Exercise 2

TMA4300 Computer Intensive Statistical Models

Mads Adrian Simonsen, William Scott Grundeland Olsen

19 mars, 2021

Problem A: The coal-mining disaster data

In this problem, we analyze a data set from (Jarrett 1979) on the dates of 191 explosions in coal mines in UK involving 10 or more fatalities, covering the period 15 March, 1851 to 22 March, 1962 inclusive.

Subproblem 1.

Figure 1 shows the 191 explosions over the period. It appears to be a steep linear trend from the beginning to about year 1890 where there is a sudden decrease in the rate. This suggests that there has been a major improvement in the safety regulations. By digging a little deeper, we find that according to (Mills 2010), in the late 1890s, all underground extractive industries in the UK, were subjected to varying degrees of regulation and control, and the inspectorate had greatly expanded in numbers, experience and authority.

```
data(coal, package = "boot")
ggplot(cbind(coal, explosions = 1:nrow(coal)), aes(date, explosions)) +
  geom_step(aes(color = "Cumulative explosions")) +
  theme_minimal() +
  theme(legend.position = "top", legend.title = element_blank())
```
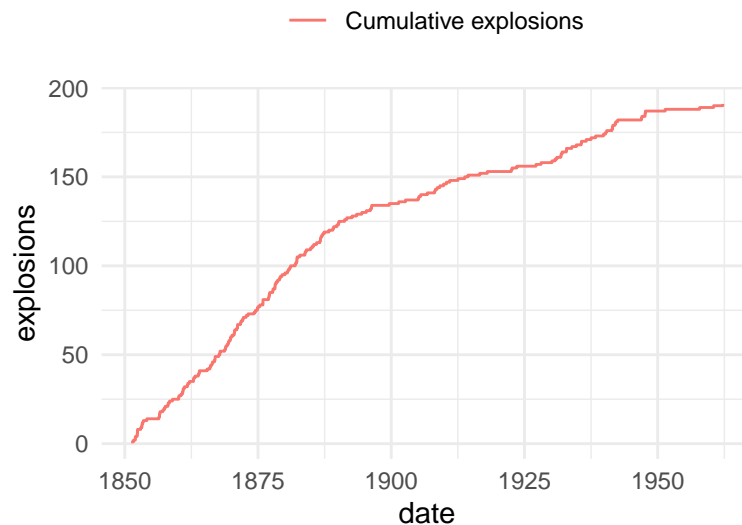

Figure 1: Cumulative plot of coal mines explosions from the period 15 March, 1851 to 22 March, 1962.

Subproblem 2.

We assume the coal-mining disasters follow an inhomogeneous Poisson process with intensity function $\lambda(t)$ (number of events per year). We assume $\lambda(t)$ to be piecewise constant with n breakpoints at times $t_k : k = 1, \ldots, n$. We let t_0 and t_{n+1} be the start and end times corresponding to 15 March, 1851 and 22 March, 1962, respectively, and let $t_0 < t_1 < t_2 < \ldots < t_n < t_{n+1}$. Then the intensity function becomes

$$\lambda(t) = \begin{cases} \lambda_{k-1}, & \text{if } t \in [t_{k-1}, t_k), \quad k = 1, \ldots, n, \\ \lambda_n, & \text{if } t \in [t_n, t_{n+1}]. \end{cases} \tag{1}$$

Furthermore, let x_k be the y_k-vector of the exact times when the disasters occurred in the time interval $[t_k, t_{k+1})$. We then let x be the y-vector of the exact times when the disasters occurred in the whole time interval $[t_0, t_{n+1}]$, where $y = \sum_{k=0}^{n} y_k$. We assume that the coal-mining disasters follow a hierarchical Bayesian structure, with

$$[y_k \mid \lambda_k, t_k, t_{k+1}] \sim \text{Poisson}(\lambda_k(t_{k+1} - t_k)), \quad k = 0, 1, \ldots, n,$$

$$[x_k \mid y_k, t_k, t_{k+1}] \sim \frac{y_k!}{(t_{k+1} - t_k)^{y_k}}, \quad k = 0, 1, \ldots, n,$$

$$\Rightarrow \quad [x_k, y_k \mid \lambda_k, t_k, t_{k+1}] \sim \lambda_k^{y_k} \exp(-\lambda_k(t_{k+1} - t_k)), \quad k = 0, 1, \ldots, n,$$

$$t_k \sim \text{Uniform}(t_{k-1}, t_{k+1}), \quad k = 1, 2, \ldots, n, \tag{2}$$

$$[\lambda_k \mid \beta] \sim \text{Gamma}(2, 1/\beta), \quad k = 0, 1, \ldots, n,$$

$$\beta \sim f_\beta(\beta) \propto \frac{\exp\left(-\frac{1}{\beta}\right)}{\beta}, \quad \beta > 0,$$

where $\lambda_0, \ldots, \lambda_n$ are conditionally independent of each other with respect to β and independent of $t_1, \ldots, t_n$, where the first and second argument in $\text{Gamma}(\cdot, \cdot)$ corresponds to the shape and rate parameter, respectively, and where $f_\beta(\beta)$ is an improper prior.

We will in this problem assume $n = 1$, such that the model parameters are $\theta = (t_1, \lambda_0, \lambda_1, \beta)$. The posterior distribution for θ given the observed data x is then given by

$$f_{\theta|x}(\theta \mid x) \propto f_{\theta,x,y}(\theta, x, y)$$

$$= f_{x,y|t_1,\lambda_0,\lambda_1}(x, y \mid t_1, \lambda_0, \lambda_1) \cdot f_{t_1}(t_1) \cdot f_{\lambda_0|\beta}(\lambda_0 \mid \beta) \cdot f_{\lambda_1|\beta}(\lambda_1 \mid \beta) \cdot f_\beta(\beta)$$

$$\propto \lambda_0^{y_0} \exp\{-\lambda_0(t_1 - t_0)\} \lambda_1^{y_1} \exp\{-\lambda_1(t_2 - t_1)\} \cdot \frac{1}{t_2 - t_0} \cdot \frac{\lambda_0}{\beta^2} e^{-\frac{\lambda_0}{\beta}} \cdot \frac{\lambda_1}{\beta^2} e^{-\frac{\lambda_1}{\beta}} \cdot \frac{\exp\left(-\frac{1}{\beta}\right)}{\beta}$$

$$\propto \frac{\lambda_0^{y_0+1} \lambda_1^{y_1+1}}{\beta^5} \exp\left\{-\left(\lambda_0(t_1 - t_0) + \lambda_1(t_2 - t_1) + \frac{1}{\beta}(\lambda_0 + \lambda_1 + 1)\right)\right\},$$

with $\quad \beta > 0, \; \lambda_0 \geq 0, \; \lambda_1 \geq 0 \quad \text{and} \quad t_0 \leq t_1 \leq t_2.$

That is, the posterior distribution is equal to

$$f_{\theta|x}(\theta \mid x) = \frac{1}{c} \cdot \frac{\lambda_0^{y_0+1} \lambda_1^{y_1+1}}{\beta^5} \exp\left\{-\left(\lambda_0(t_1 - t_0) + \lambda_1(t_2 - t_1) + \frac{1}{\beta}(\lambda_0 + \lambda_1 + 1)\right)\right\}, \tag{3}$$

where c is the normalizing constant.

Subproblem 3.

We calculate the full conditional for each of the elements in θ.

$$
\begin{aligned}
[t_1 \mid \lambda_0, \lambda_1, \beta, x] &\sim f_{t_1 \mid \lambda_0, \lambda_1, \beta, x}(t_1 \mid \lambda_0, \lambda_1, \beta, x) \\
&\propto f_{\theta \mid x}(\theta \mid x) \\
&\propto I_{(t_0, t_2)}(t_1) \cdot \lambda_0^{y_0+1} \lambda_1^{y_1+1} \exp\left\{-(\lambda_0 - \lambda_1)t_1\right\} \\
&\sim \text{Unknown distribution;} \\[6pt]
[\lambda_0 \mid t_1, \lambda_1, \beta, x] &\sim f_{\lambda_0 \mid t_1, \lambda_1, \beta, x}(\lambda_0 \mid t_1, \lambda_1, \beta, x) \\
&\propto f_{\theta \mid x}(\theta \mid x) \\
&\propto I_{[0,\infty)}(\lambda_0) \cdot \lambda_0^{(y_0+2)-1} \exp\left\{-\left(t_1 - t_0 + \frac{1}{\beta}\right)\lambda_0\right\} \\
&\sim \text{Gamma}\left(y_0 + 2, t_1 - t_0 + \frac{1}{\beta}\right); \\[6pt]
[\lambda_1 \mid t_1, \lambda_0, \beta, x] &\sim f_{\lambda_1 \mid t_1, \lambda_0, \beta, x}(\lambda_1 \mid t_1, \lambda_0, \beta, x) \\
&\propto I_{[0,\infty)}(\lambda_1) \cdot f_{\theta \mid x}(\theta \mid x) \\
&\propto \lambda_1^{(y_1+2)-1} \exp\left\{-\left(t_2 - t_1 + \frac{1}{\beta}\right)\lambda_1\right\} \\
&\sim \text{Gamma}\left(y_1 + 2, t_2 - t_1 + \frac{1}{\beta}\right); \\[6pt]
[\beta \mid t_1 \lambda_0, \lambda_1, x] &\sim f_{\beta \mid t_1 \lambda_0, \lambda_1, x}(\beta \mid t_1 \lambda_0, \lambda_1, x) \\
&\propto f_{\theta \mid x}(\theta \mid x) \\
&\propto I_{(0,\infty)}(\beta) \cdot \beta^{-4-1} \exp\left\{-\frac{\lambda_0 + \lambda_1 + 1}{\beta}\right\} \\
&\sim \text{Inv-Gamma}(4, \lambda_0 + \lambda_1 + 1).
\end{aligned}
$$

Subproblem 4.

We implement a single site McMC algorithm for $f(\theta \mid x)$, where we need to do a Gibbs step for all the full conditionals except for $[t_1 \mid \lambda_0, \lambda_1, \beta, x]$, where we will do a Metropolis Hastings step. We choose a random walk proposal

$$
[t_1^* \mid t_1^{(k)}] \sim \text{Normal}(t_1^{(k)}, d), \tag{4}
$$

where $t_1^{(k)}$ is the current state of the Markov chain, and d is a tuning parameter. We denote the proposal pdf by $Q(t_1^* \mid t_1^{(k)})$ and the target pdf by $\pi(t_1^{(k)}) = f_{t_1 \mid \lambda_0, \lambda_1, \beta, x}(t_1^{(k)} \mid \lambda_0, \lambda_1, \beta, x)$. The acceptance probability is then given by

$$
\alpha(t_1^* \mid t_1^{(k)}) = \min\left\{1, \frac{\pi(t_1^*) \cdot Q(t_1^{(k)} \mid t_1^*)}{\pi(t_1^{(k)}) \cdot Q(t_1^* \mid t_1^{(k)})}\right\}. \tag{5}
$$

Since the proposal distribution is a symmetric, we have that $Q(t_1^* \mid t_1^{(k)}) = Q(t_1^{(k)} \mid t_1^*)$. What is left to do, is to compute the ratio of the target distribution. First we notice that the change point t_1 directly affects the event counts y_0 and y_1. That is,

$$
y_0(t) = \sum_x I(x < t) \quad \text{and} \quad y_1(t) = y - t_0(t).
$$

Let $y_i^* = y_i(t_1^*)$ and $y_i^{(k)} = y_i(t_1^{(k)})$. We compute the ratio

$$\frac{\pi(t_1^*)}{\pi(t_1^{(k)})} = \exp\left\{\log\frac{\lambda_0^{y_0^*+1}\lambda_1^{y_1^*+1}e^{-(\lambda_0-\lambda_1)t_1^*}}{\lambda_0^{y_0^{(k)}+1}\lambda_1^{y_1^{(k)}+1}e^{-(\lambda_0-\lambda_1)t_1^k}}\right\}$$

$$= \exp\left\{(y_0^* - y_0^{(k)})\log\lambda_0 + (y_1^* - y_1^{(k)})\log\lambda_1 - (\lambda_0 - \lambda_1)(t_1^* - t_1^{(k)})\right\},$$

which we insert into (5) to get the acceptance probability.

```r
# Performs a Metropolis Hastings step to sample t_1
MH_t1 <- function(t_1, lambda_0, lambda_1, t_0, t_2, x, y_0, y_1, y, d) {
  accept <- 0
  t_star <- rnorm(1, mean = t_1, sd = d)   # Proposal
  if (t_0 < t_star & t_star < t_2) {
    y_0_star <- sum(x < t_star)  # Number of disasters before t_star
    y_1_star <- y - y_0_star     # Number of disasters after t_star
    alpha <- min(
      1,
      exp((y_0_star - y_0)*log(lambda_0) + (y_1_star - y_1)*log(lambda_1)
          - (lambda_0 - lambda_1)*(t_star - t_1))
    )
    if (runif(1) < alpha) {
      # Accept
      t_1 <- t_star
      y_0 <- y_0_star
      y_1 <- y_1_star
      accept <- 1
    }
  }
  list(t_1 = t_1, y_0 = y_0, y_1 = y_1, accept = accept)
}


# Samples from the posterior distribution using single site McMC
# data:     number of observations
# t_1:      initial changepoint
# lambda_0: initial rateparameter
# lambda_1: initial rateparameter
# beta:     initial hyperparameter
# d:        tuning parameter for the proposal distribution
# n_iter:   number of iterations
single_site_McMC <- function(data, t_1, lambda_0, lambda_1, beta, d = 1, n_iter = 5000) {
  x <- data$date
  y <- length(x)  # Number of disasters
  t_0 <- x[1]     # Start date
  t_2 <- x[y]     # End date
  theta <- matrix(nrow = n_iter, ncol = 4,
            dimnames = list(NULL, c("t_1", "lambda_0", "lambda_1", "beta"))
          )
  theta[1, ] <- c(t_1, lambda_0, lambda_1, beta)
  y_0 <- sum(x < t_1)  # Number of disasters before t_1
  y_1 <- y - y_0       # Number of disasters after t_1
  accept_count <- 0
  if (n_iter > 1) {
    for (i in 2:n_iter) {
      # Metropolis Hastings step
```

```
        param <- MH_t1(t_1, lambda_0, lambda_1, t_0, t_2, x, y_0, y_1, y, d)
        t_1 <- param$t_1
        y_0 <- param$y_0
        y_1 <- param$y_1
        accept_count <- accept_count + param$accept

        # Gibbs step, sample lambda_0
        lambda_0 <- rgamma(1,
                    shape = y_0 + 2,
                    rate  = t_1 - t_0 + 1/beta
                )

        # Gibbs step, sample lambda_1
        lambda_1 <- rgamma(1,
                    shape = y_1 + 2,
                    rate  = t_2 - t_1 + 1/beta
                )

        # Gibbs step, sample beta
        beta <- 1 / rgamma(1,
                    shape = 4,
                    rate  = lambda_0 + lambda_1 + 1
                )
        # Update
        theta[i, ] <- c(t_1, lambda_0, lambda_1, beta)
    }
  }
  print(sprintf("Acceptance rate for t_1: %.2f%%", 100*accept_count/(n_iter - 1)))
  as.data.frame(theta)
}
```

Subproblem 5.

```
set.seed(731)
# Trace plot function
# ...: Takes the chains as input
plot_trace <- function(...) {
  input_list <- list(...)
  n <- nrow(input_list[[1]])
  nchains <- length(input_list)
  bind_rows(..., .id = "chain") %>%
    mutate(n = rep(1:n, nchains)) %>%
    pivot_longer(c("t_1", "lambda_0", "lambda_1", "beta"), names_to = "parameter") %>%
    ggplot(aes(n, value, color = chain)) +
    geom_line() +
    facet_wrap(~ parameter, nrow = 4, scales = "free_y") +
    theme_minimal()
}

# Sample two chains from different starting values
n <- 5000
theta1 <- single_site_McMC(coal, coal$date[1], 0.1, 10, 20, d = 10, n_iter = n)
```

```
## [1] "Acceptance rate for t_1: 19.00%"
theta2 <- single_site_McMC(coal, coal$date[nrow(coal)], 30, 0.1, 2, d = 10, n_iter = n)
```

```
## [1] "Acceptance rate for t_1: 18.90%"
plot_trace(theta1, theta2)
```
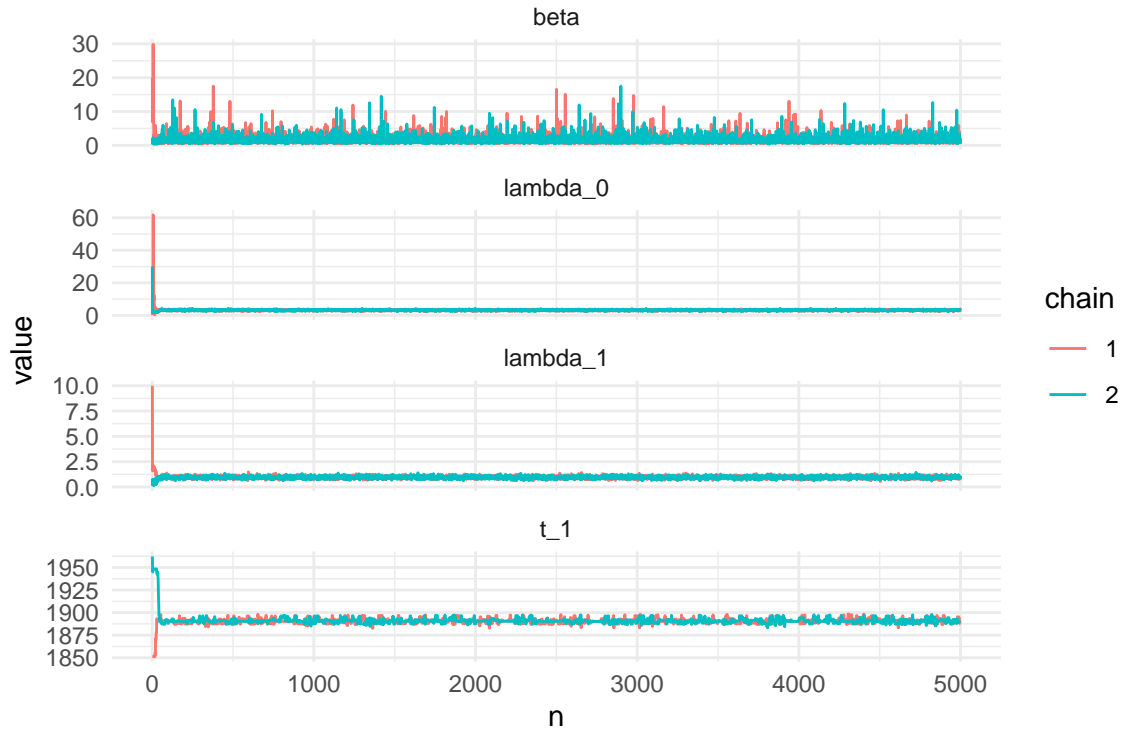

Figure 2: Trace plot of two chains from different starting values using a single site McMC.

A trace plot of two chains from different starting values is shown in Figure 2. The chain appears to converge instantly, with an acceptance rate just below 20%, which is a good rate as we want an acceptance rate somewhere between 20% and 50%. We let the burn-in period be the 300 first steps to be sure.

```
burnin <- 300
theta1 <- tail(theta1, -burnin)
theta2 <- tail(theta2, -burnin)

# Plot trace
plot_trace(theta1, theta2)
```

Figure 3 shows the same trace plot but now after discarding the burn-in period. The chain shows no trend. We only see a homogeneous band from each of the parameters. Looking specifically at the trace of t_1, it seems to be centered around 1890, which we anticipated by seeing a clear rate of change at the given point in Figure 1. We also see that t_1 mixes quite well in the limiting distribution, but not as good as the other parameters.

Subproblem 6.

The proposal distribution given by (4) has a tuning parameter d. In Figure 2 and 3, the tuning parameter was set to $d = 10$. We now try to set it at $d = 1$ and 200. The result is shown in Figure 4. For the parameters β and λ_1, it is hard to see any difference. Looking at the other parameters, t_1 and λ_0, we see that the chain
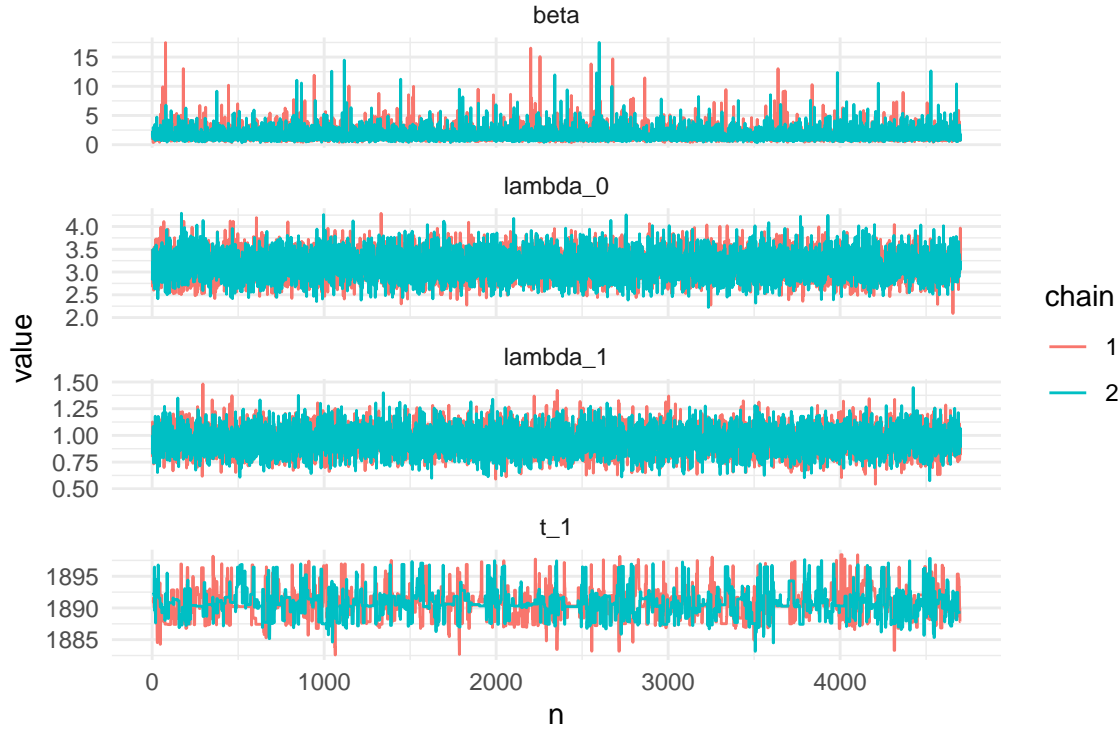
Figure 3: Trace plot of two chains from different starting values using a single site McMC after discarding the burn-in.

with tuning parameter $d = 200$ mixes almost instantly, but the chain with tuning parameter does not mix until 17500 steps. We also see that the acceptance rate is only 1% for the chain with parameter $d = 200$, while the chain with parameter $d = 1$ has an acceptance rate of 50%. After discarding the burn-in period, we can get a clearer picture of the behavior of the two chains in Figure 5. The chain with $d = 1$ takes very small frequent steps, and stays fixed around $t_1 = 1950$ for a long time before it moves to the neighborhood of $t_1 = 1890$. Looking at Figure 1, one can see a small change of rate, so it makes sense that the distribution of t_1 has a small mode at this position. The chain with $d = 200$ takes very big steps, and therefore gets rejected almost all the time.

```
set.seed(30)
n_iter = 20000
d <- c(1, 200)

# Sample two chains with different values for tuning parameter
theta3 <- single_site_McMC(coal, coal$date[nrow(coal)], 10, 10, 10,  d[1], n_iter)
```

```
## [1] "Acceptance rate for t_1: 50.10%"
```

```
theta4 <- single_site_McMC(coal, coal$date[nrow(coal)], 10, 10, 10,  d[2], n_iter)
```

```
## [1] "Acceptance rate for t_1: 1.00%"
```

```
# Plot trace
plot_trace(theta3, theta4) +
  scale_color_hue(name = "d", labels = d)
```

Figure 5 shows the same trace plot, but after discarding the burn-in period. We see that both of them has reached the limiting distribution, but they both yield a poor mixing, especially the chain with high tuning
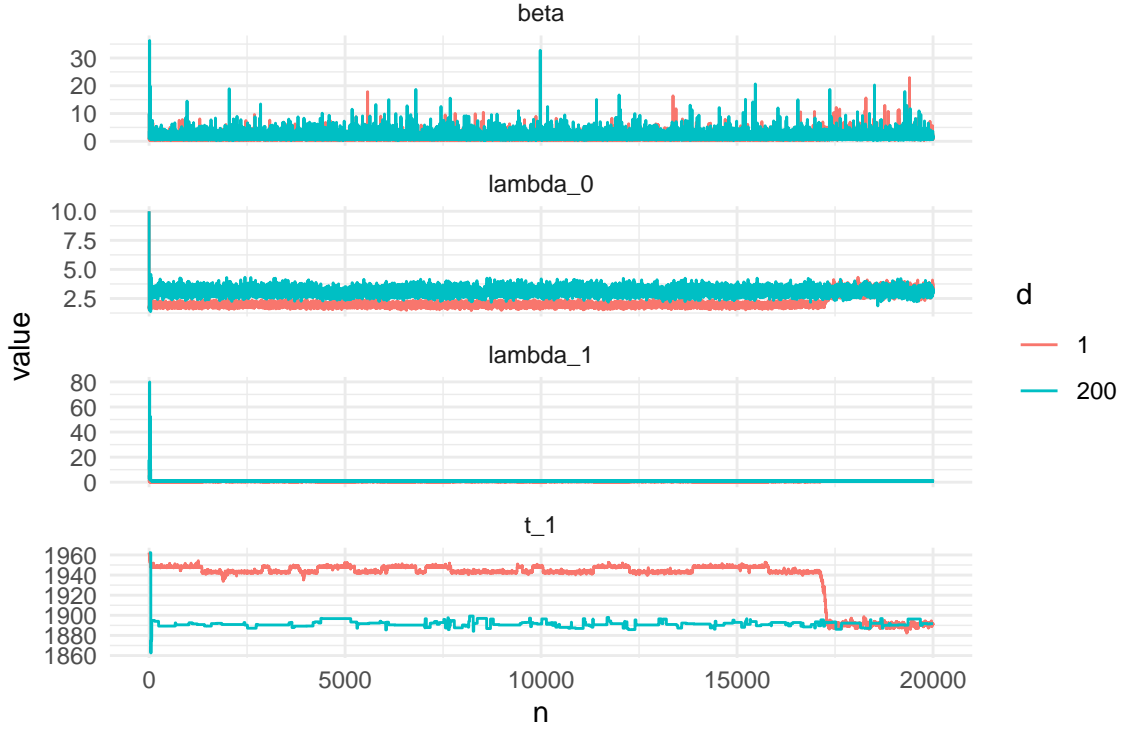
7

Figure 4: Trace plots of two chains with tuning parameters $d = 1$ and $d = 200$ using single-site McMC.

parameter.

```
burnin <- 17500
theta3 <- tail(theta3, -burnin)
theta4 <- tail(theta4, -burnin)

# Plot trace
plot_trace(theta3, theta4) +
  scale_color_hue(name = "d", labels = d)
```

Subproblem 7.

We will define and implement a block Metropolis-Hastings algorithm for $f_{\theta|x}(\theta \mid x)$ using two block proposals.

Subsubproblem (a)

We will use a block proposal $(t_1, \lambda_0, \lambda_1)$ keeping β unchanged. We generate a proposal $(t_1^*, \lambda_0^*, \lambda_1^*)$ by proposing t_1 as before according to (4). We then sample from the joint full conditional of $(\lambda_0^*, \lambda_1^*)$ including the new proposed value t_1^*. That is,

$$
\begin{aligned}
t_1^* &\sim \text{Normal}(t_1^{(k)}, d_t); \\
[\lambda_0^*, \lambda_1^* \mid t_1^*, \beta, x] &\sim f_{\lambda_0, \lambda_1 \mid t_1, \beta, x}(\lambda_0^*, \lambda_1^* \mid t_1^*, \beta, x) \\
&\propto f_{\lambda_0 \mid t_1, \beta, x}(\lambda_0^* \mid t_1^*, \beta, x) f_{\lambda_1 \mid t_1, \beta, x}(\lambda_1^* \mid t_1^*, \beta, x) \\
&= \text{Gamma}\left(\lambda_0; y_0^* + 2, t_1^* - t_0 + \frac{1}{\beta}\right) \text{Gamma}\left(\lambda_1; y_1^* + 2, t_2 - t_1^* + \frac{1}{\beta}\right),
\end{aligned}
$$

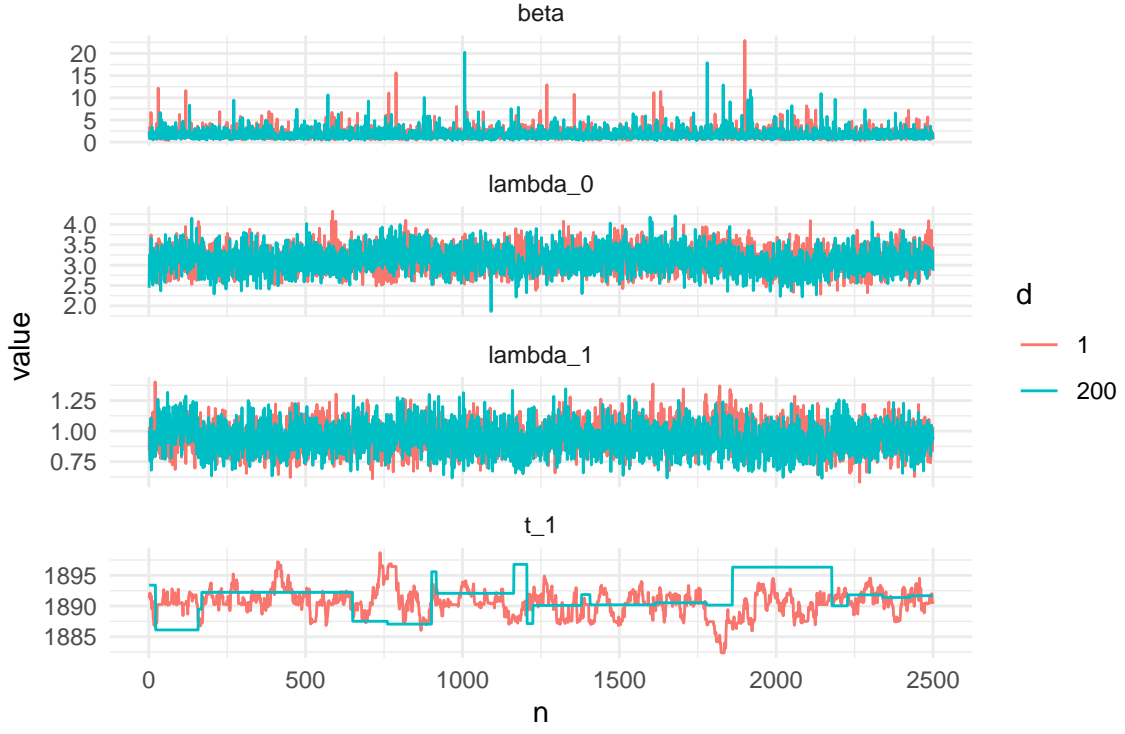where d_t is a tuning parameter. The target distribution is given by

Figure 5: Trace plots of two chains with tuning parameters $d = 1$ and $d = 200$ using single-site McMC after discarding the burn-in.

$$\pi(t_1, \lambda_0, \lambda_1) \propto f_{\theta|x}(\theta \mid x) \propto \lambda_0^{y_0+1} \lambda_1^{y_1+1} \exp\left\{-\left(\lambda_0\left(t_1 - t_0 + \frac{1}{\beta}\right) + \lambda_1\left(t_2 - t_1 + \frac{1}{\beta}\right)\right)\right\},$$

so the acceptance probability becomes

$$\alpha(t_1^*, \lambda_0^*, \lambda_1^* \mid t_1^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)})$$

$$= \min\left\{1, \frac{\pi(t_1^*, \lambda_0^*, \lambda_1^*) \cdot Q(t_1^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)} \mid t_1^*, \lambda_0^*, \lambda_1^*)}{\pi(t_1^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)}) \cdot Q(t_1^*, \lambda_0^*, \lambda_1^* \mid t_1^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)})}\right\}$$

$$= \min\left\{1, \frac{\pi(t_1^*, \lambda_0^*, \lambda_1^*) \cdot \mathrm{Gamma}\left(\lambda_0^{(k)}; y_0^* + 2, t_1^* - t_0 + \frac{1}{\beta}\right) \mathrm{Gamma}\left(\lambda_1^{(k)}; y_1^* + 2, t_2 - t_1^* + \frac{1}{\beta}\right)}{\pi(t_1^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)}) \cdot \mathrm{Gamma}\left(\lambda_0^*; y_0^{(k)} + 2, t_1^{(k)} - t_0 + \frac{1}{\beta}\right) \mathrm{Gamma}\left(\lambda_1^*; y_1^{(k)} + 2, t_2 - t_1^{(k)} + \frac{1}{\beta}\right)}\right\},$$

where we have omitted the Normal proposal for t_1^* because of the symmetry properties.

Subsubproblem (b)

The other block proposal is $(\beta, \lambda_0, \lambda_1)$ keeping t_1 unchanged. We generate a proposal $(\beta^*, \lambda_0^*, \lambda_1^*)$ by proposing β^* from a Normal distribution centered at the current state β, like we did previously with t_1. We then sample

9

from the joint full conditional of $(\lambda_0^*, \lambda_1^*)$ including the new proposed value β^*. That is,

$$\beta^* \sim \text{Normal}(\beta^{(k)}, d_\beta);$$

$$[\lambda_0^*, \lambda_1^* \mid t_1, \beta^*, x] \sim f_{\lambda_0, \lambda_1 \mid t_1, \beta, x}(\lambda_0^*, \lambda_1^* \mid t_1, \beta^*, x)$$

$$\propto f_{\lambda_0 \mid t_1, \beta, x}(\lambda_0^* \mid t_1, \beta^*, x) f_{\lambda_1 \mid t_1, \beta, x}(\lambda_1^* \mid t_1, \beta^*, x)$$

$$= \text{Gamma}\left(\lambda_0; y_0 + 2, t_1 - t_0 + \frac{1}{\beta^*}\right) \text{Gamma}\left(\lambda_1; y_1 + 2, t_2 - t_1 + \frac{1}{\beta^*}\right),$$

where d_β is a tuning parameter. The target distribution is given by

$$\pi(\beta, \lambda_0, \lambda_1) \propto f_{\theta \mid x}(\theta \mid x) \propto \frac{\lambda_0^{y_0+1} \lambda_1^{y_1+1}}{\beta^5} \exp\left\{-\left(\lambda_0(t_1 - t_0) + \lambda_1(t_2 - t_1) + \frac{1}{\beta}(\lambda_0 + \lambda_1 + 1)\right)\right\},$$

so the acceptance probability becomes

$$\alpha(\beta^*, \lambda_0^*, \lambda_1^* \mid \beta^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)})$$

$$= \min\left\{1, \frac{\pi(\beta^*, \lambda_0^*, \lambda_1^*) \cdot Q(\beta^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)} \mid \beta^*, \lambda_0^*, \lambda_1^*)}{\pi(\beta^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)}) \cdot Q(\beta^*, \lambda_0^*, \lambda_1^* \mid \beta^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)})}\right\}$$

$$= \min\left\{1, \frac{\pi(\beta^*, \lambda_0^*, \lambda_1^*) \cdot \text{Gamma}\left(\lambda_0^{(k)}; y_0 + 2, t_1 - t_0 + \frac{1}{\beta^*}\right) \text{Gamma}\left(\lambda_1^{(k)}; y_1 + 2, t_2 - t_1 + \frac{1}{\beta^*}\right)}{\pi(\beta^{(k)}, \lambda_0^{(k)}, \lambda_1^{(k)}) \cdot \text{Gamma}\left(\lambda_0^*; y_0 + 2, t_1 - t_0 + \frac{1}{\beta^{(k)}}\right) \text{Gamma}\left(\lambda_1^*; y_1 + 2, t_2 - t_1 + \frac{1}{\beta^{(k)}}\right)}\right\},$$

where we have omitted the Normal proposal for β^* because of the symmetry properties.

```r
# Computes the log target core of (t_1, lambda_0, lambda_1)
ltarget_1 <- function(t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2) {
  (y_0 + 1)*log(lambda_0) + (y_1 + 1)*log(lambda_1) -
    (lambda_0*(t_1 - t_0 + 1/beta) + lambda_1*(t_2 - t_1 + 1/beta))
}


# Computes the log target core of (beta, lambda_0, lambda_1)
ltarget_2 <- function(t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2) {
  (y_0 + 1)*log(lambda_0) + (y_1 + 1)*log(lambda_1) - 5*log(beta) -
    (lambda_0*(t_1 - t_0) + lambda_1*(t_2 - t_1) + 1/beta * (lambda_0 + lambda_1 + 1))
}


# Performs a Metropolis Hastings step to sample (t_1, lambda_0, lambda_1)
MH_t1_lamb0_lamb1 <- function(t_1, lambda_0, lambda_1, beta, y_0, y_1, y, x, t_0, t_2, d)
{
  accept <- 0
  t_star <- rnorm(1, mean = t_1, sd = d)   # Proposal
  if (t_0 < t_star & t_star < t_2) {
    y_0_star <- sum(x < t_star)  # Number of disasters before t_star
    y_1_star <- y - y_0_star      # Number of disasters after t_star
    # Proposal
    lambda_0_star <- rgamma(1,
                            shape = y_0_star + 2,
                            rate  = t_star - t_0 + 1/beta
    )
    # Proposal
    lambda_1_star <- rgamma(1,
                            shape = y_1_star + 2,
```

```r
                                rate  = t_2 - t_star + 1/beta
    )
    # Log target ratio
    ltarget_ratio <- ltarget_1(
      t_star, lambda_0_star, lambda_1_star, beta, y_0_star, y_1_star,
      t_0, t_2
    ) - ltarget_1(t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2)
    # Log proposal ratio
    lprop_ratio <- dgamma(lambda_0,
                          shape = y_0_star + 2,
                          rate  = t_star - t_0 + 1/beta,
                          log   = TRUE
    ) + dgamma(lambda_1,
               shape = y_1_star + 2,
               rate  = t_2 - t_star + 1/beta,
               log   = TRUE
    ) - dgamma(lambda_0,
               shape = y_0 + 2,
               rate  = t_1 - t_0 + 1/beta,
               log   = TRUE
    ) - dgamma(lambda_1,
               shape = y_1 + 2,
               rate  = t_2 - t_1 + 1/beta,
               log   = TRUE
    )
    alpha <- min(1, exp(ltarget_ratio + lprop_ratio))
    if (runif(1) < alpha) {
      # Accept proposals
      t_1 <- t_star
      y_0 <- y_0_star
      y_1 <- y_1_star
      lambda_0 <- lambda_0_star
      lambda_1 <- lambda_1_star
      accept <- 1
    }
  }
  list(t_1 = t_1, y_0 = y_0, y_1 = y_1, lambda_0 = lambda_0, lambda_1 = lambda_1,
       accept = accept)
}

# Performs a Metropolis Hastings step to sample (beta, lambda_0, lambda_1)
MH_beta_lamb0_lamb1 <- function(t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2, d) {
  accept <- 0
  beta_star <- rnorm(1, mean = beta, sd = d)
  if (beta_star > 0) {
    # Proposal
    lambda_0_star <- rgamma(1,
                            shape = y_0 + 2,
                            rate  = t_1 - t_0 + 1/beta_star
    )
    # Proposal
    lambda_1_star <- rgamma(1,
                            shape = y_1 + 2,
```

```r
                                     rate  = t_2 - t_1 + 1/beta_star
    )
    # Log target ratio
    ltarget_ratio <- ltarget_2(
      t_1, lambda_0_star, lambda_1_star, beta_star, y_0, y_1, t_0, t_2
    ) - ltarget_2(t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2)
    # Log proposal ratio
    lprop_ratio <- dgamma(lambda_0,
                          shape = y_0 + 2,
                          rate  = t_1 - t_0 + 1/beta_star,
                          log   = TRUE
    ) + dgamma(lambda_1,
               shape = y_1 + 2,
               rate  = t_2 - t_1 + 1/beta_star,
               log   = TRUE
    ) - dgamma(lambda_0,
               shape = y_0 + 2,
               rate  = t_1 - t_0 + 1/beta,
               log   = TRUE
    ) - dgamma(lambda_1,
               shape = y_1 + 2,
               rate  = t_2 - t_1 + 1/beta,
               log   = TRUE
    )
    alpha <- min(1, exp(ltarget_ratio + lprop_ratio))
    if (runif(1) < alpha) {
      # Accept proposals
      beta <- beta_star
      lambda_0 <- lambda_0_star
      lambda_1 <- lambda_1_star
      accept <- 1
    }
  }
  list(beta = beta, lambda_0 = lambda_0, lambda_1 = lambda_1, accept = accept)
}

# Samples from the posterior distribution using block McMC
# data:     number of observations
# t_1:      initial changepoint
# lambda_0: initial rateparameter
# lambda_1: initial rateparameter
# beta:     initial hyperparameter
# d:        tuning parameter for the proposal distribution
# n_iter:   number of iterations
block_McMC <- function(data, t_1, lambda_0, lambda_1, beta, d1 = 1, d2 = 1, n_iter = 5000)
  {
  x <- data$date
  y <- length(x)  # Number of disasters
  t_0 <- x[1]      # Start date
  t_2 <- x[y]      # End date
  theta <- matrix(nrow = n_iter, ncol = 4,
                  dimnames = list(NULL, c("t_1", "lambda_0", "lambda_1", "beta")))
  )
```

```r
    theta[1, ] <- c(t_1, lambda_0, lambda_1, beta)
  y_0 <- sum(x < t_1)  # Number of disasters before t_1
  y_1 <- y - y_0       # Number of disasters after t_1
  accept_count_1 <- 0
  accept_count_2 <- 0
  if (n_iter > 1) {
    for (i in 2:n_iter) {
      if (i %% 2 == 0) {
        # Block proposal for (t_1, lambda_0, lambda_1) keeping beta unchanged
        param <- MH_t1_lamb0_lamb1(
                   t_1, lambda_0, lambda_1, beta, y_0, y_1, y, x, t_0, t_2, d1
                 )
        t_1 <- param$t_1
        y_0 <- param$y_0
        y_1 <- param$y_1
        lambda_0 <- param$lambda_0
        lambda_1 <- param$lambda_1
        accept_count_1 <- accept_count_1 + param$accept
      } else {
        # Block proposal for (beta, lambda_0, lambda_1) keeping t_0 unchanged
        param2 <- MH_beta_lamb0_lamb1(
                    t_1, lambda_0, lambda_1, beta, y_0, y_1, t_0, t_2, d2
                  )
        beta <- param2$beta
        lambda_0 <- param2$lambda_0
        lambda_1 <- param2$lambda_1
        accept_count_2 <- accept_count_2 + param2$accept
      }
      theta[i, ] <- c(t_1, lambda_0, lambda_1, beta)  # Update
    }
  }
  print(sprintf("Acceptance rate for (t_1, lambda_0, lambda_1): %.2f%%",
                2*100*accept_count_1/(n_iter - 1)))
  print(sprintf("Acceptance rate for (beta, lambda_0, lambda_1): %.2f%%",
                2*100*accept_count_2/(n_iter - 1)))
  as.data.frame(theta)
}
```

Subproblem 8.

Figure 6 shows a nine chains with combinations of different tuning parameters for β in the first proposal block, and for t_1 in the second proposal block. We see that three of the chains did not converge. We will eliminate these, together with the chains that had an acceptance probability less than 15%, which leaves the second chain with $(d_t, t_\beta) = (10, 1)$ with a burn-in period of about $n = 500$.

```r
set.seed(93)
n_iter <- 5000
d1 <- c(1, 10, 200)
d2 <- c(1, 2, 10)
d_grid <- expand.grid(d1, d2)
lab <- with(d_grid, paste0("(", Var1, ", ", Var2, ")"))

# Sample 3 chains with different values for tuning parameter
theta5 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[1], d2[1], n_iter)
```

```
## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 37.09%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 36.53%"
theta6 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[2], d2[1], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 15.88%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 45.57%"
theta7 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[3], d2[1], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 0.56%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 43.37%"
theta8 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[1], d2[2], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 34.93%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 20.88%"
theta9 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[2], d2[2], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 12.84%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 27.93%"
theta10 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[3], d2[2], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 1.40%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 26.77%"
theta11 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[1], d2[3], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 33.53%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 5.44%"
theta12 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[2], d2[3], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 15.00%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 6.24%"
theta13 <- block_McMC(coal, coal$date[nrow(coal)], 10, 10, 10, d1[3], d2[3], n_iter)

## [1] "Acceptance rate for (t_1, lambda_0, lambda_1): 0.88%"
## [1] "Acceptance rate for (beta, lambda_0, lambda_1): 6.56%"
# Plot trace
plot_trace(theta5, theta6, theta7, theta8, theta9, theta10, theta11, theta12, theta13) +
  scale_color_hue(name = "(d_t, d_beta)", labels = lab)
```

We now compare the single-site chain with tuning parameter $d = 10$ and our candidate from the block proposal chain. We see that the single-site chain explores more of the domain. A possibility is that there is a computational error in the acceptance probability for the block proposal. However, they both yield a quite good mixing, but not perfect, as t_1 looks a bit sparse.

```
burnin <- 500
theta6 <- tail(theta6, -burnin)

theta1 <- tail(theta1, nrow(theta6))

# Plot trace
plot_trace(theta1, theta6) + scale_color_hue(labels = c("Single-site", "Block"))
```

We use the single-site chain to estimate the marginal posterior density of each of the variables. Figure 8 shows that the expected change point is at $t_1 = 1890.66$, where the disasters per year is reduced from $\lambda_0 = 3.14$ to
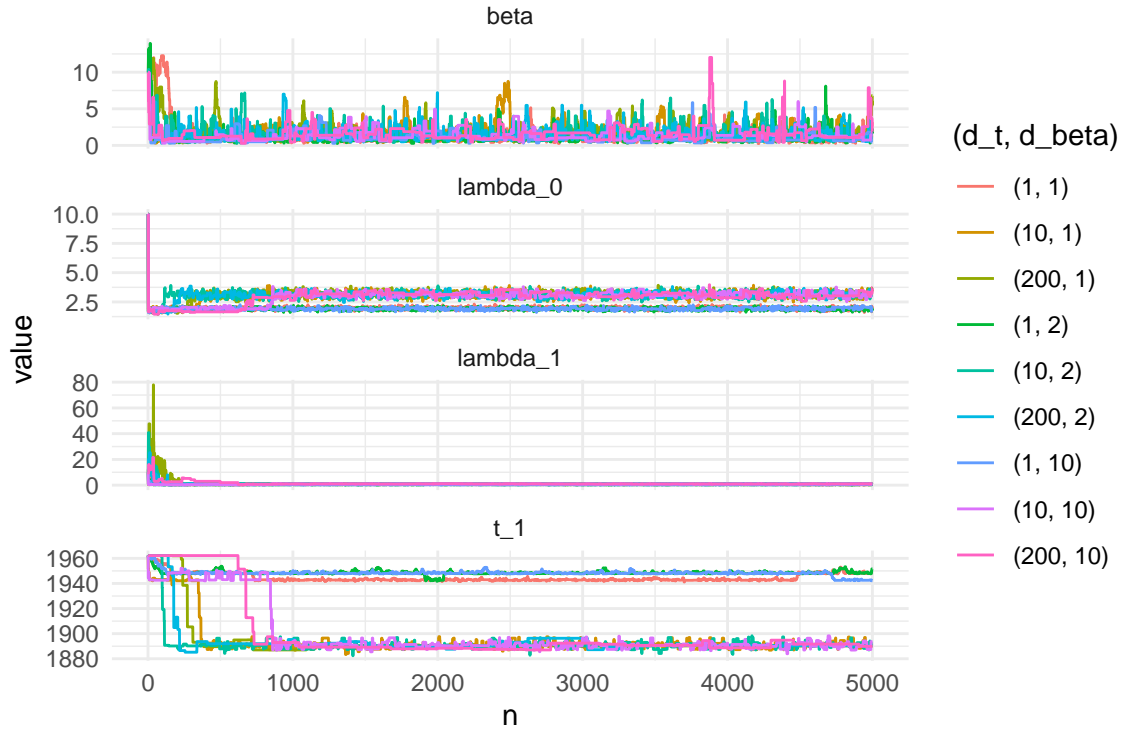
Figure 6: Trace plots of nine chains with blocking proposals with different tuning parameter values
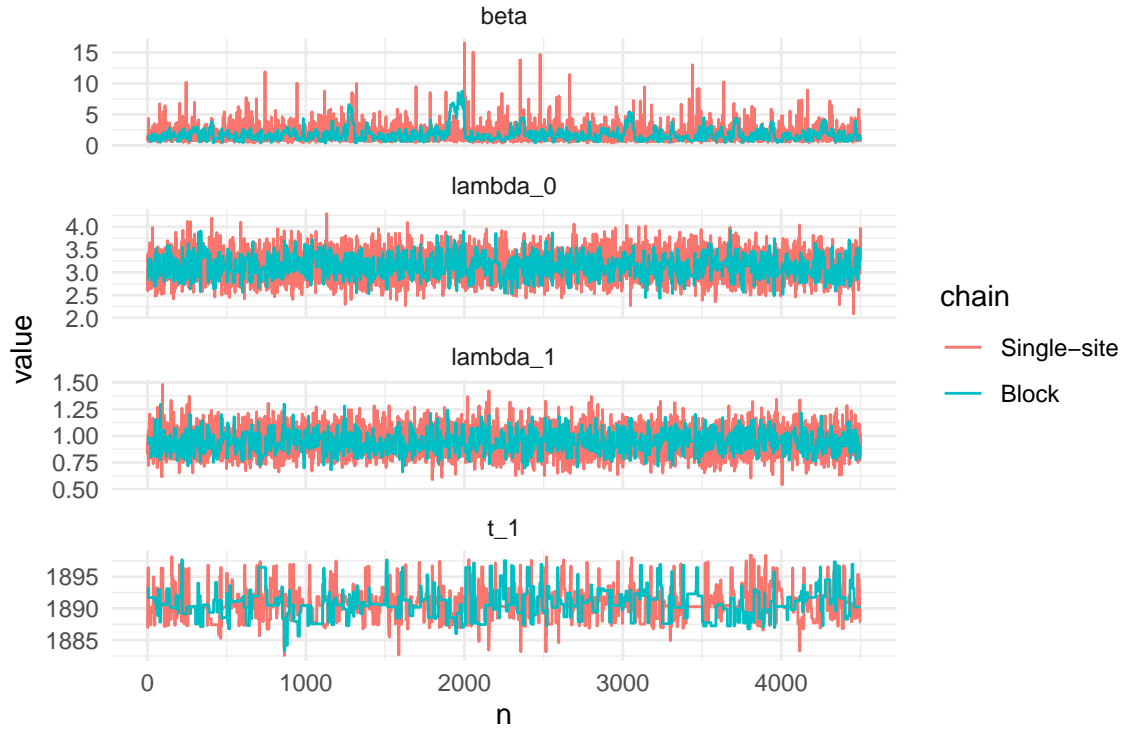

Figure 7: The best candidate of the chains with blocking proposals and the single-site proposal chain.

15

$\lambda_1 = 0.95$, which is a major improvement for the coal miners, although still not ideal.

```
theta1 %>%
  pivot_longer(everything(), names_to = "parameter") %>%
  group_by(parameter) %>%
  mutate(Expectation = mean(value)) %>%
  ggplot(aes(value)) +
  geom_density() +
  geom_vline(aes(xintercept = Expectation, color = factor(round(Expectation, 2)))) +
  facet_wrap(~ parameter, nrow = 2, scales = "free") +
  labs(color = "Expectation") +
  theme_minimal()
```
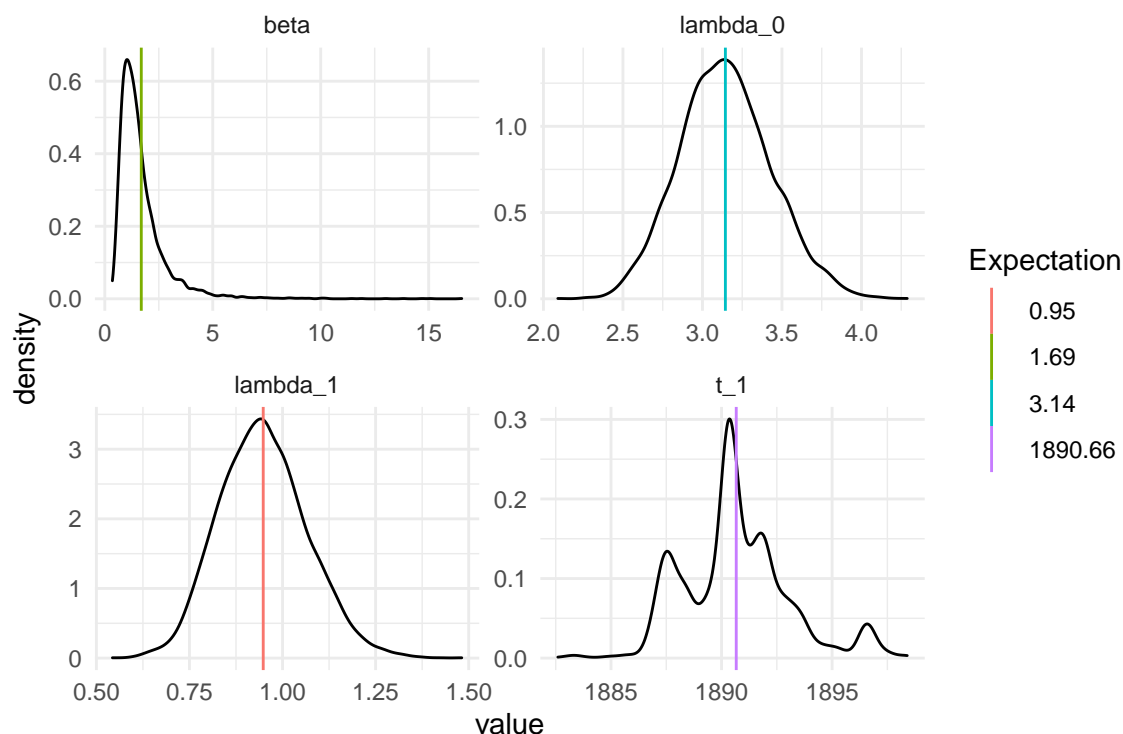

Figure 8: Estimated marginal posterior distributions from single-site McMC, with the empirical mean.

The scatter plot of $[\lambda_0, \lambda_1 \mid x]$ is shown in Figure 9 also showing $\text{Cov}[\lambda_0, \lambda_1 \mid x]$. We see that there is a very small correlation.

```
theta1 %>%
  ggplot(aes(lambda_0, lambda_1)) +
  geom_point() +
  labs(
    subtitle = sprintf("Cov(lambda_0, lambda_1 | x) = %.4f",
                       with(theta1, cov(lambda_0, lambda_1)))
  ) +
  theme_minimal()
```
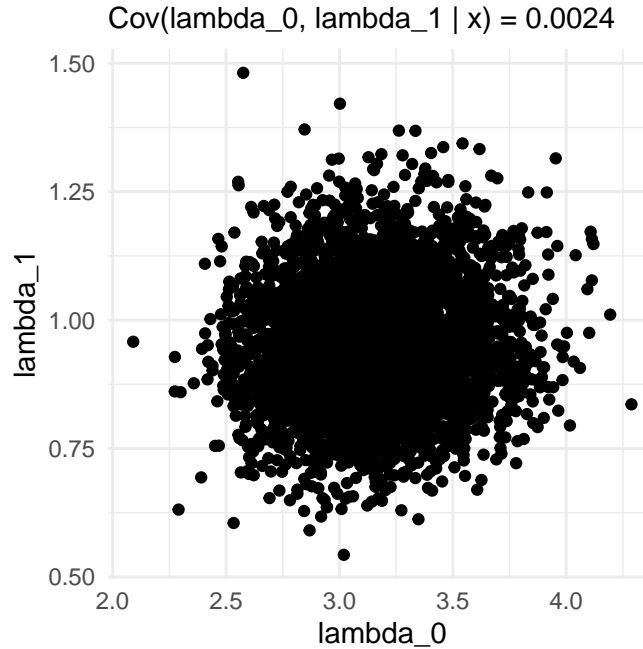
Figure 9: Scatter plot of $[\lambda_0 \mid x]$ and $[\lambda_1 \mid x]$ and their covariance.

Problem B: INLA for Gaussian Data

We plot the dataset in the following code block, and the result is shown in Figure 10.

```r
# Importing the data and making relevant variables:
path <- "https://www.math.ntnu.no/emner/TMA4300/2020v/Exercise/exercise2/Gaussiandata.txt"
gauss_data <- read.delim(path, header = FALSE)  # There is no header in the dataset
y <- gauss_data$V1      # Data values
TT <- length(y)         # End point of the data
t <- seq(1, TT, 1)      # A sequence from 1 to 20 for plotting

# Plotting:
ggplot(mapping = aes(t, y)) +
  geom_point() +
  ggtitle("Plot of the Gaussian data") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Subproblem 1.

We assume that, given $\boldsymbol{\eta} = \begin{bmatrix} \eta_1 & \cdots & \eta_T \end{bmatrix}^\top$, the observations y_t of the time series are independent and Gaussian distributed with mean η_t and known unit variance. Furthermore, the linear predictor η_t is linked to a smooth effect of time t as $\eta_t = f_t$, where the vector $\mathbf{f} = \begin{bmatrix} f_1 & \cdots & f_T \end{bmatrix}^\top$ is the latent field. Choosing a second order random walk model as a prior distribution for $\mathbf{f}$ we get that $\mathbf{f} \mid \theta \sim \text{Normal}(\mathbf{0}, Q(\theta)^{-1})$, for some precision matrix $Q(\theta)$. The model is completed by assigning the prior $\theta \sim \text{Gamma}(1, 1)$.

The class of latent Gaussian models can be represented by a hierarchical structure containing three stages. In
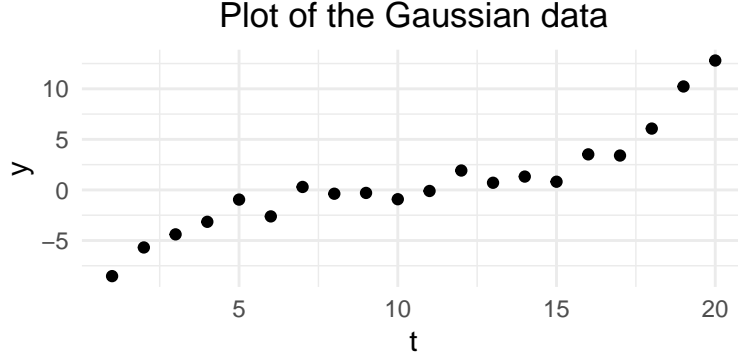
Figure 10: The dataset `Gaussiandata.txt` plotted.

the case of this model it can be written in the stages

$$\mathbf{y} \mid \mathbf{f} \sim \prod_{t=1}^{T} \pi(y_t \mid \eta_t), \quad \mathbf{f} \mid \theta \sim \text{Normal}(\mathbf{0}, Q(\theta)^{-1}) \quad \text{and} \quad \theta \sim \text{Gamma}(1,1),$$

where $Q(\theta)$ is the precision matrix, and $y_t \mid \eta_t \sim \text{Normal}(\eta_t, 1)$, for $t = 1, \ldots, T$. The field $\mathbf{f}$ is also a Gaussian Markov random field, as can be seen from its distribution and the sparseness discussed below. Thus, because the model here can be written in the above form, it is a latent Gaussian model.

INLA is applicable for use in latent Gaussian models under some constraints. Firstly, the hyperparameter vector cannot be to large, and in this model it is just a scalar, so this is satisfied. Secondly, the precision matrix needs to be sparse, which is the case for a second order random walk model. We look further at

$$\pi(\mathbf{f} \mid \theta) \propto \theta^{(T-2)/2} \exp\left[-\frac{\theta}{2} \sum_{t=3}^{T} (f_t - 2f_{t-1} + f_{t-2})^2\right] = \theta^{(T-2)/2} \exp\left[-\frac{1}{2} \mathbf{f}^\top Q(\theta) \mathbf{f}\right],$$

and want to determine $Q(\theta)$, so we look more at the exponent. We get that

$$\theta \sum_{t=3}^{T} (f_t - 2f_{t-1} + f_{t-2})^2 = \theta \sum_{t=3}^{T} (f_t^2 + 4f_{t-1}^2 + f_{t-2}^2 - 4f_t f_{t-1} + 2f_t f_{t-2} - 4f_{t-1} f_{t-2}) = \mathbf{f}^\top Q(\theta) \mathbf{f},$$

with the precision matrix

$$Q(\theta) = \theta \begin{bmatrix} 1 & -2 & 1 & & & & & & \\ -2 & 5 & -4 & 1 & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & 1 & -4 & 5 & -2 \\ & & & & & & 1 & -2 & 1 \end{bmatrix},$$

where the elements not filled in are zero, that is, a sparse matrix. There is also a multiplication, such that all elements gets a factor θ.

All the criteria to be able to use INLA is thus satisfied, and it can be applied to the model.

Subproblem 2.

We now want to define and implement a block Gibbs sampling algorithm for $f(\boldsymbol{\eta}, \theta \mid \mathbf{y})$ by proposing a new value for θ from the full conditional $\pi(\theta \mid \boldsymbol{\eta}, \mathbf{y})$, and proposing a new value for $\boldsymbol{\eta}$ from the full conditional $\pi(\boldsymbol{\eta} \mid \theta, \mathbf{y})$. Noting that $\mathbf{f} = \boldsymbol{\eta}$, we firstly have that

$$f(\boldsymbol{\eta}, \theta \mid \mathbf{y}) \propto \pi(\mathbf{y} \mid \boldsymbol{\eta}, \theta) \pi(\boldsymbol{\eta} \mid \theta) \pi(\theta)$$

$$\propto \left\{ \prod_{t=1}^{T} \exp\left[-\frac{1}{2}(y_t - \eta_t)^2 \right] \right\} \left\{ \theta^{(T-2)/2} \exp\left[-\frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 \right] \right\} \exp(-\theta)$$

$$= \theta^{(T-2)/2} \exp\left[-\theta - \frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 - \frac{1}{2} \sum_{t=1}^{T} (y_t - \eta_t)^2 \right],$$

and we can from this find the full conditionals. This gives us

$$\pi(\theta \mid \boldsymbol{\eta}, \mathbf{y}) \propto \theta^{(T-2)/2} \exp\left[-\theta - \frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 \right] = \theta^{(T-2)/2} \exp\left[-\theta - \frac{\theta}{2} \boldsymbol{\eta}^\top Q(1)\boldsymbol{\eta} \right],$$

and we see that

$$\theta \mid \boldsymbol{\eta}, \mathbf{y} \sim \text{Gamma}\left(\frac{T}{2}, 1 + \frac{1}{2}\boldsymbol{\eta}^\top Q(1)\boldsymbol{\eta} \right),$$

where the gamma distribution is parameterized with the shape and rate parameter. We can also find

$$\pi(\boldsymbol{\eta} \mid \theta, \mathbf{y}) \propto \exp\left[-\frac{\theta}{2} \sum_{t=3}^{T} (\eta_t - 2\eta_{t-1} + \eta_{t-2})^2 - \frac{1}{2} \sum_{t=1}^{T} (y_t - \eta_t)^2 \right]$$

$$= \exp\left[-\frac{1}{2}\boldsymbol{\eta}^\top Q(\theta)\boldsymbol{\eta} - \frac{1}{2}(\mathbf{y} - \boldsymbol{\eta})^\top (\mathbf{y} - \boldsymbol{\eta}) \right]$$

$$= \exp\left[-\frac{1}{2}(\boldsymbol{\eta}^\top Q(\theta)\boldsymbol{\eta} + \boldsymbol{\eta}^\top \boldsymbol{\eta} + \mathbf{y}^\top \mathbf{y}) + \mathbf{y}^\top \boldsymbol{\eta} \right]$$

$$\propto \exp\left[\mathbf{y}^\top \boldsymbol{\eta} - \frac{1}{2}\boldsymbol{\eta}^\top (Q(\theta) + I)\boldsymbol{\eta} \right],$$

where $I \in \mathbb{R}^{T \times T}$ is the identity matrix. It is then clear that

$$\boldsymbol{\eta} \mid \theta, \mathbf{y} \sim \text{Normal}((Q(\theta) + I)^{-1}\mathbf{y}, (Q(\theta) + I)^{-1}),$$

because we recognize the canonical form of the Gaussian.

We can then create the matrix $Q(\theta)$ as `Q(TT, theta)` in R, done in the following code block. Here we use that $Q(\theta) = \theta M^\top M$ for the matrix M coming from the fact that we have $\sum_{t=3}^{T} (f_t - 2f_{t-1} + f_{t-2})^2$.

```
Q_matrix <- function(TT, theta) {
  M <- matrix(0, nrow = TT, ncol = TT)  # Matrix of zeros of size TT times TT
  for (i in 3:TT) { # Create M corresponding to RW2
    M[i, c(i-2, i-1, i)] <- c(1, -2, 1)
  }
  return(theta * t(M) %*% M)  # Q(theta) = theta * M^T * M
}
```

We can then create a function to sample using the block updating using Gibbs sampling. This is done in the following code block.

```
block_Gibbs <- function(n, theta0, eta0, y) {
  TT <- length(eta0)
  theta <- rep(0, n)  # Initialize vector to store theta
```

```
  eta <- matrix(0, nrow = n, ncol = TT)   # Initialize matrix to store eta (row-wise)
  theta[1] <- theta0   # Initial value
  eta[1, ] <- eta0     # Initial value
  Q1 <- Q_matrix(TT, 1)   # Q(1)
  for (i in 2:n) {
    # Using old eta to find new theta from gamma distribution:
    theta[i] <- rgamma(1, shape = TT/2, rate = 1 + 1/2 * eta[i-1, ] %*% Q1 %*% eta[i-1, ])
    Qt <- Q_matrix(TT, theta[i])    # Using new theta to find Q(theta)
    inv <- solve(Qt + diag(TT))     # (Q(theta) + I)^(-1)
    eta[i, ] <- rmvnorm(1, inv %*% y, inv)   # Drawing from multivariate normal
  }
  return(list("theta" = theta, "eta" = eta))
}
```

Now we may run the implementation done in the following code block. Here we initialize $\theta = 1$ and $\boldsymbol{\eta} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\top$.

```
set.seed(420)
n <- 5000
theta0 <- 1          # Initial value
eta0 <- rep(1, TT)   # Initial value

Gibbs <- block_Gibbs(n, theta0, eta0, y)
theta <- Gibbs$theta     # theta values from Gibbs sampling

# Plotting:
as.data.frame(theta) %>%
  ggplot(aes(x = seq(1, n, 1), y = theta)) +
  geom_line() +
  ggtitle("Traceplot of theta") +
  xlab("Iteration") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

From Figure 11 the burn-in period seems to be no more than 50, but to be conservative, we disregard the 100 first iterations. Knowing this, we discard the first 100 iterations. This is done in the following code block.

```
thetaGibbs <- tail(theta, -100)  # Not containing the first 100 values

# Plotting:
as.data.frame(thetaGibbs) %>%
  ggplot(aes(x = thetaGibbs, y = ..density..)) +
  geom_histogram(binwidth = 0.1, boundary = 0) +
  ggtitle("Histogram of the theta samples") +
  xlab("theta") +
  ylab("Density") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Figure 12 shows a simulated histogram of the θ samples from the block Gibbs sampler. This is an approximation of the posterior marginal for the hyperparameter, $\pi(\theta \mid \mathbf{y})$.

We can now calculate the mean and 95% confidence interval of the smooth effect. This is done in the following code block, and the result is shown in Figure . From this we see that almost all the data points are within the confidence interval.
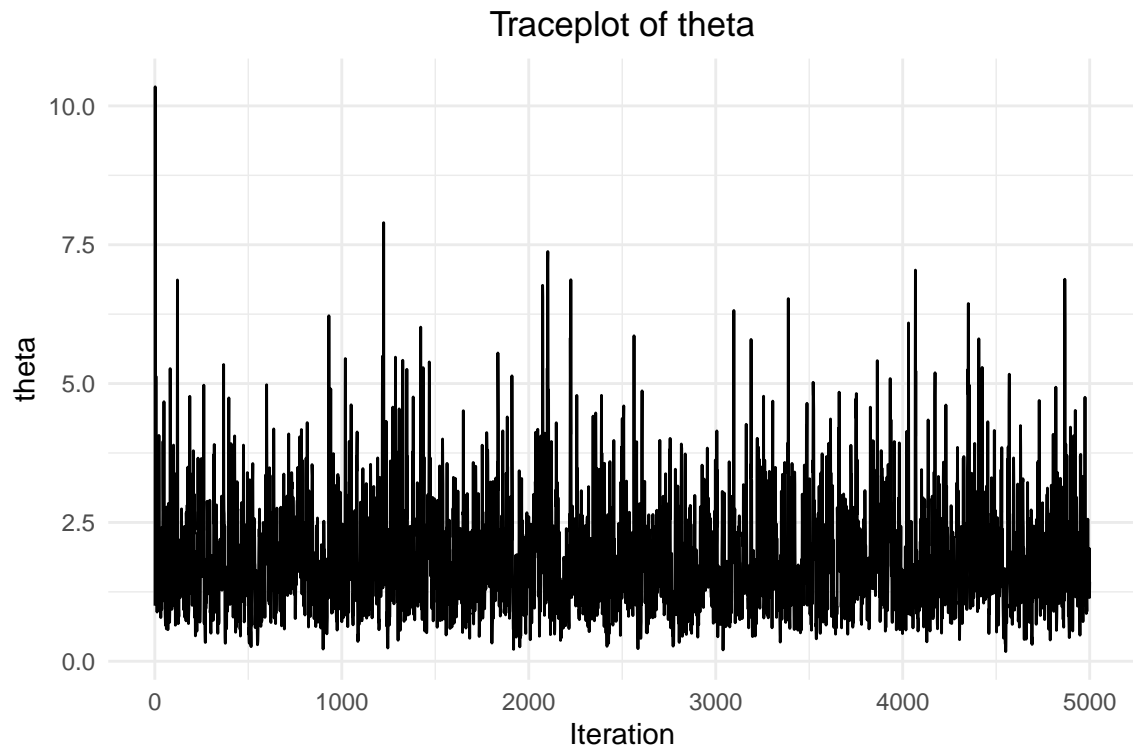
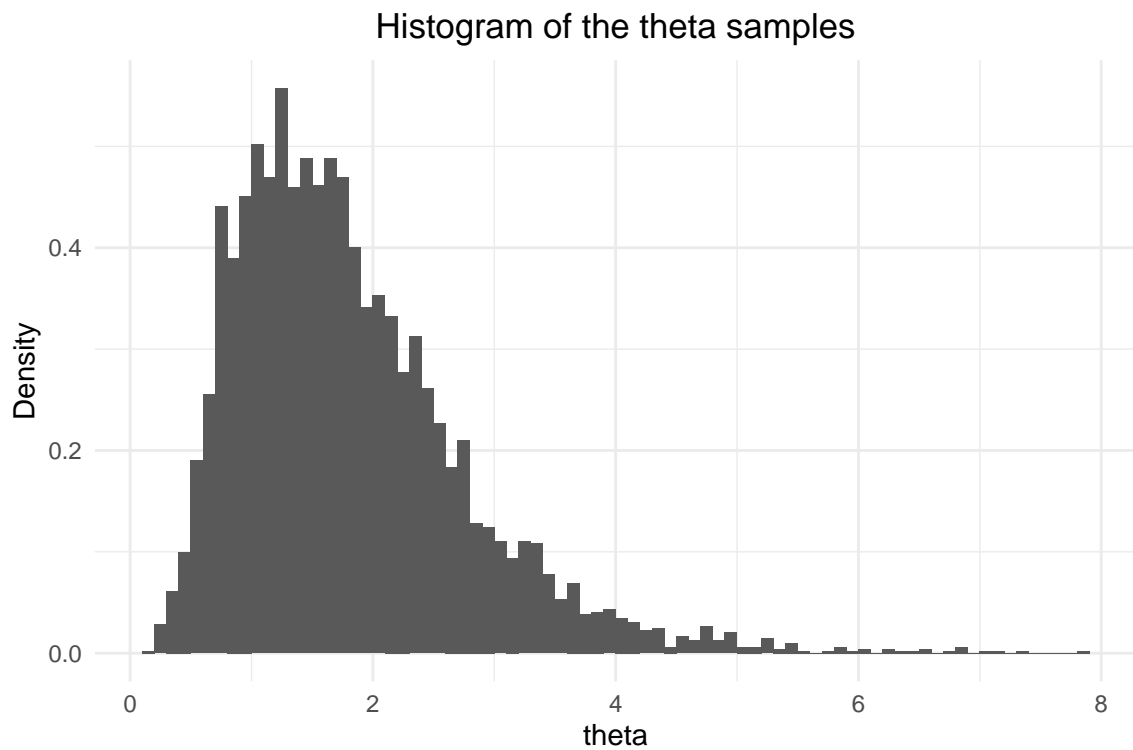Figure 11: The traceplot of θ giving an indication of the burn-in period.

Figure 12: Histogram of the θ samples.

21

```r
eta <- tail(Gibbs$eta, -100)   # eta values from Gibbs sampler omitting the first 100
eta_mean <- colMeans(eta) # Column-wise mean of eta
eta_var <- colVars(eta)    # Column-wise variance of eta

z <- qnorm(0.025)    # z-value for 95% confidence interval
CI_upp <- eta_mean + z * sqrt(eta_var)  # Upper bound of confidence interval
CI_low <- eta_mean - z * sqrt(eta_var)  # Lower bound of confidence interval

# Plotting:
ggplot() +
  geom_line(aes(x = t,
                y = eta_mean,
                col = "Mean")
            ) +
  geom_point(aes(x = t,
                 y = y,
                 col = "Observed data")
             ) +
  geom_line(aes(x = t,
                y = CI_upp,
                col = "95% confidence interval")
            ) +
  geom_line(aes(x = t,
                y = CI_low,
                col = "95% confidence interval")
            ) +
  geom_ribbon(aes(x = t,
                  y = y,
                  ymin = CI_low,
                  ymax = CI_upp),
              fill = "red",
              alpha = 0.05
              ) +
  ylab("y") +
  ggtitle("Mean and confidence interval for the smooth effect") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```
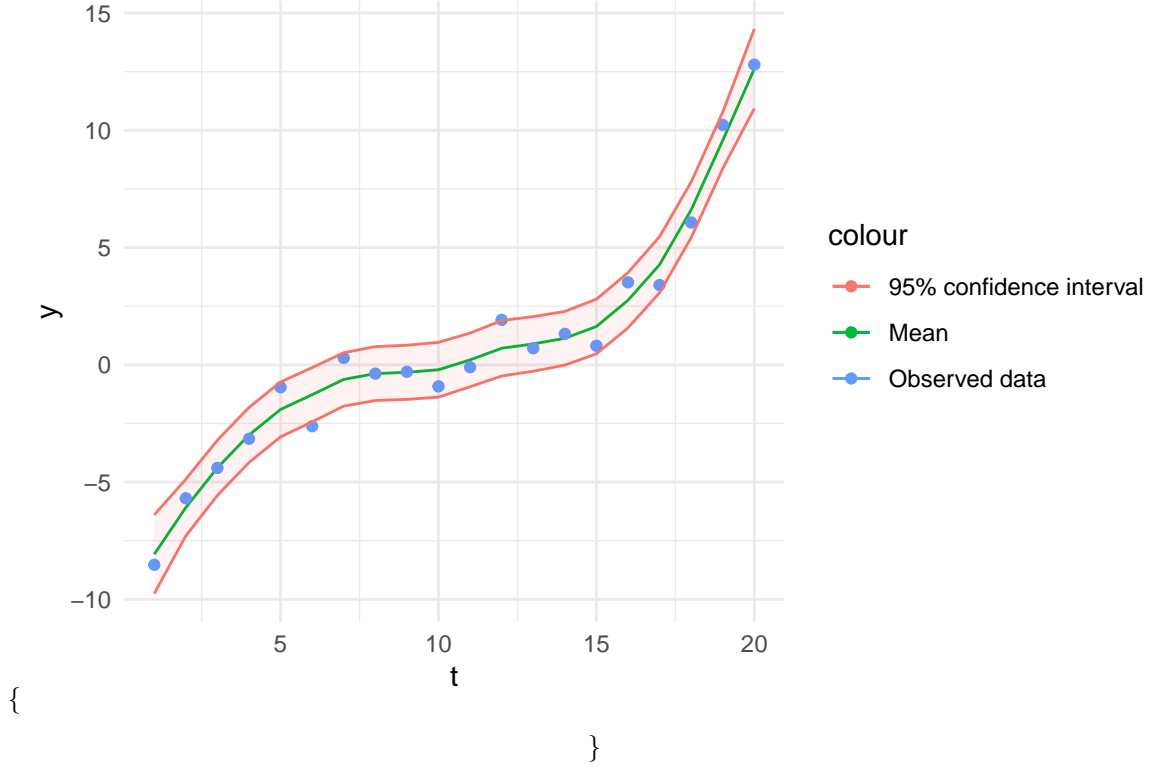
\begin{figure}

Mean and confidence interval for the smooth effect

{

}

\caption{The mean and the 95% confidence interval for the smooth effect.} \end{figure}

Subproblem 3.

We now want to approximate the posterior marginal for the hyperparameter using INLA. We have that

$$\pi(\theta \mid \mathbf{y}) \propto \frac{\pi(\mathbf{y} \mid \boldsymbol{\eta}, \theta)\pi(\boldsymbol{\eta} \mid \theta)\pi(\theta)}{\pi(\boldsymbol{\eta} \mid \theta, \mathbf{y})},$$

where we previously found that $\boldsymbol{\eta} \mid \theta, \mathbf{y} \sim \text{Normal}((Q(\theta) + I)^{-1}\mathbf{y}, (Q(\theta) + I)^{-1})$, and also the other quantities are known up to a proportionality constant. We thus have that

$$\pi(\theta \mid \mathbf{y}) \propto \frac{\exp\left[-\frac{1}{2}(\boldsymbol{\eta} - \mathbf{y})^\top(\boldsymbol{\eta} - \mathbf{y})\right]\theta^{(T-2)/2}\exp\left[-\frac{1}{2}\boldsymbol{\eta}^\top Q(\theta)\boldsymbol{\eta}\right]\exp(-\theta)}{\det(Q(\theta) + I)^{1/2}\exp\left\{-\frac{1}{2}[\boldsymbol{\eta} - (Q(\theta) + I)^{-1}\mathbf{y}]^\top(Q(\theta) + I)[\boldsymbol{\eta} - (Q(\theta) + I)^{-1}\mathbf{y}]\right\}}$$

$$= \theta^{(T-2)/2}\det(Q(\theta) + I)^{-1/2}$$

$$\cdot \exp\left\{-\theta - \frac{1}{2}(\boldsymbol{\eta} - \mathbf{y})^\top(\boldsymbol{\eta} - \mathbf{y}) - \frac{1}{2}\boldsymbol{\eta}^\top Q(\theta)\boldsymbol{\eta} + \frac{1}{2}[\boldsymbol{\eta} - (Q(\theta) + I)^{-1}\mathbf{y}]^\top(Q(\theta) + I)[\boldsymbol{\eta} - (Q(\theta) + I)^{-1}\mathbf{y}]\right\}$$

$$\propto \theta^{(T-2)/2}\det(Q(\theta) + I)^{-1/2}\exp\left[-\theta - \frac{1}{2}\mathbf{y}^\top\mathbf{y} + \frac{1}{2}\mathbf{y}^\top(Q(\theta) + I)^{-1}\mathbf{y}\right]$$

$$= \theta^{(T-2)/2}\det(Q(\theta) + I)^{-1/2}\exp\left\{-\theta - \frac{1}{2}\mathbf{y}^\top[I - (Q(\theta) + I)^{-1}]\mathbf{y}\right\},$$

where we have used that $(Q(\theta) + I)^\top = Q(\theta)^\top + I^\top = Q(\theta) + I$, because $Q(\theta)$ is symmetric. We can then based on this simulate the posterior marginal for the hyperparameter as done in the following code block.

```
post_marg_theta <- function(thetas, y) {
  post <- rep(0, length(thetas))
  TT <- length(y)
```

```
  for (i in 1:length(post)) {
    theta <- thetas[i]
    Q <- Q_matrix(TT, theta)
    QpI <- Q + diag(TT)
    post[i] <- theta^(TT/2-1) * det(QpI)^(-1/2) *
               exp(-theta - 1/2 * t(y) %*% (diag(TT) - solve(QpI)) %*% y)
  }
  return(post)
}
```

To test this we make a grid of θ's from 0 to 6 and find $\pi(\theta \mid \mathbf{y})$ on this grid. We could just plot this and see that the form is the same as the one we obtained in Figure 12. However, for this simple example it is easy to find the normalizing constant by integration. Doing this, and plotting together with the previously found histogram, we get Figure 13. We see that there is a strong correspondence between the two.

```
set.seed(420)

# Using INLA scheme:
thetas <- seq(0, 6, 0.01)    # Grid of thetas
post <- post_marg_theta(thetas, y)   # Un-normalized posterior marginal

k <- integrate(function(theta)(post_marg_theta(theta, y)),
               lower = 0,
               upper = Inf)$value
post <- post / k    # Normalized posterior marginal

# Plotting:
ggplot() +
  geom_histogram(aes(x = thetaGibbs,
                     y = ..density..),
                 binwidth = 0.1,
                 boundary = 0
                 ) +
  geom_line(aes(x = thetas,
                y = post,
                col = "Using INLA\nscheme")
            ) +
  xlab("theta") +
  ylab("pi(theta | y)") +
  ggtitle("Posterior marginal for the hyperparameter") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Subproblem 4.

In this problem we want to approximate the marginal posterior of the smooth effect given by

$$\pi(\eta_t \mid \mathbf{y}) = \int_{\mathbb{R}} \pi(\eta_t \mid \mathbf{y}, \theta)\pi(\theta \mid \mathbf{y}) \, d\theta,$$

for $t = 1, \ldots, T$. We are interested in the marginal posterior of the smooth effect for $t = 10$, that is $\pi(\eta_{10} \mid \mathbf{y})$, and we approximate this as

$$\pi(\eta_{10} \mid \mathbf{y}) \approx \Delta \sum_{\theta \in \Theta} \pi(\eta_{10} \mid \mathbf{y}, \theta)\pi(\theta \mid \mathbf{y}).$$
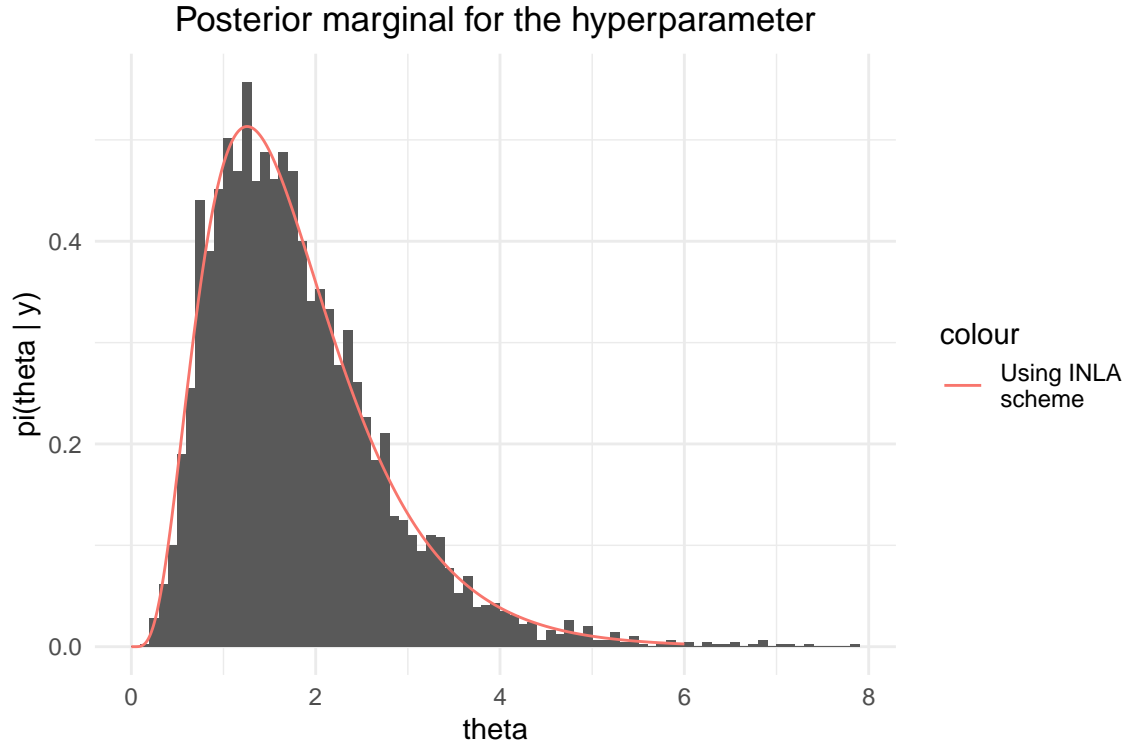
24

Figure 13: Posterior marginal for the hyperparameter both using block Gibbs sampling and INLA.

Here we denote the grid of θ values by Θ, and the step size, which is in our case equal for each grid-point, by Δ.

In the following code block we first make functions for $\pi(\eta_t \mid \mathbf{y}, \theta)$ and $\pi(\eta_t \mid \mathbf{y})$, before we make grids and plot the result. We plot it together with the result from the block Gibbs sampling, and again we normalize the result from the INLA scheme, as it is simple to do in our case. This is shown in Figure 14, and we see that the two posterior marginals correspond well.

```r
# Using INLA scheme:
pi_eta_given_theta_y <- function(y, theta, etas, t) {    # pi(eta_t | theta, y)
  TT <- length(y)
  Q <- Q_matrix(TT, theta)
  inv <- solve(Q + diag(TT))
  # eta_t | theta, y ~ Normal(inv * y, inv):
  normal <- dnorm(etas, mean = (inv %*% y)[t], sd = sqrt(inv[t, t]))
  return(normal)
}

post_marg_eta <- function(y, thetas, etas, t) {    # pi(eta_t | y)
  res <- rep(0, length(etas))    # Initializing result to zero
  Delta <- thetas[2] - thetas[1]    # Step size
  post_theta <- post_marg_theta(thetas, y)    # pi(theta | y)
  for (j in 1:length(thetas)) {    # Approximation of the integral
    res <- res + Delta * pi_eta_given_theta_y(y, thetas[j], etas, t) * post_theta[j]
  }
  return(res)
}
```

```
# Un-normalized:
etas <- seq(-3, 3, 0.01)     # Grid of etas
thetas <- seq(0, 6, 0.01)    # Grid of thetas
eta10 <- post_marg_eta(y, thetas, etas, 10)

# Normalizing:
k <- integrate(function(etas)(post_marg_eta(y, thetas, etas, 10)),
               lower = -Inf,
               upper = Inf)$value
eta10 <- eta10 / k

# From block Gibbs sampling:
eta10Gibbs <- eta[, 10]

# Plotting:
ggplot() +
  geom_histogram(mapping = aes(x = eta10Gibbs, y = ..density..),
                 binwidth = 0.1,
                 boundary = 0
                 ) +
  geom_line(aes(x = etas,
                y = eta10,
                col = "Using INLA\nscheme")
            ) +
  xlab("eta10") +
  ylab("pi(eta10 | y)") +
  ggtitle("Marginal posterior for the smooth effect for t = 10") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Subproblem 5.

We are now going to use the `R-INLA` library to do what we have done previously in this problem. We first perform the calculations using the `inla()` function in the following code block.

```
df <- data.frame(y = y, t = t)
hyper <- list(theta = list(prior = "loggamma", param = c(1, 1)))
formula <- y ~ f(t, model = "rw2", hyper = hyper, constr = FALSE) - 1
res <- inla(formula = formula,
            family = "gaussian",
            data = df,
            control.family = list(hyper = list(prec = list(initial = 0, fixed = TRUE))))
```

Firstly, we may then show the posterior marginal for the hyperparameter, as shown in Figure 15. We see that there indeed is a close correspondence between the methods used.

```
ggplot() +
  geom_histogram(aes(x = thetaGibbs,
                     y = ..density..),
                 binwidth = 0.1,
                 boundary = 0
                 ) +
  geom_line(aes(x = res$marginals.hyperpar$`Precision for t`[, 1],
                y = res$marginals.hyperpar$`Precision for t`[, 2],
```
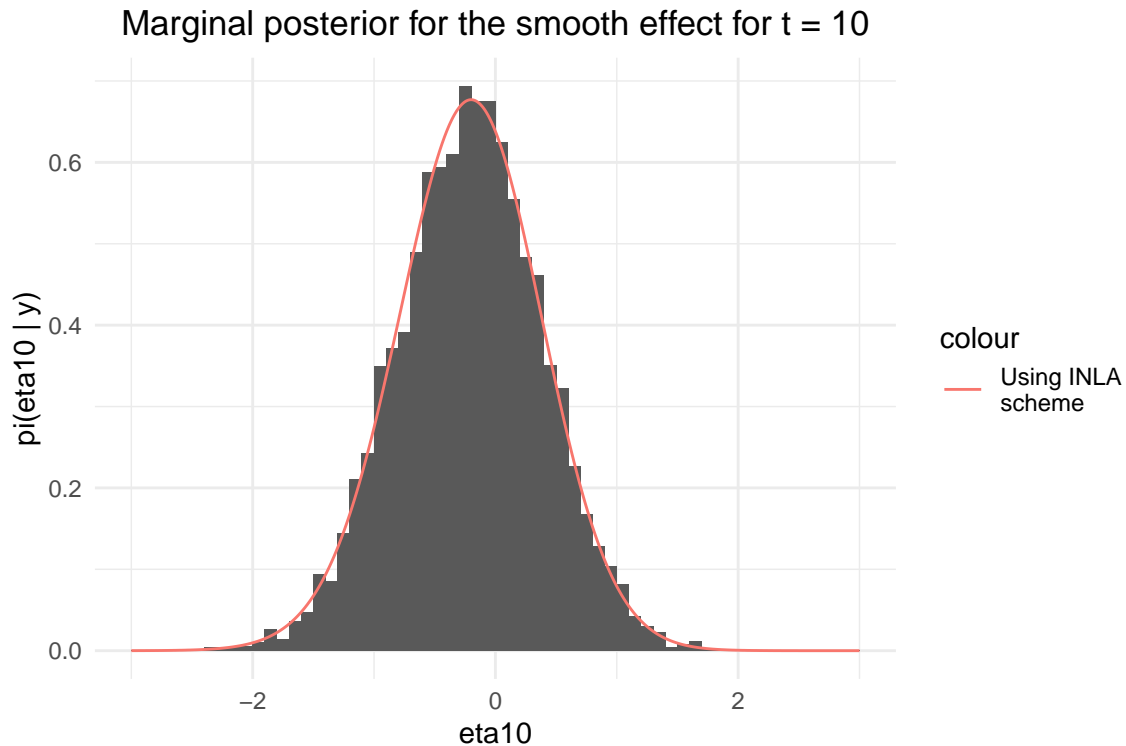
Figure 14: The posterior marginal for the smooth effect using the INLA scheme and using block Gibbs sampling.

```
               col = "Using R-INLA")
           ) +
 geom_line(aes(x = thetas,
               y = post,
               col = "Using INLA\nscheme"),
           linetype = "dashed"
           ) +
 xlim(0, 7) +
 ggtitle("Posterior marginal for the hyperparameter") +
 xlab("theta") +
 ylab("pi(theta | y)") +
 theme_minimal() +
 theme(plot.title = element_text(hjust = 0.5))
```

Secondly, we plot the mean of the smooth effect, as shown in Figure 16. Also here we see a strong correspondence.

```
ggplot() +
 geom_line(aes(x = t,
               y = res$summary.random$t$mean,
               col = "Using R-INLA")
           ) +
 geom_line(aes(x = t,
               y = eta_mean,
               col = "Using block Gibbs"),
```
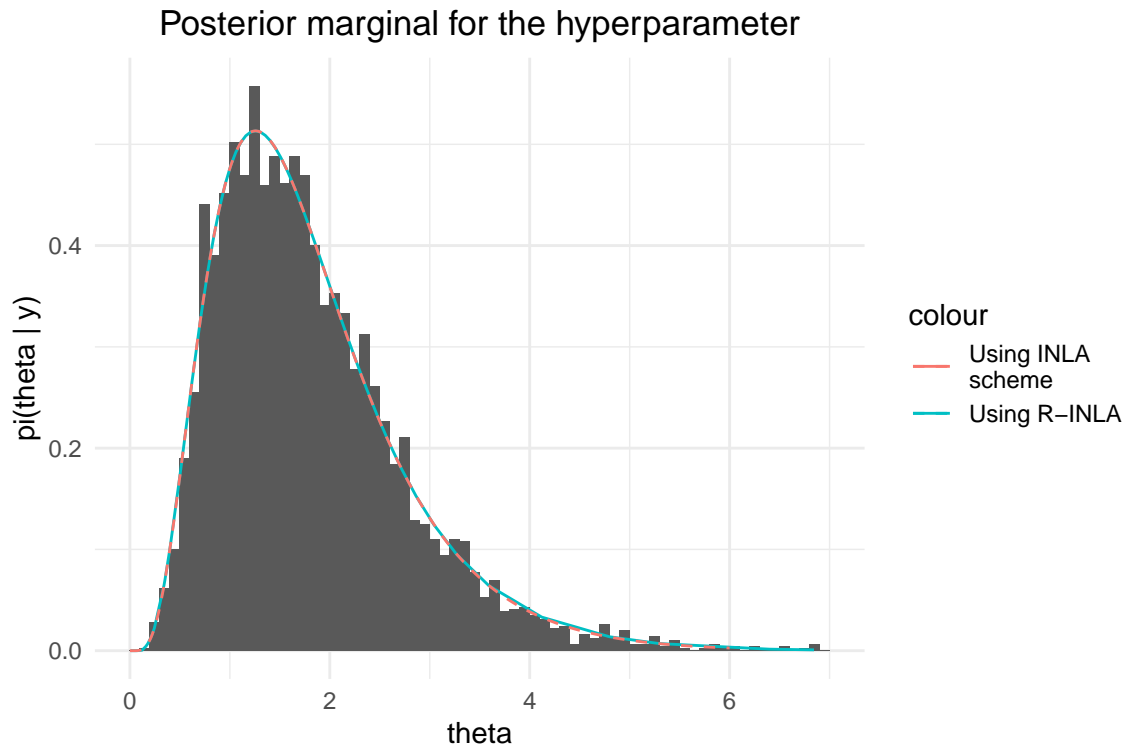
Figure 15: The posterior marginal for the hyperparameter. It shows the results of the block Gibbs sampling, the INLA scheme, and the 'R-INLA' package.

```
           linetype = "dashed"
         ) +
ylab("y") +
ggtitle("Mean for the smooth effect") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))
```

Lastly, we may use the result to plot the posterior marginal for the smooth effect, as shown in Figure 17. Finally we here see a good correspondence with the methods we used.

```
ggplot() +
  geom_histogram(mapping = aes(x = eta10Gibbs, y = ..density..),
                 binwidth = 0.1,
                 boundary = 0
                ) +
  geom_line(aes(x = res$marginals.random$t$index.10[, 1],
                y = res$marginals.random$t$index.10[, 2],
                col = "Using R-INLA")
           ) +
  geom_line(aes(x = etas,
                y = eta10,
                col = "Using INLA\nscheme"),
            linetype = "dashed"
           ) +
  xlim(-3, 3) +
```
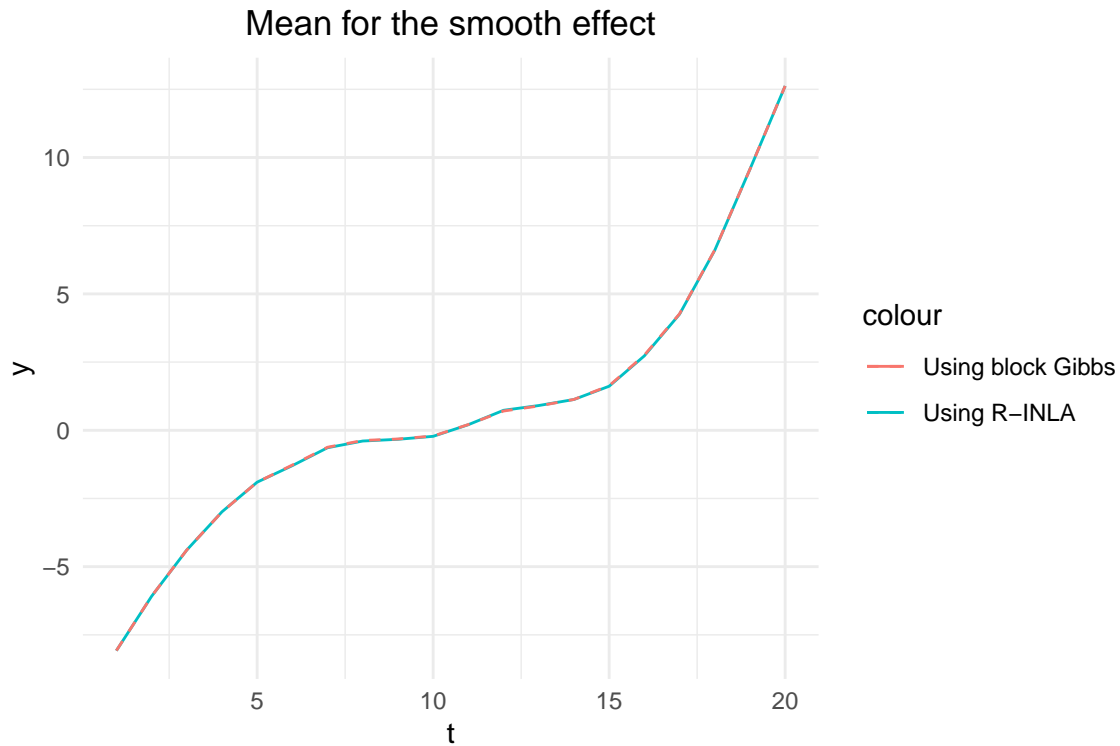
Figure 16: The mean of the smooth effect. It shows the result using block Gibbs sampling and the 'R-INLA' package.

```
xlab("eta10") +
ylab("pi(eta10 | y)") +
ggtitle("Marginal posterior for the smooth effect for t = 10") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))
```

Jarrett, RG. 1979. "A Note on the Intervals Between Coal-Mining Disasters." *Biometrika* 66 (1): 191–93.

Mills, Catherine. 2010. *Regulating Health and Safety in the British Mining Industries, 1800-1914*. Ashgate Publishing, Ltd.
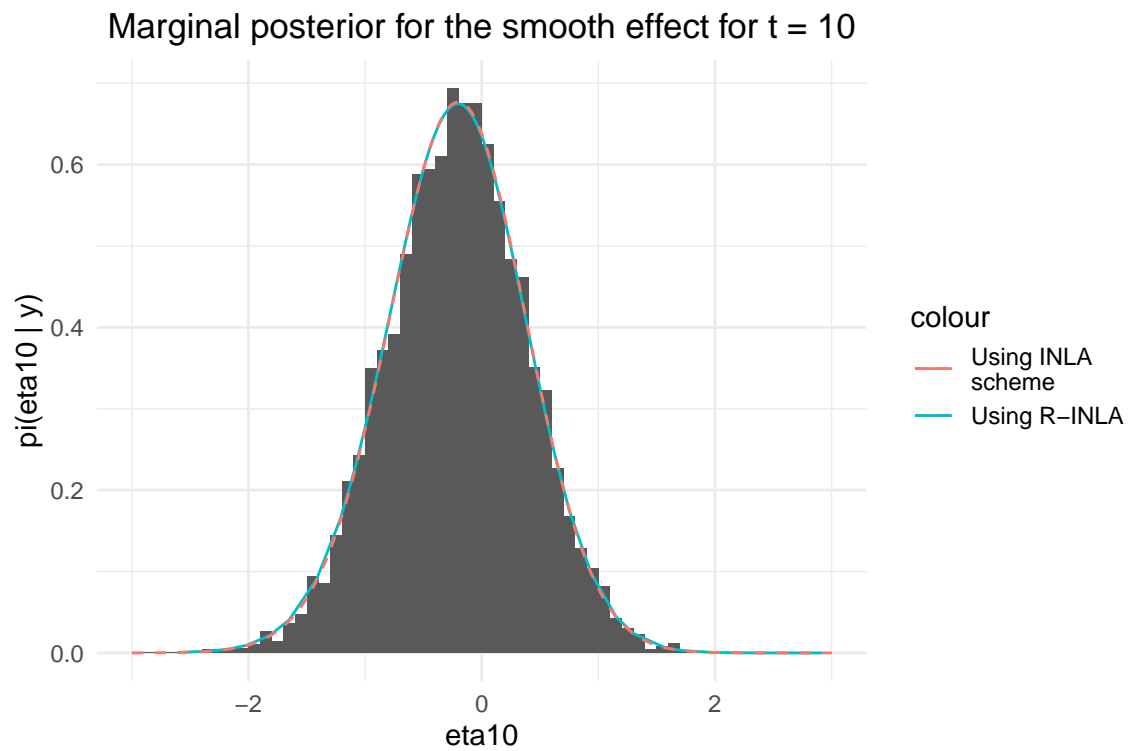
Figure 17: The posterior marginal for the smooth effect for $t = 10$. It shows the results of the block Gibbs sampling, the INLA scheme, and the 'R-INLA' package.