Hwk#7

Euchie Jn Pierre

2022-11-13

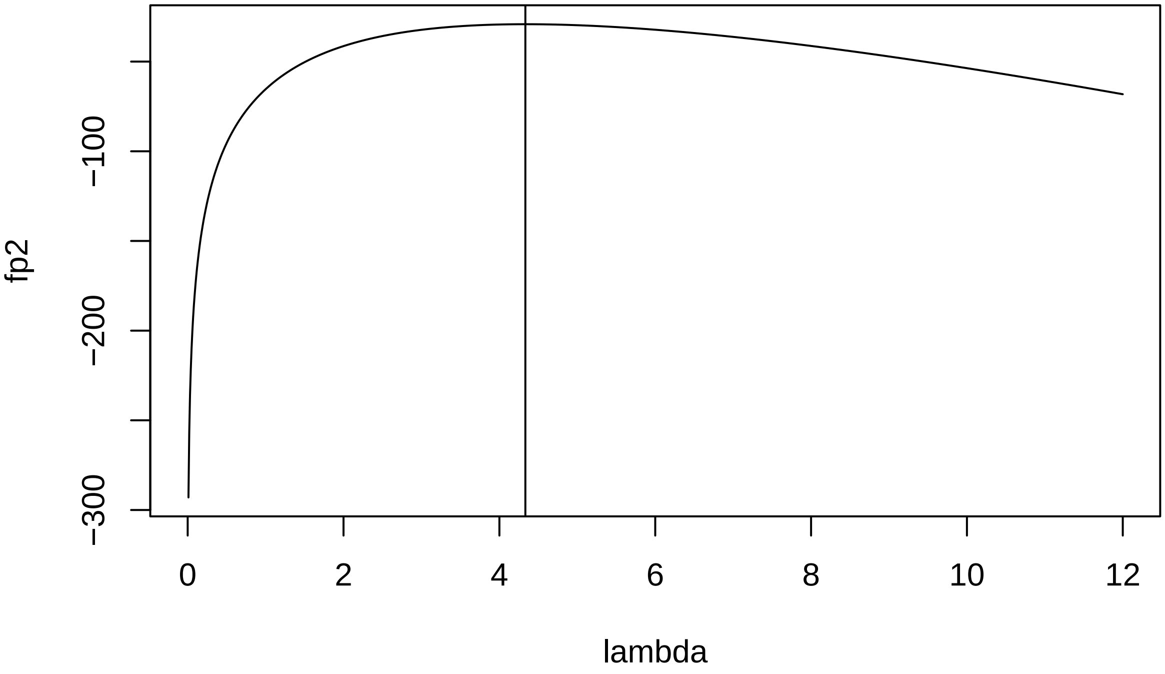
This homework is to practice constructing a log-likelihood function.

NewtonRaphson function

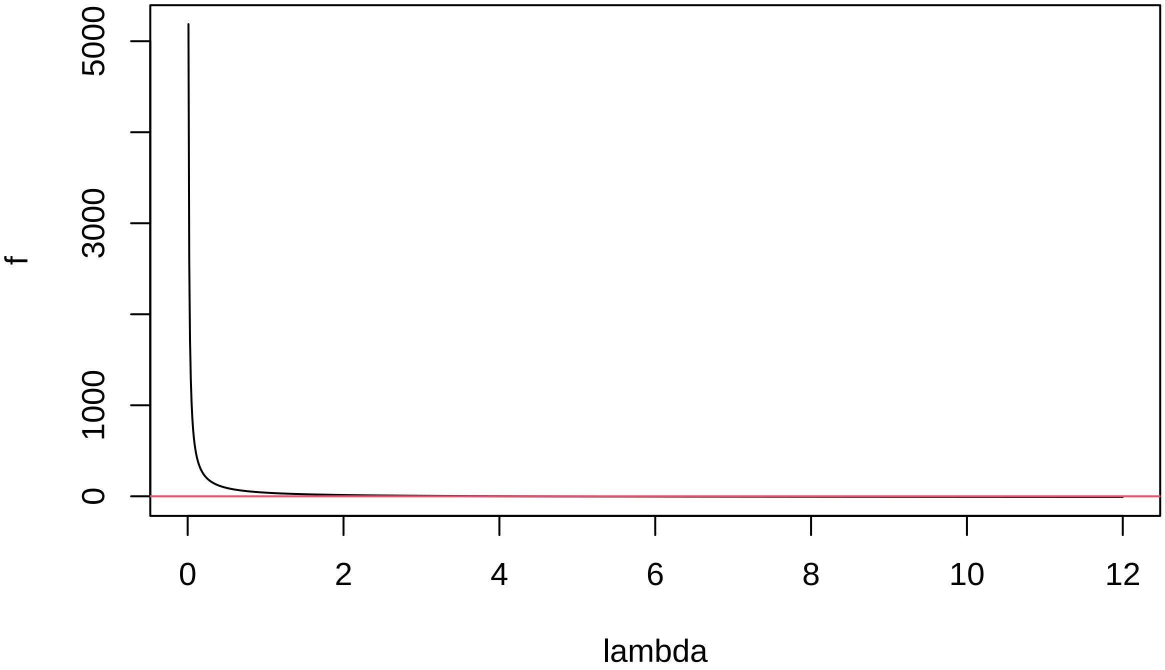
```r
#newtonraphson function
newtonraphson <- function(ftn, x0, tol = 1e-9, max.iter = 100) {
  x <- x0 # x0: the initial value
  fx <- ftn(x)
  iter <- 0
  while ((abs(fx[1]) > tol) & (iter < max.iter)) {
    x <- x - fx[1]/fx[2]
    fx <- ftn(x)
    iter <- iter + 1
    cat("At iteration", iter, "value of x is:", x, "\n")
  }
  if (abs(fx[1]) > tol) {
    cat("Algorithm failed to converge\n")
    return(NULL)
  } else { # abs(fx[1]) <= tol
    cat("Algorithm converged\n")
    return(x)
  }
}
```

Ex 18_1 (log-likelihood function of mean number and maximum likelihood estimate)

```r
#plotting log likelihood function for mean number of accidents(lambda) in one month.
loglike <- function(x, lambda){
  loglikelihood <- 0
  for (i in 1:length(x)){
      loglikelihood <- loglikelihood + log(exp(-lambda)*(lambda^x[i])/factorial(x[i]))
    }
  return(loglikelihood)
}
x <- c(5, 5, 6, 0, 1, 2, 4, 4, 3, 5, 7, 10)
lambda <- seq(0,12,0.01)
fp2 <- loglike(x,lambda)
plot(lambda,fp2,col=1, type="l")
lambdahat <- sum(x)/length(x)
abline(v=lambdahat)
```



```
# solving the maximum likelihood estimate using the newton-raphson method
f <- (-length(x))+sum(x)/lambda
plot(lambda,f, type="l") # plotting graph to find initial value
abline(h=0, col=2)
```



```
ftn <- function(lambda) {
  f <- (-length(x))+sum(x)/lambda
  df <- -(lambda^(-2))*sum(x)
  return(c(f,df))
}
newtonraphson(ftn, 2, 1e-6)
```

```
## At iteration 1 value of x is: 3.076923
## At iteration 2 value of x is: 3.969049
## At iteration 3 value of x is: 4.302709
## At iteration 4 value of x is: 4.333117
## At iteration 5 value of x is: 4.333333
## Algorithm converged
```

```
## [1] 4.333333
```

Ex S5_1 (calculating the log-likelihood of a model)

```
bmi <- read.csv("Data/BMIrepeated.csv")
head(bmi)
```

```
##    ID SEX AGE BMI0 BMI1 BMI2 BMI3 Treatment
## 1 ID1   F  35   25   24   25   22         0
## 2 ID2   M  32   23   24   23   22         0
## 3 ID3   F  30   22   21   21   19         0
## 4 ID4   F  31   20   21   20   22         0
## 5 ID5   F  52   21   22   20   20         0
## 6 ID6   F  59   21   22   21   20         0
```

```
#constructing design matrix
b <-cbind(rep(1, length(bmi$BMI3)),as.numeric(as.factor(bmi$SEX)), bmi$AGE, bmi$Treatment)
head(b)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1   35    0
## [2,]    1    2   32    0
## [3,]    1    1   30    0
## [4,]    1    1   31    0
## [5,]    1    1   52    0
## [6,]    1    1   59    0
```

```
#calculating regression coefficients
beta <- solve(t(b)%*%b)%*%t(b)%*%matrix(bmi$BMI3,ncol = 1)
beta
```

```
##            [,1]
## [1,] 15.2696323
## [2,]  0.8314382
## [3,]  0.2516927
## [4,] -3.4052445
```

```
#Calculating residuals
res<-bmi$BMI3-b%*%beta
head(res)
```

```
##            [,1]
## [1,]  -2.910316
## [2,]  -2.986676
## [3,]  -4.651852
## [4,]  -1.903545
## [5,]  -9.189092
## [6,] -10.950942
```

```
#estimating sigma2
sigma2 <- sum(res^2)/length(bmi$BMI3)
-(length(bmi$BMI3)/2)*(log(2*pi)+log(sigma2)+1)
```

```
## [1] -337.3221
```

```
#Constructing a model to compare our results
model1 <- lm(BMI3~SEX+AGE+Treatment, data = bmi)
logLik(model1)
```

```
## 'log Lik.' -337.3221 (df=5)
```