GENERALISED LINEAR MODELLING
LECTURE NOTES: BASICS OF GENERAL LINEAR MODELLING

I. Introduction

We begin this section of the course by examining an example which will highlight the limitations of the classic linear model.

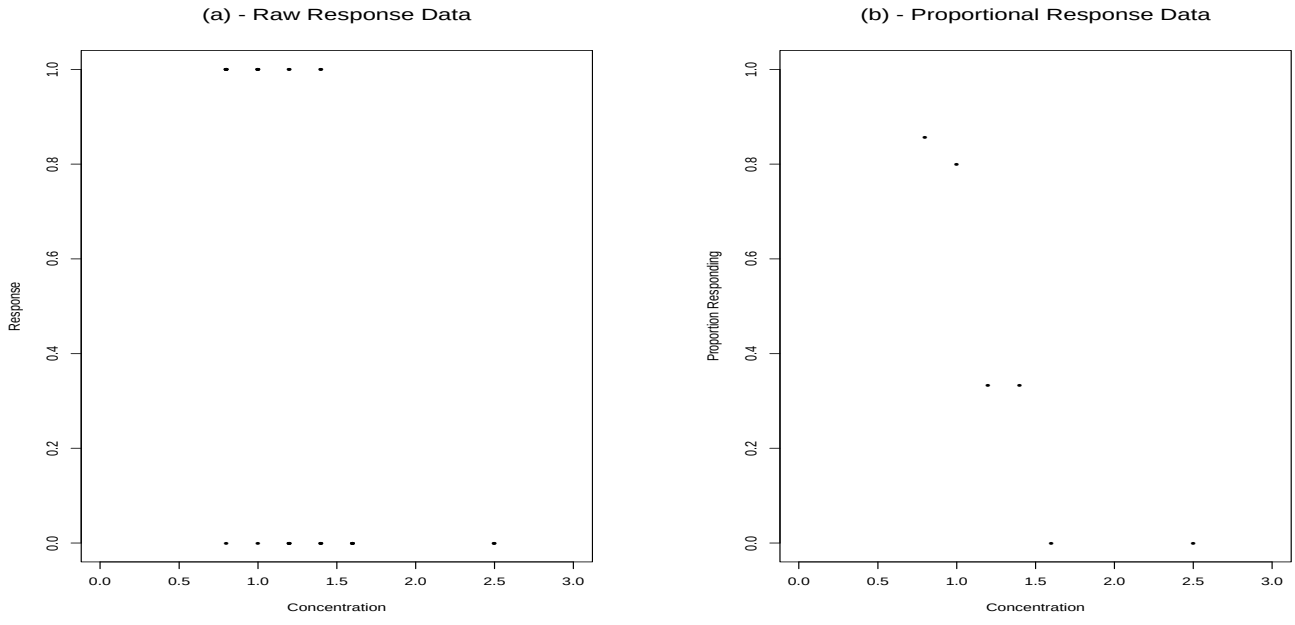
Example 1 - Anaesthetic Depth: The potency of an anaesthetic agent is measured in terms of the minimum concentration (generally measured in terms of gas pressures, and thus having units of *atmospheres*) at which at least 50% of patients exhibit no response to stimulation. Thirty patients were administered a particular anaesthetic at various predetermined concentrations for 15 minutes before a stimulus was applied. The response variable was simply an indication as to whether the patient responded to the stimulus in any way. Clearly, this is a situation where the response variable (instead of a predictor) is a factor (i.e., it has a categorical rather than numerical value). We might initially try and perform a simple linear regression in this case, but we quickly see that, as the data stands, this makes little sense. In particular, the response variable is a factor and thus is not even numerical. Of course, the standard way to handle such situations is through the use of indicator variables, and so we could define our response value as a 1 if the patient responded to the stimulus and a 0 otherwise. However, even with this modification, a simple plot of the data (see Plot *a* below) shows that a linear regression is a silly model for this dataset.

Now, it is reasonable to assume that the response values are independent and that the probability that a patient will respond is the same for all patients who received the same treatment concentration. Thus, we can summarise our data in the following table:

Conc.(x_i)	n_i	Y_i	Conc.(x_i)	n_i	Y_i
0.8	7	6/7=0.857	1.4	6	2/6=0.333
1.0	5	4/5=0.800	1.6	4	0/4=0.000
1.2	6	2/6=0.333	2.5	2	0/2=0.000

where n_i is the number of individuals who received concentration x_i, and Y_i is the proportion of these n_i who responded to the stimulus. A plot of these new data (see Plot *b* below) show that perhaps a linear model may be possible in the form

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \qquad \epsilon_i \sim \mathrm{N}(0, \sigma^2).$$



Unfortunately, this approach is still inappropriate. To see why, we note that the structure of the problem clearly indicates that n_iY_i, the number of patients who responded to the stimulus at concentration x_i, has a binomial distribution with n_i trials and $\pi(x_i)$ probability of success. Clearly then, we have $E(Y_i) = \pi(x_i)$, which would imply that $\pi(x_i) = \beta_0 + \beta_1 x_i$ if the linear model were correct. Now, it must be the case that $0 \leq \pi(x_i) \leq 1$, but $\beta_0 + \beta_1 x_i$ will eventually be outside this range for some values of the predictor, x_i. In particular, a simple linear regression for this dataset gives parameter estimates of $b_0 = 1.12$ and $b_1 = -0.52$, which means that $\hat{Y}(0.2) = 1.01953$ and $\hat{Y}(2.2) = -0.01975022$. This fact may not necessarily preclude the use of a linear model for such data when the observed proportions are far from 0 and 1, and we acknowledge our model as a simple approximation to the relationship for some restricted range of the predictor. In the current example, however, we have proportions near 0 and 1, and a linear approach seems to be unwarranted.

Before we deal directly with the non-linearity problem, we should note that even if we were in a case where we were willing to use a linear model, we would still have problems to overcome. First, the data are binomially, not normally, distributed. This problem might possibly be overcome by an appeal to the central limit theorem in some cases, but in this case our individual sample sizes (the n_i's) are rather small. We will come back to this problem shortly. Perhaps a more troublesome problem is the fact that, since Y_i is just a "within-group" sample proportion, we know that

$$Var(Y_i) = \frac{1}{n_i}\pi(x_i)\{1 - \pi(x_i)\} = \frac{1}{n_i}(\beta_0 + \beta_1 x_i)(1 - \beta_0 - \beta_1 x_i),$$

which is clearly not constant. So, we have a heteroscedasticity problem.

We know that a transformation of the response can sometimes fix problems of heteroscedasticity, however, in the case of indicator response variables, we must be extremely careful, since square roots or cube roots of variables taking only the values zero and one make no change, and logarithms or reciprocals will be undefined for response values of zero. Another useful way to handle heteroscedasticity is through the use of *weighted* linear regression. If we define

$$w_i^2 = \frac{n_i}{(\beta_0 + \beta_1 x_i)(1 - \beta_0 - \beta_1 x_i)} = \{Var(Y_i)\}^{-1},$$

then

$$E(w_iY_i) = \beta_0 w_i + \beta_1 x_i w_i,$$

which is still in the form of a simple linear model, but now $Var(w_iY_i) = 1$ for each i, so the model is homoscedastic. (Note that the definition of the w_i^2's as the reciprocal of the variances of the Y_i's can be used to apply this approach to many general settings)

In order to fit a weighted linear regression, we need to minimise the *weighted sum of squares* function:

$$d_w(\beta_0, \beta_1) = \sum_{i=1} w_i^2 (Y_i - \beta_0 - \beta_1 x_i)^2.$$

Using algebra similar to that for the usual linear regression situation, we can find the *weighted least-squares estimates* from the matrix formula

$$b_w = (X^T W X)^{-1} X^T W Y,$$

where W is an $n \times n$ diagonal matrix with i^{th} diagonal element equal to w_i^2. To perform a weighted regression in *S-Plus*, we can use the `lm` (or `lsfit`) command with the optional argument `weights` (or `wt` for `lsfit`) set equal to a vector of the w_i^2's. For the situation at hand, we have a slight

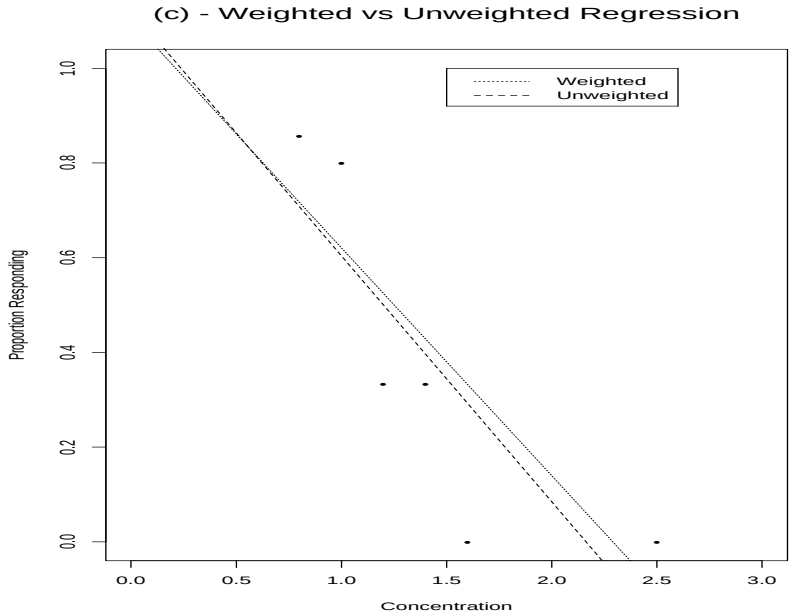
problem since the values of the weights we want to use depend on β_0 and β_1. To overcome this problem, we can use an *iterative* algorithm as follows:

1. Fit the unweighted regression Y_i on x_i to obtain the fitted values $\hat{Y}_i$.
2. Set the weights equal to $w_i^2 = n_i/\hat{Y}_i(1-\hat{Y}_i)$. Note that a weight less than or equal to zero is not allowed, so if $\hat{Y}_1 \leq 0$ or $\hat{Y}_i \geq 1$ then we must modify it to $\hat{Y}_i = 0.05$ or $\hat{Y}_i = 0.95$, respectively.
3. Fit a weighted regression of Y on x using the weights w_i^2 from Step 2 and obtain the new fitted values as $\hat{Y}_i = b_{w,0} + b_{w,1}x_i$, where $b_{w,0}$ and $b_{w,1}$ are the weighted least-squares estimates of the intercept and slope, respectively.
4. Repeat Steps 2 and 3 until your fitted values (or correspondingly the parameter estimates) are no longer changing to any substantial degree.

To perform this algorithm in *S-Plus* for the problem at hand, we would use the commands:

```
> ansth <- read.table("AnsthcSum.txt",header=T)
> attach(ansth)
> names(ansth)
[1] "conc"  "ssize" "numresp" "prptn"
> ansth.reg <- lm(prptn ~ conc)
> coef(ansth.reg)
 (Intercept)        conc
    1.123458 -0.5196401
> fitted(ansth.reg)
         1         2         3         4         5          6
 0.7077463 0.6038183 0.4998903 0.3959623 0.2920342 -0.1756419
> fts <- ifelse(fitted(ansth.reg)<=0,0.05,fitted(ansth.reg))
> wgt <- ssize/(fts*(1-fts))
> ansth.reg <- lm(prptn ~ conc,weights=wgt)
> fitted(ansth.reg)
         1         2         3         4         5         6
 0.7127002 0.6165633 0.5204265 0.4242896 0.3281528 -0.104463
   .
(repeat until fitted values stop changing)
   .
> fts <- ifelse(fitted(ansth.reg)<=0,0.05,fitted(ansth.reg))
> wgt <- ssize/(fts*(1-fts))
> ansth.reg <- lm(prptn ~ conc,weights=wgt)
> fitted(ansth.reg)
         1         2         3         4         5         6
 0.7170999 0.6207982 0.5244964 0.4281946 0.3318929 -0.101465
> fts <- ifelse(fitted(ansth.reg)<=0,0.05,fitted(ansth.reg))
> wgt <- ssize/(fts*(1-fts))
> ansth.reg <- lm(prptn ~ conc,weights=wgt)
> fitted(ansth.reg)
         1         2         3         4         5         6
 0.7170999 0.6207982 0.5244964 0.4281946 0.3318929 -0.101465
> coef(ansth.reg)
 (Intercept)        conc
    1.102307 -0.4815088
```

Notice that the final fitted values still contain a value which is less than zero, and the values for the weighted least-squares estimates still yield predicted values of $\hat{Y}(0.2) = 1.006$ and $\hat{Y}(2.3) = -0.005$. Clearly, the non-linearity problem must be dealt with (along with the non-normality). Just before we do this, however, we should look at the difference between the weighted and unweighted linear regression fits (see Plot c below):



weighted tried, but still sucks at its job.

Notice that neither of these models seems to be accurately capturing the relationship between concentration of anaesthetic and proportion responding to stimulation. However, the weighted model looks to be trying to keep the fitted values between zero and one a bit more faithfully than the unweighted model.

$\pi(x_i)$: prob of success. IDEA: a function of $\pi(x_i)$ as linear regression.

Of course, what we really want is to fit a model which more accurately describe the shape of the relationship between $\pi(x_i)$ and x_i. Namely, we want a relationship which drops smoothly down from $\pi(0) = 1$ towards $\pi(\infty) = 0$. In other words, we would like to model the relationship as:

$$\pi(x_i) = g^{-1}(\beta_0 + \beta_1 x_i) \qquad \text{or equivalently} \qquad g\{\pi(x_i)\} = \beta_0 + \beta_1 x_i,$$

for some function $g^{-1}(u)$ which satisfies $0 \leq g^{-1}(u) \leq 1$, is monotonically decreasing (or increasing). Some common choices for this function are:

$g^{-1}(u)$	$g(u)$	Name
$\Phi(u)$	$\Phi^{-1}(u)$	Probit Model
$\frac{e^u}{1+e^u}$	$\log\left(\frac{u}{1-u}\right)$	Logit (or Logistic) Model
$1 - e^{-e^u}$	$\log\{-\log(1-u)\}$	Complementary Log-Log Model

probit logit cloglog

The first two of these models are very difficult to choose between in practice. The Probit model was frequently used in the past because of its nice connections to normal distributions (recall that $\Phi(u)$ is just the CDF of the standard normal distribution). However, it is not particularly interpretable, and recently the logistic model has become the preferred one because

now we choose logit over probit.

advantages of logit

$$g\{\pi(u)\} = \log\left(\frac{\pi(u)}{1-\pi(u)}\right)$$

(1)

is the natural logarithm of an odds ratio which has a natural interpretation. In addition, some theoretical work has shown that the logistic model is more robust than the probit model.

(2)

So, suppose we decide to fit the model $g\{\pi(x_i)\} = \beta_0 + \beta_1 x_i$ to our anaesthic dataset. Since this is now a linear model, we could propose to fit it using a least-squares approach. Of course, there is still the problem of heteroscedasticity, so perhaps a weighted least-squares approach would be more applicable. Unfortunately, even a weighted approach is not really that useful here, primarily due to the fact that the (transformed) data are still not normally distributed. However, since we have pointed out that least-squares estimation is equivalent to maximum likelihood estimation when we are dealing with normal distributions and linear regressions, it seems sensible to try and use maximum likelihood in this new situation. A more detailed discussion of the method of maximum likelihood will be given in the next section; however, in simple terms, the general concept behind maximum likelihood is to examine the chosen family of probability distributions for the stochastic component of any model and then estimate the parameters by choosing those values which make the probability of the actually observed values as large as possible. When treated in this way, the probability density or probability mass function is generally referred to as the likelihood function, though in principle the two functions are actually identical. As such, maximising a likelihood does not require constant variances or normal distributions and is thus a much more flexible tool for model fitting, all we need is a specification of the likelihood function for the parameters we wish to estimate. As we noted earlier, the data in this example are clearly binomial. In particular, the data for each individual is a Bernoulli random variable, say y_i, and so the likelihood (i.e., the probability mass function) based on individual i is just

$$\pi(x_i)^{y_i}\{1-\pi(x_i)\}^{1-y_i} = g^{-1}(\beta_0+\beta_1 x_i)^{y_i}\{1-g^{-1}(\beta_0+\beta_1 x_i)\}^{1-y_i}.$$

Since each of the data points are independent, the overall (or joint) likelihood is just the product of these individual likelihoods, which can be grouped into the six concentration categories to give:

$$L(\beta_0,\beta_1) = \prod_{k=1}^{6} g^{-1}(\beta_0+\beta_1 x_k)^{n_k Y_k}\{1-g^{-1}(\beta_0+\beta_1 x_k)\}^{n_k(1-Y_k)}.$$

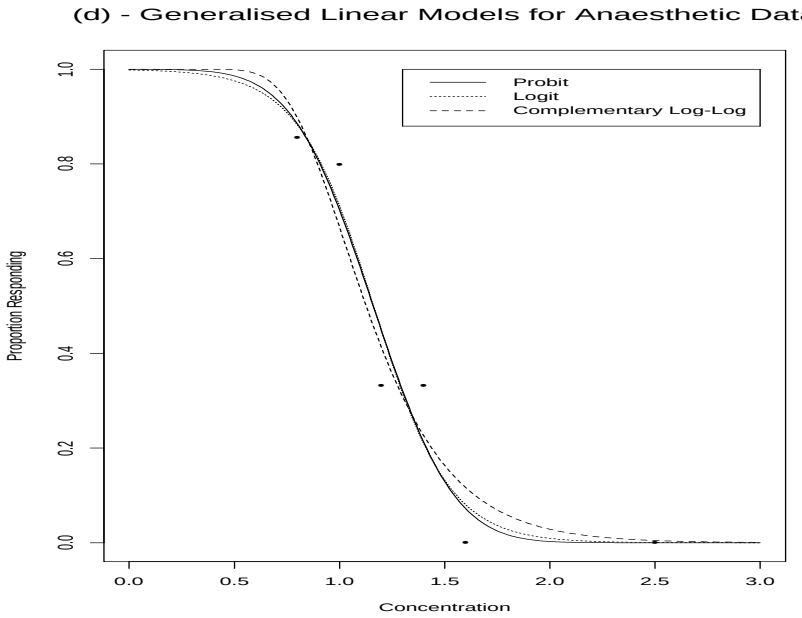
So, we can find estimates for the parameter $\beta = (\beta_0, \beta_1)$ by maximising $L(\beta_0, \beta_1)$. Of course, this looks like a horrible process to have to perform, but fortunately, *S-Plus* will do it for us (using a scheme called *iteratively re-weighted least-squares* or *IRLS* which is similar in nature to the iterative scheme discussed previously regarding weighted least-squares estimation when the weights were unknown).

```
> ansth.prbt <- glm(prptn ~ conc,family=binomial(link=probit),weights=ssize)
> fitted(ansth.prbt)
         1         2         3         4          5            6
 0.8846092 0.7031393 0.4477114 0.2129165 0.07197489 4.228517e-06
> coef(ansth.prbt)
 (Intercept)        conc
    3.857935 -3.324484
> ansth.lgt <- glm(prptn ~ conc,family=binomial(link=logit),weights=ssize)
> fitted(ansth.lgt)
         1         2         3         4          5            6
 0.8823905 0.7113424 0.4473362 0.2100222 0.08030988 0.0005821328
> coef(ansth.lgt)
 (Intercept)        conc
    6.468675 -5.566762
```

```
> ansth.cll <- glm(prptn ~ conc,family=binomial(link=cloglog),weights=ssize)
> fitted(ansth.cll)
        1         2        3         4         5           6
0.8972076 0.6668678 0.412042 0.2263281 0.1166052 0.004685568
> coef(ansth.cll)
 (Intercept)      conc
    3.731609 -3.637012
```

all fitted values in [0,1]

Note that now none of the fitted values are outside of the range zero to one. Indeed, if we look at a plot of the fitted curves (see Plot *d* below) we see how well the curves appear to fit the data. Of course, we will still need to perform diagnostic checks of our model just as we did for linear regression analysis, and we will see how best to do this shortly. In addition, recall that the real question for this example was to estimate the minimum concentration level which produced no more than 50% response (see Tutorial Week 6). So, we need to be able to estimate and find confidence intervals for functions of the parameters. Again, this is a topic which we will investigate shortly. Finally, we note that the use of the `weights` option to the `glm` function was necessary here to tell *S-Plus* how many observations each of the observed proportions was based on. We will discuss this and other details more fully in the next sections.



Summary of a basic GLM

It is now worthwhile summarising the components of the above models, as they are the basic components of all generalised linear models. We have:

1. A set of (independent) response data $Y_1, \ldots, Y_n$ with $E(Y_i) = \mu_i$;
2. A set of (vector) covariates $x_1, \ldots, x_n$;
3. A *link* function g relating the mean of the responses, μ_i, to the covariates: $g(\mu_i) = x_i^T \beta$. The quantity $\eta_i = x_i^T \beta$ is sometimes referred to as the *linear predictor*;
4. A *family* of distributions for the error vector in the model. It is this family which determines the likelihood function to be maximised for estimation of the parameter vector β. Generally, we will choose the family from a common collection called *exponential families*, which include distributions such as the normal, Poisson and binomial. We will discuss these components in more detail in the following sections.

As a final note, it is clear that if we choose the function g to be the *identity link* (i.e., $g(u) = u$) and the error distribution family to be normal than we are back in the case of the classical linear

$\begin{cases} g(u)=u \\ \varepsilon \sim N(0,\sigma^2) \end{cases}$ $\Rightarrow$ classical linear regression.

regression framework. Thus, classical regression can be seen as a special case of generalised linear modelling.

II. Exponential Families, Link Functions and Maximum Likelihood

In order to find the estimates in the example of the last section, we had to specify the *likelihood function* for the data, which is simply another name for the probability density or probability mass function of the data, regarded now as a function of the parameters instead of as a function of the data values. Once we have specified the likelihood function, we then simply estimate the parameters by those values which make the likelihood function as large as possible for our particular set of observed data. As such, these estimates are then the ones which have the "maximum likelihood" of having produced the observed data. In the example of the previous section, the choice of likelihood function was not difficult, since the data had a clear binomial distribution. As noted at the very end of the last section, we will generally restrict ourselves to choosing error distributions from the so-called *exponential families* (actually, the form that we will introduce below is technically for exponential families *with dispersion*). These are simply any distribution with a density of the following form:

$$f(y;\mu,\phi) = \exp\left\{\frac{yb(\mu) - c(\mu)}{\phi} + d(y,\phi)\right\},$$

for some specified functions $b(\mu)$, $c(\mu)$ and $d(y,\phi)$.

For example, we can write the normal density as:

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left\{\frac{-1}{2\sigma^2}(y-\mu)^2\right\} = \exp\left\{\frac{y\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \frac{y^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})\right\},$$

which has the desired form if we let $\phi = \sigma^2$, $b(\mu) = \mu$, $c(\mu) = \mu^2/2$ and $d(y,\phi) = -(y^2/2\phi) - \log(\sqrt{2\pi\phi})$. This example also demonstrates why ϕ is generally referred to as a dispersion (i.e., "spread") parameter, since it corresponds to the variance in a normal distribution.

In general, we can show that for any exponential family with dispersion, we have $E(Y) = \mu = c'(\mu)/b'(\mu)$ and $Var(Y) = \phi V(\mu)$, where

$$V(\mu) = \frac{c''(\mu) - \mu b''(\mu)}{b'(\mu)^2}.$$

(Note: We have employed the standard $'$ and $''$ notation for indicating first and second derivatives, respectively, in these formulae.)

The following table shows that some of our most commonly used distributions are of the form of exponential families with dispersion:

Distribution	$E(Y)=\mu$	$Var(Y)$	$b(\mu)$	$V(\mu)$	ϕ
Normal(μ,σ^2)	μ	σ^2	μ	1	σ^2
Binomial(n,p)	np	$np(1-p)$	$\log\left(\frac{\mu}{n-\mu}\right)$	$\frac{\mu(n-\mu)}{n}$	1
Binomial$(n,p)/n$	p	$\frac{p(1-p)}{n}$	$\log\left(\frac{\mu}{1-\mu}\right)$	$\mu(1-\mu)$	$\frac{1}{n}$
Poisson(λ)	λ	λ	$\log(\mu)$	μ	1
Poisson$(\lambda T)/T$	λ	$\frac{\lambda}{T}$	$\log(\mu)$	μ	$\frac{1}{T}$
Gamma(α,β)	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$	$-\mu^{-1}$	μ^2	$\frac{1}{\alpha}$

[Note: The density for the Gamma(α,β) distribution is given by

$$f_{\alpha,\beta}(y) = \frac{\beta^\alpha}{\Gamma(\alpha)}y^{\alpha-1}e^{-\beta y},$$

where

$$\Gamma(\alpha) = \int_0^\infty y^{\alpha-1}e^{-y}dy.]$$

We now examine the relationship between the error distribution and the link function. Recall that the link function g relates the means of the responses, $\mu_i = E(Y_i)$, to the parameter β through the linear predictor $\eta_i = x_i^T\beta$, so that

$$g(\mu_i) = x_i^T\beta = \eta_i, \qquad 1 \le i \le n.$$

Thus, the link function determines the scale on which the relationship between the mean response and the covariates is linear. Careful thought about the data generating process will often suggest suitable choices for the error distribution and the link function, but in practice it may be necessary to try various links (and error distribution families as well, though this is usually more clearly determined from the data generation process than is the link function) to obtain a model which fits the data satisfactorily.

There is, however, one particular link function associated with each exponential family which makes the likelihood function particularly simple. For a sample of independent responses $Y_1, \ldots, Y_n$ with means μ_i and common dispersion parameter ϕ (which we will generally assume to be constant, and often just set to 1), the log-likelihood function is given by:

$$l(\mu,\phi) = \log L(\mu,\phi) = \log\left[\prod_{i=1}^n \exp\left\{\frac{Y_i b(\mu_i) - c(\mu_i)}{\phi} + d(Y_i,\phi)\right\}\right] = \sum_{i=1}^n \left\{\frac{Y_i b(\mu_i) - c(\mu_i)}{\phi} + d(Y_i,\phi)\right\}.$$

[Note that the form of exponential families makes the logarithm of the likelihood a more natural function to work with, and clearly, maximising the log-likelihood is the same as maximising the original likelihood, since the logarithm is a monotonically increasing function, that is, the function $l(\mu,\phi)$ will be largest at exactly the same place as the function $L(\mu,\phi)$.]

So, inserting the relationship $\mu_i = g^{-1}(x_i^T\beta)$, we see that the log-likelihood can be written as:

$$l(\beta,\phi) = \sum_{i=1}^n \left\{\frac{Y_i b(g^{-1}(x_i^T\beta)) - c(g^{-1}(x_i^T\beta))}{\phi} + d(Y_i,\phi)\right\}.$$

Therefore, if we choose the link function g to be the same as the function b from the chosen error family, we see that the log-likelihood simplifies to

$$l(\beta,\phi) = \sum_{i=1}^n \left\{\frac{Y_i x_i^T\beta - c(g^{-1}(x_i^T\beta))}{\phi} + d(Y_i,\phi)\right\} = \sum_{i=1}^n \left\{\frac{Y_i\eta_i - c(g^{-1}(\eta_i))}{\phi} + d(Y_i,\phi)\right\}.$$

Thus, the function b has a special property when it is used as the link function and is generally referred to as the *canonical* link. The preceding table of exponential families shows the canonical links. Of course, just because the canonical link function has simple mathematical properties does not imply that it is necessarily the most appropriate link function for a given set of data. Often, however, as in the case of the binomial error family discussed in the anaesthetic example, the canonical link will have good statistical properties in addition to its nice mathematical properties (e.g., the logistic link is the canonical link for binomial data and it has been found to be more robust than the probit link function).

Once we have chosen an error family and a link function, it remains only to estimate the parameter vector β. Recall that we do so in principle by finding the value $\hat{\beta}$ which maximises the log-likelihood function. In practice, this can be a difficult task to tackle directly. Of course, we

have the aid of *S-Plus* so we don't really need to do any direct calculations, but it is useful to understand how *S-Plus* (and other statistical computer packages) solve the maximisation problem. They use a method known as $IRLS$, *Iteratively Re-weighted Least-Squares*, which we now briefly describe.

Consider the following plausible approach to estimating β. If we knew the μ_i's then we could estimate β using a linear regression of $g(\mu_i)$ on x_i, since we have assumed that $g(\mu_i) = x_i^T\beta$, that is, the $g(\mu_i)$ values are linearly related to the β's. Unfortunately, we don't know the μ_i's. Nonetheless, we do have available a very obvious unbiased estimate of the μ_i's, namely, the response values themselves, the Y_i's. So, we might consider using least-squares to regress $g(Y_i)$ on x_i. However, it can be shown (using Taylor series approximations to arrive at the so-called *Delta Method* variance estimate) that

$$Var\{g(Y_i)\} \approx Var(Y_i)g'(\mu_i)^2 = \phi V(\mu_i)g'(\mu_i)^2,$$

so the $g(Y_i)$'s do not have constant variance. Fortunately, we know how to deal with this, and we can use weighted least-squares with weights equal to, say,

$$w_i^2 = \frac{1}{V(\mu_i)g'(\mu_i)^2},$$

(recall that we generally assume ϕ is a constant, though in the case of the anaesthetic example, the exponential family table shows that $\phi = 1/n_i$ for the Binomial$(n,p)/n$ distribution, which is why we needed to add the additional weight argument to our *S-Plus* analysis using the `glm` function). Of course, we don't actually know the values of the weights w_i^2 since they depend on the μ_i's but again we can estimate the μ_i's by using the Y_i's. So, the $IRLS$ algorithm proceeds as follows:

1. Set $\hat{Y}_i = Y_i$;
2. Set the weights equal to $w_i^2 = 1/\{V(\hat{Y}_i)g'(\hat{Y}_i)^2\}$. Recall that weights must be larger than zero. If any of the weights from the above calculation are less than or equal to zero, we must make a "minor" modification to our choice for $\hat{Y}_i$ so that the weight becomes greater than zero (see the discussion of the weighted regression for the anaesthetic data example);
3. Perform a weighted regression of $g(Y_i)$ on x_i using the weights from the previous step to obtain estimates $\hat{\beta}$;
4. Calculate new fitted values $\hat{Y}_i = g^{-1}(x_i^T\hat{\beta})$ and repeat steps 2 and 3 until successive values of the parameter estimates (or equivalently the values of the weights) do not change substantially.

This procedure will quite generally arrive at the maximum likelihood estimate $\hat{\beta}$ (provided, that we have used an exponential family with constant dispersion for our error structure, however, even for other error structures, it can be shown that this $IRLS$ estimate will converge to the maximum likelihood estimate as the sample size increases). Notice that we have used least-squares methodology to arrive at a maximum likelihood estimate, which shows the close connection between the two approaches despite their apparent differences.

As a final note, we point out that if we want to do a weighted generalised linear model (as we did for the logistic regression in Example 1), then we must redefine the w_i^2's in our algorithm appropriately. If we are using additional weights ω_i^2 for each data point (e.g., for the logistic regression example in Example 1, $\omega_i^2 = n_i = 1/\phi_i$), then the proper definition for the w_i^2's is

$$w_i^2 = \frac{\omega_i^2}{V(\mu_i)\{g'(\mu_i)\}^2},$$

which we estimate as usual by substituting $\hat{Y}_i$ for μ_i.

Example 2 - Survival Time of Leukemia Patients: The white blood cell counts at the time of diagnosis for seventeen patients with leukemia are recorded below, along with their survival times from initial diagnosis:

Survival (weeks)	White Blood Cell Count	Survival (weeks)	White Blood Cell Count	Survival (weeks)	White Blood Cell Count
65	2290	156	760	100	4265
134	2570	16	6025	108	10470
121	10000	4	16980	39	5370
143	7080	56	9330	26	32360
22	34675	1	100000	1	100000
5	52480	65	100000		

We want to model survival time, S, as a function of initial white blood cell count, W. An initial plot of the data (see Plot a below) shows that a standard linear model of the form

$$E(S) = \beta_0 + \beta_1 W$$

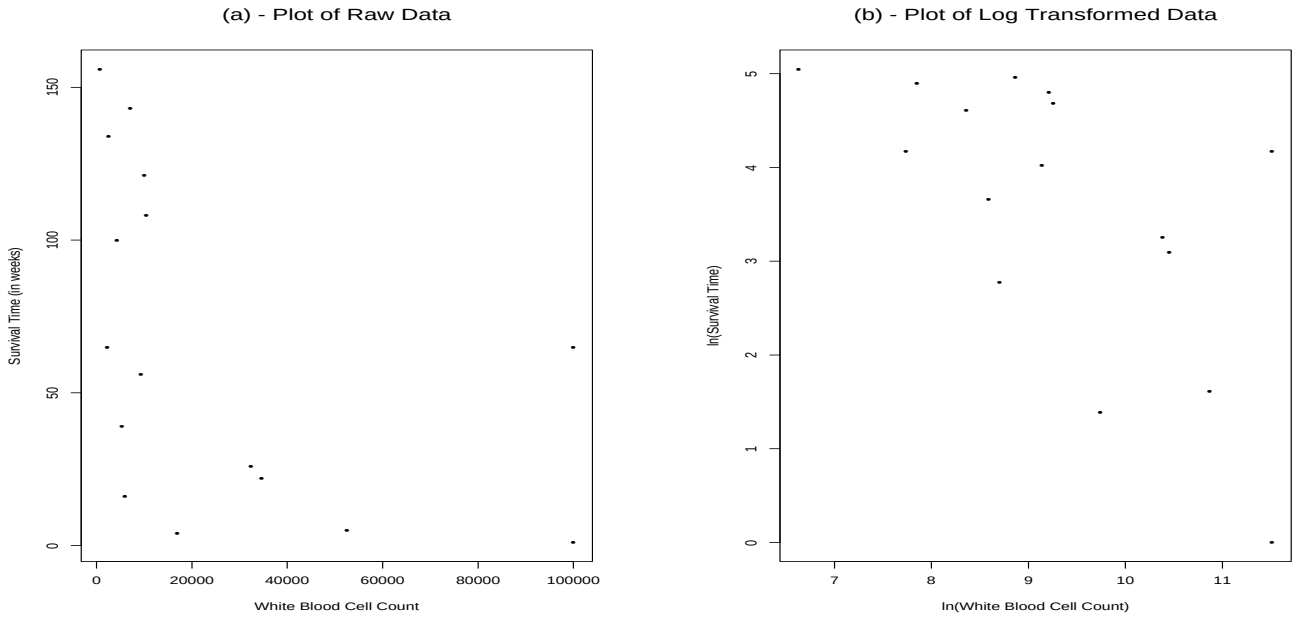
will not adequately describe the data. In particular, a linear model will lead to negative predicted survival times for individual with very high white blood cell counts. One way to combat the problem of negative survival times is to use a so-called *power model* which has the form

$$E(S) = \gamma_0 W^{\gamma_1},$$

which will smoothly decrease towards zero as W increases (provided the estimate of γ_1 is negative and the estimate of γ_0 is positive, of course), but never fall below. This model can be re-written in an equivalent linear form by taking logarithmic transformations:

$$\log E(S) = \beta_0 + \beta_1 \log W \qquad \text{where } \beta_0 = \log \gamma_0;\ \beta_1 = \gamma_1.$$

Looking at a plot of the transformed variables (see Plot b below) shows that a linear relationship may now be reasonable, however, a homoscedastic normal error structure is clearly not justifiable.



Moreover, we know that survival times are often modelled using the exponential distribution, which is a member of the gamma family (specifically, the exponential distributions are the members of

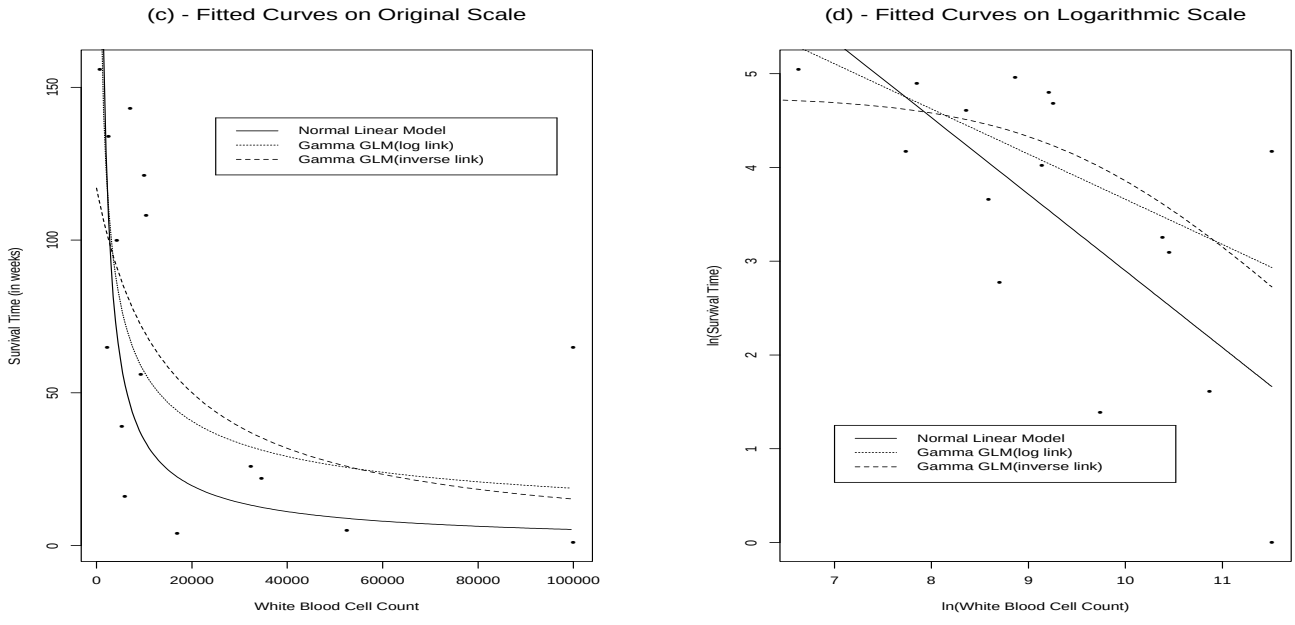
the gamma family with $\alpha = 1$). So, we might try a generalised linear model of survival time on the logarithm of the white blood cell count with a logarithmic link function and a gamma error structure for this data. Alternatively, we know that the inverse function, $g(\mu) = 1/\mu$, is the canonical link function for the gamma family, so we might also try a gamma generalised linear model of the form

$$\frac{1}{E(S)} = \beta_0 + \beta_1 W,$$

which will also ensure that the fitted survival times never become negative (provided the estimates of the β's are positive). To fit each of these models, we use the *S-Plus* commands:

```
> leuk <- read.table("LeukPos.txt",header=T)
> attach(leuk)
> names(leuk)
[1] "surv" "wbc"
> leuk.slr <- lm(log(surv) ~ log(wbc))
> coef(leuk.slr)
 (Intercept)   log(wbc)
    11.07501 -0.8178552
> leuk.glm <- glm(surv ~ log(wbc),family=Gamma(link=log))
> coef(leuk.glm)
 (Intercept)   log(wbc)
    8.477938 -0.4818071
> leuk.glm1 <- glm(surv ~ wbc,family=Gamma)
> coef(leuk.glm1)
 (Intercept)         wbc
 0.00854635 5.71935e-007
```

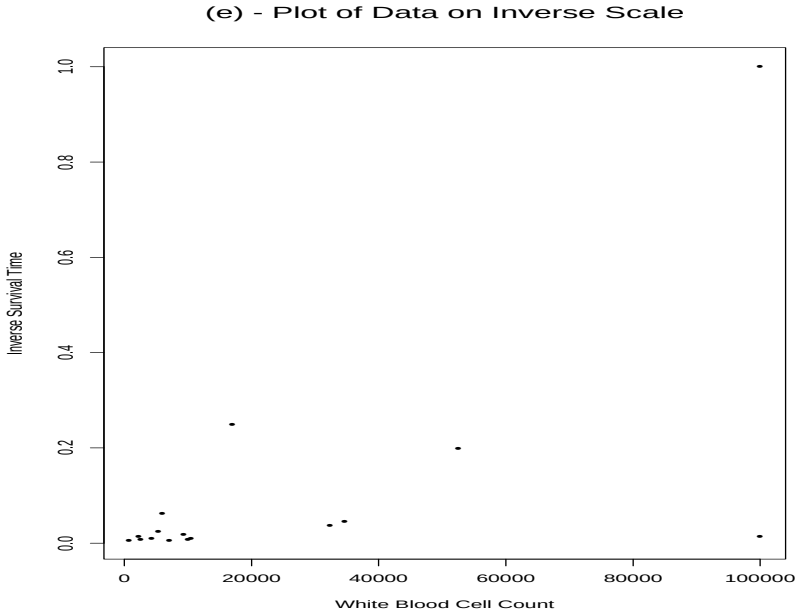
Plots of the fitted relationships for both the normal regression and the gamma generalised linear models are shown in Plots *c* and *d* on both the raw and logarithmic scale. Note that if we do not specify a `link=` option for the `family` in the `glm` command, *S-Plus* will automatically choose the appropriate the canonical link.



Note the large difference in the fitted curves. We anticipate that the diagnostics for the normal linear model will indicate heteroscedasticity. As yet, we do not have any diagnostic tools to indicate

whether the gamma generalised linear models are fitting the data adequately, but we will develop such tools shortly. However, the simple plots of the fitted curves, particularly on the logarithmic scale give us some indication of how well the models are fitting the observed data. The model employing the inverse link function appears to be capturing the possible mild curvature of the data on the logarithmic scale, however, a plot of the data on the inverse scale (see Plot e below) does not overly indicate that this is the scale on which the data are the most linear. As noted previously, the canonical link is not always the most appropriate one, despite its simple mathematical properties.

STILL PROBLEMATIC



Finally, recall that we originally thought that the appropriate error structure was exponential, but we used a full gamma family for convenience. Now, if the distribution really was exponential then we would expect that an estimate of the dispersion parameter (which is $1/\alpha$ for the gamma family) should be near 1. We can use *S-Plus* to find the (approximate) maximum likelihood estimate of the dispersion parameter:

```
> summary(leuk.glm)$dispersion
    Gamma
0.9387238
> summary(leuk.glm1)$dispersion
   Gamma
1.14073
```

log - link

Inverse - link

So, both generalised linear model structures do seem to agree that the error structure is indeed exponential (assuming, of course, that the initial choice of the gamma family is itself the correct one for the data). We will discuss estimation of the dispersion parameter further in the next section.

III. Precision of Parameter Estimates and Confidence Intervals

Now that we have estimated the parameters of our model, we would like to have some idea of how precise these estimates are. In other words, we need to calculate standard errors for our maximum likelihood estimates so that we can construct confidence intervals and perform hypothesis tests. To do this requires some small amount of mathematics to examine the behavior of the likelihood function and the maximum likelihood estimates.

First, note that maximum likelihood estimates are solutions to the system of equations:

$$\frac{\partial}{\partial \beta} l(\beta, \phi) = 0.$$

In other words, the maximum likelihood estimates, $\hat{\beta}$, satisfy $U(\hat{\beta}) = 0$, where $U(\beta) = \partial l(\beta, \phi)/\partial\beta$ is called the *score statistic* or *score function* and is just the gradient vector of the log-likelihood with respect to β. In fact, for exponential families with dispersion, (recalling the chain rule for differentiation) we have:

$$\begin{aligned} U(\beta) = \frac{\partial}{\partial\beta} l(\beta, \phi) &= \frac{1}{\phi} \sum_{i=1}^{n} \{Y_i b'(\mu_i) - c'(\mu_i)\} \frac{\partial \mu_i}{\partial \beta} \\ &= \frac{1}{\phi} \sum_{i=1}^{n} \frac{\partial \mu_i}{\partial \beta} b'(\mu_i)(Y_i - \mu_i), \end{aligned}$$

where we have used the fact that $\mu_i = c'(\mu_i)/b'(\mu_i)$. [Note that differentiation with respect to β yields a gradient *vector*. Also, note that the form of $U(\beta)$ implies that the solution to the equation $U(\hat{\beta}) = 0$ is the same regardless of the value of ϕ. In other words, whether we assume ϕ is known or try to estimate it, we still arrive at the same estimates of β, though our assessment of their precision will depend on the value we give to ϕ, as we shall see shortly.] Indeed, this fact further implies that:

$$\begin{aligned} \frac{\partial \mu_i}{\partial \beta} &= \frac{\partial}{\partial \beta} \left\{ \frac{c'(\mu_i)}{b'(\mu_i)} \right\} \\ &= \left[\frac{c''(\mu_i) b'(\mu_i) - c'(\mu_i) b''(\mu_i)}{\{b'(\mu_i)\}^2} \right] \frac{\partial \mu_i}{\partial \beta} \\ &= b'(\mu_i) \left[\frac{c''(\mu_i) - \mu_i b''(\mu_i)}{\{b'(\mu_i)\}^2} \right] \frac{\partial \mu_i}{\partial \beta} \\ &= b'(\mu_i) V(\mu_i) \frac{\partial \mu_i}{\partial \beta}, \end{aligned}$$

which implies that $b'(\mu_i) = V(\mu_i)^{-1}$. Furthermore, differentiating the relationship $g(\mu_i) = x_i^T \beta$ yields:

$$g'(\mu_i) \frac{\partial \mu_i}{\partial \beta} = x_i \qquad \Longrightarrow \qquad \frac{\partial \mu_i}{\partial \beta} = x_i g'(\mu_i)^{-1}.$$

So, we can now write U as:

$$\begin{aligned} U(\beta) &= \frac{1}{\phi} \sum_{i=1}^{n} x_i g'(\mu_i)^{-1} V(\mu_i)^{-1} (Y_i - \mu_i) \\ &= \frac{1}{\phi} \sum_{i=1}^{n} x_i g'(\mu_i) w_i^2 (Y_i - \mu_i) \\ &= \frac{1}{\phi} X^T W_g (Y - \mu), \end{aligned}$$

where the last line of the above equality is in vector notation, so that X is the design matrix (i.e., a matrix with one row for each observation and columns corresponding to the values of the predictors, including an initial column of ones for the intercept) and W_g is a diagonal matrix with i^{th} diagonal element equal to $g'(\mu_i) w_i^2$, recalling that we defined $w_i^2 = 1/\{g'(\mu_i)^2 V(\mu_i)\}$ in our discussion of the $IRLS$ algorithm.

Now, from the above formulation of $U(\beta) = \phi^{-1} X^T W_g (Y - \mu)$, we can see that the score statistic is just a vector whose components are weighted sums of the independent random variables Y_i. A generalised version of the Central Limit Theorem can be developed which shows that as the sample size n increases, such a quantity will have an approximate multivariate normal distribution with appropriate mean and variance. For the case at hand, we have:

$$E\{U(\beta)\} = \phi^{-1} X^T W_g E(Y - \mu) = 0,$$

and

$$Var\{U(\beta)\} = \phi^{-1}X^T W_g Var(Y)\{\phi^{-1}X^T W_g\}^T = \phi^{-2}X^T W_g Var(Y) W_g X = \phi^{-1}X^T W X,$$

where we have used the fact that $W_g^T = W_g$ since any diagonal matrix is clearly symmetric, and we have defined W to be the diagonal matrix with i^{th} diagonal element equal to w_i^2, so that $W_g Var(Y) W_g = \phi W$, since this quantity is just a diagonal matrix (since it is the product of diagonal matrices, recalling that the independence of the Y_i's implies that $Var(Y)$ is diagonal) with i^{th} diagonal element equal to

$$\{g'(\mu_i)w_i^2\}\{\phi V(\mu_i)\}\{g'(\mu_i)w_i^2\} = \phi g'(\mu_i) w_i^2 V(\mu_i) g'(\mu_i)\{g'(\mu_i)^{-2} V(\mu_i)^{-1}\} = \phi w_i^2.$$

It may at first seem that this result is not overly important or useful, however, it turns out to be the most direct way to attack the problem of the distribution of the maximum likelihood estimator, $\hat{\beta}$. Specifically, note that we can approximate the value of $h(z)$ for any function h by:

$$h(z) \approx h(z_0) + h'(z_0)(z - z_0),$$

where z_0 is any point which is "near enough" to z. If h is a vector-valued function of a vector argument z, this formula becomes:

$$h(z) \approx h(z_0) + \frac{\partial h(z_0)}{\partial z^T}(z - z_0).$$

(Note: $\partial h(z_0)/\partial z^T$ is just a shorthand notation for $\left\{\partial h(z_0)/\partial z\right\}^T$, recalling that if h is a vector-valued function and z is a vector argument to the function then $\partial h(z_0)/\partial z$ is a matrix.) Thus, since we know that $\hat{\beta}$ should be "close" to the true value of β, and $U(\hat{\beta}) = 0$, we can write:

$$0 = U(\hat{\beta}) \approx U(\beta) + \frac{\partial U(\beta)}{\partial \beta^T}(\hat{\beta} - \beta) \approx U(\beta) + E\left\{\frac{\partial U(\beta)}{\partial \beta^T}\right\}(\hat{\beta} - \beta),$$

where the last approximation follows from the fact that

$$\begin{aligned}\frac{\partial U(\beta)}{\partial \beta^T} &= \frac{1}{\phi}\sum_{i=1}^n x_i \frac{\partial\{g'(\mu_i)w_i^2\}}{\partial \beta^T}(Y_i - \mu_i) - \frac{1}{\phi}\sum_{i=1}^n x_i g'(\mu_i) w_i^2 \frac{\partial \mu_i}{\partial \beta^T} \\ &= \frac{1}{\phi}\sum_{i=1}^n x_i \frac{\partial\{g'(\mu_i)w_i^2\}}{\partial \beta^T}(Y_i - \mu_i) - \frac{1}{\phi}\sum_{i=1}^n x_i w_i^2 x_i^T,\end{aligned}$$

so that $\partial U(\beta)/\partial \beta^T$ has the form of a weighted sum of independent random quantities, and a generalised version of the Law of Large Numbers shows that such quantities will be "close" to their expectation as the number of terms in the sum increases. Taking expectations in the above equation shows that $E\{\partial U(\beta)/\partial \beta^T\} = -(1/\phi)X^T W X$. Therefore, we can write

$$\hat{\beta} - \beta \approx -E\{\partial U(\beta)/\partial \beta^T\}^{-1} U(\beta) = \phi(X^T W X)^{-1} U(\beta).$$

In other words, $\hat{\beta}$ can be approximated by a linear transformation of the score statistic, and thus it must have an approximate multivariate normal distribution (since linear transformations of normal random quantities are still normally distributed) with mean $E(\hat{\beta}) \approx \beta - \phi(X^T W X)^{-1} E\{U(\beta)\} = \beta$ and variance

$$\begin{aligned}Var(\hat{\beta}) &\approx \phi(X^T W X)^{-1} Var\{U(\beta)\}\left\{\phi(X^T W X)^{-1}\right\}^T \\ &= \phi(X^T W X)^{-1}\{\phi^{-1}(X^T W X)\}\{\phi(X^T W X)^{-1}\} \\ &= \phi(X^T W X)^{-1}.\end{aligned}$$

Moreover, any linear combination of the maximum likelihood estimators, say $c^T\hat{\beta}$, will have an approximate normal distribution with mean $E(c^T\hat{\beta}) \approx c^T\beta$ and variance

$$Var(c^T\hat{\beta}) \approx \phi c^T(X^TWX)^{-1}c.$$

(Note the similarity to the corresponding result for ordinary multiple linear regression.)

Of course, once we are dealing with non-linear modelling curves, we will often be interested in finding confidence intervals for non-linear combinations of the parameters. For example, in the anaesthetic example, we were interested in estimating the concentration at which 50% of individuals did not respond to the stimulus, which turns out to be a non-linear function of the maximum likelihood estimates (see Tutorial Week 6).

So, in order to find an estimate and confidence interval for some function of the parameters, say $h(\beta)$, we can use the estimate $h(\hat{\beta})$ but we need some measure of the standard error of this estimator. To find an estimate of the standard error, we again note that

$$h(\hat{\beta}) \approx h(\beta) + \frac{\partial h(\beta)}{\partial \beta^T}(\hat{\beta} - \beta).$$

Thus, the distribution of $h(\hat{\beta})$ can again be approximated by a normal distribution with mean $h(\beta)$ and variance

$$Var\{h(\hat{\beta})\} \approx \frac{\partial h(\beta)}{\partial \beta^T} Var(\hat{\beta}) \left\{ \frac{\partial h(\beta)}{\partial \beta^T} \right\}^T = \phi \frac{\partial h(\beta)}{\partial \beta^T}(X^TWX)^{-1}\frac{\partial h(\beta)}{\partial \beta}.$$

At this point, it is important to note that while this approach to confidence intervals is extremely general, it is susceptible to poor performance in some circumstances. This is primarily due to the fact that the linear approximation to $h(\hat{\beta})$ given above can be rather poor for some functions h. Indeed, for some special cases of interest it is well known that better confidence interval methods exist. The most important such case is that of finding a confidence interval for a mean response value. Suppose that we want to find a confidence interval for the mean response value for some specified values of the predictors, say x_0. In other words, we want to find a confidence interval for $\mu_0 = g^{-1}(x_0^T\beta)$. Using the preceding scheme, we would estimate the mean response by $g^{-1}(x_0^T\hat{\beta})$ and then form a confidence interval based on a standard error derived from the approximation:

$$Var\{g^{-1}(x_0^T\hat{\beta})\} \approx \phi \frac{\partial}{\partial \beta^T}\{g^{-1}(x_0^T\beta)\}(X^TWX)^{-1}\frac{\partial}{\partial \beta}\{g^{-1}(x_0^T\beta)\} = \frac{\phi x_0^T(X^TWX)^{-1}x_0}{\{g'(\mu_0)\}^2},$$

where we have used the fact that

$$\begin{aligned} z = g\{g^{-1}(z)\} \qquad &\Longrightarrow \qquad 1 = \frac{d}{dz}[g\{g^{-1}(z)\}] = g'\{g^{-1}(z)\}\frac{d}{dz}\{g^{-1}(z)\} \\ &\Longrightarrow \qquad \frac{d}{dz}\{g^{-1}(z)\} = \frac{1}{g'\{g^{-1}(z)\}}. \end{aligned}$$

It turns out that this approach is not as good as an alternative approach based on computing a confidence interval for the value of $x_0^T\beta$ and then transforming the endpoints using the function g^{-1}. In other words, we can find a confidence interval for $x_0^T\beta$ using the fact that

$$Var(x_0^T\hat{\beta}) \approx \phi x_0^T(X^TWX)^{-1}x_0$$

and derive an interval of the form (l, u). We could then compute a confidence interval for $g^{-1}(x_0^T\beta)$ as $\{g^{-1}(l), g^{-1}(u)\}$.

The best method of constructing confidence intervals is still a subject of much debate and research in statistics. One suggestion is to calculate several confidence intervals and to present all of them for comparison. Unfortunately, the various confidence interval procedures can give quite different interval estimates, making interpretation difficult. Another angle of attack is based on the fact that symmetric confidence intervals are generally only appropriate on certain scales of measurement, and we should construct symmetric confidence intervals on this scale (when possible) and then transform these intervals to the desired scale for estimation. To elucidate this point, recall the example of a confidence interval for a mean response discussed above. If we were to construct a confidence interval for $g^{-1}(x_0^T\beta)$ directly, we might wind up with an interval which contains values outside the allowable range for the quantity in question.

Example 1 (Continued): Suppose that we wanted to estimate the response probability at an anaesthetic concentration of 1.5 atmospheres. Our estimate (using the logistic model) would be

$$\hat{\pi}(1.5) = g^{-1}(\hat{\beta}_0+1.5\hat{\beta}_1) = \frac{\exp(\hat{\beta}_0 + 1.5\hat{\beta}_1)}{1+\exp(\hat{\beta}_0 + 1.5\hat{\beta}_1)} = \frac{\exp\{6.468675 + 1.5(-5.566762)\}}{1+\exp\{6.468675 + 1.5(-5.566762)\}} = 0.13222.$$

Now, simple differentiation shows that $g'(z) = 1/\{z(1-z)\}$, which means that $g'[g^{-1}\{6.468675 + 1.5(-5.566762)\}] = 8.71552$. Therefore, the variance of our estimate in this case is

$$\frac{\phi x_0^T(X^TWX)^{-1}x_0}{8.71552^2}.$$

Recall that for the binomial model the dispersion parameter is $\phi = 1$, and we need only calculate the value of $x_0^T(X^TWX)^{-1}x_0$. (Note that technically speaking, the dispersion parameter for this example was $\phi_i = 1/n_i$, however, these were taken into account via the additional weights which will appear in the W matrix; see the discussion immediately following the description of the *IRLS* algorithm. Also note that it is possible to have *S-Plus* estimate ϕ instead of just assuming that it is one, and we will discuss this in more detail in the next section on diagnostics). In actuality, this value depends on the true values of the μ_i's which we do not know. Fortunately, *S-Plus* provides an estimate of the matrix $(X^TWX)^{-1}$ directly as the `cov.unscaled` element of the `summary()` list. So, we can estimate $x_0^T(X^TWX)^{-1}x_0$ as:

```
> x0 <- c(1,1.5)
> x0xwxx0 <- t(x0)%*%summary(ansth.lgt)$cov.unscaled%*%x0
> x0xwxx0
             [,1]
[1,] 0.7011042
```

So, the estimated variance of $\hat{\pi}(1.5)$ is $0.7011042/8.71552^2 = 0.009230$, and we can use this value to construct an appropriate confidence interval. The question now arises as to what multiplier to use. That is, should we use a normal quantile or that of a Student's-t distribution. Since we only have an estimate of the variance at hand (recall that the true variance depends on the true values of the μ_i's which we had to estimate), we clearly need to employ a Student's-t quantile, and the degrees of freedom we use is just the same as it would be in the case of a normal linear regression, namely, $n-p$, where p is the number of parameters in the model and n is the number of data points used in the model, which in this example is 6 (and not 30 as we might first think). So, a 95% confidence interval could then be calculated as:

$$0.13222 \pm t_{6-2}(0.975)\sqrt{0.009230} = 0.13222 \pm 2.776445(0.09607) = (-0.13451, 0.39879),$$

which is clearly a very poor interval estimate.

Alternatively, if we form a confidence interval for the quantity $x_0^T\beta$ by adding and subtracting some margin of error from $x_0^T\hat{\beta} = 6.468675 + 1.5(-5.566762) = -1.881465$, and then calculating g^{-1} of the resulting endpoint values, we are guaranteed that this interval will have endpoints within the desired range between zero and one. Specifically, we know that

$$Var(x_0^T\hat{\beta}) \approx \phi x_0^T(X^TWX)^{-1}x_0,$$

which we have already calculated above as 0.7011042. Thus, a 95% confidence interval for $x_0^T\beta$ is given by:

$$-1.881465 \pm t_{6-2}(0.975)\sqrt{0.7011042} = -1.881465 \pm 2.776445(0.83732) = (-4.206, 0.4433).$$

This means that we can calculate a 95% confidence interval for $g^{-1}(x_0^T\beta) = \pi(1.5)$ as:

$$\{g^{-1}(-4.206), g^{-1}(0.4433)\} = (0.01469, 0.6090).$$

This interval is still extremely wide (indicating that we cannot estimate $\pi(1.5)$ very accurately from the data at hand), but at least its endpoints are between zero and one. Note also that our point estimate $\hat{\pi}(1.5) = 0.13222$ is no longer in the middle of our confidence interval. Of course, a binomial distribution is highly skewed in cases where its expectation is small, so that it really makes little sense to force confidence intervals based on binomial data to be symmetric.

Of course, it is not always quite so apparent what quantity is the best one to use for the formation of the initial symmetric interval. In practice, we must rely on experience and the details of the specific problem to tell us what to do. By way of final note, we point out that the above problem is not always so serious. Indeed, if we had desired a 95% confidence interval for $\pi(1.2)$ instead of $\pi(1.5)$ the intervals would have been (0.1199,0.7742) and (0.1772,0.7525), respectively. The second interval based on transforming the interval for $x_0^T\beta$ is still shorter and probably more appropriate, but the first interval is at least reasonable now.

Finally, we note that the above example was somewhat simplified by the fact that ϕ is assumed to be 1 for a binomial error structure. If the error structure had been some other exponential family we would have had to estimate its value. In general, we could estimate ϕ using maximum likelihood methods as we did for β. Of course, we have already seen that the *S-Plus* `summary` command produces a list containing the `dispersion` element which is an estimate of ϕ. We will simply rely on *S-Plus* for our dispersion estimates (though we will talk briefly about diagnostics based on estimating the dispersion in a later section), and we give here the standard estimates of dispersion for the most common exponential families:

$$\hat{\phi} = \begin{cases} MSE = \frac{1}{n-p}\sum_{i=1}^n (Y_i - \hat{Y}_i)^2 & \text{Normal}(\mu, \sigma^2) \\ 1 & \text{Binomial}(n,\pi) \text{ and Poisson}(\lambda) \\ CV = \frac{1}{n-p}\sum_{i=1}^n \left(\frac{Y_i - \hat{Y}_i}{\hat{Y}_i}\right)^2 & \text{Gamma}(\alpha, \beta) \end{cases}$$

where *CV* is called the estimated *coefficient of variation* for the gamma distribution.

IV. Model Simplification and Analysis of Deviance

Individual confidence intervals for the parameters are important for determining the effect of individual predictors, however, we also need methods for determining whether groups of predictors are necessary for the model or not, just as we did for linear regression analyses. For classical linear modelling, we accomplished this through the sums-of-squares decompositions. However, since we are

now employing maximum likelihood estimation instead of a least-squares approach, sums-of-squares are no longer the appropriate method for determining the contribution of groups of predictors to the explanatory power of our generalised linear models. The appropriate statistic for generalised linear models is the *deviance* or *residual deviance*, $D(\hat{Y}, Y)$, defined by

$$D(\hat{Y}, Y) = 2\phi\{l(Y, \phi) - l(\hat{Y}, \phi)\},$$

which measures the (scaled) difference between the log-likelihood for the observed data and the log-likelihood of the fitted values (or equivalently the ratio of the corresponding likelihoods themselves), and thus small values of the deviance indicate that a model fits the observed data well. Now, if we are fitting a model with link function g and an error distribution with log-likelihood function:

$$l(\mu, \phi) = \sum_{i=1}^{n} \left\{ \frac{Y_i b(\mu_i) - c(\mu_i)}{\phi} + d(Y_i, \phi) \right\},$$

then the deviance is calculated as:

$$\begin{aligned} D(\hat{Y}, Y) &= 2\phi \left[\sum_{i=1}^{n} \left\{ \frac{Y_i b(Y_i) - c(Y_i)}{\phi} + d(Y_i, \phi) \right\} - \sum_{i=1}^{n} \left\{ \frac{Y_i b(\hat{Y}_i) - c(\hat{Y}_i)}{\phi} + d(Y_i, \phi) \right\} \right] \\ &= 2 \sum_{i=1}^{n} \left[Y_i \{ b(Y_i) - b(\hat{Y}_i) \} - \{ c(Y_i) - c(\hat{Y}_i) \} \right], \end{aligned}$$

where $\hat{Y}_i = g^{-1}(x_i^T \hat{\beta})$ are the fitted values and $\hat{\beta}$ is the maximum likelihood estimator of the parameter vector.

So, for example, if we have the identity link and normal error structure, so that

$$l(\mu, \phi) = -\frac{1}{2\phi} \sum_{i=1}^{n} (Y_i - \mu_i)^2 - \frac{n}{2} \log(2\pi\phi),$$

the deviance then becomes:

$$D(\hat{Y}, Y) = -\frac{\phi}{\phi} \sum_{i=1}^{n} (Y_i - Y_i)^2 - n\phi \log(2\pi\phi) + \frac{\phi}{\phi} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + n\phi \log(2\pi\phi) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = SSE.$$

Like the sums of squares in the least-squares approach, the deviances measure the deviation of fitted values from the observed data. Similarly, the difference in deviances between two "nested" models is a measure of how much better the "larger" model is at fitting the data. We note that "nested" models are simply two models of the same form (i.e., the same link function and error structure), the "smaller" of which contains only a subset of the predictor variables which are contained in the "larger" model. This is exactly the structure we have when we are testing whether a subset of the predictors are significant in a multiple linear regression modelling situation. Just as the difference in sums of squares is a measure of how important the subset of predictors being tested is to the least-squares model, the difference in deviance is a measure of how important the subset of predictors which have been left out of the smaller model is in explaining the variation in the observed data. Of course, as we did for the difference in sums-of-squares, we must properly scale the difference in deviances. Thus, we define the scaled decrease in deviance (sometimes called the *deviance statistic*) for two nested models as:

$$\frac{D(\hat{Y}_S, Y) - D(\hat{Y}_L, Y)}{\hat{\phi}_L} = \frac{D(\hat{Y}_S, \hat{Y}_L)}{\hat{\phi}_L} = D^*(\hat{Y}_S, \hat{Y}_L),$$

which measures the extent to which the larger model is a better fit to the data. Here, $\hat{Y}_L = g^{-1}(x_{i,L}^T\hat{\beta}_L)$ and $\hat{\phi}_L$ indicate the fitted values and dispersion estimate from the larger model (although we may want to use $\hat{\phi}_{full}$ for $\hat{\phi}_L$ in cases where both models under investigation are subsets of some larger overall model, just as we did for standard linear regression), while $\hat{Y}_S = g^{-1}(x_{i,S}^T\hat{\beta}_S)$ is the vector of fitted values for the smaller model, and the statement that the models are nested simply indicates that the predictor vector $x_{i,L}$ contains all of the values in $x_{i,S}$ plus some additional values, so that $x_{i,L} = (x_{i,S}, x_{i,\bar{S}})$ and $\beta_L = (\beta_S, \beta_{\bar{S}})$. Clearly, large values of the deviance statistic indicate that the smaller model is not fitting the data as well as the larger model and provides evidence against the hypothesis that $\beta_{\bar{S}} = 0$. The question, of course, is how to decide how large a value indicates a "significant" difference in deviance.

Using arguments similar to those employed to determine the distribution of the maximum likelihood estimator, it can be shown that, under the null hypothesis that the smaller model is correct, the drop in deviance statistic has an approximate chi-squared distribution with degrees of freedom equal to the difference in the number of parameters between the two nested models being compared. So, to test the hypotheses:

$$H_0 : \beta_{\bar{S}} = 0 \qquad \text{versus} \qquad H_A : \beta_{\bar{S}} \neq 0,$$

at significance level α, we would reject H_0 if

$$D^*(\hat{Y}_S, \hat{Y}_L) \geq \chi^2_{(p-q)}(1-\alpha),$$

where we assume that β_L has dimension p and β_S has dimension $q < p$, so that and $\beta_{\bar{S}}$ has dimension $p - q$, and $\chi^2_{(p-q)}(1-\alpha)$ is the $(1-\alpha)$-quantile of a chi-squared distribution with $p - q$ degrees of freedom.

Again for the case of an identity link and normal error structure, we see that for the constant model $Y_i = \beta_0 + \epsilon_i$, the fitted values are clearly just $\hat{Y}_i = \overline{Y}$ for all data points, implying that the deviance for this model is

$$D(\overline{Y}, Y) = \sum_{i=1}^{n}(Y_i - \overline{Y})^2 = SST.$$

Thus, the drop in deviance test statistic for the overall significance of a multiple regression with $p-1$ predictors (i.e., testing the null hypothesis $H_0 : \beta_1 = \ldots = \beta_{p-1} = 0$) would be:

$$D^*(\overline{Y}, \hat{Y}) = \frac{D(\overline{Y}, Y) - D(\hat{Y}, Y)}{\hat{\phi}_L} = \frac{SST - SSE}{MSE} = \frac{SSR}{MSE} = \frac{(p-1)MSR}{MSE} = (p-1)F,$$

where F is the usual test statistic in this situation, and we have used the standard dispersion estimate for the normal situation, $\hat{\phi}_L = MSE$. So, we see that the deviance statistic is equivalent to the standard F-statistic (and is exactly equal to it in the case where $p = 2$, i.e., simple linear regression). However, the standard F-test is more appropriate here since it is based on the exact distribution of the F-statistic (assuming that the normality assumptions are justified), whereas the test based on the chi-squared distribution is only an approximation (though as the sample size increases, the two approaches become identical). In other words, the flexibility of the likelihood and deviance approach to modelling comes at some cost, though it is not an expensive one if the sample size, n, is large compared to the number of parameters in the model.

As with sums of squares, it is useful to display drops in deviance in tabular form. So, we construct the so-called *Analysis of Deviance* table with one row for each predictor (or group of

predictors) and columns indicating the degrees of freedom, (unscaled) drop in deviance from the previous row and residual deviance:

Predictor(s)	df of drop in deviance	Drop In Deviance	df of res dev.	Residual Deviance
NULL			$n-1$	$D(\overline{Y}, Y)$
X_1	1	$D(\overline{Y}, \hat{Y}_{\{1\}})$	$n-2$	$D(\hat{Y}_{\{1\}}, Y)$
X_2	1	$D(\hat{Y}_{\{1\}}, \hat{Y}_{\{2\}})$	$n-3$	$D(\hat{Y}_{\{2\}}, Y)$
$(X_3, \ldots, X_q)$	$q-2$	$D(\hat{Y}_{\{2\}}, \hat{Y}_{\{q\}})$	$n-q-1$	$D(\hat{Y}_{\{q\}}, Y)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(X_{p-3}, X_{p-2})	2	$D(\hat{Y}_{\{p-4\}}, \hat{Y}_{\{p-2\}})$	$n-p-1$	$D(\hat{Y}_{\{p-2\}}, Y)$
X_{p-1}	1	$D(\hat{Y}_{\{p-2\}}, \hat{Y}_{\{p-1\}})$	$n-p$	$D(\hat{Y}_{\{p-1\}}, Y)$

where the notation $\hat{Y}_{\{k\}}$ indicates the fitted values based on a model containing the first k predictors, $X_1, \ldots, X_k$. Note that the first row of the table is the "NULL" row, which is for the model with just an intercept term. Also, note that there are two columns containing degrees of freedom values, the first is the degrees of freedom associated with the drop in deviance (and thus is the proper one for use in hypothesis testing) and the second is the degrees of freedom associated with the residual deviance (which is analogous to the residual degrees of freedom in normal linear regression models, and is thus the appropriate value for use in the t-multiplier of confidence intervals). The rows of the table will be displayed in the order that the predictors were included into the model, and note that predictors can be grouped together (in *S-Plus* this is accomplished by using a matrix object in the model formula). As a final note of caution, we point out that the values given in the drop in deviance column are "unscaled". In other words, we must divide by $\hat{\phi}_L$, which we will generally take to be $\hat{\phi}_{full}$ the dispersion estimate from the full (i.e., largest) model. Examples will serve to illustrate the deviance table most effectively.

Example 3 - Normal Linear Regression as a GLM: Recall the blood pressure versus age data from Exercise 1 of Tutorial Week 6. If we ignore the heteroscedasticity, we can fit a standard normal linear regression to this data in two ways:

```
> bp <- read.table("BP.txt",header=T)
> attach(bp)
> names(bp)
[1] "age" "diasbp"
> bp.lm <- lm(diasbp ~ age)
> bp.glm <- glm(diasbp ~ age,family=gaussian)
> anova(bp.lm)
Analysis of Variance Table

Response:  diasbp

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value       Pr(F)
age        1  2374.968 2374.968 35.79284 2.05032e-07
Residuals 52  3450.365   66.353
> anova(bp.glm)
Analysis of Deviance Table

Gaussian model

Response:  diasbp
```

```
Terms added sequentially (first to last)
     Df Deviance Resid. Df Resid. Dev
NULL                    53   5825.333
 age  1 2374.968        52   3450.365
> summary(bp.glm)$dispersion
 Gaussian
 66.35317
> 1-pchisq(anova(bp.glm)$Deviance[2]/summary(bp.glm)$dispersion,1)
     Gaussian
 2.194532e-09
```

We can now see that the drop in deviance is exactly the same as the regression sum-of-squares. Further, the dispersion estimate is just the mean squared error, and so the F-statistic will be the same as the deviance statistic. However, the p-value for the deviance test is not quite the same as that for the F-test, due to the different comparison distributions employed.

Further, if we perform the more appropriate weighted regression using $w_i^2 = 1/x_i^2$, we again see the similarity between the `lm()` and the `glm()` approaches.

```
> wgts <- 1/(age^2)
> bp.lm1 <- lm(diasbp ~ age,weights=wgts)
> bp.glm1 <- glm(diasbp ~ age,family=gaussian,weights=wgts)
> anova(bp.lm1)
Analysis of Variance Table

Response:  diasbp

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value       Pr(F)
age        1  1.864409 1.864409 52.09383 2.22591e-09
Residuals 52  1.861051 0.035789
> anova(bp.glm1)
Analysis of Deviance Table

Gaussian model

Response:  diasbp

Terms added sequentially (first to last)
     Df Deviance Resid. Df Resid. Dev
NULL                    53   3.725461
 age  1 1.864409        52   1.861051
> summary(bp.glm1)$dispersion
   Gaussian
 0.03578945
> summary(bp.lm1)$coef
                Value Std. Error   t value    Pr(>|t|)
(Intercept) 55.831037 2.78093210 20.076376 0.00000e+00
        age  0.588828 0.08158219  7.217605 2.22591e-09
> summary(bp.glm1)$coef
                Value Std. Error   t value
(Intercept) 55.831037 2.78093210 20.076376
        age  0.588828 0.08158219  7.217605
```

As a final note, we point out that in the case of a weighted GLM, with weights ω_i^2, the form of the deviance is: $D(\hat{Y}, Y) = 2\sum_{i=1}^{n} \omega_i^2 \left[Y_i\{b(Y_i) - b(\hat{Y}_i)\} - \{c(Y_i) - c(\hat{Y}_i)\}\right]$. For instance, $\omega_i^2 = n_i$ in the logistic regression of Example 1, and $\omega_i^2 = w_i^2$ in Example 3 above (which is why the deviance is equal to the appropriate weighted sum of squares).

Example 4 - Insurance Claims: Data on the number of car insurance policy holders and the number of claims made by those policy holders was gathered in England during 1973. The number of policy holders and number of claims were broken down by the four general areas in which the policy holders lived, the engine size of the car being insured and the age group of the policy holder are shown in the following table:

		Age Group							
Engine		< 25		25-29		30-35		35+	
Size	District	Policies	Claims	Policies	Claims	Policies	Claims	Policies	Claims
<1 Litre	1	197	38	264	35	246	20	1680	156
	2	85	22	139	19	151	22	931	87
	3	35	5	73	11	89	10	648	67
	4	20	2	33	5	40	4	316	36
1-1.5 Litre	1	284	63	536	84	696	89	3582	400
	2	149	25	313	51	419	49	2443	290
	3	53	10	155	24	240	37	1635	187
	4	31	7	81	10	122	22	724	102
1.5-2 Litre	1	133	19	286	52	355	74	1640	233
	2	66	14	175	46	221	39	1110	143
	3	24	8	78	19	121	24	692	101
	4	18	5	39	7	68	16	344	63
>2 Litre	1	24	4	71	18	99	19	452	77
	2	9	4	48	15	72	12	322	53
	3	7	3	29	2	43	8	245	37
	4	3	0	16	6	25	8	114	33

Initially, we might think that the appropriate error structure for these data is a binomial distribution. However, once we realise that each policy holder may make more than one claim, it is clear that a better error structure is likely to be a Poisson distribution. So, if each policy holder in the $(i, j, k)^{\text{th}}$ cell of the table (i.e., the cell corresponding to level i of the age factor, level j of the engine size factor and level k of the district factor) makes a number of claims having a Poisson distribution with mean parameter λ_{ijk}, then the total number of claims within this cell, C_{ijk}, will again be Poisson (since the sum of independent Poisson random variables remains Poisson distributed) with mean parameter $N_{ijk}\lambda_{ijk}$, where N_{ijk} is the total number of policy holders in the appropriate table cell. Again, without any external information to guide us, we might start with the canonical link for this error structure, namely the logarithmic one. To start, we can fit the "fully interactive" model, which includes all the possible pairwise interactions as well as the three-way interaction, so that the model is:

$$\log\left\{\frac{E(C_{ijk})}{N_{ijk}}\right\} = \log(\lambda_{ijk}) = \mu + \tau_i + \alpha_j + \delta_k + \gamma_{\tau\alpha,ij} + \gamma_{\tau\delta,ik} + \gamma_{\alpha\delta,jk} + \gamma_{\tau\alpha\delta,ijk}$$

where μ is an "overall" mean, the τ's are the effects of the age groups, the α's are the effects of the engine sizes and the δ's are the effects of the districts. Furthermore, the γ's are the interaction effects. Of course, we must put some constraints on this model, and the easiest set of constraints

is, as usual, the "baseline" constraints, which set $\tau_1 = \alpha_1 = \delta_1 = 0$ along with all the interaction terms which contain a "1" in their subscripts. This model is the logical extension of the two-way ANOVA model structure with interaction to the case of three factors (and indeed, if we wanted to fit a standard three-way ANOVA model, this is exactly the model structure would employ, except of course we would use normal errors and an identity link function). To actually fit this model, we must construct appropriate indicators for each of the three factors and recall that the indicators for the interaction terms are derived from simply multiplying together the indicators for the individual factors. In this way, we see that the three-way interaction consists of all 27 possible products of one of the three indicators for each of the three factors, while each two-way interaction consists of all 9 possible products of the indicators from each the corresponding pair of factors. Note that this model therefore has $1+3+3+3+9+9+9+27 = 64$ parameters and we have only 64 data points. Fortunately, we do not need any extra degrees of freedom for a Poisson model, since we know that the dispersion parameter is $\phi = 1$ and therefore need not be estimated. However, if we wanted to estimate the dispersion as a simple diagnostic check on the Poisson error structure assumption we would be unable to do so for this model. Of course, we hope that all the interactions are not necessary to the model and can be removed. To test this, we can fit this model in *S-Plus* (recalling that the model actually has a weighted Poisson error structure with weights equal to the number of policies) and examine the deviance table:

```
> ins <- read.table("CarIns.txt",header=T)
> attach(ins)
> names(ins)
[1] "pols"  "clms"  "agegp" "dist"  "engsz"
> unique(agegp)
[1] <25   25-29 30-35 >35
> unique(dist)
[1] D1 D2 D3 D4
> unique(engsz)
[1] <1L    1-1.5L 1.5-2L >2L
> rtes <- clms/pols
> age2 <- ifelse(agegp=="25-29",1,0)
> age3 <- ifelse(agegp=="30-35",1,0)
> age4 <- ifelse(agegp==">35",1,0)
> esz2 <- ifelse(engsz=="1-1.5L",1,0)
> esz3 <- ifelse(engsz=="1.5-2L",1,0)
> esz4 <- ifelse(engsz==">2L",1,0)
> dst2 <- ifelse(dist=="D2",1,0)
> dst3 <- ifelse(dist=="D3",1,0)
> dst4 <- ifelse(dist=="D4",1,0)
> ages <- cbind(age2,age3,age4)
> eszs <- cbind(esz2,esz3,esz4)
> dsts <- cbind(dst2,dst3,dst4)
> ins.glm <- glm(rtes ~ ages*dsts*eszs,family=poisson,weights=pols)
Warning messages:
 linear convergence not obtained in  30  iterations. in: glm.fitter(x = X, y =
       Y, w = w, start = start, offset = offset, family = family, ....
```

So, our first attempt at fitting the model resulted in a warning message, indicating that *S-Plus* stopped iterating the *IRLS* procedure after 30 steps and it had not converged (to *S-Plus*'s satisfaction, anyway), which simply means that the parameter estimates and fitted values were still noticably changing from one iteration to the next. This is due to an internal default which tells the `glm()` function to stop after 30 iterations of the *IRLS* algorithm (almost all reasonable models fit to a dataset will have converged to their estimates before this number of iterations is reached, and indeed, the `summary` display for the output list from a `glm()` contains the element `iter` which indicates how many iterations of the *IRLS* algorithm were required for convergence to be achieved). However, we can override the default by using the optional argument `maxit=`, which tells `glm()` the maximum number of iterations it can use for the *IRLS* algorithm to reach appropriate parameter estimates (in fact in this case the algorithm never properly converges - which strongly indicates an inappropriate model, however setting `maxit=50` ensures the model gives the results we want):

```
> ins.glm <- glm(rtes ~ ages*dsts*eszs,family=poisson,weights=pols,maxit=50)
Warning messages:
 linear convergence not obtained in  30  iterations. in: glm.fitter(x = X, y =
       Y, w = w, start = start, offset = offset, family = family, ....
> anova(ins.glm)
Analysis of Deviance Table

Poisson model

Response:  rtes

Terms added sequentially (first to last)
               Df Deviance Resid. Df Resid. Dev
          NULL                    63   236.2590
          ages  3 80.86372        60   155.3952
          dsts  3 15.30839        57   140.0868
          eszs  3 88.66681        54    51.4200
     ages:dsts  9  6.56080        45    44.8592
     ages:eszs  9 10.40272        36    34.4565
     dsts:eszs  9  7.16685        27    27.2897
ages:dsts:eszs 27 27.28967         0     0.0000
> qchisq(0.95,9)
[1] 16.91898
> qchisq(0.95,27)
[1] 40.11327
```

Also, notice that we have fit the model with all possible interaction terms using an abbreviated model formula designation which employed the `*` operator which automatically performs the appropriate multiplications of columns to create the interaction indicators. This abbreviated model formula will result in an analysis of deviance table appropriate for testing whether the interactions are necessary in the model (of course, we could have arrived at this table by manually entering all the interaction terms, but that would be quite time consuming).

The analysis of deviance table shows that none of the interaction terms appear overly necessary in the model (i.e., their associated deviance statistics, which are the same as the drop in deviances reported in the table for this example since $\phi = 1$ for a Poisson model, are less than the 95% percentile of the appropriate chi-squared distribution). In addition, it also shows that the residual deviance for the full model is 0, which is due to the fact that we have exactly as many parameters as data points. Such a model is generally referred to as "saturated". Suppose now, that we wanted

to see if the district factor was significant, after having removed all of the unnecessary interaction terms from the model:

```
> ins.glm1 <- glm(rtes ~ ages + eszs + dsts,family=poisson,weights=pols)
> anova(ins.glm1)
Analysis of Deviance Table

Poisson model

Response:  rtes

Terms added sequentially (first to last)
     Df Deviance Resid. Df Resid. Dev
NULL                    63   236.2590
ages  3 80.86372        60   155.3952
eszs  3 90.10394        57    65.2913
dsts  3 13.87126        54    51.4200
> qchisq(0.95,3)
[1] 7.814728
```

So, it appears that districts are indeed significant. Indeed, all three factors are highly significant, and it remains to investigate the relationship between the claim rate and the factor levels:

```
> summary(ins.glm1)$coef
                   Value Std. Error     t value
(Intercept) -1.82173992 0.07678762 -23.7243958
   agesage2 -0.19101011 0.08285644  -2.3053139
   agesage3 -0.34495066 0.08137414  -4.2390700
   agesage4 -0.53667071 0.06995562  -7.6715884
   eszsesz2  0.16133698 0.05053239   3.1927440
   eszsesz3  0.39281049 0.05499780   7.1422944
   eszsesz4  0.56341239 0.07231533   7.7910494
   dstsdst2  0.02586819 0.04301579   0.6013650
   dstsdst3  0.03852393 0.05051157   0.7626754
   dstsdst4  0.23420533 0.06167328   3.7975172
```

All of the coefficients appear to be significantly different from zero except for those associated with the second and third levels of the district factor. This suggests that these categories can be amalgamated with the first level of the district factor (it turns out that the fourth district is listed as "London and other major cities", which could certainly explain its difference from other English districts, such as rural areas or small towns). We can test this using another analysis of deviance table:

```
> ins.glm2 <- glm(rtes ~ ages + eszs + dst4 + cbind(dst2,dst3),family=poisson,
+   weights=pols)
> anova(ins.glm2)
Analysis of Deviance Table

Poisson model

Response:  rtes

Terms added sequentially (first to last)
                 Df Deviance Resid. Df Resid. Dev
NULL                                63   236.2590
ages  3 80.86372                    60   155.3952
```

```
                eszs  3 90.10394        57    65.2913
                dst4  1 13.15643        56    52.1349
cbind(dst2, dst3)  2  0.71483         54    51.4200
> qchisq(0.95,2)
[1] 5.991465
```

So, our impression is confirmed and we can amalgamate the first three district factor levels. Also, note that the rows of the deviance table behave in a similar fashion to those of an analysis of variance table in that the deviances sum as we would expect, a fact which is demonstrated by the last two rows in the above deviance table summing to $0.71483 + 13.15643 = 13.87126$ which is equal to the drop in deviance in the final row of the previous analysis of deviance table (which is associated with the three indicators `dst2`, `dst3` and `dst4` grouped together).

Amalgamating the first three district factor levels amounts to just ignoring the indicators `dst2` and `dst3`, so that our final model would be:

```
> ins.glm3 <- glm(rtes ~ ages + eszs + dst4,family=poisson,weights=pols)
> summary(ins.glm3)$coef
                 Value Std. Error    t value
(Intercept) -1.8102073 0.07531995 -24.033570
   agesage2 -0.1890172 0.08282214  -2.282207
   agesage3 -0.3421109 0.08130117  -4.207946
   agesage4 -0.5327488 0.06978738  -7.633884
   eszsesz2  0.1622910 0.05051669   3.212621
   eszsesz3  0.3935180 0.05498449   7.156892
   eszsesz4  0.5653953 0.07227744   7.822570
       dst4  0.2184953 0.05853209   3.732914
```

We now note that we can use the deviance table for some simple diagnostic procedures. For example, the model we have arrived at is:

$$\log\left\{\frac{E(C_{ijk})}{N_{ijk}}\right\} = \beta_0 + \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + \beta_4 s_1 + \beta_5 s_2 + \beta_6 s_3 + \beta_7 d,$$

where the a's are the indicators for the age factor, the s's are the indicators for the engine size factor, and d is an indicator of the fourth district. This model can be recast in the form

$$\log\{E(C_{ijk})\} = \beta_0 + \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + \beta_4 s_1 + \beta_5 s_2 + \beta_6 s_3 + \beta_7 d + \log(N_{ijk}).$$

Now, we might consider a model which includes a coefficient for the "predictor" $\log(N_{ijk})$ to be estimated, instead of simply assuming that the value of this coefficient is 1. One useful representation is to give $\log(N_{ijk})$ the coefficient $(1 + \beta_8)$ so that the new model would be

$$\log\{E(C_{ijk})\} = \beta_0 + \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + \beta_4 s_1 + \beta_5 s_2 + \beta_6 s_3 + \beta_7 d + (1 + \beta_8)\log(N_{ijk}),$$

which can be rearranged to:

$$\log\left\{\frac{E(C_{ijk})}{N_{ijk}}\right\} = \beta_0 + \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + \beta_4 s_1 + \beta_5 s_2 + \beta_6 s_3 + \beta_7 d + \beta_8 \log(N_{ijk}).$$

So, we can now test whether our original model is reasonable by fitting this larger model (noting that the form still requires us to use weights equal to the N_{ijk}'s, since we are still modelling the rates and not the number of claims) and testing whether β_8 is equal to zero or not. This can

be done in two ways, either by examining a t-statistic for the coefficient estimate or by using the appropriate chi-squared test for the drop in deviance:

```
> ins.glm4 <- glm(rtes ~ ages + eszs + dst4 + log(pols),family=poisson,
+   weights=pols)
> summary(ins.glm4)$coef
                  Value Std. Error   t value
(Intercept) -1.74397840 0.23182296 -7.5228889
   agesage2 -0.17924250 0.08894372 -2.0152350
   agesage3 -0.32871584 0.09264766 -3.5480208
   agesage4 -0.49464554 0.14429923 -3.4279152
   eszsesz2  0.17406942 0.06384007  2.7266481
   eszsesz3  0.39434731 0.05505122  7.1632801
   eszsesz4  0.54780846 0.09281130  5.9023897
       dst4  0.19869517 0.08787594  2.2610873
  log(pols) -0.01473596 0.04881221 -0.3018909
> qt(0.975,64-9)
[1] 2.004045
> 2*(1-pt(abs(-0.301926),55))
[1] 0.7638472
> anova(ins.glm4)
Analysis of Deviance Table

Poisson model

Response:  rtes

Terms added sequentially (first to last)
          Df Deviance Resid. Df Resid. Dev
     NULL                    63   236.2590
     ages  3 80.86372        60   155.3952
     eszs  3 90.10394        57    65.2913
     dst4  1 13.15643        56    52.1349
log(pols)  1  0.09101        55    52.0439
> qchisq(0.95,1)
[1] 3.841459
> 1-pchisq(0.09101,1)
[1] 0.7628971
```

So, clearly the hypothesis that $\beta_8 = 0$ is not rejected, and our original model seems reasonable. Also note that the square of the t-statistic is $(-0.301926)^2 = 0.09115931$ which is approximately equal to the corresponding deviance value in this case (which again draws a parallel with the case of normal linear regression models, where the F-statistic for testing a single predictor is equal to the square of the associated t-statistic).

Finally, recall that we initially thought that a binomial model might be the appropriate one for these data. If we fit such a model, using a logistic link, we get:

```
> ins.glm5 <- glm(rtes ~ ages + eszs + dst4,family=binomial,weights=pols)
> range(fitted(ins.glm3))
[1] 0.0960433 0.3583243
> range(fitted(ins.glm5))
[1] 0.09546814 0.33439682
```

```
> summary(fitted(ins.glm5)-fitted(ins.glm3))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.02393 -0.001156 -6.269e-05 -0.001311 0.0007424 0.002624
```

So, the fitted values seem to be extremely close for both model approaches. The reason for this is that, even for the largest estimated rate of approximately 0.358, the probability that a single policy holder makes more than one claim is just:

$$1 - (0.358 + 1)e^{-0.358} \approx 0.05,$$

so that the difference between the binomial approach which requires that each individual can make no more than one claim will give a reasonable approximation to the Poisson approach in this case. [NOTE: If X has a Poisson distribution with parameter λ, then we know that

$$Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

which means that $Pr(X = 0) = e^{-\lambda}$ and $Pr(X = 1) = \lambda e^{-\lambda}$. Thus,

$$Pr(X > 1) = 1 - Pr(X \leq 1) = 1 - \{Pr(X = 0) + Pr(X = 1)\} = 1 - e^{-\lambda} - \lambda e^{-\lambda} = 1 - (\lambda + 1)e^{-\lambda},$$

which is how the preceding calculation was derived.] Unfortunately, since these two different models are not nested (i.e., they have different links and error structures) we cannot use deviances to help decide which one is a better description of the observed data. The only way that we can assess the relative merits of non-nested models is through examination of various diagnostic tools, which we now discuss in detail.

V. Model Diagnostics

As with classical linear regression, the focus of model diagnostics for GLMs is based on examining how the fitted values differ from the observed values and assessing whether the discrepancies are "reasonable". For normal linear regression models, the most important diagnostic tools available were the residuals, $e_i = Y_i - \hat{Y}_i$. So, we might consider examining the adequacy of our generalised linear model by using the residuals. However, for a GLM, the quantities e_i do not behave as nicely as they did for normal linear regression. In particular, we know that the variance of the Y_i's is not constant, but rather is instead proportional to the variance function $V(\mu_i)$. Thus, even if the chosen model is correct, the residuals will not display a homoscedastic spread.

To account for this problem, we can construct the Pearson residuals as:

$$r_i = \frac{e_i}{\sqrt{V(\hat{Y}_i)}},$$

⇒ could discover over-dispersion

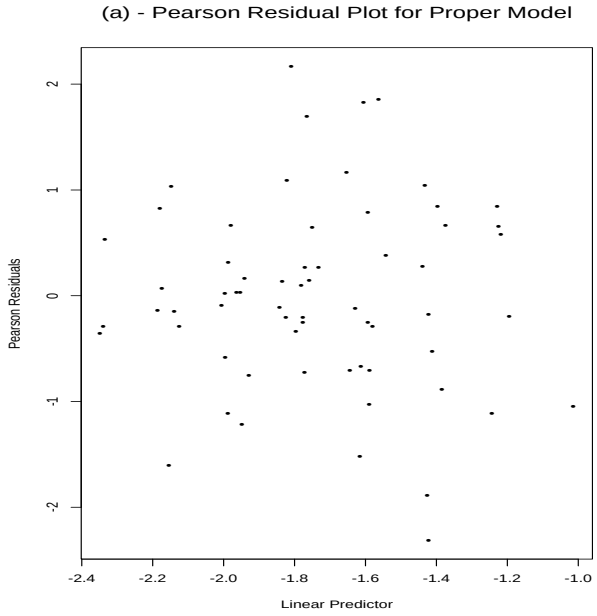
which should now display a homoscedastic spread if the variance function of the chosen model is correct. Recall that we have assumed that the dispersion parameter ϕ is constant for all the data points. If a plot of the Pearson residuals against the fitted values shows a heteroscedastic spread, this indicates that the assumption of constant ϕ is not satisfied, and that there is *overdispersion* in the data (actually, it is generally advisable to plot against the linear predictor values, $\hat{\eta}_i = x_i^T \hat{\beta}$, since these values will tend to be more evenly spread across the horizontal axis, making pattern recognition easier and more reliable). Overdispersion indicates that some component of the model structure needs modification. Overdispersion problems are occasionally fixed by a change in the choice of link function, but more often we need to pick a new error structure for the model with

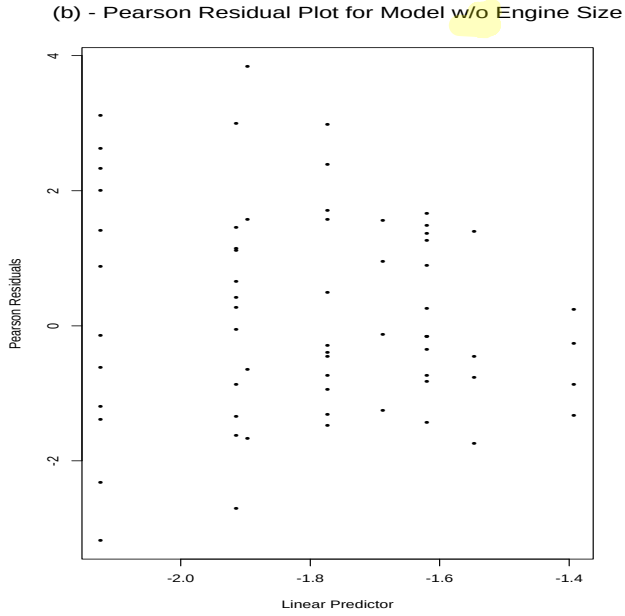
fix: new error structure

a variance function, $V(\mu)$, which more closely matches the behavior of the observed residuals. (NOTE: In some sense, the link function and the variance function are all that are required to fit a GLM, so that we do not truly need to have a full specification of an error distribution. In other words, we can merely pick a link function and then assume that the error structure for the model is just some distribution with the appropriate variance structure. This is the basic scheme of so-called *quasi-likelihood*, which is a more advanced and robust method of model fitting.) Another common source of overdispersion is due to the omission of important predictor variables.

Example 4 (Continued): For the model which includes the age group factor, the engine size factor and the indicator for the fourth district, the Pearson residuals plotted against the linear predictor values (see Plot *a* below) show an even spread indicating that the assumption of constant dispersion is satisfied. On the other hand, if we fit a model without the engine size factor and plot the Pearson residuals from this analysis (see Plot *b* below) we can see a distinct overdispersion problem.

```
> plot(ins.glm3$linear.predictors,residuals(ins.glm3,"pearson"),
+   xlab="Linear Predictor",ylab="Pearson Residuals",
+   main="(a) - Pearson Residual Plot for Proper Model")
> ins.glm6 <- glm(rtes ~ ages + dst4,family=poisson,weights=pols)
> plot(ins.glm6$linear.predictors,residuals(ins.glm6,"pearson"),
+   xlab="Linear Predictor",ylab="Pearson Residuals",
+   main="(b) - Pearson Residual Plot for Model w/o Engine Size")
```





Of course, plots of the Pearson residuals can also be examined for trends which might indicate that the chosen link function is inappropriate or that some of the predictors may need to be transformed (this is exactly analogous to looking for trends in the residual plot of a normal linear regression to assess the linearity of the data). Neither of the above plots appear to indicate any trends which would indicate a new link or transformation of the predictor variables was necessary.

We should be careful in our interpretations of the Pearson residual plot, however, since these residuals do not behave as the residuals for a normal linear regression do, and thus we cannot apply the same criteria to them as we did to the residuals in the normal linear case. In particular, even though the Pearson residuals are now homoscedastic (under the assumption of a proper model) they are not normally distributed, and indeed need not even be symmetrically distributed and are often quite skewed. Furthermore, the true usefulness and appropriateness of the residuals for use in

diagnostics arose from the fact that the residual sum-of-squares could be used to estimate the scale parameter of the model and that differences in residual sums-of-squares could be used to compare the fit of different models. Recall that the place of the residual sum-of-squares for a GLM is taken by the residual deviance, and so we would like to define a residual which is based on this quantity. Specifically, we would like to define *deviance residuals*, d_i, so that

$$\sum_{i=1}^{n} d_i^2 = D(\hat{Y}, Y).$$

Clearly, this definition gives rise to the usual residuals in the case of a normal error structure with identity link since $D(\hat{Y}, Y) = SSE$ in this case. In general, however, we need to define

$$D_i = 2Y_i\{b(Y_i) - b(\hat{Y}_i)\} - 2\{c(Y_i) - c(\hat{Y}_i)\},$$

and then set the deviance residuals to be:

$$d_i = \begin{cases} \omega_i\sqrt{D_i} & \text{if } Y_i > \hat{Y}_i \\ -\omega_i\sqrt{D_i} & \text{if } Y_i < \hat{Y}_i \end{cases}$$

where the ω_i^2's are the additional weight values (which are equal to 1 if the GLM was unweighted, and, for example, are equal to $\omega_i^2 = n_i$ in the case of a logistic regression on proportion data, where the n_i's are the denominators associated with each observed proportion). Some simple algebra then verifies that the sum of the d_i^2's is indeed equal to the deviance. Note that if the observed value is larger than its associated fitted value we take the positive square root and otherwise we take the negative square root. This is done so that the sign of the deviance residual properly reflects whether the fitted value is above or below the observed value. We now give the form of D_i for the common exponential families (of course, as usual, *S-Plus* will calculate the values for us automatically):

$$D_i = \begin{cases} (Y_i - \hat{Y}_i)^2 & \text{Normal}(\mu, \sigma^2) \\ 2Y_i \log\left(\frac{Y_i}{\hat{Y}_i}\right) + 2(n_i - Y_i)\log\left(\frac{n_i - Y_i}{n_i - \hat{Y}_i}\right) & \text{Binomial}(n_i, \pi) \\ 2Y_i \log\left(\frac{Y_i}{\hat{Y}_i}\right) - 2(Y_i - \hat{Y}_i) & \text{Poisson}(\lambda) \\ -2\log\left(\frac{Y_i}{\hat{Y}_i}\right) + 2\left(\frac{Y_i - \hat{Y}_i}{\hat{Y}_i}\right) & \text{Gamma}(\alpha, \beta) \end{cases}$$

In addition, we note that the inclusion of the ω_i's in the formula for d_i ensures that the deviance residuals are properly weighted, so that if we were doing a weighted linear regression with normal errors and identity link, then $d_i = \omega_i e_i$, the proper weighted residual. In this same vein, we will actually define the Pearson residual to be:

$$r_i = \frac{\omega_i e_i}{\sqrt{V(\hat{Y}_i)}},$$

if the GLM in question had weights ω_i^2 (indeed, it is these values that *S-Plus* produces and which we plotted in Example 4, otherwise the residuals of the correct model including the engine size factor would have reflected a heteroscedasticity which was already being accounted for by the weights in the analysis; recall the comments made in the solution for Exercise 1 of Tutorial 6). Of course, *S-Plus* will take care of this for us automatically.

Again, the appropriate plot to examine for the deviance residuals is a plot of d_i versus the linear predictor values, $\hat{\eta}_i = x_i^T\hat{\beta}$. This plot should be examined for non-constant spread, which would indicate an overdispersion problem, and for trends, which would indicate an incorrect model

structure of some kind (we will see how other plots can help distinguish which part of the model structure needs attention). In addition, it is often useful to plot the deviance residuals (and the Pearson residuals, as well) versus the predictor variables to look for trends which might indicate an incorrect link or the need for transformation (or inclusion of a quadratic component) of the predictors. However, for discrete data problems such as binomial and Poisson models, caution in interpretation of the residual plot is required. For Poisson models, any residual associated with a fitted value less than 2 will generally cause distortion in the plot, even when the Poisson model is perfectly adequate. Similarly, for a binomial model, residuals with fitted values near either zero or one will cause plot distortion. Generally speaking, we should ignore the residuals for such data points when making determinations about the suitability of a discrete model from the residual plot.

Now, before moving on to more detailed graphical diagnostic plots, we point out that

$$\frac{1}{n-p}E\left(\sum_{i=1}^{n} d_i^2\right) \approx \phi.$$

For the normal case, this is just a statement that the expected value of the MSE is equal to the underlying variance for the normal errors. However, in the case of a binomial or Poisson model, this gives us a useful check on the assumption that $\phi = 1$. If $\sum_{i=1}^{n} d_i^2$ is much larger than $n-p$ than this is a strong indication of overdispersion in the data. As a rule of thumb, we will say that there is significant overdispersion for either a binomial or Poisson model if:

$$\frac{1}{n-p}\sum_{i=1}^{n} d_i^2 > 1 + 3\sqrt{\frac{2}{n-p}}.$$

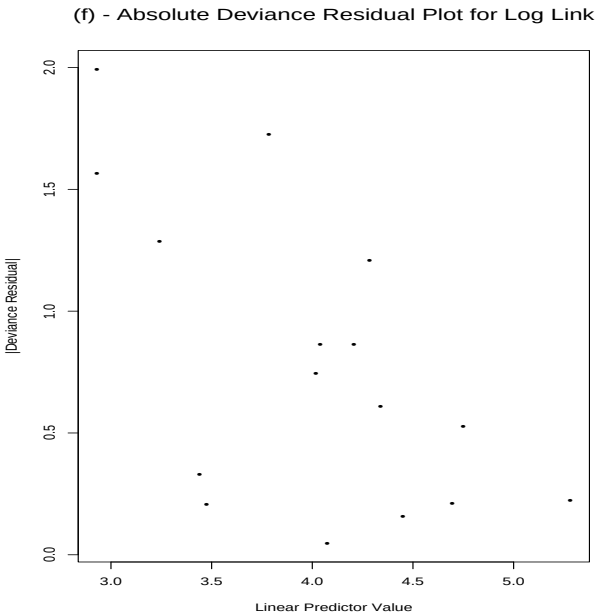
Finally, for the gamma distribution, we can use $\sum_{i=1}^{n} d_i^2/(n-p)$ as an estimate of dispersion and compare it to the CV estimator noted earlier. However, in this case the sum of the squared deviance residuals is highly sensitive to data points with fitted values near zero, and should not be used in cases where such data points occur.

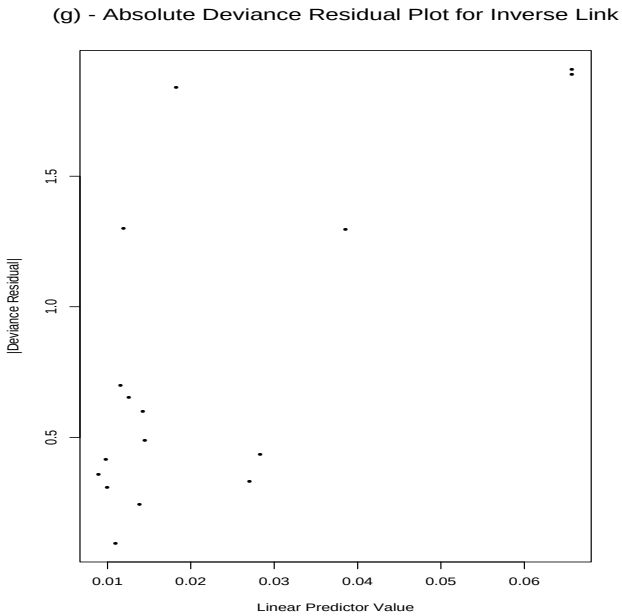
As has been mentioned, there are several possible causes of overdispersion in the residuals. We can try and isolate which of these causes is the one to focus on for a given dataset using several additional diagnostic plots. With each of these plots, however, the same cautions which went with the interpretation of the residual plots discussed above are still relevant.

1. To check whether the variance function, $V(\mu)$, of the chosen error structure is appropriate for the data, we can plot the absolute value of the residuals versus either the fitted values or the linear predictor values. If the variance function of the chosen error structure is adequately modelling the spread in the data, then there should be no upward or downward trend in this plot. An increasing trend indicates that the chosen variance function is not increasing quickly enough as the mean increases, while a decreasing trend in the absolute value of the residuals indicates the reverse. In other words, the variance function of any error structure, $V(\mu)$, models how the spread in the data changes as the expected value changes, and trends in the absolute residual plot indicate that the chosen relationship between the variance and the expectation is not correct. For example, the variance function for a Poisson model is $V(\mu) = \mu$, and an increasing trend in the absolute residuals indicates that the variance is actually increasing at a faster rate than this. Perhaps a more appropriate variance structure might be $V(\mu) = \mu^2$ in such a case, indicating that a gamma model is in order (though, of course a gamma error structure is a continuous one and the Poisson is discrete, so this is somewhat problematic, however, one might use a geometric error structure which is, in some sense, the discrete analog of the gamma family; of course, such an error structure is not directly implemented in *S-Plus* so more sophisticated computing would be required). Plots

of the absolute residuals versus the linear predictors for each of the links used in our analysis of the leukemia survival data of Example 2 show that there is perhaps some concern that the gamma family is not overly appropriate for this data. Recall that our informal check using the estimate of dispersion to examine the suitability of an exponential distribution for the data seemed to indicate that this assumption was justified. However, the point was made that this diagnostic was only truly relevant under the assumption that the overall gamma family was itself the correct error structure. These plots call that assumption into question.

```
> plot(leuk.glm$linear.predictor,abs(residuals(leuk.glm,"deviance")),
+  xlab="Linear Predictor Value",ylab="|Deviance Residual|",
+  main="(f) - Absolute Deviance Residual Plot for Log Link")
> plot(leuk.glm1$linear.predictor,abs(residuals(leuk.glm1,"deviance")),
+  xlab="Linear Predictor Value",ylab="|Deviance Residual|",
+  main="(g) - Absolute Deviance Residual Plot for Inverse Link")
```





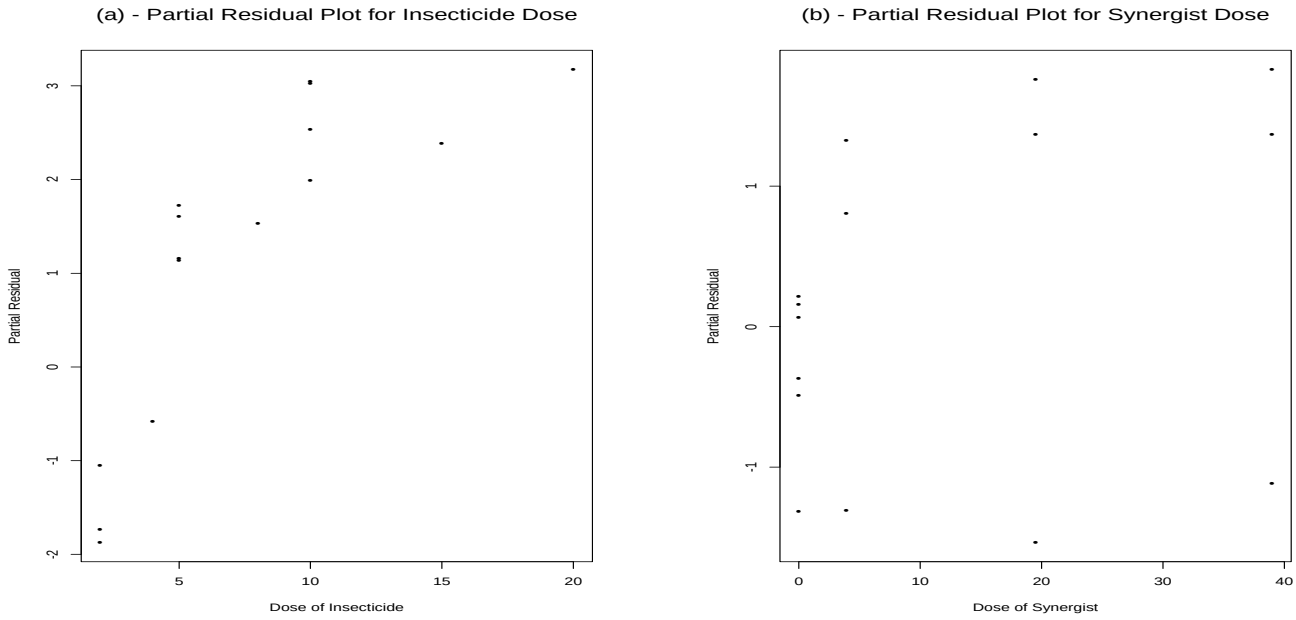
To check the adequacy of the choice of link function we plot $g(Y_i)$ versus $\hat{\eta}_i = x_i^T\hat{\beta}$, the fitted value of the linear predictor. If the link function is appropriate then this plot should look like a straight line, since the whole idea of a GLM is that the observed values are linearly related to the predictors on the scale determined by the link function g. Of course, we must be careful in our interpretation, since there is no reason for the plot to have uniform spread (i.e., heteroscedasticity is likely to be present, and indeed should be present for many of our common choices of error structure). The adequacy of the link function may also be strongly influenced by transformations of the predictor variables. So, we should generally check the appropriate scale for the predictors before examining the adequacy of the link function. To do this for the j^{th} predictor variable, X_j, we plot the *partial residual* values, $u_i = g(Y_i) - x_i^T\hat{\beta} + x_{i,j}\hat{\beta}_j$, versus the values of the predictor variable under investigation, the $x_{i,j}$'s themselves. In other words, we "adjust" the $g(Y_i)$ values for the other covariates in the model (i.e., we subtract off the predicted value calculated without using the predictor in question) and then plot these "residual-like" quantities versus the predictor value (which is an analog to the partial residual plot used in normal linear regression diagnostics).

Example 5 - Grasshopper Insecticide Potency: An experiment was run to investigate the potency of various mixtures of an insecticide and a synergist, a chemical which enhances the toxicity of the insecticide. The data on the dose levels and the number of grasshoppers killed are given below:

Number Killed	Sample Size	Dose of Insecticide	Dose of Synergist	Number Killed	Sample Size	Dose of Insecticide	Dose of Synergist
7	100	4	0	76	100	10	3.9
59	200	5	0	4	100	2	19.5
115	300	8	0	57	100	5	19.5
149	300	10	0	83	100	10	19.5
178	300	15	0	6	100	2	39
229	300	20	0	57	100	5	39
5	100	2	3.9	84	100	10	39
43	100	5	3.9				

Initially, we have no reason to choose any particular link function, so we start with the canonical link and examine the scale of the predictors:

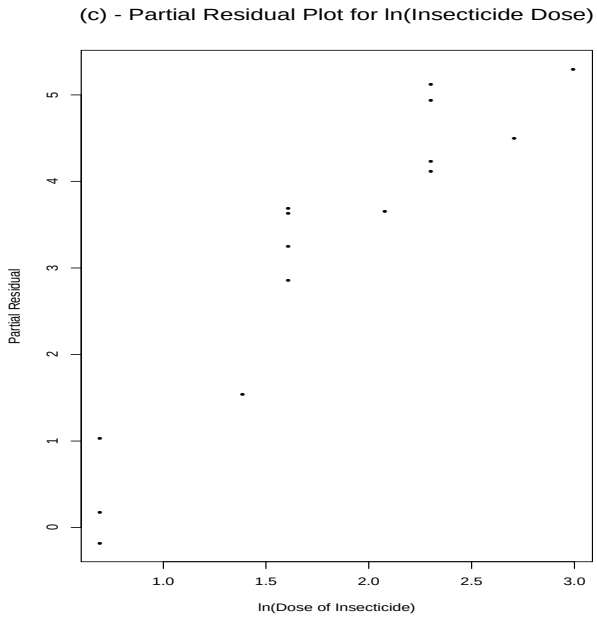
```
> ghp <- read.table("Grasshopper.txt",header=T)
> attach(ghp)
> names(ghp)
[1] "idose" "sdose" "ssize" "klld"
> prp <- klld/ssize
> ghp.glm <- glm(prp ~ idose+sdose,family=binomial,weights=ssize)
> u1 <- log(prp/(1-prp))-ghp.glm$coef[1]-(ghp.glm$coef[3]*sdose)
> u2 <- log(prp/(1-prp))-ghp.glm$coef[1]-(ghp.glm$coef[2]*idose)
> plot(idose,u1,xlab="Dose of Insecticide",ylab="Partial Residual",
+  main="(a) - Partial Residual Plot for Insecticide Dose")
> plot(sdose,u2,xlab="Dose of Synergist",ylab="Partial Residual",
+  main="(b) - Partial Residual Plot for Synergist Dose")
```

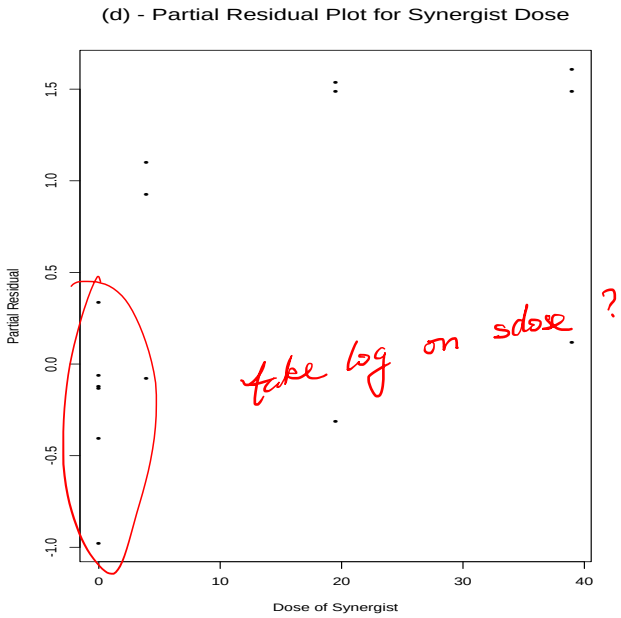


So, the scale of both the predictors seems to require some transformation. If we use a logarithmic transform for `idose` the corresponding partial residual plots are:

```
> ghp.glm1 <- glm(prp ~ log(idose)+sdose,family=binomial,weights=ssize)
> u1 <- log(prp/(1-prp))-ghp.glm1$coef[1]-(ghp.glm1$coef[3]*sdose)
> u2 <- log(prp/(1-prp))-ghp.glm1$coef[1]-(ghp.glm1$coef[2]*log(idose))
> plot(log(idose),u1,xlab="ln(Dose of Insecticide)",ylab="Partial Residual",
+  main="(c) - Partial Residual Plot for ln(Insecticide Dose)")
```

```
> plot(sdose,u2,xlab="Dose of Synergist",ylab="Partial Residual",
+  main="(d) - Partial Residual Plot for Synergist Dose")
```

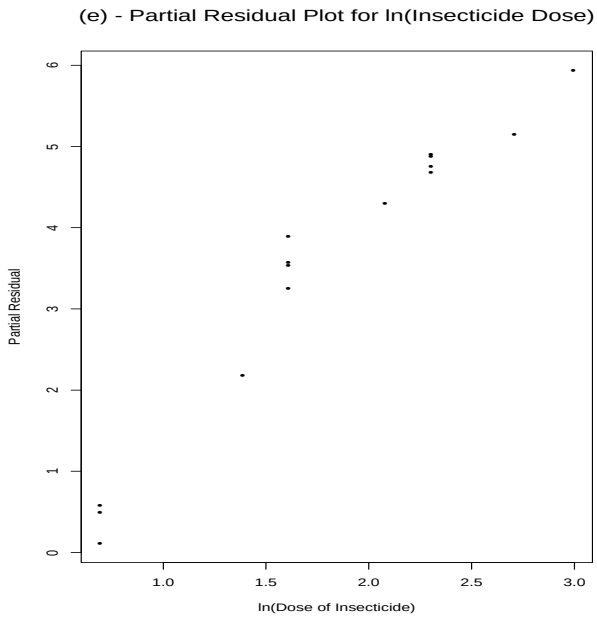


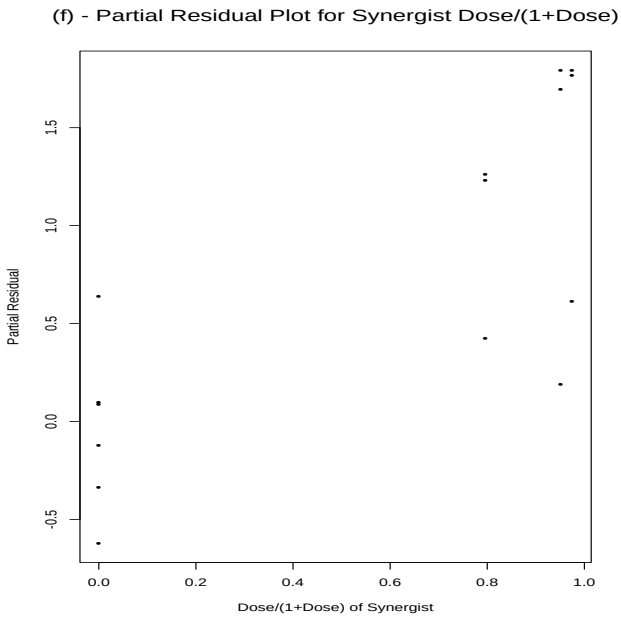


The scale of the `idose` predictor now appears quite reasonable. On the other hand, the scale of the synegist predictor still seems questionable, particularly if we ignore the two points in the lower right of the plot. Without these points, it appears that a logarithmic transformation is again in order, however, this is not possible due to the large number of zero values. Furthermore, external scientific information suggests that a transformation of the form $h(x) = x/(1+x)$ is potentially reasonable for this situation:

hmm, this needs some external knowledge.

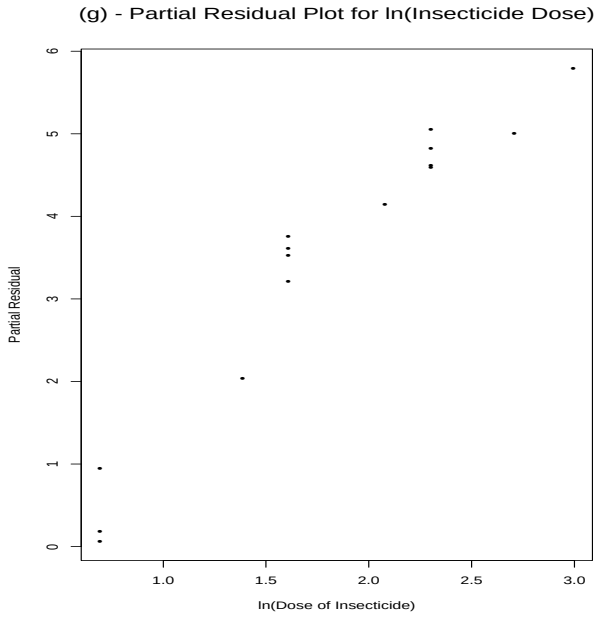
```
> nwsdose <- sdose/(1+sdose)
> ghp.glm2 <- glm(prp ~ log(idose)+nwsdose,family=binomial,weights=ssize)
> u1 <- log(prp/(1-prp))-ghp.glm2$coef[1]-(ghp.glm2$coef[3]*nwsdose)
> u2 <- log(prp/(1-prp))-ghp.glm2$coef[1]-(ghp.glm2$coef[2]*log(idose))
> plot(log(idose),u1,xlab="ln(Dose of Insecticide)",ylab="Partial Residual",
+  main="(e) - Partial Residual Plot for ln(Insecticide Dose)")
> plot(nwsdose,u2,xlab="Dose/(1+Dose) of Synergist",ylab="Partial Residual",
+  main="(f) - Partial Residual Plot for Synergist Dose/(1+Dose)")
```

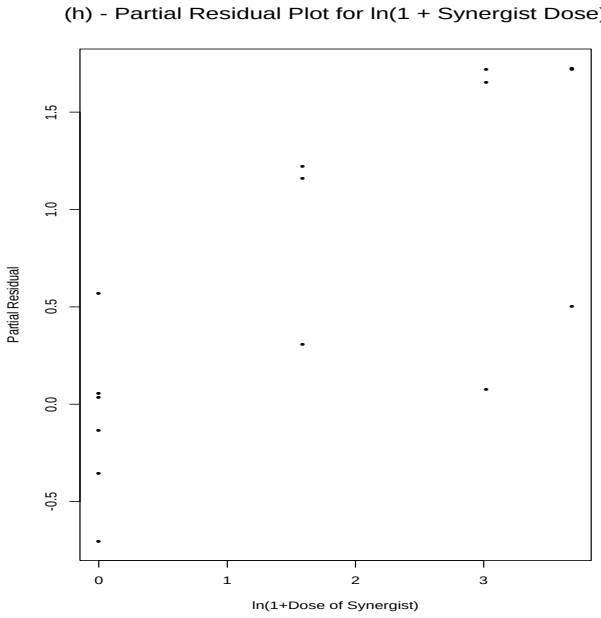




This new scale for `sdose` appears to fix the problem. Now, if we had not been in possession of the external information, we might have tried to use the logarithmic transformation, with a correction for the zero values, namely, $h(x) = \log(1 + x)$: → a potential fix for zero values.

```
> nwsdse1 <- log(1+sdose)
> ghp.glm3 <- glm(prp ~ log(idose)+nwsdse1,family=binomial,weights=ssize)
> u1 <- log(prp/(1-prp))-ghp.glm3$coef[1]-(ghp.glm3$coef[3]*nwsdse1)
> u2 <- log(prp/(1-prp))-ghp.glm3$coef[1]-(ghp.glm3$coef[2]*log(idose))
> plot(log(idose),u1,xlab="ln(Dose of Insecticide)",ylab="Partial Residual",
+  main="(g) - Partial Residual Plot for ln(Insecticide Dose)")
> plot(nwsdse1,u2,xlab="ln(1+Dose of Synergist)",ylab="Partial Residual",
+  main="(h) - Partial Residual Plot for ln(1 + Synergist Dose)")
```

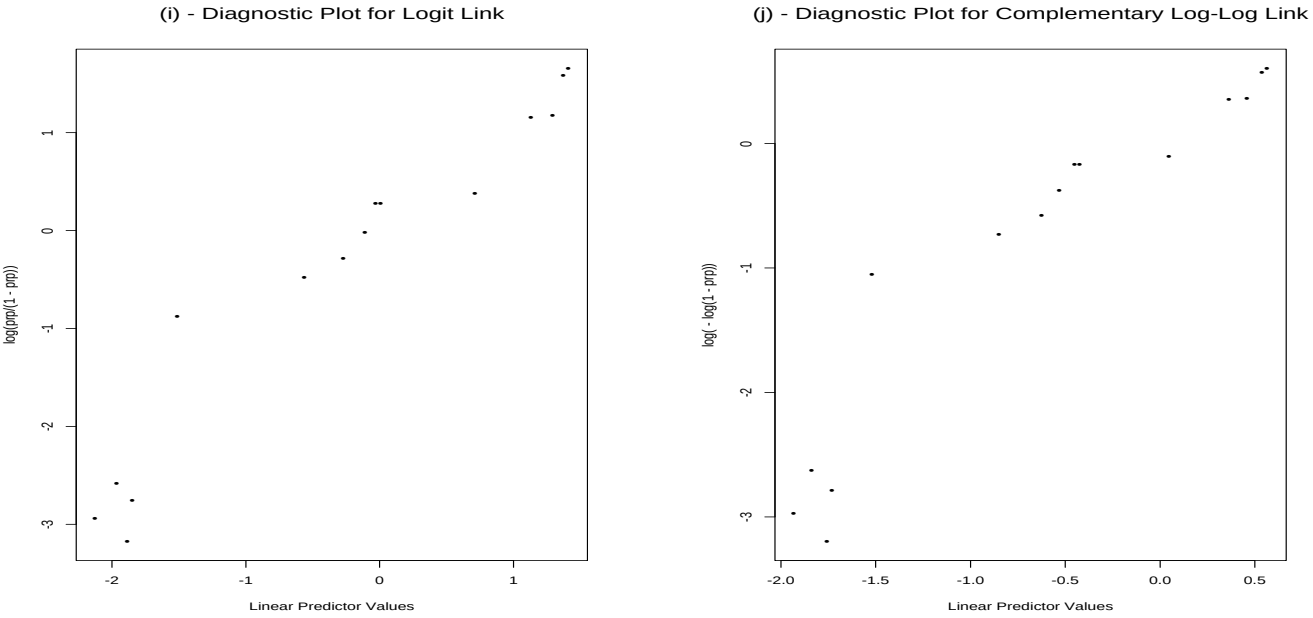




This second transformation also appears to yield a more reasonable partial residual plot, and thus may be an appropriate scale for the synergist predictor as well. However, the external information, combined with the fact that the preceding diagnostic plots were marginally better than the current plots indicates that the initial transformation of $h(x) = x/(1 + x)$ is the better one to employ. It

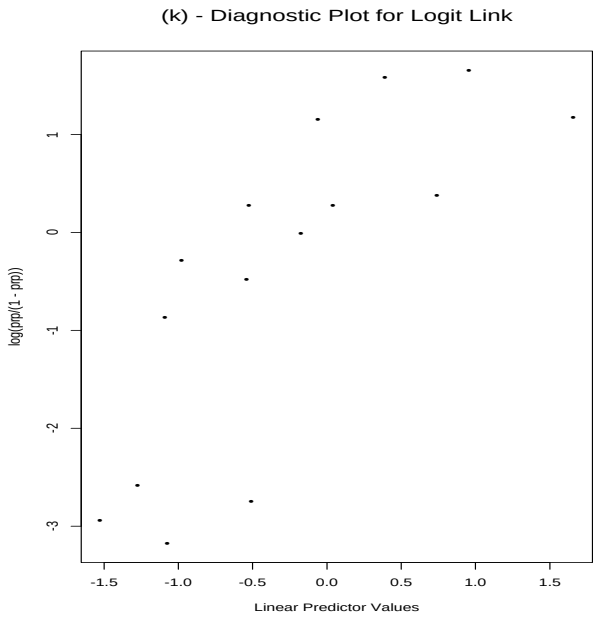
remains, then, to examine whether the logistic link function is appropriate. For the model with `sdose/(1+sdose)`, the appropriate diagnostic plots for the logistic and the complementary log-log links are:

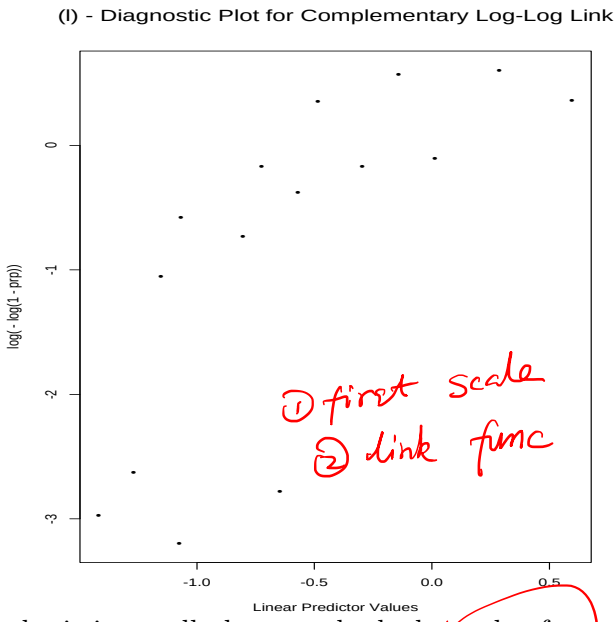
```
> plot(ghp.glm2$linear.predictor,log(prp/(1-prp)),
+  xlab="Linear Predictor Values",
+  main="(i) - Diagnostic Plot for Logit Link")
> ghp.glm2a <- glm(prp ~ log(idose)+nwsdose,family=binomial(link=cloglog),
+   weights=ssize)
> plot(ghp.glm2a$linear.predictor,log(-log(1-prp)),
+  xlab="Linear Predictor Values",
+  main="(j) - Diagnostic Plot for Complementary Log-Log Link")
```



From these plots, we can see that once the appropriate transformations of the data have been performed, there is no strong distinction between the two link functions, and both appear reasonable. This is in stark contrast to the corresponding plots for the original model using the untransformed predictors:

```
> plot(ghp.glm$linear.predictor,log(prp/(1-prp)),
+  xlab="Linear Predictor Values",
+  main="(k) - Diagnostic Plot for Logit Link")
> ghp.glma <- glm(prp ~ idose+sdose,family=binomial(link=cloglog),
+   weights=ssize)
> plot(ghp.glma$linear.predictor,log(-log(1-prp)),
+  xlab="Linear Predictor Values",
+  main="(l) - Diagnostic Plot for Complementary Log-Log Link")
```





where neither link appears to be appropriate. This is why it is usually best to check the scale of the predictors prior to checking the link function. Since, both links appear equally linearly related to the linear predictor values, we have no real diagnostic reason to prefer one over the other. However, the clearer interpretability of the logistic link function, combined with the fact that it is the canonical link in this case, indicates that it is probably the one to choose for this model.

VI. Influence and Outliers

The model diagnostics of the previous section deal with assessing the suitability of the chosen model for the observed data. In this section, we shall discuss diagnostics which help determine if an individual data point is not following the chosen model. The basic concepts will be exactly analogous to those for the similar diagnostic analysis in normal linear regression modelling; namely, leverage, influence and outliers. The leverage of a data point in a GLM is defined as h_{ii} the i^{th} diagonal element of the matrix:

$$H = W^{1/2}X(X^TWX)^{-1}X^TW^{1/2},$$

where X is again the design matrix (including the initial column of ones corresponding to the intercept parameter) and W is again the diagonal matrix of the weights, so that the i^{th} diagonal element of W is given by:

$$W_{ii} = \frac{\omega_i^2}{\{g'(\hat{Y}_i)\}^2 V(\hat{Y}_i)},$$

where the ω_i^2 are any additional weights used in a weighted GLM. Also, $W^{1/2}$ is just the matrix square root of W, which is simply the diagonal matrix with i^{th} diagonal element equal to $\sqrt{W_{ii}}$. We note the strong similarity of this definition to the case of normal linear regression, and indeed in this case, the H matrix defined here reduces exactly to the so called "hat matrix" of multiple linear regression. This parallel structure is further strengthened by the fact that, as for normal linear regression models:

$$\sum_{i=1}^{n} h_{ii} = p,$$

where p is the number of parameters (including the intercept) in the model. Thus, we will say that a point has large "leverage" if $h_{ii} > 2p/n$, just as we did for normal linear regression. Of course,

as for normal linear regression, just because a point has high leverage does not necessarily mean that it is "influential" (and we shall give a measure of influence shortly). However, they are points which warrant investigation. As a final note, we point out that, unlike the case for multiple linear regression, a data point with an x-value which is far from the rest of the predictor values will not necessarily have high leverage, since the matrix H now also includes the W matrix, so that if a point has a small value of W_{ii} then this may counteract the effect of an extreme covariate value.

Our analogy with the case of normal linear regression is seen to be complete when we note that in fact it can be shown (with some considerable algebraic effort) that:

$$Var(r_i) \approx Var(d_i) \approx \hat{\phi}(1 - h_{ii}).$$

This result leads to the definition of *Studentised* Pearson and deviance residuals in exactly the same way that Studentised residuals were defined for normal linear regression. However, since these variance results are now approximate (rather than exact as they were in the normal case) these Studentised residuals are rarely used. Another method of "correcting" the residuals to more accurately assess outliers is through the use of the *deletion residuals* (which are similar to the so-called *PRESS* residuals of multiple linear regression). In principle, the i^{th} deletion residuals, r_i^* and d_i^*, are defined in terms of a comparison between the models fit with and without the i^{th} data point. Specifically,

$$r_i^* = \frac{\omega_i(Y_i - \hat{Y}_{i,-i})}{\sqrt{V(\hat{Y}_{i,-i})}}$$

and

$$d_i^* = \begin{cases} \omega_i\sqrt{D_i^*} & \text{if } Y_i > \hat{Y}_{i,-i} \\ -\omega_i\sqrt{D_i^*} & \text{if } Y_i < \hat{Y}_{i,-i} \end{cases} \qquad \text{with} \qquad D_i^* = 2Y_i\{b(Y_i) - b(\hat{Y}_{i,-i})\} - 2\{c(Y_i) - c(\hat{Y}_{i,-i})\},$$

where the ω_i^2's are again any additional weights used in the GLM, $\hat{Y}_{i,-i} = g^{-1}(x_i^T\hat{\beta}_{-i})$ and $\hat{\beta}_{-i}$ is the maximum likelihood estimate derived from the model fit without the i^{th} data point. This can be quite time consuming to actually perform, since each refit of the GLM requires the *IRLS* procedure to be re-run. Recall that for multiple linear regression, special formulae were available to avoid the need for multiple refitting of the regressions. It turns out that GLM model refitting can also be avoided, however this can now only be accomplished at the cost of developing an approximation to the deletion residuals. Therefore, as with the Studentised residuals, the deletion residuals are rarely used in practice. Generally, then, assessment of outliers is performed using the deviance residuals themselves (which are preferred over the Pearson residuals), an outlier being determined as any point whose residual is "extreme" with respect to the rest of the values of the residuals.

Finally, we would like a measure of influence for each data point. Recall that influence is defined in terms of how much effect a particular data point has on the actual parameter estimates. To this end, we can define a quantity C_i, which is closely related to Cook's distance from multiple linear regression diagnostics, and indeed this quantity retains the name "Cook's distance" for GLMs. Specifically, we define:

$$C_i = \frac{(\hat{\beta}_{-i} - \hat{\beta})^T X^T W X(\hat{\beta}_{-i} - \hat{\beta})}{p\hat{\phi}},$$

where $\hat{\beta}_{-i}$ is again the parameter estimate from a model fit without the i^{th} data point and p the number of parameters. Clearly, data points which have a large Cook's distance are influential and should be further examined. However, assessing what constitutes a large value is often difficult.

large Cook's D ⇏ influential
but comparatively large CD ⇒ influential.

Generally speaking, it is not the actual value of a Cook's distance which constitutes an influential point, but whether its value is extreme in the context of the rest of the Cook's distances for the other data points. Thus, the usual method of assessing influence is to examine a barplot of C_i and note any values of C_i which appear to be excessively large with respect to the rest of the values.

Example 6 - Treatment for Disease in Carrot Crops: An experiment was performed to test the efficacy of a particular chemical at preventing disease damage in carrots. The data detailing the total number of carrots and number of damaged carrots at several different dosages of the chemical and in three different fields are given below:

	Damaged/Total Carrots		
Dose	Field *A*	Field *B*	Field *C*
1.52	10/35	17/38	10/34
1.64	16/42	10/40	10/38
1.76	8/50	8/33	5/36
1.88	6/42	8/39	3/35
2.00	9/35	5/47	2/49
2.12	9/42	17/42	1/40
2.24	1/32	6/35	3/22
2.36	2/28	4/35	2/31

n = 3×8 = 24

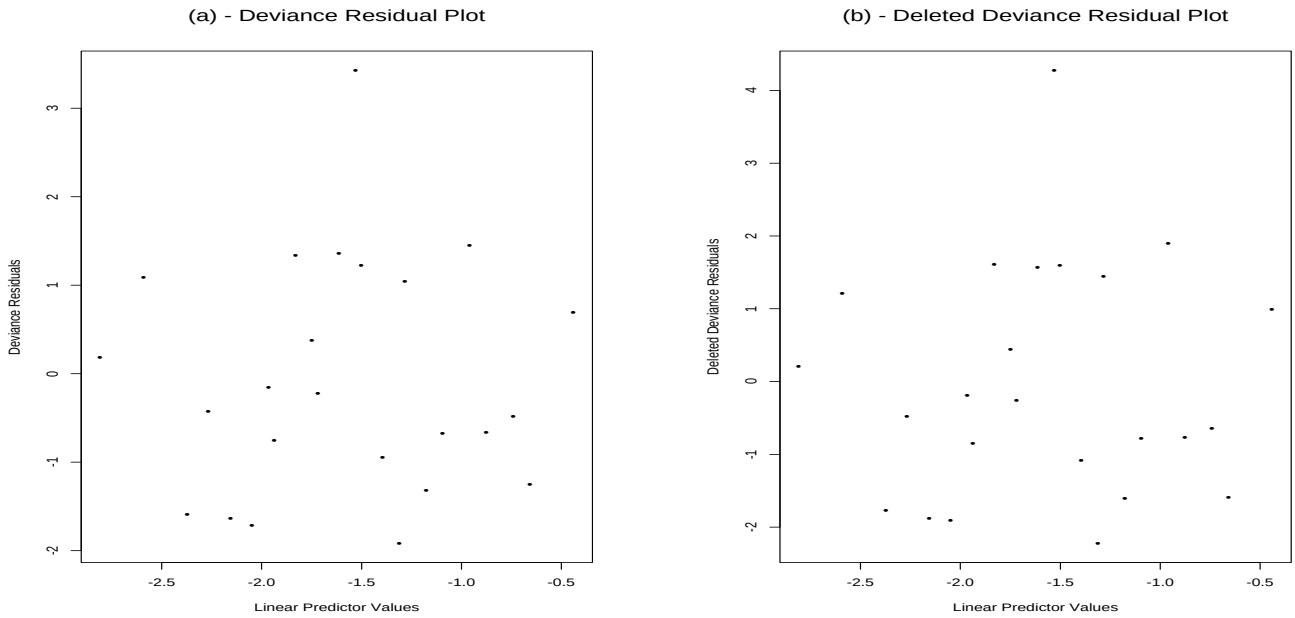
p = 3+1 = 4

If we fit a logistic regression to this data, we have:

```
> crts <- read.table("Carrots.txt",header=T)
> attach(crts)
> names(crts)
[1] "dose"  "field" "ni"    "dmg"
> prp <- dmg/ni
> fldb <- ifelse(field=="B",1,0)
> fldc <- ifelse(field=="C",1,0)
> crts.glm <- glm(prp ~ dose + fldb + fldc,family=binomial,weights=ni)
> plot(crts.glm$linear.predictor,residuals(crts.glm,"deviance"),
+  xlab="Linear Predictor Values",ylab="Deviance Residuals",
+  main="(a) - Deviance Residual Plot")
> as.vector(residuals(crts.glm,"deviance"))
 [1] -0.4774261  1.4533372 -1.3183505 -0.9449231  1.3588182  1.3394450
 [7] -1.7202178 -0.4246936  0.6941637 -1.2513449 -0.6650874 -0.6757535
[13] -1.9199610  3.4324058  0.3761215 -0.1562594  1.0485155  1.2279172
[19] -0.2231334 -0.7572206 -1.6303215 -1.5886992  1.0952426  0.1806589
> crts.glm$deviance/crts.glm$df.residual
[1] 1.998787
> 1+(3*sqrt(2/crts.glm$df.residual))
[1] 1.948683
```

number i ?

deviance / df res. > cut-off



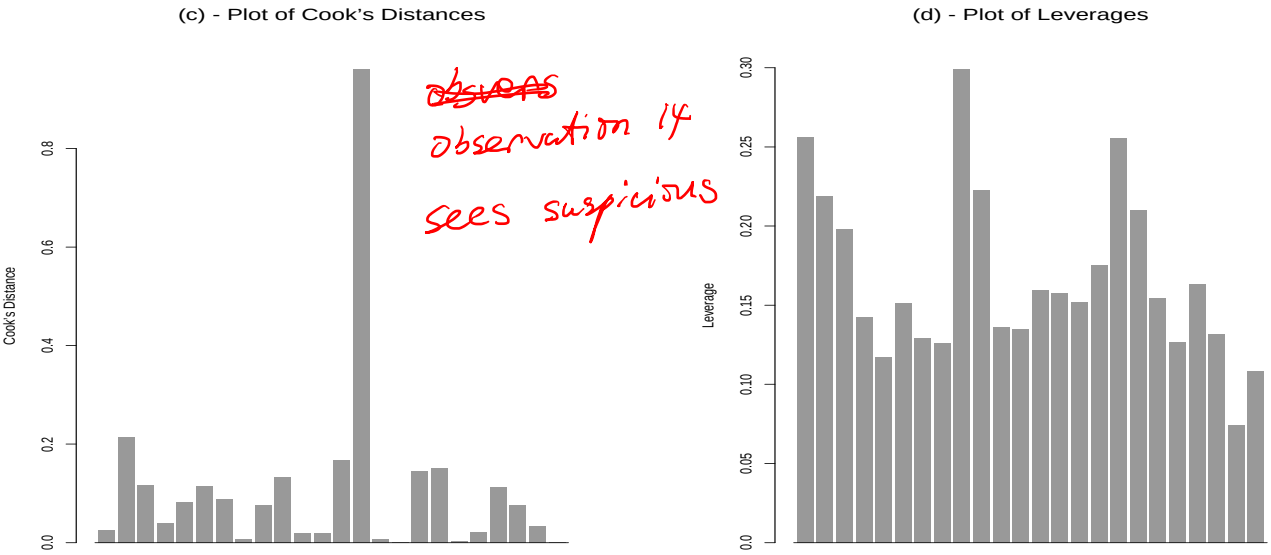
So, it appears that there is a dispersion problem, and it is likely caused by the apparent outlier, namely, the 14^{th} data point, the one at a dose of 2.12 in field B. This suspicion is further confirmed by using the deleted residuals and the Cook's distance:

```
> dstar <- 0
> CooksD <- 0
> for(i in 1:24) {
>  tmp <- glm(prp[-i] ~ dose[-i]+fldb[-i]+fldc[-i],family=binomial,
+   weights=ni[-i])
>  eta <- t(tmp$coef)%*%c(1,dose[i],fldb[i],fldc[i])
>  yhat <- exp(eta)/(1+exp(eta))
>  bigD <- (2*prp[i]*log(prp[i]/yhat))+
+   (2*(1-prp[i])*log((1-prp[i])/(1-yhat)))
>  dstar[i] <- sqrt(bigD*ni[i])
>  dstar[i] <- ifelse(prp[i]<yhat,-dstar[i],dstar[i])
>  CooksD[i] <- t(tmp$coef-crts.glm$coef)%*%
+   solve(summary(crts.glm)$cov.unscaled)%*%(tmp$coef-crts.glm$coef)/4
> }
> dstar
 [1] -0.6380215  1.8952580 -1.6124361 -1.0886073  1.5652668  1.6156420
 [7] -1.9067808 -0.4811989  0.9957151 -1.5898860 -0.7655803 -0.7758647
[13] -2.2260705  4.2800582  0.4465188 -0.1888024  1.4427703  1.5971458
[19] -0.2626680 -0.8548478 -1.8756975 -1.7643970  1.2063149  0.2035237
> plot(crts.glm$linear.predictor,dstar,
+  xlab="Linear Predictor Values",ylab="Deleted Deviance Residuals",
+  main="(b) - Deleted Deviance Residual Plot")
> barplot(CooksD,ylab="Cook's Distance",xlab=" ",
+  main="(c) - Plot of Cook's Distances")
```

Note that the above *S-Plus* calculation of the deletion residuals and Cook's distance required refitting the model $n = 24$ times, which took some time (and indeed, once the sample size exceeds about 100 values, the time and computing power necessary to perform the refits is prohibitive).

We can also examine the leverage values for this data, and compare them to the cut-off value of $2p/n = 2(4/24) = 0.33$. From the barplot we see that none of the data points appear to have large leverage in this example (recall that points with large leverage tend to draw the fitted curve close to them and thus we would not expected outliers to have large leverage values).

```
> Xmat <- cbind(1,dose,fldb,fldc)
> Wmat <- diag(crts.glm$weights)
> Hii <- diag(sqrt(Wmat)%*%Xmat%*%summary(crts.glm)$cov.unscaled%*%
+   t(Xmat)%*%sqrt(Wmat))
> barplot(Hii,ylab="Leverage",xlab=" ",
+  main="(d) - Plot of Leverages")
```



We can formally test whether our suspect point is an outlier using methods similar to those for mean-shift outliers in normal linear regression. Specifically, we set up a model which includes an indicator for the point in question and then test whether this indicator is significant:

```
> p14 <- rep(0,24)
> p14[14] <- 1
> crts.glm1 <- glm(prp ~ dose + cbind(fldb,fldc) + p14,family=binomial,
+   weights=ni)
> anova(crts.glm1)
Analysis of Deviance Table

Binomial model

Response:  prp

Terms added sequentially (first to last)

                 Df Deviance Resid. Df Resid. Dev
              NULL                    23   83.34426
              dose  1 28.61519        22   54.72906
cbind(fldb,fldc)  2 14.75331        20   39.97575
               p14  1 14.68635        19   25.28940
> 1-pchisq(14.68635,1)
[1] 0.0001269625
```

So, clearly this point is an outlier. If we remove it, the resulting deviance from the logistic regression becomes:

```
> crts.glm2 <- glm(prp[-14] ~ dose[-14] + cbind(fldb[-14],fldc[-14]),
+   family=binomial,weights=ni[-14])
> crts.glm2$deviance
[1] 25.2894
> crts.glm2$deviance/crts.glm2$df.residual
[1] 1.331021
> 1+(3*sqrt(2/crts.glm2$df.residual))
[1] 1.973329
```

showing that the dispersion is now quite reasonable.

$1.331021 < 1.973329$