Dynamic linear models

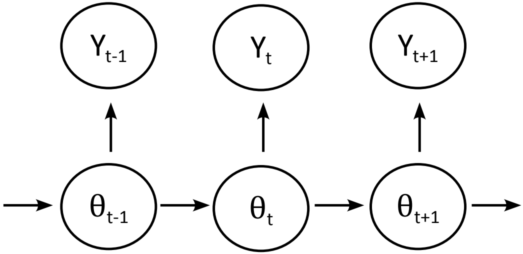
Dr. Jarad Niemi

STAT 615 - Iowa State University

October 10, 2017

Structure

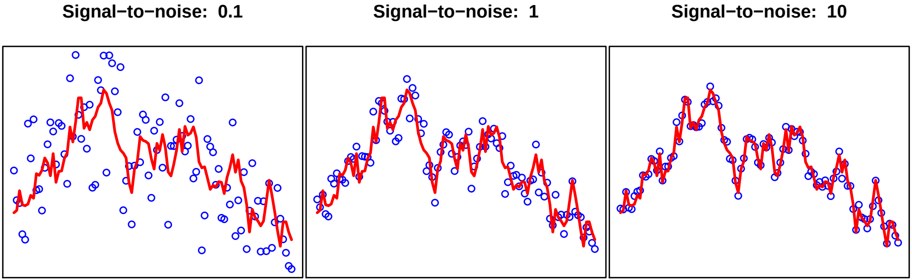
$$\begin{aligned}
Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, V_t) \\
\theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0, W_t) \\
\theta_0 &\sim N_p(m_0, C_0)
\end{aligned}$$



Local level model

$$\begin{aligned} Y_t &= \theta_t + v_t & v_t &\sim N_1(0, V) \\ \theta_t &= \theta_{t-1} + w_t & w_t &\sim N_1(0, W) \\ \theta_0 &\sim N_1(m_0, C_0) \end{aligned}$$

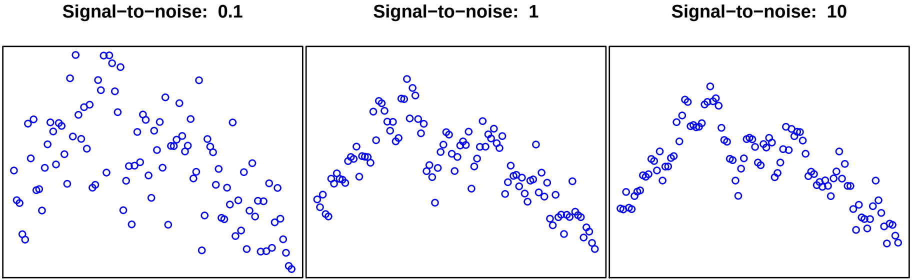
Signal-to-noise, $r = W/V$.



Local level model

$$
\begin{aligned}
Y_t &= \theta_t + v_t & v_t &\sim N(0, V) \\
\theta_t &= \theta_{t-1} + w_t & w_t &\sim N(0, W) \\
p(\theta_0) &= N(m_0, C_0)
\end{aligned}
$$

Signal-to-noise, $r = W/V$.



Kalman filter idea

Goal: obtain $p(\theta_t|y_{1:t})$

Recursive procedure:

- Assume $p(\theta_{t-1}|y_{1:t-1})$ $\qquad p(\theta_{t-1}|y_{1:t-1}) = N(m_{t-1}, C_{t-1})$
- Prior for θ_t

$$p(\theta_t|y_{1:t-1}) = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|y_{1:t-1})d\theta_{t-1}$$

- One-step ahead predictive distribution for y_t

$$p(y_t|y_{1:t-1}) = \int p(y_t|\theta_t)p(\theta_t|y_{1:t-1})d\theta_t$$

- Filtered distribution for θ_t

$$p(\theta_t|y_{1:t}) = \frac{p(y_t|\theta_t)p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

- Prior for θ_t

$$
\begin{aligned}
p(\theta_t|y_{1:t-1}) &= \int N(\theta_t; G_t\theta_{t-1}, W_t) N(\theta_{t-1}; m_{t-1}, C_{t-1}) d\theta_{t-1} \\
&= \int \frac{1}{(2\pi)^{p/2}|W_t|^{1/2}} \exp\left(-\frac{1}{2}(\theta_t - G_t\theta_{t-1})^\top W_t^{-1}(\theta_t - G_t\theta_{t-1})\right) \\
&\quad \frac{1}{(2\pi)^{p/2}|C_{t-1}|^{1/2}} \exp\left(-\frac{1}{2}(\theta_{t-1} - m_{t-1})^\top C_{t-1}^{-1}(\theta_{t-1} - m_{t-1})\right) d\theta_{t-1} \\
&= N(a_t, R_t)
\end{aligned}
$$

- One-step ahead predictive distribution for y_t

$$
\begin{aligned}
p(y_t|y_{1:t-1}) &= \int p(y_t|\theta_t) p(\theta_t|y_{1:t-1}) d\theta_t \\
&= \int N(y_t; F_t\theta_t, V_t) N(\theta_t; a_t, R_t) d\theta_t \\
&= \int \frac{1}{(2\pi)^{m/2}|V_t|^{1/2}} \exp\left(-\frac{1}{2}(y_t - F_t\theta_t)^\top V_t^{-1}(y_t - F_t\theta_t)\right) \\
&\quad \frac{1}{(2\pi)^{p/2}|R_t|^{1/2}} \exp\left(-\frac{1}{2}(\theta_t - a_t)^\top R_t^{-1}(\theta_t - a_t)\right) d\theta_t \\
&= N(f_t, Q_t)
\end{aligned}
$$

- Filtered distribution for θ_t

$$
\begin{aligned}
p(\theta_t|y_{1:t}) &= \frac{p(y_t|\theta_t)p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} = \frac{N(y_t; F_t\theta_t, V_t)N(\theta_t; a_t, R_t)}{N(y_t; f_t, Q_t)} \\
&= N(m_t, C_t)
\end{aligned}
$$

Latent state prior

Assume $p(\theta_{t-1}|y_{1:t-1}) = N(m_{t-1}, C_{t-1})$. Find $p(\theta_t|y_{1:t-1})$.

Evolution equation: $\theta_t = G_t\theta_{t-1} + w_t$ where $w_t \sim N_p(0, W_t)$.

- CONAN $\implies$ $\theta_t|y_{1:t-1}$ is normal
- $E[\theta_t|y_{1:t-1}] = G_t m_{t-1} = a_t$
- $Var[\theta_t|y_{1:t-1}] = G_t C_{t-1} G_t^\top + W_t = R_t$
- $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$.

One-step ahead prediction

Prior is $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$. Find $p(y_t|y_{1:t-1})$.

Observation equation: $y_t = F_t\theta_t + v_t$ where $v_t \sim N_m(0, V_t)$.

- CONAN $\implies$ $y_t|y_{1:t-1}$ is normal
- $E[y_t|y_{1:t-1}] = F_t a_t = f_t$
- $Var[y_t|y_{1:t-1}] = F_t R_t F_t^\top + V_t = Q_t$
- $p(y_t|y_{1:t-1}) = N(f_t, Q_t)$.

Latent state posterior - linear regression approach

We have $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$. Find $p(\theta_t|y_{1:t})$.

Observation equation: $y_t = F_t\theta_t + v_t$ where $v_t \sim N_m(0, V_t)$.

- Linear regression $\implies$ $\theta_t|y_{1:t}$ is normal
- $Var[\theta_t|y_{1:t}] = (R_t^{-1} + F_t^\top V_t^{-1} F_t)^{-1} = C_t$
- $E[\theta_t|y_{1:t}] = C_t(R_t^{-1} a_t + F_t^\top V_t^{-1} y_t) = m_t$
- $p(\theta_t|y_{1:t}) = N(m_t, C_t)$.

Latent state posterior - multivariate normal approach

Prior is $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$ and $p(y_t|y_{1:t-1}) = N(f_t, Q_t)$. Find $p(\theta_t|y_{1:t})$.

Consider

$$p\left(\left[\begin{array}{c} y_t \\ \theta_t \end{array} \right] \middle| y_{1:t-1} \right) = N\left(\left[\begin{array}{c} f_t \\ a_t \end{array} \right], \left[\begin{array}{cc} Q_t & F_t R_t \\ R_t F_t^\top & R_t \end{array} \right] \right)$$

- MVN theory $\implies$ $\theta_t|y_{1:t-1}, y_t$ is normal
- $E[\theta_t|y_{1:t}] = a_t + R_t F_t^\top Q_t^{-1}(y_t - f_t) = m_t$
- $Var[\theta_t|y_{1:t}] = R_t - R_t F_t^\top Q_t^{-1} F_t R_t = C_t$
- $p(\theta_t|y_{1:t}) = N(m_t, C_t)$.

Kalman filter

- Assume $p(\theta_{t-1}|y_{1:t-1}) = N(m_{t-1}, C_{t-1})$.
- Obtain prior $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$ where

$$a_t = G_t m_{t-1} \quad \text{and} \quad R_t = G_t C_{t-1} G_t^\top + W_t.$$

- Obtain one step ahead predictive $p(y_t|y_{1:t-1}) = N(f_t, Q_t)$ where

$$f_t = F_t a_t \quad \text{and} \quad Q_t = F_t R_t F_t^\top + V_t.$$

- Obtain posterior $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ where

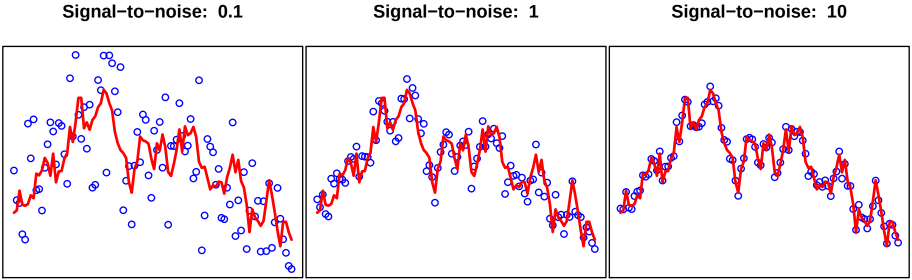
$$\begin{array}{rclcrcl} m_t &=& a_t + K_t e_t & \text{and} & C_t &=& R_t - K_t Q_t K_t^\top \\ e_t &=& y_t - f_t & \text{and} & K_t &=& R_t F_t^\top Q_t^{-1} \end{array}$$

K_t is the Kalman gain or adaptive coefficient.

Local level model

$$
\begin{aligned}
Y_t &= \theta_t + v_t & v_t &\sim N(0, V) \\
\theta_t &= \theta_{t-1} + w_t & w_t &\sim N(0, W) \\
p(\theta_0) &= N(m_0, C_0)
\end{aligned}
$$

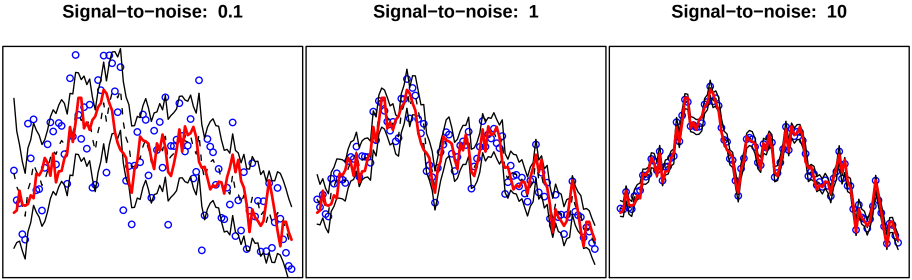
Signal-to-noise, $r = W/V$.



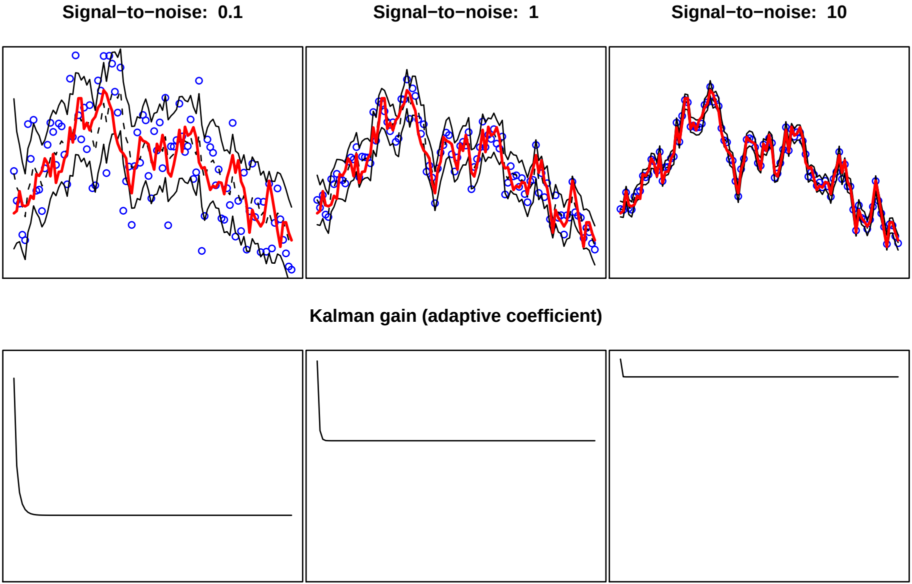
Local level model

$$
\begin{aligned}
Y_t &= \theta_t + v_t & v_t &\sim N(0, V) \\
\theta_t &= \theta_{t-1} + w_t & w_t &\sim N(0, W) \\
p(\theta_0) &= N(m_0, C_0)
\end{aligned}
$$

Signal-to-noise, $r = W/V$.



Kalman gain (adaptive coefficient)



Kalman filter

- Assume $p(\theta_{t-1}|y_{1:t-1}) = N(m_{t-1}, C_{t-1})$.
- Obtain prior $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$ where

$$a_t = G_t m_{t-1} \quad \text{and} \quad R_t = G_t C_{t-1} G_t^\top + W_t.$$

- Obtain one step ahead predictive $p(y_t|y_{1:t-1}) = N(f_t, Q_t)$ where

$$f_t = F_t a_t \quad \text{and} \quad Q_t = F_t R_t F_t^\top + V_t.$$

- Obtain posterior $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ where

$$\begin{array}{rclcrcl} m_t &=& a_t + K_t e_t & \text{and} & C_t &=& R_t - K_t Q_t K_t^\top \\ e_t &=& y_t - f_t & \text{and} & K_t &=& R_t F_t^\top Q_t^{-1} \end{array}$$

Kalman filter with missing data

- Assume $p(\theta_{t-1}|y_{1:t-1}) = N(m_{t-1}, C_{t-1})$.
- Obtain prior $p(\theta_t|y_{1:t-1}) = N(a_t, R_t)$ where

$$a_t = G_t m_{t-1} \quad \text{and} \quad R_t = G_t C_{t-1} G_t^\top + W_t.$$

- Obtain one step ahead predictive $p(y_t|y_{1:t-1}) = N(f_t, Q_t)$ where

$$f_t = F_t a_t \quad \text{and} \quad Q_t = F_t R_t F_t^\top + V_t.$$

- Obtain posterior $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ where
  - If y_t is not observed, $m_t = a_t$ and $C_t = R_t$.
  - If y_t is observed,

$$\begin{array}{rclcrcl} m_t &=& a_t + K_t e_t & \text{and} & C_t &=& R_t - K_t Q_t K_t^\top \\ e_t &=& y_t - f_t & \text{and} & K_t &=& R_t F_t^\top Q_t^{-1} \end{array}$$

Forecasting

Forecasting is simply the Kalman filter with missing observations. So,

$$\left(\begin{array}{c} \theta_{t+k} \\ Y_{t+k} \end{array} \right) \sim N \left(\left[\begin{array}{c} a_t(k) \\ f_t(k) \end{array} \right], \left[\begin{array}{cc} R_t(k) & F_{t+k}R_t(k) \\ R_t(k)F_{t+k}^\top & Q_t(k) \end{array} \right] \right)$$

where

$$\begin{array}{rl} a_t(k) & = G_{t+k}a_t(k-1) \\ R_t(k) & = G_{t+k}R_t(k-1)G_{t+k}^\top + W_{t+k} \\ f_t(k) & = F_{t+k}a_t(k) \\ Q_t(k) & = F_{t+k}R_t(k)R_{t+k}^\top + V_t \end{array}$$

with $a_t(0) = m_t$ and $R_t(0) = C_t$.

Kalman Smoother

Smoothing can be accomplished in a manner similar to the Kalman filter via the Kalman smoother. If we have $\theta_{t+1}|y_{1:T} \sim N(s_{t+1}, S_{t+1})$, then $\theta_t|y_{1:T} \sim N(s_t, S_t)$ where

$$\begin{aligned} s_t &= m_t + C_t G_{t+1}^\top R_{t+1}^{-1}(s_{t+1} - a_{t+1}) \\ S_t &= C_t - C_t G_{t+1}^\top R_{t+1}^{-1}(R_{t+1} - S_{t+1}) R_{t+1}^{-1} G_{t+1} C_t. \end{aligned}$$

Backward sampling

Recall

- $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ is available for all t from filtering and
- $p(\theta_t|\theta_{t+1}, y_{1:T}) = N(h_t, H_T)$ with

$$\begin{aligned} H_t &= (C_t^{-1} + G_{t+1}^\top W_{t+1}^{-1} G_{t+1})^{-1} \\ h_t &= H_t(C_t^{-1} m_t + G_{t+1}^\top W_{t+1}^{-1} \theta_{t+1}) \end{aligned}$$

The algorithm is then

- Forward filter to obtain $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ for all t.
- Sample $\theta_T \sim H(m_T, C_T)$.
- For $t = T-1, T-2, \ldots, 1, 0$,
  - Calculate h_t and H_t based on θ_{t+1}.
  - Draw $\theta_t \sim N(h_t, H_T)$.

This is then a joint draw of $\theta_{0:T} \sim p(\theta_0, \ldots, \theta_T|y_{1:T})$.

Inference questions?

Any questions on performing inference on the latent states in a DLM?

Decomposition of time series

Consider a univariate series Y_t. Think of this series has being the sum of independent components

$$Y_t = Y_{1,t} + \cdots + Y_{h,t} = \sum_{i=1}^{h} Y_{i,t}$$

where each component has its own independent DLM (dynamic linear model),

$$\begin{aligned} Y_{i,t} &= F_{i,t}\theta_{i,t} + v_{i,t} & v_{i,t} &\sim N(0, V_{i,t}) \\ \theta_{i,t} &= G_{i,t}\theta_{i,t-1} + w_{i,t} & w_{i,t} &\sim N(0, W_{i,t}) \end{aligned}$$

Then

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\sim N(0, V_t) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N(0, W_t) \end{aligned}$$

Observation equation

Recall

$$
\begin{aligned}
Y_t &= Y_{1,t} + \cdots + Y_{h,t} \\
Y_{i,t} &= F_{i,t}\theta_{i,t} + v_{i,t} \qquad\qquad v_{i,t} \sim N(0, V_{i,t})
\end{aligned}
$$

$$
\begin{aligned}
Y_t &= \sum_{i=1}^h Y_{i,t} \\
&= \sum_{i=1}^h \left[F_{i,t}\theta_{i,t} + v_{i,t}\right] \\
&= \sum_{i=1}^h F_{i,t}\theta_{i,t} + \sum_{i=1}^h v_{i,t} \\
&= \sum_{i=1}^h F_{i,t}\theta_{i,t} + v_t \qquad\qquad v_t \sim N(0, V_t) \quad V_t = \sum_{i=1}^h V_{i,t} \\
&= F_t\theta_t + v_t
\end{aligned}
$$

where

$$
\theta_t = \left[\begin{array}{c} \theta_{1,t} \\ \vdots \\ \theta_{h,t} \end{array}\right] \qquad\qquad F_t = [F_{1,t}|\cdots|F_{h,t}]
$$

Evolution equation

Recall

$$
\begin{aligned}
\theta_t &= \begin{bmatrix} \theta_{1,t} \\ \vdots \\ \theta_{h,t} \end{bmatrix} = \begin{bmatrix} G_{1,t}\theta_{1,t-1} + w_{1,t} \\ \vdots \\ G_{h,t}\theta_{h,t-1} + w_{h,t} \end{bmatrix} &= \begin{bmatrix} G_{1,t}\theta_{1,t-1} \\ \vdots \\ G_{h,t}\theta_{h,t-1} \end{bmatrix} + \begin{bmatrix} w_{1,t} \\ \vdots \\ w_{h,t} \end{bmatrix} \\
&= \begin{bmatrix} G_{1,t}\theta_{1,t-1} \\ \vdots \\ G_{h,t}\theta_{h,t-1} \end{bmatrix} + w_t &= G_t\theta_{t-1} + w_t
\end{aligned}
$$

where $w_t \sim N(0, W_t)$ and

$$
W_t = \begin{bmatrix} W_{1,t} & & \\ & \ddots & \\ & & W_{h,t} \end{bmatrix} \qquad G_t = \begin{bmatrix} G_{1,t} & & \\ & \ddots & \\ & & G_{h,t} \end{bmatrix}
$$

Combining model components

Consider

$$\begin{array}{ll} Y_t &= F_t\theta_t + v_t \qquad & v_t \sim N(0, V_t) \\ \theta_t &= G_t\theta_{t-1} + w_t \qquad & w_t \sim N(0, W_t) \end{array}$$

where $V_t = \sum_{i=1}^h V_{i,t}$,

$$\theta_t = \left[\begin{array}{c} \theta_{1,t} \\ \vdots \\ \theta_{h,t} \end{array}\right] \qquad F_t = [F_{1,t}|\cdots|F_{h,t}]$$

and

$$W_t = \left[\begin{array}{ccc} W_{1,t} & & \\ & \ddots & \\ & & W_{h,t} \end{array}\right] \qquad G_t = \left[\begin{array}{ccc} G_{1,t} & & \\ & \ddots & \\ & & G_{h,t} \end{array}\right]$$

Polynomial trend model definition

A polynomial model of order n is a DLM with constant [and specified] matrices $F_t = F$ and $G_t = G$, and a forecast function of the form

$$f_t(k) = E(Y_{t+k}|y_{1:t}) = a_{t,0} + a_{t,1}k + \cdots + a_{t,n-1}k^{n-1},\ k \geq 0$$

where $a_{t,0}, \ldots, a_{t,n-1}$ are linear functions of $m_t = E(\theta_t|y_{1:t})$ and are independent of k. Thus, the forecast function is a polynomial or order $n-1$ in k.

In practice we use,

- Local level model ($n = 1$).
- Linear trend model ($n = 2$).
- Exponential trends are accommodated by taking logs and then using a linear trend model.

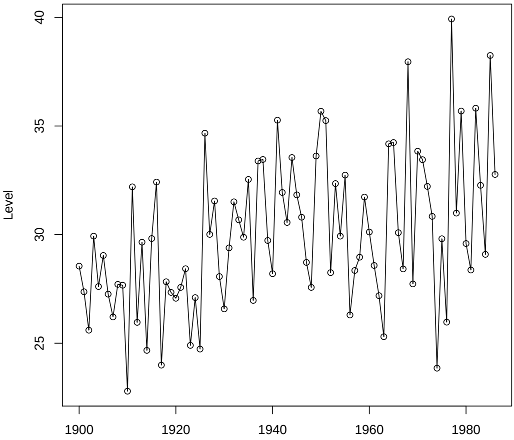
Local level model

$$\begin{aligned} Y_t &= \theta_t + v_t & v_t &\sim N(0, V) \\ \theta_t &= \theta_{t-1} + w_t & w_t &\sim N(0, W) \\ p(\theta_0) &= N(m_0, C_0) \end{aligned}$$

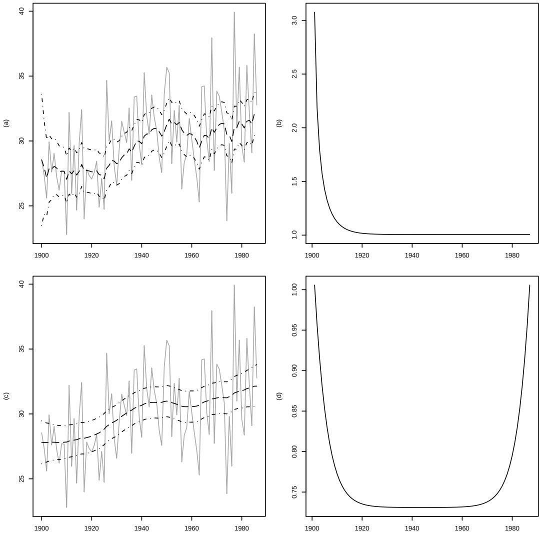
where $F_t = F = 1$, $G_t = G = 1$, $V_t = V$, and $W_t = W$.

What is the forecast function? $f_t(k) = E(Y_{t+k}|y_{1:t}) = m_t$.

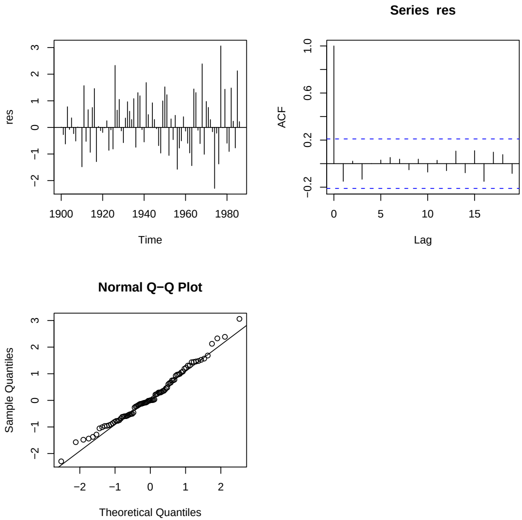
Lake Superior data



Lake Superior data



Lake Superior data

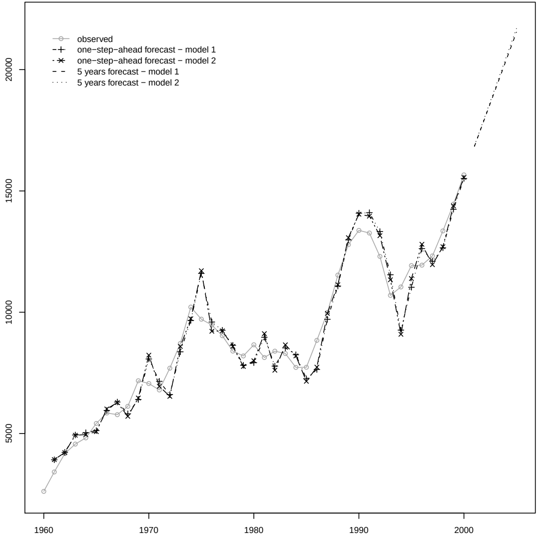


Linear trend model

$$
\begin{aligned}
Y_t &= \mu_t + v_t & v_t &\sim N(0, V) \\
\mu_t &= \mu_{t-1} + \beta_{t-1} + w_{t,1} & w_{t,1} &\sim N(0, \sigma^2_\mu) \\
\beta_t &= \beta_{t-1} + w_{t,2} & w_{t,2} &\sim N(0, \sigma^2_\beta) \\
p(\theta_0) &= N(m_0, C_0)
\end{aligned}
$$

- $F_t = F = (1, 0)$
- $\theta_t = (\mu_t, \beta_t)^\top$
- $$G_t = \left[\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array}\right]$$
- Forecast function $f_t(k) = E[Y_{t+k}|y_{1:t}] = \hat{\mu}_t + k\hat{\beta}_t$

Lake Superior data



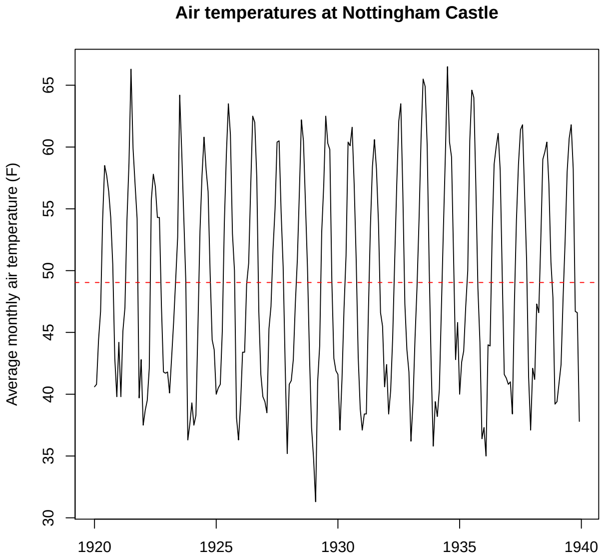
Polynomial trend models

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, V_t) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0, W_t) \\ p(\theta_0) &= N(m_0, C_0) \end{aligned}$$

- $F_t = F = (1, 0, \dots, 0)$
- $$G_t = G = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & & \cdots & 0 & 1 & 1 \\ 0 & & \cdots & & 0 & 1 \end{bmatrix}$$
- $W_t = W = \mathsf{diag}(W_1, \dots, W_n)$

Specifying a local level model in R

Example seasonal time series



Regression - air temp on month

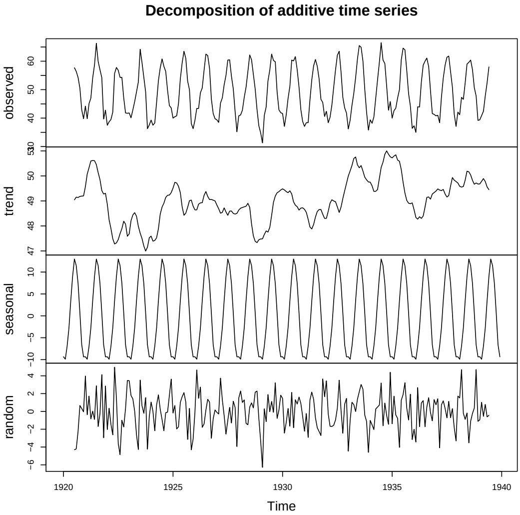
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  46.2900     0.5176  89.433  < 2e-16 ***
xAug         14.2300     0.7320  19.440  < 2e-16 ***
xDec         -6.7600     0.7320  -9.235  < 2e-16 ***
xFeb         -7.1000     0.7320  -9.700  < 2e-16 ***
xJan         -6.5950     0.7320  -9.010  < 2e-16 ***
xJul         15.6100     0.7320  21.325  < 2e-16 ***
xJun         11.7500     0.7320  16.052  < 2e-16 ***
xMar         -4.0950     0.7320  -5.594 6.32e-08 ***
xMay          6.2700     0.7320   8.566 1.62e-15 ***
xNov         -3.7100     0.7320  -5.068 8.29e-07 ***
xOct          3.2050     0.7320   4.378 1.82e-05 ***
xSep         10.1900     0.7320  13.921  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 2.315 on 228 degrees of freedom
Multiple R-squared: 0.9304,     Adjusted R-squared: 0.9271
F-statistic: 277.3 on 11 and 228 DF,  p-value: < 2.2e-16
```

No intercept regression - air temp on month

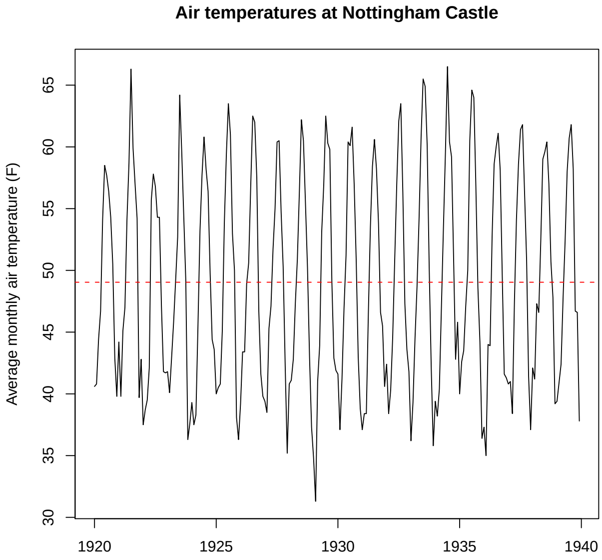
```
Coefficients:
     Estimate Std. Error t value Pr(>|t|)    
xApr  46.2900     0.5176   89.43   <2e-16 ***
xAug  60.5200     0.5176  116.93   <2e-16 ***
xDec  39.5300     0.5176   76.37   <2e-16 ***
xFeb  39.1900     0.5176   75.72   <2e-16 ***
xJan  39.6950     0.5176   76.69   <2e-16 ***
xJul  61.9000     0.5176  119.59   <2e-16 ***
xJun  58.0400     0.5176  112.13   <2e-16 ***
xMar  42.1950     0.5176   81.52   <2e-16 ***
xMay  52.5600     0.5176  101.55   <2e-16 ***
xNov  42.5800     0.5176   82.27   <2e-16 ***
xOct  49.4950     0.5176   95.62   <2e-16 ***
xSep  56.4800     0.5176  109.12   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 2.315 on 228 degrees of freedom
Multiple R-squared: 0.9979,     Adjusted R-squared: 0.9978 
F-statistic:  9231 on 12 and 228 DF,  p-value: < 2.2e-16
```

Time series decomposition

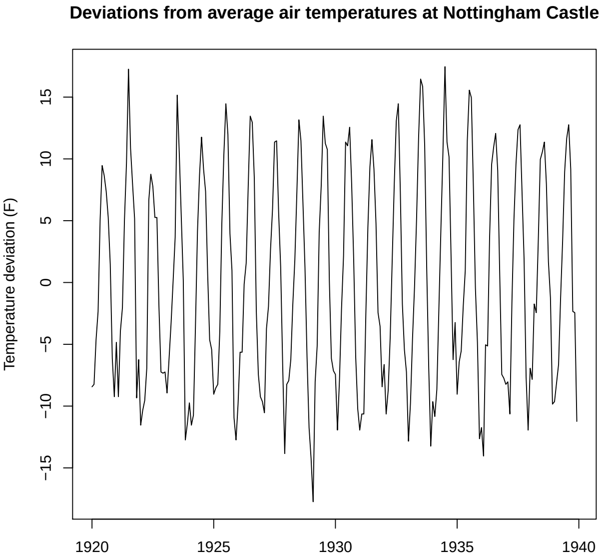
$>$ `plot(decompose(nottem))`



Example seasonal time series



Zero-mean seasonal time series



Seasonal factors

Suppose all s seasons have a factor α_i. Then

$$\begin{array}{rcl}
Y_1 &=& \alpha_1 + v_1 \\
Y_2 &=& \alpha_2 + v_2 \\
&\vdots& \\
Y_s &=& \alpha_s + v_s \\
Y_{s+1} &=& \alpha_1 + v_{s+1} \\
Y_{s+2} &=& \alpha_2 + v_{s+2} \\
&\vdots& \\
Y_{2s} &=& \alpha_s + v_{2s} \\
Y_{2s+1} &=& \alpha_1 + v_{2s+1} \\
Y_{2s+2} &=& \alpha_2 + v_{2s+2} \\
&\vdots&
\end{array}$$

$$\begin{array}{rcl}
Y_t &=& F_t\theta_t + v_t \\
\theta_t &=& G_t\theta_{t-1} + w_t
\end{array}$$

where

$$\begin{array}{rcl}
\theta_1 &=& (\alpha_1, \alpha_2, \dots, \alpha_s)^\top \\
F_t &=& F = (1, 0, 0, \dots, 0) \\
G_t &=& G = \left[\begin{array}{ccccc} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \vdots & 0 \end{array}\right] \\
W_t &=& W = 0 \quad \text{(maybe)}
\end{array}$$

Rotating factors

If $\theta_1 = (\alpha_1, \alpha_2, \ldots, \alpha_s)^\top$ and $W_t = 0$, what is θ_t?

$$\begin{array}{rcl} \theta_2 &=& G\theta_1 = (\alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_s, \alpha_1)^\top \\ \theta_3 &=& G\theta_2 = (\alpha_3, \alpha_4, \ldots, \alpha_s, \alpha_1, \alpha_2)^\top \\ &\vdots& \\ \theta_t &=& G\theta_{t-1} = (\alpha_j, \alpha_{j+1}, \ldots, \alpha_s, \alpha_1, \ldots, \alpha_{j-1})^\top \end{array}$$

where $j = t$ mod s ($j\%\%s$ in R).

Alternative DLM for seasonal factors

Suppose all s seasons have a factor α_i. Then

$$
\begin{aligned}
Y_1 &= \alpha_1 + v_1 \\
Y_2 &= \alpha_2 + v_2 \\
&\vdots \\
Y_s &= \alpha_s + v_s \\
Y_{s+1} &= \alpha_1 + v_{s+1} \\
Y_{s+2} &= \alpha_2 + v_{s+2} \\
&\vdots \\
Y_{2s} &= \alpha_s + v_{2s} \\
Y_{2s+1} &= \alpha_1 + v_{2s+1} \\
Y_{2s+2} &= \alpha_2 + v_{2s+2} \\
&\vdots
\end{aligned}
$$

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t \\
\theta_t &= G_t\theta_{t-1} + w_t
\end{aligned}
$$

where

$$
\begin{aligned}
\theta_1 &= (\alpha_1, \alpha_s, \alpha_{s-1}, \ldots, \alpha_2)^\top \\
F_t &= F = (1, 0, 0, \ldots, 0) \\
G_t &= G = \left[\begin{array}{ccccc}
0 & 0 & \cdots & 0 & 1 \\
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \vdots & 1 & 0
\end{array}\right] \\
W_t &= W = 0 \quad \text{(maybe)}
\end{aligned}
$$

Identifiability in seasonal factor models

- Modeling mean separately
- Looking at deviations from the mean

$\Longrightarrow$ parameter identifiability issue

- Identifiability constraints:
  - Set $\alpha_j = 0$ for some $j \in \{1, 2, \ldots, s\}$.
  - Sum-to-zero constraint, i.e. $\sum_{i=1}^{s} \alpha_i = 0$.

Parsimonious seasonal factor model

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t \\
\theta_t &= G_t\theta_{t-1} + w_t
\end{aligned}
$$

where

$$
\begin{aligned}
\theta_1 &= (\alpha_1, \alpha_s, \alpha_{s-1}, \ldots, \alpha_3)^\top \\
F_t &= F = (1, 0, \ldots, 0) \\
G_t &= G = \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \vdots & 1 & 0 \end{bmatrix} \\
W_t &= W = 0 \quad \text{(maybe)}
\end{aligned}
$$

What is θ_2 if $W = 0$?

$$
\begin{aligned}
\theta_2 &= G\theta_1 = (\alpha_2, \alpha_1, \alpha_s, \alpha_{s-1}, \ldots, \alpha_4)^\top \\
\theta_3 &= G\theta_2 = (\alpha_3, \alpha_2, \alpha_1, \alpha_s, \alpha_{s-1}, \ldots, \alpha_5)^\top \\
&\vdots \\
\theta_t &= G\theta_{t-1} \\
&= (\alpha_j, \alpha_{j-1}, \ldots, \alpha_1, \alpha_s, \alpha_{s-1}, \ldots, \alpha_{j+2})^\top
\end{aligned}
$$

where $j = t \bmod s$.

Seasonal factor model in R

```
##
## Call:
## lm(formula = y - mean(y) ~ x - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.890 -1.369  0.285  1.405  6.270
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## xJan  -9.3446     0.5176 -18.054  < 2e-16 ***
## xFeb  -9.8496     0.5176 -19.030  < 2e-16 ***
## xMar  -6.8446     0.5176 -13.224  < 2e-16 ***
## xApr  -2.7496     0.5176  -5.312 2.57e-07 ***
## xMay   3.5204     0.5176   6.802 9.00e-11 ***
## xJun   9.0004     0.5176  17.389  < 2e-16 ***
## xJul  12.8604     0.5176  24.847  < 2e-16 ***
## xAug  11.4804     0.5176  22.180  < 2e-16 ***
## xSep   7.4404     0.5176  14.375  < 2e-16 ***
## xOct   0.4554     0.5176   0.880     0.38
## xNov  -6.4596     0.5176 -12.480  < 2e-16 ***
## xDec  -9.5096     0.5176 -18.373  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.315 on 228 degrees of freedom
## Multiple R-squared:  0.9304, Adjusted R-squared:  0.9268
## F-statistic: 254.2 on 12 and 228 DF,  p-value: < 2.2e-16
```

Seasonal factor model in R

```
library(dlm)
nottem.model = dlmModSeas(12,dV=2.315^2,dW=rep(0,11))
nottem.filter = dlmFilter(y-mean(y),nottem.model)
n = length(y) + 1 # Due to theta_0
data.frame(mean = c(-sum(nottem.filter$m[n,]), rev(nottem.filter$m[n,])),
           lm_mean = m$coefficients)
```

```
##            mean    lm_mean
## xJan -9.3445833 -9.3445833
## xFeb -9.8495830 -9.8495833
## xMar -6.8445831 -6.8445833
## xApr -2.7495832 -2.7495833
## xMay  3.5204166  3.5204167
## xJun  9.0004164  9.0004167
## xJul 12.8604163 12.8604167
## xAug 11.4804164 11.4804167
## xSep  7.4404165  7.4404167
## xOct  0.4554167  0.4554167
## xNov -6.4595831 -6.4595833
## xDec -9.5095831 -9.5095833
```

Canonical basis

Suppose all s seasons have a factor α_i. Then, think of α as a linear combination of basis vectors

$$\alpha = (\alpha_1, \ldots, \alpha_s) = \sum_{i=1}^{s} \alpha_i u_i$$

where u_i is the s-dimensional vector having the ith component equal to one and all other elements zero.

- This representation lacks
  - Interpretation
  - Smoothness differentiation
  - Parsimony

Fourier frequencies

Let

$$\omega_j = 2\pi\frac{j}{s}, \quad j = 0, 1, \ldots, \frac{s}{2}$$

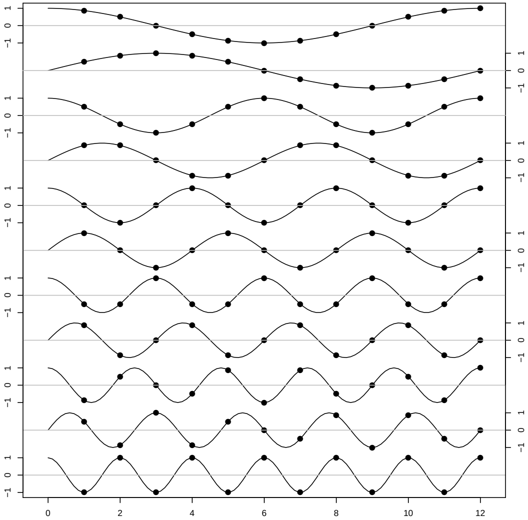
where s is the length of the period.

Consider

$$\begin{aligned}
e_0 &= (1, 1, \ldots, 1)^\top \\
c_1 &= (\cos\omega_1, \cos 2\omega_1, \ldots \cos s\omega_1)^\top \\
s_1 &= (\sin\omega_1, \sin 2\omega_1, \ldots \sin s\omega_1)^\top \\
&\vdots \\
c_j &= (\cos\omega_j, \cos 2\omega_j, \ldots \cos s\omega_j)^\top \\
s_j &= (\sin\omega_j, \sin 2\omega_j, \ldots \sin s\omega_j)^\top \\
&\vdots \\
c_{s/2} &= (\cos\omega_{s/2}, \cos 2\omega_{s/2}, \ldots \cos s\omega_{s/2})^\top
\end{aligned}$$

$s_{s/2}$ is all zeros

Plotting Fourier basis



Fourier basis

Basis for $\mathbb{R}^s$

$$\alpha = a_0 e_0 + \sum_{j=1}^{s/2-1} (a_j c_j + b_j s_j) + a_{s/2} c_{s/2}.$$

Assume $a_0 = 0$ since the mean will be modeled separately, e.g. through a polynomial trend model. For $j = 1, 2, \ldots, s/2$, the jth harmonic is

$$\begin{aligned} S_j(t) &= a_j \cos(t\omega_j) + b_j \sin(t\omega_j) \\ &= A_j \cos(t\omega_j + \gamma_j) \end{aligned}$$

where $b_{s/2} = 0$, $A_j = \sqrt{a_j^2 + b_j^2}$ is the amplitude, and $\gamma_j = \text{arctan}(-b_j/a_j)$ is the phase.

Evolving a harmonic

For $j = 1, 2, \ldots, s/2$, the jth harmonic is

$$S_j(t) = a_j \cos(t\omega_j) + b_j \sin(t\omega_j)$$

What is $S_j(t+1)$? It is

$$S_j(t+1) = a_j \cos([t+1]\omega_j) + b_j \sin([t+1]\omega_j).$$

For $j < s/2$, if we only know the value of $S_j(t)$, i.e. we don't know the value of a_j and b_j individual, we cannot determine $S_j(t+1)$. But we can find that

$$\begin{aligned} S_j(t+1) = & \quad a_j \cos([t+1]\omega_j) + b_j \sin([t+1]\omega_j) \\ & \quad \vdots \\ = & \quad (a_j \cos(t\omega_j) + b_j \sin(t\omega_j)) \cos(\omega_j) + \\ & \quad +(-a_j \sin(t\omega_j) + b_j \cos(t\omega_j)) \sin(\omega_j) \\ \\ = & \quad S_j(t) \cos(\omega_j) + S_j^*(t) \sin(\omega_j) \end{aligned}$$

Building a DLM - constructing G

For $j = 1, 2, \ldots, s/2$, the jth harmonic is

$$S_j(t) = a_j \cos(t\omega_j) + b_j \sin(t\omega_j)$$

The conjugate harmonic is

$$S_j^*(t) = -a_j \sin(t\omega_j) + b_j \cos(t\omega_j).$$

And the jth harmonic evolves in time according to

$$\left[\begin{array}{c} S_j(t+1) \\ S_j^*(t+1) \end{array}\right] = \left[\begin{array}{cc} \cos\omega_j & \sin\omega_j \\ -\sin\omega_j & \cos\omega_j \end{array}\right] \left[\begin{array}{c} S_j(t) \\ S_j^*(t) \end{array}\right]$$

and

$$S_{s/2}(t+1) = -S_{s/2}(t)$$

Period j DLMs

Consider DLMs containing only seasonality of frequency $j < s/2$.

$$\left[\begin{array}{c} S_j(t+1) \\ S_j^*(t+1) \end{array}\right] = \left[\begin{array}{rr} \cos\omega_j & \sin\omega_j \\ -\sin\omega_j & \cos\omega_j \end{array}\right] \left[\begin{array}{c} S_j(t) \\ S_j^*(t) \end{array}\right].$$

So this is a DLM with evolution matrix

$$H_j = \left[\begin{array}{rr} \cos\omega_j & \sin\omega_j \\ -\sin\omega_j & \cos\omega_j \end{array}\right],$$

state vector $(S_j(t), S_j^*(t))^\top$, and observation matrix $F = [1\,0]$.

If $j = s/2$, we have

$$S_{s/2}(t+1) = \cos((t+1)\pi) = -\cos(t\pi) = -S_{s/2}(t)$$

which is a DLM with state vector $S_{s/2}(t)$, evolution matrix $H_{s/2} = [-1]$, and observation matrix $F = [1]$.

Fourier form seasonal DLM

Combine these period j DLMs using our combining rules:

$$\theta_t = \left(S_1(t), S_1^*(t), \ldots, S_{\frac{s}{2}-1}(t), S^*_{\frac{s}{2}-1}(t), S_{\frac{s}{2}}(t)\right)^\top$$

with evolution matrix

$$G = \text{blockdiag}(H_1, \ldots, H_{\frac{s}{2}})$$

and observation matrix

$$F = [1\,0\,1\,0\,\ldots\,0\,1]$$

and initial vector

$$\theta_0 = (a_1, b_1, \ldots, a_{\frac{s}{2}-1}, b_{\frac{s}{2}-1}, a_{\frac{s}{2}})^\top.$$

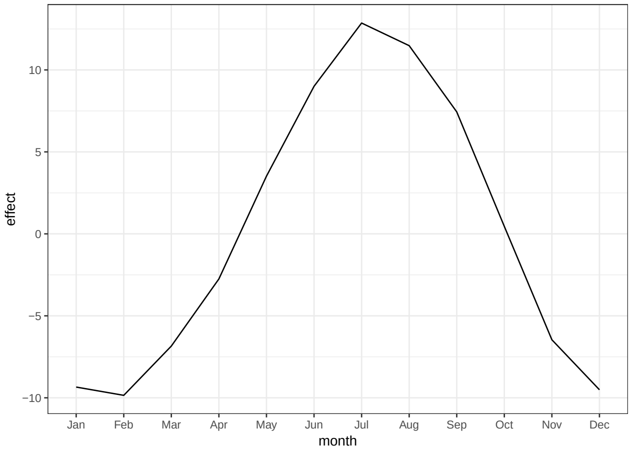
The seasonal process can be made more smooth by dropping higher harmonics.

Nottingham data

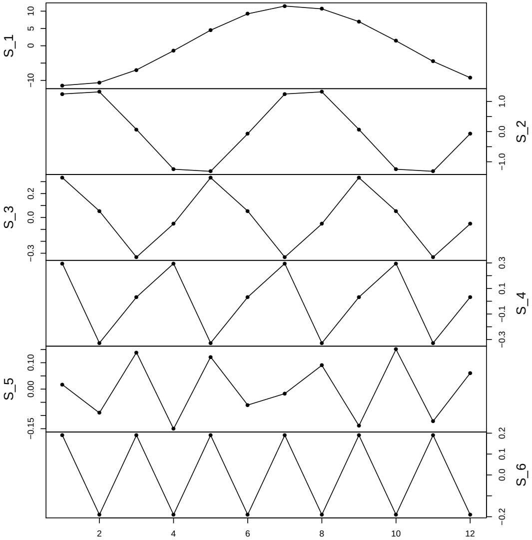
```
d = ddply(data.frame(harmonics=1:6), .(harmonics), function(x) {
  nottem.model = dlmModTrig(12,x$harmonics,dV=2.315^2,dW=rep(0,11))
  nottem.filter = dlmFilter(y-mean(y),nottem.model)
  forecast = dlmForecast(nottem.filter, 12)
  data.frame(month = factor(months, levels=months),
             effect = forecast$f)
})
```

Nottingham data

```
ggplot(subset(d, harmonics==6), aes(month, effect, group=1)) +
  geom_line() +
  theme_bw()
```

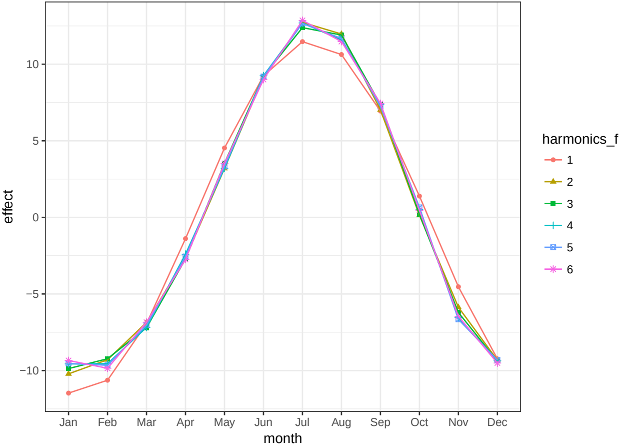


Plotting Harmonics for Nottingham data



Nottingham data - fewer harmonics

```
d$harmonics_f = factor(d$harmonics)
ggplot(d, aes(month, effect, color=harmonics_f, shape=harmonics_f, group=harmonics_f)) +
  geom_point() +
  geom_line() +
  theme_bw()
```



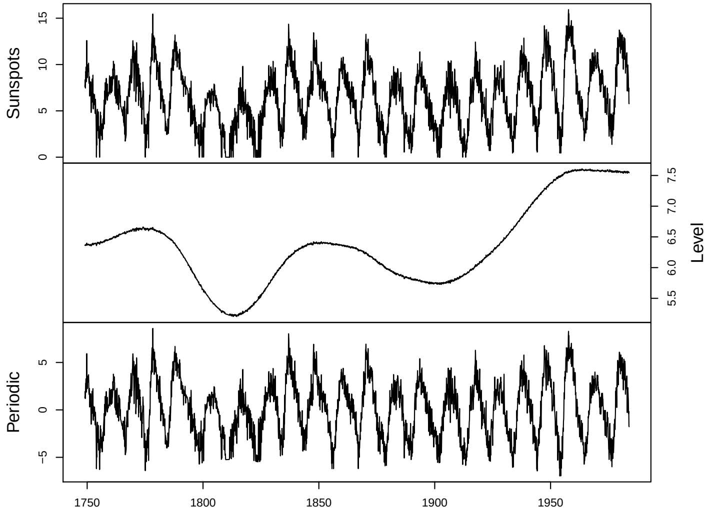
Sunspots

The previous development works for even period s and odd period s with the slight change that $j = 0, \ldots, (s-1)/2$. A similar development can be constructed for periodic observations with a non-integer period. In this example, the period for sunspots was estimated to be 130.51 months. This quantity and the variances were estimated via MLEs.

```
mod <- dlmModTrig(q = 2, tau = 130.51, dV = 0,
                  dW = rep(c(1765e-2, 3102e-4), each = 2)) +
        dlmModPoly(1, dV = 0.7452, dW = 0.1606)

sspots <- sqrt(sunspots)
sspots.smooth <- dlmSmooth(sspots, mod)
y <- cbind(sspots,
           tcrossprod(dropFirst(sspots.smooth$s[, c(1, 3, 5)]),
                      matrix(c(0, 0, 1, 1, 1, 0), nr = 2,
                             byrow = TRUE)))
colnames(y) <- c("Sunspots", "Level", "Periodic")
```

Sunspots

```
plot(y, yax.flip = TRUE, oma.multi = c(2, 0, 1, 0))
```



ARMA(p,q) [Box-Jenkins models]

Assuming $\mu = 0$, the autoregressive moving average model is :

$$Y_t = \sum_{j=1}^{p} \phi_j Y_{t-j} + \sum_{j=1}^{q} \psi_j \epsilon_{t-j} + \epsilon_t$$

$$Y_t = \sum_{j=1}^{r} \phi_j Y_{t-j} + \sum_{j=1}^{r-1} \psi_j \epsilon_{t-j} + \epsilon_t$$

where $r = \max\{p, q+1\}$ with $\phi_j = 0$ for $j > p$ and $\psi_j = 0$ for $j > q$ and $\epsilon_t \stackrel{ind}{\sim} N(0, \sigma^2)$.

DLM representation of ARMA model

Then an ARMA(p,q) is a DLM with

$$\begin{aligned} Y_t &= F\theta_t \\ \theta_t &= G\theta_{t-1} + R\epsilon_t \end{aligned}$$

with

$$\begin{aligned} V &= 0 \\ W &= RR^\top \sigma^2 \\ \theta_t &= (\theta_{1,t}, \ldots, \theta_{r,t})^\top \\ F &= (1, 0, \ldots, 0) \\ G &= \left[\begin{array}{ccccc} \phi_1 & 1 & 0 & \cdots & 0 \\ \phi_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \phi_{r-1} & 0 & \cdots & 0 & 1 \\ \phi_r & 0 & \cdots & 0 & 0 \end{array} \right] \\ R &= (1, \psi_1, \ldots, \psi_{r-2}, \psi_{r-1})^\top \end{aligned}$$

Verification of DLM representation of an ARMA model

$$
\begin{array}{rcl}
Y_t &=& \theta_{1,t} \\
\theta_{1,t} &=& \phi_1\theta_{1,t-1} + \theta_{2,t-1} + \epsilon_t \\
\theta_{2,t} &=& \phi_2\theta_{1,t-1} + \theta_{3,t-1} + \psi_1\epsilon_t \\
&\vdots& \\
\theta_{r-1,t} &=& \phi_{r-1}\theta_{1,t-1} + \theta_{r,t-1} + \psi_{r-2}\epsilon_t \\
\theta_{r,t} &=& \phi_r\theta_{1,t-1} + \psi_{r-1}\epsilon_t
\end{array}
$$

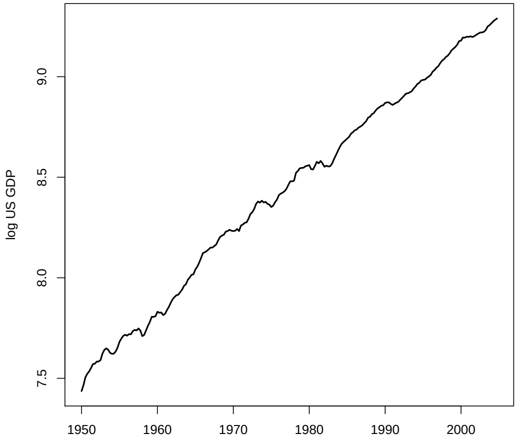
Verification of DLM representation of an ARMA model

$$
\begin{aligned}
Y_t &= \theta_{1,t} \\
\theta_{1,t} &= \phi_1\theta_{1,t-1} + \theta_{2,t-1} + \epsilon_t \\
\theta_{2,t-1} &= \phi_2\theta_{1,t-2} + \theta_{3,t-2} + \psi_1\epsilon_{t-1} \\
&\vdots \\
\theta_{r-1,t-(r-2)} &= \phi_{r-1}\theta_{1,t-(r-1)} + \theta_{r,t-(r-1)} + \psi_{r-2}\epsilon_{t-(r-2)} \\
\theta_{r,t-(r-1)} &= \phi_r\theta_{1,t-r} + \psi_{r-1}\epsilon_{t-(r-1)} \\
&\Longrightarrow \\
\theta_{1,t} &= \phi_1\theta_{1,t-1} + \phi_2\theta_{1,t-2} + \theta_{3,t-2} + \psi_1\epsilon_{t-1} + \epsilon_t \\
&\Longrightarrow \\
\theta_{1,t} &= \phi_1\theta_{1,t-1} + \cdots + \phi_r\theta_{1,t-r} + \psi_1\epsilon_{t-1} + \cdots + \psi_{r-1}\epsilon_{t-(r-1)} + \epsilon_t \\
Y_t &= \phi_1 Y_{t-1} + \cdots + \phi_r Y_{t-r} + \psi_1\epsilon_{t-1} + \cdots + \psi_{r-1}\epsilon_{t-(r-1)} + \epsilon_t \\
Y_t &= \sum_{j=1}^{r} \phi_j Y_{1,t-j} + \sum_{j=1}^{r-1} \psi_j \epsilon_{t-j} + \epsilon_t
\end{aligned}
$$

ARIMA(p,d,q)

- An ARIMA(p,d,q) can be fit by taking the dth order difference of the data and then applying an ARMA(p,q) model.
  - Clearly we can do the previous with the DLM representation.
- Alternatively, ARIMA models have a direct DLM representation (see section 3.2.5 in Petris)
- ARIMA models are typically used for non-stationary time series.
- In the DLM framework, modeling non-stationarity through a polynomial trend or seasonal model is more common than using the ARIMA framework.

Log US GDP



A model for Log US GDP

$$Y_t = Y_t^{(p)} + Y_t^{(g)}$$

$$\begin{aligned}
Y_t^{(p)} &= F^{(p)}\theta_t^{(p)} \\
\theta_t^{(p)} &= G^{(p)}\theta_{t-1}^{(p)} + w_t^{(p)}
\end{aligned}$$

$$\begin{aligned}
\theta_t^{(p)} &= (Y_t^{(p)}, \delta_t)^\top \\
F^{(p)} &= (1,0)^\top \\
G^{(p)} &= \left[\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array}\right] \\
W^{(p)} &= \text{diag}(\sigma_\epsilon^2, \sigma_z^2)
\end{aligned}$$

$$\begin{aligned}
Y_t^{(g)} &= F^{(g)}\theta_t^{(g)} \\
\theta_t^{(g)} &= G^{(g)}\theta_{t-1}^{(g)} + w_t^{(g)}
\end{aligned}$$

$$\begin{aligned}
\theta_t^{(g)} &= (Y_t^{(g)}, \theta_{t,2}^{(g)})^\top \\
F^{(g)} &= (1,0)^\top \\
G^{(g)} &= \left[\begin{array}{cc} \phi_1 & 1 \\ \phi_2 & 0 \end{array}\right] \\
W^{(g)} &= \text{diag}(\sigma_u^2, 0)
\end{aligned}$$

A model for Log US GDP

$$
\begin{aligned}
Y_t &= F\theta_t + v_t \\
\theta_t &= G\theta_{t-1} + w_t
\end{aligned}
$$

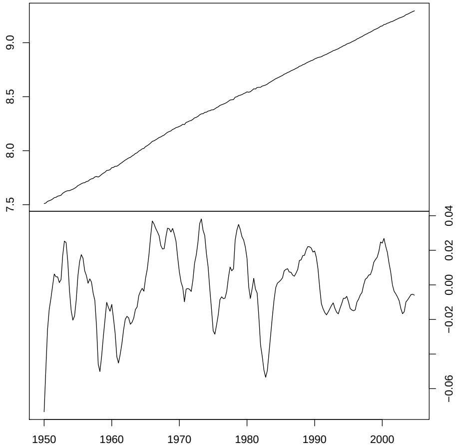
$$
\begin{aligned}
\theta_t &= (Y_t^{(p)}, \delta_t, Y_t^{(g)}, \theta_{t,2}^{(g)})^\top \\
F &= (1, 0, 1, 0) \\
G &= \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \phi_1 & 1 \\ 0 & 0 & \phi_2 & 0 \end{bmatrix} \\
V &= 0 \\
W &= \mathsf{diag}(\sigma_\epsilon^2, \sigma_z^2, \sigma_u^2, 0)
\end{aligned}
$$

MLE parameter estimates in Log US GDP model

```
> level0 <- Lgdp[1]
> slope0 <- mean(diff(Lgdp))
> buildGap <- function(u) {
    trend <- dlmModPoly(dV = 1e-7, dW = exp(u[1:2]),
                        m0 = c(level0, slope0), C0 = 2 * diag(2))
    gap <- dlmModARMA(ar = u[4:5], sigma2 = exp(u[3]))
    return(trend + gap)}
> init <- c(-3, -1, -3, .4, .4)
> outMLE <- dlmMLE(Lgdp, init, buildGap)
> dlmGap <- buildGap(outMLE$par)
> sqrt(diag(W(dlmGap))[1:3])
[1] 0.0057817835 0.0000763763 0.0061453639
> GG(dlmGap)[3:4, 3]
[1]  1.4806256 -0.5468107
```

MLE parameter estimates in Log US GDP model

```
gdpSmooth <- dlmSmooth(Lgdp, dlmGap)
```



Standard regression model

Consider a temporal regression problem with $y_t \in \mathbb{R}$ and $x_t \in \mathbb{R}^p$. A standard regression model assumes

$$y_t = x_t^\top \theta + \epsilon_t, \qquad \epsilon_t \stackrel{ind}{\sim} N(0, \sigma^2).$$

But, if the effect of x_t on y_t changes with t, we may want to consider the model

$$y_t = x_t^\top \theta_t + \epsilon_t, \qquad \epsilon_t \stackrel{ind}{\sim} N(0, \sigma^2).$$

And put some kind of evolution on θ_t, e.g.

$$\theta_t = G_t \theta_{t-1} + \omega_t, \qquad \omega_t \stackrel{ind}{\sim} N_p(0, W_t).$$

Outstanding problem: How can we differentiate dynamic coefficients from autocorrelated errors?

Dynamic regression model

This dynamic regression model written as a DLM is

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\stackrel{ind}{\sim} N(0,\sigma_t^2) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\stackrel{ind}{\sim} N_p(0,W_t) \end{aligned}$$

where

- $F_t = x_t^\top$,
- (typically) $G_t = I_p$, and
- (typically) W_t is diagonal.

Background

Use the following notation:

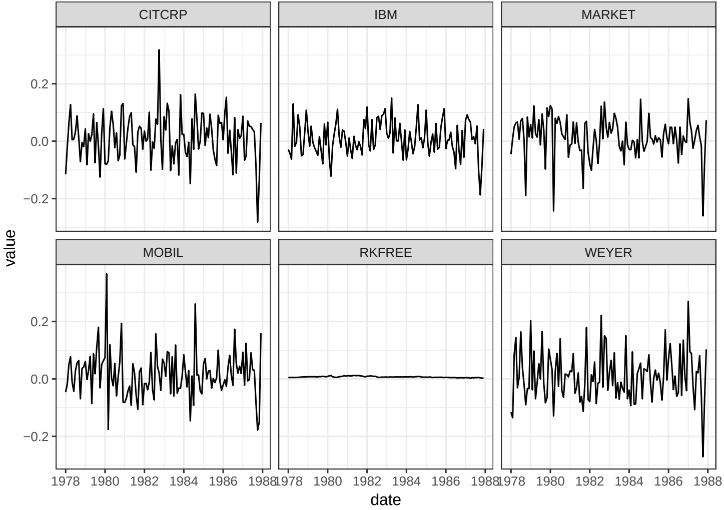
r_t : return at time t of the asset under study
$r_t^{(M)}$: return at time t of the market
$r_t^{(f)}$: return at time t of a risk free asset

Let

$$\begin{aligned} y_t &= r_t - r_t^{(f)} \\ x_t &= r_t^{(M)} - r_t^{(f)} \end{aligned}$$

A univariate capital asset pricing model (CAPM) assumes that

$$y_t = \alpha + \beta x_t + v_t, \qquad v_t \stackrel{ind}{\sim} N(0, \sigma^2).$$


CITCRP
IBM
MARKET
MOBIL
RKFREE
WEYER
value
0.2
0.0
−0.2
1978
1980
1982
1984
1986
1988
date

Static regression of IBM returns

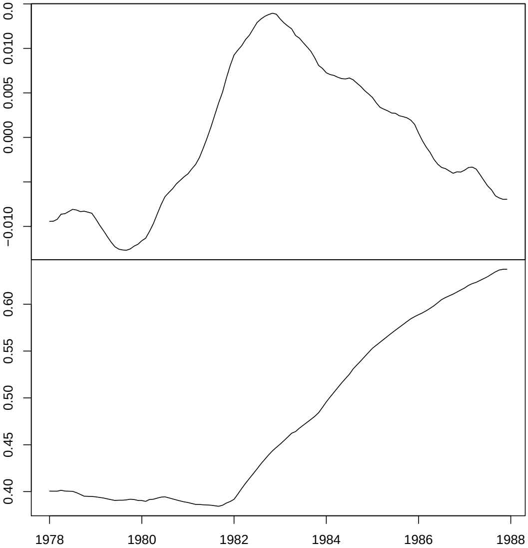
```
> outLM <- lm(IBM ~ x)
> outLM$coef
  (Intercept)             x
-0.0004895937  0.4568207721
>
> mod <- dlmModReg(x, dV = 0.00254, m0 = c(0, 0),
+                  C0 = diag(c(1e+07, 1e+07)))
> outF <- dlmFilter(IBM, mod)
> outF$m[1 + length(IBM), ]
[1] -0.0004895937  0.4568207719
```

Since the estimate for $\beta < 1$ the stock would be considered conservative.

Dynamic regression of IBM returns

```
> buildCapm <- function(u) {
+     dlmModReg(x, dV = exp(u[1]), dW = exp(u[2 : 3]))
+ }
> outMLE <- dlmMLE(IBM, parm = rep(0, 3), buildCapm)
> exp(outMLE$par)
[1] 2.328402e-03 1.100214e-05 6.495784e-04
> outMLE$value
[1] -276.7014
> mod <- buildCapm(outMLE$par)
> outS <- dlmSmooth(IBM, mod)
> plot(as.zoo(dropFirst(outS$s)), main = "",
       mar = c(0, 2.1, 0, 1.1),
       oma = c(2.1,0,.1,.1), cex.axis = 0.5)
```

CAPM example



Longitudinal data

Suppose at each time point, you have observations from multiple units, e.g. people, countries, stocks, etc. The data can be represented as a matrix.

	Time			
Item	1	2	$\cdots$	T
1	$Y_{1,1}$	$Y_{1,2}$	$\cdots$	$Y_{1,T}$
2	$Y_{2,1}$	$Y_{2,2}$	$\cdots$	$Y_{2,T}$
$\vdots$		$\vdots$		
m	$Y_{m,1}$	$Y_{m,2}$	$\cdots$	$Y_{m,T}$

Individual univariate DLMs

Suppose for each unit $i = 1, \ldots, m$, we can write down a univariate DLM:

$$\begin{aligned} Y_{i,t} &= F\theta_t^{(i)} + v_{i,t}, & v_{i,t} &\sim N(0, V_i) \\ \theta_t^{(i)} &= G\theta_{t-1}^{(i)} + w_t^{(i)}, & w_t^{(i)} &\sim N_p(0, W_i) \end{aligned}$$

- F and G are constant for all i and t
- V_i and W_i are constant for all t
- $\theta_t^{(i)}$ are still vectors, thus the notation
- Is there a relationship between V_i and V_j or W_i and W_j for $i \neq j$?
- How about $Cov(v_{i,t}, v_{j,t})$ or $Cov(w_t^{(i)}, w_t^{(j)})$ for $i \neq j$?

Linear trend models

Suppose you have a linear trend model for each unit i:

$$\begin{aligned} Y_{i,t} &= F_{i,t}\theta_t^{(i)} + v_{i,t} & \qquad v_{i,t} &\sim N_m(0, V_{i,t}) \\ \theta_{i,t} &= G_{i,t}\theta_{t-1}^{(i)} + w_t^{(i)} & \qquad w_t^{(i)} &\sim N_p(0, W_{i,t}) \end{aligned}$$

- What is $F_{i,t}$? $F_{i,t} = (1,0)^\top$
- What is $G_{i,t}$?

$$G_{i,t} = \left[\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array}\right]$$

- What is $V_{i,t}$? $V_{i,t} = V_i$
- What is $W_{i,t}$? $W_{i,t} = W_i$
- What is $\theta_t^{(i)}$? $\theta_t^{(i)} = (\mu_{i,t}, \beta_{i,t})$

Linear trend models

Combine these into a DLM with $m = 2$:

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, V_t) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0, W_t) \end{aligned}$$

- $\theta_t = (\mu_{1,t}, \mu_{2,t}, \beta_{1,t}, \beta_{2,t})$
- What is F_t?

$$F_t = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]$$

- What is G_t?

$$G_t = \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

- What is V_t?

$$V_t = V = \left[\begin{array}{cc} V_1 & ? \\ ? & V_2 \end{array}\right]$$

- What is W_t?

$$W_t = W = \left[\begin{array}{cc} W_\mu & 0 \\ 0 & W_\beta \end{array}\right]$$

- What are W_μ and W_β? If diagonal?

Kronecker products

Given two matrices A and B, the Kronecker product $A \otimes B$ is defined as

$$\begin{bmatrix} a_{1,1}B & \cdots & a_{1,n}B \\ \vdots & \vdots & \vdots \\ a_{m,1}B & \cdots & a_{m,n}B \end{bmatrix}$$

- If A is 2x2, what is $A \otimes I_2$?

SUTSE model

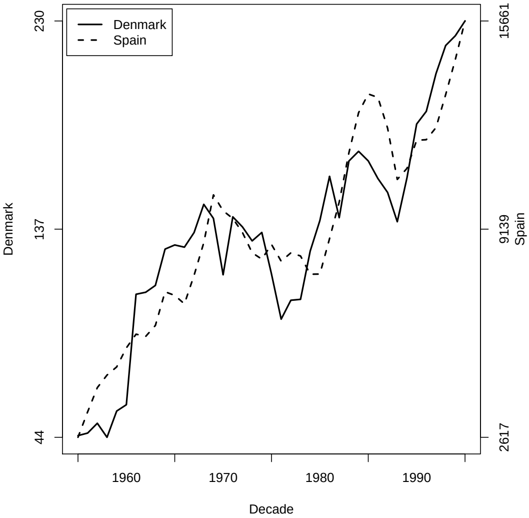
$$
\begin{aligned}
Y_t &= (F \otimes I_m)\theta_t + v_t & v_t &\sim N_m(0, V) \\
\theta_t &= (G \otimes I_m)\theta_{t-1} + w_t & w_t &\sim N_p(0, W)
\end{aligned}
$$

where

The covariance matrices V and W are typically somewhat sparse. Often

- off-diagonal elements of V are zero
- off-diagonal elements of W are zero
- some diagonal elements of W are zero

The data


Denmark
Spain
230
137
44
15661
9139
2617
Denmark
Spain
1960
1970
1980
1990
Decade

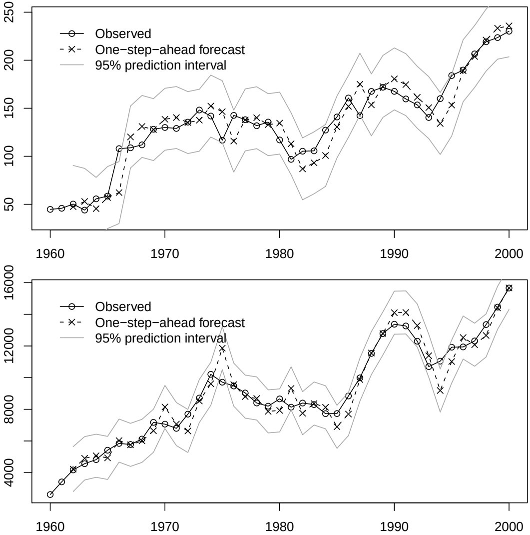
Model setup

```
> mod <- dlmModPoly(2)
> mod$FF <- mod$FF %x% diag(2)
> mod$GG <- mod$GG %x% diag(2)
>
> W1 <- matrix(0, 2, 2)
> W2 <- diag(c(49, 437266))
> W2[1, 2] <- W2[2, 1] <- 155
> mod$W <- bdiag(W1, W2)
>
> V <- diag(c(72, 14353))
> Var[1, 2] <- Var[2, 1] <- 1018
> mod$V <- V
>
> mod$m0 <- rep(0, 4)
> mod$C0 <- diag(4) * 1e7
> mod <- as.dlm(mod)
```

Filtering and one-step ahead prediction errors

```
> investFilt <- dlmFilter(invest, mod)
>
> sdev <- residuals(investFilt)$sd
> lwr <- investFilt$f + qnorm(0.025) * sdev
> upr <- investFilt$f - qnorm(0.025) * sdev
```

The data



Observed
One−step−ahead forecast
95% prediction interval
50
100
150
200
250
1960
1970
1980
1990
2000
Observed
One−step−ahead forecast
95% prediction interval
4000
8000
12000
16000
1960
1970
1980
1990
2000

Individual univariate dynamic regressions

Suppose for each unit i, we have a univariate dynamic regression model:

$$\begin{aligned} Y_{i,t} &= F_t^{(i)}\theta_t^{(i)} + v_{i,t}, & v_{i,t} &\sim N(0, V_i) \\ \theta_t^{(i)} &= G_{i,t}\theta_{t-1}^{(i)} + w_t^{(i)}, & w_t^{(i)} &\sim N_p(0, W_i) \end{aligned}$$

- What is $F_t^{(i)}$? $F_t^{(i)} = (x_{1,t}, \ldots, x_{p,t})$ if covariates are common to all series, as they will be in the example to follow
- What is $G_{i,t}$? $G_{i,t} = I_p$
- What is $V_{i,t}$? $V_{i,t} = V_i$
- What is $W_{i,t}$? $W_{i,t} = I_p$
- What is $\theta_t^{(i)}$? $\theta_t^{(i)} = (\beta_{1,t}^{(i)}, \ldots, \beta_{p,t}^{(i)})^\top$

Simple multivariate dynamic regression

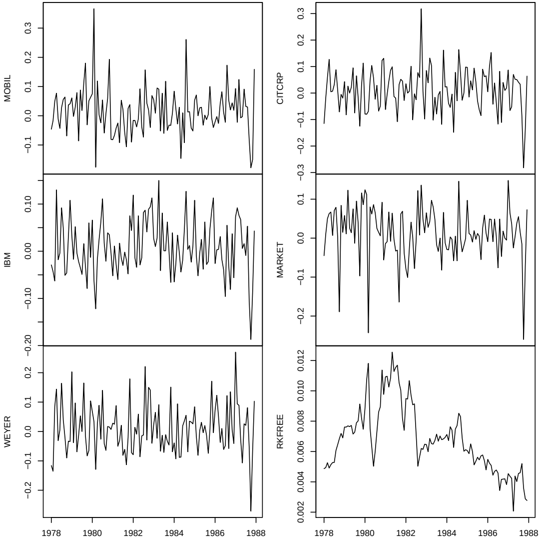
$$\begin{aligned} y_t &= (F_t \otimes I_m)\theta_t + v_t & v_t &\stackrel{ind}{\sim} N(0,V) \\ \theta_t &= (G \otimes I_m)\theta_{t-1} + w_t & w_t &\stackrel{ind}{\sim} N(0,W) \end{aligned}$$

where

$$y_t = \begin{bmatrix} y_{1,t} \\ \vdots \\ y_{m,t} \end{bmatrix}, \quad \theta_t = \begin{bmatrix} \alpha_{1,t} \\ \vdots \\ \alpha_{m,t} \\ \beta_{1,t} \\ \vdots \\ \beta_{m,t} \end{bmatrix}, \quad v_t = \begin{bmatrix} v_{1,t} \\ \vdots \\ v_{m,t} \end{bmatrix}, \quad w_t = \begin{bmatrix} w_{1,t} \\ \vdots \\ w_{m,t} \\ w_{m+1,t} \\ \vdots \\ w_{2m,t} \end{bmatrix}$$

with $F_t = (1, x_t)^\top, G = I_2$, and $W = \text{blockdiag}(W_\alpha, W_\beta)$.

Data


MOBIL
CITCRP
IBM
MARKET
WEYER
RKFREE
1978
1980
1982
1984
1986
1988

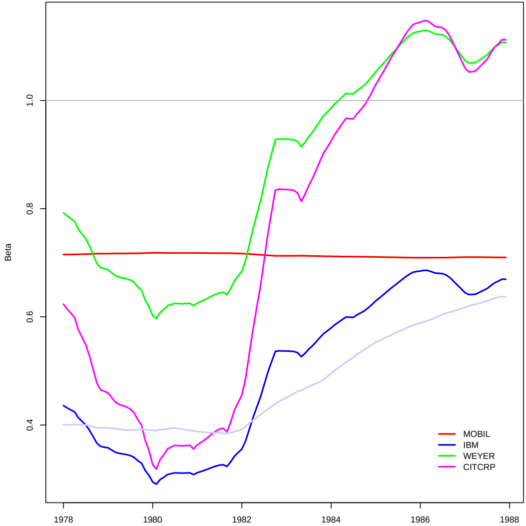
Model setup

```
> CAPM <- dlmModReg(market)
> CAPM$FF <- CAPM$FF %x% diag(m)
> CAPM$GG <- CAPM$GG %x% diag(m)
> CAPM$JFF <- CAPM$JFF %x% diag(m)
> CAPM$W <- CAPM$W %x% matrix(0, m, m)
> CAPM$W[-(1 : m), -(1 : m)] <-
+     c(8.153e-07,  -3.172e-05, -4.267e-05, -6.649e-05,
+       -3.172e-05, 0.001377,   0.001852,   0.002884,
+       -4.267e-05, 0.001852,   0.002498,   0.003884,
+       -6.649e-05, 0.002884,   0.003884,   0.006057)
> CAPM$V <- CAPM$V %x% matrix(0, m, m)
> CAPM$Var[] <- c(41.06,     0.01571, -0.9504, -2.328,
+                  0.01571, 24.23,     5.783,   3.376,
+                 -0.9504,   5.783,   39.2,     8.145,
+                 -2.328,    3.376,    8.145,  39.29)
> CAPM$m0 <- rep(0, 2 * m)
> CAPM$C0 <- diag(1e7, nr = 2 * m)
```

Smoothing inference

```
> CAPMsmooth <- dlmSmooth(y, CAPM)
```

Smoothed estimates



Longitudinal data

Suppose at each time point, you have observations from multiple units, e.g. people, countries, stocks, etc. The data can be represented as a matrix.

	Time			
Item	1	2	$\cdots$	T
1	$Y_{1,1}$	$Y_{1,2}$	$\cdots$	$Y_{1,T}$
2	$Y_{2,1}$	$Y_{2,2}$	$\cdots$	$Y_{2,T}$
$\vdots$		$\vdots$		
m	$Y_{m,1}$	$Y_{m,2}$	$\cdots$	$Y_{m,T}$

Individual DLMs

$$Y_{i,t} = F_{i,t}\theta_{i,t} + v_{i,t}, \quad v_{i,t} \sim N(0, \sigma^2_{i,t})$$

SUTSE: Necessary to estimate blocks of $m \times m$ matrices in the covariance matrix W. Instead:

$$\begin{array}{rll} \theta_{i,t} & = \lambda_t + \epsilon_{i,t}, & \epsilon_{i,t} \sim N_k(0, \Sigma_t) \\ \lambda_t & = G\lambda_{t-1} + w_t, & w_t \sim N_k(0, W_t) \end{array}$$

Hierarchical DLM:

$$\begin{array}{rll} Y_t & = F_{y,t}\theta_t + v_t, & v_t \sim N_m(0, V_{y,t}) \\ \theta_t & = F_{\theta,t}\lambda_t + \epsilon_t, & \epsilon_t \sim N_P(0, V_{\theta,t}) \\ \lambda_t & = G_t\lambda_{t-1} + w_t, & w_t \sim N_k(0, W_t) \end{array}$$

Hierarchical DLMs

Hierarchical DLM:

$$\begin{aligned} Y_t &= F_{y,t}\theta_t + v_t, & v_t &\sim N_m(0, V_{y,t}) \\ \theta_t &= F_{\theta,t}\lambda_t + \epsilon_t, & \epsilon_t &\sim N_P(0, V_{\theta,t}) \\ \lambda_t &= G_t\lambda_{t-1} + w_t, & w_t &\sim N_k(0, W_t) \end{aligned}$$

where

$$\begin{aligned} \theta_t &= (\theta_{1,t}^\top, \ldots, \theta_{m,t}^\top)^\top \\ F_{y,t} &= \text{blockdiag}(F_{i,t}) \\ F_{\theta,t} &= [I_p|\cdots|I_p]^\top \\ V_{y,t} &= \text{diag}(\sigma^2_{i,t}) \\ V_{\theta,t} &= \text{blockdiag}(\Sigma_t) \\ G_t &= G \end{aligned}$$

Integrating out θ_t

Let's write this as a standard DLM

$$\begin{aligned}
Y_t &= F_{y,t}F_{\theta,t}\lambda_t + v_t^*, & v_t^* &\sim N_m(0, F_{y,t}V_{\theta,t}F_{y,t}^\top + V_{y,t}) \\
\lambda_t &= G_t\lambda_{t-1} + w_t, & w_t &\sim N_k(0, W_t)
\end{aligned}$$

Longitudinal data with common covariates

Suppose at each time point, you have observations from multiple units, e.g. people, countries, stocks, etc., with a common covariate. The data can be represented as a matrix.

	Time				
Item	1	2	$\cdots$	T	$Covariate$
1	$Y_{1,1}$	$Y_{1,2}$	$\cdots$	$Y_{1,T}$	x_1
2	$Y_{2,1}$	$Y_{2,2}$	$\cdots$	$Y_{2,T}$	x_2
$\vdots$		$\vdots$			
m	$Y_{m,1}$	$Y_{m,2}$	$\cdots$	$Y_{m,T}$	x_m

Let's allow flexibility in how Y relates to x:

$$E(Y_t|x) = \sum_{j=1}^{k} \beta_{j,t} h_j(x)$$

where $h_j(x)$ may be, e.g.

$$\begin{array}{ll} \text{powers} & : h_j(x) = x^j \\ \text{trig functions} & : h_j(x) = a_j \sin(b_j x) \\ \text{cubic splines} & : h_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \end{array}$$

Then

$$Y_{i,t} = \sum_{j=1}^{k} \beta_{j,t} h_j(x_i) + \epsilon_{i,t}, \qquad \epsilon_{i,t} \sim N(0, \sigma^2).$$

In matrix notation,

$$Y_t = F\beta_t + \epsilon_t, \qquad \epsilon_t \sim N_m(0, V).$$

What are F and V?

Dynamic nonparametric regression

$$
\begin{aligned}
Y_t &= F\theta_t + \epsilon_t, & \epsilon_t &\sim N_m(0, V) \\
\theta_t &= G\theta_{t-1} + w_t, & w_t &\sim N_p(0, W_t)
\end{aligned}
$$

where

$$
\begin{aligned}
\theta_t &= \beta_t \\
F &= \left[\begin{array}{ccc} h_1(x_1) & \cdots & h_p(x_1) \\ \vdots & & \vdots \\ h_1(x_m) & \cdots & h_p(x_m) \end{array} \right] \\
V &= \sigma^2 I_m
\end{aligned}
$$

Dynamic factor model

$$\begin{aligned} Y_t &= A\mu_t + v_t, & v_t &\sim N_m(0, V) \\ \mu_t &= \mu_{t-1} + w_t, & w_t &\sim N_p(0, W) \end{aligned}$$

where A is a fixed $m \times p$ matrix of factor loadings with $p < m$.

Identifiability of A. Suppose H is a $p \times p$ invertible matrix. Then

$$\begin{aligned} Y_t &= \tilde{A}\tilde{\mu}_t + v_t, & v_t &\sim N_m(0, V) \\ \tilde{\mu}_t &= \tilde{\mu}_{t-1} + \tilde{w}_t, & \tilde{w}_t &\sim N_p(0, HWH^\top) \end{aligned}$$

with $\tilde{\mu}_t = H\mu_t$ and $\tilde{A} = AH^{-1}$.

- How many parameters in A? mp
- How many parameters in W? $\frac{1}{2}p(p+1)$
- Number of free parameters is $mp - \frac{1}{2}p(p-1)$.

Identifiable dynamic factor models

One method of enforcing identifiability is

- Let $W = I_p$,
- Let $A = \left[\begin{array}{c} L \\ B \end{array} \right]$ where L is a lower triangular matrix and B can be anything.
- Total parameters are $mp - \frac{1}{2}p(p-1)$.

Another method is

- Let W be diagonal
- Let $A_{i,i} = 1$ and $A_{i,j} = 0$ for $j > i$.

Unknown parameters in polynomial trend models

What is known?

- $F_t = (1, 0, \ldots, 0)$
- $$G_t = G = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & & \cdots & 0 & 1 & 1 \\ 0 & & \cdots & & 0 & 1 \end{array}\right]$$

What are the unknown parameters?

- θ_t
- $V_t \stackrel{?}{=} V$
- $W_t \stackrel{?}{=} W$

Unknown parameters in seasonal models

What is known?

- F_t
  - Seasonal factor: $F_t = (1, 0, \ldots, 0)$
  - Fourier form: $F_t = (1, 0, 1, 0, \ldots, 1, 0)$
- $G_t = G$
  - Seasonal factor: rotation matrix
  - Fourier form: block diagonal with blocks H_j

$$H_j = \begin{bmatrix} \cos\omega_j & \sin\omega_j \\ -\sin\omega_j & \cos\omega_j \end{bmatrix}$$

What are the unknown parameters?

- θ_t
- $V_t \stackrel{?}{=} V$
- $W_t \stackrel{?}{=} W \stackrel{?}{=} 0$

Unknown parameters in dynamic regression models

What is known?

- $F_t = x_t$
- $G_t \stackrel{?}{=} G \stackrel{?}{=} I$

What are the unknown parameters?

- θ_t
- $V_t \stackrel{?}{=} V \stackrel{?}{=} \sigma^2 I$ or $\sigma^2 D$
- $W_t \stackrel{?}{=} W \stackrel{?}{=} D$

The bottom line is...

- In all of these univariate models,
  - the unknowns are θ_t, W_t, and V_t,
  - θ_t has always been unknown
  - and often, $W_t = W$ and $V_t = V$.
- In our multivariate models,
  - commonly $W_t = W$ and $V_t = V$, but now they are (block-diagonal) matrices.

For a parameter vector ψ and data vector y, the likelihood function

$$L(\psi) \propto p(y|\psi).$$

The maximum likelihood estimate is

$$\hat{\psi} = \text{argmax}_\psi L(\psi).$$

Which is equivalent to

$$\hat{\psi} = \text{argmax}_\psi \ell(\psi)$$

where $\ell(\psi) = \log L(\psi)$.

Likelihood function for DLMs

If $\psi = (W, V)$, what is $L(\psi)$ for a general DLM?

What do we know?

$$\begin{array}{rcl}
p(y_t|\theta_t, V) &=& N(y_t; F_t\theta_t, V) \\
p(\theta_t|\theta_{t-1}, W) &=& N(\theta_t; G_t\theta_{t-1}, W) \\
p(\theta_0) &=& N(m_0, C_0)
\end{array}$$

$$\begin{array}{rcl}
p(\theta_t|y_{1:t-1}, \psi) &=& N(a_t, R_t) \\
p(y_t|y_{1:t-1}, \psi) &=& N(f_t, Q_t)
\end{array}$$

$$p(y|\psi) = \prod_{t=1}^{n} p(y_t|y_{1:t-1}, \psi)$$

Finding MLEs for DLMs

If y_t is multivariate, the likelihood function is

$$L(\psi) \propto \prod_{t=1}^{n} \frac{1}{(2\pi)^{k/2}|Q_t|^{1/2}} \exp\left(-\frac{1}{2}(y_t - f_t)^\top Q_t^{-1}(y_t - f_t)\right).$$

Log-likelihood function

$$\ell(\psi) = C + -\frac{1}{2}\sum_{t=1}^{n} \log|Q_t| - \frac{1}{2}\sum_{t=1}^{n}(y_t - f_t)^\top Q_t^{-1}(y_t - f_t).$$

The MLE is then

$$\hat{\psi} = \text{argmax}_\psi \ell(\psi)$$

The R function `dlmMLE` does all of this for you.

Bayesian inference

What do we have to specify to perform Bayesian inference, i.e. parameter estimation, for data y?

- A statistical model $p(y|\psi)$
- A prior $p(\psi)$

What is the objective of Bayesian inference?

- The posterior $p(\psi|y) \propto p(y|\psi)p(\psi)$.

Conjugacy

Conjugate Bayesian inference is one where if

$$\psi \sim f(\alpha) \implies \psi|y \sim f(\alpha').$$

Remember the examples

- $y \sim N(\mu, I), \mu \sim N(\cdot,\cdot) \implies \mu|y \sim N(\cdot,\cdot)$
- $y \sim N(0, \phi^{-1}I), \phi \sim Ga(\cdot,\cdot) \implies \phi|y \sim Ga(\cdot,\cdot)$
- $y \sim N(\mu, \phi^{-1}I), \mu, \phi \sim NG(\cdot) \implies \mu, \phi|y \sim NG(\cdot)$
- $y \sim N(X\beta, \phi^{-1}I), \beta, \phi \sim NG(\cdot) \implies \beta, \phi|y \sim NG(\cdot)$
- $y \sim Bin(n, p), p \sim Be(\cdot,\cdot) \implies p|y \sim Be(\cdot,\cdot)$

What are the unknowns in DLMs?

So for $\psi = (F_{1:n}, G_{1:n}, W_{1:n}, V_{1:n})$, we are looking for

$$\psi \sim f(\alpha) \implies \psi|y \sim f(\alpha').$$

This only happens in simple examples. Today, we will discuss

- $V_t = \phi^{-1}\tilde{V}_t, W_t = \phi^{-1}\tilde{W}_t, C_0 = \phi^{-1}\tilde{C}_0$
- W_t specified by a discount factor
- Evolving $\phi = 1/\sigma^2$

common ϕ^{-1}

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, \phi^{-1}\tilde{V}_t) \\
\theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0, \phi^{-1}\tilde{W}_t) \\
p(\theta_0) &= N(m_0, \phi^{-1}\tilde{C}_0) \\
\phi &\sim Ga(\alpha_0, \beta_0)
\end{aligned}
$$

Everything is known except

- θ_t for all t
- ϕ

Starting with

$$\theta_{t-1}, \phi | y_{1:t-1} \sim NG(m_{t-1}, \tilde{C}_{t-1}, \alpha_{t-1}, \beta_{t-1})$$

One step ahead prior

$$\theta_t, \phi | y_{1:t-1} \sim NG(a_t, \tilde{R}_t, \alpha_{t-1}, \beta_{t-1})$$

where $a_t = G_t m_{t-1}$ and $\tilde{R}_t = G_t \tilde{C}_{t-1} G_t^\top + \tilde{W}_t$.

One step ahead predictive density

$$Y_t | y_{1:t-1} \sim t_{2\alpha_{t-1}}(f_t, \tilde{Q}_t \beta_{t-1}/\alpha_{t-1})$$

with $f_t = F_t a_t$ and $\tilde{Q}_t = F_t \tilde{R}_t F_t^\top + \tilde{V}_t$.

Filtering density

$$\theta_t, \phi | y_{1:t} \sim NG(m_t, \tilde{C}_t, \alpha_t, \beta_t)$$

with

$$\begin{aligned}
m_t &= a_t + \tilde{R}_t F_t \tilde{Q}_t^{-1}(y_t - f_t) \\
\tilde{C}_t &= \tilde{R}_t - \tilde{R}_t F_t^\top \tilde{Q}_t^{-1} \tilde{R}_t^\top \\
\alpha_t &= \alpha_{t-1} + \frac{m}{2} \\
\beta_t &= \beta_{t-1} + \frac{1}{2}(y_t - f_t)^\top \tilde{Q}_t^{-1}(y_t - f_t)
\end{aligned}$$

Discount factor

Let's specify how adaptive we want our model to be.

- Do this by specifying W_t relative to V_t and C_t using a discount factor $\delta \in (0, 1]$.
- $\delta = 1$ means no loss of information, i.e. $W_t = 0$
- $\delta = 0$ means no information retained
- Often $\delta > 0.9$

To implement, set

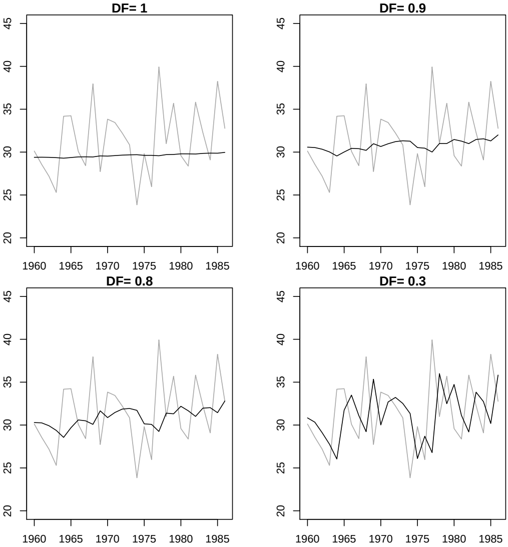
$$W_t = \frac{1-\delta}{\delta} G_t^\top C_{t-1} G_t$$

or

$$\tilde{W}_t = \frac{1-\delta}{\delta} G_t^\top \tilde{C}_{t-1} G_t$$

if using a common σ^2.

Discount factor effect



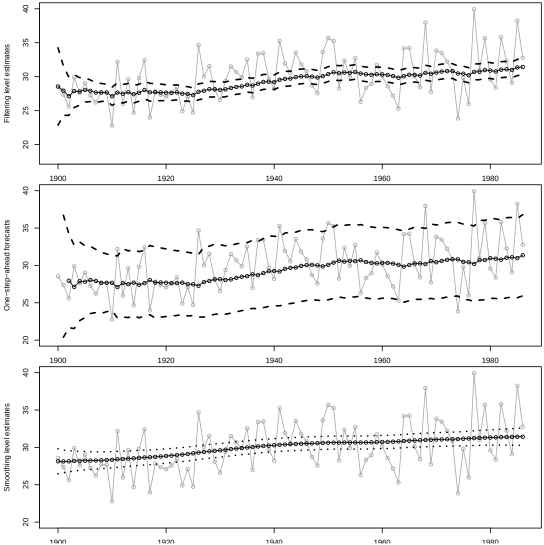
Choosing the discount factor

Specify δ based on one-step ahead prediction errors.

DF	MAPE	MAD	MSE	sigma2
1.0	0.10	3.02	21.54	12.00
0.9	0.09	2.86	19.92	9.64
0.8	0.10	2.87	20.29	8.94
0.3	0.11	3.42	25.12	5.07

Last column is posterior expectation for σ^2, i.e. $E[\sigma^2|y_{1:187}]$.

Inference for $\delta = 0.95$ on Lake Superior Data



Evolving ϕ_t

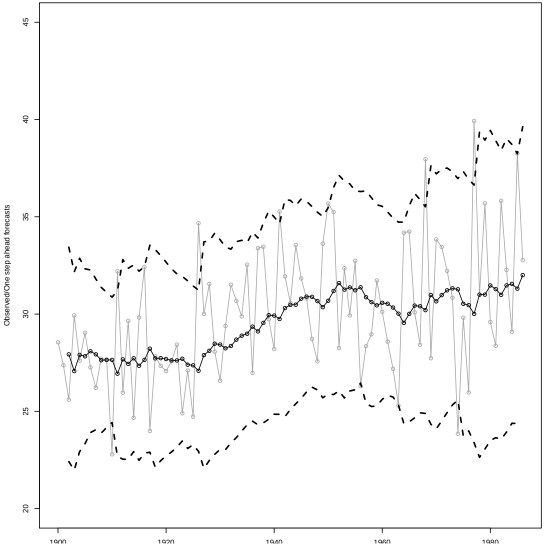
Choose $\delta^* \in (0,1)$

- $\phi_{t-1}|y_{1:t-1} \sim Ga(\alpha_{t-1}, \beta_{t-1})$
- $\phi_t|y_{1:t-1} \sim Ga(\delta^*\alpha_{t-1}, \delta^*\beta_{t-1})$

What is

- $E[\phi_t|y_{1:t-1}] = E[\phi_{t-1}|y_{1:t-1}]$
- $Var[\phi_t|y_{1:t-1}] = \frac{1}{\delta^*} Var[\phi_{t-1}|y_{1:t-1}]$

Evolving ϕ_t for Lake Superior data



The situations for conjugate Bayesian analysis are small, therefore we need more advanced techniques.

Gibbs sampling algorithm

Start with an initial guess for all parameters and call it $\psi^{(0)}$. Set $j = 1$.

1. Sample $\psi_1^{(j)} \sim p\left(\psi_1|\psi_2^{(j-1)}, \ldots, \psi_K^{(j-1)}, y\right)$
2. Sample $\psi_2^{(j)} \sim p\left(\psi_2|\psi_1^{(j)}, \psi_3^{(j-1)}, \ldots, \psi_K^{(j-1)}, y\right)$
3. $\vdots$
4. Sample $\psi_k^{(j)} \sim p\left(\psi_k|\psi_1^{(j)}, \ldots, \psi_{k-1}^{(j)}, \psi_{k+1}^{(j-1)}, \ldots, \psi_K^{(j-1)}, y\right)$
5. $\vdots$
6. Sample $\psi_{K-1}^{(j)} \sim p\left(\psi_{K-1}|\psi_1^{(j)}, \ldots, \psi_{K-2}^{(j)}, \psi_K^{(j-1)}, y\right)$
7. Sample $\psi_K^{(j)} \sim p\left(\psi_K|\psi_1^{(j)}, \ldots, \psi_{K-1}^{(j)}, y\right)$
8. If $j < J$, $j = j + 1$ and return to step 1.

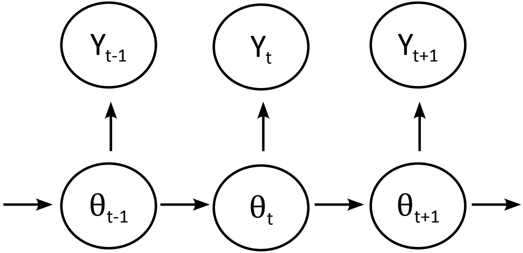
What full conditionals are required?

Suppose our goal is to draw from $p(\theta_{0:T}|y_{1:T})$ using univariate Gibbs sampling. We will implicitly assume conditioning on any other unknown parameters.

- What are the required full condition distributions?
  - $p(\theta_0|\theta_{1:T}, y_{1:T})$
  - $p(\theta_t|\theta_{-t}, y_{1:T})$ where θ_{-t} is $\theta_{0:T}$ with the t^{th} element removed
  - $p(\theta_T|\theta_{0:T-1}, y_{1:T})$

DLMs

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, V_t) \\
\theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0, W_t) \\
p(\theta_0) &= N(m_0, C_0)
\end{aligned}
$$



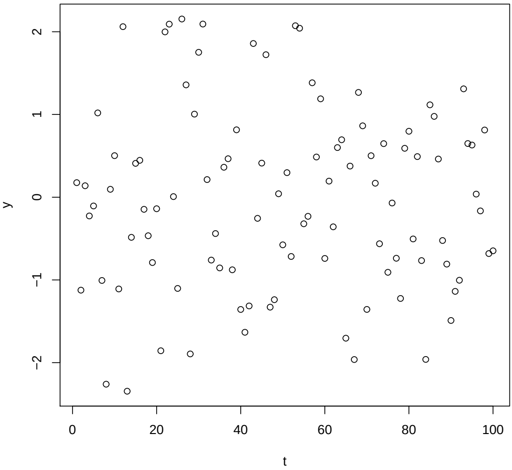
What are the full conditionals?

$$
\begin{aligned}
p(\theta_0|\ldots) &= p(\theta_0|\theta_1) \\
&\propto N(\theta_1; G_1\theta_0, W_1) N(\theta_0; m_0, C_0) \\
&\propto N(\theta_0; k_0, K_0) \\
K_0 &= (C_0^{-1} + G_1^\top W_1^{-1} G_1)^{-1} \\
k_0 &= K_0(C_0^{-1} m_0 + G_1^\top W_1^{-1} \theta_1)
\end{aligned}
$$

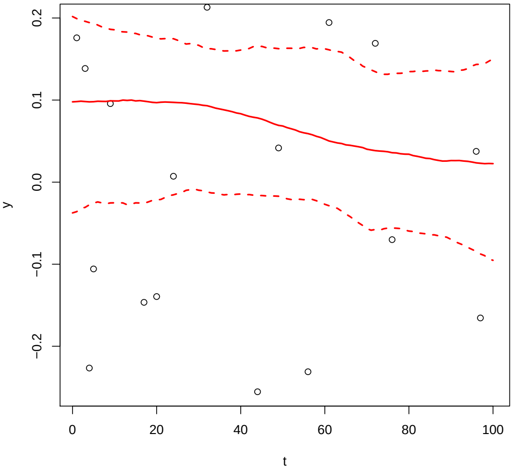
$$
\begin{aligned}
p(\theta_T|\ldots) &= p(\theta_T|\theta_{T-1}, y_T) \\
&\propto N(y_T; F_T\theta_T, V_T) N(\theta_T; G_T\theta_{T-1}, W_T) \\
&\propto N(\theta_T; k_T, K_T) \\
K_T &= (W_T^{-1} + F_T^\top V_T^{-1} F_T)^{-1} \\
k_T &= K_T(W_T^{-1} G_T\theta_{T-1} + F_T^\top V_T^{-1} y_T)
\end{aligned}
$$

$$
\begin{aligned}
p(\theta_t|\ldots) &= p(\theta_t|\theta_{t-1}, \theta_{t+1}, y_t) \\
&\propto N(y_t; F_t\theta_t, V_t) N(\theta_{t+1}; G_{t+1}\theta_t, W_t) N(\theta_t; G_t\theta_{t-1}, W_{t+1}) \\
&\propto N(\theta_t; k_t, K_t) \\
K_t &= (W_t^{-1} + F_t^\top V_t^{-1} F_t + G_{t+1}^\top W_{t+1}^{-1} G_{t+1})^{-1} \\
k_t &= K_t(W_t^{-1} G_t\theta_{t-1} + F_t^\top V_t^{-1} y_t + G_{t+1}^\top W_{t+1}^{-1} \theta_{t+1})
\end{aligned}
$$

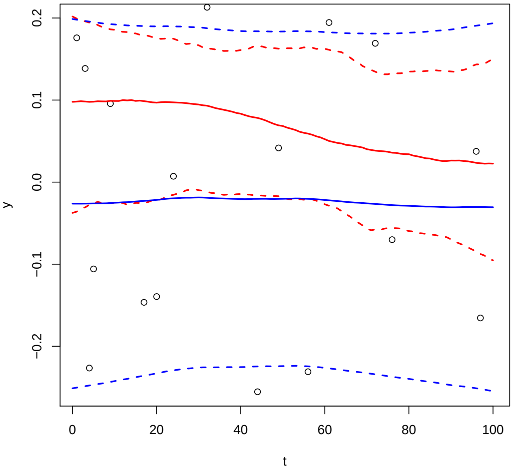
Consider the local level model with $V = 1$ and $W = 0.01^2$.



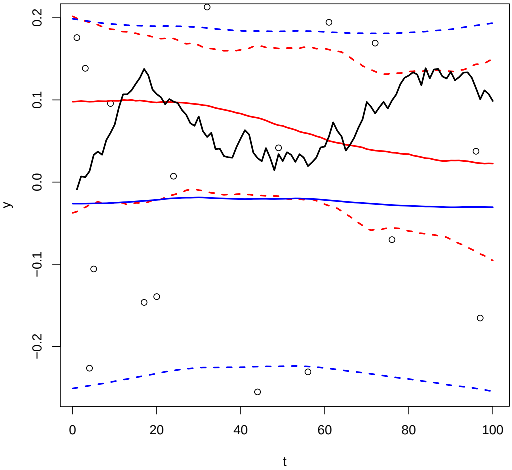
Univariate Gibbs sampling for states



Exact quantiles for states



True underlying state



Filtering

Goal: $p(\theta_t|y_{1:t})$ where $y_{1:t} = (y_1, y_2, \ldots, y_t)$ (filtered distribution)

Recursive procedure:

- Assume $p(\theta_{t-1}|y_{1:t-1})$
- Prior for θ_t

$$
\begin{aligned}
p(\theta_t|y_{1:t-1}) &= \int p(\theta_t, \theta_{t-1}|y_{1:t-1}) d\theta_{t-1} \\
&= \int p(\theta_t|\theta_{t-1}, y_{1:t-1}) p(\theta_{t-1}|y_{1:t-1}) d\theta_{t-1} \\
&= \int p(\theta_t|\theta_{t-1}) p(\theta_{t-1}|y_{1:t-1}) d\theta_{t-1}
\end{aligned}
$$

- One-step ahead predictive distribution for y_t

$$
\begin{aligned}
p(y_t|y_{1:t-1}) &= \int p(y_t, \theta_t|y_{1:t-1}) d\theta_t \\
&= \int p(y_t|\theta_t, y_{1:t-1}) p(\theta_t|y_{1:t-1}) d\theta_t \\
&= \int p(y_t|\theta_t) p(\theta_t|y_{1:t-1}) d\theta_t
\end{aligned}
$$

- Filtered distribution for θ_t

$$
p(\theta_t|y_{1:t}) = \frac{p(y_t|\theta_t, y_{1:t-1}) p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} = \frac{p(y_t|\theta_t) p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}
$$

Smoothing

Goal: $p(\theta_t|y_{1:T})$ for $t < T$

- Backward transition probability $p(\theta_t|\theta_{t+1}, y_{1:T})$

$$
\begin{aligned}
p(\theta_t|\theta_{t+1}, y_{1:T}) &= p(\theta_t|\theta_{t+1}, y_{1:t}) \\
&= \frac{p(\theta_{t+1}|\theta_t, y_{1:t})p(\theta_t|y_{1:t})}{p(\theta_{t+1}|y_{1:t})} \\
&= \frac{p(\theta_{t+1}|\theta_t)p(\theta_t|y_{1:t})}{p(\theta_{t+1}|y_{1:t})}
\end{aligned}
$$

- Recursive smoothing distributions $p(\theta_t|y_{1:T})$ assuming we know $p(\theta_{t+1}|y_{1:T})$

$$
\begin{aligned}
p(\theta_t|y_{1:T}) &= \int p(\theta_t, \theta_{t+1}|y_{1:T})d\theta_{t+1} \\
&= \int p(\theta_{t+1}|y_{1:T})p(\theta_t|\theta_{t+1}, y_{1:T})d\theta_{t+1} \\
&= \int p(\theta_{t+1}|y_{1:T})\frac{p(\theta_{t+1}|\theta_t)p(\theta_t|y_{1:t})}{p(\theta_{t+1}|y_{1:t})}d\theta_{t+1} \\
&= p(\theta_t|y_{1:t})\int \frac{p(\theta_{t+1}|\theta_t)}{p(\theta_{t+1}|y_{1:t})}p(\theta_{t+1}|y_{1:T})d\theta_{t+1}
\end{aligned}
$$

Start from $p(\theta_T|y_{1:T})$.

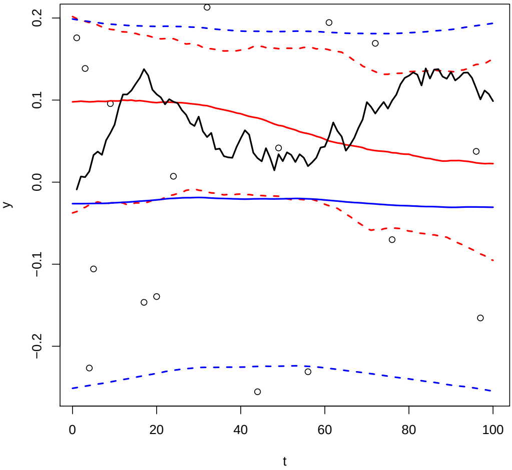
Kalman smoother

If $p(\theta_{t+1}|y_{1:T}) = N(s_{t+1}, S_{t+1})$, then

$$\begin{aligned}
p(\theta_t|\theta_{t+1}, y_{1:T}) &= p(\theta_t|\theta_{t+1}, y_{1:t}) \\
&\propto p(\theta_{t+1}|\theta_t, y_{1:t})p(\theta_t|y_{1:t}) \\
&= N(\theta_{t+1}; G_{t+1}\theta_t, W_{t+1})N(\theta_t; m_t, C_t) \\
&\propto N(\theta_t; h_t, H_t) \\
H_t &= (C_t^{-1} + G_{t+1}^\top W_{t+1}^{-1} G_{t+1})^{-1} \\
h_t &= H_t(C_t^{-1} m_t + G_{t+1}^\top W_{t+1}^{-1} \theta_{t+1})
\end{aligned}$$

$$\begin{aligned}
p(\theta_t|y_{1:T}) &= \int p(\theta_t|\theta_{t+1}, y_{1:T})p(\theta_{t+1}|y_{1:T})d\theta_{t+1} \\
&= N(\theta_t; s_t, S_t) \\
S_t &= C_t - C_t G_{t+1}^\top R_{t+1}^{-1}(R_{t+1} - S_{t+1})R_{t+1}^{-1} G_{t+1} C_t \\
s_t &= m_t + C_t G_{t+1}^\top R_{t+1}^{-1}(s_{t+1} - a_{t+1})
\end{aligned}$$

True underlying state



Let ψ represent any unknown, non-dynamic model parameters such that the data follows a DLM conditional on ψ.

$$\begin{array}{rll} Y_t & = F_t(\psi)\theta_t + v_t & v_t \sim N_m(0, V_t(\psi)) \\ \theta_t & = G_t(\psi)\theta_{t-1} + w_t & w_t \sim N_p(0, W_t(\psi)) \\ p(\theta_0) & = N(m_0(\psi), C_0(\psi)) & \end{array}$$

For example, $\psi = (V, W)$ where $V_t(\psi) = V$ and $W_t(\psi) = W$ while $F_t(\psi) = F$ and $G_t(\psi) = G$ are known, as in polynomial trend, seasonal factor, and dynamic regression models.

- The Bayesian inferential objective is then $p(\theta_{0:T}, \psi | y_{1:T})$.
- While $p(\theta_{0:T}|y_{1:T}, \psi)$ is known analytically, generally $p(\theta_{0:T}, \psi|y_{1:T})$ is not.
- So resort to numerical methods, most often MCMC

MCMC Schemes

Scheme I - all univariate samples

- For $t \in \{0, 1, \ldots, T\}$ sample $p(\theta_t | \ldots)$.
- For $j \in \{1, \ldots, J\}$ sample $p(\psi_j | \ldots)$ for J parameters.

Scheme II - block sampling of states

- Sample $p(\theta_{0:T} | \ldots)$.
- For $j \in \{1, \ldots, J\}$ sample $p(\psi_j | \ldots)$ for J parameters.

MCMC Schemes (cont.)

Scheme III - block sampling of parameters

- Sample $p(\theta_{0:T}|\ldots)$.
- Sample $p(\psi|\ldots)$.

e.g. polynomial trend, seasonal factor, and dynamic regression models

Scheme IV - hybrid

- Sample $p(\psi_{J'}|\psi_{J\setminus J'}, y_{1:T})$ for some subset J' of parameters.
- Sample $p(\theta_{0:T}|\ldots)$.
- Sample $p(\psi_{J\setminus J'}|\ldots)$.

MCMC Schemes

Generally better to jointly sampling unknowns, a.k.a. block sampling.

- Scheme I has all univariate draws
- Scheme II samples latent state jointly
- Scheme III samples latent state jointly and parameters jointly
- Scheme IV samples some parameters $\psi_{J'}$ and all latent states jointly and then samples remaining parameters jointly

Bottom line: if parameters are highly correlated in the posterior, it is better to sample those parameters jointly.

Forward filtering backward sampling (FFBS)

Recall

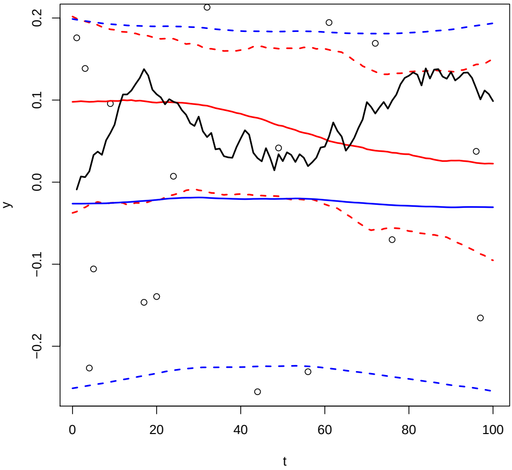
- $p(\theta_T|y_{1:T}) = N(m_T, C_T)$ is available from filtering
- $p(\theta_t|\theta_{t+1}, y_{1:T}) = N(h_t, H_T)$ is available from smoothing

$$\begin{aligned} H_t &= (C_t^{-1} + G_{t+1}^\top W_{t+1}^{-1} G_{t+1})^{-1} \\ h_t &= H_t(C_t^{-1} m_t + G_{t+1}^\top W_{t+1}^{-1} \theta_{t+1}) \end{aligned}$$

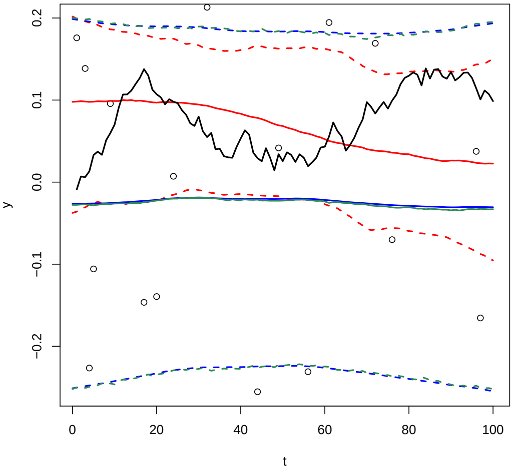
The algorithm is then

- Forward filter to obtain $p(\theta_t|y_{1:t}) = N(m_t, C_t)$ for all t.
- Sample $\theta_T \sim H(m_T, C_T)$.
- For $t \in \{T-1, T-2, \ldots, 1, 0\}$,
  - Calculate h_t and H_t based on θ_{t+1}.
  - Draw $\theta_t \sim N(h_t, H_T)$.

Local level model



Local level model



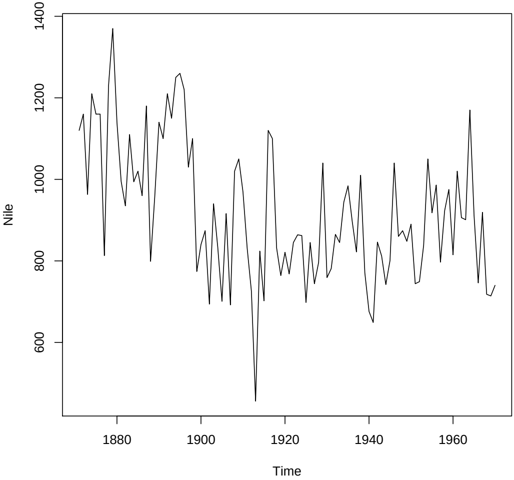
Local level model - unknown variances

$$\begin{aligned} Y_t &= \theta_t + v_t & v_t &\sim N_m(0, V) \\ \theta_t &= \theta_{t-1} + w_t & w_t &\sim N_p(0, W) \\ p(\theta_0) &= N(m_0, C_0) \end{aligned}$$

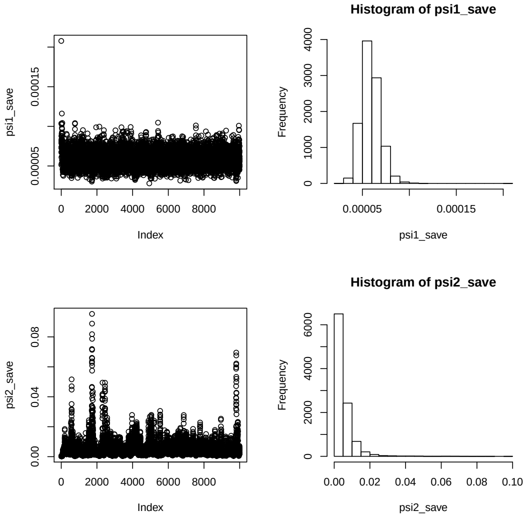
MCMC Scheme:

- Sample $p(\theta_{0:T}|\ldots)$ using FFBS
- Sample $p(V, W|\ldots)$

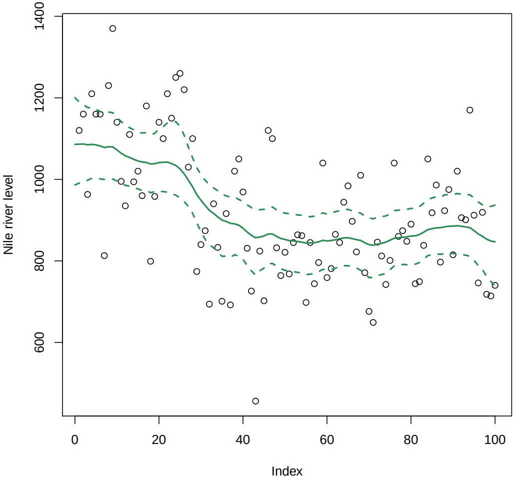
Nile river level



Nile river level



Nile river level



MCMC in DLMs

Recall the inferential objective of the Bayesian approach in DLMs:

$$p(\theta_{0:n}, \psi | y_{1:n})$$

Since $p(\theta_{0:n}, \psi | y_{1:n})$ is not typically available analytically, we commonly use Markov chain Monte Carlo. These approaches sample from full conditional distributions, e.g.

- $p(\theta_t | \theta_{-t}, \psi, y_{1:n})$ for $t = 0, 1, 2, \ldots, n$.
- $p(\psi_j | \theta_{0:n}, \psi_{-j}, y_{1:n})$ for $j = 1, 2, \ldots, J$.

These draws could be Gibbs or Metropolis-Hastings.

MCMC Univariate Sampling Activity

Fill in ? with i or $i-1$.

$$\theta_0^{(?)} \sim p(\theta_0|\theta_1^{(?)},\ldots,\theta_n^{(?)},\psi^{(?)},y_{1:n})$$

For $t \in 1,\ldots,n-1$, sample from

$$\theta_t^{(?)} \sim p(\theta_t|\theta_0^{(?)},\ldots,\theta_{t-1}^{(?)},\theta_{t+1}^{(?)},\ldots,\theta_n^{(?)},\psi^{(?)},y_{1:n})$$

$$\theta_n^{(?)} \sim p(\theta_n|\theta_0^{(?)},\ldots,\theta_{n-1}^{(?)},\psi^{(?)},y_{1:n})$$

$$\psi_1^{(?)} \sim p(\psi_1|\theta^{(?)},\psi_2^{(?)},\ldots,\psi_J^{(?)},y_{1:n})$$

For $j \in 2,\ldots,J-1$, sample from

$$\psi_j^{(?)} \sim p(\psi_j|\theta^{(?)},\psi_1^{(?)},\ldots,\psi_{j-1}^{(?)},\psi_{j+1}^{(?)},\ldots,\psi_J^{(?)},y_{1:n})$$

$$\psi_J^{(?)} \sim p(\psi_J|\theta^{(?)},\psi_1^{(?)},\ldots,\psi_{J-1}^{(?)},y_{1:n})$$

Convergence to stationary distribution

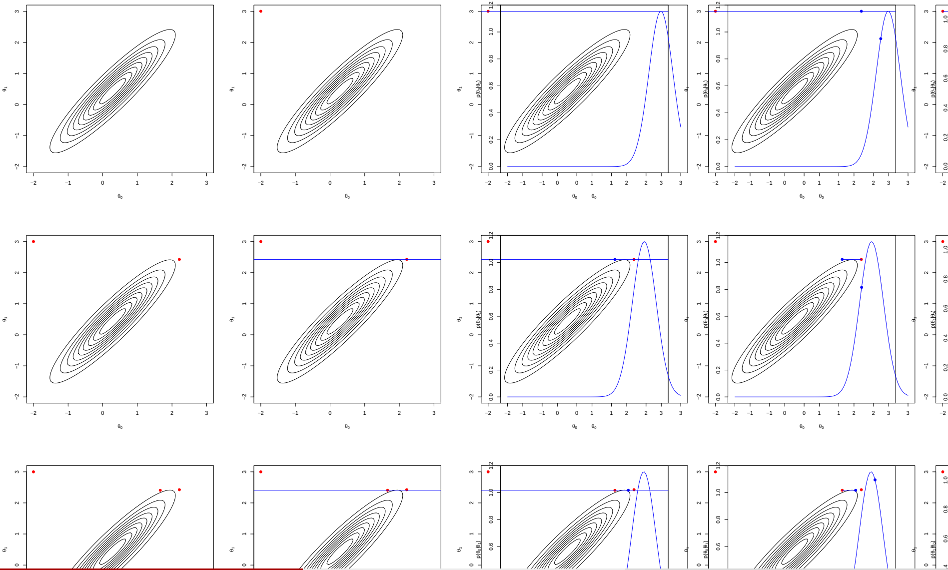
The samples $(\theta^{(i)}, \psi^{(i)})$ converge to samples from $p(\theta_{0:n}, \psi|y_{1:n})$, regardless of what $(\theta^{(0)}, \psi^{(0)})$ was.

Let's look at an example: local level model.

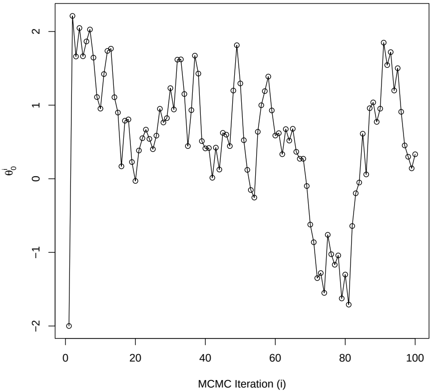
$$\begin{aligned}
Y_t &= \theta_t + v_t & v_t &\sim N(0,2) \\
\theta_t &= \theta_{t-1} + w_t & w_t &\sim N(0,0.5) \\
p(\theta_0) &= N(0,1)
\end{aligned}$$

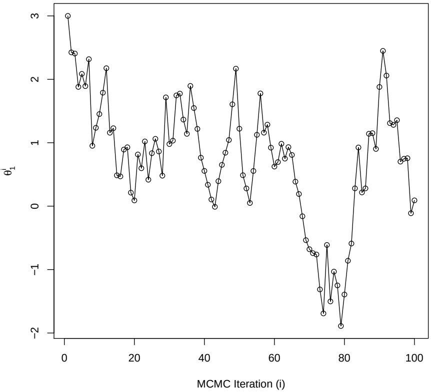
with $y_1 = 1$. The objective is $p(\theta_0, \theta_1|y_1)$.

Local level convergence example

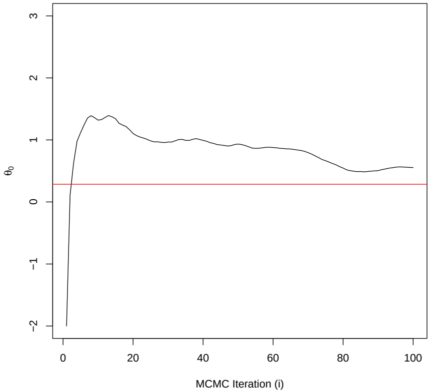


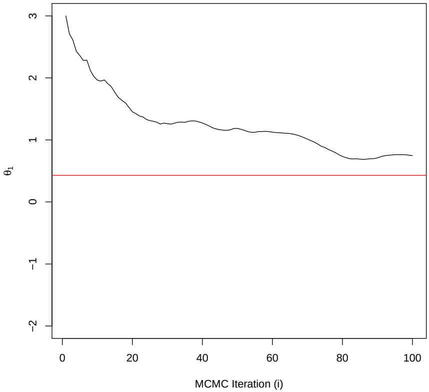
Traceplots


θ_0^i
MCMC Iteration (i)
0
20
40
60
80
100
−2
−1
0
1
2

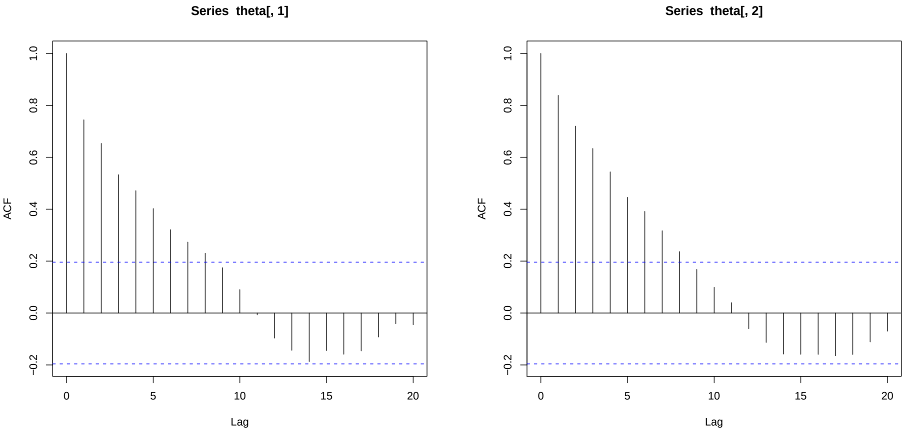

θ_1^i
MCMC Iteration (i)
0
20
40
60
80
100
−2
−1
0
1
2
3

Running average





Auto-correlation plots



MCMC Convergence diagnostics

- Graphical techniques
  - Traceplots
  - Ergodic mean
- Non-graphical techniques
  - Geweke diagnostic - single chain
  - Gelman/Rubin diagnostic - multiple chains

Lack of convergence

We can never know if our chain has converged.

All convergence diagnostics detect a lack of convergence.

So instead of saying ‘the chain has converged’ you should be saying ‘the chain shows no lack of convergence’.

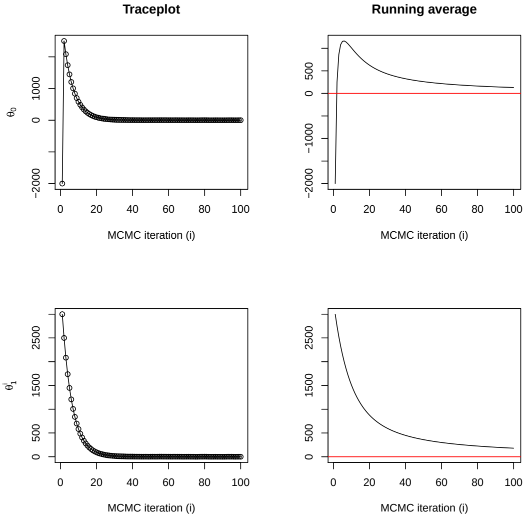
Burn-in

Definition

Burn-in is the number of MCMC iterations before the chain shows no lack of convergence.

Burn-in is thrown-out to eliminate the bias associated with the starting point.

Burn-in example



Burn-in

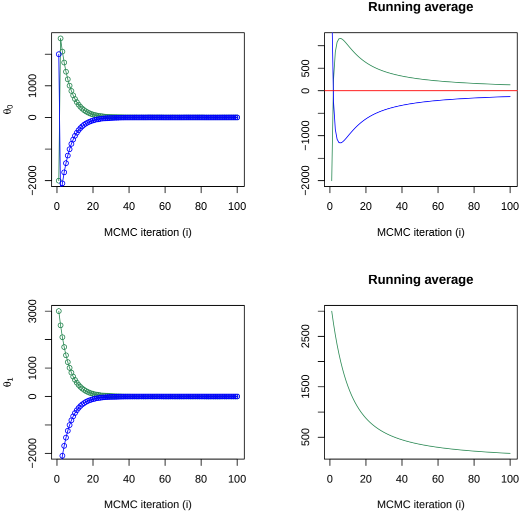
Definition

Burn-in is the period of time before the chain has converged.

Burn-in is thrown-out to eliminate the bias associated with the starting point.

If the starting point is crucial, why not start multiple chains in different locations? With the local level model, start chain 1 at (-2000, 3000) and start chain 2 at (3000, -2000).

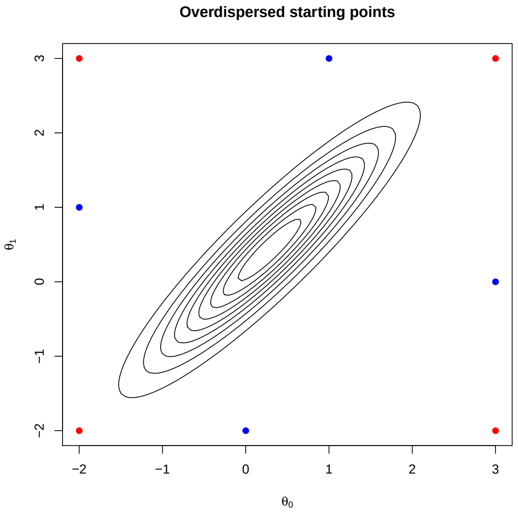
Multiple chains



Gelman-Rubin diagnostic

- Start multiple chains at locations that are overdispersed relative to the posterior.
- ANOVA comparison
  - Within-chain versus between-chain variances
  - Represented as a scale reduction factor such that values around 1 indicate no lack of convergence.

Local level model example

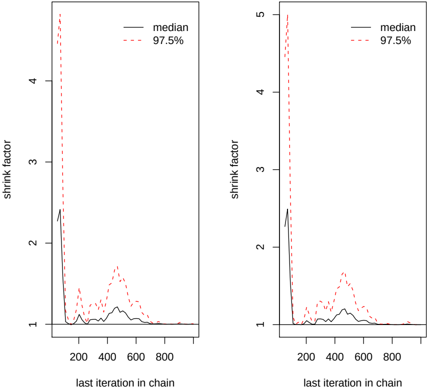


Local level model example - 100 iterations

In package coda, use function `gelman.diag`.

```
Potential scale reduction factors:

     Point est. 97.5% quantile
[1,]       1.12           1.45
[2,]       1.13           1.48

Multivariate psrf

1.09
```

Values substantially above 1 indicate lack of convergence.

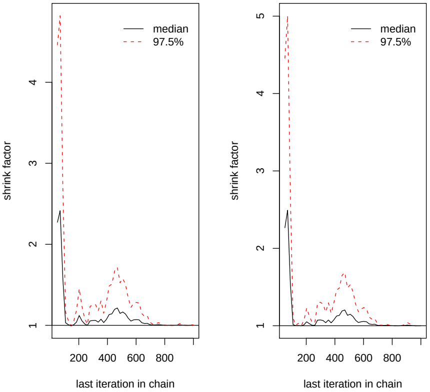


Local level model example - 1000 iterations

In package coda, use function `gelman.diag`.

```
Potential scale reduction factors:

     Point est. 97.5% quantile
[1,]          1           1.00
[2,]          1           1.00

Multivariate psrf

1.00
```

Values substantially above 1 indicate lack of convergence.



Iterations for inference

Now that no lack of convergence is apparent, how long should I run my chain?

- The longer you run the chain, the lower your Monte Carlo error.
- Monte Carlo error reduces by the $\sqrt{N}$ where N is the number of MCMC iterations.
- So, if you want a 10-fold decrease in Monte Carlo error, you need to run 10^2 times your current number of iterations.

Simple Monte Carlo example

Consider the model $y_i \stackrel{ind}{\sim} N(\mu, 1)$ and our goal is to estimate $E[y_i] = \mu$. The Monte Carlo approximation is

$$\mu \approx \mu_{MC} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

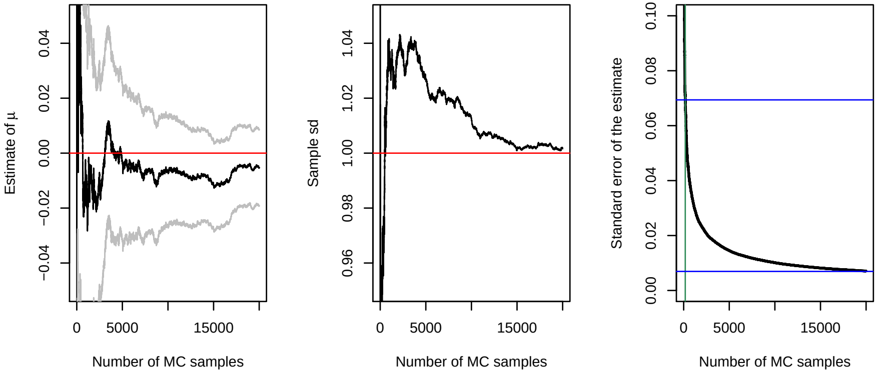
with the variance of this approximation given by

$$\text{se}(\mu_{MC}) \approx \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(y_i - \mu_{MC})^2} = \frac{1}{\sqrt{n}} sd_y$$

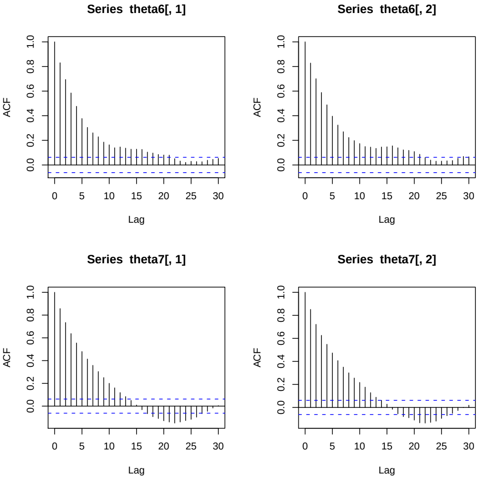
where sd_y is the standard deviation of the sample $y = (y_1, y_2, \dots, y_n)$. Since this standard deviation converges to 1, by our model assumption above, the standard error of the Monte Carlo estimate decrease by the square root of n.

Simple Monte Carlo example

At 176 simulations, the standard error of μ_{MC} is ~ 0.07. To decrease this to 0.007 (an order of magnitude increase in accuracy), we would need to take a total of $176 \cdot 10^2 = 17600$ simulations.



Iterations for inference



Use `effectiveSize`

```
    var1     var2
169.6231 165.3914
```

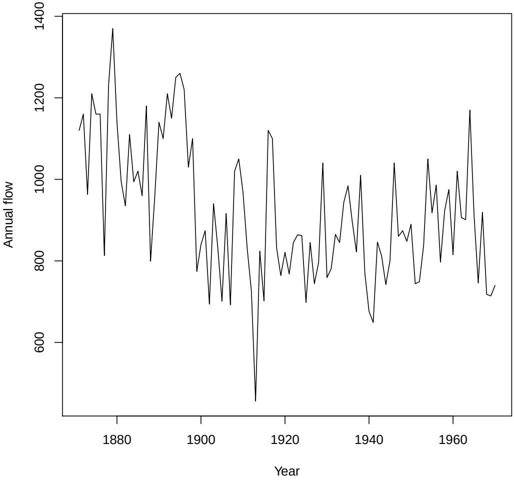
Work flow

- Exploratory data analysis
- Define a model with priors
- Fit the model using MLE techniques
- Inference
  - Fit in WinBUGS
  - Code it up in R/C
    - Choose an MCMC scheme
    - Find the full conditional distributions (if available)
  - Monitor chain convergence
  - Summarize the posterior
- Model checking
  - Diagnostic plots to evaluate model assumptions, e.g. one-step head forecasts

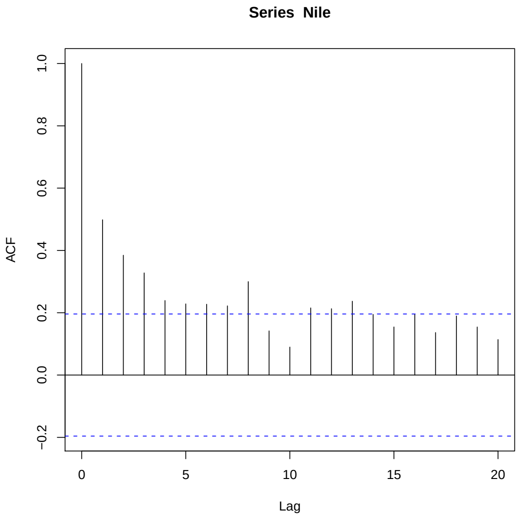
Examples

- Nile flow - local level model
- Spain/Denmark investments - SUTSE

Nile river level



Nile river level



Local level model

$$
\begin{aligned}
Y_t &= \theta_t + v_t & \qquad v_t &\sim N(0, V) \\
\theta_t &= \theta_{t-1} + w_t & \qquad w_t &\sim N(0, W)
\end{aligned}
$$

$$
\begin{aligned}
V &\sim IG(a_V, b_V) \\
W &\sim IG(a_W, b_W) \\
\theta_0 &\sim N(m_0, C_0)
\end{aligned}
$$

where $p(V, W, \theta_0) = p(V)p(W)p(\theta_0)$ and v_t and w_t are independent across time and mutually independent of each other as well as independent of θ_0.

Non-informative priors

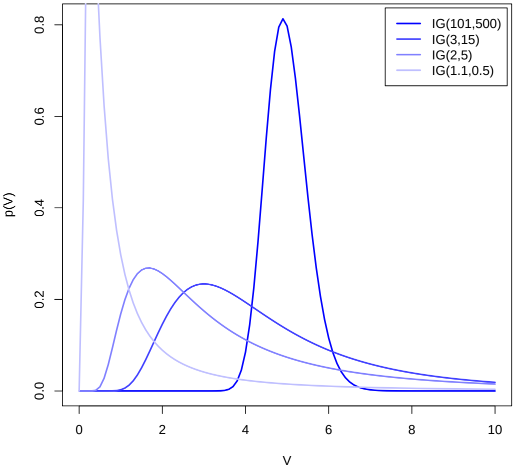
$$
\begin{aligned}
V &\sim IG(a_V, b_V) \\
W &\sim IG(a_W, b_W) \\
\theta_0 &\sim N(m_0, C_0)
\end{aligned}
$$

Non-informative prior

$$
\begin{aligned}
V &\propto 1/V & &\implies a_V = b_V = 0 \\
W &\propto 1/W & &\implies a_W = b_W = 0 \\
\theta_0 &\propto 1 & &\implies m_0 = 0, C_0 = \infty
\end{aligned}
$$

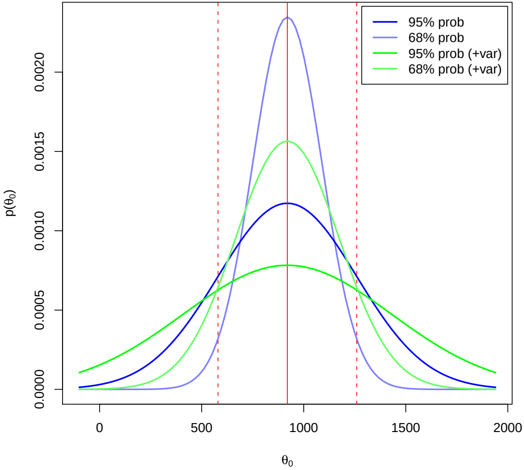
Informative variance priors

Informative prior for V (or W), $E[V] = 5$ with varying accuracy



Informative state prior

"On average, the Nile flow is around 920 $\pm$ 340." With what probability?



MCMC scheme

In DLMs, conditional on unknown parameters, we can sample from the joint state vector at all times using FFBS.

- $p(\theta_{0:T}|V, W, y_{1:T})$ (for references see page 161 of Petris et al.)
- $p(V|\theta_{0:T}, W, y_{1:T})$
- $p(W|\theta_{0:T}, V, y_{1:T})$

Full conditional distributions - the hard way

$$
\begin{aligned}
&p(V|\theta_{0:T}, W, y_{1:T}) \\
&\quad \propto \prod_{t=1}^T p(y_t|\theta_t, V)p(\theta_t|\theta_{t-1}, W)p(V)p(W)p(\theta_0) \\
&\quad \propto \prod_{t=1}^T p(y_t|\theta_t, V)p(V) \\
&\quad = \prod_{t=1}^T N(y_t; \theta_t, V)IG(V; a_V, b_V) \\
&\quad \propto V^{-T/2} \exp\left(-\frac{1}{2V}\sum_{t=1}^T (y_t-\theta_t)^2\right) V^{-a_V-1}\exp\left(-b_V/V\right) \\
&\quad = V^{-(a_V+T/2)-1}\exp\left(-\left[b_V+\frac{1}{2}\sum_{t=1}^T(y_t-\theta_t)^2\right]/V\right) \\
&\quad \propto IG\left(a_V+T/2, b_V+\frac{1}{2}\sum_{t=1}^T(y_t-\theta_t)^2\right)
\end{aligned}
$$

Full conditional distributions - the hard way

$$
\begin{aligned}
&p(W|\theta_{0:T}, V, y_{1:T}) \\
&\quad\propto \textstyle\prod_{t=1}^T p(y_t|\theta_t, V)p(\theta_t|\theta_{t-1}, W)p(V)p(W)p(\theta_0) \\
&\quad\propto \textstyle\prod_{t=1}^T p(\theta_t|\theta_{t-1}, W)p(W) \\
&\quad= \textstyle\prod_{t=1}^T N(\theta_t; \theta_{t-1}, W) IG(W; a_W, b_W) \\
&\quad\propto W^{-T/2} \exp\left(-\frac{1}{2W}\textstyle\sum_{t=1}^T (\theta_t - \theta_{t-1})^2\right) W^{-a_W-1}\exp\left(-b_W/W\right) \\
&\quad= V^{-(a_W+T/2)-1}\exp\left(-\left[b_W + \frac{1}{2}\textstyle\sum_{t=1}^T(\theta_t-\theta_{t-1})^2\right]/W\right) \\
&\quad\propto IG\left(a_W + T/2, b_W + \frac{1}{2}\textstyle\sum_{t=1}^T(\theta_t-\theta_{t-1})^2\right)
\end{aligned}
$$

Full conditional distributions - the easy way

Recall from HW 1b, if $\sigma^2 \sim IG(a,b)$ and $x_i \stackrel{ind}{\sim} N(0,\sigma^2)$, then

$$p(\sigma^2|x_1,x_2,\ldots,x_n) = IG\left(a+n/2, b+\frac{1}{2}\sum_{t=1}^n x_t^2\right).$$

Notice

$$\begin{array}{rl}
V & \sim IG(a_V,b_V) \\
v_t = y_t - \theta_t & \stackrel{ind}{\sim} N(0,V) \\
p(V|y_{1:T},\theta_{0:T},W) & = p(V|y_{1:T},\theta_{1:T}) \\
 & = IG(a_V + T/2, b_V + \frac{1}{2}\sum_{t=1}^T v_t^2)
\end{array}$$

$$\begin{array}{rl}
W & \sim IG(a_W,b_W) \\
w_t = \theta_t - \theta_{t-1} & \stackrel{ind}{\sim} N(0,W) \\
p(W|y_{1:T},\theta_{0:T},V) & = p(W|\theta_{1:T}) \\
 & = IG(a_W + T/2, b_W + \frac{1}{2}\sum_{t=1}^T w_t^2)
\end{array}$$

MCMC scheme revisited

- $p(\theta_{0:T}|\ldots)$ using FFBS
- $p(V|\ldots) = IG(a_V + T/2, b_V + \frac{1}{2}\sum_{t=1}^{T} v_t^2)$
- $p(W|\ldots) = IG(a_W + T/2, b_W + \frac{1}{2}\sum_{t=1}^{T} w_t^2)$

Notice that $p(V|\ldots)$ doesn't depend on W and $p(W|\ldots)$ doesn't depend on V. So our scheme is actually

- $p(\theta_{0:T}|\ldots)$ using FFBS
- $p(V,W|\ldots) =$ $IG(a_V + T/2, b_V + \frac{1}{2}\sum_{t=1}^{T} v_t^2) IG(a_W + T/2, b_W + \frac{1}{2}\sum_{t=1}^{T} w_t^2)$

Coding it up

Begin by creating a function to draw from the posterior of a conjugate inverse gamma

```
drawIGpost <- function(x, a=0, b=0) {
  return(rinvgamma(1, a+length(x)/2, b+sum(x^2)/2))
}
```

Coding it up

Begin by creating a function to draw from the posterior of a conjugate inverse gamma

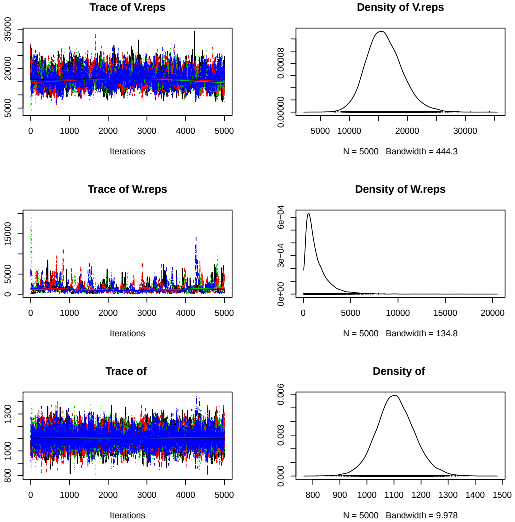
```
for (i in 1:n.reps) {
  cat(i,"\n")
  # Sample states
  mod   <- dlmModPoly(1, dV=V, dW=W)
  filt  <- dlmFilter(Nile, mod)
  theta <- dlmBSample(filt)

  # Sample V and W
  V <- drawIGpost(y-theta[-1])
  W <- drawIGpost(theta[-1]-theta[-n])

  # Save iterations
  V.reps[i] <- V
  W.reps[i] <- W
  theta.reps[i,] = theta
}
```

Running the MCMC

- Run 1
  - Run 1 chain starting from the MLEs
  - Check traceplots for this run
  - Obtain posterior summaries for model parameters
  - Choose initial values that are $<$ minimum and $>$ maximum for each model parameter
- Multi-runs
  - Start multiple chains from combinations of these values
  - Check traceplots and Gelman-Rubin diagnostic for these chains
  - Discard burn-in and produce posterior summaries on remaining iterations
  - If more iterations are needed, initialize new chains from the last iteration of the old chains

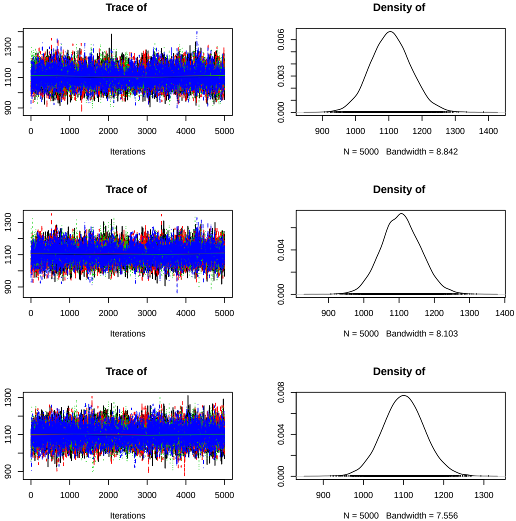
Monitoring convergence

Use `plot.mcmc` in `coda` package.



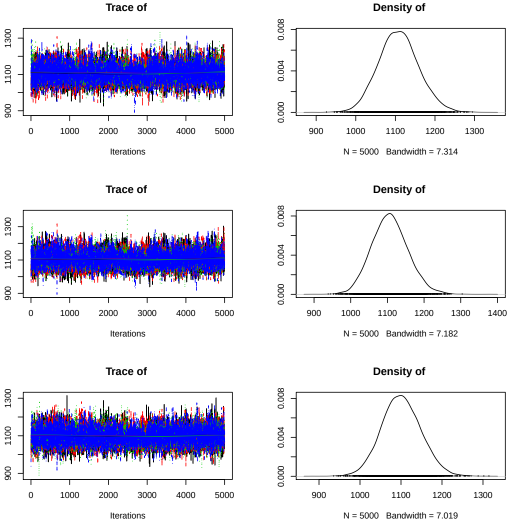
Monitoring convergence

Use `plot.mcmc` in `coda` package.



Monitoring convergence

Use `plot.mcmc` in `coda` package.



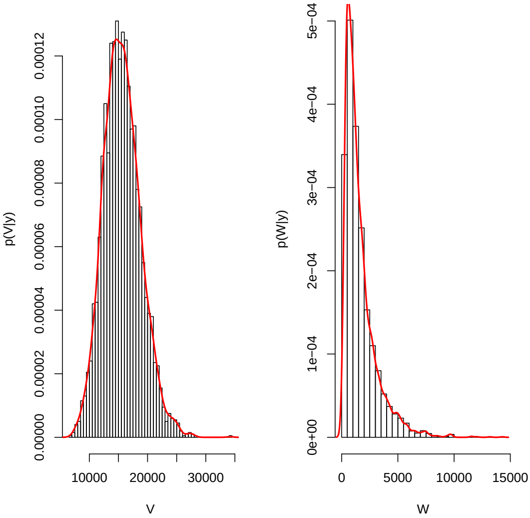
Gelman-Rubin diagnostic

```
> gelman.diag(window(mcmc.results,1,4000))
Potential scale reduction factors:

       Point est. 97.5% quantile
V.reps       1.00           1.00
W.reps       1.00           1.00
             1.00           1.00
             1.00           1.00

             1.00           1.01
             1.00           1.00

Multivariate psrf

1.01
```

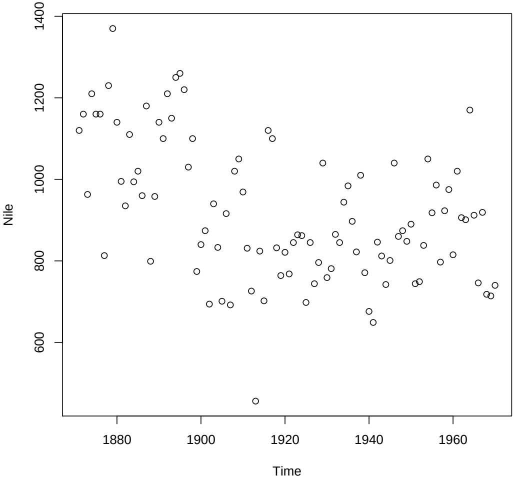
Posterior summaries

```
> summary(window(mcmc.results,4001,5000))

Iterations = 4001:5000
Thinning interval = 1
Number of chains = 4
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean      SD Naive SE Time-series SE
V.reps 15642.8 3187.29  50.3955       125.8969
W.reps  1630.4 1449.03  22.9111       100.2639
        1107.9   73.84   1.1676         1.3148
        1108.7   62.93   0.9951         1.1196

2. Quantiles for each variable:

         2.5%     25%     50%     75%   97.5%
V.reps 9854.5 13459.2 15450.6 17640.9 22337.6
W.reps  241.2   651.7  1183.1  2092.0  5529.9
        964.9  1060.0  1106.7  1153.6  1261.0
        987.1  1066.5  1107.5  1149.0  1238.8
```

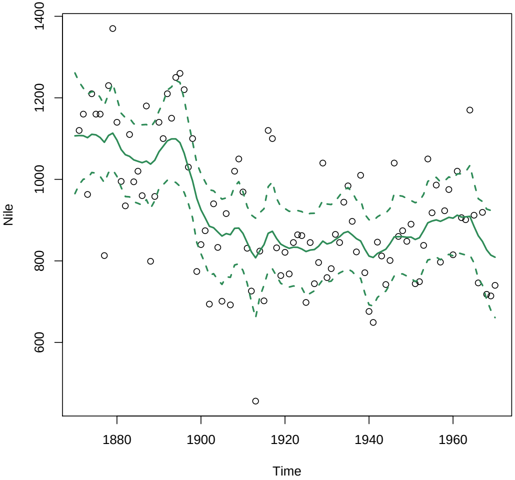
Posterior summaries



Posterior summaries



Posterior summaries



Summaries of functions of parameters

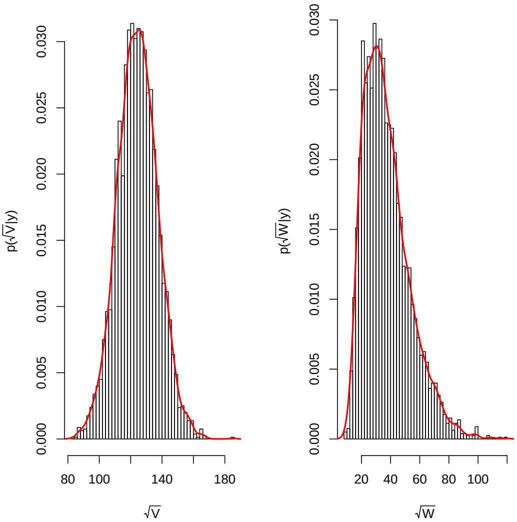
The posterior for $f(\psi)$ is available using the MCMC simulations by plugging our iterations $\psi^{(i)}$ into $f(\cdot)$ and calculating desired quantities, e.g.

$$E[f(\psi)] \approx \frac{1}{n}\sum_{i=1}^{n} f(\psi^{(i)}).$$

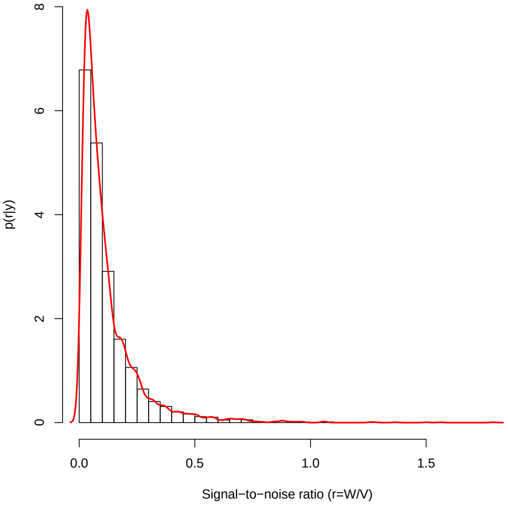
For example,

- $f(\theta_{0:T}, V, W) = \sqrt{(}V)$
- $f(\theta_{0:T}, V, W) = \sqrt{(}W)$
- $f(\theta_{0:T}, V, W) = W/V$ (signal-to-noise ratio)
- $f(\theta_{0:T}, V, W) = P(W/V < 1)$

Standard deviations



Signal-to-noise ratio



$$P(r < 1) = 0.998.$$

The model

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t & v_t &\sim N(0,V) \\
\theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0,W)
\end{aligned}
$$

where V is scalar and W is diagonal with elements W_i and assumed priors

$$
\begin{aligned}
p(V, W_1, \ldots, W_p, \theta_0) &= p(V)p(\theta_0)\textstyle\prod_{i=1}^p p(W_i) \\
V &\sim IG(a_V, b_V) \\
W_i &\sim IG(a_{W_i}, b_{W_i}) \\
\theta_0 &\sim N(m_0, C_0)
\end{aligned}
$$

Linear trend model

For example

$$\begin{aligned} Y_t &= F\theta_t + v_t & v_t &\sim N(0,V) \\ \theta_t &= G\theta_{t-1} + w_t & w_t &\sim N_p(0,W) \end{aligned}$$

where $F = (1,0)$, $G[1,1] = G[1,2] = G[2,2] = 1$, $G[2,1] = 0$, and W is diagonal with elements W_i and assumed priors

$$\begin{aligned} p(V, W_1, \ldots, W_p, \theta_0) &= p(V)p(\theta_0)p(W_1)p(W_2) \\ V &\sim IG(a_V, b_V) \\ W_1 &\sim IG(a_{W_1}, b_{W_1}) \\ W_2 &\sim IG(a_{W_2}, b_{W_2}) \\ \theta_0 &\sim N(m_0, C_0) \end{aligned}$$

Rewrite the linear trend model

$$\begin{aligned} Y_t &= \mu_t + v_t & v_t &\sim N(0,\sigma^2) \\ \mu_t &= \mu_{t-1} + \beta_t + w_{t,1} & w_{t,1} &\sim N(0,\sigma^2_\mu) \\ \beta_t &= \beta_{t-1} + w_{t,2} & w_{t,2} &\sim N(0,\sigma^2_\beta) \end{aligned}$$

where $w_{t,1}$ and $w_{t,2}$ are independent.

What are the full conditionals for σ^2, σ^2_μ, and σ^2_β?

- $p(\sigma^2|\ldots) = IG\left(a_{\sigma^2} + T/2, b_{\sigma^2} + \frac{1}{2}\sum_{t=1}^T v_t^2\right)$
- $p(\sigma^2_\mu|\ldots) = IG\left(a_{\sigma^2_\mu} + T/2, b_{\sigma^2_\mu} + \frac{1}{2}\sum_{t=1}^T w_{t,1}^2\right)$
- $p(\sigma^2_\beta|\ldots) = IG\left(a_{\sigma^2_\beta} + T/2, b_{\sigma^2_\beta} + \frac{1}{2}\sum_{t=1}^T w_{t,2}^2\right)$

and importantly, they are independent!

More generally

$$
\begin{aligned}
Y_t &= F_t\theta_t + v_t & v_t &\sim N(0,V) \\
\theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p(0,W)
\end{aligned}
$$

where W is diagonal with elements W_i and all variances have independent inverse gamma priors.

The full conditionals for parameters are

- $p(V|\ldots) = IG\left(a_V + T/2, b_V + \frac{1}{2}\sum_{t=1}^T v_t^2\right)$
- $p(W_i|\ldots) = IG\left(a_{W_i} + T/2, b_{W_i} + \frac{1}{2}\sum_{t=1}^T w_{t,i}^2\right)$

and again, they are independent!

MCMC scheme for models with d inverse gamma priors

Two-stage Gibbs sampler

- Use FFBS to sample from $p(\theta_{0:T}|\ldots)$
- Jointly sample $V, W_1, \ldots, W_p$ by sampling their full conditionals
  - $p(V|\ldots)$
  - $p(W_i|\ldots)$ for $i \in (1, 2, \ldots, p)$.

Implemented in `dlmGibbsDIG`.

Suppose we assume the model

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\sim N_m\left(0, \Phi_0^{-1}\right) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_p\left(0, \Phi_1^{-1}\right) \end{aligned}$$

where Φ_0 is an $m \times m$ observation precision matrix and Φ_1 is a $p \times p$ evolution precision matrix. It will be convenient to choose independent Wishart distributions for the prior for these precision matrices, i.e.

$$p\left(\Phi_0, \Phi_1\right) = p\left(\Phi_0\right) p\left(\Phi_1\right) = \mathcal{W}\left(\Phi_0; \nu_0, S_0\right) \mathcal{W}\left(\Phi_1; \nu_1, S_1\right)$$

where

$$\mathcal{W}(P; \nu, S) = \frac{|S|^\nu |P|^{\frac{\nu-p-1}{2}}}{\Gamma_p(\nu)} \exp\left(-tr(SP)\right)$$

is a distribution on symmetric, positive definite matrices P with parameters $\nu > (p-1)/2$ and S, symmetric non-singular matrix.

Full conditionals for the precision matrices

Wishart distributions are conditionally conjugate in this model:

$$p\left(\Phi_0 \middle| \ldots\right) = \mathcal{W}\left(\nu_0 + T/2, S_0 + \frac{1}{2}SS_y\right)$$

where $SS_y = \sum_{t=1}^T \left(y_t - F_t\theta_t\right)\left(y_t - F_t\theta_t\right)^\top$.

$$p\left(\Phi_1 \middle| \ldots\right) = \mathcal{W}\left(\nu_1 + T/2, S_1 + \frac{1}{2}SS_\theta\right)$$

where $SS_\theta = \sum_{t=1}^T \left(\theta_t - G_t\theta_{t-1}\right)\left(\theta_t - G_t\theta_{t-1}\right)^\top$.

Again, $p\left(\Phi_0, \Phi_1 \middle| \ldots\right) = p\left(\Phi_0 \middle| \ldots\right) p\left(\Phi_1 \middle| \ldots\right)$.

To draw from these distributions, use `rwishart` in `dlm` package which has arguments degrees of freedom δ and scale matrix V_0^{-1} where $\mathcal{W}(\delta/2, V_0/2)$.

The model

Consider the model with block-diagonal evolution covariance:

$$\begin{aligned} Y_t &= F_t\theta_t + v_t & v_t &\sim N_m(0, \Phi_0^{-1}) \\ \theta_t &= G_t\theta_{t-1} + w_t & w_t &\sim N_{p*}(0, W) \end{aligned}$$

where W is block-diagonal with elements W_i.Set $\Phi_i^{-1} = W_i$ and give $\Phi_0, \Phi_1, \ldots, \Phi_d$ independent Wishart priors $\Phi_i \sim \mathcal{W}(\nu_i, S_i)$.

Rewritten univariate model

For combining individual components, e.g. polynomial trend, seasonal, dynamic regression, G_t is block diagonal with elements $G_{i,t}$ relating to W_i and the model can be re-written

$$\begin{array}{rll} Y_t & = F_t\theta_t + v_t & v_t \sim N(0, \Phi_0^{-1}) \\ \theta_{1,t} & = G_{1,t}\theta_{1,t-1} + w_{1,t} & w_{1,t} \sim N_{p_1}(0, \Phi_1^{-1}) \\ & \vdots & \\ \theta_{i,t} & = G_{i,t}\theta_{i,t-1} + w_{i,t} & w_{i,t} \sim N_{p_i}(0, \Phi_i^{-1}) \\ & \vdots & \\ \theta_{p,t} & = G_{p,t}\theta_{p,t-1} + w_{p,t} & w_{p,t} \sim N_{p_d}(0, \Phi_d^{-1}) \end{array}$$

where $w_{i,t}$ are independent across i. Then

$$SS_{ii,t} = (\theta_{i,t} - G_{i,t}\theta_{i,t-1})(\theta_{i,t} - G_{i,t}\theta_{i,t-1})^\top.$$

Multivariate models

Let

$$SS_t = (\theta_t - G_t\theta_{t-1})(\theta_t - G_t\theta_{t-1})^\top$$

and partition it according to

$$SS_t = \left[\begin{array}{ccc} SS_{11,t} & \cdots & SS_{1d,t} \\ \vdots & \ddots & \vdots \\ SS_{d1,t} & \cdots & S_{dd,t} \end{array}\right]$$

where the partition is according to the partition in $\Phi = \text{blockdiag}(\Phi_1, \ldots, \Phi_d)$.

Full conditional distributions

$$p\left(\Phi_0^{-1}|\ldots\right) = \mathcal{W}\left(\nu_0 + T/2, S_0 + \frac{1}{2}SS_y\right)$$

where $SS_y = \sum_{t=1}^T \left(y_t + F_t\theta_t\right)\left(y_t - F_t\theta_t\right)^\top$.

$$p\left(\Phi_i^{-1}|\ldots\right) = \mathcal{W}\left(\nu_i + T/2, S_i + \frac{1}{2}SS_{\theta_i}\right)$$

where $SS_{\theta_i} = \sum_{t=1}^T SS_{ii,t}$ given on the previous page.

Once again, $p\left(\Phi_0^{-1}, \Phi_1^{-1}, \ldots, \Phi_d^{-1}|\ldots\right) = p\left(\Phi_0^{-1}|\ldots\right)\prod_{i=1}^d p\left(\Phi_i^{-1}|\ldots\right)$.

Denmark and Spain investments

SUTSE model

$$\begin{aligned} Y_t &= (F \otimes I_2)\theta_t + v_t & v_t &\sim N_2(0, \Phi_0^{-1}) \\ \theta_t &= (G \otimes I_2)\theta_{t-1} + w_t & w_t &\sim N_4(0, W) \end{aligned}$$

where $W = \text{blockdiag}(W_1, W_2)$, $\Phi_1^{-1} = W_1$, and $\Phi_2^{-1} = W_2$.

Assume independent Wishart priors

$$\begin{aligned} p(\Phi_0) &= \mathcal{W}\left(\tfrac{\delta_0+1}{2}, \tfrac{1}{2}V_0\right) & V_0 &= (\delta_0 - 2)\begin{bmatrix} 10^2 & 0 \\ 0 & 500^2 \end{bmatrix} \\ p(\Phi_1) &= \mathcal{W}\left(\tfrac{\delta_1+1}{2}, \tfrac{1}{2}W_{\mu,0}\right) & W_{\mu,0} &= (\delta_1 - 2)\begin{bmatrix} 0.01^2 & 0 \\ 0 & 0.01^2 \end{bmatrix} \\ p(\Phi_2) &= \mathcal{W}\left(\tfrac{\delta_2+1}{2}, \tfrac{1}{2}W_{\beta,0}\right) & W_{\beta,0} &= (\delta_2 - 2)\begin{bmatrix} 5^2 & 0 \\ 0 & 100^2 \end{bmatrix} \end{aligned}$$

where $\delta_0 = \delta_2 = 3$ and $\delta_1 = 100$.

MCMC sampling

MCMC Scheme:

- Sample $\theta_{0:T} \sim p(\theta_{0:T}|\ldots)$ using FFBS
- Sample $p(\Phi_0, \Phi_1, \Phi_2|\ldots)$ jointly

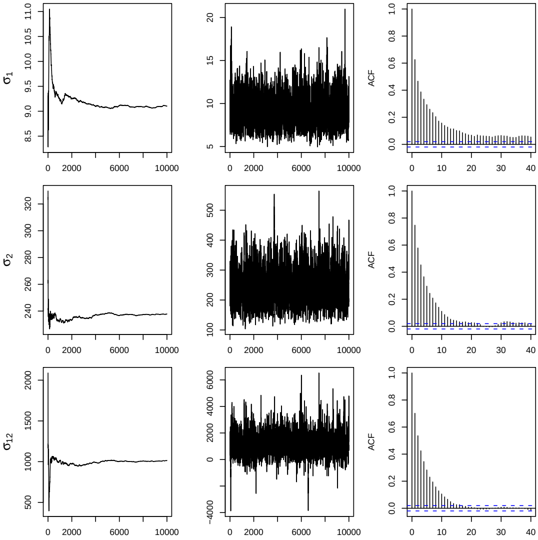
$$\begin{aligned}
p(\Phi_0|\ldots) &= \mathcal{W}\left(\tfrac{\delta_0+1+T}{2}, \tfrac{1}{2}(V_0 + SS_y)\right) \\
p(\Phi_1|\ldots) &= \mathcal{W}\left(\tfrac{\delta_1+1+T}{2}, \tfrac{1}{2}(W_{\mu,0} + SS_{1\cdot})\right) \\
p(\Phi_2|\ldots) &= \mathcal{W}\left(\tfrac{\delta_2+1+T}{2}, \tfrac{1}{2}(W_{\beta,0} + SS_{2\cdot})\right)
\end{aligned}$$

where

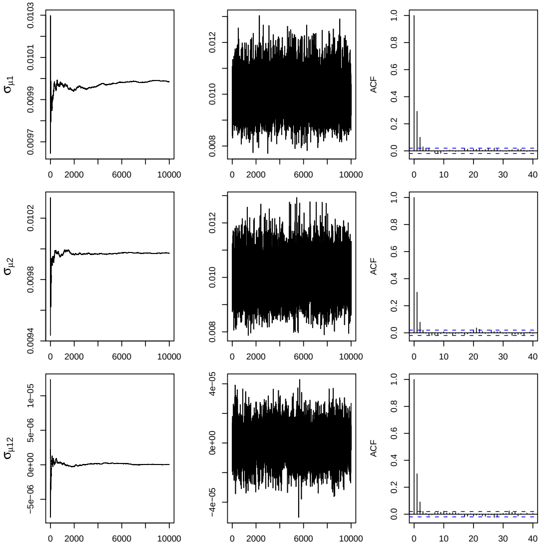
$$SS_{i\cdot} = \sum_{t=1}^{T} SS_{ii,t}.$$

provided earlier.

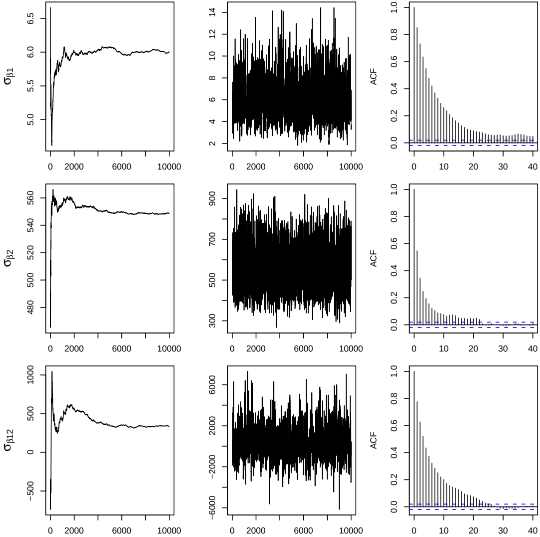
Convergence and autocorrelation



Convergence



Convergence



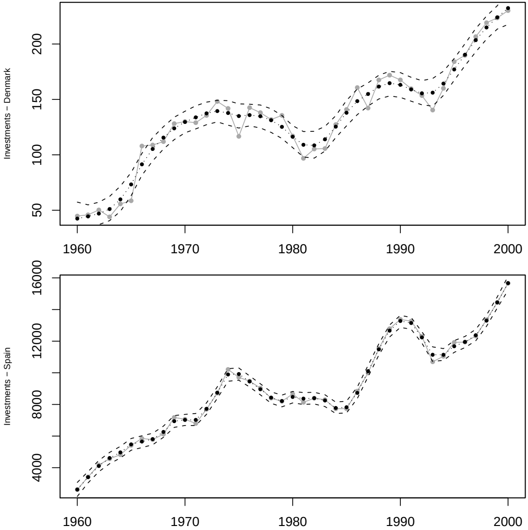
Posterior covariance expectations

$$E[V|y_{1:T}] \quad = \quad \left[\begin{array}{cccc} 86 & (1) & 1026 & (23) \\ & & 59340 & (807) \end{array}\right]$$

$$E[W_\mu|y_{1:T}] \quad = 1e-5 \left[\begin{array}{cccc} 9.97 & (0.02) & 0.016 & (0.014) \\ & & 10.04 & (0.02) \end{array}\right]$$

$$E[W_\beta|y_{1:T}] \quad = \quad \left[\begin{array}{cccc} 38.3 & (0.8) & 305 & (41) \\ & & 311073 & (2346) \end{array}\right]$$

Posterior μ_t



Types of missing data

Complete data $Y_{i,t}$ and missing indicator $M_{i,t}$ where $M_{i,t} = 1$ if observation Y_{it} is missing and 0 otherwise. Let Y_{obs} contain all the data that is observed while Y_{mis} contains all the data that is missing with $Y = (Y_{\text{obs}}, Y_{\text{mis}})$. Then several types of missing-ness are possible:

- Missing completely at random (MCAR)

$$p(M|Y,\phi) = p(M|\phi).$$

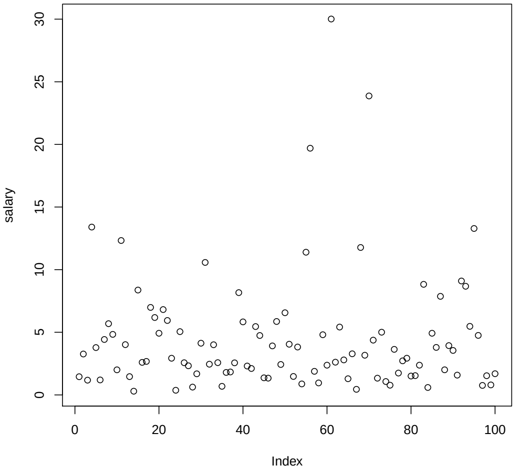
- Missing at random (MAR)

$$p(M|Y,\phi) = p(M|Y_{obs},\phi).$$

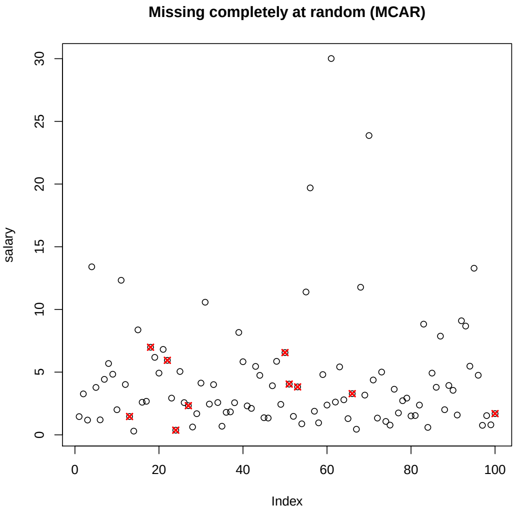
- Not missing at random

$$p(M|Y,\phi) \text{ depends on } Y_{\text{mis}}.$$

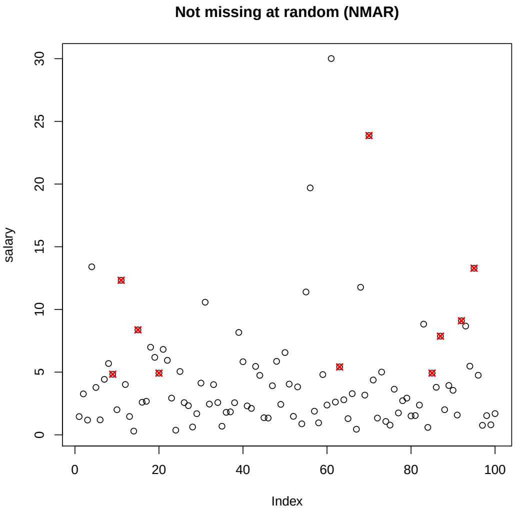
No missing data



Missing completely at random



Not missing at random



Missing data in multivariate DLMs

Two situations:

- Totally missing: at time t, Y_t is completely missing
- Partially missing: at time t, part of Y_t is observed

Totally missing

Recall 'Kalman filter' lecture: missing data are handled trivially while filtering.

$$m_t = a_t \qquad C_t = R_t.$$

Unknown fixed parameters are sampled without these data, e.g. scalar V

$$p(\phi_V | \ldots) \quad = G\left(a + \frac{T'}{2}, b + \sum_{t \in \text{obs}} (y_t - F_t\theta_t)^2\right)$$

where 'obs' is a vector of times when the data are observed and $T' \le T$ is the length of obs.

e.g. matrix V

$$p(\Phi_V | \ldots) \quad = W\left(a + \frac{T'}{2}, b + \sum_{t \in \text{obs}} (y_t - F_t\theta_t)(y_t - F_t\theta_t)^\top\right).$$

Partially missing when filtering

Suppose M_t is the matrix that is built by taking an identity matrix and removing the rows of any missing observations in y_t. Then $\tilde{y}_t = M_t y_t$ contains only the observed data. The correct observation equation to consider is

$$\tilde{y}_t = \tilde{F}_t \theta_t + \tilde{v}_t \qquad \tilde{v}_t \sim N(0, \tilde{V}_t).$$

What are $\tilde{F}_t$ and $\tilde{V}_t$?

- $\tilde{F}_t = M_t F_t$
- $\tilde{V}_t = M_t V_t M_t^\top$

Partially missing in MCMC

Let $Y = (Y_{\text{obs}}, Y_{\text{mis}})$. If we build an MCMC with only the observed data, then our scheme will look like

- Sample $p(\theta|Y_{\text{obs}}, \psi)$ via FFBS
- Sample $p(\psi|Y_{\text{obs}}, \theta)$.

For example, consider the observation precision matrix Φ_V as the only unknown parameter. What is it's full conditional distribution?

$$p(\Phi_V|Y_{\text{obs}}, \theta) \propto p(Y_{\text{obs}}|\Phi_V, \theta)p(\Phi_V)$$

Who knows?

Partially missing in MCMC

Let $Y = (Y_{\text{obs}}, Y_{\text{mis}})$. Augment the MCMC to simulate the missing values, then our scheme will look like

- Sample $p(\theta|Y,\psi)$ via FFBS
- Sample $p(\psi|Y,\theta)$
- Sample $p(Y_{\text{mis}}|Y_{\text{obs}},\theta,\psi)$.

This works since

$$p(\theta,\psi|Y_{\text{obs}}) = \int p(\theta,\psi,Y_{\text{mis}}|Y_{\text{obs}})dY_{\text{mis}}.$$

Partially missing in MCMC

How to simulate $p(Y_{\text{mis}}|Y_{\text{obs}}, \theta, \psi)$?

First note,

$$p(Y_{\text{mis}}|Y_{\text{obs}}, \theta, \psi) = \prod_{t=1}^{T} p(Y_{\text{mis},t}|Y_{\text{obs},t}, \theta_t, \psi).$$

Second note,

$$\left(\begin{array}{c} Y_{\text{mis},t} \\ Y_{\text{obs},t} \end{array} \right) \sim N\left(\left[\begin{array}{c} F\theta_{\text{mis},t} \\ F\theta_{\text{obs},t} \end{array} \right], \left[\begin{array}{cc} V_{\text{mis}} & V_{\text{m,o}} \\ V_{\text{o,m}} & V_{\text{obs}} \end{array} \right] \right).$$

Goal

With all fixed parameters known:

$$\begin{aligned} p(y_{t+k},\theta_{t+k}|y_{1:t}) &= \int p(y_{t+k},\theta_{t+k},\theta_{t+(k-1)}|y_{1:t})d\theta_{t+(k-1)} \\ &= \int p(y_{t+k},\theta_{t+k}|\theta_{t+(k-1)})p(\theta_{t+(k-1)}|y_{1:t})d\theta_{t+(k-1)} \end{aligned}$$

To get $p(y_{t+k},\theta_{t+k}|\theta_t)$ just use the Kalman filter with missing data from y_{t+1} up to $y_{t+(k-1)}$.

With unknown fixed parameters:

$$\begin{aligned} &p(y_{t+k},\theta_{t+k}|y_{1:t}) = \\ &\quad = \int p(y_{t+k},\theta_{t+k}|\theta_{t+(k-1)},\psi)p(\theta_{t+(k-1)},\psi|y_{1:t})d\theta_{t+(k-1)}d\psi. \end{aligned}$$

Now we can't just use the Kalman filter due to the unknown fixed parameters. Instead, we need to integrate over their posteriors.

MCMC Forecasting

After completing the MCMC, follow this procedure

- For each iteration $j = 1, 2, \ldots, J$ in the MCMC chain post burn-in:
  - Run a Kalman filter (`dlmFilter`) on your data using $\psi^{(j)}$ to obtain $p(\theta_t|y_{1:t}, \psi^{(j)}) = N(m_t^{(j)}, C_t^{(j)})$.
  - Forecast ahead (`dlmForecast`) to obtain $p(y_{t+k}|y_{1:t}, \psi^{(j)}) = N(f_t(k)^{(j)}, Q_t(k)^{(j)})$ (see section 2.8 in Petris)
  - Calculate mean and 95% intervals for $p(y_{t+k}|y_{1:t}, \psi^{(j)})$, i.e. $f_t(k)^{(j)}$ $(f_t(k)^{(j)} - 1.96\sqrt{Q_t(k)^{(j)}}, f_t(k)^{(j)} + 1.96\sqrt{Q_t(k)^{(j)}})$ if Q is scalar, otherwise do this component-wise.

This provides a set of means and 95% intervals, one for each MCMC iteration j.

MCMC Forecasting

To find the marginal mean and 95% interval, average these means and 95% intervals for all j, i.e.

$$
\begin{aligned}
E[y_{t+k}|y_{1:t}] &\approx \frac{1}{J}\sum_{j=1}^{J} f_t(k)^{(j)} \\
Q_{2.5\%}[y_{t+k}|y_{1:t}] &\approx \frac{1}{J}\sum_{j=1}^{J} f_t(k)^{(j)} - 1.96\sqrt{Q_t(k)^{(j)}} \\
Q_{97.5\%}[y_{t+k}|y_{1:t}] &\approx \frac{1}{J}\sum_{j=1}^{J} f_t(k)^{(j)} + 1.96\sqrt{Q_t(k)^{(j)}}
\end{aligned}
$$

If you have many MCMC iterations, you can use fewer iterations for this forecast by thinning the chain.