# Homework1_Question1

Hannah Zmuda

## 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:

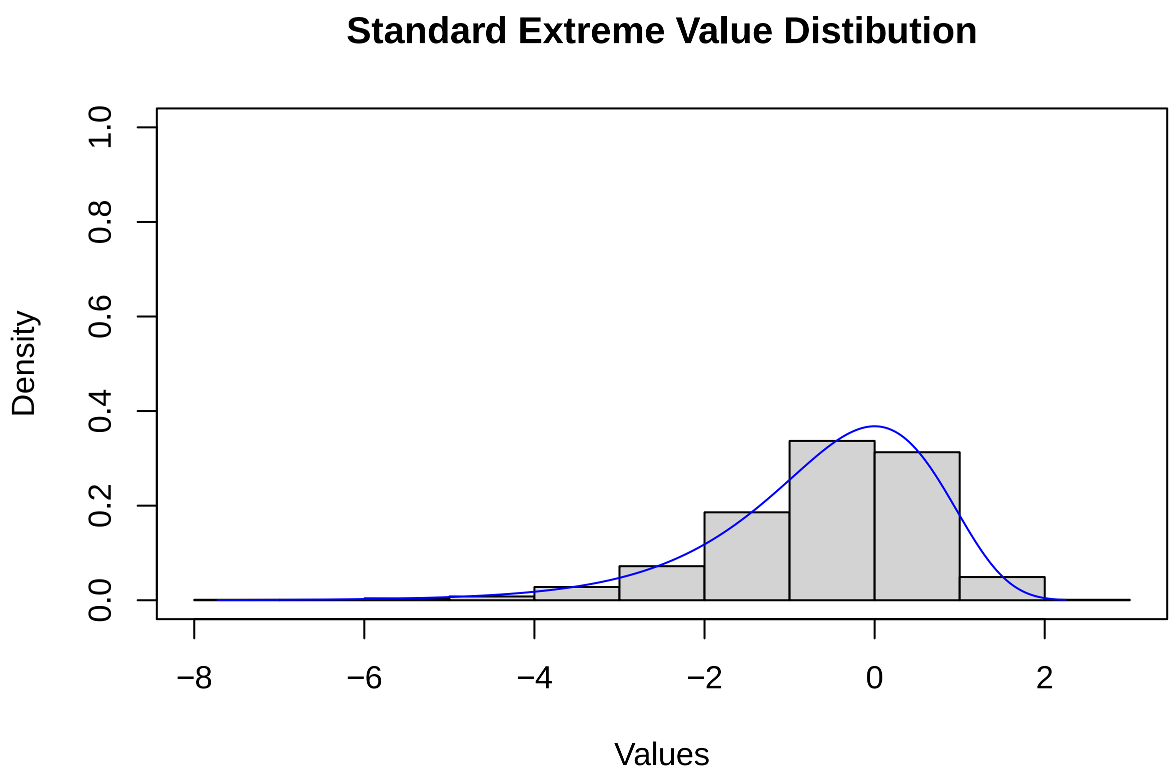
$$F_x(x) = -e^{-e^x}$$

And then take the inverse of the equation:

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

```
set.seed(200)
n <- 1000 #sample number
U1 <- runif(n,0,1) #Random Number Generator, numbers are evenly distributed between 0 and 1
f <- function(x){exp(x-exp(x))}#the standard extreme value distribution density function
q <- function(U1){log(-log(1-U1))}#inverse of the standard extreme value distribution density function,
#histogram of simulated data
hist(q(U1), prob = TRUE, main = "Standard Extreme Value Distibution", ylim = c(0, 1), xlab = "Values", 
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 $\sigma$. The Rayleigh distribution has a density of;

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

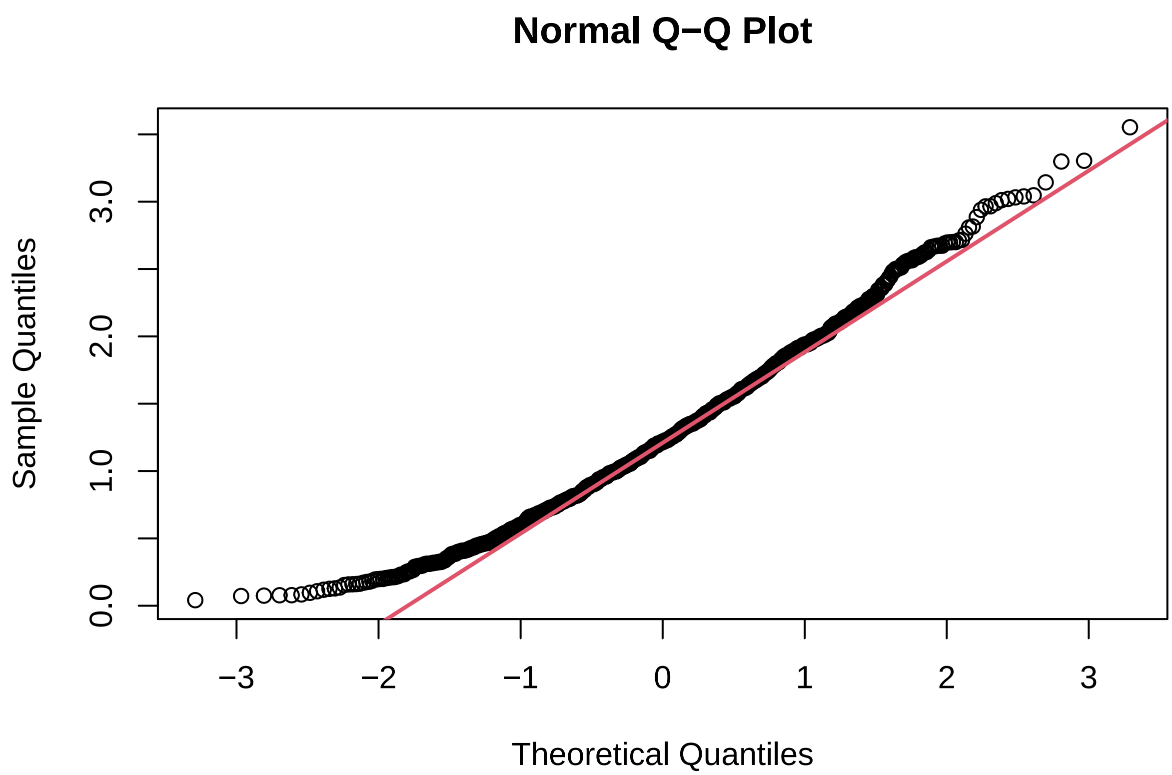
Integrating density:

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

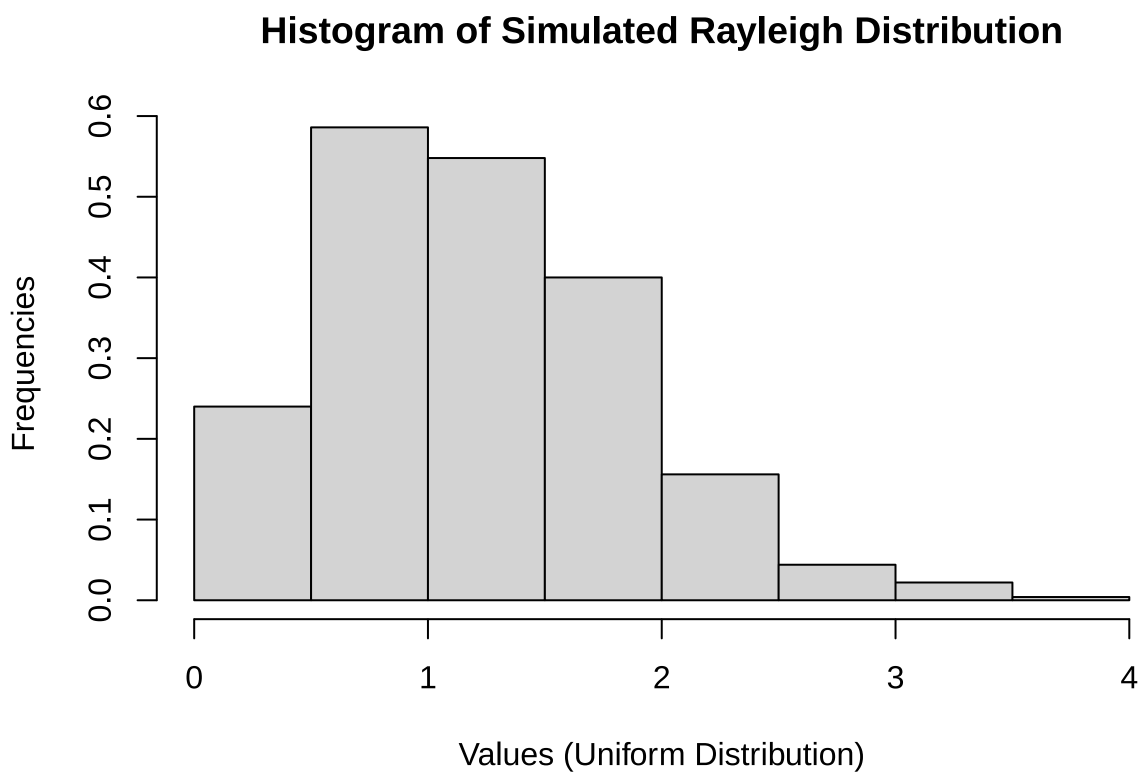
Inverse of the CDF (of the given density function) is:

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

```
set.seed(475)
#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 (Uniform Distribution)",ylab = "Frequencies",main = "Histogra
```


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 (Uniform Distribution)

# Question 3

When using Bayesian Inference, we sometimes simulate the prior, $\theta$, and maximize it to ensure we are maximizing the mean of the distribution. This can be done by accepting $\theta$ 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:

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

Based on this equation, we can simplify the equation through the following proof:

$$u \leq \frac{f(x|\theta)}{\int_\theta \theta e^{-\theta x} \pi(\theta) d\theta}$$

$$u \leq \frac{\theta e^{-\theta x}}{\int_\theta \theta e^{-\theta x} (\frac{\beta^\alpha \theta^{\alpha-1} e^{-\beta\theta}}{\gamma(\alpha)}) d\theta}$$

$$u \leq \frac{1}{(\frac{\beta^\alpha}{\gamma(\alpha)})} \frac{\theta e^{-\theta x}}{\int_0^1 \theta e^{-\theta x} \theta^{\alpha-1} e^{-\beta\theta} d\theta}$$

Because $\pi(\theta)$ does not depend on $x$, we can take that out of the integral.

$$u \leq \frac{1}{(\frac{\beta^\alpha \theta^{\alpha-1} e^{-\beta\theta}}{\gamma(\alpha)})} \frac{\theta e^{-\theta x}}{\frac{(-x^\theta+1)(e^{-\theta x})}{x^2}}$$

$$u \leq \frac{1}{M} \frac{f(\theta)}{g(\theta)}$$

With this in mind, we can set our equations as follows:

$$f(\theta) = \theta e^{-\theta x}$$

$$g(\theta) = \theta e^{-\theta x}$$

$$M = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\gamma(\alpha)}$$

These can be applied to find the posterior density $P(\theta|x)$.

```r
#Functions
f <- function(t){t*exp(-t*x)}
g <- function(t){-(((x^t) + 1)*(exp(-t*x)))/(x^2)}
acceptReject <- function(alpha,beta,n){
  #functions from proofs
  x <- rexp(1)#first x value is randomly generated, then updated as the algorithm moves on
  print(x)
  #step1: generate Y (aka U1 in acceptance-rejection thm.)
  Y <- rgamma(n,alpha + 1, beta + x) #theta value
  print(Y)
  #step2: generate U2 to be a uniform distribution
  U2 <- runif(n,0,1)
  print(U2)
  #step 3: accept U2 and set theta=U2, if U2 <= f(theta)/g(theta)/M
  i = 1
  M <- (beta^alpha)/(factorial(alpha-1)) #M is a constant but changes with x
    while(i <= n)
```

```
      {
      if(U2[i] <= f(Y[i])/g(Y[i])/M )
        {
          assign(x,Y[f((Y[i])/g(Y[i])/M)])
          }
      i = i +1
      M <- (beta^alpha)/(factorial(alpha-1)) #M is a constant but changes with x
    }
  return(x)
}

set.seed(475)
n <- 5#number of samples
alpha <- 1 #alpha
beta <- 1 #beta
#X <- acceptReject(alpha,beta,n)
#hist(X,freq=FALSE,ylim=c(0,0.5), main=paste("Acceptance Rate:",length(X)/n))
#lines(X,dgamma(X),col = 'red')#true normal pdf
#lines(xx,g(xx)*m(x),col = 'green')#adjusted pdf
```

# Question 4

```r
library(pracma)
##  Quadrature with Gauss-Legendre nodes and weights
#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("weights and x values")
```

```
## [1] "weights and x values"
```

```r
print(cc)
```

```
## $x
## [1] -7.745967e-01  8.881784e-16  7.745967e-01
##
## $w
## [1] 0.5555556 0.8888889 0.5555556
```

```r
print("Approximate Value")
```

```
## [1] "Approximate Value"
```

```r
print(Q)
```

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

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

```
## [1] "Exact Value"
```

```r
print(exactValue)
```

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

```r
error <- abs((Q-exactValue)/exactValue)*100
print("Error")
```

```
## [1] "Error"
```

```r
print(error)
```

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