Assignment#3

Peyman Kor

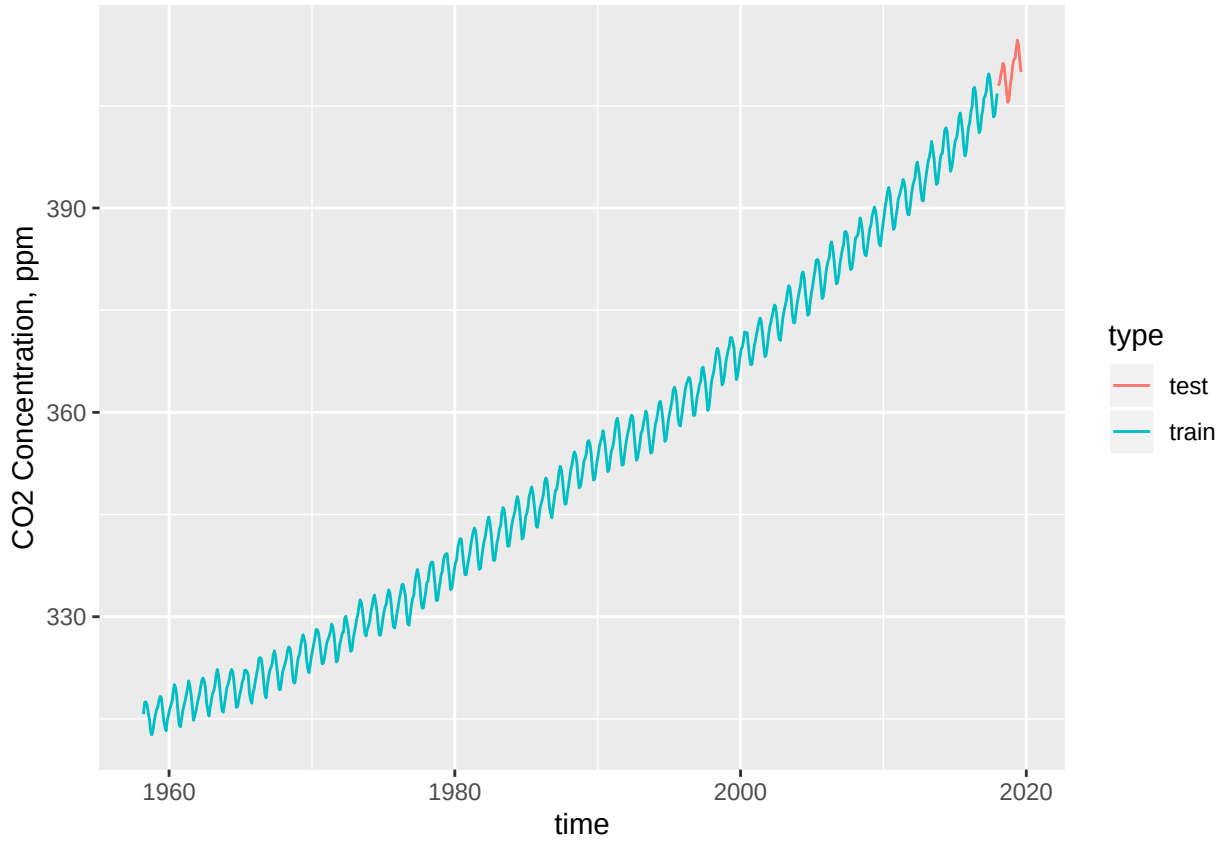
11/1/2019

Question 3.1 Plotting

The data set is divided to two sets. From the beginning until the beginning of the 2018 is considered as the **Train** Data set and from Jan 2018 beyond is considered the **Test** data-set.

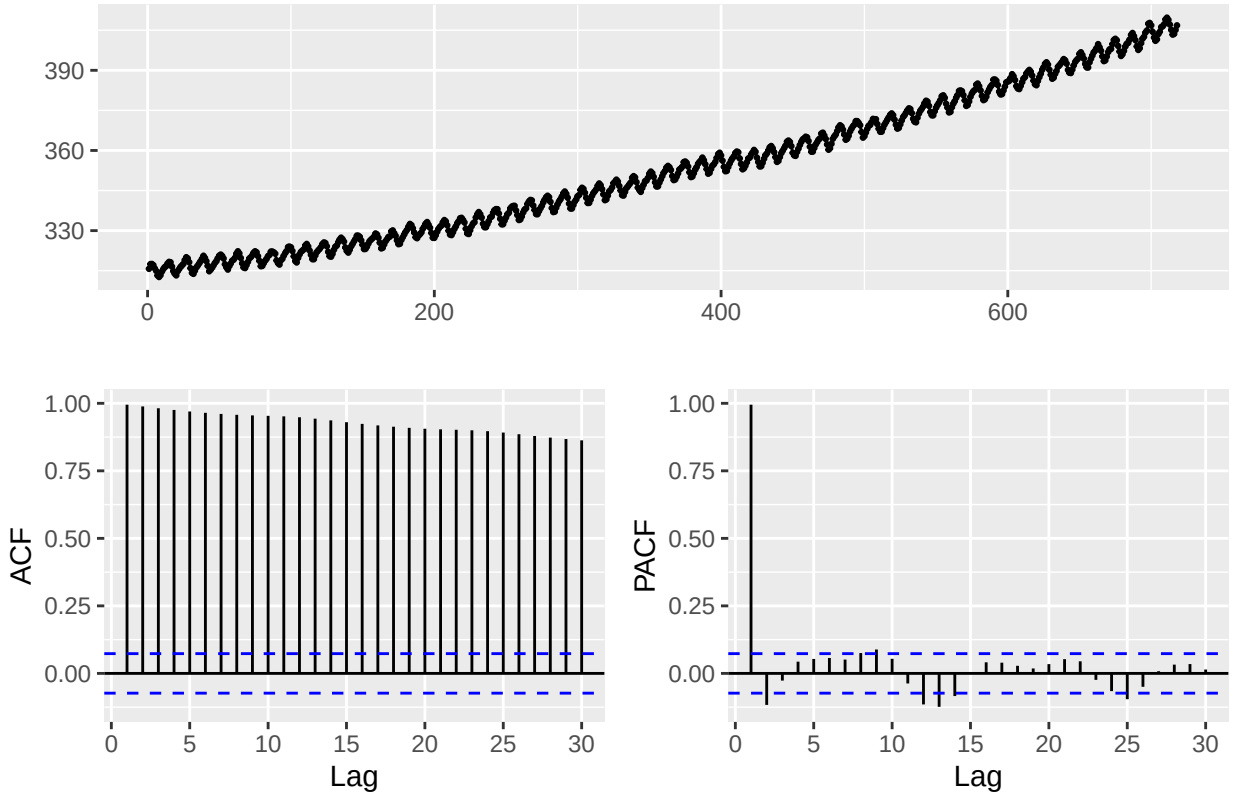
```
suppressMessages(library(tidyverse))
suppressMessages(library(forecast))
suppressMessages(library(car))

data <- read.csv('A3_co2.txt', sep = '')
data_train <- data[1:718,]
data_test <- data[719:738,]
data[1:718,'type'] <- 'train'
data[719:738, 'type'] <- 'test'
data['type'] <- as.factor(data$type)
ggplot(data, aes(time,co2, color=type)) +
  ylab("CO2 Concentration, ppm") +
  geom_line()
```



Question 3.2 Correlation Structure

```
data_train$co2%>% ggtsdisplay(lag.max = 30)
```

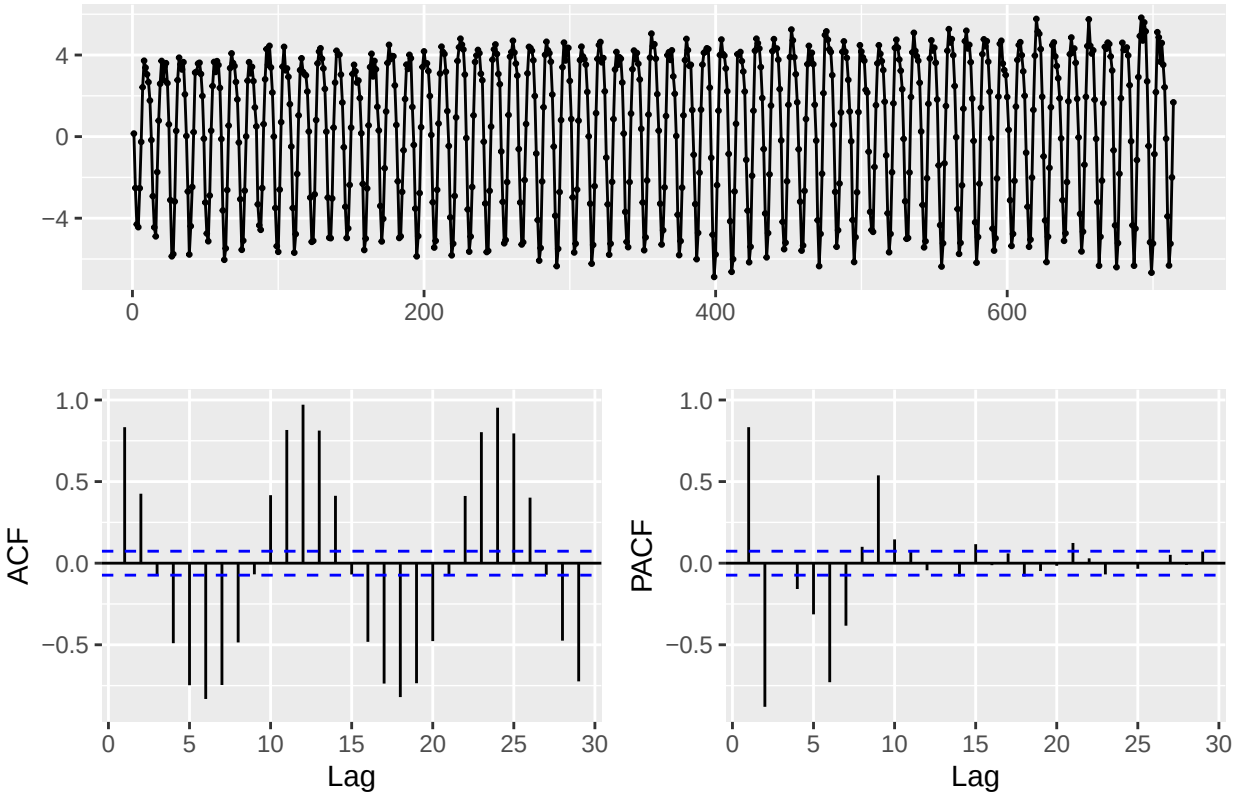


The ACF plot and PACF provides some hints regarding the structure we have here:

First, we can see that the data is obviously strongly seasonal as well non stationary. To deal with this seasonality, we will have look on the seasonal difference for ACF and PACF. However, several seasonal lag is considered to have further look on the data and also, we will then think whether the second difference is needed or we could go the first one, although the clear answer will not be obvious.

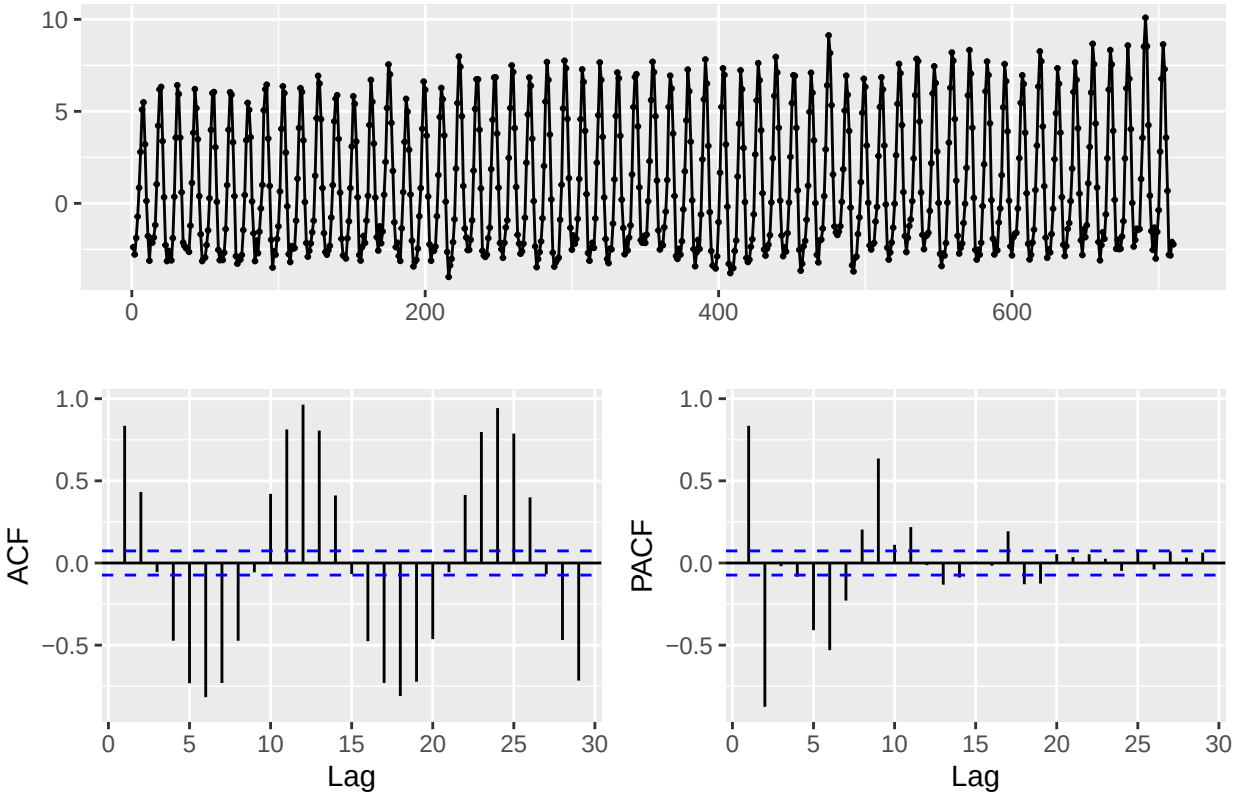
Differences with lag = 4

```
data_train$co2 %>% diff(lag=4) %>% ggtsdisplay()
```



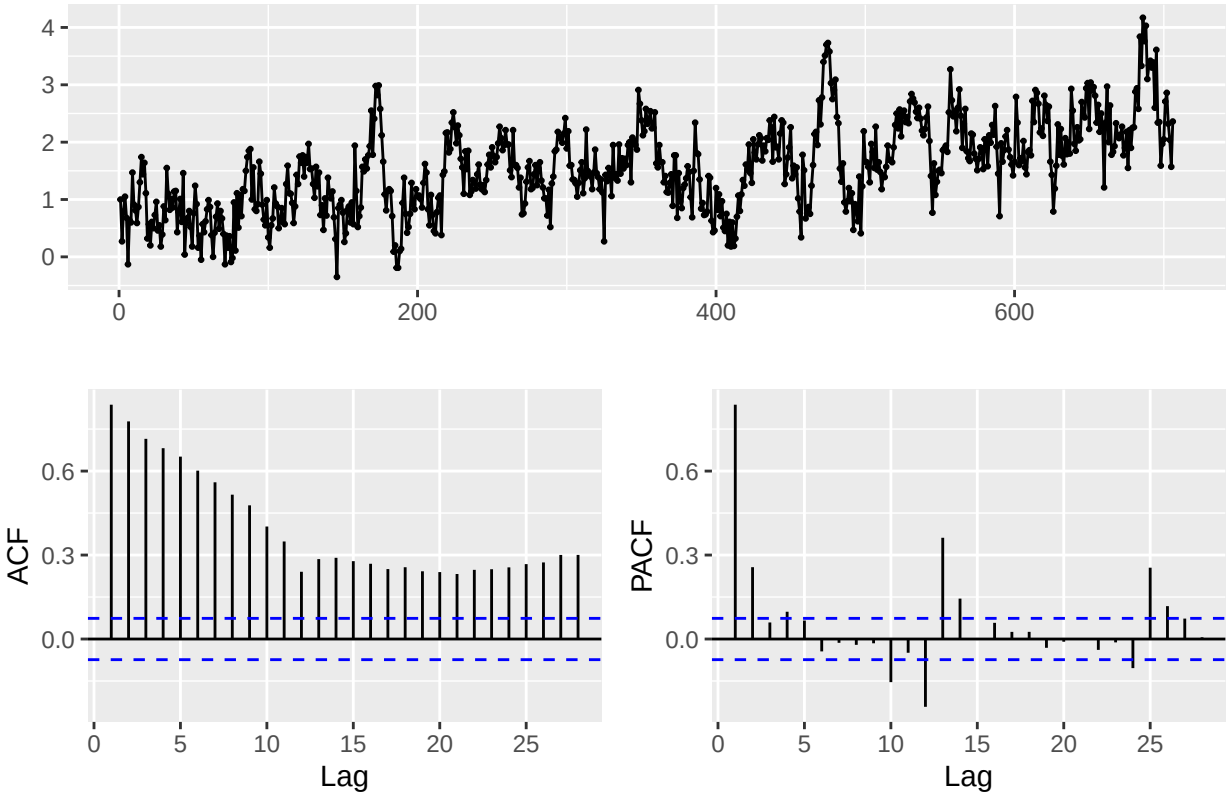
Differentiating with lag = 8

```
data_train$co2 %>% diff(lag=8) %>% ggtsdisplay()
```



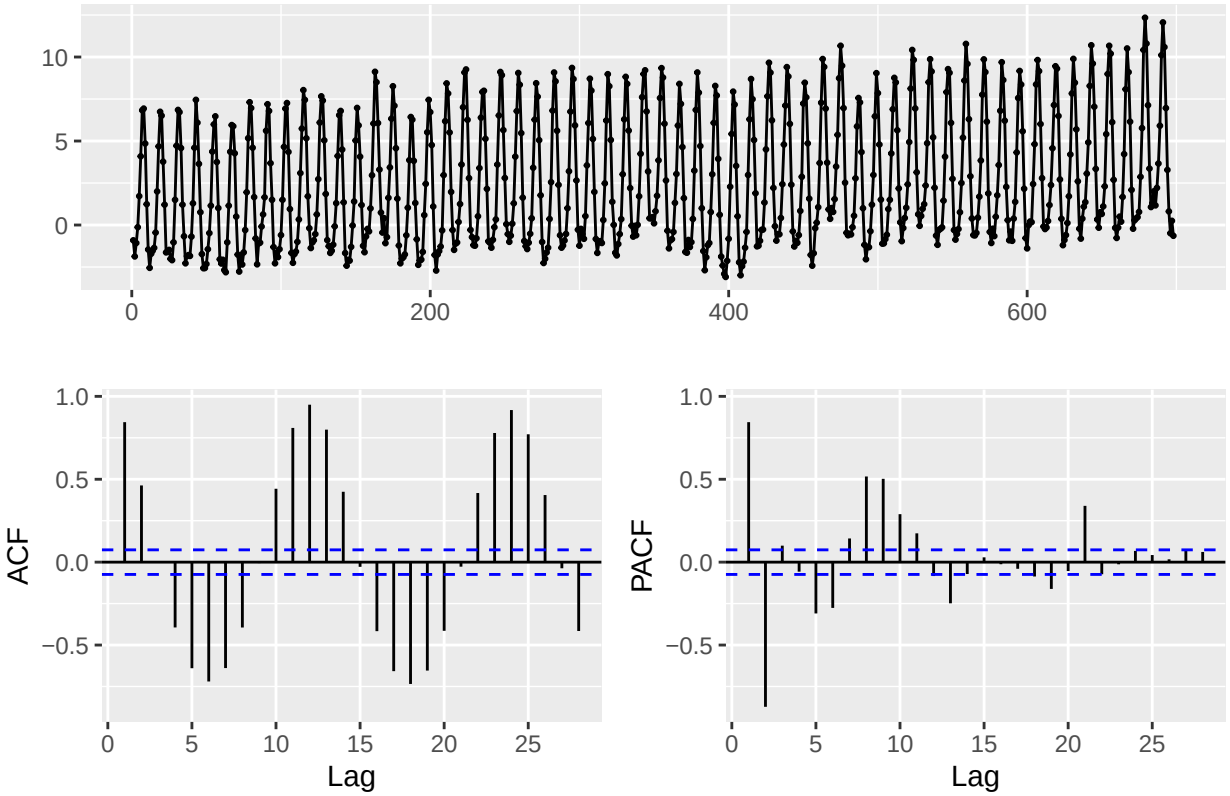
Differentiating with lag = 12

```
data_train$co2 %>% diff(lag=12) %>% ggtsdisplay()
```



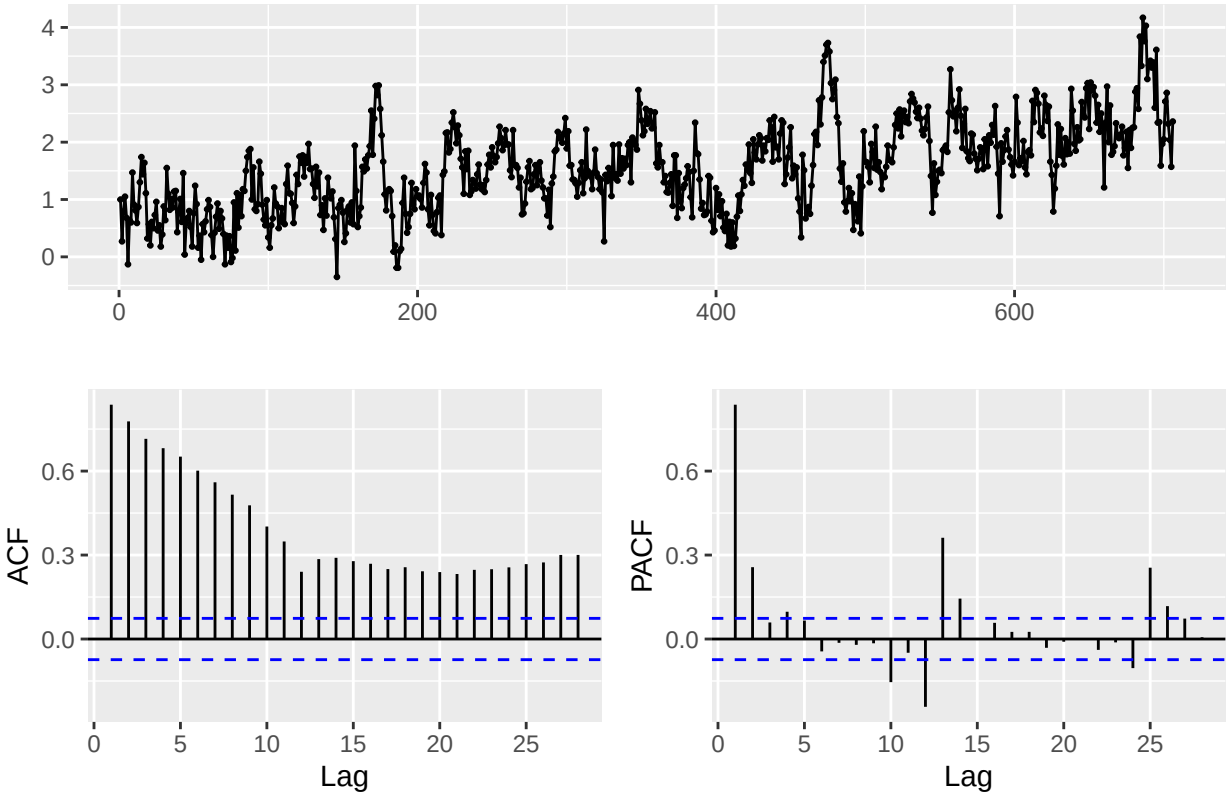
Differentiating with lag = 20

```
data_train$co2 %>% diff(lag=20) %>% ggtsdisplay()
```



As we can see that the difference helpful for make the dataset more stationary rather than have the trend in the data. Considering this initial analysis, we could argue that perhaps the lag = 12 could provide the more insight about the data than other lag size, however this lag can not fully make the data stationary, yet we stay with first order difference and will try to include the trend in ARIMA model to include a weak trend.

```
data_train$co2 %>% diff(1,lag=12) %>% ggtsdisplay()
```

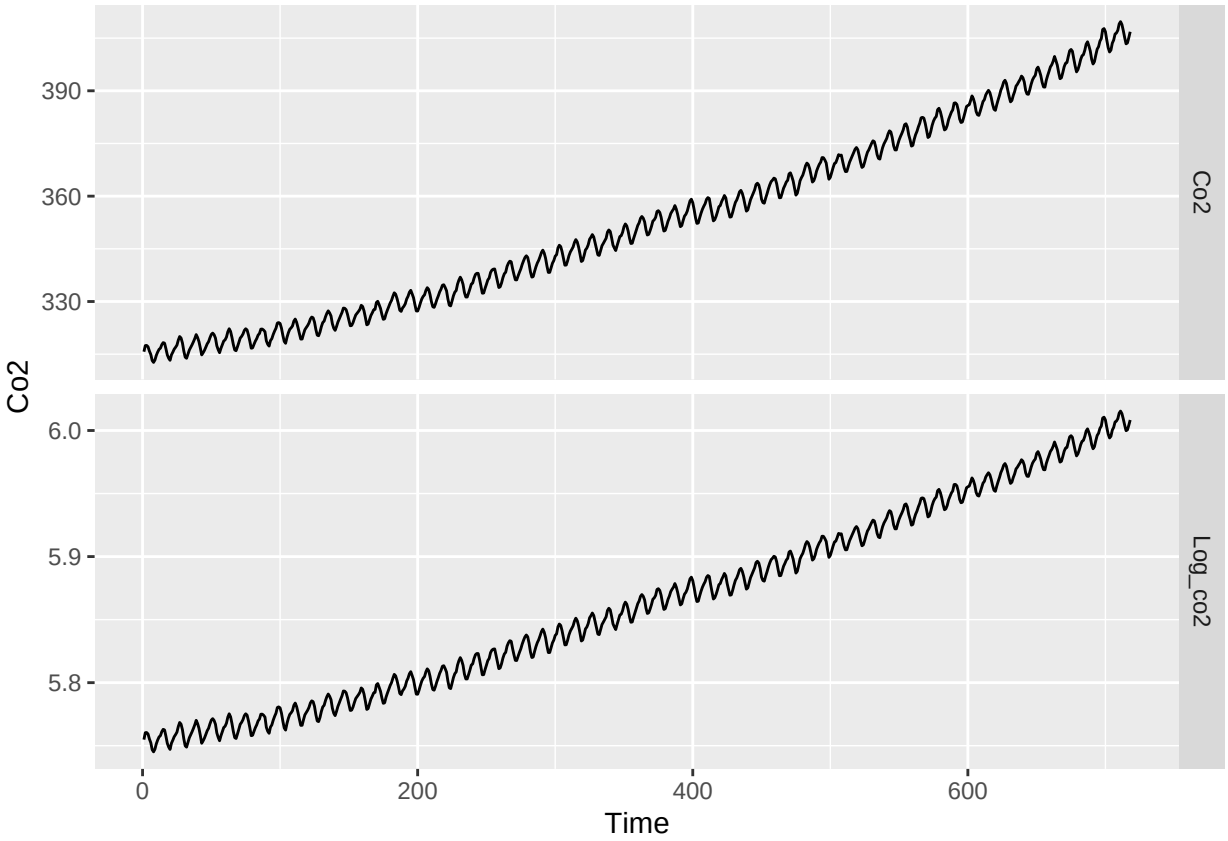


A few more points about the above plot is that in the non-seasonal lags, there are three significant spikes in the PACF, suggesting a possible AR(3) term. The pattern in the ACF is not indicative of any simple model.

On the other hand, we could potential utilize the logarithmic transformation of the co2 concentration in order to the levelize the variance of the data and which lead on the minimizing the curvature of the data, although it is not very obvious.

```
dataco2log <- cbind("Co2" = data_train$co2,
                    "Log_co2"=log(data_train$co2),
                    'Time'=seq(length(data_train$co2)))
dataco2log <- as.data.frame(dataco2log)
dataco2log_tidy=gather(dataco2log, "type", "Co2", 1:2)

p <- ggplot(dataco2log_tidy, aes(Time, Co2)) + geom_line()
p + facet_grid(rows = vars(type), scales = 'free_y')
```



Question 3.3: Procedure for identifying ARIMA model

The procedure to find the most appropriate model for the data is followed as :

1. First the data is plotted to have overview whether there is any seasonality and non stationary behavior (like trend) in the data set
2. Then if we see the changing variance, we may use the transformation of the data (like log) in order to stabilize the variance
3. If there is trend is data set, we take the difference until the stationary state appears in the data set
4. Now, having done some pre cleaning on the data, we evaluate the the ACF and PACF and try to determine the possible candidate model.
5. After building the model, we perform AICc analysis and take that one as the criteria for best model if the same model order was decided.

Note: Since we are using the forecast package in R, the information criteria is Akaike's Information Criterion (AIC) and we try to minimize the AICc (defined in the forecast pacakege) suitable for ARIMA package.

6. Now, the final criteria for selecting the model would be the ACF of PACF of the residual and also histogram of the residuals in order to make sure that the residual follow the white noise.
7. However, the above is valid for best model considering the train data set, ultimately all candidate models must be feed into the models to see the RMSE values in order to select the the model which goes to the forecast stage.

In summary, the goal is to find the minimum RMSE while make sure the residual follows the white noise distribution.

Question 3.4: ARIMA model

Now having look on ACF and PACF we could say that in non-seasonal lags, there are three significant spikes in the PACF, suggesting a possible AR(3) term. In addition, the seasonal part potentially could be considered with the D=12 and the lag of m=12. Having said that and considering the discussion in the class regarding using the drift to include the linear trend, the initial suggestion would be:

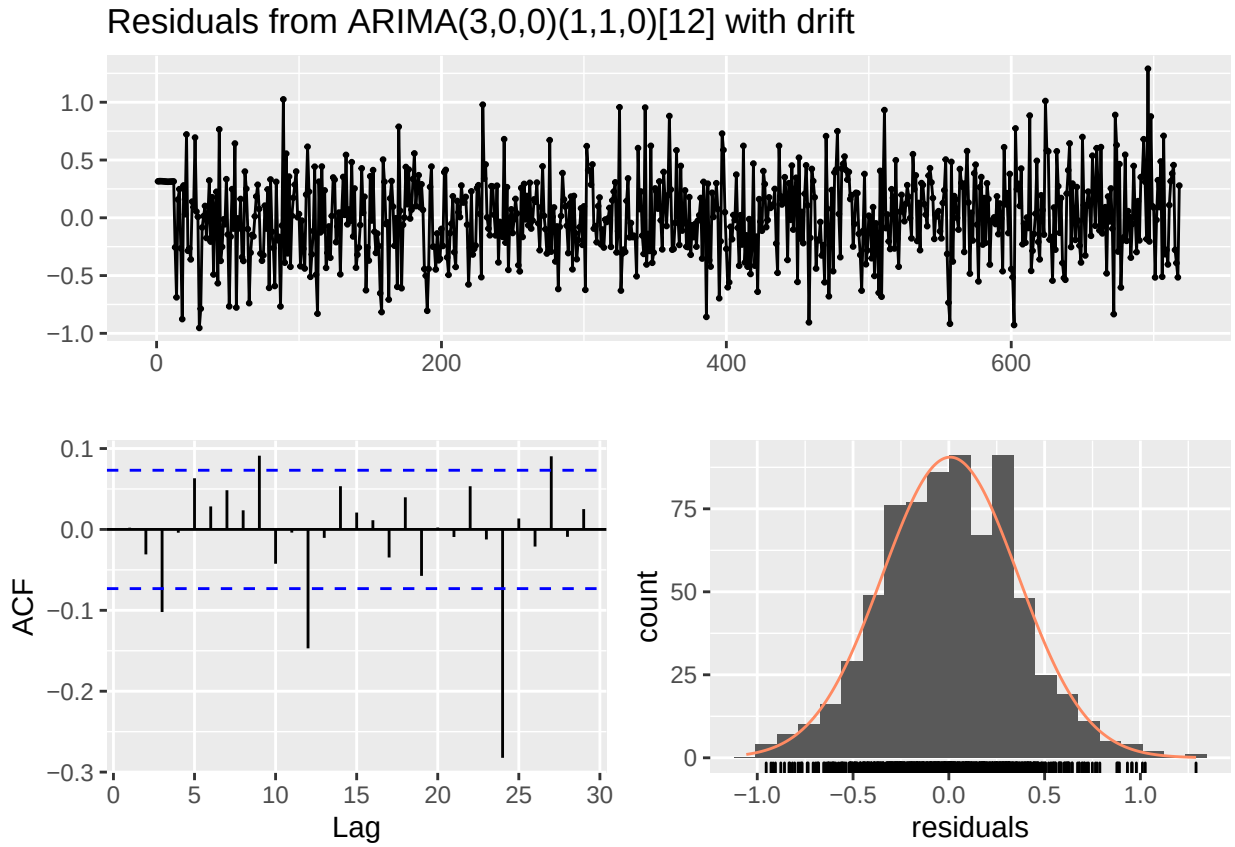
$$ARIMA(3,0,0)(1,1,0)_{12}$$

Having this model and figuring the AIC and the residual distribution:

```
library(forecast)
co2 <- data_train$co2
(fitnew_1 <- Arima(co2, order=c(3,0,0),seasonal=list(order=c(1,1,0),period=12),
                   include.drift = T
              ))
```

```
## Series: co2
## ARIMA(3,0,0)(1,1,0)[12] with drift
##
## Coefficients:
##          ar1     ar2     ar3     sar1   drift
##       0.6150  0.2378  0.0944  -0.4878  0.1300
## s.e.  0.0374  0.0433  0.0377   0.0339  0.0138
##
## sigma^2 estimated as 0.1288:  log likelihood=-273.57
## AIC=559.13   AICc=559.25   BIC=586.49
```

```
checkresiduals(fitnew_1, lag=36)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,0,0)(1,1,0)[12] with drift
## Q* = 123.05, df = 31, p-value = 6.469e-13
##
## Model df: 5.   Total lags used: 36
```

Now, considering this assumption, we add the a few variation to this structure while keeping the order of the Arima same, while we change the q,P and Q terms.

$$ARIMA(3,0,1)(0,1,1)_{12}$$

```
(fitnew_2 <- Arima(co2, order=c(3,0,1),seasonal=list(order=c(0,1,1),period=12),
                   include.drift = T
                ))
```

```
## Series: co2
## ARIMA(3,0,1)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sma1   drift
##       1.2888  -0.2226  -0.0667  -0.6582  -0.8644  0.1388
## s.e.  0.1106   0.0848   0.0587   0.1021   0.0196  0.0463
```

```
##
## sigma^2 estimated as 0.09767:  log likelihood=-180.48
## AIC=374.97   AICc=375.13   BIC=406.88
```

$$ARIMA(3,0,1)(1,1,1)_{12}$$

```
(fitnew_3 <- Arima(co2, order=c(3,0,1),seasonal=list(order=c(1,1,1),period=12),
                   include.drift = T
                ))
```

```
## Series: co2
## ARIMA(3,0,1)(1,1,1)[12] with drift
##
## Coefficients:
##           ar1     ar2     ar3     ma1    sar1     sma1   drift
##       -0.2995  0.9822  0.3124  0.9994  0.0133  -0.8694  0.1296
## s.e.   0.0363  0.0078  0.0363  0.0229  0.0436   0.0230  0.0353
##
## sigma^2 estimated as 0.09953:  log likelihood=-188.47
## AIC=392.95   AICc=393.16   BIC=429.43
```

$$ARIMA(3,0,2)(2,1,0)_{12}$$

```
(fitnew_4 <- Arima(co2, order=c(3,0,2),seasonal=list(order=c(2,1,0),period=12),
                   include.drift = T
                ))
```

```
## Series: co2
## ARIMA(3,0,2)(2,1,0)[12] with drift
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     sar1     sar2   drift
##       0.8146  0.2607  -0.0878  -0.2368  -0.1893  -0.6557  -0.3417  0.1317
## s.e.  0.5633  0.6104   0.0920   0.5630   0.2926   0.0365   0.0369  0.0218
##
## sigma^2 estimated as 0.1139:  log likelihood=-229.45
## AIC=476.9   AICc=477.16   BIC=517.94
```

$$ARIMA(3,0,2)(1,1,3)_{12}$$

```
(fitnew_5 <- Arima(co2, order=c(3,0,2),seasonal=list(order=c(1,1,3),period=12),
                   include.drift = T
                ))
```

```
## Series: co2
## ARIMA(3,0,2)(1,1,3)[12] with drift
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     sar1     sma1     sma2
```

```
##       0.9507  0.2031  -0.1545  -0.3208  -0.2058  -0.4196  -0.4427  -0.3835
## s.e.  0.4088  0.4754   0.1031   0.4087   0.2270   1.2521   1.2517   1.0662
##         sma3   drift
##       0.0201  0.1331
## s.e.  0.0416  0.0449
##
## sigma^2 estimated as 0.09808:  log likelihood=-179.98
## AIC=381.95   AICc=382.33   BIC=432.11
```

$$ARIMA(3,0,1)(1,1,4)_{12}$$

```
(fitnew_6 <- Arima(co2, order=c(3,0,1),seasonal=list(order=c(1,1,4),period=12),
                   include.drift = T
              ))
```

```
## Series: co2
## ARIMA(3,0,1)(1,1,4)[12] with drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sar1     sma1     sma2    sma3
##       1.2919  -0.2248  -0.0675  -0.6620  -0.3584  -0.5049  -0.3301  0.0222
## s.e.  0.1104   0.0846   0.0589   0.1016   1.3816   1.3807   1.1958  0.0533
##          sma4   drift
##       -0.0015  0.1355
## s.e.   0.0561  0.0475
##
## sigma^2 estimated as 0.09815:  log likelihood=-180.27
## AIC=382.54   AICc=382.92   BIC=432.7
```

Now considering the above evaluation, while the order is the same d=0, D=1, we can see that the fit_2 has the minimum AICc among the models. Now, let's change the order and see what happens if we do not include the linear trend, instead use the difference to the non-seasonal term as well to seasonal term. Now, considering same order, here the AICc is the measure of the quality of model:

$$ARIMA(3,1,0)(1,1,0)_{12}$$

```
(fitnew_11 <- Arima(co2, order=c(3,1,0),seasonal=list(order=c(1,1,0),period=12)))
```

```
## Series: co2
## ARIMA(3,1,0)(1,1,0)[12]
##
## Coefficients:
##           ar1      ar2      ar3     sar1
##       -0.3794  -0.1600  -0.1246  -0.4934
## s.e.   0.0374   0.0397   0.0375   0.0334
##
## sigma^2 estimated as 0.129:  log likelihood=-273.26
## AIC=556.53   AICc=556.61   BIC=579.32
```

$$ARIMA(3,1,2)(0,1,1)_{12}$$

```r
(fitnew_12 <- Arima(co2, order=c(3,1,2),seasonal=list(order=c(0,1,1),period=12)
             ))
```

```
## Series: co2
## ARIMA(3,1,2)(0,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2     sma1
##       0.5459  -0.1189  -0.1067  -0.9142  0.2891  -0.8629
## s.e.  0.5674   0.1806   0.0643   0.5698  0.3680   0.0197
##
## sigma^2 estimated as 0.09755:  log likelihood=-178.53
## AIC=371.07   AICc=371.23   BIC=402.98
```

$$ARIMA(3,1,0)(1,1,2)_{12}$$

```r
(fitnew_13 <- Arima(co2, order=c(3,1,0),seasonal=list(order=c(1,1,2),period=12)
             ))
```

```
## Series: co2
## ARIMA(3,1,0)(1,1,2)[12]
##
## Coefficients:
##          ar1      ar2      ar3     sar1    sma1     sma2
##       -0.3552  -0.1453  -0.1140  -0.9661  0.1211  -0.8557
## s.e.   0.0383   0.0399   0.0378   0.0451  0.0530   0.0449
##
## sigma^2 estimated as 0.0978:  log likelihood=-180.62
## AIC=375.24   AICc=375.4   BIC=407.15
```

$$ARIMA(3,1,2)(0,1,3)_{12}$$

```r
(fitnew_14 <- Arima(co2, order=c(3,1,2),seasonal=list(order=c(0,1,3),period=12)
             ))
```

```
## Series: co2
## ARIMA(3,1,2)(0,1,3)[12]
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2     sma1     sma2    sma3
##       0.5088  -0.1062  -0.1038  -0.8773  0.2623  -0.8602  -0.0231  0.0216
## s.e.  0.5536   0.1841   0.0662   0.5557  0.3672   0.0388   0.0524  0.0400
##
## sigma^2 estimated as 0.09778:  log likelihood=-178.39
## AIC=374.77   AICc=375.03   BIC=415.8
```

$$ARIMA(3,1,1)(1,1,3)_{12}$$

```
(fitnew_15 <- Arima(co2, order=c(3,1,1),seasonal=list(order=c(1,1,3),period=12)
             ))
```

```
## Series: co2
## ARIMA(3,1,1)(1,1,3)[12]
##
## Coefficients:
##          ar1     ar2      ar3      ma1     sar1     sma1     sma2    sma3
##       0.1690  0.0290  -0.0623  -0.5369  -0.3271  -0.5340  -0.3045  0.0206
## s.e.  0.1633  0.0702   0.0492   0.1603   1.3053   1.3056   1.1174  0.0397
##
## sigma^2 estimated as 0.09784:  log likelihood=-178.63
## AIC=375.26   AICc=375.52   BIC=416.28
```

$$ARIMA(3,1,1)(1,1,2)_{12}$$

```
(fitnew_16 <- Arima(co2, order=c(3,1,1),seasonal=list(order=c(1,1,2),period=12)
             ))
```

```
## Series: co2
## ARIMA(3,1,1)(1,1,2)[12]
##
## Coefficients:
##          ar1     ar2      ar3      ma1     sar1    sma1     sma2
##       0.1111  0.0100  -0.0782  -0.4789  -0.9533  0.1102  -0.8447
## s.e.  0.1744  0.0725   0.0478   0.1728   0.0459  0.0484   0.0413
##
## sigma^2 estimated as 0.09732:  log likelihood=-177.78
## AIC=371.57   AICc=371.77   BIC=408.03
```

RMSE and Test set Evaluation:

Now, it should be our understanding that when compare the model forecast with test data set using the metric (RMSE) no matter which order we used, the evaluation is always valid. Therefore, we compare the five models comparing their RMSE to select the final model to go for forecasting.

```
accuracy(forecast(fitnew_1,h=20)$mean, data_test$co2)
```

```
##                 ME      RMSE       MAE        MPE     MAPE
## Test set 0.1947858 0.6570739 0.5311732 0.04724705 0.129263
```

```
accuracy(forecast(fitnew_5,h=20)$mean, data_test$co2)
```

```
##                  ME      RMSE       MAE         MPE       MAPE
## Test set 0.08950194 0.4768108 0.3848975 0.02155764 0.09369852
```

```
accuracy(forecast(fitnew_12,h=20)$mean, data_test$co2)
```

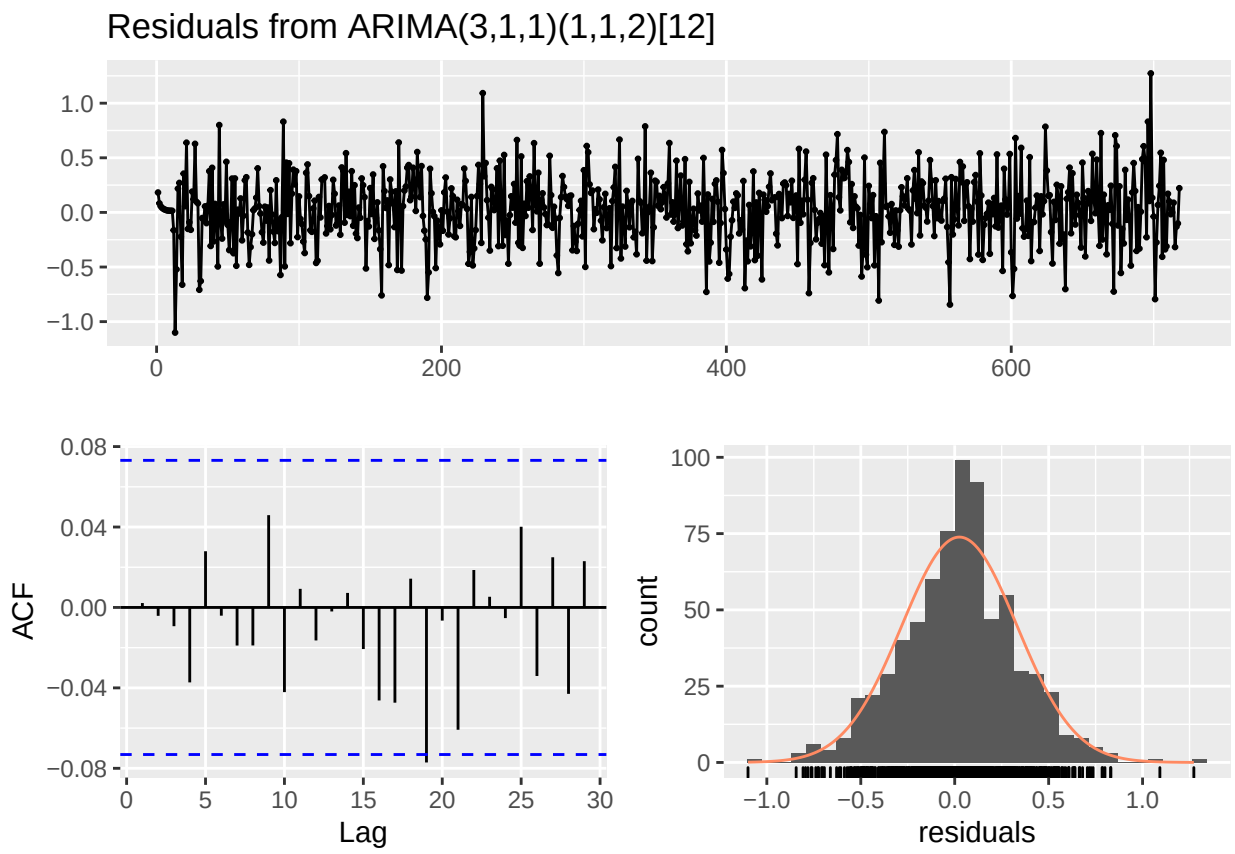
```
##                     ME      RMSE       MAE         MPE       MAPE
## Test set 0.04514225 0.4597656 0.3717613 0.01074921 0.09051388
```

```
accuracy(forecast(fitnew_16,h=20)$mean, data_test$co2)
```

```
##                     ME      RMSE       MAE         MPE       MAPE
## Test set 0.06330761 0.4500885 0.3652271 0.01519789 0.08891738
```

Now, see that the model number fitnew_6 has the lowest RMSE among the models, yet at the end we will have look on the residuals and ACF plot of residuals, and if confirm they represent the white noise, the model be considered for prediction.

```
checkresiduals(fitnew_16, lag=30)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,1)(1,1,2)[12]
## Q* = 21.617, df = 23, p-value = 0.5435
##
## Model df: 7.   Total lags used: 30
```

So, let's plot the qqplot to make sure that the residual are white noise- As we can see, the models number 1, 12, 16 diverge from the low and high quantiles of the data, but it seems the model number one is more consistent with the extreme quntile of data set.

```
par(mfrow=c(2,2))
qqPlot(fitnew_1$residuals)
```

```
## [1] 696  89
```

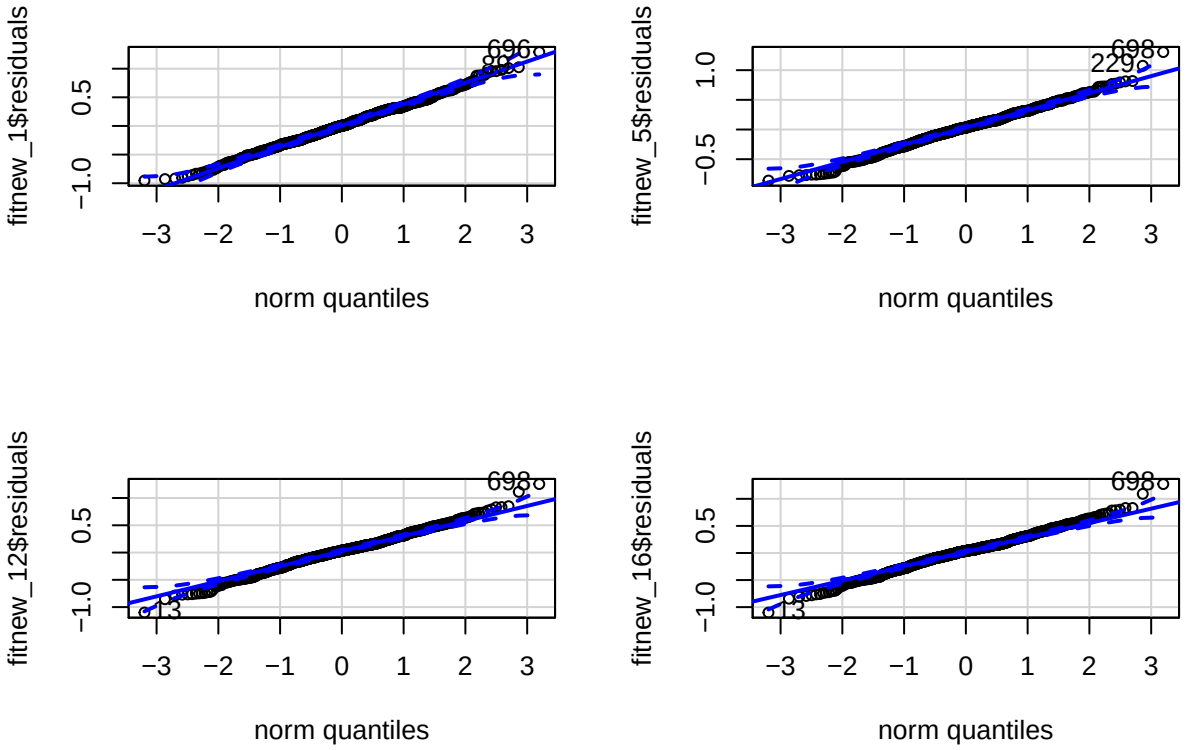
```
qqPlot(fitnew_5$residuals)
```

```
## [1] 698 229
```

```
qqPlot(fitnew_12$residuals)
```

```
## [1] 698  13
```

```
qqPlot(fitnew_16$residuals)
```




```
## [1] 698  13
```

However, since we defined that the RMSE is the final criteria after comparing with the test data, we will continue to work with the model number 16.

Question 3.5: Predictions

```
prediction <- forecast(fitnew_16,h=48)
prediction_point <- prediction$mean
table_pre_test=data_frame(mounth= seq(20),
                          Test_data = data_test$co2,
           mean_prediction=prediction_point[1:20],
           Low_95 = prediction$lower[1:20,2],
           High_95 = prediction$upper[1:20,2])
```

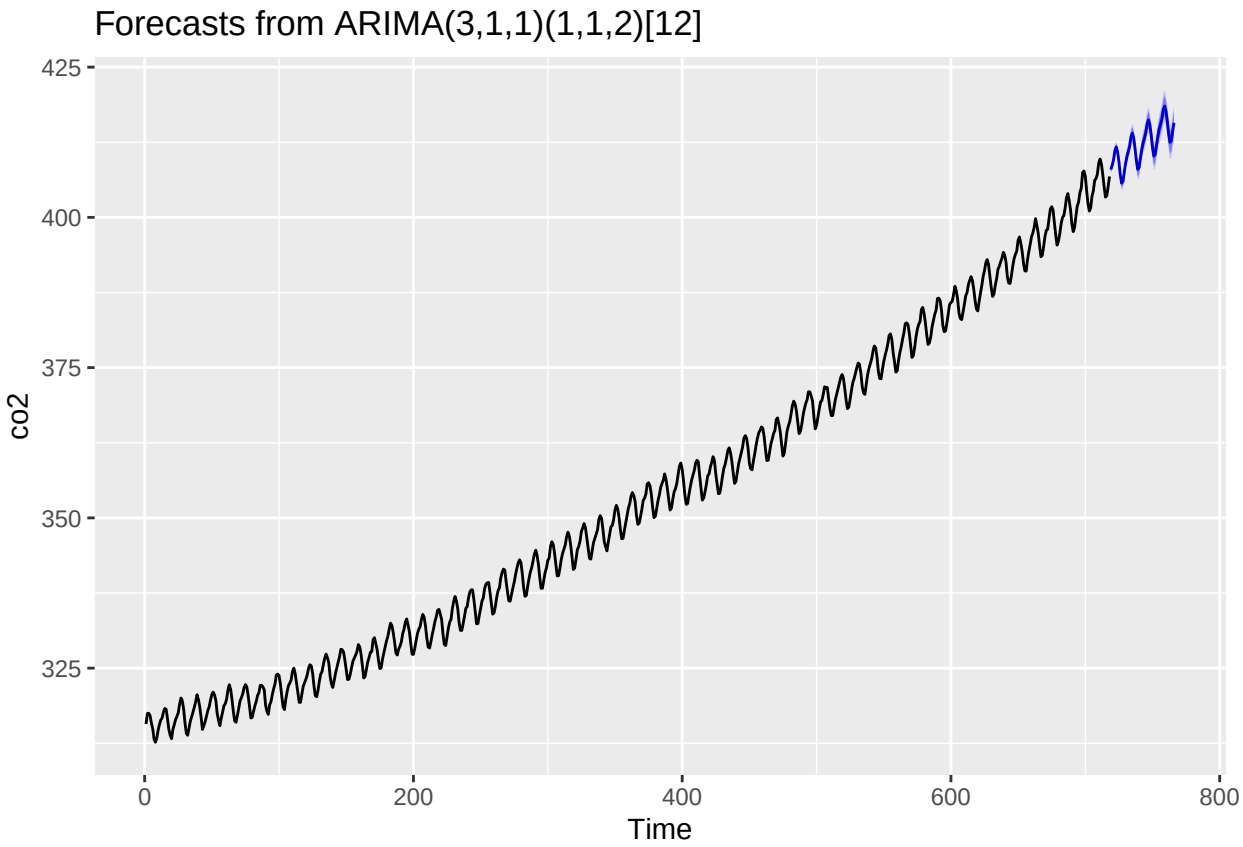
```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
table_pre_test
```

```
## # A tibble: 20 x 5
##    mounth Test_data mean_prediction Low_95 High_95
##     <int>     <dbl>           <dbl>  <dbl>   <dbl>
##  1      1      408.            408.   407.    409.
##  2      2      408.            409.   408.    409.
##  3      3      409.            410.   409.    410.
##  4      4      410.            411.   410.    412.
##  5      5      411.            412.   411.    413.
##  6      6      411.            411.   410.    412.
##  7      7      409.            409.   408.    410.
##  8      8      407.            407.   406.    408.
##  9      9      406.            406.   405.    407.
## 10     10      406             406.   405.    407.
## 11     11      408.            407.   406.    409.
## 12     12      409.            409.   408.    410.
## 13     13      411.            410.   409.    412.
## 14     14      412.            411.   410.    412.
## 15     15      412.            412.   410.    413.
## 16     16      413.            413.   412.    415.
## 17     17      415.            414.   412.    416.
## 18     18      414.            413.   412.    415.
## 19     19      412.            412.   410.    413.
## 20     20      410.            409.   408.    411.
```

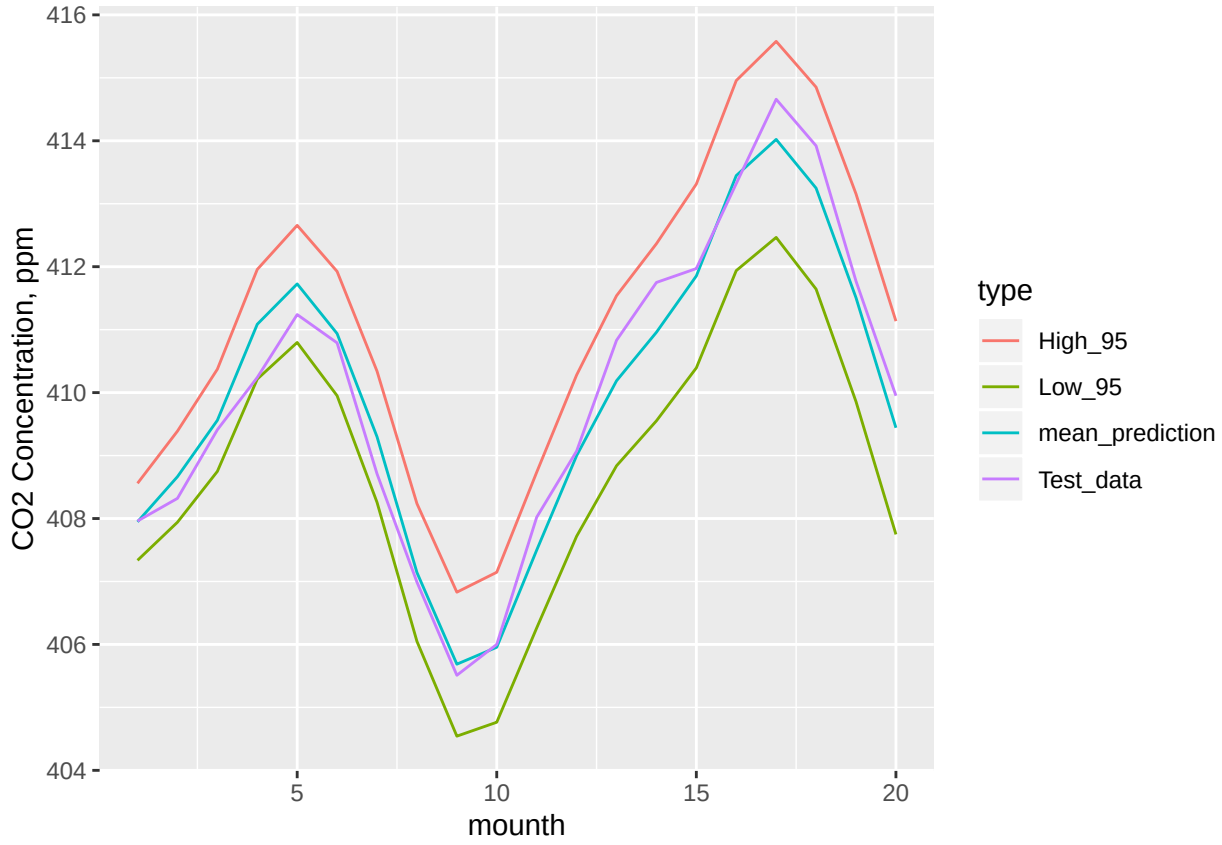
We can plot the data for the prediction for next 4 years as :

```
fitnew_16 %>% forecast(h=48) %>% autoplot()
```



In order to have some visualization, let's plot the mean prediction, 95% confidence interval next 24 mount in one single plot:

```
tidy_pred_test <- gather(table_pre_test, "type", "Co2", 2:5)
ggplot(tidy_pred_test, aes(mounth,Co2, color=type)) +
  ylab("CO2 Concentration, ppm") +
  geom_line()
```



Question 3.6: Attaining 460 ppm

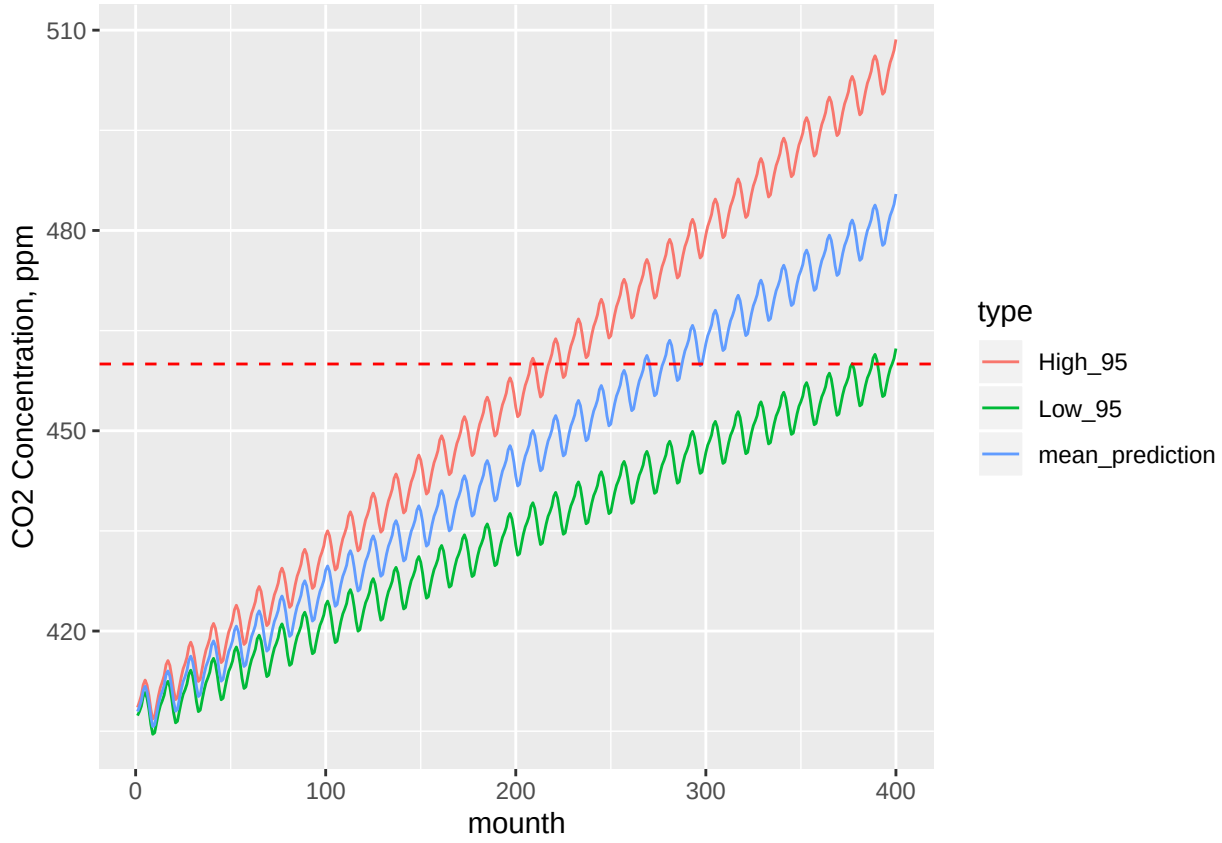
```
prediction_460 <- forecast(fitnew_16,h=400)
table_pre_460=data_frame(mounth= seq(400),
           mean_prediction=prediction_460$mean,
           Low_95 = prediction_460$lower[,2],
           High_95 = prediction_460$upper[,2])

table_pre_460
```

```
## # A tibble: 400 x 4
##    mounth mean_prediction Low_95 High_95
##     <int>           <dbl>  <dbl>   <dbl>
##  1      1            408.   407.    409.
##  2      2            409.   408.    409.
##  3      3            410.   409.    410.
##  4      4            411.   410.    412.
##  5      5            412.   411.    413.
##  6      6            411.   410.    412.
##  7      7            409.   408.    410.
##  8      8            407.   406.    408.
##  9      9            406.   405.    407.
## 10     10            406.   405.    407.
## # ... with 390 more rows
```

```
tidy_pred_460 <- gather(table_pre_460, "type", "Co2", 2:4)
ggplot(tidy_pred_460, aes(mounth,Co2, color=type)) +
  ylab("CO2 Concentration, ppm") +
  geom_line() +
  geom_hline(yintercept=460, linetype="dashed",
                color = "red")
```

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




```
which(grepl(460, table_pre_460$mean_prediction))
```

```
## [1]  11  39 132 268 270 278 288 298
```

According to mean prediction, the mouth 268 is the first month that the Co2 concentration meet reach the 460 ppm. This 268 is the number of months after the first month of the 2018 therefore we could say:

Mean Point Prediction

number of month = 268 / 12 = 22.3

The year first mean prediction reach the 460 ppm = 2018 + 22.3 = 2040.3

95 % Interval:

Lower

number of month = 208 / 12 = 17.3

The year first mean prediction reach the 460 ppm = 2018 + 17.3 = 2035.3

Higher

number of month = 388 / 12 = 32.3

The year first mean prediction reach the 460 ppm = 2018 + 32.3 = 2050.3

References

- Madsen, Henrik. Time series analysis. Chapman and Hall/CRC, 2007.
- Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2