Ch 11 : Multivariate ARMA Model

Contents

1 Testing Independence of two TS **2**
1.1 Testing Independence . . . 3
1.2 Vector ARMA . . . 16
1.3 Example: Lead Sales Data . . . 22
1.4 Example: DJI and Australian Index . . . 28

April 19, 2017

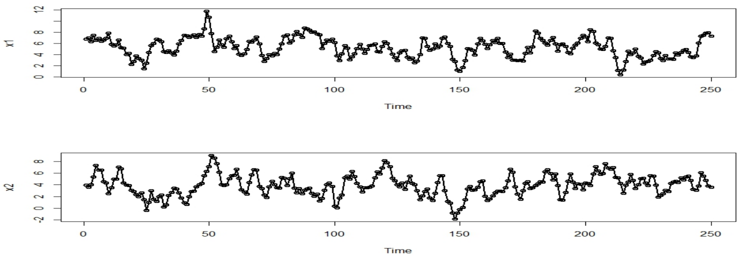
Testing Independence of Two TS

[ToC]

1.1 Testing Independence

[ToC]

If you have two series $\{Y_{t1}\}$ and $\{Y_{t2}\}$, how can we check if they are correlated with some lag?



```
#--- 1. Two independent stationary time series ---


  mu <- 5
  x1  <- rnorm(150)
  x2  <- rnorm(150)

  layout(matrix(1:2, 2, 1))
  plot(x1, type="o", lwd=2)
  plot(x2, type="o", lwd=2)

  acf( cbind(x1,x2) );   #- Can use this as a basis of correlation
```

```r
#--- 1. Two independent stationary time series ---
  mu <- 5
  x1  <- arima.sim(n = 250, list(ar = c(0.7), ma = c(0.5) )) + mu
  x2  <- arima.sim(n = 250, list(ar = c(0.7), ma = c(0.5) )) + mu


  layout(matrix(1:2, 2, 1))
  plot(x1, type="o", lwd=2)
  plot(x2, type="o", lwd=2)


  acf( cbind(x1,x2) );   #- Can NOT use this as a basis of correlation



  #--- Pre-whiten them
  Est1 <- arima(x1, order=c(1,0,1), include.mean=F)
  Est2 <- arima(x2, order=c(1,0,1), include.mean=F)

  acf( cbind(Est1$residuals,Est2$residuals) );
```

```
#--- 2. Two correlated stationary time series ---


  library(forecast)

  e1  <- rnorm(252)

  x1 <- e1[1:250]
  x2 <- e1[1:250] + .75*e1[3:252]


  layout(matrix(1:2, 2, 1))
  plot(x1, type="o")
  plot(x2, type="o")


  acf( cbind(x1,x2) );



  #-- Pre-whiten them
  Est2 <- arima(x2, order=c(0,0,2), include.mean=FALSE ); Est2

  r12  <- ts( cbind(x1, Est2$residuals) )
  acf(r12)
```

```
#-- Pre-whiten them
Est1 <- auto.arima(x1); Est1
Est2 <- auto.arima(x2); Est2

r12  <- ts( cbind(Est1$residuals, Est2$residuals) )
acf(r12)
```

```
#--- 3. Another correlated stationary time series ---


  e1  <- rnorm(252)
  e2  <- rnorm(252)

  x1 <- e1[1:250] + .5*e1[2:251]
  x2 <- e2[1:250] + .7*e1[2:251] + .4*e2[3:252]


  layout(matrix(1:2, 2, 1))
  plot(x1, type="o")
  plot(x2, type="o")


  acf( cbind(x1,x2) );


  #-- Pre-whiten them
  Est1 <- auto.arima(x1); Est1
  Est2 <- auto.arima(x2); Est2

  r12  <- ts( cbind(Est1$residuals, Est2$residuals) )
  acf(r12)
```

```
#-- Independent version ---
e1  <- rnorm(252)
e2  <- rnorm(252)

x1 <- e1[1:250] + .5*e1[2:251]
x2 <- e2[1:250] + .7*e2[2:251] + .4*e2[3:252]

acf( cbind(x1,x2) );

#-- Pre-whiten them
Est1 <- auto.arima(x1); Est1
Est2 <- auto.arima(x2); Est2

r12  <- ts( cbind(Est1$residuals, Est2$residuals) )
acf(r12)
```

```
#--- 4. Correlated AR stationary time series ---

  e2  <- rnorm(250)

  x1  <- arima.sim(n = 252, list(ar = c(0.7) )) + mu
  x2  <- x1[3:252] + .3*x1[2:251] + e2
  x1  <- x1[1:250]


  layout(matrix(1:2, 2, 1))
  plot(x1, type="o")
  plot(x2, type="o")


  acf( cbind(x1,x2) );


  #-- Pre-whiten them
  Est1 <- auto.arima(x1); Est1
  Est2 <- auto.arima(x2); Est2

  r12  <- ts( cbind(Est1$residuals, Est2$residuals) )
  acf(r12)
```

```
#-- Independent version ---
e1  <- rnorm(252)
e2  <- rnorm(252)

x1 <- e1[1:250] + .5*e1[2:251]
x2 <- e2[1:250] + .7*e2[2:251] + .4*e2[3:252]

#-- Pre-whiten them
Est1 <- auto.arima(x1); Est1
Est2 <- auto.arima(x2); Est2

r12  <- ts( cbind(Est1$residuals, Est2$residuals) )
acf(r12)
```

Cross-ACVF Matrix

$$\mathbf{\Gamma}(h) = \begin{bmatrix} \operatorname{Cov}(Y_{t,1}, Y_{t+h,1}) & \operatorname{Cov}(Y_{t,1}, Y_{t+h,2}) \\ \operatorname{Cov}(Y_{t,2}, Y_{t+h,1}) & \operatorname{Cov}(Y_{t,2}, Y_{t+h,2}) \end{bmatrix} = \begin{bmatrix} \gamma_{11}(h) & \gamma_{12}(h) \\ \gamma_{21}(h) & \gamma_{22}(h) \end{bmatrix}$$

$$\gamma_{11}(-h) = \operatorname{Cov}(Y_{t,1}, Y_{t-h,1}) = \operatorname{Cov}(Y_{t+h,1}, Y_{t,1}) = \gamma_{11}(h)$$

$$\gamma_{12}(-h) = \operatorname{Cov}(Y_{t,1}, Y_{t-h,2}) = \operatorname{Cov}(Y_{t+h,1}, Y_{t,2}) = \operatorname{Cov}(Y_{t,2}, Y_{t+h,1}) = \gamma_{21}(h)$$

Problem with Cross-ACF plot

Bartlett's Formula:

$$\sqrt{n}\hat{\rho}_{12} \quad \sim \quad N\Big(0, \sum_{j=-\infty}^{\infty} \rho_{11}(j)\rho_{22}(j)\Big)$$

Cross-ACF of two independent WN:

$$\begin{aligned} Y_{t,1} &= e_{t1} \qquad & e_{t,1} \sim_{iid} N(0,\sigma^2) \\ Y_{t,2} &= e_{t2} \qquad & e_{t,2} \sim_{iid} N(0,\sigma^2) \end{aligned}$$

and e_{t1} and e_{t2} are independent.

Cross-ACF of two independent ARMA(1,1):

$$\begin{aligned} Y_{t,1} - .5Y_{t-1,1} &= e_{t,1} - .3e_{t-11} \qquad & e_{t1} \sim_{iid} N(0,\sigma^2) \\ Y_{t,2} - .8Y_{t-1,2} &= e_{t,2} - .2e_{t-12} \qquad & e_{t2} \sim_{iid} N(0,\sigma^2) \end{aligned}$$

and e_{t1} and e_{t2} are independent.

Pre-whitening

1. Unless Y_{t1} and Y_{t2} are white noise themselves, $\hat{\rho}_{12}(h)$ cannot be used to check if Y_{t1} and Y_{t2} are uncorrelated, because its variance depends on ρ_{11} and ρ_{22}.

2. We firtst fit Y_{t1} and Y_{t2} individually with ARMA, then check if residuals $\hat{e}_{t1}$ and $\hat{e}_{t2}$ are independent.

1.2 Vector ARMA

[ToC]

VMA

$$
\begin{aligned}
Y_{t1} &= e_t + .5a_{t-1} \qquad\qquad e_t \sim IID(0, \sigma_1^2) \\
Y_{t2} &= a_t + .3e_{t-1} + .4a_{t-2} \quad a_t \sim IID(0, \sigma_1^2)
\end{aligned}
$$

$$
\begin{bmatrix} Y_{t1} \\ Y_{t2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_t \\ a_t \end{bmatrix} + \begin{bmatrix} 0 & .5 \\ .3 & 0 \end{bmatrix} \begin{bmatrix} e_{t-1} \\ a_{t-1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & .4 \end{bmatrix} \begin{bmatrix} e_{t-2} \\ a_{t-2} \end{bmatrix}
$$

$$
\begin{bmatrix} Y_{t1} \\ Y_{t2} \end{bmatrix} = \left(\boldsymbol{I} + \boldsymbol{\Theta}_1 B + \boldsymbol{\Theta}_2 B^2 \right) \begin{bmatrix} e_t \\ a_t \end{bmatrix}
$$

$$
\begin{aligned}
Y_{t1} &= e_t + .5a_{t-1} \\
Y_{t2} &= a_t + .3e_{t-1} + .4a_{t-2}
\end{aligned}
$$

$$
\begin{aligned}
\gamma_{12}(1) &= \text{Cov}(Y_{t+1,1}, \quad Y_{t,2}) \\
&= \text{Cov}(.3Y_{t,1} + e_{t+1}, \quad -.5Y_{t-1,1} + .3Y_{t-2,2} + a_t) \\
&= E(.3Y_{t,1}, -.5Y_{t-1,1}) + E(.3Y_{t,1}, .3Y_{t-2,2}) \\
&= -(.5)(.3)\gamma_{11}(1) + (.3)(.3)\gamma_{12}(2)
\end{aligned}
$$

VAR

$$
\begin{aligned}
Y_{t,1} - .3Y_{t-1,1} &= e_t \qquad\qquad e_t \sim IID(0, \sigma_1^2) \\
Y_{t,2} - .5Y_{t-1,1} + .3Y_{t-2,2} &= a_t \quad a_t \sim IID(0, \sigma_1^2)
\end{aligned}
$$

$$
\begin{bmatrix} Y_{t1} \\ Y_{t2} \end{bmatrix} - \begin{bmatrix} .3 & 0 \\ 0 & .5 \end{bmatrix} \begin{bmatrix} Y_{t-1,1} \\ Y_{t-1,2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -.3 \end{bmatrix} \begin{bmatrix} Y_{t-2,1} \\ Y_{t-2,2} \end{bmatrix} = \begin{bmatrix} e_{t-2} \\ a_{t-2} \end{bmatrix}
$$

$$
\left(\boldsymbol{I} + \boldsymbol{\Phi}_1 B + \boldsymbol{\Phi}_2 B^2\right) \begin{bmatrix} Y_{t,1} \\ Y_{t,2} \end{bmatrix} = \begin{bmatrix} e_t \\ a_t \end{bmatrix}
$$

$$
\begin{aligned}
Y_{t,1} - .3Y_{t-1,1} &= e_t \\
Y_{t,2} - .5Y_{t-1,1} + .3Y_{t-2,2} &= a_t
\end{aligned}
$$

$$
\begin{aligned}
\gamma_{12}(1) &= \mathrm{Cov}(Y_{t+1,1}, \quad Y_{t,2}) \\
&= \mathrm{Cov}(.3Y_{t,1} + e_{t+1}, \quad -.5Y_{t-1,1} + .3Y_{t-2,2} + a_t) \\
&= E(.3Y_{t,1}, -.5Y_{t-1,1}) + E(.3Y_{t,1}, .3Y_{t-2,2}) \\
&= -(.5)(.3)\gamma_{11}(1) + (.3)(.3)\gamma_{12}(2)
\end{aligned}
$$

Causality of VAR

If

$$\det\left(\boldsymbol{I} - \boldsymbol{\Phi}_1 z - \cdots - \Phi_p z^p\right)$$

has all roots outside of unit circle, then VARMA(p,q) is causal. i.e.

$$\boldsymbol{Y}_t = \sum_{i=0}^{\infty} \boldsymbol{\Psi}_j \boldsymbol{e}_{t-j}$$

Matrices $\boldsymbol{\Psi}_j$ can be found recursively, by equation

$$\boldsymbol{\Psi}_j = \boldsymbol{\Theta}_j + \sum_{i=1}^{n} \boldsymbol{\Phi}_k \boldsymbol{\Psi}_{j-k} \qquad \boldsymbol{\Psi}_0 = \boldsymbol{I}.$$

Prediction with VARMA

For MA(1)

$$Y_t = e_t - \theta_1 e_{t-1} \qquad \hat{Y}(1) = -\theta_1 \hat{e}_n$$

For AR(1)

$$Y_t = \phi_1 Y_{t-1} + e_t \qquad \hat{Y}(1) = \phi_1 Y_n$$

1.3 Example: Lead Sales Data

[ToC]

Brockwell (2002) p228, 248.

```
library(forecast)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")
D  <- read.table("http://gozips.uakron.edu/~nmimoto/pages/datasets/LS2.txt", header=T)

I <- ts(D$Index, start=c(2000,1), freq=12)  # - indicator
S <- ts(D$Sales, start=c(2000,1), freq=12)  # - sales

layout(matrix(1:2, 2, 1))
plot(I, type='o');  plot(S, type='o')

layout(1,1,1);  acf(D)

#--- Take difference ---
S1 <- diff(S);   I1 <- diff(I)

layout(matrix(1:2, 2, 1))
plot(S1, type='o'); plot(I1, type='o')

layout(1,1,1);  acf(cbind(S1, I1))
```

```
#-- Pre-whiten them
Fit1 <- auto.arima(S1); Fit1
Fit2 <- auto.arima(I1); Fit2

acf( cbind(Fit1$resid, Fit2$resid) )


cor(Fit1$resid[1:141], Fit2$resid[4:144])
cor(Fit1$resid[4:144], Fit2$resid[1:141])  # Current Sales is correlated with Indicator 3 mo. ago
```

```
#-- use vector ARMA -- #   install.package("MTS")
library(MTS)

Fit3 <- VARMA(cbind(S1,I1), p=0, q=1, include.mean=T)
refVARMA(Fit3, thres=2)


Fit4  <- VARMA(cbind(S1,I1), p=0, q=3, include.mean=T)
Fit4b <- refVARMA(Fit4, thres=2)
Fit4c <- refVARMA(Fit4b, thres=2)     #- AIC = -5.04

Randomness.tests( Fit4c$resid[,1] );  Randomness.tests( Fit4c$resid[,2] )
acf(Fit4c$resid)


Fit5  <- VARMA(cbind(S1,I1), p=1, q=3, include.mean=T)
Fit5b <- refVARMA(Fit5,  thres=1)     #- AIC = -3.33

Randomness.tests( Fit5b$resid[,1] );  Randomness.tests( Fit5b$resid[,2] )


Fit6  <- VARMA(cbind(S1,I1), p=2, q=3, include.mean=T)
Fit6b <- refVARMA(Fit6,   thres=1)
Fit6c <- refVARMA(Fit6b,  thres=1)     #- AIC = -5.37

Randomness.tests( Fit6c$resid[,1] );  Randomness.tests( Fit6c$resid[,2] )
```

```
#-- 10-step Prediction --
VARMApred(Fit6c, h=10)

S1.h[i] <- VARMApred(Fit6c, h=1)$pred[1]
I1.h[i] <- VARMApred(Fit6c, h=1)$pred[2]


#-- Rolling 1-step Prediction --
S2.h <- 0
I2.h <- 0
for (i in 1:40) {

  print(i)

  A2 <- window(S1, start=2+i-1, end=202+i-1)
  B2 <- window(I1, start=2+i-1, end=202+i-1)

  Fit3 <- VARMA(cbind(S2,I2), p=0, q=1, include.mean=FALSE)

  A2.h[i] <- VARMApred(Fit3, h=1)$pred[1]
  B2.h[i] <- VARMApred(Fit3, h=1)$pred[2]

}


A3 <- window(A1, start=203, end=242)
```

```
B3 <- window(B1, start=203, end=242)
A2.h <- ts(A2.h, start=203)
B2.h <- ts(B2.h, start=203)


layout(matrix(1:4, 4, 1))
plot(A3, ylim=c(-.02, .02) ); lines(A2.h, col="red")
plot(B3, ylim=c(-.02, .02) ); lines(B2.h, col="red")
plot(abs(A3-A2.h) , ylim=c(0, .02) )
plot(abs(B3-B2.h) , ylim=c(0, .02) )


Result1 <- cbind(A3>0, A2.h>0, (A3>0)==(A2.h>0) )
Result2 <- cbind(B3>0, B2.h>0, (B3>0)==(B2.h>0) )
colMeans(Result1)
colMeans(Result2)

MSE1 <- mean( (A3-A2.h)^2 )/ sd(A3) ; MSE1
MSE2 <- mean( (B3-B2.h)^2 )/ sd(B3) ; MSE2

layout(matrix(1:4, 4, 1))
plot( as.numeric(A3) * ((A2.h>0)*2-1) )
plot( as.numeric(B3) * ((B2.h>0)*2-1) )
plot( cumsum( as.numeric(A3) * ((A2.h>0)*2-1) ) )
plot( cumsum( as.numeric(B3) * ((B2.h>0)*2-1) ) )
```

```
.04/sum(abs(A3))
.14/sum(abs(B3))
```

```
#--- Rolling 1-step predicton

Rolling.len = 24  #- Size of out-sample (validation)
Window.size = 125 #- Size of in-sample (training)

p = 2 # VARMA parameter
d = 0
q = 0

Y <- cbind(S1,I1)

Y.hat <- matrix(0, Rolling.len, 2)   #- Initialize
for (i in 1:Rolling.len) {
  w.bgn <- i
  w.end <- i+Window.size-1

  Fit0  <- VARMA(Y[w.bgn:w.end, ], p=2, q=3, include.mean=T)
  Fit0b <- refVARMA(Fit0,   thres=1)
  Fit0c <- refVARMA(Fit0b,  thres=1)

  S1.h<- VARMApred(Fit0c, h=1)$pred[1]
  I1.h<- VARMApred(Fit0c, h=1)$pred[2]
  Y.hat[i,] <- cbind(S1.h, I1.h)

}
```

```
X  <- window(Y, start=time(Y)[1], end=time(Y)[Window.size])
Y2 <- window(Y, start=time(Y)[Window.size+1], end=time(Y)[Window.size+Rolling.len])
Yhat <- ts(Y.hat, start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y) )

Yhat.CIu <- ts(Yhat.CIu, start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y) )
Yhat.CIl <- ts(Yhat.CIl, start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y) )

Pred.error <- Y2-Yhat
Pred.rMSE = sqrt( mean( (Pred.error)^2 ) ) #- prediction root Mean Squared Error
Pred.rMSE

mean(Pred.error)


layout(matrix(c(1,1,1,2,2,3), 2, 3, byrow=TRUE))

plot(Y, type="o", col="blue", main=paste("Rolling 1-step prediction with window size", Window.size) ) #- Entire dataset
lines(X, type="o")
lines(Yhat, type="o", col="red")
lines(Yhat.CIu, type="l", col="red", lty=2)
lines(Yhat.CIl, type="l", col="red", lty=2)

plot(Pred.error, type="o", main="Prediction Error (Blue-Red)")
abline(h=c(-1.96, 1.96), col="blue", lty=2)
abline(h=0)
```

```
acf(Pred.error, main="ACF of prediction error")
```

1.4 Example: DJI and Australian Index

[ToC]

Brockwell (2002) p248.

```
library(forecast)
source("http://gozips.uakron.edu/~nmimoto/689/TS_R-90.txt")

D  <- read.table("http://gozips.uakron.edu/~nmimoto/pages/datasets/djao2.txt", header=T)
A <- ts(D$DJ, start=c(1,1), freq=1)
B <- ts(D$AO, start=c(1,1), freq=1)

layout(matrix(1:2, 2, 1))
plot(A, type='o')
plot(B, type='o')
layout(1,1,1)


A1 <- diff(log(A))
B1 <- diff(log(B))

layout(matrix(1:2, 2, 1))
plot(A1, type='o')
plot(B1, type='o')
```

```
layout(1,1,1)

acf(cbind(A1, B1))



#-- Check what h=1 means --
x <- rnorm(101)

x1 <- x[(10:100)]
y1 <- x[(10:100)-2]
acf( cbind(x1, y1) )
cbind(x1, y1)          #h=-2 means x1 is 2-days early


acf( cbind(A1, B1) )

plot( A1[10:250], B1[10:250] )


#-- Pre-whiten them
Fit1 <- auto.arima(A1); Fit1
Fit2 <- auto.arima(B1); Fit2

acf( cbind(Fit1$residuals, Fit2$residuals) )
```

```
#-- use vector ARMA --
install.packages("MTS")
library(MTS)


Fit3 <- VARMA(cbind(A1,B1), p=0, q=1, include.mean=FALSE)
Randomness.tests( Fit3$residuals[,1] )
Randomness.tests( Fit3$residuals[,2] )
acf(Fit3$residuals)


Fit4 <- VARMA(cbind(A1,B1), p=0, q=2, include.mean=FALSE)
Randomness.tests( Fit4$residuals[,1] )
Randomness.tests( Fit4$residuals[,2] )
acf(Fit4$residuals)

Fit4$coef
Fit4$secoef

Fit5 <- VARMA(cbind(A1,B1), p=0, q=3, include.mean=FALSE)
Randomness.tests( Fit5$residuals[,1] )
Randomness.tests( Fit5$residuals[,2] )
acf(Fit5$residuals)


#--- prediction with VARMA --
```

```
VARMApred(Fit3, h=1)

t(Fit3$coef) %*% Fit3$residuals[249,] #- same as pred
Fit3$residuals[249,] %*% Fit3$coef



#--- Rolling prediction with VARMA(0,1) --
A1 <- diff(log(A))
B1 <- diff(log(B))


A2.h <- 0
B2.h <- 0
for (i in 1:40) {

  print(i)

  A2 <- window(A1, start=2+i-1, end=202+i-1)
  B2 <- window(B1, start=2+i-1, end=202+i-1)

  Fit3 <- VARMA(cbind(A2,B2), p=0, q=1, include.mean=FALSE)

  A2.h[i] <- VARMApred(Fit3, h=1)$pred[1]
  B2.h[i] <- VARMApred(Fit3, h=1)$pred[2]

}
```

```r
A3 <- window(A1, start=203, end=242)
B3 <- window(B1, start=203, end=242)
A2.h <- ts(A2.h, start=203)
B2.h <- ts(B2.h, start=203)


layout(matrix(1:4, 4, 1))
plot(A3, ylim=c(-.02, .02) ); lines(A2.h, col="red")
plot(B3, ylim=c(-.02, .02) ); lines(B2.h, col="red")
plot(abs(A3-A2.h) , ylim=c(0, .02) )
plot(abs(B3-B2.h) , ylim=c(0, .02) )


Result1 <- cbind(A3>0, A2.h>0, (A3>0)==(A2.h>0) )
Result2 <- cbind(B3>0, B2.h>0, (B3>0)==(B2.h>0) )
colMeans(Result1)
colMeans(Result2)

MSE1 <- mean( (A3-A2.h)^2 )/ sd(A3) ; MSE1
MSE2 <- mean( (B3-B2.h)^2 )/ sd(B3) ; MSE2

layout(matrix(1:4, 4, 1))
plot( as.numeric(A3) * ((A2.h>0)*2-1) )
plot( as.numeric(B3) * ((B2.h>0)*2-1) )
plot( cumsum( as.numeric(A3) * ((A2.h>0)*2-1) ) )
```

```
plot( cumsum( as.numeric(B3) * ((B2.h>0)*2-1) ) )


.04/sum(abs(A3))
.14/sum(abs(B3))


#-- How about VARMA(p,0) --
Fit6 <- VARMA(cbind(A1,B1), p=1, q=0, include.mean=FALSE)
Randomness.tests( Fit6$residuals[,1] )
Randomness.tests( Fit6$residuals[,2] )
acf(Fit6$residuals)


Fit7 <- VARMA(cbind(A1,B1), p=2, q=0, include.mean=FALSE)
Randomness.tests( Fit7$residuals[,1] )
Randomness.tests( Fit7$residuals[,2] )
acf(Fit7$residuals)



#--- prediction with VARMA --
VARMApred(Fit6, h=1)

t(Fit6$coef) %*% cbind(A1,B1)[250,] #- same as pred
cbind(A1,B1)[250,] %*% Fit6$coef
```

```
#--- Rolling prediction with VARMA(1,0) --
A2.h <- 0
B2.h <- 0
for (i in 1:40) {

  print(i)

  A2 <- window(A1, start=2+i-1, end=202+i-1)
  B2 <- window(B1, start=2+i-1, end=202+i-1)

  Fit3 <- VARMA(cbind(A2,B2), p=1, q=0, include.mean=FALSE)

  A2.h[i] <- VARMApred(Fit3, h=1)$pred[1]
  B2.h[i] <- VARMApred(Fit3, h=1)$pred[2]

}

A3 <- window(A1, start=203, end=242)
A2.h <- ts(A2.h, start=203)

plot(A3)
lines(A2.h, col="red")

Result1 <- cbind(A3>0, A2.h>0, (A3>0)==(A2.h>0) )
```

```
colMeans(Result1)
```