State-space models

Robin Aldridge-Sutton

Definition

A state-space model can be made up of:

- An unobserved state variable x_t, taking values in some state-space
- An output variable y_t
- An observation equation $P(y_t|x_t)$
- A state transition equation $P(x_{t+1}|x_t)$
- Where x_t and y_t satisfy the Markov equations:

$$P(y_{t+1}, x_{t+1}|y_t, x_t, ..., y_1, x_1) = P(y_{t+1}, x_{t+1}|y_t, x_t)$$
$$P(y_{t+1}|x_{t+1}, y_t, x_t) = P(y_{t+1}|x_{t+1})$$

- Which implies that $P(y_{t+1}, x_{t+1}|y_t, x_t) = P(y_{t+1}|x_{t+1})P(x_{t+1}|x_t)$

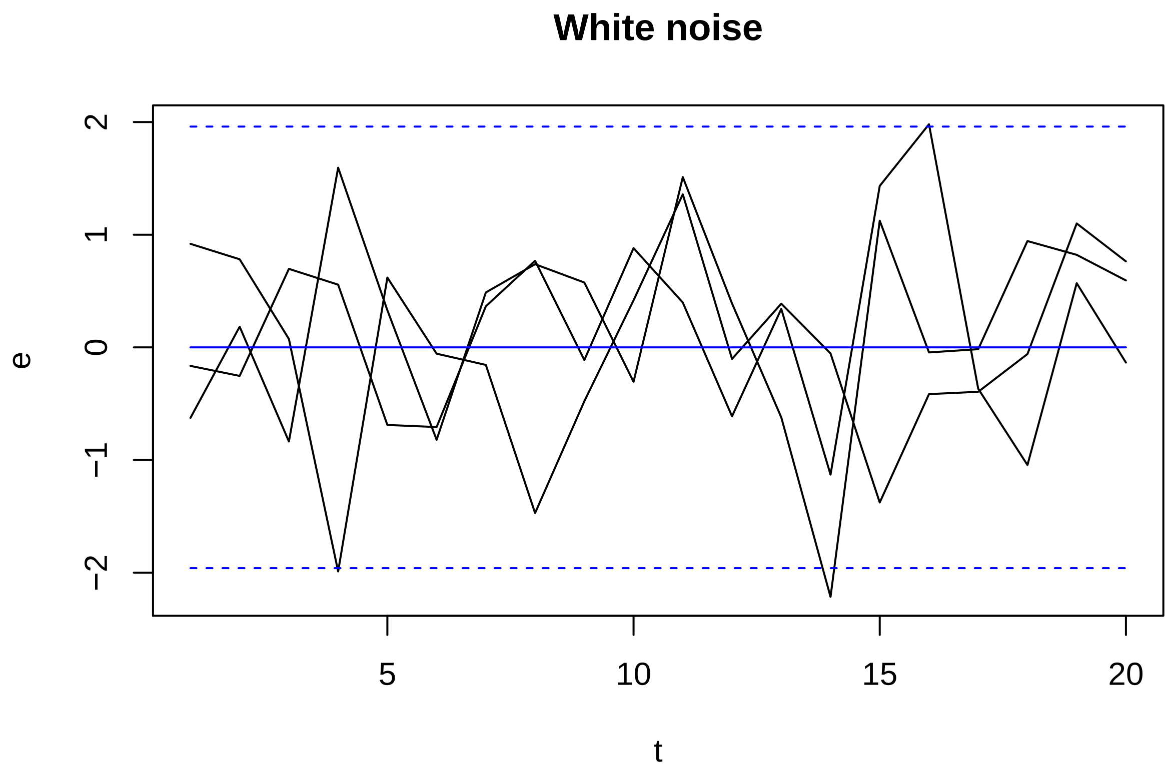
Obtaining the conditional distribution $P(x_t|y_t, ..., y_1)$, known as the smoothing density, from which $P(x_{t+1})$ and $P(y_{t+1})$ can be derived/sampled, is known as filtering.

White noise

White noise is the core of a stochastic time series model.

$$e_t \sim N(0, \sigma^2)$$

```
# Set simulation parameters
set.seed(1)
n_t <- 20
t <- 1:n_t
n_reals <- 3
n_samps <- n_t * n_reals
quants <- t(matrix(c(0, 1.96, -1.96), nrow = 3, ncol = n_t))
q_lty <- c(1, 2, 2)

# Simulate white noise
e <- matrix(rnorm(n_samps), nrow = n_t)

# Plot it
matplot(t, e, type = 'l', main = "White noise", col = 1, lty = 1)
matlines(t, quants, col = 'blue', lty = q_lty)
```

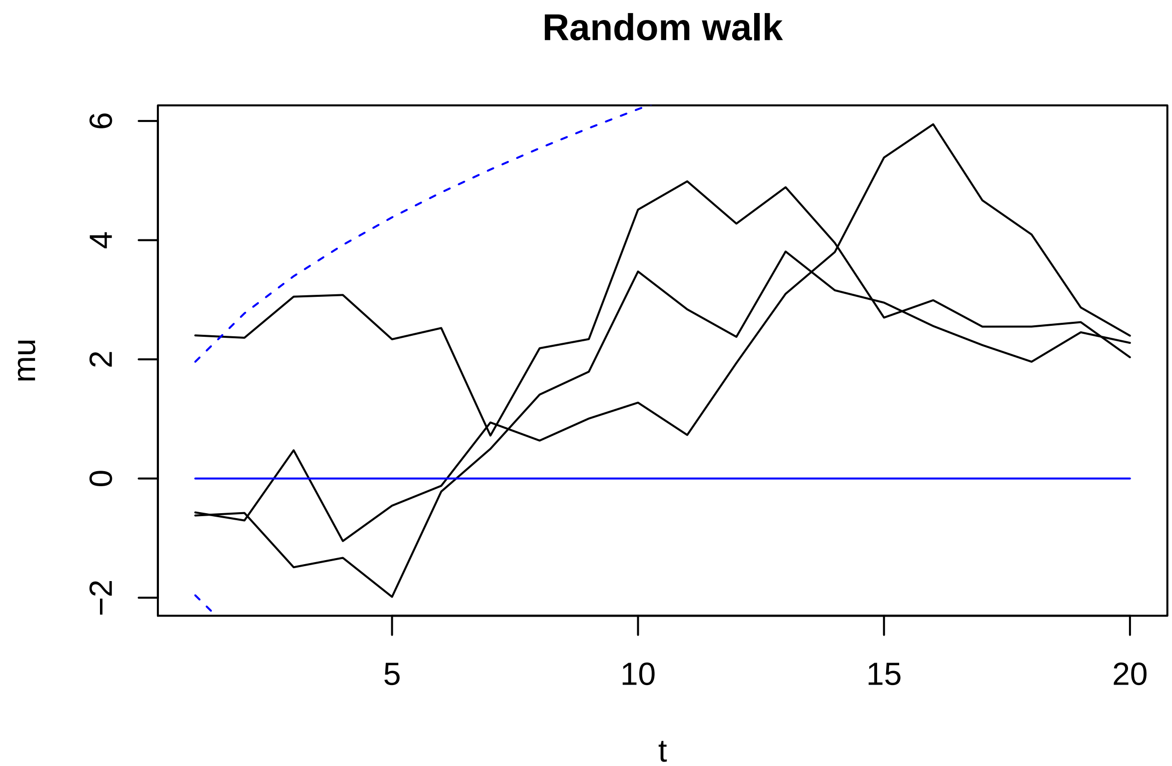


Random walk

The steps in a random walk are white noise.

$$a_t \sim N(0, \tau^2)$$
$$\mu_{t+1} = \mu_t + a_t$$
$$\mu_{t+1} | \mu_t \sim N(\mu_t, \tau^2)$$
$$u_0 = 0 \implies \mu_t \sim N(0, t\tau^2)$$

```
# Simulate random walk
a <- matrix(rnorm(n_samps), nrow = n_t)
mu <- apply(a, 2, cumsum)

# Plot it
matplot(t, mu, type = 'l', main = "Random walk", col = 1, lty = 1)
matlines(t, quants * sqrt(t), col = 'blue', lty = q_lty)
```



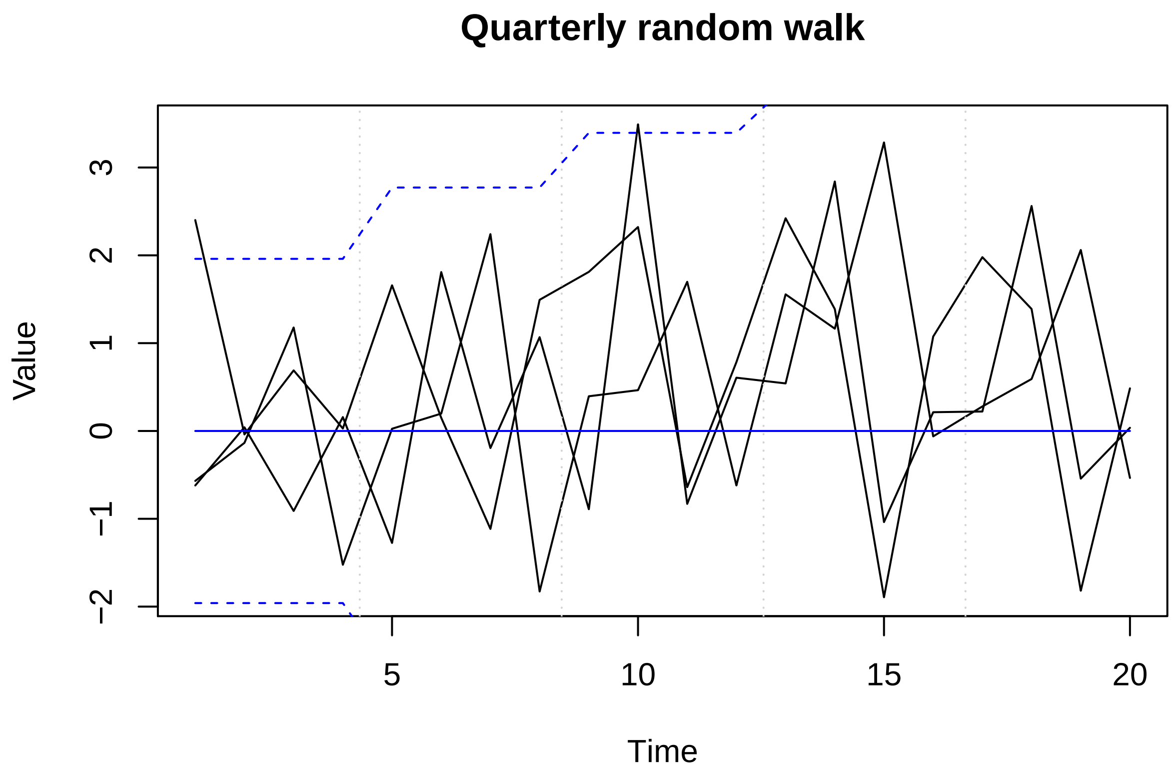
Quarterly random walk

$$\mu_{t+1} = \mu_{t-3} + a_t$$

$$\mu_{t+1} | \mu_{t-3} \sim N\left(\mu_{t-3}, \tau^2\right)$$

$$u_0 = 0 \implies \mu_t \sim N\left(0, ceiling\left(\frac{t}{4}\right)\tau^2\right)$$

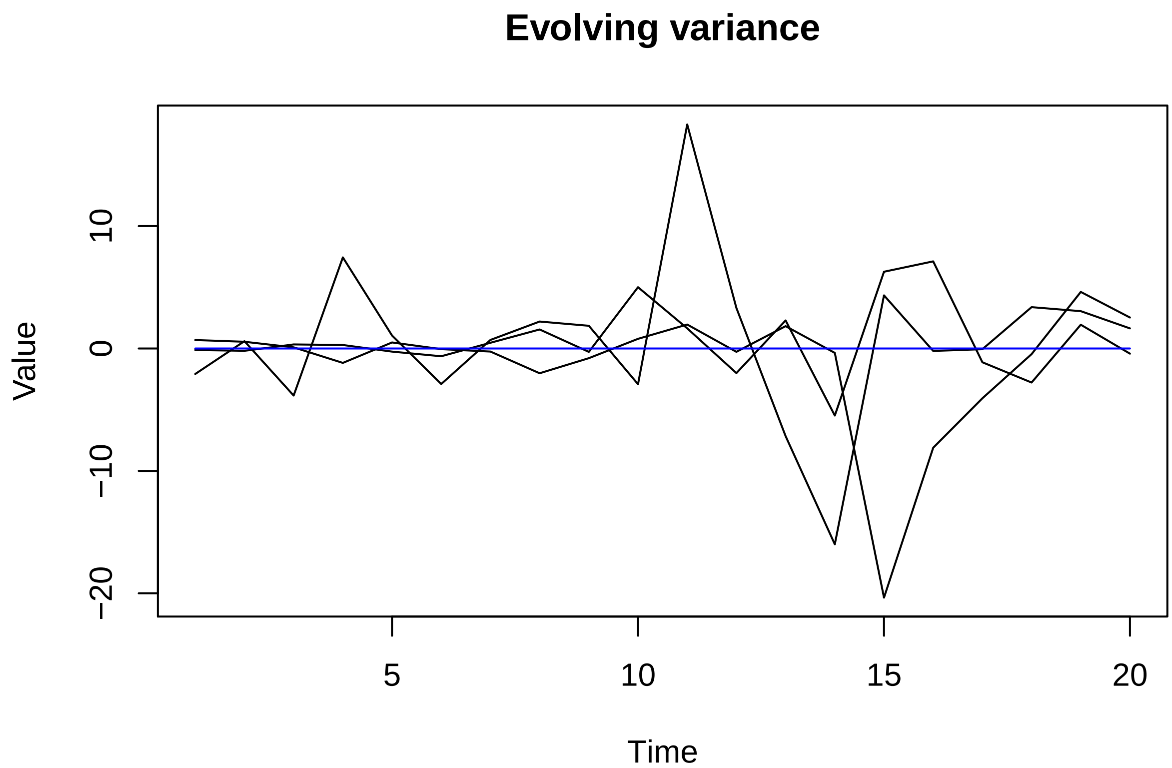
```
# Simulate quarterly random walk
mu_q <- apply(a, 2, function(column)
  as.vector(t(apply(matrix(column, ncol = 4, byrow = T), 2, cumsum))))

# Plot it
matplot(mu_q, type = 'l', xlab = "Time", ylab = "Value",
        main = "Quarterly random walk", col = 1, lty = 1)
grid(nx = n_t / 4, ny = NA)
matlines(t, quants * sqrt(ceiling(t / 4)), col = 'blue', lty = q_lty)
```



Evolving variance

$$h_{t+1} = h_t + a_t$$
$$y_t = \exp\left(\frac{h_t}{2}\right) e_t$$
$$y_t \not\sim N$$
$$y_t | h_t \sim N(0, \exp(h_t))$$
$$h_0 = 0 \implies \log\{var(y_t)\} \sim N(0, t\tau^2)$$

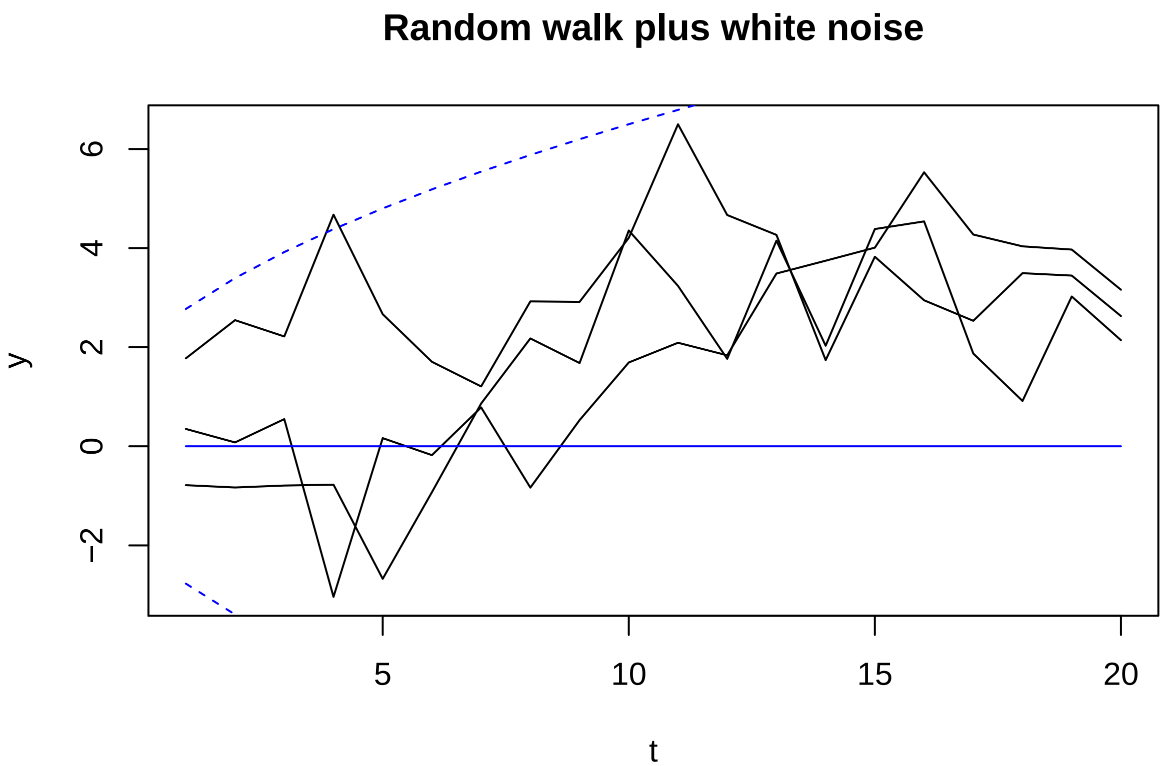
```
# Simulate evolvng variance
h <- mu
y_v <- exp(h / 2) * e

# Plot it
matplot(y_v, type = 'l', xlab = "Time", ylab = "Value",
        main = "Evolving variance", col = 1, lty = 1)
lines(rep(0, n_t), col = 'blue')
```



Filtering for a random walk plus white noise model

$$y_t = \mu_t + e_t$$
$$y_t | \mu_t \sim N(\mu_t, \sigma^2)$$
$$y_t \sim N(0, t\tau^2 + \sigma^2)$$

```
# Simulate and plot random walk plus white noise
y <- mu + e
matplot(t, y, type = 'l', main = "Random walk plus white noise", col = 1,
        lty = 1)
matlines(t, quants * sqrt(t + 1), col = 'blue', lty = q_lty)
```



$$x_t := \mu_t$$

$$\tilde{y}_t \sim N(\tilde{x}_t, \sigma^2 I_t)$$

$$\tilde{x}_t \sim N(\tilde{x}_{t-1}, \tau^2 I_t)$$

$$\implies (\tilde{x}_t, \tilde{y}_t) \sim N$$

$$E(\tilde{x}_t|\tilde{y}_t) = E(\tilde{x}_t) + Cov(\tilde{x}_t, \tilde{y}_t)Var(\tilde{y}_t)^{-1}\{\tilde{y}_t - E(\tilde{y}_t)\}$$

$$Var(\tilde{x}_t|\tilde{y}_t) = Var(\tilde{x}_t) - Cov(\tilde{x}_t, \tilde{y}_t)Var(\tilde{y}_t)^{-1}Cov(\tilde{y}_t, \tilde{x}_t)$$

$$E(\tilde{x}_t) = E(\tilde{y}_t) = 0$$

$$Cov(\tilde{x}_t, \tilde{y}_t) = Cov(\tilde{y}_t, \tilde{x}_t)' = (Cov(\tilde{x}_t, \tilde{x}_t) + Cov(\tilde{e}_t, \tilde{x}_t))' = Var(\tilde{x}_t)$$

$$Var(\tilde{y}_t) = Var(\tilde{x}_t) + \sigma^2 I_t$$

$$\begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \cdots & & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \cdots \\ x_t \end{bmatrix} = \begin{bmatrix} 1 \\ \\ \\ \end{bmatrix} x_1 + \begin{bmatrix} a_1 \\ \cdots \\ a_t \end{bmatrix}$$

$$D_t \tilde{x}_t = \tilde{g}_t x_1 + \tilde{a}_t$$

$$D_t Var(\tilde{x}_t) D_t' = K^2 \tilde{g}_t \tilde{g}_t' + \tau^2 I_t$$

$$x_1 \sim N(0, K^2)$$

$$Var(\tilde{x}_t) = K^2 D_t^{-1} \tilde{g}_t \tilde{g}_t' D_t'^{-1} + \tau^2 D_t^{-1} D_t'^{-1}$$

$$E(\tilde{x}_t|\tilde{y}_t) = Var(\tilde{x}_t)\{Var(\tilde{x}_t) + \sigma^2 I_t\}^{-1}\tilde{y}_t$$

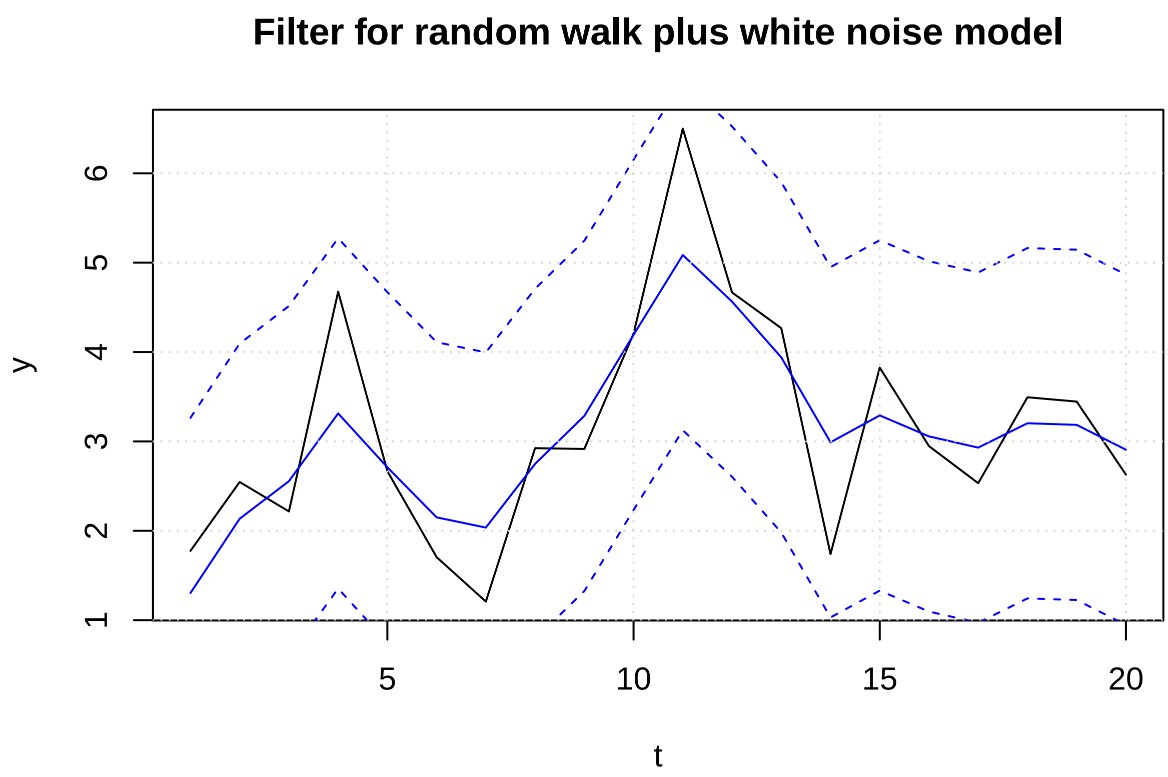
$$Var(\tilde{x}_t|\tilde{y}_t) = Var(\tilde{x}_t) - Var(\tilde{x}_t)\{Var(\tilde{x}_t) + \sigma^2 I_t\}^{-1}Var(\tilde{x}_t)$$

$$\implies \tilde{x}_t|\tilde{y}_t$$

$$\implies \tilde{y}_{t+k}|\tilde{y}_t$$

$$K = 0, \tau = 1, \sigma = 1 \implies Var(\tilde{x}_t) = D_t^{-1} D_t'^{-1}$$

```
# Compute filter given all samples for each realization
D <- diag(n_t)
D[cbind(2:n_t, 1:(n_t - 1))] <- -1
var_x_t <- solve(D) %*% solve(t(D))
E_x_given_y <- var_x_t %*% solve(var_x_t + diag(n_t)) %*% y

# Plot it
plot(t, y[, 1], type = 'l', col = 1, lty = 1, ylab = "y",
     main = "Filter for random walk plus white noise model")
matlines(t, rep(E_x_given_y[, 1], 3) + quants, col = 'blue', lty = q_lty)
grid()
```




```
# CI's look a bit wide
```

Likelihood computation using LDL decomposition and innovations ϵ_t

$Var(\tilde{y}_t)$ positive definite $\implies$ $Var(\tilde{y}_t) = L_t D_t L_t'$, where L is lower triangular with ones on the main diagonal, and D is diagonal.

$$\tilde{\epsilon}_t := \begin{bmatrix} \epsilon_1 \\ ... \\ \epsilon_t \end{bmatrix} = L_t^{-1} \tilde{y}_t$$

$$Var(\tilde{\epsilon}_t) = L_t^{-1} Var(\tilde{y}_t)(L_t^{-1})' = D_t$$

L_t^{-1} is also lower triangular with ones on the main diagonal.

$$l(\theta) \propto -\frac{1}{2} \left\{ \tilde{y}_t'(Var(\tilde{y}_t|\theta))^{-1}\tilde{y}_t + \log|Var(\tilde{y}_t|\theta)| \right\}$$

$$l(\theta) \propto -\frac{1}{2} \left\{ \tilde{y}_t'(L_t D_t L_t')^{-1}\tilde{y}_t + \log|L_t D_t L_t'| \right\}$$

$$= -\frac{1}{2} \left(\tilde{\epsilon}_t' D_t^{-1} \tilde{\epsilon}_t + \log|D_t| \right)$$

$$= -\frac{1}{2} \left(\sum_{i=1}^{t} \frac{\epsilon_i^2}{d_i} + d_i \right)$$

Kalman filter