# STAT 261 Lab 2

Arjun Banik, Mary Lesperance

February 9, 2024

## General Instructions

- Execute each chunk of code to ensure that your code works properly.
- Save this .Rmd file and then knit the entire document to pdf.

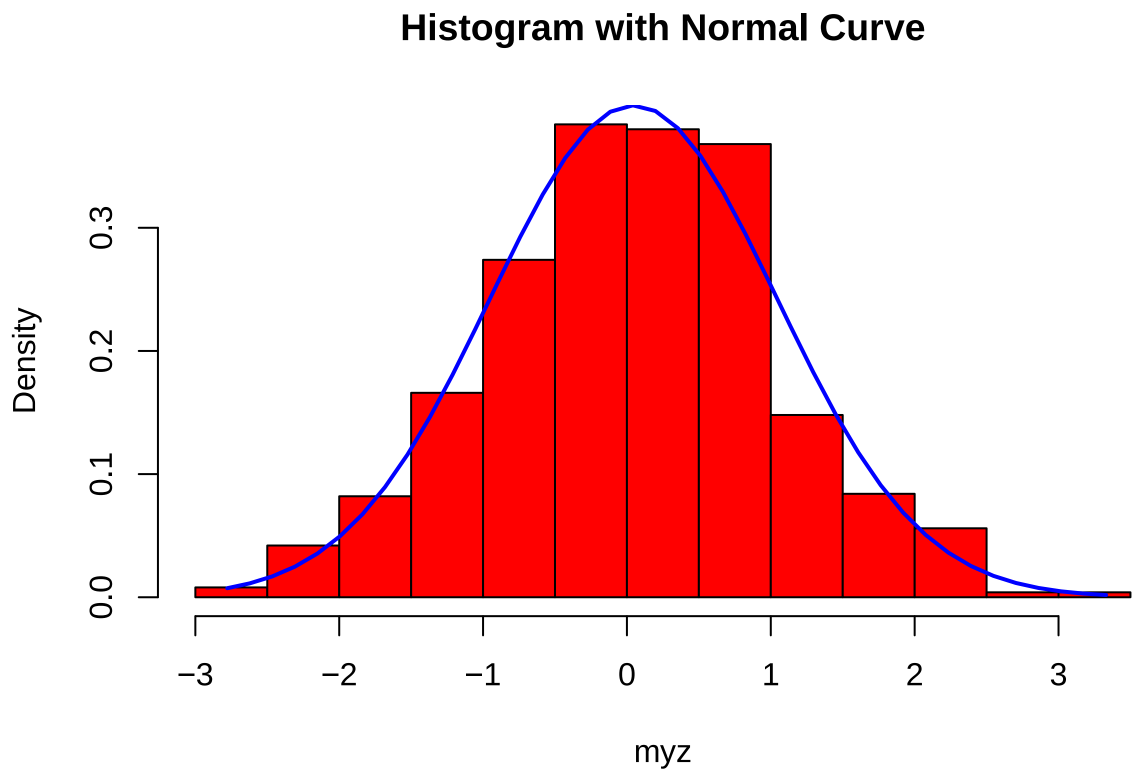
## Learning outcomes

- use of `hist()` to plot histograms of generated/observed data (and appearance customization!)
- Adding density lines to histograms to model the relationship between the Chi-square and standard normal distributions
- use of geometric distribution functions, `rgeom()` and `dgeom()`
- understanding how to code (log-)likelihood and (log-)relative likelihood functions in R
- use of `optimize()` to find the MLE of your (log-)likelihood function
- use of `uniroot()` to find 100p% likelihood intervals
- use of `round()` to present rounded numerical values

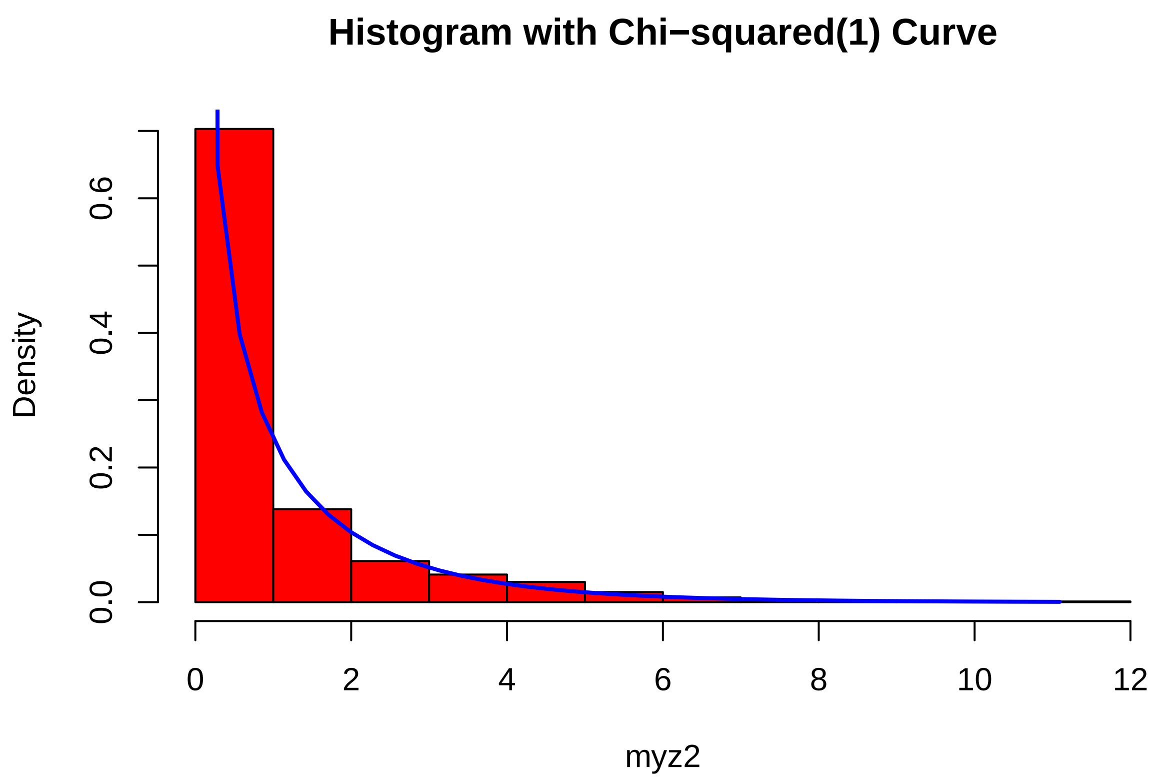
### 1. The Normal(0,1) and Chi-squared distribution.

In this section, we see the relationship between Normal(0,1) random variables and the Chi-squared(1) distribution. Normal(0,1)^2 ~ Chi-squared(1)

```r
# First generate 1000 random observations from N(0,1)
set.seed(12345)
myz <- rnorm(1000)

# Plot a scaled histogram and overlay normal pdf
h <- hist(myz, breaks=10, col="red",
    main="Histogram with Normal Curve", prob=TRUE)
xfit<-seq(min(myz), max(myz), length=40) #sequence of length 40
yfit<-dnorm(xfit, mean=mean(myz), sd=sd(myz)) #normal density
lines(xfit, yfit, col="blue", lwd=2) #overlay fitted normal density
```




```r
# as a little exercise, change the color in line 39 above to "forestgreen" and
# the linewidth to 50. Congratulations, you have just made your
# first piece of Data Art!! A very hungry caterpillar :^)

# Square these observations, plot a histogram and overlay the chi-square(1) pdf
myz2 <- myz^2
h <- hist(myz2, breaks=10, col="red",
    main="Histogram with Chi-squared(1) Curve", prob=TRUE)
xfit <- seq(min(myz2), max(myz2), length=40)
yfit <- dchisq(xfit, 1)     #chi-squared(1) density
lines(xfit, yfit, col="blue", lwd=2) #overlay Chi-squared(1) curve
```




### 2. Generate n=10 observations from the Geometric distribution.

*type ?rgeom() to read which form of the PMF R computes (i.e. does it include or exclude the final successful trial?)*

```
set.seed(54321)
n <- 10
geo.dat <- rgeom(n, prob=.07)    # this generates n geometric observations

geo.mle <- 1/(1+mean(geo.dat)) #We derived this MLE during the basketball tryouts example
geo.mle
```

```
## [1] 0.04048583
```

### 3. Write a function to compute the log-likelihood given the n observations and plot the log-likelihood.

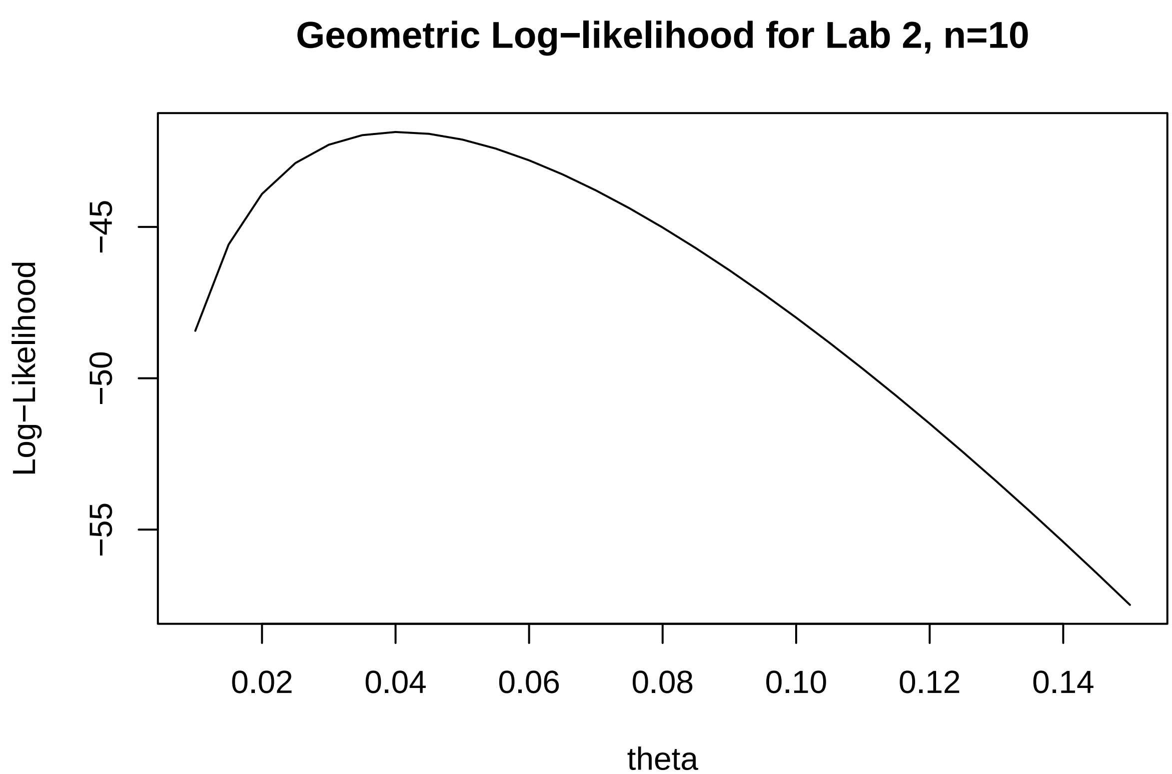
In this section, we write a function which computes the Geometric log-likelihood given arguments:
* theta, a scalar or vector of probabilities and
* x, a vector of observations.

```
# Write a function to compute geometric log-likelihood with arguments
#  theta = scalar or vector of geometric probabilities
#  x = vector of observations

ell <-  function(theta, x){
  loglike <- 0
```

```
  for (i in 1:length(x)){
    loglike <- loglike + dgeom(x[i], prob=theta, log=TRUE)
      #computes the sum of the log of geometric probabilities over each of the observations
  }
  return(loglike)
}

theta <- seq(0.01, 0.15, by=.005)  #a vector sequence of theta values
gloglike <- ell(theta, geo.dat)    #compute the log-likelihood for each value in theta, given data geo.
head(cbind(theta, gloglike))    #head prints the top values
```

```
##      theta  gloglike
## [1,] 0.010 -48.43363
## [2,] 0.015 -45.57898
## [3,] 0.020 -43.90827
## [4,] 0.025 -42.88912
## [5,] 0.030 -42.28441
## [6,] 0.035 -41.96771
```

```
plot(gloglike ~ theta, ylab='Log-Likelihood', xlab='theta', type='l') #plot(y ~ x) version
title(paste('Geometric Log-likelihood for Lab 2, n=', n, sep=''))
```




**4. Compute the MLE using the function we defined.**

```
#?optimize   #see the arguments and outputs for optimize; see optional arguments ...
# we pass a function to optimize, starting interval, and any other arguments required by ell
```

```r
thetahat <- optimize(ell, c(.02, .10), maximum=TRUE, x=geo.dat)
thetahat   #how does the maximum compare with 1/(1+mean(geo.dat)) computed above?
```

```
## $maximum
## [1] 0.04047153
##
## $objective
## [1] -41.86282
```

```r
thetahat$maximum  #extract the maximum
```

```
## [1] 0.04047153
```

```r
thetahat$objective  #extract value of function ell at maximum
```

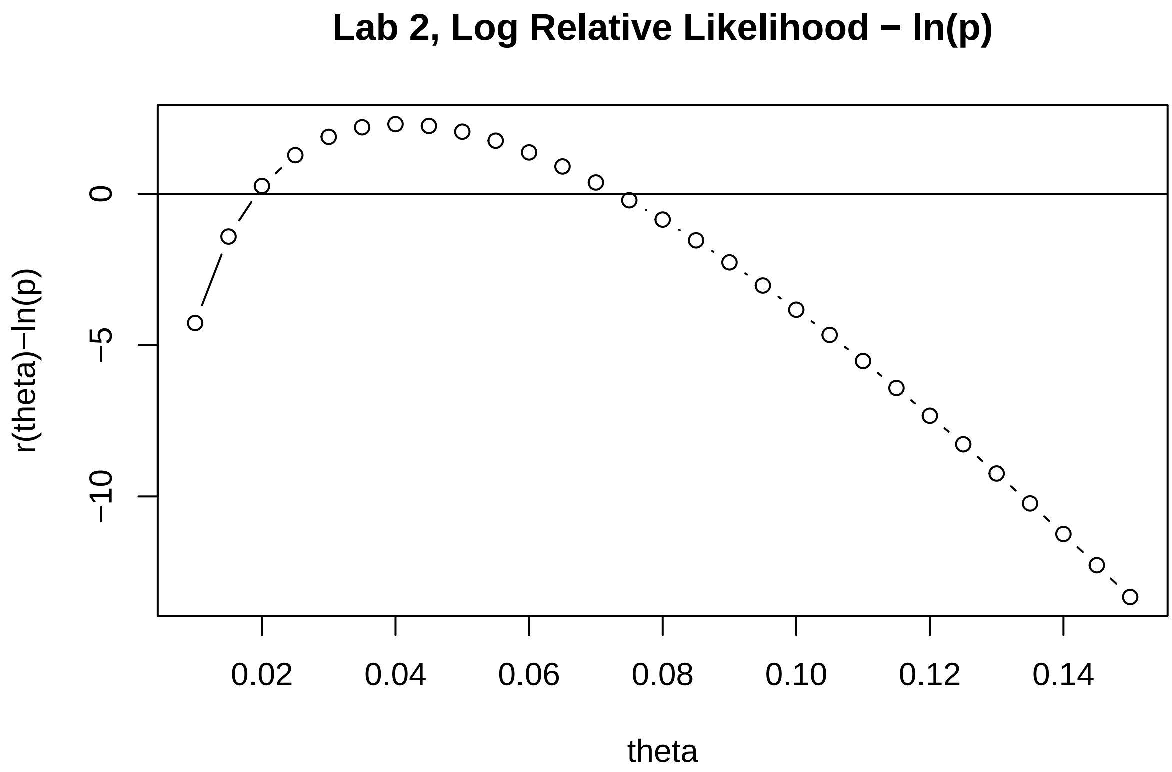
```
## [1] -41.86282
```

### 5. Write a function to compute the log relative likelihood, r(theta) and graph.

```r
# Function to compute the log relative likelihood, r(theta)
#  theta = scalar or vector of Binomial probabilities
#  thetahat = the MLE of theta
#  x = vector of observations

logR <- function(theta, thetahat, x){
  ell(theta, x) - ell(thetahat, x)
}

logR(theta, thetahat$maximum, geo.dat)
```

```
##  [1] -6.570816e+00 -3.716168e+00 -2.045456e+00 -1.026300e+00 -4.215957e-01
##  [6] -1.048979e-01 -7.555313e-04 -6.052590e-02 -2.510181e-01 -5.485788e-01
## [11] -9.357624e-01 -1.399338e+00 -1.929039e+00 -2.516742e+00 -3.155912e+00
## [16] -3.841222e+00 -4.568272e+00 -5.333388e+00 -6.133478e+00 -6.965914e+00
## [21] -7.828448e+00 -8.719145e+00 -9.636329e+00 -1.057854e+01 -1.154450e+01
## [26] -1.253310e+01 -1.354334e+01 -1.457435e+01 -1.562537e+01
```

```r
# Function to compute the log relative likelihood - ln(p) for
#      100p% Likelihood interval computations
#  theta = scalar or vector of Binomial probabilities
#  thetahat = the MLE of theta
#  x = vector of observations
logR.m.lnp <- function(theta, thetahat, x, p) {logR(theta, thetahat, x) - log(p)}

p <- .1  #10% likelihood interval
plot(logR.m.lnp(theta, thetahat$maximum, geo.dat, p) ~ theta, ylab='r(theta)-ln(p)',
     xlab='theta', type='b')
abline(h=0)  #add horizontal line at zero
title('Lab 2, Log Relative Likelihood - ln(p)')
```




### 6. Compute the 10% Likelihood Interval as the roots of r(theta) - ln(.10) = 0

```
#?uniroot  #see the arguments for uniroot
# use the graph to obtain starting interval for root finding search

#Likelihood intervals, supply the function, starting interval and arguments
lower <- uniroot(logR.m.lnp, c(.01, .04), thetahat$maximum, geo.dat, p)
lower
```

```
## $root
## [1] 0.01905715
##
## $f.root
## [1] 0.002134804
##
## $iter
## [1] 5
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

```
upper <- uniroot(logR.m.lnp, c(.06, .10), thetahat$maximum, geo.dat, p)
upper
```

```
## $root
## [1] 0.0732256
##
## $f.root
## [1] 0.0006077995
##
## $iter
## [1] 4
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

## Summary:

The maximum likelihood estimate of theta is, 0.0404715 and its 10% likelihood interval is (0.0190571, 0.0732256).

(Rounded version) The maximum likelihood estimate of theta is, 0.04 and its 10% likelihood interval is (0.019, 0.073).