HW7 - Exercise7

JooChul Lee

26 October 2018

Contents

1 Normal mixture revisited

Under the assumption, δ is an uninformative prior and we use a simple random walk. The prior for μ_1 and μ_2 are $N(0, 10^2)$, that the prior for σ_1^2 and σ_2^2 are $IV\Gamma(a, b)$ with shape $a = .5$ and scale $b = 10$,$\pi(\delta), \pi(\mu_1), \pi(\mu_2), \pi(\sigma_1^2)$, and $\pi(\sigma_2^2)$, respectively.

Suppose that we consider finite mixture normal distribution, **X**

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

$f(x, \delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) \propto (\delta\frac{1}{\sigma_1}exp\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\} + (1-\delta)\frac{1}{\sigma_2}exp\{-\frac{(x-\mu_2)^2}{2\sigma_2^2}\}) \times \pi(\delta) \times \pi(\mu_1) \times \pi(\mu_2) \times \pi(\sigma_1^2) \times \pi(\sigma_2^2)$

Then, MCMC using Gibbs sampling is the following;

Step 1 : Draw δ^t from $\pi(\delta^t|\mu_1^{t-1}, \mu_2^{t-1}, \sigma_1^{t-1}, \sigma_2^{t-1}, x^t)$

Step 2 : Draw μ_1^t from $\pi(\mu_1|\delta^t, \mu_2^{t-1}, \sigma_1^{t-1}, \sigma_2^{t-1}, x^t)$

Step 3 : Draw μ_2^t from $\pi(\mu_2|\delta^t, \mu_1^t, \sigma_1^{t-1}, \sigma_2^{t-1}, x^t)$

Step 4 : Draw σ_1^t from $\pi(\mu_2|\delta^t, \mu_1^t, \mu_2^t, \sigma_2^{t-1}, x^t)$

Step 5 : Draw σ_2^t from $\pi(\mu_2|\delta^t, \mu_1^t, \mu_2^t, \sigma_1^t, x^t)$

Repeat Step 1 - 5

```
library(MCMCpack)
```

```
## Loading required package: coda
```

```
## Loading required package: MASS
```

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

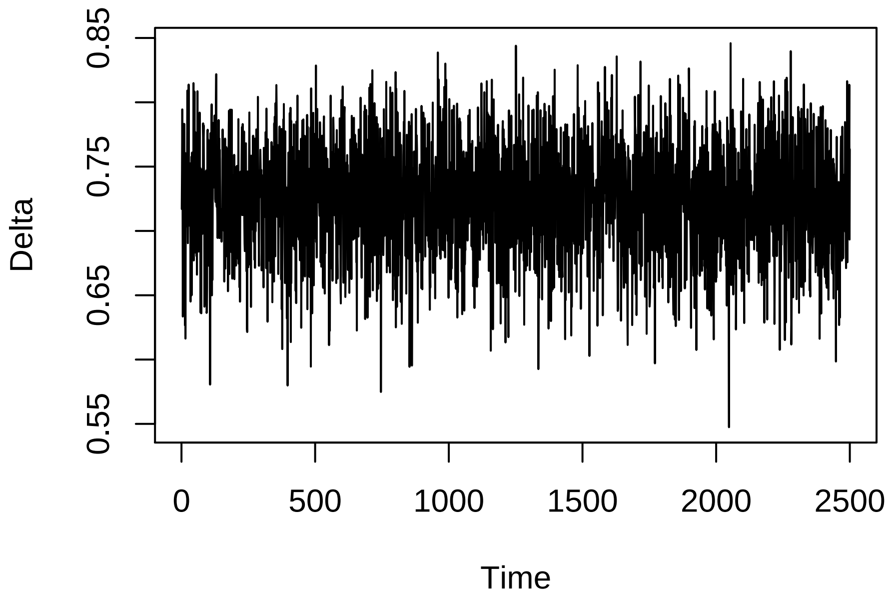
```
## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

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

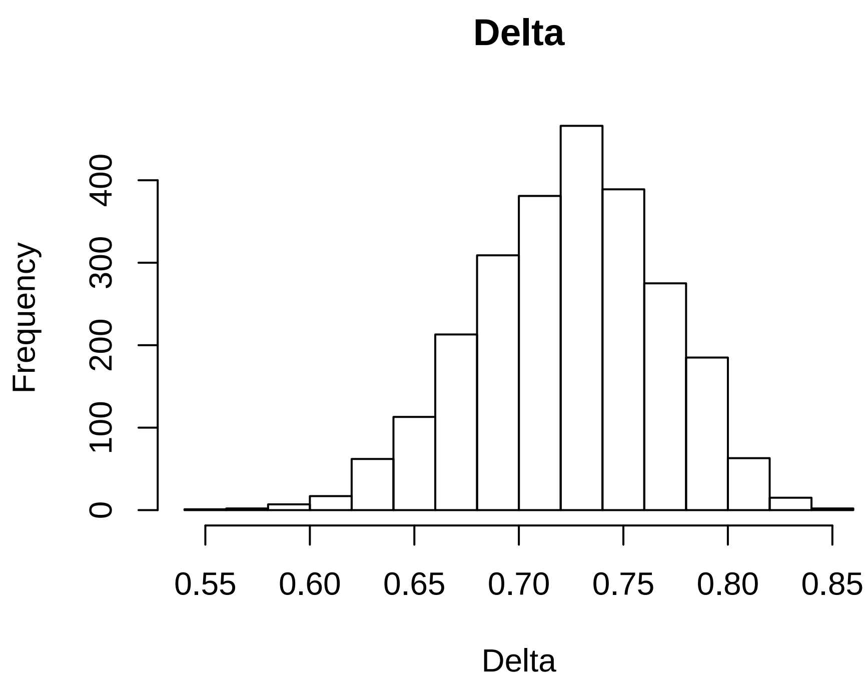
```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
library(HI)
delta <- 0.7
n <- 100
set.seed(254)
u <- rbinom(n, prob = delta, size = 1)
x <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)

loglik = function(x,mu1,mu2,sigma1,sigma2,delta)
{
   Fisrt =  log( ( delta * dnorm(x, mu1,sigma1) ) + ( (1-delta) * dnorm(x,mu2,sigma2) ) )
   return(sum(Fisrt))
}

HW7 = function(x,ini.D,ini.mu1,ini.mu2, ini.sigma1,ini.sigma2, iter)
{
   inix=x
   current.D <- ini.D; current.mu1 <- ini.mu1; current.mu2 <- ini.mu2; current.sigma1 <- ini.si
   current.sigma2 <- ini.sigma2
   new.D <- new.mu1 <- new.mu2 <- new.sigma1 <- new.sigma2 <- rep(0,iter)
   for(i in 1:iter)
   {
      D_v =function(delta) loglik(x =inix, mu1 = current.mu1, mu2 = current.mu2, sigma1 = curre
                                  sigma2 = current.sigma2, delta = delta)
      new.D[i] =  arms(current.D, D_v, function(delta) {(delta>0)*(delta<1)}, 1)

      mu1.v = function(mu1) loglik(x =inix, mu1 = mu1, mu2 = current.mu2,
                             sigma1 = current.sigma1, sigma2 = current.sigma2, delta = new.D[i]
         log( dnorm(mu1,0,10^2) )
      new.mu1[i] = arms(current.mu1, mu1.v, function(mu1) (mu1 > -30) * (mu1 < 30), 1)

      mu2.v = function(mu2) loglik(x =inix, mu1 = new.mu1[i], mu2 = mu2,
                             sigma1 = current.sigma1, sigma2 = current.sigma2, delta = new.D[i]
         log( dnorm(mu2,0,10^2) )
      new.mu2[i] = arms(current.mu2, mu2.v, function(mu2) (mu2 > -30) * (mu2 < 30), 1)

      sigma1.v = function(sigma1) loglik(x =inix, mu1 = new.mu1[i], mu2 = new.mu2[i],
                             sigma1 = sigma1, sigma2 = current.sigma2, delta = new.D[i]) +
         log(dinvgamma(sigma1, 0.5 , 0.1))
      new.sigma1[i] = arms(current.sigma1, sigma1.v, function(sigma1) (sigma1> 0) * (sigma1 < 1

      sigma2.v = function(sigma2) loglik(x =inix, mu1 = new.mu1[i], mu2 = new.mu2[i],
                                   sigma1 = new.sigma1[i], sigma2 = sigma2, delta = new.D[i]) +
         log(dinvgamma(sigma2, 0.5 , 0.1))
      new.sigma2[i] = arms(current.sigma2, sigma2.v, function(sigma2) (sigma2 > 0) * (sigma2 <
```

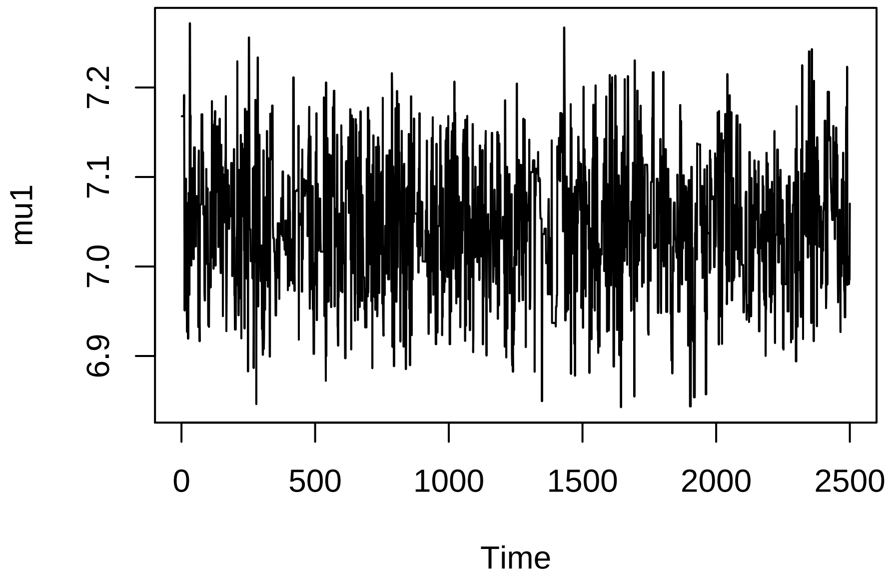
```r
      current.D = new.D[i]; current.mu1 = new.mu1[i]; current.mu2 = new.mu2[i];
      current.sigma1 = new.sigma1[i]; current.sigma2 = new.sigma2[i]
   }
  list(new.D = new.D, new.mu1= new.mu1, new.mu2 = new.mu2, new.sigma1= new.sigma1, new.sigma2 =
}
joo = HW7(x,0.5,1,1,1,1, iter = 3000)
plot(ts(joo$new.D[-(1:500)]), ylab = 'Delta')
```



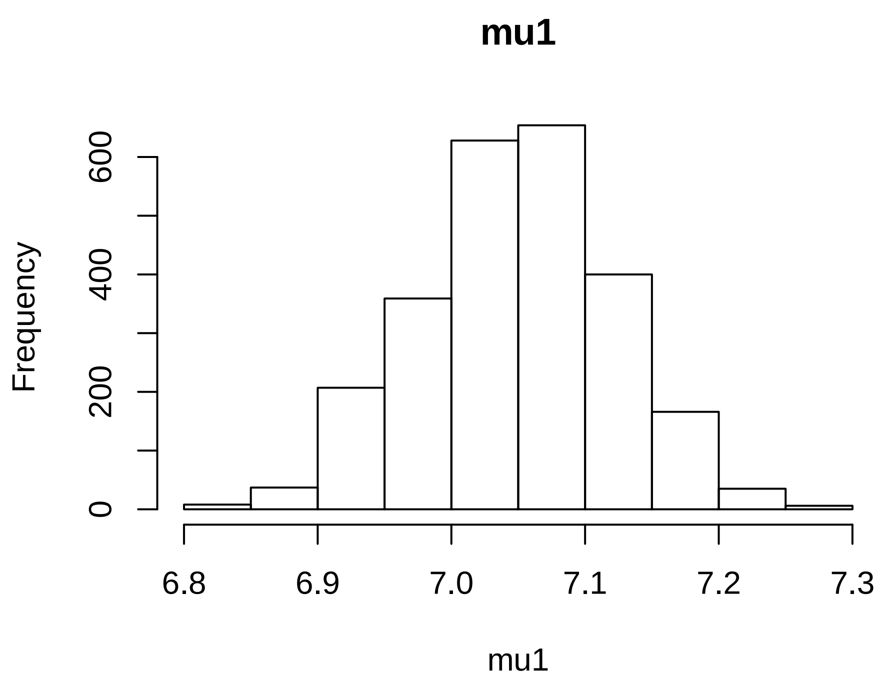

```r
hist(ts(joo$new.D[-(1:500)]), main = 'Delta',xlab = 'Delta')
```



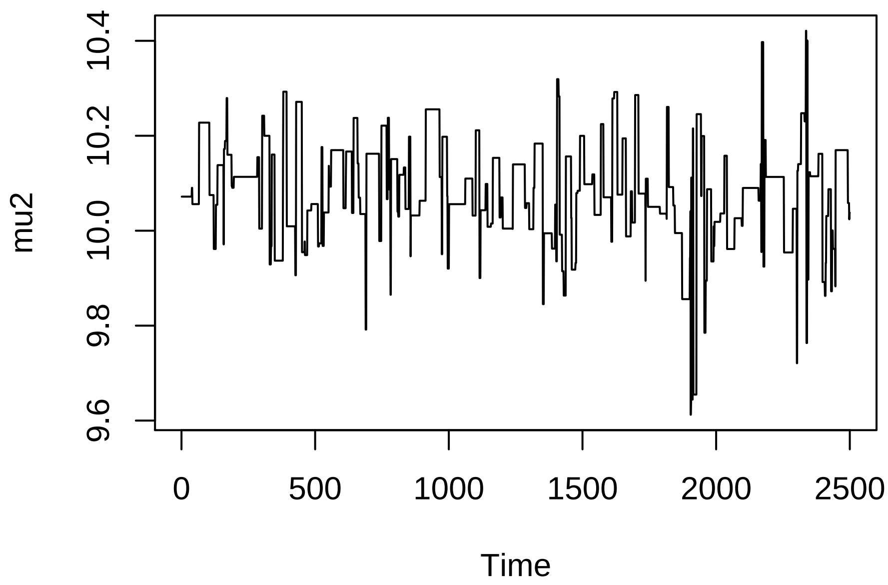
```r
plot(ts(joo$new.mu1[-(1:500)]), ylab = 'mu1')
```



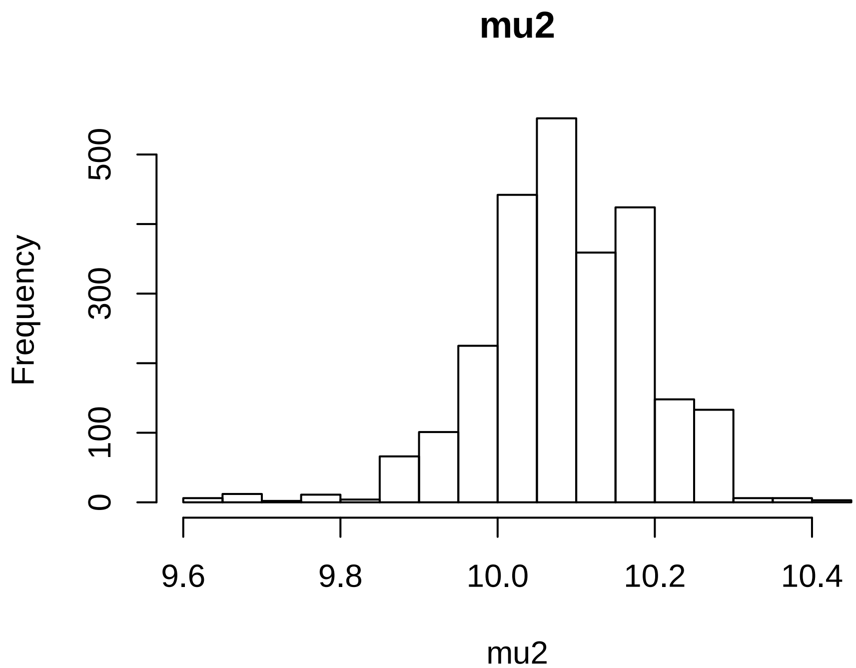

```r
hist(ts(joo$new.mu1[-(1:500)]), main = 'mu1',xlab = 'mu1')
```



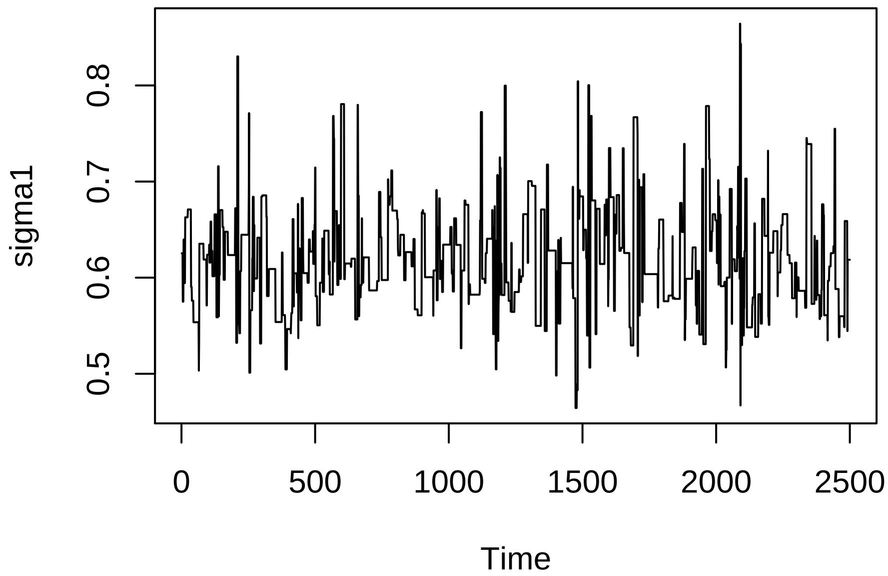

```r
plot(ts(joo$new.mu2[-(1:500)]), ylab = 'mu2')
```



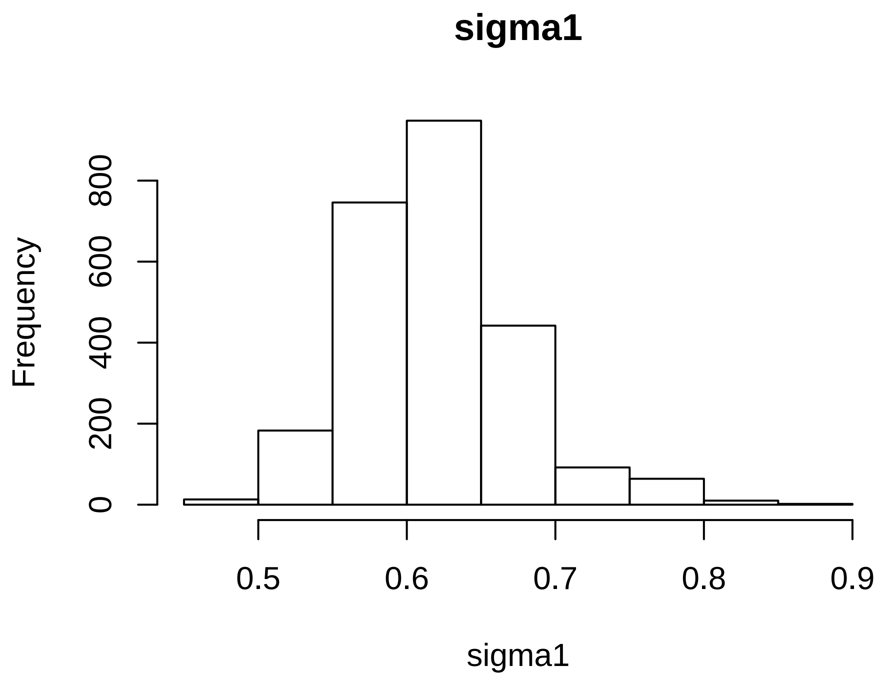

```
hist(ts(joo$new.mu2[-(1:500)]), main = 'mu2',xlab = 'mu2')
```



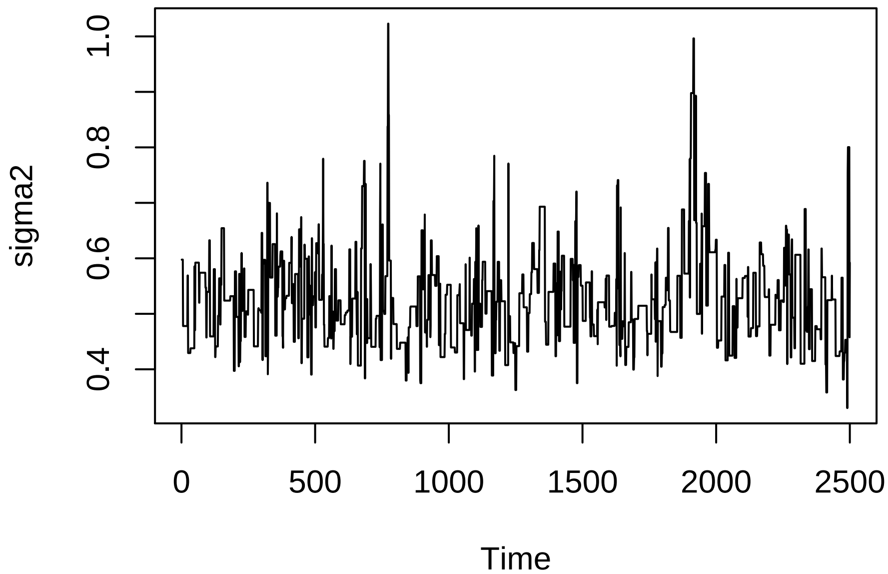

```
plot(ts(joo$new.sigma1[-(1:500)]), ylab = 'sigma1')
```



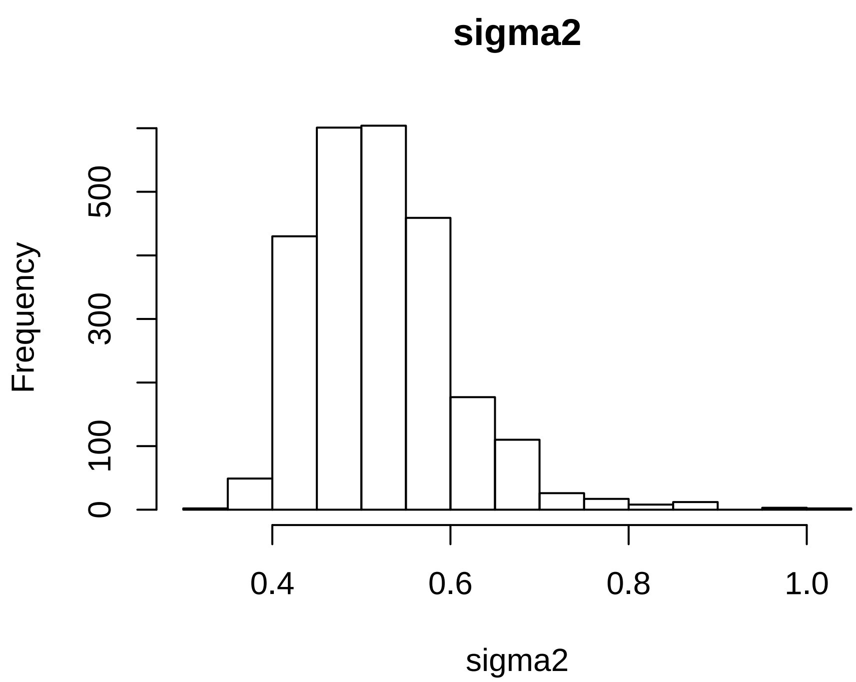

```
hist(ts(joo$new.sigma1[-(1:500)]), main = 'sigma1',xlab = 'sigma1')
```




```
plot(ts(joo$new.sigma2[-(1:500)]), ylab = 'sigma2')
```




```r
hist(ts(joo$new.sigma2[-(1:500)]), main = 'sigma2',xlab = 'sigma2')
```



500 runs are droped for the burn-in period.