575_HW3

Matthew Stoebe

2025-03-14

```
library(coda)
library(ggplot2)
```

a.

```
library(coda)
library(ggplot2)
set.seed(69)


nSchools <- 8
school_data <- vector("list", nSchools)

for (i in 1:nSchools) {
  fileName <- paste0("school", i, ".dat")
  school_data[[i]] <- scan(fileName, quiet = TRUE)
}

n_i <- sapply(school_data, length)
ybar <- sapply(school_data, mean)
all_data <- unlist(school_data)
N <- length(all_data)


mu0      <- 7
sigma2_mu0 <- 5
nu0      <- 2
sigma20  <- 15
eta0     <- 2
tau20    <- 10


n_iter  <- 10000
burn_in <- 1000

theta_samples <- matrix(NA, nrow = n_iter, ncol = nSchools)
mu_samples    <- numeric(n_iter)
sigma2_samples <- numeric(n_iter)
tau2_samples  <- numeric(n_iter)
R_samples     <- numeric(n_iter)
```

```r
mu_current     <- mu0
sigma2_current <- sigma20
tau2_current   <- tau20
theta_current  <- ybar


for (iter in 1:n_iter) {
  for (i in 1:nSchools) {
    vi <- 1 / (n_i[i] / sigma2_current + 1 / tau2_current)
    mi <- vi * (n_i[i] * ybar[i] / sigma2_current + mu_current / tau2_current)
    theta_current[i] <- rnorm(1, mean = mi, sd = sqrt(vi))
  }

  v_mu <- 1 / (nSchools / tau2_current + 1 / sigma2_mu0)
  m_mu <- v_mu * (sum(theta_current) / tau2_current + mu0 / sigma2_mu0)
  mu_current <- rnorm(1, mean = m_mu, sd = sqrt(v_mu))

  nu_new <- nu0 + N
  sum_sq <- 0
  for (i in 1:nSchools) {
    sum_sq <- sum_sq + sum((school_data[[i]] - theta_current[i])^2)
  }
  scale_sigma2 <- (nu0 * sigma20 + sum_sq) / nu_new
  sigma2_current <- nu_new * scale_sigma2 / rchisq(1, df = nu_new)

  nu_tau <- eta0 + nSchools
  sum_theta_sq <- sum((theta_current - mu_current)^2)
  scale_tau2 <- (eta0 * tau20 + sum_theta_sq) / nu_tau
  tau2_current <- nu_tau * scale_tau2 / rchisq(1, df = nu_tau)

  R_samples[iter] <- 1 + sigma2_current / tau2_current

  theta_samples[iter, ] <- theta_current
  mu_samples[iter]     <- mu_current
  sigma2_samples[iter] <- sigma2_current
  tau2_samples[iter]   <- tau2_current
}

theta_samples <- theta_samples[-(1:burn_in), ]
mu_samples    <- mu_samples[-(1:burn_in)]
sigma2_samples <- sigma2_samples[-(1:burn_in)]
tau2_samples  <- tau2_samples[-(1:burn_in)]
R_samples     <- R_samples[-(1:burn_in)]


mu_mcmc     <- mcmc(mu_samples)
sigma2_mcmc <- mcmc(sigma2_samples)
tau2_mcmc   <- mcmc(tau2_samples)

ess_mu     <- effectiveSize(mu_mcmc)
ess_sigma2 <- effectiveSize(sigma2_mcmc)
ess_tau2   <- effectiveSize(tau2_mcmc)
```

```r
cat("Effective Sample Size for mu:", round(ess_mu, 1), "\n")
```

```
## Effective Sample Size for mu: 7789
```

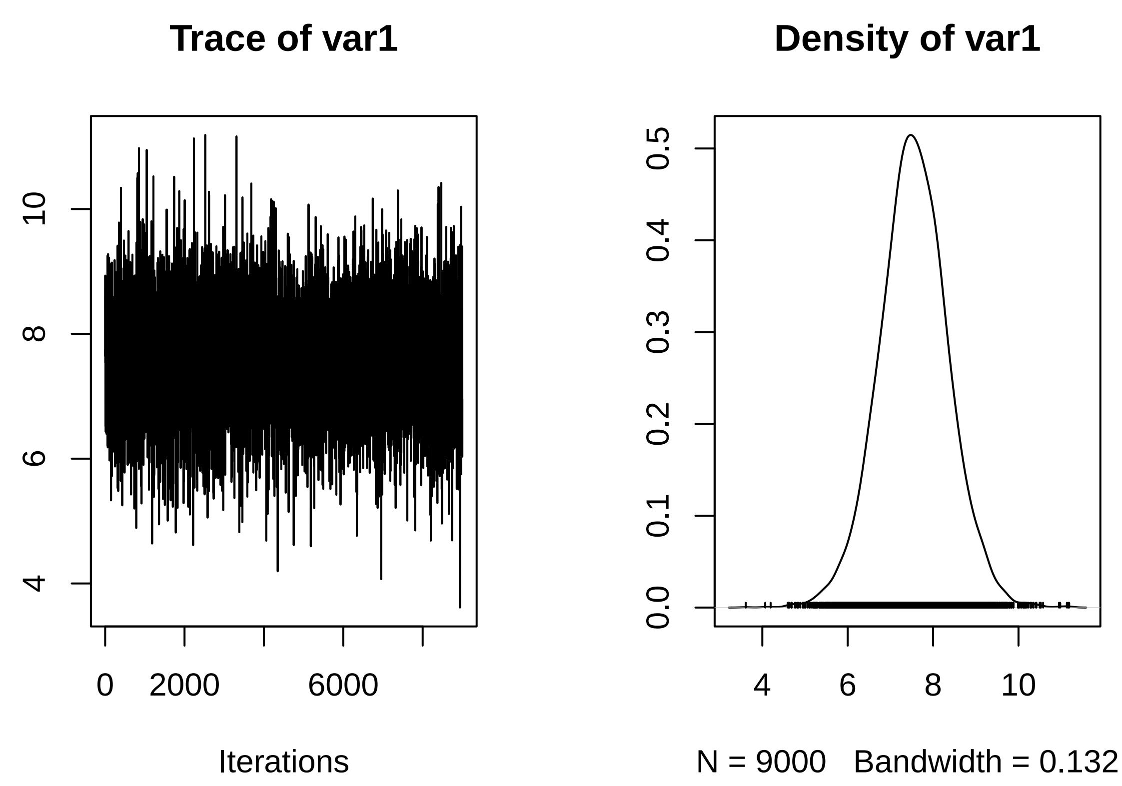
```r
cat("Effective Sample Size for sigma2:", round(ess_sigma2, 1), "\n")
```

```
## Effective Sample Size for sigma2: 8484.7
```

```r
cat("Effective Sample Size for tau2:", round(ess_tau2, 1), "\n")
```

```
## Effective Sample Size for tau2: 6837
```

```r
plot(mu_mcmc)
```




```r
posterior_summary <- function(samples) {
  mean_val <- mean(samples)
  ci <- quantile(samples, c(0.025, 0.975))
  list(mean = mean_val, CI = ci)
}

mu_summary <- posterior_summary(mu_samples)
sigma2_summary <- posterior_summary(sigma2_samples)
```

```r
tau2_summary<- posterior_summary(tau2_samples)

cat("Posterior summary for mu:\n")
```

```
## Posterior summary for mu:
```

```r
print(mu_summary)
```

```
## $mean
## [1] 7.550256
##
## $CI
##     2.5%    97.5%
## 5.936771 9.162568
```

```r
cat("Posterior summary for sigma2:\n")
```

```
## Posterior summary for sigma2:
```

```r
print(sigma2_summary)
```

```
## $mean
## [1] 14.46881
##
## $CI
##     2.5%    97.5%
## 11.76766 17.72840
```

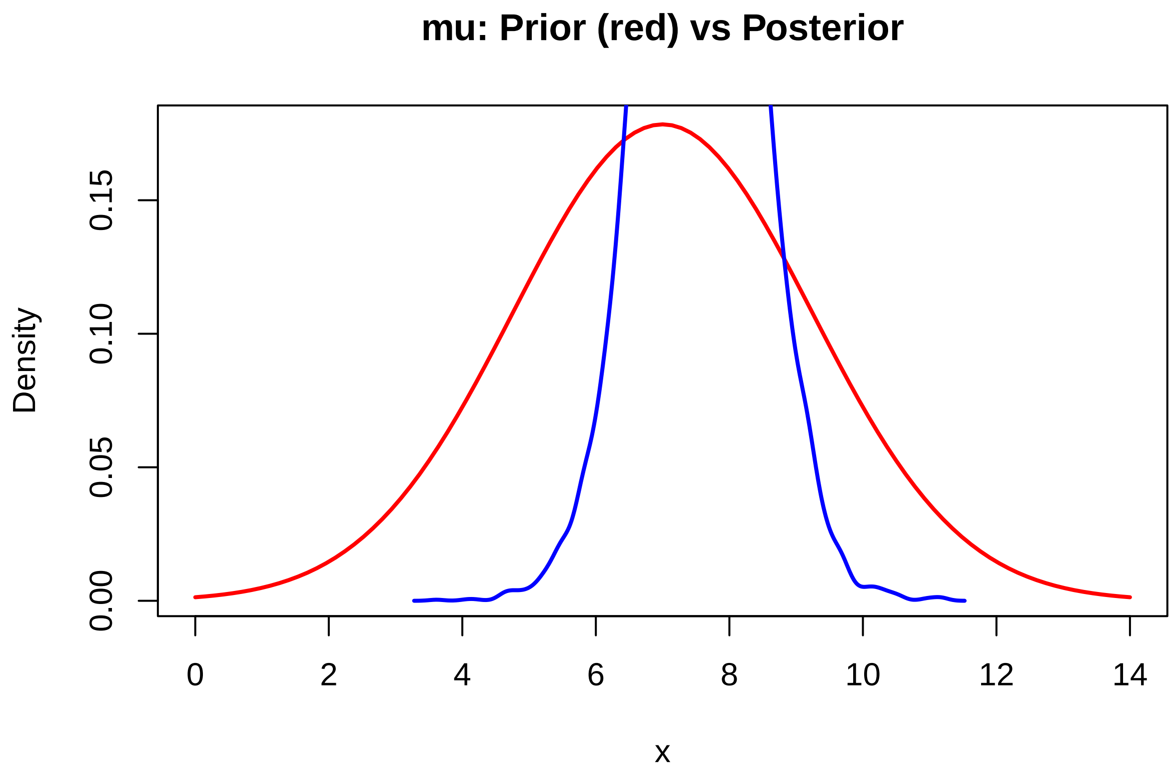
```r
cat("Posterior summary for tau2:\n")
```

```
## Posterior summary for tau2:
```

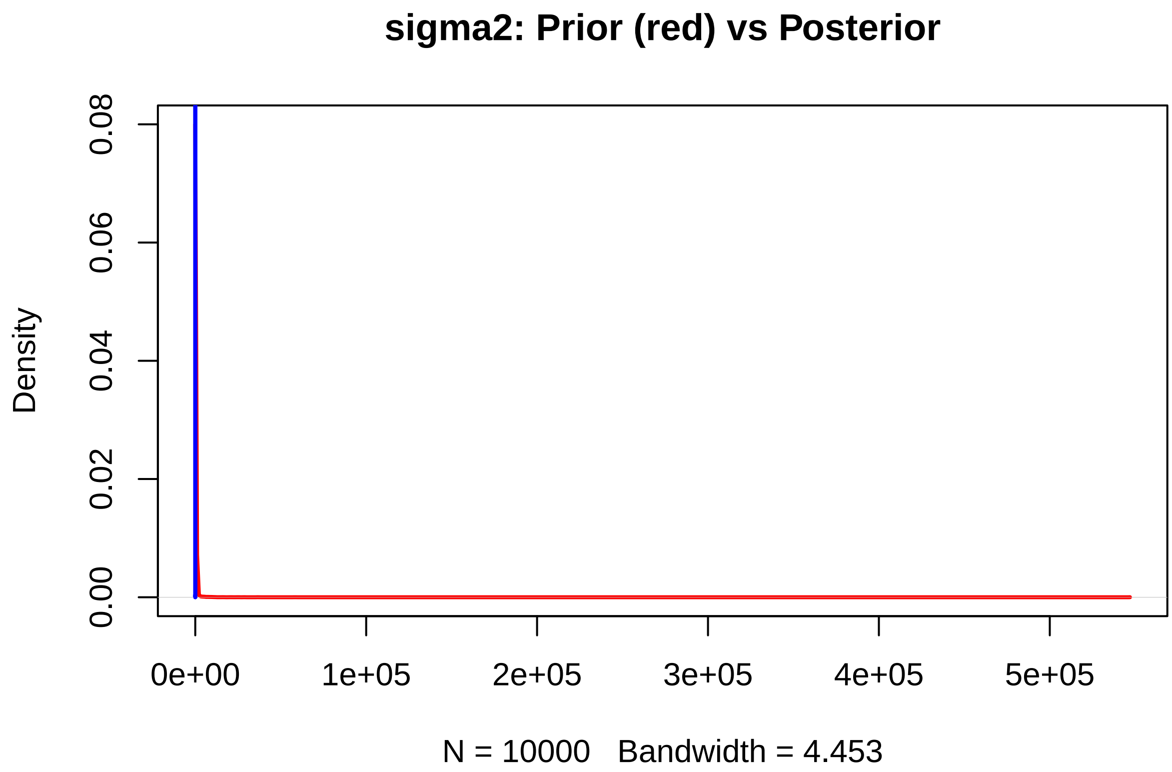
```r
print(tau2_summary)
```

```
## $mean
## [1] 5.553585
##
## $CI
##      2.5%     97.5%
##  1.895186 14.813303
```

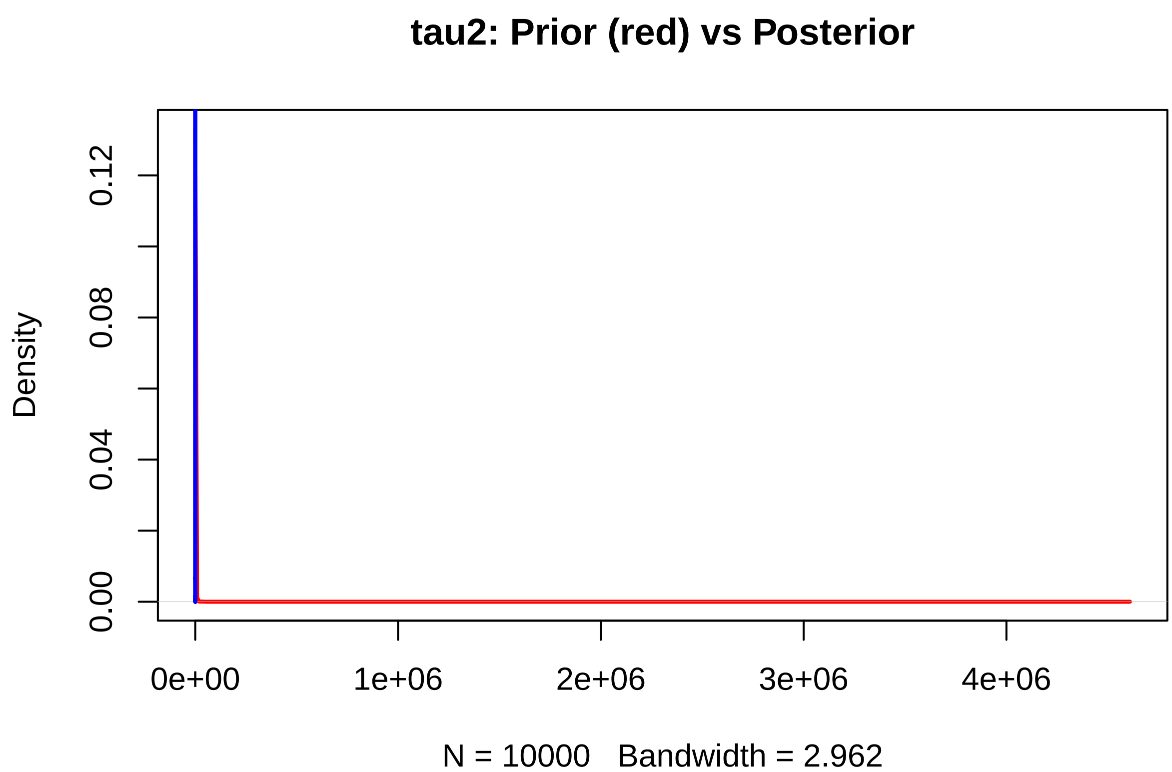
```r
curve(dnorm(x, mean = mu0, sd = sqrt(sigma2_mu0)),
      from = 0, to = 14, col = "red", lwd = 2, ylab = "Density",
      main = "mu: Prior (red) vs Posterior")
lines(density(mu_samples), col = "blue", lwd = 2)
```




```r
n_prior <- 10000
sigma2_prior <- nu0 * sigma20 / rchisq(n_prior, df = nu0)
plot(density(sigma2_prior), col = "red", lwd = 2,
     main = "sigma2: Prior (red) vs Posterior", ylab = "Density")
lines(density(sigma2_samples), col = "blue", lwd = 2)
```



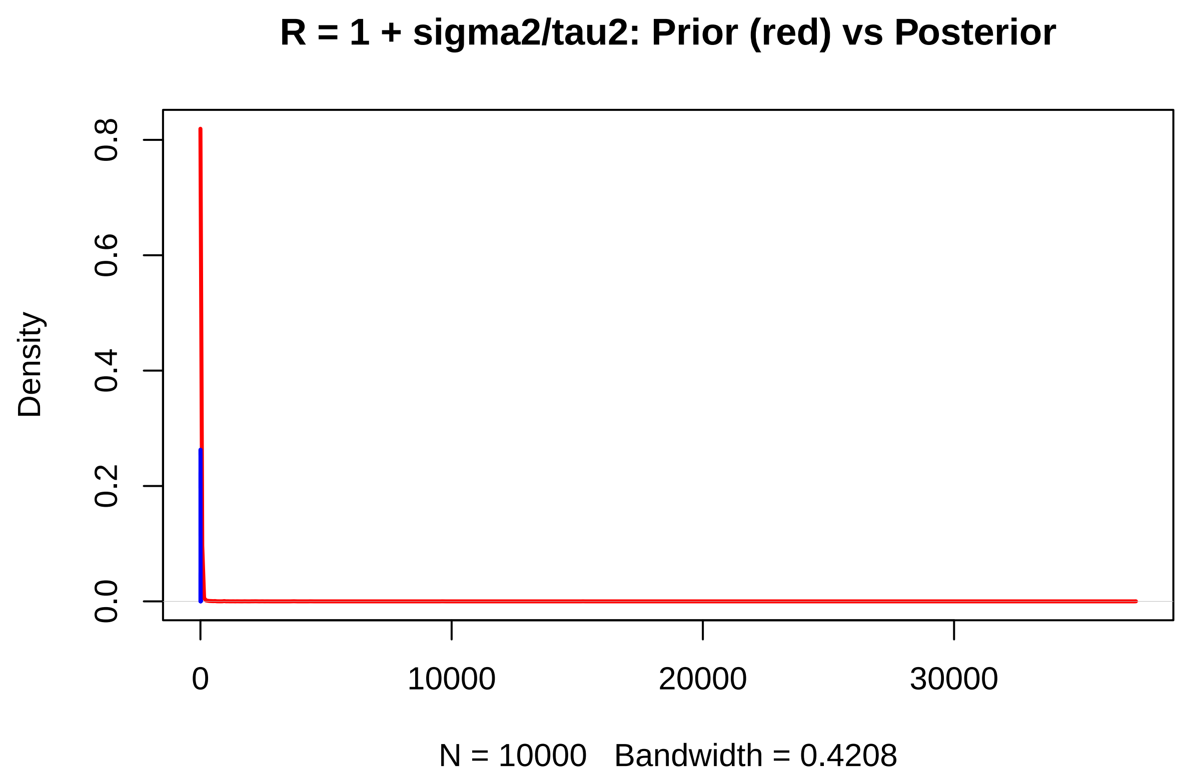

```
tau2_prior <- eta0 * tau20 / rchisq(n_prior, df = eta0)
plot(density(tau2_prior), col = "red", lwd = 2,
     main = "tau2: Prior (red) vs Posterior", ylab = "Density")
lines(density(tau2_samples), col = "blue", lwd = 2)
```




```r
par(mfrow = c(1, 1))


sigma2_prior_sim <- nu0 * sigma20 / rchisq(n_prior, df = nu0)
tau2_prior_sim   <- eta0 * tau20 / rchisq(n_prior, df = eta0)
R_prior <- 1 + sigma2_prior_sim / tau2_prior_sim

plot(density(R_prior), col = "red", lwd = 2,
     main = "R = 1 + sigma2/tau2: Prior (red) vs Posterior", ylab = "Density")
lines(density(R_samples), col = "blue", lwd = 2)
```



Im clearly doing something wrong here^^ but not sure what it is...

```
theta6_samples <- theta_samples[, 6]
theta7_samples <- theta_samples[, 7]

prob_theta7_lt_theta6 <- mean(theta7_samples < theta6_samples)
cat("Posterior probability that theta7 < theta6:", round(prob_theta7_lt_theta6, 3), "\n")
```

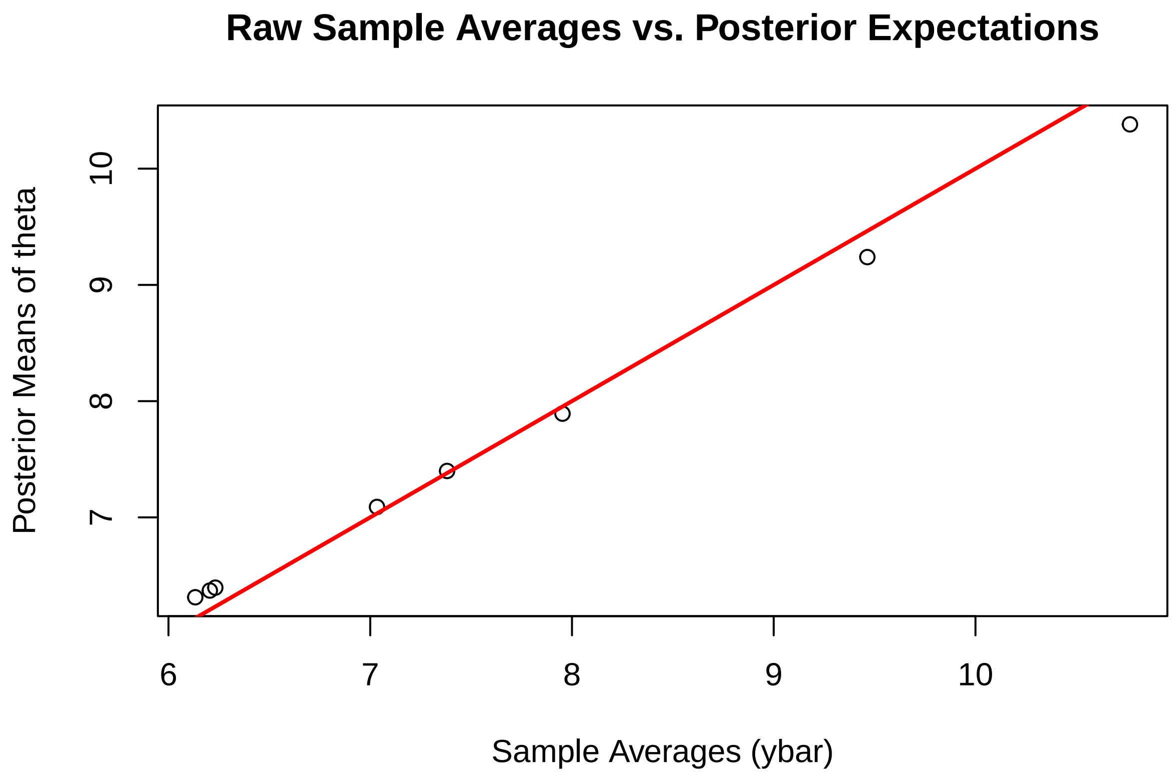
```
## Posterior probability that theta7 < theta6: 0.525
```

```
prob_theta7_smallest <- mean(apply(theta_samples, 1, function(x) which.min(x) == 7))
cat("Posterior probability that theta7 is the smallest:", round(prob_theta7_smallest, 3), "\n")
```

```
## Posterior probability that theta7 is the smallest: 0.329
```

```
theta_posterior_means <- colMeans(theta_samples)

plot(ybar, theta_posterior_means,
     xlab = "Sample Averages (ybar)", ylab = "Posterior Means of theta",
     main = "Raw Sample Averages vs. Posterior Expectations")
abline(0, 1, col = "red", lwd = 2)
```




```r
overall_ybar <- mean(all_data)
posterior_mu_mean <- mean(mu_samples)
cat("Overall sample mean:", round(overall_ybar, 3), "\n")
```

```
## Overall sample mean: 7.691
```

```r
cat("Posterior mean of mu:", round(posterior_mu_mean, 3), "\n")
```

```
## Posterior mean of mu: 7.55
```