```
## Loading required package:  Rcpp
## Loading required package:  inline
## Loading required package:  methods
##
## Attaching package:  'inline'
##
## The following object is masked from 'package:Rcpp':
##
##    registerPlugin
##
## rstan (Version 2.2.0, packaged:  2014-05-13 20:40:04 UTC)
```

1 Model Terminology

These models are called either state space models (econometrics) or dynamic (linear) models (statistics / Bayesian).

The following defines a *state space model*

$$
\begin{aligned}
y_t &= f(\theta_t | b_t, F_t, \nu_t) \\
\theta_t &= f(\theta_{t-1} | g_t, G_t, \nu_t)
\end{aligned}
$$

If θ_t is continuous then it is a *continuous state space model*, if θ_t is discrete then it is a *discrete state space model*.

If those equations can be written as

$$
\begin{aligned}
y_t &= b_t + F_t \theta_t + \nu_t \\
\theta_t &= g_t + G_t \theta_{t-1} + \omega_t
\end{aligned}
$$

then the model is a *Dynamic Linear Model (DLM)* (linear SSM), otherwise it is a non-linear dynamic model. If ν_t and ω_t are normal distributions, then it is *Guassian* or *Normal Dynamic Linear Model* (GDLM or NDLM).

A dynamic linear model is defined by the following set of equations,

$$
\begin{aligned}
y_t &= b_t + F_t \theta_{t-1} + \nu_t & \nu_t &\sim N(0, V_t) & (1) \\
\theta_t &= g_t + G_t \theta_{t-1} + \omega_t & \omega_t &\sim N(0, W_t) & (2) \\
\theta_0 &\sim N(m_0, C_0) & & & (3)
\end{aligned}
$$

where equation 1 is the observation or measurement equation, equation 2 is the system equation, and equation 3 is the initial information. The number of variables is r and the number of states is p.

1.1 Filtering Equations

See Petris, Petrone, and Campagnoli [2, Chapter 2.7, p. 53] and West and Harrison [4, Chapter 4] for proofs.

matrix	dimensions
F_t	$r \times p$
G_t	$p \times p$
V_t	$r \times r$
W_t	$p \times p$
C_0	$p \times p$

vector	dimensions
Y_t	r
θ_t	p
b_t	r
ν_t	r
g_t	p
ω_t	p
m_0	p

variable	dim
a_t	p
R_t	p, p
f_t	r
Q_t	r, r
m_t	p
C_t	p, p
e_t	r
K_t	p, r

Assume the posterior distribution at $t-1$. Let $\theta_{t-1}|y_{1:t-1} \sim N(m_{t-1}, C_{t-1})$. The one step ahead predictive distribution of θ_t given $y_{1:t-1}$ is $N(a_t, R_t)$,

$$a_t = \mathrm{E}(\theta_t|y_{1:t-1}) \qquad = g_t + G_t m_{t-1} \tag{4}$$
$$R_t = \mathrm{Var}(\theta_t|y_{1:t-1}) \qquad = G_t C_{t-1} G_t' + W_t \tag{5}$$

The one step ahead predictive distribution of Y_t given $y_{1:t-1}$ is $N(f_t, Q_t)$,

$$f_t = \mathrm{E}(Y_t|y_{1:t-1}) \qquad = b_t + F_t a_t \tag{6}$$
$$Q_t = \mathrm{Var}(Y_t|y_{1:t-1}) \qquad = F_t R_t F_t' + V_t \tag{7}$$

The filtered distribution of θ_t given $y_{1:t}$ is $N(m_t, C_t)$

$$m_t = \mathrm{E}(\theta_t|y_{1:t}) \qquad = a_t + R_t F_t' Q_t^{-1} e_t \tag{8}$$
$$= a_t + K_t e_t \tag{9}$$
$$C_t = \mathrm{Var}(\theta_t|y_{1:t}) \qquad = R_t - R_t F_t' Q_t^{-1} F_t R_t \tag{10}$$
$$= R_t - K_t Q_t K_t' \tag{11}$$
$$= (I_p - K_t F_t) R_t (I_p - K_t F_t)' + K_t V_t K_t' \tag{12}$$
$$e_t \qquad = y_t - f_t \tag{13}$$
$$K_t \qquad = R_t F_t' Q_t^{-1} \tag{14}$$

where e_t is the one step forecast error and K_t is the Kalman gain (adaptative coefficient). Equation 12 is the "Joseph stablized form" [3, p. 3].

A few key identies [4, pp. 106-107] are,

$$K_t = R_t F_t' Q_t^{-1} = C_t F_t' V_t^{-1} \tag{15}$$
$$C_t = R_t^{-1} - K_t Q_t K_t = R_t (I_p - F_t K_t') \tag{16}$$
$$C_t^{-1} = R_t^{-1} + F_t' V_t^{-1} F_t \tag{17}$$
$$Q_t = (I_r - F_t K_t)^{-1} V_t \tag{18}$$
$$F_t K_t = I_r - V_t Q_t^{-1} \tag{19}$$

1.2 Missing Values

Missing values are equivalent to setting $F_t = 0$ or $V_t = \infty$.

If all values in t are missing, replace the filter steps (equations 8 and 10) with,

$$m_t = a_t \tag{20}$$

$$C_t = R_t \tag{21}$$

Suppose some, but not all variables in time t are missing. Let $r_t \in [0, r]$ is the number of non-missing values in each time period, $r_t = \sum_{j=1}^{r}(y_{j,t} \neq \emptyset)$. If some are missing (let $r > r_t > 0$ be observed), let M be a $r_t \times r$ selection matrix and define,

$$y_t^* = M_t y_t \tag{22}$$

$$F_t^* = M_t F_t \tag{23}$$

$$V_t^* = M_t V_t M_t' \tag{24}$$

and replace y_t, F_t and V_t in the filtering equations with those. Note that the smoothing (section 1.4) and backward sampling (section **??**) algorithms do not depend on F, V and y and thus do not need to be altered when there are missing values.

1.3 Likelihood

If there are no missing values, then the log likelihood is,

$$L(y_{1:n}) = -\frac{nr}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\left(\log|Q_t| + e_t' Q_t^{-1} e_t\right) \tag{25}$$

If there are missing values,

$$L(y_{1:n}) = -\frac{1}{2}\sum_{t=1}^{n}(r_t > 0)\left(r_t \log(2\pi) + \log|Q_t| + e_t' Q_t^{-1} e_t\right) \tag{26}$$

See Koopman, Shephard, and Doornik [1, Chapter 5, p. 57].

1.4 Smoothing

If $\theta_{t+1}|y_{1:n} \sim N(s_{t+1}, S_{t+1})$, then $\theta_t|y_{1:n} \sim N(s_t, S_t)$ where

$$s_t = \mathrm{E}(\theta_t|y_{1:n}) \qquad = m_t + C_t G_{t+1}' R_{t+1}^{-1}(s_{t+1} - a_{t+1}) \tag{27}$$

$$S_t = \mathrm{Var}(\theta_t|y_{1:n}) \qquad = C_t - C_t G_{t+1}' R_{t+1}^{-1}(R_{t+1} - S_{t+1}) R_{t+1}^{-1} G_{t+1} C_t \tag{28}$$

Note that $s_n = m_n$ and $S_n = C_n$. See Petris, Petrone, and Campagnoli [2, Prop 2.4, p. 61] for a proof.

1.5 Backward Sampling

Supposing that $m_{1:n}$, $C_{1:n}$, $a_{1:n}$ and $R_{1:n}$ have been calculated by the filter,[1] To draw $\theta_{1:n}|y_{1:n}$,

1. From the Kalman filter, $\theta_n|y_{1:n} \sim N(m_n, C_n)$
2. For $t = (n-1), \dots, 0$, $\theta_t|y_{1:n} \sim N(h_t, H_t)$ where

$$h_t = m_t + C_t G'_{t+1} R^{-1}_{t+1} (\theta_{t+1} - a_{t+1}) \tag{29}$$
$$H_t = C_t - C_t G'_t R^{-1}_{t+1} G_{t+1} C_t \tag{30}$$

See Petris, Petrone, and Campagnoli [2, Chapter 4.4.1, p. 161] for more details.

2 Sequential Estimation

First, consider the case in which all V_t are diagonal. The vector series $y_1, \dots, y_n$ is treated as a scalar series

$$y_{1,1}, \dots, y_{1,r}, y_{2,1}, \dots, y_{n,r} \tag{31}$$

The prior distribution of the state is $\theta_t|y_{1:(t-1)} \sim N(a_t, R_t)$,

$$a_t = \mathrm{E}(\theta_t|y_{1:t-1}) \qquad = g_t + G_t m_{t-1} \tag{32}$$
$$R_t = \mathrm{Var}(\theta_t|y_{1:t-1}) \qquad = G_t C_{t-1} G'_t + W_t \tag{33}$$

For each variable, $i = 1, \dots, r$. The prediction equation is $y_{t,i}|y_{1:t-1}, y_{t,j|j<i} \sim N(f_{t,i}, Q_{t,i})$. Note that $f_{t,i}$ and $Q_{t,i}$ are scalars.

$$f_{t,i} = \mathrm{E}(Y_{t,i}|y_{1:t-1}, y_{t,j|j<i}) \qquad = b_{t,i} + F_{t,i} m_{t,i-1} \tag{34}$$
$$Q_{t,i} = \mathrm{Var}(Y_{t,i}|y_{1:t-1}, y_{t,j|j<i}) \qquad = F_{t,i} C_{t,i-1} F'_{t,i} + v_{t,i} \tag{35}$$

For the first variable, let $m_{t,0} = a_t$ and $C_{t,0} = R_t$. The filtered distribution of the latent state is $\theta_t|y_{1:t-1}, y_{t,j|j<i} \sim N(m_{t,i}, C_{t,i})$,

$$m_{t,i} = \mathrm{E}(\theta_t|y_{1:t-1}, y_{t,j|j\leq i}) \qquad = m_{t,i-1} + C_{t,i-1} F'_{t,i} Q^{-1}_{t,i} e_{t,i} \tag{36}$$
$$= m_{t,i-1} + K_{t,i} e_{t,i} \tag{37}$$
$$C_{t,i} = \mathrm{Var}(\theta_t|y_{1:t-1}, y_{t,j|j\leq i}) \qquad = C_{t,i-1} - C_{t,i-1} F'_{t,i} Q^{-1}_{t,i} F_{t,i} C_{t,i-1} \tag{38}$$
$$= Q_{t,i} - K_{t,i} Q_{t,i} K'_{t,i} \tag{39}$$
$$= (I_n - K_{t,i} F_{t,i}) C_{t,i} (I_n - K_{t,i} F_{t,i})' + K_{t,i} v_{t,i} K'_{t,i} \tag{40}$$
$$e_{t,i} \qquad = y_{t,i} - f_{t,i} \tag{41}$$
$$K_{t,i} \qquad = C_{t,i-i} F'_{t,i} Q^{-1}_{t,i} \tag{42}$$

[1] No additional adjustment for intercepts is required because they are already incorporated in a_t

Define $m_t = m_{t,r}$ and $C_t = C_{t,r}$. If $y_{t,i}$ is missing, then

$$m_{t,i} = m_{t,i-1} \tag{43}$$
$$C_{t,i} = C_{t,i-1} \tag{44}$$

The likelihood in the sequential case is

$$L(y_{1:n}) = -\frac{1}{2}\sum_{t=1}^{n}\sum_{i=1}^{r}(y_{i,j} \neq \emptyset)\left(\log(2\pi) + \log|Q_{t,i}| + \frac{e_{t,i}^2}{Q_{t,i}}\right) \tag{45}$$

If V_t is not diagonal, then diagonalize it with the Cholesky decomposition of V_t, such that $V_t = L_t D_t L_t'$ where D_t is diagonal, and L_t is lower triangular.

$$y_t^* = L_t^{-1} y_t \tag{46}$$
$$F_t^* = L_t^{-1} F_t \tag{47}$$
$$b_t^* = L_t^{-1} b_t \tag{48}$$
$$\epsilon^* = L_t^{-1}\epsilon_t \sim N(0, D_t) \tag{49}$$

2.1 Univariate Local Level Model

For the univariate local level model, with $r = n = 1$, $F_t = G_t = 1$, and $g_t = b_t = 0$, the calculations can be simplified.

One step ahead predictive distribution of θ_{t-1} given $y_{1:t-1}$ is $N(a_t, R_t)$

$$a_t = E(\theta_t|y_{1:t-1}) = m_{t-1}$$
$$R_t = Var(\theta_t|y_{1:t-1}) = C_{t-1} + W_t$$

The One step ahead predictive distribution of Y_t given $y_{1:t-1}$ is $N(f_t, Q_t)$

$$f_t = E(Y_t|y_{1:t-1}) = a_t = m_{t-1}$$
$$Q_t = Var(Y_t|y_{1:t-1}) = R_t + V_t = C_{t-1} + W_t + V_t$$

The posterior distribution of θ_t given $y_{1:t}$ is $N(m_t, C_t)$

$$f_t = E(\theta_t|y_{1:t}) = a_t + R_t Q_t^{-1} e_t = m_{t-1} + \frac{C_{t-1} + W_t}{C_{t-1} + W_t + V_t} e_t$$
$$Q_t = Var(\theta_t|y_{1:t}) = R_t - \frac{(C_{t-1} + W_t)^2}{C_{t-1} + W_t + V_t}$$

where $e_t = Y_t - f_t$. The Kalman gain is defined as $K_t = R_t/Q_t$.

3 Notes

- Use discounting
  - Use a beta rectangular distribution
  - Beta distribution with priors over α and β

4 Examples

4.1 Nile

The data is included with R as `datasets::Nile`.

1. Standard
2. With missing
3. With intervention

$$
\begin{aligned}
y_t &\sim N(\theta_t, V) && (50) \\
\theta_t &\sim N(\theta_{t-1}, W) && (51)
\end{aligned}
$$

where $r = p = 1$, $F = G = 1$, and $b = g = 0$.

The stan model is as follows

```
str_c(readLines("stan/kalman_batch.stan"), sep = "\n")
```

```
 [1] "data {"
 [2] "  // dimensions"
 [3] "  int n; // number of observations"
 [4] "  int r; // number of variables"
 [5] "  int p; // number of states"
 [6] "  // observations"
 [7] "  vector[r] y[n];"
 [8] "  // system matrices"
 [9] "  // observation equation"
[10] "  matrix[r, p] F;"
[11] "  real b;"
[12] "  // system equation"
[13] "  matrix[p, p] G;"
[14] "  real g;"
[15] "  // initial conditions"
[16] "  vector[p] m0;"
[17] "  cov_matrix[p] C0;"
[18] "}"
[19] "transformed data {"
[20] "  matrix[p, p] Ip;"
[21] "  {"
[22] "    vector[p] Ip_vector;"
[23] "    Ip_vector <- rep_vector(1, p);"
[24] "    Ip <- diag_matrix(Ip_vector);"
[25] "  }"
```

```
[26] "}"
[27] "parameters {"
[28] "  cov_matrix[r] V;"
[29] "  cov_matrix[p] W;"
[30] "}"
[31] "transformed parameters {"
[32] "  // log-likelihood"
[33] "  real loglik_obs[n];"
[34] "  real loglik;"
[35] "  // prior of state: p(theta_t | y_t, ..., y_{t-1})"
[36] "  vector[r] a[n];"
[37] "  matrix[r, r] R[n];"
[38] "  // likelihood of obs: p(y_t | y_t, ..., y_t-1)"
[39] "  vector[r]  f[n];"
[40] "  matrix[r, r] Q[n];"
[41] "  // posterior of states: p(theta_t | y_t, ..., y_t)"
[42] "  vector[p] m[n + 1];"
[43] "  matrix[p, p] C[n + 1];"
[44] ""
[45] "  {"
[46] "    vector[r] err;"
[47] "    matrix[p, r] K;"
[48] "    matrix[r, r] Qinv;"
[49] "    matrix[p, p] C_tmp;"
[50] "    matrix[p, p] J;"
[51] "    "
[52] "    // set initial states"
[53] "    m[1] <- m0;"
[54] "    C[1] <- C0;"
[55] "    // loop through observations"
[56] "    for (t in 1:n) {"
[57] "      // one step ahead predictive distribion of \\theta_t | y_{1:(t-1)}"
[58] "      a[t] <- g + G * m[t];"
[59] "      // R[t] <- G * C[t] * G ' + W;"
[60] "      R[t] <- quad_form(C[t], G ') + W;"
[61] "      // one step ahead predictive distribion of y_t | y_{1:(t-1)}"
[62] "      f[t] <- b + F * a[t];"
[63] "      // Q[t] <- F * R[t] * F ' + V;"
[64] "      Q[t] <- quad_form(R[t], F ') + V;      "
[65] "      Qinv <- inverse(Q[t]);"
[66] "      // error"
[67] "      err <- y[t] - f[t];"
[68] "      // Kalman gain"
[69] "      K <- R[t] * F ' * Qinv;"
[70] "      // posterior distribution of \\theta_t | y_{1:t}"
```

```
[71] "      m[t + 1] <- a[t] + K * err;"
[72] "      // matrix used in Joseph stabilized form"
[73] "      // C_tmp <- R[t] - K * Q[t] * K ';"
[74] "      // C_tmp <- R[t] - quad_form(Q[t], K ');"
[75] "      // C[t + 1] <- 0.5 * (C_tmp + C_tmp ');"
[76] "      J <- (Ip - K * F);"
[77] "      C[t + 1] <- quad_form(R[t], J ') + quad_form(V, K ');"
[78] "      // log likelihood"
[79] "      // loglik_obs[t] <- - 0.5 * (r * log(2 * pi())"
[80] "      // \t\t\t\t+ log_determinant(Q[t])"
[81] "      // \t\t\t\t+ err ' * Qinv * err);"
[82] "      loglik_obs[t] <- - 0.5 * (r * log(2 * pi())"
[83] "      \t\t\t\t+ log_determinant(Q[t])"
[84] "      \t\t\t\t+ quad_form(Qinv, err));"
[85] ""
[86] "    }"
[87] "  }"
[88] "  loglik <- sum(loglik_obs);"
[89] "}"
[90] "model {"
[91] "  increment_log_prob(loglik);"
[92] "}"
```

```
data("Nile")
y <- as.numeric(Nile)

standata <-
    within(list(), {
        y <- matrix(y)
        n <- length(y)
        r <- 1L
        p <- 1L
        b <- 0
        g <- 0
        F <- matrix(1, 1, 1)
        G <- matrix(1, 1, 1)
        m0 <- array(0, 1)
        C0 <- matrix(1000, 1, 1)
        })

m <- stan_model("stan/kalman_batch.stan")

##
## TRANSLATING MODEL 'kalman_batch' FROM Stan CODE TO C++ CODE NOW.
## COMPILING THE C++ CODE FOR MODEL 'kalman_batch' NOW.
```

```
ret <- sampling(m, data = standata)
```

```
## SAMPLING FOR MODEL 'kalman_batch' NOW (CHAIN 1).
##
Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 2000 / 2000 [100%]  (Sampling)
## Elapsed Time: 10.0398 seconds (Warm-up)
##               10.2988 seconds (Sampling)
##               20.3385 seconds (Total)
##
## SAMPLING FOR MODEL 'kalman_batch' NOW (CHAIN 2).
##
Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 2000 / 2000 [100%]  (Sampling)
## Elapsed Time: 9.94662 seconds (Warm-up)
##               8.2413 seconds (Sampling)
##               18.1879 seconds (Total)
##
## SAMPLING FOR MODEL 'kalman_batch' NOW (CHAIN 3).
##
Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 2000 / 2000 [100%]  (Sampling)
## Elapsed Time: 9.67374 seconds (Warm-up)
##                9.98924 seconds (Sampling)
##                19.663 seconds (Total)
##
## SAMPLING FOR MODEL 'kalman_batch' NOW (CHAIN 4).
##
Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 2000 / 2000 [100%]  (Sampling)
## Elapsed Time: 9.82326 seconds (Warm-up)
##                9.10817 seconds (Sampling)
##                18.9314 seconds (Total)
```

```
apply(extract(ret, "V")[[1]], 2, mean)
```

```
## [1] 6601
```

```
apply(extract(ret, "W")[[1]], 2, mean)
```

```
## [1] 29954
```

```
mean(extract(ret, "loglik")[[1]])
```

```
## [1] -671.7
```

4.2 UK Gas

This is a model of quarterly consumption of gas in the UK from 1960 to 1986. This data is included with R as `datasets::UKGas`. The model to be estimated is a local linear trend model and a quarterly seasonal factor model.

$$
\begin{aligned}
F &= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} & G &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
V &= \sigma_y^2 & W &= \operatorname{diag}(0, \sigma_\beta^2, \sigma_s^2, 0, 0)
\end{aligned}
$$

4.3 Industrial Production

A linear trend model of U.S. industrial production of consumption goods, seasonally adjusted. The data is IPCONGD; also available on Quandl.

$$
\begin{aligned}
F &= \begin{bmatrix} 1 & 0 \end{bmatrix} & G &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\
V &= \sigma_y^2 & W &= \operatorname{diag}(\sigma_1^2, \sigma_2^2)
\end{aligned}
$$

4.4 CAPM

The state space are the betas of the stocks,

$$
\theta_t = (\beta_{1,t}, \dots, \beta_{m,t})' \tag{52}
$$

The system equations are,

$$
\begin{aligned}
F_t &= x_t I & G_t &= I \\
V_t &= \Sigma_\epsilon & W_t &= \Sigma_\beta
\end{aligned} \tag{53}
$$

References

[1] Siem Jan Koopman, Neil Shephard, and Jurgen A. Doornik. *Statistical Algorithms for Models in State Space Form: SsfPack 3.0*. Timberlake Consultants Press, 2008. ISBN: 9780955707636. URL: `http://books.google.com/books?id=hLyfQAAACAAJ`.

[2] G. Petris, S. Petrone, and P. Campagnoli. *Dynamic Linear Models with R*. Use R! Springer, 2009. ISBN: 9780387772370. URL: `http://books.google.com/books?id=VCt3zVq8TO8C`.

[3] Fernando Tusell. “Kalman Filtering in R”. In: *Journal of Statistical Software* 39.2 (Mar. 1, 2011), pp. 1–27. ISSN: 1548-7660. URL: `http://www.jstatsoft.org/v39/i02`.

[4] M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer series in statistics. Springer, 1997. ISBN: 9780387947259. URL: `http://books.google.com/books?id=jcl8lD75fkYC`.