

StatComp_hw1

凌浩东

10/3/2021

3.3

The inverse is

$$F^{-1}(x) = \frac{b}{(1 - y)^{\frac{1}{a}}}$$

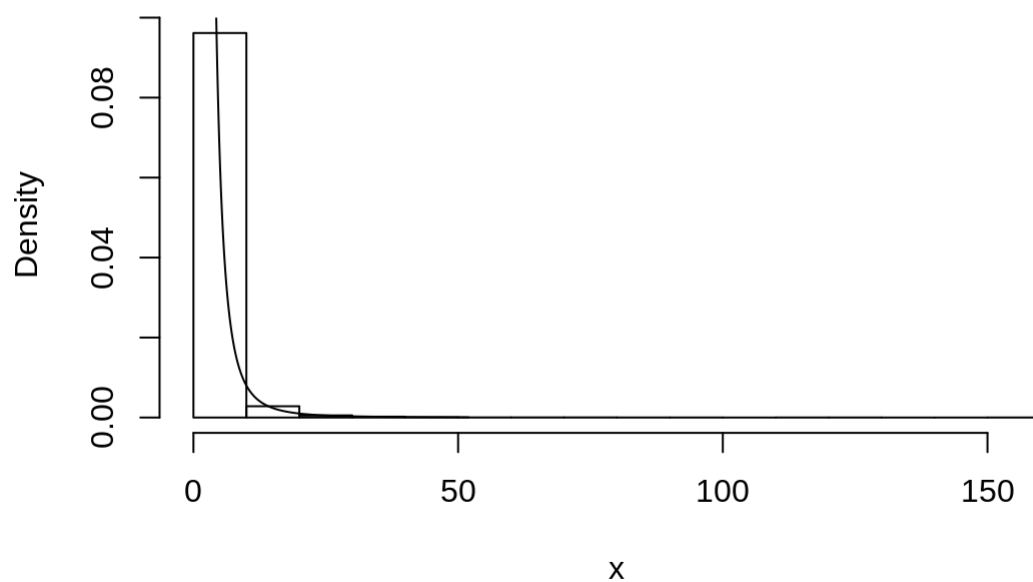
.

Use the inverse transformation method to simulate a random sample from Pareto(2, 2) distribution:

```
library(ggplot2)
set.seed(42)

n <- 10000
u <- runif(n)
x <- 2 / sqrt(1 - u)
hist(x, probability = T)
y <- seq(2, 52, 0.01)
lines(y, 8 / y^3)
```

Histogram of x



3.5

```

rpmf <- function(n) {
  # return a random sample from the pmf
  # given by the question with sample size n
  u <- runif(n)
  x = integer(n)
  for (i in 1:n) {
    if (0.1 < u[i] & u[i] <= 0.3)
      x[i] = 1
    else if (0.3 < u[i] & u[i] <= 0.5)
      x[i] = 2
    else if (0.5 < u[i] & u[i] <= 0.7)
      x[i] = 3
    else if (0.7 < u[i] & u[i] <= 1)
      x[i] = 4
  }
  x
}

n <- 1000
x <- rpmf(n)
y <- sample(c(0:4), size = n, replace = T, prob = c(0.1,0.2,0.2,0.2,0.3))
round(rbind(c(0.1,0.2,0.2,0.2,0.3), table(x)/n, table(y)/n), 3)

```

```

##           0      1      2      3      4
## [1,] 0.100 0.200 0.200 0.200 0.300
## [2,] 0.105 0.183 0.198 0.206 0.308
## [3,] 0.103 0.207 0.188 0.199 0.303

```

3.6

In the discrete case,

$$P(\text{accept}|Y) = P(U < \frac{f(Y)}{cg(Y)}|Y) = \frac{f(Y)}{cg(Y)}$$

The last equality comes from U is uniform distributed and $\frac{f(t)}{cg(t)} \leq 1$ for any t such that $f(t) \geq 0$.

The total probability of acceptance for ant iteration is

$$P(\text{accept}) = \sum_y P(\text{accept}|y)P(y) = \sum_y \frac{f(y)}{cg(y)}g(y) = \frac{1}{c}$$

Hence, the probability that the accept variants generated from Acceptance-rejection method equals to k is

$$P(k|\text{accept}) = \frac{P(\text{accept}|k)g(k)}{P(\text{accept})} = \frac{\frac{f(k)}{cg(k)}g(k)}{\frac{1}{c}} = f(k)$$

Thus, in the discrete case, the distribution of the accept variants generated from Acceptance-rejection method is same to the target distribution.

In the continuous case,

$$P(\text{accept}|Y \leq y) = \frac{\int_{-\infty}^y \frac{f(t)}{cg(t)} g(t) dt}{G(y)} = \frac{F(y)}{cG(y)}$$

$$\begin{aligned} P(\text{accept}) &= P\left(U \leq \frac{f(Y)}{cG(Y)}\right) = E\left[\mathbb{I}\left(U \leq \frac{f(Y)}{cG(Y)}\right)\right] \\ &= E\left[E\left[\mathbb{I}\left(U \leq \frac{f(Y)}{cG(Y)}\right) \middle| Y\right]\right] \\ &= E\left[P\left(U \leq \frac{f(Y)}{cG(Y)} \middle| Y\right)\right] \\ &= E\left[\frac{f(Y)}{cG(Y)}\right] \\ &= \int \frac{f(y)}{cg(y)} g(y) dy \\ &= \frac{1}{c} \end{aligned}$$

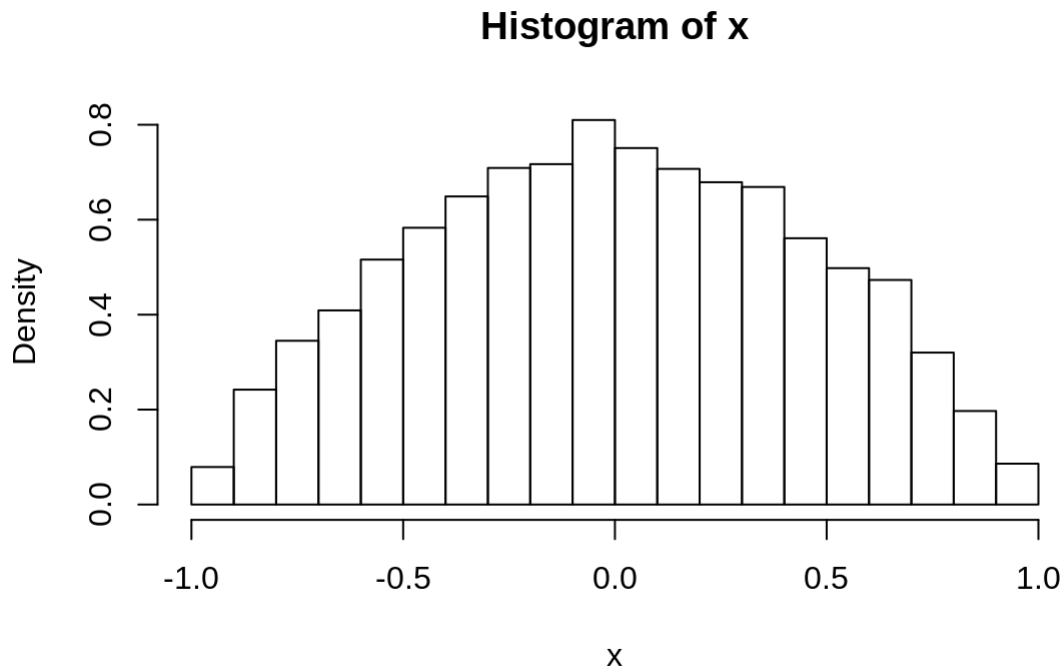
$$P(Y \leq k|\text{accept}) = \frac{P(\text{accept}|Y \leq k)P(Y \leq k)}{P(\text{accept})} = \frac{\frac{F(y)}{cG(y)}G(y)}{\frac{1}{c}} = F(y)$$

Hence, the distribution of the accept variants generated from Acceptance-rejection method is same to the target distribution.

3.9

```
rREK <- function(n) {
  u1 <- runif(n, -1, 1)
  u2 <- runif(n, -1, 1)
  u3 <- runif(n, -1, 1)
  x <- integer(n)
  for (i in 1:n) {
    if (abs(u3[i]) > abs(u2[i]) & abs(u3[i]) > abs(u1[i])) {
      x[i] <- u2[i]
    } else {
      x[i] <- u3[i]
    }
  }
  x
}

n <- 10000
x <- rREK(n)
hist(x, probability = T)
```



3.10

The essence of the algorithm is

- generate three random variables from standard uniform distribution.
- randomly choose one from the two with smaller value.
- negate it with probability $\frac{1}{2}$.

Suppose we choose X and $X \leq t$. If the three absolute value of the random variables are all smaller than t , then the probability is

$$t^3$$

If there are only two variables smaller than t , the probability is

$$\binom{3}{2} t^2 (1 - t) = 3t^2(1 - t)$$

If there are only one variables smaller than t , then this variable has $\frac{1}{2}$ chance to be X . Hence the probability of $X \leq t$ is

$$\frac{1}{2} \binom{3}{1} t(1 - t)^2 = \frac{3}{2} t(1 - t)^2$$

Thus the probability of X smaller than t is

$$P(X \leq t) = t^3 + 3t^2(1 - t) + \frac{3}{2} t(1 - t)^2 = \frac{3}{2} t - \frac{1}{2} t^3$$

for $t \in [0, 1]$.

With $\frac{1}{2}$ probability to negate, assume

$$Y = \begin{cases} X & \frac{1}{2} \\ -X & \frac{1}{2} \end{cases}$$

When $t \in [0, 1]$,

$$P(-X \leq t) = 1$$

$$P(Y \leq t) = \frac{1}{2} \left(\frac{3}{2}t - \frac{1}{2}t^3 \right) + \frac{1}{2} \times 1 = \frac{3}{4}t - \frac{1}{4}t^3 + \frac{1}{2}$$

When $t \in [-1, 0]$,

$$P(X \leq t) = 0$$

$$P(Y \leq t) = \frac{1}{2}P(-X \leq t) = \frac{1}{2}(1 - P(X \geq -t)) = \frac{1}{2} \left(1 - \frac{3}{2}(-t) - \frac{1}{2}(-t)^3 \right) = \frac{3}{4}t - \frac{1}{4}t^3 + \frac{1}{2}$$

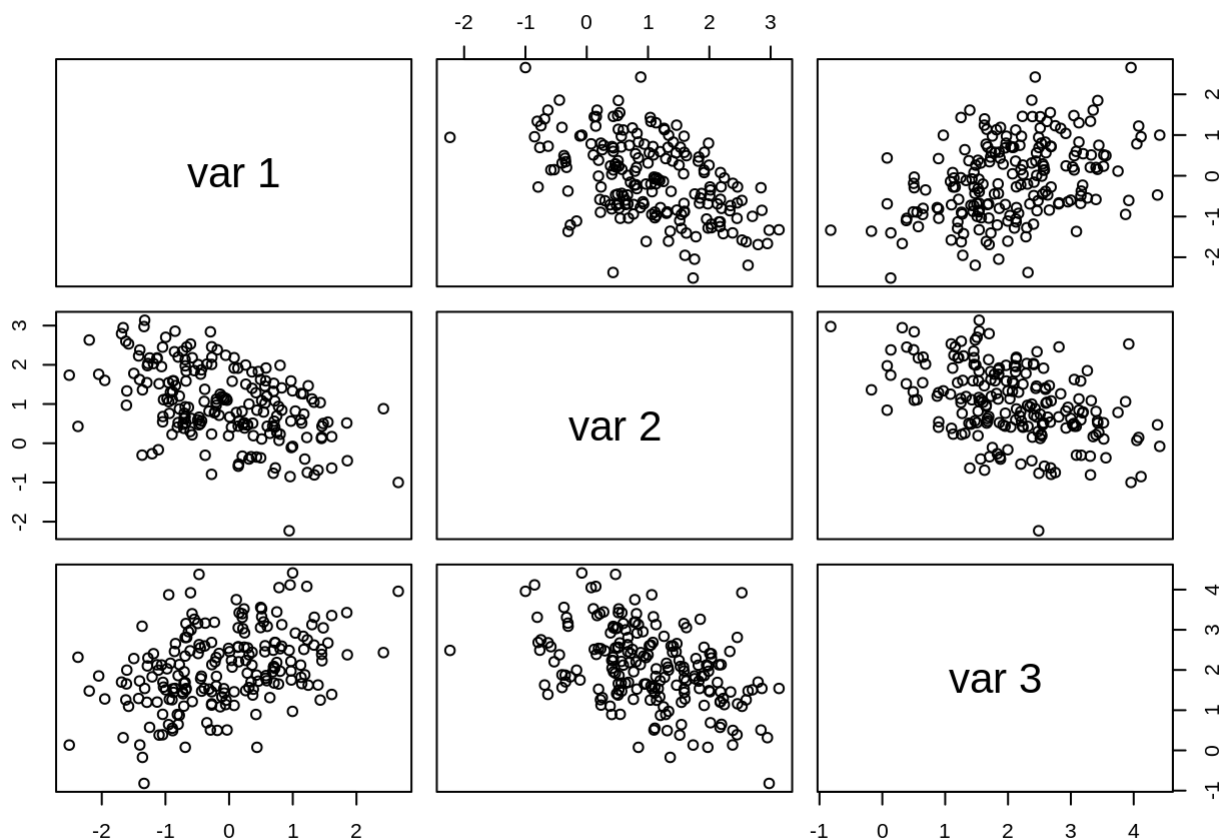
Thus $P(Y \leq t) = \frac{3}{4}t - \frac{1}{4}t^3 + \frac{1}{2}$ for $t \in [-1, 1]$.

$$f(t) = \frac{3}{4} - \frac{1}{4}t^2$$

for $t \in [-1, 1]$.

3.14

```
mu <- c(0, 1, 2)
sigma <- matrix(c(1, -0.5, 0.5, -0.5, 1, -0.5, 0.5, -0.5, 1), nrow = 3, ncol = 3)
rmvn.Choleski <- function(n, mu, sigma) {
  d <- length(mu)
  Q <- chol(sigma)
  Z <- matrix(rnorm(n*d), nrow = n, ncol = d)
  X <- Z %*% Q + matrix(mu, n, d, byrow = T)
  X
}
X <- rmvn.Choleski(200, mu, sigma)
pairs(X)
```



5.2

```
x <- seq(.1, 2.5, length = 10)
m <- 10000
cdf <- numeric(length(x))
for (i in 1:length(x)) {
  u <- runif(m, 0, x[i])
  g <- exp(- u^2/2)
  cdf[i] <- x[i] * mean(g) / sqrt(2*pi) + 0.5
}
Phi <- pnorm(x)
print(round(rbind(x, cdf, Phi), 3))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## x    0.10 0.367 0.633 0.900 1.167 1.433 1.700 1.967 2.233 2.500
## cdf 0.54 0.643 0.737 0.816 0.879 0.923 0.954 0.974 0.986 1.000
## Phi 0.54 0.643 0.737 0.816 0.878 0.924 0.955 0.975 0.987 0.994
```

```
x <- 2
m <- 10000
u <- runif(m, 0, x)
g <- x * exp(- u^2/2) / sqrt(2 * pi)
v <- mean((g - mean(g))^2) / m
cdf <- mean(g)
c(cdf, v)
```

```
## [1] 4.775627e-01 5.364028e-06
```

```
c(cdf - 1.96 * sqrt(v), cdf + 1.96 * sqrt(v))
```

```
## [1] 0.4730232 0.4821021
```

Hence, the an estimate of the variance of Monte Carlo estimate of $\phi(2)$ is 5.26. A 95% confidence interval is [0.47, 0.48].

5.3

```
x <- 0.5
m <- 10000
u <- runif(m, 0, x)
g <- x * exp(-u)
cdf <- mean(g)
v <- mean((g - mean(g))^2) / m
c(cdf, v)
```

```
## [1] 3.935316e-01 3.154967e-07
```

```
z <- rexp(m)
z.g <- (z < x)
z.v <- mean((z.g - mean(z.g))^2) / m
z.cdf <- mean(z.g)
c(z.cdf, z.v)
```

```
## [1] 3.866000e-01 2.371404e-05
```

$\hat{\theta}$ is smaller. Because in the case of sampling from exponential distribution, there is a big probability of get very close to 0, which makes the computer harder to be precise.