MCMC

Yaqiong Yao

10/26/2018

```r
require("HI")
```

```
## Loading required package: HI
```

```r
require("MCMCpack")
```

```
## Loading required package: MCMCpack

## Loading required package: coda

## Loading required package: MASS

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

We know that :

$$x|\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2 \sim \delta N(\mu_1, \sigma_1^2) + (1-\delta) N(\mu_2, \sigma_2^2)$$

The prior distributions for five parameters are:

$$\begin{gathered}\pi(\delta) = 1 \\ \mu_1 \sim N(0, 100) \\ \mu_2 \sim N(0, 100) \\ \sigma_1^2 \sim IG(0.5, 10) \\ \sigma_2^2 \sim IG(0.5, 10)\end{gathered}$$

The full conditional distribution :

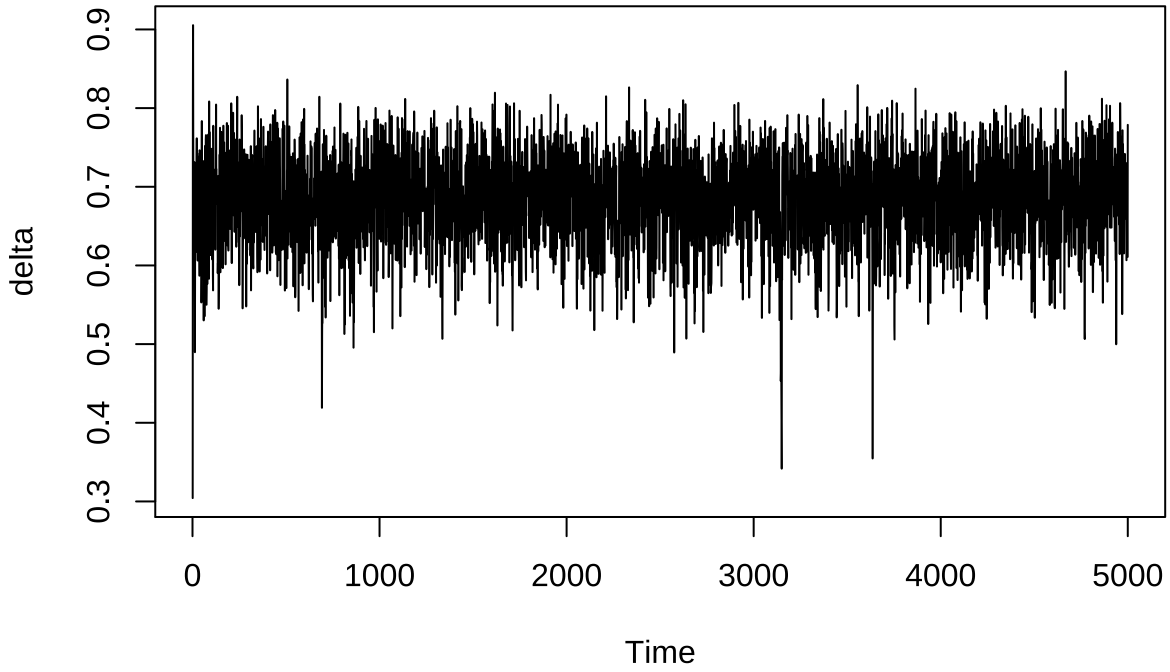
$$f(\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2|\mathbf{x}) \propto \prod_{i=1}^{n} \left(\delta \frac{1}{\sqrt{2\pi\sigma_1^2}} exp\left\{ -\frac{(x_i-\mu_1)^2}{2\sigma_1^2} \right\} + (1-\delta)\frac{1}{\sqrt{2\pi\sigma_2^2}} exp\left\{ -\frac{(x_i-\mu_2)^2}{2\sigma_2^2} \right\} \right)$$
$$\times exp\left\{ -\frac{\mu_1^2}{200} \right\} \times exp\left\{ -\frac{\mu_2^2}{200} \right\} \times (\sigma_1^2)^{-1.5} exp\left\{ -\frac{(\sigma_1^2)^2}{10} \right\} \times (\sigma_2^2)^{-1.5} exp\left\{ -\frac{(\sigma_2^2)^2}{10} \right\}$$

Take log on both side to get the loglikelihood function.

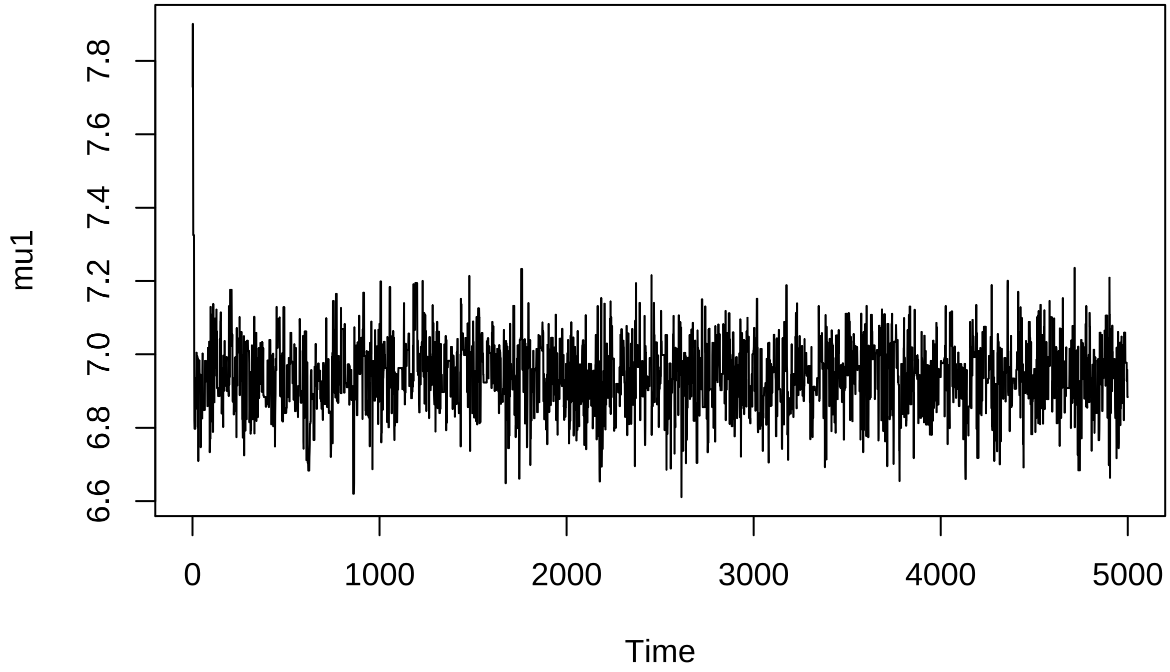
```r
delta <- 0.7
n <- 100
set.seed(123)
u <- rbinom(n, prob = delta, size = 1)
x <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)
```

```
mylike <- function(theta, x){
  delta <- theta[1]
  mu1 <- theta[2]; sigma21 <- theta[3]
  mu2 <- theta[4]; sigma22 <- theta[5]
  return(sum(log(delta*dnorm(x, mu1, sqrt(sigma21)) + (1-delta)*dnorm(x, mu2, sqrt(sigma22)))) +
    log(dnorm(mu1, 0, 10)) + log(dnorm(mu2, 0, 10)) + log(dinvgamma(sigma21, shape = 0.5, scale = 10)) +
    log(dinvgamma(sigma22, shape = 0.5, scale = 10)))
}
```

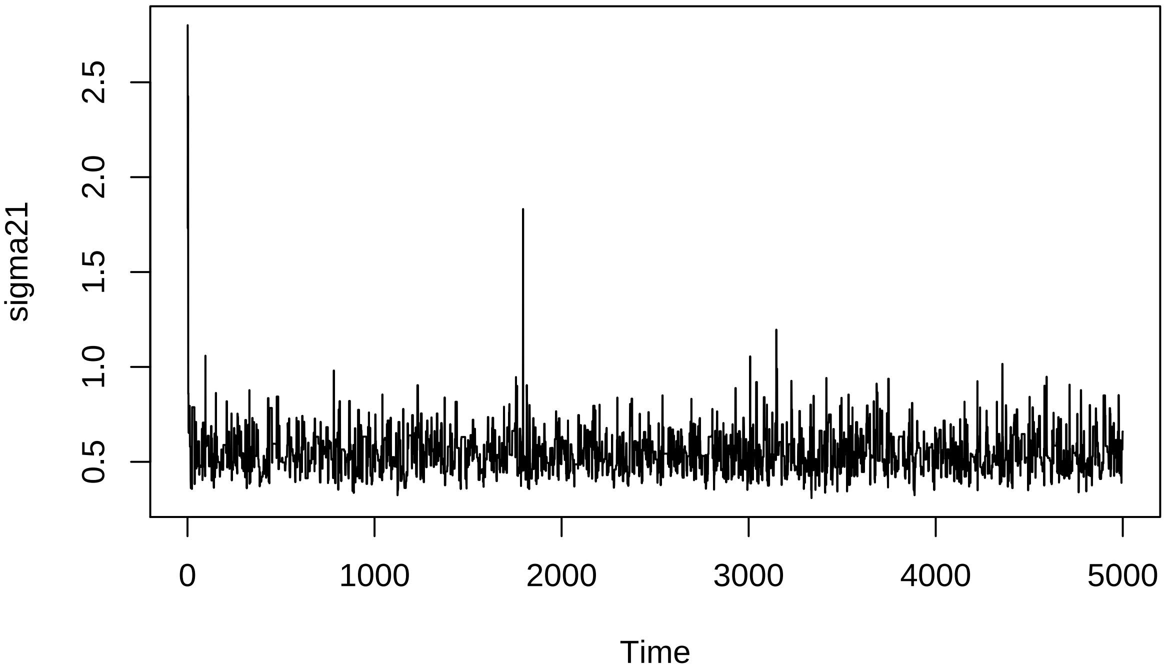
```
mymcmc <- function(niter, thetaInit, x) {
  p <- length(thetaInit)
  thetaCurrent <- thetaInit
  out <- matrix(NA, nrow = niter, ncol = p)
  for (i in 1:niter) {
    for (j in 1:p) {
      logFC <- function(thj) {
        theta <- thetaCurrent
        theta[j] <- thj
        mylike(theta, x)
      }

      if(j == 1){
        out[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
                 function(x) ((x > 0) * (x < 1)), 1)
      } else if(j %in% c(2, 4)){
        out[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
                 function(x) ((x > -10) * (x < 20)), 1)
      } else if(j %in% c(3, 5)){
        out[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
                 function(x) ((x > 0) * (x < 10)), 1)
      }
    }
  }
  out
}
niter <- 5000
thetaInit <- c(0.5, 0, 1, 0, 1)
set.seed(1026)
result <- mymcmc(niter, thetaInit, x)

plot(ts(result[,1]), ylab = "delta")
```



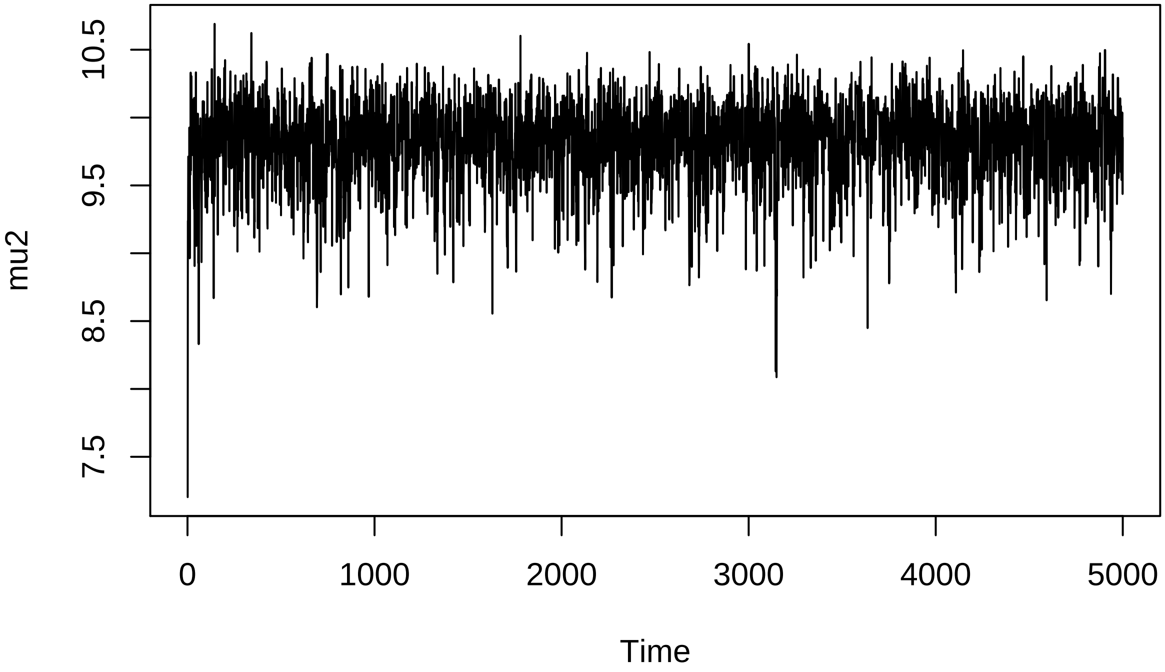
```
plot(ts(result[,2]), ylab = "mu1")
```



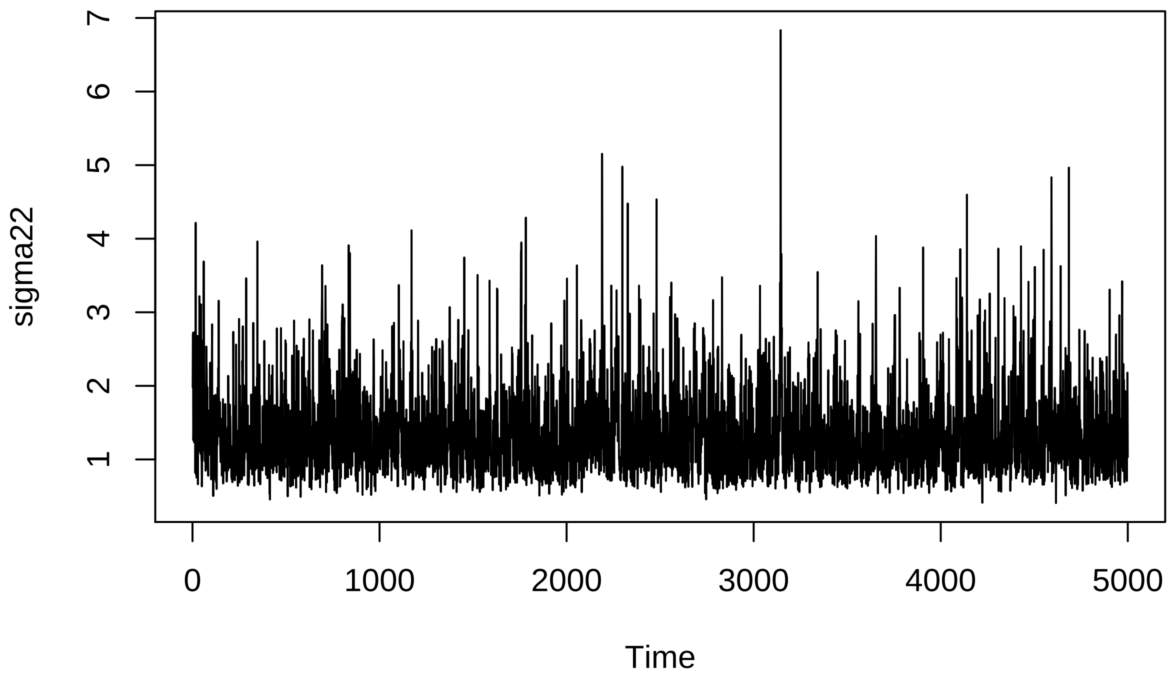
```
plot(ts(result[,3]), ylab = "sigma21")
```



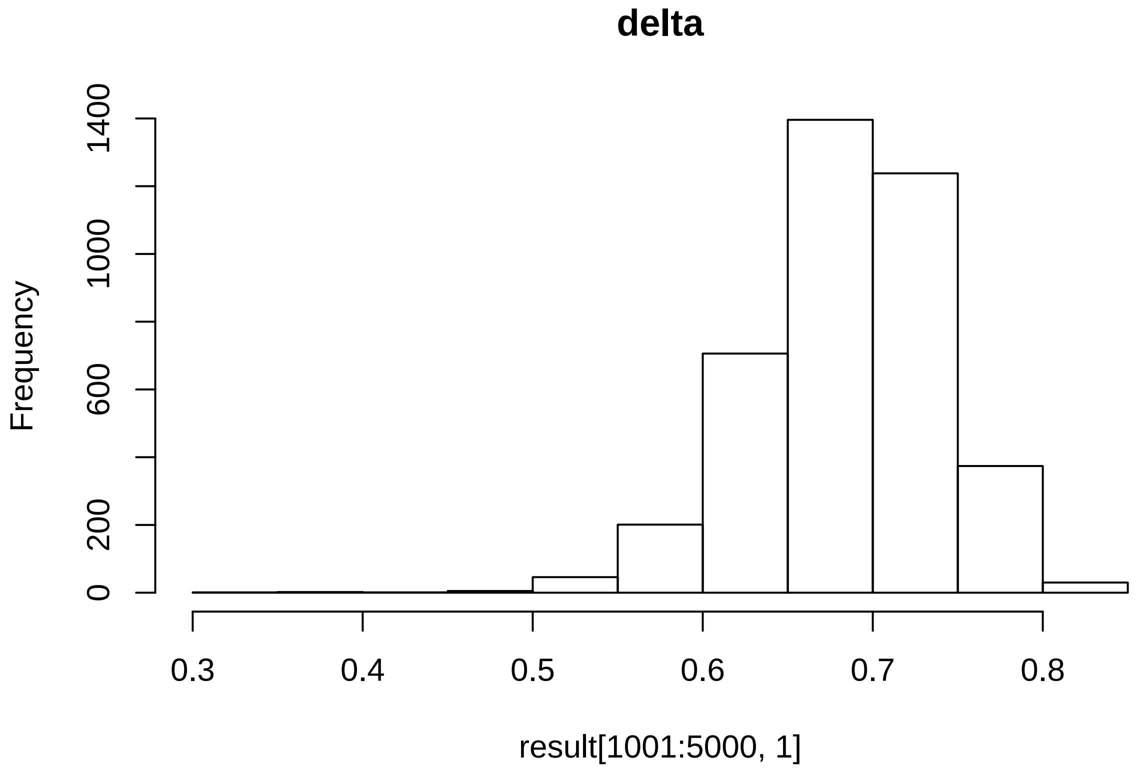
```r
plot(ts(result[,4]), ylab = "mu2")
```



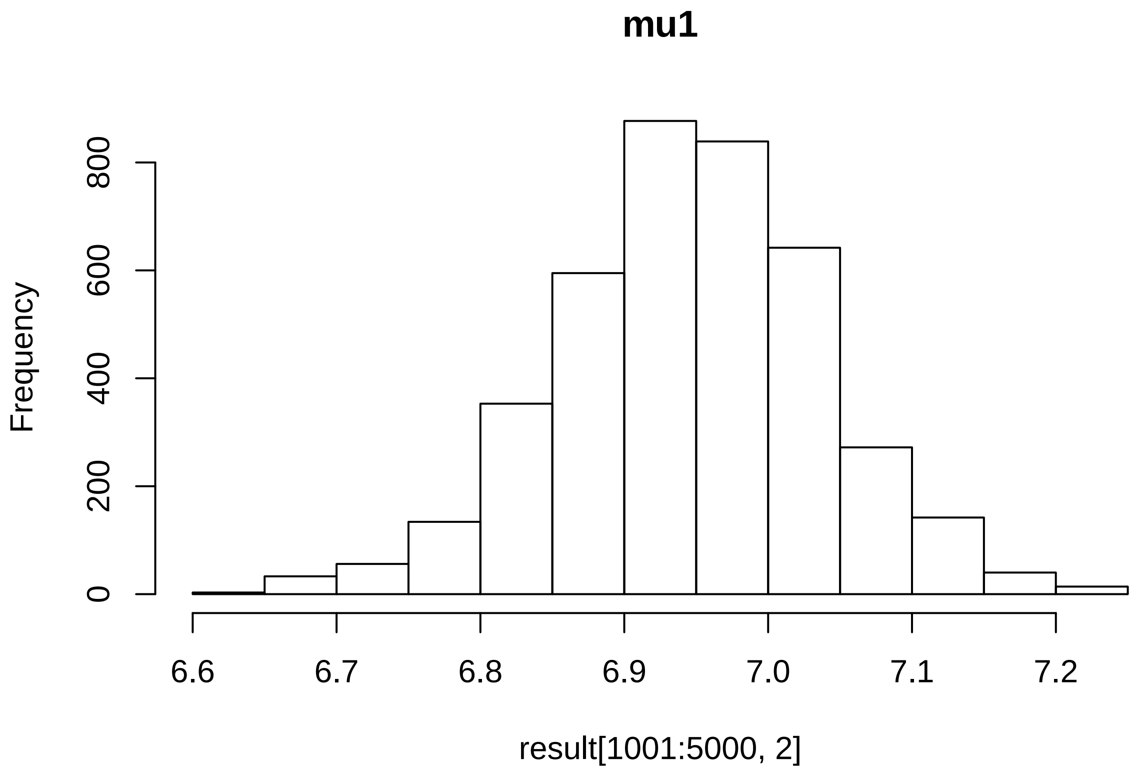
```r
plot(ts(result[,5]), ylab = "sigma22")
```



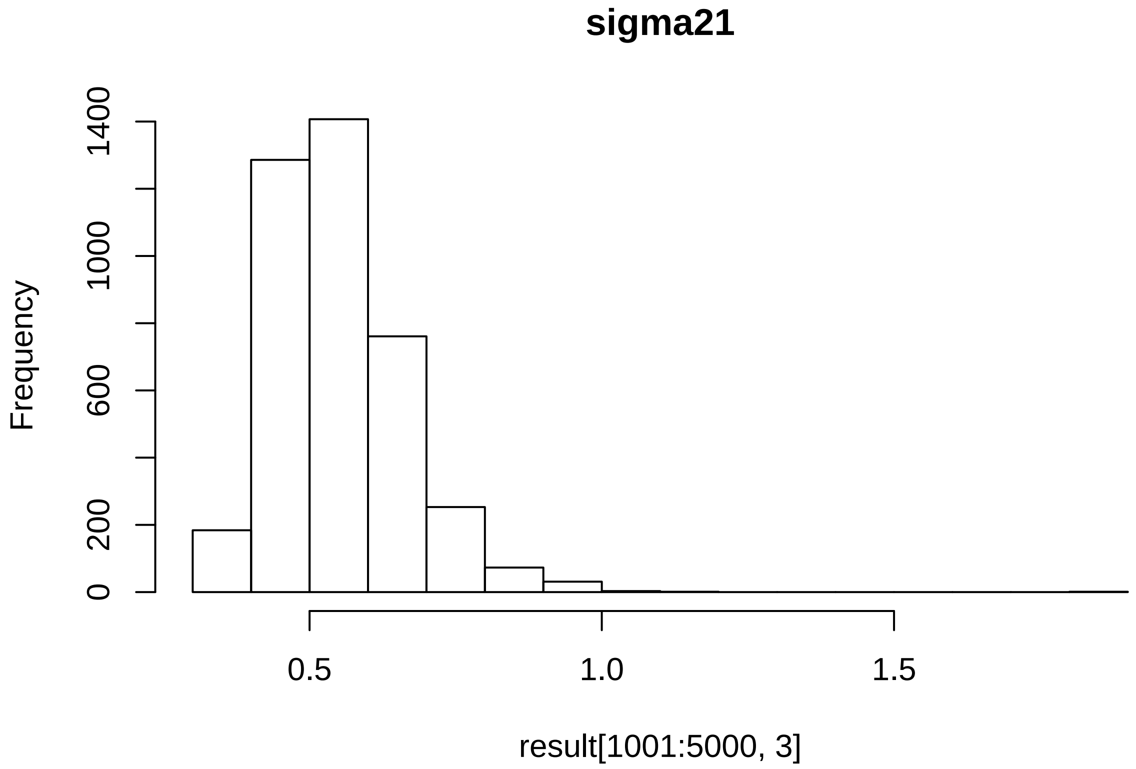
```
hist(result[1001:5000, 1], main = "delta")
```



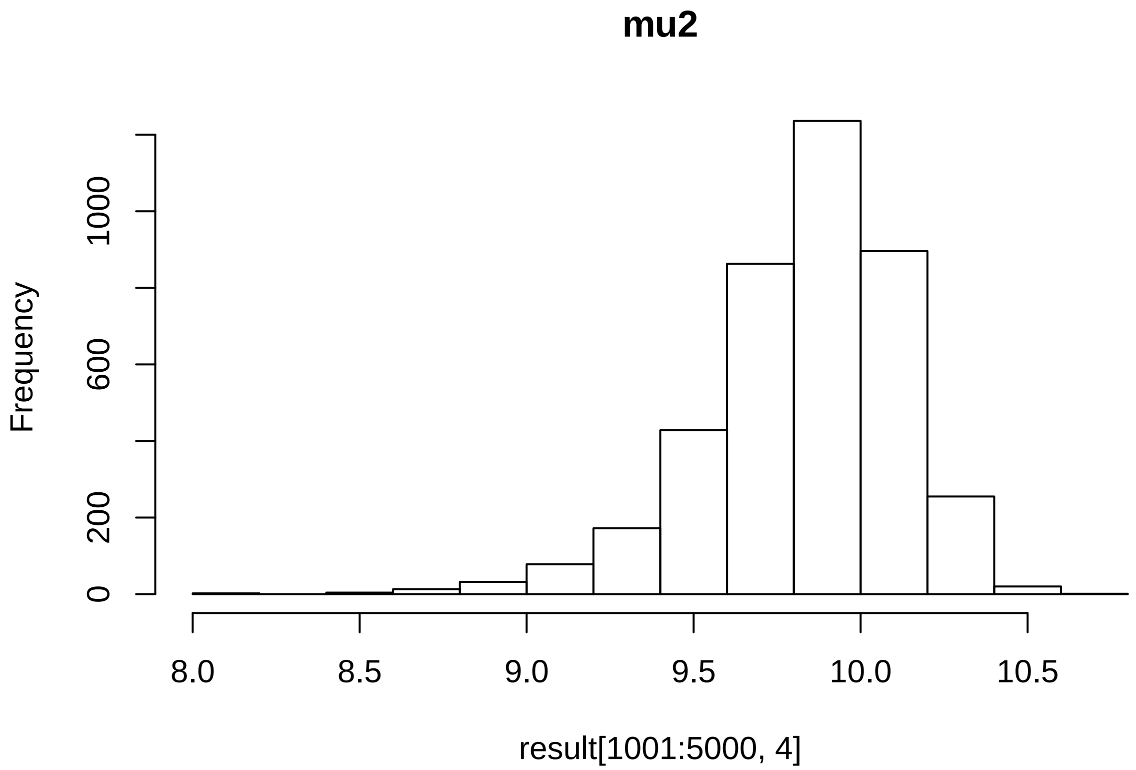
```
hist(result[1001:5000, 2], main = "mu1")
```



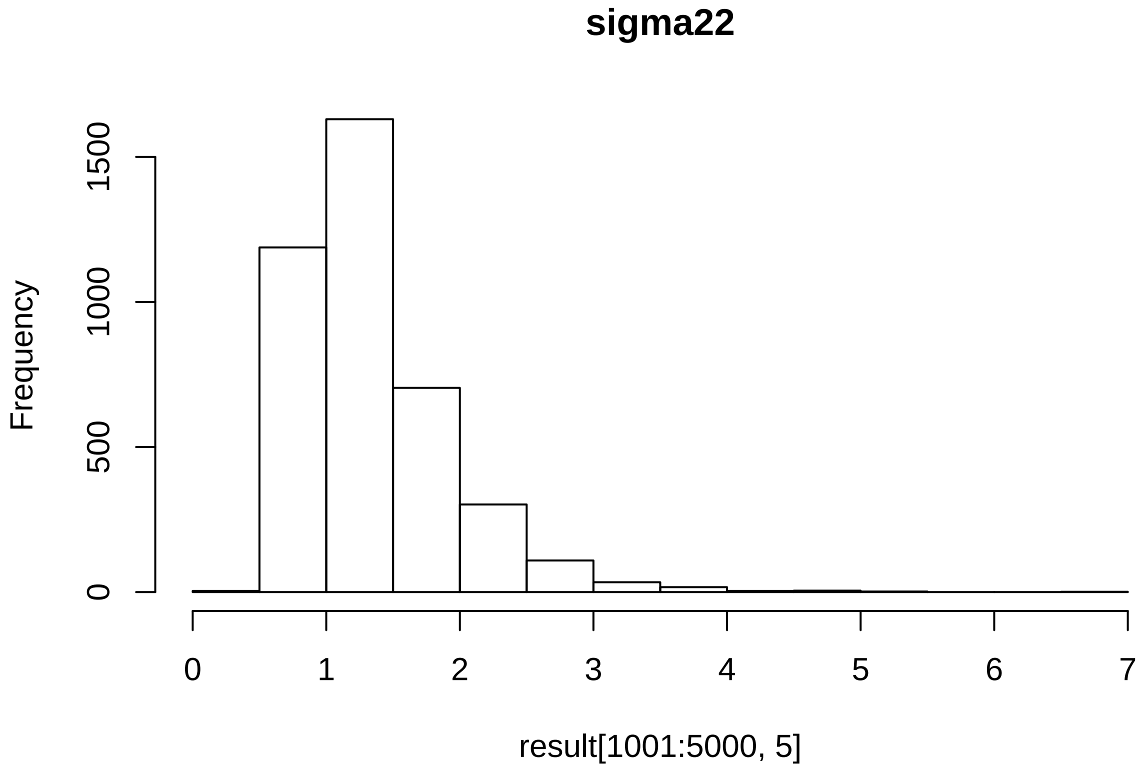
```
hist(result[1001:5000, 3], main = "sigma21")
```



```
hist(result[1001:5000, 4], main = "mu2")
```



```r
hist(result[1001:5000, 5], main = "sigma22")
```



I only take last 4000 result in histogram and the result we gets from mcmc is similar to the true value.