

STAT 231 Assignment 2

The purpose of this assignment is to use the software R to graph relative likelihood functions, determine likelihood intervals and to simulate a sampling distribution as described in Section 4.2 of the Course Notes.

The code for this assignment is posted both as a text file called `RCodeAssignment2.txt` and an R file called `RCodeAssignment2R.R` which are posted in the Assignment 2 folder in the Assignments folder under Content on Learn.

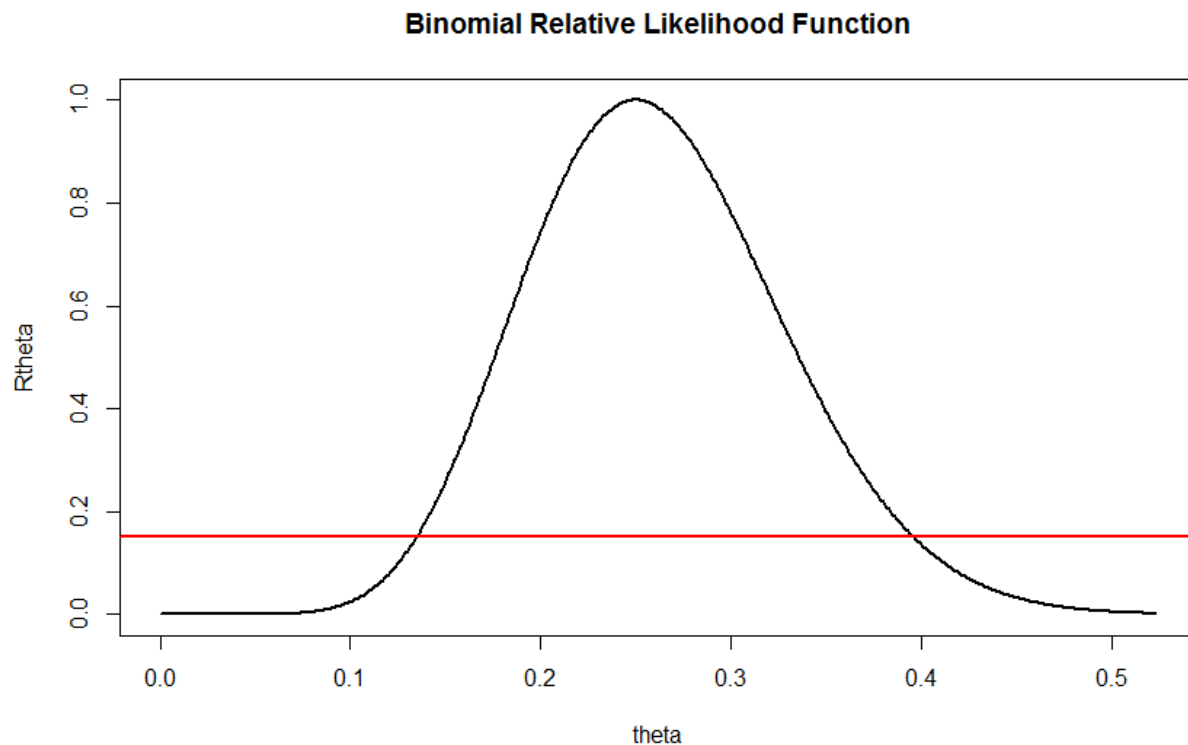
Problem 1: Run the following R code.

```
#####  
# run this code only once  
library(MASS) # truehist is in the library MASS  
#####  
  
#####  
# Problem 1: R code for Binomial relative likelihood function  
# The following R code will plot the Binomial relative likelihood function and the line  $y = 0.15$   
# which can be used to determine a 15% likelihood interval.  
id<-20456484  
set.seed(id)  
#generate a random value of theta from Beta distribution last 2 digits of ID avoiding zero values  
theta<-rbeta(1, max(1,id-10*trunc(id/10)), max(1,trunc(id/10)-10*trunc(id/100)))  
n<-40  
y<-rbinom(1,n,theta) # observation from Binomial(n,theta) distribution  
thetahat<-y/n #maximum likelihood estimate of theta  
# display values  
cat("n = ", n, ", theta = ", theta, ", y = ", y, ", thetahat = ", thetahat)  
s<-(thetahat*(1-thetahat)/n)^0.5  
# interval of values for plotting relative likelihood function  
th<-seq(max(0,thetahat-4*s), min(1,thetahat+4*s),0.001)  
# create function to calculate Binomial relative likelihood function  
BinRLF <- function(x) {exp(y*log(x/thetahat)+(n-y)*log((1-x)/(1-thetahat)))}  
plot(th, BinRLF(th),xlab="theta",type="l",lwd=2) # plot relative likelihood function  
# draw a horizontal line at 0.15  
abline(a=0.15,b=0,col="red",lwd=2)  
title(main="Binomial Relative Likelihood Function")  
#####
```

Verify that you obtain the following output and plots:

```
> y<-rbinom(1,n,theta) # observation from Binomial(n,theta) distribution  
> thetahat<-y/n #maximum likelihood estimate of theta
```

```
> # display values
> cat("n = ", n, ", theta = ", theta, ", y = ", y, ", thetahat = ", thetahat)
n = 40 , theta = 0.2666948 , y = 10 , thetahat = 0.25
```



From the graph of the relative likelihood function the 15% likelihood interval is approximately [0.13,0.40]. The R function uniroot can be used to determine the endpoints more accurately. Note that the lower endpoint lies between 0.1 and 0.15 and the upper endpoint lies between 0.35 and 0.42.

Run the following R code.

```
#####
# R code for finding the endpoints of a 15% likelihood interval for Binomial example
uniroot(function(x) BinRLF(x)-0.15,lower=0.1,upper=0.15)
uniroot(function(x) BinRLF(x)-0.15,lower=0.35,upper=0.42)
#####
```

Verify that you obtain the following output:

```
> uniroot(function(x) BinRLF(x)-0.15,lower=0.1,upper=0.15)
$root
[1] 0.1346055
$f.root
[1] -9.582756e-05
```

```
$iter
[1] 4
$estim.prec
[1] 6.103516e-05
```

```
> uniroot(function(x) BinRLF(x)-0.15,lower=0.35,upper=0.42)
$root
[1] 0.3960402
$f.root
[1] 0.0001050066
$iter
[1] 5
$estim.prec
[1] 6.103516e-05
```

From this information we now know the 15% likelihood interval is [0.1346055, 0.3960402].

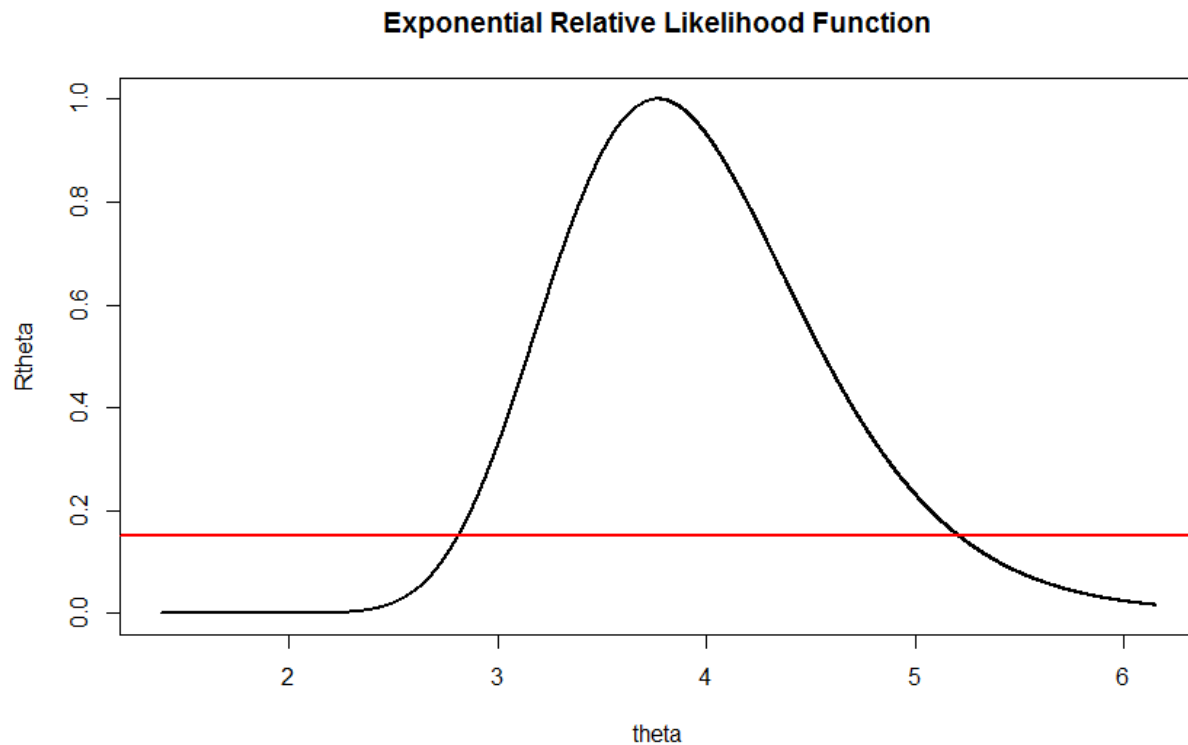
Problem 2: Run the following R code.

```
#####
# Problem 2: R code for Exponential relative likelihood function
# The following R code will plot the Exponential relative likelihood function and the line y = 0.15
# which can be used to determine a 15% likelihood interval.
set.seed(id)
theta<-rexp(1, 1/(max(1,id-10*trunc(id/10)))) #generate a random value of theta
n<-40
# generate random sample of n observation from Exponential(theta) distribution
y<-sort(rexp(n,1/theta))
y[1:3] # display first 3 numbers in the data set
thetahat<-mean(y) #maximum likelihood estimate of theta
cat("n = ",n," theta = ",theta," thetahat = ",thetahat) # display values
s<-thetahat/(n^0.5)
# interval of values for plotting relative likelihood function
th<-seq(max(0,thetahat-4*s), thetahat+4*s,0.001)
# create function to calculate Exponential relative likelihood function
ExpRLF<-function(x) {(thetahat/x)^n*exp(n*(1-thetahat/x))}
plot(th,ExpRLF(th),xlab="theta",type="l",lwd=2) # plot relative likelihood function
# draw a horizontal line at 0.15
abline(a=0.15,b=0,col="red",lwd=2)
title(main="Exponential Relative Likelihood Function")
#####
```

Verify that you obtain the following output and plots:

```
> y<-sort(rexp(n,1/theta))
> y[1:3] # display first 3 numbers in the data set
```

```
[1] 0.02241854 0.19706745 0.26119882
> thetahat<-mean(y)      #maximum likelihood estimate of theta
> cat("n = ",n," theta = ",theta," thetahat = ",thetahat)  # display values
n = 40  theta = 3.990256 , thetahat = 3.767605
```



From the graph of the relative likelihood function the 15% likelihood interval is approximately [2.8,5.2]. The R function uniroot can be used to determine the endpoints more accurately. Note that the lower endpoint lies between 2.5 and 3 and the upper endpoint lies between 5 and 5.5.

Run the following R code.

```
#####
# R code for finding the endpoints of a 15% likelihood interval for Exponential example
uniroot(function(x) ExpRLF(x)-0.15,lower=2.8,upper=3)
uniroot(function(x) ExpRLF(x)-0.15,lower=5,upper=5.5)
#####
```

Verify that you obtain the following output:

```
> uniroot(function(x) ExpRLF(x)-0.15,lower=2.8,upper=3)
$root
[1] 2.810856
$f.root
[1] 1.241236e-05
$iter
[1] 3
$estim.prec
```

```
[1] 6.103516e-05
```

```
> uniroot(function(x) ExpRLF(x)-0.15,lower=5,upper=5.5)
$root
[1] 5.212614
$f.root
[1] -3.365328e-07
$iter
[1] 5
$estim.prec
[1] 6.103516e-05
```

From this information we now know the 15% likelihood interval is [2.810856, 5.212614].

Problem 3: Run the following R code.

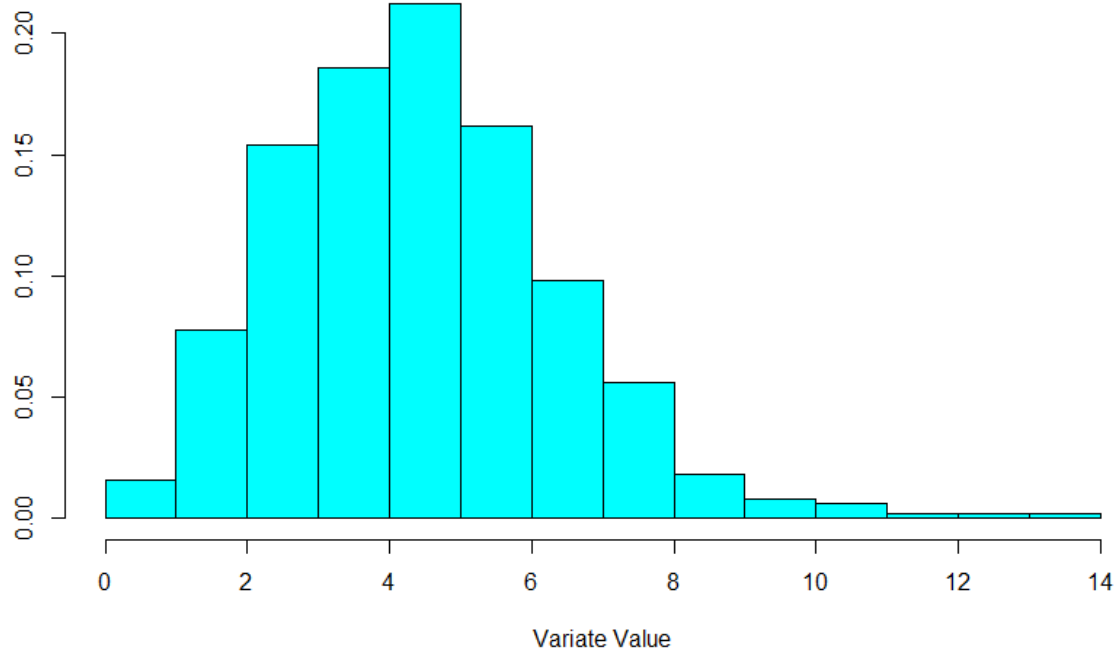
```
#####
# Problem 3: Sampling Distribution
set.seed(id)
# pop is a vector of 500 variate values for a finite population
pop<-rpois(500,max(1,id-10*trunc(id/10)))
mu<-mean(pop)      # population mean
popsd<-(499*var(pop)/500)^0.5
cat("population mean = ",mu, " population standard deviation = ",popsd)
truehist(pop,h=1,main="Population Histogram",xlab="Variate Value",)
k<-10000          # number of simulations
n<-15             # sample size
sim<-rep(0,k)     # vector to store sample means
# Calculate k sample means for samples of size n drawn from population pop
for (i in 1:k)
  sim[i]=mean(sample(pop,n,replace=F))
# display percentage of times sample mean is within 0.5 of true mean mu
prop<-mean(abs(sim-mu)<0.5)
cat("percentage of times sample mean is within 0.5 of true mean = ",prop)
truehist(sim,h=0.25,xlab="Sample Mean",main="Sampling Distribution of Sample Mean")
#####
```

Verify that you obtain the following output and plots:

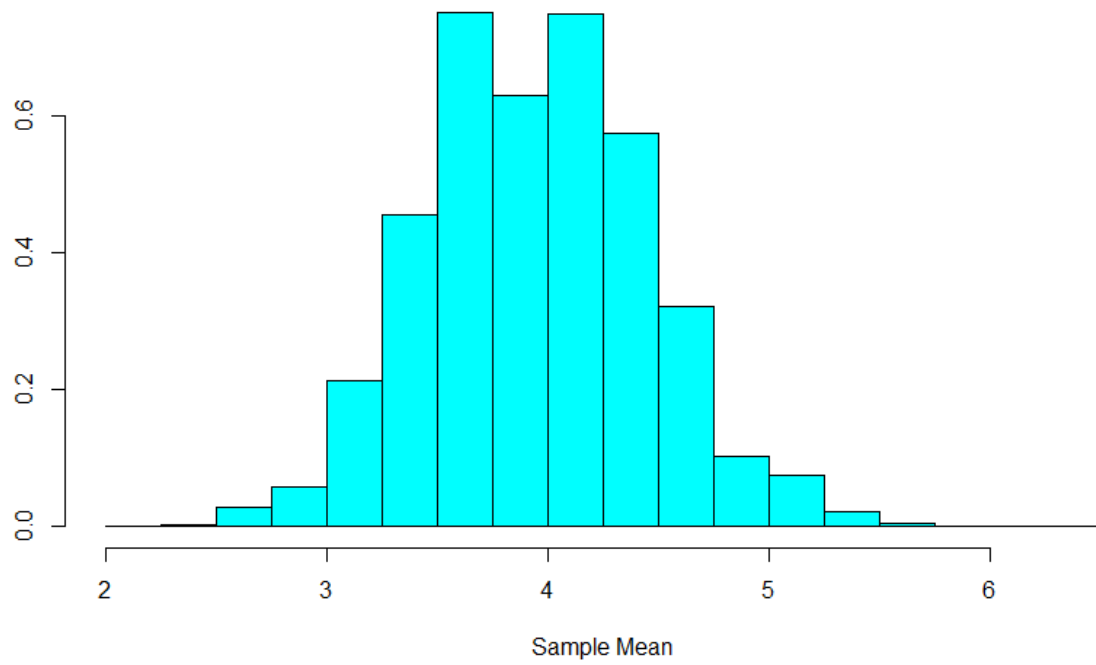
```
> cat("population mean = ",mu, " population standard deviation = ",popsd)
population mean = 3.93 population standard deviation = 1.976133

> cat("percentage of times sample mean is within 0.5 of true mean = ",prop)
percentage of times sample mean is within 0.5 of true mean = 0.6807
```

Population Histogram



Sampling Distribution of Sample Mean



Run the R code for the 3 problems above again except modify the line

```
"id<-20456484"
```

in Problem 1 by replacing the number 20456484 with your UWaterloo ID number.

When you run these examples with your ID number you will generate 4 new plots. Export these 4 plots as .png files using RStudio (See Introduction to R and RStudio Section 6).

Download the Assignment 2 Template which is posted as a Word document on Learn. Fill in the required information and plots based on the output for the data generated using your ID number. Your assignment must follow the template exactly. See Assignment 2 Example posted on Learn.

Create a .pdf file for the answer to EACH problem.

Upload your assignment to Crowdmark using the link which was emailed to you.