Econometrics

輪読中の Hayashi, Fumio. Econometrics. に関連する R プログラムなど。

最終更新時間：2014 年 08 月 28 日 18 時 31 分 42 秒

Ch. 1 Finite-Sample Properties of OLS

Empirical Exercise

初心者向けに html 形式で公開します。四角で囲まれた部分を順に R のコンソールにペーストしていけば動きます。R とそのインストールについては、リンク先参照。無償のソフトなので、とりあえずインストールして以下のプログラムを走らせてみることをおススメします。

・R ｊ pWiki
・R のインストール

・データセット : nerlove.txt

NERLOVE.ASC と（最後の余計な文字を除いて）実質的に同一です。データ読み込みをこのページから直接できるようにするための措置です。

```
# データの読み込み
Elec<-read.table("http://www.rhasumi.net/data/econometrics/nerlove.txt")
# データセットの作成
LTC<-log(Elec[,1]); LQ <- log(Elec[,2]); LPL <- log(Elec[,3]);
LPF <- log(Elec[,4]); LPK <- log(Elec[,5])
```

#(b) OLS 推定 Model (1.7.4)

```
result_b<-lm(LTC~LQ+LPL+LPF+LPK)
# 結果の表示
# cf. (1.7.7) in p. 65
summary(result_b)
```

(c) Model (1.7.6)

```
LTC1<-LTC-LPF; LPL1<-LPL-LPF; LPK1<-LPK-LPF;
result_c<-lm(LTC1~LQ+LPL1+LPK1)
# cf. (1.7.8) in p. 66
summary(result_c)
SSR_c<-deviance(result_c)
```

#(d) Estimation of Model 1

```
result_d <- vector("list", 5)

for( j in 1:5){
   jj <- (1+(j-1)*29):(j*29)
   result_d[[j]] <- lm(LTC1[jj]~LQ[jj]+LPL1[jj]+LPK1[jj])
}
for( j in 1:5)
```

```
  print( summary(result_d[[j]]))
# SSR
SSR_d <- c()
for( j in 1:5)
  SSR_d[j] <- deviance(result_d[[j]])
```

#(e) Estimation of Model 2

```
DD<-matrix(0,ncol=20,nrow=145)
for(i in 0:4){
  DD[(1+i*29):(29+i*29),(1+i*4):(4+i*4)]<-1
 }
X2<-matrix(0,ncol=4,nrow=145)
X2[,1]<-1
X2[,2]<-LQ
X2[,3]<-LPL1
X2[,4]<-LPK1
Xlist<-cbind(X2,X2,X2,X2,X2)
XX<- DD * Xlist

result_e<-lm(LTC1~XX-1)
summary(result_e)
SSR_e <- deviance(result_e)
# Check that the SSRs are almost the same
SSR_e - sum(SSR_d)
```

(f) F test

```
F_f <- ((SSR_c - SSR_e) / 16 ) / (SSR_e / (125) )
PF_f <- 1 - pf(F_f,16,125)
PF_f
```

(g) Estimation of Model 3

```
XX_g<-matrix(0,ncol=12,nrow=145)
for(i in 0:4){
  XX_g[(1+29*i):(29+29*i),(1+2*i)]<-1
  XX_g[(1+29*i):(29+29*i),(2+2*i)]<-log(Elec[(1+29*i):(29+29*i),2])
 }
XX_g[,11]<-log(Elec[,3]/Elec[,4])
XX_g[,12]<-log(Elec[,5]/Elec[,4])

result_g <- lm(LTC1~XX_g-1)
summary(result_g)
# F test
SSR_g<-deviance(result_g)
F_g <- ((SSR_g - SSR_e) / 8 ) / (SSR_e / 125 )
PF_g <- 1 - pf(F_g,8,125)
PF_g
```

(h) Estimation of Model 4

```
weight <- (0.0565 + 2.1377/Elec[,2])
y_h <- log( Elec[[1]] / Elec[[4]]) / sqrt( weight )

x1_h <- rep(1, 145) / sqrt( weight )
x2_h <- log( Elec[[2]] ) / sqrt( weight )
x3_h <- log( Elec[[2]] ) ^2 / sqrt( weight )
x4_h <- log( Elec[[3]] / Elec[[4]]) / sqrt( weight )
x5_h <- log( Elec[[5]] / Elec[[4]]) / sqrt( weight )

result_h<-lm(y_h~x1_h + x2_h + x3_h + x4_h + x5_h -1 )
summary(result_h)

y_h2 <- log( Elec[[1]] / Elec[[4]])
x1_h2 <- rep(1, 145)
x2_h2 <- log( Elec[[2]] )
x3_h2 <- log( Elec[[2]] ) ^2
x4_h2 <- log( Elec[[3]] / Elec[[4]])
```

```
x5_h2 <- log( Elec[[5]] / Elec[[4]])

# これでも可
# Weighted Liest Squares
weight_h <- 1/weight
result_hh <-lm(y_h2~x1_h2 + x2_h2 + x3_h2 + x4_h2 + x5_h2 -1, weights=weight_h)
summary(result_hh)

# OLS
result_h2<-lm(y_h2~x1_h2 + x2_h2 + x3_h2 + x4_h2 + x5_h2 -1 )
# (0.0565 + 2.1377/Q)
y_h3 <- resid(result_h2)^2
invQ <- 1 / Elec[[2]]
result_h3 <- lm( y_h3 ~ invQ )
result_h3
```

グラフ出力

```
plot(log(Elec[,2]) ,y_h3, main = "QQ plot", xlab = "ln Q", ylab = "residual^2")
lines(log(Elec[,2]), predict( result_h3  ) )
lines(log(Elec[,2]), weight , col=2 )


plot(log(Elec[,2]),resid(result_c), main = "QQ plot", xlab = "ln Q", ylab ="residual")
abline(h=0)


plot(Elec[,2],resid(result_h), main = "QQ plot", xlab = "Q", ylab = "residual")
abline(h=0)


plot(log(Elec[,2]),resid(result_h), main = "QQ plot", xlab = "ln Q", ylab = "residual")
abline(h=0)
```

MCMC (bayesian)

Model (1.7.4)

```
library(MCMCpack) # this package is required.
posterior  <- MCMCregress(LTC~LQ+LPL+LPF+LPK, verbose=1000)
plot(posterior)
raftery.diag(posterior)
summary(posterior)
```

Monte Carlo Exercise

・プログラム : monte_carlo_ch1.r

Ch. 2 Large-Sample Theory

Empirical Exercise

一部分のみ

```
data <- read.table("MISHKIN.asc",header=F)
len <- length(data[[1]])
pai1 <- data[[3]][2:len]
tb1  <- data[[5]][2:len];
cpi  <- data[[7]]

pai <- ((cpi[2:len]/cpi[1:(len-1)])^12-1)*100
r <- tb1 - pai
r_1 <- r[35:(35+223-1)]
mean(r_1)
sd(r_1)
```

```
gam_hat <- c()
rho_hat <- c()
std_err <- c()
lb_q <- c()
p_val <- c()
n <- 223
p <- 12
```

#(d) Table 2.1. の再現

```
var_z <- sum((r_1-mean(r_1))^2)/n
for (j in 1:p)
  gam_hat[j] <- sum( (r_1[1:(n-j)]-mean(r_1))*(r_1[(1+j):n]-mean(r_1)) )/n
rho_hat <- gam_hat/var_z

for (j in 1:p){
  lb_q[j] <- n*(n+2) * sum( rho_hat[1:j]^2 / (n-c(1:j)))
  p_val[j] <- (1-pchisq(lb_q[j],j))
}

print(rho_hat ,digits=1)
print(1/sqrt(n),digits=2)
print(lb_q,digits=)
print(p_val*100,digits=1)
```

#(e) 2.11.9 の再現

```
pai_1 <- pai[35:(35+223-1)]
tb1_1 <- tb1[35:(35+223-1)]
ols_e <- lm(pai_1~ tb1_1)

summary(ols_e)

xx <- cbind(rep(1, length(tb1_1)),tb1_1)
g <- xx * resid(ols_e)
Sxx <- t(xx) %*% xx
Avar_b <- sqrt( solve(Sxx) %*% (t(g)%*%g) %*% solve(Sxx) )
Avar_b[1,1]
Avar_b[2,2]
```

(h) Breusch-Godfrey test

```
resid_h <- c(rep(0, 12), resid(ols_e))
AR12 <- arima(resid_h, order=c(12,0,0),include.mean=F)
summary(AR12)
R_sq <- 1 - sum(resid(AR12)^2)/sum( (resid_h-mean(resid_h) )^2 )
R_sq * length(resid(ols_e))
```

Monte Carlo Exercise

・プログラム : monte_carlo_ch2.r

Ch. 3 Single-Equation GMM

（古いかも）

・プログラム hayashi_ch3.r

grilic.csv は grilic.xls を csv 形式で保存したものです。

Ch. 4 Multiple-Equation GMM

（古いかも）

・プログラム hayashi_ch4.r

greene.csv は greene.xls を csv 形式で保存したものです。

Ch. 5 Panel Data

・プログラム hayashi_ch5.r

Ch. 6 Serial Correlation

・プログラム ch_6.r

Ch. 7, 8 はなし

Ch. 9 Unit-Root Econometrics

・プログラム ch_9.r

Ch. 10 Cointegration

・プログラム ch_10.r

関連文献およびリンク

・Gate to Fumio Hayashi's Homepage

・Hayashi, Fumio. Econometrics. Princeton University Press, 2000.