HW3\_*4*.R

USER

Tue Dec 11 00:38:23 2018

library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

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

library(forecast)  
rm(list=ls())  
setwd("C:\\Users\\USER\\Documents\\Github\\Econometrics\\시계열 분석\\HW3")  
rv.data<-read.csv("RV\_IV\_data.csv", header=TRUE)  
head(rv.data)

## Date VKOSPI VIX SNP.RV KOSPI.RV  
## 1 2003-01-06 33.04 24.91 0.008186903 0.01243283  
## 2 2003-01-07 33.98 25.13 0.010548387 0.01506374  
## 3 2003-01-08 33.17 25.53 0.011272095 0.01632494  
## 4 2003-01-09 34.02 24.25 0.010671167 0.01859559  
## 5 2003-01-10 38.05 24.32 0.012709980 0.03435511  
## 6 2003-01-13 37.56 24.90 0.011114250 0.01659640

#============================#  
# VIX = Y, SNP.RV = X #  
#============================#  
# MAKE DATA  
y.t = rv.data$VIX[1:nrow(rv.data)] # VIX  
head(y.t); length(y.t)

## [1] 24.91 25.13 25.53 24.25 24.32 24.90

## [1] 3558

x.t = rv.data$SNP.RV[1:(nrow(rv.data))] # SNP.RV  
head(x.t); length(x.t)

## [1] 0.008186903 0.010548387 0.011272095 0.010671167 0.012709980 0.011114250

## [1] 3558

# AR 모형 BIC : p=5  
AR.bic = c()  
for( p in 1:10){  
 AR.fit = arima(y.t, order = c(p,0,0))  
 AR.bic[p] = AIC(AR.fit,k = log(length(y.t)))  
}  
par(mfrow=c(1,2))  
plot(AR.bic, type = "b", pch =19, main = "BIC", ylab = "", xlab = "p")  
abline(v = which.min(AR.bic), col=2, lty = 2) # AR(5)  
  
y\_t = y.t[6:length(y.t)]  
y\_t\_1 = y.t[5:(length(y.t)-1)]  
y\_t\_2 = y.t[4:(length(y.t)-2)]  
y\_t\_3 = y.t[3:(length(y.t)-3)]  
y\_t\_4 = y.t[2:(length(y.t)-4)]  
y\_t\_5 = y.t[1:(length(y.t)-5)]  
  
x\_t = x.t[6:length(x.t)]  
x\_t\_1 = x.t[5:(length(x.t)-1)]  
x\_t\_2 = x.t[4:(length(x.t)-2)]  
x\_t\_3 = x.t[3:(length(x.t)-3)]  
x\_t\_4 = x.t[2:(length(x.t)-4)]  
x\_t\_5 = x.t[1:(length(x.t)-5)]  
  
y.data = cbind(y\_t\_1 = y\_t\_1, y\_t\_2= y\_t\_2, y\_t\_3 = y\_t\_3,   
 y\_t\_4 = y\_t\_4, y\_t\_5 = y\_t\_5)  
x.data = cbind(x\_t\_1 = x\_t\_1, x\_t\_2= x\_t\_2, x\_t\_3 = x\_t\_3,   
 x\_t\_4 = x\_t\_4, x\_t\_5 = x\_t\_5)  
  
# ADL 모형 BIC : p=5, q=2  
ADL.bic = matrix(0, ncol = 5, nrow = 5)  
for( p in 1:5){  
 for(q in 1:5){   
 ADL.fit = lm(y\_t ~ y.data[,1:p] + x.data[,1:q])   
 ADL.bic[p,q] = AIC(ADL.fit,k = log(length(y.t)))  
 }  
}  
colnames(ADL.bic) = c("q=1", "q=2", "q=3", "q=4", "q=5")  
row.names(ADL.bic) = c("p=1", "p=2", "p=3", "p=4", "p=5")  
ADL.bic; min(ADL.bic) # ADL(5,2)

## q=1 q=2 q=3 q=4 q=5  
## p=1 13997.55 13996.56 13961.55 13959.49 13961.68  
## p=2 13952.31 13926.94 13916.32 13915.06 13919.92  
## p=3 13920.84 13894.67 13898.65 13903.59 13908.70  
## p=4 13923.04 13899.70 13904.10 13910.71 13916.54  
## p=5 13890.32 13877.14 13879.06 13886.47 13894.03

## [1] 13877.14

## 1-step ahead forecasting  
AR5.fore = c(); ADL52.fore = c()  
for(i in 3106:(length(y.t)-1)){  
 train.data = data.frame(y.t = y.t[1:i], x.t = x.t[1:i])  
   
 y\_t = train.data[6:nrow(train.data),1]  
 y\_t\_1 = train.data[5:(nrow(train.data)-1),1]  
 y\_t\_2 = train.data[4:(nrow(train.data)-2),1]  
 y\_t\_3 = train.data[3:(nrow(train.data)-3),1]  
 y\_t\_4 = train.data[2:(nrow(train.data)-4),1]  
 y\_t\_5 = train.data[1:(nrow(train.data)-5),1]  
  
 x\_t = train.data[6:nrow(train.data),2]  
 x\_t\_1 = train.data[5:(nrow(train.data)-1),2]  
 x\_t\_2 = train.data[4:(nrow(train.data)-2),2]  
 x\_t\_3 = train.data[3:(nrow(train.data)-3),2]  
 x\_t\_4 = train.data[2:(nrow(train.data)-4),2]  
 x\_t\_5 = train.data[1:(nrow(train.data)-5),2]  
   
 # AR(5)  
 AR5.fit = lm(y\_t ~ y\_t\_1 + y\_t\_2 + y\_t\_3 + y\_t\_4 + y\_t\_5 )  
 AR5.fore[i-3105] = sum(AR5.fit$coef\*c(1, y\_t[length(y\_t)],   
 y\_t\_1[length(y\_t)],   
 y\_t\_2[length(y\_t)],   
 y\_t\_3[length(y\_t)],  
 y\_t\_4[length(y\_t)] ))   
 # ADL(5,2)  
 ADL.fit1 = lm(y\_t ~   
 y\_t\_1 + y\_t\_2 +y\_t\_3 + y\_t\_4 + y\_t\_5 +   
 x\_t + x\_t\_1 );  
 ADL52.fore[i-3105] = sum(ADL.fit1$coef\*c(1, y\_t[length(y\_t)],   
 y\_t\_1[length(y\_t)],   
 y\_t\_2[length(y\_t)],   
 y\_t\_3[length(y\_t)],  
 y\_t\_4[length(y\_t)],  
 x\_t[length(y\_t)],   
 x\_t\_1[length(y\_t)] ))  
}  
  
# 1. VKOSPI가 KOSPI 5분 실현변동성을 GRANGER CAUSE하는가  
grangertest(y.t~x.t)

## Granger causality test  
##   
## Model 1: y.t ~ Lags(y.t, 1:1) + Lags(x.t, 1:1)  
## Model 2: y.t ~ Lags(y.t, 1:1)  
## Res.Df Df F Pr(>F)  
## 1 3554   
## 2 3555 -1 1.0976 0.2949

# 2. 예측력 비교  
# AR(5) VS ADL(5,2)   
# AR5.fore VS ADL52.fore  
# MAE  
AR5.MAE = mean(abs(AR5.fore - y.t[3107:length(y.t)]))  
ADL52.MAE = mean(abs(ADL52.fore - y.t[3107:length(y.t)]))  
AR5.MAE < ADL52.MAE # AR(5) 모형이 MAE 값이 더 작음

## [1] TRUE

# MSE  
AR5.MSE = mean((AR5.fore - y.t[3107:length(y.t)])^2)  
ADL52.MSE = mean((ADL52.fore - y.t[3107:length(y.t)])^2)  
AR5.MSE < ADL52.MSE # AR(5) 모형이 MSE 값이 더 작음

## [1] TRUE

result = matrix(c(AR5.MAE, ADL52.MAE, AR5.MSE, ADL52.MSE), nrow = 2, byrow = T)  
row.names(result) = c("MAE", "MSE")  
colnames(result) = c("AR5", "ADL(5,2)")  
result # AR(5) 모형이 더 잘 예측

## AR5 ADL(5,2)  
## MAE 0.7955403 0.8435758  
## MSE 1.4964288 1.6484881

