Chapter 3

Methods for Generating Random Variables

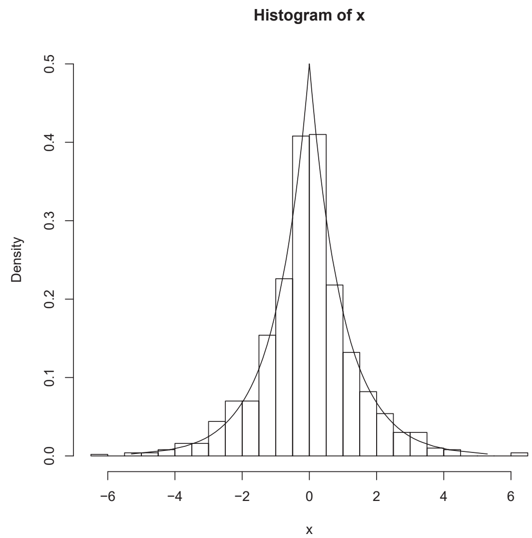
3.2 *The standard Laplace distribution has density* $f(x) = \frac{1}{2}e^{-|x|}, \quad , x \in R$. *Use the inverse transform method to generate a random sample of size 1000 from this distribution.*

Generate a random u from Uniform(0, 1). To compute the inverse transform, consider the cases $u < \frac{1}{2}$ and $u \geq \frac{1}{2}$ separately. If $u \geq \frac{1}{2}$ then $u = \int_{-\infty}^{x} f(t)dt = \frac{1}{2} + \frac{1}{2}(1 - e^{-x})$. If $u < \frac{1}{2}$ then $u = \int_{-\infty}^{x} f(t)dt = \frac{1}{2} - \frac{1}{2}(1 - e^{-x}) = \frac{1}{2}e^{-x}$. Deliver

$$x = F^{-1}(u) = \begin{cases} -\log(2u - 1), & \frac{1}{2} \leq u < 1; \\ \log(2u), & 0 < u < \frac{1}{2}. \end{cases}$$

```
n <- 1000
u <- runif(n)
i <- which(u >= 0.5)
x <- c(- log(2*u[i] - 1), log(2*u[-i]))

a <- c(0, qexp(ppoints(100), rate = 1))
b <- -rev(a)
```

```
hist(x, breaks="Scott", prob=TRUE, ylim=c(0,.5))
lines(a, .5 * exp(-a))
lines(b, .5 * exp(b))
```



3.3 *The Pareto(a, b) distribution has cdf*

$$F(x) = 1 - \left(\frac{b}{x}\right)^a, \qquad x \geq b > 0, a > 0.$$

Derive the probability inverse transformation $F^{-1}(U)$ *and use the inverse transform method to simulate a random sample from the Pareto(2, 2) distribution.*

The inverse transform is

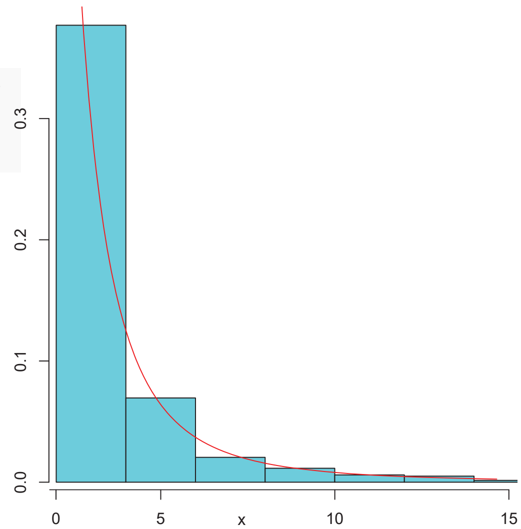
$$u = F(x) = 1 - (b/x)^a \Rightarrow x = b(1-u)^{-1/a},$$

and $U \sim$ Uniform(0,1) has the same distribution as $1 - U$.

```
a <- b <- 2
n <- 1000
u <- runif(n)
x <- b * u^(-1/a)
print(summary(x))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.001   2.278   2.787   4.059   3.949 109.944
```

The density of X is $f(x) = F'(x) = ab^a x^{-(a+1)}$, $x \geq b$.

```r
MASS::truehist(x, xlim=c(0, quantile(x, .98)))
fy <- function(y, a, b) {a*b^a * y^(-(a+1))}
curve(fy(x, a, b), add=TRUE, col=2)
```



3.7 *Generate a random sample of size 1000 from the Beta(3,2) distribution by acceptance-rejection method.*

Note that if $g(x)$ is the Uniform(0,1) density, then

$$\frac{f(x)}{g(x)} = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{x^{a-1}(1-x)^{b-1}}{1} \leq \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}, \qquad 0 < x < 1.$$

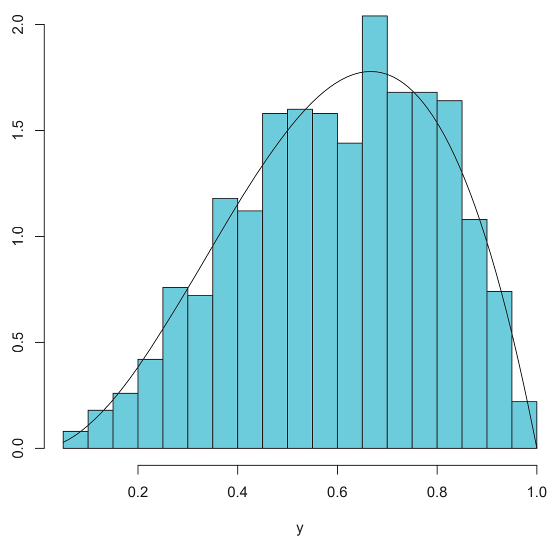
The R function below is a generator above for arbitrary parameters (a, b). It can be applied to generate the Beta(3, 2) sample.

Generate x from $g(x) \sim$ Uniform(0,1) and accept x if $x^{a-1}(1-x)^{b-1} > u$. This generator can be quite inefficient if a or b is large.

```r
rBETA <- function(n, a, b) {
    n <- 1000
    k <- 0       #counter for accepted
    y <- numeric(n)
    while (k < n) {
        u <- runif(1)
        x <- runif(1)  #random variate from g
        if (x^(a-1) * (1-x)^(b-1) > u) {
            #we accept x
            k <- k + 1
            y[k] <- x
        }
    }
    return(y)
}
```

The function is applied below to generate 1000 Beta(3, 2) variates and the histogram of the sample is shown with the Beta(3, 2) density superimposed.

```r
y <- rBETA(1000, a=3, b=2)
MASS::truehist(y, ylim=c(0, 2))
fz <- function(z) 12*z^2*(1-z)
curve(fz(x), add=TRUE)
```



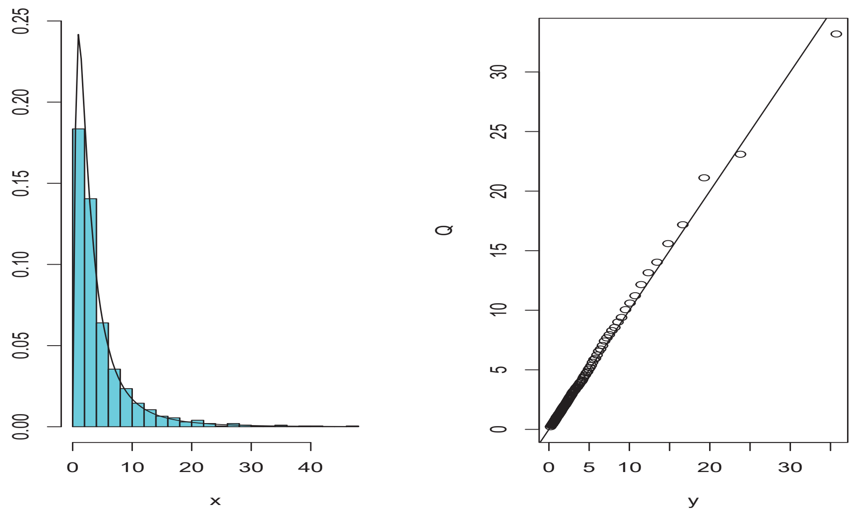
3.8 *Write a function to generate random variates from a Lognormal(μ, σ) distribution using a transformation method.*

If $X \sim$ Lognormal(μ, σ^2) then $X = e^Y$ where $Y \sim N(\mu, \sigma^2)$.

```r
rLOGN <- function(n, mu, sigma)
   return(exp(rnorm(n, mu, sigma)))

x <- rLOGN(1000, 1, 1)
print(summary(x))

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.09791  1.42400  2.81817  4.57061  5.49971 46.45014
```

The function `rLOGN` is applied to generate a sample of size 1000, and the histogram of the sample with the lognormal density curve superimposed is shown below. Another graphical comparison can be made with a QQ plot.

```r
par(mfrow = c(1, 2))
MASS::truehist(x, ylim=c(0, dlnorm(1,1,1)))
curve(dlnorm(x, 1, 1), add=TRUE)
y <- qlnorm(ppoints(100), 1, 1)
Q <- quantile(x, ppoints(100))
qqplot(y, Q)
abline(0, 1)
```

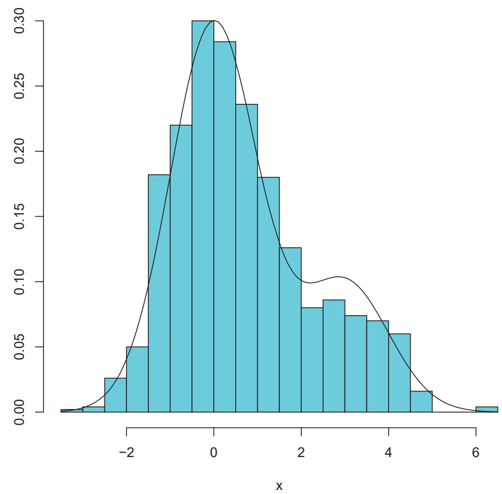


3.11 *Generate a random sample of size 1000 from a normal location mixture. The components of the mixture have $N(0,1)$ and $N(3,1)$ distributions with mixing probabilities p_1 and $p_2 = 1 - p_1$.*

```r
n <- 1000
p <- .75
mu <- sample(c(0, 3), size = 1000,
             replace = TRUE, prob = c(p, 1-p))
x <- rnorm(n, mu, 1)
```

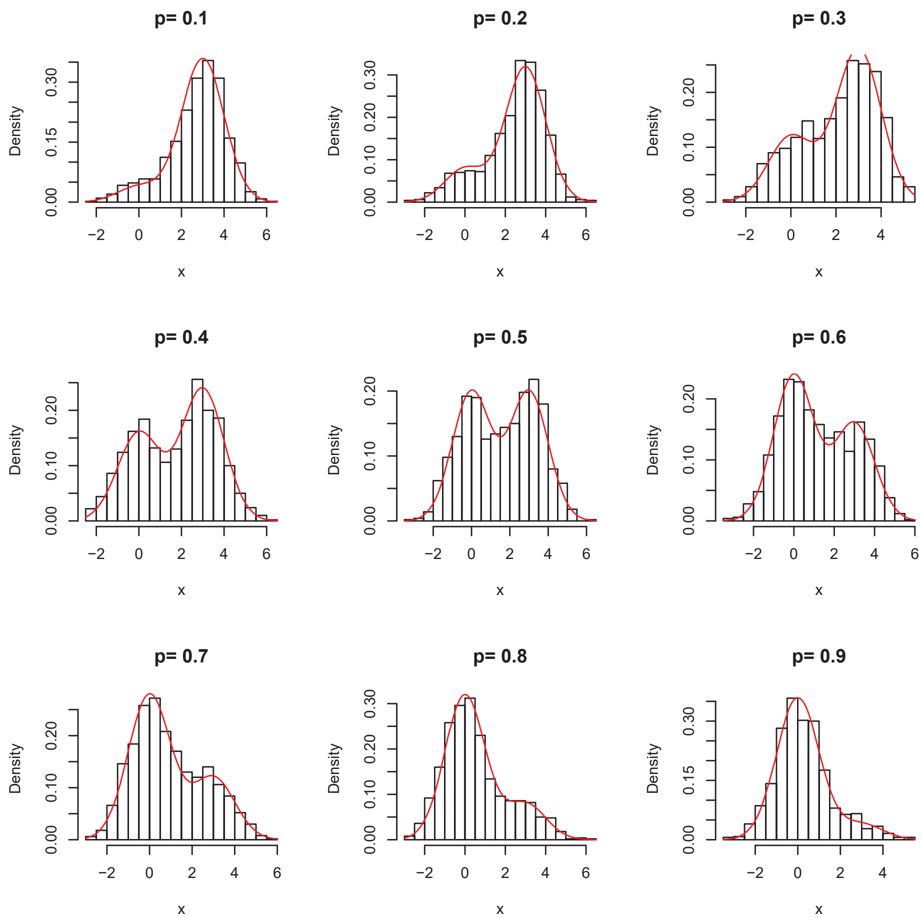
Below is the histogram of the sample with density superimposed, for $p_1 = 0.75$.

```r
MASS::truehist(x)
fy <- function(y, p) {
  p * dnorm(y) + (1 - p) * dnorm(y, 3, 1)
}
curve(fy(x, p), add=TRUE)
```



Repeating with different values for p_1:

```
par(mfrow = c(3, 3))
p <- seq(.1, .9, length = 9)
for (i in 1:9) {
    mu <- sample(c(0, 3), size = 1000,
                 replace = TRUE, prob = c(p[i], 1-p[i]))
    x <- rnorm(n, mu, 1)
    hist(x, breaks="Scott", prob=TRUE,
         main = paste("p=",p[i]))
    curve(fy(x, p[i]), add=TRUE, col=2)
}
```




```r
par(mfrow = c(1, 1))
```

From the graphs, we might conjecture that the mixture is bimodal if $0.2 < p < 0.8$. (Some results characterizing the shape of a normal mixture density are given by I. Eisenberger (1964), "Genesis of Bimodal Distributions," *Technometrics* **6**, 357–363.)

3.12 *Simulate a continuous Exponential-Gamma mixture. Suppose that the rate parameter* Λ *has Gamma*(r, β) *distribution and* Y *has Exp*(Λ) *distribution. That is,* $(Y|\Lambda = \lambda) \sim f_Y(y|\lambda) = \lambda e^{-\lambda y}$. *Generate 1000 random observations from this mixture with* $r = 4$ *and* $\beta = 2$.

Supply the sample of randomly generated λ as the Exponential rate argument in `rexp`.

```r
n <- 1000
r <- 4
beta <- 2
lambda <- rgamma(n, r, beta) #lambda is random
x <- rexp(n, rate = lambda)         #the mixture
```

3.13 The mixture in Exercise 3.12 has a Pareto distribution with cdf

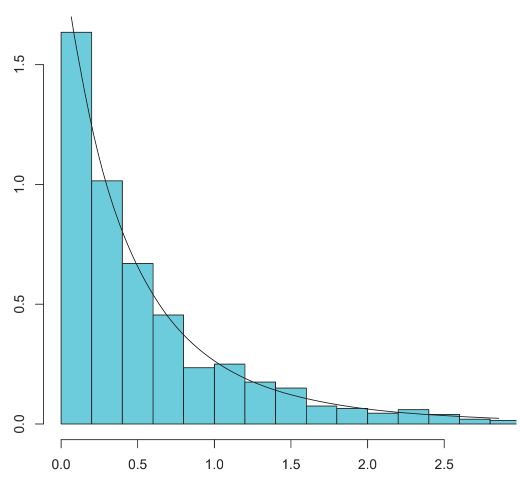
$$F(y) = 1 - \left(\frac{\beta}{\beta + y}\right)^r, \quad y \geq 0.$$

(This is an alternative parameterization of the Pareto cdf given in Exercise 3.) The Pareto density is

$$f(y) = F'(y) = \frac{r\beta^r}{(\beta + y)^{r+1}}, \qquad y \geq 0.$$

Below we generate 1000 random observations from the mixture with $r = 4$ and $\beta = 2$ and compare the empirical and theoretical (Pareto) distributions by graphing the density histogram of the sample and superimposing the Pareto density curve.

```r
MASS::truehist(x, xlim=c(0, quantile(x, .98)))
fy <- function(y, r) {
  r * beta^r * (beta + y)^(-r - 1)
}
curve(fy(x, r), add=TRUE)
```



3.14 *Generate 200 random observations from the 3-dimensional multivariate normal distribution having mean vector* $\mu = (0, 1, 2)$ *and covariance matrix*

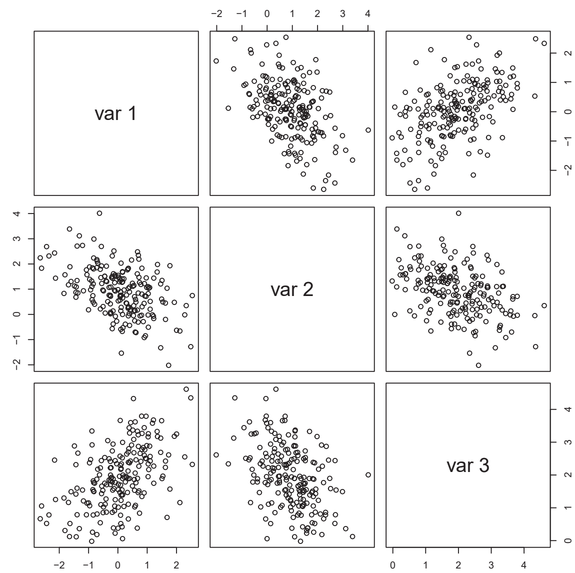
$$\Sigma = \begin{bmatrix} 1.0 & -0.5 & 0.5 \\ -0.5 & 1.0 & -0.5 \\ 0.5 & -0.5 & 1.0 \end{bmatrix}.$$

using the Choleski factorization method.

```
    rmvn.Choleski <-
    function(n, mu, Sigma) {
        # generate n random vectors from MVN(mu, Sigma)
        # dimension is inferred from mu and Sigma
        d <- length(mu)
        Q <- chol(Sigma) # Choleski factorization of Sigma
        Z <- matrix(rnorm(n*d), nrow=n, ncol=d)
        X <- Z %*% Q + matrix(mu, n, d, byrow=TRUE)
        X
    }

    Sigma <- matrix(c(1, -.5, .5, -.5, 1,
                      -.5, .5, -.5, 1), 3, 3)
    mu <- c(0, 1, 2)
    x <- rmvn.Choleski(200, mu, Sigma)
    colMeans(x)

## [1] 0.07480759 0.91773863 1.95379227

    cor(x)

##            [,1]       [,2]       [,3]
## [1,]  1.0000000 -0.4624160  0.4999933
## [2,] -0.4624160  1.0000000 -0.4186511
## [3,]  0.4999933 -0.4186511  1.0000000
```

From the **pairs** plot below it appears that the centers of the distributions agree with the parameters in μ, and the correlations also agree approximately with the parameters in Σ.

```
  pairs(x)
```


var 1
var 2
var 3