時系列解析（6）

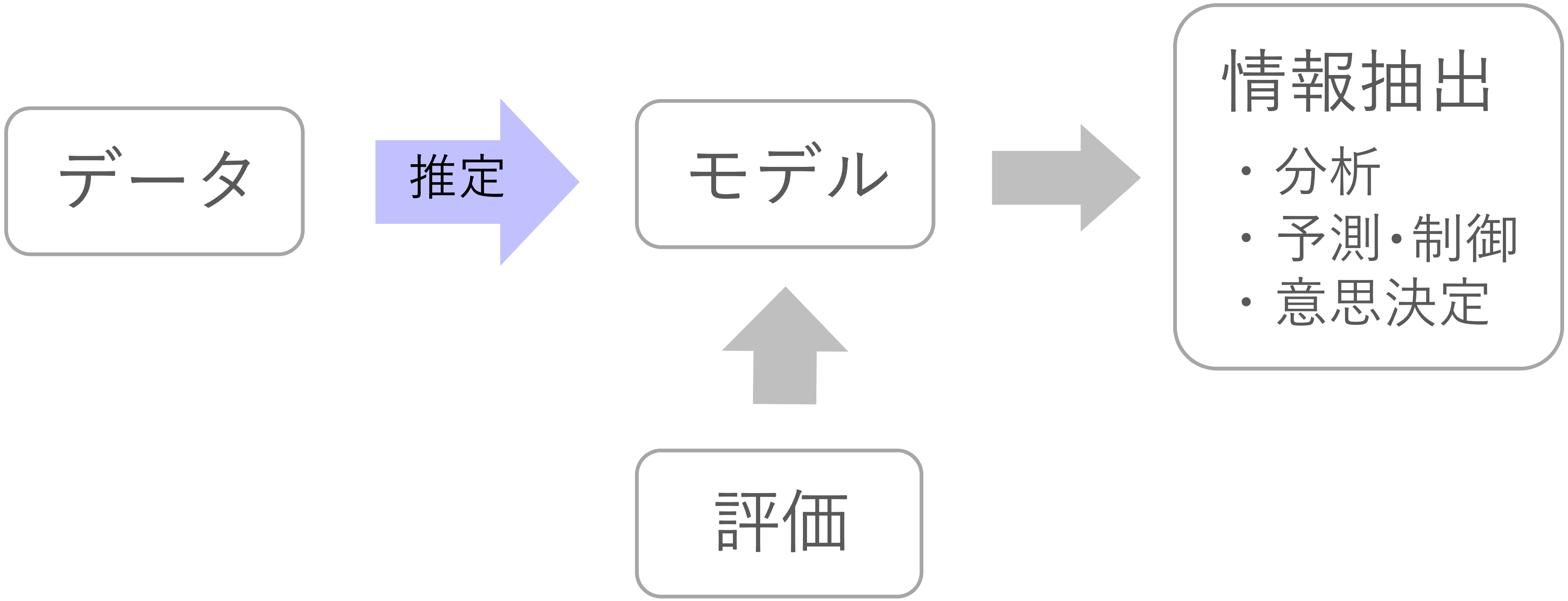
− ARモデルの推定 −

東京大学 数理・情報教育研究センター
北川 源四郎

概　要

１．ARモデルによる予測
２．1変量ARモデルの推定
　（１）Yule-Walker法
　（２）最尤法
　（３）最小二乗法（Householder法）
　（４）PARCOR法
　（５）数値例
２．多変量ARモデルの推定
　（１）Levinson-Durvin法
　（２）Householder法
　（３）変数選択例
３．関連する話題
　（１）最終予測誤差FPE
　（２）統計的最適制御

時系列のモデリング



自己回帰モデル（AR Model）

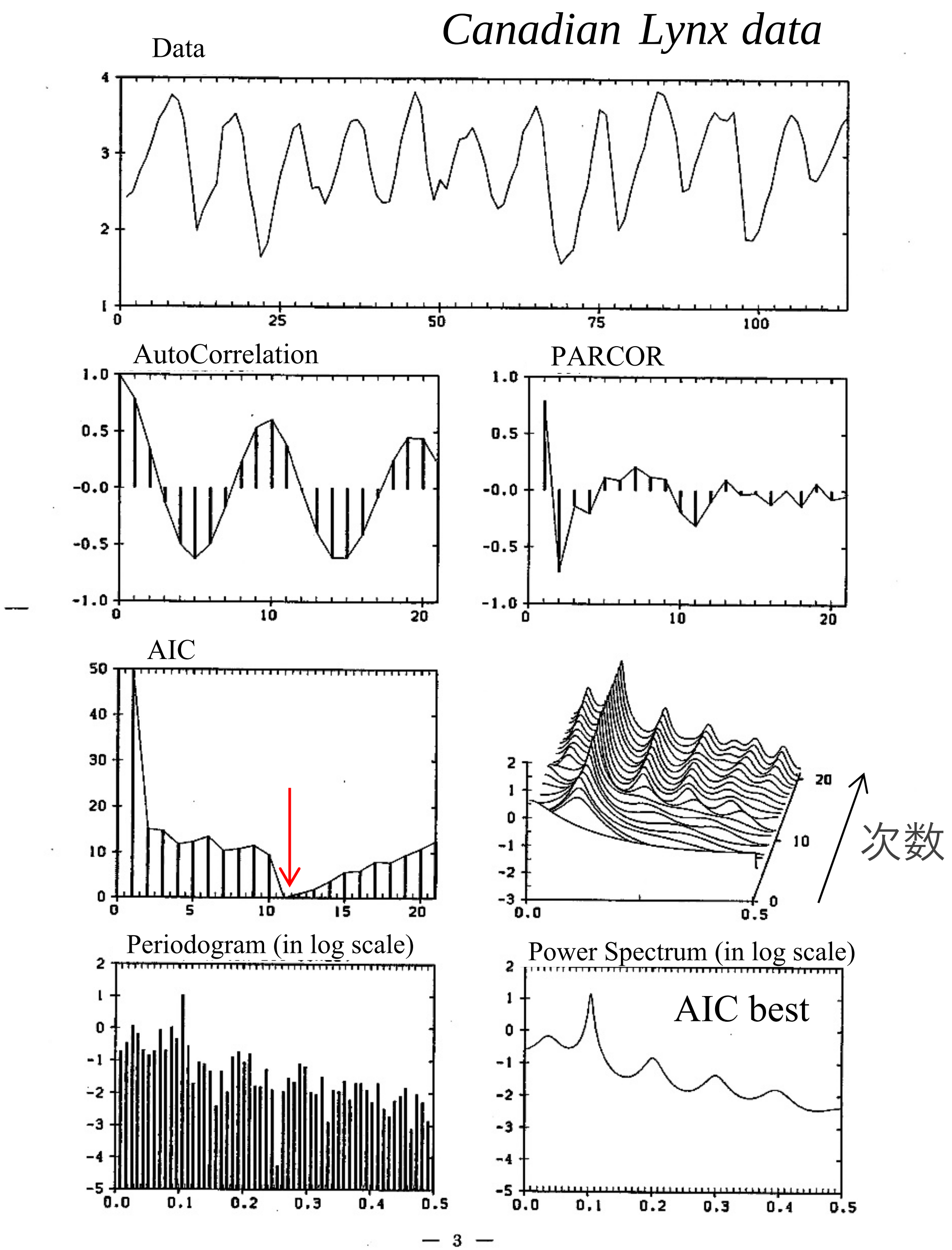
$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n$$

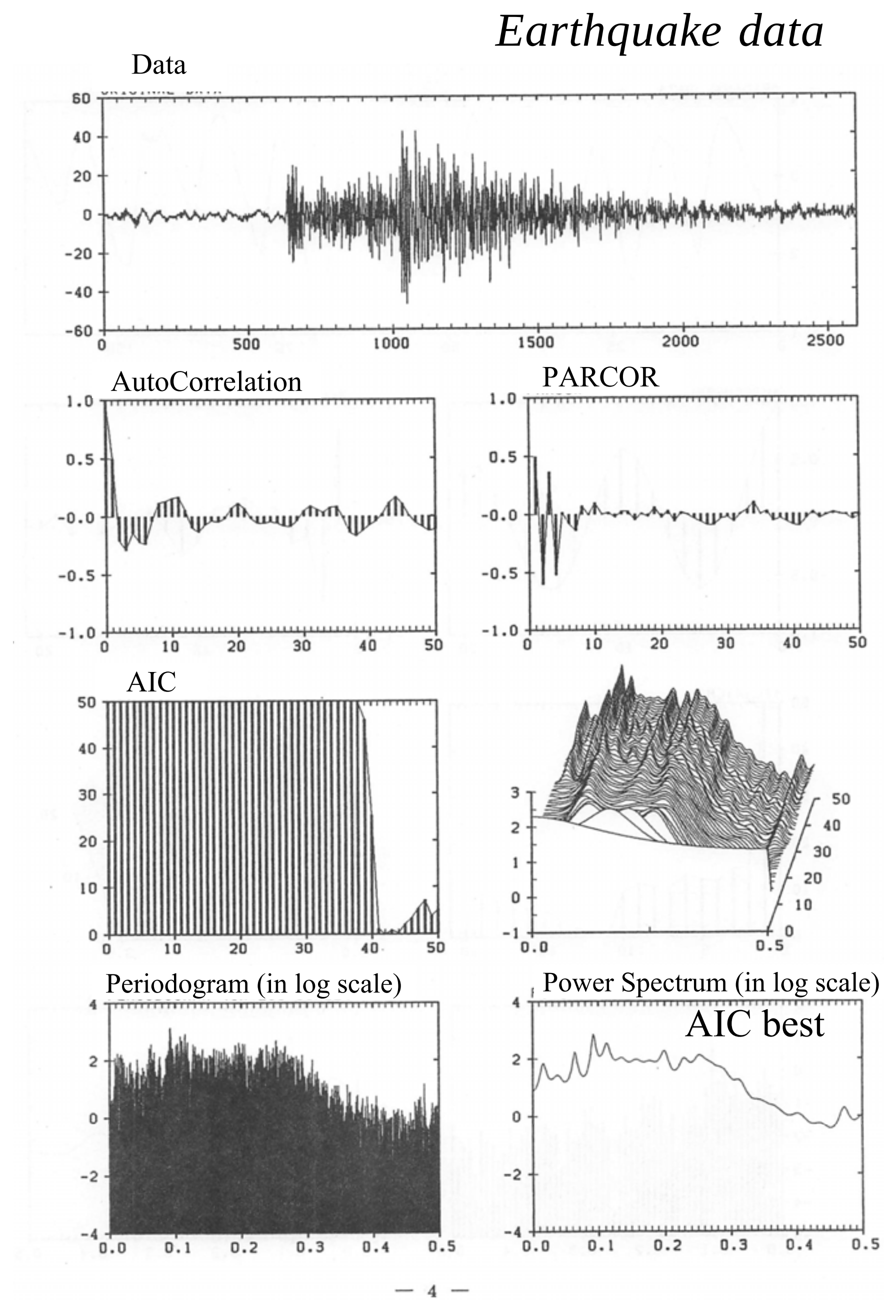
y_n	定常時系列	
v_n	正規白色雑音	$v_n \sim N(0, \sigma^2)$
m	自己回帰の次数	$E[v_n v_k] = 0 \quad n \neq k$
a_j	自己回帰係数	$E[v_n y_{n-j}] = 0 \quad j > 0$

ARスペクトル

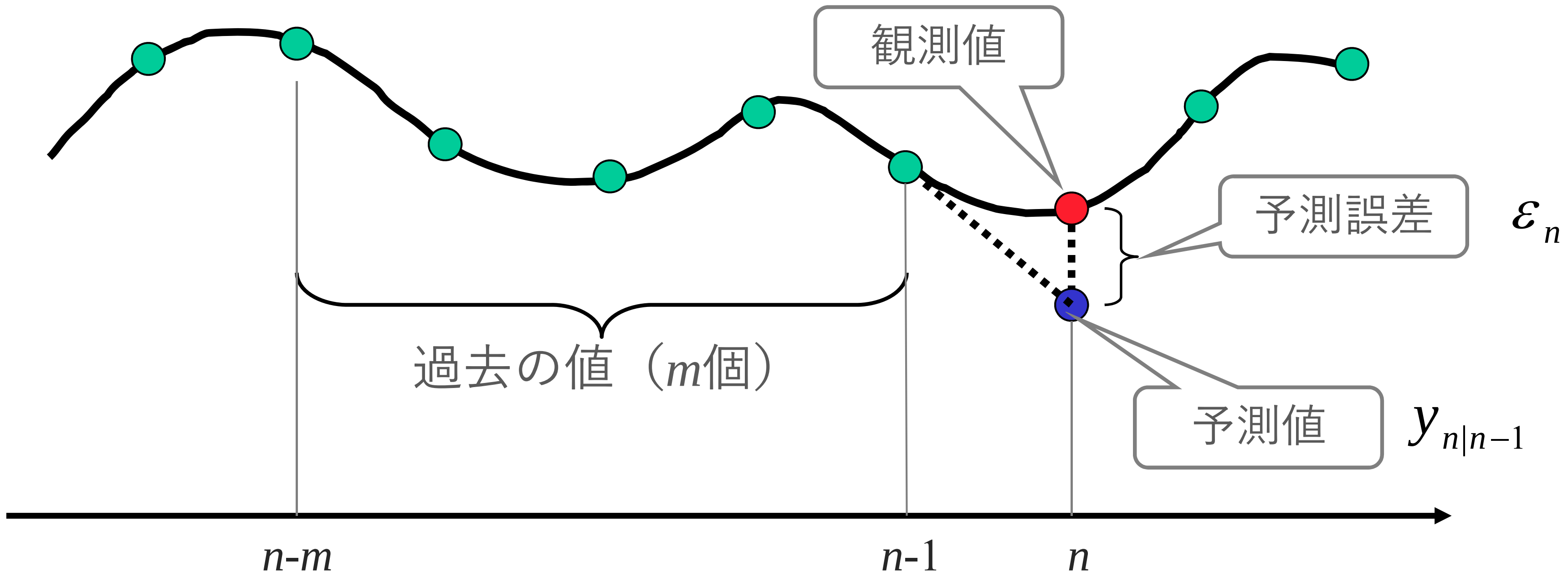
$$p(f) = \frac{\sigma^2}{\left|1 - \sum_{j=1}^{m} a_j e^{-2\pi ijf}\right|^2}, \qquad -\frac{1}{2} \le f \le \frac{1}{2}$$

次数によって結果は変わる（スペクトル）





ARモデルと予測



$$y_n = a_1 y_{n-1} + \cdots + a_m y_{n-m} + v_n$$

ARモデルによる予測

ARモデル

$$y_{n+1} = \underbrace{a_1 y_n + \cdots + a_m y_{n-m+1}}_{\text{現在までのデータで決まる部分}} + \underbrace{v_{n+1}}_{\text{ノイズ}}$$

1期先予測

$$y_{n+1|n} \equiv a_1 y_n + \cdots + a_m y_{n-m+1}$$

$y_{n+1|n}$: 時刻 n までの情報に基づくy_{n+1}の予測値

一期先予測誤差

$$\varepsilon_{n+1} = y_{n+1} - y_{n+1|n} = v_{n+1}$$

一期先予測誤差の性質

平均 $\quad E\varepsilon_{n+1} = E v_{n+1} = 0$

分散 $\quad E\varepsilon_{n+1}^2 = E v_{n+1}^2 = \sigma^2$

ARモデルによる長期予測

$$y_{n+2} = a_1 y_{n+1} + a_2 y_n + \cdots + a_m y_{n-m+2} + v_{n+2}$$

$$y_{n+2|n} = a_1 y_{n+1|n} + a_2 \underbrace{y_{n|n}}_{y_n} + \cdots + a_m \underbrace{y_{n-m+2|n}}_{y_{n+2-m}} + \underbrace{v_{n+2|n}}_{0}$$

$$y_{n+1|n} = a_1 y_n + a_2 y_{n-1} + \cdots + a_m y_{n-m+1}$$

$$y_{n+2|n} = a_1 y_{n+1|n} + a_2 y_n + \cdots + a_m y_{n-m+2}$$

$$y_{n+3|n} = a_1 y_{n+2|n} + a_2 y_{n+1|n} + \cdots + a_m y_{n-m+3}$$

$$\vdots$$

長期予測の誤差

$$\varepsilon_{n+1} = y_{n+1|n} - y_{n+1} = v_{n+1}$$

$$y_{n+2} = \underline{a_1 y_{n+1}} + a_2 y_n + \cdots + a_m y_{n-m+2} + v_{n+2}$$

$$y_{n+2|n} = \underline{a_1 y_{n+1|n}} + a_2 y_n + \cdots + a_m y_{n-m+2}$$

$$\varepsilon_{n+2} = y_{n+2|n} - y_{n+2}$$

$$= a_1 (y_{n+1} - y_{n+1|n}) + v_{n+2} = a_1 \varepsilon_{n+1} + v_{n+2}$$

$$E\varepsilon_{n+2} = a_1 E\varepsilon_{n+1} + Ev_{n+2} = 0$$

$$E\varepsilon_{n+2}^2 = a_1^2 E\varepsilon_{n+1}^2 + 2a_1 E\varepsilon_{n+1} v_{n+2} + Ev_{n+2}^2 = (1 + a_1^2)\sigma^2$$

$$E\varepsilon_{n+3}^2 = \left\{ 1 + a_1^2 + (a_1^2 + a_2)^2 \right\} \sigma^2$$

長期予測の誤差の性質

$$y_n = \sum_{j=0}^{\infty} g_j v_{n-j}, \qquad y_{n+k} = \sum_{j=0}^{\infty} g_j v_{n+k-j}$$

一般論は状態空間モデルで

$$v_{n+k|n} = \begin{cases} v_{n+k} & k \le 0 \\ 0 & k > 0 \end{cases}$$

$$y_{n+k|n} = \sum_{j=0}^{\infty} g_j v_{n+k-j|n} = \sum_{j=k}^{\infty} g_j v_{n+k-j}$$

$$\varepsilon_{n+k} = y_{n+k} - y_{n+k|n} = \sum_{j=0}^{\infty} g_j v_{n+k-j} - \sum_{j=k}^{\infty} g_j v_{n+k-j} = \sum_{j=0}^{k-1} g_j v_{n+k-j}$$

$$E[\varepsilon_{n+k}] = \sum_{j=0}^{k-1} g_j E[v_{n+k-j}] = 0$$

$$E[\varepsilon_{n+k}^2] = \sum_{j=0}^{k-1} g_j^2 E[v_{n+k-j}^2] = \sigma^2 \sum_{j=0}^{k-1} g_j^2$$

（例）

$$g_0 = 1$$
$$g_1 = a_1$$
$$g_2 = a_1^2 + a_2$$

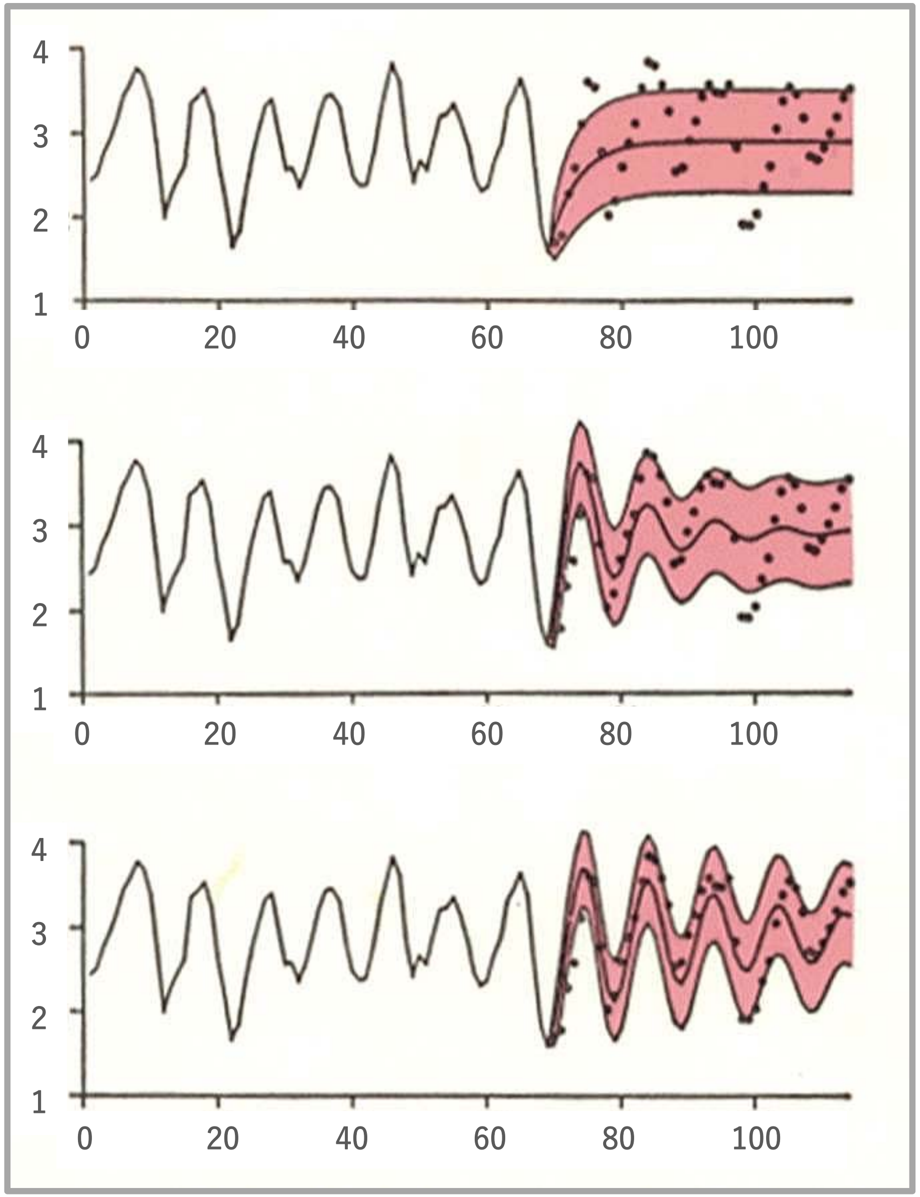
次数によって結果は変わる（予測）

$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n$$

いずれも最適予測分布
（平均、分散）

長期予測

Canadian Lynx data

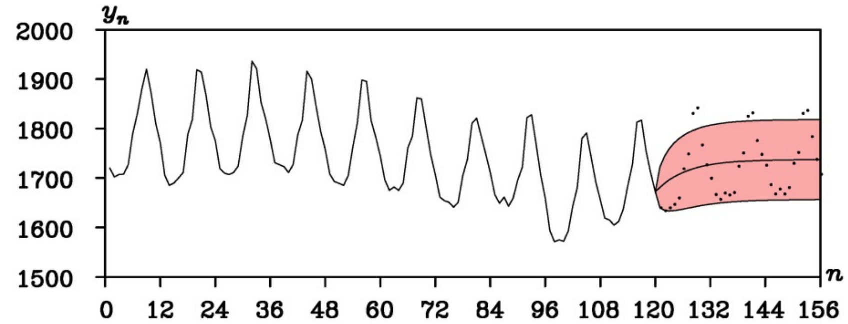


$m = 1$

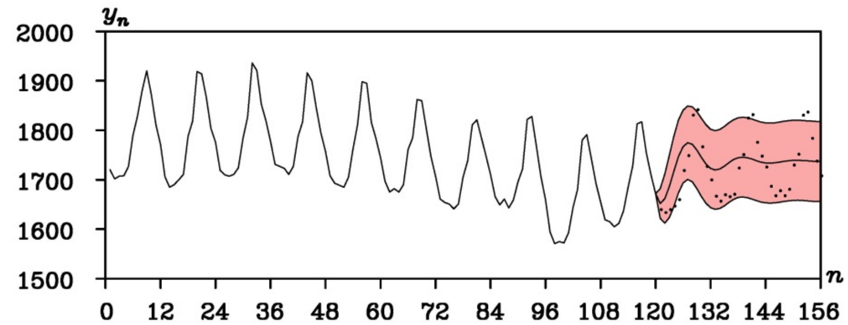
$m = 5$

$m = 10$

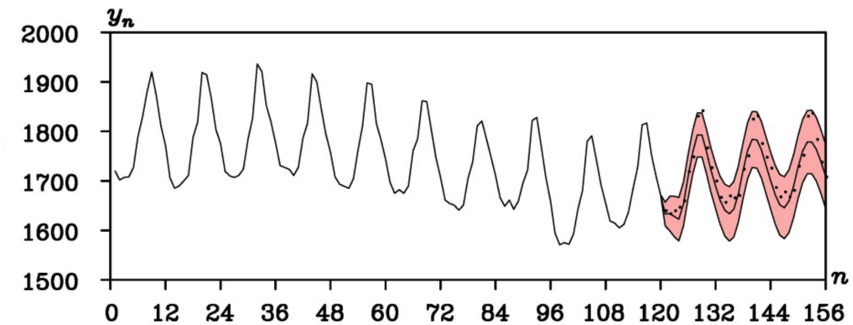
例：　長期予測　BIsfoodデータ



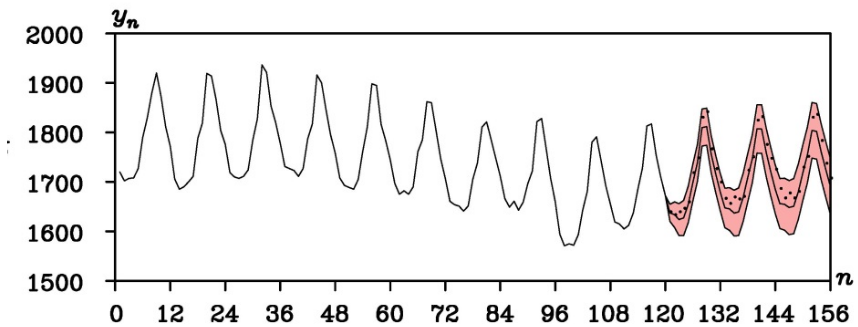
$m = 1$



$m = 5$



$m = 10$



$m = 15$

自己回帰モデル（AR Model)の推定

なぜ一般性のあるARMAモデルでなくARモデルの推定を考えるのか？

1．この時点でARMAモデルの推定（最尤法など）は困難（状態空間モデルが必要）
2．ARモデルの推定は簡単（最小二乗法など）
3．ARモデルの実用性が高い
4．ARMAモデルは自由度が高すぎ，やや不安定

ARモデルの同定問題

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

- パラメータ推定

  AR係数 a_j と分散 σ^2 の推定

- 次数 m の選択

  AR次数 m の決定

✓ 多数の次数とパラメータを推定する必要がある

ARモデルのパラメータ推定の方法

1．Yule-Walker法
2．最尤法
3．最小二乗法
4．PARCOR法（3種類）

（1）Yule-Walker法

$$y_n = \sum_{i=1}^{m} a_j y_{n-j} + v_n$$

Yule-Walker方程式

$$C_0 = \sum_{j=1}^{m} a_j C_j + \sigma^2$$

$$C_k = \sum_{j=1}^{m} a_j C_{j-k} \qquad (k = 1,2,\cdots)$$

$$a_1,\ldots,a_m,\ \sigma^2$$

$$\Downarrow \Uparrow$$

$$C_0,\ C_1,\ldots,\ C_m$$

係数に関する1次方程式

$$\begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{m-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{m-1} & \hat{C}_{m-2} & \cdots & \hat{C}_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \\ \vdots \\ \hat{C}_m \end{bmatrix}$$

（Toeplitz matrix）

$$\hat{\sigma}^2 = \hat{C}_0 - \sum_{j=1}^{m} \hat{a}_j \hat{C}_j$$

Yule-Walker方程式の導出（再掲）

$$y_n = \sum_{i=1}^{m} a_j y_{n-j} + v_n$$

$$E(\underline{v_{n+k}} y_n) = \sum_{i=1}^{m} a_j E(\underline{v_{n+k}} y_{n-j}) + E(\underline{v_{n+k}} v_n)$$

$$E(v_{n+k} y_n) = \begin{cases} 0 & k > 0 \\ \sigma^2 & k = 0 \end{cases}$$

$$E(y_n \underline{y_{n-k}}) = \sum_{i=1}^{m} a_j E(y_{n-j} \underline{y_{n-k}}) + E(v_n \underline{y_{n-k}})$$

$$C_0 = \sum_{j=1}^{m} a_j C_{-k} + \sigma^2$$

$$C_k = \sum_{j=1}^{m} a_j C_{j-k}, \quad (k = 1, 2, \ldots)$$

Yule-Walker法は何をやっているか

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

$$\sigma^2 = E[v_n^2] = E\left[\left(y_n - \sum_{j=1}^{m} a_j y_{n-j}\right)^2 \right]$$

$$= C_0 - 2\sum_{j=1}^{m} a_j C_j + \sum_{j=1}^{m} \sum_{i=1}^{m} a_j a_i C_{i-j}$$

$$\frac{\partial \sigma^2}{\partial a_j} = -2C_j + 2\sum_{i=1}^{m} a_i C_{i-j} = 0, \qquad (j = 1, \ldots, m)$$

Yule-Walker法では予測誤差分散の期待値を
最小にするように係数 a_j を決めている.

Levinson’s Algorithm (Levinson-Durbin)

- 一連のYule-Walker方程式を単純に計算すると
  - m 元一次方程式の計算量 （Gauss消去法： $m^3/3 + O(m^2)$）
  - 次数は未知：　１次～M 次

$$(1+2^3+\cdots+M^3)/3 = M^2(M+1)^2/12 \approx M^4/12$$

- Levinsonのアルゴリズムは一連のYule-Walker方程式を効率よく求める方法
  - 計算量は $2M^2$ 程度になる

$M=100$ のとき

$$\frac{2M^2}{M^4/12} = \frac{24}{M^2} = \frac{24}{10,000} = 0.0024$$

Levinson's Algorithm

1. $\sigma_0^2 = C_0$

$$\mathrm{AIC}_0 = N(\log 2\pi\hat{\sigma}_0^2 + 1) + 2$$

2. $m = 1,\ldots,M$ について

(a) $a_m^m = (C_m - \sum_{j=1}^{m-1} a_j^{m-1} C_{m-j})(\sigma_{m-1}^2)^{-1}$

(b) $a_j^m = a_j^{m-1} - a_m^m a_{m-j}^{m-1}$

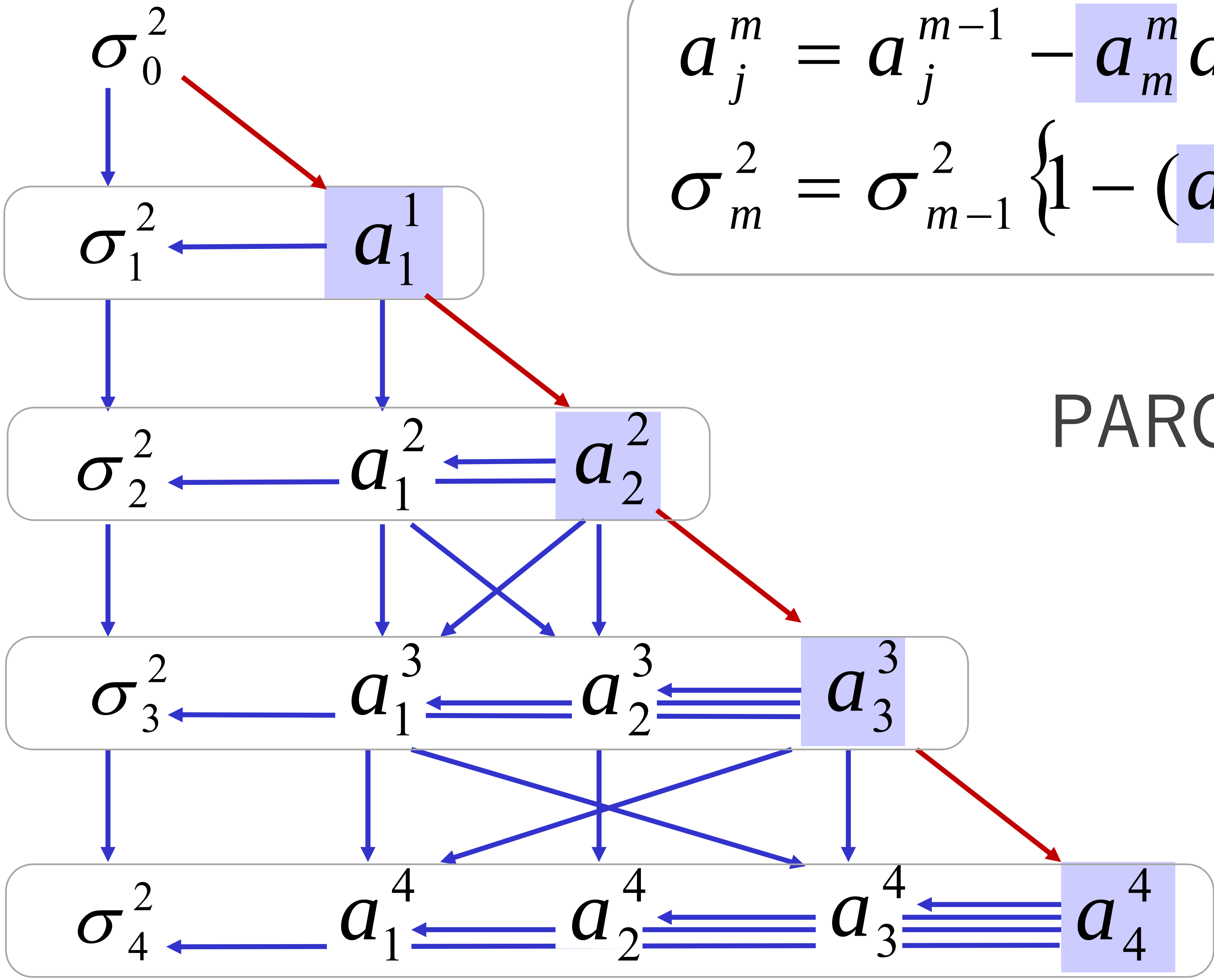
(c) $\sigma_m^2 = \sigma_{m-1}^2 \left\{1 - (a_m^m)^2\right\}$

(d) $\mathrm{AIC}_\mathrm{m} = N(\log 2\pi\hat{\sigma}_m^2 + 1) + 2(m+1)$

積除	和差
m	m-1
m-1	m-1
2	1
2m+1	2m-1

計算量： $\sum_{m=1}^{M} 4m = 2M(M+1) \approx 2M^2$

Levinson's Algorithm



$$a_j^m = a_j^{m-1} - a_m^m a_{m-j}^{m-1}$$

$$\sigma_m^2 = \sigma_{m-1}^2 \left\{1 - (a_m^m)^2\right\}$$

PARCOR

Levinson Algorithmの導出

C_k, (k=0,1,…)とm −1次のARモデルのYule-Wlaker推定値が与えられているとき，m次のARモデルのYule-Walker推定値を効率よく求める方法.

$\hat{a}_1^{m-1},\ldots,\hat{a}_{m-1}^{m-1}$はYule-Walker方程式をみたす.

$$C_k = \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_{j-k}, \qquad (k = 1,\ldots,m-1)$$

予測誤差 v_n^{m-1}について

$$E\left[v_n^{m-1} y_{n-k} \right] = E\left[(y_n - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-j}) y_{n-k} \right] = 0, \qquad (k = 1,\ldots,m-1)$$

Levinson Algorithmの導出（2）

$m-1$次の後向きARモデル

$$y_n = \sum_{j=1}^{m-1} a_j^{m-1} y_{n+j} + w_n^{m-1}$$

$C_{-k}=C_k$ なので，後向きARモデルも同じYule－Walker方程式をみたす．

$$E\left[w_{n-m}^{m-1} y_{n-k} \right] = E\left[(y_{n-m} - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-m+j}) y_{n-k} \right] = 0, \qquad (k = 1, \ldots, m-1)$$

$$\begin{aligned} z_n &\equiv v_n^{m-1} - \beta w_{n-k}^{m-1} \\ &= y_n - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-j} - \beta (y_{n-m} - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-m+j}) \\ &= y_n - \sum_{j=1}^{m-1} (\hat{a}_j^{m-1} - \beta \hat{a}_{m-j}^{m-1}) y_{n-j} - \beta y_{n-m} \end{aligned}$$

$E\left[z_n y_{n-k} \right] = 0, \qquad k = 1, \ldots, m-1$ については自動的になりたつので

$E\left[z_n y_{n-m} \right] = 0$ が成りたつように β を決めればよい．

Levinson Algorithmの導出（３）

$$E\left[z_n y_{n-m}\right] = E\left[\left\{y_n - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} y_{n-j} - \beta(y_{n-m} - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} y_{n-m+j})\right\} y_{n-m}\right]$$

$$= C_m - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} C_{m-j} - \beta(C_0 - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} C_j) = 0$$

$$\beta = (C_0 - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} C_j)^{-1}(C_m - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} C_{m-j})$$

$$= (\sigma_{m-1}^2)^{-1}(C_m - \sum_{j=1}^{m-1}\hat{a}_j^{m-1} C_{m-j})$$

$$a_m^m = \beta$$

$\hat{a}_j^m \equiv \hat{a}_j^{m-1} - \hat{a}_m^m \hat{a}_{m-j}^{m-1}$ とおくと

$$E\left[(y_n - \sum_{j=1}^{m-1}\hat{a}_j^m y_{n-j}) y_{n-k}\right] = C_k - \sum_{j=1}^{m-1}\hat{a}_j^m C_{k-j} = 0, \qquad (k = 1, \ldots, m)$$

$\hat{a}_1^m, \ldots, \hat{a}_m^m$は$m$次のYule-Walker方程式の解

Levinson Algorithmの導出（４）

$$\hat{\sigma}_m^2 = C_0 - \sum_{j=1}^{m-1} \hat{a}_j^m C_j$$

$$= C_0 - \sum_{j=1}^{m-1} (\hat{a}_j^{m-1} - \hat{a}_m^m \hat{a}_{m-j}^{m-1}) C_j - \hat{a}_m^m C_m$$

$$= C_0 - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_j - \hat{a}_m^m (C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j)$$

$$= \hat{\sigma}_{m-1}^2 - \hat{a}_m^m (C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j)$$

$$C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j = \hat{a}_m^m \hat{\sigma}_{m-1}^2$$

$$\hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2 (1 - (\hat{a}_m^m)^2)$$

（２）最尤法：ARモデルの尤度

$$y_n = \sum_{j=1}^{m} a_j y_{n-j} + v_n, \qquad v_n \sim N(0, \sigma^2) \qquad\qquad \theta = (a_1, \ldots, a_m, \sigma^2)^T$$

- 定義通りの方法

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \qquad \mu = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_1 & C_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_1 \\ C_{N-1} & \cdots & C_1 & C_0 \end{bmatrix}$$

$$y \sim N(\mu, C)$$

$$L(\theta) = p(y_1, \ldots, y_N \mid \theta) = (2\pi)^{-\frac{N}{2}} \mid C \mid^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(x-\mu)^T C^{-1} (x-\mu) \right\}$$

$\ell(\theta) = \log L(\theta)$ の数値的最適化により最尤推定値を求める

（2）最尤法：ARモデルの尤度

● 各時刻の条件付き分布に分解する方法

$$
\begin{aligned}
L(\theta) &= p(y_1,\ldots,y_N \mid \theta), \\
&= p(y_1 \mid \theta)\, p(y_2,\ldots,y_N \mid y_1,\theta) \\
&= p(y_1 \mid \theta)\, p(y_2 \mid y_1,\theta)\, p(y_3,\ldots,y_N \mid y_1,y_2,\theta) \\
&= \cdots \\
&= \prod_{n=1}^{N} p(y_n \mid y_1,\ldots,y_{n-1},\theta)
\end{aligned}
$$

$n \geq m$ のとき

$$
p(y_n \mid y_1,\ldots,y_{n-1},\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(y_n - \sum_{j=1}^{m} a_j y_{n-j} \right)^2 \right\}
$$

$n < m$ のとき ？

- 低次のAR係数を計算して，同様に条件付き分布を計算
- $(y_1,\ldots,y_m)$の部分だけ定義通りの方法で計算
- 厳密な最尤法には状態空間モデルが便利

（3）最小二乗法：尤度の近似

$n \geq m$ のとき

$$p(y_n \mid y_1, \ldots, y_{n-1}, \theta)$$

$$= p(y_n \mid y_{n-m}, \ldots, y_{n-1}, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} \left(y_n - \sum_{j=1}^{m} a_j y_{n-j} \right)^2 \right\}$$

$$\ell(\theta) = \sum_{n=1}^{N} \log p(y_n \mid y_1, \ldots, y_{n-1})$$

$$\cong \sum_{n=m+1}^{N} \log p(y_n \mid y_1, \ldots, y_{n-1})$$

$$= -\frac{N-m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=m+1}^{N} (y_n - \sum_{j=1}^{m} a_j y_{n-j})^2$$

- 最初のm個 $y_1,\ldots,y_m$の分布を無視(条件付けだけに使う)。
- モデル比較のためには無視するデータ数を同じにする。

（３）最小二乗法によるARモデルの推定

$$\ell'(\sigma^2, a_j) = -\frac{N-m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=m+1}^{N}(y_n - \sum_{j=1}^{m} a_j y_{n-j})^2$$

$$\frac{\partial \ell'(\sigma^2, a_j)}{\partial \sigma^2} = -\frac{N-m}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{n=m+1}^{N}(y_n - \sum_{j=1}^{m} a_j y_{n-j})^2 = 0$$

$$\hat{\sigma}^2 = \frac{1}{N-m}\sum_{n=m+1}^{N}(y_n - \sum_{j=1}^{m} a_j y_{n-j})^2$$

$$\ell'(\hat{\sigma}^2, a_j) = -\frac{N-m}{2}\log(2\pi\hat{\sigma}^2) - \frac{N-m}{2}$$

$$\max_{a_j} \ell'(\hat{\sigma}^2, a_j) \Leftrightarrow \min_{a_j} \hat{\sigma}^2 \Leftrightarrow \min_{a_j} \sum_{n=m+1}^{N}(y_n - \sum_{j=1}^{m} a_j y_{n-j})^2$$

- 近似最尤法 〜 最小二乗法

最小二乗法

$$y = \begin{bmatrix} y_{m+1} \\ \vdots \\ y_N \end{bmatrix},\quad Z = \begin{bmatrix} y_m & \cdots & y_1 \\ \vdots & & \vdots \\ y_{N-1} & \cdots & y_{N-m} \end{bmatrix},\quad a = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix},\quad v = \begin{bmatrix} v_{m+1} \\ \vdots \\ v_N \end{bmatrix}$$

$$y = Za + v$$

$$\|v\|^2 = \|y - Za\|^2$$

$$\hat{a} = (Z^T Z)^{-1} Z^T y$$

Householder 法

Householder法

- 平方根アルゴリズム（フィルタリングでも使える）
- 共分散行列を使わずデータ行列から直接推定

Householder法を使うメリット

1．計算精度（2倍精度）
2．トータルの計算量が少ない
2．モデルの自由度（変数選択、次数選択）
3．計算上の便利さ（モデル併合、データ逐次併合）

デメリット

1．Householder法特有のデメリットはほぼない
2．最小二乗法自体のデメリットは継承する

Householder 法

$$
Z = \begin{array}{c} \xleftrightarrow{\qquad\qquad m \qquad\qquad} \\ \begin{bmatrix} y_m & y_{m-1} & \cdots & y_1 \\ y_{m+1} & y_m & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_{N-m} \end{bmatrix} \end{array}, \qquad y = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_N \end{bmatrix} \Big\updownarrow N\text{-}m
$$

$$
X = [Z \mid y] = \left[\begin{array}{cccc|c} y_m & y_{m-1} & \cdots & y_1 & y_{m+1} \\ y_{m+1} & y_m & \cdots & y_2 & y_{m+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_{N-m} & y_N \end{array}\right]
$$

$$
HX = \begin{bmatrix} S \\ O \end{bmatrix} = \left[\begin{array}{ccc|c} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ & & & s_{m+1,m+1} \\ \hline & & O & \end{array}\right]
$$

Householder 法（予測誤差分散とAIC）

$$\hat{\sigma}_k^2 = \frac{1}{N-m}\sum_{i=k+1}^{m+1} s_{i,m+1}^2$$

$$\mathrm{AIC}_k = (N-m)(\log 2\pi\hat{\sigma}_k^2 + 1) + 2(k+1)$$

$$\begin{bmatrix} s_{11} & \cdots & s_{1k} \\ & \ddots & \vdots \\ & & s_{kk} \end{bmatrix}\begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{k,m+1} \end{bmatrix}$$

Householder法（再掲）

U :　任意の直交変換　（ベクトルの長さを変えない）

$$\|\varepsilon\|_N^2 = \|y - Za\|_N^2 = \|U(y - Za)\|_N^2 = \|Uy - UZa\|_N^2$$

$$\min_a \|\varepsilon\|^2 \Leftrightarrow \min_a \|Uy - UZa\|^2$$

$$X = \overbrace{[\, Z \mid y \,]}^{m+1} \updownarrow N \quad \Rightarrow \quad UX = S = \begin{bmatrix} S_{11} & \cdots & S_{1m} & S_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & S_{mm} & S_{m,m+1} \\ & & & S_{m+1,m+1} \\ & 0 & & \end{bmatrix}$$

最小二乗法（Householder法）

$$
\left\| Uy - UZa \right\|_N^2 = \left\| \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{1,m+1} \\ s_{1,m+1} \\ 0 \end{bmatrix} - \begin{bmatrix} s_{11} & \cdots & s_{11} \\ & \ddots & \vdots \\ & & s_{11} \\ & & \\ & 0 & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \right\|_N^2
$$

$$
= \left\| \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{m,m+1} \end{bmatrix} - \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ & \ddots & \vdots \\ & & s_{mm} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \right\|_m^2 + s_{m+1,m+1}^2
$$

最小二乗解

$$\begin{bmatrix} s_{11} & \cdots & s_{1m} \\ & \ddots & \vdots \\ & & s_{mm} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{m,m+1} \end{bmatrix}$$

$$\hat{a}_m = \frac{s_{m,m+1}}{s_{mm}}$$

$$\hat{a}_i = \frac{s_{i,m+1} - s_{i,i+1}\hat{a}_{i+1} - \cdots s_{i,m}\hat{a}_m}{s_{ii}} \qquad i = m-1,\ldots,1$$

$$\hat{\sigma}_m^2 = \frac{s_{m+1,m+1}^2}{n}$$

AICによる次数選択

$$\ell(\hat{\theta}) = -\frac{N}{2}\log 2\pi\hat{\sigma}_m^2 - \frac{N}{2}$$

$$\begin{aligned}\mathrm{AIC}_m &= -2\ell(\hat{\theta}) + 2(\text{パラメータ数}) \\ &= N(\log 2\pi\hat{\sigma}_m^2 + 1) + 2(m+1)\end{aligned}$$

$$\begin{bmatrix} s_{11} & \cdots & s_{1k} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & & \vdots & \vdots \\ & & s_{kk} & \cdots & s_{km} & s_{k,m+1} \\ & & & \ddots & \vdots & \vdots \\ & & & & s_{mm} & s_{m,m+1} \\ & & & & & s_{m+1,m+1} \\ & & 0 & & & \end{bmatrix}$$

for $k = 1,\ldots,m$

$$\hat{\sigma}_k^2 = \frac{1}{n}\left(s_{k+1,m+1}^2 + \cdots + s_{m+1,m+1}^2\right)$$

$$\mathrm{AIC}_k = N(\log 2\pi\hat{\sigma}_k^2 + 1) + 2(k+1)$$

$$\begin{bmatrix} s_{11} & \cdots & s_{1k} \\ & \ddots & \vdots \\ & & s_{kk} \end{bmatrix}\begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{k,m+1} \end{bmatrix}$$

（４）PARCOR法によるARモデルの推定

$$\hat{a}_j^m = \hat{a}_j^{m-1} - \hat{a}_m^m \hat{a}_{m-j}^{m-1} \qquad (j = 1, \ldots, m-1)$$

$$\hat{a}_m^m = (\hat{\sigma}_{m-1}^2)^{-1} \left\{ \hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_{m-j} \right\}$$

➡ Levinson's algorithm

$\hat{a}_m^m$ をデータから直接推定する

$$y_n = \sum_{i=1}^{m-1} a_j^{m-1} y_{n+j} + w_n^{m-1}$$

$$\begin{aligned} C_m - \sum_{j=1}^{m-1} a_j^{m-1} C_{m-j} &= E\left\{ \left(y_n - \sum_{i=1}^{m-1} a_j^{m-1} y_{n-j} \right) y_{n-m} \right\} \\ &= E\left(v_n^{m-1} y_{n-m} \right) \\ &= E\left(v_n^{m-1} w_{n-m}^{m-1} \right) \\ &\cong \frac{1}{N-m} \sum_{n=m+1}^{N} \left(v_n^{m-1} w_{n-m}^{m-1} \right) \end{aligned}$$

$$C_0 - \sum_{j=1}^{m-1} a_j^{m-1} C_j = E\left\{\left(y_{n-m} - \sum_{i=1}^{m-1} a_j^{m-1} y_{n-m+j}\right) y_{n-m}\right\}$$

$$= E\left(w_{n-m}^{m-1} y_{n-m}\right)$$

$$= E\left(w_{n-m}^{m-1}\right)^2$$

$$E\left(w_{n-m}^{m-1}\right)^2 = E\left(v_n^{m-1}\right)^2$$

$$\cong \begin{cases} \dfrac{1}{N-m} \displaystyle\sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 \\ \dfrac{1}{N-m} \left\{ \displaystyle\sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 \sum_{n=m+1}^{N} \left(v_n^{m-1}\right)^2 \right\}^{\frac{1}{2}} \\ \dfrac{1}{2(N-m)} \left\{ \displaystyle\sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 + \sum_{n=m+1}^{N} \left(v_n^{m-1}\right)^2 \right\} \end{cases}$$

$$a_m^m = \begin{cases} \displaystyle\sum_{n=m+1}^{N} v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 \right\}^{-1} & \text{Partial Regression} \\ \displaystyle\sum_{n=m+1}^{N} v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 \sum_{n=m+1}^{N} \left(v_n^{m-1}\right)^2 \right\}^{-\frac{1}{2}} & \text{PARCOR法} \\ \displaystyle\sum_{n=m+1}^{N} v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^{N} \left(w_{n-m}^{m-1}\right)^2 + \sum_{n=m+1}^{N} \left(v_n^{m-1}\right)^2 \right\}^{-1} & \text{Burg法（MEM）} \end{cases}$$

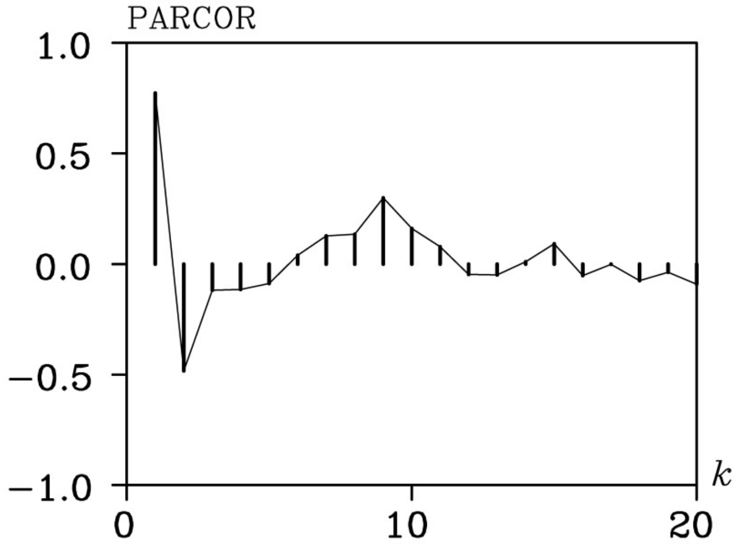
推定法の特徴

	定常性	数値精度	モデル精度	速度	自由度
Yule-Walker法	○	△	△	○	△
最尤法	◎	○	◎	×	○
最小二乗法	×	◎	○	◎	◎
PARCOR法	△	○	◎	◎	△

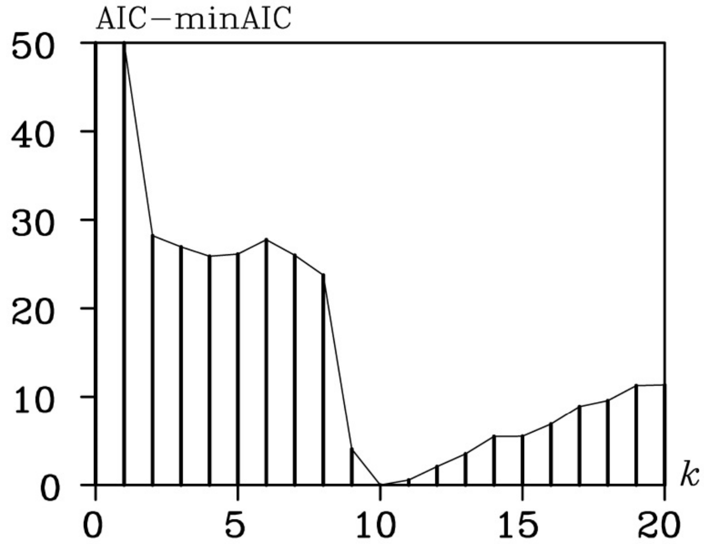
Yule-Walker法では実際は非定常な場合でも定常になる。

Householder最小二乗法でYule-Walker法の計算もできる。

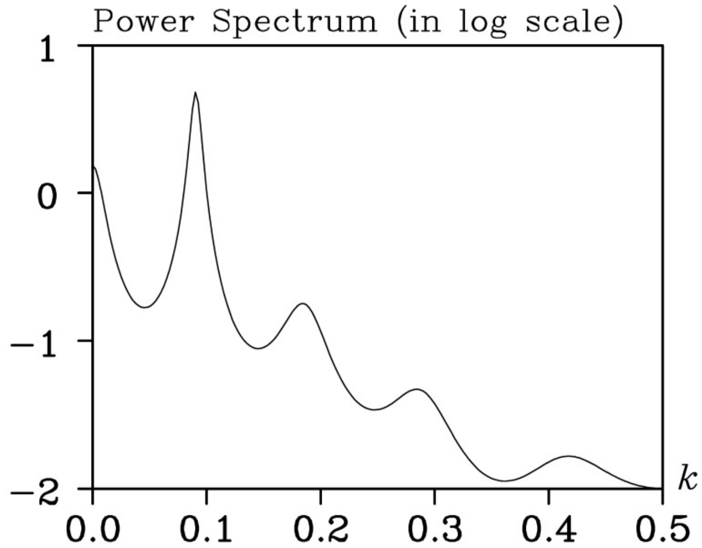
数値例　Candian Lynx data



(a)



(b)

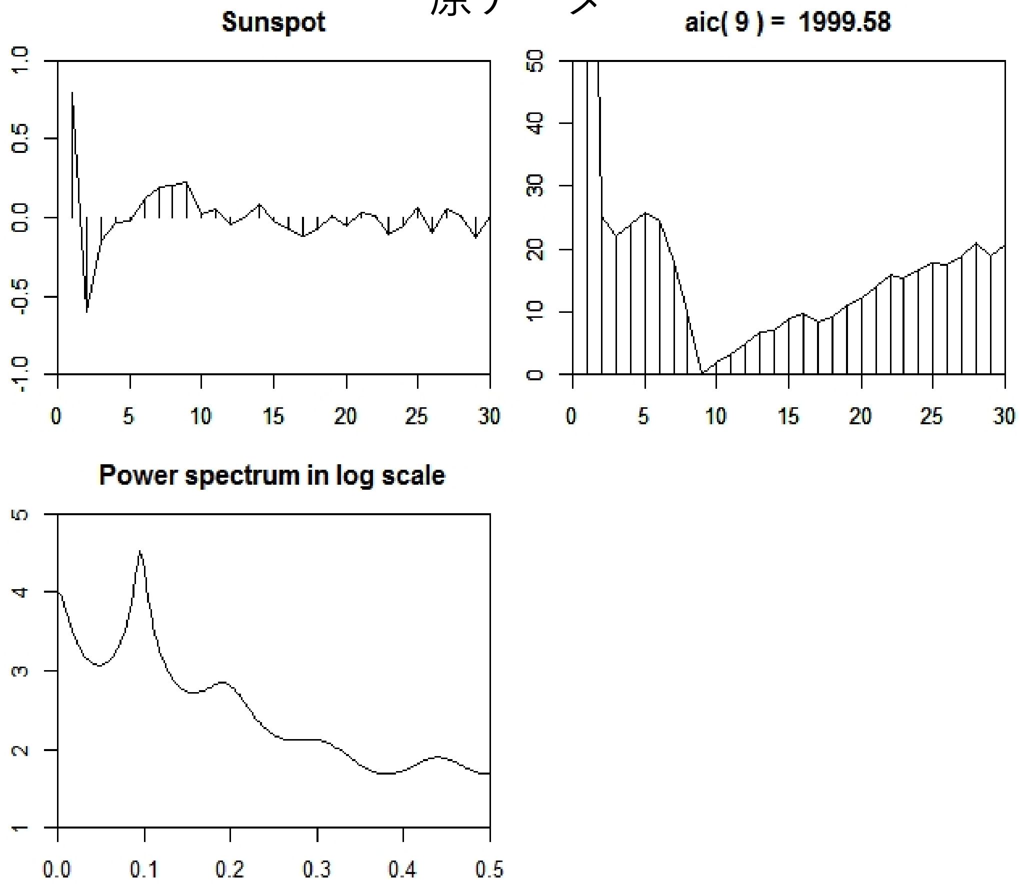


(c)

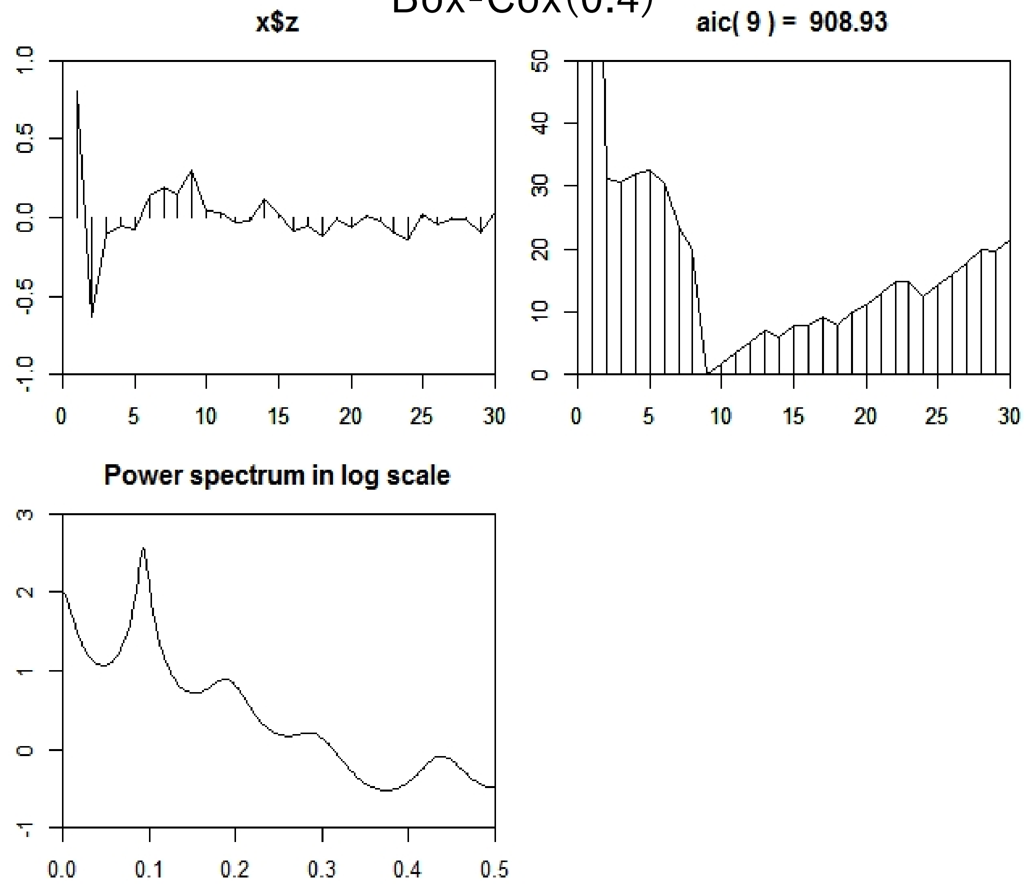
```
# Package TSSS
# AR model fitting for Candian Lynx data
# method=1 (default) Yule-Walker method
# method=2 Householder least squares method
# method=3 Parcor method (Partial autoregrssion)
# method=4 PARCOR
# method=5 Burg's algorithm (MEM)
#
arfit(lynx,method=2)
```

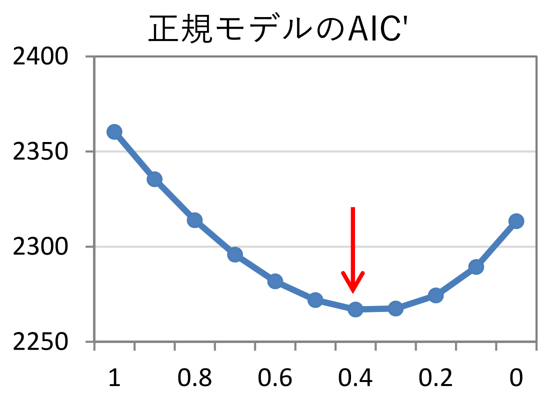
Sunspot data（Box-Cox変換の選択）

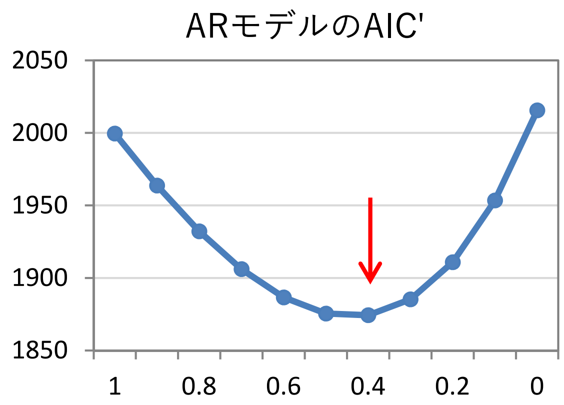
原データ



Box-Cox(0.4)

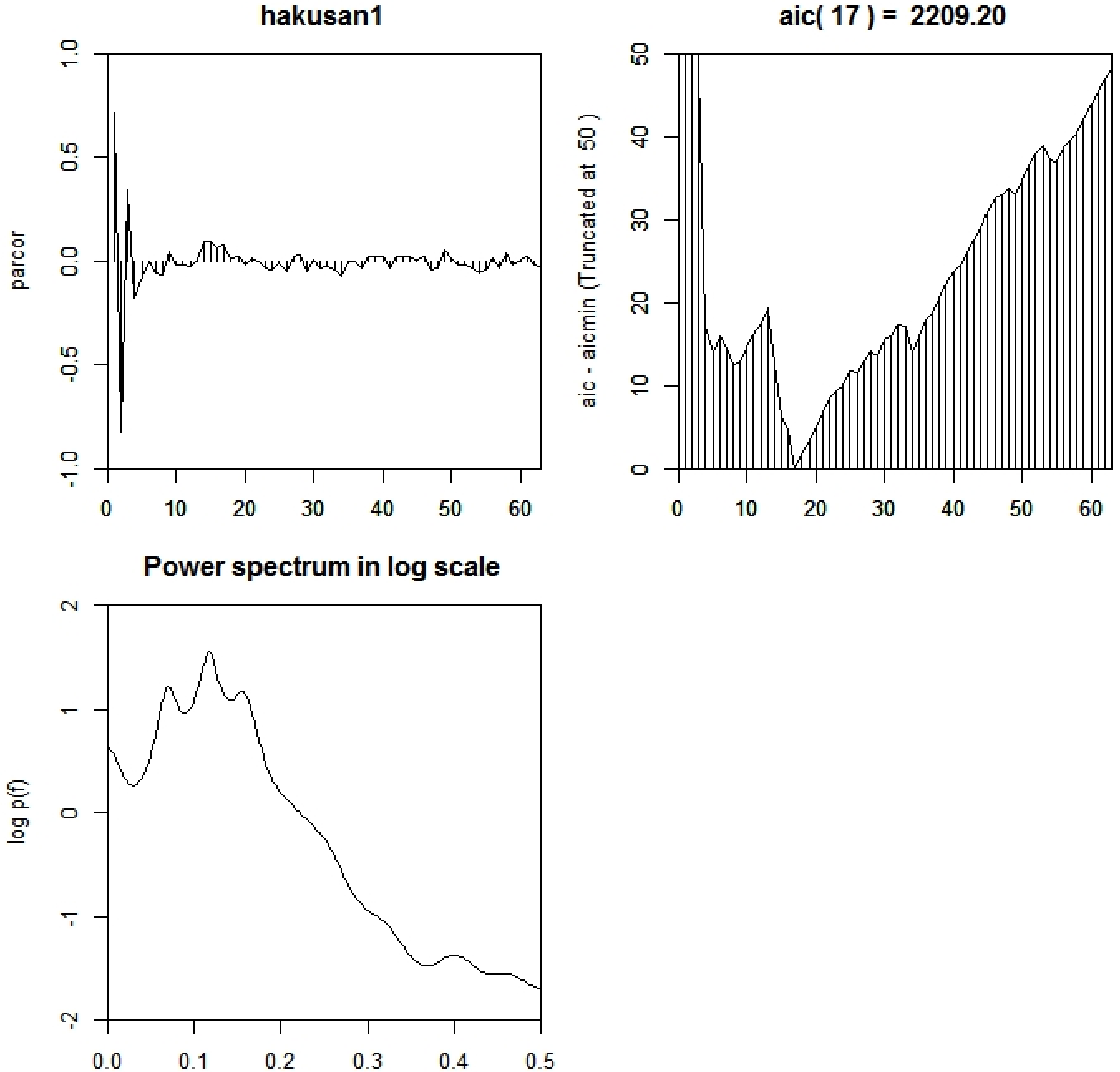






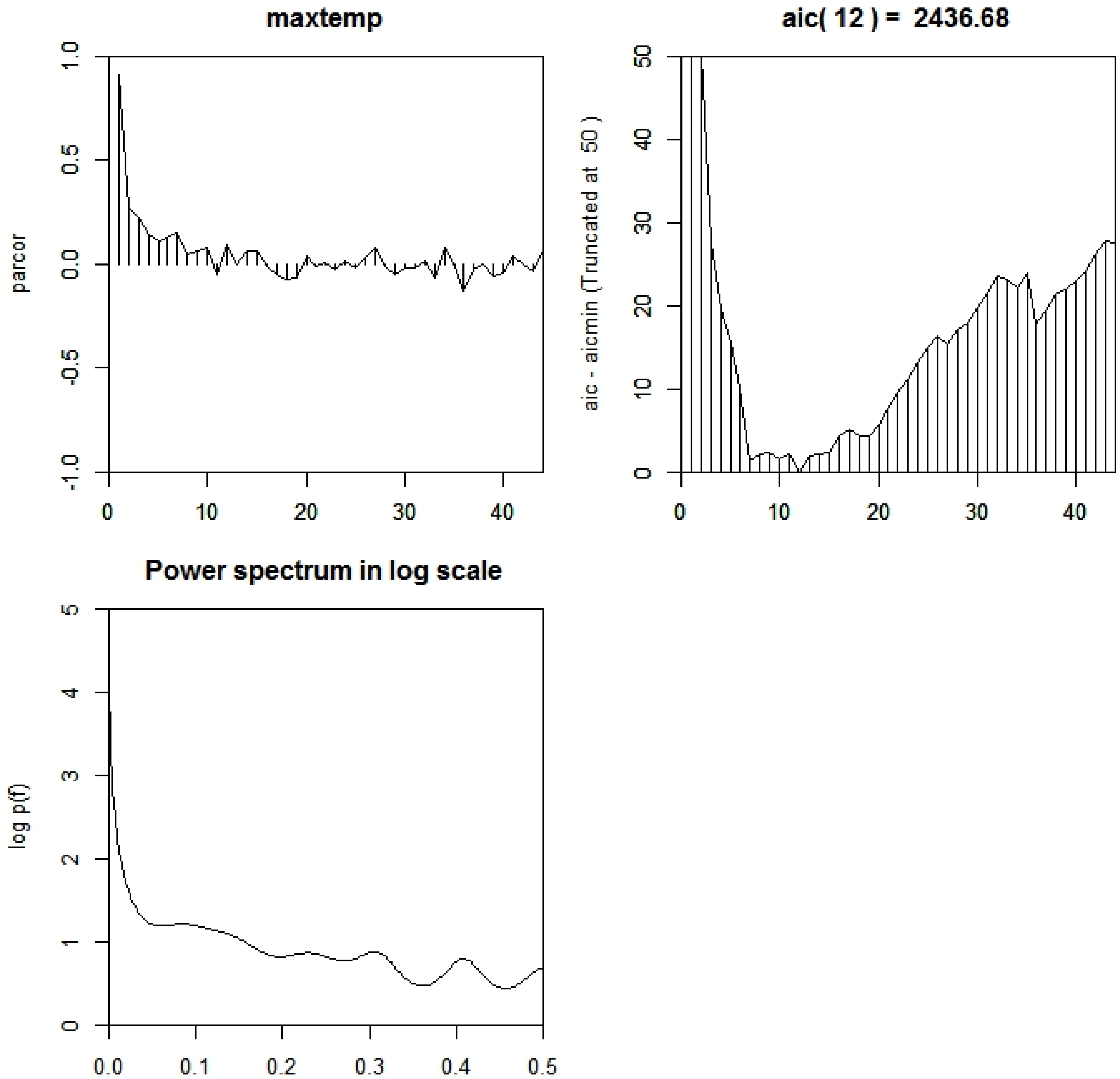
Hakusan (ship) data

arfit(hakusan1)



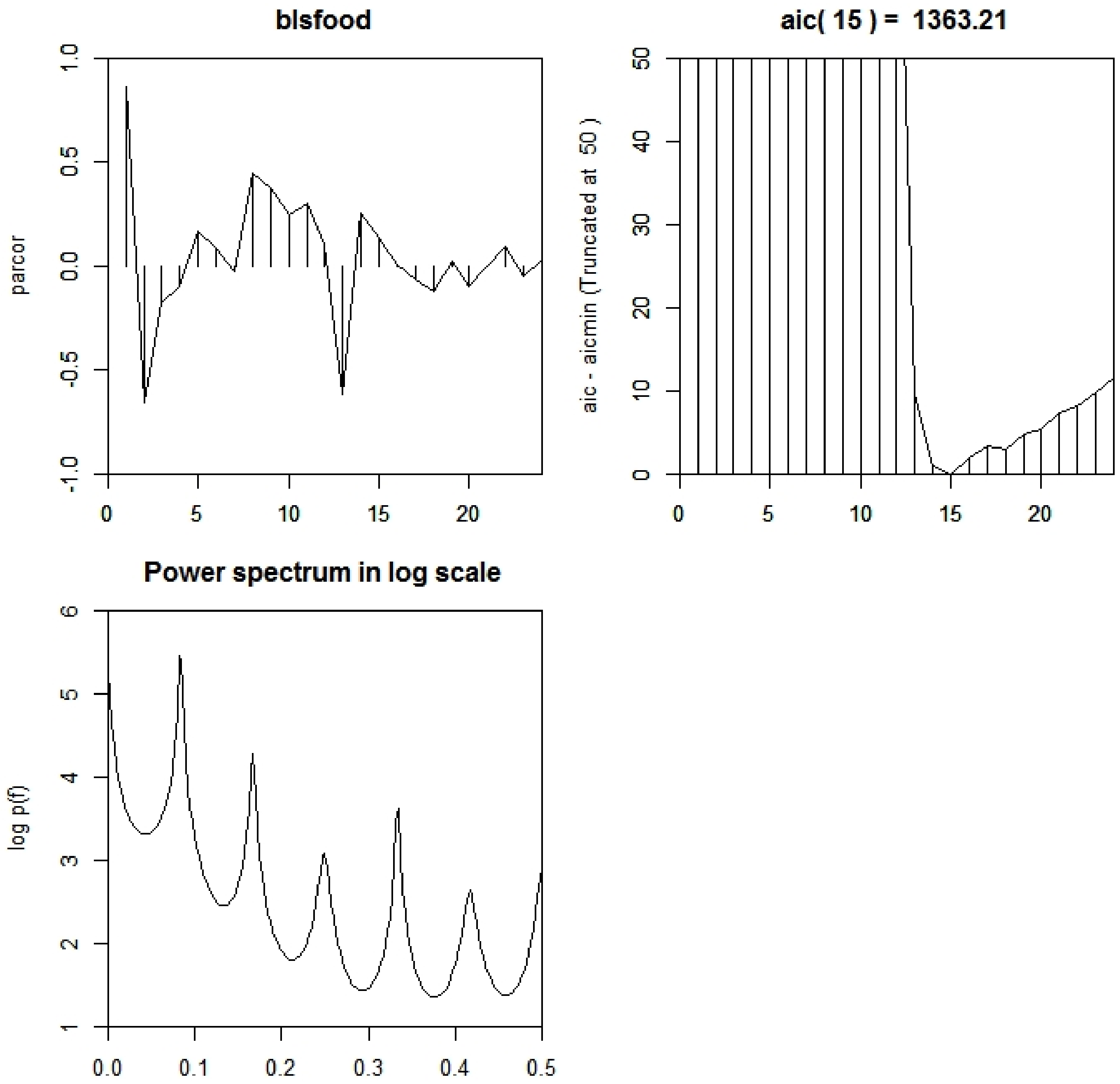
Daily maximum temperature data

arfit(maxtemp)



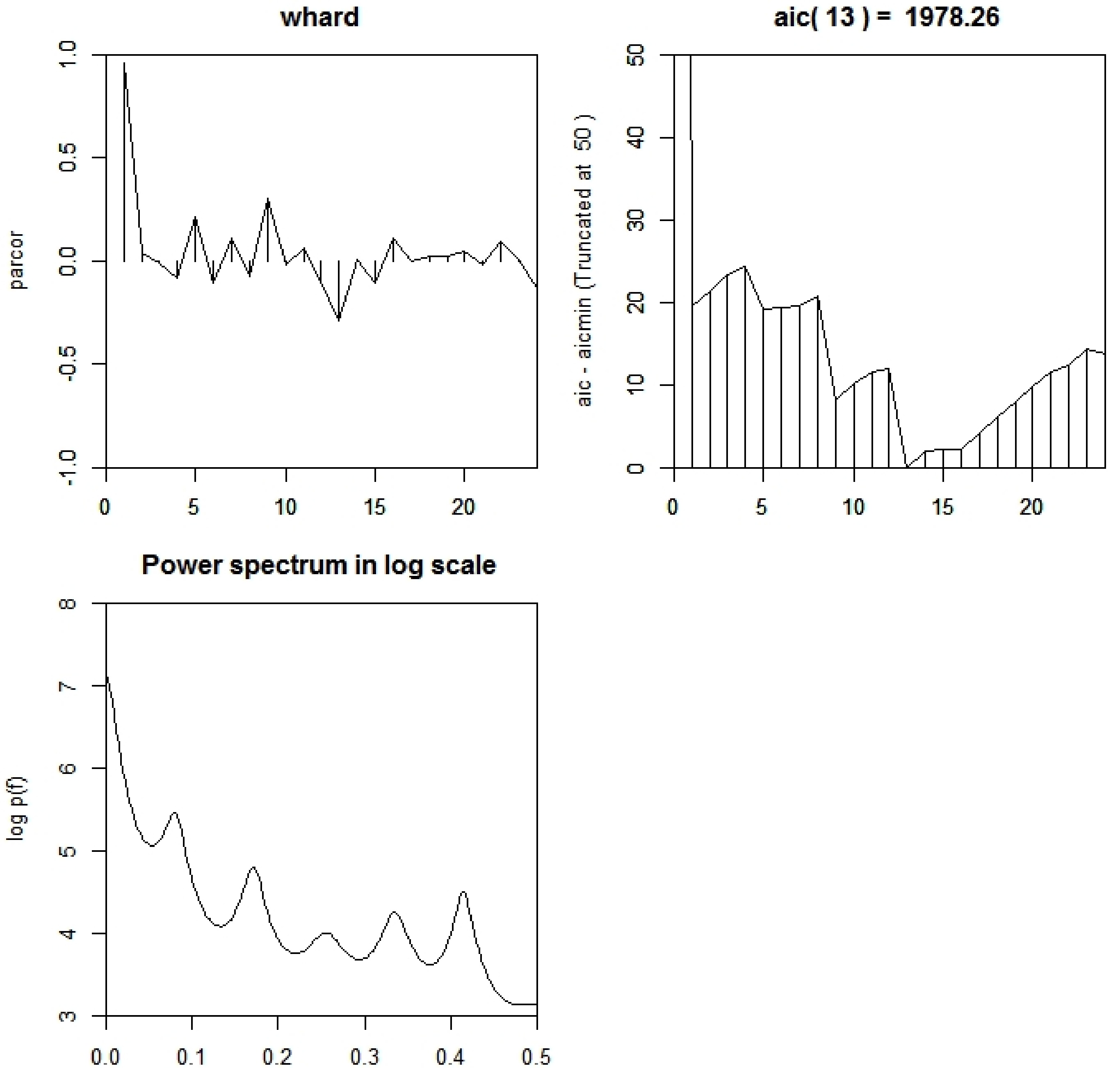
BLSFOOD data

arfit(blsfood)



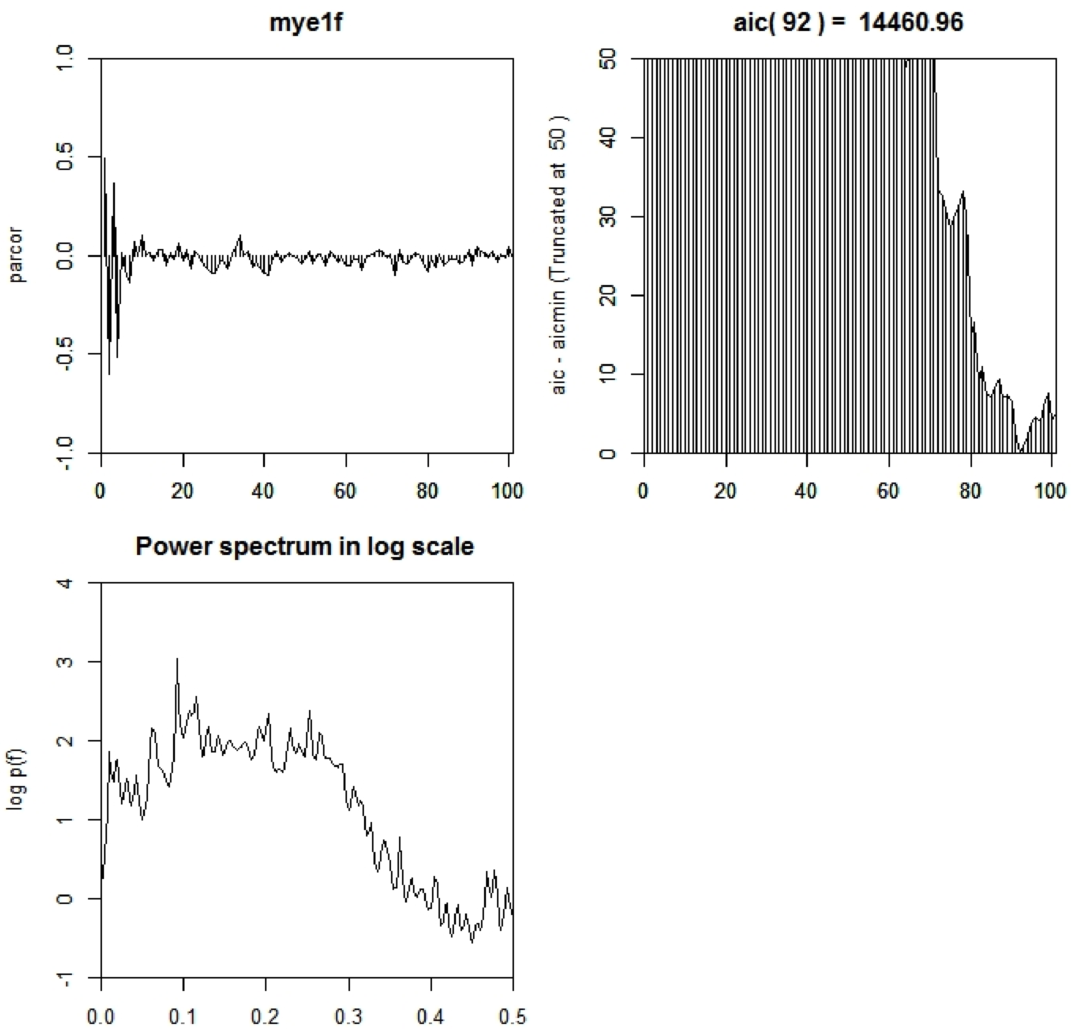
WHARD data

arfit(whard)

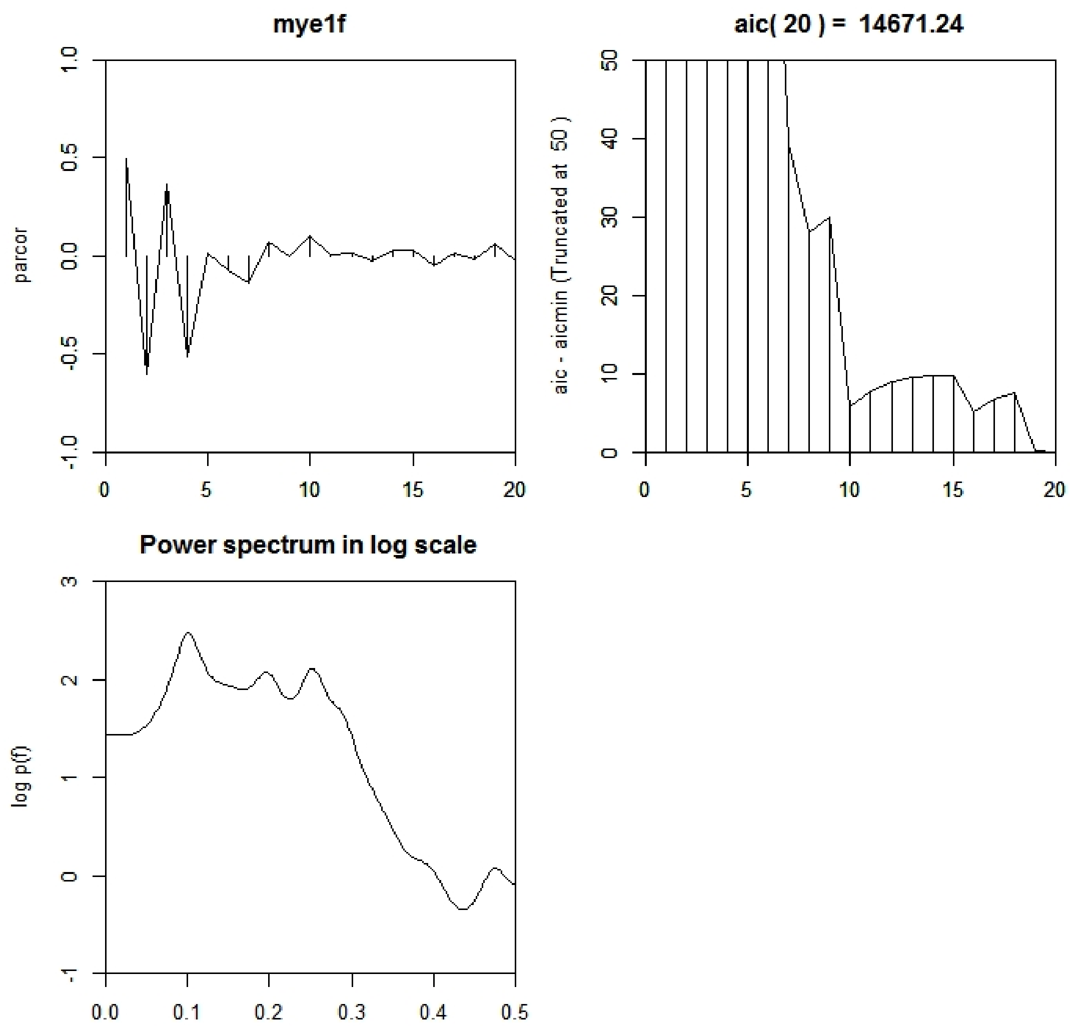


MYE1F earthquake data

arfit(mye1f)



arfit(mye1f,lag=20)



多変量ARモデルの推定：Yule-Walker法

$$y_n = \sum_{i=1}^{m} A_j^m y_{n-j} + v_n, \qquad v_n \sim N(0, V_m)$$

$$C_k = E\left[y_n y_{n-k}^T \right]$$

$$C_0 = \sum_{i=1}^{m} A_j C_{-j} + V$$

$$C_k = \sum_{i=1}^{m} A_j C_{k-j} \qquad (k = 1, 2, \cdots)$$

$$\begin{bmatrix} C_0 & C_{-1} & \cdots & C_{1-m} \\ C_1 & C_0 & \cdots & C_{2-m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m-1} & C_{m-2} & \cdots & C_0 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix}$$

多変量ARモデルの推定：Levinson - (Durbin)法

- 1変量ARモデルの場合，Levinsonアルゴリズムにより，Yule-Walker推定値を効率よく推定できる.
- アルゴリズムの導出には前向きモデルと後向きモデルを利用
- 多変量の場合にも同様のアルゴリズムがある．ただし, 前向きモデルと後向きモデルが異なるため 2倍のパラメータが必要

● 前向きモデル

$$y_n = \sum_{i=1}^{m} A_j^m y_{n-j} + v_n,$$

$$v_n \sim N(0, V_m)$$

● 後ろ向きモデル

$$y_n = \sum_{i=1}^{m} B_j^m y_{n+j} + u_n,$$

$$u_n \sim N(0, U_m)$$

● 推定するパラメータ

$$\{((A_j^m, B_j^m), j = 1, \ldots, m), V_m, U_m\}, m = 1, \ldots, M$$

パラメータ数 $\ell^2 (M(M+1)+2)$

多変量ARモデルの推定：Levinson - (Durbin)法

1. 0 次のMARモデル

$$V_0 = U_0 = C_0$$

$$\mathrm{AIC}_0 = N(k\log 2\pi + \log|V_0| + k) + k(k+1)$$

2. $m = 1,\ldots,M$ について

$$(a)\quad W_m = C_m - \sum_{j=1}^{m-1} A_j^{m-1} C_{m-j}$$

$$(b)\quad A_m^m = W_m U_{m-1}^{-1}, \qquad B_m^m = W_m^T V_{m-1}^{-1}$$

$$(c)\quad A_j^m = A_j^{m-1} - A_m^m B_{m-j}^{m-1}, \qquad B_j^m = B_j^{m-1} - B_m^m A_{m-j}^{m-1}$$

$$(c)\quad V_m = C_0 - \sum_{j=1}^{m} A_j^m C_j^T, \qquad U_m = C_0 - \sum_{j=1}^{m} B_j^m C_j$$

$$(d)\quad \mathrm{AIC}_m = N(k\log 2\pi + \log|V_m|\hat{\sigma}_m^2 + k) + k(k+1) + 2k^2 m$$

最小二乗法による多変量ARモデルの推定

$$y_n = \sum_{i=1}^{m} A_j y_{n-j} + v_n, \qquad v_n \sim N(0,V)$$

パラメータ数　$mk^2 + k(k+1)/2$

- 同時応答モデル

$$y_n = B_0 y_n + \sum_{i=1}^{m} B_j y_{n-j} + w_n, \qquad w_n \sim N(0,W)$$

$$B_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ b_0(2,1) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ b_0(k,1) & \cdots & b_0(k,k-1) & 0 \end{bmatrix}, \quad W = \begin{bmatrix} \sigma_1^2 & & & O \\ & \sigma_2^2 & & \\ & & \ddots & \\ O & & & \sigma_k^2 \end{bmatrix}$$

$$y_n = (I - B_0)^{-1} \sum_{i=1}^{m} B_j y_{n-j} + (I - B_0)^{-1} w_n,$$

$$A_j = (I - B_0)^{-1} B_j$$

$$V = (I - B_0)^{-1} W (I - B_0)^{-T}$$

- MARモデルと1対1対応
- Wは対角行列

Householder 法のメリット

W は対角行列 ➡ 行ごとに独立に推定できる

- モデリングが自由に行える。変数ごと（説明変数，目的変数）に異なる次数を決められる
- 計算量が減少する　$(mk^2)^3=m^3k^6$ ➡ $k(mk)^3=mk^4$

$$X=[Z|y]=\overbrace{\begin{bmatrix} y_m^T & \cdots & y_1^T & y_{m+1}^T \\ y_{m+1}^T & \cdots & y_2^T & y_{m+2}^T \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ y_{N-1}^T & \cdots & y_{N-m}^T & y_N^T \end{bmatrix}}^{k(m+1)},\ N\text{-}m$$

➡

$$HX=\begin{bmatrix} S \\ O \end{bmatrix}=\begin{bmatrix} s_{11} & \cdots & s_{1,km+k} & \cdots & s_{1,km+k} \\ & \ddots & \vdots & & \vdots \\ & & s_{1,km+k} & \cdots & s_{1,km+k} \\ & & & \ddots & \vdots \\ & & & & s_{1,km+k} \\ & & O & & \end{bmatrix}$$

（km，k）

$$\frac{mk^4}{m^3k^6}=\frac{1}{m^2k^2}=\frac{1}{5^2 10^2}=\frac{1}{2500}$$

k=10, m=5のとき

第1成分のモデル推定

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1,km} & s_{1,km+1} \\ & \ddots & \vdots & \vdots \\ & & s_{km,km} & s_{km,km+1} \\ & & & s_{km+1,km+1} \end{bmatrix} \quad \updownarrow km+1$$

$$\begin{array}{c} \xleftrightarrow{\quad kj \quad} \\ \xleftrightarrow{\qquad\qquad km+1 \qquad\qquad} \end{array}$$

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1,kj} & \cdots & s_{1,km} & s_{1,km+1} \\ & \ddots & \vdots & & \vdots & \vdots \\ & & s_{kj,kj} & \cdots & s_{kj,km} & s_{kj,km+1} \\ & & & \ddots & \vdots & \vdots \\ & & & & s_{km,km} & s_{km,km+1} \\ & & & & & s_{km+1,km+1} \end{bmatrix}$$

$$y_n(1) = \sum_{i=1}^{j} b_i(1,1) y_{n-i}(1) + \cdots + \sum_{i=1}^{j} b_i(1,k) y_{n-i}(k) + w_n$$

$$\hat{\sigma}_j^2(1) = \frac{1}{N-m} \sum_{i=kj+1}^{km+1} s_{i,km+1}^2$$

$$\mathrm{AIC}_j(1) = (N-m)\log(2\pi\hat{\sigma}_j^2(1)+1) + 2(kj+1)$$

$$\begin{bmatrix} s_{11} & \cdots & s_{1,kj} \\ & \ddots & \vdots \\ & & s_{kj,kj} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{kj} \end{bmatrix} = \begin{bmatrix} s_{1,km+1} \\ \vdots \\ s_{kj,km+1} \end{bmatrix}$$

Householder変換

$ind = (ind_1, ..., ind_L)$ ：　変換前の行の高さ（非零成分）

$jnd = (jnd_1, ..., jnd_L)$ ：　変換後の行の高さ（非零成分）

$$\forall ind \text{ and } \forall jnd \ \exists H \text{ s.t. } H \bullet S(ind) = S(jnd)$$

$$S(ind) \quad \overset{H}{\Rightarrow} \quad S(jnd)$$

第 2 成分のモデル推定

$$
S=\begin{bmatrix}
s_{11} & \cdots & s_{1,km} & s_{1,km+1} & s_{1,km+2} & \cdots & s_{1,km+k} \\
s_{21} & \cdots & s_{2,km} & & s_{2,km+2} & \cdots & s_{2,km+k} \\
 & \ddots & \vdots & & \vdots & & \vdots \\
 & & s_{km+1,km} & & s_{km+1,km+2} & \cdots & s_{km+1,km+k} \\
 & & & & s_{km+2,km+2} & \cdots & s_{km+1,km+k} \\
 & & & & & \ddots & \vdots \\
 & & & & & & s_{km+k,km+k}
\end{bmatrix}
$$

$$
y_n(2)=\sum_{i=1}^{j} b_i(2,1)\,y_{n-i}(1)+\cdots+\sum_{i=1}^{j} b_i(2,k)\,y_{n-i}(k)+w_n
$$

$$
\hat{\sigma}_j^2(2)=\frac{1}{N-m}\sum_{i=kj+2}^{km+2} s_{i,km+2}^2
$$

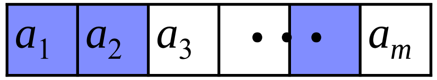
$$
\mathrm{AIC}_j(2)=(N-m)\log(2\pi\hat{\sigma}_j^2(2)+1)+2(kj+2)
$$

$$
\begin{bmatrix}
s_{11} & \cdots & s_{1,kj} & s_{1,km+1} \\
s_{21} & \cdots & s_{2,kj} & \\
 & \ddots & \vdots & \\
 & & s_{kj+1,kj} &
\end{bmatrix}
\begin{bmatrix}
c_1 \\ c_2 \\ \vdots \\ c_{kj+1}
\end{bmatrix}
=
\begin{bmatrix}
s_{1,km+2} \\ s_{2,km+2} \\ \vdots \\ s_{kj+1,km+2}
\end{bmatrix}
$$

変数選択の方法

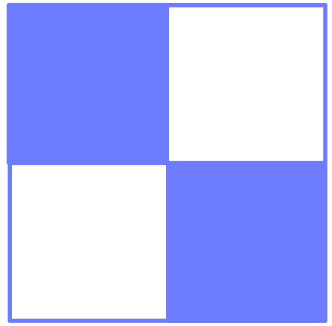
重回帰モデルの場合

$$y_n = \sum_{j=1}^{m} a_j x_{nj} + \varepsilon_n \quad \Rightarrow \quad y_n = \sum_{i=1}^{k} a_{j_i} x_{nj_i} + \delta_n$$



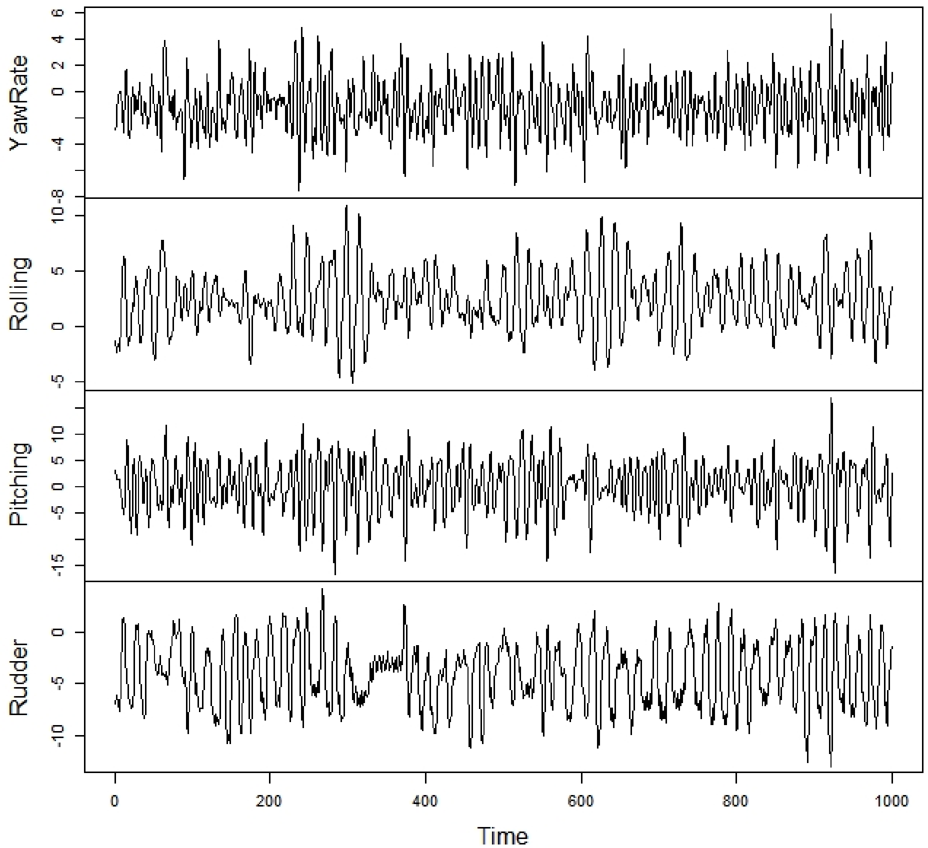
多変量時系列の場合

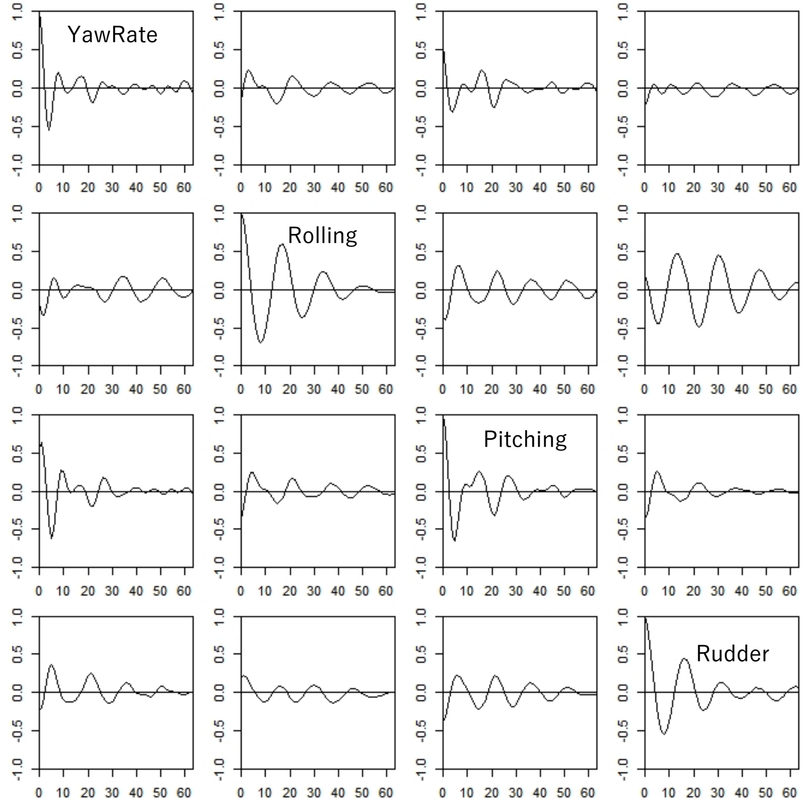
$$\begin{bmatrix} y_{n1} \\ y_{n2} \\ \vdots \\ y_{n\ell} \end{bmatrix} = \sum_{j=1}^{m} \begin{bmatrix} a_j(1,1) & a_j(1,2) & \cdots & a_j(1,\ell) \\ a_j(2,1) & a_j(2,2) & \cdots & a_j(2,\ell) \\ \vdots & \vdots & \ddots & \vdots \\ a_j(\ell,1) & a_j(\ell,2) & \cdots & a_j(\ell,\ell) \end{bmatrix} \begin{bmatrix} y_{n-1,1} \\ y_{n-1,2} \\ \vdots \\ y_{n-1,\ell} \end{bmatrix} + \begin{bmatrix} \varepsilon_{n1} \\ \varepsilon_{n2} \\ \vdots \\ \varepsilon_{n\ell} \end{bmatrix}$$



- 目的変数 ＝ 説明変数 ⇒ 変数を削除すると比較評価ができない
- ２つ以上のグループにわけて，AICの和を比較する

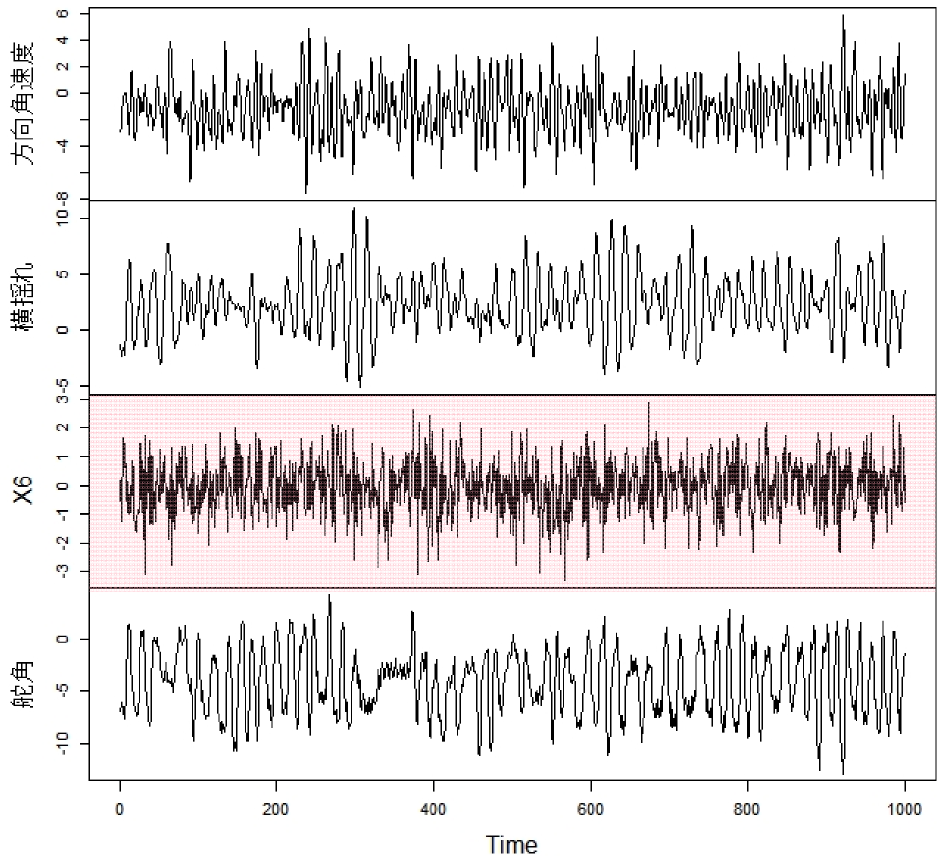
船舶データ（Yaw,Roll,Pitch,Rudder）

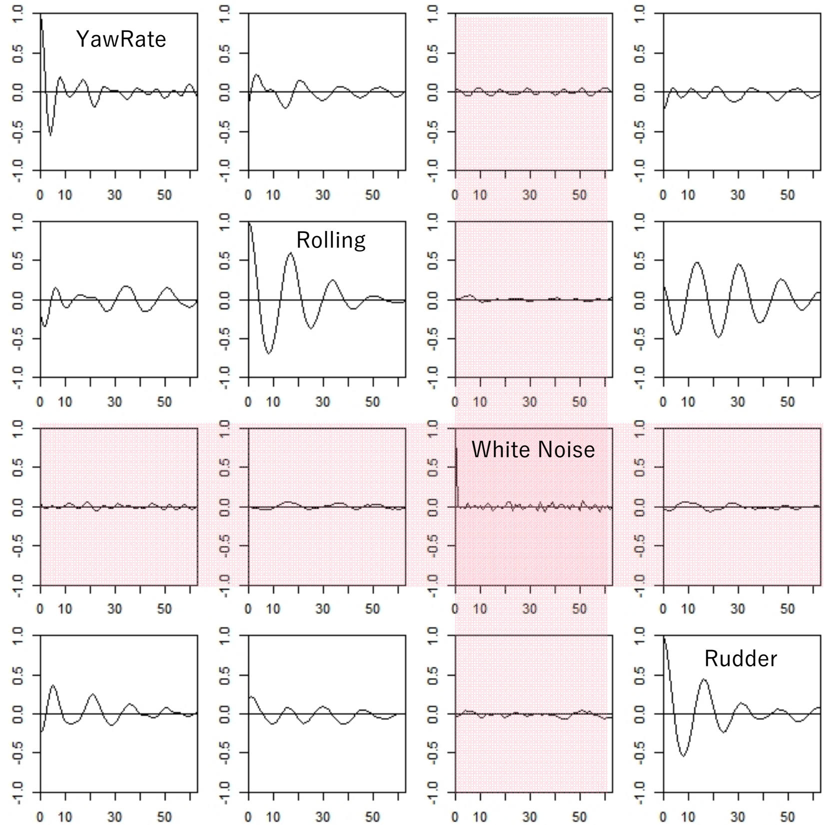




Y	R	P	Ru	AIC1					AIC2	AIC
■	■	■	■	9400.82						9400.82
■	■	■		6443.16				■	3085.15	9528.31
■	■		■	6658.67			■		2945.51	9604.18
■		■	■	7929.95		■			1866.90	9796.86
	■	■	■	7640.95	■				2209.20	9850.15
■	■			3734.66			■	■	5982.36	9717.03
■		■		4954.60		■		■	4834.74	9789.35
	■	■		4667.95	■			■	5166.17	9834.12

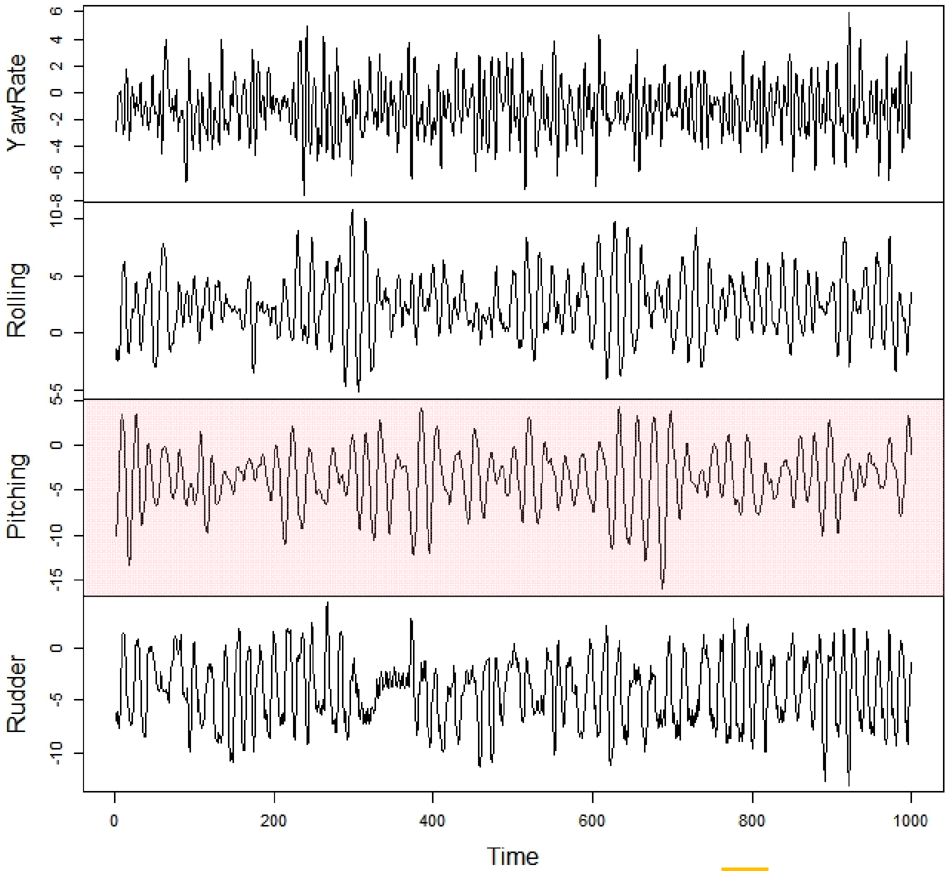
テストデータ（3ch: 白色雑音）

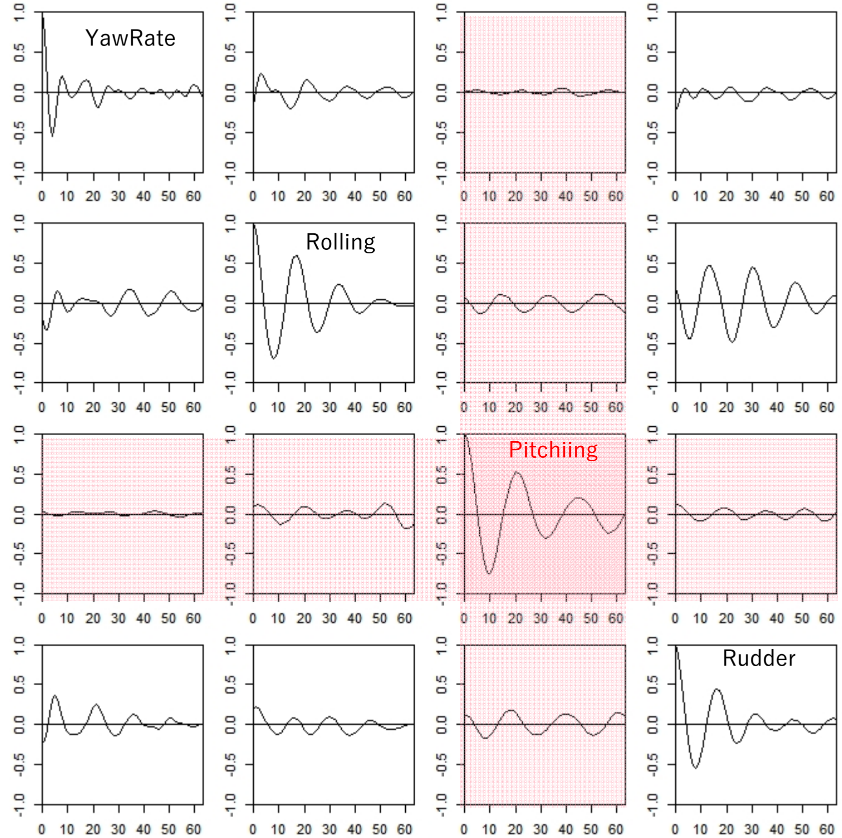




Y	R	W	Ru	AIC1					AIC2	AIC
■	■	■	■	9533.72						9533.72
■	■	■		6608.60				■	3085.15	9430.06
■	■		■	6658.67			■		2821.46	8525.57
■		■	■	8032.19		■			1866.90	10241.39
	■	■	■	7715.43	■				2209.20	9924.63
■	■			3734.66			■	■	5928.63	9663.29
■		■		5066.87		■		■	4834.74	9901.61
	■	■		4734.23	■			■	5166.17	9900.40

テストデータ（Pitchingを入れ替え）





Y	R	P	Ru	AIC1					AIC2	AIC
■	■	■	■	7777.53						7777.53
■	■	■		4848.33				■	3085.15	7933.48
■	■		■	6658.67			■		1112.85	7771.52
■		■	■	6301.14		■			1866.90	8168.04
	■	■	■	5971.24	■				2209.20	8180.44
■	■			3734.66			■	■	4213.76	7948.43
■		■		3337.37		■		■	4834.74	8172.11
	■	■		2979.00	■			■	5166.17	8145.17

FPE（Final Prediction Error）

推定したモデルによる予測誤差
＝　真のモデルによる予測誤差
×モデルの推定誤差の影響

$$\left.\begin{aligned} \mathrm{FPE}_m &= \left(1+\frac{m}{N}\right)\sigma^2 \\ \sigma^2 &= \frac{1}{1-\frac{m}{N}}\hat{\sigma}_m^2 \end{aligned}\right\} \Rightarrow \mathrm{FPE}_m = \frac{1+\frac{m}{N}}{1-\frac{m}{N}}\hat{\sigma}_m^2 = \frac{N+m}{N-m}\hat{\sigma}_m^2$$

情報量規準（AIC）との関係

$$\mathrm{AIC}_m = N \log \hat{\sigma}_m^2 + 2(m+1)$$

FPEとAICの関係

$$\begin{aligned}
N \log(\mathrm{FPE}_m) &= N \log\left(\frac{N+m}{N-m}\hat{\sigma}_m^2\right) \\
&= N \log\left(\frac{N+m}{N-m}\right) + N \log \hat{\sigma}_m^2 \\
&\approx N \log\left(1 + \frac{2m}{N}\right) + N \log \hat{\sigma}_m^2 \\
&\approx N\left(\frac{2m}{N}\right) + N \log \hat{\sigma}_m^2 = \mathrm{AIC}_m
\end{aligned}$$