stat HW2

JIAYI LIU

February 5, 2018

Question 1

(a)

The Cauchy (x,theta) has probability density:

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

Let $x_1, x_2 ... x_n$be an i.i.d smple, $l(\theta)$ the log-likelihood function is:

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

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

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

$$P(x) = \frac{1}{\pi(1+x^2)}$$

$$P^{'}(x) = -\frac{2x}{\pi(1+x^2)^2}$$

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

Set $x = \tan(\alpha); \alpha \in (-\frac{\pi}{2}, \frac{\pi}{2})$ So,

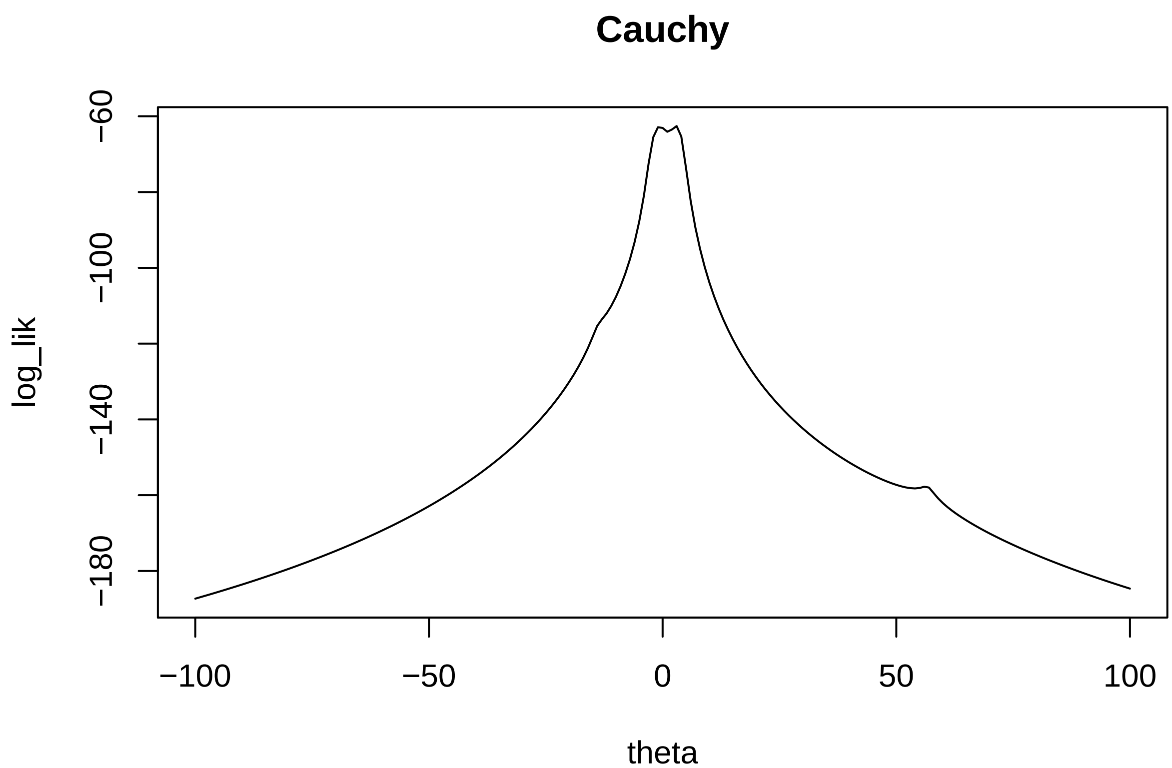
$$I(\theta) = \frac{4n}{\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\cos^{-2}(\alpha)-1}{(cos^{-2}(\alpha))^3} = \frac{4n}{\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\tan^2(\alpha)}{(1+tan^2(\alpha))^3}\frac{1}{\cos^2(\alpha)} d\alpha = \frac{4n}{\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} sin^2(\alpha) * cos^2(\alpha) d\alpha = \frac{4n}{\pi} * \frac{\pi}{8} = \frac{n}{2}$$

(b)

Graph the log-likelihood function with given simple

```
# Sample data
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,
# Range of theta
theta_1<- seq(-100,100,1)
# Log-likely function of Cauchy dist.
```

```r
n<- length(x)
log_lik<- c()
for (i in 1:length(theta_1)){
  log_lik[i]<- -n*log(pi)-sum(log(1+(theta_1[i]-x)^2))
}
# Graph the function of Cauchy dist.
plot(theta_1,log_lik,type="l",ylab= "log_lik",xlab="theta",main="Cauchy")
```



###Find the MLE for theta using the Newton-Raphson method.

```r
# Initial value of theta
initial_value<- c(-11,-1,0,1.5,4,4.7,7,8,38)
# Function of l'(theta)
log_lik_1<- function(x,theta_2){
  -2*sum((theta_2-x)/(1+(theta_2-x)^2))
}
# Function of l'(theta)
log_lik_2<- function(x,theta_2){
  -2*sum((1-(theta_2-x)^2)/((1+(theta_2-x)^2)^2))
}
# Sample mean
s_m<- mean(x)
# Newton method using my own fuction
newton<- function(th){
theta_2<- array()
theta_2[1]<- th # start point theta
i<- 1
```

```r
difference<- 10
while(abs(difference)> 10^(-10) & i<1000){
  theta_2[i+1]<- theta_2[i]-log_lik_1(x,theta_2[i])/log_lik_2(x,theta_2[i])
  difference<- theta_2[i+1]-theta_2[i]
  i=i+1
}
print(theta_2[i])
}
newton(-1)
```

```
## [1] -0.5914735
```

```r
newton(-0)
```

```
## [1] -0.5914735
```

```r
newton(1.5)
```

```
## [1] 1.09273
```

```r
newton(4)
```

```
## [1] 3.021345
```

```r
newton(4.7)
```

```
## [1] -0.5914735
```

```r
newton(s_m)
```

```
## [1] 3.021345
```

```r
## With my own fuction, I only find MLE of theta when starting from ( -1 , 0 , 1.5 , 4 , 4.7 )
## MLE = -0.5914735 when starting from -1
## MLE = -0.5914735 when starting from 0
## MLE = 1.09273 when starting from 1.5
## MLE = 3.021345 when starting from 4
## MLE = -0.5914735 when starting from 4.7
## None MLE when starting from ( -11 , 7 , 8 , 38 )
## MLE = 3.021345 when starting from sample mean, so sample mean is a good starting point

# Newton method using nlminb

## Function being minimized
theta_3<- array()
g<- function(theta_3){
  n*log(pi)+sum(log(1+(theta_3-x)^2))
  }
## Gradient of the function
gr.g <- function(theta_3){
  2*sum((theta_3-x)/(1+(theta_3-x)^2))
  }
## Hessian of the function
hess.g <- function(theta_3){
  hg<- 2*sum((theta_3-x)/(1+(theta_3-x)^2))/(2*sum((1-(theta_3-x)^2)/(1+(theta_3-x)^2)^2))
  return(matrix(hg,nrow = 1))
}
## Get MLE
```

```r
MLE_nlm<- array()
for (i in 1:length(initial_value)){
  MLE_nlm[i] <- nlminb(initial_value[i],g,gr.g,hess.g)$par
  }

## List MLE
print(MLE_nlm)
```

```
## [1] -0.5914735 -0.5914735 -0.5914735  3.0213456  3.0213454  3.0213469
## [7] -0.5914735 -0.5914732 -0.5914735
```

```r
## If sample mean is the starting point
print(newton_nlm <- nlminb(mean(x),g,gr.g,hess.g)$par)
```

```
## [1] 3.021345
```

```r
## Sample mean is a good starting point
```

(c)

Fixed-point iterations using,

$$\alpha \in (1, 0.64, 0.25)$$

```r
fixed_point<- function(al,th){ #al = alpha;th = start point
alpha<- al # When alpha = 1
theta_4<- array()
theta_4[1]<- th
i<- 1
difference<- 10
while(abs(difference)> 10^(-10) & i<1000){
  theta_4[i+1]<- theta_4[i]+alpha*log_lik_1(x,theta_4[i])
  difference<- theta_4[i+1]-theta_4[i]
  i=i+1
}
print(theta_4[i])
}
fixed_point(1,-11)
```

```
## [1] -0.5914735
```

```r
fixed_point(1,-1)
```

```
## [1] 0.1035079
```

```r
fixed_point(1,0)
```

```
## [1] -1.106309
```

```r
fixed_point(1,1.5)
```

```
## [1] 0.1035079
```

```r
fixed_point(1,4)
```

```
## [1] -1.106309
```

```r
fixed_point(1,4.7)
```

```
## [1] -1.171392
```

```r
fixed_point(1,7)
```

```
## [1] -1.171392
```

```r
fixed_point(1,8)
```

```
## [1] 0.2417269
```

```r
fixed_point(1,38)
```

```
## [1] 0.2417269
```

```
## MLE = -0.5914735 when starting from -11
## MLE = 0.1035079 when starting from -1
## MLE = -1.106309 when starting from 0
## MLE = 0.1035079 when starting from 1.5
## MLE = -1.106309 when starting from 4
## MLE = -1.171392 when starting from 4.7
## MLE = -1.171392 when starting from 7
## MLE = 0.2417269 when starting from 8
## MLE = 0.2417269 when starting from 38

# When alpha = 0.64
fixed_point(0.64,-11)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,-11)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,-1)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,0)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,1.5)
```

```
## [1] 3.239838
```

```r
fixed_point(0.64,4)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,4.7)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,7)
```

```
## [1] 2.591518
```

```r
fixed_point(0.64,8)
```

```
## [1] -0.5914735
```

```r
fixed_point(0.64,38)
```

```
## [1] 2.591518
```

```
## MLE = -0.5914735 when starting from -11
## MLE = -0.5914735 when starting from -1
## MLE = -0.5914735 when starting from 0
## MLE = 3.239838 when starting from 1.5
## MLE = -0.5914735 when starting from 4
## MLE = -0.5914735 when starting from 4.7
## MLE = 2.591518 when starting from 7
## MLE = -0.5914735 when starting from 8
## MLE = 2.591518 when starting from 38

# When alpha = 0.25
fixed_point(0.25,-11)
```

```
## [1] -0.5914735
```

```
fixed_point(0.25,-11)
```

```
## [1] -0.5914735
```

```
fixed_point(0.25,-1)
```

```
## [1] -0.5914735
```

```
fixed_point(0.25,0)
```

```
## [1] -0.5914735
```

```
fixed_point(0.25,1.5)
```

```
## [1] 3.021345
```

```
fixed_point(0.25,4)
```

```
## [1] 3.021345
```

```
fixed_point(0.25,4.7)
```

```
## [1] 3.021345
```

```
fixed_point(0.25,7)
```

```
## [1] 3.021345
```

```
fixed_point(0.25,8)
```

```
## [1] 3.021345
```

```
fixed_point(0.25,38)
```

```
## [1] 3.021345
```

```
## MLE = -0.5914735 when starting from -11
## MLE = -0.5914735 when starting from -1
## MLE = -0.5914735 when starting from 0
## MLE = 3.239838 when starting from 1.5
## MLE = 3.021345 when starting from 4
## MLE = 3.021345 when starting from 4.7
## MLE = 3.021345 when starting from 7
## MLE = 3.021345 when starting from 8
## MLE = 3.021345 when starting from 38
```

(d)

Fisher&Newton

```
##Using Fisher scoring to find MLE for theta
theta_5<- array()
fisher<- function(th) {
  theta_5[1]<- th
i<- 1
difference<- 10
while(abs(difference)> 10^(-10) & i<1000){
  theta_5[i+1]<- theta_5[i]+log_lik_1(x,theta_5[i])/(n/2)
  difference<- theta_5[i+1]-theta_5[i]
  i=i+1
}
print(theta_5[i])
}
fisher(-11)
```

```
## [1] -0.5914735
```

```
fisher(-1)
```

```
## [1] -0.5914735
```

```
fisher(0)
```

```
## [1] -0.5914735
```

```
fisher(1.5)
```

```
## [1] 3.021345
```

```
fisher(4)
```

```
## [1] 3.021345
```

```
fisher(4.7)
```

```
## [1] 3.021345
```

```
fisher(7)
```

```
## [1] 3.021345
```

```
fisher(8)
```

```
## [1] 3.021345
```

```
fisher(38)
```

```
## [1] 3.021345
```

```
## MLE = -0.5914735 when starting from -11
## MLE = -0.5914735 when starting from -1
## MLE = -0.5914735 when starting from 0
## MLE = 3.021345 when starting from 1.5
## MLE = 3.021345 when starting from 4
## MLE = 3.021345 when starting from 4.7
## MLE = 3.021345 when starting from 7
## MLE = 3.021345 when starting from 8
## MLE = 3.021345 when starting from 38
```

```
## Refine the estimate by running Newton-Raphson method
newton(-0.5914735)
```

```
## [1] -0.5914735
```

```
newton(3.021345)
```

```
## [1] 3.021345
```

```
## MLE = -0.5914735 when starting from -0.5914735
## MLE = 3.021345 when starting from 3.021345
```

(e)

Comment From Q1 part b we know that not all θ converged by using Newton method. From part c we know that the fixed-point method is slower than Newton method because it has one more variable- α.

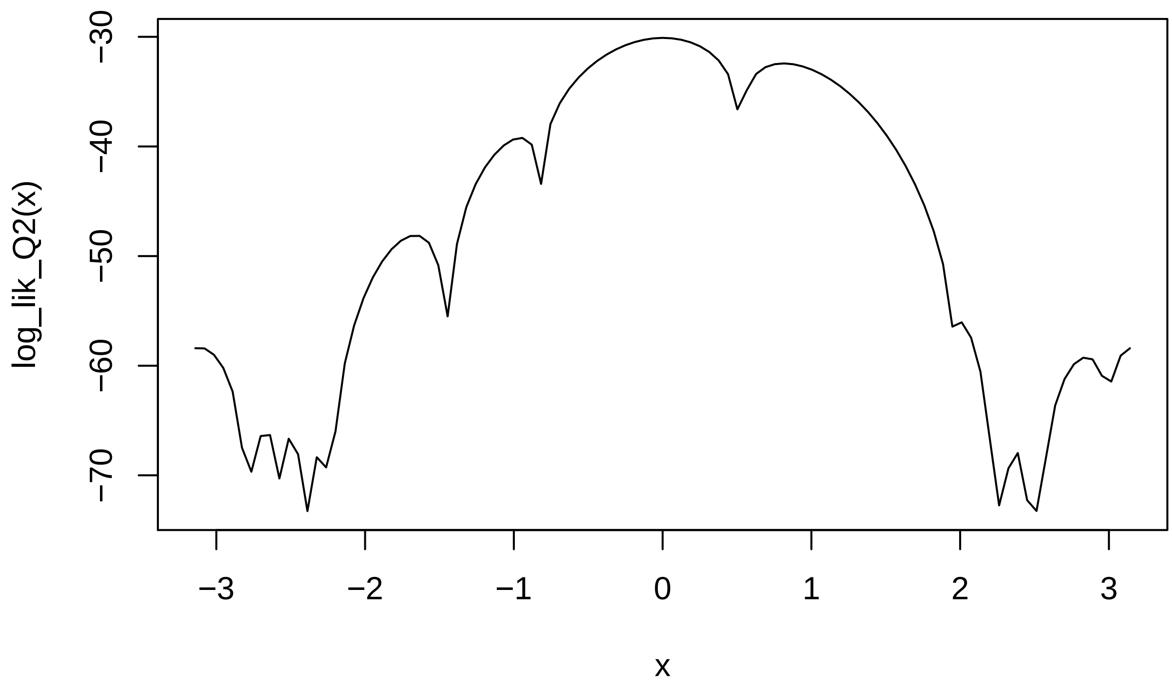
Question 2

(a)

The log-likelihood function is:

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

```
## The graph of the log-likelihood function
# Sample data
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
log_lik_Q2<-function(theta, par=x){
  n <- length(x)
  ll <- sapply(theta, function(theta_6) sum(log(1 - cos(x - theta_6))) - n * log(2 * pi))
  return(ll)
}
curve(log_lik_Q2, -pi, pi) # Graph the function with (-pi < theta < pi)
```



##(b)

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

Using intergration by part:

$$E[X|\theta] = \pi - \frac{1}{2\pi}[x\sin(x-\theta)|_0^{2\pi} + \int_0^{2\pi}\sin(x-\theta)dx] = \pi - \sin(2\pi-\theta) - \frac{1}{2\pi}[\cos(x-\theta)]|_0^{2\pi} = \pi - \sin(2\pi-\theta) - \frac{1}{2\pi}[\cos(2\pi-\theta) - \cos(\theta)] =$$

Then we have,

$$E[X|\theta] = \pi - \sin(\theta) = \bar{x}$$

So,

$$\sin(\theta) = \pi - \bar{x}\theta = \arcsin(\pi - \bar{x})$$

$$\hat{\theta}_{\text{moment}} = \arcsin(\pi - \bar{x})$$

```
sample_mean<- mean(x)
moment<- asin(pi-sample_mean)
```

The moment estimator $\hat{\theta}_{\text{moment}}$ = -0.09539407 ##(c)

```
# Function of l^{'}( \theta)
log_lik_Q2_1<- function(theta_7){
  -1*sum((sin(x-theta_7)) / (1-cos(x-theta_7)))
}
# Function of l^{''}( \theta)
log_lik_Q2_2<- function(theta_8){
```

```r
  -1*sum(1 / (1-cos(x-theta_8)))
}

# Newton method
newton_2<- function(th){
theta_9<- array()
theta_9[1]<- th # start point theta
i<- 1
difference<- 10
while(abs(difference)> 10^(-10) & i<1000){
  theta_9[i+1]<- theta_9[i]-log_lik_Q2_1(theta_9[i])/log_lik_Q2_2(theta_9[i])
  difference<- theta_9[i+1]-theta_9[i]
  i=i+1
}
print(theta_9[i])
}

newton_2(-0.09539407)
```

```
## [1] 0.003118157
```

MLE for

$$\theta$$

using the Newton-Raphson method with

$$\theta_0$$

=

$$\hat{\theta}_{\text{moment}} =$$

-0.09539407 is 0.003118157 ##(d)

```r
newton_2(-2.7)
```

```
## [1] -2.668857
```

```r
newton_2(2.7)
```

```
## [1] 2.848415
```

MLE of

$$\theta_0$$

= -2.7 is -2.668857 and MLE of

$$\theta_0$$

= 2.7 is 2.848415. the MLE is very close to the start point. ##(e)

```r
starting_value_200 <- seq(-pi,pi,length.out = 200)
a_200 <- array()
for (i in 1:length(starting_value_200)){
  a_200[i] <- newton_2(starting_value_200[i])
}
```

```
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
```

```
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -3.112471
## [1] -2.786557
## [1] -2.786557
## [1] -2.668857
## [1] -2.668857
## [1] -2.668857
## [1] -2.668857
## [1] -2.668857
## [1] -2.509356
## [1] -2.509356
## [1] -2.509356
## [1] -2.509356
## [1] -2.509356
## [1] -2.509356
## [1] -2.388267
## [1] -2.297926
## [1] -2.297926
## [1] -2.297926
## [1] -2.297926
## [1] -2.232192
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.662712
## [1] -1.447503
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
```

```
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] -0.9544058
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.003118157
```

```
## [1] 0.003118157
## [1] 0.003118157
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 0.8126374
## [1] 2.007223
## [1] 2.007223
## [1] 2.007223
## [1] 2.007223
## [1] 2.007223
## [1] 2.007223
```

```
## [1] 2.007223
## [1] 2.007223
## [1] 2.237013
## [1] 2.237013
## [1] 2.374712
## [1] 2.374712
## [1] 2.374712
## [1] 2.374712
## [1] 2.374712
## [1] 2.374712
## [1] 2.48845
## [1] 2.48845
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 2.848415
## [1] 3.170715
## [1] 3.170715
## [1] 3.170715
## [1] 3.170715
## [1] 3.170715
```

```
mx<-cbind(starting_value_200,a_200)
print(mx)
```

```
##        starting_value_200        a_200
##   [1,]        -3.14159265 -3.112470507
##   [2,]        -3.11001886 -3.112470507
##   [3,]        -3.07844506 -3.112470507
##   [4,]        -3.04687127 -3.112470507
##   [5,]        -3.01529747 -3.112470507
##   [6,]        -2.98372368 -3.112470507
##   [7,]        -2.95214988 -3.112470507
##   [8,]        -2.92057608 -3.112470507
##   [9,]        -2.88900229 -3.112470507
##  [10,]        -2.85742849 -3.112470507
##  [11,]        -2.82585470 -3.112470507
##  [12,]        -2.79428090 -2.786556852
##  [13,]        -2.76270711 -2.786556852
##  [14,]        -2.73113331 -2.668857459
##  [15,]        -2.69955952 -2.668857459
##  [16,]        -2.66798572 -2.668857459
##  [17,]        -2.63641193 -2.668857459
##  [18,]        -2.60483813 -2.668857459
```

```
## [19,]        -2.57326433 -2.509356033
## [20,]        -2.54169054 -2.509356033
## [21,]        -2.51011674 -2.509356033
## [22,]        -2.47854295 -2.509356033
## [23,]        -2.44696915 -2.509356033
## [24,]        -2.41539536 -2.509356033
## [25,]        -2.38382156 -2.388266628
## [26,]        -2.35224777 -2.297925969
## [27,]        -2.32067397 -2.297925969
## [28,]        -2.28910017 -2.297925969
## [29,]        -2.25752638 -2.297925969
## [30,]        -2.22595258 -2.232191899
## [31,]        -2.19437879 -1.662712395
## [32,]        -2.16280499 -1.662712395
## [33,]        -2.13123120 -1.662712395
## [34,]        -2.09965740 -1.662712395
## [35,]        -2.06808361 -1.662712395
## [36,]        -2.03650981 -1.662712395
## [37,]        -2.00493602 -1.662712395
## [38,]        -1.97336222 -1.662712395
## [39,]        -1.94178842 -1.662712395
## [40,]        -1.91021463 -1.662712395
## [41,]        -1.87864083 -1.662712395
## [42,]        -1.84706704 -1.662712395
## [43,]        -1.81549324 -1.662712395
## [44,]        -1.78391945 -1.662712395
## [45,]        -1.75234565 -1.662712395
## [46,]        -1.72077186 -1.662712395
## [47,]        -1.68919806 -1.662712395
## [48,]        -1.65762426 -1.662712395
## [49,]        -1.62605047 -1.662712395
## [50,]        -1.59447667 -1.662712395
## [51,]        -1.56290288 -1.662712395
## [52,]        -1.53132908 -1.662712395
## [53,]        -1.49975529 -1.662712395
## [54,]        -1.46818149 -1.662712395
## [55,]        -1.43660770 -1.447502553
## [56,]        -1.40503390 -0.954405837
## [57,]        -1.37346010 -0.954405837
## [58,]        -1.34188631 -0.954405837
## [59,]        -1.31031251 -0.954405837
## [60,]        -1.27873872 -0.954405837
## [61,]        -1.24716492 -0.954405837
## [62,]        -1.21559113 -0.954405837
## [63,]        -1.18401733 -0.954405837
## [64,]        -1.15244354 -0.954405837
## [65,]        -1.12086974 -0.954405837
## [66,]        -1.08929595 -0.954405837
## [67,]        -1.05772215 -0.954405837
## [68,]        -1.02614835 -0.954405837
## [69,]        -0.99457456 -0.954405837
## [70,]        -0.96300076 -0.954405837
## [71,]        -0.93142697 -0.954405837
## [72,]        -0.89985317 -0.954405837
```

```
##  [73,]        -0.86827938 -0.954405837
##  [74,]        -0.83670558 -0.954405837
##  [75,]        -0.80513179  0.003118157
##  [76,]        -0.77355799  0.003118157
##  [77,]        -0.74198419  0.003118157
##  [78,]        -0.71041040  0.003118157
##  [79,]        -0.67883660  0.003118157
##  [80,]        -0.64726281  0.003118157
##  [81,]        -0.61568901  0.003118157
##  [82,]        -0.58411522  0.003118157
##  [83,]        -0.55254142  0.003118157
##  [84,]        -0.52096763  0.003118157
##  [85,]        -0.48939383  0.003118157
##  [86,]        -0.45782003  0.003118157
##  [87,]        -0.42624624  0.003118157
##  [88,]        -0.39467244  0.003118157
##  [89,]        -0.36309865  0.003118157
##  [90,]        -0.33152485  0.003118157
##  [91,]        -0.29995106  0.003118157
##  [92,]        -0.26837726  0.003118157
##  [93,]        -0.23680347  0.003118157
##  [94,]        -0.20522967  0.003118157
##  [95,]        -0.17365588  0.003118157
##  [96,]        -0.14208208  0.003118157
##  [97,]        -0.11050828  0.003118157
##  [98,]        -0.07893449  0.003118157
##  [99,]        -0.04736069  0.003118157
## [100,]        -0.01578690  0.003118157
## [101,]         0.01578690  0.003118157
## [102,]         0.04736069  0.003118157
## [103,]         0.07893449  0.003118157
## [104,]         0.11050828  0.003118157
## [105,]         0.14208208  0.003118157
## [106,]         0.17365588  0.003118157
## [107,]         0.20522967  0.003118157
## [108,]         0.23680347  0.003118157
## [109,]         0.26837726  0.003118157
## [110,]         0.29995106  0.003118157
## [111,]         0.33152485  0.003118157
## [112,]         0.36309865  0.003118157
## [113,]         0.39467244  0.003118157
## [114,]         0.42624624  0.003118157
## [115,]         0.45782003  0.003118157
## [116,]         0.48939383  0.003118157
## [117,]         0.52096763  0.812637417
## [118,]         0.55254142  0.812637417
## [119,]         0.58411522  0.812637417
## [120,]         0.61568901  0.812637417
## [121,]         0.64726281  0.812637417
## [122,]         0.67883660  0.812637417
## [123,]         0.71041040  0.812637417
## [124,]         0.74198419  0.812637417
## [125,]         0.77355799  0.812637417
## [126,]         0.80513179  0.812637417
```

```
## [127,]         0.83670558  0.812637417
## [128,]         0.86827938  0.812637417
## [129,]         0.89985317  0.812637417
## [130,]         0.93142697  0.812637417
## [131,]         0.96300076  0.812637417
## [132,]         0.99457456  0.812637417
## [133,]         1.02614835  0.812637417
## [134,]         1.05772215  0.812637417
## [135,]         1.08929595  0.812637417
## [136,]         1.12086974  0.812637417
## [137,]         1.15244354  0.812637417
## [138,]         1.18401733  0.812637417
## [139,]         1.21559113  0.812637417
## [140,]         1.24716492  0.812637417
## [141,]         1.27873872  0.812637417
## [142,]         1.31031251  0.812637417
## [143,]         1.34188631  0.812637417
## [144,]         1.37346010  0.812637417
## [145,]         1.40503390  0.812637417
## [146,]         1.43660770  0.812637417
## [147,]         1.46818149  0.812637417
## [148,]         1.49975529  0.812637417
## [149,]         1.53132908  0.812637417
## [150,]         1.56290288  0.812637417
## [151,]         1.59447667  0.812637417
## [152,]         1.62605047  0.812637417
## [153,]         1.65762426  0.812637417
## [154,]         1.68919806  0.812637417
## [155,]         1.72077186  0.812637417
## [156,]         1.75234565  0.812637417
## [157,]         1.78391945  0.812637417
## [158,]         1.81549324  0.812637417
## [159,]         1.84706704  0.812637417
## [160,]         1.87864083  0.812637417
## [161,]         1.91021463  0.812637417
## [162,]         1.94178842  0.812637417
## [163,]         1.97336222  2.007223238
## [164,]         2.00493602  2.007223238
## [165,]         2.03650981  2.007223238
## [166,]         2.06808361  2.007223238
## [167,]         2.09965740  2.007223238
## [168,]         2.13123120  2.007223238
## [169,]         2.16280499  2.007223238
## [170,]         2.19437879  2.007223238
## [171,]         2.22595258  2.237012923
## [172,]         2.25752638  2.237012923
## [173,]         2.28910017  2.374711666
## [174,]         2.32067397  2.374711666
## [175,]         2.35224777  2.374711666
## [176,]         2.38382156  2.374711666
## [177,]         2.41539536  2.374711666
## [178,]         2.44696915  2.374711666
## [179,]         2.47854295  2.488449651
## [180,]         2.51011674  2.488449651
```

```
## [181,]         2.54169054  2.848415325
## [182,]         2.57326433  2.848415325
## [183,]         2.60483813  2.848415325
## [184,]         2.63641193  2.848415325
## [185,]         2.66798572  2.848415325
## [186,]         2.69955952  2.848415325
## [187,]         2.73113331  2.848415325
## [188,]         2.76270711  2.848415325
## [189,]         2.79428090  2.848415325
## [190,]         2.82585470  2.848415325
## [191,]         2.85742849  2.848415325
## [192,]         2.88900229  2.848415325
## [193,]         2.92057608  2.848415325
## [194,]         2.95214988  2.848415325
## [195,]         2.98372368  2.848415325
## [196,]         3.01529747  3.170714800
## [197,]         3.04687127  3.170714800
## [198,]         3.07844506  3.170714800
## [199,]         3.11001886  3.170714800
## [200,]         3.14159265  3.170714800
```

```
# The set of starting values are divided into 18 groups
Group.1 <- mx[c(1:11), 1]
Group.2 <- mx[c(12:13), 1]
Group.3 <- mx[c(14:18), 1]
Group.4 <- mx[c(19:24), 1]
Group.5 <- mx[25, 1]
Group.6 <- mx[c(26:29), 1]
Group.7 <- mx[30, 1]
Group.8 <- mx[c(31:54), 1]
Group.9 <- mx[55, 1]
Group.10 <- mx[c(56:74), 1]
Group.11 <- mx[c(75:116), 1]
Group.12 <- mx[c(117:162), 1]
Group.13 <- mx[c(163:170), 1]
Group.14 <- mx[c(171:172), 1]
Group.15 <- mx[c(173:178), 1]
Group.16 <- mx[c(179:180), 1]
Group.17 <- mx[c(181:195), 1]
Group.18 <- mx[c(196:200), 1]
```

Queation 3

(a)

```
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))
N_0<- 2
k<- 1500 # we set k equals to 1500
r_t<- log((beetles$beetles*(k-2))/(k - beetles$beetles)*2)
r<- (r_t/beetles$days)
```

```r
r # r is a set which has 10 numbers with first number inf
```

```
##  [1]        Inf 0.57172441 0.21736694 0.15668594 0.12782607 0.12173043
##  [7] 0.09366955 0.07178974 0.06908107 0.05695511
```

```r
mean(r[2:10]) # mean is 0.1652033
```

```
## [1] 0.1652033
```

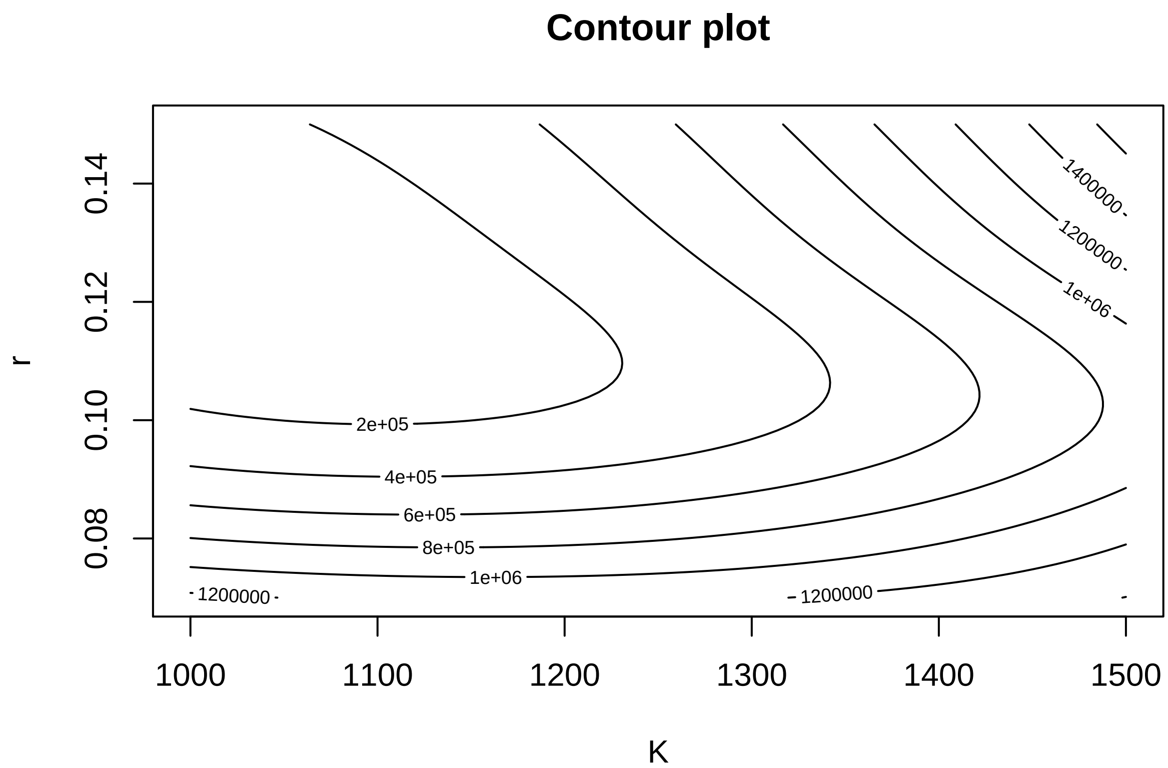
```r
N_t<- function(k,r,t){
  N_0*k/(N_0+(k-N_0)*exp((-r)*t))# t is days
}

bb <- nls(beetles~N_t(k,r,days),data = beetles,start=list(k=1500,r=0.1652033),trace=TRUE)
```

```
## 1980051 :  1500.0000000    0.1652033
## 117991 :  997.5137294   0.1394446
## 78622.19 :  1019.990913    0.123586
## 73557.02 :  1044.6290515    0.1190586
## 73423.02 :  1048.865916    0.118403
## 73419.79 :  1049.3219774    0.1182912
## 73419.7 :  1049.3929905    0.1182723
## 73419.7 :  1049.4048400    0.1182691
## 73419.7 :  1049.4068376    0.1182685
```

From the form, we can find out initial value of k is 1500, r is 0.1652033. And optimized value of k is 1049.4068376, r is 0.1182685. ##(b)

```r
k <- seq(1000,1500, length.out = 100)
r <- seq(0.07,0.15, length.out = 100)
days = c(0, 8, 28, 41, 63, 69, 97, 117, 135, 154)
beetles = c(2, 47, 192, 256, 768, 896, 1120, 896, 1184, 1024)
sum_error <- function(k,r){
  sum((beetles-2*k/(2+(k-2)*exp(-r*days)))^2)
}
z <- matrix(rep(0,10000),nrow = 100)
a<- 1
b<- 1
for (a in 1:100){
  for(b in 1:100){
    z[a,b] <- sum_error(k[a],r[b])
  }
}
contour(k, r, z, xlab = 'K', ylab = 'r', plot_title = title ("Contour plot"))
```



Contour plot
r
0.08
0.10
0.12
0.14
K
1000
1100
1200
1300
1400
1500
2e+05
4e+05
6e+05
8e+05
1e+06
1200000
1200000
1e+06
1200000
1400000