Likelihood theory: Maximum likelihood estimation

(An overview)

N. Torelli, L. Egidi, G. Di Credico

Spring 2020

University of Trieste

The likelihood function

Maximum likelihood estimation: theory

Some numerical aspects

The likelihood function

The likelihod function

Introduced by Sir Ronald Fisher, the **likelihood function** for a certain statistical model $f_\theta(\mathbf{y})$ for the data $\mathbf{y}$ is given by the following function of the parameter $\boldsymbol{\theta}$

$$\begin{aligned} L \quad : \quad & \boldsymbol{\Theta} \to \mathbb{R}^+ \\ & \boldsymbol{\theta} \to c(\mathbf{y})\, f_\theta(\mathbf{y})\,, \end{aligned}$$

where $c(\mathbf{y}) > 0$ is an arbitrary constant of proportionality.

We may write $L(\boldsymbol{\theta}; \mathbf{y})$ to stress the fact that the data enter the function, though its argument is given by $\boldsymbol{\theta}$.

Interpreting the likelihood function

The likelihood function assigns support (*credibility*) to possible values of $\boldsymbol{\theta}$, meaning that if $L(\boldsymbol{\theta}_1) > L(\boldsymbol{\theta}_2)$ then $\boldsymbol{\theta}_1$ is more supported by the observed data than $\boldsymbol{\theta}_2$.

So the *likelihood ratio* $L(\boldsymbol{\theta}_1)/L(\boldsymbol{\theta}_2)$ allows for the comparison between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$; note that the constant $c(\mathbf{y})$ cancels out.

A mathematical justification for the above interpretation is given by the **Wald inequality**: if $\boldsymbol{\theta}_t$ is the **true parameter value**, then

$$E_{\boldsymbol{\theta}_t}\{\log L(\boldsymbol{\theta}_t; \mathbf{Y})\} > E_{\boldsymbol{\theta}_t}\{\log L(\boldsymbol{\theta}; \mathbf{Y})\} \qquad \boldsymbol{\theta} \neq \boldsymbol{\theta}_t\,.$$

The above fact can be proven by straightforward application of the Jensen's inequality.

The log likelihood function

In the previous slide the **log likelihood function** has been introduced, which is simply the logarithm of $L(\boldsymbol{\theta})$, namely

$$\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) \,.$$

The log likelihood function carries the same information of the likelihood function, but it is much more manageable. Indeed, for a random sample

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f_{\boldsymbol{\theta}}(y_i)$$

but

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f_{\boldsymbol{\theta}}(y_i) \,.$$

Notice that $\ell(\boldsymbol{\theta})$ is defined up to an additive constant, depending only on the data $\mathbf{y}$.

Example 1: the Poisson model

For a random sample $y_1, \ldots, y_n$, with $Y_i \sim \mathcal{P}(\lambda)$ i.i.d., we readily get

$$L(\lambda) = \frac{\lambda^{\sum_{i=1}^n y_i} \exp\{-n\,\lambda\}}{\prod_{i=1}^n y_i!}\,,$$

so that

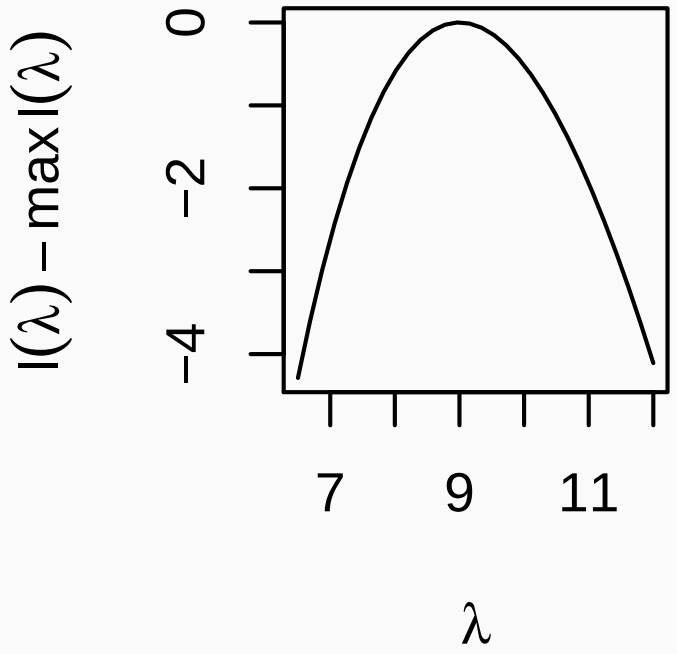
$$\ell(\lambda) = \log(\lambda) \sum_{i=1}^n y_i - n\,\lambda\,,$$

neglecting the term which does not depend on λ.

R lab: the Poisson log likelihood

Assume that for a sample $n = 10$ we observe $\sum_i y_i = 90$.

```
lik_pois <- function(lam, n, sumy) log(lam) * sumy - n * lam
xx <- seq(6.5, 12, l = 30)
ll <- sapply(xx, lik_pois, sumy = 90, n = 10)
par(pty = "s")
plot(xx, ll - max(ll), type = "l", xlab = expression(lambda),
     ylab = expression(l(lambda)-max(l(lambda))))
```



Example 2: the normal model

For a random sample $y_1, \ldots, y_n$, with $Y_i \sim \mathcal{N}(\mu, \sigma^2)$ i.i.d.

$$L(\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\,\pi\,\sigma^2}} \exp\left\{ -\frac{(y_i - \mu)^2}{2\,\sigma^2} \right\},$$

and then with some simple algebra

$$\ell(\mu, \sigma^2) = -\frac{n}{2} \log(\sigma^2) - \frac{1}{2\,\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2 .$$

Sufficient statistics

The definition of sufficient statistic, given in the probability part, can be re-interpreted for the log likelihood function: $t(\mathbf{y})$ is **sufficient** for θ if $L(\theta)$ can be written as

$$L(\theta) = h(\mathbf{y})\, g_\theta\{t(\mathbf{y})\}\,.$$

The **minimal sufficient statistic** allows for the maximal reduction of dimensionality, in the sense that a minimal sufficient statistic is a function of every other sufficient statistic.

For the Poisson model, the $\sum_i y_i$ (or, equivalently, the sample mean $\overline{y}$) is sufficient for λ, whereas for the normal model the sufficient statistic is given by the pair $(\sum_i y_i, \sum_i y_i^2)$ (or, equivalently, by the pair $(\overline{y}, s^2)$).

These two statistical models are an instance of an **exponential family**, an important model class that includes also other important elements, such as the binomial distribution. They play an important role in the theory of *generalized linear models*.

Maximum likelihood estimation

Given the interpretation of the (log) likelihood, the maximum of $\ell(\boldsymbol{\theta})$ is the value of the parameter which is most supported by the data.

A natural step is to take it as the point estimate, the **maximum likelihood estimate** (MLE) of $\boldsymbol{\theta}$

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta})$$

Notice that since $\ell(\theta)$ is also a function of $\mathbf{y}$, the MLE is a statistic.

The MLE in the two examples

For the Poisson model, simple calculus gives

$$\widehat{\lambda} = \frac{1}{n} \sum_{i=1}^{y} y_i = \overline{y}\,.$$

For the normal model, we need to maximize a function of two variables, and we get

$$\begin{cases} \widehat{\mu} = \overline{y} \\ \widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \overline{y})^2\,. \end{cases}$$

##MLE: comments Maximum likelihood estimation has a **central role** in modern statistics (and machine learning). There are several reasons for this:

1. The MLE algorithm is **automatic**: given a parametric statistical model for the data, the MLE follows from the chosen model.
2. The MLE of a function of a parameter $\psi = g(\boldsymbol{\theta})$ is defined by the simple plug-in rule $\widehat{\psi} = g(\widehat{\boldsymbol{\theta}})$, which is very convenient for

Maximum likelihood estimation: theory

Likelihood quantities

The first two derivatives of $\ell(\boldsymbol{\theta})$ play an important role.

The vector of first derivatives is called the **score function**

$$U(\boldsymbol{\theta}) = U(\boldsymbol{\theta}; \mathbf{y}) = \frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

The matrix of second derivatives, with negative sign, is called the **observed information matrix**:

$$J(\boldsymbol{\theta}) = J(\boldsymbol{\theta}; \mathbf{y}) = -\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}$$

Some properties

The derivatives of the log likelihood function satisfy some important properties, provide some **regularity conditions** hold (we shall return on them later on).

The proofs are simple, and they are reported in the CS book.

1. *Zero expected score*

$$E_{\boldsymbol{\theta}}\left\{U(\boldsymbol{\theta};\mathbf{Y})\right\} = \mathbf{0}$$

2. *2nd Bartlett identity*

$$\mathrm{cov}_{\boldsymbol{\theta}}\left\{U(\boldsymbol{\theta};\mathbf{Y})\right\} = E_{\boldsymbol{\theta}}\left\{J(\boldsymbol{\theta};\mathbf{Y})\right\} = \mathcal{I}(\boldsymbol{\theta})$$

The expected value $\mathcal{I}(\boldsymbol{\theta})$ of the observed information matrix is called the *Fisher information matrix* (or just the *expected information matrix*).

The Cramér-Rao lower bound

The third property is important, and we first state it for a one-parameter model (scalar θ).

3. *The Cramér-Rao lower bound*: the variance of *any unbiased estimator* $\tilde{\theta}$ cannot be smaller than the reciprocal of the expected information:

$$\mathrm{var}_\theta\{\tilde{\theta}(\mathbf{Y})\} \geq \frac{1}{\mathcal{I}(\theta)}\,.$$

Actually, by differentiation of the unbiasedness condition with respect to θ if follows that $\mathrm{cov}_\theta\{\tilde{\theta}, U(\theta; \mathbf{Y})\} = 1$, which readily implies the Cramér-lower bound.

The extension to multiparameter models is given by the condition that the matrix $\mathrm{cov}(\tilde{\boldsymbol{\theta}}) - \mathcal{I}(\boldsymbol{\theta})^{-1}$ is positive semi-definite.

Consistency of MLE

We are ready to state the first crucial property of the MLE:

Maximum likelihood estimators are usually consistent, that is if the sample size tends to infinity $\widehat{\boldsymbol{\theta}}$ tends to $\boldsymbol{\theta}_t$.

A justification for the result is given by the fact that in regular situations $\ell(\boldsymbol{\theta})/n \to E_{\boldsymbol{\theta}}\{\ell(\boldsymbol{\theta})\}/n$ as $n \to \infty$, so that eventually the maximum of $\ell(\boldsymbol{\theta})$ and $E\{\ell(\boldsymbol{\theta})\}$ must coincide at $\boldsymbol{\theta}_t$ by the Wald inequality.

The formal proof (typically) employs the law of large numbers.

Consistency can fail if the number of parameters increases with the sample size.

Large-sample distribution of MLE

We establish it by a Taylor expansion for the score function:

$$U(\widehat{\boldsymbol{\theta}}) \doteq U(\boldsymbol{\theta}_t) - (\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_t)\, J(\boldsymbol{\theta}_t)\,,$$

with equality when $n \to \infty$ since $\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_t \to \mathbf{0}$.

From the definition of $\widehat{\boldsymbol{\theta}}$, we get $U(\widehat{\boldsymbol{\theta}}) = \mathbf{0}$. Under mild assumptions

$$\frac{J(\boldsymbol{\theta}_t)}{n} \to \frac{\mathcal{I}(\boldsymbol{\theta}_t)}{n}\,,$$

whereas $U(\boldsymbol{\theta}_t)$ is a random vector with mean vector $\mathbf{0}$ and covariance matrix $\mathcal{I}(\boldsymbol{\theta}_t)$.

In the large sample limit

$$\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_t \,\dot{\sim}\, \mathcal{I}(\boldsymbol{\theta}_t)^{-1}\, U(\boldsymbol{\theta}_t; \mathbf{y})\,,$$

implying that $E(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_t) = \mathbf{0}$ and $\mathrm{cov}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_t) = \mathcal{I}(\boldsymbol{\theta}_t)^{-1}$.

Large-sample normality of MLE

In the case when the sample is formed by independent observations, it follows that the log likelihood is the sum of independent contributions: under mild conditions the central limit theorem applies, and in the large sample limit

$$\widehat{\boldsymbol{\theta}} \,\dot{\sim}\, \mathcal{N}\{\boldsymbol{\theta}_t, \mathcal{I}(\boldsymbol{\theta}_t)^{-1}\}\,.$$

Notice that whenever this holds, it would be possible (and recommendable, in some sense) to use $J(\boldsymbol{\theta}_t)$ in place of $\mathcal{I}(\boldsymbol{\theta}_t)$.

Again, since $\boldsymbol{\theta}_t$ is unknown, we replace it by $\widehat{\boldsymbol{\theta}}$, obtaining the following estimated standard error for the k-th component of $\boldsymbol{\theta}$

$$\mathrm{SE}(\widehat{\boldsymbol{\theta}}_k) = \sqrt{\left[J(\widehat{\boldsymbol{\theta}})^{-1}\right]_{kk}}$$

Note: for *regular models* (see next slide), the observed information is positive definite at $\widehat{\boldsymbol{\theta}}$, so that the SE above is well defined.

Regularity conditions

We end the summary of the theory by mentioning the **regularity conditions**, which are some assumptions on the statistical model, required for the previous results to be valid.

The CS book lists the following ones:

1. The pdf of $\mathbf{y}$ defined by different values of $\boldsymbol{\theta}$ are distinct, namely the model is *identifiable*.
2. The true parameter value $\boldsymbol{\theta}_t$ is interior to $\boldsymbol{\Theta}$.
3. Within some neighbourhood of $\boldsymbol{\theta}_t$, the first three derivatives of $\ell(\boldsymbol{\theta})$ exist and are bounded, while the expected information satisfies the 2nd Bartlett identity, is positive definite and finite.

These are mild conditions, which are generally valid in most cases.

Winding up

The previous results have illustrated that

1. The MLE is a **consistent estimator**.
2. The MLE is **asymptotic efficient**, since its asymptotic variance attains the Cramér-Rao lower bound.
3. The large sample distribution (aka the approximate distribution) of the MLE is **multivariate normal**, with standard error that can be estimated by the observed information evaluated at the parameter estimate.

Example 1: Poisson model

Here $\widehat{\lambda} = \overline{y}$, and consistency follows from the law of large numbers, in agreement with likelihood theory.

Furthermore, the CLT states that for large n

$$\widehat{\lambda} \dot{\sim} \mathcal{N}(\lambda, \lambda/n)\,.$$

This result can be obtained also from likelihood theory. Indeed, we get

$$J(\lambda) = \frac{\sum_i y_i}{\lambda^2}$$

so that $\mathcal{I}(\lambda) = n/\lambda$ and $\mathcal{I}(\lambda)^{-1} = \lambda/n$.

Example 2: normal example

Here we get

$$J(\mu, \sigma^2) = \begin{pmatrix} \frac{n}{\sigma^2} & \frac{n}{\sigma^4}(\bar{y} - \mu) \\ \frac{n}{\sigma^4}(\bar{y} - \mu) & -\frac{n}{2\,\sigma^4} + \frac{1}{\sigma^6}\sum_{i=1}^{n}(y_i - \mu)^2 \end{pmatrix}$$

and therefore

$$\mathcal{I}(\mu, \sigma^2) = \begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\,\sigma^4} \end{pmatrix}$$

The implication is that $\widehat{\mu}$ and $\widehat{\sigma}^2$ are (asymptotically) uncorrelated, and the two estimated standard errors are

$$\mathrm{SE}(\widehat{\mu}) = \frac{\widehat{\sigma}}{\sqrt{n}}, \qquad \mathrm{SE}(\widehat{\sigma}^2) = \frac{\sqrt{2}\,\widehat{\sigma}^2}{\sqrt{n}}.$$

Some numerical aspects

Numerical optimisation

The algorithmic nature of the MLE estimation method translates the statistical model into an optimisation problem: once a (sensible) statistical model has been specified for the data, we obtain parameter estimates with excellent properties by maximizing the log likelihood.

In some simple settings, like in the examples above, it is possible to find the analytical expression for the MLE, but in general we must resort to **numerical optimisation** of the log likelihood.

There are indeed several methods available for the task. Some knowledge of the most important issues related to it turns out particularly useful even for the application of off-the-shelf routines in `R` (or other environments).

Newton's method

Newton's method for optimisation is commonly used for minimization, in this case of the objective function $f(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta})$.

The theory is well described in the CS book, here we mention the most important aspects. The idea is to locally approximate $f(\boldsymbol{\theta})$ as a quadratic function, which is repeatedly minimised.

The resulting method consists in an **iterative algorithm**, which is started with $k = 0$ and a *guesstimate* $\boldsymbol{\theta}^{[0]}$, and iterates the following steps:

1. Evaluate $\ell(\boldsymbol{\theta}^{[k]})$, $U(\boldsymbol{\theta}^{[k]})$ and $J(\boldsymbol{\theta}^{[k]})$.
2. If $U(\boldsymbol{\theta}^{[k]}) \doteq \mathbf{0}$ and $J(\boldsymbol{\theta}^{[k]})$ is positive definite then stop.
3. If $\mathbf{H} = J(\boldsymbol{\theta}^{[k]})$ is not positive definite, perturb it so that it is.
4. Solve $\mathbf{H}\,\boldsymbol{\delta} = U(\boldsymbol{\theta}^{[k]})$ for the search direction $\boldsymbol{\delta}$.
5. If $\ell(\boldsymbol{\theta}^{[k]} + \boldsymbol{\delta})$ is not $> \ell(\boldsymbol{\theta}^{[k]})$, repeatedly halve $\boldsymbol{\delta}$ until it is (*this is the step-length control*).
6. Set $\boldsymbol{\theta}^{[k+1]} = \boldsymbol{\theta}^{[k]} + \boldsymbol{\delta}$, increment k by one and return to step 1.

Fisher scoring and Quasi-Newton.

Whenever available, it is always a good idea to replace the observed information with the expected information $\mathcal{I}(\boldsymbol{\theta}^{[k]})$ in the Newton's method.

The resulting algorithm has a long successful tradition in statistics, it is called **Fisher scoring** and, indeed, it has better convergence properties.

Another variant avoids the computation of either $J(\boldsymbol{\theta}^{[k]})$ or $\mathcal{I}(\boldsymbol{\theta}^{[k]})$, by building an approximation to the second derivative of $\ell(\boldsymbol{\theta})$ as the optimization proceeds. This is the approach of the **Quasi-Newton** methods, such as the widely used BFGS algorithm.

Quasi-Newton methods are implemented in several `R` functions and packages; see the `CRAN Task View` for *Optimisation* (https://cran.r-project.org/web/views/Optimization.html).

An example: logistic regression

We follow the MASS book for a simple example on a dose-response model.

Namely, we assume that y_i is the number of dead budworms (out of 20) for a dose of insecticide x_i^*. In particular, the statistical model is

$$Y_i \sim \mathcal{B}_i(20, \pi_i) \qquad i = 1, \ldots, 12, \text{ independent}$$

with

$$\pi_i(\alpha, \beta) = \frac{e^{\alpha + \beta\, x_i}}{1 + e^{\alpha + \beta\, x_i}}$$

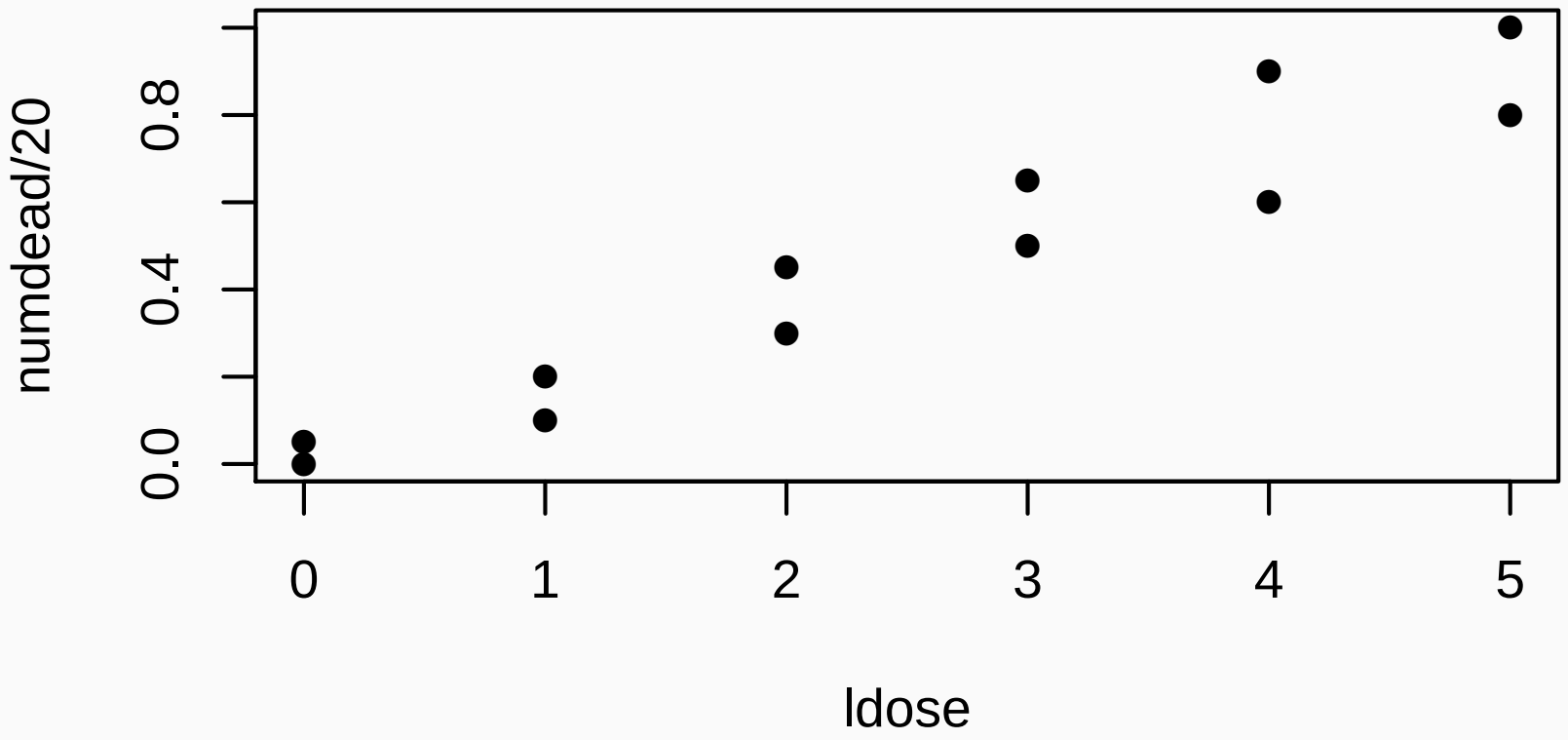
with $x_i = \log(x_i^*)$.

This is a simple instance of a *logistic regression model*.

R lab: budworm data

There are two observations at each dose (M/F budworms), but here for the sake of simplicity we ignore the different sex.

```
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
plot(ldose, numdead / 20, pch=16)
```



Logistic regression: likelihood quantities

With some simple algebra we get:

$$\ell(\alpha, \beta) = \sum_i \left\{ y_i \left(\alpha + \beta\, x_i\right) - 20\, \log(1 + e^{\alpha + \beta\, x_i}) \right\}$$

$$U(\alpha, \beta) = \begin{pmatrix} \sum_i \{y_i - 20\, \pi_i(\alpha, \beta)\} \\ \sum_i \{y_i - 20\, \pi_i(\alpha, \beta)\}\, x_i \end{pmatrix}$$

$$\mathcal{I}(\alpha, \beta) = \begin{pmatrix} \sum_i 20\, \pi_i(\alpha, \beta)\, \{1 - \pi_i(\alpha, \beta)\} & \sum_i 20\, \pi_i(\alpha, \beta)\, \{1 - \pi_i(\alpha, \beta)\}\, x_i \\ \sum_i 20\, \pi_i(\alpha, \beta)\, \{1 - \pi_i(\alpha, \beta)\}\, x_i & \sum_i 20\, \pi_i(\alpha, \beta)\, \{1 - \pi_i(\alpha, \beta)\}\, x_i^2 \end{pmatrix}$$

Notice that for this model $J(\alpha, \beta) = \mathcal{I}(\alpha, \beta)$.

R lab: likelihood and score functions

```
loglik <- function(theta, data){
           eta <- theta[1] + theta[2] * data$x
           out <- sum(data$y * eta - 20 * log(1+exp(eta)))
           return(out)
           }

score <- function(theta, data){
          prob <- plogis(theta[1] + theta[2] * data$x)
          out <- c(sum(data$y - prob * 20),
                   sum((data$y - prob * 20) * data$x))
          return(out)
        }
```

R lab: information function

```
info <- function(theta, data){
        prob <- plogis(theta[1] + theta[2] * data$x)
        info11 <- sum(20 * prob * (1-prob))
        info12 <- sum(20 * prob * (1-prob) * data$x)
        info22 <- sum(20 * prob * (1-prob) * data$x^2)
        out <- matrix(c(info11, info12, info12, info22), 2, 2)
        return(out)
        }
```

R lab: starting point

Let's start from $\alpha = \beta = 0$: we obtain

```
theta0 <- c(0, 0); budw <- data.frame(y = numdead, x = ldose)

loglik(theta0, budw)

## [1] -166.3553
score(theta0, budw)

## [1]  -9 105
info(theta0, budw)

##      [,1] [,2]
## [1,]   60  150
## [2,]  150  550
```

R lab: first step

```
H <- info(theta0, budw)
u0 <- score(theta0, budw)
delta <- solve(H, u0)
theta1 <- theta0 + delta

theta1
```

```
## [1] -1.9714286  0.7285714
```

```
loglik(theta1, budw)
```

```
## [1] -114.7219
```

which is clearly an improvement.

R lab: first 10 steps

```
## k = 1 theta= -1.971429 0.7285714 loglik= -114.7219
## k = 2 theta= -2.621436 0.9572079 loglik= -111.8192
## k = 3 theta= -2.760585 1.004947 loglik= -111.734
## k = 4 theta= -2.766079 1.006804 loglik= -111.7339
## k = 5 theta= -2.766087 1.006807 loglik= -111.7339
## k = 6 theta= -2.766087 1.006807 loglik= -111.7339
## k = 7 theta= -2.766087 1.006807 loglik= -111.7339
## k = 8 theta= -2.766087 1.006807 loglik= -111.7339
## k = 9 theta= -2.766087 1.006807 loglik= -111.7339
## k = 10 theta= -2.766087 1.006807 loglik= -111.7339
```

The algorithm converges quickly, and actually after 10 iterations

```
cat(score(theta10, budw), det(info(theta10, budw)),
    sqrt(diag(solve(info(theta10, budw)))))
```

```
## 1.776357e-15 5.329071e-15 2361.462 0.3701342 0.1235889
```

R lab: glm analysis

```
budworm.lg0 <- glm(cbind(y, 20-y) ~ x, binomial, budw)
summary(budworm.lg0, cor = FALSE)

##
## Call:
## glm(formula = cbind(y, 20 - y) ~ x, family = binomial, data =
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7989  -0.8267  -0.1871   0.8950   1.9850
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.7661     0.3701  -7.473 7.82e-14 ***
## x             1.0068     0.1236   8.147 3.74e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
##
```