



R Time Series

인하대학교 데이터사이언스학과

김 승 환

swkim4610@inha.ac.kr

Time Series

▶ Discrete Time Series: 연도별 쌀 생산량, 월별 판매량 등으로 관측시간이 등간격인 경우 (equally spaced in time) 의 계열자료

▶ 시계열자료의 구성요인

전통적으로 시계열자료의 분석은 시계열이 여러 요인으로 구성되어 있다고 보는 견해가 있다.
그 요인으로 아래와 같은 요인을 고려한다.

1) 불규칙요인(irregular component): 시간에 따라 규칙적으로 움직이지 않는 Random한 요인으로 통계적인 방법으로 설명이 불가능한 요인이다.

2) 규칙요인(Regular component): 불규칙요인과는 반대로 규칙성을 가지는 요인이다.

추세요인(trend component): 시간이 경과함에 따라 지속적으로 증가하거나 감소하는 요인으로 반드시 직선일 필요는 없다.

계절요인(Seasonal component): 계절에 의한 주기적인 요인

순환요인(Cycle component): 계절에 의한 주기적인 변동이 아니라 경기순환과 같이 큰 주기를 가지는 요인

Time Series

□ 일반적인 iid Observation과 Time Series Data

$$Z_1, Z_2, \dots, Z_n \stackrel{iid}{\sim} F(x)$$

위의 표현에서 Z_i 들은 서로 독립이고 같은 분포를 갖는 모집단에서 추출된 n 개의 표본이란 의미로 이 경우 자료들은 관측순서에 상관없이 그 크기만이 관심의 대상이 되므로 이론전개가 용이하게 된다. (i.i.d: Identically, Independently Distributed)

하지만, 시계열 자료는 관측의 순서를 바꾸면 의미를 상실하게 되고 일반적으로 i 번째 관측값은 $i-1$, $i-2$ 등과 같이 과거의 관측값에 영향을 받는 구조를 보인다.
따라서, 시계열 자료는 일반적 i.i.d 관측값과 다르게 순서도 중요한 정보가 된다.

□ Prediction과 Forecasting 의 차이

Prediction: 추측이라는 의미로 우리가 알지 못하는 미지의 값(하지만, 이미 결정되어 있는 값)을 객관적 방법으로 추측하는 것을 말한다.

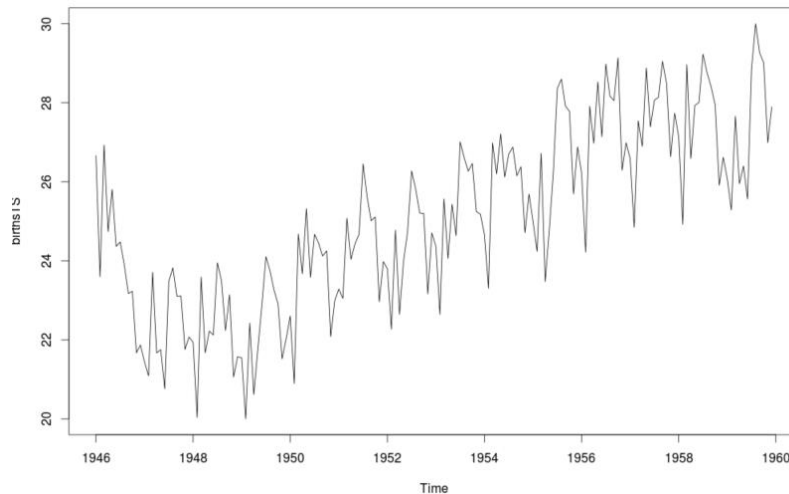
Forecasting: 예측이란 의미로 우리가 알지 못하는 미지의 값(결정되어 있지 않음)의 객관적인 방법으로 추측하는 것을 말한다.

Time Series Plotting

An example is a data set of the number of births per month in New York city, from January 1946 to December 1959 (originally collected by Newton).

This data is available in the file <http://robjhyndman.com/tsdldata/data/nybirths.dat>

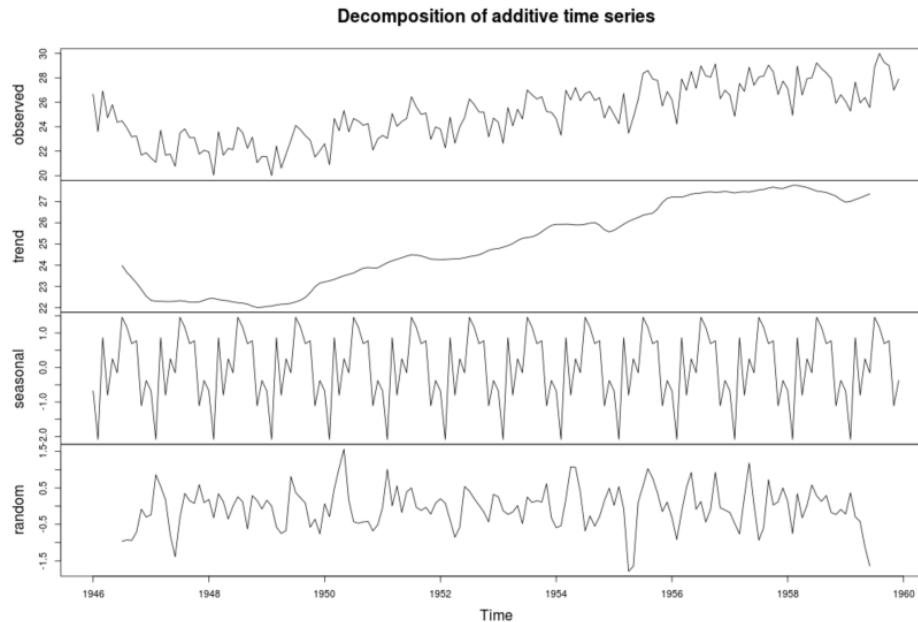
```
births <- scan("http://robjhyndman.com/tsdldata/data/nybirths.dat")
birthsTS <- ts(births, frequency=12, start=c(1946,1))
plot.ts(birthsTS)
```



Time Series Decomposing

Decomposing a time series means separating it into its constituent components, which are usually a trend component and an irregular component, and if it is a seasonal time series, a seasonal component.

```
birthsTSDecomp <- decompose(birthsTS)  
plot(birthsTSDecomp)
```



Time Series Decomposing

시계열 분해 모형은 Additive Model과 Multiplicative Model이 있다.

아래는 시계열이 Trend 요소($\beta_0 + \beta_1 t$) 와 계절요소 $\sum_{i=1}^s \delta_i IND_{t=i}$ 그리고, 불규칙 요소가 덧셈 형태로 구성되어 있다는 가정이다.

$$Z_t = \beta_0 + \beta_1 t + \sum_{i=1}^s \delta_i IND_{t=i} + \varepsilon_t$$

또 다른 모형을 Multiplicative Model 으로 Trend, 계절요소, 불규칙요소가 서로 곱해져 있다는 가정이다. 승법 모형의 경우, 로그변환을 하면 가법모형이 된다.

그러므로 자료에 로그변환을 취한 후, 가법 모형으로 추정한 후, exp 함수로 복원하면 승법 모형을 구현할 수 있다.

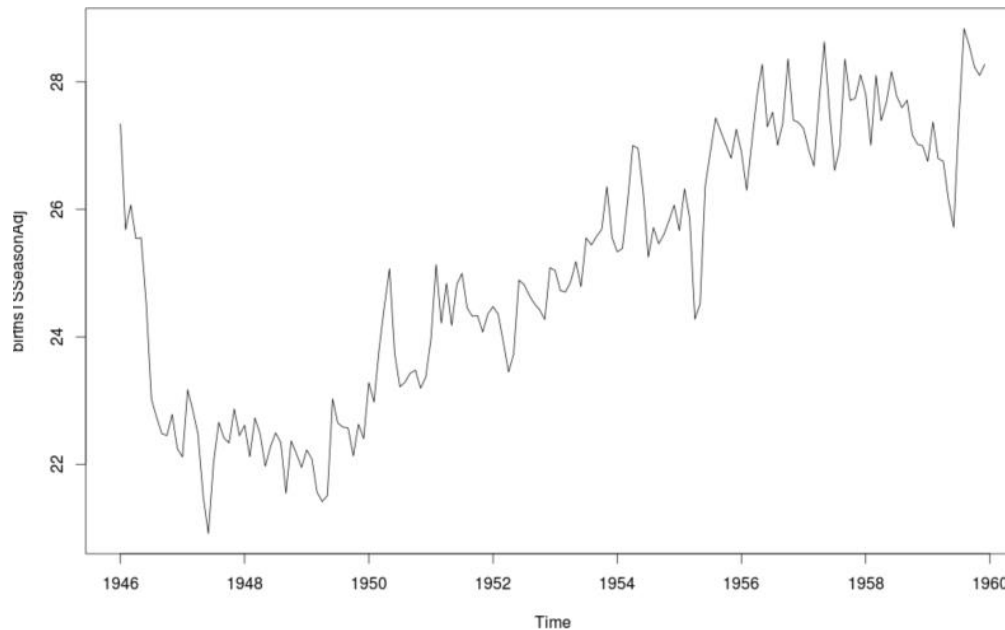
$$Z_t = T_t \cdot S_t \cdot \varepsilon_t$$

$$\log Z_t = \log T_t + \log S_t + \log \varepsilon_t$$

Time Series Decomposing

Decomposing a time series means separating it into its constituent components, which are usually a trend component and an irregular component, and if it is a seasonal time series, a seasonal component.

```
birthsTSSeasonAdj <- birthsTS - birthsTSDecomp$seasonal  
plot(birthsTSSeasonAdj)
```



Forecasting Methods

널리 사용되고 있는 시계열분석 방법은 아래와 같다.

- 1) 회귀분석방법(Regression method)
- 2) 평활법(Smoothing method)
- 3) 요소분해법(Decomposition method)
- 4) Box-Jenkin's 방법(ARIMA model)

회귀분석방법은 일반적으로 i.i.d. 데이터를 다룬다. 시계열 자료는 i.i.d.가 아니므로 자기상관이 있다는 가정하에 전개되는 회귀분석이론을 사용하게 된다.

평활법은 예측원점(Forecast origin)에서 가까운 관측값에는 큰 가중치를 주고 먼 관측치일수록 작은 가중치를 주는 일종의 가중평균을 이용하는 방법이다. 가중치를 주는 방법에 따라 이동평균법, 지수평활법으로 구분한다. 이들 방법은 이론적으로는 미흡한 점이 많으나 이해가 쉽고 계산이 용이한 방법이다.

3)의 요소분해법은 시계열 자료를 추세요인, 순환요인, 계절요인, 불규칙요인으로 분해하여 각각을 예측한 다음 다시 결합시키는 방법이다.

Box-Jenkin's 방법은 이론적으로 우수한 방법이고 또한 예측값이 가장 정확한 방법으로 널리 알려져 있는 방법이다. 그러나 이론적으로 이해가 어렵고 계산이 어려워 전문가들에 의해 주로 사용하는 방법이다.

Regression based Forecasting

□ 회귀분석 방법

<1> $f(\bullet)$ 이 선형(linear)인 경우의 예

$$Y_t = \beta_0 + \varepsilon_t \quad \text{constant mean model}$$

$$Y_t = \beta_0 + \beta_1 x_t + \varepsilon_t \quad \text{simple linear regression model}$$

$$Y_t = \beta_0 + \beta_1 x_t + \beta_2 x_t^2 + \varepsilon_t \quad \text{Quadratic model}$$

$$Y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \varepsilon_t$$

$$Y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \beta_3 x_{1t}^2 + \beta_4 x_{2t}^2 + \beta_5 x_{1t} x_{2t} + \varepsilon_t$$

<2> $f(\bullet)$ 이 비선형(non-linear)인 경우의 예

$$Y_t = \beta_0 \sin(\beta_1 x_t) + \varepsilon_t$$

시계열 데이터를 위의 모형으로 적합하면 잔차가 독립이라는 회귀분석의 가정을 위배하게 된다. 그러므로, 잔차가 독립이 아닌 경우의 회귀분석 방법을 사용하여야 한다.

이 과정을 위해서는 잔차의 독립성 검정(Durbin and Watson Method)과 자기상관이 존재하는 회귀분석으로 Cochrane Orcutt, Yule and Walker 추정법을 사용한다.

Regression based Forecasting

□ Durbin-Watson Statistic

일반적으로 회귀분석에서 1차 자기상관관계를 조사하는 방법으로 **Durbin-Watson Statistics** 을 많이 사용하고 있다.

$$DW = \frac{\sum_{t=2}^n (e_t - e_{t-1})^2}{\sum_{t=1}^n e_t^2} = \frac{\sum_{t=2}^n e_t^2 + \sum_{t=2}^n e_{t-1}^2 - 2 \sum_{t=2}^n e_t e_{t-1}}{\sum_{t=1}^n e_t^2} \cong 2(1 - r_1) \quad r_1 = \frac{\sum_{t=2}^n e_t e_{t-1}}{\sum_{t=1}^n e_t^2}$$

$-1 \leq r_1 \leq 1$ 이므로 DW 는 0 ~ 4 사이에 있게 되고 $r_1 > 0$ 이면 $0 < DW < 2$, $r_1 < 0$ 이면 $2 < DW < 4$ 에 있게 된다.

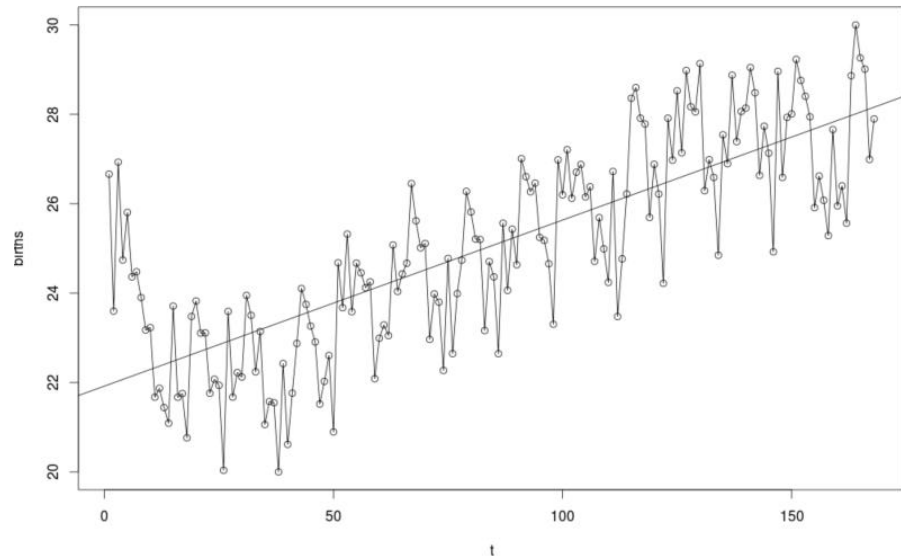
DW는 1차 자기상관을 조사하는데는 효율적이거나 2차 이상의 자기상관은 검사할 수 없다. 일반적으로 이와 같은 경우 $SACF(r_k)$ 를 구하여 이 값의 절대값이 $2/\sqrt{n}$ 보다 크면 자기상관관계가 그 시점에서 존재한다고 볼 수 있다.

Regression based Forecasting

```
t<-c(1:168)
reg1<-lm(births~t)
plot(t,births)
lines(t,births)
abline(reg1,col='red')
```

Call: `lm(formula = births ~ t)`
Coefficients: (Intercept) t
21.92028 0.03715

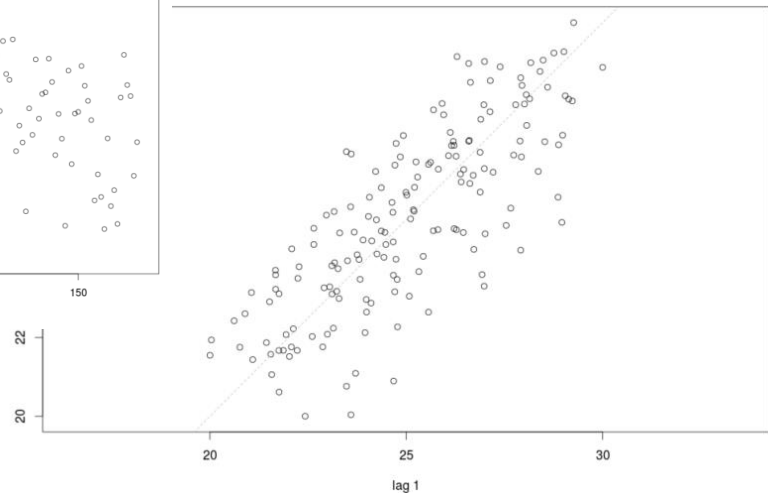
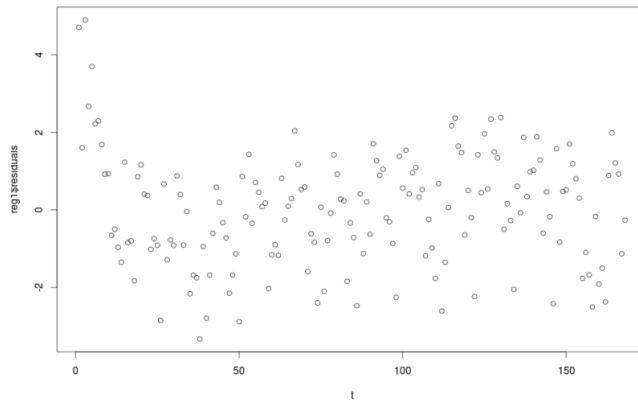
이 결과로 1960년 1월(169번째) 신생아의 수
예측값은 $21.92 + 0.03715 \times 169 = 28.2$ 가 된다.



Regression based Forecasting

```
plot(t,reg1$residuals)
lag.plot(birthsTS, 1)
library(lmtest)
dwtest(reg1)
```

Durbin-Watson test data: reg1 DW = 1.0733, p-value = 4.646e-10
alternative hypothesis: true autocorrelation is greater than 0



Cochrane-Orcutt Estimation

□ Cochrane Orcutt Estimation

시계열 자료를 일반적 회귀분석으로 예측식을 만들어 잔차에 자기 상관이 존재할 때, 자기상관을 없애는 방법으로 아래와 같은 변수변환을 통해 회귀식을 추정하는 방법을 Cochrane Orcutt Estimation 이라고 한다.

$$Y_1^* = \sqrt{1 - \rho^2} Y_1$$

$$Y_t^* = Y_t - \rho Y_{t-1}, t = 2, 3, \dots, n$$

$$X_{1j}^* = \sqrt{1 - \rho^2} X_{1j}$$

$$X_{tj}^* = X_{tj} - \rho X_{t-1,j}, t = 2, 3, \dots, n, j = 1, 2, \dots, p$$

일반적으로 자료의 갯수가 큰 경우에는 첫번째 변환된 자료를 제외하게 되는데 이는 첫번째 변환과 다른 변환(두번째 이후의 변환)이 서로 다른 구조를 갖기 때문이다. 이러한 방법을 Cochrane & Orcutt 추정법이라고 하는데 이 방법은 오차항이 1차 자기상관을 가질때 많이 사용되는 방법이다. 표본의 크기가 큰 경우 첫번째 관측치를 제거하는 것은 모수의 추정에 큰 영향을 줄 수 없다. 하지만, 표본의 크기가 작은 경우는 분산이 일반적인 회귀방법보다 클 수 있다. 특히, 설명변수가 추세를 가지는 경우에는 더욱 그러하게 된다.

Cochrane-Orcutt Estimation

ρ 의 추정치는 일반적인 상관계수의 MME(Method of Moments Estimator) $\hat{\rho} = \frac{\sum_{t=2}^n e_t e_{t-1}}{\sum_{t=2}^n e_t^2}$ 로 얻을 수

있고 여기서, e_t 는 (2.6)식의 OLS 추정에 의한 잔차(Residual)이다.

```
reg2<-cochrane.orcutt(reg1)
reg2
```

Cochrane-orcutt estimation for first order autocorrelation

Call:

```
lm(formula = births ~ t)
number of interaction: 3
rho 0.431913
```

Durbin-Watson statistic

```
(original): 1.07328 , p-value: 4.646e-10
(transformed): 2.31604 , p-value: 9.759e-01
coefficients:
```

```
(Intercept)          t
21.719459      0.038898
```

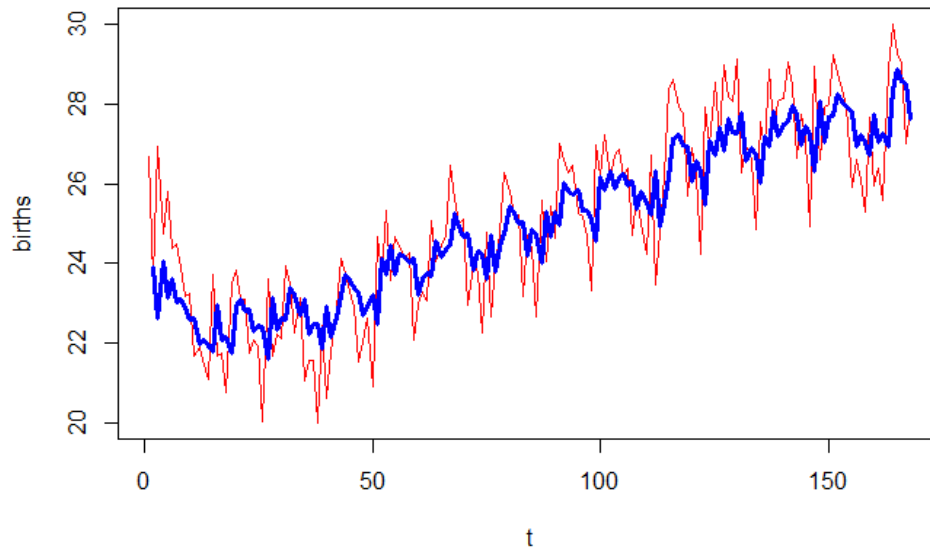
$$\hat{Y}_{t+1} = \hat{\rho}Y_t + \hat{\alpha}(1 - \hat{\rho}) + \hat{\beta}(X_{t+1} - \hat{\rho}X_t)$$

$$\hat{Y}_{t+1} = 0.43Y_t + 21.72(1 - 0.43) + 0.038(X_{t+1} - 0.43X_t)$$

Cochrane-Orcutt Estimation

```
lag <- function(x)c(NA, x[1:(length(x)-1)])  
births1<-lag(births)  
birthsEst<-0.43*births1 + 21.72*(1-0.43) + 0.038*(t-0.43*(t-1))  
plot(t,births,type="l",col="red")  
lines(t,birthsEst,col="blue", lwd = 3)  
births169<-0.43*27.897 + 21.72*(1-0.43) + 0.038*(169-0.43*168)
```

169주차 출산수는 28.05로 예측할 수 있다.



Exponential Smoothing

□ Simple Exponential Smoothing

지수평활법은 일종의 가중평균법이다. 즉, t 번째의 시계열 값은 그 전의 시계열값의 가중평균으로 나타내는 모형인데 이 때, 가중치는 과거로 갈수록 지수적으로 감소(exponentially decay)하는 가중치를 선택하는 방법이다.

이때에는 과거의 관측치 보다는 최근의 관측치에 좀 더 많은 가중치를 주는 방법을 생각할 수 있는데 만약, 선택한다면 n번째 데이터가 주어졌을 때, 그 이후 1 번째 예측값은 아래와 같이 추정할 수 있다. Simple Exponential Smoothing은 평균이 시간에 대해 거의 일정하거나 천천히 변하는 자료에 적합하다.

$$\begin{aligned} Z_n(l) &= c \sum_{t=0}^{n-1} (1-w)^t Z_{n-t} \\ &= c[Z_n + (1-w)Z_{n-1} + \dots + (1-w)^{n-1}Z_1] \end{aligned}$$

여기서, w는 평활상수라고 하고 대개 0.05~0.3 사이의 값이 많이 사용된다.

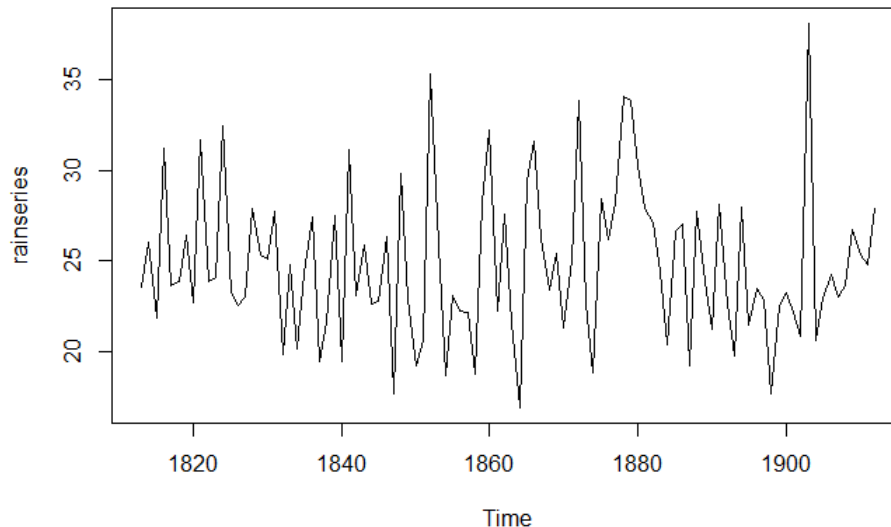
c는 이 가중치의 합이 "1"이 되도록 조정해주는 상수로 w가 결정되면 아래의 식으로 구한다.

$$c = \frac{1}{\sum_{t=0}^{n-1} (1-w)^t} = \frac{1}{\frac{1(1-(1-w)^{n-1})}{1-(1-w)}} \xrightarrow{n \rightarrow \infty} w$$

Exponential Smoothing

For example, the file <http://robjhyndman.com/tsdldata/hurst/precip1.dat> contains total annual rainfall in inches for London, from 1813-1912.

```
# Simple Exponential Smoothing
rain <- scan("http://robjhyndman.com/tsdldata/hurst/precip1.dat", skip=1)
rainseries <- ts(rain, start=c(1813))
plot.ts(rainseries)
```



Exponential Smoothing

```
rainSeriesForecasts <- HoltWinters(rainSeries, beta=FALSE, gamma=FALSE)
rainSeriesForecasts
plot(rainseriesforecasts)
```

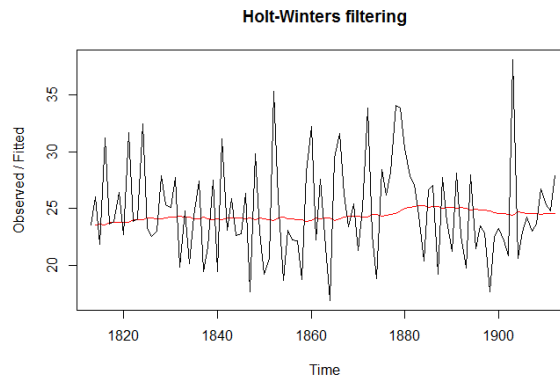
Holt-Winters exponential smoothing without trend and without seasonal component.

Call: `HoltWinters(x = rainSeries, beta = FALSE, gamma = FALSE)`

Smoothing parameters:

alpha: 0.02412151 beta : FALSE gamma: FALSE

위의 결과로 부터 w 는 0.024의 값이 예측오차 제곱합을 최소화하는 평활상수로 추정되었다.

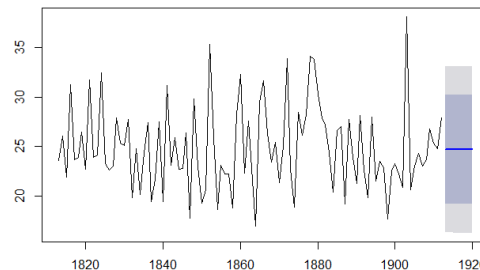


Exponential Smoothing

```
library(forecast)
rainSeriesForecasts2 <- forecast(rainSeriesForecasts, h=8)
rainSeriesForecasts2
plot(forecast(rainSeriesForecasts2))
```

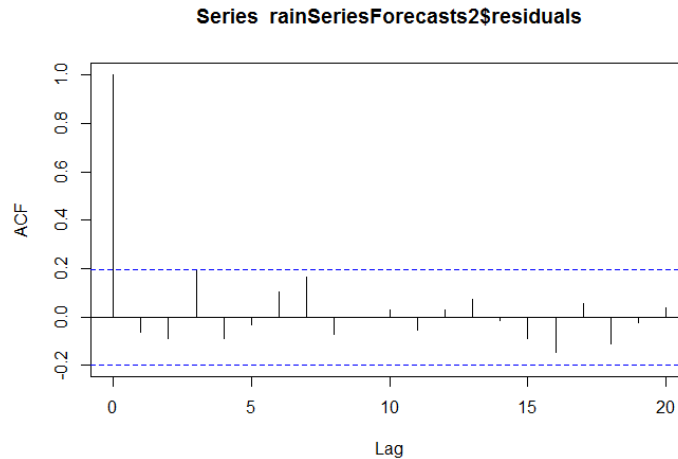
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1913	24.67819	19.17493	30.18145	16.26169	33.09470
1914	24.67819	19.17333	30.18305	16.25924	33.09715
1915	24.67819	19.17173	30.18465	16.25679	33.09960
1916	24.67819	19.17013	30.18625	16.25434	33.10204
1917	24.67819	19.16853	30.18785	16.25190	33.10449
1918	24.67819	19.16694	30.18945	16.24945	33.10694
1919	24.67819	19.16534	30.19105	16.24701	33.10938
1920	24.67819	19.16374	30.19265	16.24456	33.11182

Forecasts from HoltWinters



Exponential Smoothing

```
acf(rainSeriesForecasts2$residuals, lag.max=20, na.action = na.pass)
```



잔차에 대한 ACF(Autocorrelation Function)에서 $\text{lag} > 0$ 인 부분의 ACF 값이 작아 예측이 잘 이루어졌음을 알 수 있다.

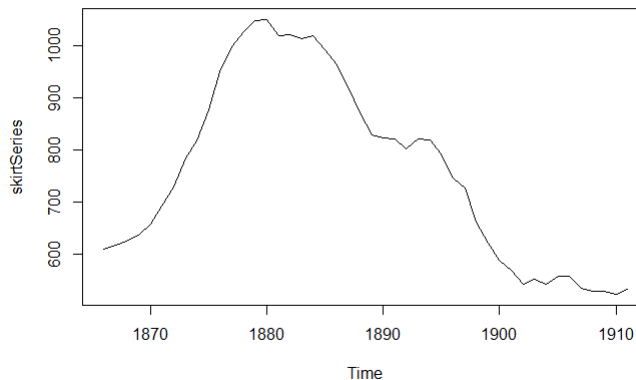
Exponential Smoothing

□ Double Exponential Smoothing

Double Exponential Smoothing은 stochastic trend가 존재하는 시계열에 적합한 방법이다.

Time series of the annual diameter of women's skirts at the hem(밑단), from 1866 to 1911. The data is available in the file <http://robjhyndman.com/tsdldata/roberts/skirts.dat>

```
# Double Exponential Smoothing
skirts <- scan("http://robjhyndman.com/tsdldata/roberts/skirts.dat", skip=5)
skirtSeries <- ts(skirts, start=c(1866))
plot.ts(skirtSeries)
```



Exponential Smoothing

```
skirtSeriesForecasts <- HoltWinters(skirtSeries, gamma=FALSE)
skirtSeriesForecasts
plot(skirtSeriesForecasts)
```

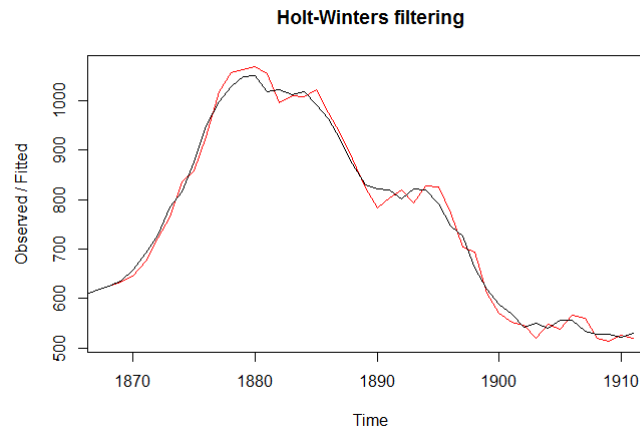
Holt-Winters exponential smoothing with trend and without seasonal component. Call: `HoltWinters(x = skirtSeries, gamma = FALSE)`

Smoothing parameters:

alpha: 0.8383481

beta : 1

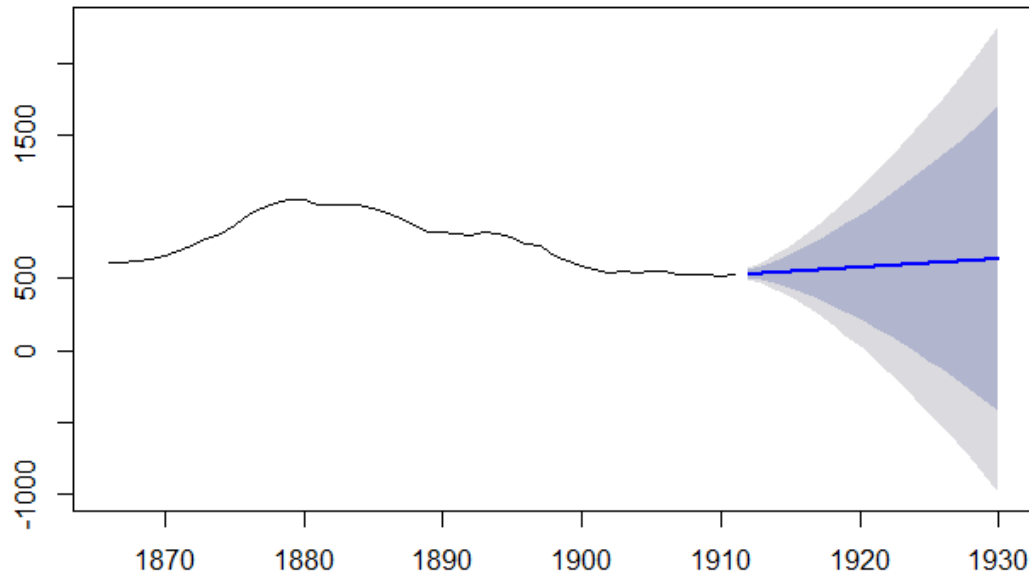
gamma: FALSE



Exponential Smoothing

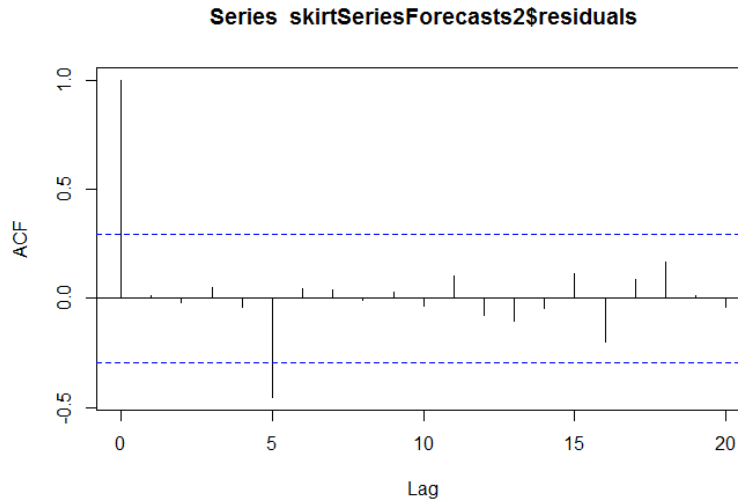
```
skirtSeriesForecasts2 <- forecast(skirtSeriesForecasts, h=19)  
plot(forecast(skirtSeriesForecasts2))
```

Forecasts from HoltWinters



Exponential Smoothing

```
acf(skirtSeriesForecasts2$residuals, lag.max=20, na.action = na.pass)
```



Exponential Smoothing

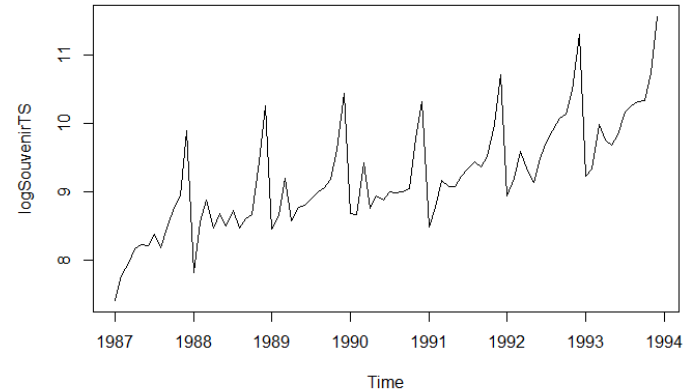
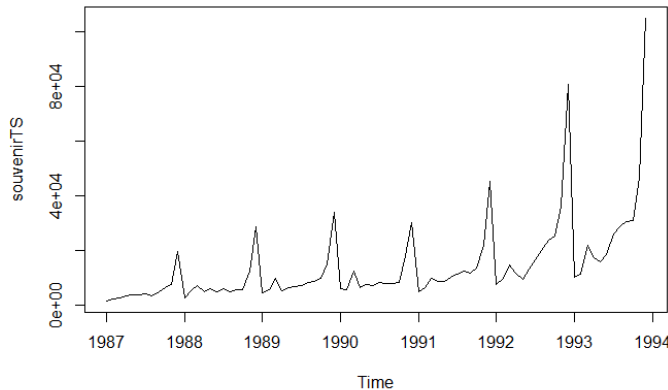
□ Holt & Winter Exponential Smoothing

Holt & Winter Exponential Smoothing 은 stochastic trend와 계절성이 존재하는 시계열에 적합한 방법이다.

An example of a time series that can probably be described using an additive model with a trend and seasonality is the time series of the log of monthly sales for the souvenir shop at a beach resort town in Queensland, Australia

```
# Holt and Winter Seasonal Exponential Smoothing
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirTS <- ts(souvenir, frequency=12, start=c(1987,1))
plot.ts(souvenirTS)
logSouvenirTS <- log(souvenirTS)
plot.ts(logSouvenirTS)
```

Exponential Smoothing



```
souvenirTSforecasts <- HoltWinters(logSouvenirTS)  
souvenirTSforecasts
```

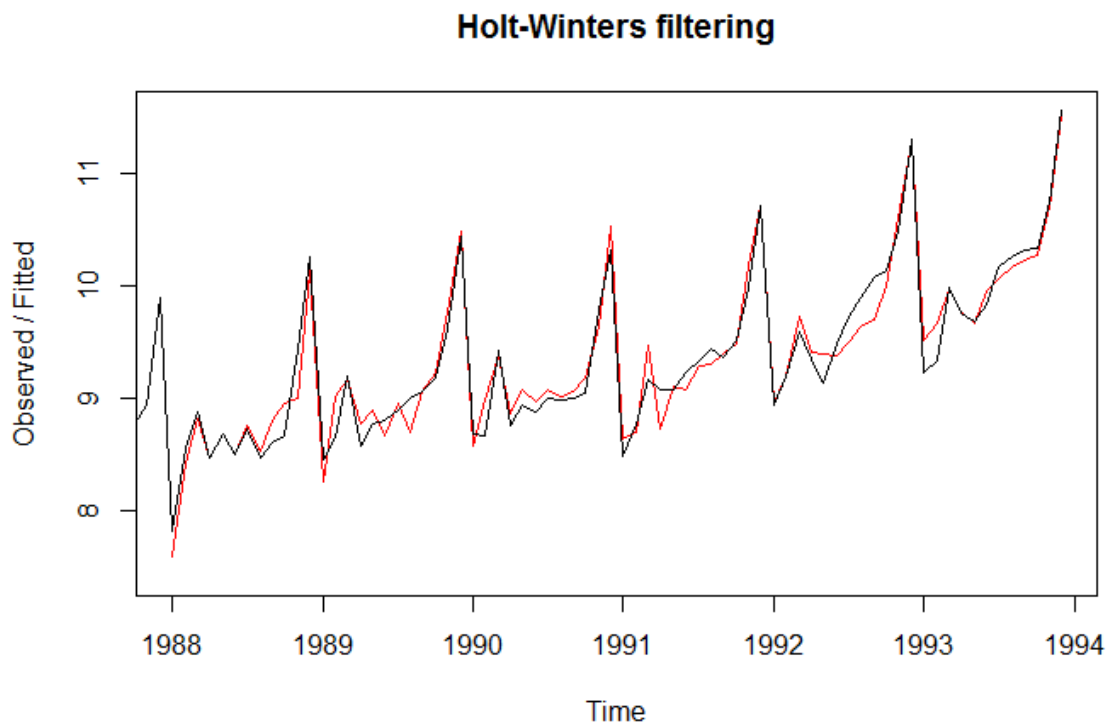
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call: `HoltWinters(x = logSouvenirTS)`

Smoothing parameters: alpha: 0.413418 beta : 0 gamma: 0.9561275

Exponential Smoothing

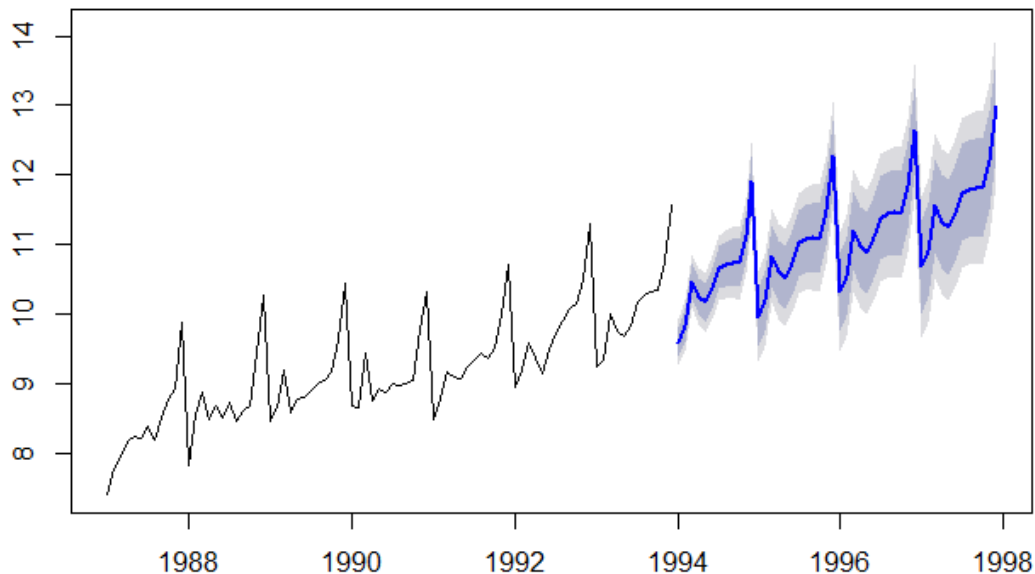
```
plot(souvenirTSForecasts)
```



Exponential Smoothing

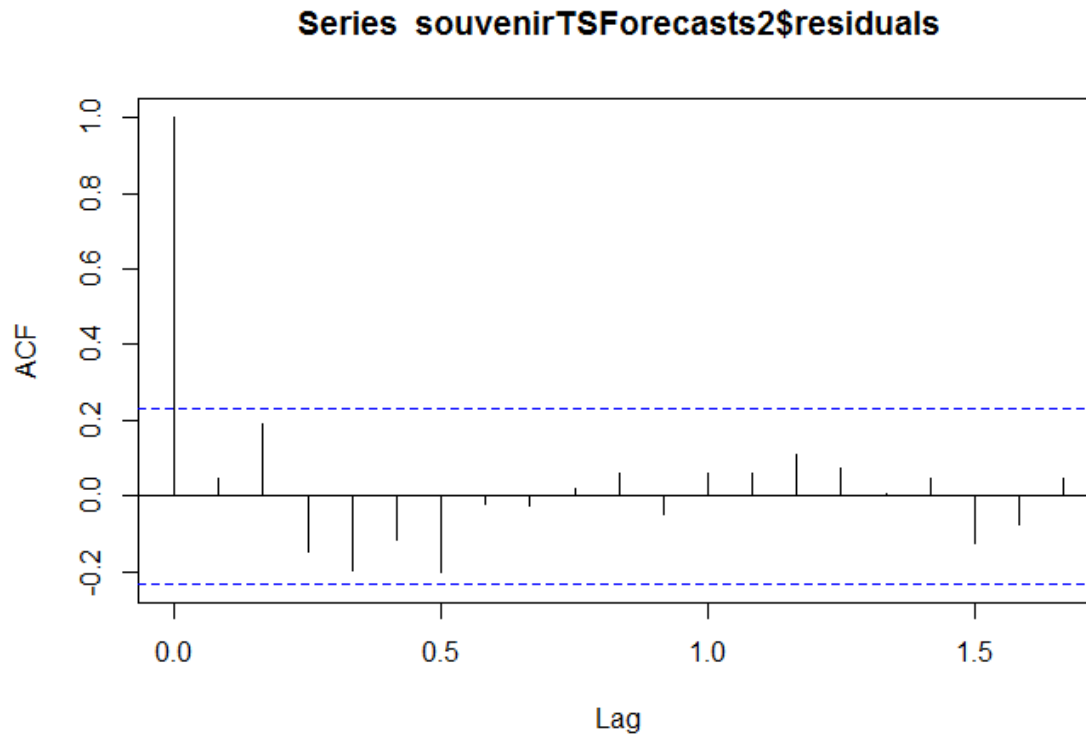
```
souvenirTSForecasts2 <- forecast(souvenirTSForecasts, h=48)  
plot(forecast(souvenirTSForecasts2))
```

Forecasts from HoltWinters



Exponential Smoothing

```
acf(souvenirTSForecasts2$residuals, lag.max=20, na.action = na.pass)
```



ARIMA Model

ARIMA: Auto-Regressive Integrated Moving Average

- *Stationary Process*: 엄격한 의미의 정상시계열은 시계열의 확률적인 성질들이 시간의 흐름에 따라 불변하는 시계열을 말한다.

$$f(z_{t_1}, z_{t_2}, \dots, z_{t_n}) = f(z_{t_1+k}, z_{t_2+k}, \dots, z_{t_n+k})$$

정상시계열은 모든 시간 t 에서 평균과 분산이 일정하고 공분산과 상관계수는 시차 k 에만 영향을 받게 된다.

- *Weakly Stationary Process*: 실제 상황에서 시계열이 정상성을 만족하는지 알 수 없기 때문에 모든 시간 t 에 대하여 평균과 분산이 일정한지를 파악하여 정상시계열로 간주한다.

First order Autoregressive process : AR(1) (Markov process)

$$Z_t - \mu = \phi(Z_{t-1} - \mu) + a_t \quad a_t : \text{White Noise}$$

시계열 값에서 평균을 빼주면 $Z_t = \phi Z_{t-1} + a_t$ 으로 표현 가능하다. 이를 확장하면

AR(p)는 $Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t$ 로 쓸 수 있다.

백색잡음이란 Pure Noise의 의미로 더 이상 분해할 수 없는 Random한 프로세스를 의미한다.

ARIMA Model

First order Moving Average process : MA(1)

$$Z_t - \mu = a_t - \theta a_{t-1}$$

MA(q)는 $Z_t - \mu = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}$ 로 쓸 수 있다.

ARMA(p,q) process

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \cdots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}$$

이 모형은 모두 Stationary Process에서 사용할 수 있는 모형이다.

만약, 시계열이 Non-Stationary Process 라면 위의 모형을 직접적으로 사용할 수 없다.

일반적으로 Non-Stationary Process는 차분(difference)을 통해 Stationary Process로 변환할 수 있다.

$$\text{First order Difference : } \nabla Z_t = Z_t - Z_{t-1}$$

$$\text{Second order Difference : } \nabla^2 Z_t = \nabla Z_t - \nabla Z_{t-1}$$

시계열을 차분하고 그래프를 그려서 시간에 따라 평균과 분산이 일정하게 변환이 되었다면 정상시계열로 변환되었다고 판단한다.

ARIMA(p,d,q) process : d차 차분 후, ARMA(p,q)로 표현 가능한 모형을 말한다.

ARIMA Model

diameter of women's skirts at the hem



위의 시계열은 Non-Stationary process로 보인다. 차분을 통해 Stationary Process로 변환해보자.

```
skirtSeriesD1 <- diff(skirtSeries, differences=1)
plot.ts(skirtSeriesD1)
skirtSeriesD2 <- diff(skirtSeries, differences=2)
plot.ts(skirtSeriesD2)
```


ARIMA Model

AR(1) 모형 $Z_t = \phi Z_{t-1} + a_t$ 에서 $\phi = 1$ 인 시계열을 Random Walk 모형이라고 하는데 이 경우, AR 모형을 적용할 수 없다.

아래 식에서 $\phi = 1$ 인 경우, γ_0 는 정의할 수 없다. 또한, $\rho_k=1$ 이 된다.

이를 검정하기 위해 Dickey-Fuller 검정을 사용한다.

$$\gamma_0 = Cov(Z_t, Z_t) = V(Z_t) = V(\phi Z_{t-1} + a_t) = \phi^2 V(Z_{t-1}) + \sigma^2 = \phi^2 \gamma_0 + \sigma^2$$

$$\gamma_0 = \frac{\sigma^2}{1 - \phi^2} \rightarrow \infty$$

$$\gamma_k = Cov(Z_t, Z_{t-k}) = E(Z_t \cdot Z_{t-k}) = E(\phi Z_{t-1} Z_{t-k} + Z_{t-k} a_t) = \phi \gamma_{k-1} = \phi^k \gamma_0$$

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\phi^k \gamma_0}{\gamma_0} = \phi^k, \text{ if } \phi = 1, \rho_k = 1$$

ARIMA Model

Dickey-Fuller 검정은 $H_0: \phi = 1, H_1: |\phi| < 1$ 을 검정하는 문제로 아래 식에서 $\gamma = \phi - 1 = 0$ 인지를 검정하는 문제다.

OLS 추정을 통해 γ 를 추정하고 $DF = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$ 값을 구해 검정통계량과 비교해 stationary 여부를 판단한다.

$$Z_t = \phi Z_{t-1} + a_t$$

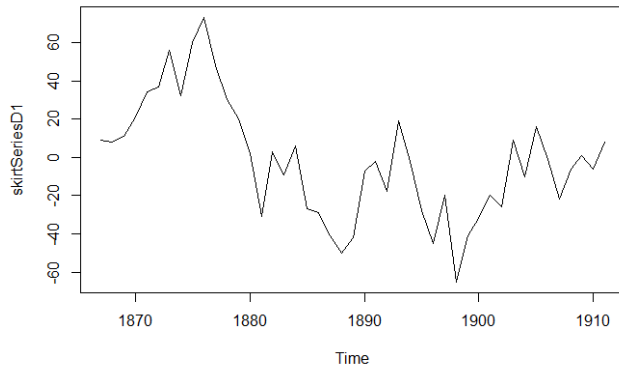
$$Z_t - Z_{t-1} = \phi Z_{t-1} - Z_{t-1} + a_t = (\phi - 1)Z_{t-1} + a_t$$

$$H_0: \gamma = 0, H_1: \gamma < 0$$

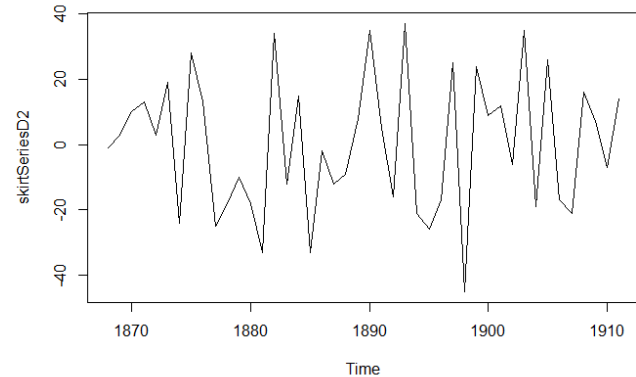
$$DF = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$$

ARIMA Model

Difference=1



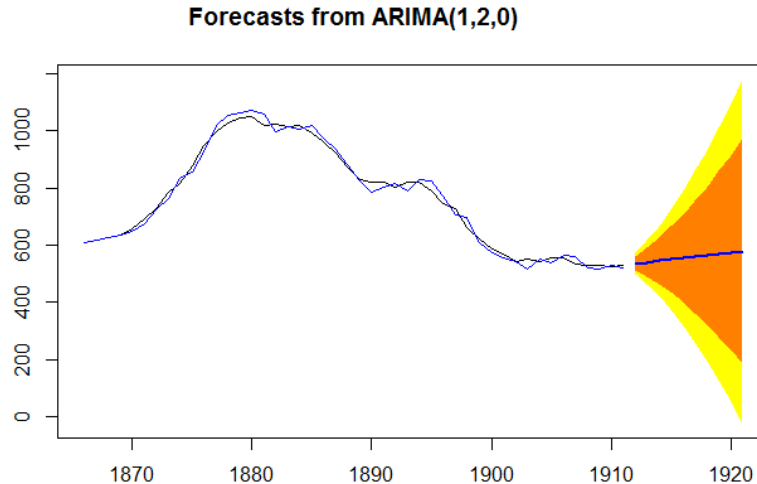
Difference=2



3차 차분의 결과는 어느 정도 Stationary Process로 간주할 수 있다.
그러므로, ARIMA(p,3,q) 모형을 사용할 수 있다. p, q는?

```
library(forecast)
fit<-auto.arima(skirtSeries)
forecast(fit, level=c(95), h=8)
plot(forecast(fit), shadecols="oldstyle")
lines(fitted(fit),col="blue")
```

ARIMA Model



auto.arima는 여러가지, p , q 를 대입해 예측오차를 계산한 결과, ARIMA(1,2,0)을 최적의 모형으로 선정하였다. auto.arima는 AIC 최소화하는 모형을 자동으로 선정한다.

ARIMA Model

Series: skirtSeries ARIMA(1,2,0)

Coefficients:

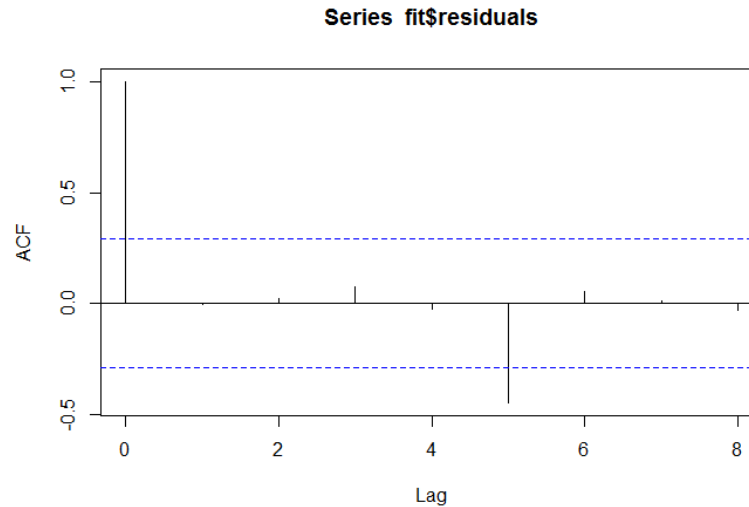
ar1 -0.2997

s.e. 0.1424

sigma^2 estimated as 388.7:

log likelihood=-193.66 AIC=391.33 AICc=391.62 BIC=394.9

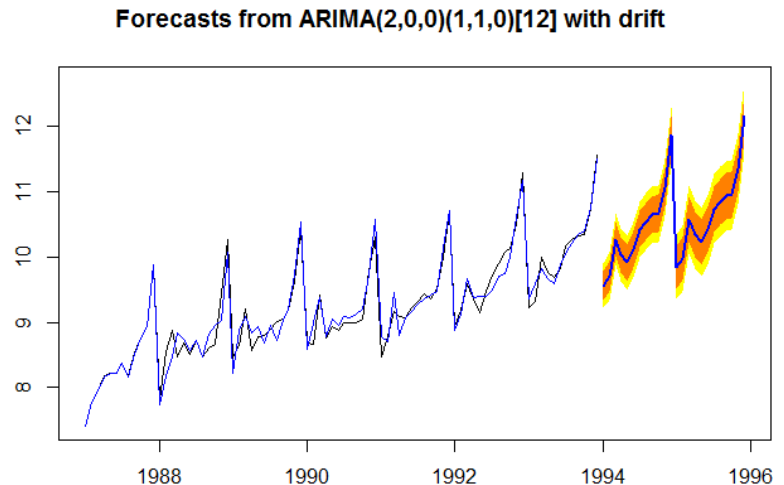
```
acf(fit$residuals, lag.max=8)
```



ARIMA Model

```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirTS <- ts(souvenir, frequency=12, start=c(1987,1))
plot.ts(souvenirTS)
logSouvenirTS <- log(souvenirTS)
plot.ts(logSouvenirTS)
fit<-auto.arima(logSouvenirTS)
forecast(fit, level=c(95), h=8)
plot(forecast(fit), shadedcols="oldstyle")
lines(fitted(fit),col="blue")
```

auto.arima 가 자동으로
계절모형을
고려해서 예측해준다.



ARIMA Model

auto.arima 대신 수동으로 직접 $(p,d,q)(P,D,Q)_s$ 를 지정해서 모형을 적합할 수 있다.
아래의 코드는 ARIMA(1,0,0)(1,1,0)₁₂ 로 적합한 예이다.

```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirTS <- ts(souvenir, frequency=12, start=c(1987,1))
plot.ts(souvenirTS)
logSouvenirTS <- log(souvenirTS)
ARIMAForecast<-Arima(logSouvenirTS, order=c(1,0,0), seasonal=c(1,1,0))
forecast(ARIMAForecast, level=c(95), h=8)
plot(forecast(ARIMAForecast), shadecols="oldstyle")
lines(fitted(ARIMAForecast),col="blue")
```

```
Series: logSouvenirTS ARIMA(1,0,0)(1,1,0)[12]
Coefficients: ar1      sar1
              0.8579 -0.1868
              s.e. 0.0663  0.1434
sigma^2 estimated as 0.04255:
log likelihood=10.64 AIC=-15.29 AICc=-14.94 BIC=-8.46
```

ARIMAX

ARIMAX는 독립변수(외생변수)가 존재하는 경우의 ARIMA 모형이다.

일반적으로 여러 개의 시계열 변수가 서로 연관되어 움직이는 경우가 많다.

R에서는 Arima 함수에서 아래와 같은 명령을 통해 사용할 수 있다.

```
fit <- Arima(y, xreg=x, order=c(1,1,0))
```

차분에 의해 회귀식은 아래 식이 되고 상수항은 차분에 의해 사라진다.

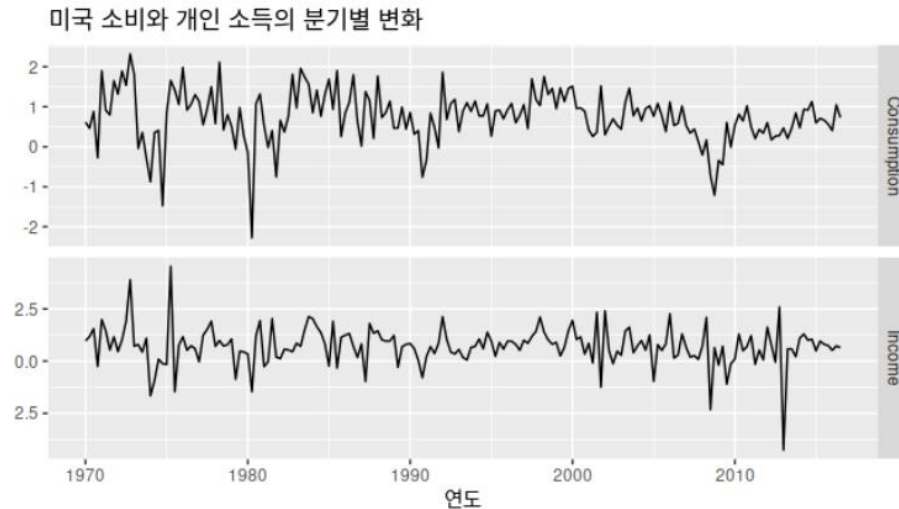
만약, 상수항을 넣으려면 include.drift=TRUE 옵션을 사용해야 한다.

$$y'_t = \beta_1 x'_t + \eta'_t \quad \eta'_t = \phi_1 \eta'_{t-1} + \varepsilon_t \text{는 AR(1) 오차}$$

ARIMAX

아래는 소비액과 개인소득의 분기별 시계열 자료다.

```
library(fpp2)
autoplot(uschange[,1:2], facets=TRUE) +
  xlab("연도") + ylab("") +
  ggtitle("미국 소비와 개인 소득의 분기별 변화")
```



ARIMAX

Income을 X로 두고 소비액을 Y로 회귀한 결과 auto.arima 모형은 arima(1,0,2)를 선택했다.

```
fit <- auto.arima(uschange[, "Consumption"], xreg=uschange[, "Income"])
fit
```

```
Series: uschange[, "Consumption"]
Regression with ARIMA(1,0,2) errors
```

Coefficients:

	ar1	ma1	ma2	intercept	xreg
	0.692	-0.576	0.198	0.599	0.203
s.e.	0.116	0.130	0.076	0.088	0.046

```
sigma^2 estimated as 0.322: log likelihood=-156.9
AIC=325.9 AICc=326.4 BIC=345.3
```

$$y_t = 0.599 + 0.203x_t + \eta_t,$$

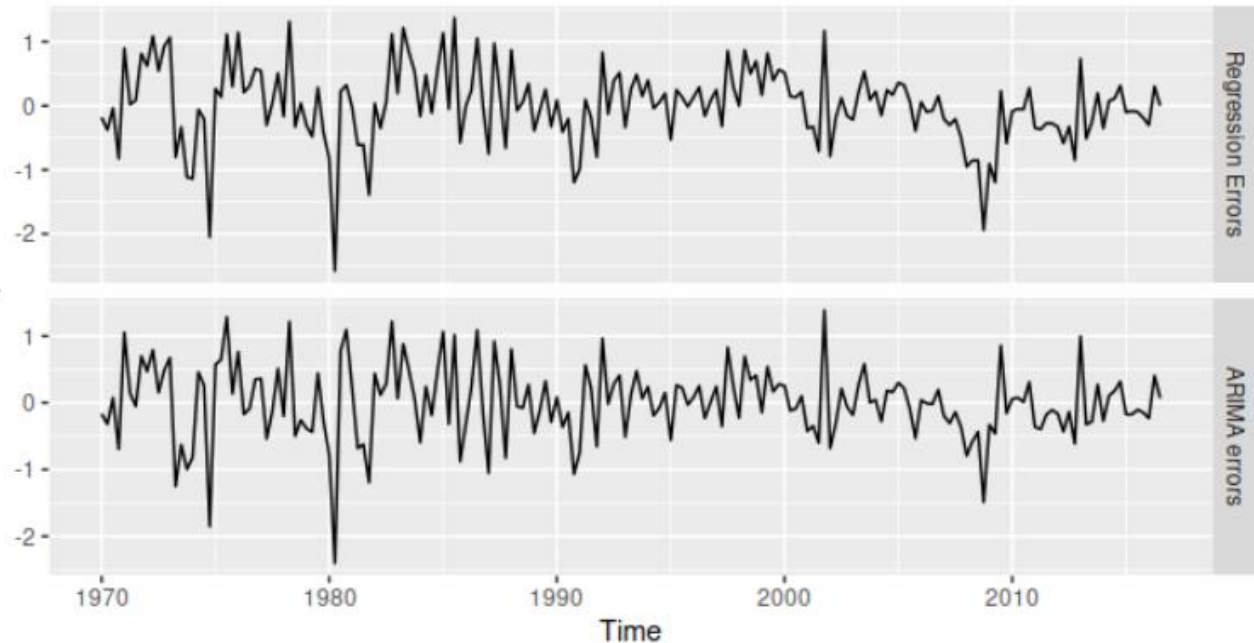
$$\eta_t = 0.692\eta_{t-1} + \varepsilon_t - 0.576\varepsilon_{t-1} + 0.198\varepsilon_{t-2},$$

$$\varepsilon_t \sim \text{NID}(0, 0.322).$$

ARIMAX

잔차 그림이 회귀모형보다 더 i.i.d. 에 가까운 것을 알 수 있다.

```
cbind("Regression Errors" = residuals(fit, type="regression"),  
      "ARIMA errors" = residuals(fit, type="innovation")) %>%  
  autoplot(facets=TRUE)
```

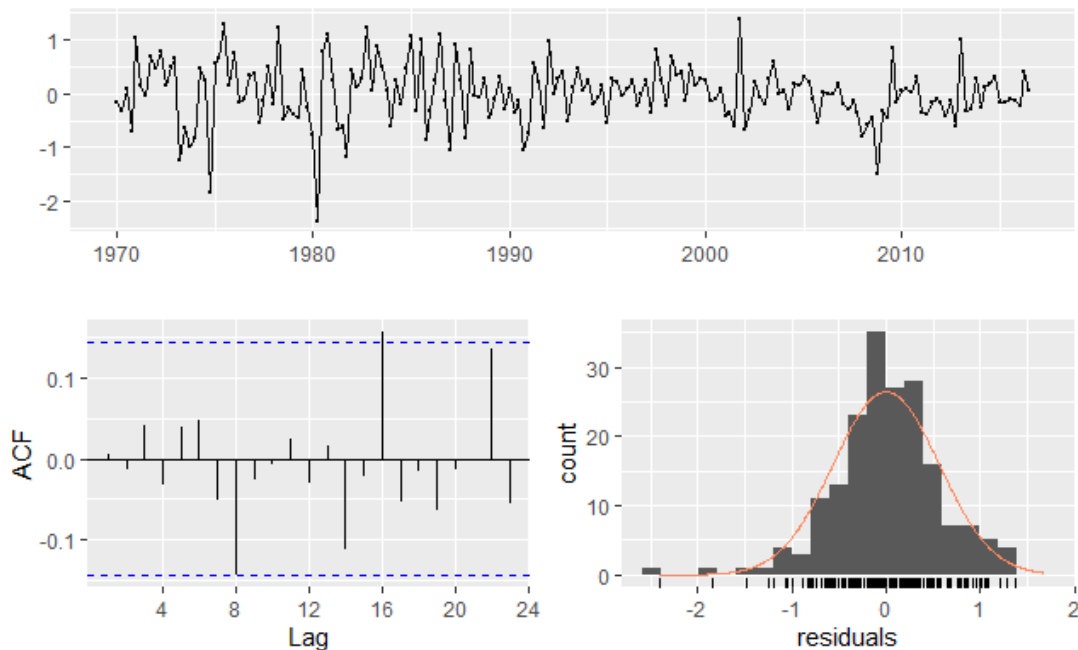


ARIMAX

잔차가 백색잡음의 성질을 만족하는지 알아보자.

`checkresiduals(fit)`

Residuals from Regression with ARIMA(1,0,2) errors



ARCH

주식시장에서 Shock가 발생하면 주가가 폭락하거나 폭등하는 경우 처럼 시계열에서 변동성이 갑자기 커지는 경우가 많이 있다.

이러한 경우, 변동이 특정 시점에 집중되는 변동 집중성(volatility clustering) 현상이 나타난다. 이러한 경우, ARIMA 모형은 등분산 가정을 위배하는 문제가 발생한다.

이러한 경우, 로그 변환 등을 수행하는데 변환 후에도 이런 경향이 남아 있는 경우가 많다.

이 경우, ARIMA 모형을 통과한 잔차가 백색 잡음이 아니라는 것이다.

다시 말해서 이전 오차가 커지면 다음 오차도 커질 가능성이 높다는 것을 모형에 반영하는 것이다.

아래는 t 시점의 오차 분산이 이전 시점 오차의 크기에 영향을 받는다는 가정이다.

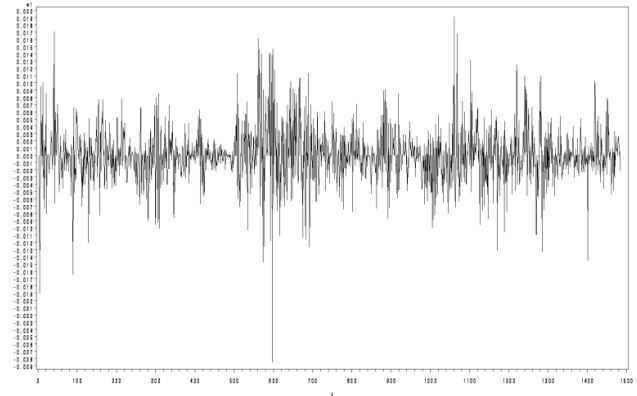
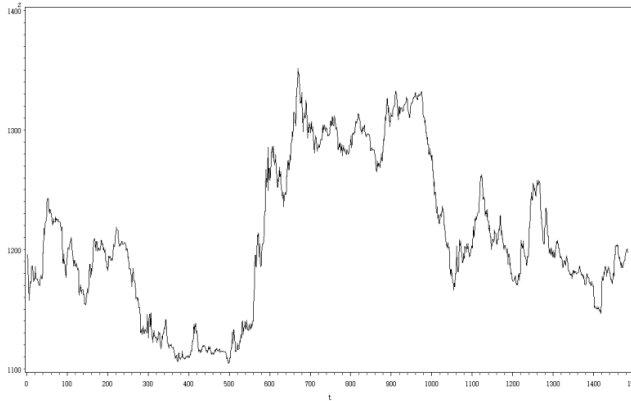
시계열의 예측과 동시에 변동성의 크기를 추정하는 것은 리스크 관리의 중요한 테마다.

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-1}^2 + \cdots + \alpha_p \epsilon_{t-p}^2$$

ARCH

ARCH(AutoRegressive Conditional Heteroskedasticity) 모형은 오차의 분산이 변하는 변동성 모형이다. 원자료에 1차 차분 후에 추세는 거의 제거되었으나 분산이 일정하지 않은 것을 알 수 있다. 분산이 변하는 것은 경제시계열의 경우, Risk를 나타낸다.

아래 식은 ARCH(q) 모형이다.



$$Y_t = \gamma_0 + \gamma_1 X_{1t} + \dots + \gamma_k X_{kt} + e_t, \quad e_t \sim N(0, \sigma_t^2)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + \dots + \alpha_q e_{t-q}^2 \rightarrow \text{ARCH}(q)$$

GARCH

GARCH 모형은 ARCH 모형을 개선한 모형으로 대부분의 경우, GARCH: Generalized ARCH 모형을 사용한다.

ARCH 모형은 q 값을 결정하기 어려운 문제와 q 가 커져 추정해야할 모수가 증가하는 문제점을 가지고 있는 반면 GARCH는 그렇지 않다.

아래 GARCH 모형은 GARCH(1,1) 이다.

GARCH(p, q) 모형은 아래 우측 모형이다.

$$Y_t = \gamma_0 + \gamma_1 X_{1t} + \dots + \gamma_k X_{kt} + e_t, \quad e_t \sim N(0, \sigma_t^2)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + \dots + \alpha_q e_{t-q}^2 \rightarrow \text{ARCH}(q)$$



$$Y_t = \gamma_0 + \gamma_1 X_{1t} + \dots + \gamma_k X_{kt} + e_t,$$

$$\sigma_t^2 = \omega + \alpha e_{t-1}^2 + \beta \sigma_{t-1}^2$$



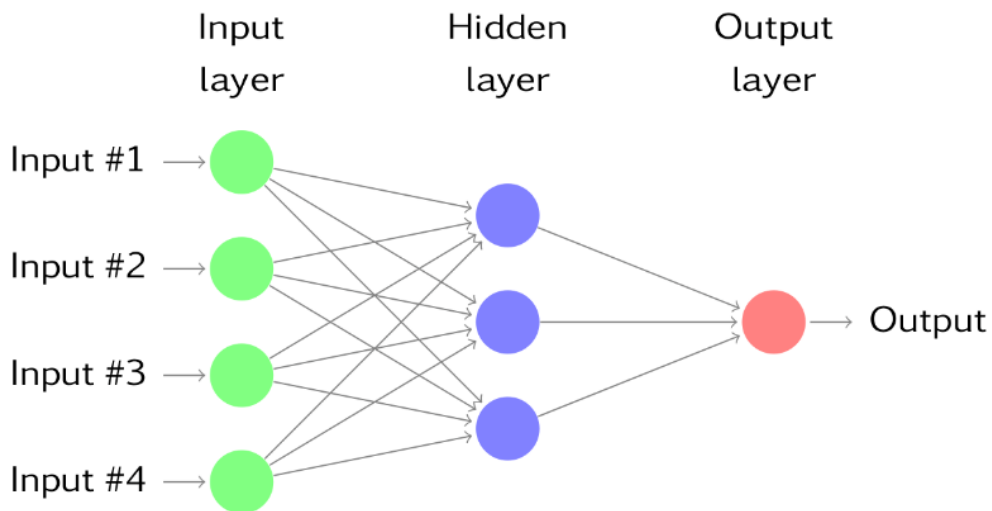
$$\sigma_t^2 = \omega + \sum_{j=1}^p \beta_j \sigma_{t-j}^2 + \sum_{i=1}^q \alpha_i e_{t-i}^2$$

p : 는 GARCH 항의 차수, q : ARCH 항의 차수

NNAR

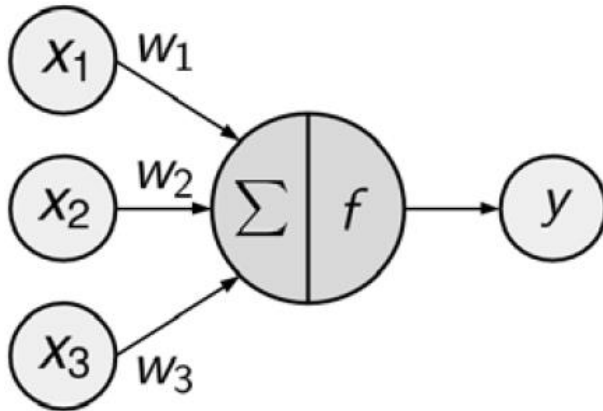
NNAR(Neural Network Autoregressive) 은 AR 구조를 신경망으로 학습하는 모델을 말한다.
아래는 4개의 입력이 들어와서 3개의 히든노드를 통과해 최종 결과가 나오는 모형으로
NNAR(4,3) 모형이다.

NNAR(3,1,2)는 $y_{t-1}, y_{t-2}, y_{t-3}, y_{t-12}$ 가 입력이고 2개의 히든노드를 통과하는 모형을
의미한다.

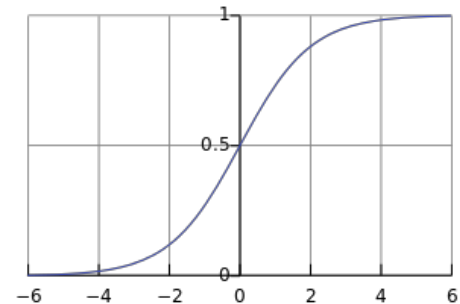


NNAR

신경망 모형은 아래와 같은 기본 구조를 가진다.



$$0 < Y = \frac{1}{1 + e^{-x}} < 1$$



$$\begin{aligned} Y &= S(w_1x_1 + w_2x_2 + w_3x_3 + b) \\ &= \frac{1}{1 + \exp(-(w_1x_1 + w_2x_2 + w_3x_3 + b))} \end{aligned}$$

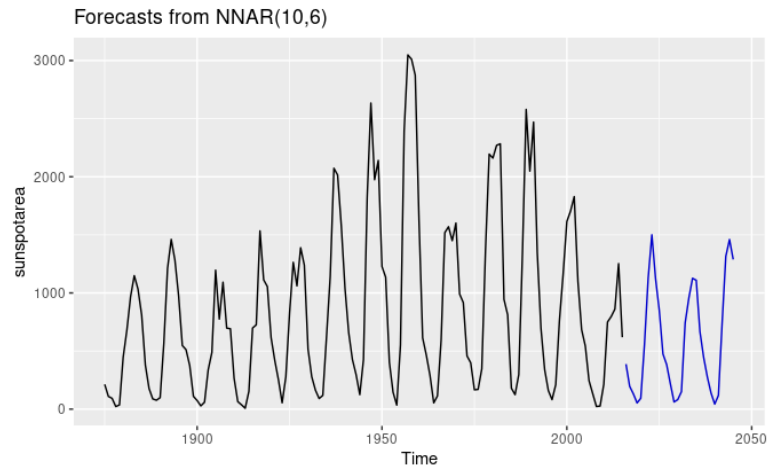
NNAR

AR 모형과 다른 점은 입력과 가중치의 선형 결합이 시그모이드 함수를 통과한 후, 출력이 만들어 진다는 것이다.

이 때문에 AR 모형이 선형인데 NNAR 모형이 비선형 모형이 된다.

또한, 이를 확장하여 히든레이어를 더 많이 만들게 되면 더 복잡한 비선형 모형을 만들 수 있다.

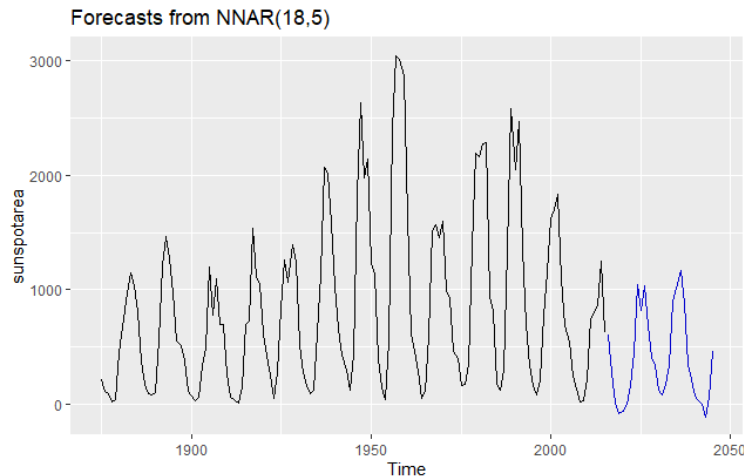
```
fit1 <- nnetar(sunspotarea)  
autoplot(forecast(fit1,h=30))
```



NNAR

아래와 같이 입력과 히든 노드 수를 조절할 수 있다.

```
fit2 <- nnetar(sunspotarea, p=18, size=5)
autoplot(forecast(fit2,h=30))
```



```
> accuracy(fit1)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.2893599	100.0812	72.2345	-19.979	28.59144	0.2089776	-0.05150467

```
> accuracy(fit2)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.3075482	58.20701	36.62952	-11.06442	15.32944	0.1059708	0.0360303

NNAR

참고로 accuracy 함수 measure는 아래와 같다.

~~$$ME = \frac{\sum_{i=1}^n (e_i)}{n} = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)}{n}$$

(여기서 n 은 데이터 수)~~

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}}$$

(여기서 n 은 데이터 수)

$$MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n}$$

(여기서 n 은 데이터 수)

$$MAPE = \frac{\sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{Y_i} \cdot 100}{n}$$

(여기서 n 은 데이터 수)