

R Handout: Descriptive Statistics

This handout will introduce a basic method of simulation. This pdf file and the R program are provided on Blackboard.

1. Prerequisite: Probability Integral Transform (PIT)

Suppose that a random variable X has a continuous distribution for which the cumulative distribution function (CDF) is F_X . Then the random variable Y defined as

$$Y = F_X(X) \sim \text{Uniform}(0, 1)$$

- R code

```
# Set a seed to make this process repeatable
set.seed(12345)

# Set the number of simulations
n <- 100000

# Simulate data from a standard normal distribution
X <- rnorm(n)

# Find out the corresponding quantiles
Y <- pnorm(X)

# Use histogram to graph the distribution
hist(Y, freq = F, ylim = c(0, 1.5))
```

2. Simulate data from a standard normal distribution

(a) Direct simulation

- R code

```
# Set a seed to make this process repeatable
set.seed(12345)

# Set the number of simulations
n <- 100000

# Simulate data from a standard normal distribution
X <- rnorm(n)

# Use histogram to graph the distribution
hist(X, freq = F)
```

(b) Indirect simulation by PIT

- R code

```
# Set a seed to make this process repeatable
set.seed(12345)

# Set the number of simulations
n <- 100000

# Simulate data from Uniform(0,1)
Y <- runif(n)

# By PIT, get a sample from a standard normal distribution
X <- qnorm(Y)

# Use histogram to graph the distribution
hist(X, freq = F)
```

3. Simulate data from a contaminated normal distribution. Suppose we have a contaminated normal distribution as section 3.4.1 in the 7th edition. Because $W = ZI_{1-\epsilon} + \sigma_c Z(1 - I_{1-\epsilon})$, we can consider this distribution is consisted by "jumping" between two normal distributions. One is $N(0, 1)$ and another is $N(0, \sigma_c)$, and the "button" triggering the "jump" is $I_{1-\epsilon}$. We need to simulate this process to get the distribution.

- R code

```
# Set a seed to make this process repeatable
set.seed(12345)

# Set the number of simulations
n <- 100000

# Build a function to simulate the "jumping" process
rctnorm <- function(n, eps = 0.5, mu = c(0, 0),
sigma = c(1, 1)) {

# Define the "jumping" process
jump <- sample(c(1, 2), prob = c(1 - eps, eps),
size = n, replace = TRUE)
# Simulate data with "jumps"
rnorm(n, mean = mu[jump], sd = sigma[jump])

}

# 3.4.26 part b, eps = 0.15, mu1 = mu2 = 0,
# sigma1 = 1, sigma2 = sigma.c = 10
X1 <- rctnorm(n, 0.15, mu = c(0, 0), sigma = c(1, 10))
# Use histogram to graph the distribution
hist(X1, freq = F)

# 3.4.26 part b, eps = 0.15, mu1 = mu2 = 0,
# sigma1 = 1, sigma2 = sigma.c = 20
X2 <- rctnorm(n, 0.15, mu = c(0, 0), sigma = c(1, 20))
# Use histogram to graph the distribution
hist(X2, freq = F)

# 3.4.26 part b, eps = 0.25, mu1 = mu2 = 0,
# sigma1 = 1, sigma2 = sigma.c = 20
```

```

X3 <- rctnorm(n, 0.25, mu = c(0, 0), sigma = c(1, 20))
# Use histogram to graph the distribution
hist(X3, freq = F)

# An example when we have 2 normal distribution
# with different locations. eps = 0.5, mu1 = 0,
# mu2 = 2, sigma1 = sigma2 = 1
X4 <- rctnorm(n, 0.5, mu = c(0, 5), sigma = c(1, 1))
hist(X4, freq = F)

```

4. Practice: Simulate data from a gamma distribution with $\alpha = 2$ and $\beta = 5$. (You will need to use `rgamma` and `help(rgamma)` in R)