MCMC

Yuance He

10/26/2018

Normal mixture:

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

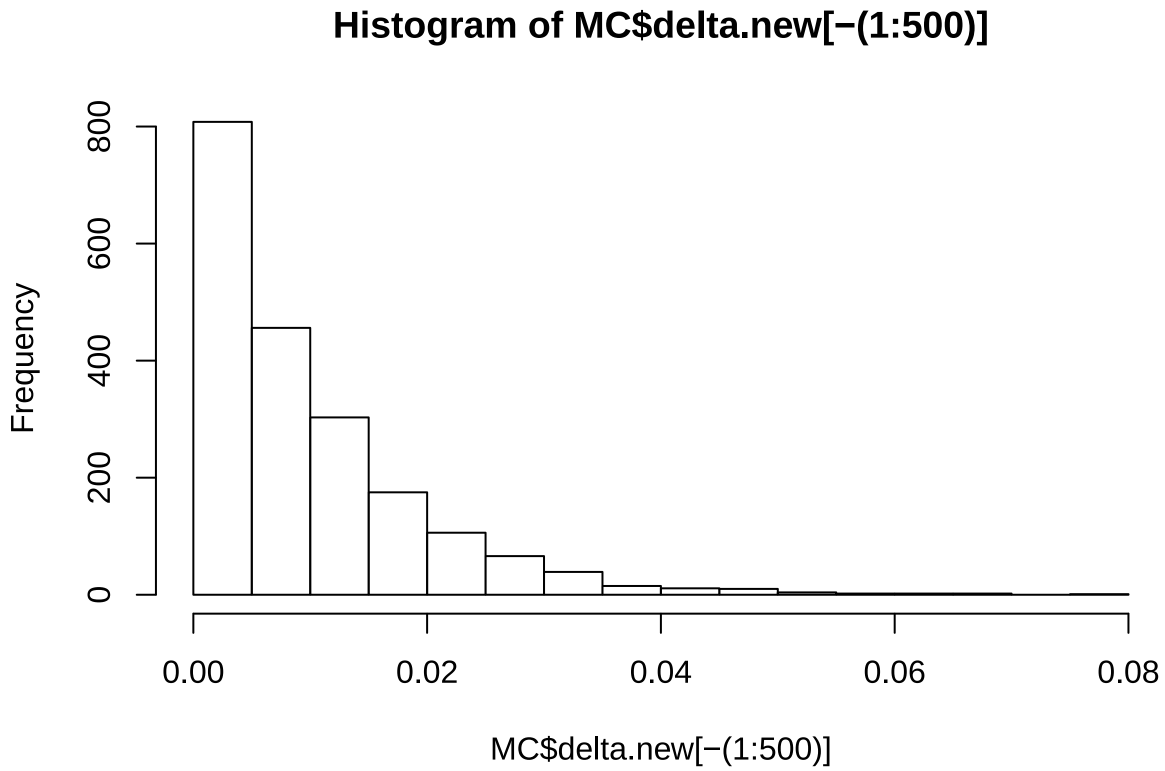
This time we don't kwon exact value of μ_1, μ_2, σ_1 and σ_2, what we only know is their distribution Normal and Inverse Gamma.

First we generate data:

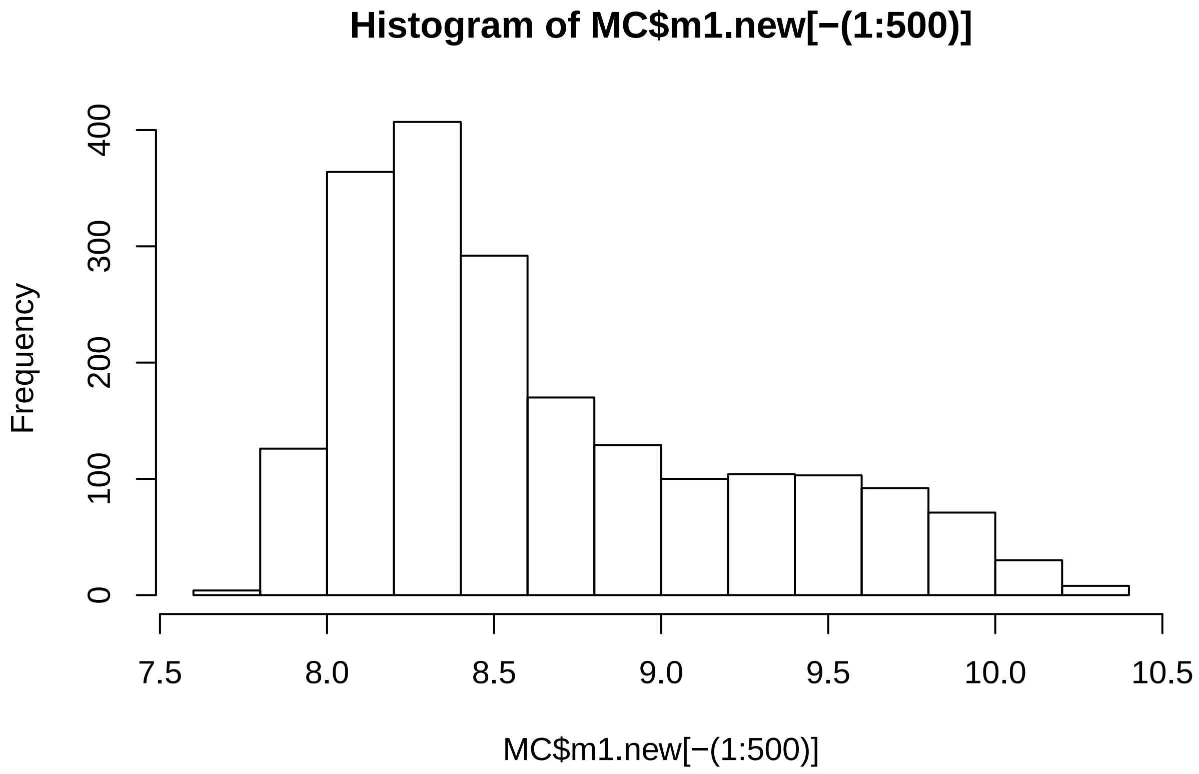
```r
delta <- 0.7
n <- 100
set.seed(123)
u <- rbinom(n, prob = delta, size = 1)
x <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)
##Now we build likelihood function, in this case
##delta, mu_1, sigma_1, mu_2 and sigma_2 are all unknown,
##so our log-likelihood function is:
myloglike <- function(delta,m1,m2,s1,s2,X){

  sum(log(delta * dnorm(x, m1, sqrt(s1)) + (1 - delta) * dnorm(x, m2, sqrt(s2))))
}
##Based on the normal log-likelihood function,
##we need to generate LOGPOST.
library(invgamma)
logpost <- function(delta, m1, m2, s1, s2,x){
  m1.log <- dnorm(m1, 0, 10, log = T)
  m2.log <- dnorm(m2, 0, 10, log = T)
  s1.log <- dinvgamma(s1, shape = 0.5, scale = 0.1, log = T)
  s2.log <- dinvgamma(s2, shape = 0.5, scale = 0.1, log = T)
  sum(m1.log + m2.log +s1.log + s2.log) + myloglike(delta, m1, m2, s1, s2,x)
}
```

Then we need to build our MCMC using Gibbs sampling approach:

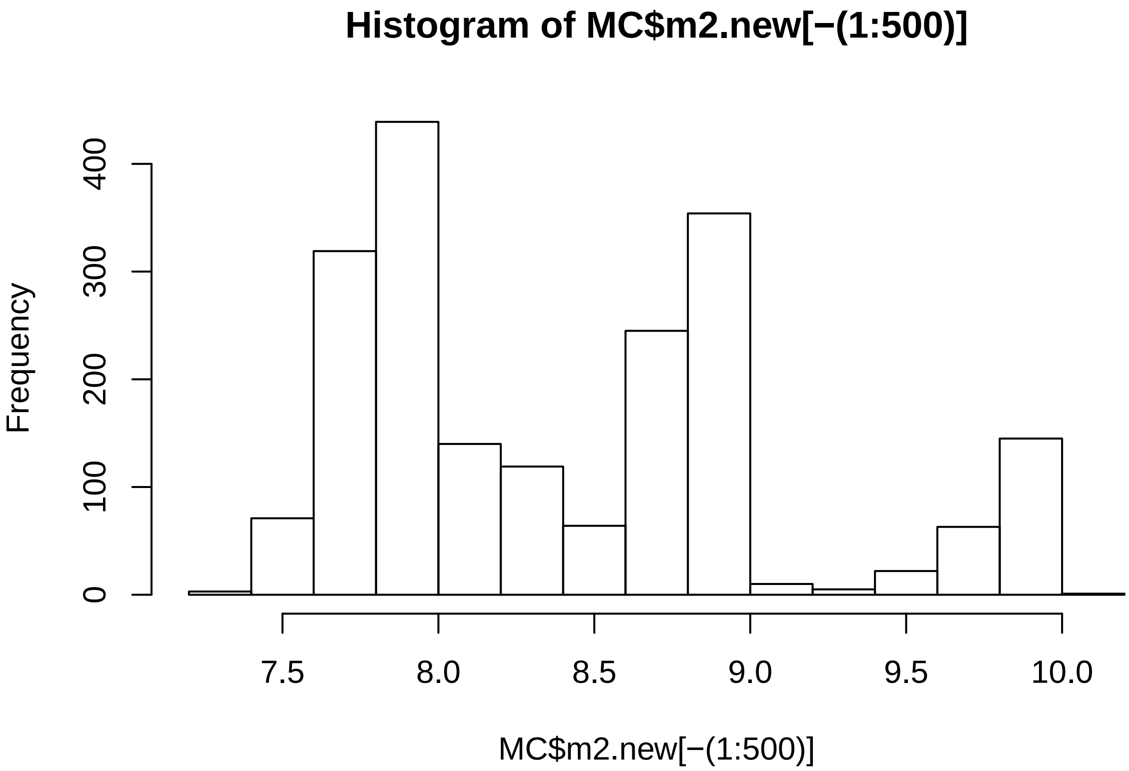
```r
library(HI)
##Define MCMC function
mymcmc <- function(initial, x, niter){
  ##Set up the output list
  delta.new <- m1.new <- m2.new <- s1.new <- s2.new <- rep(0,niter)

  ##Set up initial value
  delta.int <- initial[1]
  m1.int <- initial[2]
  m2.int <- initial[3]
  s1.int <- initial[4]
  s2.int <- initial[5]
  initial <- c(delta.int,m1.int,m2.int,s1.int,s2.int)

  ##Set up values in the loop
  delta.c <- delta.int; m1.c <- m1.int; m2.c <- m2.int; s1.c <- s1.int; s2.c <- s2.int
  current <- c(delta.c,m1.c,m2.c,s1.c,s2.c)

  for (i in 1:niter) {
    Db <- function(delta) logpost(delta,m1.c,m2.c,s1.c,s2.c,x)
    delta.new[i] <- arms(delta.c,Db,function(delta)(delta>0)*(delta<1),1)
    M1b <- function(m1) logpost(delta.new[i],m1,m2.c,s1.c,s2.c,x)
    m1.new[i] <- arms(m1.c,M1b,function(m1)(m1>-50)*(m1<50),1)
    M2b <- function(m2)logpost(delta.new[i],m1.new[i],m2,s1.c,s2.c,x)
    m2.new[i] <- arms(m2.c,M2b,function(m2)(m2>-50)*(m2<50),1)

    ##Import INVGAMMA
    library(invgamma)
    S1b <- function(s1) logpost(delta.new[i],m1.new[i],m2.new[i],s1,s2.c,x)
    s1.new[i] <- arms(s1.c,S1b,function(s1)(s1>0)*(s1<50),1)
    S2b <- function(s2) logpost(delta.new[i],m1.new[i],m2.new[i],s1.new[i],s2,x)
    s2.new[i] <- arms(s2.c,S2b,function(s2)(s2>0)*(s2<50),1)
    new <- c(delta.new[i],m1.new[i],m2.new[i],s1.new[i],s2.new[i])
    current <- new
  }
  list(delta.new=delta.new,m1.new=m1.new,m2.new=m2.new,s1.new=s1.new,s2.new=s2.new)
}
niter <- 2500
initial <- c(0.5,1,2,1,2)
MC <- mymcmc(initial,x,niter)
##Graphs and 500 burn-in period
hist(MC$delta.new[-(1:500)])
```



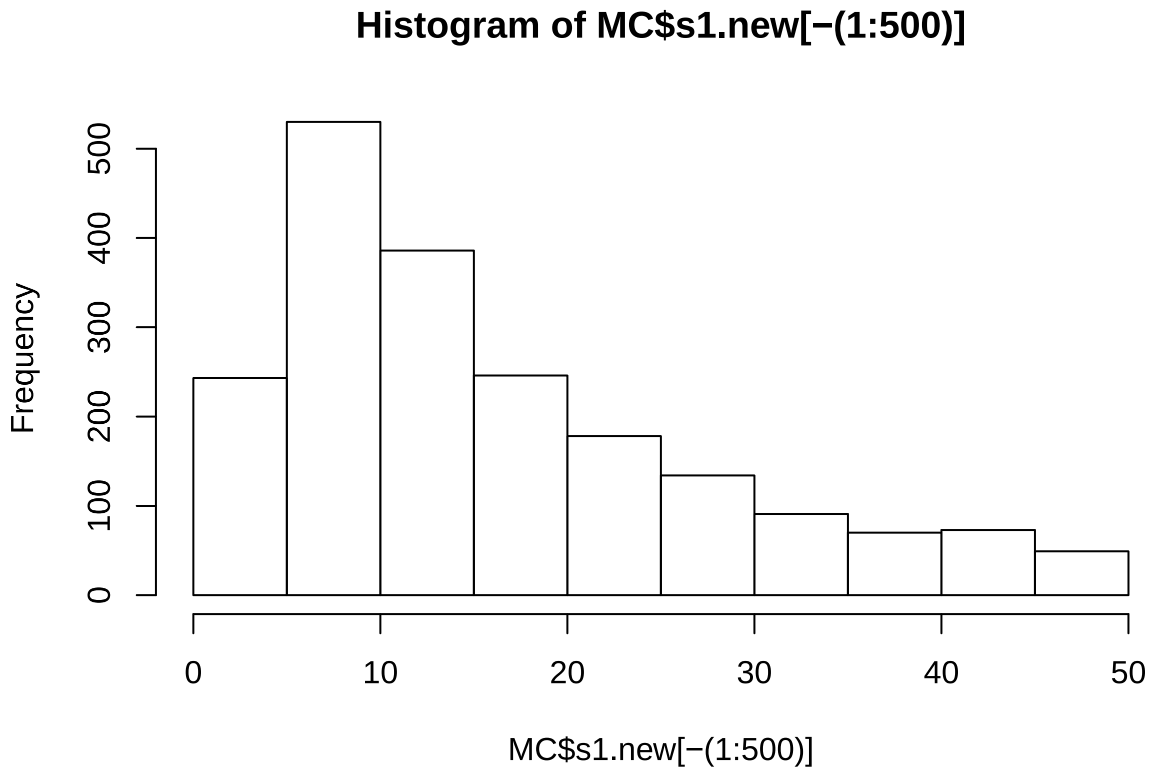
```
hist(MC$m1.new[-(1:500)])
```



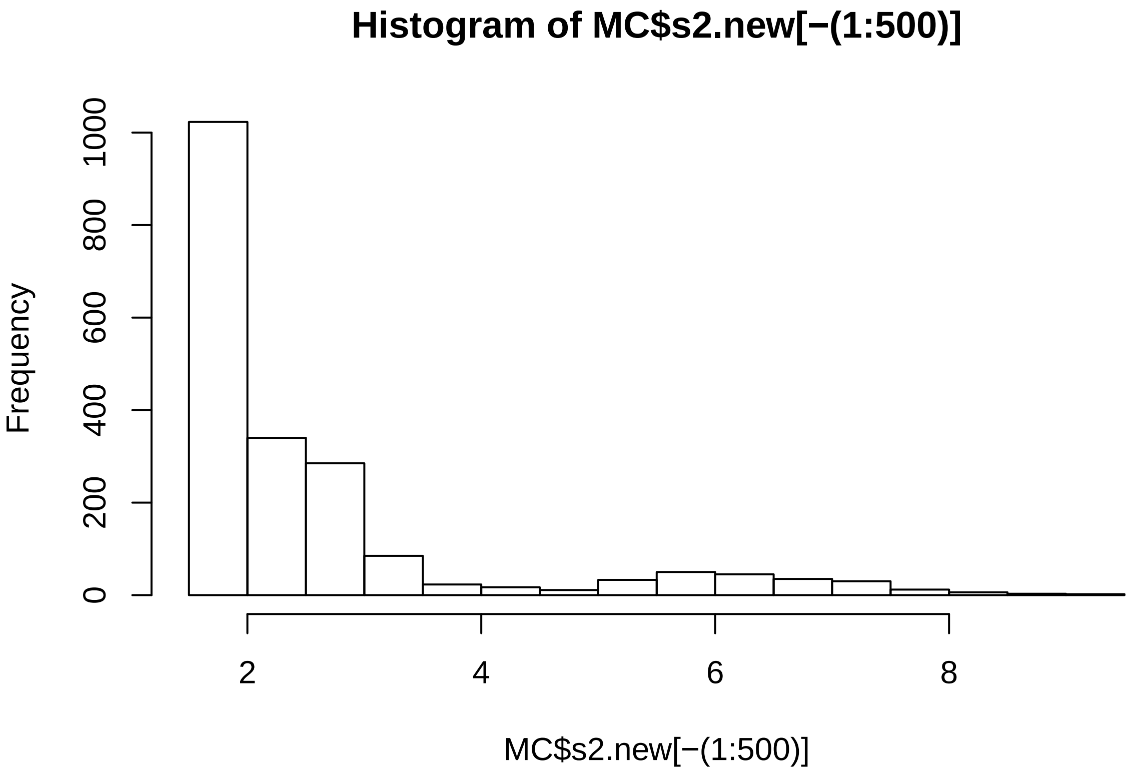
```
hist(MC$m2.new[-(1:500)])
```



```r
hist(MC$s1.new[-(1:500)])
```



```r
hist(MC$s2.new[-(1:500)])
```


Histogram of MC$s2.new[−(1:500)]
Frequency
0
200
400
600
800
1000
2
4
6
8
MC$s2.new[−(1:500)]