HW9

Holly Cui

2024-11-18

Question 1

The objective of the question is to derive a Gibbs sampler for the one sample HNM:

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

where the priors are defined as:

$$
\begin{aligned}
\mu &\sim N(\mu_0, v_0) \\
\sigma^2 &\sim \text{Inverse-Gamma}(\frac{\gamma_0}{2}, \frac{\gamma_0\sigma_0^2}{2}) \\
\tau^2 &\sim \text{Inverse-Gamma}(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2})
\end{aligned}
$$

(a)

Full Conditional for μ:

$$
\begin{aligned}
p(\mu|\text{all other params}) &\propto \{\prod_{j=1}^{m} p(\theta_j|\mu, \tau^2)\} \cdot p(\mu) \\
&= \{\prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\tau^2}} \exp(-\frac{(\theta_j-\mu)^2}{2\tau^2})\} \cdot \frac{1}{\sqrt{2\pi v_0}} \exp(-\frac{(\mu-\mu_0)^2}{2v_0}) \\
&\propto \exp(-\frac{\sum_{j=1}^{m}(\theta_j-\mu)^2}{2\tau^2}) \cdot \exp(-\frac{(\mu-\mu_0)^2}{2v_0}) \\
&= \exp\left(-\frac{1}{2\tau^2}(\sum_{j=1}^{m}\theta_j^2 - 2\mu\sum_{j=1}^{m}\theta_j + m\mu^2) - \frac{1}{2v_0}(\mu^2 - 2\mu\mu_0 + \mu_0^2)\right) \\
&= \exp\left(-\frac{1}{2}(\frac{m}{\tau^2} + \frac{1}{v_0})\mu^2 + (\frac{\sum_{j=1}^{m}\theta_j}{\tau^2} + \frac{\mu_0}{v_0})\mu - \frac{1}{2}(\frac{\sum_{j=1}^{m}\theta_j^2}{\tau^2} + \frac{\mu_0^2}{v_0})\right)
\end{aligned}
$$

To find the distribution of the posterior of μ, we focus on completing the square in the exponent:

$$\begin{aligned}
&-\frac{1}{2}(\frac{m}{\tau^2}+\frac{1}{v_0})\mu^2+(\frac{\sum_{j=1}^m \theta_j}{\tau^2}+\frac{\mu_0}{v_0})\mu-\frac{1}{2}(\frac{\sum_{j=1}^m \theta_j^2}{\tau^2}+\frac{\mu_0^2}{v_0})\\
&=-\frac{1}{2}(\frac{m}{\tau^2}+\frac{1}{v_0})(\mu^2-2\cdot\frac{\frac{\sum_{j=1}^m \theta_j}{\tau^2}+\frac{\mu_0}{v_0}}{\frac{m}{\tau^2}+\frac{1}{v_0}}\mu+\text{constant})\\
&=-\frac{1}{2}(\frac{m}{\tau^2}+\frac{1}{v_0})(\mu-\frac{\frac{\sum_{j=1}^m \theta_j}{\tau^2}+\frac{\mu_0}{v_0}}{\frac{m}{\tau^2}+\frac{1}{v_0}})^2+\text{constant}
\end{aligned}$$

Thus, we get:

$$\begin{aligned}
p(\mu|\theta_1,...,\theta_m,\tau^2) &\propto \exp\left(-\frac{1}{2}(\frac{m}{\tau^2}+\frac{1}{v_0})(\mu-\frac{\frac{\sum_{j=1}^m \theta_j}{\tau^2}+\frac{\mu_0}{v_0}}{\frac{m}{\tau^2}+\frac{1}{v_0}})^2\right)\\
&=\exp\left(-\frac{1}{2\frac{1}{\frac{m}{\tau^2}+\frac{1}{v_0}}}(\mu-\frac{\frac{\sum_{j=1}^m \theta_j}{\tau^2}+\frac{\mu_0}{v_0}}{\frac{m}{\tau^2}+\frac{1}{v_0}})^2\right)\\
&=\exp\left(-\frac{1}{2\frac{\tau^2 v_0}{mv_0+\tau^2}}(\mu-\frac{v_0\sum_{j=1}^m \theta_j+\mu_0\tau^2}{mv_0+\tau^2})^2\right)
\end{aligned}$$

The above follows the kernel of a normal distribution, with mean $\frac{v_0\sum_{j=1}^m \theta_j+\mu_0\tau^2}{mv_0+\tau^2}$ and variance $\frac{\tau^2 v_0}{mv_0+\tau^2}$. Therefore, we derived the full condition for μ as:

$$\boxed{\mu|\theta_1,...,\theta_m,\tau^2\sim N(\frac{v_0\sum_{j=1}^m \theta_j+\mu_0\tau^2}{mv_0+\tau^2},\frac{\tau^2 v_0}{mv_0+\tau^2})}$$

Full Conditional for τ^2:

$$\begin{aligned}
p(\tau^2|\text{all other params}) &\propto \{\prod_{j=1}^m p(\theta_j|\mu,\tau^2)\}\cdot p(\tau^2)\\
&=\frac{1}{(2\pi\tau^2)^{m/2}}\exp\left(-\frac{1}{2\tau^2}\sum_{j=1}^m(\theta_j-\mu)^2\right)\times\frac{1}{\Gamma(\eta_0/2)}\left(\frac{\eta_0\tau_0^2}{2}\right)^{\eta_0/2}(\tau^2)^{-\eta_0/2-1}\exp\left(-\frac{\eta_0\tau_0^2}{2\tau^2}\right)\\
&\propto(\tau^2)^{-(m/2+\eta_0/2+1)}\exp\left(-\frac{1}{2\tau^2}(\sum_{j=1}^m(\theta_j-\mu)^2+\eta_0\tau_0^2)\right)
\end{aligned}$$

The above follows the kernel of an inverse gamma distribution, with $\alpha=\frac{m+\eta_0}{2}$ and $\beta=\frac{\sum_{j=1}^m(\theta_j-\mu)^2+\eta_0\tau_0^2}{2}$. Therefore, we derived the full condition for τ^2 as:

$$\boxed{\tau^2|\theta_1,...,\theta_m,\mu\sim\text{Inverse-Gamma}(\frac{m+\eta_0}{2},\frac{\sum_{j=1}^m(\theta_j-\mu)^2+\eta_0\tau_0^2}{2})}$$

Full Conditional for σ^2:

$$
\begin{aligned}
p(\sigma^2|\text{all other params}) &\propto \{\prod_{j=1}^{m}\prod_{i=1}^{n_j} p(y_{i,j}|\theta_j,\sigma^2)\}\cdot p(\sigma^2) \\
&= \frac{1}{(2\pi\sigma^2)^{\sum_{j=1}^{m} n_j/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2\right)\times \\
&\quad \frac{1}{\Gamma(\gamma_0/2)}\left(\frac{\gamma_0\sigma_0^2}{2}\right)^{\gamma_0/2}(\sigma^2)^{-\gamma_0/2-1}\exp\left(-\frac{\gamma_0\sigma_0^2}{2\sigma^2}\right) \\
&\propto (\sigma^2)^{-(\sum_{j=1}^{m} n_j/2+\gamma_0/2+1)}\exp\left(-\frac{1}{2\sigma^2}(\sum_{j=1}^{m}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2+\gamma_0\sigma_0^2)\right)
\end{aligned}
$$

The above follows the kernel of an inverse gamma distribution, with $\alpha = \frac{\sum_{j=1}^{m} n_j+\gamma_0}{2}$ and $\beta = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2+\gamma_0\sigma_0^2}{2}$. Therefore, we derived the full condition for τ^2 as:

$$
\boxed{\sigma^2|\{\theta_j\},\{y_{i,j}\} \sim \text{Inverse-Gamma}(\frac{\sum_{j=1}^{m} n_j+\gamma_0}{2}, \frac{\sum_{j=1}^{m}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2+\gamma_0\sigma_0^2}{2})}
$$

Full Conditional for θ_j:

$$
\begin{aligned}
p(\theta_j|\text{all other params}) &\propto \{\prod_{i=1}^{n_j} p(y_{i,j}|\theta_j,\sigma^2)\}\cdot p(\theta_j|\mu,\tau^2) \\
&= \frac{1}{(2\pi\sigma^2)^{n_j/2}}\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2\right)\times\frac{1}{(2\pi\tau^2)^{1/2}}\exp\left(-\frac{1}{2\tau^2}(\theta_j-\mu)^2\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n_j}(y_{i,j}-\theta_j)^2\right)\times\exp\left(-\frac{1}{2\tau^2}(\theta_j-\mu)^2\right) \\
&= \exp\left(-\frac{1}{2}(\frac{n_j}{\sigma^2}+\frac{1}{\tau^2})\theta_j^2+(\frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}+\frac{\mu}{\tau^2})\theta_j+\text{constant}\right)
\end{aligned}
$$

To find the distribution of the posterior of θ_j, we focus on completing the square in the exponent:

$$
\begin{aligned}
&-\frac{1}{2}(\frac{n_j}{\sigma^2}+\frac{1}{\tau^2})\theta_j^2+(\frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}+\frac{\mu}{\tau^2})\theta_j+\text{constant} \\
&= -\frac{1}{2}(\frac{n_j}{\sigma^2}+\frac{1}{\tau^2})(\theta_j-\frac{\frac{\sum_{i=1}^{n_j} y_{i,j}}{\sigma^2}+\frac{\mu}{\tau^2}}{\frac{n_j}{\sigma^2}+\frac{1}{\tau^2}})^2+\text{constant} \\
&\propto -\frac{1}{2\frac{\sigma^2\tau^2}{n_j\tau^2+\sigma^2}}(\theta_j-\frac{\tau^2\sum_{i=1}^{n_j} y_{i,j}+\mu\sigma^2}{n_j\tau^2+\sigma^2})^2
\end{aligned}
$$

The above exponent follows the kernel of a normal distribution, with mean $\frac{\tau^2\sum_{i=1}^{n_j} y_{i,j}+\mu\sigma^2}{n_j\tau^2+\sigma^2}$ and variance $\frac{\sigma^2\tau^2}{n_j\tau^2+\sigma^2}$. Therefore, we derived the full condition for θ_j as:

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

(b)

Expectation for μ:

Previously, we've derived the full conditional of μ as:

$$\mu|\theta_1, ..., \theta_m, \tau^2 \sim N(\frac{v_0 \sum_{j=1}^m \theta_j + \mu_0 \tau^2}{m v_0 + \tau^2}, \frac{\tau^2 v_0}{m v_0 + \tau^2})$$

Thus, the expectation of the full conditional μ is:

$$\boxed{\mathbb{E}[\mu|\theta_1, ..., \theta_m, \tau^2] = \frac{v_0 \sum_{j=1}^m \theta_j + \mu_0 \tau^2}{m v_0 + \tau^2} = \frac{\frac{m\bar{\theta}}{\tau^2} + \frac{\mu_0}{v_0}}{\frac{m}{\tau^2} + \frac{1}{v_0}}}$$

Here:

- $\frac{m}{\tau^2}$ is the weight given to the data (denoted by $\bar{\theta}$)
- $\frac{1}{v_0}$ is the weight given to the prior (denoted by μ_0)

Expectation for τ^2:

Previously, we've derived the full conditional of τ^2 as:

$$\tau^2|\theta_1, ..., \theta_m, \mu \sim \text{Inverse-Gamma}(\frac{m + \eta_0}{2}, \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0 \tau_0^2}{2})$$

The expectation under an inverse-gamma distribution Inverse-Gamma(α, β) is $\frac{\beta}{\alpha - 1}$. Thus, we get:

$$\boxed{\mathbb{E}[\tau^2|\theta_1, ..., \theta_m, \mu] = \frac{\frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0 \tau_0^2}{2}}{\frac{m + \eta_0}{2} - 1} = \frac{\sum_{j=1}^m (\theta_j - \mu)^2 + \eta_0 \tau_0^2}{m + \eta_0 - 2}}$$

Here:

- $\sum_{j=1}^m (\theta_j - \mu)^2$ represents the variation in the group means θ_j around the overall mean μ
- $\eta_0 \tau_0^2$ represents the prior information on τ^2

Expectation for σ^2:

Previously, we've derived the full conditional of σ^2 as:

$$\sigma^2|\{\theta_j\}, \{y_{i,j}\} \sim \text{Inverse-Gamma}(\frac{\sum_{j=1}^m n_j + \gamma_0}{2}, \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \gamma_0 \sigma_0^2}{2})$$

Thus, the expectation of the full conditional σ^2 is:

$$\boxed{\mathbb{E}[\sigma^2|\{\theta_j\}, \{y_{i,j}\}] = \frac{\frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \gamma_0 \sigma_0^2}{2}}{\frac{\sum_{j=1}^m n_j + \gamma_0}{2} - 1} = \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2 + \gamma_0 \sigma_0^2}{\sum_{j=1}^m n_j + \gamma_0 - 2}}$$

Here:

- $\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{i,j} - \theta_j)^2$ represents the total variation in the data
- $\gamma_0 \sigma_0^2$ represents the prior information on σ^2

Expectation for θ_j:

Previously, we've derived the full conditional of θ_j as:

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

Thus, the expectation of the full conditional θ_j is:

$$\boxed{\mathbb{E}[\theta_j|\{y_{i,j}\},\mu,\sigma^2,\tau^2] = \frac{\tau^2\sum_{i=1}^{n_j} y_{i,j}+\mu\sigma^2}{n_j\tau^2+\sigma^2} = \frac{\frac{n_j\bar{y_j}}{\sigma^2}+\frac{\mu}{\tau^2}}{\frac{n_j}{\sigma^2}+\frac{1}{\tau^2}}}$$

Here:

- $\frac{n_j}{\sigma^2}$ is the weight given to the data (denoted by $\bar{y_j}$)
- $\frac{1}{\tau^2}$ is the weight given to the prior (denoted by μ)

In conclusion, we can see that each of the expectations of the full conditionals shows a combination of data information and prior information. For θ_j and μ, the expectations are in the form of weighted averages where the weights depend on the relative precision of the data versus the prior. For σ^2 and τ^2, the expectations bring together the data variation with the prior variation with an overall adjustment.

(c)

Given starting values $(\mu^{(0)}, \tau^{2\ (0)}, \sigma^{2\ (0)}, \theta_1^{(0)}, ..., \theta_m^{(0)})$, we define the Gibbs sampling process by iterating the following steps for $s = 0, ..., S$ times:

1. Sample $\mu^{(s+1)} \sim p(\mu|\theta_1^{(s)}, ..., \theta_m^{(s)}, \tau^{2\ (s)})$, which follows the derived normal distribution.
2. Sample $\tau^{2\ (s+1)} \sim p(\tau^2|\theta_1^{(s)}, ..., \theta_m^{(s)}, \mu^{(s+1)})$, which follows the derived inverse gamma distribution.
3. Sample $\sigma^{2\ (s+1)} \sim p(\sigma^2|\theta_1^{(s)}, ..., \theta_m^{(s)}, \{y_{i,j}\})$, which follows the derived inverse gamma distribution.
4. For each $j = 1, ..., m$, sample $\theta_j^{(s+1)} \sim p(\theta_j|\{y_{i,j}\}, \mu^{(s+1)}, \tau^{2\ (s+1)}, \sigma^{2\ (s+1)})$, which follows the derived normal distribution.

After each iteration, we store the current samples for $(\mu, \tau^2, \sigma^2, \theta_1, ..., \theta_m)$. The final set of samples should be approximately the joint posterior $p(\mu, \tau^2, \sigma^2, \theta_1, ..., \theta_m|\{y_{i,j}\})$.

Below is the pseudocode for adopting the Gibbs sampler:

```
gibbs_sampler <- function(
  y,                  # List of observed data for each group
  mu_0, v_0,          # Prior mean and variance for mu
  sigma_0_sq, gamma_0 # Prior scale and shape for sigma^2
  tau_0_sq, eta_0,    # Prior scale and shape for tau^2
  num_iter,           # Number of Gibbs iterations
  burn_in,            # Number of burn-in iterations
  init_values         # Initial values for mu, sigma^2, tau^2, theta
) {
  # Initialization
  m <- length(y)  # Number of groups
  n <- sapply(y, length)  # Number of observations per group
  N <- sum(n)  # Total number of observations

  # Extract initial values
  mu <- init_values$mu
  tau2 <- init_values$tau2
  sigma2 <- init_values$sigma2
  theta <- init_values$theta

  # Storage for samples
  theta_samples <- matrix(0, nrow = num_iter, ncol = m)
  mu_samples <- numeric(num_iter)
  tau2_samples <- numeric(num_iter)
  sigma2_samples <- numeric(num_iter)

  # Gibbs sampling iterations
  for (iter in 1:num_iter) {
    # Step 1: Sample mu
    theta_bar <- mean(theta)
    mu_var <- 1 / (m / tau2 + 1 / v_0)
    mu_mean <- mu_var * (sum(theta) / tau2 + mu_0 / v_0)
    mu <- rnorm(1, mu_mean, sqrt(mu_var))

    # Step 2: Sample tau^2
    ssq_theta <- sum((theta - mu)^2)
    shape_tau2 <- (m + eta_0) / 2
```

```r
    rate_tau2 <- (ssq_theta + eta_0 * tau_0_sq) / 2
    tau2 <- 1 / rgamma(1, shape_tau2, rate_tau2)

    # Step 3: Sample sigma^2
    ssq_y <- sum(sapply(1:m, function(j) sum((y[[j]] - theta[j])^2)))
    shape_sigma2 <- (N + gamma_0) / 2
    rate_sigma2 <- (ssq_y + gamma_0 * sigma_0_sq) / 2
    sigma2 <- 1 / rgamma(1, shape_sigma2, rate_sigma2)

    # Step 4: Sample each theta_j
    for (j in 1:m) {
      y_j <- y[[j]]
      n_j <- n[j]
      y_bar_j <- mean(y_j)
      theta_var <- 1 / (n_j / sigma2 + 1 / tau2)
      theta_mean <- theta_var * (sum(y_j) / sigma2 + mu / tau2)
      theta[j] <- rnorm(1, theta_mean, sqrt(theta_var))
    }

    # Store samples
    theta_samples[iter, ] <- theta
    mu_samples[iter] <- mu
    tau2_samples[iter] <- tau2
    sigma2_samples[iter] <- sigma2
  }

  # Discard burn-in period and return posterior samples
  list(
    theta_samples = theta_samples[(burn_in + 1):num_iter, ],
    mu_samples = mu_samples[(burn_in + 1):num_iter],
    tau2_samples = tau2_samples[(burn_in + 1):num_iter],
    sigma2_samples = sigma2_samples[(burn_in + 1):num_iter]
  )
}
```

Question 2

```r
radon_df <- readRDS("radonMN.rds")
```

- `county`: The name of the county where measurements were taken.
- `lon`: Longitude of the location.
- `lat`: Latitude of the location.
- `Uppm`: Uranium concentration in ppm (parts per million).
- `radon`: Radon levels in some concentration measurement.

The objective of the question is to build an HNM on the log radon levels and estimate with Bayesian strategy:

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

(a)

Before we get started, we retrieve the variables from the dataset:

```r
# Preprocess the data
log_radon <- log(radon_df$radon)
county <- radon_df$county
counties <- unique(county)
m <- length(counties)
group_indices <- match(county, counties)
```

To run a Gibbs sampler, we first decide on choosing the **weakly informative priors** (derived based on empirical summaries of the data) as our approach. Then, we calculate the summary statistics from the samples to obtain the initial values of the parameters:

```r
# Initialize priors based on weakly informative assumptions
y_bar <- tapply(log_radon, group_indices, mean)  # County means
global_mean <- mean(y_bar)                       # Prior mean for mu
# Empirical variances for priors
hat_tau2 <- sum((y_bar - global_mean)^2) / m     # Across-county
hat_sigma2 <- sum((log_radon - mean(log_radon))^2) / length(log_radon)  # Within-county
```

To iteratively update the parameters in Gibbs sampler, we assume the following:

$$
\begin{aligned}
\mu &\sim N(\mu_0, v_0) \\
\tau^2 &\sim \text{Inverse-Gamma}(a_\tau, b_\tau) \\
\sigma^2 &\sim \text{Inverse-Gamma}(a_\sigma, b_\sigma)
\end{aligned}
$$

Ensuring that we have weak priors, we define the following hyperparameters:

- For μ: $\mu_0 = \bar{y}, v_0 = 10^7$ (very large).
- For τ^2: $a_\tau = 2, b_\tau = \hat{\tau}^2(a_\tau - 1)$ (since $\mathbb{E}[\tau^2] = \frac{b_\tau}{a_\tau - 1} = \hat{\tau}^2$).
- For μ: $a_\sigma = 2, b_\sigma = \hat{\sigma}^2(a_\sigma - 1)$.

```r
# Hyperparameters
# Weak prior for mu
mu_0 <- global_mean
v_0 <- 1e7
# Weak prior for tau2
a_tau <- 2
b_tau <- (a_tau - 1) * hat_tau2
# Weak prior for sigma2
a_sigma <- 2
b_sigma <- (a_sigma - 1) * hat_sigma2
```

With the parameters initialized, we proceed to the sampling stage. Since the setup is roughly the same as question 1, we can directly use the properties derived before and iteratively update each quantity. Specifically, the mathematical expressions for the posterior distributions are as follows:

- $\mu|\theta, \tau^2 \sim N(\mu^*, v^*)$ where $\mu^* = v^* \cdot (\frac{\mu_0}{v_0} + \frac{m\bar{\theta}}{\tau^2}), v^* = \frac{1}{m/\tau^2 + 1/v_0}$
- $\tau^2|\theta, \mu \sim$ Inverse-Gamma(a_τ^*, b_τ^*) where $a_\tau^* = a_\tau + \frac{m}{2}, b_\tau^* = b_\tau + \frac{\sum_{j=1}^m (\theta_j - \mu)^2}{2}$
- $\sigma^2|y, \theta \sim$ Inverse-Gamma(a_σ^*, b_σ^*) where $a_\sigma^* = a_\sigma + \frac{N}{2}, b_\sigma^* = b_\sigma + \frac{\sum_{j=1}^m \sum_{i=1}^{n_j} (y_{ij} - \theta_j)^2}{2}$
- $\theta_j|y_j, \mu, \tau^2, \sigma^2 \sim N(\mu^*, \tau^{2*})$ where $\mu^* = \tau^{2*} \cdot (\frac{\mu}{\tau^2} + \frac{\sum_{i=1}^{n_j} y_{ij}}{\sigma^2}), \tau^{2*} = \frac{1}{n_j/\sigma^2 + 1/\tau^2}$

Following this structure, we run the Gibbs sampler of $50,000$ iterations:

```r
# Initial values
mu <- mu_0
sigma2 <- hat_sigma2
tau2 <- hat_tau2
theta <- y_bar

# Storage for posterior samples
num_iter <- 50000
theta_samples <- matrix(0, nrow = num_iter, ncol = m)
mu_samples <- numeric(num_iter)
sigma2_samples <- numeric(num_iter)
tau2_samples <- numeric(num_iter)

set.seed(1218)
# Gibbs sampling iterations
for (iter in 1:num_iter) {
  # Step 1: Sample mu given theta and tau^2
  theta_bar <- mean(theta)
  mu_var <- 1 / (m / tau2 + 1 / v_0)
  mu_mean <- mu_var * (m * theta_bar / tau2 + mu_0 / v_0)
  mu <- rnorm(1, mu_mean, sqrt(mu_var))

  # Step 2: Sample tau^2 given theta and mu
  ssq_theta <- sum((theta - mu)^2)
  shape_tau2 <- m / 2 + a_tau
  rate_tau2 <- ssq_theta / 2 + b_tau
  tau2 <- 1 / rgamma(1, shape_tau2, rate_tau2)

  # Step 3: Sample sigma^2 given log_radon and theta
```

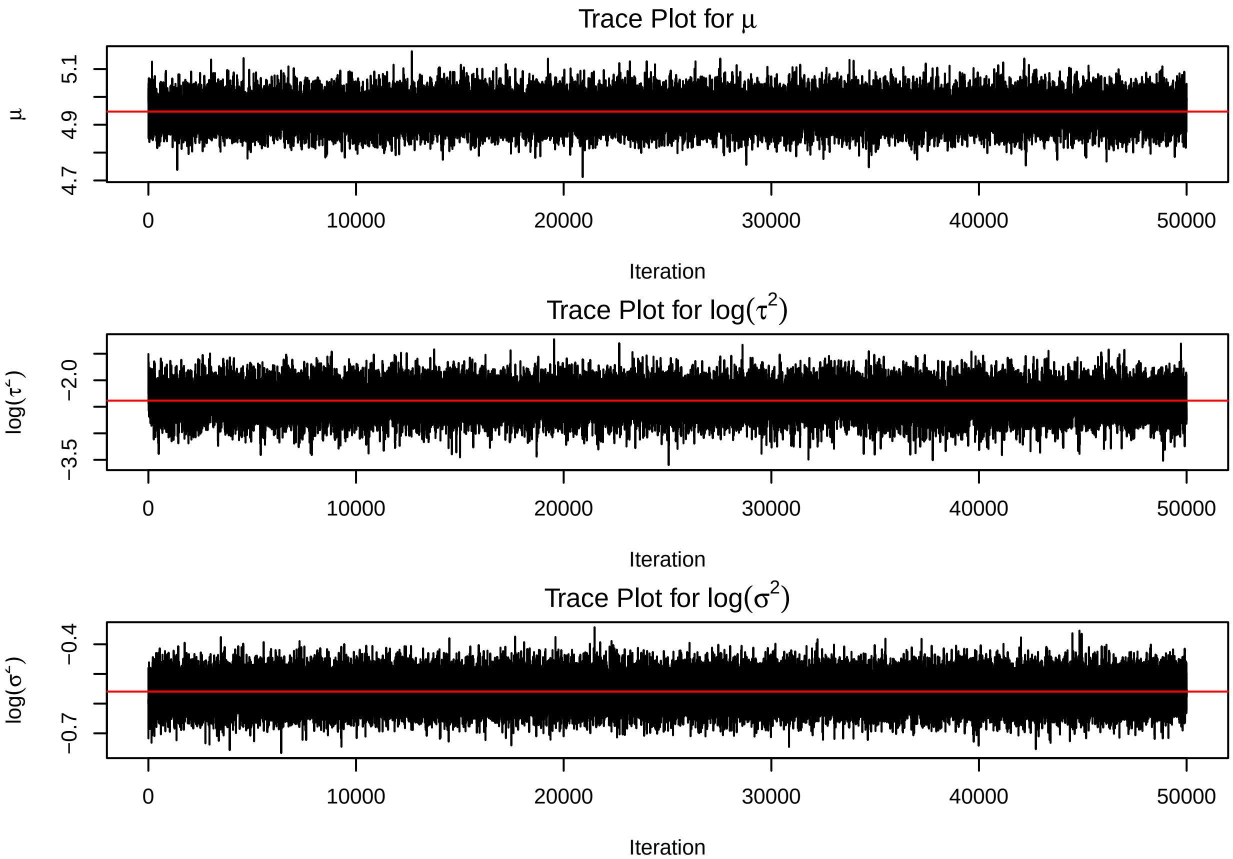
```r
  residuals <- log_radon - theta[group_indices]
  ssq_y <- sum(residuals^2)
  shape_sigma2 <- length(log_radon) / 2 + a_sigma
  rate_sigma2 <- ssq_y / 2 + b_sigma
  sigma2 <- 1 / rgamma(1, shape_sigma2, rate_sigma2)

  # Step 4: Sample each theta_j given log_radon, sigma^2, mu, tau^2
  for (j in 1:m) {
    group_indices_j <- which(group_indices == j)
    y_j <- log_radon[group_indices_j]
    n_j <- length(y_j)
    theta_var <- 1 / (n_j / sigma2 + 1 / tau2)
    theta_mean <- theta_var * (sum(y_j) / sigma2 + mu / tau2)
    theta[j] <- rnorm(1, theta_mean, sqrt(theta_var))
  }

  # Store samples
  theta_samples[iter, ] <- theta
  mu_samples[iter] <- mu
  tau2_samples[iter] <- tau2
  sigma2_samples[iter] <- sigma2
}
```

The trace plots are obtained thereafter:

```r
# Trace plots
par(mfrow = c(3, 1), mar = c(4, 4, 2, 1))
## mu
plot(mu_samples, type = "l", main = expression("Trace Plot for "*mu),
     xlab = "Iteration", ylab = expression(mu))
abline(h = mean(mu_samples), col = "red", lwd = 1)
## log tau^2
plot(log(tau2_samples), type = "l", main = expression("Trace Plot for "*log(tau^2)),
     xlab = "Iteration", ylab = expression(log(tau^2)))
abline(h = mean(log(tau2_samples)), col = "red", lwd = 1)
## log sigma^2
plot(log(sigma2_samples), type = "l", main = expression("Trace Plot for "*log(sigma^2)),
     xlab = "Iteration", ylab = expression(log(sigma^2)))
abline(h = mean(log(sigma2_samples)), col = "red", lwd = 1)
```



The traceplots show good mixing, as the samples densely cover the parameter space without evident trends or long-term drift. The horizontal red lines (posterior means) align well with the trace distributions, suggesting convergence of the Gibbs sampler.

(b)

```r
mu_ci <- quantile(mu_samples, probs = c(0.025, 0.975))
tau2_ci <- quantile(tau2_samples, probs = c(0.025, 0.975))
sigma2_ci <- quantile(sigma2_samples, probs = c(0.025, 0.975))

cat("95% Confidence Interval for mu: (", mu_ci, ")", "\n",
    "95% Confidence Interval for tau^2: (", tau2_ci, ")", "\n",
    "95% Confidence Interval for sigma^2: (", sigma2_ci, ")", "\n")
```

```
## 95% Confidence Interval for mu: ( 4.855134 5.041079 )
##  95% Confidence Interval for tau^2: ( 0.05306434 0.155591 )
##  95% Confidence Interval for sigma^2: ( 0.520636 0.6292083 )
```

```r
# Get CI from lmer
fit1 <- lmer(log(radon) ~ 1 + (1 | county), data = radon_df)
mu_lmer_ci <- confint(fit1, level = 0.95)["(Intercept)", ]
cat("95% Confidence Interval for mu (lmer): (", mu_lmer_ci, ")", "\n")
```

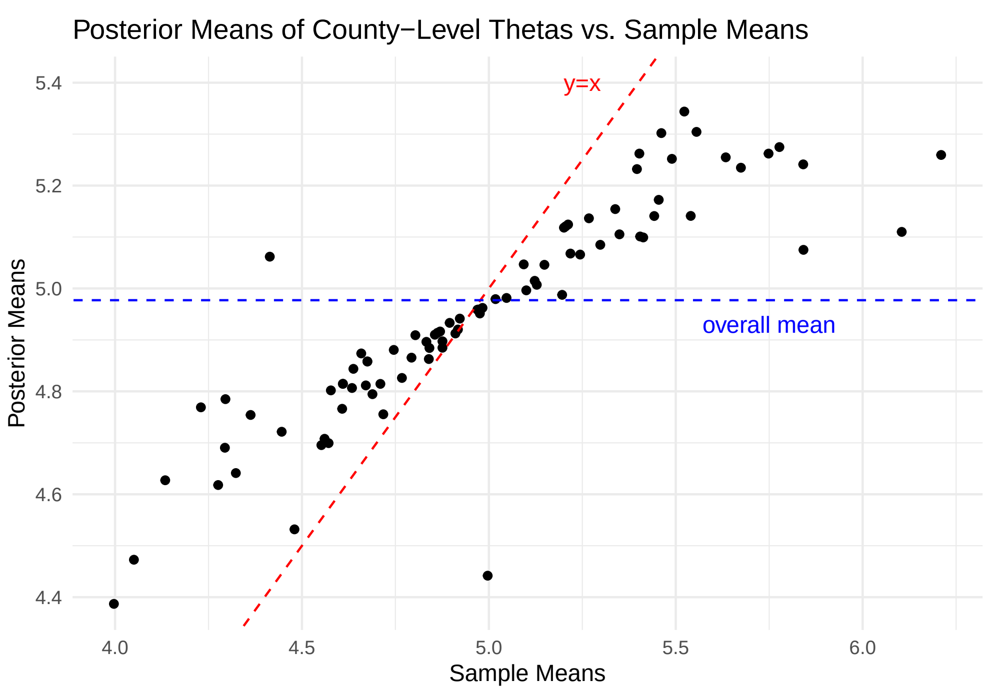
```
## 95% Confidence Interval for mu (lmer): ( 4.854804 5.040563 )
```

Comparing the two CIs for μ, we can see that they are almost the same with each other.

(c)

```r
# Compute posterior means theta_j
theta_posterior_means <- colMeans(theta_samples)
# Compute sample means ybar_j
sample_means <- tapply(log_radon, county, mean)
# Create a data frame for plotting
theta_comparison_df <- data.frame(
  PosteriorMeanTheta = theta_posterior_means,
  SampleMeanY = sample_means
) %>%
  rownames_to_column(var = "County")
```

```r
ggplot(theta_comparison_df, aes(x = SampleMeanY, y = PosteriorMeanTheta)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  annotate("text", x = 5.25, y = 5.4, label = "y=x", color = "red") +
  geom_abline(slope = 0, intercept = global_mean, linetype = "dashed", color = "blue") +
  annotate("text", x = 5.75, y = 4.93, label = "overall mean", color = "blue") +
  labs(title = "Posterior Means of County-Level Thetas vs. Sample Means",
       x = "Sample Means", y = "Posterior Means") +
  theme_minimal()
```



The plot shows that the posterior means of the county-level θ_j's exhibit a shrinkage effect, where they are pulled toward the global mean μ compared to the sample means $\bar{y}_j$'s. This is most apparent for counties with fewer observations, as their posterior means deviate more from the red calibration line `y=x`, indicating stronger influence from the prior. In contrast, counties with larger sample sizes have posterior means that align closely with their sample means, as the data dominates the prior.

(d)

First, we identify the top 5 counties with the largest sample means:

```
# Identify the top 5 counties in sample means
top5 <- names(sort(sample_means, decreasing = T)[1:5])
top5
```

```
## [1] "LACQUIPARLE" "MURRAY"      "WILKIN"      "WATONWAN"    "NICOLLET"
```

At the same time, we obtain the ranks of θ_j's in each iteration across all counties:

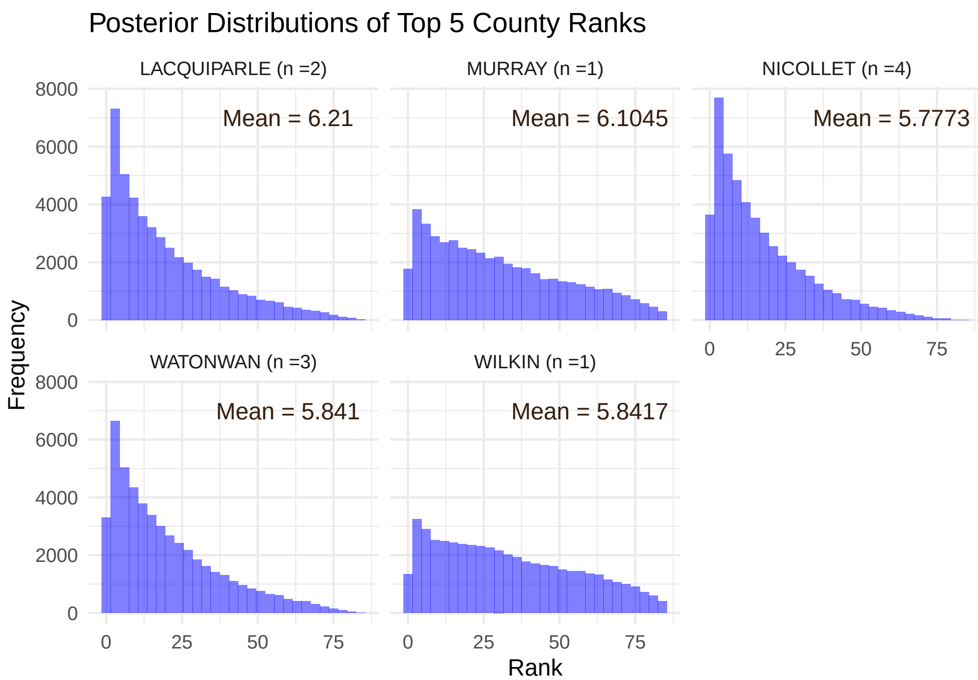
```
# Calculate ranks for each iteration across all counties
all_ranks <- t(apply(theta_samples, 1, function(x) rank(-x, ties.method = "min")))
# Retrieve ranks for the top 5 counties
col_idx <- match(top5, counties)
top5_ranks <- all_ranks[, col_idx]
# Create a dataframe
ranks_df <- as.data.frame(top5_ranks) %>%
  `colnames<-`(top5) %>%
  pivot_longer(
    cols = everything(),
    names_to = "County",
    values_to = "Rank"
  )
```

To add more information to our plot, we obtain the sample means and sample sizes of the five counties:

```
sample5 <- radon_df %>%
  filter(county %in% top5) %>%
  group_by(county) %>%
  summarise(
    sample_mean = mean(log(radon)),
    sample_size = n()) %>%
  rename(County = county)
```

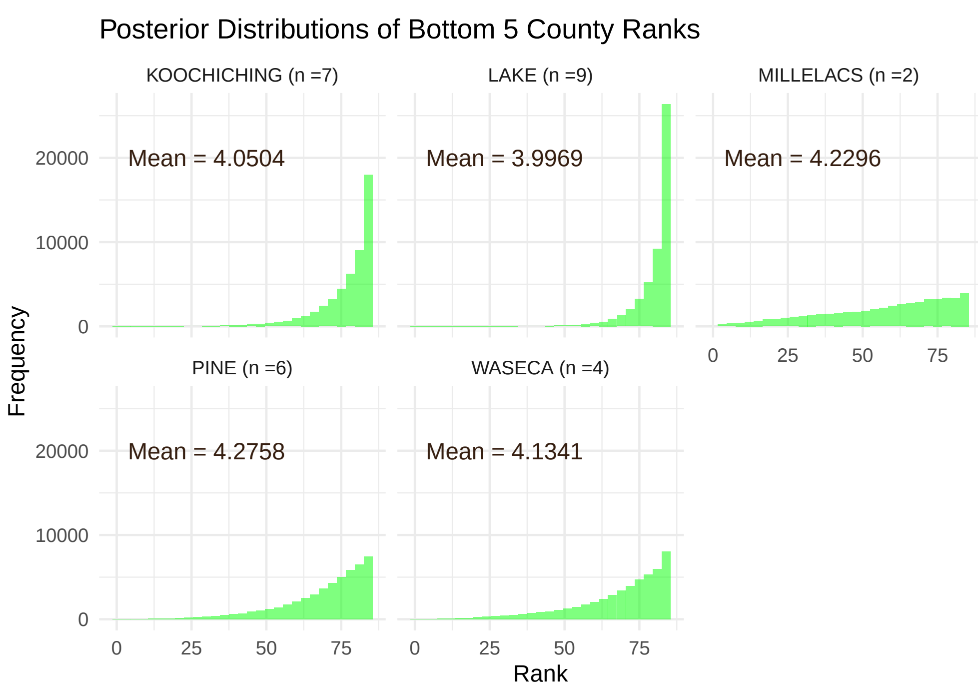
From here, we can plot the posterior distribution of the ranks:

```
# Create facet labels with sample sizes
facet_labels <- paste0(sample5$County, " (n =", sample5$sample_size, ")")
names(facet_labels) <- sample5$County
# Plot
ggplot(ranks_df, aes(x = Rank)) +
  geom_histogram(fill = "blue", alpha = 0.5, binwidth=3) +
  facet_wrap(~ County, labeller = labeller(County = facet_labels)) +
  labs(title = "Posterior Distributions of Top 5 County Ranks",
       x = "Rank", y = "Frequency") +
  theme_minimal() +
  geom_text(
    data = sample5 %>% select(-sample_size),
    aes(x = 60, y = 7000, label = paste("Mean =", round(sample_mean, 4))),
    color = "#351e10"
  )
```



From the plot, the distributions reflect a strong influence of sample size. Counties with smaller sample sizes, such as Murray and Wilkin, exhibit broader and more uncertain rank distributions, as the posterior ranks vary significantly across iterations. In contrast, counties with larger sample sizes, like Nicollet and Watonwan, display tighter rank distributions, indicating more stable posterior estimates.

For the sample means, since the values for the top 5 counties are very close to one another, the rank differences among these counties are subtle, making it challenging to discern clear separation based on ranks alone. Therefore, we can plot the distributions of the bottom five counties using the same approach (code omitted):



Here, the relationship between ranks and sample means becomes more evident. Counties with smaller sample means, such as Lake and Koochiching, consistently rank lower, with distributions skewed toward higher ranks (indicating worse performance). Additionally, the influence of sample size remains consistent: counties with smaller sample sizes, such as Millelacs and Waseca, show broader distributions.