bssm: Bayesian Inference of Exponential Family State Space Models in R

Jouni Helske and Matti Vihola
University of Jyväskylä, Department of Mathematics and Statistics, Finland

June 8, 2016

Introduction

The R package `bssm` is designed for Bayesian inference of exponential family state space models of form

$$\begin{aligned} p(y_t|Z_t\alpha_t, x_t'\beta) & \qquad \text{(observation equation)} \\ \alpha_{t+1} = T_t\alpha_t + R_t\eta_t, & \qquad \text{(transition equation)} \end{aligned}$$

where $\eta_t \sim N(0, I_k)$ and $\alpha_1 \sim N(a_1, P_1)$ independently of each other, x_t contains the exogenous covariate values at time time, with β corresponding to the regression coefficients. Currently, following observational level distributions are supported:

- Gaussian distribution: $p(y_t|Z_t\alpha_t, x_t\beta) = x_t'\beta + Z_t\alpha_t + H_t\epsilon_t$ with $\epsilon_t \sim N(0,1)$.
- Poisson distribution: $p(y_t|Z_t\alpha_t, x_t\beta) = \text{Poisson}(\phi_t\text{exp}(x_t'\beta + Z_t\alpha_t))$, where ϕ_t is the exposure at time t.
- Binomial distribution: $p(y_t|Z_t\alpha_t, x_t\beta) = \text{binomial}(\phi_t, \exp(x_t'\beta + Z_t\alpha_t)/(1 + \exp(x_t'\beta + Z_t\alpha_t)))$, where ϕ_t is the size and $\exp(x_t\beta + Z_t\alpha_t)/(1 + \exp(x_t'\beta + Z_t\alpha_t))$ is the probability of the success.
- Binomial distribution: $p(y_t|Z_t\alpha_t, x_t\beta) = \text{negative binomial}(\exp(x_t'\beta + Z_t\alpha_t), \phi)$, where $\exp(x_t'\beta + Z_t\alpha_t)$ is the expected value and ϕ_t is the dispersion parameter.

The `bssm` package aims to provide easy-to-use functions for fully Bayesian inference, especially forecasting, of common time series models such basic structural time series model (BSM) (Harvey 1989) with exogenous covariates, making it straighforward and efficient to make predictions and other inference in a Bayesian setting.

The Bayesian framework of `bssm` is based on Markov chain Monte Carlo (MCMC) approach with adaptive random walk Metropolis updating, using RAM algorithm by (Vihola 2012). This approach seems to work very well in practice, and it allows an efficient "black-box" type of modelling were the user does not need to be too concerned of "tuning" of the MCMC algoritm. The priors for the unknown model parameters θ (regression coefficients β and unknown elements in system matrices of the observation or state equations) are currently assumed to be uniform on some user-defined closed interval $[a, b]$, although this assumption could be relaxed in the future.

MCMC algorithm for Gaussian state space models

For Gaussian models given the parameters θ, the likelihood of the model can be computed using the well known Kalman filter recursions. The complete adaptive MCMC algorithm of `bssm` for Gaussian models is as follows (modified from (Vihola 2012)).

Given the target acceptance rate α^* (e.g. 0.234) and $\gamma \in (0, 1]$, at iteration i

1. Compute the proposal $\theta_i' = \theta_{i-1} + S_{i-1}u_i$, where u_i is simulated from the standard d-dimensional Gaussian distribution and S_{i-1} is a lower diagonal matrix with positive diagonal elements.
2. Accept the proposal with probability $\alpha_i := \min\{1, p(y|\theta_i')/p(y|\theta_{i-1})\}$.

3. If the proposal θ'_i is accepted, set $\theta_i = \theta'_i$ and simulate a realization (or multiple realizations) of the states α from $p(\alpha|y, \theta'_i)$ using the simulation smoothing algorithm by (Durbin and Koopman 2002). Otherwise, set $\theta_i = \theta_{i-1}$ and $(\alpha_1, \ldots, \alpha_n)_i = (\alpha_1, \ldots, \alpha_n)_{i-1}$.
4. Compute (using Cholesky update or downdate algorithm) the Cholesky factor matrix S_i satisfying the equation
$$S_i S_i^T = S_{i-1}\left(I + \min\{1, di^{-\gamma}\}(\alpha_i - \alpha^*)\frac{u_i u_i^T}{\|u_i\|^2}\right)S_{i-1}^T.$$

If the interest is in the posterior means and variances of the states, we can replace the simulation smoothing in step 3 with standard fixed interval smoothing which gives the smoothed estimates (expected values and variances) of the states given the data and the model parameters. From these, the posterior means and variances of the states can be computed straightforwardly.

Non-Gaussian models

For non-Gaussian models we focus on exponential family models so that we can use the Laplace approximation of the observational density, giving us an approximating Gaussian model which has the same mode of $p(\alpha|y)$ as the original model. This is based on (Durbin and Koopman 2000). This approximating Gaussian model seems to work very well in practice, and thus we can use it in MCMC scheme directly, which results in an approximate Bayesian inference. We can also use the approximating model together with importance sampling which produces exact Bayesian inference on $p(\alpha, \theta|y)$. We can factor the likelihood of the non-Gaussian model as (Durbin and Koopman 2012)
$$\begin{aligned}
p(y|\theta) &= \int g(\alpha, y|\theta)\mathrm{d}\alpha \\
&= g(y|\theta)E_g\left[\frac{p(y|\alpha, \theta)}{g(y|\alpha, \theta)}\right],
\end{aligned}$$
where $g(y|\theta)$ is the likelihood of the Gaussian approximating model and the expectation is taken with respect to the Gaussian density $g(\alpha|y, \theta)$. Equivalently we can write
$$\begin{aligned}
\log p(y|\theta) &= \log g(y|\theta) + \log E_g\left[\frac{p(y|\alpha, \theta)}{g(y|\alpha, \theta)}\right] \\
&= \log g(y|\theta) + \log\frac{p(y|\hat{\alpha}, \theta)}{g(y|\hat{\alpha}, \theta)} + \log E_g\left[\frac{p(y|\alpha, \theta)/p(y|\hat{\alpha}, \theta)}{g(y|\alpha, \theta)/g(y|\hat{\alpha}, \theta)}\right] \\
&= \log g(y|\theta) + \log\hat{w} + \log E_g w^* \\
&\approx \log g(y|\theta) + \log\hat{w} + \log\frac{1}{N}\sum_{j=1}^N w_j^*,
\end{aligned}$$
where $\hat{\alpha}$ is the conditional mode estimate obtained from the approximating Gaussian model. For approximating inference without the importance sampling we simply omit the term $\log\frac{1}{N}\sum_{j=1}^N w_i^*$. In principle, using the exact Bayesian inference we should simulate multiple realizations of the states α in each iteration of MCMC in order to compute $\log\frac{1}{N}\sum_{j=1}^N w_j^*$. Fortunately, we can use so called delayed acceptance approach (Christen and Fox 2005; Banterle et al. 2015) which speeds up the computation considerably:

1. Make initial acceptance of the given proposal θ'_i with probability $\min\left\{1, \frac{g(y|\theta'_i)\hat{w}_i}{g(y|\theta_{i-1})\hat{w}_{i-1}}\right\}$.
2. If accepted, perform the importance sampling of the states α and make the delayed acceptance with probability $\min\{1, \sum_{j=1}^N w_{i,j}^{*'}/\sum_{j=1}^N w_{i-1,j}^*\}$.
3. If the delayed acceptance is successful, set $\theta_i = \theta'_i$ and sample one (or multiple) realization of the previously simulated states with weights $w_{i,j}, j = 1, \ldots, N$ (with replacement in case of multiple samples are stored). Otherwise, set $\theta_i = \theta_{i-1}$ and similarly for the states.

Prediction intervals

For predicting future observations, one can extend the time series data with missing values which are handled straightforwardly by Kalman filter. This gives us point predictions in form of expected values, but for prediction intervals, no analytical formulas are available as the posterior distribution of the state is not necessarily normal. A standard way of obtaining the prediction intervals is to simulate future observations and use empirical quantiles for the limits of prediction interval. However, this is computationally expensive as a very large number of simulations is needed for accurate values of tail probabilities. In addition of standard quantile method, the bssm package contains alternative method for Gaussian models based on the parametric method by (Helske et al. 2015, J. Helske (2016)). Here, instead of simulating the future values, we compute the expected values and variances of the future values using the Kalman filtering, and then solve simple equation based on the cumulative distribution function of Normal distribution.

For Gaussian models with fixed number of MCMC iterations, the parametric method gives considerably more accurate results with approximately equal computational burden. It is also possible to use the parametric method for non-Gaussian models as well, but as we need the importance sampling step for the estimation of the inverse link-function of the model (such as $\phi_t \exp(Z_t \alpha_t)$ in Poisson case), the method is computationally slightly more intensive than the quantile method.

Example

Here is a short example:

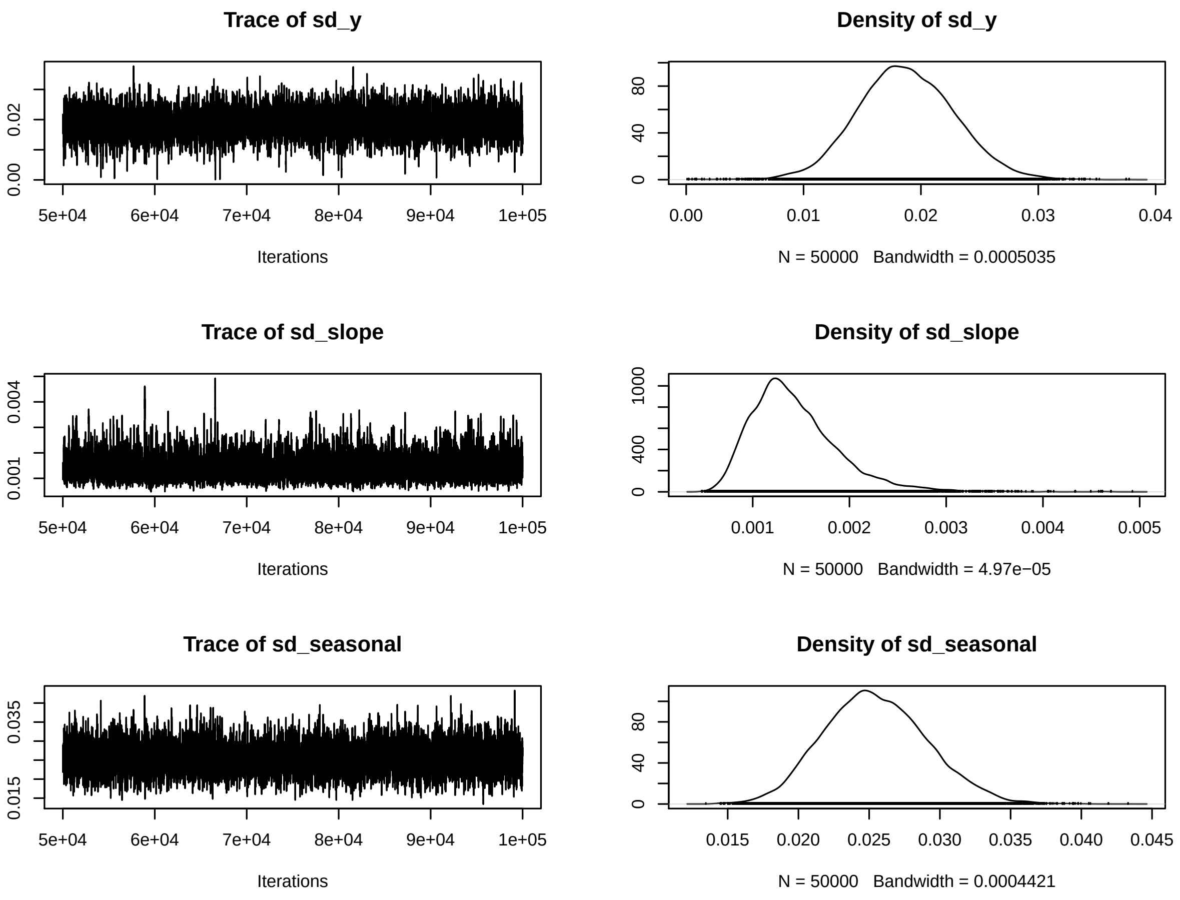
```r
library("bssm")
set.seed(123)

init_sd <- 0.1 * sd(log10(UKgas))
model <- bsm(log10(UKgas), sd_y = init_sd,
  sd_slope = init_sd, sd_seasonal = init_sd)
mcmc_out <- run_mcmc(model, n_iter = 1e5, seed = 123)
mcmc_out$acceptance_rate
```

```
## [1] 0.2338
```

```r
summary(mcmc_out$theta)
```

```
##
## Iterations = 50001:1e+05
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 50000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                 Mean        SD  Naive SE Time-series SE
## sd_y        0.018734 0.0041412 1.852e-05      6.135e-05
## sd_slope    0.001434 0.0004563 2.041e-06      7.796e-06
## sd_seasonal 0.025450 0.0036308 1.624e-05      5.728e-05
##
## 2. Quantiles for each variable:
##
##                 2.5%      25%     50%      75%    97.5%
## sd_y        0.010868 0.015940 0.01863 0.021481 0.026990
## sd_slope    0.000771 0.001121 0.00136 0.001668 0.002533
## sd_seasonal 0.018799 0.022907 0.02530 0.027846 0.032879
```

```
plot(mcmc_out$theta)
```



```
# posterior mode estimates (diffuse priors):
mcmc_out$theta[which.max(mcmc_out$logLik), ]
```

```
##        sd_y    sd_slope sd_seasonal
## 0.018657305 0.001239168 0.025118866
```

```
# posterior covariance matrix:
cov(mcmc_out$theta)
```

```
##                      sd_y      sd_slope   sd_seasonal
## sd_y         1.714962e-05 -4.005506e-07 -9.339645e-06
## sd_slope    -4.005506e-07  2.082470e-07  1.920072e-07
## sd_seasonal -9.339645e-06  1.920072e-07  1.318288e-05
```

```
# compare to shape of the proposal distribution:
cor(mcmc_out$theta)
```

```
##                   sd_y   sd_slope sd_seasonal
## sd_y         1.0000000 -0.2119536  -0.6211523
## sd_slope    -0.2119536  1.0000000   0.1158839
## sd_seasonal -0.6211523  0.1158839   1.0000000
```

```
cov2cor(mcmc_out$S %*% t(mcmc_out$S))
```

```
##                   sd_y    sd_slope sd_seasonal
## sd_y         1.0000000 -0.24708355 -0.61881987
## sd_slope    -0.2470836  1.00000000  0.09644382
## sd_seasonal -0.6188199  0.09644382  1.00000000
```

Smoothed trend:

```
ts.plot(model$y, rowMeans(mcmc_out$alpha[, "level", ]), col = 1:2)
```

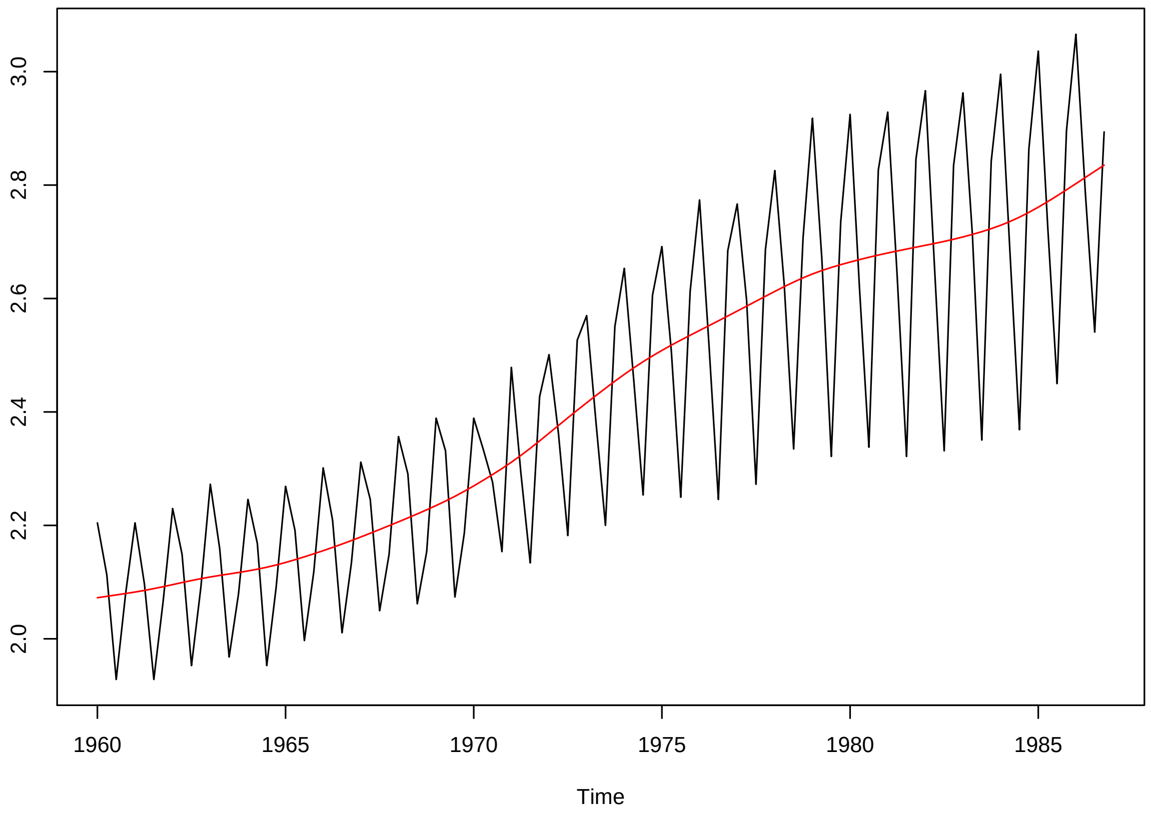


Figure 1: Smoothed trend component.

Prediction intervals:

```
pred <- predict(model, n_iter = 1e4, n_ahead = 40,
  probs = c(0.025, 0.1, 0.9, 0.975), S = mcmc_out$S)
ts.plot(log10(UKgas), pred$mean, pred$intervals[,-3],
  col = c(1, 2, c(3, 4, 4, 3)), lty = c(1, 1, rep(2, 4)))
```

With ggplot2:

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

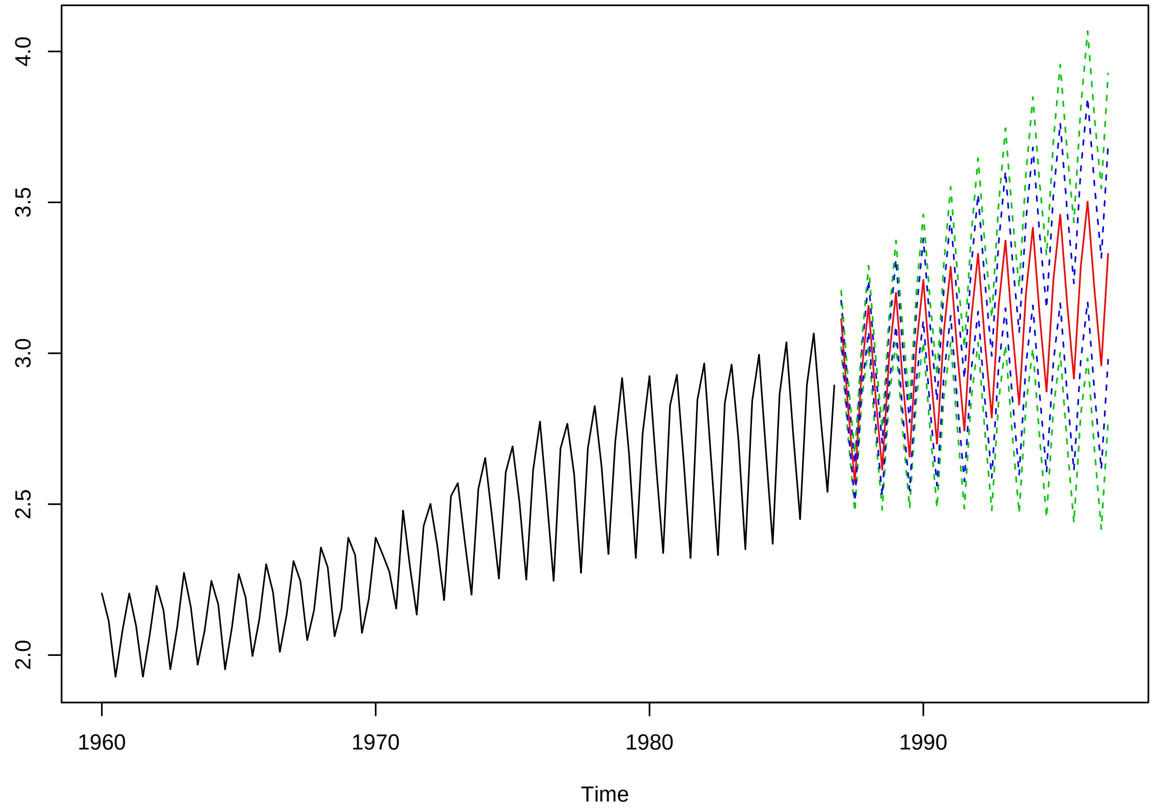


Figure 2: Mean predictions and prediction intervals.

```
autoplot(pred, interval_colour = "red", alpha_fill = 0.2)
```

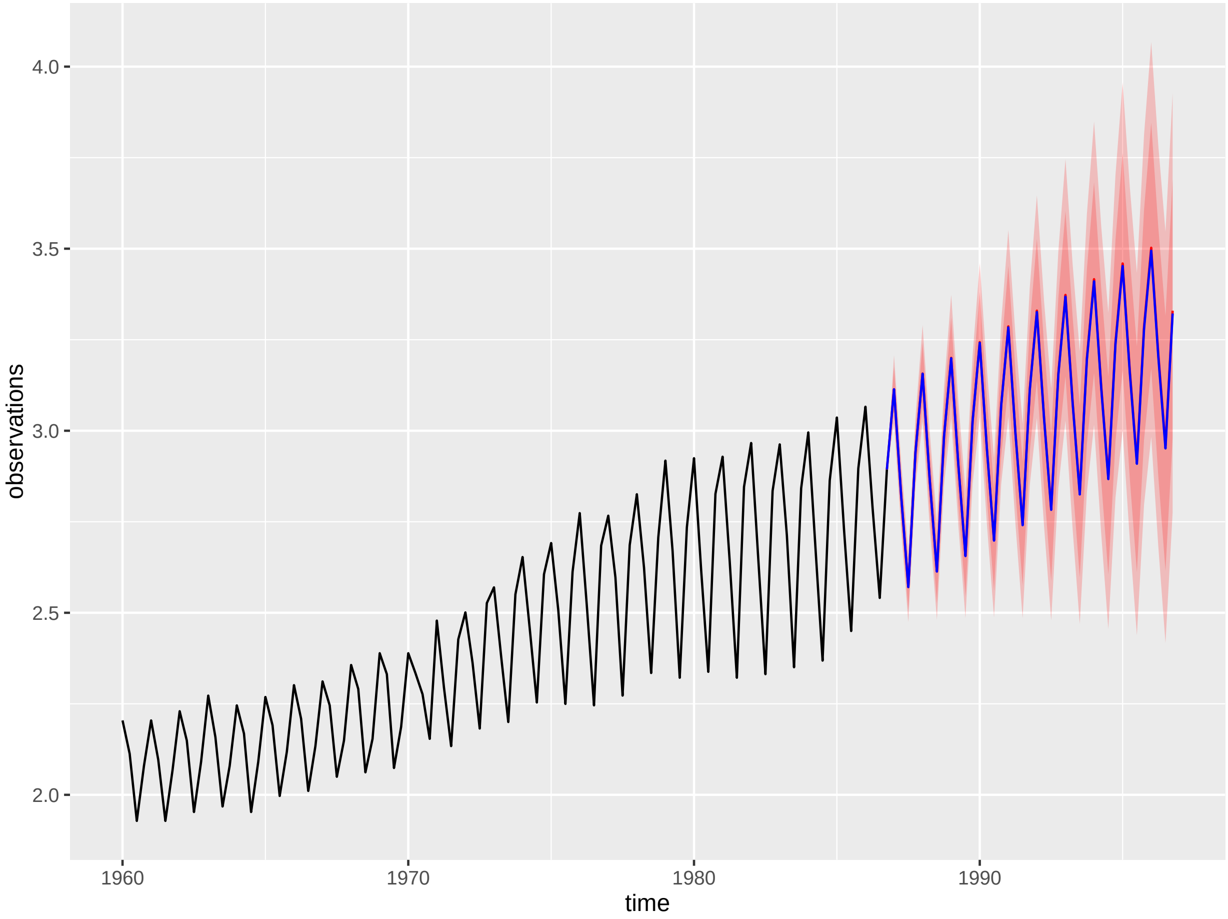


Figure 3: Prediction plots with ggplot2.

References

Banterle, M., C. Grazian, A. Lee, and C. P. Robert. 2015. "Accelerating Metropolis-Hastings algorithms by Delayed Acceptance." *ArXiv E-Prints*, March. http://arxiv.org/abs/1503.00996.

Christen, J. Andrés, and Colin Fox. 2005. "Markov Chain Monte Carlo Using an Approximation." *Journal of Computational and Graphical Statistics* 14 (4): 795–810. doi:10.1198/106186005X76983.

Durbin, J., and S. J. Koopman. 2000. "Time Series Analysis of Non-Gaussian Observations Based on State Space Models from Both Classical and Bayesian Perspectives." *Journal of Royal Statistical Society B* 62: 3–56.

———. 2002. "A Simple and Efficient Simulation Smoother for State Space Time Series Analysis." *Biometrika* 89: 603–15.

———. 2012. *Time Series Analysis by State Space Methods*. 2nd ed. New York: Oxford University Press.

Harvey, A. C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University

Press.

Helske, Jouni. 2016. “Computationally Efficient Bayesian Prediction Intervals for Gaussian State State Models.” *Working Paper*.

Helske, J., Nyblom, and J. 2015. “Improved Frequentist Prediction Intervals for Autoregressive Models by Simulation.” In *Unobserved Components and Time Series Econometrics*, edited by Koopman, S. J., Shephard, and N., 291–309. Oxford University Press.

Vihola, Matti. 2012. “Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate.” *Statistics and Computing* 22 (5): 997–1008. doi:10.1007/s11222-011-9269-5.