Time series Lab 2

Omkar Bhutra (omkbh878)

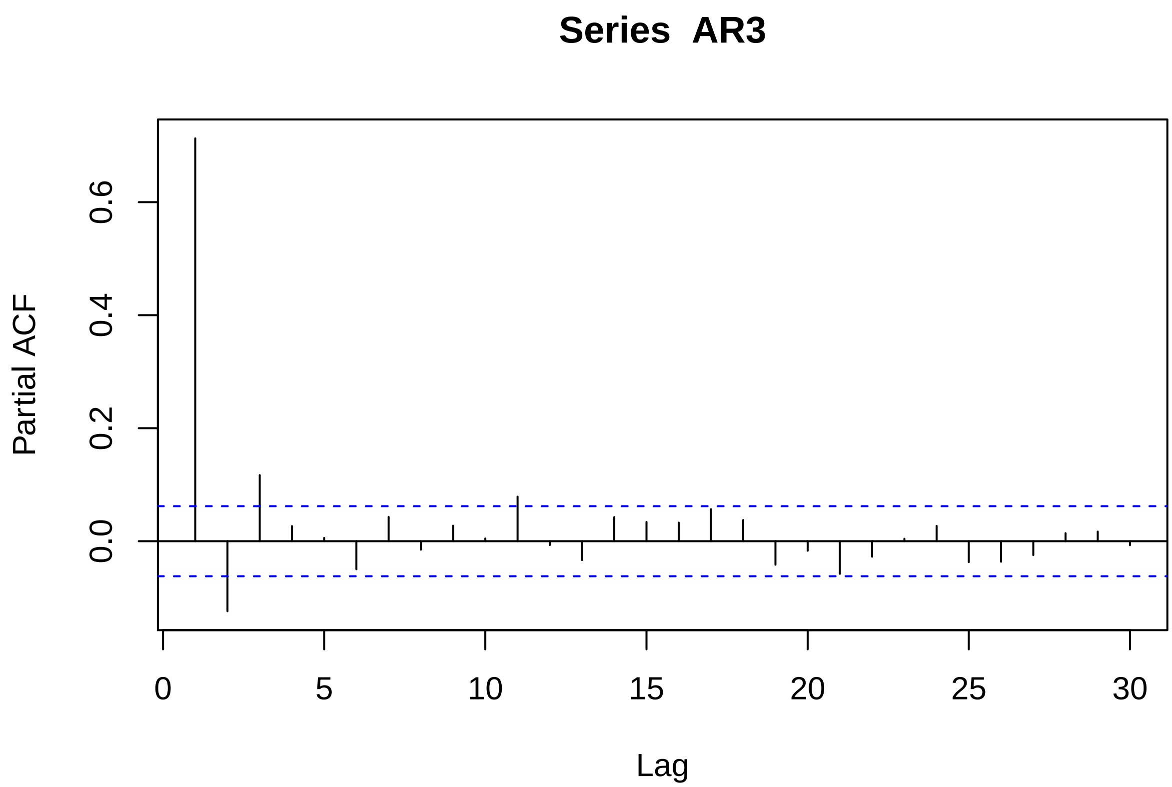
24 September 2019

Assignment 1. Computations with simulated data

1a. Generate 1000 observations from AR(3) process with phi's 1=0.8, 2=−0.2, 3=0.1. Use these data and the definition of PACF to compute 33 from the sample, i.e. write your own code that performs linear regressions on necessarily lagged variables and then computes an appropriate correlation. Compare the result with the output of function pacf() and with the theoretical value of 33

1000 observations are generated from $AR(3)$ with $\phi_1 = 0.8\phi_2 = -0.2\phi_3 = 0.1$ with the *arima.sim*. And we used the correlation from the first lag from the *PACF* from the generated data to calculate the theoretical PACF using the *ARMAacf* function. To compare the theoretical with the simulated data we used a linear regression we created a $dataframe$. X is the simulated data from $ARIMA(0.8, -0.2, 0.1)$, $X1$ is X minus one lag, $X2$ is X minus 2 lags and $X3$ is x minus 3 lags. Then saving the residuals from residuals from the linear regression where X is the dependent variable explained by $x1$ and $x2$. Also when $x3$ is the dependent variable explained by $x1$ and $x2$. The manual calculated is preformed by the correlation between the both residuals from the linear regressions.

```
set.seed(12345)
#simulate
AR3 <- arima.sim(list(ar=c(0.8,-0.2,0.1)), n=1000)
#theoretical pacf
AR3pacf <- pacf(AR3)
```




```
AR3data <-  ts.intersect(x = AR3,x1=stats::lag(AR3,1), x2=stats::lag(AR3,2), x3=stats::lag(AR3,3))

AR_lm <- lm(x ~ x1+x2,data=AR3data)
AR_lm_lag3 <- lm(x3 ~ x2+x1,data=AR3data)
r1=residuals(AR_lm)
r2=residuals(AR_lm_lag3)
cor(cbind(r1,r2))
```

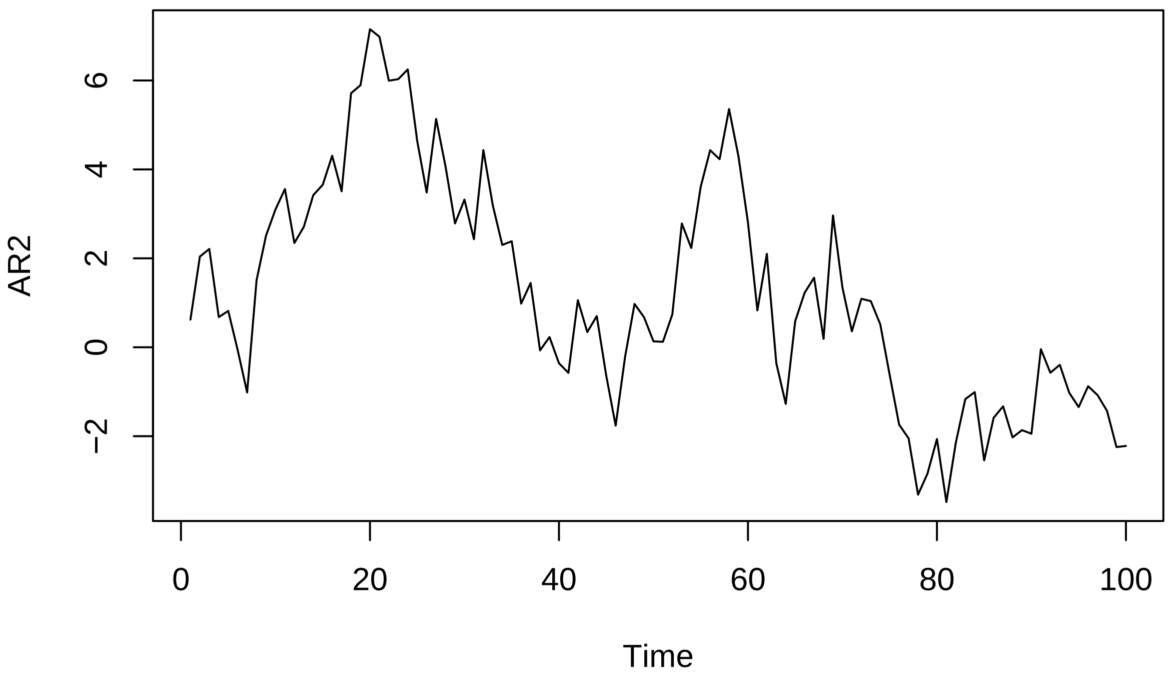
```
##           r1        r2
## r1 1.0000000 0.1146076
## r2 0.1146076 1.0000000
```

```
cat(paste("The theoretical pacf by lag 3:", cor(cbind(r1,r2))[1,2]))
```

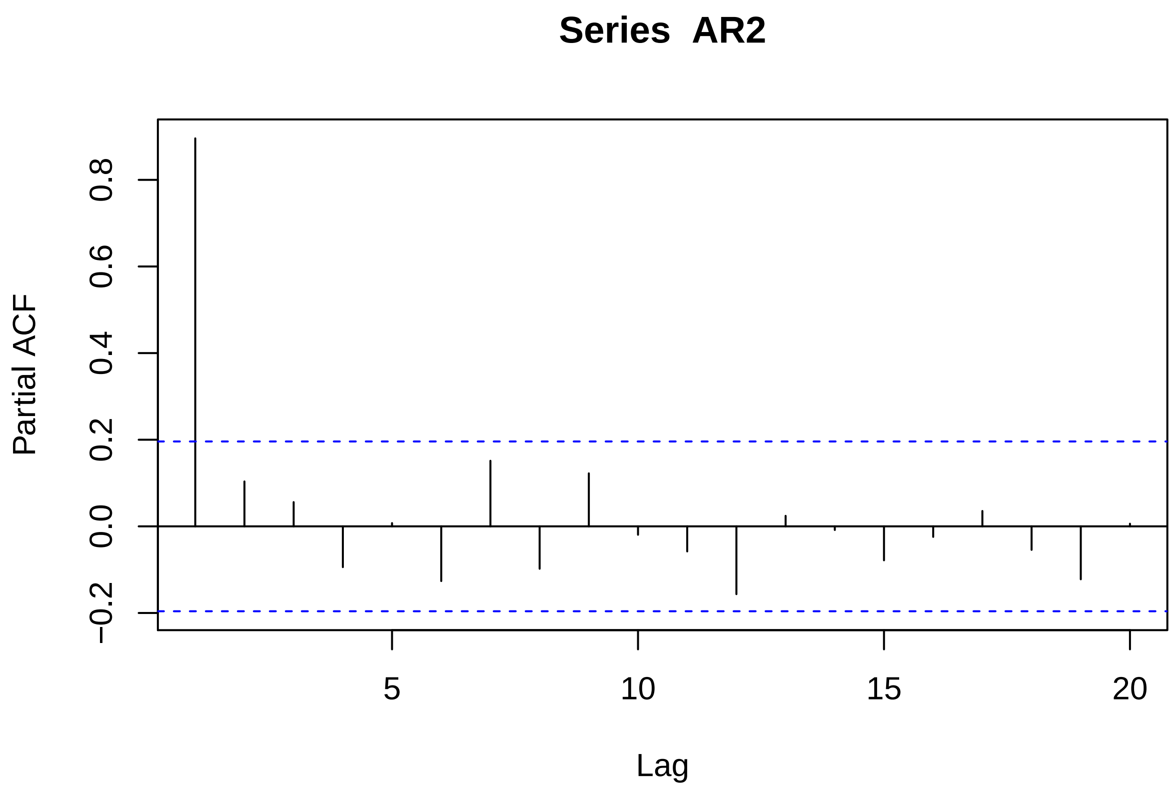
```
## The theoretical pacf by lag 3: 0.114607639249897
```

1b. Simulate an AR(2) series with phi's 1=0.8, 2=0.1 and n =100. Compute the estimated parameters and their standand errors by using three methods: method of moments (Yule-Walker equations), conditional least squares and maximum likelihood (ML) and compare their results to the true values. Which method does seem to give the best result? Does theoretical value for phi 2 fall within confidence interval for ML estimate?

```r
set.seed(12345)
AR2 <- arima.sim(list(ar=c(0.8,0.1)), n=100)
plot(AR2)
```




```r
pacf(AR2)
```




```r
paste("method of moments (Yule-Walker equations)")
```

```
## [1] "method of moments (Yule-Walker equations)"
```

```r
yulewalker <- ar.yw(AR2, aic=F, order.max=2)
paste("estimated parameters")
```

```
## [1] "estimated parameters"
```

```r
yulewalker$ar
```

```
## [1] 0.8029146 0.1037053
```

```r
paste("standard errors")
```

```
## [1] "standard errors"
```

```r
yulewalker$asy.var.coef
```

```
##              [,1]         [,2]
## [1,]  0.010198404 -0.009135888
## [2,] -0.009135888  0.010198404
```

```r
paste("conditional least squares")
```

```
## [1] "conditional least squares"
```

```r
condleastsquares <- arima(AR2,order=c(2,0,0), method = c("CSS"))
paste("coefficient estimates")
```

```
## [1] "coefficient estimates"
```

```r
condleastsquares$model$phi
```

```
## [1] 0.8066846 0.1205352
```

```r
paste("standard errors")
```

```
## [1] "standard errors"
```

```r
condleastsquares$var.coef
```

```
##                    ar1         ar2    intercept
## ar1        0.009691316 -0.00883412 -0.004071024
## ar2       -0.008834120  0.00988207 -0.013287047
## intercept -0.004071024 -0.01328705  2.290286649
```

```r
paste("maximum likelihood")
```

```
## [1] "maximum likelihood"
```

```r
maxliklihood <- ar.mle(AR2, aic=F, order.max=2)
paste("ar coefficient estimates")
```

```
## [1] "ar coefficient estimates"
```

```r
maxliklihood$ar
```

```
##       ar1       ar2
## 0.7968774 0.1189369
```

```r
paste("standard errors")
```

```
## [1] "standard errors"
```

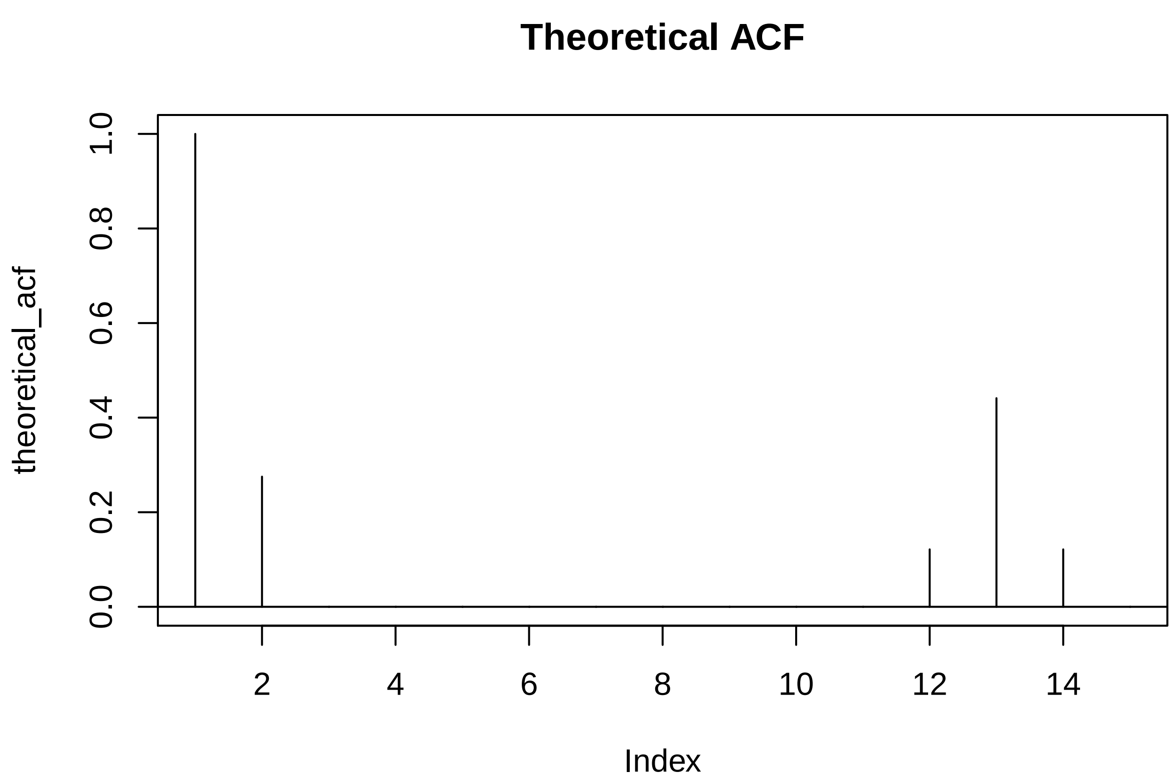
```r
maxliklihood$asy.var.coef
```

```
##              [,1]         [,2]
## [1,]  0.009072681 -0.008127448
## [2,] -0.008127448  0.009072681
```

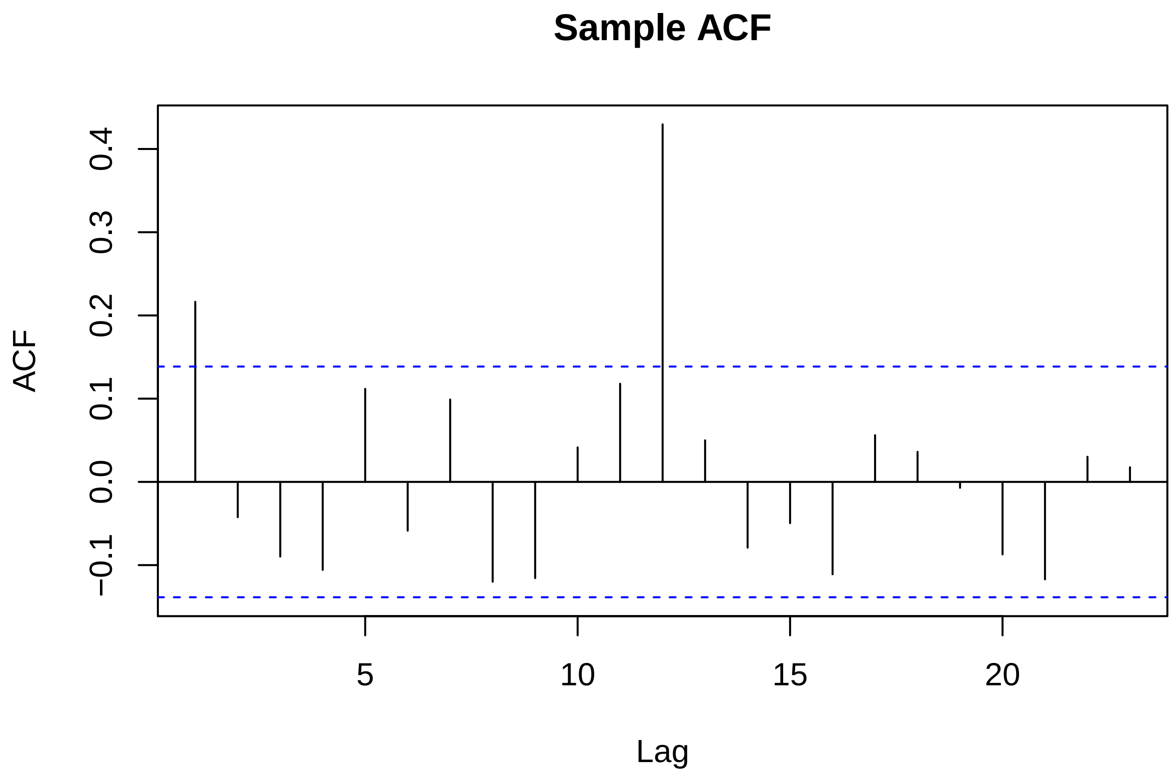
The results from the Yule walker seem to be the best parameters obtained. Yes, theoretical phi_2 is inside the confidence interval for the maximum liklihood estimate.

1c. Generate 200 observations of a seasonal ARIMA (0,0,1)×(0,0,1)12 model with coefficients Θ=0.6 and =0.3 by using arima.sim(). Plot sample ACF and PACF and also theoretical ACF and PACF. Which patterns can you see at the theoretical ACF and PACF? Are they repeated at the sample ACF and PACF?

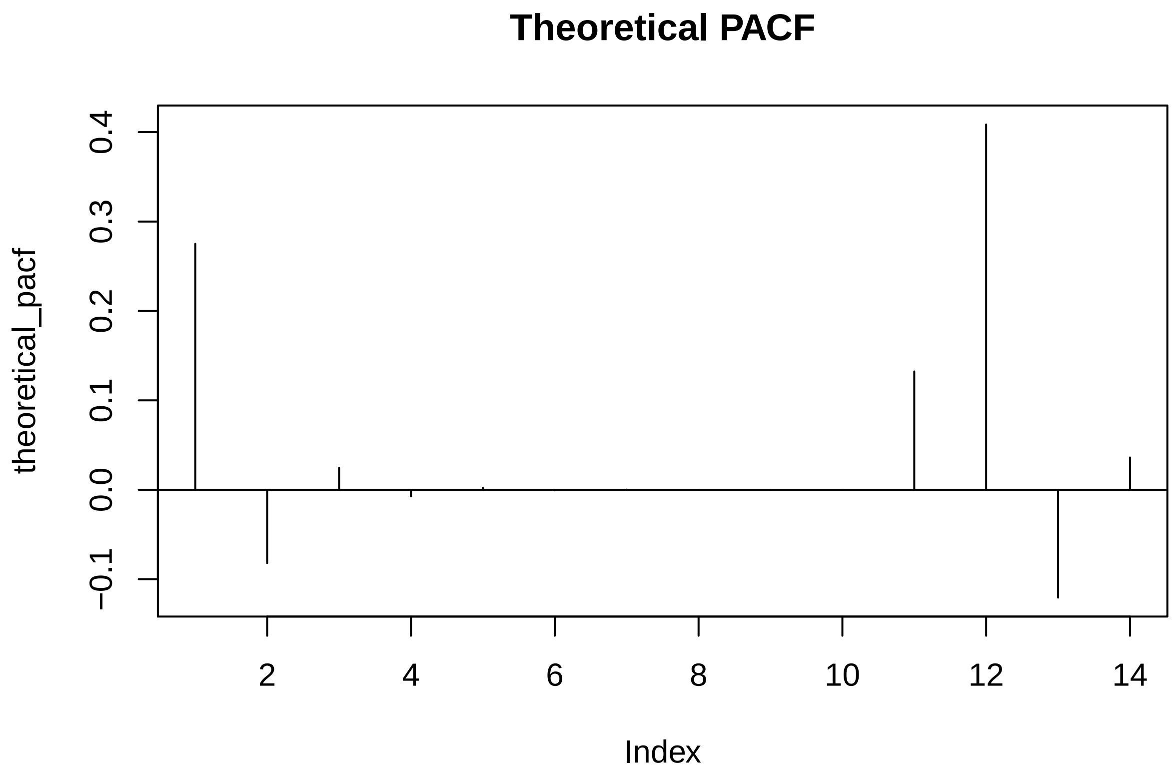
```
set.seed(12345)
ma.coeff <- c(0.3,rep(0,10),0.6)
ar <- arima.sim(n=200, model = list(order=c(0,0,12), ma = ma.coeff))
#ar.seasonal<- sarima(ar, 0, 0, 1, P = 0, D = 0, Q = 1, S = 12,Model = TRUE)
theoretical_acf <- ARMAacf(ma = c(ma.coeff,0.3*0.6))
theoretical_pacf <- ARMAacf(ma = c(ma.coeff,0.3*0.6),pacf = TRUE)


plot(theoretical_acf, type="h", main="Theoretical ACF")
abline(h=0)
```



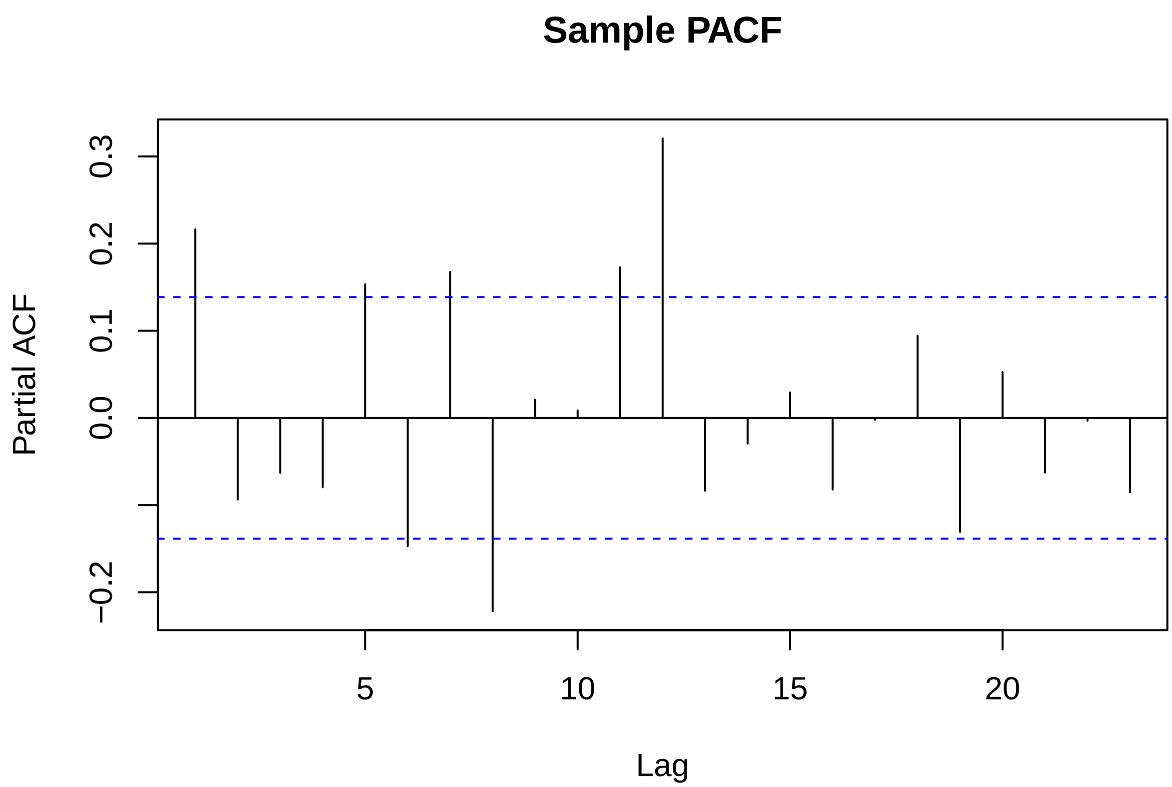

```
sample_acf <- acf(ar, main="Sample ACF")
```




```r
plot(theoretical_pacf, type="h", main="Theoretical PACF")
abline(h=0)
```




```r
sample_pacf <- pacf(ar, main="Sample PACF")
```



The ACF patterns are similar between the theoretical and sample observations. In theoretical ACF, there are large spikes at lags 1 and 13 while in sample ACF we have large spikes at 1 and 12 suggesting some correlation exits along the lags. In PACF patterns, the spikes are at lags 1 and 12 for both theoretical and sample.

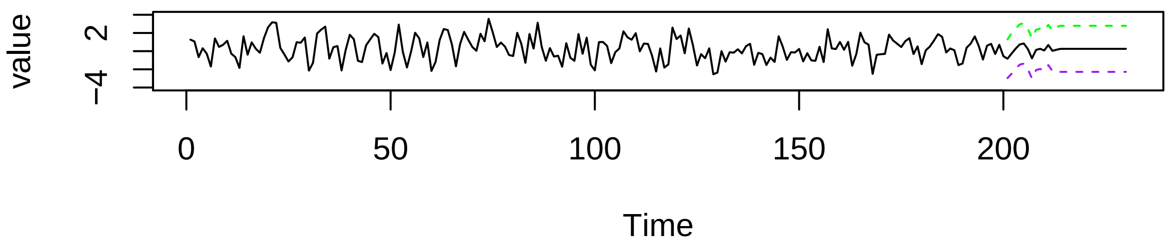
1c. Generate 200 observations of a seasonal ARIMA (0,0,1)×(0,0,1)12 model with coefficients Θ=0.6 and =0.3 by using arima.sim(). Fit (0,0,1)×(0,0,1)12 model to the data, compute forecasts and a prediction band 30 points ahead and plot the original data and the forecast with the prediction band. Fit the same data with function gausspr from package kernlab (use default settings). Plot the original data and predicted data from =1 to =230. Compare the two plots and make conclusions.

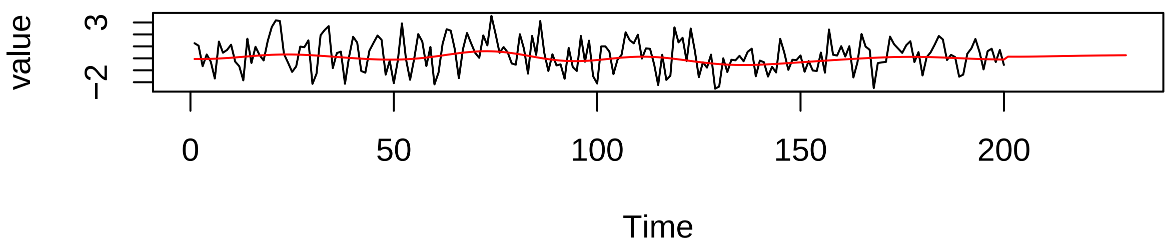
```
set.seed(12345)
ma.coeff <- c(0.3,rep(0,10),0.6)
ar <- arima.sim(n=200, model = list(order=c(0,0,12), ma = ma.coeff))
ar_fit <- arima(ar,order = c(0,0,1),seasonal = list(order = c(0,0,1),period = 12))
ar_pred <- predict(ar_fit,n.ahead = 30, se.fit = TRUE)

#gausspr is an implementation of Gaussian processes for classification and regression
gausspr_data <- data.frame(y = ar, x = 1:200)
gausspr_fit <- kernlab::gausspr(y ~ x, gausspr_data)
```

```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

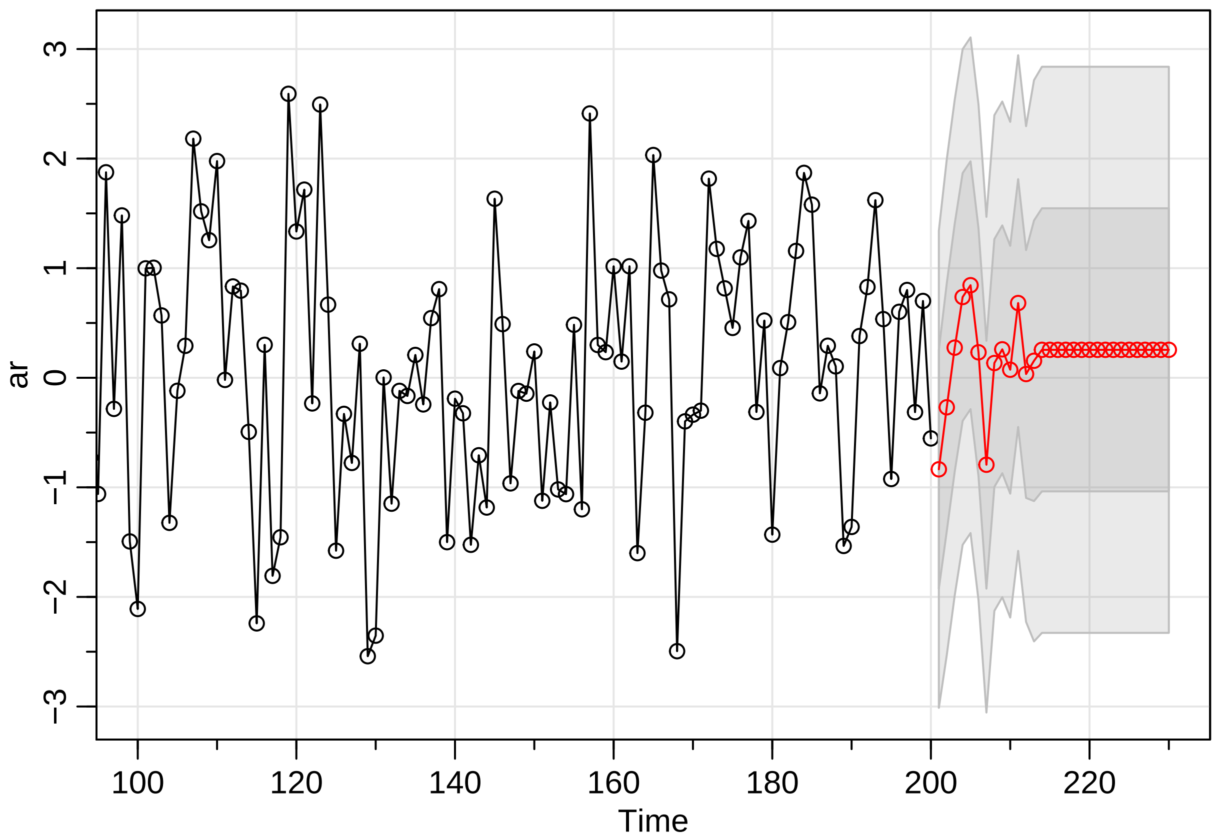
```
gausspr_pred <- predict(gausspr_fit, data.frame(x=201:230))

par(mfrow=c(2, 1))
```

```
plot(ts(c(ar, ar_pred$pred)), ylim=c(-4, 4), ylab="value", title = "Moving average model with predictio
lines(200 + 1:length(ar_pred$pred), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="green")
lines(200 + 1:length(ar_pred$pred), ar_pred$pred - 1.96 * ar_pred$se, lty=2, col="purple")
plot(ar, xlim=c(0, 230), ylab="value",title = "Moving average model")
lines(c(fitted(gausspr_fit), gausspr_pred), , col="red")
```






```
par(mfrow=c(1, 1))
#sarima for prediciton
sarima.for(ar,30,0,0,1,0,0,1,12)
```

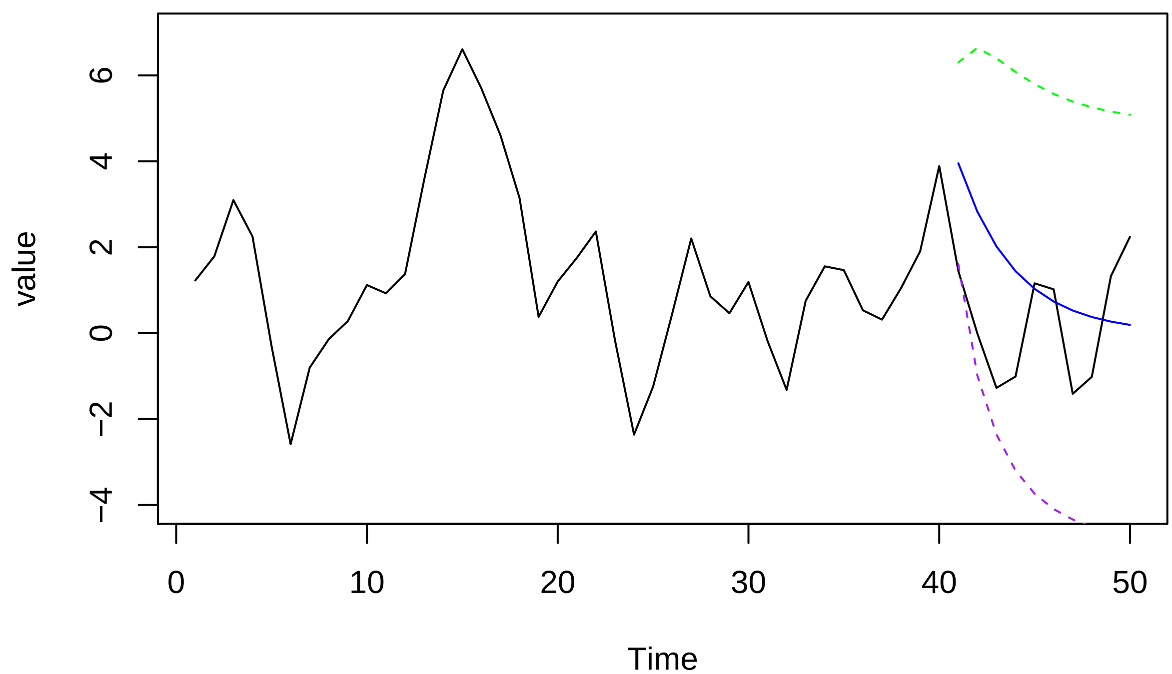



```
## $pred
## Time Series:
## Start = 201
## End = 230
## Frequency = 1
##  [1] -0.83557722 -0.26951120  0.27478352  0.73678068  0.84455443
##  [6]  0.23224666 -0.79358727  0.13448054  0.25951986  0.07335756
## [11]  0.68187185  0.03408012  0.15501563  0.25496000  0.25496000
## [16]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
## [21]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
## [26]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
##
## $se
## Time Series:
## Start = 201
## End = 230
## Frequency = 1
##  [1] 1.088418 1.130806 1.130806 1.130806 1.130806 1.130806 1.130806
##  [8] 1.130806 1.130806 1.130806 1.130806 1.130806 1.280439 1.291578
## [15] 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578
## [22] 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578
## [29] 1.291578 1.291578
```

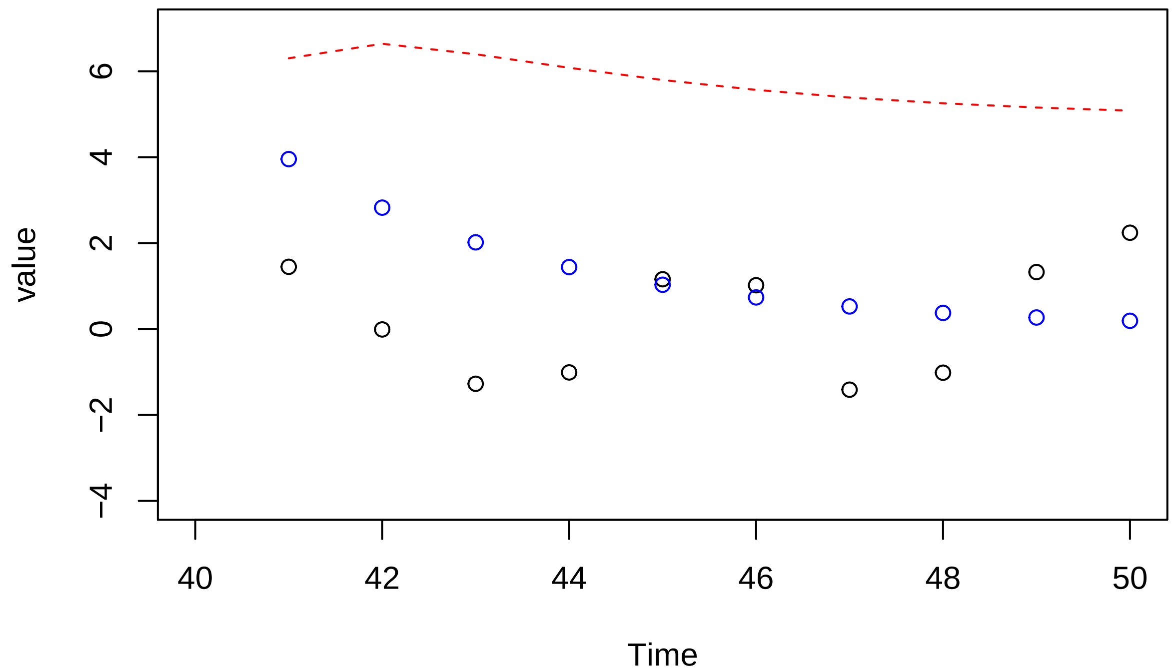
The prediction is reasonable for 12 months but then the model predicts the mean values for the next 18 months. The Gaussian process has an intial kink in the prediction and after that it remains linear due to the smooth fit to the observated data

1e.Generate 50 observations from ARMA(1,1) process with phi =0.7, =0.5. Use first 40 values to fit an ARMA(1,1) model with mu =0. Plot the data, the 95% prediction band and plot also the true 10 values that you initially dropped. How many of them are outside the prediction band? How can this be interpreted?

```
set.seed(12345)
ar <- arima.sim(model=list(ma=c(0.5), ar=c(0.7)), n=50)
train <- ts(ar[1:40])
test <- ts(ar[41:50])
ar_fit <- arima(train, order=c(1, 0, 1), include.mean = F)
ar_pred <- predict(ar_fit, n.ahead=10)

plot(ts(c(train, test)), ylim=c(-4, 7), type="l", ylab="value")
lines(40 + 1:length(test), ar_pred$pred, col="blue")
lines(40 + 1:length(test), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="green")
lines(40 + 1:length(test), ar_pred$pred - 1.96 * ar_pred$se, lty=2, col="purple")
```




```
plot(40 + 1:length(test), test, ylim=c(-4, 7), xlim=c(40, 50), type="p", ylab="value", xlab="Time")
points(40 + 1:length(test), ar_pred$pred, col="blue")
lines(40 + 1:length(test), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="red")
```

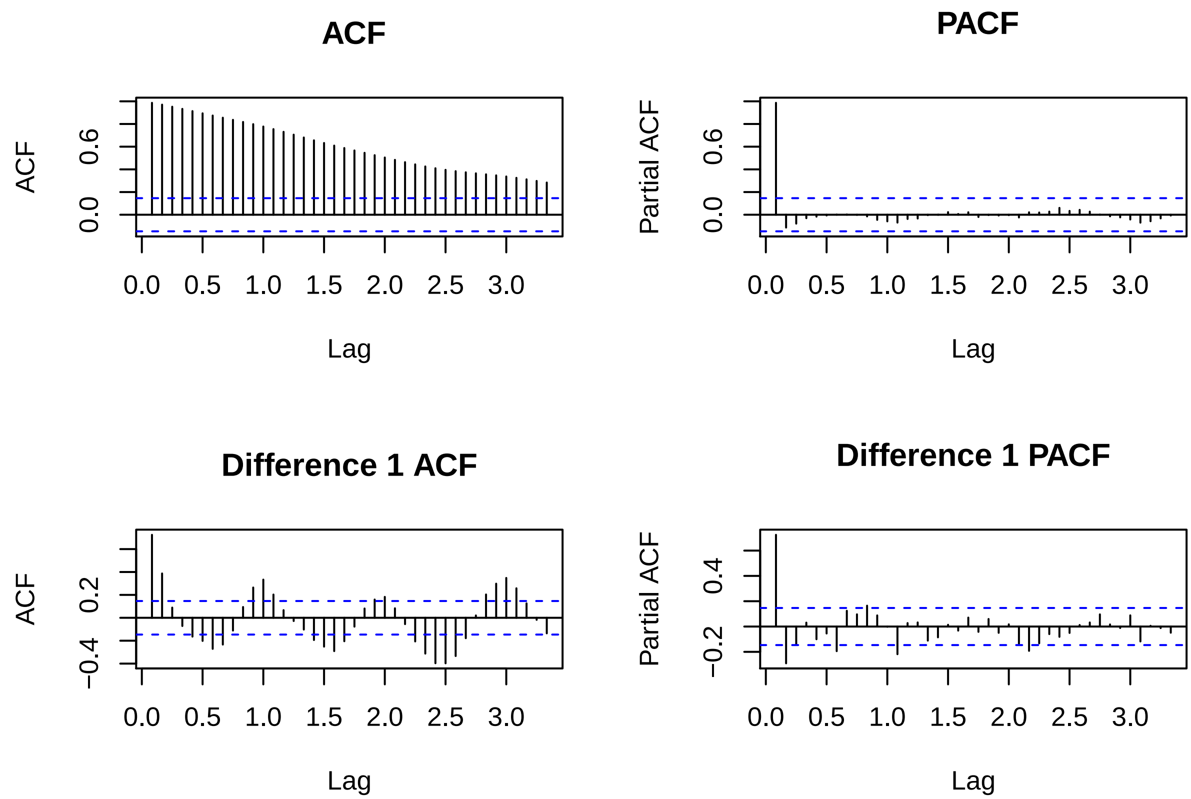


Only one observaation lies outside the confidence interval , this means that since we expected 5/100 observations to be outside the 95% confidence interval, it is reasonable to find 1/10 to be outside.

Assigntment 2. ACF and PACF diagnostics

```
acf_pacf_diag<- function(data){
    par(mfrow = c(2, 2))
    acf(data, lag.max = 40, main=" ACF")
    pacf(data, lag.max = 40, main=" PACF")
    acf(diff(data, lag = 1), lag.max = 40, main= "Difference 1 ACF")
    pacf(diff(data, lag = 1), lag.max = 40, main= "Difference 1 PACF")
    par(mfrow = c(2, 2))
}
```

```
acf_pacf_diag(chicken)
```



ACF :The ACF on the original data suggests an AR or ARMA model since the ACF tails off.

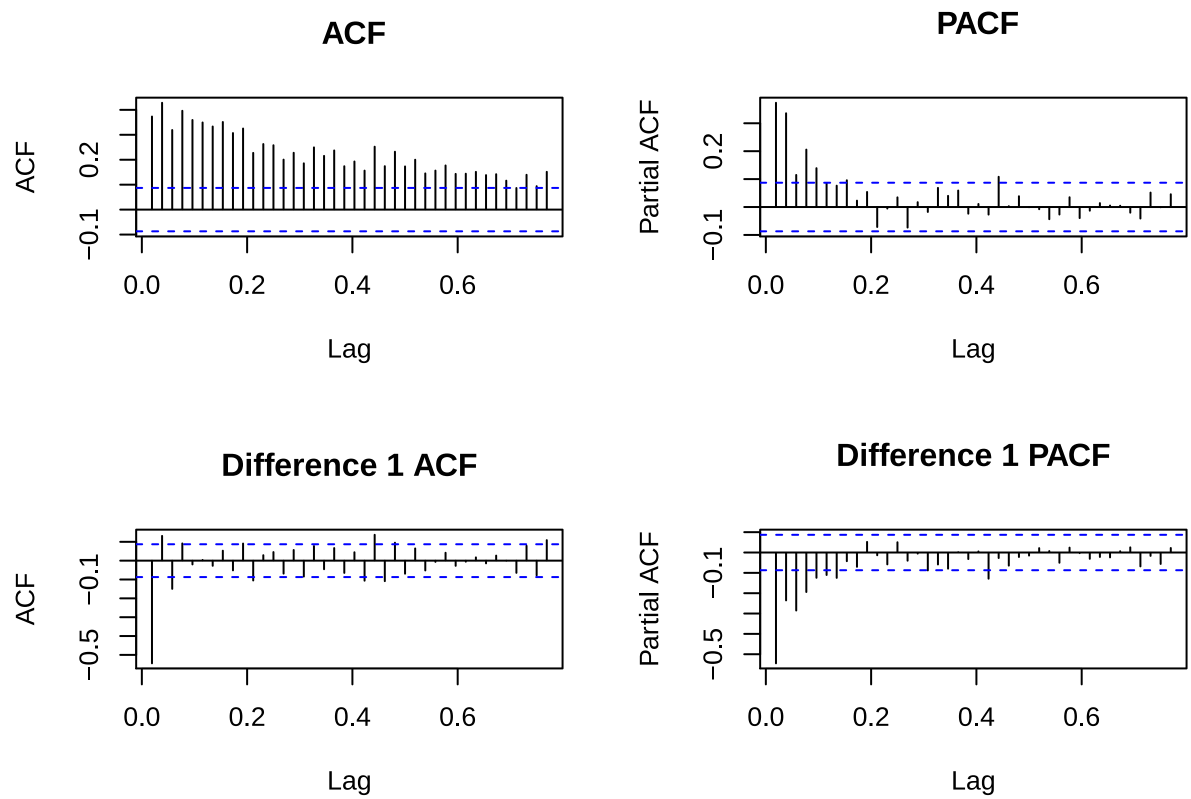
PACF: The PACF on the original data cuts off after lag 1 suggesting an AR(1) model.

Difference 1 ACF: 1st Differencing suggests seasonality in the data. The ACF tails off which suggests an AR model.

Difference 1 PACF: The PACF shows that the seasonality cuts off after lag 12 which is 1 * 12 indicating an AR(1) model.

$ARIMA(1,0,0)x(1,1,0)_{12}$ can be suggested.

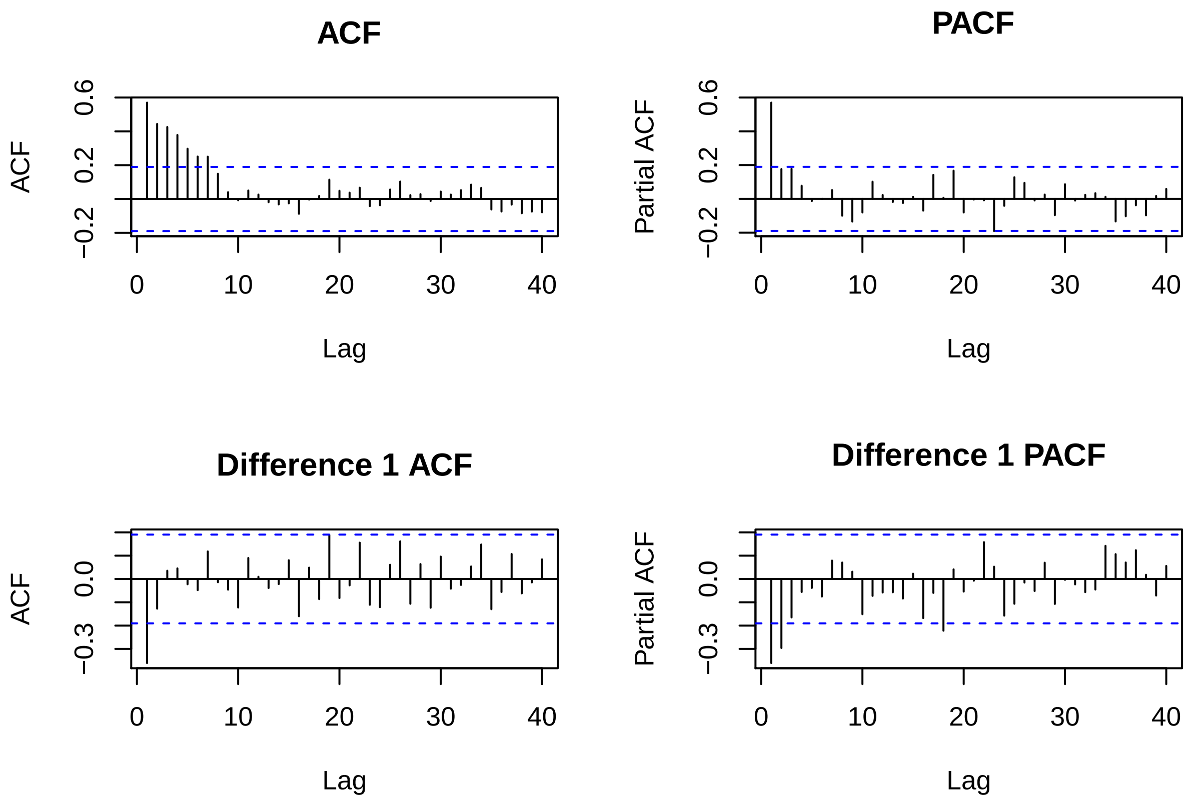
```
acf_pacf_diag(so2)
```



ACF: The ACF tails off suggesting either an AR or ARMA model. PACF: The PACF tails off , also suggesting an ARMA model. Difference 1 ACF: The ACF after difference cuts off after lag 1 suggesting a MA(1) model. Difference 1 PACF: The PACF after difference tails off further suggesting a MA(1) model.

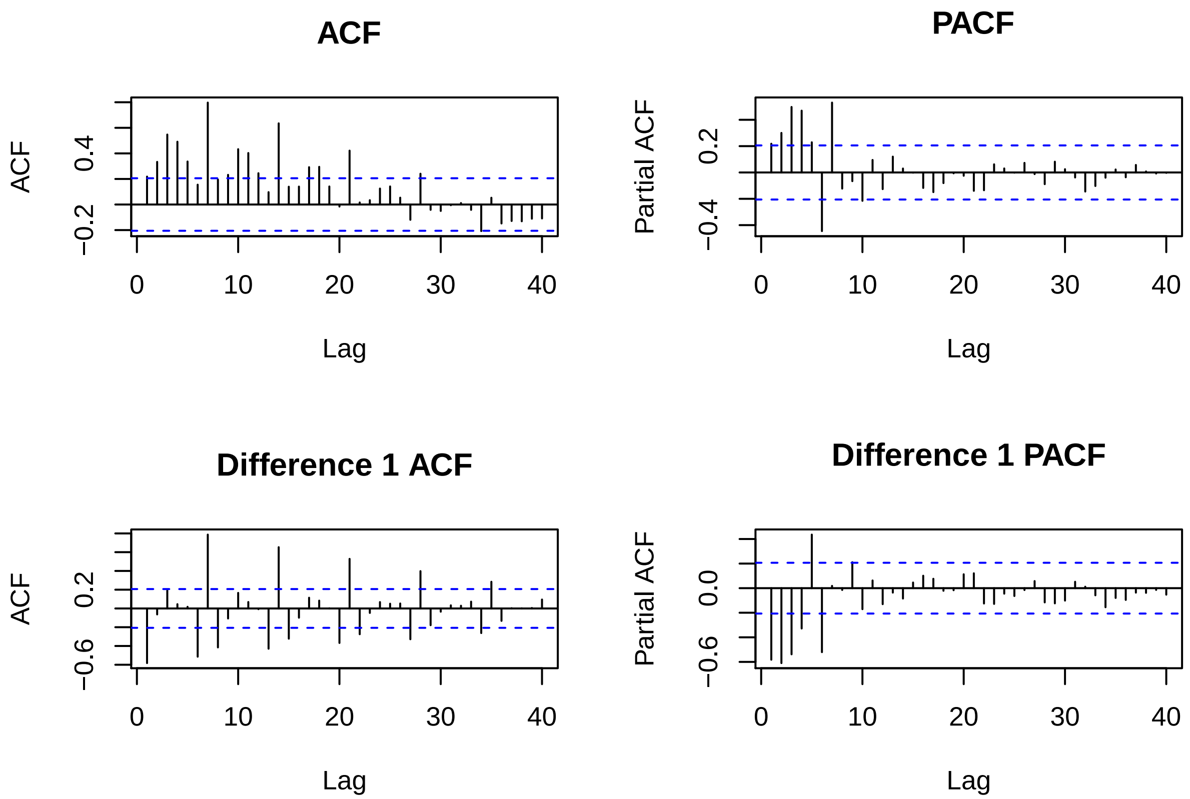
$ARIMA(0,1,1)$ can be suggested.

```r
acf_pacf_diag(EQcount)
```



ACF: The ACF tails off suggesting an AR or ARMA model. PACF: The PACF cuts off after lag 1 suggesting AR(1) model. Difference 1 Data ACF: The ACF after difference cuts off after lag 1 suggesting a MA(1) model. Difference 1 Data PACF: The PACF after difference tails off further suggesting a MA model. Either a ARMA(1, 0, 0) or ARMA(0, 1, 1) is suggested.

```r
acf_pacf_diag(HCT)
```



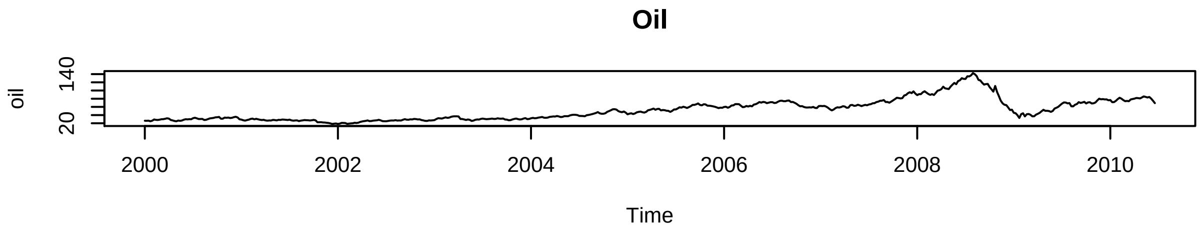
ACF: The ACF tails off suggesting either an AR or ARMA model. PACF: The PACF cuts off after lag 7 suggesting an AR(7) model. Difference 1 ACF: The ACF suggests seasonality that tails off after lag 7 suggesting an seasonality of 7. Difference 1 PACF: The PACF cuts off after 6 lags suggesting an AR(6) seasonality model.

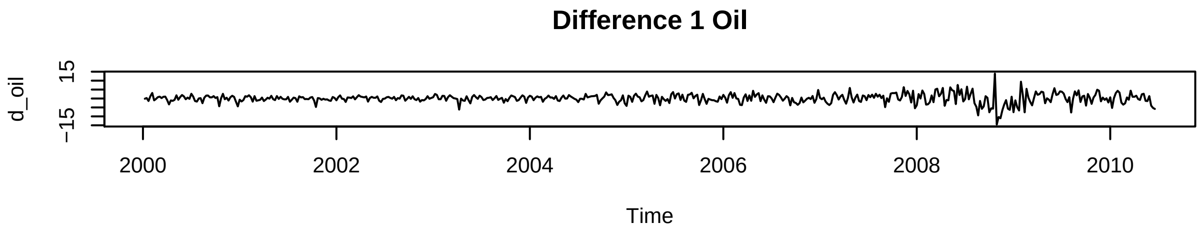
$ARIMA(7,0,0)X(1,1,0)_7$ model can be suggested.

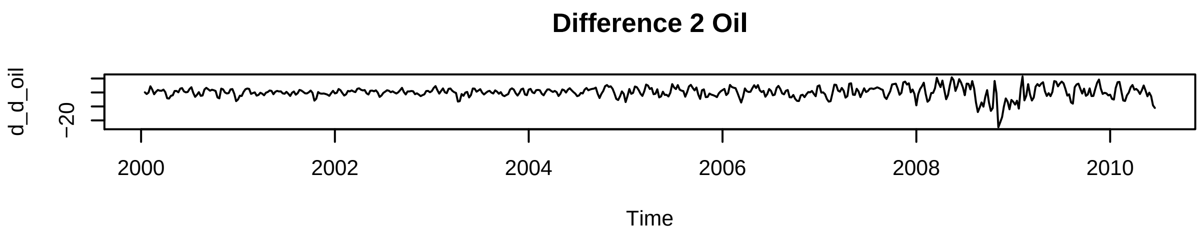
Assignment 3. ARIMA modelling cycle

Question 1

```r
log_oil <- log(oil)
d_oil <- diff(oil)
d_d_oil <- diff(oil, 2)
d_log_oil <- diff(log_oil)
d_d_log_oil <- diff(log_oil, 2)

par(mfrow=c(3, 1))
plot(oil, main="Oil")
plot(d_oil, main="Difference 1 Oil")
plot(d_d_oil, main="Difference 2 Oil")
```

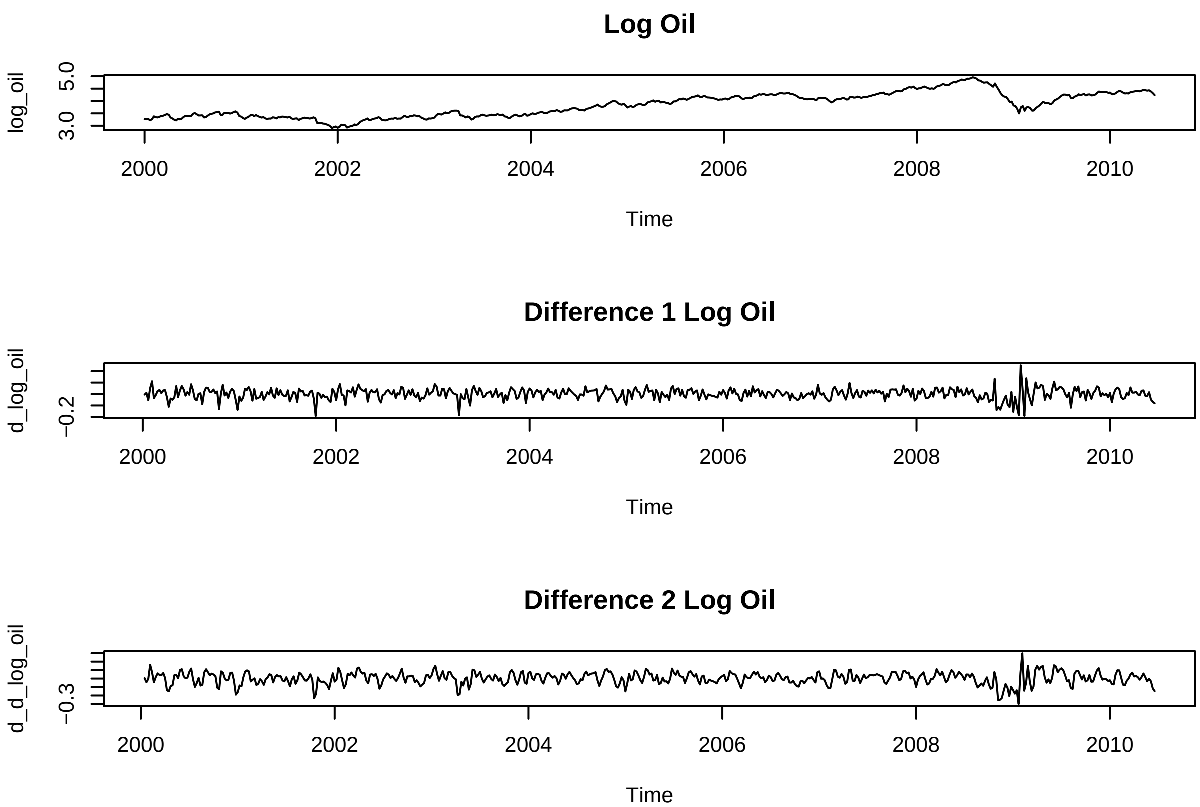







```
par(mfrow=c(3, 1))
```

```r
par(mfrow=c(3, 1))
plot(log_oil, main="Log Oil")
plot(d_log_oil, main="Difference 1 Log Oil")
plot(d_d_log_oil, main="Difference 2 Log Oil")
```




```r
par(mfrow=c(3, 1))
```

The logarithm transformed data is what we have to work with as it adjusts the scale appropriately for the price fluctuations. The first difference can be considered stationary and while the 2nd difference does not do much visual change. We can keep both models tentatively.

```r
adf.test(oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag     ADF p.value
## [1,]   0  0.0628   0.662
## [2,]   1 -0.1556   0.599
## [3,]   2 -0.1983   0.587
## [4,]   3 -0.3284   0.549
## [5,]   4 -0.3675   0.538
## [6,]   5 -0.4799   0.506
```

```
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -1.29   0.600
## [2,]   1 -1.51   0.521
## [3,]   2 -1.55   0.506
## [4,]   3 -1.74   0.432
## [5,]   4 -1.75   0.429
## [6,]   5 -1.82   0.401
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -1.77   0.674
## [2,]   1 -2.15   0.513
## [3,]   2 -2.23   0.481
## [4,]   3 -2.45   0.385
## [5,]   4 -2.53   0.353
## [6,]   5 -2.74   0.263
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
adf.test(log_oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 0.717   0.850
## [2,]   1 0.582   0.811
## [3,]   2 0.665   0.835
## [4,]   3 0.527   0.796
## [5,]   4 0.539   0.799
## [6,]   5 0.405   0.761
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -1.30   0.596
## [2,]   1 -1.46   0.539
## [3,]   2 -1.34   0.580
## [4,]   3 -1.63   0.477
## [5,]   4 -1.48   0.533
## [6,]   5 -1.46   0.539
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -2.03   0.563
## [2,]   1 -2.35   0.428
## [3,]   2 -2.16   0.509
## [4,]   3 -2.54   0.350
## [5,]   4 -2.42   0.399
## [6,]   5 -2.63   0.308
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
adf.test(d_log_oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -20.30    0.01
## [2,]   1 -16.56    0.01
## [3,]   2 -11.36    0.01
## [4,]   3 -10.77    0.01
## [5,]   4  -9.22    0.01
## [6,]   5  -9.19    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -20.31    0.01
## [2,]   1 -16.58    0.01
## [3,]   2 -11.38    0.01
## [4,]   3 -10.79    0.01
## [5,]   4  -9.23    0.01
## [6,]   5  -9.21    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -20.29    0.01
## [2,]   1 -16.57    0.01
## [3,]   2 -11.37    0.01
## [4,]   3 -10.78    0.01
## [5,]   4  -9.22    0.01
## [6,]   5  -9.20    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

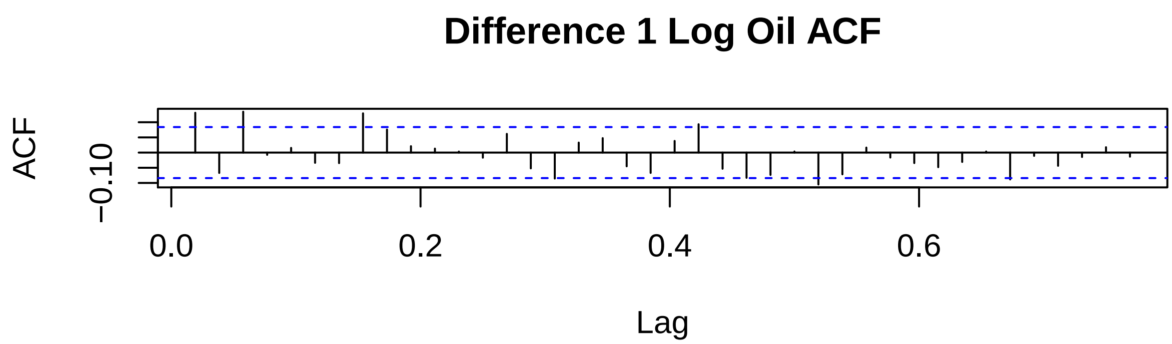
```r
adf.test(d_d_log_oil)
```

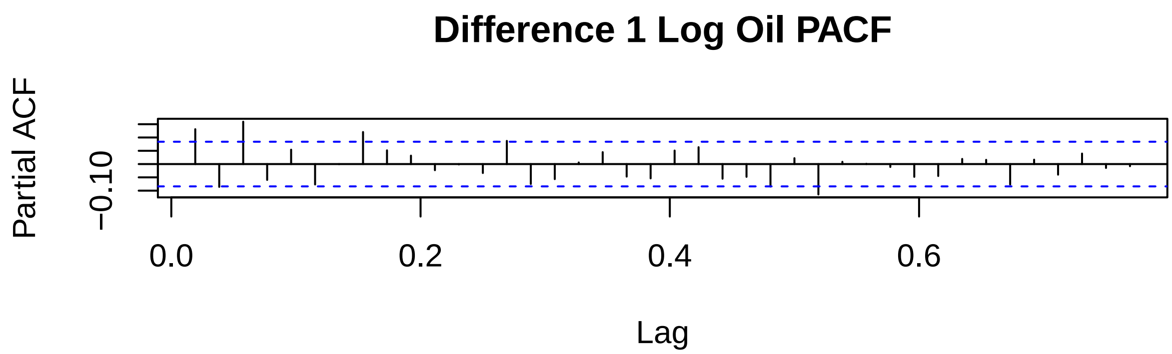
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -12.80    0.01
## [2,]   1 -15.34    0.01
## [3,]   2  -9.09    0.01
## [4,]   3 -10.93    0.01
## [5,]   4  -8.12    0.01
## [6,]   5  -9.56    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -12.81    0.01
## [2,]   1 -15.37    0.01
## [3,]   2  -9.11    0.01
## [4,]   3 -10.95    0.01
## [5,]   4  -8.14    0.01
## [6,]   5  -9.59    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -12.80    0.01
```

```
## [2,]   1 -15.36    0.01
## [3,]   2  -9.10    0.01
## [4,]   3 -10.94    0.01
## [5,]   4  -8.13    0.01
## [6,]   5  -9.58    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

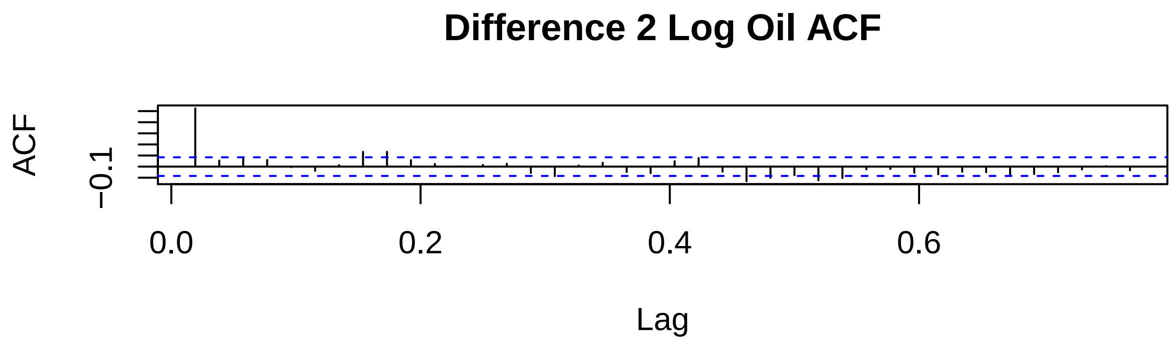
The augmented Dickey-Fuller test indicatest that we perform differencing to make the data stationary.

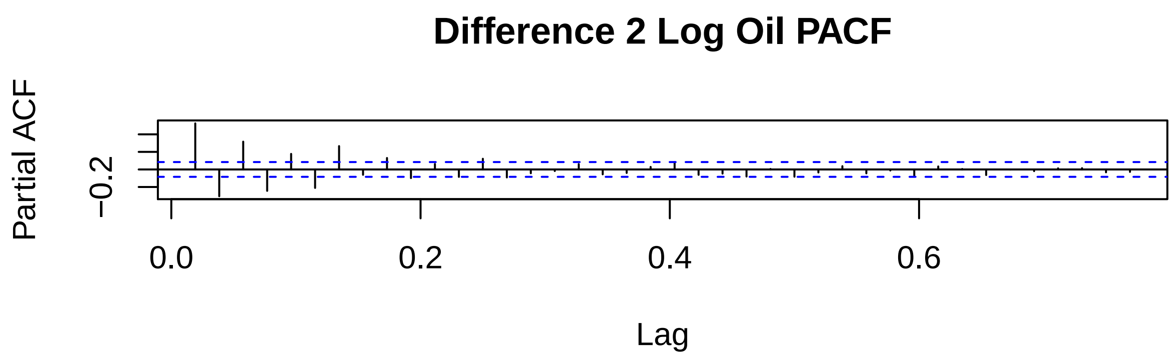
```r
par(mfrow=c(2, 1))
acf(d_log_oil, lag.max=40, main="Difference 1 Log Oil ACF")
pacf(d_log_oil, lag.max=40, main="Difference 1 Log Oil PACF")
```






```r
acf(d_d_log_oil, lag.max=40, main="Difference 2 Log Oil ACF")
pacf(d_d_log_oil, lag.max=40, main="Difference 2 Log Oil PACF")
```





ACF and PACF for the 1st order differencing do not have any clearly defined patterns compared to the 2nd order differencing, where it is observed that ACF cuts off at lag 1 and trails off in the PACF suggesting an ARIMA(0,2,1) model.

Computing the sample extended ACF: EACF analysis

```r
eacf(d_log_oil)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x o o o o x o o o  o  o  o
## 1 x o x o o o o x o o o  o  o  o
## 2 x x x o o o o x o o o  o  o  o
## 3 x x x o o o o x o o o  o  o  o
## 4 x o x o o o o x o o o  o  o  o
## 5 x x x o x o o x o o o  o  o  o
## 6 o x x o x x o x o o o  o  o  x
## 7 o x x x x x x x o x o  o  o  o
```

```r
eacf(d_d_log_oil)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o x x o o  o  o  o
## 1 x x o o o o o x x o o  o  o  o
```

```
## 2 x x x o o o o x x o o  o  o  o
## 3 x x x x o o o x x o o  o  o  o
## 4 x x x x o o o x x o o  o  o  o
## 5 x o x x x o o x x o o  o  o  o
## 6 x o x x x x x x o x o  o  o  o
## 7 x x x x x x x x o x o  o  o  o
```

The EACF matrix for 1st order differencing make a general pattern of rectangles as compared to triangles. The EACF matrix for the 2nd order differencing has the triangular pattern with a point at AR(1) which confirms the previous statements in the ACF.

```r
ljungbox<- function(data) {
print(Box.test(data, lag = 1, type = "Ljung-Box"))
}
ljungbox(oil)
```

```
##
##  Box-Ljung test
##
## data:  data
## X-squared = 541.1, df = 1, p-value < 2.2e-16
```

```r
fit1 <- Arima(log_oil, order=c(1, 1, 1))
fit1
```

```
## Series: log_oil
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.5253  0.7142
## s.e.   0.0872  0.0683
##
## sigma^2 estimated as 0.002112:  log likelihood=904.58
## AIC=-1803.15   AICc=-1803.11   BIC=-1790.25
```

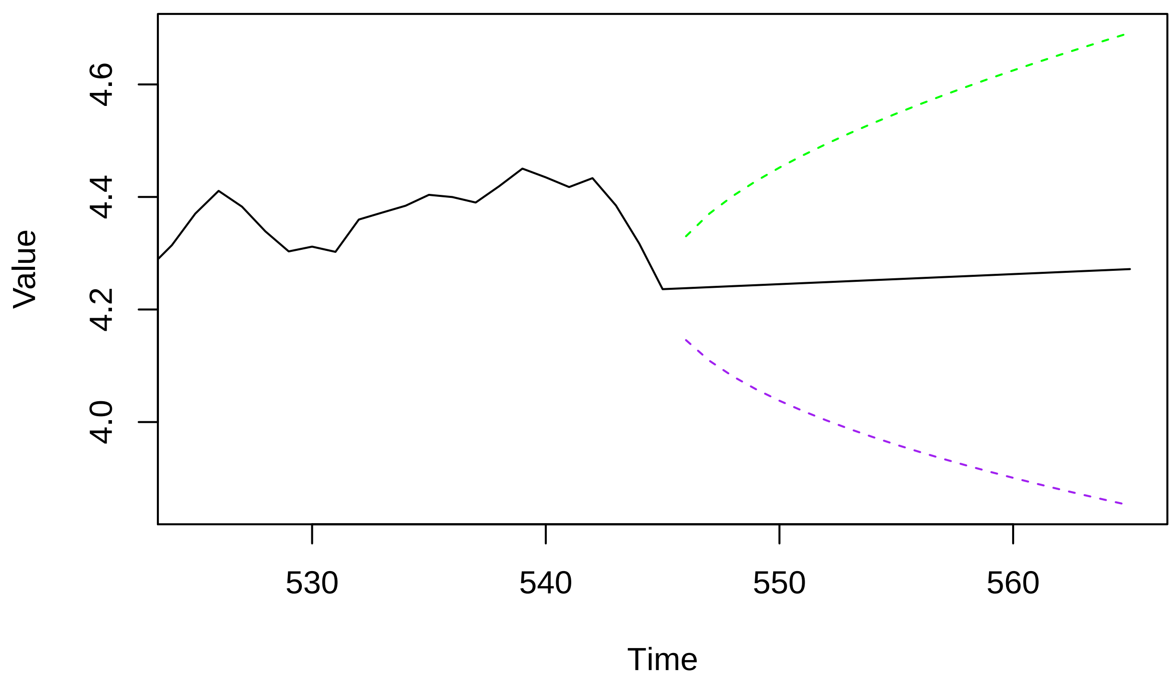
```r
fit2 <- Arima(log_oil, order=c(1, 2, 2))
fit2
```

```
## Series: log_oil
## ARIMA(1,2,2)
##
## Coefficients:
##           ar1      ma1      ma2
##       -0.5247  -0.2861  -0.7139
## s.e.   0.0873   0.0686   0.0685
##
## sigma^2 estimated as 0.002118:  log likelihood=899.7
## AIC=-1791.4   AICc=-1791.32   BIC=-1774.21
```

```r
fit3 <- Arima(log_oil, order=c(0, 2, 1))
fit3
```

```
## Series: log_oil
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -1.0000
## s.e.   0.0061
##
## sigma^2 estimated as 0.002213:  log likelihood=886.63
## AIC=-1769.26   AICc=-1769.24   BIC=-1760.67
```

```r
fit_plot <- function(model) {
    pred <- predict(model, n.ahead=20, se.fit=TRUE)
    upper_band <- pred$pred + 1.96 * pred$se
    lower_band <- pred$pred - 1.96 * pred$se
    n <- length(model$x)
    plot(c(model$x, pred$pred), type="l",
         xlim=c(n - 20, n + 20),
         ylim=c(min(lower_band), max(upper_band)), ylab="Value", xlab="Time")
    lines(n + 1:20, upper_band, lty=2, col="green")
    lines(n + 1:20, lower_band, lty=2, col="purple")
}
fit_plot(fit3)
```



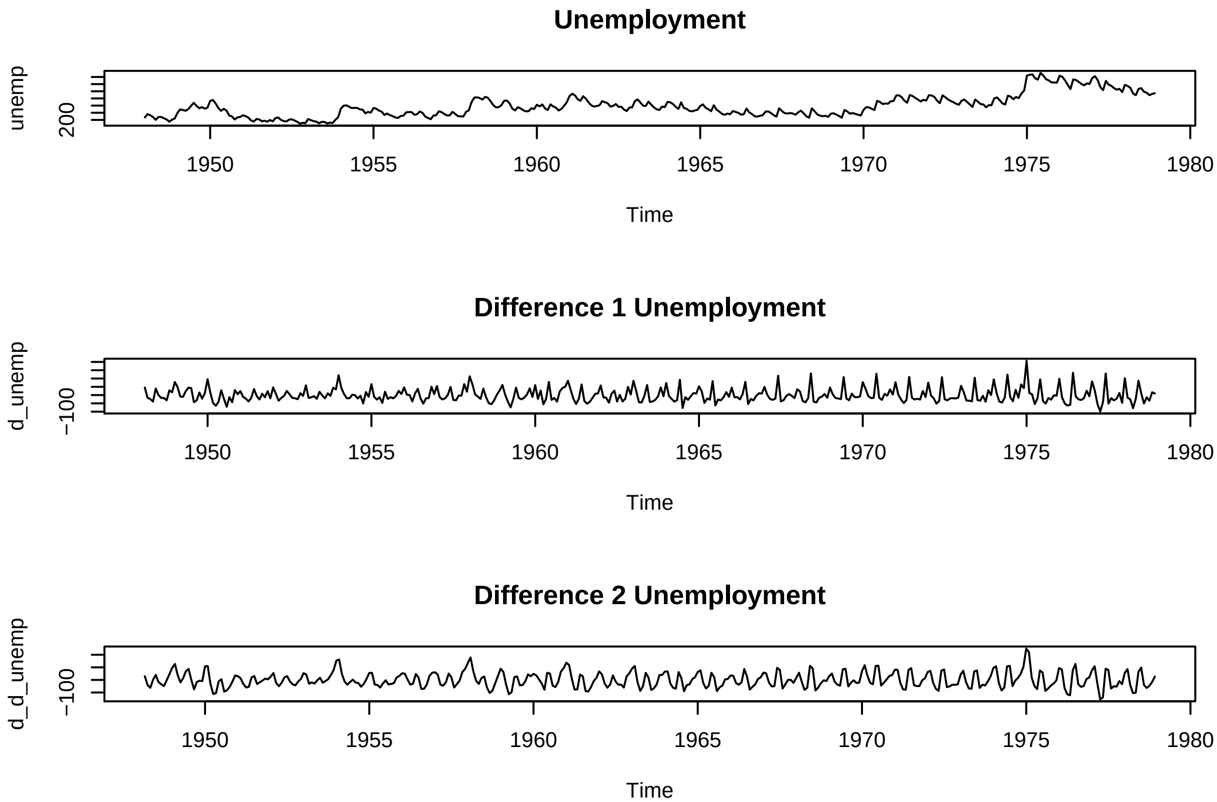
The best model if we evaluate the BIC criterion we choose the ARIMA(0,2,1) which looks like following

$$\nabla^2 x_t = (1 + \theta B) w_t$$

The fig. above displays a prediction of 20 timesteps. The prediction is marked by a kink and a linear prediction value. The intervals also look reasonable

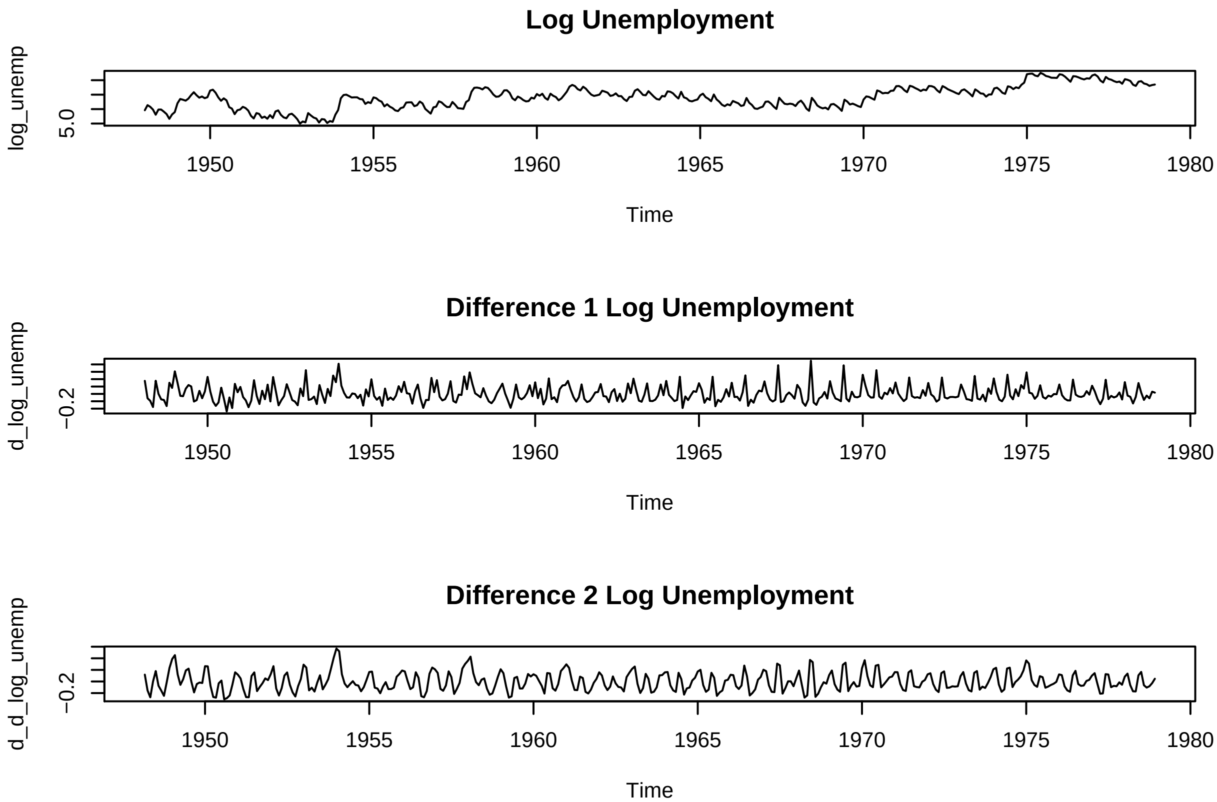
Question 2

```
log_unemp <- log(unemp)
d_unemp <- diff(unemp)
d_d_unemp <- diff(unemp, 2)
d_log_unemp <- diff(log_unemp)
d_d_log_unemp <- diff(log_unemp, 2)

par(mfrow=c(3, 1))
plot(unemp, main="Unemployment")
plot(d_unemp, main="Difference 1 Unemployment")
plot(d_d_unemp, main="Difference 2 Unemployment")
```




```r
par(mfrow=c(3, 1))
```

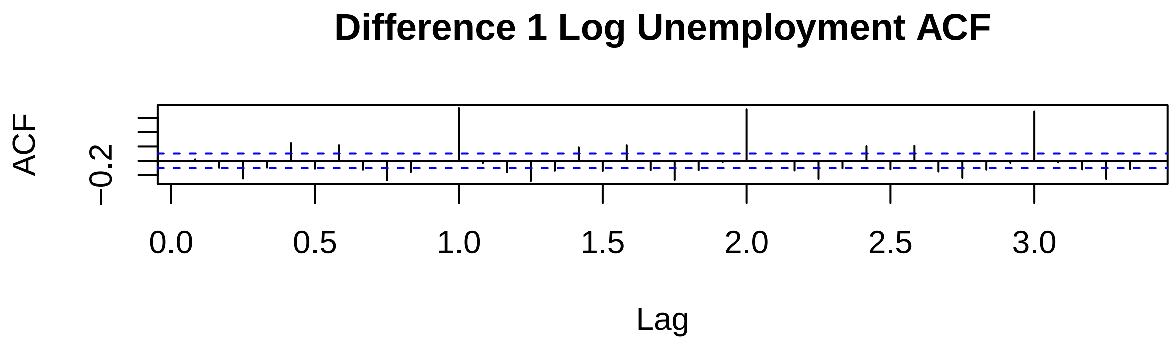
```r
par(mfrow=c(3, 1))
plot(log_unemp, main="Log Unemployment")
plot(d_log_unemp, main="Difference 1 Log Unemployment")
plot(d_d_log_unemp, main="Difference 2 Log Unemployment")
```

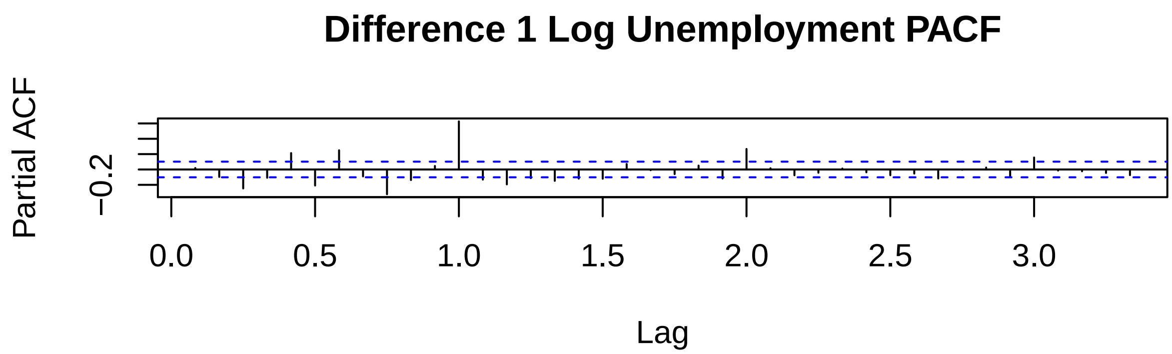



```r
par(mfrow=c(3, 1))
```

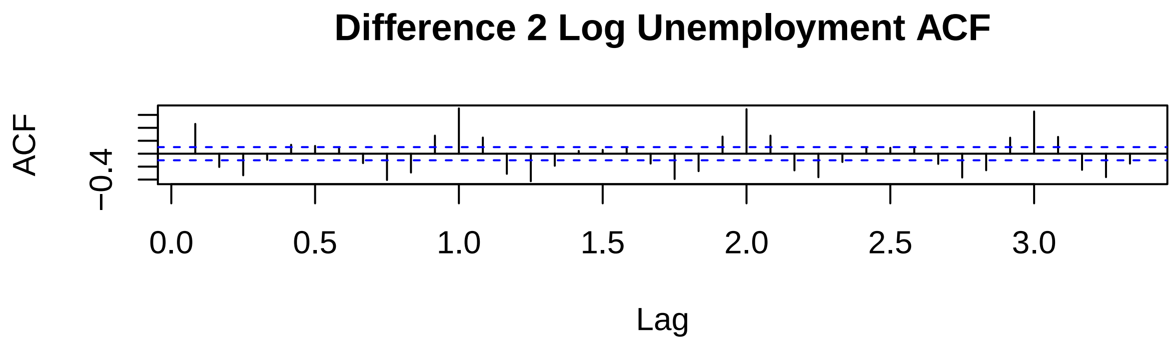
The data is not stationary visually and hence, differencing is done by the order of 1. Variance seems to be increasing with time which is reduced on transformation with the log scale. Differencing of the order 2 gives a smoother result.

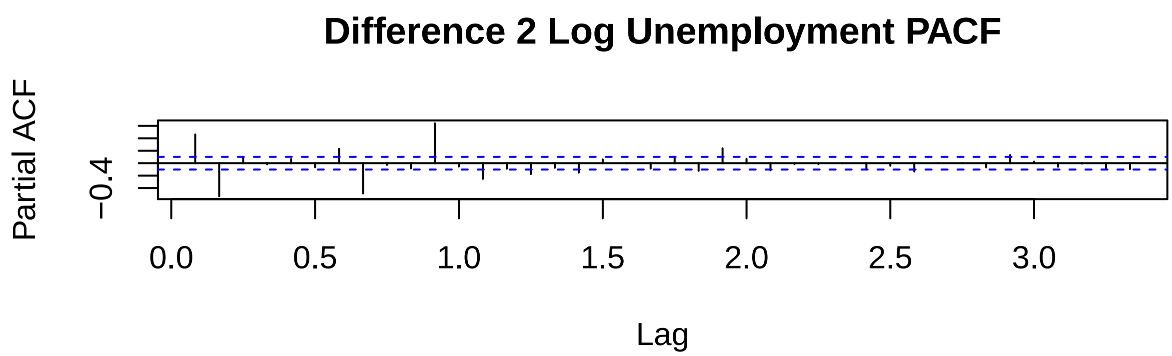
```r
par(mfrow=c(2, 1))
acf(d_log_unemp, lag.max=40, main="Difference 1 Log Unemployment ACF")
pacf(d_log_unemp, lag.max=40, main="Difference 1 Log Unemployment PACF")
```






```r
acf(d_d_log_unemp, lag.max=40, main="Difference 2 Log Unemployment ACF")
pacf(d_d_log_unemp, lag.max=40, main="Difference 2 Log Unemployment PACF")
```





Difference 1 **Seasonality behavior:** The ACF plot suggests seasonality at 12 lags that tails off both in the ACF and the PACF. This suggests an $ARMA_12$ seasonality component. The PACF spikes at 3 multiples which is indicative of AR(3). **Non-seasonality behavior:** There is no distinct non-seasonal pattern to be seen. Our model for this data is $\text{SARMA}(3,1,0)_{12}$.

Difference 2 **Seasonality behavior:** Large spikes at lags 9, 12, 15 are observed and then it is fades till the pattern repeats at lags 21, 24, 27. So there is definitely a seasonality pattern in the data. The PACF shows that the pattern tails off over time with spikes at two 2 multiples which indicates AR(2).

Non-seasonality behavor: Apart from the seasonal behavior the ACF tails off and the PACF cuts off after lag 2 indicating an AR(2) model.

```r
adf.test(d_log_oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -20.30    0.01
## [2,]   1 -16.56    0.01
## [3,]   2 -11.36    0.01
## [4,]   3 -10.77    0.01
## [5,]   4  -9.22    0.01
## [6,]   5  -9.19    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
```

```
## [1,]   0 -20.31    0.01
## [2,]   1 -16.58    0.01
## [3,]   2 -11.38    0.01
## [4,]   3 -10.79    0.01
## [5,]   4  -9.23    0.01
## [6,]   5  -9.21    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -20.29    0.01
## [2,]   1 -16.57    0.01
## [3,]   2 -11.37    0.01
## [4,]   3 -10.78    0.01
## [5,]   4  -9.22    0.01
## [6,]   5  -9.20    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
adf.test(d_d_log_oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -12.80    0.01
## [2,]   1 -15.34    0.01
## [3,]   2  -9.09    0.01
## [4,]   3 -10.93    0.01
## [5,]   4  -8.12    0.01
## [6,]   5  -9.56    0.01
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -12.81    0.01
## [2,]   1 -15.37    0.01
## [3,]   2  -9.11    0.01
## [4,]   3 -10.95    0.01
## [5,]   4  -8.14    0.01
## [6,]   5  -9.59    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -12.80    0.01
## [2,]   1 -15.36    0.01
## [3,]   2  -9.10    0.01
## [4,]   3 -10.94    0.01
## [5,]   4  -8.13    0.01
## [6,]   5  -9.58    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

The EACF suggests that there no ARMA model well suited to this data.

```r
fit1 <- Arima(log_unemp, order=c(0, 0, 0), seasonal=c(3, 1, 0))
fit1
```

```
## Series: log_unemp
## ARIMA(0,0,0)(3,1,0)[12]
##
## Coefficients:
##          sar1     sar2     sar3
##       -0.2589  -0.2801  -0.2872
## s.e.   0.0522   0.0526   0.0547
##
## sigma^2 estimated as 0.06287:  log likelihood=-13.62
## AIC=35.24   AICc=35.36   BIC=50.79
```

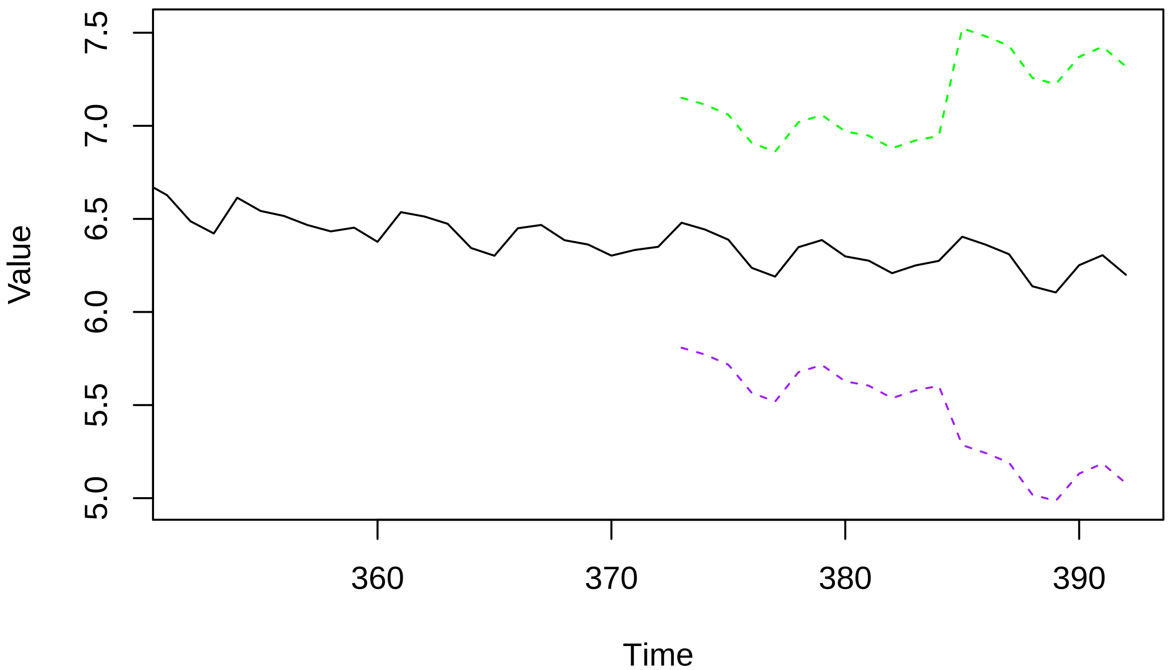
```r
fit2 <- Arima(log_unemp, order=c(0, 0, 0), seasonal=c(2, 2, 0))
fit2
```

```
## Series: log_unemp
## ARIMA(0,0,0)(2,2,0)[12]
##
## Coefficients:
##          sar1     sar2
##       -0.6655  -0.3370
## s.e.   0.0548   0.0553
##
## sigma^2 estimated as 0.1173:  log likelihood=-123.05
## AIC=252.1   AICc=252.17   BIC=263.66
```

The second fit has better BIC/AIC which is then our final model. It can be written formally as

$$(1 + 0.6655B + 0.3370B^2) \bigtriangledown^2 x_t = w_t.$$

```r
fit_plot(fit2)
```


Value
7.5
7.0
6.5
6.0
5.5
5.0
360
370
380
390
Time