HW3 - Exercise3

JooChul Lee

28 September 2018

Abstract

This is the .

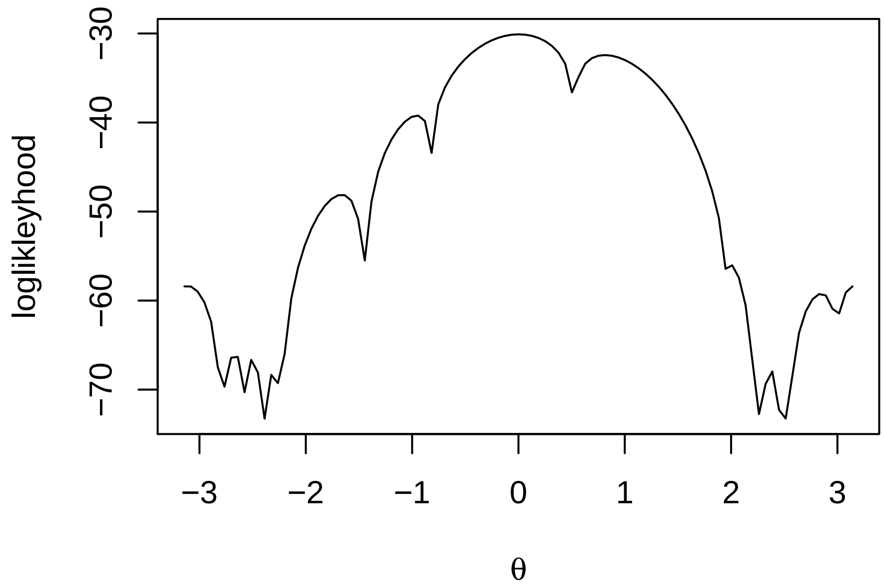
Contents

1 Many local maxima

1.1 Find the the log-likelihood function and plot it.

log-likelihood function of θ :

$$l(\theta) = -n \ln 2\pi + \sum_{i=1}^{n} \ln[1 - \cos(x_i - \theta)]$$



1.2 Find the method-of-moments estimator

$$
\begin{aligned}
E(X) &= \int_0^{2\pi} \frac{1-cos(x-\theta)}{2\pi} x dx && (1)\\
&= \int_0^{2\pi} \frac{x}{2\pi} dx - \int_0^{2\pi} xcos(x-\theta) dx && (2)\\
&= \frac{1}{2\pi}(2\pi^2 + 2\pi sin\theta) && (3)\\
&= \pi - \frac{1}{2\pi}(-2\pi sin(\theta)) && (4)\\
&= \pi + sin(\theta) && (5)
\end{aligned}
$$

Thus, we can get

$$\hat{\theta} = sin^{-1}(\bar{X} - \pi)$$

Thus,

$$\hat{\theta} = 0.0953 \; or \; 3.046$$

1.3 Find the MLE using the Newton–Raphson method

This is the code for table 1.

```
sample <- c(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)
F_S_D = function(theta)
{
   First = sum( -sin(sample - theta)/(1-cos(sample - theta)) )
   Second = sum( 1/(cos(sample - theta) - 1 ) )
   list(First = First, Second= Second )
```

```
}

N_R = function(initial, max = 100, tol = 1e-5)
{
   current = initial
   for(i in 1:max)
   {
      new = current - F_S_D(current)$First/F_S_D(current)$Second
      if(abs(new -current) < tol) break
      current1 = current
      current = new
   }
   return( c(current, i, diff = abs(current -current1) ) )
}
N_R(0.0953941)[1]
```

```
##
## 0.003118157
```

```
N_R(3.046199)[1]
```

```
##
## 3.170713
```

When the initial value is 0.0953941, the mle is 0.003118157. When the initial value is 3.046199, the mle is 3.170713.

1.4 What solutions do you find when you start at $\theta_0 = -2.7$ and $\theta_0 = 2.7$

```
sample <- c(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)
F_S_D = function(theta)
{
   First = sum( -sin(sample - theta)/(1-cos(sample - theta)) )
   Second = sum( 1/(cos(sample - theta) - 1 ) )
   list(First = First, Second= Second )
}

N_R = function(initial, max = 100, tol = 1e-5)
{
   current = initial
   for(i in 1:max)
   {
      new = current - F_S_D(current)$First/F_S_D(current)$Second
      if(abs(new -current) < tol) break
      current1 = current
      current = new
```

```
  }
  return( c(current, i, diff = abs(current -current1) ) )
}
N_R(-2.7)[1]
```

```
##
## -2.668857
```

```
N_R(2.7)[1]
```

```
##
## 2.848423
```

When the initial value is -2.7, the mle is -2.668857 When the initial value is 2.7, the mle is 2.848423

1.5 Repeat the above using 200 equally spaced starting values

```
sample <- c(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)
F_S_D = function(theta)
{
  First = sum( -sin(sample - theta)/(1-cos(sample - theta)) )
  Second = sum( 1/(cos(sample - theta) - 1 ) )
  list(First = First, Second= Second )
}

N_R = function(initial, max = 100, tol = 1e-5)
{
  current = initial
  for(i in 1:max)
  {
    new = current - F_S_D(current)$First/F_S_D(current)$Second
    if(abs(new -current) < tol) break
    current1 = current
    current = new
  }
  return( c(current, i, diff = abs(current -current1) ) )
}
initial = seq(-pi, pi, length.out = 200)
result = matrix(0,2, 200)
result[1,] =initial
for(i in 1:200)
  result[2,i] = N_R(initial[i],tol = 1e-6)[1]
rownames(result) <- c("Initial Value", "Root")

knitr::kable(result[,1:8] ,booktabs = TRUE, align='c')
```

Initial Value	-3.141593	-3.110019	-3.078445	-3.046871	-3.015297	-2.983724	-2.952150	-2.920576
Root	-3.112471	-3.112471	-3.112471	-3.112471	-3.112471	-3.112471	-3.112471	-3.112471

```
knitr::kable(result[,9:16] ,booktabs = TRUE, align='c')
```

Initial Value	-2.889002	-2.857429	-2.825855	-2.794281	-2.762707	-2.731133	-2.699559	-2.667986
Root	-3.112471	-3.112471	-3.112471	-2.786557	-2.786557	-2.668858	-2.668858	-2.668858

```
knitr::kable(result[,17:24] ,booktabs = TRUE, align='c')
```

Initial Value	-2.636412	-2.604838	-2.573264	-2.541691	-2.510117	-2.478543	-2.446969	-2.415395
Root	-2.668858	-2.668858	-2.509356	-2.509356	-2.509356	-2.509356	-2.509356	-2.509355

```
knitr::kable(result[,25:32] ,booktabs = TRUE, align='c')
```

Initial Value	-2.383822	-2.352248	-2.320674	-2.289100	-2.257526	-2.225953	-2.194379	-2.162805
Root	-2.388267	-2.297926	-2.297926	-2.297926	-2.297926	-2.232191	-1.662712	-1.662712

```
knitr::kable(result[,33:40] ,booktabs = TRUE, align='c')
```

Initial Value	-2.131231	-2.099657	-2.068084	-2.036510	-2.004936	-1.973362	-1.941788	-1.910215
Root	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712

```
knitr::kable(result[,41:48] ,booktabs = TRUE, align='c')
```

Initial Value	-1.878641	-1.847067	-1.815493	-1.783919	-1.752346	-1.720772	-1.689198	-1.657624
Root	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712

```
knitr::kable(result[,29:56] ,booktabs = TRUE, align='c')
```

Initial Value	-2.257526	-2.225953	-2.194379	-2.162805	-2.131231	-2.099657	-2.068084	-2.036510	-2
Root	-2.297926	-2.232191	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1.662712	-1

```
knitr::kable(result[,57:64] ,booktabs = TRUE, align='c')
```

Initial Value	-1.3734601	-1.3418863	-1.3103125	-1.2787387	-1.2471649	-1.2155911	-1.1840173	-1.152
Root	-0.9544054	-0.9544058	-0.9544058	-0.9544058	-0.9544058	-0.9544053	-0.9544058	-0.954

```
knitr::kable(result[,65:72] ,booktabs = TRUE, align='c')
```

Initial Value	-1.1208697	-1.0892959	-1.0577221	-1.0261484	-0.9945746	-0.9630008	-0.9314270	-0.899
Root	-0.9544053	-0.9544058	-0.9544056	-0.9544058	-0.9544052	-0.9544048	-0.9544058	-0.954

```
knitr::kable(result[,73:80] ,booktabs = TRUE, align='c')
```

Initial Value	-0.8682794	-0.8367056	-0.8051318	-0.7735580	-0.7419842	-0.7104104	-0.6788366	-0.647
Root	-0.9544058	-0.9544058	0.0031186	0.0031182	0.0031187	0.0031182	0.0031182	0.003

```
knitr::kable(result[,81:88] ,booktabs = TRUE, align='c')
```

Initial Value	-0.6156890	-0.5841152	-0.5525414	-0.5209676	-0.4893938	-0.4578200	-0.4262462	-0.394
Root	0.0031183	0.0031182	0.0031182	0.0031182	0.0031182	0.0031182	0.0031182	0.003

```r
knitr::kable(result[,89:96] ,booktabs = TRUE, align='c')
```

Initial Value	-0.3630986	-0.3315249	-0.2999511	-0.2683773	-0.2368035	-0.2052297	-0.1736559	-0.142
Root	0.0031185	0.0031182	0.0031182	0.0031182	0.0031182	0.0031182	0.0031184	0.003

```r
knitr::kable(result[,97:104] ,booktabs = TRUE, align='c')
```

Initial Value	-0.1105083	-0.0789345	-0.0473607	-0.0157869	0.0157869	0.0473607	0.0789345	0.110508
Root	0.0031182	0.0031182	0.0031185	0.0031182	0.0031182	0.0031188	0.0031182	0.003118

```r
knitr::kable(result[,105:112] ,booktabs = TRUE, align='c')
```

Initial Value	0.1420821	0.1736559	0.2052297	0.2368035	0.2683773	0.2999511	0.3315249	0.3630986
Root	0.0031182	0.0031182	0.0031188	0.0031182	0.0031182	0.0031182	0.0031182	0.0031182

```r
knitr::kable(result[,113:120] ,booktabs = TRUE, align='c')
```

Initial Value	0.3946724	0.4262462	0.4578200	0.4893938	0.5209676	0.5525414	0.5841152	0.6156890
Root	0.0031182	0.0031182	0.0031182	0.0031182	0.8126374	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,121:128] ,booktabs = TRUE, align='c')
```

Initial Value	0.6472628	0.6788366	0.7104104	0.7419842	0.7735580	0.8051318	0.8367056	0.8682794
Root	0.8126364	0.8126374	0.8126374	0.8126372	0.8126374	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,129:136] ,booktabs = TRUE, align='c')
```

Initial Value	0.8998532	0.9314270	0.9630008	0.9945746	1.0261484	1.0577221	1.0892959	1.1208697
Root	0.8126373	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,137:144] ,booktabs = TRUE, align='c')
```

Initial Value	1.1524435	1.1840173	1.2155911	1.2471649	1.2787387	1.3103125	1.3418863	1.3734601
Root	0.8126374	0.8126374	0.8126374	0.8126374	0.8126373	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,145:152] ,booktabs = TRUE, align='c')
```

Initial Value	1.4050339	1.4366077	1.4681815	1.4997553	1.5313291	1.5629029	1.5944767	1.6260505
Root	0.8126374	0.8126371	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,153:160] ,booktabs = TRUE, align='c')
```

Initial Value	1.6576243	1.6891981	1.7207719	1.7523457	1.7839194	1.8154932	1.8470670	1.8786408
Root	0.8126374	0.8126365	0.8126372	0.8126374	0.8126374	0.8126374	0.8126374	0.8126374

```r
knitr::kable(result[,161:168] ,booktabs = TRUE, align='c')
```

Initial Value	1.9102146	1.9417884	1.973362	2.004936	2.036510	2.068084	2.099657	2.131231
Root	0.8126374	0.8126374	2.007223	2.007223	2.007223	2.007223	2.007223	2.007223

```r
knitr::kable(result[,169:176] ,booktabs = TRUE, align='c')
```

Initial Value	2.162805	2.194379	2.225953	2.257526	2.289100	2.320674	2.352248	2.383822
Root	2.007223	2.007222	2.237013	2.237013	2.374712	2.374712	2.374712	2.374712

```r
knitr::kable(result[,177:184] ,booktabs = TRUE, align='c')
```

Initial Value	2.415395	2.446969	2.478543	2.510117	2.541691	2.573264	2.604838	2.636412
Root	2.374712	2.374712	2.488450	2.488450	2.848416	2.848415	2.848415	2.848416

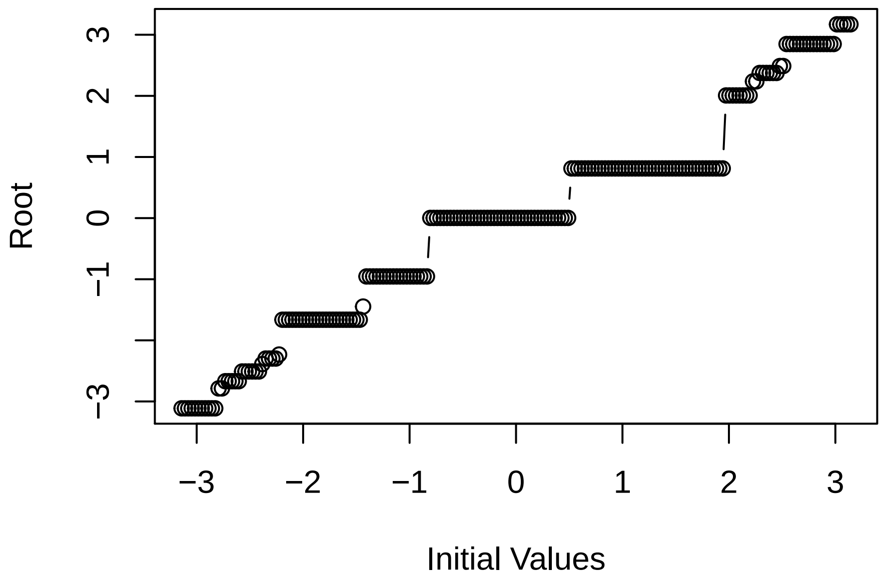
```r
knitr::kable(result[,185:192] ,booktabs = TRUE, align='c')
```

Initial Value	2.667986	2.699559	2.731133	2.762707	2.794281	2.825855	2.857429	2.889002
Root	2.848415	2.848415	2.848416	2.848415	2.848415	2.848415	2.848416	2.848415

```r
knitr::kable(result[,193:200] ,booktabs = TRUE, align='c')
```

Initial Value	2.920576	2.952150	2.983724	3.015297	3.046871	3.078445	3.110019	3.141593
Root	2.848415	2.848415	2.848415	3.170715	3.170715	3.170715	3.170715	3.170715

```r
plot(seq(-pi, pi, length.out = 200), result[2,], xlab="Initial Values", ylab="Root", type="b")
```




```r
DATA <- as.data.frame(round(result[2,], digits = 4))
colnames(DATA) = c('Root')
uniq <- unique(DATA)
knitr::kable(uniq)
```

	Root
1	-3.1125
12	-2.7866
14	-2.6689
19	-2.5094
25	-2.3883
26	-2.2979
30	-2.2322
31	-1.6627
55	-1.4475
56	-0.9544
75	0.0031
117	0.8126
163	2.0072
171	2.2370
173	2.3747
179	2.4884
181	2.8484
196	3.1707

2 Modeling beetle data

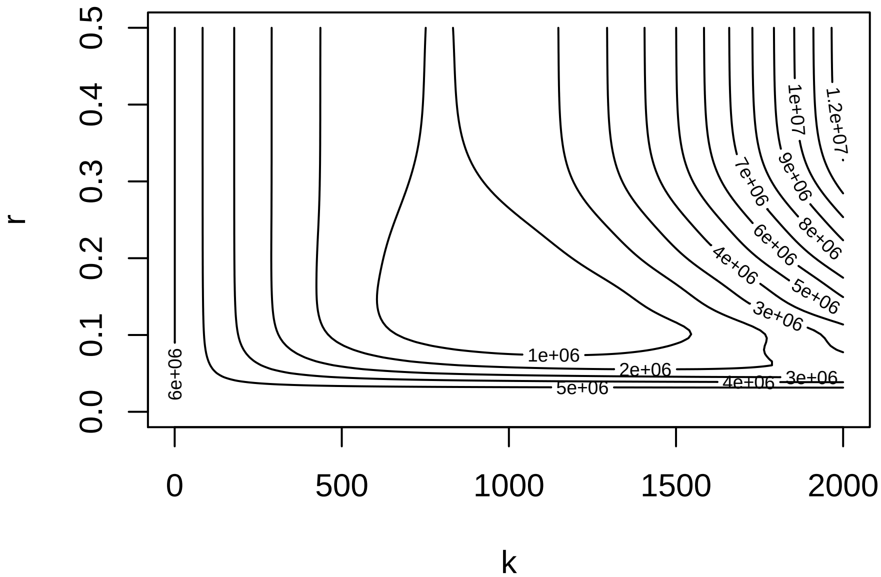
2.1 Show the contour plot of the sum of squared errors AND Gauss-Newton method

```
library(plotly)
```

```
## Loading required package: ggplot2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
Data <- data.frame(days = c(0, 8, 28, 41, 63, 69, 97, 117, 135, 154), beetles = c(2, 47, 192, 2
t <- Data$days
b <- Data$beetles
N0 <- b[1]
ct <- function(k, r)
{
   sum((k * (N0) / (N0 + (k - N0) * exp(-r * t)) - b)^2)
}
k <- seq(0, 2000, length.out = 1e4)
r <- seq(0, 0.5, length.out = 1e2)
z <- outer(k,r, Vectorize(ct) )
contour(k, r, z, xlab = "k", ylab = "r", main = "contour plot")
```

contour plot




```r
beetles <- data.frame(
   days = c(0, 8, 28, 41, 63, 69, 97, 117, 135, 154),
   beetles = c(2, 47, 192, 256, 768, 896, 1120, 896, 1184, 1024))

errors <- rep(NA, nrow(beetles))

nls(beetles ~ (K*beetles[1])/(beetles[1]+(K-beetles[1])*exp(-r*days)),
    start = list(K = 1000, r = 1), data = beetles, trace = TRUE)
```

```
## 1986152 :  1000    1
## 951504.1 :  792.0000022   0.3313653
## 316776.6 :  812.6160939   0.1314344
## 142639.1 :  920.0366145   0.1255537
## 73781.4 :  1039.9824403    0.1191475
## 73424.36 :  1048.7520644    0.1184275
## 73419.83 :  1049.3061406    0.1182953
## 73419.7 :  1049.390377    0.118273
## 73419.7 :  1049.4043999    0.1182692
## 73419.7 :  1049.4067631    0.1182685
```

```
## Nonlinear regression model
##   model: beetles ~ (K * beetles[1])/(beetles[1] + (K - beetles[1]) * exp(-r *     days))
##    data: beetles
##         K         r
## 1049.4068    0.1183
##  residual sum-of-squares: 73420
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 5.892e-06
```

By Gauss-Newton method, I can get 1049.4068 and 0.1183 for K and r

2.2 Find the maximum likelihood estimators and Estimate the variance your parameter estimates

```
rm(list=ls())
l <- expression(
  log(1/(sqrt(2*pi)*sigma))-(log((2*2+2*(K-2)*exp(-r*0))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*47+47*(K-2)*exp(-r*8))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*192+192*(K-2)*exp(-r*28))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*256+256*(K-2)*exp(-r*41))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*768+768*(K-2)*exp(-r*63))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*896+896*(K-2)*exp(-r*69))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*1120+1120*(K-2)*exp(-r*97))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*896+896*(K-2)*exp(-r*117))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*1184+1184*(K-2)*exp(-r*135))/(2*K)))^2/(2*sigma^2)+
  log(1/(sqrt(2*pi)*sigma))-(log((2*1024+1024*(K-2)*exp(-r*154))/(2*K)))^2/(2*sigma^2))

lpk <- D(l,"K")
lpr <- D(l,"r")
lps <- D(l,"sigma")
lppkk <- D(D(l,"K"),"K")
lppkr <- D(D(l,"K"),"r")
lppks <- D(D(l,"K"),"sigma")
lpprr <- D(D(l,"r"),"r")
lpprs <- D(D(l,"r"),"sigma")
lppss <- D(D(l,"sigma"),"sigma")
count <- 0
process <- TRUE
krs <- matrix(c(1050, 0.12, 0.5))
while(process){
  K <- krs[1]
  r <- krs[2]
  sigma <- krs[3]
  gp <- matrix(c(eval(lpk), eval(lpr), eval(lps)))
  gpt <- t(gp)
  M <- matrix(c(eval(lppkk),eval(lppkr),eval(lppks),eval(lppkr),eval(lpprr),
                eval(lpprs),eval(lppks),eval(lpprs),eval(lppss)),byrow=TRUE,nrow=3)
  Minv <- solve(M)
  krs <-  krs - Minv %*% gp
  count <- count + 1
  if(gpt%*%gp < 1e-6 | count == 1000)
    process = FALSE
}
count
```

```
## [1] 8
```

```r
krss <- matrix(c(K,r,sigma^2), ncol = 3)
colnames(krss) <- c("K", "r", "sigma2")
knitr::kable(krss)
```

K	r	sigma2
820.3801	0.19264	0.4148444

```r
vari <- solve(-M)
colnames(vari) <- row.names(vari) <- c("K", "r", "sigma")
knitr::kable(vari)
```

	K	r	sigma
K	62464.5546683	-9.0597875	0.0000049
r	-9.0597875	0.0039784	0.0000000
sigma	0.0000049	0.0000000	0.0207422