HW2

Cohort 2, Group 7 - Hyeuk Jung, Jiaqi Li, Xichen Luo, Huanyu Liu

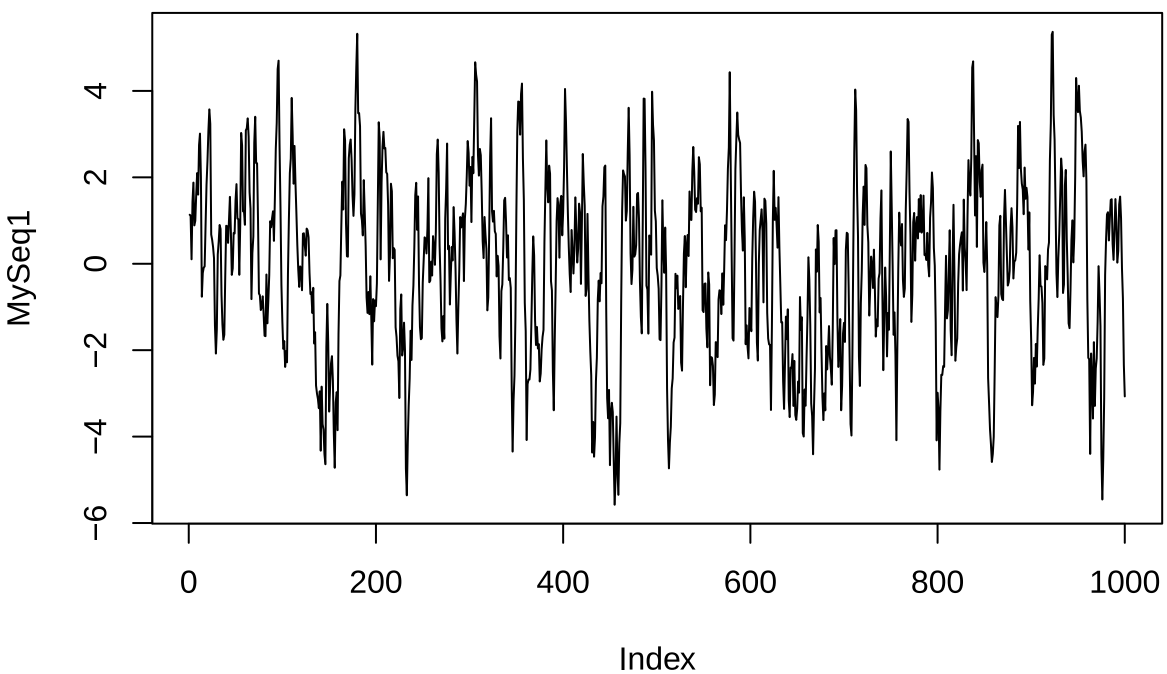
1/26/2019

Problem 1

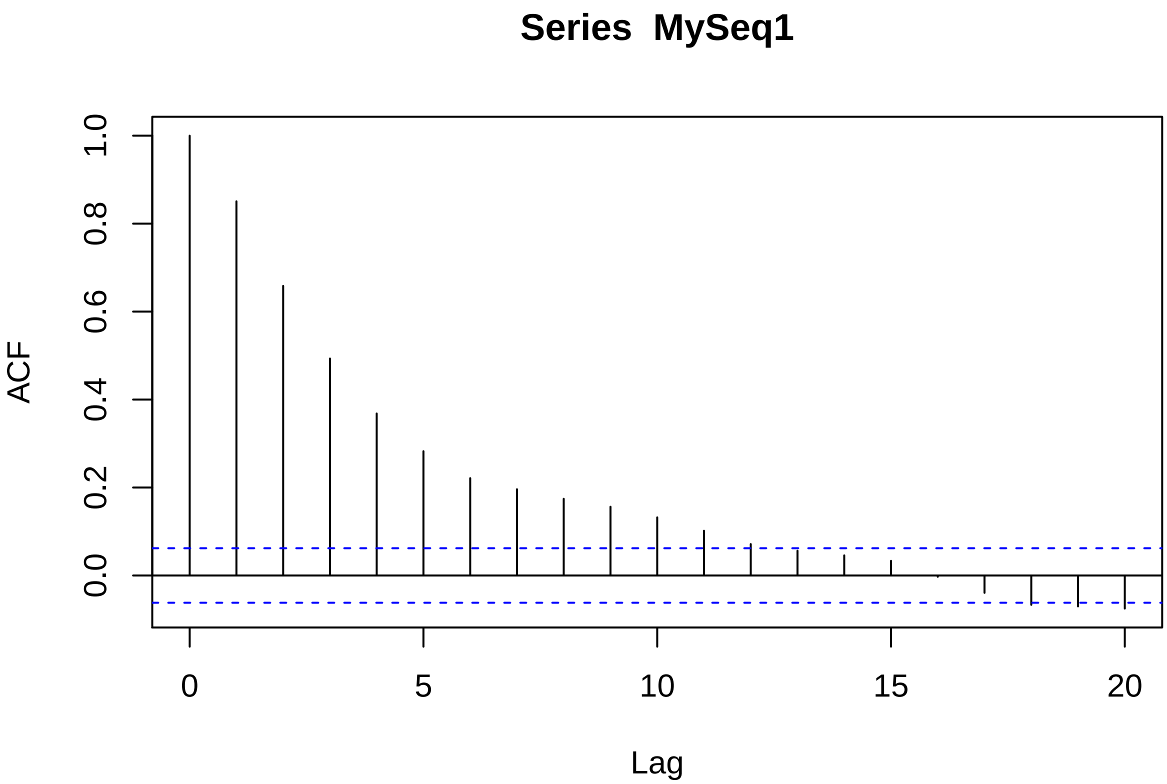
(a)

```r
set.seed(1)
b0 = 0
b1.1 = 1.1
b2.1 = -0.25
WN1 = rnorm(1000, mean = 0, sd = 1)
MySeq1 = rnorm(2)
for (i in 3:1000){
  MySeq1 = append(MySeq1, b0+b1.1*MySeq1[i-1]+b2.1*MySeq1[i-2]+WN1[i])
}

plot(MySeq1, type = "l")
```




```r
acf(MySeq1, lag.max = 20)
```



(b)

$$
\begin{aligned}
1 - \phi_1 x - \phi_2 x^2 &= 0 \\
1 - 1.1x + 0.25x^2 &= 0 \\
x_1 = \frac{1.1 + \sqrt{1.1^2 - 4 \times 0.25 \times 1}}{2 \times 0.25} &\approx 3.12 \\
\omega_1 = x_1^{-1} = 3.12^{-1} &= 0.3209 \\
x_2 = \frac{1.1 - \sqrt{1.1^2 - 4 \times 0.25 \times 1}}{2 \times 0.25} &\approx 1.28 \\
\omega_2 = x_2^{-1} &= 0.7791
\end{aligned}
$$

Both of the two different characteristic roots modulus of the above polynomial are less than 1, so this AR(2) process is stationary.

(c)

$$
\begin{aligned}
\mu &= \phi_0 = 2 \\
\text{Let } X_t &= r_t - \mu \\
X_t &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + \epsilon_t
\end{aligned}
$$

$$
\begin{aligned}
\mu &= \phi_0 = 2 \\
\text{Let } X_t &= r_t - \mu \\
X_t &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + \epsilon_t \\
X_{t+6} &= \phi_1 X_{t+5} + \phi_2 X_{t+4} + \epsilon_{t+6} \\
&= \phi_1(\phi_1 X_{t+4} + \phi_2 X_{t+3} + \epsilon_{t+5}) + \\
&\phi_2(\phi_1 X_{t+3} + \phi_2 X_{t+2} + \epsilon_{t+4}) + \epsilon_{t+6} \\
&= \phi_1^2 X_{t+4} + 2\phi_1\phi_2 X_{t+3} + \phi_2^2 X_{t+2} + \\
&\phi_1 \epsilon_{t+5} + \phi_2 \epsilon_{t+4} + \epsilon_{t+6} \\
&\ldots \\
&= \cdots + (\phi_2^3 + 6\phi_1^2\phi_2^2 + 5\phi_1^4\phi_2^1 + \phi_1^6)\epsilon_t \\
&\therefore \frac{\partial[r_{t+6} - \mu]}{\partial \epsilon_t} = \phi_2^3 + 6\phi_1^2\phi_2^2 + 5\phi_1^4\phi_2^1 + \phi_1^6 \\
&\approx 0.37956
\end{aligned}
$$

(d)

$$
\begin{aligned}
1 - 0.9x - 0.8x^2 &= 0 \\
x_1 = \frac{0.9 + \sqrt{0.9^2 + 4 \times 0.8 \times 1}}{2 \times -0.8} &\approx -1.814 \\
\omega_1 = x_1^{-1} = (-1.8141)^{-1} &= -0.5512 \\
x_2 = \frac{0.9 - \sqrt{0.9^2 + 4 \times 0.8 \times 1}}{2 \times -0.8} &\approx 0.689 \\
\omega_2 = x_2^{-1} &= 1.4512
\end{aligned}
$$

The modulus of one of the two different characteristic roots of the above polynomial is larger than 1, so this AR(2) process is not stationary.
For the dynamic multiplier, the calculation is similar as in (c).

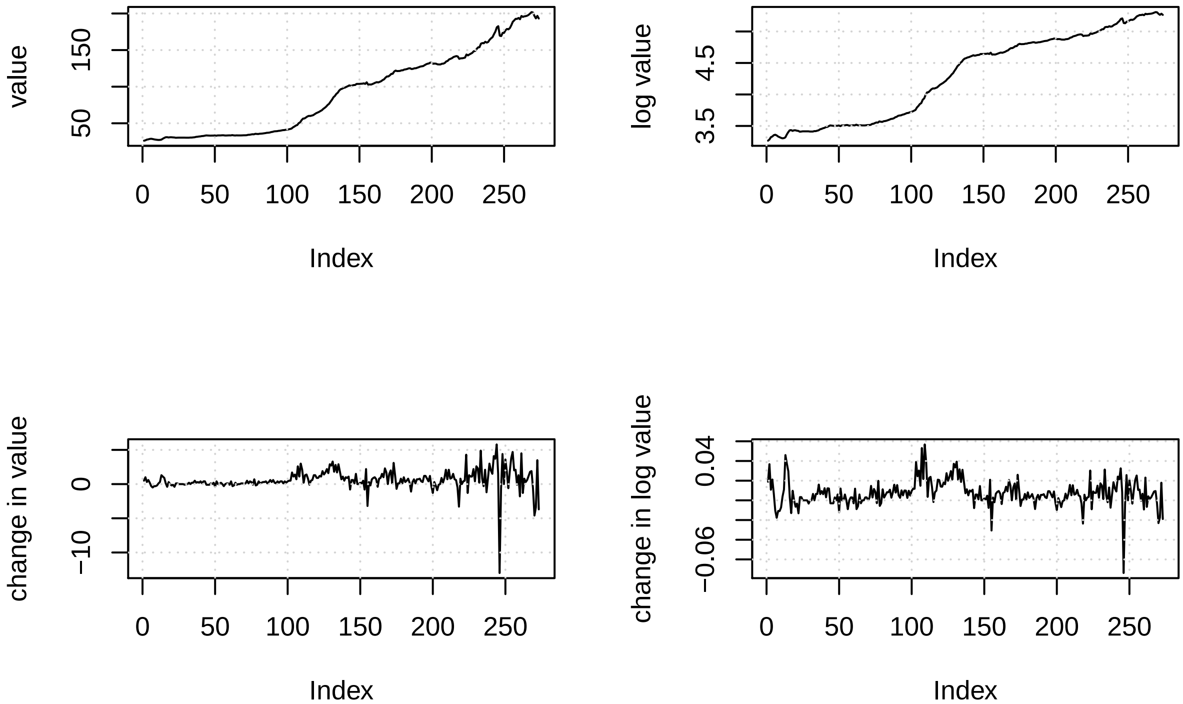
$$
\begin{aligned}
\frac{\partial[r_{t+6} - \mu]}{\partial \epsilon_t} &= \phi_2^3 + 6\phi_1^2\phi_2^2 + 5\phi_1^4\phi_2^1 + \phi_1^6 \\
&\approx 6.77824
\end{aligned}
$$

Problem 2

Q1.

```r
PPI = read.csv("PPIFGS.csv")
valPPI = PPI$VALUE
delPPI = diff(valPPI)
logPPI = log(valPPI)
dellogPPI = diff(log(valPPI))
par(mfrow = c(2,2))
plot(valPPI, type = "l", ylab = "value")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
plot(logPPI, type = "l", ylab = "log value")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
plot(delPPI, type = "l", ylab = "change in value")
```

```r
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
plot(dellogPPI, type = "l", ylab = "change in log value")
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = par("lwd"), equilogs = TRUE)
```



Q2.

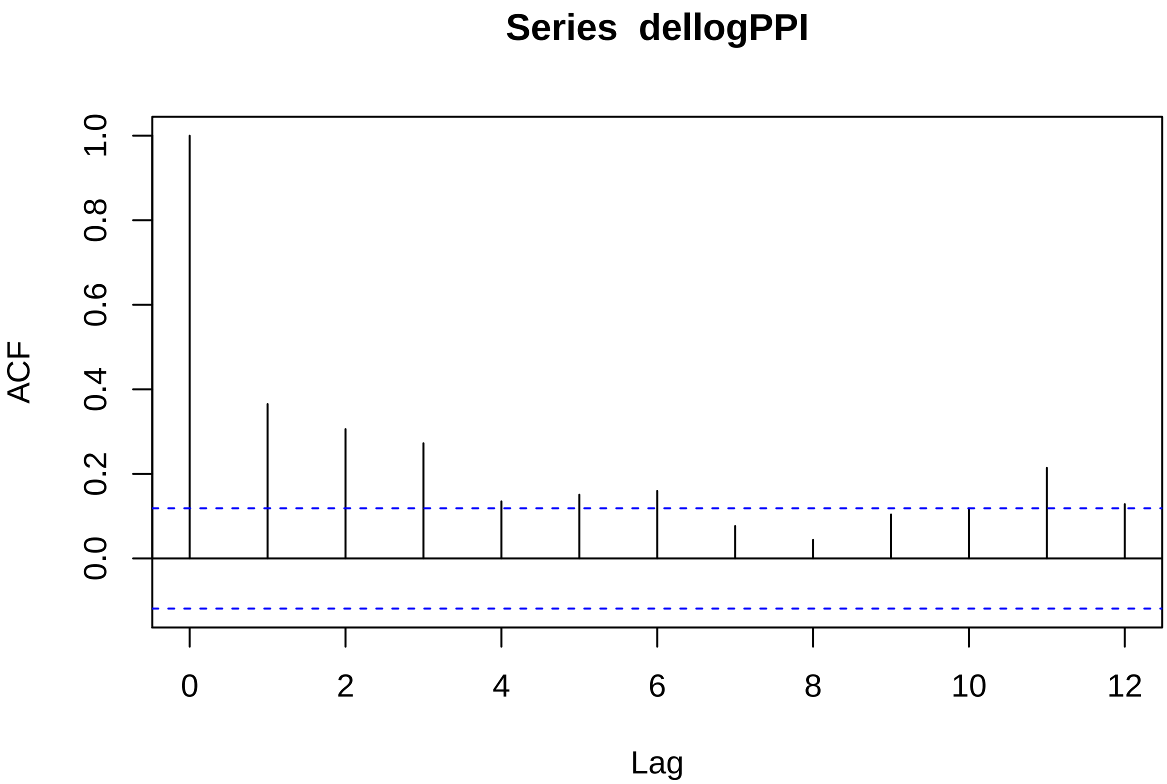
$\Delta logPPI$ looks covariance-stationary.

Among these four versions, the plot PPI in levels, and the plot Log PPI, show an uptrending. In that case, these two series are not covariance-stationary.

Compared the plot, PPI Difference, and the plot, Log PPI Difference, their fluctations don't show an uptrending or downtrending, while the volatility of the second plot, Log PPI Difference, shows more constant. In that case, the plot, Log PPI Difference, is covariance-stationary.

Q3.

```r
par(mfrow = c(1,1))
acf(dellogPPI, lag.max = 12)
```

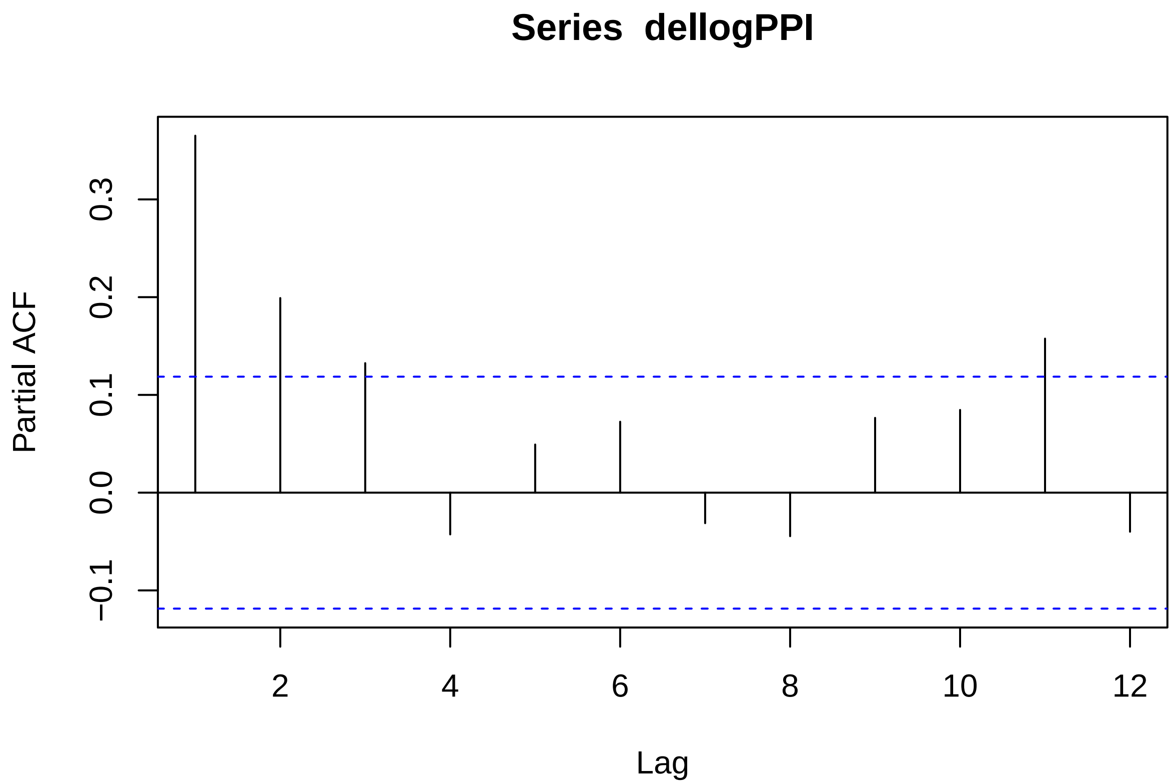


Since the plot of ACF converges very quickly to zero, it means that as time passing, in the long run, the lags could be less likely to have an effect on Log PPI Difference.

If the ACF converges very slowly, it means that a lot of short-term lags could affect the Log PPI Difference. In that case, the volatility of Log PPI Difference could be related to time instead of a constant.

Q4.

```
par(mfrow = c(1,1))
pacf(dellogPPI, lag.max = 12)
```



Since the Partial ACF shows significance of lag 1, 2, 3, and 11 compared to other lags. In that case, they should be included in the regression analysis. Therefore, we can consider AR(1), AR(2), AR(3), and AR(11) could better explain the changing of series y.

Q5.

(a)

```
n = length(dellogPPI)
yt = dellogPPI

yt.lag1 = yt[1:(n-1)]
Y1 = yt[2:n]

ar1 = lm(Y1~yt.lag1)

yt.lag1 = yt[1:(n-2)]
yt.lag2 = yt[2:(n-1)]
Y2 = yt[3:n]

ar2 = lm(Y2~yt.lag1+yt.lag2)

yt.lag1 = yt[1:(n-3)]
yt.lag2 = yt[2:(n-2)]
yt.lag3 = yt[3:(n-1)]
Y3 = yt[4:n]

ar3 = lm(Y3~yt.lag1+yt.lag2+yt.lag3)
```

```r
for (i in 1:11){
  assign(paste("yt.lag", i, sep = ""),
         yt[i:(n-(12-i))])
}
Y11 = yt[12:n]

ar11 = lm(Y11~yt.lag1+yt.lag2+yt.lag3+
            yt.lag4+yt.lag5+yt.lag6+
            yt.lag7+yt.lag8+yt.lag9+
            yt.lag10+yt.lag11)

ar1.coef = summary(ar1)
ar2.coef = summary(ar2)
ar3.coef = summary(ar3)
ar11.coef = summary(ar11)

1/polyroot(c(1,-ar1.coef$coefficients[2,1]))
```

```
## [1] 0.3707102+0i
```

```r
1/polyroot(c(1,-ar2.coef$coefficients[2,1], -ar2.coef$coefficients[3,1]))
```

```
## [1]  0.6472580+0i -0.4446352-0i
```

```r
1/polyroot(c(1,-ar3.coef$coefficients[2,1],
           -ar3.coef$coefficients[3,1],
           -ar3.coef$coefficients[4,1]))
```

```
## [1]  0.7805586+0.0000000i -0.3207432-0.4890206i -0.3207432+0.4890206i
```

```r
1/polyroot(c(11,-ar11.coef$coefficients[2,1],
           -ar11.coef$coefficients[3,1],
           -ar11.coef$coefficients[4,1],
           -ar11.coef$coefficients[5,1],
           -ar11.coef$coefficients[6,1],
           -ar11.coef$coefficients[7,1],
           -ar11.coef$coefficients[8,1],
           -ar11.coef$coefficients[9,1],
           -ar11.coef$coefficients[10,1],
           -ar11.coef$coefficients[11,1],
           -ar11.coef$coefficients[12,1]))
```

```
##  [1]  0.6037606-0.4191605i -0.6622935-0.2021894i -0.4375012+0.5084891i
##  [4]  0.6037606+0.4191605i  0.2706054-0.6636678i -0.4375012-0.5084891i
##  [7] -0.1399119+0.6679610i  0.7460541-0.0000000i -0.1399119-0.6679610i
## [10] -0.6622935+0.2021894i  0.2706054+0.6636678i
```

Stationary test for AR(3): Using the polyroot function, we got three characteristic roots: 0.7805586, -0.3207432-0.4890206i, and -0.3207432+0.4890206i. For the imaginary results, we checked if they are less than one in modulus.

$$
\begin{aligned}
\omega_1 &= 0.7805586 \\
\omega_2 &= \sqrt{(-0.3207432)^2 + (-0.4890206)^2} = \sqrt{0.3420} \approx 0.5848 \\
\omega_3 &= \sqrt{(-0.3207432)^2 + (0.4890206)^2} = \sqrt{0.3420} \approx 0.5848
\end{aligned}
$$

As all characteristic roots are less than one, we can conclude that the model is stationary.

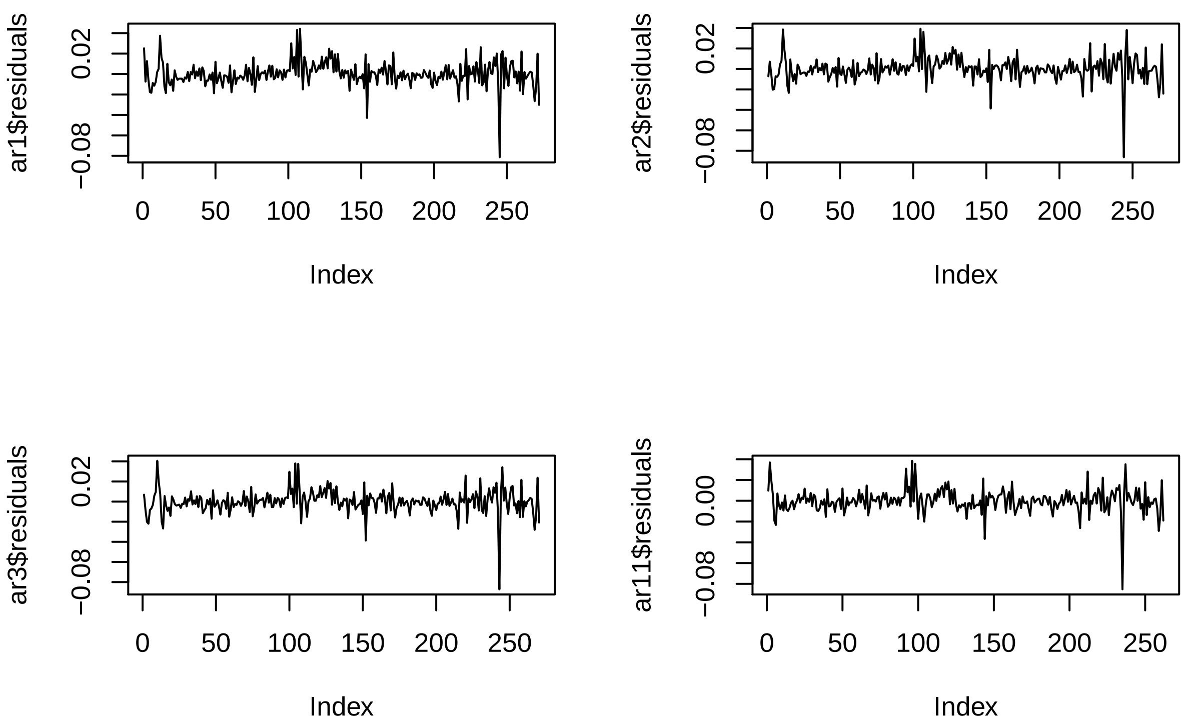
Stationary test for AR(11):

$$
\begin{aligned}
\omega_1 &= 0.7460541 \\
\omega_2 and \omega_3 &= \sqrt{(0.6037606)^2 + (\pm 0.4191605)^2} \approx 0.7350 \\
\omega_4 and \omega_5 &= \sqrt{(-0.6622935)^2 + (\pm 0.2021894)^2} \approx 0.6925 \\
\omega_6 and \omega_7 &= \sqrt{(-0.4375012)^2 + (\pm 0.5084891)^2} \approx 0.6708 \\
\omega_8 and \omega_9 &= \sqrt{(0.2706054)^2 + (\pm 0.6636678)^2} \approx 0.7167 \\
\omega_{10} and \omega_{11} &= \sqrt{(-0.1399119)^2 + (\pm 0.6679610)^2} \approx 0.6825
\end{aligned}
$$

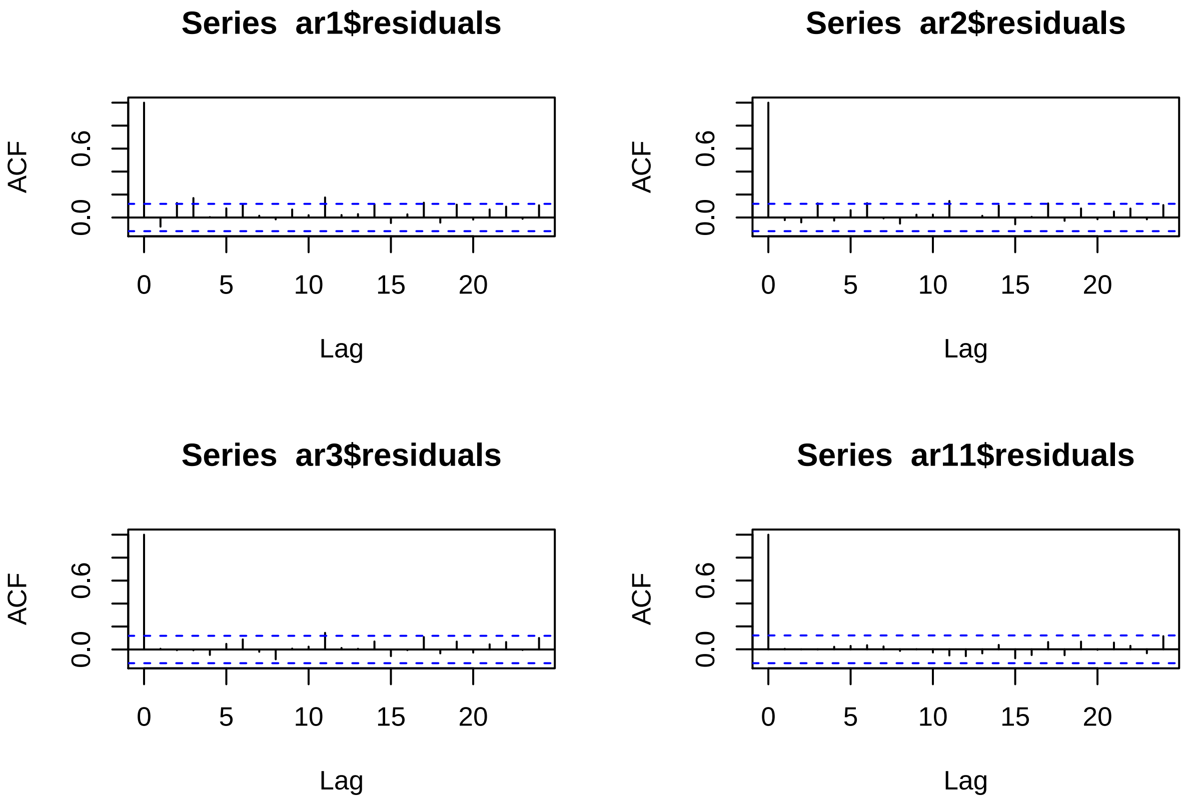
As all characteristic roots are less than one, we can conclude that the model is stationary.

(b)

```
par(mfrow = c(2,2))
plot(ar1$residuals, type = "l")
plot(ar2$residuals, type = "l")
plot(ar3$residuals, type = "l")
plot(ar11$residuals, type = "l")
```



```r
par(mfrow = c(2,2))
acf(ar1$residuals)
acf(ar2$residuals)
acf(ar3$residuals)
acf(ar11$residuals)
```



(c)

```r
Box.test(ar1$residuals, lag = 8, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar1$residuals
## X-squared = 19.989, df = 8, p-value = 0.01038
```

```r
Box.test(ar1$residuals, lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar1$residuals
## X-squared = 30.374, df = 12, p-value = 0.002452
```

```r
Box.test(ar2$residuals, lag = 8, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar2$residuals
## X-squared = 11.345, df = 8, p-value = 0.1829
```

```r
Box.test(ar2$residuals, lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar2$residuals
## X-squared = 17.582, df = 12, p-value = 0.129
```

```r
Box.test(ar3$residuals, lag = 8, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar3$residuals
## X-squared = 5.6426, df = 8, p-value = 0.6872
```

```r
Box.test(ar3$residuals, lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar3$residuals
## X-squared = 11.845, df = 12, p-value = 0.4582
```

```r
Box.test(ar11$residuals, lag = 8, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar11$residuals
## X-squared = 0.95008, df = 8, p-value = 0.9985
```

```r
Box.test(ar11$residuals, lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar11$residuals
## X-squared = 2.9547, df = 12, p-value = 0.9959
```

```r
aic = c(AIC(ar1),AIC(ar2),AIC(ar3),AIC(ar11))
bic = c(BIC(ar1),BIC(ar2),BIC(ar3),BIC(ar11))
```

6.

```r
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
PPI = read.csv("PPIFGS.csv")
date = as.Date(as.vector(PPI$DATE))
PPI.xts = xts(PPI$VALUE, order.by = date)
PPI.sample = diff(log(as.vector(PPI.xts["/2005-10-01"])))
PPI.pred = diff(log(as.vector(PPI.xts["2005-10-01/"])))

ar3.test = ar(PPI.sample,order.max = 3, aic = F)
ar11.test = ar(PPI.sample,order.max = 11, aic = F)

n.sample = length(PPI.sample)
n.pred = length(PPI.pred)

pred3 = predict(ar3.test, n.ahead = 39)$pred
pred11 = predict(ar11.test, n.ahead = 39)$pred


v3 = sum((PPI.pred-pred3)^2)/n.pred
v11 = sum((PPI.pred-pred11)^2)/n.pred
c(v3,v11)
```

```
## [1] 0.0003354779 0.0003378032
```

```r
#rw---------------------
rw.sd = sqrt(var(PPI.sample))
monte = vector()
for (i in 1:1000)
{
  set.seed(i)
  rw.test = PPI.sample[n.sample]
  for (i in 1:n.pred){
    rw.test = append(rw.test, rw.test[i]+rnorm(1,0,rw.sd))
  }
  rw.v = sum((PPI.pred-rw.test[-1])^2)/n.pred
  monte = append(monte,rw.v)
}
mean(monte)
```

```
## [1] 0.003251679
```