Homework 6: MCMC

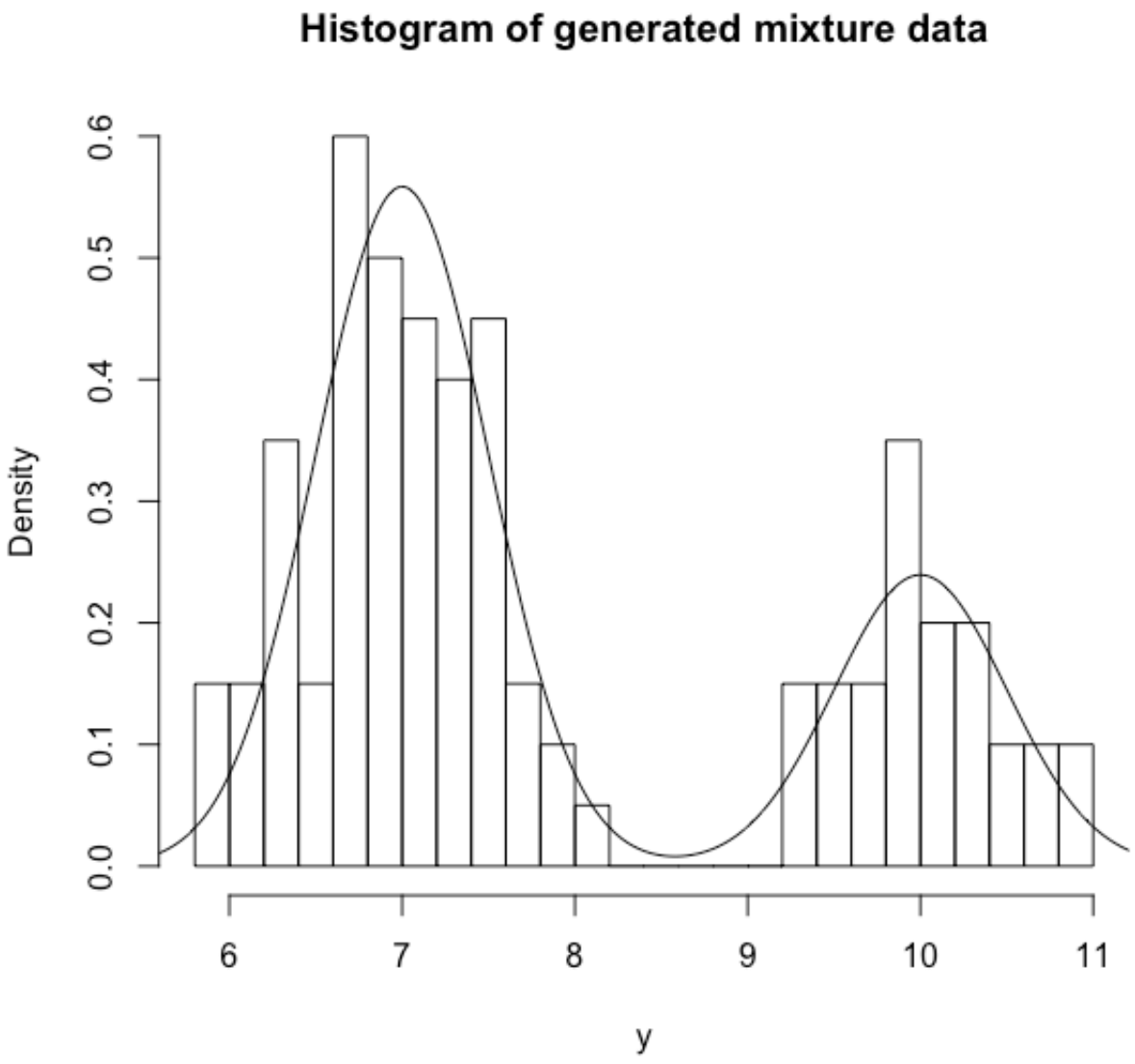
> Shixiang Zhu (GTID # 903280826)
>
> Email: shixiang.zhu@gatech.edu

Problem 7.1

The goal of this problem is to investigate the role of the proposal distribution in a Metropolis–Hastings algorithm designed to simulate from the posterior distribution of a parameter δ. In part (a), you are asked to simulate data from a distribution with δ known. For parts (b)–(d), assume δ is unknown with a Unif(0,1) prior distribution for δ. For parts (b)–(d), provide an appropriate plot and a table summarizing the output of the algorithm. To facilitate comparisons, use the same number of iterations, random seed, starting values, and burn-in period for all implementations of the algorithm.

a. Simulate 200 realizations from the mixture distribution in Equation (7.6) with δ = 0.7. Draw a histogram of these data.

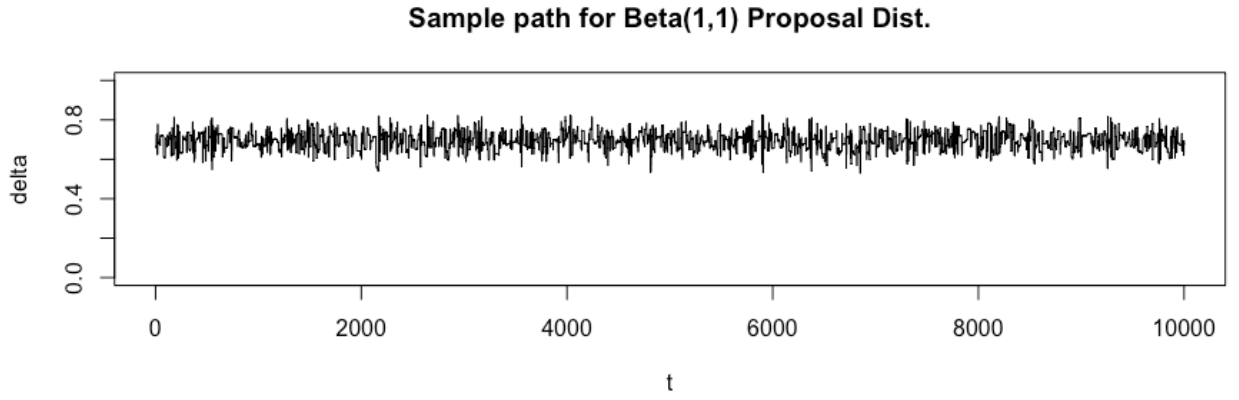
```
# generate 200 realizations from mixture distribution
N          <- 100
components <- sample(1:2,prob=c(0.7,0.3),size=N,replace=TRUE)
mus        <- c(7, 10)
sds        <- c(0.5, 0.5)

y <- rnorm(n=N,mean=mus[components],sd=sds[components])
# plot the histogram of these 200 realizations
par(mfrow=c(1,1))
x <- seq(5, 14, by=.01)
d <- 0.7 * dnorm(x, 7, .5) + 0.3 * dnorm(x, 10, .5)
hist(y, breaks=20, freq=FALSE, main="Histogram of generated mixture
data",ylab="Density")
points(x,d,type="l")
```

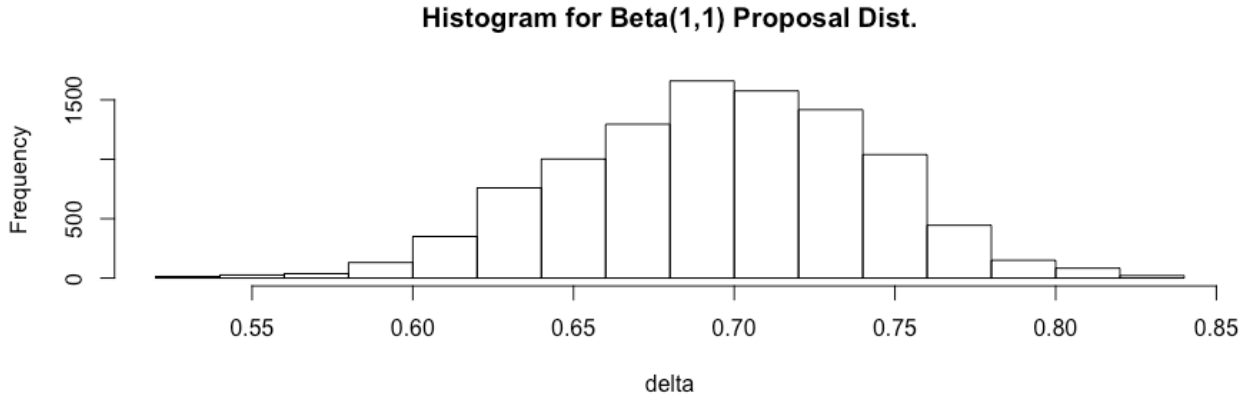


b. Implement an independence chainMCMCprocedure to simulate from the posterior distribution of δ, using your data from part (a).

```
# Init values
n     <- 10000
x.val <- NULL

# Functions
f <- function(x){prod(x*dnorm(y, 7, 0.5) + (1-x)*dnorm(y, 10, 0.5))}
R <- function(xt,x){f(x)*g(xt)/(f(xt)*g(x))}

# Beta(1,1) proposal density
g <- function(x){ dbeta(x, 1, 1) }
x.val <- rbeta(1, 1, 1)
for(i in 1:n){
  xt = x.val[i]
```

```r
  x = rbeta(1,1,1)
  p = min(R(xt,x),1)
  d = rbinom(1,1,p)
  x.val[i+1] = x*d + xt*(1-d)
}
mean(x.val[1:(n+1)])
par(mfrow=c(2,1))
plot(x.val[1:(n+1)],ylim=c(0,1),type="l",ylab="delta",xlab="t")
title("Sample path for Beta(1,1) Proposal Dist.")
hist(x.val[1:(n+1)],breaks=20,xlab="delta",
     main="Histogram for Beta(1,1) Proposal Dist.")
```

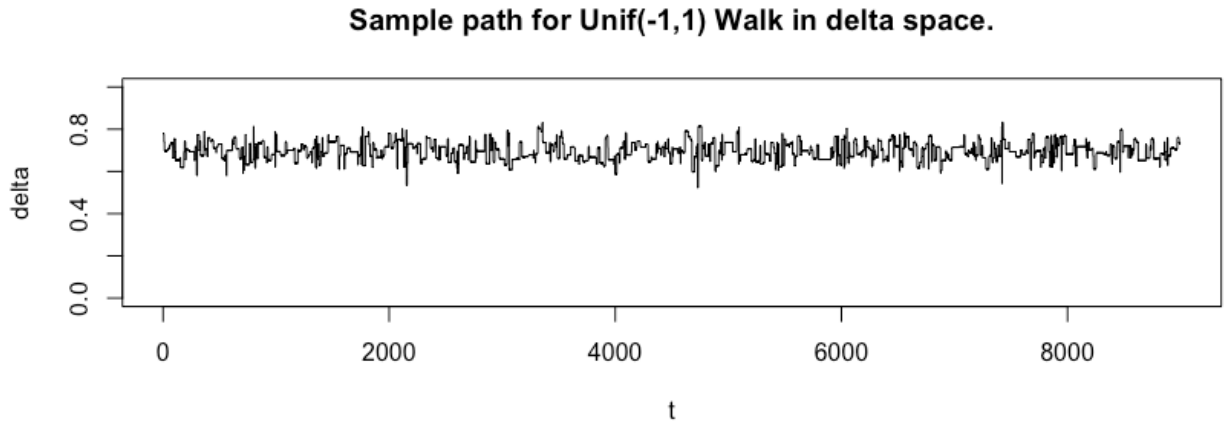


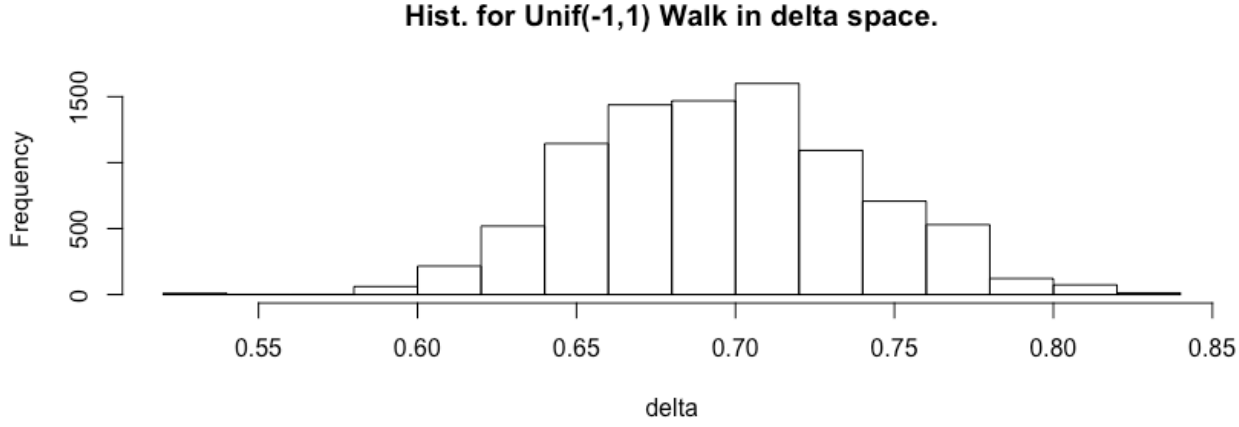


c. Implement a random walk chain with $\delta^* = \delta^{(t)} + \epsilon$ with $\epsilon \sim Unif(-1, 1)$.

```r
n        <- 10000
burnIn   <- 1:1000
x.val    <- rep(0, n)
x.val[1] <- runif(1,0,1)
# functions
loglikFunc <- function(p,y) {
  sum(log(p*dnorm(y,7,.5)+(1-p)*dnorm(y,10,.5)))
}
```

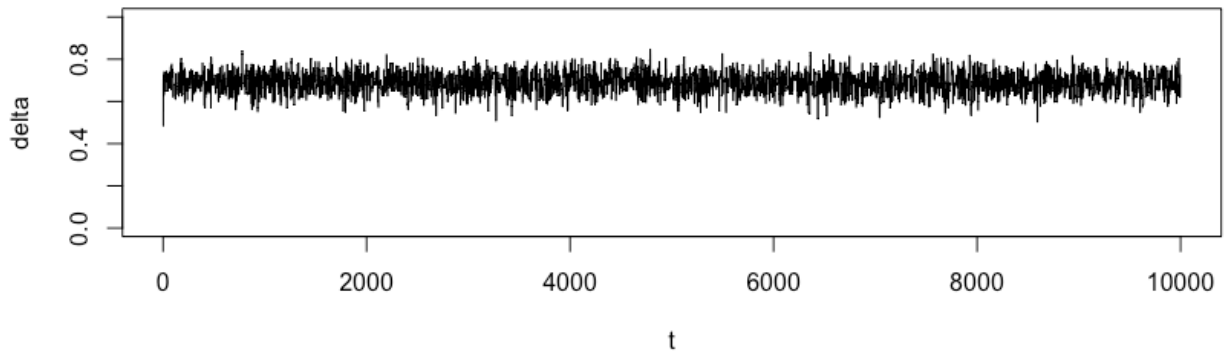
```r
# main
for (i in 1:(n-1)) {
  x.val[i+1]=x.val[i] + runif(1,-1,1)
  if ((x.val[i+1]<0)||(x.val[i+1]>1)){
    x.val[i+1] = x.val[i]
  }else{
    R <- exp(loglikFunc(x.val[i+1],y) - loglikFunc(x.val[i],y))
    if (R < 1)
      if(rbinom(1,1,R)==0) {x.val[i+1] <- x.val[i]}
  }
}
effectiveSize(x.val[-burnIn])
summary(x.val[-burnIn])
mean(x.val[-burnIn])
plot(x.val[-burnIn],ylim=c(0,1),type="l",ylab="delta",xlab="t")
title("Sample path for Unif(-1,1) Walk in delta space.")
hist(x.val[-burnIn],breaks=20,xlab="delta",
     main="Hist. for Unif(-1,1) Walk in delta space.")
```

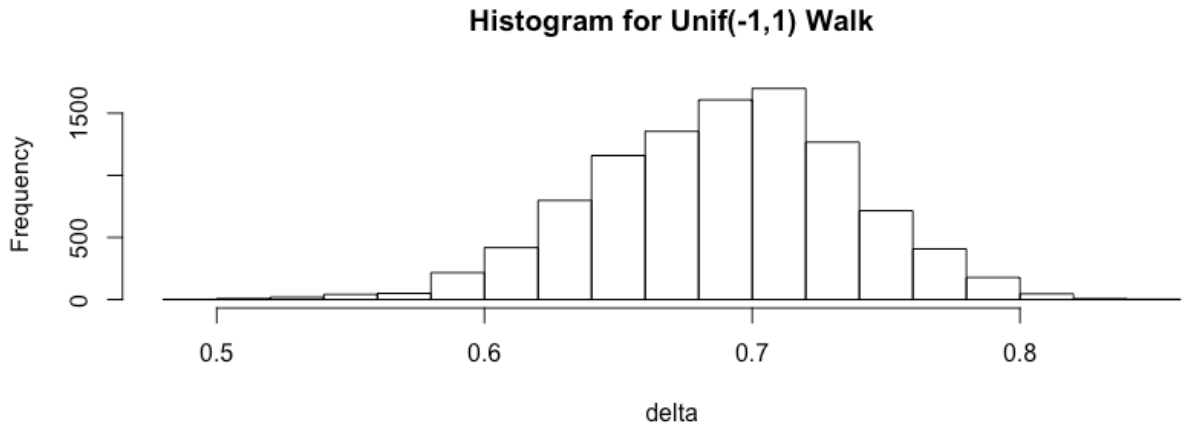




d. Reparameterize the problem letting $U = log\{\delta/(1-\delta)\}$ and $U^* = u^{(t)} + \epsilon$. Implement a random walk chain in U-space as in Equation (7.8).

```r
# Uniform(-1,1) walk
n      <- 10000
burnIn <- 1:1000
# Setup initial values
u    <- rep(0, n)
u[1] <- runif(1, -1, 1)
p    <- rep(0, n)
p[1] <- exp(u[1])/(1+exp(u[1]))

# loglikelihood function
loglikFunc <- function(p,x) {
  sum(log(p*dnorm(x,7,.5)+(1-p)*dnorm(x,10,.5)))
}

# Main for random walk
for (i in 1:(n-1)) {
  u[i+1] <- u[i] + runif(1, -1, 1)
  p[i+1] <- exp(u[i+1])/(1+exp(u[i+1]))
  R=exp(loglikFunc(p[i+1], y) - loglikFunc(p[i], y)) *
exp(u[i])/exp(u[i+1])
  if (R < 1)
    if(rbinom(1, 1, R)==0)  {p[i+1] <- p[i]; u[i+1] <- u[i]}
}
mean(p[-burnIn])
par(mfrow=c(2,1))
plot(p,ylim=c(0,1),type="l",ylab="delta",xlab="t")
hist(p,breaks=20,xlab="delta",
     main="Histogram for Unif(-1,1) Walk")
```





d. Compare the estimates and convergence behavior of the three algorithms.

According to the sample paths and effective sample sizes, It is obvious that reparameterization of the problem had the best result against others. The MCMC is less autocorrelated and faster into stationary distribution.

Problem 7.2

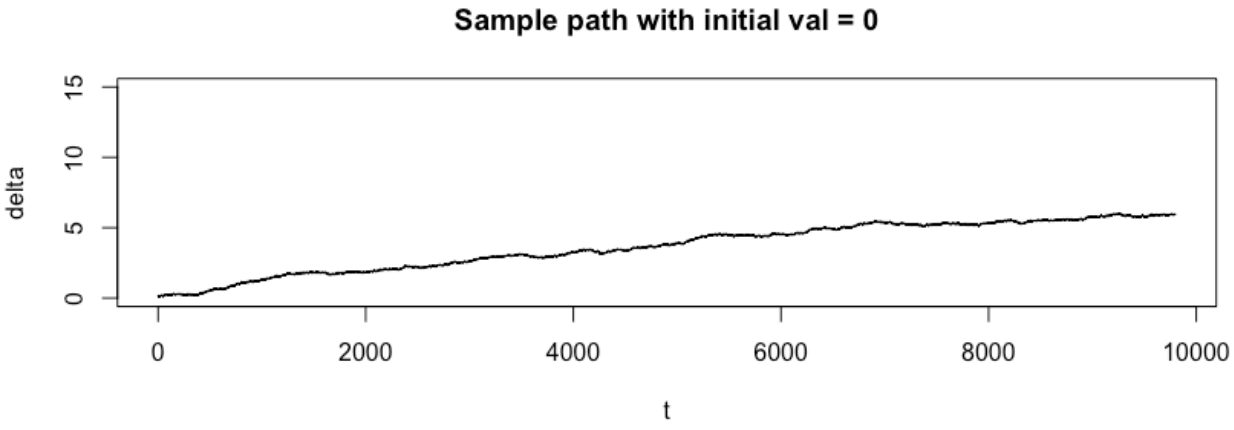
Simulating from the mixture distribution in Equation (7.6) is straightforward [see part (a) of Problem 7.1]. However, using the Metropolis–Hastings algorithm to simulate realizations from this distribution is useful for exploring the role of the proposal distribution.

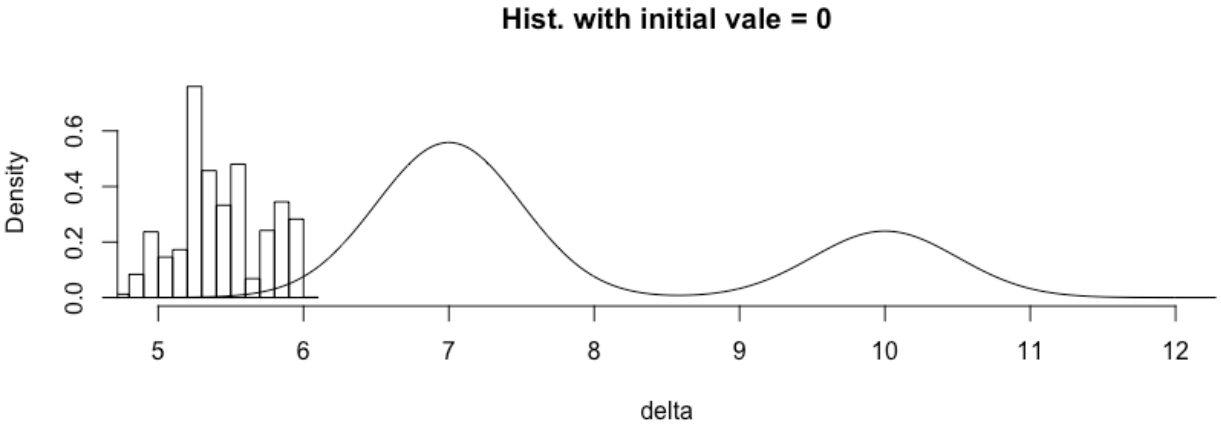
a. Implement a Metropolis–Hastings algorithm to simulate from Equation (7.6) with δ = 0.7, using $N(x^{(t)}, 0.01)$ as the proposal distribution. For each of three starting values, $x^{(0)}$ = 0, 7, and 15, run the chain for 10,000 iterations. Plot the sample path of the output from each chain. If only one of the sample paths was available, what would you conclude about the chain? For each of the simulations, create a histogram of the realizations with the true density superimposed on the histogram. Based on your output from all three chains, what can you say about the behavior of the chain?

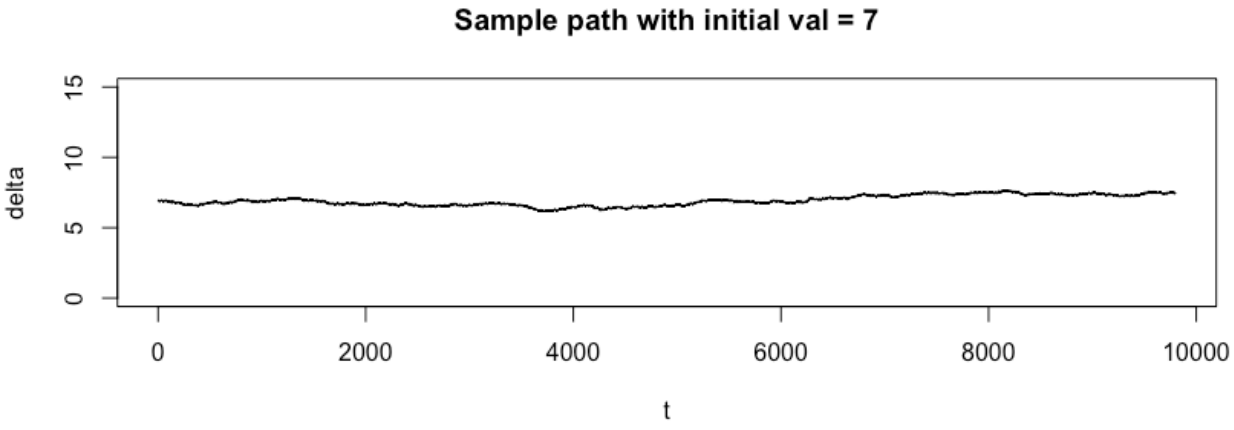
```
1  # Init values
```

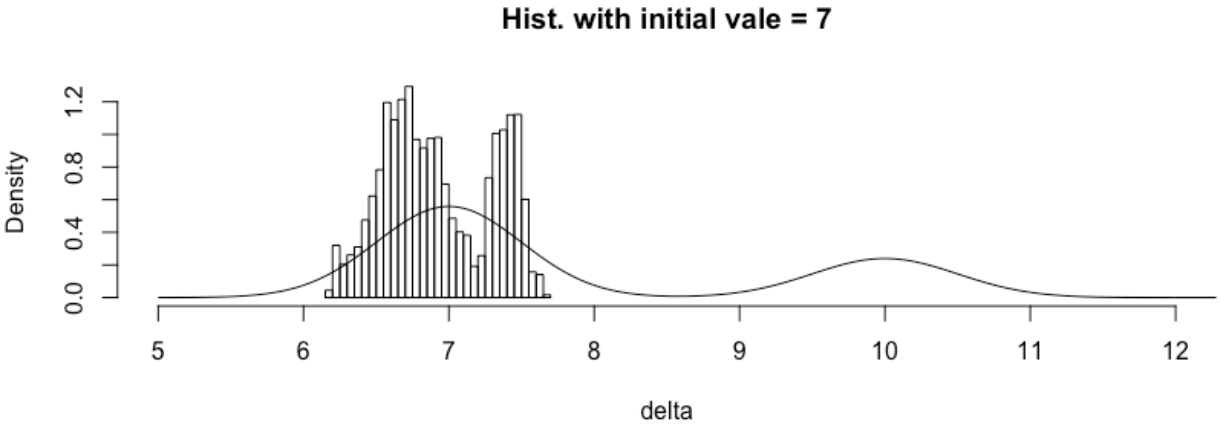
```r
set.seed(920804)
n        <- 10000
burn.in  <- 200
delta    <- 0.7
x.val    <- NULL
x.val[1] <- 15
sigma    <- 0.01 # 0.01, 0.5
for(i in 1:n){
  xt = x.val[i]
  x = rnorm(1, xt, sigma)
  p = min(R(xt,x),1)
  d = rbinom(1,1,p)
  x.val[i+1] = x*d + xt*(1-d)
}

f <- function(x){delta*dnorm(x,7,0.5) + (1-delta)*dnorm(x,10,0.5)}
R <- function(xt,x){f(x)*g(xt, x)/(f(xt)*g(x, xt))}

# N(xt, 0.01) PROPOSAL DENSITY
g <- function(x, xt){dnorm(x, xt, sigma)}

par(mfrow=c(2, 1))
plot(x.val[-(1:burn.in)],ylim=c(0,15),type="l",ylab="delta",xlab="t",
main="Sample path with initial val = 0")

x <- seq(5,14,by=.01)
d <- .7*dnorm(x, 7, 0.5) + .3*dnorm(x, 10, 0.5)
hist(xlim = c(5, 12),x.val[-(1:burn.in)],breaks=50,freq=F,xlab="delta",
main="Hist. with initial vale = 0")
points(x, d, "l")
```

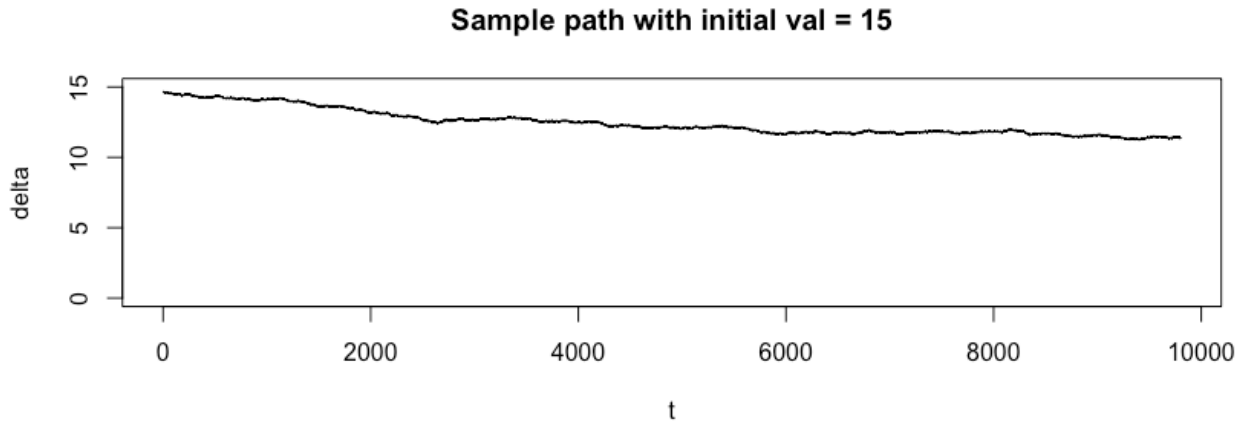
From the sample path, we can easily tell the final sample can't converge into the stationary distribution. Since the variance of proposal distribution is too small to cover the sample space, the choices of sample are being stuck around the starting point during the sampling process.

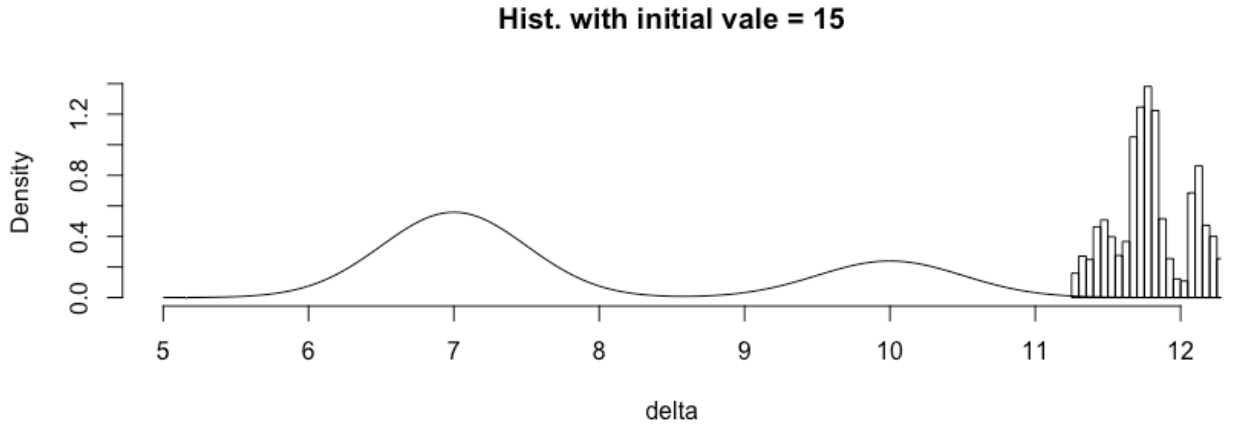

Sample path with initial val = 0
delta
0
5
10
15
0
2000
4000
6000
8000
10000
t


Hist. with initial vale = 0
Density
0.0
0.2
0.4
0.6
5
6
7
8
9
10
11
12
delta


Sample path with initial val = 7
delta
0
5
10
15
0
2000
4000
6000
8000
10000
t

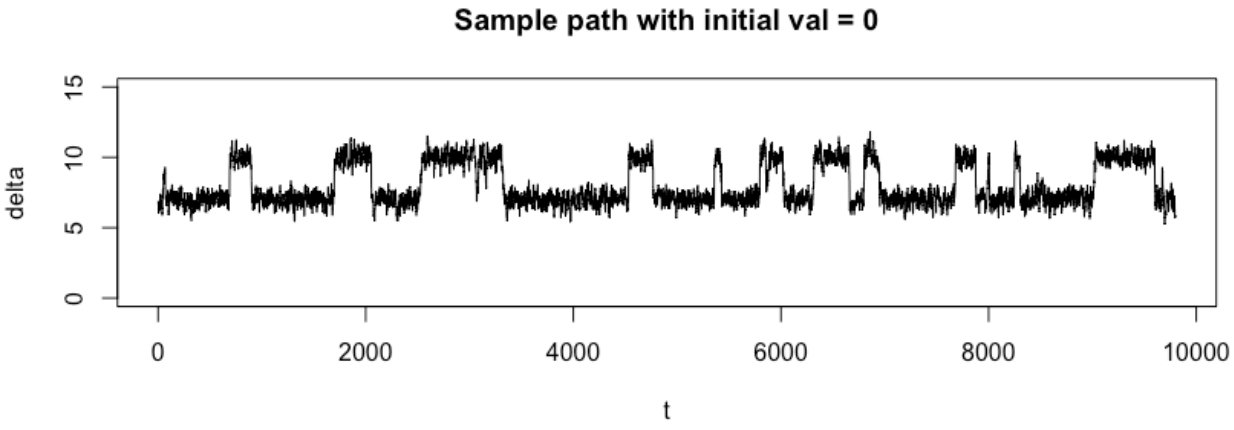

Hist. with initial vale = 7
Density
0.0
0.4
0.8
1.2
5
6
7
8
9
10
11
12
delta

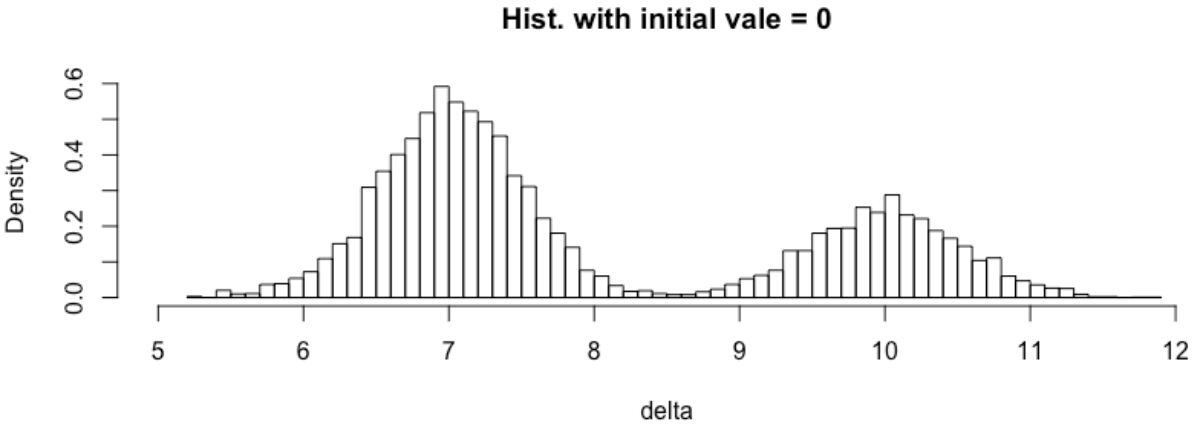


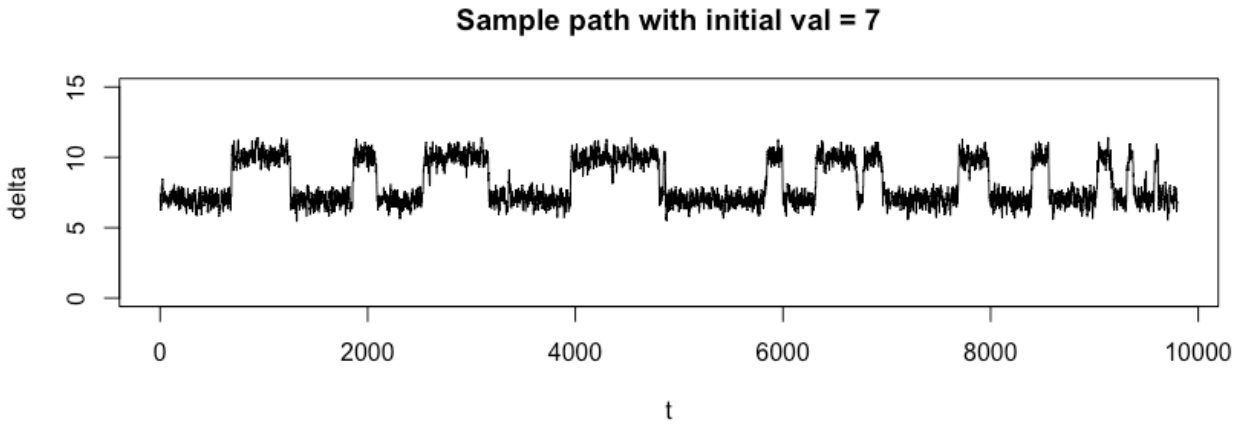


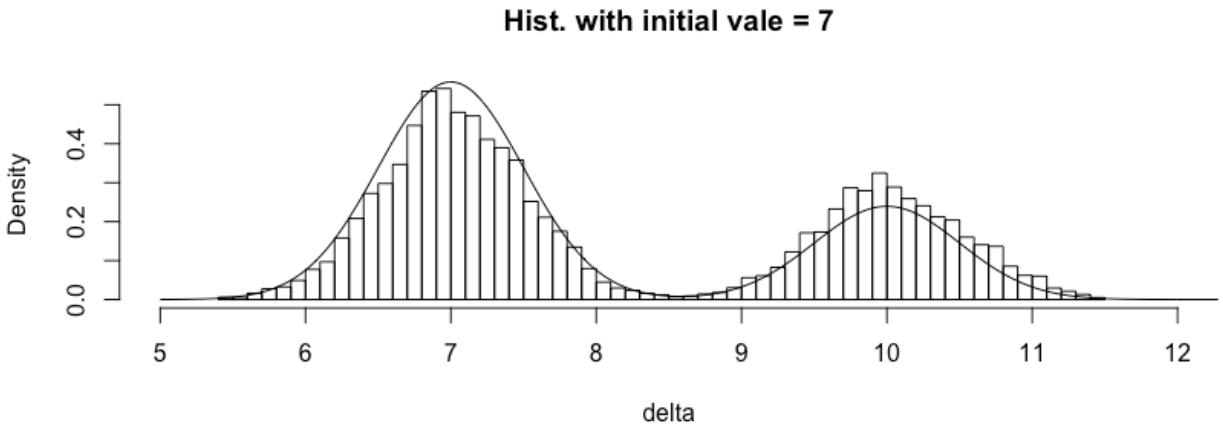
b. Now change the proposal distribution to improve the convergence properties of the chain. Using the new proposal distribution, repeat part (a).

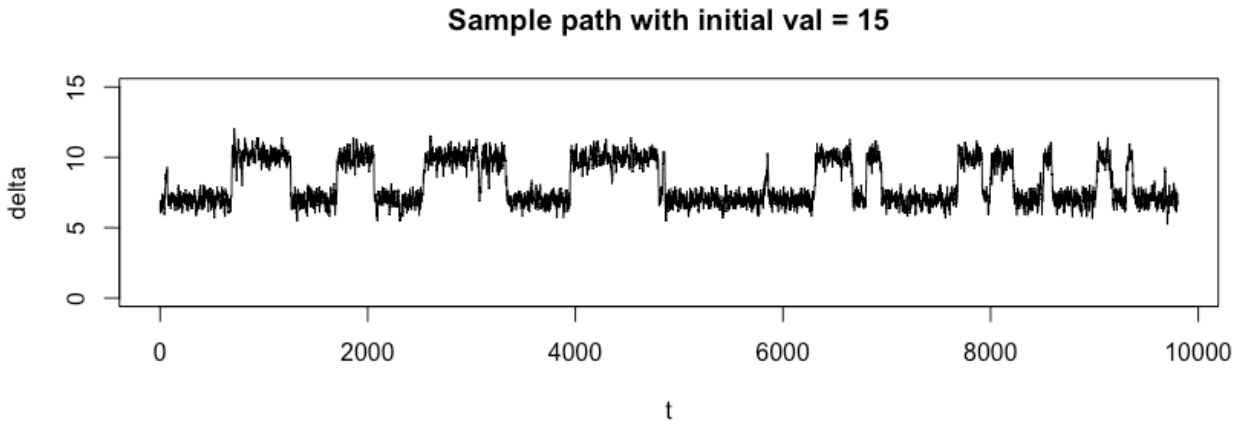
Change the proposal distribution to $N(x^{(t)}, 0.5^2)$.

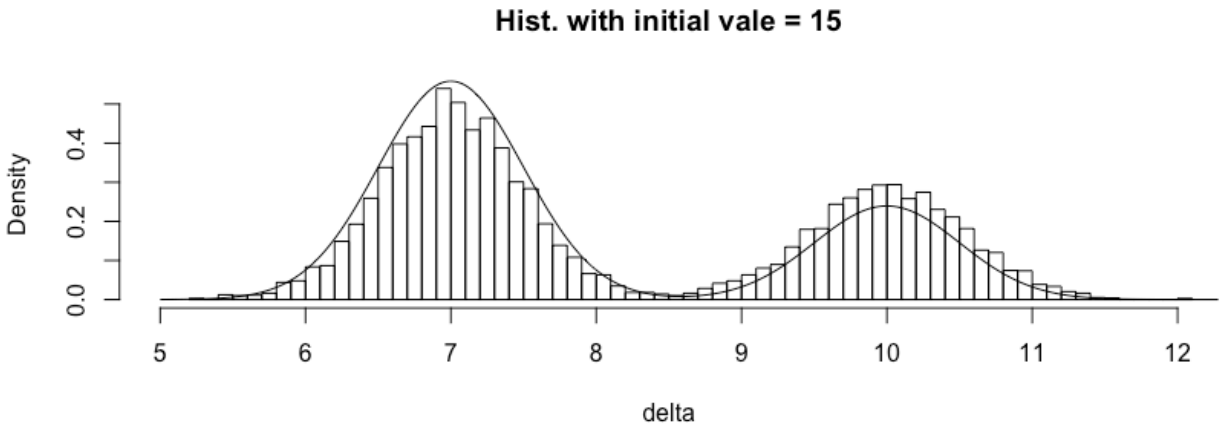

Sample path with initial val = 0
delta
0
5
10
15
0
2000
4000
6000
8000
10000
t


Hist. with initial vale = 0
Density
0.0
0.2
0.4
0.6
5
6
7
8
9
10
11
12
delta


Sample path with initial val = 7
delta
0
5
10
15
0
2000
4000
6000
8000
10000
t


Hist. with initial vale = 7
Density
0.0
0.2
0.4
5
6
7
8
9
10
11
12
delta


Sample path with initial val = 15
delta
0
5
10
15
0
2000
4000
6000
8000
10000
t


Hist. with initial vale = 15
Density
0.0
0.2
0.4
5
6
7
8
9
10
11
12
delta