제10강: 시계열분석 2

금융 통계 및 시계열 분석

TRADE INFORMATIX

2014년 2월 11일

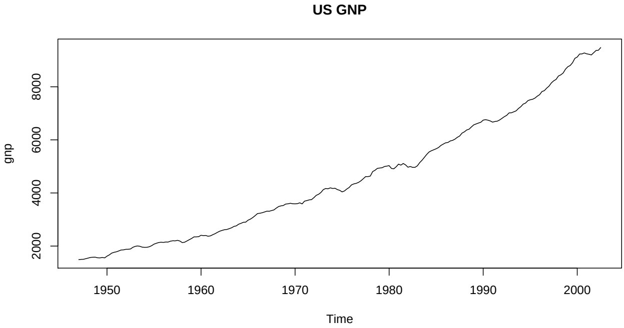
목차

비정상 시계열 (Non-stationary Time Series)

- 비정상 시계열 (Non-stationary Time Series)
  - 시간에 따라 평균 (mean), 공분산 (covariance) 등이 변화
- 비정상 시계열의 예
  - GDP, CPI 등 경제성장과 함께 증가하는 수치
  - 누적 거래량 등 시간에 따라 누적된 수치
  - 자연현상, 에너지 사용량 등 강한 계절성을 가지는 수치
  - 변동성의 크기가 변화하는 수치
- 모형 방법론
  - 성분 (Component) 모형
  - 분산변동 (Heteroskedastic) 모형

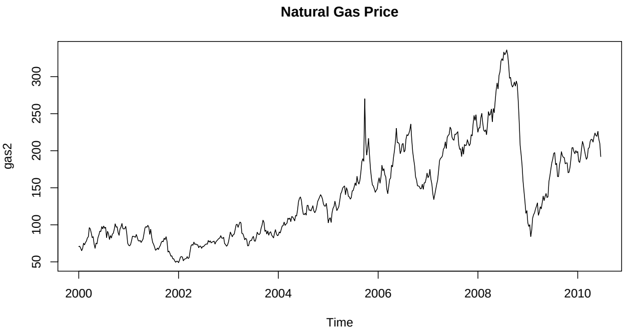
비정상 시계열의 예: 추세

```
> require(astsa, quietly=TRUE)
> data(gnp)
> plot(gnp, main="US GNP")
```



비정상 시계열의 예: 계절성

```
> require(astsa, quietly=TRUE)
> data(gas)
> N <- length(gas)
> gas2 <- window(gas, start=1975)
> plot(gas2, type='l', main="Natural Gas Price")
```



정상상태 테스트 (Stationary Test)

- 시계열이 정상상태 조건을 만족하는지 검사
- Unit Root Test
- 종류
  - ADF(Augmented Dickey-Fuller) Test
  - PP(Phillps-Peron) Test
  - ERS(Elliot-Rothenberg-Stock) Test
  - SP(Schmidt-Phillips) Test
  - KPSS(Kwiatkowski-Phillips-Schmidt-Shin) Test

(Augmented) Dickey-Fuller Test

- Original DF Test
  - AR(1) 시계열의 계수 a_1 이 1인지 테스트
  - 실제로는 Δx_t 와 x_{t-1} 의 회귀분석 계수가 0인지 테스트

$$
\begin{aligned}
x_t &= a_1 x_{t-1} + w_t \\
x_t - x_{t-1} &= a_1 x_{t-1} - x_{t-1} + w_t \\
\Delta x_t &= (1 - a_1) x_{t-1} + w_t
\end{aligned}
$$

  - residual의 autocorrelation 문제
- Augmented DF Test
  - AR(p) 시계열 모형 이용

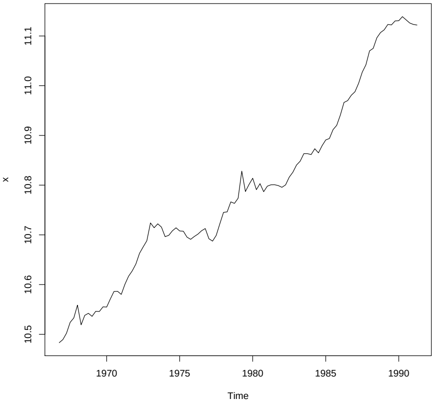
$$
\Delta x_t = (1 - a_1) x_{t-1} + \sum_{j=1}^{p} c_j \Delta x_j + w_t
$$

R에서 ADF 테스트 사용

- 다양한 패키지에서 구현
  - fUnitRoots 패키지의 `adftest`
  - tseries 패키지의 `adf.test`
  - urca 패키지의 `ur.df`
  - uroot 패키지의 `ADF.test`
- `ur.df(x, type, lags=1, selectlags)`
  - `x)` : 시계열 자료
  - `type)` : 추세성분을 추가하는 경우 `"trend"`
  - `lags` : residual autocorrelation 모형차수

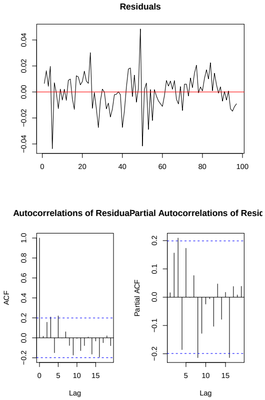
ADF 테스트 예 1

```
> require(urca, quietly=TRUE)
> data(Raotbl3)
> attach(Raotbl3)
> x <- ts(lc,
+         start=c(1966,4),
+         end=c(1991,2),
+         frequency=4)
> plot(x)
```



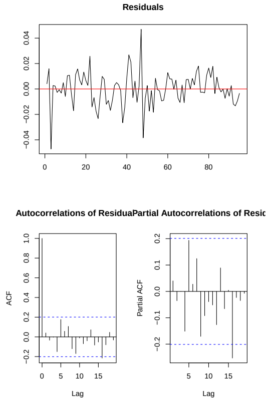
ADF 테스트 예 2

```
> m <- ur.df(x, lags=1)
> plot(m)
> summary(m)

###############################################
# 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
-0.043588 -0.007067  0.000962  0.007999  0.048462

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
z.lag.1     0.0006805  0.0001433   4.749 7.24e-06 ***
z.diff.lag -0.1243891  0.1020458  -1.219    0.226
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0137 on 95 degrees of freedom
Multiple R-squared: 0.198,  Adjusted R-squared: 0.1811
F-statistic: 11.73 on 2 and 95 DF,  p-value: 2.812e-05


Value of test-statistic is: 4.7485

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



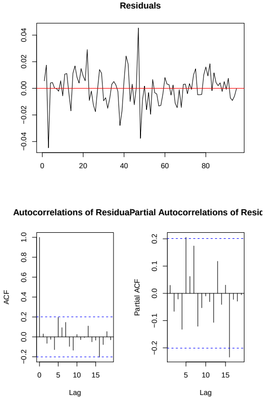
ADF 테스트 예 3

```
> m <- ur.df(x, lags=3)
> plot(m)
> summary(m)

###############################################
# 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
-0.047220 -0.007276  0.000229  0.007674  0.046921

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1      0.0004083  0.0001695   2.409   0.0180 *
z.diff.lag1 -0.1444994  0.1007615  -1.434   0.1550
z.diff.lag2  0.1599782  0.1009153   1.585   0.1164
z.diff.lag3  0.2568572  0.1015353   2.530   0.0131 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0133 on 91 degrees of freedom
Multiple R-squared: 0.2546, Adjusted R-squared: 0.2218
F-statistic:  7.77 on 4 and 91 DF,  p-value: 1.967e-05


Value of test-statistic is: 2.4089

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



ADF 테스트 예 4

```
> m <- ur.df(x, lags=3, type="trend")
> plot(m)
> summary(m)

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

Test regression trend


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

Residuals:
      Min        1Q    Median        3Q       Max
-0.044714 -0.006525  0.000129  0.006225  0.045353

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.7976591  0.3547775   2.248   0.0270 *
z.lag.1     -0.0758706  0.0338880  -2.239   0.0277 *
tt           0.0004915  0.0002159   2.277   0.0252 *
z.diff.lag1 -0.1063957  0.1006744  -1.057   0.2934
z.diff.lag2  0.2011373  0.1012373   1.987   0.0500 .
z.diff.lag3  0.2998586  0.1020548   2.938   0.0042 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01307 on 89 degrees of freedom
Multiple R-squared: 0.1472, Adjusted R-squared: 0.09924
F-statistic: 3.071 on 5 and 89 DF,  p-value: 0.01325


Value of test-statistic is: -2.2389 3.7382 2.5972

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -4.04 -3.45 -3.15
phi2  6.50  4.88  4.16
phi3  8.73  6.49  5.47
```



Pillips-Perron Test

- ❏ Non-parametric test로 AR차수지정이 필요하지 않음

$$\Delta x_t \quad = \quad \mu + \alpha x_{t-1} + w_t$$

- ❏ 데이터 수가 적을 경우 정확성이 떨어짐
  - ▶ tseries 패키지의 `pp.test`
  - ▶ urca 패키지의 `ur.pp`
- ❏ `ur.pp(x, model)`
  - ▶ `x)` : 시계열 자료
  - ▶ `model)` : 추세성분을 추가하는 경우 `"trend"`

PP 테스트 예

```
> m <- ur.pp(x)
> summary(m)

##################################
# Phillips-Perron Unit Root Test #
##################################

Test regression with intercept


Call:
lm(formula = y ~ y.l1)

Residuals:
      Min        1Q    Median        3Q       Max
-0.047503 -0.007042  0.001483  0.007879  0.047975

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.010867   0.082462   0.132    0.895
y.l1        0.999597   0.007643 130.779   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01374 on 96 degrees of freedom
Multiple R-squared: 0.9944, Adjusted R-squared: 0.9944
F-statistic: 1.71e+04 on 1 and 96 DF,  p-value: < 2.2e-16


Value of test-statistic, type: Z-alpha  is: -0.0572

         aux. Z statistics
Z-tau-mu            0.1509
```

성분 모형 (Component Model)

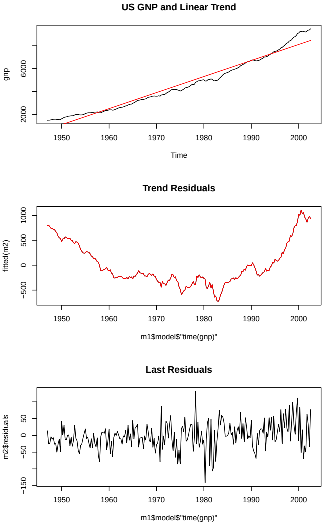
- 다양한 비정상 특성을 보이는 시계열을 특성별로 나누어 모형화하는 방법
  - 추세 (trend)
    - 결정론적 (deterministic) 추세
    - 확률적 (stochastic) 추세
  - 계절성 (seasonality)
    - 결정론적 계절성
    - 확률적 계절성
  - 확률론적 정상 신호
    - ARMA
- 추세 조합 방법
  - Additive Model
  - Multiplicative Model

결정론적 추세 모형

- ❑ 회귀분석을 이용, 추세 성분을 표시하는 결정론적 함수을 추정
- ❑ 보통 1차(linear) 혹은 2차(quadratic) 회귀분석 사용
- ❑ 비선형성을 보정하기 위한 log 변환(transformation) 등을 사용할 수 있음

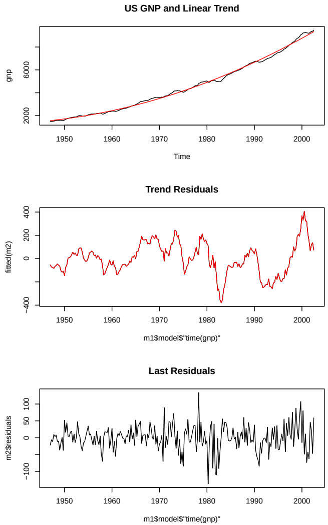
결정론적 추세 모형의 예 1: 선형 추세 + ARMA

```
> require(forecast, quietly=TRUE)
> (m1 <- lm(gnp~time(gnp)))

Call:
lm(formula = gnp ~ time(gnp))

Coefficients:
(Intercept)    time(gnp)
  -272289.7        140.2

> (m2 <- arima(m1$residuals, order=c(1, 0, 1)))

Series: m1$residuals
ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1  intercept
      0.9960  0.2779   613.1519
s.e.  0.0038  0.0534   497.7602

sigma^2 estimated as 1591:  log likelihood=-1141.13
AIC=2290.26   AICc=2290.44   BIC=2303.89

> layout(matrix(1:3))
> plot(gnp, main="US GNP and Linear Trend")
> lines(m1$model$"time(gnp)", m1$fitted.values,
+       col="red")
> plot(m1$model$"time(gnp)", fitted(m2),
+      type='l', main="Trend Residuals")
> lines(m1$model$"time(gnp)", fitted(m2),
+       col="red")
> plot(m1$model$"time(gnp)", m2$residuals,
+      type='l', main="Last Residuals")
```



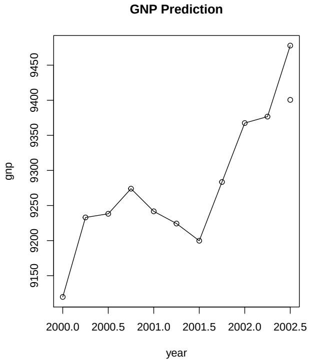
결정론적 추세 모형의 예 2: 2차 추세 + ARMA

```
> require(forecast, quietly=TRUE)
> (m1 <- lm(gnp~time(gnp)+I(time(gnp)^2)))

Call:
lm(formula = gnp ~ time(gnp) + I(time(gnp)^2))

Coefficients:
   (Intercept)       time(gnp)  I(time(gnp)^2)
     6.462e+06      -6.681e+03       1.727e+00

> (m2 <- arima(m1$residuals, order=c(1, 0, 1)))

Series: m1$residuals
ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1  intercept
      0.9420  0.2499     4.8782
s.e.  0.0221  0.0560    50.8461

sigma^2 estimated as 1422:  log likelihood=-1127.26
AIC=2262.51   AICc=2262.69   BIC=2276.14

> layout(matrix(1:3))
> plot(gnp, main="US GNP and Linear Trend")
> lines(m1$model$"time(gnp)", m1$fitted.values,
+       col="red")
> plot(m1$model$"time(gnp)", fitted(m2),
+      type='l', main="Trend Residuals")
> lines(m1$model$"time(gnp)", fitted(m2),
+       col="red")
> plot(m1$model$"time(gnp)", m2$residuals,
+      type='l', main="Last Residuals")
```



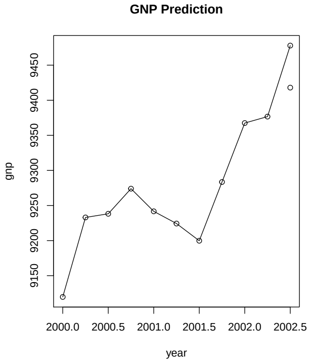
선형 추세 + ARMA 모형을 이용한 예측

```
> N <- length(gnp); y <- gnp[-N]; x <- time(gnp)[-N];
> m1 <- lm(y~x)
> m2 <- arima(m1$residuals, order=c(1, 0, 1))
> new <- data.frame(x=time(gnp)[N])
> (p1 <- predict(m1, new))

       1
8444.629

> (p2 <- predict(m2, n.ahead=1))

$pred
Time Series:
Start = 223
End = 223
Frequency = 1
[1] 955.8833

$se
Time Series:
Start = 223
End = 223
Frequency = 1
[1] 39.65396

> (p <- p1 + p2$pred)

Time Series:
Start = 223
End = 223
Frequency = 1
       1
9400.512

> gnp[N]

[1] 9477.9

> plot(window(gnp, start=2000),
+      type='o',
+      xlab="year", ylab="gnp",
+      main="GNP Prediction")
> points(time(gnp)[N], p)
```



2차 추세 + ARMA 모형을 이용한 예측

```
> N <- length(gnp); y <- gnp[-N]; x <- time(gnp)[-N];
> m1 <- lm(y~x+I(x^2))
> m2 <- arima(m1$residuals, order=c(1, 0, 1))
> new <- data.frame(x=time(gnp)[N])
> (p1 <- predict(m1, new))

       1
9340.238

> (p2 <- predict(m2, n.ahead=1))

$pred
Time Series:
Start = 223
End = 223
Frequency = 1
[1] 77.76912

$se
Time Series:
Start = 223
End = 223
Frequency = 1
[1] 37.59129

> (p <- p1 + p2$pred)

Time Series:
Start = 223
End = 223
Frequency = 1
       1
9418.007

> gnp[N]

[1] 9477.9

> plot(window(gnp, start=2000),
+      type='o',
+      xlab="year", ylab="gnp",
+      main="GNP Prediction")
> points(time(gnp)[N], p)
```



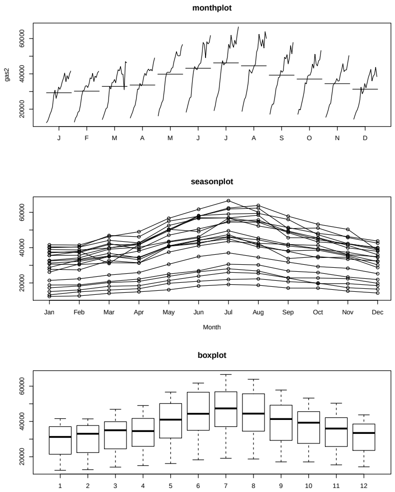
결정론적 계절성 모형

- 사이클 모형 (Cycle Model)
  - 단일 주기를 가지는 경우
- 주파수 모형 (Harmonics Model)
  - 복수의 sin/cos 주기신호의 조합으로 표시

R 사이클 모형 명령 1

- `monthplot` (forecast 패키지) : 연단위 평균과 월 패턴을 같이 플롯
- `seasonplot` (forecast 패키지) : 월 간격으로 플롯

```
> require(astsa, quietly=TRUE)
> data(gas)
> N <- length(gas)
> gas2 <- window(gas, start=1975)
> layout(matrix(1:3))
> monthplot(gas2,
+           main="monthplot")
> seasonplot(gas2,
+            main="seasonplot")
> boxplot(gas2~cycle(gas2),
+         main="boxplot")
```



R 사이클 모형 명령 2

- ❑ `cycle` : 1년 주기의 시계열의 월 팩터 시계열 생성
- ❑ `aggregate` : 연간 총합 시계열 생성

```
> cycle(gas2)

     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1975   1   2   3   4   5   6   7   8   9  10  11  12
1976   1   2   3   4   5   6   7   8   9  10  11  12
1977   1   2   3   4   5   6   7   8   9  10  11  12
1978   1   2   3   4   5   6   7   8   9  10  11  12
1979   1   2   3   4   5   6   7   8   9  10  11  12
1980   1   2   3   4   5   6   7   8   9  10  11  12
1981   1   2   3   4   5   6   7   8   9  10  11  12
1982   1   2   3   4   5   6   7   8   9  10  11  12
1983   1   2   3   4   5   6   7   8   9  10  11  12
1984   1   2   3   4   5   6   7   8   9  10  11  12
1985   1   2   3   4   5   6   7   8   9  10  11  12
1986   1   2   3   4   5   6   7   8   9  10  11  12
1987   1   2   3   4   5   6   7   8   9  10  11  12
1988   1   2   3   4   5   6   7   8   9  10  11  12
1989   1   2   3   4   5   6   7   8   9  10  11  12
1990   1   2   3   4   5   6   7   8   9  10  11  12
1991   1   2   3   4   5   6   7   8   9  10  11  12
1992   1   2   3   4   5   6   7   8   9  10  11  12
1993   1   2   3   4   5   6   7   8   9  10  11  12
1994   1   2   3   4   5   6   7   8   9  10  11  12
1995   1   2   3   4   5   6   7   8

> aggregate(gas2)

Time Series:
Start = 1975
End = 1994
Frequency = 1
 [1] 190428 220645 245077 269459 291470 345982 418393 432898
 [9] 438038 455014 461061 474263 517757 535174 580150 560521
[17] 537544 571926 555988 612430
```

R 사이클 모형 추정

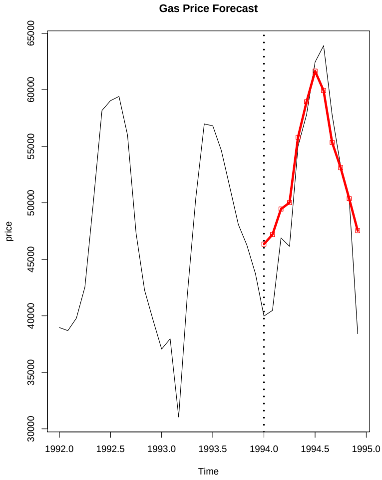
- `lm` 명령 사용
- 설명계수: 0, `time`, `factor(cycle)`

```
> (m <- lm(gas2 ~ 0 + time(gas2) + factor(cycle(gas2))))

Call:
lm(formula = gas2 ~ 0 + time(gas2) + factor(cycle(gas2)))

Coefficients:
           time(gas2)   factor(cycle(gas2))1
                 1764               -3473201
 factor(cycle(gas2))2   factor(cycle(gas2))3
             -3472429               -3469859
 factor(cycle(gas2))4   factor(cycle(gas2))5
             -3469261               -3463255
 factor(cycle(gas2))6   factor(cycle(gas2))7
             -3460049               -3457147
 factor(cycle(gas2))8   factor(cycle(gas2))9
             -3458960               -3463517
factor(cycle(gas2))10  factor(cycle(gas2))11
             -3465871               -3468614
factor(cycle(gas2))12
             -3471909
```

R 사이클 모형 예측

- lm 명령 사용
- 설명계수 0, time factor(cycle)

```
> gas3 <- window(gas2, end=c(1993,12))
> t <- time(gas3)
> c <- factor(cycle(gas3))
> m <- lm(gas3 ~ 0 + t + c)
> new <- data.frame(t=rep(1994, 12),
+                   c=factor(1:12))
> p <- ts(predict(m, new),
+         start=c(1994,1),
+         freq=12)
> plot(window(gas2, type='o',
+             start=c(1992,1),
+             end=c(1994,12)),
+      main="Gas Price Forecast",
+      ylab="price")
> lines(p, col="red", lwd=4)
> points(p, col="red", pch=0)
> abline(v=1994, lwd=3, lty="dotted")
```



확률적 추세 모형: Integrated 모형, ARIMA 모형

- order-d Integrated 모형, I(d)
  - d 번 차분 (difference) 하면 정상 (stationary) 상태가 되는 시계열
  - 차분 정상 (difference-stationary) 모형
  - 정상시계열은 0차 integrated 모형, I(0)
- Unit Root 프로세스, I(1)
  - 1차 Integrated 모형
  - $\Delta x_t = x_{t+1} - x_t$ 가 정상
- ARIMA(p,d,q) 모형 (Auto-Regressive Integrated Moving Average)
  - d 번 차분 (difference) 하면 ARMA(p,q) 모형이되는 시계열
  - 특성방정식의 해 중 d개가 1, 나머지는 절대값이 1보다 큼

$$\theta_p(B)(1-B)^d x_t = \phi_q(B) w_t \tag{1}$$

랜덤워크 (Random Walk)

❑ 랜덤워크

- ARIMA(0,1,0) 모형

$$x_t - x_{t-1} = w_t \text{ or, } x_t = x_{t-1} + w_t \tag{2}$$

여기에서 $w_t \sim N(0, \sigma)$

- 평균은 0이지만 분산은 시간에 따라 증가

$$E[x_t] = 0 \tag{3}$$

$$\text{Var}[x_t] = \sigma^2 t \tag{4}$$

- unit root process의 특별한 경우

❑ drift가 있는 랜덤워크

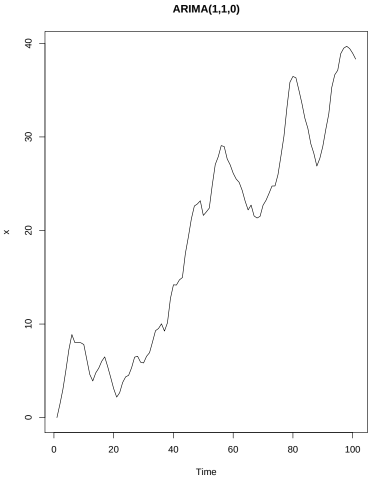
$$x_t = x_{t-1} + \mu + w_t = x_0 + \mu t + \sum_{i=0}^{t} w_t \tag{5}$$

- 결정론적 추세가 있는 랜덤워크
- 단기/일중 주가 모형

ARIMA 모형의 시뮬레이션

- `arima.sim(model, n)`
  - `model=list(order, ar, ma)` : 차수 및 계수 벡터
  - `n` : 시뮬레이션 갯수

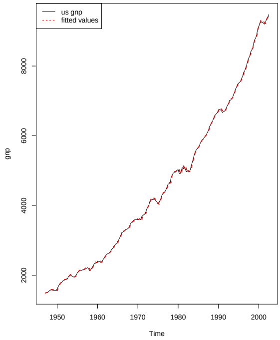
```
> set.seed(1)
> m <- list(order = c(1,1,0), ar = 0.7)
> x <- arima.sim(m, 100)
> plot(x, main="ARIMA(1,1,0)")
```



ARIMA 모형의 추정

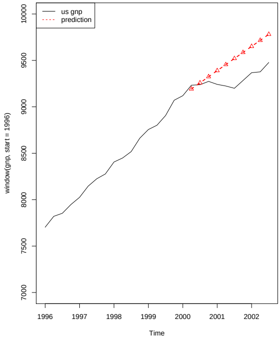
- ❑ `arima(x, order)`
  - ▶ `x` : 시계열 자료
  - ▶ `order=c(p,d,q)` : 차수
- ❑ `auto.arima(x)` : forecast 패키지
  - ▶ `x` : 시계열 자료

```
> m <- auto.arima(gnp)
> summary(m)

Series: gnp
ARIMA(2,2,1)

Coefficients:
         ar1     ar2      ma1
      0.2799  0.1592  -0.9735
s.e.  0.0682  0.0682   0.0142

sigma^2 estimated as 1451:  log likelihood=-1119.01
AIC=2246.02   AICc=2246.21   BIC=2259.62

Training set error measures:
                   ME     RMSE      MAE        MPE
Training set 3.735674 37.91694 29.15062 0.08532321
                  MAPE      MASE          ACF1
Training set 0.7254322 0.1855389 -0.009664696

> plot(gnp)
> lines(fitted(m), col="red", lwd=2, lty=2)
> legend("topleft", col=c("black", "red"),
+        lty=1:2,
+        c("us gnp", "fitted values"))
```



ARIMA 모형을 이용한 예측

```
> m <- auto.arima(window(gnp, end=2000))
> p <- predict(m, n.ahead=10)
> plot(window(gnp, start=1996),
+      ylim=c(7000, 10000))
> lines(p$pred, col="red", lwd=2, lty=2)
> points(p$pred, col="red", pch=2)
> legend("topleft", col=c("black", "red"),
+        lty=1:2,
+        c("us gnp", "prediction"))
```



- Seasonal ARIMA 모형 : ARIMA(p,d,q)(P,D,Q)[S]

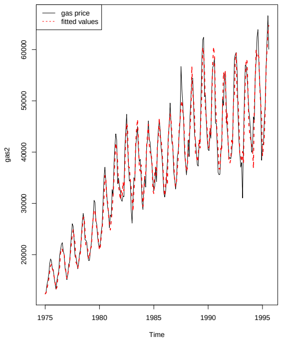
$$\theta_P(B^s)\theta_p(B)(1-B^s)^D(1-B)^d x_t = \phi_Q(B^s)\phi_q(B)w_t \tag{6}$$

  - ARIMA(p,d,q) 모형에 주기 s의 ARIMA(P,D,Q) 모형을 결합
- 주기 s의 ARIMA(P,D,Q) 모형
  - s의 배수만큼 과거의 신호만을 이용한 모형
  - x_t 는 $x_{t-s}, x_{t-s2}, \cdots, w_t, w_{t-s}$ 등에만 영향을 받음

SARIMA 모형의 추정

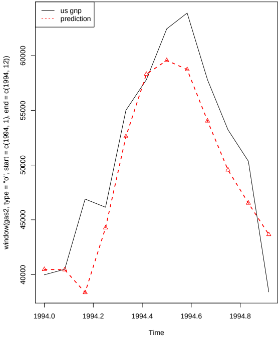
- `arima(x, order, seasonal)`
  - `seasonal=list(order=c(P,D,Q),period))` : 계절성 차수
- `auto.arima(x, d, D, max.p, max.q, max.P, max.Q, seasonal)`
  - `d, D` : integration 차수
  - `max.p, max.q, max.P, max.Q` : 최대 가능 ARMA 차수
  - `seasonal` : `TRUE`이면 계절성 추가

```
> m <- auto.arima(gas2)
> summary(m)

Series: gas2
ARIMA(2,1,1)(2,0,0)[12]

Coefficients:
         ar1     ar2      ma1    sar1    sar2
      0.4815  0.2072  -0.9606  0.5481  0.3401
s.e.  0.0733  0.0724   0.0297  0.0619  0.0629

sigma^2 estimated as 5308472:  log likelihood=-2271.23
AIC=4554.47   AICc=4554.82   BIC=4575.52

Training set error measures:
                   ME     RMSE      MAE       MPE     MAPE
Training set 178.4515 2299.362 1641.488 0.4011571 4.404438
                  MASE         ACF1
Training set 0.5566454 -0.005337921

> plot(gas2)
> lines(fitted(m), col="red", lwd=2, lty=2)
> legend("topleft", col=c("black", "red"),
+        lty=1:2,
+        c("gas price", "fitted values"))
```



SARIMA 모형을 이용한 예측

```
> m <- auto.arima(window(gas2, end=c(1993,12)))
> p <- predict(m, n.ahead=12)
> plot(window(gas2, type='o',
+             start=c(1994,1), end=c(1994,12)))
> lines(p$pred, col="red", lwd=2, lty=2)
> points(p$pred, col="red", pch=2)
> legend("topleft", col=c("black", "red"),
+        lty=1:2,
+        c("us gnp", "prediction"))
```



Transfer Function 모형

- ARMAX 모형
- ARMA 모형과 형식은 같으나 random innovation이 아닌 기지의 (known) 외부 입력 시계열을 가진다.

$$\theta_p(B)x_t = \phi_q(B)x_t + \psi_r(B)w_t \tag{7}$$

- $\theta_p(B)^{-1}\phi_q(B)$는 외부입력에 대한 Impulse Response Function
- TSA 패키지의 `arima` 혹은 `arimax` 명령을 사용
- `arimax(x, order, xreg)`
  - `xreg` : 외부 입력 시계열