STAT40180 — Stochastic Models

Brendan Murphy

Week 2

Practical Inference

Practical Statistical Inference

- In this class, we will recap on maximum likelihood inference.
- The emphasis will be on using computational methods.
- We will consider the problems of:
    - Estimation
    - Interval Estimation
    - Testing

Time-to-Event: Kiama Blowhole

- Let's consider a time-to-event dataset.
- Spectacular eruptions of water are formed through a hole in the cliff at Kiama, about 120km south of Sydney, Australia.
- The times at which 64 successive eruptions occurred were observed.

Time-to-Event: Kiama Blowhole

- Let's consider a time-to-event dataset.
- Spectacular eruptions of water are formed through a hole in the cliff at Kiama, about 120km south of Sydney, Australia.
- The times at which 64 successive eruptions occurred were observed.

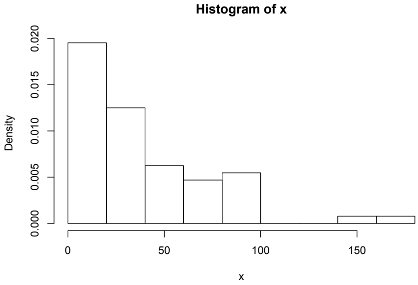
- The data are:

83	51	87	60	28	95	8	27	15	10	18	16	29	54	91	8
17	55	10	35	47	77	36	17	21	36	18	40	10	7	34	27
28	56	8	25	68	146	89	18	73	69	9	37	10	82	29	8
60	61	61	18	169	25	8	26	11	83	11	42	17	14	9	12

Time-to-Event: Model

- A potential model for the data is the exponential model

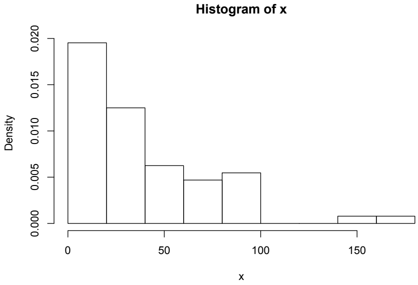
$$f(x) = \theta \exp(-\theta x), \text{ where } \theta > 0.$$



Time-to-Event: Model

- A potential model for the data is the exponential model

$$f(x) = \theta \exp(-\theta x), \text{ where } \theta > 0.$$



- The exponential can be easily fitted using maximum likelihood.

Maximum likelihood

- The likelihood is of the form

$$L(\theta) = \prod_{i=1}^{n} f(x_i)$$

Maximum likelihood

- The likelihood is of the form

$$L(\theta) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \theta \exp(-\theta x_i)$$

Maximum likelihood

- The likelihood is of the form

$$L(\theta) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \theta \exp(-\theta x_i) = \theta^n \exp\left(-\theta \sum_{i=1}^{n} x_i\right)$$

Maximum likelihood

- The likelihood is of the form

$$L(\theta) = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \theta \exp(-\theta x_i) = \theta^n \exp\left(-\theta \sum_{i=1}^{n} x_i\right)$$

- Thus, the log-likelihood is

$$\ell(\theta) = n \log \theta - \theta \sum_{i=1}^{n} x_i.$$

- This is straightforward to maximize.
- We compute

$$S(\theta) = \frac{\partial}{\partial \theta} \ell(\theta) = \frac{n}{\theta} - \sum_{i=1}^{n} x_i$$

Maximum likelihood 2

- Solving

$$S(\theta) = 0$$

gives

$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} x_i} = 1/\overline{x}.$$

- We check if the second derivative is negative when $\theta = \hat{\theta}$.
- Equivalently, we check if minus the second derivative is positive.

Maximum likelihood 2

- Solving
$$S(\theta) = 0$$
gives
$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} x_i} = 1/\overline{x}.$$
- We check if the second derivative is negative when $\theta = \hat{\theta}$.
- Equivalently, we check if minus the second derivative is positive.
$$I(\theta) = -\frac{\partial^2}{\partial\theta^2}\ell(\theta) = -\frac{\partial}{\partial\theta}S(\theta) = \frac{n}{\theta^2}.$$

Maximum likelihood 2

- Solving

$$S(\theta) = 0$$

gives

$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} x_i} = 1/\overline{x}.$$

- We check if the second derivative is negative when $\theta = \hat{\theta}$.
- Equivalently, we check if minus the second derivative is positive.

$$I(\theta) = -\frac{\partial^2}{\partial \theta^2}\ell(\theta) = -\frac{\partial}{\partial \theta}S(\theta) = \frac{n}{\theta^2}.$$

- Clearly,

$$I(\hat{\theta}) > 0.$$

Maximum likelihood 3

- The estimator of θ is
$$\hat{\theta} = 1/\overline{x}.$$
- The approximate standard error is
$$SE(\hat{\theta}) = \sqrt{1/I(\hat{\theta})} = 1/\sqrt{I(\hat{\theta})}.$$

Maximum likelihood 3

- The estimator of θ is
$$\hat{\theta} = 1/\overline{x}.$$
- The approximate standard error is
$$SE(\hat{\theta}) = \sqrt{1/I(\hat{\theta})} = 1/\sqrt{I(\hat{\theta})}.$$
- Thus, an approximate 95% confidence interval would be
$$\hat{\theta} \pm \frac{1.96}{\sqrt{I(\hat{\theta})}}.$$

Maximum likelihood 3

- The estimator of θ is
$$\hat{\theta} = 1/\overline{x}.$$
- The approximate standard error is
$$SE(\hat{\theta}) = \sqrt{1/I(\hat{\theta})} = 1/\sqrt{I(\hat{\theta})}.$$
- Thus, an approximate 95% confidence interval would be
$$\hat{\theta} \pm \frac{1.96}{\sqrt{I(\hat{\theta})}}.$$
- For the Kiama blowhole data we get
$$\hat{\theta} \pm \frac{1.96}{\sqrt{I(\hat{\theta})}} = 0.025 \pm 0.006$$

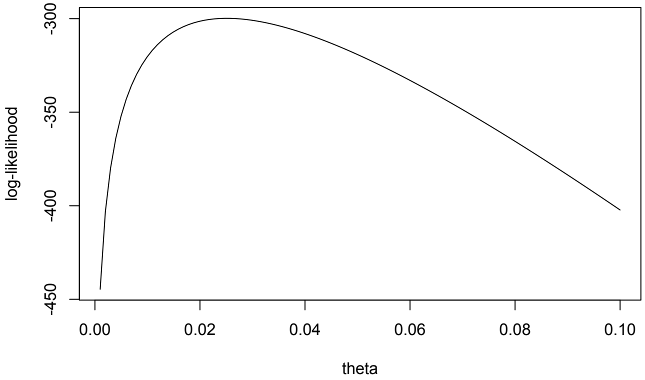
Maximum Likelihood: Code

- The code for doing the maximum likelihood estimation and inference.

```
x <- scan()
83  51  87  60  28  95   8  27  15  10  18  16  29  54  91   8
17  55  10  35  47  77  36  17  21  36  18  40  10   7  34  27
28  56   8  25  68 146  89  18  73  69   9  37  10  82  29   8
60  61  61  18 169  25   8  26  11  83  11  42  17  14   9  12

hist(x,probability=TRUE)

n <- length(x)
thetahat <- 1/mean(x)
se <- sqrt(thetahat^2/n)

UB <- thetahat+1.96*se
LB <- thetahat-1.96*se
```

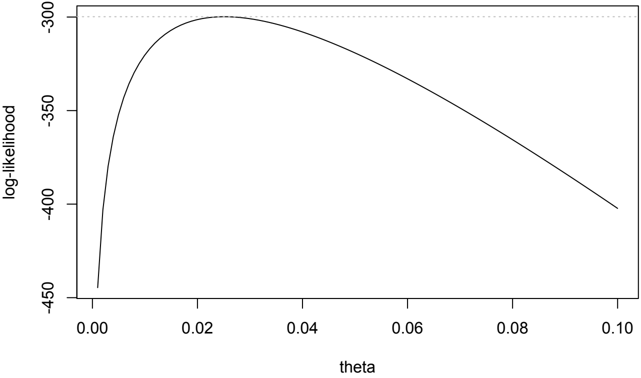
Maximum likelihood 4a

- This is how maximum likelihood works (in pictures).
- This is the likelihood function



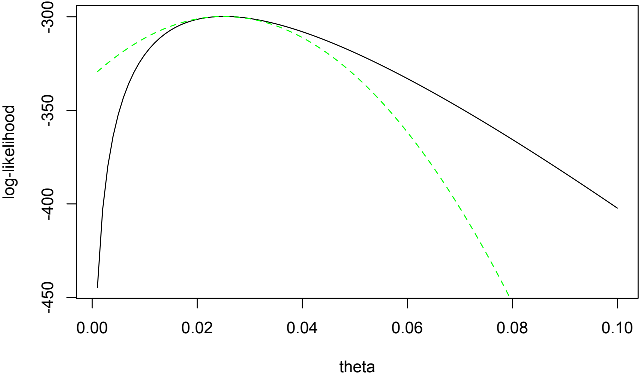
Maximum likelihood 4b

- We find where the function is flat ($S(\theta) = 0$).



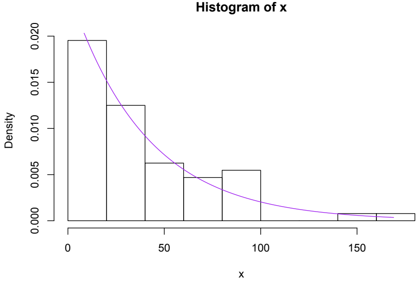
Maximum likelihood 4c

- We check that the function bends downwards at this point ($I(\hat{\theta}) > 0$)



Model Fit

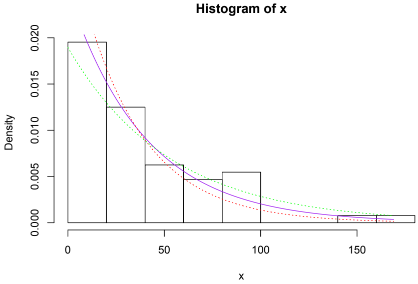
- We can informally compare the model fit to the data histogram.



- The model is fitting the data well.

Model Fit

- Adding the fits at the upper and lower end of the confidence interval helps assess fit.



Maximum Likelihood: Code

- The code for adding the fit to the histogram plot.

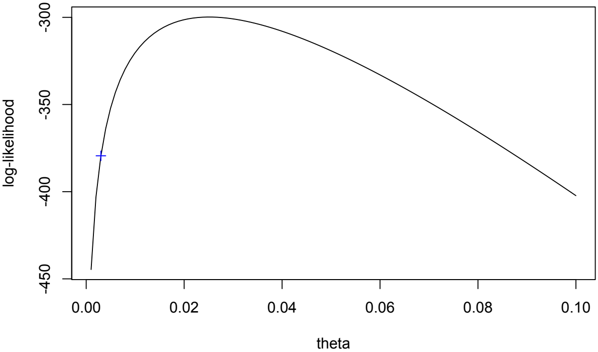
```
xseq <- seq(0,max(x),length=101)
points(xseq,dexp(xseq,thetahat),col="purple",type="l")

points(xseq,dexp(xseq,UB),col="red",type="l",lty=3)
points(xseq,dexp(xseq,LB),col="green",type="l",lty=3)
```

Maximum Likelihood: Computational Approach

- We could have taken a lazier approach to our model fitting.
- Once we had formed the likelihood, we could use a numerical optimizer to find the maximum likelihood estimator.
- For example, the Newton-Raphson algorithm can be used.
- I will demonstrate this using, pictures and then show how to do this in R.

Newton-Raphson 0a

- Suppose we have our log-likelihood function $\ell(\theta)$ and an initial guess at a value that maximizes it (call this θ_0).



- The guess wasn't very good!

Newton-Raphson 0b

- We can use a second order Taylor expansion to approximate the likelihood for θ near θ_0.

$$\ell(\theta) \approx \ell(\theta_0) + \ell'(\theta_0)(\theta - \theta_0) + \frac{1}{2}\ell''(\theta_0)(\theta - \theta_0)^2$$

Newton-Raphson 0b

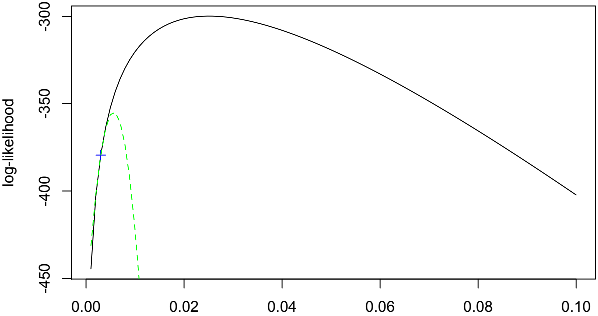
- We can use a second order Taylor expansion to approximate the likelihood for θ near θ_0.

$$
\begin{aligned}
\ell(\theta) &\approx \ell(\theta_0) + \ell'(\theta_0)(\theta - \theta_0) + \frac{1}{2}\ell''(\theta_0)(\theta - \theta_0)^2 \\
&= \ell(\theta_0) + S(\theta_0)(\theta - \theta_0) - \frac{1}{2}I(\theta_0)(\theta - \theta_0)^2
\end{aligned}
$$

Newton-Raphson 0b

- We can use a second order Taylor expansion to approximate the likelihood for θ near θ_0.

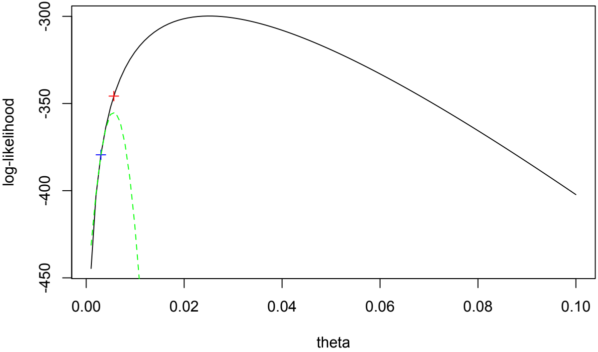
$$
\begin{aligned}
\ell(\theta) &\approx \ell(\theta_0) + \ell'(\theta_0)(\theta - \theta_0) + \frac{1}{2}\ell''(\theta_0)(\theta - \theta_0)^2 \\
&= \ell(\theta_0) + S(\theta_0)(\theta - \theta_0) - \frac{1}{2}I(\theta_0)(\theta - \theta_0)^2
\end{aligned}
$$



Newton-Raphson 0c

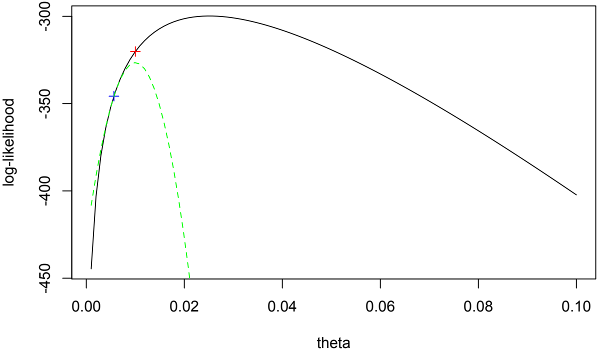
- We can find the θ that maximizes the approximation.

$$\theta_1 = \theta_0 + \frac{S(\theta_0)}{I(\theta_0)}$$



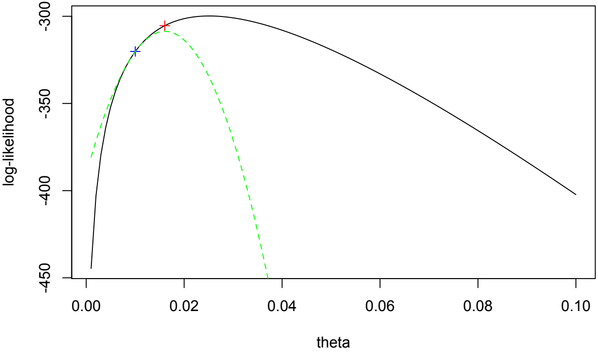
Newton-Raphson 1

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



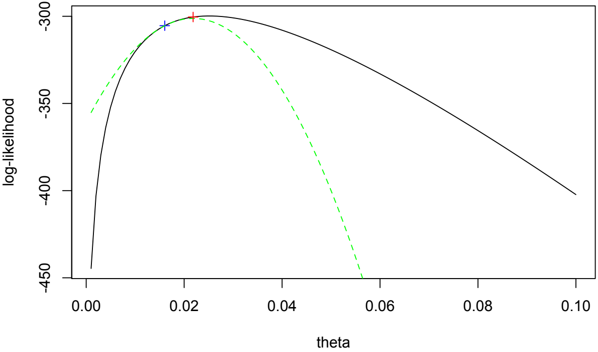
Newton-Raphson 2

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



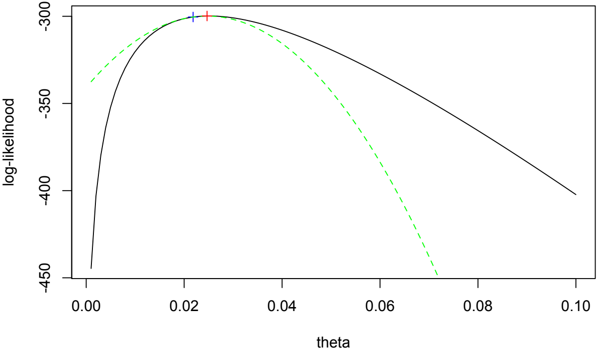
Newton-Raphson 3

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



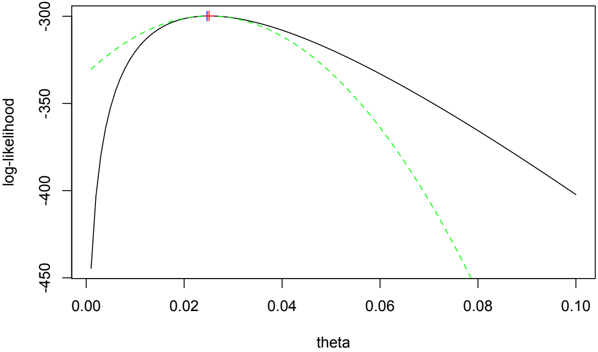
Newton-Raphson 4

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



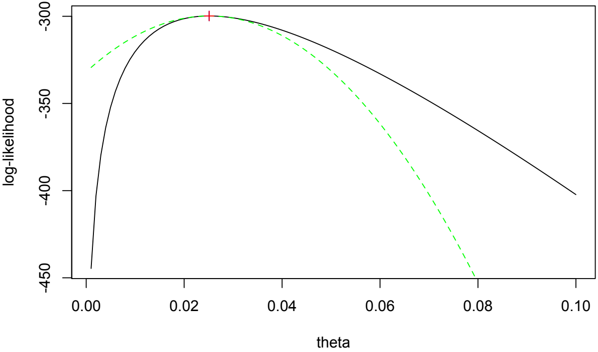
Newton-Raphson 5

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



Newton-Raphson 6

- We now repeat the process of approximating the likelihood and then maximizing the approximation.



Newton-Raphson: Code

- We can use the optim() command in R to do this optimization.

```
loglik <- function(theta,x)
{
sum(dexp(x,theta,log=TRUE))
}
theta0 <- 0.03
fit <- optim(par=theta0,fn=loglik,method="BFGS",x=x,control=list(fnscale=-1),hessian=TRUE)
```

Newton-Raphson: Code

- We can use the `optim()` command in R to do this optimization.

```
loglik <- function(theta,x)
{
sum(dexp(x,theta,log=TRUE))
}
theta0 <- 0.03
fit <- optim(par=theta0,fn=loglik,method="BFGS",x=x,control=list(fnscale=-1),hessian=TRUE)
```

- We get the following output.

```
$par
[1] 0.02511322

$value
[1] -299.8127

$counts
function gradient
      51        5

$convergence
[1] 0

$message
NULL

$hessian
        [,1]
[1,] -101802
```

Newton-Raphson: Code

- The code can be more efficient if the derivative is also supplied.

```
score <- function(theta,x)
{
n<-length(x)
n/theta-sum(x)
}
fit <- optim(par=theta0,fn=loglik,gr=score,method="BFGS",x=x,control=list(fnscale=-1)
   ,hessian=TRUE)
```

- We get the following output.

```
$par
[1] 0.02510746

$value
[1] -299.8127

$counts
function gradient
      19        4

$convergence
[1] 0

$message
NULL

$hessian
          [,1]
[1,] -101686.6
```