Homework 4

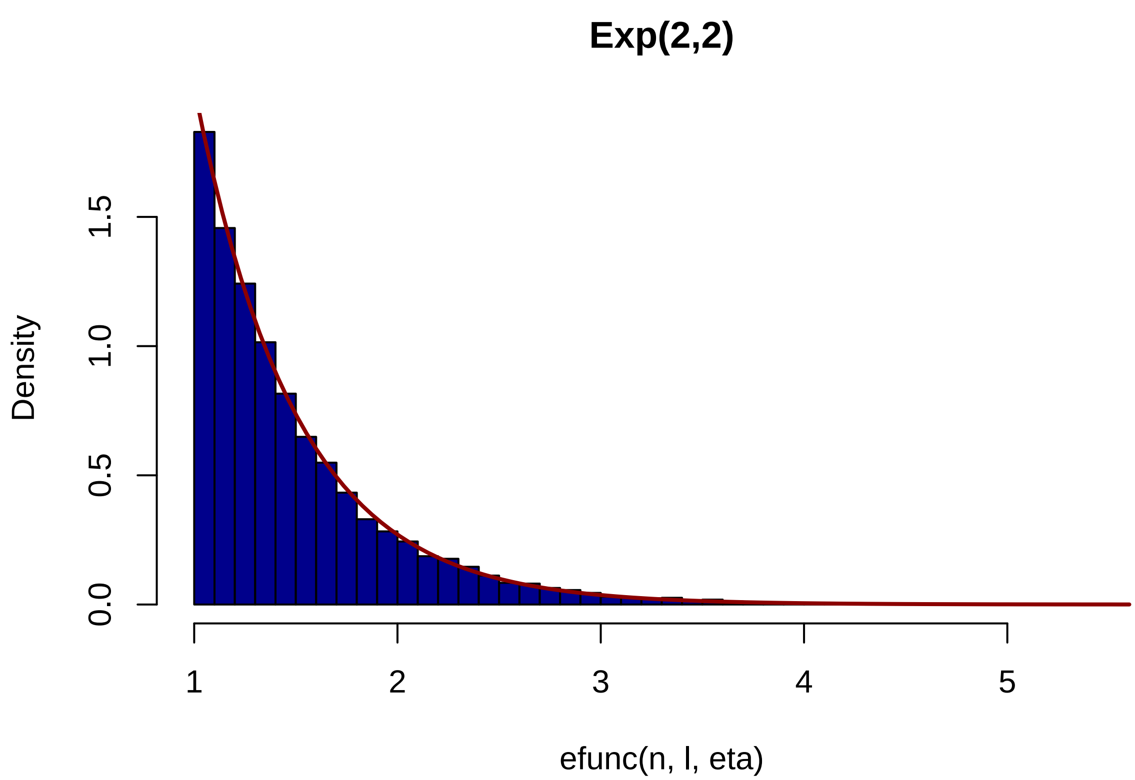
10/01/2021

3.1.

Write a function that will generate and return a random sample of size n from the two-parameter exponential distribution $Exp(\lambda, \eta)$ for arbitrary n, λ, and η. (See Examples 2.3 and 2.6.) Generate a large sample from $Exp(\lambda, \eta)$ and compare the sample quantiles with the theoretical quantiles.

$$F(x) = 1 - e^{-\lambda(x-\eta)} => F^{-1}(x) = \frac{log(1-u)}{-\lambda} + \eta$$

```
efunc <- function(n, lambda, eta){
  efunc <- numeric(n)
  u <- runif(n)
  efunc <- (log(1-u)/(-lambda))+eta
  return (efunc)
}
n <- 10000; l <-2; eta <- 1
hist(efunc(n, l,eta), col='blue4', freq=F, breaks=50, main='Exp(2,2)')
curve(l*exp(-l*(x-eta)), add=T, col='red4', lw=2)
```

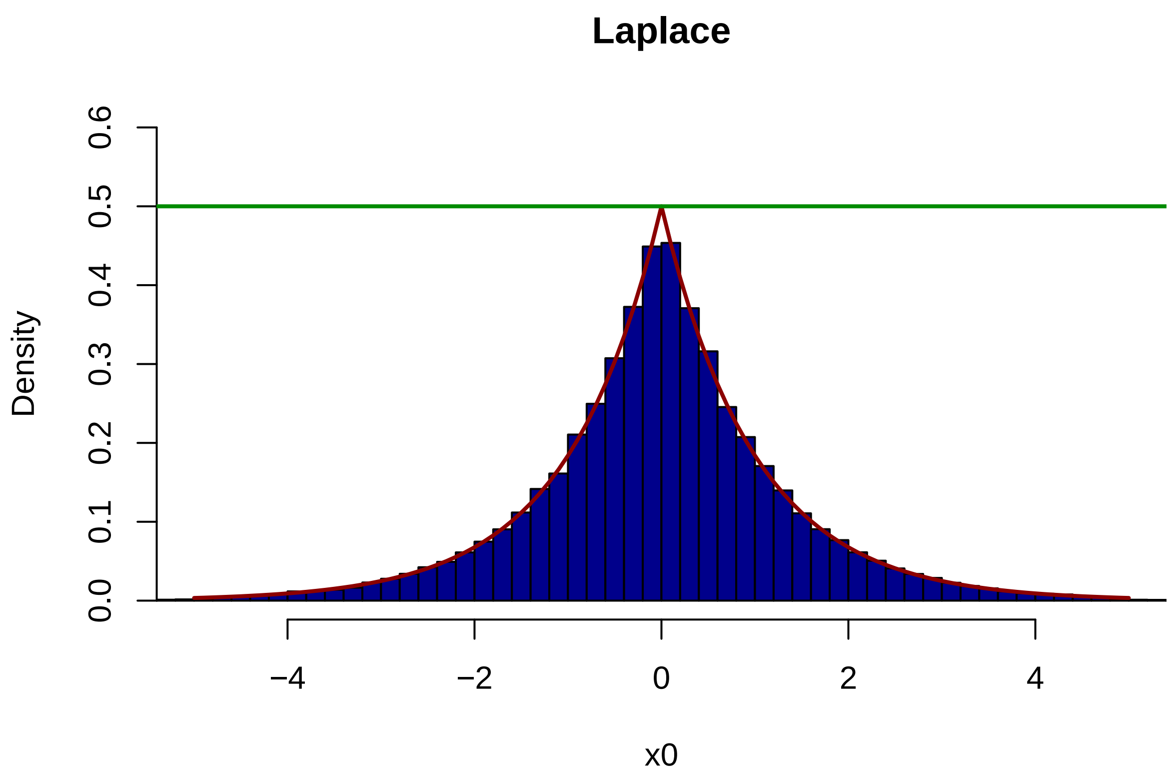


3.2.

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

```
laplace <- function(n, a, b, c){
k <- 0 #indicator for accepted value
x <- integer(n) #vector to store the accepted value

# define the function for f(x)/cg(x)
ratio0 <- function(x, a, b, c){
  f <- (1/2)*exp(-abs(x))
  g <- dnorm(x,0, 1.5)
  f/(c*g)
}

while (k < n) {
  y <- rnorm(1,0, 1.5)
  u <- runif(1)
  if (u < ratio0(y, a, b, c)){
    x[k] <- y
    k <- k+1
  }
}
x
}
```

```
n0 <- 100000
a <- -5
b <- 5
c <- 2

x0 <- laplace(n0, a, b, c)
hist(x0, freq=F, col='blue4', main='Laplace', breaks=50, xlim=c(a,b), ylim=c(0,0.6))
curve((1/2)*exp(-abs(x)), add=T, col='red4', lw=2)
abline(h=0.5, col='green4', lw=2)
```



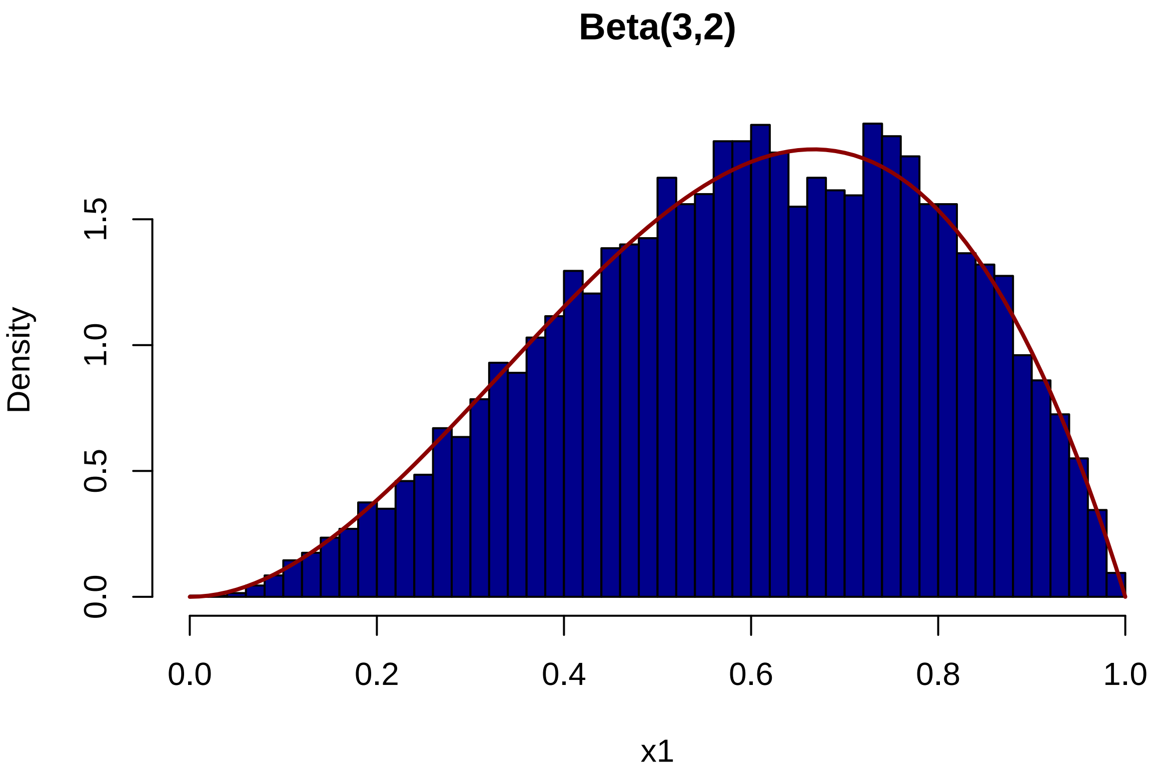
3.7.

Write a function to generate a random sample of size n from the $Beta(a, b)$ distribution by the acceptance-rejection method. Generate a random sample of size 1000 from the $Beta(3, 2)$ distribution. Graph the histogram of the sample with the theoretical $Beta(3, 2)$ density superimposed.

```
beta <- function(n, a, b){
k <- 0 #indicator for accepted value
x <- integer(n) #vector to store the accepted value

# define the function for f(x)/cg(x)

ratio <- function(x, a, b){
  f <- (factorial(a+b)/(factorial(a)*factorial(b)))*(x^(a-1))*((1-x)^(b-1))
  g <- 1 #Uniform (0,1)
  c <- 3 #Assumption
  f/(c*g)
}

while (k < n) {
  y <- runif(1)
  u <- runif(1)
  if (u < ratio(y, a, b)){
    x[k] <- y
    k <- k+1
  }
}
```

```
x
}
```

```
n1 <- 10000
a <- 3
b <- 2

x1 <- beta(n1, a, b)
hist(x1, freq=F, col='blue4', main='Beta(3,2)', breaks=50)
curve(dbeta(x,a,b),add=T, col='red4', lw=2)
```




```
p <- seq(.1, .9, .1)
Qhat <- quantile(x1, p) #quantile of sample
Q <- qbeta(p,3,2)      #theoretical quantiles
round(rbind(Qhat,Q),2)
```

```
##       10%  20%  30%  40%  50%  60%  70%  80%  90%
## Qhat 0.33 0.42 0.49 0.56 0.61 0.67 0.73 0.79 0.86
## Q    0.32 0.42 0.49 0.56 0.61 0.67 0.73 0.79 0.86
```

4.

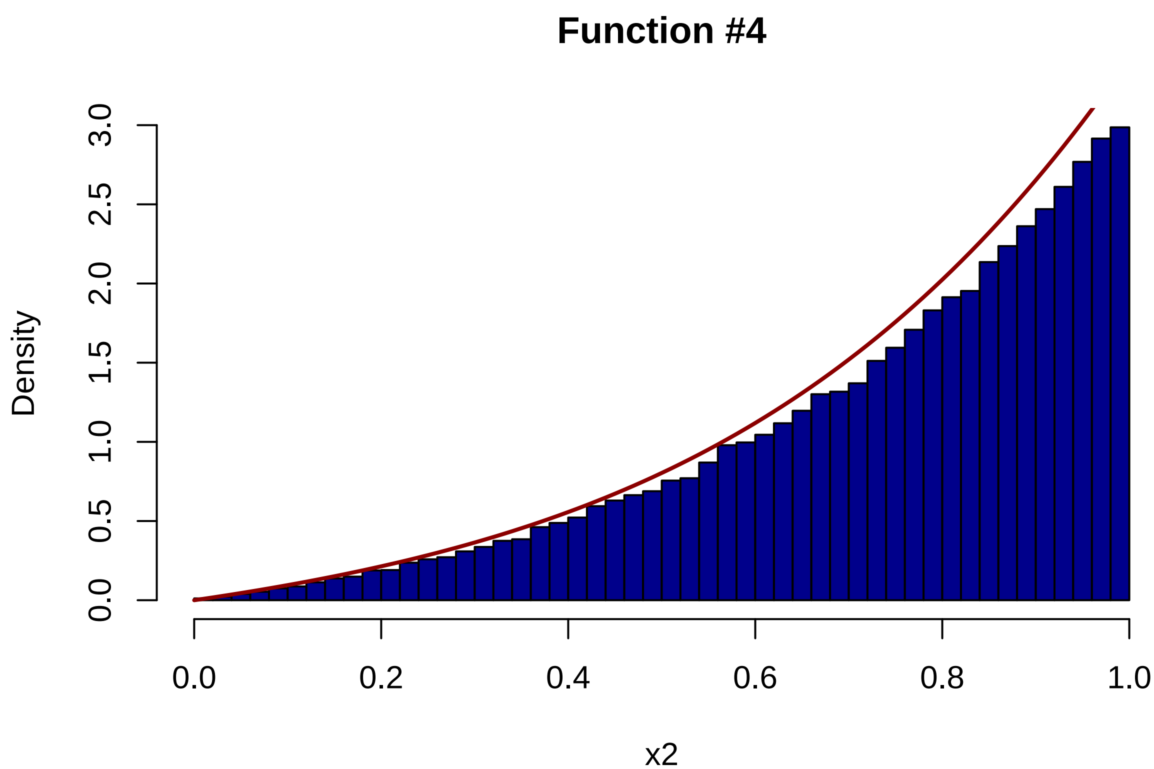
Write a function to generate a random sample of size n from the distribution with pdf,

$$f(x) = \frac{60}{70}(x + x^2 + x^3 + x^4), \quad 0 < x < 1.$$

Generate a random sample of size 1000 from this distribution. Graph the density histogram of the sample and compare it with the density function $f(x)$.

```r
func4 <- function(n){
k <- 0 #indicator for accepted value
x <- integer(n) #vector to store the accepted value

# define the function for f(x)/cg(x)
ratio2 <- function(x){
  f <- (60/70)*(x+x^2+x^3+x^4)
  g <- 1 #Uniform (0,1)
  c <- 4 #Assumption
  f/(c*g)
}

while (k < n) {
  y <- runif(1)
  u <- runif(1)
  if (u < ratio2(y)){
    x[k] <- y
    k <- k+1
  }
}
x
}
```

```r
n2 <- 100000
x2 <- func4(n2)
hist(x2, freq=F, col='blue4', main='Function #4', breaks=50)
curve((60/70)*(x+x^2+x^3+x^4),add=T, col='red4', lw=2)
```



5.

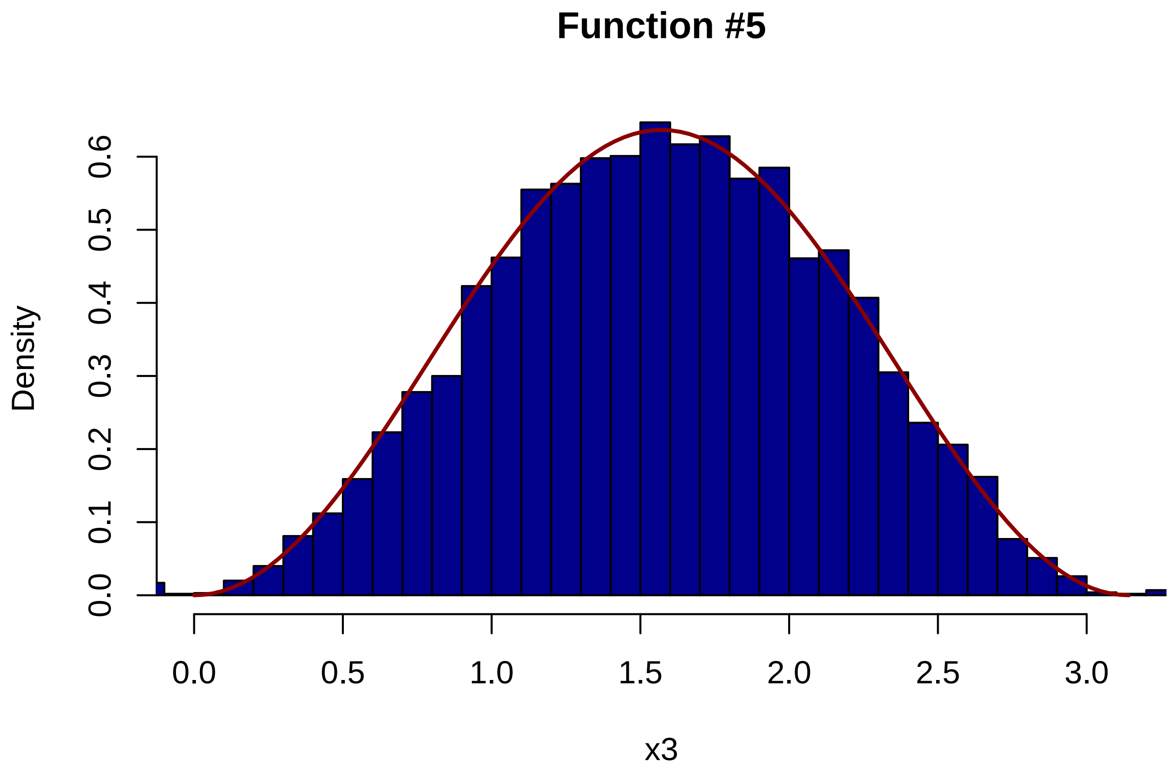
Write a function to generate a random sample of size n from the distribution with pdf,

$$f(x) = \frac{2}{\pi} sin^2 x, \quad 0 < x < \pi.$$

Generate a random sample of size 1000 from this distribution. Graph the density histogram of the sample and compare it with the density function f(x).

```
func5 <- function(n){
k <- 0 #indicator for accepted value
x <- integer(n) #vector to store the accepted value

# define the function for f(x)/cg(x)
ratio3 <- function(x){
  f <- (2/pi)*(sin(x)^2)
  g <- dnorm(x,1.58,1/1.5) #Uniform (0,1)
  c <- 1.15 #Assumption
  f/(c*g)
}

while (k < n) {
  y <- rnorm(1,1.58,1/1.5)
  u <- runif(1)
  if (u < ratio3(y)){
    x[k] <- y
    k <- k+1
  }
}
x
}
```

```
n3 <- 10000
x3 <- func5(n3)
hist(x3, freq=F, col='blue4', main='Function #5', breaks=50, xlim=c(0,pi))
curve((2/pi)*(sin(x)^2), col='red4', add=T, lw=2)
```


Function #5
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x3