```r
###########################################################################
###########################################################################
########################## Mixture of Normals #############################
###########################################################################
###########################################################################


###### Clear environment and load libraries
rm(list = ls())
library(coda)

###### Now to the exercise
#Use a normal proposal
#Choose delta > 0 such that the acceptance probability is very close to 45%
#we will try different values: 0.01, 0.05, 2, 4, 8 and 100
delta <- 0.01

#Initial values for sampler
theta <- 0

#First set number of iterations and burn-in, then set seed
n_iter <- 10000; burn_in <- 0.3*n_iter
set.seed(1234)

#Set counter for acceptances
accept_counter <- 0

#Set null matrices to save samples
THETA <- matrix(0,nrow=n_iter,ncol=1)

#Now, to the Gibbs sampler
for(s in 1:(n_iter+burn_in)){

  #generate proposal
  theta_star <- rnorm(1,theta, delta)

  #compute acceptance ratio/probability
  #do so on log scale because r can be numerically unstable
  log_r <- log(exp(-0.5*theta_star^2) + 0.5*exp(-0.5*(theta_star-3)^2)) -
    log(exp(-0.5*theta^2) + 0.5*exp(-0.5*(theta-3)^2))

  if(log(runif(1)) < log_r){
    accept_counter <- accept_counter + 1
    theta <- theta_star
    }

  if(s > burn_in){
    THETA[(s-burn_in),] <- theta
  }
}

#Check acceptance rate
accept_counter/(n_iter+burn_in)


plot(mcmc(THETA))

autocorr.plot(mcmc(THETA))




x <- seq(from=-5,to=7,by=.05)
y <- (2/(3*sqrt(2*pi))) * (exp(-0.5*(x^2)) + 0.5*exp(-0.5*(x-3)^2))
plot(density(THETA),col="red4",lwd=1.5,type="l",xlim=c(-5,7))
points(x,y,col="blue3",xlab=expression(theta),ylab="Density",
     main=expression(paste(pi,"(", theta,"|y)")))
labels <- c("True Density", "Accepted Samples")
legend("topright", labels, lwd=2, lty=c(1.5,1.5),
       col=c('blue3',"red4"))
```