Homework2_HZ

Question 1

Bisection Method

```
bisection <- function(f, a, b, nMax, tol)
{
  #initiate the a and b value, assume intervals will be proper
  iteration <- 0
  #check bounds
  if(f(a) == 0.0){
    return(a)
  }
  if(f(b) == 0.0){
    return(b)
  }
  # Begin method's loop
  for (i in 1:nMax){
    c <- (a + b)/2 #Calc the midpoint
    if(f(c) != 0) {
      #TRUE: f(c) > tol AND i <=NMAX
      if((abs(f(c)) > tol)) {
        if(sign(f(c)) == sign(f(a))) {
          a <- c
          b <- b
        }
        else {
          a <- a
          b <- c
        }
        c <- (a + b)/2
        iteration = iteration + 1
      }
      else {
        #the f(c) is within the range of tolerance
        break
      }
    }
    else {
      #FALSE: f(c) is a root
      break
    }
  }
  return(list("it" = iteration, "root" = c))
}
```

```r
fcn <- function(x){sqrt(x)-cos(x)}
bmMe <- bisection(fcn, 0, 2, 3, 1e-7)
bmMe$root
```

```
## [1] 0.625
```

Newton-Raphson Method

```r
newton <- function(f, dx, a, b, inital, nMax, tol){
  #set initial value
  x0 <- inital
  rootArray <- nMax

  #Check bounds
  if(f(a) == 0.0){
    return(a)
  }
  if(f(b) == 0.0){
    return(b)
  }

  #begin loop for loop method
  for (i in 1:nMax) {
    x1 = x0 - (f(x0)/dx(x0))
    rootArray[i] <- x1
    if (abs(x1 - x0) <= tol){
      root <- tail(rootArray,n = 1)
      result <- list('root' = root, 'iterations' = rootArray)
      return(result)
    }
    x0 <- x1
  }
}

f <- function(x){sqrt(x)-cos(x)}
dx <- function(x){0.5*(x^(-0.5)) + sin(x)}
newMe <- newton(f, dx, 0, 2, 1, 3, 1e-3)
newMe$root
```

```
## [1] 0.6417144
```

The Newton-Raphuson Method finds the root within the three iterations, compared to the Bisection Method. The Bisection Method found 0.625, which is not even within the tolerance.

Question 2

Part a: Deriving the Newton-Raphson Method

In the problem, we are told we can use the Poisson process assumption so we can have the likelihood function:

$$L(N|\lambda_i) = \sum_{i=1}^{n} \frac{\lambda^{N_i} e^{-\lambda}}{N!} \tag{1}$$

and because $\lambda_i = \alpha_1 b_{i1} + \alpha_2 b_{i2}$ we can substitute λ_i into Eq (1):

$$L(N|\alpha_1, \alpha_2) = \sum_{i=1}^{n} \frac{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^{N_i} e^{-(\alpha_1 b_{i1} + \alpha_2 b_{i2})}}{N!} \tag{2}$$

In order to find the parameters α_1 and α_2 we can use the Newton-Raphson update which needs to become Eq (3):

$$\begin{bmatrix} \alpha_1(t+1) \\ \alpha_2(t+1) \end{bmatrix} = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \end{bmatrix} - \frac{L\prime}{L\prime\prime} \tag{3}$$

In order to get Eq (3), we first need to get the log likelihood of Eq (2):

$$l(N|\alpha_1, \alpha_2) = \sum_{i=1}^{n} N_i \ln(\alpha_1 b_{i1} + \alpha_2 b_{i2}) - \sum_{i=1}^{n} \alpha_1 b_{i1} + \alpha_2 b_{i2} - \sum_{i=1}^{n} \ln(N!) \tag{4}$$

We can then use Eq (4) and take the partial first derivative in regard to both parameters α_1 and α_2:

$$l\prime(N|\alpha_1, \alpha_2) = \begin{bmatrix} \sum_{i=1}^{n} \frac{N_i b_{i1}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i1} \\ \sum_{i=1}^{n} \frac{N_i b_{i2}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i2} \end{bmatrix} \tag{5}$$

To get the double derivative we use the Hessian matrix

$$l\prime\prime(N|\alpha_1, \alpha_2) = \begin{bmatrix} \frac{\partial^2 l}{\partial \alpha_1^2} & \frac{\partial^2 l}{\partial \alpha_1 \partial \alpha_2} \\ \frac{\partial^2 l}{\partial \alpha_2^2} & \frac{\partial^2 l}{\partial \alpha_2 \partial \alpha_1} \end{bmatrix} \tag{6}$$

$$l\prime\prime(N|\alpha_1, \alpha_2) = \begin{bmatrix} \sum_{i=1}^{n} -\frac{N_i b_{i1}^2}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} & \sum_{i=1}^{n} -\frac{N_i b_{i1} b_{i2}}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} \\ \sum_{i=1}^{n} -\frac{N_i b_{i1} b_{i2}}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} & \sum_{i=1}^{n} -\frac{N_i b_{i2}^2}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} \end{bmatrix} \tag{7}$$

with this we can get the equation to be used in the Newton-Raphson update to get a final equation

$$\begin{bmatrix} \alpha_1(t+1) \\ \alpha_2(t+1) \end{bmatrix} = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^{n} -\frac{N_i b_{i1}^2}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} & \sum_{i=1}^{n} -\frac{N_i b_{i1} b_{i2}}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} \\ \sum_{i=1}^{n} -\frac{N_i b_{i1} b_{i2}}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} & \sum_{i=1}^{n} -\frac{N_i b_{i2}^2}{(\alpha_1 b_{i1} + \alpha_2 b_{i2})^2} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} \frac{N_i b_{i1}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i1} \\ \sum_{i=1}^{n} \frac{N_i b_{i2}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i2} \end{bmatrix} \tag{8}$$

Part b: Deriving the Fisher Scoring Method

Similar to part (a), we will use the log likelihood to find the equation to find the parameters α_1 and α_2. Instead of only taking the second derivative, we will take the variance of the first derivative to get the Fisher Information. This output will be a two by two matrix as well but instead is the Covariance matrix instead of the Hessian matrix:

$$\begin{bmatrix} Var(\sum_{i=1}^{n} \frac{N_i b_{i1}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i1}) & Covariance \\ Covariance & Var(\sum_{i=1}^{n} \frac{N_i b_{i2}}{\alpha_1 b_{i1} + \alpha_2 b_{i2}} - \sum_{i=1}^{n} b_{i2}) \end{bmatrix} \tag{9}$$

Or we could take the expectation of the second derivative of the log likelihood function.

Part c: Implementing Newton and Fisher Methods in R

Part d: Standard Error

Part e: Quasi-newton Method

Question 3

Question 4