ST540 HW7

4. (a) The kernel of the full posterior can be expressed as:

$$P(\sigma_1^2, ..., \sigma_n^2, b|Y_1, ..., Y_n) \propto P(Y_1, ..., Y_n|\sigma_1^2, ..., \sigma_n^2, b)P(\sigma_1^2, ..., \sigma_n^2|b)P(b) \tag{1}$$

Note that $\{Y_i\}_{i=1}^n$ are independent of each other given $\{\sigma_i^2\}_{i=1}^n$, and $\{\sigma_i^2\}_{i=1}^n$ are independent of each other given b:

$$P(\sigma_1^2, ..., \sigma_n^2, b|Y_1, ..., Y_n) \propto [\prod_{i=1}^n P(Y_i|\sigma_i^2)P(\sigma_i^2|b)]P(b) \tag{2}$$

The full conditional posterior of σ_1^2 can be obtained by treating all of the other parameters as constants and discarding any term that doesn't involve σ_1^2:

$$\begin{aligned} P(\sigma_1^2|\sigma_2^2..., \sigma_n^2, b, Y_1, ..., Y_n) &\propto P(Y_1|\sigma_1^2)P(\sigma_1^2|b) \\ &= \mathrm{N}(Y_1|0, \sigma_1^2)\mathrm{InvGamma}(1, b) \quad \text{(conjugate pair)} \end{aligned} \tag{3}$$

Therefore, $P(\sigma_1^2|\sigma_2^2..., \sigma_n^2, b, Y_1, ..., Y_n) = \mathrm{InvGamma}(1 + \frac{1}{2}, b + \frac{Y_1^2}{2})$.
Similarity,

$$\begin{aligned} P(b|\sigma_1^2..., \sigma_n^2, Y_1, ..., Y_n) &\propto [\prod_{i=1}^n P(\sigma_i^2|b)]P(b) \\ &\propto [\prod_{i=1}^n be^{-b/\sigma_i^2}]e^{-b} \\ &= b^n e^{-b(1+\sum_{i=1}^n 1/\sigma_i^2)} \end{aligned} \tag{4}$$

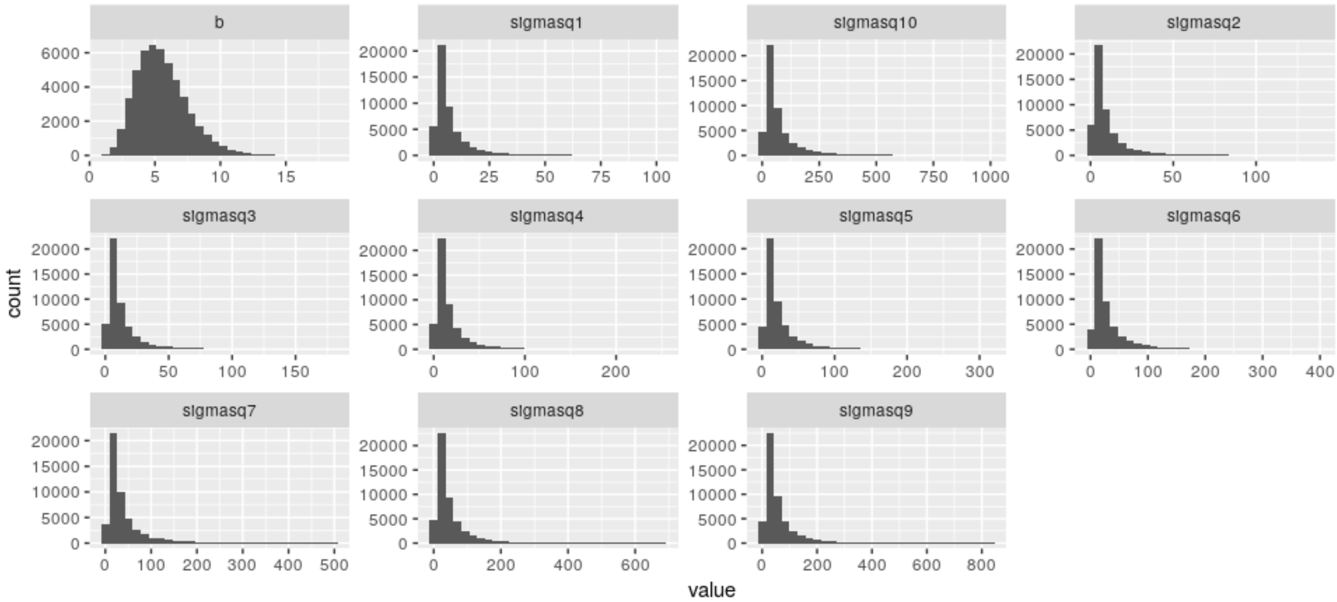
Therefore, $b|\sigma_1^2..., \sigma_n^2, Y_1, ..., Y_n \sim \mathrm{Gamma}(1 + n, 1 + \sum_{i=1}^n 1/\sigma_i^2)$

(b) Pseudo code:

initialization $b^{(0)}, \sigma_1^{2(0)}, ..., \sigma_{10}^{2(0)}$;
for $t \leftarrow 1$ **to** N **by** 1 **do**
  for $i \leftarrow 1$ **to** 10 **by** 1 **do**
    sample $\sigma_i^{2(t)} \sim \mathrm{InvGamma}(1 + \frac{1}{2}, b^{(t-1)} + \frac{Y_i^2}{2})$;
  end
  sample $b^{(t)} \sim \mathrm{Gamma}(1 + 10, 1 + \sum_{i=1}^{10} 1/\sigma_i^{2(t)})$;
end

Algorithm 1: Gibbs Sampler

(c) The code is attached.




```
library(invgamma)
library(purrr)
library(tidyr)
library(ggplot2)
# data
Y = seq(1, 10)
# number of iterations
N = 200000
# number of burn-ins
burnin = 150000
# matrix to save the posterior samples
samples = matrix(NA, N - burnin, 10 + 1)
colnames(samples) = c(paste('sigmasq', seq(1,10), sep = ""), c("b"))
# initialization
sigmasq = rep(0.1, 10)
b = 1
# Gibbs sampling
for(t in 1:N){
  # sigmasq = 1 / rgamma(10, shape = 3/2, rate = b + Y^2/2)
  sigmasq = rinvgamma(10, shape = 3/2, rate = b + Y^2/2)
  b = rgamma(1, shape=1+10, rate=1 + sum(1/sigmasq))
  if(t > burnin)
    samples[t-burnin,] = c(sigmasq,b)
}


plot_df = as.data.frame(samples)
# remove the sigma^2 that are larger than 99% quantile for plotting
for(i in 1:10) {
  plot_df[, i][plot_df[, i] > quantile(plot_df[, i], probs = (0.99))] = NA
}

plot_df %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram()
```