Sabine WOSCHITZ

Random Effect Models and their implementation in R

MASTERARBEIT

zur Erlangung des akademischen Grades einer/s Diplom-Ingenieur/in

Masterstudium Operations Research und Statistik

Technische Universität Graz

Betreuer/in:
Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Herwig FRIEDL

Institut für Statistik

Graz, im Juni 2011

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....

(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitely marked all material which has been quotes either literally or by content from the used sources.

....

date (signature)

Preface

The literature on random effect models is now extensive. One general form of a random effect model is a generalized linear model where a random effect is added into the linear predictor. If the distribution of this random effect is conjugate to the distribution of an exponential family, maximum likelihood estimation is straightforward, at least in simple models like the negative binomial and the beta binomial model. By assuming a normal distribution for the distribution of this random effect the parameters can be estimated via EM-algorithm and Gaussian quadrature. If no assumption for the distribution of the random effect can be made, a non-parametric maximum likelihood approach can be used.

The thesis is divided into three chapters. Chapter 1 serves as an introduction to the class of generalized linear models, where maximum likelihood estimation and quasi-likelihood functions are discussed. Chapter 2 gives detailed account of simple overdispersion models, including the beta binomial and the negative binomial model. Furthermore, extra-binomial and extra-Poisson variation is explained and analyzed. Chapter 3 deals with random effect models, especially random intercept and variance component models.

I want to thank Prof. Dr. Herwig Friedl for giving me the opportunity of writing the thesis about this very interesting topic and for his faithful supervision. Without that I wouldn't have managed to finish this master thesis on time.

Moreover, I want to thank all people who helped me with the work on this thesis, in particular my family and Nikolaus Furian for supporting me during my study time.

Any errors in the text and procedures are entirely my responsibility.

Graz, Mai 2011 Sabine Woschitz

Contents

List of Tables

Chapter 1

Generalized Linear Models

In this chapter the classical linear model will be generalized to cover many other models. **Generalized linear models** include for example log-linear models for the analysis of count data, logit and probit models for data in the form of proportions and models with continuous data. All these models share some properties, such as linearity and a common method to compute parameter estimates.
First, we revisit the linear model, as the basis for further regression analysis. Then there will be a short introduction into the terms exponential family and link function, to specify the generalized linear model. Section 1.3. discusses the maximum likelihood estimation and the algorithm of iterative weighted least squares, in the following denoted by IWLS. In the next section we consider the quasi-likelihood approach with the maximum quasi-likelihood estimation and some of the properties of the quasi-likelihood function. Finally two measures of fit for a model will be discussed in Section 1.5.

The chapter is based on McCullagh and Nelder (1989).

1.1 The Linear Model

Generally one can say that regression analysis is based on analyzing the relationship between explanatory variables $x_1, \ldots, x_p$ and response variables $y = (y_1, \ldots, y_n)^T$.
The assumptions made for the classical linear model are:

- The response variables y_i, $i = 1, \ldots, n$, are independent normally distributed and the variance of y_i, $\mathrm{Var}(y_i) = \sigma^2$, is constant.
- The explanatory variables $x_1, x_2, \ldots, x_p$, all of length n, form the $n \times p$ model matrix X.

- $\mathbb{E}(y) = \mu$ can be modeled through a linear combination of the explanatory variables,

  $$\mu = X\beta$$

  with β being a parameter vector of length p, or

  $$y = X\beta + \epsilon$$

  with ϵ having mean zero and variance matrix $\sigma^2 I_{n\times n}$.

The assumptions regarding constant variance and normality are not very common in real world datasets. For dealing with discrete data, the normality assumption is not appropriate at all. Moreover, the constant-variance assumption is sometimes neglected, as the variance could depend on the mean. To circumvent this difficulty one approach is to transform the dataset. One famous transformation is the Box-Cox transformation (see Box and Cox, 1964), where the transformed response variables are assumed to be normally distributed with a constant variance. The other approach leads to the so called generalized linear models. There it is possible to assume a whole class of distributions for the response variables. Furthermore, a function of the mean is modeled linearly and not the mean itself as in the classical linear model. In addition the variance is allowed to also depend on the mean.

1.2 Generalized Linear Models

In order to define the class of generalized linear models we first need to introduce the terms exponential family and link function.

1.2.1 Exponential Family

First we define the general k-parameter exponential family (see also McCullagh, Searle, and Neuhaus, 2008) and then the one-parameter exponential family.

Definition 1.1 (k-parameter exponential family) *A family of probability density functions (pdf) or probability mass functions (pmf) is called an exponential family with parameter vector $\theta = (\theta_1, \ldots, \theta_k)^T$, if it can be written as*

$$f(y, \theta) = h(y) \cdot c(\theta) \cdot \exp\left(\sum_{i=1}^{k} b_i(\theta) \cdot t_i(y)\right),$$

where $h(y) \geq 0$ and $t_1(y), \ldots, t_k(y)$ are functions of y and independent of θ, $c(\theta) \geq 0$ and $b_1(\theta), \ldots, b_k(\theta)$ are functions of the parameter θ, independent of y. Further θ is called the canonical vector of the exponential family.

In generalized linear regression analysis a k-parameter exponential family is not useful since we only need a one-parameter exponential family. Therefore, McCullagh and Nelder (1989) proposed an alternative formulation of the exponential family with one parameter.

Definition 1.2 (one-parameter exponential family)

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

for known functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ with $a(\phi) > 0$. If $\phi > 0$ is known, $f(y,\theta)$ is called the one-parameter exponential family with canonical parameter θ.

To derive the mean and the variance of this exponential family, we will first have a look at the score function.

Theorem 1.1 *For the score function of an exponential family, the following equations hold:*

$$\mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right) = 0, \tag{1.1}$$

$$\mathrm{Var}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right) = \mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2 = \mathbb{E}\left(-\frac{\partial^2 \log f(y,\theta)}{\partial\theta^2}\right) \tag{1.2}$$

Proof:
The derivative of the log-likelihood function is

$$\frac{\partial \log f(y,\theta)}{\partial\theta} = \frac{1}{f(y,\theta)}\frac{\partial f(y,\theta)}{\partial\theta}$$

and since a pmf or a pdf is normalized,

$$\int_{\mathbb{R}} f(y,\theta)dy = 1.$$

Therefore

$$\begin{aligned}
\mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right) &= \mathbb{E}\left(\frac{\partial f(y,\theta)}{\partial\theta}\frac{1}{f(y,\theta)}\right) \\
&= \int_{\mathbb{R}} \frac{\partial f(y,\theta)}{\partial\theta}\frac{1}{f(y,\theta)} f(y,\theta)dy \\
&= \int_{\mathbb{R}} \frac{\partial f(y,\theta)}{\partial\theta}dy \\
&= \frac{\partial}{\partial\theta}\int_{\mathbb{R}} f(y,\theta)dy \\
&= 0
\end{aligned}$$

and (1.1) follows.

Result (1.2) is derived by using the chain rule

$$\begin{aligned}
\mathbb{E}\left(-\frac{\partial^2 \log f(y,\theta)}{\partial\theta^2}\right) &= \mathbb{E}\left(-\frac{\partial^2(y,\theta)}{\partial\theta^2}\frac{1}{f(y,\theta)} + \frac{\partial f(y,\theta)}{\partial\theta}\frac{\partial f(y,\theta)}{\partial\theta}\frac{1}{f(y,\theta)^2}\right) \\
&= \mathbb{E}\left(-\frac{\partial^2 f(y,\theta)}{\partial\theta^2}\frac{1}{f(y,\theta)}\right) + \mathbb{E}\left(\frac{\partial f(y,\theta)}{\partial\theta}\frac{\partial f(y,\theta)}{\partial\theta}\frac{1}{f(y,\theta)^2}\right) \\
&= \int_{\mathbb{R}} -\frac{\partial^2 f(y,\theta)}{\partial\theta^2}\frac{1}{f(y,\theta)} f(y,\theta)\,\mathrm{d}y + \int_{\mathbb{R}}\left(\frac{\partial f(y,\theta)}{\partial\theta}\right)^2 \frac{f(y,\theta)}{f(y,\theta)^2}\,\mathrm{d}y \\
&= -\frac{\partial^2}{\partial\theta^2}\int_{\mathbb{R}} f(y,\theta), dy + \int_{\mathbb{R}}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2 f(y,\theta)^2 \frac{f(y,\theta)}{f(y,\theta)^2}\,\mathrm{d}y \\
&= \int_{\mathbb{R}}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2 f(y,\theta)\,\mathrm{d}y \\
&= \mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2.
\end{aligned}$$

□

From this it follows that the corresponding score equation is

$$\begin{aligned}
\frac{\partial \log f(y,\theta)}{\partial\theta} &= \frac{1}{a(\phi)}\left(y - b'(\theta)\right) \\
\Leftrightarrow\ \mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right) &= \frac{1}{a(\phi)}\mathbb{E}\left(y - b'(\theta)\right) = 0 \\
\Leftrightarrow\ \mathbb{E}(y) &= b'(\theta) = \mu
\end{aligned}$$

and

$$\begin{aligned}
0 &= \mathbb{E}\left(\frac{\partial^2 \log f(y,\theta)}{\partial\theta^2}\right) + \mathbb{E}\left(\frac{\partial \log f(y,\theta)}{\partial\theta}\right)^2 \\
&= -\frac{1}{a(\phi)}b''(\theta) + \frac{1}{a^2(\theta)}\mathbb{E}(y - b'(\theta))^2
\end{aligned}$$

with

$$\mathbb{E}(y - b'(\theta)) = \mathrm{Var}(y).$$

Thus,

$$\mathrm{Var}(y) = a(\phi)b''(\theta) = a(\phi)V(\mu),$$

where mean and variance are also called the first two moments or cumulants of the exponential family. The function $V(\mu)$ is called the variance function, which dependents on μ but which is independent of ϕ. Furthermore, ϕ is called dispersion parameter and $a(\phi)$ is independent from μ. The function $a(\phi)$ is commonly of the form

$$a(\phi) = a \cdot \phi$$

with ϕ being constant over all observations and a is a known weight which varies from observation to observation.

In the following we will have a short overview over the cumulant generating function, to derive moments of order greater than two.

The kth cumulant can be calculated by using the cumulant generating function $K(t) = \log M(t)$ with $M(t)$ being the moment generating function. The kth cumulant κ_k is given by

$$\kappa_k(y) = K^{(k)}(t)|_{t=0}.$$

To derive the cumulant generating function we will first consider the moment generating function. With

$$1 = \int_{\mathbb{R}} f(y, \theta) dy$$

it follows for $f(y, \theta)$ being from the exponential family that

$$1 = \int_{\mathbb{R}} \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right) \mathrm{d}y = \exp\left(-\frac{b(\theta)}{a(\phi)}\right) \int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}\theta + c(y, \phi)\right) \mathrm{d}y.$$

From this it follows that

$$\exp\left(\frac{b(\theta)}{a(\phi)}\right) = \int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}\theta + c(y, \phi)\right) \mathrm{d}y$$

and the moment generation function $M(t)$ is

$$\begin{aligned} M(t) = \mathbb{E}(e^{ty}) &= \int_{\mathbb{R}} e^{ty} f(y,\theta) dy \\ &= \int_{\mathbb{R}} \exp(ty) \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right) \mathrm{d}y \\ &= \exp\left(-\frac{b(\theta}{a(\phi)}\right) \int_{\mathbb{R}} \exp\left(\frac{y}{a(\phi)}(\theta + a(\phi)t) + c(y,\phi)\right) \\ &= \exp\left(-\frac{b(\theta)}{a(\phi)}\right) \exp\left(\frac{b(\theta + a(\phi)t)}{a(\phi)}\right) \\ &= \exp\left(\frac{b(\theta + a(\phi)t) - b(\theta)}{a(\phi)}\right). \end{aligned}$$

Therefore, we have

$$\kappa_k(y) = K^{(k)}(t)|_{t=0} = a(\phi)^{k-1} b^{(k)}(\theta + a(\phi)t)\,|_{t=0} = a(\phi)^{k-1} b^{(k)}(\theta).$$

So for the first two cumulants we get

$$\mathbb{E}(y) = b'(\theta) = \mu$$

and

$$\mathrm{Var}(y) = a(\phi) b''(\theta) = a(\phi) V(\mu).$$

As in the following we will often model Poisson and binomial responses, we show that the Poisson and the binomial distribution are members of the one-parameter exponential family.

Example 1.1 (Poisson distribution) *Suppose the response y follows a Poisson distribution with parameter μ: $y \sim P(\mu)$. The pmf is then given by*

$$f(y,\mu) = P(y = y) = \frac{\mu^y}{y!} e^{-\mu} = \exp(y \log \mu - \mu - \log y!), \quad y = 0, 1, 2, \ldots.$$

Assuming that $\theta = \log \mu$ and ϕ=1 leads to an exponential family with

$$a(\phi) = \phi, \quad b(\theta) = \mu = \exp(\theta), \quad c(y,\phi) = -\log y!$$

and

$$\begin{aligned} \mathbb{E}(y) &= b'(\theta) = \exp(\theta) = \mu, \\ \mathrm{Var}(y) &= a(\phi) b''(\theta) = 1 \cdot \exp(\theta) = \mu, \\ \kappa_k(y) &= a(\phi)^{k-1} b^{(k)}(\theta) = \exp(\theta) = \mu, \quad \textit{for} \quad k > 2. \end{aligned}$$

Example 1.2 (Binomial distribution in its standard form) *Let us assume that the response y follows a standardized binomial distribution, that is $my \sim B(m, \mu)$. Then the pmf is given by*

$$\begin{aligned} f(y,m,\mu) &= P(y=y) = P(my=my) = \binom{m}{my}\mu^{my}(1-\mu)^{m-my} \\ &= \exp\left(\log\binom{m}{my} + my\log\mu + (m-my)\log(1-\mu)\right) \\ &= \exp\left(\log\binom{m}{my} + my\log\mu + m\log(1-\mu) - my\log(1-\mu)\right) \\ &= \exp\left(\frac{y\log\frac{\mu}{1-\mu} - \log\frac{1}{1-\mu}}{1/m} + \log\binom{m}{my}\right). \end{aligned}$$

With $\theta = \log\frac{\mu}{1-\mu}$ and $a = 1/m$, $\phi = 1$, this is an exponential family with

$$a(\phi) = a\cdot\phi, \quad b(\theta) = \log\frac{1}{1-\mu} = \log\left(1+\exp(\theta)\right), \quad c(y,\phi) = \log\binom{1/\phi}{y/\phi}$$

and

$$\begin{aligned} \mathbb{E}(y) &= b'(\theta) = \frac{\exp(\theta)}{1+\exp(\theta)} = \mu \\ \mathrm{Var}(y) &= a(\phi)b''(\theta) = \phi\frac{\exp(\theta)}{(1+\exp(\theta))^2} = \frac{1}{m}\mu(1-\mu) \\ \kappa_3(y) &= a(\phi)^2 b^{(3)}(\theta) = \frac{1}{m^2}\frac{\exp(\theta)(1+\exp(\theta))^2 - 2\exp^2(\theta)(1+\exp(\theta))}{(1+\exp(\theta))^4} \\ &= \frac{1}{m^2}\frac{\exp(\theta)-\exp^2(\theta)}{(1+\exp(\theta))^3} = \frac{1}{m^2}\frac{\exp(\theta)(1-\exp(\theta))}{(1+\exp(\theta))^3} \\ &= \frac{1}{m^2}\mu(1-\mu)\underbrace{\frac{1-\exp(\theta)}{1+\exp(\theta)}}_{\frac{1}{1+\exp(\theta)}-\frac{\exp(\theta)}{1+\exp(\theta)}=1-\mu-\mu=1-2\mu} \\ &= \frac{1}{m^2}(1-2\mu)\mu(1-\mu) \\ \kappa_4(y) &= \frac{1}{m^3}(1-6\mu(1-\mu))\mu(1-\mu). \end{aligned}$$

Other continuous members of the exponential family are for example normal, gamma and inverse Gaussian. All with a different variance functions $V(\mu)$, different canonical parameters θ and therefore different functions $b(\theta)$. Table 1.1 shows the forms of $V(\mu)$, θ, $b(\theta)$ for the five main GLM distributions (see also Lee, Nelder, and Pawitan, 2006).

	$V(\mu)$	θ	$b(\theta)$
normal	1	μ	$\theta^2/2$
Poisson	μ	$\log \mu$	$\exp(\theta)$
binomial	$\mu(1-\mu)$	$\log \frac{\mu}{1-\mu}$	$\log(1+\exp(\theta))$
gamma	μ^2	$-1/\mu$	$-\log(-\theta)$
inverse Gaussian	μ^3	$-1/2\mu^2$	$-(-2\theta)^{1/2}$

Table 1.1: $V(\mu)$, θ and $b(\theta)$ for some exponential family members.

1.2.2 Link Function

In contrast to the linear model, the predictor of a generalized linear model is allowed to be any monoton function of the mean. We write

$$g(\mu) = \eta = X^T\beta,$$

where g is called the link function, since it links the mean of y and the linear predictor η. In the Poisson case, with $y = 0$, transformation of the data to $\log y$ is useless, since $\log y$ is not defined for $y = 0$. To overcome this, $\log \mu = X\beta$ is considered instead, which causes no difficulty when $y = 0$.
Moreover, if $\eta = \theta$, the canonical parameter, then the corresponding link function is called canonical link. Canonical links are often used because they give rise to a simple sufficient statistic for the regression parameters. Some of the main distributions and their canonical links are shown in Table 1.2 (see also Faraway, 2006).

	link	
normal	identity	μ
Poisson	log	$\log \mu$
binomial	logit	$\log \frac{\mu}{1-\mu}$
gamma	reciprocal	$1/\mu$
inverse Gaussian		$1/\mu^2$

Table 1.2: Canonical links for some exponential family members.

1.2.3 Definition of the Generalized Linear Model

Generalized linear models can be derived from the the classical linear model through the following generalizations.

- The response variables y_i follow a distribution which is a member of the exponential family, i.e.

$$y_i \sim \text{Exponential family}(\theta_i)$$

with $\mathbb{E}(y_i) = \mu_i = \mu(\theta_i)$ and $\text{Var}(y_i) = a_i(\phi)V(\mu_i)$.

- The mean $\mathbb{E}(y_i) = \mu_i$ is modeled through a link function

$$g(\mu_i) = \eta_i$$

and a linear predictor

$$\eta_i = x_i^T \beta,$$

where $x_i = (x_{i1}, \ldots, x_{ip})^T$ is the vector of explanatory variables for the ith response which form the design matrix $X = (x_1, \ldots, x_n)^T$, $\beta = (\beta_1, \ldots, \beta_p)^T$ is the vector of the unknown parameters, $\eta = (\eta_1, \ldots, \eta_n)^T$ is the vector with the linear predictors and $g(.)$ is a known link function.

- The variance may depend on the mean by allowing

$$\text{Var}(y_i) = a_i(\phi)V(\mu_i),$$

where $V(\mu_i)$ is called the variance function and depends on μ_i, whereas ϕ is called the dispersion parameter and $a_i(\phi)$ is independent from μ_i.

1.3 Maximum Likelihood Estimation

In this section an introduction to maximum likelihood estimation in GLMs is given. For deriving the maximum likelihood estimates $\hat{\beta}$ the IWLS algorithm is used.

Suppose we have a GLM as defined in Section 1.2.3, then the parameter vector is given by (θ_i, ϕ), where $\theta = (\theta_1, \ldots, \theta_n)^T$ contains the unknown canonical parameters and ϕ is considered to be known for the moment. Then the likelihood function is given by

$$L(y, \theta) = \prod_{i=1}^{n} \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right)$$

and the corresponding log-likelihood function is therefore

$$l(y, \theta) = \log L(y, \theta) = \sum_{i=1}^{n} \left(\frac{y_i\theta - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right). \tag{1.3}$$

Since we are mostly interested in modeling the mean and thus in the parameter vector β, we consider the β score function

$$\begin{aligned}\frac{\partial l(y,\theta(\beta))}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial l(y_i,\theta(\beta))}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j} \\ &= \sum_{i=1}^{n} \frac{\partial l(y_i,\theta(\beta))}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j}, \qquad j=1,\ldots,p\ .\end{aligned}$$

With

$$\frac{\partial \mu}{\partial \theta} = \frac{\partial b'(\theta)}{\partial \theta} = b''(\theta) = V(\mu)$$

and

$$\frac{\partial \mu}{\partial \beta} = \frac{\partial \mu}{\partial \eta}\underbrace{\frac{\partial \eta}{\partial \beta}}_{x} = \frac{\partial \mu}{\partial g(\mu)}x = \frac{x}{g'(\mu)}$$

it follows that

$$\frac{\partial l(y,\theta(\beta))}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i-\mu_i}{a_i(\phi)V(\mu_i)}\frac{x_{ij}}{g'(\mu_i)} = 0, \qquad j=1,\ldots,p. \tag{1.4}$$

Remark 1.1 *With $\mu = b'(\theta)$ it follows for the canonical link $g(\mu) = \eta = \theta$ that $g(b'(\theta)) = \theta$. In this case $g(\cdot)$ is the inverse of $b'(\cdot)$ and*

$$g'(\mu) = \frac{\partial g(\mu)}{\partial \mu} = \frac{\partial \theta}{\partial \mu} = \frac{\partial \theta}{\partial b'(\theta)} = \frac{1}{b''(\theta)} = \frac{1}{V(\mu)}.$$

Thus, the score equation simplifies to

$$\frac{\partial l(y,\theta(\beta))}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i-\mu_i}{a_i(\phi)V(\mu_i)}x_{ij}V(\mu_i) = \sum_{i=1}^{n} \frac{y_i-\mu_i}{a_i(\phi)}x_{ij} = 0, \qquad j=1,\ldots,p. \tag{1.5}$$

To obtain the maximum likelihood estimate $\hat{\beta}$ one has to solve equations (1.4) resp. (1.5) in case of the canonical link function. Both of these equations can be solved iteratively with the Newton Raphson method. This leads to the IWLS algorithm.

1.3.1 Iterative Weighted Least Squares

To solve the equations (1.4) resp. (1.5) McCullagh and Nelder (1989) discussed the method of IWLS involving an adjusted dependent variable z. First we will apply the Newton Raphson method for finding successively better approximations to the roots of the score functions (1.4) and (1.5). This leads to

$$\beta^{(t+1)} = \beta^{(t)} + \left(-\frac{\partial^2 l(y,\theta(\beta))}{\partial \beta \partial \beta^T}\right)^{-1}\frac{\partial l(y,\theta(\beta))}{\partial \beta}, \qquad t=0,1,\ldots,$$

where the derivatives are evaluated in $\beta^{(t)}$. The score function can be written as

$$\begin{aligned}\frac{\partial l(y,\theta(\beta))}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{y_i-\mu_i}{a_i(\phi)V(\mu_i)}\frac{x_{ij}}{g'(\mu_i)} = \\ &= \sum_{i=1}^{n} \frac{y_i-\mu_i}{a_i(\phi)V(\mu_i)(g'(\mu_i))^2} g'(\mu_i)x_{ij}, \qquad j=1,\ldots,p.\end{aligned}$$

If we define

$$\begin{aligned} d_i &= g'(\mu_i), \\ 1/w_i &= a_i(\phi)V(\mu_i)(g'(\mu_i))^2 \end{aligned} \tag{1.6}$$

and $D = diag(d_i)$, $W = diag(w_i)$, the score equation can be rewritten in matrix notation as

$$\frac{\partial l(y,\theta(\beta))}{\partial \beta} = X^T DW(y-\mu).$$

Thus, the resulting negative Hessian matrix is given by

$$\begin{aligned} -\frac{\partial^2 l(y,\theta(\beta))}{\partial\beta\partial\beta^T} &= -\frac{\partial(X^T DW(y-\mu))}{\partial\beta^T} = -\frac{\partial(X^T DW(y-\mu))}{\partial\eta^T}\frac{\partial\eta^T}{\partial\beta^T} \\ &= -X^T\left(\frac{\partial DW}{\partial\eta^T}(y-\mu) + DW(-1)\frac{\partial\mu}{\partial\eta^T}\right)X \\ &= -X^T\left(\frac{\partial DW}{\partial\eta^T}(y-\mu) - DW\underbrace{\frac{\partial\mu}{\partial\eta^T}}_{1/g'(\mu)=D^{-1}}\right)X \\ &= X^T\left(W - \frac{\partial DW}{\partial\eta^T}(y-\mu)\right)X \end{aligned}$$

with

$$\begin{aligned} \frac{\partial d_i w_i}{\partial\eta_i} &= \frac{\partial\left(1/(a_i(\phi)V(\mu_i)g'(\mu_i))\right)}{\partial\eta_i} \\ &= -\frac{a_i(\phi)V'(\mu_i)\frac{\partial\mu_i}{\partial\eta_i}g'(\mu_i) + a_i(\phi)V(\mu_i)g''(\mu_i)\frac{\partial\mu_i}{\partial\eta_i}}{(a_i(\phi)V(\mu_i)g'(\mu_i))^2} \\ &= -\frac{a_i(\phi)\frac{\partial\mu_i}{\partial\eta_i}\left(V'(\mu_i)g'(\mu_i)+V(\mu)g''(\mu_i)\right)}{a_i(\phi)^2V(\mu_i)g'^3(\mu_i)}g'(\mu_i) \\ &\overset{\frac{\partial\mu_i}{\partial\eta_i}=\frac{1}{g'(\mu_i)}}{=} -\frac{V'(\mu_i)g'(\mu_i)+V(\mu_i)g''(\mu_i)}{a_i(\phi)V(\mu_i)g'^3(\mu_i)}. \end{aligned} \tag{1.7}$$

If we define

$$w_i^* = w_i - \frac{\partial d_i w_i}{\partial\eta_i}(y_i-\mu_i) \tag{1.8}$$

and $W^* = diag(w_i^*)$, then $\mathbb{E}(W^*) = W$ and the Newton Raphson method yields

$$\beta^{(t+1)} = \beta^{(t)} + \left(X^T W^* X\right)^{-1} X^T D W (y - \mu), \qquad t = 0, 1, \dots \ . \tag{1.9}$$

Introducing so called adjusted dependent variables z with

$$z = X\beta + W^{*-1} D W (y - \mu),$$

the Newton Raphson procedure (1.9) can be written as

$$\begin{aligned} \beta^{(t+1)} &= \beta^{(t)} + \left(X^T W^* X\right)^{-1} X^T D W (y - \mu) \\ &= \left(X^T W^* X\right)^{-1} X^T W^* X \beta^{(t)} + \left(X^T W^* X\right)^{-1} X^T W^* W^{*-1} D W (y - \mu) \\ &= \left(X^T W^* X\right)^{-1} X^T W^* \left(X\beta^{(t)} + W^{*-1} D W (y - \mu)\right) \\ &= \left(X^T W^* X\right)^{-1} X^T W^* z, \end{aligned} \tag{1.10}$$

which is an IWLS representation with an adjusted dependent variable z, where the right side is evaluated at $\beta^{(t)}$.

Simplification occurs when canonical links are considered because then the derivatives of (1.7) are

$$\begin{aligned} \frac{\partial d_i w_i}{\partial \eta_i} &= -\frac{V'(\mu_i) g'(\mu_i) + V(\mu_i) g''(\mu_i)}{a_i(\phi) V(\mu_i) g'^3(\mu_i)} \\ &= -\frac{V'(\mu_i)/V(\mu_i) - V'(\mu_i)/V(\mu_i)}{a_i(\phi)/V(\mu_i)} \\ &= 0, \end{aligned}$$

since

$$g'(\mu) = \frac{1}{V(\mu)}$$

as in Remark 1.1 and

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

Thus, the resulting IWLS procedure under a canonical link model is

$$\beta^{(t+1)} = (X^T W X)^{-1} X^T W z \tag{1.11}$$

with

$$z = X\beta + D(y - \mu).$$

We get the same IWLS equation for arbitrary link models, if we use the expected instead of the observed Hessian matrix in (1.10). This approach is called Fisher Scoring and was first introduced in the context of probit analysis by Fisher (1935). Since

$$\mathbb{E}(W^*) = W \quad \text{and} \quad \mathbb{E}\left(X^T W^* X\right) = X^T W X$$

it follows that the resulting IWLS equation has the same form as in (1.11).

Furthermore, mean and variance of z are given by

$$\mathbb{E}(z) = X\beta \quad \text{and} \quad \operatorname{Var}(z) = D \operatorname{Var}(y) D^T = W^{-1}.$$

Remark 1.2 *The least squares estimate of the parameter β in the classical linear model is*

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{1.12}$$

which must not be solved iteratively since $W = I$ in this case. Moreover, (1.12) is based on known responses y and not on unknown adjusted dependent variables z.

1.4 Quasi-Likelihood Function

To find the maximum likelihood estimates for the parameters β it is essential to know the underlying distribution in order to determine the likelihood function. The purpose of this section is to show how inference can be drawn from experiments where it is impossible to construct a likelihood function, because only a relationship between mean and variance is given.

The term quasi-likelihood function was first introduced in Wedderburn (1974).

1.4.1 Introduction

For further analysis, let us assume without any loss of generality that the dispersion parameter is given by $a(\phi) = \phi$.

Definition 1.3 *Let y be a random response variable with $\mathbb{E}(y) = \mu$ and $\operatorname{Var}(y) = \phi V(\mu)$ and known variance function $V(\cdot)$. Then the quasi-likelihood function $q(y, \mu)$ is defined as*

$$q(y, \mu) = \int_y^\mu \frac{y - t}{\phi V(t)} \mathrm{d}t$$

plus a function in y. Equivalent to this definition is the following

$$\frac{\partial q(y, \mu)}{\partial \mu} = \frac{y - \mu}{\phi V(\mu)}.$$

The derivative $\partial q / \partial \mu$ is called the quasi-score function.

In the following we will have a short look on some characteristics of the quasi-likelihood function.

Theorem 1.2 *Let y be the response variable with $\mathbb{E}(y) = \mu$ and $\mathrm{Var}(y) = \phi V(\mu)$. Then the following equations hold:*

$$\mathbb{E}\left(\frac{\partial q(y,\mu)}{\partial \mu}\right) = 0 \tag{1.13}$$

and

$$\mathrm{Var}\left(\frac{\partial q(y,\mu)}{\partial \mu}\right)^2 = \mathbb{E}\left(\frac{\partial q(y,\mu)}{\partial \mu}\right)^2 = -\mathbb{E}\left(\frac{\partial^2 q(y,\mu)}{\partial \mu^2}\right) = \frac{1}{\phi V(\mu)}. \tag{1.14}$$

Proof:
Equation (1.13) is proven by

$$\mathbb{E}\left(\frac{\partial q(y,\mu)}{\partial \mu}\right) = \mathbb{E}\left(\frac{y-\mu}{\phi V(\mu)}\right) = 0$$

and equation (1.14) by

$$\begin{aligned}
\mathbb{E}\left(\frac{\partial q(y,\mu)}{\partial \mu}\right)^2 &= \mathbb{E}\left(\frac{y-\mu}{\phi V(\mu)}\right) = \frac{\mathrm{Var}(y)}{\phi^2 V^2(\mu)} = \frac{1}{\phi V(\mu)} \\
\mathbb{E}\left(\frac{\partial^2 q(y,\mu)}{\partial \mu^2}\right) &= \mathbb{E}\left(\frac{\partial}{\partial \mu}\left(\frac{y-\mu}{\phi V(\mu)}\right)\right) = \mathbb{E}\left(\frac{-\phi V(\mu) - (y-\mu)\frac{\partial}{\partial \mu}(\phi V(\mu))}{\phi^2 V^2(\mu)}\right) \\
&= \mathbb{E}\left(-\frac{1}{\phi V(\mu)} - \frac{(y-\mu)}{\phi^2 V^2(\mu)}\frac{\partial}{\partial \mu}(\phi V(\mu))\right) = -\frac{1}{\phi V(\mu)}
\end{aligned}$$

□

The following theorem proves that the log-likelihood function equals the quasi-likelihood function, if and only if the underlying distribution is from an exponential family.

Theorem 1.3 (Wedderburn, 1974) *For a response y from Definition 1.3, the log-likelihood has the following characteristic*

$$\frac{\partial l(y,\mu)}{\partial \mu} = \frac{y-\mu}{V(\mu)}, \tag{1.15}$$

if and only if the pmf or pdf of y is of the form

$$\exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right) \tag{1.16}$$

with θ being a function of μ and where ϕ is independent from μ.

Proof:
"⇒": The log-likelihood function can be integrated with respect to μ as

$$\begin{aligned} l(y,\mu) &= \int \frac{\partial l(y,\mu)}{\partial \mu}\mathrm{d}\mu = \int \frac{y-\mu}{\phi V(\mu)}\mathrm{d}\mu \\ &= \frac{y}{\phi}\int \frac{1}{V(\mu)}\mathrm{d}\mu - \frac{1}{\phi}\int \frac{\mu}{V(\mu)}\mathrm{d}\mu \\ &= \frac{y}{\phi}\int \frac{1}{b''(\theta)}\mathrm{d}\mu - \frac{1}{\phi}\int \frac{b'(\theta)}{b''(\theta)}\mathrm{d}\mu \\ &= \frac{y}{\phi}\int \frac{\partial\theta}{\partial b'(\theta)}\mathrm{d}\mu - \frac{1}{\phi}\int \frac{b'(\theta)\partial\theta}{\partial b'(\theta)}\mathrm{d}\mu \\ &= \frac{y}{\phi}\underbrace{\int \frac{\partial\theta}{\partial\mu}\mathrm{d}\mu}_{\theta} - \frac{1}{\phi}\underbrace{\int \frac{b'(\theta)\partial\theta}{\partial\mu}\mathrm{d}\mu}_{b(\theta)} \\ &= \frac{y\theta - b(\theta)}{\phi} + c(y,\phi). \end{aligned} \tag{1.17}$$

Therefore it follows that

$$f(y,\mu) = \exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right).$$

"⇐": Mean and variance of a member from the exponential family are $\mathbb{E}(y) = \mu = b'(\theta)$ and $\mathrm{Var}(y) = \phi V(\mu) = \phi b''(\theta)$. Now it follows that

$$\frac{d\mu}{d\theta} = \frac{db'(\theta)}{d\theta} = b''(\theta) = V(\mu).$$

As we know, θ is a function of μ and

$$l(y,\mu) = \left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right).$$

As a result we get

$$\begin{aligned} \frac{\partial l(y,\mu)}{\partial\mu} &= \frac{\partial}{\partial\mu}\left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right) \\ &= \frac{y}{\phi}\frac{d\theta}{d\mu} - \frac{b'(\theta)}{\phi}\frac{d\theta}{d\phi} \\ &= \frac{y}{\phi V(\mu)} - \frac{\mu}{\phi V(\mu)} \\ &= \frac{y-\mu}{\phi V(\mu)}. \end{aligned} \tag{1.18}$$

□

1.4.2 Characteristics of the Quasi-Score Function

The quasi-score function with $\mathbb{E}(y) = \mu$,$\text{Var}(y) = \phi V(\mu)$, $g(\mu) = x^T\beta$ and $\mu = \mu(\beta)$ is

$$\frac{\partial q(y,\mu(\beta))}{\partial \beta_j} = \frac{\partial q(y,\mu(\beta))}{\partial \mu}\frac{\partial \mu}{\partial \beta_j} = \frac{y-\mu}{\phi V(\mu)}\frac{\partial \mu}{\partial \beta_j}$$

with

$$\begin{aligned}
\mathbb{E}\left(\frac{\partial q(y,\mu(\beta))}{\partial \beta_j}\right) &= \mathbb{E}\left(\frac{\partial q(y,\mu(\beta))}{\partial \mu}\frac{\partial \mu}{\partial \beta_j}\right) = \mathbb{E}\left(\frac{y-\mu}{\phi V(\mu)}\frac{\partial \mu}{\partial \beta_j}\right) = 0,\\
\mathbb{E}\left(\frac{\partial^2 q(y,\mu(\beta))}{\partial \beta_j \partial \beta_k}\right) &= \mathbb{E}\left(\frac{\partial}{\partial \beta_j}\left(\frac{\partial q(y,\mu(\beta))}{\partial \mu}\frac{\partial \mu}{\partial \beta_k}\right)\right)\\
&= \mathbb{E}\left(\frac{\partial^2 q(y,\mu(\beta))}{\partial \mu^2}\frac{\partial \mu}{\partial \beta_j}\frac{\partial \mu}{\partial \beta_k} + \frac{\partial q(y,\mu(\beta))}{\partial \mu}\frac{\partial^2 \mu}{\partial \beta_j \partial \beta_k}\right)\\
&= -\frac{1}{\phi V(\mu)}\frac{\partial \mu}{\partial \beta_j}\frac{\partial \mu}{\partial \beta_k},\\
\mathbb{E}\left(\frac{\partial q(y,\mu(\beta))}{\partial \beta_j}\frac{\partial q(y,\mu(\beta))}{\partial \beta_k}\right) &= \mathbb{E}\left(\left(\frac{y-\mu}{\phi V(\mu)}\right)^2\frac{\partial \mu}{\partial \beta_j}\frac{\partial \mu}{\partial \beta_k}\right)\\
&= \frac{1}{\phi^2 V^2(\mu)}\text{Var}(y-\mu)\frac{\partial \mu}{\partial \beta_j}\frac{\partial \mu}{\partial \beta_k}\\
&= \frac{1}{\phi V(\mu)}\frac{\partial \mu}{\beta_j}\frac{\partial \mu}{\partial \beta_k}\\
&= -\mathbb{E}\left(\frac{\partial^2 q(y,\mu(\beta))}{\partial \beta_j \partial \beta_k}\right). \qquad (1.19)
\end{aligned}$$

Comparison of the quasi-score function to the score function in Theorem (1.1) yields equal results and relationships for the expected quasi-score.

1.4.3 Maximum Quasi-Likelihood Estimation

Suppose we have n independent responses y_i, $i = 1,\ldots,n$, then the resulting quasi-score vector u is given by

$$u(\beta) = \sum_{i=1}^{n}\frac{\partial q(y_i,\mu_i(\beta))}{\partial \beta} = \sum_{i=1}^{n}\frac{\partial q(y_i,\mu_i(\beta))}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta}. \qquad (1.20)$$

By setting

$$c_i = \frac{\partial \mu_i}{\partial \beta},$$

equation (1.20) can be written in matrix notation as

$$u(\beta) = C^T V^{-1}(y-\mu)/\phi.$$

The maximum quasi-likelihood estimator $\hat{\beta}$ is achieved by solving

$$u(\hat{\beta}) = 0$$

with $u(\beta)$ being the quasi-score function. Moreover, mean and variance of u can easily be calculated as

$$\mathbb{E}(u(\beta)) = \mathbb{E}(C^T V^{-1}(y-\mu)/\phi) = 0$$

and with the characteristics of the quasi-score function from Subsection 1.4.2 it follows that

$$\begin{aligned}
\operatorname{Var}(u(\beta)) &= \operatorname{Var}\left(\phi^{-1} C^T V^{-1}(y-\mu)\right) \\
&= \phi^{-1} C^T V^{-1} \operatorname{Var}(y-\mu)(\phi^{-1} C^T V^{-1})^T \\
&= \phi^{-1} C^T V^{-1} C \\
&\overset{(1.19)}{=} -\mathbb{E}\left(\frac{\partial u(\beta)}{\partial \beta}\right).
\end{aligned}$$

A Taylor series expansion of the quasi-score in β around $\hat{\beta}$ yields the following approximation

$$0 = u(\hat{\beta}) \approx u(\beta) + \frac{\partial u(\beta)}{\partial \beta}(\hat{\beta} - \beta).$$

For ordinary likelihood estimation we used the Fisher-Scoring technique, so using the mean of the Hessian instead of the observed Hessian itself. If we use this approximation here, we can approximate $\frac{\partial u(\beta)}{\partial \beta}$ by the negative mean $-\phi^{-1} C^T V^{-1} C$. This results in

$$\hat{\beta} - \beta \approx \phi \left(C^T V^{-1} C\right)^{-1} u(\beta) = \left(C^T V^{-1} C\right)^{-1} C^T V^{-1}(y-\mu). \tag{1.21}$$

This is one step of a weighted least squares regression of the residuals $(y-\mu)$ on C with weights V^{-1}. If μ is close enough to y, this method converges and gives $\hat{\beta}$.

Moreover, the first two moments of the maximum quasi-likelihood estimator $\hat{\beta}$ are

$$\begin{aligned}
\mathbb{E}(\hat{\beta}) &\approx \beta \\
\operatorname{Var}(\hat{\beta}) &\approx \operatorname{Var}\left(\phi \left(C^T V^{-1} C\right)^{-1} u(\beta)\right) \\
&= \phi^2 \left(C^T V^{-1} C\right)^{-1} \operatorname{Var}(u(\beta)) \left(C^T V^{-1} C\right)^{-1} \\
&= \phi \left(C^T V^{-1} C\right)^{-1}.
\end{aligned}$$

An important property of this method is that the estimation of β does not depend on the value of ϕ. To estimate ϕ, the mean Pearson estimator is used

$$\hat{\phi} = \frac{1}{n-p} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)} = \frac{1}{n-p} \hat{X}^2,$$

where X^2 is the Pearson X^2 statistic.

1.5 Measuring the Goodness of Fit

The process of fitting a model to the data can be described as a way to replace the real data values y by the fitted values $\hat{\mu}$ from the model under investigation. In general the fitted values will not equal the real values exactly, therefore a measure of how discrepant they are is required. Two such measures are in common use. The first one we consider is the likelihood ratio test statistic, also called deviance, and the second is the Pearson X^2 statistic.

1.5.1 Deviance

The scaled deviance takes the form

$$\frac{1}{\phi} D(y, \hat{\mu}) = -2 \left(l(y, \hat{\mu}) - l(y, y) \right), \tag{1.22}$$

where $l(y, y)$ describes the full or saturated model, which gives us a measure of how well any model could possibly fit. In the saturated model we need to use n parameters for n data points, so one per observation, and the $\hat{\mu}$s derived from this model match the data exactly. In practice, the full model is uninformative, as the data is not summarized, but fully repeated. However, the full model gives us a baseline for measuring the discrepancy for any other model with p parameters. The model under investigation is given by $l(y, \hat{\mu})$. Furthermore, the minimal scaled deviance is achieved for the maximum likelihood estimates.

As we will show below, for normally distributed responses the deviance is exactly χ^2 distributed with $n - p$ degrees of freedom (df), whereas for other distributions of the exponential family only asymptotic results are available.

Example 1.3 (Deviance for Gaussian data) *Let y_i be independent normally distributed with mean μ_i and constant variance σ^2. Then the log-likelihoods for the model under consideration and the saturated model are given by*

$$\begin{aligned} l(y, \mu) &= \sum_{i=1}^{n} \left(-\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_i - \mu_i)^2}{2\sigma^2} \right) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2} \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{\sigma^2} \\ l(y, y) &= -\frac{n}{2} \log(2\pi\sigma^2) \end{aligned}$$

Therefore, the scaled deviance is

$$\frac{1}{\phi} D(y, \hat{\mu}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2 = \frac{1}{\sigma^2} SSE(\hat{\beta}),$$

which has a χ^2_{n-p} distribution with mean $n-p$.

As in the following chapter we often need deviances of Poisson and binomial distributed data, we give a short overview of how to achieve those two.

Example 1.4 (Deviance for binomial data) *Let $m_i y_i \overset{ind}{\sim} B(m_i, \mu_i)$ with $y_i = 0, 1/m_i, 2/m_i, \ldots, 1$. Then the log-likelihood for this model is given by (see also 1.2)*

$$\log f(y,\mu) = \sum_{i=1}^{n} \left(m_i y_i \log \frac{\mu_i}{1-\mu_i} - m_i \log \frac{1}{1-\mu_i} + \log \binom{m_i}{m_i y_i} \right)$$

and the full model is

$$\log f(y,y) = \sum_{i=1}^{n} \left(m_i y_i \log \frac{y_i}{1-y_i} - m_i \log \frac{1}{1-y_i} + \log \binom{m_i}{m_i y_i} \right).$$

Thus, the resulting scaled deviance with $\phi = 1$ and $a_i(\phi) = 1/m_i$ is

$$\begin{aligned}
\frac{1}{\phi} D(y,\hat{\mu}) &= -2 \sum_{i=1}^{n} \left(m_i y_i \log \frac{\hat{\mu}_i}{1-\hat{\mu}_i} - m_i \log \frac{1}{1-\hat{\mu}_i} - m_i y_i \log \frac{y_i}{1-y_i} + m_i \log \frac{1}{1-y_i} \right) \\
&= -2 \sum_{i=1}^{n} \left(m_i y_i \left(\log \frac{\hat{\mu}_i}{y_i} + \log \frac{1-y_i}{1-\hat{\mu}_i} \right) - m_i \log \frac{1-y_i}{1-\hat{\mu}_i} \right) \\
&= 2 \sum_{i=1}^{n} m_i \left((1-y_i) \log \frac{1-y_i}{1-\hat{\mu}_i} + y_i \log \frac{y_i}{\hat{\mu}_i} \right).
\end{aligned}$$

Example 1.5 (Deviance for Poisson data) *Let $y_i \overset{ind}{\sim}$ Poisson(μ_i) with $y_i \geq 0$. Then the log-likelihood for this model is given by*

$$\log f(y,\mu) = \sum_{i=1}^{n} (y_i \log \mu_i - \mu_i - \log y_i!)$$

and the saturated model is

$$\log f(y,y) = \sum_{i=1}^{n} (y_i \log y_i - y_i - \log y_i!).$$

The resulting deviance with $\phi = 1$ and $a_i(\phi) = \phi$ is therefore

$$\begin{aligned}
\frac{1}{\phi} D(y,\hat{\mu}) &= -2 \sum_{i=1}^{n} (y_i \log \hat{\mu}_i - \hat{\mu}_i - y_i \log y_i + y_i) \\
&= 2 \sum_{i=1}^{n} (y_i (\log y_i - \log \hat{\mu}_i) - (y_i - \hat{\mu}_i)) \\
&= 2 \sum_{i=1}^{n} \left(y_i \log \frac{y_i}{\hat{\mu}_i} - (y_i - \hat{\mu}_i) \right).
\end{aligned}$$

Since for intercept models $\sum_{i=1}^{n} (y_i - \hat{\mu}_i) = 0$, we only use its relevant part $2 \sum_{i=1}^{n} y_i \log y_i/\hat{\mu}_i$.

1.5.2 Pearson X^2 Statistic

As mentioned above, the other important measure of discrepancy is the Pearson X^2 statistic, which is of the form

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

with $V(\hat{\mu}_i)$ being the estimated variance function.

Both, the deviance and the Pearson X^2 statistic follow a χ^2 distribution with $n-p$ degrees of freedom for normally distributed responses. For other distributions we can rely just on asymptotic results. A general advantage of the deviance is that it is additive for nested sets of models, leading to likelihood ratio tests.

1.5.3 Quasi-Deviance

By analogy to (1.22) we define the quasi-deviance function as

$$D(y,\mu) = -2\phi\left(q(y,\mu) - q(y,y)\right) = -2\sum_{i=1}^{n}\int_{y_i}^{\mu_i} \frac{y_i - t}{V(t)}\mathrm{d}t,$$

which is strictly positive except at $y = \mu$ and minimal for the maximum quasi-likelihood estimation.

Example 1.6 (Incidence of leaf-blotch on barley) *McCullagh and Nelder (1989) gave an example concerning the incidence of leaf blotch in* 1965 *on* 10 *varieties of barley grown at nine sites. The response variable is the percentage of leaf area affected. They first suggested to fit the data as a pseudo-binomial model with a variance of* $\phi\mu(1-\mu)$*. However, this variance function does not satisfy the variation of the data. Therefore, following Wedderburn's suggestion with a variance function of the form* $\mu^2(1-\mu)^2$*, this yields the following quasi-likelihood function, with* $0 < \mu < 1$ *and* $0 \le y \le 1$

$$q(y,\mu) = \int^{\mu} \frac{y-t}{t^2(1-t)^2}\mathrm{d}t = (2y-1)\log\frac{\mu}{1-\mu} - \frac{y}{\mu} - \frac{1-y}{1-\mu}$$

and

$$q(y,y) = (2y-1)\log\frac{y}{1-y} - 2.$$

The resulting quasi-deviance is therefore given by

$$\begin{aligned} D(y,\mu) &= -2\phi\sum_{i=1}^{n}\left((2y_i-1)\left(\log\frac{1-y_i}{1-\mu_i} - \log\frac{y_i}{\mu_i}\right) + 2 - \frac{y_i}{\mu_i} - \frac{1-y_i}{1-\mu_i}\right) \\ &= -2\phi\sum_{i=1}^{n}\left((2y_i-1)\left(\log\frac{1-y_i}{1-\mu_i} - \log\frac{y_i}{\mu_i}\right) + (2\mu_i-1)\frac{y_i-\mu_i}{\mu_i(1-\mu_i)}\right). \end{aligned}$$

Unfortunately, this function is not defined for $y_i = 1$ *or* $y_i = 0$*, because then the terms* $(2y_i - 1) \log \frac{y_i}{\mu_i}$ *resp.* $(2y_i - 1) \log \frac{1-y_i}{1-\mu_i}$ *do not exist. However, the maximum quasi-likelihood estimate* $\hat{\beta}$ *exits nevertheless since by setting the quasi-score equal to zero, no difficulty occurs.*

Chapter 2

Simple Overdispersion Models

In the previous chapter we discussed generalized linear models but in practice, generalized linear models often do not fit or represent the data adequately. This can be due to several reasons: The distribution of y may not be a member of the exponential family or the fitted regression model may not be appropriate. A simple way of expressing this problem is discussed in this chapter. Section 2.1 introduces random effect models and one simple overdispersion model, namely the beta binomial model, is discussed in Section 2.4. In addition a R-function will be discussed to model beta binomial data. In Section 2.6 the negative binomial distribution will be considered to model overdispersed count data and a R-function for negative binomial distributed data will be discussed. If only a mean-variance relationship is specified, Section 2.8 and Section 2.10 discuss two approaches to model data with this property.

2.1 Random Effect Models

In Chapter 1 we considered linear predictors of the form

$$\eta_i = g(\mu_i) = x_i^T \beta, \quad i = 1, \ldots, n,$$

where the vector $x_i = (x_{i1}, \ldots, x_{ip})^T$ denotes a p-dimensional vector of given explanatory variables, corresponding to the ith response y_i, $i = 1, \ldots, n$. The vectors x_i form the $n \times p$ design matrix X, and $\beta = (\beta_1, \ldots, \beta_p)^T$ are the unknown parameters which are subject to estimation. We will now discuss an extension of the previous assumptions, where we include a random effect into the linear predictor.

This has the followings reasons: Suppose that in the model we have estimated there are one or more important explanatory variables missing. Would those explanatory variables

have been recorded, we would add them into the linear predictor. In reality we even don't know what predictor variables should have been recorded to construct an adequate model.

A good way of expressing this problem is to add random variables into the linear predictor. Assuming that there is another vector of unobserved variables $u_i = (u_{i1}, \ldots, u_{ip'})^T$ in addition to the observed predictor x_i leads to the following model

$$\eta_i = g(\mu_i) = x_i^T\beta + u_i^T\gamma, \quad i = 1, \ldots, n, \tag{2.1}$$

where $\gamma = (\gamma_1, \ldots \gamma_{p'})^T$ is the p'-dimensional unknown parameter vector for the unobserved predictor variables u_i. Since we know nothing about u_i and γ , the term $u_i^T\gamma$ is in fact considered as a single random variable. So we can write with $z_i = u_i^T\gamma$

$$\eta_i = g(\mu_i) = x_i^T\beta + z_i.$$

Another reason for using random variables is that the data might be grouped. Sometimes the grouping structure is simple, where each case belongs to a single group and there is only one grouping factor. But in reality the grouping structure is rather complicated which for example implicates the correlation of observations within the same group. Therefore, a model which assumes independence of the observations is inappropriate. Suppose we have n independent groups of observations $y_i = (y_{i1}, \ldots, y_{in_i})^T$ with correlation between the y_{ij}s in group $i = 1, \ldots, n$, $j = 1, \ldots, n_i$. If we include a random effect for each group into the linear predictor,

$$\eta_{ij} = g(\mu_{ij}) = x_{ij}^T\beta + z_i$$

we get a model for the correlated data.

2.2 Conjugate Distributions

As we will use the term conjugate distribution in the following section, we need to define a priori, a posteriori and conjugate distributions first. This section is based on Casella and Berger (2002).

Theorem 2.1 (Bayes Theorem) *Let $A \subset \Omega$ be an event and $B_1, B_2, \ldots, B_k$ a decomposition of the sample space Ω with $P(B_j) > 0$, $j = 1, \ldots, k$. Then*

$$P(B_j|A) = \frac{P(A|B_j)P(B_j)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}.$$

There exists an analogue formula for the pdf in the continuous case.

Theorem 2.2 *Let the pdf of an observation (y, z) be $f(y, z)$. Then*

$$f(z|y) = \frac{f(y|z)f(z)}{\int_{-\infty}^{\infty} f(y|z)f(z)\mathrm{d}z}.$$

The basic idea of Bayes Theorem is that before doing an experiment one has a certain "a priori" idea of the underlying value of z. This idea of the value can be described as an a priori density function $f(z)$. Then Bayes Theorem can be used to determine the a posteriori distribution $f(z|y)$ after the experiment.

For determining the a posteriori distribution, one has to solve the integral $\int_{-\infty}^{\infty} f(y|z)f(z)\mathrm{d}z$ which could be hard to evaluate. If we use a conjugate distribution for z, the integral can be solved analytically.

Definition 2.1 (Conjugate distribution) *The distribution of $f(z)$ is called conjugate distribution for $f(y|z)$ if $f(z|y)$ and $f(z)$ are of the same form, that means that the a priori- and the a posteriori distributions belong to the same family of distributions.*

Remark 2.1 *Some important distributions and their corresponding conjugate distributions are given in Table 2.1.*

distribution	**conjugate distribution**
Gaussian	Gaussian
gamma	gamma
Poisson	gamma
binomial	beta

Table 2.1: Distributions and corresponding conjugate distributions for two continuous and two discrete distributions.

2.3 Maximum Likelihood Estimation

Let the unobserved random variables z_i be independent and identically distributed. Compared to the mean $\mu_i = \mathbb{E}(y_i)$ in the generalized linear model we have to consider now the conditional mean

$$\mu_i = \mathbb{E}(y_i|z_i)$$

or

$$\mu_{ij} = \mathbb{E}(y_{ij}|z_i).$$

In addition we assume that the conditional distribution of the response, given the unobserved variables, is a member of the exponential family,

$$y_i|z_i \sim \text{Exponential family} \quad \text{or} \quad y_{ij}|z_i \sim \text{Exponential family}.$$

For the calculation of the maximum likelihood estimator we need the marginal distribution of y.

Suppose that the pdf of an observation (y, z) is $f(y, z, \theta)$ where θ is the vector of all unknown parameters. Then the marginal distribution of y is $f(y, \theta)$ given by

$$f(y, \theta) = \int_{\mathbb{R}} f(y, z, \theta) \mathrm{d}z.$$

Furthermore, the marginal log-likelihood function $l(y, \theta)$ is

$$l(y, \theta) = \log f(y, \theta) = \log \int_{\mathbb{R}} f(y, z, \theta) \mathrm{d}z.$$

If we use the relationship

$$f(y, z, \theta) = f(y|z, \theta) f(z, \theta),$$

the integral gets

$$l(y, \theta) = \log f(y, \theta) = \log \int_{\mathbb{R}} f(y|z, \theta) f(z, \theta) \mathrm{d}z.$$

To estimate the parameter vector θ one has to maximize the marginal log-likelihood function $l(y, \theta)$. In reality solving this integral often causes difficulties. Therefore one approach to evaluate this integral is to approximate it (see next chapter). The other opportunity is to use conjugate distributions for the distribution of z, because the integral can then be evaluated explicitly.

2.4 The Beta Binomial Distribution

Suppose one observes binomial distributed frequencies, where the probability of success is not known or random. Then one opportunity is to assume a beta distribution for the

probability of success. For example we might consider using the beta binomial distribution to model the number of cars that crash in n races. The probability for a crash is varying since the skills of every driver differs.

In this section we will have a short look at the beta binomial distribution and how to use the beta distribution in terms of the conjugate approach.

Definition 2.2 (Beta-function) *The beta-function for two positive integers $a > 0$ and $b > 0$ is defined by*

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\mathrm{d}t.$$

The following lemma gives us one way of using a simple conjugate distribution. As for this distribution, the marginal distribution of y can easily be achieved, see also Friedl (1991).

Lemma 2.1 *Let z be independent beta distributed with pdf*

$$f(z,p) = \frac{1}{B(a,b)}p^{a-1}(1-p)^{b-1} \quad \textit{with} \quad 0 < p < 1;\ a,b > 0.$$

Then

$$\mathbb{E}(z) = \frac{a}{a+b} = \mu,$$

$$\mathrm{Var}(z) = \frac{ab}{(a+b)^2(a+b+1)} = \frac{\mu(1-\mu)}{a+b+1} = \phi\mu(1-\mu),$$

where $\phi = 1/(a+b+1)$.
In addition let $y|z \sim Binomial(m,z)$ with

$$P(y=k|z=p) = \binom{m}{k}p^k(1-p)^{m-k}, \qquad k = 0,1,\ldots,m.$$

Then the marginal distribution of y is the beta binomial distribution,

$$P(y=k) = \binom{m}{k}\frac{B(a+k,m+b-k)}{B(a,b)} \quad \textit{with} \quad k = 0,1,\ldots,m; \quad a,b > 0,$$

and

$$\mathbb{E}(y) = m\mu,$$

$$\mathrm{Var}(y) = m\mu(1-\mu)(1+\phi(m-1)).$$

Proof:
The marginal probability function of y is given by

$$\begin{aligned} P(y=k) &= \int_0^1 P(y=k|z=p)P(z=p)\mathrm{d}p \\ &= \int_0^1 \frac{p^{a-1}(1-p)^{b-1}}{B(a,b)}\binom{m}{k}p^k(1-p)^{m-k}\mathrm{d}p \\ &= \binom{m}{k}\frac{1}{B(a,b)}\int_0^1 p^{a-1+k}(1-p)^{b-1+m-k}\mathrm{d}p \\ &= \binom{m}{k}\frac{B(a+k,m+b-k)}{B(a,b)}. \end{aligned}$$

The last identity results from applying Definition 2.3. Moreover, we know

$$\mathbb{E}(y) = \mathbb{E}(\mathbb{E}(y|z)) = m\mathbb{E}(z) = m\frac{a}{a+b} = m\mu$$

and since $\mathbb{E}(z^2) = \mathrm{Var}(z) + \mathbb{E}^2(z) = \phi\mu(1-\mu) + \mu^2$ we further get

$$\begin{aligned} \mathrm{Var}(y) &= \mathbb{E}(\mathrm{Var}(y|z)) + \mathrm{Var}(\mathbb{E}(y|z)) \\ &= \mathbb{E}(mz(1-z)) + \mathrm{Var}(mz) \\ &= m(\mathbb{E}(z) - \mathbb{E}(z^2)) + m^2\,\mathrm{Var}(z) \\ &= m(\mu - (\phi\mu(1-\mu)+\mu^2)) + m^2\phi\mu(1-\mu) \\ &= m\mu(1-\phi(1-\mu)-\mu+m\phi(1-\mu)) \\ &= m\mu(1-\mu+\phi(1-\mu)(m-1)) \\ &= m\mu(1-\mu)(1+\phi(m-1)). \end{aligned}$$

□

Remark 2.2 *The parameter ϕ is often referred to as the dispersion parameter and the term $(1+\phi(m-1))$ as overdispersion. If we have binary responses, then $m=1$ and no overdispersion is possible under this model.*

Crowder (1978) analyzed the `orob2` dataset (see Table 2.2) where he considered the group specific dispersion parameter ϕ_i to be a global dispersion parameter ϕ for illustration purposes. For a proper calculation Griffiths (1973) also used the beta binomial distribution for the incidence of noninfectious disease in households. He considered a truncated beta binomial distribution, because cases of no disease were not included in the data.

In the example below we assume a generalized linear model and consider the predictor to be

$$\eta = \mathrm{logit}(\mu) = x^T\beta.$$

2.5 R-Function for the Beta Binomial Distribution

In this section we will look at one R-function which can handle beta binomial distributions. The one considered here is called `betabin`, provided by the package `aod` (see also Lesnoff and Lancelot, 2010).

The function `betabin` fits a beta binomial generalized linear model accounting for overdispersion in clustered binomial data. The explicit call of the function is described in Appendix A.1.1.

First we will consider a short example, given in the package `aod`, to see how the function `betabin` works. The experiment is about comparing 2 types of seeds and 2 types of root extracts. There are 5 to 6 replicates in each of the 4 treatment groups, so altogether there are 21 observations. Each replicate comprises a number of seeds exposed to germination and a number of seeds which actually terminated. Seed is a factor with 2 levels, namely `O73` and `O75`, and root is a factor with levels `BEAN` and `CUCUMBER`.

BEAN				CUCUMBER			
O75		O73		O75		O73	
y	n	y	n	y	n	y	n
10	39	8	16	5	6	3	12
23	62	10	30	53	74	22	41
23	81	8	28	55	72	15	30
26	51	23	45	32	51	32	51
17	39	0	4	46	79	3	7
				10	13		

Table 2.2: `orob2`: Data describing the germination for seed

2.5.1 Logistic Regression Model

A simple logistic regression model for the data given in Table 2.2

```
> library(aod)
> data(orob2)
> attach(orob2)
> m.glm.binomial<-glm(cbind(y,n-y)~seed*root, family=binomial)
```

yields the following summary

```
> summary(m.glm.binomial)

  Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
  (Intercept)          -0.4122     0.1842  -2.238   0.0252 *
  seed075              -0.1459     0.2232  -0.654   0.5132
  rootCUCUMBER          0.5401     0.2498   2.162   0.0306 *
  seed075:rootCUCUMBER  0.7781     0.3064   2.539   0.0111 *

       Null deviance: 98.719  on 20  degrees of freedom
    Residual deviance: 33.278  on 17  degrees of freedom
    AIC: 117.87
```

Now we see that the residual deviance from model `m.glm.binomial` is quite large (33.23 compared to 17 df), suggesting that other relevant factors are varying over the data. We now fit the same data with a beta binomial model, assuming that the random effects z_i follow a beta distribution.

2.5.2 Beta Binomial Model

A big advantage of the function `betabin` from the package `aod` is that the user can either fix the parameter ϕ to a constant or the parameter ϕ will be estimated from the sample. Moreover one can also specify the random argument either as a global dispersion parameter (`random=~1`) or as a specific dispersion parameter for the levels of a given group factor (`random=~group`). Let us consider the following two models:

```
> m.betabin.1<-betabin(cbind(y,n-y)~seed*root, random=~1)
> m.betabin.seed<-betabin(cbind(y,n-y)~seed*root, random=~seed)
```

Then `m.betabin.1` is a model with a global dispersion parameter and `m.betabin.seed` has a specific dispersion parameter for each `seed` group. The output of `m.betabin.1` and `m.betabin.seed` is as follows.

```
> m.betabin.1
  Fixed-effect coefficients:
                        Estimate Std. Error    z value Pr(> |z|)
  (Intercept)          -4.456e-01  2.183e-01 -2.041e+00 4.124e-02
  seed075              -9.612e-02  2.737e-01 -3.512e-01 7.255e-01
  rootCUCUMBER          5.235e-01  2.968e-01  1.764e+00 7.780e-02
  seed075:rootCUCUMBER  7.962e-01  3.779e-01  2.107e+00 3.514e-02
```

```
  Overdispersion coefficients:
                  Estimate Std. Error   z value   Pr(> z)
  phi.(Intercept) 1.236e-02  1.131e-02 1.093e+00 1.373e-01

  Log-likelihood statistics
     Log-lik      nbpar    df res.   Deviance        AIC       AICc
  -5.377e+01          5         16  3.094e+01  1.175e+02  1.215e+02
```

Two different hypotheses H_0 and H_1 specify values for ϕ, namely

$$\begin{aligned} H_0 &: \phi = 0 \\ H_1 &: \phi \neq 0. \end{aligned}$$

By accepting H_0, a binomial distribution is supported, whereas by rejecting H_0 a beta binomial distribution is supported. A p-value of 0.1373 provides evidence that no global dispersion parameter is needed.

```
> m.betabin.seed
  Fixed-effect coefficients:
                        Estimate Std. Error    z value Pr(> |z|)
  (Intercept)          -4.391e-01  2.162e-01 -2.031e+00 4.224e-02
  seed075              -1.025e-01  2.725e-01 -3.763e-01 7.067e-01
  rootCUCUMBER          5.255e-01  2.893e-01  1.816e+00 6.930e-02
  seed075:rootCUCUMBER  7.949e-01  3.739e-01  2.126e+00 3.352e-02

  Overdispersion coefficients:
              Estimate Std. Error   z value   Pr(> z)
  phi.seed073 9.782e-03  2.221e-02 4.404e-01 3.298e-01
  phi.seed075 1.313e-02  1.334e-02 9.844e-01 1.625e-01

  Log-likelihood statistics
     Log-lik      nbpar    df res.   Deviance        AIC
  -5.376e+01          6         15  3.092e+01  1.195e+02
```

An analysis of variance (ANOVA) yields the following result:

```
> anova(m.betabin.1,m.betabin.seed)

                 logL  k   AIC  Res.dev. Res.Df
                 Deviance Df P(> Chi2)
  m.betabin.1    -53.77 5 117.5     30.94     16
  m.betabin.seed -53.76 6 119.5     30.92     15   0.01585  1    0.8998
```

One important conclusion from the ANOVA table is that there is strong evidence that there is no need of group specific parameters, as the χ^2-test with a p-value of 0.8998 shows.

2.5.3 Deviance

The deviance concerning the function `betabin` is calculated as

$$D(y, \mu) = -2(\log f(y, \mu) - \log f(y, y)) \tag{2.2}$$

with $f(y, y)$ being the full or saturated model. The log-likelihood for the saturated model is taken from a binomial distribution, because the value of $f(y, y)$ under the beta binomial assumption is getting maximal for $\phi \to 0$ and the binomial distribution approximates the beta binomial distribution arbitrarily well for small ϕ ($\phi \to 0$). Note that the log-likelihoods in (2.2) comprises the dispersion.

To derive the log-likelihood of the saturated model, we consider

```
> sum(dbinom(y,n,y/n,log=TRUE))
  [1] -38.29813.
```

To check the log-likelihood of the model under consideration we notice that for a beta binomial distribution we further need two parameters, a and b, because the probability function is

$$P(y_i = k) = \binom{m_i}{k} \frac{B(a_i + k, m_i + b_i - k)}{B(a_i, b_i)}.$$

`m.betabin.1` delivers us an estimate of $\phi_i = 1/(a_i+b_i+1)$ and $\mu_i = a_i/(a_i+b_i)$. Therefore

$$\begin{aligned} a_i &= \frac{\mu_i - \phi_i\mu_i}{\phi_i} = \mu_i \frac{1-\phi_i}{\phi_i} \\ b_i &= \frac{1-\phi_i}{\phi_i} - \frac{\mu_i - \phi_i\mu_i}{\phi_i} = \frac{1-\phi_i}{\phi_i} - a_i. \end{aligned}$$

For `m.betabin.1` we assumed a global dispersion parameter so $\phi_i = \phi, i = 1, \ldots, n$. For calling the function `dbetabin` from package `VGAM`, which calculates the value of the pmf from beta binomial distributed data, we first have to define the vectors `mu1` and `phi1`. With

```
> mu1<-fitted(m.betabin.1)
> phi1<-m.betabin.1@random.param
```

the function `dbetabin` gives the value of the pmf:

```
> sum(dbetabin(y, n, mu1, phi1, log=TRUE))
[1] -53.76678
```

which is the same as the log-likelihood reported in `m.betabin.1`. Therefore

$$D(y,\mu) = -2(\log f(y,\mu) - \log f(y,y)) = -2(-53.77 + 38.298) = 30.94.$$

For deriving the log-likelihood from model `m.betabin.seed`, we first have to define new vectors `mu2` and `phi2`. Important to note is that in the data set there are 11 observations of seed `O73` and 10 of seed `O75`.

```
> mu2<-fitted(m.betabin.seed)
> phi2<-c(rep(m.betabin.seed@random.param[2],11),
          rep(m.betabin.seed2@random.param[1],10))
```

Those estimates yield to the log-likelihood value in `m.betabin.seed`:

```
> sum(dbetabin(y, n, mu2, phi2, log=TRUE))
  [1] -53.75885
```

Therefore, the deviance is calculated as $-2(-53.76 + 38.298) = 30.92$.

To compare the parameter estimates, Table 2.3 shows the parameter estimates and their standard errors for the three different models. The parameter estimates under both beta binomial models are almost the same as in the logistic regression case, but the standard errors are slightly larger.
Further investigation for this example is given in Crowder (1978).

2.6 The Negative Binomial Distribution

The negative binomial distribution describes the distribution of the number of trials, which are necessary to determine a given number of successes. It arises from the Poisson(λ) distribution, where the parameter λ follows a gamma distribution.

Definition 2.3 *The gamma function is defined for $a > 0$ as*

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} \mathrm{d}t.$$

The following lemma shows us how to use the gamma distribution as a conjugate distribution to the Poisson distribution.

	m.glm.binomial		m.betabin.1		m.betabin.seed	
Intercept	−0.4122	(0.1842)	−0.4456	(0.2183)	−0.4391	(0.2162)
seed075	−0.1459	(0.2232)	−0.0961	(0.2737)	−0.1025	(0.2893)
rootCUCUMBER	0.5401	(0.2498)	0.5235	(0.2968)	0.5255	(0.2893)
seed075:rootCUCUMBER	0.7781	(0.3064)	0.7962	(0.3779)	0.7949	(0.3739)
phi			0.0123	(0.0113)		
phi.seed073					0.0098	(0.0222)
phi.seed075					0.0131	(0.0133)
res.dev	33.28		30.94		30.92	
df	17		16		15	

Table 2.3: orob2 data: Parameter estimates and deviances under three different models: m.glm.binomial, m.betabin.1 and m.betabin.seed.

Lemma 2.2 *Let z be gamma distributed with density function*

$$f(z,a,b) = \frac{1}{\Gamma(a)b^a} z^{a-1} e^{-z/b}, \quad \textit{with} \quad a,b>0.$$

Then

$$\mathbb{E}(z) = ab,$$
$$\mathrm{Var}(z) = ab^2.$$

In addition let $y|z \sim Poisson(z)$ with

$$P(y=k|z=z) = \frac{e^{-z}z^k}{k!} \quad k=0,1,\ldots.$$

Then the marginal pmf of y is the negative binomial pmf, defined by

$$P(y=k) = \frac{\Gamma(a+k)}{\Gamma(a)k!}\left(\frac{\mu}{\mu+a}\right)^k\left(\frac{a}{\mu+a}\right)^a \quad \textit{with} \quad k=0,1,\ldots; \quad a,b>0,$$

and

$$\begin{aligned}\mathbb{E}(y) &= ab = \mu,\\ \mathrm{Var}(y) &= ab+ab^2 = \mu+\frac{1}{a}\mu^2 = \mu+\phi\mu^2 \quad \textit{where} \quad \phi = 1/a.\end{aligned}$$

Proof:
The marginal pmf of y is given by

$$\begin{aligned}P(y=k) &= \int_0^\infty P(y=k|z=z)P(z=z)\mathrm{d}z\\ &= \int_0^\infty \frac{e^{-z}z^k}{k!}\frac{1}{\Gamma(a)b^a}z^{a-1}e^{-z/b}\mathrm{d}z\\ &= \frac{1}{k!\Gamma(a)b^a}\int_0^\infty z^{a+k-1}e^{-z(1+1/b)}\mathrm{d}z.\end{aligned}$$

Substituting $w = z(1 + 1/b) = z\frac{b+1}{b}$, therefore $dw = dz(1 + 1/b)$, further yields

$$\begin{aligned} P(y = k) &= \frac{1}{k!\Gamma(a)b^a} \int_0^\infty \left(\frac{b}{b+1} w\right)^{a+k-1} e^{-w} \frac{b}{b+1} \mathrm{d}w \\ &= \frac{1}{k!\Gamma(a)b^a} \int_0^\infty \left(\frac{b}{b+1}\right)^{a+k} w^{a+k-1} e^{-w} \mathrm{d}w \\ &= \frac{1}{k!\Gamma(a)b^a} \Gamma(a+k) \left(\frac{b}{b+1}\right)^{a+k} \\ &= \frac{\Gamma(a+k)}{k!\Gamma(a)} \left(\frac{1}{b+1}\right)^a \left(\frac{b}{b+1}\right)^k. \end{aligned}$$

Reparametrize this by using $b = \mu/a$ gives

$$P(y = k) = \frac{\Gamma(a+k)}{\Gamma(a)k!} \left(\frac{\mu}{\mu+a}\right)^k \left(\frac{a}{\mu+a}\right)^a. \tag{2.3}$$

In the special case when a is an integer, equation (2.3) can be written as

$$P(y = k) = \binom{a+k-1}{k} \left(\frac{\mu}{\mu+a}\right)^k \left(\frac{a}{a+\mu}\right)^a.$$

Mean and variance are:

$$\mathbb{E}(y) = \mathbb{E}(\mathbb{E}(y|z)) = \mathbb{E}(z) = ab = \mu$$

and

$$\begin{aligned} \mathrm{Var}(y) &= \mathbb{E}(\mathrm{Var}(y|z)) + \mathrm{Var}(\mathbb{E}(y|z)) \\ &= \mathbb{E}(z) + \mathrm{Var}(z) \\ &= ab + ab^2 \\ &= \mu + \frac{1}{a}\mu^2 \\ &= \mu + \phi\mu^2 \quad \text{for} \quad \phi = 1/a. \end{aligned}$$

□

The following is based on Aitkin, Francis, Hinde, and Darnell (2009).

Suppose we have n groups of counts $y_i = (y_{i1}, \ldots y_{in_i})$, where there might be some kind of correlation between the y_{ij}s in group $i, i = 1, \ldots, n$, $j = 1, \ldots, n_i$, and we consider a log-linear Poisson model for the response

$$\log \mu_{ij}(z_i) = \eta_{ij} + z_i$$

with $z_i,\ i = 1, \ldots, n$, being a random effect for each group. For this model, the conjugate distribution of z_i is log-gamma. To use a similar approach as in Lemma 2.2 for getting the marginal pmf, we will use $w_i = e^{z_i}$. Thus, the model becomes

$$\log \mu_{ij}(w_i) = \eta_{ij} + \log w_i \tag{2.4}$$

with w_i having a gamma distribution. We take the gamma distribution in the standard form, that means that the pdf of w_i is of the form

$$f(w_i, a_i, 1/a_i) = \frac{a_i^{a_i}}{\Gamma(a_i)} w_i^{a_i - 1} e^{-w_i a_i} \quad \text{with} \quad a_i > 0.$$

With this formulation, w_i has mean 1 and variance $1/a_i$. Together with the previous assumptions it follows that $y_{ij}|w_i \sim \text{Poisson}(\mu_{ij}(w_i))$, where

$$\mu_{ij}(w_i) = w_i e^{\eta_{ij}}.$$

The marginal distribution of y_{ij} is again the negative binomial with pmf

$$\begin{aligned}
P(y_{ij} = y) &= \int_0^\infty P(y_{ij} = y | w_i = w) P(w_i = w) \\
&= \int_0^\infty \frac{a_i^{a_i}}{\Gamma(a_i)} e^{-a_i w} w^{a_i - 1} \frac{e^{-we^{\eta_{ij}}} (we^{\eta_{ij}})^y}{y!} dw \\
&= \frac{a_i^{a_i} e^{\eta_{ij} y}}{\Gamma(a_i) y!} \int_0^\infty e^{-w(e^{\eta_{ij}} + a_i)} w^{y + a_i - 1} dw \\
&= \frac{a_i^{a_i} e^{\eta_{ij} y}}{\Gamma(a_i) y!} \frac{\Gamma(y + a_i)}{(e^{\eta_{ij}} + a_i)^{y + a_i}} \\
&= \frac{\Gamma(y + a_i)}{\Gamma(a_i) y!} \left(\frac{e^{\eta_{ij}}}{e^{\eta_{ij}} + a_i} \right)^y \left(\frac{a_i}{e^{\eta_{ij}} + a_i} \right)^{a_i}.
\end{aligned}$$

With

$$\mathbb{E}(y_{ij}) = \mathbb{E}(\mathbb{E}(y_{ij}|z_i)) = \mathbb{E}(\mu_{ij}(w_i)) = \mathbb{E}(w_i e^{\eta_{ij}}) = e^{\eta_{ij}} \underbrace{\mathbb{E}(w_i)}_{=1} = e^{\eta_{ij}} = \mu_{ij}$$

we get the following negative binomial pmf

$$P(y_{ij} = y | a_i = a, \mu_{ij} = \mu) = \frac{\Gamma(y + a)}{\Gamma(a) y!} \left(\frac{\mu}{\mu + a} \right)^y \left(\frac{a}{\mu + a} \right)^a.$$

Moreover, the variance is derived by

$$\begin{aligned}
\mathrm{Var}(y_{ij}) &= \mathbb{E}(\mathrm{Var}(y_{ij}|z_i)) + \mathrm{Var}(\mathbb{E}(y_{ij}|z_i)) \\
&= \mathbb{E}(\mu_{ij}(w_i)) + \mathrm{Var}(\mu_{ij}(w_i)) \\
&= \mu_{ij} + \mathrm{Var}(w_i e^{\eta_{ij}}) \\
&= \mu_{ij} + e^{2\eta_{ij}} \mathrm{Var}(w_i) \\
&= \mu_{ij} + \mu_{ij}^2 \frac{1}{a_i} \\
&= \mu_{ij} + \mu_{ij}^2 \phi_i \quad \text{with} \quad \phi_i = 1/a_i. \qquad (2.5)
\end{aligned}$$

With

$$p_{ij} = \frac{e^{\eta_{ij}}}{e^{\eta_{ij}} + a_i}$$

a logit model for the probability $p_{ij},\ i = 1, \ldots, n; j = 1, \ldots, n_i$, yields

$$\begin{aligned}
\mathrm{logit}\, p_{ij} = \log \frac{p_{ij}}{1 - p_{ij}} &= \log \frac{\frac{e^{\eta_{ij}}}{e^{\eta_{ij}} + a_i}}{1 - \frac{e^{\eta_{ij}}}{e^{\eta_{ij}} + a_i}} = \\
&= \log \frac{e^{\eta_{ij}}(e^{\eta_{ij}} + a_i)}{a_i(e^{\eta_{ij}} + a_i)} = \eta_{ij} - \log a_i = x_{ij}^T \beta - \log a_i.
\end{aligned}$$

Thus, maximum likelihood estimation for the negative binomial distribution is rather simple, as for fixed values of $a_1, \ldots, a_n$, the parameter vector β could be estimated directly from a binomial logit model for p_{ij}. In this special case, where the a_is are known, the negative binomial distribution is part of the exponential family (using the standard iteratively re-weighted least squares (IRLS) algorithm for generalized linear models). The correct intercept is achieved by using an offset of $-\log(a_i)$ in the model. To estimate $a_i, i = 1, \ldots, n$, one can use the Newton-Raphson algorithm for the score equation. For obtaining the maximum likelihood estimates cycling between the estimation of β and $a_i, i = 1, \ldots, n$, is essential. See also Lawless (1987) and Hinde and Demetrio (1998) for further details on this estimation technique.

2.7 R-Function for the Negative Binomial Distribution

In the following section we will discuss two important functions of the statistical programm R, which can handle models for negative binomial distributed responses. The first

one is called `glm.nb` in the package `MASS` (see also Venables and Ripley, 2002). The other one is called `negbin`, provided by the package `aod` (see also Lesnoff and Lancelot, 2010). The explicit call of both functions is described in Appendix A.1.2 and A.3.1.

We will consider a dataset which is also given by Lesnoff and Lancelot (2010). A field trial in Senegal was conducted to assess the effect of ewes deworming on the mortality of their offspring. The data consists of the factor `group` with two levels, `CTRL` and `TREAT`, the numeric vector `trisk` indicating the exposition time to mortality and the values `n` and `y`, where `n` indicates the number of animals exposed to mortality and `y` the number of deaths. The data is given in Table 2.4

	CTRL				TREAT		
	n	trisk	y		n	trisk	y
1	12	11.534	0	29	30	25.422	5
2	3	2.364	2	30	9	8.515	1
3	4	2.337	2	31	8	7.036	2
4	3	3.000	0	32	10	7.885	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
28	8	4.381	5	47	2	3.165	0

Table 2.4: `dja` data: Mortality of Djallonke Lambs in Senegal.

2.7.1 Log-linear Poisson Model

The model we consider here is

$$\log \frac{\mu}{\text{trisk}} = \beta_0 + \beta_1(\text{group=TREAT}). \tag{2.6}$$

At first we will have a look at a log-linear Poisson model without any random effects

```
> library(aod)
> data(dja)
> attach(dja)
> m.glm.poisson<-glm(y~group+offset(log(trisk)), family=poisson)
```

which gives the summary

```
> summary(m.glm.poisson)
  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  -0.6975     0.1170  -5.960 2.53e-09 ***
groupTREAT   -0.8754     0.1712  -5.112 3.19e-07 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 162.67  on 74  degrees of freedom
Residual deviance: 136.86  on 73  degrees of freedom
AIC: 271.33
```

with a residual deviance of 136.86 on 73 df, indicating that the assumed variance-model does not properly fit.

2.7.2 Negative Binomial Model

A log-linear Poisson model does not seem to fit the data adequately. Therefore, we assume that we have a global overdispersion parameter ϕ. In the first section we will discuss the function `glm.nb` and in the second the function `negbin`.

Function `glm.nb`

A negative binomial regression model using `glm.nb`

```
> m.nb<-glm.nb(y~group+offset(log(trisk)))
```

yields the following summary:

```
> summary(m.nb)

  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept)  -0.5976     0.1743  -3.429 0.000605 ***
  groupTREAT   -0.9782     0.2399  -4.077 4.56e-05 ***

  (Dispersion parameter for Negative Binomial(2.6136) family taken to be 1)

      Null deviance: 101.649  on 74  degrees of freedom
  Residual deviance:  85.268  on 73  degrees of freedom
  AIC: 255.08
```

```
Number of Fisher Scoring iterations: 1

              Theta:  2.61
          Std. Err.:  1.15

2 x log-likelihood:  -249.076
```

Here, the parameter `Theta` in the summary is the a_i from Section 2.6 with $a_i = a$ as the function `glm.nb` always assumes a global dispersion parameter. To get the dispersion parameter ϕ, we have to calculate $\hat{\phi} = 1/\hat{a} = 0.383$.

Function `negbin`

The other function, `negbin`, produces slightly different estimates.

```
> m.negbin.1<-negbin(y~group+offset(log(trisk)),~1)
> m.negbin.1
  Fixed-effect coefficients:
               Estimate Std. Error    z value Pr(> |z|)
  (Intercept) -5.976e-01  1.796e-01 -3.327e+00 8.788e-04
  groupTREAT  -9.784e-01  2.439e-01 -4.012e+00 6.022e-05

  Overdispersion coefficients:
                  Estimate Std. Error   z value   Pr(> z)
  phi.(Intercept) 3.825e-01  1.694e-01 2.258e+00 1.196e-02

  Log-likelihood statistics
     Log-lik      nbpar    df res.   Deviance        AIC
  -1.245e+02          3         72  1.186e+02  2.551e+02
```

The estimates for the parameters are the same as in `m.nb`, but the standard erros slightly differ. Moreover, there is a big difference between the deviance in `m.nb` and that in `m.negbin.1`. This will be explained in the next section.

2.7.3 Deviance

One may be a bit confused about the different deviances in `m.nb` and `m.negbin.1`. The reason why there is such a big difference is that different implementations make different choices for the full or saturated model, as the deviance is calculated by

$$D(y, \mu) = -2(\log f(y, \mu) - \log f(y, y))$$

with $\log f(y, y)$ being the log-likelihood under the saturated model and $\log f(y, \mu)$ being the likelihood for the model under consideration.

The function `negbin` assumes that the maximum value of the saturated model is reached for a Poisson model. This can be shown easily as for a tending to infinity, i.e.

$$\begin{aligned} P(y_i = y) &= \frac{\Gamma(y+a)}{\Gamma(a)y!}\left(\frac{\mu}{\mu+a}\right)^y\left(\frac{a}{\mu+a}\right)^a \\ &= \frac{\Gamma(y+a)}{\Gamma(a)y!}\left(\frac{\mu}{\mu+a}\right)^y\left(\frac{1}{1+\mu/a}\right)^a \\ &= \underbrace{\frac{\Gamma(y+a)}{\Gamma(a)(\mu+a)^y}}_{\xrightarrow{a\to\infty}1}\frac{\mu^y}{y!}\underbrace{\left(\frac{1}{1+\mu/a}\right)^a}_{\xrightarrow{a\to\infty}\frac{1}{e^\mu}}. \end{aligned}$$

Therefore,

$$\lim_{a\to\infty} P(y_i = y) = \frac{\mu^y}{y!}e^{-\mu},$$

which is the pmf of a Poisson(μ) distribution.

Figure 2.1 shows this characteristic with $\lambda = 4$ for the Poisson distribution, and for increasing values of a for the negative binomial distribution.

The deviance of `m.negbin.1` can easily be calculated through the log-likelihood for the model under consideration

```
> sum(dnbinom(x=y, size=a, mu=fitted(m.negbin.1), log=TRUE))
 [1] -124.5383
```

and the log-likelihood under the saturated model

```
> sum(dpois(x=y, lambda=y, log=TRUE))
 [1] -65.2358,
```

which results into $-2(-124.54 + 65.24) = 118.60$ and this is the same as in the summary of `m.negbin.1`.

The function `glm.nb` instead assumes that the maximal likelihood for the saturated model is reached for the estimated ϕ. This can be seen in Figure 2.2, where the negative binomial

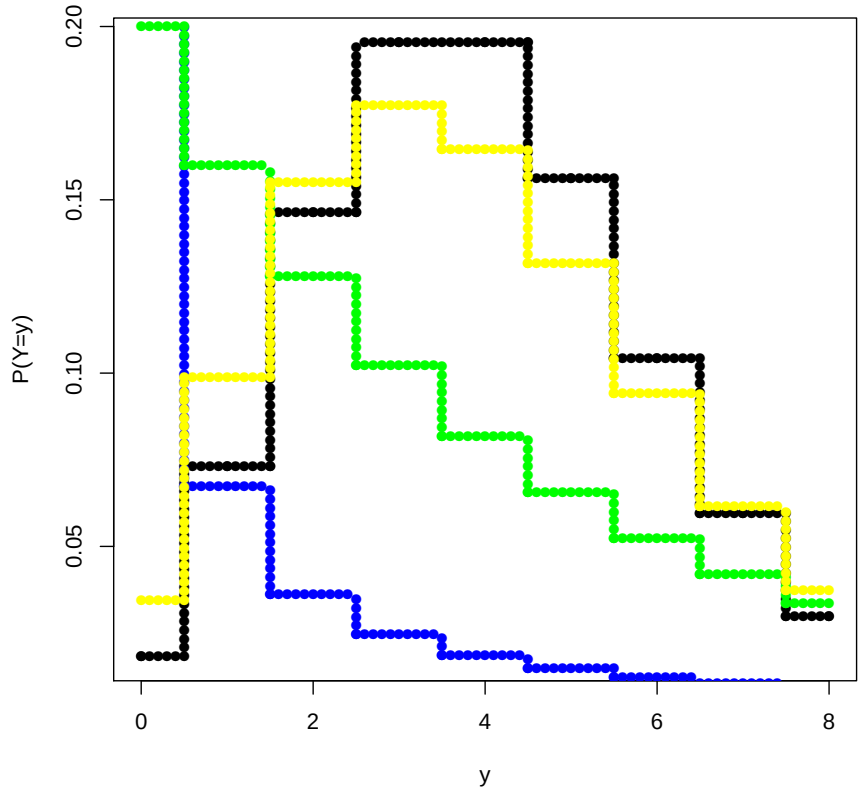

Figure 2.1: Convergence of the Negative Binomial pmf for growing a to the Poisson pmf. Blue: Negative binomial with $a = 0.1$, green: Negative binomial with $a = 1$, yellow: Negative binomial with $a = 10$, black: Poisson with $\lambda = 4$.

log-likelihood function with various values of ϕ is shown.

Now, the log-likelihood function under the saturated model is calculated as

```
> sum(dnbinom(x=y, size=size, mu=y, log=TRUE))
  [1] -81.90406
```

therefore the deviance is $-2(-124.5383 + 81.904) = 85.268$.

Note that for the function `dnbinom` one has to use for `size` the parameter a, because the variance of `dnbinom` for this implementation is given by $\mu + \mu^2/size$. (Further information and the explicit call is described in A.5.2.)

A big advantage of the function `negbin` over the function `glm.nb` is that the user can either fix the parameter ϕ to a constant or the parameter ϕ will be estimated from the sample. Further one can also specify the random argument either to allow only for a global

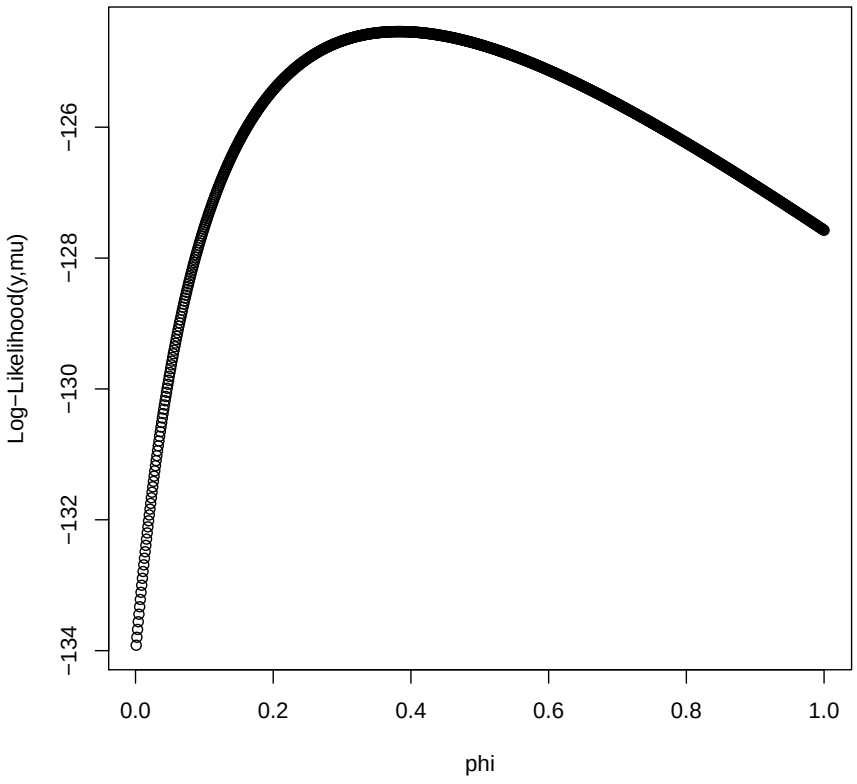

Figure 2.2: Log-Likelihood function corresponding to a negative binomial model, evaluated at $\hat{\mu}$, allowing various values of phi.

parameter (`random=~1`) or for specific parameters for the levels of a given group factor (`random=~group`). The function `glm.nb` always assumes a global dispersions parameter.

In order to decrease the deviance we can try the same model but with group specific dispersion parameters ϕ_g. So that the variance of the model varies over the different levels of the group factor. This is supported by the function `negbin`.

```
> m.negbin.group<-negbin(y~group+offset(log(trisk)), ~group, dja)
> m.negbin.group
  Fixed-effect coefficients:
                Estimate Std. Error    z value Pr(> |z|)
  (Intercept) -5.404e-01  2.297e-01 -2.353e+00 1.862e-02
  groupTREAT  -1.035e+00  2.616e-01 -3.958e+00 7.547e-05
```

```
  Overdispersion coefficients:
                  Estimate Std. Error   z value   Pr(> z)
  phi.groupCTRL  8.388e-01   4.18e-01 2.007e+00 2.239e-02
  phi.groupTREAT 1.232e-07   2.00e-13 0.000e+00 1.000e+00

  Log-likelihood statistics
     Log-lik      nbpar    df res.   Deviance        AIC       AICc
  -1.212e+02          4         71  1.118e+02  2.503e+02  2.509e+02
```

The output of `m.negbin.group` indicates that group specific dispersion parameters are supported only for group `CTRL` (p-value of 0.02239), whereas there seems to be no need for a dispersion parameter for group `TREAT` (p-value of 1.0).
An ANOVA comparing the model with a global ϕ to the one with group specific dispersions yields the following result:

```
> anova(m.negbin.1, m.negbin.group)
                    logL k   AIC  Res.dev. Res.Df  Df P(> Chi2)
  m.negbin.1     -124.5 3 255.1     118.6     72
  m.negbin.group -121.2 4 250.3     111.8     71   1  0.009242
```

There is a strong evidence for the second model, as the value of the χ^2-test statistic indicates. The deviance slightly decreased, but we can't compare the deviance with the degrees of freedom because of the choice of the saturated model. Now we recalculate the deviance and take for the saturated model the negative binomial distribution with the estimated ϕ-vector.

First we have to define the ϕ vector. As there are two different groups `CTRL` and `TREAT` we have two group specific ϕ's, where ϕ_{CTRL} belongs to group `CTRL` and ϕ_{TREAT} to group `TREAT`.

```
> a<-rep(0,75);
> for(i in 1:75)
   if( group[i]=="CTRL" )
   {a[i]= 1/m.negbin.group@random.param[1]}

> for(i in 1:75)
   if(group[i]=="TREAT")
   {a[i]=1/m.negbin.group@random.param[2]}
```

The deviance is then calculated as

```
> -2*(sum(dnbinom(x=y, size=a, mu=fitted(m.negbin.group), log=TRUE))-
  sum(dnbinom(x=y, size=a, mu=y, log=TRUE)))
  [1] 85.45406.
```

This is not smaller than the deviance of a model with a global dispersion parameter (dev=85.268). So if we consider this model and calculate the saturated model for the deviance by the negative binomial distribution with the estimated ϕ's, then a global dispersion parameter would be supported. Otherwise, for calculating the saturated model via the Poisson distribution, a group specific overdispersion parameter seems to be more appropriate.

Maybe the best idea for other models is to decide model-based whether a global dispersion parameter or a group specific dispersion parameter is more appropriate. If the former investigation gives rise to a group specific dispersion parameter, the function `negbin` is the one which can handle that, otherwise, for a global dispersion parameter, both functions yield the same parameter estimates.

To sum up, in Table 2.5 we see, that the parameter estimates are slightly affected by the overdispersion modeling with a smaller slope and a larger intercept. Important to note is that there is almost no dispersion in the `TREAT` group and a rather large dispersion in the `CTRL` group. Also the standard errors of the coefficients increase, as the variability is increasing, induced by the overdispersion parameters. In order to correct this, the dataset would need further investigation. Compared to a log-linear Poisson model without a random effect the model with random effects fits the dataset better, but as said before, it would still need further investigation.

Figure 2.3 shows the difference between the fitted values of `m.nb` and `m.negbin.group` versus `trisk`. The green points correspond to `m.nb` and the purple points to `m.negbin.group`. As one can see, there is a difference between green and purple points in the upper group, meaning that the upper points correspond to group `CTRL`. This also illustrates the nearly zero dispersion coefficient in group `TREAT`.

2.8 Extra-Binomial Variation

Logistic models are often used when the response variable is a proportion. Then it can happen, that even when all available explanatory variables have been used in the predictor, the residual variation is still greater than assumed by a binomial model. Therefore, we allow somewhat called extra-binomial variation.

	m.glm.poisson		m.nb		m.negbin.1		m.negbin.group	
Intercept	−0.6975	(0.1170)	−0.5976	(0.1743)	−0.5976	(0.1796)	−0.5404	(0.2297)
groupTREAT	−0.8754	(0.1712)	−0.9782	(0.2399)	−0.9784	(0.2439)	−1.035	(0.2616)
phi			0.3826		0.3825	(0.1694)		
phi.groupTREAT					$1.2e^{-07}$	$(2e^{-13})$		
phi.groupCTRL							0.8388	(0.418)
res.dev.	136.86		85.26		118.6		111.8	
df	73		73		72		71	

Table 2.5: `dja` data: Parameter estimates and deviances under four different models: `m.glm.poisson`, `m.nb`, `m.negbin.1`, `m.negbin.group`.

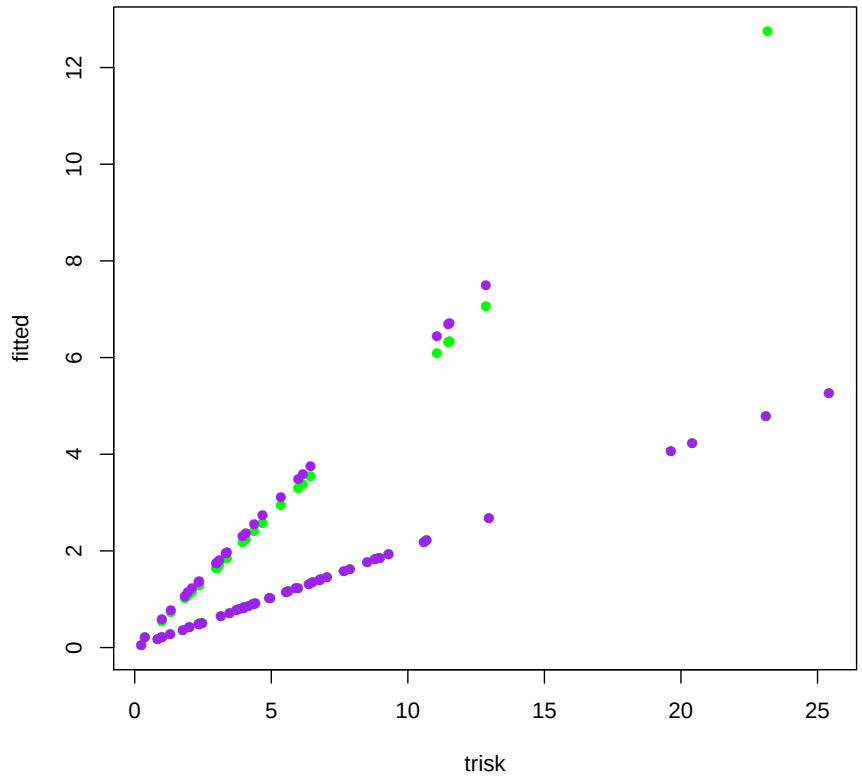

Figure 2.3: Example `dja`: Fitted versus `trisk` with a global (green) dispersion parameter and with group specific (purple) dispersion parameters.

This section is based on Williams (1982) and Friedl (1991).

To allow for extra-binomial variation we introduce unobserved continuous random vari-

ables z_i, independent distributed on $(0,1)$ with

$$\mathbb{E}(z_i) = \pi_i \quad \text{and} \quad \text{Var}(z_i) = \phi\pi_i(1-\pi_i).$$

Furthermore, the conditional pmf of $y_i, i = 1,\ldots,n$, given z_i is assumed to be binomial, so

$$y_i|z_i \sim \text{Binomial}(m_i, z_i).$$

From the proof of Lemma 2.6, the first two moments of the responses are

$$\begin{aligned} \mathbb{E}(y_i) &= m_i\pi_i = \mu_i \\ \text{Var}(y_i) &= (1+\phi(m_i-1))m_i\pi_i(1-\pi_i) = a_i(\phi)V(\mu_i). \end{aligned}$$

In this scenario we cannot use maximum likelihood estimation because the distribution of y_i is not fully specified. The relationship between mean and variance restricts us to the use of the quasi-likelihood function which is maximized by iterative use of the weighted least squares equations. The adjusted dependent variables introduced in the IRLS algorithm from Section 1.3.1 have the form

$$z = X\beta + D(y-\mu) = \eta + \frac{\partial\eta}{\partial\mu}(y-\mu).$$

In the following we will consider the canonical link $g(\mu)$. The canonical link for the binomial variance is the logit-link

$$g(\mu) = \eta = \log\frac{\mu}{m-\mu}.$$

From this it follows that

$$\frac{\partial\mu}{\partial\eta} = \frac{\exp(\eta)}{(1+\exp(\eta))^2} = V(\mu) = \frac{1}{g'(\mu_i)}.$$

Mean and variance of z_i are given by

$$\begin{aligned} \mathbb{E}(z_i) &= \mathbb{E}\left(\eta_i + \frac{y_i-\mu_i}{V(\mu_i)}\right) = \eta_i \\ \text{Var}(z_i) &= \mathbb{E}\left(\frac{(y_i-\mu_i)^2}{V^2(\mu_i)}\right) = \frac{\text{Var}(y_i)}{V^2(\mu_i)} \\ &= \frac{a_i(\phi)V(\mu_i)}{V^2(\mu_i)} = \frac{a_i(\phi)}{V(\mu_i)}. \end{aligned}$$

If we use the matrix $W = \text{diag}(w_i)$ as in equation (1.6) with

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

then the estimate for the parameter vector β becomes

$$\hat{\beta} = \left(X^T W X\right)^{-1} X^T W z. \tag{2.7}$$

With $\hat{\eta} = X\hat{\beta}$ it follows from the IRLS algorithm that

$$\hat{\eta} = X\left(X^T W X\right)^{-1} X^T W z.$$

Furthermore, we define residuals

$$z - \hat{\eta} = \left(I - X\left(X^T W X\right)^{-1} X^T W\right) z.$$

Since $\mathbb{E}(z) = \eta = X\beta$ we have

$$\mathbb{E}(z - \hat{\eta}) = 0$$

and because of $\text{Var}(z) = W^{-1}$ we get

$$\begin{aligned}
\text{Var}(z-\hat{\eta}) &= \text{Var}\left((I - X(X^TWX)^{-1}X^TW)z\right) \\
&= \text{Var}\left((X(X^TWX)^{-1}X^TW - I)z\right) \\
&= \left(X(X^TWX)^{-1}X^TW - I\right)\text{Var}(z)\left(WX(X^TWX)^{-1}X^T - I\right) \\
&= \underbrace{X\,(X^TWX)^{-1}X^T\underbrace{WW^{-1}}_{=I}WX(X^TWX)^{-1}X^T}_{=I} - \underbrace{W^{-1}W}_{=I}X(X^TWX)^{-1}X^T \\
&\quad -X(X^TWX)^{-1}X^T\underbrace{WW^{-1}}_{=I}+W^{-1} \\
&= W^{-1} - X(X^TWX)^{-1}X^T = W^{-1} - H_w
\end{aligned}$$

with H_w being the weighted hat matrix $H_w = (h_{ij}) = X(X^TWX)^{-1}X^T$.
The weights w_i depend on ϕ which is usually unknown. If the weights w_i are calculated from an initial estimate of ϕ, and β is estimated iteratively from (2.7), then Williams (1982) approximated the goodness of fit statistic

$$\hat{X}^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{a_i(\phi)V(\hat{\mu}_i)},$$

through the weighted sum of squares of the residuals, i.e.

$$\hat{X}^2 \approx (z-\hat{\eta})^T W (z-\hat{\eta}).$$

The mean of this approximation is then

$$\begin{aligned}\mathbb{E}(\hat{X}^2) &= \sum_{i=1}^{n} \mathrm{Var}(z_i-\hat{\eta}_i)w_i = \sum_{i=1}^{n}(w_i^{-1}-h_{ii})w_i \\ &= \sum_{i=1}^{n}(1-w_i h_{ii}) = \sum_{i=1}^{n}(1-V(\mu_i)h_{ii}/a_i(\phi)).\end{aligned}$$

If we now substitute $a_i(\phi) = 1+\phi(m_i-1)$, the resulting general mean of the approximation is given by

$$\mathbb{E}(\hat{X}^2) = \sum_{i=1}^{n} \frac{1}{a_i(\phi)} (1-V(\mu_i)h_{ii}/a_i(\phi)) (1+\phi(m_i-1)). \tag{2.8}$$

In case of $a_i(\phi) = 1$, $i = 1,\ldots,n$, equation (2.8) gets

$$\begin{aligned}\mathbb{E}(\hat{X}^2) &= \sum_{i=1}^{n} (1-V(\mu_i)h_{ii}) (1+\phi(m_i-1)) \\ &= \sum_{i=1}^{n} 1 - \sum_{i=1}^{n} V(\mu_i)h_{ii} + \sum_{i=1}^{n}(1-V(\mu_i)h_{ii})\phi(m_i-1) \\ &= n - \sum_{i=1}^{n} V(\mu_i)h_{ii} + \phi\sum_{i=1}^{n}(1-V(\mu_i)h_{ii})(m_i-1). \end{aligned}\tag{2.9}$$

Furthermore, for $a_i(\phi) = 1$ it follows that $WH_w = VH_w = WX(X^TWX)^{-1}X^T = X(X^TX)^{-1}X^T = H$ which is the unweighted hat matrix with $trace(H) = p$. With this knowledge, equation (2.9) simplifies to

$$\mathbb{E}(\hat{X}^2) = n - p + \phi\sum_{i=1}^{n}(m_i-1)(1-V(\mu_i)h_{ii}). \tag{2.10}$$

Furthermore, if all the m_i are equal, $m_i = m$, $i = 1,\ldots,n$, equation (2.10) reduces to

$$\begin{aligned}
\mathbb{E}(\hat{X}^2) &= n - p + \phi \sum_{i=1}^{n} (m-1)(1 - V(\mu_i) h_{ii}) \\
&= n - p + \phi \sum_{i=1}^{n} (m-1) - \phi \sum_{i=1}^{n} (m-1)(V(\mu_i) h_{ii}) \\
&= n - p + \phi n(m-1) - \phi(m-1)p \\
&= (n-p)(1 + \phi(m-1)),
\end{aligned}$$

and the heterogeneity factor $1+\phi(m-1)$ is estimated by $\hat{X}^2/(n-p)$. Altogether Williams (1982) derived the following procedure: ("Model II")

1. Assume $\phi = 0$, then it follows that $a_i(\phi) = 1$. From the estimation of the model $\text{logit}(\mu) = \eta$ by maximum likelihood, the estimate $\hat{\beta}$ results. Evaluate the $\hat{X}^2$ goodness of statistic.

2. Compare $\hat{X}^2$ with the mean of a χ^2_{n-p} distribution. If $\hat{X}^2$ is unacceptably large conclude that $\phi > 0$ and calculate the estimate

$$\hat{\phi} = \frac{\hat{X}^2 - (n-p)}{\sum_{i=1}^{n} (m_i - 1)(1 - V(\hat{\mu}_i) h_{ii})}. \tag{2.11}$$

3. Calculate new iterates $a_i(\hat{\phi}) = 1 + \hat{\phi}(m_i - 1)$ and estimate β iteratively using (2.7) and recalculate $\hat{X}^2$.

4. If $\hat{X}^2$ is close to $n - p$ the estimate $\hat{\phi}$ is satisfactory. If not, re-estimate $\hat{\phi}$ as

$$\hat{\phi} = \frac{\hat{X}^2 - \sum_{i=1}^{n} \frac{1}{a_i(\hat{\phi})} (1 - V(\hat{\mu}_i) h_{ii} / a_i(\hat{\phi}))}{\sum_{i=1}^{n} \frac{1}{a_i(\hat{\phi})} (1 - V(\hat{\mu}_i) h_{ii} / a_i(\hat{\phi}))(m_i - 1)}$$

   and return to step 3.

2.9 R-Function for Extra-Binomial Variation

The function considered here fits the generalized linear model II proposed by Williams (1982) and is called `quasibin` provided by the package `aod`. The explicit call is described in Appendix A.1.3

We will again consider the `orob2` dataset given in Table 2.2. Let us have a look at the following three models.

```
> m.glm.binomial<-glm(cbind(y,n-y)~seed*root, family=binomial, data=orob2)
> m.quasibin.0<-quasibin(cbind(y,n-y)~seed*root, data=orob2, phi=0)
> m.quasibin<-quasibin(cbind(y,n-y)~seed*root, data=orob2)
```

The predictor model we consider here is given by

$$\eta = \beta_0 + \beta_1(\text{seed=O75}) + \beta_2(\text{root=CUCUMBER}) + \beta_3(\text{seed=O75*root=CUCUMBER}).$$

Due to the fact that in the second model the overdispersion parameter is set to zero, the resulting parameter estimates should have the same values as the parameter estimates in the logistic model with `family=binomial` (see Table 2.3). Model `m.quasibin` results in:

```
> m.quasibin
  Fixed-effect coefficients:
                       Estimate Std. Error z value Pr(>|z|)
  (Intercept)           -0.4653     0.2439 -1.9081   0.0564
  seedO75               -0.0701     0.3115 -0.2250   0.8219
  rootCUCUMBER           0.5102     0.3347  1.5244   0.1274
  seedO75:rootCUCUMBER   0.8196     0.4352  1.8831   0.0597

  Overdispersion parameter:
     phi
  0.0249

  Pearson's chi-squared goodness-of-fit statistic = 17.0007
```

The binomial model results into a residual deviance of 33.278 on 17 df and in a Pearson statistic of 31.651 on 17 df, see Table 2.3, indicating clearly evidence against this binomial assumption and the supposed variance structure. As a result of Williams procedure, we get an overdispersion coefficient of 0.0249 and slightly other parameter estimates for the linear predictor. The Pearson statistic corresponds to its df since in step 4 of the algorithm we check whether $X^2 = (n - p)$. Furthermore we get larger standard errors, as the variability increases.

One disadvantage of the function `quasibin` is that there is no `summary`-method available. Therefore the output of the function `quasibin` includes all relevant information.

Parameter estimates and residual deviances of this three models are given in Table 2.6.

	m.glm.binomial		m.quasibin.0		m.quasibin	
Intercept	−0.4122	(0.1842)	−0.4122	(0.1842)	−0.4653	(0.2439)
seed075	−0.1459	(0.2232)	−0.1459	(0.2232)	−0.0701	(0.3115)
rootCUCUMBER	0.5401	(0.2498)	0.5401	(0.2498)	0.5102	(0.3347)
seed075:rootCUCUMBER	0.7781	(0.3064)	0.7781	(0.3064)	0.8196	(0.4352)
phi					0.0249	
res.dev.	33.278					
Pearson X^2	31.651		31.651		17.0007	
df	17		17		17	

Table 2.6: orob2 data: parameter estimates and deviances for the models m.glm.binomial, m.quasibin.0 and m.quasibin.

2.10 Extra-Poisson Variation

Count data y_i, $i = 1, \ldots, n$, are often fit by a log-linear model as described in Section 1.2.2. Then it can happen that the data exhibits greater variability than is assumed by the implicit mean-variance relationship. To overcome this we allow somewhat called extra-Poisson variation (compare with Breslow, 1984).

We now assume that there exist unobserved random variables λ_i with

$$\mathbb{E}(\lambda_i) = \mu_i \quad \text{and} \quad \text{Var}(\lambda_i) = \phi\mu_i^2. \tag{2.12}$$

Furthermore, we assume that the conditional pmf of y_i given λ_i is a Poisson pmf, i.e.

$$y_i|\lambda_i \sim \text{Poisson}(\lambda_i).$$

It follows that

$$\mathbb{E}(y_i) = \mathbb{E}\left(\mathbb{E}(y_i|\lambda_i)\right) = \mathbb{E}(\lambda_i) = \mu_i$$

and

$$\text{Var}(y_i) = \mathbb{E}(\mathbb{E}(y_i|\lambda_i)) + \text{Var}(\mathbb{E}(y_i|\lambda_i)) = \mu_i + \phi\mu_i^2 = \mu_i(1 + \phi\mu_i).$$

For fitting this model, Breslow (1984) suggested, to set the chi-square criterion equal to its degrees of freedom, i.e.

$$\sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\widehat{\text{Var}}(y_i)} = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i(1 + \hat{\phi}\hat{\mu}_i)} \stackrel{!}{=} n - p$$

or

$$\hat{\phi} = \frac{1}{n-p} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i(\hat{\mu}_i + 1/\hat{\phi})}. \tag{2.13}$$

This and setting weights $w_i = (1 + \phi\hat{\mu}_i)^{-1}$ leads to the so called Procedure II:
The initial fit is made with weights $w_i = 1$.

1. Fit the Poisson model as described in Section 1.3.1. If the deviance is close to its degrees of freedom stop and conclude that the residual variation is adequately explained, otherwise go to step 2.

2. Solve equation (2.14) iteratively for ϕ

$$\phi = (n-p)^{-1} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i(\hat{\mu}_i + 1/\phi)}) \tag{2.14}$$

3. Define new weights $w_i = (1 + \phi\hat{\mu}_i)^{-1}$ and return to step 1.

If the model with weights $w_i = 1$ does not fit adequately, an initial estimate of ϕ is

$$\phi^{(0)} = \frac{\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2/\hat{\mu}_i - (n-p)}{\sum_{i=1}^{n} \hat{\mu}_i(1 - \hat{\mu}_i h_{ii})}, \tag{2.15}$$

where the matrix of iterated weights $\hat{W}$ has diagonal elements $\hat{\mu}_i$ and the h_{ii} are the diagonal elements of $H_W = X(X^T\hat{W}X)X^T$. Equation (2.15) is derived by considering the associated least squares problem.

2.11 R-Function for Extra-Poisson Variation

In this section we will consider one function in R which fits models based on Procedure II in Breslow (1984). The function is called `quasipois` provided by the package `aod`. The explicit call is described in Appendix A.1.4

Table (2.7) shows the number of revertant colonies of salmonella observed on each of three replicate plates tested at each of six dose levels of quinoline. This data is also given in the package `aod`. We consider a log-linear model with

$$\log(\mu) = \beta_0 + \beta_1(dose) + \beta_2 \log(dose + 10).$$

As in the previous section, we will have a look at the following three models.

Dose of quinoline (μg *per plate*)					
0	10	33	100	333	1000
15	16	16	27	33	20
21	18	26	41	38	27
29	21	33	60	41	42

Table 2.7: Numbers of revertant colonies of salmonella.

```
> library(aod)
> data(salmonella)
> attach(salmonella)
> m.glm.salm<-glm(y~dose+log(dose+10),family=poisson)
> m.quasipois.0<-quasipois(y~dose+log(dose+10), phi=0)
> m.quasipois<-quasipois(y~dose+log(dose+10))
```

A `summary` of `m.glm.salm`, a log-linear Poisson model, yields

```
> summary(m.glm.salm)

  Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
  (Intercept)     2.1727730  0.2184269   9.947  < 2e-16 ***
  dose           -0.0010130  0.0002452  -4.131 3.61e-05 ***
  log(dose + 10)  0.3198250  0.0570014   5.611 2.01e-08 ***

      Null deviance: 78.358  on 17  degrees of freedom
  Residual deviance: 43.716  on 15  degrees of freedom
  AIC: 142.25
```

with a residual deviance of 43.72 on 15 df, which indicates that there might be overdispersion in the data. Therefore, fitting the data with `quasipois` yields to:

```
> m.quasipois
  Fixed-effect coefficients:
                Estimate Std. Error z value Pr(>|z|)
  (Intercept)      2.2031     0.3636  6.0591   < 1e-4
  dose            -0.0010     0.0004 -2.2284   0.0259
  log(dose + 10)   0.3110     0.0991  3.1394   0.0017

  Overdispersion parameter:
     phi
  0.0718
```

```
Pearson's chi-squared goodness-of-fit statistic = 15.0004
```

	m.glm.salm		m.quasipois.0		m.quasipois	
Intercept	2.173	(0.2184)	2.173	(0.2184)	2.203	(0.3636)
dose	−0.001	(0.0002)	−0.001	(0.0002)	−0.001	(0.0004)
log(dose+10)	0.320	(0.0570)	0.320	(0.0570)	0.311	(0.0991)
phi					0.0718	
res.dev.	43.716					
Pearson X^2	46.27		46.27		15.004	
df	15		15		14	

Table 2.8: Salmonella data: parameter estimates and deviances for m.glm.salm, m.quasibin.0 and m.quasibin.

Like before, the parameter estimates in Table (2.8) for the Poisson model and the parameter estimates for the quasipois model when ϕ is fixed to 0 are exactly the same. Using a likelihood approach results in a residual deviance of 43.716 on 15 degrees of freedom, which could mean that the data exhibit greater variability than is assumed by the mean-variance relationship. So there is strong evidence for the model Breslow (1984) described. Fitting the data with this model results in a overdispersion coefficient of 0.0718, almost the same parameter estimates, but slightly larger standard errors.

2.12 Conclusion

In this chapter we considered models for negative binomial and beta binomial distributed data. Furthermore, we discussed how to handle extra-Poisson variation and extra-binomial variation. For each of the topics discussed appropriate R functions exist, which are introduced in two different R packages, MASS and aod. All of them deliver results of the parameter estimates similar to the parameter estimates of the log-linear Poisson model or the logistic binomial model, but induced by allowing larger variability, the standard errors are also larger.

All of the functions seem to work properly, moreover the documentation of the package aod is explained in full detail.

Chapter 3

Random Effect Models

In the previous chapter we discussed one approach for obtaining the marginal pmf or pdf $f(y, \theta)$. There we assumed that the density function of an observation (y, z) is $f(y, z, \theta)$ with θ being the vector of all unknown parameters. Therefore, to obtain $f(y, \theta)$ we solved the integral

$$l(y, \theta) = \log f(y, \theta) = \log \int f(y, z, \theta) \mathrm{d}z = \log \int f(y|z, \theta) f(z, \theta) \mathrm{d}z. \tag{3.1}$$

This could be easily calculated when assuming a conjugate distribution for the distribution of $f(z, \theta)$. In this chapter another approach to solve this integral is discussed. Section 3.1 briefly introduces the EM-algorithm and Section 3.2 deals with two kinds of overdispersion models, namely the non-parametric maximum likelihood estimation and the normal random effect models. Section 3.3 is dedicated to the investigation of variance component models and Section 3.4 to random coefficient models. In addition, in every section some R-functions are considered to discuss the explained theory in practice.

3.1 EM-Algorithm

This section is based on Dempster, Laird, and Rubin (1977) and Lee et al. (2006). Since the following algorithm consists of two steps, namely the expectation step and the maximization step, the algorithm is called EM-algorithm.

The conditional density of z given y is denoted by

$$f(z|y, \theta) = \frac{f(y, z, \theta)}{f(y, \theta)}.$$

With this, the marginal log-likelihood function (3.1) can be written as

$$l(y,\theta) = \log f(y,\theta) = \log f(y,z,\theta) - \log f(z|y,\theta)$$

or

$$\log f(y,z,\theta) = \log f(z|y,\theta) + l(y,\theta). \tag{3.2}$$

If we consider $l(y,\theta)$ then we notice that its conditional expectation given the observation y is again $l(y,\theta)$, since

$$\mathbb{E}\left(l(y,\theta)|y,\theta_0\right) = \int l(y,\theta) f(z|y,\theta_0)\mathrm{d}z = l(y,\theta) \underbrace{\int f(z|y,\theta_0)\mathrm{d}z}_{1} = l(y,\theta).$$

Meaning that optimizing $\mathbb{E}\left(l(y,\theta)|y,\theta_0\right)$ leads to the same result as optimizing $l(y,\theta)$. Therefore, the following equation holds,

$$\begin{aligned} \mathbb{E}\left(\log f(y,z,\theta)|y,\theta_0\right) &= \mathbb{E}\left(\log f(z|y,\theta)|y,\theta_0\right) + \mathbb{E}\left(l(y,\theta)|y,\theta_0\right) \\ \int \log f(y,z,\theta) f(z|y,\theta_0)\mathrm{d}z &= \int \log f(z,\theta) f(z|y,\theta_0)\mathrm{d}z + l(y,\theta) \\ Q(\theta|\theta_0) &= H(\theta|\theta_0) + l(y,\theta). \end{aligned}$$

To determine the maximum likelihood estimate $\hat{\theta}$ we have to maximize the marginal log-likelihood function $l(y,\theta)$ in θ. This can be alternatively achieved by maximizing $Q(\theta|\theta_0)$ in θ and θ_0:
Suppose that θ', maximizes the function $Q(\theta|\theta_0)$ for a given value θ_0. Therefore, the difference between the Q-functions is

$$Q(\theta'|\theta_0) - Q(\theta_0|\theta_0) \geq 0, \tag{3.3}$$

and the resulting difference of the marginal log-likelihood functions is

$$l(\theta'|y) - l(\theta_0|y) = Q(\theta'|\theta_0) - Q(\theta_0|\theta_0) - \left(H(\theta'|\theta_0) - H(\theta_0|\theta_0)\right).$$

Lemma 3.1 *For any pair (θ,θ_0) the inequality*

$$H(\theta|\theta_0) \leq H(\theta_0|\theta_0) \tag{3.4}$$

holds, with equality if and only if $f(z|y,\theta) = f(z|y,\theta_0)$ almost everywhere.

Proof:
Equation (3.4) is a consequence of Jensen's inequality $\mathbb{E}(g(x)) \leq g(\mathbb{E}(x))$ for concave functions $g(x)$ like the logarithm.

$$\begin{aligned}
H(\theta|\theta_0) - H(\theta_0|\theta_0) &= \int \log f(z|y,\theta) f(z|y,\theta_0) \mathrm{d}z - \int \log f(z|y,\theta_0) f(z|y,\theta_0) \mathrm{d}z \\
&= \int \log \frac{f(z|y,\theta)}{f(z|y,\theta_0)} f(z|y,\theta_0) \mathrm{d}z \\
&= \mathbb{E}\left(\log \frac{f(z|y,\theta)}{f(z|y,\theta_0)} \middle| y, \theta_0 \right) \\
&\leq \log \mathbb{E}\left(\frac{f(z|y,\theta)}{f(z|y,\theta_0)} \middle| y, \theta_0 \right) \\
&= \log \int \frac{f(z|y,\theta)}{f(z|y,\theta_0)} f(z|y,\theta_0) \mathrm{d}z \\
&= \log \underbrace{\int f(z|y,\theta) \mathrm{d}z}_{=1} \\
&= 0.
\end{aligned}$$

□

Thus, with (3.3) and (3.4) it is clear that maximizing $Q(\theta|\theta_0)$ results in an increase of $l(y,\theta)$, therefore

$$l(\theta'|y) - l(\theta_0|y) \geq 0.$$

As mentioned above, the EM-algorithm contains two important steps. The first is the expectation step where the conditional mean $Q(\theta|\theta_0)$ for a given value θ_0 is calculated. The second step is called maximization step where $Q(\theta|\theta_0)$ is maximized in θ. By denoting θ' as the maximizer, an E-step is performed with $\theta_0 = \theta'$. These two steps are repeated until convergence occurs.

The EM-algorithm is known to have a slow (linear) convergence rate, but is remarkable because of its simplicity and the generality of the associated theory. Instead of maximizing the integral (3.1), one has to consider

$$Q(\theta|\theta_0) = \sum_{i=1}^{n} \int \log f(y_i, z_i, \theta) f(z_i|y_i, \theta_0) \mathrm{d}z_i,$$

which can be written with the conditional density of z given y denoted by

$$f(z|y,\theta) = \frac{f(y|z,\theta)f(z,\theta)}{f(y,\theta)},$$

as

$$\begin{aligned} Q(\theta|\theta_0) &= \sum_{i=1}^{n} \int \log f(y_i, z_i, \theta) \frac{f(y_i|z_i,\theta_0)}{f(y_i,\theta_0)} f(z_i,\theta_0) \mathrm{d}z_i \\ &= \sum_{i=1}^{n} \frac{1}{f(y_i,\theta_0)} \int \log f(y_i, z_i, \theta) f(y_i|z_i,\theta_0) f(z_i,\theta_0) \mathrm{d}z_i. \end{aligned} \tag{3.5}$$

Unfortunately it is still often hard to evaluate this integral and further $f(y_i, \theta_0)$ explicitly. An approach to avoid this difficulty is numerical integration, therefore two methods of numerical integration are described in the Sections 3.2.1 and 3.2.2. The first method uses the Gauss-Hermite quadrature, the second estimates the integral via a non-parametric approach. However, this can become computationally heavy as the dimensionality of the integral increases. For such cases, we can approximate the marginal likelihood function by the Laplace approximation, compare to Lee et al. (2006),

$$\int_{\mathbb{R}^d} f(y|z,\theta) f(z,\theta) \mathrm{d}z = f(z,\theta) f(y|z,\theta) \left(\frac{2\pi}{n}\right)^{d/2} |D|^{-1/2} \{1 + \mathcal{O}(n^{-1})\} \Big|_{z=z^*} \tag{3.6}$$

where $D = \partial^2 \log f(y|z,\theta)/\partial z^2$ and z^* solves $\partial \log f(y|z,\theta)/\partial z = 0$ for fixed θ and d denotes the dimensionality of the random effect.

3.2 Overdispersion Models

Sometimes one or more important explanatory variables are missing, therefore adding random variables into the linear predictor is a good way of expressing this problem. Furthermore, if the specified variance structure does not represent the data appropriately, this could be an indication for using random intercept models. Refer to this as overdispersed data.

Without any loss of generality we can write the linear predictor as in equation (2.1) as

$$\eta_i = g(\mu_i) = x_i^T \beta + \sigma z_i,$$

with (β, σ) being the unknown parameter vector. Two common methods are described in the next two sections. The first method assumes that z follows a standard normal distribution, $z \sim N(0,1)$, and the second estimates $f(z)$ by the non-parametric maximum likelihood estimate $\hat{f}(z)$.

3.2.1 Normal Random Effect Models

Hinde (1982) introduced fitting compound Poisson models, where z is assumed to be normally distributed. This approach can be extended to fit models with another distribution than Poisson, see e.g. Friedl (1998). By assuming a normal distribution for z, the model is given by

$$\begin{aligned} y_i|z_i &\sim F(\mu_i) \\ F &\in \text{Exponential family}(\mu_i) \\ z_i &\sim N(0,1) \end{aligned}$$

with the linear predictor

$$\eta_i = g(\mu_i) = x_i^T\beta + \sigma z_i, \qquad z_i \overset{iid}{\sim} N(0,1).$$

Thus, we write $\phi(z_i)$, the standard normal density, for $f(z_i, \theta_0)$. However, since z_i were not observed, the maximum likelihood estimates can be determined by the EM-algorithm. The $Q(\theta|\theta_0)$ function is then given by

$$Q(\theta|\theta_0) = \sum_{i=1}^{n} \frac{1}{f(y_i,\theta_0)} \int \log f(y_i, z_i, \theta) f(y_i|z_i,\theta_0)\phi(z_i)\mathrm{d}z_i. \tag{3.7}$$

In the following we consider an approach to approximate this integral numerically. For this purpose we define in the next subsection the Gaussian quadrature, which evaluates the continuous integral as the discrete sum of a finite number of terms.

K-point Gauss-Hermite Quadrature

Consider an integral of the form

$$\int f(u)e^{-u^2}\mathrm{d}u,$$

then the K-point Gaussian Hermite quadrature approximates this integral as a discrete sum of a finite number of terms

$$\int f(u)e^{-u^2}\mathrm{d}z \approx \sum_{k=1}^{K} w_k f(u_k) \tag{3.8}$$

with masses

$$w_k = \frac{2^{K-1}K!\sqrt{\pi}}{K^2[H_{K-1}(u_k)]^2}$$

and mass points u_k, $k = 1, \ldots, K$. The mass points u_k are the zeros of the Kth order Hermite polynomial $H_K(u)$. The approximation is exact if $f(u)$ is a polynomial of order $2K-1$.

Definition 3.1 (Hermite polynomials) *The standardized Hermite polynomials are defined by*

$$H_K(u) = (-1)^K e^{u^2} \frac{\partial^K}{\partial u^K} e^{-u^2}$$

with $H_K(u)$ satisfies the differential equation

$$y'' - 2uy' + 2Ky = 0.$$

Remark 3.1 *The first 7 Hermite polynomials are*

$$\begin{aligned} H_0(u) &= 1 \\ H_1(u) &= x \\ H_2(u) &= x^2 - 1 \\ H_3(u) &= x^3 - 3x \\ H_4(u) &= x^4 - 6x^2 + 3 \\ H_5(u) &= x^5 - 10x^3 + 15x \\ H_6(u) &= x^6 - 15x^4 + 45x^2 - 15. \end{aligned}$$

If the integral is of the form

$$\int f(u)\phi(u)\mathrm{d}u = \int f(u)\frac{1}{\sqrt{2\pi}} e^{-u^2/2}\mathrm{d}u,$$

a linear transformation of u will be used to convert the function $\phi(u)$ into e^{-u^2}. By transforming $z = u/\sqrt{2}$, therefore $u = \sqrt{2}z$ and $du = \sqrt{2}dz$ the above integral can be approximated by

$$\int \frac{1}{\sqrt{\pi}} f(\sqrt{2}z) e^{-z^2}\mathrm{d}x \approx \sum_{k=1}^{K} \pi_k f(z_k) \tag{3.9}$$

where $z_k = \sqrt{2}u_k$ are the transformed mass points and $\pi_k = w_k/\sqrt{\pi}$ the transformed masses.

Using Gaussian quadrature and writing $f(z, \theta_0) = \phi(z)$ in equation (3.7), $f(y, \theta_0)$ can be approximated by

$$f(y, \theta_0) = \int f(y|z, \theta_0)\phi(z)\mathrm{d}z \approx \sum_{k=1}^{K} f(y|z_k, \theta_0)\pi_k, \tag{3.10}$$

where z_k are the transformed mass points and π_k the associated masses, as in (3.9). Using Gaussian quadrature, the other integral in (3.7) can be approximated by

$$\int \log f(y,z,\theta) f(y|z,\theta_0)\phi(z)\mathrm{d}z \approx \sum_{k=1}^{K} \log f(y,z_k,\theta) f(y|z_k,\theta_0)\pi_k. \tag{3.11}$$

With (3.10) and (3.11) we derive the following approximation for $Q(\theta|\theta_0)$:

$$\begin{aligned} Q(\theta|\theta_0) &\approx \sum_{i=1}^{n} \frac{\sum_{k=1}^{K} \log f(y_i,z_k,\theta) f(y_i|z_k,\theta_0)\pi_k}{\sum_{j=1}^{K} f(y_i|z_j,\theta_0)\pi_j} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik} \log f(y_i,z_k,\theta) \end{aligned} \tag{3.12}$$

with weights

$$w_{ik} = \frac{\pi_k f(y_i|z_k,\theta_0)}{\sum_{j=1}^{K} f(y_i|z_j,\theta_0)\pi_j}. \tag{3.13}$$

Since the weights w_{ik} are evaluated at $\theta = \theta_0$, they are fixed terms for the subsequent maximization.

By writing

$$f(y,z,\theta) = f(y|z,\theta)\phi(z)$$

it follows that

$$\log f(y,z,\theta) = \log f(y|z,\theta) + \log \phi(z)$$

and approximation (3.12) gets

$$Q(\theta|\theta_0) \approx \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}(\log f(y_i|z_k,\theta) + \log \pi_k). \tag{3.14}$$

The M-step of the EM-algorithm consists of setting the partial derivatives of $Q(\theta|\theta_0)$ equal to zero and solving the resulting equations. Furthermore, maximizing (3.14) is equivalent to a weighted maximum likelihood estimation with weights given in (3.13). This is the same as fitting a regression model with nK observations with an y-variable given as

$$y = \underbrace{(y_1,\ldots,y_n,y_1,\ldots,y_n,\ldots,y_1,\ldots,y_n)}_{\text{K copies}}$$

and K copies of the explanatory variables. The resulting linear predictor can be written as

$$\eta_{ik} = x_i^T \beta + \sigma z_k,$$

with known quadrature points and unknown parameters β and σ. Therefore, for every iteration of the EM-algorithm a weighted maximum likelihood estimation of a generalized linear model has to be computed.

Table 3.1 shows the resulting structure of the data set, see also Friedl (1998).

y	w	β_1	$\dots$	β_p	σ_z
y_1	w_{11}	x_{11}	$\dots$	x_{1p}	z_1
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	w_{n1}	x_{n1}	$\dots$	x_{np}	z_1
y_1	w_{12}	x_{11}	$\dots$	x_{1p}	z_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	w_{n2}	x_{n1}	$\dots$	x_{np}	z_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_1	w_{1K}	x_{11}	$\dots$	x_{1p}	z_K
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_n	w_{nK}	x_{n1}	$\dots$	x_{np}	z_K

Table 3.1: Structure of a model with normally distributed random intercept

3.2.2 NPML Estimation

If no assumption can be made for the distribution of z, Aitkin (1994) or Aitkin (1996) proposed to use a non-parametric maximum likelihood approach. Thus, the linear predictor can be written as

$$\eta_i = x_i^T \beta + z_i, \qquad z_i \overset{iid}{\sim} F(z) \tag{3.15}$$

with $F(z)$ being any unknown distribution function with zero mean, which has to be estimated. An analogous approach, as in the Gaussian quadrature before, results in the same approximation as (3.14):

$$Q(\theta|\theta_0) \approx \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} (\log f(y_i|z_k, \theta) + \log \pi_k)$$

with weights

$$w_{ik} = \frac{\pi_k f(y_i|z_k, \theta_0)}{\sum_{j=1}^{K} f(y_i|z_j, \theta_0)\pi_j},$$

where z_k and π_k are now unknown. Therefore, the following estimates for π_ks are used: With the condition $\sum_{k=}^{K} \pi_k = 1$ and a Lagrange multiplier it follows that

$$\frac{\partial}{\partial \pi_k}\left(Q(\theta|\theta_0) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)\right) = \frac{1}{\pi_k}\sum_{i=1}^{n} w_{ik} - \lambda.$$

Setting this equation equal to zero yields to

$$\pi_k = \sum_{i=1}^{n} \frac{w_{ik}}{\lambda}$$

and summation over all k gives

$$1 = \sum_{k=1}^{K}\pi_k = \sum_{i=1}^{n}\sum_{k=1}^{K}\frac{w_{ik}}{\lambda} = \frac{n}{\lambda}.$$

Therefore $\lambda = n$ and we get the estimation for π_k,

$$\hat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_{ik}.$$

Moreover, the linear predictor can be written as

$$\eta_{ik} = x_i^T\beta + z_k,$$

with unknown mass points z_k which have to be estimated. With the help of a k-level factor, we write the linear predictor as

$$\eta_{ik} = x_i^T\beta + z_1 \cdot 0 + \cdots + z_{k-1} \cdot 0 + z_k \cdot 1 + z_{k+1} \cdot 0 + \cdots + z_K \cdot 0,$$

where the new data set has nK observations. Further, it is important to note that in the NPML case, no difference between the intercept and z_1 can be made, if and only if the explanatory variable x contains an intercept. Table 3.2 shows the enlarged data set with nK responses.

Therefore, at each iteration of the M-step a weighted maximum likelihood estimation of a GLM hast to be calculated and in the E-step the weights w_{ik} have to be updated.

y	w	β_1	$\ldots$	β_p	z_1	z_2	$\ldots$	z_K
y_1	w_{11}	x_{11}	$\ldots$	x_{1p}	1	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{n1}	x_{n1}	$\ldots$	x_{np}	1	0	$\vdots$	0
y_1	w_{12}	x_{11}	$\ldots$	x_{1p}	0	1	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{n2}	x_{n1}	$\ldots$	x_{np}	0	1	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_1	w_{1K}	x_{11}	$\ldots$	x_{1p}	0	0	$\vdots$	1
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{nK}	x_{n1}	$\ldots$	x_{np}	0	0	$\vdots$	1

Table 3.2: Response structure of a NPML model.

In the NPML approach the unknown distribution of the random effects is approximated by a discrete mixture, yielding estimated mass points $\hat{z}_1, \ldots, \hat{z}_K$ and estimated masses $\hat{\pi}_1, \ldots, \hat{\pi}_K$. Thus, a NPML estimate for the distribution of the random effects is derived through the K pairs

$$\hat{f}(z) = \{(\hat{z}_1, \hat{\pi}_1), \ldots, (\hat{z}_K, \hat{\pi}_K)\},$$

if convergence occur.

3.2.3 R-Function for the Quadrature Points

The package `npmlreg` provides a function `gqz` which calculates the Gaussian quadrature points for the normal distribution, by using abscissas and weights for Hermite integration, compare to z_k and π_k in (3.9). For illustration purposes, the location of the mass points and their corresponding weights for $K = 2, 3, 4, 5, 6$ are shown in Table 3.3.

```
> library(npmlreg)
> gqz(3)
        location    weight
  1  1.732051e+00 0.1666667
  2  1.256074e-15 0.6666667
  3 -1.732051e+00 0.1666667
```

The package `glmmML` provides another function called `ghq` for calculating mass points and their weights, which are slightly different to those used in the package `npmlreg`. The reason for that is, that `ghq` displays the values of u_k and w_k, compare to (3.8). By

	mass points	masses
$k = 2$	1	0.5
	-1	0.5
$k = 3$	$1.732051e + 00$	0.1666667
	$1.256074e - 15$	0.6666667
	$-1.732051e + 00$	0.1666667
$k = 4$	2.3344142	0.04587585
	0.7419638	0.45412415
	-0.7419638	0.45412415
	-2.3344142	0.04587585
$k = 5$	$2.856970e + 00$	0.01125741
	$1.355626e + 00$	0.22207592
	$1.256074e - 15$	0.53333333
	$-1.355626e + 00$	0.22207592
	$-2.856970e + 00$	0.01125741
$k = 6$	3.3242574	0.002555784
	1.8891759	0.088615746
	0.6167066	0.408828470
	-0.6167066	0.408828470
	-1.8891759	0.088615746
	-3.3242574	0.002555784

Table 3.3: Location of mass points and their corresponding masses calculated by `gqz`.

transforming the values as in (3.9) one gets the resulting values of `gqz`. The usage of the function `ghq` is explained in A.5.2. Table 3.4 shows the location of the mass points and the corresponding masses for $K = 2, 3, 4, 5, 6$.

```
> library(glmmML)
> ghq(3,FALSE)
  $weights
  [1] 0.295409 1.181636 0.295409

  $zeros
  [1]  1.224745  0.000000 -1.224745
```

3.2.4 R-Functions for Normal Random Effect Models

Some functions in different packages support the fitting of overdispersed models by maximum likelihood and numerical integration via Gaussian quadrature. The functions considered here are `glmmML` provided by the package `glmmML` proposed by Broström (2009)

	mass points	masses
$k = 2$	0.707106	0.886227
	−0.7071068	0.886227
$k = 3$	1.224745	0.295409
	0.000000	1.181636
	−1.224745	0.295409
$k = 4$	1.6506801	0.08131284
	0.5246476	0.80491409
	−0.5246476	0.80491409
	−1.6506801	0.08131284
$k = 5$	2.0201829	0.01995324
	0.9585725	0.39361932
	0.0000000	0.94530872
	−0.9585725	0.39361932
	−2.0201829	0.01995324
$k = 6$	2.3506050	0.00453001
	1.3358491	0.15706732
	0.4360774	0.72462960
	−0.4360774	0.72462960
	−1.3358491	0.15706732
	−2.3506050	0.00453001

Table 3.4: Location of mass points and their corresponding masses calculated by `ghq`

and `alldist` provided by the package `npmlreg` proposed by Einbeck, Darnell, and Hinde (2009).

Example

In the following we will fit the `dja` data, given in Table 2.4, by the two R-functions mentioned above. The `dja` data contains 75 observations with a factor `group` of two levels, `CTRL` and `TREAT`, the numeric vector `trisk` indicating the exposition time to mortality and the values `n` and `y`, where `n` indicates the number of animals exposed to mortality and `y` the number of deaths.

As described in Section 2.7.1, an appropriate model has the mean number of deaths divided by `trisk` modeled by the grouping factor. Moreover, the number of deaths can be assumed to follow a Poisson distribution. Using the log-link results in

$$\eta = \log \frac{\mu}{\text{trisk}} = \beta_0 + \beta_1(\text{group=TREAT}).$$

In Section 2.7.1 we fitted a log-linear Poisson model, given by

```
> library(aod)
> data(dja)
> attach(dja)
> summary(glm(y~group+offset(log(trisk)), family=poisson))

  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept)  -0.6975     0.1170  -5.960 2.53e-09 ***
  groupTREAT   -0.8754     0.1712  -5.112 3.19e-07 ***

      Null deviance: 162.67  on 74  degrees of freedom
  Residual deviance: 136.86  on 73  degrees of freedom
  AIC: 271.33
```

which clearly indicates overdispersion, as the deviance of the fitted model (136.86) is very large compared to its df (73). The most plausible explanation is that there are some unobserved variables varying over the data. In the following sections we consider therefore functions which include a random effect into the linear predictor, to model the unobserved variables.

Function `alldist`

Hinde and Demetrio (1998) proposed to model the unexplained variation with random effects z_i for every response y_i, that are assumed to be independent normally distributed, i.e.

$$\begin{aligned} y_i|z_i &\overset{ind}{\sim} Poisson(\mu_i) \\ z &\overset{iid}{\sim} N(0,1), \end{aligned}$$

and

$$\eta_i = \log \frac{\mu_i}{\text{trisk}_i} = \beta_0 + \beta_1(\text{group}_i) + \sigma z_i.$$

Since we assumed that the z_i are independent standard normally distributed, a large value of K results into a good approximation of the standard normal pdf. Calling the function `alldist` for the `dja` data with a starting number of thirty quadrature points yields

```
> library(npmlreg)
> m.alldist.gq<-alldist(y~group, family=poisson(link=log),data=dja,
```

```
                offset=log(trisk), random=~1, k=30, random.distribution="gq")
> summary(m.alldist.gq)

Coefficients:
              Estimate Std. Error   t value
(Intercept) -0.8077248  0.1266790 -6.376154
groupTREAT  -0.9136324  0.1714024 -5.330335
z            0.5957891  0.0867942  6.864389

Random effect distribution - standard deviation:          0.5957891


-2 log L:           250.2      Convergence at iteration  7
```

with a smaller intercept and a smaller slope coefficient than under the log-linear Poisson model. The standard deviation parameter for the random effect is estimared by 0.596. Furthermore, Figure 3.1 shows the convergence of the EM- algorithm.

In order to compare the deviance of the random effect model to the log-linear Poisson model we consider the following call

```
> m.aldist.gq$deviance
 [1] 119.7591
```

which is derived by

$$D(y, \mu) = -2 \left(\log f(y, \mu) - \log f(y, y) \right).$$

The full or saturated model is calculated by the Poisson pmf at $\mu = y$, i.e.

$$\log f(y, y) = -y + y \log y - \log y!$$

as

```
> sum(dpois(x=y, lambda=y, log=TRUE))
  [1] -65.2358
```

To derive the value of $\log f(y, \mu)$ we have to calculate

$$\sum_{i=1}^{n} \left(\log \left(\sum_{k=1}^{K} f(y_i|z_k)\pi_k \right) \right).$$

With

$$y_i|z_k \overset{ind}{\sim} Poisson(\mu_{ik})$$

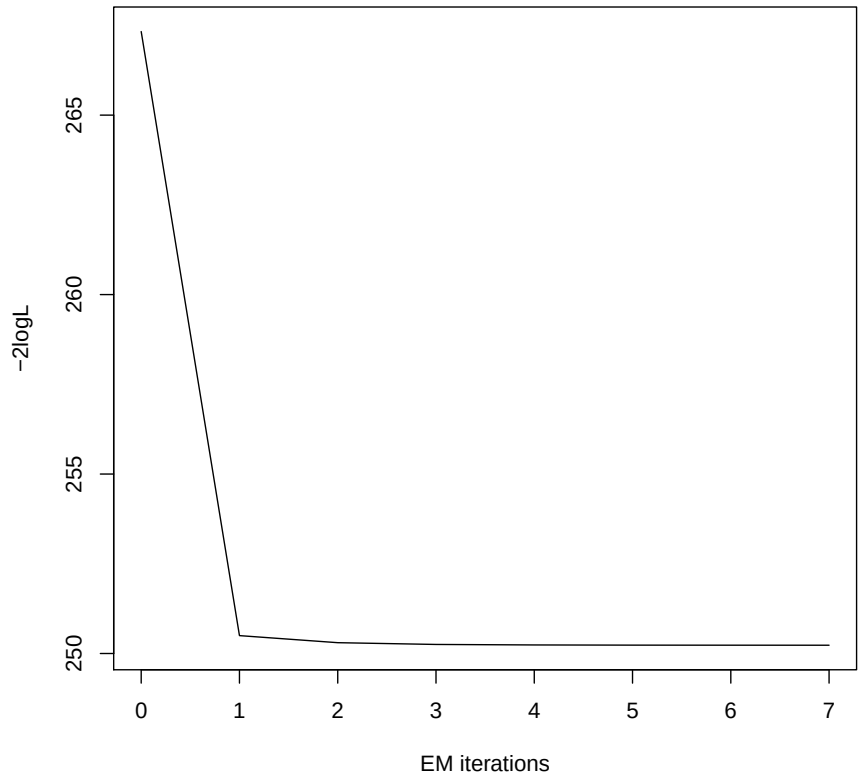

Figure 3.1: `dja` data: Convergence of the EM-algorithm for K=30.

it follows that

$$f(y_i|z_k) = \frac{e^{-\mu_{ik}}\mu_{ik}^{y_i}}{y_i!},$$

where

$$\mu_{ik} = \exp\left(\beta_0 + \beta_1 \log(\text{group}_i) + \log(\text{trisk}_i) + \sigma z_k\right).$$

Evaluating $\log f(y, \mu)$ in R gives:

```
> intercept<-coefficients(m.alldist.gq)[1]
> beta1<-coefficients(m.alldist.gq)[2]
> sigma<-coefficients(m.alldist.gq)[3]

> group.1.0<-rep(0,75);
> for(i in 1:75)
```

```
    if(group[i]=="TREAT"){group.1.0[i]<-1}

> mu<-matrix(ncol=16, nrow=75)
> for(j in 1:16)
  {
  mu[,j]<-exp(beta1*group.1.0+intercept+sigma*gqz(30)[j,1]+log(trisk))
  }

> f_y_mu<-0
> for(j in 1:16)
  {
  f_y_mu<-exp(-mu[,j])*mu[,j]^y*gqz(30)[j,2]/factorial(y)+f_y_mu
  }
```

and yields the following result

```
> sum(log(f_y_mu))
 [1] -125.1155
```

and therefore the deviance is $-2 \cdot (-125.12 + 65.24) = 119.76$, which is the same as in the deviance of `m.alldist.gq` on page 71. Important to note is that several mass points with masses less than 0.000001 will be omitted. For this reason, we only have 16 "real" mass points with noticeable masses. Table 3.5 shows the parameter estimates for various values of K.

K	intercept	group	sigma	dev.
1	−0.6974	−0.8754	−	136.85
30	−0.8077	−0.9136	0.5958	119.76
40	−0.8076	−0.9138	0.5960	119.76
50	−0.8076	−0.9138	0.5960	119.76
70	−0.8076	−0.9138	0.5960	119.76

Table 3.5: Parameter estimates and deviances calculated by `alldist` for the `dja` data with increasing values of K.

The normal mass points model shows reasonable stability in the parameter estimates. In the following we will fit the same data with another function than `alldist`.

Function `glmmML`

Another function for random effect models is the function `glmmML`, see also Broström (2003). In fact, `glmmMl` is designed to model grouped data with a random effect for every group, but by assuming n (the number of observations) groups, this function can also be

used to model overdispersed data. If the number of quadrature points is set to 1, Laplace approximation, as in approximation (3.11) is used. We consider the same model as for the function `alldist`,

$$\eta = \log \frac{\mu}{\text{trisk}} = \beta_0 + \beta_1(\text{group=TREAT}) + \sigma z.$$

For the call of `glmmML` one needs to define a certain cluster, indicating which of the responses are correlated. By defining a cluster `id` with n different groups for n responses, we get

```
> id<-factor(1:75)
> m.glmmML.gq<-glmmML(y~group+offset(log(trisk)), family=poisson,
               cluster=id, method="ghq", n.points=30)

> summary(m.glmmML.gq)

               coef se(coef)      z Pr(>|z|)
(Intercept) -0.8076   0.1847 -4.372 1.23e-05
groupTREAT  -0.9151   0.2441 -3.749 1.78e-04

Scale parameter in mixing distribution:  0.5994 gaussian
Std. Error:                              0.1446

Residual deviance: 119.8 on 72 degrees of freedom       AIC: 125.8
```

with thirty quadrature points. To compare this with `alldist`, Table 3.6 shows parameter estimates and deviances for increasing K.

K	intercept	group	sigma	dev.
1	−0.8118	−0.9153	0.6094	119.26
30	−0.8076	−0.9151	0.5994	119.76
40	−0.8076	−0.9151	0.5994	119.76
50	−0.8077	−0.9150	0.5994	119.76
70	−0.8076	−0.9151	0.5994	119.76

Table 3.6: Parameter estimates and deviances calculated by `glmmML` for the `dja` data with increasing values of K.

Compared to Table 3.5, Table 3.6 gives nearly the same estimates with an equal intercept and a little smaller slope and a larger sigma estimate. This can be due to different implementations and therefore different stopping criteria of the EM-algorithm. Moreover, the Laplace approximation results into quite the same estimates as the Gauss-Hermite quadrature. Further, the function `glmmML` stabilizes for more than 5 quadrature points whereas

the function `alldist` stabilizes for more than 30 quadrature points. This indicates that `glmmML` might use an adaptive Gauss-Hermite quadrature technique. In contrast to the Gauss-Hermite quadrature, where the transformed mass points are symmetrical around the mean of the weight function, the adaptive Gauss-Hermite quadrature approximates the integrand by the normal density centered at the mode of the integrand. Here this weight function is the Gaussian pdf with mean 0 and variance 1. An adaptive Gauss-Hermite quadrature method often decreases the number of required quadrature points, especially for functions with maxima far from zero.

3.2.5 R-Function for NPML Estimation

The function `alldist` can also be used to model overdispersed data with a NPML approach, see also Einbeck and Hinde (2009).

Function alldist

In the following we will again fit the `dja` data, given in Table 2.4. The call of the function `alldist` differs only with respect to the choice of the `random.distribution`. Now this has to be set to `np`.

```
> summary(m.alldist.np<-alldist(y~group, family=poisson(link=log),data=dja,
                offset=log(trisk), random=~1, k=3, random.distribution="np"))
  Coefficients:
               Estimate Std. Error    t value
  groupTREAT -1.1539315  0.1740137 -6.6312692
  MASS1      -2.4341566  0.3925273 -6.2012411
  MASS2      -0.3840778  0.1365674 -2.8123688
  MASS3       0.1780657  0.1813111  0.9821004

  Mixture proportions:
      MASS1      MASS2      MASS3
  0.2237459  0.6539367  0.1223174

  Random effect distribution - standard deviation:           0.909377
  -2 log L:           244.1     Convergence at iteration  15
```

The deviance of `m.alldist.np` is

```
> deviance(m.alldist.np)
[1] 113.649
```

This value is derived through

```
> beta1<-coefficients(m.alldist.np)[1]
> z1<-coefficients(m.alldist.np)[2]
> z2<-coefficients(m.alldist.np)[3]
> z3<-coefficients(m.alldist.np)[4]

> mu1<-exp(beta1*group.1.0+z1+log(trisk))
> mu2<-exp(beta1*group.1.0+z2+log(trisk))
> mu3<-exp(beta1*group.1.0+z3+log(trisk))


> sum(log(exp(-mu1)*mu1^y*m.dja.alldist.np$masses[1]/factorial(y)
+        +exp(-mu2)*mu2^y*m.dja.alldist.np$masses[2]/factorial(y)
+        exp(-mu3)*mu3^y*m.dja.alldist.np$masses[3]/factorial(y)))
[1] -122.0604

> deviance<--2*(f_y_mu-sum(dpois(x=y, lambda=y, log=TRUE)))
  113.649
```

since we use again the Poisson pmf for calculating the value of the saturated model. Moreover, Figure 3.2 shows the convergence of the log-likelihood maximum under the EM-algorithm and the estimation of the corresponding mass points. There we can see that it takes 15 iterations for the convergence of the EM-algorithm and furthermore, it is graphically displayed how the mass points converge to their estimates.

To compare the results of the Gaussian quadrature with the NPML estimation, Table 3.7 shows the corresponding parameter estimates and the different deviances. The estimated intercept can be calculated as

$$\hat{\beta}_0 = \hat{\mathbb{E}}(z) = \sum_{k=1}^{K} \hat{\pi}_k \hat{z}_k$$

and in R as

```
> sum(m.alldist.np$mass.points*m.alldist.np$masses)
```

For a short overview Table 3.8 shows results of the log-linear Poisson model compared to `alldist` with Gaussian quadrature, `glmmML` and `alldist` with the non-parametric approach.

The deviance for the non-parametric model is smaller than for the Gaussian quadrature, the slope is also smaller but the intercept is somewhat larger.

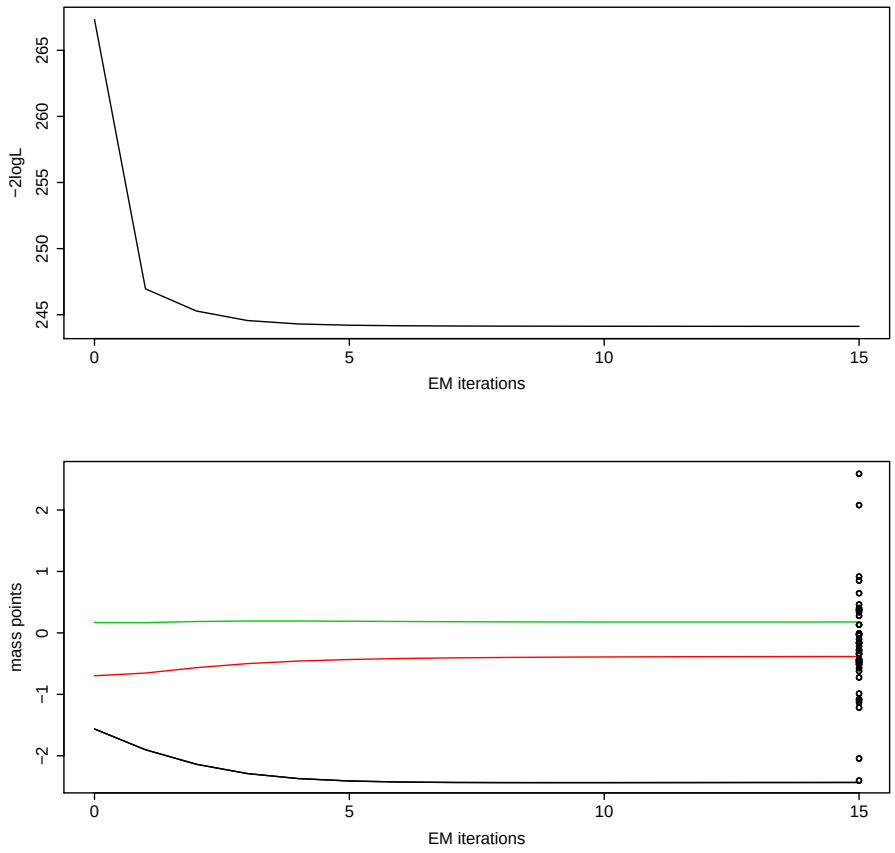

Figure 3.2: Convergence of the EM-algorithm for K=3 and the estimation of the corresponding mass points.

3.3 Variance Component Models

Suppose the observations arise through repeated measurement or are grouped in a different way, e.g. in a control group and a treatment group. Therefore, a model which includes a random effect for each group in the linear predictor would be appropriate. These models are called variance component models.

Thus, the theory of overdispersion ($n_i = 1 \ \forall i$) models can be used to fit variance component models, where the count of observations of the same group is bigger than one, $n_i \geq 1$, compare to Friedl (1998) or Aitkin et al. (2009).
The linear predictor is now

$$\eta_{ij} = x_{ij}^T \beta + \sigma z_i \quad \text{with} \quad z_i \overset{iid}{\sim} N(0,1)$$

K	intercept	group	dev.
1	0.6974	−0.8754	136.85
3	−0.7740	−1.1539	113.65
5	−0.7733	−1.1558	113.65
7	−0.7917	−1.1387	113.57
9	−0.8000	−1.1252	113.56
11	−0.7831	−1.1269	113.56
21	−0.7769	−1.1380	113.57
31	−0.7858	−1.1261	113.56
41	−0.7815	−1.1317	113.56

Table 3.7: Parameter estimates and deviances of `alldist` for the `dja` data with increasing K and NPML estimation.

	K	intercept	group	sigma	dev.
m.glm.poisson	1	−0.6974	−0.8754	-	136.85
m.alldist.gq	40	−0.8076	−0.9138	0.5960	119.76
m.glmmML.gq	7	−0.8076	−0.9151	0.5994	119.76
m.alldist.np	11	−0.7831	−1.1269	0.9277	113.56

Table 3.8: Comparison of the four models, `m.glm.poisson`, `m.alldist.gq`, `m.glmmML` and `m.alldist.np`.

for the Gaussian quadrature and

$$\eta_{ij} = x_{ij}^T \beta + z_i \quad \text{with} \quad z_i \overset{iid}{\sim} G \tag{3.16}$$

for the non-parametric estimation approach. Meaning that all observations of the same group or cluster share the same random intercept. The Q function is nearly the same as in the overdispersed case, with the additional assumption that observations within the same group are conditionally independent. Therefore, the conditional pdf is

$$f(y_i|z_i) = \prod_{j=1}^{n_i} = f(y_{ij}|z_i)$$

and the Q function for the Gaussian quadrature and for the non-parametric approach is approximately

$$Q(\theta|\theta_0) \approx \sum_{i=1}^{n} \frac{\sum_{k=1}^{K} \log f(y_i, z_k, \theta) f(y_i|z_k, \theta_0)\pi_k}{\sum_{j=1}^{K} f(y_i|z_j, \theta_0)\pi_j}.$$

Thus, it follows that

$$Q(\theta|\theta_0) \approx \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik} \log f(y_i, z_k|\theta) \tag{3.17}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{n_i} w_{ik} \left(\log f(y_{ij}|z_k, \theta) + \log \pi_k\right), \tag{3.18}$$

with weights

$$w_{ik} = \frac{\pi_k \prod_{j=1}^{n_i} f(y_{ij}|z_k, \theta_0)}{\sum_{l=1}^{K} \prod_{j=1}^{n_i} f(y_{ij}|z_l, \theta_0)\pi_l}.$$

The most important difference to the Q function in (3.14) is the additional summation over the elements of the same cluster. In the non-parametric approach the π's are assumed to be unknown with $\sum_{k=1}^{K} \pi_k = 1$. Using a Lagrange multiplier λ we get

$$\frac{\partial}{\partial \pi_k}\left(Q(\theta|\theta_0) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)\right) = \frac{1}{\pi_k}\sum_{i=1}^{n} n_i w_{ik} - \lambda.$$

Therefore, $\hat{\pi}_k = \sum_{i=1}^{n} n_i \hat{w}_{ik}/\hat{\lambda}$ and with $\sum_{k=1}^{K} w_{ik} = 1$ follows that

$$\sum_{k=1}^{K} \hat{\pi}_k = \sum_{i=1}^{n} \frac{n_i}{\hat{\lambda}}.$$

By maximizing the criterion (3.17) the following estimate for π_k results,

$$\hat{\pi}_k = \frac{\sum_{i=1}^{n} n_i \hat{w}_{ik}}{\sum_{i=1}^{n} n_i}.$$

Again, for maximizing approximation (3.17), a weighted maximum likelihood estimation has to be calculated, which is related to the structure of the design matrix in the overdispersed case. Due to the fact that there are n_i replications within the ith group, the row which corresponds to the first observation with $k = 2$, is replaced by the matrix given in Table 3.9 and in case of an unknown distribution the row is replaced by the matrix given in Table 3.10.

3.3.1 R-Functions for Variance Component Models with Normally Distributed Random Intercept

In the following we will have a closer look at two R-functions to fit variance component models. As said before, the function `glmmML` is mainly designed to model variance component models. The other function considered here is `allvc`, proposed by Einbeck et al.

y	w	β_1	$\dots$	β_p	σ
y_{11}	w_{12}	x_{111}	$\dots$	x_{11p}	z_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_{1n_1}	w_{12}	x_{1n_11}	$\dots$	x_{1n_1p}	z_2

Table 3.9: Structure of a variance component model with normally distributed random intercept.

y	w	β_1	$\dots$	β_p	z_1	z_2	$\dots$	z_{n_1}
y_{11}	w_{12}	x_{111}	$\dots$	x_{11p}	1	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_{1n_1}	w_{12}	x_{1n_11}	$\dots$	x_{1n_1p}	1	0	$\dots$	0

Table 3.10: Response structure of variance component model with the non-parametric approach.

(2009). We will consider an example of Irish suicide rates, where the data is also given in the package `npmlreg`. This dataset describes the mortality due to suicide and intentional self-harm in the Republic of Ireland from $1989 - 1998$, which is obtained from the All Ireland Mortality Database. This data were investigated using Poisson mixed models by Sofroniou, Einbeck, and Hinde (2006). In this study, the Republic of Ireland is divided into 13 "health regions", including the 8 former health boards which existed during this period (the health board system was initially created in 1979 with 8 health boards and was reformed in 1999 from 8 to 11 health boards) and the cities Cork, Dublin, Galway, Limerick and Waterford, extracted from these health boards. Figure 3.3 graphically displays the regions, also given in Sofroniou et al. (2006). The data consists of 104 observations with variables `death`, `age`, `sex`, `Region`, `ID` and `pop`. `pop` is a numeric vector, giving the population sizes. `ID` is a factor denoting the different Regions and `age` is a factor with levels `1` $(0-29)$, `2` $(30-39)$, `3` $(40-59)$, `4` $(60+\text{years})$. The data is given in Table 3.11.

nr	Region	ID	pop	death	sex	age
1	Cork	1	31923	6	0	1
2	Cork	1	31907	52	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	Cork	1	8299	22	1	4
9	Dublin	2	117575	33	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
16	Dublin	2	33385	37	1	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
103	WHB - Gal.	13	29832	13	0	4
104	WHB - Gal.	13	27260	58	1	4

Table 3.11: Irish suicide data

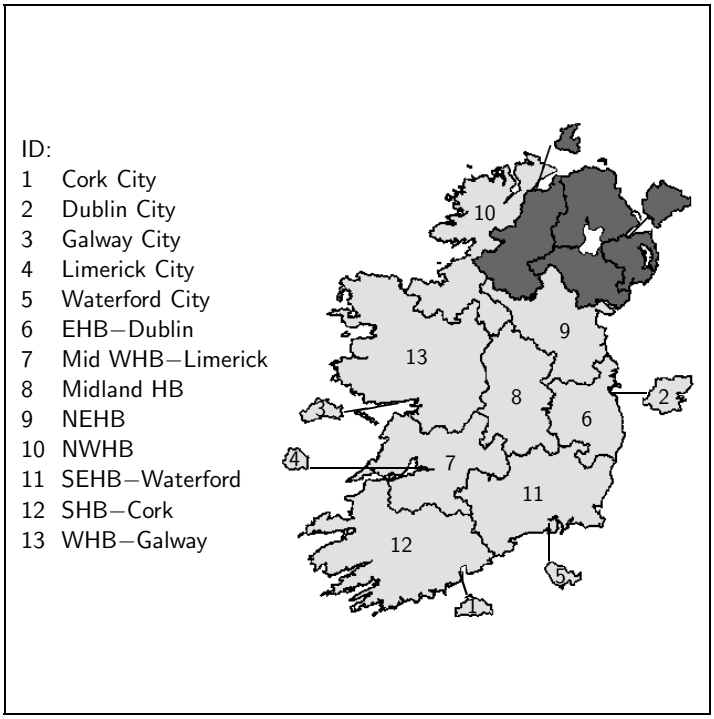

Figure 3.3: Map of health boards and cities for the Republic of Ireland. The excluded regions of Northern Ireland are shown in dark grey. The '-' sign indicates that a city is excluded from its health board.

A simple log-linear Poisson model for the number of deaths in relation to `sex` and `age` with an offset of `log(pop)` results into

```
> summary(m.glm.irish<-glm(death~sex+age+offset(log(pop)), data=irlsuicide,
  family=poisson))

  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept) -8.20092    0.04348 -188.61   <2e-16 ***
  sex1         1.44468    0.04092   35.30   <2e-16 ***
  age2         0.77089    0.04540   16.98   <2e-16 ***
  age3         0.72734    0.04053   17.95   <2e-16 ***
  age4         0.50125    0.04932   10.16   <2e-16 ***

      Null deviance: 2299.12  on 103  degrees of freedom
  Residual deviance:  290.97  on  99  degrees of freedom
  AIC: 803.76
```

which shows a large residual deviance of 290.97 on 99 df. Therefore, we will add a random intercept for every region. The following model has an unobserved common random effect z_i for each region i, $i = 1, \ldots, 13$. By writing the linear predictor as

$$\eta_{ij} = \log\left(\frac{\mu_{ij}}{\mathtt{pop}_{ij}}\right) = \beta_0 + \beta_1(\mathtt{sex1}_{ij}) + \beta_2(\mathtt{age2}_{ij}) + \beta_3(\mathtt{age3}_{ij}) + \beta_4(\mathtt{age4}_{ij}) + \sigma z_i$$

with $i = 1, \ldots, n,\ j = 1, \ldots, n_i$, we get a model for the correlated data.

Function `allvc`

The function `allvc` provided by the package `npmlreg` is able to fit grouped models with a group specific random intercept. By setting `random.distributin="gq"` we first approximate the Q-function with Gaussian quadrature and forty quadrature points, to get an acceptable approximation of the normal pdf.

```
> summary(m.allvc.irish.gq<-allvc(death~sex+age, random=~1|ID, offset=log(pop),
  k=40, data=irlsuicide, family=poisson, random.distribution="gq"))

  Coefficients:
               Estimate Std. Error     t value
  (Intercept) -8.1599326 0.04353230 -187.445480
  sex1         1.4405493 0.04092213   35.202210
  age2         0.7741483 0.04540167   17.051096
  age3         0.7234966 0.04053309   17.849529
```

```
  age4          0.4742595 0.04937286    9.605672
  z             0.1549697 0.01467215   10.562174

  Random effect distribution - standard deviation:           0.1549697

  -2 log L:           703.3      Convergence at iteration  39
```

By calculating the saturated model as

```
> sum(dpois(x=death, lambda=death, log=TRUE))
[1] -251.3939
```

we get a deviance of $-2(-351.639 + 251.3939) = 200.49$, which is much smaller than for the log-linear Poisson model without random effects. Table 3.12 shows the parameter estimates for increasing K.

K	int.	sex1	age2	age3	age4	sigma	dev.
1	−8.20	1.44	0.77	0.73	0.50	-	290.97
40	−8.16	1.44	0.77	0.72	0.47	0.15	200.49
50	−8.16	1.44	0.77	0.72	0.47	0.16	200.51
70	−8.16	1.44	0.77	0.72	0.47	0.16	200.51

Table 3.12: Parameter estimates and deviances of `allvc` for the `irlsuicide` data with increasing K and Gaussian quadrature.

We also tried different values of K, especially values which are smaller than 40, but by doing so, some weird characteristics occur. The parameter estimates for β are nearly the same, in all iterations of K, but the deviance and the estimate of `sigma` differ, especially for odd values of K compared to even values of K. Odd values of K lead to comparable results with other odd values of K, and vice versa for even values of K. But by considering a sequence of odd and even values, the differences between single steps are large. However, for a choice of K larger than 40, it stabilizes and odd and even values of K lead to the same result. Implying that Gaussian quadrature is not useful for a poor approximation of the normal pdf.

Fitting the same model with an NPML approach and further `random.distribution="np"` yields into

```
> summary(m.allvc.irish.np<-allvc(death~sex*age, random=~1|ID, offset=log(pop),
  k=3, data=irlsuicide, family=poisson, random.distribution="np"))

  Coefficients:
          Estimate Std. Error    t value
  sex1   1.4412361 0.04092216   35.21896
```

```
age2   0.7743245 0.04540173   17.05496
age3   0.7243116 0.04053299   17.86968
age4   0.4733938 0.04940511    9.58188
MASS1 -8.5184120 0.05666877 -150.31934
MASS2 -8.1596255 0.04514065 -180.76002
MASS3 -7.9461554 0.05555707 -143.02689

Mixture proportions:
    MASS1      MASS2      MASS3
0.0953224  0.7151025  0.1895751

Random effect distribution - standard deviation:          0.1444652

-2 log L:           697.2     Convergence at iteration  5
```

with nearly equal estimates as for the Gaussian quadrature. The intercept can again be calculated as

$$\hat{\beta}_0 = \sum_{k=1}^{K} \hat{z}_k \hat{\pi}_k.$$

Thus, Table 3.13 gives parameter estimates for increasing K.

K	intercept	sex1	age2	age3	age4	dev.
1	−8.20	1.44	0.77	0.73	–	290.97
3	−8.15	1.44	0.77	0.72	0.47	194.44
5	−8.15	1.44	0.77	0.72	0.47	193.83
10	−8.15	1.44	0.77	0.72	0.47	193.98

Table 3.13: Parameter estimates and deviances of `allvc` for the `irlsuicide` data with increasing K and NPML estimation

As investigated in Sofroniou et al. (2006), for the model with $K = 3$ the deviance has dropped compared to the log-linear Poisson model and does not fall significantly further for increasing K.

Function `glmmML`

Fitting the same model with the Gaussian quadrature, `glmmML` and forty quadrature points results into

```
> summary(m.glmmML.irish<-glmmML(death~sex+age+offset(log(pop)),cluster=ID,
  data=irlsuicide, n.points=40, family=poisson)
```

```
               coef se(coef)        z Pr(>|z|)
(Intercept) -8.1589  0.06262 -130.286        0
sex1         1.4406  0.04093   35.195        0
age2         0.7741  0.04540   17.050        0
age3         0.7235  0.04054   17.847        0
age4         0.4743  0.04944    9.594        0

Scale parameter in mixing distribution:  0.1554 gaussian
Std. Error:                              0.03626

Residual deviance: 200.5 on 98 degrees of freedom       AIC: 212.5
```

where all explanatory variables are highly significant. If the optimization does not converge, we have the option to select different start values for σ. Testing this model with different values of `start.sigma` always yields the same results.

Furthermore, Table 3.14 gives parameter estimates for increasing K.

K	int.	sex1	age2	age3	age4	sigma	dev.
1	−8.16	1.44	0.77	0.72	0.47	0.16	200.5
40	−8.16	1.44	0.77	0.72	0.47	0.16	200.5
50	−8.16	1.44	0.77	0.72	0.47	0.16	200.5
70	−8.16	1.44	0.77	0.72	0.47	0.16	200.5

Table 3.14: Parameter estimates and deviances of `glmmML` for the `irlsuicide` data with increasing K and Gaussian quadrature.

This confirmed the assumption that the function `glmmML` uses an adaptive Gauss-Hermite quadrature, since also for small values of K the estimates do not change significantly.

We also tried here different values of K and noticed that there exists also a difference between the deviances for different values of K, but rather for large K, $K > 80$. For these values of K, the values of the deviances change sometimes, although the parameter estimates are completely the same. But as far as Gaussian quadrature is concerned, there is no big difference in the mass points and masses when changing $K = 80$ to $K = 100$. So using far more than 80 mass points not recommended.

Another weird fact is that when we order the data set in different ways, the function `glmmML` calculates different deviances and `sigma`s. For example the irish suicide data is ordered by the factor levels of `ID`, which is a numeric vector with increasing integers. If we take the factor `Region` as clustering variable, which is of the same content as `ID` but

not numeric, the estimate of `sigma` strongly differs, and so does the deviance. To sum up, for the function `glmmML` it is important that the clustering variable is a numeric vector, otherwise the parameter estimate for `sigma` could differ compared to the parameter estimate calculated by the function `allvc`.

Comparison of the four different models is shown in Table 3.15.

	K	int.	sex1	age2	age3	age4	sigma	dev.
m.glm.irish	1	−8.20	1.44	0.77	0.72	0.50	-	290.97
m.allvc.irish.gq	41	−8.16	1.44	0.77	0.72	0.47	0.16	200.51
m.allvc.irish.np	3	−8.15	1.44	0.77	0.72	0.47	0.14	194.44
m.glmmML.irish	2	−8.16	1.44	0.77	0.72	0.47	0.16	200.5

Table 3.15: Parameter estimates and deviances for the `irlsuicide` data for the four different models: `m.glm.irish`, `m.allvc.irish.gq`, `m.allvc.irish.np`, `m.glmmML`.

Actually, there is no difference between the estimates of `glmmML` and `allvc`. The deviance for the non-parametric approach is smaller than for the Gaussian quadrature, but the parameter estimates are the same. Further investigation into this dataset is given in Sofroniou et al. (2006).

3.4 Random Coefficient Models

Since all of the functions mentioned above fit random intercept models, we will have a short overview over random coefficient models in this section, which is based on Aitkin et al. (2009) and Friedl (1998). Now, the intercept is assumed to be fixed and the slope varies across the data set. Usually there is also a random intercept included in the model, but here we consider the intercept as fixed. Under the NPML approach, the model is given by

$$\eta_i = x_i^T \beta + x_{ij} z_i \quad \text{with} \quad z_i \overset{iid}{\sim} G.$$

If we compare this expression to equation (3.16), we notice, that instead of the random intercept term there is its interaction with the jth variable. For the application of the EM-algorithm, we again define a K-level factor and rewrite the linear predictor as

$$\eta_{ik} = x_i^T \beta + z_1 x_{ij} \cdot 0 + \cdots + z_{k-1} x_{ij} \cdot 0 + z_k x_{ij} \cdot 1 + z_{k+1} x_{ij} \cdot 0 + \cdots + z_K x_{ij} \cdot 0$$

which leads to Table 3.16, giving the response design structure.

y	w	β_1	$\ldots$	β_p	z_1	z_2	$\ldots$	z_K
y_1	w_{11}	x_{11}	$\ldots$	x_{1p}	x_{1j}	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{n1}	x_{n1}	$\ldots$	x_{np}	x_{nj}	0	$\vdots$	0
y_1	w_{12}	x_{11}	$\ldots$	x_{1p}	0	x_{1j}	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{n2}	x_{n1}	$\ldots$	x_{np}	0	x_{nj}	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_1	w_{1K}	x_{11}	$\ldots$	x_{1p}	0	0	$\vdots$	x_{1j}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	w_{nK}	x_{n1}	$\ldots$	x_{np}	0	0	$\vdots$	x_{nj}

Table 3.16: Response structure of a NPML model with a random slope coefficient.

3.4.1 R-Functions for Random Coefficient Models

The two functions mentioned before, `alldist` and `allvc`, can also be used to model random coefficient models. In the first subsection we will consider a different example than the previous examples, because the previous included factors as explanatory variables and this allows only to model a factor specific random intercept. The example we consider here is the Oxboys data frame, describing the height of 26 boys from Oxford, England, which was measured on nine occasions over two years. The data set hast 234 rows and 4 columns and is given in the package `nlme`. It contains the vector `Subject`, which is an ordered factor giving a unique identifier for each boy in the experiment, the vector `age`, which is a numeric vector giving the standardized age (dimensionless) and `height` is a numeric vector, giving the height of the boys (cm). Figure 3.4 graphically displays the data.

Simple Generalized Linear Model

A simple generalized linear model, with normal pdf, where the height is modeled through the age of the boys, yields

```
> glm.Ox<-glm(height~age, data=Oxboys)
> summary(glm.Ox)

  Coefficients:
              Estimate Std. Error t value Pr(>|t|)
  (Intercept) 149.3718     0.5286 282.599  < 2e-16 ***
  age           6.5210     0.8170   7.982 6.64e-14 ***
```

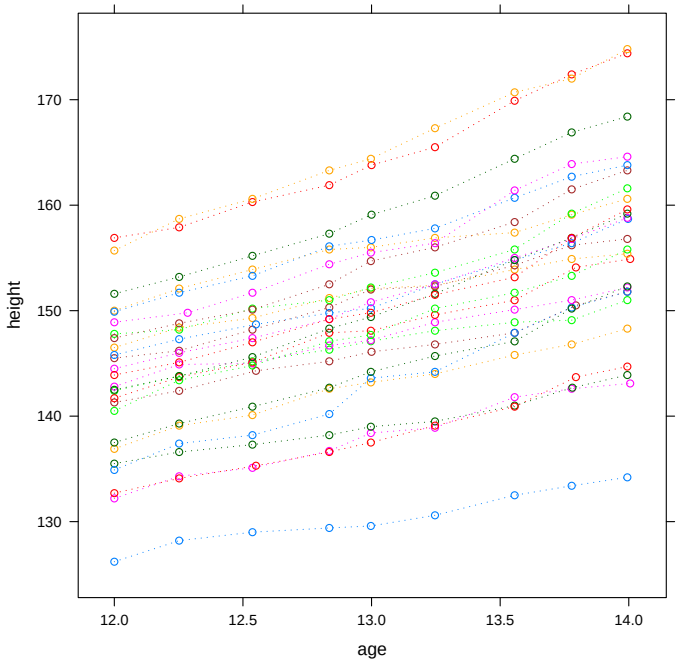

Figure 3.4: Heights of 26 boys over two years.

```
    Null deviance: 19308  on 233  degrees of freedom
Residual deviance: 15148  on 232  degrees of freedom
AIC: 1645.9

Number of Fisher Scoring iterations: 2
```

with a large deviance of 15148 on 232 df. Thus, we will include a random coefficient on `age`.

Random Coefficient Model

Using a random coefficient on the variable `age` and 3 mass points results into

```
> summary(m.allvc.Ox<-allvc(height~age, random=~age|Subject, data=Oxboys,
         k=3))

Coefficients:
            Estimate Std. Error    t value
age         7.919030  0.4065465  19.478782
MASS1     138.588240  0.2827517 490.141113
```

```
MASS2     149.249701  0.1921859 776.590184
MASS3     158.909797  0.2627202 604.863195
MASS1:age  -2.350977  0.5966915  -3.940021
MASS2:age  -1.701525  0.5034540  -3.379703

Mixture proportions:
    MASS1      MASS2      MASS3
0.2313332  0.5007243  0.2679425

Component distribution - MLE of sigma:     3.586
Random effect distribution - standard deviation:           7.161265

-2 log L:           1315     Convergence at iteration  10
```

with a residual deviance of

```
> deviance(m.allvc.Ox)
[1] 3694.751
```

Since for random slopes the NPML approach is the default and nothing else can be chosen, we need not to specify the `random.distribution` here.

Table 3.17 shows the deviances for increasing K for a random coefficient model on `age`.

Random Coefficient Model	
K	`dev.`
1	15148.46
3	3694.75
4	2201.36
5	1648.34
6	1010.08
7	820.31
8	467.47

Table 3.17: `Oxboys` data: Deviances for increasing K

Further increasing of K leads to another deviance reduction, but as the scaled deviance for Gaussian data is given by

$$\frac{1}{\phi}D(y,\hat{\mu}) = \frac{1}{\sigma^2}SSE(\hat{\beta})$$

the deviance is tending to zero, as the number of mass point (the number of parameters) increase. Aitkin et al. (2009) recommended to fit this data with a quadratic term on age. They fully analyzed this data set, therefore no further analysis is given here.

Factor Specific Random Intercept Models

As mentioned before, the models considered in the previous sections included only factors as explanatory variables and this allows only to model a factor specific random intercept and not a random slope coefficient. Thus, we will fit a factor specific random intercept for the model in the next section.

Let us consider the irish suicide data again and additionally assume that the factor `sex` has a factor specific random intercept. Fitting this random coefficient model with three quadrature points results into

```
> summary(m.allvc.irish.rc<-allvc(death~sex+age, random=~sex|ID, data=irlsuicide,
        family=poisson, offset=log(pop), k=3 ))

  Coefficients:
               Estimate Std. Error      t value
  sex1        1.48446575 0.09783953   15.1724535
  age2        0.77416374 0.04540229   17.0512044
  age3        0.72418451 0.04053330   17.8664084
  age4        0.47341257 0.04940393    9.5824886
  MASS1      -8.45991330 0.09067320  -93.3011461
  MASS2      -8.16520193 0.05115727 -159.6098070
  MASS3      -7.98105290 0.09125134  -87.4623083
  MASS1:sex1 -0.11637736 0.13906112   -0.8368792
  MASS2:sex1 -0.03619804 0.11012785   -0.3286911

  Mixture proportions:
       MASS1       MASS2       MASS3
  0.09862076  0.70994388  0.19143536

  Random effect distribution - standard deviation:          0.1225526

  -2 log L:           696.5     Convergence at iteration  5
```

with a deviance of 193.7 on 93 df. Sofroniou et al. (2006) fully analyzed this data set, therefore no further analysis will be considered here.

3.5 Conclusion

In this chapter we considered overdispersion, variance component and random coefficient models. As we have seen, the analysis of variance component models parallels closely that

for overdispersion models. Moreover, two packages have been discussed which are able to fit both, namely `npmlreg` and `glmmML` with the 3 functions `alldist`, `allvc` and `glmmML`. The package `npmlreg`, as its name says, is accounted to model data via a non-parametric maximum likelihood estimation and therefore offers more flexibility and variation in defining and trying different models than the package `glmmML`.
Four types of conditional pdf and pmf are offered in `npmlreg`, namely Gaussian, gamma, Poisson or binomial. Currently, the only valid families to choose in the `glmmML` package are binomial and Poisson. Furthermore, `glmmML` fits only models with a random intercept, whereas `allvc` and `alldist` fit random intercept and random coefficient models. Nevertheless, for the same models they all yield the same parameter estimates, they differ only with respect to the 2nd or 3rd decimal place, which is due to different implementation and therefore different convergence criteria of the EM-algorithm. Of course, there are many other packages and functions for random effect models, like `glmer` in the package `lme4`, `glmmPQL` in `MASS` or `hglm` in package `hglm`, but considering all of them is beyond the scope of this thesis.

Appendix A

R-Packages

A.1 Package `aod`

A.1.1 Function `betabin`

The usage of the function `betabin` is as follows, see also Lesnoff and Lancelot (2010).

```
> betabin(formula, random, data, link=c("logit","cloglog"),
          phi.ini=NULL, warnings=FALSE, na.action=na.omit,
          fixpar=list(), hessian=TRUE, control=list(maxit=2000), ...)
```

with arguments

`formula`	A formula for the fixed effects. The left-hand side of the formula must be of the form `cbind(y,n-y)` where the modeled probability is `y/n`.
`random`	A right-hand formula for the overdispersion parameter(s) ϕ.
`link`	The link function for the mean p: "logit" or "cloglog".
`data`	A data frame containing the response (n and y) and explanatory variable(s).
`phi.ini`	Initial values for the overdispersion parameter(s) ϕ. Default to 0.1.
`warnings`	Logical to control printings of warnings occurring during log-likelihood maximization. Default to FALSE (no printing).
`na.action`	A function name. Indicates which action should be taken in the case of missing value(s)
`hessian`	A logical. When set to FALSE, the hessian and the variances-covariances matrices of the parameters are not computed.
`control`	A list to control the optimization parameters. See `optim`. By default, set the maximum number of iterations to 2000.

`fixpar`	A list with two components (scalars or vectors) of the same size, indicating which parameters are fixed (i.e., not optimized) in the global parameter vector (β, ϕ) and the corresponding fixed values. For example, `fixpar=list(c(4,5),c(0,0))` means that the 4th and 5th parameters of the model are set to 0.
...	Further arguments passed to `optim`.

A.1.2 Function `negbin`

The usage of the function negbin is as follows, see also Lesnoff and Lancelot (2010)

```
> negbin(formula, random, data, phi.ini=NULL, warnings=FALSE,
         na.action=na.omit, fixpar=list(), hessian=TRUE,
         control=list(maxit=2000), ...)
```

with arguments

`formula`	A formula for the fixed effects. The left-hand side of the formula must be the counts y, i.e., positive integers $y \geq 0$. The right-hand side can involve an offset term.
`random`	A right-hand formula for the overdispersion parameter(s) ϕ.
`data`	A data frame containing the response (y) and explanatory variable(s).
`phi.ini`	Initial values for the overdispersion parameter(s) ϕ. Default to 0.1.
`warnings`	Logical to control printings of warnings occurring during log-likelihood maximization. Default to FALSE (no printing).
`na.action`	A function name. Indicates which action should be taken in the case of missing value(s).
`fixpar`	A list with two components (scalars or vectors) of the same size, indicating which parameters are fixed (i.e., not optimized) in the global parameter vector (β, ϕ) and the corresponding fixed values. For example, `fixpar=list(c(4,5),c(0,0))` means that the 4th and 5th parameters of the model are set to 0.
`hessian`	A logical. When set to FALSE, the hessian and the variances-covariances matrices of the parameters are not computed.
`control`	A list to control the optimization parameters. See `optim`. By default, set the maximum number of iterations to 2000.
...	Further arguments passed to `optim`.

A.1.3 Function quasibin

The usage of the function quasibin is as follows, see also Lesnoff and Lancelot (2010):

```
> quasibin(formula, data, link=c("logit", "cloglog"), phi=NULL, tol=0.001)
```

with arguments

formula	Formula for the fixed effects. The left-hand side of the formula must be of the form cbind(y,n-y) where the modeled probability is y/n.
link	Link function for the mean p: "logit" or "cloglog".
data	Data frame containing the response (n and y) and explanatory variable(s).
phi	When phi is NULL (the default), the overdispersion parameter ϕ is estimated from the data. Otherwise, its value is considered as fixed.
tol	A positive scalar (default to 0.001). The algorithm stops at iteration $r+1$ when the condition $\chi^2[r+1] - \chi^2[r] \leq tol$ is met by the χ^2 statistics.

A.1.4 Function quasipois

The usage of the function quasipois is as follows, see also Lesnoff and Lancelot (2010)

```
> quasipois(formula, data, phi=NULL, tol=0.001)
```

with arguments

formula	A formula for the fixed effects. The left hand side of the formula must be the counts y i.e., positive integers ($y \geq 0$). The right hand side can involve an offset term.
data	A data frame containing the response (y) and explanatory variable(s).
phi	When phi is NULL (the default), the overdispersion parameter ϕ is estimated from the data. Otherwise, its value is considered as fixed.
tol	A positive scalar (default to 0.001). The algorithm stops at iteration $r+1$ when the condition $\chi^2[r+1] - \chi^2[r] \leq tol$ is met by the χ^2 statistics.

A.2 Package npmlreg

A.2.1 Function alldist and allvc

The usage of the functions alldist and allvc is as follows, see also Einbeck et al. (2009).

```
> alldist(formula, random = ~1, family = gaussian(), data, k = 4,
          random.distribution = "np", tol = 0.5, offset, weights,
          pluginz, na.action, EMmaxit = 500, EMdev.change = 0.001,
          lambda = 0, damp = TRUE, damp.power = 1, spike.protect =0,
          sdev, shape, plot.opt = 3, verbose = TRUE, ...)

> allvc(formula, random = ~1, family = gaussian(), data, k = 4,
        random.distribution = "np", tol = 0.5, offset, weights,
        pluginz, na.action, EMmaxit = 500, EMdev.change = 0.001,
        lambda = 0, damp = TRUE, damp.power = 1, spike.protect = 0,
        sdev, shape, plot.opt = 3, verbose = TRUE, ...)
```

with arguments

`formula`	A formula defining the response and the fixed effects (e.g. `y~x`).
`random`	A formula defining the random model. In the case of `alldist`, set `random =~1` to model overdispersion, and for instance `random =~x` to introduce a random coefficient x. In the case of `allvc`, set `random =~1|PSU` to model overdispersion on the upper level, where `PSU` is a factor for the primary sampling units, e.g. groups, clusters, classes, or individuals in longitudinal data, and define random coefficients accordingly.
`data`	The data frame (mandatory, even if it is attached to the workspace)
`k`	The number of mass points/integration points (supported are up to 600 mass points)
`random.distribution`	The mixing distribution, Gaussian Quadrature (`gq`) or NPML (`np`) can be set.
`tol`	The tol scalar (usually, $0 \leq$`tol`≤ 1)
`offset`	An optional offset to be included in the model.
`weights`	Optional prior weights for the data.
`pluginz`	Optional numerical vector of length `k` specifying the starting mass points of the EM algorithm.
`na.action`	A function indicating what should happen when `NA`'s occur, with possible arguments `na.omit` and `na.fail`. The default is set by the `na.action` setting in `options()`.
`EMmaxit`	Maximum number of EM iterations.
`EMdev.change`	Stops EM algorithm when deviance change falls below this value.

`lambda`	Only applicable for Gaussian and Gamma mixtures. If set, standard deviations/shape parameters are calculated smoothly across components via a Aitchison-Aitken kernel (`dkern`) with parameter `lambda`. The setting `lambda = 0` is automatically mapped to `lambda = 1/k` and corresponds to the case "maximal smoothing" (i.e. equal component dispersion parameters), while `lambda = 1` means "no smoothing" (unequal disp. param.)
`damp`	Switches EM damping on or off.
`damp.power`	Steers degree of damping applied on dispersion parameter according to formula `1-(1-tol)^(damp.power*iter+1)`, see Einbeck and Hinde (2006).
`spike.protect`	Protects algorithm to converge into likelihood spikes for Gaussian and Gamma mixtures with unequal or smooth component standard deviations, by stopping the EM algorithm if one of the component standard deviations (shape parameters, resp.), divided by the fitted mass points, falls below (exceeds, resp.) a certain threshold, which is `0.000001*spike.potect` (`10^ 6*spike.protect`, resp.) Setting `spike.protect=0` means disabling the spike protection. If set, then `spike.protect=1` is recommended. Note that the displayed disparity may not be correct when convergence is not achieved. This can be checked with `EMconverged`.
`sdev`	optional; specifies standard deviation for normally distributed response. If unspecified, it will be estimated from the data.
`shape`	optional; specifies shape parameter for gamma-distributed response. Setting `shape=1` gives an exponential distribution. If unspecified, it will be estimated from the data.
`plot.opt`	if equal to zero, then no graphical output is given. For `plot.opt=1` the development of the disparity $-2 \log L$ over iteration number is plotted, for `plot.opt=2` the EM trajectories are plotted, and for `plot.opt=3` both plots are shown.
`verbose`	if set to `FALSE`, no printed output is given during function execution. Useful for `tolfind`.
`...`	generic options for the `glm` function. Not all options may be supported under any circumstances.

A.2.2 Function gqz

The usage of the function `gqz` is as follows, see also Einbeck et al. (2009).

```
> gqz(numnodes=20, minweight=0.000001)
```

`numnodes`	theoretical number of quadrature points.
`minweight`	locations with weights that are less than this value will be omitted.

A.3 Package `MASS`

A.3.1 Function `glm.nb`

The usage of the function `glm.nb` is as follows, see also Venables and Ripley (2002).

```
> glm.nb(formula, data, weights, subset, na.action,
         start = NULL etastart, mustart, control  glm.control(...),
         method = "glm.fit",model = TRUE, x = FALSE, y = TRUE,
         contrasts = NULL, ..., init.theta, link=log)
```

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. A typical predictor has the form `response~terms` where `response` is the (numeric) response vector and `terms` is a series of terms which specifies a linear predictor for `response`. For `binomial` and `quasibinomial` families the response can also be specified as a `factor` (when the first level denotes failure and all other success) or as a two-column matrix with the columns giving the numbers of successes and failures. A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in second with any duplicates removed. A specification of the form `first:second` indicates that the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the *cross* of `first` and `second`.
`data`	an optional data frame, list or environment containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `glm` is called.
`weights`	an optional vector of *prior weights* to be used in the fitting process. Should be `NULL` or a numeric vector.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.

na.action	a function which indicates what should happen when the data contains NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The *factory fresh* default is na.omit. Another possible value is NULL. no action. Value na.exclude can be useful.
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	starting values for the vector of means.
control	a list of parameters for controlling the fitting process. See the documentation for glm.control for details.
method	the method to be used in fitting the model. The default method "glm.fit" used iteratively reweighted least squares (IWLS), and "model.frame" which returns the model frame and does no fitting. User-supplied fitting functions can be supplied either as a function or a character string naming a function, with a function which takes the arguments as glm.fit.
model	a logical value indicating whether 'model frame' should be included as a component of the returned value.
x, y	For glm: logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value. For glm.fit: x is a design matrix of dimension n*p, and y is a vector of observations of length n.
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
⋮	
init.theta	Optional initial value for the theta parameter. If omitted a moment estimator after an initial fit using a Poisson GLM is used.
link	The link function. Currently must be one of log, sqrt or identity.

A.4 Package stats

A.4.1 Function dnbinom

The usage of the function dnbinom is as follows, see also
http://127.0.0.1:24648/library/stats/html/NegBinomial.html.

```
> dnbinom(x, size, prob, mu, log=FALSE)
```

with arguments

`x`	Vector of (non-negative integer) quantiles.
`size`	Target for number of successful trials or dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer.
`prob`	Probability of success in each trial. $0 <$`prob`≤ 1.
`mu`	Alternative parametrization via mean. (Use only with `size` and `x`. The variance is $\mu + \mu^2/size$ is this parametrization.)
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

A.5 Package `glmmML`

A.5.1 Function `glmmML`

The usage of the function `glmmML` is as follows, see also Broström (2009).

```
> glmmML(formula, family = binomial, data, cluster, weights, cluster.weigths,
         subset, na.action, offset, prior=c(‘‘gaussian’’, ‘‘logistic’’,
         ‘‘’cauchy’), start.coef = NULL, start.sigma = NULL, fix.sigma = FALSE,
         control=list(epsilon=1e-08, maxit=200, trace = FALSE),
         method = c(‘‘Laplace’’, ‘‘ghq’’), n.points = 8, boot = 0)
```

with arguments

`formula`	A symbolic description of the model to be fit.
`family`	Currently, the only valid values are `binomial` and `poisson`. The binomial family allows for the `logit` and `cloglog` link.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from ‘environment(formula)’, typically the environment from which ‘glmmML’ is called.
`cluster`	Factor indicating which items are correlated.
`weights`	Case weights. Default to one.
`cluster.weights`	Cluster weights. Default to one.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`na.action`	See glm.
`start.coef`	starting values for the parameters in the linear predictor. Defaults to zero.

`start.sigma`	starting value for the mixing standard deviation. Defaults to 0.5
`fix.sigma`	Should sigma be fixed at `start.sigma`?
`offset`	this can be used to specify an a priori known component to be included in the linear predictor during the fitting.
`prior`	Which "prior" distribution (for the random effects)? Possible choices are "gaussian"(default), "logistic", and "cauchy".
`control`	Controls the convergence criteria. See `glm.control` for details.
`method`	There are two choices "Laplace" (default) and "ghq" (Gauss-Hermite).
`n.points`	Number of points in the Gaussian Hermite quadrature. If `n.points ==1`, the Gauss-Hermite is the same as the Laplace approximation. If `method` is set to "Laplace", this parameter is ignored.
`boot`	Do you want a bootstrap estimate of cluster effect? The default is No (`boot = 0`). If you want to say yes, enter a positive integer here. It should be equal to the number of bootstrap samples you want to draw. A recommended absolute minimum value is `boot = 2000`.

A.5.2 Function ghq

The usage of the function `ghq` is as follows, see also Broström (2009).

```
> ghq(n.points = 1, modified = TRUE)
```

`n.points`	Number of points.
`modified`	Multiply by exp(zeros**2)? Default is TRUE.

References

Aitkin, M. (1994). An EM-algorithm for overdispersion in generalized linear models. In J. Hinde, J. Einbeck, and J. Newell (Eds.), *Proceedings of the 9th international workshop on statistical modelling.* Exeter.

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and Computing*, *6*, 251-262.

Aitkin, M., Francis, B., Hinde, J., and Darnell, R. (2009). *Statistical Modelling in R.* New York: Oxford University Press Inc.

Box, G., and Cox, D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, *26*, 211-252.

Breslow, N. (1984). Extra-Poisson variation in log-linear models. *Journal of the Royal Statistical Society, Series C*, *33*, 38-44.

Broström, G. (2003). Generalized linear models with clustered data: The glmmml package [Computer software manual]. (Vignette to R package `glmmML`)

Broström, G. (2009). glmmml: Generalized linear models with clustering [Computer software manual]. Available from `http://cran.r-project.org/package=glmmML` (R package version 0.81-6)

Casella, G., and Berger, R. (2002). *Statistical Inference* (2nd ed.). Pacific Grove, Calif.: Duxbury.

Crowder, M. (1978). Beta-binomial anova for proportions. *Journal of the Royal Statistical Society, Series C*, *27*, 34-37.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1-38.

Einbeck, J., Darnell, R., and Hinde, J. (2009). npmlreg: Nonparametric maximum likelihood estimation for random effect models [Computer software manual]. Available from `http://cran.r-project.org/package=npmlreg` (R package version 0.44)

Einbeck, J., and Hinde, J. (2006). A note on npml estimation for exponential family regression models with unspecified dispersion parameter. *Austrian Journal of Statistics*, *35*, 233-243.

Einbeck, J., and Hinde, J. (2009). Nonparametric maximum likelihood estimation for random effect models in R [Computer software manual].

Faraway, J. (2006). *Extending the Linear Model with R.* Boca Raton: Taylor and Francis Group,LLC.

Fisher, R. (1935). The logic of inductive inference. *Journal of the Royal Statistical Society, Series A*, *98*, 39-82.

Friedl, H. (1991). *Verallgemeinerte logistische Modelle in der Analyse von Zervix-Karzinomen.* Unpublished doctoral dissertation, Technische Universität Graz.

Friedl, H. (1998). Nichtparametrische Maximum Likelihood Schätzung in Generalisierten Linearen Mischmodellen. *Österreichische Zeitschrift für Statistik*, *26*, 7-30.

Griffiths, D. (1973). Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of a disease. *Biometrics*, *29*, 637-648.

Hinde, J. (1982). Compound Poisson regression models. *GLIM 1982 International Conference on Generalized Linear Models*, 109-121.

Hinde, J., and Demetrio, C. (1998). Overdispersion: Models and estimation. *Computational Statistics & Data Analysis*, *27*, 151-170.

Lawless, J. (1987). Negative binomial and mixed poisson regression. *The Canadian Journal of Statistics*, *15*, 209-225.

Lee, Y., Nelder, J., and Pawitan, Y. (2006). *Generalized Linear Models with Random Effects*. Boca Raton: Taylor and Francis Group,LLC.

Lesnoff, M., and Lancelot, R. (2010). aod: Analysis of overdispersed data [Computer software manual]. Available from `http://cran.r-project.org/package=aod` (R package version 1.2)

McCullagh, C., Searle, S., and Neuhaus, J. (2008). *Generalized, Linear, and Mixed Models*. New Jersey: John Wiley and Sons.

McCullagh, P., and Nelder, J. (1989). *Generalized Linear Models*. New York: Chapmann and Hall.

Sofroniou, N., Einbeck, J., and Hinde, J. (2006). Analyzing Irish suicide rates with mixture models. In *Proceedings of the 21st international workshop on statistical modelling* (p. 474-481). Galway, Ireland.

Venables, W., and Ripley, B. (2002). *Modern Applied Statistics with S* (4th ed.). New York: Springer. Available from `http://www.stats.ox.ac.uk/pub/MASS4` (ISBN 0-387-95457-0)

Wedderburn, R. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, *61*, 439-447.

Williams, D. (1982). Extra-binomial variation in logistic linear models. *Journal of the Royal Statistical Society, Series C*, *31*, 144-148.