

# STA3127: Statistical Computing

## Homework 8 (Due: Dec 11)

Note: You can only use `runif(·, 0, 1)` for random number generation.

**Q1.** We want to generate a sample from the unnormalized density

$$g(x) = \exp(-(x+1)^2 - y^2) + \exp(-150(x^2 - y)^2 - 150(x - y^2)^2).$$

We use the random walk Metropolis-Hastings algorithm with a normal proposal distribution; that is, with the current value  $(x, y)$ , a new value  $(x', y')$  is proposed from

$$N\left(\begin{pmatrix} x \\ y \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right),$$

where  $\sigma^2 > 0$  and  $\rho \in (-1, 1)$ .

- i) © Visualize the unnormalized density  $g$  using a contour plot on a reasonable range.  
[Note: This step shows how the density looks like so that you don't make a mistake in the following steps. In higher-dimensional situations, visualization is usually much more difficult.]
- ii) © Using  $\sigma = 0.1$  and  $\rho = 0$ , generate  $n = 10^6$  MCMC draws starting from  $(-10, -10)$ . Let  $\mathbf{x} = (x_1, \dots, x_n)$  and  $\mathbf{y} = (y_1, \dots, y_n)$  be the generated chains for  $X$  and  $Y$ , respectively. Without (!) burn-in, give a scatter plot between  $\mathbf{x}$  and  $\mathbf{y}$ , a histogram for each of  $\mathbf{x}$  and  $\mathbf{y}$ , a trace plot for each of  $\mathbf{x}$  and  $\mathbf{y}$  (`ts.plot`), and an autocorrelation plot for each of  $\mathbf{x}$  and  $\mathbf{y}$  (`acf`).
- iii) © Repeat ii) with  $\sigma = 1$  and  $\rho = 0.8$ .
- iv) © Repeat ii) with  $\sigma = 1$  and  $\rho = -0.8$ .
- v) ⌚ Among the three combinations, determine which pair of  $(\sigma, \rho)$  is the best and which is the worst for this problem.

- vi) ☺ Based on the results above, explain why checking only a histogram, a trace plot and an autocorrelation plot can be dangerous when using MCMC.
- vii) ⓒ & ☺ Based on practical insights gained through trial-and-error, propose a set of values for  $(\sigma, \rho)$  that outperforms the best combination you determined above among the three. Provide visual representations illustrating the convergence performance of this recommendation. In light of the shapes of the target and proposal densities, explain why this recommendation led to improved performance.

**Q2.** Consider  $X_i \stackrel{\text{iid}}{\sim} \text{Exp}(1.5)$ ,  $i = 1, 2, 3$ , where 1.5 is the rate parameter of the exponential distribution. We want to generate  $(X_1, X_2, X_3)$  conditional on  $1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2$  using the Gibbs sampling.

- i) ☺ Show that each conditional distribution is the left-truncated distribution of a shifted exponential random variable, that is, for  $Y_i \stackrel{\text{iid}}{\sim} \text{Exp}(1.5)$ , there exist  $a_i$  and  $c_i$  such that

$$X_i | (X_j, j \neq i, 1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2) \stackrel{d}{=} (Y_i + a_i) | (Y_i \leq c_i), \quad i = 1, 2, 3,$$

where  $a_i$  and  $c_i$  depend on  $X_j$ ,  $j \neq i$ . Specify  $a_i$  and  $c_i$ ,  $i = 1, 2, 3$ .

- ii) ☺ Explain how to generate  $X_i$  from each conditional distribution of  $X_i | \{X_j, j \neq i\}$ ,  $i = 1, 2, 3$ , using the inverse transform method for truncated distributions.
- iii) ⓒ Generate  $10^6$  sets of  $(X_1, X_2, X_3)$  using the fixed-scan Gibbs sampling after some burn-in period. Store the values whenever every cycle is completed, i.e., obtain a sequence of vectors,

$$(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}), (X_1^{(2)}, X_2^{(2)}, X_3^{(2)}), (X_1^{(3)}, X_2^{(3)}, X_3^{(3)}), \dots,$$

rather than

$$(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}), (X_1^{(2)}, X_2^{(1)}, X_3^{(1)}), (X_1^{(2)}, X_2^{(2)}, X_3^{(1)}), \dots$$

Use a reasonable initial value and a burn-in period. Estimate the mean vector and the covariance matrix of the target distribution with the generated MCMC samples.

# STA3127 Statistical Computing

Jeong Geonwoo

DUE Monday, Dec 11

```
rm(list=ls())
set.seed(2018122062)
library(coda); library(scatterplot3d)
```

## Q1

We want to generate a sample from the unnormalized density

$$g(x) = \exp(-(x+1)^2 - y^2) + \exp\left(-150(x^2 - y)^2 - 150(x - y^2)^2\right).$$

We use the random walk Metropolis-Hastings algorithm with a normal proposal distribution; that is, with the current value  $(x, y)$ , a new value  $(x', y')$  is proposed from

$$N\left(\begin{pmatrix} x \\ y \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

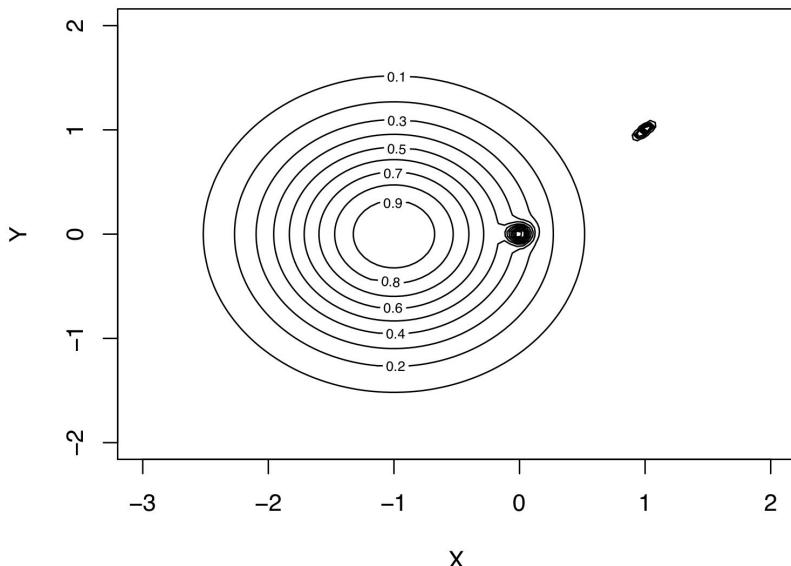
where  $\sigma^2 > 0$  and  $\rho \in (-1, 1)$ .

i) (C) Visualize the unnormalized density  $g$  using a contour plot on a reasonable range. [Note: This step shows how the density looks like so that you don't make a mistake in the following steps. In higher-dimensional situations, visualization is usually much more difficult.]

```
g <- function(x, y) {
  exp(-(x + 1)^2 - y^2) + exp(-150*(x^2 - y)^2 - 150*(x - y^2)^2)}

x <- seq(-3, 2, length.out = 100); y <- seq(-2, 2, length.out = 100)
z <- outer(x, y, g)
par(mfrow=c(1,1))
contour(x, y, z, xlab = "X", ylab = "Y",
        main="The contour plot of density g")
```

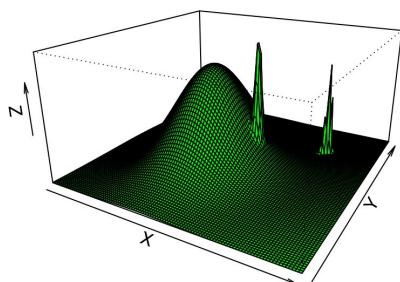
## The contour plot of density g



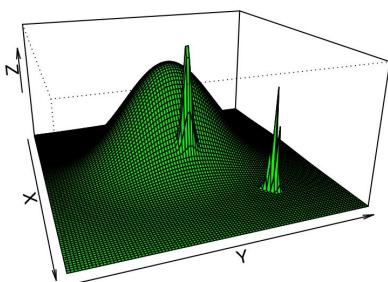
I will report additional visualization for understanding the density of  $g$ .

```
par(mfrow=c(1,2))
x <- seq(-3, 2, length.out = 100); y <- seq(-2, 2, length.out = 100)
z <- outer(x, y, g)
res1 <- persp(x, y, z, theta = 30, phi = 20, expand = 0.5,
               col = "green", xlab = "X", ylab = "Y", zlab = "Z",
               main = "The density plot of g")
res2 <- persp(x, y, z, theta = 70, phi = 20, expand = 0.5,
               col = "green", xlab = "X", ylab = "Y", zlab = "Z",
               main = "The density plot of g")
```

The density plot of  $g$



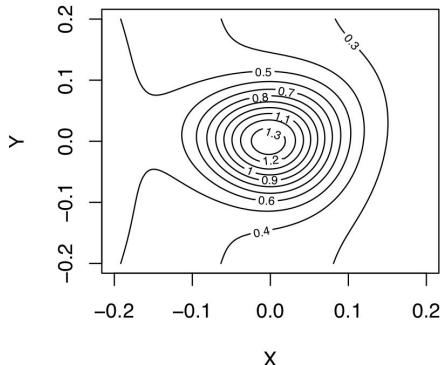
The density plot of  $g$



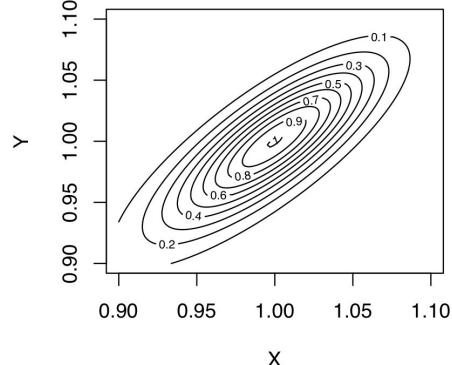
From the contour, we can see that the density of the function  $g$  has a bell shape centered at  $x,y=(-1,0)$ , and has a narrow and high density at  $x,y=(0,0)$  and  $x,y=(1,1)$ .

```
par(mfrow=c(1,2))
x <- seq(-0.2, 0.2, length.out = 100); y <- seq(-0.2, 0.2, length.out = 100)
z <- outer(x, y, g)
contour(x, y, z, xlab = "X", ylab = "Y",
        main="Density of g : around x,y=(0,0)")
x <- seq(0.9, 1.1, length.out = 100); y <- seq(0.9, 1.1, length.out = 100)
z <- outer(x, y, g)
contour(x, y, z, xlab = "X", ylab = "Y",
        main="Density of g : around x,y=(1,1)")
```

**Density of g : around x,y=(0,0)**



**Density of g : around x,y=(1,1)**



The two figures above are local visualizations for  $x,y = (0,0)$  and  $(1,1)$  where the function  $g$  has high density. We can see the shape and density, which are difficult to see in the overall contour.

ii) (C) Using  $\sigma = 0.1$  and  $\rho = 0$ , generate  $n = 10^6$  MCMC draws starting from  $(-10, -10)$ . Let  $\mathbf{x} = (x_1, \dots, x_n)$  and  $\mathbf{y} = (y_1, \dots, y_n)$  be the generated chains for  $X$  and  $Y$ , respectively. Without (!) burn-in, give a scatter plot between  $\mathbf{x}$  and  $\mathbf{y}$ , a histogram for each of  $\mathbf{x}$  and  $\mathbf{y}$ , a trace plot for each of  $\mathbf{x}$  and  $\mathbf{y}$  (`ts.plot`), and an autocorrelation plot for each of  $\mathbf{x}$  and  $\mathbf{y}$  (`acf`).

I will define the function used for sampling from multivariate normal distribution. (Because I can not use the `mvrnorm` function in this course, I use the Cholesky decomposition and Box-Muller Transformation.)

```
my_rmvnorm <- function(n, mean, sigma) {
  L <- chol(sigma) # Cholesky decomposition
  my_norm <- function(n) { # Box-Muller Transformation
    u1 <- runif(n/2); u2 <- runif(n/2)
    z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
    z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)
    return(c(z1, z2))
  }
  # Standard Normal distribution
  z <- matrix(my_norm(n * length(mean)), ncol = length(mean), byrow = TRUE)
  # cov matrix
  mean + z %*% L}
```

I will get MCMC samples of  $x, y$  using Metropolis-Hastings algorithm.

```

MCMC.MH <- function(n, init, sigma, rho) {
  x <- rep(NA, n); y <- rep(NA, n)
  x[1] <- init[1]; y[1] <- init[2]

  for (i in 2:n) {
    current <- c(x[i-1], y[i-1])
    proposal <- my_rmvnorm(1, mean = current,
                           sigma = matrix(c(1, rho, rho, 1) * sigma^2, ncol = 2))
    a <- g(proposal[1], proposal[2]) / g(current[1], current[2])

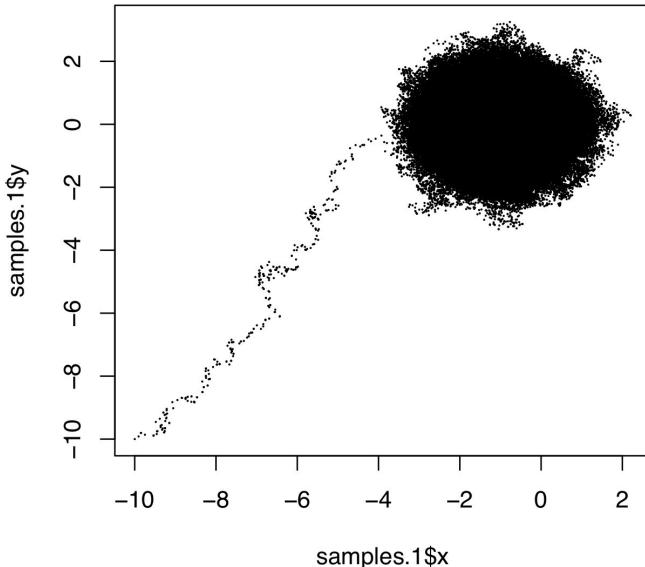
    if (runif(1) < a) {x[i] <- proposal[1]; y[i] <- proposal[2]}
    else {x[i] <- current[1]; y[i] <- current[2]}}
  return(list(x = x, y = y))
}

```

When  $\sigma = 0.1$ ,  $\rho = 0$ , I will check the samples

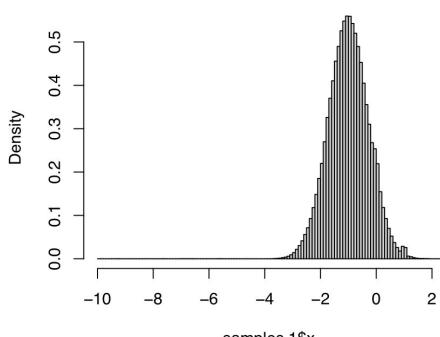
```
n <- 1e6
samples.1 <- MCMC.MH(n = n, init = c(-10, -10), sigma = 0.1, rho = 0)
par(mfrow=c(1,1))
plot(samples.1$x, samples.1$y, main = "Scatter plot of x and y", pch = 20, cex = 0.1)
```

Scatter plot of x and y

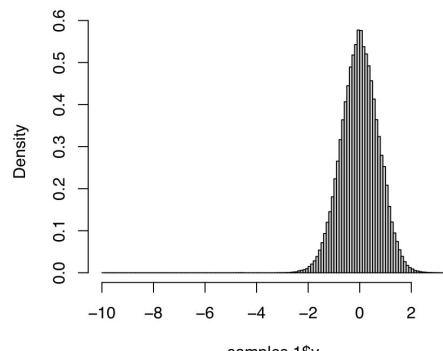


```
par(mfrow = c(1,2))
hist(samples.1$x, main = "Histogram of x", freq=FALSE, breaks=100)
hist(samples.1$y, main = "Histogram of y", freq=FALSE, breaks=100)
```

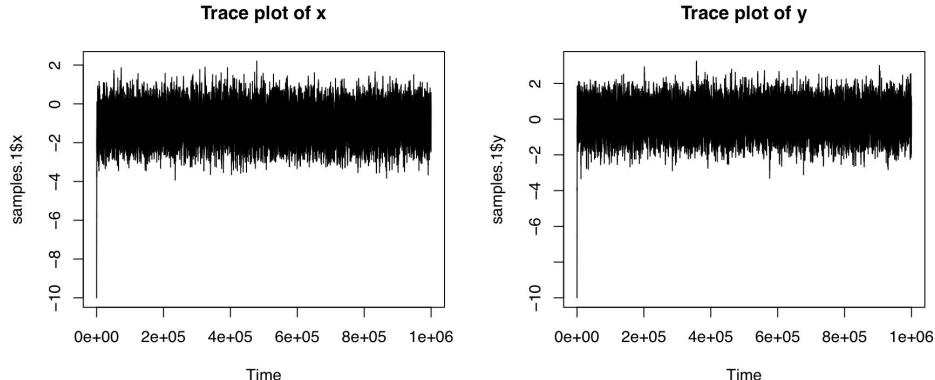
Histogram of x



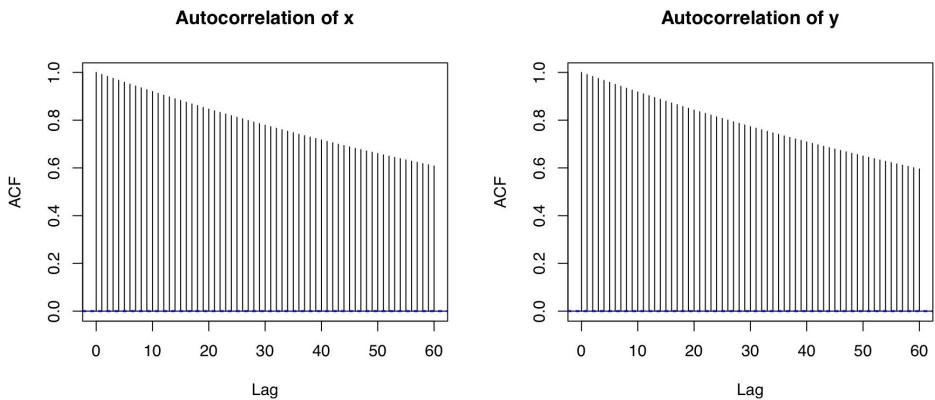
Histogram of y



```
ts.plot(samples.1$x, main = "Trace plot of x")
ts.plot(samples.1$y, main = "Trace plot of y")
```



```
acf(samples.1$x, main = "Autocorrelation of x")
acf(samples.1$y, main = "Autocorrelation of y")
```

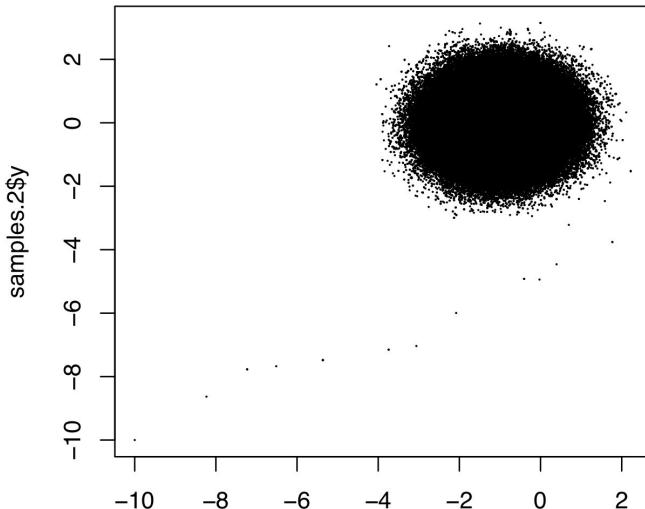


Interpretation : First, From the scatter plot, we can see that  $x,y$  starts at  $(-10,-10)$  and moves to a location where the density of the function  $g$  is high. Second, from the around  $x=1$  of the histogram, we can see that the function  $g$  is characterized. (not perfectly) Third, the trace plot shows that the samples are converging. Fourth, however, when from the ACF, too small  $\sigma = 0.1$  made the autocorrelation between the samples high, so there is a concern that effective sample size is too small.

iii) (C) Repeat ii) with  $\sigma = 1$  and  $\rho = 0.8$ .

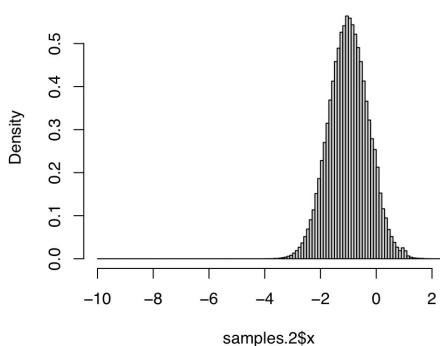
```
samples.2 <- MCMC.MH(n = n, init = c(-10, -10), sigma = 1, rho = 0.8)
par(mfrow=c(1,1))
plot(samples.2$x, samples.2$y, main = "Scatter plot of x and y", pch = 20, cex = 0.1)
```

**Scatter plot of x and y**

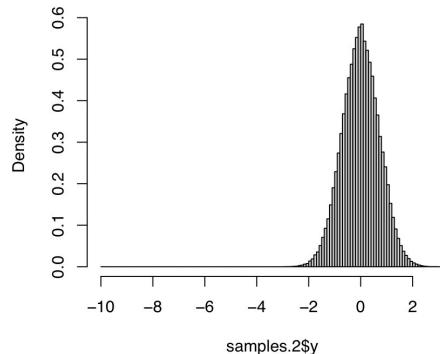


```
par(mfrow = c(1,2))
hist(samples.2$x, main = "Histogram of x", freq=FALSE, breaks=100)
hist(samples.2$y, main = "Histogram of y", freq=FALSE, breaks=100)
```

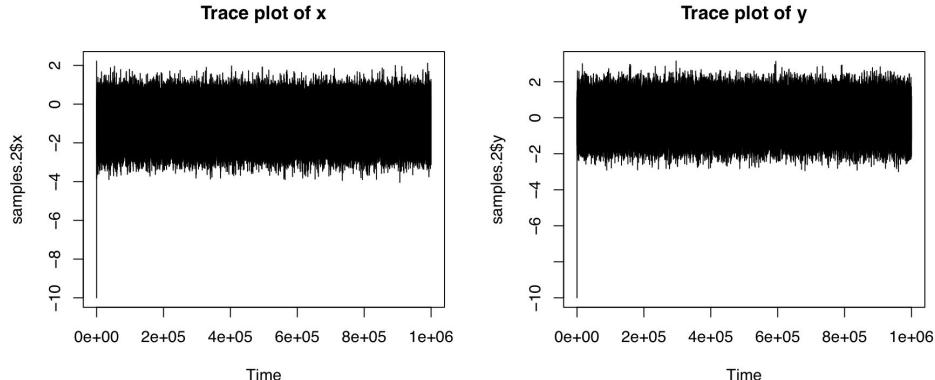
**Histogram of x**



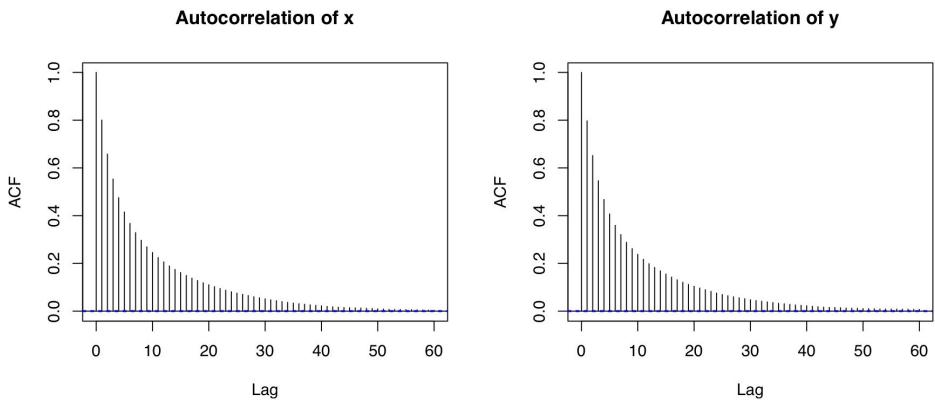
**Histogram of y**



```
ts.plot(samples.2$x, main = "Trace plot of x")
ts.plot(samples.2$y, main = "Trace plot of y")
```



```
acf(samples.2$x, main = "Autocorrelation of x")
acf(samples.2$y, main = "Autocorrelation of y")
```

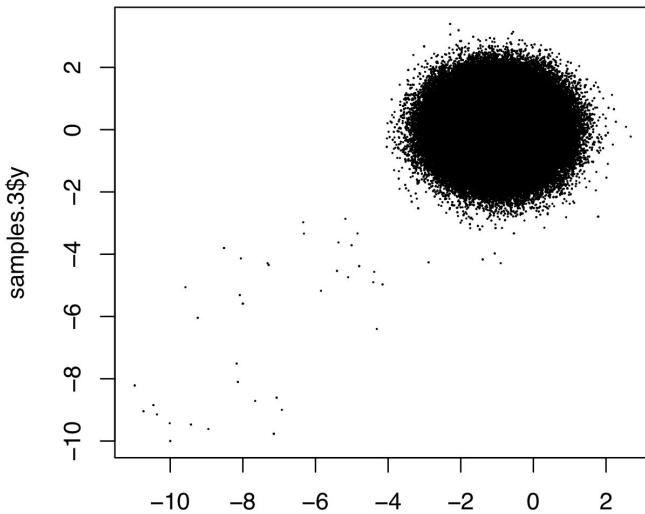


Interpretation : First, from the scatter plot, we can see that  $x,y$  starts at  $(-10,-10)$  and moves to a location where the density of the function  $g$  is high. It is important to emphasize that in fact that sample reached a location where  $g$  has high density (which means convergence region)  $<20$  iterations, **the convergence speed is faster** than in ii) ( $\sigma = 0.1, \rho = 0$ ). Second, the  $x=1$  part of the histogram shows that the function  $g$  is characterized. (not perfectly) Third, the trace plot shows that the samples are converging. Fourth, the ACF shows that the autocorrelation between the samples is high but decreasing rapidly.

iv) (C) Repeat ii) with  $\sigma = 1$  and  $\rho = -0.8$ .

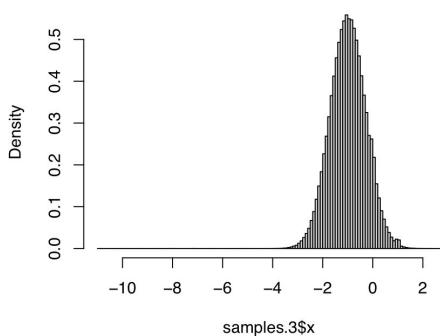
```
samples.3 <- MCMC.MH(n = n, init = c(-10, -10), sigma = 1, rho = -0.8)
par(mfrow=c(1,1))
plot(samples.3$x, samples.3$y, main = "Scatter plot of x and y", pch = 20, cex = 0.1)
```

**Scatter plot of x and y**

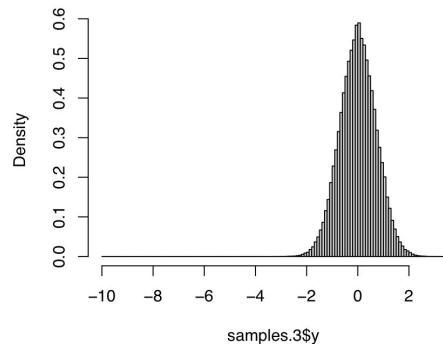


```
par(mfrow = c(1,2))
hist(samples.3$x, main = "Histogram of x", freq=FALSE, breaks=100)
hist(samples.3$y, main = "Histogram of y", freq=FALSE, breaks=100)
```

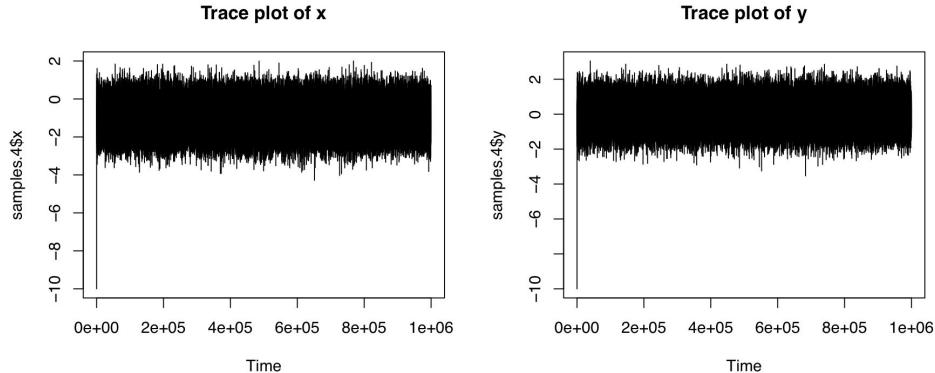
**Histogram of x**



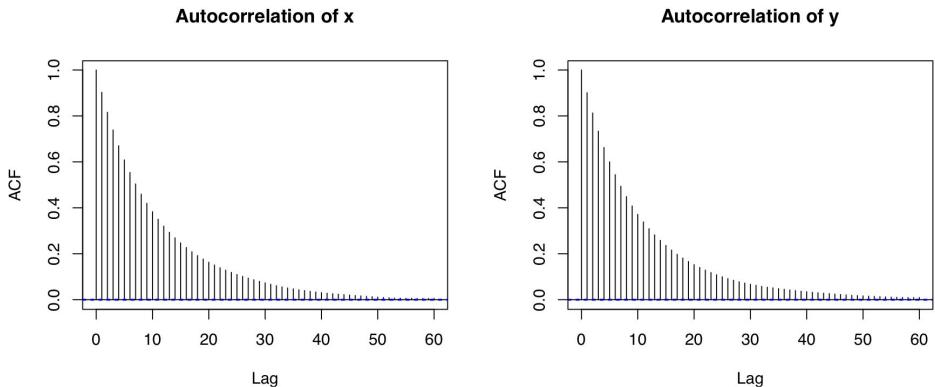
**Histogram of y**



```
ts.plot(samples.4$x, main = "Trace plot of x")
ts.plot(samples.4$y, main = "Trace plot of y")
```



```
acf(samples.4$x, main = "Autocorrelation of x")
acf(samples.4$y, main = "Autocorrelation of y")
```



As a result, the sample can move from  $x,y=(-10,-10)$  to a location with a high target density in only <10 iterations. However, this does not mean that  $\sigma$  should be very large. Because too large  $\sigma$  will cause the sample to move out of the high target density region after convergence, which is not acceptable so can cause large autocorrelation.

vi) Based on the results above, explain why checking only a histogram, a trace plot and an autocorrelation plot can be dangerous when using MCMC.

As discussed above, we can see from the scatter plot that the convergence speed of x,y is different in ii), iii), and iv). In ii) ( $\sigma = 0.1, \rho = 0$ ),  $\sigma$  was too small and suggested a sample too close to the current sample, so many samples were needed for x,y to reach from (-10,10) to near (-1,0). Also, in iv) ( $\sigma = 1, \rho = -0.8$ ), both x and y need to grow in order for x,y to move from (-10,-10) to near (-1,0), but due to  $\rho = -0.8 < 0$ , x and y move in opposite directions from the current sample, so many samples are needed for reaching to near (-1,0).

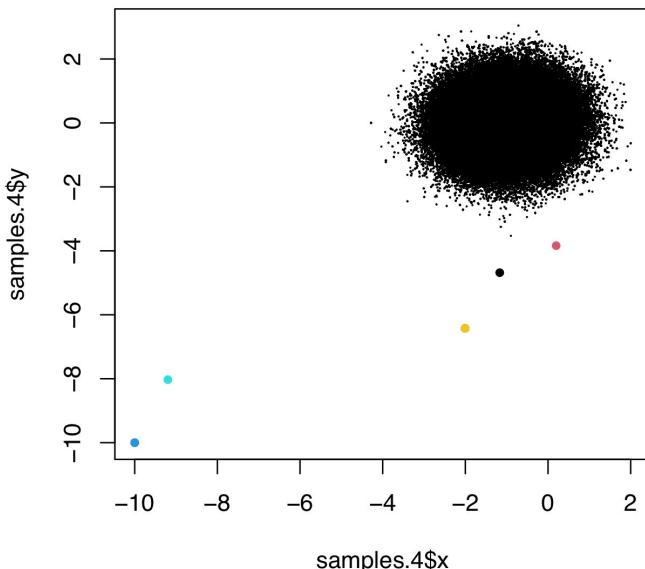
However, these results are difficult to see from the histogram, trace plot, and autocorrelation plot because they show the distribution of each of the resulting x,y after all the sampling is done. This makes it difficult to understand what is happening at the beginning of an iteration (especially when the sample size(iteration) is very large, as in this problem). Therefore, for an efficient MCMC algorithm, it is not only important to check for convergence through histograms, trace plots, and autocorrelation plots, but also to consider how quickly the samples are approaching the target distribution through scatter plots. This is especially important when the initial sample is arbitrarily chosen due to lack of information about the target distribution.

vii) (C) & Based on practical insights gained through trial-and-error, propose a set of values for  $(\sigma, \rho)$  that outperforms the best combination you determined above among the three. Provide visual representations illustrating the convergence performance of this recommendation. In light of the shapes of the target and proposal densities, explain why this recommendation led to improved performance.

```
samples.4 <- MCMC.MH(n = n, init = c(-10, -10), sigma = 4, rho = 0.8)
par(mfrow=c(1,1))
plot(samples.4$x, samples.4$y, main = "Scatter plot of x and y", pch = 20, cex = 0.1)

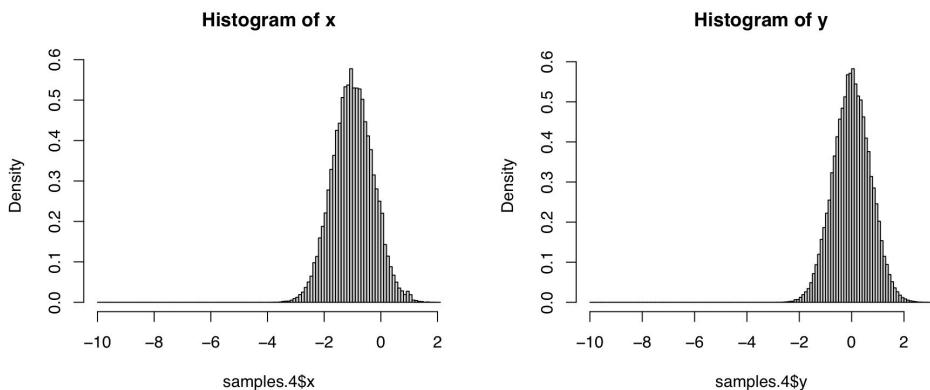
for (i in 1:10){points(samples.4$x[i], samples.4$y[i], col = i, pch = 20, cex = 1)}
```

### Scatter plot of x and y

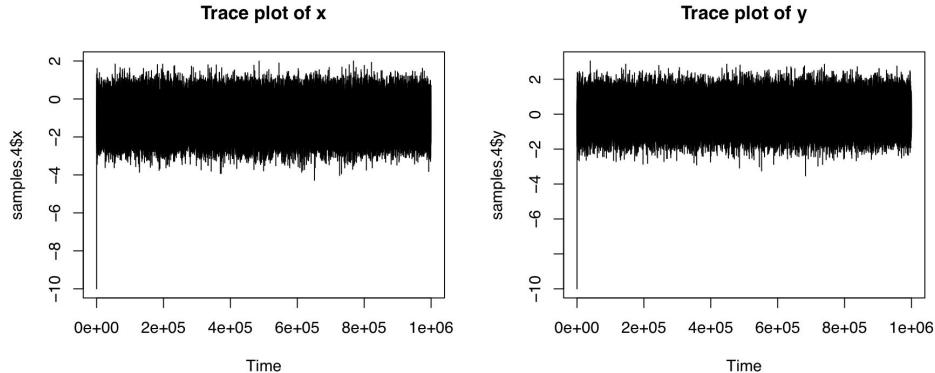


I set  $(\sigma = 4, \rho = 0.8)$  and colored the first 10 samples. Considering the density of the target density, I set  $\sigma$  to a large value so that the sample can quickly move from  $x,y=(-10,-10)$  to a location with a high target density. Also, from the fact that the proposal density is multivariate normal distribution (i.e., elliptical bell-shaped), I set  $\rho > 0$  so that when the sample moves from the current location to the next location, the probability that  $x$  and  $y$  move in a similar direction is high.

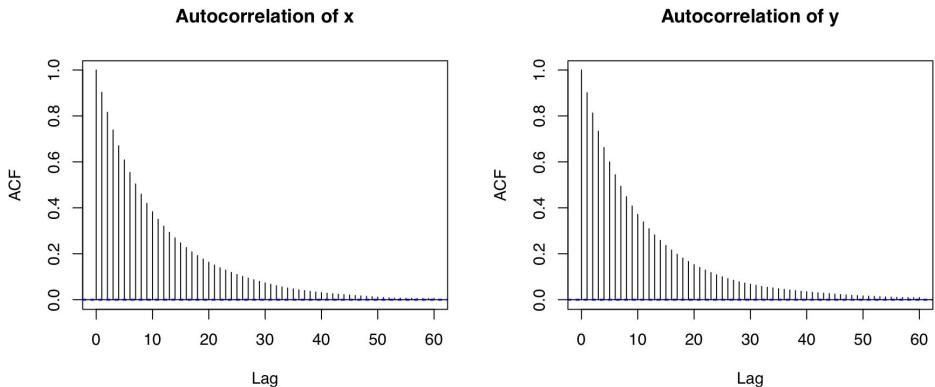
```
par(mfrow = c(1,2))
hist(samples.4$x, main = "Histogram of x", freq=FALSE, breaks=100)
hist(samples.4$y, main = "Histogram of y", freq=FALSE, breaks=100)
```



```
ts.plot(samples.4$x, main = "Trace plot of x")
ts.plot(samples.4$y, main = "Trace plot of y")
```



```
acf(samples.4$x, main = "Autocorrelation of x")
acf(samples.4$y, main = "Autocorrelation of y")
```



As a result, the sample can move from  $x,y=(-10,-10)$  to a location with a high target density in only <10 iterations. However, this does not mean that  $\sigma$  should be very large. Because too large  $\sigma$  will cause the sample to move out of the high target density region after convergence, which is not acceptable so can cause large autocorrelation.

## Q2

Consider  $X_i \stackrel{\text{iid}}{\sim} \text{Exp}(1.5)$ ,  $i = 1, 2, 3$ , where 1.5 is the rate parameter of the exponential distribution. We want to generate  $(X_1, X_2, X_3)$  conditional on  $1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2$  using the Gibbs sampling.

i) Show that each conditional distribution is the left-truncated distribution of a shifted exponential random variable, that is, for  $Y_i \stackrel{\text{iid}}{\sim} \text{Exp}(1.5)$ , there exist  $a_i$  and  $c_i$  such that

$$X_i \mid (X_j, j \neq i, 1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2) \stackrel{d}{=} (Y_i + a_i) \mid (Y_i \leq c_i), \quad i = 1, 2, 3,$$

Let  $S = X_1^2 + 2X_2^2 + 3X_3^2$

i=1)

$$\begin{aligned} & X_1 \mid (X_2, X_3, 1 < s < 2) \\ &= X_1 \mid X_2, X_3, 1 - 2X_2^2 - 3X_3^2 < X_1^2 < 2 - 2X_2^2 - 3X_3^2 \\ &= X_1 \mid X_2, X_3, \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)} < X_1 < \sqrt{2 - 2X_2^2 - 3X_3^2} \\ & \text{Let } A = \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)}, B = \sqrt{2 - 2X_2^2 - 3X_3^2} \\ & f_{X_1 \mid X_2, X_3, A < X_1 < B}(k) \\ &= \frac{f_{X_1}(k)}{P(A < X_1 < B)} \\ &= \frac{1.5e^{-1.5k}}{(1 - e^{-1.5B}) - (1 - e^{-1.5A})} \\ &= \frac{1.5e^{-1.5k}}{e^{-1.5A} - e^{-1.5B}} = \frac{1.5e^{-1.5(k-A)}}{1 - e^{-1.5(B-A)}} \\ & f_{Y_1+a_1 \mid Y_1 \leq c_1}(k) \\ &= \frac{f_{Y_1+a_1}(k)}{P(Y_1 \leq c_1)} = \frac{1.5e^{-1.5(k-a_1)}}{1 - e^{-1.5c_1}} \end{aligned}$$

$$\therefore X_1 \mid (X_2, X_3, 1 < S < 2) \stackrel{d}{=} (Y_1 + a_1) \mid Y_1 \leq c_1$$

where  $a_1 = A = \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)}$ ,  $c_1 = B - A = \sqrt{2 - 2X_2^2 - 3X_3^2} - \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)}$

i=2)

$X_2 \mid (X_1, X_3, 1 < S < 2)$

$$\begin{aligned} &= X_2 \mid X_1, X_3, 1 - X_1^2 - 3X_3^2 < 2X_2^2 < 2 - X_1^2 - 3X_3^2 \\ &= X_2 \mid X_1, X_3, \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)} < X_2 < \sqrt{\frac{2 - X_1^2 - 3X_3^2}{2}} \\ & \text{Let } A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)}, B = \sqrt{\frac{2 - X_1^2 - 3X_3^2}{2}} \end{aligned}$$

$$\therefore X_2 \mid (X_1, X_3, 1 < S < 2) \stackrel{d}{=} (Y_2 + a_2) \mid Y_2 \leq c_2$$

where  $a_2 = A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)}$ ,  $c_2 = B - A = \sqrt{\frac{2 - X_1^2 - 3X_3^2}{2}} - \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)}$

i=3)

$$\begin{aligned}
& X_3 \mid (X_1, X_2, 1 < S < 2) \\
& = X_3 \mid X_1, X_2, 1 - X_1^2 - 2X_2^2 < 3X_3^2 < 2 - X_1^2 - 2X_2^2 \\
& = X_3 \mid X_1, X_2, \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)} < X_3 < \sqrt{\frac{2 - X_1^2 - 2X_2^2}{3}} \\
& \text{Let } A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)}, B = \sqrt{\frac{2 - X_1^2 - 2X_2^2}{3}} \\
& \therefore X_3 \mid (X_1, X_2, 1 < S < 2) \stackrel{d}{=} Y_3 + a_3 \mid Y_3 \leq c_3 \\
& \text{where } a_3 = A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)}, c_3 = B - A = \sqrt{\frac{2 - X_1^2 - 2X_2^2}{3}} - \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)}
\end{aligned}$$

ii) Explain how to generate  $X_i$  from each conditional distribution of  $X_i \mid \{X_j, j \neq i\}$ ,  $i = 1, 2, 3$ , using the inverse transform method for truncated distributions.

From i), we saw that :

$$\begin{aligned}
& X_1 \mid (X_2, X_3, 1 < S < 2) \stackrel{d}{=} (Y_1 + a_1) \mid Y_1 \leq c_1 \\
& \text{where } a_1 = A = \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)}, c_1 = B - A = \sqrt{2 - 2X_2^2 - 3X_3^2} - \sqrt{\max(0, 1 - 2X_2^2 - 3X_3^2)} \\
& X_2 \mid (X_1, X_3, 1 < S < 2) \stackrel{d}{=} (Y_2 + a_2) \mid Y_2 \leq c_2 \\
& \text{where } a_2 = A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)}, c_2 = B - A = \sqrt{\frac{2 - X_1^2 - 3X_3^2}{2}} - \sqrt{\max\left(0, \frac{1 - X_1^2 - 3X_3^2}{2}\right)} \\
& X_3 \mid (X_1, X_2, 1 < S < 2) \stackrel{d}{=} Y_3 + a_3 \mid Y_3 \leq c_3 \\
& \text{where } a_3 = A = \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)}, c_3 = B - A = \sqrt{\frac{2 - X_1^2 - 2X_2^2}{3}} - \sqrt{\max\left(0, \frac{1 - X_1^2 - 2X_2^2}{3}\right)}
\end{aligned}$$

Therefore, we can generate  $X_i$  from each conditional distribution of  $X_i \mid \{X_j, j \neq i\}, i = 1, 2, 3$  by i) generating  $Y_i \mid Y_i < c_i$  from truncated exponential distribution and ii) add  $a_i$ , respectively. Details are follow :

i) Consider the cumulative distribution function (CDF) of the exponential distribution and its inverse. The CDF of an exponential distribution is as follows:

$$\begin{aligned}
F(x; \lambda) &= 1 - e^{-\lambda x} \\
\text{where } \lambda &\text{ is rate parameter}
\end{aligned}$$

The inverse CDF of exponential distribution is as follows:

$$\begin{aligned}
u &= F(x; \lambda) \\
&= 1 - e^{-\lambda x} \\
x &= F^{-1}(u; \lambda) \\
&= -\frac{1}{\lambda} \log(1 - u)
\end{aligned}$$

So if we want to generate  $Y_i$  from truncated exponential distribution (which means  $Y_i \mid a < Y_i < b$ ) using inverse transformation method, we should generate  $u \sim \text{Unif}(F(a), F(b))$  where  $F$  is the CDF of exponential

distribution,  $a < b$ . Then, if we input generated  $u$  into  $F^{-1}(u)$  as above, we can get the random generated sample from truncated exponential distribution. In this problem, we generate  $u \sim \text{Unif}(F(0), F(c_i))$  which means  $Y_i \mid Y_i \leq c_i$

ii) Finally, we just add  $a_i$  to  $Y_i$ . Then we can get  $Y_i + a_i \mid Y_i \leq c_i$ , which are  $X_i \mid \{X_j, j \neq i\}, i = 1, 2, 3$

iii) (C) Generate  $10^6$  sets of  $(X_1, X_2, X_3)$  using the fixed-scan Gibbs sampling after some burn-in period. Store the values whenever every cycle is completed, i.e., obtain a sequence of vectors,

$$\left(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}\right), \left(X_1^{(2)}, X_2^{(2)}, X_3^{(2)}\right), \left(X_1^{(3)}, X_2^{(3)}, X_3^{(3)}\right), \dots,$$

rather than

$$\left(X_1^{(1)}, X_2^{(1)}, X_3^{(1)}\right), \left(X_1^{(2)}, X_2^{(1)}, X_3^{(1)}\right), \left(X_1^{(2)}, X_2^{(2)}, X_3^{(1)}\right), \dots$$

Use a reasonable initial value and a burn-in period. Estimate the mean vector and the covariance matrix of the target distribution with the generated MCMC samples.

```
# Random generating From truncated exponential distribution
r.trunc.exp <- function(rate, lower, upper) {
  # CDF of exponential distribution at lower and upper
  F_lower <- 1 - exp(-rate * lower)
  F_upper <- 1 - exp(-rate * upper)

  # runif from (F_lower, F_upper)
  p <- runif(1) * (F_upper - F_lower) + F_lower

  # Truncated sample by inverse CDF of exponential distribution
  x <- -log(1 - p) / rate

  return(x)
}
```

Above function can generate the sample from truncated exponential distribution. The generating process is described in Q2 ii).

```
# arbitrary initial value
X <- rep(0.5, 3)
# sample size (iteration)
n <- 1e6
# burn in size
burn_in <- 1e2

# sample space
samples <- matrix(NA, nrow = n, ncol = 3)
```

I set the arbitrary initial values that satisfy  $1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2$ . i.e.  $0.5^2 + 2 \cdot 0.5^2 + 3 \cdot 0.5^2 = 1.5$

```
# Gibbs sampling
for (i in 1:n) {
  a1 <- sqrt(max(0, 1 - 2*X[2]^2 - 3*X[3]^2))
  c1 <- sqrt(2 - 2*X[2]^2 - 3*X[3]^2) - a1
  X[1] <- r.trunc.exp(1.5, 0, c1) + a1

  a2 <- sqrt(max(0, (1 - X[1]^2 - 3*X[3]^2)/2))
  c2 <- sqrt((2 - X[1]^2 - 3*X[3]^2)/2) - a2
  X[2] <- r.trunc.exp(1.5, 0, c2) + a2
```

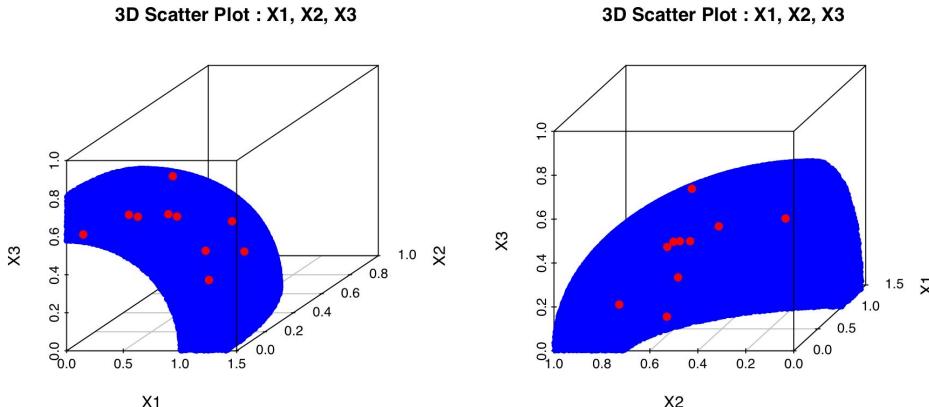
```

a3 <- sqrt(max(0, (1 - X[1]^2 - 2*X[2]^2)/3))
c3 <- sqrt((2 - X[1]^2 - 2*X[2]^2)/3) - a3
X[3] <- r.trunc.exp(1.5, 0, c3) + a3
samples[i,] <- X
}

# burn-in
samples <- samples[(burn_in + 1):n, ]

par(mfrow=c(1,2))
s3d <- scatterplot3d(samples[,1], samples[,2], samples[,3],
                     main="3D Scatter Plot : X1, X2, X3",
                     xlab="X1", ylab="X2", zlab="X3",
                     color="blue", pch=20, angle=45)
s3d$points3d(samples[1:10,1], samples[1:10,2], samples[1:10,3],
              col="red", pch=19)
s3d2 <- scatterplot3d(samples[,1], samples[,2], samples[,3],
                      main="3D Scatter Plot : X1, X2, X3",
                      xlab="X1", ylab="X2", zlab="X3",
                      color="blue", pch=20, angle=225)
s3d2$points3d(samples[1:10,1], samples[1:10,2], samples[1:10,3],
               col="red", pch=19)

```



We can see the each datapoint(each row of samples) is plotted as a blue dot. After burn-in period, I colored first 10 samples red. From the figure above, we can see that the first 10 samples are well included in the overall distribution of samples.

```

# mean vector and covariance matrix
mean_vector <- apply(samples, 2, mean)
covariance_matrix <- cov(samples)

print(mean_vector)

## [1] 0.5172439 0.4262889 0.3736444
print(covariance_matrix)

```

```

##          [,1]      [,2]      [,3]
## [1,] 0.13296838 -0.03504182 -0.03234351
## [2,] -0.03504182  0.07187958 -0.02947898
## [3,] -0.03234351 -0.02947898  0.04871589

```

We can see that the mean of  $X_1, X_2, X_3$  satisfy  $1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2$ . Also, we can see the non-diagonal elements of the covariance matrix are less than 0, which means that when one of  $X_1, X_2, X_3$  become large, the other two become smaller due to the condition on the weighted sum of squares  $1 < X_1^2 + 2X_2^2 + 3X_3^2 < 2$ .