Homework 1

Yiming Gao (NetID: yimingg2)

2017/1/22

Exercise 1

(a) The density function is $f(x) = \lambda X^{\lambda-1}$, we have

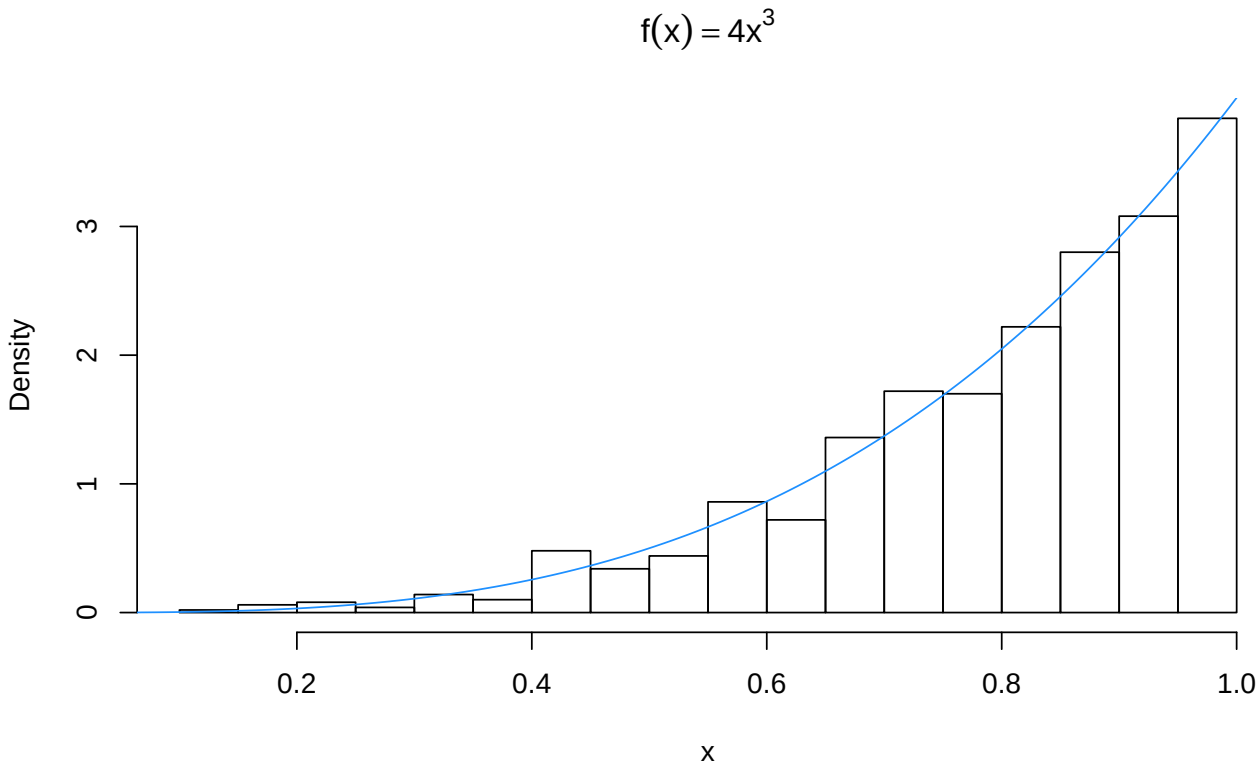
$$F(x) = \int_0^x \lambda t^{\lambda-1} dt$$

$$x = F^{-1}(u) = u^{1/\lambda}$$

```
sample1 <- function(n, lambda){
  Usample = runif(n, 0, 1)
  Xsample = Usample^(1/lambda)
  return(Xsample)
}
```

(b) Here we draw a sample with sample size $n = 1000$ and $\lambda = 4$,and plot the empirical c.d.f (with seed 1234).

```
set.seed(1234)
x = sample1(1000, 4)
hist(x, prob = TRUE, nclass = 25, main = expression(f(x)==4*x^3))
y = seq(0, 1, 0.001)
lines(y, 4*y^3, col = "dodgerblue")
```



The histogram and density plot suggests that the empirical and theoretical distributions approximately agree.

Exercise 2

$Z_{(k)}$ is the kth order statistic in a sample of size n from a $N(\mu, \sigma^2)$ distribution. We first generate a random matrix of m*n, then we sort the rows ascendingly.

(a)

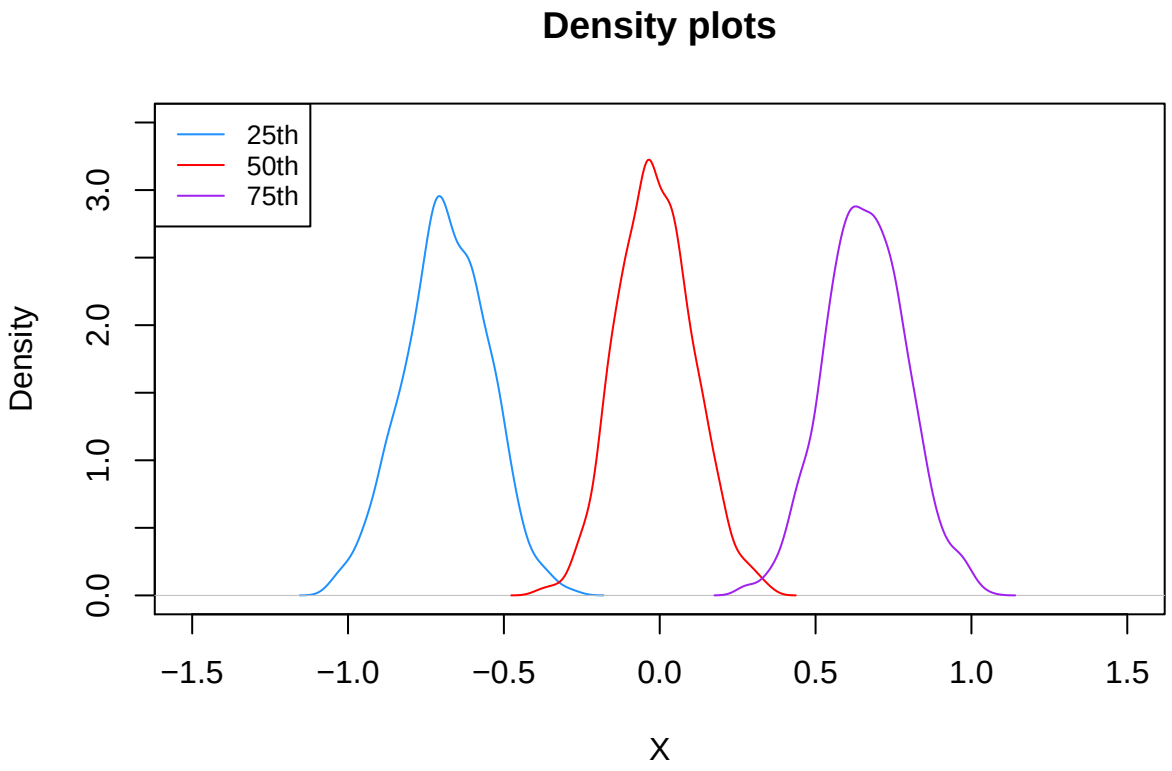
```
sample2 <- function(m, n, k, mu, var){
  # generate random matrix
  Nmatrix = matrix(rnorm(m*n, mean = mu, sd = sqrt(var)), m, n)

  # sort by row
  Nsort = apply(Nmatrix, 1, sort)

  # draw the sample
  Xsample = Nsort[k,]
  return(Xsample)
}
```

(b) For $n = 100$, we draw 1000 samples of 25th, 50th and 75th order statistics. Here we set $\mu = 0, \sigma^2 = 1$ and 1234 as seed.

```
set.seed(1234)
x.25 = sample2(1000, 100, 25, 0, 1)
x.50 = sample2(1000, 100, 50, 0, 1)
x.75 = sample2(1000, 100, 75, 0, 1)
plot(density(x.25), col = "dodgerblue", xlim=c(-1.5, 1.5), ylim = c(0,3.5),
     main = "Density plots", xlab = "X")
lines(density(x.50), col = "red")
lines(density(x.75), col = "purple")
legend("topleft",legend = c("25th", "50th", "75th"), col = c("dodgerblue", "red", "purple"),
      lty = rep(1,3), cex = 0.8)
```



Exercise 3

(a)

X have probability mass function $f(x) = p(1-p)^{x-1}$, for $0 < p < 1$,and $x = 1, 2, 3...$ From which we can tell that X have geometric distribution. Here we use the inverse transform method to generate a random geometric sample with parameter p.

The cdf is

$$F = 1 - (1-p)^{x+1}$$

For each sample element we need to generate a random uniform u and solve

$$1 - (1-p)^{x} < u \leq 1 - (1-p)^{x+1}$$

This inequality simplifies to $x < log(1-u)/log(1-p) \leq x+1$. Thus we have R codes as follows:

```
sample3 <- function(n, p){
  u = runif(n)
  Xsample = floor(log(u)/ log(1-p))
  return(Xsample)
}
```

(b)

Y represent the number of Bernoulli trials required to observe the *kth* success. X (Geometric distribution) represents the number of trials until the first success. Actually Y has a negative binomial distribution with parameter p and n, it is the sum of independent geometrically distributed variables $X_1, X_2, ...$

```
sample4 <- function(n, p, k){
  # Get n*k matrix of random uniforms
  Umatrix = matrix(runif(n*k), n, k)

  # Draw geometic variables(from (a))
  Xmatrix = floor(log(Umatrix)/ log(1-p))

  # Sum over rows to get Y
  Ysample = apply(Xmatrix, 1, sum)
  return(Ysample)
}
```

We can investigate how sample matches with theorectical quantiles. We can know that the empirical and theoretical distributions approximately agree.

```
set.seed(1234)
Ysample = sample4(10, 0.5, 15)
p = seq(0.1, 0.9, 0.1)
Qhat = quantile(Ysample, p)
Q = qnbinom(p, 15, 0.5)
round(rbind(Qhat, Q), 2)
```

```
##        10% 20%  30% 40% 50%  60% 70%  80%  90%
## Qhat 10.2  11 15.2  17  17 17.8  19 19.2 20.4
## Q     8.0  10 12.0  13  14 16.0  17 19.0 22.0
```

Exercise 4

(a)

Since $Y \sim exp(1)$ and has marginal p.d.f. $f(y) = e^{-y}$ for $y > 0$. We could first draw samples from Y, then condition on Y, draw from the corresponding Normal distribution of X. The function takes n (sample size) as its argument.

```
sample5 <- function(n){
  # Y is random
  Ysample = rexp(n, rate = 1)

  # now apply the Ysample as the normal mean
  Xsample = rnorm(n, mean = Ysample, sd = 1)
  return(Xsample)
}
```

(b)

We write a function with sample size n as its argument. It will return a vector of estimated mean and variance of the population. Since we have

$$E(\bar{X}) = \theta, \quad E(S^2) = \frac{\sigma^2}{n}$$

where S^2 is the sample variance.

```
estimate <- function(n){
  X.sample = sample5(n)
  X.mean = mean(X.sample)
  X.var = var(X.sample)
  return(round(c(X.mean, X.var), digits = 2))
}
```

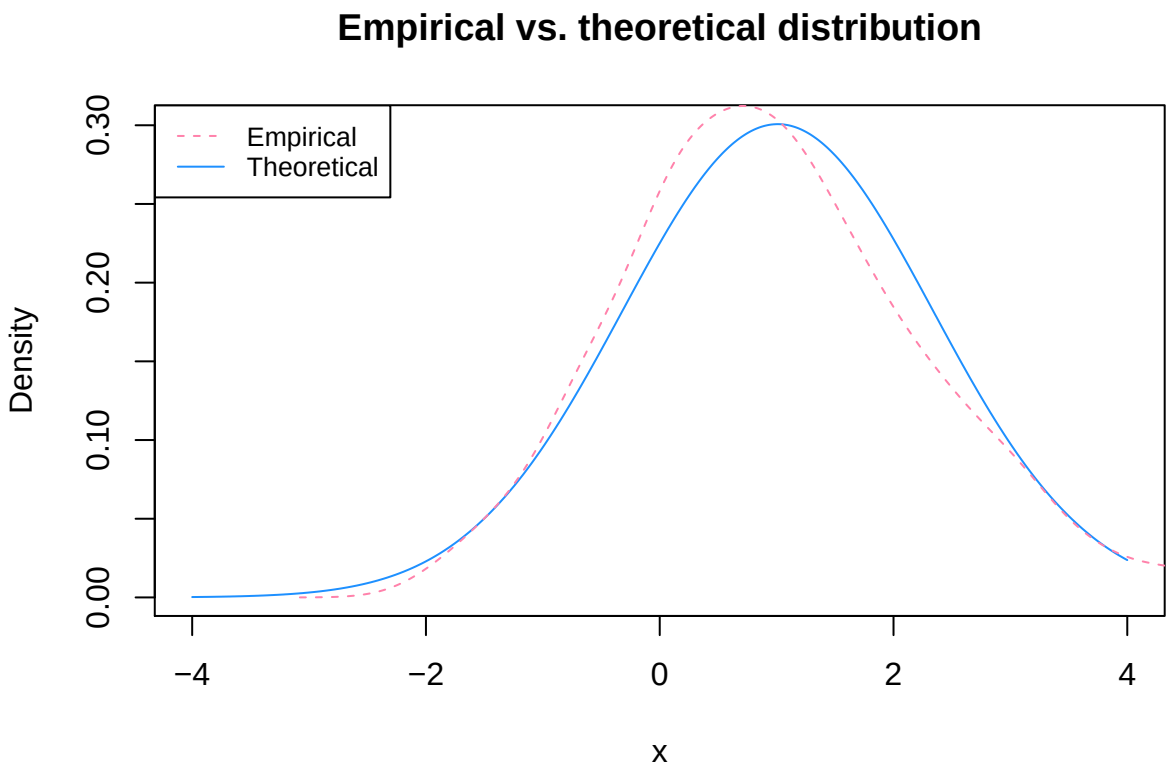
Then we try the function above, and plot the estimated p.d.f. of X alongside the density of a $N(\mu, \sigma^2)$ distribution. We will draw 1000 samples.

```
set.seed(1234)
x1 = sample5(1000)
```

```r
x2 = seq(-4, 4, length = 1000)
y2 = dnorm(x2, mean = estimate(1000)[1], sd = sqrt(estimate(1000)[2]))

plot(x2, y2, xlab = "x", ylab = "Density",
     main = "Empirical vs. theoretical distribution",
     type = 'l', col = 'dodgerblue')

lines(density(x1), col = "palevioletred1", lty = 2)
legend("topleft",legend = c("Empirical", "Theoretical"),
       col = c("palevioletred1", "dodgerblue"),
       lty = c(2, 1), cex = 0.8)
```



From the plot above, we can observe that the empirical (pink) and theoretical (blue) distributions approximately agree, but they are not completely overlapped. Because X is conditionally normal distributed.

Exercise 5 (Rizzo 3.3)

We have Pareto(a,b) distribution, which has c.d.f.

$$F(x) = 1 - (\frac{b}{x})^a, \quad x \leq b \leq 0, \quad a > 0$$

We are asked to derice the probability inverse transformation $F^{-1}(U)$ and use inverse CDF to draw samples. Meanwhile, plot the density histogram together with Pareto(2,2). We know that Pareto(2,2) has p.d.f.

$$f(x) = \frac{8}{x^3}$$

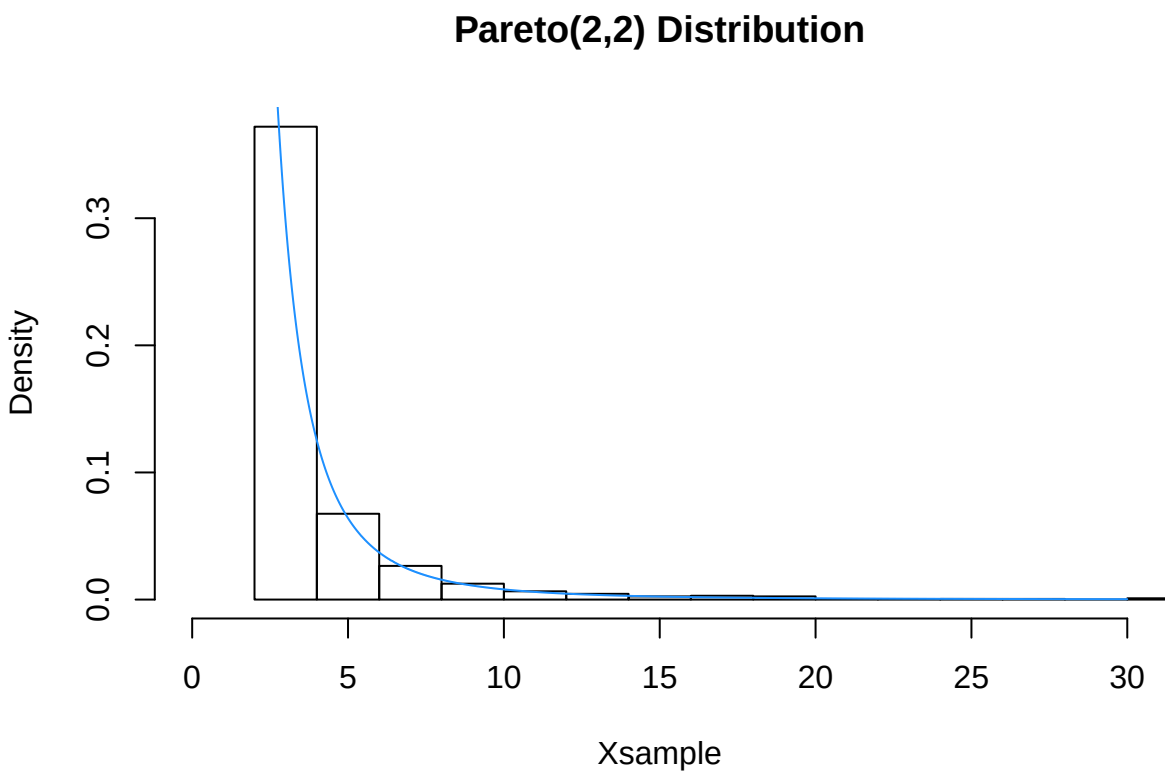
and

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

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

Then we draw 1000 samples from Pareto(2,2) where $a = b = 2$ and plot them.

```
Usample = runif(1000, 0, 1)
Xsample = 2/(sqrt(1-Usample))
hist(Xsample, prob = TRUE, breaks = 25, main = "Pareto(2,2) Distribution",xlim = c(0,30))
y = seq(0, 30, 0.001)
lines(y, 8/(y^3), col = "dodgerblue")
```



Exercise 6 (Rizzo 3.4)

The Rayleigh density is

$$f(x) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}, \quad x \leq 0, \quad \sigma > 0$$

Thus we can calculate its c.d.f.

$$F(x) = 1 - e^{-x^2/2\sigma^2}$$

and we have

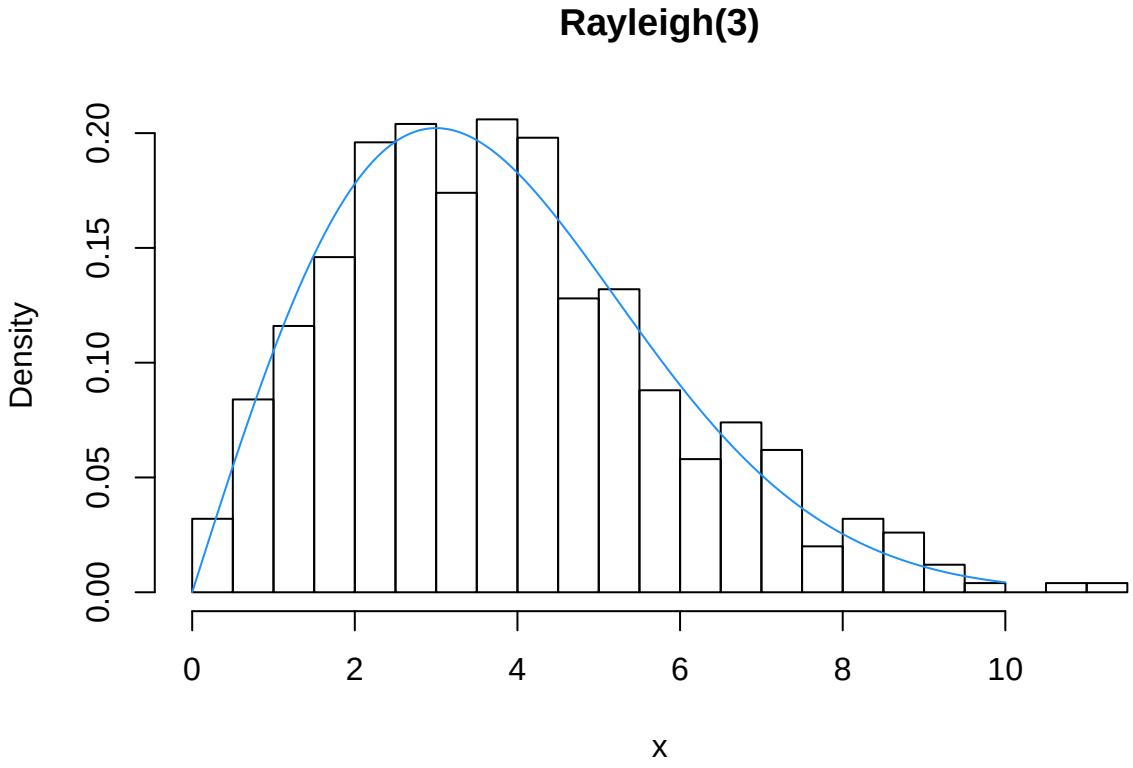
$$x = F^{-1}(u) = \sqrt{-2\sigma^2 log(1-u)}$$

(1) We will write a function with n (sample size) and σ as its arguments.

```
sample6 <- function(n, sigma){
  Usample = runif(n, 0, 1)
  Xsample = sqrt(-2*sigma^2*log(1-Usample))
  return(Xsample)
}
```

(2) Here we draw a sample with sample size $n = 1000$ and $\sigma = 3$, and plot the empirical c.d.f (with seed 1234).

```
set.seed(1234)
x = sample6(1000, 3)
hist(x, prob = TRUE, nclass = 25, main = "Rayleigh(3)")
y = seq(0, 10, 0.001)
lines(y, (y/9)*exp(-y^2/18), col = "dodgerblue")
```



The histogram and density plot suggests that the empirical and theoretical distributions approximately agree.

Exercise 7 (Rizzo 3.5)

(1) Inverse CDF approach

```
set.seed(1234)
# CDF
Fx = cumsum(c(0.1, 0.2, 0.2, 0.2, 0.3))
# Generate uniform samples
Usample = runif(1000, 0, 1)
Xsample = rep(1, 1000)
for (i in 1:5){
  Xsample = Xsample + (Usample > Fx[i])
}

# Create the frenquency table
rbind(table((Xsample))/1000, c(0.1,0.2,0.2,0.2,0.3))
```

```
##          1     2     3     4     5
```

```
## [1,] 0.092 0.199 0.191 0.216 0.302
## [2,] 0.100 0.200 0.200 0.200 0.300
```

We can see that the sample distribution is close to the theoretical distribution.

(2) R sample function

```
set.seed(1234)
probs = c(0.1, 0.2, 0.2, 0.2, 0.3)
Xsample = sample(0: 4, 1000, replace = TRUE, probs)
# Create the frenquency table
rbind(table((Xsample))/1000, c(0.1,0.2,0.2,0.2,0.3))
```

```
##         0     1     2     3     4
## [1,] 0.12 0.182 0.191 0.216 0.291
## [2,] 0.10 0.200 0.200 0.200 0.300
```

Exercise 8 (Rizzo 3.9)

We are asked to generate random variates from f_e, where

$$f_e(x) = \frac{3}{4}(1 - x^2), \quad |x| \leq 1$$

with the rules: if $|U_3| \geq |U_2|$ and $|U_3| \geq |U_1|$, deliver $|U_2|$; otherwise deliver $|U_3|$.

(a) Construct function `sample7`

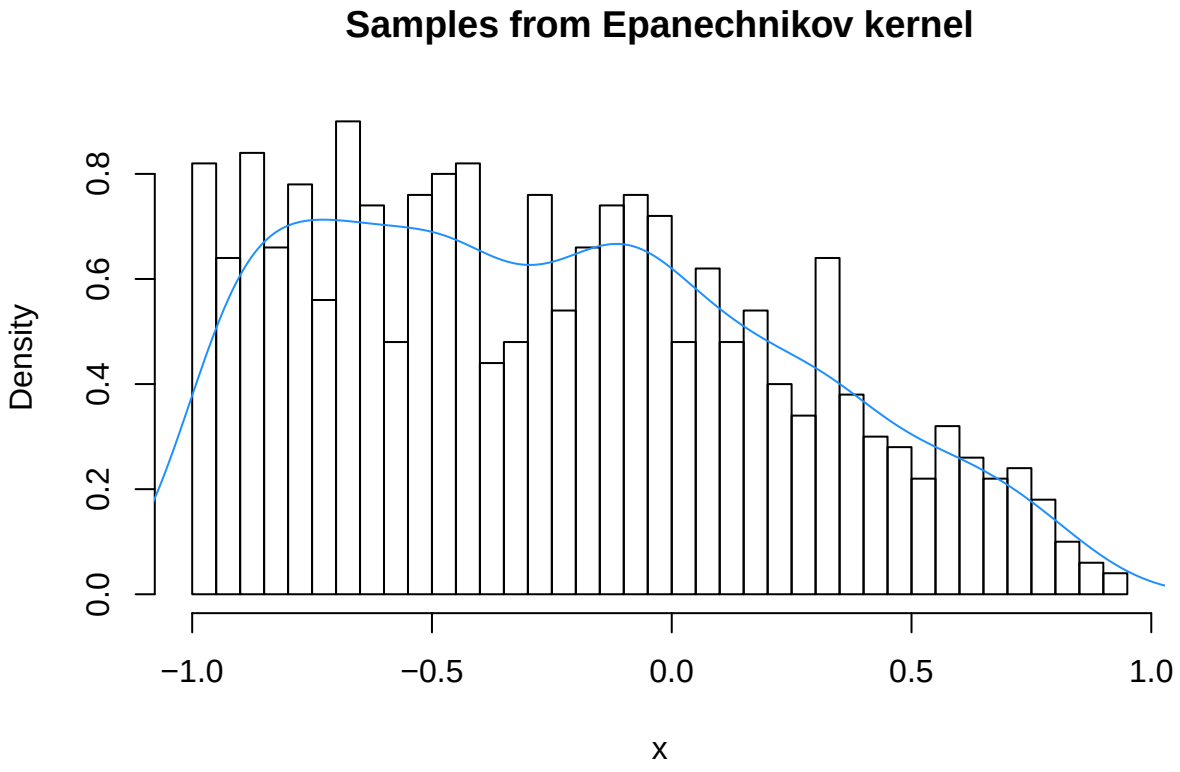
We first write a function which follows the given rule.

```
sample7 <- function(n){
  Umatrix = matrix(runif(n*3, -1, 1), n, 3)
  Xsample = ifelse(((Umatrix[,3]>=Umatrix[,2])&(Umatrix[,3]>=Umatrix[,1])),
                   Umatrix[,2], Umatrix[,3])
  return(Xsample)
}
```

(b) Draw samples and plot the density

We plot the histogram density of a large simulated random sample.

```
set.seed(1234)
x = sample7(1000)
hist(x, prob = TRUE, nclass = 30, main = "Samples from Epanechnikov kernel")
lines(density(x), col = 'dodgerblue')
```



Bonus

Since $f(x,y) = 60x^2y$ for $0 < x < 1, 0 < y < 1, x + y < 1$, let $g(x,y) = 1$ for $0 < x, y < 1$, which has square support.

Let

$$c = max_{s,t\in(0,1)}\frac{f(s,t)}{g(s,t)} = max_{s,t\in(0,1)}60s^2t = max_{s\in(0,1)}60s^2(1-s) = \frac{80}{9},$$

where $s = \frac{2}{3}$.

We should accept $\frac{9}{80}$ of the time, so to get a sample of about 100 from the distribution $f(x,y) = 60x^2y$, we have to draw about 889 from $g(x,y) = 1$.

```
set.seed(1234)
X0 = runif(889)
Y0 = runif(889)
Usample = runif(889)
ratio = 60*X0^2*Y0/(80/9)
X = X0[Usample < ratio]
Y = Y0[Usample < ratio]
```

We first print out the first several observations to have a look.

```r
# the first several samples
head(cbind(X,Y))
```

```
##              X         Y
## [1,] 0.6222994 0.9203998
## [2,] 0.6092747 0.7464459
## [3,] 0.6233794 0.6054089
## [4,] 0.8609154 0.3776961
## [5,] 0.6403106 0.3414909
## [6,] 0.6660838 0.6329661
```

Then we plot the empirical density as well as theoretical distribution line.