STAA 567: HW 2

Matthew Stoebe

# Q1

## Q1A

Write the Log likelyhood Function

## Q1B

Find the First Derrivative

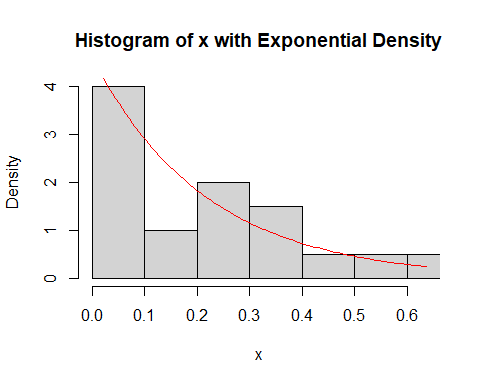
## Q1C

Find the Fomula for Lambda hat

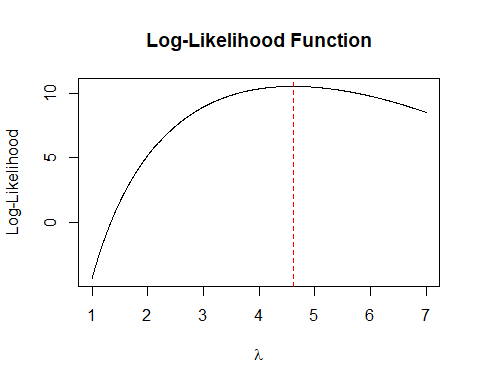
## Q1D

## [1] 4.607375

## Q1E



## Q1f



## [1] 4.607375

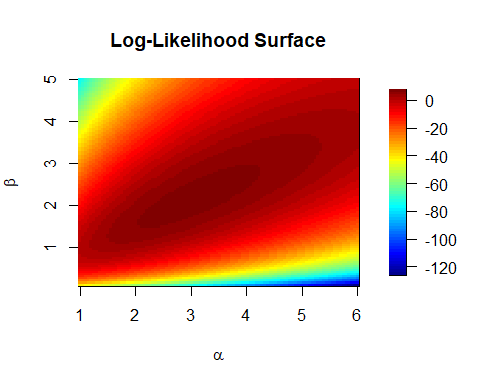
## Q1I

## Iteration 1 : lambda = 4.04661   
## Iteration 2 : lambda = 4.539124   
## Iteration 3 : lambda = 4.606364   
## Iteration 4 : lambda = 4.607375   
## Iteration 5 : lambda = 4.607375

## [1] 4.607375

# Q2

## Q2A

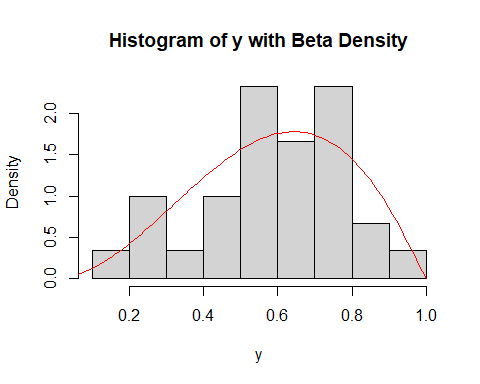


## Q2B

## alpha beta   
## 3.014761 2.115292

## function gradient   
## 55 NA

## Q2C



## Q2D

## alpha beta   
## 3.015065 2.115614

## function gradient   
## 23 9

# Appendix

#Retain this code chunk!!!  
library(knitr)  
knitr::opts\_chunk$set(echo = FALSE)  
knitr::opts\_chunk$set(message = FALSE)  
knitr::opts\_chunk$set(warning = FALSE)  
  
#install.packages("field")  
  
#install.packages("tinytex")  
  
#Q1D  
load("./Data/expData.RData")  
  
lambda\_hat <- 1 / mean(x)  
lambda\_hat  
hist(x, freq = FALSE, xlim = c(0, max(x)), main = "Histogram of x with Exponential Density", xlab = "x")  
  
x\_vals <- seq(0, max(x), length.out = 100)  
dens\_vals <- dexp(x\_vals, rate = lambda\_hat)  
lines(x\_vals, dens\_vals, col = "red")  
lambda\_seq <- seq(1, 7, length.out = 1000)  
  
logLik\_vals <- sapply(lambda\_seq, function(lambda) {  
 n <- length(x)  
 n \* log(lambda) - lambda \* sum(x)  
})  
  
plot(lambda\_seq, logLik\_vals, type = 'l', xlab = expression(lambda), ylab = "Log-Likelihood", main = "Log-Likelihood Function")  
  
abline(v = lambda\_hat, col = 'red', lty = 2)  
  
neg\_logLik <- function(lambda) {  
 if (lambda <= 0) return(Inf)  
 n <- length(x)  
 - (n \* log(lambda) - lambda \* sum(x))  
}  
  
opt\_result <- optimize(neg\_logLik, interval = c(1, 7))  
lambda\_hat\_opt <- opt\_result$minimum  
lambda\_hat\_opt  
newton\_raphson <- function(x, lambda\_init = 3, tol = 1e-6, max\_iter = 100) {  
 n <- length(x)  
 S <- sum(x)  
 lambda\_old <- lambda\_init  
 for (i in 1:max\_iter) {  
 dL <- n / lambda\_old - S  
 d2L <- - n / lambda\_old^2  
 lambda\_new <- lambda\_old - dL / d2L  
 cat("Iteration", i, ": lambda =", lambda\_new, "\n")  
 if (abs(lambda\_new - lambda\_old) < tol) {  
 break  
 }  
 lambda\_old <- lambda\_new  
 }  
 return(lambda\_new)  
}  
  
lambda\_hat\_nr <- newton\_raphson(x)  
lambda\_hat\_nr  
#Q2A  
load("./Data/betaData.RData")  
  
logLik\_beta <- function(alpha, beta, y) {  
 if (alpha <= 0 || beta <= 0) {  
 return(NA)  
 }  
 sum(dbeta(y, shape1 = alpha, shape2 = beta, log = TRUE))  
}  
  
alpha\_seq <- seq(1, 6, length.out = 100)  
beta\_seq <- seq(0.1, 5, length.out = 100)  
  
logLik\_vals <- matrix(NA, nrow = length(alpha\_seq), ncol = length(beta\_seq))  
  
for (i in 1:length(alpha\_seq)) {  
 for (j in 1:length(beta\_seq)) {  
 logLik\_vals[i, j] <- logLik\_beta(alpha\_seq[i], beta\_seq[j], y)  
 }  
}  
  
library(fields)  
image.plot(alpha\_seq, beta\_seq, logLik\_vals, xlab = expression(alpha), ylab = expression(beta), main = "Log-Likelihood Surface")  
#Q2B  
neg\_logLik\_beta <- function(params, y) {  
 alpha <- params[1]  
 beta <- params[2]  
 if (alpha <= 0 || beta <= 0) {  
 return(Inf)  
 }  
 -sum(dbeta(y, shape1 = alpha, shape2 = beta, log = TRUE))  
}  
  
start\_params <- c(alpha = 3, beta = 3)  
  
result\_nelder <- optim(start\_params, neg\_logLik\_beta, y = y, method = "Nelder-Mead")  
result\_nelder$par # Estimated parameters  
result\_nelder$counts # Number of iterations  
  
#Q2C  
hist(y, freq = FALSE, main = "Histogram of y with Beta Density", xlab = "y")  
  
y\_vals <- seq(0, 1, length.out = 100)  
dens\_vals <- dbeta(y\_vals, shape1 = result\_nelder$par[1], shape2 = result\_nelder$par[2])  
lines(y\_vals, dens\_vals, col = 'red')  
#Q2D  
result\_bfgs <- optim(start\_params, neg\_logLik\_beta, y = y, method = "BFGS")  
result\_bfgs$par # Estimated parameters  
result\_bfgs$counts # Number of iterations