STATS 330

Handout 3
Parameter estimation for generalised linear models

Department of Statistics, University of Auckland

Parameter estimation

In this handout, we answer the following question:

- Given a data set and a generalised linear model, how do we estimate the regression coefficients?

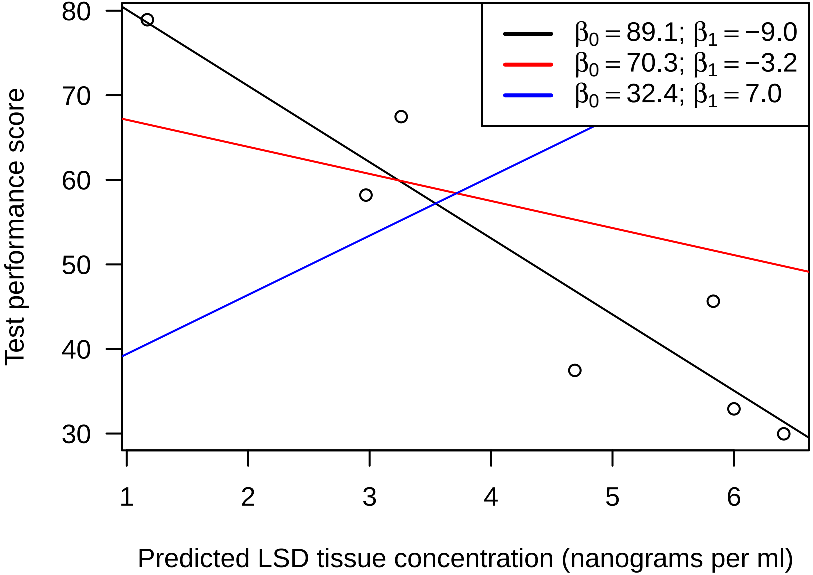
In other words, when we use the `glm()` function, what is R doing behind-the-scenes to calculate the estimated coefficients?

We will cover the following topics:

- Least-squares estimation
- Maximum likelihood estimation

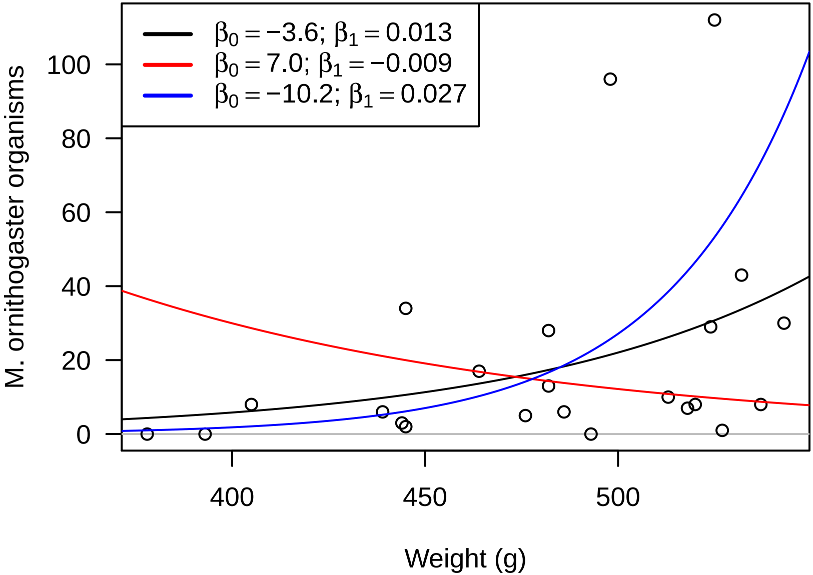
Parameter estimation

Linear regression: LSD effect on maths analysis



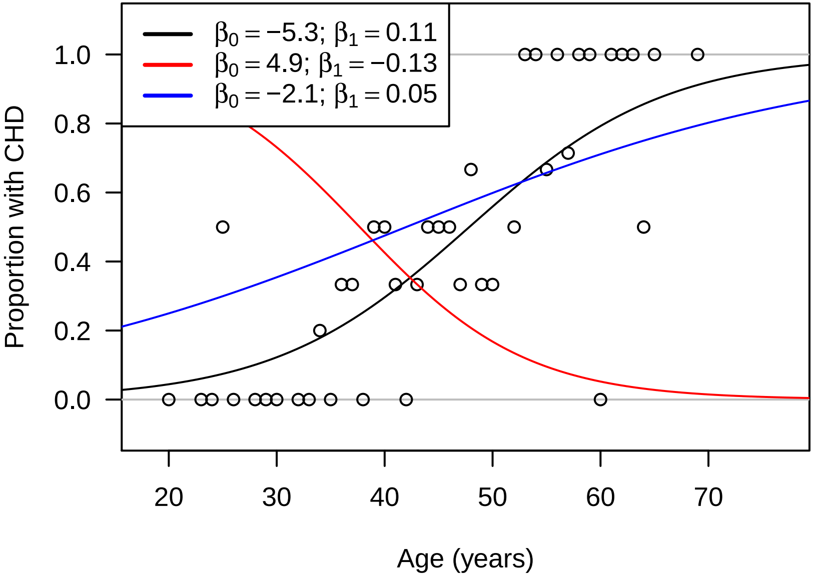
Parameter estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



Parameter estimation

Logistic regression: Coronary heart disease analysis



Parameter estimation

In each of the preceding plots, the black line appears to fit the data ‘better’ than the others. But what makes one line ‘better’ than another?

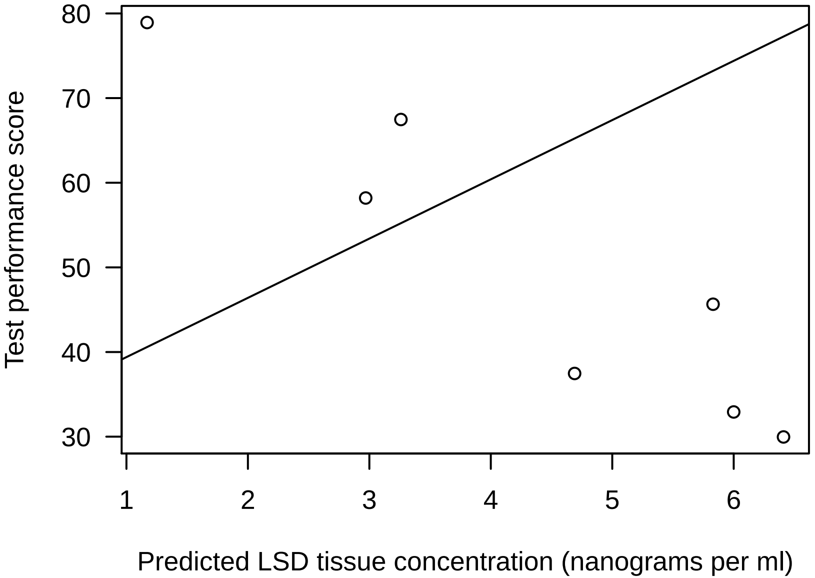
Parameter estimation involves determining values for a model’s coefficients (or ‘parameters’) that ‘best’ fit the data at hand

In order to do this, we must somehow measure how ‘good’ a particular candidate set of parameter estimates are:

- For linear regression models, we can use the sum of the squared residuals.
- For generalised linear models, we can use the likelihood function.

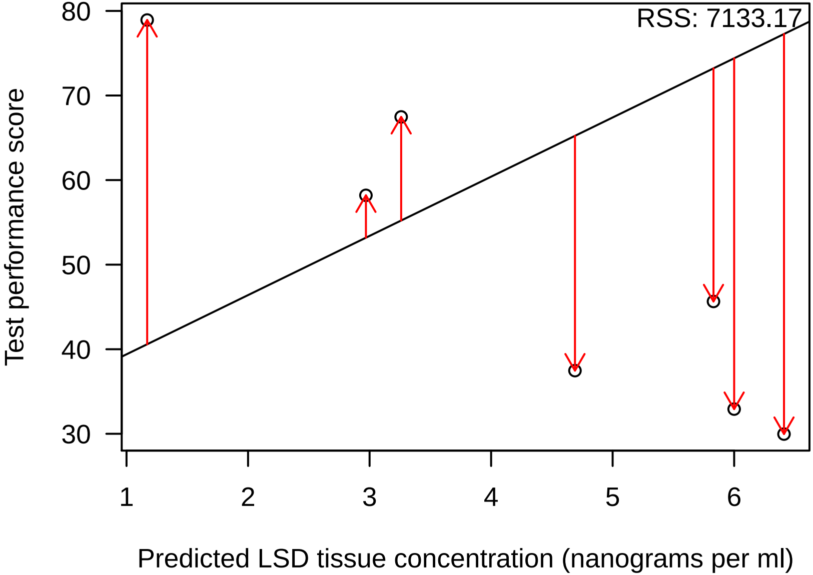
Least squares estimation

Linear regression: LSD effect on maths analysis



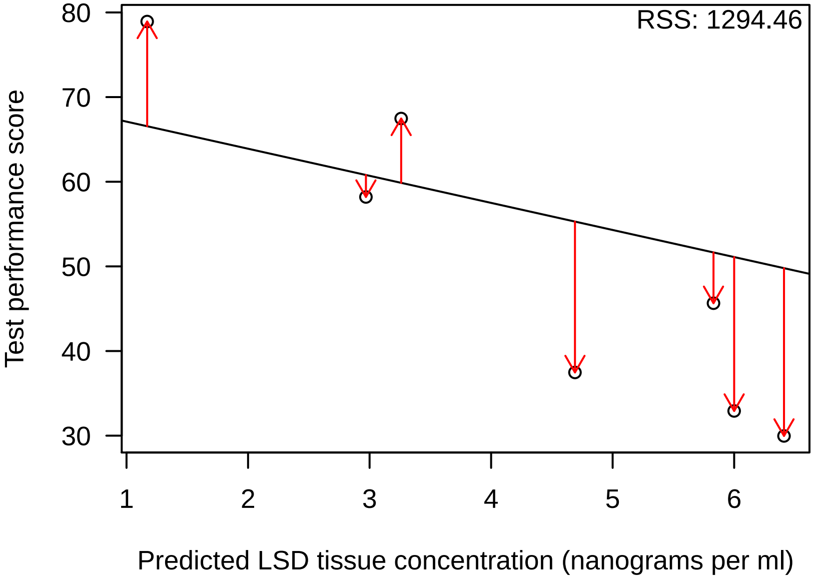
Least squares estimation

Linear regression: LSD effect on maths analysis



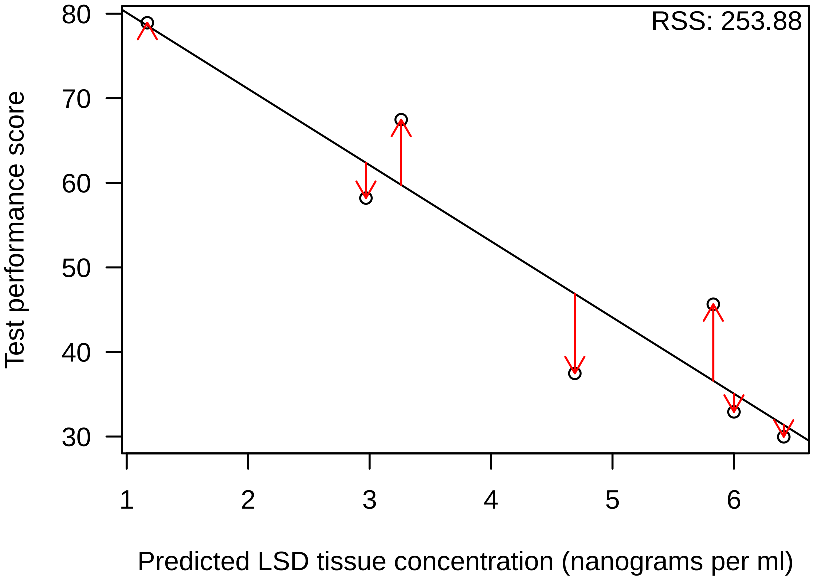
Least squares estimation

Linear regression: LSD effect on maths analysis



Least squares estimation

Linear regression: LSD effect on maths analysis



Least squares estimation

The residual sum of squares is given by

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2.$$

The lower the residual sum of squares, the ‘better’ a candidate set of parameter values is. We estimate the coefficients by finding those that give us the lowest possible residual sum of squares. This method is known as least squares estimation.

It can be proven (see STATS 310) that the coefficients minimising the residual sum of squares are given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

where $\boldsymbol{y}$ is a vector containing the observed response for all observations, and $\boldsymbol{X}$ is the design matrix (see end of Handout 1).

Least squares estimation

So, if we have the following R objects...

```
y

## [1] 78.93 58.20 67.47 37.47 45.65 32.92 29.97

X

##      [,1] [,2]
## [1,]    1 1.17
## [2,]    1 2.97
## [3,]    1 3.26
## [4,]    1 4.69
## [5,]    1 5.83
## [6,]    1 6.00
## [7,]    1 6.41
```

Least squares estimation

... We could compute the vector of estimated coefficients, $\hat{\beta}$, as follows:

```
solve(t(X) %*% X) %*% t(X) %*% y

##           [,1]
## [1,] 89.123874
## [2,] -9.009466
```

In R

- The `solve()` function computes an inverse of a matrix,
- The `t()` function transposes a matrix, and
- The operator %*% carries out matrix/vector multiplication.

Least squares estimation

But of course it's easier to let R do the work for us:

```
lsd.fit <- lm(score ~ lsd)
coef(lsd.fit)

## (Intercept)         lsd
##   89.123874   -9.009466
```

Also note that we can extract the minimised residual sum of squared residuals:

```
anova(lsd.fit)

## Analysis of Variance Table
##
## Response: score
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## lsd        1 1824.30 1824.30  35.928 0.001854 **
## Residuals  5  253.88   50.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Least squares estimation

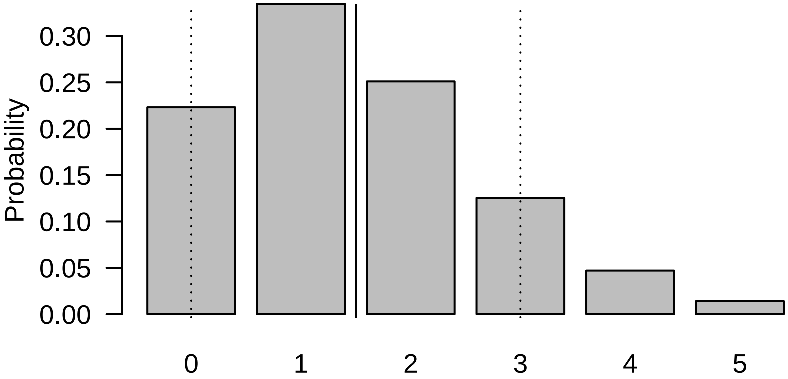
Least squares estimation looks to minimise the sum of the squared differences between observed and expected values. Negative differences have the same effect as positive differences, and so they are treated in the same way. A consequence of this is that least squares is only appropriate when the response distribution is symmetric.

For asymemetric distributions, we need a different approach. An observation z units above the expected value might be plausible, but a value z values below the expected value might be quite strange (or even impossible). We should not treat them the same.

Additionally, least-squares regression weights residuals the same regardless of the amount of variance we'd expect them to have. We should not equally weight residuals if our model incorporates nonconstant variance.

Least squares estimation

An asymmetric distribution



The Poisson distribution with expectation $\mu = 1.5$. An observation 1.5 less than the expectation (left dotted line) is almost twice as probable as an observation 1.5 units larger than the expectation (right dotted line).

Maximum likelihood estimation

In general, for GLMs, we estimate coefficients by maximum likelihood. You may have encountered this approach in other statistics courses (e.g., STATS 210, 310).

To calculate the likelihood function for a candidate set of parameter values, $\boldsymbol{\beta}$, we do the following:

1. Calculate the probability of observing each observation's response, and
2. Take the product of these probabilities.

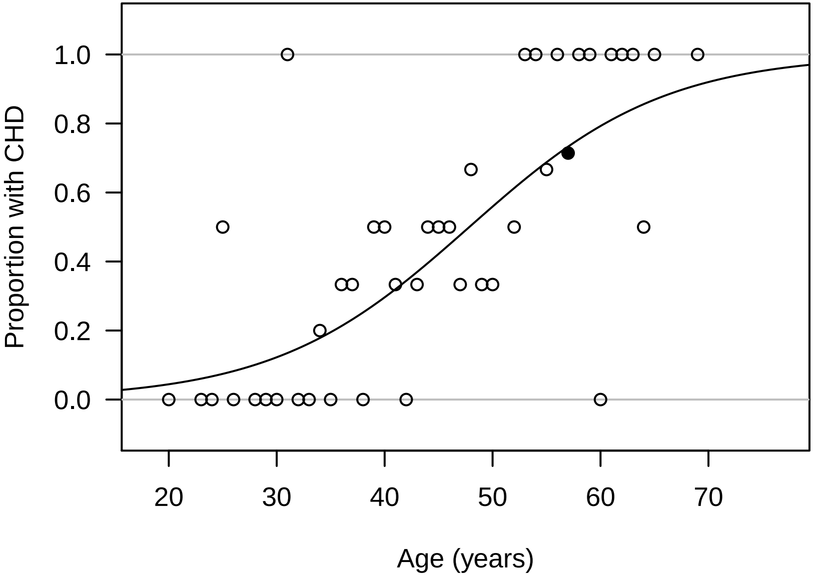
As an equation:

$$L = \prod_{i=1}^{n} f(y_i; \boldsymbol{\beta}),$$

where $f(y_i; \boldsymbol{\beta})$ is the probability mass function of the assumed response distribution.

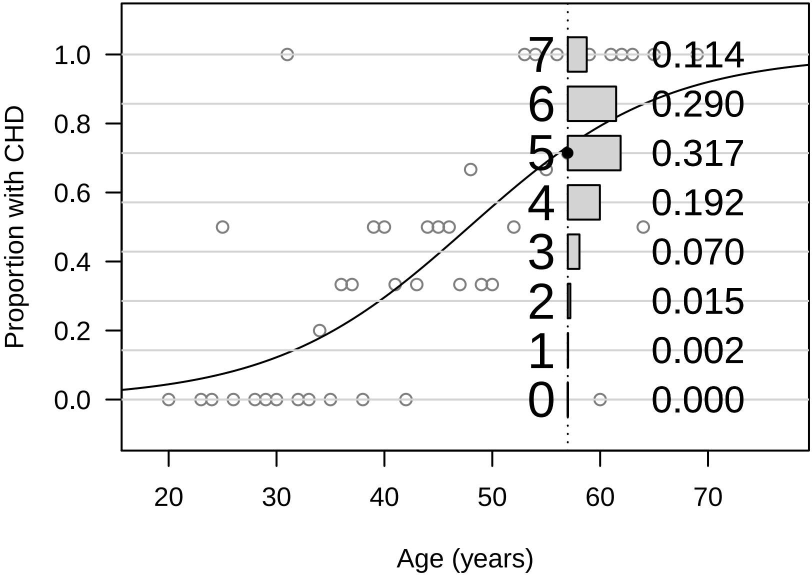
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



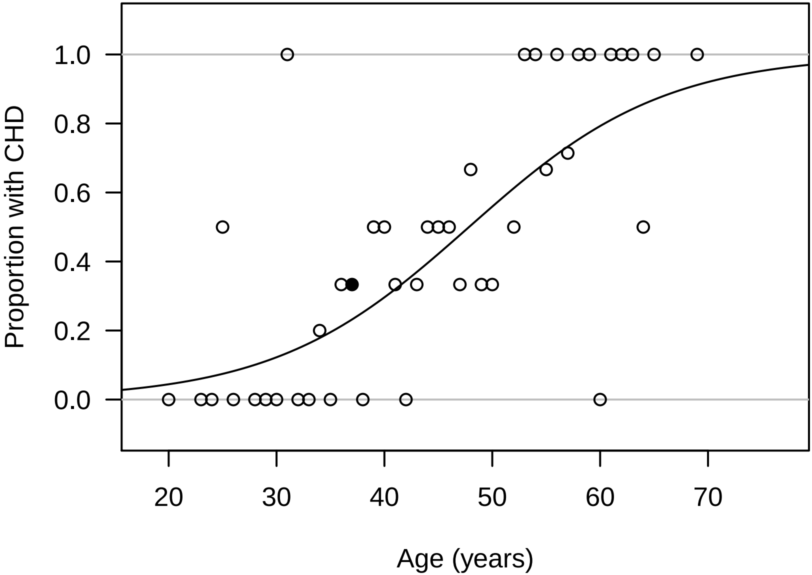
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



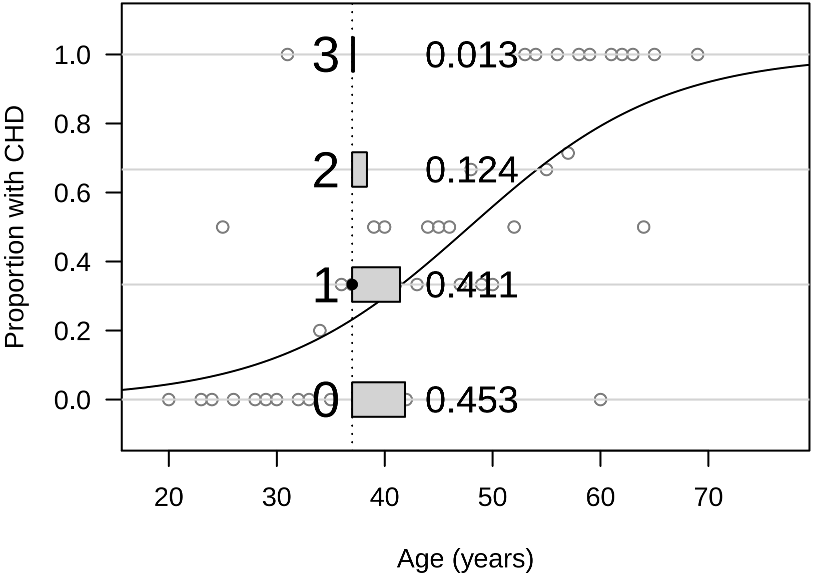
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



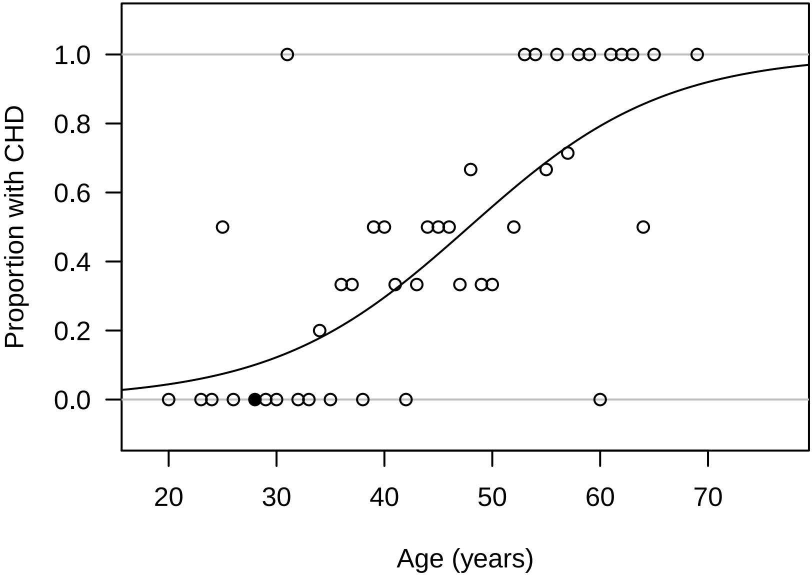
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



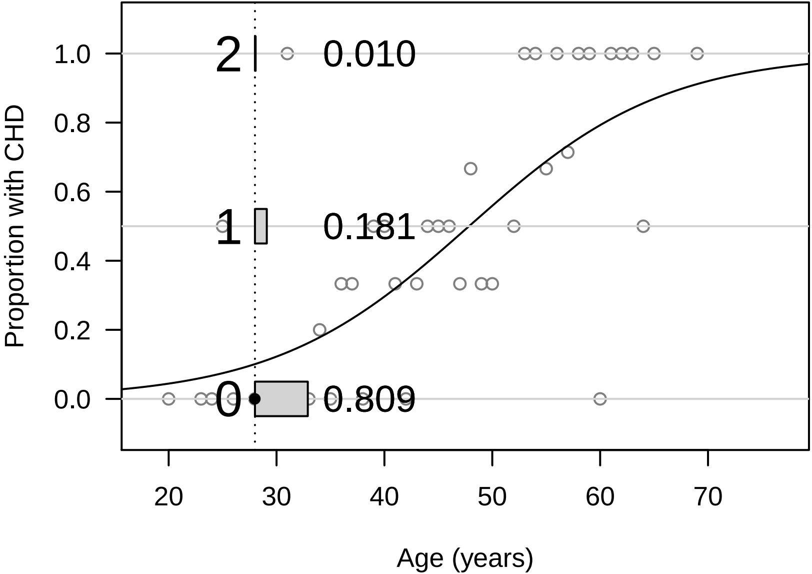
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



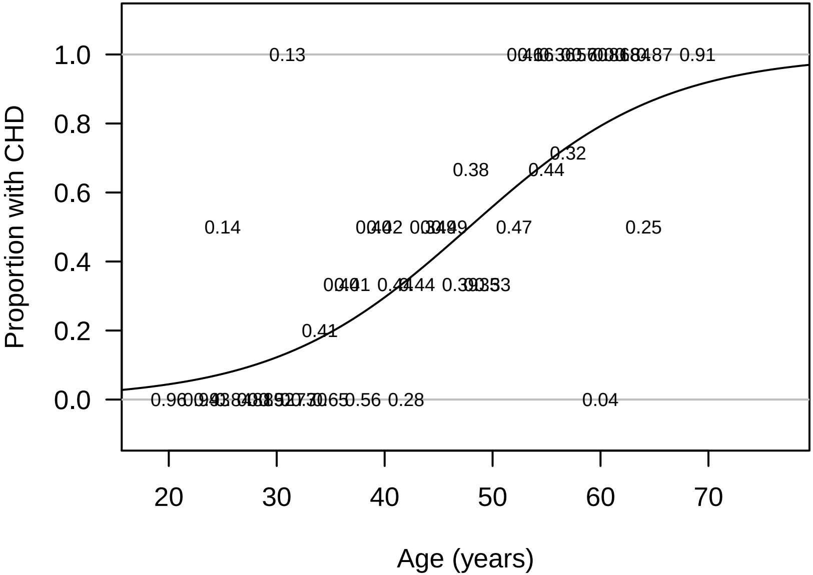
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



Maximum likelihood estimation

Roughly speaking,

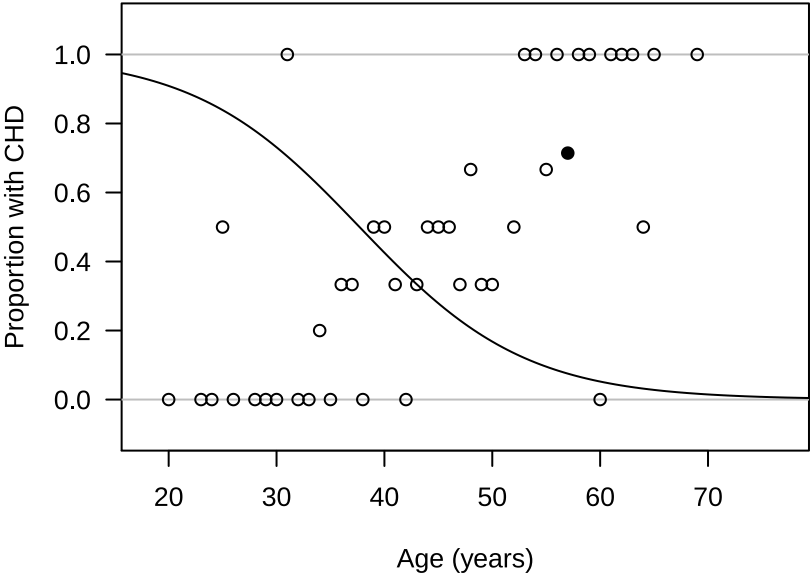
- Points close to the line will have high likelihood contributions, because they are consistent with the candidate parameters
- Points far from the line will have low likelihood contributions, because they are not consistent with the candidate parameters.

Maximum likelihood estimation involves finding the $\boldsymbol{\beta}$ parameter values that maximise the likelihood function.

- The regression line that gives the largest likelihood will be as consistent as possible with as many observations as possible.

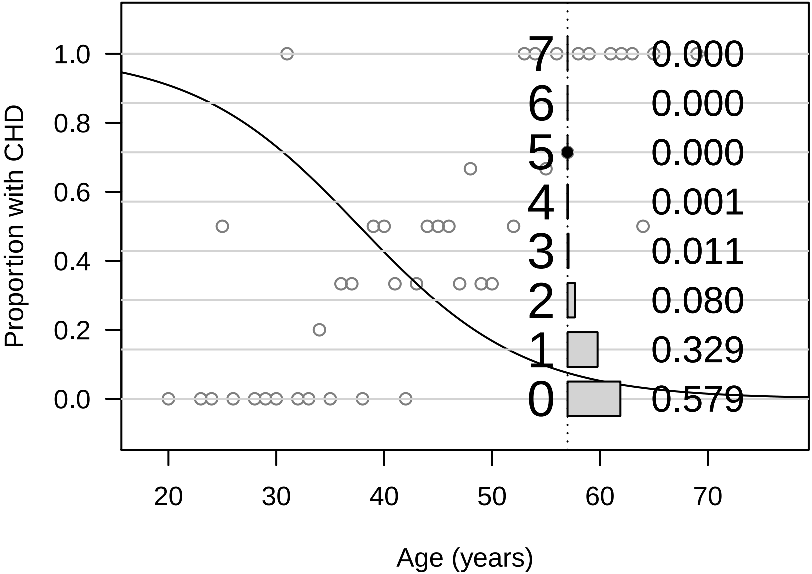
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



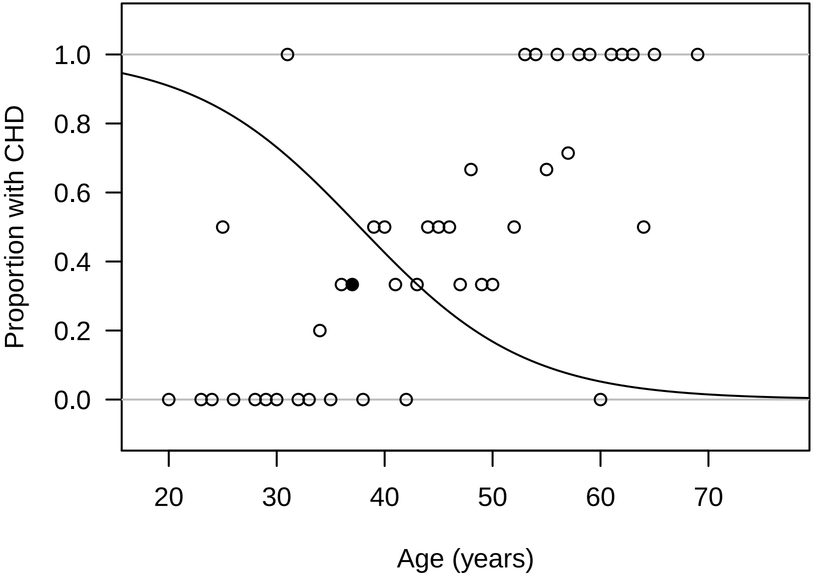
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



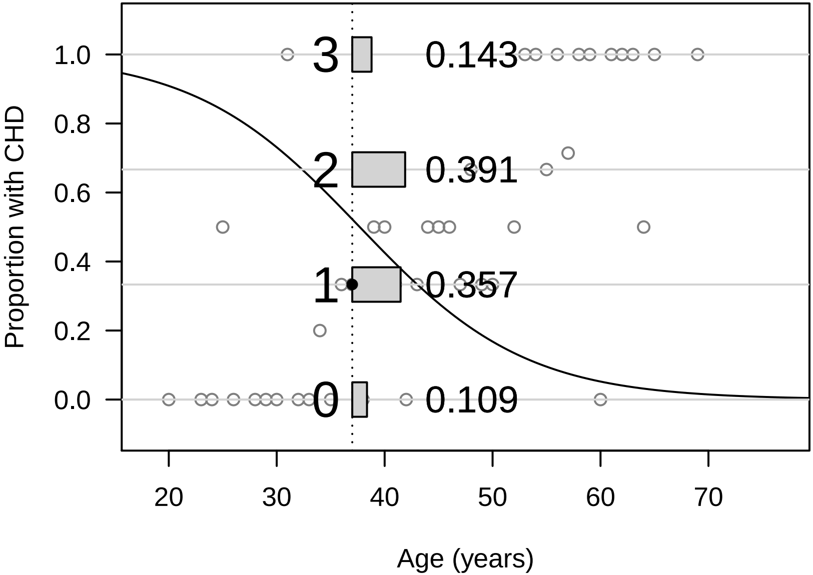
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



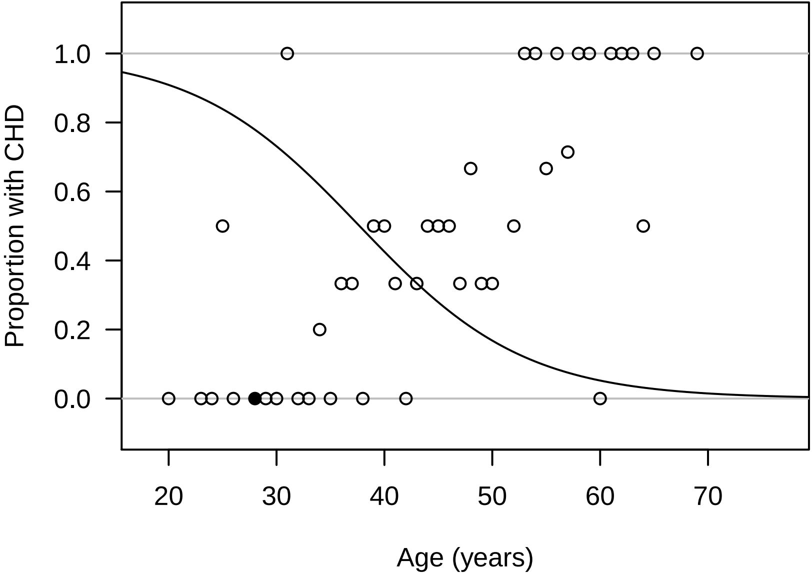
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



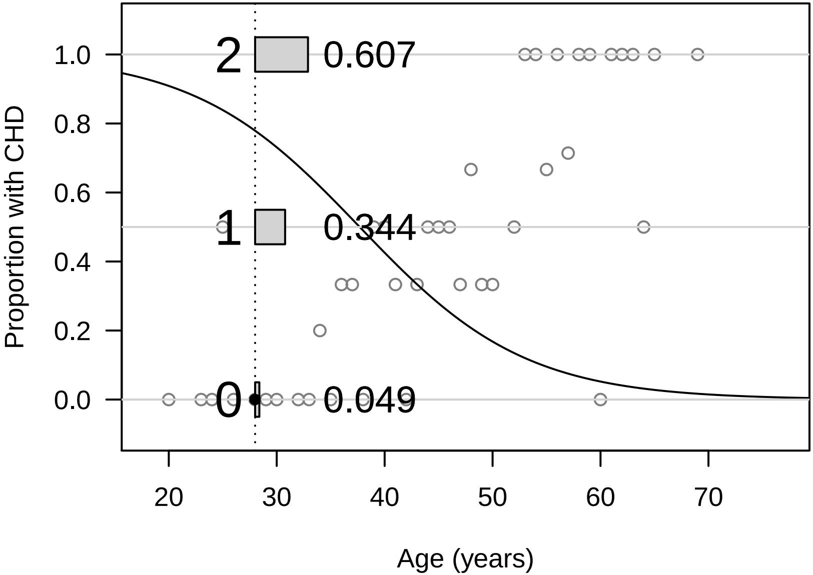
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



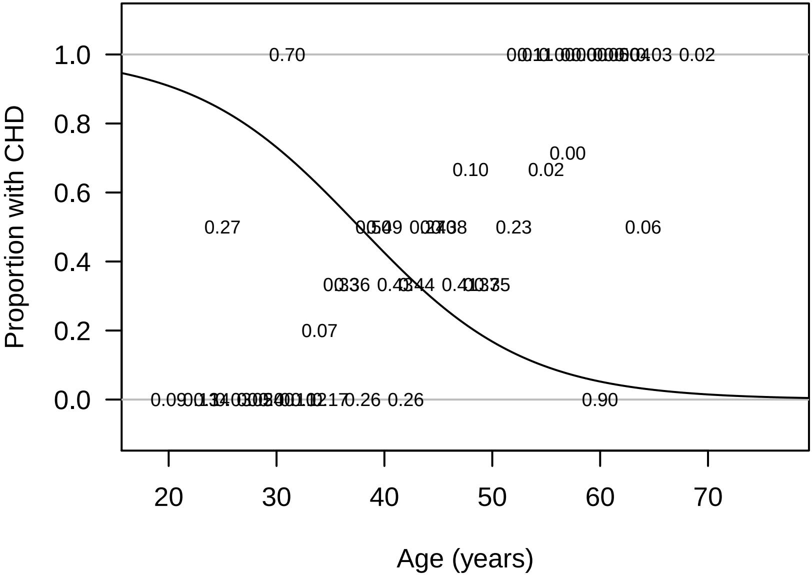
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



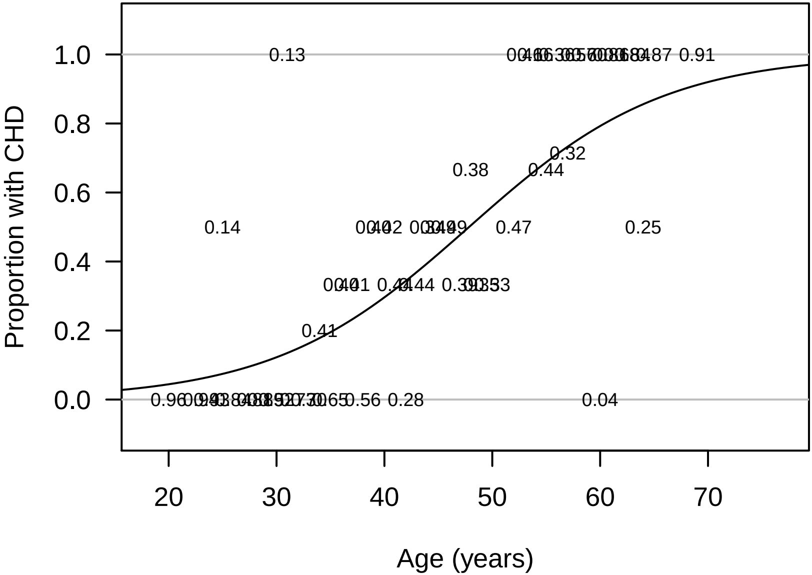
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low contributions



Maximum likelihood estimation

Logistic regression: Good choice of parameters gives high contributions



Maximum likelihood estimation

In practice, instead of dealing with the likelihood function

$$L = \prod_{i=1}^{n} f(y_i; \boldsymbol{\beta}),$$

we deal with the log-likelihood function

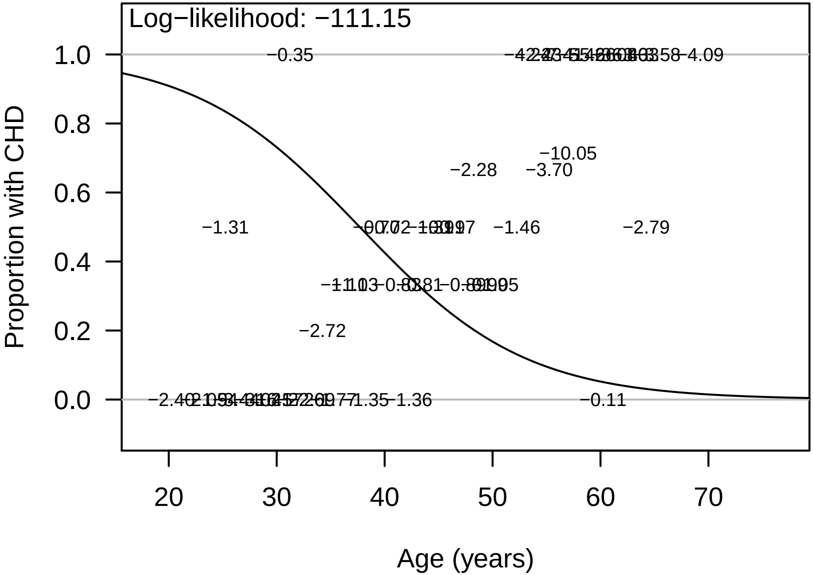
$$\ell = \log(L) = \sum_{i=1}^{n} \log[f(y_i; \boldsymbol{\beta})]$$

Instead of taking the product of the probabilities, we take the log of the probabilities and sum them.

This doesn't affect estimation, because the regression line that maximises the likelihood also maximises the log-likelihood.

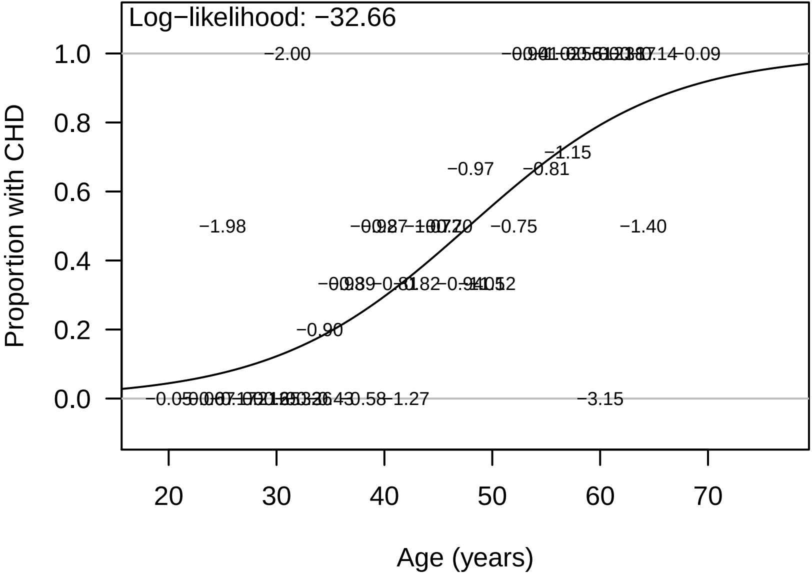
Maximum likelihood estimation

Logistic regression: Poor choice of parameters gives low log-likelihood



Maximum likelihood estimation

Logistic regression: Good choice of parameters gives high log-likelihood



Maximum likelihood estimation

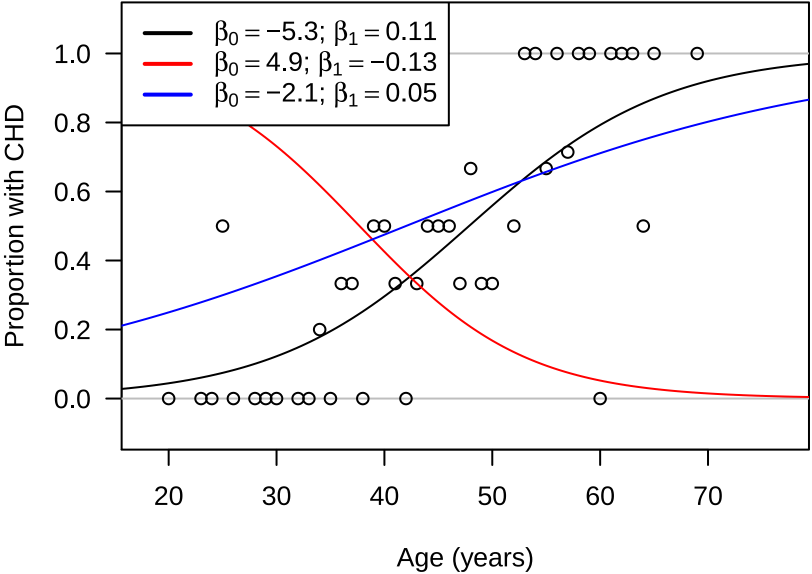
So now we have a measure to tell us how ‘good’ a candidate set of coefficients is

- A ‘better’ set of coefficients will give a higher log-likelihood.
- We estimate the coefficients using the values that give us the highest possible log-likelihood

So instead of minimising the residual sum of squares, for GLMs in general, we maximise the log-likelihood.

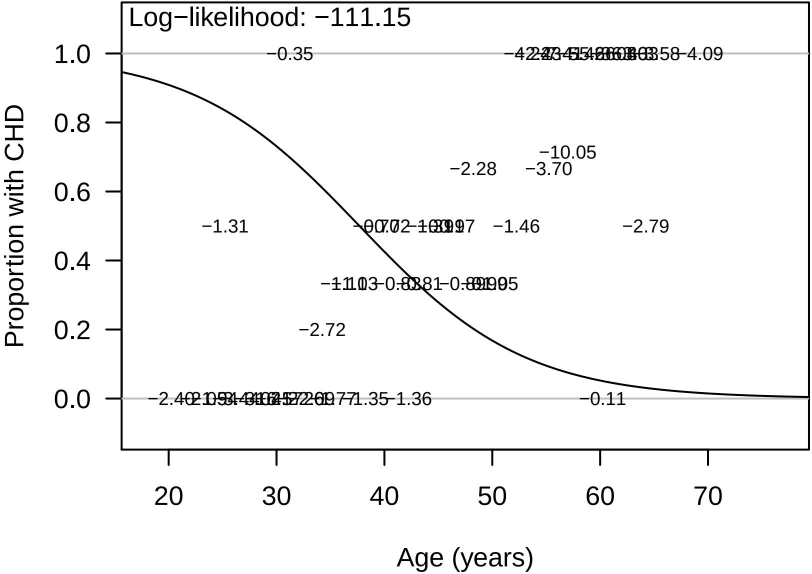
Maximum likelihood estimation

Logistic regression: Coronary heart disease analysis



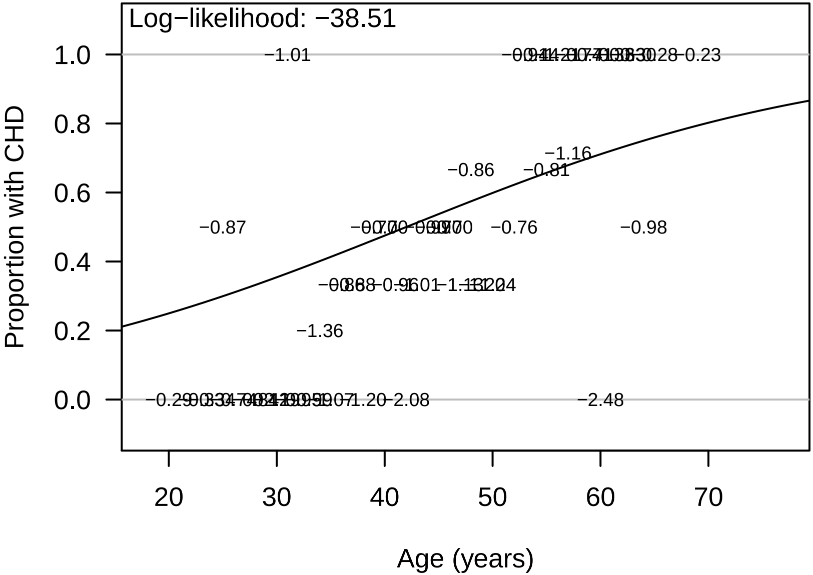
Maximum likelihood estimation

Logistic regression: Bad choice of parameters



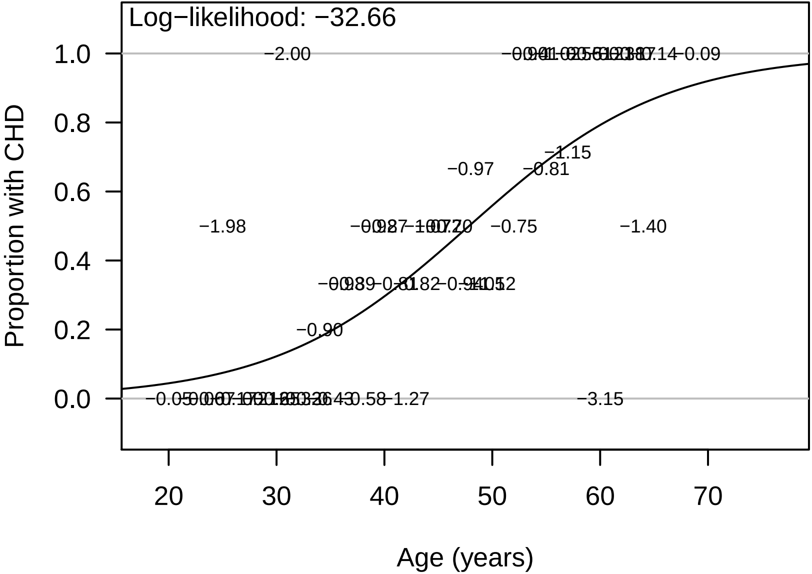
Maximum likelihood estimation

Logistic regression: Better choice of parameters



Maximum likelihood estimation

Logistic regression: Best choice of parameters



Maximum likelihood estimation

Poisson regression

Maximum likelihood estimation works in the same way for Poisson regression, and indeed for a model with any other response distribution.

We can calculate the log-likelihood for a candidate set of coefficients using

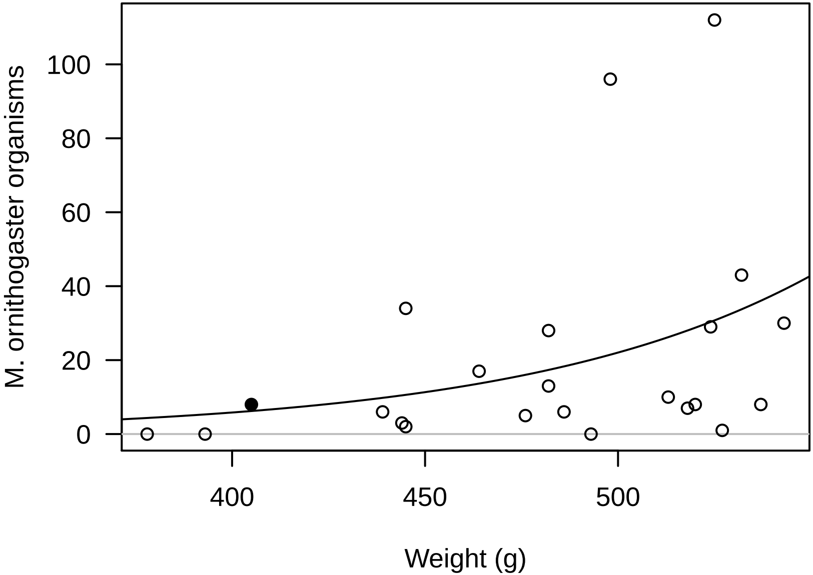
$$\ell = \log(L) = \sum_{i=1}^{n} \log[f(y_i; \boldsymbol{\beta})],$$

but now $f(y_i; \boldsymbol{\beta})$ is the probability mass function of the Poisson distribution.

Our parameter estimates, $\hat{\boldsymbol{\beta}}$, are those that give the largest possible log-likelihood.

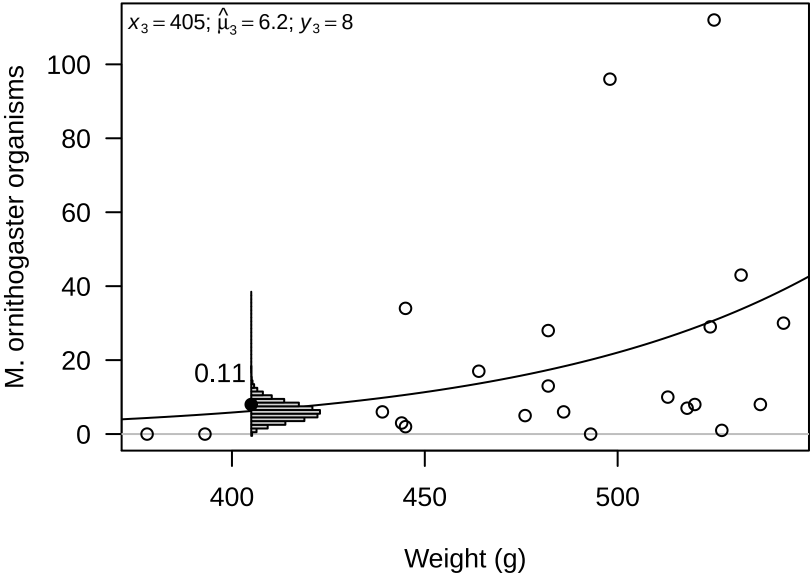
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



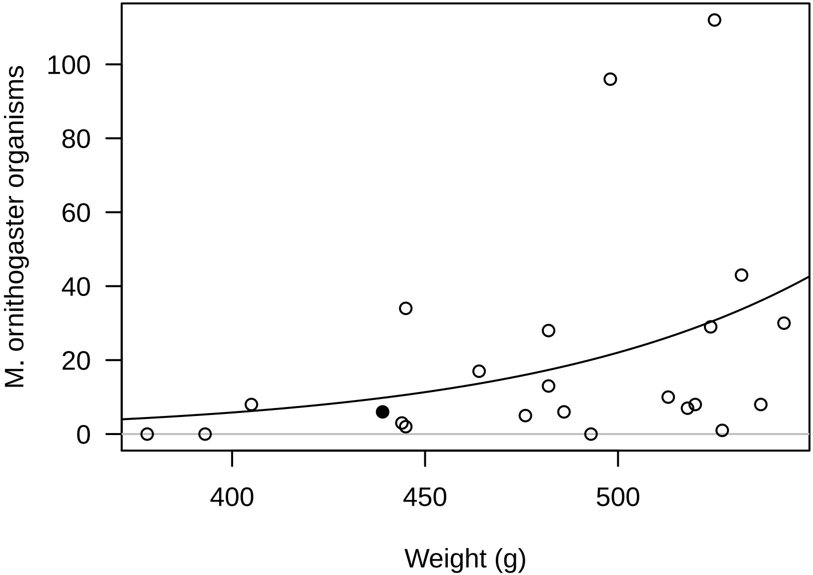
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



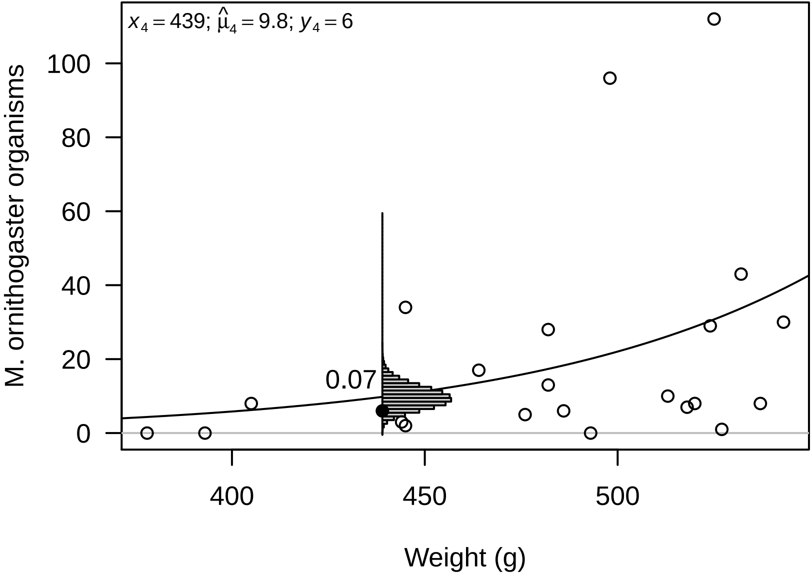
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



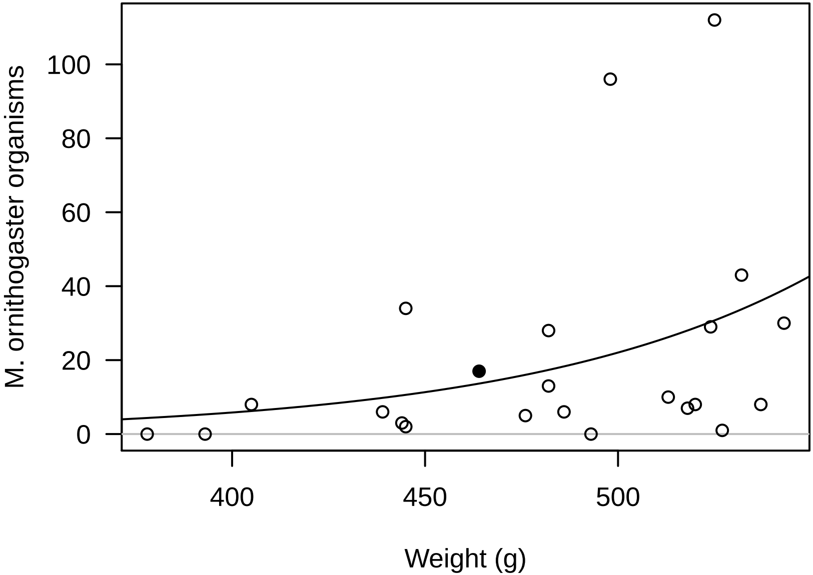
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



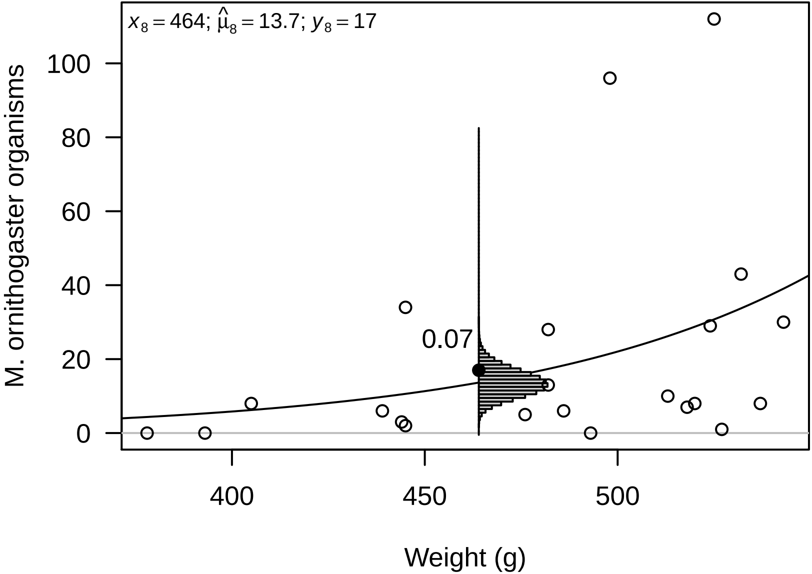
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



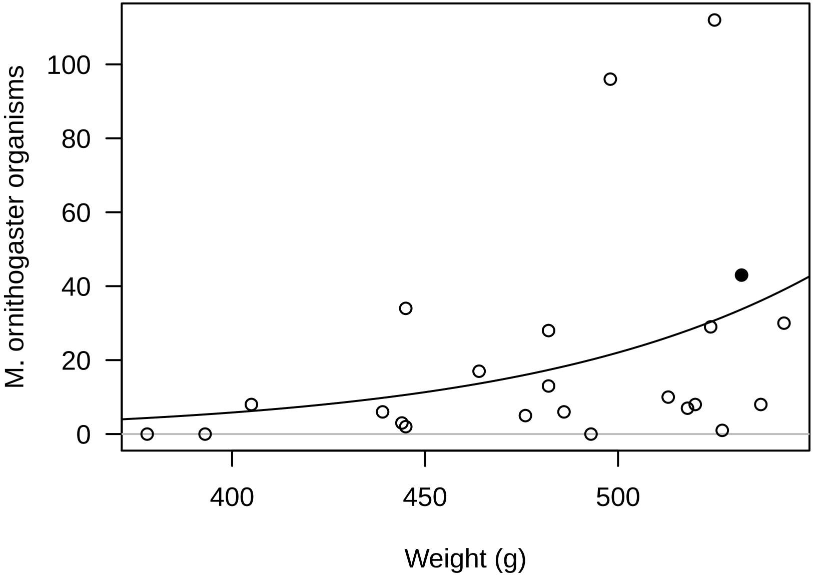
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



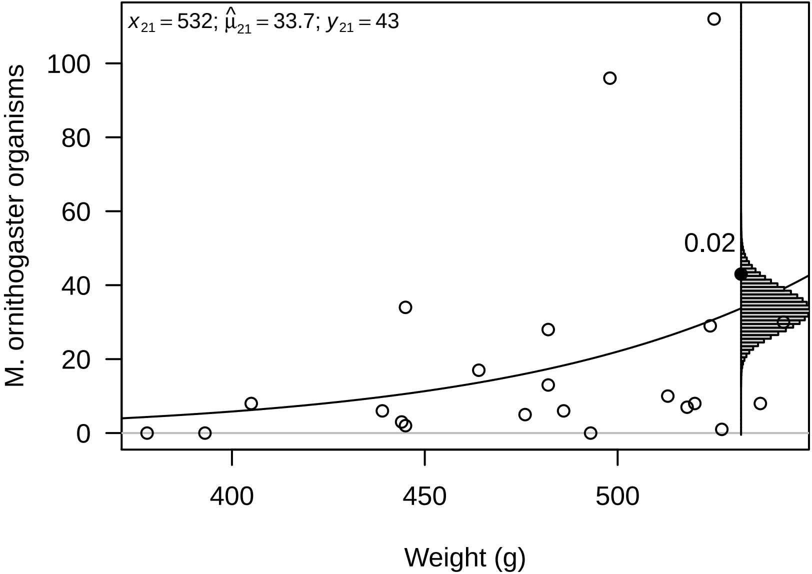
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



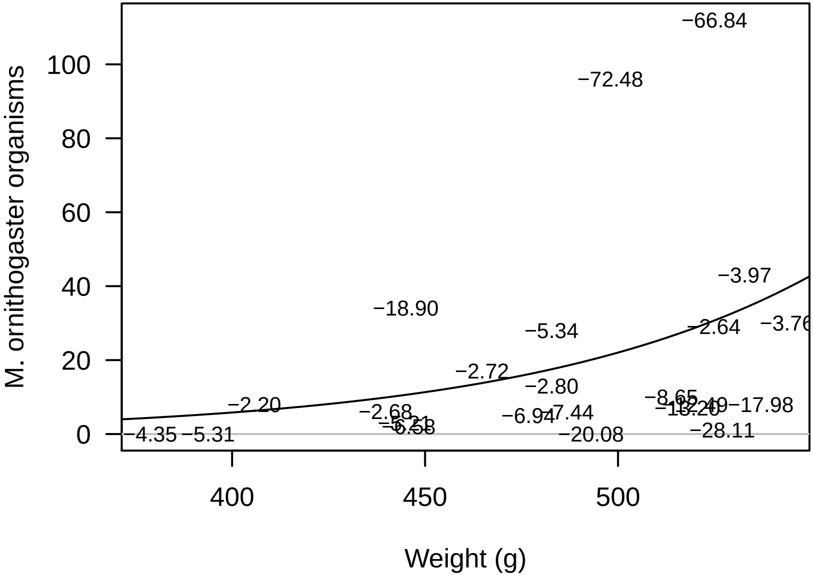
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



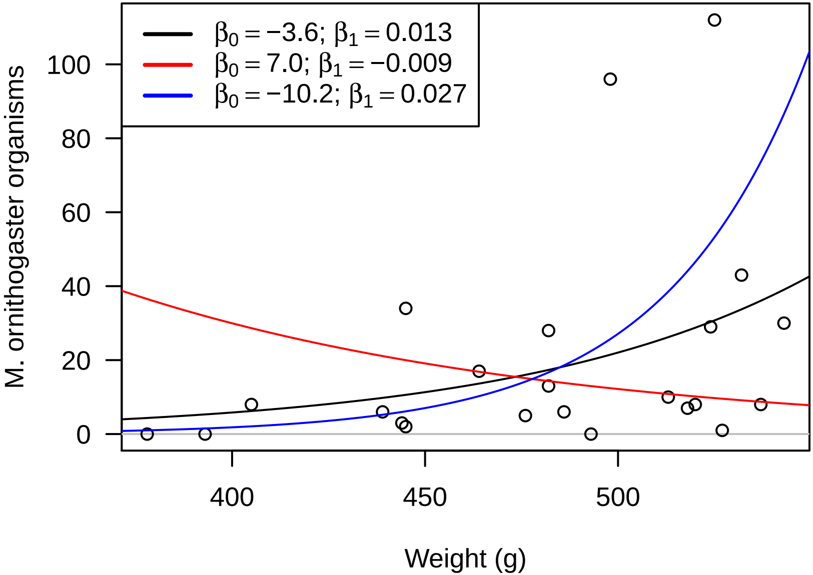
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



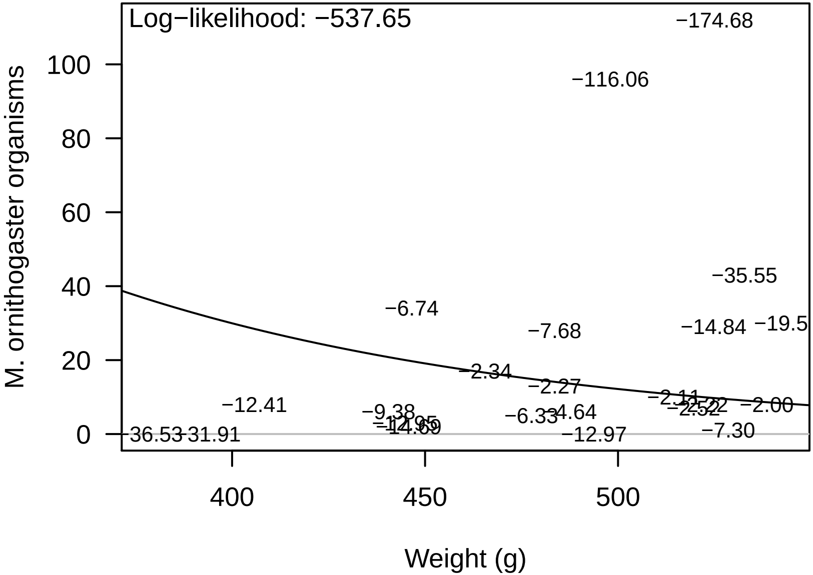
Maximum likelihood estimation

Poisson regression: *Macrorhabdus ornithogaster* chicken analysis



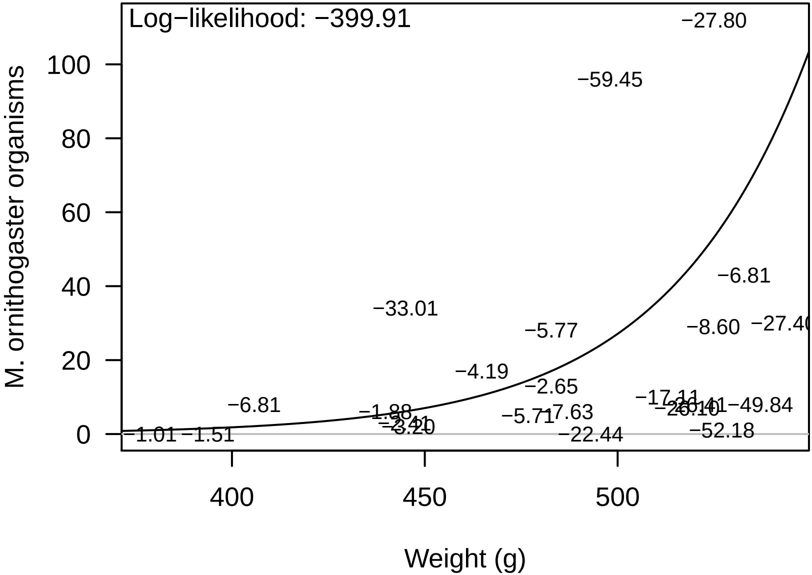
Maximum likelihood estimation

Poisson regression: Bad choice of parameters



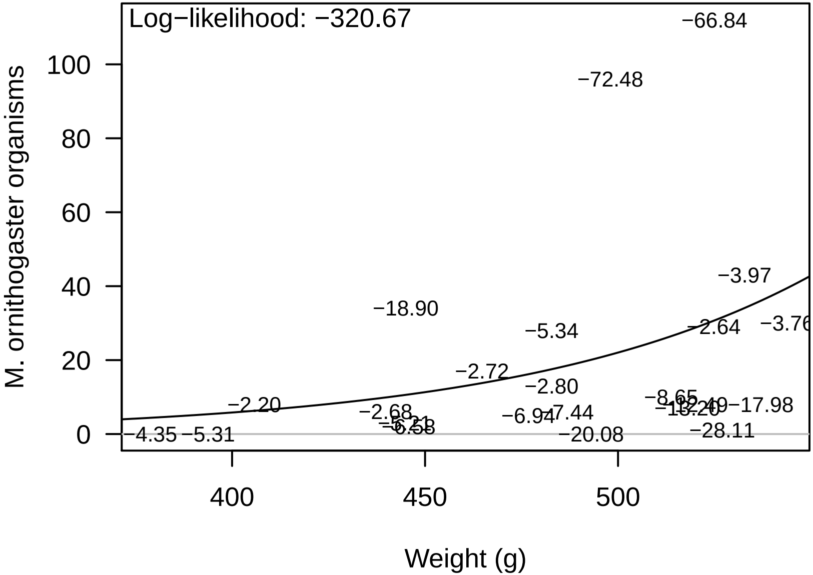
Maximum likelihood estimation

Poisson regression: Better choice of parameters



Maximum likelihood estimation

Poisson regression: Best choice of parameters



Maximum likelihood estimation

The `glm()` function finds the coefficient values that maximise the log-likelihood function. The `logLik()` function extracts the maximised log-likelihood.

```
chickens.fit <- glm(mo ~ weight, family = "poisson")
coef(chickens.fit)

## (Intercept)      weight
## -3.55820556  0.01330221

logLik(chickens.fit)

## 'log Lik.' -320.667 (df=2)

chd.fit <- glm(cbind(y, n - y) ~ age, family = "binomial")
coef(chd.fit)

## (Intercept)         age
##  -5.2784444   0.1103208

logLik(chd.fit)

## 'log Lik.' -32.65503 (df=2)
```