Homework 3

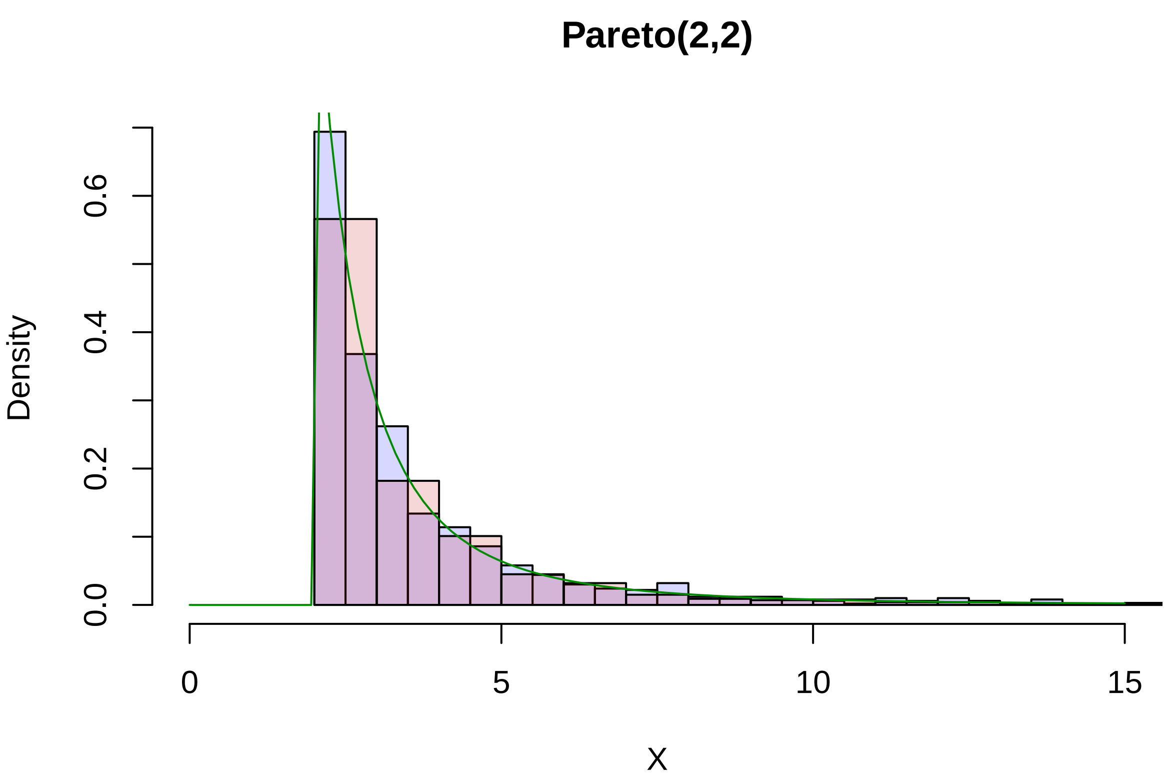
3.3

The $Pareto(a, b)$ distribution has cdf $F(x) = 1 - (\frac{b}{x})^a, \quad x \geq b > 0,\ a > 0$. Derive the probability inverse transformation $F^{-1}(U)$ and use the inverse transformation method to simulate a random sample from the Pareto(2,2) distribution. Graph the density histogram of the sample with the Pareto(2,2) density superimposed for comparison.

Ans.

$$F^{-1}(U) = b(1-u)^{-1/a}$$

```
set.seed(165)
n <- 1000
u <- runif(n) #Uniform Sample
a <- 2
b <- 2
X <- b*(1-u)^(-1/a) #Pareto Sample Using Inverse Transformation
p <- rpareto(n,2,2)
hist(X, freq=F, col=mycol1,breaks=200, xlim=c(0,15), main="Pareto(2,2)") #Histogram of Inverse Transfor
hist(p, freq=F, col=mycol2,breaks=50, add=T, xlim=c(0,15)) #Histogram of Random Pareto Sample
curve(dpareto(x,2,2), add = T, col='green4', xlim=c(0,15)) #Density curve for Pareto(2,2)
```



3.4

The Rayleigh density is $f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}$, $x \geq 0$, $\sigma > 0$. Develop an algorithm to generate random samples from a Rayleigh(σ) samples for several choices of $\sigma > 0$ and check that the mode of the generated samples is close to the theoretical mode σ (check the histogram).

Ans.

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

```r
raysample <- function(s, u){
  r <- s*sqrt(-2*log(u))
  return (r)
  }

par(mfrow=c(1,3))

hist(raysample(1, u), freq=F, col='blue4', main="Rayleigh(1)")
curve(drayleigh(x,1), add=T, col='red4', lw=2)
abline(v=1, col='green4')

hist(raysample(2, u), freq=F, col='blue4', main="Rayleigh(2)")
curve(drayleigh(x,2), add=T, col='red4', lw=2)
abline(v=2, col='green4')

hist(raysample(3, u), freq=F, col='blue4', main="Rayleigh(3)")
curve(drayleigh(x,3), add=T, col='red4', lw=2)
abline(v=3, col='green4')
```

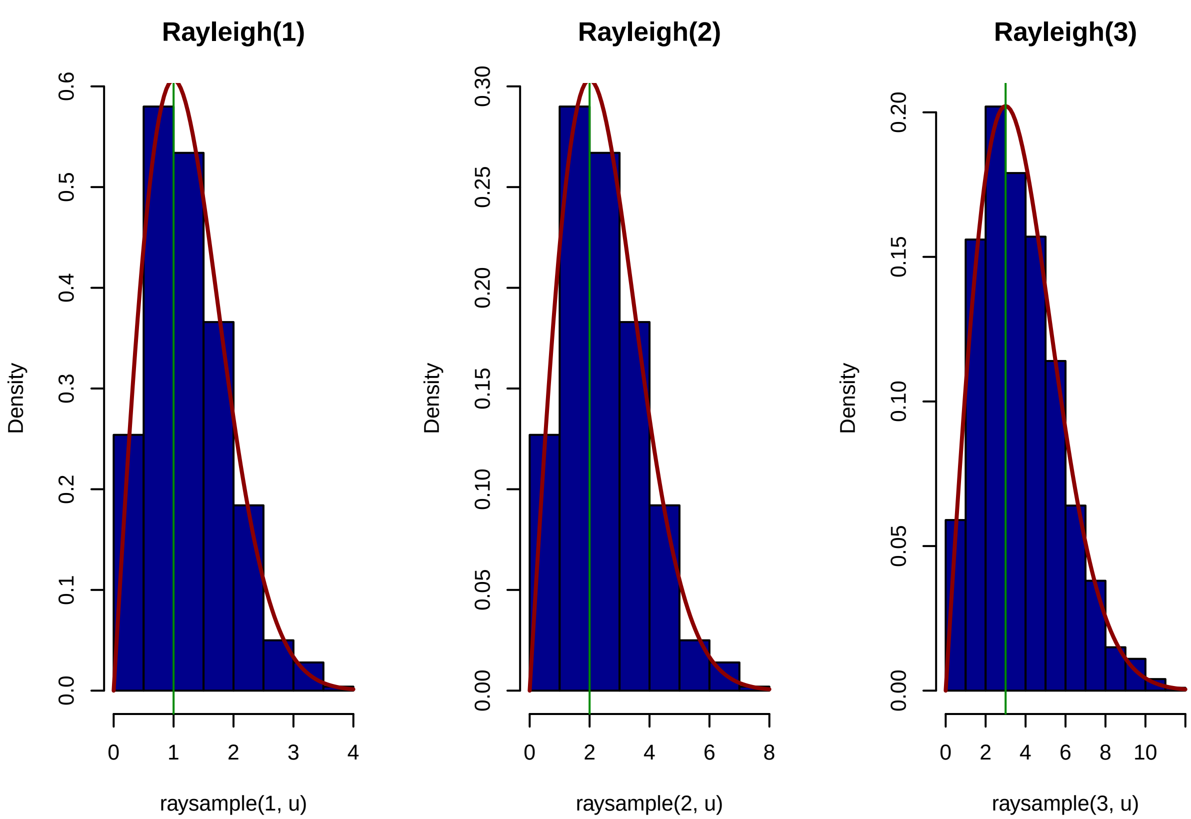


Table of the modes from the Rayleigh samples (rounded to one decimal) given different σ values:

σ	Mode	σ	Mode
1	0.9	5	4.7
2	1.9	6	5.7
3	2.8	7	6.6
4	3.8	8	7.6

3.5

A discrete random variable X has probability mass function

x	0	1	2	3	4
p(x)	0.1	0.2	0.2	0.2	0.3

Use the inverse transform method to generate a random sample of size 1,000 from the distribution of X. Construct a relative frequency table and compare the empirical with the theoretical probabilities. Repeat using the *R sample* function.

```r
set.seed(165)

u2 <- runif(1000)

X2 <- numeric(1000)

for (i in 1:1000){

  if (u2[i] <= 0.1){
    X2[i] <- 0
  } else {
    if (u2[i] <= 0.3 && u2[i] >0.1){
      X2[i] <- 1
    } else{
      if( u2[i] <= 0.5 && u2[i] >0.3){
      X2[i] <- 2
      } else{
        if( u2[i] <= 0.7 && u2[i] >0.5){
        X2[i] <- 3
        } else{
          if( u2[i] <= 1 && u2[i] >0.7){
          X2[i] <- 4
          }
        }
      }
    }
  }

}
```

Comparative of Empirical Probabilities from the sample X and the Theoretical Probabilities:

```r
df <- as.data.frame(round(table(X2)/1000,2))
df <- cbind(df, c(0.1,0.2,0.2,0.2,0.3))
colnames(df) <- c("x", "Empirical Prob.", "Theoretical Prob.")
df
```

```
##   x Empirical Prob. Theoretical Prob.
## 1 0            0.10               0.1
## 2 1            0.20               0.2
## 3 2            0.18               0.2
## 4 3            0.20               0.2
## 5 4            0.32               0.3
```

Sample from X #1 (size = 100)

```r
set.seed(165)
X2_1 <- sample(X2, 100)

df1 <- as.data.frame(round(table(X2_1)/100,2))
df1 <- cbind(df1, c(0.1,0.2,0.2,0.2,0.3))
colnames(df1) <- c("x", "Empirical Prob.", "Theoretical Prob.")
df1
```

```
##   x Empirical Prob. Theoretical Prob.
## 1 0            0.08               0.1
## 2 1            0.21               0.2
## 3 2            0.21               0.2
## 4 3            0.18               0.2
## 5 4            0.32               0.3
```

Sample from X #2 (size = 250)

```r
set.seed(165)
X2_2 <- sample(X2, 250)

df2 <- as.data.frame(round(table(X2_2)/250,2))
df2 <- cbind(df2, c(0.1,0.2,0.2,0.2,0.3))
colnames(df2) <- c("x", "Empirical Prob.", "Theoretical Prob.")
df2
```

```
##   x Empirical Prob. Theoretical Prob.
## 1 0            0.07               0.1
## 2 1            0.20               0.2
## 3 2            0.18               0.2
## 4 3            0.22               0.2
## 5 4            0.33               0.3
```

Sample from X #3 (size = 500)

```r
set.seed(165)
X2_2 <- sample(X2, 500)
```

```r
df2 <- as.data.frame(round(table(X2_2)/500,2))
df2 <- cbind(df2, c(0.1,0.2,0.2,0.2,0.3))
colnames(df2) <- c("x", "Empirical Prob.", "Theoretical Prob.")
df2
```

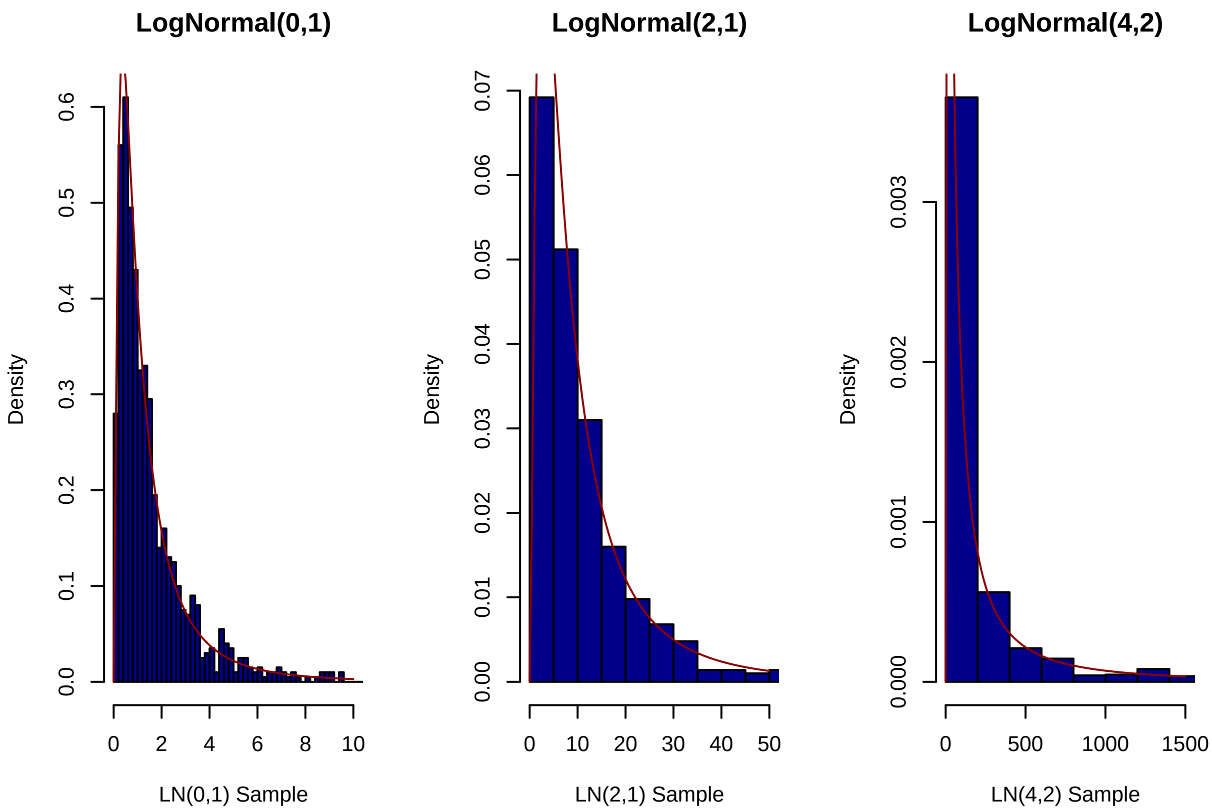
```
##   x Empirical Prob. Theoretical Prob.
## 1 0            0.09               0.1
## 2 1            0.21               0.2
## 3 2            0.18               0.2
## 4 3            0.20               0.2
## 5 4            0.31               0.3
```

3.8

Write a function to generate random variates from a Lognormal(μ, σ) distribution using a transformation method, and generate a random sample of size 1,000. Compare the histogram with the Lognormal density curve given by the *dlnorm* function in *R*.

Ans. Given $X \sim N(\mu, \sigma)$ then $log(X) \sim LN(\mu, \sigma)$. Using this transformation from the *rnorm* function in *R*.

```r
set.seed(165)

normsample <- function(mu, sigma, n){
  return (rnorm(n,mu,sigma))
}


par(mfrow=c(1,3))

hist(exp(normsample(0,1,n)), col='blue4', freq=F, breaks=100, xlim=c(0,10), xlab = "LN(0,1) Sample", ma
curve(dlnorm(x,0,1), col='red4', add=T)

hist(exp(normsample(2,1,n)), col='blue4', freq=F, breaks=100, xlim=c(0,50), xlab = "LN(2,1) Sample", ma
curve(dlnorm(x,2,1), col='red4', add=T)

hist(exp(normsample(4,2,n)), col='blue4', freq=F, breaks=250, xlim=c(0,1500), xlab = "LN(4,2) Sample", 
curve(dlnorm(x,4,2), col='red4', add=T)
```



3.9

The rescaled Epanechnikov kernel [92] is a symmetric density function

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

Devroye and Györfi give the following algorithm for simulation for this distribution. Generate *iid* $U_1, U_2, U_3 \sim Uniform(-1, 1)$. If $|U_3| \geq |U_2|$ and $|U_3| \geq |U_1|$, deliver U_2; otherwise deliver U_3. Write a function to generate random variates from f_e, and construct the histogram density estimate of a large simulated random sample.

```
set.seed(555)
n2 <- 10000
U1 <- runif(n2,-1,1)
U2 <- runif(n2,-1,1)
U3 <- runif(n2,-1,1)
Uepa <- numeric(n2)

epa <- function(U1,U2,U3, n){
  Uepa <- numeric(n)
  for (i in 1:n) {
    if (abs(U3[i]) >= abs(U2[i]) && abs(U3[i]) >= abs(U1[i])) {
      Uepa[i] <- U2[i]
      } else {
        Uepa[i] <- U3[i]
      }
  }
```

```
  return (Uepa)
}

hist(epa(U1,U2,U3, n2), col='blue4', freq=F, xlim=c(-1,1), main='Epanechnikov Distribution', xlab='Epane
curve((3/4)*(1-x^2), add=T, col='red4', lw=2)
```