

# HW5MuyangShi

Muyang Shi

2024-03-21

Note: the cpp source code to this document can be found on my Github, listed as `MCMC.cpp`, [here](#).

## Problem

Assume the model

$$X_j \sim \begin{cases} \text{Poisson}(\lambda_1), & j = 1, \dots, \theta \\ \text{Poisson}(\lambda_2), & j = \theta + 1, \dots, 112 \end{cases}$$

Assume  $\lambda_i | \alpha \sim \text{Gamma}(3, \alpha)$  for  $i = 1, 2$  where  $\alpha \sim \text{Gamma}(10, 10)$  and assume  $\theta$  follows a discrete uniform distribution over  $\{1, \dots, 111\}$ . We would like to estimate the posterior distribution of the model parameters via a Gibbs sampler.

**(a) Derive the conditional distributions necessary to carry out Gibbs sampling for the change-point model.**

The joint likelihood is:

$$\begin{aligned} p(\boldsymbol{\Theta} | \mathbf{X}) &\propto p(\mathbf{X} | \boldsymbol{\Theta}) p(\boldsymbol{\Theta}) \\ &= p(\mathbf{X} | \lambda_1, \lambda_2, \theta) \cdot p(\lambda_1 | \alpha) p(\lambda_2 | \alpha) \cdot p(\alpha) p(\theta) \\ &= \prod_{j=1}^{\theta} \frac{\lambda_1^{X_j} e^{-\lambda_1}}{X_j!} \prod_{j=\theta+1}^{112} \frac{\lambda_2^{X_j} e^{-\lambda_2}}{X_j!} \cdot \frac{\alpha^3}{\Gamma(3)} \lambda_1^{3-1} e^{-\alpha \lambda_1} \frac{\alpha^3}{\Gamma(3)} \lambda_2^{3-1} e^{-\alpha \lambda_2} \cdot \frac{10^{10}}{\Gamma(10)} \alpha^{10-1} e^{-10\alpha} \frac{1}{111} \mathbf{1}(\theta \in \{1, \dots, 111\}) \end{aligned}$$

And the full conditionals are:

For  $\lambda_1$ :

$$\begin{aligned} p(\lambda_1 | \mathbf{X}, \lambda_2, \alpha, \theta) &\propto p(\mathbf{X} | \lambda_1, \lambda_2, \alpha, \theta) p(\lambda_1 | \alpha) \\ &\propto \prod_{j=1}^{\theta} \frac{\lambda_1^{X_j} e^{-\lambda_1}}{X_j!} \lambda_1^{3-1} e^{-\alpha \lambda_1} \\ &\propto \lambda_1^{\sum_{j=1}^{\theta} X_j + 3 - 1} e^{-(\theta + \alpha) \lambda_1} \\ &\sim \text{Gamma}(3 + \sum_{j=1}^{\theta} X_j, \theta + \alpha) \end{aligned}$$

Similarly, for  $\lambda_2$ :

$$\begin{aligned} p(\lambda_2 | \mathbf{X}, \lambda_1, \alpha, \theta) &\propto \lambda_2^{\sum_{j=\theta+1}^{112} X_j + 3 - 1} e^{-(112 - \theta + \alpha) \lambda_2} \\ &\sim \text{Gamma}(3 + \sum_{j=\theta+1}^{112} X_j, 112 - \theta + \alpha) \end{aligned}$$

And for  $\alpha$ :

$$\begin{aligned} p(\alpha|\mathbf{X}, \lambda_1, \lambda_2, \theta) &\propto \alpha^3 e^{-\alpha\lambda_1} \alpha^3 e^{-\alpha\lambda_2} \alpha^{10-1} e^{-10\alpha} \\ &= \alpha^{16-1} e^{-(10+\lambda_1+\lambda_2)\alpha} \\ &\sim \text{Gamma}(16, 10 + \lambda_1 + \lambda_2) \end{aligned}$$

Finally for  $\theta$ :

$$\begin{aligned} p(\theta|\mathbf{X}, \lambda_1, \lambda_2, \alpha) &\propto \prod_{j=1}^{j=\theta} \frac{\lambda_1^{X_j} e^{-\lambda_1}}{X_j!} \prod_{j=\theta+1}^{j=112} \frac{\lambda_2^{X_j} e^{-\lambda_2}}{X_j!} \mathbf{1}(\theta \in \{1, \dots, 111\}) \\ &\propto \lambda_1^{\sum_{j=1}^{\theta} X_j} e^{-\theta\lambda_1} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-(112-\theta)\lambda_2} \mathbf{1}(\theta \in \{1, \dots, 111\}) \end{aligned}$$

Note that we can't see the distribution of  $\theta$  through just eyeballing the equation. We'd be using a Metropolis approach – 1.) proposing  $\theta$  from the discrete Uniform between 1 and 111, 2.) calculate the likelihood (equation above) ratio to decide whether we accept or reject. (Also note that this proposal is symmetric, so there's no “Hasting” ratio.)

**(b) Implement the Gibbs sampler. Use a suite of convergence diagnostics to evaluate the convergence and mixing of your sampler.**

The Gibbs sampler is implemented in `RcppArmadillo` with the function name `MCMC`. We ran four chains (for the Gelman diagnostic), and for inference we will use the first chain.

Note that, as we plot below, where

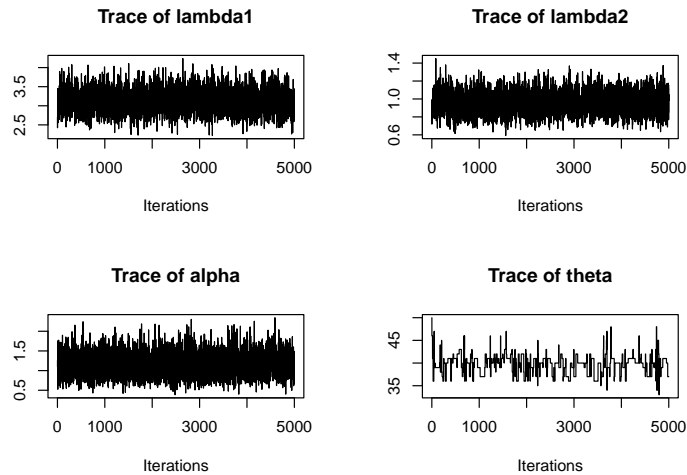
- the traceplot looks fine in that the chain seems to have converged and the mixing is good,
- the autocorrelation plot also seems fine in that we have decent effective sample size;  $\theta$  is discrete so the traceplot for it actually doesn't look that bad
- and the Gelman diagnostic R is very close to 1 which is good (indicate that the chains have converged).

```
coal <- read.table('coal.dat', skip = 1) # load data

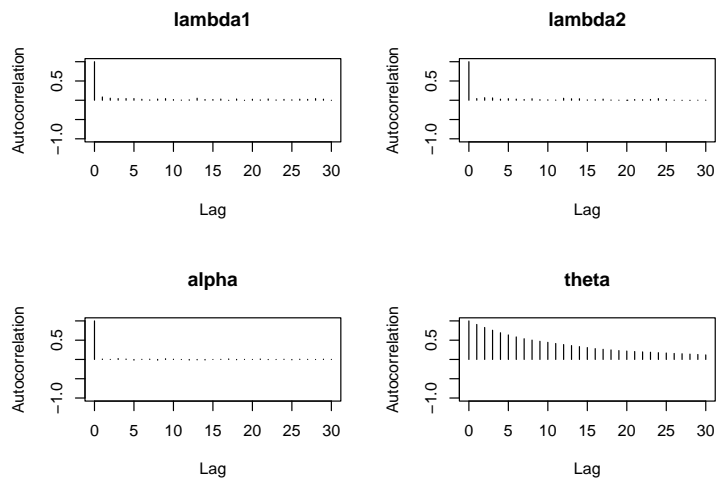
# initial values
theta_0 <- 50
alpha_0 <- 1.1
lambda1_0 <- mean(coal[1:theta_0,2])
lambda2_0 <- mean(coal[theta_0:nrow(coal),2])

# run a couple chains using Rcpp
set.seed(2345)
chain1 <- MCMC(5000, coal$V2, lambda1_0, lambda2_0, alpha_0, theta_0)
chain2 <- MCMC(5000, coal$V2, lambda1_0, lambda2_0, alpha_0, theta_0)
chain3 <- MCMC(5000, coal$V2, lambda1_0, lambda2_0, alpha_0, theta_0)
chain4 <- MCMC(5000, coal$V2, lambda1_0, lambda2_0, alpha_0, theta_0)
```

```
plot(chain1.mcmc, density = FALSE)
```



```
autocorr.plot(chain1.mcmc)
```

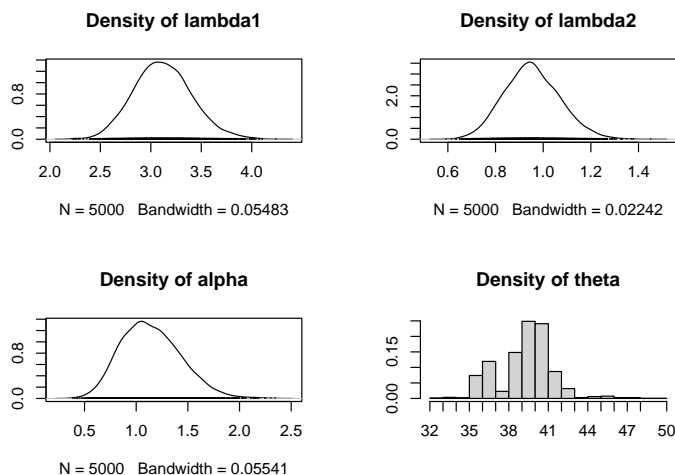


```
gelman.diag(mcmc_list)
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## lambda1      1.00      1.00
## lambda2      1.00      1.00
## alpha        1.00      1.00
## theta        1.01      1.02
##
## Multivariate psrf
##
## 1.01
```

(c) Construct density histograms and a table of summary statistics for the approximate posterior distributions of  $\theta$ ,  $\lambda_1$ , and  $\lambda_2$ . Are symmetric HPD intervals appropriate for all of these parameters?

Here is the density histograms for the four parameters,  $\lambda_1$ ,  $\lambda_2$ ,  $\alpha$  and  $\theta$ :



Here is a table of summary statistics for the approximate posterior distributions of  $\theta$ ,  $\lambda_1$ ,  $\lambda_2$ :

```
##
## Iterations = 1:5000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## lambda1  3.120 0.2865 0.004051      0.005849
## lambda2  0.953 0.1162 0.001643      0.002437
## alpha    1.145 0.2871 0.004061      0.004061
## theta    39.746 2.0683 0.029250      0.137149
##
## 2. Quantiles for each variable:
##
##           2.5%      25%      50%      75%      97.5%
## lambda1  2.5928  2.9245  3.1098  3.305   3.726
## lambda2  0.7381  0.8735  0.9487  1.030   1.195
## alpha    0.6593  0.9341  1.1219  1.333   1.757
## theta    36.0000  39.0000  40.0000  41.000  43.000
```

The skewness in the distributions aren't "severe" – hence using symmetric HPD intervals are fine; however, one might argue that the distributions for  $\lambda_1$ ,  $\lambda_2$ , as well as for  $\theta$  are slightly right skewed, hence it might be better to use equal-tail intervals.

**(d) Interpret the results in the context of the problem.**

Based on the posterior distributions from the MCMC, we believe there is a 95% probability that:

- $\lambda_1$ , average number of yearly coal-mining disasters during the first period, is anywhere between 2.59 to 3.73
- $\lambda_2$ , average number of yearly coal-mining disasters during the second period, is anywhere between 0.74 to 1.20
- $\theta$  is anywhere between 36 to 43, i.e. the transition happened anywhere between the year of 1886 to 1893.

**(e) Change the prior for  $\lambda_1$  and  $\lambda_2$  to a half-normal distribution  $\propto N(0, \sigma^2)1_{[\lambda > 0]}$  with  $\sigma^2$  known. Derive the appropriate algorithm (MH or Gibbs) to carry out inference on the posterior distribution of  $\lambda_1$  and  $\lambda_2$ .**

We begin with the joint likelihood:

$$\begin{aligned}
 p(\Theta|\mathbf{X}) &\propto p(\mathbf{X}|\Theta)p(\Theta) \\
 &= p(\mathbf{X}|\lambda_1, \lambda_2, \theta) \cdot p(\lambda_1|\sigma^2)p(\lambda_2|\sigma^2) \cdot p(\theta) \\
 &= \prod_{j=1}^{\theta} \frac{\lambda_1^{X_j} e^{-\lambda_1}}{X_j!} \prod_{j=\theta+1}^{112} \frac{\lambda_2^{X_j} e^{-\lambda_2}}{X_j!} \cdot \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{\lambda_1^2}{2\sigma^2}\right) \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{\lambda_2^2}{2\sigma^2}\right) \cdot \frac{1}{111} \mathbf{1}(\theta \in \{1, \dots, 111\}) \\
 &\propto \prod_{j=1}^{\theta} \lambda_1^{X_j} e^{-\lambda_1} \prod_{j=\theta+1}^{112} \lambda_2^{X_j} e^{-\lambda_2} \cdot \frac{1}{\sigma^2} \exp\left(-\frac{\lambda_1^2}{2\sigma^2}\right) \exp\left(-\frac{\lambda_2^2}{2\sigma^2}\right) \cdot \mathbf{1}(\theta \in \{1, \dots, 111\})
 \end{aligned}$$

And the full conditionals,

For  $\lambda_1$ :

$$p(\lambda_1|\mathbf{X}, \lambda_2, \sigma, \theta) \propto \lambda_1^{\sum_{j=1}^{\theta} X_j} \exp\left(-\theta\lambda_1 - \frac{\lambda_1^2}{2\sigma^2}\right)$$

For  $\lambda_2$ :

$$p(\lambda_2|\mathbf{X}, \lambda_1, \sigma, \theta) \propto \lambda_2^{\sum_{j=\theta+1}^{112} X_j} \exp\left(-(112-\theta)\lambda_2 - \frac{\lambda_2^2}{2\sigma^2}\right)$$

And for  $\theta$ :

$$p(\theta|\mathbf{X}, \lambda_1, \lambda_2, \sigma) \propto \lambda_1^{\sum_{j=1}^{\theta} X_j} e^{-\theta\lambda_1} \lambda_2^{\sum_{j=\theta+1}^{112} X_j} e^{-(112-\theta)\lambda_2} \mathbf{1}(\theta \in \{1, \dots, 111\})$$

We will be using **random-walk metropolis** to update the parameters  $\lambda_1$  and  $\lambda_2$  (i.e. immediately reject if  $\lambda_1$  and  $\lambda_2$  falls out of support, so by doing it this way the hasting ratio will be 1).

**(f) Implement your MCMC algorithm from part (e). Try several values of  $\sigma^2$  based on your understanding of the problem. (What is a reasonable variance for  $\lambda$  based on what you know about the problem?).**

The MH sampler is implemented in `RcppArmadillo` with the function name `MCMC_halfnorm`. We ran four chains (for the Gelman diagnostic) across three values of  $\sigma \in \{0.5, 1.5, 2.5\}$ , and for inference we will use the first chain for each  $\sigma$ .

```

sigmas <- c(0.5,1.5,2.5)
set.seed(2345)

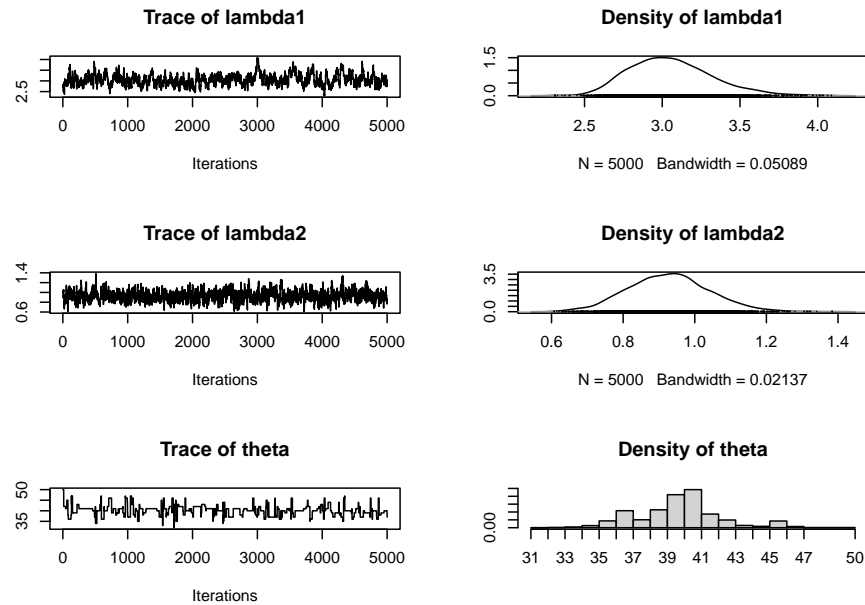
chain1.1 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[1], theta_0)
chain1.2 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[1], theta_0)
chain1.3 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[1], theta_0)
chain1.4 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[1], theta_0)

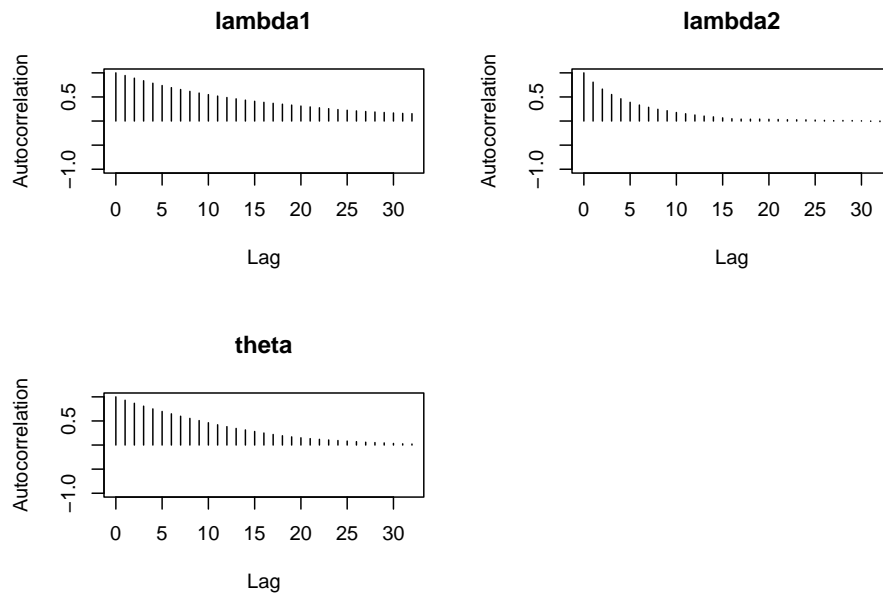
chain2.1 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[2], theta_0)
chain2.2 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[2], theta_0)
chain2.3 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[2], theta_0)
chain2.4 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[2], theta_0)

chain3.1 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[3], theta_0)
chain3.2 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[3], theta_0)
chain3.3 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[3], theta_0)
chain3.4 <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, sigmas[3], theta_0)

```

We looked at the traceplot, autocorrelation plots, and the Gelman potential scale reduction factors, and decide that the model with  $\sigma = 1.5$  yields the best result. Below are the traceplots, density plots, autocorrelation plots, and the Gelman PSRF for the model with  $\sigma = 1.5$ :





```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## lambda1      1.00      1.01
## lambda2      1.00      1.01
## theta        1.01      1.02
##
## Multivariate psrf
##
## 1
```

(g) Provide a few key results to compare the inferences that you'd make based on the two models. Which model do you prefer and why?

Here is a summary of posterior for the halfnorm model with  $\sigma = 1.5$ :

```
##
## Iterations = 1:5000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## lambda1  3.0551 0.2657 0.003757      0.021709
## lambda2  0.9274 0.1130 0.001599      0.005055
## theta   40.1808 2.4859 0.035156      0.184296
##
## 2. Quantiles for each variable:
```

```
##
##           2.5%    25%    50%    75%  97.5%
## lambda1  2.6047  2.8659  3.0339  3.2192  3.634
## lambda2  0.7168  0.8501  0.9272  0.9984  1.157
## theta    36.0000 39.0000 40.0000 41.0000 46.000
```

Based on the posterior distributions from the MCMC, we believe there is a 95% probability that:

- $\lambda_1$ , average number of yearly coal-mining disasters during the first period, is anywhere between 2.60 to 3.63
- $\lambda_2$ , average number of yearly coal-mining disasters during the second period, is anywhere between 0.72 to 1.16
- $\theta$  is anywhere between 36 to 46, i.e. the transition happened anywhere between the year of 1886 to 1896.

Note that qualitatively, the results is similar to the Gamma-Poisson model – estimates for  $\lambda_1$  being larger than  $\lambda_2$  suggests that the reduction in yearly coal-mining disasters; the estimates for  $\theta$  (change point) also being similar between the two models.

However, I would prefer the Gamma-Poisson model, because:

- Gibbs sampling does not involve the Metropolis-Hasting step, so mixing would not be an issue as we have the closed form posterior.
- Due to not having the Metropolis-Hasting update, the Gibbs sampler is also marginally faster than the MH.

**(h) Run both models in Nimble. Compare the results and speed from your code and Nimble's. Briefly discuss.**

Below we implement the model in `nimble`. Note that `nimble` by default treats all variables as continuous, so we use the step function for  $\theta$  to make it discrete uniform in  $\{1, \dots, 111\}$  when deciding between  $\lambda_1$  and  $\lambda_2$  for the Poisson, and for the traceplot we use the `floor` function on  $\theta$ .

```
coalCode <- nimbleCode({
  alpha ~ dgamma(10, 10)
  lambda1 ~ dgamma(3, alpha)
  lambda2 ~ dgamma(3, alpha)
  theta ~ dunif(1, 112)

  for (i in 1:112){
    X[i] ~ dpois(lambda1 * step(theta - i) + lambda2 * (1-step(theta-i)))
  }
})
coalData <- list(X = coal[,2])
coalInits <- list(alpha = 1.1,
                 theta = 50,
                 lambda1 = 2.7,
                 lambda2 = 0.89)
coalModel <- nimbleModel(coalCode,
                        data = coalData,
                        inits = coalInits) # pump
CcoalModel <- compileNimble(coalModel)
```

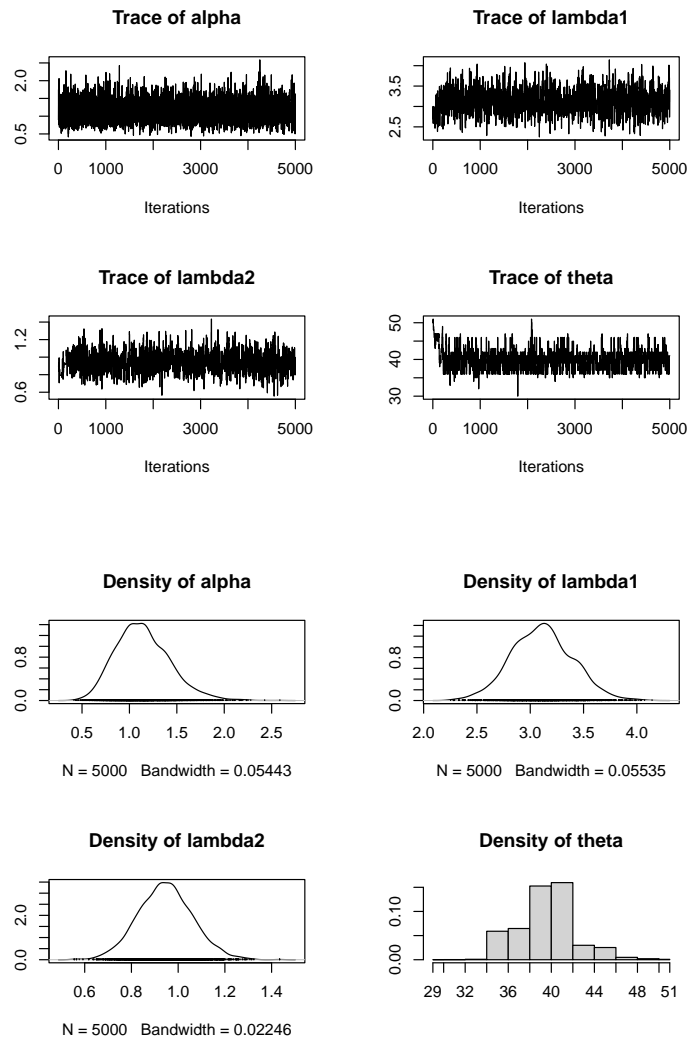


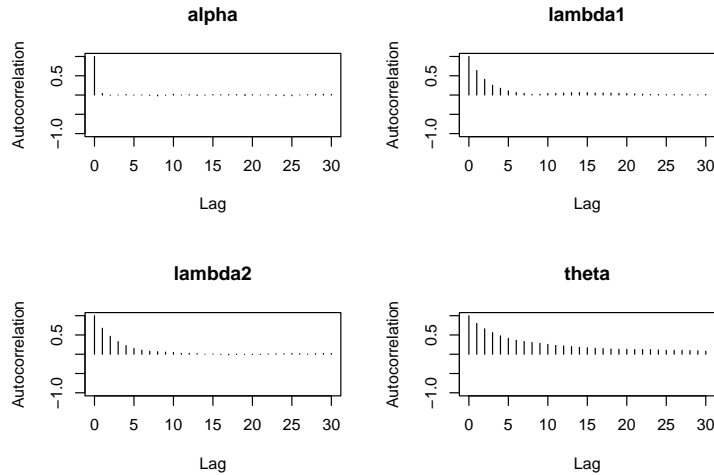
```

coalConf <- configureMCMC(coalModel)
coalConf$addMonitors(c("lambda1", "lambda2", "alpha", "theta"))
coalMCMC <- buildMCMC(coalConf)
CcoalMCMC <- compileNimble(coalMCMC, project = coalModel)
coalSamples.1 <- runMCMC(CcoalMCMC, niter = 5000)
coalSamples.2 <- runMCMC(CcoalMCMC, niter = 5000)
coalSamples.3 <- runMCMC(CcoalMCMC, niter = 5000)
coalSamples.4 <- runMCMC(CcoalMCMC, niter = 5000)

```

Below are traceplots, density plots, auto-correlation plots, and Gelman diagnostics for the chain run by `nimble` – the chains have converged.





```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## alpha          1      1.00
## lambda1         1      1.01
## lambda2         1      1.01
## theta          1      1.00
##
## Multivariate psrf
##
## 1.01
```

Based on the summary of the posterior distributions from the `nimble` MCMC below, we believe there is a 95% probability that:

- $\lambda_1$ , average number of yearly coal-mining disasters during the first period, is anywhere between 2.55 to 3.69
- $\lambda_2$ , average number of yearly coal-mining disasters during the second period, is anywhere between 0.73 to 1.18
- $\theta$  is anywhere between 36 to 46, i.e. the transition happened anywhere between the year of 1886 to 1896.

Note that again qualitatively, the results is very similar to the models we implemented by ourselves – estimates for  $\lambda_1$  being larger than  $\lambda_2$  suggests that the reduction in yearly coal-mining disasters; the estimates for  $\theta$  (change point) also being similar between the two models:

```
##
## Iterations = 1:5000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
```

```
##           Mean      SD Naive SE Time-series SE
## alpha    1.142 0.2851 0.004032      0.004198
## lambda1  3.111 0.2868 0.004056      0.009054
## lambda2  0.946 0.1165 0.001648      0.003819
## theta   40.031 2.6846 0.037966      0.145718
##
## 2. Quantiles for each variable:
##
##           2.5%      25%      50%      75%  97.5%
## alpha    0.6573  0.9419  1.1210  1.320  1.782
## lambda1  2.5649  2.9111  3.1060  3.298  3.678
## lambda2  0.7196  0.8655  0.9448  1.021  1.182
## theta   36.0000 38.7500 40.0000 41.000 46.000
```

Speed-wise, note that **our models** are of 2 orders of magnitude faster than the **nimble** model.

```
start.time <- Sys.time()
a <- runMCMC(CcoalMCMC, niter = 5000, progressBar = FALSE)
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

## Time difference of 0.120379 secs

```
start.time <- Sys.time()
b <- MCMC(5000, coal$V2, lambda1_0, lambda2_0, alpha_0, theta_0)
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

## Time difference of 0.001757145 secs

```
start.time <- Sys.time()
c <- MCMC_halfnorm(5000, coal$V2, lambda1_0, lambda2_0, 1.5, theta_0)
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```

## Time difference of 0.001721144 secs