Assignment#1

#install.packages('tidyverse')  
library(tidyverse)

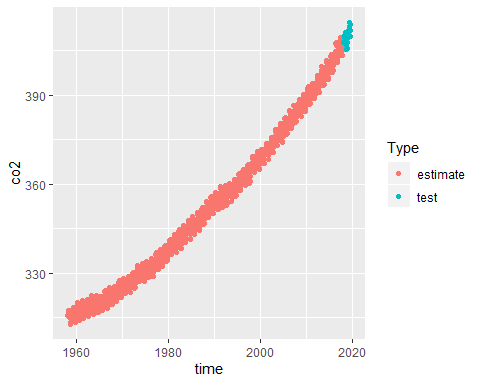
## -- Attaching packages --------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

Here, the data set was imported. Then, the dataset was divided as the assignment requested and and the estimate data and the test data in two different color has been plotted.

DATA <- read.csv('A1\_co2.txt',sep = '')  
  
DATA\_est <- DATA %>%   
 filter(year !=2019 & year !=2018) %>%   
 mutate(Type='estimate')  
#tail(DATA\_estimate)  
DATA\_test <- DATA %>%   
 filter(year ==2019 | year ==2018) %>%   
 mutate(Type='test')  
  
DATA\_est\_tes <- bind\_rows(DATA\_est,DATA\_test)  
  
#  
ggplot(data = DATA\_est\_tes,aes(x=time,y=co2)) +  
 geom\_jitter(aes(colour = Type))



# tail(Data)

The lm function was used here in order to perform the Ordinary Least Sqaure based on the the training data set provide in the previous section.

p <- 365  
#x <- cbind(1,DATA\_est$time,sin(2\*pi\*DATA\_est$time/p),cos(2\*pi\*DATA\_est$time/p))  
  
lm1 <- lm(co2 ~ time + I(sin(2\*pi\*time/p)) + I(cos(2\*pi\*time/p)), data=DATA\_est) ## Notice the I(...) to get what is calculated instead of interaction

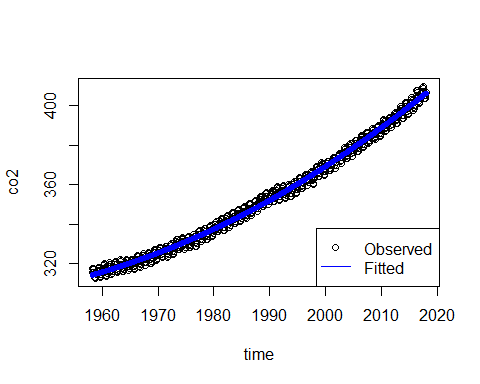
Here, the estimated parameters and as well the standard deviation of each parameters (4 in this case) were estimated.

sum\_param=summary(lm1)$coefficient  
row.names(sum\_param) <- c('alpha','beta\_t','beta\_s','beta\_c')  
sum\_param

## Estimate Std. Error t value Pr(>|t|)  
## alpha -2649.759905 814.0681612 -3.254961 1.187573e-03  
## beta\_t 1.551843 0.4094726 3.789859 1.634528e-04  
## beta\_s -27.579930 23.0858185 -1.194670 2.326128e-01  
## beta\_c 81.721452 8.2633543 9.889622 1.062263e-21

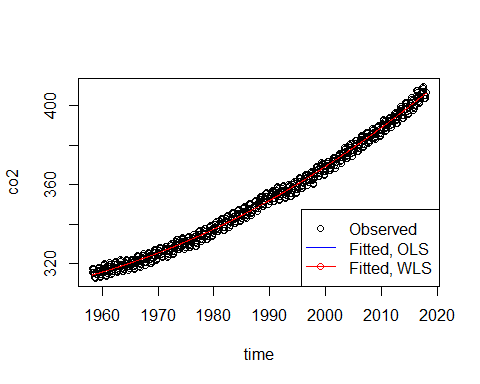
Plot of the data Vs. Fitted values based on Ordinary Least Sqaure:

DATA\_est$fit <- lm1$fitted.values  
plot(co2~time, DATA\_est)  
lines(fit~time, DATA\_est, type="l", lwd=5, col = "blue")  
legend("bottomright", legend = c("Observed","Fitted"), col = c('black','blue'), pch=c(1,NA), lty=c(NA,1))



here, the algorithem was provided in order to ro estimate the rho values after 5 iterations. At the end, the plot of the WLS method based on the estimated paramters (after 5 iteraions) were plotted:

x <- cbind(1,DATA\_est$time,sin(2\*pi\*DATA\_est$time/p),cos(2\*pi\*DATA\_est$time/p))  
sigma <- diag(718)  
Y <- DATA\_est$co2   
theta\_relax <- c()  
for (k in 1:5){  
 theta\_relax <- solve((t(x)%\*%solve(sigma)%\*%x))%\*%t(x)%\*%solve(sigma)%\*%Y  
 Y\_hat <- x%\*%theta\_relax  
 res <- DATA\_est$co2 - Y\_hat  
 res1 <- res[1:718-1]  
 res2 <- res[2:718]  
 rho <- cor(res1,res2)  
 J <- 1:718  
 P <- rho^(J-1)  
 sigma <- toeplitz(P)  
 Y <- Y\_hat  
}  
DATA\_est$wfit <- Y\_hat  
plot(co2~time, DATA\_est)  
lines(Y\_hat~time, DATA\_est, type="l", lwd=1, col = "blue")  
points(fit~time, DATA\_est, type="l", lwd=1, col = "red")  
legend("bottomright", legend = c("Observed","Fitted, OLS","Fitted, WLS"), col = c('black','blue','red'), pch=c(1,NA,1), lty=c(NA,1,1))



The difference between two plots are negligible and two methods almost rprovide identical solution.

The L value of the Linear + harmonic model in the case of the problem could be written as:

The f(0) as well could be written as: