STA 601/360 Homework 8

Yi Mi

07 十一月, 2019

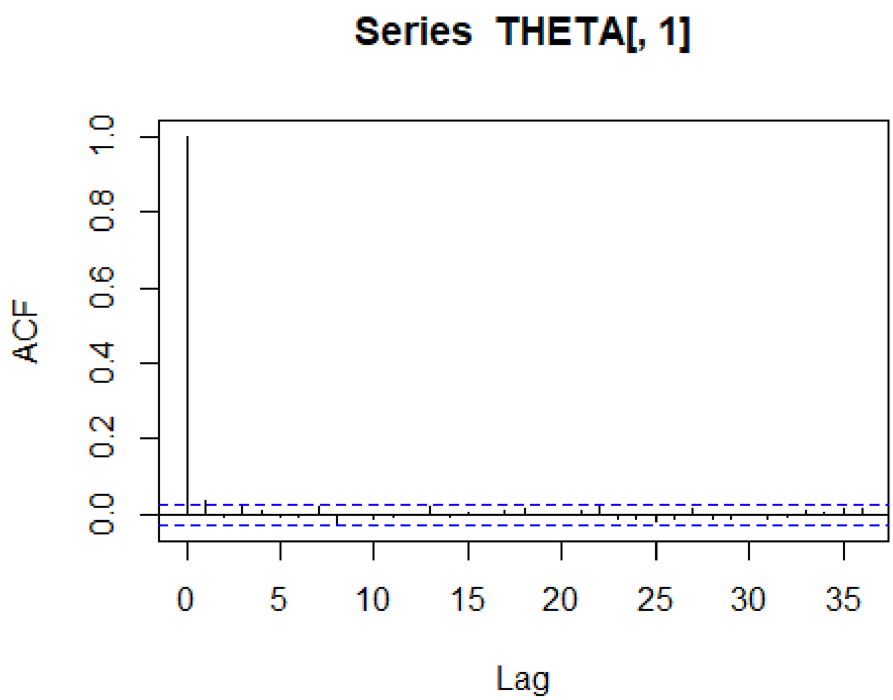
Exercise 1

Hoff 8.3

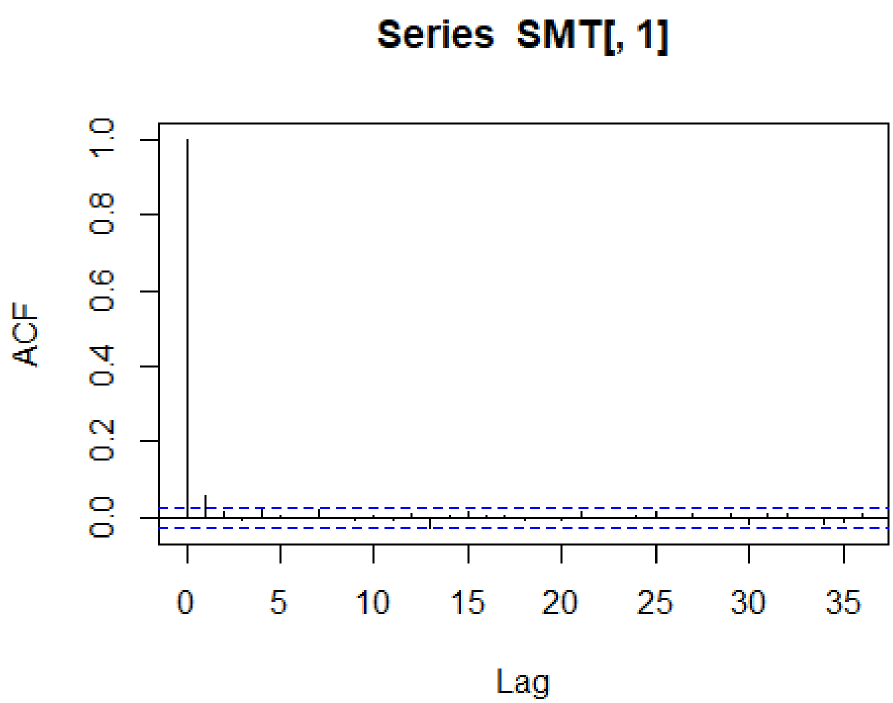
Part (a)

```
### read data
link = c()
Y = data.frame()
for(i in 1:8){
  link = paste0("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises/school
",i,'.dat', collapse = '')
  school = read.table(url(link))
  Y = rbind(Y, data.frame(school = i, h = school))
}

### weakly informative priors
nu0 = 2; s20 = 15
eta0 = 2; t20 = 10
mu0 = 7; g20 = 5
###

### starting values
m = length(unique(Y[, 1]))
n = sv = ybar = rep(NA,m)
for(j in 1:m){
  ybar[j] = mean(Y[Y[, 1] == j, 2])
  sv[j] = var(Y[Y[, 1] == j, 2])
  n[j] = sum(Y[, 1] == j)
}
theta = ybar
sigma2 = mean(sv)
mu = mean(theta)
tau2 = var(theta)
###

### setup MCMC
set.seed(15)
S = 5000
THETA = matrix(nrow = S, ncol = m)
SMT = matrix(nrow = S, ncol = 3)
###
```

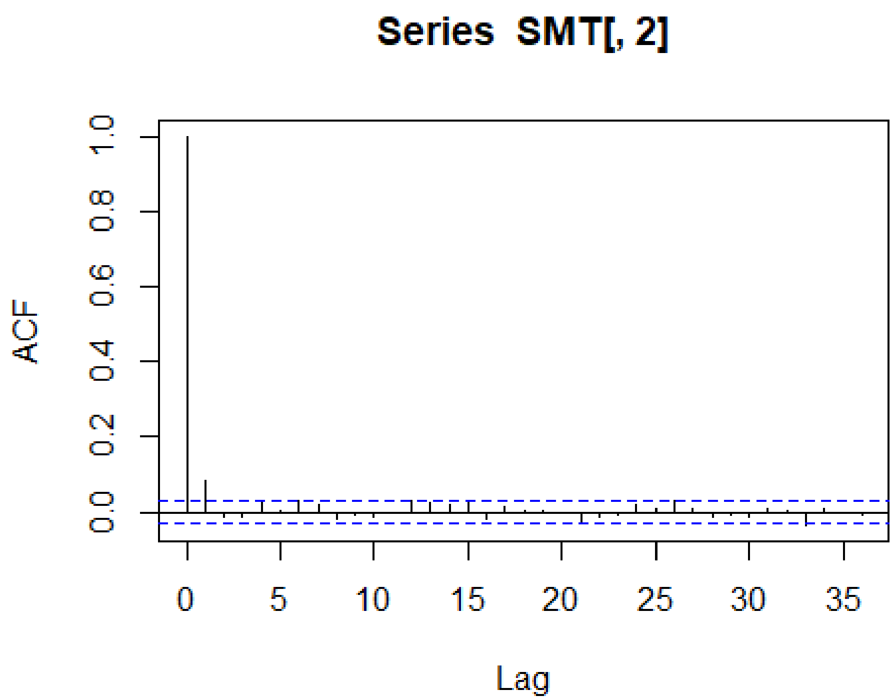
```r
### MCMC algorithm
for(s in 1:S){
  # sample new values of the thetas
  for(j in 1:m){
    vtheta = 1/(n[j]/sigma2 + 1/tau2)
    etheta = vtheta*(ybar[j]*n[j]/sigma2 + mu/tau2)
    theta[j] = rnorm(1, etheta, sqrt(vtheta))
  }
  # sample new value of sigma2
  nun = nu0 + sum(n)
  ss = nu0*s20
  for(j in 1:m){
    ss = ss + sum((Y[Y[, 1] == j, 2] - theta[j])^2)
  }
  sigma2 = 1/rgamma(1, nun/2, ss/2)
  # sample a new value of mu
  vmu = 1/(m/tau2 + 1/g20)
  emu = vmu*(m*mean(theta)/tau2 + mu0/g20)
  mu = rnorm(1, emu, sqrt(vmu))
  # sample a new value of tau2
  etam = eta0 + m
  ss = eta0*t20 + sum((theta - mu)^2)
  tau2 = 1/rgamma(1, etam/2, ss/2)
  # store results
  THETA[s, ] = theta
  SMT[s, ] = c(sigma2, mu, tau2)
}
###

### Assess the convergence of the Markov chain
acf(THETA[,1])
```



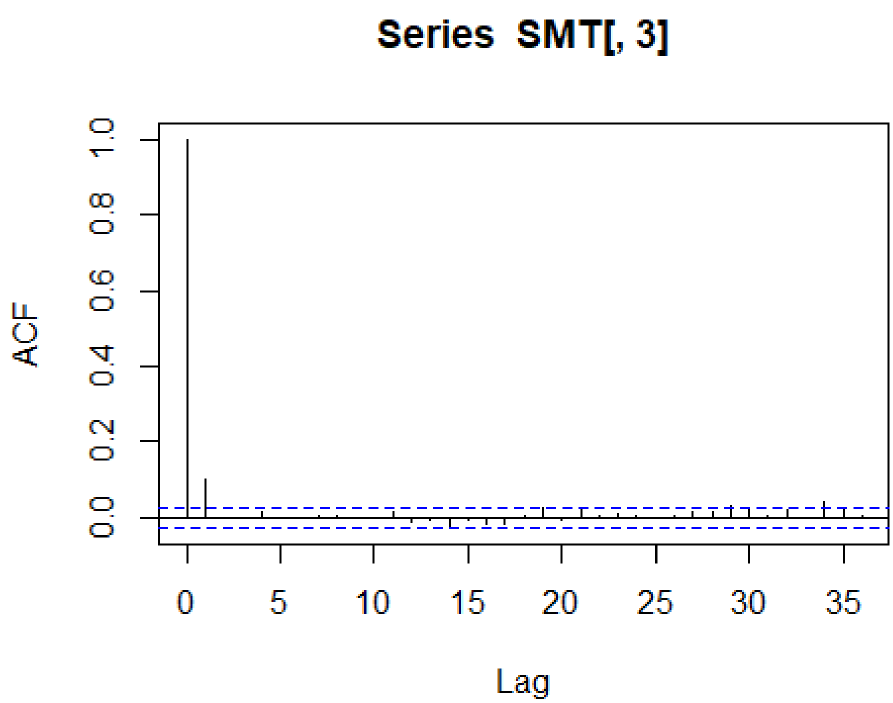

```
acf(SMT[,1])
```




```
acf(SMT[,2])
```




```
acf(SMT[,3])
```

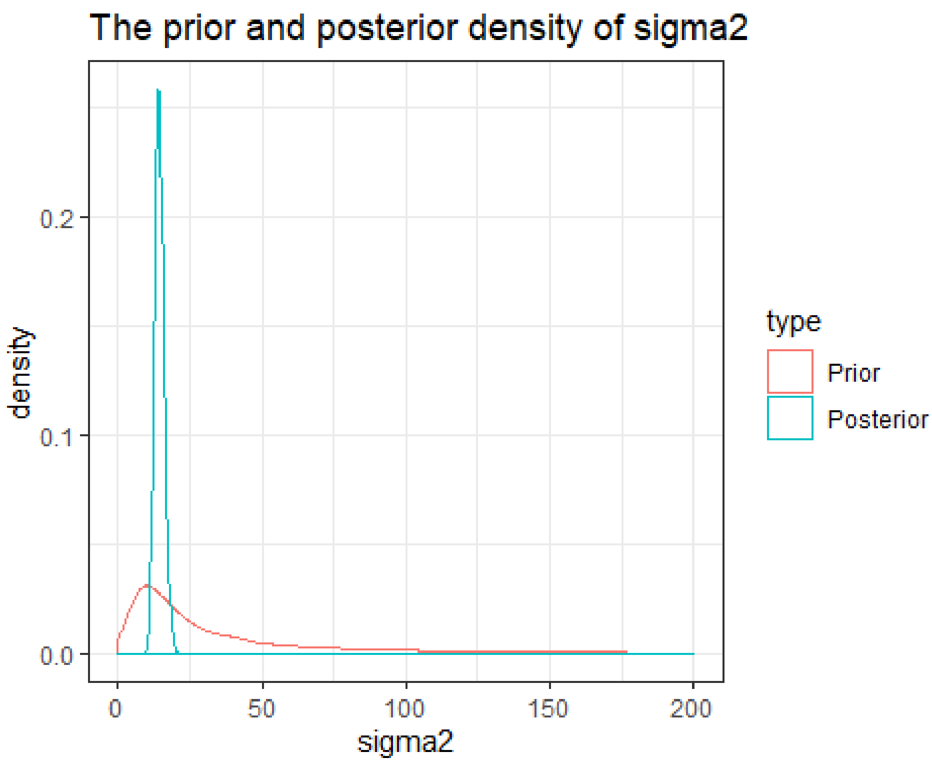


According to the acf plot, we see that the Markov chain converged quickly.

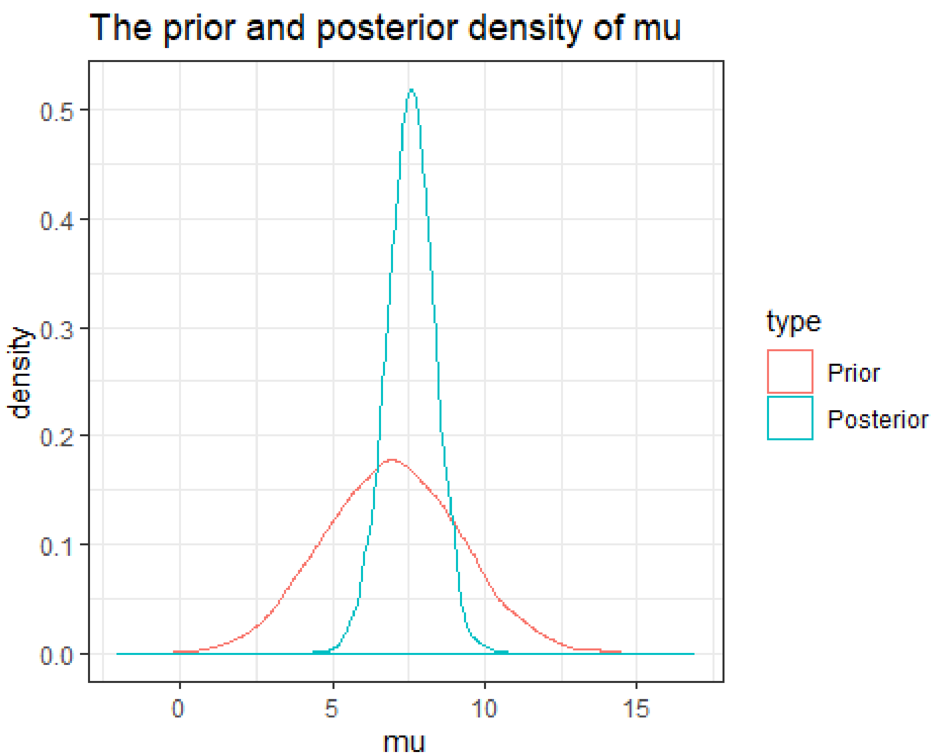
```
### Compute the effective sample size
effectiveSize(SMT[,1])

##     var1
## 4441.634

effectiveSize(SMT[,2])

##     var1
## 3644.017

effectiveSize(SMT[,3])

##     var1
## 4098.949
```

Part (b)

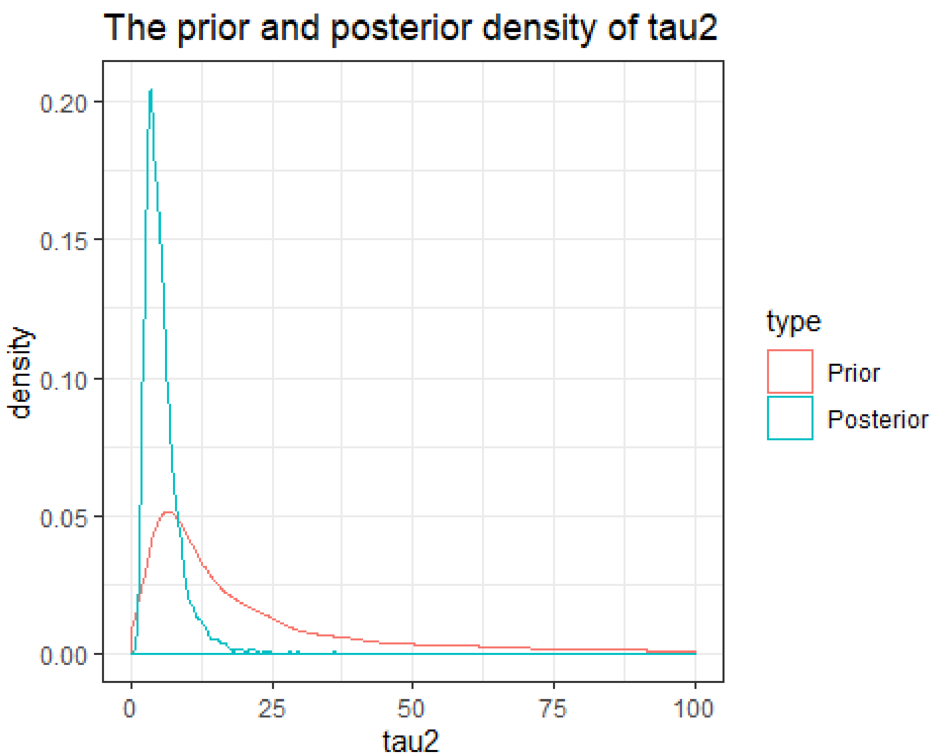
```
### Compute mean and CI
df = data.frame(rbind(Mean = apply(SMT, 2, FUN = mean),
                      apply(SMT, 2, FUN = quantile, probs=c(0.025, 0.97
5))))
names(df) = c('sigma2', 'mu', 'tau2')
df

##         sigma2       mu      tau2
## Mean  14.48008 7.560600  5.473300
## 2.5%  11.76444 5.976180  1.918859
## 97.5% 17.88119 9.060375 13.980738

### Compare Posterior and Prior density
## sigma2
sigma2.prior = data.frame(sigma2 = 1/rgamma(S, nu0/2, nu0*s20/2),
                          type = 'Prior')
sigma2.post = data.frame(sigma2 = SMT[,1],
                         type = 'Posterior')
sigma2.cp = rbind(sigma2.prior, sigma2.post)
ggplot(data = sigma2.cp, aes(x = sigma2,color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of sigma2') +
  xlim(0,200)
```




```
## mu
mu.prior = data.frame(mu = rnorm(S, mu0, sqrt(g20)),
                      type = 'Prior')
mu.post = data.frame(mu = SMT[,2],
                     type = 'Posterior')
mu.cp = rbind(mu.prior, mu.post)
ggplot(data = mu.cp, aes(x = mu ,color=type)) +
  labs(title = 'The prior and posterior density of mu') +
  geom_density()
```



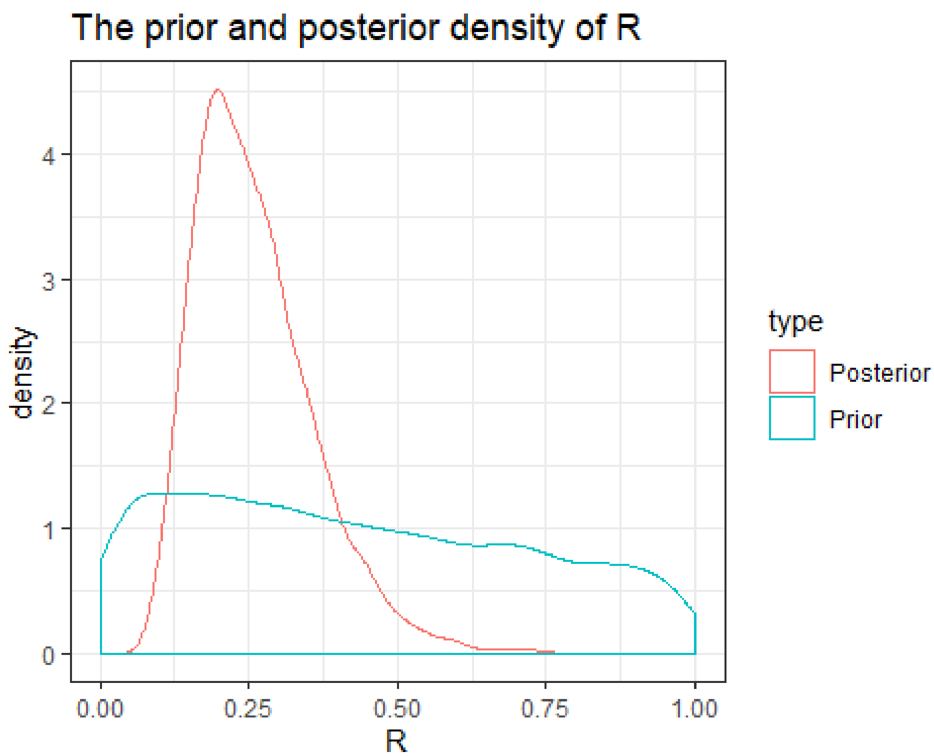

```r
## tau2
tau2.prior = data.frame(tau2 = 1/rgamma(S, eta0/2,  eta0*t20/2),
                        type = 'Prior')
tau2.post = data.frame(tau2 = SMT[,3],
                       type = 'Posterior')
tau2.cp = rbind(tau2.prior, tau2.post)
ggplot(data = tau2.cp, aes(x = tau2,color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of tau2') +
  xlim(0,100)
```



What we learn from the data is that we have poor prior for σ^2 and τ^2, but somewhat proper prior for μ.

Part (c)

```
sigma2.prior.value = 1/rgamma(S, nu0/2, nu0*s20/2)
tau2.prior.value = 1/rgamma(S, eta0/2, eta0*t20/2)
R.prior = tau2.prior.value/(tau2.prior.value + sigma2.prior.value)
R.post = SMT[,3]/(SMT[,1] + SMT[,3])
R.df = rbind(data.frame(R = R.post, type = 'Posterior'),
             data.frame(R = R.prior, type = 'Prior'))
ggplot(R.df, aes(x = R, color = type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of R')
```




```
mean(R.prior)

## [1] 0.431837

mean(R.post)

## [1] 0.2594916
```

Between school variance is τ^2, total variance is $\tau^2 + \sigma^2$. R is the proportion of the between school variance over total variance. The posterior value of R is 0.438, indicating that 43.8.7% of the total variation comes from the between school variation.

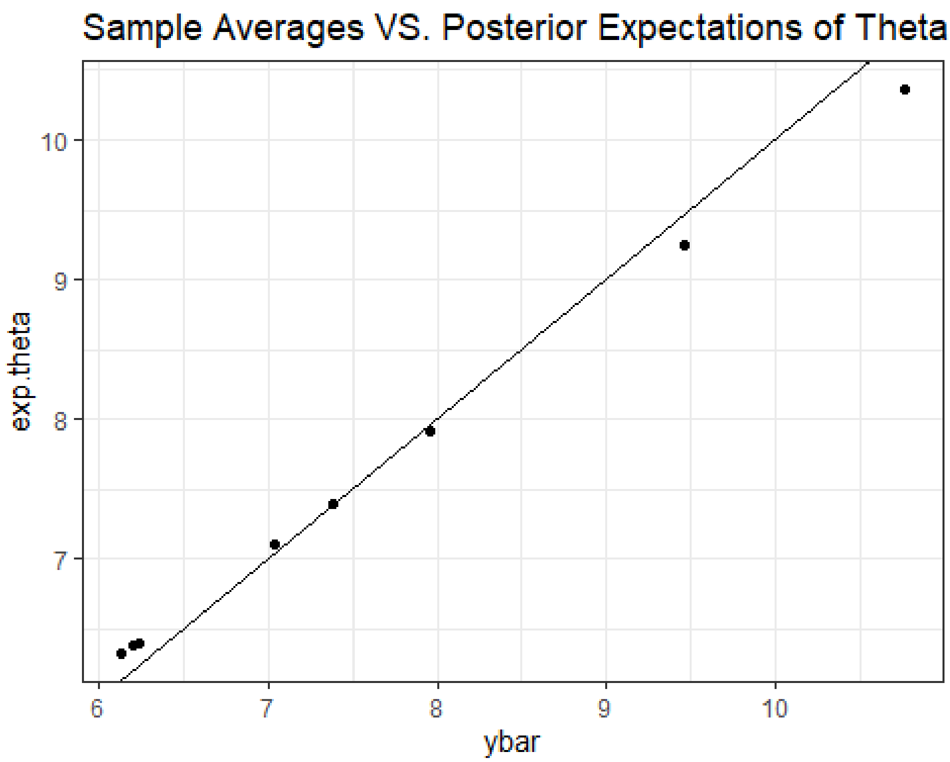
Part (d)

```
theta.7 = THETA[,7]
theta.6 = THETA[,6]
mean(theta.7 < theta.6)

## [1] 0.5124

min.theta = apply(THETA,1,min)
mean(theta.7 == min.theta)

## [1] 0.3178
```

The posterior probability that θ_7 is smaller than θ_6 is 0.5124. The posterior probability that θ_7 is the smallest of all the θ's is 0.3178.

Part (e)

```
ybarVStheta = data.frame(ybar = ybar,
                         exp.theta = apply(THETA, 2, mean))
ggplot(ybarVStheta, aes(x = ybar,y = exp.theta)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0) +
  labs(title = 'Sample Averages VS. Posterior Expectations of Theta')
```



From the plot we can see that the sample averages and posterior expectations are roughly equal.

```
mean(Y[,2])

## [1] 7.691278

mean(SMT[,2])

## [1] 7.5606
```

From the plot we can see that the sample mean of all observations and posterior mean μ are roughly equal.

Exercise 2

Hoff 9.1

Part (a)

$$\begin{aligned}
\beta &\sim \text{mvn}(\beta_0, \Sigma_0) \\
\beta \mid X, y, \sigma^2 &\sim \text{mvn}(E(\beta \mid X, y, \sigma^2), Var(\beta \mid X, y, \sigma^2)) \\
\text{where} \quad & Var(\beta \mid X, y, \sigma^2) = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1}, \\
\text{and} \quad & E(\beta \mid X, y, \sigma^2) = (\Sigma_0^{-1} + X^T X/\sigma^2)^{-1}(\Sigma_0^{-1}\beta_0 + X^T y/\sigma^2)
\end{aligned}$$

$$\begin{aligned}
\sigma^2 &\sim \text{inverse} - \text{gamma}(v_0/2, v_0\sigma_0^2/2) \\
\sigma^2 \mid X, y, \beta &\sim \text{inverse} - \text{gamma}(v_1/2, v_1\sigma_1^2/2) \\
\text{where} \quad & v_1 = v_0 + n, \\
\text{and} \quad & v_1\sigma_1^2 = v_0\sigma_0^2 + SSR(\beta)
\end{aligned}$$

Gibbs scheme: 1. updating β: a) compute $V = Var(\beta \mid X, y, \sigma^2)$ and $m = E(\beta \mid X, y, \sigma^2)$; b) sample $\beta^{(s+1)} \sim \text{mvn}(m, V)$ 2. updating σ^2: a) compute $SSR(\beta^{(s+1)})$ b) sample $\sigma^{2(s+1)} \mid X, y, \beta \sim \text{inverse} - \text{gamma}((v_0 + n)/2, [v_0\sigma_0^2 + SSR(\beta)]/2)$

```
swim =  read.table(url("http://www2.stat.duke.edu/~pdh10/FCBS/Exercises
/swim.dat"))

Gibbs.swim = function(y){

  X = cbind(rep(1, 6), c(1, 3, 5, 7, 9, 11))
  S = 5000
  BETA = matrix(nrow = S, ncol = 2)
  SIGMA2 = c()

  # prior
  S0 = rbind(c(0.25, 0), c(0, 0.1))
  beta0 = c(23, 0)
  v0 = 1
  s20 = 0.25

  # starting values
  beta = beta0
  sigma2 = 0.49

  for(s in 1:S){
    # update beta
    V = solve(solve(S0) + t(X) %*% X/sigma2)
    m = V %*% (solve(S0) %*% beta0 + (t(X) %*% y)/sigma2)
    beta = mvrnorm(1, m, V)
    # update sigma2
    SSR = t(y) %*% y - 2*t(beta) %*% t(X) %*% y + t(beta) %*% t(X) %*%
X %*% beta
    sigma2 = 1/rgamma(1, (v0+n)/2, (v0*s20 + SSR)/2)
    # store values
    BETA[s,] = beta
    SIGMA2[s] = sigma2
```

```
  }

  X.pred = c(1, 13)
  y.pred = rnorm(S, BETA %*% X.pred, sqrt(SIGMA2))
  y.pred
}
swimmer.1 = Gibbs.swim(as.numeric(swim[1,]))
swimmer.2 = Gibbs.swim(as.numeric(swim[2,]))
swimmer.3 = Gibbs.swim(as.numeric(swim[3,]))
swimmer.4 = Gibbs.swim(as.numeric(swim[4,]))
```

Part (b)

```
swim.df = cbind(swimmer.1, swimmer.2, swimmer.3, swimmer.4)
swim.min = apply(swim.df, 1, min)
mean(swimmer.1 == swim.min)

## [1] 0.9026

mean(swimmer.2 == swim.min)

## [1] 0

mean(swimmer.3 == swim.min)

## [1] 0.0974

mean(swimmer.4 == swim.min)

## [1] 0
```

According to the result, we can see that the probability of first swimmer being the fastest one is 90.14%, meaning that we should recommend the first swimmer.

Exercise 3

$$
\begin{aligned}
Y_{ijk} &\sim N(\theta_{ij}, \sigma^2) \\
\theta_{ij} &= \mu + a_i + b_j + (ab)_{ij} \\
&priors: \\
\mu &\sim N(0, \sigma_\mu^2) \\
a_1, \ldots, a_{m_1} &\sim N(0, \sigma_a^2) \\
b_1, \ldots, b_{m_2} &\sim N(0, \sigma_b^2) \\
(ab)_{11}, \ldots, (ab)_{m_1 m_2} &\sim N(0, \sigma_{ab}^2)
\end{aligned}
$$

$$
\begin{aligned}
\text{Let } \sigma^2 &\sim \text{inverse} - \text{gamma}(v_0/2, v_0\sigma_0^2/2) \\
\sigma_a^2 &\sim \text{inverse} - \text{gamma}(\eta_{a0}/2, \eta_{a0}\tau_{a0}^2/2) \\
\sigma_b^2 &\sim \text{inverse} - \text{gamma}(\eta_{b0}/2, \eta_{b0}\tau_{b0}^2/2) \\
\sigma_{ab}^2 &\sim \text{inverse} - \text{gamma}(\eta_{ab0}/2, \eta_{ab0}\tau_{ab0}^2/2) \\
\sigma_\mu^2 &\sim \text{inverse} - \text{gamma}(\eta_{\mu 0}/2, \eta_{\mu 0}\tau_{\mu 0}^2/2)
\end{aligned}
$$

Full conditional distribution for mu

$$
\begin{aligned}
\mu &\sim N(0, \sigma_\mu^2) \\
p(Y \mid \mu, \sigma^2, a, b, (ab)) &\underset{\mu}{\propto} exp(-1/2\sigma^2 \Sigma_{ijk}(y_{ijk} - \theta_{ij})^2) \\
&\underset{\mu}{\propto} exp(-1/2\sigma^2 \Sigma_{ijk}(y_{ijk} - (\mu + a_i + b_j + (ab)_{ij})^2) \\
&\underset{\mu}{\propto} exp(-1/2\sigma^2 \Sigma_{ijk}(r_{ijk}^\mu - \mu)^2) \\
&\underset{\mu}{\propto} exp(-1/2\sigma^2 (n m_1 m_2 \mu^2 - 2\mu \Sigma_{ijk} r_{ijk}^\mu)) \\
\text{where } \ r_{ijk}^\mu &= y_{ijk} - (a_i + b_j + (ab)_{ij})
\end{aligned}
$$

$$
\begin{aligned}
p(\mu \mid Y, \sigma^2, \sigma_\mu^2, a, b, (ab)) &\propto p(Y \mid \mu, \sigma^2, a, b, (ab)) p(\mu) \\
&\underset{\mu}{\propto} exp(-\frac{1}{2\sigma^2}(n m_1 m_2 \mu^2 - 2\mu \Sigma_{ijk} r_{ijk}^\mu)) * exp(-\frac{1}{2\sigma_\mu^2}\mu^2) \\
&\underset{\mu}{\propto} exp\{-\frac{1}{2}[(\frac{n m_1 m_2}{\sigma^2} + \frac{1}{\sigma_\mu^2})\mu^2 - 2\frac{\Sigma_{ijk} r_{ijk}^\mu}{\sigma^2}\mu]\} \\
&= dnorm(\mu_1, \sigma_{\mu 1}^2) \\
\text{where } \ \sigma_{\mu 1}^2 &= (\frac{n m_1 m_2}{\sigma^2} + \frac{1}{\sigma_\mu^2})^{-1}, \ \mu_1 = \sigma_{\mu 1}^2 \frac{\Sigma_{ijk} r_{ijk}^\mu}{\sigma^2}, \\
\text{with } \ r_{ijk}^\mu &= y_{ijk} - (a_i + b_j + (ab)_{ij})
\end{aligned}
$$

Full conditional distribution for sigma^2

$$\begin{aligned}
\sigma^2 &\sim \text{inverse}-\text{gamma}(v_0/2, v_0\sigma_0^2/2) \\
p(Y \mid \mu, \sigma^2, a, b, (ab)) &\underset{\sigma^2}{\propto} (\sigma^2)^{-nm_1m_2/2} exp(-1/2\sigma^2 \Sigma(y_{ijk} - \theta_{ij})^2)
\end{aligned}$$

$$\begin{aligned}
p(\sigma^2 \mid Y, \mu, v_0, \sigma_0^2, a, b, (ab)) &\propto p(Y \mid \mu, \sigma^2, a, b, (ab))p(\sigma^2) \\
&\underset{\sigma^2}{\propto} \{(\sigma^2)^{-nm_1m_2/2} exp(-\frac{\Sigma_{ijk}(y_{ijk} - \theta_{ij})^2}{2\sigma^2})\} * \{(\sigma^2)^{-v_0/2-1} exp(-\frac{v_0\sigma_0^2}{2\sigma^2})\} \\
&\underset{\sigma^2}{\propto} (\sigma^2)^{-(v_0+nm_1m_2)/2-1} exp(-\frac{v_0\sigma_0^2 + \Sigma_{ijk}(y_{ijk} - \theta_{ij})^2}{2\sigma^2}) \\
&= \text{inverse}-\text{gamma}(\frac{v_1}{2}, \frac{v_1\sigma_1^2}{2}) \\
\text{where } v_1 &= v_0 + nm_1m_2,\ v_1\sigma_1^2 = v_0\sigma_0^2 + \Sigma_{ijk}(y_{ijk} - \theta_{ij})^2
\end{aligned}$$

Full conditional distribution for a_i

$$\begin{aligned}
a_i &\sim N(0, \sigma_a^2) \\
p(Y \mid \mu, \sigma^2, a, b, (ab)) &\underset{a_i}{\propto} exp(-1/2\sigma^2 \Sigma_{jk}(y_{ijk} - \theta_{ij})^2) \\
&\underset{a_i}{\propto} exp(-1/2\sigma^2 \Sigma_{jk}(y_{ijk} - (\mu + a_i + b_j + (ab)_{ij})^2) \\
&\underset{a_i}{\propto} exp(-1/2\sigma^2 \Sigma_{jk}(r_{ijk}^a - a_i)^2) \\
&\underset{a_i}{\propto} exp(-1/2\sigma^2 (nm_2 a_i^2 - 2a_i \Sigma_{jk} r_{ijk}^a)) \\
\text{where } r_{ijk}^a &= y_{ijk} - (\mu + b_j + (ab)_{ij})
\end{aligned}$$

$$\begin{aligned}
p(a_i \mid Y, \sigma^2, \sigma_a^2, a, b, (ab)) &\propto p(Y \mid \mu, \sigma^2, a, b, (ab))p(a_i) \\
&\underset{a_i}{\propto} exp(-\frac{1}{2\sigma^2}(nm_2 a_i^2 - 2a_i \Sigma_{jk} r_{ijk}^a)) * exp(-\frac{1}{2\sigma_a^2} a_i^2) \\
&\underset{a_i}{\propto} exp\{-\frac{1}{2}[(\frac{nm_2}{\sigma^2} + \frac{1}{\sigma_a^2})a_i^2 - 2\frac{\Sigma_{jk} r_{ijk}^a}{\sigma^2} a_i]\} \\
&= dnorm(\mu_a, \sigma_{a1}^2) \\
\text{where } \sigma_{a1}^2 &= (\frac{nm_2}{\sigma^2} + \frac{1}{\sigma_a^2})^{-1},\ \mu_a = \sigma_{a1}^2 \frac{\Sigma_{jk} r_{ijk}^a}{\sigma^2}, \\
\text{with } r_{ijk}^a &= y_{ijk} - (\mu + b_j + (ab)_{ij})
\end{aligned}$$

Full conditional distribution for b_j

$$
\begin{aligned}
b_j &\sim N(0, \sigma_b^2) \\
p(Y \mid \mu, \sigma^2, a, b, (ab)) &\underset{b_j}{\propto} exp(-1/2\sigma^2 \Sigma_{ik}(y_{ijk} - \theta_{ij})^2) \\
&\underset{b_j}{\propto} exp(-1/2\sigma^2 \Sigma_{ik}(y_{ijk} - (\mu + a_i + b_j + (ab)_{ij})^2) \\
&\underset{b_j}{\propto} exp(-1/2\sigma^2 \Sigma_{ik}(r_{ijk}^b - b_j)^2) \\
&\underset{b_j}{\propto} exp(-1/2\sigma^2 (nm_1 b_j^2 - 2b_j \Sigma_{ik} r_{ijk}^b)) \\
\text{where} \quad r_{ijk}^b &= y_{ijk} - (\mu + a_i + (ab)_{ij})
\end{aligned}
$$

$$
\begin{aligned}
p(b_j \mid Y, \sigma^2, \sigma_b^2, a, b, (ab)) &\propto p(Y \mid \mu, \sigma^2, a, b, (ab)) p(b_j) \\
&\underset{b_j}{\propto} exp(-\frac{1}{2\sigma^2}(nm_1 b_j^2 - 2b_j \Sigma_{ik} r_{ijk}^b)) * exp(-\frac{1}{2\sigma_b^2} b_j^2) \\
&\underset{b_j}{\propto} exp\{-\frac{1}{2}[(\frac{nm_1}{\sigma^2} + \frac{1}{\sigma_b^2}) b_j^2 - 2\frac{\Sigma_{ik} r_{ijk}^b}{\sigma^2} b_j]\} \\
&= dnorm(\mu_b, \sigma_{b1}^2) \\
\text{where} \quad \sigma_{b1}^2 &= (\frac{nm_1}{\sigma^2} + \frac{1}{\sigma_b^2})^{-1}, \; \mu_b = \sigma_{b1}^2 \frac{\Sigma_{ik} r_{ijk}^b}{\sigma^2}, \\
\text{with} \quad r_{ijk}^b &= y_{ijk} - (\mu + a_i + (ab)_{ij})
\end{aligned}
$$

Full conditional distribution for (ab)_ij

$$
\begin{aligned}
(ab)_{ij} &\sim N(0, \sigma^2_{ab}) \\
p(Y \mid \mu, \sigma^2, a, b, (ab)) &\underset{(ab)_{ij}}{\propto} exp(-1/2\sigma^2 \Sigma_k (y_{ijk} - \theta_{ij})^2) \\
&\underset{(ab)_{ij}}{\propto} exp(-1/2\sigma^2 \Sigma_k (y_{ijk} - (\mu + a_i + b_j + (ab)_{ij})^2) \\
&\underset{(ab)_{ij}}{\propto} exp(-1/2\sigma^2 \Sigma_k (r^{(ab)}_{ijk} - (ab)_{ij})^2) \\
&\underset{(ab)_{ij}}{\propto} exp(-1/2\sigma^2 (n(ab)^2_{ij} - 2(ab)_{ij}\Sigma_k r^{(ab)}_{ijk})) \\
\text{where} \quad r^{(ab)}_{ijk} &= y_{ijk} - (\mu + a_i + b_j)
\end{aligned}
$$

$$
\begin{aligned}
p((ab)_{ij} \mid Y, \sigma^2, \sigma^2_a, a, b, (ab)) &\propto p(Y \mid \mu, \sigma^2, a, b, (ab)) p((ab)_{ij}) \\
&\underset{(ab)_{ij}}{\propto} exp(-\frac{1}{2\sigma^2}(n(ab)^2_{ij} - 2(ab)_{ij}\Sigma_k r^{(ab)}_{ijk})) \\
&\quad * exp(-\frac{1}{2\sigma^2_{ab}}(ab)^2_{ij}) \\
&\underset{(ab)_{ij}}{\propto} exp\{-\frac{1}{2}[(\frac{n}{\sigma^2} + \frac{1}{\sigma^2_{ab}})(ab)^2_{ij} - 2\frac{\Sigma_k r^{(ab)}_{ijk}}{\sigma^2}(ab)_{ij}]\} \\
&= dnorm(\mu_{ab}, \sigma^2_{ab1}) \\
\text{where} \quad \sigma^2_{ab1} &= (\frac{n}{\sigma^2} + \frac{1}{\sigma^2_{ab}})^{-1}, \; \mu_{ab} = \sigma^2_{ab1}\frac{\Sigma_k r_{ijk}}{\sigma^2}, \\
\text{with} \quad r^{(ab)}_{ijk} &= y_{ijk} - (\mu + a_i + b_j)
\end{aligned}
$$

Full conditional distribution for sigma_a^2

$$
\begin{aligned}
\sigma^2_a &\sim \text{inverse} - \text{gamma}(\eta_{a0}/2, \eta_{a0}\tau^2_{a0}/2) \\
p(\sigma^2_a \mid a_1, \ldots, a_{m_1}) &\propto p(a_1, \ldots, a_{m_1} \mid \sigma^2_a) p(\sigma^2_a) \\
&\propto \prod_i d\,norm(0, \sigma^2_a) * p(\sigma^2_a) \\
&\propto \{(\sigma^2_a)^{-m_1/2} exp(-\frac{\Sigma_i a^2_i}{2\sigma^2_a})\} * \{(\sigma^2_a)^{-\eta_{a0}/2-1} exp(-\frac{\eta_{a0}\tau^2_{a0}}{2\sigma^2_a})\} \\
&\propto (\sigma^2_a)^{-(\eta_{a0}+m_1)/2-1} exp(-\frac{\eta_{a0}\tau^2_{a0} + \Sigma_i a^2_i}{2\sigma^2_a}) \\
&= \text{inverse} - \text{gamma}(\eta_{a1}/2, \eta_{a1}\tau^2_{a1}/2) \\
\text{where } \eta_{a1} &= \eta_{a0} + m_1, \; \eta_{a1}\tau^2_{a1} = \eta_{a0}\tau^2_{a0} + \Sigma_i a^2_i
\end{aligned}
$$

Full conditional distribution for sigma_b^2

$$
\begin{aligned}
\sigma_b^2 &\sim \text{inverse} - \text{gamma}(\eta_{b0}/2, \eta_{b0}\tau_{b0}^2/2) \\
p(\sigma_b^2 \mid b_1, \ldots, b_{m_2}) &\propto p(b_1, \ldots, b_{m_2} \mid \sigma_b^2) p(\sigma_b^2) \\
&\propto \prod_j d\, norm(0, \sigma_b^2) * p(\sigma_b^2) \\
&\propto \{(\sigma_b^2)^{-m_2/2} exp(-\frac{\Sigma_j b_j^2}{2\sigma_b^2})\} * \{(\sigma_b^2)^{-\eta_{b0}/2-1} exp(-\frac{\eta_{b0}\tau_{b0}^2}{2\sigma_b^2})\} \\
&\propto (\sigma_b^2)^{-(\eta_{b0}+m_2)/2-1} exp(-\frac{\eta_{b0}\tau_{b0}^2 + \Sigma_j b_j^2}{2\sigma_b^2}) \\
&= \text{inverse} - \text{gamma}(\eta_{b1}/2, \eta_{b1}\tau_{b1}^2/2) \\
\text{where } \eta_{b1} &= \eta_{b0} + m_2,\ \eta_{b1}\tau_{b1}^2 = \eta_{b0}\tau_{b0}^2 + \Sigma_j b_j^2
\end{aligned}
$$

Full conditional distribution for sigma_ab^2

$$
\begin{aligned}
\sigma_{ab}^2 &\sim \text{inverse} - \text{gamma}(\eta_{ab0}/2, \eta_{ab0}\tau_{ab0}^2/2) \\
p(\sigma_{ab}^2 \mid (ab)_{11}, \ldots, (ab)_{m_1 m_2}) &\propto p((ab)_{11}, \ldots, (ab)_{m_1 m_2} \mid \sigma_{ab}^2) p(\sigma_{ab}^2) \\
&\propto \prod_{ij} d\, norm(0, \sigma_{ab}^2) * p(\sigma_{ab}^2) \\
&\propto \{(\sigma_{ab}^2)^{-m_1 m_2/2} exp(-\frac{\Sigma_{ij}(ab)_{ij}^2}{2\sigma_{ab}^2})\} \\
&\quad * \{(\sigma_{ab}^2)^{-\eta_{ab0}/2-1} exp(-\frac{\eta_{ab0}\tau_{ab0}^2}{2\sigma_{ab}^2})\} \\
&\propto (\sigma_{ab}^2)^{-(\eta_{ab0}+m_1 m_2)/2-1} exp(-\frac{\eta_{ab0}\tau_{ab0}^2 + \Sigma_{ij}(ab)_{ij}^2}{2\sigma_{ab}^2}) \\
&= \text{inverse} - \text{gamma}(\eta_{ab1}/2, \eta_{ab1}\tau_{ab1}^2/2) \\
\text{where } \eta_{ab1} &= \eta_{ab0} + m_1 m_2,\ \eta_{ab1}\tau_{ab1}^2 = \eta_{ab0}\tau_{ab0}^2 + \Sigma_{ij}(ab)_{ij}^2
\end{aligned}
$$

Full conditional distribution for sigma_mu^2

$$
\begin{aligned}
\sigma_\mu^2 &\sim \text{inverse} - \text{gamma}(\eta_{\mu0}/2, \eta_{\mu0}\tau_{\mu0}^2/2) \\
p(\sigma_\mu^2 \mid \mu) &\propto p(\mu \mid \sigma_\mu^2)p(\sigma_\mu^2) \\
&\propto dnorm(0, \sigma_\mu^2) * p(\sigma_\mu^2) \\
&\propto \{(\sigma_\mu^2)^{-1/2} exp(-\frac{\mu^2}{2\sigma_\mu^2})\} * \{(\sigma_\mu^2)^{-\eta_{\mu0}/2-1} exp(-\frac{\eta_{\mu0}\tau_{\mu0}^2}{2\sigma_\mu^2})\} \\
&\propto (\sigma_\mu^2)^{-(\eta_{\mu0}+1)/2-1} exp(-\frac{\eta_{\mu0}\tau_{\mu0}^2 + \mu^2}{2\sigma_\mu^2}) \\
&= \text{inverse} - \text{gamma}(\eta_{\mu1}/2, \eta_{\mu1}\tau_{\mu1}^2/2) \\
\text{where } \eta_{\mu1} &= \eta_{\mu0} + 1,\ \eta_{\mu1}\tau_{\mu1}^2 = \eta_{\mu0}\tau_{\mu0}^2 + \mu^2
\end{aligned}
$$

Get data

```
data(ToothGrowth, package = 'datasets')
ToothGrowth = ToothGrowth %>%
  mutate(i.index=c(rep(1,30), rep(2,30))) %>%
  mutate(j.index = c(rep(1,10),rep(2,10),rep(3,10),rep(1,10),rep(2,10),
rep(3,10))) %>%
  mutate(k.index = c(rep(1:10,6)))
```

Gibbs sampling

```
S = 5000
m1 = 2
m2 = 3
n = 10

MU = c()
A = matrix(nrow = S, ncol = m1)
B = matrix(nrow = S, ncol = m2)
AB = matrix(nrow = S, ncol = m1*m2)
THETA = matrix(nrow = S, ncol = m1*m2)
SIGMA2 = c()
SIGMA2A = c()
SIGMA2B = c()
SIGMA2AB = c()
SIGMA2MU = c()

# starting values
mu = mean(ToothGrowth$len)
a=c(0,0)
b=c(0,0,0)
ab = matrix(0, nrow = 2, ncol = 3)
```

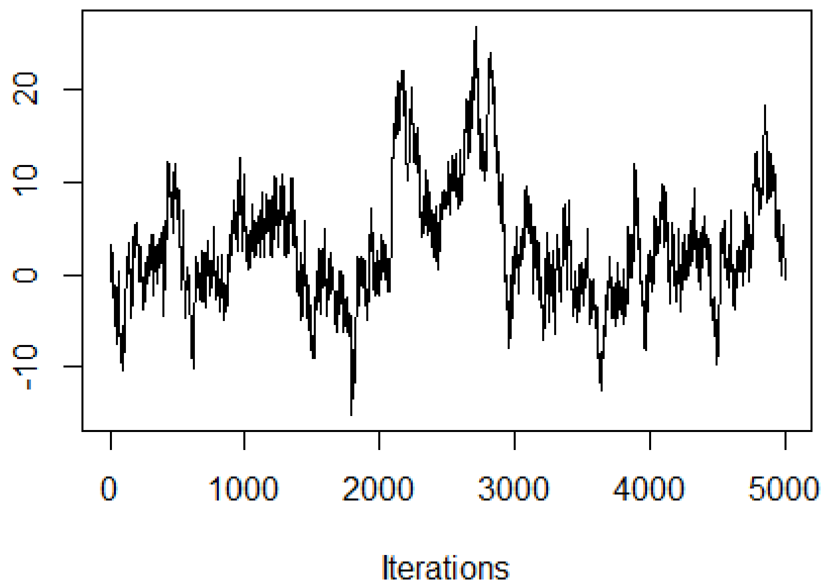
```
theta = matrix(0, nrow = 2, ncol = 3)

sigma2 = 50
sigma2.a = 50
sigma2.b = 50
sigma2.ab = 50
sigma2.mu = 50

v0 = 1; s20 = 50
eta0.a = 1; t0.a = 50
eta0.b = 1; t0.b = 50
eta0.ab = 1; t0.ab = 50
eta0.mu = 1; t0.mu = 50

for(s in 1:S){
  ### mu
  r.mu = 0
  for(i in 1:m1){
    for(j in 1:m2){
      for(k in 1:n){
        y = ToothGrowth$len[which(ToothGrowth$i.index == i & ToothGrowt
h$j.index == j & ToothGrowth$k.index == k)]
        r.mu = r.mu + y - a[i] - b[j] - ab[i,j]
      }
    }
  }
  vmu = 1/(n*m1*m2/sigma2 + 1/sigma2.mu)
  emu = vmu*r.mu/sigma2
  mu = rnorm(1, emu, sqrt(vmu))

  ### a
  va = 1/(n*m2/sigma2 + 1/sigma2.a)
  ea=c()
  r.a = c(0,0)
  for(i in 1:m1){
    for(j in 1:m2){
      for(k in 1:n){
        y = ToothGrowth$len[which(ToothGrowth$i.index == i & ToothGrowt
h$j.index == j & ToothGrowth$k.index == k)]
        r.a[i] = r.a[i] + y - mu - b[j] - ab[i,j]
      }
    }
    ea[i] = va*r.a[i]/sigma2
    a[i] = rnorm(1, ea[i], sqrt(va))
  }

  ### b
  vb = 1/(n*m1/sigma2 + 1/sigma2.b)
  eb = c(0,0,0)
```

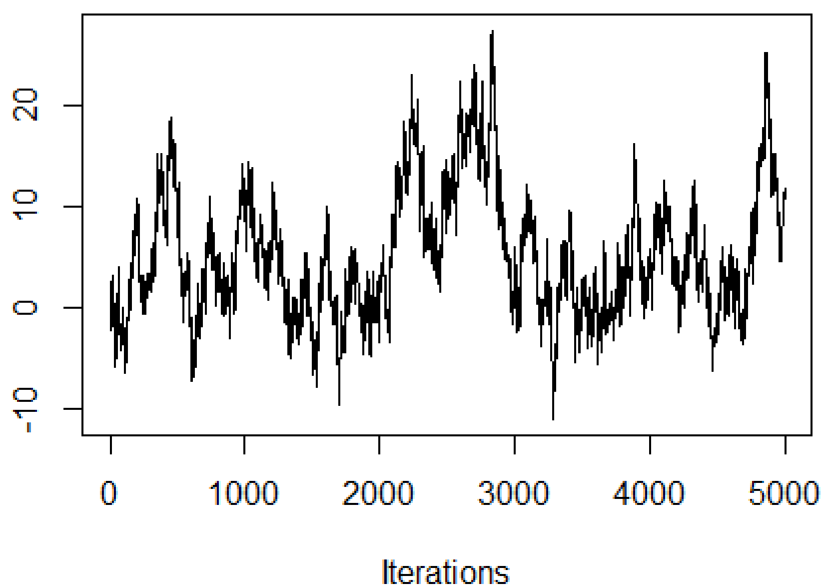
```r
  r.b = c(0,0,0)
  for(j in 1:m2){
    for(i in 1:m1){
      for(k in 1:n){
        y = ToothGrowth$len[which(ToothGrowth$i.index == i & ToothGrowt
h$j.index == j & ToothGrowth$k.index == k)]
        r.b[j] = r.b[j] + y - mu - a[i] - ab[i,j]
      }
    }
    eb[j] = vb*r.b[j]/sigma2
    b[j] = rnorm(1, eb[j], sqrt(vb))
  }

  ### ab
  vab = 1/(n/sigma2 + 1/sigma2.ab)
  eab = matrix(rep(0,6), nrow = 2, ncol = 3)
  r.ab = matrix(rep(0,6), nrow = 2, ncol = 3)
  for(i in 1:m1){
    for(j in 1:m2){
      for(k in 1:n){
        y = ToothGrowth$len[which(ToothGrowth$i.index == i & ToothGrowt
h$j.index == j & ToothGrowth$k.index == k)]
        r.ab[i,j] = r.ab[i,j] + y - mu - a[i] - b[j]
      }
      eab[i,j] = vab*r.ab[i,j]/sigma2
      ab[i,j] = rnorm(1, eab[i,j], sqrt(vab))
    }
  }

  ### sigma2
  s1 = 0
  for(i in 1:m1){
    for(j in 1:m2){
      for(k in 1:n){
        y = ToothGrowth$len[which(ToothGrowth$i.index == i & ToothGrowt
h$j.index == j & ToothGrowth$k.index == k)]
        r = y - mu - a[i] - b[j] - ab[i,j]
        s1 = s1 + r^2
      }
    }
  }
  v1 = v0 + n*m1*m2
  ss1 = v0*s20 + s1
  sigma2 = 1/rgamma(1, v1/2, ss1/2)

  ### sigma2.a
  eta1.a = eta0.a + m1
  ss.a = eta0.a*t0.a + sum((a)^2)
  sigma2.a = 1/rgamma(1, eta1.a/2, ss.a/2)
```

```
  ### sigma2.b
  eta1.b = eta0.b + m2
  ss.b = eta0.b*t0.b + sum((b)^2)
  sigma2.b = 1/rgamma(1, eta1.b/2, ss.b/2)

  ### sigma2.ab
  eta1.ab = eta0.ab + m1*m2
  ss.ab = eta0.ab*t0.ab + sum((ab)^2)
  sigma2.ab = 1/rgamma(1, eta1.ab/2, ss.ab/2)

  ### sigma2.mu
  eta1.mu = eta0.mu + 1
  ss.mu = eta0.mu*t0.mu + (mu)^2
  sigma2.mu = 1/rgamma(1, eta1.mu/2, ss.mu/2)

  ### theta
  for(i in 1:m1){
    for(j in 1:m2){
      theta[i,j] = mu + a[i] + b[j] + ab[i,j]
    }
  }

  MU[s] = mu
  A[s,] = a
  B[s,] = b
  AB[s,] = c(ab)
  THETA[s,] = c(theta)
  SIGMA2[s] = sigma2
  SIGMA2A[s] = sigma2.a
  SIGMA2B[s] = sigma2.b
  SIGMA2AB[s] = sigma2.ab
  SIGMA2MU[s] = sigma2.mu

}
```

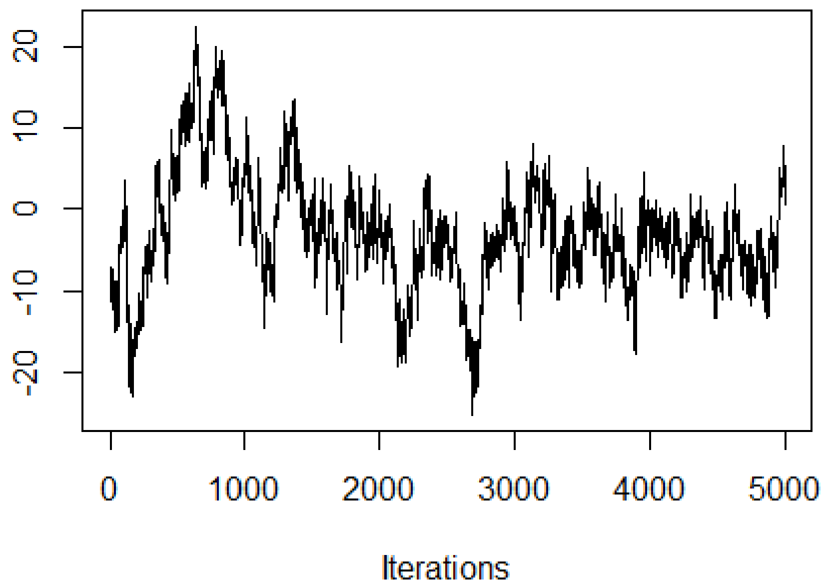
Convergence

```
traceplot(as.mcmc(A))
```

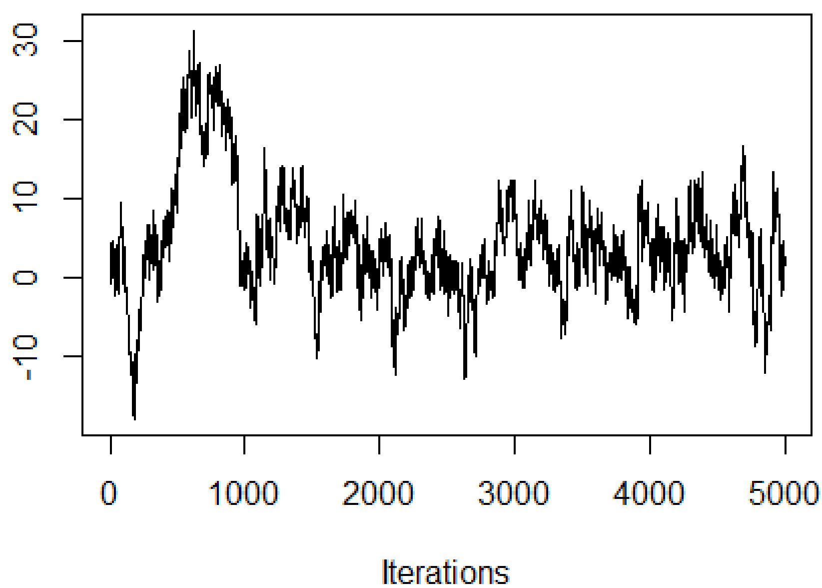




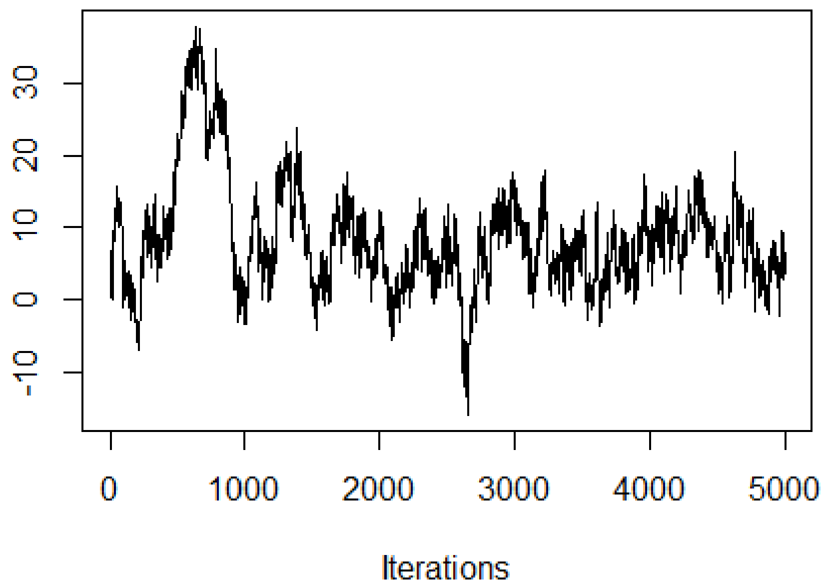

```
traceplot(as.mcmc(B))
```



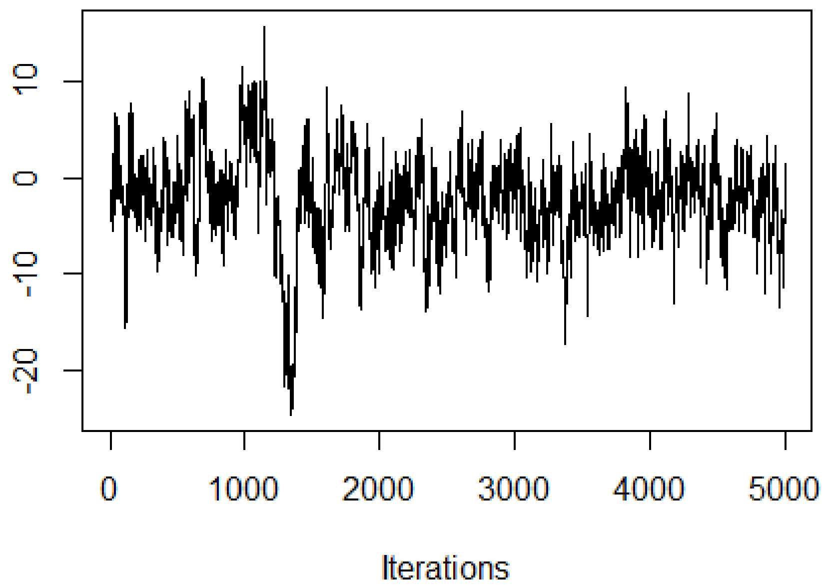
20
10
0
-10
-20
0
1000
2000
3000
4000
5000
Iterations

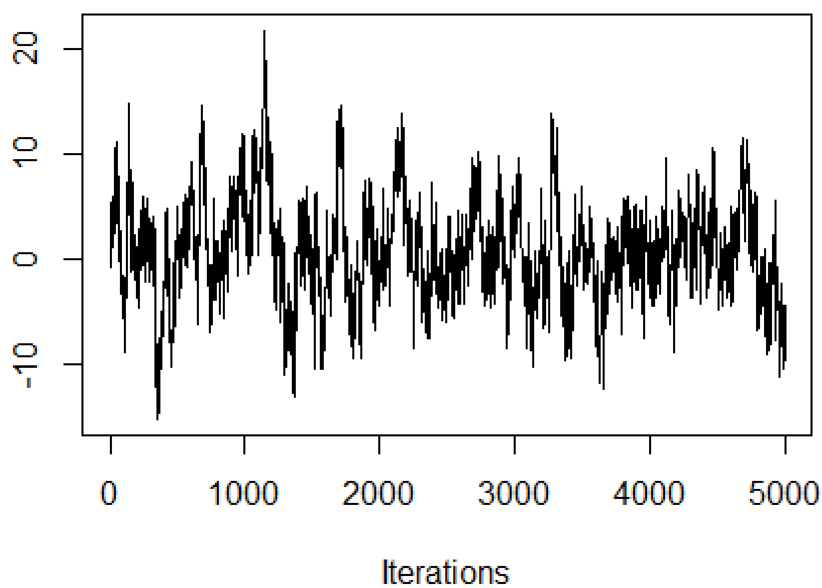


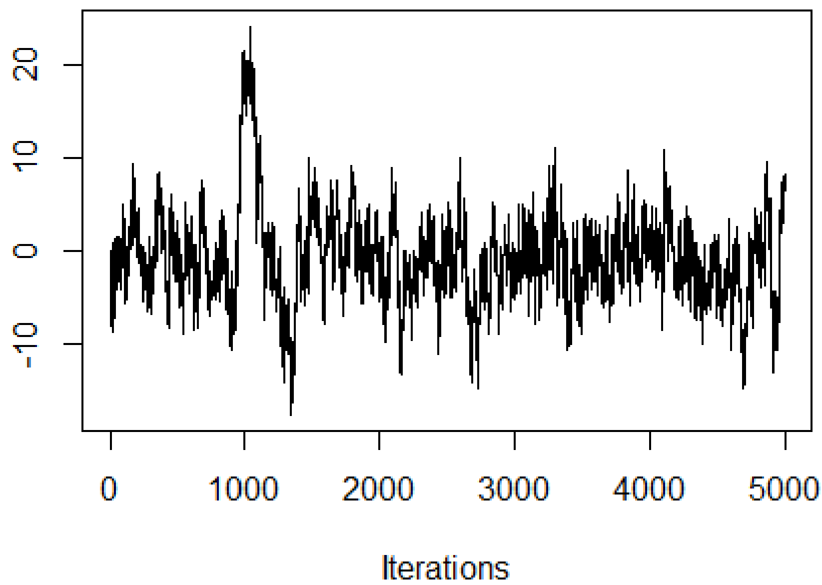
30
20
10
0
-10
0
1000
2000
3000
4000
5000
Iterations



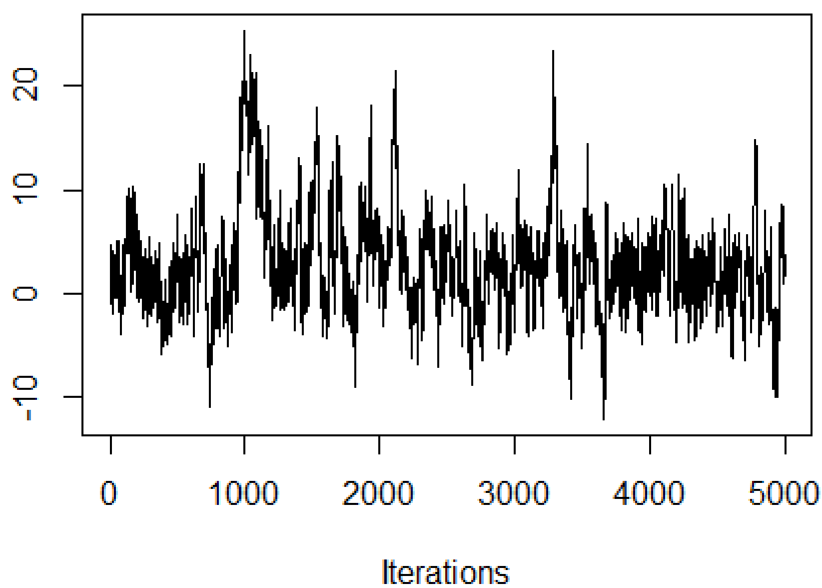

```
traceplot(as.mcmc(AB))
```


10
0
-10
-20
0
1000
2000
3000
4000
5000
Iterations

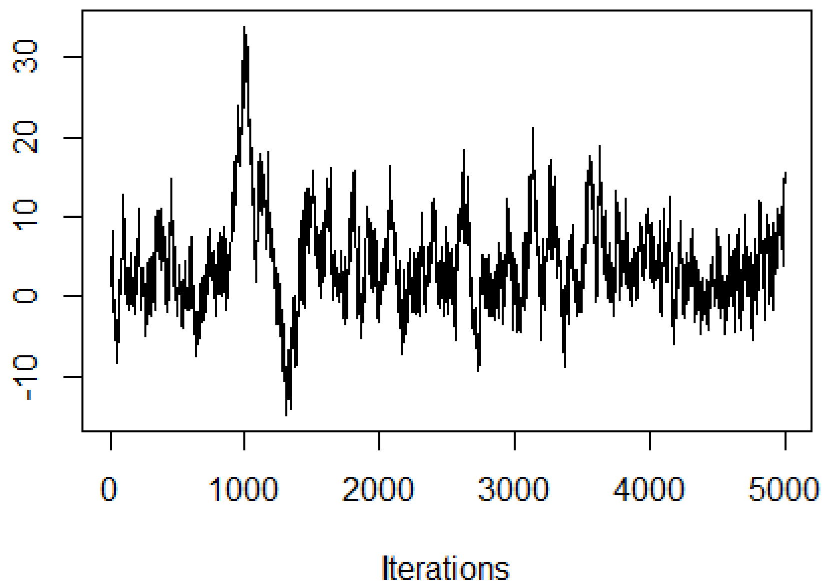

20
10
0
-10
0
1000
2000
3000
4000
5000
Iterations

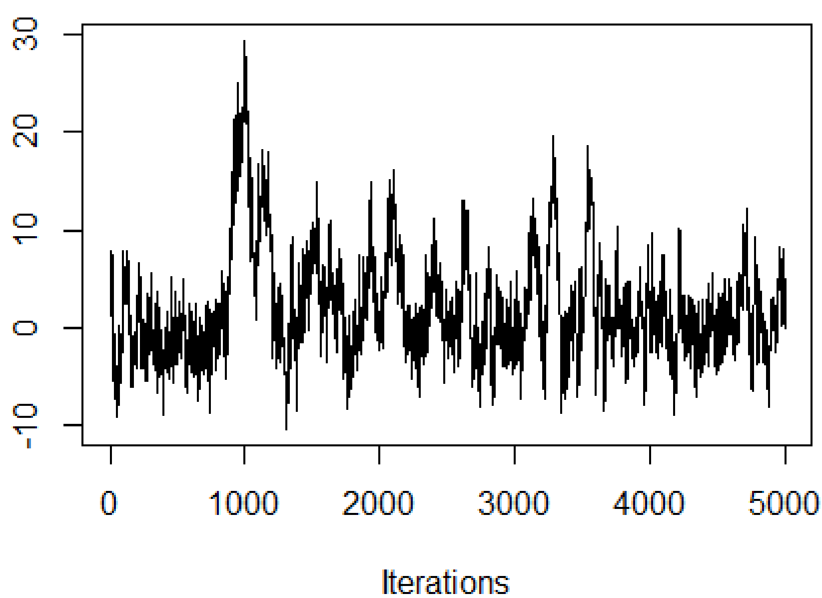


20
10
0
-10
0
1000
2000
3000
4000
5000
Iterations

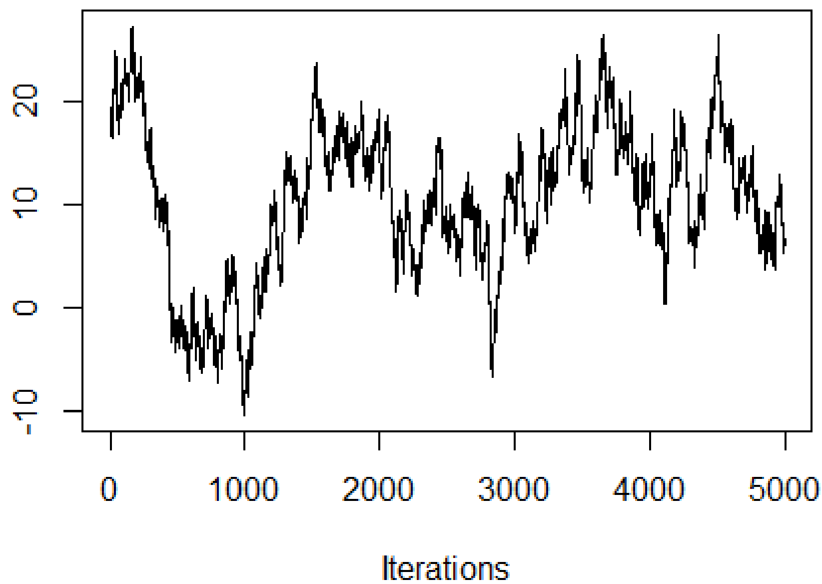


20
10
0
-10
0
1000
2000
3000
4000
5000
Iterations

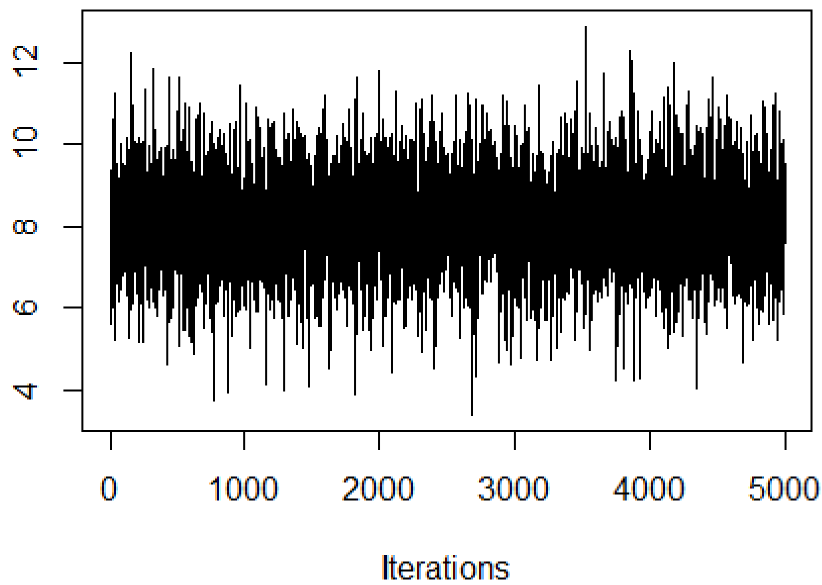




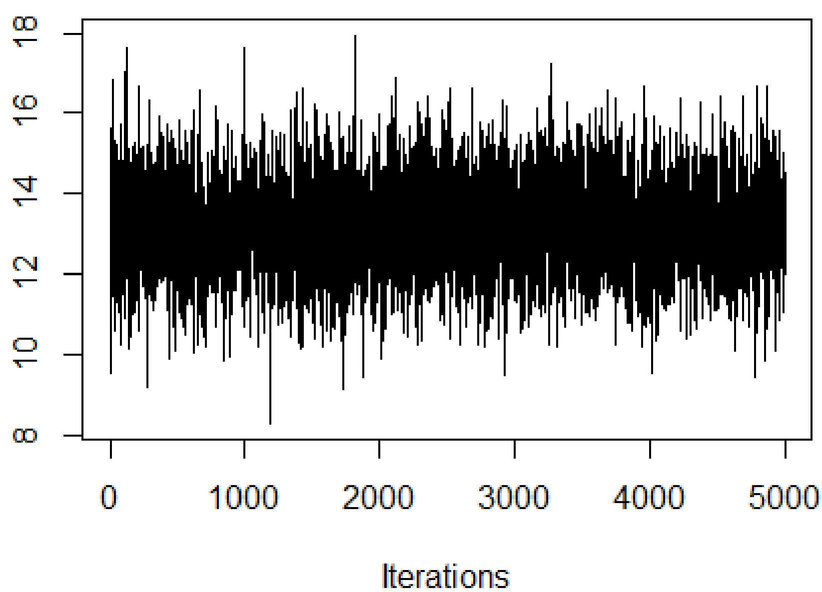

```
traceplot(as.mcmc(MU))
```



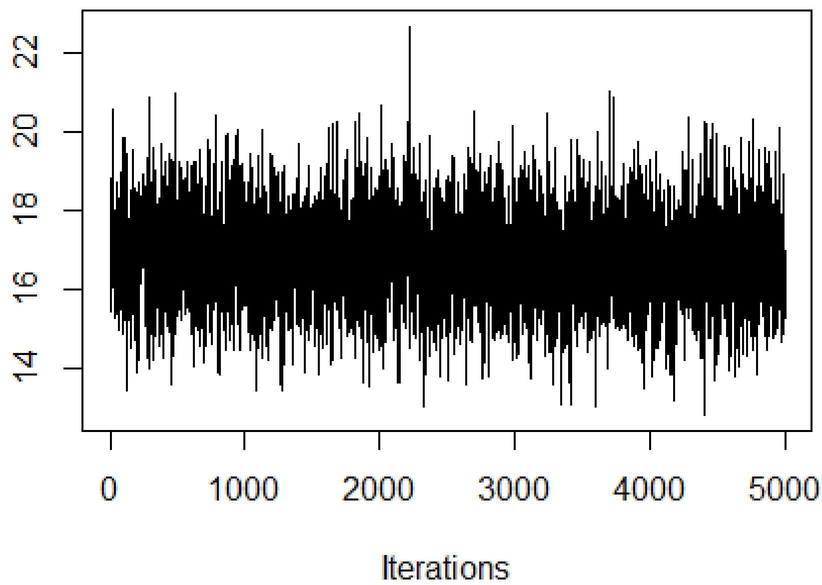

```
traceplot(as.mcmc(THETA))
```

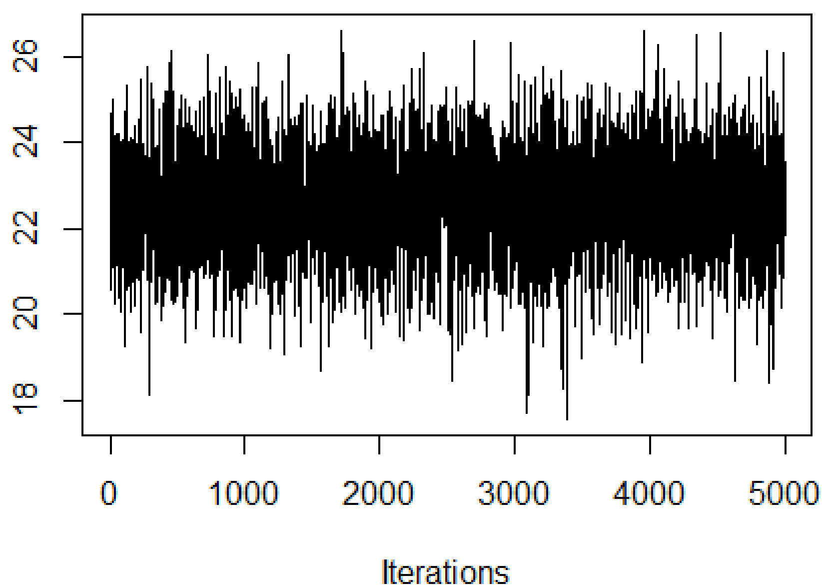


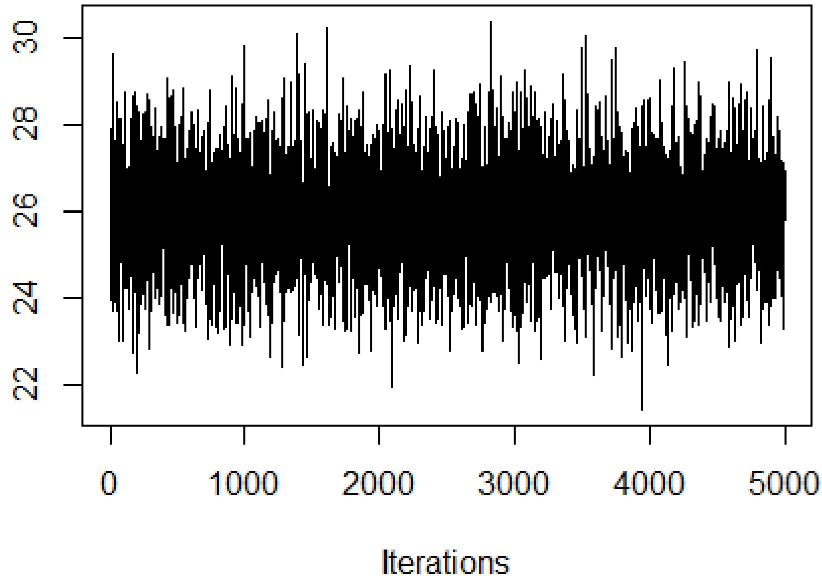
12
10
8
6
4
0
1000
2000
3000
4000
5000
Iterations

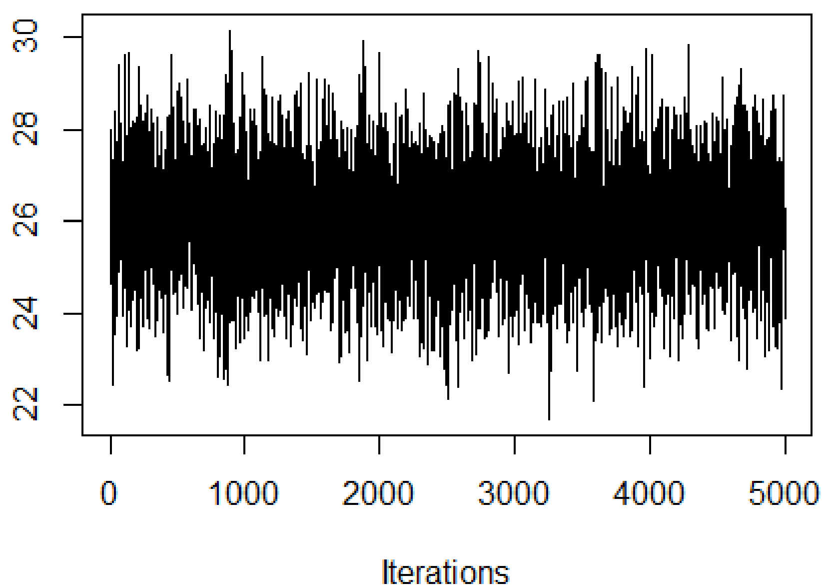


18
16
14
12
10
8
0
1000
2000
3000
4000
5000
Iterations


22
20
18
16
14
0
1000
2000
3000
4000
5000
Iterations


26
24
22
20
18
0
1000
2000
3000
4000
5000
Iterations






```
effectiveSize(A)

##     var1     var2
## 28.80347 27.62700
```

```r
effectiveSize(B)

##     var1     var2     var3
## 34.82263 27.22128 19.71153

effectiveSize(AB)

##      var1      var2      var3      var4      var5      var6
## 110.13178 100.98265  72.11270  93.53426  45.54367  53.57694

effectiveSize(MU)

##    var1
## 11.3708

effectiveSize(THETA)

##     var1     var2     var3     var4     var5     var6
## 5226.326 4883.513 5000.000 5000.000 4593.375 5427.124
```

We can see from the traceplot and effective sample size that a_i, b_j, $(ab)_{ij}$ and μ do not converge, but their sum θ do converge.

Factors makes the teeth of guinea pigs grow

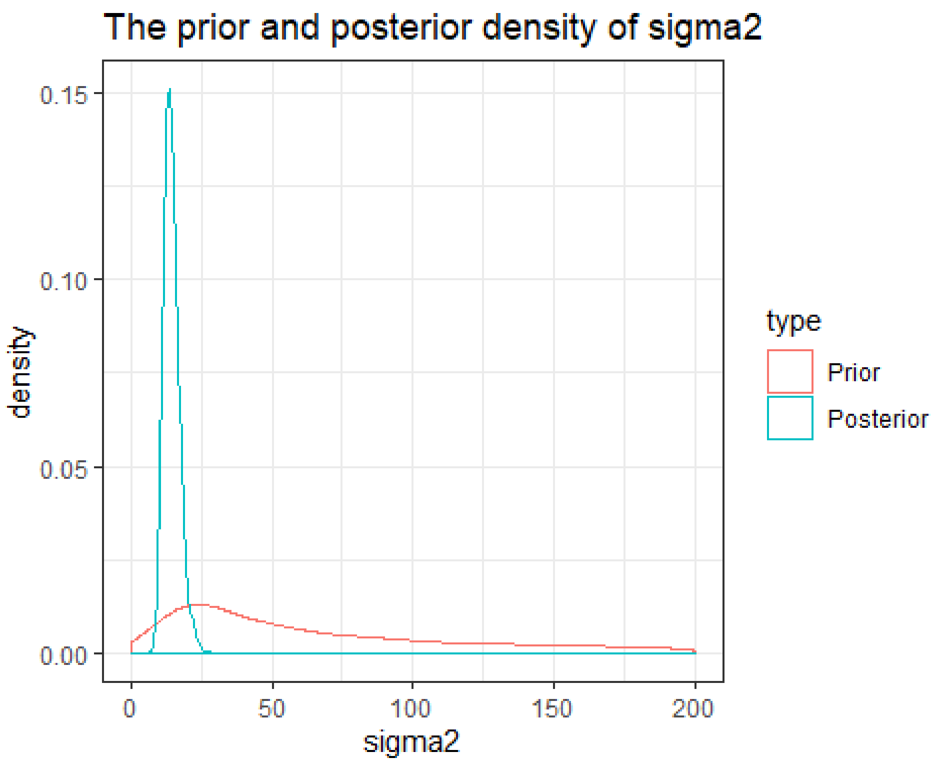
```r
apply(A,2,mean)

## [1] 3.058649 5.173398

apply(B,2,mean)

## [1] -3.027393  4.160089  8.421595

apply(AB,2,mean)

## [1] -2.2483941  0.7269147 -0.7265177  2.8944289  4.1211944  2.121268
7
```

We can see from the results of the posterior mean of factor a, b and ab, which shows that the orange juice delivery method with all does levels and ascorbic acid delivery method with 2mg/day does level make the teeth of guinea pigs grow.

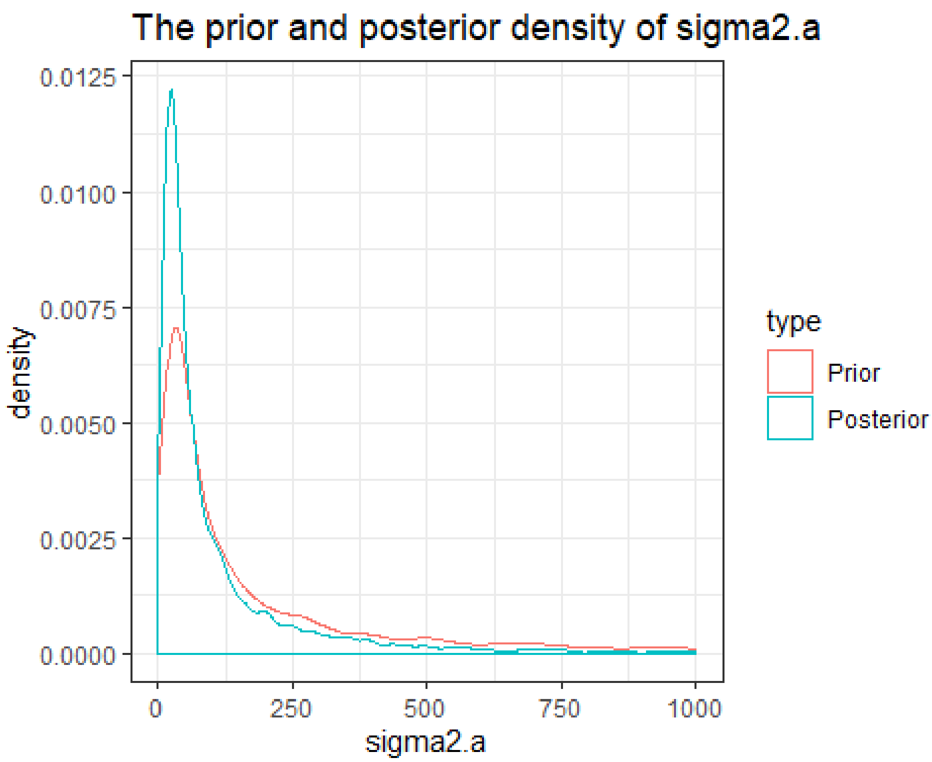
Justify the prior parameter choice

```r
### sigma2
sigma2.prior = data.frame(sigma2 = 1/rgamma(S, v0/2, v0*s20/2),
                          type = 'Prior')
sigma2.post = data.frame(sigma2 = SIGMA2,
                         type = 'Posterior')
sigma2.cp = rbind(sigma2.prior, sigma2.post)
ggplot(data = sigma2.cp, aes(x = sigma2,color=type)) +
  geom_density() +
```

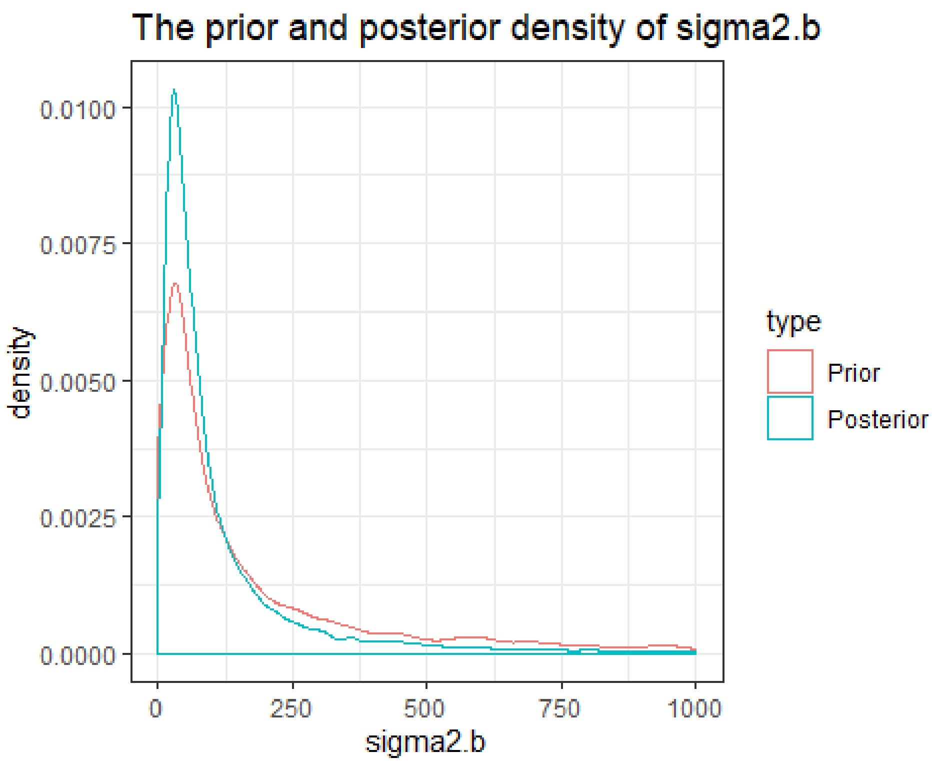
```
  labs(title = 'The prior and posterior density of sigma2') +
  xlim(0,200)
```



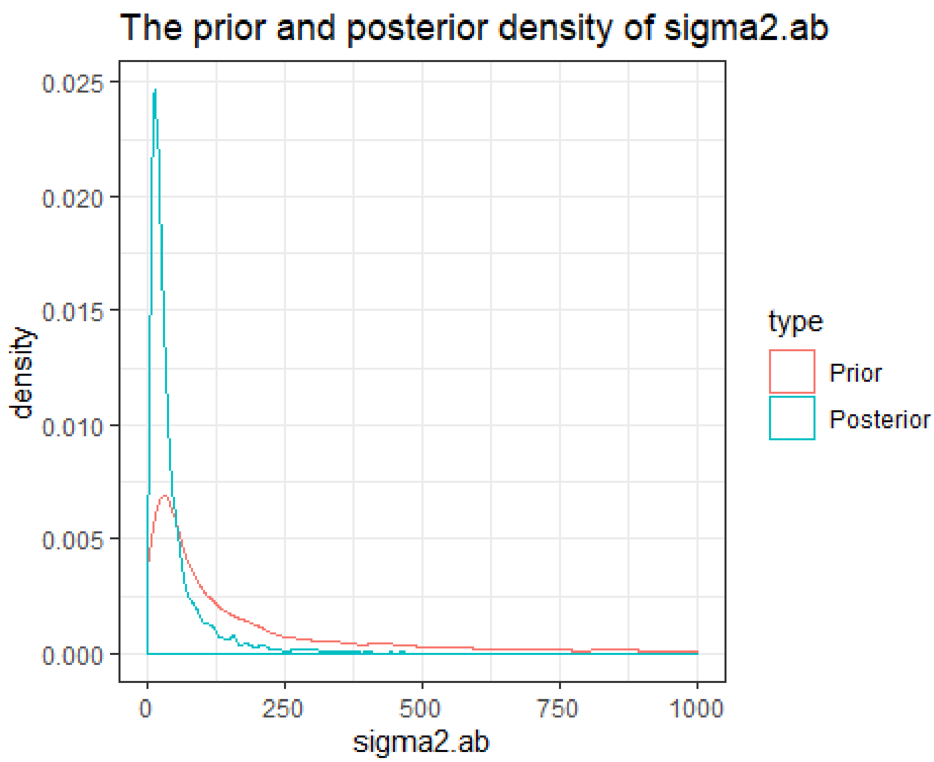

```
### sigma2.a
sigma2.a.prior = data.frame(sigma2.a = 1/rgamma(S, eta0.a/2, eta0.a*t0.
a/2),
                            type = 'Prior')
sigma2.a.post = data.frame(sigma2.a = SIGMA2A,
                           type = 'Posterior')
sigma2.a.cp = rbind(sigma2.a.prior, sigma2.a.post)
ggplot(data = sigma2.a.cp, aes(x = sigma2.a, color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of sigma2.a') +
  xlim(0,1000)
```



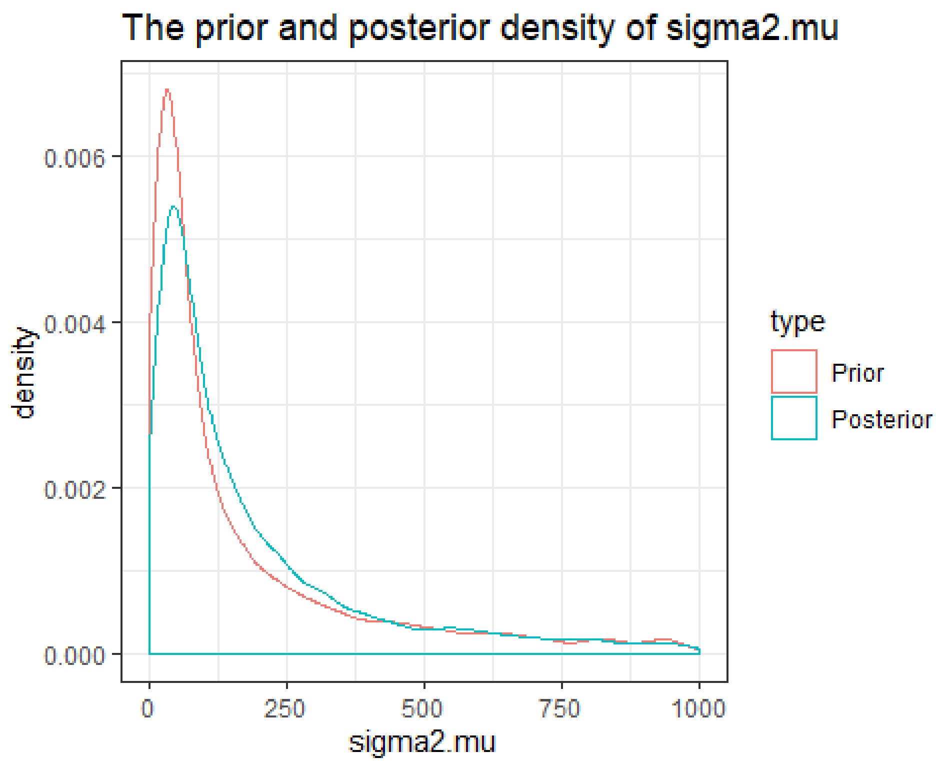

```
### sigma2.b
sigma2.b.prior = data.frame(sigma2.b = 1/rgamma(S, eta0.b/2, eta0.b*t0.
b/2),
                           type = 'Prior')
sigma2.b.post = data.frame(sigma2.b = SIGMA2B,
                          type = 'Posterior')
sigma2.b.cp = rbind(sigma2.b.prior, sigma2.b.post)
ggplot(data = sigma2.b.cp, aes(x = sigma2.b, color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of sigma2.b') +
  xlim(0,1000)
```




```
### sigma2.ab
sigma2.ab.prior = data.frame(sigma2.ab = 1/rgamma(S, eta0.ab/2, eta0.ab
*t0.ab/2),
                          type = 'Prior')
sigma2.ab.post = data.frame(sigma2.ab = SIGMA2AB,
                          type = 'Posterior')
sigma2.ab.cp = rbind(sigma2.ab.prior, sigma2.ab.post)
ggplot(data = sigma2.ab.cp, aes(x = sigma2.ab, color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of sigma2.ab') +
  xlim(0,1000)
```




```
### sigma2.mu
sigma2.mu.prior = data.frame(sigma2.mu = 1/rgamma(S, eta0.mu/2, eta0.mu
*t0.mu/2),
                          type = 'Prior')
sigma2.mu.post = data.frame(sigma2.mu = SIGMA2MU,
                         type = 'Posterior')
sigma2.mu.cp = rbind(sigma2.mu.prior, sigma2.mu.post)
ggplot(data = sigma2.mu.cp, aes(x = sigma2.mu, color=type)) +
  geom_density() +
  labs(title = 'The prior and posterior density of sigma2.mu') +
  xlim(0,1000)
```



As for the prior parameter choice, we can see from the comparison prior and posterior density plots of σ^2, σ_a^2, σ_b^2, σ_{ab}^2 and σ_μ^2 above, which shows that the prior parameters set for σ^2 and σ_{ab}^2 may be poor, but those for the others are somewhat proper.