HW 8

Problem 1

(1)

The integral is

$$
\begin{aligned}
&\int_{-\infty}^{\infty} p(\theta|y)\; d\theta \\
\propto &\int_{-\infty}^{\infty} \prod_{i=1}^{2} \frac{1}{\pi\left\{1+(y_i-\theta)^2\right\}}\; d\theta
\end{aligned}
$$

It is of order θ^{-4} so if the inverse temperature is $1/4$ the integral will be divergent.

(2)

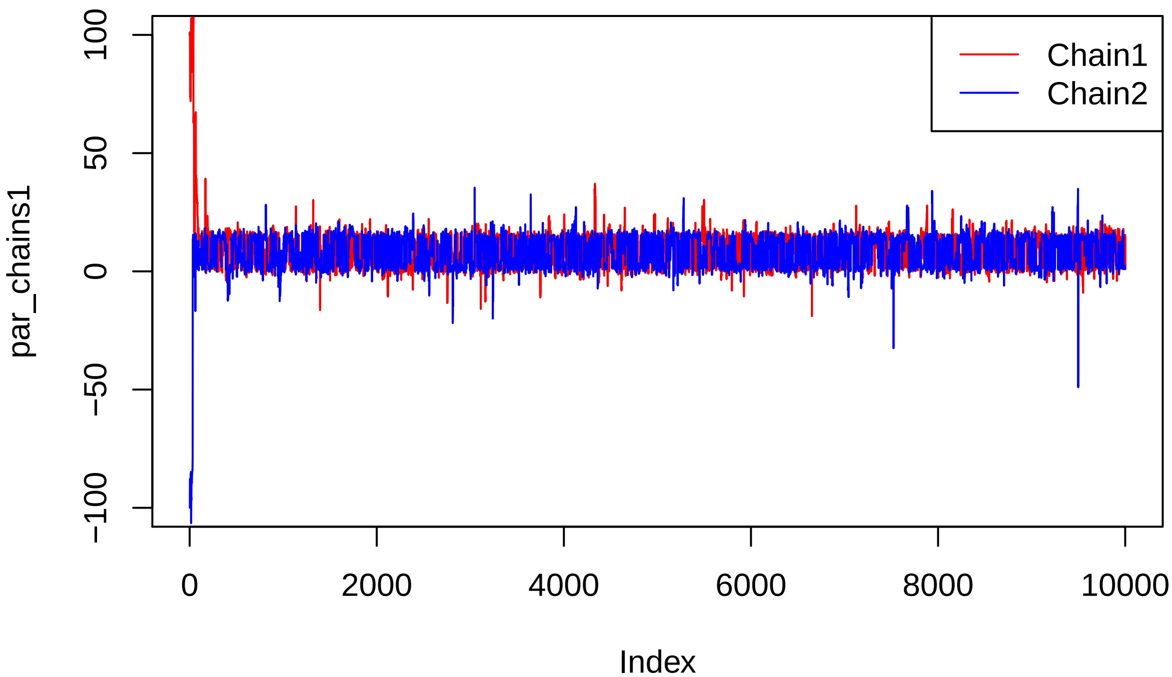
```
set.seed(1)
y1 <- 1.3
y2 <- 15.0
inv_Temper <- seq(1, .3, -.1)

post <- function(theta, inv_temper) {
  (1/pi^2/(1+(y1-theta)^2)/(1+(y2-theta)^2))^inv_temper
}

normalize <- function(inv_temper) {
  1/integrate(post, -Inf, Inf, inv_temper)$value
}

normalize_consts <- sapply(inv_Temper,  normalize)

post <- function(theta, inv_temper) {
  normalize_consts[which(inv_Temper == inv_temper)] * (1/pi^2/(1+(y1-theta)^2)/(1+(y2-theta)^2))^inv_ter
}
```

Let the jump distribution be $J_t(a|b) = \frac{1}{\pi\gamma}\frac{1}{1+(a-b)^2/\gamma^2}$. Obviously $J_t(a|b) = J_t(b|a)$ for all a, b since $(a-b)^2 = (b-a)^2$. The scale parameter γ is of our choice.

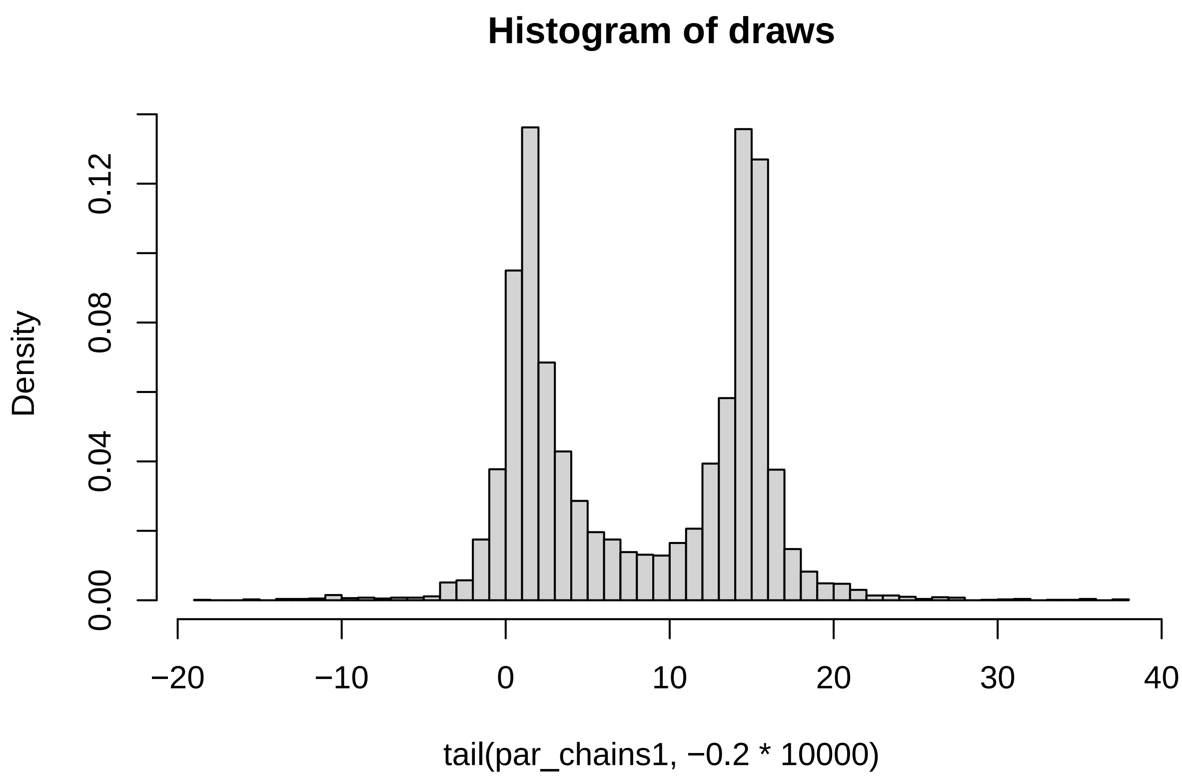
```
problem1 <- function(gamma, n, init, inv_Temper) {
  traj <- matrix(nrow=n+1, ncol=length(inv_Temper))
  traj[1, ] <- rep(init, length(inv_Temper))

  for (i in 1:n) {
    temp_theta <- traj[i, ] + rcauchy(length(inv_Temper), location=0, scale=gamma)
    log_ratio <-  pmin(0, log(unlist(Map(post, temp_theta, inv_Temper))) -
                        log(unlist(Map(post, traj[i, ], inv_Temper))))
    accept <- (log(runif(length(inv_Temper))) <= log_ratio)
```

```
    traj[i+1, ] <- traj[i, ]
    traj[i+1, accept] <- temp_theta[accept]
    swap <- sample(1:length(inv_Temper), 2)
    log_swap <- min(0, sum(log(unlist(Map(post, traj[i+1, rev(swap)], inv_Temper[swap])))) -
                     sum(log(unlist(Map(post, traj[i+1, swap], inv_Temper[swap])))))
    if (log(runif(1)) <= log_swap) {
      traj[i+1, swap] <- traj[i+1, rev(swap)]
    }
  }
  return(traj)
}
```

```
par_chains1 <- problem1(1, 1e4, 100, inv_Temper)[, 1]
par_chains2 <- problem1(1, 1e4, -100, inv_Temper)[, 1]
plot(par_chains1, type='l', col='red', ylim=c(-100, 100))
lines(par_chains2, col='blue')
legend('topright', col=c('red', 'blue'), lty=c(1, 1), legend=c('Chain1', 'Chain2'))
```




```
hist(tail(par_chains1, -.2*1e4), freq=F, breaks=60, main='Histogram of draws')
```



Convergence diagnostics.

```
par_chains1 <- tail(par_chains1, -.3*1e4)
par_chains2 <- tail(par_chains2, -.3*1e4)
# Within-chain var
W <- 1/2*(var(par_chains1) + var(par_chains2))
# Between-chain var
B <- length(par_chains1)*
  ((mean(par_chains1) - mean(c(par_chains1, par_chains2)))^2 +
     (mean(par_chains2) - mean(c(par_chains1, par_chains2)))^2)

V_plus <- (length(par_chains1)-1)/length(par_chains1)*W + B/length(par_chains1)

PSRF <- sqrt(V_plus/W)
```

```
cat(paste("The potential scale reduction factor is ", PSRF))
```

```
## The potential scale reduction factor is  1.0000574070944
```

The PSRF looks good.

```
library(purrr)
my_diff <- function(t, x) {
  mean(diff(x, lag = t)^2, na.rm = TRUE)/2
}

V <- sapply(seq_len(length(par_chains1) - 1), my_diff, x=par_chains1) +
         sapply(seq_len(length(par_chains2) - 1), my_diff, x=par_chains2)

rho <- head_while(1 - V / (2*V_plus), ~. >0)
```

```
n_hat_eff <- 2*length(par_chains1) / (1 + 2*sum(rho))
n_hat_eff
```

```
## [1] 1398.759
```

The effective sample size is relatively small comparing to actual sample size 7000 (after burn-in). The chain by its nature is highly dependent.

Overall the performance is good. From the statistics and the mixing we are able to say that the chains converged.

(3)

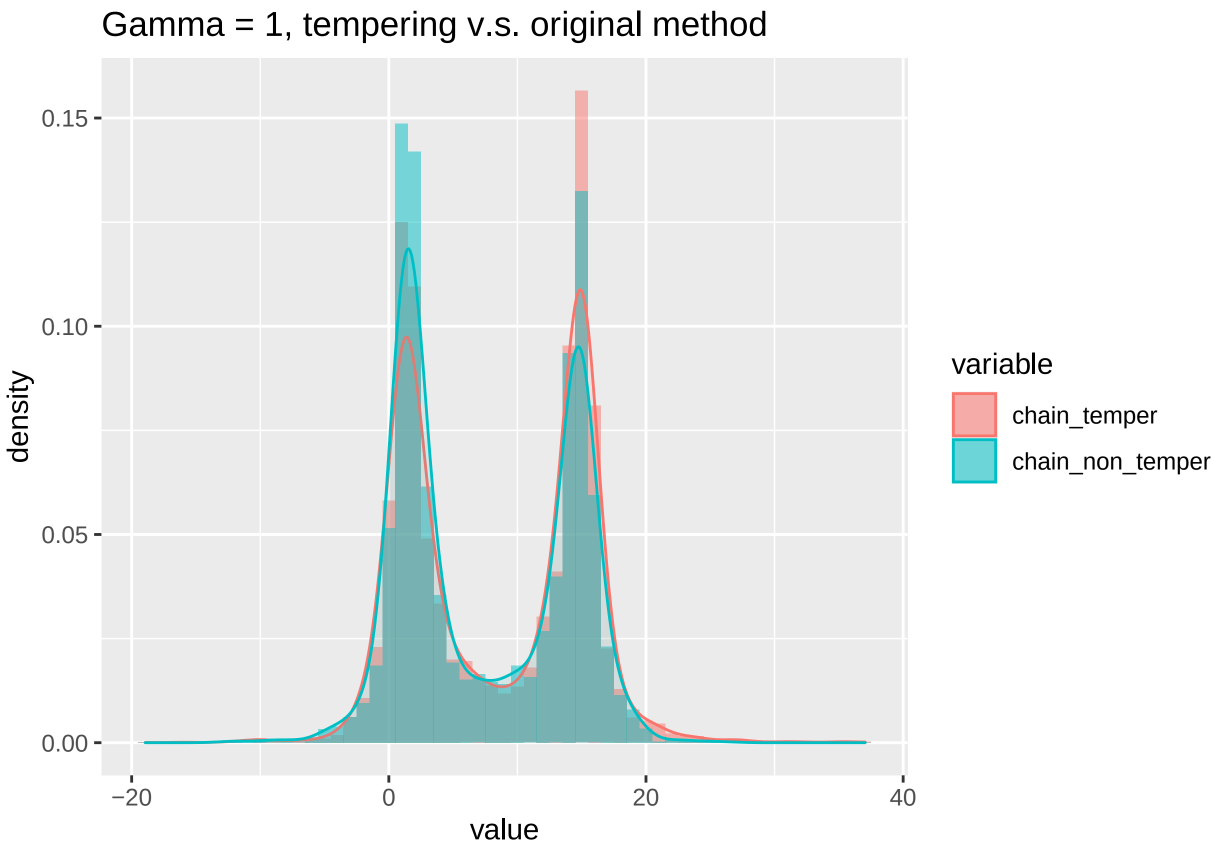
```
y1 <- 1.3
y2 <- 15.0
post_hw7 <- function(theta) {
  1/pi^2/(1+(y1-theta)^2)/(1+(y2-theta)^2)
}

normalize_const <- 1/integrate(post_hw7, -Inf, Inf)$value

post_hw7 <- function(theta) {
  normalize_const * 1/pi^2/(1+(y1-theta)^2)/(1+(y2-theta)^2)
}

problem1_hw7 <- function(gamma, n, init) {
  traj <- matrix(nrow=n+1, ncol=2)
  traj[1, ] <- init

  accept <- matrix(0, nrow=n, ncol=2)
  for (i in 1:n) {
    temp_theta <- rcauchy(1, location=traj[i, 1], scale=gamma)
    log_ratio <- min(0, log(post_hw7(temp_theta)) - log(post_hw7(traj[i, 1])))
    log_U <- log(runif(1))
    if (log_U <= log_ratio) {
      traj[i+1, 1] <- temp_theta
      accept[i, 1] <- 1
    } else {
      traj[i+1, 1] <- traj[i, 1]
    }
    temp_theta <- rcauchy(1, location=traj[i, 2], scale=gamma)
    log_ratio <- min(0, log(post_hw7(temp_theta)) - log(post_hw7(traj[i, 2])))
    log_U <- log(runif(1))
    if (log_U <= log_ratio) {
      traj[i+1, 2] <- temp_theta
      accept[i, 2] <- 1
    } else {
      traj[i+1, 2] <- traj[i, 2]
    }
  }
  return(traj)
}

chain_hw7 <- problem1_hw7(1, 1e4, c(100, -100))[, 1]
```

```r
chain_hw7 <- tail(chain_hw7, -.3*1e4)
```

```r
library(ggplot2)
library(reshape2)
two_chains <- melt(data.frame(chain_temper = par_chains1, chain_non_temper = chain_hw7))
```

```
## No id variables; using all as measure variables
```

```r
ggplot(two_chains, aes(x=value, y=..density.., fill=variable)) +
  geom_histogram(alpha=.5, binwidth = 1, position="identity") +
  geom_density(alpha=0.1, aes(color=variable)) +
  ggtitle("Gamma = 1, tempering v.s. original method")
```



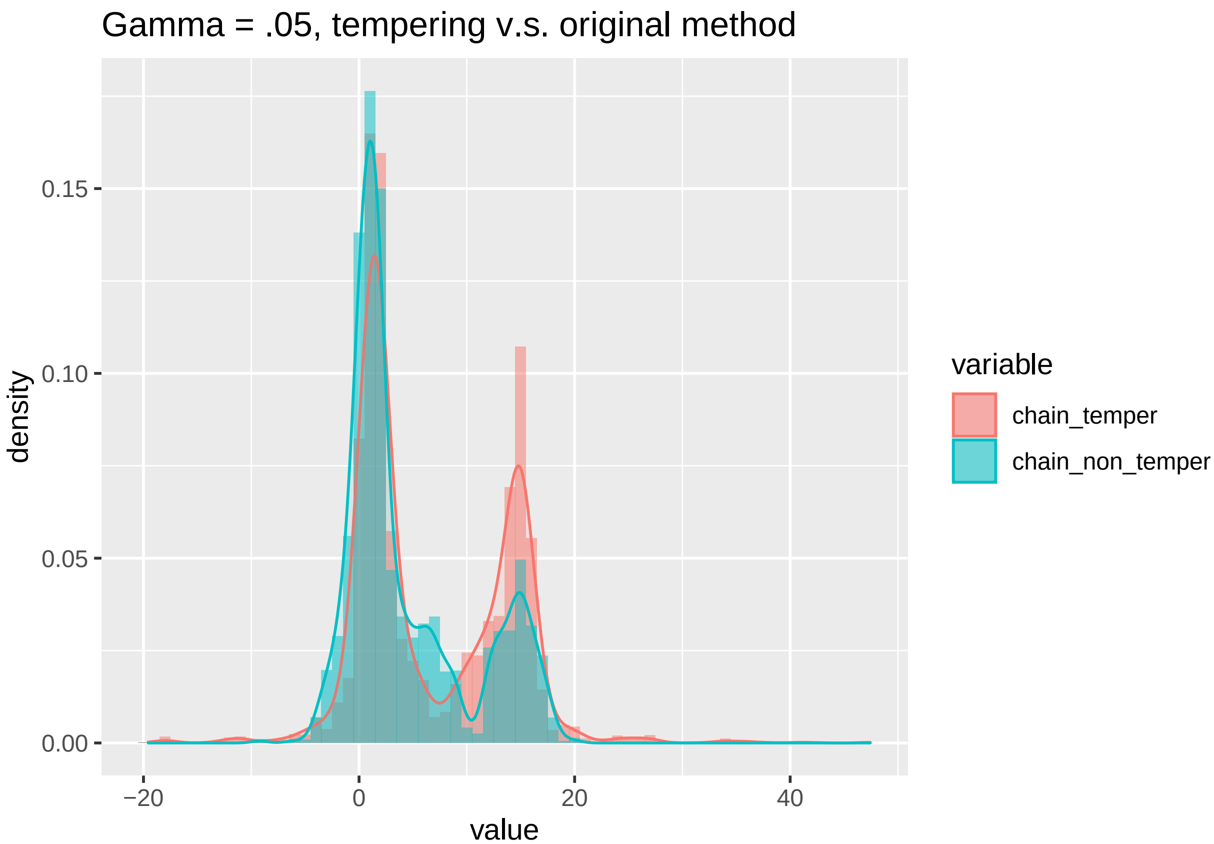
In my implementation the tempering does not help that much comparing to what I have in homework 7. Possibly because the gamma setting. Let's try to lower gamma to 0.05 for both methods and see what happens.

```r
chain_hw7 <- problem1_hw7(.05, 1e4, c(100, -100))[, 1]
chain_hw7 <- tail(chain_hw7, -.3*1e4)
par_chains1 <- problem1(.05, 1e4, 100, inv_Temper)[, 1]
par_chains1 <- tail(par_chains1, -.3*1e4)
```

```r
two_chains <- melt(data.frame(chain_temper = par_chains1, chain_non_temper = chain_hw7))
```

```
## No id variables; using all as measure variables
```

```r
ggplot(two_chains, aes(x=value, y=..density.., fill=variable)) +
  geom_histogram(alpha=.5, binwidth = 1, position="identity") +
  geom_density(alpha=0.1, aes(color=variable)) +
  ggtitle("Gamma = .05, tempering v.s. original method")
```



Now if we lower the gamma, we see that the original one (green color) does not converge well and a big chunk of draws are located in the left mode. Using parallel tempering (red color) we get draws that are more balanced. Although non of those two chains converged due to small gamma, it is safe to say that tempering helps a lot.

Problem 2

```
library(extraDistr)
```

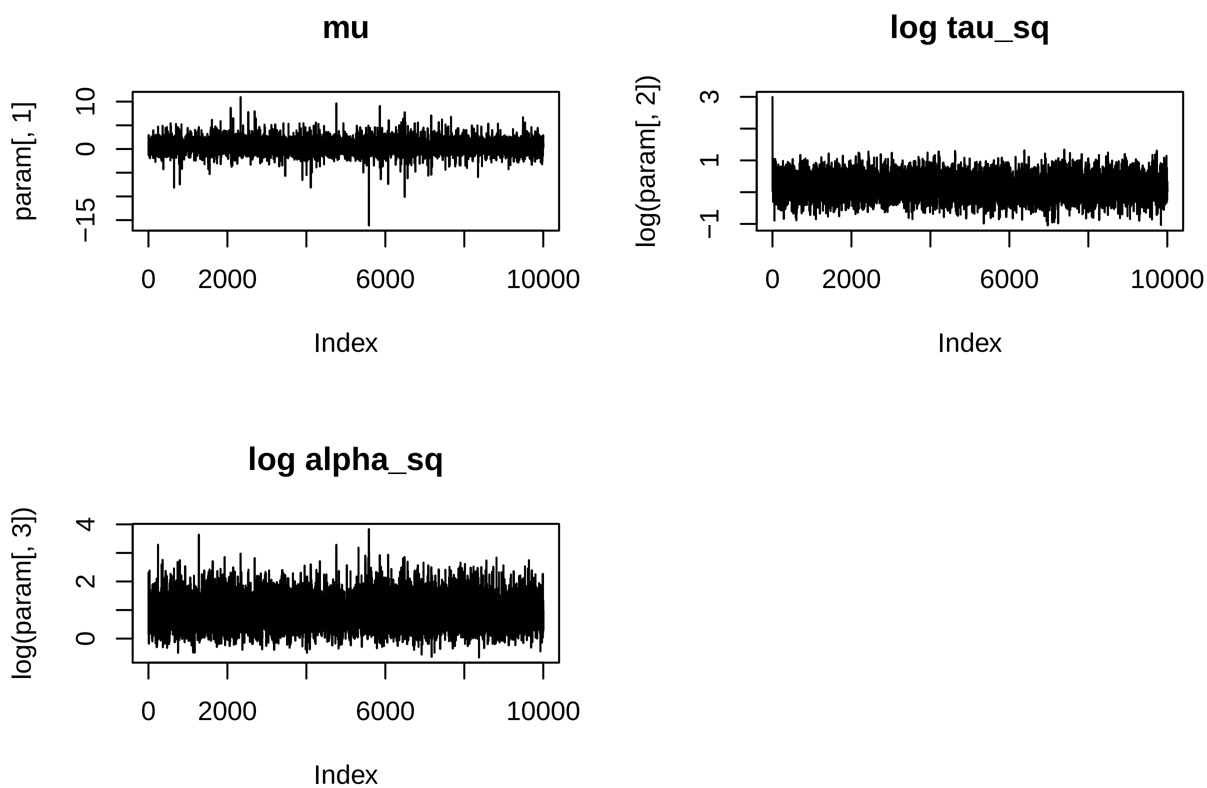
```
##
## Attaching package: 'extraDistr'

## The following object is masked from 'package:purrr':
##
##     rdunif
```

```
N <- 10

y <- rlst(N, df=4, mu = 1, sigma = sqrt(5))
```

```
Nu <- 4
n <- 1e4
param <- matrix(nrow=n+1, ncol=3+N)
param[1, ] <- c(mean(y), N*Nu/2, var(y), rep(NA, N))

for (i in 1:n) {
  U_scale <- sqrt((Nu*param[i, 2] + ((y-param[i, 1])/sqrt(param[i, 3]))^2) /
                  (Nu + 1))
```

```r
  param[i+1, 4:(N+3)] <- rinvchisq(N, nu=Nu+1, tau=U_scale)

  mu_mean <- sum(y/(param[i, 3]^2*param[i+1, 4:(N+3)])) /
    sum(1/(param[i, 3]^2*param[i+1, 4:(N+3)]))
  mu_sd <- sqrt(1 / sum(1/(param[i, 3]^2*param[i+1, 4:(N+3)])))
  param[i+1, 1] <- rnorm(1, mean=mu_mean, sd=mu_sd)

  tau_shape <- N*Nu/2
  tau_rate <- Nu/2*sum(1/param[i+1, 4:(N+3)])
  param[i+1, 2] <- rgamma(1, shape=tau_shape, rate=tau_rate)

  alpha_df <- N
  alpha_scale <- sqrt(sum((y-param[i+1, 1])^2/param[i+1, 4:(N+3)])/N)
  param[i+1, 3] <- rinvchisq(1, nu=alpha_df, tau=alpha_scale)
}
```

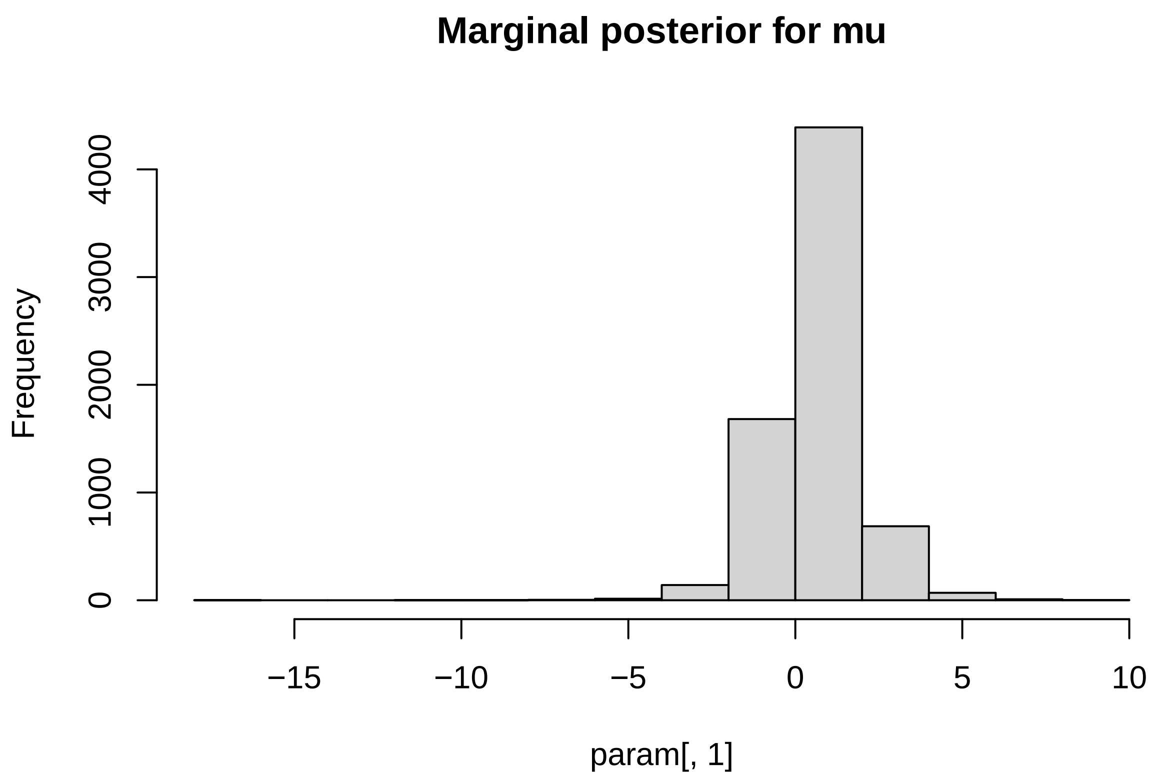
```r
par(mfrow=c(2,2))
plot(param[, 1], type='l', main='mu')
plot(log(param[, 2]), type='l', main='log tau_sq')
plot(log(param[, 3]), type='l', main='log alpha_sq')
```



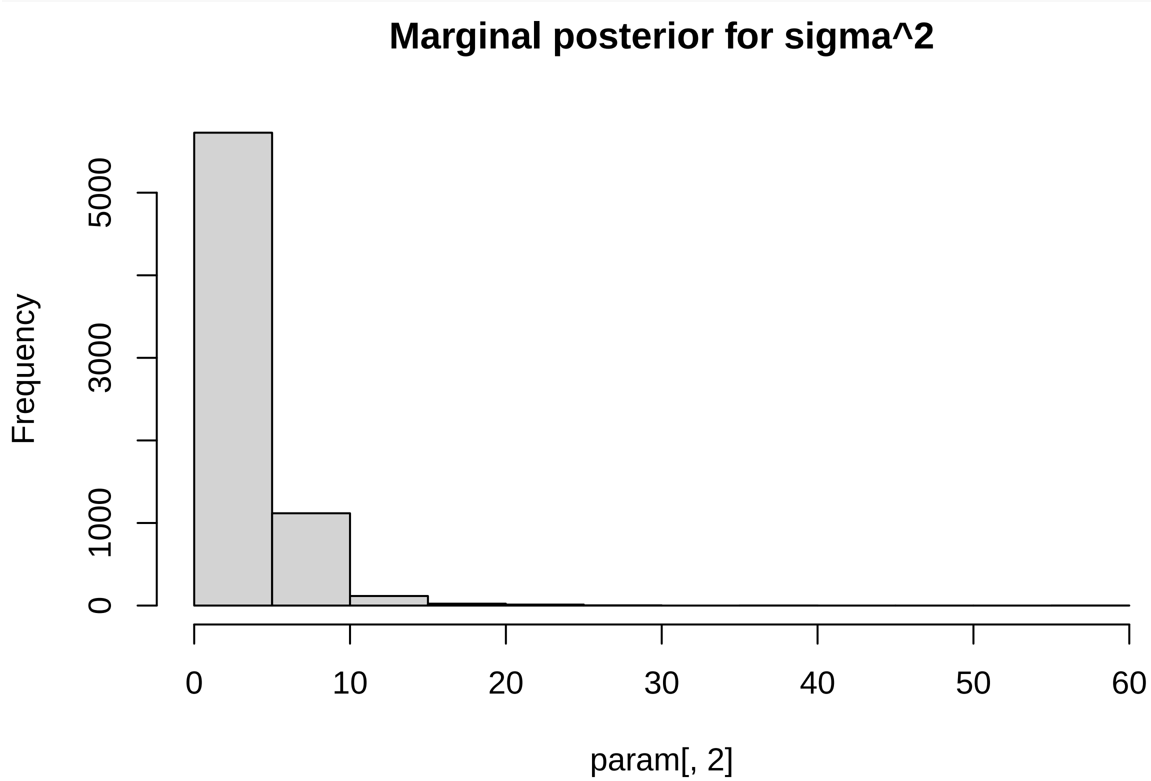
From the path we know that the chain converges well.

```r
param[, 4] <- param[, 2]*param[, 3]
param <- param[, c(1,4)]
param <- tail(param, -.3*n)
```

```r
hist(param[, 1], main='Marginal posterior for mu')
```




```
hist(param[, 2], main='Marginal posterior for sigma^2')
```




```
cat("The estimated mean of posterior of (mu, sigma^2) is ", colMeans(param))
```

```
## The estimated mean of posterior of (mu, sigma^2) is  0.6478893 3.616518
```