

HW7

Cheng Huang 2658312

26 October 2018

Normal Mixture revisite

Find posterior density

The likelihood function given $\delta, \mu_1, \mu_2, \sigma_1, \sigma_2$ is

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

The prior for μ_1, μ_2 are normal $N(0, 10^2)$, so

$$\pi(\mu_1) \propto \exp\left(-\frac{\mu_1^2}{200}\right)$$

$$\pi(\mu_2) \propto \exp\left(-\frac{\mu_2^2}{200}\right)$$

The prior for σ_1^2, σ_2^2 are $IG(0.5, 10)$, so

$$\pi(\sigma_1^2) \propto (\sigma_1^2)^{-1.5} \exp\left(-\frac{10}{\sigma_1^2}\right)$$

$$\pi(\sigma_2^2) \propto (\sigma_2^2)^{-1.5} \exp\left(-\frac{10}{\sigma_2^2}\right)$$

Therefore the posterior distribution propotional to

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

MCMC

```
library("invgamma")
library("HI")
n <- 100
## set true value of parameters
delta <- 0.7
mu1 <- 5
mu2 <- 8
```

```

sigmasq1 <- .3
sigmasq2 <- .9

## simulation
set.seed(123)
u <- rbinom(n, prob = delta, size = 1)
d <- rnorm(n, ifelse(u == 1, mu1, mu2), ifelse(u == 1, sigmasq1, sigmasq2))

## calculate the posterior distribution
logposterior <- function(delta, mu1, mu2, sigmasq1, sigmasq2, x = d) {
  logL <- sum(log(delta * dnorm(x, mu1, sqrt(sigmasq1)) + (1-delta)*dnorm(x, mu2, sqrt(sigmasq2))
  logprior.mu1 <- dnorm(mu1, 0, 10, log = T)
  logprior.mu2 <- dnorm(mu2, 0, 10, log = T)
  logprior.sigma1 <- dinvgamma(sigmasq1, 0.5, 10, log = T)
  logprior.sigma2 <- dinvgamma(sigmasq2, 0.5, 10, log = T)

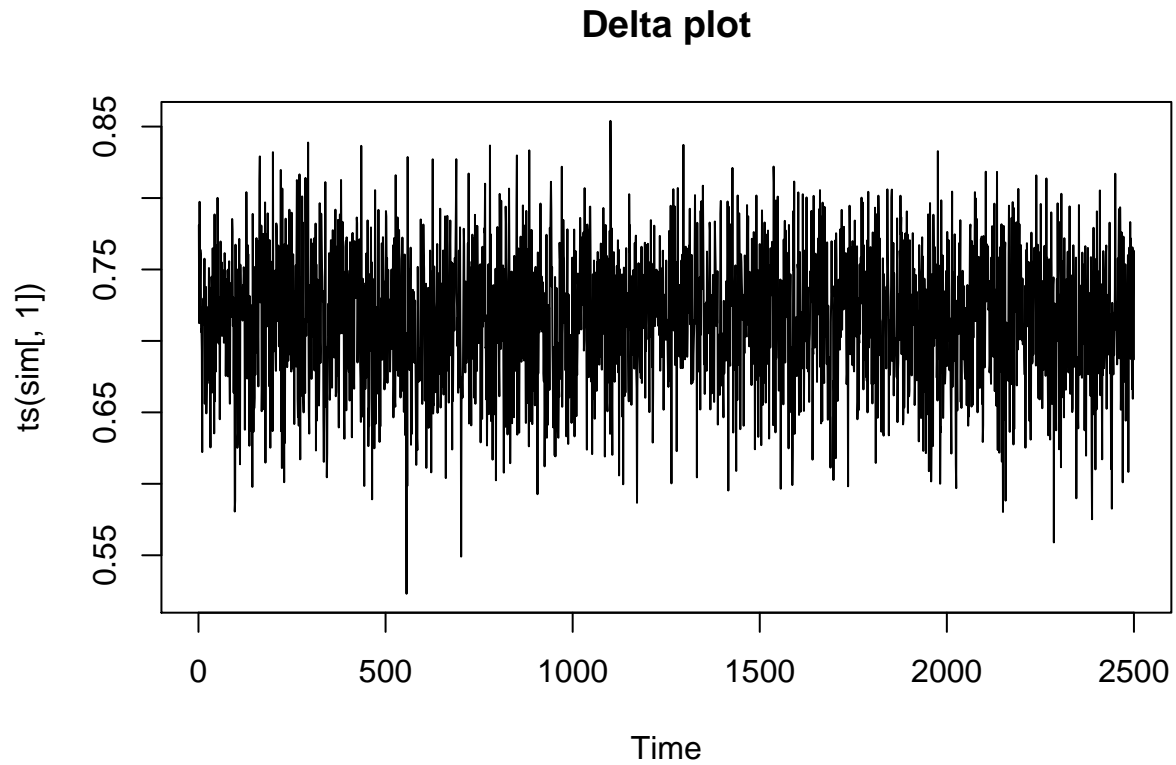
  return(logL + logprior.mu1 + logprior.mu2 + logprior.sigma1 + logprior.sigma2)
}

mymcmc <- function (niter, thetaInit, x = d){
  p <- length(thetaInit)
  thetaCurrent <- thetaInit
  out <- matrix(NA, niter, p)
  result <- matrix(NA, niter, p)
  for (i in 1:niter) {
    for (j in 1:p) {
      logFC <- function(thj) {
        theta <- thetaCurrent
        theta[j] <- thj
        logposterior(theta[1], theta[2], theta[3], theta[4], theta[5], d)
      }
      sp <- function(x){
        if (j==1 || j==4 || j==5) {
          ((x > 0) * (x < 1))
        } else if(j==2 || j==3 ){
          ((x > 0) * (x < 20))
        }
      }
      result[i, j] <- thetaCurrent[j] <-
        HI::arms(thetaCurrent[j], logFC,
                  sp, 1)
    }
  }
  result
}

niter <- 2500
thetaInit <- c(0.5, 5, 5, 0.5, 0.5)

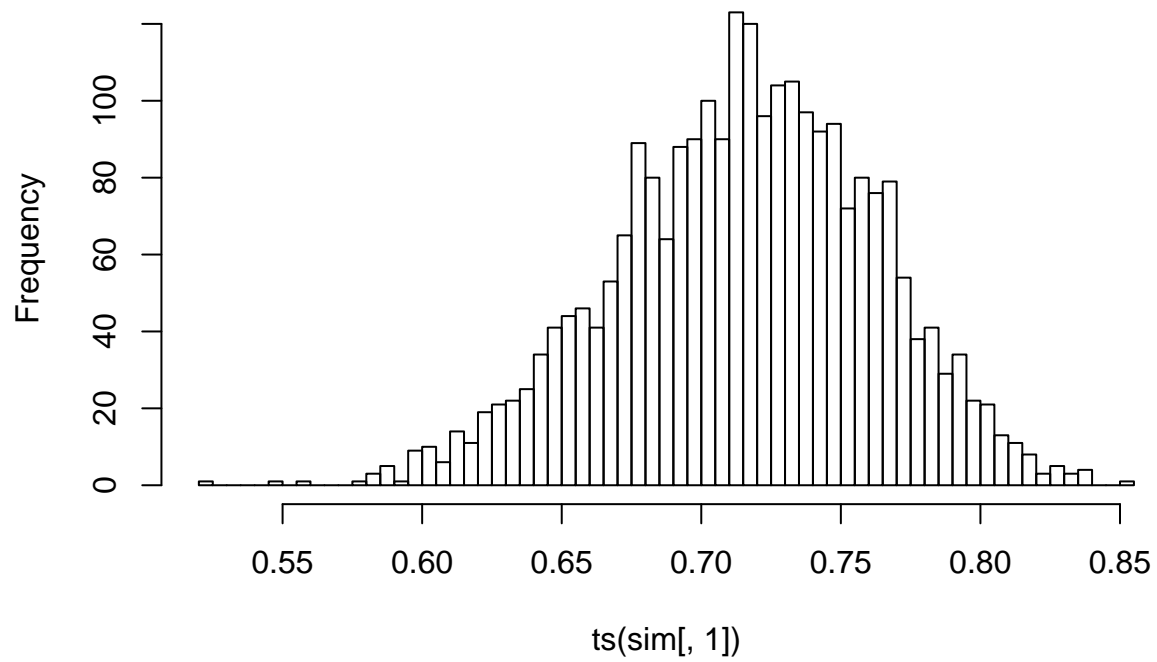
```

```
sim <- mymcmc(niter, thetaInit, d)
plot(ts(sim[,1]), main = "Delta plot")
```



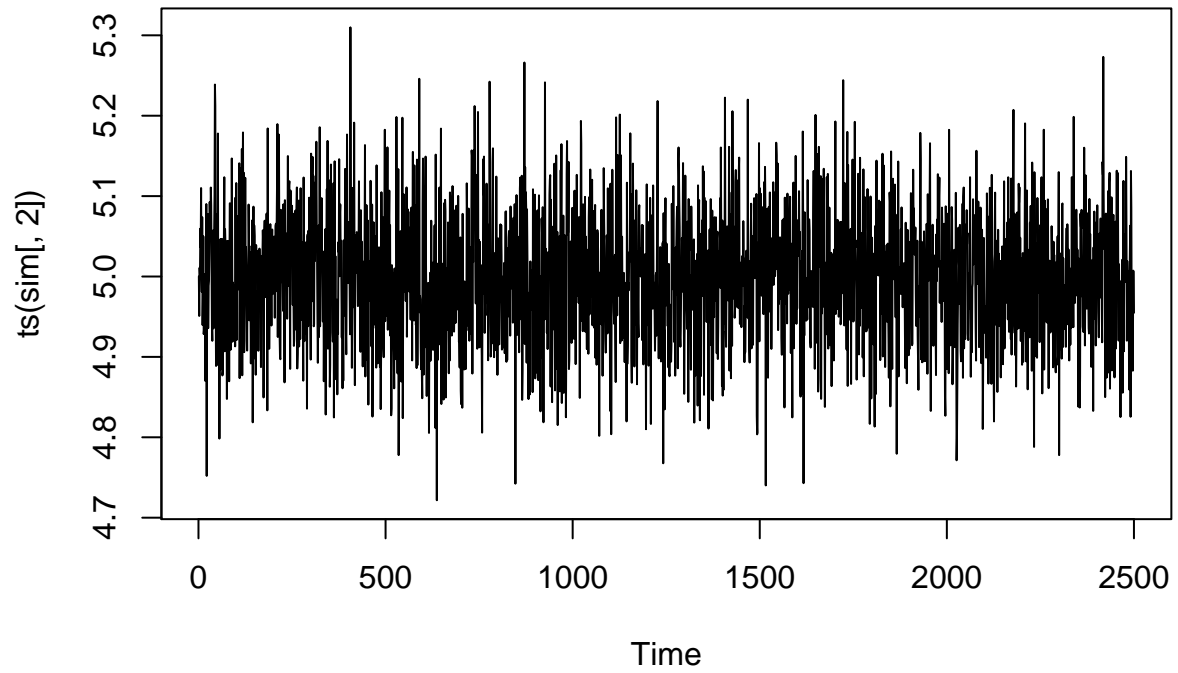
```
hist(ts(sim[,1]), breaks = 50, main = "Histogram for Delta")
```

Histogram for Delta

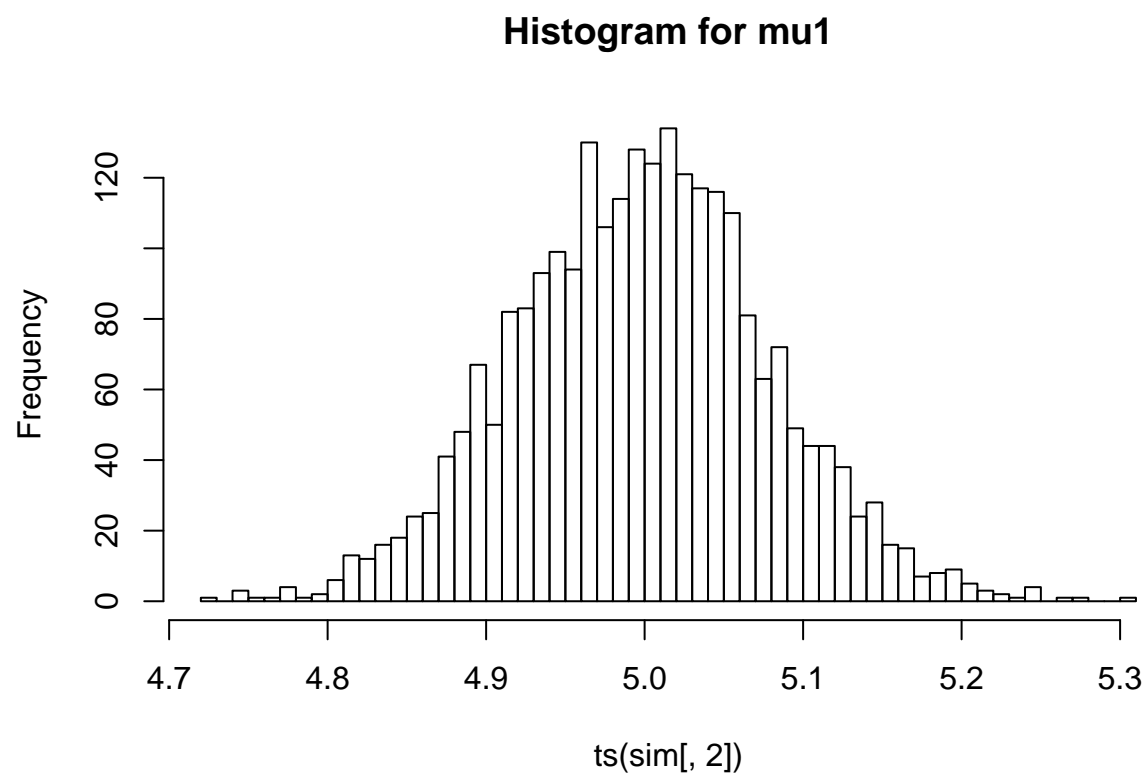


```
plot(ts(sim[,2]), main = "mu1 plot")
```

mu1 plot

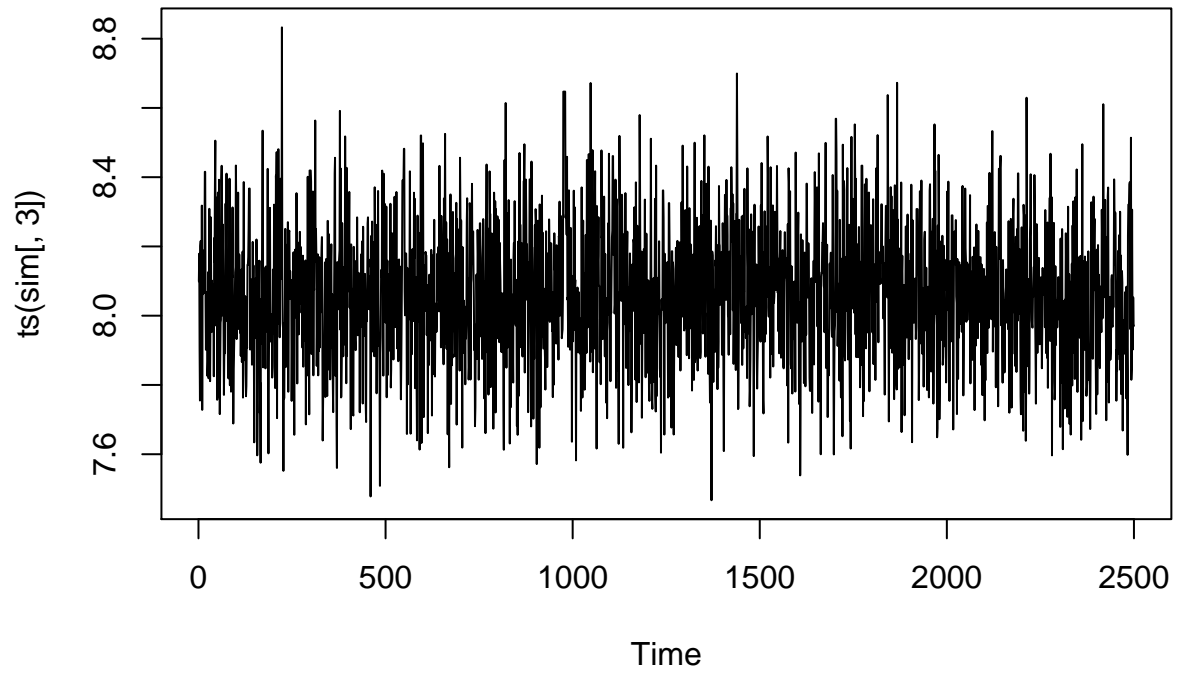


```
hist(ts(sim[,2]), breaks = 50, main = "Histogram for mu1")
```



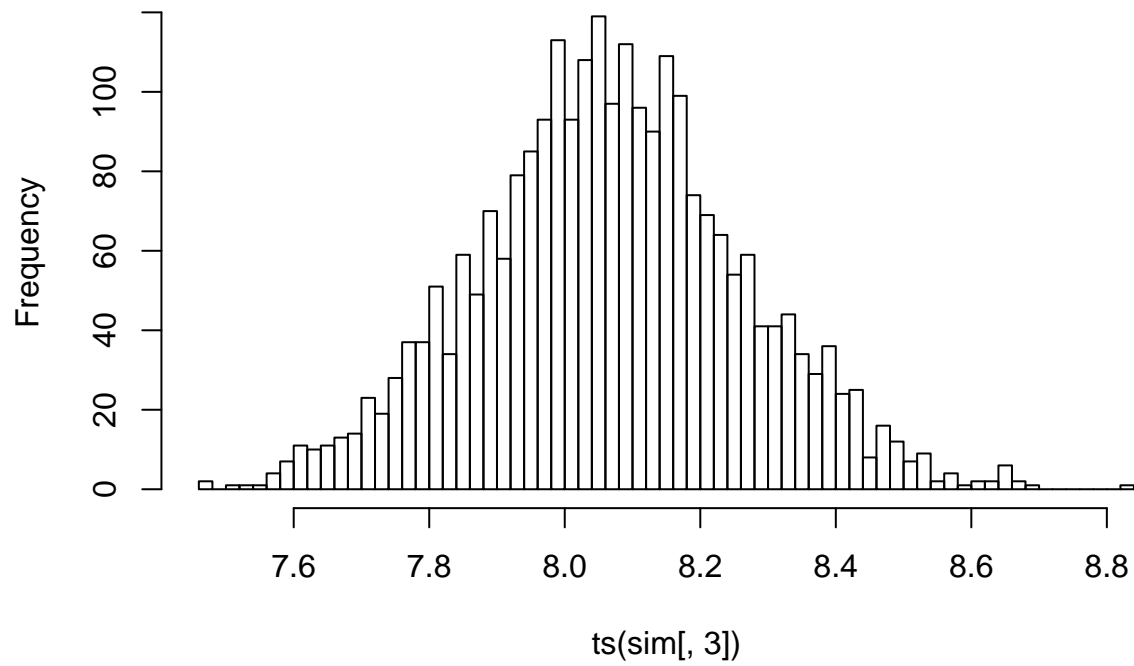
```
plot(ts(sim[,3]), main = "mu2 plot")
```

mu2 plot

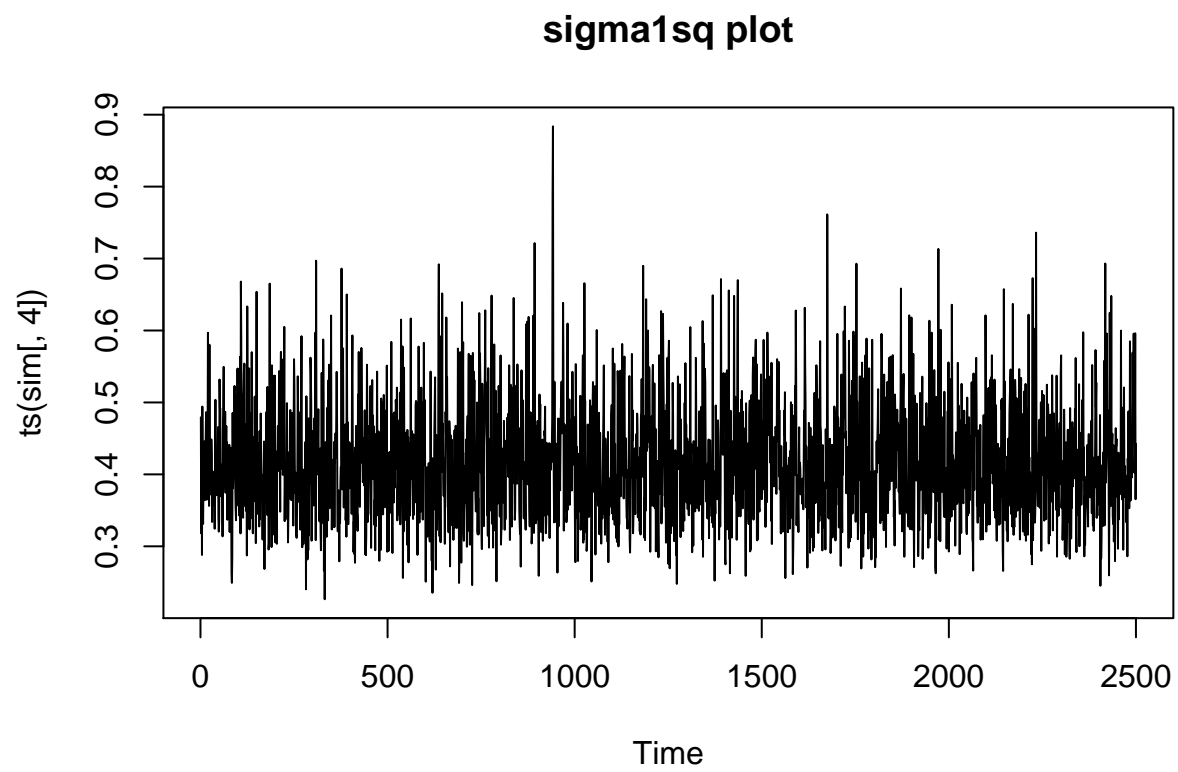


```
hist(ts(sim[,3]), breaks = 50, main = "Histogram for mu2")
```

Histogram for mu2

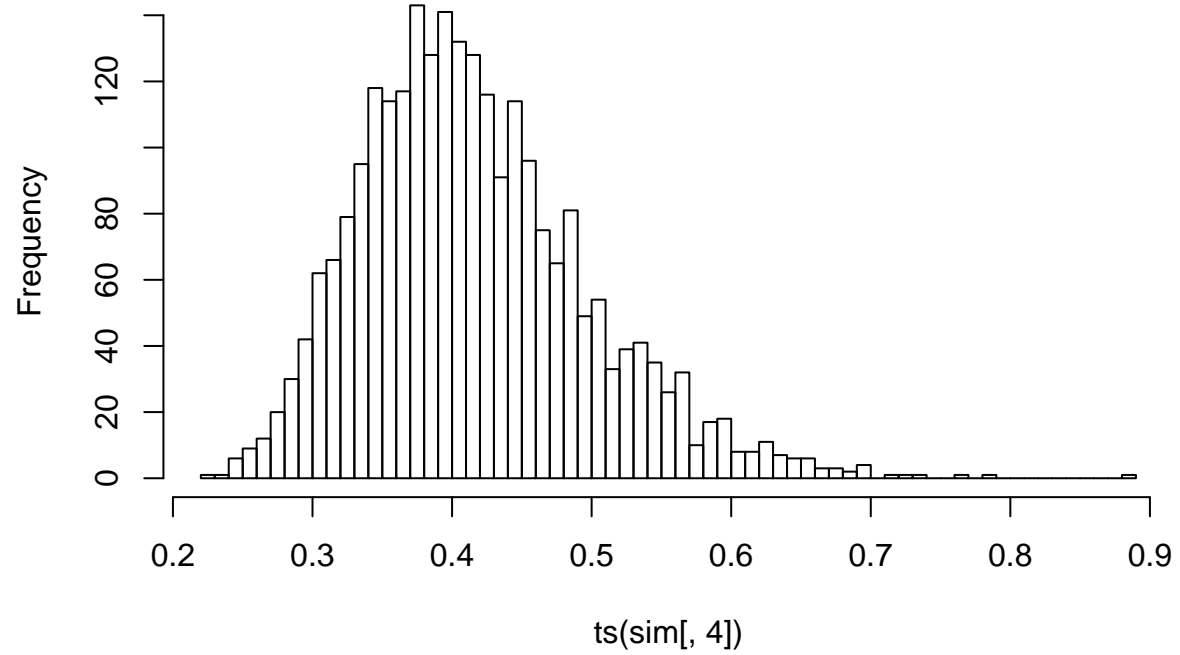


```
plot(ts(sim[,4]), main = "sigma1sq plot")
```

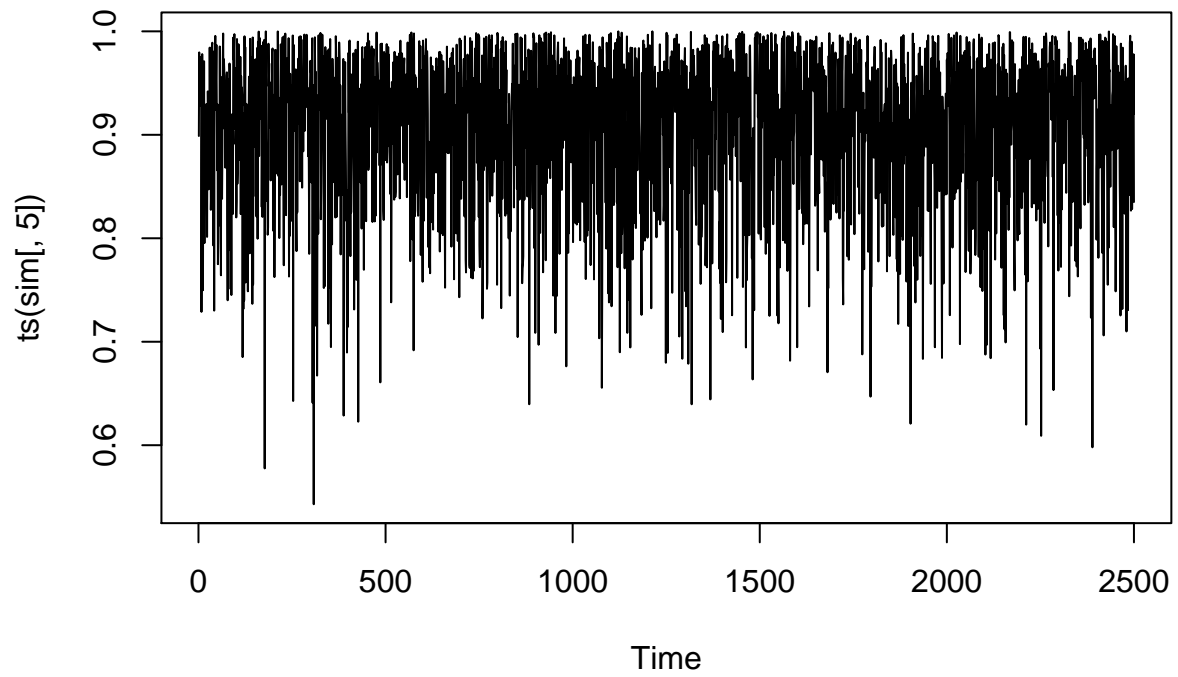
```
hist(ts(sim[,4]), breaks = 50, main = "Histogram for sigma1 square")
```

Histogram for sigma1 square



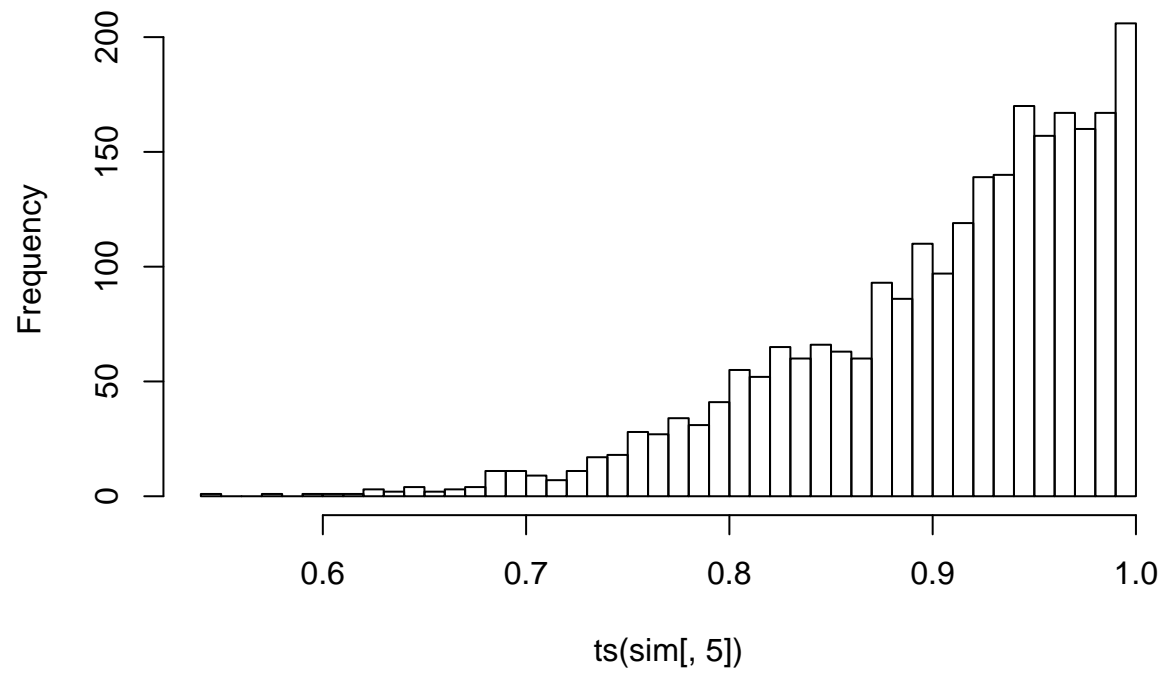
```
plot(ts(sim[,5]), main = "sigma2sq plot")
```

sigma2sq plot



```
hist(ts(sim[,5]), breaks = 50, main = "Histogram for sigma2 square")
```

Histogram for sigma2 square



Reference

[jun-yan/stat-5361]<https://github.com/jun-yan/stat-5361>