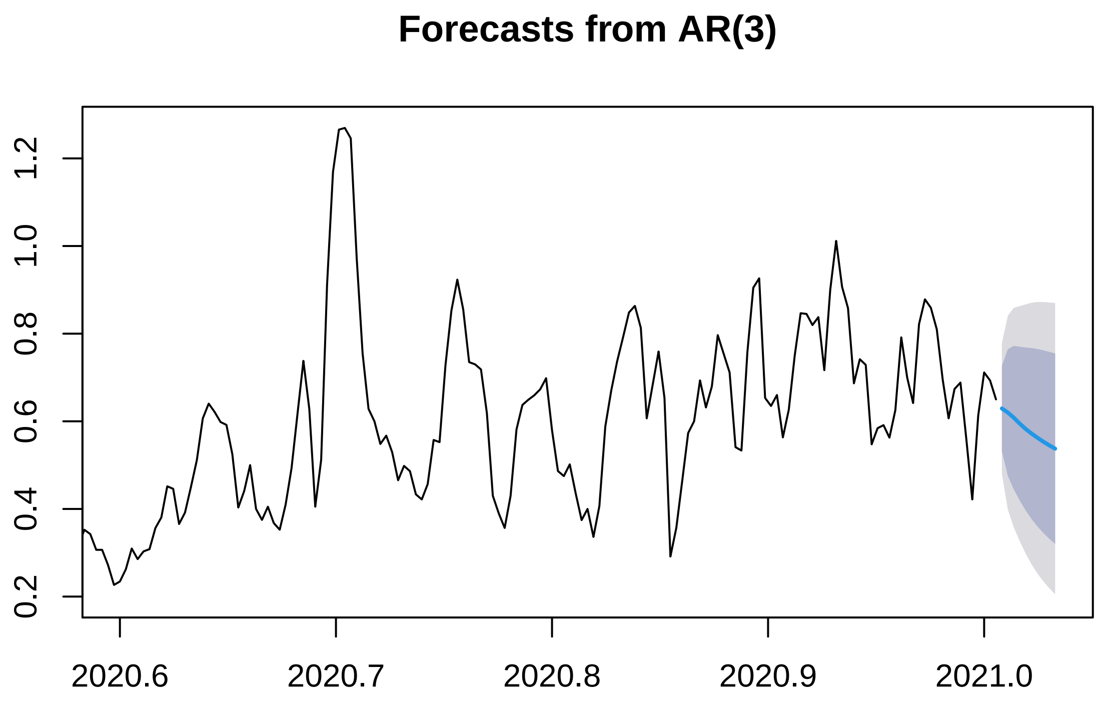
Both models look to be very close to each other as there is not much difference in the training MSE, testing MSE, and AIC values. However, looking closer into the numbers the AR(5) model has a lower AIC value and a slightly better training MSE and testing MSE. Hence Model 2 which is an AR(5) model for Carbon Monoxide is the better model from the two.

Forecasting

```
co_ar_1 <-ar(CO, FALSE, 3)
forecast(co_ar_1, 10)
```

```
##           Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021.0082      0.6292613 0.5317548 0.7267678 0.4801380 0.7783846
## 2021.0110      0.6200151 0.4755681 0.7644622 0.3991025 0.8409278
## 2021.0137      0.6084791 0.4448864 0.7720719 0.3582856 0.8586727
## 2021.0164      0.5948974 0.4196536 0.7701413 0.3268851 0.8629098
## 2021.0192      0.5826364 0.3968291 0.7684436 0.2984687 0.8668041
## 2021.0219      0.5721283 0.3770373 0.7672194 0.2737623 0.8704943
## 2021.0247      0.5625316 0.3600490 0.7650142 0.2528612 0.8722020
## 2021.0274      0.5535191 0.3451562 0.7618820 0.2348555 0.8721827
## 2021.0301      0.5451746 0.3319570 0.7583923 0.2190864 0.8712629
## 2021.0329      0.5375306 0.3202500 0.7548112 0.2052286 0.8698326
```

```
plot(forecast(co_ar_1, 10), xlim = c(2020.6,2021.033))
```

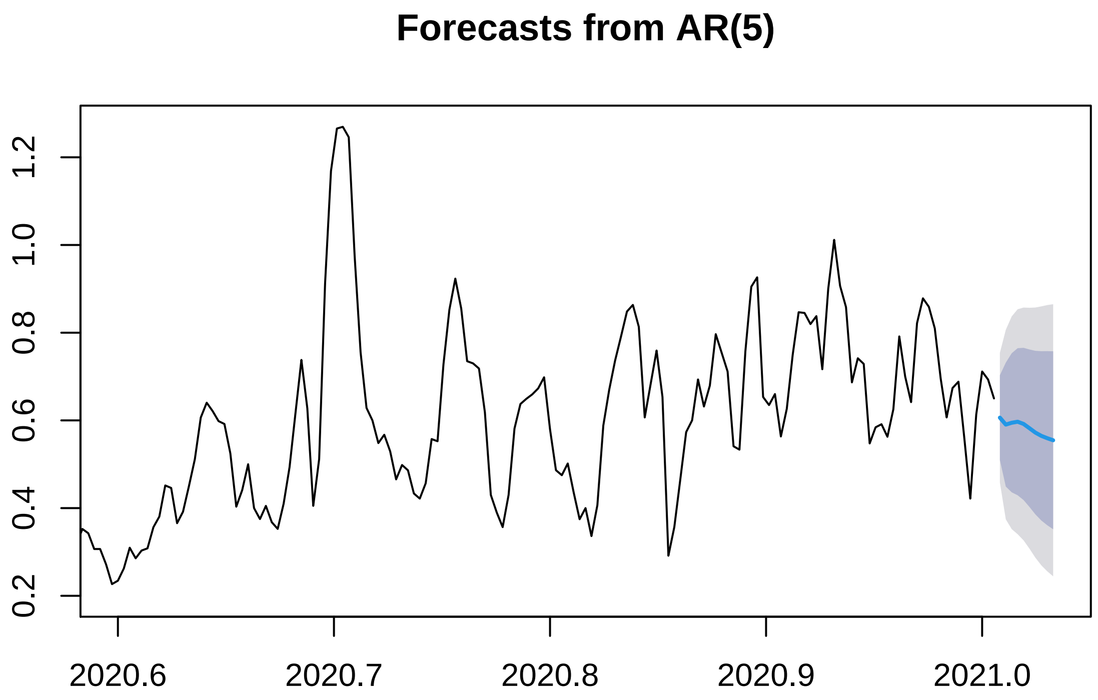


```
#Reduced x scale for better visibility of forecasts
```

```
co_ar_2 <-ar(CO, FALSE, 5)
forecast(co_ar_2, 10)
```

```
##            Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021.0082      0.6062434 0.5096107 0.7028762 0.4584564 0.7540304
## 2021.0110      0.5904323 0.4491577 0.7317070 0.3743714 0.8064932
## 2021.0137      0.5945982 0.4360066 0.7531897 0.3520533 0.8371430
## 2021.0164      0.5969316 0.4292469 0.7646163 0.3404800 0.8533832
## 2021.0192      0.5918857 0.4183240 0.7654474 0.3264460 0.8573255
## 2021.0219      0.5821123 0.4025004 0.7617241 0.3074196 0.8568049
## 2021.0247      0.5721271 0.3855844 0.7586698 0.2868347 0.8574196
## 2021.0274      0.5647356 0.3716844 0.7577868 0.2694893 0.8599819
## 2021.0301      0.5594313 0.3609595 0.7579031 0.2558948 0.8629677
## 2021.0329      0.5546713 0.3517905 0.7575521 0.2443919 0.8649507
```

```r
plot(forecast(co_ar_2, 10), xlim = c(2020.6,2021.033))
```



```r
#Reduced x scale for better visibility of forecasts
```

Predictions/ Forecast look good for both graphs as it is following the general pattern of Carbon Monoxide concentration in California.

Nitrogen Dioxide model, AR(7) and AR(10)

```r
no2_model1 <- dynlm(NO2 ~ L(NO2,1:7))
summary(no2_model1)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(8), End = 2021(3)
##
## Call:
## dynlm(formula = NO2 ~ L(NO2, 1:7))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3860 -0.6555  0.0083  0.6339  6.2317
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.59760    0.16624   3.595 0.000332 ***
## L(NO2, 1:7)1  0.61399    0.02119  28.970  < 2e-16 ***
## L(NO2, 1:7)2  0.01664    0.02494   0.667 0.504714
## L(NO2, 1:7)3  0.10950    0.02492   4.394 1.17e-05 ***
## L(NO2, 1:7)4  0.01101    0.02503   0.440 0.660104
## L(NO2, 1:7)5  0.04780    0.02492   1.918 0.055267 .
## L(NO2, 1:7)6  0.02279    0.02492   0.914 0.360640
## L(NO2, 1:7)7  0.14720    0.02119   6.946 4.96e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9626 on 2178 degrees of freedom
## Multiple R-squared:  0.8599, Adjusted R-squared:  0.8594
## F-statistic:  1909 on 7 and 2178 DF,  p-value: < 2.2e-16
```

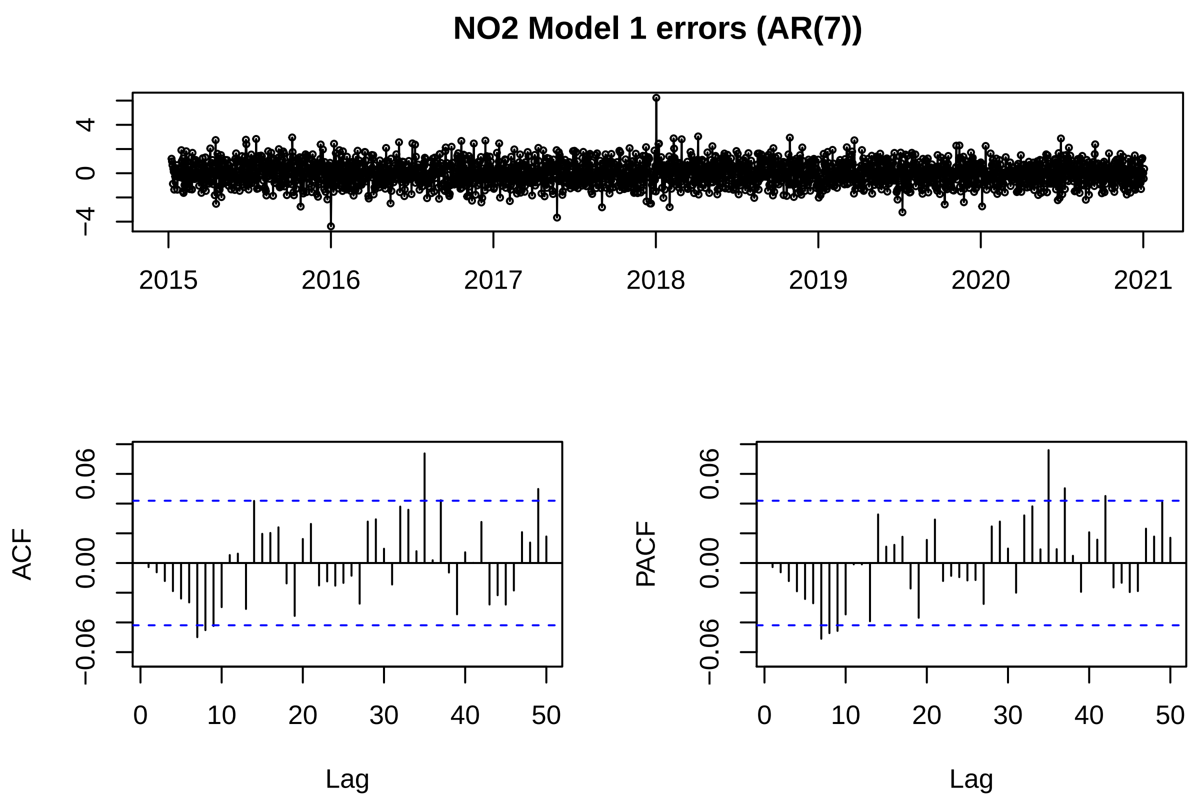
```
no2_model2 <- dynlm(NO2 ~ L(NO2,1:10))
summary(no2_model2)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(11), End = 2021(3)
##
## Call:
## dynlm(formula = NO2 ~ L(NO2, 1:10))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4887 -0.6546 -0.0001  0.6285  6.2374
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5469640  0.1676055   3.263  0.00112 **
## L(NO2, 1:10)1  0.6084495  0.0214338  28.387  < 2e-16 ***
## L(NO2, 1:10)2  0.0125411  0.0250974   0.500  0.61734
## L(NO2, 1:10)3  0.1019919  0.0250976   4.064 5.00e-05 ***
## L(NO2, 1:10)4  0.0086900  0.0250343   0.347  0.72853
## L(NO2, 1:10)5  0.0427644  0.0250375   1.708  0.08778 .
## L(NO2, 1:10)6  0.0196895  0.0250372   0.786  0.43171
## L(NO2, 1:10)7  0.1302416  0.0250437   5.201 2.17e-07 ***
## L(NO2, 1:10)8  0.0001048  0.0251075   0.004  0.99667
## L(NO2, 1:10)9  0.0021746  0.0250811   0.087  0.93092
## L(NO2, 1:10)10 0.0448107  0.0214290   2.091  0.03663 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9617 on 2172 degrees of freedom
## Multiple R-squared:  0.8603, Adjusted R-squared:  0.8597
## F-statistic:  1338 on 10 and 2172 DF,  p-value: < 2.2e-16
```

We chose an AR(7) and AR(10) model based on the spikes seen in the PACF plots for Nitrogen Dioxide in question 2.

Comparing Residuals

```
tsdisplay(no2_model1$residuals, main =  "NO2 Model 1 errors (AR(7))", lag.max = 50)
```

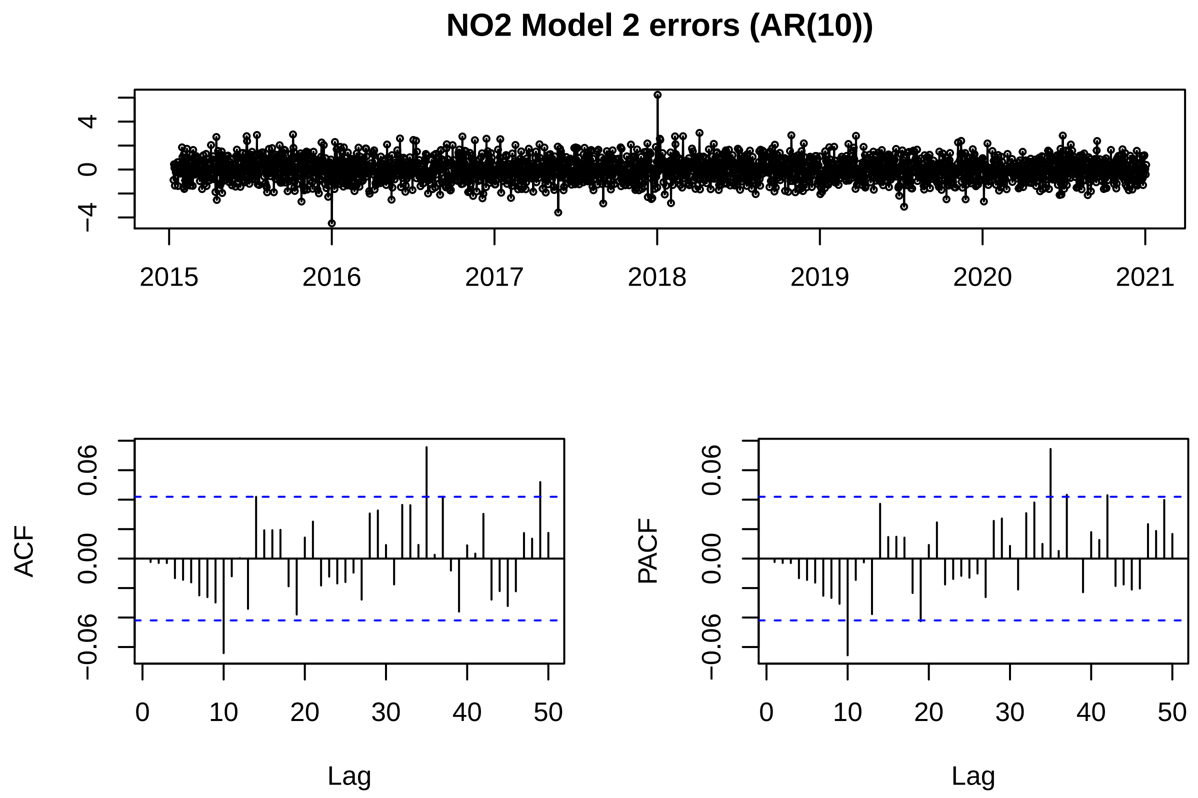


Looking at the ACF and PACF plots above for the AR(7) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the 35th lag which is highly significant, showing that there is high order serial correlation. Even though the spikes for the initial values are low, they pickup as the number of lags increase. This is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```
bgtest(no2_model1, order =35)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 35
##
## data:  no2_model1
## LM test = 68.632, df = 35, p-value = 0.0005833
```

```
tsdisplay(no2_model2$residuals, main =  "NO2 Model 2 errors (AR(10))", lag.max = 50)
```



Looking at the ACF and PACF plots above for the AR(10) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to above the 45th lag, where the 35th lag is highly significant along with the 10th lag, showing that there is high order serial correlation. Even though the spikes for the initial values are low, they pickup as the number of lags increase. This is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```
bgtest(no2_model2, order =45)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 45
##
## data:  no2_model2
## LM test = 72.973, df = 45, p-value = 0.005219
```

Training MSE, Testing MSE, AIC

```
train_no2 <- pollution_df[(1:((2/3)*nrow(pollution_df))),] #66.67% split
test_no2 <- pollution_df[-(1:((2/3)*nrow(pollution_df))),]
```

```
NO2_train1 <- ts(data = train_CO$NO2_Conc, frequency = 365)
NO2_test1 <- ts(data = test_CO$NO2_Conc, frequency = 365)
```

```r
# Constructing training model for AR(7)
no2_train_model1 <- dynlm(NO2_train1 ~ L(NO2_train1,1:7))
no2_train_model2 <- ar(NO2_train1, FALSE, 7)
# OOS Predictions
num_predictions <- dim(test_no2)[1]
test_preds <- forecast(no2_train_model2, num_predictions )
#MSE Calculations
cat("The Training MSE for AR(7) is:",
    sqrt(mean((train_no2[-(1:7),]$NO2_Conc - predict(no2_train_model1)) ^ 2)),
    ", while the Testing MSE for AR(7) is:",
    sqrt(mean((test_no2$NO2_Conc - test_preds$mean) ^ 2)))
```

```
## The Training MSE for AR(7) is: 0.9904101 , while the Testing MSE for AR(7) is: 3.08569
```

```r
# Constructing training model for AR(10)
no2_train_model3 <- dynlm(NO2_train1 ~ L(NO2_train1,1:10))
no2_train_model4 <- ar(NO2_train1, FALSE, 10)
#OOS Predicitions
num_predictions <- dim(test_no2)[1]
test_preds <- forecast(no2_train_model4, num_predictions )
#MSE Calculations
cat("The Training MSE for AR(10) is:",
    sqrt(mean((train_no2[-(1:10),]$NO2_Conc - predict(no2_train_model3)) ^ 2)),
    ", while the Testing MSE for AR(7) is:",
    sqrt(mean((test_no2$NO2_Conc - test_preds$mean) ^ 2)))
```

```
## The Training MSE for AR(10) is: 0.9884412 , while the Testing MSE for AR(7) is: 3.095976
```

```r
cat("The AIC for the AR(7) model is:", AIC(co_model1),
    ", while the AIC for the AR(10) model is:", AIC(co_model2))
```

```
## The AIC for the AR(7) model is: -5090.899 , while the AIC for the AR(10) model is: -5123.472
```

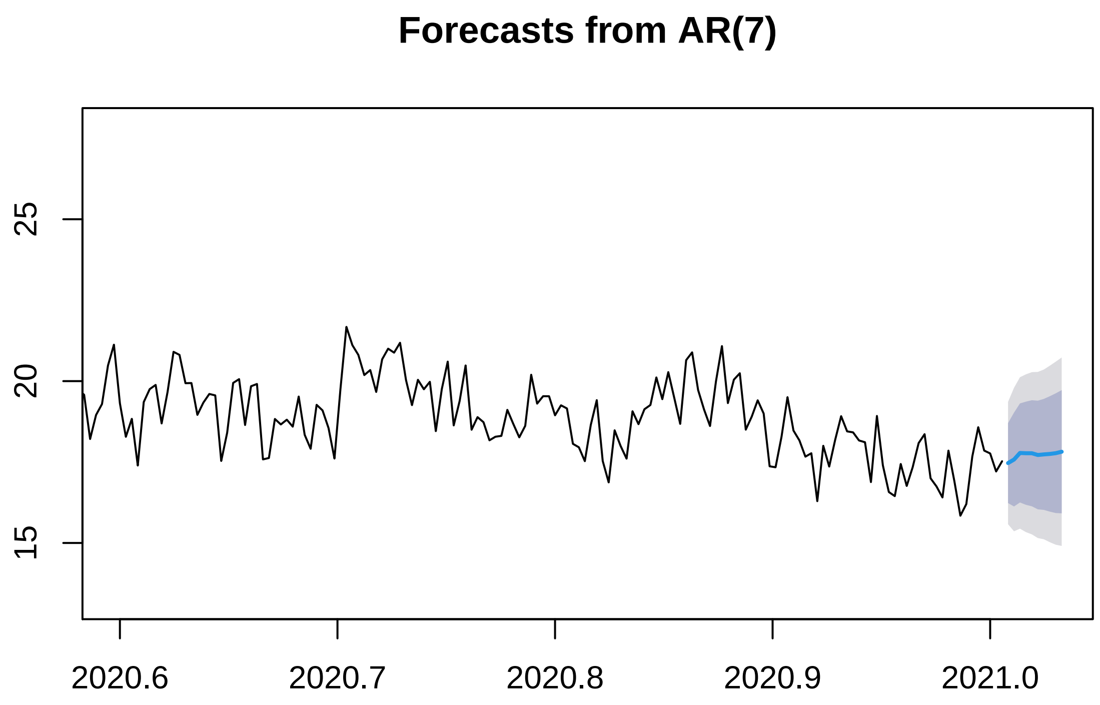
Both models look to be very close to each other as there is not much difference in the training MSE, testing MSE, and AIC values. However, looking closer into the numbers the AR(10) model has a lower AIC value and a slightly better training MSE but a slightly worse testing MSE by about 0.01. Despite this, we would still conclude model 2 which is the AR(10) model for Nitrogen Dioxide is the better model from the two.

Forecasting

```r
no2_ar_1 <-ar(NO2, FALSE, 7)
forecast(no2_ar_1, 10)
```

```
##           Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2021.0082       17.46797 16.23261 18.70333 15.57865 19.35729
## 2021.0110       17.57727 16.12801 19.02653 15.36081 19.79372
## 2021.0137       17.78084 16.25160 19.31009 15.44207 20.11962
## 2021.0164       17.77235 16.17894 19.36576 15.33544 20.20926
## 2021.0192       17.77123 16.13300 19.40946 15.26578 20.27669
## 2021.0219       17.71781 16.03995 19.39567 15.15174 20.28387
## 2021.0247       17.73621 16.02244 19.44999 15.11522 20.35720
## 2021.0274       17.75027 15.96850 19.53203 15.02529 20.47525
## 2021.0301       17.77366 15.92520 19.62211 14.94669 20.60062
## 2021.0329       17.81887 15.91584 19.72190 14.90844 20.72930
```

```
plot(forecast(no2_ar_1, 10), xlim = c(2020.6,2021.03))
```

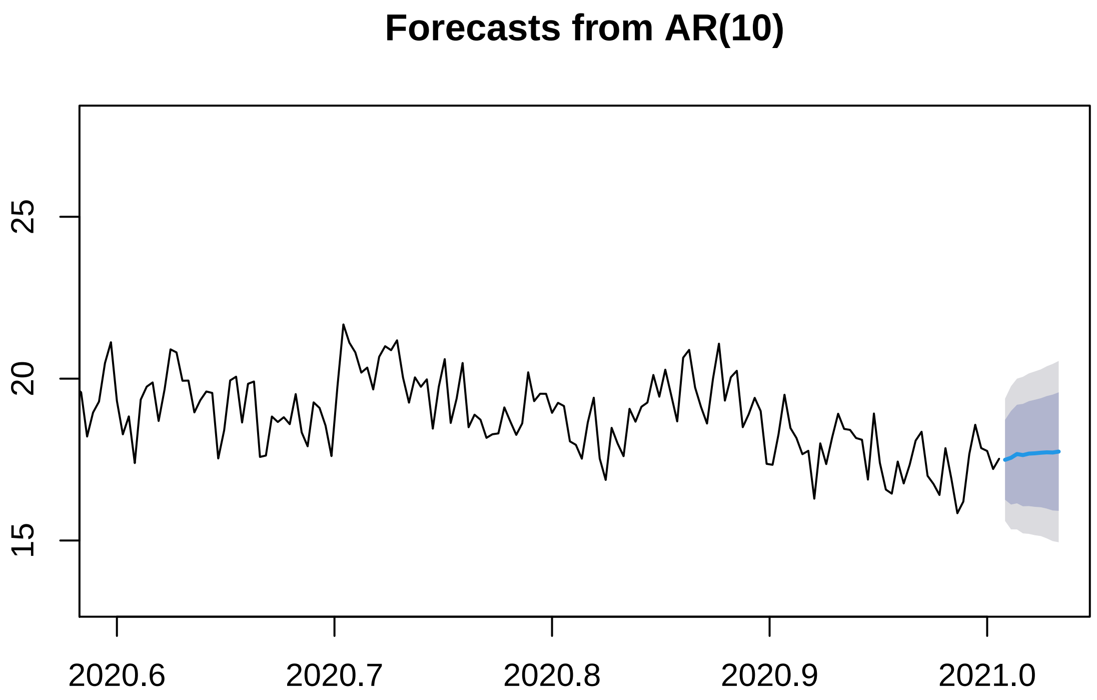


```
#Reduced x scale for better visibilty of forecasts
```

```
no2_ar_2 <-ar(NO2, FALSE, 10)
forecast(no2_ar_2, 10)
```

```
##           Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## 2021.0082       17.49328 16.25886 18.72770 15.60540 19.38116
## 2021.0110       17.55583 16.11020 19.00146 15.34493 19.76673
## 2021.0137       17.67147 16.14924 19.19369 15.34343 19.99950
## 2021.0164       17.63868 16.05781 19.21955 15.22095 20.05641
## 2021.0192       17.68373 16.06351 19.30396 15.20581 20.16166
## 2021.0219       17.69396 16.04011 19.34781 15.16462 20.22330
## 2021.0247       17.71154 16.02844 19.39464 15.13747 20.28561
## 2021.0274       17.72370 15.98617 19.46123 15.06638 20.38102
## 2021.0301       17.71813 15.92862 19.50764 14.98131 20.45495
## 2021.0329       17.74671 15.91557 19.57785 14.94623 20.54720
```

```r
plot(forecast(no2_ar_2, 10), xlim = c(2020.6,2021.03))
```



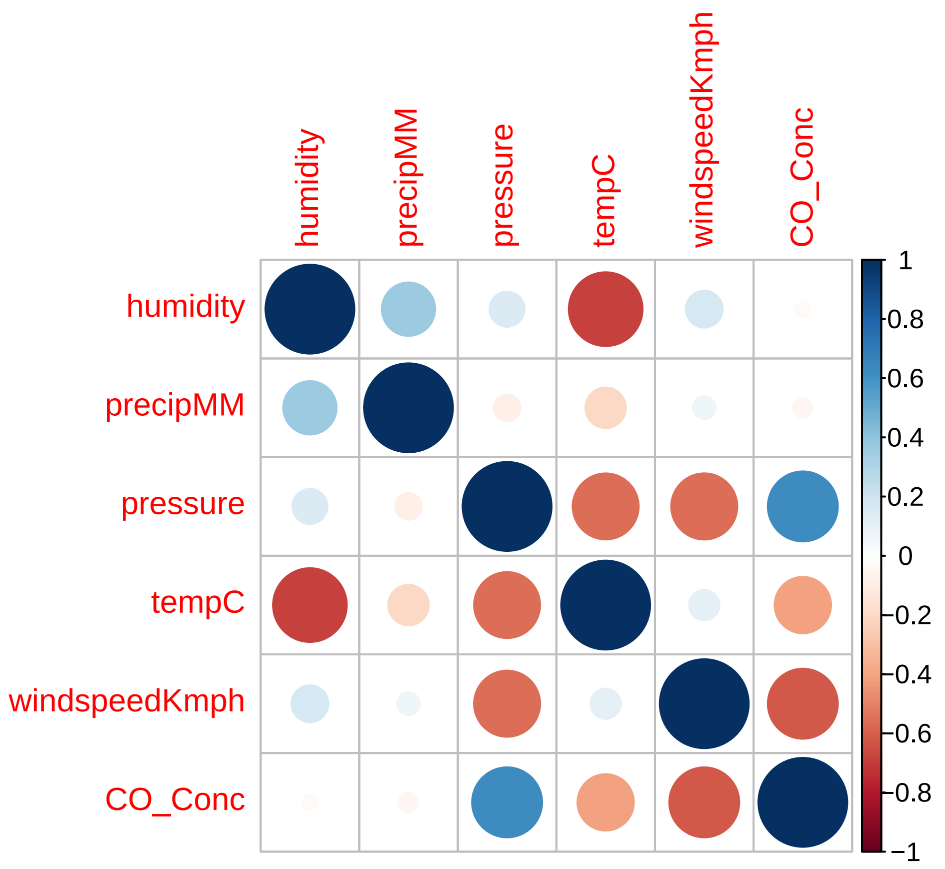
```r
#Reduced x scale for better visibility of forecasts
```

Forecasts don't look the best for Nitrogen Dioxide, showing that there is information that is unaccounted for.

Question 4

Carbon Monoxide, ARDL (3,2), ARDL(5,5)

```r
library(corrplot)
corrplot(cor(pollution_df[,c(2:6,8)]))
```



We see Pressure is has the highest absolute correlation with CO_Conc and this is the variable we will use in our ARDL model

```
pressure <- ts(data = pollution_df$pressure, start = c(2015,1), frequency = 365)
df_ardl_CO <- data.frame(CO, pressure)
```

```
results <- matrix(, nrow = 5, ncol = 5)
library(ARDL)
```

```
## To cite ARDL in publications use:
##
## Kleanthis Natsiopoulos and Nickolaos Tzeremes (2021). ARDL: ARDL, ECM and Bounds-Test for Cointegrat
```

```
for(i in 1:5){
  for(j in 1:5){
    reg_ardl <- ardl(CO ~ pressure, data = df_ardl_CO , order = c(i,j))
    results[i,j] <- AIC(reg_ardl)
  }
}
results
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] -5424.590 -5557.282 -5552.417 -5554.310 -5555.243
## [2,] -5489.977 -5589.517 -5591.255 -5590.236 -5588.467
## [3,] -5555.326 -5626.370 -5626.414 -5643.917 -5644.261
## [4,] -5552.961 -5622.831 -5623.915 -5649.086 -5655.240
## [5,] -5562.364 -5633.953 -5635.600 -5654.083 -5665.235
```

Looks like (4,5) and (5,5) produce the lowest AIC but for variability's sake we will compare an ARDL (5,5) model to an ARDL (3,2) (Randomly chosen)

Model Building

```r
co_ardl_model1 <- dynlm(CO ~ L(CO, 1:3) + L(pressure, 0:2))
summary(co_ardl_model1)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(4), End = 2021(3)
##
## Call:
## dynlm(formula = CO ~ L(CO, 1:3) + L(pressure, 0:2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41419 -0.03454 -0.00180  0.03414  0.33346
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.4984617  0.4862470 -13.365  < 2e-16 ***
## L(CO, 1:3)1        0.9606428  0.0209009  45.962  < 2e-16 ***
## L(CO, 1:3)2       -0.2500757  0.0284813  -8.780  < 2e-16 ***
## L(CO, 1:3)3        0.1243717  0.0191405   6.498 1.01e-10 ***
## L(pressure, 0:2)0  0.0066541  0.0005499  12.101  < 2e-16 ***
## L(pressure, 0:2)1  0.0050641  0.0007823   6.473 1.18e-10 ***
## L(pressure, 0:2)2 -0.0052408  0.0006091  -8.605  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06683 on 2183 degrees of freedom
## Multiple R-squared:  0.8705, Adjusted R-squared:  0.8701
## F-statistic:  2445 on 6 and 2183 DF,  p-value: < 2.2e-16
```

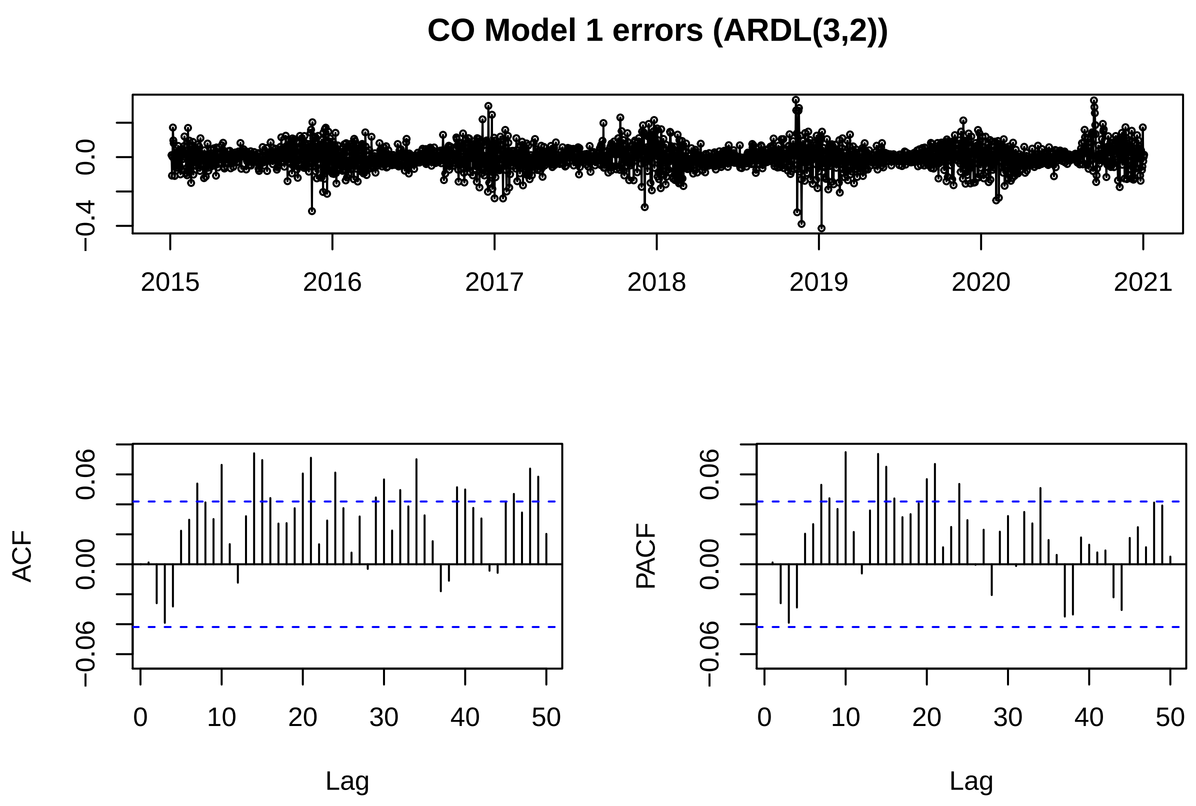
```r
co_ardl_model2 <- dynlm(CO ~ L(CO, 1:5) + L(pressure, 0:5))
summary(co_ardl_model2)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(6), End = 2021(3)
##
## Call:
## dynlm(formula = CO ~ L(CO, 1:5) + L(pressure, 0:5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42207 -0.03429 -0.00168  0.03520  0.32265
```

```
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -5.2081602  0.5755550  -9.049  < 2e-16 ***
## L(CO, 1:5)1         0.9535386  0.0214386  44.478  < 2e-16 ***
## L(CO, 1:5)2        -0.2642395  0.0297098  -8.894  < 2e-16 ***
## L(CO, 1:5)3         0.1017601  0.0300533   3.386 0.000722 ***
## L(CO, 1:5)4         0.0070818  0.0290710   0.244 0.807561
## L(CO, 1:5)5         0.0678968  0.0196313   3.459 0.000553 ***
## L(pressure, 0:5)0  0.0063984  0.0005598  11.431  < 2e-16 ***
## L(pressure, 0:5)1  0.0059808  0.0008191   7.302 3.97e-13 ***
## L(pressure, 0:5)2 -0.0070367  0.0008787  -8.008 1.88e-15 ***
## L(pressure, 0:5)3  0.0028563  0.0008958   3.189 0.001450 **
## L(pressure, 0:5)4 -0.0007612  0.0008539  -0.891 0.372792
## L(pressure, 0:5)5 -0.0022456  0.0006200  -3.622 0.000299 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06609 on 2176 degrees of freedom
## Multiple R-squared:  0.873,  Adjusted R-squared:  0.8724
## F-statistic:  1360 on 11 and 2176 DF,  p-value: < 2.2e-16
```

Comparing Residuals

```
tsdisplay(co_ardl_model1$residuals, main =  "CO Model 1 errors (ARDL(3,2))", lag.max = 50)
```

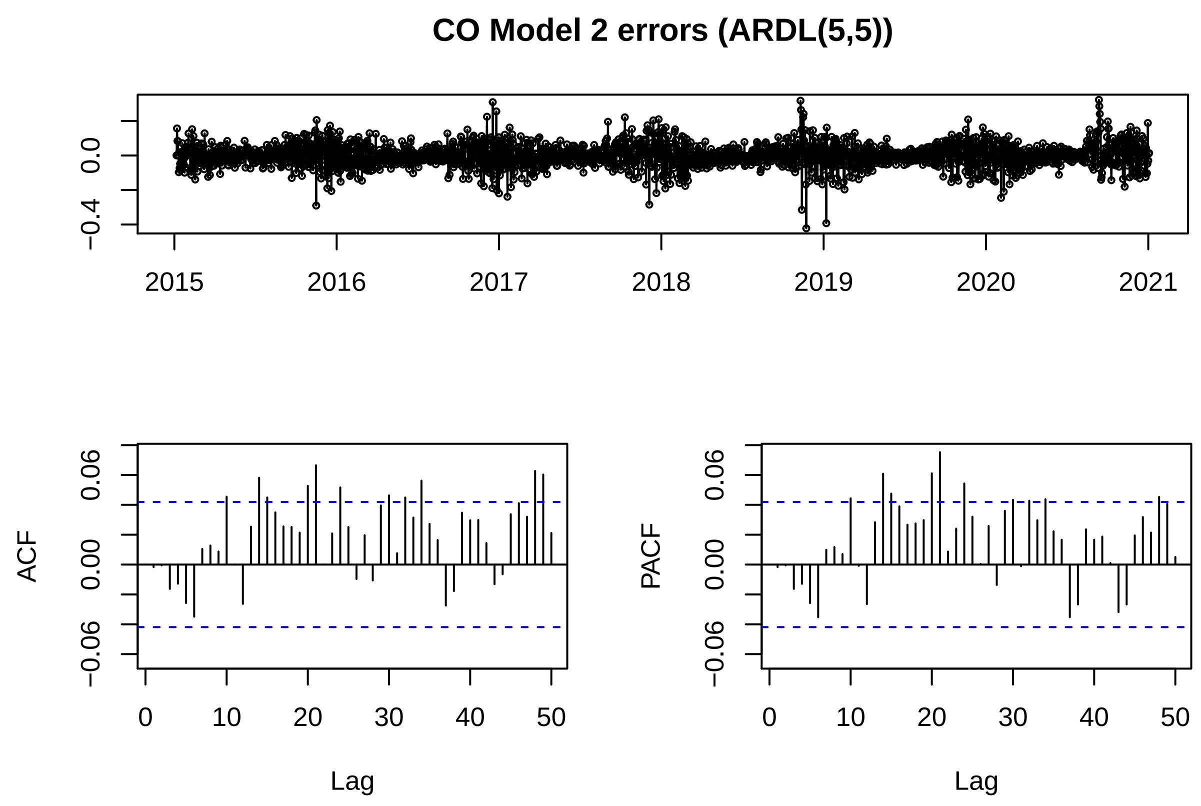


Looking at the ACF and PACF plots above for the ARDL(3,2) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the very last few lags which are significant, showing that there is high order serial correlation. Lags between 10-20 look to have the highest correlation. Even though the spikes for the initial values are low, they pickup as the number of lags increase. This is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```
bgtest(co_ardl_model1, order = 50)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 50
##
## data:  co_ardl_model1
## LM test = 166.65, df = 50, p-value = 1.834e-14
```

```
tsdisplay(co_ardl_model2$residuals, main =  "CO Model 2 errors (ARDL(5,5))", lag.max = 50)
```



Looking at the ACF and PACF plots above for the ARDL(5,5) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the very last few lags which are significant, showing that there is high order serial correlation. Lags between 10-25 look to have the highest correlation. Even though the spikes for the initial values are low, they pickup as the number of lags increase. This is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```
bgtest(co_ardl_model2, order = 50)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 50
##
## data:  co_ardl_model2
## LM test = 149.45, df = 50, p-value = 7.632e-12
```

Training MSE, Testing MSE, AIC

```
CO_train1 <- ts(data = train_CO$CO_Conc, frequency = 365)
CO_test1 <- ts(data = test_CO$CO_Conc, frequency = 365)
pressure_train1 <- ts(data = train_CO$pressure,  frequency = 365)
pressure_test1 <- ts(data = test_CO$pressure, frequency = 365)
df_ardl_CO_train <- data.frame(CO_train1, pressure_train1) # creating a data frame for easier computing
```

```r
library(dLagM)
```

```
## Warning: package 'dLagM' was built under R version 4.0.5

## Loading required package: nardl

##
## Attaching package: 'dLagM'

## The following object is masked from 'package:forecast':
##
##     forecast
```

```r
#Building training model for ARDL(2,3)
co_train_ardl_model1 <- dynlm(CO_train1 ~ L(CO_train1, 1:3) + L(pressure_train1, 0:2))
co_train_ardl_model2 <-ardlDlm(x = train_CO$pressure, y = train_CO$CO_Conc, p = 2, q = 3)
#OOS predictions
num_predictions <- dim(test_CO)[1]
test_preds <- forecast(co_train_ardl_model2, x = pressure_test1, num_predictions)
#MSE calculations
cat("The Training MSE for ARDL(3,2) is:",
    sqrt(mean((train_CO[-(1:3),]$CO_Conc - predict(co_train_ardl_model1)) ^ 2)),
    ", while the Testing MSE for ARDL(3,2) is:",
    sqrt(mean((test_CO$CO_Conc - test_preds$forecasts) ^ 2)))
```

```
## The Training MSE for ARDL(3,2) is: 0.06583137 , while the Testing MSE for ARDL(3,2) is: 0.1501444
```

```r
#Building training model for ARDL(5,5)
co_train_ardl_model3 <- dynlm(CO_train1 ~ L(CO_train1, 1:5) + L(pressure_train1, 0:5))
co_train_ardl_model4 <-ardlDlm(x = train_CO$pressure, y = train_CO$CO_Conc, p = 5, q = 5)
#OOS predictions
num_predictions <- dim(test_CO)[1]
test_preds <- forecast(co_train_ardl_model4, x = pressure_test1, num_predictions)
#MSE Calculations
cat("The Training MSE for ARDL(5,5) is:",
    sqrt(mean((train_CO[-(1:5),]$CO_Conc - predict(co_train_ardl_model3)) ^ 2)),
    ", while the Testing MSE for ARDL(5,5) is:",
    sqrt(mean((test_CO$CO_Conc - test_preds$forecasts) ^ 2)))
```

```
## The Training MSE for ARDL(5,5) is: 0.06502584 , while the Testing MSE for ARDL(5,5) is: 0.151944
```

The Training and Testing MSE for both the ARDL models is lower than those for the AR models, which shows the better performance of the model after adding lags of pressure

```r
cat("The AIC for the ARDL(3,2) model is:", AIC(co_ardl_model1),
    ", while the AIC for the ARDL(5,5) model is:", AIC(co_ardl_model2))
```

```
## The AIC for the ARDL(3,2) model is: -5626.37 , while the AIC for the ARDL(5,5) model is: -5665.235
```

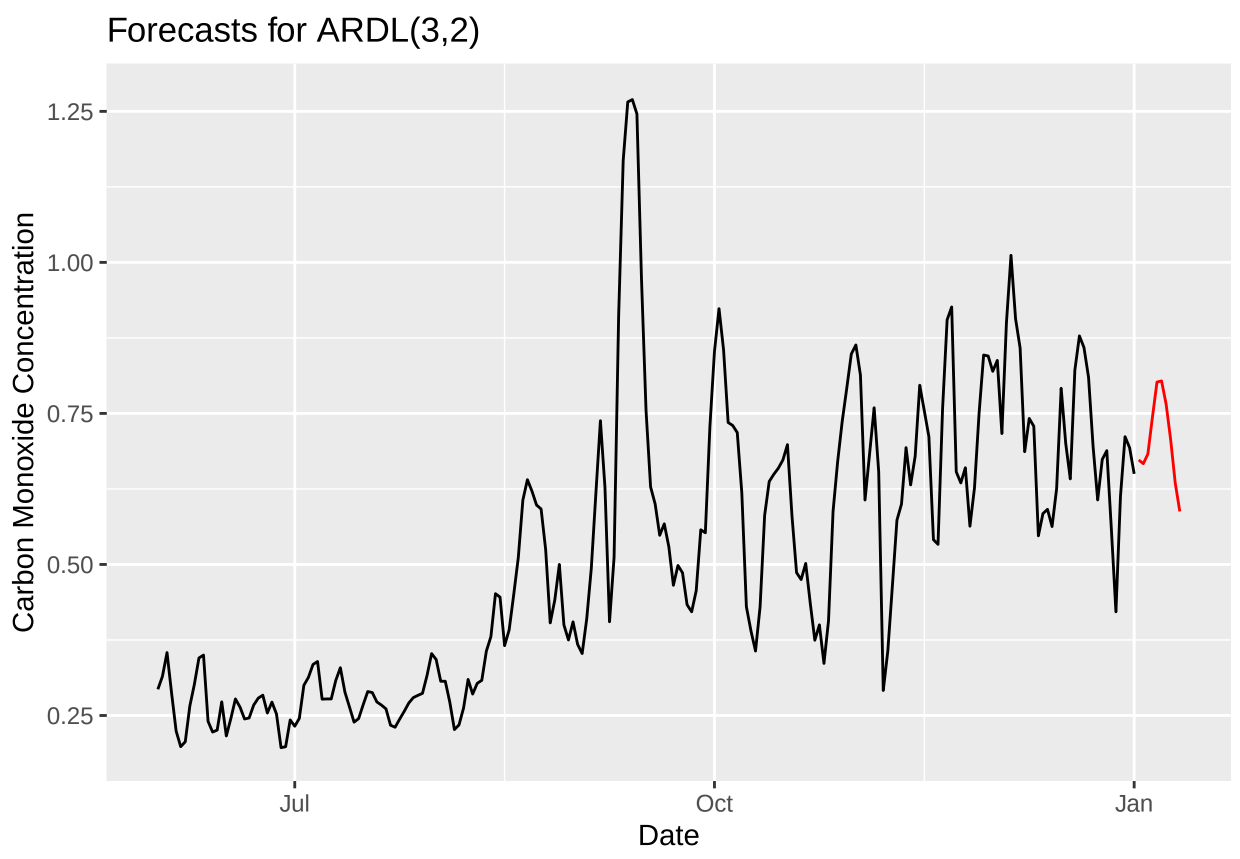
Both models look to be very close to each other as there is not much difference in the training MSE, testing MSE, and AIC values. However, looking closer into the numbers the ARDL(5,5) model has a lower AIC value and a slightly better training MSE but a slightly worse testing MSE by about 0.01. As the difference is so minute, we would still conclude model 2 which is the ARDL(5,5) model for Carbon Monoxide is the better model from the two.

Forecasting

```
co_ardl_1 <- ardlDlm(x = pollution_df$pressure, y = pollution_df$CO_Conc, p = 3, q = 2)
forecast_ardl_1 <- forecast(co_ardl_1, x = pollution_df$pressure, 10)$forecasts
forecast_ardl_1
```

```
##  [1] 0.6729803 0.6668236 0.6826637 0.7437522 0.8018976 0.8037151 0.7655031
##  [8] 0.7063037 0.6348341 0.5877422
```

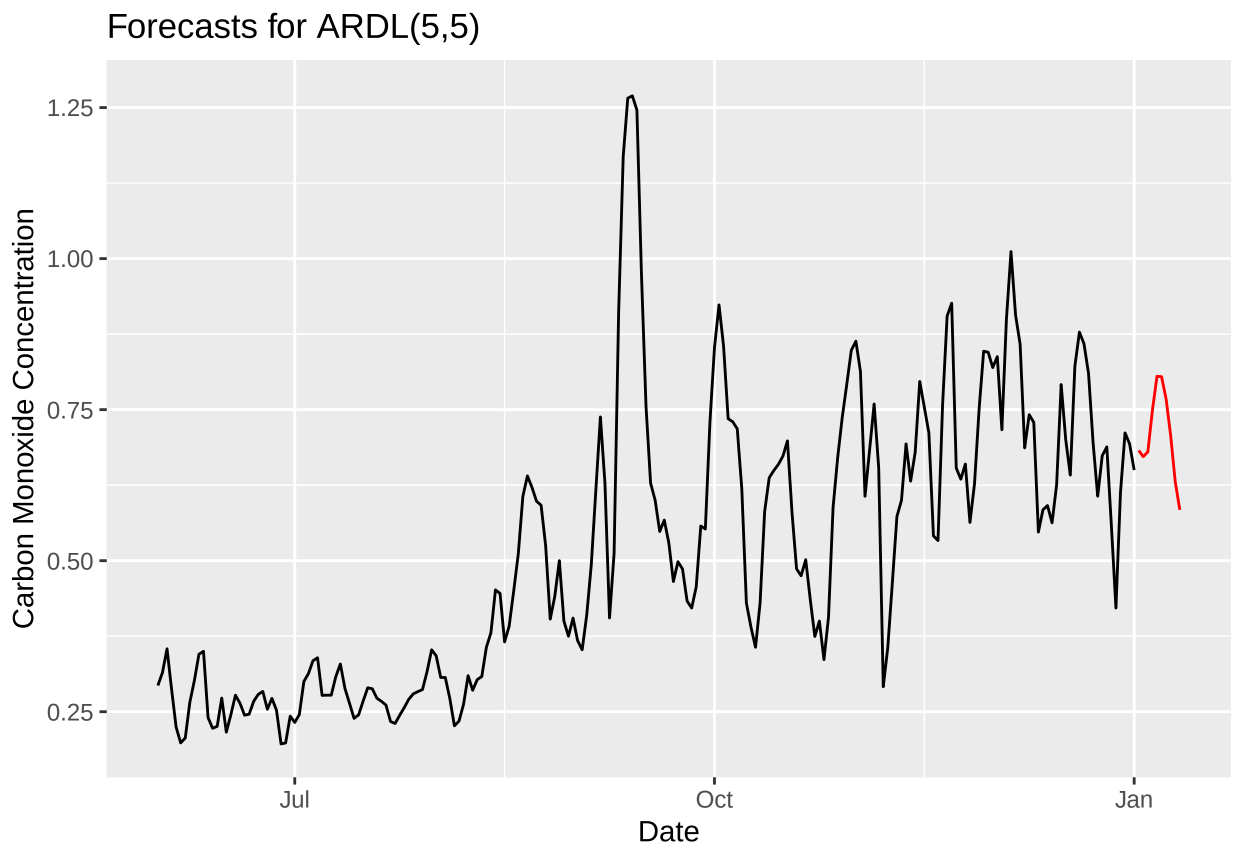
```
df_ardl_forecasts <- data.frame(seq(as.Date("2021-01-02"), as.Date("2021-01-11"),
                                    by = "day"), forecast_ardl_1)
colnames(df_ardl_forecasts) <- c("date_time", "CO_Conc")
library(ggplot2)
ggplot(pollution_df, aes (x = date_time, y = CO_Conc))+
  geom_line()+
  geom_line(data = df_ardl_forecasts, aes(x = date_time, y = CO_Conc), colour = "red")+
  scale_x_date(limits = as.Date(c("2020-06-01", "2021-01-11")))+
  ggtitle("Forecasts for ARDL(3,2)")+
  xlab("Date")+ylab("Carbon Monoxide Concentration")
```



```
co_ardl_2 <- ardlDlm(x = pollution_df$pressure, y = pollution_df$CO_Conc, p = 5, q = 5)
forecast_ardl_2 <- forecast(co_ardl_2, x = pollution_df$pressure, 10)$forecasts
forecast_ardl_2
```

```
##  [1] 0.6824018 0.6721973 0.6801662 0.7491841 0.8052224 0.8045854 0.7675617
##  [8] 0.7069516 0.6307980 0.5841308
```

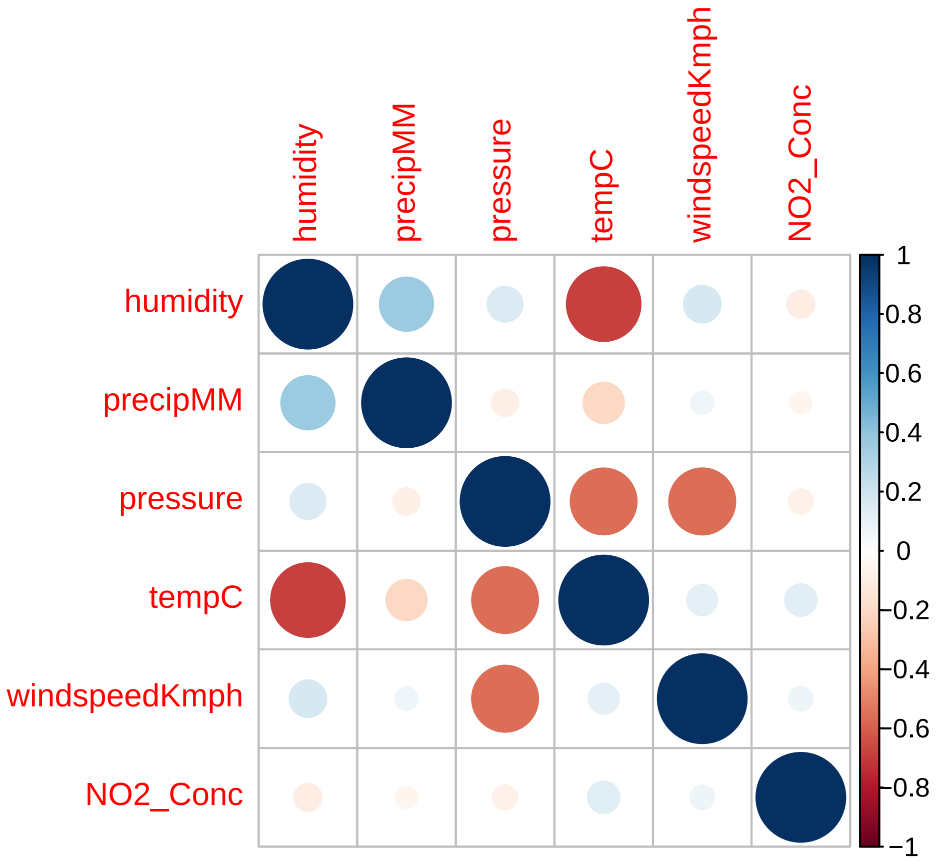
```r
df_ardl_forecasts <- data.frame(seq(as.Date("2021-01-02"), as.Date("2021-01-11")
                                  , by = "day"), forecast_ardl_2)
colnames(df_ardl_forecasts) <- c("date_time", "CO_Conc")
library(ggplot2)
ggplot(pollution_df, aes (x = date_time, y = CO_Conc))+
  geom_line()+
  geom_line(data = df_ardl_forecasts, aes(x = date_time, y = CO_Conc), colour = "red")+
  scale_x_date(limits = as.Date(c("2020-06-01", "2021-01-11")))+
  ggtitle("Forecasts for ARDL(5,5)")+
  xlab("Date")+ylab("Carbon Monoxide Concentration")
```



The forecasts here look way better than those from the AR models, the forecasts here actually follow the pattern in a better manner.

Nitrogen Dioxide, ARDL (5,2), ARDL(2,5)

```r
library(corrplot)
corrplot(cor(pollution_df[,c(2:6,9)]))
```



We see TempC is has the highest absolute correlation with NO2_Conc and this is the variable we will use in our ARDL model

```
temp <- ts(data = pollution_df$tempC, start = c(2015,1), frequency = 365)
df_ardl_NO2 <- data.frame(NO2, temp)
```

```
results <- matrix(, nrow = 5, ncol = 5)
library(ARDL)
for(i in 1:5){
  for(j in 1:5){
    reg_ardl <- ardl(NO2 ~ temp, data = df_ardl_NO2 , order = c(i,j))
    results[i,j] <- AIC(reg_ardl)
  }
}
results
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 6385.160 6379.995 6377.863 6376.765 6376.588
## [2,] 6288.537 6287.606 6286.485 6285.325 6285.110
## [3,] 6186.362 6185.855 6187.837 6187.622 6187.159
## [4,] 6157.600 6157.470 6159.460 6161.052 6160.967
## [5,] 6122.699 6122.656 6124.655 6126.177 6128.059
```

Looks like (1,5) and (2,5) produce the lowest AIC but for variations sake we will compare an ARDL (2,5) model to an ARDL (5,2) (Randomly chosen)

Model Building

```r
no2_ardl_model1 <- dynlm(NO2 ~ L(NO2, 1:5) + L(temp, 0:2))
summary(no2_ardl_model1)
```

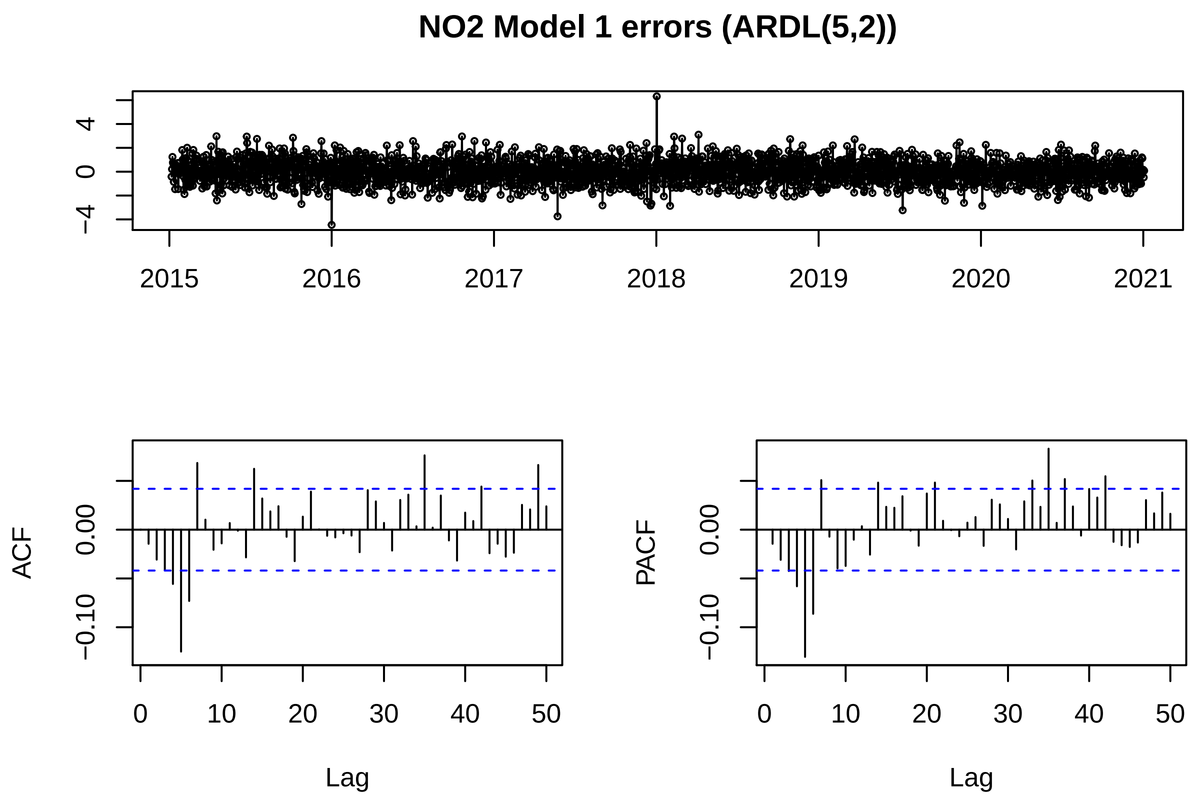
```
##
## Time series regression with "ts" data:
## Start = 2015(6), End = 2021(3)
##
## Call:
## dynlm(formula = NO2 ~ L(NO2, 1:5) + L(temp, 0:2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4608 -0.6524 -0.0031  0.6573  6.3182
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.737392   0.170585   4.323 1.61e-05 ***
## L(NO2, 1:5)1    0.644375   0.021244  30.333  < 2e-16 ***
## L(NO2, 1:5)2    0.027679   0.025306   1.094    0.274
## L(NO2, 1:5)3    0.128610   0.025162   5.111 3.48e-07 ***
## L(NO2, 1:5)4    0.030747   0.025284   1.216    0.224
## L(NO2, 1:5)5    0.125412   0.021203   5.915 3.85e-09 ***
## L(temp, 0:2)0 -0.006802   0.007455  -0.912    0.362
## L(temp, 0:2)1  0.021444   0.010591   2.025    0.043 *
## L(temp, 0:2)2 -0.010649   0.007463  -1.427    0.154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9779 on 2179 degrees of freedom
## Multiple R-squared:  0.8553, Adjusted R-squared:  0.8548
## F-statistic:  1610 on 8 and 2179 DF,  p-value: < 2.2e-16
```

```r
no2_ardl_model2 <- dynlm(NO2 ~ L(NO2, 1:2) + L(temp, 0:5))
summary(no2_ardl_model2)
```

```
##
## Time series regression with "ts" data:
## Start = 2015(6), End = 2021(3)
##
## Call:
## dynlm(formula = NO2 ~ L(NO2, 1:2) + L(temp, 0:5))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5894 -0.6840 -0.0024  0.6815  6.1035
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.241464   0.172232   7.208 7.79e-13 ***
## L(NO2, 1:2)1    0.725711   0.020969  34.608  < 2e-16 ***
## L(NO2, 1:2)2   0.204380   0.020957   9.752  < 2e-16 ***
## L(temp, 0:5)0 -0.006628   0.007936  -0.835   0.4037
## L(temp, 0:5)1  0.021765   0.011130   1.956   0.0506 .
## L(temp, 0:5)2 -0.010398   0.011246  -0.925   0.3553
## L(temp, 0:5)3 -0.008827   0.011242  -0.785   0.4325
## L(temp, 0:5)4  0.004613   0.011133   0.414   0.6786
## L(temp, 0:5)5  0.004052   0.007939   0.510   0.6098
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.015 on 2179 degrees of freedom
## Multiple R-squared:  0.8442, Adjusted R-squared:  0.8436
## F-statistic:  1476 on 8 and 2179 DF,  p-value: < 2.2e-16
```

Comparing Residuals

```
tsdisplay(no2_ardl_model1$residuals, main =  "NO2 Model 1 errors (ARDL(5,2))", lag.max = 50)
```

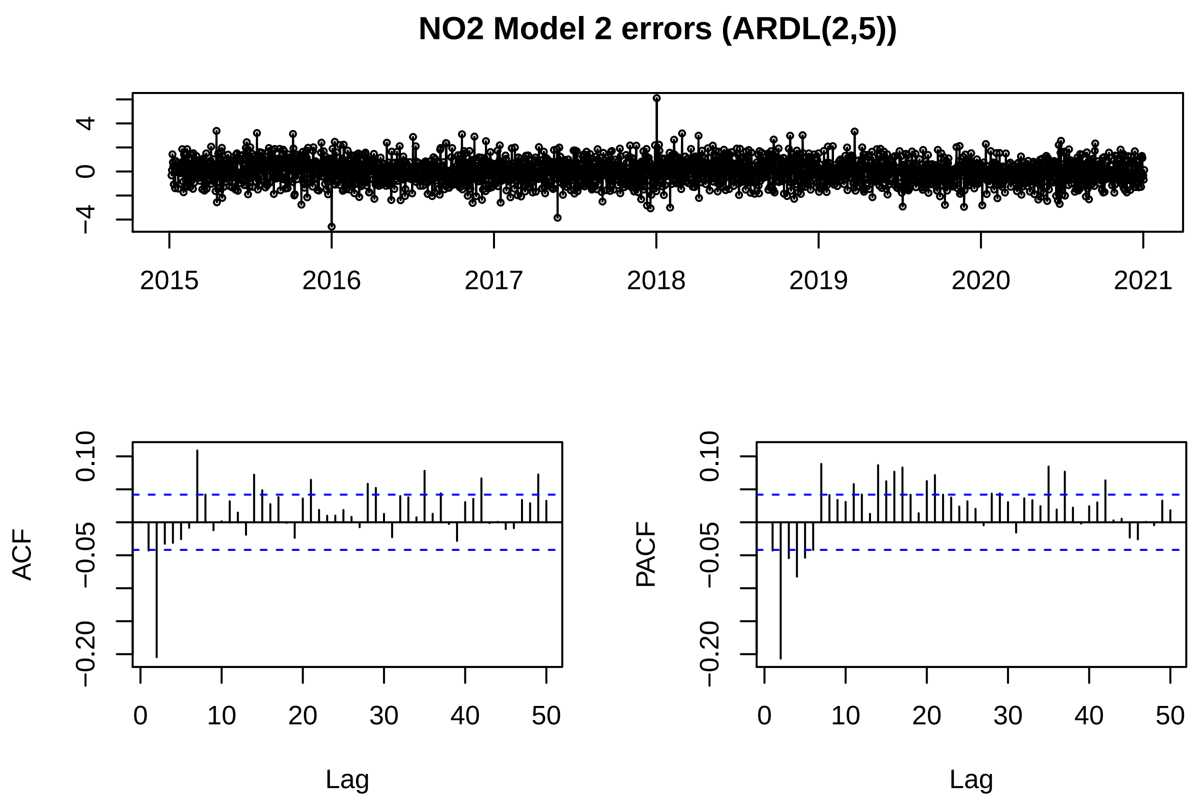


Looking at the ACF and PACF plots above for the ARDL(5,2) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the very last few lags which are significant, showing that there is high order serial correlation. Lags between 30-40 look to have the highest correlation. Even though the spikes for the initial values are low, they pickup quickly between lags 5 and 10. The serial correlation is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```r
bgtest(no2_ardl_model1, order = 10)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  no2_ardl_model1
## LM test = 103.17, df = 10, p-value < 2.2e-16
```

```r
tsdisplay(no2_ardl_model2$residuals, main =  "NO2 Model 2 errors (ARDL(2,5))", lag.max = 50)
```



Looking at the ACF and PACF plots above for the ARDL(2,5) model it looks like there is proof of serial correlation. This is because there are multiple spikes for lags up to the very last few lags which are significant, showing that there is high order serial correlation. Initial lags between 1-10 look to have the highest correlation. There is consistent correlation between lags 20-40 too. The serial correlation is confirmed by the low p value of the test below. This implies that this model may not be the best as there are dynamics that are unaccounted for.

```r
bgtest(no2_ardl_model2, order = 10)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  no2_ardl_model2
## LM test = 242.35, df = 10, p-value < 2.2e-16
```

Training MSE, Testing MSE, AIC

```r
NO2_train1 <- ts(data = train_CO$NO2_Conc, frequency = 365)
NO2_test1 <- ts(data = test_CO$NO2_Conc, frequency = 365)
temp_train1 <- ts(data = train_CO$tempC,  frequency = 365)
temp_test1 <- ts(data = test_CO$tempC, frequency = 365)
df_ardl_NO2_train <- data.frame(NO2_train1, temp_train1)
```

```r
#Building ARDL(5,2) training model
no2_train_ardl_model1 <- dynlm(NO2_train1 ~ L(NO2_train1, 1:3) + L(temp_train1, 0:5))
no2_train_ardl_model2 <-ardlDlm(x = train_no2$tempC, y = train_no2$NO2_Conc, p = 2, q = 5)
#OOS predictions
num_predictions <- dim(test_CO)[1]
test_preds <- forecast(no2_train_ardl_model2, x = temp_test1, num_predictions)
#MSE Calculations
cat("The Training MSE for ARDL(5,2) is:",
    sqrt(mean((train_no2[-(1:5),]$NO2_Conc - predict(no2_train_ardl_model1)) ^ 2)),
    ", while the Testing MSE for ARDL(5,2) is:",
    sqrt(mean((test_no2$NO2_Conc - test_preds$forecasts) ^ 2)))
```

```
## The Training MSE for ARDL(5,2) is: 1.025425 , while the Testing MSE for ARDL(5,2) is: 2.990738
```

```r
# Building AR(2,5) training model
no2_train_ardl_model3 <- dynlm(NO2_train1 ~ L(NO2_train1, 1:5) + L(temp_train1, 0:2))
no2_train_ardl_model4 <-ardlDlm(x = train_no2$tempC, y = train_no2$NO2_Conc, p = 5, q = 2)
#OOS Predictions
num_predictions <- dim(test_CO)[1]
test_preds <- forecast(no2_train_ardl_model4, x = temp_test1, num_predictions)
#MSE Calculations
cat("The Training MSE for ARDL(2,5) is:",
    sqrt(mean((train_no2[-(1:5),]$NO2_Conc - predict(no2_train_ardl_model3)) ^ 2)),
  ", while the Testing MSE for ARDL(2,5) is:",
sqrt(mean((test_no2$NO2_Conc - test_preds$forecasts) ^ 2)))
```

```
## The Training MSE for ARDL(2,5) is: 1.009666 , while the Testing MSE for ARDL(2,5) is: 2.998237
```

```r
cat("The AIC for the ARDL(5,2) model is:", AIC(no2_ardl_model1),
    ", while the AIC for the ARDL(2,5) model is:", AIC(no2_ardl_model2))
```

```
## The AIC for the ARDL(5,2) model is: 6122.656 , while the AIC for the ARDL(2,5) model is: 6285.11
```

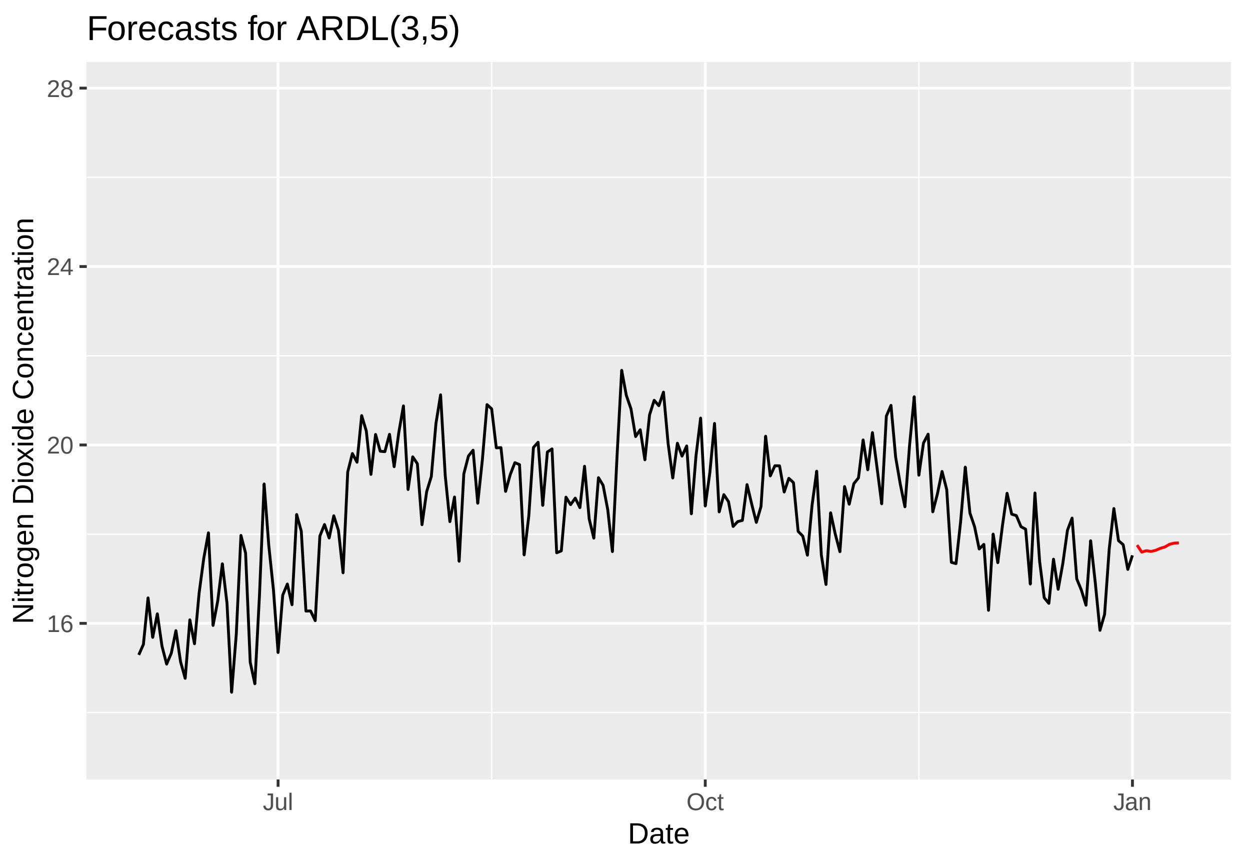
From the above numbers it looks like the ARDL(5,2) is clearly the best model as it has the lowest AIC, Training MSE and Testing MSE.

Forecasting

```r
no2_ardl_1 <- ardlDlm(x = pollution_df$tempC, y = pollution_df$NO2_Conc, p = 2, q = 5)
forecast_ardl_1 <- forecast(no2_ardl_1, x = pollution_df$tempC, 10)
forecast_ardl_1
```

```
## $forecasts
##  [1] 17.75444 17.59720 17.62932 17.61239 17.63688 17.68145 17.71256 17.77289
##  [9] 17.79865 17.80288
##
## $call
## forecast.ardlDlm(model = no2_ardl_1, x = pollution_df$tempC,
##     h = 10)
##
## attr(,"class")
## [1] "forecast.ardlDlm" "dLagM"
```

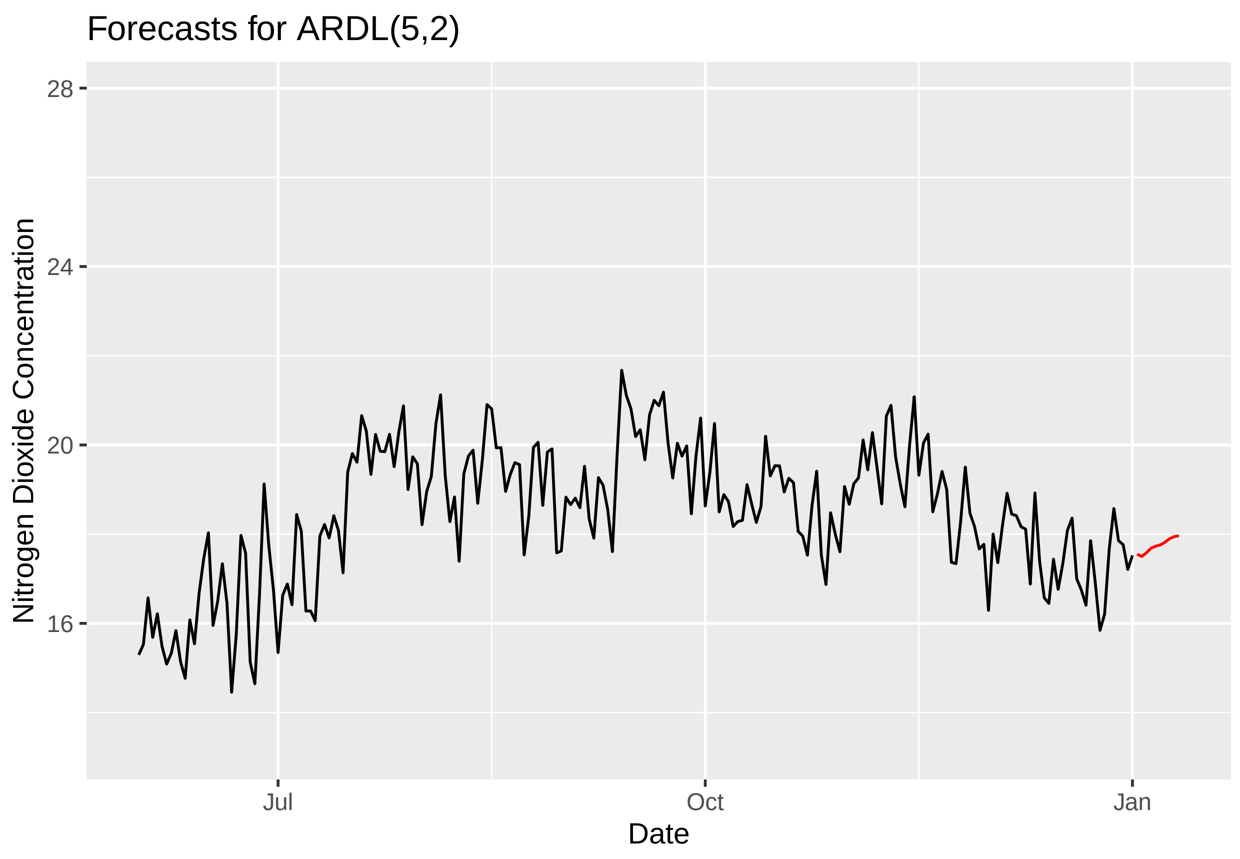
```r
df_ardl_forecasts <- data.frame(seq(as.Date("2021-01-02"), as.Date("2021-01-11"),
                                    by = "day"), forecast_ardl_1$forecasts)
colnames(df_ardl_forecasts) <- c("date_time", "NO2_Conc")
library(ggplot2)
ggplot(pollution_df, aes (x = date_time, y = NO2_Conc))+
  geom_line()+
  geom_line(data = df_ardl_forecasts, aes(x = date_time, y = NO2_Conc), colour = "red")+
  scale_x_date(limits = as.Date(c("2020-06-01", "2021-01-11")))+
  ggtitle("Forecasts for ARDL(3,5)")+
  xlab("Date")+ylab("Nitrogen Dioxide Concentration")
```



```r
no2_ardl_2 <- ardlDlm(x = pollution_df$tempC, y = pollution_df$NO2_Conc, p = 5, q = 2)
forecast_ardl_2 <- forecast(no2_ardl_2, x = pollution_df$tempC, 10)
forecast_ardl_2
```

```
## $forecasts
##  [1] 17.55053 17.49948 17.58208 17.68364 17.72954 17.75833 17.81695 17.89915
##  [9] 17.94370 17.96464
##
## $call
## forecast.ardlDlm(model = no2_ardl_2, x = pollution_df$tempC,
##     h = 10)
##
## attr(,"class")
## [1] "forecast.ardlDlm" "dLagM"
```

```
df_ardl_forecasts <- data.frame(seq(as.Date("2021-01-02"), as.Date("2021-01-11"),
                                    by = "day"), forecast_ardl_2$forecasts )
colnames(df_ardl_forecasts) <- c("date_time", "NO2_Conc")
library(ggplot2)
ggplot(pollution_df, aes (x = date_time, y = NO2_Conc))+
  geom_line()+
  geom_line(data = df_ardl_forecasts, aes(x = date_time, y = NO2_Conc), colour = "red")+
  scale_x_date(limits = as.Date(c("2020-06-01", "2021-01-11")))+
  ggtitle("Forecasts for ARDL(5,2)")+
  xlab("Date")+ylab("Nitrogen Dioxide Concentration")
```



The forecasts here look better than the AR models, as TempC adds a bit more information than just the AR model.

Question 5

a)

```
library(vars)
```

```
## Loading required package: MASS

## Loading required package: strucchange

## Loading required package: sandwich

## Loading required package: urca
```

```
var_df <- data.frame(CO, NO2)
VARselect(var_df)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      7      7     10
```

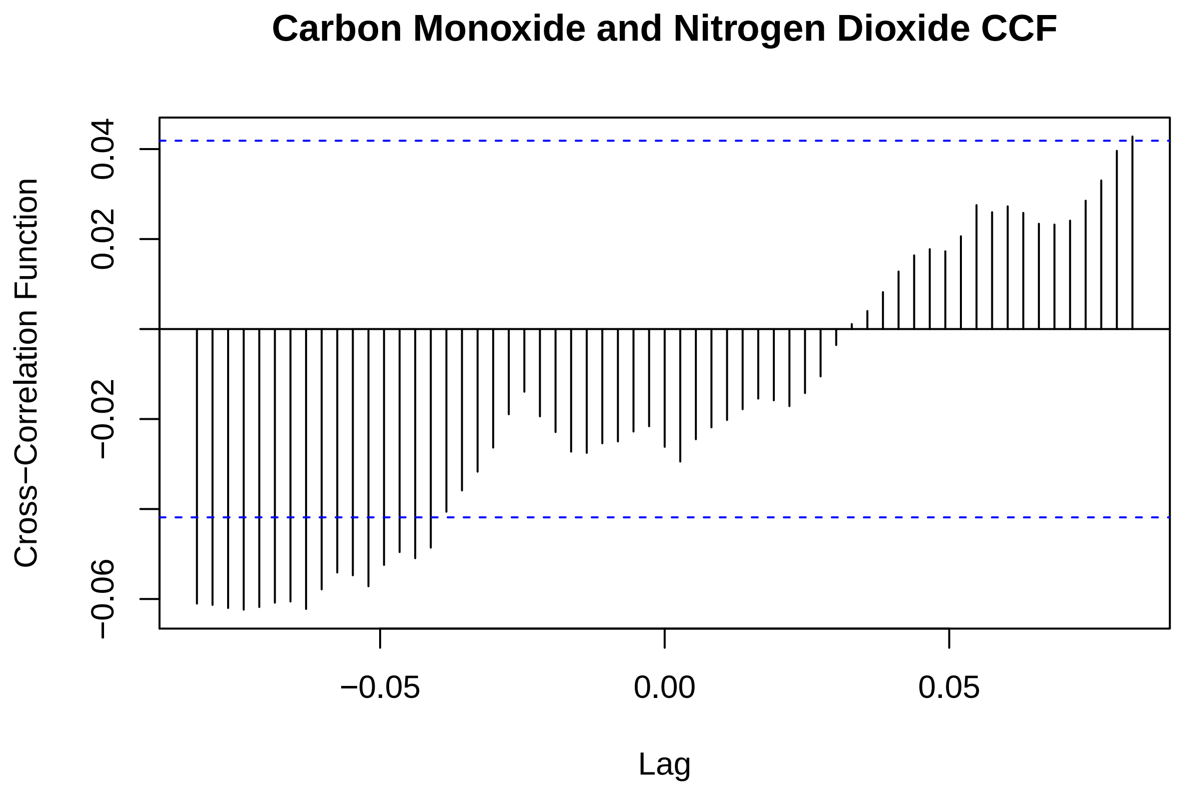
Looks like p = 10 is the best

```
var_model1 <- VAR(var_df, p =10)
summary(var_model1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: CO, NO2
## Deterministic variables: const
## Sample size: 2183
## Log Likelihood: -412.386
## Roots of the characteristic polynomial:
## 0.9897 0.9733 0.8028 0.8028 0.7714 0.7714 0.7644 0.7644 0.7535 0.7535 0.7445 0.7445 0.7158 0.7158 0.
## Call:
## VAR(y = var_df, p = 10)
##
##
## Estimation results for equation CO:
## ===================================
## CO = CO.l1 + NO2.l1 + CO.l2 + NO2.l2 + CO.l3 + NO2.l3 + CO.l4 + NO2.l4 + CO.l5 + NO2.l5 + CO.l6 + NO
##
##          Estimate Std. Error t value Pr(>|t|)
## CO.l1    1.0389612  0.0214877  48.351  < 2e-16 ***
## NO2.l1  -0.0031282  0.0016588  -1.886   0.0594 .
## CO.l2   -0.3886381  0.0310010 -12.536  < 2e-16 ***
## NO2.l2   0.0028543  0.0019412   1.470   0.1416
## CO.l3    0.1644708  0.0321081   5.122 3.28e-07 ***
## NO2.l3  -0.0015974  0.0019421  -0.823   0.4109
## CO.l4   -0.0240510  0.0323017  -0.745   0.4566
## NO2.l4   0.0004616  0.0019374   0.238   0.8117
## CO.l5    0.0397163  0.0323384   1.228   0.2195
## NO2.l5   0.0003811  0.0019377   0.197   0.8441
## CO.l6    0.0320798  0.0323412   0.992   0.3213
## NO2.l6   0.0008301  0.0019374   0.428   0.6684
## CO.l7    0.0189787  0.0323382   0.587   0.5573
## NO2.l7  -0.0005037  0.0019380  -0.260   0.7950
## CO.l8   -0.0137540  0.0321525  -0.428   0.6689
## NO2.l8  -0.0011913  0.0019431  -0.613   0.5399
## CO.l9    0.0315894  0.0310491   1.017   0.3091
## NO2.l9   0.0009922  0.0019406   0.511   0.6092
## CO.l10   0.0469029  0.0214851   2.183   0.0291 *
## NO2.l10  0.0008908  0.0016580   0.537   0.5912
## const    0.0240113  0.0138061   1.739   0.0821 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.07431 on 2162 degrees of freedom
## Multiple R-Squared: 0.838,   Adjusted R-squared: 0.8365
## F-statistic: 559.2 on 20 and 2162 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation NO2:
## ====================================
## NO2 = CO.l1 + NO2.l1 + CO.l2 + NO2.l2 + CO.l3 + NO2.l3 + CO.l4 + NO2.l4 + CO.l5 + NO2.l5 + CO.l6 + NO
##
##          Estimate Std. Error t value Pr(>|t|)
## CO.l1    0.200740   0.278006   0.722  0.47033
## NO2.l1   0.609874   0.021461  28.418  < 2e-16 ***
## CO.l2   -0.179525   0.401086  -0.448  0.65449
## NO2.l2   0.012095   0.025115   0.482  0.63016
## CO.l3   -0.167701   0.415411  -0.404  0.68647
## NO2.l3   0.101667   0.025126   4.046 5.39e-05 ***
## CO.l4    0.189772   0.417916   0.454  0.64981
## NO2.l4   0.009089   0.025066   0.363  0.71694
## CO.l5   -0.193750   0.418390  -0.463  0.64335
## NO2.l5   0.042076   0.025070   1.678  0.09342 .
## CO.l6   -0.176749   0.418426  -0.422  0.67276
## NO2.l6   0.018781   0.025066   0.749  0.45379
## CO.l7    0.439851   0.418387   1.051  0.29324
## NO2.l7   0.130118   0.025073   5.190 2.30e-07 ***
## CO.l8   -0.646964   0.415985  -1.555  0.12003
## NO2.l8  -0.001219   0.025139  -0.048  0.96133
## CO.l9    1.170904   0.401709   2.915  0.00360 **
## NO2.l9   0.002607   0.025107   0.104  0.91730
## CO.l10  -0.677774   0.277971  -2.438  0.01484 *
## NO2.l10  0.046439   0.021451   2.165  0.03051 *
## const    0.564050   0.178622   3.158  0.00161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.9614 on 2162 degrees of freedom
## Multiple R-Squared: 0.861,   Adjusted R-squared: 0.8598
## F-statistic: 669.8 on 20 and 2162 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             CO       NO2
## CO   0.005522 -0.001952
## NO2 -0.001952  0.924252
##
## Correlation matrix of residuals:
##            CO      NO2
## CO   1.00000 -0.02732
## NO2 -0.02732  1.00000
```

```
ccf(CO, NO2, ylab="Cross-Correlation Function", main = "Carbon Monoxide and Nitrogen Dioxide CCF")
```



It looks like Carbon Monoxide levels and Nitrogen Dioxide levels are negatively correlated with Carbon Monoxide lagged by few days (around 20-40 days). This means that increases in Carbon Monoxide levels relate to Nitrogen Dioxide levels to falling over the next few days, and vice-versa. This is interesting as one would believe atmospheric pollutants move in similar directions.

b)

```r
grangertest(CO~ NO2, order = 10)
```

```
## Granger causality test
##
## Model 1: CO ~ Lags(CO, 1:10) + Lags(NO2, 1:10)
## Model 2: CO ~ Lags(CO, 1:10)
##   Res.Df  Df      F Pr(>F)
## 1   2162
## 2   2172 -10 0.5886 0.8245
```

The Granger Causality Test above has a very high p value so we fail to reject the null hypothesis. This implies that Nitrogen Dioxide doesn't Granger Cause Carbon Monoxide, i.e. changes in Nitrogen Dioxide don't necessarily lead to changes in Carbon Monoxide.

```r
grangertest(NO2 ~ CO, order = 10)
```

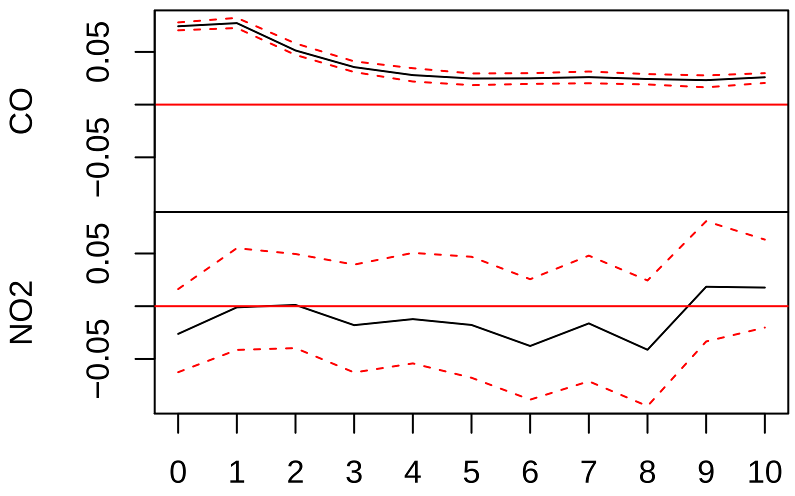
```
## Granger causality test
##
## Model 1: NO2 ~ Lags(NO2, 1:10) + Lags(CO, 1:10)
## Model 2: NO2 ~ Lags(NO2, 1:10)
##   Res.Df  Df      F Pr(>F)
## 1   2162
## 2   2172 -10 1.1401 0.3278
```

The Granger Causality Test above has a high p value so we fail to reject the null hypothesis. This implies that Carbon Monoxide doesn't Granger Cause Nitrogen Dioxide, i.e. changes in Carbon Monoxide don't necessarily lead to changes in Nitrogen Dioxide.

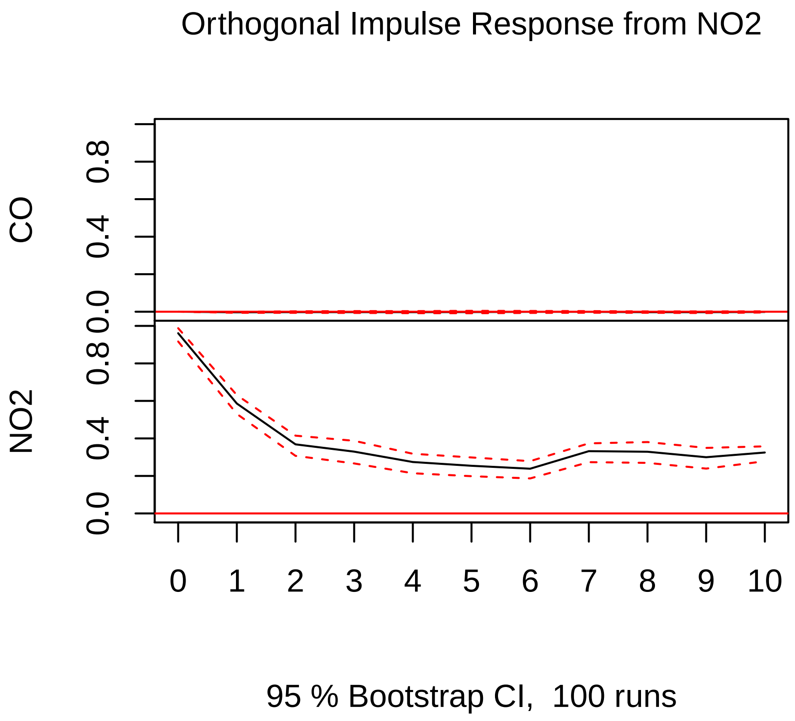
c)

```r
plot(irf(var_model1))
```

Orthogonal Impulse Response from CO



95 % Bootstrap CI, 100 runs



From the first Impulse Response Function we can see the effect of a shock (Sudden Change) of Carbon Monoxide. A Carbon Monoxide shock initially has a large effect on subsequent CO values but then decays to 0 slowly. However, from the bottom plot on the initial plot it looks like a Carbon Monoxide shock has no real effect on subsequent nitrogen dioxide values as the IRF plot fluctuates around zero.

From the second set of IRF plots for Nitrogen Dioxide we see a Nitrogen Dioxide shock has no effect on Carbon Monoxide values. This is interesting as one would expect atmospheric pollutants to increase and decrease in the same directions. However, looking at the plot in the bottom half it looks like a Nitrogen Dioxide shock has an initial large effect of subsequent NO2 values and then slowly decreases but doesn't go to 0, it remain constant at around 0.4. So a NO2 shock has a long lasting impact on subsequent NO2 values.

d)

```
#plot(var_model1)
#When plotting the line of code above, there was an error that figure margins
#were too large,hence screenshots of the graph were included
```

```
knitr::include_graphics("CO_VAR_plot.png")
```

```
knitr::include_graphics("NO2_VAR_plot.png")
```

The model looks pretty good for CO, with the fit being almost perfect. Looking at the ACF and PACF plots we can see that there are no significant lag (except the 0th lag) showing that there maybe no serial

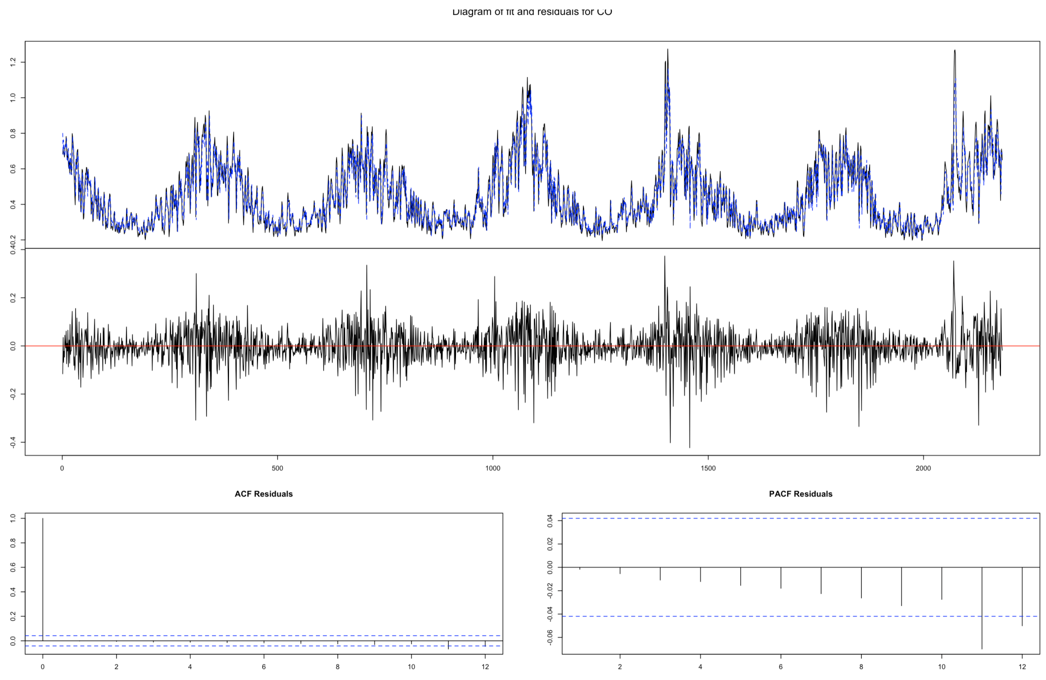


Figure 1: VAR(10) plot for CO

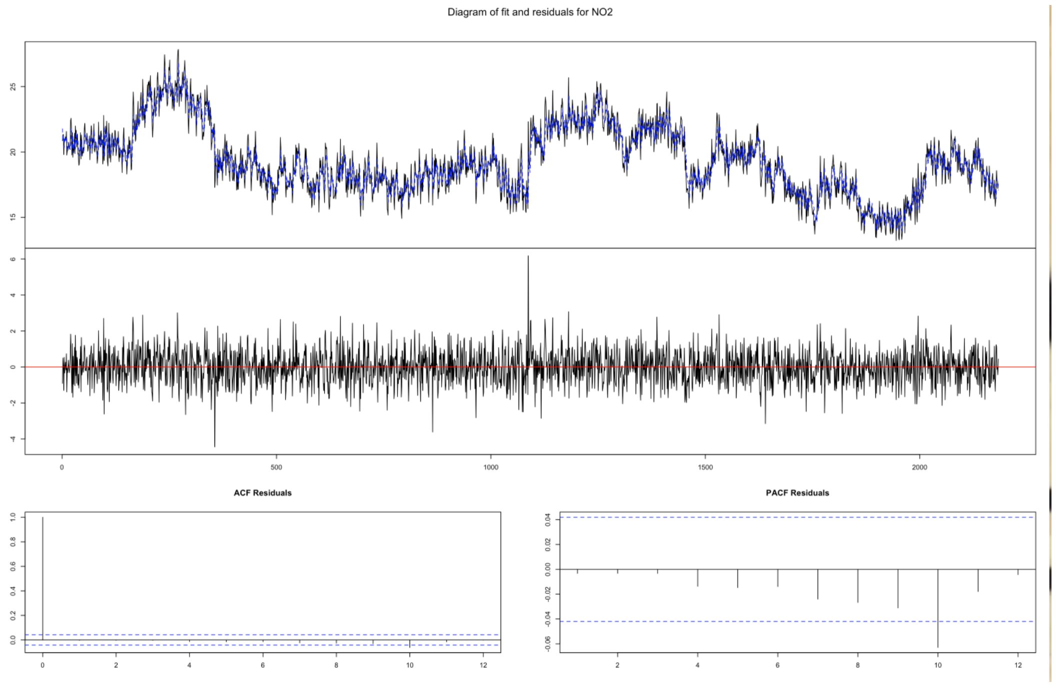


Figure 2: VAR(10) plot for CO

autocorrelation, implying that our model may capture almost all the necessary information. Similarly for NO2, the model fits pretty well, capturing the general trend. Looking at the ACF and PACF plots we can see that there are no significant lag (except the 0th lag and 10th lag of the PACF) showing that there maybe no serial autocorrelation, implying that our model may capture almost all the necessary information.

e)

```r
train_var_1 <- data.frame(train_CO$CO_Conc, train_no2$NO2_Conc)
var_train_model <- VAR(train_var_1, p =10)
```

```r
num_predictions <- dim(test_CO)[1]
test_fitted <- predict(var_train_model, n.ahead = num_predictions)
cat("The Training MSE for VAR(10) for CO is:", sqrt(mean((train_CO[-(1:10),]$CO_Conc - var_train_model$
    ", while the Testing MSE for VAR(10) for CO is:",
    sqrt(mean((test_CO$CO_Conc - test_fitted$fcst$train_CO.CO_Conc[,1]) ^ 2)))
```

```
## The Training MSE for VAR(10) for CO is: 0.07253911 , while the Testing MSE for VAR(10) for CO is: 0.
```

```r
cat("The Training MSE for VAR(10) for NO2 is:", sqrt(mean((train_no2[-(1:10),]$NO2_Conc - var_train_mode
    ", while the Testing MSE for VAR(10) for NO2 is:",
    sqrt(mean((test_no2$NO2_Conc - test_fitted$fcst$train_no2.NO2_Conc[,1]) ^ 2)))
```

```
## The Training MSE for VAR(10) for NO2 is: 0.9826341 , while the Testing MSE for VAR(10) for NO2 is: 3
```

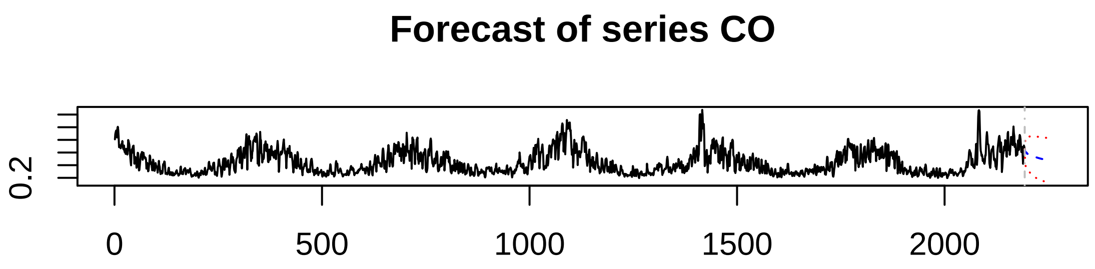
```r
cat("The AIC for the VAR(10) model for CO is:", AIC(var_model1$varresult$CO), ",while the AIC
    for the VAR(10) model for NO2 is:", AIC(var_model1$varresult$NO2))
```

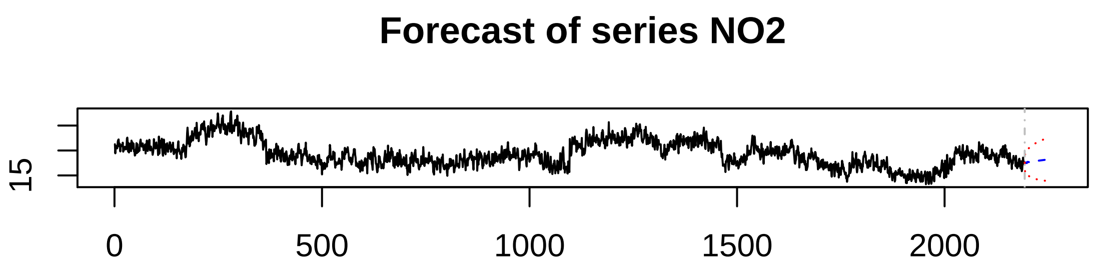
```
## The AIC for the VAR(10) model for CO is: -5131.625 ,while the AIC
##     for the VAR(10) model for NO2 is: 6046.027
```

While it is tough to compare the models due to different units, based on the AIC criterion and the fact that the training and testing MSE of the CO model are closer than those of than those of the NO2 model, we can conclude that the CO model is better than the NO2 model.

f)

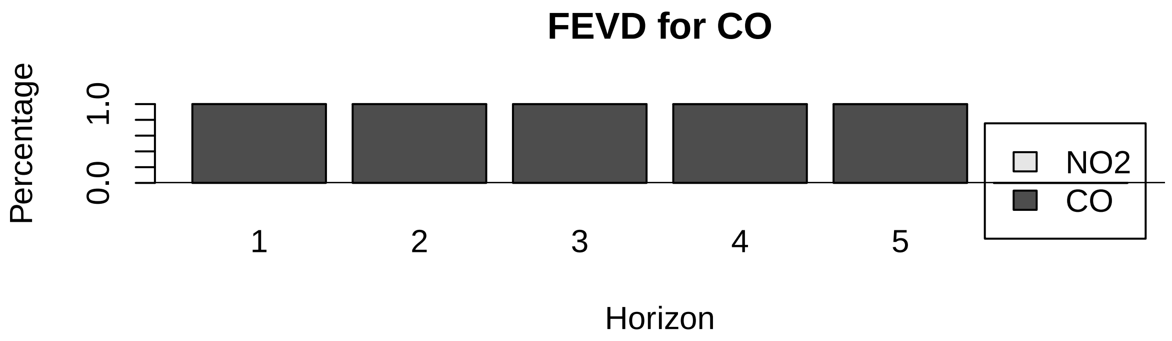
```r
plot(predict(var_model1, n.ahead = 62))
```

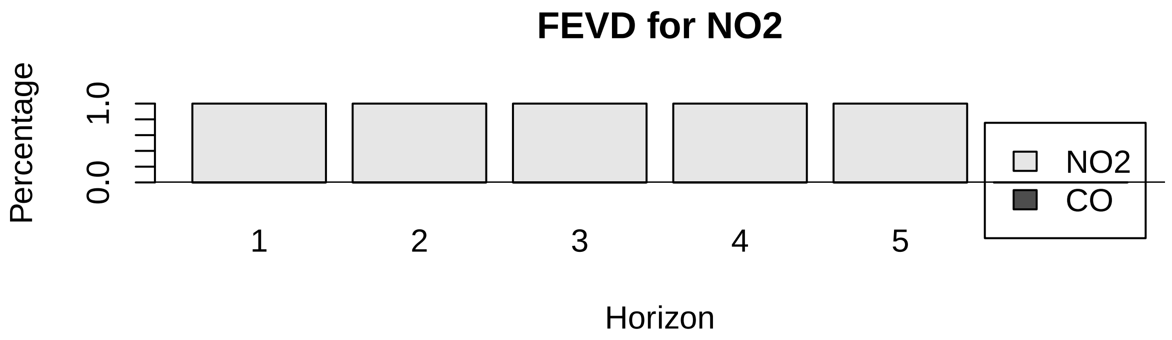




Forecasted for 2 months ahead. The forecasts capture the general trend of both the time series. ## g)

```
plot(fevd(var_model1, n.ahead = 5))
```





From the FEVD plots above it looks like for each period CO contributes to 100% of the variation in CO, while it looks like CO contributes <5% of the variation in NO2. This shows that the variables aren't really related as they don't contribute to each others variation.

Question 6

To conclude, we used 2 predictors pressure, tempC and lagged values of itself to predict concentration levels of Nitrogen Dioxide and Carbon Monoxide. The main finding is from the VAR(10) analysis where we see that there is no real connection between Carbon Monoxide levels and Nitrogen Dioxide levels. This is an interesting result as one would expect atmospheric pollutants to influence each other. Additionally, we see that the AR, ARDL, and VAR models for Carbon Monoxide are consistently better than those made for NO2. This maybe due to that fact that the Carbon Monoxide concentration has a more stationary looking series than that of Nitrogen Dioxide. For future research, we should try differencing the series for Nitrogen Dioxide for better predictions.