

610 HW 9

Hathaway Liu

2024-11-13

1. Derive a Gibbs sampler for the one sample hierarchical normal model,

$$\begin{aligned} y_{i,j} &= \theta_j + \epsilon_{i,j}, i = 1, \dots, n_j, j = 1, \dots, m \\ \theta_1, \dots, \theta_m &\sim \text{i.i.d. } N(\mu, \tau^2) \\ \{\epsilon_{i,j}\} &\sim \text{i.i.d. } N(0, \sigma^2). \end{aligned}$$

Specifically, for the prior distributions $\mu \sim N(\mu_0, v_0)$, $1/\sigma^2 \sim \text{gamma}(v_0/2, nu_0\sigma_0^2/2)$, $1/\tau^2 \sim \text{gamma}(\eta_0/2, \eta_0\tau_0^2/2)$, do the following:

- a. Compute the full conditional distributions of μ, τ^2, σ^2 and each θ_j .

- 1) Full conditional distribution of μ From the question, we can see that $\mu \sim N(\mu_0, v_0)$ and $\theta_j \mid \mu, \tau^2 \sim N(\mu, \tau^2)$. Then we could use Bayes' Theorem to find out our posterior distribution:

$$\begin{aligned} p(\mu \mid \theta, \tau^2) &\propto \exp\left(-\frac{\sum_{j=1}^m (\theta_j - \mu)^2}{2\tau^2}\right) \cdot \exp\left(-\frac{(\mu - \mu_0)^2}{2v_0}\right) \\ &\propto \exp\left(-\frac{1}{2} \left[\sum_{j=1}^m \frac{(\theta_j - \mu)^2}{\tau^2} + \frac{(\mu - \mu_0)^2}{v_0} \right]\right) \\ &\propto \exp\left(-\frac{1}{2} \left[\frac{1}{\tau^2} \sum_{j=1}^m (\theta_j^2 - 2\theta_j\mu + \mu^2) + \frac{1}{v_0} (\mu^2 - 2\mu_0\mu + \mu_0^2) \right]\right) \\ &\propto \exp\left(-\frac{1}{2} \left[\left(\frac{m}{\tau^2} + \frac{1}{v_0}\right) \mu^2 - 2 \left(\frac{\sum_{j=1}^m \theta_j}{\tau^2} + \frac{\mu_0}{v_0}\right) \mu + (\text{terms not involving } \mu) \right]\right) \end{aligned}$$

Then we could complete the square by assigning $A = \frac{m}{\tau^2} + \frac{1}{v_0}$, $B = -2 \left(\frac{\sum_{j=1}^m \theta_j}{\tau^2} + \frac{\mu_0}{v_0} \right)$, and doing the following operation:

$$A\mu^2 + B\mu = A \left(\mu^2 + \frac{B}{A}\mu \right) = A \left(\left(\mu + \frac{B}{2A} \right)^2 - \left(\frac{B}{2A} \right)^2 \right)$$

Thus in this case, the mean will be $\frac{B}{A}$, variance will be $\frac{1}{A}$. Thus the full conditional distribution for μ will be:

$$\mu \mid \theta, \tau^2 \sim N \left(\frac{\sum_{j=1}^m \theta_j / \tau^2 + \mu_0 / v_0}{m / \tau^2 + 1 / v_0}, \frac{1}{m / \tau^2 + 1 / v_0} \right)$$

- 2) Full Conditional Distribution of τ^2 Since we have the prior $1/\tau^2 \sim \text{Gamma}\left(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2}\right)$, and the likelihood for θ_j is $\theta_j \mid \mu, \tau^2 \sim N(\mu, \tau^2)$. Then we can derive the the posterior:

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

Thus the full conditional distribution of τ^2 should follow a inverse-Gamma distribution:

$$\begin{aligned} 1/\tau^2 \mid \theta, \mu &\sim \text{Gamma}\left(\frac{m+\eta_0}{2}, \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0\tau_0^2}{2}\right) \\ \tau^2 \mid \theta, \mu &\sim \text{invGamma}\left(\frac{m+\eta_0}{2}, \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0\tau_0^2}{2}\right) \end{aligned}$$

- 3) Full conditional distribution of σ^2 : Since we have the prior for $1/\sigma^2$: $1/\sigma^2 \sim \text{Gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right)$, and the likelihood is $y_{i,j} \mid \theta_j, \sigma^2 \sim N(\theta_j, \sigma^2)$. Then we can derive the the posterior:

$$\begin{aligned} p(\sigma^2 \mid y, \theta) &\propto (\sigma^2)^{-\sum_j n_j/2} \exp\left(-\frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2}{2\sigma^2}\right) \cdot (1/\sigma^2)^{\nu_0/2-1} \exp\left(-\frac{\nu_0\sigma_0^2}{2\sigma^2}\right) \\ &\propto (\sigma^2)^{-(\sum_j n_j + \nu_0)/2} \exp\left(-\frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \nu_0\sigma_0^2}{2\sigma^2}\right) \end{aligned}$$

Thus the full conditional distribution of σ^2 should follow a inverse-Gamma distribution:

$$\begin{aligned} 1/\sigma^2 \mid y, \theta &\sim \text{Gamma}\left(\frac{\sum_{j=1}^m n_j + \nu_0}{2}, \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \nu_0\sigma_0^2}{2}\right) \\ \sigma^2 \mid y, \theta &\sim \text{invGamma}\left(\frac{\sum_{j=1}^m n_j + \nu_0}{2}, \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \nu_0\sigma_0^2}{2}\right) \end{aligned}$$

- 4) Full Conditional Distribution of Each θ_j Since the prior for θ_j is: $\theta_j \sim N(\mu, \tau^2)$, and the likelihood contribution from $y_{i,j}$: $y_{i,j} \mid \theta_j, \sigma^2 \sim N(\theta_j, \sigma^2)$, we can compute the posterior distribution using Bayes' Theorem:

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

We can apply the similar strategy as computing full conditional for μ , where we complete the square by assigning $A = \frac{n_j}{\sigma^2} + \frac{1}{\tau^2}$ and $B = -2 \left(\frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2} + \frac{\mu}{\tau^2} \right)$. Then with the same operation, we will get the full conditional distribution:

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

- b. For each full conditional, compute the full conditional expectation, and show how the expectation is a combination of data information and prior information.

- 1) Since we get the full conditional for μ is $\mu \mid \theta, \tau^2 \sim N\left(\frac{\sum_{j=1}^m \theta_j / \tau^2 + \mu_0 / v_0}{m / \tau^2 + 1 / v_0}, \frac{1}{m / \tau^2 + 1 / v_0}\right)$, its expectation is:

$$E[\mu \mid \theta, \tau^2] = \frac{\sum_{j=1}^m \theta_j / \tau^2 + \mu_0 / v_0}{m / \tau^2 + 1 / v_0}$$

$\sum_{j=1}^m \theta_j / \tau^2$ represents the contribution of the data. It is the sum of the observed θ_j 's, weighted by the precision ($1/\tau^2$). And μ_0 / v_0 represents the prior mean weighted by the prior precision ($1/v_0$). The expectation is essentially a weighted average of the sample mean of θ_j and the prior mean μ_0 , where the weights are given by the precisions of the prior and data.

- 2) Since we get the full conditional for τ^2 is $\tau^2 \mid \theta, \mu \sim \text{invGamma}\left(\frac{m + \eta_0}{2}, \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0 \tau_0^2}{2}\right)$, its expectation is:

$$E[\tau^2 \mid \theta, \mu] = \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0 \tau_0^2}{m + \eta_0 - 2}$$

$\sum_{j=1}^m (\theta_j - \mu)^2$ is the sum of squared deviations of the observed θ_j 's from the current estimate μ . This reflects the variability in the data. $\eta_0 \tau_0^2$ represents the contribution of the prior belief about the variability. The expectation reflects a balance between the empirical variability of the data (θ_j) and the prior variance (τ_0^2), with the weight given by the degrees of freedom ($m + \eta_0 - 2$).

- 3) Since we get the full conditional for σ^2 is $\sigma^2 \mid y, \theta \sim \text{invGamma}\left(\frac{\sum_{j=1}^m n_j + \nu_0}{2}, \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \nu_0 \sigma_0^2}{2}\right)$, its expectation is:

$$E[\sigma^2 \mid y, \theta] = \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \nu_0 \sigma_0^2}{\sum_{j=1}^m n_j + \nu_0 - 2}$$

$\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2$ represents the total sum of squared residuals, capturing the variability in the data. $\nu_0 \sigma_0^2$ represents the contribution of the prior belief about the variability.

- 4) Since we get the full conditional for θ_j is $\theta_j \mid y_{i,j}, \mu, \tau^2, \sigma^2 \sim N\left(\frac{\sum_{i=1}^{n_j} y_{i,j} / \sigma^2 + \mu / \tau^2}{n_j / \sigma^2 + 1 / \tau^2}, \frac{1}{n_j / \sigma^2 + 1 / \tau^2}\right)$, its expectation is:

$$E[\theta_j \mid y_{i,j}, \mu, \tau^2, \sigma^2] = \frac{\sum_{i=1}^{n_j} y_{i,j} / \sigma^2 + \mu / \tau^2}{n_j / \sigma^2 + 1 / \tau^2}$$

$\sum_{i=1}^{n_j} y_{i,j} / \sigma^2$ is the sum of the observed data points $y_{i,j}$, each weighted by the precision ($1/\sigma^2$). This term reflects the influence of the observed data. μ / τ^2 represents the prior mean of θ_j , weighted by the prior precision ($1/\tau^2$). The denominator, $n_j / \sigma^2 + 1 / \tau^2$, is the combined precision (total weight from both the data and the prior). The expectation is a weighted average of the observed data (mean of $y_{i,j}$) and the prior mean (μ), with the weights determined by the precisions of the data and prior.

- c. Describe with pseudocode a Gibbs sampling algorithm to approximate the joint posterior distribution $p(\theta_1, \dots, \theta_m, \mu, \tau^2, \sigma^2 \mid \{y_{i,j}, i = 1, \dots, n_j, j = 1, \dots, m\})$.

- Initializing $(\theta^{(0)}, \mu^{(0)}, \sigma^{2(0)}, \tau^{2(0)})$
- For $t = 1, 2, \dots, T$

- 1) Update θ_j

- For $j = 1, 2, \dots, m$

- Compute $\tau_{\theta_j}^{(t)} = \left(\frac{n_j}{\sigma^{2(t)}} + \frac{1}{\tau^{2(t)}}\right)^{-1}$ and $\mu_{\theta_j}^{(t)} = \tau_{\theta_j}^{2(t)} \left(\frac{n_j \bar{y}_j}{\sigma^{2(t)}} + \frac{\mu^{(t)}}{\tau^{2(t)}}\right)$
- Draw $\theta_j^{(t)} \sim N\left(\mu_{\theta_j}^{(t)}, \tau_{\theta_j}^{2(t)}\right)$

2) Update μ :

- Compute $\mu_M = \frac{\sum_{j=1}^m \theta_j / \tau^2 + \mu_0 / v_0}{m / \tau^2 + 1 / v_0}$, $\nu_M = \frac{1}{m / \tau^{2(t-1)} + 1 / v_0}$
- Draw $\mu^{(t)} \sim (\mu_M, \nu_M)$

3) Update τ^2 :

- Compute $\alpha_\tau = \frac{\eta_0 + m}{2}$ and $\beta_\tau = \frac{\eta_0 \tau_0^2 + \sum_{j=1}^m (\theta_j^{(t-1)} - \mu^{(t)})^2}{2}$
- Draw $\tau^{2(t)} \sim \text{InvGamma}\left(\alpha_\tau, \beta_\tau^{(t)}\right)$

4) Update σ^2 :

- Compute $\alpha_\sigma = \frac{\nu_0 + \sum_{j=1}^m n_j}{2}$ and $\beta_\sigma = \frac{\nu_0 \sigma_0^2 + \sum_j \sum_i (y_{ij} - \theta_j^{(t-1)})^2}{2}$
- Draw $\sigma^{2(t)} \sim \text{InvGamma}\left(\alpha_\sigma, \beta_\sigma^{(t)}\right)$

2. Do a Bayesian analysis of the radon data in the homework directory, using the hierarchical normal model on the log radon levels. For the hyperparameters, you may either choose “diffuse” values corresponding to spread-out prior distributions, or you can weakly center the priors around MLEs obtained from the data (for example, you can set $\mu_0 = \bar{y}$, but make νu_0 very large).

a. Run a Gibbs sampler for at least 10,000 iterations. You may thin the Gibbs sampler if you want. Make trace plots of μ , $\log \tau^2$ and $\log \sigma^2$ to assess the mixing of the Gibbs sampler.

```
radon_data <- readRDS("radonMN.rds")
```

```
library(coda)
library(ggplot2)
library(gridExtra)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
## combine

## The following objects are masked from 'package:stats':
##
## filter, lag

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```

radon_data$log_radon <- log(radon_data$radon)

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

n_j <- table(radon_data$county)[counties]
y_j <- tapply(radon_data$log_radon, radon_data$county, c)

# Initialization
mu_0 <- mean(radon_data$log_radon)
v_0 <- 1e7
eta_0 <- 0.001
tau_sq_0 <- 1
nu_0 <- 0.001
sigma_sq_0 <- 1

# Number of iterations
T <- 20000

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

# Initialize parameters
mu <- mu_0
tau_sq <- tau_sq_0
sigma_sq <- sigma_sq_0
theta_j <- mean(y_j)

## Warning in mean.default(y_j): argument is not numeric or logical: returning NA

# Set seed for reproducibility
set.seed(123)

# 3. Gibbs Sampler Implementation
for (t in 1:T) {
  # Update theta_j for each county
  for (j in 1:m) {
    tau_theta_j <- 1 / (1 / tau_sq + n_j[j] / sigma_sq)
    mu_theta_j <- tau_theta_j * (mu / tau_sq + sum(y_j[[j]]) / sigma_sq)
    theta_j[j] <- rnorm(1, mu_theta_j, sqrt(tau_theta_j))
  }

  # Update mu
  mu_M <- (sum(theta_j) / tau_sq + mu_0 / v_0) / (m / tau_sq + 1 / v_0)
  nu_M <- 1 / (m / tau_sq + 1 / v_0)
  mu <- rnorm(1, mu_M, sqrt(nu_M))

  # Update tau_sq
  alpha_tau <- (eta_0 + m) / 2
  beta_tau <- (eta_0 * tau_sq_0 + sum((theta_j - mu)^2)) / 2

```

```

inv_tau_sq <- rgamma(1, alpha_tau, beta_tau)
tau_sq <- 1 / inv_tau_sq

# Update sigma_sq
alpha_sigma <- (nu_0 + length(radon_data$log_radon)) / 2
sum_sq_res <- sum(sapply(1:m, function(j) {
  sum((y_j[[j]] - theta_j[j])^2)
}))
beta_sigma <- (nu_0 * sigma_sq_0 + sum_sq_res) / 2
inv_sigma_sq <- rgamma(1, shape = alpha_sigma, rate = beta_sigma)
sigma_sq <- 1 / inv_sigma_sq

# Store samples
mu_samples[t] <- mu
tau_sq_samples[t] <- tau_sq
sigma_sq_samples[t] <- sigma_sq
theta_samples[t, ] <- theta_j
}

# Trace Plots
mcmc_samples <- mcmc(data = cbind(
  mu = mu_samples,
  log_tau2 = log(tau_sq_samples),
  log_sigma2 = log(sigma_sq_samples)
), start = 1, end = length(mu_samples))

samples_df <- as.data.frame(mcmc_samples)
samples_df$Iteration <- 1:nrow(samples_df)

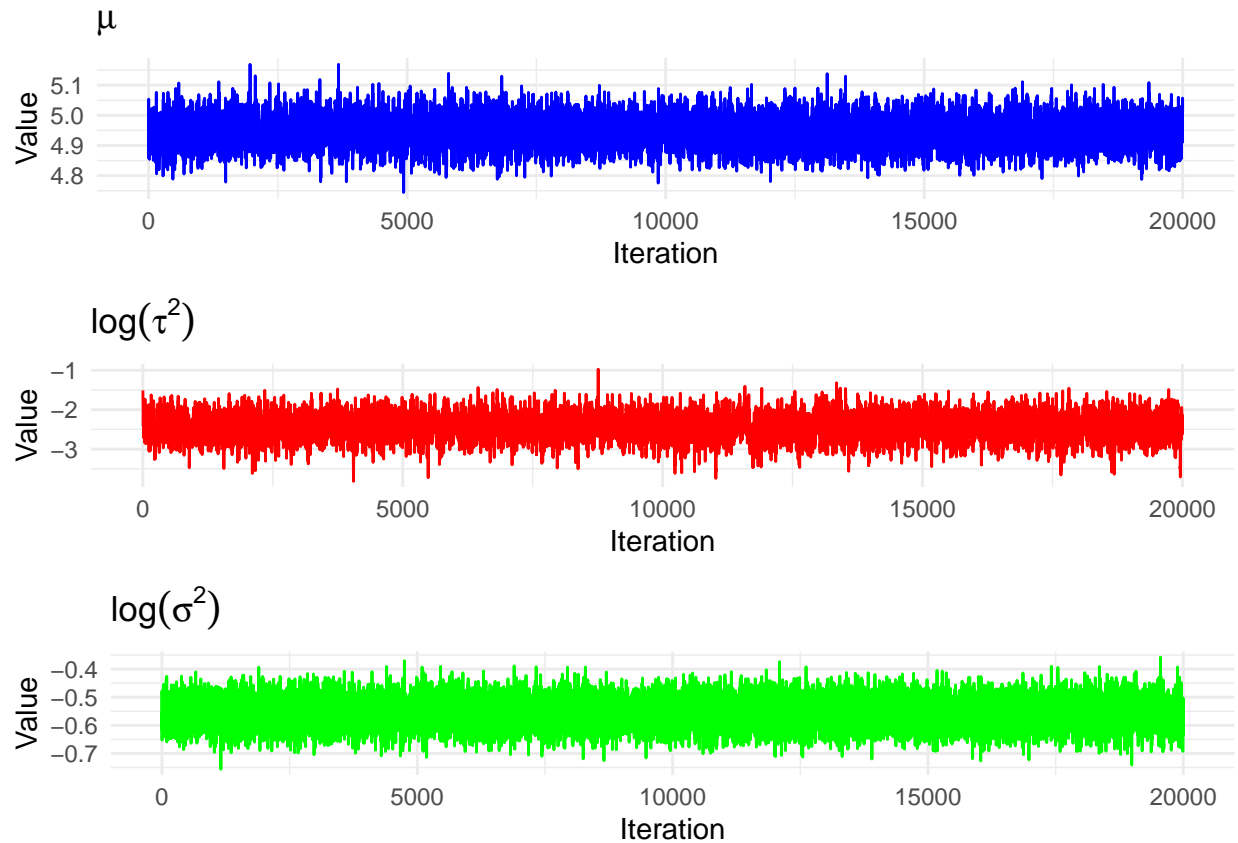
p1 <- ggplot(samples_df, aes(x = Iteration, y = mu)) +
  geom_line(color = "blue") +
  ggtitle(expression(mu)) +
  ylab("Value") +
  theme_minimal()

p2 <- ggplot(samples_df, aes(x = Iteration, y = log_tau2)) +
  geom_line(color = "red") +
  ggtitle(expression(log(tau^2))) +
  ylab("Value") +
  theme_minimal()

p3 <- ggplot(samples_df, aes(x = Iteration, y = log_sigma2)) +
  geom_line(color = "green") +
  ggtitle(expression(log(sigma^2))) +
  ylab("Value") +
  theme_minimal()

grid.arrange(p1, p2, p3, ncol = 1)

```



- b. Obtain 95% posterior confidence intervals for μ , τ^2 and σ^2 . Compare the confidence interval for μ to the confidence interval obtained from lmer.

```
# Compute 95% posterior confidence intervals for mu, tau^2, and sigma^2
mu_ci <- quantile(mu_samples, probs = c(0.025, 0.975))
tau_sq_ci <- quantile(tau_sq_samples, probs = c(0.025, 0.975))
sigma_sq_ci <- quantile(sigma_sq_samples, probs = c(0.025, 0.975))

cat("\n95% Posterior Credible Intervals:\n")
```

```
##
## 95% Posterior Credible Intervals:
```

```
cat(sprintf("mu: [%.4f, %.4f]\n", mu_ci[1], mu_ci[2]))
```

```
## mu: [4.8545, 5.0405]
```

```
cat(sprintf("tau^2: [%.4f, %.4f]\n", tau_sq_ci[1], tau_sq_ci[2]))
```

```
## tau^2: [0.0463, 0.1573]
```

```
cat(sprintf("sigma^2: [%.4f, %.4f]\n", sigma_sq_ci[1], sigma_sq_ci[2]))
```

```
## sigma^2: [0.5212, 0.6310]
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
lmer_model <- lmer(log_radon ~ 1 + (1 | county), data = radon_data)
```

```
lmer_ci <- confint(lmer_model, parm = "(Intercept)", level = 0.95)
```

```
## Computing profile confidence intervals ...
```

```
cat("95% Confidence Interval for mu from lmer:", lmer_ci, "\n")
```

```
## 95% Confidence Interval for mu from lmer: 4.854804 5.040563
```

The estimated CI and LMER confidence intervals are very similar, due to the fact that we set a weaker prior so the CI for posterior distribution is heavily dependent on data.

- c. Obtain posterior means of the county-level θ_j 's, and plot against the sample means, the \bar{y}_j 's. Explain the relationship that you see.

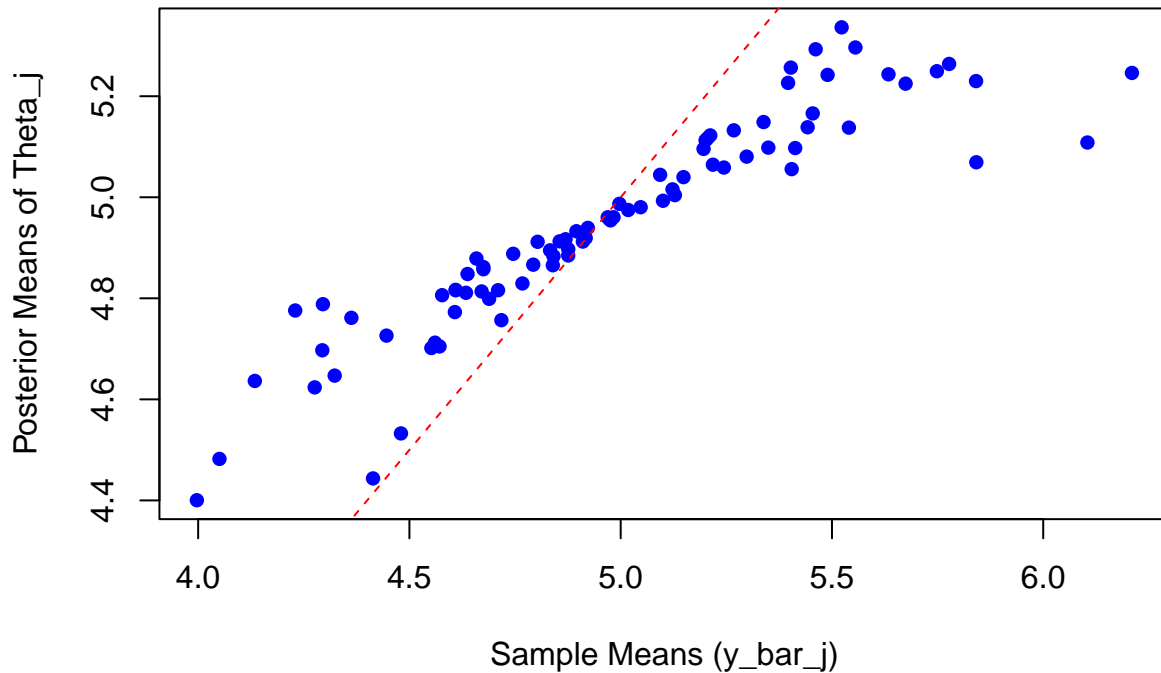
```
posterior_theta_means <- colMeans(theta_samples)
```

```
# Calculate sample means of y_j (observed data means for each county)
```

```
sample_means <- sapply(y_j, mean)
```

```
plot(sample_means, posterior_theta_means,  
      xlab = "Sample Means (y_bar_j)",  
      ylab = "Posterior Means of Theta_j",  
      main = "Posterior Means of Theta_j vs Sample Means",  
      pch = 16, col = "blue")  
abline(0, 1, col = "red", lty = 2)
```


Posterior Means of Theta_j vs Sample Means



We can see that from the plot, we can see a shrinkage effect, where the posterior mean of county-level θ_j shrink toward the sample mean, since we use sample mean as prior mean of μ .

- d. Identify the top five counties in terms of the sample means. For each county j and each iteration of the Gibbs sampler, compute the rank r_j of θ_j among all of the θ -values (so if θ_j is the largest value then $r_j = 1$). For each of these five counties, make a plot of the posterior distribution of its rank. Comment on the relationship between this distribution, the sample mean for county j , and the sample size for county j .

```
sample_means <- sapply(y_j, mean)

top_county_indices <- order(sample_means, decreasing = TRUE)[1:5]

top_counties <- counties[top_county_indices]
top_sample_means <- sample_means[top_county_indices]
top_sample_sizes <- n_j[top_county_indices]
cat("Top 5 Counties by Sample Means:\n")
```

Top 5 Counties by Sample Means:

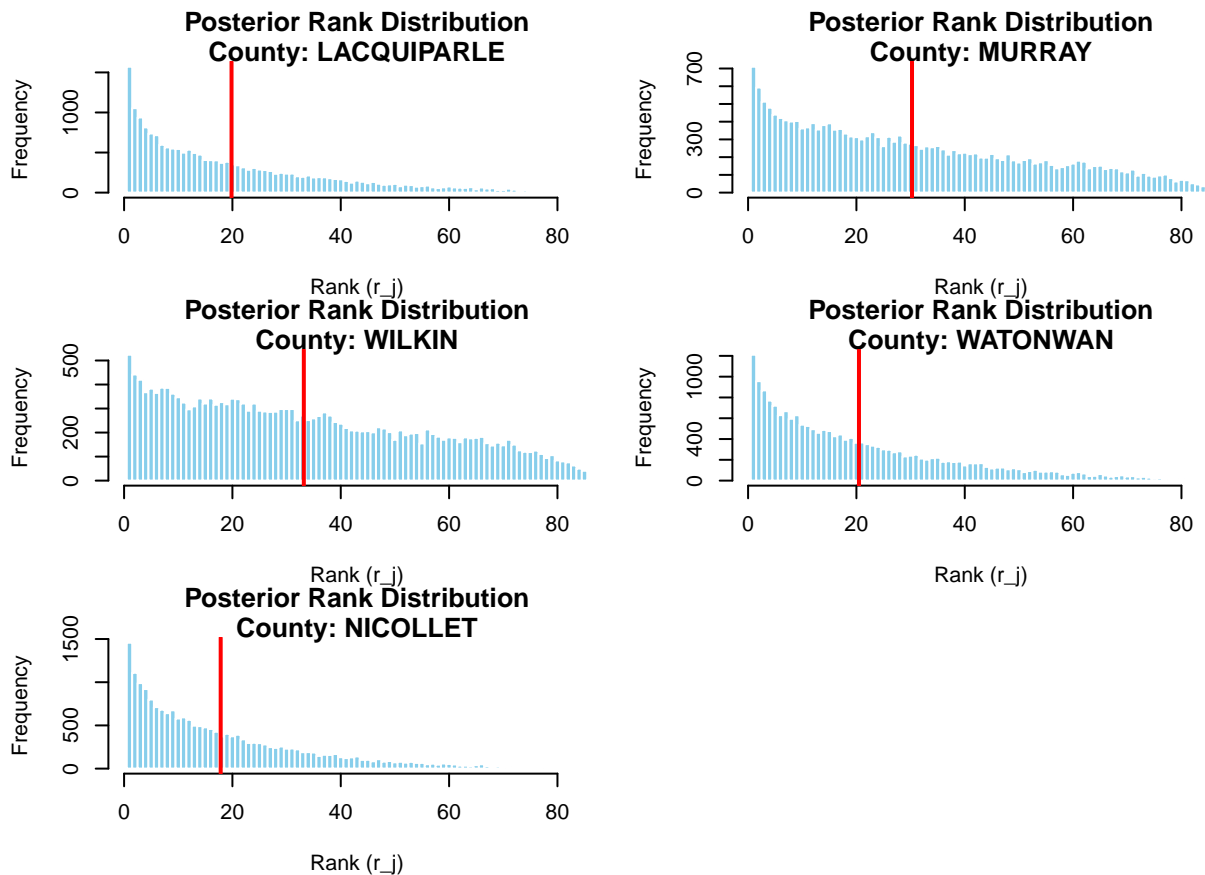
```
for (i in 1:5) {
  cat(paste0("County: ", top_counties[i],
            ", Sample Mean: ", round(top_sample_means[i], 3),
            ", Sample Size: ", top_sample_sizes[i], "\n"))
}
```

```
## County: LACQUIPARLE, Sample Mean: 6.21, Sample Size: 2
## County: MURRAY, Sample Mean: 6.105, Sample Size: 1
## County: WILKIN, Sample Mean: 5.842, Sample Size: 1
## County: WATONWAN, Sample Mean: 5.841, Sample Size: 3
## County: NICOLLET, Sample Mean: 5.777, Sample Size: 4
```

```
theta_ranks <- matrix(0, nrow = T, ncol = m)

for (t in 1:T) {
  theta_ranks[t, ] <- rank(-theta_samples[t, ])
}
top_county_ranks <- theta_ranks[, top_county_indices]

par(mfrow = c(3, 2), mar = c(4, 4, 2, 1))
for (i in 1:5) {
  hist(top_county_ranks[, i],
       breaks = seq(0.5, m + 0.5, by = 1),
       main = paste("Posterior Rank Distribution\nCounty:", top_counties[i]),
       xlab = "Rank (r_j)",
       ylab = "Frequency",
       col = "skyblue",
       border = "white")
  abline(v = mean(top_county_ranks[, i]), col = "red", lwd = 2)
}
```



```
bottom_county_indices <- order(sample_means, decreasing = FALSE)[1:5]
```

```
bottom_counties <- counties[bottom_county_indices]  
bottom_sample_means <- sample_means[bottom_county_indices]  
bottom_sample_sizes <- n_j[bottom_county_indices]
```

```
cat("Bottom 5 Counties by Sample Means:\n")
```

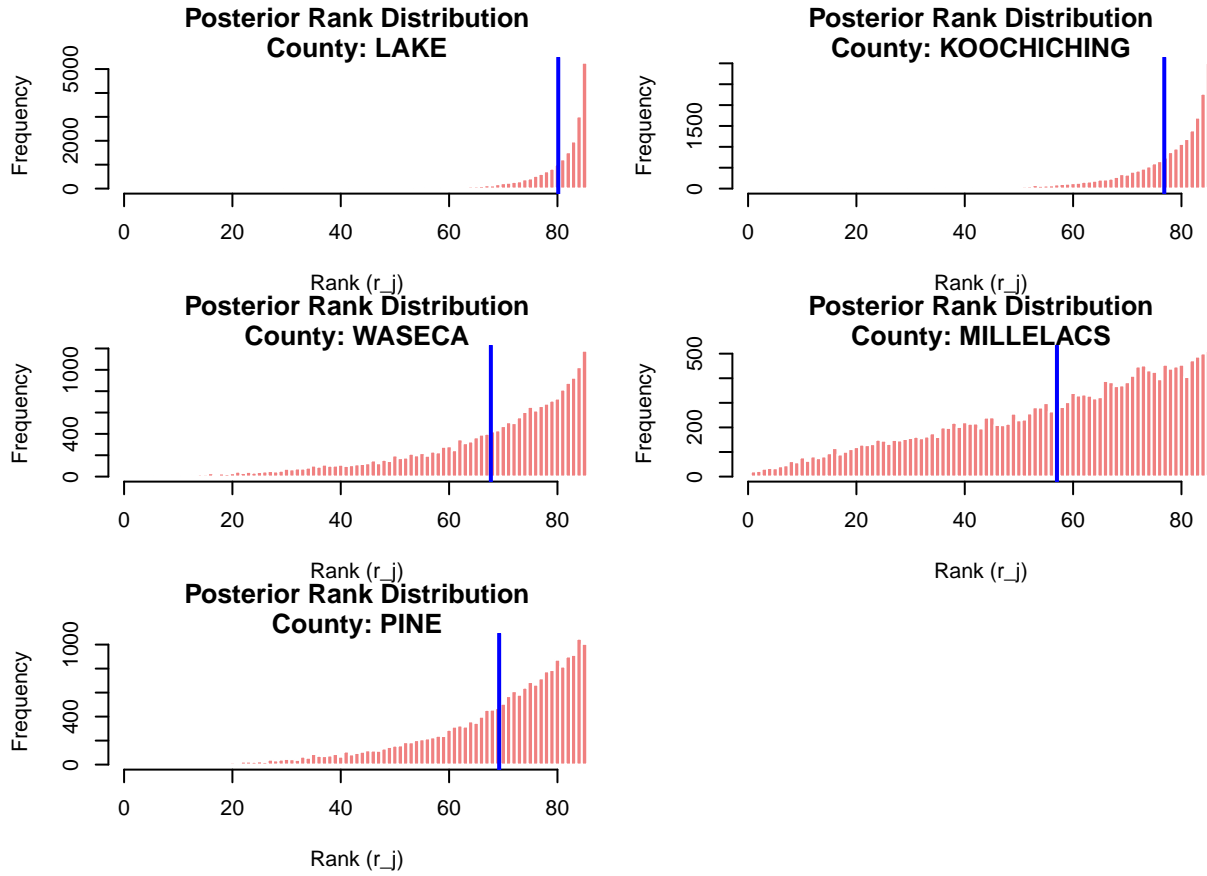
```
## Bottom 5 Counties by Sample Means:
```

```
for (i in 1:5) {  
  cat(paste0("County: ", bottom_counties[i],  
            ", Sample Mean: ", round(bottom_sample_means[i], 3),  
            ", Sample Size: ", bottom_sample_sizes[i], "\n"))  
}
```

```
## County: LAKE, Sample Mean: 3.997, Sample Size: 9  
## County: KOOCHICHING, Sample Mean: 4.05, Sample Size: 7  
## County: WASECA, Sample Mean: 4.134, Sample Size: 4  
## County: MILLELACS, Sample Mean: 4.23, Sample Size: 2  
## County: PINE, Sample Mean: 4.276, Sample Size: 6
```

```
bottom_county_ranks <- theta_ranks[, bottom_county_indices]
```

```
par(mfrow = c(3, 2), mar = c(4, 4, 2, 1))  
for (i in 1:5) {  
  hist(bottom_county_ranks[, i],  
        breaks = seq(0.5, m + 0.5, by = 1),  
        main = paste("Posterior Rank Distribution\nCounty:", bottom_counties[i]),  
        xlab = "Rank (r_j)",  
        ylab = "Frequency",  
        col = "lightcoral",  
        border = "white")  
  abline(v = mean(bottom_county_ranks[, i]), col = "blue", lwd = 2)  
}
```



To explore the relationship, we can draw both top 5 counties and bottom 5 counties. When we fix the sample size, larger sample mean means the distribution more peaked, while smaller sample mean means the distribution is more diffused; we can see this by comparing LACQUIPARLE and MURRAY. When we fix the sample mean, larger sample size means the distribution is more concentrated, while smaller sample size means less concentrated distribution; we can see this by comparing LAKE and MILLELACS since LAKE has sample size 2 and MILLELACS has sample size 9.