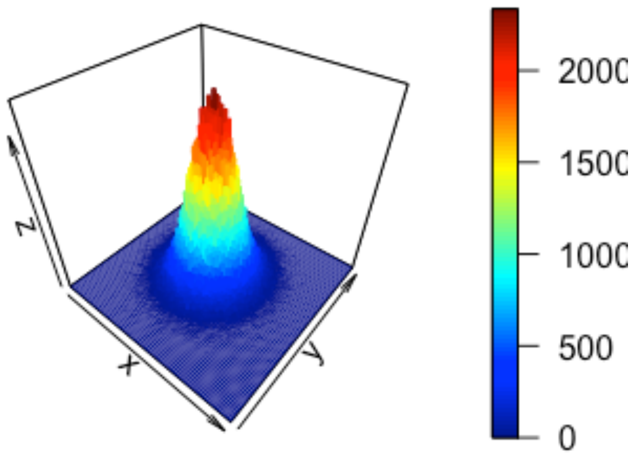
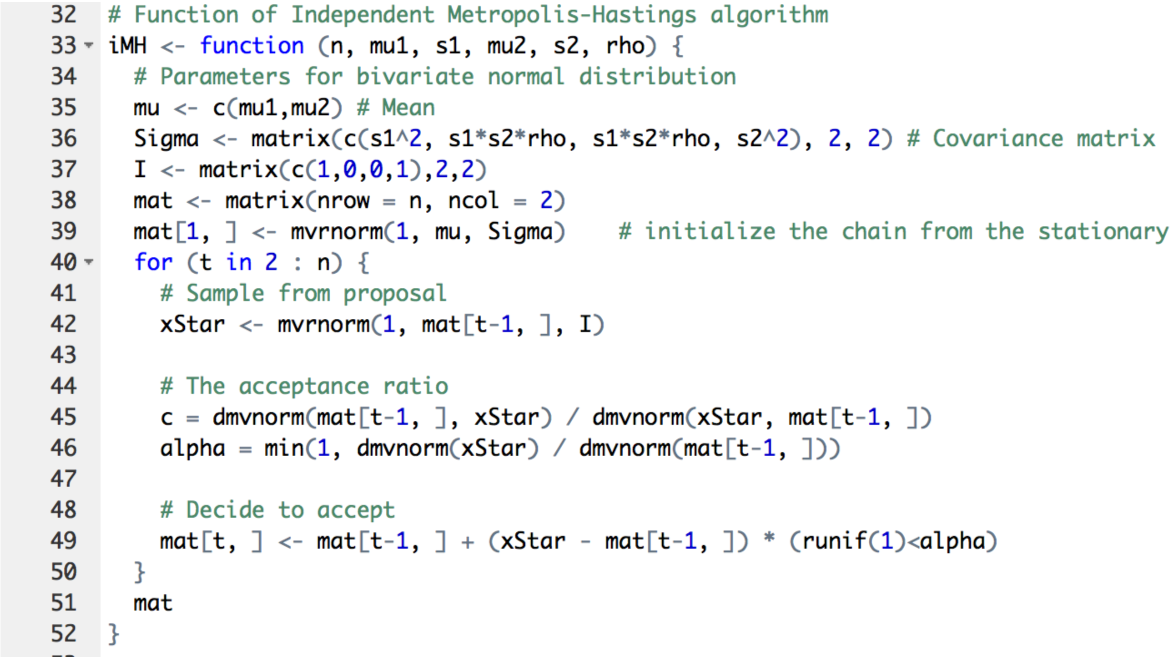
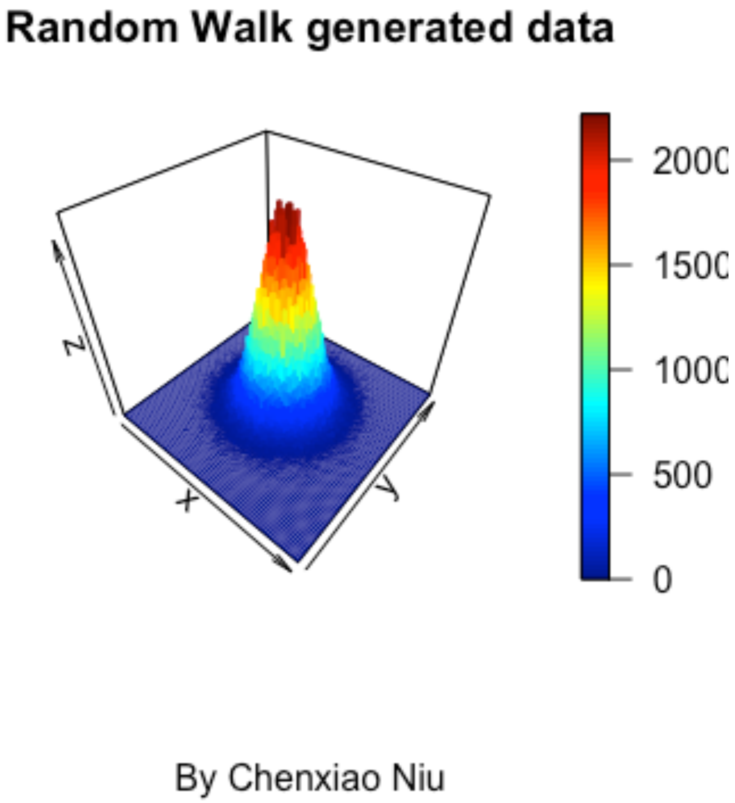
iMH generated data

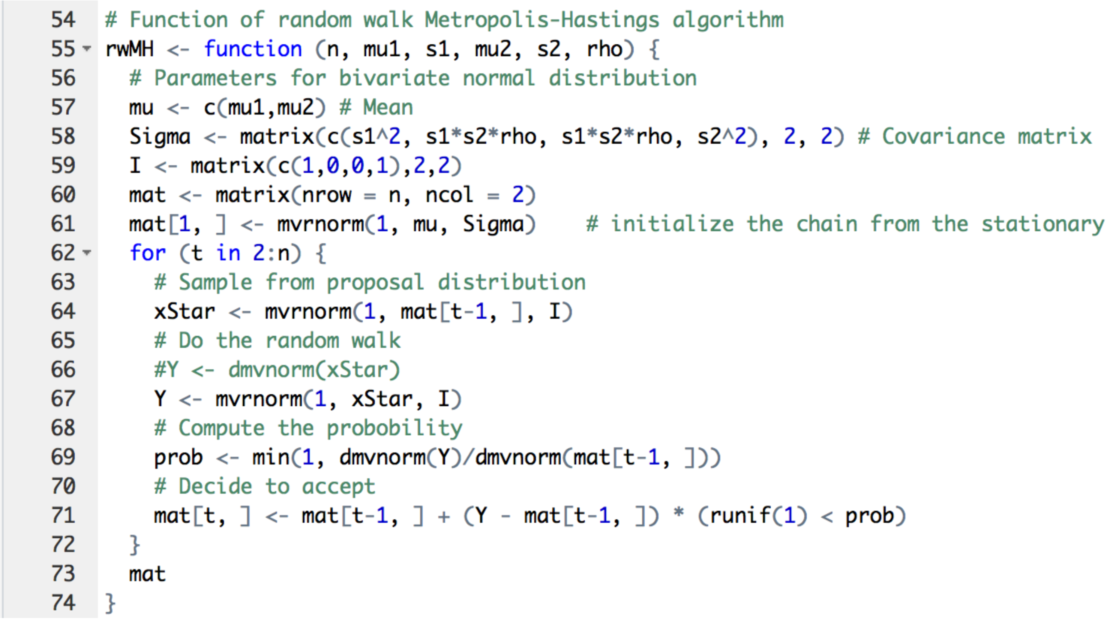


By Chenxiao Niu

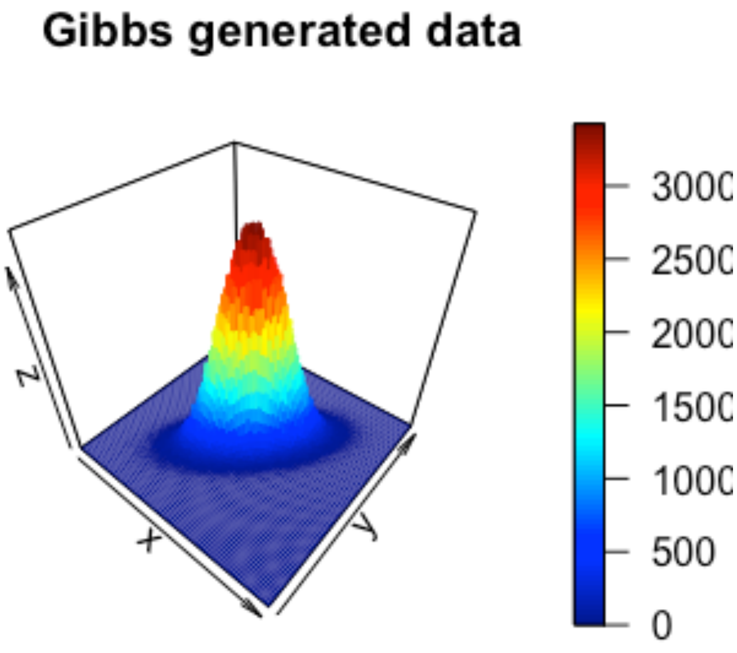


```r
# Function of Independent Metropolis-Hastings algorithm
iMH <- function (n, mu1, s1, mu2, s2, rho) {
  # Parameters for bivariate normal distribution
  mu <- c(mu1,mu2) # Mean
  Sigma <- matrix(c(s1^2, s1*s2*rho, s1*s2*rho, s2^2), 2, 2) # Covariance matrix
  I <- matrix(c(1,0,0,1),2,2)
  mat <- matrix(nrow = n, ncol = 2)
  mat[1, ] <- mvrnorm(1, mu, Sigma)    # initialize the chain from the stationary
  for (t in 2 : n) {
    # Sample from proposal
    xStar <- mvrnorm(1, mat[t-1, ], I)

    # The acceptance ratio
    c = dmvnorm(mat[t-1, ], xStar) / dmvnorm(xStar, mat[t-1, ])
    alpha = min(1, dmvnorm(xStar) / dmvnorm(mat[t-1, ]))

    # Decide to accept
    mat[t, ] <- mat[t-1, ] + (xStar - mat[t-1, ]) * (runif(1)<alpha)
  }
  mat
}
```

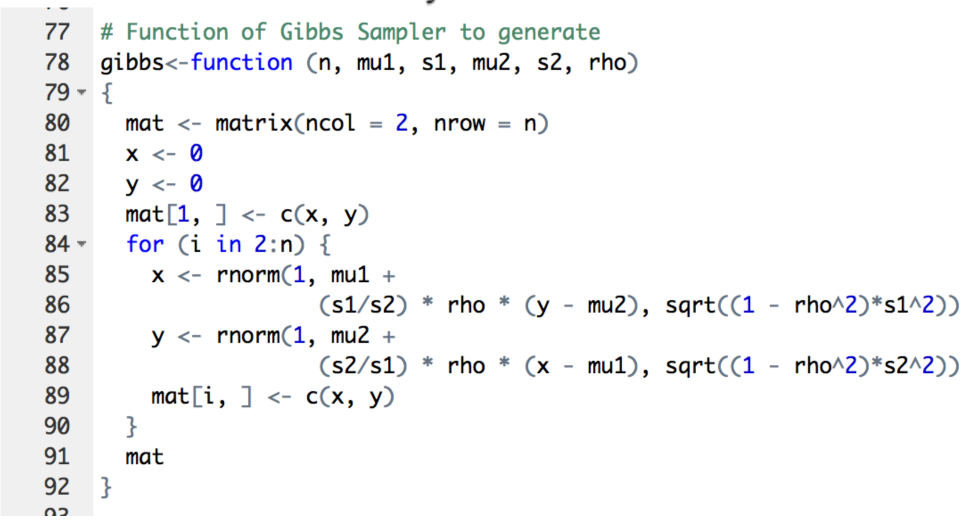




```r
# Function of random walk Metropolis-Hastings algorithm
rwMH <- function (n, mu1, s1, mu2, s2, rho) {
  # Parameters for bivariate normal distribution
  mu <- c(mu1,mu2) # Mean
  Sigma <- matrix(c(s1^2, s1*s2*rho, s1*s2*rho, s2^2), 2, 2) # Covariance matrix
  I <- matrix(c(1,0,0,1),2,2)
  mat <- matrix(nrow = n, ncol = 2)
  mat[1, ] <- mvrnorm(1, mu, Sigma)     # initialize the chain from the stationary
  for (t in 2:n) {
    # Sample from proposal distribution
    xStar <- mvrnorm(1, mat[t-1, ], I)
    # Do the random walk
    #Y <- dmvnorm(xStar)
    Y <- mvrnorm(1, xStar, I)
    # Compute the probobility
    prob <- min(1, dmvnorm(Y)/dmvnorm(mat[t-1, ]))
    # Decide to accept
    mat[t, ] <- mat[t-1, ] + (Y - mat[t-1, ]) * (runif(1) < prob)
  }
  mat
}
```



By Chenxiao Niu



```
# Function of Gibbs Sampler to generate
gibbs<-function (n, mu1, s1, mu2, s2, rho)
{
  mat <- matrix(ncol = 2, nrow = n)
  x <- 0
  y <- 0
  mat[1, ] <- c(x, y)
  for (i in 2:n) {
    x <- rnorm(1, mu1 +
                 (s1/s2) * rho * (y - mu2), sqrt((1 - rho^2)*s1^2))
    y <- rnorm(1, mu2 +
                 (s2/s1) * rho * (x - mu1), sqrt((1 - rho^2)*s2^2))
    mat[i, ] <- c(x, y)
  }
  mat
}
```