STATS 207: Time Series Analysis
Autumn 2020

Lecture 15: Estimation of State-Space Models

Dr. Alon Kipnis

November 2nd 2020

Genera Info

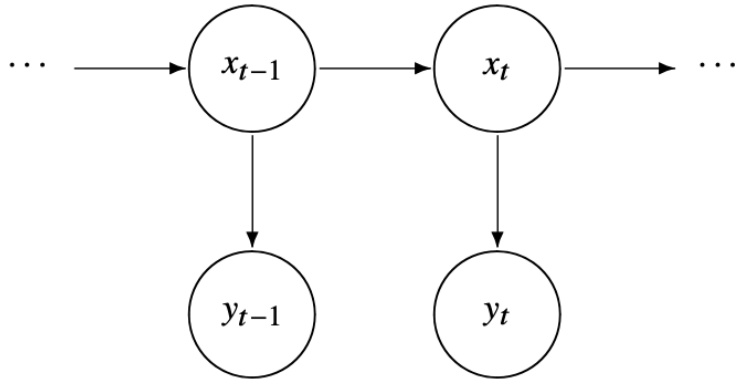
- HW3 is due Today.
- HW4 is out. Due on Monday 11/16/2020.
- Next Professor David Donoho will give another guest lecture: "Bootstrap Reality Check & Technical Trading Rules"

Outline

State-Space Model (review)

- **State Equation**:

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \Upsilon \boldsymbol{u}_t + \boldsymbol{w}_t,$$



where

  - $\boldsymbol{x}_t$, $\boldsymbol{w}_t$ have dimension p,
  - $\boldsymbol{u}_t$ has dimension r,
  - $\boldsymbol{w}_t \overset{iid}{\sim} \mathcal{N}(0, Q)$.

- **Observation Equation**:

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \Gamma \boldsymbol{u}_t + \boldsymbol{v}_t$$

  - $\boldsymbol{y}_t$, $\boldsymbol{v}_t$ have dimension q,
  - $\boldsymbol{v}_t \overset{iid}{\sim} \mathcal{N}(0, R)$.

- **Initial Conditions**: $\boldsymbol{x}_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$.

Kalman Filter (review)

- **Goal**: $x_t^t = \mathbb{E}[\boldsymbol{x}_t|\boldsymbol{y}_{1:t}]$, $P_t^t = \mathrm{Var}(\mathbb{E}[\boldsymbol{x}_t|\boldsymbol{y}_{1:t}])$
- **Initial Conditions**: $\boldsymbol{x}_0^0 = \mu_0$, $P_0^0 = \Sigma_0$
- **Recursion** $t = 1, 2, \ldots, n$:

- **State Prediction**: $\boldsymbol{x}_t^{t-1} = \Phi \boldsymbol{x}_{t-1}^{t-1} + \Upsilon \boldsymbol{u}_t$
- **State Covariance**: $P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q$

- Observation **Prediction Errors**: $\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t$
- **Kalman Gain**: $K_t \equiv P_t^{t-1} A_t' \left(A_t P_t^{t-1} A_t' + R\right)^{-1}$

- **State Estimation**: $\boldsymbol{x}_t^t = \boldsymbol{x}_t^{t-1} + K_t \boldsymbol{\epsilon}_t$
- **State Uncertainty**: $P_t^t = (I - K_t A_t) P_t^{t-1}$

Kalman Smoother (reviw)

- **Goal**: Estimate the **Best non-causal Estimate**

$$\boldsymbol{x}_t^n = \mathbb{E}\left[\boldsymbol{x}_t | \boldsymbol{y}_{1:n}\right], \quad t < n.$$

and the **Variance of Best non-causal Estimate**:

$$P_t^n = \mathbb{E}\left[(\boldsymbol{x}_t - \boldsymbol{x}_t^n)(\boldsymbol{x}_t - \boldsymbol{x}_t^n)'\right].$$

- **Initial Conditions**: $\boldsymbol{x}_n^n$, P_n^n (output of standard Kalman filter at time $t = n$).
- **Recursion** $t = n, n-1, \ldots, 1$:

$$\boldsymbol{x}_{t-1}^n = \boldsymbol{x}_{t-1}^{t-1} + J_{t-1}(\boldsymbol{x}_t^n - \boldsymbol{x}_t^{t-1})$$
$$P_{t-1}^n = P_{t-1}^{t-1} + J_{t-1}(P_t^n - P_t^{t-1})J_{t-1}',$$

where

$$J_{t-1} = P_{t-1}^{t-1}\Phi_t'[P_t^{t-1}]^{-1}.$$

Example 6.5: Prediction, Filtering and Smoothing

- Univariate state-space model:

$$x_t = x_{t-1} + w_t,$$

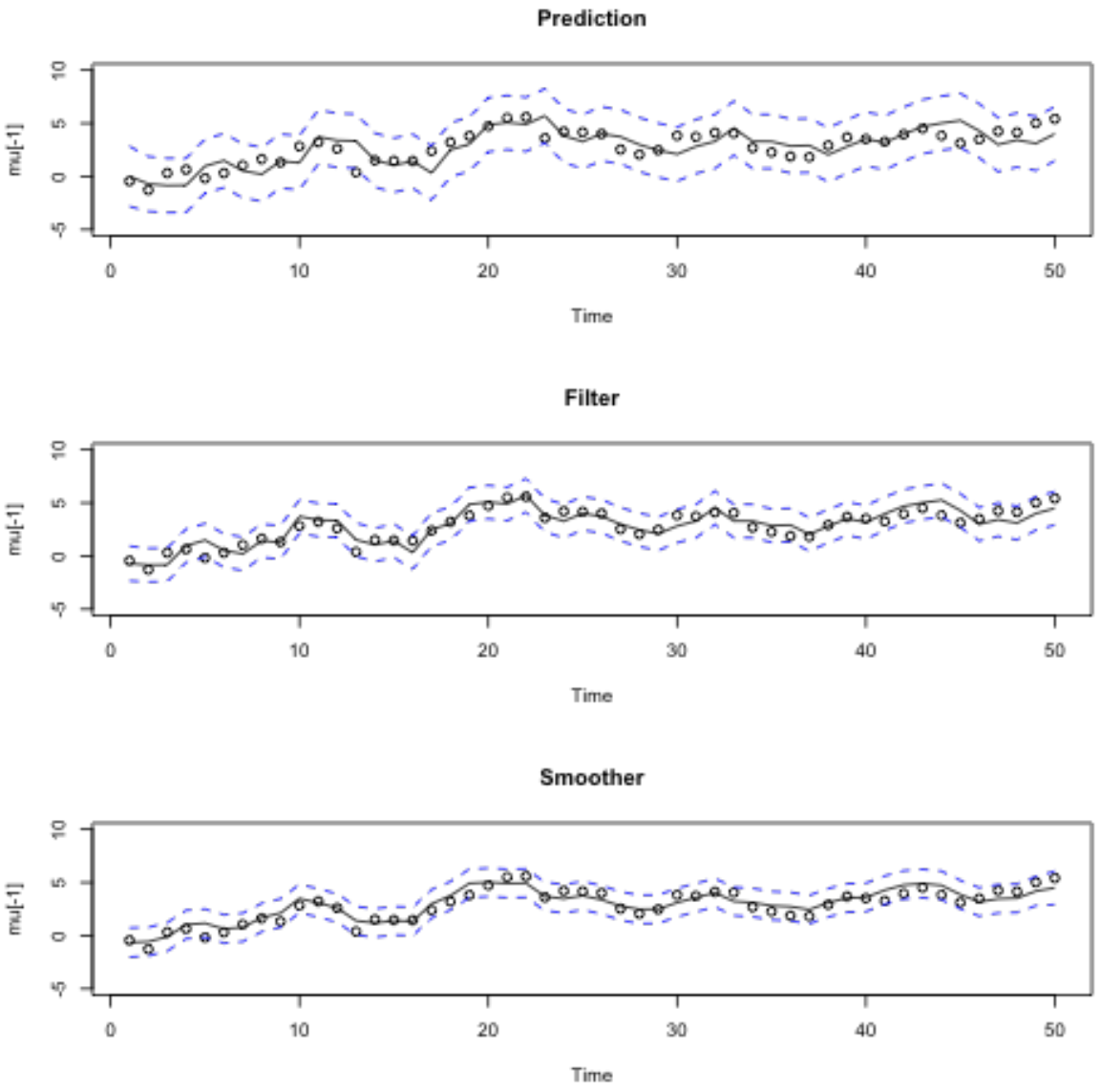
$$y_t = x_t + v_t,$$

$x_0 \sim \mathcal{N}(0, 1)$.

```
# generate data
set.seed(1); num = 50
w = rnorm(num+1,0,1); v = rnorm(num,0,1)
mu = cumsum(w) # state: mu[0], mu[1],..., mu[50]
y = mu[-1] + v # obs: y[1],..., y[50]

# filter, smooth and predict
ks = Ksmooth0(num, y, A=1, mu0=0, Sigma0=1, Phi=1, cQ=1, cR=1)
```

Example 6.5: Prediction, Filtering and Smoothing (cont'd)



Estimation of State-Space Model Parameters

Estimation, I

- **Parameter Vector**:

$$\Theta \equiv (\mu_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma)\,.$$

Estimation, I

- **Parameter Vector**:

$$\Theta \equiv (\mu_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma)\,.$$

- **Innovations**:

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t.$$

Estimation, I

- **Parameter Vector**:

$$\Theta \equiv (\mu_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma)\,.$$

- **Innovations**:

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t.$$

- Innovations **Covariance**:

$$\Sigma_t = A_t P_t^{t-1} A_t' + R.$$

Estimation, I

- **Parameter Vector**:

$$\Theta \equiv (\mu_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma)\,.$$

- **Innovations**:

$$\boldsymbol{\epsilon}_t = \boldsymbol{y}_t - A_t \boldsymbol{x}_t^{t-1} - \Gamma \boldsymbol{u}_t.$$

- Innovations **Covariance**:

$$\Sigma_t = A_t P_t^{t-1} A_t' + R.$$

- **Likelihood**:

$$-\log L(y_{1:n}; \Theta) = \text{const} + \frac{1}{2}\sum_{t=1}^{n} \log\det\left(\Sigma_t(\Theta)\right) + \frac{1}{2}\sum_{t=1}^{n} \boldsymbol{\epsilon}_t(\Theta)' \Sigma_t^{-1}(\Theta) \boldsymbol{\epsilon}_t(\Theta)$$

Estimation of State-Space Models, II

Newton-Raphson Approach:

- Select **initial values** $\Theta^{(0)}$.
- For $j = 1, 2, \ldots,$
  1. Run **Kalman Filter** with $\Theta^{(j-1)}$; get **innovations** and **covariances**.
  2. Apply **Newton-Raphson** with $-\log L(y_{1:n}; \Theta)$; get **next estimate** $\Theta^{(j)}$.

Newton-Raphson (Example 3.30)

- **Goal:** Minimize $L(\Theta)$.

Newton-Raphson (Example 3.30)

- **Goal:** Minimize $L(\Theta)$.
- **Assume**:

$$\nabla L(\Theta) = \left(\frac{\partial L(\Theta)}{\partial \Theta_1}, \ldots, \frac{\partial L(\Theta)}{\partial \Theta_k} \right)',$$

$$\mathrm{H}L(\Theta) = \left\{ \frac{\partial L^2(\Theta)}{\partial \Theta_i \partial \Theta_j}, \right\}_{i,j=1}^{k},$$

are available and $\mathrm{H}L(\Theta)$ is invertible.

Newton-Raphson (Example 3.30)

- **Goal:** Minimize $L(\Theta)$.
- **Assume**:

$$\nabla L(\Theta) = \left(\frac{\partial L(\Theta)}{\partial \Theta_1}, \ldots, \frac{\partial L(\Theta)}{\partial \Theta_k} \right)',$$

$$\mathrm{H}L(\Theta) = \left\{ \frac{\partial L^2(\Theta)}{\partial \Theta_i \partial \Theta_j}, \right\}_{i,j=1}^{k},$$

  are available and $\mathrm{H}L(\Theta)$ is invertible.

- **Newton-Raphson:** For $j = 1, 2, \ldots$, apply

$$\Theta^{(j)} = \Theta^{(j-1)} - \left[\mathrm{H}L(\Theta^{(j-1)}) \right]^{-1} \nabla L \left(\Theta^{(j-1)} \right).$$

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, I

- AR(1) hidden in noise model:

$$x_t = \phi x_{t-1} + w_t, \qquad x_0 \sim \mathcal{N}(0, 1)$$
$$y_t = x_t + v_t.$$

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, I

- AR(1) hidden in noise model:

$$x_t = \phi x_{t-1} + w_t, \qquad x_0 \sim \mathcal{N}(0, 1)$$
$$y_t = x_t + v_t.$$

- In **state-space** form:
  - State $[x_t]$
  - $\Phi = [\phi]$
  - $A = [1]$
  - $Q = R = [1]$
  - $\Upsilon = \Gamma = [0]$.

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, I

- AR(1) hidden in noise model:

$$x_t = \phi x_{t-1} + w_t, \qquad x_0 \sim \mathcal{N}(0, 1)$$
$$y_t = x_t + v_t.$$

- In **state-space** form:
  - State $[x_t]$
  - $\Phi = [\phi]$
  - $A = [1]$
  - $Q = R = [1]$
  - $\Upsilon = \Gamma = [0]$.

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, I

- AR(1) hidden in noise model:

$$x_t = \phi x_{t-1} + w_t, \qquad x_0 \sim \mathcal{N}(0,1)$$
$$y_t = x_t + v_t.$$

- In **state-space** form:
  - State $[x_t]$
  - $\Phi = [\phi]$
  - $A = [1]$
  - $Q = R = [1]$
  - $\Upsilon = \Gamma = [0]$.

```
# Generate Data
set.seed(999); num = 100
x = arima.sim(n=num+1, list(ar=.8), sd=1)
y = ts(x[-1] + rnorm(num,0,1))
```

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, II

- **Initial estimate** using **method of moments**:

$$\gamma_x(h) = \frac{\phi^h}{1-\phi^2}\sigma_w^2, \qquad \gamma_y(h) = \gamma_x(h) + \sigma_v^2.$$

Use

$$\phi^{(0)} = \frac{\hat{\gamma}_y(2)}{\hat{\gamma}_y(1)}, \qquad \sigma_w^{2\,(0)} = \frac{1-\phi^{(0)^2}}{\phi^{(0)}}\hat{\gamma}_y(1),$$

$$\sigma_v^{2(0)} = \hat{\gamma}_y(0) - \frac{\sigma_w^{2\,(0)}}{1-\phi^{(0)^2}}.$$

(good estimate when the observation noise is small $\sigma_v^2 \ll \gamma_x(0)$)

```
# Initial Estimates
u = ts.intersect(y, lag(y,-1), lag(y,-2))
varu = var(u)
coru = cor(u)
phi = coru[1,3]/coru[1,2]
q = (1-phi^2)*varu[1,2]/phi
r = varu[1,1] - q/(1-phi^2)
(init.par = c(phi, sqrt(q), sqrt(r))) # = .91, .51, 1.03
```

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, III

```
# Function to evaluate the likelihood
Linn = function(para){
    phi = para[1]; sigw = para[2]; sigv = para[3]
    Sigma0 = (sigw^2)/(1-phi^2); Sigma0[Sigma0<0]=0
    kf = Kfilter0(num, y, 1, mu0=0, Sigma0, phi, sigw, sigv)
    return(kf$like) }

# Estimation
(est = optim(init.par, Linn, gr=NULL, method='BFGS', hessian=TRUE,
        control=list(trace=1, REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
cbind(estimate=c(phi=est$par[1],sigw=est$par[2],sigv=est$par[3]),SE)
```

```
initial  value 81.313627
iter   2 value 80.169051
iter   3 value 79.866131
iter   4 value 79.222846
iter   5 value 79.021504
iter   6 value 79.014723
iter   7 value 79.014453
iter   7 value 79.014452
iter   7 value 79.014452
final  value 79.014452
converged
```

Example 6.6: Newton-Raphson for AR(1) Process Hidden in Observational Noise, IV

```
$par
0.813762322557583 0.850786309703814 0.874396781396708
$value
79.0144524103211
...
$message
NULL
$hessian
253.36290        67.39775         -9.64101
67.39775         78.99067         48.61052
-9.64101         48.61052         92.20472
```

```
SE = sqrt(diag(solve(est$hessian)))
cbind(estimate=c(phi=est$par[1],sigw=est$par[2],sigv=est$par[3]), SE)
```

```
        estimate         SE
phi     0.8137623        0.08060636
sigw    0.8507863        0.17528895
sigv    0.8743968        0.14293192
```

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
- **Data**: $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
- **Data**: $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.
- **Complete** data likelihood

$$\mathbf{P}(X_n, Y_n|\Theta) = \mathbf{P}(x_0|\mu_0, \Sigma_0) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \Phi, Q) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, R).$$

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
- **Data**: $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.
- **Complete** data likelihood

$$\mathbf{P}(X_n, Y_n|\Theta) = \mathbf{P}(x_0|\mu_0, \Sigma_0) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \Phi, Q) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, R).$$

E-Step: **Expected** log-likelihood given **previous parameters**

$$Q(\Theta|\Theta^{(j-1)}) = \mathbb{E}\left[-2\log \mathbf{P}\left(X_n, Y_n|\Theta\right)|\, Y_n, \Theta^{(j-1)}\right]$$

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
- **Data**: $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.
- **Complete** data likelihood

$$\mathbf{P}(X_n, Y_n|\Theta) = \mathbf{P}(x_0|\mu_0, \Sigma_0) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \Phi, Q) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, R).$$

E-Step: **Expected** log-likelihood given **previous parameters**

$$Q(\Theta|\Theta^{(j-1)}) = \mathbb{E}\left[-2\log \mathbf{P}\left(X_n, Y_n|\Theta\right)|\, Y_n, \Theta^{(j-1)}\right]$$

M-Step: **Maximize** expected log-likelihood

$$\Theta^{(j)} = \arg\max_{\tilde{\Theta}} Q\left(\tilde{\Theta}|\Theta^{(j-1)}\right)$$

Estimation of State-Space Models by EM

- **States**: $X_n = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.
- **Data**: $Y_n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$.
- **Complete** data likelihood

$$\mathbf{P}(X_n, Y_n|\Theta) = \mathbf{P}(x_0|\mu_0, \Sigma_0) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \Phi, Q) \cdot \prod_{t=1}^{n} \mathbf{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, R).$$

E-Step: **Expected** log-likelihood given **previous parameters**

$$Q(\Theta|\Theta^{(j-1)}) = \mathbb{E}\left[-2\log \mathbf{P}\left(X_n, Y_n|\Theta\right)|Y_n, \Theta^{(j-1)}\right]$$

M-Step: **Maximize** expected log-likelihood

$$\Theta^{(j)} = \arg\max_{\tilde{\Theta}} Q\left(\tilde{\Theta}|\Theta^{(j-1)}\right)$$

- **EM Algorithm**: **Initialize** $\Theta^{(0)}$; **iterate** over E and M steps until convergence.

Estimation of State-Space Models by EM: E-Step

- Complete-data log-likelihood:

$$-2\log \mathbf{P}(X_n, Y_n|\Theta) = \text{const} + \log\det\Sigma_0 + (\boldsymbol{x}_0 - \mu_0)'\Sigma_0^{-1}(\boldsymbol{x}_0 - \mu_0)$$
$$+ n\log\det(Q) + \sum_{t=1}^{n}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})'Q^{-1}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})$$
$$+ n\log\det(R) + \sum_{t=1}^{n}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)'R^{-1}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t).$$

Estimation of State-Space Models by EM: E-Step

- Complete-data log-likelihood:

$$-2\log \mathbf{P}(X_n, Y_n|\Theta) = \text{const} + \log\det\Sigma_0 + (\boldsymbol{x}_0 - \mu_0)'\Sigma_0^{-1}(\boldsymbol{x}_0 - \mu_0)$$
$$+ n\log\det(Q) + \sum_{t=1}^{n}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})'Q^{-1}(\boldsymbol{x}_t - \Phi\boldsymbol{x}_{t-1})$$
$$+ n\log\det(R) + \sum_{t=1}^{n}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t)'R^{-1}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t).$$

- Expected log-likelihood, given $\Theta^{(j-1)}$:

$$Q(\Theta|\Theta^{(j-1)}) = \log\det\Sigma_0 + \text{Tr}\left\{\Sigma_0^{-1}\left[P_0^n + (\boldsymbol{x}_0^n - \mu_0)(\boldsymbol{x}_0^n - \mu_0)'\right]\right\}$$
$$+ n\log\det(Q) + \text{Tr}\left\{Q^{-1}\left[S_{11} - S_{10}\Phi' - \Phi S_{10}' + \Phi S_{00}\Phi'\right]\right\}$$
$$+ n\log\det(R) + \text{Tr}\left\{R^{-1}\left[\sum_{t=1}^{n}(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^n)(\boldsymbol{y}_t - A_t\boldsymbol{x}_t^n)' + A_tP_t^nA_t'\right]\right\},$$

$$S_{11} = \sum_{t=1}^{n}(\boldsymbol{x}_t^n\boldsymbol{x}_t^{n'} + P_t^n), \qquad S_{10} = \sum_{t=1}^{n}(\boldsymbol{x}_t^n\boldsymbol{x}_t^{n'} + P_{t,t-1}^n), \qquad S_{00} = \sum_{t=1}^{n}(\boldsymbol{x}_{t-1}^n\boldsymbol{x}_{t-1}^{n'} + P_{t-1}^n).$$

Estimation of State-Space Models by EM (cont'd)

- **Smoothed** $\{\boldsymbol{x}_t^n\}_{t=1}^n$ and **uncertainties** $\{P_t^n\}_{t=1}^n$ are calculated under the **current value** of parameters $\Theta^{(j-1)}$.

Estimation of State-Space Models by EM (cont'd)

- **Smoothed** $\{\boldsymbol{x}_t^n\}_{t=1}^n$ and **uncertainties** $\{P_t^n\}_{t=1}^n$ are calculated under the **current value** of parameters $\Theta^{(j-1)}$.
- Overall procedure alternates between:

  E-Step: **Kalman Filtering** and **Smoothing**.

  M-Step: Multivariate normal **maximum likelihood**.

Estimation of State-Space Models by EM: M-Step

M-step updates (multivariate normal **maximum likelihood**):

$$\Phi^{(j)} = S_{10} S_{00}^{-1},$$

$$Q^{(j)} = n^{-1} \left(S_{11} - S_{10} S_{00}^{-1} S_{10}' \right),$$

$$R^{(j)} = n^{-1} \sum_{t=1}^{n} [(\boldsymbol{y}_t - A_t \boldsymbol{x}_t^n)(\boldsymbol{y}_t - A_t \boldsymbol{x}_t^n)' + A_t P_t^n A_t']$$

$$\mu_0^{(j)} = \boldsymbol{x}_0^n, \qquad \Sigma_0^{(j)} = P_0^n.$$

Example 6.8: Hidden AR(1) by EM

- Same data as in Example 6.6.
- Same initial estimates as in Example 6.6.

```
library(nlme) # loads package
# EM procedure
# use the code from Exm. 6.6. to get r, q, phi.
cr = sqrt(r); cq = sqrt(q); mu0 = 0; Sigma0 = 2.8
(em = EM0(num, y, 1, mu0, Sigma0, phi, cq, cr, 75, .00001))

# Standard Errors  (this uses nlme)
phi = em$Phi; cq = chol(em$Q); cr = chol(em$R)
mu0 = em$mu0; Sigma0 = em$Sigma0
para = c(phi, cq, cr)

# Evaluate likelihood at estimates
Linn=function(para){
  kf = Kfilter0(num, y, 1, mu0, Sigma0, para[1], para[2], para[3])
  return(kf$like)
  }
emhess = fdHess(para, function(para) Linn(para))
SE = sqrt(diag(solve(emhess$Hessian)))
```

Example 6.8: Hidden AR(1) by EM

```
# Display summary of estimation
estimate = c(para, em$mu0, em$Sigma0); SE = c(SE,NA,NA)
u = cbind(estimate, SE)
rownames(u) = c("phi","sigw","sigv","mu0","Sigma0")
u
```

EM (59 iterations)

	estimate	SE
phi	0.80975110	0.07850146
sigw	0.85326930	0.16414648
sigv	0.86354667	0.13659399
mu0	-1.96487182	NA
Sigma0	0.02227538	NA

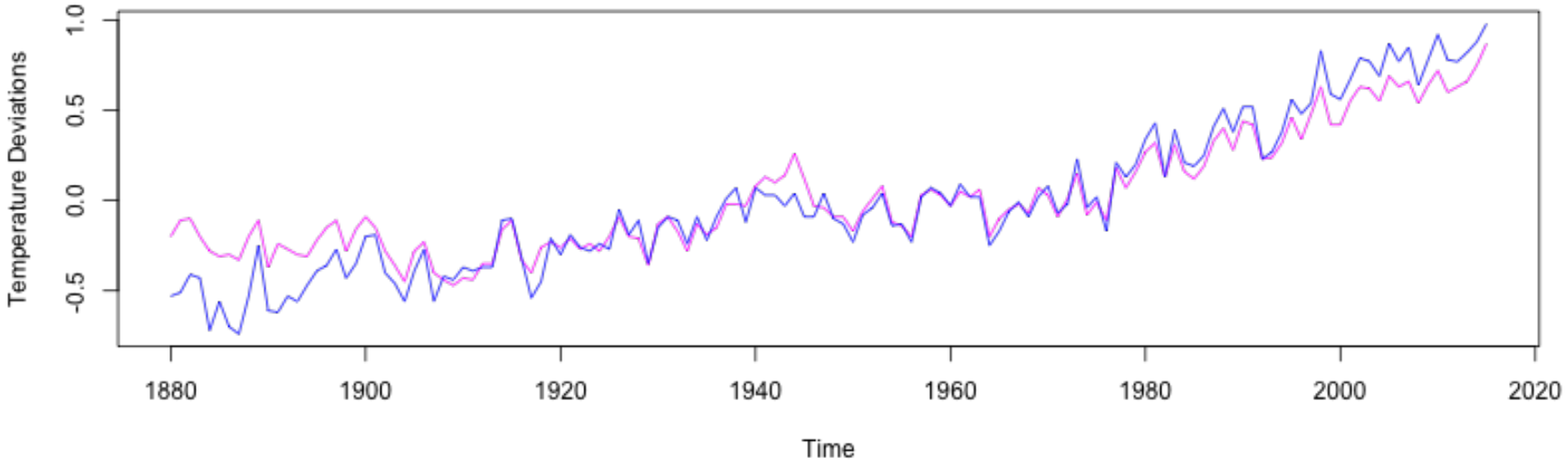
Newton-Raphson (7 iterations)

	estimate	SE
phi	0.8137623	0.08060636
sigw	0.8507863	0.17528895
sigv	0.8743968	0.14293192

Examples

Example 6.2: Global Warming, I (review)

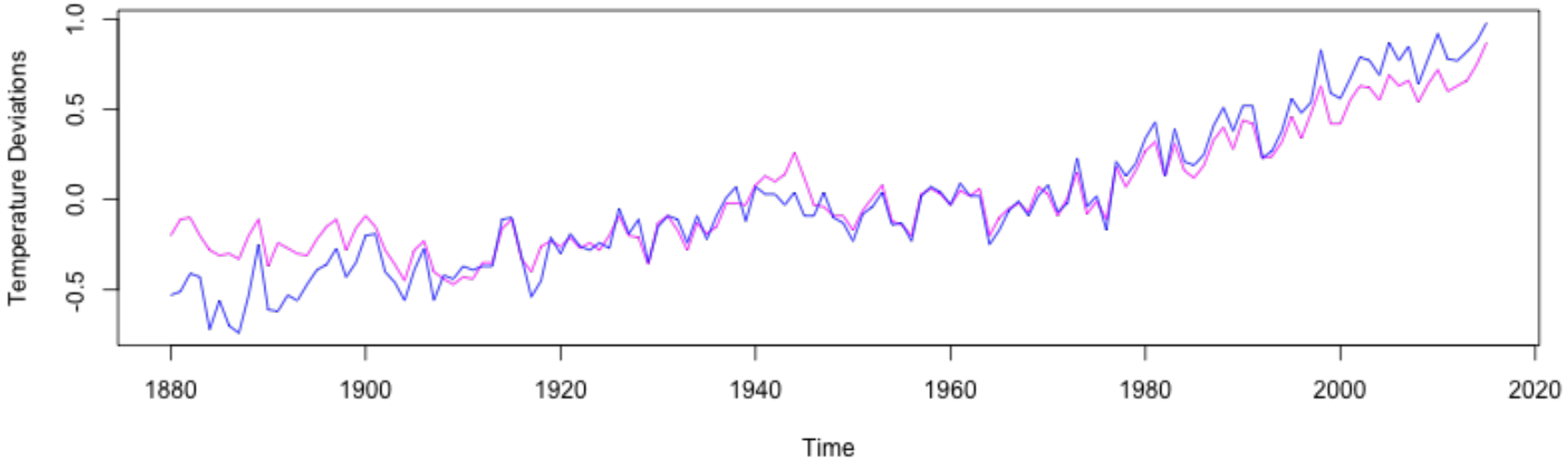
```
ts.plot(globtemp, globtempl, col=c(6,4), ylab='Temperature Deviations')
```



- Two global temperature series:
  1. Land-based.
  2. Marine-based.

Example 6.2: Global Warming, I (review)

```
ts.plot(globtemp, globtempl, col=c(6,4), ylab='Temperature Deviations')
```

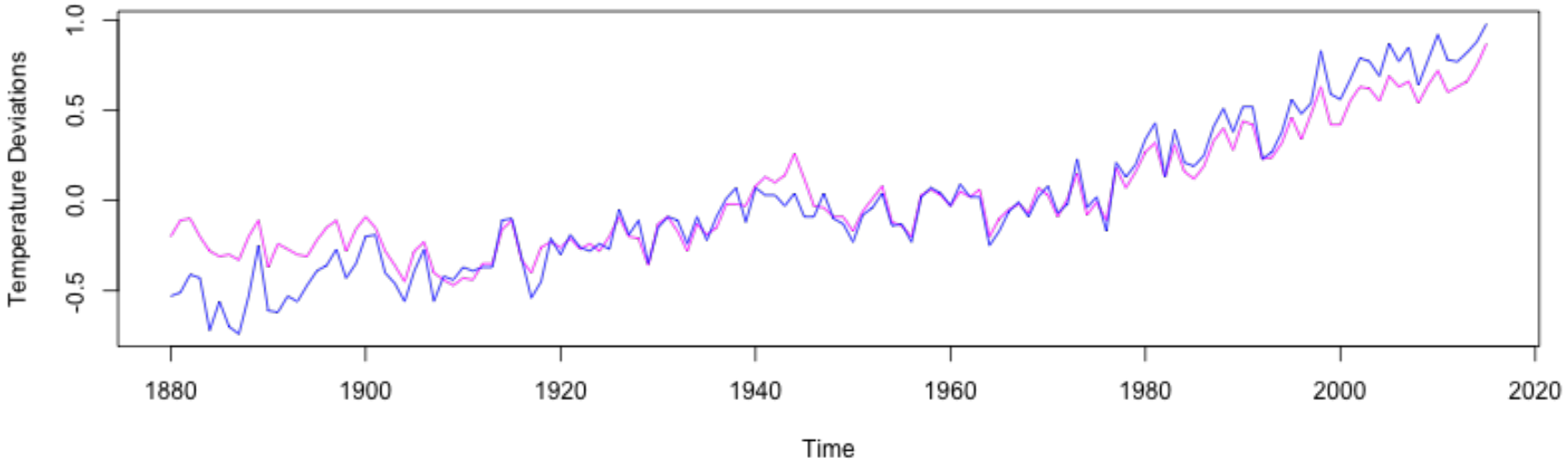


- Two global temperature series:
  1. Land-based.
  2. Marine-based.
- Suppose both series are observing the **same signal** with **different noises**

$$y_{t1} = x_t + v_{t1}, \qquad y_{t2} = x_t + v_{t2}.$$

Example 6.2: Global Warming, I (review)

```
ts.plot(globtemp, globtempl, col=c(6,4), ylab='Temperature Deviations')
```



- Two global temperature series:
  1. Land-based.
  2. Marine-based.
- Suppose both series are observing the **same signal** with **different noises**

$$y_{t1} = x_t + v_{t1}, \qquad y_{t2} = x_t + v_{t2}.$$

- Common trend model: $x_t = x_{t-1} + \delta + w_t$, where $\delta > 0$ is annual rate of warming.

Example 6.2: Global Warming, II (review)

Represent in State-Space form:

- **State Equation**:

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \Upsilon \boldsymbol{u}_t + \boldsymbol{w}_t$$

  - **State Vector**: $\boldsymbol{x}_t = [x_t]$.
  - **Exogenous Signal**: $\boldsymbol{u}_t \equiv [1]$.
  - **State Dynamics**: $\Phi = [1]$, $\Upsilon = [\delta]$
  - $Q = [q_{11}]$.

Example 6.2: Global Warming, II (review)

Represent in State-Space form:

- **State Equation**:

$$\boldsymbol{x}_t = \Phi \boldsymbol{x}_{t-1} + \Upsilon \boldsymbol{u}_t + \boldsymbol{w}_t$$

  - **State Vector**: $\boldsymbol{x}_t = [x_t]$.
  - **Exogenous Signal**: $\boldsymbol{u}_t \equiv [1]$.
  - **State Dynamics**: $\Phi = [1]$, $\Upsilon = [\delta]$
  - $Q = [q_{11}]$.

- **Observation Equation**:

$$\boldsymbol{y}_t = \begin{bmatrix} y_{t1} \\ y_{t2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} v_{t1} \\ v_{t2} \end{bmatrix}, \qquad R = \mathrm{Var}\left(\begin{bmatrix} v_{t1} \\ v_{t2} \end{bmatrix} \right) = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}.$$

Example 6.7: Newton Raphson for Global Temperature Data

- Parameters:

$$\Theta = (q_{11}, r_{11}, r_{22}, r_{12}, \delta, \phi; \mu_0, \Sigma_0)$$

- Initial values:

$$\Theta^{(0)} = (.1, .1, .1, .0, .05, 1; -.35, 1)$$

```
# Setup
y = cbind(globtemp, globtempl)
num = nrow(y)
input = rep(1, num) # u series
A = array(rep(1,2), dim=c(2,1,num))
mu0 = -.35; Sigma0 = 1;  Phi = 1

# Function to Calculate Likelihood
Linn=function(para){
  cQ = para[1]       # sigma_w
  cR1 = para[2]     # 11 element of chol(R)
  cR2 = para[3]     # 22 element of chol(R)
  cR12 = para[4]    # 12 element of chol(R)
  cR = matrix(c(cR1,0,cR12,cR2),2)  # put the matrix together
  drift = para[5]
  kf = Kfilter1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
  return(kf$like)
 }
```

Example 6.7: Newton Raphson for Global Temperature Data (cont'd)

```
# Estimation
init.par = c(.1,.1,.1,0,.05)  # initial values of parameters
(est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE, control=lis
SE = sqrt(diag(solve(est$hessian)))

# Summary of estimation
estimate = est$par;
```

```
initial  value -411.324427
...
iter  20 value -442.972884
iter  20 value -442.972884
final  value -442.972884
converged
$par
0.055 0.074 0.127 0.13 0.0065
...
$hessian
16049.88    15522.86    1531.31     -3056.18    -393.52
15522.86    33596.24    -3152.01    -7422.54    374.22
1531.31     -3152.01    12537.60    4427.73     -409.02
-3056.18    -7422.54    4427.73     3502.81     347.25
-393.52     374.22      -409.02     347.25      44061.82
```

Example 6.7: Newton Raphson for Global Temperature Data (cont'd)

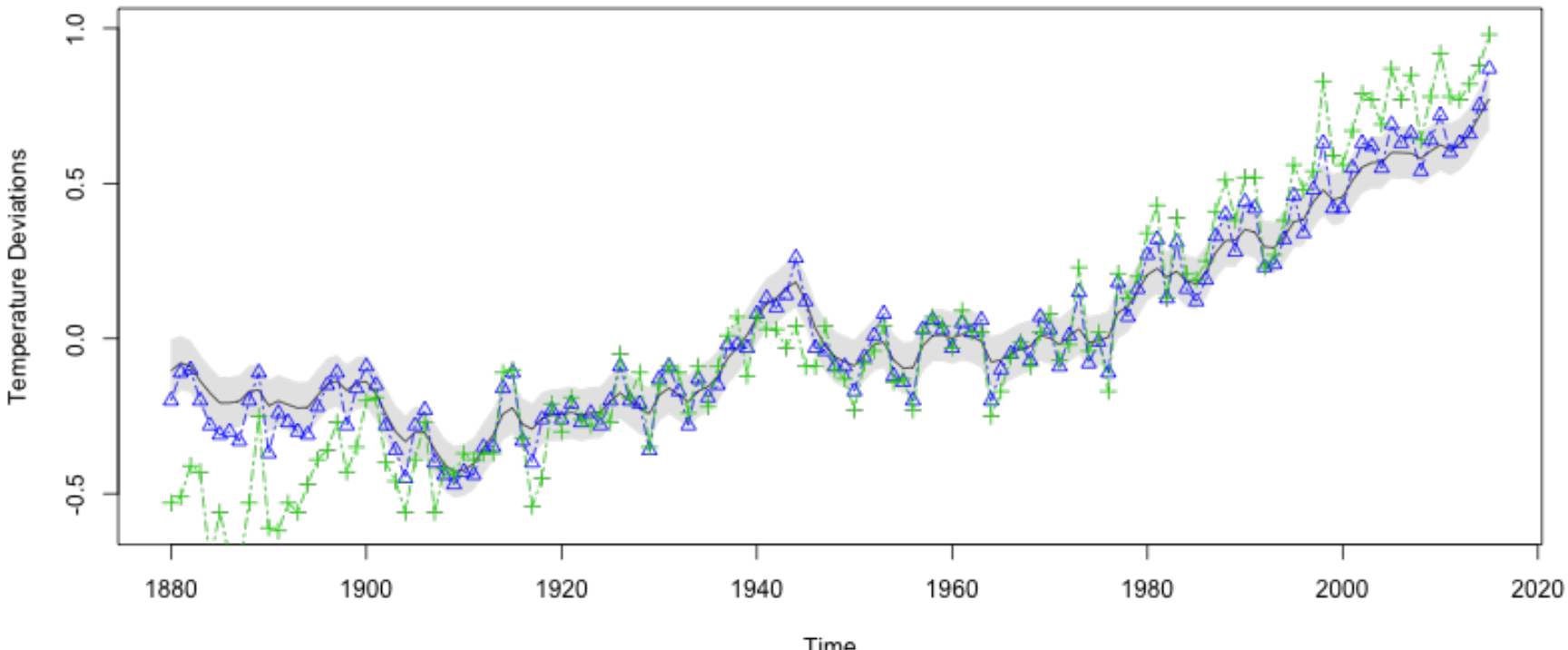
```
# Summary of estimation
SE = sqrt(diag(solve(est$hessian)))
estimate = est$par; u = cbind(estimate, SE)
rownames(u)=c("sigw","cR11", "cR22", "cR12", "drift");
u
```

	estimate	SE
sigw	0.05501124	0.011358852
cR11	0.07418174	0.009856923
cR22	0.12694400	0.015481675
cR12	0.12925308	0.038230357
drift	0.00649545	0.004787053

```
# Smooth (first set parameters to their final estimates)
cQ     = est$par[1]
cR1    = est$par[2]
cR2    = est$par[3]
cR12   = est$par[4]
cR     = matrix(c(cR1,0,cR12,cR2), 2)
(R     = t(cR)%*%cR)     #  to view the estimated R matrix
drift  = est$par[5]   # 0.0065
ks     = Ksmooth1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
```

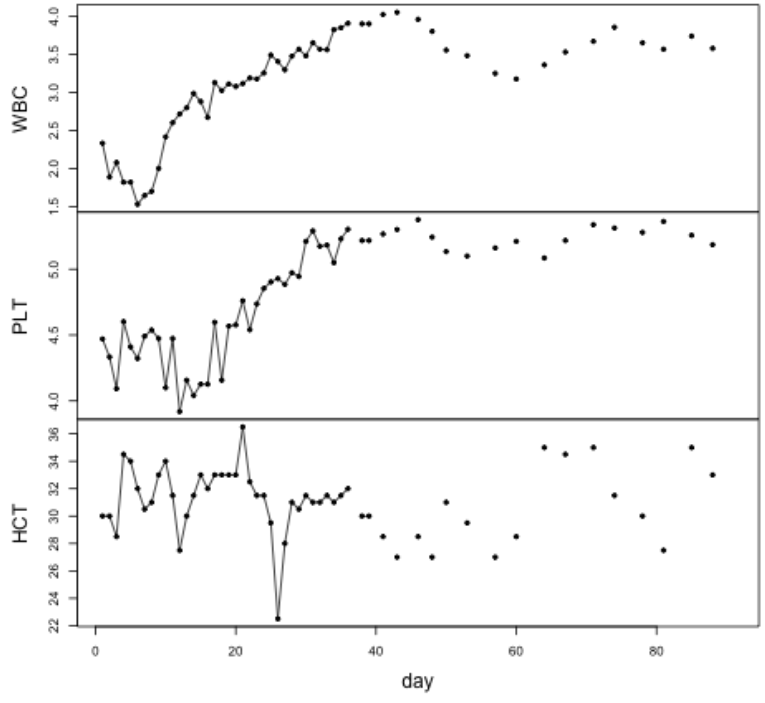
Example 6.7: Newton Raphson for Global Temperature Data (cont'd)

```
# Plot
xsm  = ts(as.vector(ks$xs), start=1880)
rmse = ts(sqrt(as.vector(ks$Ps)), start=1880)
plot(xsm, ylim=c(-.6, 1), ylab='Temperature Deviations')
  xx = c(time(xsm), rev(time(xsm)))
  yy = c(xsm-2*rmse, rev(xsm+2*rmse))
polygon(xx, yy, border=NA, col=gray(.6, alpha=.25))
lines(globtemp, type='o', pch=2, col=4, lty=6)
lines(globtempl, type='o', pch=3, col=3, lty=6)
```



Example 6.1: Biomedical Monitoring (review)

```
plot(blood, type='o', pch=19, xlab='day', main='')
```



- 3 variables:
  1. log(white blood cell count) [WBC];
  2. log(platelet count) [PLT];
  3. Hematocrit level [HCT];
- Feature: **Missing values**.
- We can **model** all 3 variables using the SS approach and **estimate** the missing values.

Example 6.1: Biomedical Monitoring (review)

- State Equation

$$\begin{bmatrix} x_{t1} \\ x_{t2} \\ x_{t3} \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \begin{bmatrix} x_{t-1,1} \\ x_{t-1,2} \\ x_{t-1,3} \end{bmatrix} + \begin{bmatrix} w_{t1} \\ w_{t2} \\ w_{t3} \end{bmatrix}$$

- Observation Equation:

$$\boldsymbol{y}_t = A_t \boldsymbol{x}_t + \boldsymbol{v}_t, \qquad A_t = \begin{cases} I_{3\times 3} & \text{observed,} \\ 0_{3\times 3} & \text{missing.} \end{cases}$$

- Covariances:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}, \qquad R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}.$$

Example 6.9: Longitudinal Biomedical Data, I

Use EM to estimate Φ, Q, and R:

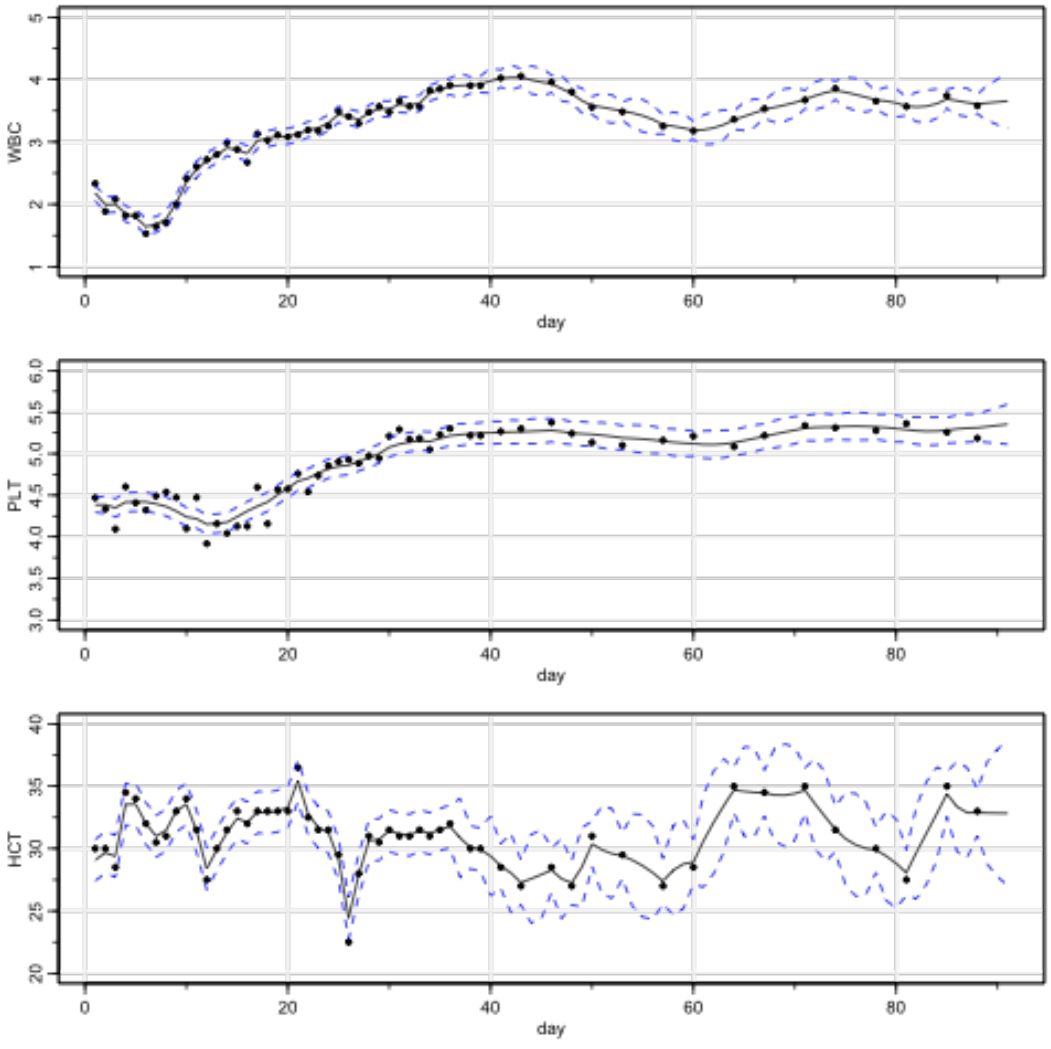
```
y    = cbind(WBC, PLT, HCT); num  = nrow(y)
A    = array(0, dim=c(3,3,num))  # creates num 3x3 zero matrices
for(k in 1:num) if (y[k,1] > 0) A[,,k]= diag(1,3)

# Initial values
mu0 = matrix(0,3,1); Sigma0 = diag(c(.1,.1,1) ,3); Phi    = diag(1,3)
cQ     = diag(c(.1,.1,1), 3); cR     = diag(c(.1,.1,1), 3)
(em = EM1(num, y, A, mu0, Sigma0, Phi, cQ, cR, 100, .001))
```

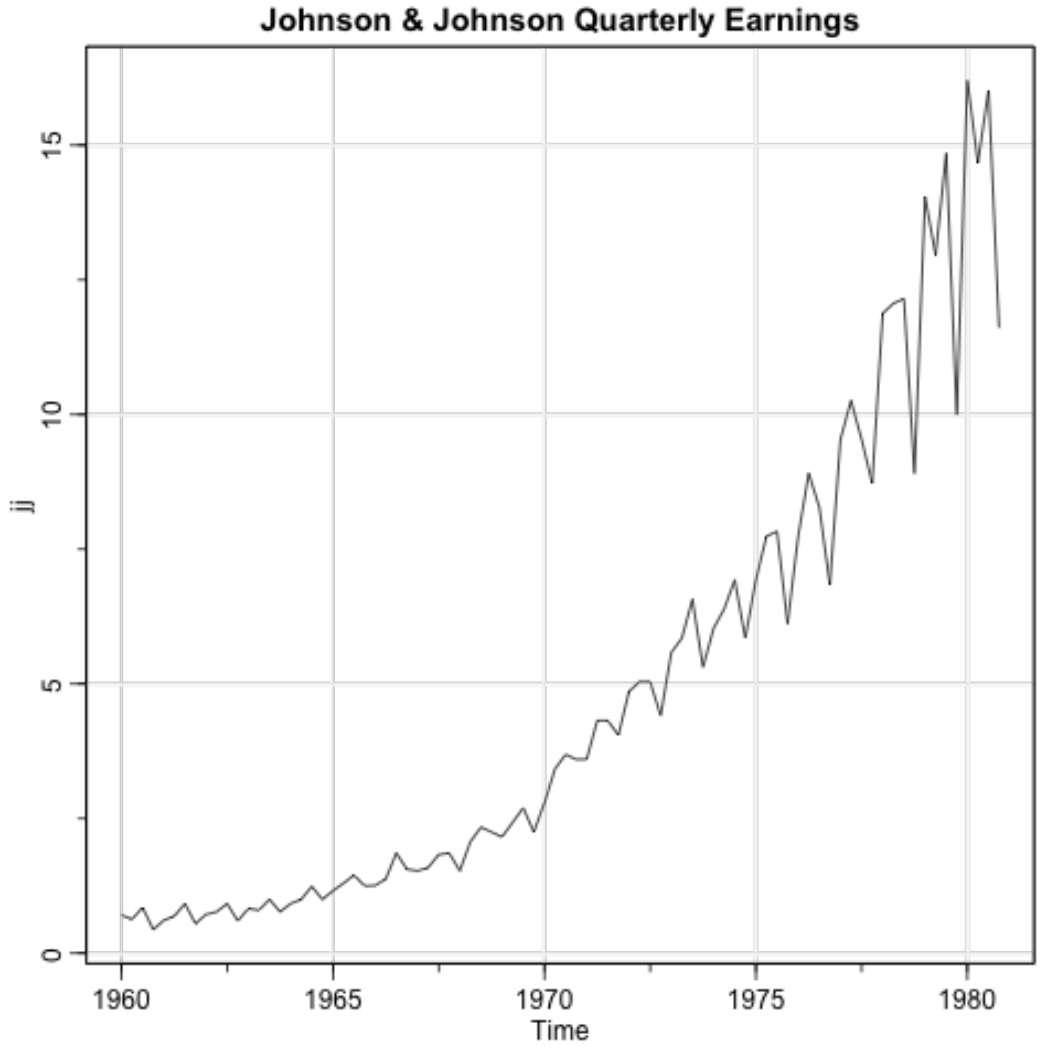
```
$Phi
0.98052698      -0.03494377     0.008287009
0.05279121      0.93299479      0.005464917
-1.46571679     2.25780951      0.795200344
$Q
0.013786772     -0.001724166    0.01882951
-0.001724166    0.003032109     0.03528162
0.018829510     0.035281625     3.61897901
$R
0.007124671     0.0000000       0.0000000
0.000000000     0.0168669       0.0000000
0.000000000     0.0000000       0.9724247
```

$$\hat{x}_{t3} = -1.466x_{t-1,1} + 2.258x_{t-1,2} + .795x_{t-1,3}.$$

Example 6.9: Longitudinal Biomedical Data, II



Example 6.10: Seasonal Decomposition of JnJ Quarterly Earnings



Exponential growth trend + 4 seasonal components

JnJ Seasonal Decomposition, I

- Random **trend** and **seasonal** components in **noise**:

$$y_t = T_t + S_t + v_t,$$

where:

- T_t is **trend** increasing **exponentially**: $T_t = \phi T_{t-1} + w_{t1}$,
- S_t is a **seasonal** component: $S_t + S_{t-1} + S_{t-2} + S_{t-3} = w_{t2}$.

State-space form ($p = 4$, $q = 1$):

- Observation Equation:

$$y_t = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{bmatrix} + v_t$$

- State Dynamics:

$$\begin{bmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{bmatrix} = \begin{bmatrix} \phi & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} T_{t-1} \\ S_{t-1} \\ S_{t-2} \\ S_{t-3} \end{bmatrix} + \begin{bmatrix} w_{t1} \\ w_{t2} \\ 0 \\ 0 \end{bmatrix}$$

- Covariances:

$$Q = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R = [r_{11}].$$

JnJ Seasonal Decomposition, II

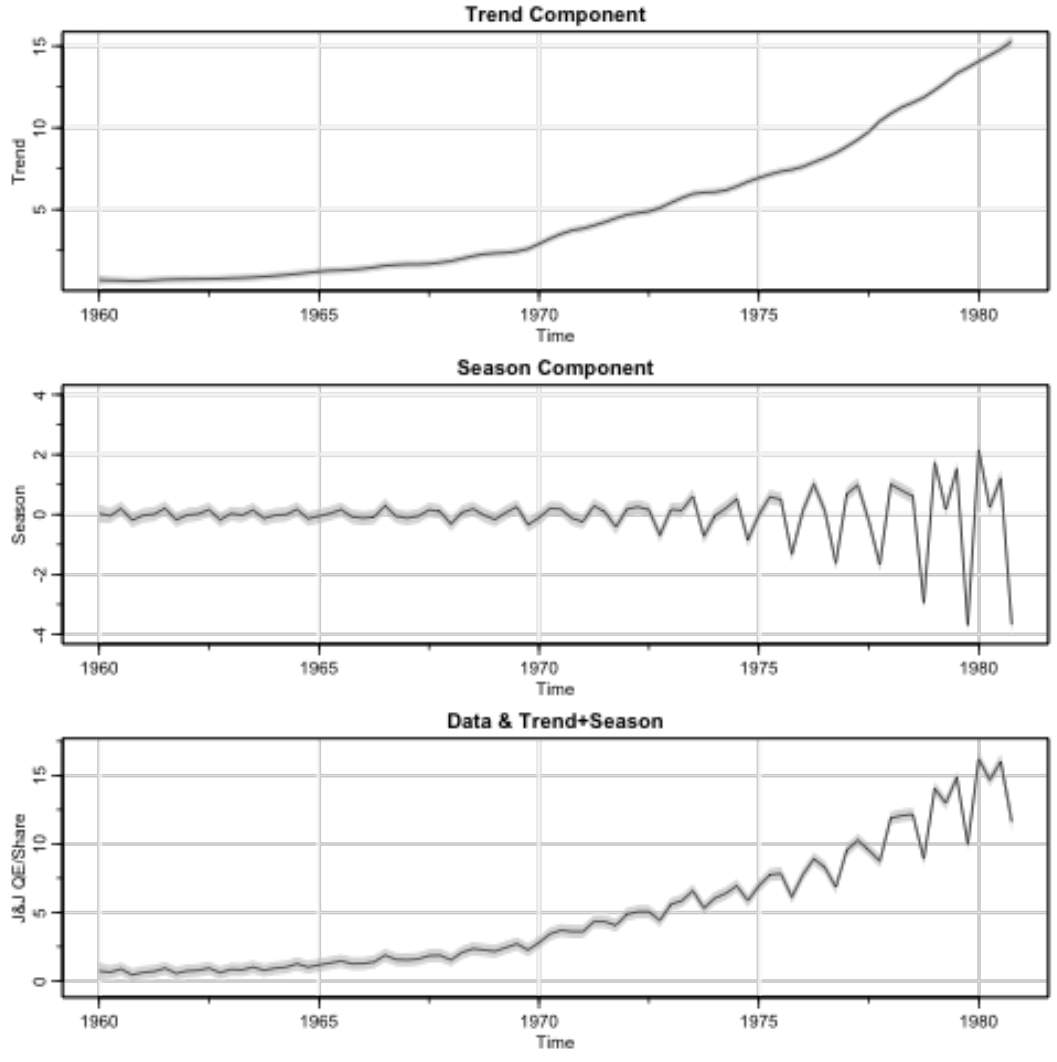
- Parameters to estimate: $\Theta = (\phi, r_{11}, q_{11}, q_{22})$
- Initial conditions:
  - $\phi^{(0)} = 1.03$ (3% annual growth rate)
  - **Initial State mean**: $\mu_0 = (.7, 0, 0, 0)'$
  - **Initial State uncertainty**: $\Sigma_0 = .04 \cdot I_{4\times 4}$
  - **State covarinace**: $q_{11} = q_{22} = .01$
  - **Measurement error covariance**: $r_{11} = .25$

```
# Initial Parameters
mu0       = c(.7,0,0,0)
Sigma0    = diag(.04,4)
init.par = c(1.03, .1, .1, .5)  # Phi[1,1], the 2 Qs and R

# Function to Calculate Likelihood
Linn=function(para){
 ...}

# Estimation
est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
                  control=list(trace=1,REPORT=1))
SE  = sqrt(diag(solve(est$hessian)))
```

JnJ Seasonal Decomposition, III



JnJ Seasonal Decomposition, IV

Prediction $\pm$2SE: