Normal Mixture in R Markdown

*Cosmin Borsa**

11/02/2018

Abstract

This document is a homework assignment for the course Statistical Computing at the University of Connecticut.

1 Normal Mixture

In this section we will now consider a normal mixture of unknown parameters of the normal distribution. The prior probability distribution for the parameters μ_1 and μ_2 is $N(0, 10^2)$, while the prior probability distribution for $\frac{1}{\sigma_1^2}$ and $\frac{1}{\sigma_2^2}$ is given by $\Gamma(0.5, 10)$. Since $\frac{1}{\sigma_1^2}$ and $\frac{1}{\sigma_2^2}$ are Gamma distributed, the parameters σ_1^2 and σ_2^2 are distributed with Inv-Gamma$(0.5, 10)$. Thus, the probability density function for the Inverse Gamma distributed parameters σ_1^2 and σ_2^2 is given by

$$f_{\mathrm{IG}}(x) = \frac{1}{10^{0.5}\Gamma(0.5)} x^{-1.5} e^{-\frac{1}{10x}}$$

All the prior distributions are independent. Thus, for the density function of the mixture normal distributed random variable X with δ as the mixing parameter, we have

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

Therefore, we obtain the following mixture distribution for X

$$f(x) = \delta \cdot \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} + (1-\delta) \cdot \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}$$

Next, we would like to compute the likelihood function for a sample of size n. To do that we are going to let $\mathbf{x}$ be a vector that stores n random variables distributed with the mixture normal distribution. We then have

$$L(\mathbf{x}|\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \prod_{i=1}^{n} \left(\delta \cdot \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + (1-\delta) \cdot \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}} \right)$$

Before we apply the Gibbs sampling, we have to obtain the posterior distribution for the parameters. Let θ be a vector such that $\theta = (\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$. Hence, we have

$$p(\theta|\mathbf{x}) \propto L(x|\theta) \cdot \phi(\mu_1|0, 10^2) \cdot \phi(\mu_2|0, 10^2) \cdot f_{\mathrm{IG}}(\sigma_1^2) \cdot f_{\mathrm{IG}}(\sigma_2^2)$$

*cosmin.borsa@uconn.edu; M.S. in Applied Financial Mathematics, Department of Mathematics, University of Connecticut.

We will now compute the log-posterior distribution

$$\log(p(\theta|x)) = \sum_{i=1}^{n} \log(f(x_i)) + \log(\phi(\mu_1|0, 10^2)) + \log(\phi(\mu_2|0, 10^2)) + \log(f_{\text{IG}}(\sigma_1^2)) + \log(f_{\text{IG}}(\sigma_2^2))$$

With the log-posterior distribution we can code the Gibbs sampling using the function `arms` in the R-package `HI`. Hoewever, before we do that we need to generate some data.

```
library('invgamma')
library('HI')

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)
```

Next, we will implement the log-posterior distribution.

```
logpost <- function(theta, x) {
  delta <- theta[1]
  mu.1 <- theta[2]
  mu.2 <- theta[3]
  sigma.1 <- theta[4]
  sigma.2 <- theta[5]
  return(sum(log(delta * dnorm(x, mu.1, sigma.1^0.5) + (1 - delta) *
         dnorm(x, mu.2, sigma.2^0.5))) + dnorm(mu.1, 0, 10, log = T) +
         dnorm(mu.2, 0, 10, log = T) +
         dinvgamma(sigma.1, shape = 0.5, scale = 10, log = T) +
         dinvgamma(sigma.2, shape = 0.5, scale = 10, log = T))
}
```

Now, we will code the Gibbs Sampling with using the `arms` function.

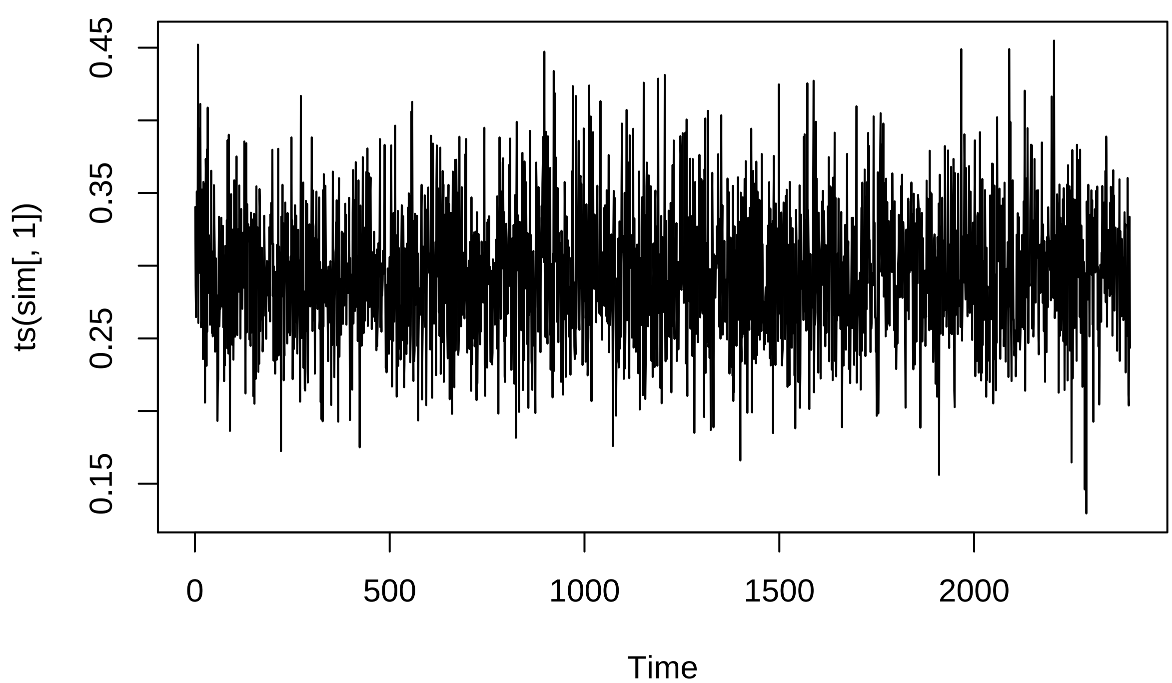
```
mymcmc <- function(niter, thetaInit, x, nburn= 100) {
  p <- length(thetaInit)
  thetaCurrent <- thetaInit
  logFC <- function(th, idx) {
    theta <- thetaCurrent
    theta[idx] <- th
    logpost(theta, x)
  }
  out <- matrix(thetaInit, niter, p, byrow = TRUE)
  ## Gibbs sampling
  for (i in 2:niter) {
    for (j in 1:p) {
    if (j == 1 | j == 4 | j == 5){
      out[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
```

```
                 function(x, idx) ((x > 0) * (x < 1)),
                 1, idx = j)
    } else if (j == 2 | j == 3) {
      out[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
                 function(x, idx) ((x > -50) * (x < 50)),
                 1, idx = j)
    }
    }
  }
  out[-(1:nburn), ]
}
```

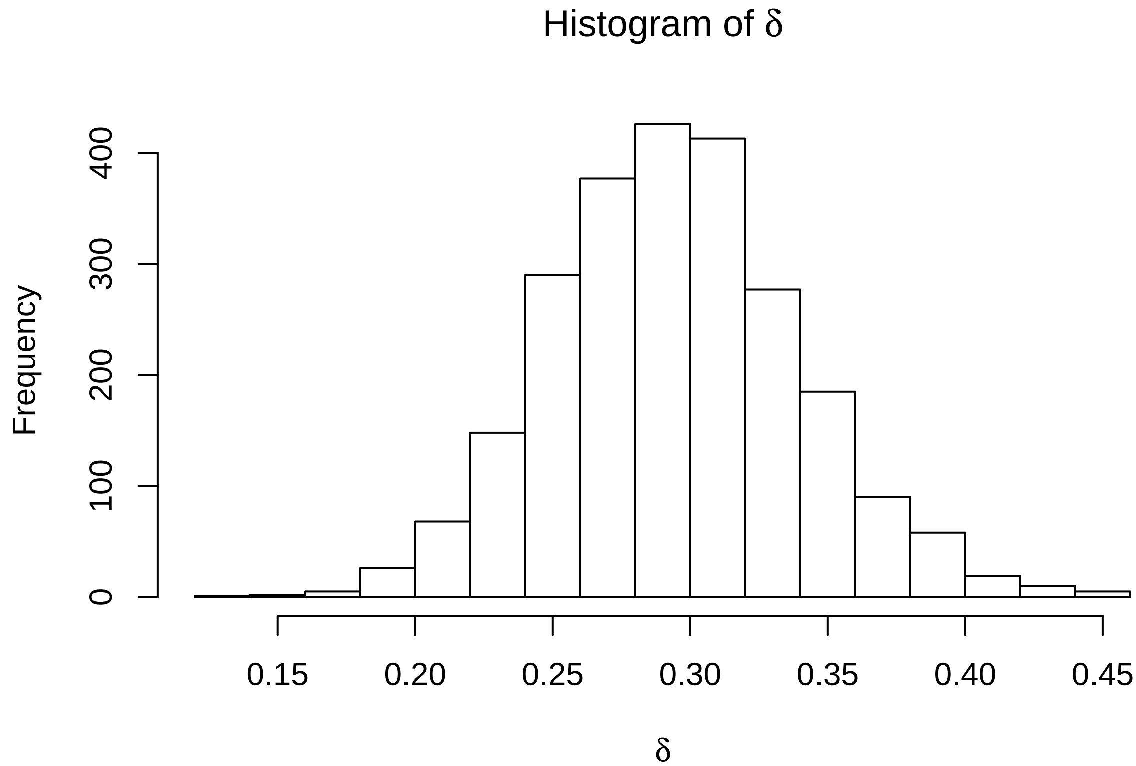
Last, we will plot the histogram of the results for all the parameters.

```
niter <- 2500
nburn <- 100
thetaInit <- c(0.3, 10, 7, 0.25, 0.25)
sim <- mymcmc(niter, thetaInit, x)

plot(ts(sim[,1]))
```

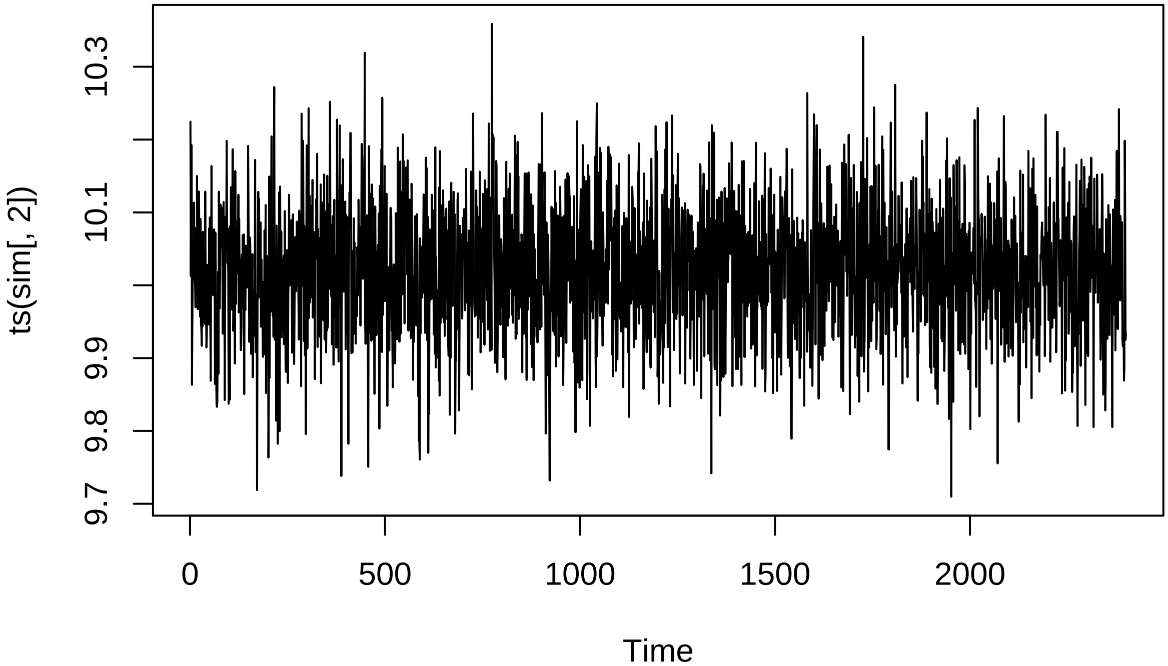



```
hist(sim[,1], main = expression(paste("Histogram of ", delta)),
      xlab = expression(paste(delta)))
```

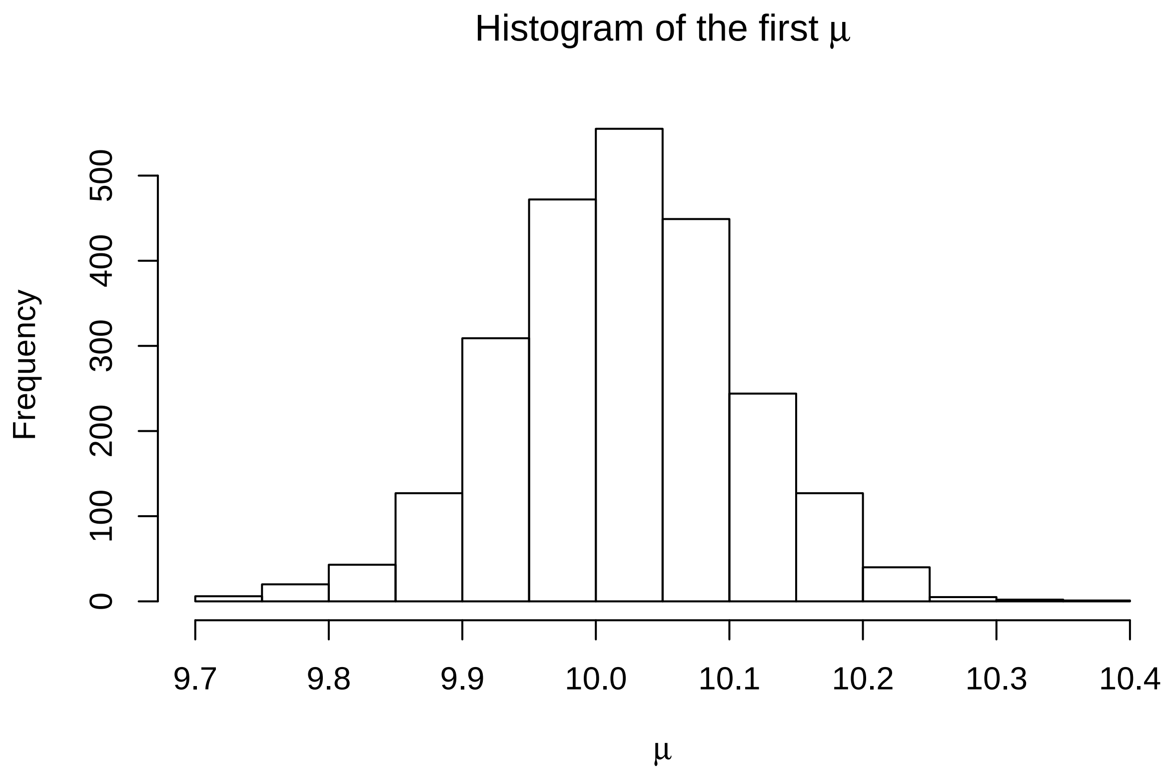


Histogram of δ
Frequency
0
100
200
300
400
0.15
0.20
0.25
0.30
0.35
0.40
0.45
δ

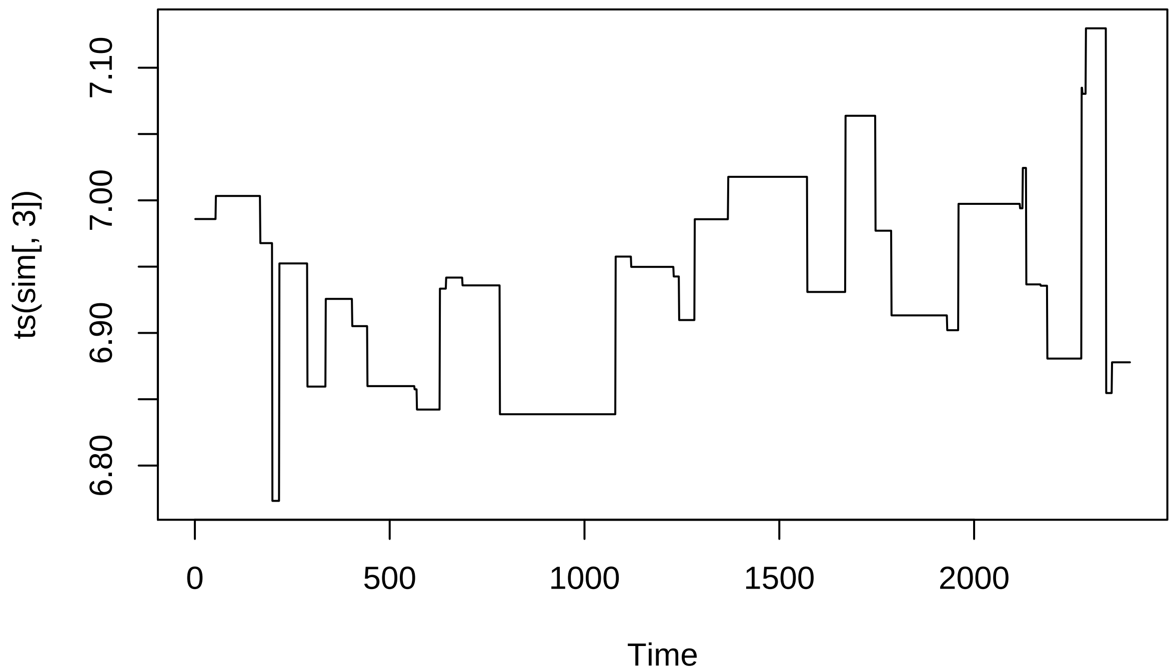

```
plot(ts(sim[,2]))
```



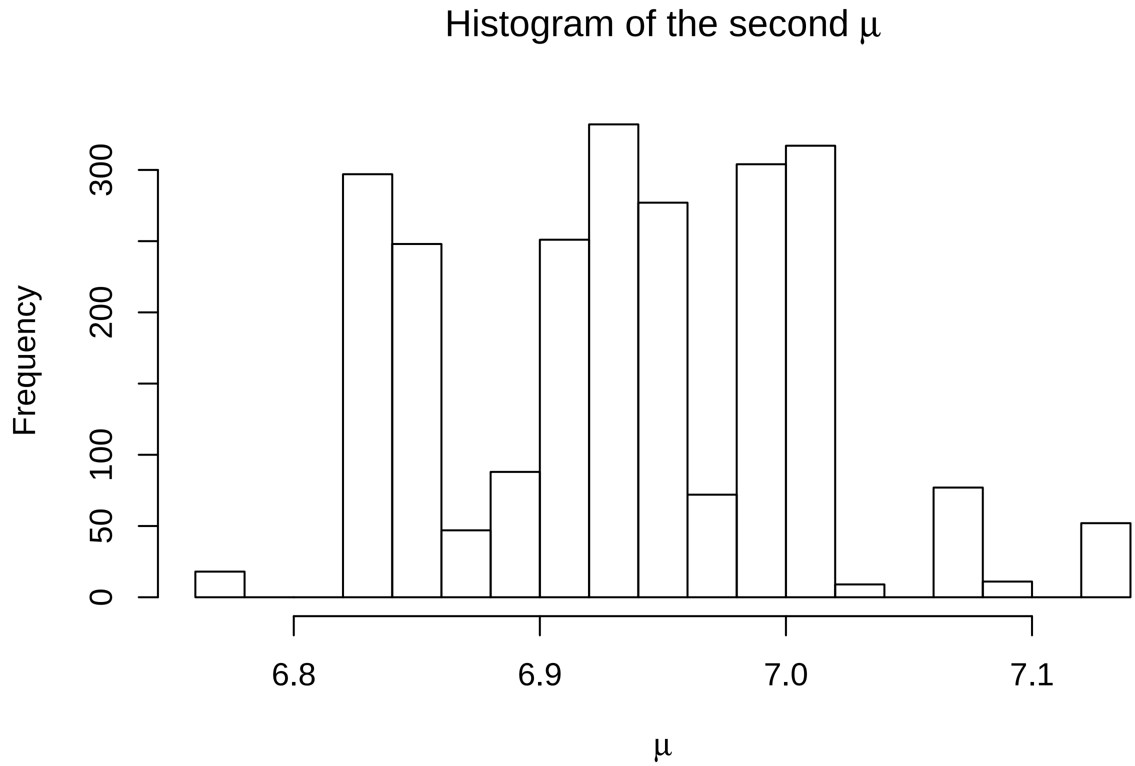

```r
hist(sim[,2], main = expression(paste("Histogram of the first ", mu)),
      xlab = expression(paste(mu)))
```



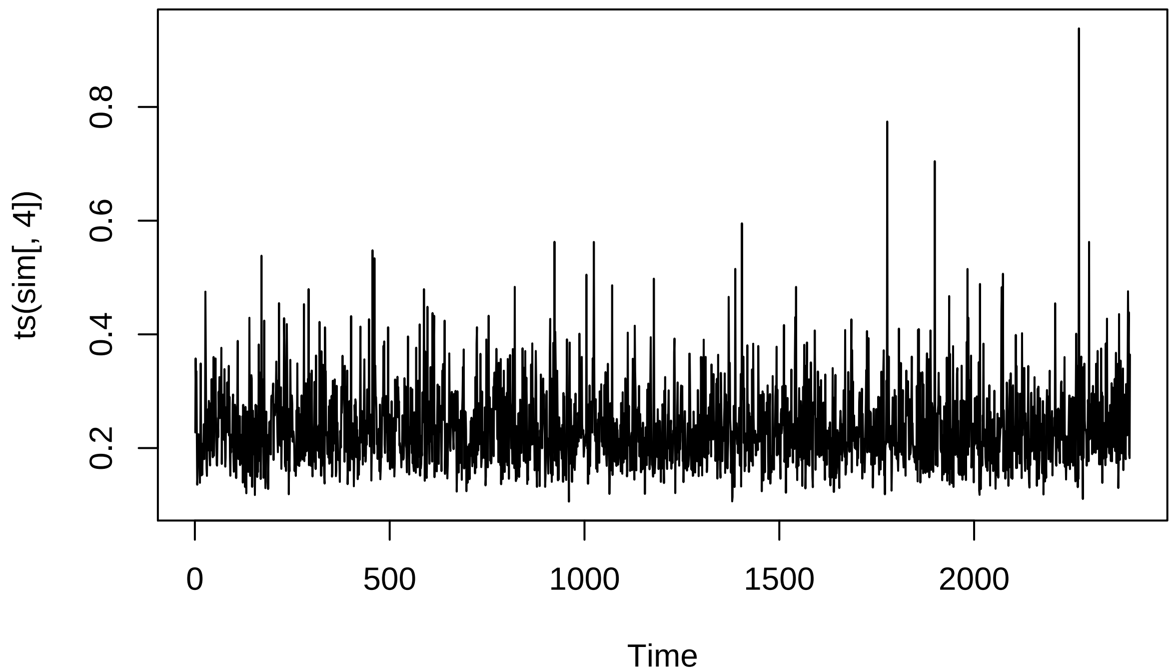

```r
plot(ts(sim[,3]))
```



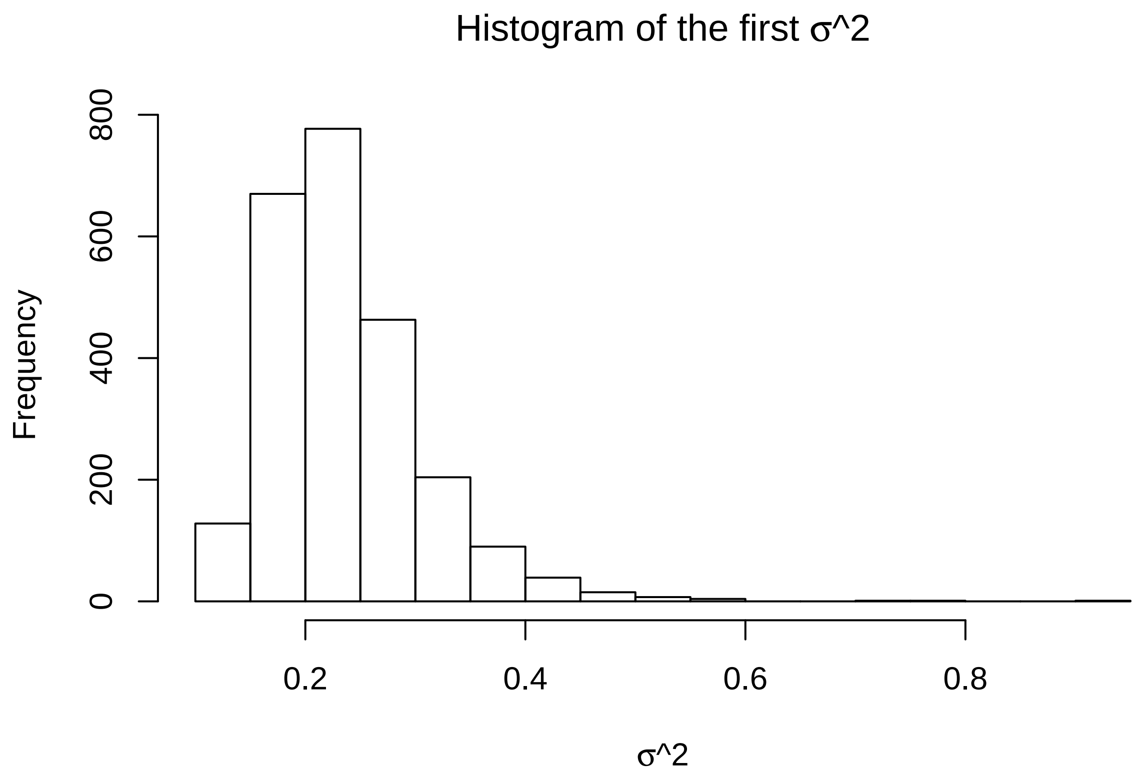

```r
hist(sim[,3], main = expression(paste("Histogram of the second ", mu)),
      xlab = expression(paste(mu)))
```


Histogram of the second μ
Frequency
0
50
100
200
300
6.8
6.9
7.0
7.1
μ

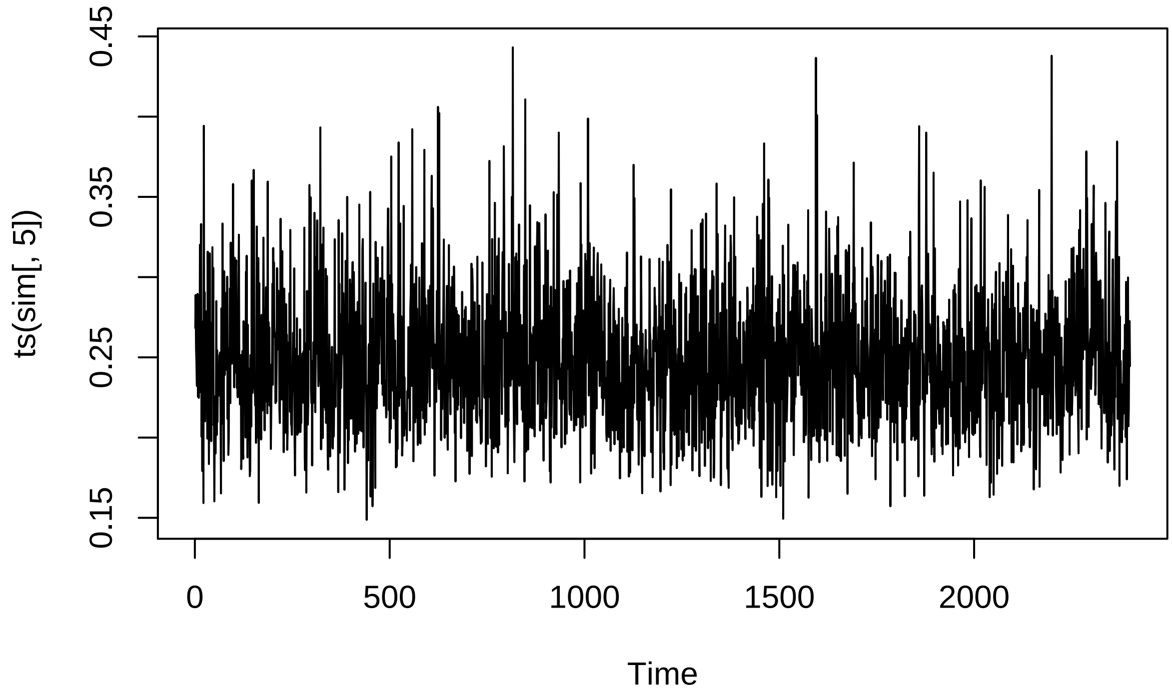
```
plot(ts(sim[,4]))
```



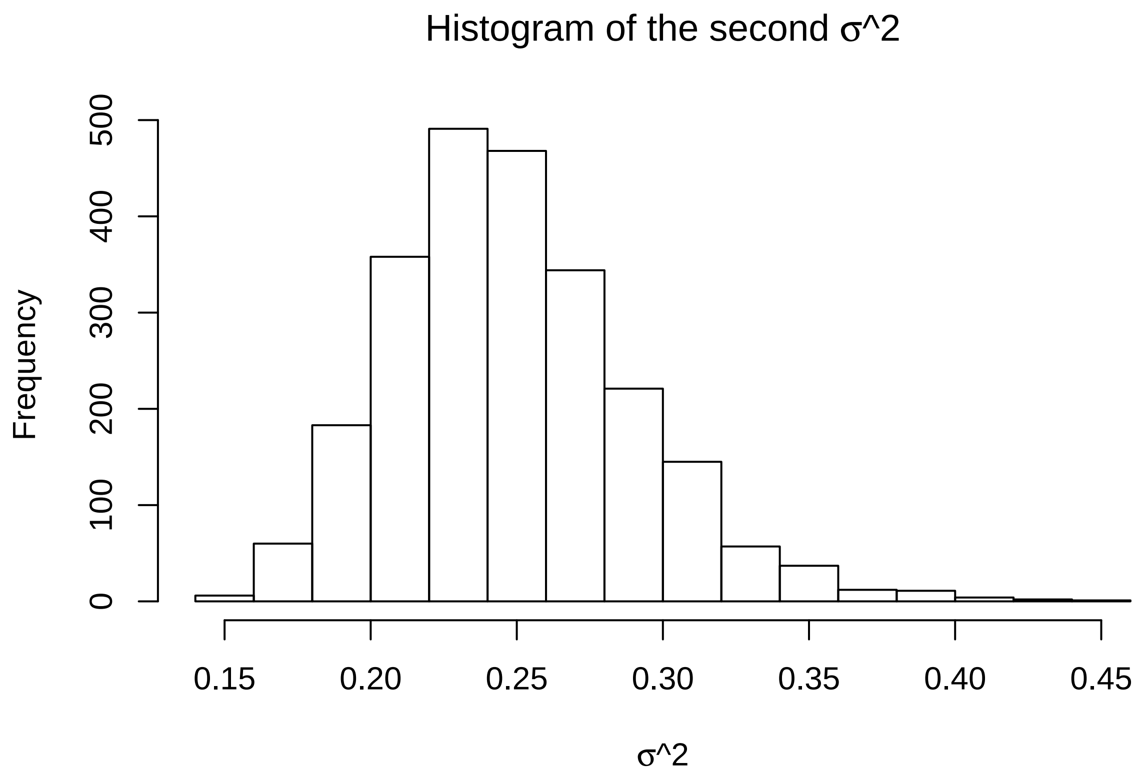

```
hist(sim[,4], main = expression(paste("Histogram of the first ", sigma, "^2")),
      xlab = expression(paste(sigma, "^2")))
```


Histogram of the first σ^2
Frequency
0
200
400
600
800
0.2
0.4
0.6
0.8
σ^2

```
plot(ts(sim[,5]))
```




```r
hist(sim[,5], main = expression(paste("Histogram of the second ", sigma, "^2")),
      xlab = expression(paste(sigma, "^2")))
```



From the histograms we can see that the estimated values of the vector θ gives us $\delta = 0.3$, $\mu_1 = 10$, $\mu_2 = 7$, $\sigma_1^2 = 0.25$, and $\sigma_2^2 = 0.25$. To incease the speed of the algorithm, we have choosen these values as the initial vector θ.

2 Acknowledgment

I would like to thank Professor Jun Yan for granting me a deadline extension for this homework assignment.