Statistical Inference

Lecture 03b

ANU - RSFAS

Last Updated: Wed Mar 8 14:29:53 2017

Point Estimation

Definition 1: A point estimator (statistic) is any function $T \equiv T(X_1, \ldots, X_n)$ of a sample.

- Typically we say:
  - Estimator $= T(\boldsymbol{X})$
  - Estimate $= T(\boldsymbol{x})$
- Our interest lies in determining a "good" estimate of θ (parameter(s) in our statistical model) or some $g(\theta)$ [eg. θ^2]
- What does "good" mean in this context?

Point Estimation

Definition 2: The bias of a point estimator $T = T(\boldsymbol{X})$ of a parameter θ is the difference between the expected value of T and θ.

$$\mathrm{Bias}_\theta = E[T] - \theta \quad \text{or} \quad \mathrm{Bias}_\theta = E[T] - g(\theta)$$

Definition 3: The mean squared error (MSE) of an estimator T of a parameter θ is the function

$$E_\theta\left[(T - \theta)^2\right] \quad \text{or} \quad E_\theta\left[(T - g(\theta))^2\right]$$

Point Estimation

$$
\begin{aligned}
E_\theta\left[(T-\theta)^2\right] &= E\left[(T-E(T)+E(T)-\theta)^2\right] \\
&= E[(T-E(T))^2+2(T-E(T))(E(T)-\theta)+(E(T)-\theta)^2] \\
&= E[(T-E(T))^2]+2(E(T)-\theta)E[(T-E(T))]+E[(E(T)-\theta)^2] \\
&= E[(T-E(T))^2]+0+E[(E(T)-\theta)^2] \\
&= E[(T-E(T))^2]+(E(T)-\theta)^2 \\
&= V(T)+\mathrm{Bias}(T)^2
\end{aligned}
$$

Point Estimation - Rice Chapter 8

- What are some general approaches to determine a good guess?
  - method of moments
  - maximum likelihood
  - Bayesian estimation

Method of Moments

- One of the oldest approaches. We equate the moments of a distribution to the sample moments.

Consider the following distributional moments:

$$\mu_k = E_\theta(X^k)$$

And sample moments:

$$\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^{n} x_i^k$$

for $k = 1, \ldots, K$.

Method of Moments - Rice Section 8.4

- Typically the population moments are implicitely defined by parameters $\theta = (\theta_1, \dots, \theta_K)$.
- Equate the sample and population moments:

$$\begin{aligned}
\mu_1(\theta_1, \dots, \theta_K) &= \hat{\mu}_1(x_1, \dots, x_n) \\
\vdots \quad &\quad \vdots \\
\mu_K(\theta_1, \dots, \theta_K) &= m_K(x_1, \dots, x_n)
\end{aligned}$$

- The estimator $T(\boldsymbol{X}) = \tilde{\theta}$ is the value for θ which solves the system of K equations.

Method of Moments

Eg. Poisson distribution

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{Poisson}(\lambda)$$

- Set $E[X] = \bar{X}$.

$$\tilde{\lambda} = \bar{X}$$

Method of Moments

Eg. Normal distribution

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, \sigma)$$

$$E(X) = \mu; \quad Var(X) = \sigma^2$$

Other Similar Approaches - Many!

We can replace the raw and sample moments with the so-called central moments:

$$\mu_k' = E_\theta(\{X - E_\theta(X)\}^k)$$

And sample moments:

$$\hat{\mu}_k' = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^r$$

for $k = 2, \ldots, K$. Set $\mu_1 = \hat{\mu}_1 = \bar{x}$.

Do we actually need to pick the first k moments? How about any k moments?

Other Similar Approaches - Many!

- Can we generalize the functions? Yes. This leads to generalized method of moments based on some functions $g_1(), \ldots, g_k()$:

$$
\begin{aligned}
E_\theta(g_1(X)) &= \frac{1}{n}\sum_{i=1}^{n} g_1(x_i) \\
&\vdots \\
E_\theta(g_k(X)) &= \frac{1}{n}\sum_{i=1}^{n} g_k(x_i)
\end{aligned}
$$

Note: If we set $g_i(x) = x^i$ then we recover the standard method of moments.

Maximum Likelihood Estimation - Rice Section 8.5

This is the best known, most widely used, most intuitive, most important, ... of estimation procedures.

Simply, we find the estimator $\hat{\theta}$ which maximizes the likelihood function $L(\theta|x)$.

$$L(\theta|x) = L(\theta|x_1, \ldots, x_n) = f(x_1, \ldots, x_n|\theta)$$

Before we move forward let's consider an example . . .

Maximum Likelihood Estimation

- Suppose that a particular population contains individuals of two types, A and B.
- Suppose that we are told that there are three times more of one type of individual than the other.
- We don't which is more prevalent As or Bs.
- We would like to know which is more prevalent. (our scientific question!)
- To try and answer this question we will sample 3 individuals ($n = 3$).
- Let X denote the number of A individuals in the sample (binomial distribution - what assumptions were made?).

$$P(X = x) = L(p|x) = \frac{3!}{x!(3 - x!)} p^x (1 - p)^{n-x}$$

Maximum Likelihood Estimation

	Outcome of X			
Θ	0	1	2	3
$p_1 = \frac{3}{4}$	$\frac{1}{64}$	$\frac{9}{64}$	$\frac{27}{64}$	$\frac{27}{64}$
$p_2 = \frac{1}{4}$	$\frac{27}{64}$	$\frac{27}{64}$	$\frac{9}{64}$	$\frac{1}{64}$

- Based on this table of probabilities, we can now devise a reasonable estimator for the true population value of p, based on the notion of the "preponderance of evidence" or the likelihood.

$$\hat{p} = \underset{p \in \{1/4, 3/4\}}{\operatorname{argmax}} P(X = x) \begin{cases} 1/4 & \text{if } x = 0, 1 \\ 3/4 & \text{if } x = 2, 3 \end{cases}$$

Maximum Likelihood Estimation

Eg. continued: Suppose that we don't have a restricted parameter space $p \in \{1/4, 3/4\}$ but the full natural space $[0, 1]$:

$$P(X = x) = L(p|x) = \frac{3!}{x!(3 - x!)} p^x (1 - p)^{n-x}$$

$$\frac{d}{dp} L(p|x)\Big|_{p=\hat{p}} = \frac{3!}{x!(3 - x)!} \left[x\hat{p}^{x-1}(1 - \hat{p})^{3-x} - (3 - x)\hat{p}^x (1 - \hat{p})^{2-x} \right] = 0$$

$$\hat{p} = \frac{x}{n}$$

Maximum Likelihood Estimation

Definition 4: For each sample point in $\boldsymbol{x}$, let $\hat{\theta}(\boldsymbol{x})$ be a parameter value at which $L(\theta|\boldsymbol{x})$ attains its maximum as a function of θ, with $\boldsymbol{x}$ held fixed. A maximum likelihood estimator (MLE) of the parameter θ based on a sample $\boldsymbol{X}$ is $\hat{\theta}(\boldsymbol{X})$.

- If the likelihood is differentiable in (θ_i), possible candidates for the MLE are the values $(\theta_1, \ldots, \theta_k)$ that solve

$$\frac{\partial}{\partial \theta_i} L(\theta|\boldsymbol{x}) = 0, \quad i = 1, \ldots, k$$

- Possible: local vs. global maximum, extrema may occur on the boundary and thus the first derivative may not be 0, ...
- All the good points and bad points of optimizing a function are here!

Maximum Likelihood Estimation

Eg. Poisson: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{Poisson}(\lambda)$.

$$\begin{aligned} L(\lambda|\boldsymbol{x}) &= \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \\ &= \frac{e^{-n\lambda}\lambda^{\sum_{i=1}^n x_i}}{\prod_{i=1}^n x_i!} \\ \ell(\lambda|\boldsymbol{x}) &= -n\lambda + \sum_{i=1}^{n} x_i log(\lambda) - \sum_{i=1}^{n} log(x_i!) \end{aligned}$$

$$\ell'(\lambda) = -n + \frac{\sum_{i=1}^n x_i}{\lambda} = 0$$

$$\begin{aligned} \hat{\lambda} &= \bar{x} \quad \text{estimate} \\ \hat{\lambda} &= \bar{X} \quad \text{estimator} \end{aligned}$$

Maximum Likelihood Estimation

- Do we have a maximum? Yes.

$$\ell''(\lambda) = -\frac{\sum_{i=1}^{n} x_i}{\lambda^2} < 0$$

Maximum Likelihood Estimation

Eg. (Taken from Prof. Richard Lockhart): Consider $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{Cauchy}(\theta)$.

$$f(x|\theta) = \frac{1}{\pi(1 + (x - \theta)^2)}$$

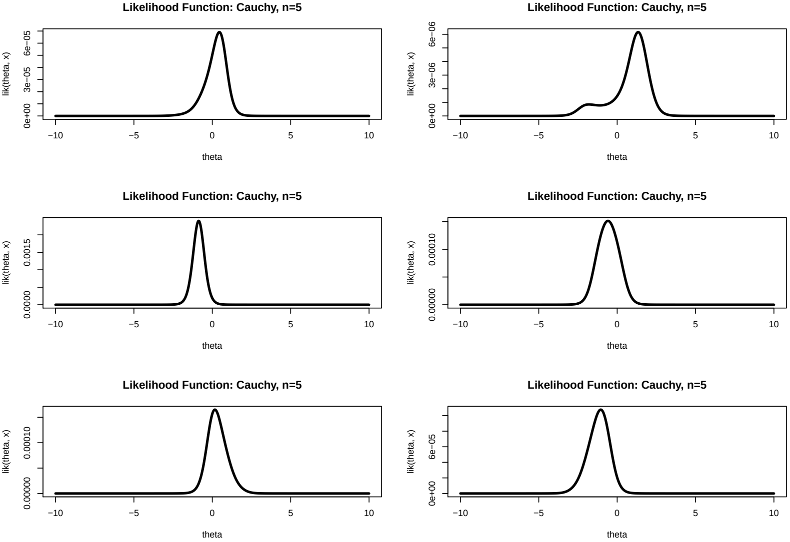
- The likelihood function is

$$L(\theta|\boldsymbol{x}) = L(\theta) = \prod_{i=1}^{n} \frac{1}{\pi(1 + (x - \theta)^2)}$$

- Here are some likelihood plots.

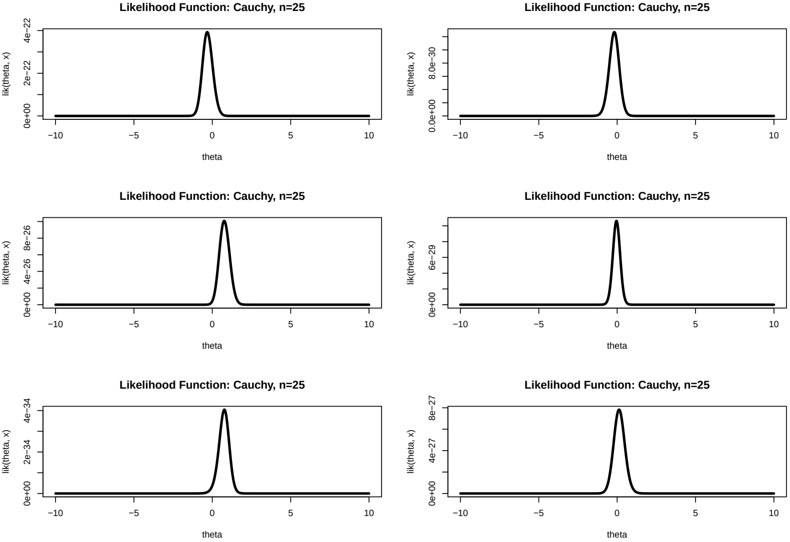
```r
set.seed(2001)
n <- 5

lik <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- prod(dcauchy(x, theta[k], 1))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, lik(theta, x), type="l", lwd=3,
     main="Likelihood Function: Cauchy, n=5")
}
```

Likelihood $n = 5$



```r
set.seed(2001)
n <- 25

lik <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- prod(dcauchy(x, theta[k], 1))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, lik(theta, x), type="l", lwd=3,
     main="Likelihood Function: Cauchy, n=25")
}
```

Likelihood $n = 25$



Things to See in the Plots

- The likelihood functions have peaks near the true value of θ (which is 0 for the data sets I generated).
- The peaks are narrower for the larger sample size.
- The peaks have a more regular shape for the larger value of n.

Maximim Likelihood Estimation

- To maximize this likelihood: differentiate $L(\theta)$, set result equal to 0.
- Notice $L(\theta)$ is product of n terms

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} \frac{1}{\pi(1+(x_i-\theta)^2)} \\
&= \left(\frac{1}{\pi(1+(x_1-\theta)^2)}\right) \times \cdots \times \left(\frac{1}{\pi(1+(x_n-\theta)^2)}\right)
\end{aligned}
$$

Maximim Likelihood Estimation

- The derivative is

$$\sum_{i=1}^{n} \prod_{j \neq i} \frac{1}{\pi(1 + (x_j - \theta)^2)} \frac{2(x_i - \theta)}{\pi(1 + (x_i - \theta)^2)^2}$$

Not fun!!

- Much easier to work with logarithm of $L(\theta)$: log of product is sum and logarithm is monotone increasing.
- The log likelihood function is

$$\ell(\theta) = log[L(\theta)]$$

Maximim Likelihood Estimation

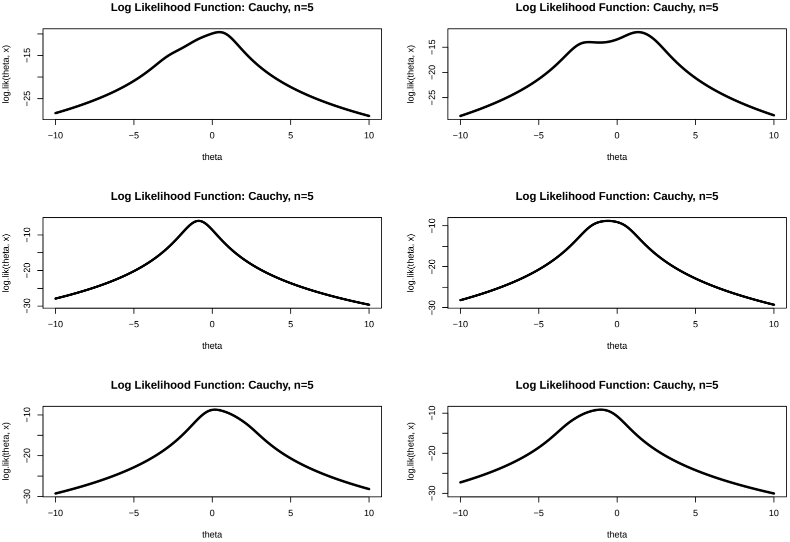
- For the Cauchy problem we have

$$L(\theta) = \prod_{i=1}^{n} \frac{1}{\pi(1 + (x - \theta)^2)}$$

$$\ell(\theta) = -\sum_{i=1}^{n} log(1 + (x_i - \theta)^2) - nlog(\pi)$$

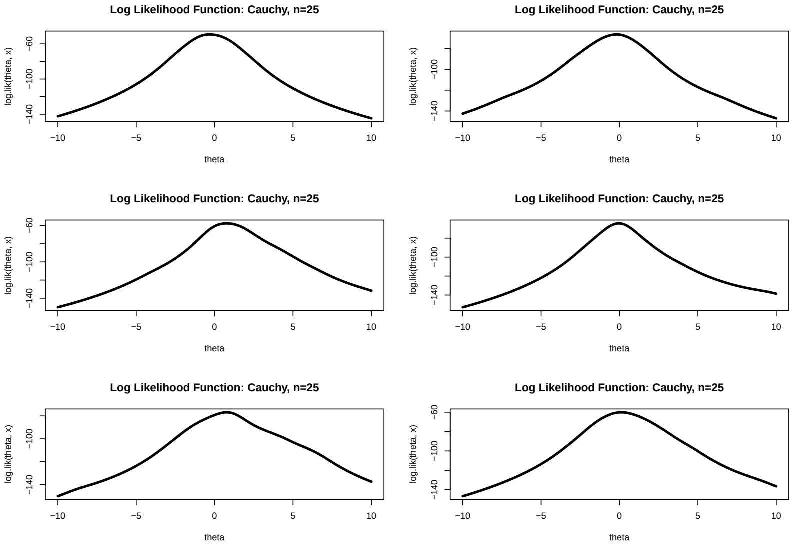
```r
set.seed(2001)
n <- 5

log.lik <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- sum(dcauchy(x, theta[k], 1, log=TRUE))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, log.lik(theta, x), type="l", lwd=3,
     main="Log Likelihood Function: Cauchy, n=5")
}
```

Cauchy log-likelihood $n = 5$



```r
set.seed(2001)
n <- 25

log.lik <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- sum(dcauchy(x, theta[k], 1, log=TRUE))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, log.lik(theta, x), type="l", lwd=3,
     main="Log Likelihood Function: Cauchy, n=25")
}
```

Cauchy log-likelihood $n = 25$



Things to Notice

- Plots of $\ell(\theta)$ for n $=$ 25 quite smooth, rather parabolic.
- For $n = 5$ many local maxima and minima of ℓ.
- Likelihood tends to 0 as $|\theta| \to \infty$, so max of $\ell(\theta)$ occurs at a root of $\ell'(\theta)$.
- Score Function is the gradient of $\ell(\theta)$

$$U(\theta) = \frac{\partial \ell}{\partial \theta} = \ell'(\theta)$$

Maximum Likelihood Estimation

- As stated the MLE is usually a root of $U(\theta)$

$$U(\theta) = 0$$

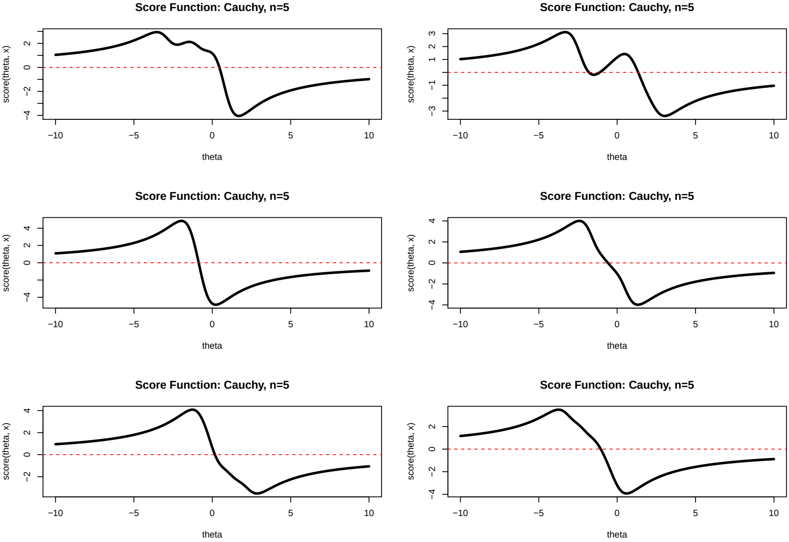
- In our Cauchy example we find

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

- Now let's examine plots of score function.
- Notice: often multiple roots.

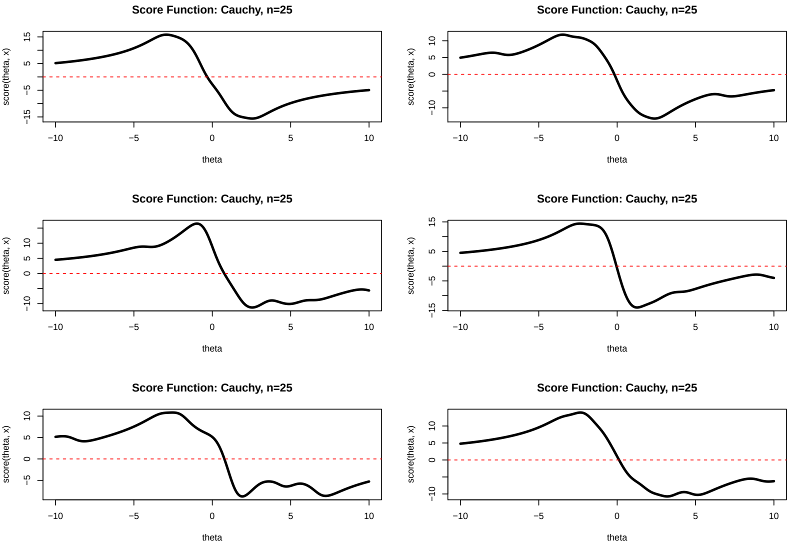
```r
set.seed(2001)
n <- 5

score <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- sum( ( 2*(x - theta[k]))/( 1 + (x - theta[k])^2))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, score(theta, x), type="l", lwd=3,
     main="Score Function: Cauchy, n=5")
}
```

Cauchy Score Function $n = 5$



```r
set.seed(2001)
n <- 25

score <- function(theta,x){
  K <- length(theta)
  n <- length(x)
  out <- rep(0, K)
  for(k in 1:K){
  out[k] <- sum( ( 2*(x - theta[k]))/( 1 + (x - theta[k])^2))
  }
  return(out)
  }

theta <- seq(-10, 10, by=0.01)
par(mfrow=c(3,2))
for(i in 1:6){
x <- rcauchy(n, location=0, scale=1)
plot(theta, score(theta, x), type="l", lwd=3,
     main="Score Function: Cauchy, n=25")
}
```

Cauchy Score Function $n = 25$



Maximum Likelihood Estimation

Eg. Exponential distribution with censoring. Consider:

$$X_1, \ldots, X_n \stackrel{\text{iid}}{\sim} \text{exponential}(\theta)$$

$$f_X(x|\theta) = \theta exp\left(-\theta x\right)$$

$$X > 0, \quad \theta > 0$$

$$E(X) = 1/\theta; \qquad Var(X) = 1/\theta^2$$

- The experiment is run until time T.
- The values $X_1, \ldots, X_m$ are observed.
- The values $X_{m+1}, \ldots, X_n$ were not observed by time T. All we know is that they exceed time T (right censored).

Maximum Likelihood Estimation

- Based on this information we can derive the likelihood:

$$L(\theta) = \prod_{i=1}^{m} f_X(x_i|\theta) \prod_{i=(m+1)}^{n} (1 - F_X(T|\theta))$$

$$F_X(T) = P(X \leq T) = \int_0^T \theta exp\left(-\theta x\right) dx = 1 - exp\left(-\theta T\right)$$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} \theta exp\left(-\theta x_i\right) \prod_{i=(m+1)}^{n} exp\left(-\theta T\right) \\ &= \theta^m exp\left(-\theta \sum_{i=1}^{m} x_i\right) exp\left(-\theta T\right)^{n-m} \end{aligned}$$

Maximum Likelihood Estimation

- So the log likelihood is:

$$l(\theta) = m \log(\theta) - \theta \sum_{i=1}^{m} x_i - (n-m)T\theta$$

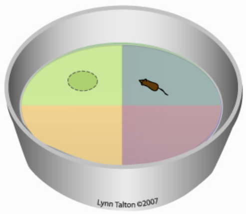
- From here we can get the score equation and solve for θ:

$$l'(\theta) = \frac{m}{\theta} - \sum_{i=1}^{m} x_i - (n-m)T = 0$$

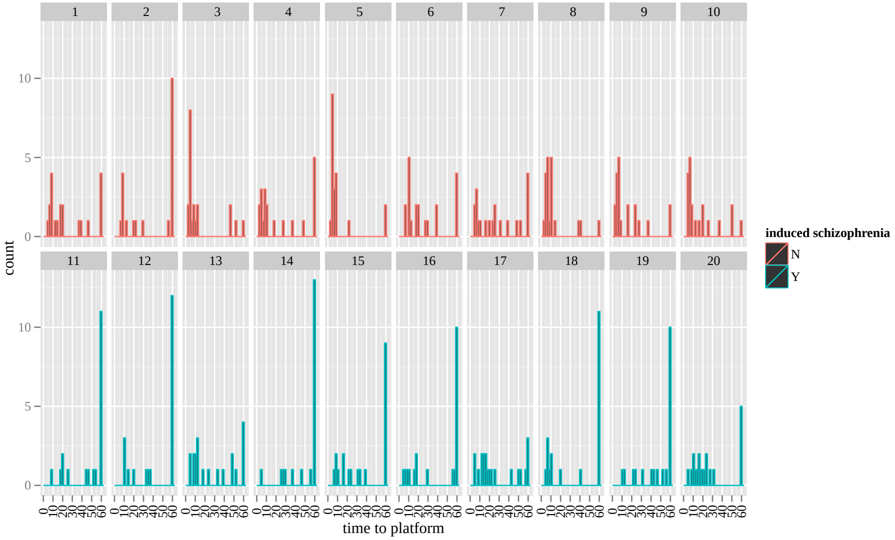
$$\hat{\theta} = \frac{m}{(n-m)T + \sum_{i=1}^{m} x_i}$$

- Check the second derivative.

A Real Example - Chemically Induced Schizophrenia (Ketamine) in Rats



- Question: Do rats with "schizophrenia" find a hidden platform slower than those without?
- Each rat attempted to find the platform on 20 occasions over 5 days (4 times per day).


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
count
0
5
10
time to platform
0 10 20 30 40 50 60
induced schizophrenia
N
Y

Maximum Likelihood Estimation

Example: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, 1)$, where it is know that μ must not be negative.

- Writing that in terms of an indicator function:

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, 1)\mathbb{I}_{[0,\infty)}(\mu)$$

$$\begin{aligned}
L(\mu|\boldsymbol{x}) &= \prod_{i=1}^{n} (2\pi)^{-1/2} exp\left(-\frac{1}{2}(x_i - \mu)^2\right) \\
&= (2\pi)^{-n/2} exp\left(-\frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right) \\
\ell(\mu|\boldsymbol{x}) &= -\frac{n}{2} log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2
\end{aligned}$$

Maximum Likelihood Estimation

$$\ell'(\mu|\boldsymbol{x}) = \sum_{i=1}^{n}(x_i - \mu) = 0$$

$$\hat{\mu} = \bar{x} \ \text{ if } \bar{x} \geq 0 \quad \text{and} \quad \hat{\mu} = 0 \ \text{ if } \bar{x} < 0$$

$$\ell''(\mu|\boldsymbol{x}) = -n < 0$$

Maximum Likelihood Estimation

Example 7.2.11: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, \sigma^2)$. No range restriction, but two parameters.

$$\begin{aligned}
L(\mu, \sigma^2|\boldsymbol{x}) &= \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \\
&= (2\pi\sigma^2)^{-n/2} exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right) \\
\ell(\mu, \sigma^2|\boldsymbol{x}) &= -\frac{n}{2} log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2
\end{aligned}$$

Maximum Likelihood Estimation

- Taking the first partial derivatives $\theta = \{\mu, \sigma^2\}$, we get the vector of the score equations:

$$U = \begin{pmatrix} \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) \\ \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - \mu)^2 - \frac{n}{2\sigma^2} \end{pmatrix}$$

- Setting $U = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and solving for the parameters we get:

$$\hat{\mu} = \bar{x}, \quad \hat{\sigma^2} = \sum_{i=1}^{n}(x_i - \bar{x})^2/n$$

- Do we have a maximum?

Maximum Likelihood Estimation

- General approach for checking. See pp. Casella and Berger 322-323. For a function $H(\theta_1, \theta_2)$ to have a local maximum at $\hat{\theta}_1, \hat{\theta}_2$ the following three conditions must hold:

1. First-order partial derivatives (score equations) at $\hat{\theta}_1, \hat{\theta}_2$ are zero:

$$\frac{\partial}{\partial \theta_1} H(\theta_1, \theta_2)\bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} = 0$$
$$\frac{\partial}{\partial \theta_2} H(\theta_1, \theta_2)\bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} = 0$$

2. At least one second-order partial derivative is negative:

$$\frac{\partial}{\partial \theta_1^2} H(\theta_1, \theta_2)\bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} < 0$$
$$\frac{\partial}{\partial \theta_2^2} H(\theta_1, \theta_2)\bigg|_{\theta_1=\hat{\theta}_1, \theta_2=\hat{\theta}_2} < 0$$

$$\frac{\partial}{\partial \mu^2}\ell(\mu, \sigma^2) = \frac{-n}{\sigma^2}|_{\sigma^2=\hat{\sigma^2}} < 0$$

$$\begin{aligned}
\frac{\partial}{\partial (\sigma^2)^2}\ell(\mu, \sigma^2) &= \frac{n}{2\sigma^4} - \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^6}|_{\mu=\hat{\mu}, \sigma^2=\hat{\sigma^2}} \overset{?}{<} 0 \\
&\Rightarrow \frac{n}{2\hat{\sigma^2}^2} \overset{?}{<} \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{\hat{\sigma^2}^3} \\
&\Rightarrow \frac{n}{2\hat{\sigma^2}^2} \overset{?}{<} \frac{n\hat{\sigma^2}}{\hat{\sigma^2}^3} \\
&\Rightarrow \frac{n}{2\hat{\sigma^2}^2} \overset{?}{<} \frac{n}{\hat{\sigma^2}^2} \\
&\Rightarrow \frac{1}{2} < 1
\end{aligned}$$

3. The determinant of the matrix of second order partial derivatives (the Hessian matrix) is positive.

$$\left|\begin{matrix} \frac{\partial}{\partial\theta_1^2} & \frac{\partial}{\partial\theta_1\partial\theta_2} \\ \frac{\partial}{\partial\theta_2\partial\theta_1} & \frac{\partial}{\partial\theta_2^2} \end{matrix}\right|_{\theta_1=\hat{\theta}_1,\theta_2=\hat{\theta}_2} > 0$$

- For our case we have:

$$\left|\begin{matrix} \frac{-n}{\sigma^2} & -\frac{\sum_{i=1}^{n}(x_i-\mu)}{\sigma^4} \\ \\ -\frac{\sum_{i=1}^{n}(x_i-\mu)}{\sigma^4} & \frac{n}{2\sigma^4}-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{\sigma^6} \end{matrix}\right|_{\mu=\hat{\mu},\sigma^2=\hat{\sigma^2}}$$

$$\left|\begin{matrix} \frac{-n}{\hat{\sigma^2}} & -\frac{\sum_{i=1}^{n}(x_i-\bar{x})}{\hat{\sigma^2}^2} \\ \\ -\frac{\sum_{i=1}^{n}(x_i-\bar{x})}{\hat{\sigma^2}^2} & \frac{n}{2\hat{\sigma^2}^2}-\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{\hat{\sigma^2}^3} \end{matrix}\right|$$

$$\begin{vmatrix} \frac{-n}{\hat{\sigma^2}} & 0 \\ 0 & \frac{n}{2\hat{\sigma^2}^2} - \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{\hat{\sigma^2}^3} \end{vmatrix} = \frac{n^2}{2\hat{\sigma^2}^3} = \frac{n^2}{2\hat{\sigma}^6} > 0$$

- So the conditions are met. We should check boundaries as well.
- We could also just check that the eigenvalues of the Hessian matrix are negative. Here we can easily see them as we have a diagonal matrix.

$$\frac{-n}{\hat{\sigma^2}} \quad \text{and} \quad \frac{-n}{2\hat{\sigma^4}}$$

MLEs - Some Computation

- Newton-Raphson (N-R) Method:
- N-R is an extremely fast root finding approach, but is sensitive to starting values.
- Again let's consider a log likelihood function $\ell(\theta|\boldsymbol{x})$.
- Let $U(\theta) = \ell'(\theta|\boldsymbol{x})$ denote first derivative of $\ell(\theta)$.
- Let $H(\theta) = \ell''(\theta|\boldsymbol{x})$ denote the derivative of $\ell(\theta)$.
- Let θ_0 be an initial estimate of θ.
- Let $\hat{\theta}$ be the MLE.
- Let's do a Taylor series expansion of $U(\theta)$ around θ_0

$$U(\theta) = U(\theta_0) + (\theta - \theta_0)H(\theta_0) + \cdots$$

- At $\theta = \hat{\theta}$ we know $U(\hat{\theta}) = 0$, so we have

$$0 = U(\theta_0) + (\hat{\theta} - \theta_0)H(\theta_0) + \cdots$$

MLEs - Computation

- Let's say the one-step approximation is reasonable enough.

$$\hat{\theta} = \theta_0 - H^{-1}(\theta_0)U(\theta_0)$$

- This suggests that $\hat{\theta}$ is well approximated by θ_1:

$$\theta_1 = \theta_0 - H^{-1}(\theta_0)U(\theta_0)$$

- We can then get an improved estimate:

$$\theta_2 = \theta_1 - U(\theta_1)H^{-1}(\theta_1)$$

- We can continue with $\theta_3, \theta_4, \ldots$ until convergence is achieved.

$$|\theta_k - \theta_{k-1}| < \epsilon = 1e - 07$$

MLEs - Computation

Eg. Poisson: $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathrm{Poisson}(\lambda)$.

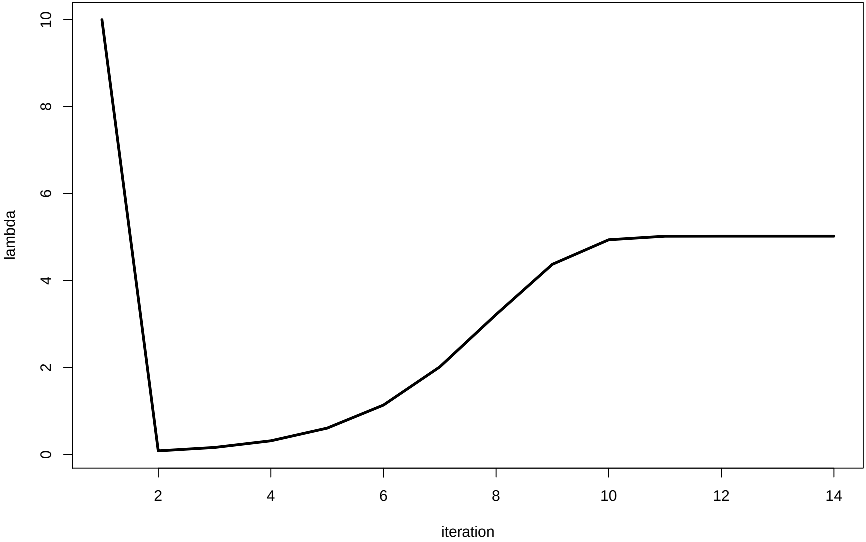
$$\begin{aligned}
\ell(\lambda) &= -n\lambda + \sum_{i=1}^{n} x_i log(\lambda) - \sum_{i=1}^{n} log(x_i!) \\
\ell'(\lambda) &= -n + \frac{\sum_{i=1}^{n} x_i}{\lambda} \\
\ell''(\lambda) &= -\frac{\sum_{i=1}^{n} x_i}{\lambda^2}
\end{aligned}$$

$$\lambda_{t+1} = \lambda_t - \left(-\frac{\sum_{i=1}^{n} x_i}{\lambda_t^2}\right)^{-1} \left(\frac{\sum_{i=1}^{n} x_i}{\lambda_t} - n\right)$$

```r
set.seed(1001)
n <- 100
x <- rpois(n, 5)

## Let's find the MLEs using the Newton-Raphson Approach
## Starting values - typically we have to be a bit careful
lambda <- 10

## Write some functions for U and H
U <- function(lambda, x){
    n <- length(x)
    out <- -n + sum(x)/lambda
    return(out)
    }

H <- function(lambda, x){
    n <- length(x)
    out <- -sum(x)/lambda^2
    return(out)
    }
```

```
## set a stopping point
eps <- 1e-07
check <- 10

## We are only interested in the final results.
## Why not save them as we go along.
out <- lambda

## Run the algorithm
while(check > eps){
lambda.new <- lambda -   U(lambda, x)/H(lambda, x)
check <- abs(lambda - lambda.new)
lambda <- lambda.new
out <- c(out, lambda)
    }
```

```r
plot(out, type="l", lwd=3, xlab="iteration", ylab="lambda")
```



```r
lambda
```

```
## [1] 5.02
```

```r
mean(x)
```

```
## [1] 5.02
```

- We find that $\hat{\lambda} = \bar{x}$. What we want.

MLEs - Some Computation

- We can naturally extend the N-R approach to a multivariate setting.
- Let $U(\boldsymbol{\theta})$ denote the vector of first partial derivatives of $\ell(\boldsymbol{\theta})$.
- Let $H(\boldsymbol{\theta})$ denote the matrix of second partial derivatives of $\ell(\boldsymbol{\theta})$.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - H^{-1}(\boldsymbol{\theta}_t)U(\boldsymbol{\theta}_t)$$

MLEs - Computation

Eg. Normal: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, \sigma^2)$.

- We saw U and H on previous slides.

```r
set.seed(1001)
x <- rnorm(100, 2, 5)

## Let's find the MLEs using the Newton-Raphson Approach
## Write some functions for U and H
U <- function(mu, sigma.sq, x){
  n <- length(x)
    out <- matrix( c( sum(x-mu)/sigma.sq, - n/(2*sigma.sq) +
                      (1/(2*sigma.sq^2))*sum((x-mu)^2) ), 2, 1)
    return(out)
    }

H <- function(mu, sigma.sq, x){
    n <- length(x)
    out <- matrix( c( -n/sigma.sq, -sum((x-mu))/sigma.sq^2,
                      -sum((x-mu))/sigma.sq^2, n/(2*sigma.sq^2) -
              sum((x-mu)^2)/sigma.sq^3), 2,2,byrow=TRUE)
    return(out)
    }
```

```r
## Starting values - typically we have to be a bit careful.
mu <- 5
sigma.sq <- 1
theta <- c(mu, sigma.sq)

## set a stopping point
eps <- 1e-07
check <- 10
c <- 2

## Save the results.
out <- theta

## Run the algorithm
while(check > eps){
theta.new <- theta - solve(H(mu, sigma.sq, x)) %*% U(mu, sigma.sq, x)
check <- sum(abs(theta-theta.new))
mu <- theta.new[1]
sigma.sq <- theta.new[2]
theta <- theta.new
out <- rbind(out, t(theta))
c <- c+1
    }
```

```r
theta
```

```
##              [,1]
## [1,]  1.995762
## [2,] 32.418953
```

```r
mean(x)
```

```
## [1] 1.995762
```

```r
n <- length(x)
sum((x-mean(x))^2)/n
```

```
## [1] 32.41895
```

- Again $\hat{\mu} = \bar{x}$ and $\hat{\sigma^2} = \sum_{i=1}^{n}(x_i - \bar{x})^2/n$.

MLE Computation - Fisher Scoring

- The method of "scoring" is a simple modification of the Newton-Raphson method.
- The Hessian $H(\boldsymbol{\theta})$ is replaced by its expectation.

$$E[H(\boldsymbol{\theta})] = -I(\boldsymbol{\theta})$$

where $I(\boldsymbol{\theta})$ is Fisher's information matrix.

$$I(\boldsymbol{\theta}) = E\left[\left(\frac{\partial \ell(\boldsymbol{\theta}|\boldsymbol{x})}{\partial \theta_i}\right)\left(\frac{\partial \ell(\boldsymbol{\theta}|\boldsymbol{x})}{\partial \theta_j}\right)\right] = -E\left[\frac{\partial^2 \ell(\boldsymbol{\theta}|\boldsymbol{x})}{\partial \theta_i \partial \theta_j}\right]$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + I^{-1}(\boldsymbol{\theta}_t) U(\boldsymbol{\theta}_t)$$

- A great advantage is that $E[H(\boldsymbol{\theta})]$ is guaranteed to be positive definite, thus eliminating some possible convergence issue with the Newton-Raphson approach.

MLE Computation - Fisher Scoring

Eg. Normal: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, \sigma^2)$.

$$H = \begin{bmatrix} \frac{-n}{\sigma^2} & -\frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^4} \\ -\frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^4} & \frac{n}{2\sigma^4} - \frac{\sum_{i=1}^n (x_i - \mu)^2}{\sigma^6} \end{bmatrix}$$

$$I(\boldsymbol{\theta}) = -E[H] = - \begin{bmatrix} \frac{-n}{\sigma^2} & -\frac{\sum_{i=1}^n (E[x_i] - \mu)}{\sigma^4} \\ -\frac{\sum_{i=1}^n (E[x_i] - \mu)}{\sigma^4} & \frac{n}{2\sigma^4} - \frac{\sum_{i=1}^n E[(x_i - \mu)^2]}{\sigma^6} \end{bmatrix}$$

$$
I(\boldsymbol{\theta}) = -\begin{bmatrix} \frac{-n}{\sigma^2} & 0 \\ \\ 0 & \frac{n}{2\sigma^4} - \frac{n\sigma^2}{\sigma^6} \end{bmatrix}
$$

$$
I(\boldsymbol{\theta}) = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ \\ 0 & \frac{n}{2\sigma^4} \end{bmatrix}
$$

$$
I(\boldsymbol{\theta})^{-1} = \begin{bmatrix} \frac{\sigma^2}{n} & 0 \\ \\ 0 & \frac{2\sigma^4}{n} \end{bmatrix}
$$

```r
set.seed(1001)
x <- rnorm(100, 2, 5)

## Let's find the MLEs using the Fisher Scoring Approach
## Write some functions for U and H
U <- function(mu, sigma.sq, x){
  n <- length(x)
  out <- matrix( c( sum(x-mu)/sigma.sq, - n/(2*sigma.sq) +
                      (1/(2*sigma.sq^2))*sum((x-mu)^2) ), 2, 1)
  return(out)
    }

I.fish <- function(mu, sigma.sq, x){
    n <- length(x)
    out <- matrix( c( n/sigma.sq, 0,
                               0, n/(2*sigma.sq^2)), 2,2, byrow=TRUE)
    return(out)
    }
```

```r
## Starting values - We don't need to be careful now!!!
mu <- -25
sigma.sq <- 50
theta <- c(mu, sigma.sq)

## set a stopping point
eps <- 1e-07
check <- 10
c <- 2

## Save the results.
out <- theta

## Run the algorithm
while(check > eps){
theta.new <- theta + solve(I.fish(mu, sigma.sq, x)) %*% U(mu, sigma.sq, x)
check <- sum(abs(theta-theta.new))
mu <- theta.new[1]
sigma.sq <- theta.new[2]
theta <- theta.new
out <- rbind(out, t(theta))
c <- c+1
    }
```

```r
theta
```

```
##              [,1]
## [1,]  1.995762
## [2,] 32.418953
```

```r
mean(x)
```

```
## [1] 1.995762
```

```r
n <- length(x)
sum((x-mean(x))^2)/n
```

```
## [1] 32.41895
```

- Again $\hat{\mu} = \bar{x}$ and $\hat{\sigma^2} = \sum_{i=1}^{n}(x_i - \bar{x})^2/n$.

MLE Computation - `optim()`

- R has a fairly robust optimizer (`optim()`). Additionally, R has several of the "new" genetic optimizer (we will not cover those).
- `optim()` has four different procedures built into it. We will use the "BFGS" (Broyden–Fletcher–Goldfarb–Shanno) method which is a quasi Newton-Raphson approach (just consider it a more robust version).
- `optim()` is actually a minimizer, so we have to tell it to maximize.

Eg. Normal: $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{normal}(\mu, \sigma^2)$.

MLE Computation - optim()

```r
set.seed(1001)
x <- rnorm(100, 2, 5)

## likelihood function
log.lik <- function(theta){
  mu <- theta[1]
  sigma.sq <- theta[2]
  out <- sum(dnorm(x, mu, sqrt(sigma.sq), log=TRUE))
  return(out)
  }

## starting values
theta.start <- c(-25,50)

##
out <- optim(theta.start, log.lik, hessian = TRUE,
             control = list(fnscale=-1), method="BFGS")
```

```
## Warning in sqrt(sigma.sq): NaNs produced

## Warning in sqrt(sigma.sq): NaNs produced
```

```r
out
```

```
## $par
## [1]  1.995893 32.392865
##
## $value
## [1] -315.831
##
## $counts
## function gradient
##       30       23
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##               [,1]          [,2]
## [1,] -3.087100e+00  1.239897e-05
## [2,]  1.239897e-05 -4.772767e-02
```

MLE Computation - `optim()`

- We did get some warnings. Likely the algorithm tried values where $\sigma^2 < 0$.
- However, the convergence value is 0, which means the algorithm met the convergence criterion (see the help for `optim()` for other codes).
- $\ell(\hat{\theta}) = -315.8$ and $\hat{\theta} = (\hat{\mu} = 1.996, \hat{\sigma^2} = 32.393)$.
- There are a number of ways to tweak this function (more iterations, convergence criterion, provide the first and second derivative, . . .)

MLE Computation - `optim()`

- The estimated variances of the estimators are given by the inverse of the Fisher information matrix (we will get to this):

$$V(\hat{\boldsymbol{\theta}}) = I^{-1}(\hat{\boldsymbol{\theta}})$$
$$I(\hat{\boldsymbol{\theta}}) = -H(\hat{\boldsymbol{\theta}})$$

```r
diag(solve(-out$hessian))
```

```
## [1]  0.3239287 20.9522078
```

- Quick check:
  $$V(\hat{\mu}) = V(\bar{X}) = \sigma^2/n \Rightarrow \widehat{\sigma^2}/n = 32.393/100 = 0.32393$$

- Wait. We didn't take expecations? That is ok. This is actually called the Observed Fisher Information.