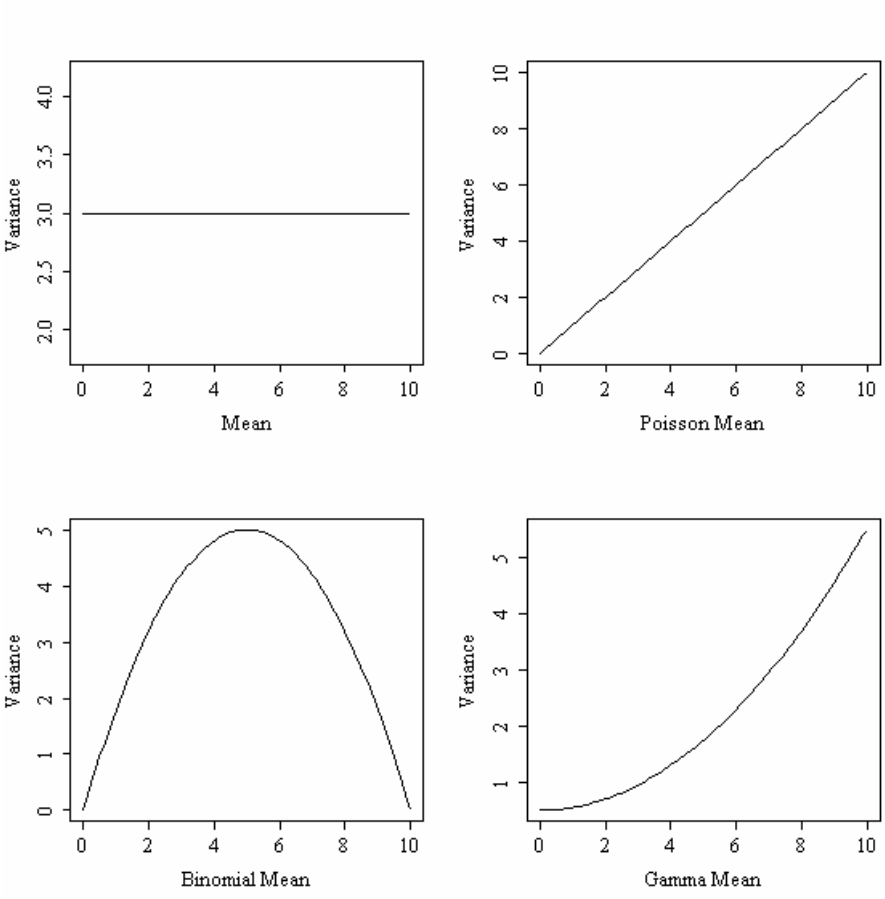
STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

9. GENERALISED LINEAR MODELS

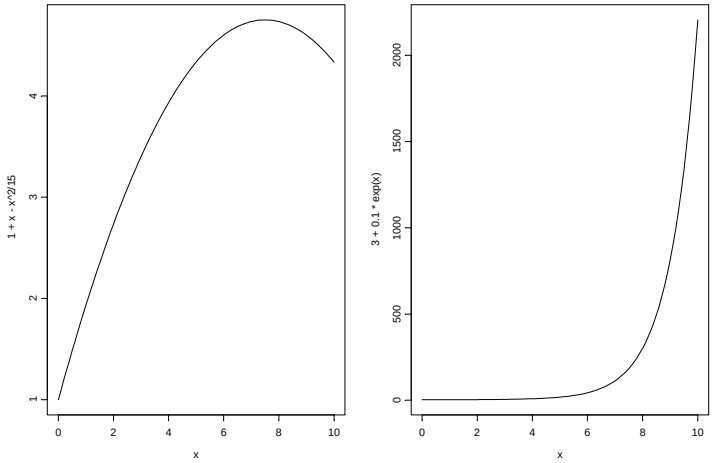
So far, we have assumed that the variance is constant and that the errors are normally distributed. For many kinds of data, one or both of these assumptions is wrong, and we need to be able to deal with this if our analysis is to be unbiased and our interpretations scientifically correct. In count data, for example, where the response variable is an integer and there are often lots of zero's in the data frame, the variance may increase linearly with the mean. With proportion data, where we have a count of the number of failures of an event as well as the number of successes, the variance will be a n-shaped function of the mean. Where the response variable follows a gamma distribution (e.g. in data on time-to-death) the variance increases faster than linearly with the mean. So our assumption has been like the top left panel (where variance is constant) but the data are often like one of the other three panels:



The way to deal with all these problems in a single theoretical framework was discovered by John Nelder who christened the technique Generalised Linear Models (or glims for short). The directive to fit a Generalised Linear Model in S-Plus is **glm**. It is used in exactly the same way as the model fitting directives that are now familiar to you: **aov** and **lm**. There are yet more ways of fitting models that you can discover later if you want to (Generalised Additive Models (**gam**), non-parametric surface fitting models (**loess**), tree models (**tree**) and so on).

A common misconception about generalised linear models (hereafter **glm**'s) is that linear models involve a straight-line relationship between the response variable and the explanatory variables. This is not the case, as you can see from these two linear models. In the plot command use "l" = "lower case L" to generate a line:

```
par(mfrow=c(1,2))
x<-seq(0,10,0.1)
plot(x,1+x-x^2/15,type="l")
plot(x,3+0.1*exp(x),type="l")
```



The definition of a linear model is an equation that contains mathematical variables, parameters and random variables that is *linear in the parameters and in the random variables*. What this means is that if *a*, *b* and *c* are parameters then obviously

$$y = a + bx$$

is a linear model, but so is

$$y = a + bx - cx^2$$

because x^2 can be replaced by z which gives a linear relationship

$$y = a + bx + cz$$

and so is

$$y = a + be^x$$

because we can create a new variable $z = \exp(x)$, so that

$$y = a + bz$$

Some models are non-linear but can be readily linearized by transformation. For example:

$$y = \exp(a + bx)$$

on taking logs of both sides, becomes

$$\ln y = a + bx$$

This kind of relationship is handled in a glm by specifying the log link (see below).

Again, the much-used asymptotic relationship (known in different disciplines as Michaelis-Menten, Briggs-Haldane or Holling 'disk equation') given by:

$$y = \frac{x}{b + ax}$$

is non-linear in the parameter a, but it is readily linearized by taking reciprocals:

$$\frac{1}{y} = a + b\frac{1}{x}$$

Generalised linear models handle this family of equations by a transformation of the explanatory variable ($z = 1/x$) and using the reciprocal link (often, for data like this, associated with gamma rather than normal errors).

Other models are *intrinsically non-linear* because there is no transformation that can linearize them in all the parameters. Some important examples include the hyperbolic function

$$y = a + \frac{b}{c + x}$$

and the asymptotic exponential

$$y = a(1 - be^{-cx})$$

where both models are non-linear unless the parameter c is known in advance. In cases like this, a **glm** is unable to estimate the full set of parameters, and we need to resort to non linear modelling.

Generalised linear models

A generalised linear model has three important properties:

- the *error structure*
- the *linear predictor*
- the *link function*

These are all likely to be unfamiliar concepts. The ideas behind them are straightforward, however, and it is worth learning what each of the concepts involves.

The error structure

Up to this point, we have dealt with the statistical analysis of data with normal errors. In practice, however, many kinds of data have non-normal errors: for example:

- errors that are strongly skewed
- errors that are kurtotic
- errors that are strictly bounded (as in proportions)
- errors that can not lead to negative fitted values (as in counts)

In the past, the only tools available to deal with these problems were transformation of the response variable or the adoption of non-parametric methods. A **glm** allows the specification of a variety of different error distributions:

- Poisson errors, useful with count data
- binomial errors, useful with data on proportions
- gamma errors, useful with data showing a constant coefficient of variation
- exponential errors, useful with data on time to death (survival analysis)

The error structure is defined by means of the **family** directive, used as part of the model formula like this:

```
glm( y ~ z, family = poisson )
```

which means that the response variable *y* has Poisson errors. Or

```
glm(y ~ z, family = binomial )
```

which means that the response is binary, and the model has binomial errors. As with previous models, the explanatory variable *z* can be continuous (leading to a regression analysis) or categorical (leading to an anova-like procedure called analysis of deviance; see below).

The linear predictor

The structure of the model relates each observed y-value to a predicted value. The predicted value is obtained *by transformation of the value emerging from the linear*

predictor. The linear predictor, η (eta), is a linear sum of the effects of one or more explanatory variables, x_j:

$$\eta_i = \sum_{j=1}^{p} x_{ib}\beta_j$$

where the *x*'s are the values of the *p* different explanatory variables, and the β's are the (usually) unknown parameters to be estimated from the data. The right hand side of the equation is called the *linear structure*.

There are as many terms in the linear predictor as there are parameters, *p*, to be estimated from the data. Thus with a simple regression, the linear predictor is the sum of two terms; the intercept and the slope. With a 1-way ANOVA with 4 treatments, the linear predictor is the sum of 4 terms; a mean for each level of the factor. If there are covariates in the model, they add one term each to the linear predictor (the slope of the relationship). Each interaction term in a factorial ANOVA adds one or more parameters to the linear predictor, depending upon the degrees of freedom of each factor (e.g. 3 extra parameters for the interaction between a 2-level factor and a 4-level factor (2-1) x (4-1) = 3).

Fit

To determine the fit of a given model, a **glm** evaluates the linear predictor for each value of the response variable, then compares the observed value with a *back-transformed* value of the linear predictor. The transformation to be employed is specified in the link function (see below). The fitted value is computed by applying the reciprocal of the link function, in order to get back to the original scale of measurement of the response variable. Thus, with a log link, the fitted value is the antilog of the linear predictor, and with the reciprocal link, the fitted value is the reciprocal of the linear predictor.

The link function

One of the difficult things to grasp about **glm** is the relationship between the values of the response variable (as measured in the data and predicted by the model in fitted values) and the linear predictor. The thing to remember is that the *link function relates the mean value of y to its linear predictor*. In symbols, this means that:

$$\eta = g(\mu)$$

which is simple, but needs thinking about. The linear predictor, η (eta), emerges from the linear model as a sum of the terms for each of the *p* parameters. *This is not a value of y* (except in the special case of the *identity link* that we have been using (implicitly) up to now). The value of η is obtained by transforming the value of *y* by the link function, and the predicted value of *y* is obtained by applying the inverse link function to η.

The most frequently used link functions are shown below. An important criterion in the choice of link function is to ensure that the fitted values stay within reasonable bounds. We would want to ensure, for example, that counts were all greater than or equal to zero (negative count data would be nonsense). Similarly, if the response variable was the proportion of animals that died, then the fitted values would have to lie between zero and one (fitted values greater than 1 or less than 0 would be meaningless). In the first case, a log link is appropriate because the fitted values are antilogs of the linear predictor, and all antilogs are greater than or equal to zero. In the second case, the logit link is appropriate because the fitted values are calculated as the antilogs of the log-odds, log(p/q).

By using different link functions, the performance of a variety of models can be compared directly. The total deviance is the same in each case and we can investigate the consequences of altering our assumptions about precisely how a given change in the linear predictor brings about a response in the fitted value of y. The most appropriate link function is the one which produces the minimum residual deviance.

The log link

The log link has many uses, but the most frequent are:

- for count data, where negative fitted values are prohibited
- for explanatory variables that have multiplicative effects, where the log link introduces additivity (remember that “the log of times is plus”)

The model parameters inspected with **summary(model)** are in natural logarithms, and the fitted values are the natural antilogs (**exp**) of the linear predictor.

The logit link

This is the link used for proportion data, and the logit link is generally preferred to the more old fashioned **probit** link. If a fraction *p* of the insects in a bioassay died, then a fraction *q* = (1 - *p*) must have survived out of the original cohort of *n* animals. The logit link is

$$\text{logit} = \ln\left(\frac{p}{q}\right)$$

This is beautifully simple, and it ensures that the fitted values are bounded both above and below (the predicted proportions may not be greater than 1 or less than 0). The details of how the logit link linearizes proportion data are explained in Practical 10.

The logit link does, however, make it a little tedious to calculate *p* from the parameter estimates. Suppose we had a predicted value *x* on the logit scale of -0.328. To get the value of *p* we evaluate (taking care with the signs)

$$p = \frac{1}{1 + e^{-x}}$$

which gives $p = 0.42$. Note that confidence intervals on p will be asymmetric when back-transformed, and it is good practice to draw barcharts and error bars on the logit scale rather than the proportion (or percentage) scale to avoid this problem.

Other link functions

Two other commonly used link functions are the **probit** and the **complementary log-log** links. They are used in bioassay and in dilution analysis respectively, and examples of their use are discussed later.

The names of the links that can be used in R include:

`"logit"`, `"probit"`, `"cloglog"`, `"identity"`, `"log"`, `"sqrt"`, `"1/mu^2"`, `"inverse"`

Use of probits for bioassay is largely traditional, because probit paper used to be available for converting percentage mortality to a linear scale against log dose. Since computers have become widely available the need for the probit transformation

$$\frac{y_i}{n_i} = \Phi(\eta_i) + \varepsilon_i$$

has declined. The proportion responding (y/n) is linked to the linear predictor by $\Phi(.)$, the unit normal probability integral. Because the logit is so much simpler to interpret, and because the results of modelling with the two transformations are almost always identical, the logit link function is nowadays recommended for bioassay work, even though probits are based on a reasonable distributional argument for the tolerance levels of individuals.

The complementary log-log link:

$$\theta = \ln[-\ln(1-p)]$$

is not symmetrical about p=0.5 and is often used in simple dilution assay. If the proportion of tubes containing bacteria p is related to dilution x like this:

$$p = 1 - e^{-\lambda x}$$

then the complementary log-log transformation gives

$$\eta = \ln[-\ln(1-p)] = \ln\lambda + \ln x$$

which means that the linear predictor has a slope of 1 when plotted against $\ln(x)$. We fit the model, therefore, with $\ln(x)$ as an offset, and **glm** estimates the maximum likelihood value of $\ln(\lambda)$.

The complementary log-log link should be assessed during model criticism for binary data and for data on proportional responses (see Practical 10). It will sometimes lead to a lower residual deviance than the symmetrical logit link.

Canonical link functions

The canonical link functions are the default options employed when a particular error structure is specified in the **family** directive in the model formula. Omission of a **link** directive means that the following settings are used:

Error	Canonical link
normal	identity
Poisson	log
binomial	logit
gamma	reciprocal

You should try to memorise these canonical links and to understand why each is appropriate to its associated error distribution.

The likelihood function

The concept of maximum likelihood is unfamiliar to most non-statisticians. Fortunately, the methods that scientists have encountered in linear regression and traditional ANOVA (i.e. least squares) are the maximum likelihood estimators when the data have normal errors and the model has an identity link. For other kinds of error structure and different link functions, however, the methods of least squares do not give unbiased parameter estimates, and maximum likelihood methods are preferred. It is easiest to see what maximum likelihood involves by working through two simple examples based on the binomial and Poisson distributions.

The binomial distribution

Suppose we have carried out a single trial, and have found $r = 5$ parasitised animals out of a sample of $n = 9$ insects. Our intuitive estimate of the proportion parasitised is $5/9 = r/n = 0.555$. What is the maximum likelihood estimate of the proportion parasitised? With n=9 and r=5 the formula for the binomial looks like this:

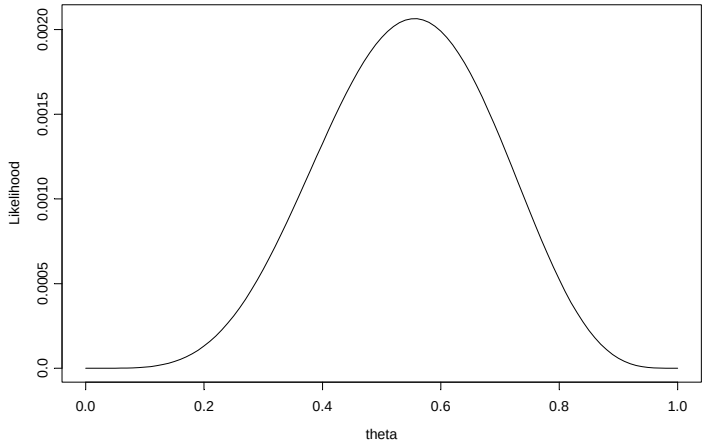
$$P(5) = \left(\frac{9!}{(9-5)! \times 5!} \right) \theta^5 (1-\theta)^{(9-5)}$$

Now the likelihood L does not depend upon the combinatorial part of the formula, because θ , the parameter we are trying to estimate, does not appear there. This simplifies the problem, because all we need to do now is to find the value of θ which maximises the likelihood

$$L(\theta) = \theta^5 (1-\theta)^{(9-5)}$$

To do this we might plot L(θ) against θ like this

```
theta<-seq(0,1,.01)
par(mfrow=c(1,1))
plot(theta,theta^5*(1-theta)^4,type="l",ylab="Likelihood")
```



from which it is clear that the maximum likelihood occurs at = *r/n* = 5/9 = 0.555. It is reassuring that our intuitive estimate of the proportion parasitised is *r/n* as well.

A more general way to find the maximum likelihood estimate of θ is to use calculus. We need to find the derivative of the likelihood with respect to θ, then set this to zero, and solve for θ. In the present case it is easier to work with the log of the likelihood. Obviously, the maximum likelihood and the maximum log likelihood will occur at the same value of θ .

$$\text{log likelihood} = r\ln(\theta) + (n-r)\ln(1-\theta)$$

so the derivative of the log likelihood with respect to θ is:

$$\frac{dL(\theta)}{d\theta} = \frac{r}{\theta} - \frac{n-r}{1-\theta}$$

remembering that the derivative of $\ln\theta$ is $1/\theta$ and of ln (1-θ) is -1/(1-θ). We set this to zero, rearrange, then take reciprocals to find θ:

$$\frac{r}{\theta} = \frac{n-r}{1-\theta} \quad \text{so} \quad \theta = \frac{r}{n}$$

The maximum likelihood estimate of the binomial parameter is the same as our intuitive estimate.

The Poisson distribution

As a second example, we take the problem of finding the maximum likelihood estimate of μ for a Poisson process in which we observed, say, *r* lightening strikes per parish in *n* parishes, giving a total of $\sum r$ lightening strikes in all. The probability density function for the number of strikes per parish is

$$f(r) = \frac{e^{-\mu}\mu^{r}}{r!}$$

so the initial likelihood is the density function multiplied by itself as many times as there are individual parishes:

$$L(\mu) = \prod_{1}^{n} \frac{e^{-\mu}\mu^{r}}{r!} = e^{-n\mu}\mu^{\sum r}$$

because the constant r! can be ignored. Note that *nr* is replaced by $\sum r$ the observed total number of lightening strikes. Now it is straightforward to obtain the log likelihood:

$$L(\mu) = -n\mu + \sum r \ln \mu$$

The next step is to find the derivative of the log likelihood with respect to μ :

$$\frac{dL(\mu)}{d\mu} = -n + \frac{\sum r}{\mu}$$

We set this to zero, and rearrange to obtain

$$\mu = \frac{\sum r}{n}$$

Again, the maximum likelihood estimator for the single parameter of the Poisson distribution conforms with intuition; it is the mean (in this case, the mean number of lightning strikes per parish).

Maximum likelihood estimation

The object is to determine the values for the parameters of the model that lead to the best fit of the model to the data. It is in the definition of what constitutes 'best' that maximum likelihood methods can differ from the more familiar, least squares estimates. This is how it works

- given the data
- and given our choice of model
- what values of the parameters

- make the observed data most likely ?

The data are sacrosanct, and they tell us what actually happened under a given set of circumstances. It is a common mistake to say 'the data were fitted to the model' as if the data were something flexible, and we had a clear picture of the structure of the model. On the contrary, what we are looking for is the minimal adequate model to describe the data. The model is fit to data, not the other way around. The best model is the model that produces the minimal residual deviance, subject to the constraint that all the parameters in the model should be statistically significant.

You have to specify the model. It embodies your best hypothesis about the factors involved, and the way they are related to the response variable. We want the model to be minimal because of the principle of parsimony, and adequate because there is no point in retaining an inadequate model that does not describe a significant fraction of the variation in the data. It is very important to understand that *there is not one model*; this is one of the common implicit errors involved in traditional regression and anova, where the same models are used, often uncritically, over and over again. In most circumstances, there will be a large number of different, more or less plausible models that might be fit to any given set of data. Part of the job of data analysis is to determine which, if any, of the possible models are adequate, and then, out of the set of adequate models, which is the minimal adequate model. In some cases there may be no single best model and a set of different models may all describe the data equally well.

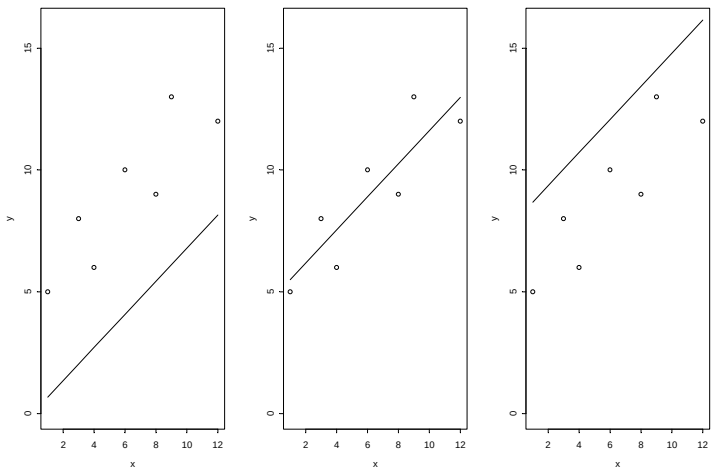
Here are the data:

```
x<-c(1,3,4,6,8,9,12)
y<-c(5,8,6,10,9,13,12)
```

and here is the model

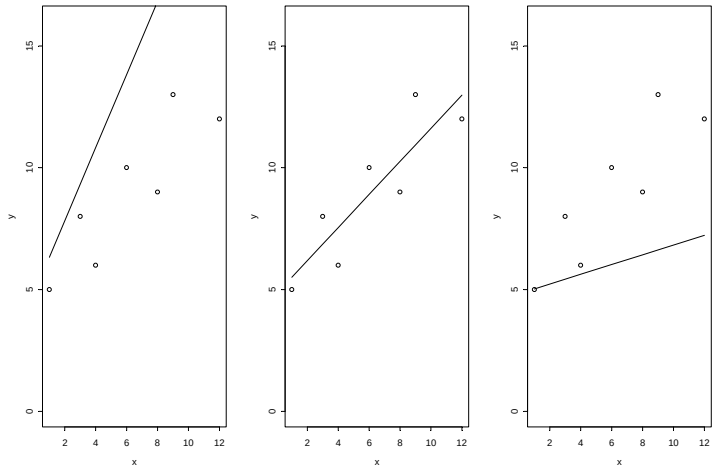
$$y = a + bx$$

Suppose that we know that the slope is 0.68, then the maximum likelihood question can be applied to the intercept *a*. If the intercept were 0 (left graph), would the data be likely?



The answer of course, is no. If the intercept were 8 (right graph) would the data be likely? Again, the answer is obviously no. The maximum likelihood estimate of the intercept is shown in the central graph (its value turns out to be 4.827).

We could have a similar debate about the slope. Suppose we knew that the intercept was 4.827, would the data be likely if the graph had a slope of 1.5 (left graph)?



The answer, of course, is no. What about a slope of 0.2 (right graph)? Again, the data are not at all likely if the graph has such a gentle slope. The maximum likelihood of the data is obtained with a slope of 0.679 (centre graph).

This is not how the procedure is carried out, but it makes the point that we judge the model of the basis *how likely the data would be if the model were correct*. In practice of course, the parameters are estimated simultaneously.

Parameter estimation in generalised linear models

The method of parameter estimation is *iterative, weighted least-squares*. You know about least-squares methods from Practicals 4, 5 and 6. A **glm** is different in that the regression is not carried out on the response variable, y, but on *a linearized version of the link function applied to y*. The *weights* are functions of the fitted values. The procedure is *iterative* because both the adjusted response variable and the weight depend upon the fitted values.

This is how it works (the technical details are on pp 31-34 in McCullagh & Nelder, 1983). Take the data themselves as starting values for estimates of the fitted values. Use this to derive the linear predictor, the derivative of the linear predictor ($d\eta / d\mu$) and the variance function. Then re-estimate the adjusted response variable *z* and the weight W, as follows:

$$z_0 = \eta_0 + (y - \mu_0)\left(\frac{d\eta}{d\mu}\right)_0$$

where the derivative of the link function is evaluated at μ_0 and

$$W_0^{-1} = \left(\frac{d\eta}{d\mu}\right)_0^2 V_0$$

where V_O is the variance function of *y* (see below). Keep repeating the cycle until the changes in the parameter estimates are sufficiently small. It is the difference ($y - \mu_0$) between the data *y* and the fitted values μ_0 that lies at the heart of the procedure. The maximum likelihood parameter estimates are given by

$$\sum W(y - \mu) \frac{d\eta}{d\mu} x_i = 0$$

for each explanatory variable x_i (summation is over the rows of the data frame). For more detail, see McCullagh & Nelder (1989) and Aitkin et al. (1989); a good general introduction to the methods of maximum likelihood is to be found in Edwards (1972).

Deviance: measuring the goodness of fit of a glm.

The fitted values produced by the model are most unlikely to match the values of the data perfectly. The size of the discrepancy between the model and the data is a measure of the inadequacy of the model; a small discrepancy may be tolerable, but a large one will not be. The measure of discrepancy in a **glm** used to assess the goodness of fit of the model to the data is called the *deviance*. Deviance is defined as –2 times the difference in log likelihood between the current model and a saturated model (i.e. a model that fits the data perfectly). Because the latter does not depend on the parameters of the model, minimising the deviance is the same as maximising the likelihood.

Deviance is estimated in different ways for different families within glm. Numerical examples of the calculation of deviance for different glm families are in Practical 11 (Poisson errors), Practical 10 (binomial errors) and Practical 12 (gamma errors).

Table. Deviance formulas for different families of glm. y is observed data, $\bar{y}$ mean value of y, μ fitted values of y from the maximum likelihood model, n is the binomial denominator in a binomial error glm.

Family (Error structure)	Deviance
Normal	$\sum (y - \bar{y})^2$
Poisson	$2\sum y \ln(y / \mu) - (y - \mu)$
Binomial	$2\sum y \ln(y / \mu) + (n - y) \ln(n - y) / (n - \mu)$
Gamma	$2\sum (y - \mu) / y - \ln(y / \mu)$
Inverse Gaussian	$\sum (y - \mu)^2 / (\mu^2 y)$

The following 3 practicals deal with the 3 main applications of **glm**'s

- proportion data using binomial errors
- count data using Poisson errors
- survival data using various error distributions

In this practical, we investigate the use of a range of different **link functions**

```
timber<-read.table("c:\\temp\\timber.txt", header=T)
attach(timber)
names(timber)
```

```
[1] "volume" "girth"  "height"
```

We begin with data inspection: how does timber volume depend upon girth and height? Volume of a cylinder is

$$v = \pi r^2 h$$

(i.e. the cross-sectional area times the height). This means that we expect volume to be proportional to height and proportional to the square of girth (since $g = 2\pi r$). Also, we note that the model for volume is multiplicative rather than additive. Taking logs of both sides gives

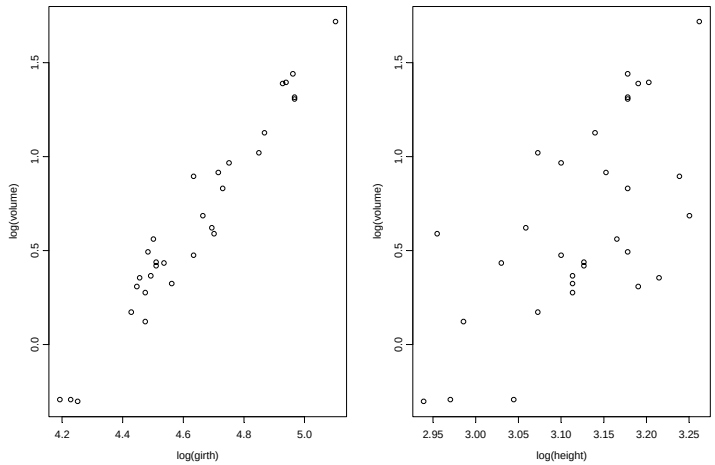
$$\ln(v) = \ln(\pi) + 2\ln(r) + \ln(h)$$

which is useful because it makes the relationship linear and additive. If we regress log volume on log height we expect the slope of the graph to be 1. If we regress log volume on girth we expect the slope of the graph to be 2. And if the timber really is cylindrical (rather than tapered or conical) then the intercept will be a function of pi. Let's rewrite the equation with girth in place of radius and see if we can predict what the intercept of a multiple regression should be:

$$\ln(v) = \ln(\pi) + 2\ln(r) + \ln(h) = \ln(\pi) + 2\ln(g/2\pi) + \ln(h) = \ln(1/4\pi) + 2\ln(g) + \ln(h)$$

so we expect the intercept of a log-log plot to be $\ln(1/4\pi) =$ -2.531024. Let's see what these data look like:

```
par(mfrow=c(1,2))
plot(log(girth),log(volume))
plot(log(height),log(volume))
```



The relationship between timber volume and girth is very close (left), but the relationship with height is much more noisy (right). It is sensible to convert girth into metres, so that all of the units are in the same currency:

```
girth<-girth/100
```

We begin by testing the prediction that the slopes of the graph should be 2 and 1 for the regressions of log volume on log girth and log height respectively:

```
model1<-lm(log(volume)~log(girth)+log(height))
summary(model1)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.89938    0.63767  -4.547 9.56e-05 ***
log(girth)   1.98267    0.07503  26.426  < 2e-16 ***
log(height)  1.11714    0.20448   5.463 7.83e-06 ***
```

The values are close to our predictions: log(girth) has a slope of 1.98 against a predicted value of 2.0, log(height) has 1.12 against a predicted value of 1.0 and the intercept is -2.8994 against a predicted value of -2.531024. A simple test of whether the observed and predicted values are significantly different is to carry out 3 t-tests, dividing the differences between observed and predicted parameter values by their standard errors:

```
(2-1.9827)/0.075
```

```
[1] 0.2306667
```

```
(1-1.1171)/0.2045
```

```
[1] -0.5726161
```

```
(2.8994 -2.531024)/0.6377
```

```
[1] 0.5776635
```

So there is no evidence of any significant difference there: none of the t values is bigger than 2.0. An alternative is to test whether a different model explains the data any less well than the full regression. In this alternative model, we constrain the two slopes to be exactly 2.0 and1.0 by the use of an **offset**, like this:

```
model1<-glm(log(volume)~log(girth)+log(height))
model2<-glm(log(volume)~offset(2*log(girth)+log(height)))
```

Now we can compare the 3 models using **anova** to see how the full model and the model with the offset constraining the 2 slopes to exactly 2.0 and 1.0 compare with

```
anova(model1,model2,test="F")
```

```
Analysis of Deviance Table

Model 1: log(volume) ~ log(girth) + log(height)
Model 2: log(volume) ~ offset(2 *log(girth)+log(height))

  Resid. Df Resid. Dev Df  Deviance      F Pr(>F)
1        28   0.185548
2        30   0.187772 -2 -0.002223 0.1677 0.8464
```

Note that in order for **offset** to work properly, we need to fit the models as **glm** rather than **lm** (an offset is strictly redundant in a Gaussian model, because in principle one can work directly with the residuals).

The **anova** shows that the reduction in explanatory power of the offset model is only 0.002223 compared with the full model, despite a saving of 2 degrees of freedom. This is assessed by an F test as (0.002223/2) / (0.185548/28) = 0.1677 compared with a value of 3.34 in tables (p = 0.8464).

What about the shape of the timber? Theory might have predicted that the taper on the logs would be enough to prefer a conical model to a cylindrical model. A cone would have an intercept of –3.630 on a log-log plot, while as we have seen, a cylinder would have –2.531. We can specify this full model in an offset and fit the model without an intercept (by specifying –1 in the model formula to remove the intercept):

```
model3<-glm(log(volume)~-1+offset(-2.531 + 2*log(girth)+log(height)))
anova(model1,model2,model3,test="F")
```

```
Analysis of Deviance Table

Model 1: log(volume) ~ log(girth) + log(height)
Model 2: log(volume) ~ offset(2 * log(girth)+log(height))
Model 3: log(volume) ~ -1 +
                       offset(-2.531+2*log(girth)+log(height))

  Resid. Df Resid. Dev Df  Deviance      F Pr(>F)
1        28   0.185548
2        30   0.187772 -2 -0.002223 0.1677 0.8464
3        31   0.188057 -1 -0.000285 0.0430 0.8372
```

The model3 in which we specify all 3 parameters is not significantly worse than model1 in which all 3 parameters were estimated from the data. The data therefore provide no support for a model for the volume of this timber that is any more complicated than a cylinder.

Transformations of the response and explanatory variables

We can now assess the explanatory power of a variety of different transformations. The first model transformed both explanatory variables to a log scale, making the model additive and dimensionally consistent. A different way of making the model dimensionally consistent is to take the cube root of the volume as the response variable, in which case all of the variables on both sides of the equation have units of metres.

```
model4<-glm(volume^(0.3333)~girth+height)

summary(model4)
```

```
Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.035478   0.076824  -0.462     0.648
girth        0.791345   0.029453  26.868  < 2e-016 ***
height       0.020104   0.003858   5.211 1.56e-005 ***

    Null deviance: 1.492964  on 30  degrees of freedom
Residual deviance: 0.033371  on 28  degrees of freedom
AIC: -115.88
```

so r^2 is (1.492964 – 0.033371) / 1.492964 = 0.9776478.

Finally, we can use the **quasi** function to allow the use of **anova** to compare different link functions like log, power or reciprocal. Recall that if we transform the response variable in different ways (e.g. log in one case, cube root in another case) then we can not compare the resulting models using anova. Models 5-7 fit the same model

volume~girth+height

using a different link function in each case: model5 uses the cube root link (this is specified as **power(0.333)**), model6 uses the **log** link, while model7 uses the reciprocal link (this is specified as **inverse**).

```
model5<-glm(volume~girth+height,family=quasi(link=power(0.3333)))

model6<-glm(volume~girth+height,family=quasi(link=log))

model7<-glm(volume~girth+height,family=quasi(link=inverse))
```

The beauty of this approach is that we can now compare the models using **anova** because they all have the same response variable (i.e. untransformed volume).

anova(model5,model6,model7,test="F")

```
Analysis of Deviance Table

Model 1: volume ~ girth + height
Model 2: volume ~ girth + height
Model 3: volume ~ girth + height

  Resid. Df Resid. Dev Df Deviance
1        28     0.9655
2        28     1.4291  0  -0.4636
3        28     5.3189  0  -3.8898
```

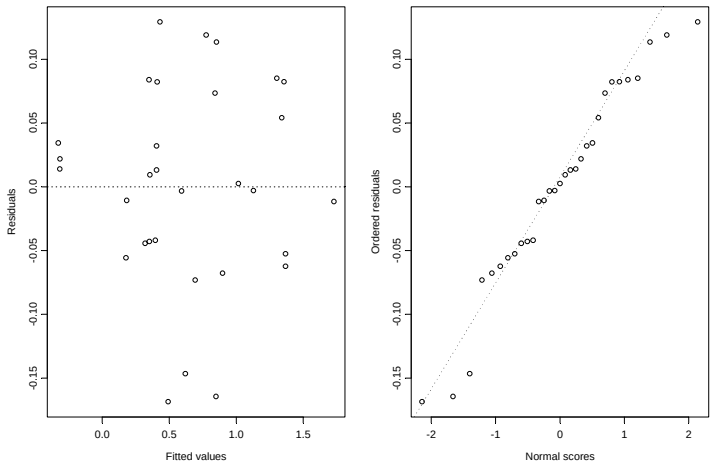
The cube root link is better than the log link (residual deviance of 0.966 compared with 1.429) and both are much better than the inverse link (deviance = 5.319). Note that the **anova** table gives us no indication of how the 3 models differ from one another in their link functions, so we need to be very careful in our book-keeping.

So which is the best model for these data ? We have 3 serious candidates:

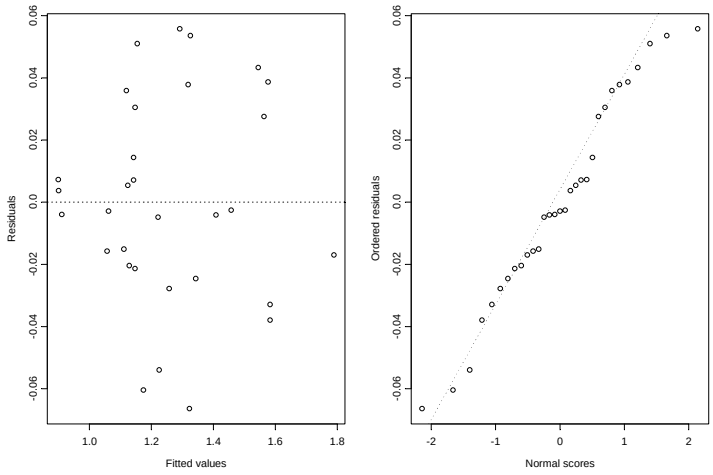
Response	**Model**	**Link**	**r^2**
log(volume)	1: log(girth)+log(volume)	identity	0.9777
(volume)^0.333	4: girth+volume	identity	0.9776
volume	5: girth+volume	power(0.333)	0.9773

We can not compare them using **anova** because they all have different response variables. We can compute their r^2 values, and this suggests that our original log-log transformed model is best (but there is precious little in it). We can inspect their residuals using **plot(model**) : model 1, then model 6, then model 8

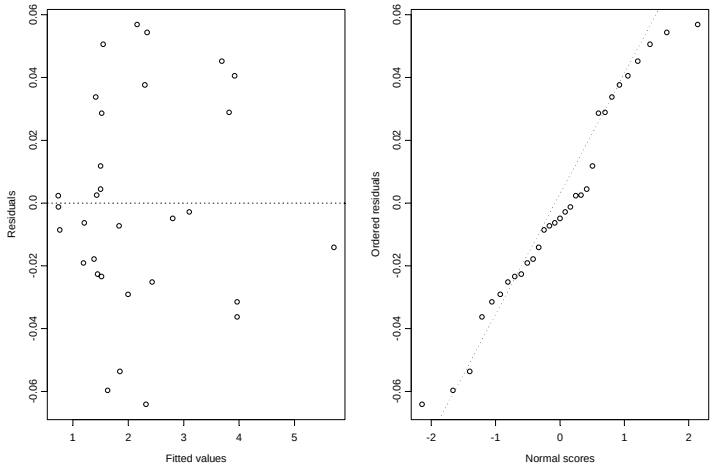
log(volume) ~ log(girth) + log(height)



(volume^0.333) ~ girth + height



volume ~ girth + height, family = quasi(link = power(0.3333))



Again, there is little to choose between them, but the power link has the worst qq-plot (the slope in the central part is shallower than the 1 to 1 line). Overall, be would probably go for the original model, involving log-log transformation, because:

- it is dimensionless on both sides
- it is additive in a way that makes good mathematical sense ($vol \propto g^2 \times h$)
- it has the highest r^2
- its residuals are well behaved

Box Cox transformations

Sometimes it is not clear from theory what the optimal transformation of the response variable should be. In these circumstances, the Box Cox transformation offers a simple empirical solution. The idea is to find the power transformation, λ (lambda), that maximises the likelihood when a specified set of explanatory variables is fit to

$$\frac{y^{\lambda}-1}{\lambda}$$

as the response. The value of lambda can be positive or negative, but it can't be zero (you would get a zero-divide error when the formula was applied to the response variable, y). For the case $\lambda = 0$ the Box Cox transformation is defined as log(y). Suppose that $\lambda = -1$. The formula now looks like this:

$$\frac{y^{-1}-1}{-1} = \frac{\frac{1}{y}-1}{-1} = 1-\frac{1}{y}$$

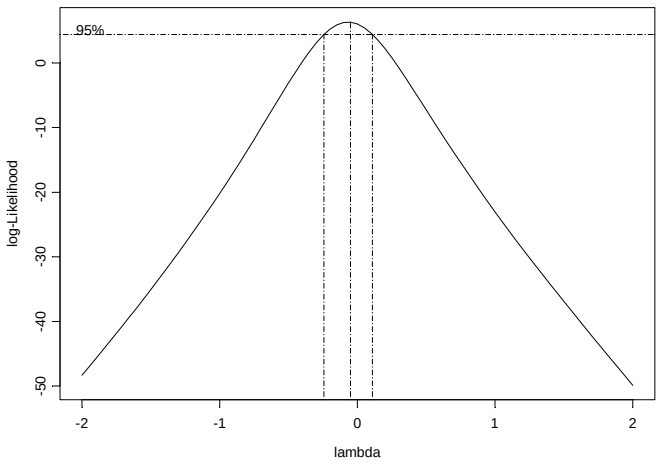
and this quantity is regressed against the explanatory variables and the log likelihood computed.

We start by loading the MASS library of Venables and Ripley:

```
library(MASS)
```

The **boxcox** function is very easy to use: just specify the model formula, and the default options take care of everything else

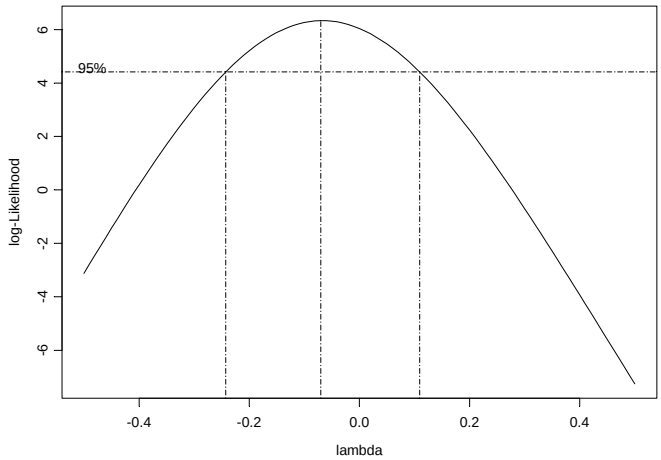
```
boxcox(volume~log(girth)+log(height))
```



It is clear that the optimal value of lambda is close to zero (i.e. the log transformation). We can zoom in to get a more accurate estimate by specifying our

own, non-default, range of lambda values. It looks as if it would be sensible to plot from –0.5 to +0.5:

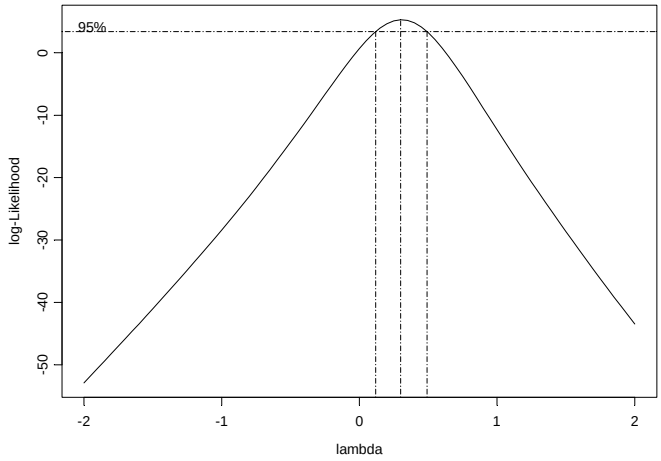
```
boxcox(volume~log(girth)+log(height),lambda=seq(-0.5,0.5,0.01))
```



The likelihood is maximised at lambda about –0.08, but the log likelihood for lambda = 0 is very close to the maximum. This also gives a much more straightforward interpretation, so we would go with that, and model log(volume) as a function of log(girth) and log(height).

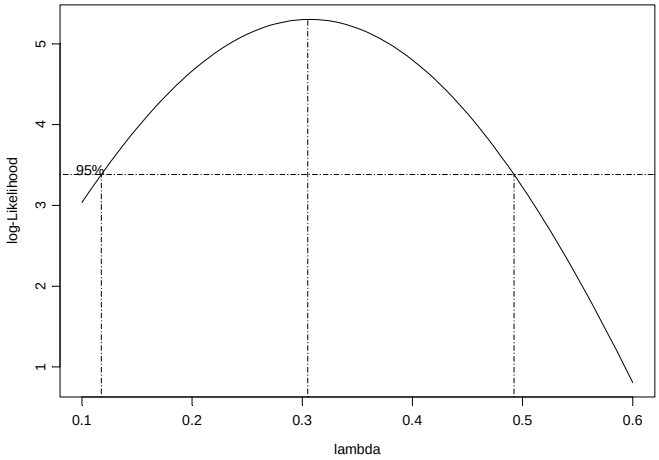
What if we had not log-transformed the explanatory variables? What would have been the optimal transformation of volume in that case? To find out, we re-run the **boxcox** function, simply changing the model formula like this:

```
boxcox(volume~girth+height)
```



We can zoom in from 0.1 to 0.6 like this:

```
boxcox(volume~girth+height,lambda=seq(0.1,0.6,0.01))
```



This suggests that the cube root transformation would be best ($\lambda = 1/3$). Again, this accords with dimensional arguments, since the response and explanatory variables would all have dimensions L in this case.