

TMA4300 MCMC Practice Solutions

John Paige

The document is intended as a tool for students to practice coding up their own MCMC sampler. Students are welcome to work together in groups, and to ask questions as they progress in this worksheet. When coding, students should take turns:

1. *coding up various components of the samplers, and*
2. *looking over and double checking other group members' code as it is being written.*

Make sure your code is well commented and has understandable variable and function names.

Assume we are interested in the response, relative to a baseline, of a patient after being assigned to one of two groups. We have n patients in a treatment group that receive medication for a condition, and n other patients in a control group that receive a placebo. Measurements are obtained for the n patients in each group, denoted by Y_{Ti} and Y_{Ci} for the treatment and control groups respectively, and for $i = 1, 2, \dots, n$.

The responses are modeled as, for $G \in \{T, C\}$ denoting the patient group,

$$Y_{Gi} = \mu_G + \epsilon_{Gi},$$

where we model μ_G as a Gaussian latent effect with prior $\mu_T, \mu_C \mid \nu^2 \stackrel{iid}{\sim} N(0, \nu^2)$, we assume Gaussian error, $\epsilon_{Ti}, \epsilon_{Ci} \mid \nu^2 \stackrel{iid}{\sim} N(0, \sigma^2)$ for $i = 1, \dots, n$, and we place inverse gamma hyperpriors on both σ^2 and ν^2 so that $\sigma^2, \nu^2 \stackrel{iid}{\sim} \text{Inv-Gamma}(\alpha, \beta)$. Hence, our hyperpriors have density:

$$p(\sigma^2) = \frac{\beta^\alpha}{\Gamma(\alpha)} (1/\sigma^2)^{\alpha+1} \exp\{-\beta/\sigma^2\}$$
$$p(\nu^2) = \frac{\beta^\alpha}{\Gamma(\alpha)} (1/\nu^2)^{\alpha+1} \exp\{-\beta/\nu^2\}.$$

Let $\mathbf{Y}_T = Y_{T1}, \dots, Y_{Tn}$ and $\mathbf{Y}_C = Y_{C1}, \dots, Y_{Cn}$.

1

Draw a conditional dependency graph for $\mathbf{Y}_T, \mathbf{Y}_C, \mu_T, \mu_C, \sigma^2$, and ν^2 . (*Why can we ignore the ϵ terms here?*)

We can ignore the ϵ terms since we are already considering as random $\mathbf{Y}_T, \mathbf{Y}_C, \mu_T$, and μ_C , and $\epsilon_{Gi} = Y_{Gi} - \mu_G$, so there are no additional degrees of freedom for the ϵ terms. We are reparameterizing in terms of Y rather than ϵ .

Dependency graph drawn in the notes on Blackboard.

2

Give an expression that the posterior density, $p(\mu_T, \mu_C, \sigma^2, \nu^2 \mid \mathbf{Y}_T, \mathbf{Y}_C)$, is proportional to, removing unnecessary multiplicative constants. Make sure to factor out the final result into contributions due to the likelihood, latent effects, and hyperpriors.

Answer:

$$\begin{aligned}
p(\mu_T, \mu_C, \sigma^2, \nu^2 \mid \mathbf{Y}_T, \mathbf{Y}_C) &\propto p(\mathbf{Y}_T, \mathbf{Y}_C \mid \mu_T, \mu_C, \sigma^2, \nu^2) \times p(\mu_T, \mu_C \mid \sigma^2, \nu^2) \times p(\sigma^2, \nu^2) \\
&= \underbrace{p(\mathbf{Y}_T \mid \mu_T, \sigma^2)p(\mathbf{Y}_C \mid \mu_C, \sigma^2)}_{\text{likelihood}} \times \underbrace{p(\mu_T \mid \nu^2)p(\mu_C \mid \nu^2)}_{\text{latent effects}} \times \underbrace{p(\sigma^2)p(\nu^2)}_{\text{hyperpriors}} \\
&= \left\{ \prod_{i=1}^n p(Y_{Ti} \mid \mu_T, \sigma^2)p(Y_{Ci} \mid \mu_C, \sigma^2) \right\} \{p(\mu_T \mid \nu^2)p(\mu_C \mid \nu^2)\} \{p(\sigma^2)p(\nu^2)\} \\
&\propto \left\{ \prod_{i=1}^n \frac{1}{\sigma^2} \exp\left\{-\frac{1}{2\sigma^2}[(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2]\right\} \right\} \times \left\{ \frac{1}{\nu^2} \exp\left\{-\frac{1}{2\nu^2}(\mu_T^2 + \mu_C^2)\right\} \right\} \\
&\times \left\{ \left(\frac{1}{\sigma^2\nu^2}\right)^{\alpha+1} \exp\{-\beta(1/\sigma^2 + 1/\nu^2)\} \right\} \\
&= \left\{ \frac{1}{\sigma^{2n}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2]\right\} \right\} \times \left\{ \frac{1}{\nu^2} \exp\left\{-\frac{1}{2\nu^2}(\mu_T^2 + \mu_C^2)\right\} \right\} \\
&\times \left\{ \left(\frac{1}{\sigma^2\nu^2}\right)^{\alpha+1} \exp\{-\beta(1/\sigma^2 + 1/\nu^2)\} \right\}
\end{aligned}$$

3

Find the conditional density $p(\nu^2 \mid \mu_T, \mu_C, \sigma^2, \mathbf{Y}_T, \mathbf{Y}_C)$ and simplify it, removing unnecessary multiplicative constants and combining exponential terms. Name the distribution and provide its parameters.

Do the same for $p(\sigma^2 \mid \mu_T, \mu_C, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C)$ (and, optionally, for $p(\mu_T, \mu_C \mid \sigma^2, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C)$).

Answer:

$$\begin{aligned}
p(\nu^2 \mid \mu_T, \mu_C, \sigma^2, \mathbf{Y}_T, \mathbf{Y}_C) &\propto \left\{ \frac{1}{\sigma^{2n}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2]\right\} \right\} \times \left\{ \frac{1}{\nu^2} \exp\left\{-\frac{1}{2\nu^2}(\mu_T^2 + \mu_C^2)\right\} \right\} \\
&\times \left\{ \left(\frac{1}{\sigma^2\nu^2}\right)^{\alpha+1} \exp\{-\beta(1/\sigma^2 + 1/\nu^2)\} \right\} \\
&\propto \left\{ \frac{1}{\nu^2} \exp\left\{-\frac{1}{2\nu^2}(\mu_T^2 + \mu_C^2)\right\} \right\} \left\{ \left(\frac{1}{\nu^2}\right)^{\alpha+1} \exp\{-\beta/\nu^2\} \right\} \\
&= \left(\frac{1}{\nu^2}\right)^{(\alpha+1)+1} \exp\left\{-\left(\beta + \frac{1}{2}(\mu_T^2 + \mu_C^2)\right) \frac{1}{\nu^2}\right\}.
\end{aligned}$$

This is proportional to the kernel of an inverse gamma distribution with parameters $\alpha + 1$ and $\beta + \frac{1}{2}(\mu_T^2 + \mu_C^2)$, so:

$$\nu^2 \mid \mu_T, \mu_C, \sigma^2, \mathbf{Y}_T, \mathbf{Y}_C \sim \text{Inv-Gamma}(\alpha + 1, \beta + \frac{1}{2}(\mu_T^2 + \mu_C^2)).$$

$$\begin{aligned}
p(\sigma^2 \mid \mu_T, \mu_C, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C) &\propto \left\{ \frac{1}{\sigma^{2n}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\} \right\} \times \left\{ \frac{1}{\nu^2} \exp \left\{ -\frac{1}{2\nu^2} (\mu_T^2 + \mu_C^2) \right\} \right\} \\
&\times \left\{ \left(\frac{1}{\sigma^2 \nu^2} \right)^{\alpha+1} \exp \{ -\beta(1/\sigma^2 + 1/\nu^2) \} \right\} \\
&\propto \left\{ \frac{1}{\sigma^{2n}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\} \right\} \times \left\{ \left(\frac{1}{\sigma^2} \right)^{\alpha+1} \exp \{ -\beta/\sigma^2 \} \right\} \\
&= \left(\frac{1}{\sigma^2} \right)^{n+\alpha+1} \exp \left\{ -\beta/\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\} \\
&= \left(\frac{1}{\sigma^2} \right)^{n+\alpha+1} \exp \left\{ -\left(\beta + \frac{1}{2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right) \frac{1}{\sigma^2} \right\}.
\end{aligned}$$

This is the kernel of an inverse gamma distribution with parameters $\alpha + n$ and $\beta + \frac{1}{2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2]$. Hence,

$$\sigma^2 \mid \mu_T, \mu_C, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C \sim \text{Inv-Gamma}(\alpha + n, \beta + \frac{1}{2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2]).$$

Note: the inverse gamma prior is conjugate to the Gaussian likelihood provided the mean is known (as given in Lecture 7), and we are conditioning on μ_T and μ_C (and we know *a priori* that the mean of μ_T and μ_C are 0), so we know the Gaussian mean in both conditionals. Hence, from the outset it should be clear that the σ^2 and ν^2 conditionals will be inverse gamma.

$$\begin{aligned}
p(\mu_T, \mu_C \mid \sigma^2, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C) &\propto \left\{ \frac{1}{\sigma^{2n}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\} \right\} \times \left\{ \frac{1}{\nu^2} \exp \left\{ -\frac{1}{2\nu^2} (\mu_T^2 + \mu_C^2) \right\} \right\} \\
&\times \left\{ \left(\frac{1}{\sigma^2 \nu^2} \right)^{\alpha+1} \exp \{ -\beta(1/\sigma^2 + 1/\nu^2) \} \right\} \\
&\propto \left\{ \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\} \right\} \times \left\{ \exp \left\{ -\frac{1}{2\nu^2} (\mu_T^2 + \mu_C^2) \right\} \right\} \\
&\propto \exp \left\{ -\frac{1}{2\nu^2} (\mu_T^2 + \mu_C^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right\}.
\end{aligned}$$

Now,

$$\begin{aligned}
\exp \left\{ -\frac{1}{2\nu^2} \mu_T^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (Y_{Ti} - \mu_T)^2 \right\} &\propto \exp \left\{ -\frac{1}{2\nu^2} \mu_T^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n [-2Y_{Ti}\mu_T + \mu_T^2] \right\} \\
&= \exp \left\{ -\left(\frac{1}{2\nu^2} + \frac{n}{2\sigma^2} \right) \mu_T^2 + 2 \left(\frac{1}{2\sigma^2} \sum_{i=1}^n Y_{Ti} \right) \mu_T \right\} \\
&\propto \exp \left\{ -\left(\frac{1}{2\nu^2} + \frac{n}{2\sigma^2} \right) \left(\mu_T - \frac{2 \left(\frac{1}{2\sigma^2} \sum_{i=1}^n Y_{Ti} \right)}{2 \left(\frac{1}{2\nu^2} + \frac{n}{2\sigma^2} \right)} \right)^2 \right\} \\
&\propto \exp \left\{ -\frac{1}{2} \frac{\sigma^2 + n\nu^2}{\nu^2 \sigma^2} \left(\mu_T - \frac{\sum_{i=1}^n Y_{Ti}}{n + \frac{\sigma^2}{\nu^2}} \right)^2 \right\},
\end{aligned}$$

which is the kernel of a Gaussian distribution with mean $\frac{\sum_{i=1}^n Y_{Ti}}{n + \frac{\sigma^2}{\nu^2}}$ and variance $\frac{\nu^2 \sigma^2}{\sigma^2 + n\nu^2}$. Hence,

$$\mu_T \mid \sigma^2, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C \sim N\left(\frac{\sum_{i=1}^n Y_{Ti}}{n + \frac{\sigma^2}{\nu^2}}, \frac{\nu^2 \sigma^2}{\sigma^2 + n\nu^2}\right)$$

A similar argument shows $\mu_C \mid \sigma^2, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C \sim N\left(\frac{\sum_{i=1}^n Y_{Ci}}{n + \frac{\sigma^2}{\nu^2}}, \frac{\nu^2 \sigma^2}{\sigma^2 + n\nu^2}\right)$.

Note: The fact that the conditional density $p(\mu_T, \mu_C \mid \sigma^2, \nu^2, \mathbf{Y}_T, \mathbf{Y}_C)$ can be factored into the form $f(\mu_T) \cdot f(\mu_C)$ implies μ_T and μ_C are conditionally independent given $\sigma^2, \nu, \mathbf{Y}_T$, and \mathbf{Y}_C (also this should be clear from the dependence graph we drew earlier). The above conditional distributions for μ_T and μ_C should therefore be no surprise, since, conditional on the variance, ν^2 , a Gaussian prior is conjugate to a Gaussian likelihood.

4

Despite the fact that we could use a Gibbs step for every parameter due to the closed forms of the conditional distributions, assume we wish to take a bivariate Gaussian Metropolis step for the parameters μ_T and μ_C with step variance $\tau^2 I_2$ for 2×2 identity matrix I_2 . We will let $\mathbf{x} = (\sigma^2, \nu^2, \mu_T, \mu_C)$ be the current state of our MCMC sampler so that \mathbf{x}^j is the j -th element of \mathbf{x} .

What is the **log** acceptance probability of the proposal $\mathbf{y} = (\sigma^2, \nu^2, \tilde{\mu}_T, \tilde{\mu}_C)$ where $\mathbf{y}^{3,4} \sim N_2(\mathbf{x}^{3,4}, \tau^2 I_2)$? Simplify the expression as much as possible.

Answer:

$$\begin{aligned} \log \alpha(\mathbf{y} \mid \mathbf{x}) &= \min \left\{ 0, \log \left(\frac{p(\mathbf{y}^{3,4} \mid \mathbf{y}^{1,2}) Q(\mathbf{x} \mid \mathbf{y})}{p(\mathbf{x}^{3,4} \mid \mathbf{x}^{1,2}) Q(\mathbf{y} \mid \mathbf{x})} \right) \right\} \\ &= \min \left\{ 0, \log p(\mathbf{y}^{3,4} \mid \mathbf{y}^{1,2}) - \log p(\mathbf{x}^{3,4} \mid \mathbf{x}^{1,2}) \right\} \\ &= \min \left\{ 0, -\frac{1}{2\nu^2}(\tilde{\mu}_T^2 + \tilde{\mu}_C^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \tilde{\mu}_T)^2 + (Y_{Ci} - \tilde{\mu}_C)^2] \right. \\ &\quad \left. - \left(-\frac{1}{2\nu^2}(\mu_T^2 + \mu_C^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \mu_T)^2 + (Y_{Ci} - \mu_C)^2] \right) \right\} \\ &= \min \left\{ 0, -\frac{1}{2\nu^2}(\tilde{\mu}_T^2 + \tilde{\mu}_C^2 - \mu_T^2 - \mu_C^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n [(Y_{Ti} - \tilde{\mu}_T)^2 + (Y_{Ci} - \tilde{\mu}_C)^2 - (Y_{Ti} - \mu_T)^2 - (Y_{Ci} - \mu_C)^2] \right\}. \end{aligned}$$

No need to spend more time simplifying than this.

5

Now for the fun part: we will code up our own hybrid Metropolis-within-Gibbs MCMC sampler. We will let $\mathbf{x} = (\sigma^2, \nu^2, \mu_T, \mu_C)$ so that \mathbf{x}_i^j is the j -th element of the i -th sample from our MCMC sampler. The MCMC algorithm will proceed as follows:

1. Set $\mathbf{x} \leftarrow \mathbf{x}_0 \leftarrow (\sigma^2, \nu^2, \mu_T, \mu_C)$ to a reasonable initial value
2. For i in 1 to M iterations:
 - (a) Gibbs step: set x^1 to a random draw from the conditional $x^1 \mid \mathbf{x}^{-1}, \mathbf{Y}_T, \mathbf{Y}_C$
 - (b) Gibbs step: set x^2 to a random draw from the conditional $x^2 \mid \mathbf{x}^{-2}, \mathbf{Y}_T, \mathbf{Y}_C$
 - (c) MH step: propose $\mathbf{y}^{3,4} \sim N_2(\mathbf{x}^{3,4}, \tau^2 I_2)$. Accept the proposal with the probability from problem 4, setting $\mathbf{x}^{3,4} \leftarrow \mathbf{y}^{3,4}$ if accept, or $\mathbf{x}^{3,4} \leftarrow \mathbf{x}_{i-1}^{3,4}$ if reject.
 - (d) Update current state: set $\mathbf{x}_i \leftarrow \mathbf{x}$

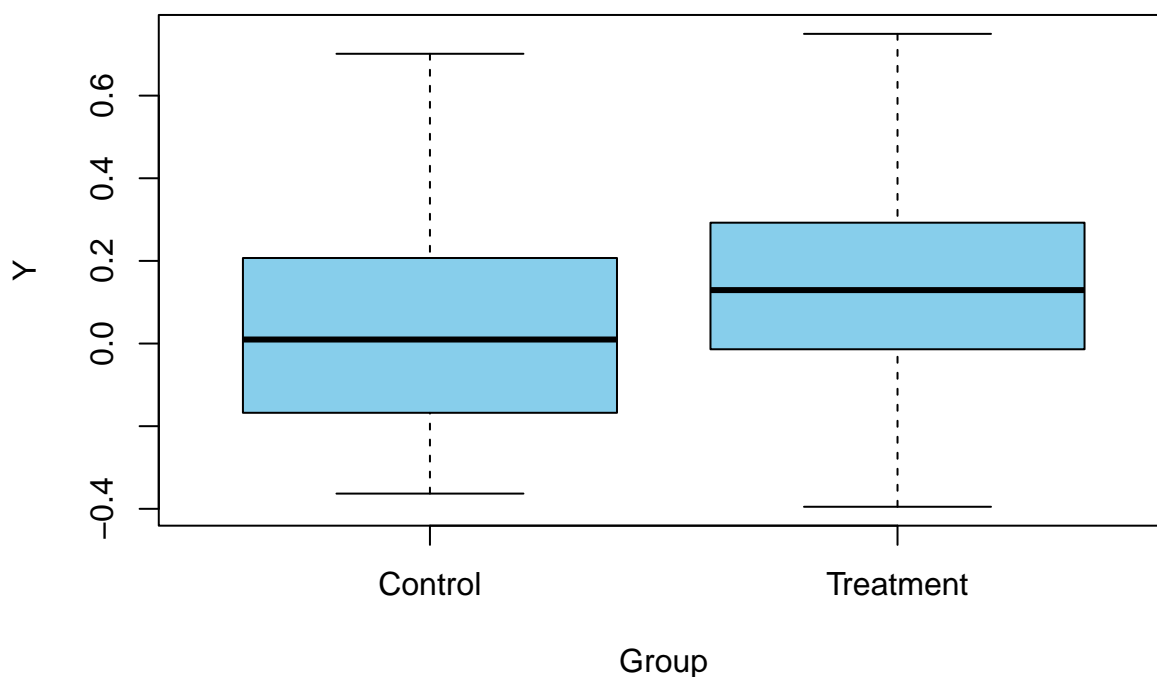
3. return x_0, \dots, x_M

Assume $M = 100000$ and remove 10000 samples due to burn-in. We will let the inverse gamma hyperprior parameters be $\alpha = 2$ and $\beta = 0.05$. Assume the following are our observations:

```
library(invgamma)

# simulate data based on true parameters
alpha = 2
beta = 0.05
set.seed(1)
sigma2 = rinvgamma(1, alpha, beta)
nu2 = rinvgamma(1, alpha, beta)
muT = rnorm(1, sd=sqrt(nu2))
muC = rnorm(1, sd=sqrt(nu2))
n=100
YT = rnorm(n, muT, sd=sqrt(sigma2))
YC = rnorm(n, muC, sd=sqrt(sigma2))

# make dataset and a boxplot of the responses in the 2 groups
dat = data.frame(Group=c(rep("Treatment", n), rep("Control", n)), Y=c(YT, YC))
boxplot(Y ~ Group, data=dat, col="skyblue")
```



Construct your MCMC sampler. Decide on a reasonable starting value and a step size resulting in reasonably good mixing. Keep track of your acceptance probability, and report it. Make traceplots and histograms of all variables, and compare estimates of the mean of each distribution to the true values generated above. Has the chain converged? What do the results imply about the treatment? Does it significantly improve (increase) patient outcomes relative to the placebo?

Answer:

First we define a function for the acceptance probability, then we define a Metropolis-Hastings function for doing MCMC.

```

# calculates the acceptance probability on a log scale. Unlike the notation in the solutions,
# I name variables "prop" instead of "tilde", since they are the proposed values.
acceptProb = function(muTprop, muCprop, muT, muC, nu2, sigma2) {
  firstTerm = (-0.5/nu2) * (muTprop^2 + muCprop^2 - muT^2 - muC^2)
  secondTerm = (-0.5/sigma2) * sum((YT - muTprop)^2 + (YC - muCprop)^2 - (YT - muT)^2 - (YC - muC)^2)
  min(c(1, exp(firstTerm + secondTerm)))
}

# function for doing Metropolis-Hastings. Keeps track of acceptance rate.
# returns:
#   sampleMat: (M-burnin) x 4 matrix of samples
#   acceptRate: proportion of MH steps accepted
MH = function(M=10000, burnin=2000, tau2=.1, printEvery=100, verbose=FALSE) {
  # initialize matrix of samples and acceptances and x_0. Note indices start at 1 not 0
  xmat = matrix(0, nrow=M+1, ncol=4)
  xmat[1,] = c(1, 1, 0, 0)
  accepts = rep(FALSE, M)

  # set current state (sigma2, nu2, muT, muC)
  x = xmat[1,]

  # generate samples
  startTime = proc.time()[3]
  for(i in 1:M) {
    if(((i %% printEvery) == 0) && verbose) {
      # print current state and expected computation time left:
      currState = paste(x, collapse=", ")
      print(paste0("current state for iteration ", i, "/", M, ": ", currState))

      currTime = proc.time()[3]
      timeTaken = currTime - startTime
      fracDone = (i-1)/M
      fracLeft = 1 - fracDone
      timeLeftEst = (timeTaken / fracDone) * fracLeft
      print(paste0("estimated time left: ", round(timeLeftEst), " seconds"))
    }

    # 2a: Gibbs step, draw sigma2 from invgam()
    shapeSigma2 = alpha+n
    rateSigma2 = beta + 0.5*(sum((YT - x[3])^2 + (YC - x[4])^2))
    x[1] = rinvgamma(1, shape=shapeSigma2, rate=rateSigma2)

    # 2b: Gibbs step, draw nu2 from invgam()
    shapeNu2 = alpha+1
    rateNu2 = beta + 0.5*(x[3]^2 + x[4]^2)
    x[2] = rinvgamma(1, shape=shapeNu2, rate=rateNu2)

    # 2c: MH step propose muT, muC from N(x[3:4], tau^2 I_2)
    ## proposal
    muTprop = rnorm(1, mean=x[3], sd=sqrt(tau2))
    muCprop = rnorm(1, mean=x[4], sd=sqrt(tau2))

    ## accept/reject step

```

```

acceptP = acceptProb(muTprop, muCprop, muT=x[3], muC=x[4], nu2=x[2], sigma2=x[1])
accept = runif(1) < acceptP
if(accept) {
  x[3:4] = c(muTprop, muCprop)
  accepts[i] = TRUE
}
# otherwise, x[3:4] is already set to the previous state, so we don't
# need to do anything

# 2d: store our sample. Note the index of xmat starts at 1 not 0, so
# xmat[i+1,] is the i-th sample
xmat[i+1,] = x
}

acceptRate=mean(accepts)

# print total time take and acceptance rate
currTime = proc.time()[3]
print(paste0("Total time taken: ", round((currTime - startTime)/60, digits=2), " minutes"))
print(paste0("Acceptance rate: ", round(acceptRate, digits=4)))
list(sampleMat = xmat[(burnin+1):(M+1),], acceptRate=acceptRate)
}

```

We assume $\tau^2 = .1$ to start, and eventually decrease it to $\tau^2 = 0.0025$ so the acceptance probability is between 20% and 50% (it is approximately 25%.) Now we run the MCMC sampler:

```

set.seed(123)

# run MH and collect results
results = MH(M=100000, burnin=10000, printEvery=10000, tau2 = .0025)

## [1] "Total time taken: 0.05 minutes"
## [1] "Acceptance rate: 0.2532"

samples = results$sampleMat
acceptRate = results$acceptRate

```

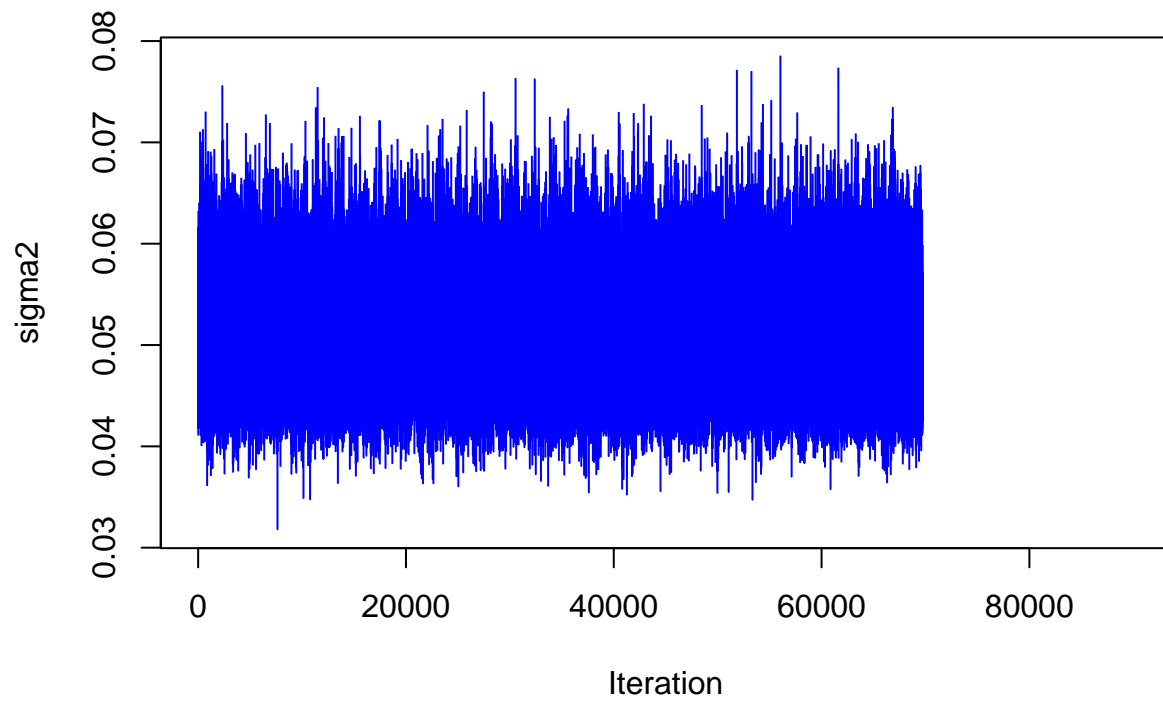
Plot the results:

```

# generate traceplots
plot(samples[,1], ylab="sigma2", type="l", col="blue", xlab="Iteration", main="Traceplot")

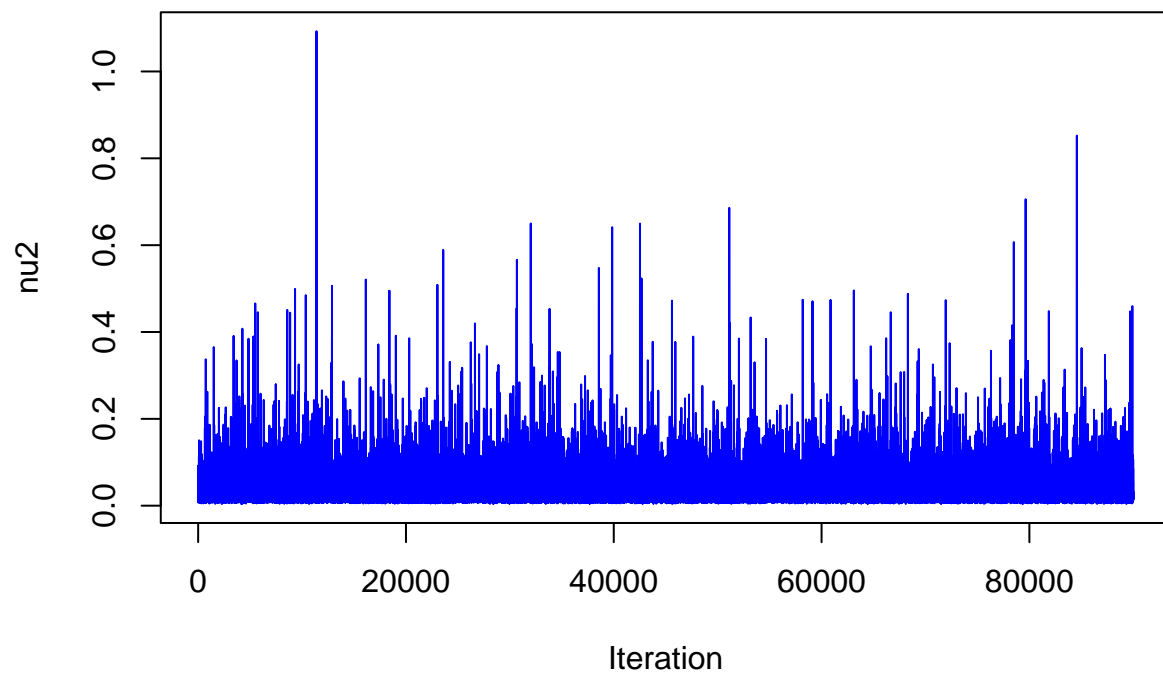
```

Traceplot



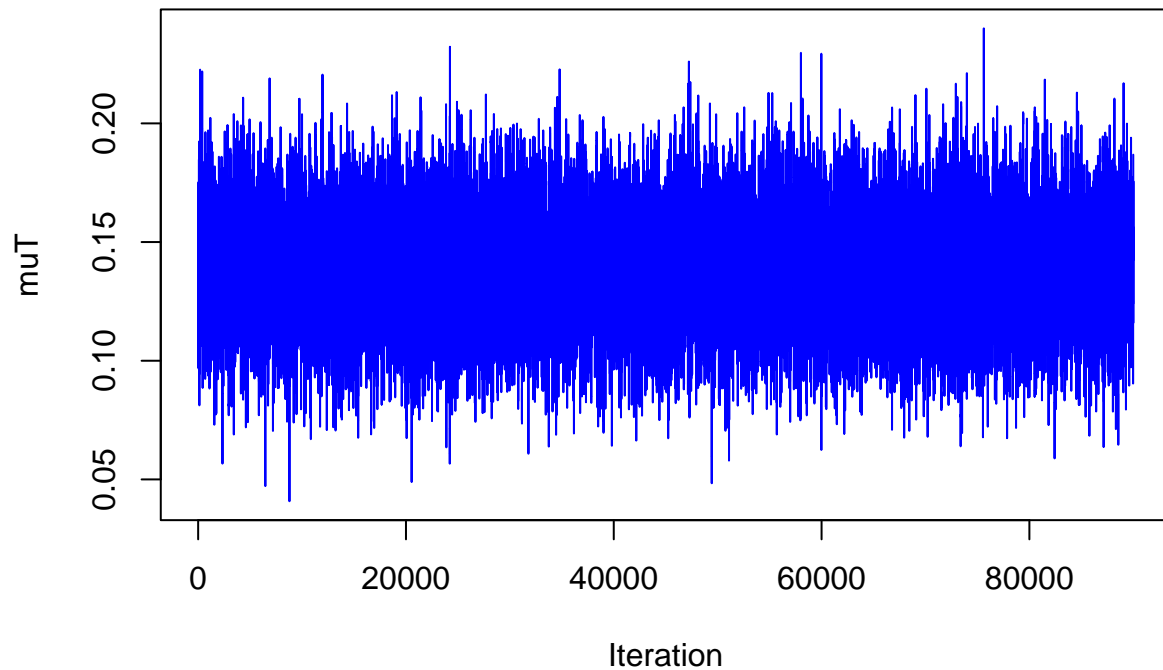
```
plot(samples[,2], ylab="nu2", type="l", col="blue", xlab="Iteration", main="Traceplot")
```

Traceplot



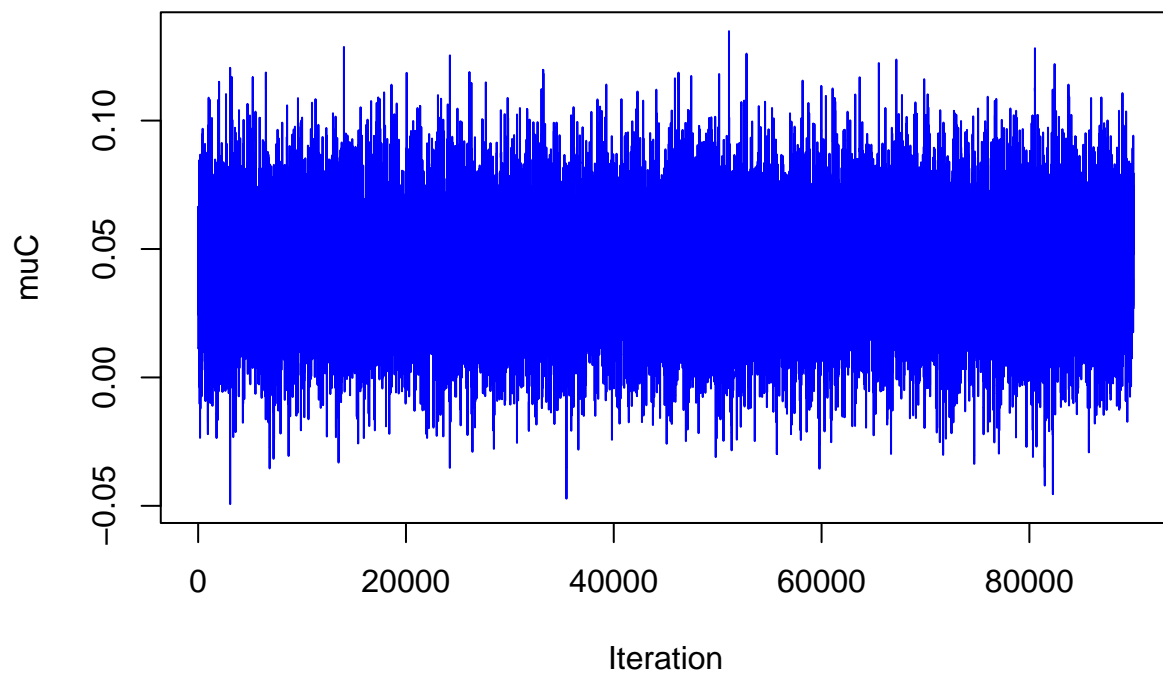
```
plot(samples[,3], ylab="muT", type="l", col="blue", xlab="Iteration", main="Traceplot")
```


Traceplot



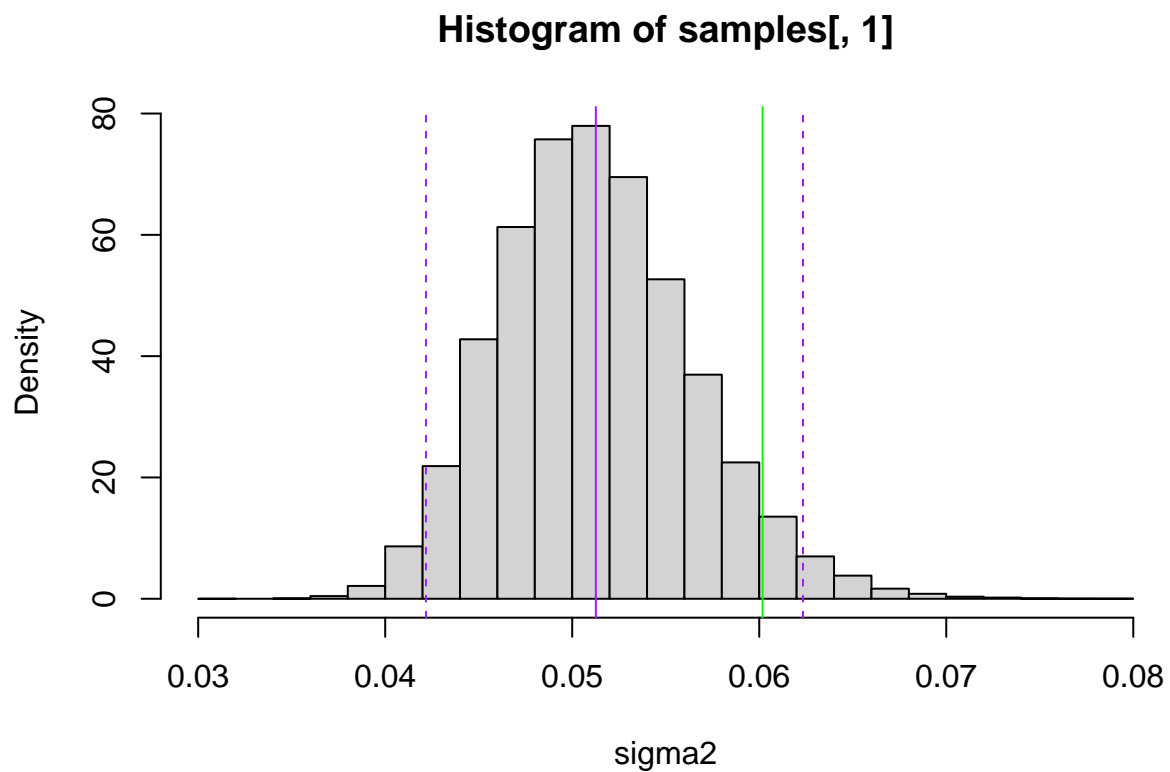
```
plot(samples[,4], ylab="muC", type="l", col="blue", xlab="Iteration", main="Traceplot")
```

Traceplot



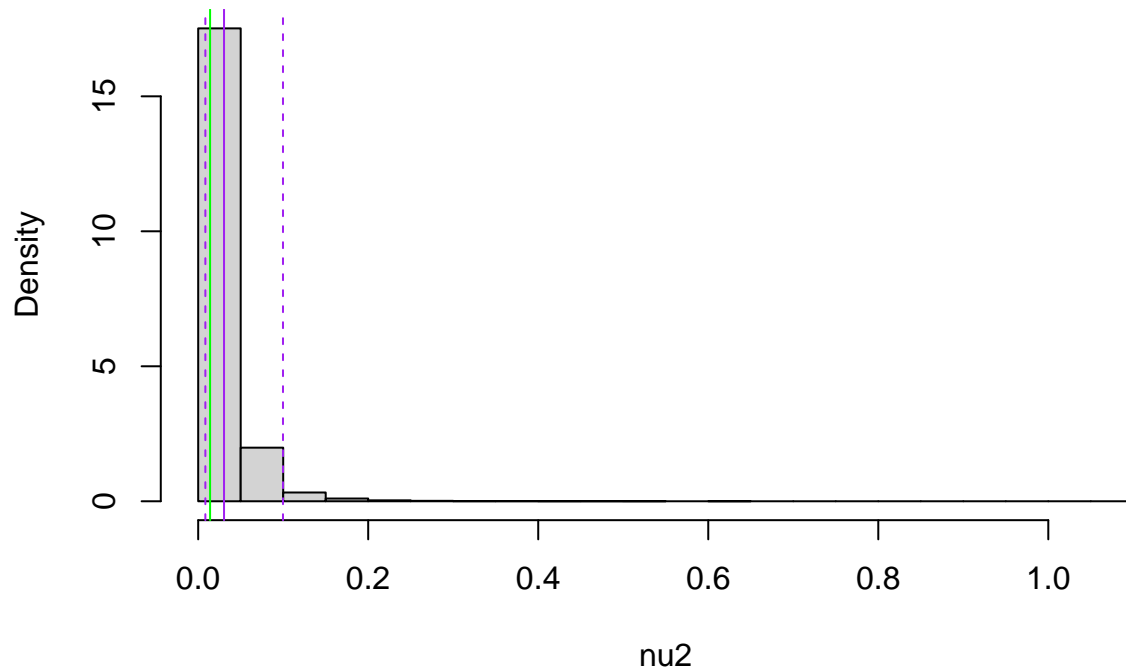
```
# generate histogram of marginals  
hist(samples[,1], xlab="sigma2", freq=F)  
abline(v=sigma2, col="green")  
abline(v=mean(samples[,1]), col="purple")
```

```
abline(v=quantile(samples[,1], prob=c(0.025, .975)), col="purple", lty=2)
```



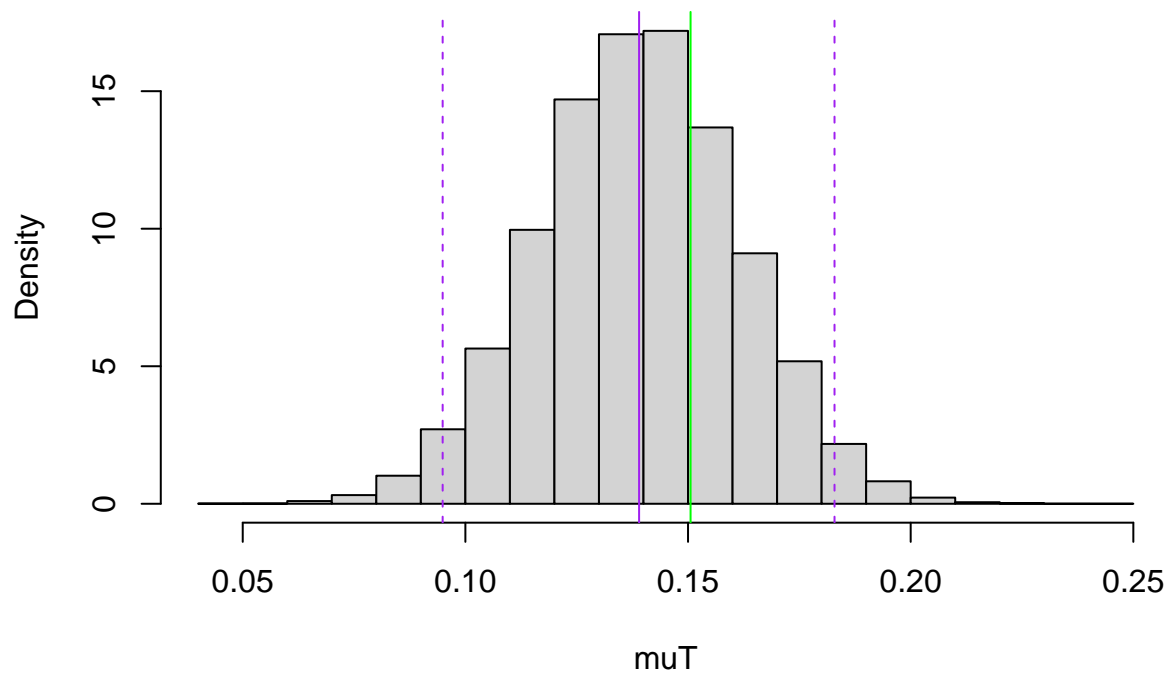
```
hist(samples[,2], xlab="nu2", freq=F)
abline(v=nu2, col="green")
abline(v=mean(samples[,2]), col="purple")
abline(v=quantile(samples[,2], prob=c(0.025, .975)), col="purple", lty=2)
```

Histogram of samples[, 2]

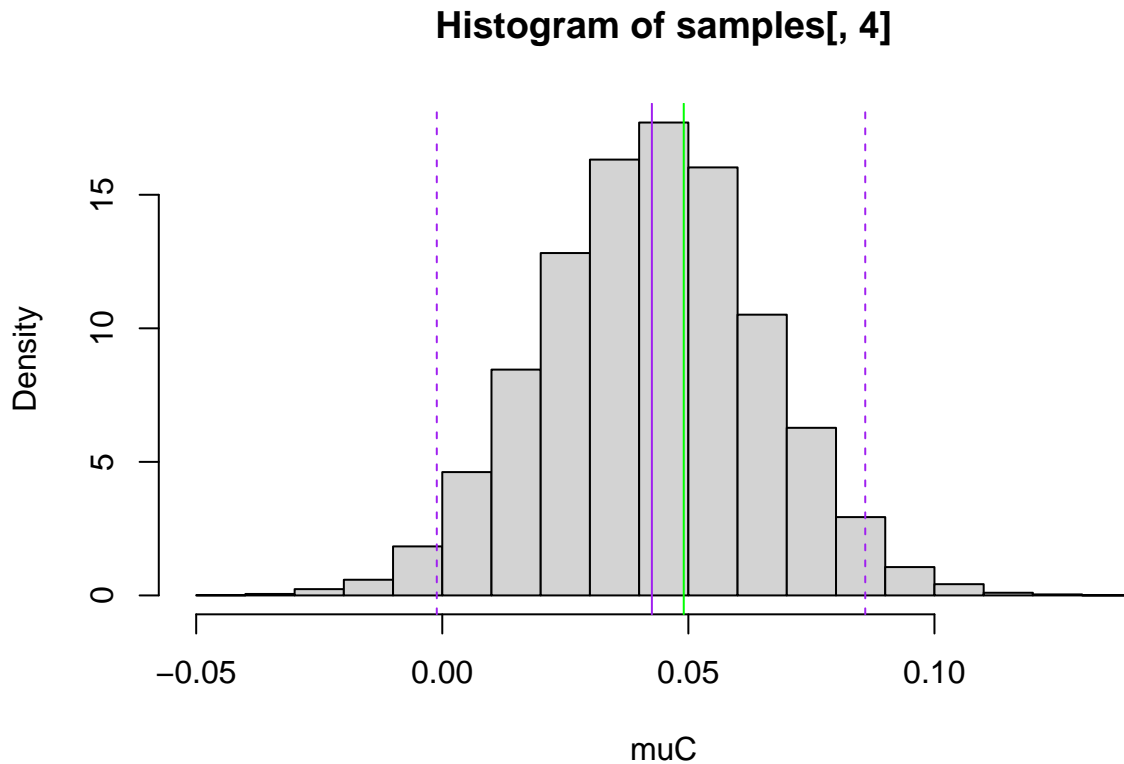


```
hist(samples[,3], xlab="muT", freq=F)
abline(v=muT, col="green")
abline(v=mean(samples[,3]), col="purple")
abline(v=quantile(samples[,3], prob=c(0.025, .975)), col="purple", lty=2)
```

Histogram of samples[, 3]



```
hist(samples[,4], xlab="muC", freq=F)
abline(v=muC, col="green")
abline(v=mean(samples[,4]), col="purple")
abline(v=quantile(samples[,4], prob=c(0.025, .975)), col="purple", lty=2)
```



Compare estimates to the truth:

```
print(paste0("sigma2 Est: ", mean(samples[,1])))
```

```
## [1] "sigma2 Est: 0.0512624009622565"
```

```
print(paste0("sigma2: ", sigma2))
```

```
## [1] "sigma2: 0.0601782477982532"
```

```
print(paste0("nu2 Est: ", mean(samples[,2])))
```

```
## [1] "nu2 Est: 0.0303954809716331"
```

```
print(paste0("nu2: ", nu2))
```

```
## [1] "nu2: 0.014002635615208"
```

```
print(paste0("muT Est: ", mean(samples[,3])))
```

```
## [1] "muT Est: 0.1390085796288"
```

```
print(paste0("muT: ", muT))
```

```
## [1] "muT: 0.15057003870223"
```

```
print(paste0("muC Est: ", mean(samples[,4])))
```

```
## [1] "muC Est: 0.0425913284846819"
```

```
print(paste0("muC: ", muC))
```

```
## [1] "muC: 0.0490656540070256"
```

Our estimates are pretty good! The worst estimate is ν^2 , which makes sense, because there are only two latent effects directly depending on the variance ν^2 .

```
diffSamples = samples[,3] - samples[,4]
quants = quantile(diffSamples, prob=c(.025, .975))
print(paste0("95% CI for muT - muC: (", quants[1], ", ", quants[2], ")"))
```

```
## [1] "95% CI for muT - muC: (0.0347004833197217, 0.15779398061979)"
```

Based on the data and our prior beliefs, we conclude that the treatment is very likely to improve patient outcomes. Here is our posterior for the distribution of the improvement in patient outcomes relative to the placebo along with a 95% credible interval:

```
diffSamples = samples[,3] - samples[,4]
quants = quantile(diffSamples, prob=c(.025, .975))
hist(diffSamples, freq=FALSE, col="skyblue", xlab="muT - muC posterior", main="Posterior of muT - muC",
abline(v=mean(diffSamples), col="purple")
abline(v=quants, lty=2, col="purple")
```

Posterior of muT – muC

