クレジット:
UTokyo Online Education 数理手法Ⅶ 2019 北川源四郎

ライセンス:
利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。
http://creativecommons.org/licenses/by-nc-nd/4.0/

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。

東京大学
THE UNIVERSITY OF TOKYO

BY NC ND

時系列解析（９）

－ARMAモデルの最尤推定とトレンドモデル－

東京大学 数理・情報教育研究センター
北川 源四郎

UTokyo Online Education 数理手法VII 2019 北川源四郎 CC BY-NC-ND

概要

1．欠測値の処理（前回の続き）

2．ARMAモデルの状態空間表現

3．ARモデルの最尤推定

4．ARMAモデルの最尤推定

5．Example

6．非定常時系列のモデリング

7．平滑化事前分布・正則化

8．トレンドモデル

9．不等間隔データの平滑化

 CC BY-NC-ND

欠測値の処理

$I_n \equiv$ {時点 1,…, n のうち実際に観測した時点}

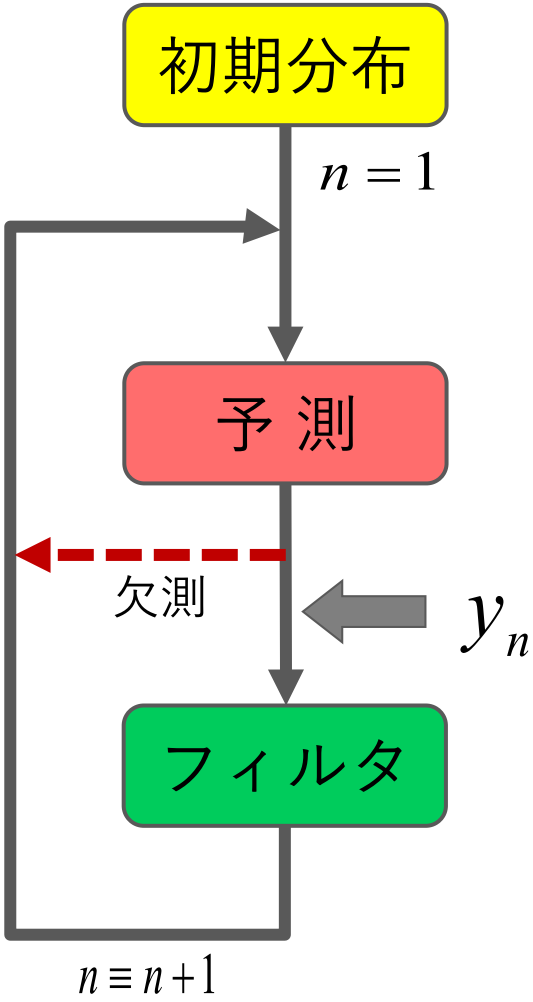
$$Y_n \equiv \{y_i \mid i \in I_n\}$$

- 欠測がない場合

$$I_n = \{1,\ldots,n\}$$
$$Y_n = \{y_i \mid i = 1,\ldots,n\}$$

- $y_{m+1},\ldots,y_{m+k}$ が欠測の場合

$$Y_m = Y_{m+1} = \cdots = Y_{m+k}$$



欠測値がある場合の対数尤度

$I_n = \{ j :$ 時刻 n までに実際に時系列が観測された時点 $j \}$

$$Y_n \equiv \{ y_i \mid i \in I_n \}$$

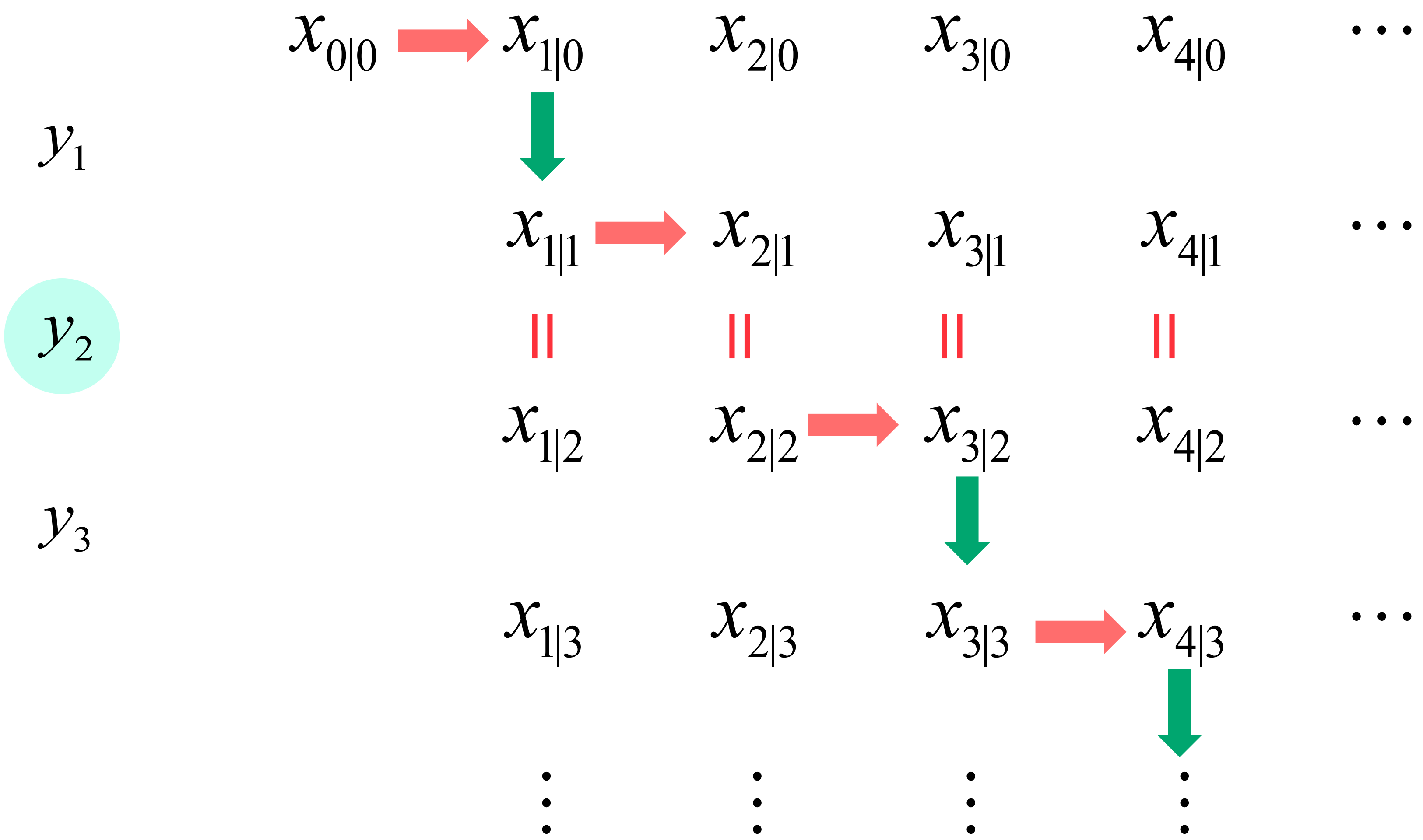
$$\begin{aligned} \ell(\theta) &= \log p(Y_N \mid \theta) \\ &= \sum_{n \in I_N} \log p(y_n \mid Y_{n-1}, \theta) \end{aligned}$$

$$\ell(\theta) = -\frac{1}{2} \sum_{n \in I_N} \left\{ k \log 2\pi + \log | d_{n|n-1} | + \varepsilon_n^T d_{n|n-1}^{-1} \varepsilon_n \right\}$$

k：時系列の次元

CC BY-NC-ND

欠測値の処理(y_2 が欠測の場合)



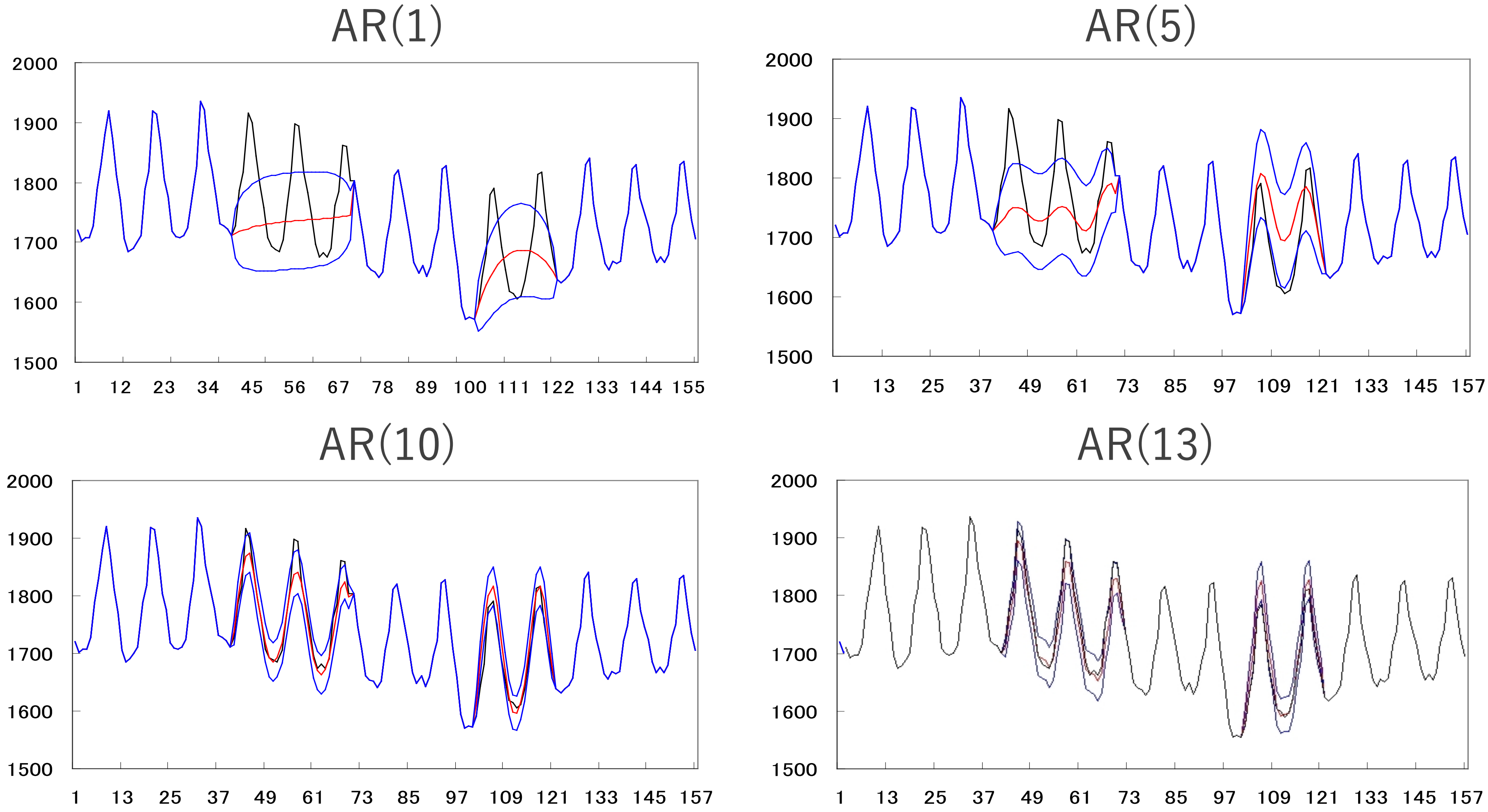
$\rightarrow$ 一期先予測　　$=$ 欠測処理　　$\downarrow$ フィルタ

欠測値の補間

一般に、欠測値の処理は立派な研究課題と考えられているが時系列の場合、状態空間モデルを用いれば（データ数が減少することを除き）原理的には何の問題も生じない.

- フィルタと平滑化により補間値（最適推定値）が得られる
- ただし，モデリングやパラメータ推定においては補間する必要もない.（原理的には補間しない方がよい）
- 同じデータ数の場合，ランダムな欠測がある方が精度が良いこともある.

欠測値の補間



AR(1)
2000
1900
1800
1700
1600
1500
1 12 23 34 45 56 67 78 89 100 111 122 133 144 155
AR(5)
2000
1900
1800
1700
1600
1500
1 13 25 37 49 61 73 85 97 109 121 133 145 157
AR(10)
2000
1900
1800
1700
1600
1500
1 13 25 37 49 61 73 85 97 109 121 133 145 157
AR(13)
2000
1900
1800
1700
1600
1500
1 13 25 37 49 61 73 85 97 109 121 133 145 157

多変量時系列の一部だけが観測された場合

$$x_n = F_n x_{n-1} + G_n v_n$$
$$y_n = H_n x_n + w_n$$

$$y_n = \begin{bmatrix} y_n(1) \\ y_n(2) \end{bmatrix}$$

$y_n(1)$と $y_n(2)$ が同時に欠測となるとは限らない

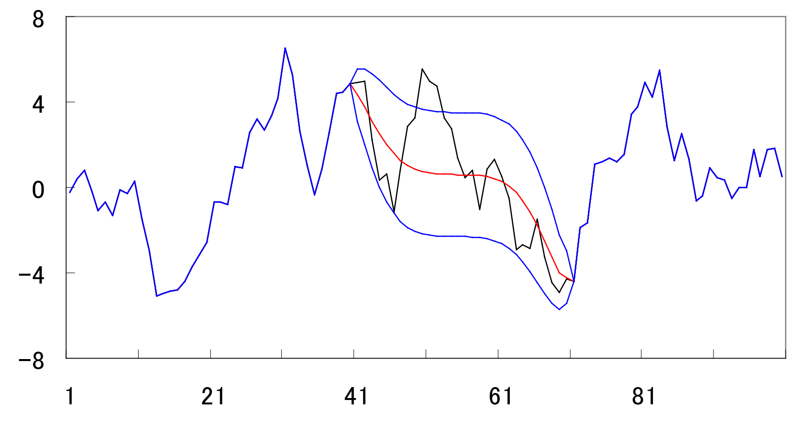
$$H'_n = T_n H_n$$

$$T_n = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \text{両方を観測} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & y_n(2)\text{が欠測} \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & y_n(1)\text{が欠測} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \text{両方欠測} \end{cases}$$

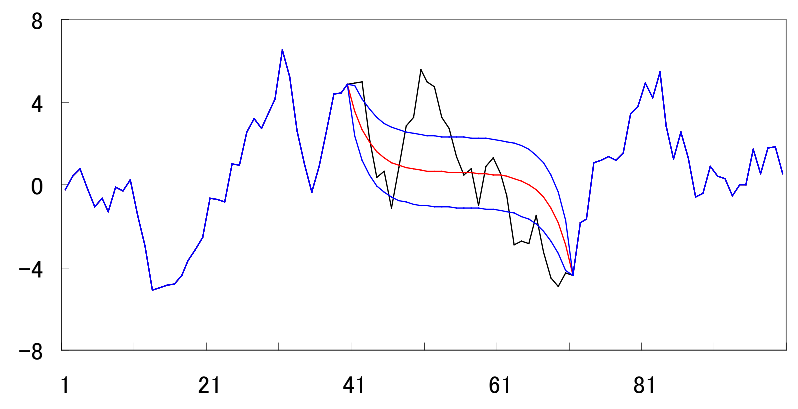
$$H'_n = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \end{cases}$$

1変量補間と2変量補間

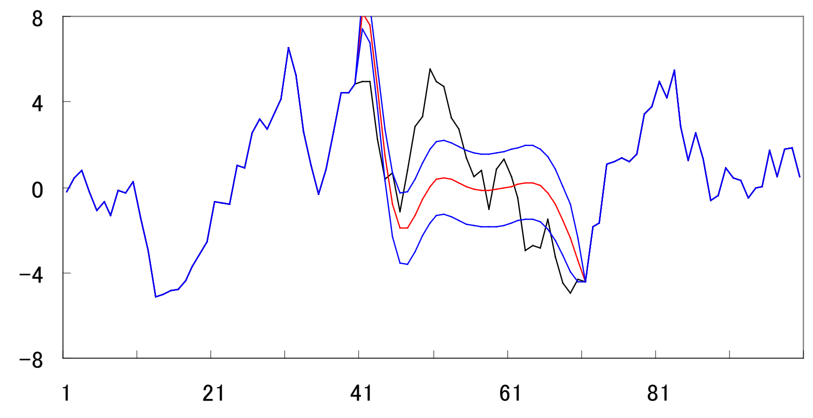
AR(1)



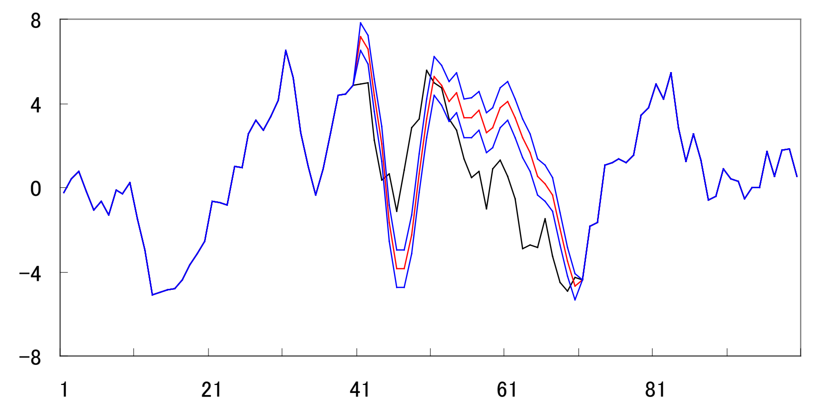
AR(3)



2変量AR(2)

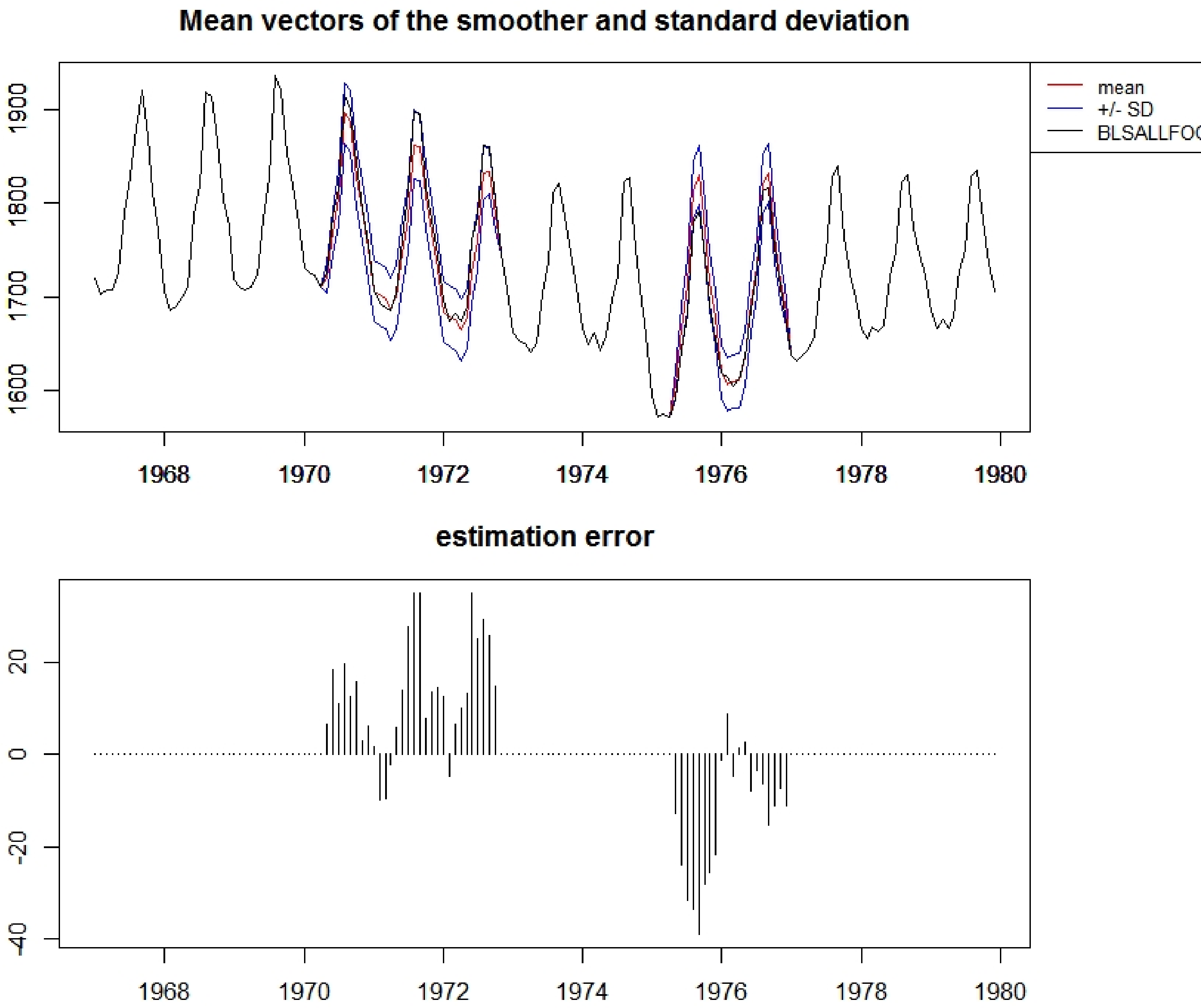


Partially Observed



Rによる欠測値の補間（次数：自動）

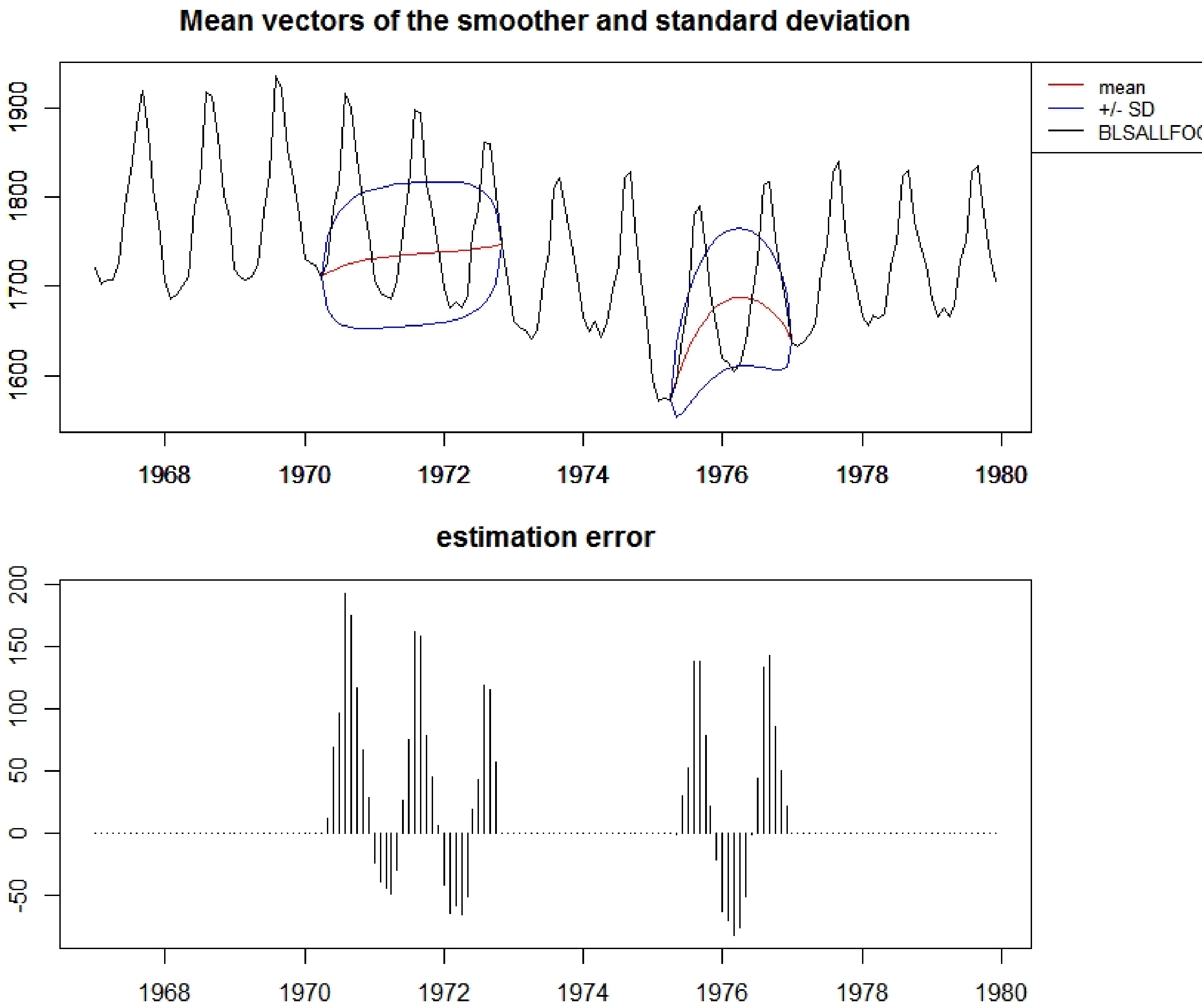
```
#
# ARモデルによる欠測値の補間
#
## AR model (l=1, k=1)     AR次数は自動決定
#
## Example of interpolation of missing values  (AR
model : m=15, k=1)
z2 <- arfit(BLSALLFOOD, plot = FALSE)
tau2 <- z2$sigma2

# m = maice.order, k=1
m2 <- z2$maice.order
arcoef <- z2$arcoef[[m2]]
f <- matrix(0.0e0, m2, m2)
f[1, ] <- arcoef
if (m2 != 1)
  for (i in 2:m2) f[i, i-1] <- 1
g <- c(1, rep(0.0e0, m2-1))
h <- c(1, rep(0.0e0, m2-1))
q <- tau2[m2+1]
r <- 0.0e0
x0 <- rep(0.0e0, m2)
v0 <- NULL

tsmooth(BLSALLFOOD, f, g, h, q, r, x0, v0, missed
= c(41, 101), np = c(30, 20))
```



CC BY-NC-ND

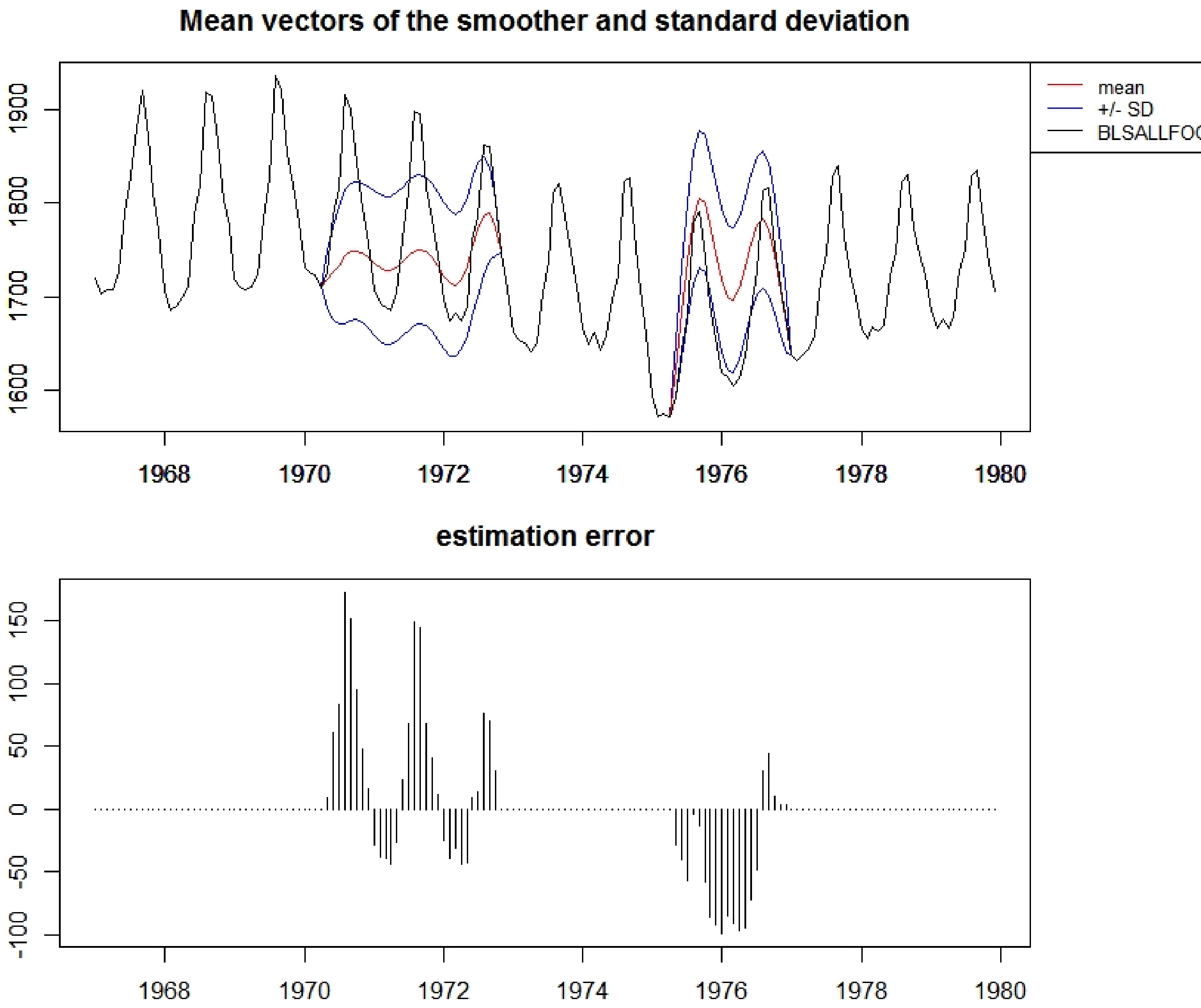
Rによる欠測値の補間（AR次数＝１）

```
# AR order=1
m2 <- 1
arcoef <- z2$arcoef[[m2]]
f <- matrix(0.0e0, m2, m2)
f[1, ] <- arcoef
if (m2 != 1)
  for (i in 2:m2) f[i, i-1] <- 1
g <- c(1, rep(0.0e0, m2-1))
h <- c(1, rep(0.0e0, m2-1))
q <- tau2[m2+1]
r <- 0.0e0
x0 <- rep(0.0e0, m2)
v0 <- NULL

tsmooth(BLSALLFOOD, f, g, h, q, r, x0, v0,
missed = c(41, 101), np = c(30, 20))
```



Rによる欠測値の補間（AR次数＝5）

```
# AR order=5
m2 <- 5
arcoef <- z2$arcoef[[m2]]
f <- matrix(0.0e0, m2, m2)
f[1, ] <- arcoef
if (m2 != 1)
  for (i in 2:m2) f[i, i-1] <- 1
g <- c(1, rep(0.0e0, m2-1))
h <- c(1, rep(0.0e0, m2-1))
q <- tau2[m2+1]
r <- 0.0e0
x0 <- rep(0.0e0, m2)
v0 <- NULL

tsmooth(BLSALLFOOD, f, g, h, q, r, x0, v0,
missed = c(41, 101), np = c(30, 20))
```



 CC BY-NC-ND

ARMAモデルの最尤推定

- 状態空間モデルの最尤推定値は統一的方法によって求められる。
- ARMAモデルの最尤推定のためには
  - ARMAモデルの状態空間表現
  - 初期分布の設定
  - 対数尤度の計算
  - 数値的最適化

状態空間モデルの初期分布

- 平均非定常モデルの場合： $c\,I_k$ を初期分散共分散行列とすることがある(c：適当な大きな値)．
- 定常モデルの場合： 厳密な最尤推定値を求めるためには，フィルタの初期分布を以下のようにする．
  - データがないときの分布 $p(x_0 \mid Y_0),\quad Y_0 = \phi$

    ～ 定常状態の平均ベクトルと分散共分散行列
- ベイズモデルの立場で、意図的に異なる事前分布を用いることもある．

以下では２つの場合を考える．

・ARモデルの場合

・ARMAモデルの場合

ARモデルの状態空間表現

状態空間表現（1）

$$F = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_n = \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-m+1} \end{bmatrix}$$

$$H = [1 \quad 0 \quad \cdots \quad 0]$$

状態空間表現（2）

$$F = \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_m & & & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_n = \begin{bmatrix} y_n \\ \tilde{y}_{n+1|n-1} \\ \vdots \\ \tilde{y}_{n-m+1|n-1} \end{bmatrix}$$

$$H = [1 \quad 0 \quad \cdots \quad 0]$$

ARモデルの場合はどちらでもよいが、（2）はARMAモデルに拡張可能

ARモデルの初期状態分布（表現（1））

$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n \qquad v_n \sim N(0, \sigma^2)$$

$$F = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_n = \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-m+1} \end{bmatrix}$$

$$H = [1 \quad 0 \quad \cdots \quad 0]$$

$$x_n = F x_{n-1} + G v_n$$
$$y_n = H x_n$$

$$E[y_n] = \sum_{j=1}^{m} a_j E\left[y_{n-j}\right] + E[v_n] \qquad \mu \equiv E[y_n] \qquad \mu = \sum_{j=1}^{m} a_j \mu + 0 \qquad \mu = 0$$

CC BY-NC-ND

ARモデルの初期状態（表現（1）続き）

$$x_0 = \begin{bmatrix} y_0 & y_{-1} & \cdots & y_{1-m} \end{bmatrix}^T$$

$$E[x_0] = \left(E[y_0], E[y_{-1}], \cdots, E[y_{1-m}]\right)^T = \begin{pmatrix} 0 & 0 & \cdots & 0 \end{pmatrix}^T$$

$$V_{0|0} = \mathrm{Cov}(x_0, x_0) = E[x_0 x_0^T]$$

$$= E\begin{bmatrix} y_0 y_0 & y_0 y_{-1} & \cdots & y_0 y_{1-m} \\ y_{-1} y_0 & y_{-1} y_{-1} & \cdots & y_{-1} y_{1-m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1-m} y_0 & y_{1-m} y_{-1} & \cdots & y_{1-m} y_{1-m} \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & \cdots & C_{1-m} \\ C_1 & C_0 & \cdots & C_{2-m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1-m} & C_{2-m} & \cdots & C_0 \end{bmatrix}$$

$$\boxed{a_1, \ldots, a_m, \hat{\sigma}^2 \Rightarrow C_0, C_1, \ldots, C_{m-1} \Rightarrow V_{0|0}}$$

 CC BY-NC-ND

ARモデルの初期状態分布（表現（２））

$$x_0 = \begin{bmatrix} y_0 & \tilde{y}_{1|-1} & \cdots & \tilde{y}_{m-1|-1} \end{bmatrix}^T$$

$$\tilde{y}_{n+i|n-1} = \sum_{j=i+1}^{m} a_j y_{n+i-j}$$

$$E[x_0] = \begin{pmatrix} 0 & 0 & \cdots & 0 \end{pmatrix}^T$$

$$V_{0|0} = \mathrm{Cov}(x_0, x_0) = (V_{ij})$$

$$V_{11} = \mathrm{E}(y_0, y_0) = C_0$$

$$V_{1i} = V_{i1} = E(y_0 \tilde{y}_{i-1|-1})$$

$$= E\Big(y_0 \sum_{j=i}^{m} a_j y_{i-1-j}\Big) = \sum_{j=i}^{m} a_j C_{j+1-i}$$

$$V_{ij} = E(\tilde{y}_{i-1|-1} \tilde{y}_{j-1|-1})$$

$$= \sum_{p=i}^{m} \sum_{q=j}^{m} a_p a_q E(y_{i-1-p} y_{j-1-q})$$

$$= \sum_{p=i}^{m} \sum_{q=j}^{m} a_p a_q C_{q-j-p+i}$$

$$V_{0|0} = \begin{bmatrix} C_0 & \sum_{q=2}^{2} a_q C_{q-1} & \cdots & a_m C_1 \\ \sum_{p=2}^{m} a_p C_{p-1} & \sum_{p=2}^{m}\sum_{q=2}^{m} a_p a_q C_{q-p} & \cdots & \sum_{p=2}^{m}\sum_{q=m}^{m} a_p a_q C_{2-p} \\ \vdots & \vdots & \ddots & \vdots \\ a_m C_1 & \sum_{p=m}^{m}\sum_{q=2}^{m} a_p a_q C_{q-2} & \cdots & a_m a_m C_0 \end{bmatrix}$$

ARモデルの初期状態分布（表現（2）続き）

$$x_0 = \begin{bmatrix} y_0 \\ \tilde{y}_{1|-1} \\ \vdots \\ \tilde{y}_{m-1|-1} \end{bmatrix} = T \begin{bmatrix} y_0 \\ y_{-1} \\ \vdots \\ y_{1-m} \end{bmatrix} \equiv T z_0, \qquad T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & a_m \\ \vdots & \vdots & \iddots & 0 \\ 0 & a_m & 0 & 0 \end{bmatrix}$$

$$E\left[x_0 x_0^T\right] = E\left[T z_0 z_0^T T^T\right] = TCT^T$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & a_m \\ \vdots & \vdots & \iddots & 0 \\ 0 & a_m & 0 & 0 \end{bmatrix} \begin{bmatrix} C_0 & C_1 & \cdots & C_{1-m} \\ C_1 & C_0 & \cdots & C_{2-m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1-m} & C_{2-m} & \cdots & C_0 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & a_m \\ \vdots & \vdots & \iddots & 0 \\ 0 & a_m & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} C_0 & C_1 & \cdots & C_{1-m} \\ \sum_{p=2}^{m} a_p C_{p-1} & \sum_{p=2}^{m} a_p C_{p-2} & \cdots & \sum_{p=2}^{m} a_p C_{p-m} \\ \vdots & \vdots & \ddots & \vdots \\ a_m C_1 & a_m C_0 & \cdots & a_m C_{2-m} \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & a_m \\ \vdots & \vdots & \iddots & 0 \\ 0 & a_m & 0 & 0 \end{bmatrix} = \begin{bmatrix} C_0 & \sum_{q=2}^{m} a_q C_{q-1} & \cdots & a_m C_1 \\ \sum_{p=2}^{m} a_p C_{p-1} & \sum_{p=2}^{m}\sum_{q=2}^{m} a_p a_q C_{q-p} & \cdots & \sum_{p=2}^{m}\sum_{q=m}^{m} a_p a_q C_{2-p} \\ \vdots & \vdots & \ddots & \vdots \\ a_m C_1 & \sum_{p=m}^{m}\sum_{q=2}^{m} a_p a_q C_{q-2} & \cdots & a_m a_m C_0 \end{bmatrix}$$

ARMAモデルの状態空間表現

$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n - \sum_{j=1}^{\ell} b_j v_{n-j} \qquad v_n \sim N(0, \sigma^2)$$

$$k = \max(m, l+1)$$

$$F = \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_k & & & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ -b_1 \\ \vdots \\ -b_{k-1} \end{bmatrix}, \quad x_n = \begin{bmatrix} y_n \\ \tilde{y}_{n+1|n-1} \\ \\ \tilde{y}_{n+k-1|n-1} \end{bmatrix}$$

$$H = [1 \quad 0 \quad \cdots \quad 0]$$

$$x_n = F x_{n-1} + G v_n$$
$$y_n = H x_n$$

$$\tilde{y}_{n+i|n-1} = \sum_{\underline{j=i+1}}^{m} a_j y_{n+i-j} - \sum_{\underline{j=i}}^{\ell} b_j v_{n+i-j}$$

CC BY-NC-ND

ARMAモデルの状態空間表現の確認

$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n - \sum_{j=1}^{\ell} b_j v_{n-j}$$
$$\tilde{y}_{n+i|n-1} = \sum_{j=i+1}^{m} a_j y_{n+i-j} - \sum_{j=i}^{\ell} b_j v_{n+i-j}$$

$$x_n = \begin{bmatrix} y_n \\ \tilde{y}_{n+1|n-1} \\ \\ \tilde{y}_{n+k-1|n-1} \end{bmatrix}$$

$$\begin{aligned}
y_n &= a_1 y_{n-1} + \boxed{\sum_{j=2}^{m} a_j y_{n-j} - \sum_{j=1}^{\ell} b_j v_{n-j}} + v_n \\
&= a_1 y_{n-1} + y_{n|n-2} + v_n \\
\tilde{y}_{n+1|n-1} &= \sum_{j=2}^{m} a_j y_{n+1-j} - \sum_{j=1}^{\ell} b_j v_{n+1-j} \\
&= a_2 y_{n-1} + \boxed{\sum_{j=3}^{m} a_j y_{n+1-j} - \sum_{j=2}^{\ell} b_j v_{n+1-j}} - b_1 v_n \\
&= a_2 y_{n-1} + y_{n+1|n-2} - b_1 v_n \\
&\vdots \\
\tilde{y}_{n+k-1|n-1} &= a_k y_{n-1} - b_{k-1} v_n
\end{aligned}$$

$$F = \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_k & & & \end{bmatrix}$$

$$G = \begin{bmatrix} 1 \\ -b_1 \\ \vdots \\ -b_{k-1} \end{bmatrix}$$

ARMAモデルの初期状態分布

$$\boxed{\tilde{y}_{n+i|n-1} = \sum_{j=i+1}^{m} a_j y_{n+i-j} - \sum_{j=i}^{\ell} b_j v_{n+i-j}}$$

$$k = \max(m, l+1)$$

$$x_0 = \begin{bmatrix} y_0 \\ \tilde{y}_{1|-1} \\ \vdots \\ \tilde{y}_{k-1|-1} \end{bmatrix}, \qquad E[x_0] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$V_{0|0} = \mathrm{Cov}(x_0, x_0)$$

$$= \left[\begin{array}{c|cccc} V_{11} & V_{12} & \cdots & V_{1k} \\ \hline V_{21} & V_{22} & \cdots & V_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ V_{k1} & V_{k2} & \cdots & V_{kk} \end{array}\right]$$

$$V_{11} = E(y_0, y_0) = C_0$$

$$V_{1i} = V_{i1} = E(y_0 \tilde{y}_{i-1|-1})$$

$$= E\{y_0 (\sum_{j=i}^{m} a_j y_{i-1-j} - \sum_{j=i-1}^{\ell} b_j v_{i-1-j})\}$$

$$= \sum_{j=i}^{m} a_j C_{j+1-i} - \sum_{j=i-1}^{\ell} b_j g_{j+1-i}$$

$$V_{ij} = E(\tilde{y}_{i-1|-1} \tilde{y}_{j-1|-1})$$

$$= \sum_{p=i}^{m} \sum_{q=j}^{m} a_p a_q E(y_{i-1-p} y_{j-1-q}) - \sum_{p=i}^{m} \sum_{q=j-1}^{\ell} a_p b_q E(y_{i-1-p} v_{j-1-q})$$

$$- \sum_{p=i-1}^{\ell} \sum_{q=j}^{m} b_p a_q E(v_{i-1-p} y_{j-1-q}) + \sum_{p=i-1}^{\ell} \sum_{q=j-1}^{\ell} b_p b_q E(v_{i-1-p} v_{j-1-q})$$

$$= \sum_{p=i}^{m} \sum_{q=j}^{m} a_p a_q C_{q-j-p+i} - \sum_{p=i}^{m} \sum_{q=j-1}^{\ell} a_p b_q g_{q-j-p+i}$$

$$- \sum_{p=i-1}^{\ell} \sum_{q=j}^{m} b_p a_q g_{p-i-q+j} + \sum_{p=i-1}^{\ell} b_p b_{q+j-i} \sigma^2$$

 CC BY-NC-ND

ARMAモデルのパラメータ推定

パラメータ　$\theta = (\sigma^2, a_1, \ldots, a_m, b_1, \ldots, b_\ell)^T$

対数尤度　$\ell(\theta) = -\frac{1}{2}\left\{ N \log 2\pi + \sum_{n=1}^{N} \log r_n + \sum_{n=1}^{N} \frac{\varepsilon_n^2}{r_n} \right\}$

$$\varepsilon_n = y_{n|n-1} - Hx_{n|n-1}, \qquad r_n = HV_{n|n-1}H^T$$

最尤推定値　$\max_{\theta} \ell(\theta) \Rightarrow \hat{\theta}$

最適化では

$$(a_1, \ldots, a_m)^T \Leftrightarrow (p_1, \ldots, p_m)^T$$
$$(b_1, \ldots, b_\ell)^T \Leftrightarrow (q_1, \ldots, q_\ell)^T$$

PARCOR

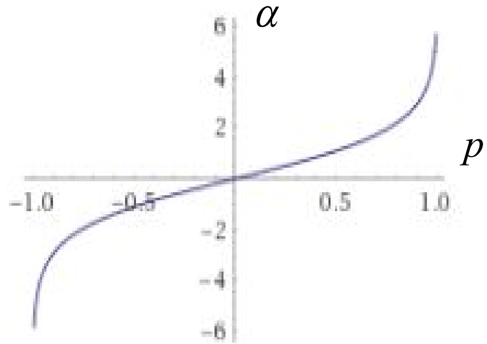
$$a_j^m = a_j^{m-1} - a_m^m a_{m-j}^{m-1}, \qquad p_m \equiv a_m^m$$
$$b_j^\ell = b_j^{\ell-1} - b_\ell^\ell b_{\ell-j}^{\ell-1}, \qquad q_\ell \equiv b_\ell^\ell$$

パラメータ推定における変数変換

$$\alpha_j = \log\left(\frac{1+p_j}{1-p_j}\right), \qquad \beta_j = \log\left(\frac{1+q_j}{1-q_j}\right)$$

$$-1 < p_j < 1, \quad -1 < q_j < 1 \qquad \Rightarrow -\infty < \alpha_j < \infty, \quad -\infty < \beta_j < \infty$$

$$\theta = (\sigma^2, a_1, \ldots, a_m, b_1, \ldots, b_\ell)^T \Rightarrow \theta' = (\sigma^2, p_1, \ldots, p_m, q_1, \ldots, q_\ell)^T$$
$$\Rightarrow \theta'' = (\log \sigma^2, \alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_\ell)^T$$



逆に $\quad p_j = \dfrac{e^{\alpha_j}-1}{e^{\alpha_j}+1}, \qquad q_j = \dfrac{e^{\beta_j}-1}{e^{\beta_j}+1},$

$$\Rightarrow -1 < p_j < 1, \quad -1 < q_j < 1$$

実際は $-c < p_j < c, \quad -d < q_j < d \quad |c|<1, \ |d|<1$ とすることが多い

次数選択（m,ℓ）

情報量規準

$$\text{AIC} = -2\ell(\hat{\theta}) + 2(\text{パラメータ数})$$

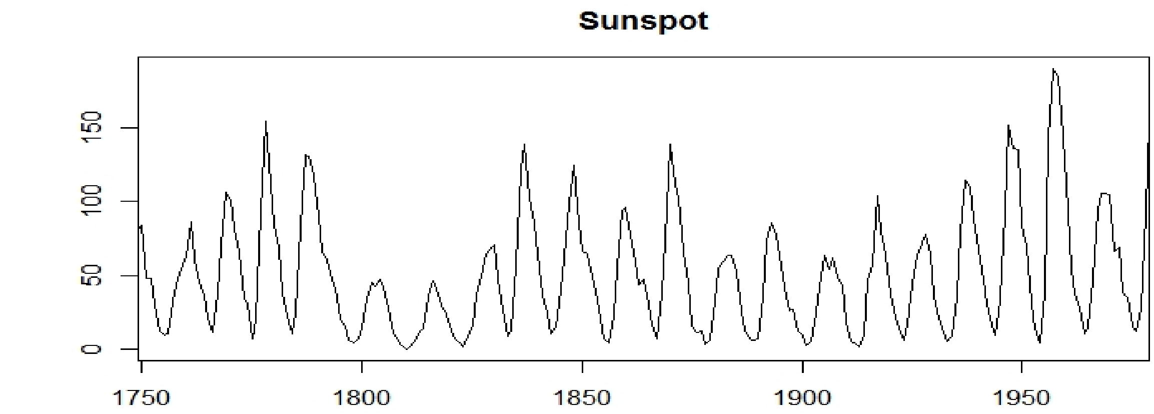
$$= N\log 2\pi + \sum_{n=1}^{N}\log d_{n|n-1} + \sum_{n=1}^{N}\frac{\varepsilon_n^2}{d_{n|n-1}} + 2(m+\ell+1)$$

m　AR次数

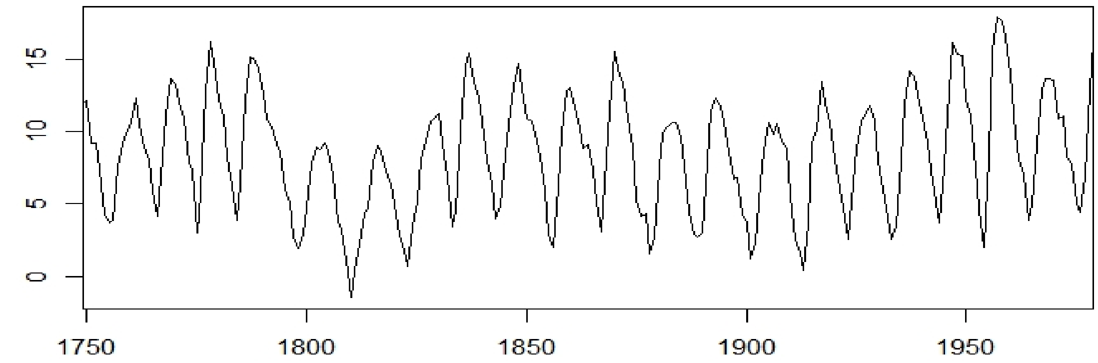
ℓ　MA次数

Sunspot data （Box-Cox変換）

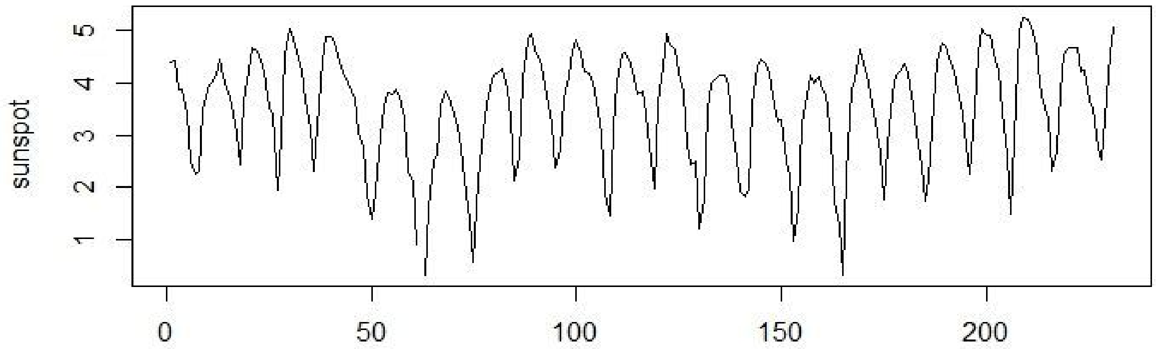
```
# Sunspot number data
data(Sunspot)
#
# Box-Cox transformation
y <- boxcox(Sunspot)
#
# lambda = 0.40
#
y1 <- log10(Sunspot)
```



Transformed data Box-Cox(λ=0.4)



$\log y(n)$

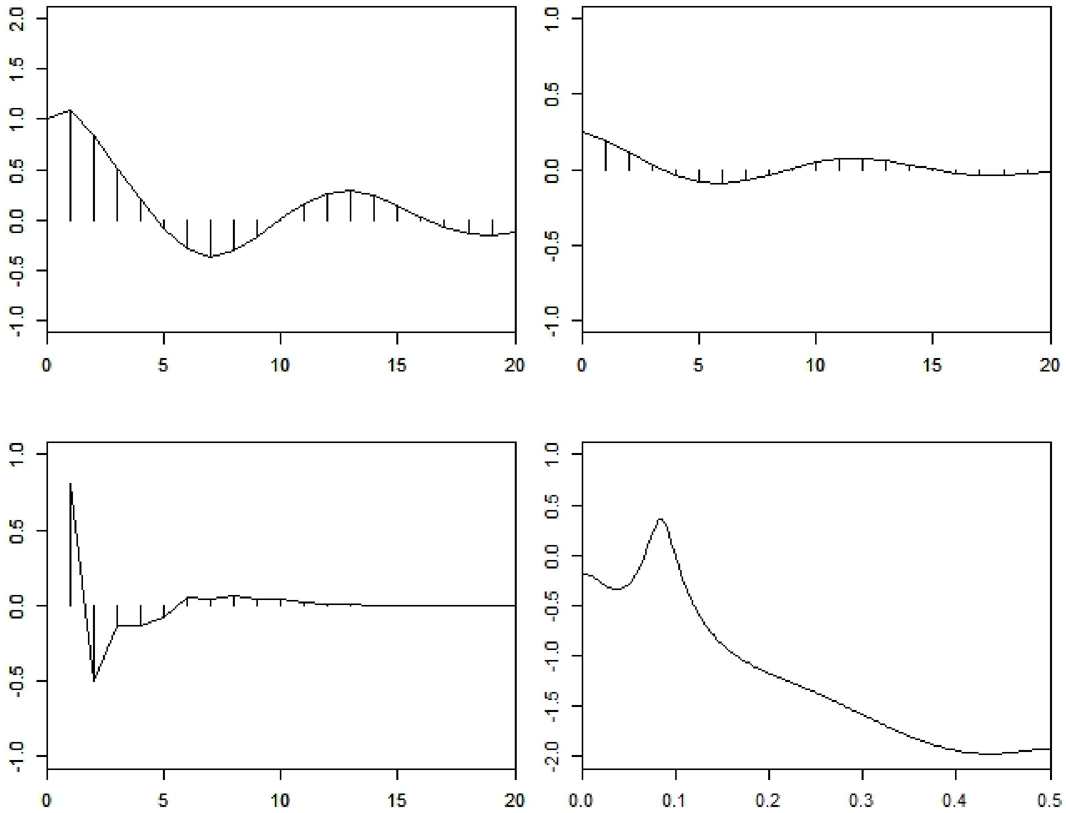


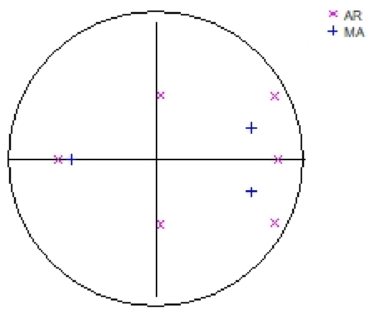
Rによる推定: ARMA(6,3)

```
data(Sunspot)          # Sunspot number data
y <- log10(Sunspot)   # y = log( sunspot data )
z1 <- armafit(y$z, ar.order=6, ma.order=3)
z1
```

```
sigma2              6.28197e-02
log-likelihood     -7.888
aic                 35.775
AR coefficients
   1.789    -0.844    -0.333    0.382    -0.187    0.102
MA coefficients
   0.694    0.283    -0.269
```

```
len <- length(Sunspot)
z <- armaimp(arcoef=z1$arcoef,
macoef=z1$macoef, v=z1$sigma2, n=len,
lag=20)
```



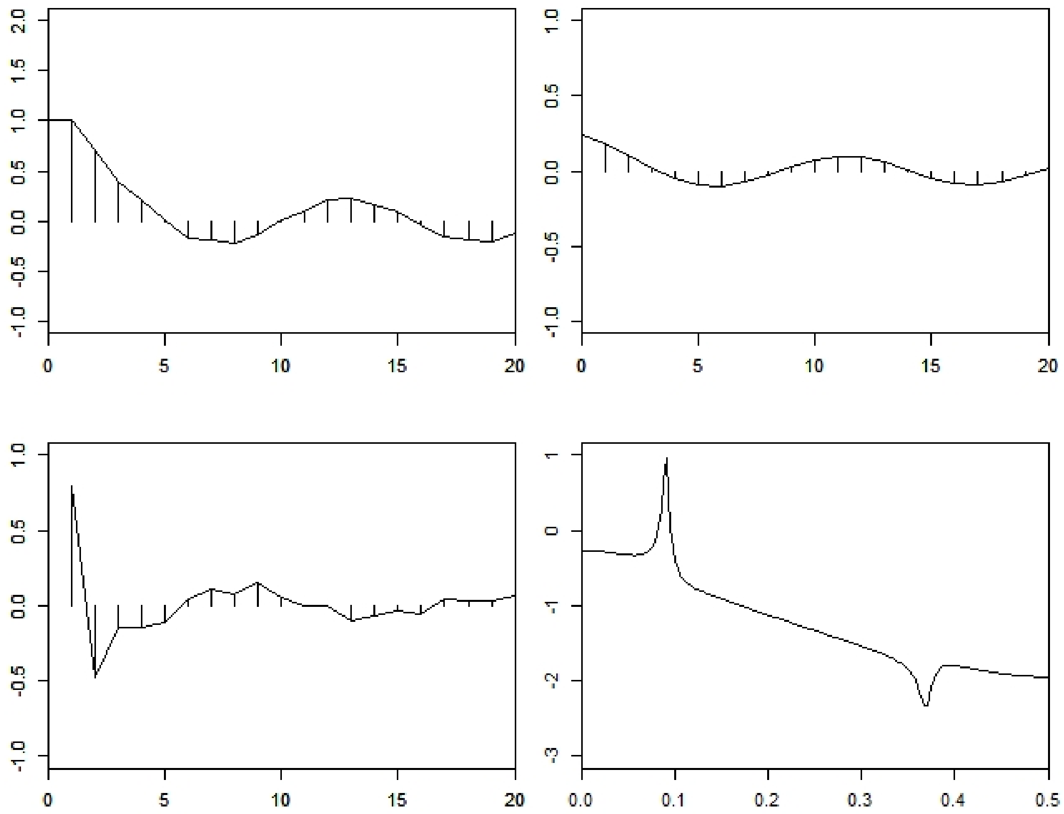


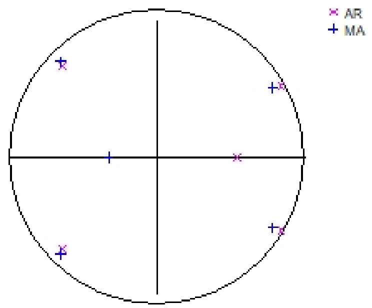
Rによる推定: ARMA(5,5)

```
data(Sunspot)          # Sun spot number data
y <- log10(Sunspot)    # Log transformation
#
z1 <- armafit(y, ar.order=5, ma.order=5)
# AR order=5, MA order=5
z1
```

```
sigma2          5.97581e-02
log-likelihood  -1.300
aic             24.599
AR coefficients
   0.920     0.104     0.017     -0.939     0.451
MA coefficients
   -0.093    0.328     0.387     -0.701     -0.267
```

```
len <- length(Sunspot)
z <- armaimp(arcoef=z1$arcoef, macoef=z1$macoef,
v=z1$sigma2, n=len, lag=20)
```





 CC BY-NC-ND

ARMAモデルのAICと対数尤度

AIC

		MA order					
		0	1	2	3	4	5
AR order	0		158.0	96.5	83.5	80.1	80.3
	1	120.4	87.5	292.7	80.6	81.0	82.4
	2	69.8	64.4	66.4	60.3	49.8	51.3
	3	66.9	66.4	71.0	57.0	51.8	49.1
	4	65.9	67.3	62.3	52.6	53.0	48.3
	5	66.3	68.2	64.5	52.6	47.5	47.2

対数尤度

		MA order					
		0	1	2	3	4	5
AR order	0		-77.0	-45.2	-37.8	-35.1	-34.1
	1	-58.2	-40.7	-142.3	-35.3	-34.5	-34.2
	2	-31.9	-28.2	-28.2	-24.2	-17.9	-17.7
	3	-29.4	-28.2	-29.5	-21.5	-17.9	-15.6
	4	-28.0	-27.6	-24.2	-18.4	-17.5	-14.1
	5	-27.2	-27.1	-23.8	-17.3	-13.7	-12.6

ARMAモデル推定における注意点

- 最適化が収束していても局所解の可能性がある。

$$\mathrm{AIC}(i,j) \le \min\left\{\mathrm{AIC}(i-1,j), \mathrm{AIC}(i,j-1)\right\} + 2$$

- これが成り立たないときは，明らかに局所解。
  ARMA$(i-1,j)$とARMA$(i,j-1)$のうち，AICが小さいモデルの推定値をARMA(i,j)の初期値として再計算

CC BY-NC-ND

```
z1 <- armafit(y, ar.order=1, ma.order=2)
z1
$sigma2    [1] 0.1964677
$lkfood    [1] -142.342
$aic       [1] 292.684
$arcoef    [1] 0.95
$macoef    [1] -0.0475  0.9500
```

AIC(1,1) < AIC(0,2)

```
z1 <- armafit(y, ar.order=1, ma.order=1)
$sigma2    [1] 0.0828955
$lkfood    [1] -40.74151
$aic       [1] 87.48302
$arcoef    [1] 0.6423625
$macoef    [1] -0.3999504
```

```
z1 <- armafit(y, ar.order=1, ma.order=2, ar=c(0.64236), ma=c(-0.39995,0.0))
$sigma2    [1] 0.07960113
$lkfood    [1] -36.13133
$aic       [1] 80.26266
$arcoef    [1] 0.500305
$macoef    [1] -0.5742810 -0.2494204
```

AIC(1,2) < min{AIC(1,1),AIC(0,2)} + 2

```
z1 <- armafit(y, ar.order=3, ma.order=2)
z1
$sigma2    [1] 0.07557714
$lkfood    [1] -29.51644
$aic       [1] 71.03288
$arcoef    [1]  0.2835428  0.5978096 -0.4849157
$macoef    [1] -0.8357265  0.1202879
```

AIC(3,2) > AIC(3,1) = AIC(2,2)

```
z1 <- armafit(y, ar.order=3, ma.order=1)
$sigma2    [1] 0.07426225
$lkfood    [1] -28.18457
$aic       [1] 66.36913
$arcoef    [1]  1.42757798 -0.69842079  0.00998119
$macoef    [1] 0.3974137
```

```
z1 <- armafit(y, ar.order=3, ma.order=2, ar=c(1.42758,-0.69843,0.00998), ma=c(0.3974,0.0))
$sigma2    [1] 0.07381827
$lkfood    [1] -26.95467
$aic       [1] 65.90933
$arcoef    [1]  0.5221791  0.6013328 -0.6154390
$macoef    [1] -0.5281880  0.3931269
```

```
z1 <- armafit(y, ar.order=2, ma.order=2)
$sigma2    [1] 0.07426049
$lkfood    [1] -28.18267
$aic       [1] 66.36534
$arcoef    [1]  1.4129259 -0.6770232
$macoef    [1] 0.381496210 0.008517393
```

```
z1 <- armafit(y, ar.order=3, ma.order=2, ar=c(1.4129,-0.6770,0.0), ma=c(0.38250,0.00852))
$sigma2    [1] 0.07381827
$lkfood    [1] -26.95467
$aic       [1] 65.90933
$arcoef    [1]  0.5221787  0.6013322 -0.6154379
$macoef    [1] -0.5281880  0.3931269
```

ARMAモデルのAICと対数尤度

AIC

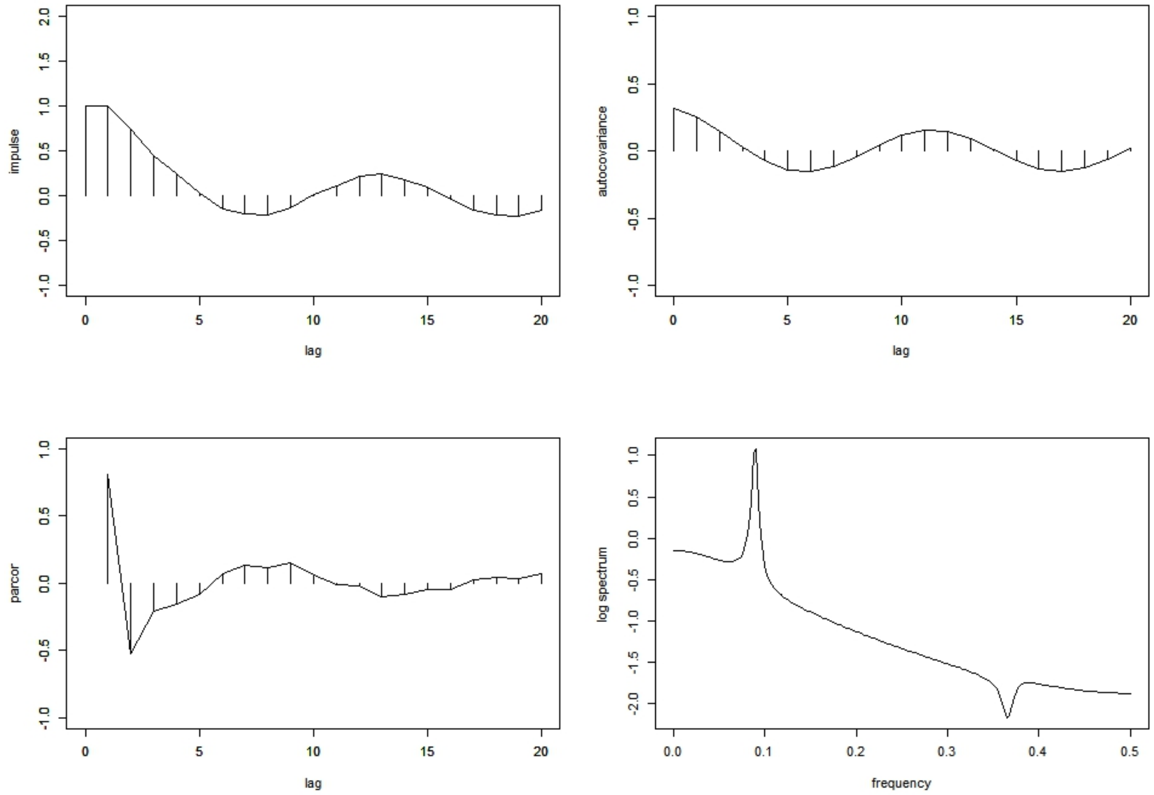
		MA order					
		0	1	2	3	4	5
AR order	0		158.0	96.5	83.5	80.1	80.3
	1	120.4	87.5	80.3	80.6	81.0	82.4
	2	69.8	64.4	66.4	60.3	49.8	51.3
	3	66.9	66.4	65.9	57.0	51.8	49.1
	4	65.9	67.3	62.3	52.6	53.0	48.3
	5	66.3	68.2	64.5	52.6	47.5	47.2

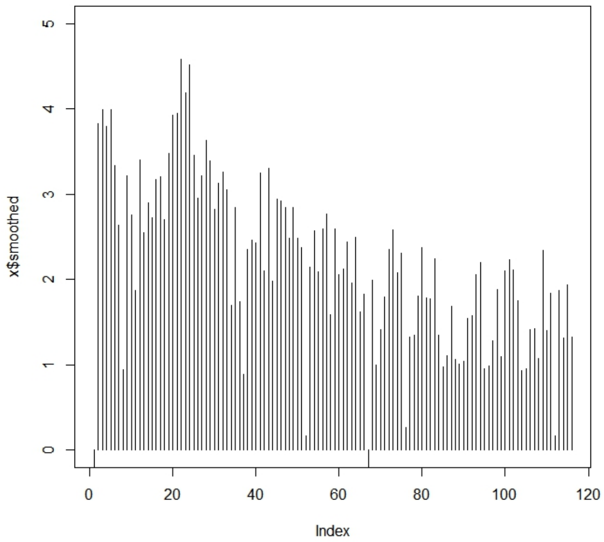
対数尤度

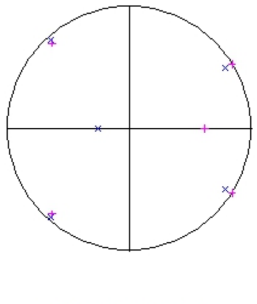
		MA order					
		0	1	2	3	4	5
AR order	0		-77.0	-45.2	-37.8	-35.1	-34.1
	1	-58.2	-40.7	-36.1	-35.3	-34.5	-34.2
	2	-31.9	-28.2	-28.2	-24.2	-17.9	-17.7
	3	-29.4	-28.2	-27.0	-21.5	-17.9	-15.6
	4	-28.0	-27.6	-24.2	-18.4	-17.5	-14.1
	5	-27.2	-27.1	-23.8	-17.3	-13.7	-12.6

ARMA(5,5)

```
z1 <- armafit(y, ar.order=5, ma.order=5)
z1
$sigma2     [1] 0.06715806
$lkfood     [1] -12.58323
$aic        [1] 47.16646
$arcoef     [1]  1.025978205  0.008235232
   0.089720795 -1.029191174  0.538194592
$macoef     [1]  0.01873066  0.29533306
   0.41014808 -0.71579430 -0.20821791
```



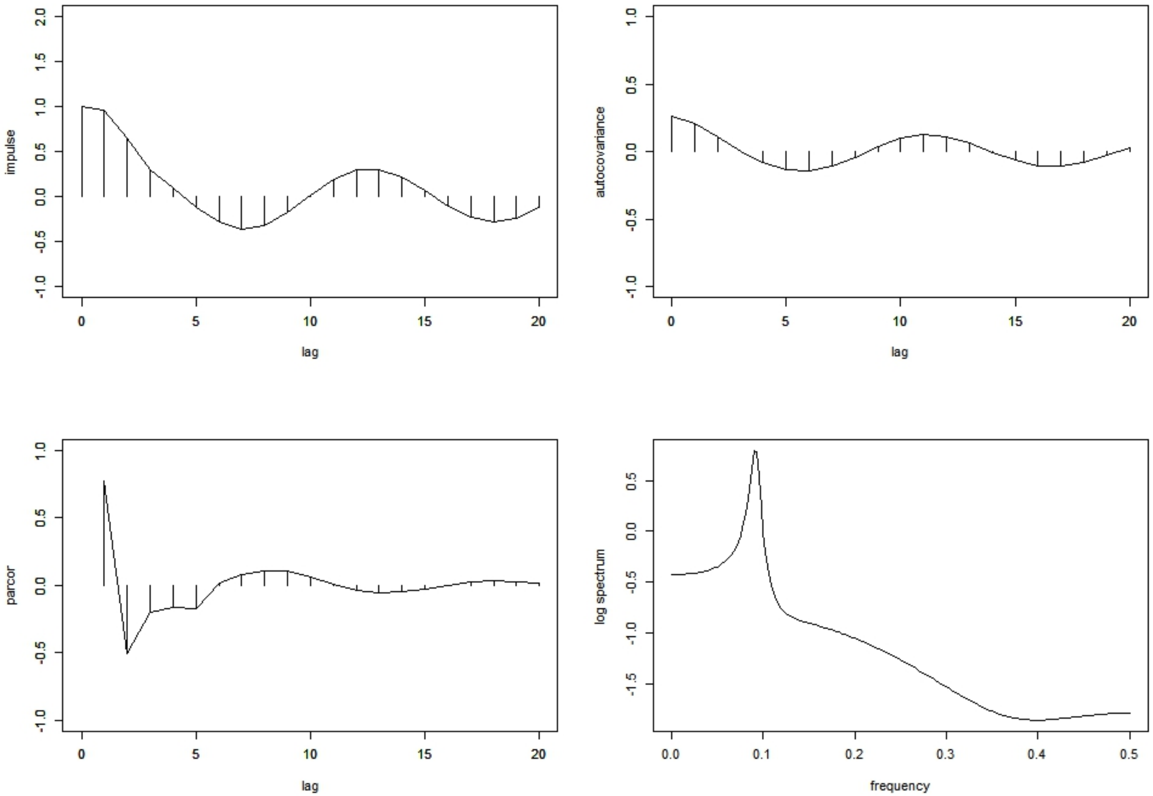


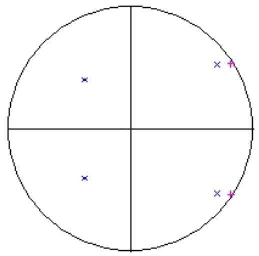


ARMA(2,4)

```
z1 <- armafit(y, ar.order=2, ma.order=4)
```

```
$sigma2    [1] 0.06810441
$lkfood    [1] -17.90792
$aic       [1] 49.81583
$arcoef    [1]  1.635564 -0.950000
$macoef    [1]  0.66611049 -0.01741201
              -0.15433929 -0.23316617
```





ARMA characteristic roots
(AR:pink, MA:blue)

Rによる自動探索

```
#  Automatic fitting of ARMA models
#
mmax <- 10                                          # maximum AR order
lmax <- 5                                           # maximum MA order
m2 <- (mmax+1)*(lmax+1)
aic <- matrix(1:m2,nrow=mmax+1,ncol=lmax+1)
aic[1,1] <- 0
ii <- seq(1,mmax,length=mmax)
jj <- seq(1,lmax,length=lmax)
#
for (i in ii){
 z1 <- armafit(y$z,ar.order=i,ma.order=0)           # AR(i) model
 aic[i+1,1] <- z1$aic
 for (j in jj){
  ar <- z1$arcoef
  ma <- z1$macoef
  ma[j] <- 0.001
  z1 <- armafit(y$z,ar.order=i,ma.order=j,ar,ma)    # ARMA(i,j)
  aic[i+1,j+1] <- z1$aic
 }
}
for (j in jj){                                      # MA(j) models (j=1,…,lmax)
  ma <- z1$macoef
  z1 <- armafit(y$z,ar.order=0,ma.order=j,ma)
  aic[1,j+1] <- z1$aic
}
```

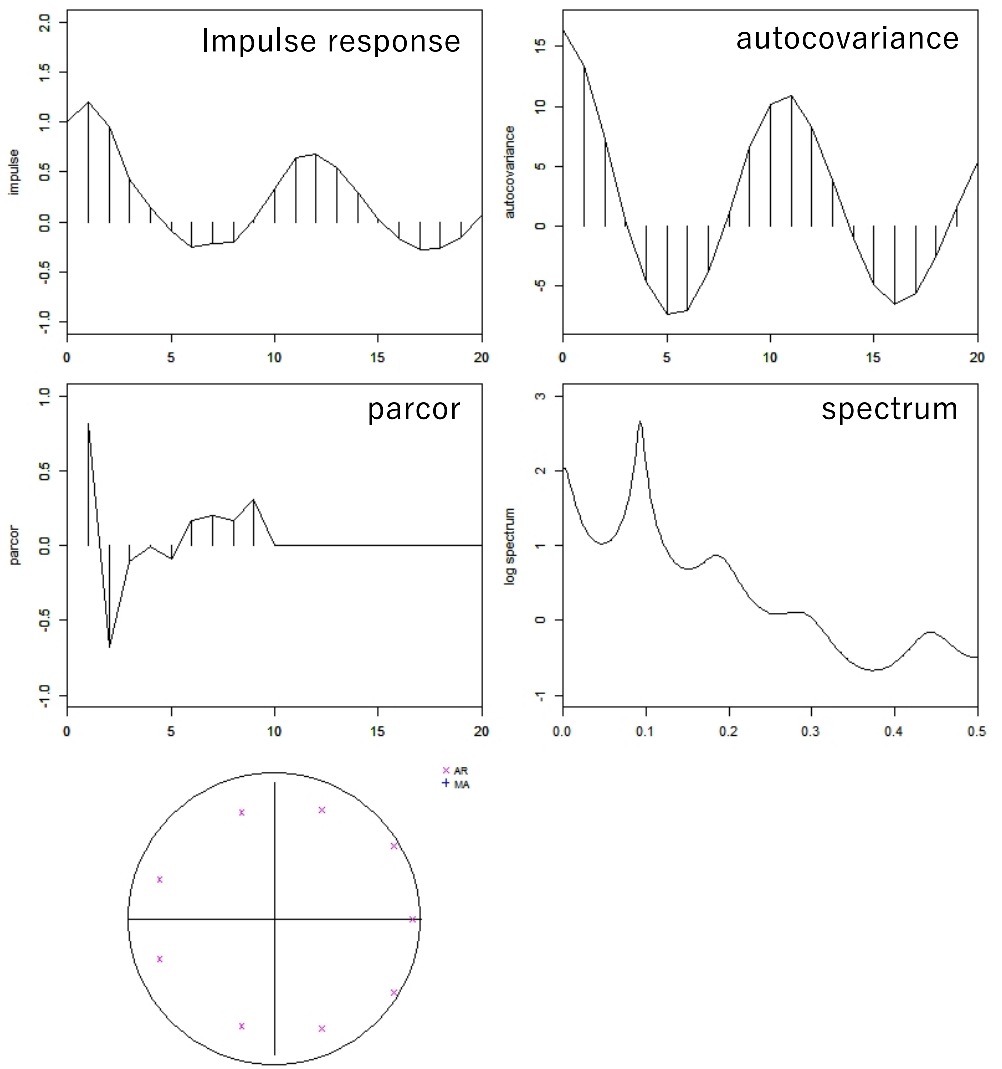
ARMAモデルのAICと対数尤度

AIC

		MA order					
		0	1	2	3	4	5
AR order	0	1301.6	1092.8	965.7	943.7	931.3	924.8
	1	1051.0	972.7	933.7	934.3	929.8	925.4
	2	910.9	910.5	912.5	914.5	892.6	894.0
	3	910.5	912.5	913.8	914.0	893.7	889.0
	4	912.5	914.5	904.6	897.3	896.0	898.1
	5	912.9	913.4	906.3	908.1	909.3	883.5
	6	907.8	887.6	884.2	878.2	880.9	882.6
	7	901.0	886.2	885.6	880.1	882.4	884.2
	8	896.5	885.8	883.0	880.0	882.8	884.9
	9	876.0	877.9	879.9	881.8	879.5	877.2
	10	877.9	879.9	876.5	878.3	880.7	882.5

data：boxcox(Sunspot)

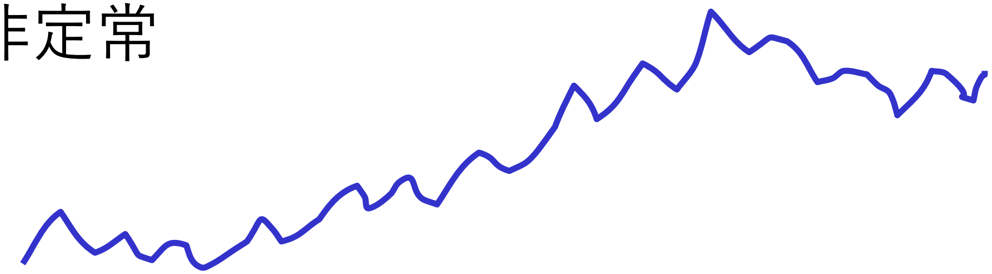
ARMA(9,0)



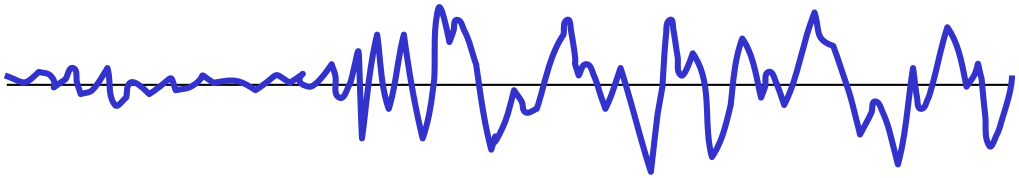
非定常時系列のモデリング

非定常：　平均，分散，共分散の何れかが時間変化

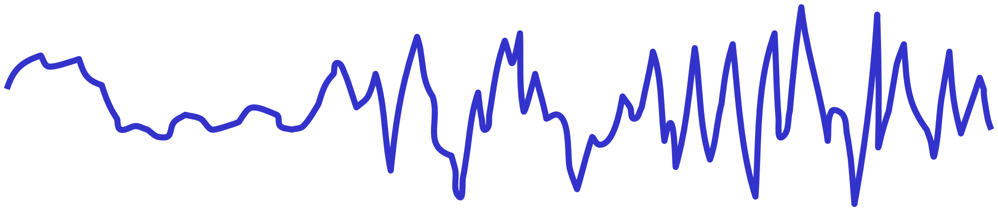
平均非定常



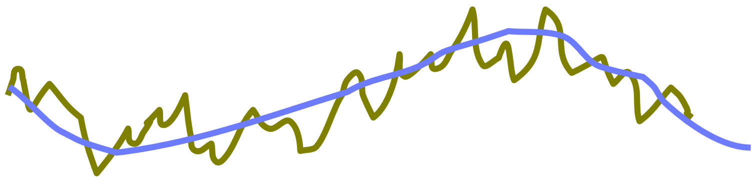
分散非定常



共分散非定常



トレンドの推定

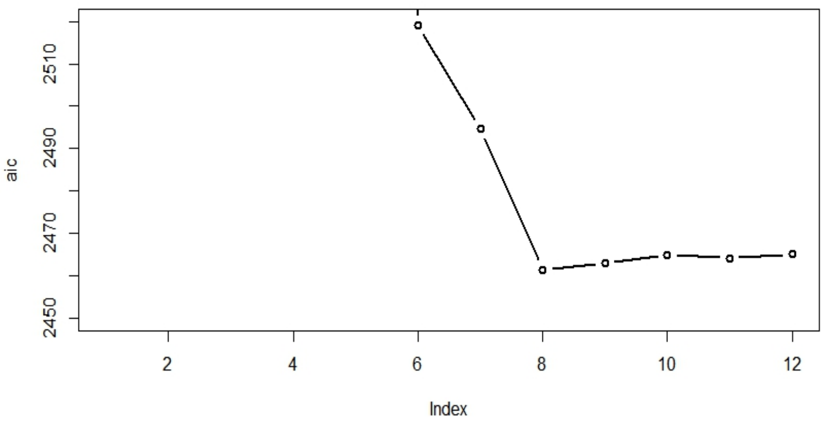


- 移動平均，移動メディアン　　　ノンパラメトリックモデル
- 回帰モデル　$y_n = f(n) + \varepsilon_n$　　　パラメトリックモデル
  - 多項式回帰　$f(n) = a_0 + a_1 n + \cdots + a_k n^k$
  - 三角関数回帰　$f(n) = a_0 + \sum_{j=1}^{m} b_j \sin(j\omega n) + \sum_{j=1}^{\ell} c_j \cos(j\omega n)$
- トレンドモデル

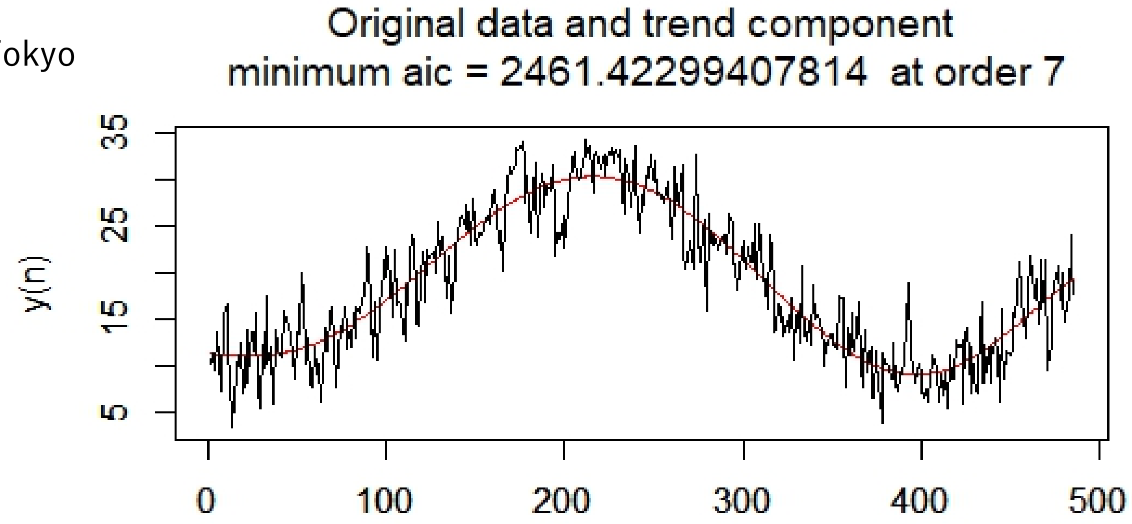
多項式回帰モデル

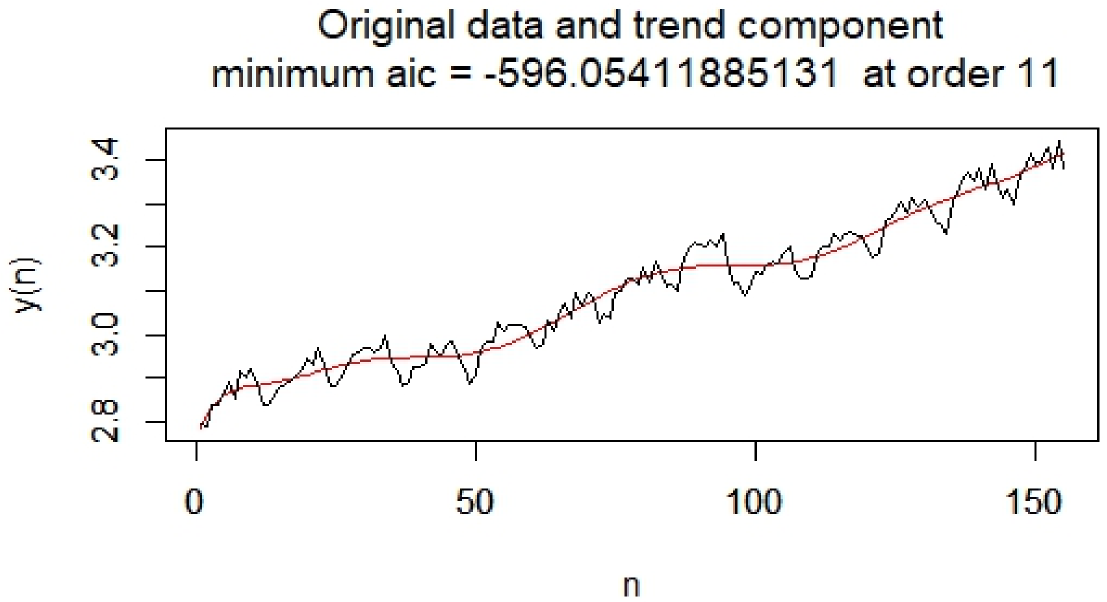
data(Temperature)　　# Highest Temperature Data of Tokyo
polreg(Temperature, 7)

$$f(n) = a_0 + a_1 n + \cdots + a_k n^k$$



data(WHARD)　# Wholesale hardware data
y <- log10(WHARD)
polreg(y, 14)





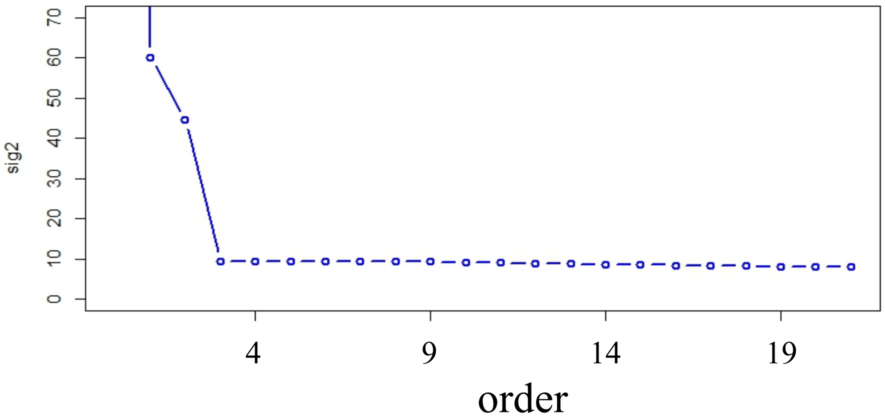
 CC BY-NC-ND

三角関数回帰モデル

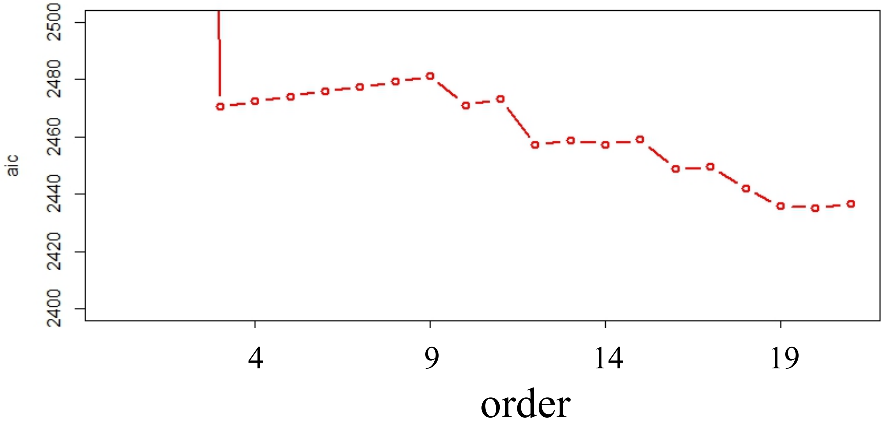
$$y_n = a_0 + \sum_{j=1}^{m} b_j \sin(j\omega n) + \sum_{j=1}^{\ell} c_j \cos(j\omega n) + \varepsilon_n$$

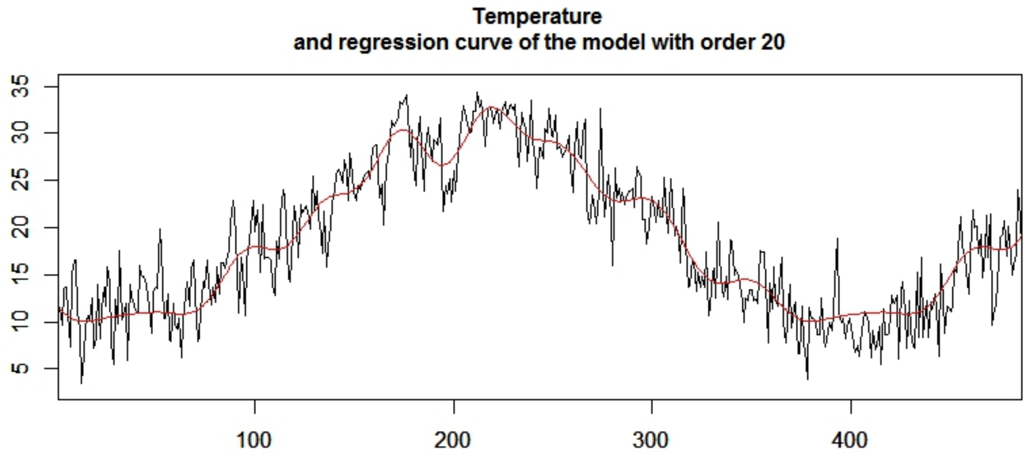
lsqr(Temperature)

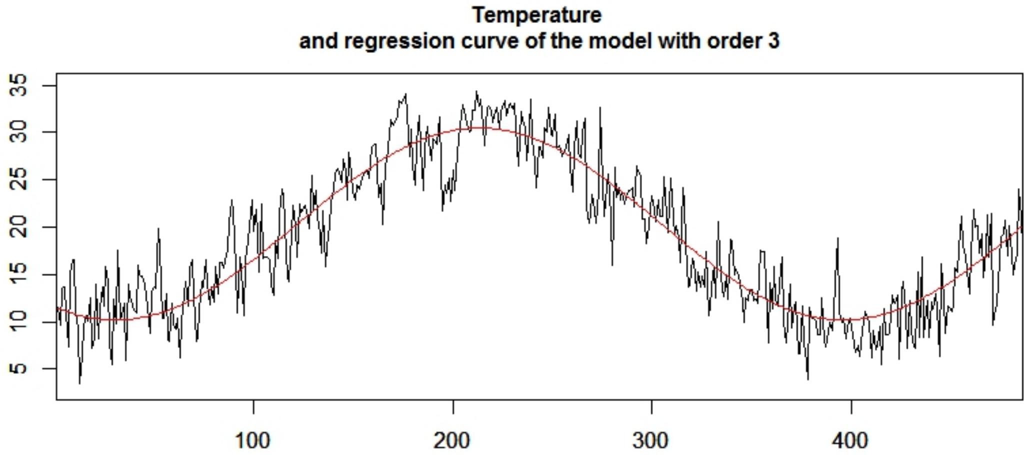
Sigma2



AIC







UTokyo Online Education 数理手法VII 2019 北川源四郎 CC BY-NC-ND

平均非定常のモデル

- 回帰モデル

$$y_n = f(n) + \varepsilon_n \qquad f(n)：パラメトリック関数$$

  - 結果は関数形に依存する

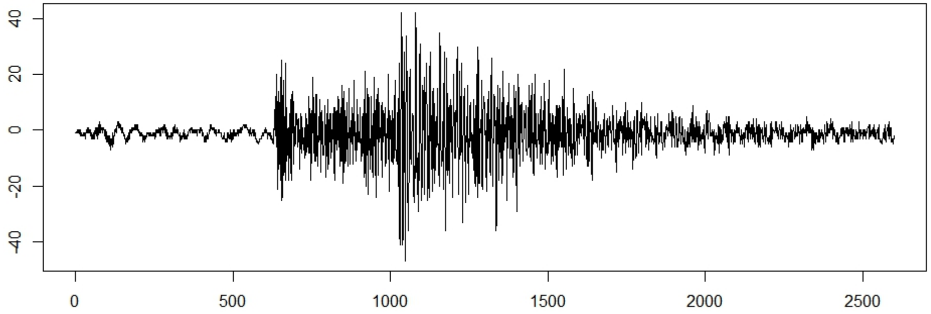
- トレンドモデル

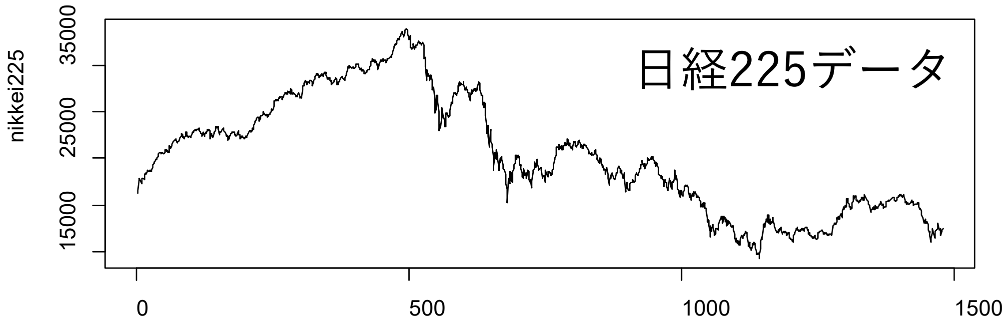
$$y_n = t_n + \varepsilon_n \qquad t_n：パラメータ$$

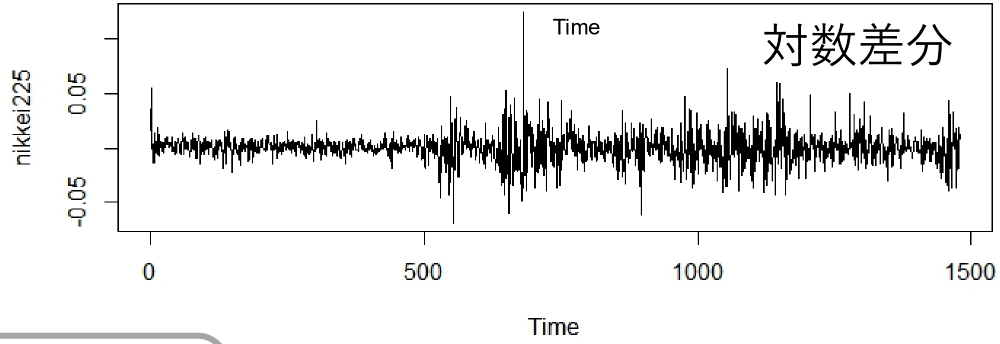
  - 柔軟にトレンドの変化を表現できる
  - 問題：　パラメータ数 > データ数

分散非定常のモデル

地震データ







$$y_n = \sigma_n w_n \qquad \sigma_n^2: \text{ 時変分散}$$

$$y_n = \Delta \log r_n$$

ボラティリティ

σ_n : パラメータ

- 柔軟に分散の変化を表現できる
- 問題：　パラメータ数 > データ数

共分散非定常のモデル

時変係数ARモデル

$$y_n = \sum_{j=1}^{m} a_{jn} y_{n-j} + w_n, \qquad w_n \sim N(0, \sigma_n^2)$$

$$(a_1, \ldots, a_m) \;\Rightarrow\; (a_{1n}, \ldots, a_{mn})$$

n に依存して係数は変化

未知パラメータ数： $n(m+1) \quad > \quad$ データ数 n

CC BY-NC-ND

共通の課題

- 多数の未知数（パラメータ数 > データ数）
- 通常の最小二乗法、最尤法は使えない

ペナルティ付き最小二乗法（正則化法）

平滑化の問題

$$y_n = f_n + \varepsilon_n, \quad n = 1, \ldots, N$$

y_n　観測値
f_n　未知パラメータ
ε_n　ノイズ（残差）

ペナルティ付き最小二乗法（L_2正則化法）

$$\min_f \left[\underbrace{\sum_{n=1}^{N} (y_n - f_n)^2}_{\text{データからの乖離}} + \lambda^2 \underbrace{\sum_{n=1}^{N} (\nabla^k f_n)^2}_{\text{滑らかさからの乖離}} \right]$$

平滑化パラメータのベイズモデルの観点からの選択

最も重要な問題は平滑化パラメータの決定

$$\sum_{n=1}^{N}(y_n - f_n)^2 + \lambda^2 \sum_{n=1}^{N}(\Delta^k f_n)^2$$

$-1/(2\sigma^2)$ をかけて指数をとる

$$\exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - f_n)^2\right\}\exp\left\{-\frac{\lambda^2}{2\sigma^2}\sum_{n=1}^{N}(\Delta^k f_n)^2\right\}$$

平滑化事前分布

ベイズモデルによる解釈 $\theta = (\lambda^2, \sigma^2)$

$$\pi(f \mid y, \theta) \propto p(y \mid f, \theta)\boxed{\pi(f \mid \theta)}$$

ABICによる θ の決定

Akaike(1980)

時系列的解釈と状態空間モデル

$$\sum_{n=1}^{N}(y_n - t_n)^2 + \lambda^2 \sum_{n=2}^{N}(t_n - t_{n-1})^2$$

等価なモデル

$$t_n = t_{n-1} + v_n \qquad v_n \sim N(0, \tau^2)$$

$$y_n = t_n + w_n \qquad w_n \sim N(0, \sigma^2)$$

$$\lambda^2 = \frac{\sigma^2}{\tau^2}$$

一般化（状態空間モデル）

$$x_n = Fx_{n-1} + Gv_n$$

$$y_n = Hx_n + w_n$$

CC BY-NC-ND

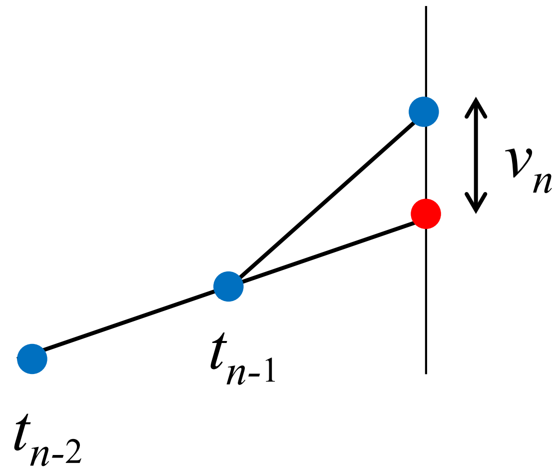
構造変化のモデル

- 多項式 $y_n = \underbrace{a_0 + a_1 n + \cdots + a_{k-1} n^{k-1}}_{t_n} + \varepsilon_n$

  $\Delta^k t_n = 0$ の解　　$\Delta t_n = t_n - t_{n-1}$

- 多項式の拡張

  $$\Delta^k t_n \cong 0 \;\Rightarrow\; \Delta^k t_n = v_n$$



トレンド成分モデル

$$\Delta^k t_n = v_n$$

- トレンド成分モデルはトレンドモデルだけでなく，成分分解，時変係数モデルなどで利用される

構造変化のモデル

多項式の拡張

$$\Delta^k t_n \cong 0 \quad \Rightarrow \quad \Delta^k t_n = v_n$$

$$\Delta t_n = t_n - t_{n-1}$$

$$k = 1: \qquad t_n - t_{n-1} = v_n \quad \Rightarrow \quad t_n = t_{n-1} + v_n$$

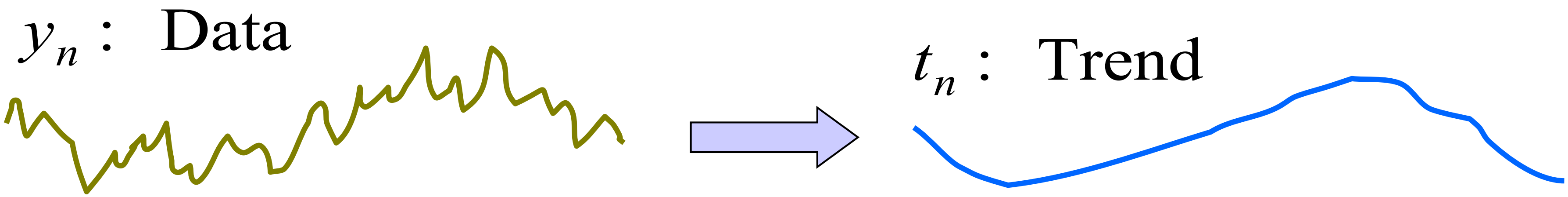
ランダムウォーク

$$k = 2: \qquad \Delta t_n - \Delta t_{n-1} = v_n$$

$$\Rightarrow t_n - 2t_{n-1} + t_{n-2} = v_n$$

$$\Rightarrow t_n = 2t_{n-1} - t_{n-2} + v_n$$

トレンドモデル



$$\Delta^k t_n = v_n$$

トレンド成分モデル

$$y_n = t_n + w_n$$

観測モデル

状態空間表現

$$x_n = F x_{n-1} + G v_n$$
$$y_n = H x_n + w_n$$

 CC BY-NC-ND

トレンドモデル（例：k=1）

k=1の場合

$$\Delta t_n = v_n$$

$$\Rightarrow \quad t_n - t_{n-1} = v_n$$

$$\Rightarrow \quad t_n = t_{n-1} + v_n$$

$$t_n = t_{n-1} + v_n$$
$$y_n = t_n + w_n$$

$$x_n = [t_n]$$
$$F = [1]$$
$$G = [1]$$
$$H = [1]$$

状態空間表現

$$x_n = Fx_{n-1} + Gv_n$$
$$y_n = Hx_n + w_n$$

トレンドモデル（例：k=2）

k =2の場合

$$\Delta^2 t_n = v_n$$

$$\Rightarrow \Delta t_n - \Delta t_{n-1} = v_n$$

$$\Rightarrow t_n - 2t_{n-1} + t_{n-2} = v_n$$

$$\Rightarrow t_n = 2t_{n-1} - t_{n-2} + v_n$$

$$x_n = \begin{bmatrix} t_n \\ t_{n-1} \end{bmatrix},$$

$$F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, G = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

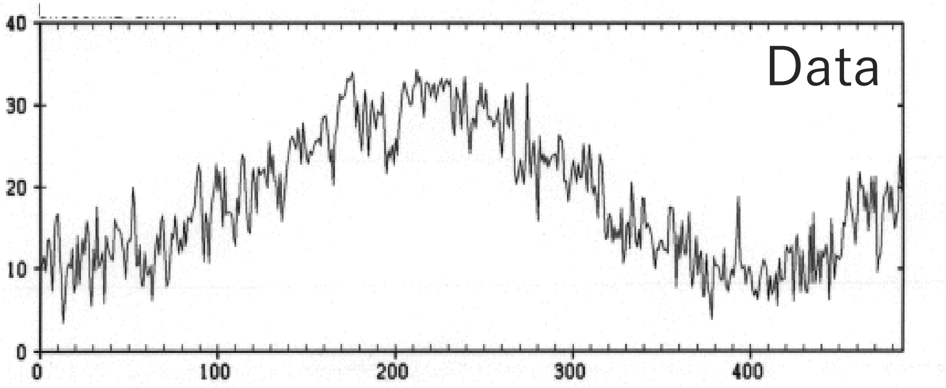
$$H = [1 \quad 0]$$

$$t_n = 2t_{n-1} - t_{n-2} + v_n$$
$$y_n = t_n + w_n$$

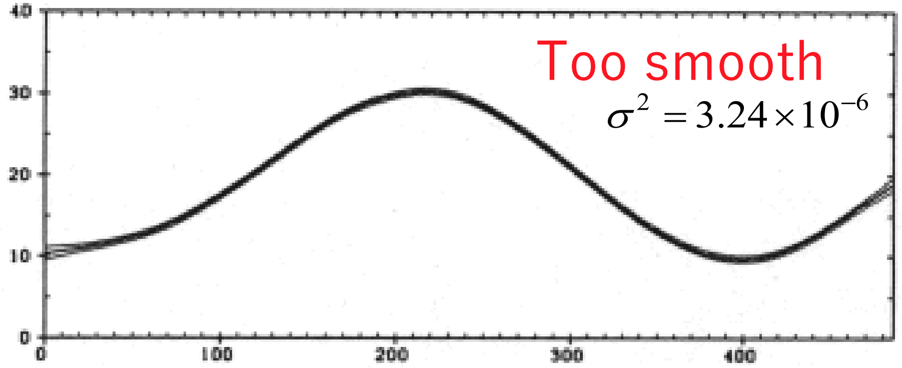
状態空間表現

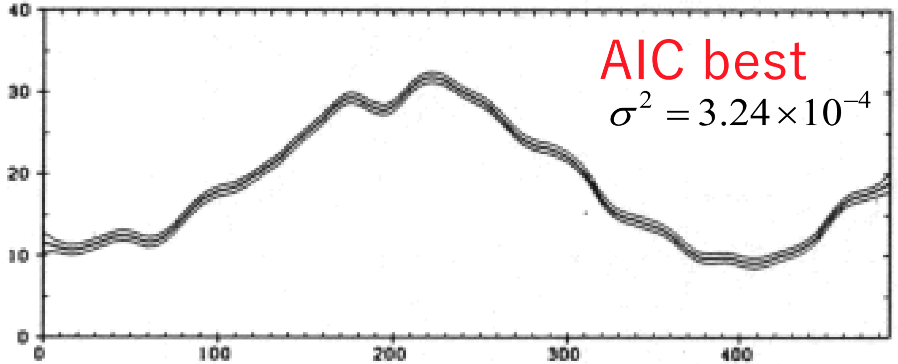
$$x_n = F x_{n-1} + G v_n$$
$$y_n = H x_n + w_n$$

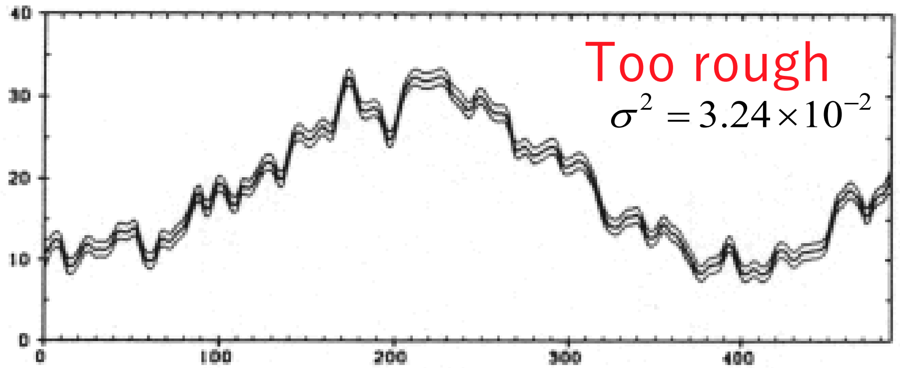
平滑化パラメータの選択



τ^2	σ^2	AIC
3.24×10^{-6}	10.100	2555.4
3.24×10^{-5}	8.827	2510.3
3.24×10^{-4}	8.077	2505.3
3.24×10^{-3}	7.179	2517.9
3.24×10^{-2}	6.047	2561.6





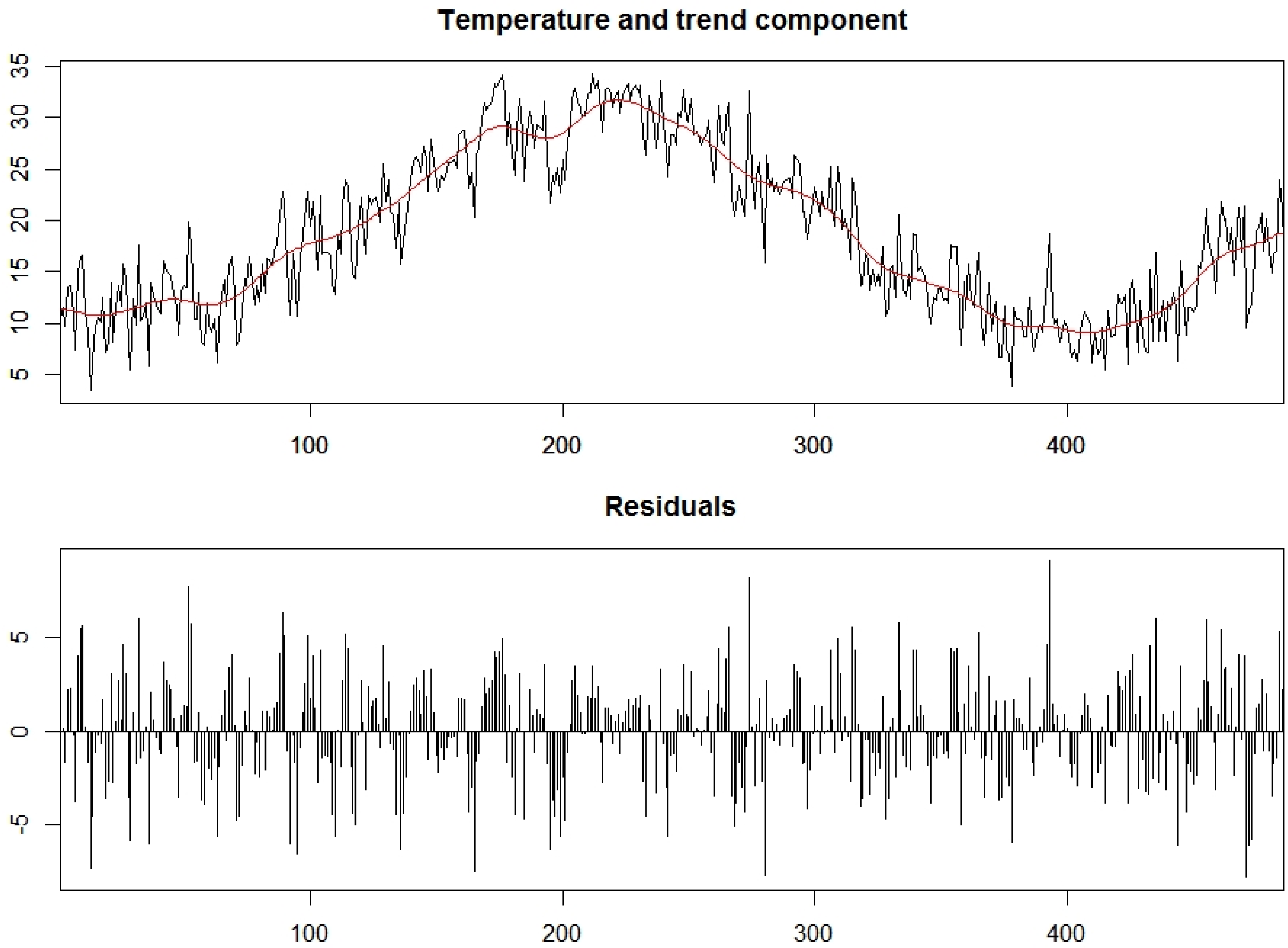


UTokyo Online Education 数理手法VII 2019 北川源四郎 CC BY-NC-ND

R による計算：トレンドモデル（k=2）

```
trend(Temperature, trend.order = 2)
```

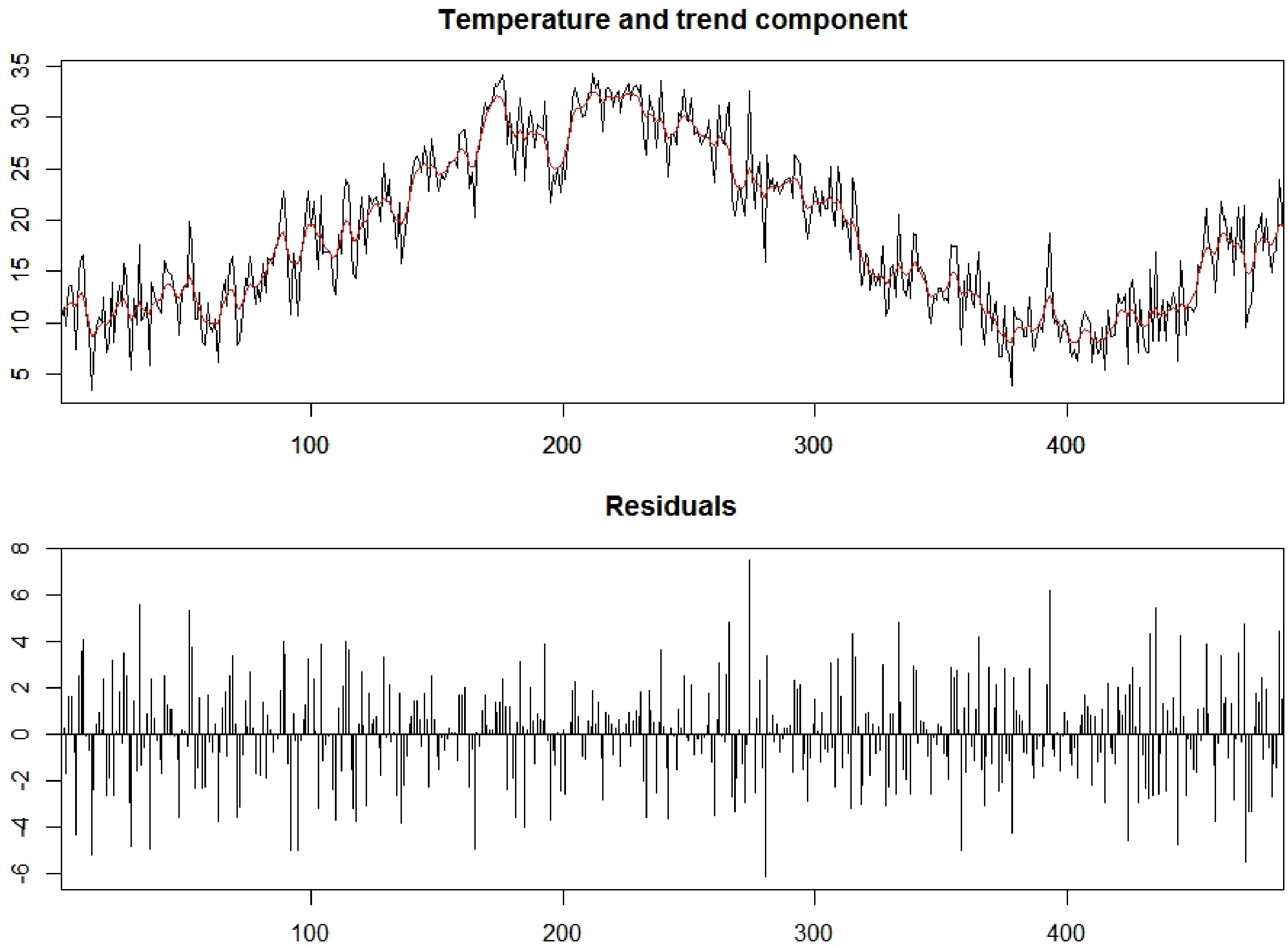
tau2	2.44141e-04
sigma2	8.17920e+00
log-likelihood	-1248.696
aic	2505.392



Rによる計算：トレンドモデル（k=1）

trend(Temperature, trend.order = 1, tau2.ini = 0.223, delta = 0.001)

```
tau2            2.23000e-01
sigma2          5.54743e+00
log-likelihood    -1220.841
aic                2447.682
```



R による計算

k=1 の場合（初期値指定）

```
data(Temperature)
trend(Temperature, trend.order = 1, tau2.ini = 0.223,
delta = 0.001)
```

tau2	2.23000e-01
sigma2	5.54743e+00
log-likelihood	-1220.841
aic	2447.682

```
trend(Temperature, trend.order = 1, tau2.ini = 2.23e-2,
delta = 0.00)
```

tau2	2.23000e-02
sigma2	8.26754e+00
log-likelihood	-1240.908
aic	2487.815

```
trend(Temperature, trend.order = 1, tau2.ini = 2.23e-3,
delta = 0.00)
```

tau2	2.23000e-03
sigma2	1.38749e+01
log-likelihood	-1342.511
aic	2691.022

k=2 の場合（tau2：自動）

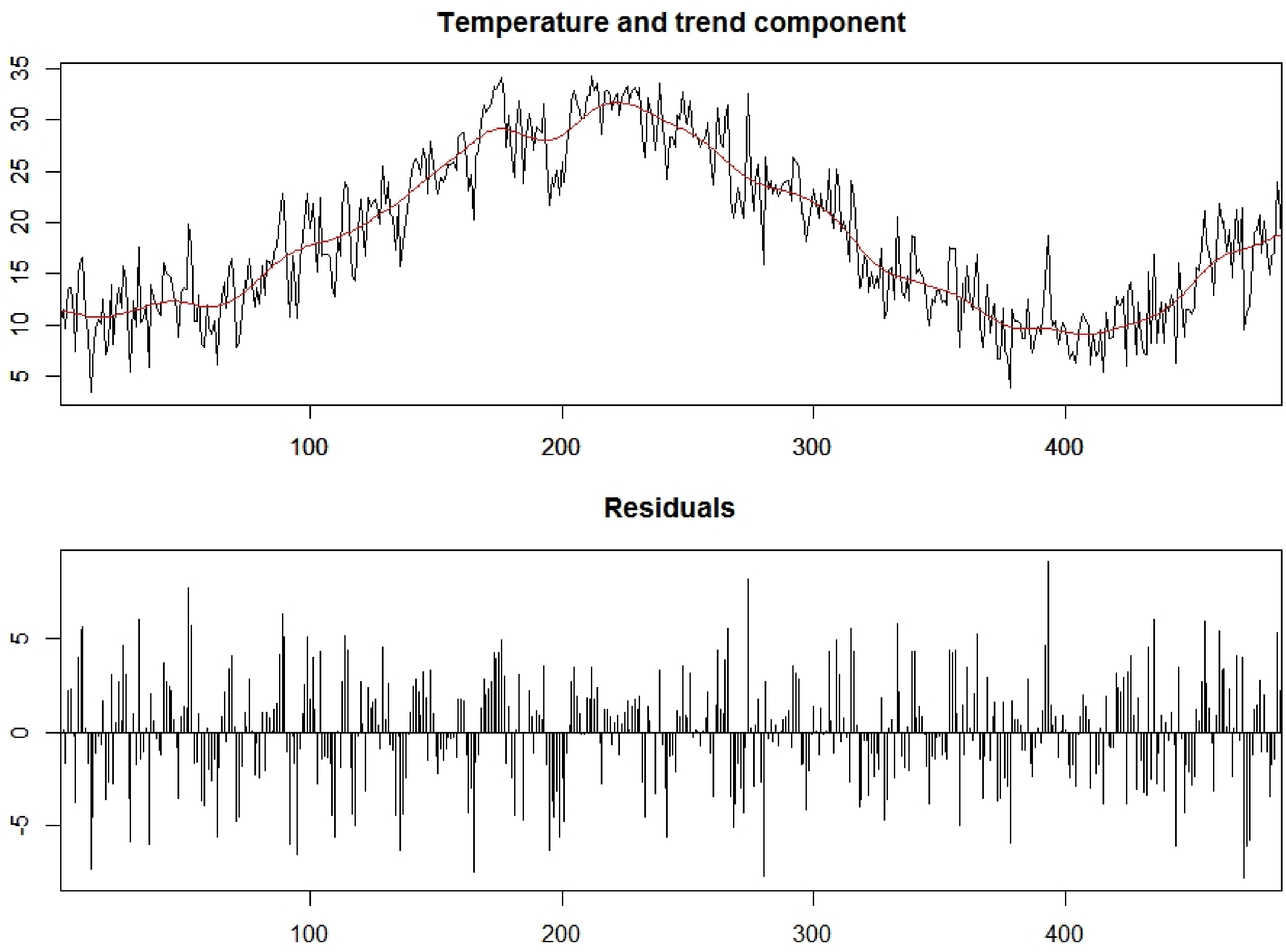
```
trend(Temperature, trend.order = 2)
```

tau2	2.44141e-04
sigma2	8.17920e+00
log-likelihood	-1248.696
aic	2505.392

CC BY-NC-ND

R による計算(k=2)

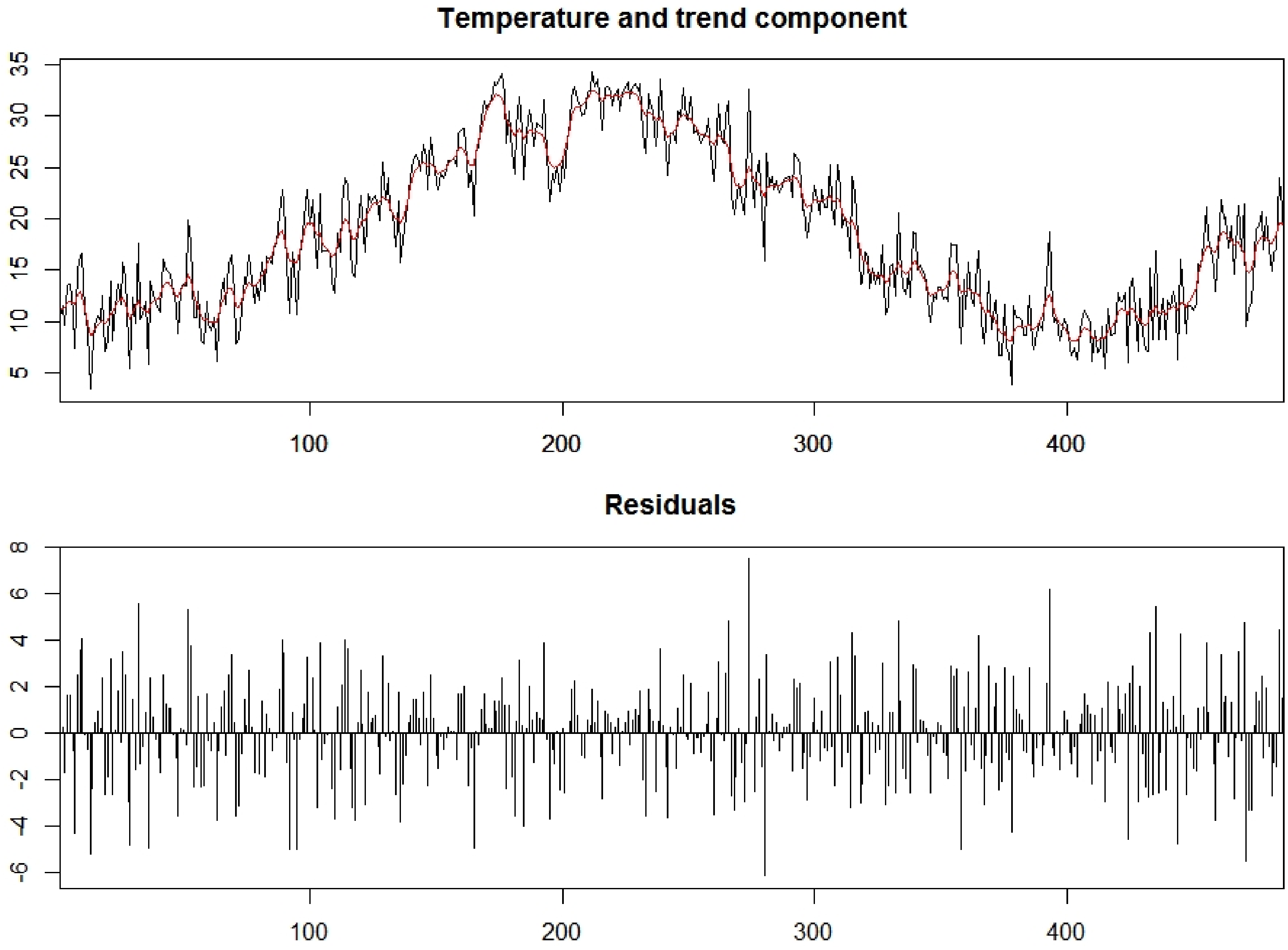
trend(Temperature, trend.order = 2)



R による計算(k=1)

```
data(Temperature)
trend(Temperature, trend.order = 1, tau2.ini =
0.223, delta = 0.001)
```

tau2	2.23000e-01
sigma2	5.54743e+00
log-likelihood	-1220.841
aic	2447.682



トレンドモデル（拡張例）

$$\begin{aligned} t_n &= 2t_{n-1} - t_{n-2} + v_n \\ &= t_{n-1} + (t_{n-1} - t_{n-2}) + v_n \\ &= t_{n-1} + \Delta t_{n-1} + v_n \\ \Delta t_n &= \Delta t_{n-1} + v_n \end{aligned}$$

$$T = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}$$

- トレンドモデルと等価な表現

$$x_n = \begin{bmatrix} t_n \\ \Delta t_n \end{bmatrix},\quad F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, G = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, H = [1 \quad 0]$$

レベルと傾きが連動して変化

- 拡張したモデル

$$\begin{aligned} t_n &= t_{n-1} + \Delta t_{n-1} + v_n + u_n \\ \Delta t_n &= \Delta t_{n-1} + v_n \end{aligned}$$

レベルと傾きが独立に変化する

$$G = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

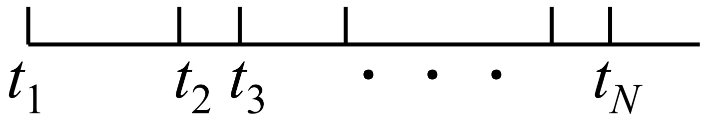
不等間隔時間モデル

不等間隔に観測された時系列のフィルタリング・平滑化

観測点：　$t_1, \ldots, t_N$

観測値：　$y(t_1), \ldots, y(t_N)$

$f(t)$のk階微分：　$f^{(k)}(t)$



$$y(t) = f(t) + \varepsilon(t)$$

$$f^{(k)}(t) = a(1) f^{(k-1)} + \cdots + a(k) f(t) + w(t)$$

状態空間モデル（連続型）

$$dx(t) = Ax(t)dt + BdW(t)$$

$$y(t) = Cx(t) + \varepsilon(t)$$

$$x(t) = (f(t), f^{(1)}(t), \ldots, f^{(k-1)}(t))^T$$

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a(k) & a(k-1) & \cdots & a(1) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

$$C = [1 \ \ 0 \ \ \cdots \ \ 0]$$

不等間隔時間モデル

積分型表現

$$x(t_n) = F(t_n - t_{n-1})x(t_{n-1}) + \int_{t_{n-1}}^{t_n} F(t_n - u)BdW(u)$$

ただし，$F(u)$は微分方程式 $\dot{x}(t) = Ax(t)$ の遷移行列

状態空間モデル（離散型，不等間隔時間）

$$x(t_n) = F(t_n - t_{n-1})x(t_{n-1}) + Gv(t_n, t_{n-1})$$
$$y(t_n) = Hx(t_n) + \varepsilon(t_n)$$

$$G = I_k, \quad H = [1 \quad 0 \quad \cdots \quad 0]$$

$$v(t,s) = \int_s^t F(t-u)BdW(u)$$

$$\mathrm{cov}(v(t,s)) = \tau^2 \int_s^t F(t-u)BB^T F(t-u)^T du$$

不等間隔時間モデル(例)

$$f^{(k)}(t) = dW(t)$$ 連続時間モデル

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C^T = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & t & \frac{t^2}{2} & \cdots & \frac{t^{k-1}}{(k-1)} \\ & 1 & t & \ddots & \vdots \\ & & 1 & \ddots & \frac{t^2}{2} \\ & & & \ddots & t \\ & & & & 1 \end{bmatrix}$$

$$F(t-u)B = \begin{bmatrix} \frac{(t-u)^{k-1}}{(k-1)!} & \frac{(t-u)^{k-2}}{(k-2)!} & \cdots & t-u & 1 \end{bmatrix}$$

$$\mathrm{cov}(v_i(t,s), v_j(t,s)) = \tau^2 \int_s^t \frac{(t-u)^{k-i}(t-u)^{k-j}}{(k-i)!(k-j)!} du = \frac{\tau^2 (t-s)^{2k-i-j+1}}{(2k-i-j+1)!(k-i)!(k-j)!}$$

$$\ell(\theta) = -\frac{1}{2}\left\{ N \log 2\pi + \sum_{i=1}^{N} \log v(t_i) + \sum_{i=1}^{n} \frac{(y(t_i) - Hx(t_i \mid t_{i-1}))^2}{2v(t_i)} \right\}$$

$$v(t_i) = HV(t_i \mid t_{i-1})H^T + R$$