hw 5 solution

Statistical Computing, Jieun Shin

Autumn 2022

문제 1.

(a) 로지스틱 분포 $f(x) = \frac{e^x}{(1+e^x)^2}, \quad -\infty < x < \infty$를 따르는 확률난수를 역변환법에 의해 발생시키고자 한다.

먼저 로지스틱 분포의 cdf

$$F_X(x) = \mathbb{P}(X \le x) = \int_{-\infty}^{x} \frac{e^x}{(1+e^x)^2} dt = \int_0^{e^x} \frac{1}{(1+y)^2} dy = 1 - \frac{1}{1+e^x}.$$

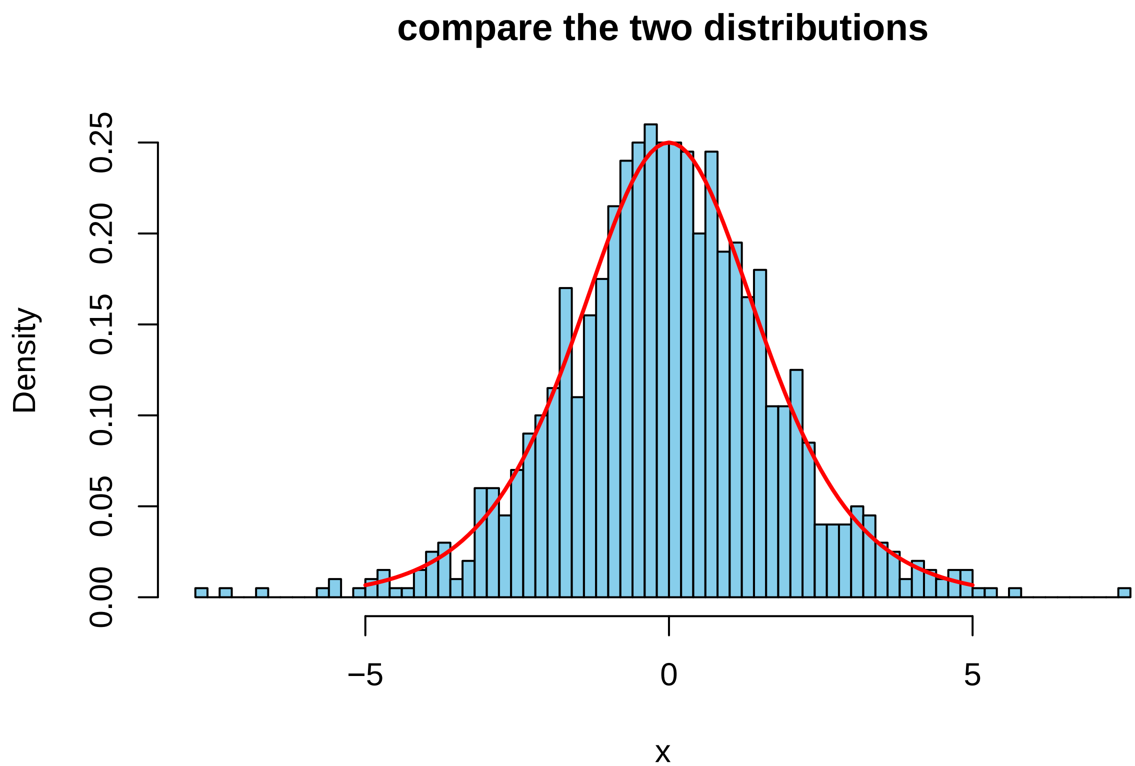
를 구하고 균일난수에 의해 대응되는 cdf의 역함수값이 f를 따르는 난수가 된다. 알고리즘으로 작성하면 다음과 같다.

1. $u \sim \text{unif}(0,1)$
2. $x = \log\left(\frac{1}{1-u} - 1\right)$

(b) 두 히스토그램을 비교해 보았을 때 역볍환볍으로 생성한 분포 (파란색)가 이론적 분포 (붉은색)와 비슷하게 생성됨을 확인할 수 있다.

R 코드

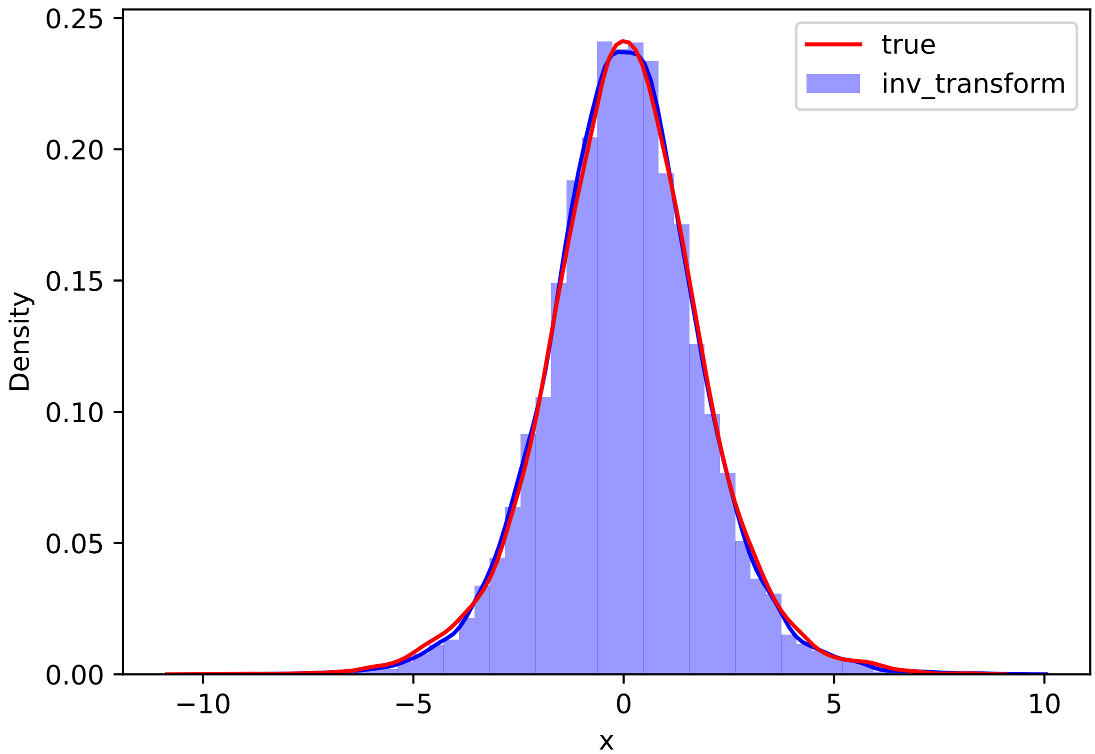
```r
set.seed(123)
N = 1000
u = runif(N)
x = log(1/(1 - u) - 1)

# 그래프 비교
hist(x, breaks = 100, col = "skyblue", freq = FALSE,
     main = "compare the two distributions", xlab = "x") # 히스토그램: 역변환법으로 생성한 분포
curve(dlogis, from = -5, to = 5, add = TRUE, col = 'red', lwd = 2)  # curve: 이론적 분포
```



파이썬 코드

```
import numpy as np
from numpy import random
import scipy.integrate as sci
import matplotlib.pyplot as plt
import seaborn as sns

# 균일난수 발생
n = 10000
u1 = []

for i in range(0, n):
  u1.append(random.uniform(0, 1))

u1 = np.array(u1)

# 역변환법으로 로지스틱분포 난수 생성
x = np.log(1/(1-u1) - 1)

# 실제 분포에서 난수 생성
true_x = random.logistic(loc = 0, scale = 1, size = n)

sns.distplot(x, color = "blue", hist = True, kde = True,
label = 'inv_transform')        # 히스토그램: 역변환법으로 생성한 분포
sns.distplot(true_x, color = "red", hist = False, kde = True,
label = 'true')   # curve: 이론적 분포
```

```
plt.xlabel('x')
plt.ylabel('Density')
plt.show()
```



문제 2.

(a) 값이 5를 넘는다는 조건이 주어졌을 때 Gamma(5, 1)를 따르는 난수를 발생시키기 위한 기각법 알고리즘을 유도해보자.

목표가 되는 분포를 $f(x) = \frac{xe^{-x}}{6e^{-5}}, x \geq 5$라 하자. 기각법 알고리즘을 위해 $f(x) = C \cdot h(x) \cdot g(x), \quad C \geq 1$로 나누어야 한다. 여기서 $h(x) = \frac{e^{-x/2}/2}{e^{-5/2}}, x \geq 5$라 하면 각 $C \geq 1$와 $0 \leq g(x) \leq 1$을 만족하는 $C, g(x)$를 다음과 같이 설정할 수 있다:

$$C = \frac{1}{e^{-5/2}}, \quad h(x) = \frac{e^{-x/2}/2}{e^{-5/2}}, \quad g(x) = xe^{-x/2}/3.$$

여기서 $h(x)$를 따르는 난수는 역변환법에 의해 구한다. $h(x)$의 cdf $H(x) = \int_5^x \frac{e^{-t/2}/2}{e^{-5/2}} dt = -e^{5/2}e^{-x/2} + 1$이고, 균일난수 u를 생성하여 확률난수 $X = -2\log(1-u)+5$를 생성한다. 그러면 $f(x)$을 따르는 확률난수를 발생시킬 기각법 알고리즘은 다음과 같다.

(1) $u \sim \text{unif}(0, 1)$

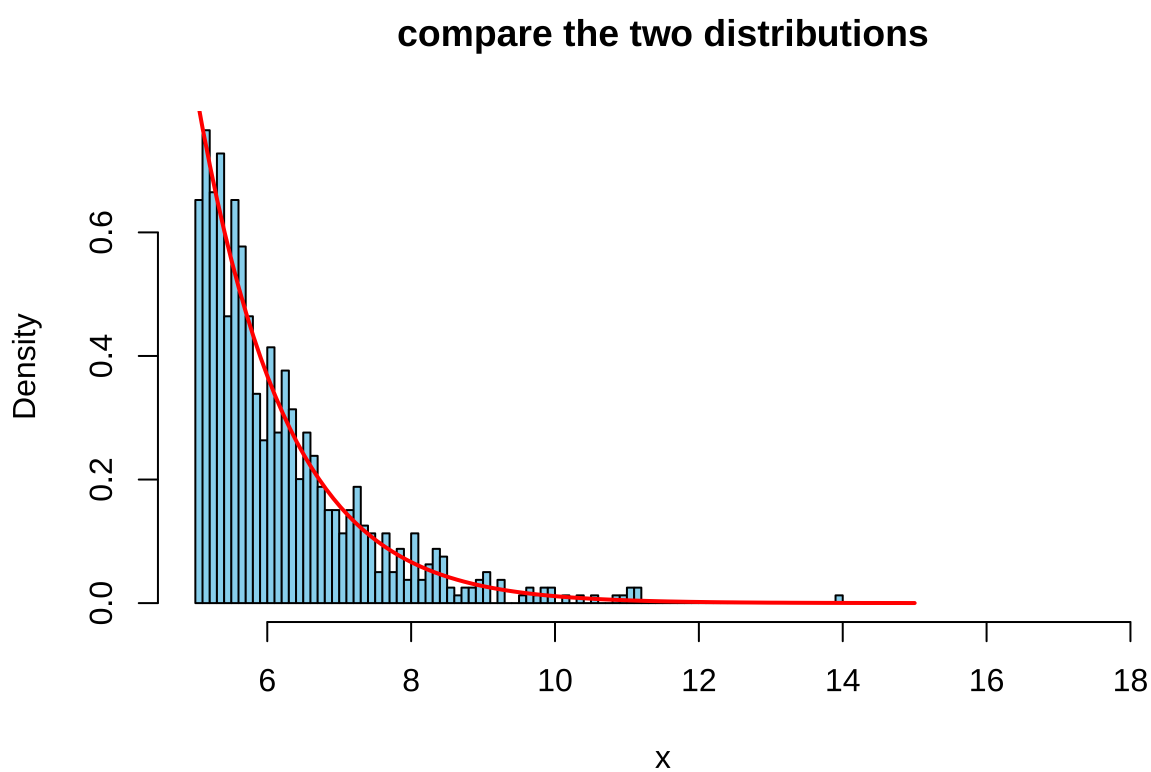
(2) $x = -2\log(1-u) + 5$

(3) 만약 $u \leq \frac{h(x)}{Cf(x)}$이면 x를 반환하고, 그렇지 않으면 (1)로 돌아간다.

(b) 두 히스토그램을 비교해 보았을 때 기각법으로 생성한 분포 (파란색)가 이론적 분포 (붉은색)와 비슷하게 생성됨을 확인할 수 있다.

R 코드

```
set.seed(123)
N = 10000

# algorithm
u1 =  runif(N)        # step 1
u2 =  runif(N)
x = -2*log(1-u1) + 5 # step 2
g = x * exp(-x/2) /3

f = ifelse(u2 <= g, x, NA)  # step 3
f = f[!is.na(f)]

# 그래프 비교
hist(f, breaks = 70, col = "skyblue", freq = FALSE, xlim = c(5, 18),
     main = "compare the two distributions", xlab = "x")   #  히스토그램: 기각법으로 생성한 분포
f_true = function(x) x*exp(-x)*exp(5) / 6
xx = seq(5, 15, length.out = 100)
points(xx, f_true(xx), type = 'l', col = "red", lwd = 2)                 # curve: 이론적 분포
```



파이썬 코드

```
import numpy as np
from numpy import random
import scipy.integrate as sci
import matplotlib.pyplot as plt
```

```
import seaborn as sns

# 균일난수 발생
n = 10000
u1 = []
u2 = []
for i in range(0, n):     # step 1
  u1.append(random.uniform(0, 1))
  u2.append(random.uniform(0, 1))

u1 = np.array(u1)         # step 2
u2 = np.array(u2)

# 역변환법으로 h(x)따르는 난수 생성
x = -2 * np.log(1-u1) + 5
g = x * np.exp(-x/2) /3

f = []
for i in range(0, n):
  if u2[i] <= g[i]:
    f.append(x[i])       # step 3

# 참인 분포 생성
def f_true (x):
  return x*np.exp(-x)*np.exp(5)/6


# 그래프 비교
xx = np.arange(1, 18)

g1 = sns.distplot(f, color = "blue", hist = True, kde = True,
label = 'rejection')       # 히스토그램: 기각법으로 생성한 분포
g2 = sns.lineplot(xx, f_true(xx), color = "red", label = 'true')  # curve: 이론적 분포
g1.set(xlim = (5, None))
```

```
## [(5, 17.8)]
```

```
g2.set(xlim = (5, None))
```

```
## [(5, 17.8)]
```

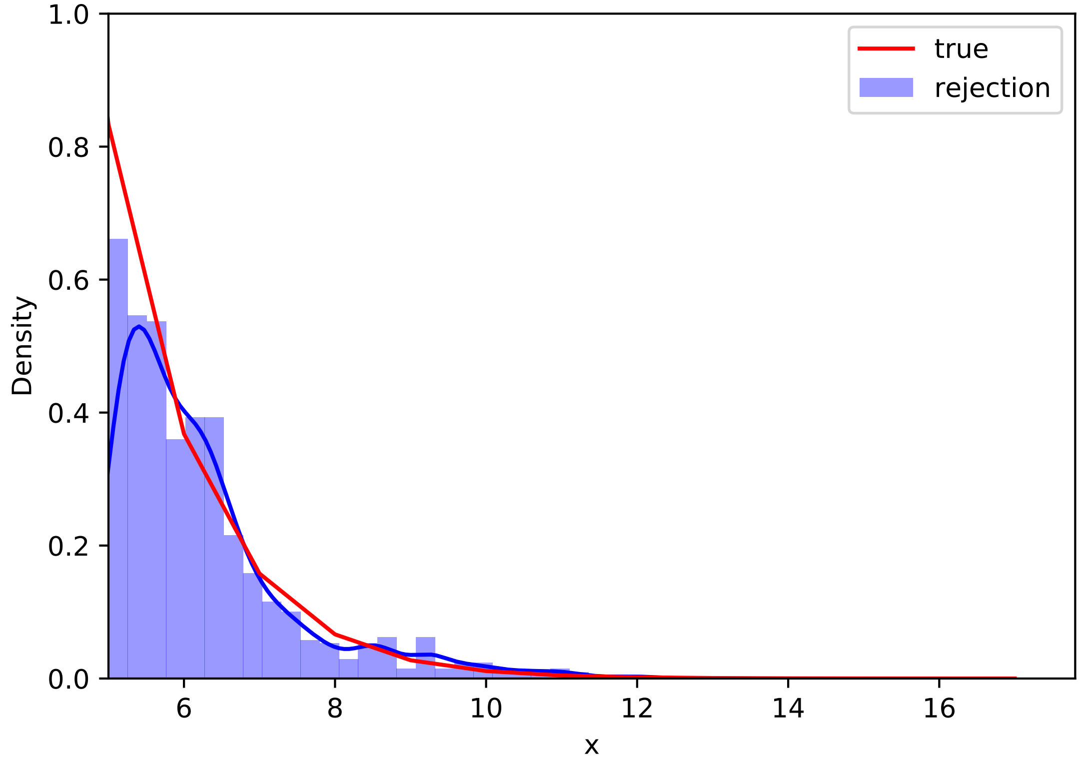
```
g1.set(ylim = (0, 1))
```

```
## [(0, 1)]
```

```
g2.set(ylim = (0, 1))
```

```
## [(0, 1)]
```

```
plt.xlabel('x')
plt.ylabel('Density')
plt.show()
```



문제 3.

(a) $f(x) = \sum_{i=1}^{3} p_i \frac{1}{\sigma_i} \phi\left(\frac{x-\mu_i}{\sigma_i}\right)$는 평균과 분산이 각각 μ_i, σ_i^2인 세 정규분포의 혼합분포의 밀도함수이다. 각 정규분포의 cdf를 $\Phi_i(\frac{x-\mu_i}{\sigma_i}) = \int_{-\infty}^{x} \frac{1}{\sigma_i} \phi\left(\frac{t-\mu_i}{\sigma_i}\right) dt$ 라고 하면 균일난수 u로부터 역변환법을 이용하여 각 정규분포를 따르는 확률난수 $\Phi_i^{-1}(u) = \frac{x-\mu_i}{\sigma_i}$를 생성할 수 있다.

따라서 각 정규분포에서 발생할 확률이 p_i임을 이용하여 $f(x)$을 따르는 난수를 생성하는 분해법 알고리즘을 다음과 같이 유도할 수 있다.

(1) $u_1, u_2 \sim \text{unif}(0,1)$

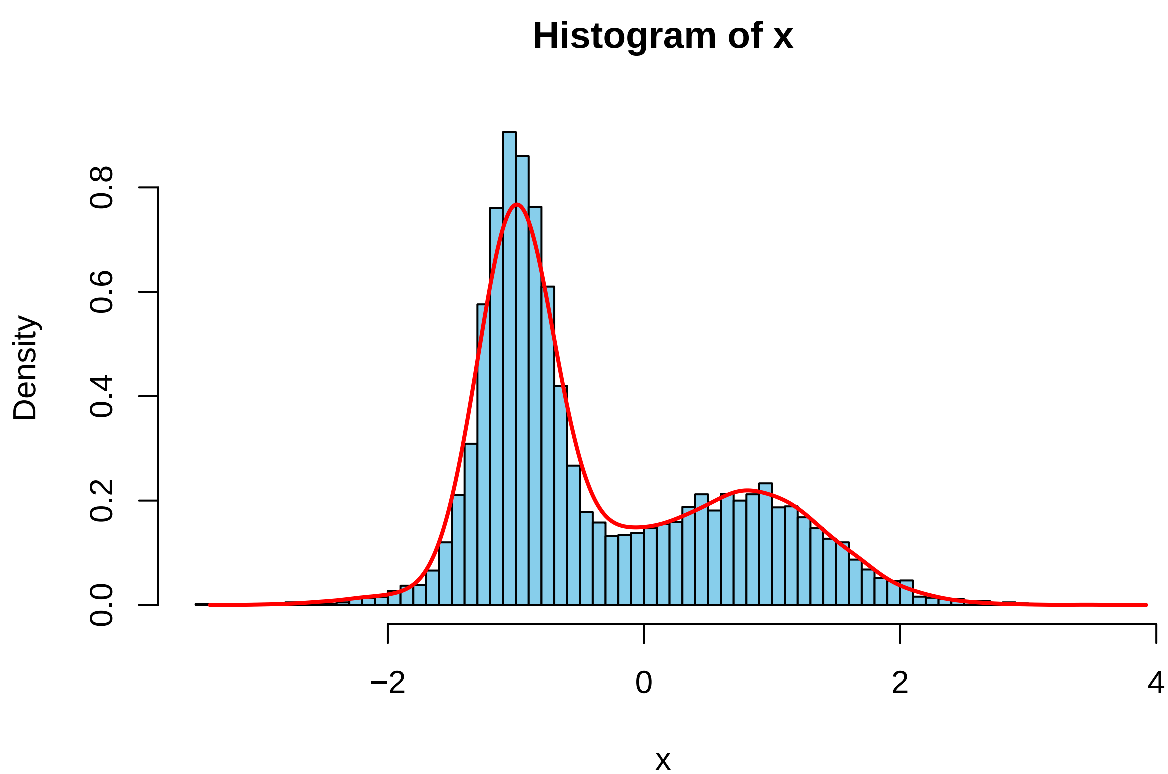
(2)

$$X = \begin{cases} \phi_1^{-1}(u_2), & \text{if } u_1 < 1/2 \\ \phi_2^{-1}(u_2), & \text{if } 1/2 \le u_1 < 5/6 \\ \phi_3^{-1}(u_2), & \text{if } u_1 \ge 5/6 \end{cases}$$

(b) 두 히스토그램을 비교해 보았을 때 분해법으로 생성한 분포 (파란색)가 이론적 분포 (붉은색)와 비슷하게 생성됨을 확인할 수 있다. **R 코드**

```
n = 10000
prob = c(1/2, 1/3, 1/6)
ni = floor(n*prob)
mui = c(-1, 0, 1)
sigi = c(1/4, 1, 1/2)

x = c()
for(i in 1:n){
```

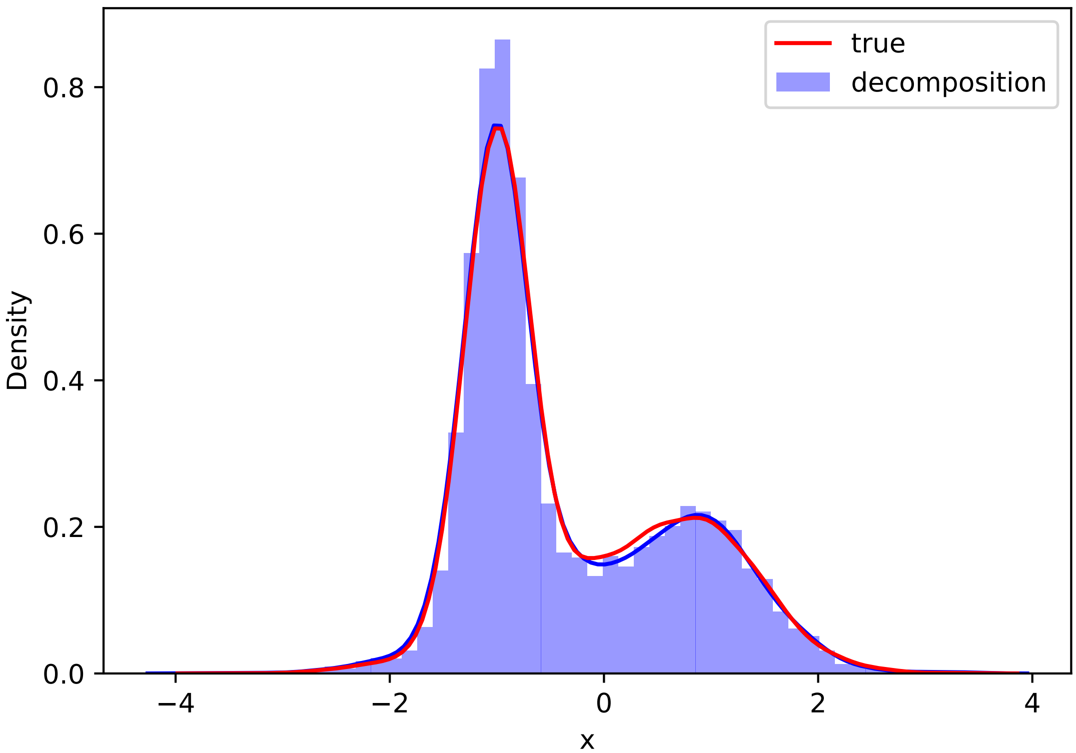
```r
  u1 = runif(1)
  u2 = runif(1)

  if(u1 < 1/2){
    x[i] = qnorm(u2, mui[1], sigi[1])
  } else if(1/2 <= u1 & u1 < 5/6){
    x[i] = qnorm(u2, mui[2], sigi[2])
  } else if(5/6 <= u1){
    x[i] = qnorm(u2, mui[3], sigi[3])
  }
}

# 참인 분포
true_x = sapply(1:3, function(j) rnorm(ni[j], mui[j], sigi[j]))
true_x = do.call(c, true_x)

hist(x, breaks=100, freq = FALSE, col = "skyblue")   # 히스토그램: 분해법으로 생성한 분포
lines(x = density(x = true_x), col = "red", lwd = 2)  # curve: 이론적 분포
```



파이썬 코드

```python
import numpy as np
from numpy import random
from scipy.stats import norm
import scipy.integrate as sci
import matplotlib.pyplot as plt
```

```python
import seaborn as sns

# 균일난수 발생
n = 10000
# prob = np.array([1/2, 1/3, 1/6])
# ni = np.dot(n, prob)
# mui = np.array([-1, 0, 1])
# sigi = np.array([1/4, 1, 1/2])

prob = [1/2, 1/3, 1/6]
ni = [n * prob[i] for i in range(len(prob))]
mui = [-1, 0, 1]
sigi = [1/4, 1, 1/2]


x = []
for i in range(0, n):
  u1 = random.uniform(0, 1)
  u2 = random.uniform(0, 1)

  if u1 < 1/2:
    x.append(norm.ppf(u2, loc = mui[0], scale = sigi[0]))
  elif 1/2 <= u1 and u1 < 5/6:
    x.append(norm.ppf(u2, loc = mui[1], scale = sigi[1]))
  elif 5/6 <= u1:
    x.append(norm.ppf(u2, loc = mui[2], scale = sigi[2]))

# 참인 분포
true_x = [random.normal(loc = mui[0], scale = sigi[0], size = int(ni[0])),
          random.normal(loc = mui[1], scale = sigi[1], size = int(ni[1])),
          random.normal(loc = mui[2], scale = sigi[2], size = int(ni[2]))]
true_x = np.concatenate(true_x)

sns.distplot(x, color = "blue", hist = True, kde = True,
label = 'decomposition')      # 히스토그램: 분해법으로 생성한 분포
sns.distplot(true_x, color = "red", hist = False, kde = True,
label = 'true')        # curve: 이론적 분포
plt.xlabel('x')
plt.ylabel('Density')
plt.show()
```



true
decomposition
Density
0.0
0.2
0.4
0.6
0.8
−4
−2
0
2
4
x