Modeling beetle data

Guanting Wei

Sep.26.2018

1. Squared errors

```
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))
```

$$N_t = f(t) = \frac{KN_0}{N_0 + (K - N_0)e^{-rt}}$$

According to the data, $N_0 = 2$, then

$$N_t = f(t) = \frac{2K}{2 + (K - 2)e^{-rt}}$$

Squared Errors:

$$SE = \sum[N_t - \frac{2K}{2 + (K - 2)e^{-rt}}]^2$$

In order solve the problem, we need to find range of K and r. We already know K is parameter that represents the population carrying capacity of the environment, so K should be bigger thanN_t.Then I choose K ranging from 1300 to 1600 and then we decide r by using N_t,t and K

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

```
t=beetles$days
n=beetles$beetles
r=function(t,k,n){
  r=1/t*log(n*(k-2)/(2*(k-n)))
}
k=1300:1600
a=matrix(0,length(k),10)
for(m in 1:length(k)){
  for(i in 2:10){
  a[m,i]=r(t[i],k[m],n[i])
  }
}
max=max(a[a>0])
min=min(a[a>0])
max
```

```
## [1] 0.3990355
```

```
min
```

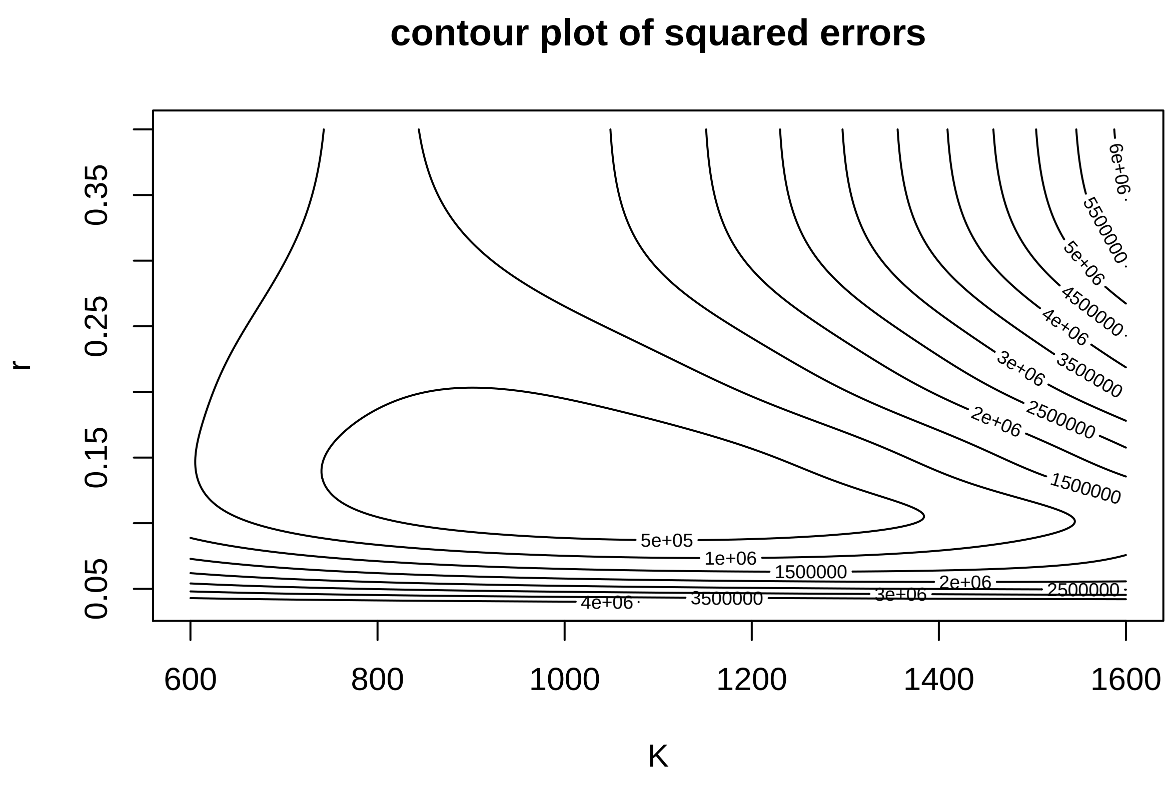
```
## [1] 0.04713458
```

Then we choose t from 0.04 to 0.4. However, N_t decreases after 97 days, so the environment is overloaded. As a result, we change K to range from 600 to 1600

```
k=600:1600
r=seq(0.04,0.4,length=1000)
```

Plot the contour plot depending on K and r above:

```
se=function(k,r){
  se=0
  for(i in 1:length(beetles$days)){
    se=se+(n[i]-(2*k)/(2+(k-2)*exp(-r*t[i])))^2
  }
  se
}
b=matrix(0,length(k),length(r))
for(m in 1:length(k)){
  for(i in 1:length(r)){
    b[m,i]=se(k[m],r[i])
  }
}
contour(k,r,b,xlab="K",ylab="r",main="contour plot of squared errors")
```



2. MLE

We assume that$\log N_t \sim N(\log f(t), \sigma^2)$ The Likelihood function is:

$$L(k, r, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\log N_i - \log \frac{2K}{2+(K-2)e^{-rt}})^2}{2\sigma^2}}$$

The Loglikelihood function is:

$$l(k, r, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(\log N_i - \log \frac{2K}{2+(K-2)e^{-rt}})^2}{2\sigma^2}$$

```
m=function(x){
  k=x[1]
  r=x[2]
  sigma_2=x[3]
  sum=0
  for(i in 1:10){
  sum=sum+(-(log(2*pi*sigma_2))/2-(log(n[i])-log((2*k)/(2+(k-2)*exp(-r*t[i]))))^2/(2*sigma_2))
  }
  -sum
}
options(warn=-1)
opt=optim(c(800,0.2,4),m,method="BFGS",hessian=TRUE)
opt
```

```
## $par
## [1] 799.9972095   0.1958194   0.4151383
##
## $value
## [1] 9.793501
##
## $counts
## function gradient
##       42       14
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##              [,1]          [,2]          [,3]
## [1,] 2.572564e-05  5.421957e-02  0.0008013523
## [2,] 5.421957e-02  3.236987e+02 -0.0007279626
## [3,] 8.013523e-04 -7.279626e-04 29.0115198278
```

```
k0=opt$par[1]
r0=opt$par[2]
sigma0_2=opt$par[3]
var=diag(solve(opt$hessian))
```

$\theta = (r, K, \sigma^2) = (0.1958194, 800, 0.4151383)$
$var(k) = 6.016225e + 04$

$var(r) = 4.777223e - 03$
$var(\sigma^2) = 3.451498e - 02$