제11강: 다변량시계열 분석 및 페어트레이딩

금융 통계 및 시계열 분석

TRADE INFORMATIX

2014년 2월 14일

목차

다변량 시계열 (Multivariate Time Series)

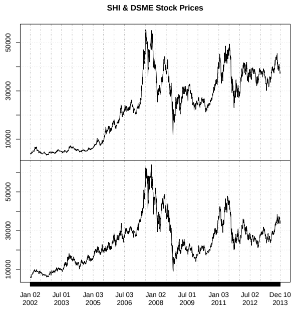
- ❏ 벡터값을 가지는 (vector-valued) 시계열

$$x_t = (x_{1,t}, x_{2,t}, \ldots, x_{k,t})$$

- ❏ 각 시계열간 상호연관성 존재
- ❏ 다변량 시계열 모형
  - ▶ VARMA (Vector ARMA)
- ❏ 다변량 시계열 문제
  - ▶ 의사 (擬似) 상관 (spurous correlation)
  - ▶ 공적분 (cointegration)
  - ▶ 페어 트레이딩 (pair trading)

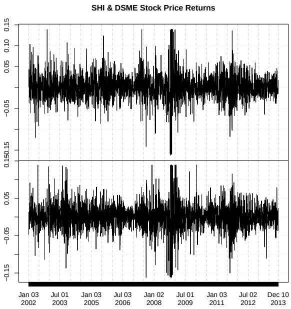
다변량 시계열의 예 1: 비정상 다변량 시계열

```
> require(rquantbook, quietly=TRUE)
> require(xts, quietly=TRUE)
> require(xtsExtra, quietly=TRUE)
> d1 <- get_quantbook_data("krx_stock_daily_price",
+   ticker="010140",
+   date_start="2002-01-01", date_end="2013-12-10")
> d2 <- get_quantbook_data("krx_stock_daily_price",
+   ticker="042660",
+   date_start="2002-01-01", date_end="2013-12-10")
> x1 <- xts(d1$close, order.by=as.POSIXct(d1$date))
> x2 <- xts(d2$close, order.by=as.POSIXct(d2$date))
> x <- merge(x1, x2)
> plot(x, main="SHI & DSME Stock Prices")
```



다변량 시계열의 예 2: 정상 다변량 시계열

```
> require(TTR, quietly=TRUE)
> y1 <- ROC(x1)
> y2 <- ROC(x2)
> x <- na.omit(merge(x1, x2))
> y <- na.omit(merge(y1, y2))
> plot(y, main="SHI & DSME Stock Price Returns")
```



정상 다변량 시계열

❑ 정상 다변량 시계열 : $k \times 1$ 벡터

$$x_t = (x_{1,t}, x_{2,t}, \ldots, x_{k,t})^T$$

❑ 평균 벡터 (mean vector) : $k \times 1$ 벡터

$$\mu = E[x_t]$$

❑ 공분산 행렬 (covariance matrix) : $k \times k$ 행렬

$$\Gamma(l) = E[(x_t - \mu)(x_{t-l} - \mu)^T]$$

❑ 분산대각행렬 : $k \times k$ 대각행렬

$$D = \text{diag}(\sqrt{\Gamma_{1,1}(0)}, \ldots, \sqrt{\Gamma_{k,k}(0)})$$

❑ 상호상관계수 행렬 (crosscorrelation matrix) : $k \times k$ 행렬

$$\rho(l) = D^{-1}\Gamma(l)D^{-1}$$

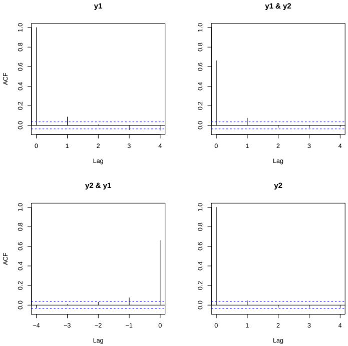
개별 원소로 보면

$$\rho(l)_{i,j} = \frac{\Gamma(l)_{i,j}}{\sqrt{\Gamma(0)_{i,i}}\sqrt{\Gamma(0)_{j,j}}}$$

상호상관계수 행렬의 예

❑ dse 패키지의 acf 명령 이용

```
> library("dse")
> (acf(coredata(y),lag.max=4))

Autocorrelations of series ‘coredata(y)’, by lag

, , y1

 y1            y2
  1.000 ( 0)   0.662 ( 0)
  0.087 ( 1)   0.076 (-1)
  0.008 ( 2)   0.030 (-2)
 -0.045 ( 3)   0.007 (-3)
 -0.051 ( 4)  -0.027 (-4)

, , y2

 y1            y2
  0.662 ( 0)   1.000 ( 0)
  0.075 ( 1)   0.046 ( 1)
 -0.022 ( 2)  -0.024 ( 2)
 -0.023 ( 3)  -0.026 ( 3)
 -0.016 ( 4)  -0.026 ( 4)
```



VAR 모형

- VAR (Vector Autoregressive) 모형의 예

$$\begin{aligned}\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} w_{1,t} \\ w_{2,t} \end{pmatrix} \\ x_t &= Ax_{t-1} + Bw_t \end{aligned}$$

- 정상조건
  - 행렬특성식 $A(x) = I - Ax - \cdots - A^p x^p = 0$ 의 determinant equation의 zero의 크기가 1보다 커야한다.

VAR 모형 시뮬레이션 명령

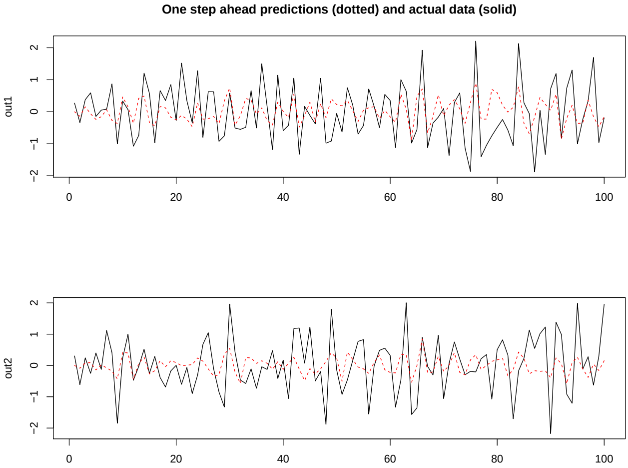
- dse 패키지 이용
- `ARMA(A, B)` : Vector ARMA 모형 생성
  - `A, B` : $A(L)y(t) = B(L)w(t)$ 형태의 모형 계수 어레이. 사이즈는 각각 $a \times p \times p$, $b \times p \times p$
- `simulate(model)` : 시뮬레이션
  - `model` : `ARMA` 명령으로 생성한 모형
- `l(model, simout)` : 시뮬레이션과 모형 결합
  - `simout` : `simulate` 명령으로 생성한 시뮬레이션 결과
- `tfplot(model)` or `tfplot(simout)` : 시뮬레이션/모형 플롯

VAR 모형 시뮬레이션 예 1

```
> require(dse)
> AR <- array(c(1, .5, .3, 0, .2, .1, 0, .2, .05, 1, .5, .3), c(3,2,2))
> MA <- array(c(1, .2, 0, .1, 0, 0, 1, .3), c(2,2,2))
> (arma <- ARMA(A=AR, B=MA, C=NULL))

A(L) =
1+0.5L1+0.3L2    0+0.2L1+0.05L2
0+0.2L1+0.1L2    1+0.5L1+0.3L2

B(L) =
1+0.2L1    0
0+0.1L1    1+0.3L1

> summary(arma)

ARMA:
inputs :
outputs:  out1 out2
     input  dimension =  0     output dimension =  2
     order A =  2     order B =  1     order C =
      11  actual parameters      4  non-zero constants
     trend not estimated.
```

VAR 모형 시뮬레이션 예 2

```
> simout <- simulate(arma)
> arma <- l(arma, simout)
> tfplot(arma)
```



VAR 모형 추정 - 차수 선택

❑ vars 패키지의 `VARselect(x, lag.max)` 명령 이용

```
> require(vars)
> VARselect(y, lag.max=10)

$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     6      5      1      6

$criteria
                   1             2             3
AIC(n) -1.442452e+01 -1.442665e+01 -1.443280e+01
HQ(n)  -1.442014e+01 -1.441934e+01 -1.442257e+01
SC(n)  -1.441235e+01 -1.440636e+01 -1.440439e+01
FPE(n)  5.438875e-07  5.427334e-07  5.394063e-07
                   4             5             6
AIC(n) -1.443464e+01 -1.444159e+01 -1.444201e+01
HQ(n)  -1.442149e+01 -1.442552e+01 -1.442301e+01
SC(n)  -1.439811e+01 -1.439695e+01 -1.438925e+01
FPE(n)  5.384144e-07  5.346841e-07  5.344611e-07
                   7             8             9
AIC(n) -1.444196e+01 -1.444163e+01 -1.444088e+01
HQ(n)  -1.442005e+01 -1.441679e+01 -1.441312e+01
SC(n)  -1.438109e+01 -1.437263e+01 -1.436377e+01
FPE(n)  5.344844e-07  5.346638e-07  5.350641e-07
                  10
AIC(n) -1.443917e+01
HQ(n)  -1.440849e+01
SC(n)  -1.435395e+01
FPE(n)  5.359775e-07
```

VAR 모형 추정 - 계수추정

❑ vars 패키지의 `VAR` 명령 이용

```
> (m1 <- VAR(y, p=6))

VAR Estimation Results:
=======================

Estimated coefficients for equation y1:
=======================================
Call:
y1 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + y1.l3 + y2.l3 + y1.l4 + y2.l4 + y1.l5 + y2.l5 + y1.l6 + y2.l6 + const

        y1.l1         y2.l1         y1.l2         y2.l2
 0.0585470168  0.0334482791  0.0273330196 -0.0326126536
        y1.l3         y2.l3         y1.l4         y2.l4
-0.0559204256  0.0184600117 -0.0642216257  0.0306664182
        y1.l5         y2.l5         y1.l6         y2.l6
-0.0700158461  0.0226418353  0.0229919189 -0.0040388758
        const
 0.0007691783


Estimated coefficients for equation y2:
=======================================
Call:
y2 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + y1.l3 + y2.l3 + y1.l4 + y2.l4 + y1.l5 + y2.l5 + y1.l6 + y2.l6 + const

        y1.l1         y2.l1         y1.l2         y2.l2
 0.1032610441 -0.0204457224  0.0978613652 -0.0887221099
        y1.l3         y2.l3         y1.l4         y2.l4
 0.0571463312 -0.0619233846 -0.0045065673 -0.0274443266
        y1.l5         y2.l5         y1.l6         y2.l6
 0.0417930919 -0.0357960328  0.0646815228 -0.0544276540
        const
 0.0004930344
```

VAR 모형 추정 - 계수추정 (계속)

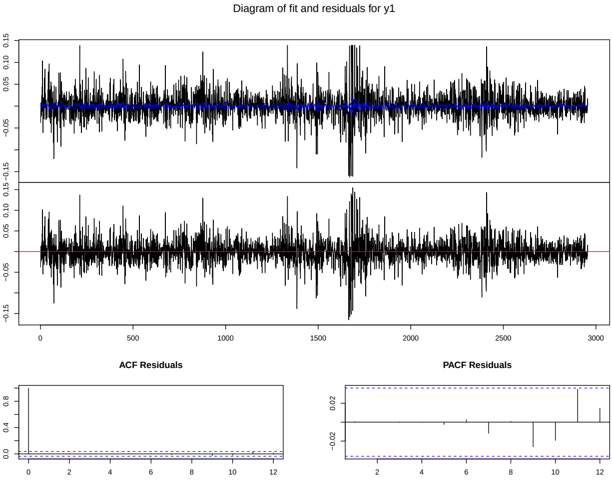
❑ vars 패키지의 `VAR` 명령 이용

```
> summary(m1, "y1")

VAR Estimation Results:
=========================
Endogenous variables: y1, y2
Deterministic variables: const
Sample size: 2956
Log Likelihood: 12983.689
Roots of the characteristic polynomial:
0.6905 0.6905 0.6004  0.58  0.58 0.5795 0.5795 0.5672 0.5672 0.5543 0.5543 0.3098
Call:
VAR(y = y, p = 6)


Estimation results for equation y1:
===================================
y1 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + y1.l3 + y2.l3 + y1.l4 + y2.l4 + y1.l5 + y2.l5 + y1.l6 + y2.l6 + const

        Estimate Std. Error t value Pr(>|t|)
y1.l1  0.0585470  0.0246880   2.371  0.01778 *
y2.l1  0.0334483  0.0215270   1.554  0.12034
y1.l2  0.0273330  0.0246258   1.110  0.26712
y2.l2 -0.0326127  0.0214832  -1.518  0.12911
y1.l3 -0.0559204  0.0246327  -2.270  0.02327 *
y2.l3  0.0184600  0.0215053   0.858  0.39075
y1.l4 -0.0642216  0.0246298  -2.607  0.00917 **
y2.l4  0.0306664  0.0214758   1.428  0.15341
y1.l5 -0.0700158  0.0246661  -2.839  0.00456 **
y2.l5  0.0226418  0.0214642   1.055  0.29157
y1.l6  0.0229919  0.0247163   0.930  0.35233
y2.l6 -0.0040389  0.0214501  -0.188  0.85066
const  0.0007692  0.0005376   1.431  0.15262
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.02917 on 2943 degrees of freedom
Multiple R-Squared: 0.01762, Adjusted R-squared: 0.01362
F-statistic: 4.399 on 12 and 2943 DF,  p-value: 5.235e-07
```

VAR 모형 추정 결과

❑ vars 패키지의 `plot` 명령 이용

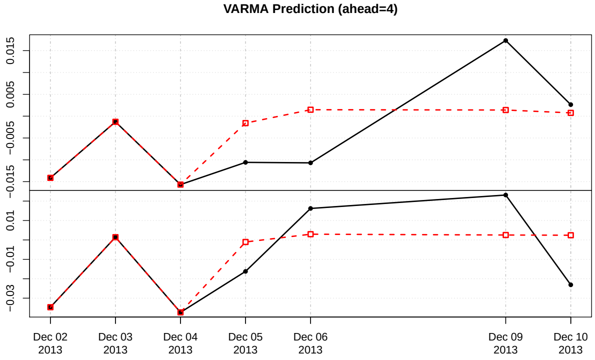
```
> plot(m1, "y1")
```



VAR 모형 예측

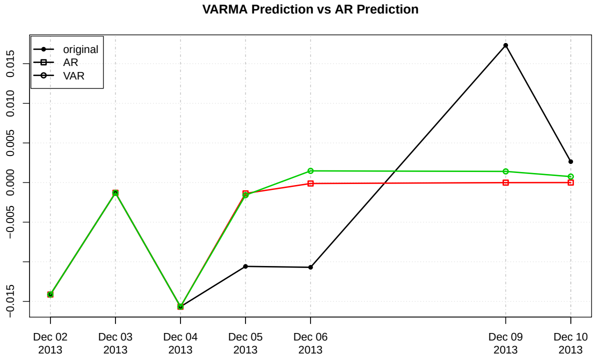
❑ vars 패키지의 `predict` 명령 이용

```
> Np <- 4; N <- dim(y)[1]; y2 <- y[-seq(N-(Np-1), N),]
> m2 <- VAR(y2, p=6); pred <- predict(m2, n.ahead=Np)
> y3 <- c(y2, xts(cbind(pred$fcst$y1[,1], pred$fcst$y2[,1]), order.by=index(y)[seq(N-(Np-1), N)]))
> plot(cbind(y, y3)["2013-12-01/"], screens=c(1,2), lwd=2, type='o', lty=c(1,1,2,2), pch=c(20,20,0,0),
+      main="VARMA Prediction (ahead=4)")
```



❑ AR 모형 결과와 비교

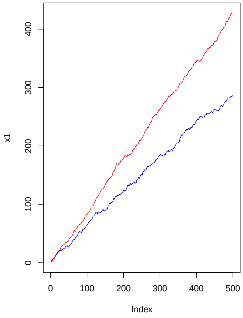
```
> require(forecast)
> Np <- 4; N <- dim(y1)[1]; y1.2 <- y1[-seq(N-(Np-1), N),]
> m.ar <- auto.arima(y1.2); pred <- predict(m.ar, n.ahead=Np)
> y1.3 <- c(y1.2, xts(pred$pred, order.by=index(y1)[seq(N-(Np-1), N)]))
> plot(cbind(y1, y1.3, y3$y1)["2013-12-01/"], screens=c(1), lwd=2, type='o', lty=c(1,1,1), pch=c(20, 0, 1),
+      main="VARMA Prediction vs AR Prediction")
> legend("topleft", c("original", "AR", "VAR"), lwd=2, pch=c(20, 0, 1))
```



spurious correlation

- 추세를 가지는 두 개의 비정상 시계열에서 상관계수가 크게 나타나는 현상
- 회귀분석 R^2 가 Durbin-Watson Statistics보다 크면 의심
  - lmtest 패키지의 `dwtest` 또는 car 패키지의 `durbinWatsonTest`

```
> require(lmtest, quietly=TRUE)
> x1 <- 0.8 * 1:500 + cumsum(rnorm(500))
> x2 <- 0.6 * 1:500 + cumsum(rnorm(500))
> cor(x1, x2)

[1] 0.9960331

> m <- lm(x1 ~ x2); summary(m); dwtest(m)

Call:
lm(formula = x1 ~ x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-30.483  -6.678   1.873   8.129  23.805 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 11.088082   0.979973   11.31   <2e-16 ***
x2           1.358176   0.005437  249.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.05 on 498 degrees of freedom
Multiple R-squared: 0.9921, Adjusted R-squared: 0.9921 
F-statistic: 6.24e+04 on 1 and 498 DF,  p-value: < 2.2e-16

Durbin-Watson test

data:  m
DW = 0.0233, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

> plot(x1, type='l', col='red'); lines(x2, col='blue')
```



공적분 (Cointegration)

- 두 integrated 비정상 신호의 선형조합 (linear combination) 이 integration 차수가 낮아지거나 정상상태가 되는 경우
- “술취한 사람과 개” 의 비유
- CI(d, b) 모형
  - multiple-valued time series $x_t = (x_{1,t}, x_{2,t}, \cdots, x_{n,t}$ 의 각 신호 x_i 는 I(d) 모형 신호이지만 선형조합을 하는 경우 I(d-b) 모형 신호가 되는 0-벡터가 아닌 선형조합이 존재하는 시계열
  - $d = b$ 인 경우 정상시계열이 됨
- cointegration vector
  - multiple-valued time series
- 공적분의 응용
  - 페어 트레이딩 (pair)
  - 통계적 차익거래 (statistical arbitrage)
  - 복수 주식의 포트폴리오 혹은 스프레드가 정상 상태가 되므로 과매도/과매수 상태를 이용한 contrarian 전략을 사용 가능

공적분 시계열의 예

- 비정상 ARMA(1,1) 모형

$$\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} = \begin{pmatrix} -0.5 & 1.0 \\ 0.25 & -0.5 \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} w_{1,t} \\ w_{2,t} \end{pmatrix} + \begin{pmatrix} 0.2 & -0.4 \\ -0.1 & 0.2 \end{pmatrix} \begin{pmatrix} w_{1,t-1} \\ w_{2,t-1} \end{pmatrix}$$

- 선형 변형

$$\begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix}$$

$$\begin{pmatrix} v_{1,t} \\ v_{2,t} \end{pmatrix} = \begin{pmatrix} 1 & -2 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} w_{1,t} \\ w_{2,t} \end{pmatrix}$$

- y_2 는 다음과 같은 정상 시계열이 된다.

$$y_{2,t} = v_{2,t} - 0.4 v_{1,t-1}$$

Error Correction Model

- 다음 VAR는 아래의 Vector Error Correction Model로 변형 가능

$$x_t = A_1 x_{t-1} + A_2 x_{t-2} + \cdots + A_p x_{t-p} + e_t$$

- long-run form VECM

$$\Delta x_t = \Phi x_{t-p} + \Gamma_1 \Delta x_{t-1} + \Gamma_2 \Delta x_{t-2} + \cdots + \Gamma_{p-1} \Delta x_{t-p+1} + e_t$$

$$\Phi = -(I - A_1 - \cdots A_p)$$

$$\Gamma_i = -(I - A_1 - \cdots A_i)$$

- transitory form VECM

$$\Delta x_t = \Phi x_{t-1} + \Gamma_1 \Delta x_{t-1} + \Gamma_2 \Delta x_{t-2} + \cdots + \Gamma_{p-1} \Delta x_{t-p+1} + e_t$$

$$\Phi = -(I - A_1 - \cdots A_p)$$

$$\Gamma_i = -(A_{i+1} + \cdots A_p)$$

- Vector Error Correction Model은 양변이 모두 stationary

Engle-Granger’ s Representation Theorem

❑ 두 시계열 $x_t \sim I(1)$, $y_t \sim I(1)$ 이 공적분이면 다음과 같은 ECM이 존재 (또는 그 반대)

$$\Delta y_t = \gamma_1 z_{t-1} + \sum_{i=1}^{K} \psi_{1,i} \Delta x_{t-i} + \sum_{i=1}^{L} \psi_{2,i} \Delta y_{t-i} + e_{1,t}$$

여기에서 정상신호 z_t 는 OLS $y_t \sim x_t$ 의 잔차항, 즉

$$y_t = A x_t + z_t$$

Engle and Granger 2-스텝 방법

1. 개별 시계열에 대해 unit root test로 I(1) 증명
2. 개별 시계열간 선형회귀분석

$$y_t = ax_t + z_t$$

3. unit root test로 선형회귀분석 잔차 z_t가 stationary임을 증명
4. ECM 모형으로 정상신호 OLS 회귀분석

$$\Delta y_t = \gamma z_{t-1} + \psi_1 \Delta x_{t-1} + \psi_2 \Delta y_{t-1} + e_t$$

5. ECM 회귀분석의 계수 γ의 부호가 0이 아님 $(\gamma \neq 0)$을 확인

Engle and Granger 2-스텝 예 1-1

```
> set.seed(123456)
> x <- cumsum(rnorm(100))
> y <- 0.6 * x + rnorm(100)
> (m1 <- lm(y ~ x))

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    0.03785      0.58112

> summary(m1)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-2.44465 -0.66254  0.03931  0.83827  2.04555

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03785    0.13477   0.281    0.779
x            0.58112    0.02138  27.186   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.985 on 98 degrees of freedom
Multiple R-squared: 0.8829, Adjusted R-squared: 0.8817
F-statistic: 739.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

Engle and Granger 2-스텝 예 1-2

```
> error <- residuals(m1)
> summary(ur.df(error))

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

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-2.44306 -0.60537  0.03453  0.72982  1.97666

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.98521    0.14865  -6.628 1.98e-09 ***
z.diff.lag -0.07874    0.10091  -0.780    0.437
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9796 on 96 degrees of freedom
Multiple R-squared: 0.5407, Adjusted R-squared: 0.5312
F-statistic: 56.51 on 2 and 96 DF,  p-value: < 2.2e-16


Value of test-statistic is: -6.6278

Critical values for test statistics:
     1pct  5pct 10pct
tau1 -2.6 -1.95 -1.61
```

Engle and Granger 2-스텝 예 1-3

```
> error.lagged <- error[-c(99, 100)]
> dx <- diff(x)
> dy <- diff(y)
> df <- data.frame(embed(cbind(dx, dy), 2))
> colnames(df) <- c('dx', 'dy', 'dx.1', 'dy.1')
> m2 <- lm(dy ~ error.lagged + dx.1 + dy.1, data=df)
> summary(m2)

Call:
lm(formula = dy ~ error.lagged + dx.1 + dy.1, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9588 -0.5439  0.1370  0.7114  2.3065 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.003398   0.103611   0.033    0.974    
error.lagged -0.968796   0.158554  -6.110 2.24e-08 ***
dx.1          0.808633   0.112042   7.217 1.35e-10 ***
dy.1         -1.058913   0.108375  -9.771 5.64e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.026 on 94 degrees of freedom
Multiple R-squared: 0.5464, Adjusted R-squared: 0.5319 
F-statistic: 37.74 on 3 and 94 DF,  p-value: 4.243e-16
```

Engle and Granger 2-스텝 예 2-1

```
> d1 <- get_quantbook_data("krx_stock_daily_price",
+   ticker="010140",
+   date_start="2002-01-01", date_end="2013-12-10")
> d2 <- get_quantbook_data("krx_stock_daily_price",
+   ticker="042660",
+   date_start="2002-01-01", date_end="2013-12-10")
> x1 <- xts(d1$close, order.by=as.POSIXct(d1$date))
> x2 <- xts(d2$close, order.by=as.POSIXct(d2$date))
> x <- na.omit(merge(x1, x2))
> p1 <- x[,1]
> p2 <- x[,2]
> (m1 <- lm(p2 ~ p1))

Call:
lm(formula = p2 ~ p1)

Coefficients:
(Intercept)           p1
  7737.8711       0.7109
```

Engle and Granger 2-스텝 예 2-2

```
> error <- residuals(m1)
> summary(ur.df(coredata(error)))

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

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-3773.2  -298.5    -9.3   293.7  5035.6

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.006084   0.002006  -3.033  0.00244 **
z.diff.lag  0.005418   0.018386   0.295  0.76827
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 642.1 on 2959 degrees of freedom
Multiple R-squared: 0.003105, Adjusted R-squared: 0.002431
F-statistic: 4.608 on 2 and 2959 DF,  p-value: 0.01005


Value of test-statistic is: -3.0328

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Engle and Granger 2-스텝 예 2-3

```
> N <- dim(p1)[1]
> error.lagged <- error[-c(N-1, N)]
> dx <- diff(coredata(p1))
> dy <- diff(coredata(p2))
> df <- data.frame(embed(cbind(dx, dy), 2))
> colnames(df) <- c('dx', 'dy', 'dx.1', 'dy.1')
> m2 <- lm(dy ~ error.lagged + dx.1 + dy.1, data=df)
> summary(m2)

Call:
lm(formula = dy ~ error.lagged + dx.1 + dy.1, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-7146.9  -376.1   -13.8   399.6  5946.5

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.312446  16.177064   0.514    0.607
error.lagged -0.006301   0.002750  -2.291    0.022 *
dx.1          0.074353   0.029240   2.543    0.011 *
dy.1          0.037792   0.025343   1.491    0.136
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 880.2 on 2957 degrees of freedom
Multiple R-squared: 0.01081, Adjusted R-squared: 0.009806
F-statistic: 10.77 on 3 and 2957 DF,  p-value: 4.879e-07
```

Johansen 방법

- Engle and Granger 2-스텝 방법의 단점
  - 두번의 연속 OLS로 오차 누적
  - 3개 이상의 시계열에 대해 개별적 관계를 구함
- Johansen 방법
  - 3개 이상의 시계열에 대해 가능한 공적분 벡터를 한번에 추정
  - ECM 모형의 Φ 행렬의 rank 조건 사용하여 한번에 공적분 테스트
  - H_0 : 최소 r 개의 공적분 벡터가 존재한다.

R에서 Johansen 방법 사용

- urca 패키지 이용
- `ca.jo(x)` : Johansen 공적분 테스트
  - `x` : 다변량 시계열 자료
- `cajorls(m, r)` : VECM 모형 추정
  - `m` : `ca.jo` 명령으로 생성된 결과
  - `r` : 공적분 rank

Johansen 방법의 예 1-1

```
> library(urca)
> set.seed(12345); e1 <- rnorm(250, 0, 0.5); e2 <- rnorm(250, 0, 0.5); e3 <- rnorm(250, 0, 0.5)
> u1.ar1 <- arima.sim(model=list(ar=0.75), innov=e1, n=250)
> u2.ar1 <- arima.sim(model=list(ar=0.30), innov=e2, n=250)
> y3 <- cumsum(e3); y1 <- 0.8 * y3 + u1.ar1; y2 <- -0.3 * y3 + u2.ar1
> vecm <- ca.jo(data.frame(y1,y2,y3))
> summary(vecm)

######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.27036254 0.15474942 0.01884032

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 2 |  4.72  6.50  8.18 11.65
r <= 1 | 41.69 12.91 14.90 19.19
r = 0  | 78.17 18.90 21.07 25.75

Eigenvectors, normalised to first column:
(These are the cointegration relations)

          y1.l2      y2.l2     y3.l2
y1.l2  1.000000  1.0000000 1.0000000
y2.l2 -4.732436  0.2273774 0.1513858
y3.l2 -2.129850 -0.6657324 2.3153224

Weights W:
(This is the loading matrix)

            y1.l2       y2.l2        y3.l2
y1.d -0.034235501 -0.29705774 -0.008294582
y2.d  0.145988517 -0.08137695  0.003335110
y3.d  0.002429191  0.01025326 -0.010523045
```

Johansen 방법의 예 1-2

```
> (vecm.r2 <- cajorls(vecm, r=2))
$rlm

Call:
lm(formula = substitute(form1), data = data.mat)

Coefficients:
          y1.d       y2.d       y3.d
ect1      -0.331293   0.064612   0.012682
ect2       0.094473  -0.709385  -0.009165
constant   0.168371  -0.027019   0.025255
y1.dl1    -0.227677   0.027012   0.068158
y2.dl1     0.144452  -0.715607   0.040487
y3.dl1     0.123467  -0.290828  -0.075251


$beta
            ect1      ect2
y1.l2  1.0000000 0.0000000
y2.l2  0.0000000 1.0000000
y3.l2 -0.7328534 0.2951962
```

Johansen 방법의 예 2

```
> x <- merge(x1, x2)
> p1 <- x["2009/2010",1]
> p2 <- x["2009/2010",2]
> summary(ca.jo(data.frame(p1,p2)))

######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.0138055195 0.0004489489

Values of teststatistic and critical values of test:

         test 10pct  5pct  1pct
r <= 1 | 0.23  6.50  8.18 11.65
r = 0  | 6.98 12.91 14.90 19.19

Eigenvectors, normalised to first column:
(These are the cointegration relations)

           x1.l2    x2.l2
x1.l2  1.0000000 1.000000
x2.l2 -0.9800749 0.279042

Weights W:
(This is the loading matrix)

             x1.l2        x2.l2
x1.d -0.0237984500 -0.002126786
x2.d  0.0002816635 -0.003035136
```

Johansen 방법의 예 3

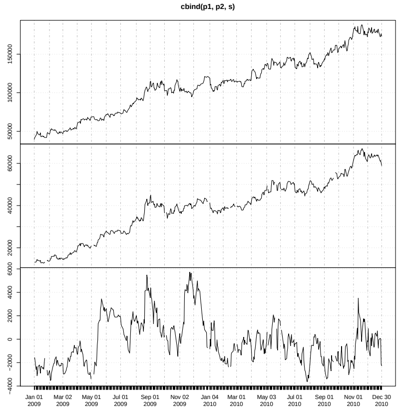
```
> library(urca)
> require("quantmod")
> d1 <- getSymbols("005380.KS", auto.assign=FALSE)
> d2 <- getSymbols("005385.KS", auto.assign=FALSE)
> x1 <- xts(d1$"005380.KS.Close")
> x2 <- xts(d2$"005385.KS.Close")
> x <- merge(x1, x2)
> p1 <- x["2009/2010",1]
> p2 <- x["2009/2010",2]
> summary(ca.jo(data.frame(p1,p2)))

######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.026675965 0.001876778

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 1 |  0.94  6.50  8.18 11.65
r = 0  | 13.57 12.91 14.90 19.19

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                    X005380.KS.Close.l2 X005385.KS.Close.l2
X005380.KS.Close.l2            1.000000            1.000000
X005385.KS.Close.l2           -2.870843           -1.185905

Weights W:
(This is the loading matrix)

                   X005380.KS.Close.l2 X005385.KS.Close.l2
X005380.KS.Close.d        -0.008540124        -0.005493401
X005385.KS.Close.d         0.014179567        -0.001574582
```

페어 트레이딩/통계적 차익거래

- 주가가 공적분 관계에 있는 복수 주식의 스프레드/포트폴리오 매매
- 스프레드/포트폴리오가 정상상태이므로 회귀특성 (mean-reverting)
- 과매도/과매수 상태에서 매수/매도하는 contrarian 매매

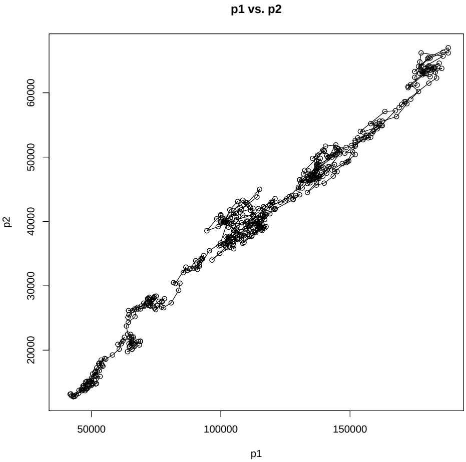
페어 트레이딩 종목의 예

```
> require("quantmod")
> d1 <- getSymbols("005380.KS",
+   auto.assign=FALSE)
> d2 <- getSymbols("005385.KS",
+   auto.assign=FALSE)
> x1 <- xts(d1$"005380.KS.Close")
> x2 <- xts(d2$"005385.KS.Close")
> x <- merge(x1, x2)
> p1 <- x["2009/2010",1]
> p2 <- x["2009/2010",2]
> m <- lm(p2 ~ p1,
+   na.action=na.omit)
> s <- (p2 - coef(m)[2] * p1)
> plot.xts(cbind(p1, p2, s))
```

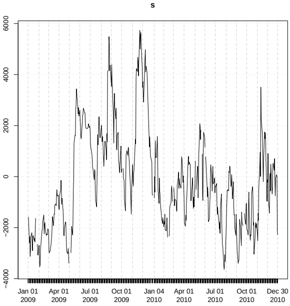


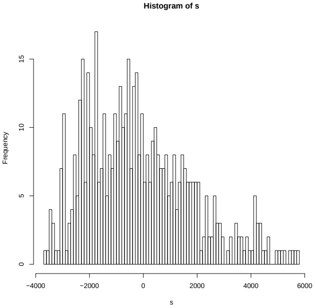
페어 트레이딩 분석의 예

```
> plot(p1, p2,
+      type='o')
```



페어 트레이딩 분석의 예


s
6000
4000
2000
0
−2000
−4000
Jan 01 2009
Apr 01 2009
Jul 01 2009
Oct 01 2009
Jan 04 2010
Apr 01 2010
Jul 01 2010
Oct 01 2010
Dec 30 2010


Histogram of s
Frequency
0
5
10
15
−4000
−2000
0
2000
4000
6000
s

페어 트레이딩 절차

1. 페어트레이딩 종목 및 비중 선정
   - ❏ 공적분 테스트 방법 이용
2. 스프레드 시계열 모형
   - ❏ AR 모형 / OU 모형 사용
3. 과매도/과매수 시점 결정
   - ❏ First Passage Time 분포 결정
4. 구조적 변화 (Structural Breakpoint) 모니터링
   - ❏ 지속적인 공적분 테스트
   - ❏ 스프레드 변화 요인 감시
   - ❏ 공적분 관계를 만들어내는 근본적 원인 감시

스프레드 시계열 모형

- Ornstein-Uhlenbeck 모형
  - 연속시간모형 (continuous time stochastic process)

$$dX_t = \rho(\theta - X_t)dt + \sigma dW_t$$

- ARMA 모형
  - 이산시간모형 (discrete time stochastic process)

$$x_t - x_{t-1} = \theta(1 - e^{-\rho}) + (e^{-\rho} - 1)x_t - 1 + e_t$$

$$e_t \sim N(0, \sigma_e^2) = N\left(0, \left(1 - e^{-2\rho}\frac{\sigma^2}{2\rho}\right)^2\right)$$

스프레드 시계열 모형 추정

❏ 선형 회귀

$$\Delta x_t \sim x_{t-1}$$

$$x_t - x_{t-1} = a + b x_{t-1} + z_t$$

❏ OU process 계수

$$\begin{aligned} \theta &= -a/b \\ \rho &= -\log(1+b) \\ \sigma_e &= \sigma_z \sqrt{\frac{\log(1+b)}{(1+b)^2 - 1}} \end{aligned}$$

매매 시점 결정

❑ First-Passage Time (Hitting Time)
  - ▶ 시계열값이 특정값에서 0로 돌아오는데 걸리는 시간

❑ First-Passage Time Distributions
  - ▶ Ornstein-Uhlenbeck 모형
    - 2007, Stefan Rampertshammer, An Ornstein-Uhlenbeck Framework for Pairs Trading
    - x : 매매를 시작하는 시점의 스프레드

$$F(t,x) = \frac{|x|}{\sqrt{2\pi}\sigma}\left(\frac{\rho}{\sinh(\rho t)}\right)^{\frac{2}{3}} \exp\left\{-\frac{\rho x^2 e^{-\rho t}}{2\sigma^2 \sinh(\rho t)} + \frac{\rho t}{2}\right\}$$

❑ 매매 시점의 결정
  - ▶ x가 커지면 수익은 증가하나 매매횟수가 줄고 구조적 변화 리스크 증가
  - ▶ 수치 최적화 필요

페어 트레이딩의 어려움

- ❏ short 물량 확보
- ❏ uptick rule에 따른 매매 시점 확보
- ❏ short market impact
- ❏ spread 비용
- ❏ 구조적 변화