Math 575 HW 1

Fall 2021

Washington University in St. Louis, University College

Due date: 9/26/2021

Instruction:

Please type your answers clearly and show your work neatly. You are encouraged to use the Rmarkdown version of this assignment as a template to submit your work. Unless stated otherwise, all programming references in the assignment will be in R. For this assignment, problems roughly covers content from the first 3 lectures, and chapters 1 and 2 of the text by Givens and Hoeting.

Problem 1

Let $X_1, X_2, ..., X_n$ be i.i.d. exponentially distributed random variables. That is, each X_i has probability density function

$$f_X(x) = \lambda e^{-\lambda x}, x > 0.$$

a. Derive the log-likelihood function for λ analytically.

b. Check that the MLE you got from a) is indeed a maximum using the second-derivative test

c. The data `claims.csv` is a sample of observations that measured the inter-arrival times between insurance claims from a fictional company. Consider each of the observations as a realization from an i.i.d. exponentially distributed random variable with unknown rate, λ. Let $\hat{\lambda}$ be the MLE of λ. Calculate the value of $\hat{\lambda}$ using the data.

d. Use the R command `rexp()` to generate a set of exponentially distributed random variables with the $\hat{\lambda}$ you found from c, compare the quantile of the two datasets using `qqplot()`. Comment on the results.

Problem 2 (2.2. in book)

Consider the density

$$f(x) = \frac{1 - \cos(x - \theta)}{2\pi}, 0 \leq x \leq 2\pi,$$

where θ is a parameter between $-\pi$ and π. The following i.i.d. data arise from this density: 3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96, 2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52, 2.50. We wish to estimate θ.

a. Graph the log-likelihood function of θ between $-\pi$ and π.

b. Find the method-of-moments estimator of θ.

c. Find the MLE for θ using the Newton-Raphson method, using the result from b) as the starting value. What solutions do you find when you start at -2.7 and 2.7?

d. Repeat part (c) using 200 equally spaced starting values between $-\pi$ and π. Partition the interval between $-\pi$ and π into sets of attraction. In other words, divid the set of starting values into separate groups, with each group corresponding to a separate uniquee outcome of the optimization (a local mode). Discuss your results.

Table 1: Beetle count

Days	0	8	28	41	63	79	97	117	135	154
Beetles	2	47	192	256	768	896	1120	896	1184	1024

Problem 3 (2.6 in book)

Table 1 below provides counts of a flour beetle (Tribolium confusum) population at various points in time. Beetles in all stages of development were counted, and the food supply was carefully controlled. An elementary model for population growth is the logistic model given by

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

,

where N is population size, t is time, r is a growth rate parameter, and K is a parameter that represents the population carrying capacity of the environment. The solution to this differential equation is given by

$$N_t = f(t) = \frac{kN_0}{N_0 + (K - N_0)\exp\{-rt\}}$$

where N_t denotes the population size at time t.

a. Fit the logistic growth model to the flour beetle data using the Gauss–Newton approach to minimize the sum of squared errors between model predictions and observed counts.

b. Fit the logistic growth model to the flour beetle data using the Newton–Raphson approach to minimize the sum of squared errors between model predictions and observed counts.

Solution

First, let's establish some notation. Let

- y_i = beetle count at time t_i
- $\theta^{(t+1)} = (k, r)$
- $N(t) = f(t_i, \theta) = f$

Our objective is to solve θ by minimizing the sum of squared errors between model predictions and observed counts. In other words, we need to find $\hat{\theta}$ such that

$$\hat{\theta} = \arg\max g(\theta)$$

where

$$g(\theta) = -\sum_{i=1}^{n}[(y_i - f(t_i, \theta)^2)]$$

a. Using the Gauss-Newton approach, at each iteration i, we have:

$$\theta^{(i+1)} = \theta^{(i)} + (A^{(i)}(A^{(i)})^T)^{-1}(A^{(i)})^T(X^{(i)})$$

where

- $\theta^{(t+1)} = (k, r)$
- $A^{(i)} = [\frac{\partial}{\partial k}f, \frac{\partial}{\partial r}f]$

b. Using the Newton-Rapson approach, at each iteration i, we have:

$$\theta^{(i+1)} = \theta^{(i)} - (g\prime\prime(\theta))^{-1}g\prime(\theta)$$

where

$$g\prime\prime(\theta) = \begin{bmatrix} \frac{\partial^2}{\partial k^2} g & \frac{\partial^2}{\partial k \partial r} g \\ \frac{\partial^2}{\partial r \partial k} g & \frac{\partial^2}{\partial r^2} g \end{bmatrix}$$

Let $\gamma(\theta) = (\mathbf{y} - f) \in \mathbb{R}^{n \times 1}$. That means, $\gamma(\theta)$ is an $n \times 1$ vector whose i^{th} element is

$$\gamma_i = y_i - f(t_i;\theta) = y_i - \frac{kN_0}{N_0 + (K - N_0)\exp\{-rt_i\}}$$

we can express $g\prime(\theta)$ and $g\prime\prime(\theta)$ as :

$$g\prime(\theta) = \begin{bmatrix} \gamma(\theta)^T \frac{\partial}{\partial k} f \\ \gamma(\theta)^T \frac{\partial}{\partial r} f \end{bmatrix}$$

and

$$g\prime\prime(\theta) = 2 \begin{bmatrix} \gamma(\theta)^T \frac{\partial^2}{\partial k^2} f - (\frac{\partial}{\partial k} f)^2 & \gamma(\theta)^T \frac{\partial^2}{\partial k \partial r} f - (\frac{\partial}{\partial k} f)(\frac{\partial}{\partial r} f) \\ \gamma(\theta)^T \frac{\partial^2}{\partial r \partial k} f - (\frac{\partial}{\partial r} f)(\frac{\partial}{\partial k} f) & \gamma(\theta)^T \frac{\partial^2}{\partial r^2} f - (\frac{\partial}{\partial r} f)^2 \end{bmatrix}$$

The solution from both algorithm should yield $\hat{\theta} = (1033.5, 0.118)$ after only 5 iterations

Below are the functions implemented :

```r
## define N(t)
N.func <-function(t,x){
  N0 <-2
  out<-(N0*x[1])/(N0+(x[1]-N0)*exp(-x[2]*t))
  return(out)
}

## FIRST DERIVATIVES
N.prime <-function(t,x){
  ## 1 = k, 2 = r
  N0 <-2
  base <- N0+(x[1]-N0)*exp(-x[2]*t)
  out.1<- N0^2*(1-exp(-x[2]*t))/(base^2)
  out.2<- (x[1]*(x[1]-N0)*N0*t*exp(-x[2]*t))/(base^2)
  out = matrix(c(out.1,out.2),nrow=2)
  return(out)
}

## SECOND DERIVATIVES:
N.prime2 <-function(t,x){
  N0 <-2
  base <- N0 + (x[1]-N0)*exp(-x[2]*t)
  out.kk<- (-2)*(N0^2)*exp(-x[2]*t)*(1-exp(-x[2]*t))/(base^3)
  out.kr <- (N0^2*exp(-x[2]*t)*t)*(exp(-x[2]*t)*(N0-x[1])+2*x[1]-N0)/(base^3)
  out.rr<- (N0^2*exp(-x[2]*t)*t)*(N0*exp(-x[2]*t)-x[1]*exp(-x[2]*t)+2*x[1]-N0)/(base^3)
  out = matrix(c(out.kk,out.kr,out.kr,out.rr),nrow=4)
  return(out)
}

N.func <- Vectorize(N.func,'t')
N.prime <- Vectorize(N.prime,'t')
N.prime2 <- Vectorize(N.prime2,'t')
```

Define $\gamma(\theta)$ to be the difference between y and $f(t;\theta)(= N_t)$. Notice that the function `gamma.x` is an n-by-1 vector measuring the difference between observed beetle count and the model. It is a function of x (theta) as the data is already embedded.

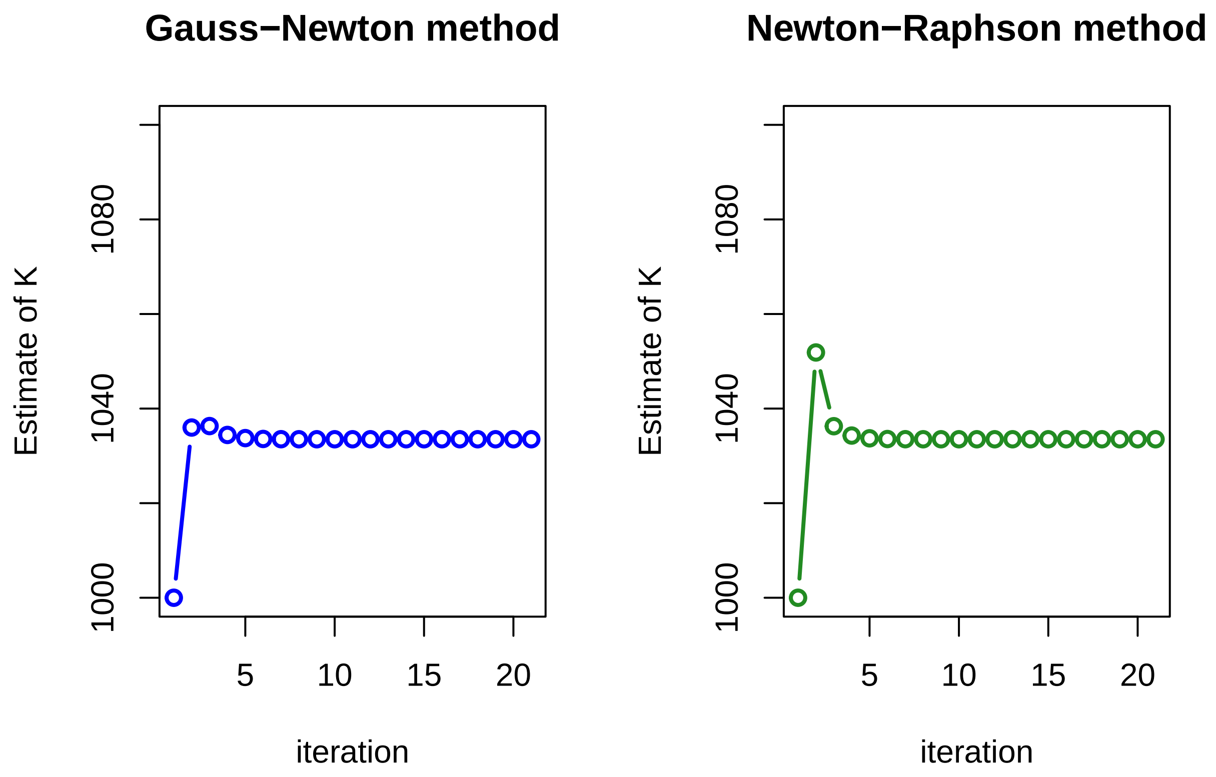
```r
gamma <- function(x,t,y){

  out <- y - N.func(t,x)

  return(out)
}

gamma.x <- function(x){
  t = c(0,8,28,41,63,79,97,117,135,154)
  y=c(2,47,192,256,768,896,1120,896,1184,1024)
  out <- gamma(x,t,y)
  return(out)
}
```

We then define the derivatives $g\prime$ and $g\prime\prime$

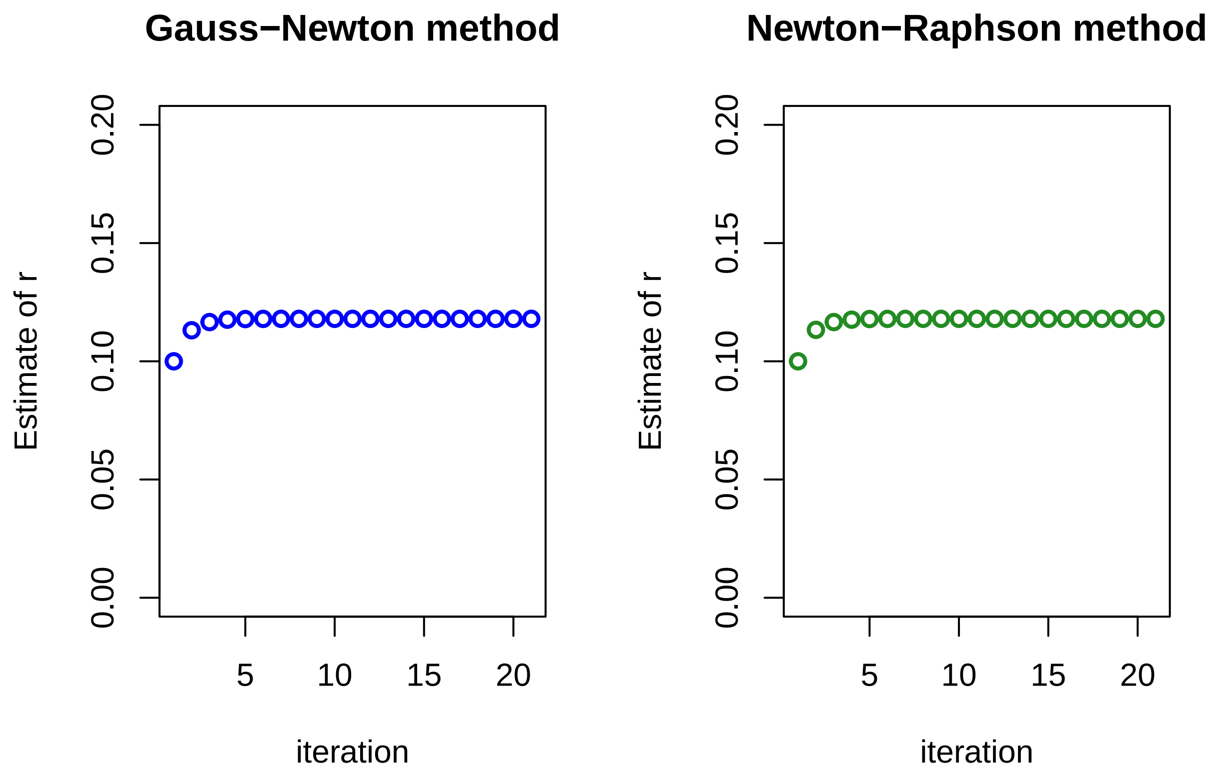
```r
N.prime.x <-function(x){
  t = c(0,8,28,41,63,79,97,117,135,154)
  out<-N.prime(t,x)
  return(out)
}


N.prime2.x <-function(x){
  t = c(0,8,28,41,63,79,97,117,135,154)
  out<-N.prime2(t,x)
  return(out)
}


g.prime <-function(x){

  g.prime.1 <-2*gamma.x(x)%*%N.prime.x(x)[1,]
  g.prime.2 <- 2*gamma.x(x)%*%N.prime.x(x)[2,]
  out <- matrix(c(g.prime.1,g.prime.2),nrow=2,byrow = TRUE)
  return(out)

}



g.prime.2 <-function(x){

  g.kk <-2*(gamma.x(x)%*%N.prime2.x(x)[1,] - (N.prime.x(x)[1,])%*%(N.prime.x(x)[1,]))
  g.kr <-2*((gamma.x(x)%*%N.prime2.x(x)[2,]) - (N.prime.x(x)[1,])%*%(N.prime.x(x)[2,]))
  g.rk <-g.kr
  g.rr <-2*((gamma.x(x)%*%N.prime2.x(x)[2,]) - (N.prime.x(x)[2,])%*%(N.prime.x(x)[2,]))
  out <- matrix(c(g.kk,g.kr,g.rk,g.rr),nrow=2, byrow=TRUE)
  return(out)

}
```

Below is the code to implement both part 3a and 3b. We start by setting some initial condition, then using a for-loop to update each θ using the Gauss-Newton and Newton-Raphson approach respectively

```r
### PROBLEM 3.a




k0 = 1000
r0 = 0.1
x0 <-c(k0,r0)


iter <-20

x.vals.nr <-matrix(0,nrow = 2,ncol = iter+1)
x.vals.gn <-matrix(0,nrow = 2,ncol = iter+1)

x.vals.nr[1,1]<-x0[1]
x.vals.nr[2,1]<-x0[2]


x.vals.gn[1,1]<-x0[1]
x.vals.gn[2,1]<-x0[2]

## Algorithm starts:

for (i in 1:iter){
  ## Newton-Raphson:
  x.vals.nr[,i+1]<-x.vals.nr[,i]-solve(g.prime.2(x.vals.nr[,i]))%*%g.prime(x.vals.nr[,i])

  ## Gauss-Newton:
  A_i <-N.prime.x(x.vals.nr[,i])
  x.vals.gn[,i+1]<-x.vals.gn[,i]+solve(A_i%*%t(A_i))%*%A_i%*%(gamma.x(x.vals.gn[,i]))

}

par(mfrow = c(1,2))

plot(1:(iter+1), x.vals.gn[1,],type = 'b',col = 'blue',
     lwd = 2, xlim = c(1,iter+1),ylim = c(1000,1100),
     ylab = 'Estimate of K', xlab = 'iteration',main ='Gauss-Newton method')

plot(1:(iter+1), x.vals.nr[1,],type = 'b',col='forestgreen',
     lwd = 2, xlim = c(1,iter+1),ylim = c(1000,1100),
     ylab = 'Estimate of K', xlab = 'iteration', main ='Newton-Raphson method')
```

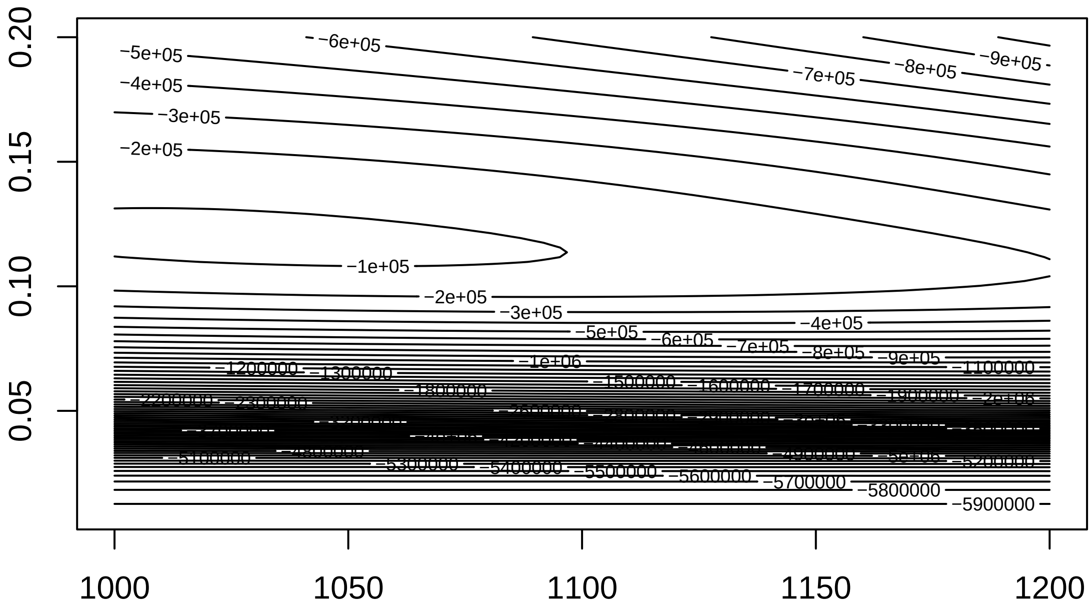


```
par(mfrow = c(1,2))

plot(1:(iter+1), x.vals.gn[2,],type = 'b',col = 'blue',
     lwd = 2, xlim = c(1,iter+1),ylim = c(0,0.2),
     ylab = 'Estimate of r', xlab = 'iteration',main ='Gauss-Newton method')

plot(1:(iter+1), x.vals.nr[2,],type = 'b',col='forestgreen',
     lwd = 2, xlim = c(1,iter+1),ylim = c(0,0.2),
     ylab = 'Estimate of r', xlab = 'iteration', main ='Newton-Raphson method')
```



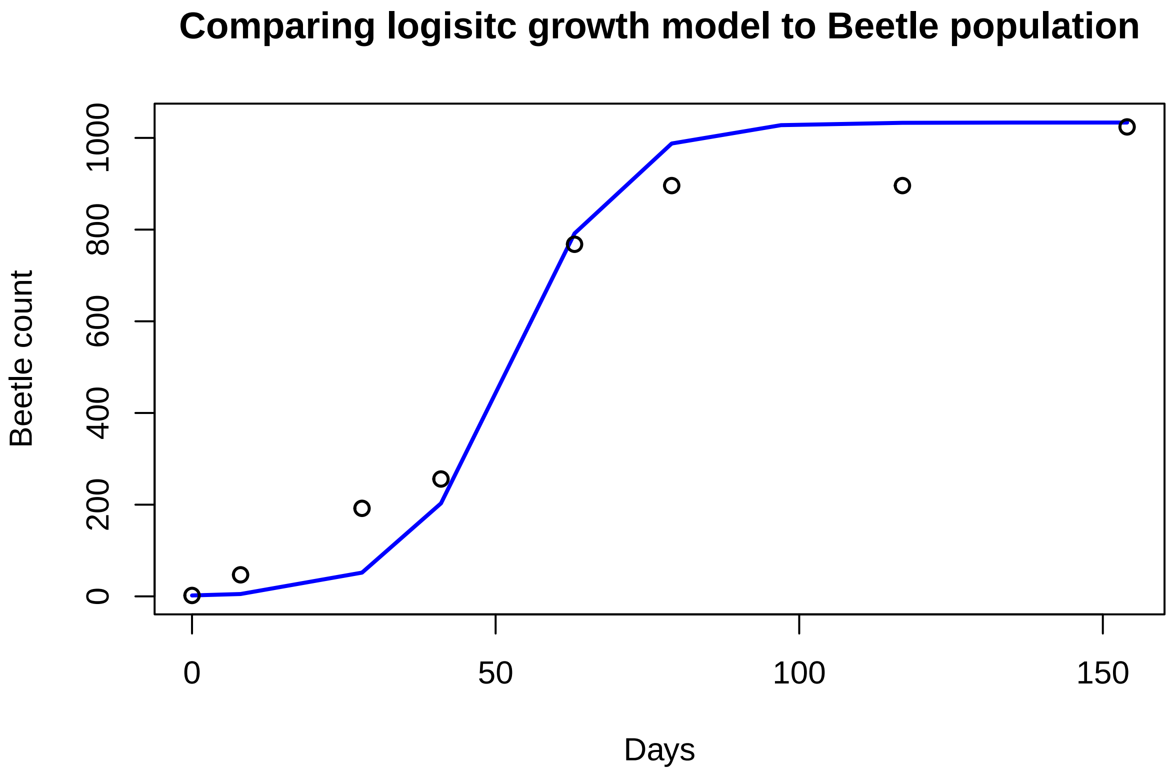
We can also take a look at the contour plot of $g(\theta)$, where the $x - y$ plane are different values of (k, r) and the labels indicate the corresponding values of g. The maximum of g is achieved at around -83271.005.

```
par(mfrow=c(1,1))

## contour plot:
x1 = seq(1000,1200,length=100)
x2 = seq(0.01,0.2,length=100)

g <-function(x){

  return(-1*gamma.x(x)%*%gamma.x(x))

}
z<-matrix(0,ncol = 100,nrow = 100)
for(i in 1:100){
  for(j in 1:100){
    z[i,j] = g(c(x1[i],x2[j]))
  }
}

contour(x1,x2,z,nlevels=50)
```



Let's take a look at the end result:

```
t <- c(0,8,28,41,63,79,97,117,135,154)
y <- c(2,47,192,256,768,896,1120,896,1184,1024)
k <-x.vals.gn[1,iter+1]
r <-x.vals.gn[2,iter+1]

plot(t, N.func(t,x=c(k,r)),type = 'l',  col = 'blue',lwd =2,
     ylab = 'Beetle count', xlab = 'Days',
     main = 'Comparing logisitc growth model to Beetle population')
points(t, y,lwd = 1.5)
```


Comparing logisitc growth model to Beetle population
Beetle count
Days