應用財務計量經濟學- HW2

魏上傑

R11323008

目錄

1 Exercise 5.2

Consider AR model:

$$Y_t = a_0 + a_2 Y_{t-2} + \epsilon_t \tag{1}$$

with $|a_2| < 1, \epsilon_t \sim i.i.d(0, \sigma^2)$

令 Ω_t 為第 t 期的資訊集合,i.e., $\Omega_t = \{Y_t, Y_{t-1} ...\}$

假設 $E_t(\epsilon_{t+1}) = E(\epsilon_{t+1}|\Omega_t) = E(\epsilon_{t+1}|X_{1t}, X_{2t}, ..., X_{kt}, X_{1t-1}, X_{2t-1}, ...) = 0$

(a) $E_{t-2}(Y_t)$

$$\begin{aligned}
E_{t-2}(Y_t) &= E(Y_t|\Omega_{t-2}) &(2)\\
&= E(Y_t|Y_{t-2}, Y_{t-3}, ...) &(3)\\
&= E(a_0 + a_2 Y_{t-2} + \epsilon_t|Y_{t-2}, Y_{t-3}, ...) &(4)\\
&= a_0 + a_2 Y_{t-2} + E(\epsilon_t|Y_{t-2}, Y_{t-3}, ...) &(5)
\end{aligned}$$

(b) $E_{t-1}(Y_t)$

$$\begin{aligned}
E_{t-1}(Y_t) &= E(Y_t|\Omega_{t-1}) && (6)\\
&= E(Y_t|Y_{t-1}, Y_{t-2}, ...) && (7)\\
&= E(a_0 + a_2 Y_{t-2} + \epsilon_t | Y_{t-1}, Y_{t-2}, ...) && (8)\\
&= a_0 + a_2 Y_{t-2} + E(\epsilon_t | Y_{t-1}, Y_{t-2}, ...) && (9)\\
&= a_0 + a_2 Y_{t-2} && (10)
\end{aligned}$$

$E_{t-1}(Y_t)$ 與 $E_{t-2}(Y_t)$ 的差異在於是否用了所有可用的歷史資料，$E_{t-2}(Y_t)$ 並未使用 t-1 期的歷史資料，造成條件期望值的差異。

(c) $E_t(Y_{t+2})$

$$\begin{aligned}
E_t(Y_{t+2}) &= E(Y_{t+2}|\Omega_t) && (11)\\
&= E(Y_{t+2}|Y_t, Y_{t-1}, ...) && (12)\\
&= E(a_0 + a_2 Y_t + \epsilon_{t+2} | Y_t, Y_{t-1}, ...) && (13)\\
&= a_0 + a_2 Y_t + E(\epsilon_{t+2} | Y_t, Y_{t-1}, ...) && (14)\\
&= a_0 + a_2 Y_t + E(\epsilon_{t+2} | Y_t, Y_{t-1}, ...) && (15)
\end{aligned}$$

(d) $Cov(Y_t, Y_{t-1})$

$$Cov(Y_t, Y_{t-1}) = Cov(a_0 + a_2 Y_{t-2} + \epsilon_t, a_0 + a_2 Y_{t-3} + \epsilon_{t-1}) = 0 \quad (16)$$

(e) $Cov(Y_t, Y_{t-2})$

$$Cov(Y_t, Y_{t-2}) = Cov(a_0 + a_2 Y_{t-2} + \epsilon_t, a_0 + a_2 Y_{t-4} + \epsilon_{t-2}) = 0 \quad (17)$$

(f) $\frac{\partial Y_{t+j}}{\partial \epsilon_t}, j = 0, 1..., 6$

$$\begin{aligned}
Y_t &= a_0 + a_2 Y_{t-2} + \epsilon_t && (18)\\
&= a_0 + a_2[a_0 + a_2 Y_{t-4} + \epsilon_{t-2}] + \epsilon_t && (19)\\
&= a_0 + a_2 a_0 + a_2^2[a_0 + a_2 Y_{t-6} + \epsilon_{t-4}] + a_2\epsilon_{t-2} + \epsilon_t && (20)\\
&= a_0 + a_2 a_0 + a_2^2 a_0 + a_2^3[a_0 + a_2 Y_{t-8} + \epsilon_{t-6}] + a_2^2\epsilon_{t-4} + a_2\epsilon_{t-2} + \epsilon_t && (21)
\end{aligned}$$

$$\frac{\partial Y_{t+j}}{\partial \epsilon_t} = \begin{cases} \frac{\partial Y_t}{\partial \epsilon_t} = 1 & \text{, if } j = 0 \\ \frac{\partial Y_{t+1}}{\partial \epsilon_t} = 0 & \text{, if } j = 1 \\ \frac{\partial Y_{t+2}}{\partial \epsilon_t} = a_2 & \text{, if } j = 2 \\ \frac{\partial Y_{t+3}}{\partial \epsilon_t} = 0 & \text{, if } j = 3 \\ \frac{\partial Y_{t+4}}{\partial \epsilon_t} = a_2^2 & \text{, if } j = 4 \\ \frac{\partial Y_{t+5}}{\partial \epsilon_t} = 0 & \text{, if } j = 5 \\ \frac{\partial Y_{t+6}}{\partial \epsilon_t} = a_2^3 & \text{, if } j = 6 \end{cases}$$

2 Exercise 5.4

在主計處下載台灣以固定價格計算的民間消費資料 (1961Q1-2006Q4)，命名為 `ctw`。將 `ctw` 取對數後命名為 `lctw`。將 `lctw` 取一階差分後，命名為 `dlctw`

(a) 設定 1961Q1-2003Q4 為樣本內期間，2004Q1-2006Q4 為樣本外期間。

```
# 資料取自國民消費季資料
# 民間消費季資料只從 1981Q1 開始

library(readxl)
consumption <- read_excel("consume.xlsx", skip = 4)
```

```
## New names:
## * `` -> `...1`
```

```
colnames(consumption) <- c("Date", "ctw")
consumption$ctw <- consumption$ctw*(10^6) # 轉為單位新台幣

consumption$lctw <- log(consumption$ctw)
dlctw <- diff(consumption$lctw, lag=1, differences=1)
trn.df <- data.frame(dlctw[1:171]) # in-sample
tst.df <- data.frame(dlctw[-(1:171)]) # out-of-sample

colnames(trn.df) <- c("dlctw")
colnames(tst.df) <- c("dlctw")
```

(b) 以 AR(1) 模型估計 `dlctw`，執行樣本外預測，並計算 $\widehat{\text{RMSE}}$, 命名為 RMSE_1

```
regr <- ar.ols(trn.df$dlctw, order=1, demean = FALSE, intercept = TRUE)
regr
```

```
##
```

```
## Call:
## ar.ols(x = trn.df$dlctw, order.max = 1, demean = FALSE, intercept = TRUE)
##
## Coefficients:
##       1
## -0.322
##
## Intercept: 0.02484 (0.003817)
##
## Order selected 1  sigma^2 estimated as  0.002176
```

```
regr$asy.se.coef # std.errors of the estimates
```

```
## $x.mean
## [1] 0.003817428
##
## $ar
## [1] 0.07196572
```

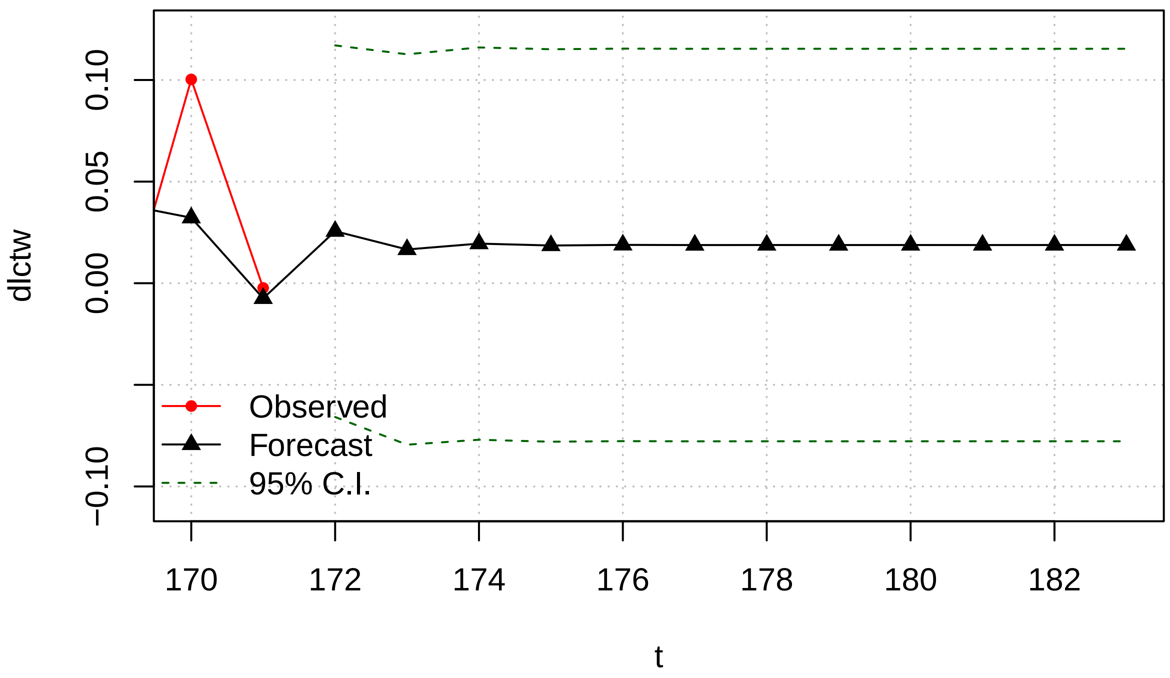
Let $v_t =$ `dlctw`, then the estimated AR(1) model is

$$v_t = \phi_0 + \phi_1 v_{t-1} + \epsilon_t \tag{22}$$
$$\hat{v}_t = \hat{\phi}_0 + \hat{\phi}_1 v_{t-1} \tag{23}$$
$$\hat{v}_t = 0.02484_{(0.003817)} - 0.322_{(0.071967)} v_{t-1} \tag{24}$$

with $\hat{\sigma}^2_\epsilon = 0.002176$

```
fore <- predict(regr, n.ahead=12)

low <- fore$pred - qnorm(p = 0.975, mean = 0, sd =
         1)*fore$se

up <- fore$pred + qnorm(p = 0.975, mean = 0, sd =
         1)*fore$se

plot(x = trn.df$dlctw, ylab = expression(dlctw), xlab = "t", type =
       "o", col = "red", lwd = 1, pch = 20, main = NULL,
    panel.first = grid(col = "gray", lty = "dotted"), xlim
    = c(170, 183))
```

```
lines(x = c(trn.df$dlctw - regr$resid, fore$pred), lwd
    = 1, col = "black", type = "o", pch = 17)

lines(y = low, x = 172:183, lwd = 1, col = "darkgreen",
    lty = "dashed")

lines(y = up, x = 172:183, lwd = 1, col = "darkgreen",
    lty = "dashed")

legend("bottomleft", legend = c("Observed", "Forecast",
    "95% C.I."), lty = c("solid", "solid", "dashed"), col
    = c("red", "black", "darkgreen"), pch = c(20, 17,
    NA), bty = "n")
```




```
# calculate RMSE
error1 <- (tst.df$dlctw-fore$pred)^2
MSE1 <- mean(error1)
RMSE1 <- sqrt(MSE1)
RMSE1
```

```
## [1] 0.02718167
```

```r
# one can also use arima() to fit AR model
# the estimated result is very close to ar.ols
mod.fit <- arima(x=trn.df$dlctw, order=c(1, 0, 0))
mod.fit
```

```
##
## Call:
## arima(x = trn.df$dlctw, order = c(1, 0, 0))
##
## Coefficients:
##           ar1  intercept
##       -0.3256     0.0185
## s.e.   0.0727     0.0027
##
## sigma^2 estimated as 0.0022:  log likelihood = 280.5,  aic = -555
```

```r
fore.mod <- predict(mod.fit, n.ahead=12, se.fit=TRUE)
fore.mod$pred
```

```
## Time Series:
## Start = 172
## End = 183
## Frequency = 1
##  [1] 0.02533243 0.01632782 0.01925965 0.01830507 0.01861587 0.01851468
##  [7] 0.01854763 0.01853690 0.01854039 0.01853926 0.01853963 0.01853950
```

```r
# the intercept in arima() result is actually the estimated mean of the series
# hence, need to convert to intercept
0.0185*(1-(-0.3256))
```

```
## [1] 0.0245236
```

(c) 以 AR(3) 模型估計 `dlctw`，執行樣本外預測，並計算 $\widehat{\text{RMSE}}$, 命名為 RMSE_3

```r
regr <- ar.ols(trn.df$dlctw, order=3, demean = FALSE, intercept = TRUE)
regr
```

```
##
## Call:
## ar.ols(x = trn.df$dlctw, order.max = 3, demean = FALSE, intercept = TRUE)
##
## Coefficients:
##       1        2        3
## -0.7771  -0.7491  -0.7531
```

```
##
## Intercept: 0.0613 (0.003084)
##
## Order selected 3  sigma^2 estimated as  0.0007581
```

```
regr$asy.se.coef # std.errors of the estimates
```

```
## $x.mean
## [1] 0.003083553
##
## $ar
## [1] 0.05008405 0.05127500 0.04968987
```

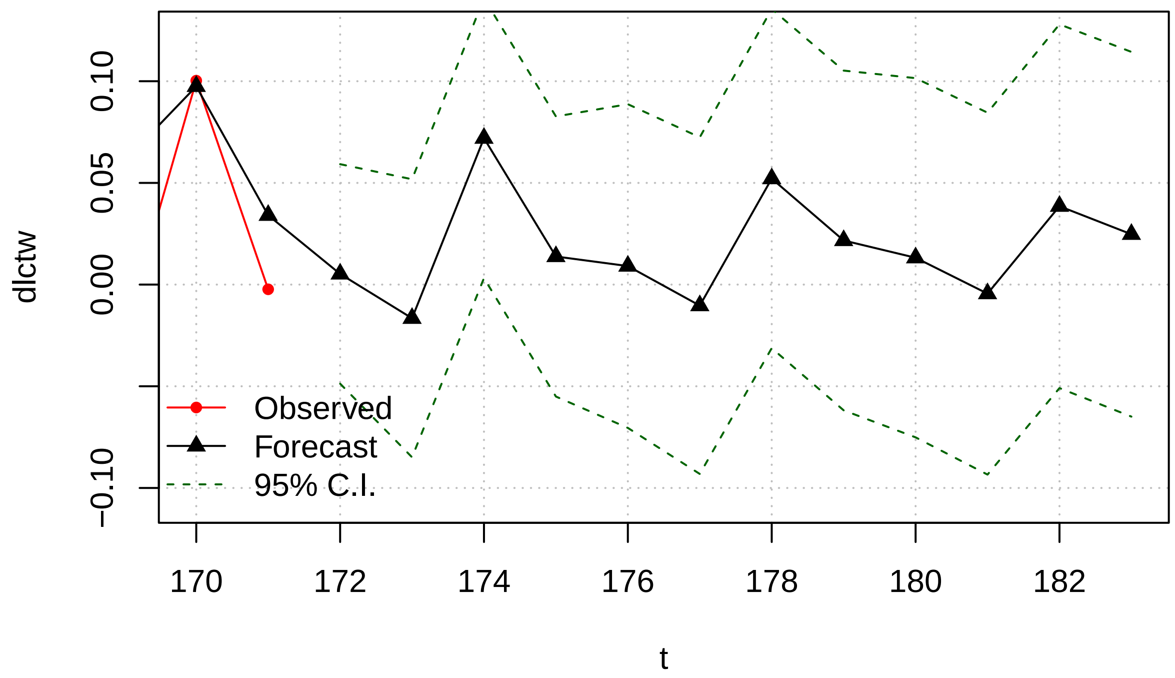
Let $v_t =$ `dlctw`, then the estimated AR(3) model is

$$v_t = \phi_0 + \phi_1 v_{t-1} + \phi_2 v_{t-2} + \phi_3 v_{t-3} + \epsilon_t \tag{25}$$
$$\hat{v}_t = \hat{\phi}_0 + \hat{\phi}_1 v_{t-1} + \hat{\phi}_2 v_{t-2} + \hat{\phi}_3 v_{t-3} \tag{26}$$
$$\hat{v}_t = 0.0613_{(0.003)} - 0.7771_{(0.0500)} v_{t-1} - 0.7491_{(0.0513)} - 0.7531_{(0.0497)} \tag{27}$$

with $\hat{\sigma}^2_\epsilon = 0.0007581$

```
fore <- predict(regr, n.ahead=12)

low <- fore$pred - qnorm(p = 0.975, mean = 0, sd =
        1)*fore$se

up <- fore$pred + qnorm(p = 0.975, mean = 0, sd =
        1)*fore$se

plot(x = trn.df$dlctw, ylab = expression(dlctw), xlab = "t", type =
    "o", col = "red", lwd = 1, pch = 20, main = NULL,
    panel.first = grid(col = "gray", lty = "dotted"), xlim
    = c(170, 183))

lines(x = c(trn.df$dlctw - regr$resid, fore$pred), lwd
    = 1, col = "black", type = "o", pch = 17)

lines(y = low, x = 172:183, lwd = 1, col = "darkgreen",
    lty = "dashed")
lines(y = up, x = 172:183, lwd = 1, col = "darkgreen",
    lty = "dashed")
```

```
legend("bottomleft", legend = c("Observed", "Forecast",
    "95% C.I."), lty = c("solid", "solid", "dashed"), col
    = c("red", "black", "darkgreen"), pch = c(20, 17,
    NA), bty = "n")
```




```
# calculate RMSE
error3 <- (tst.df$dlctw-fore$pred)^2
MSE3 <- mean(error3)
RMSE3 <- sqrt(MSE3)
RMSE3
```

```
## [1] 0.02255923
```

(d) 比較 RMSE_1 與 RMSE_3，哪個模型樣本外預測較佳

```
RMSE1
```

```
## [1] 0.02718167
```

```
RMSE3
```

```
## [1] 0.02255923
```

```
RMSE3<RMSE1
```

```
## [1] TRUE
```

AR(3) model has a better out-of-sample forecast performance.

(e) 計算 AR(1) 與 AR(3) 模型之間的 DM 統計量。

$$H_0 : \text{MSE}^{\text{AR(1)}} = \text{MSE}^{\text{AR(3)}}$$
$$H_1 : \text{MSE}^{\text{AR(3)}} < \text{MSE}^{\text{AR(1)}}$$

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```
d <- error1-error3
N <- length(d)
m <- floor(0.75 * N^(1/3))
d_df <- data.frame(d)
regr_DM <- lm(d~1, data = d_df)
coeftest(regr_DM, vcov=NeweyWest(regr_DM, prewhite = F, adjust=T, lag=m-1))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.00022992 0.00014790  1.5546   0.1483
```

The DM statistics is $\frac{0.00022992}{0.00014790} = 1.5546$

3 Exercise 6.4

Consider GDP growth ΔY_t as an AR(1) model:

$$\Delta Y_t = c + \phi \Delta Y_{t-1} + \epsilon_t \tag{28}$$

where $Y_t = log(GDP_t)$ and $\epsilon_t \sim WN(0, \sigma^2)$

(a) Y_t 為何種自我回歸序列

$$Y_t - Y_{t-1} = c + \phi(Y_{t-1} - Y_{t-2}) + \epsilon_t \tag{29}$$
$$\implies Y_t = c + (1+\phi)Y_{t-1} - \phi Y_{t-2} + \epsilon_t \tag{30}$$

Y_t 為 AR(2) 序列

(b) Y_t 是否定態

否。

We have $[1-(1+\phi)L + \phi L^2]Y_t = c + \epsilon_t$

i.e., $\beta(z) = 1 - (1+\phi)z + \phi z^2$

Notice that $\beta(1) = 0$ i.e., Y_t has unit root.

(c) $\frac{\partial \Delta Y_{t+j}}{\partial \epsilon_t}$

Let $v_t = \Delta Y_t$, then $v_t = c + \phi v_{t-1} + \epsilon_t$

$$v_t = c + \phi v_{t-1} + \epsilon_t \tag{31}$$
$$= c + \phi[c + \phi v_{t-2} + \epsilon_{t-1}] + \epsilon_t \tag{32}$$
$$= ... \tag{33}$$
$$\implies \frac{\partial v_{t+j}}{\partial \epsilon_t} = \frac{\partial \Delta Y_{t+j}}{\partial \epsilon_t} = \phi^j \tag{34}$$

(d) $\lim_{j\to\infty} \frac{\partial E_t \Delta Y_{t+j}}{\partial \epsilon_t}$

$$E_t(v_{t+j}) = E(v_{t+j}|\Omega_t) = E(v_{t+j}|v_t, v_{t-1}, ...) \tag{35}$$
$$= E(c + \phi v_{t+j-1} + \epsilon_{t+j}|v_t, v_{t-1}, ...) \tag{36}$$
$$= c + \phi E(v_{t+j-1}|v_t, v_{t-1}, ...) + E(\epsilon_{t+j}|v_t, v_{t-1}, ...) \tag{37}$$
$$= ...$$

In particular,

$$E_t(v_{t+1}) = E(v_{t+1}|\Omega_t) = E(v_{t+1}|v_t, v_{t-1}, ...) \quad (38)$$
$$= E(c + \phi v_t + \epsilon_{t+1}|v_t, v_{t-1}, ...) \quad (39)$$
$$= c + \phi E(v_t|v_t, v_{t-1}, ...) + E(\epsilon_{t+1}|v_t, v_{t-1}, ...) \quad (40)$$
$$= c + \phi v_t + E(\epsilon_{t+1}|v_t, v_{t-1}, ...) \quad (41)$$
$$= c + \phi(c + \phi v_{t-1} + \epsilon_t) + E(\epsilon_{t+1}|v_t, v_{t-1}, ...) \quad (42)$$
$$\implies \frac{\partial E_t v_{t+1}}{\partial \epsilon_t} = \phi \quad (43)$$

Assume that $\phi > 0$

$$\lim_{j\to\infty} \frac{\partial E_t \Delta Y_{t+j}}{\partial \epsilon_t} = \lim_{j\to\infty} \phi^j = \begin{cases} 0 & \text{if } |\phi| < 1 \\ \infty & \text{if } |\phi| \geq 1 \end{cases}$$

(e) $\frac{\partial Y_{t+j}}{\partial \epsilon_t}$

Let $\beta_1 = (1 + \phi), \beta_2 = -\phi$

Then $Y_t = c + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \epsilon_t$

$$\frac{\partial Y_t}{\partial \epsilon_t} = 1 \quad (44)$$
$$\frac{\partial Y_t}{\partial \epsilon_{t-1}} = \beta_1 \quad (45)$$
$$\frac{\partial Y_t}{\partial \epsilon_{t-2}} = \beta_1 \frac{\partial Y_t}{\partial \epsilon_{t-1}} + \beta_2 \frac{\partial Y_t}{\partial \epsilon_t} = \beta_1^2 + \beta_2 \quad (46)$$
$$\frac{\partial Y_t}{\partial \epsilon_{t-3}} = \beta_1 \frac{\partial Y_t}{\partial \epsilon_{t-2}} + \beta_2 \frac{\partial Y_t}{\partial \epsilon_{t-1}} = \beta_1(\beta_1^2 + \beta_2) + \beta_2(\beta_1) = \beta_1^3 + 2\beta_1\beta_2 \quad (47)$$
$$\vdots$$
$$\frac{\partial Y_t}{\partial \epsilon_{t-j}} = \beta_1 \frac{\partial Y_t}{\partial \epsilon_{t-j+1}} + \beta_2 \frac{\partial Y_t}{\partial \epsilon_{t-j+2}} \quad (48)$$
$$\implies \frac{\partial Y_t}{\partial \epsilon_{t-j}} = \frac{\partial Y_{t+j}}{\partial \epsilon_t} = \beta_1 \frac{\partial Y_{t+j-1}}{\partial \epsilon_t} + \beta_2 \frac{\partial Y_{t+j-2}}{\partial \epsilon_t} \quad (49)$$

Write in first-order form:

$$\begin{bmatrix} Y_t \\ Y_{t-1} \\ Y_{t-2} \end{bmatrix} = \begin{bmatrix} c & \beta_1 & \beta_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ Y_{t-1} \\ Y_{t-2} \end{bmatrix} + \begin{bmatrix} \epsilon_t \\ 0 \\ 0 \end{bmatrix} \quad (50)$$

Let $\mathbf{Y}_t = (Y_t, Y_{t-1}, Y_{t-2})', \varepsilon_t = (\epsilon_t, 0, 0)'$ and Φ be the corresponding 3x3 matrix, then we have $\mathbf{Y}_t = \Phi\mathbf{Y}_{t-1} + \varepsilon_t$

Notice by the derivation in textbook, we get

$$\frac{\partial Y_{t+j}}{\partial \epsilon_t} = \begin{bmatrix} 1 & 0 & ... & 0 \end{bmatrix} \Phi^j \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{51}$$

(f) $\lim_{j\to\infty} \frac{\partial E_t Y_{t+j}}{\partial \epsilon_t}$

$$\lim_{j\to\infty} \frac{\partial E_t Y_{t+j}}{\partial \epsilon_t} = \begin{bmatrix} 1 & 0 & ... & 0 \end{bmatrix} \lim_{j\to\infty} \Phi^j \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{52}$$

$$\lim_{j\to\infty} \frac{\partial E_t Y_{t+j}}{\partial \epsilon_t} = \lim_{j\to\infty} \beta_1 \frac{\partial E_t Y_{t+j-1}}{\partial \epsilon_t} + \lim_{j\to\infty} \beta_2 \frac{\partial E_t Y_{t+j-2}}{\partial \epsilon_t} \tag{53}$$

4 Exercise 6.6

至 FRED 下載日本 CPI (code: JPNCPIALLQINMEI) 與美國實質 GDP (code: GDPC1)。將日本 CPI 取對數,命名為 `lcpi_jp`,將美國實質 GDP 取對數後,命名為 `lrgdp_us`,樣本期間為 1972Q1-2020Q4。

```
library(quantmod)
```

```
## Loading required package: xts

##
## ###################################### WARNING ######################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                             #
## # If you call library(dplyr) later in this session, then calls to lag(my_xts) #
## # that you enter or source() into this session won't work correctly.          #
## #                                                                             #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                  #
## #                                                                             #
```

```
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop   #
## # dplyr from breaking base R's lag() function.                                 #
## ################################### WARNING ###################################
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
getSymbols("JPNCPIALLQINMEI", src="FRED")
```

```
## [1] "JPNCPIALLQINMEI"
```

```
getSymbols("GDPC1", src="FRED")
```

```
## [1] "GDPC1"
```

```
JPNCPIALLQINMEI <- JPNCPIALLQINMEI["1972-01-01/2020-10-01"]
GDPC1 <- GDPC1["1972-01-01/2020-10-01"]

lcpi_jp <- log(JPNCPIALLQINMEI)
lrgdp_us <- log(GDPC1)

df <- data.frame(lcpi_jp, lrgdp_us)
colnames(df) <- c("lcpi_jp", "lrgdp_us")
```

(a) 對 `lcpi_jp` 與 `lrgdp_us` 以 ADF, PP, KPSS, DF-GLS, ERS 以及 NP 檢定，落後期數以修正 AIC 決定。

It seems that R does not have a package that allows us to choose lags based on modified AIC. Also, no package for conducting NP test. We attach the test results conducted in EViews at the end. Below is some implementations using R.

```
library(urca)
summary(ur.df(lcpi_jp, type=c("drift"), lags=4))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
```

```
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020072 -0.003464 -0.000406  0.003091  0.053694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.160776   0.022693   7.085 2.83e-11 ***
## z.lag.1     -0.035105   0.004967  -7.067 3.13e-11 ***
## z.diff.lag1 -0.013710   0.069359  -0.198  0.84352
## z.diff.lag2  0.150250   0.067716   2.219  0.02772 *
## z.diff.lag3 -0.061221   0.064741  -0.946  0.34558
## z.diff.lag4  0.167986   0.063871   2.630  0.00925 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007121 on 185 degrees of freedom
## Multiple R-squared:  0.6555, Adjusted R-squared:  0.6462
## F-statistic: 70.39 on 5 and 185 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -7.067 25.277
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
summary(ur.pp(lcpi_jp, type = c("Z-alpha", "Z-tau"), model=c("constant"), use.lag=7))
```

```
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
```

```
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.029301 -0.003877 -0.000798  0.003529  0.057178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.167331   0.010669   15.68   <2e-16 ***
## y.l1        0.963641   0.002394  402.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.008134 on 193 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9988
## F-statistic: 1.62e+05 on 1 and 193 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-alpha  is: -7.1486
##
##          aux. Z statistics
## Z-tau-mu            12.886
```

```
summary(ur.kpss(lcpi_jp, type=c("mu"), use.lag = 11)) # type="mu": with a drift
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 11 lags.
##
## Value of test-statistic is: 1.1762
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
summary(ur.ers(lcpi_jp, type=c("DF-GLS"), model = c("constant"), lag.max=12))
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type DF-GLS
## detrending of series with intercept
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0116653 -0.0029310 -0.0006262  0.0026944  0.0206526
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## yd.lag         0.0002318  0.0005310   0.437  0.66303
## yd.diff.lag1   0.1470945  0.0738952   1.991  0.04813 *
## yd.diff.lag2   0.1492165  0.0727953   2.050  0.04192 *
## yd.diff.lag3   0.0576463  0.0729520   0.790  0.43052
## yd.diff.lag4   0.3378447  0.0727797   4.642 6.87e-06 ***
## yd.diff.lag5  -0.1013630  0.0606979  -1.670  0.09677 .
## yd.diff.lag6  -0.0427966  0.0598796  -0.715  0.47577
## yd.diff.lag7  -0.1016050  0.0588813  -1.726  0.08624 .
## yd.diff.lag8   0.1871693  0.0576911   3.244  0.00142 **
## yd.diff.lag9   0.0196598  0.0587031   0.335  0.73811
## yd.diff.lag10  0.0612920  0.0567269   1.080  0.28146
## yd.diff.lag11 -0.0825287  0.0552446  -1.494  0.13706
## yd.diff.lag12  0.1192060  0.0530335   2.248  0.02588 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005105 on 170 degrees of freedom
## Multiple R-squared:  0.6899, Adjusted R-squared:  0.6662
## F-statistic:  29.1 on 13 and 170 DF,  p-value: < 2.2e-16
##
```

```
##
## Value of test-statistic is: 0.4365
##
## Critical values of DF-GLS are:
##                   1pct  5pct 10pct
## critical values -2.58 -1.94 -1.62
```

```
summary(ur.ers(lcpi_jp, type=c("P-test"), model = c("constant"), lag.max=4))
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type P-test
## detrending of series with intercept
##
## Value of test-statistic is: 1823.174
##
## Critical values of P-test are:
##                 1pct 5pct 10pct
## critical values 1.91 3.17  4.33
```

```
summary(ur.df(lrgdp_us, type=c("drift"), lags=0))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.093085 -0.003119  0.000887  0.003880  0.070853
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.041604   0.019387   2.146   0.0331 *
## z.lag.1     -0.003774   0.002087  -1.808   0.0721 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0114 on 193 degrees of freedom
## Multiple R-squared:  0.01666,    Adjusted R-squared:  0.01157
## F-statistic:  3.27 on 1 and 193 DF,  p-value: 0.07211
##
##
## Value of test-statistic is: -1.8083 34.0592
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
summary(ur.pp(lrgdp_us, type = c("Z-alpha", "Z-tau"), model=c("constant"), use.lag=3))
```

```
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.093085 -0.003119  0.000887  0.003880  0.070853
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.041604   0.019387   2.146   0.0331 *
## y.l1        0.996226   0.002087 477.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.0114 on 193 degrees of freedom
## Multiple R-squared:  0.9992, Adjusted R-squared:  0.9991
## F-statistic: 2.279e+05 on 1 and 193 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-alpha  is: -0.736
##
##          aux. Z statistics
## Z-tau-mu             2.145
```

```
summary(ur.kpss(lrgdp_us, type=c("mu"), use.lag = 11)) # type="mu": with a drift
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 11 lags.
##
## Value of test-statistic is: 1.7259
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
summary(ur.ers(lrgdp_us, type=c("DF-GLS"), model = c("constant"), lag.max=11))
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type DF-GLS
## detrending of series with intercept
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.094777 -0.002742  0.001370  0.004716  0.071274
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## yd.lag         0.001498   0.001795   0.835    0.405
## yd.diff.lag1   0.008013   0.076739   0.104    0.917
## yd.diff.lag2   0.077299   0.094783   0.816    0.416
## yd.diff.lag3   0.119614   0.135620   0.882    0.379
## yd.diff.lag4   0.104070   0.132622   0.785    0.434
## yd.diff.lag5   0.044009   0.132807   0.331    0.741
## yd.diff.lag6   0.157403   0.131124   1.200    0.232
## yd.diff.lag7  -0.034947   0.131557  -0.266    0.791
## yd.diff.lag8  -0.201448   0.129473  -1.556    0.122
## yd.diff.lag9   0.190851   0.130312   1.465    0.145
## yd.diff.lag10  0.137779   0.129713   1.062    0.290
## yd.diff.lag11  0.107482   0.122402   0.878    0.381
##
## Residual standard error: 0.01194 on 172 degrees of freedom
## Multiple R-squared:  0.2355, Adjusted R-squared:  0.1821
## F-statistic: 4.414 on 12 and 172 DF,  p-value: 3.847e-06
##
##
## Value of test-statistic is: 0.8347
##
## Critical values of DF-GLS are:
##                   1pct  5pct 10pct
## critical values -2.58 -1.94 -1.62
```

```
summary(ur.ers(lrgdp_us, type=c("P-test"), model = c("constant"), lag.max=0))
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type P-test
## detrending of series with intercept
##
## Value of test-statistic is: 1415.955
##
## Critical values of P-test are:
##                 1pct 5pct 10pct
```

```
## critical values 1.91 3.17  4.33
```

(b) 估計以下迴歸式

$$\text{lrgdp\_us}_t = \alpha + \beta \text{lcpi\_jp}_t + e_t \tag{54}$$

報告並解釋結果。

```
regr <- lm(lrgdp_us~lcpi_jp, data=df)
summary(regr)
```

```
##
## Call:
## lm(formula = lrgdp_us ~ lcpi_jp, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39109 -0.21221 -0.03127  0.23007  0.45116
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.58831    0.31515   11.39   <2e-16 ***
## lcpi_jp      1.27994    0.07071   18.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2405 on 194 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6262
## F-statistic: 327.7 on 1 and 194 DF,  p-value: < 2.2e-16
```

$$\text{lrgdp\_us}_t = 3.58831 + 1.27994 \text{lcpi\_jp}_t \tag{55}$$

About a 1.28% increase in US GDP for every 1% increase in Japan CPI.

(c)

對 $\hat{e}_t = \text{lrgdp\_us}_t - \hat{\alpha} - \hat{\beta} \times \text{lcpi\_jp}_t$ 以 DF-GLS 做單根檢定，落後期數以修正 AIC 決定。

```
library(writexl)
df <- data.frame(lcpi_jp, lrgdp_us , regr$residuals )
colnames(df) <- c("lcpi_jp", "lrgdp_us", "e")
write_xlsx(df, path="output.xlsx")
```

```
# we include a constant
# very similar result to that of in Eveiws
summary(ur.ers(df$e, type="DF-GLS", model=c("constant"), lag.max = 4))
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type DF-GLS
## detrending of series with intercept
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.094679 -0.005532 -0.000509  0.005521  0.095170
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## yd.lag       -0.002489   0.002694  -0.924 0.356781
## yd.diff.lag1  0.194840   0.073900   2.637 0.009084 **
## yd.diff.lag2  0.326636   0.083521   3.911 0.000129 ***
## yd.diff.lag3  0.077155   0.094166   0.819 0.413632
## yd.diff.lag4  0.138489   0.090223   1.535 0.126493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01616 on 186 degrees of freedom
## Multiple R-squared:  0.324,  Adjusted R-squared:  0.3059
## F-statistic: 17.83 on 5 and 186 DF,  p-value: 1.998e-14
##
##
## Value of test-statistic is: -0.9238
##
## Critical values of DF-GLS are:
##                  1pct  5pct 10pct
## critical values -2.58 -1.94 -1.62
```