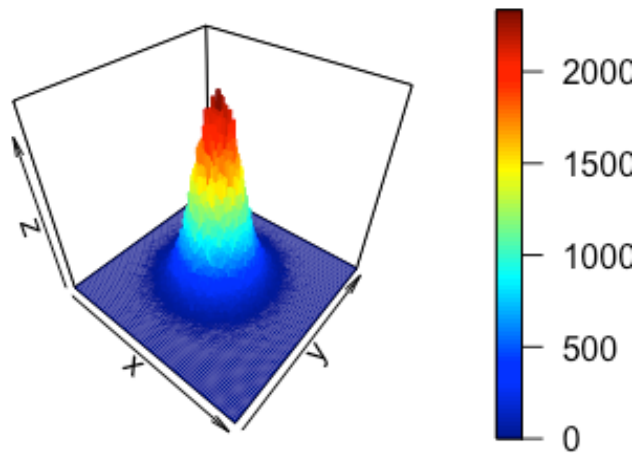


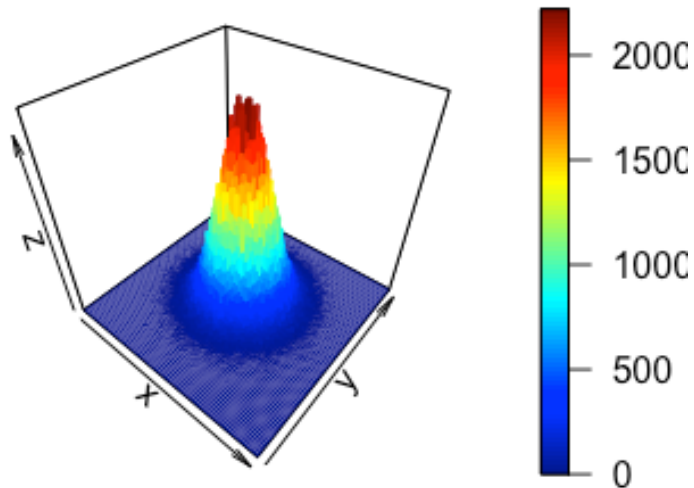
iMH generated data



By Chenxiao Niu

```
32 # Function of Independent Metropolis-Hastings algorithm
33 iMH <- function (n, mu1, s1, mu2, s2, rho) {
34   # Parameters for bivariate normal distribution
35   mu <- c(mu1, mu2) # Mean
36   Sigma <- matrix(c(s1^2, s1*s2*rho, s1*s2*rho, s2^2), 2, 2) # Covariance matrix
37   I <- matrix(c(1, 0, 0, 1), 2, 2)
38   mat <- matrix(nrow = n, ncol = 2)
39   mat[1, ] <- mvrnorm(1, mu, Sigma) # initialize the chain from the stationary
40   for (t in 2 : n) {
41     # Sample from proposal
42     xStar <- mvrnorm(1, mat[t-1, ], I)
43
44     # The acceptance ratio
45     c = dmvnorm(mat[t-1, ], xStar) / dmvnorm(xStar, mat[t-1, ])
46     alpha = min(1, dmvnorm(xStar) / dmvnorm(mat[t-1, ]))
47
48     # Decide to accept
49     mat[t, ] <- mat[t-1, ] + (xStar - mat[t-1, ]) * (runif(1) < alpha)
50   }
51   mat
52 }
```

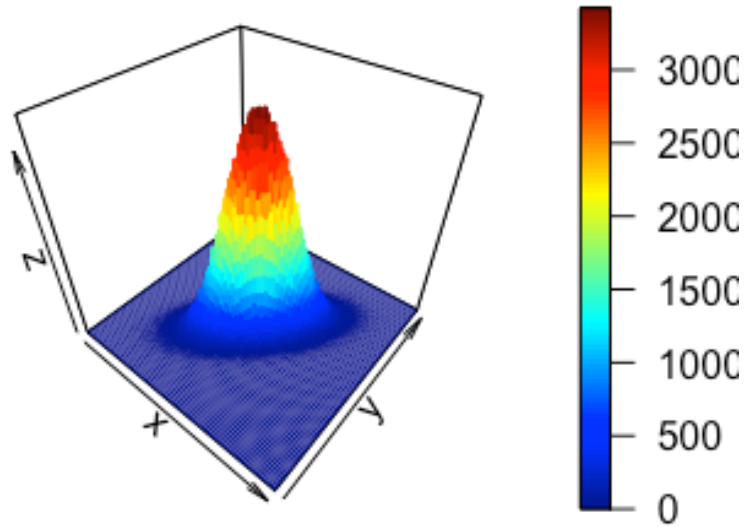
Random Walk generated data



By Chenxiao Niu

```
54 # Function of random walk Metropolis-Hastings algorithm
55 rwMH <- function(n, mu1, s1, mu2, s2, rho) {
56   # Parameters for bivariate normal distribution
57   mu <- c(mu1, mu2) # Mean
58   Sigma <- matrix(c(s1^2, s1*s2*rho, s1*s2*rho, s2^2), 2, 2) # Covariance matrix
59   I <- matrix(c(1, 0, 0, 1), 2, 2)
60   mat <- matrix(nrow = n, ncol = 2)
61   mat[1, ] <- mvrnorm(1, mu, Sigma) # initialize the chain from the stationary
62   for (t in 2:n) {
63     # Sample from proposal distribution
64     xStar <- mvrnorm(1, mat[t-1, ], I)
65     # Do the random walk
66     #Y <- dmvnorm(xStar)
67     Y <- mvrnorm(1, xStar, I)
68     # Compute the probability
69     prob <- min(1, dmvnorm(Y)/dmvnorm(mat[t-1, ]))
70     # Decide to accept
71     mat[t, ] <- mat[t-1, ] + (Y - mat[t-1, ]) * (runif(1) < prob)
72   }
73   mat
74 }
```

Gibbs generated data



By Chenxiao Niu

```
77 # Function of Gibbs Sampler to generate
78 gibbs<-function(n, mu1, s1, mu2, s2, rho)
79 {
80   mat <- matrix(ncol = 2, nrow = n)
81   x <- 0
82   y <- 0
83   mat[1, ] <- c(x, y)
84   for (i in 2:n) {
85     x <- rnorm(1, mu1 +
86               (s1/s2) * rho * (y - mu2), sqrt((1 - rho^2)*s1^2))
87     y <- rnorm(1, mu2 +
88               (s2/s1) * rho * (x - mu1), sqrt((1 - rho^2)*s2^2))
89     mat[i, ] <- c(x, y)
90   }
91   mat
92 }
```