Math 475: Homework 1

Hannah Zmuda

October 1, 2020

Question 1

The goal of this problem is to find the inverse CDF of the given density:

$$f_X(x) = exp(x - e^x)$$

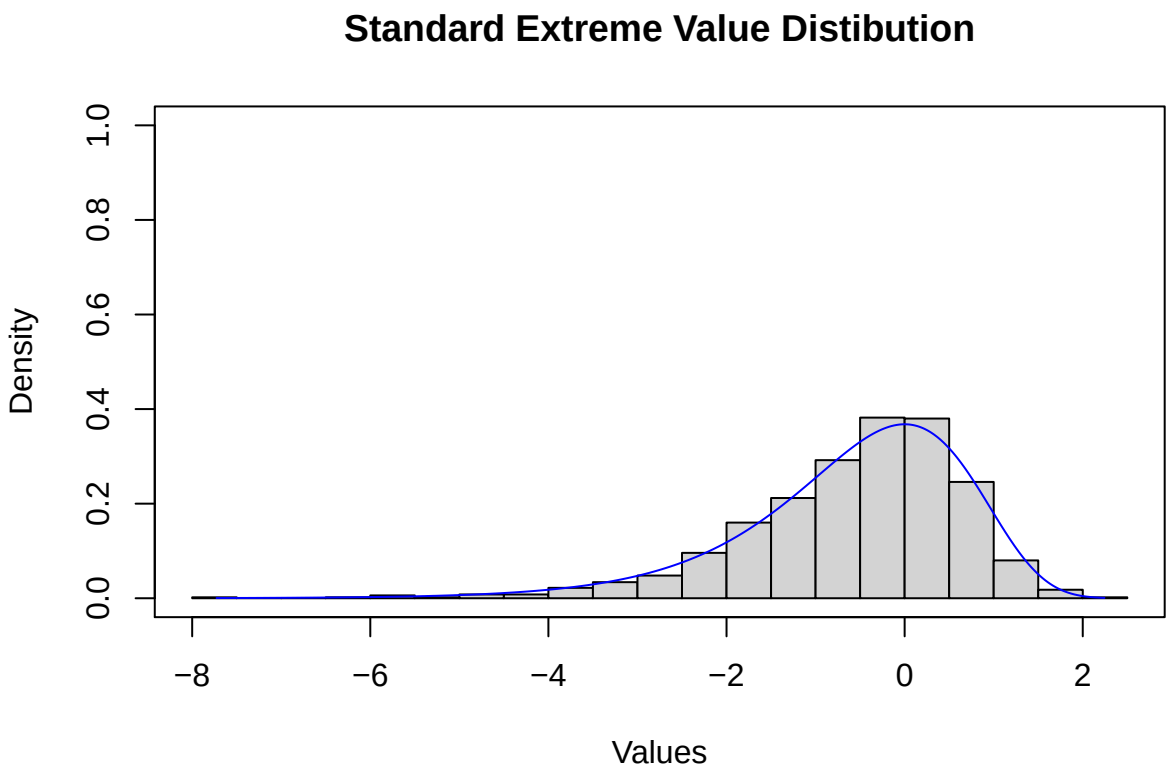
This will be done by creating an algorithm to simulate the standard extreme value distribution. In order to find the distribution of the given density, i first find the CDF of the density by integrating the equation above:

$$F_x(x) = -e^{-e^x}$$

And then take the inverse of the equation:

$$F_x^{-1}(x) = ln(-ln(-x))$$

```
set.seed(200)
n <- 1000 #sample number
#Random Number Generator, numbers are evenly distributed between 0 and 1
U1 <- runif(n,0,1)
#the standard extreme value distribution density function
f <- function(x){exp(x-exp(x))}
#inverse of the standard extreme value distribution density function,
#run uniform data through (U1)
q <- function(U1){log(-log(1-U1))}
#histogram of simulated data
hist(q(U1), prob = TRUE, main = "Standard Extreme Value Distibution",
     ylim = c(0, 1), xlab = "Values", ylab ="Density",breaks = 20)
x <- seq(min(q(U1)), max(q(U1)), 0.01)
lines(x,f(x), col = "blue") #density curve of f(x)

box()
```


Standard Extreme Value Distibution
Density
0.0
0.2
0.4
0.6
0.8
1.0
−8
−6
−4
−2
0
2
Values

Question 2

The objective of question 2 is to develop an algorithm to simulate the Rayleigh distribution. This will be in the form of a function with two inputs: the sample size as n and the scale parameter as σ. The Rayleigh distribution has a density of;

$$f(x) = \frac{x}{\sigma^2} exp(-\frac{x^2}{2\sigma^2})$$

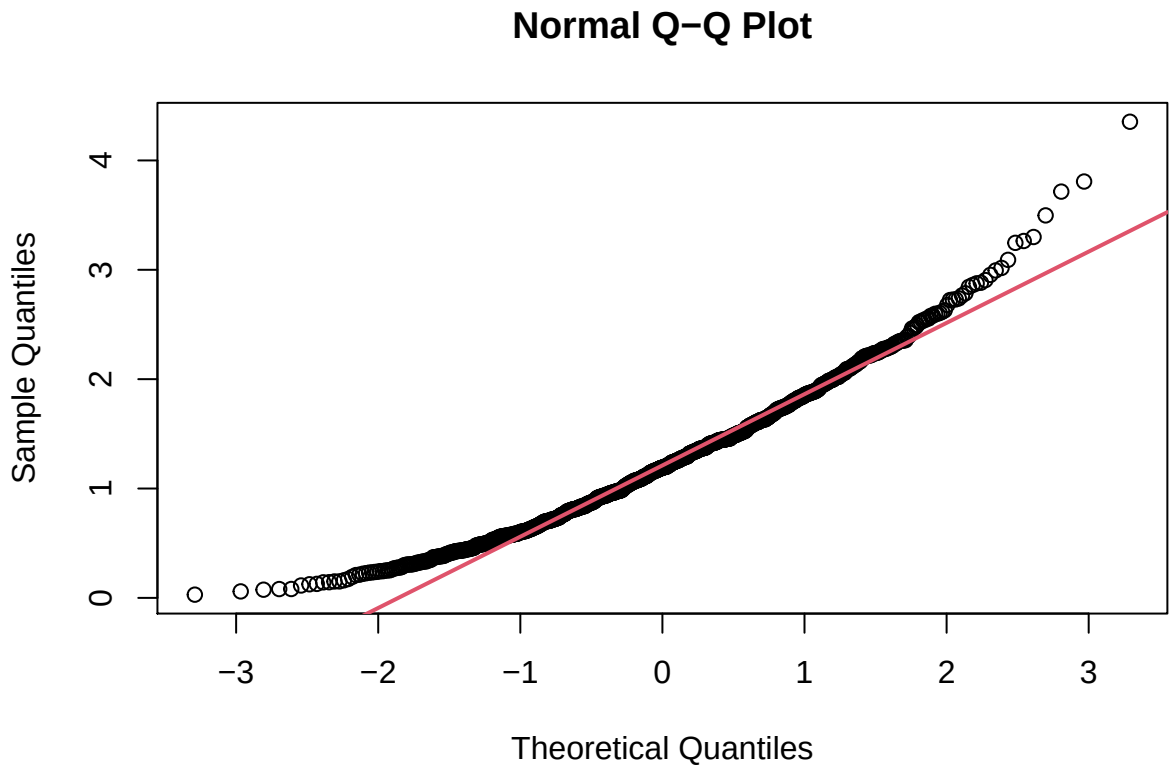
Integrating density:

$$F_x(x) = -exp(-\frac{x^2}{2\sigma^2})$$

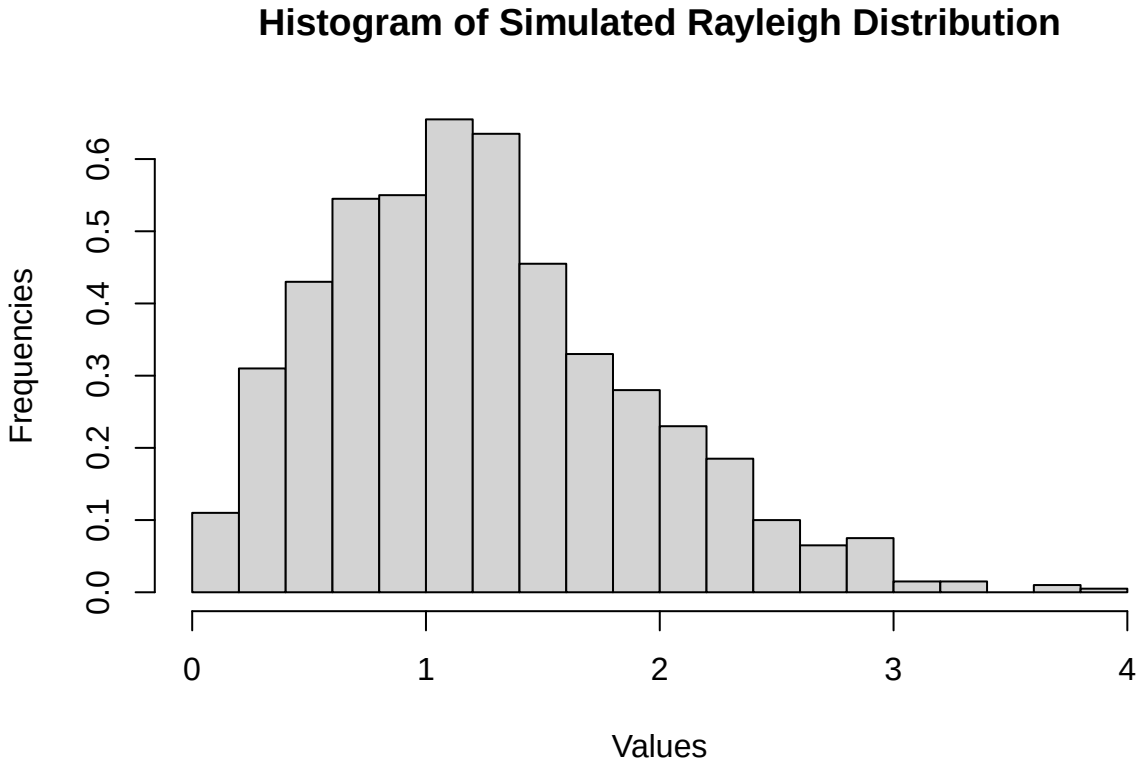
Inverse of the CDF (of the given density function) is:

$$F^{-1}(x, \sigma) = \sigma\sqrt{-2ln(1-x)}$$

```
set.seed(200)
#Inverse CDF approach for the Rayleigh distribution
r <- function(n,s){
  U2 <- runif(n,0,1)
  x <- s*sqrt(-2*log(1-U2))
  return(x)
}
#Check normality
qqnorm(r(1000,1))
qqline(r(1000,1),lwd=2,col=2)
```




```
#Histogram of simulated data
hist(r(1000,1), prob = TRUE,xlab = "Values",ylab = "Frequencies",
     main = "Histogram of Simulated Rayleigh Distribution",breaks = 20)
```


Histogram of Simulated Rayleigh Distribution
Frequencies
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
2
3
4
Values

Question 3

When using Bayesian Inference, we sometimes simulate the prior, θ, and maximize it to ensure we are maximizing the mean of the distribution. This can be done by accepting θ as $u \leq \frac{f(x|\theta)}{f(x|\hat{\theta})}$ where u $U(0,1)$ and $\hat{\theta}$ is the MLE from maximizing $f(x|\theta)$. Because we are given the following:

$$u \leq \frac{f(x|\theta)}{f(x|\hat{\theta})}$$

In order to use the maximum acceptance-rejection method, we need to confirm we can use it using the above equation. Based on the derivation in Rizzo et al, we can use a randomly generate number, U, and write the following integral:

$$P(\theta \leq U|x) = \int_0^U \pi(\theta)\frac{f(x|\theta)}{f(x|\hat{\theta})}d\theta$$

Solving the integral gets us the following:

$$p(U|x) = \pi(U)\frac{f(x|U)}{f(x|\hat{\theta})}$$

where $\pi(U)$ is a gamma distribution, the likelihood is $f(x|\theta) = \theta e^{-\theta x}$, and the maximum likelihood can be solved by the follwoing:

$$\frac{d}{d\hat{\theta}}(\hat{\theta}e^{-\hat{\theta}x}) = 0$$

$$-(x\hat{\theta} - 1)e^{-\hat{\theta}x} = 0$$

$$-\hat{\theta}x + 1 = 0$$

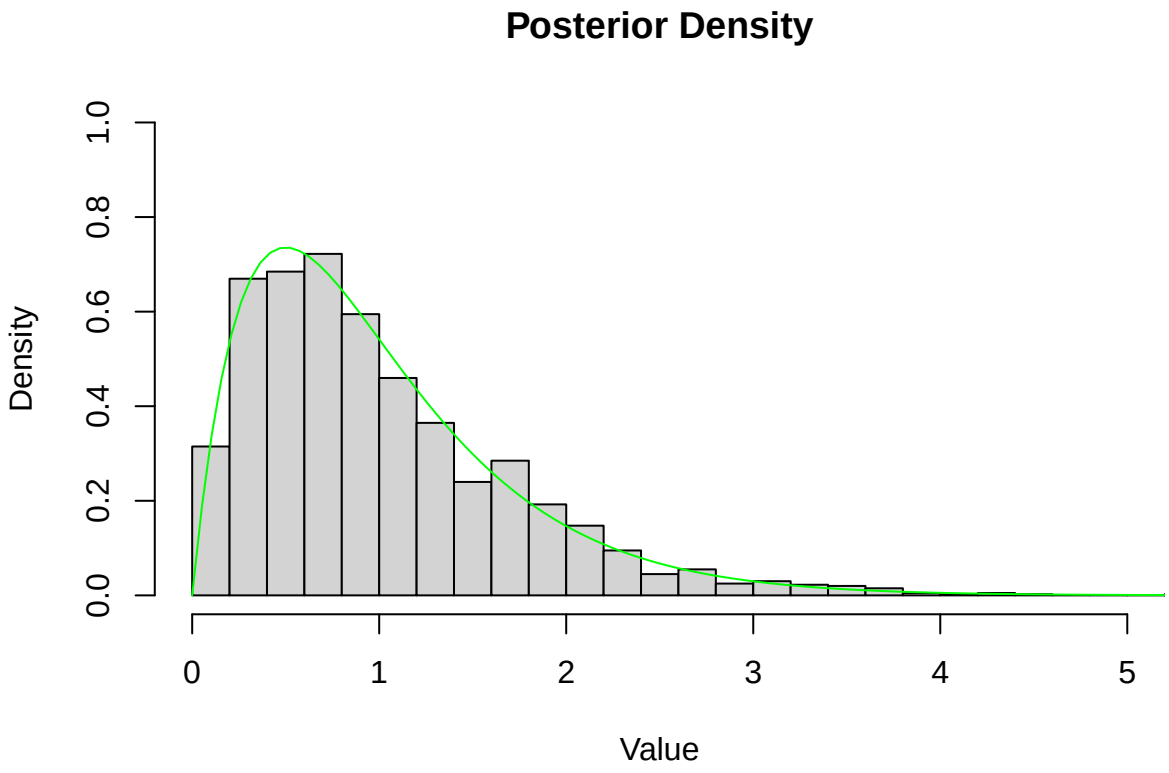
$$1 = \hat{\theta}x$$

$$\hat{\theta} = \frac{1}{x}$$

With this in mind, we can use the acceptance-rejection method to simulate $u \leq \frac{f(x|\theta)}{f(x|\hat{\theta})}$ to find teh posterior density $P(\theta|x)$.

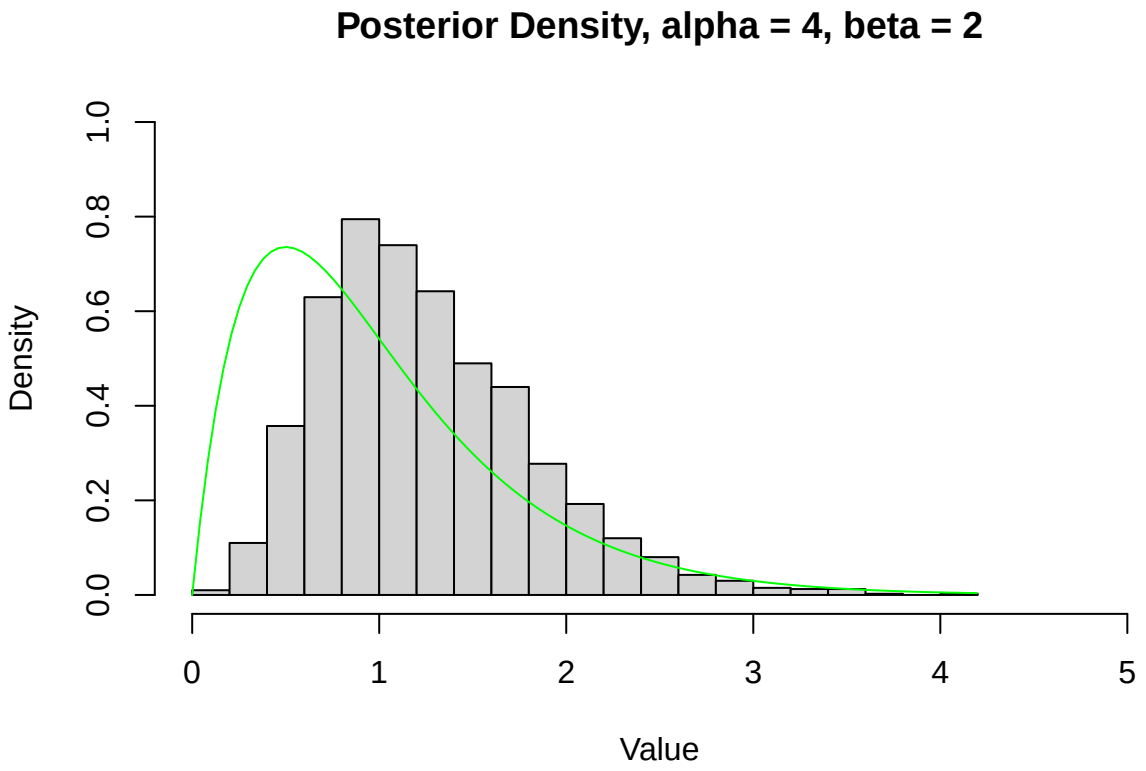
```
#Functions
f <- function(t,x){return(t*exp(-t*x))}
acceptReject <- function(alpha,beta,n,x){
  #MLE/theta hat equation
  thetaHat <- 1/x
  #number of times theta hat is accepted
  acceptRate <- 0
  #number of times gone through loop
  i <- 1
  #initialize the output vector
  result <- vector(length = n)
  while(acceptRate <= n){
    #step2a: generate U2 to be a uniform distribution
    U2 <- runif(1)
    #step2b: create theta in terms of a gamma distribution
    theta <- rgamma(1,shape = alpha, rate = beta) #don't need alpha + 1 and
    #beta + x becuase that is conjugate prior for posterior
    #step 3: accept U2 and set theta=U2, if U2 <= f(theta)/f(MLE)
    if(U2 <= f(theta,x)/f(thetaHat,x)){
      acceptRate = acceptRate + 1
```

```
      result[acceptRate] = theta
    }
    i = i + 1
  }
  return(list(result,acceptRate))
}

#posterior
set.seed(100)
n <- 2000#number of samples
alpha <- 1 #alpha
beta <- 1 #beta
x <- 1 #x-value
X <- acceptReject(alpha,beta,n,x)
result <- X[[1]]

#graphs
hist(result, probability = T, breaks = 20,xlim = c(0,5),ylim = c(0,1),
     main = "Posterior Density",xlab = "Value",ylab = "Density")
f2 <- function(x2){return(4*x2*exp(-2*x2)/gamma(2))}
curve(f2,0,max(result),add = T, col = "green")
```




```
#posterior
set.seed(100)
n <- 2000#number of samples
alpha <- 4 #alpha
beta <- 2 #beta
x <- 2 #x-value
X <- acceptReject(alpha,beta,n,x)
```

```r
result <- X[[1]]

#graphs
hist(result, probability = T, breaks = 20,xlim = c(0,5),ylim = c(0,1),
     main = paste("Posterior Density, alpha = 4, beta = 2" ),
     xlab = "Value",ylab = "Density")
f2 <- function(x2){return(4*x2*exp(-2*x2)/gamma(2))}
curve(f2,0,max(result),add = T, col = "green")
```



Based on the histogram produced, the histogram moves over towards $x- > \infty$ as α and β increase, and less is enveloped by the expected posterior curve. As alpha and beta increase, fewer points are under the envelope compared to alpha and beta equal to 1. As x increases, the posterior moves back under the curve, but then the peak increases.

Question 4

For the approximation $\int_0^4 te^{2t}dt$ via Gaussian quadrature with three evaluation points, we get the following set of equations.

$$2 = w_1 + w_2 + w_3$$
$$0 = w_1x_1 + w_2x_2 + w_3x_3$$
$$\frac{2}{3} = w_1x_1^2 + w_2x_2^2 + w_3x_3^2$$
$$0 = w_1x_1^3 + w_2x_2^3 + w_3x_3^3$$
$$\frac{2}{5} = w_1x_1^4 + w_2x_2^4 + w_3x_3^4$$
$$0 = w_1x_1^5 + w_2x_2^5 + w_3x_3^5$$

Using symmetry, $w_2 = 0$ and $x_2 = 0$. This then simplifies the eset of equations and gives us the the following solutions:

$$w_1 = w_3 = 1$$
$$x_1 = \sqrt{\frac{3}{5}}$$
$$x_3 = -\sqrt{\frac{3}{5}}$$

We can confirm these values using the gaussLegendre function.

```
library(pracma)
##  Quadrature with Gauss-Legendre nodes and weights
#evaluation points: sqrt(3/5),0,-sqrt(3/5)
#function
f <- function(x) x*exp(2*x)
# m = 2n-1 = (2*3)-1 = 5
cc <- gaussLegendre(3,-1,1)
#sum the weights*f(x)
Q <- sum(cc$w * f(cc$x*2 + 2))*2
#print weights
print(cc$w) #weights
```

```
## [1] 0.5555556 0.8888889 0.5555556
```

```
print(cc$x) #x-values
```

```
## [1] -7.745967e-01  8.881784e-16  7.745967e-01
```

```
print(Q) #Approximate Value
```

```
## [1] 4967.107
```

```
#Solve for exact function
integrand <-function(x) x*exp(2*x)
exact <- integrate(f,0,4)
exactValue <- exact$value
print(exactValue)#"Exact Value"
```

```
## [1] 5216.926
```

```
error <- abs((Q-exactValue)/exactValue)*100
print(error)#error
```

```
## [1] 4.788639
```

Based on the simulation, the approximation error is 4967.107 with an approximation error of 4.789%.