R Handout: Simulation

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

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

   - 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 coresponding 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 have this definition

$$W = XI_{1-\epsilon} + Y(1 - I_{1-\epsilon})$$

where $X \sim N(0, 1)$, $Y \sim N(0, \sigma_c^2)$ and $I \sim Bernoulli(1 - eps)$

- 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 a contaminated normal distribution
rctnorm <- function(n, eps = 0.5, mu = c(0, 0),
sigma = c(1, 1)) {

# Simulate I
I <- rbinom(n, 1, 1 - eps)
# Simulate X
X <- rnorm(n, mu[1], sigma[1])
# Simulate Y
Y <- rnorm(n, mu[2], sigma[2])

# Simulate W
W <- rep(NA, n)
for (i in 1:n){
W[i] <- ifelse(I[i] == 1, X[i], Y[i])
}

W

}

# the p.d.f of contaminated normal distribution
dctnorm <- function(x, eps = 0.5, mu = c(0, 0),
sigma = c(1, 1)){

dnorm(x, mu[1], sigma[1]) * (1 - eps) +
```

```
dnorm(x, mu[2], sigma[2]) * eps

}

x <- seq(-100, 100, 0.01)

par(mfrow = c(1, 1))

# 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, main = 'eps = 0.15 and sigma.c = 10'
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 10), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.15, c(0, 0), c(1, 10)), type = 'l'
, col = 'black')

# 3.4.26 part c, 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, main = 'eps = 0.15 and sigma.c = 20'
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 20), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.15, c(0, 0), c(1, 20)), type = 'l'
, col = 'black')

# 3.4.26 part d, 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, main = 'eps = 0.25 and sigma.c = 20'
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 20), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.25, c(0, 0), c(1, 20)), type = 'l'
, col = 'black')

# An example when we have 2 normal distribution
# with different locations. eps = 0.5, mu1 = 0,
# mu2 = 5, sigma1 = sigma2 = 1
X4 <- rctnorm(n, 0.5, mu = c(0, 5), sigma = c(1, 1))
hist(X4, freq = F, main = 'eps = 0.5 with different locations
and same sigmas', ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 5, 1), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.5, c(0, 5), c(1, 1)), type = 'l'
, col = 'black')

# Comparing these histograms
# Define a 2 by 2 graph
par(mfrow = c(2, 2))
hist(X1, freq = F, main = 'eps = 0.15 and sigma.c = 10'
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 10), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.15, c(0, 0), c(1, 10)), type = 'l'
, col = 'black')
hist(X2, freq = F, main = 'eps = 0.15 and sigma.c = 20'
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 20), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.15, c(0, 0), c(1, 20)), type = 'l'
, col = 'black')
hist(X3, freq = F, main = 'eps = 0.25 and sigma.c = 20'
```

```
, ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 0, 20), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.25, c(0, 0), c(1, 20)), type = 'l'
, col = 'black')
hist(X4, freq = F, main = 'eps = 0.5 with different locations
and same sigmas', ylim = c(0, 0.45))
points(x, dnorm(x, 0, 1), type = 'l', col = 'red')
points(x, dnorm(x, 5, 1), type = 'l', col = 'blue')
points(x, dctnorm(x, 0.5, c(0, 5), c(1, 1)), type = 'l'
, col = 'black')
```

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)