STA610 - HW9

Zhankai Ye

Nov 13, 2024

Q1

(a)

- Full Conditional of μ

The posterior distribution of μ given $\{\theta_j\}$ and τ^2 is proportional to its prior multiplied by the likelihood from θ_j :

$$p\left(\mu \mid \{\theta_j\}, \tau^2\right) \propto p(\mu) \prod_{j=1}^{m} p\left(\theta_j \mid \mu, \tau^2\right)$$

Since

$$p(\mu)=\frac{1}{\sqrt{2\pi v_0}} \exp\left(-\frac{1}{2v_0}\left(\mu-\mu_0\right)^2\right)$$

$$p\left(\theta_j \mid \mu, \tau^2\right)=\frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{1}{2\tau^2}\left(\theta_j-\mu\right)^2\right)$$

Then,

$$\begin{aligned}
p(\mu \mid \{\theta_j\}, \tau^2) &\propto \exp\left(-\frac{1}{2v_0}\left(\mu-\mu_0\right)^2-\sum_{j=1}^{m} \frac{1}{2\tau^2}\left(\theta_j-\mu\right)^2\right) \\
&=\exp\left(-\frac{1}{2}\left[\left(\frac{1}{v_0}+\frac{m}{\tau^2}\right)\mu^2-2\mu\left(\frac{\mu_0}{v_0}+\frac{\sum_{j=1}^{m}\theta_j}{\tau^2}\right)+\text{ constants }\right]\right) \\
&\propto \exp\left(-\frac{1}{2V_\mu}\left(\mu-\mu_\mu\right)^2\right)
\end{aligned}$$

where

$$V_\mu=\left(\frac{1}{v_0}+\frac{m}{\tau^2}\right)^{-1}$$

$$\mu_\mu=V_\mu\left(\frac{\mu_0}{v_0}+\frac{\sum_{j=1}^{m}\theta_j}{\tau^2}\right)$$

- Full Conditional of τ^2

The posterior of τ^2 given $\{\theta_j\}$ and μ is proportional to its prior and the likelihood from θ_j :

$$p\left(\left.\frac{1}{\tau^2}\right| \{\theta_j\}, \mu\right) \propto p\left(\frac{1}{\tau^2}\right) \prod_{j=1}^{m} p\left(\theta_j \mid \mu, \tau^2\right)$$

Denote $\lambda=\frac{1}{\tau^2}$, then

$$p(\lambda)=\operatorname{Gamma}\left(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2}\right)=\frac{\left(\frac{\eta_0\tau_0^2}{2}\right)^{\frac{\eta_0}{2}}}{\Gamma\left(\frac{\eta_0}{2}\right)} \lambda^{\frac{\eta_0}{2}-1} \exp\left(-\frac{\eta_0\tau_0^2}{2}\lambda\right)$$

$$\prod_{j=1}^{m} p\left(\theta_j \mid \mu, \lambda\right)=\left(\frac{\lambda}{2 \pi}\right)^{\frac{m}{2}} \exp \left(-\frac{\lambda}{2} \sum_{j=1}^{m}\left(\theta_j-\mu\right)^2\right)$$

Then,

$$\begin{aligned}
p\left(\lambda \mid\left\{\theta_j\right\}, \mu\right) & \propto \lambda^{\frac{\eta_0}{2}-1} \exp \left(-\frac{\eta_0 \tau_0^2}{2} \lambda\right) \times \lambda^{\frac{m}{2}} \exp \left(-\frac{\lambda}{2} \sum_{j=1}^{m}\left(\theta_j-\mu\right)^2\right) \\
& \propto \lambda^{\frac{\eta_0+m}{2}-1} \exp \left(-\lambda\left(\frac{\eta_0 \tau_0^2+\sum_{j=1}^{m}\left(\theta_j-\mu\right)^2}{2}\right)\right)
\end{aligned}$$

Therefore,

$$\frac{1}{\tau^2} \mid\left\{\theta_j\right\}, \mu \sim \operatorname{Gamma}\left(\alpha_\tau, \beta_\tau\right)$$

where

$$\begin{aligned}
\alpha_\tau & =\frac{\eta_0+m}{2} \\
\beta_\tau & =\frac{\eta_0 \tau_0^2+\sum_{j=1}^{m}\left(\theta_j-\mu\right)^2}{2}
\end{aligned}$$

Or equivalently:

$$\tau^2 \mid\left\{\theta_j\right\}, \mu \sim \text{Inverse-Gamma}\left(\alpha_\tau, \beta_\tau\right)$$

- Full Conditional of σ^2

The posterior distribution of $1/\sigma^2$ is proportional to its prior multiplied by the likelihood from all $y_{i,j}$:

$$p\left(\left.\frac{1}{\sigma^2} \right\rvert\,\left\{y_{i, j}\right\},\left\{\theta_j\right\}\right) \propto p\left(\frac{1}{\sigma^2}\right) \prod_{j=1}^{m} \prod_{i=1}^{n_j} p\left(y_{i, j} \mid \theta_j, \sigma^2\right)$$

Denote $\kappa=\frac{1}{\sigma^2}$, then

$$\begin{aligned}
p(\kappa)=\operatorname{Gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0 \sigma_0^2}{2}\right) & =\frac{\left(\frac{\nu_0 \sigma_0^2}{2}\right)^{\frac{\nu_0}{2}}}{\Gamma\left(\frac{\nu_0}{2}\right)} \kappa^{\frac{\nu_0}{2}-1} \exp \left(-\frac{\nu_0 \sigma_0^2}{2} \kappa\right) \\
\prod_{j=1}^{m} \prod_{i=1}^{n_j} p\left(y_{i, j} \mid \theta_j, \kappa\right) & =\left(\frac{\kappa}{2 \pi}\right)^{\frac{N}{2}} \exp \left(-\frac{\kappa}{2} \sum_{j=1}^{m} \sum_{i=1}^{n_j}\left(y_{i, j}-\theta_j\right)^2\right)
\end{aligned}$$

where $N=\sum_{j=1}^{m} n_j$ is the total number of observations. Then,

$$\begin{aligned}
p\left(\kappa \mid\left\{y_{i, j}\right\},\left\{\theta_j\right\}\right) & \propto \kappa^{\frac{\nu_0}{2}-1} \exp \left(-\frac{\nu_0 \sigma_0^2}{2} \kappa\right) \times \kappa^{\frac{N}{2}} \exp \left(-\frac{\kappa}{2} \sum_{j=1}^{m} \sum_{i=1}^{n_j}\left(y_{i, j}-\theta_j\right)^2\right) \\
& \propto \kappa^{\frac{\nu_0+N}{2}-1} \exp \left(-\kappa\left(\frac{\nu_0 \sigma_0^2+\sum_{j=1}^{m} \sum_{i=1}^{n_j}\left(y_{i, j}-\theta_j\right)^2}{2}\right)\right)
\end{aligned}$$

Therefore,

$$\frac{1}{\sigma^2} \mid\left\{y_{i, j}\right\},\left\{\theta_j\right\} \sim \operatorname{Gamma}\left(\alpha_\sigma, \beta_\sigma\right)$$

where

$$\begin{aligned}
\alpha_\sigma & =\frac{\nu_0+N}{2} \\
\beta_\sigma & =\frac{\nu_0 \sigma_0^2+\sum_{j=1}^{m} \sum_{i=1}^{n_j}\left(y_{i, j}-\theta_j\right)^2}{2}
\end{aligned}$$

Or equivalently:

$$\sigma^2 \mid\left\{y_{i, j}\right\},\left\{\theta_j\right\} \sim \text{Inverse-Gamma}\left(\alpha_\sigma, \beta_\sigma\right)$$

- Full Conditional of each θ_j

The posterior of θ_j given μ, τ^2, σ^2, and $\{y_{i,j}\}$ is proportional to its prior and the likelihood from $y_{i,j}$:

$$p\left(\theta_j \mid \mu, \tau^2, \sigma^2, \{y_{i,j}\}\right) \propto p\left(\theta_j \mid \mu, \tau^2\right) \prod_{i=1}^{n_j} p\left(y_{i,j} \mid \theta_j, \sigma^2\right)$$

Since

$$p\left(\theta_j \mid \mu, \tau^2\right) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{1}{2\tau^2}\left(\theta_j - \mu\right)^2\right)$$

$$\prod_{i=1}^{n_j} p\left(y_{i,j} \mid \theta_j, \sigma^2\right) \propto \left(\sigma^2\right)^{-n_j/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n_j}\left(y_{i,j} - \theta_j\right)^2\right)$$

Therefore,

$$\begin{aligned} p\left(\theta_j \mid \mu, \tau^2, \sigma^2, \{y_{i,j}\}\right) &\propto \exp\left(-\frac{1}{2\tau^2}\left(\theta_j - \mu\right)^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n_j}\left(y_{i,j} - \theta_j\right)^2\right) \\ &= \exp\left(-\frac{1}{2}\left[\left(\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}\right)\theta_j^2 - 2\theta_j\left(\frac{\mu}{\tau^2} + \frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}\right) + \text{ constants }\right]\right) \end{aligned}$$

Therefore,

$$\theta_j \mid \mu, \tau^2, \sigma^2, \{y_{i,j}\} \sim N\left(\theta_j^*, V_j\right)$$

where

$$V_j = \left(\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}\right)^{-1}$$

$$\theta_j^* = V_j\left(\frac{\mu}{\tau^2} + \frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}\right)$$

(b)

- For μ:

From part (a), we know the full conditional distribution of μ is:

$$\mu \mid \{\theta_j\}, \tau^2 \sim N\left(\mu_\mu, V_\mu\right)$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[\mu \mid \{\theta_j\}, \tau^2\right] &= \mu_\mu \\ &= V_\mu\left(\frac{\mu_0}{v_0} + \frac{\sum_{j=1}^{m}\theta_j}{\tau^2}\right) \\ &= \left(\frac{1}{\frac{1}{v_0} + \frac{m}{\tau^2}}\right)\left(\frac{\mu_0}{v_0} + \frac{\sum_{j=1}^{m}\theta_j}{\tau^2}\right) \end{aligned}$$

This shows that μ_μ is a weighted average of the prior mean μ_0 and the sample mean of θ_j:

$$\mu_\mu = w_{\text{prior}}\,\mu_0 + w_{\text{data}}\,\bar{\theta}$$

where

$$\bar{\theta} = \frac{1}{m}\sum_{j=1}^{m}\theta_j$$

$$w_{\text{prior}} = \frac{\frac{1}{v_0}}{\frac{1}{v_0} + \frac{m}{\tau^2}}$$

$$w_{\text{data}} = \frac{\frac{m}{\tau^2}}{\frac{1}{v_0} + \frac{m}{\tau^2}}$$

- For τ^2:

From (a), we know that

$$\tau^2 \mid \{\theta_j\}, \mu \sim \text{Inverse-Gamma}\left(\alpha_\tau, \beta_\tau\right)$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\tau^2 \mid \{\theta_j\}, \mu\right] &= \frac{\beta_\tau}{\alpha_\tau - 1} \\
&= \frac{\beta_\tau}{\left(\frac{\eta_0+m}{2}\right) - 1} \\
&= \frac{2\beta_\tau}{\eta_0 + m - 2} \\
&= \frac{\eta_0\tau_0^2 + \sum_{j=1}^m \left(\theta_j - \mu\right)^2}{\eta_0 + m - 2}
\end{aligned}$$

$\eta_0\tau_0^2$ represents the prior belief about the variance among the parameters θ_j, $\sum_{j=1}^m \left(\theta_j - \mu\right)^2$ is the sum of squared deviations of the parameters from the hyperparameter μ and represents the influence from the data. Therefore, the expectation is a weighted average of the prior variance and the sample variance among θ_j, adjusted by their respective degrees of freedom.

- For σ^2:

$$\begin{aligned}
\mathbb{E}\left[\sigma^2 \mid \{y_{i,j}\}, \{\theta_j\}\right] &= \frac{\beta_\sigma}{\alpha_\sigma - 1} \\
&= \frac{\beta_\sigma}{\left(\frac{\nu_0+N}{2}\right) - 1} \\
&= \frac{2\beta_\sigma}{\nu_0 + N - 2} \\
&= \frac{\nu_0\sigma_0^2 + \sum_{j=1}^m \sum_{i=1}^{n_j} \left(y_{i,j} - \theta_j\right)^2}{\nu_0 + N - 2}
\end{aligned}$$

The expectation is also a weighted average of the prior variance and the sample variance, adjusted by their respective degrees of freedom.

- For θ_j:

$$\begin{aligned}
\mathbb{E}\left[\theta_j \mid \mu, \tau^2, \sigma^2, \{y_{i,j}\}\right] = \theta_j^* &= \left(\frac{1}{\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}}\right)\left(\frac{\mu}{\tau^2} + \frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}\right) \\
&= w_{\text{prior}}\, \mu + w_{\text{data}}\, \bar{y}_j
\end{aligned}$$

where

$$\begin{aligned}
\bar{y}_j &= \frac{1}{n_j}\sum_{i=1}^{n_j} y_{i,j} \\
w_{\text{prior}} &= \frac{\frac{1}{\tau^2}}{\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}} \\
w_{\text{data}} &= \frac{\frac{n_j}{\sigma^2}}{\frac{1}{\tau^2} + \frac{n_j}{\sigma^2}}
\end{aligned}$$

This reflects the influence of both the prior belief about θ_j (through μ and τ^2) and the observed data for group j.

(c) The pseudocode is as follows:

Algorithm 1 Gibbs Sampling for Hierarchical Normal Model

Input: Data $\{y_{i,j}\}$; Hyperparameters μ_0, v_0, ν_0, σ_0^2, η_0, τ_0^2; Number of iterations T; Initial values $\mu^{(0)}$, $\tau^{2(0)}$, $\sigma^{2(0)}$, $\theta_j^{(0)}$.
Output: Samples $\{\mu^{(t)}, \tau^{2(t)}, \sigma^{2(t)}, \theta_j^{(t)}\}$ for $t = 1$ to T.
Initialize: Set $\mu^{(0)}$, $\tau^{2(0)}$, $\sigma^{2(0)}$; For all j, set $\theta_j^{(0)}$.
for $t = 1$ to T **do**
  for $j = 1$ to m **do**
    Compute $V_j^{(t)} = \left(\frac{1}{\tau^{2(t-1)}} + \frac{n_j}{\sigma^{2(t-1)}}\right)^{-1}$.
    Let $S_j = \sum_{i=1}^{n_j} y_{i,j}$.
    Compute $\theta_j^{*(t)} = V_j^{(t)}\left(\frac{\mu^{(t-1)}}{\tau^{2(t-1)}} + \frac{S_j}{\sigma^{2(t-1)}}\right)$.
    Sample $\theta_j^{(t)} \sim \mathcal{N}\left(\theta_j^{*(t)},\ V_j^{(t)}\right)$.
  end for
  Compute $V_\mu^{(t)} = \left(\frac{1}{v_0} + \frac{m}{\tau^{2(t-1)}}\right)^{-1}$.
  Compute $\mu^{*(t)} = V_\mu^{(t)}\left(\frac{\mu_0}{v_0} + \frac{\sum_{j=1}^m \theta_j^{(t)}}{\tau^{2(t-1)}}\right)$.
  Sample $\mu^{(t)} \sim \mathcal{N}\left(\mu^{*(t)},\ V_\mu^{(t)}\right)$.
  Compute $\alpha_\tau = \frac{\eta_0 + m}{2}$ and $\beta_\tau^{(t)} = \frac{\eta_0\tau_0^2 + \sum_{j=1}^m\left(\theta_j^{(t)} - \mu^{(t)}\right)^2}{2}$.
  Sample $1/\tau^{2(t)} \sim \text{Gamma}\left(\alpha_\tau,\ \beta_\tau^{(t)}\right)$; Set $\tau^{2(t)} = 1/\left(1/\tau^{2(t)}\right)$.
  Compute $N = \sum_{j=1}^m n_j$, $\alpha_\sigma = \frac{\nu_0 + N}{2}$.
  Compute $\beta_\sigma^{(t)} = \frac{\nu_0\sigma_0^2 + \sum_{j=1}^m\sum_{i=1}^{n_j}\left(y_{i,j} - \theta_j^{(t)}\right)^2}{2}$.
  Sample $1/\sigma^{2(t)} \sim \text{Gamma}\left(\alpha_\sigma,\ \beta_\sigma^{(t)}\right)$; Set $\sigma^{2(t)} = 1/\left(1/\sigma^{2(t)}\right)$.
end for
Return: Samples $\{\mu^{(t)},\ \tau^{2(t)},\ \sigma^{2(t)},\ \theta_j^{(t)}\}$ for $t = 1$ to T.

The Gibbs sampling process for the hierarchical normal model involves iteratively updating each parameter-namely, the overall mean μ, the between-group variance τ^2, the within-group variance σ^2, and the group-specific means θ_j for $j = 1, \ldots, m$. Starting with initial estimates, the algorithm proceeds by first sampling each θ_j from its conditional distribution given the current values of μ, τ^2, σ^2, and the observed data $y_{i,j}$ for group j. Next, it updates the overall mean μ by sampling from its conditional distribution based on the updated θ_j and current τ^2. Then, it samples the between-group variance τ^2 using the latest θ_j and μ. After that, the withingroup variance σ^2 is updated by sampling from its conditional distribution, which depends on the observed data $y_{i,j}$ and the updated θ_j. This sequence of updates constitutes one iteration of the sampler, and the process is repeated for a large number of iterations, allowing each parameter to be updated based on the most recent information from the others and from the data, ultimately generating samples that approximate the joint posterior distribution of all the parameters in the model.

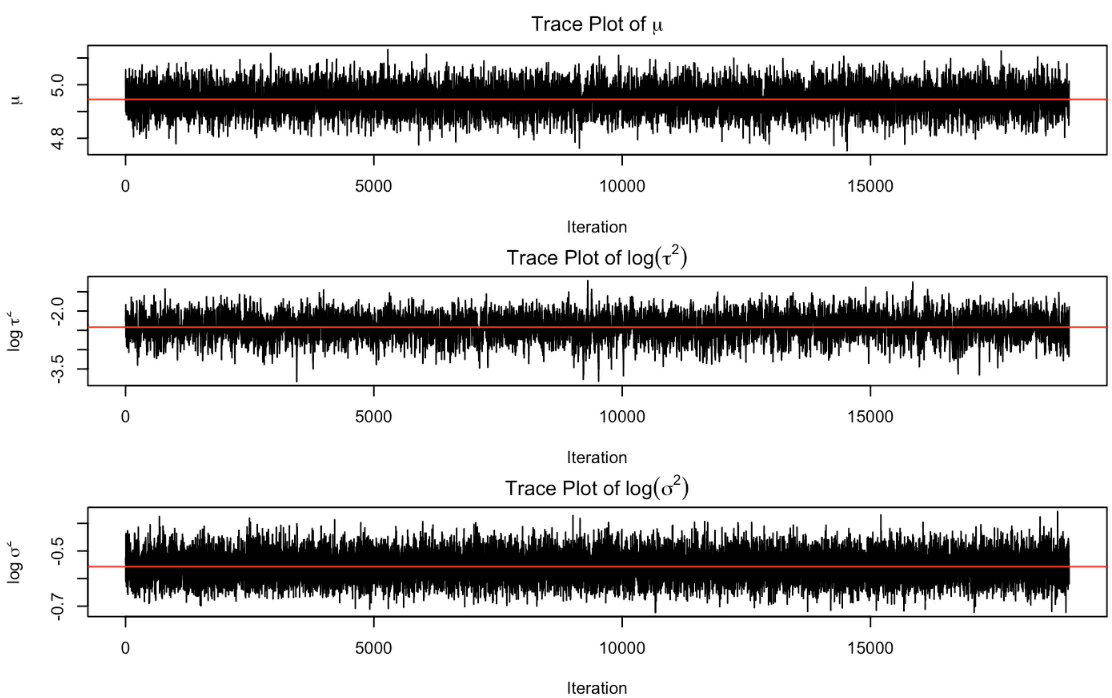
Q2

(a) The code is as follows:

```r
library(ggplot2)

radonMN <- readRDS("/Users/moukaii/Downloads/radonMN.Rds")
radonMN$log_radon <- log(radonMN$radon)
```

```

counties <- sort(unique(radonMN$county))
m <- length(counties)

n_j <- sapply(counties, function(c) sum(radonMN$county == c))
y_j <- split(radonMN$log_radon, factor(radonMN$county, levels = counties))

mu_0 <- mean(radonMN$log_radon)
v_0 <- 1e6
eta_0 <- 0.002
tau_0_sq <- 1
nu_0 <- 0.002
sigma_0_sq <- 1

# Number of iterations and burn-in
T <- 20000
burn_in <- 1000
thin <- 1

# Initialization
mu_samples <- numeric(T)
tau_sq_samples <- numeric(T)
sigma_sq_samples <- numeric(T)
theta_samples <- matrix(0, nrow = T, ncol = m)

mu <- mu_0
tau_sq <- tau_0_sq
sigma_sq <- sigma_0_sq
theta_j <- sapply(y_j, mean)  # Initial theta_j set to sample means

set.seed(42)

for (t in 1:T) {
  for (j in 1:m) {
    V_theta_j <- 1 / (1 / tau_sq + n_j[j] / sigma_sq)
    mu_theta_j <- V_theta_j * (mu / tau_sq + sum(y_j[[j]]) / sigma_sq)
    theta_j[j] <- rnorm(1, mean = mu_theta_j, sd = sqrt(V_theta_j))
  }

  V_mu <- 1 / (1 / v_0 + m / tau_sq)
  mu_mu <- V_mu * (mu_0 / v_0 + sum(theta_j) / tau_sq)
  mu <- rnorm(1, mean = mu_mu, sd = sqrt(V_mu))

  alpha_tau <- (eta_0 / 2) + (m / 2)       # Shape parameter
  beta_tau <- (eta_0 * tau_0_sq / 2) + sum((theta_j - mu)^2) / 2  # Rate parameter
  inv_tau_sq <- rgamma(1, shape = alpha_tau, rate = beta_tau)
  tau_sq <- 1 / inv_tau_sq

  N <- length(radonMN$log_radon)
  alpha_sigma <- (nu_0 / 2) + (N / 2)    # Shape parameter
  sum_sq <- 0
  for (j in 1:m) {
    sum_sq <- sum_sq + sum((y_j[[j]] - theta_j[j])^2)
  }
```

```r
  beta_sigma <- (nu_0 * sigma_0_sq / 2) + sum_sq / 2  # Rate parameter
  inv_sigma_sq <- rgamma(1, shape = alpha_sigma, rate = beta_sigma)
  sigma_sq <- 1 / inv_sigma_sq

  mu_samples[t] <- mu
  tau_sq_samples[t] <- tau_sq
  sigma_sq_samples[t] <- sigma_sq
  theta_samples[t, ] <- theta_j

  if (t %% 1000 == 0) {
    cat("Iteration:", t, "mu:", mu, "tau_sq:", tau_sq, "sigma_sq:", sigma_sq, "\n")
  }
}

indices <- seq(burn_in + 1, T, by = thin)
mu_samples <- mu_samples[indices]
tau_sq_samples <- tau_sq_samples[indices]
sigma_sq_samples <- sigma_sq_samples[indices]
theta_samples <- theta_samples[indices, ]

par(mfrow = c(3, 1), mar = c(4, 4, 2, 1))
plot(mu_samples, type = 'l', main = expression("Trace Plot of "*mu), xlab = "Iteration",
    ↪ ylab = expression(mu))
abline(h = mean(mu_samples), col = 'red')
plot(log(tau_sq_samples), type = 'l', main = expression("Trace Plot of "*log(tau^2)),
    ↪ xlab = "Iteration", ylab = expression(log~tau^2))
abline(h = mean(log(tau_sq_samples)), col = 'red')
plot(log(sigma_sq_samples), type = 'l', main = expression("Trace Plot of "*log(sigma^2)),
    ↪  xlab = "Iteration", ylab = expression(log~sigma^2))
abline(h = mean(log(sigma_sq_samples)), col = 'red')
```



Overall, the trace plots suggest that the Gibbs sampler is mixing well for all three parameters.

(b) The code is as follows:

```
# Compute 95% credible intervals
ci_mu_gibbs <- quantile(mu_samples, probs = c(0.025, 0.975))
ci_tau_sq_gibbs <- quantile(tau_sq_samples, probs = c(0.025, 0.975))
ci_sigma_sq_gibbs <- quantile(sigma_sq_samples, probs = c(0.025, 0.975))

# Display the credible intervals
cat("95% Credible Interval for \mu from Gibbs Sampler:\n")
print(ci_mu_gibbs)

cat("\n95% Credible Interval for \tau^2 from Gibbs Sampler:\n")
print(ci_tau_sq_gibbs)

cat("\n95% Credible Interval for \sigma^2 from Gibbs Sampler:\n")
print(ci_sigma_sq_gibbs)
```

```
95% Credible Interval for \mu from Gibbs Sampler:
    2.5%    97.5%
4.853211 5.038932

95% Credible Interval for \tau^2 from Gibbs Sampler:
      2.5%      97.5%
0.04757508 0.15640561

95% Credible Interval for \sigma^2 from Gibbs Sampler:
     2.5%     97.5%
0.5225333 0.6317875
```

```
library(lme4)

# Fit the linear mixed-effects model
lmer_model <- lmer(log_radon ~ 1 + (1 | county), data = radonMN)

confint_lmer <- confint(lmer_model, parm = "(Intercept)",
                        method = "profile", level = 0.95)

ci_mu_lmer <- confint_lmer["(Intercept)", ]

cat("\n95% Confidence Interval from lmer:\n")
print(ci_mu_lmer)
```

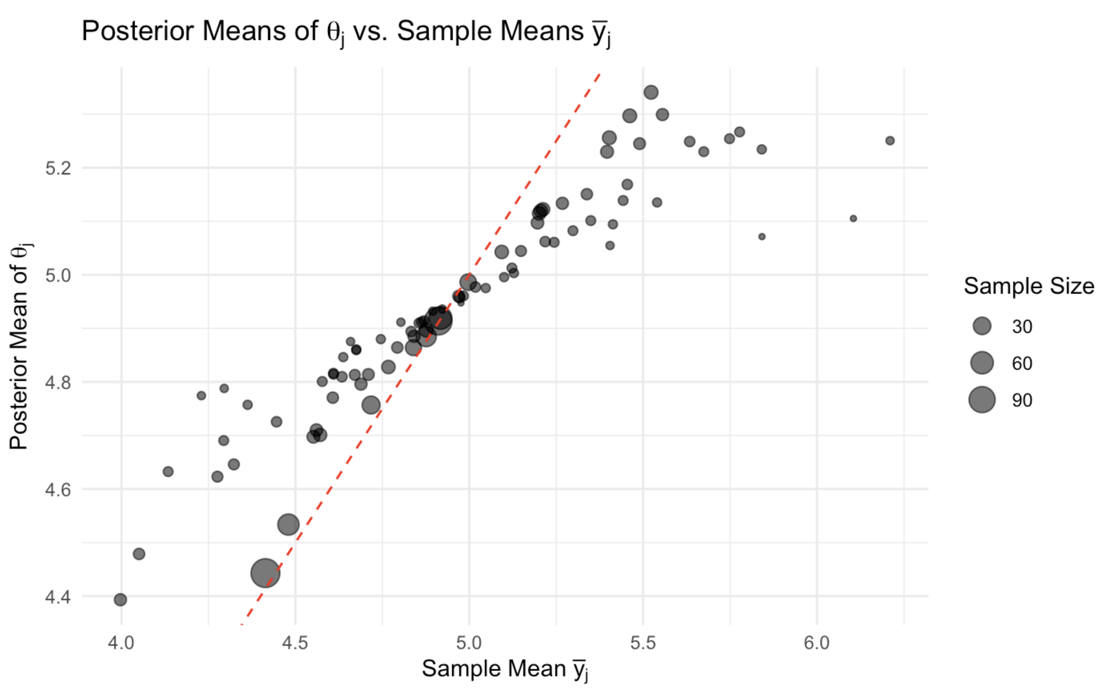
```
95% Confidence Interval from lmer:
   2.5 %   97.5 %
4.854804 5.040563
```

The estimates and confidence intervals are very similiar, indicating that when non-informative or diffuse priors are used in the Bayesian approach (as in your Gibbs sampler), the posterior distributions are heavily influenced by the data, leading to estimates that align closely with frequentist methods like lmer.

(c) The code is as follows:

```
# Compute posterior means of theta_j
theta_posterior_means <- colMeans(theta_samples)
```

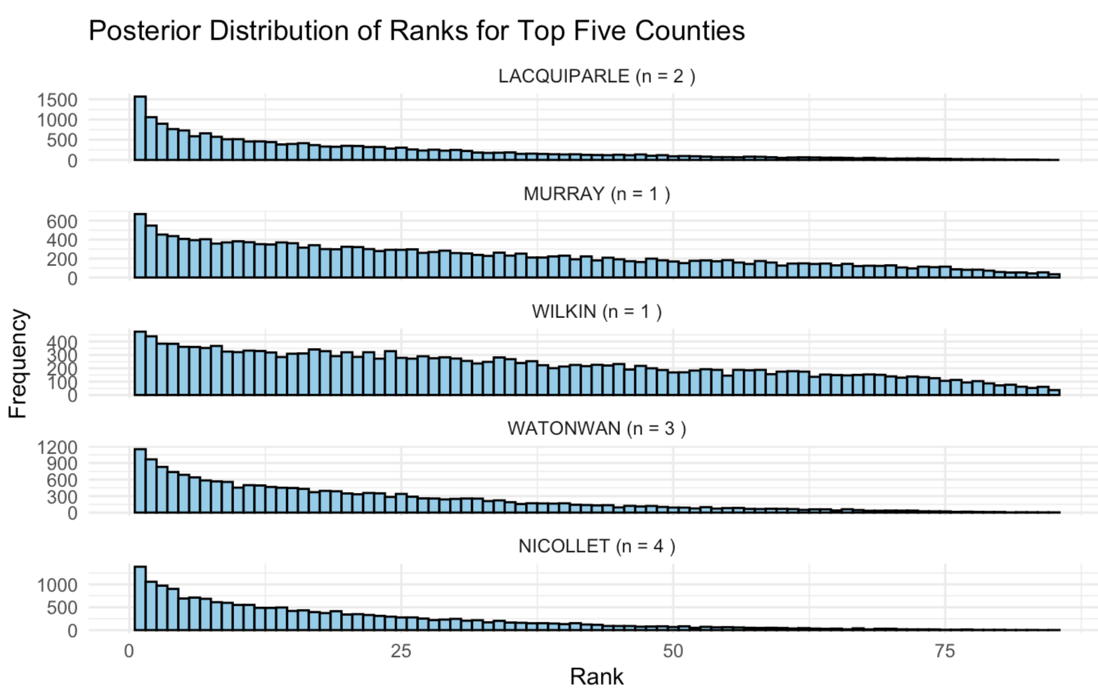
```

theta_df <- data.frame(
  county = counties,
  theta_posterior_mean = theta_posterior_means
)

# Compute sample means for each county
sample_means <- sapply(y_j, mean)


theta_df$sample_mean <- sample_means
theta_df$sample_size <- n_j

ggplot(theta_df, aes(x = sample_mean, y = theta_posterior_mean)) +
  geom_point(aes(size = sample_size), alpha = 0.6) +
  geom_abline(intercept = 0, slope = 1, color = ’red’, linetype = ’dashed’) +
  labs(
    title = expression("Posterior Means of "*theta[j]*" vs. Sample Means "*bar(y)[j]),
    x = expression("Sample Mean "*bar(y)[j]),
    y = expression("Posterior Mean of "*theta[j]),
    size = "Sample Size"
  ) +
  theme_minimal()
```



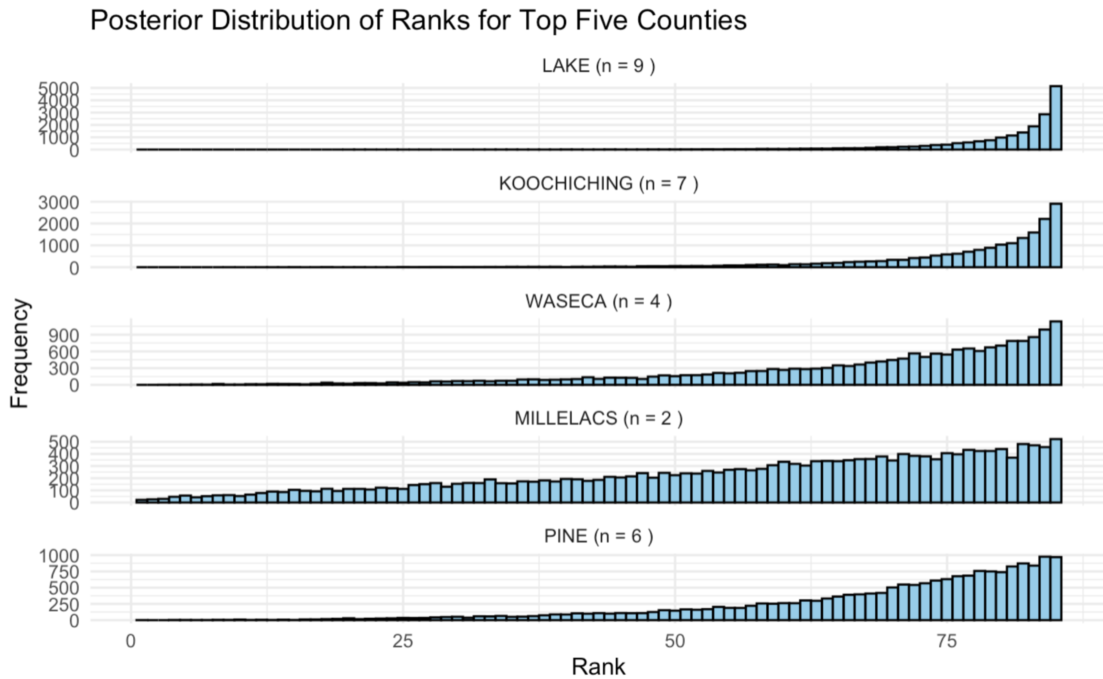
The plot illustrates a shrinkage effect typical of hierarchical Bayesian models, where the posterior means of county-level parameters θ_j are pulled towards the overall mean, especially for counties with smaller sample sizes. Counties with larger sample sizes have posterior means closer to their sample means, reflecting greater confidence in those estimates. This balancing act between the individual county data and the overall distribution stabilizes estimates, preventing extreme sample means from unduly influencing posterior estimates for counties with limited data, thereby providing more robust and reliable inferences.

(d) The code is as follows:

```
theta_df_sorted <- theta_df[order(-theta_df$sample_mean), ]
top_five_counties <- theta_df_sorted$county[1:5]

theta_ranks <- matrix(0, nrow = nrow(theta_samples), ncol = m)
colnames(theta_ranks) <- counties

for (i in 1:nrow(theta_samples)) {
  ranks <- rank(-theta_samples[i, ], ties.method = "first")
  theta_ranks[i, ] <- ranks
}

top_five_ranks <- theta_ranks[, top_five_counties]

library(reshape2)
ranks_df <- data.frame(top_five_ranks)
ranks_df$Iteration <- 1:nrow(ranks_df)
ranks_long <- melt(ranks_df, id.vars = "Iteration", variable.name = "County",
                   value.name = "Rank")

sample_sizes <- setNames(theta_df$sample_size, theta_df$county)
ranks_long$County <- factor(ranks_long$County,levels = top_five_counties,
                            labels = paste(top_five_counties,
                                           "(n =", sample_sizes[top_five_counties], ")"))

ggplot(ranks_long, aes(x = Rank)) +
  geom_histogram(binwidth = 1, fill = 'skyblue', color = 'black') +
  facet_wrap(~ County, ncol = 1, scales = "free_y") +
  labs(title = "Posterior Distribution of Ranks for Top Five Counties",
    x = "Rank", y = "Frequency") +
  theme_minimal()
```



```r
theta_df_sorted <- theta_df[order(theta_df$sample_mean), ]
bottom_five_counties <- theta_df_sorted$county[1:5]
print("Bottom Five Counties by Sample Means:")
print(bottom_five_counties)

# Initialize a matrix to store ranks
theta_ranks <- matrix(0, nrow = nrow(theta_samples), ncol = m)
colnames(theta_ranks) <- counties

# Compute ranks for each iteration
for (i in 1:nrow(theta_samples)) {
  ranks <- rank(-theta_samples[i, ], ties.method = "first")
  theta_ranks[i, ] <- ranks
}

# Extract ranks for the top five counties
top_five_ranks <- theta_ranks[, bottom_five_counties]

# Reshape data for plotting
library(reshape2)
ranks_df <- data.frame(top_five_ranks)
ranks_df$Iteration <- 1:nrow(ranks_df)
ranks_long <- melt(ranks_df, id.vars = "Iteration", variable.name = "County",
                   value.name = "Rank")

# Assuming sample sizes are stored in a named vector `sample_sizes` with county names as names
sample_sizes <- setNames(theta_df$sample_size, theta_df$county)

# Modify the county names to include sample sizes in the labels
ranks_long$County <- factor(ranks_long$County,levels = bottom_five_counties,
                            labels = paste(bottom_five_counties,
                                           "(n =", sample_sizes[bottom_five_counties], ")"))

# Plot the posterior distribution of ranks with annotated sample sizes
ggplot(ranks_long, aes(x = Rank)) +
  geom_histogram(binwidth = 1, fill = 'skyblue', color = 'black') +
  facet_wrap(~ County, ncol = 1, scales = "free_y") +
  labs(title = "Posterior Distribution of Ranks for Top Five Counties",
    x = "Rank", y = "Frequency") +
  theme_minimal()
```



Counties with very small sample sizes, exhibit broader rank distributions, indicating more uncertainty in their relative ranks. This variability reflects the model's shrinkage effect, where limited data leads to a higher influence of the prior, resulting in more dispersed rankings. In contrast, counties with slightly larger sample sizes, show a more concentrated rank distribution towards the top, suggesting greater confidence in their relative ranking. This trend highlights how sample size influences rank stability in Bayesian hierarchical models, with smaller sample sizes leading to broader, less certain rank distributions.