Optimization-ZhuangZimin

Zimin Zhuang

2/7/2018

```
## Warning: package 'knitr' was built under R version 3.3.2
```

1

(a)

$$L(\theta) = p_1 p_2 \ldots p_n$$

$$lnL(\theta) = ln(p_1) + \cdots + p_n$$

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

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

$$l'(\theta) = -0 - \sum_{i=1}^{n} \frac{2(\theta - x_i)}{1 + (\theta - x_i)^2}$$

$$l'(\theta) = -2\sum_{i=1}^{n} \frac{(\theta - x_i)}{1 + (\theta - x_i}$$

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

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

$$p'(x) = -\frac{2(x - \theta)}{\pi[1 + (x - \theta)^2]^2}$$

$$I(\theta) = \frac{4n}{\pi} \int_{-\infty}^{\infty} \frac{(x - \theta)^2}{[1 + (x - \theta)^2]^3} dx$$

$$I(\theta) = \frac{4n}{\pi} \int_{-\infty}^{\infty} \frac{x^2}{[1 + x^2]^3} dx$$

Take substitution: $x = tan\theta$, $1 + x^2 = \frac{1}{cos^2\theta}$, we have:

$$I(\theta) = \frac{4n}{\pi} \int_{-\infty}^{\infty} \frac{sin^2\theta}{cos^2\theta} cos^6\theta d(tan\theta)$$

$$I(\theta) = \frac{4n}{\pi} \int_{-\infty}^{\infty} \frac{1}{cos^2\theta} sin^2\theta cos^4\theta d\theta$$

$$I(\theta) = \frac{4n}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{sin^2 2\theta}{4} d\theta$$

$$I(\theta) = \frac{4n}{\pi} \frac{\pi}{8} = n/2$$

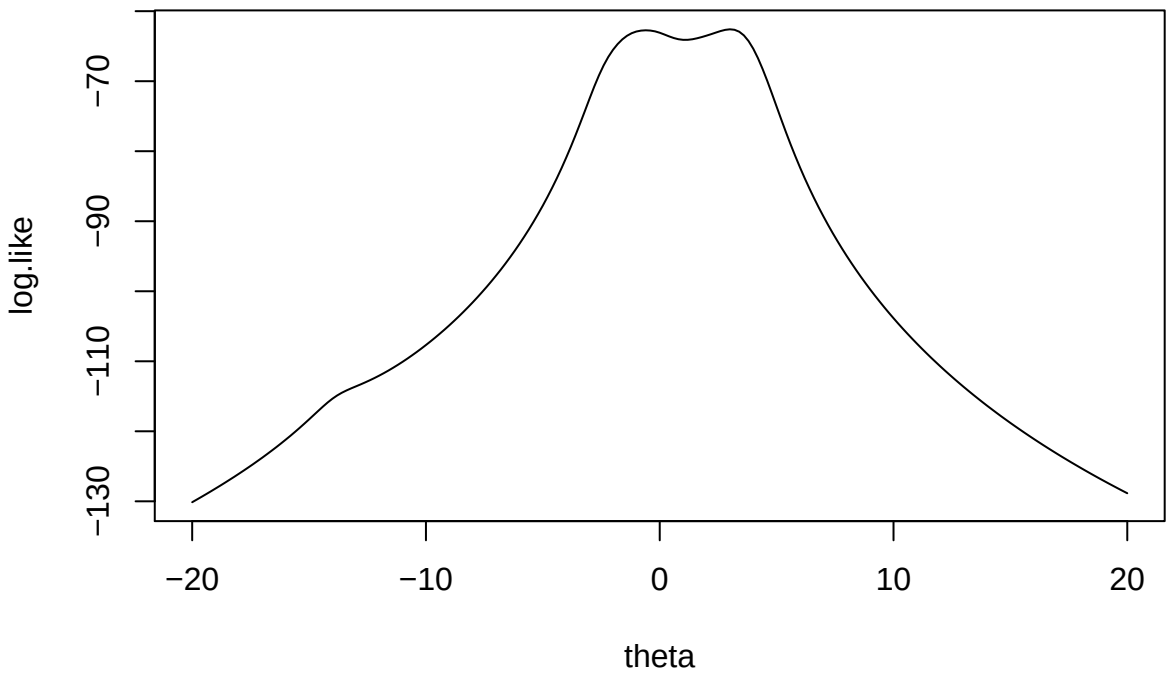
(b)

Log-likelihood function are given as follows:

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

From the output, it is reasonable to conclude that sample mean is a good starting point. Compare to other starting point, if the starting point is largely departed from actual value (for example -11 or 38), optimization result will become unreasonable.

```
x <- c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44,
       3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75)
#theta <- c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)
theta <- seq(from = -20, to = 20, by = 0.2)
i <- 1
log.like <- rep(0,201)
for (i in 1:201) {
  log.like[i] <- -18*log(pi) - sum(log(1+(theta[i]-x)^2))
}
plot(theta,log.like,type = "l")
```




```
library(elliptic)
x <- c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44,
       3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75)
f <- function(theta){-2*sum((theta-x)/(1+(theta-x)^2))}
fdash <- function(theta){-2*sum((1-(theta-x)^2)/((1+(theta-x)^2))^2)}

s.p <- c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)

for (i in 1:9) {
  result <- newton_raphson(s.p[i],f,fdash,maxiter = 100)
  print(paste0("starting point is: ",s.p[i]))
```

```
  print(paste0("Newton Raphson result: ",result$root))
}
```

```
## [1] "starting point is: -11"
## [1] "Newton Raphson result: -208068807539362688"
## [1] "starting point is: -1"
## [1] "Newton Raphson result: -0.591473520517995"
## [1] "starting point is: 0"
## [1] "Newton Raphson result: -0.591473520517994"
## [1] "starting point is: 1.5"
## [1] "Newton Raphson result: 1.09273022412554"
## [1] "starting point is: 4"
## [1] "Newton Raphson result: 3.0213454432316"
## [1] "starting point is: 4.7"
## [1] "Newton Raphson result: -0.591473520517994"
## [1] "starting point is: 7"
## [1] "Newton Raphson result: 173648209384964096"
## [1] "starting point is: 8"
## [1] "Newton Raphson result: 183820724297020736"
## [1] "starting point is: 38"
## [1] "Newton Raphson result: 322657596589048704"
```

```
mean(x)
```

```
## [1] 3.257778
```

```
#sample mean 3.257778
newton_raphson(3.257778,f,fdash,maxiter = 100)
```

```
## $root
## [1] 3.021345
##
## $f.root
## [1] -6.938894e-17
##
## $iter
## [1] 5
```

(c)

```
p.fixed <- function(p0,alpha,obs,tol = 1E-6,max.iter = 1000,verbose = F){
  pold <- p0
  pnew <- pold + alpha * (-2)*sum((pold-obs)/(1+(pold-obs)^2))
  iter <- 1
  while ((abs(pnew - pold) > tol) && (iter < max.iter)){
    pold <- pnew
    pnew <-  pold + alpha * (-2)*sum((pold-obs)/(1+(pold-obs)^2))
    iter <- iter + 1
    if(verbose)
      cat("At iteration", iter, "value of p is:", pnew, "\n")
  }
  if (abs(pnew - pold) > tol) {
    cat("Algorithm failed to converge \n")
    return(c("Failed to Converge"))
```

```
  }
  else {
    cat("Algorithm converged, in :" ,iter,"iterations \n")
    return(pnew)
  }
}

x <- c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44,
       3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75)

p.fixed(p0 = -1,alpha = 1,obs = x)
```

```
## Algorithm failed to converge
## [1] "Failed to Converge"
```

```
p.fixed(p0 = -1,alpha = 0.64,obs = x)
```

```
## Algorithm converged, in : 11 iterations
## [1] -0.5914734
```

```
p.fixed(p0 = -1,alpha = 0.25,obs = x)
```

```
## Algorithm converged, in : 20 iterations
## [1] -0.5914741
```

```
#alpha = 1 will not converge.

s.p <- c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)
alpha <- c(1, 0.64, 0.25)
i <- 1
for (i in 1:9){
  j <- 1
  for (j in 1:3){
    result <- p.fixed(p0 = s.p[i],alpha = alpha[j],obs = x)
    print(paste0("For starting point ",s.p[i], ", Alpha ",
                 alpha[j], ". Fix-point result is ", result,"."))
  }
}
```

```
## Algorithm converged, in : 218 iterations
## [1] "For starting point -11, Alpha 1. Fix-point result is -0.591473976533586."
## Algorithm converged, in : 17 iterations
## [1] "For starting point -11, Alpha 0.64. Fix-point result is -0.591473377966645."
## Algorithm converged, in : 28 iterations
## [1] "For starting point -11, Alpha 0.25. Fix-point result is -0.591474479397574."
## Algorithm failed to converge
## [1] "For starting point -1, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm converged, in : 11 iterations
## [1] "For starting point -1, Alpha 0.64. Fix-point result is -0.591473394970767."
## Algorithm converged, in : 20 iterations
## [1] "For starting point -1, Alpha 0.25. Fix-point result is -0.591474110188776."
## Algorithm failed to converge
## [1] "For starting point 0, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm converged, in : 11 iterations
## [1] "For starting point 0, Alpha 0.64. Fix-point result is -0.591473627757117."
```

```
## Algorithm converged, in : 20 iterations
## [1] "For starting point 0, Alpha 0.25. Fix-point result is -0.591472555721519."
## Algorithm failed to converge
## [1] "For starting point 1.5, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm failed to converge
## [1] "For starting point 1.5, Alpha 0.64. Fix-point result is Failed to Converge."
## Algorithm converged, in : 12 iterations
## [1] "For starting point 1.5, Alpha 0.25. Fix-point result is 3.02134543948525."
## Algorithm failed to converge
## [1] "For starting point 4, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm converged, in : 12 iterations
## [1] "For starting point 4, Alpha 0.64. Fix-point result is -0.591473611258194."
## Algorithm converged, in : 8 iterations
## [1] "For starting point 4, Alpha 0.25. Fix-point result is 3.0213454021169."
## Algorithm failed to converge
## [1] "For starting point 4.7, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm converged, in : 13 iterations
## [1] "For starting point 4.7, Alpha 0.64. Fix-point result is -0.591473571540865."
## Algorithm converged, in : 9 iterations
## [1] "For starting point 4.7, Alpha 0.25. Fix-point result is 3.02134544032359."
## Algorithm failed to converge
## [1] "For starting point 7, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm failed to converge
## [1] "For starting point 7, Alpha 0.64. Fix-point result is Failed to Converge."
## Algorithm converged, in : 8 iterations
## [1] "For starting point 7, Alpha 0.25. Fix-point result is 3.02134543411167."
## Algorithm failed to converge
## [1] "For starting point 8, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm converged, in : 14 iterations
## [1] "For starting point 8, Alpha 0.64. Fix-point result is -0.591473571289484."
## Algorithm converged, in : 10 iterations
## [1] "For starting point 8, Alpha 0.25. Fix-point result is 3.02134544078601."
## Algorithm failed to converge
## [1] "For starting point 38, Alpha 1. Fix-point result is Failed to Converge."
## Algorithm failed to converge
## [1] "For starting point 38, Alpha 0.64. Fix-point result is Failed to Converge."
## Algorithm converged, in : 93 iterations
## [1] "For starting point 38, Alpha 0.25. Fix-point result is 3.02134543943626."
```

(d)

```
x <- c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44,
       3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75)
s.p <- c(-11, -1, 0, 1.5, 4, 4.7, 7, 8, 38)

i <- 2

for (i in 1:9) {
  f <- function(theta){length(x)*log(pi)+sum(log(1+(theta-x)^2))}
  grf <- function(theta){2*sum((theta-x)/(1+(theta-x)^2))}
  fs <- function(theta){matrix(length(x)/2,nrow =1)}
  print(paste0("starting point =",s.p[i]))
```

```
  z <- nlminb(s.p[i],f,grf,fs)
  print(paste0("Fisher scoring result is ",z$par))
  f <- function(theta){-2*sum((theta-x)/(1+(theta-x)^2))}
  fdash <- function(theta){-2*sum((1-(theta-x)^2)/((1+(theta-x)^2))^2)}
  result <- newton_raphson(z$par,f,fdash,maxiter=1000)
  print(paste0("Newton-Rasphon result is ", result$root))
}
```

```
## [1] "starting point =-11"
## [1] "Fisher scoring result is -0.591600260088552"
## [1] "Newton-Rasphon result is -0.591473520517995"
## [1] "starting point =-1"
## [1] "Fisher scoring result is -0.591596692772024"
## [1] "Newton-Rasphon result is -0.591473520517995"
## [1] "starting point =0"
## [1] "Fisher scoring result is -0.591358826338748"
## [1] "Newton-Rasphon result is -0.591473520517995"
## [1] "starting point =1.5"
## [1] "Fisher scoring result is 3.021315609095"
## [1] "Newton-Rasphon result is 3.0213454432316"
## [1] "starting point =4"
## [1] "Fisher scoring result is 3.02139279796859"
## [1] "Newton-Rasphon result is 3.0213454432316"
## [1] "starting point =4.7"
## [1] "Fisher scoring result is 3.02138744391034"
## [1] "Newton-Rasphon result is 3.0213454432316"
## [1] "starting point =7"
## [1] "Fisher scoring result is 3.02138193047027"
## [1] "Newton-Rasphon result is 3.0213454432316"
## [1] "starting point =8"
## [1] "Fisher scoring result is 3.02137588743538"
## [1] "Newton-Rasphon result is 3.0213454432316"
## [1] "starting point =38"
## [1] "Fisher scoring result is 19.1751084727589"
## [1] "Newton-Rasphon result is 288965687247457152"
```

(e)

Comparing methods, fixed-point shows less stability than Fisher scoring-Newton Raphson method. In fixed-point method, the chosen of α will dramatically impact the result and it is likely fail to converge. As for speed, fixed-point are more effcient than the two steped Fisher scoring-Newton Raphson method, however, as we only have limited observations, the difference in speed is not significant.

2

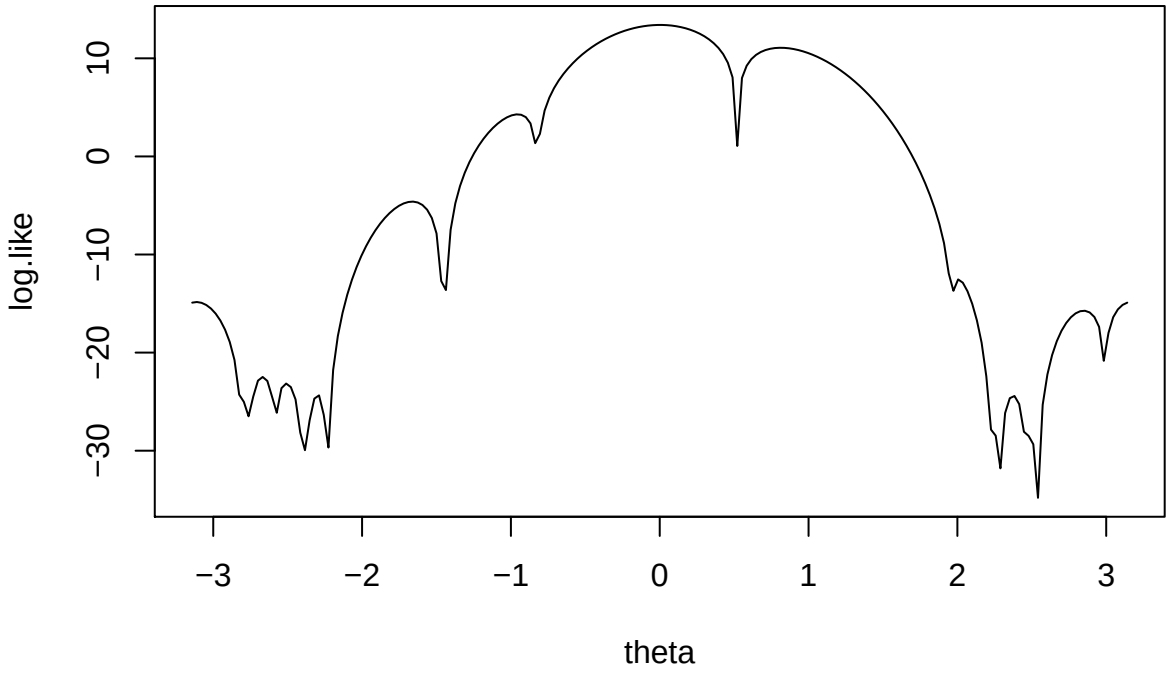
(a)

From the given density function, we have the log-likelihood function as:

$$l(\theta) = \sum_{i=1}^{19} ln(p(x;\theta))$$

$$l(\theta) = \sum_{i=1}^{19} ln(\frac{1 - cos(x - \theta)}{2\pi})$$

```
x <- 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)

theta <- seq(from = -pi, to = pi, by = 2*pi/199)
i <- 1
log.like <- rep(0,200)
for (i in 1:200) {
  log.like[i] <- sum(log((1-cos(x-theta[i]))/2*pi))
}
plot(theta,log.like,type = "l")
```



(b)

$$E[x|\theta] = \int_0^{2\pi} x\frac{1 - cos(x - \theta)}{2\pi}$$

$$E[x|\theta] = \frac{1}{2\pi}[\int_0^{2\pi} xdx - \int_0^{2\pi} xcos(x - \theta)dx)]$$

$$E[x|\theta] = \pi - \frac{1}{2\pi}(xsin(x - \theta) + cos(x - \theta))|_0^{2\pi}$$

$$E[x|\hat{\theta}_{moment}] = \pi + sin(\hat{\theta}_{moment})$$

Here, solve for θ is equivalent to find a numerical solution to $\pi + sin(\hat{\theta}_{moment}) - \bar{x} = 0$. We will now use "uniroot" function to solve it. Notice that interval for θ are further seprarted into four partitions. Three unique solutions are found: -3.236988, 0.09539388 and 3.046199.

```
mean(x)
```

```
## [1] 3.236842
```

```r
f1<-function(theta){
  pi+sin(theta)-mean(x)
}

uniroot(f1,lower = -pi,upper = -pi/2,extendInt = "yes")$root[1]
```

```
## [1] -3.236988
```

```r
uniroot(f1,lower = -pi/2,upper = 0,extendInt = "yes")$root[1]
```

```
## [1] -3.236988
```

```r
uniroot(f1,lower = 0,upper = pi/2,extendInt = "yes")$root[1]
```

```
## [1] 0.09539388
```

```r
uniroot(f1,lower = pi/2,upper = pi,extendInt = "yes")$root[1]
```

```
## [1] 3.046199
```

```r
theta.mom <- c(-3.236988, 0.09539388, 3.046199)
```

(c)

Based on solutions in (b): -3.236988 gives the result as -3.112471. 0.09539388 gives the result as 0.003118157. 3.046199 gives the result as 3.170715.

```r
library(elliptic)
x <- 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 <- function(theta){
  f <- 0
  i <- 1
  for (i in 1:19){
  f <- f + sin(theta-x[i])/(1-cos(theta-x[i]))
  }
  return(f)
}


fdash <- function(theta){
  f.d <- sum( -1 / (1-cos(theta-x)))
  return(f.d)
}

theta.mom <- c(-3.236988, 0.09539388, 3.046199)

newton_raphson( -3.236988,f,fdash,maxiter = 1000)
```

```
## $root
## [1] -3.112471
##
## $f.root
## [1] 1.904032e-14
##
```

```
## $iter
## [1] 7
```

```
newton_raphson( 0.09539388,f,fdash,maxiter = 1000)
```

```
## $root
## [1] 0.003118157
##
## $f.root
## [1] 0
##
## $iter
## [1] 6
```

```
newton_raphson( 3.046199,f,fdash,maxiter = 1000)
```

```
## $root
## [1] 3.170715
##
## $f.root
## [1] -1.720846e-15
##
## $iter
## [1] 7
```

(d)

Starting point of -2.7 will give the result as -2.668857. Starting point of 2.7 will give the result as 2.848415. This indicates that the chosen of initial point is a very sensitive factor to the optimized result.

```
newton_raphson(-2.7, f, fdash, maxiter = 100)
```

```
## $root
## [1] -2.668857
##
## $f.root
## [1] -2.88259e-13
##
## $iter
## [1] 5
```

```
newton_raphson(2.7, f, fdash, maxiter = 100)
```

```
## $root
## [1] 2.848415
##
## $f.root
## [1] -3.069767e-14
##
## $iter
## [1] 6
```

(e)

```
N_R <- function (initial, f, fdash, maxiter, give = TRUE, tol = .Machine$double.eps)
{
  old.guess <- initial
  for (i in seq_len(maxiter)) {
    new.guess <- old.guess - f(old.guess)/fdash(old.guess)
    jj <- f(new.guess)
    if (is.na(jj) | is.infinite(jj)) {
      break
    }
    if (near.match(new.guess, old.guess) | abs(jj) < tol) {
      if (give) {
        return(list(root = new.guess, f.root = jj, iter = i))
      }
      else {
        return(new.guess)
      }
    }
    old.guess <- new.guess
  }
  return(list(root = "Failed to Converge", f.root = jj, iter = i))
}

s.p <- seq(from = -pi, to = pi, length.out = 200)


out <- data.frame(
  start.point <- s.p[1:200],
  root <- rep(0,200)
)
names(out) <- c("start.point","root")

for(i in 1:200) {
  result <- N_R(s.p[i],f ,fdash, maxiter=1000)
  out[i,2] <- result$root
}
target <- which(out$root == "Failed to Converge")
out$root <- round(as.numeric(out$root), digits = 10)
out$root[target] <- c("Failed to Converge")
out$root <- as.factor(out$root)
#There are 18 levels
i <- 1
for (i in 1:length(levels(out$root))){
  subgrp <- data.frame(
    start.point <- rep(0,length(which(out$root == levels(out$root)[i]))),
    root <- rep(0,length(which(out$root == levels(out$root)[i])))
  )
  names(subgrp) <- c("start.point","root")
  subgrp$start.point <- out[which(out$root == levels(out$root)[i]),1]
  subgrp$root <- out[which(out$root == levels(out$root)[i]),2]
  assign(paste0("root.",i), subgrp)
}
```

Table 1: Result type 1

start.point	root
-3.141593	-3.1124705067
-3.110019	-3.1124705067
-3.078445	-3.1124705067
-3.046871	-3.1124705067
-3.015297	-3.1124705067
-2.983724	-3.1124705067
-2.952150	-3.1124705067
-2.920576	-3.1124705067
-2.889002	-3.1124705067
-2.857429	-3.1124705067
-2.825855	-3.1124705067

Table 2: Result type 2

start.point	root
-2.794281	-2.7865568524
-2.762707	-2.7865568524

Table 3: Result type 3

start.point	root
-2.731133	-2.668857459
-2.699559	-2.668857459
-2.667986	-2.668857459
-2.636412	-2.668857459
-2.604838	-2.668857459

Table 4: Result type 4

start.point	root
-2.573264	-2.5093560332
-2.541691	-2.5093560332
-2.510117	-2.5093560332
-2.478543	-2.5093560332
-2.446969	-2.5093560332
-2.415395	-2.5093560332

Table 5: Result type 5

start.point	root
-2.289100	-2.297925969
-2.257526	-2.297925969

Table 6: Result type 6

start.point	root
-2.225953	-2.2321918989

Table 7: Result type 7

start.point	root
-2.194379	-1.6627123955
-2.162805	-1.6627123955
-2.131231	-1.6627123955
-2.099657	-1.6627123955
-2.068084	-1.6627123955
-2.036510	-1.6627123955
-2.004936	-1.6627123955
-1.973362	-1.6627123955
-1.941788	-1.6627123955
-1.910215	-1.6627123955
-1.878641	-1.6627123955
-1.847067	-1.6627123955
-1.815493	-1.6627123955
-1.783919	-1.6627123955
-1.752346	-1.6627123955
-1.720772	-1.6627123955
-1.689198	-1.6627123955
-1.657624	-1.6627123955
-1.626051	-1.6627123955
-1.594477	-1.6627123955
-1.562903	-1.6627123955
-1.531329	-1.6627123955
-1.499755	-1.6627123955
-1.468182	-1.6627123955

Table 8: Result type 8

start.point	root
-1.436608	-1.4475025527

Table 9: Result type 9

start.point	root
-1.4050339	-0.9544058371
-1.3734601	-0.9544058371
-1.3418863	-0.9544058371
-1.3103125	-0.9544058371
-1.2787387	-0.9544058371
-1.2471649	-0.9544058371
-1.2155911	-0.9544058371
-1.1840173	-0.9544058371

start.point	root
-1.1524435	-0.9544058371
-1.1208697	-0.9544058371
-1.0892959	-0.9544058371
-1.0577221	-0.9544058371
-1.0261484	-0.9544058371
-0.9945746	-0.9544058371
-0.9630008	-0.9544058371
-0.9314270	-0.9544058371
-0.8998532	-0.9544058371
-0.8682794	-0.9544058371
-0.8367056	-0.9544058371

Table 10: Result type 10

start.point	root
-0.8051318	0.0031181571
-0.7735580	0.0031181571
-0.7419842	0.0031181571
-0.7104104	0.0031181571
-0.6788366	0.0031181571
-0.6472628	0.0031181571
-0.6156890	0.0031181571
-0.5841152	0.0031181571
-0.5525414	0.0031181571
-0.5209676	0.0031181571
-0.4893938	0.0031181571
-0.4578200	0.0031181571
-0.4262462	0.0031181571
-0.3946724	0.0031181571
-0.3630986	0.0031181571
-0.3315249	0.0031181571
-0.2999511	0.0031181571
-0.2683773	0.0031181571
-0.2368035	0.0031181571
-0.2052297	0.0031181571
-0.1736559	0.0031181571
-0.1420821	0.0031181571
-0.1105083	0.0031181571
-0.0789345	0.0031181571
-0.0473607	0.0031181571
-0.0157869	0.0031181571
0.0157869	0.0031181571
0.0473607	0.0031181571
0.0789345	0.0031181571
0.1105083	0.0031181571
0.1420821	0.0031181571
0.1736559	0.0031181571
0.2052297	0.0031181571
0.2368035	0.0031181571
0.2683773	0.0031181571
0.2999511	0.0031181571

start.point	root
0.3315249	0.0031181571
0.3630986	0.0031181571
0.3946724	0.0031181571
0.4262462	0.0031181571
0.4578200	0.0031181571
0.4893938	0.0031181571

Table 11: Result type 11

start.point	root
0.5209676	0.8126374167
0.5525414	0.8126374167
0.5841152	0.8126374167
0.6156890	0.8126374167
0.6472628	0.8126374167
0.6788366	0.8126374167
0.7104104	0.8126374167
0.7419842	0.8126374167
0.7735580	0.8126374167
0.8051318	0.8126374167
0.8367056	0.8126374167
0.8682794	0.8126374167
0.8998532	0.8126374167
0.9314270	0.8126374167
0.9630008	0.8126374167
0.9945746	0.8126374167
1.0261484	0.8126374167
1.0577221	0.8126374167
1.0892959	0.8126374167
1.1208697	0.8126374167
1.1524435	0.8126374167
1.1840173	0.8126374167
1.2155911	0.8126374167
1.2471649	0.8126374167
1.2787387	0.8126374167
1.3103125	0.8126374167
1.3418863	0.8126374167
1.3734601	0.8126374167
1.4050339	0.8126374167
1.4366077	0.8126374167
1.4681815	0.8126374167
1.4997553	0.8126374167
1.5313291	0.8126374167
1.5629029	0.8126374167
1.5944767	0.8126374167
1.6260505	0.8126374167
1.6576243	0.8126374167
1.6891981	0.8126374167
1.7207719	0.8126374167
1.7523457	0.8126374167
1.7839194	0.8126374167

start.point	root
1.8154932	0.8126374167
1.8470670	0.8126374167
1.8786408	0.8126374167
1.9102146	0.8126374167
1.9417884	0.8126374167

Table 12: Result type 12

start.point	root
1.973362	2.007223238
2.004936	2.007223238
2.036510	2.007223238
2.068084	2.007223238
2.099657	2.007223238
2.131231	2.007223238
2.162805	2.007223238
2.194379	2.007223238

Table 13: Result type 13

start.point	root
2.225953	2.2370129227
2.257526	2.2370129227

Table 14: Result type 14

start.point	root
2.289100	2.3747116661
2.320674	2.3747116661
2.352248	2.3747116661
2.383822	2.3747116661
2.415395	2.3747116661
2.446969	2.3747116661

Table 15: Result type 15

start.point	root
2.510117	2.4884496509

Table 16: Result type 16

start.point	root
2.541691	2.8484153255
2.573264	2.8484153255

start.point	root
2.604838	2.8484153255
2.636412	2.8484153255
2.667986	2.8484153255
2.699559	2.8484153255
2.731133	2.8484153255
2.762707	2.8484153255
2.794281	2.8484153255
2.825855	2.8484153255
2.857429	2.8484153255
2.889002	2.8484153255
2.920576	2.8484153255
2.952150	2.8484153255
2.983724	2.8484153255

Table 17: Result type 17

start.point	root
3.015297	3.1707148005
3.046871	3.1707148005
3.078445	3.1707148005
3.110019	3.1707148005
3.141593	3.1707148005

Table 18: Result type 18

start.point	root
-2.383822	Failed to Converge
-2.352248	Failed to Converge
-2.320674	Failed to Converge
2.478543	Failed to Converge

3

(a)

First, we need to find a suitable initial value for K and r. From the results. From the given data, it is reasonable to assume $K_0 = 1200$. For r, we need to derive the equation of r involving K and N_t. We have:

$$N_t = \frac{2K}{2 + (K-2)exp(-rt)}$$

$$exp(-rt) = \frac{2(K-N_t)}{N_t(K-2)}$$

$$rt = ln\frac{N_t(K-2)}{2(K-N_t)}$$

$$r = \frac{1}{t} * ln\frac{N_t(K-2)}{2(K-N_t)}$$

Next we plug in all observed N_t and t to calculated a series of r. The mean of this series will be used as the initial value of r.

Based on these, initial values are:

$$(K_0, r_0) = (1200, 0.1716788)$$

Using "nls" function, the optimized value is:

$$(K, r) = (1049.4069185, 0.1182685)$$

```
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))

K <- 1200
r.t <- log((beetles$beetles*(K-2))/(K - beetles$beetles)*2)
r.series <- r.t / beetles$days
mean(r.series[2:10])
```

```
## [1] 0.1716788
```

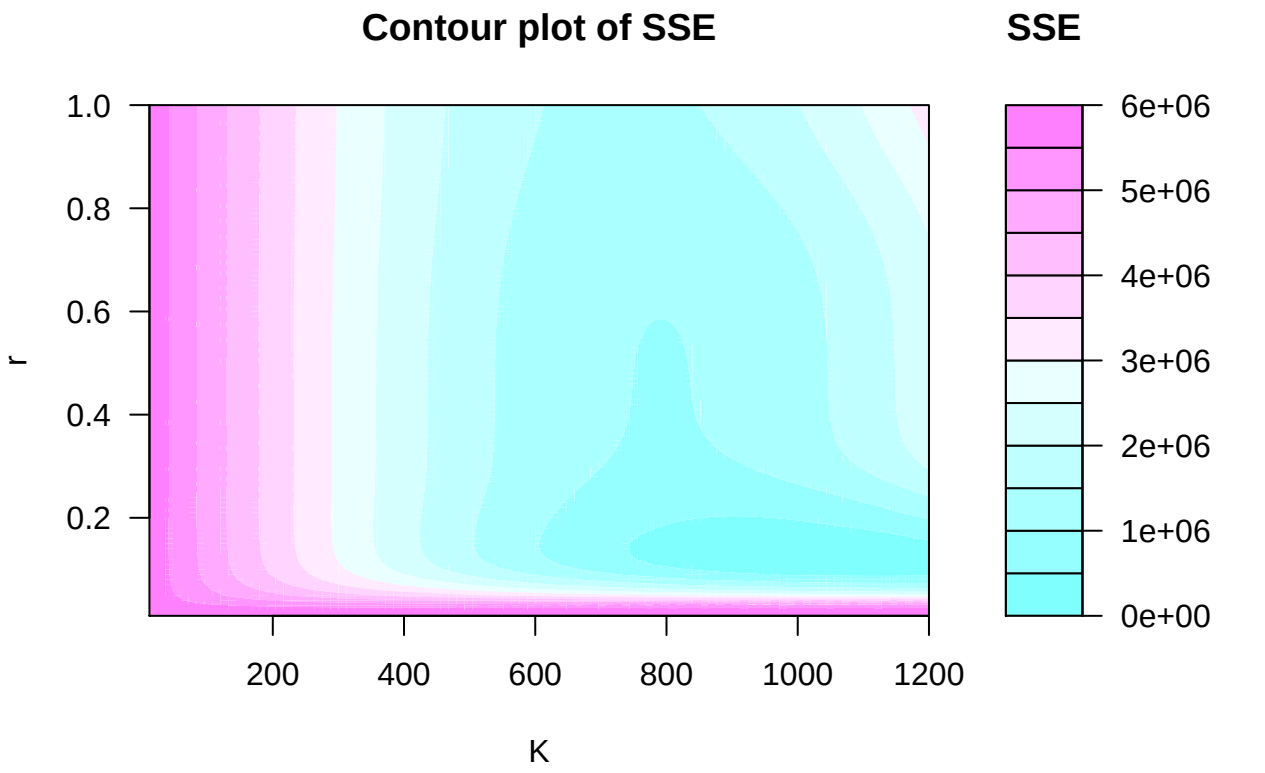
```
#r <- 0.1716788

#theta1 <- N
#theta2 <- r
#beetles ~ theta1*2/(2+(theta1-2)exp(-1*theta2*days))
pop.mod <- nls(beetles ~ N*2/(2+(N-2)*exp((-r)*days)),start = list(N = 1200, r = 0.1716788),data = beetl
```

```
## 677146.3 :  1200.0000000    0.1716788
## 110309.7 :  988.5766717   0.1367234
## 76986.61 :  1024.1437803    0.1224583
## 73503.58 :  1045.8468846    0.1189023
## 73421.83 :  1048.9806817    0.1183764
## 73419.76 :  1049.3390375    0.1182867
## 73419.7 :  1049.3958171    0.1182715
## 73419.7 :  1049.4053152    0.1182689
## 73419.7 :  1049.4069185    0.1182685
```

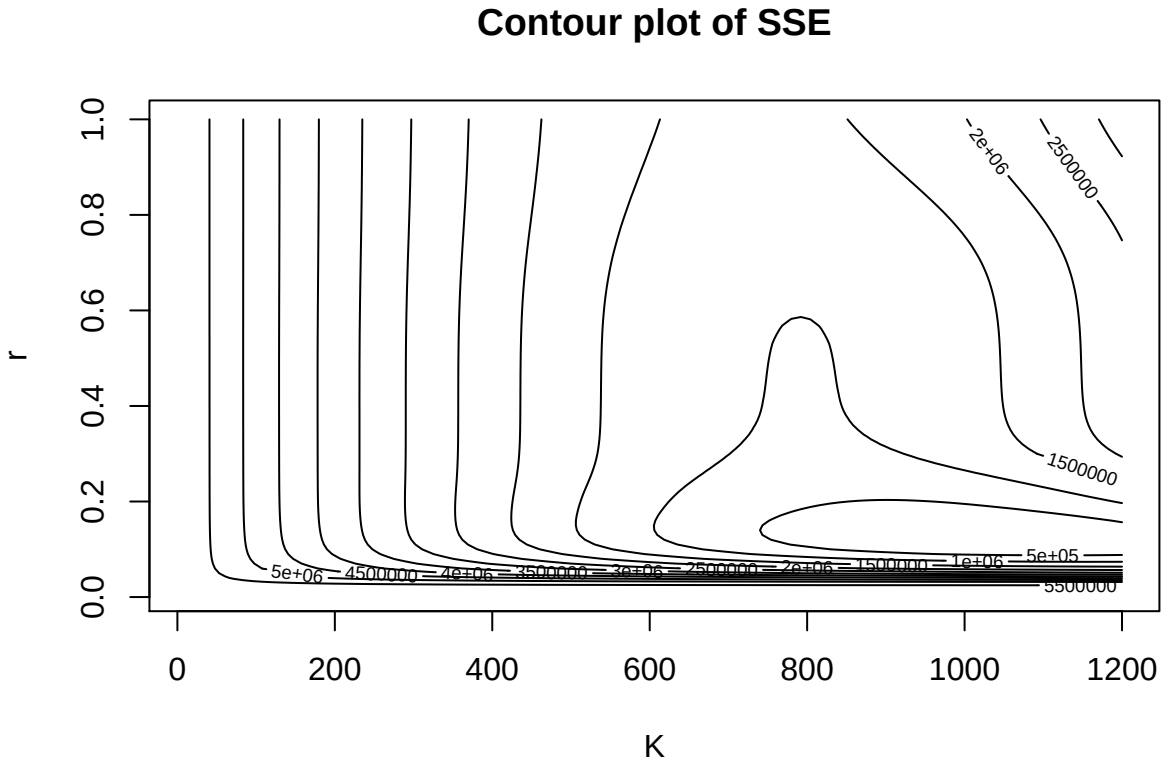
(b)

```
K <- seq(12,1200, by = (1200-12)/99)
r <- seq(0.0100,1, by = (1 - 0.0100)/99)

n.b <- as.numeric(beetles$beetles)
t.d <- as.numeric(beetles$days)

sse <- function(K,r){
    error.sq <- sum((n.b - (K*2)/(2+(K-2)*exp(-r*t.d)))^2)
    return(error.sq)
}

z <- matrix(rep(0,10000),nrow = 100)
j <- 1 #for K
k <- 1 #for r
for (j in 1:100){
```

```
  for(k in 1:100){
    z[j,k] <- sse(K[j],r[k])
  }
}


filled.contour(K,r,z, plot.title = title (main = "Contour plot of SSE", xlab = "K",
                                          ylab = "r"), key.title = title(main = "SSE"))
```




```
contour(K, r, z, xlab = 'K', ylab = 'r', plot.title = title ("Contour plot of SSE"))
```



(c)

Using "nlm" function again, we have the estimated value as follows:

$$(K, r, \sigma) = (820.3811422, 0.1926394, 0.6440836)$$

By solving the hessian matrix, the variances of the related parameters are:

$$(var(K), var(r), var(\sigma) = (6.262790 * 10^4, 4.006745 * 10^{-3}, 2.075824 * 10^{-2})$$

These results shows a larger departure from pervious K and r, one explaination could be the size of the observation set is not sufficient.

```
mlogl3 <- function(theta, N, days) {
  K <- theta[1]
  r <- theta[2]
  sigma <- theta[3]
  t <- days
  mu <- log((K*2)/(2+(K-2)*exp(-r*t)))
  - sum(dnorm(log(N), mu, sigma, log = TRUE))
}

sqrt(var(log(beetles$beetles)))
```

```
## [1] 2.031806
```

```
#2.03
theta.start <- c(1200, 0.17,2.03)
out <- nlm(mlogl3, theta.start, N = beetles$beetles,days = beetles$days,hessian = TRUE)
out
```

```
## $minimum
## [1] 9.790127
##
## $estimate
## [1] 820.3815695   0.1926394   0.6440836
##
## $gradient
## [1]  1.138721e-08  2.900435e-05 -2.714273e-06
##
## $hessian
##               [,1]         [,2]          [,3]
## [1,]  2.389705e-05   0.05442387 -3.048933e-06
## [2,]  5.442387e-02 373.52578204 -5.767475e-02
## [3,] -3.048933e-06  -0.05767475  4.817365e+01
##
## $code
## [1] 2
##
## $iterations
## [1] 38
```

```
theta.hat <- out$estimate
#K = 820.3811422 , r = 0.1926394, sigma = 0.6440836
theta.hat
```

```
## [1] 820.3815695   0.1926394   0.6440836
```

```
hes <- out$hessian
hes
```

```
##               [,1]         [,2]          [,3]
## [1,]  2.389705e-05   0.05442387 -3.048933e-06
## [2,]  5.442387e-02 373.52578204 -5.767475e-02
## [3,] -3.048933e-06  -0.05767475  4.817365e+01
```

```
var.matrix <- solve(hes)
# 6.262790e+04, 4.006745e-03, 2.075824e-02
diag(var.matrix)
```

```
## [1] 6.262790e+04 4.006745e-03 2.075824e-02
```