This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Since I am not a meterologist, I had to read about the different pollutants in the Madrid Air Quality file.

The four I looked at are:

1. O_3 - Is the Ozone that protects us from the sun
2. PM10 - Particle Pollution - PM10 is course an example - an example of dust being disturbed by cars driving down the road
3. NO_2 - Nitrogen dioxide - This forms the brown cloud in large cites. ** ***This will be the value used in forecasting*** **
4. SO_2 - Sulfur dioxide - We know this because of the oder

References links Boxplot https://www.rdocumentation.org/packages/reshape2/versions/1.4.3/topics/melt.data.frame pictures https://airnow.gov/index.cfm?action=pubs.aqguidepart EPA https://www.epa.gov/air-trends/particulate-matter-pm10-trends

---

Some of the packages are my normal ones. However, for this project, since I will be using the ARIMA model for forecasting, I needed to add a couple of new packages:

1. forecast
2. date
3. tseries
4. rio

```
install.packages("dplyr")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("tidyverse")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("readr")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("plyr")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("lubridate")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("ggplot2")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("reshape")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("data.table")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("sqldf")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("forecast")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("ddplry")
```

```
## Installing package into 'C:/Users/ep927/Documents/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## Warning in install.packages :
##   package 'ddplry' is not available (for R version 3.5.2)
```

```
install.packages("tidyr")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("rio")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("date")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("tseries")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("knitr")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("markdown")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("rmarkdown")
```

```
## Error in install.packages : Updating loaded packages
```

```
install.packages("devtools")
```

```
## Error in install.packages : Updating loaded packages
```

```
library(dplyr)
library(tidyverse)
library(readr)
library(plyr)
library(lubridate)
library(ggplot2)
library(reshape)
library(data.table)
library(sqldf)
library(forecast)
library(tidyr)
library(rio)
library(date)
library(tseries)
library(markdown)
library(rmarkdown)
library(knitr)
library(devtools)

getwd()
```

```
## [1] "C:/Users/ep927/Documents/Maydrid_Air_Quality"
```

The below code is where, I'm getting all 18 individual csv files into a single file call MadridSingleFile. I can now explore from a highlevel what the str looks like and get a statistical overview by running the summay.

```
#This segment of code is orgainzing the file and by using view it will show us what the data

filenames <- list.files(path = "./", pattern = "*.csv", full.names=TRUE)
MadridSingleFile <- ldply(filenames, read.csv)

MadridSingleFile<- data.frame(MadridSingleFile)
view(MadridSingleFile)

MadridSingleFile$NewDate <- strptime(MadridSingleFile$date, "%m/%d/%Y %H:%S")
MadridSingleFile$day <- day(MadridSingleFile$NewDate)
```

```
MadridSingleFile$month <- month(MadridSingleFile$NewDate)
MadridSingleFile$year <- year(MadridSingleFile$NewDate)
MadridSingleFile$hour <- hour(MadridSingleFile$NewDate)
MadridSingleFile$dates <- as.Date(MadridSingleFile$date, "%m/%d/%Y")
MadridSingleFile$NewDate <- as.character(MadridSingleFile$NewDate,format="%m/%d/%Y")
str(MadridSingleFile)
```

```
## 'data.frame':    3808248 obs. of  31 variables:
##  $ date     : Factor w/ 151896 levels "1/1/2001 1:00",..: 7298 7298 7298 7298 7298 7298 7
##  $ BEN      : num  NA 1.5 NA NA NA ...
##  $ CO       : num  0.37 0.34 0.28 0.47 0.39 ...
##  $ EBE      : num  NA 1.49 NA NA NA ...
##  $ MXY      : num  NA 4.1 NA NA NA ...
##  $ NMHC     : num  NA 0.07 NA NA NA ...
##  $ CH4      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ NO       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ NO_2     : num  58.4 56.2 50.7 69.8 22.8 ...
##  $ NOx      : num  87.2 75.2 61.4 73.4 24.8 ...
##  $ OXY      : num  NA 2.11 NA NA NA ...
##  $ O_3      : num  34.5 42.2 46.3 40.7 66.3 ...
##  $ PM10     : num  105 100.6 100.1 69.8 75.2 ...
##  $ PM25     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PXY      : num  NA 1.73 NA NA NA ...
##  $ SO_2     : num  6.34 8.11 7.85 6.46 8.8 ...
##  $ TCH      : num  NA 1.24 NA NA NA ...
##  $ TOL      : num  NA 10.8 NA NA NA ...
##  $ station  : int  28079001 28079035 28079003 28079004 28079039 28079006 28079007 280790
##  $ id       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ name     : Factor w/ 24 levels "Arturo Soria",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ address  : Factor w/ 24 levels " Pza. FernÃ¡ndez Ladreda - Avda. Oporto",..: NA NA NA
##  $ lon      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ lat      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ elevation: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ NewDate  : chr  "08/01/2001" "08/01/2001" "08/01/2001" "08/01/2001" ...
##  $ day      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ month    : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ year     : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
##  $ hour     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dates    : Date, format: "2001-08-01" "2001-08-01" ...
```

```
view(MadridSingleFile)
summary(MadridSingleFile)
```

```
##                date              BEN                CO
##  1/1/2004 0:00  :     28   Min.   : 0.0      Min.   : 0.0
##  10/1/2003 0:00 :     28   1st Qu.: 0.2      1st Qu.: 0.3
##  10/1/2003 1:00 :     28   Median : 0.6      Median : 0.4
```

```
##  10/1/2003 10:00:     28   Mean   : 1.3      Mean   : 0.6
##  10/1/2003 11:00:     28   3rd Qu.: 1.5      3rd Qu.: 0.6
##  (Other)        :3808084   Max.   :66.4      Max.   :18.0
##  NA's           :     24   NA's   :2766564   NA's   :1157236
##       EBE               MXY              NMHC              CH4
##  Min.   :  0.0     Min.   :  0       Min.   :0.0       Min.   :0
##  1st Qu.:  0.3     1st Qu.:  1       1st Qu.:0.1       1st Qu.:1
##  Median :  0.9     Median :  3       Median :0.2       Median :1
##  Mean   :  1.4     Mean   :  5       Mean   :0.2       Mean   :1
##  3rd Qu.:  1.6     3rd Qu.:  6       3rd Qu.:0.2       3rd Qu.:1
##  Max.   :162.2     Max.   :178       Max.   :9.1       Max.   :4
##  NA's   :2806524   NA's   :3492833   NA's   :2722936   NA's   :3799808
##        NO               NO_2              NOx               OXY
##  Min.   :   0.0    Min.   :  0.00   Min.   :   0.0    Min.   :  0
##  1st Qu.:   2.0    1st Qu.: 24.00   1st Qu.:  40.0    1st Qu.:  1
##  Median :   6.0    Median : 44.00   Median :  76.2    Median :  1
##  Mean   :  23.4    Mean   : 50.47   Mean   : 109.3    Mean   :  2
##  3rd Qu.:  20.0    3rd Qu.: 69.58   3rd Qu.: 139.7    3rd Qu.:  3
##  Max.   :1146.0    Max.   :628.60   Max.   :2537.0    Max.   :103
##  NA's   :2275851   NA's   :21198    NA's   :1431973   NA's   :3492553
##       O_3              PM10             PM25              PXY
##  Min.   :  0.0    Min.   :  0.0    Min.   :-31.0     Min.   :  0
##  1st Qu.: 12.7    1st Qu.: 11.5    1st Qu.:  6.4     1st Qu.:  1
##  Median : 34.9    Median : 21.5    Median : 11.0     Median :  1
##  Mean   : 39.8    Mean   : 28.9    Mean   : 13.7     Mean   :  2
##  3rd Qu.: 60.0    3rd Qu.: 37.8    3rd Qu.: 17.7     3rd Qu.:  3
##  Max.   :236.0    Max.   :695.0    Max.   :506.9     Max.   :106
##  NA's   :816516   NA's   :946993   NA's   :2991824   NA's   :3492664
##       SO_2              TCH              TOL              station
##  Min.   :  0.0     Min.   : 0.0      Min.   :  0.0     Min.   :28079001
##  1st Qu.:  5.8     1st Qu.: 1.3      1st Qu.:  1.1     1st Qu.:28079014
##  Median :  8.1     Median : 1.4      Median :  3.2     Median :28079024
##  Mean   : 10.7     Mean   : 1.4      Mean   :  5.9     Mean   :28079029
##  3rd Qu.: 12.3     3rd Qu.: 1.5      3rd Qu.:  7.0     3rd Qu.:28079040
##  Max.   :199.1     Max.   :10.5      Max.   :242.9     Max.   :28079099
##  NA's   :1032288   NA's   :2721807   NA's   :2769319   NA's   :24
##        id                                  name
##  Min.   :28079004   Arturo Soria        :      1
##  1st Qu.:28079022   Avda. RamÃ³n y Cajal:      1
##  Median :28079040   Barajas Pueblo      :      1
##  Mean   :28079038   Barrio del Pilar    :      1
##  3rd Qu.:28079054   Casa de Campo       :      1
##  Max.   :28079060   (Other)             :     19
##  NA's   :3808224    NA's                :3808224
##                                          address             lon
##   Pza. FernÃ¡ndez Ladreda - Avda. Oporto     :      1   Min.   :-4
```

```
##  Avd. Betanzos esq. C/  Monforte de Lemos    :      1   1st Qu.:-4
##  Avd. Moratalaz  esq. Camino de los Vinateros:      1   Median :-4
##  Avda La Gavia / Avda. Las Suertes           :      1   Mean   :-4
##  Avda. La Guardia                            :      1   3rd Qu.:-4
##  (Other)                                     :     19   Max.   :-4
##  NA's                                        :3808224   NA's   :3808224
##       lat           elevation         NewDate               day
##  Min.   :40       Min.   :599      Length:3808248     Min.   : 1.00
##  1st Qu.:40       1st Qu.:626      Class :character   1st Qu.: 8.00
##  Median :40       Median :661      Mode  :character   Median :16.00
##  Mean   :40       Mean   :658                         Mean   :15.72
##  3rd Qu.:40       3rd Qu.:687                         3rd Qu.:23.00
##  Max.   :41       Max.   :728                         Max.   :31.00
##  NA's   :3808224   NA's   :3808224                    NA's   :24
##      month             year           hour           dates
##  Min.   : 1.000   Min.   :2001   Min.   : 0.00   Min.   :2001-01-01
##  1st Qu.: 3.000   1st Qu.:2005   1st Qu.: 5.75   1st Qu.:2005-02-10
##  Median : 6.000   Median :2009   Median :11.50   Median :2009-04-11
##  Mean   : 6.445   Mean   :2009   Mean   :11.50   Mean   :2009-06-20
##  3rd Qu.: 9.000   3rd Qu.:2013   3rd Qu.:17.25   3rd Qu.:2013-10-17
##  Max.   :12.000   Max.   :2018   Max.   :23.00   Max.   :2018-05-01
##  NA's   :24       NA's   :24     NA's   :24      NA's   :24
```

```
#After added the new fields the below commands remove unnecessary columns 20-26, and remove

MadridSingleFile[20:26] <- list(NULL)
MadridSingleFile  <- slice(MadridSingleFile, 1:(n()-24))
view(MadridSingleFile)
```

This is the start of the Exploratory Data Analysis*

I am pulling out the four most common pollutants that are consider by the government as the most harmful to humans. There are two things going on here with the code below.

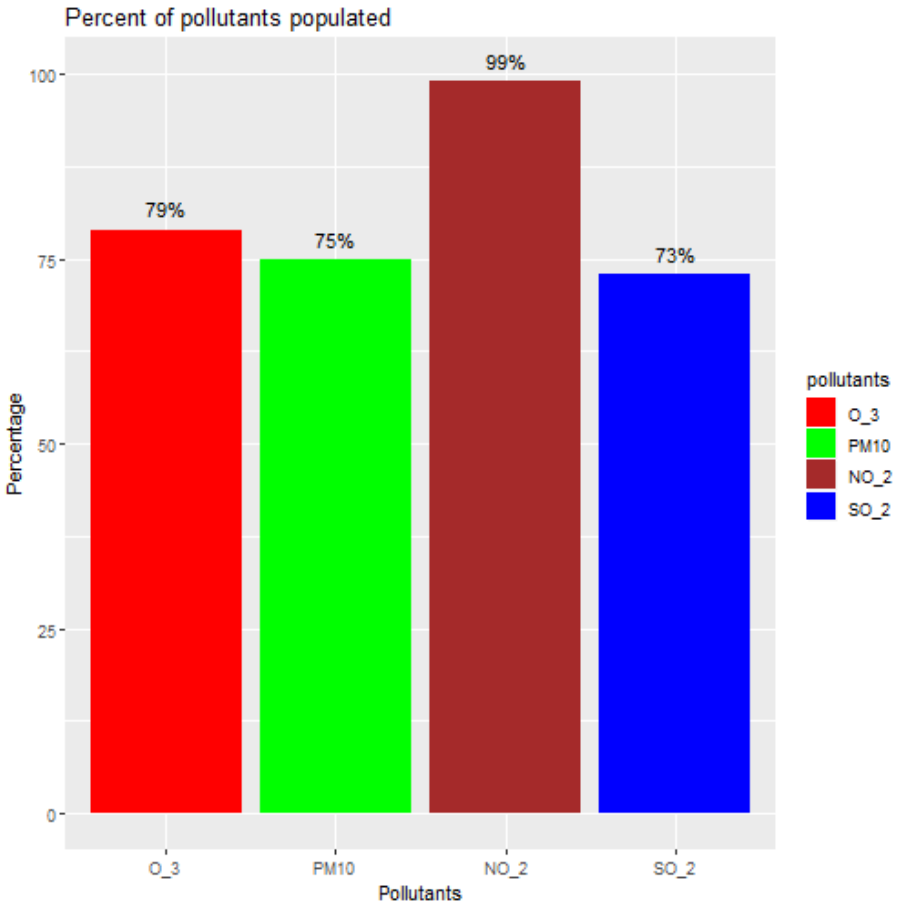
1. I am pulling out only the four pollutants that I want to look at
2. Showing how often the data is populated. NO_2 Nitrogen dioxide is the most populated.

```
#I am starting to look at the top four pollutants.

FinalFourPollutants <-MadridSingleFile[,c('O_3','PM10','NO_2','SO_2')]

summary(FinalFourPollutants)
```

```
##       O_3              PM10              NO_2              SO_2
##  Min.   :  0.0    Min.   :  0.0    Min.   :  0.00   Min.   :  0.0
```

```
##  1st Qu.: 12.7    1st Qu.: 11.5    1st Qu.: 24.00   1st Qu.:  5.8
##  Median : 34.9    Median : 21.5    Median : 44.00   Median :  8.1
##  Mean   : 39.8    Mean   : 28.9    Mean   : 50.47   Mean   : 10.7
##  3rd Qu.: 60.0    3rd Qu.: 37.8    3rd Qu.: 69.58   3rd Qu.: 12.3
##  Max.   :236.0    Max.   :695.0    Max.   :628.60   Max.   :199.1
##  NA's   :816492   NA's   :946969   NA's   :21174    NA's   :1032264
```

```
NotNAs<-data.frame(percent=round(colSums(!is.na(FinalFourPollutants))/nrow(FinalFourPollutar
NotNAs$poll <- rownames(NotNAs)
NotNAs$pollutants<-factor(NotNAs$poll, as.character(NotNAs$poll))

ggplot(NotNAs, aes(pollutants, percent,fill=pollutants))+
geom_bar(stat="identity") +scale_fill_manual(values = c("red","green","brown","blue"))+

geom_text(data=NotNAs, aes(label=paste0(percent,"%"),
                           y=percent+0.9), size=4, vjust = -.4)+
 labs(x = "Pollutants", y = "Percentage",
        title = "Percent of pollutants populated")
```



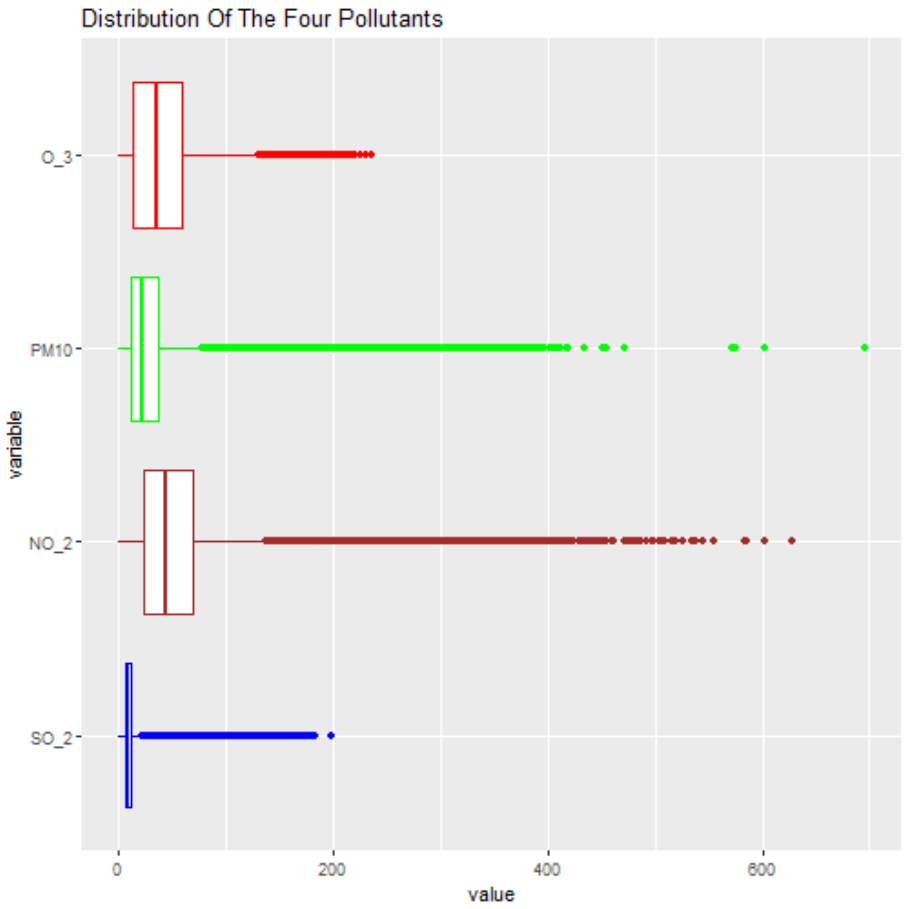
The code below are box plots taht are turned on their sides. To make this happen, I started with this code

Pollutantnames<-c("SO_2", "NO_2", "PM10", "O_3") BP <- boxplot(FinalFourPollutants$O_3,FinalFourPollutants$PM10,FinalFourPollutants$NO_2,FinalFourPollutants$SO_ names = Pollutantnames)

Which if run will only produce a black and white plot. It took me awhile to get the items to line up.

```
#In order to keep the same order as in the above code I had to reverse the order of the
#polluntants, since I was flipping the barchart. I also had to make sure the colors were
#in the correct order too.

boxplot_melt <- melt(MadridSingleFile,id.vars='station', measure.vars=c("SO_2", "NO_2", "PM
createBoxPlot4Pollutants<-ggplot(na.omit(boxplot_melt),aes(x=variable,y=value, color=variabl
      geom_boxplot()+ coord_flip()+
      scale_colour_manual(values=c("blue","brown","green","red"))+
```

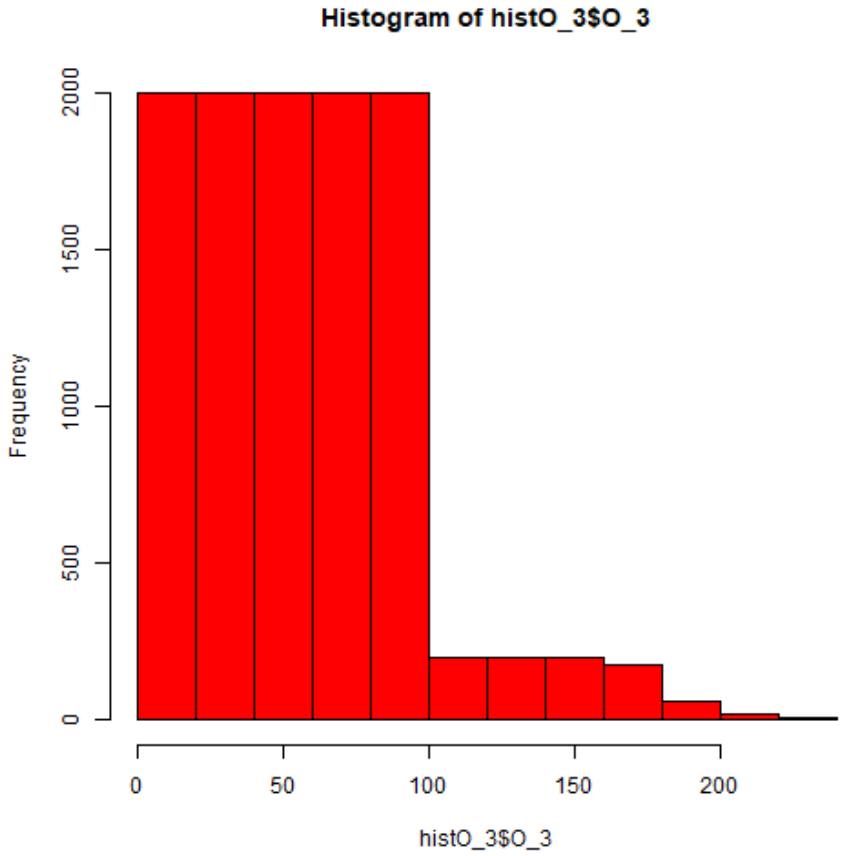
```
      theme(legend.position="none")+
      labs(title="Distribution Of The Four Pollutants")
plot(createBoxPlot4Pollutants)
```



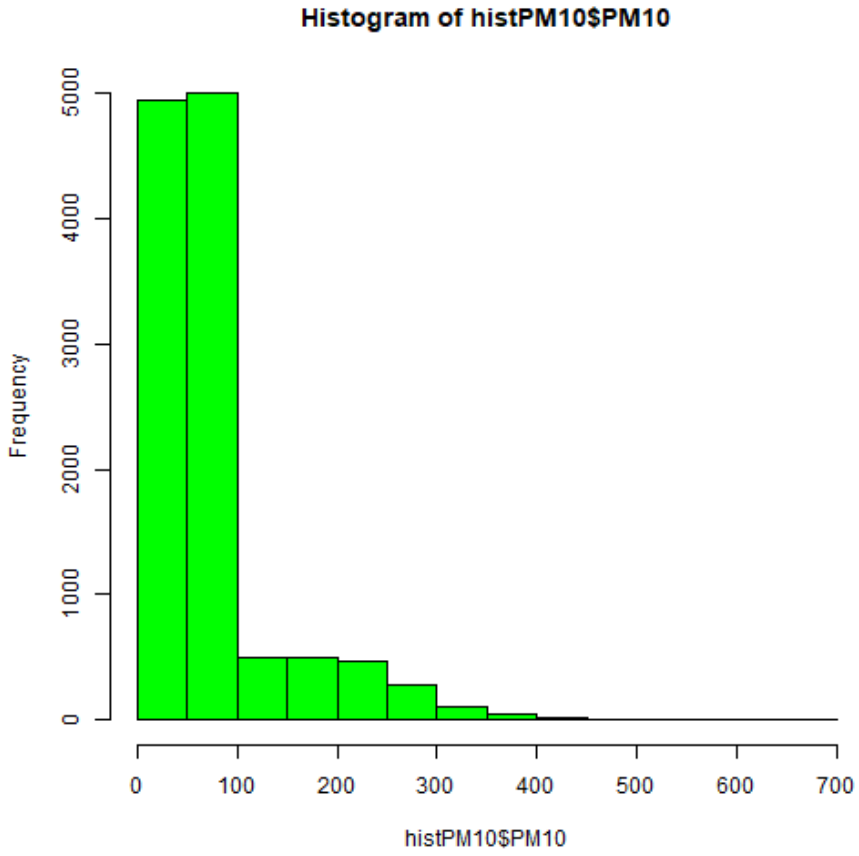
Below are the histograms for the four pollutants. I used sqldf to extract the data. Once again I used the same colors for the polluntants. This makes it easier to track of them.

```
# The first thing was to extract the data using sqldf


histO_3 <-sqldf("select O_3, count(*)
                from MadridSingleFile
                where O_3 != 'NA'
                group by O_3")

histPM10 <-sqldf("select PM10, count(*)
```

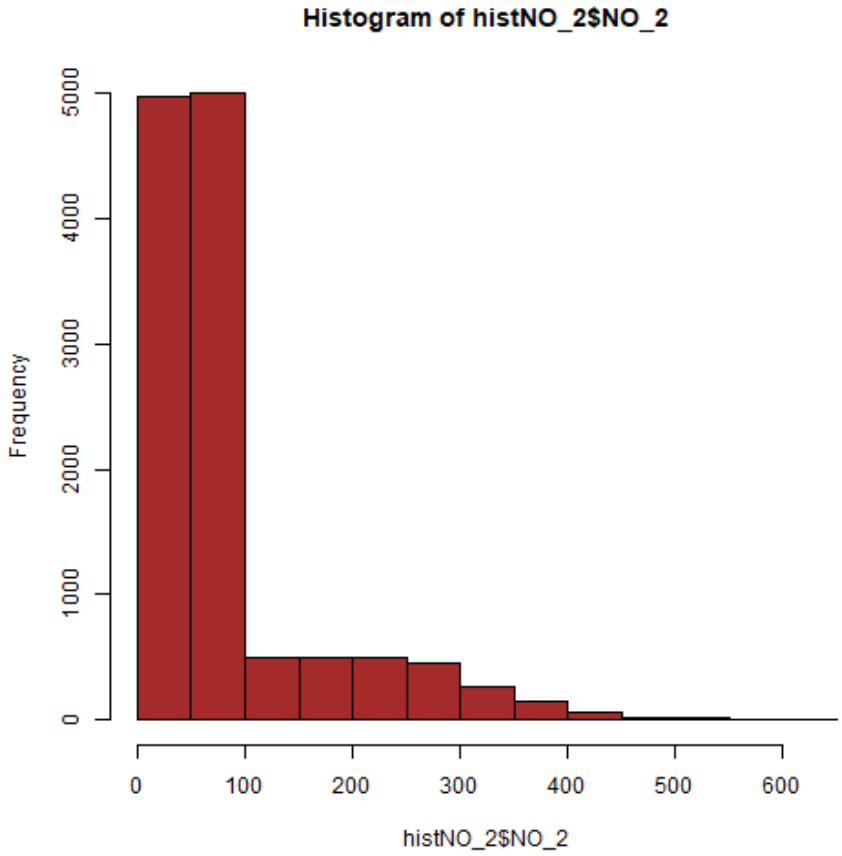
```
                from MadridSingleFile
                where PM10 !='NA'
                group by PM10")

histNO_2 <-sqldf("select NO_2, count(*)
                from MadridSingleFile
                where NO_2 !='NA'
                group by NO_2")

histSO_2 <-sqldf("select SO_2, count(*)
                from MadridSingleFile
                where SO_2 !='NA'
                group by SO_2")

#Step 2 plot the histograms.

hist(histO_3$O_3, plot=TRUE, col="red")
```



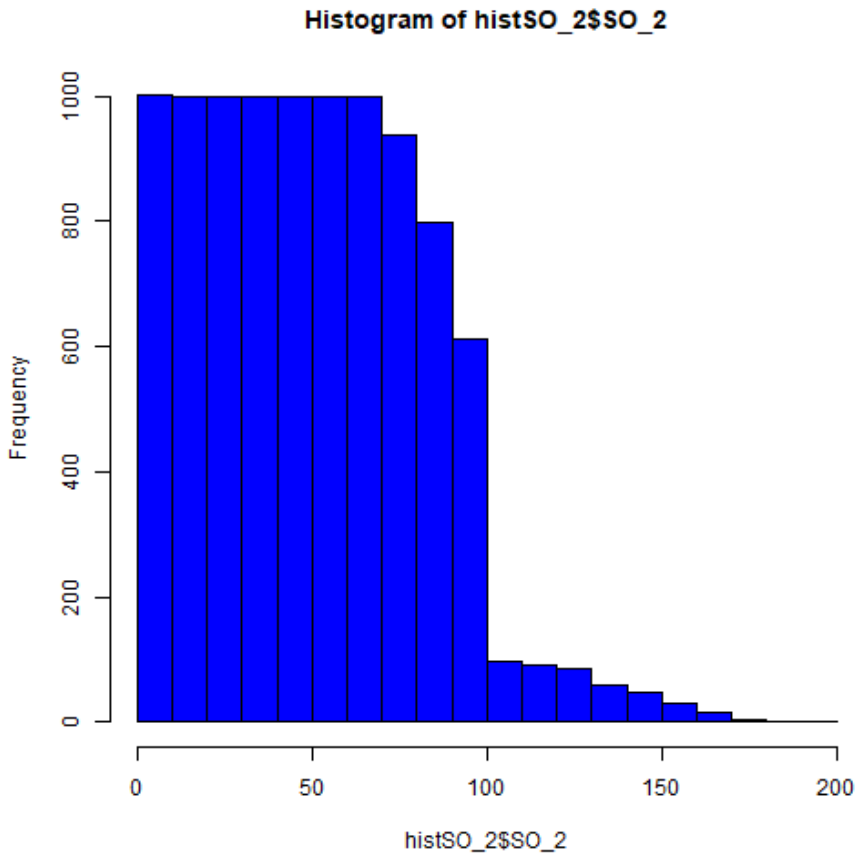

```
hist(histPM10$PM10, plot=TRUE,col="green")
```




```
hist(histNO_2$NO_2, plot=TRUE, col="brown")
```



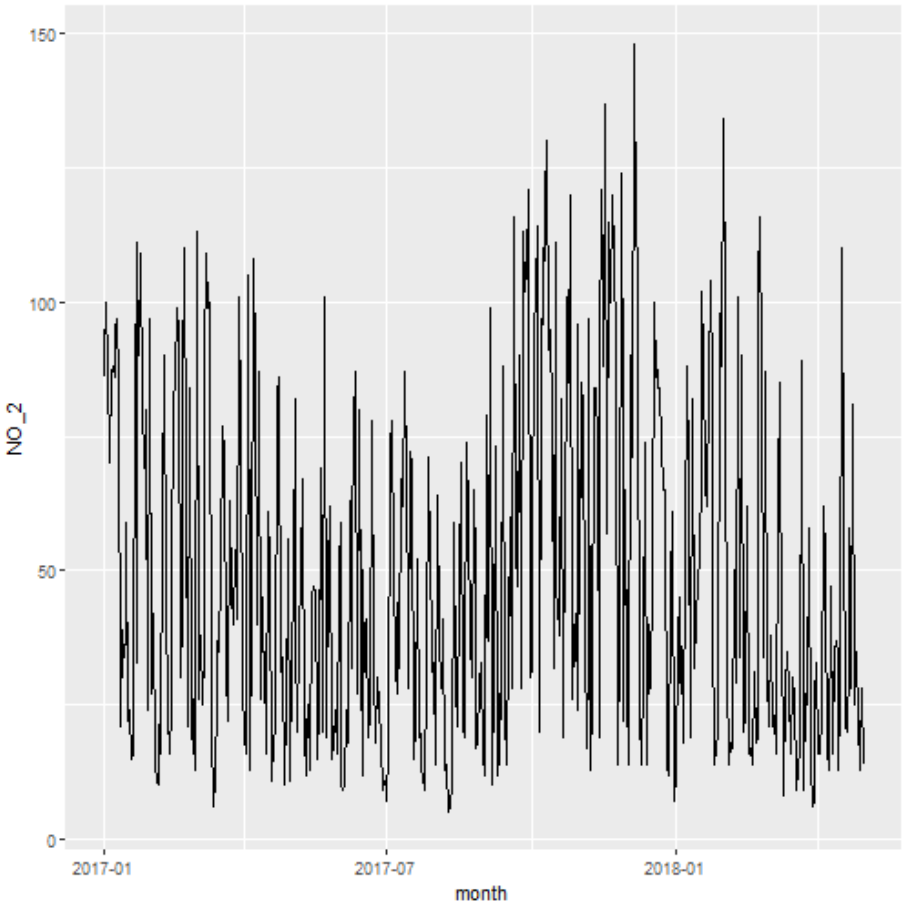

```
hist(histSO_2$SO_2, plot=TRUE,col="blue")
```



The remaining code is the time series for NO_2. So I using sqldf, I selected years greater than 2016. So, I will be using 2017-2018 to forecast 60 days out for NO_2 (Nitrogen dioxide). These same steps could be used for any timeseries, including the other pollutants.

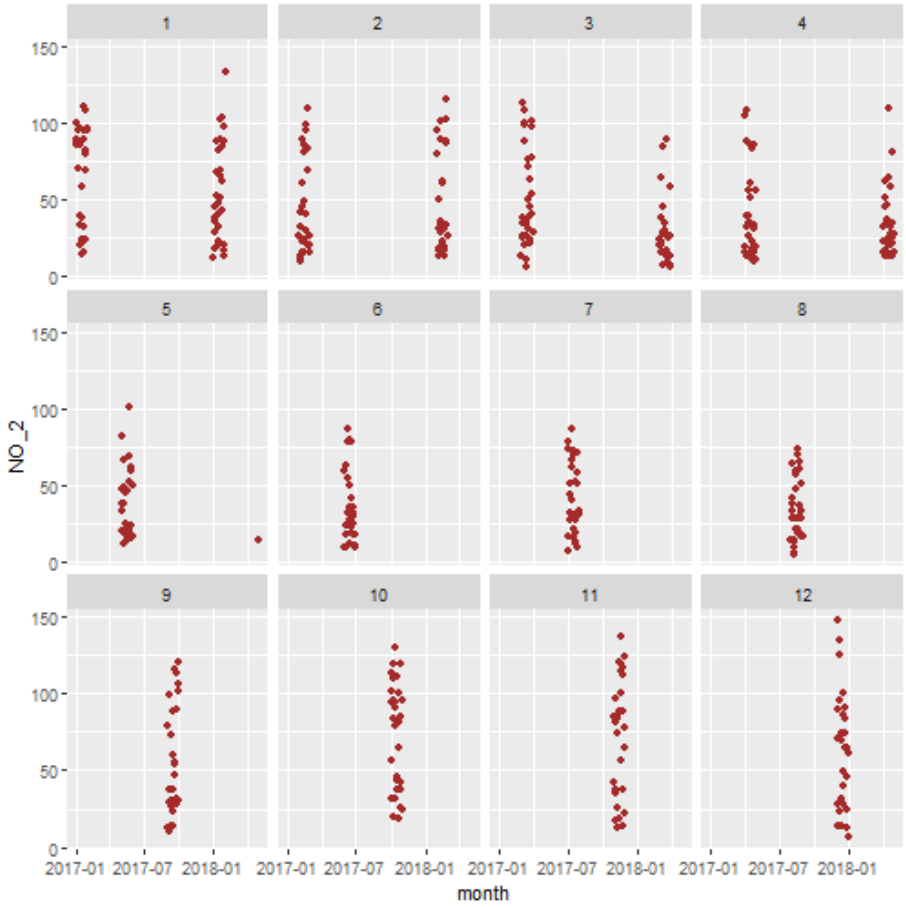
```
# Step 1.  Need to get NO_2 into a file called group_ts.  I selected the year > 2016 and whe
#NO_2 value was less than 150.  I did this based ont the EDA

group_ts <- sqldf("select *
                  from MadridSingleFile
                  where year > 2016 and (NO_2) < 150
                  group by day, month, year")


#Step 2.  Plot the graph by month

ggplot(group_ts, aes(dates,NO_2)) + geom_line()+
```

```
  scale_x_date('month')
```



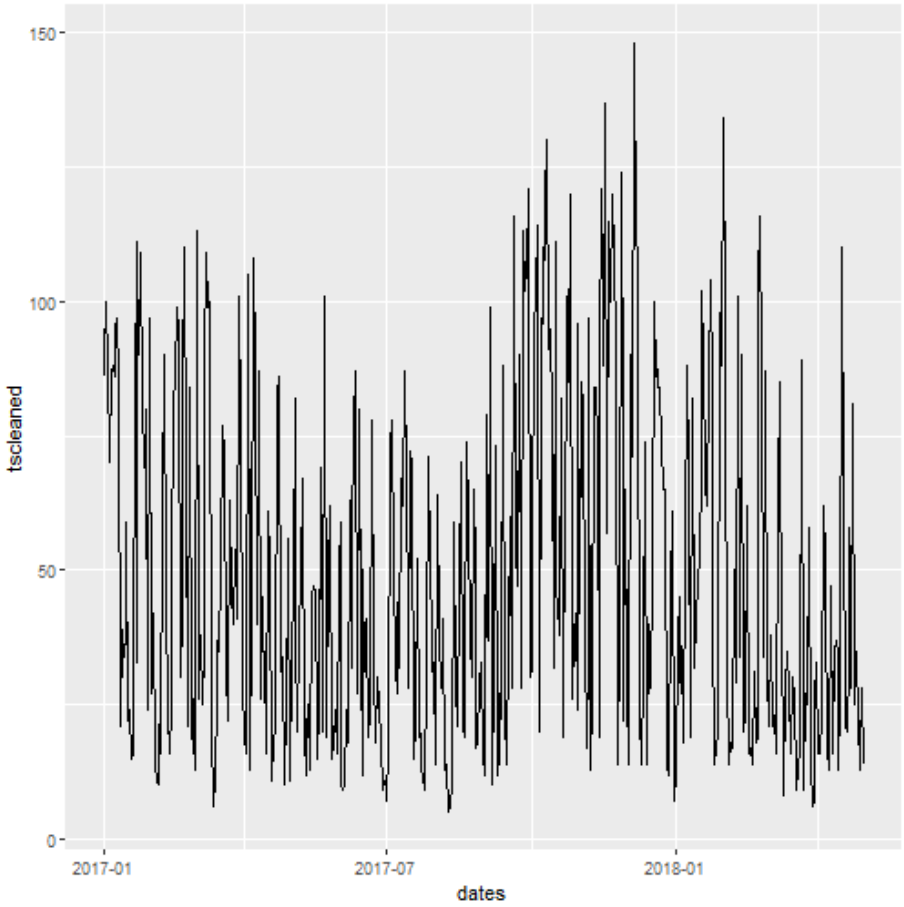
Step 2. I want to look at the data using the facet_wrap command by month.
This will show us if there are diffent years with data in the same month.
I did find this an intresting plot.

```
# Plotting NO_2 by month.  We can see that we have 2017 and 2018 through the first four mont
#the data.

ggplot(group_ts, aes(dates,NO_2)) + geom_point(color = "brown") +
  facet_wrap( ~month) + scale_x_date('month')
```



Step 3. We need to create a times series object.

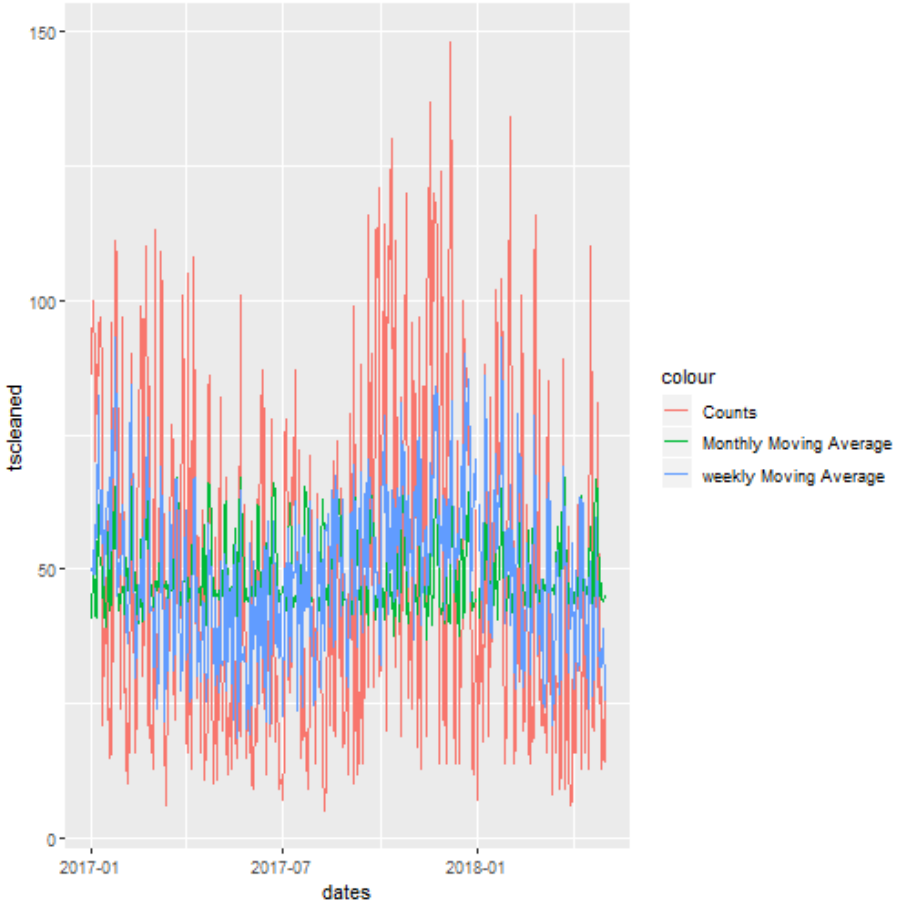
```
# I need to create a time series object. I will be using the group_ts create above for this.

NO_2.tsobject = ts(group_ts[, c("NO_2")])

# Now I will use the tsclean command to clean up the NO_2 data.
group_ts$tscleaned = tsclean(NO_2.tsobject)

#Plot the data.
ggplot() +
  geom_line(data = group_ts, aes( x= dates, y= tscleaned))

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuou
```



Step 4. Now I am going to look at the moving average for weekly and the 30. Plot them with the count to help determine which value will be used. It's hard to see, I know, the green (Monthly moving average) doesn't have a lot of variance. So I decided to us the weekly moving average. There is still quite a bit of variance. Depending on what you're looking at this can help decided what you want.

```
#Here is where I get the moving average for weekly and monthly for NO_2 using the ma functio

group_ts$NO_2.mavg7 = ma(group_ts$tscleaned,order = 7)
group_ts$NO_2.mavg30 = ma(group_ts$tscleaned,order = 30)

ggplot() +
  geom_line(data = group_ts, aes(x = dates, y = tscleaned, colour = "Counts")) +
  geom_line(data = group_ts, aes(x = dates, y = NO_2.mavg30, colour = "Monthly Moving Averag
  geom_line(data = group_ts, aes(x = dates, y = NO_2.mavg7, colour = "weekly Moving Average'

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuou
```

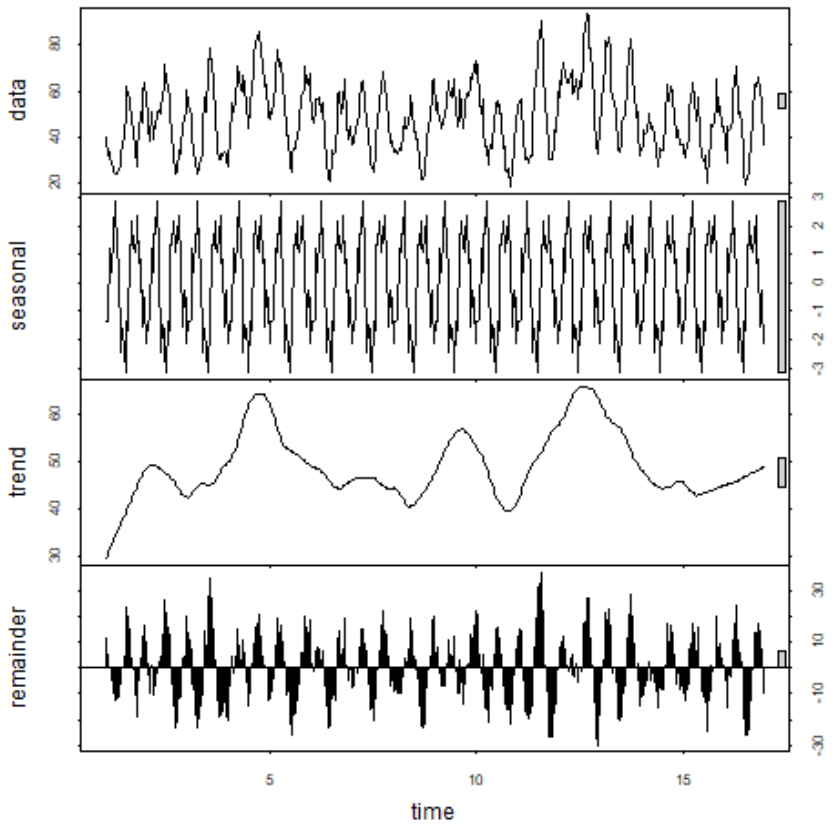
```
## Warning: Removed 2 rows containing missing values (geom_path).
## Warning: Removed 1 rows containing missing values (geom_path).
```



Step 5. Now I am going to decompose the data that will be used in the ARIMA forecasting

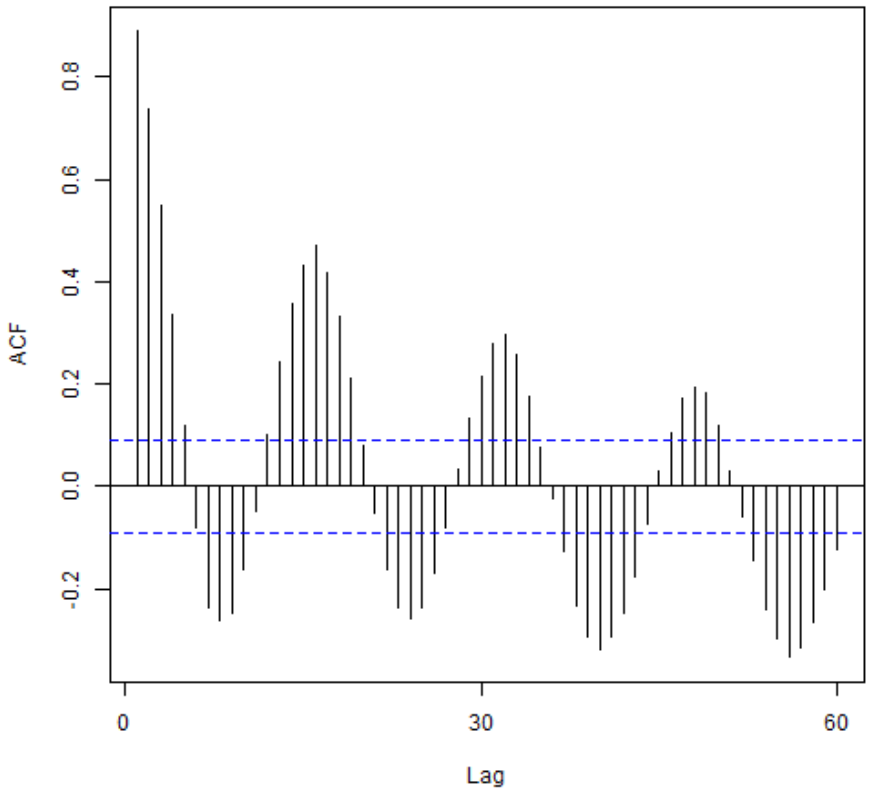
There is a lot going on in this next bit of code. 1) Need to remove an na's from NO_2. 2) Going to remove seasonality. 3) decompose it and then graph it.

```
count_ma = ts(na.omit(group_ts$NO_2.mavg7), frequency = 30)
decomp = stl(count_ma, s.window = "periodic")
rmvseasonal.NO_2 <- seasadj(decomp) # decomp will be use later on.

plot(decomp)
```

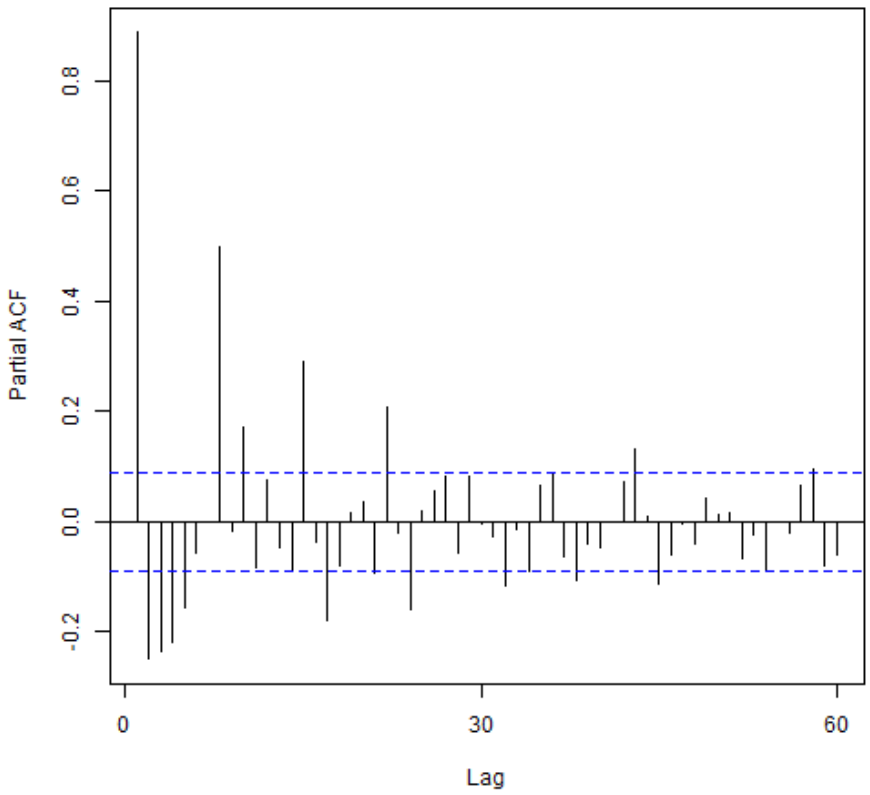



```
# Notice the data is much cleaner than it was before because
# I'm using a weekly moving average. The plot just shows what the
#code is doing
######## The NO_2 data is NOT stationary.  It's going up and down

########################################

#Need to run a Augmented Dickey-Fuller test using the adf function.
#whats key here is the lag order, which will be used later on. The
#other important thing here is the more negative they number the
#more accurate the model will be.  In this case it's -4.861
#will see if I can improve on that number.


adf.test(count_ma, alternative = "stationary")

## Warning in adf.test(count_ma, alternative = "stationary"): p-value smaller
```

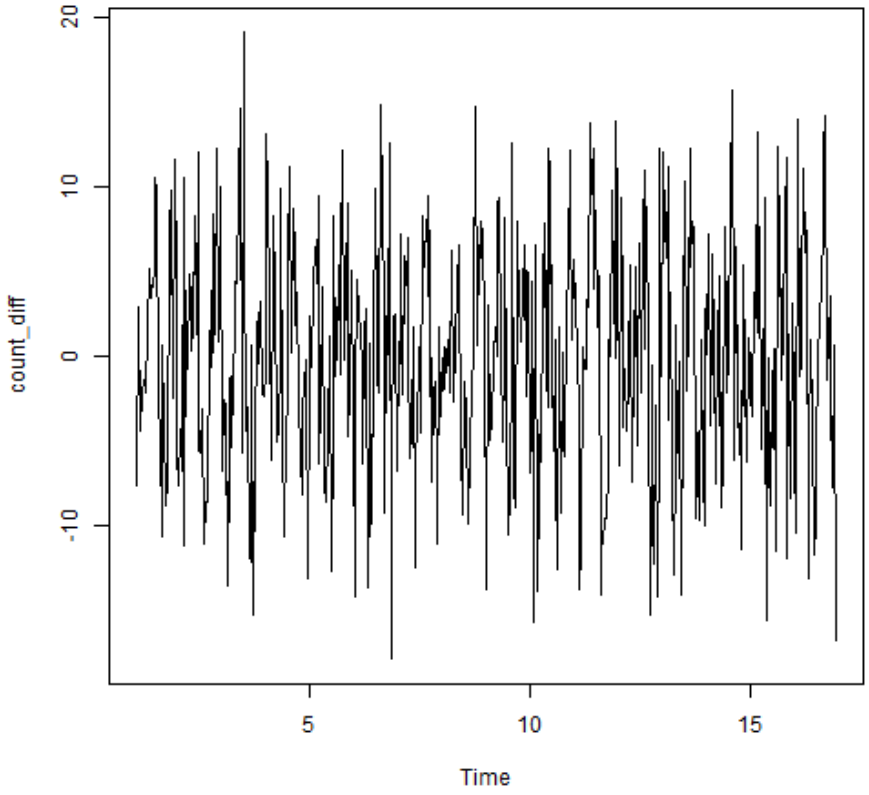
```
## than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  count_ma
## Dickey-Fuller = -4.861, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary

# The functions below check the the correlations between
# the series and its lag.
Acf(count_ma, main= "")
```




```
Pacf(count_ma, main = "")
```




```
#Here I'm seeing how close I can get the difference.  Depending on
#what you want, you can change the differences, I'v decided to use 1

count_diff = diff(rmvseasonal.NO_2, differences = 1)
plot(count_diff)
```

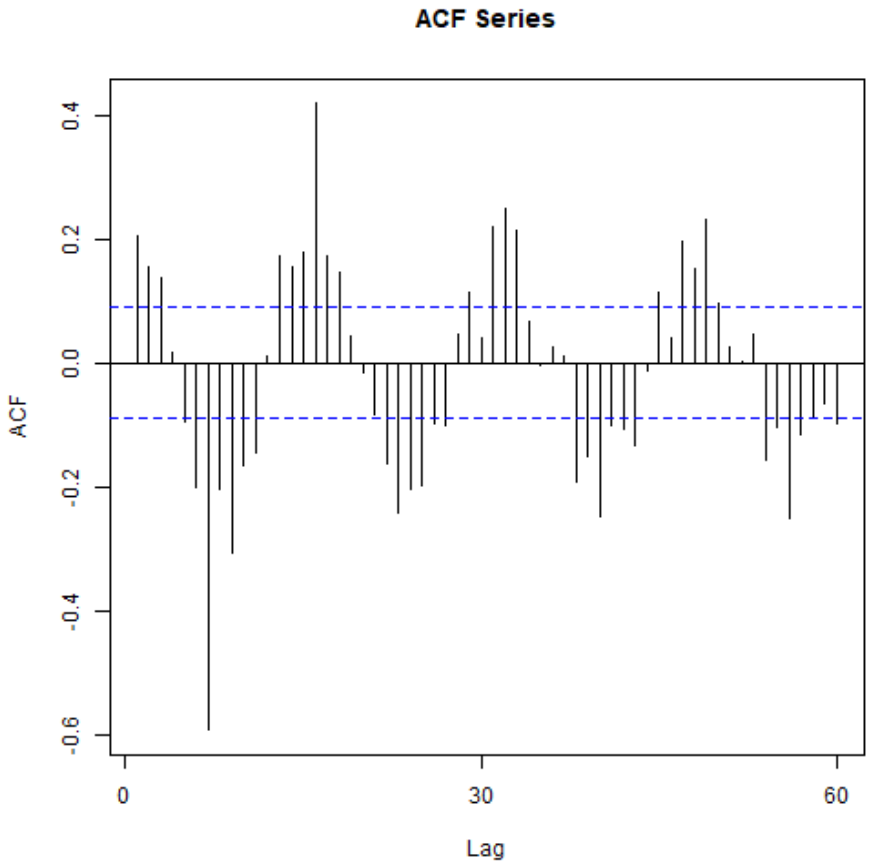



```
adf.test(count_diff, alternative = "stationary")
```

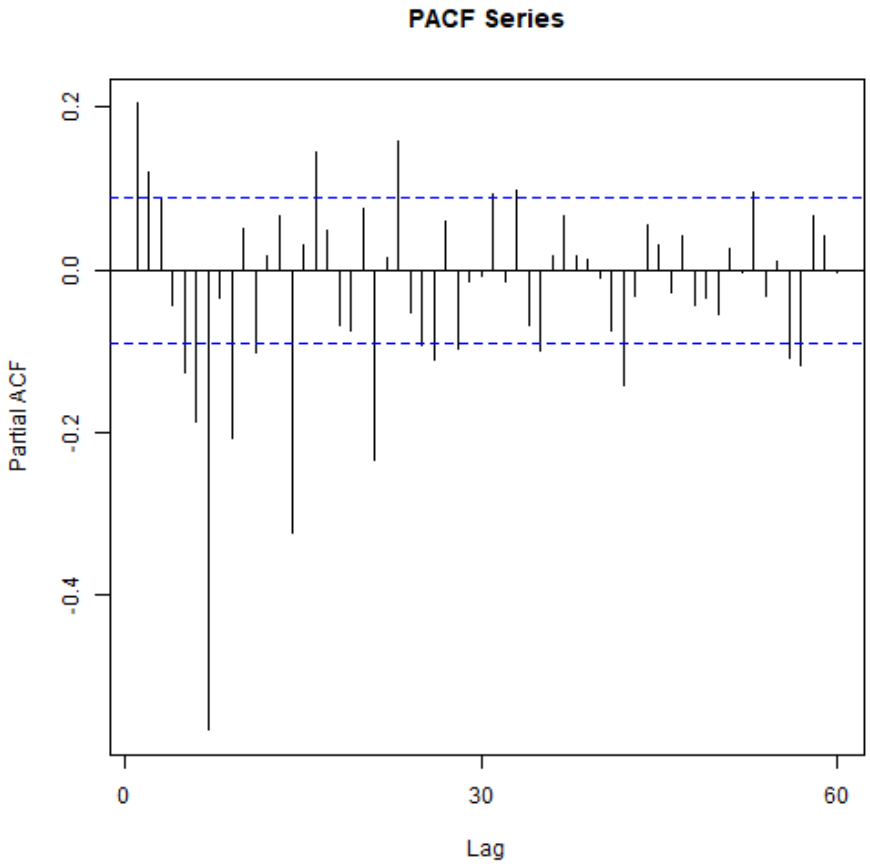
```
## Warning in adf.test(count_diff, alternative = "stationary"): p-value
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  count_diff
## Dickey-Fuller = -14.374, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
Acf(count_diff, main = "ACF Series")
```




```
Pacf(count_diff, main = "PACF Series")
```



########################## FITTING THE MODEL ####################

Step 1.

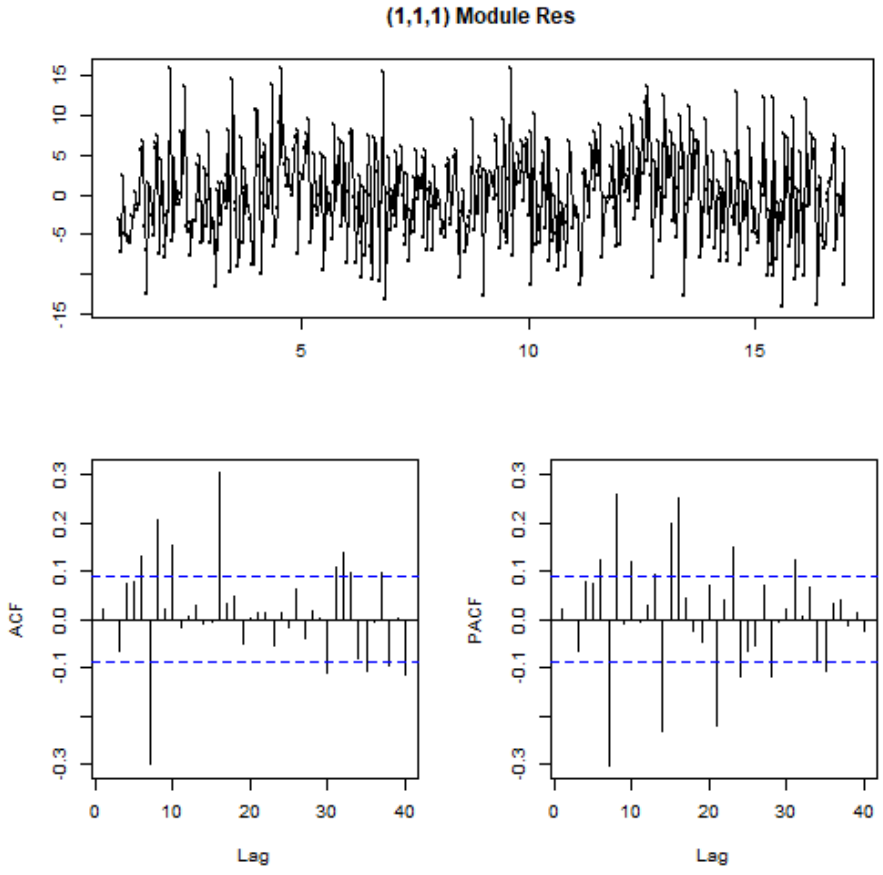
```
#Fitting the ARMIA model
#Here I'm getting the p,d,q values. By using the auto.arima
#I want to see what values it brings back for the p,d,q values.

auto.arima(rmvseasonal.NO_2, seasonal = FALSE)
```

```
## Series: rmvseasonal.NO_2
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3     mean
##       1.6770  -0.8209  -0.7891  0.1829  0.3032  48.9180
## s.e.  0.0484   0.0437   0.0585  0.0528  0.0414   1.2717
```

```
##
## sigma^2 estimated as 33.66:  log likelihood=-1523.43
## AIC=3060.85   AICc=3061.09   BIC=3090.07

#It brings back p,d,q values of 2,0,3.
```

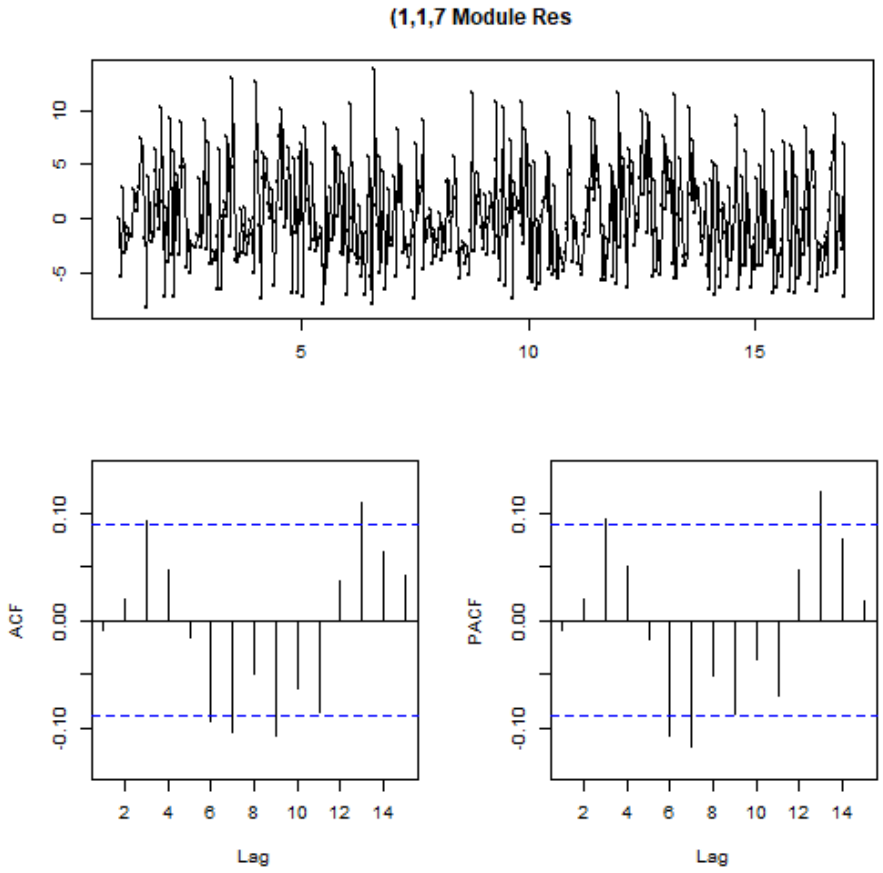
Step 2. fitting the model to see what the default p,d,q values of (1,1,1) and will compare that with the values I chose of (1,1,7) based on the previous values from above.

```
#default values to make see how the model looks
#The lag.max needs to be enough to see how it looks, get enough data.
fit <- auto.arima(rmvseasonal.NO_2, seasonal = FALSE)
tsdisplay(residuals(fit), lag.max = 40, main = "(1,1,1) Module Res")
```




```
#values I choose or 1,1,7.  This is a non auto.arima.
#Even though my from the adf, it recommended a lag of 7, I chose 8
#it was a better fit.
```

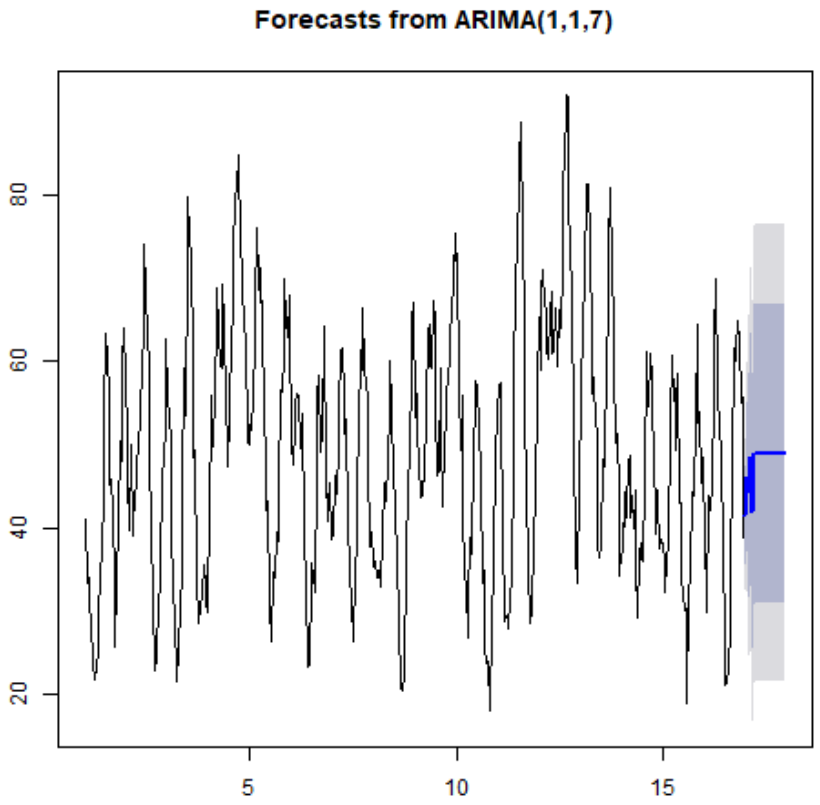
```
fit2 <- arima(rmvseasonal.NO_2, order = c(1,1,7))
tsdisplay(residuals(fit2), lag.max = 15, main = "(1,1,7 Module Res")
```



################ Forecasting ##############

I will be using the fit2 model from above. I decided to use the values for d,p, q (1,1,7) as the model showed

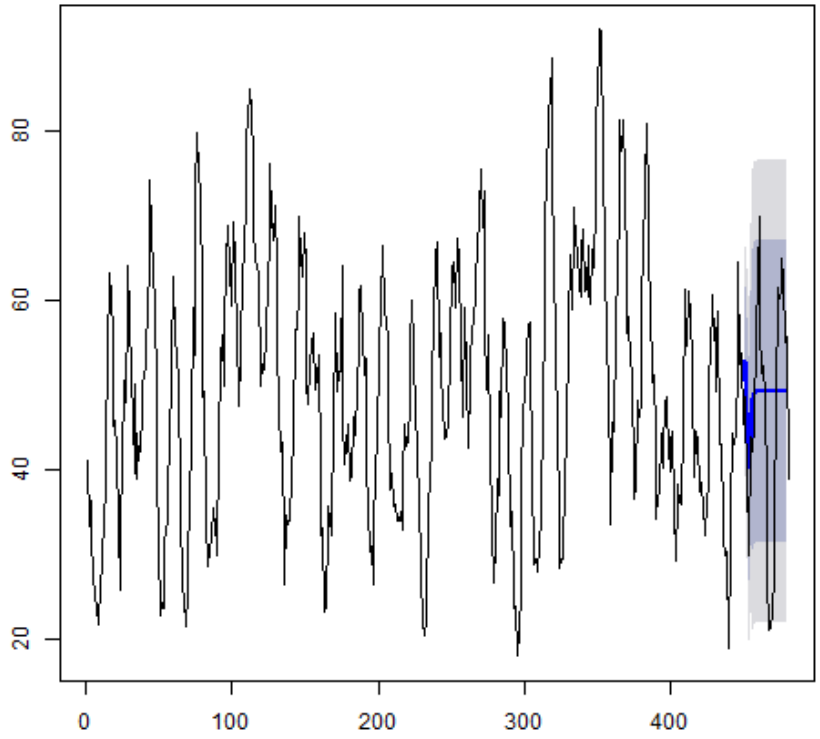
```
# Here I'm usqing the fit2 model, which is a non auto.arima.
# becasue I ordered the d,p,q vlaues.
# h=30, is 30 days.
fcast <- forecast(fit2, h=30)
plot(fcast)
```



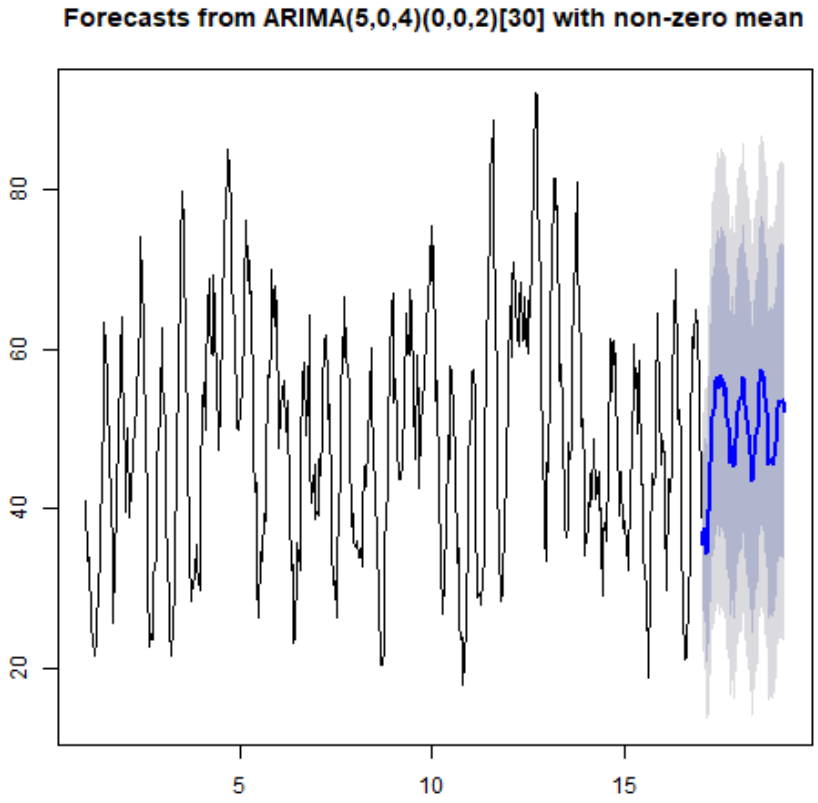

```
#the plot shows a straight line in the forecast area.  this will be
#fixed in the steps below.
```

Step. 2 I going to take a subset of the data and compare between the holdout and none hold out values. I will doing 30 days, and this will help me to understand if I need to bring back the seasonality to make the predicition more accurate allong with the forecast.

```
#since this is a subset ofd data, it will come from the rmvseasonal values
#this will be 30 days.  from 450 to 480.  which was created earlier.
hold <- window(ts(rmvseasonal.NO_2), start = 450)
fit_nohold = arima(ts(rmvseasonal.NO_2[-c(450:480)]), order=c(1,1,7))
fcast_nohold <- forecast(fit_nohold, h = 30)
plot(fcast_nohold, main= "")
lines(ts(rmvseasonal.NO_2))
```




```
fit_with_seasons = auto.arima(rmvseasonal.NO_2, seasonal = TRUE)
seasons.forecast <- forecast(fit_with_seasons,h=65)
plot(seasons.forecast)
```




```
#Forecasting out 65 days shows there is a high trust in the model
#the forecast is well with in the 95% which is the darker grey.
```