MATH 4753 Laboratory 10: Maximum likelihood estimates using Grid and N-R

In this lab we will learn how to use R to make maximum likelihood estimates from a sample. A number of functions will be used and should be well understood. While you are not expected to make your own functions I will expect you to understand the lines of code and be capable of adjusting the function to make it do whatever is required.

Some likelihood functions can be easily manipulated so that analytical solutions are available. However in some statistical applications a likelihood might be made from the contingencies of a practical problem that will not easily render an analytical solution. In such cases numerical methods like what we shall perform in this lab will need to be invoked. The likelihood proceeds directly from the joint distribution where instead of being viewed as a function in the data $\boldsymbol{x}$ it is treated as a function in the parameter θ.

$$f(\boldsymbol{x}|\theta) \to L(\theta|\boldsymbol{x}) \to l(\theta|\boldsymbol{x})$$

The process involves differentiating the log likelihood $l(\theta|\boldsymbol{x})$ and finding the maximum. Though this is the analytical process it should be remembered that all we want is the value of θ that maximizes the likelihood. We will therefore be able to use a simpler method by making R evaluate the likelihood with a range of parameter values and then find the parameter value corresponding to the maximum. This method is called Grid approximation.

The following is a function that will make maximum likelihood estimates:

```
mymaxlik=function(lfun,x,param,...){

# how many param values are there?

np=length(param)

# outer -- notice the order, x then param
```

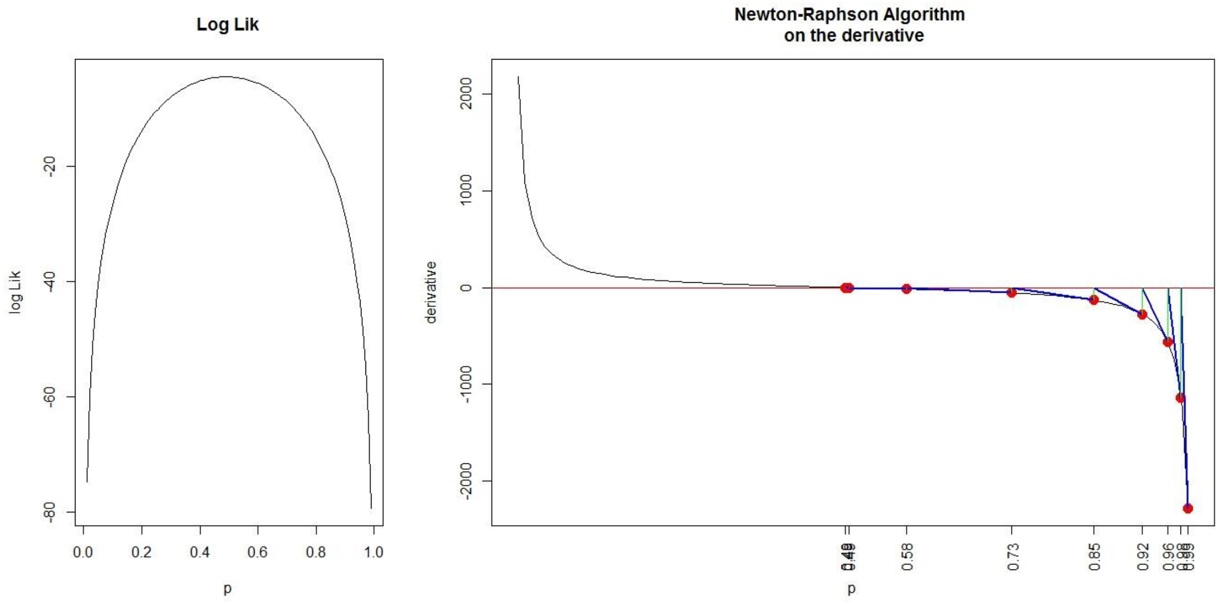
```
     # this produces a matrix - try outer(1:4,5:10,function(x,y)
paste(x,y,sep=" "))   to understand
     z=outer(x,param,lfun) # A
     # z is a matrix where each x,param is replaced with the
function evaluated at those values
     y=apply(z,2,sum)

     # y is a vector made up of the column sums
     # Each y is the log lik for a new parameter value
     plot(param,y,col="Blue",type="l",lwd=2,...)
     # which gives the index for the value of y >= max.
     # there could be a max between two values of the parameter,
therefore 2 indices
     # the first max will take the larger indice
     i=max(which(y==max(y))) # B
     abline(v=param[i],lwd=2,col="Red")

     # plots a nice point where the max lik is
     points(param[i],y[i],pch=19,cex=1.5,col="Black")
     axis(3,param[i],round(param[i],2))
     #check slopes. If it is a max the slope should change sign
from + to
     # We should get three + and two -vs
```

```
    ifelse(i-3>=1 & i+2<=np, slope<-(y[(i-2):(i+2)]-y[(i-
3):(i+1)])/(param[(i-2):(i+2)]-param[(i-3):(i+1)]),slope<-"NA")
    return(list(i=i,parami=param[i],yi=y[i],slope=slope))
}
```

The above code forms the basis of other functions used in this lab.
There is a more sophisticated method available – it uses the Newton Raphson algorithm. Here is a brief description of the method and a function that implements the technique.
First we must identify the essential issue it hand. We need to find where a derivative is zero. This is really just the problem of finding where a function cuts the x axis. We need x such that $f(x) = 0$. The following code gives the recursive algorithm that accomplishes this:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

There is a function that accomplishes this – it is called `myNRML()` – it will produce graphical output similar to the plot below.
The trick to see here is that the function we are attempting to zero is the derivative of the log likelihood. In the case below the derivative is $l'(p)$, that is we need $\hat{p}$ such that $l'(\hat{p}) = 0$, we have theory that handles $f(x) = 0$ so here $l'(p) = f(x)$. The graph on the left is the log lik – you can see that it has a max about p=0.5, and that its slope goes from positive to zero to negative. On the right you can see the derivative $l'(p)$ and it has a zero as calculated with the NR algorithm to be 0.5.



Tasks
Complete the Lab10.R file first and then paste into an RMD document and then knit into an html file.

Note: All plots you are asked to make should be created through RMD

You are expected to adjust the functions as needed to answer the questions within the tasks below.

- Task 1
  - Make a folder LAB10
  - Download the file “lab10.r”
  - Place this file with the others in LAB10.
  - Start Rstudio
  - Open “lab10.r” from within Rstudio.
  - Go to the “session” menu within Rstudio and “set working directory” to where the source files are located.
  - Issue the function `getwd()`.

- Task 2
  - Create your own R file and record the R code you used to complete the lab.
  - In the above code for `mymaxlik()` there are two lines marked A and B. Using any resources available explain what each line does
    - Line A (Use the command in the comments to explain what `outer()` does)
    - Line B
  - Suppose that a series of 8 binomial experiments were performed, with each having 20 trials and each with the same probability of success p. Find the maximum likelihood estimate for p by first answering the following
    - What is the formula (mathematical – no R code) for the likelihood?
    - What is the R code for the above likelihood?
    - Find the maximum likelihood estimate when the following is the data $y = 3,3,4,3,4,5,5,4$ (record the plot).
- Task 3
  - A radioactive source produces protons at an average rate of $\lambda\, protons/sec$. What is the maximum likelihood estimate of λ (using graphical methods) when the data collected from a random sample is $y = 4,6,7,6,5$ protons per second?
  - Show the graphical solution using `mymaxlik()`
  - What is the algebraic expression for the log likelihood?
  - You may need $p(y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$
  - Now use myNRML() to find the maximum likelihood solution, record the following
    - The graphical output
    - The command line output
    - What is the value of $\hat{\lambda}$ as given by the function `myNRML()`
- Task 4
  - Suppose we have two different binomial experiments using the same biased coin. (I.E The probability of a head is the same)
  - For the first experiment the number of trials was 6 and the number of heads 2, for the second experiment the number of trials was 10 and the number of heads 4.
  - Suppose that $p = probability\ of\ a\ head$. Use `mymaxlikg()` to find the graphical max. lik solution for p.

- Task 5
  - Suppose an experiment results in a joint density that is the product of a poisson and a binomial.
  - That is $p(y_1, y_2|\theta_1\theta_2) = bin(y_1|\theta_1)poiss(y_2|\theta_2)$
  - Write down the algebraic expression for $l(\theta_1, \theta_2)$ the log likelihood.
  - Suppose the poisson process has $y_2 = 4$ and the binomial process $n = 20, y_1 = 4$
  - Use `maxlikg2()` to give the graphical max lik solutions for θ_1 and θ_2.
- Task 6
  - Suppose that a normal experiment is performed and scientists are interested in finding maximum likelihood estimates for the population mean and standard deviation.
  - Use the function `mymlnorm()` to produce a graphical solution for $\hat{\mu}$ and $\hat{\sigma}$ when the data are $y = 10{,}12{,}13{,}15{,}12{,}11{,}10$ assuming that $Y_i \sim N(\mu, \sigma)$.
- Task 7
  - Suppose that another experiment is performed this time data is generated from a beta distribution. The data is generated as below.
  - `sam= rbeta(30,shape1=3,shape2=4)# we know the pop params`
  - Using any code available resample from the sample (sam) and create max lik estimates for shape1 and shape2. Hint: You might need the function `mymlbeta()`
  - Make 12 plots on the same graphical device.
  - Plot the distances between the estimates and the known population values.
- Task 8
  - Add one of the above functions to your package. Make sure it is well documented
  - In an R chunk within your RMD document do the following:
  - Use ILAS2019::mynewfunction()
  - Call the function with data and create a plot (catch all command line output in an object so as not to populate your document with numbers.
  - Knit into html

################### LAB FINISHES HERE ###############################

- Task 9: Extra for experts!
  - Make a function that will find max lik estimates for a gamma distribution.