Francesco Caporali, Isabel Muzio
21 November 2022

Bayesian Statistics

Assignment 2

Question 1: Probit regression (Hoff 6.3)

A panel study followed $n = 25$ married couples over a period of five years. One item of interest is the relationship between divorce rates and the various characteristics of the couples. For example, the researchers would like to model the probability of divorce as a function of age differential, recorded as the man's age minus the woman's age. The data can be found in the file `divorce.RData`. We will model these data with probit regression, in which a binary variable Y_i is described in terms of an explanatory variable x_i via the following latent variable model:

$$\begin{aligned} Z_i &= \beta x_i + \varepsilon_i \\ Y_i &= \mathbb{1}_{(c,+\infty)}(Z_i), \end{aligned}$$

where β and c are unknown coefficients, $\varepsilon_1, \dots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and $\mathbb{1}_{(c,+\infty)}(z) = 1$ if $z > c$ and equals zero otherwise. In the following, since the covariates x_i are known, they will be treated as constants and so not explicitly written in the conditioning part.

Point a.

Assuming $\beta \sim \mathcal{N}\left(0, \sigma_\beta^2\right)$, obtain the full conditional distribution $p(\beta \,|\, y_{1:n}, z_{1:n}, c)$.

First of all let us write explicitly the conditional distributions which we can deduce from the text:

- $\forall i = 1, \dots, n$ we know $p(z_i \,|\, \beta)$:

$$Z_i(\omega) \,|\, \beta = \beta x_i + \varepsilon_i(\omega) \sim \beta x_i + \mathcal{N}(0,1) \sim \mathcal{N}(\beta x_i, 1) \implies Z_i \,|\, \beta \sim \mathcal{N}(\beta x_i, 1)$$

$$\Downarrow$$

$$p(z_i \,|\, \beta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z_i - \beta x_i)^2};$$

- $\forall i = 1, \dots, n$ we know $p(y_i \,|\, c, z_i)$:

$$Y_i(\omega) = \mathbb{1}_{(c,+\infty)}(Z_i(\omega)) = \begin{cases} 1 & \text{if } Z_i(\omega) > c \\ 0 & \text{otherwise} \end{cases}$$

$$\Downarrow$$

$$\begin{aligned} p(y_i) = \mathbb{P}(Y_i = y_i) = \mathbb{P}\left(\mathbb{1}_{(c,+\infty)}(Z_i) = y_i\right) = \\ = \begin{cases} \mathbb{P}\left(\mathbb{1}_{(c,+\infty)}(Z_i) = 1\right) & \text{if } y_i = 1 \\ \mathbb{P}\left(\mathbb{1}_{(c,+\infty)}(Z_i) = 0\right) & \text{if } y_i = 0 \\ 0 & \text{otherwise} \end{cases} = \\ = \begin{cases} \mathbb{P}(\{Z_i > c\}) & \text{if } y_i = 1 \\ \mathbb{P}\left(\{Z_i > c\}^C\right) & \text{if } y_i = 0 \\ 0 & \text{otherwise} \end{cases} = \\ = \left(y_i \mathbb{P}(\{Z_i > c\}) + (1 - y_i)\mathbb{P}\left(\{Z_i > c\}^C\right)\right) \mathbb{1}_{\{0,1\}}(y_i), \end{aligned}$$

hence $Y_i \sim \text{Bernoulli}\left(\mathbb{P}\left(Z_i > c\right)\right)$.
It follows that, conditionally on Z_i, c, the r.v. Y_i is no more *random* and it holds[1]

$$p(y_i \,|\, c, z_i) = \left(y_i \mathbb{1}_{(-\infty, z_i)}(c) + (1 - y_i)\mathbb{1}_{(-\infty, z_i)^C}(c)\right) \mathbb{1}_{\{0,1\}}(y_i).$$

In order to obtain (and sample) from the full conditionals we assume β and c a priori independent. The full conditional distribution $p(\beta \,|\, y_{1:n}, z_{1:n}, c)$ can be obtained just from $p(z_i \,|\, \beta)$, indeed

$$\begin{aligned}
p(\beta \,|\, y_{1:n}, z_{1:n}, c) &= \frac{p(\beta, y_{1:n}, z_{1:n}, c)}{p(y_{1:n}, z_{1:n}, c)} \frac{p(\beta, z_{1:n}, c)}{p(\beta, z_{1:n}, c)} \frac{p(\beta, c)}{p(\beta, c)} \frac{p(c)}{p(c)} \propto \\
&\propto \frac{p(\beta, y_{1:n}, z_{1:n}, c)}{p(\beta, z_{1:n}, c)} \frac{p(\beta, z_{1:n}, c)}{p(\beta, c)} \frac{p(\beta, c)}{p(c)} = \\
&= p(y_{1:n} \,|\, \cancel{\beta}, c, z_{1:n}) p(z_{1:n} \,|\, \beta, \cancel{c}) p(\beta \,|\, \cancel{c}) \propto \\
&\propto p(z_{1:n} \,|\, \beta) p(\beta).
\end{aligned}$$

So we can write explicitly

$$\begin{aligned}
p(\beta \,|\, y_{1:n}, z_{1:n}, c) &\propto p(z_{1:n} \,|\, \beta) p(\beta) = \\
&= \prod_{i=1}^{n} p(z_i \,|\, \beta) p(\beta) \propto \\
&\propto \exp\left(-\frac{1}{2} \sum_{i=1}^{n} (z_i - \beta x_i)^2\right) \exp\left(-\frac{1}{2} \frac{1}{\sigma_\beta^2} \beta^2\right) = \\
&= \exp\left(-\frac{1}{2}\left(\beta^2 \sum_{i=1}^{n} x_i^2 + \cancel{\sum_{i=1}^{n} z_i^2} - 2\beta \sum_{i=1}^{n} x_i z_i + \beta^2 \frac{1}{\sigma_\beta^2}\right)\right) = \\
&= \exp\left(-\underbrace{\left(\sum_{i=1}^{n} x_i^2 + \frac{1}{\sigma_\beta^2}\right)}_{\stackrel{\text{def}}{=} (\sigma_{\beta,n}^2)^{-1}} \frac{\beta^2}{2} + \underbrace{\left(\sum_{i=1}^{n} x_i z_i\right)}_{\stackrel{\text{def}}{=} \frac{\mu_{\beta,n}}{\sigma_{\beta,n}^2}} \beta\right),
\end{aligned}$$

where from the 1st to the 2nd line we used $(Z_i \,|\, \beta)_{i=1}^n$ independent, identically distributed r.v.'s.
So we can conclude that

$$\beta \,|\, y_{1:n}, z_{1:n}, c \sim \mathcal{N}\left(\mu_{\beta,n}, \sigma_{\beta,n}^2\right) \text{ with } \begin{cases} \sigma_{\beta,n}^2 = \left(\sum_{i=1}^n x_i^2 + \frac{1}{\sigma_\beta^2}\right)^{-1} \\ \mu_{\beta,n} = \sigma_{\beta,n}^2 \left(\sum_{i=1}^n x_i z_i\right) \end{cases}$$

$$\Downarrow$$

$$p(\beta \,|\, y_{1:n}, z_{1:n}, c) = \frac{1}{\sqrt{2\pi\sigma_{\beta,n}^2}} \exp\left(-\frac{1}{2\sigma_{\beta,n}^2}(\beta - \mu_{\beta,n})^2\right).$$

□

Point b.

Assuming $c \sim \mathcal{N}\left(0, \sigma_c^2\right)$, show that $p(c \,|\, y_{1:n}, z_{1:n}, \beta)$ is a constrained normal density, i.e. proportional to a normal density but constrained to lie in an interval. Similarly, show that $p(z_i \,|\, y_{1:n}, z_{-i}, \beta, c)$ is proportional to a normal density but constrained to be either above c or below c, depending on y_i.

[1] We replace $\mathbb{P}\left(\{z_i > c\}\right)$ with $\mathbb{1}_{(-\infty, z_i)}(c)$ because we will use this characterization afterwards.

Hint: A constrained, or truncated, normal random variable V is obtained by restricting a normally distributed random variable $\mathcal{N}(\mu, \tau^2)$ to lie in an interval (a, b), with possibly $a = -\infty$ or $b = +\infty$. We use the notation $V \sim \mathcal{TN}_{(a,b)}(\mu, \tau^2)$. It holds:

- $p(v \,|\, \mu, \tau^2, a, b) = \frac{1}{C} \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{1}{2\tau^2}(v-\mu)^2\right) \mathbb{1}_{(a,b)}(v)$, where $C = \Phi\left(\frac{b-\mu}{\tau}\right) - \Phi\left(\frac{a-\mu}{\tau}\right)$ being $\Phi(\cdot)$ the cdf of the standard normal distribution. By definition, it holds $\Phi\left(\frac{b-\mu}{\tau}\right) = 1$ if $b = \infty$ and $\Phi\left(\frac{a-\mu}{\tau}\right) = 0$ if $a = -\infty$.
- Sampling can be performed thanks to the function `rtruncnorm(n, a, b, mean, sd)` from the package `truncnorm` [https://cran.r-project.org/web/packages/truncnorm/truncnorm.pdf]. This function receives in input the number of desired samples (n) and the four parameters specifying the distribution of V: a, b, μ, τ. Pay attention that it takes as last inputs the mean μ and the standard deviation τ (not the variance τ^2) of the un-truncated normal density.

As before, the full conditional distribution $p(c \,|\, y_{1:n}, z_{1:n}, \beta)$ can be obtained just from $p(y_i \,|\, c, z_i)$, indeed

$$\begin{aligned}
p(c \,|\, y_{1:n}, z_{1:n}, \beta) &= \frac{p(c, y_{1:n}, z_{1:n}, \beta)}{p(y_{1:n}, z_{1:n}, \beta)} \frac{p(\beta, c, z_{1:n})}{p(\beta, c, z_{1:n})} \frac{p(c, \beta)}{p(c, \beta)} \frac{p(\beta)}{p(\beta)} \propto \\
&\propto \frac{p(c, y_{1:n}, z_{1:n}, \beta)}{p(\beta, c, z_{1:n})} \frac{p(\beta, c, z_{1:n})}{p(c, \beta)} \frac{p(c, \beta)}{p(\beta)} = \\
&= p(y_{1:n} \,|\, \cancel{\beta}, c, z_{1:n}) p(z_{1:n} \,|\, \beta, \cancel{c}) p(c \,|\, \cancel{\beta}) \propto \\
&\propto p(y_{1:n} \,|\, c, z_{1:n}) p(c).
\end{aligned}$$

So we can write explicitly

$$\begin{aligned}
p(c \,|\, y_{1:n}, z_{1:n}, \beta) &\propto p(y_{1:n} \,|\, c, z_{1:n}) p(c) = \\
&= \prod_{i=1}^{n} p(y_i \,|\, c, z_i) p(c) \propto \\
&\propto \exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2} c^2\right) \prod_{i=1}^{n} \left(y_i \mathbb{1}_{(-\infty, z_i)}(c) + (1 - y_i) \mathbb{1}_{(-\infty, z_i)^C}(c)\right) \mathbb{1}_{\{0,1\}}(y_i) = \\
&= \exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2} c^2\right) \prod_{i=1,\ldots,n \,|\, y_i = 1} \mathbb{1}_{(-\infty, z_i)}(c) \cdot \prod_{i=1,\ldots,n \,|\, y_i = 0} \mathbb{1}_{[z_i, +\infty)}(c) = \\
&= \exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2} c^2\right) \mathbb{1}_{(-\infty,\, \min(z_i \,|\, i \in \{1,\ldots,n\}, y_i = 1))}(c) \mathbb{1}_{[\max(z_i \,|\, i \in \{1,\ldots,n\}, y_i = 0),\, +\infty)}(c),
\end{aligned}$$

where from the 1st to the 2nd line we used $(Y_i \,|\, c, z_i)_{i=1}^n$ independent, identically distributed r.v.'s.
More compactly, defining

$$\begin{aligned}
a_n &\stackrel{\text{def}}{=} \max\left(z_i \,|\, i \in \{1, \ldots, n\}, y_i = 0\right) \text{ and} \\
b_n &\stackrel{\text{def}}{=} \min\left(z_i \,|\, i \in \{1, \ldots, n\}, y_i = 1\right),
\end{aligned}$$

one has

$$p(c \,|\, y_{1:n}, z_{1:n}, \beta) \propto \exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2} c^2\right) \mathbb{1}_{[a_n, b_n)}(c),$$

where, for a good definition, we are using $a_n < b_n$ which is clearly true because, if a_n, b_n are finite, $\forall i, j \in \{1, \ldots, n\}$ such that $y_i = 0$, $y_j = 1$, $(-\infty, c] \ni z_i < z_j \in (c, +\infty)$.
First of all we have to observe that the indicator function constrains $c \in [a_n, b_n)$, but it is equivalent to $c \in (a_n, b_n)$ because our $p(c \,|\, y_{1:n}, z_{1:n}, \beta)$ is a density function with respect to the lebesgue measure on $\mathbb{R}$ so each point has measure 0 (so does $\{a_n\}$).

Then, let us observe that this conditional density is proportional to the kernel of a gaussian (evaluated in c) multiplied by an indicator function (also evaluated in c), which constrains the domain to an interval (not necessarily limited, possibly $a_n = -\infty$ or $b_n = +\infty$).
So completing the function $\exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2}c^2\right)\mathbb{1}_{(a_n,b_n)}(c)$ to a density one obtains

$$p(c \,|\, y_{1:n}, z_{1:n}, \beta) = \frac{1}{\Phi\left(\frac{b_n}{\sigma_c}\right) - \Phi\left(\frac{a_n}{\sigma_c}\right)} \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{1}{2}\frac{1}{\sigma_c^2}c^2\right)\mathbb{1}_{(a_n,b_n)}(c)$$
$$\Downarrow$$
$$c \,|\, y_{1:n}, z_{1:n}, \beta \sim \mathcal{TN}_{(a_n,b_n)}\left(0, \sigma_c^2\right).$$

Similarly

$$\begin{aligned}
p(z_i \,|\, y_{1:n}, z_{-i}, \beta, c) &= \frac{p(z_i, y_{1:n}, z_{-i}, \beta, c)}{p(y_{1:n}, z_{-i}, \beta, c)} \frac{p(z_i, z_{-i}, \beta, c)}{p(z_i, z_{-i}, \beta, c)} \frac{p(z_i, \beta, c)}{p(z_i, \beta, c)} \frac{p(\beta, c)}{p(\beta, c)} \propto \\
&\propto \frac{p(z_i, y_{1:n}, z_{-i}, \beta, c)}{p(z_i, z_{-i}, \beta, c)} \frac{p(z_i, z_{-i}, \beta, c)}{p(z_i, \beta, c)} \frac{p(z_i, \beta, c)}{p(\beta, c)} = \\
&= p(y_{1:n} \,|\, z_{1:n}, \cancel{\beta}, c) p(z_{-i} \,|\, \cancel{z_i}, \beta, \cancel{c}) p(z_i \,|\, \beta, \cancel{c}) \propto \\
&\propto p(y_{1:n} \,|\, z_{1:n}, c) p(z_i \,|\, \beta) \propto \\
&\propto \prod_{j=1}^{n} p(y_j \,|\, z_j, c) p(z_i \,|\, \beta) \propto \\
&\propto p(y_i \,|\, z_i, c) p(z_i \,|\, \beta) \propto \\
&\propto \Big(y_i \underbrace{\mathbb{1}_{(-\infty, z_i)}(c)}_{= \mathbb{1}_{(c,+\infty)}(z_i)} + (1 - y_i) \underbrace{\mathbb{1}_{(-\infty, z_i)^c}(c)}_{= \mathbb{1}_{(-\infty,c]}(z_i)}\Big) \mathbb{1}_{\{0,1\}}(y_i) \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) = \\
&= \begin{cases} \mathbb{1}_{(c,+\infty)}(z_i) \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) & \text{if } y_i = 1 \\ \mathbb{1}_{(-\infty,c]}(z_i) \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) & \text{if } y_i = 0 \end{cases}.
\end{aligned}$$

As before, this conditional density is proportional to the kernel of a gaussian (evaluated in z_i) multiplied by an indicator function (also evaluated in z_i) which constrains the domain to be $(c, +\infty)$ or $(-\infty, c]$ (equivalently $(-\infty, c)$, with the same motivation given above) depending on y_i.
In particular, completing to a density what we found

$$p(z_i \,|\, y_{1:n}, z_{-i}, \beta, c) = \begin{cases} \frac{1}{1 - \Phi(c - x_i\beta)} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) \mathbb{1}_{(c,+\infty)}(z_i) & \text{if } y_i = 1 \\ \frac{1}{\Phi(c - x_i\beta)} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(z_i - \beta x_i)^2\right) \mathbb{1}_{(-\infty,c)}(z_i) & \text{if } y_i = 0 \end{cases}$$
$$\Downarrow$$
$$Z_i \,|\, y_{1:n}, z_{-i}, \beta, c \sim \begin{cases} \mathcal{TN}_{(c,+\infty)}\left(\beta x_i, 1\right) & \text{if } y_i = 1 \\ \mathcal{TN}_{(-\infty,c)}\left(\beta x_i, 1\right) & \text{if } y_i = 0 \end{cases}.$$

□

Point c.

Letting $\sigma_\beta^2 = \sigma_c^2 = 16$, implement a Gibbs sampling scheme that approximates the joint posterior distribution of $Z_{1:n}, \beta$ and c. After a burnin of 1000, run the Gibbs sampler long enough so that the

effective sample sizes of all unknown parameters are greater than 1000 (including the Z_i's). Compute the autocorrelation function of the parameters and discuss the mixing of the Markov chain.

The prior distributions of β and c are

$$\beta \sim \mathcal{N}\left(0, \sigma_\beta^2\right),$$
$$c \sim \mathcal{N}\left(0, \sigma_c^2\right).$$

The full conditional distributions we found are the following

$$Z_i \,|\, y_i, \beta, c \sim \begin{cases} \mathcal{TN}_{(c,+\infty)}\left(\beta x_i, 1\right) & \text{if } y_i = 1 \\ \mathcal{TN}_{(-\infty,c)}\left(\beta x_i, 1\right) & \text{if } y_i = 0 \end{cases},$$

$$\beta \,|\, z_{1:n} \sim \mathcal{N}\left(\mu_{\beta,n}, \sigma_{\beta,n}^2\right) \text{ with } \begin{cases} \sigma_{\beta,n}^2 = \left(\sum_{i=1}^n x_i^2 + \dfrac{1}{\sigma_\beta^2}\right)^{-1} \\ \mu_{\beta,n} = \sigma_{\beta,n}^2 \left(\sum_{i=1}^n x_i z_i\right) \end{cases},$$

$$c \,|\, y_{1:n}, z_{1:n} \sim \mathcal{TN}_{(a_n,b_n)}\left(0, \sigma_c^2\right) \text{ with } \begin{cases} a_n \overset{\text{def}}{=} \max\left(z_i \,|\, i \in \{1,\dots,n\}, y_i = 0\right) \text{ and} \\ b_n \overset{\text{def}}{=} \min\left(z_i \,|\, i \in \{1,\dots,n\}, y_i = 1\right) \end{cases}.$$

```
library(coda)
library(truncnorm)
library(bayesplot)
library(dplyr)
library(ggplot2)
library(grid)
library(gridExtra)
library(lattice)

load(file = "divorce.RData")
set.seed(1)

# setting parameters
burnin = 1e3
tmax = burnin + 1e5
n = 25

# build and upload: beta, c, z_1:n, y_1:n
beta = c = matrix(0, tmax, 1)
z = y = matrix(0, tmax, n)
x = matrix(0, 1, n)
x[1, ] = divorce[, "X"]
y[1, ] = divorce[, "Y"]

# parameters for priors and full conditionals distributions
mu_beta = matrix(0, tmax, 1)
mu_beta[1] = 0
mu_c = 0
sigma_sq_beta = sigma_sq_c = 16
sigma_sq_beta_n = (sum(x^2) + (sigma_sq_beta)^(-1))^(-1)
a = matrix(-Inf, tmax, 1)
b = matrix(Inf, tmax, 1)

# prior samples
beta[1] = rnorm(1, mu_beta[1], sqrt(sigma_sq_beta))
c[1] = rnorm(1, mu_c, sqrt(sigma_sq_c))

# gibbs sampler
for (t in 2:(tmax)) {
    # z
    lower_bound = rep(c[t - 1], n)
```

```
    lower_bound[y[t - 1, ] == 0] = -Inf
    upper_bound = rep(c[t - 1], n)
    upper_bound[y[t - 1, ] == 1] = +Inf
    z[t, ] = rtruncnorm(n, lower_bound, upper_bound, beta[t - 1] * x, rep(1, n))

    # update y_1:n (redundant because they follow the behaviour of z_1:n which are sampled given y_1:n)
    y[t, ] = 1 * (z[t, ] > c[t - 1])

    # beta
    mu_beta[t] = sigma_sq_beta_n * sum(x * z[t, ])
    beta[t] = rnorm(1, mu_beta[t], sqrt(sigma_sq_beta_n))

    # c
    a[t] = max(z[t, ][y[t, ] == 0])
    b[t] = min(z[t, ][y[t, ] == 1])
    c[t] = rtruncnorm(1, a[t], b[t], mu_c, sqrt(sigma_sq_c))

    # re-update y_1:n (redundant because they follow the behaviour of z_1:n and c which are sampled given y_1:n)
    y[t, ] = 1 * (z[t, ] > c[t])

    # break if eff_size > 1000 (for all params)
    if (t > burnin + 1 & (t %% 1000 == 0)) {
        if (effectiveSize(c[c((burnin + 1):t)]) > 1000 &
            effectiveSize(beta[c((burnin + 1):t)]) > 1000 &
            prod(effectiveSize(z[c((burnin + 1):t), ]) > 1000) == 1)
        {
            c = c[c((burnin + 1):t)]
            beta = beta[c((burnin + 1):t)]
            z = z[c((burnin + 1):t), ]
            break
        }
    }
}

# mcmc conversion
beta_mcmc = mcmc(beta)
c_mcmc = mcmc(c)
z_mcmc = mcmc(z)
beta_c_mcmc = mcmc(cbind(beta, c))
```

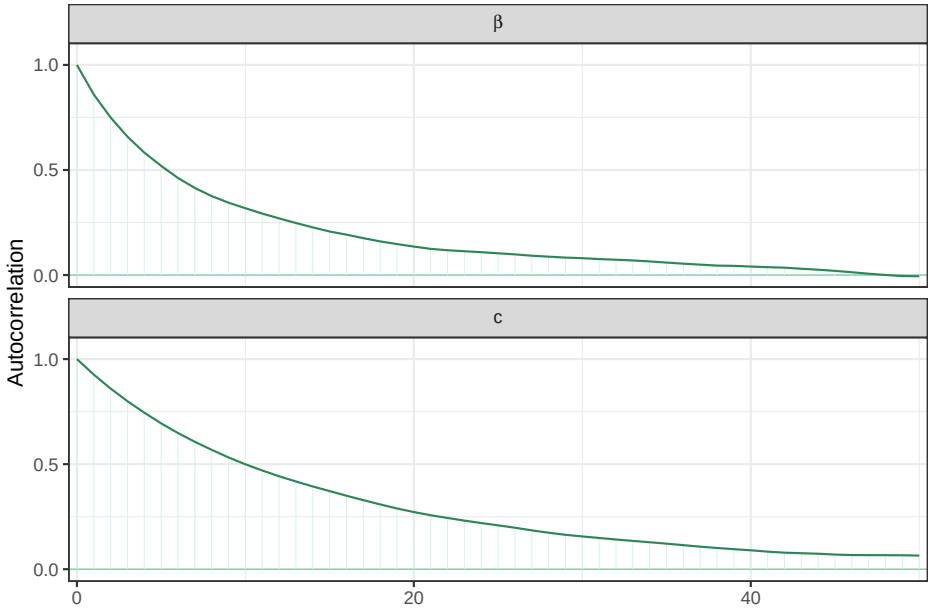
The effective sample size of the parameters are

- $S_{eff}(c) = 1028.33$;
- $S_{eff}(\beta) = 1643.82$;
- $S_{eff}(Z_{1:25}) = [4574, 16298, 14640, 7504, 16441, 25623, 3209, 2274, 13568$
  $4345, 26098, 4156, 2917, 3957, 3228, 28364, 3818, 4531$
  $4574, 16298, 14640, 7504, 16441, 25623, 3209, 2274, 13568]$.

Below are the autocorrelation functions of the parameters:

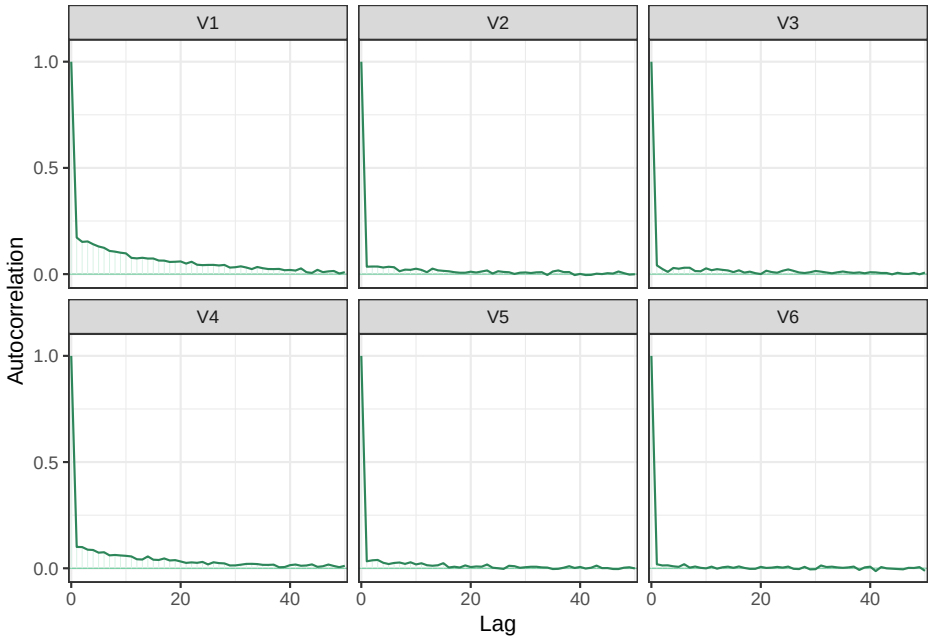
- β and c :

```
color_scheme_set("green")
# autocorrelation function of beta and c
mcmc_acf(as.data.frame(beta_c_mcmc), pars = c("beta", "c"),
         lags = 50, facet_args = list(nrow = 2, labeller = label_parsed)) +
    theme_bw() + xlab(" ")
```

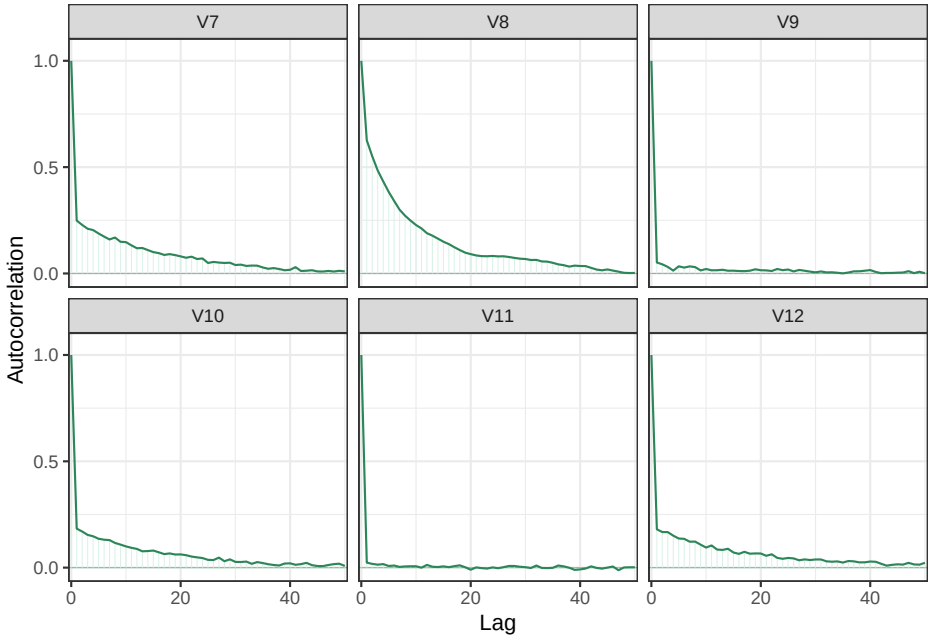


– $Z_{1:25}$ (here denoted as $(V1, \ldots V25)$):

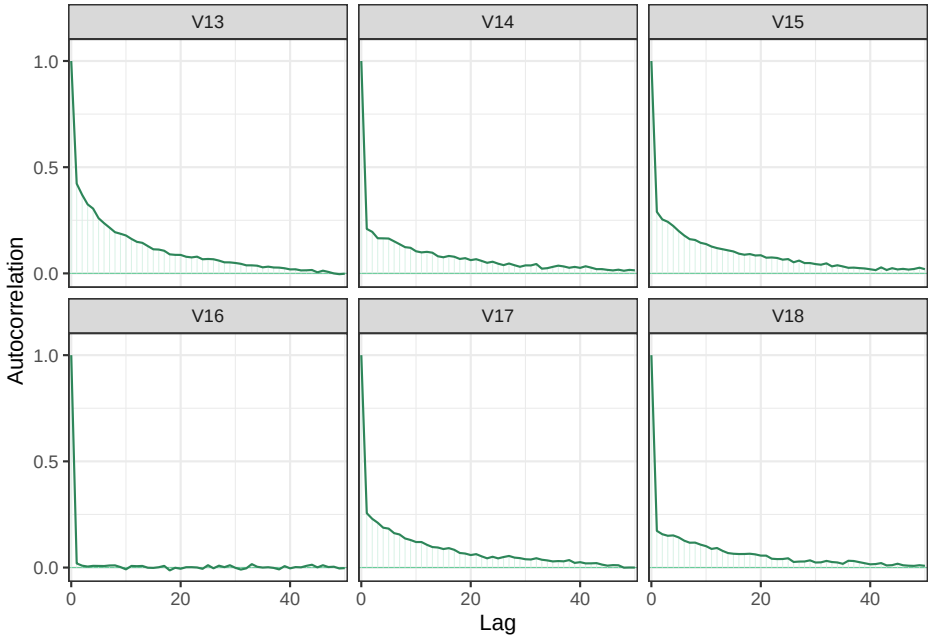
```
# autocorrelation function of z_1:25
mcmc_acf(as.data.frame(z_mcmc), pars = c(paste("V", 1:6, sep = "")), lags = 50) +
    theme_bw()
```



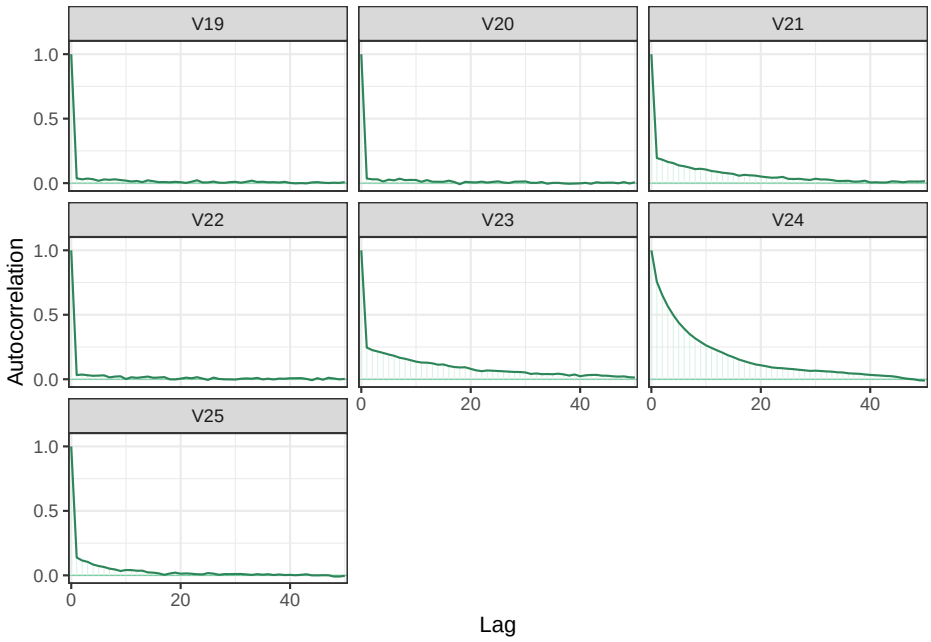

```
mcmc_acf(as.data.frame(z_mcmc), pars = c(paste("V", 7:12, sep = "")), lags = 50) +
    theme_bw()
```




```
mcmc_acf(as.data.frame(z_mcmc), pars = c(paste("V", 13:18, sep = "")), lags = 50) +
    theme_bw()
```




```
mcmc_acf(as.data.frame(z_mcmc), pars = c(paste("V", 19:25, sep = "")), lags = 50) +
    theme_bw()
```



Discuss the mixing of the markov chain. . . .

Point d.

Obtain a 95% posterior credible interval for β, as well as $\mathbb{P}(\beta > 0 \,|\, y_{1:n})$.

```
alpha = 0.05

# quantile-based 95% posterior credible interval
beta_95_quantiles = quantile(beta_mcmc, c(alpha / 2, 1 - (alpha / 2)))

# 95% hpd interval
beta_95_hpd = HPDinterval(beta_mcmc, prob = 1 - alpha)

# \prob(beta > 0 | y_1:n)
prob_beta_0 = length(beta[beta > 0]) / length(beta)
```

So the results of the two requests are the following:

- quantile-based 95% posterior credible interval for β

$$I_{\beta,\text{quantile}} = [0.1022937,\, 0.6748897]\,;$$

- highest posterior density 95% interval for β

$$I_{\beta,\text{hpd}} = [0.0785626,\, 0.6432298]\,;$$

- probability of the event $\beta > 0 \,|\, y_{1:n}$

$$\mathbb{P}(\beta > 0 \,|\, y_{1:n}) = 0.9989\,.$$

Question 2: Hierarchical Modeling

The file `schools.RData` gives weekly hours spent on homework for students sampled from eight different schools. Obtain posterior distributions for the true means for the eight different schools using a hierarchical normal model with the following prior parameters:

$$\mu_0 = 7, \gamma_0^2 = 5, \eta_0 = 2, \tau_0^2 = 10, \nu_0 = 2, \sigma_0^2 = 15.$$

That is,

$$
\begin{aligned}
y_{1,j}, \ldots, y_{n_j,j} \,|\, \theta_j, \sigma^2 &\overset{\text{iid}}{\sim} \mathcal{N}\left(\theta_j, \sigma^2\right), j = 1, \ldots, 8, \\
\theta_1, \ldots, \theta_8 \,|\, \mu, \tau^2 &\overset{\text{iid}}{\sim} \mathcal{N}\left(\mu, \tau^2\right), \\
\mu \sim \mathcal{N}\left(\mu_0, \gamma_0^2\right), &\quad 1/\tau^2 \sim \text{Gamma}(\eta_0/2, \eta_0\tau_0^2/2), \quad 1/\sigma^2 \sim \text{Gamma}(\nu_0/2, \nu_0\sigma_0^2/2)
\end{aligned}
$$

Point a.

Run a Gibbs sampling algorithm to approximate the posterior distribution of $\{\theta_1, \ldots, \theta_8, \mu, \sigma^2, \tau^2\}$. Assess the convergence of the Markov chain, and find the effective sample size for $\{\theta_1, \ldots, \theta_8, \mu, \sigma^2, \tau^2\}$. Run the chain long enough so that the effective sample sizes are all above 1000, after a burnin of 1000.

Recalling from the the theory the full conditional distributions of $\theta_{1:8}, \mu, \sigma^2$ and τ^2 (defining $m \overset{\text{def}}{=} 8$)

$$
\begin{aligned}
\theta_j \,|\, \{y_{i,n_j}\}_{i=1,\ldots,n_j,\, j=1,\ldots,m}, \mu, \tau^2, \sigma^2 &\sim \mathcal{N}\left(\frac{n_j\bar{y}_j/\sigma^2 + \mu/\tau^2}{n_j/\sigma^2 + 1/\tau^2}, \left(n_j/\sigma^2 + 1/\tau^2\right)^{-1}\right), j = 1, \ldots, 8, \\
1/\sigma^2 \,|\, \{y_{i,n_j}\}_{i=1,\ldots,n_j,\, j=1,\ldots,m}, \mu, \tau^2, \sigma^2 &\sim \text{Gamma}\left(\frac{\nu_0 + \sum_{j=1}^m n_j}{2}, \frac{\nu_0\sigma_0^2 + \sum_{j=1}^m \sum_{i=1}^{n_j}(y_{ij} - \theta_j)^2}{2}\right), \\
\mu \,|\, \theta_{1:m}, \tau^2 &\sim \mathcal{N}\left(\frac{m\bar{\theta}/\tau^2 + \mu_0/\gamma_0^2}{m/\tau^2 + 1/\gamma_0^2}, \left(m/\tau^2 + 1/\gamma_0^2\right)^{-1}\right), \\
1/\tau^2 \,|\, \theta_{1:m}, \mu &\sim \text{Gamma}\left(\frac{\eta_0 + m}{2}, \frac{\eta_0\tau_0^2 + \sum_{j=1}^m (\theta_j - \mu)^2}{2}\right),
\end{aligned}
$$

we can implement a Gibbs sampling scheme that approximates the posterior distribution of $\{\theta_{1:m}, \mu, \sigma^2, \tau^2\}$. After a burnin of 1000, we run the Gibbs sampler long enough so that the effective sample sizes of all unknown parameters are greater than 1000 (including the θ_i's).

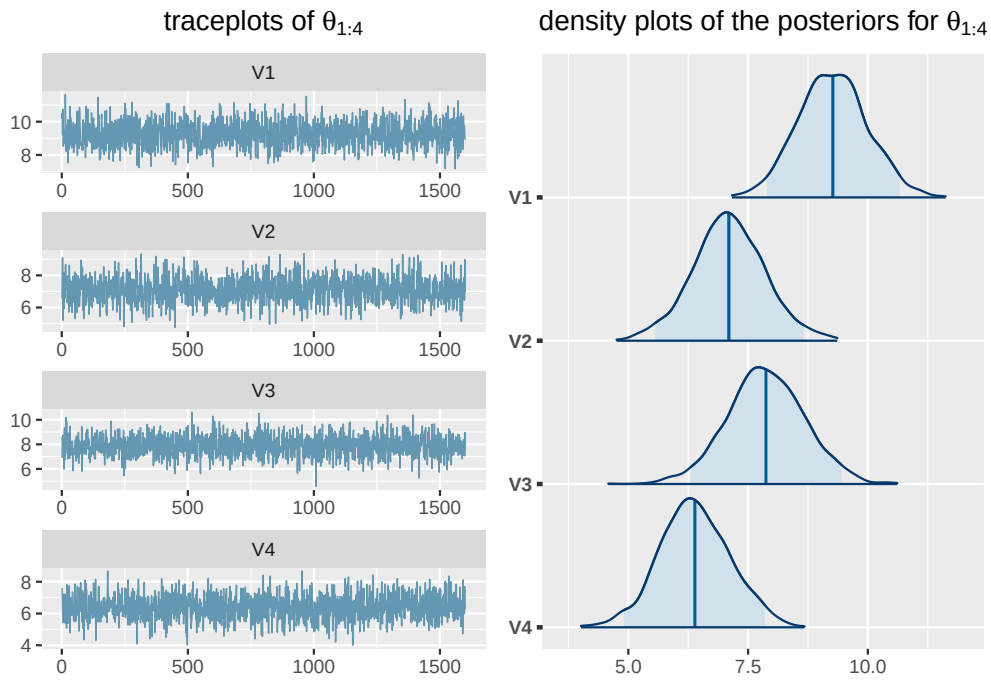
```
load(file = "schools.RData")
set.seed(1)

# setting parameters
burnin = 1e3
tmax = burnin + 6e3
m = length(unique(Y[, 1]))

# build: theta, sigma^2, mu, tau^2
theta = matrix(0, tmax, m)
sigma2 = matrix(0, tmax, 1)
mu = matrix(0, tmax, 1)
tau2 = matrix(0, tmax, 1)

# parameters for priors and full conditionals distributions
# (mu_0, gamma_0^2, eta_0, tau_0^2, nu_0, sigma_0^2)
mu0 = 7
g20 = 5
eta0 = 2
t20 = 10
nu0 = 2
s20 = 15

# compute prior values: theta, sigma^2, mu, tau^2
#   - n_j, bar_y_j, sample_var_y_j
n = matrix(0, 1, m)
ybar = matrix(0, 1, m)
sample_var_y = matrix(0, 1, m)
for (j in 1:m) {
    y_j = Y[Y[, 1] == j, 2]
```

```
    n[j] = length(y_j)
    ybar[j] = mean(y_j)
    sample_var_y[j] = var(y_j)
}
#   - theta[1, 1:8], sigma^2[1], mu[1], tau^2[1]
theta[1, ] = ybar
sigma2[1] = mean(sample_var_y)
mu[1] = mean(theta[1, ])
tau2[1] = var(theta[1, ])

# gibbs sampler
iterations = 0
for (t in 2:tmax) {

    # theta_1:m
    for (j in 1:m) {
        var_theta = 1 / (n[j] / sigma2[t - 1] + 1 / tau2[t - 1])
        mean_theta = var_theta * (n[j] * ybar[j] / sigma2[t - 1] + mu[t - 1] / tau2[t - 1])
        theta[t, j] = rnorm(1, mean_theta, sqrt(var_theta))
    }

    # sigma^2
    sigma2_shape = nu0 + sum(n) / 2
    sigma2_scale = nu0 * s20
    for (j in 1:m) {
        y_j = Y[Y[, 1] == j, 2]
        sigma2_scale = sigma2_scale + sum((y_j - theta[t, j])^2)
    }
    sigma2_scale = sigma2_scale / 2
    sigma2[t] = 1 / rgamma(1, sigma2_shape, sigma2_scale)

    # mu
    var_mu = 1 / (m / tau2[t - 1] + 1 / g20)
    mean_mu = var_mu * (m * mean(theta[t, ]) / tau2[t - 1] + mu0 / g20)
    mu[t] = rnorm(1, mean_mu, sqrt(var_mu))

    # tau^2
    tau2_shape  = (eta0 + m) / 2
    tau2_scale = (eta0 * t20 + sum((theta[t, ] - mu[t])^2)) / 2
    tau2[t] = 1 / rgamma(1, tau2_shape, tau2_scale)

    # break if eff_size > 1000 (for all params)
    if (t > burnin + 1 & (t %% 200 == 0)) {
        if (effectiveSize(mu[c((burnin + 1):t)]) > 1000 &
            effectiveSize(tau2[c((burnin + 1):t)]) > 1000 &
            effectiveSize(sigma2[c((burnin + 1):t)]) > 1000 &
            prod(effectiveSize(theta[c((burnin + 1):t), ]) > 1000) == 1)
        {
            theta = theta[c((burnin + 1):t), ]
            sigma2 = sigma2[c((burnin + 1):t)]
            mu = mu[c((burnin + 1):t)]
            tau2 = tau2[c((burnin + 1):t)]
            iterations = t - burnin
            break
        }
    }
}

# mcmc conversion
theta_mcmc = mcmc(theta)
sigma2_mcmc = mcmc(sigma2)
mu_mcmc = mcmc(mu)
tau2_mcmc = mcmc(tau2)
```

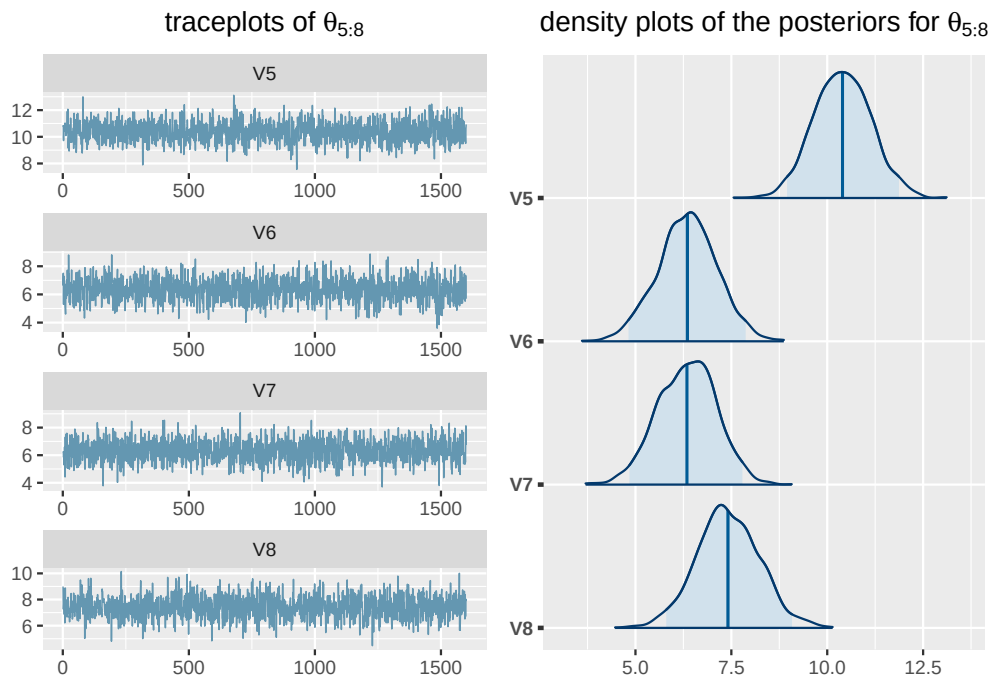
The results are the following:

```
# theta
color_scheme_set("blue")
p1 <- mcmc_trace(as.data.frame(theta_mcmc), pars = c(paste("V", 1:4, sep = "")),
            facet_args = list(nrow = 4, labeller = label_parsed)) +
```

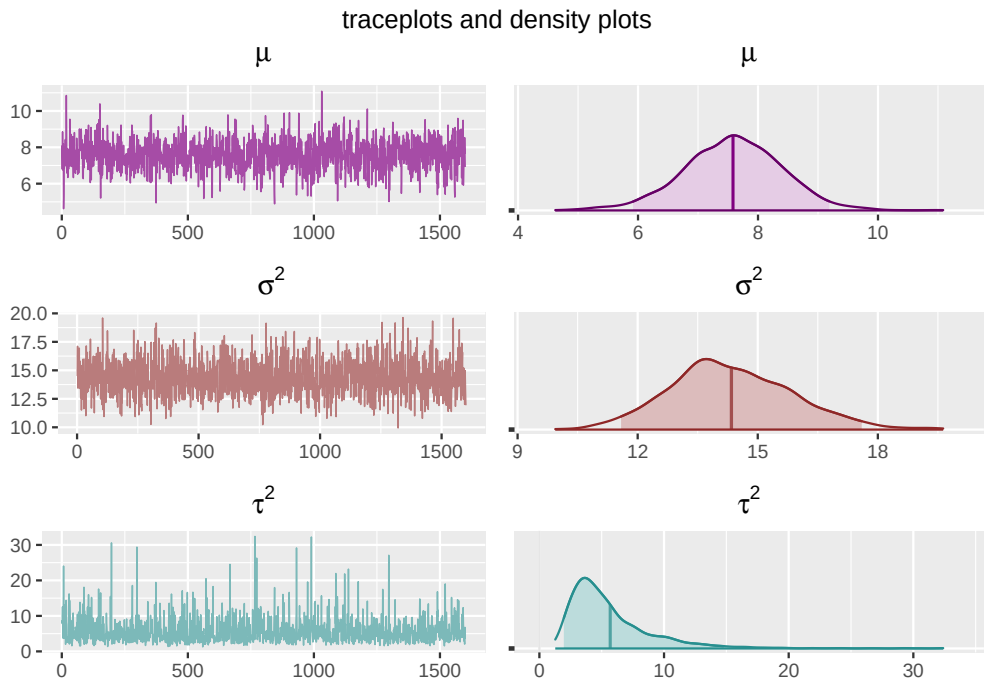
```
        ggtitle(expression(paste("traceplots of ", theta[1:4]))) +
        scale_y_continuous(breaks = c(4, 6, 8 ,10, 12, 14))
p2 <- mcmc_areas(as.data.frame(theta_mcmc), pars = c(paste("V", 1:4, sep = "")),
           prob = 0.95, prob_outer = 1, point_est = "mean") +
        ggtitle(expression(paste("density plots of the posteriors for ", theta[1:4])))
grid.arrange(p1, p2, nrow = 1)
```




```
p1 <- mcmc_trace(as.data.frame(theta_mcmc), pars = c(paste("V", 5:8, sep = "")),
           facet_args = list(nrow = 4, labeller = label_parsed)) +
        ggtitle(expression(paste("traceplots of ", theta[5:8]))) +
        scale_y_continuous(breaks = c(4, 6, 8 ,10, 12, 14))
p2 <- mcmc_areas(as.data.frame(theta_mcmc), pars = c(paste("V", 5:8, sep = "")),
           prob = 0.95, prob_outer = 1,    point_est = "mean") +
        ggtitle(expression(paste("density plots of the posteriors for ", theta[5:8])))
grid.arrange(p1, p2, nrow = 1)
```




```
# mu, sigma^2, tau^2
theme_update(plot.title = element_text(hjust = 0.5))
color_scheme_set("purple")
q1 <- mcmc_trace(as.data.frame(mu_mcmc)) +
        ggtitle(expression(mu)) +
        yaxis_title(on = FALSE)
q2 <- mcmc_areas(as.data.frame(mu_mcmc),
            prob = 0.95, prob_outer = 1, point_est = "mean") +
        ggtitle(expression(mu)) + yaxis_text(on = FALSE)
color_scheme_set("red")
q3 <- mcmc_trace(as.data.frame(sigma2_mcmc)) +
        ggtitle(expression(sigma^2)) +
        yaxis_title(on = FALSE)
q4 <- mcmc_areas(
        as.data.frame(sigma2_mcmc), prob = 0.95, prob_outer = 1, point_est = "mean") +
        ggtitle(expression(sigma^2)) +
        yaxis_text(on = FALSE)
color_scheme_set("teal")
q5 <- mcmc_trace(as.data.frame(tau2_mcmc)) +
        ggtitle(expression(tau^2)) +
        yaxis_title(on = FALSE)
q6 <- mcmc_areas(as.data.frame(tau2_mcmc),
            prob = 0.95, prob_outer = 1, point_est = "mean") +
        ggtitle(expression(tau^2)) +
        yaxis_text(on = FALSE)

grid.arrange(q1, q2, q3, q4, q5, q6, nrow = 3, top = textGrob("traceplots and density plots"))
```



As you can see, the chains span quite well and converge nicely and quite rapidly.
In fact, we can observe that the effective sample size of the parameters are

- $S_{eff}(\mu) = 1067$;
- $S_{eff}(\sigma^2) = 1600$;
- $S_{eff}(\tau^2) = 1196$;
- $S_{eff}(\theta_{1:8}) = [1600, 1659, 1600, 1600, 1306, 1386, 1600, 1600]$.

Which, compared to their length = 1600, signifies a quite fast convergence.

Point b.

Compute posterior means and 95% confidence regions for $\{\mu, \sigma^2, \tau^2\}$. Also, compare the posterior densities to the prior densities, and discuss what was learned from the data. (For the density of the inverse-Gamma distribution you can use the function `dinvgamma(x, shape, rate)` from the library `invgamma`).

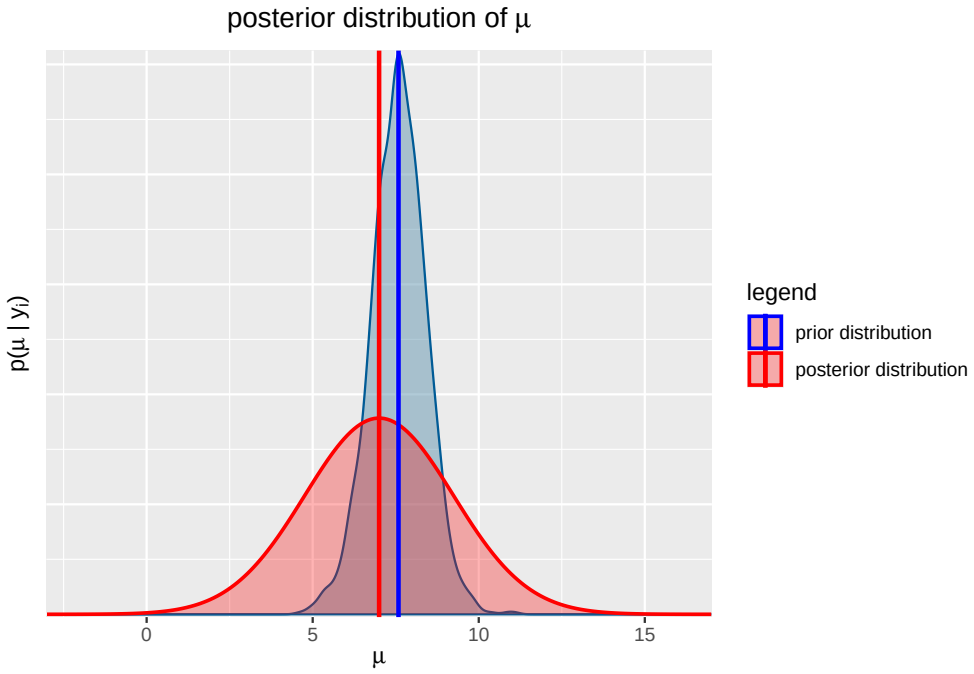
Let us compute posterior means and 95% confidence regions for $\{\mu, \sigma^2, \tau^2\}$.

```
alpha = 0.05

# posterior means, quantile-based 95% posterior credible interval
summary_mu = summary(mu_mcmc, quantiles = c(alpha / 2, 1 - (alpha / 2)))
summary_sigma2 = summary(sigma2_mcmc, quantiles = c(alpha / 2, 1 - (alpha / 2)))
summary_tau2 = summary(tau2_mcmc, quantiles = c(alpha / 2, 1 - (alpha / 2)))

# 95% hpd interval
mu_95_hpd = HPDinterval(mu_mcmc, prob = 1 - alpha)
sigma2_95_hpd = HPDinterval(sigma2_mcmc, prob = 1 - alpha)
tau2_95_hpd = HPDinterval(tau2_mcmc, prob = 1 - alpha)
```

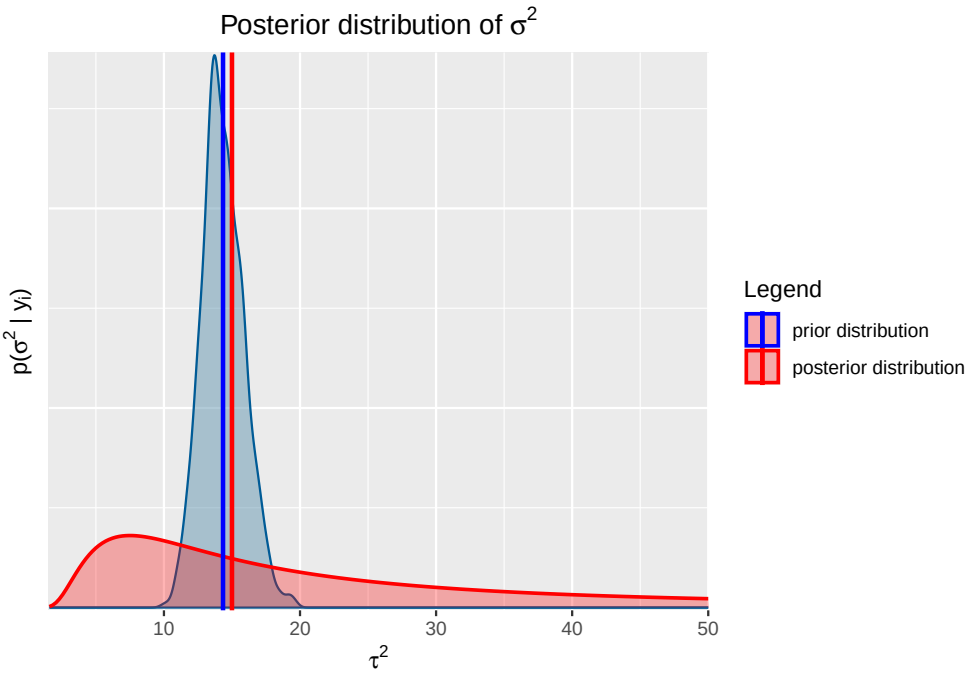
In the code below we compare the posterior densities of our parameters μ, τ^2 and σ^2 to their prior ones.

```
library(invgamma)

# density plots comparison

# data
```

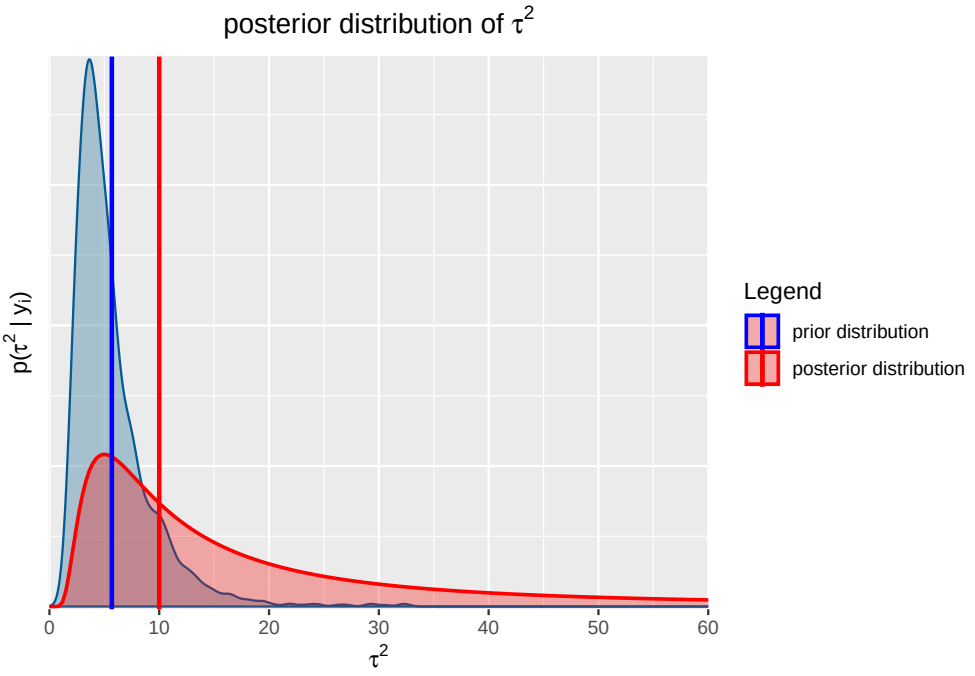
```
#   - mu
x1 = seq(mu0 - 2 * g20,  mu0 + 2 * g20, length = 200)
norm_vals = dnorm(x1, mu0, sqrt(g20))
#   - sigma^2
x2 = seq(summary_sigma2$quantiles[1] - 10, 50, length = 200)
invg_sigma2 = dinvgamma(x2, nu0 / 2, nu0 * s20 / 2)
#   - tau^2
x3 = seq(0.00001, 60, length = 200)
invg_tau2 = dinvgamma(x3, eta0 / 2, eta0 * t20 / 2)

priors = data.frame(x1, x2, x3, norm_vals, invg_sigma2, invg_tau2)
color_scheme_set("blue")

# mu
mcmc_dens(as.data.frame(mu_mcmc), alpha = 0.5) +
ggtitle(expression(paste("posterior distribution of ", mu))) +
geom_area(data = priors, aes(x = x1, y = norm_vals, col = "prior"),  fill = "red", alpha = 0.3, size = 0.8) +
geom_vline(aes(xintercept = summary_mu$statistics[1], col = "posterior"), size = 1) +
geom_vline(aes(xintercept = mu0, col = "prior"), size = 1) +
xlab(expression(mu)) +
ylab(expression(paste("p(", mu, " | ", y[i, j], ")"))) +
scale_color_manual(name = "legend", values = c("prior" = "red", "posterior" = "blue"),
    labels = c("prior distribution", "posterior distribution"))
```




```
# sigma^2
mcmc_dens(as.data.frame(sigma2_mcmc), alpha = 0.5)+
ggtitle(expression(paste("Posterior distribution of ", sigma^2))) +
geom_area(data = priors, aes(x = x2, y = invg_sigma2, col = "prior"), fill = "red", alpha = 0.3, size = 0.8) +
geom_vline(aes(xintercept = summary_sigma2$statistics[1], col = "posterior"), size = 1)+
geom_vline(aes(xintercept = s20, col = "prior"), size = 1)+
xlab(expression(tau^2)) +
ylab(expression(paste("p(", sigma^2, " | ", y[i, j], ")"))) +
scale_color_manual(name = "Legend", values = c("prior" = "red", "posterior" = "blue"),
    labels = c("prior distribution", "posterior distribution"))
```




```
# tau^2
mcmc_dens(as.data.frame(tau2_mcmc), alpha = 0.5)+
ggtitle(expression(paste("posterior distribution of ", tau^2))) +
geom_area(data = priors, aes(x = x3, y = invg_tau2, col = "prior"), fill = "red", alpha = 0.3, size = 0.8) +
geom_vline(aes(xintercept = summary_tau2$statistics[1], col = "posterior"), size = 1) +
geom_vline(aes(xintercept = t20, col = "prior"), size = 1) +
xlab(expression(tau^2)) +
ylab(expression(paste("p(", tau^2, " | ", y[i, j], ")"))) +
scale_color_manual(name = "Legend", values = c("prior" = "red", "posterior" = "blue"),
    labels = c("prior distribution", "posterior distribution"))
```



As we can observe, the posterior densities are peaked around their means, which differs from the prior ones, though not by much: the data mitigate the effect of the priors, given that the mean will converge to the sample one for each of the parameters, but the original offset was not too large.

What the data definitely affects is the variance, given that the posterior distributions are more pronouncedly peaked around their expectation value, so that we gain more certainty around each value.

This is furthermore reflected on their quantile-based 95% posterior credible intervals for μ, τ^2 and σ^2, which are respectively:

- $I_{\mu,\text{quantile}} = [5.9175769, 9.1881931]$ with posterior mean $\bar{\mu} = 7.58$;
- $I_{\sigma^2,\text{quantile}} = [11.5859236, 17.5970416]$ with posterior mean $\bar{\sigma}^2 = 14.34$;
- $I_{\tau^2,\text{quantile}} = [1.9612127, 14.7780462]$ with posterior mean $\bar{\tau}^2 = 5.68$.

The highest posterior density 95% interval for μ, τ^2 and σ^2 are rispectively:

- $I_{\mu,\text{hpd}} = [6.0386491, 9.2584451]$;
- $I_{\sigma^2,\text{hpd}} = [11.5030733, 17.3939734]$;
- $I_{\tau^2,\text{hpd}} = [1.4649314, 12.4616458]$.

Point c.

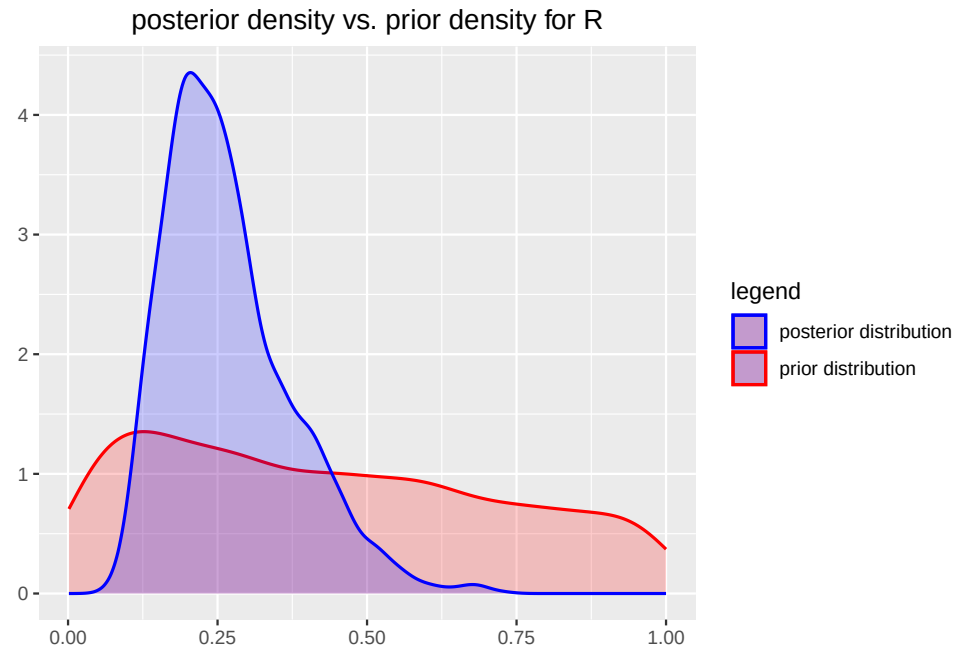
Plot the posterior density of $R = \frac{\tau^2}{\sigma^2+\tau^2}$ and compare it to a plot of the prior density of R (obtained via MC). Describe the evidence for between-school variation.

We want to study in particular the beetween-school variance, and we do so by observing the prior and posterior densities of the parameter:

$$R = \frac{\tau^2}{\tau^2 + \sigma^2}.$$

This parameter gives us an index of the relevance of the between-school variation (given that τ^2 corresponds to the variance of the θ_j's), in relation to the within-school one (as σ^2 is the variance of the $Y_{i,j}$'s).
We start by observing that this index can only assume values in $[0, 1]$, given that both parameters are positive, and that the τ^2 contribute is the more relevant, the closer R is to 1.

```
# build: prior density of R
sample_r_prior = rep(0, iterations)
sample_sigma2_prior = rinvgamma(iterations, nu0 / 2, nu0 * s20 / 2)
sample_tau2_prior = rinvgamma(iterations, eta0 / 2, eta0 * t20 / 2)

# build: posterior density of R
sample_r_post = rep(0, iterations)

# mc
for (i in 1:iterations){
    sample_r_prior[i] = sample_tau2_prior[i] / (sample_sigma2_prior[i] + sample_tau2_prior[i])
    sample_r_post[i] = tau2[i] / (sigma2[i] + tau2[i])
}

# plot both densities
sample_r = data.frame(sample_r_prior, sample_r_post)
sample_r %>% ggplot() +
    geom_density(aes(x = sample_r_prior, col = "prior"), fill = "red", alpha = 0.2, size = 0.7) +
    geom_density(aes(x = sample_r_post, col = "posterior"), fill = "blue", alpha = 0.2, size = 0.7) +
    scale_color_manual(name = "legend", values = c("prior" = "red", "posterior" = "blue"),
    labels = c("posterior distribution", "prior distribution")) +
    ylab(" ") +
    xlab(" ") +
    ggtitle("posterior density vs. prior density for R")
```



We observe that while in the prior the values are quite spread-out, which agrees with the independency of the two parameters, the posterior distribution is quite peaked and for a finite value, closer to 0 than to 1: this shows that while not being the dominant effect, we do not have enough evidence to consider the between-school variation negligible (as it is not close enough to zero).

Point d.

Compute the posterior probability that, if we were to observe a new school with school-specific parameter $\theta_9, \theta_9 > \theta_7$, as well as the posterior predictive probability that a new observation from this school would be greater than a new observation from school 7.

Suppose we now observe a new school with school-specific parameter θ_9, and suppose we make a new observation of this school $\tilde{Y}_9$. Let us compare them to school-specific parameter θ_7 and to a new observation of the same school: $\tilde{Y}_7$.
In order to see weather or not they are higher than those of the latter we will exploit Monte Carlo sampling.

```
# build samples
sample_theta_9 =  rep(0, iterations)
sample_y_9 =  rep(0, iterations)
sample_y_7 =  rep(0, iterations)

# mc
for (i in 1:iterations){
    sample_theta_9[i] = rnorm(1, mu[i], sqrt(tau2[i]))
    sample_y_9[i] = rnorm(1, sample_theta_9[i], sqrt(sigma2[i] + tau2[i]))
    sample_y_7[i] = rnorm(1, theta[i, 7], sqrt(sigma2[i]))
}

mc_theta = sum(sample_theta_9 > theta[1:iterations, 7]) / iterations
mc_y = sum(sample_y_9 > sample_y_7) / iterations
```

The Monte Carlo estimates of the posterior probabilities of the two events are, respectively:

- $\mathbb{P}(\theta_9 > \theta_7) = 0.69375$;
- $\mathbb{P}\left(\tilde{Y}_9 > \tilde{Y}_7\right) = 0.560625$.

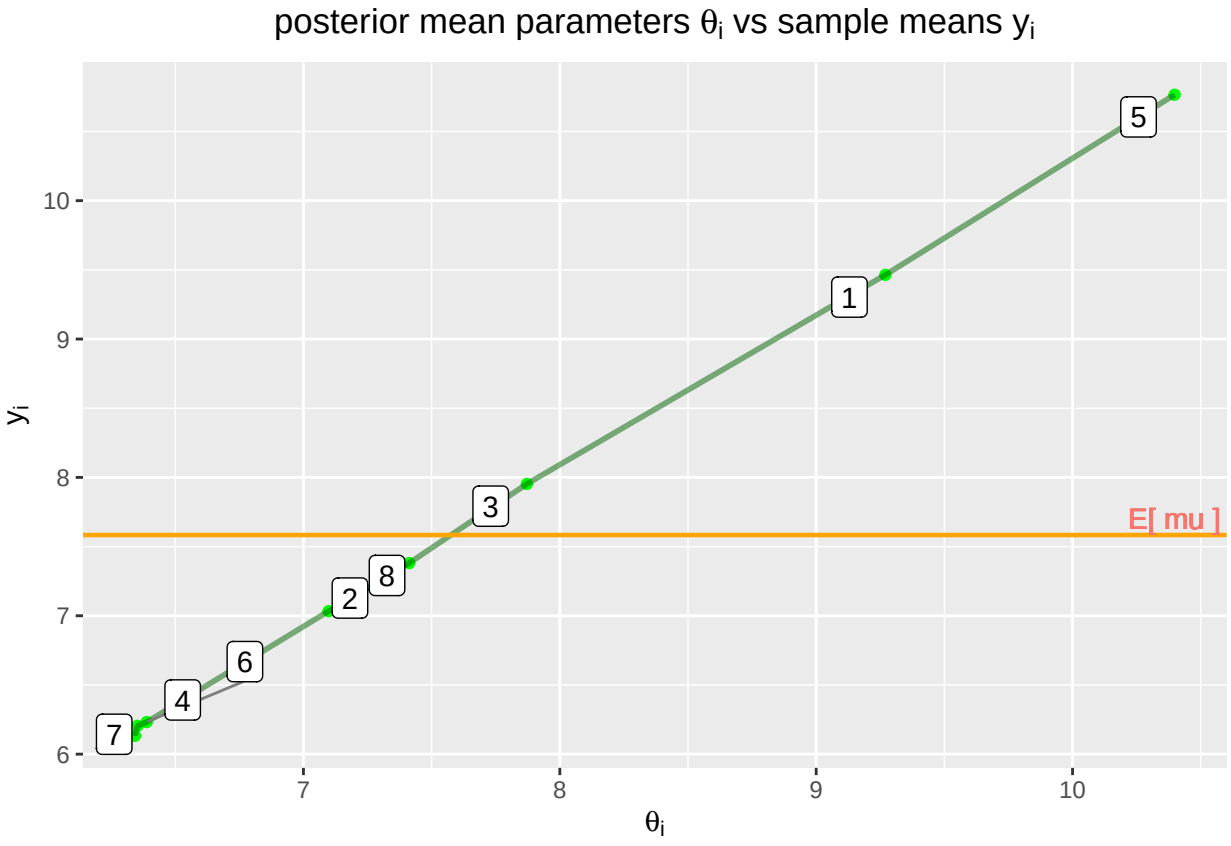
As it is expected, the probability of the second event is slightly smaller than the first one, since it takes into account sample variability.

Point e.

Plot the sample averages $\bar{y}_1, \ldots, \bar{y}_8$ against the posterior expectations of $\theta_1, \ldots, \theta_8$, and describe the relationship. Also compute the sample mean of all observations and compare it to the posterior mean of μ.

By plotting the sample averages $\bar{y}_i$ against the posterior expectation values of the θ_i, we observe that they appear to satisfy a linear relation.

```
library(ggrepel)

summary_theta = summary(theta_mcmc)
post_means = summary_theta$statistics[1:8, 1]
ybar = as.vector(ybar)
mean_plt = data.frame(post_means, names = 1:8)

mean_plt %>% ggplot(aes(x = post_means, y = ybar)) +
    geom_point(col = "green") +
    geom_line(col = "darkgreen", alpha = 0.5, size = 1) +
    geom_hline(yintercept = summary_mu$statistics[1], col = "orange", size = 0.8) +
    geom_text(aes(x = post_means[5], y = summary_mu$statistics[1], label = paste("E[", "mu", "]"), colour = "orange"), vjust = -0.3
    ylab(expression(y[i])) +
    xlab(expression(paste(theta[i]))) +
    ggtitle(expression(paste("posterior mean parameters ", theta[i], " vs sample means ", y[i]))) +
    scale_color_discrete(guide = "none") +
    geom_label_repel(aes(label = names),
        box.padding   = 0.35,
        point.padding = 0.5,
        segment.color = 'grey50')
```



```
total_sample_mean = sum(n * ybar) / sum(n)
```

The total sample mean is 7.69 and the posterior expectation of the Grand Mean is 7.58, which means that they only differ by 5.38% of the former.
This indicates that the data effect is strong on the Grand Mean, since it is converging to the general sample mean, even though the convergence is quite slow.