HW#7

Zijian Huang

10/23/2020

Ex 5.3.1

(a)

$$\begin{aligned} C\int_0^\infty (2x^{\theta-1}+x^{\theta-1/2})e^{-x}dx &= C\int_0^\infty 2x^{\theta-1}e^{-x}dx + C\int_0^\infty x^{\theta-1/2}e^{-x}dx \\ &= 2C\Gamma(\theta)\int_0^\infty Gamma(\theta,1)dx + C\Gamma(\theta+\frac{1}{2})\int_0^\infty Gamma(\theta+\frac{1}{2},1)dx \\ &= 2C\Gamma(\theta)+C\Gamma(\theta+\frac{1}{2}) \\ &= 1 \end{aligned}$$

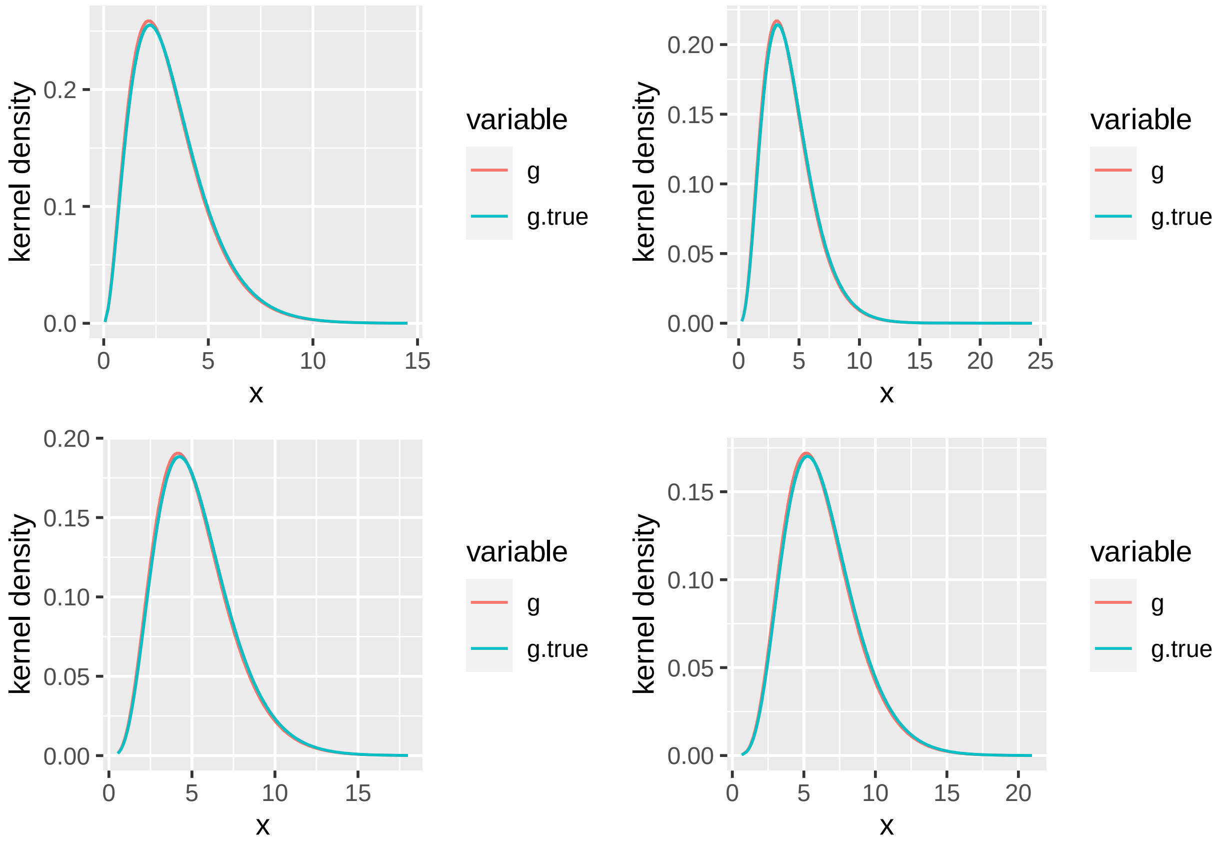
$$\text{Thus we have} \Rightarrow C = \frac{1}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}$$

$$g(x) = \frac{2}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}Gamma(\theta,1) + \frac{1}{2\Gamma(\theta)+\Gamma(\theta+\frac{1}{2})}Gamma(\theta+\frac{1}{2},1)$$

Therefore, the component distributions are $Gamma(\theta,1)$ and $\Gamma(\theta+\frac{1}{2},1)$ and their weights in the mixture are 2/3 and 1/3 respectively.

(b)

```r
set.seed(123)
n <- 10000
library(tidyverse)
library(reshape2)
sample_g <- function(theta){
  c <- 1/(2 * gamma(theta) + gamma(theta + 0.5))
  par1 <- theta; par2 <- theta + 0.5
  u <- rbinom(n, size = 1, prob = 1/3)
  x <- rgamma(n, shape = ifelse(u == 1, par1, par2), scale = 1)
  g <- 2/3 * dgamma(x, shape = par1, scale = 1) +
  1/3 * dgamma(x, shape = par2, scale = 1)
  g.true <- c * exp(-x) * (2 * x^(theta - 1) + x^(theta - 0.5))
  df <- data.frame(x, g, g.true)
  df1 <- melt(df, id.vars="x")
  # Everything on the same plot
  plot.g <- ggplot(df1, aes(x = x, y = value, group = variable, colour = variable)) +
    geom_line() +
    ylab("kernel density")
  return(plot.g)
}
# apply theta=2, 3, 4, 5, 6 respectively
library(cowplot)
plot_grid(sample_g(3), sample_g(4), sample_g(5), sample_g(6), cols=2)
```



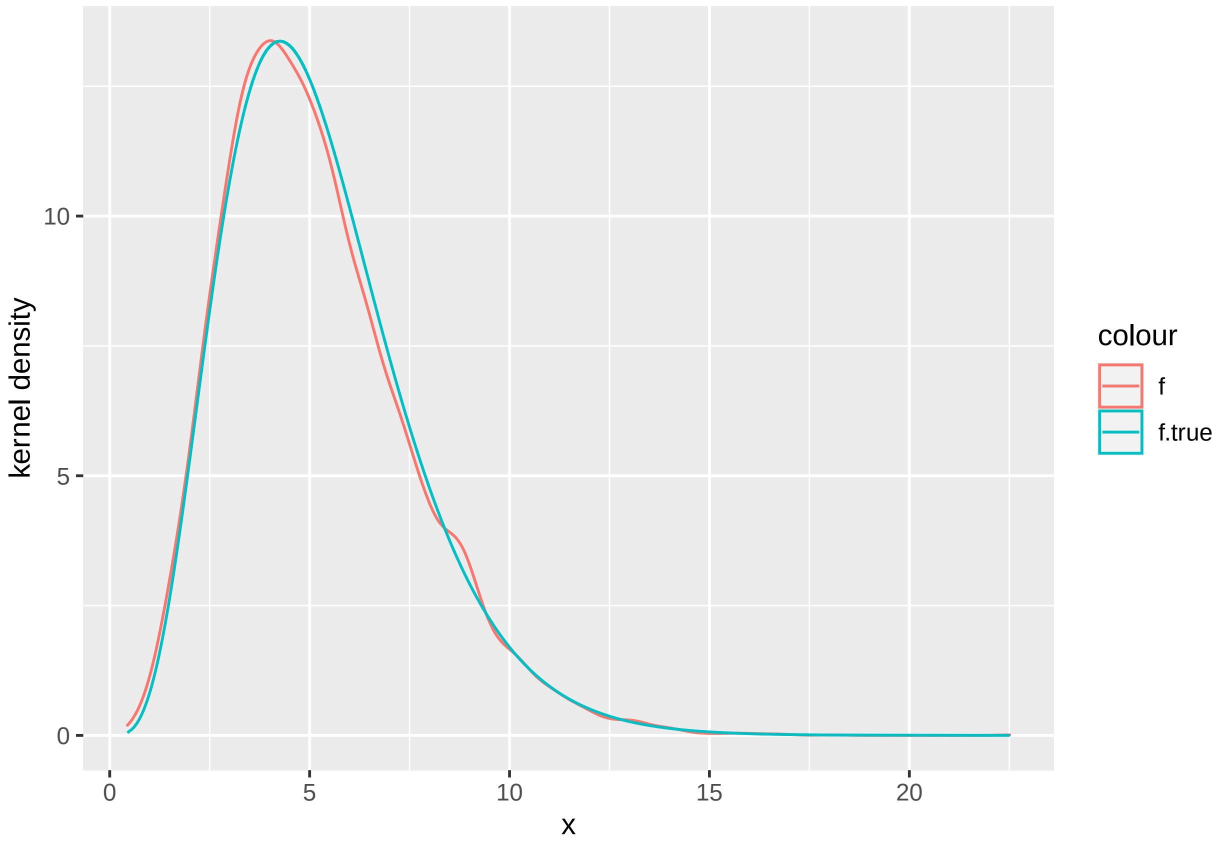
Here I take the $\theta = 3, 4, 5, 6$ respectively. The sampling density matches best with the true density of g at $\theta = 5$. However, the other plots still have the similar shape with the true density of g. As θ increase, the sampling plot seems more skew to the left and have higher peak.

(c)

$$\frac{1}{\sqrt{2}} \le ratio = \frac{f(x)}{g(x)} = \frac{\sqrt{4+x}}{2+\sqrt{x}} < 1 \text{ , for x} > 0$$

It means the value of g(x) will always above f(x).

```
gam.rej <- function(n, theta){
  x <- rep(NA, n)
  par1 <- theta; par2 <- theta + 0.5
  for (i in 1:n) {
    while (TRUE) {
      u <- rbinom(1, prob = 2/3, size = 1)
      cand <- rgamma(1, shape = ifelse(u == 1, par1, par2), scale = 1)
      ratio <- sqrt(cand + 4)/(2 + sqrt(cand))
      uni <- runif(1)
      if (uni < ratio) break
    }
    x[i] <- cand
  }
  return(x)
}
# From b, we know when theta=5, the plot match best
theta <- 5
x <- gam.rej(n, theta)
fx <- function(x, theta) sqrt(x + 4) * x^(theta - 1) * exp(-x)
df2 <- data.frame(x, fx(x, theta))

ggplot(data = df2, aes(x = x)) +
  geom_density(aes(y=..density.. * 71, colour = "f")) +
  geom_line(aes(y = fx(x, theta), colour = 'f.true')) +
  ylab("kernel density")
```



Ex 6.3.1

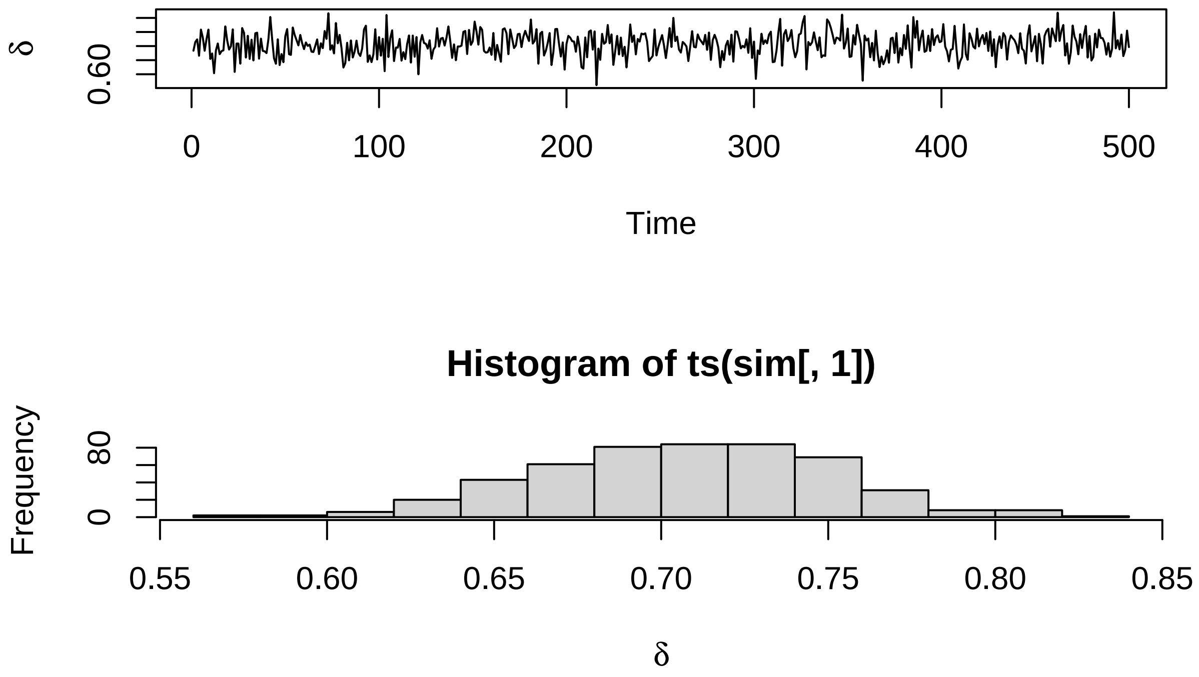
Since it is normal mixture, then the likelihood is:

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

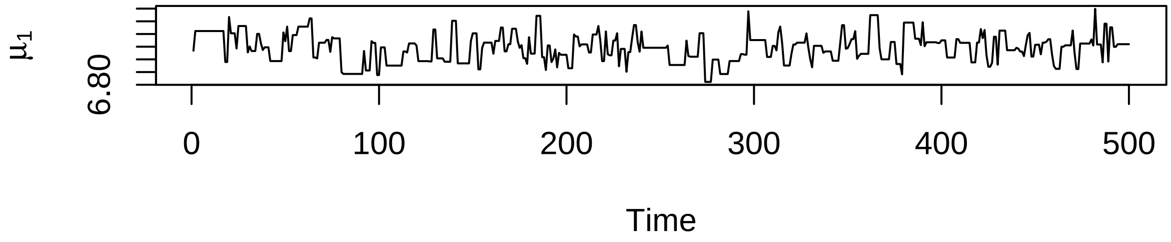
Because the priors are independent, then the posterior is:

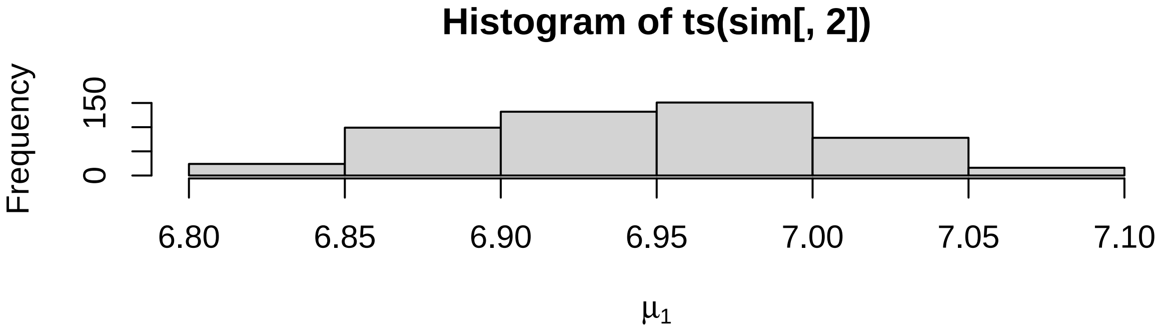
$$\begin{aligned} f(\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2|x) &= f(x|\delta, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) * [\pi(\delta)\pi(\mu_1)\pi(\mu_2)\pi(\sigma_1^2)\pi(\sigma_2^2)] \\ &\propto [\frac{\delta}{\sigma_1} exp(\frac{(x-\mu_1)^2}{2\sigma_1^2}) + \frac{\delta}{\sigma_1} exp(\frac{(x-\mu_1)^2}{2\sigma_1^2})][\pi(\delta)\pi(\mu_1)\pi(\mu_2)\pi(\sigma_1^2)\pi(\sigma_2^2)] \end{aligned}$$

```r
library(HI)
library(invgamma)
delta <- 0.7 # true value to be estimated based on the data
n <- 100
set.seed(123)
u <- rbinom(n, prob = delta, size = 1)
x <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)

logpost <- function(theta, x) {
  delta <- theta[1]
  mu1 <- theta[2]; mu2 <- theta[3]
  sig1 <- theta[4]; sig2 <- theta[5]
  loglike <- sum(log(delta * dnorm(x, mu1,sqrt(sig1)) +
                  (1-delta) * dnorm(x,mu2,sqrt(sig2)))) +
                  log(dnorm(mu1,0,10) ) + log(dnorm(mu2,0,10) ) +
                  log(dinvgamma(sig1, 0.5 , 0.1) ) +
                  log(dinvgamma(sig2, 0.5 , 0.1) )
  return(loglike)
}
mymcmc <- function(niter, thetaInit, x, nburn= 100) {
  p <- length(thetaInit)
  thetaCurrent <- thetaInit
## define a function for full conditional sampling
  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) {
## general-purpose arms algorithm
      out[i, j] <- thetaCurrent[j] <- if (j < 2) {
        HI::arms(thetaCurrent[j], logFC,
                  function(x, idx) ((x > 0) * (x < 1)), 1, idx = j)
      } else if (j < 4 ) {
        HI::arms(thetaCurrent[j], logFC,
                 function(x, idx) ((x > -30) * (x < 30)), 1, idx = j)
      } else {
        HI::arms(thetaCurrent[j], logFC,
                 function(x, idx) ((x > 0) * (x < 100)), 1, idx = j)
      }
    }
  }
  out[-(1:nburn), ]
}

niter <- 600; nburn <- 100
thetaInit <- c(0.5, 2, 2, 2, 2)
sim <- mymcmc(niter, thetaInit, x)
par(mfrow=c(2,1))     # set the plotting area into a 1*2 array
plot(ts(sim[,1]), ylab = expression(delta))
hist(ts(sim[,1]), xlab = expression(delta))
```

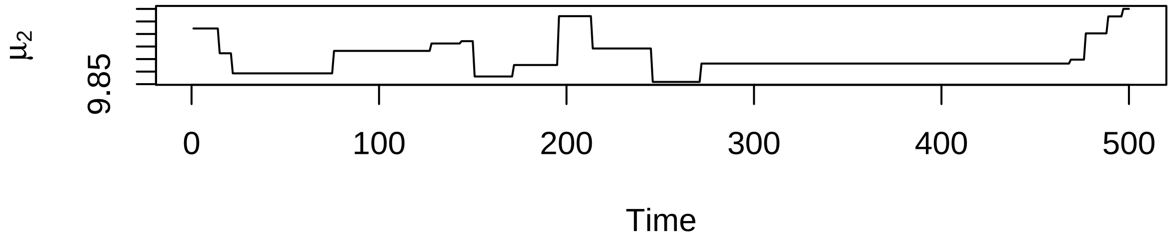


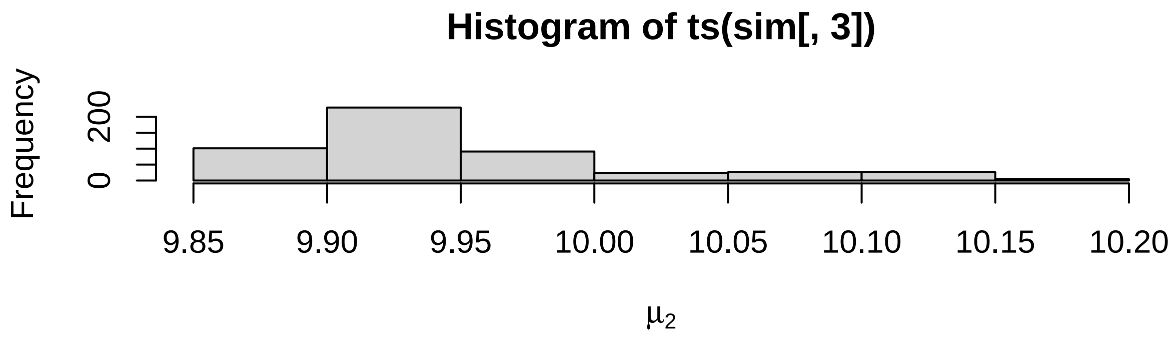
```r
plot(ts(sim[,2]), ylab = expression(mu[1]))
hist(ts(sim[,2]), xlab = expression(mu[1]))
```



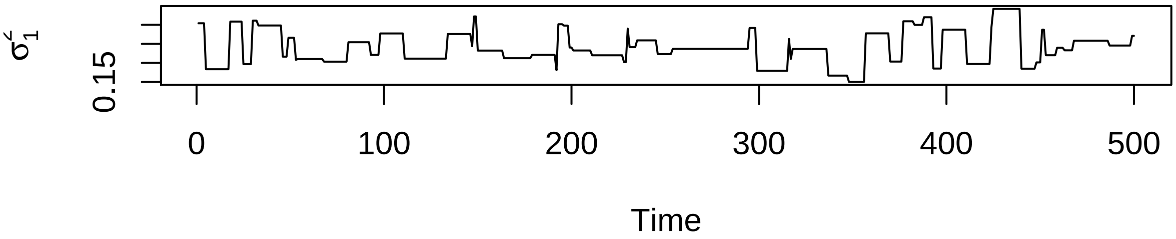


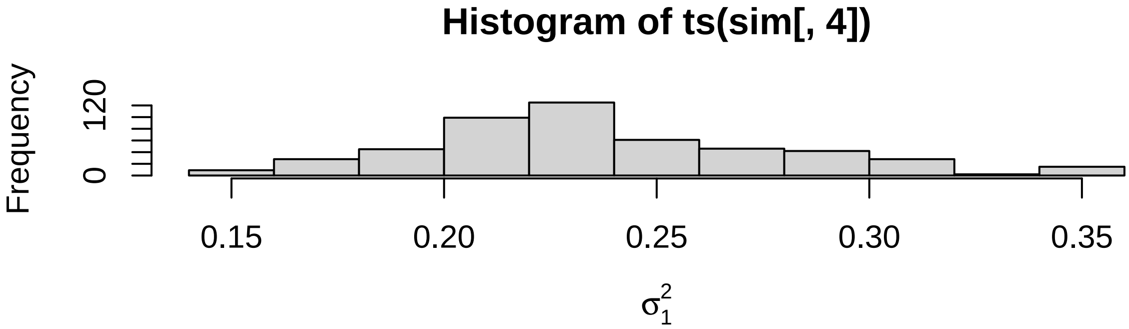
```r
plot(ts(sim[,3]), ylab = expression(mu[2]))
hist(ts(sim[,3]), xlab = expression(mu[2]))
```



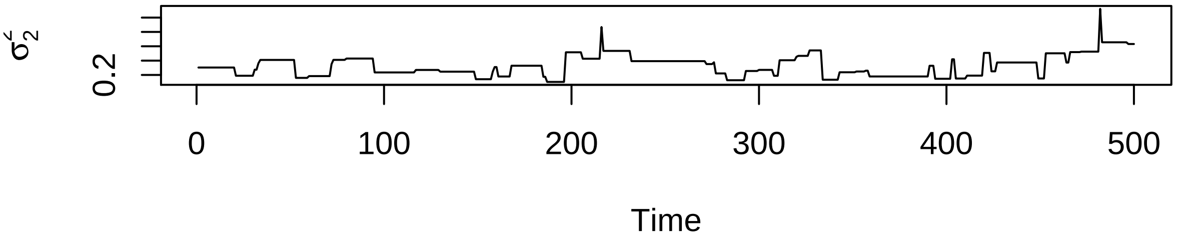


```r
plot(ts(sim[,4]), ylab = expression(sigma[1]^{2}))
hist(ts(sim[,4]), xlab = expression(sigma[1]^{2}))
```

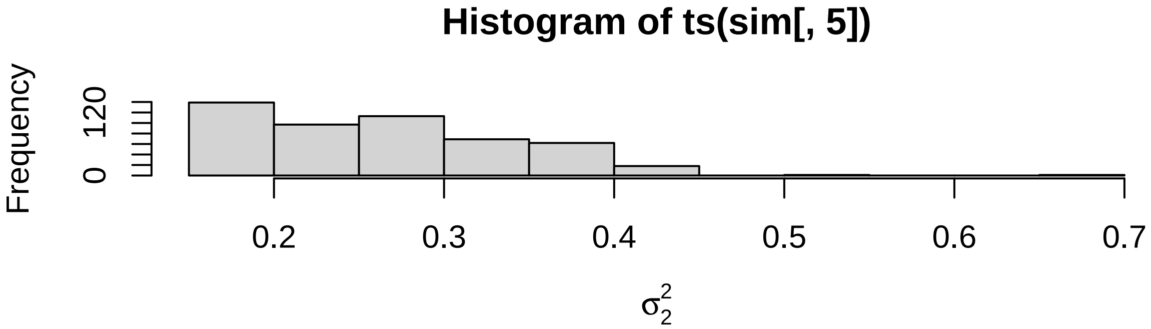




```r
plot(ts(sim[,5]), ylab = expression(sigma[2]^{2}))
hist(ts(sim[,5]), xlab = expression(sigma[2]^{2}))
```



σ_2^2
0.2
0
100
200
300
400
500
Time



Histogram of ts(sim[, 5])
Frequency
0
120
0.2
0.3
0.4
0.5
0.6
0.7
σ_2^2