제9강: 시계열분석 1

금융 통계 및 시계열 분석

TRADE INFORMATIX

2014년 2월 7일

목차

시계열 분석이란

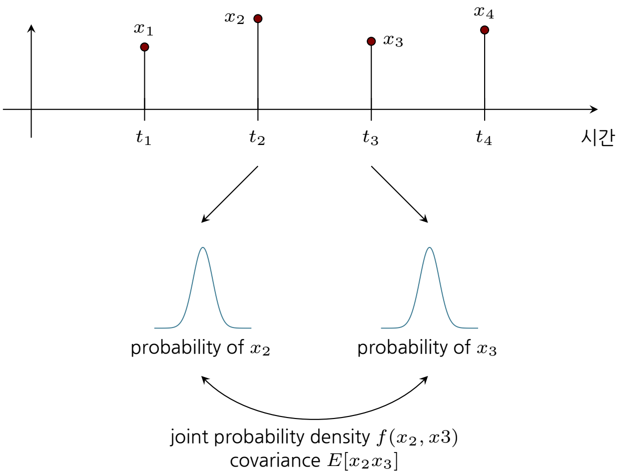
정적 통계 분석

- ❑ 시간 순서에 따른 일련의 확률변수 값이 서로 독립적
- ❑ independent and identical distribution

시계열 분석

- ❑ 시간 순서에 따른 일련의 확률변수 값이 서로 상관관계
- ❑ non-zero auto-covariance

시계열 분석이란


x_1
x_2
x_3
x_4
t_1
t_2
t_3
t_4
시간
probability of x_2
probability of x_3
joint probability density $f(x_2, x3)$
covariance $E[x_2 x_3]$

시계열의 분류

- 비정상 신호
  (Non-stationary signals)
  - 자산 가격 (asset price)
  - GDP
  - 투자자별 누적 순매수 포지션

- 정상 신호
  (Stationary signals)
  - 자산 수익률 (asset return)
  - GDP 증가율
  - 시간당 투자자별 순매수 포지션 변화

정상특성 (Stationarity)

정상 특성

- ❏ 평균 크기와 상관계수가 변하지 않으면 정상 (stationary)
- ❏ 평균 크기가 지속적으로 증가하면 비정상 (non-stationary)

시계열 신호 $\{x_i\}$에 대해,

- ❏ 강한 정상 특성 (Strictly Stationary)
  - ► 모든 양의 정수 i, l, k에 대해,
    $(x_i, x_{i+1}, \cdots, x_{i+l})$의 결합확률분포 (joint distribution)가
    $(x_{i+l}, x_{i+1+l}, \cdots, x_{i+l+k})$의 결합확률분포와 같은 경우
- ❏ 약한 정상 특성 (Weakly Stationary)
  - ► 모든 양의 정수 i, k에 대해,
    expectation $E[x_i]$가 변하지 않는 상수 (constant) 이고
    lag-k auto-covariance $Cov[x_i, x_{i-k}]$가 i가 아닌 k값에만 의존하는 경우

$$E[x_i] = \mu \tag{1}$$

$$Cov[x_i, x_{i-k}] = E[(x_i - \mu)(x_{i-k} - \mu)] = Cov_k \tag{2}$$

자기상관계수 (Autocorrelation)

❑ 자기상관계수 : covariance를 variance로 정규화 (normalization)

$$\rho_k = \frac{E[(x_i - \mu)(x_{i-k} - \mu)]}{E[(x_i - \mu)^2]} = \frac{Cov_k}{\sigma^2} \tag{3}$$

❑ 자기상관계수의 크기는 0과 1사이

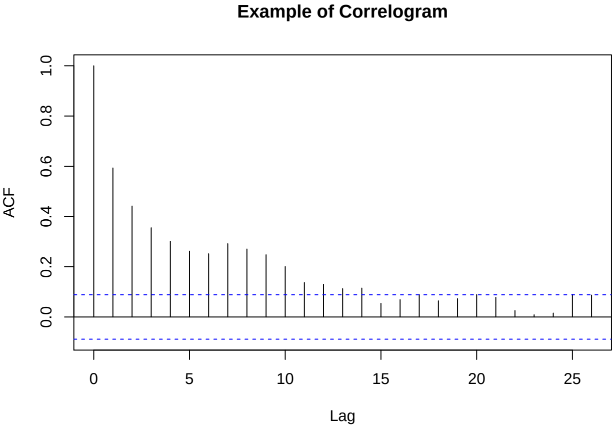
$$0 \leq \rho_k \leq 1 \tag{4}$$

❑ 샘플 자기상관계수 : 샘플 자료에서 구한 자기상관계수의 추정치

$$r_k = \frac{1}{n} \frac{\sum_{i=k+1}^{n} (x_i - \mu)(x_{i-k} - \mu)}{\sum_{i=1}^{n} (x_i - \mu)^2} \tag{5}$$

Correlogram

❑ Correlogram : lag-k에 대한 autocorrelation 값을 함수로 나타낸 그림



acf/pacf command

- autocorrelation, autocovariance, parical correlation 계산
- correlogram 그리기
- `acf(x, lag.max, type, plot=TRUE)`
  - `x` : 시계열 자료
  - `lag.max` : 계산할 최대 lag 수
  - `type` : 계산 유형. `"correlation"`, `"covariance"`, `"partial"`
  - `plot` : correlogram 그리기 여부
- `pacf(x, lag.max, plot=TRUE)` : partial acf 전용 명령

```
> require(rquantbook)
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01", date_end="2013-12-31")
> head(d, 4)

        date individual foreigner institute securities insurance  fund bank  nps government others
1 2012-01-03      -9710      3026      6776        122      1292   496  301   -7       4003    -92
2 2012-01-04      -3921      2881      1100        796       490  -109  162 -350        105    -60
3 2012-01-05      -2308      -416      2799        798       856   483  317  -41        176    -76
4 2012-01-06       3039      -483     -2691        556       -77 -1199  -54 -110      -1993    136

> a <- acf(d$foreigner)
> str(a)

List of 6
 $ acf    : num [1:27, 1, 1] 1 0.593 0.442 0.355 0.302 ...
 $ type   : chr "correlation"
 $ n.used: int 493
 $ lag    : num [1:27, 1, 1] 0 1 2 3 4 5 6 7 8 9 ...
 $ series: chr "d$foreigner"
 $ snames: NULL
 - attr(*, "class")= chr "acf"
```

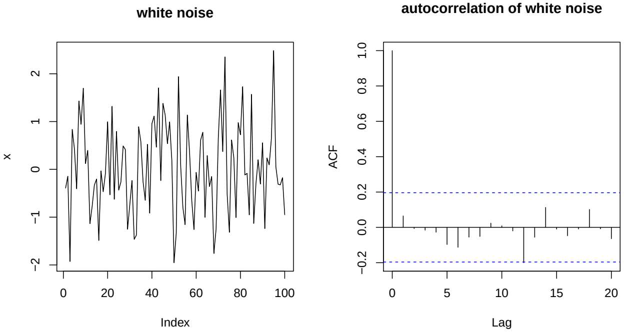
선형모형 (Linear Models)

- 백색잡음 (White Noise)
  - Normal 분포. 신호간 독립 (Independent and Indentical Distribution)
  - autocorrelation은 lag-0에서 1. 이외는 모두 0
- MA 모형 (Moving-Average model)
  - white noise의 선형조합
  - autocorrelation support가 유한
- AR 모형 (Auto-Regressive model)
  - 과거의 일부 신호값과 white noise의 선형조합
  - autocorrelation support가 무한
- ARMA 모형 (Auto-Regressive Moving-Average model)
  - 신호값이 과거의 값과 white noise의 선형조합

백색잡음 (White Noise)

- ❑ 선형모형의 입력값
- ❑ 시뮬레이션
  - ▶ **rnorm** 명령으로 생성가능

```
> x <- rnorm(100)
> par(mfrow=c(1,2))
> plot(x, type='l', main="white noise")
> acf(x, main="autocorrelation of white noise")
```



AR 모형

- innovationw_t 과 신호 자체의 과거값의 선형 조합

$$x_t = a_1 x_{t-1} + \cdots + a_p x_{t-p} + w_t \tag{6}$$

  - w_t : white noise (평균 :0 , 분산 σ^2)
  - a_i : 모형 계수

- a_i 가 다음 조건을 만족할 때 신호가 정상 (stationary)

$$\text{특성식: } 1 - a_1 x - a_2 x^2 - \cdots - a_p x^p = 0 \tag{7}$$

  의 해의 크기 (modulus) 가 1 보다 커야 한다.

AR(1) 모형

- innovation w_t 과 한 단계 과거값만의 선형 조합
- 평균 : $E[x] = 0$

$$\begin{aligned} E[x_t] &= a_1 E[x_{t-1}] + E[w_t] \\ \mu &= a_1 \mu + 0 \\ \therefore \mu &= 0 \end{aligned}$$

- 분산 : $\text{Var}[x] = \sigma^2/(1-a_1^2)$

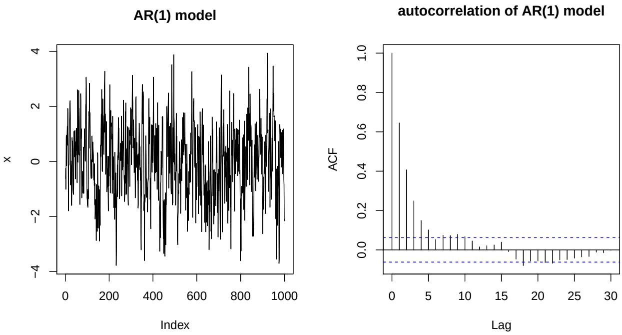
$$\begin{aligned} \text{Var}[x_t] &= E[(a_1 x_{t-1} + w_t)^2] \\ &= a_1^2 E[x_{t-1}^2] + E[x + t - 1 w_t] + E[w_t^2] \\ &= a_1^2 \text{Var}[x_t] + \sigma^2 \end{aligned}$$

- autocovariance : $\text{Cov}[x_t x_{t-k}] = a_1^k \sigma^2/(1-a_1^2)$
- autocorrelation : $\rho(k) = a_1^k$

AR 모형의 시뮬레이션 1

❑ 모형 공식을 이용한 시뮬레이션

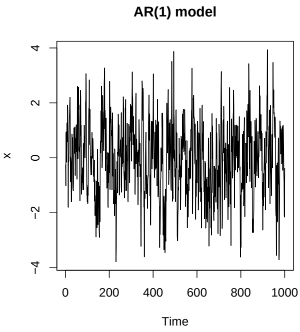
```
> set.seed(1)
> x <- w <- rnorm(1000)
> for (t in 2:1000) x[t] <- 0.7 * x[t-1] + w[t]
> par(mfrow=c(1,2))
> plot(x, type='l', main="AR(1) model")
> acf(x, main="autocorrelation of AR(1) model")
```

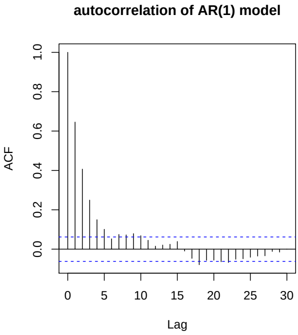


AR 모형의 시뮬레이션 2

- `arima.sim(model=list(ar=coeff), n)`
  - `coeff` : AR 모형 계수 벡터
  - `n` : 시뮬레이션 갯수

```
> set.seed(1)
> x <- arima.sim(model=list(ar=c(0.7)), 1000, n.start=1)
> par(mfrow=c(1,2))
> plot(x, type='l', main="AR(1) model")
> acf(x, main="autocorrelation of AR(1) model")
```





Partial Autocorrelation

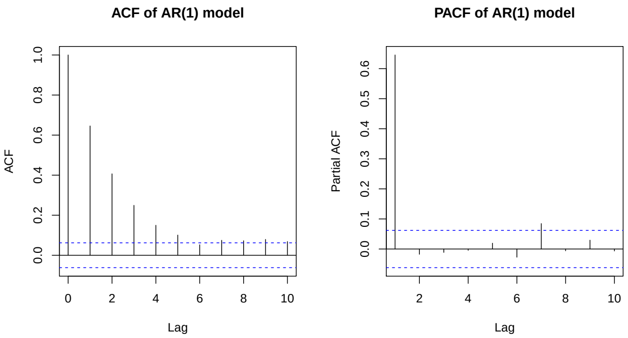
- p-th order AR 모형 회귀분석으로 구한 모형 계수 $a_{i,i}$

$$
\begin{aligned}
\text{(for } i=1) \quad x_t &= a_{1,1}x_{t-1} + e_t \\
\text{(for } i=2) \quad x_t &= a_{1,2}x_{t-1} + a_{2,2}x_{t-2} + e_t \\
\text{(for } i=3) \quad x_t &= a_{1,3}x_{t-1} + a_{2,3}x_{t-2} + a_{3,3}x_{t-3} + e_t \\
\text{(for } i=4) \quad x_t &= a_{1,4}x_{t-1} + a_{2,4}x_{t-2} + a_{3,4}x_{t-3} + a_{4,4}x_{t-3} + e_t
\end{aligned}
$$

- p-th order AR 모형 신호에서 구한 $a_{i,i}$는 다음 특성을 가진다.
  - 샘플 크기가 커지면 $a_{p,p}$는 a_p로 수렴
  - p보다큰 i에 대해 $a_{i,i} = 0$

AR 모형 PACF 결과

```
> set.seed(1)
> x <- arima.sim(model=list(ar=c(0.7)), 1000, n.start=1)
> par(mfrow=c(1,2))
> acf(x, lag.max=10, main="ACF of AR(1) model")
> acf(x, lag.max=10, "partial", main="PACF of AR(1) model")
```



AR모형 계수추정

- `ar(x, aic, order.max, method)`
  - `x` : 입력 신호
  - `aic` : AIC 기준을 이용한 order 결정
  - `order.max` : 가능한 최대 order
  - `method` : 추정 방법 (`"yule-walker", "burg", "ols", "mle", "yw"`)
- 출력 : list
  - `order` : 결정된 order
  - `ar` : 추정된 계수
  - `method` : 사용된 추정방법
  - `parialacf` : 추정된 PACF
  - `resid` : error residuals

AR모형 계수추정의 예 1

```
> set.seed(1)
> x <- arima.sim(model=list(ar=c(0.7)), 1000, n.start=1)
> (m <- ar(x))

Call:
ar(x = x)

Coefficients:
     1
0.6456

Order selected 1  sigma^2 estimated as  1.068

> str(m)

List of 14
 $ order       : int 1
 $ ar          : num 0.646
 $ var.pred    : num 1.07
 $ x.mean      : num -0.0335
 $ aic         : Named num [1:31] 537.17 0 1.69 3.57 5.55 ...
  ..- attr(*, "names")= chr [1:31] "0" "1" "2" "3" ...
 $ n.used      : int 1000
 $ order.max   : num 30
 $ partialacf  : num [1:30, 1, 1] 0.64557 -0.01749 -0.0113 -0.00444 0.01943 ...
 $ resid       : Time-Series [1:1000] from 1 to 1000: NA -0.838 1.552 0.39 -0.757 ...
 $ method      : chr "Yule-Walker"
 $ series      : chr "x"
 $ frequency   : num 1
 $ call        : language ar(x = x)
 $ asy.var.coef: num [1, 1] 0.000584
 - attr(*, "class")= chr "ar"
```

AR모형 계수추정의 예 2

```
> require(rquantbook)
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01", date_end="2013-12-31")
> (m <- ar(d$foreigner))

Call:
ar(x = d$foreigner)

Coefficients:
      1        2        3        4        5        6        7
 0.4839   0.0898   0.0359   0.0250  -0.0025  -0.0006   0.1281

Order selected 7  sigma^2 estimated as  4811024

> str(m)

List of 14
 $ order       : int 7
 $ ar          : num [1:7] 0.48389 0.08976 0.03592 0.02503 -0.00245 ...
 $ var.pred    : num 4811024
 $ x.mean      : num 405
 $ aic         : Named num [1:27] 224.5 12.68 5.12 4.4 4.86 ...
  ..- attr(*, "names")= chr [1:27] "0" "1" "2" "3" ...
 $ n.used      : int 493
 $ order.max   : num 26
 $ partialacf  : num [1:26, 1, 1] 0.5932 0.1386 0.0742 0.0558 0.04 ...
 $ resid       : num [1:493] NA NA NA NA NA ...
 $ method      : chr "Yule-Walker"
 $ series      : chr "d$foreigner"
 $ frequency   : num 1
 $ call        : language ar(x = d$foreigner)
 $ asy.var.coef: num [1:7, 1:7] 2.03e-03 -9.98e-04 -1.84e-04 -8.07e-05 -6.11e-05 ...
 - attr(*, "class")= chr "ar"
```

MA 모형

- innovation w_t 의 과거값의 선형조합

$$x_t = w_t + b_1 w_{t-1} + \cdots + b_q w_{t-q} = \sum_{i=0}^{q} b_i w_{t-i} \tag{8}$$

  - w_t : white noise (평균:0 , 분산 σ^2)
  - b_k : 모형 계수. $b_0 = 1$

- MA모형은 AR모형과 달리 계수값에 관계없이 항상 정상(stationary)

MA 모형 특성

- 평균 : $E[x] = 0$

$$E[x_t] = E[w_t] + b_1 E[w_{t-1}] + \cdots + b_q E[w_{t-q}] = 0$$

- 분산 : $\text{Var}[x] = \sigma^2 (1 + \sum_{i=1}^{q} b_i^2) = \sigma^2 \sum_{i=0}^{q} b_i^2)$

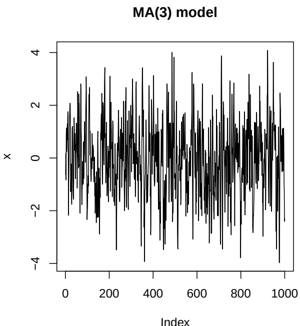
$$\begin{aligned} \text{Var}[x_t] &= E[(w_t + b_1 w_{t-1} + \cdots + b_q w_{t-q})^2] \\ &= a_1^2 E[w_t^2] + b_1^2 E[w_{t-1}^2] + \cdots + b_q^2 E[w_{t-q}^2] \\ &= \sigma^2 (1 + \sum_{i=1}^{q} b_i^2) \end{aligned}$$

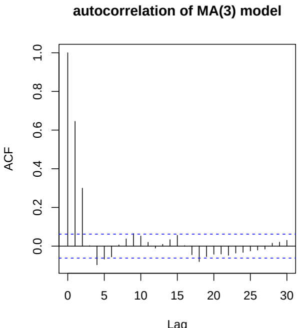
- autocorrelation

$$\rho(k) = \begin{cases} 1 & \text{if } k = 0 \\ \dfrac{\sum_{i=0}^{q-k} b_i b_{i+k}}{\sum_{i=0}^{q} b_i^2} & \text{if } k = 0 \\ 0 & \text{if } x > q \end{cases}$$

❑ 모형 공식을 이용한 시뮬레이션

```
> set.seed(1)
> b <- c(0.8, 0.6, 0.2)
> x <- w <- rnorm(1000)
> for (t in 4:1000) {
+   for (j in 1:3) x[t] <- x[t] + b[j] * w[t-j]
+ }
> par(mfrow=c(1,2))
> plot(x, type='l', main="MA(3) model")
> acf(x, main="autocorrelation of MA(3) model")
```

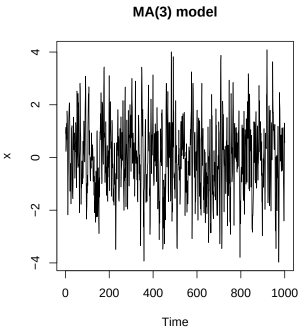


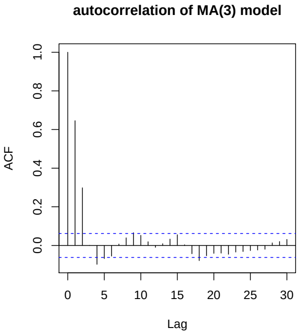


MA모형의 시뮬레이션 2

- arima.sim(model=list(ma=coeff), n)
  - coeff : MA모형 계수 벡터
  - n : 시뮬레이션 갯수

```
> set.seed(1)
> x <- arima.sim(model=list(ma=c(0.8, 0.6, 0.2)), 1000)
> par(mfrow=c(1,2))
> plot(x, type='l', main="MA(3) model")
> acf(x, main="autocorrelation of MA(3) model")
```





ARMA모형

- innovation w_t 의 과거값과 신호 자체의 과거값의 선형 조합

$$x_t = \sum_{i=1}^{p} a_i x_{t-i} + \sum_{i=0}^{q} b_i w_{t-i} \tag{9}$$

  - w_t : white noise (평균 :0 , 분산 σ^2)
  - a_i, b_i : 모형 계수. $b_0 = 1$

- ARMA모형은 AR모형처럼 계수가 다음 조건을 만족해야 정상 (stationary)

$$\text{특성식:} \quad \sum_{i=1}^{p} a_i x^i = 1 \tag{10}$$

의 해의 크기 (modulus) 가 1보다 커야 한다.

ARMA(1,1) 모형

- ❑ 신호자체와 innovation의 과거의 값 1개의 선형조합

$$x_t = ax_{t-1} + w_t + bw_{t-1}$$

- ❑ Backward operator B : $B^j x_i = x_{i-j}$

$$\begin{aligned} x_t &= (1 - aB)^{-1}(1 + bB)w_t \\ &= \left(\sum_{i=0}^{\infty} a^i B^i\right)(1 + bB)w_t \\ &= w_t + (a + b)\sum_{i=0}^{\infty} a^{i-1} w_{t-i} \end{aligned}$$

- ❑ 분산

$$\begin{aligned} \text{Var}[x_t] &= \text{Var}\left[w_t + (a + b)\sum_{i=0}^{\infty} a^{i-1} w_{t-i}\right] \\ &= \sigma^2(1 + (a + b)^2(1 - a^2)^{-1}) \end{aligned}$$

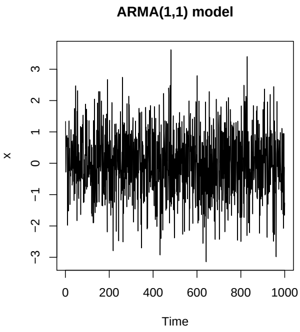
- ❑ autocorrelation

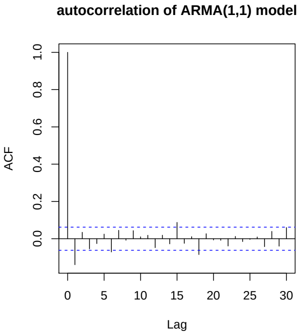
$$\rho(k) = \frac{a^{k-1}(a + b)(1 + ab)}{1 + ab + b^2}$$

ARMA모형의 시뮬레이션

- ❑ `arima.sim(model=list(ar=ar,ma=ma), n)`
  - ▶ `ar, ma` : ARMA모형 계수 벡터
  - ▶ `n` : 시뮬레이션 갯수

```
> set.seed(1)
> x <- arima.sim(model=list(ar=-0.6,ma=0.5), 1000)
> par(mfrow=c(1,2))
> plot(x, type='l', main="ARMA(1,1) model")
> acf(x, main="autocorrelation of ARMA(1,1) model")
```





ARMA모형 계수추정

- `arima` 명령으로 AR모형, MA모형, ARMA모형 모두 추정가능
- `arima(x, order, method)`
  - `x` : 입력 신호
  - `order` : 모형 order
    - AR 계수 order p, Integration 계수 order, MA 계수 order q
    - AR(p) 모형 : `order=c(p, 0, 0)`
    - MA(q) 모형 : `order=c(0, 0, q)`
    - ARMA(p,q) 모형 : `order=c(p, 0, q)`
  - `method` : 추정 방법 (`("CSS-ML", "ML", "CSS")`)
- 출력 : list
  - `coef` : 추정된 계수
  - `arma` : 추정된 계수 (compact form)
  - `residuals` : error residuals

ARMA모형 계수추정의 예 1

```
> set.seed(1)
> x <- arima.sim(model=list(ar=-0.6,ma=0.5), 5000)
> m <- arima(x, c(1,0,1), include.mean=FALSE)
> str(m)

List of 13
 $ coef     : Named num [1:2] -0.528 0.431
  ..- attr(*, "names")= chr [1:2] "ar1" "ma1"
 $ sigma2   : num 1.05
 $ var.coef : num [1:2, 1:2] 0.00609 -0.00637 -0.00637 0.00682
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "ar1" "ma1"
  .. ..$ : chr [1:2] "ar1" "ma1"
 $ mask     : logi [1:2] TRUE TRUE
 $ loglik   : num -7225
 $ aic      : num 14456
 $ arma     : int [1:7] 1 1 0 0 1 0 0
 $ residuals: Time-Series [1:5000] from 1 to 5000: 1.3262 -0.1459 0.0444 0.9071 0.8407 ...
 $ call     : language arima(x = x, order = c(1, 0, 1), include.mean = FALSE)
 $ series   : chr "x"
 $ code     : int 0
 $ n.cond   : int 0
 $ model    :List of 10
  ..$ phi  : num -0.528
  ..$ theta: num 0.431
  ..$ Delta: num(0)
  ..$ Z    : num [1:2] 1 0
  ..$ a    : num [1:2] 0.1103 0.0274
  ..$ P    : num [1:2, 1:2] 0 0 0 0
  ..$ T    : num [1:2, 1:2] -0.528 0 1 0
  ..$ V    : num [1:2, 1:2] 1 0.431 0.431 0.186
  ..$ h    : num 0
  ..$ Pn   : num [1:2, 1:2] 1 0.431 0.431 0.186
 - attr(*, "class")= chr "Arima"
```

ARMA모형모형 계수추정의 예 2

```
> require(rquantbook)
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01", date_end="2013-12-31")
> m <- arima(d$foreigner, c(1,0,1))
> str(m)

List of 13
 $ coef     : Named num [1:3] 0.778 -0.298 420.177
  ..- attr(*, "names")= chr [1:3] "ar1" "ma1" "intercept"
 $ sigma2   : num 4845625
 $ var.coef : num [1:3, 1:3] 0.00253 -0.0034 0.15422 -0.0034 0.00664 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "ar1" "ma1" "intercept"
  .. ..$ : chr [1:3] "ar1" "ma1" "intercept"
 $ mask     : logi [1:3] TRUE TRUE TRUE
 $ loglik   : num -4494
 $ aic      : num 8997
 $ arma     : int [1:7] 1 1 0 0 1 0 0
 $ residuals: Time-Series [1:493] from 1 to 493: 2072 911 -2479 -991 -770 ...
 $ call     : language arima(x = d$foreigner, order = c(1, 0, 1))
 $ series   : chr "d$foreigner"
 $ code     : int 0
 $ n.cond   : int 0
 $ model    :List of 10
  ..$ phi  : num 0.778
  ..$ theta: num -0.298
  ..$ Delta: num(0)
  ..$ Z    : num [1:2] 1 0
  ..$ a    : num [1:2] 800 -235
  ..$ P    : num [1:2, 1:2] 0 0 0 0
  ..$ T    : num [1:2, 1:2] 0.778 0 1 0
  ..$ V    : num [1:2, 1:2] 1 -0.298 -0.298 0.089
  ..$ h    : num 0
  ..$ Pn   : num [1:2, 1:2] 1 -0.298 -0.298 0.089
 - attr(*, "class")= chr "Arima"
```

추정 order 결정

- AIC (Akaike Information Criterion)

$$\text{AIC} = -\frac{2}{T}\log(\text{maximum ikelihood}) + \frac{2}{T}(\text{number of parameters})$$

- BIC (Bayesian Information Criterion)

$$\text{BIC(L)} = -\frac{2}{T}\log(\text{maximum ikelihood}) + \frac{2L\log T}{T}$$

- 선정 기준 : AIC 혹은 BIC 가 최소가 되는 order 선택
- `AIC(model)`, `BIC(model)` : AIC/BIC 계산 명령어
  - `model` : `arima` 명령으로 구한 추정결과

추정 order 결정 예

```
> require(rquantbook)
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01", date_end="2013-12-31")
> x <- d$foreigner
> best.aic <- Inf
> for (i in 0:2) for (j in 0:2) {
+   fit.aic <- AIC(arima(x, order=c(i,0,j)))
+   if (fit.aic < best.aic) {
+     best.order <- c(i,0,j)
+     best.arima <- arima(x, best.order)
+     best.aic <- fit.aic
+   }
+ }
> best.order

[1] 2 0 1

> best.arima

Call:
arima(x = x, order = best.order)

Coefficients:
         ar1      ar2      ma1  intercept
      1.2612  -0.3150  -0.7713   431.8388
s.e.  0.1090   0.0841   0.0918   411.6786

sigma^2 estimated as 4785324:  log likelihood = -4491.25,  aic = 8992.5
```

AR(p) 모형 예측

- 1-step 예측

$$
\begin{aligned}
E[x_{t+1}|x_t] &= E\left[\sum_{i=1}^{p} a_i x_{t-i+1} + w_{t+1}|x_t\right] \\
&= \sum_{i=1}^{p} a_i x_{t-i+1}
\end{aligned}
$$

- 2-step 예측

$$
\begin{aligned}
E[x_{t+2}|x_t] &= E\left[\sum_{i=1}^{p} a_i x_{t-i+2} + w_{t+2}|x_t\right] \\
&= a_1 E[x_{t+1}|x_t] + \sum_{i=2}^{p} a_i x_{t-i+2}
\end{aligned}
$$

- 예측 step이 증가하면 예측치는 0으로 수렴

$$E[x_{t+N}|x_t] \to 0 \text{ for } N \to \infty$$

MA(q) 모형 예측

❏ 1-step 예측

$$\begin{aligned} E[x_{t+1}|x_t] &= E\left[\sum_{i=0}^{q} b_i w_{t-i+1}|x_t\right] \\ &= b_1 w_t \end{aligned}$$

❏ 2-step 예측

$$\begin{aligned} E[x_{t+2}|x_t] &= E\left[\sum_{i=0}^{q} b_i w_{t-i+2}|x_t\right] \\ &= b_2 w_t \end{aligned}$$

❏ q order 모형의 경우 q-step 이후는 예측 불가

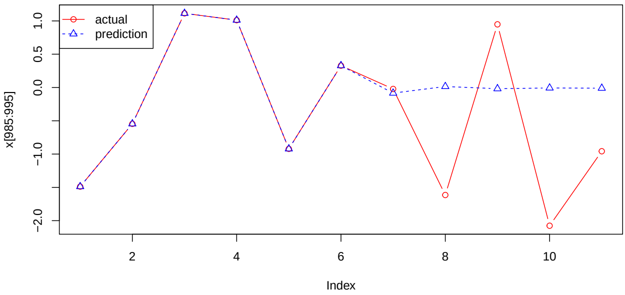
$$E[x_{t+N}|x_t] = 0 \text{ for } N > q$$

ARMA모형 예측

- `predict(model, n.ahead)`
  - `model` : `arima` 명령으로 구한 추정결과
  - `n.ahead` : 모형 order
- 출력 : list
  - `pred` : 예측치
  - `se` : standardized error

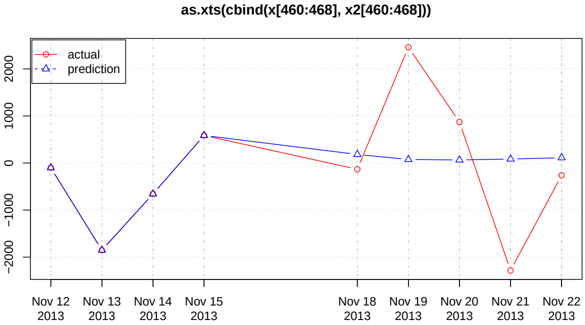
ARMA모형 예측 예 1

```
> set.seed(1)
> x <- arima.sim(model=list(ar=-0.6,ma=0.5), 1000)
> m <- arima(x[1:990], c(1,0,1))
> p <- predict(m, 10)
> x2 <- c(x[1:990], p$pred)
> plot(x[985:995], type='b', col="red", pch=1)
> lines(x2[985:995], type='b', col="blue", lty=2, pch=2)
> legend("topleft", c("actual", "prediction"), col=c("red", "blue"),
+        lty=1:2, pch=1:2)
```



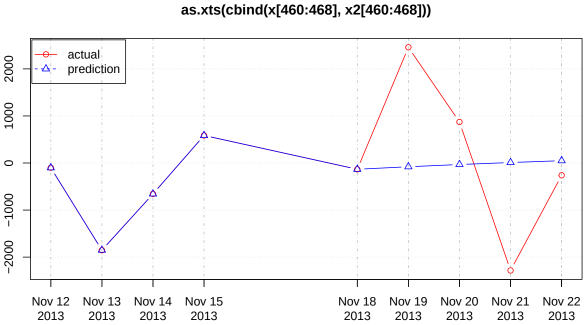
ARMA모형 예측 예 2

```
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01",
+                         date_end="2013-11-23")
> x <- d$foreigner; names(x) <- d$date
> m <- arima(x[1:463], c(2,0,1))
> p <- predict(m, 5)
> x2 <- c(x[1:463], p$pred); names(x2) <- d$date
> library(xts)
> library(xtsExtra)
> plot.xts(as.xts(cbind(x[460:468], x2[460:468])), type="b",
+          screens = factor(1, 1), col=c("red", "blue"), pch=1:2)
> legend("topleft", c("actual", "prediction"), col=c("red", "blue"),
+        lty=1:2, pch=1:2)
```



ARMA모형 예측 예 3

```
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01",
+                         date_end="2013-11-23")
> x <- d$foreigner; names(x) <- d$date
> m <- arima(x[1:464], c(2,0,1))
> p <- predict(m, 4)
> x2 <- c(x[1:464], p$pred); names(x2) <- d$date
> library(xts)
> library(xtsExtra)
> plot.xts(as.xts(cbind(x[460:468], x2[460:468])), type="b",
+          screens = factor(1, 1), col=c("red", "blue"), pch=1:2)
> legend("topleft", c("actual", "prediction"), col=c("red", "blue"),
+        lty=1:2, pch=1:2)
```



ARMA모형 예측 예 4

```
> d <- get_quantbook_data("krx_stock_daily_investor", "2012-01-01",
+                         date_end="2013-11-23")
> x <- d$foreigner; names(x) <- d$date
> m <- arima(x[1:465], c(2,0,1))
> p <- predict(m, 3)
> x2 <- c(x[1:465], p$pred); names(x2) <- d$date
> library(xts)
> library(xtsExtra)
> plot.xts(as.xts(cbind(x[460:468], x2[460:468])), type="b",
+          screens = factor(1, 1), col=c("red", "blue"), pch=1:2)
> legend("topleft", c("actual", "prediction"), col=c("red", "blue"),
+        lty=1:2, pch=1:2)
```