Introduction to Generalised Linear Modelling using R

Deirdre Toher

2018

Contents

Context

The aim of this document is to give an introduction to Generalised Linear Modelling (GLM) and in particular the use of the R statistical package to fit such models. The basic theory of GLM is presented and descriptions of some standard generalised linear models are given. The use of R is demonstrated through examples and practical exercises are left for you to complete, both in scheduled practicals and in your own time.

1 Introduction

Analysis of variance and covariance (ANOVA / ANCOVA) can be referred to as general linear modelling. The data comprise multiple measurements made on groups of subjects and are assumed to be observations from a normal distribution usually with constant variance. In addition, the mean of the distribution is assumed to be a linear function of unknown parameters with known coefficients.

Generalised linear modelling (GLM) relaxes these assumptions:

- observations may come from a very general class of distributions;

- any twice differentiable one-to-one function of the mean is represented via a linear function of unknown parameters.

Since the normal distribution belongs to the permitted class of distributions then using this and the identity function means that general linear modelling is a special case of GLM. Dealing with data from normal distributions has the consequence that decision making processes are available that use χ^2, t and F distributions. However, for non-normal GLM we do not have this luxury and instead have to rely on a number of asymptotic results the properties of which are currently only partly understood. Thus we can see that GLM provides a broader choice of modelling opportunities than general linear modelling, however the latter does have the luxury of greater precision in decision making.

Prior to the advent of GLM, data from non-normal distributions was transformed to normality; for instance, taking the square root of Poisson data. Consequently the power of the normal distribution theory could be utilised. However, these transformations are themselves only asymptotic and the resulting model can appear very strange and artificial.

Nelder and Wedderburn (1972) initially proposed the GLM methodology. Two good references for GLM are:

- Dobson - An introduction to generalised linear models;
- McCullagh and Nelder - Generalised linear models.

The following books cover the application of GLM (and many other statistical areas) from the R perspective which is very similar to how one would do things in R:

- Crawley - Statistical Computing : An Introduction to Data Analysis using R;
- Venebles and Ripley - Modern Applied Statistics with R (S).

The examples of model building and evaluation in R are given by code in GLMpart1.R and the associated output is in the final section of this document for reference purposes. You are advised to download the R file from blackboard and run it in your own time. This version of the notes are written in Rmarkdown, which can be used for reproducible research; the code is embedded within the text of the document.

2 The Exponential Family

2.1 The Natural Exponential Family

We will assume that the observations come from a distribution in the natural exponential family of distributions. This means that the probability density function (pdf) can be written in the form:

$$f(y_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right\} \tag{1}$$

Here θ_i and ϕ are parameters and $a_i(\phi)$, $b(\theta_i)$ and $c(y_i, \phi)$ are known functions. In all models considered here the function $a_i(\phi)$ has the form

$$a_i(\phi) = \frac{\phi}{p_i},$$

where p_i is a known prior weight, often 1.

The parameters θ_i and ϕ are essentially location and scale parameters. It can be shown that if Y_i has a distribution in the exponential family then it has mean and variance:

$$\mathbf{E}(Y_i) = \mu_i = b'(\theta_i) \tag{2}$$
$$var(Y_i) = \sigma_i^2 = b''(\theta_i)a_i(\phi) \tag{3}$$

where $b'(\theta_i)$ and $b''(\theta_i)$ are the first and second derivatives of $b(\theta_i)$.

The exponential family just defined includes lots of very useful special cases, including the Normal, Binomial, Poisson, Exponential, Gamma and Inverse Gaussian distributions.

2.2 The Exponential Family (more general definition)

Previously, we have considered the definition of a member of the natural exponential family of distributions, which means that they can be expressed in the form:

$$f(x) = \exp\left(\frac{\theta x - b(\theta)}{a(\psi)} + c(x, \psi)\right)$$

Distributions of this form have canonical link functions (will be introduced in Sections 4.1 and 4.4). However, more formally, a member of the exponential family of distributions can be written as:

$$f(x) = \exp\left(\psi^t T(x) - A(\psi) + q(x)\right)$$

where $T(x)$ is a sufficient statistic for the distribution. The easiest way to calculate the sufficient statistic is actually to calculate the minimal sufficient statistic: $T(x)$ is a minimal sufficient statistic if:

$$\frac{L(\mathbf{x}_n|\theta)}{L(\mathbf{y}_n|\theta)} \text{is not a function of } \theta \Leftrightarrow T(\mathbf{x}_n) = T(\mathbf{y}_n)$$

The likelihood function $L(\mathbf{x}_n|\theta) = L(x_1, x_2, \ldots, x_n|\theta)$ of an iid sample from any distribution $f(x|\theta)$ is:

$$L(\mathbf{x}_n|\theta) = L(x_1, \ldots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta).$$

So, for the Normal distribution:

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

For $\mathbf{x}_n = (x_1, x_2, \ldots, x_n)$ iid random samples from the same Normal distribution:

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

and

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

Hence:

$$
\begin{aligned}
\frac{L(\mathbf{x}_n|\theta)}{L(\mathbf{y}_n|\theta)} &= \frac{\prod_{i=1}^n \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)}{\prod_{i=1}^n \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)} \\
&= \prod_{i=1}^n \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2} + \frac{(y_i-\mu)^2}{2\sigma^2}\right) \\
&= \exp\left(\sum_{i=1}^n -\frac{(x_i-\mu)^2}{2\sigma^2} + \frac{(y_i-\mu)^2}{2\sigma^2}\right) \\
&= \exp\left(\sum_{i=1}^n -\frac{x_i^2}{2\sigma^2} + \frac{2\mu x_i}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} + \frac{y_i^2}{2\sigma^2} - \frac{2\mu y_i}{2\sigma^2} + \frac{\mu^2}{2\sigma^2}\right) \\
&= \exp\left(\sum_{i=1}^n \frac{y_i^2 - x_i^2}{2\sigma^2} + \frac{\mu(x_i-y_i)}{\sigma^2}\right)
\end{aligned}
$$

So

$$\frac{L(\mathbf{x}_n|\theta)}{L(\mathbf{y}_n|\theta)}$$

is constant with respect to μ and σ^2 if and only if $\sum_i x_i = \sum_i y_i$ and $\sum_i x_i^2 = \sum_i y_i^2$. Therefore the minimal sufficient statistic for a $N(\mu;\sigma^2)$, where both μ and σ^2 are unknown is $T(\mathbf{x}_n) = (\sum_i x_i, \sum_i x_i^2)$.

If σ^2 was known, then only $\sum_i \mu(x_i - y_i)$ needs to be made constant with respect to μ – so the minimal sufficient statistic for μ if σ^2 is known is $\sum_i x_i$.

2.3 Proof that the Beta distribution is a member of the Exponential Family

Consider the $beta(a,b)$ distribution:

$$L(\mathbf{x}_n|a,b) = \prod_{i=1}^n \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x_i^{a-1}(1-x_i)^{b-1}$$

The (minimal) sufficient statistic is:

$$
\begin{aligned}
\frac{L(\mathbf{x}_n|a,b)}{L(\mathbf{y}_n|a,b)} &= \frac{\prod_{i=1}^n \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x_i^{a-1}(1-x_i)^{b-1}}{\prod_{i=1}^n \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} y_i^{a-1}(1-y_i)^{b-1}} \\
&= \frac{\left(\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\right)^n \prod_{i=1}^n x_i^{a-1}(1-x_i)^{b-1}}{\left(\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\right)^n \prod_{i=1}^n y_i^{a-1}(1-y_i)^{b-1}} \\
&= \frac{\prod_{i=1}^n x_i^{a-1}(1-x_i)^{b-1}}{\prod_{i=1}^n y_i^{a-1}(1-y_i)^{b-1}} \\
\log\left(\frac{L(\mathbf{x}_n|a,b)}{L(\mathbf{y}_n|a,b)}\right) &= \log\left(\frac{\prod_{i=1}^n x_i^{a-1}(1-x_i)^{b-1}}{\prod_{i=1}^n y_i^{a-1}(1-y_i)^{b-1}}\right) \\
&= \sum_{i=1}^n \log(x_i)^a + \sum_{i=1}^n \log(1-x_i)^{b-1} - \sum_{i=1}^n \log(y_i)^a - \sum_{i=1}^n \log(1-y_i)^{b-1}
\end{aligned}
$$

This is constant with respect to a if and only if $\sum_{i=1}^n \log(x_i) = \sum_{i=1}^n \log(y_i)$ and constant with respect to b if and only if $\sum_{i=1}^n \log(1-x_i)^{b-1} = \sum_{i=1}^n \log(1-y_i)^{b-1}$. Thus

$$T(\mathbf{x}_n) = \left(\sum_{i=1}^n \log(x_i), \sum_{i=1}^n \log(1-x_i) \right)$$

If $n = 1$ (a single observation):

$$T(x) = (\log(x), \log(1-x)) = \begin{pmatrix} \log(x) \\ \log(1-x) \end{pmatrix}$$

so:

$$\begin{aligned} f(x|a,b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \\ \log f(x|a,b) &= \log\Gamma(a+b) - \log\Gamma(a) - \log\Gamma(b) + (a-1)\log x + (b-1)\log(1-x) \\ f(x|a,b) &= \exp\{a\log x - \log x + b\log(1-x) - \log(1-x) \\ &\quad \log\Gamma(a+b) - \log\Gamma(a) - \log\Gamma(b)\} \\ &= \exp\left\{ (a\ b) \begin{pmatrix} \log x \\ \log(1-x) \end{pmatrix} - \begin{pmatrix} \log x \\ \log(1-x) \end{pmatrix} - A(a\ b) \right\} \end{aligned}$$

where $\psi^t = (a\ b)$ and

$$A(\psi) = A(a\ b) = \log\Gamma(a) + \log\Gamma(b) - \log\Gamma(a+b)$$

and:

$$q(x) = -\begin{pmatrix} \log x \\ \log(1-x) \end{pmatrix}$$

so that

$$f(x|a,b) = \exp\left\{\psi^t T(x) - A(\psi) + q(x)\right\} \text{ as required.}$$

2.4 Example: Normal distribution is a member of the Natural Exponential Family

The Normal distribution has density:

$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-1}{2} \frac{(y_i - \mu_i)^2}{\sigma^2} \right\}.$$

Recall that, to show that a distribution is a member of the natural exponential family, it must be possible to write it in the form:

$$f(y_i) = \exp\left\{ \frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi) \right\}$$

Now rewriting the density:

$$f(y_i) = \exp\left\{ \frac{-1}{2} \frac{(y_i - \mu_i)^2}{\sigma^2} - \frac{1}{2}\log\left(2\pi\sigma^2\right) \right\}$$

Expanding the square in the exponent we get

$$(y_i - \mu_i)^2 = y_i^2 + \mu_i^2 - 2y_i\mu_i.$$

The coefficient of y_i is $\frac{\mu_i}{\sigma^2}$. This result identifies θ_i as μ_i and ϕ as σ^2, with $a_i(\phi) = \phi$.

$$f(y_i) = \exp\left\{ \frac{y_i\mu_i - \frac{1}{2}\mu_i^2}{\sigma^2} - \frac{y_i^2}{2\sigma^2} - \frac{1}{2}\log\left(2\pi\sigma^2\right) \right\}.$$

$$f(y_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right\}$$

Comparing these equations and noting that $\mu_i = \theta_i$, then

$$b(\theta_i) = \frac{1}{2}\theta_i^2, \qquad b'(\theta_i) = \theta_i \qquad \text{and } b''(\theta_i) = 1.$$

Thus the mean and the variance then are:

$$\begin{aligned} \mathbf{E}(Y_i) &= \mu_i = b'(\theta_i) = \mu_i \\ var(Y_i) &= \sigma_i^2 = b''(\theta_i)a_i(\phi) = \sigma^2. \end{aligned}$$

2.5 In the Exponential Family

Distributions that can be shown to be members of the Exponential Family of Distributions:

- Normal / Gaussian distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

- Exponential distribution

$$f(x) = \begin{cases} \lambda \exp(-\lambda x), & x \geq 0 \\ 0, & x < 0 \end{cases}$$

- Bernouilli distribution

$$f(x) = p^x(1-p)^{1-x} \ x \in \{0, 1\}$$

- Binomial distribution

$$f(x) = \binom{n}{x} p^x(1-p)^{n-x}, \ x = 0, 1, 2, \ldots, n$$

- Poisson distribution

$$f(x) = \begin{cases} \frac{\lambda^x \exp(-\lambda)}{x!}, & \lambda > 0, \ x = 0, 1, 2, \ldots \\ 0, & \text{otherwise} \end{cases}$$

- Geometric distribution

$$f(x) = (1-p)^{x-1}p \text{ for } x = 1, 2, 3, \ldots$$

- Gamma distribution

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x) \text{ for } x, \alpha, \beta > 0$$

- χ^2 distribution

$$f(x) = \begin{cases} \frac{1}{2^{k/2}\Gamma(k/2)} x^{\frac{k}{2}-1} \exp(\frac{-x}{2}), & x \geq 0 \\ 0, & x < 0 \end{cases}$$

- Beta distribution

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

- Weibull (with known shape parameter (k)) distribution

$$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left(-\left(\frac{x}{\lambda}\right)^{k}\right), & x \geq 0 \\ 0, & x < 0 \end{cases}$$

- Inverse Gaussian distribution

$$f(x) = \left[\frac{\lambda}{2\pi x^3}\right]^{\frac{1}{2}} \exp\left(\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\right) \; x > 0, \; \mu > 0 \text{ (mean)}, \lambda > 0 \text{ (shape)}$$

- Negative binomial distribution (known r)

$$f(x) = \begin{pmatrix} x+r-1 \\ x \end{pmatrix} p^r (1-p)^x \quad \text{for } x = 0, 1, 2, \ldots$$

- Multinomial distribution

$$f(x_1, \ldots, x_k) = \begin{cases} \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0, & \text{otherwise} \end{cases}$$

- Dirichlet distribution (complicated distribution used as a conjugate prior for the multinomial distribution in Bayesian statistics)

Which of these can be shown to be in the Natural Exponential Family of distributions?

A youtube playlist can be found here

3 Not Exponential Family

Distributions that are not members of the Exponential Family of Distributions:

- Uniform distribution

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$$

- Cauchy distribution

$$f(x) = \frac{1}{\pi}\left[\frac{\beta}{(x-\alpha)^2 + \beta^2}\right] \; \beta > 0 \text{ (scale)}, \alpha \text{ (location)}$$

- Laplace family of distributions with non-zero mean

$$f(x) = \frac{1}{2\beta} \exp\left(-\frac{|x-\mu|}{\beta}\right) \ \beta > 0 \text{ (scale)}, \alpha \text{ (location)}$$

- Weibull distribution with unknown shape parameter

$$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp\left(-\left(\frac{x}{\lambda}\right)^{k}\right), & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Can you demonstrate why these are **not** in the exponential family?

4 The GLM Model

Let $y_1, \ldots, y_n$ denote n independent observations on a response. We treat y_i as a realisation of a random variable Y_i. In the general linear model we assume that Y_i has a normal distribution with mean μ_i and variance σ^2

$$Y_i \sim N(\mu_i, \sigma^2),$$

and we further assume that the expected value μ_i is a linear function of p predictors that take values $\mathbf{x}_i' = (x_{i1}, \ldots, x_{ip})$ for the i^{th} case, so that

$$\mu_i = \mathbf{x}_i \beta$$

where β is a vector of unknown parameters. We will generalise this in two steps, dealing with the stochastic and systematic components of the model.

4.1 Link function

The second element of the generalisation is that instead of modelling the mean, as before, we will introduce a one-to-one continuously differentiable transformation $g(\mu_i)$ and focus on:

$$\nu_i = g(\mu_i).$$

The function $g(\mu_i)$ is known as the **link function**. Examples of commonly used link functions include the identity, log, reciprocal, logit and probit functions. We further assume that the transformed mean follows a linear model, so that

$$\nu_i = \mathbf{x}_i' \beta.$$

The quantity ν_i is the **linear predictor**. Since the link (by construction) is one-to-one it is invertible, so we can then obtain:

$$\mu_i = g^{-1}(\mathbf{x}_i' \beta).$$

An important thing to note is that we are not transforming the response y_i but rather the **expected value of the response** μ_i. So, a model where $\log(y_i)$ is linearly dependent on x_i is not the same as a generalised linear model where $\log(\mu_i)$ is linear on x_i.

When the link function makes the linear predictor ν_i the same as the canonical parameter θ_i we have what is known as a **canonical link**. The **identity function** is the canonical link for the Normal distribution. We will see that the logit is the canonical link for the binomial distribution and the log is the canonical link for the Poisson distribution. Therefore the canonical link leads to some natural pairings of types of data with link functions. These do not preclude the use of other link functions, but have the advantage that a minimal sufficient statistic for β exists so that all the information about β is contained in a function of the data of the same dimensionality as β.

4.2 Poisson Errors and Log Link

Application of general theory to the Poisson case.

4.2.1 The Poisson Distribution

A Poisson random variable has the probability distribution

$$f_i(y_i) = \frac{\exp(-\mu_i)\mu_i^{y_i}}{y_i!}$$

for $y_i = 0, 1, 2, \ldots$. The mean and the variance of Y_i both equal μ_i.

$$\log f_i(y_i) = y_i \log(\mu_i) - \mu_i - \log(y_i!)$$

It is immediately apparent that $\theta_i = \log(\mu_i)$, this being the canonical parameter, indicating that the canonical link is the natural log. Solving for μ_i we see that the inverse link is $\mu_i = \exp(\theta_i)$. Therefore, $\mu_i = b(\theta_i) = \exp(\theta_i)$. The last term is a function of the data y_i, but not the parameter, therefore $c(y_i, \phi) = -\log(y_i!)$. Left to note is that $a_i(\phi) = \phi$ where $\phi = 1$.

To confirm the mean and the variance are as expected:

$$\mu_i = b'(\theta_i) = \exp(\theta_i) = \mu_i$$

and

$$v_i = a_i(\phi)b''(\theta_i) = \exp(\theta_i) = \mu_i$$

4.3 Binomial Errors and Logit Link

Application of the theory of generalised linear models to the case of binary data, in particular to logistic regression models.

4.3.1 The Binomial Distribution

Recall that the probability distribution function (pdf) of a Binomial distribution is:

$$f_i(y_i) = \left(\begin{array}{c} n_i \\ y_i \end{array} \right) \tau_i^{y_i}(1-\tau_i)^{n_i - y_i}.$$

Taking logs we see that

$$\log f_i(y_i) = y_i \log \tau_i + (n_i - y_i)\log(1-\tau_i) + \log \left(\begin{array}{c} n_i \\ y_i \end{array} \right).$$

Collect the y_i terms we see that:

$$\log f_i(y_i) = y_i \log \left(\frac{\tau_i}{1-\tau_i} \right) + (n_i)\log(1-\tau_i) + \log \left(\begin{array}{c} n_i \\ y_i \end{array} \right).$$

Comparing to Equation ((1))

$$f(y_i) = \exp\left\{ \frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi) \right\}$$

we can see that $a_i(\phi) = 1$, $\theta_i = \log\left(\frac{\tau_i}{1-\tau_i}\right)$.

Solving for τ_i:

$$\begin{aligned}
\theta_i &= \log\left(\frac{\tau_i}{1-\tau_i}\right) \\
\exp(\theta_i) &= \left(\frac{\tau_i}{1-\tau_i}\right) \\
(1-\tau_i)\exp(\theta_i) &= \tau_i \\
\exp(\theta_i) &= \tau_i(1+\exp(\theta_i)) \\
\frac{\exp(\theta_i)}{1+\exp(\theta_i)} &= \tau_i
\end{aligned}$$

so

$$1-\tau_i = \frac{1+\exp(\theta_i)}{1+\exp(\theta_i)} - \frac{\exp(\theta_i)}{1+\exp(\theta_i)} = \frac{1}{1+\exp(\theta_i)}$$

Therefore:

$$\log(1-\tau_i) = -\log(1+\exp(\theta_i))$$

and then

$$b(\theta_i) = n_i \log(1+\exp(\theta_i)).$$

The remaining term in the pdf is not a function of τ_i but is a function of y_i; so

$$c(y_i, \phi) = \log\left(\begin{array}{c} n_i \\ y_i \end{array}\right).$$

Previously we noted that $a_i(\phi) = 1$, however this is actually because $\phi = 1$ and we would claim that $a_i(\phi) = \phi$.

Now verifying the mean and the variance. Differentiate $b(\theta_i)$ with respect to θ_i to find that:

$$\mu_i = b'(\theta_i) = n_i \frac{\exp(\theta_i)}{1+\exp(\theta_i)} = n_i\tau_i,$$

as expected.

$$v_i = a_i(\phi)b''(\theta_i) = n_i \frac{\exp(\theta_i)}{(1+\exp(\theta_i))^2} = n_i\tau_i(1-\tau_i),$$

again agreeing with our knowledge of basic statistics.

4.4 Canonical Link functions

$$f(y_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right\}$$

In this form, θ is known as the canonical parameter and ϕ as the dispersion parameter. If $\theta = g(\mu)$ for some function g of the mean μ, then $g(\mu)$ is known as the canonical link function.

Table 1: Distributions with link function.

Distribution	Link Name	Link Function	Mean Function
Normal	Identity	$X\beta = \mu$	$\mu = X\beta$
Exponential	Inverse	$X\beta = \mu^{-1}$	$\mu = (X\beta)^{-1}$
Gamma	Inverse	$X\beta = \mu^{-1}$	$\mu = (X\beta)^{-1}$
Inverse Gaussian	Inverse Squared	$X\beta = \mu^{-2}$	$\mu = (X\beta)^{-1/2}$

Distribution	Link Name	Link Function	Mean Function
Poisson	Log	$X\beta = \log(\mu)$	$\mu = \exp(X\beta)$
Binomial	Logit	$X\beta = \log\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \frac{1}{1+\exp(-X\beta)}$
Multinomial	Logit	$X\beta = \log\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \frac{1}{1+\exp(-X\beta)}$

In the exponential family of distributions, the parameter θ is known as the canonical parameter. Knowing the expression for your canonical parameter will tell you your canonical link function.

4.4.1 Example: Gamma distribution

$$\begin{aligned} f(y) &= \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1}\exp(-\beta y) \text{ for } y,\alpha,\beta > 0 \\ &= \exp\left(-y\beta + (\alpha-1)\log y + \alpha\log\beta - \log\left(\Gamma(\alpha)\right)\right) \\ &= \exp\left(\frac{y\frac{-\beta}{\alpha} + \frac{\alpha}{\alpha}\log\beta}{\frac{1}{\alpha}} + (\alpha-1)\log y - \log\left(\Gamma\left(\alpha\right)\right)\right) \\ &= \exp\left(\frac{y\frac{\beta}{\alpha} - \log\beta}{\frac{-1}{\alpha}} + (\alpha-1)\log y - \log\left(\Gamma\left(\alpha\right)\right)\right) \end{aligned}$$

now if $\theta = \frac{\beta}{\alpha}$ and $\phi = \frac{-1}{\alpha}$ so that $\alpha = \frac{-1}{\phi}$ and $\beta = -\theta\alpha = \frac{-\theta}{\phi}$ then:

$$\begin{aligned} f(y) &= \exp\left(\frac{y\theta - \log\left(-\frac{\theta}{\phi}\right)}{\phi} + \left(-\frac{1}{\phi} - 1\right)\log y - \log\left(\Gamma\left(-\frac{1}{\phi}\right)\right)\right) \\ &= \exp\left(\frac{y\theta - \log\theta}{\phi} + \frac{\log -\phi}{\phi} - \left(\frac{1}{\phi} + 1\right)\log y - \log\left(\Gamma\left(-\frac{1}{\phi}\right)\right)\right) \end{aligned}$$

Now recall that $\mathbf{E}(Y_i) = \mu_i = b'(\theta_i)$ so that comparing to our expression above:

$$b(\theta) = \log(\theta)$$

so that

$$b'(\theta) = \frac{1}{\theta} = \mu = \frac{\alpha}{\beta}.$$

The variance can be found to be $b''(\theta)\phi = \frac{\phi}{-\theta^2} = \frac{-1}{\alpha}\frac{-\alpha^2}{\beta^2} = \frac{\alpha}{\beta^2}$. Then parameterise the distribution in terms of its expected value:

$$\begin{aligned} f(y) &= \exp\left(\frac{y\frac{1}{\mu} - \log\frac{1}{\mu}}{\phi} + \frac{\log -\phi}{\phi} - (\frac{1}{\phi} + 1)\log y - \log\left(\Gamma\left(-\frac{1}{\phi}\right)\right)\right) \\ &= \exp\left(\frac{y\mu^{-1} - \log\mu^{-1}}{\phi} + \frac{\log -\phi}{\phi} - \left(\frac{1}{\phi} + 1\right)\log y - \log\left(\Gamma\left(-\frac{1}{\phi}\right)\right)\right) \\ &= \exp\left(\frac{y\mu^{-1} + \log\mu}{\phi} + \frac{\log -\phi}{\phi} - \left(\frac{1}{\phi} + 1\right)\log y - \log\left(\Gamma\left(-\frac{1}{\phi}\right)\right)\right) \end{aligned}$$

indicating that the canonical link function is:

$$g(\mu) = \frac{1}{\mu} = \mu^{-1}.$$

4.4.2 Further Examples

Use the same procedure to show that the canonical link functions for the Poisson and Binomial distributions are those listed in the table 1.

5 Poisson data

Natural model for simple counts:

$$y_i \sim \ Poisson(\mu_i)$$

where $g(\mu_i) = x_i'\beta$.

R allows the use of the three following links:

1. log: This is the most common link (hence also the default link).

   $$\log(\mu_i) = x_i'\beta$$

   Note with this link:

   $$\mu_i = \exp(x_i'\beta) = \prod_j \exp(x_{ij}\beta_j)$$

   that is a multiplicative structure for μ_i. With this link $\mu_i \geq 0$ and $-\infty < x_i'\beta < \infty$.

2. sqrt:

   $$\sqrt{(}\mu_i) = x_i'\beta$$

   Thus:

   $$\mu_i = (x_i'\beta)^2 .$$

   With this link $\mu_i \geq 0$ and $-\infty < x_i'\beta < \infty$.

3. identity:

   $$\mu_i = x_i'\beta$$

   Note: with this link μ_i is not guaranteed to be greater than zero and thus we have to be careful when using it.

5.1 Poisson Examples

5.1.1 Example 1: Poisson response with one covariate

The following data gives Poisson observations y for three values of a covariate x (x=0,1,2). Compare the fits of the model

$$\eta_i = \beta_0 + \beta^x x_i$$

under the log and sqrt links. Can the model be simplified?

x=0	x=1	x=2
1,0,1	2,3,2	3,4,2
3,0,2	1,0,2	0,5,1
0,1,2	1,4,1	4,2,3
1		3,1

Read in the data considering each level of the covariate in turn.

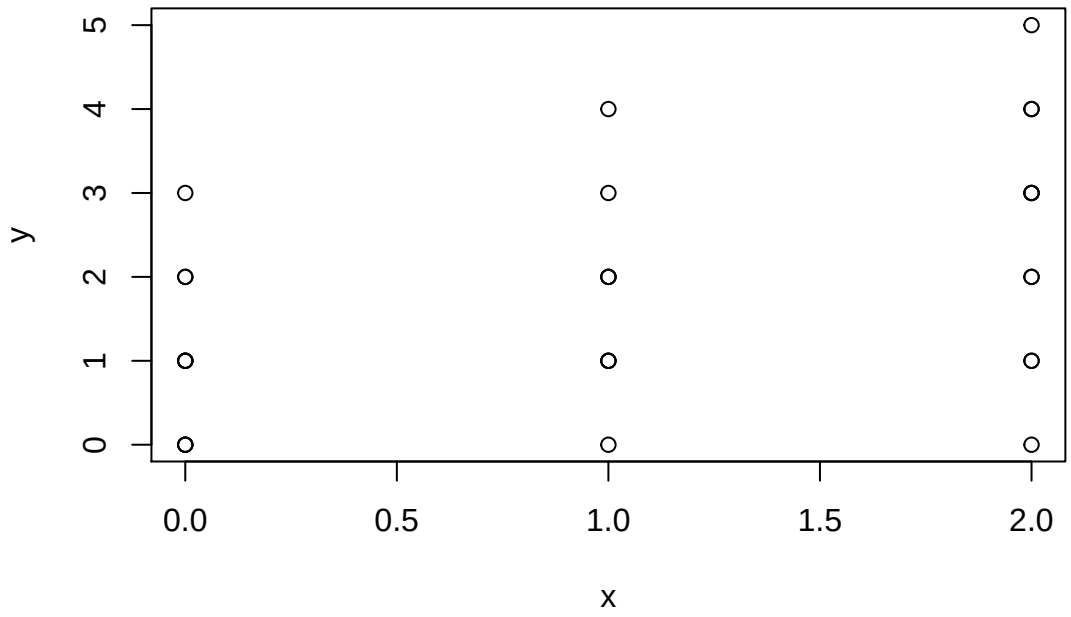

Figure 1: Exploratory Data: Analysis Poisson Counts

```
y<-c(1,0,1,3,0,2,0,1,2,1,2,3,2,1,0,2,1,4,1,3,4,2,0,5,1,4,2,3,3,1)
x<-c(rep(0,10),rep(1,9),rep(2,11))
```

Some Exploratory Data Analysis shown in Figure 1 shows possibility of Poisson counts with mean increasing with x.

```
plot(x,y)
```

Tidy up the inputted values - by storing them in a data frame and then removing the individual variables.

```
ex1.df<-data.frame(y,x)
ex1.df
```

```
   y x
1  1 0
2  0 0
3  1 0
4  3 0
5  0 0
6  2 0
7  0 0
8  1 0
9  2 0
10 1 0
11 2 1
12 3 1
```

```
13 2 1
14 1 1
15 0 1
16 2 1
17 1 1
18 4 1
19 1 1
20 3 2
21 4 2
22 2 2
23 0 2
24 5 2
25 1 2
26 4 2
27 2 2
28 3 2
29 3 2
30 1 2
```

```
rm(x,y)
```

Begin by fitting the model using the Poisson default log link.

```
ex1.glm1<-glm(y~x,poisson,data=ex1.df)
summary(ex1.glm1)
```

```

Call:
glm(formula = y ~ x, family = poisson, data = ex1.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2691  -0.5870  -0.1265   0.6162   1.4921

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1239     0.2608   0.475    0.635
x             0.4109     0.1705   2.409    0.016 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 35.057  on 29  degrees of freedom
Residual deviance: 28.941  on 28  degrees of freedom
AIC: 97.474

Number of Fisher Scoring iterations: 5
```

Then fit the model using the sqrt link.

```
ex1.glm2<-glm(y~x,poisson(link=sqrt),data=ex1.df)
summary(ex1.glm2)
```

```

Call:
glm(formula = y ~ x, family = poisson(link = sqrt), data = ex1.df)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2609  -0.6219  -0.1038   0.6384   1.4813

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0523     0.1451   7.251 4.15e-13 ***
x             0.2732     0.1092   2.502   0.0124 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 35.057  on 29  degrees of freedom
Residual deviance: 28.905  on 28  degrees of freedom
AIC: 97.438

Number of Fisher Scoring iterations: 4
```

As can be seen in the output above, the sqrt link has a slightly better deviance (NB no formal test between links – prefer one with lower deviance). For this link we shall try to see if the model can be simplified.

```
anova(ex1.glm2,test="Chi")
```

```
Analysis of Deviance Table

Model: poisson, link: sqrt

Response: y

Terms added sequentially (first to last)

     Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                    29     35.057
x     1   6.1517        28     28.905  0.01313 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(ex1.glm2)
```

From this we see the covariate is significant and thus we can not simplify the model. The residual plot in Figure 2 gives us no cause for concern.

5.1.2 Example 2: Poisson response with two factors

The following data records Poisson observations over two factors A and B at 3 and 2 levels respectively. Find a suitable model for the data.

Read in the data a row at a time (e.g. level 1 then level 2 of B). Create the data frame. Then plot a histogram of the counts ignoring the factors; it shows possibly Poisson count data in Figure 3.

```
y<-c(14,6,6,8,8,16,10,9,10,16,14,20,24,8,10,8,10,12,12,9,10,6,14,18,18,26,24,24)

A<-factor(rep(c(1,2,3,1,2,3),c(4,5,4,2,8,5)))
```

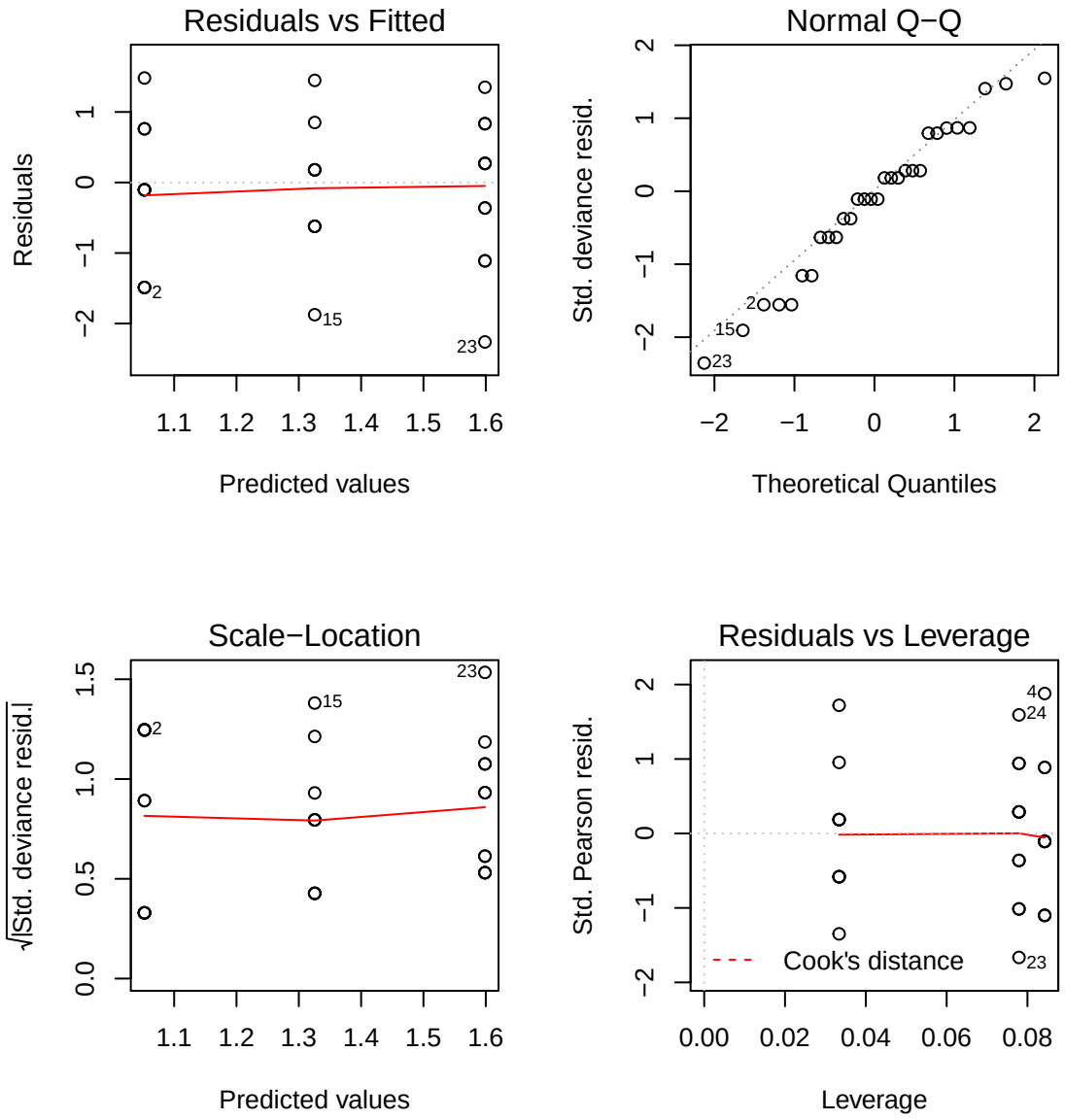

Figure 2: Diagnostic Plots: Poisson Counts

```
B<-factor(rep(c(1,2),c(13,15)))

ex2.df<-data.frame(y,A,B)
ex2.df
```

```
    y A B
1  14 1 1
2   6 1 1
3   6 1 1
4   8 1 1
5   8 2 1
6  16 2 1
7  10 2 1
8   9 2 1
9  10 2 1
10 16 3 1
11 14 3 1
12 20 3 1
13 24 3 1
14  8 1 2
15 10 1 2
16  8 2 2
17 10 2 2
18 12 2 2
19 12 2 2
20  9 2 2
21 10 2 2
22  6 2 2
23 14 2 2
24 18 3 2
25 18 3 2
26 26 3 2
27 24 3 2
28 24 3 2
```

```
rm(y,A,B)
attach(ex2.df)
```

```
hist(y)
```

Plotting boxplots in Figure 4 for each factor allows the further exploration of the data; suggesting that both levels of B may be the same and that levels 1 and 2 of A might be the same.

```
par(mfrow=c(1,2))
plot(y~A,xlab="A",ylab="Counts")
plot(y~B,xlab="B",ylab="Counts")
```

```
detach(ex2.df)
```

Fit the full model first; note deviance is not zero as there are replicates at various factor combinations. Then use the step command to knock out terms as per the code and output below.

```
ex2.glm.full<-glm(y~A*B,family=poisson,data=ex2.df)
ex2.glm.full
```

```

Call:  glm(formula = y ~ A * B, family = poisson, data = ex2.df)
```

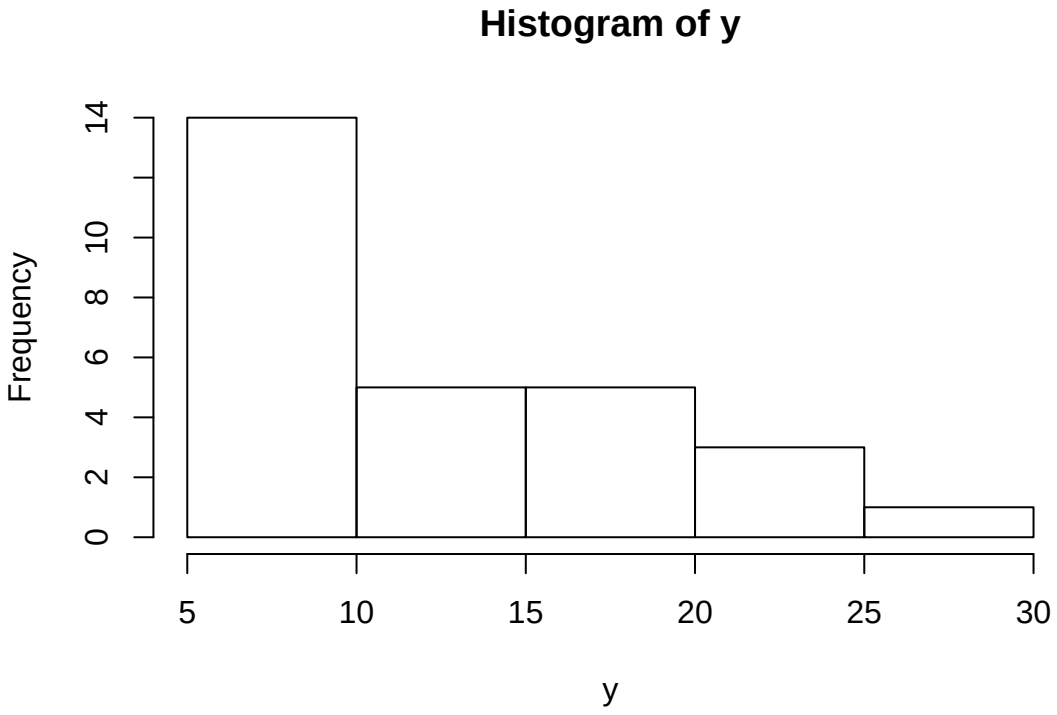

Figure 3: Histogram of y variable

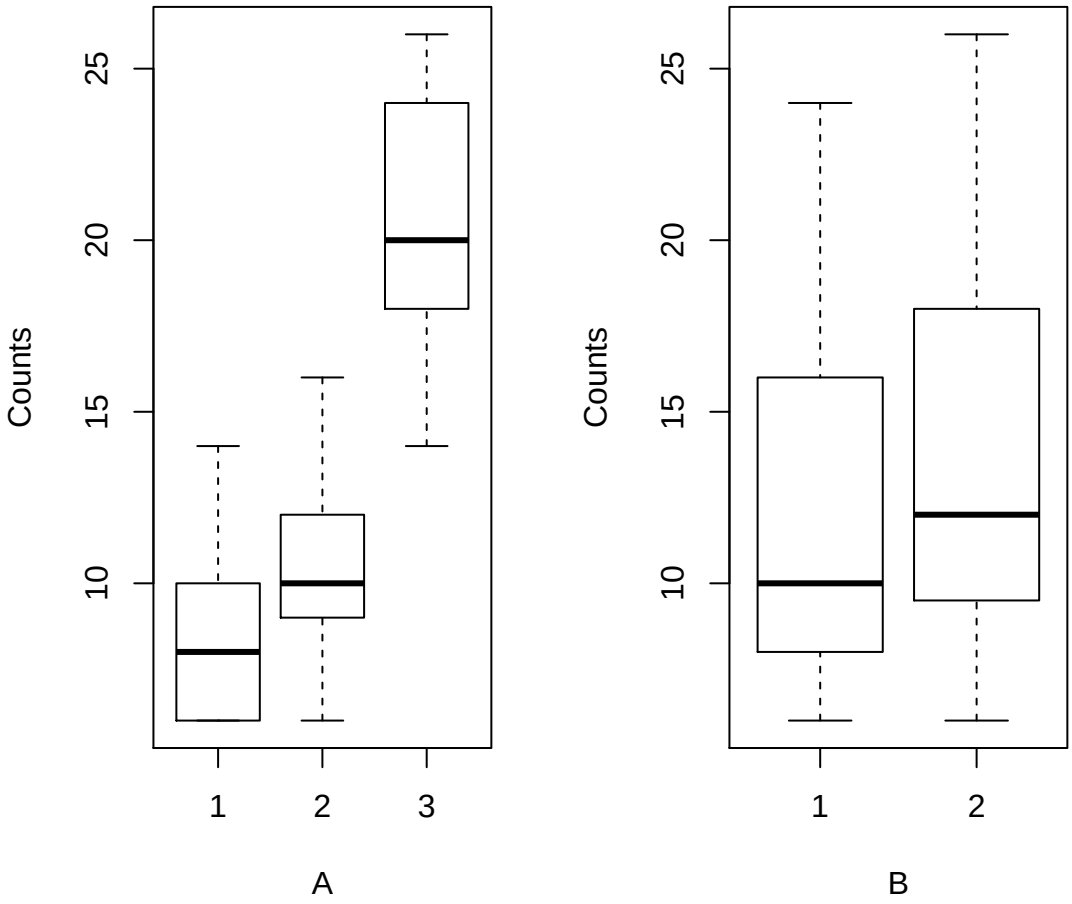

Figure 4: Boxplots of counts against Levels

```
Coefficients:
(Intercept)           A2           A3           B2        A2:B2
    2.14007      0.22079      0.77770      0.05716     -0.10300
      A3:B2
    0.11611

Degrees of Freedom: 27 Total (i.e. Null);  22 Residual
Null Deviance:      70.18
Residual Deviance: 18.58    AIC: 152.1
```

```
ex2.glm1<-step(ex2.glm.full)
```

```
Start:  AIC=152.07
y ~ A * B

       Df Deviance    AIC
- A:B   2   19.475 148.96
<none>      18.579 152.07

Step:  AIC=148.96
y ~ A + B

       Df Deviance    AIC
- B     1   20.026 147.51
<none>      19.475 148.96
- A     2   68.916 194.40

Step:  AIC=147.51
y ~ A

       Df Deviance    AIC
<none>      20.026 147.51
- A     2   70.184 193.67
```

The step algorithm leads to a model that has just factor A in it.

```
summary(ex2.glm1)
```

```

Call:
glm(formula = y ~ A, family = poisson, data = ex2.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.51214  -0.60109  -0.09749   0.57646   1.66174

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.1595     0.1387  15.572  < 2e-16 ***
A2            0.1734     0.1634   1.061    0.289
A3            0.8582     0.1571   5.465 4.64e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 70.184  on 27  degrees of freedom
Residual deviance: 20.026  on 25  degrees of freedom
AIC: 147.51

Number of Fisher Scoring iterations: 4
```

However, examination of the parameter values and t statistics suggest that level 2 of A may not be different to level 1 of A. Therefore create a new factor variable, ATRY, which has levels 1 and 2 of A the same. Add this to the data frame and remove unwanted variable. Fit new model with ATRY instead of A.

```
ATRY<-factor(rep(c(1,2,1,2),c(9,4,10,5)))
ex2.df<-data.frame(ex2.df,ATRY)
rm(ATRY)
ex2.glm2<-glm(y~ATRY,family=poisson,data=ex2.df)
```

We need to check that this model is better and if so whether it can be further simplified (unlikely from Wald statistics).

```
summary(ex2.glm2)
```

```
Call:
glm(formula = y ~ ATRY, family = poisson, data = ex2.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5121  -0.5908  -0.0158   0.7029   1.8166

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.28131    0.07332  31.113  < 2e-16 ***
ATRY2        0.73640    0.10398   7.082 1.42e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 70.184  on 27  degrees of freedom
Residual deviance: 21.181  on 26  degrees of freedom
AIC: 146.67

Number of Fisher Scoring iterations: 4
```

```
anova(ex2.glm2,ex2.glm1,test="Chi")
```

```
Analysis of Deviance Table

Model 1: y ~ ATRY
Model 2: y ~ A
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        26     21.181
2        25     20.026  1   1.1545   0.2826
```

```
drop1(ex2.glm2)
```

```
Single term deletions
```

```
Model:
y ~ ATRY
       Df Deviance    AIC
<none>      21.181 146.67
ATRY    1   70.184 193.67
```

It appears that modelling with the reduced levels of A is better and that the model can not be further simplified.

Residuals in Figure 5 give us no cause for concern.

```
par(mfrow=c(2,2))
plot(ex2.glm2)
```

5.1.3 Example 3: Poisson response with two covariates

The following data shows a Poisson response y together with two associated covariate values x1 and x2. Set up data frame.

```
y<-c(3,10,4,24,5,4,43,3,2,26,3,2)
x1<-c(14.154,31.817,2.203,22.646,8.585,2.160,53.517,6.234,2.858,34.124,2.484,6.619)
x2<-c(0.1132,4.5437,5.1989,15.0614,2.6844,11.2151,22.5853,0.7164,0.8493,16,5.6245,0.1385)
ex3.df<-data.frame(y,x1,x2)
ex3.df
```

```
    y     x1      x2
1   3 14.154  0.1132
2  10 31.817  4.5437
3   4  2.203  5.1989
4  24 22.646 15.0614
5   5  8.585  2.6844
6   4  2.160 11.2151
7  43 53.517 22.5853
8   3  6.234  0.7164
9   2  2.858  0.8493
10 26 34.124 16.0000
11  3  2.484  5.6245
12  2  6.619  0.1385
```

```
rm(y,x1,x2)
```

Then plot the response against each of the covariates; in both cases there appears to be an increase in y with x as illustrated in Figure 6.

```
attach(ex3.df)
par(mfrow=c(1,2))
plot(x1,y)
plot(x2,y)
```

Try fitting an additive model with the two covariates.

```
ex3.glm1<-glm(y~x1+x2,family=poisson,data=ex3.df)
summary(ex3.glm1)
```

```
Call:
glm(formula = y ~ x1 + x2, family = poisson, data = ex3.df)
```

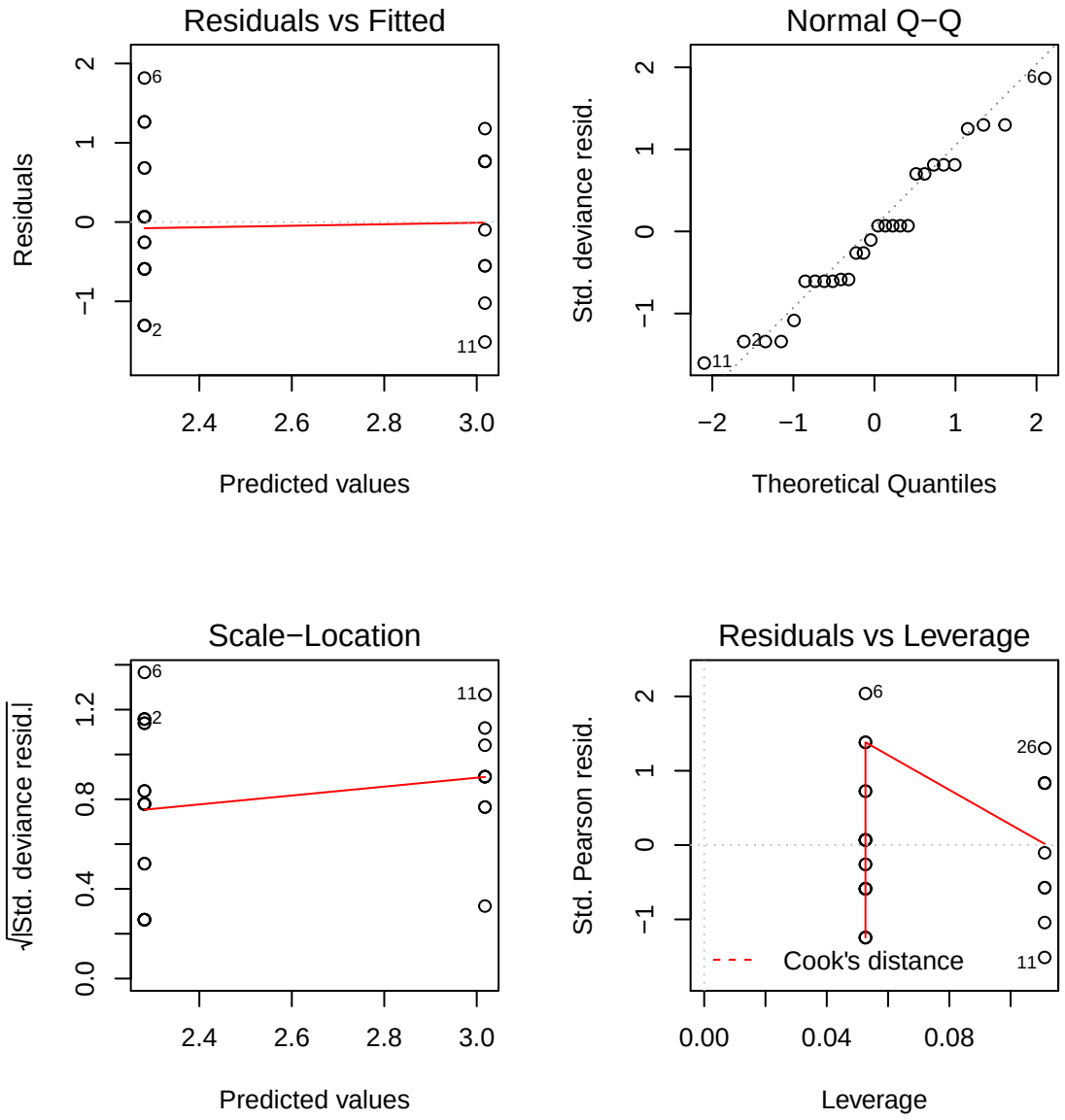

Figure 5: Diagnostic Plots

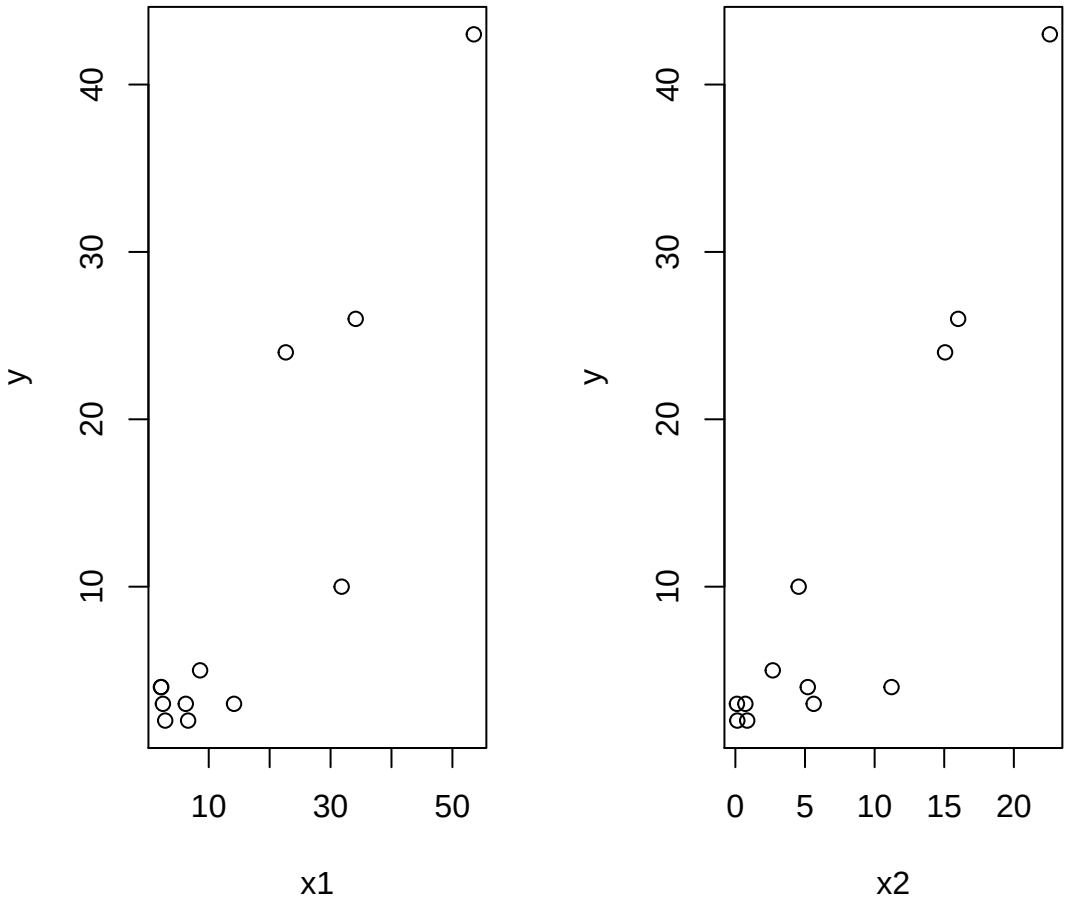

Figure 6: Plotting y against both x1 and x2

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3794  -0.7879  -0.3445   0.4833   2.0939

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.048961   0.183864   5.705 1.16e-08 ***
x1          0.019498   0.009257   2.106 0.035182 *
x2          0.081492   0.022829   3.570 0.000357 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.352  on 11  degrees of freedom
Residual deviance:  11.962  on  9  degrees of freedom
AIC: 62.063

Number of Fisher Scoring iterations: 4
```

The deviance 11.962 on 9 degrees of freedom is a respectable figure. Need to check diagnostic plots in Figure 7 to check quality of fit okay.

```
par(mfrow=c(2,2))
plot(ex3.glm1)
```

The plot of the residuals against fitted values shows a clear parabolic trend. The normal score plot is slightly curved. The 7th observation appears to have a lot of leverage. By plotting deviances against individual covariates in Figure 8 we aim to find out more.

```
par(mfrow=c(1,2))
plot(x1,ex3.glm1$res)
plot(x2,ex3.glm1$res)
```

The plot against x1 shows a divergent linear trend whilst the plot against x2 shows a fan. It would seem to be sensible to compress the x1 scale by taking logs which might sort out the plot against the deviance.

```
logx1 <- log(x1)
detach(ex3.df)
ex3.df<-data.frame(ex3.df,logx1)
rm(logx1)
ex3.df
```

```
    y     x1      x2     logx1
1   3 14.154  0.1132 2.6499973
2  10 31.817  4.5437 3.4600007
3   4  2.203  5.1989 0.7898201
4  24 22.646 15.0614 3.1199832
5   5  8.585  2.6844 2.1500165
6   4  2.160 11.2151 0.7701082
7  43 53.517 22.5853 3.9799994
8   3  6.234  0.7164 1.8300182
9   2  2.858  0.8493 1.0501221
10 26 34.124 16.0000 3.5300009
11  3  2.484  5.6245 0.9098702
12  2  6.619  0.1385 1.8899443
```

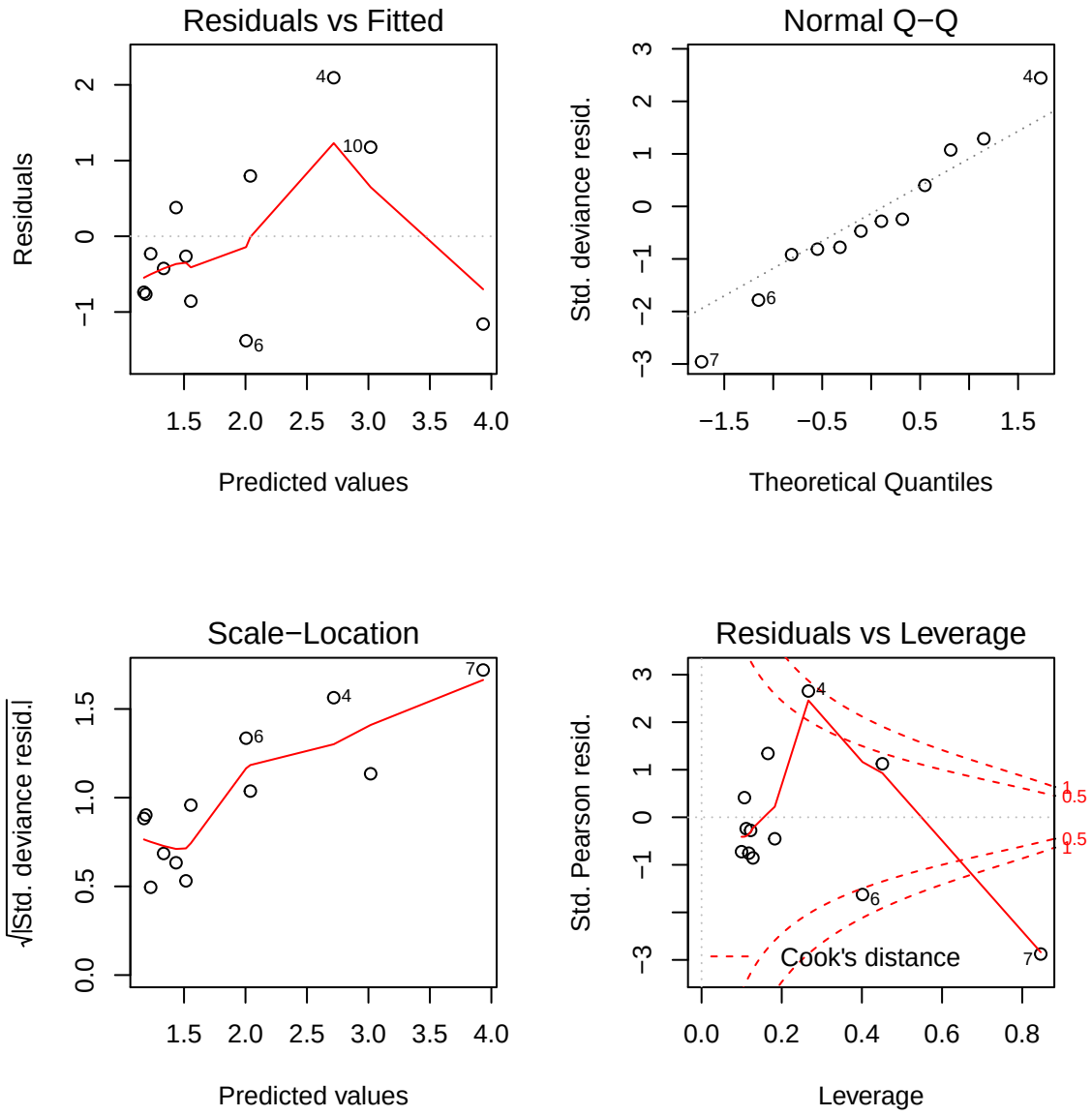

Figure 7: Diagnostic Plots

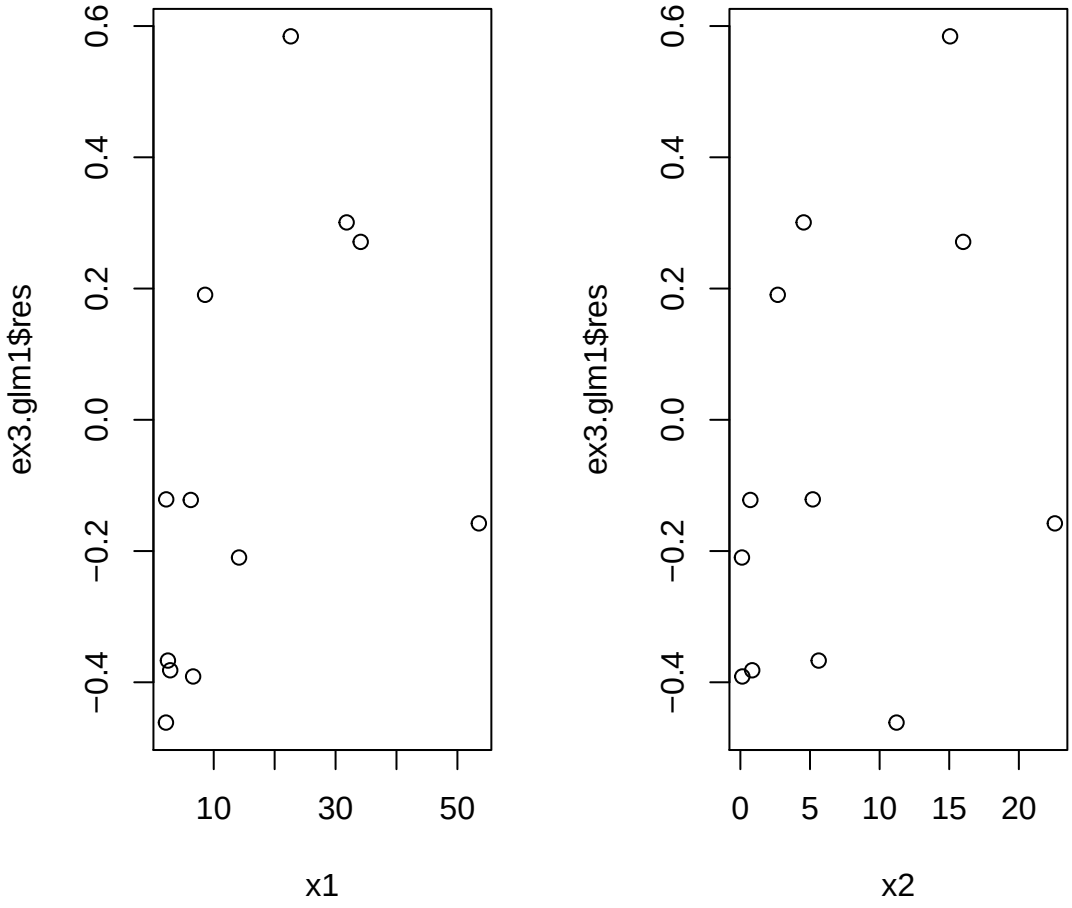

Figure 8: Residual Plots

Fit the new model with the new variable log(x1).

```
ex3.glm2<-glm(y~logx1+x2, family = poisson, data = ex3.df)
ex3.glm2
```

```
Call:  glm(formula = y ~ logx1 + x2, family = poisson, data = ex3.df)

Coefficients:
(Intercept)        logx1           x2
    0.34705      0.44301      0.07829

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:      142.4
Residual Deviance: 4.348    AIC: 54.45
```

```
summary(ex3.glm2)
```

```
Call:
glm(formula = y ~ logx1 + x2, family = poisson, data = ex3.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8047  -0.4720  -0.1825   0.2832   1.2641

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.34705    0.30770   1.128  0.25937
logx1        0.44301    0.13495   3.283  0.00103 **
x2           0.07829    0.01681   4.657 3.21e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.3521  on 11  degrees of freedom
Residual deviance:   4.3478  on  9  degrees of freedom
AIC: 54.448

Number of Fisher Scoring iterations: 4
```

The new deviance is an improvement, but what about the residuals in Figure 9?

```
par(mfrow=c(3,2))
attach(ex3.df)
plot(ex3.glm2)
plot(x1,ex3.glm2$res)
plot(x2,ex3.glm2$res)
```

The normal score plot is reasonably linear apart from a strange tail. The plot of the residuals against fitted values is better, but possible a detectable arc? The plot of the deviance against x1 is now sorted. The plot of the deviance against x2 now seems like a divergent arc. This might indicate a quadratic or square root of x2 is needed; we shall create the new variables sqx2 and sqrtx2 and try models with them instead of x2.

```
sqx2<-x2*x2
sqrtx2<-sqrt(x2)
```

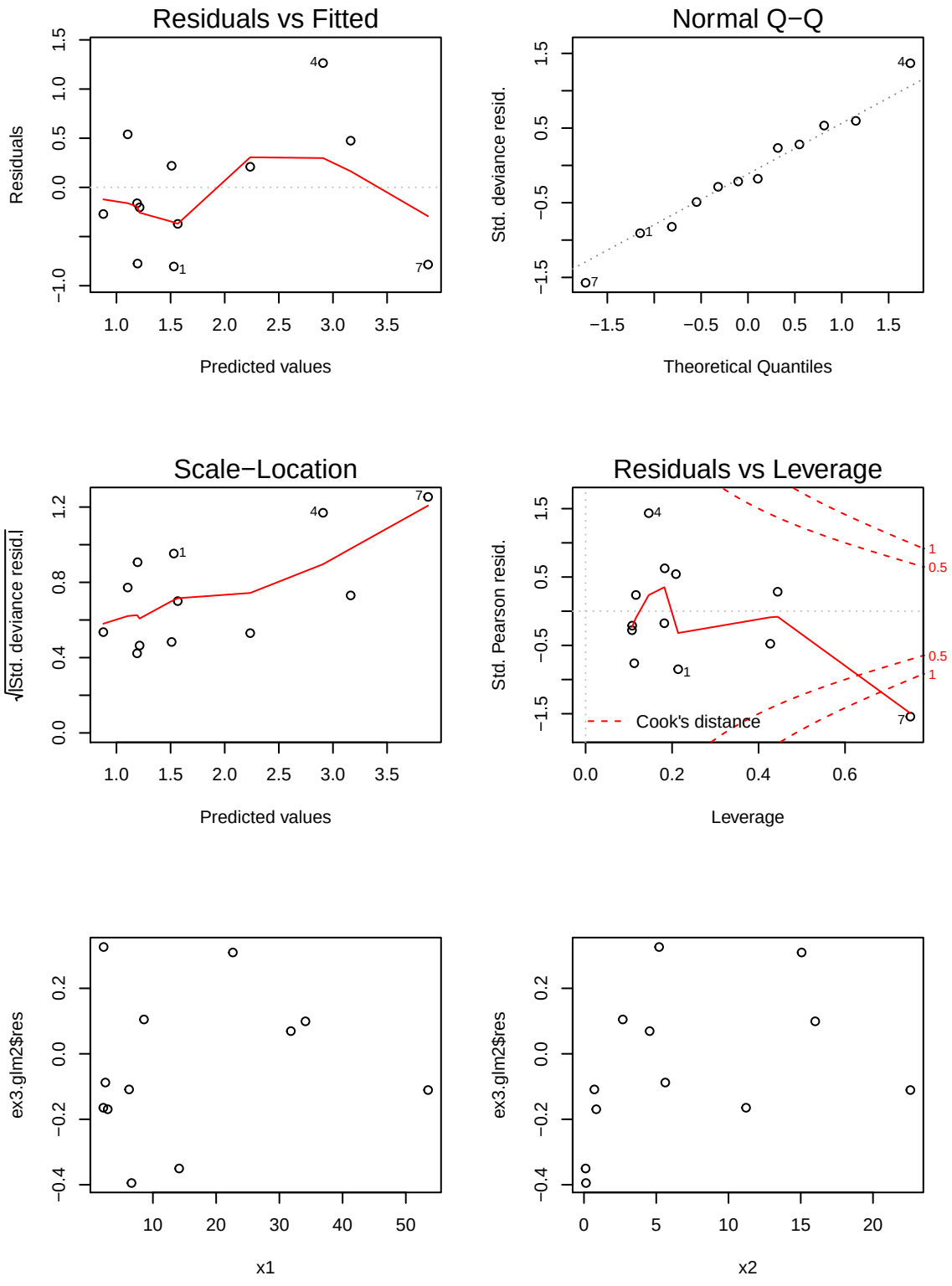

Figure 9: Diagnostic Plots and Residual Plots: including logarithm term

```
detach(ex3.df)
ex3.df<-data.frame(ex3.df,sqx2,sqrtx2)
rm(sqx2, sqrtx2)
ex3.df
```

```
    y     x1      x2     logx1         sqx2    sqrtx2
1   3 14.154  0.1132 2.6499973   0.01281424 0.3364521
2  10 31.817  4.5437 3.4600007  20.64520969 2.1315956
3   4  2.203  5.1989 0.7898201  27.02856121 2.2801096
4  24 22.646 15.0614 3.1199832 226.84576996 3.8809020
5   5  8.585  2.6844 2.1500165   7.20600336 1.6384139
6   4  2.160 11.2151 0.7701082 125.77846801 3.3488953
7  43 53.517 22.5853 3.9799994 510.09577609 4.7523994
8   3  6.234  0.7164 1.8300182   0.51322896 0.8464042
9   2  2.858  0.8493 1.0501221   0.72131049 0.9215747
10 26 34.124 16.0000 3.5300009 256.00000000 4.0000000
11  3  2.484  5.6245 0.9098702  31.63500025 2.3716028
12  2  6.619  0.1385 1.8899443   0.01918225 0.3721559
```

Trying the new models:

```
ex3.glm3<-glm(y~logx1+sqx2, family = poisson, data = ex3.df)
ex3.glm3
```

```
Call:  glm(formula = y ~ logx1 + sqx2, family = poisson, data = ex3.df)

Coefficients:
(Intercept)        logx1         sqx2
   0.501070     0.527566     0.002563

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:      142.4
Residual Deviance: 12.32    AIC: 62.42
```

```
summary(ex3.glm3)
```

```
Call:
glm(formula = y ~ logx1 + sqx2, family = poisson, data = ex3.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5991  -0.7583  -0.1730   0.4160   2.0488

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.5010699  0.3483684   1.438 0.150339
logx1       0.5275662  0.1427855   3.695 0.000220 ***
sqx2        0.0025633  0.0006604   3.882 0.000104 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.35  on 11  degrees of freedom
```

```
Residual deviance:  12.32  on  9  degrees of freedom
AIC: 62.421

Number of Fisher Scoring iterations: 4
```

```
ex3.glm4<-glm(y~logx1+sqrtx2, family = poisson, data = ex3.df)
ex3.glm4
```

```

Call:  glm(formula = y ~ logx1 + sqrtx2, family = poisson, data = ex3.df)

Coefficients:
(Intercept)        logx1       sqrtx2
    -0.2341       0.4747       0.4535

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:      142.4
Residual Deviance: 1.502    AIC: 51.6
```

```
summary(ex3.glm4)
```

```

Call:
glm(formula = y ~ logx1 + sqrtx2, family = poisson, data = ex3.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.55196  -0.25190  -0.06186   0.14680   0.81518

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.23408    0.30879  -0.758 0.448428
logx1        0.47471    0.12242   3.878 0.000105 ***
sqrtx2       0.45347    0.09362   4.844 1.27e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.352  on 11  degrees of freedom
Residual deviance:   1.502  on  9  degrees of freedom
AIC: 51.602

Number of Fisher Scoring iterations: 4
```

The deviance using sqx2 is 12.32 on 9 d.f. which is not worth following up after 4.348 with the previous model. The deviance for the model with sqrtx2 looks promising, but what about the residual plots in Figure 10?

```
par(mfrow=c(3,2))
attach(ex3.df)
plot(ex3.glm4)
plot(x1,ex3.glm4$res)
plot(x2,ex3.glm4$res)
```

The normal score plot is slightly concave. The plot of the residuals against fitted values is okay, but possibly a fan? The plot of residuals against x2 is better but still shows a divergent fan. The plot of residuals against x1 is okay. The plot of the square root of the absolute deviance residuals against predicted values shows a

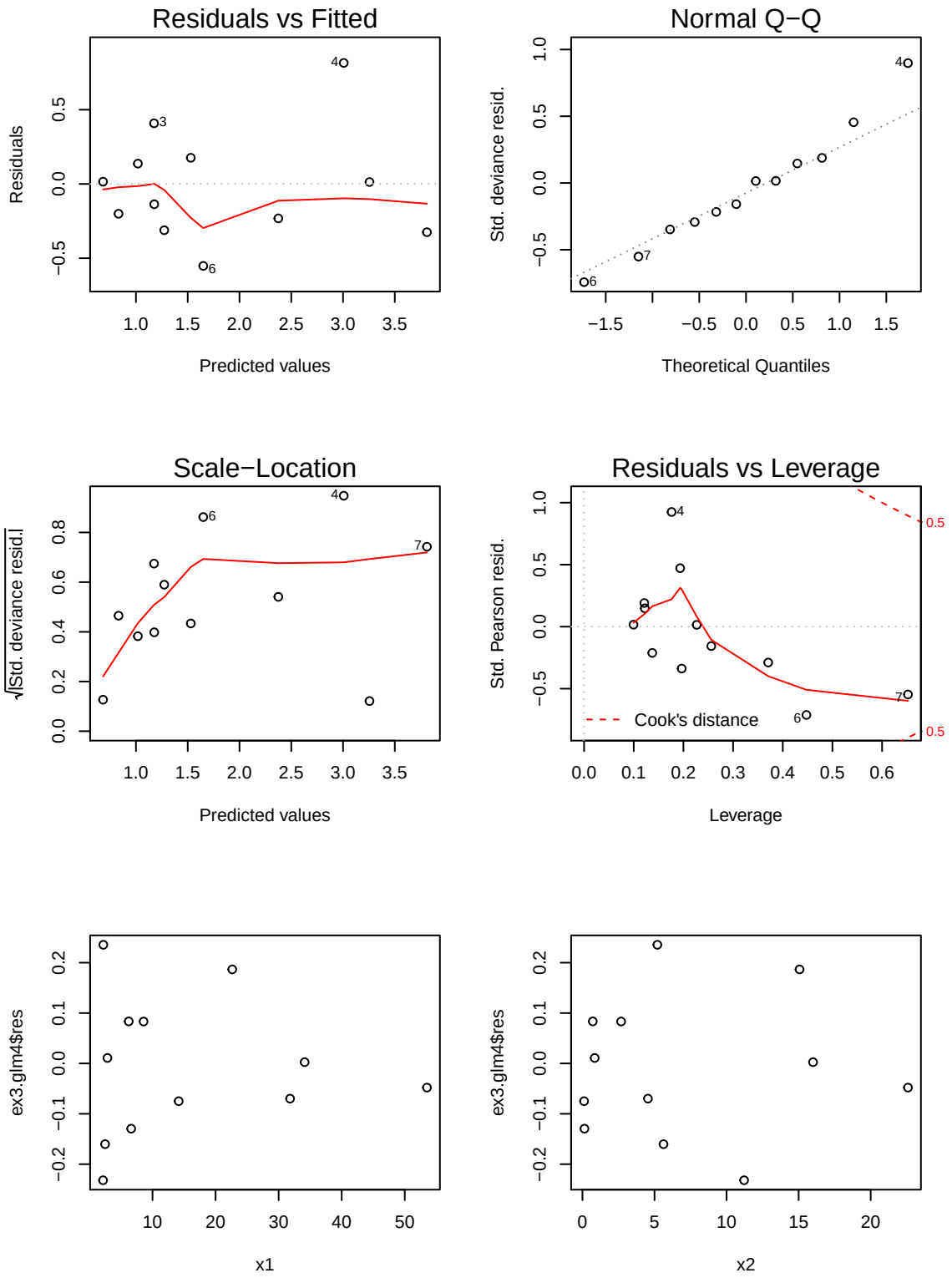

Figure 10: Diagnostic Plots and Residual Plots: including square root of x2

clear divergent arc. The leverage of the 7th data point is not as disturbing any more.

Since x1 is okay we shall concentrate on x2. Taking the square root of x2 has helped, so we shall try a new variable that is the fourth root.

```
fourthx2<-x2^0.25
detach(ex3.df)
ex3.df<-data.frame(ex3.df,fourthx2)
rm(fourthx2)
ex3.df
```

```
    y     x1      x2     logx1         sqx2    sqrtx2  fourthx2
1   3 14.154  0.1132 2.6499973   0.01281424 0.3364521 0.5800449
2  10 31.817  4.5437 3.4600007  20.64520969 2.1315956 1.4599985
3   4  2.203  5.1989 0.7898201  27.02856121 2.2801096 1.5100032
4  24 22.646 15.0614 3.1199832 226.84576996 3.8809020 1.9700005
5   5  8.585  2.6844 2.1500165   7.20600336 1.6384139 1.2800054
6   4  2.160 11.2151 0.7701082 125.77846801 3.3488953 1.8299987
7  43 53.517 22.5853 3.9799994 510.09577609 4.7523994 2.1799999
8   3  6.234  0.7164 1.8300182   0.51322896 0.8464042 0.9200023
9   2  2.858  0.8493 1.0501221   0.72131049 0.9215747 0.9599868
10 26 34.124 16.0000 3.5300009 256.00000000 4.0000000 2.0000000
11  3  2.484  5.6245 0.9098702  31.63500025 2.3716028 1.5400009
12  2  6.619  0.1385 1.8899443   0.01918225 0.3721559 0.6100458
```

```
ex3.glm5<-glm(y~logx1+fourthx2, family = poisson, data = ex3.df)
ex3.glm5
```

```
Call:  glm(formula = y ~ logx1 + fourthx2, family = poisson, data = ex3.df)

Coefficients:
(Intercept)        logx1     fourthx2
    -1.2133       0.5224       1.3280

Degrees of Freedom: 11 Total (i.e. Null);  9 Residual
Null Deviance:      142.4
Residual Deviance: 1.737    AIC: 51.84
```

```
summary(ex3.glm5)
```

```
Call:
glm(formula = y ~ logx1 + fourthx2, family = poisson, data = ex3.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.75819  -0.20401   0.05865   0.23673   0.69476

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.2133     0.4220  -2.875  0.00404 **
logx1         0.5224     0.1161   4.500 6.79e-06 ***
fourthx2      1.3280     0.2827   4.698 2.62e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.3521  on 11  degrees of freedom
Residual deviance:   1.7371  on  9  degrees of freedom
AIC: 51.837

Number of Fisher Scoring iterations: 4
```

The new deviance is still good, so then examine the diagnostic plots in Figure 11.

```
attach(ex3.df)
par(mfrow=c(3,2))
plot(ex3.glm5)
plot(x1, ex3.glm5$res)
plot(x2, ex3.glm5$res)
```

These now look okay. Thus the final model will be Poisson response with means given by:

$$\mu_i = \exp\left(-1.2133 + 0.5224\log(x1) + 1.3280x2^{0.25}\right)$$

```
summary(ex3.glm5)
```

```
Call:
glm(formula = y ~ logx1 + fourthx2, family = poisson, data = ex3.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.75819  -0.20401   0.05865   0.23673   0.69476

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.2133     0.4220  -2.875  0.00404 **
logx1         0.5224     0.1161   4.500 6.79e-06 ***
fourthx2      1.3280     0.2827   4.698 2.62e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 142.3521  on 11  degrees of freedom
Residual deviance:   1.7371  on  9  degrees of freedom
AIC: 51.837

Number of Fisher Scoring iterations: 4
```

5.2 Offsets

An offset is a term to be added to a linear predictor, such as in a generalised linear model, with known coefficient 1 rather than an estimated coefficient. A common use of an offset is when dealing with poisson counts from populations of various sizes. It is necessary to acknowledge that the observed counts are partly related to the population sizes as well as any explanatory variables.

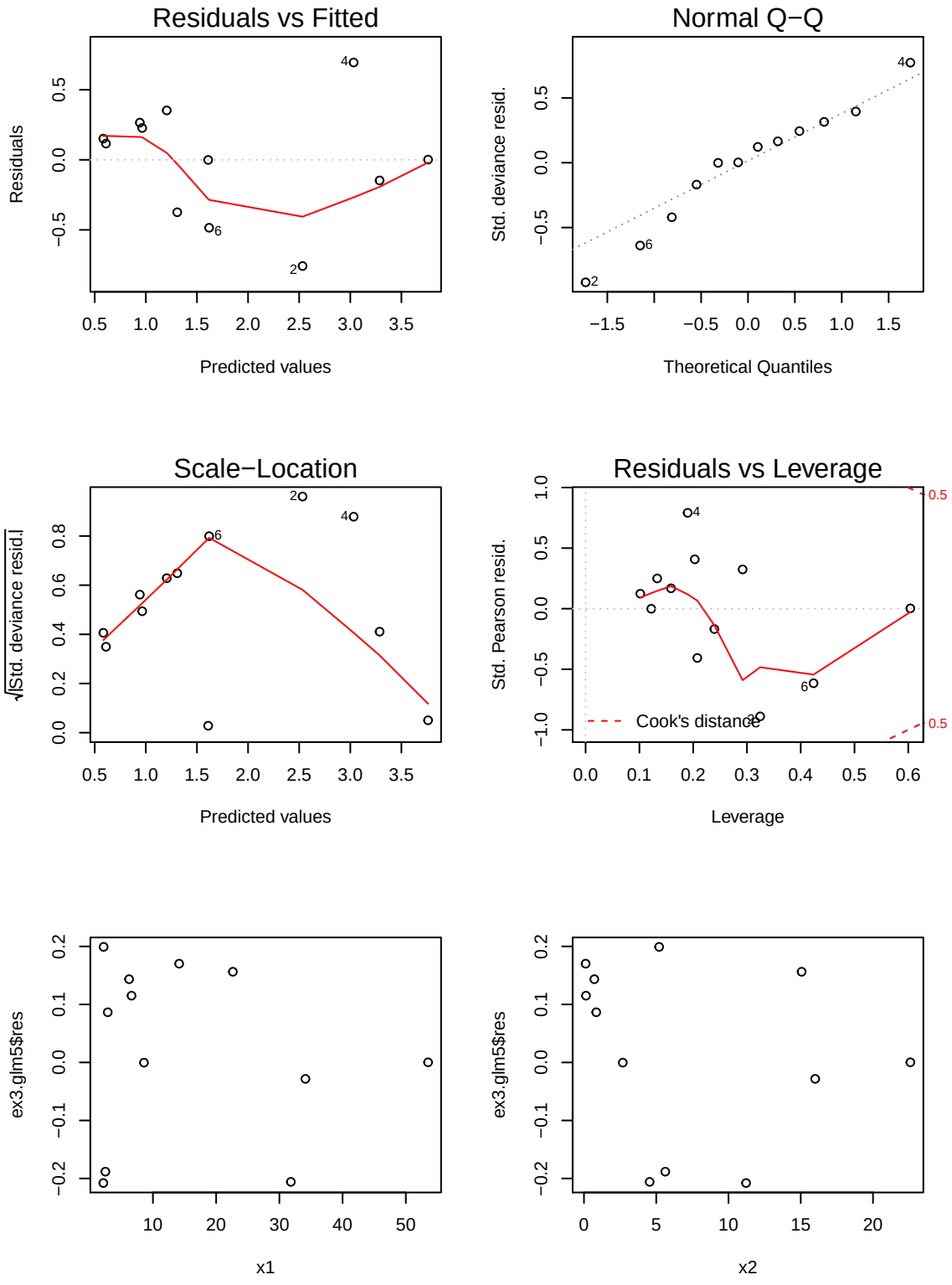

Figure 11: Diagnostic Plots and Residual Plots: $\log(x1)$ and $\sqrt[4]{x2}$

5.2.1 Example 4: Insurance data

For example, the data given in the example dataset within R *''Insurance"* consist of the numbers of policyholders of an insurance company who were exposed to risk, and the numbers of car insurance claims made by those policyholders in the third quarter of 1973.

This data frame contains the following variables.

- District: district of policyholder, A/B/C/D, D is major cities.
- Group: group of car (1 to 4), $<$1 litre, 1-1.5 litre, 1.5-2 litre, $>$2 litre.
- Age: age of driver in 4 ordered groups, $<$25, 25-29, 30-35, $>$35.
- Holders: numbers of policyholders.
- Claims: numbers of claims.

```
library(MASS)
data("Insurance")
Insurance
```

```
   District  Group   Age Holders Claims
1         1    <1l   <25     197     38
2         1    <1l 25-29     264     35
3         1    <1l 30-35     246     20
4         1    <1l   >35    1680    156
5         1 1-1.5l   <25     284     63
6         1 1-1.5l 25-29     536     84
7         1 1-1.5l 30-35     696     89
8         1 1-1.5l   >35    3582    400
9         1 1.5-2l   <25     133     19
10        1 1.5-2l 25-29     286     52
11        1 1.5-2l 30-35     355     74
12        1 1.5-2l   >35    1640    233
13        1    >2l   <25      24      4
14        1    >2l 25-29      71     18
15        1    >2l 30-35      99     19
16        1    >2l   >35     452     77
17        2    <1l   <25      85     22
18        2    <1l 25-29     139     19
19        2    <1l 30-35     151     22
20        2    <1l   >35     931     87
21        2 1-1.5l   <25     149     25
22        2 1-1.5l 25-29     313     51
23        2 1-1.5l 30-35     419     49
24        2 1-1.5l   >35    2443    290
25        2 1.5-2l   <25      66     14
26        2 1.5-2l 25-29     175     46
27        2 1.5-2l 30-35     221     39
28        2 1.5-2l   >35    1110    143
29        2    >2l   <25       9      4
30        2    >2l 25-29      48     15
31        2    >2l 30-35      72     12
32        2    >2l   >35     322     53
33        3    <1l   <25      35      5
34        3    <1l 25-29      73     11
35        3    <1l 30-35      89     10
36        3    <1l   >35     648     67
37        3 1-1.5l   <25      53     10
```

```
38        3 1-1.5l 25-29     155     24
39        3 1-1.5l 30-35     240     37
40        3 1-1.5l   >35    1635    187
41        3 1.5-2l   <25      24      8
42        3 1.5-2l 25-29      78     19
43        3 1.5-2l 30-35     121     24
44        3 1.5-2l   >35     692    101
45        3    >2l   <25       7      3
46        3    >2l 25-29      29      2
47        3    >2l 30-35      43      8
48        3    >2l   >35     245     37
49        4    <1l   <25      20      2
50        4    <1l 25-29      33      5
51        4    <1l 30-35      40      4
52        4    <1l   >35     316     36
53        4 1-1.5l   <25      31      7
54        4 1-1.5l 25-29      81     10
55        4 1-1.5l 30-35     122     22
56        4 1-1.5l   >35     724    102
57        4 1.5-2l   <25      18      5
58        4 1.5-2l 25-29      39      7
59        4 1.5-2l 30-35      68     16
60        4 1.5-2l   >35     344     63
61        4    >2l   <25       3      0
62        4    >2l 25-29      16      6
63        4    >2l 30-35      25      8
64        4    >2l   >35     114     33
```

The three explanatory variables (District/Group/Car) form a three way table with $4^3 = 64$ cells each containing a number of claims. If there was one policyholder in each cell was the same then we could (say) fit an additive model:

```
mod<-glm(Claims~District+Group+Age,family=poisson,data=Insurance)
summary(mod)
```

```
Call:
glm(formula = Claims ~ District + Group + Age, family = poisson,
    data = Insurance)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5967  -0.9877  -0.1092   0.5180   4.3268

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.92306    0.03355 116.924  < 2e-16 ***
District2   -0.43822    0.04297 -10.198  < 2e-16 ***
District3   -0.91521    0.05032 -18.187  < 2e-16 ***
District4   -1.44367    0.06158 -23.445  < 2e-16 ***
Group.L     -0.51133    0.04932 -10.368  < 2e-16 ***
Group.Q     -1.02479    0.04198 -24.413  < 2e-16 ***
Group.C      0.21633    0.03304   6.547 5.87e-11 ***
Age.L        1.50084    0.04916  30.527  < 2e-16 ***
Age.Q        0.47465    0.04882   9.722  < 2e-16 ***
```

```
Age.C        0.41495    0.04847   8.560  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4236.68  on 63  degrees of freedom
Residual deviance:  121.31  on 54  degrees of freedom
AIC: 458.63

Number of Fisher Scoring iterations: 4
```

This would give the expected claim in a cell to be:

$$Expected\ claim = \exp(\beta_0 + \beta^{District} + \beta^{Group} + \beta^{Age})$$

However there are a number of policyholders in each cell so we would require:

$$Expected\ claim = Holders \times \exp(\beta_0 + \beta^{District} + \beta^{Group} + \beta^{Age})$$

Which can be re-written as:

$$Expected\ claim = \exp\left(\beta_0 + \beta^{District} + \beta^{Group} + \beta^{Age} + \log(Holders)\right)$$

Note that the linear predictor now includes log(Holders) which is not a variable with a parameter to estimate! The appropriate main-effects fit as Poisson GLM with offset is:

```
moda<-glm(Claims~District+Group+Age+ offset(log(Holders)),
        family=poisson,data=Insurance)
 summary(moda)
```

```
Call:
glm(formula = Claims ~ District + Group + Age + offset(log(Holders)),
    family = poisson, data = Insurance)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.46558  -0.50802  -0.03198   0.55555   1.94026

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.810508   0.032972 -54.910  < 2e-16 ***
District2    0.025868   0.043016   0.601 0.547597
District3    0.038524   0.050512   0.763 0.445657
District4    0.234205   0.061673   3.798 0.000146 ***
Group.L      0.429708   0.049459   8.688  < 2e-16 ***
Group.Q      0.004632   0.041988   0.110 0.912150
Group.C     -0.029294   0.033069  -0.886 0.375696
Age.L       -0.394432   0.049404  -7.984 1.42e-15 ***
Age.Q       -0.000355   0.048918  -0.007 0.994210
Age.C       -0.016737   0.048478  -0.345 0.729910
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 236.26  on 63  degrees of freedom
Residual deviance:  51.42  on 54  degrees of freedom
AIC: 388.74

Number of Fisher Scoring iterations: 4
```

Now consider an analysis of this data, beginning with a saturated model and using model simplification approaches to reduce the complexity of the final model.

```
mod1<-glm(Claims~District*Group*Age+offset(log(Holders)),
        family=poisson,data=Insurance)
summary(mod1)
```

```
Call:
glm(formula = Claims ~ District * Group * Age + offset(log(Holders)),
    family = poisson, data = Insurance)

Deviance Residuals:
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[24]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[47]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.870e+00  4.896e-02 -38.188  < 2e-16 ***
District2             1.350e-01  7.387e-02   1.827  0.06763 .
District3             5.649e-02  9.699e-02   0.582  0.56024
District4            -1.182e+00  2.640e+03   0.000  0.99964
Group.L               3.545e-01  1.182e-01   2.998  0.00271 **
Group.Q              -4.487e-02  9.792e-02  -0.458  0.64681
Group.C               3.656e-02  7.209e-02   0.507  0.61206
Age.L                -2.816e-01  1.058e-01  -2.662  0.00778 **
Age.Q                -5.541e-02  9.792e-02  -0.566  0.57146
Age.C                 6.156e-02  8.935e-02   0.689  0.49088
District2:Group.L     5.975e-02  1.754e-01   0.341  0.73342
District3:Group.L    -6.837e-02  2.332e-01  -0.293  0.76934
District4:Group.L    -3.327e+00  7.085e+03   0.000  0.99963
District2:Group.Q     2.122e-01  1.477e-01   1.437  0.15085
District3:Group.Q    -1.755e-01  1.940e-01  -0.905  0.36554
District4:Group.Q    -2.709e+00  5.281e+03  -0.001  0.99959
District2:Group.C    -1.232e-01  1.135e-01  -1.086  0.27761
District3:Group.C    -2.240e-01  1.445e-01  -1.550  0.12107
District4:Group.C    -1.234e+00  2.362e+03  -0.001  0.99958
District2:Age.L      -2.699e-01  1.576e-01  -1.713  0.08677 .
District3:Age.L      -1.427e-01  1.928e-01  -0.740  0.45945
District4:Age.L       3.875e+00  7.085e+03   0.001  0.99956
District2:Age.Q       6.156e-02  1.477e-01   0.417  0.67691
District3:Age.Q       2.289e-01  1.940e-01   1.180  0.23801
District4:Age.Q      -2.727e+00  5.281e+03  -0.001  0.99959
District2:Age.C      -4.086e-03  1.372e-01  -0.030  0.97624
District3:Age.C      -2.941e-01  1.951e-01  -1.507  0.13169
District4:Age.C       1.122e+00  2.362e+03   0.000  0.99962
Group.L:Age.L         4.896e-01  2.567e-01   1.907  0.05655 .
Group.Q:Age.L        -8.415e-02  2.116e-01  -0.398  0.69087
```

```
Group.C:Age.L           -2.339e-01  1.537e-01  -1.522  0.12802
Group.L:Age.Q           -4.445e-01  2.365e-01  -1.880  0.06012 .
Group.Q:Age.Q            9.545e-02  1.958e-01   0.487  0.62600
Group.C:Age.Q            1.636e-01  1.442e-01   1.134  0.25662
Group.L:Age.C            1.192e-03  2.143e-01   0.006  0.99556
Group.Q:Age.C            2.331e-01  1.787e-01   1.304  0.19215
Group.C:Age.C            5.697e-02  1.340e-01   0.425  0.67072
District2:Group.L:Age.L -6.082e-01  3.748e-01  -1.623  0.10459
District3:Group.L:Age.L -6.794e-01  4.545e-01  -1.495  0.13496
District4:Group.L:Age.L  9.466e+00  1.901e+04   0.000  0.99960
District2:Group.Q:Age.L -2.889e-01  3.152e-01  -0.917  0.35929
District3:Group.Q:Age.L  1.724e-01  3.857e-01   0.447  0.65495
District4:Group.Q:Age.L  7.722e+00  1.417e+04   0.001  0.99957
District2:Group.C:Age.L  2.818e-01  2.413e-01   1.168  0.24281
District3:Group.C:Age.L  3.668e-01  3.015e-01   1.216  0.22383
District4:Group.C:Age.L  3.543e+00  6.337e+03   0.001  0.99955
District2:Group.L:Age.Q  4.292e-01  3.509e-01   1.223  0.22121
District3:Group.L:Age.Q  1.045e+00  4.663e-01   2.241  0.02501 *
District4:Group.L:Age.Q -7.033e+00  1.417e+04   0.000  0.99960
District2:Group.Q:Age.Q  1.593e-01  2.955e-01   0.539  0.58984
District3:Group.Q:Age.Q  2.985e-01  3.880e-01   0.769  0.44165
District4:Group.Q:Age.Q -5.933e+00  1.056e+04  -0.001  0.99955
District2:Group.C:Age.Q  4.501e-02  2.269e-01   0.198  0.84279
District3:Group.C:Age.Q -3.545e-03  2.890e-01  -0.012  0.99022
District4:Group.C:Age.Q -2.592e+00  4.723e+03  -0.001  0.99956
District2:Group.L:Age.C  3.166e-01  3.252e-01   0.974  0.33030
District3:Group.L:Age.C -6.739e-01  4.779e-01  -1.410  0.15847
District4:Group.L:Age.C  3.207e+00  6.337e+03   0.001  0.99960
District2:Group.Q:Age.C -4.190e-01  2.744e-01  -1.527  0.12668
District3:Group.Q:Age.C -5.423e-01  3.902e-01  -1.390  0.16457
District4:Group.Q:Age.C  2.769e+00  4.723e+03   0.001  0.99953
District2:Group.C:Age.C  4.101e-02  2.116e-01   0.194  0.84635
District3:Group.C:Age.C -3.275e-01  2.760e-01  -1.187  0.23527
District4:Group.C:Age.C  9.491e-01  2.112e+03   0.000  0.99964
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2.3626e+02  on 63  degrees of freedom
Residual deviance: 4.1222e-10  on  0  degrees of freedom
AIC: 445.32

Number of Fisher Scoring iterations: 20
```

```
mod2<-step(mod1,direction="both")
```

```
Start:  AIC=445.32
Claims ~ District * Group * Age + offset(log(Holders))

                     Df Deviance    AIC
- District:Group:Age 27    27.29 418.61
<none>                      0.00 445.32

Step:  AIC=418.61
```

```
Claims ~ District + Group + Age + District:Group + District:Age +
    Group:Age + offset(log(Holders))

                     Df Deviance    AIC
- District:Age        9   33.527 406.85
- District:Group      9   34.457 407.78
- Group:Age           9   37.685 411.01
<none>                    27.290 418.61
+ District:Group:Age 27    0.000 445.32

Step:  AIC=406.85
Claims ~ District + Group + Age + District:Group + Group:Age +
    offset(log(Holders))

                 Df Deviance    AIC
- District:Group  9   40.907 396.23
- Group:Age       9   44.132 399.45
<none>                33.527 406.85
+ District:Age    9   27.290 418.61

Step:  AIC=396.23
Claims ~ District + Group + Age + Group:Age + offset(log(Holders))

                 Df Deviance    AIC
- Group:Age       9   51.420 388.74
<none>                40.907 396.23
- District        3   54.850 404.17
+ District:Group  9   33.527 406.85
+ District:Age    9   34.457 407.78

Step:  AIC=388.74
Claims ~ District + Group + Age + offset(log(Holders))

                 Df Deviance    AIC
<none>                51.420 388.74
+ Group:Age       9   40.907 396.23
- District        3   65.291 396.61
+ District:Group  9   44.132 399.45
+ District:Age    9   44.859 400.18
- Age             3  136.290 467.61
- Group           3  140.087 471.41
```

```
summary(mod2)
```

```
Call:
glm(formula = Claims ~ District + Group + Age + offset(log(Holders)),
    family = poisson, data = Insurance)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.46558  -0.50802  -0.03198   0.55555   1.94026

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) -1.810508   0.032972 -54.910  < 2e-16 ***
District2    0.025868   0.043016   0.601 0.547597
District3    0.038524   0.050512   0.763 0.445657
District4    0.234205   0.061673   3.798 0.000146 ***
Group.L      0.429708   0.049459   8.688  < 2e-16 ***
Group.Q      0.004632   0.041988   0.110 0.912150
Group.C     -0.029294   0.033069  -0.886 0.375696
Age.L       -0.394432   0.049404  -7.984 1.42e-15 ***
Age.Q       -0.000355   0.048918  -0.007 0.994210
Age.C       -0.016737   0.048478  -0.345 0.729910
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 236.26  on 63  degrees of freedom
Residual deviance:  51.42  on 54  degrees of freedom
AIC: 388.74

Number of Fisher Scoring iterations: 4
```

Making a prediction

```
mod2.pred<-predict(mod2,se.fit=T,type="response")
mod2.pred
```

```
$fit
         1          2          3          4          5          6
 31.863585  35.275867  28.180802 158.878292  53.977716  84.160115
         7          8          9         10         11         12
 93.690431 398.060076  31.862179  56.602450  60.234044 229.717751
        13         14         15         16         17         18
  6.819092  16.665525  19.922338  75.089737  14.108529  19.060004
        19         20         21         22         23         24
 17.751277  90.352333  29.061421  50.433636  57.880788 278.599876
        25         26         27         28         29         30
 16.225653  35.541983  38.480469 159.554148   2.624171  11.562089
        31         32         33         34         35         36
 14.868666  54.894955   5.883384  10.137418  10.595927  63.688499
        37         38         39         40         41         42
 10.468941  25.293210  33.575924 188.830232   5.975384  16.043331
        43         44         45         46         47         48
 21.336824 100.736656   2.067017   7.074396   8.992994  42.299863
        49         50         51         52         53         54
  4.088580   5.573164   5.791517  37.770823   7.446839  16.074617
        55         56         57         58         59         60
 20.756777 101.689401   5.450175   9.755463  14.582657  60.900833
        61         62         63         64
  1.077335   4.746732   6.358566  23.936524

$se.fit
         1          2          3          4          5          6
 2.4467288  2.3205295  1.8336491  7.8310962  3.8413334  4.7928758
         7          8          9         10         11         12
 5.1246179 14.3024535  2.3711431  3.4120459  3.5100699  9.7020485
```

```
       13        14        15        16        17        18
0.6102392 1.2790771 1.4898542 4.7613246 1.1421776 1.3282707
       19        20        21        22        23        24
1.2209058 4.8532204 2.1783800 3.0548953 3.3675926 11.2937839
       25        26        27        28        29        30
1.2663282 2.2641762 2.3682811 7.3719026 0.2422804 0.9161009
       31        32        33        34        35        36
1.1466586 3.6141140 0.5077136 0.7620401 0.7818445 3.7658612
       37        38        39        40        41        42
0.8464585 1.6946155 2.1564105 9.0165498 0.5017790 1.1253227
       43        44        45        46        47        48
1.4423277 5.3432260 0.1998079 0.5918393 0.7299460 2.9526387
       49        50        51        52        53        54
0.3792472 0.4624035 0.4724435 2.6112194 0.6542183 1.2184624
       55        56        57        58        59        60
1.5194113 6.0900921 0.4925305 0.7626322 1.1045784 3.8778838
       61        62        63        64
0.1103283 0.4303229 0.5610220 1.8759244

$residual.scale
[1] 1
```

Printing confidence intervals:

```
print(paste(round(mod2.pred$fit[1],2)," (",
 round(mod2.pred$fit[1]-qt(.975,54)*mod2.pred$se.fit[1],2),",",
 round(mod2.pred$fit[1]+qt(.975,54)*mod2.pred$se.fit[1],2),
 ")",sep=""), quote=F)
```

```
[1] 31.86 (26.96,36.77)
```

Note that these can be embedded within your Rmarkdown code as 31.86 (26.96, 36.77). While the code may seem to be a lot more than just typing in the results directly, the advantage it has is that you have made a mistake in data entry at some point and need to rerun your code, the new results will automatically update here.

5.3 Practical exercises

5.3.1 Exercise 1: Missing persons data

The following data set appeared in The Independent, March 8, 1994, under the headline "Thousands of people who disappear without trace". Here, using figures from the Metropolitan police, the numbers in the table are of the form r/n where n = the number reported missing during the year ending March 1993 and r = the number still missing at the end of that year. Questions of interest are whether a simple model fits these data, whether the age and/or sex effect are significant, and how to interpret the statistical conclusions to the layman.

	Age in years	
	13 and under	14-18
Males	33/3271	63/7257
Females	38/2486	108/8877

- Read in the data n and r.
- Create factors sex and age.

- Create the data frame missing.df and remove unwanted files.
- Fit a Poisson regression using the appropriate offset; that is the linear predictor determined by $r \sim sex + age + offset(\log(n))$.
- Describe and interpret these results.
- Simplify the Poisson if possible. Comment on the final model.

5.3.2 Exercise 2: AIDS data

The total number of reported new cases per month of AIDS in the UK up to November 1985 are listed below (data from A. Sykes, 1986):

0,0,3,0,1,1,1,2,2,4,2,8,0,3,4,5,2,2 ,2,5,4,3,15,12,7,14,6,10,14,8,19,10,7,20,10,19

(data for 36 consecutive months - reading across).

- Read in the data y.
- Create a covariate for month number using i=1:36.
- Plot number of new cases against month; note that y (more or less) increases as i increases.
- Create aids.df data frame.
- Fit a model that has the following simple log-linear relationship; $\log(\mu_i) = \alpha + \beta_i$.
- Can β be dropped?

5.3.3 Exercise 3: Composite material cracks data

The following data records the experimental results of an investigation into the strength properties of two types of composite materials. Specimens were subjected to various forces in Newtons, x, and the ensuing number of cracks appearing was recorded, y.

Material 1	y	4	4	4	7	9	10	3
	x	0.1	0.4	0.6	0.8	1.0	1.3	1.4
Material 2	y	6	9	9	12	10	13	15
	x	0.0	0.1	0.2	0.3	0.4	0.5	0.6

Investigate two possible models:

1. $\eta_i = \beta_0 + \beta^A_{j(i)} + \beta^X x_i + \beta^{AX}_{j(i)} x_i$
2. $\eta_i = \beta_0 + \beta^A_{j(i)} + \beta^X x_i$

What do the two models represent? Perform any model simplification that is appropriate.

5.4 Over-dispersion

Over-dispersion can be a problem when working with the 1-parameter error distributions (e.g. Poisson or binomial errors) and occurs when the variance of the response exceeds the nominal variance. Over-dispersion tends to occur when:

- you have not measured one or more factors that turn out to be important;
- the underlying distribution being non-Poisson.

The net result of these will be that the residual deviance is inflated.

For the 1-parameter error distributions the scale parameter ϕ is assumed to be one. The usual estimator of ϕ is $\hat{\phi} = D/(n-p)$ and thus the residual deviance divided by the residual degrees of freedom ought to be

one. If this ratio is substantially larger than the assumed scale parameter of one (i.e. the residual deviance is much greater than the degrees of freedom), then this would suggest that the data are over-dispersed.

It is dealt with by adjusting the scale parameter to be a value other than one. This can be dealt with in two ways:

- using the quasipoisson family which differs from the poisson family because the dispersion parameter is not fixed at one, so it can model over-dispersion. The usual link functions are available.
- using the quasi family which allows us to estimate model parameters without fully specifying the error distribution of the response. However, we must specify the link and variance functions (so this requires a bit more consideration).

Note:

- Significance is then assessed by F-tests, using the estimated scale parameter as the denominator.
- The parameter estimates are not affected by this procedure, but the standard errors are inflated.
- Significance tests are therefore more stringent and can lead to fewer significant effects.
- family=quasi(link=''log'',var=''mu'') is the same as quasipoisson

Under-dispersion is when the variance of the response is less than the nominal variance. It is dealt with in a similar fashion.

5.4.1 Example 5: Poisson (slugs)

Count data were obtained on the number of slugs under 40 tiles placed over two types of grassland; nursery and rookery. Does the mean slug density differ between the two?

```
slugs<-read.table("slugsurvey.txt",header=T)
names(slugs)
```

```
[1] "count" "field"
```

```
attach(slugs)
plot(count~field,xlab="Field",ylab="count")
```

```
summary(count[field=="Nursery"])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   1.275   1.250  10.000
```

```
summary(count[field=="Rookery"])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   2.000   2.275   3.000   9.000
```

```
detach(slugs)
```

From the above looks like the medians are different, but the range of counts in both fields is large and so significance may be in doubt. The data does not appear to be normal as for the nursery it is heavily positive skewed. Also, it looks like the variance between the fields may not be constant. So standard ANOVA does not look appropriate.

We start by fitting Poisson model with log link:

```
mod1<-glm(count~field,poisson,data=slugs)
summary(mod1)
```

```
Call:
glm(formula = count ~ field, family = poisson, data = slugs)
```

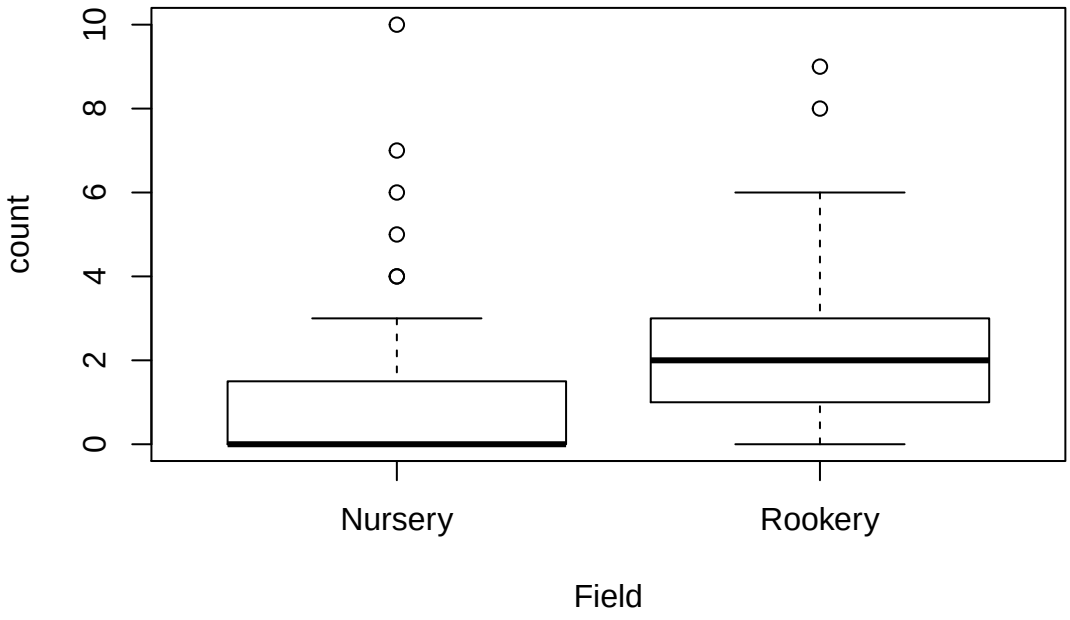

Figure 12: Slug count by Location

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1331  -1.5969  -0.9519   0.4580   4.8727

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.2429     0.1400   1.735 0.082744 .
fieldRookery   0.5790     0.1749   3.310 0.000932 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 224.86  on 79  degrees of freedom
Residual deviance: 213.44  on 78  degrees of freedom
AIC: 346.26

Number of Fisher Scoring iterations: 6
```

```
mod2<-update(mod1,.~.-field)
anova(mod1,mod2,test="Chi")
```

```
Analysis of Deviance Table

Model 1: count ~ field
Model 2: count ~ 1
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        78     213.44
2        79     224.86 -1  -11.422 0.000726 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(mod1)
```

This suggests that field is significant. BUT it appears that over-dispersion is present; from the output we would estimate the scale parameter to be 213.44/78=2.74 which is much more than one which it should be. Q-Q plot is very concave.

We shall now try the quasipoisson family still with log link. This will allow for the presence of a non unity scale parameter.

```
mod3<-glm(count~field,quasipoisson,data=slugs)
summary(mod3)
```

```
Call:
glm(formula = count ~ field, family = quasipoisson, data = slugs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1331  -1.5969  -0.9519   0.4580   4.8727

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.2429     0.2494   0.974   0.3331
```

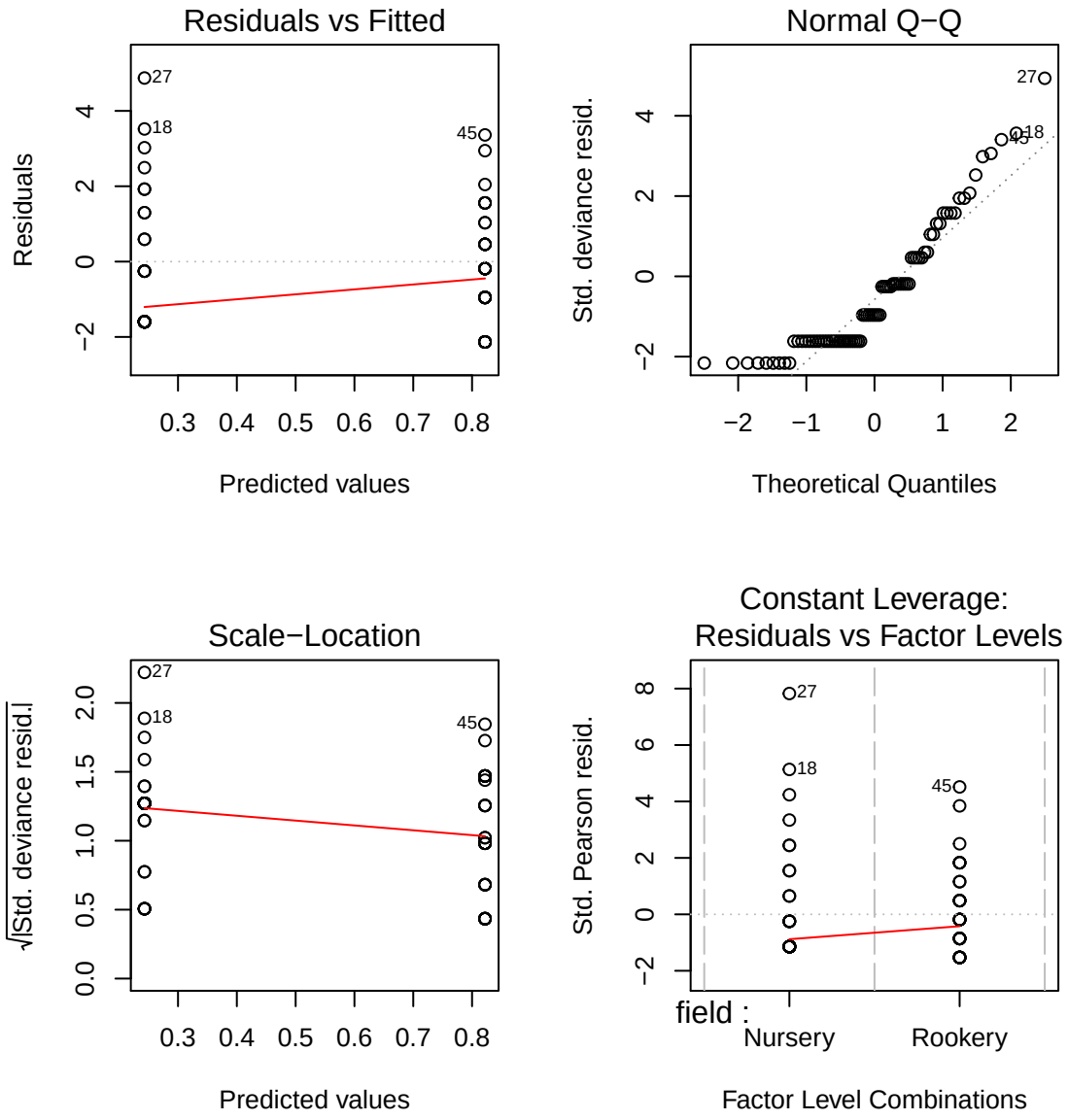

Figure 13: Diagnostic Plots for mod1

```
fieldRookery   0.5790     0.3116   1.858   0.0669 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 3.17311)

    Null deviance: 224.86  on 79  degrees of freedom
Residual deviance: 213.44  on 78  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

Deletion tests now need to be done using the F distribution.

```
mod4<-update(mod3,.~.-field)
anova(mod3,mod4,test="F")
```

```
Analysis of Deviance Table

Model 1: count ~ field
Model 2: count ~ 1
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1        78     213.44
2        79     224.86 -1  -11.422 3.5995 0.0615 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(mod3)
```

We can now see that the difference in mean slug density is not significant. The Q-Q plot isn't improved much. We could try the quasi family with log link which allows us to break away from the Poisson relationship between the mean and the variance.

```
mod5<-glm(count~field,quasi(link="log",var="mu^2"),data=slugs)
summary(mod5)
```

```
Call:
glm(formula = count ~ field, family = quasi(link = "log", var = "mu^2"),
    data = slugs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7232  -0.1261   0.0000   0.2900   3.0931

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.2429     0.2300   1.056   0.2941
fieldRookery   0.5790     0.3252   1.780   0.0789 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasi family taken to be 2.115738)

    Null deviance: 40.144  on 79  degrees of freedom
Residual deviance: 45.606  on 78  degrees of freedom
```

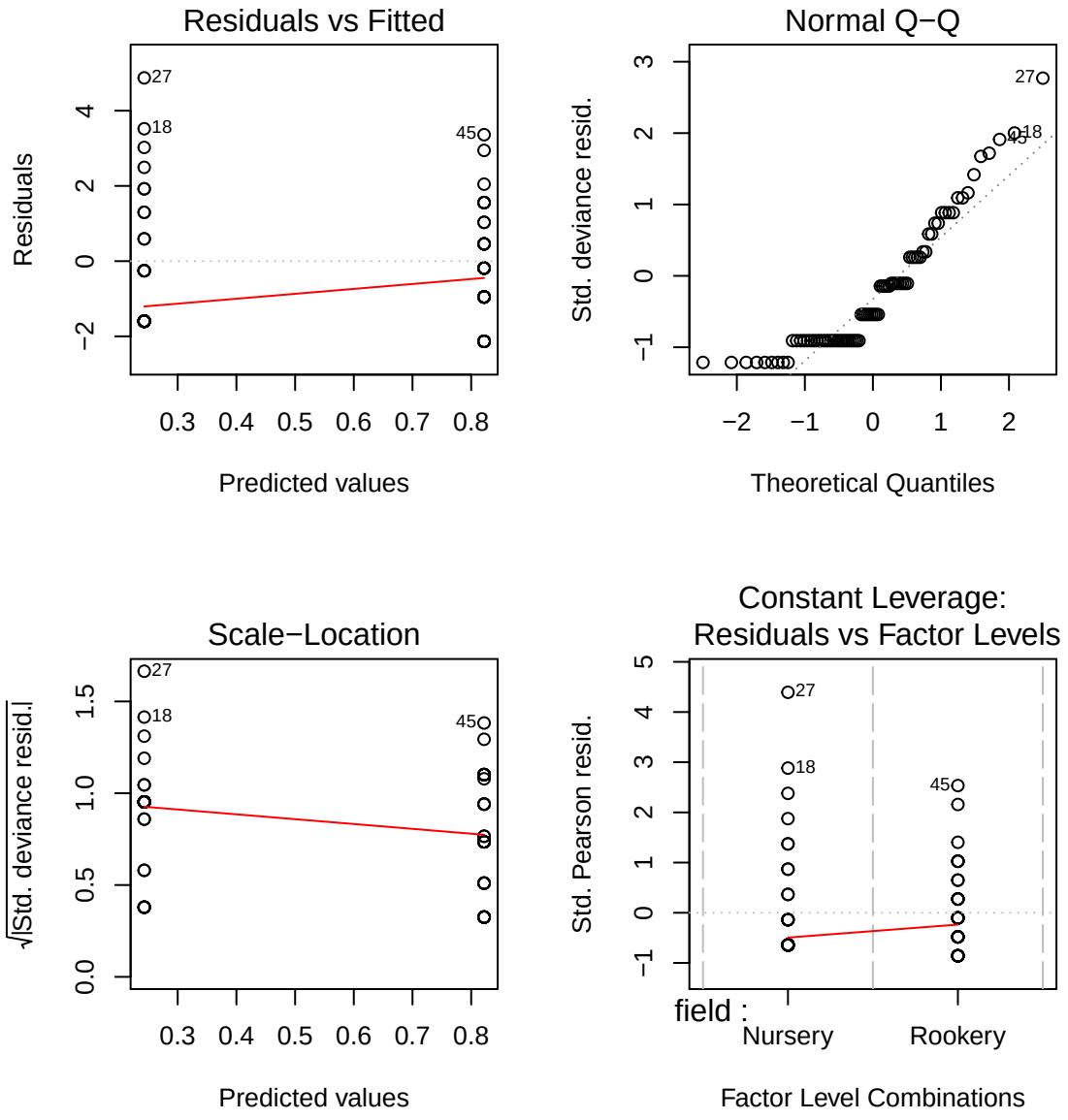

Figure 14: Diagnostic Plots for mod3

```
AIC: NA

Number of Fisher Scoring iterations: 11
```

```
mod6<-update(mod5,.~.-field)
anova(mod5,mod6,test="F")
```

```
Analysis of Deviance Table

Model 1: count ~ field
Model 2: count ~ 1
  Resid. Df Resid. Dev Df Deviance F Pr(>F)
1        78     45.606
2        79     40.144 -1   5.4615
```

```
par(mfrow=c(2,2))
plot(mod5)
```

Again a non-significant result but Q-Q plot has got worse in Figure 15.

Alternative approaches for this problem could be to use ANOVA after transformation to get normality/constant variance; try either log or sqrt transformations. Note we have to add 1 to count for log transformation as there are zero counts in this dataset.

```
attach(slugs)
plot(log(count+1)~field)
```

```
mod7<-glm(log(count+1)~field,data=slugs)
summary(mod7)
```

```
Call:
glm(formula = log(count + 1) ~ field, data = slugs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9698  -0.4967  -0.2766   0.4165   1.9012

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.4967     0.1117   4.446 2.86e-05 ***
fieldRookery   0.4730     0.1580   2.994  0.00369 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.4993769)

    Null deviance: 43.426  on 79  degrees of freedom
Residual deviance: 38.951  on 78  degrees of freedom
AIC: 175.45

Number of Fisher Scoring iterations: 2
```

```
mod8<-update(mod7,.~.-field)
anova(mod7,mod8,test="F")
```

```
Analysis of Deviance Table
```

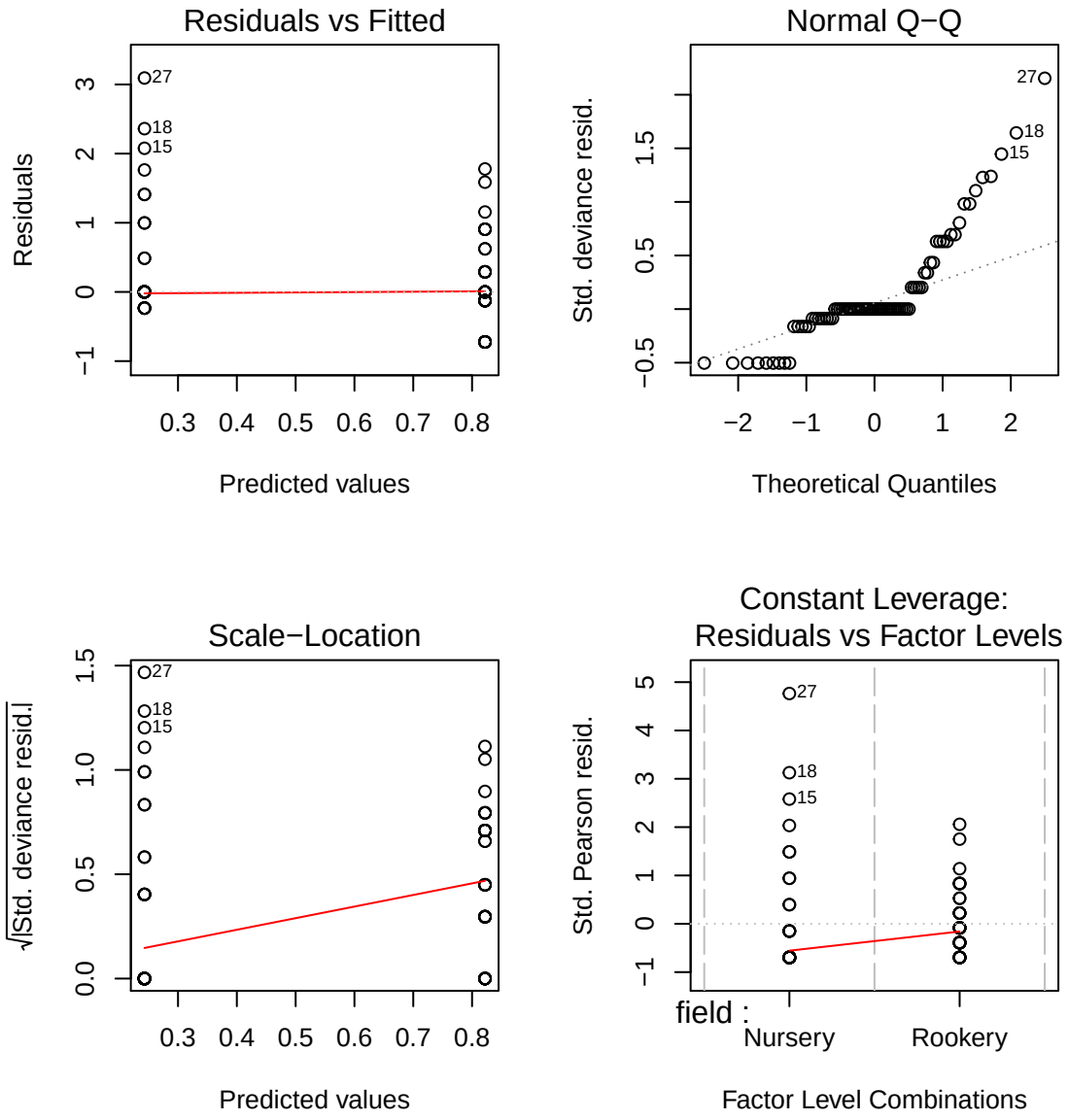

Figure 15: Diagnostic Plots for mod5

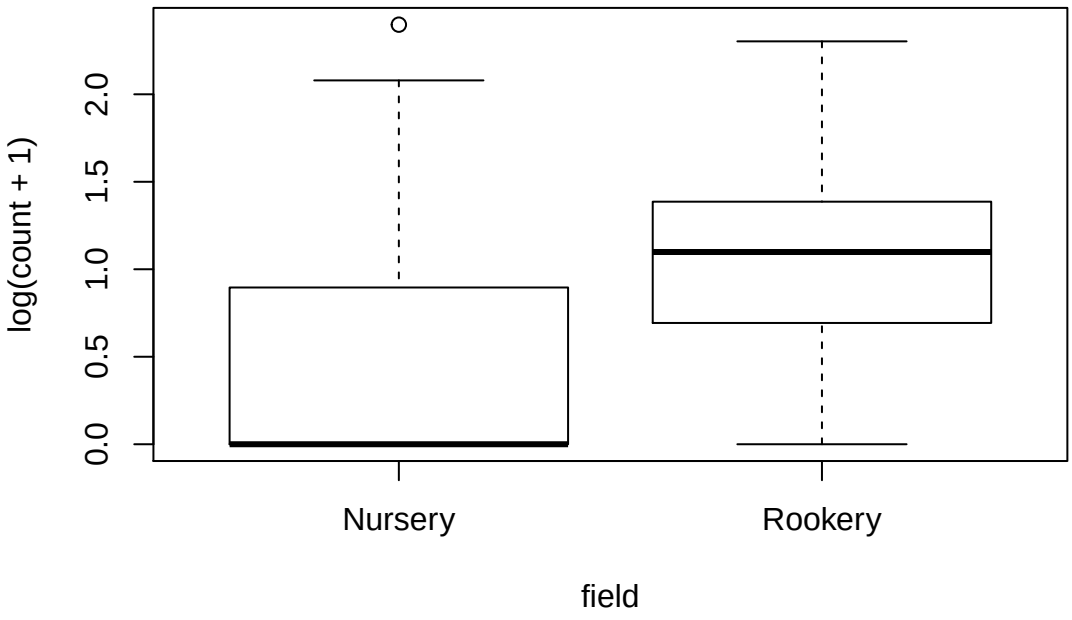

Figure 16: Log Transformation

```
Model 1: log(count + 1) ~ field
Model 2: log(count + 1) ~ 1
  Resid. Df Resid. Dev Df Deviance      F   Pr(>F)
1        78     38.951
2        79     43.426 -1   -4.475 8.9612 0.003693 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(mod7)
```

Here we get a significant effect for field. But the associated Q-Q plots still not great in Figure 17.

Try a square root transformation (as in Figure 18 and associated output).

```
 plot(sqrt(count)~field)
```

```
mod9<-glm(sqrt(count)~field,data=slugs)
summary(mod9)
```

```

Call:
glm(formula = sqrt(count) ~ field, data = slugs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2549  -0.6446  -0.2549   0.4772   2.5176

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.6447     0.1414   4.559 1.88e-05 ***
fieldRookery   0.6103     0.2000   3.052  0.00311 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.799815)

    Null deviance: 69.834  on 79  degrees of freedom
Residual deviance: 62.386  on 78  degrees of freedom
AIC: 213.13

Number of Fisher Scoring iterations: 2
```

```
mod10<-update(mod9,.~.-field)
anova(mod9,mod10,test="F")
```

```
Analysis of Deviance Table

Model 1: sqrt(count) ~ field
Model 2: sqrt(count) ~ 1
  Resid. Df Resid. Dev Df Deviance      F   Pr(>F)
1        78     62.386
2        79     69.834 -1  -7.4481 9.3123 0.003111 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

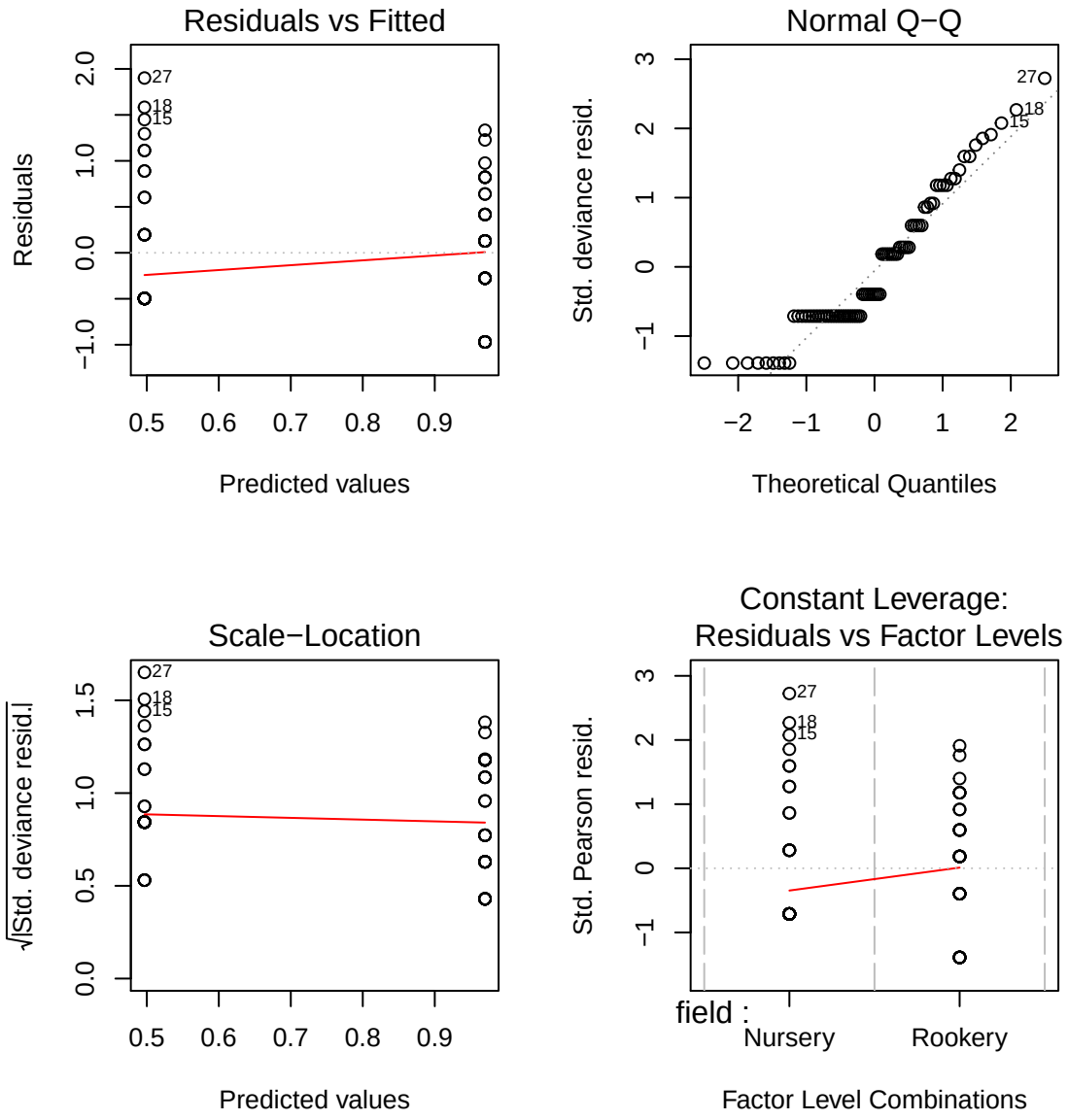

Figure 17: Diagnostic Plots for mod7

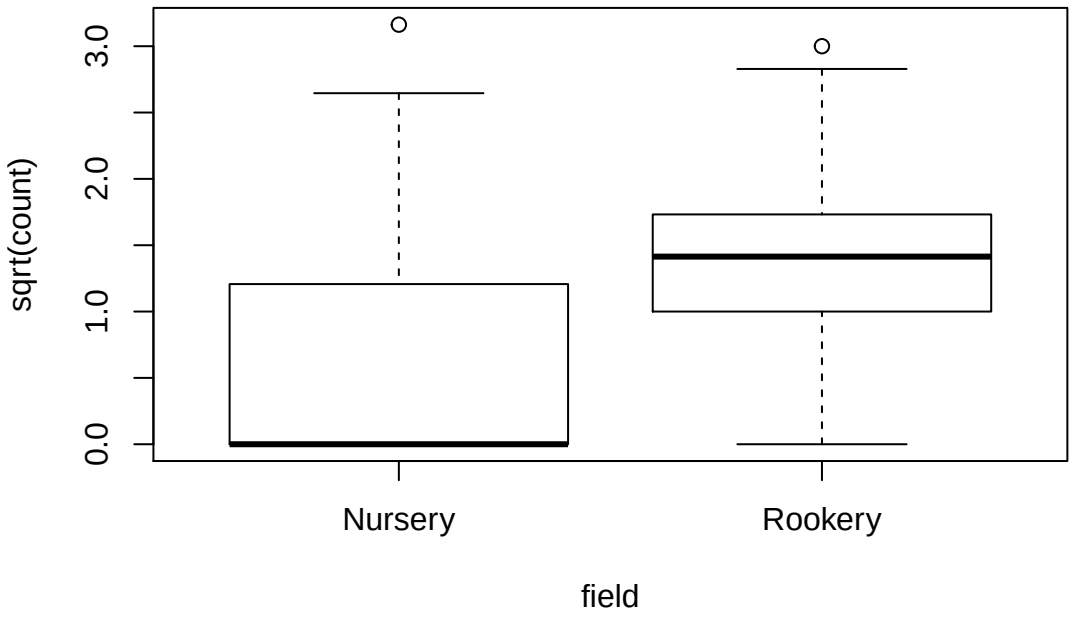

Figure 18: Square Root Transformation

```
par(mfrow=c(2,2))
plot(mod9)
```

```
detach(slugs)
```

Again a significant result for field, but Q-Q plots no better. We have contradictory results from the above which is common with data with low means and high variances. The residual plots in Figure 19 are not helping us in identifying which is the best model. There is no obvious right answer, but the results correcting for over-dispersion are warning us that conclusions from the ANOVA approach should be treated with caution especially as the transformed data does not appear normal.

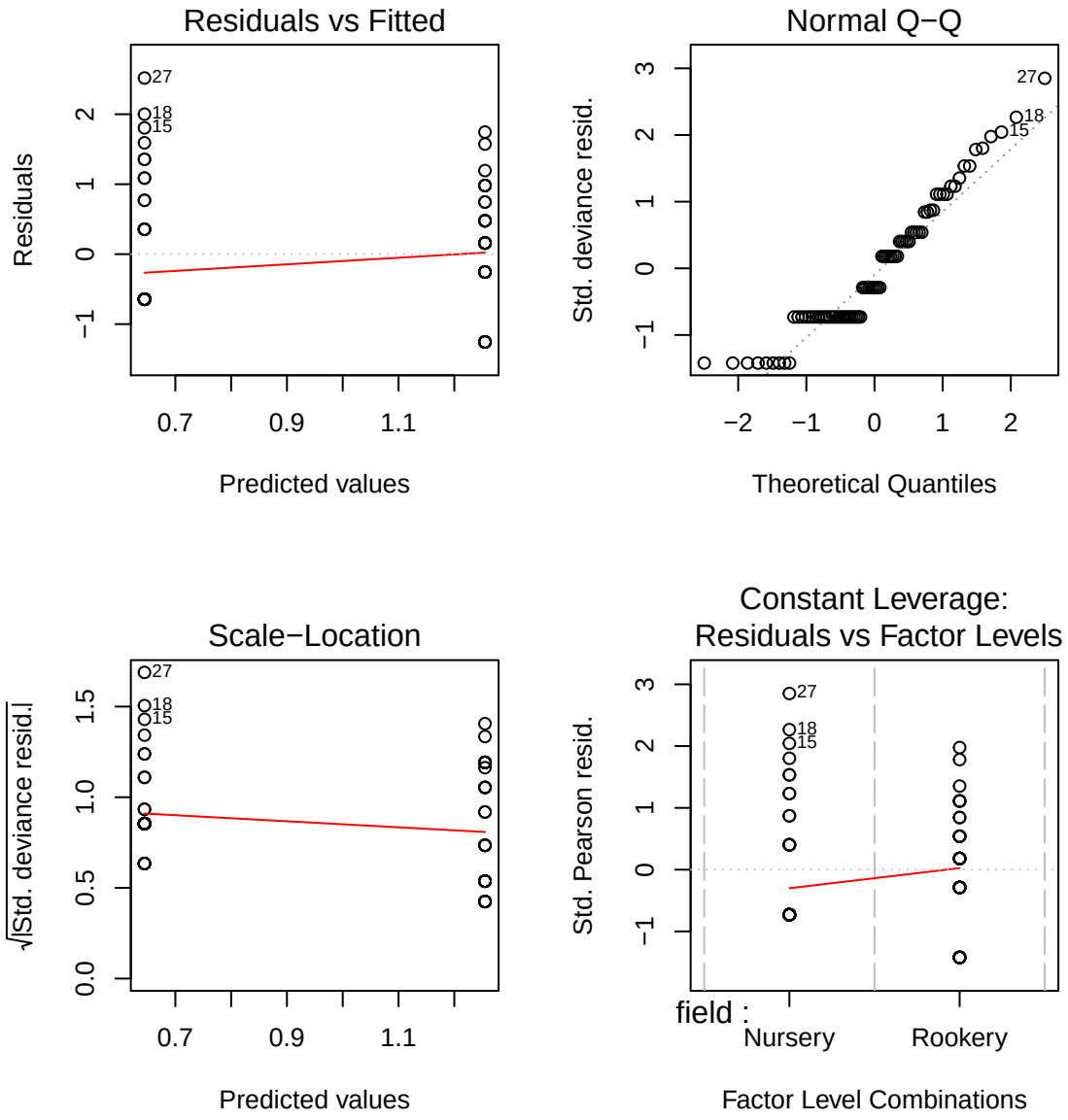

Figure 19: Diagnostic Plots for mod9