STAT 443: Assignment 3

Wenxuan Zan (61336194)

21 March, 2022

Question 1

```
set.seed(61336194)
mu = 2
alpha = c(-0.8,0,0.8)
sigma2_alpha0 = 0.25/(1-0.8^2)
sigma2 = 0.25
m = 5000
n = 500

simulation_data <- as.data.frame(matrix(data = c(rep(0,3*m)),
                                        nrow = m,
                                        ncol = 3))
for (i in 1:m) {
  x1 = arima.sim(n = 500,
                 list(ar = c(alpha[1])),
                 sd = sqrt(sigma2)) + mu
  simulation_data[i,1] = mean(x1)

  x2 = ts(rnorm(500,0,sqrt(sigma2_alpha0))) + mu
  simulation_data[i,2] = mean(x2)

  x3 = arima.sim(n = 500,
                 list(ar = c(alpha[3])),
                 sd = sqrt(sigma2)) + mu
  simulation_data[i,3] = mean(x3)
}
```

a)

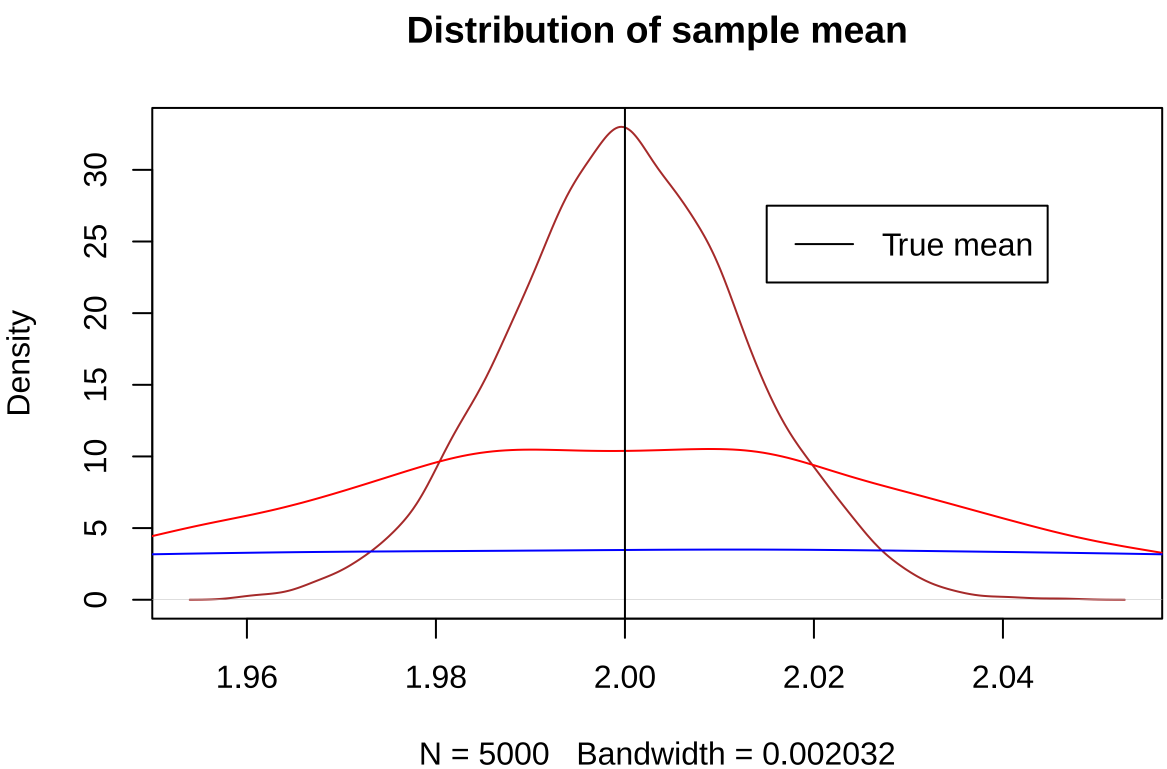
```
summary <- data.frame(alpha = c("-0.8","0","0.8"),
                      emprical_mean = c(0,0,0),
                      sd = c(0,0,0))
for (i in 1:3) {
  summary[i,2] = round(mean(simulation_data[,i]),3)
  summary[i,3] = round(sd(simulation_data[,i]),3)
}
kable(summary,
      caption = "Summary Result for AR(1) With Various Alpha")
```

Table 1: Summary Result for AR(1) With Various Alpha

alpha	emprical_mean	sd
-0.8	2.000	0.013
0	2.000	0.037
0.8	2.001	0.110

b)

```
plot(density(simulation_data$V1),
     col = "brown",
     type = "l",
     main = "Distribution of sample mean")
lines(density(simulation_data$V2), col = "red", type = "l")
lines(density(simulation_data$V3), col = "blue", type = "l")
abline(v = 2.00)
legend(2.015,
       27.5,
       legend = c("True mean"),
       lty = c("solid"),
       col = c("black"))
```

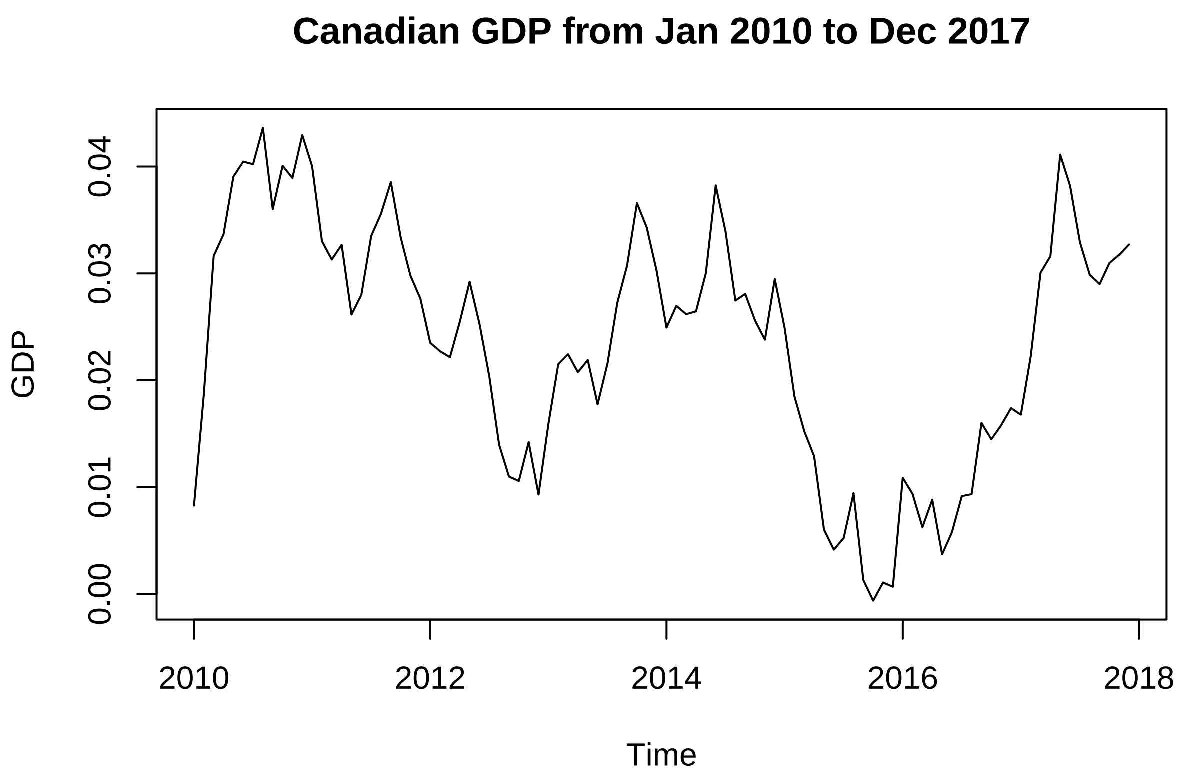


Looking at the above plot, $\bar{X}$ is unbiased under all three temporal dependence scenarios, but the variance is smallest when $\alpha = -0.8$. So It appears that when the data are negatively correlated, the variance is minimized.

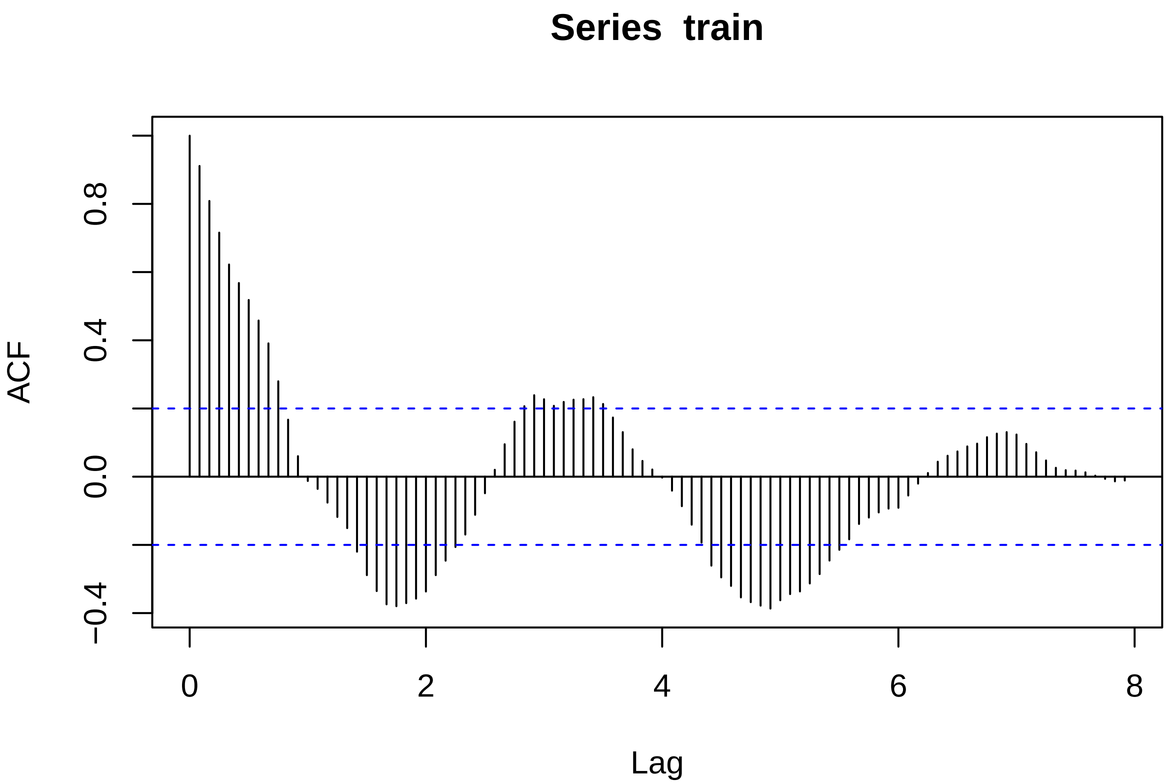
Question 2

a)

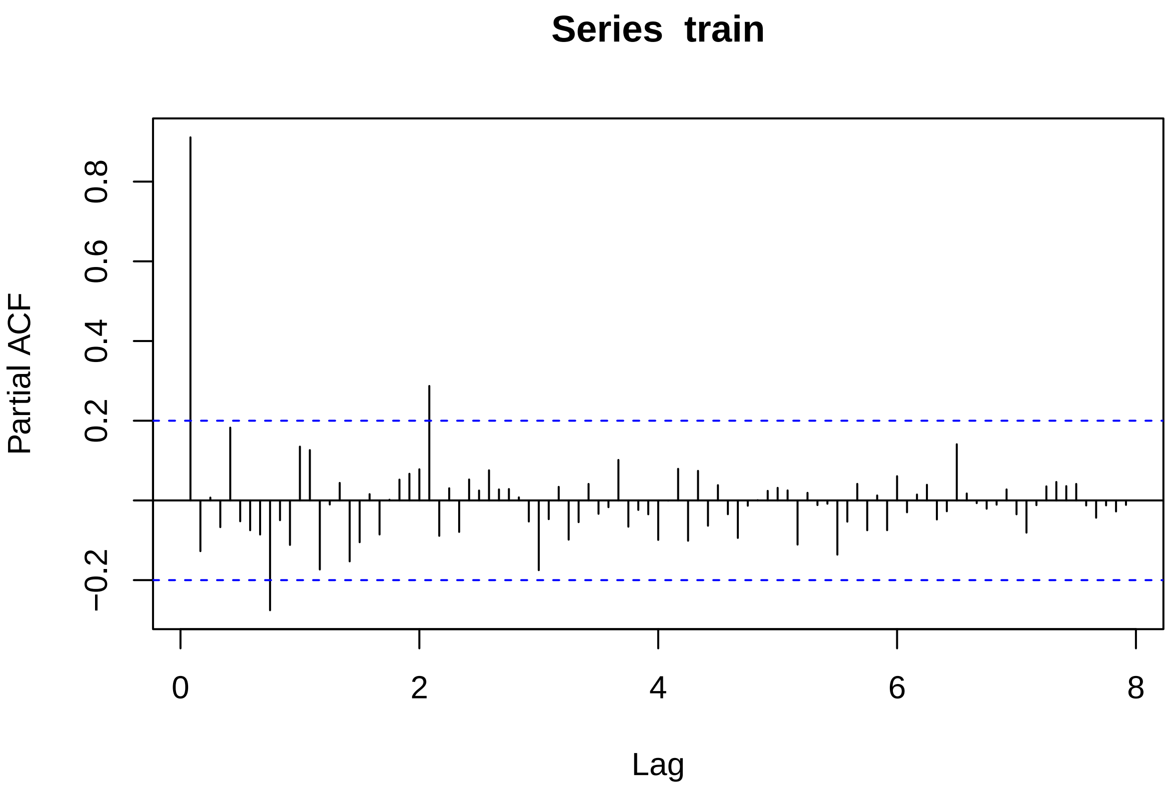
```
q2data <- read.csv("CanandaGDP.csv",header = TRUE)
n = length(q2data$GDP)
xt <- ts(log((q2data$GDP[-c(1:12)])/(q2data$GDP[-c((n-11):n)])),
                                start = c(1998,1),
                                frequency = 12)
train <- window(xt,
                start = c(2010,1),
                end = c(2017,12))
test <- window(xt,
               start = c(2018,1),
               end = c(2019,12))
plot(train,
     main = "Canadian GDP from Jan 2010 to Dec 2017",
     ylab = "GDP")
```




```
acf(train, lag.max = 100)
```




```
pacf(train, lag.max = 100)
```



The year-over-year log growth rate does not appear to be stationary. The ACF shows an damped sin wave decay pattern, and pacf shows a cut-off around lag 2. This suggests an AR process.

b)

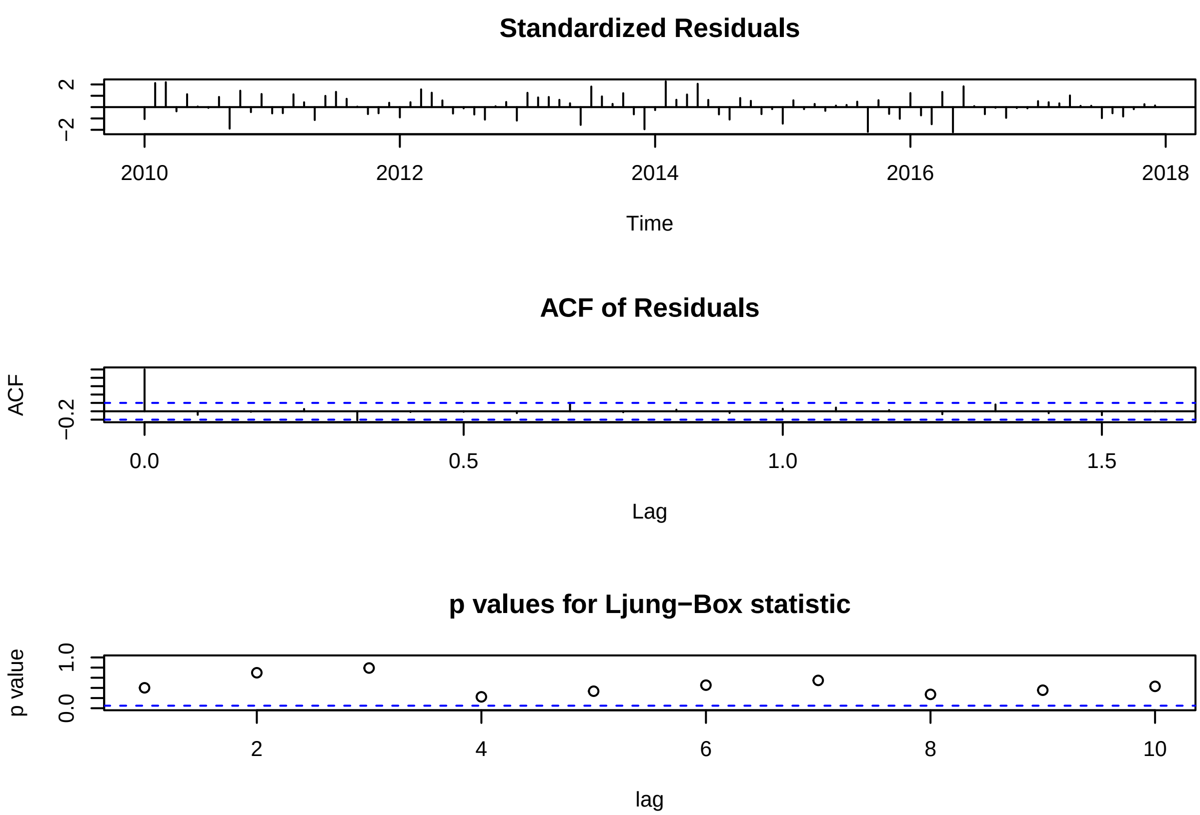
```r
result <- data.frame(p = c(rep(0:3,3)),
                     Q = c(rep(0,3), rep(1,3), rep(2,3), rep(3,3)),
                     AIC = c(rep(0,12)))
for (i in 1:12) {
  mod <- arima(train,
               order = c(result$p[i],0,0),
               seasonal = list(order = c(0,0,result$Q[i])))
  result[i,3] <- AIC(mod)
}
print(result[which.min(result$AIC),])
```

```
##    p Q       AIC
## 11 2 3 -813.0156
```

Since we know is this probably a process with ar(2) component, we search fit model with $p \in \{0, 1, 2, 3\}$ and also $Q \in \{0, 1, 2, 3\}$, and the result indicates that SARIMA(2,0,0)x(0,0,3) result in the smallest AIC.

c)

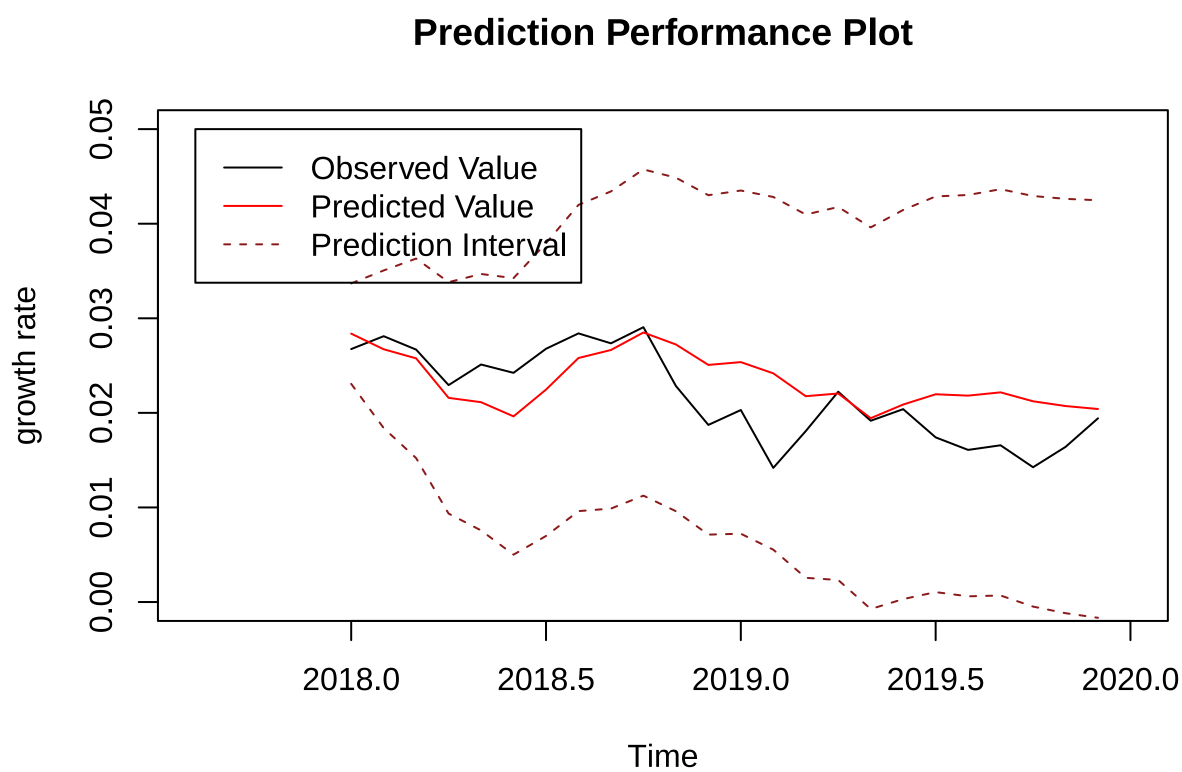
```r
tsdiag(arima(train,
             order = c(2,0,0),
             seasonal = list(order = c(0,0,3))))
```



Looking at the model fit above, all standardized residuals are within the ±2 bound and all acf values except for lag 0 are non-significant. Ljung-Box statistics have non-significant p-value until lag 10. The above indicates the model fit is good.

d)

```
model <- arima(train,
               order = c(2,0,0),
               seasonal = list(order = c(0,0,3)))
pred <- predict(model, n.ahead = 24,  prediction.interval = TRUE, level = 0.95)
plot(test, type = "l", pch = 19,
     main = "Prediction Performance Plot" ,ylab = "growth rate",
     xlim = c(2017.6, 2020.0),
     ylim = c(0,0.05))
lines(pred$pred, col = "red", type = "l", pch = 19)
lines(pred$pred - 1.96*pred$se, col = "firebrick4", lty = "dashed")
lines(pred$pred + 1.96*pred$se, col = "firebrick4", lty = "dashed")
legend(2017.6,
       0.05,
       legend = c("Observed Value",
                  "Predicted Value",
                  "Prediction Interval"),
       lty = c("solid","solid","dashed"),
       col = c("black", "red","firebrick4"))
```

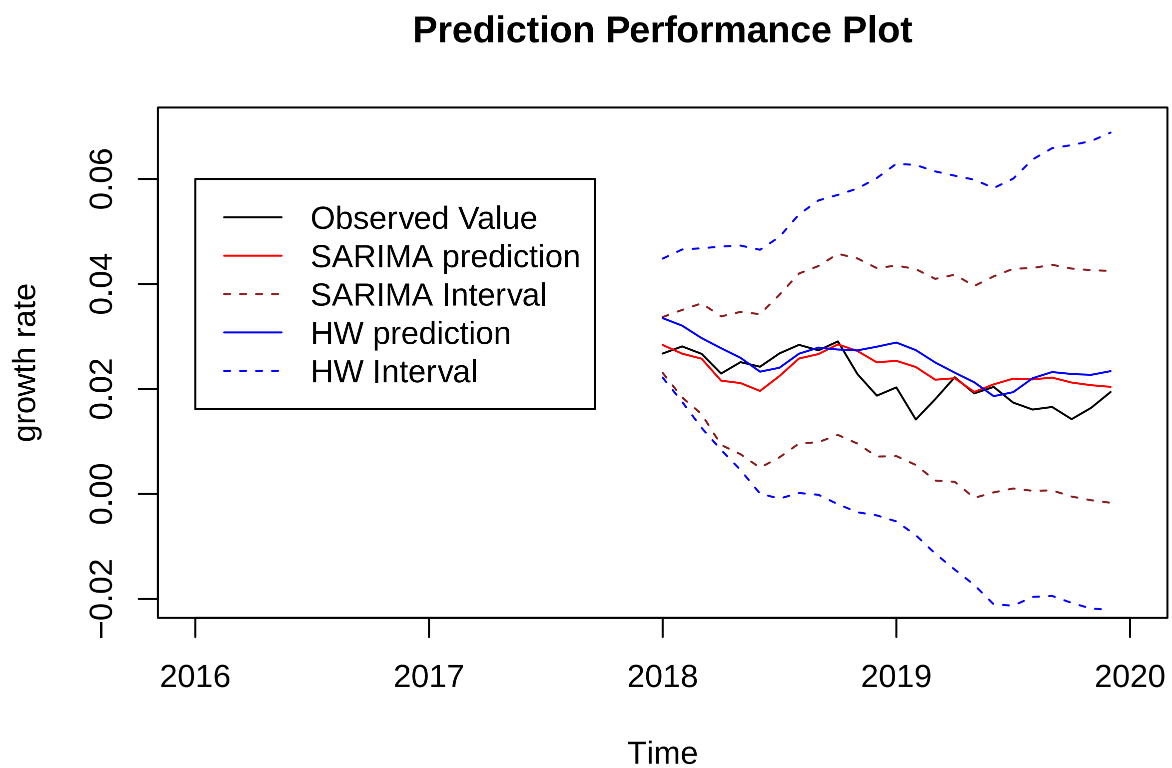


The prediction follows the observed values quite closely as we can see from above.

e)

```
HWmodel <- HoltWinters(train,seasonal = "additive")
predHW <- predict(HWmodel, n.ahead = 24,prediction.interval = TRUE, level = 0.95)
plot(test, type = "l", pch = 19,
     main = "Prediction Performance Plot" ,ylab = "growth rate",
```

```r
     xlim = c(2016, 2020.0),
     ylim = c(-0.02,0.07))
lines(pred$pred, col = "red", type = "l", pch = 19)
lines(pred$pred - 1.96*pred$se, col = "firebrick4", lty = "dashed")
lines(pred$pred + 1.96*pred$se, col = "firebrick4", lty = "dashed")
lines(predHW[,"fit"], col = "blue", type = "l", pch = 19)
lines(predHW[,"upr"], col = "blue", lty = "dashed")
lines(predHW[,"lwr"], col = "blue", lty = "dashed")
legend(2016,
       0.06,
       legend = c("Observed Value",
                  "SARIMA prediction",
                  "SARIMA Interval",
                  "HW prediction",
                  "HW Interval"),
       lty = c("solid","solid","dashed","solid","dashed"),
       col = c("black", "red","firebrick4","blue","blue"))
```



The HoltWinter prediction and Box-Jenkins' prediction have similar values, but HoltWinter yields a wider prediction interval.

f)

```r
mspeBox <- mean((test - pred$pred)^2)
mspeHW <- mean((test - predHW[,"fit"])^2)
print(mspeBox)
```

```
## [1] 1.7621e-05
```

```r
print(mspeHW)
```

```
## [1] 3.054621e-05
```

I would recommend the Box-Jenkins' method for 2 reasons:

i) Box-Jenkins' method yield a narrow prediction interval as shown in part e)
ii) Box-Jenkins' method has a smaller MSPE as calculated above.

The HoltWinter's method is easier to implement, but the Box-Jenkins' method allows for more freedom to tailor your model to the data at hand as we have done in the previous parts.

Question 3

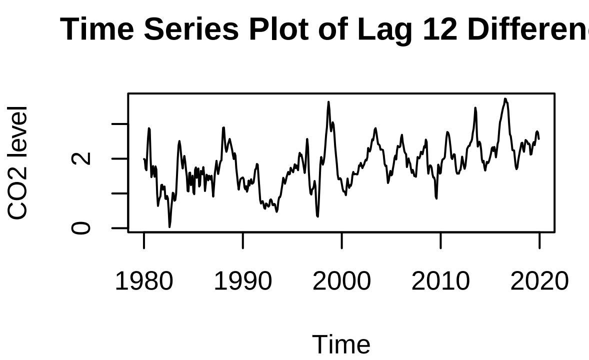
a)

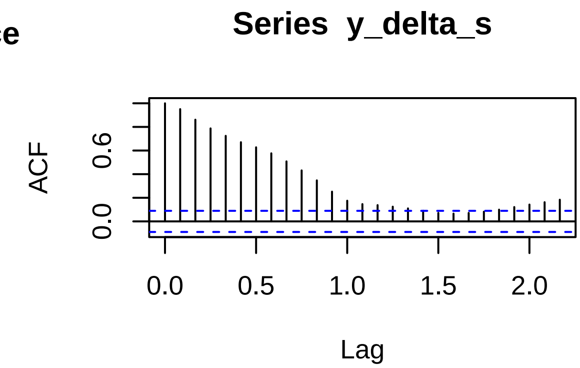
```r
q3data <- read.csv("co2_mm_gl.csv", header = TRUE, skip = 55)
co2ts <- ts(q3data[,4], start = c(1979,1), frequency = 12)
training <- window(co2ts,start = c(1979,1),end = c(2019,12),frequency=12)
testing <- window(co2ts,start = c(2020,1),end = c(2022,10),frequency=12)
```

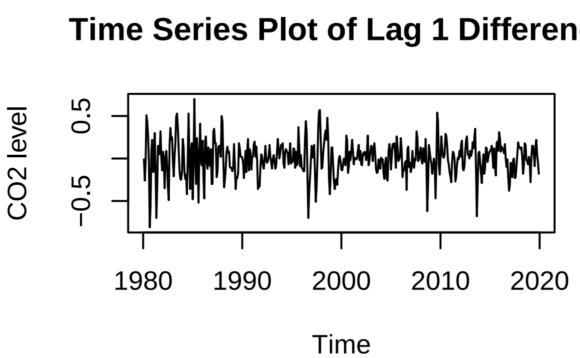
b)

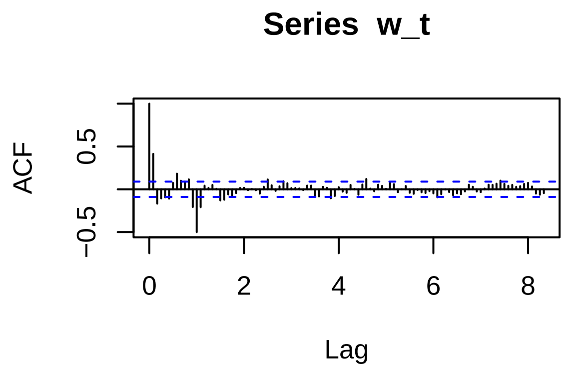
```r
par(mfrow = c(2,2))
s=12
y_delta_s = diff(training, lag = s, difference = 1)
plot(y_delta_s,
     ylab = "CO2 level",
     main = "Time Series Plot of Lag 12 Difference")
acf(y_delta_s)

#
w_t = diff(y_delta_s, lag = 1, difference = 1)
plot(w_t,
     ylab = "CO2 level",
     main = "Time Series Plot of Lag 1 Difference")
acf(w_t, lag.max = 100)
```









i) Using $s = 12$, the time series plot indicates an upward trend, and is now void of seasonal variation. The ACF plot has a slow exponential decay which reflects the positive temporal dependence observed in the differenced series.

ii) Looking at the time series plot of lag 12 difference, the time series stil possesses a upward trend. Therefore to remove the trend component, we difference the time series again at lag 1. After taking lag 1 difference, the new time series plot resembles a WN process, and the correlogram has a significant value at lag 1.

iii) I would choose

$$d = 1,\ D = 1,\ s = 12$$

iv) I would choose

$$p = 0,\ q = 0,\ P = 0,\ Q = 1$$

since there is a significant auto-correlation value at lag 1 after removing trend.

v)

```
p = 0
P = 0
q = c(0:5)
Q = c(0:5)
d = 1
D = 1
s = 12
sequence = c(rep(0,6),rep(1,6),rep(2,6),rep(3,6),rep(4,6),rep(5,6))
mods <- data.frame(p = rep(0,36),
                   d = rep(1,36),
```

```r
                q = sequence,
                P = rep(0,36),
                D = rep(1,36),
                Q = rep(0:5,6),
                AIC = rep(0,36))
for (i in 1:36) {
  q = mods$q[i]
  Q = mods$Q[i]
  mod <- arima(training,
               order=c(p, d, q),
               seasonal=list(order=c(P, D, Q),period=s))
  mods$AIC[i] <- mod$aic
}
print(mods[which.min(mods$AIC),])
```

```
##    p d q P D Q     AIC
## 20 0 1 3 0 1 1 -684.06
```

By compute AIC for all combinations of $q \in \{0, ..., 5\}$ and $Q \in \{0, ..., 5\}$, It appears the

$$SARIMA(0,1,3) \times (0,1,1)_{12}$$

has the lowest AIC, $AIC = -684.06$.

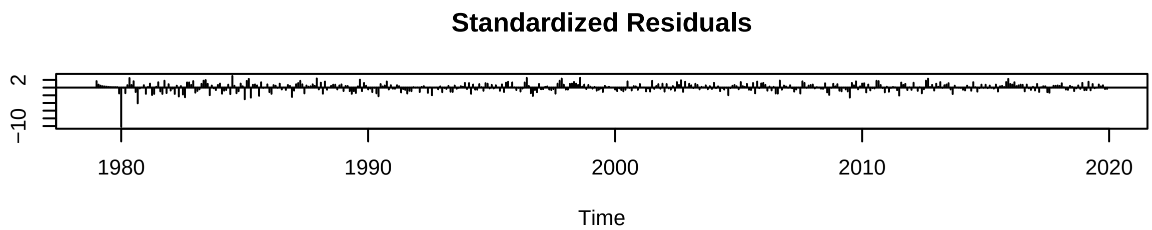
c)

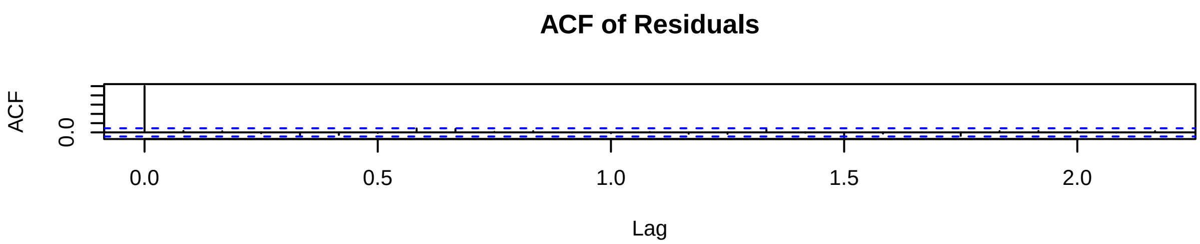
```r
model <- arima(training,
               order = c(0,1,3),
               seasonal = list(order = c(0,1,1),periods = 12))
print(model)
```

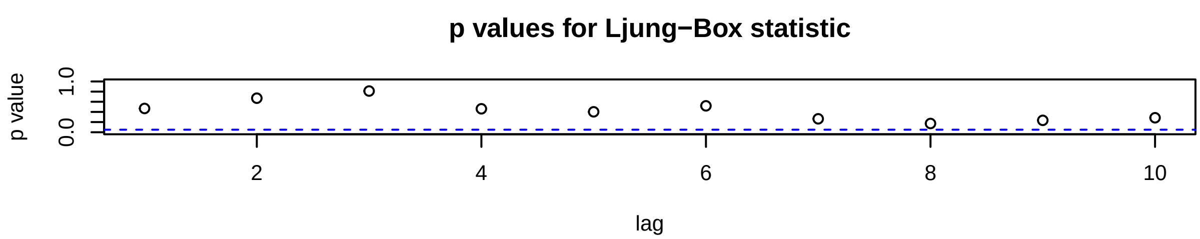
```
##
## Call:
## arima(x = training, order = c(0, 1, 3), seasonal = list(order = c(0, 1, 1),
##     periods = 12))
##
## Coefficients:
##          ma1      ma2      ma3     sma1
##       0.8302  -0.1329  -0.1797  -0.8622
## s.e.  0.0457   0.0613   0.0460   0.0261
##
## sigma^2 estimated as 0.01325:  log likelihood = 347.03,  aic = -684.06
```

d)

```r
tsdiag(model)
```



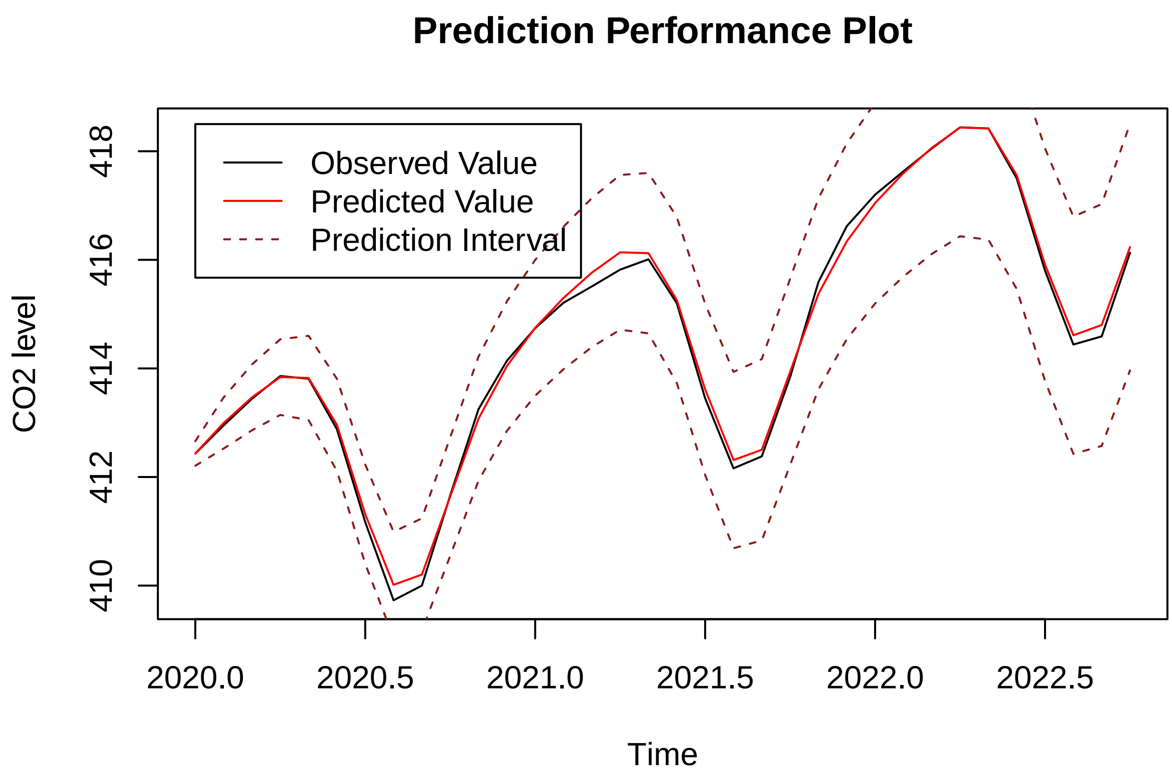




There is one standardized residual at the year 1980 fall outside of the ± 2 range, all other standardized residuals appeared normal. All acf values are non-significant except for at lag 0 which is expected. Ljung-Box statistics are not significant for the first 10 lags. All of the above suggests the model fit is good.

e)

```
pred <- predict(model,n.ahead = 34, prediction.interval = TRUE, level = 0.95)
plot(testing, type = "l", pch = 19, main = "Prediction Performance Plot" ,ylab = "CO2 level")
lines(pred$pred, col = "red", type = "l", pch = 19)
lines(pred$pred - 1.96*pred$se, col = "firebrick4", lty = "dashed")
lines(pred$pred + 1.96*pred$se, col = "firebrick4", lty = "dashed")
legend(2020.0,
       418.5,
       legend = c("Observed Value",
                  "Predicted Value",
                  "Prediction Interval"),
       lty = c("solid","solid","dashed"),
       col = c("black", "red","firebrick4"))
```



Looking at the above plot, we can see that the predicted values fall very close to the observed co2 level which suggest the forcasting procedure has performed relatively well.