HW4

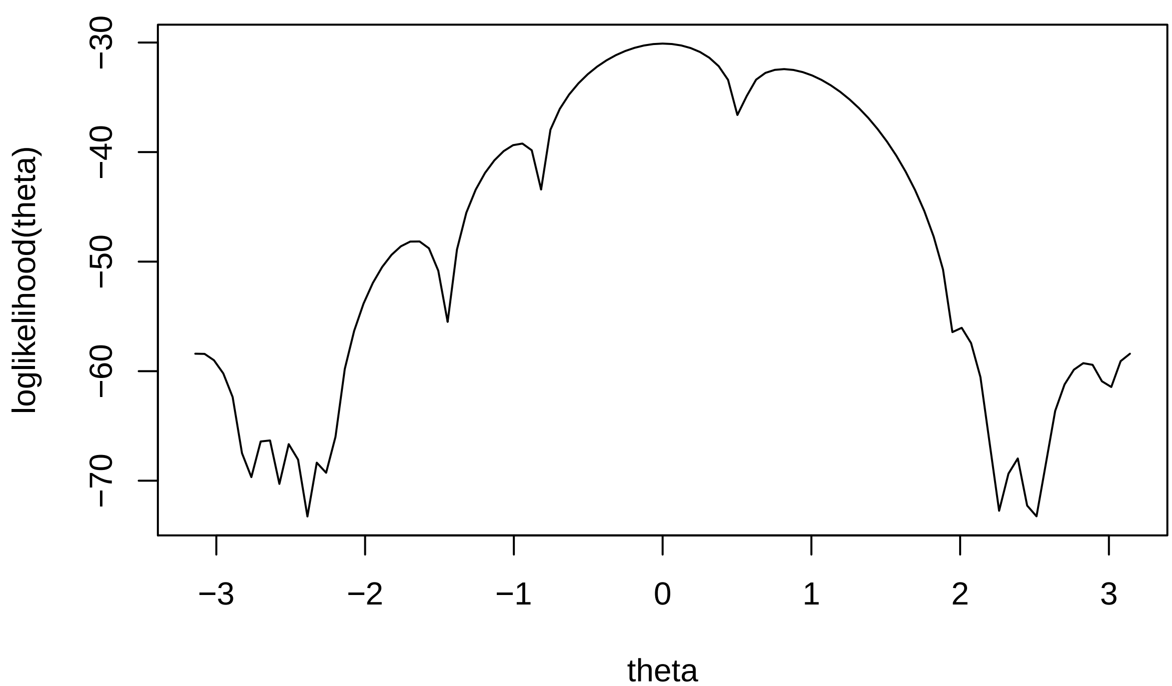
Cheng Huang[*]

28 September 2018

Many local maxima

loglikihood

```
loglkh <- function(theta){
  sum(log((1 - cos(x - theta))/(2 * pi), base = exp(1)))
}
loglikelihood <- Vectorize(loglkh)
curve(loglikelihood, -pi, pi, xname = 'theta')
```



MOM

$$E(X|\theta) = \int_{x=0}^{2\pi} \frac{1 - cos(x-\theta)}{2\pi} x dx = \int_{x=0}^{2\pi} \frac{x}{2\pi} dx - \frac{1}{2\pi} \int_0^{2\pi} x cos(x-\theta) dx = \pi - \frac{1}{2\pi}(x sin(x-\theta) + cos(x-\theta))|_0^{2\pi}$$

*; Ph.D. student at Department of Statistics, University of Connecticut.

$$= \pi - \frac{1}{2\pi}(-2\pi sin(\theta)) = \pi + sin(\theta) = \bar{X} = 3.236842$$

```r
asin(mean(x)-pi)
```

```
## [1] 0.09539407
```

So $\theta = 0.09539407$.

MLE using Newton-Raphson

```r
loglkh.1 <- function(theta){
  sum(sin(theta-x)/(1-cos(theta-x)))
}
newtonRaphson(loglkh.1, asin(mean(x)-pi), dfun = NULL)$root
```

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

start at -2.7 and 2.7

```r
newtonRaphson(loglkh.1, -2.7, dfun = NULL)$root
```

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

```r
newtonRaphson(loglkh.1, 2.7, dfun = NULL)$root
```

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

repeat 200 using start values between -pi to pi

```r
start <- seq(-pi, pi, length.out = 200)
nr <- double(200)
for (i in start){
  nr[which(start == i)] <- newtonRaphson(loglkh.1, i, dfun = NULL)$root
}
nrtble <- data.table(start = start, MLE = nr)
data.table(cn = names(nrtble), transpose(nrtble))
```

```
##       cn        V1        V2        V3        V4        V5        V6
## 1: start -3.141593 -3.110019 -3.078445 -3.046871 -3.015297 -2.983724
## 2:   MLE -3.112471 -3.112471 -3.112471 -3.112471 -3.112471 -3.112471
##           V7        V8        V9       V10       V11       V12       V13
## 1: -2.952150 -2.920576 -2.889002 -2.857428 -2.825855 -2.794281 -2.762707
## 2: -3.112471 -3.112471 -3.112471 -3.112471 -3.112471 -2.786557 -2.786557
##          V14       V15       V16       V17       V18       V19       V20
## 1: -2.731133 -2.699560 -2.667986 -2.636412 -2.604838 -2.573264 -2.541691
## 2: -2.668857 -2.668857 -2.668857 -2.668857 -2.668857 -2.509356 -2.509356
##          V21       V22       V23       V24       V25       V26       V27
## 1: -2.510117 -2.478543 -2.446969 -2.415395 -2.383822 -2.352248 -2.320674
## 2: -2.509356 -2.509356 -2.509356 -2.509356 -2.388267 -2.297926 -2.297926
##          V28       V29       V30       V31       V32       V33       V34
## 1: -2.289100 -2.257526 -2.225953 -2.194379 -2.162805 -2.131231 -2.099657
## 2: -2.297926 -2.297926 -2.232192 -1.662712 -1.662712 -1.662712 -1.662712
##          V35       V36       V37       V38       V39       V40       V41
## 1: -2.068084 -2.036510 -2.004936 -1.973362 -1.941788 -1.910215 -1.878641
## 2: -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.662712
##          V42       V43       V44       V45       V46       V47       V48
## 1: -1.847067 -1.815493 -1.783919 -1.752346 -1.720772 -1.689198 -1.657624
## 2: -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.662712
##          V49       V50       V51       V52       V53       V54       V55
## 1: -1.626050 -1.594477 -1.562903 -1.531329 -1.499755 -1.468181 -1.436608
## 2: -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.662712 -1.447503
##           V56        V57        V58        V59        V60        V61
## 1: -1.4050339 -1.3734601 -1.3418863 -1.3103125 -1.2787387 -1.2471649
## 2: -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058
##           V62        V63        V64        V65        V66        V67
## 1: -1.2155911 -1.1840173 -1.1524435 -1.1208697 -1.0892959 -1.0577221
## 2: -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058
##           V68        V69        V70        V71        V72        V73
## 1: -1.0261484 -0.9945746 -0.9630008 -0.9314270 -0.8998532 -0.8682794
## 2: -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058 -0.9544058
##           V74          V75          V76          V77          V78
## 1: -0.8367056 -0.805131786 -0.773557990 -0.741984195 -0.710410399
## 2: -0.9544058  0.003118157  0.003118157  0.003118157  0.003118157
##             V79          V80          V81          V82          V83
## 1: -0.678836604 -0.647262808 -0.615689013 -0.584115217 -0.552541421
## 2:  0.003118157  0.003118157  0.003118157  0.003118157  0.003118157
##             V84          V85          V86          V87          V88
## 1: -0.520967626 -0.489393830 -0.457820035 -0.426246239 -0.394672444
## 2:  0.003118157  0.003118157  0.003118157  0.003118157  0.003118157
##             V89          V90          V91          V92          V93
## 1: -0.363098648 -0.331524853 -0.299951057 -0.268377262 -0.236803466
## 2:  0.003118157  0.003118157  0.003118157  0.003118157  0.003118157
##             V94          V95          V96          V97          V98
## 1: -0.205229671 -0.173655875 -0.142082080 -0.110508284 -0.078934489
## 2:  0.003118157  0.003118157  0.003118157  0.003118157  0.003118157
```

```
##           V99        V100        V101        V102        V103
## 1: -0.047360693 -0.015786898 0.015786898 0.047360693 0.078934489
## 2:  0.003118157  0.003118157 0.003118157 0.003118157 0.003118157
##          V104        V105        V106        V107        V108        V109
## 1: 0.110508284 0.142082080 0.173655875 0.205229671 0.236803466 0.268377262
## 2: 0.003118157 0.003118157 0.003118157 0.003118157 0.003118157 0.003118157
##          V110        V111        V112        V113        V114        V115
## 1: 0.299951057 0.331524853 0.363098648 0.394672444 0.426246239 0.457820035
## 2: 0.003118157 0.003118157 0.003118157 0.003118157 0.003118157 0.003118157
##          V116      V117      V118      V119      V120      V121      V122
## 1: 0.489393830 0.5209676 0.5525414 0.5841152 0.6156890 0.6472628 0.6788366
## 2: 0.003118157 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V123      V124      V125      V126      V127      V128      V129
## 1: 0.7104104 0.7419842 0.7735580 0.8051318 0.8367056 0.8682794 0.8998532
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V130      V131      V132      V133      V134      V135      V136
## 1: 0.9314270 0.9630008 0.9945746 1.0261484 1.0577221 1.0892959 1.1208697
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V137      V138      V139      V140      V141      V142      V143
## 1: 1.1524435 1.1840173 1.2155911 1.2471649 1.2787387 1.3103125 1.3418863
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V144      V145      V146      V147      V148      V149      V150
## 1: 1.3734601 1.4050339 1.4366077 1.4681815 1.4997553 1.5313291 1.5629029
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V151      V152      V153      V154      V155      V156      V157
## 1: 1.5944767 1.6260505 1.6576243 1.6891981 1.7207719 1.7523457 1.7839194
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374
##        V158      V159      V160      V161      V162     V163     V164
## 1: 1.8154932 1.8470670 1.8786408 1.9102146 1.9417884 1.973362 2.004936
## 2: 0.8126374 0.8126374 0.8126374 0.8126374 0.8126374 2.007223 2.007223
##       V165     V166     V167     V168     V169     V170     V171     V172
## 1: 2.036510 2.068084 2.099657 2.131231 2.162805 2.194379 2.225953 2.257526
## 2: 2.007223 2.007223 2.007223 2.007223 2.007223 2.007223 2.237013 2.237013
##       V173     V174     V175     V176     V177     V178     V179     V180
## 1: 2.289100 2.320674 2.352248 2.383822 2.415395 2.446969 2.478543 2.510117
## 2: 2.374712 2.374712 2.374712 2.374712 2.374712 2.374712 2.488450 2.488450
##       V181     V182     V183     V184     V185     V186     V187     V188
## 1: 2.541691 2.573264 2.604838 2.636412 2.667986 2.699560 2.731133 2.762707
## 2: 2.848415 2.848415 2.848415 2.848415 2.848415 2.848415 2.848415 2.848415
##       V189     V190     V191     V192     V193     V194     V195     V196
## 1: 2.794281 2.825855 2.857428 2.889002 2.920576 2.952150 2.983724 3.015297
## 2: 2.848415 2.848415 2.848415 2.848415 2.848415 2.848415 2.848415 3.170715
##       V197     V198     V199     V200
## 1: 3.046871 3.078445 3.110019 3.141593
## 2: 3.170715 3.170715 3.170715 3.170715
```

Modeling beetle data

```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))
```

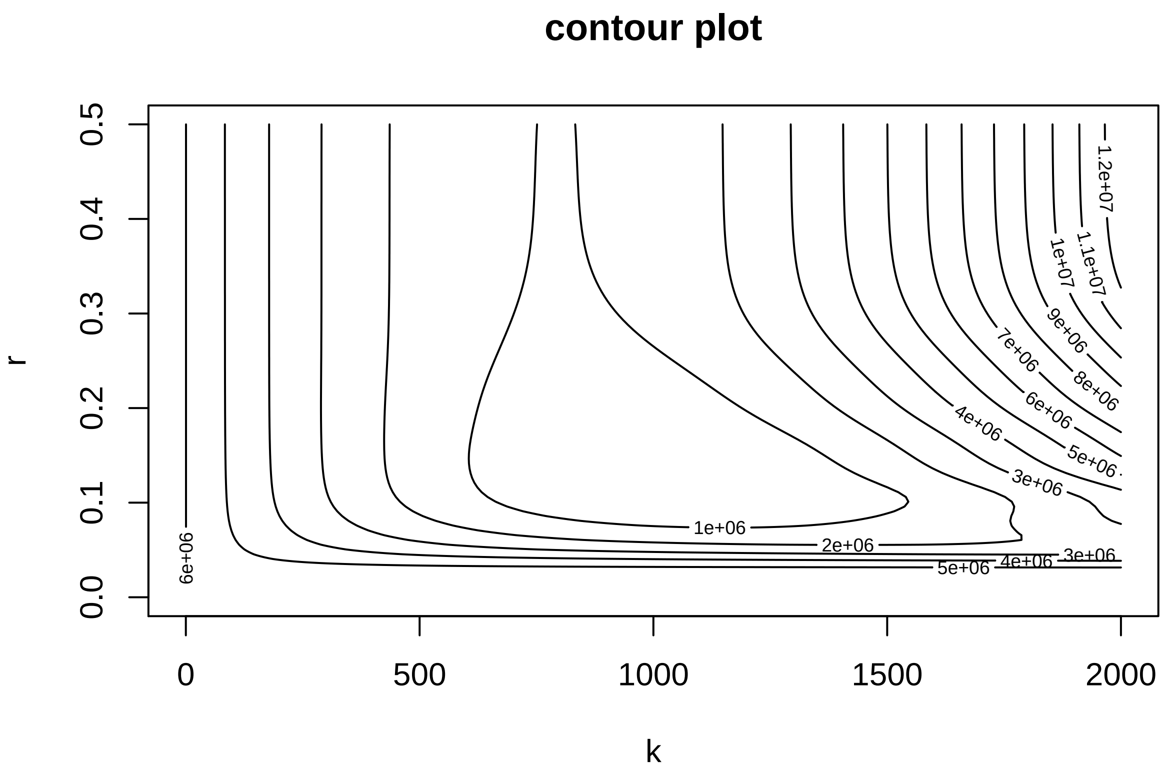
Gauss-Newton and Contour Plot

```r
library(pracma) ## gauss newton method
library(plotly) ## plot contour
t <- beetles$days
b <- beetles$beetles
N0 <- b[1]

Nt <- function(x){
  (x[1] * (N0) / (N0 + (x[1] - N0) * exp(-x[2] * t)) - b)
}
gaussNewton(c(1000, 1), Nt)
```

```
## $xs
## [1] 1049.4072441    0.1182684
##
## $fs
## [1] 73419.7
##
## $niter
## [1] 8
##
## $relerr
## [1] 7.275958e-11
```

```r
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")
```



MLE

```r
logb <- log(b, base = exp(1))
logb
```

```
##  [1] 0.6931472 3.8501476 5.2574954 5.5451774 6.6437897 6.7979404 7.0210840
##  [8] 6.7979404 7.0766538 6.9314718
```

```r
t
```

```
##  [1]   0   8  28  41  63  69  97 117 135 154
```

```r
llk <- function(x){
  k <- x[1]
  r <- x[2]
  sigmasq <- x[3]
  -sum(-(log(2 * pi * (sigmasq)) / 2 )- (logb - log((k * N0)/(N0 + (k - N0) * exp(-r * t)))) ^
}
llk(c(1e3, .2, 1))
```

```
## [1] 11.4255
```

```r
rs <- optim(c(1000, 0.2, 1), llk, method = "BFGS", hessian = TRUE)
rs
```

```
## $par
## [1] 954.3360540   0.1781541   0.4253134
##
## $value
## [1] 9.915059
##
## $counts
## function gradient
##       53       25
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##               [,1]          [,2]          [,3]
## [1,]  1.472245e-05  5.355685e-02 -0.0040449715
## [2,]  5.355685e-02  6.409873e+02 -0.0008940906
## [3,] -4.044971e-03 -8.940906e-04 27.6453070223
```

```r
solve(rs$hessian)
```

```
##               [,1]         [,2]        [,3]
## [1,] 103565.245366 -8.653235974 15.15305023
## [2,]     -8.653236  0.002283101 -0.00126604
## [3,]     15.153050 -0.001266040  0.03838961
```

Reference

https://stackoverflow.com/questions/19079152/contour-plot-of-a-custom-function-in-r [jun-yan/stat-5361]https://github.com/jun-yan/stat-5361