Week 3 April 22

Lecture slides, week 3 (pdf)
Audio companion, week 3

Main topic: Generalized Linear Mixed Models: counts and proportions.
nice overview: Generalized Linear Mixed Models from *Encyclopedia of Statistics in Behavioral Science*.

1. Dichotomous outcomes, `glmer` analysis examples.
   a. Respiratory clinical trial from HSAUR. lmList does logistic, introducing glmer lmList, glmer for respiration data (placebo group)
   b. Contraception (Bangladesh) use from Bates review Rogosa R-session glmer model slide
   c. Test scores (pass/fail outcome) from Ch 8, Multilevel Modeling Using R. Rogosa R-session
2. Count outcome, GLMM poisson models.
   a. Count data: Contagious bovine pleuropneumonia, data(cbpp) in lme4.
   Rogosa R-session herd plots
   b. Factorial design, Count outcome. From HIE Sydney. EucFACE ground cover data Rogosa glmer session.
   c. Another count data example from mlmRev package, data(Mmmec): Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.
   Rogosa glmer session more Rogosa Mmmec session

`glmer.nb` Fits a generalized linear mixed-effects model (GLMM) for the negative binomial family, building on glmer, and initializing via theta.ml from MASS.

3. hglm -- a different package for fitting hierarchical generalized linear models.
R Journal December 2010. manual vignette

Week 4 April 29

Lecture slides, week 4 (pdf)
Audio companion, week 4

1. Three-level (and above) lmer examples.
Measured outcomes
a. Achieve data from *Multilevel Modeling Using R* book. Rogosa R-session
b. example from mlmRev package, data(Chem97): Scores on A-level Chemistry in 1997. Rogosa R-session
Count outcome.
c. data(Mmmec): Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure. Rogosa 3-level session

2. Missing Data and Imputation Methods for Multilevel Data Analysis
Vignette: Analyzing Imputed Data with Multilevel Models and merTools
Rogosa R-session for vignette
Missing data wide-form imputation: `mice` multiple regression example, nhanes data in package `mice` R-session using mice package
Missing data background. Multiple Imputation. Nhanes data example (mice primer) in van Buuren S and Groothuis-Oudshoorn K (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67.
See also Flexible Imputation of Missing Data. Stef van Buuren Chapman and Hall/CRC 2012. Chapter 9, Longitudinal Data Sec 3.8 Multilevel data. He is the originator of `mice` book extras
R resources. Multivariate Analysis Task View, *Missing data* section, esp packages `mice`
New package: `hmi`: hierarchical multiple imputation, vignette

Generalized Linear Mixed Models

Introduction

Generalized linear models (GLMs) represent a class of fixed effects regression models for several types of dependent variables (i.e., continuous, dichotomous, counts). McCullagh and Nelder [32] describe these in great detail and indicate that the term 'generalized linear model' is due to Nelder and Wedderburn [35] who described how a collection of seemingly disparate statistical techniques could be unified. Common Generalized linear models (GLMs) include linear regression, **logistic regression**, and Poisson regression.

There are three specifications in a GLM. First, the linear predictor, denoted as η_i, of a GLM is of the form

$$\eta_i = \boldsymbol{x}_i'\boldsymbol{\beta}, \tag{1}$$

where $\boldsymbol{x}_i$ is the vector of regressors for unit i with fixed effects $\boldsymbol{\beta}$. Then, a link function $g(\cdot)$ is specified which converts the expected value μ_i of the outcome variable Y_i (i.e., $\mu_i = E[Y_i]$) to the linear predictor η_i

$$g(\mu_i) = \eta_i. \tag{2}$$

Finally, a specification for the form of the variance in terms of the mean μ_i is made. The latter two specifications usually depend on the distribution of the outcome Y_i, which is assumed to fall within the exponential family of distributions.

Fixed effects models, which assume that all observations are independent of each other, are not appropriate for analysis of several types of correlated data structures, in particular, for clustered and/or longitudinal data (*see* **Clustered Data**). In clustered designs, subjects are observed nested within larger units, for example, schools, hospitals, neighborhoods, workplaces, and so on. In longitudinal designs, repeated observations are nested within subjects (*see* **Longitudinal Data Analysis** and **Repeated Measures Analysis of Variance**). These are often referred to as multilevel [16] or hierarchical [41] data (*see* **Linear Multilevel Models**), in which the level-1 observations (subjects or repeated observations) are nested within the higher level-2 observations (clusters or subjects). Higher levels are also possible, for example, a three-level design could have repeated observations (level-1) nested within subjects (level-2) who are nested within clusters (level-3).

For analysis of such multilevel data, random cluster and/or subject effects can be added into the regression model to account for the correlation of the data. The resulting model is a mixed model including the usual fixed effects for the regressors plus the random effects. Mixed models for continuous normal outcomes have been extensively developed since the seminal paper by Laird and Ware [28]. For nonnormal data, there have also been many developments, some of which are described below. Many of these developments fall under the rubric of generalized linear mixed models (GLMMs), which extend GLMs by the inclusion of random effects in the predictor. Agresti et al. [1] describe a variety of social science applications of GLMMs; [12], [33], and [11] are recent texts with a wealth of statistical material on GLMMs.

Let i denote the level-2 units (e.g., subjects) and let j denote the level-1 units (e.g., nested observations). The focus will be on longitudinal designs here, but the methods apply to clustered designs as well. Assume there are $i = 1, \ldots, N$ subjects (level-2 units) and $j = 1, \ldots, n_i$ repeated observations (level-1 units) nested within each subject. A random-intercept model, which is the simplest mixed model, augments the linear predictor with a single random effect for subject i,

$$\eta_{ij} = \boldsymbol{x}_{ij}'\boldsymbol{\beta} + \nu_i, \tag{3}$$

where ν_i is the random effect (one for each subject). These random effects represent the influence of subject i on his/her repeated observations that is not captured by the observed covariates. These are treated as random effects because the sampled subjects are thought to represent a population of subjects, and they are usually assumed to be distributed as $\mathcal{N}(0, \sigma_\nu^2)$. The parameter σ_ν^2 indicates the variance in the population distribution, and therefore the degree of heterogeneity of subjects.

Including the random effects, the expected value of the outcome variable, which is related to the linear

Reproduced from the *Encyclopedia of Statistics in Behavioral Science*. © John Wiley & Sons, Ltd.
ISBN: 0-470-86080-4.

Some choices of univariate conditional distributions

generalized linear mixed models (GLMMs)

- Typical choices of univariate conditional distributions are:
  - The *Bernoulli* distribution for binary (0/1) data, which has probability mass function

$$p(y|\mu) = \mu^y (1-\mu)^{1-y}, \quad 0 < \mu < 1, \quad y = 0, 1$$

  - Several independent binary responses can be represented as a *binomial* response, but only if all the Bernoulli distributions have the same mean.
  - The *Poisson* distribution for count $(0, 1, \dots)$ data, which has probability mass function

$$p(y|\mu) = e^{-\mu}\frac{\mu^y}{y!}, \quad 0 < \mu, \quad y = 0, 1, 2, \dots$$

- All of these distributions are completely specified by the conditional mean. This is different from the conditional normal (or Gaussian) distribution, which also requires the common scale parameter, σ.

The link function, g

- When the univariate conditional distributions have constraints on μ, such as $0 < \mu < 1$ (Bernoulli) or $0 < \mu$ (Poisson), we cannot define the conditional mean, $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{U}}$, to be equal to the linear predictor, $\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{X}\boldsymbol{\Lambda}_\theta\boldsymbol{u}$, which is unbounded.
- We choose an invertible, univariate *link function*, g, such that $\eta = g(\mu)$ is unconstrained. The vector-valued link function, $\boldsymbol{g}$, is defined by applying g component-wise.

$$\boldsymbol{\eta} = \boldsymbol{g}(\boldsymbol{\mu}) \quad \text{where} \quad \eta_i = g(\mu_i), \quad i = 1, \ldots, n$$

- We require that g be invertible so that $\mu = g^{-1}(\eta)$ is defined for $-\infty < \eta < \infty$ and is in the appropriate range ($0 < \mu < 1$ for the Bernoulli or $0 < \mu$ for the Poisson). The vector-valued inverse link, $\boldsymbol{g}^{-1}$, is defined component-wise.

"Canonical" link functions

- There are many choices of invertible scalar link functions, g, that we could use for a given set of constraints.
- For the Bernoulli and Poisson distributions, however, one link function arises naturally from the definition of the probability mass function. (The same is true for a few other, related but less frequently used, distributions, such as the gamma distribution.)
- To derive the canonical link, we consider the logarithm of the probability mass function (or, for continuous distributions, the probability density function).
- For distributions in this "exponential" family, the logarithm of the probability mass or density can be written as a sum of terms, some of which depend on the response, y, only and some of which depend on the mean, μ, only. However, only one term depends on **both** y and μ, and this term has the form $y \cdot g(\mu)$, where g is the canonical link.

The canonical link for the Bernoulli distribution

- The logarithm of the probability mass function is

$$\log(p(y|\mu)) = \log(1-\mu) + y\log\left(\frac{\mu}{1-\mu}\right),\ 0<\mu<1,\ y=0,1.$$

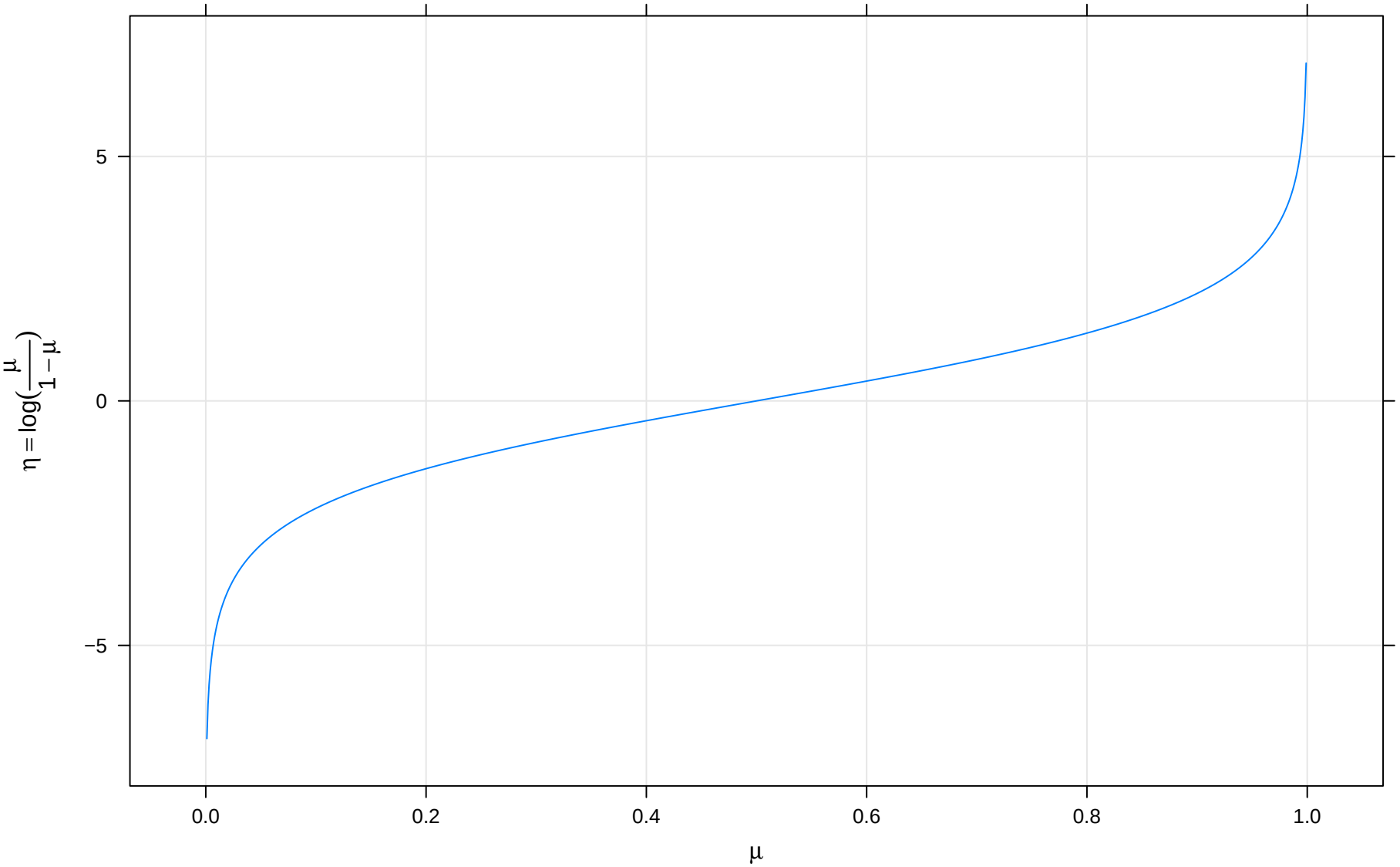
- Thus, the canonical link function is the *logit* link

$$\eta = g(\mu) = \log\left(\frac{\mu}{1-\mu}\right).$$

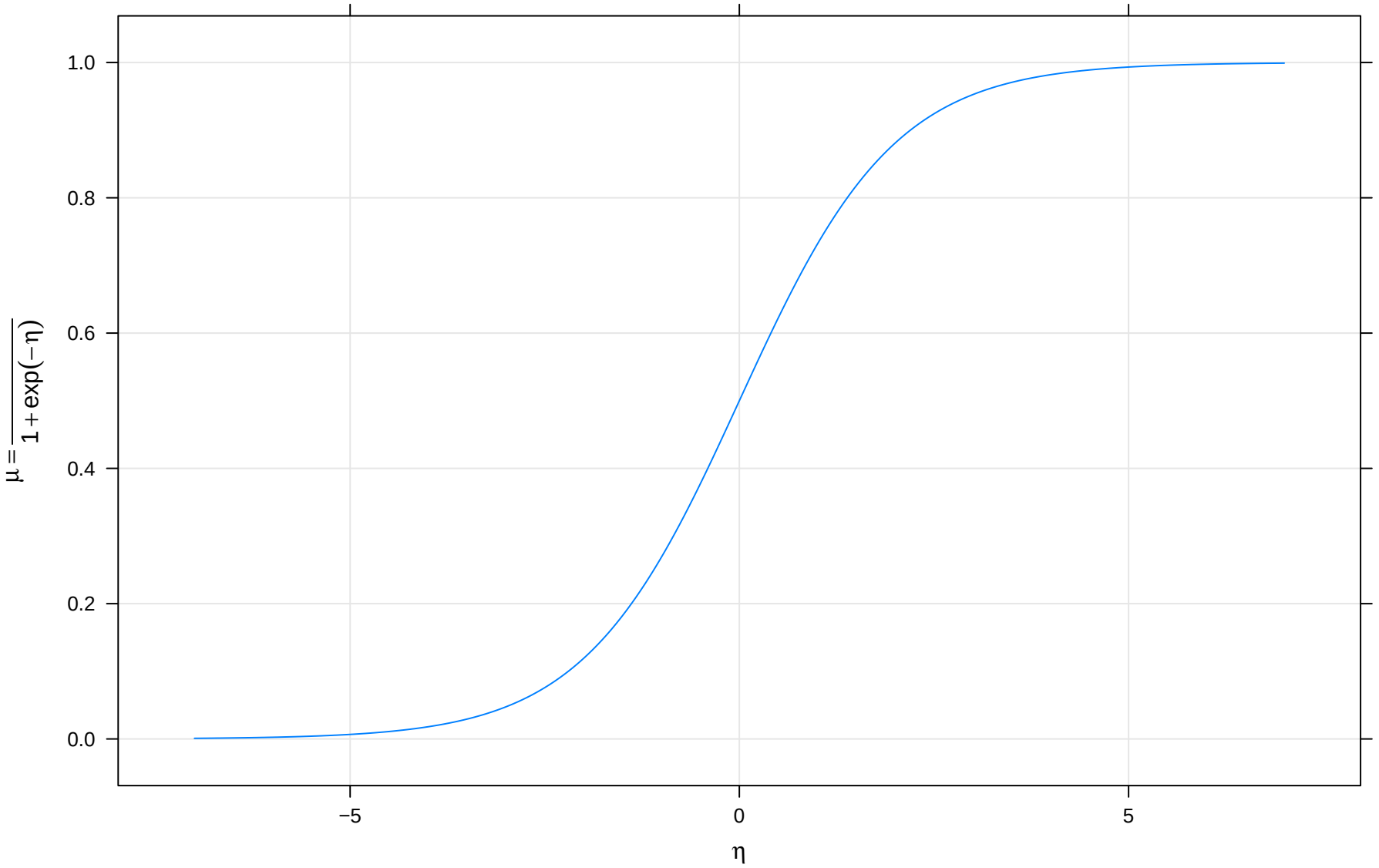
- Because $\mu = P[\mathcal{Y}=1]$, the quantity $\mu/(1-\mu)$ is the odds ratio (in the range $(0,\infty)$) and g is the logarithm of the odds ratio, sometimes called "log odds".
- The inverse link is

$$\mu = g^{-1}(\eta) = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{1+e^{-\eta}}$$

Plot of canonical link for the Bernoulli distribution



Plot of inverse canonical link for the Bernoulli distribution



The canonical link for the Poisson distribution

- The logarithm of the probability mass is

$$\log(p(y|\mu)) = \log(y!) - \mu + y\log(\mu)$$

- Thus, the canonical link function for the Poisson is the *log* link

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

- The inverse link is

$$\mu = g^{-1}(\eta) = e^{\eta}$$

The canonical link related to the variance

- For the canonical link function, the derivative of its inverse is the variance of the response.
- For the Bernoulli, the canonical link is the logit and the inverse link is $\mu = g^{-1}(\eta) = 1/(1 + e^{-\eta})$. Then

$$\frac{d\mu}{d\eta} = \frac{e^{-\eta}}{(1 + e^{-\eta})^2} = \frac{1}{1 + e^{-\eta}} \frac{e^{-\eta}}{1 + e^{-\eta}} = \mu(1 - \mu) = \mathrm{Var}(\mathcal{Y})$$

- For the Poisson, the canonical link is the log and the inverse link is $\mu = g^{-1}(\eta) = e^{\eta}$. Then

$$\frac{d\mu}{d\eta} = e^{\eta} = \mu = \mathrm{Var}(\mathcal{Y})$$

Dichotomous Outcome - Logistic Regression

$Y=1$ success, $E(Y)=\pi$ prob of success. For predictor X model π by

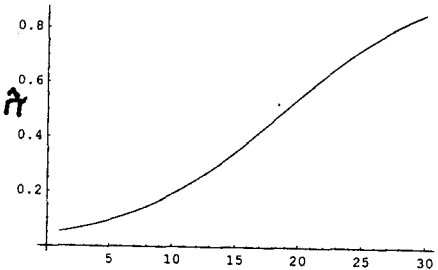
$$E(Y|X) = \frac{\exp(\beta_0+\beta_1 X)}{1+\exp(\beta_0+\beta_1 X)}$$

linear in logit

$$\ln\left(\frac{\pi}{1-\pi}\right) = \beta_0+\beta_1 X$$

ex logistic fit

```
(* programmer example *)
logitfit[x_, b0_, b1_] := Exp[b0 + b1*x] / (1 + Exp[b0 + b1*x])
```



moments

Logistic Regression Two Predictors, one measured, one dichotomous

```
Survival in the Donner Party --Logistic Regression
File donner.dat contains the ages and sexes of the adult (over 15 years) survivors
and non-survivors of the party. These data were used by an anthropologist to study
the theory that females are better able to withstand harsh conditions than are males
(Data from D. K. Grayson, 1990, "Donner Party Deaths: A Demographic Assessment,"
Journal of Anthropological Research 46: (1990): 223–42.)
> donner = read.table(file="D:\\stat141\\donner.dat", header = T) #cols age male survival
> table(survival, male)|> prop.table(table(survival, male),2)
        male            |        male                  > # females 2/3 survive, males 1/3 surv
survival  0  1          |survival         0         1  > #relrisk being male = 2
       0  5 20          |       0 0.3333333 0.6666667
       1 10 10          |       1 0.6666667 0.3333333
> smor = oddsratio(donsm, log = F)  > summary(smor)  Odds Ratio [1,]       0.25
> confint(smor)                     > confint(smor, level = .99)
            lwr       upr                      lwr      upr
[1,] 0.07007981 0.8918403          [1,] 0.04699297 1.329986
> lgstsurv = glm(survival ~ age + male, family = binomial)
> summary(lgstsurv)
Call: glm(formula = survival ~ age + male, family = binomial)
Coefficients:                Estimate Std. Error z value   Pr(>|z|)
            (Intercept)  3.23041    1.38686   2.329   0.0198 *
            age         -0.07820    0.03728  -2.097   0.0359 *
            male        -1.59729    0.75547  -2.114   0.0345 *
> cbind(age, male,predict.glm(lgstsurv), fitted(lgstsurv))
   age male   log-odds   prob surv
1   23    1 -0.16557336 0.45870097
2   40    0  0.10225091 0.52554048
3   40    1 -1.49504259 0.18316607
4   30    1 -0.71300187 0.32893588
5   28    1 -0.55659372 0.36433597
6   40    1 -1.49504259 0.18316607
7   45    0 -0.28876945 0.42830515
8   62    1 -3.21553219 0.03858538
9   65    1 -3.45014441 0.03076455
10  45    0 -0.28876945 0.42830515
...
45
> #predict gives log-odds, fitted gives prob success
      type = "response",
   > plot(age, fitted(lgstsurv),pch = male + 15)
```

donsm = table(survival, male)

refer to logit table 11/17

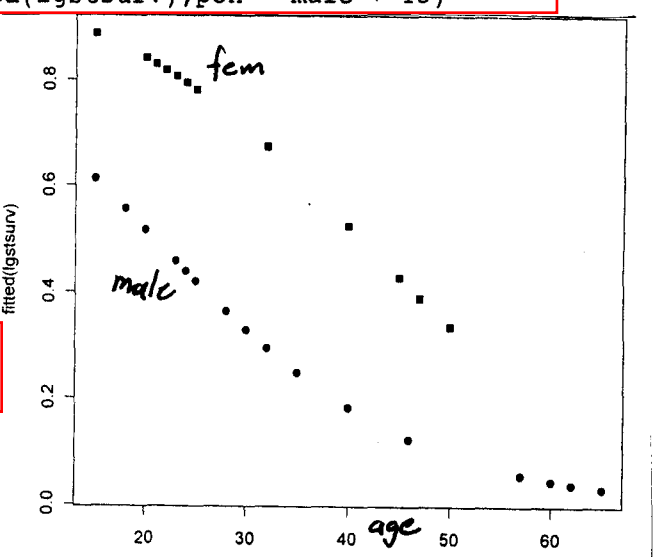
log-odds = 3.23 − .078*A − 1.597*M

$e^{-.078}$ = .925 multiplicative change in odds for 1yr increase in age.

fitted prob survival

odds = P/(1−P)

p = odds/(odds+1)




```
> exp(-1.5973)
[1] 0.2024424 #odds 1/5 as large for male, constant age
> predict.glm(lgstsurv, type="response",
newdata = data.frame(male=0, age=20))    [1] 0.8410862
> predict.glm(lgstsurv, type="response",
newdata = data.frame(male=0, age=50))    [1] 0.3363082
```

$e^{\hat{\beta}}$ change from male=0 to male=1 in odds

odds 5 F50 F20

.51 5.3

`respiratory` *Respiratory Illness Data*

Description

The respiratory status of patients recruited for a randomised clinical multicenter trial.

Usage

```
data("respiratory")
```

Format

A data frame with 555 observations on the following 7 variables.

`centre` the study center, a factor with levels `1` and `2`.

`treatment` the treatment arm, a factor with levels `placebo` and `treatment`.

`gender` a factor with levels `female` and `male`.

`age` the age of the patient.

`status` the respiratory status (response variable), a factor with levels `poor` and `good`.

`month` the month, each patient was examined at months `0`, `1`, `2`, `3` and `4`.

`subject` the patient ID, a factor with levels `1` to `111`.

Details

In each of two centres, eligible patients were randomly assigned to active treatment or placebo. During the treatment, the respiratory status (categorised `poor` or `good`) was determined at each of four, monthly visits. The trial recruited 111 participants (54 in the active group, 57 in the placebo group) and there were no missing data for either the responses or the covariates. The question of interest is to assess whether the treatment is effective and to estimate its effect.

Note that the data are in long form, i.e, repeated measurments are stored as additional rows in the data frame.

Source

C. S. Davis (1991), Semi-parametric and non-parametric methods for the analysis of repeated measurements with applications to clinical trials. *Statistics in Medicine*, **10**, 1959–1980.

Examples

```
data("respiratory", package = "HSAUR2")
mosaicplot(xtabs( ~ treatment + month + status, data = respiratory))
```

CHAPTER 13

Analysing Longitudinal Data II – Generalised Estimation Equations and Linear Mixed Effect Models: Treating Respiratory Illness and Epileptic Seizures

13.1 Introduction

The data in Table 13.1 were collected in a clinical trial comparing two treatments for a respiratory illness (Davis, 1991).

Table 13.1: `respiratory` data. Randomised clinical trial data from patients suffering from respiratory illness. Only the data of the first seven patients are shown here.

centre	treatment	gender	age	status	month	subject
1	placebo	female	46	poor	0	1
1	placebo	female	46	poor	1	1
1	placebo	female	46	poor	2	1
1	placebo	female	46	poor	3	1
1	placebo	female	46	poor	4	1
1	placebo	female	28	poor	0	2
1	placebo	female	28	poor	1	2
1	placebo	female	28	poor	2	2
1	placebo	female	28	poor	3	2
1	placebo	female	28	poor	4	2
1	treatment	female	23	good	0	3
1	treatment	female	23	good	1	3
1	treatment	female	23	good	2	3
1	treatment	female	23	good	3	3
1	treatment	female	23	good	4	3
1	placebo	female	44	good	0	4
1	placebo	female	44	good	1	4
1	placebo	female	44	good	2	4
1	placebo	female	44	good	3	4
1	placebo	female	44	poor	4	4
1	placebo	male	13	good	0	5

© 2010 by Taylor and Francis Group, LLC

Table 13.1: `respiratory` data (continued).

centre	treatment	gender	age	status	month	subject
1	placebo	male	13	good	1	5
1	placebo	male	13	good	2	5
1	placebo	male	13	good	3	5
1	placebo	male	13	good	4	5
1	treatment	female	34	poor	0	6
1	treatment	female	34	poor	1	6
1	treatment	female	34	poor	2	6
1	treatment	female	34	poor	3	6
1	treatment	female	34	poor	4	6
1	placebo	female	43	poor	0	7
1	placebo	female	43	good	1	7
1	placebo	female	43	poor	2	7
1	placebo	female	43	good	3	7
1	placebo	female	43	good	4	7
⋮	⋮	⋮	⋮	⋮	⋮	⋮

In each of two centres, eligible patients were randomly assigned to active treatment or placebo. During the treatment, the respiratory status (categorised `poor` or `good`) was determined at each of four, monthly visits. The trial recruited 111 participants (54 in the active group, 57 in the placebo group) and there were no missing data for either the responses or the covariates. The question of interest is to assess whether the treatment is effective and to estimate its effect.

variety of response functions for complete and incomplete repeated measures categorical data.[10-13]

The above normal theory and categorical methods for repeated measures, however, do not always apply. First, the parametric assumptions underlying the classical analysis methods are frequently not tenable. In some studies, the response is continuous, but the distribution of the outcome variable is extremely non-normal. In addition, in situations in which the response is dichotomous or an ordered categorical variable, the general-purpose categorical methods often have limited usefulness. The WLS methodology allows for categorical covariates only, thus one cannot use it with continuous independent variables. In addition, it requires sufficient sample size for the marginal response functions at each time point within each category of the multi-way cross-classification of the covariates to have an approximately multivariate normal distribution. In practice, this imposes limitations on the total number of measurement times, the total number of covariates and the number of distinct levels of each covariate.

Several extensions of generalized linear models[14] to the analysis of repeated measures data have appeared.[15-19] These semi-parametric approaches are useful in longitudinal data analyses with univariate outcomes for which the quasi-likelihood formulation is sensible, for example, normal, Poisson, binomial and gamma response variables. The methods allow for missing observations and continuous (possibly time-dependent) covariates. Although the semi-parametric approaches are quite flexible, they still require assumptions concerning the distribution of the response variable. Thus, they may not apply to studies with a continuous, but extremely non-normal, response. Such situations may indicate the use of non-parametric methods. In addition to the substantial literature regarding distribution-free methods for complete multivariate observations,[20-23] non-parametric methods for incomplete repeated measurements are also available.[24-28] Two of these methods[25,26] have general applicability and, although developed for the special case of comparing two treatment groups, they do offer the advantage of allowing for differential patterns of missing observations in the two groups.

The purpose of this paper is to review and compare recent semi-parametric and non-parametric methods for analysis of repeated measurements. Section 2 describes three examples from clinical trials with repeated measures. The respective response variables are binary, ordered categorical and continuous (but extremely non-normal). Section 3 reviews and compares recent semi-parametric and non-parametric statistical methodology for analysis of repeated measurements. Section 4 considers application of the various methods to the three data sets. Section 5 concludes with recommendations, issues related to software availability and areas for further research.

2. EXAMPLES

2.1. Binary response

Appendix I displays the raw data from a clinical trial comparing two treatments for a respiratory illness.[29] In each of two centres, eligible patients were randomly assigned to active treatment or placebo. During treatment, respiratory status (categorized here as 0 = poor, 1 = good) was determined at four visits. Potential covariates were centre, sex and baseline respiratory status (all dichotomous), as well as age (in years) at the time of study entry. There were 111 patients (54 active, 57 placebo) with no missing data for responses or covariates.

2.2. Ordered categorical response

In a comparison of the effects of varying dosages of an anaesthetic on post-surgical recovery, 60 young children undergoing outpatient surgery were randomized to one of four dosages (15, 20,

Chap 5 Laird-Ware ALA text

epilepsy or respiration profiles

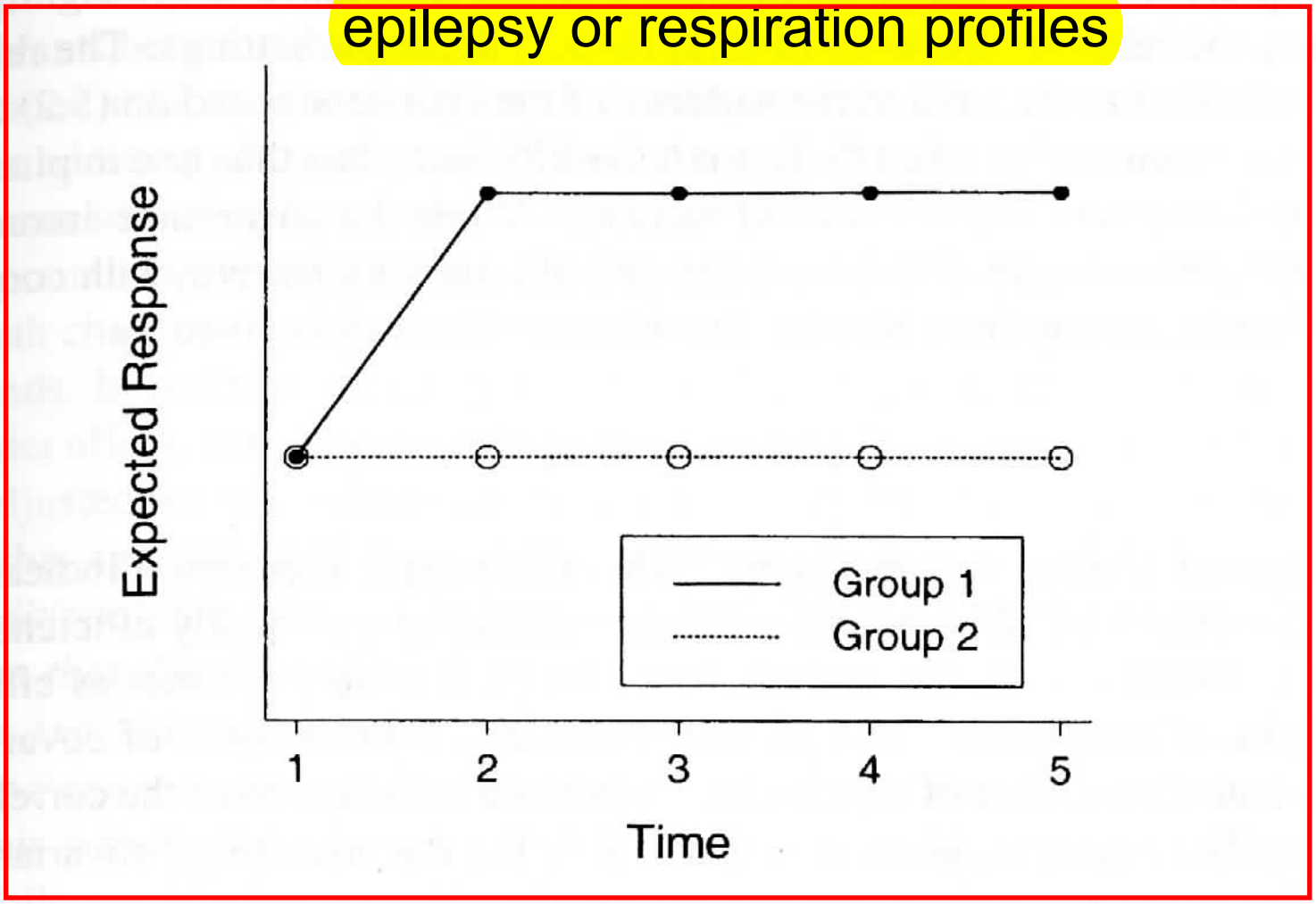


Fig. 5.4 Graphical representation of changes in the mean response from baseline (in Group 1) that persist throughout the duration of follow up.

```
   STAT 222 WEEK 3
Logistic regression for dichotomous outcomes:
  lmList and glmer, respiration data

R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

> install.packages("HSAUR2")
Installing package into 'C:/Users/rag/Documents/R/win-library/3.2'
package 'HSAUR2' successfully unpacked and MD5 sums checked
> library(HSAUR2)
Loading required package: tools
> data(respiratory)
> head(respiratory)
    centre treatment gender age status month subject
1        1   placebo female  46   poor     0       1
112      1   placebo female  46   poor     1       1
223      1   placebo female  46   poor     2       1
334      1   placebo female  46   poor     3       1
445      1   placebo female  46   poor     4       1
2        1   placebo female  28   poor     0       2
>   #Data manip from HSAUR #The baseline status, i.e., the status for month == 0, needs to
>  #  enter the models as an explanatory variable (HSAUR)
>   #rearrange the data.frame respiratory in order to create a new variable baseline
>   resp <- subset(respiratory, month > "0")
>   resp$baseline <- rep(subset(respiratory, month == "0")$status, rep(4, 111))
>   resp$nstat <- as.numeric(resp$status == "good")
>   #new variable nstat is simply a dummy coding for a poor respiratory status
>   resp$month <- resp$month[, drop = TRUE]
>  head(resp)
    centre treatment gender age status month subject baseline nstat
112      1   placebo female  46   poor     1       1     poor     0
223      1   placebo female  46   poor     2       1     poor     0
334      1   placebo female  46   poor     3       1     poor     0
445      1   placebo female  46   poor     4       1     poor     0
113      1   placebo female  28   poor     1       2     poor     0
224      1   placebo female  28   poor     2       2     poor     0
> str(resp)
'data.frame':   444 obs. of  9 variables:
 $ centre   : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ treatment: Factor w/ 2 levels "placebo","treatment": 1 1 1 1 1 1 1 1 2 2 ...
 $ gender   : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 1 1 1 1 ...
 $ age      : num  46 46 46 46 28 28 28 28 23 23 ...
 $ status   : Factor w/ 2 levels "poor","good": 1 1 1 1 1 1 1 1 2 2 ...
 $ month    : Ord.factor w/ 4 levels "1"<"2"<"3"<"4": 1 2 3 4 1 2 3 4 1 2 ...
 $ subject  : Factor w/ 111 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ baseline : Factor w/ 2 levels "poor","good": 1 1 1 1 1 1 1 1 2 2 ...
 $ nstat    : num  0 0 0 0 0 0 0 0 1 1 ...
>  resp$monthN = as.numeric(resp$month) # we could use this for trends but we don't
>  library(lme4)
Loading required package: Matrix
>  resp$statusN = as.numeric(resp$status) - 1 # we need to do this to get lmList to work cor
#### For week 3 example use just the Placebo group data (see week 5)

> respP = resp[resp$treatment == "placebo",]
> dim(resp)
[1] 444  11
> dim(respP)  #57 cases
[1] 228  11
```

```
> table(respP$treatment)
  placebo treatment
      228         0

> respList_P = lmList(statusN ~ 1 | subject, family = binomial, data = respP) # fit flat to
> coef(respList_P)[1:8,]  #in log-odds metric
[1] -23.566069 -23.566069   1.098612  23.566069   1.098612   0.000000  -1.098612 -23.566069
> exp(coef(respList_P))[1:8,]  #in odds metric
[1] 5.826215e-11 5.826215e-11 3.000000e+00 1.716381e+10 3.000000e+00 1.000000e+00 3.333333e-

> respP[1:32,]
    centre treatment gender age status month subject baseline nstat monthN statusN
112      1   placebo female  46   poor     1       1     poor     0      1       0
223      1   placebo female  46   poor     2       1     poor     0      2       0
334      1   placebo female  46   poor     3       1     poor     0      3       0
445      1   placebo female  46   poor     4       1     poor     0      4       0
113      1   placebo female  28   poor     1       2     poor     0      1       0
224      1   placebo female  28   poor     2       2     poor     0      2       0
335      1   placebo female  28   poor     3       2     poor     0      3       0
446      1   placebo female  28   poor     4       2     poor     0      4       0
115      1   placebo female  44   good     1       4     good     1      1       1
226      1   placebo female  44   good     2       4     good     1      2       1
337      1   placebo female  44   good     3       4     good     1      3       1
448      1   placebo female  44   poor     4       4     good     0      4       0
116      1   placebo   male  13   good     1       5     good     1      1       1
227      1   placebo   male  13   good     2       5     good     1      2       1
338      1   placebo   male  13   good     3       5     good     1      3       1
449      1   placebo   male  13   good     4       5     good     1      4       1
118      1   placebo female  43   good     1       7     poor     1      1       1
229      1   placebo female  43   poor     2       7     poor     0      2       0
340      1   placebo female  43   good     3       7     poor     1      3       1
451      1   placebo female  43   good     4       7     poor     1      4       1
121      1   placebo female  37   poor     1      10     good     0      1       0
232      1   placebo female  37   good     2      10     good     1      2       1
343      1   placebo female  37   good     3      10     good     1      3       1
454      1   placebo female  37   poor     4      10     good     0      4       0
124      1   placebo female  23   good     1      13     good     1      1       1
235      1   placebo female  23   poor     2      13     good     0      2       0
346      1   placebo female  23   poor     3      13     good     0      3       0
457      1   placebo female  23   poor     4      13     good     0      4       0
125      1   placebo female  30   poor     1      14     poor     0      1       0
236      1   placebo female  30   poor     2      14     poor     0      2       0
347      1   placebo female  30   poor     3      14     poor     0      3       0
458      1   placebo female  30   poor     4      14     poor     0      4       0
> # odds from lmList match the status indicators

> # fit flat line to each subj dichotomous, to get ave for each subj, then ave for group
> resp_lmera = glmer(status ~ 1 + (1| subject), family = binomial, data = respP)
> summary(resp_lmera)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod'
 Family: binomial  ( logit )
Formula: status ~ 1 + (1 | subject)
   Data: respP
     AIC      BIC   logLik deviance df.resid
   260.4    267.2   -128.2    256.4      226
Scaled residuals:
   Min     1Q Median     3Q    Max
-1.522 -0.293 -0.293  0.340  1.640

Random effects:
```

```
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 6.313    2.512
Number of obs: 228, groups:  subject, 57

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4588     0.3876  -1.184    0.237

> exp(fixef(resp_lmera))
(Intercept)
  0.6320565

> exp(confint(resp_lmera))
Computing profile confidence intervals ...
                2.5 %    97.5 %
.sig01      5.4719582 42.997216
(Intercept) 0.2533728  1.399329
> exp(confint(resp_lmera, method = "boot", nsim = 1000, boot.type = "perc"))
Computing bootstrap confidence intervals ...
               2.5 %    97.5 %
.sig01      4.526701 28.908599
(Intercept) 0.255933  1.410189
There were 45 warnings (use warnings() to see them)

> ############ maybe baseline status is a predictor for status of Placebo group?
> resp_lmerb = glmer(status ~ baseline + (1| subject), family = binomial, data = respP)
> summary(resp_lmerb)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod'
 Family: binomial  ( logit )
Formula: status ~ baseline + (1 | subject)
   Data: respP
     AIC      BIC   logLik deviance df.resid
   241.8    252.1   -117.9    235.8      225
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.7657 -0.2315 -0.2315  0.4887  1.9338

Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 4.472    2.115
Number of obs: 228, groups:  subject, 57

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.0167     0.5829  -3.460 0.000540 ***  #this changes to level when baseline
baselinegood   3.2818     0.8561   3.833 0.000126 ***                                      zero
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Correlation of Fixed Effects:
            (Intr)
baselinegod -0.781

> anova(resp_lmera, resp_lmerb) # adding baseline improves fit
Data: respP
Models:
resp_lmera: status ~ 1 + (1 | subject)
resp_lmerb: status ~ baseline + (1 | subject)
           Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
resp_lmera  2 260.39 267.25 -128.19   256.39
resp_lmerb  3 241.80 252.09 -117.90   235.80 20.591      1  5.687e-06 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> exp(confint(resp_lmerb))
Computing profile confidence intervals ...
                  2.5 %      97.5 %
.sig01       3.86158941  27.5463530
(Intercept)  0.03123183   0.3619759
baselinegood 6.21597124 229.2637429
>
>
```

Contraception data

- One of the data sets in the "mlmRev" package, derived from data files available on the multilevel modelling web site, is from a fertility survey of women in Bangladesh.
- One of the (binary) responses recorded is whether or not the woman currently uses artificial contraception.
- Covariates included the woman's age (on a centered scale), the number of live children she had, whether she lived in an urban or rural setting, and the district in which she lived.
- Instead of plotting such data as points, we use the 0/1 response to generate scatterplot smoother curves versus age for the different groups.

```
[1] 0.87 0.07 0.94 0.00 0.00

> summary(mC2)
Linear mixed-effects model fit by REML
Formula: score ~ gcsecnt + (1 | school) + (1 | lea)
   Data: Chem97
    AIC      BIC   logLik MLdeviance REMLdeviance
 141707 141748.7 -70848.5   141685.6       141697
Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 1.166190 1.07990
 lea      (Intercept) 0.014754 0.12147
 Residual             5.154206 2.27029
# of obs: 31022, groups: school, 2410; lea, 131

Fixed effects:
             Estimate Std. Error    DF t value  Pr(>|t|)
(Intercept) 5.6355e+00 3.1232e-02 31020  180.44 < 2.2e-16
gcsecnt     2.4726e+00 1.6904e-02 31020  146.27 < 2.2e-16

Correlation of Fixed Effects:
        (Intr)
gcsecnt 0.058
```

4 Two-level models for binary data

When the response variable is binary or when it represents a count we frequently model the data with a *generalized linear model* (glm) or, if we use random effects in the model, a *generalized linear mixed model* (glmm). Determining maximum likelihood estimates of the parameters in such a model is not as easy as for the linear mixed model because the likelihood for a glmm is expressed as an integral that must be approximated.

The `lmer` function has provision for fitting such models using one of three approximations. The simplest approximation is called *penalized quasi-likelihood* (PQL). This method is generally quite fast but also the least accurate of the three. The Laplace approximation is more accurate than PQL and the most accurate approximation is called *adaptive Gauss-Hermite quadrature* (AGQ). At present AGQ can only be used on models that have one-dimensional random effects associated with a single grouping factor.

4.1 The contraception use data

A fertility survey of women in Bangladesh included as a response their use of artificial contraception. Some of the covariates included the woman's age

(on a centered scale), the number of live children she had, whether she lived in an urban or rural setting, and the district in which she lived.

```
> str(Contraception)
'data.frame':        1934 obs. of  6 variables:
 $ woman   : Factor w/ 1934 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ district: Factor w/ 60 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ use     : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
 $ livch   : Factor w/ 4 levels "0","1","2","3+": 4 1 3 4 1 1 4 4 2 4 ...
 $ age     : num   18.44  -5.56   1.44   8.44 -13.56 ...
 $ urban   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
```

4.2 Fitting the model used in the review

The "Software reviews of multilevel modeling packages" fit a simple model that includes fixed effects for `urban`, `age`, and `livch` and a simple additive random effect by district. We reproduce this fit as

```
> system.time(Cm1 <- lmer(use ~ age + urban + livch + (1 | district),
+     Contraception, binomial))
[1] 0.14 0.00 0.14 0.00 0.00

> Cm1
Generalized linear mixed model fit using PQL
Formula: use ~ age + urban + livch + (1 | district)
   Data: Contraception
 Family: binomial(logit link)
      AIC      BIC    logLik deviance
 2429.664 2474.203 -1206.832 2413.664
Random effects:
     Groups        Name    Variance    Std.Dev.
   district (Intercept)     0.21518     0.46387
# of obs: 1934, groups: district, 60

Estimated scale (compare to 1)  0.9844111

Fixed effects:
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -1.6606460  0.1452147 -11.4358 < 2.2e-16
age         -0.0261558  0.0078152  -3.3468 0.0008176
urbanY       0.7193097  0.1183317   6.0788 1.211e-09
livch1       1.0921026  0.1565011   6.9782 2.989e-12
livch2       1.3545533  0.1729641   7.8314 4.824e-15
livch3+      1.3241531  0.1773558   7.4661 8.262e-14

> if (exists("nlminb", mode = "function")) {
+     system.time(Cm2 <- lmer(use ~ age + urban + livch + (1 |
+         district), Contraception, binomial, method = "Laplace"))
+     Cm2
+     system.time(Cm3 <- lmer(use ~ age + urban + livch + (1 |
+         district), Contraception, binomial, method = "AGQ"))
+     Cm3
+ }
```

Contraception

outcome [redacted] use

age live children (0, +) urban/rural (1, 2)

district

1934 women median 26 woman/distric

use N (0, 1)

clist mean use/district

ch livch=0, livch>0

model m4 (one of many)

look at effect of urban vs rural on use (variation over districts)

Level 1 t-test on urban/rural

within district use ~ urban $\alpha_0 + \alpha_1 urban$

α_0 = mean rural

Level 2

$\alpha_0 = age * ch + I(age^2) + u_0$

$\alpha_i = \gamma_{10} + u_1$

$\alpha_0 = \gamma_{00} + \gamma_{01} age + \gamma_{02} ch + \gamma_{03} age:ch + \gamma_{04} age^2 + u_0$

```
Generalized linear mixed model fit using Laplace
Formula: use ~ age + urban + livch + (urban | district)
   Data: Contraception
 Family: binomial(logit link)
      AIC      BIC    logLik deviance
 2418.972 2474.645 -1199.486 2398.972
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 district (Intercept) 0.38219  0.61821
          urbanY      0.64307  0.80192  -0.799
# of obs: 1934, groups: district, 60

Estimated scale (compare to 1)  1.007335

Fixed effects:
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -1.7074041  0.1572532 -10.8577 < 2.2e-16
age         -0.0264839  0.0079082  -3.3489 0.0008112
urbanY       0.8122464  0.1681257   4.8312 1.357e-06
livch1       1.1245573  0.1580051   7.1172 1.101e-12
livch2       1.3667151  0.1745854   7.8283 4.943e-15
livch3+      1.3526590  0.1795440   7.5339 4.926e-14
```

4.3 Data exploration

As with the `Exam` data, we examine several data plots when formulating a model for these data. Because the response is either `"Y"` or `"N"`, plots of the data points themselves tend to be uninformative because of overplotting. However, with a continuous covariate such as `age` it is useful to plot the scatterplot smoother line for the response versus the covariate to see the trend of the probability.

The model being fit assumes that the logistic transformation of the probability of a woman using artificial contraception is approximately linear in `age` given her urban/rural status and the number of live children she has. Figure 16, produced by shows that this is not the case. Indeed the proportion of women using artificial contraception is more like a quadratic in age. Very young women and older women are less likely to use artificial contraception.

We also see that the differences according to the number of live children are predominantly differences between those women who have live children and those who don't.

4.4 Reformulated models

We include a quadratic term in `age`

```
> if (exists("nlminb", mode = "function")) {
+     (Cm6 <- lmer(use ~ age + I(age^2) + urban + livch + (1 |
```

```
+         district), Contraception, binomial, method = "AGQ"))
+     anova(Cm1, Cm6)
+ }
Data: Contraception
Models:
Cm1: use ~ age + urban + livch + (1 | district)
Cm6: use ~ age + I(age^2) + urban + livch + (1 | district)
    Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
Cm1  8  2429.7  2474.2 -1206.8
Cm6  9  2390.5  2440.6 -1186.2 41.178      1  1.390e-10
```

and we see that, as indicated by the plots, the quadratic term is highly significant.

We can check if the differences in the number of live children is primarily the difference between no children and any children by fitting a model with the indicator `livch == 0` as one of the terms in the fixed effects.

```
> if (exists("nlminb", mode = "function")) {
+     (Cm7 <- lmer(use ~ age + I(age^2) + urban + (livch == 0) +
+         (1 | district), Contraception, binomial, method = "AGQ"))
+     anova(Cm6, Cm7)
+ }
Data: Contraception
Models:
Cm7: use ~ age + I(age^2) + urban + (livch == 0) + (1 | district)
Cm6: use ~ age + I(age^2) + urban + livch + (1 | district)
    Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
Cm7  7  2386.9  2425.9 -1186.5
Cm6  9  2390.5  2440.6 -1186.2 0.4572      2     0.7956
```

Based on the change in the log-likelihood the simpler model is adequate. We add a variable `children` which is an indicator of any children versus no children to the data set

```
> Contraception$children <- with(Contraception, factor(ifelse(livch ==
+     0, "N", "Y")))
```

so we can check for an interaction with the `urban` factor.

```
> if (exists("nlminb", mode = "function")) {
+     (Cm8 <- lmer(use ~ age + I(age^2) + urban * children + (1 |
+         district), Contraception, binomial, method = "AGQ"))
+ }
Generalized linear mixed model fit using AGQ
Formula: use ~ age + I(age^2) + urban * children + (1 | district)
   Data: Contraception
 Family: binomial(logit link)
      AIC      BIC    logLik deviance
 2387.868 2432.406 -1185.934 2371.868
Random effects:
     Groups        Name    Variance    Std.Dev.
   district (Intercept)      0.2274     0.47686
# of obs: 1934, groups: district, 60
```

```
Estimated scale (compare to 1)  1.027846

Fixed effects:
                    Estimate  Std. Error z value  Pr(>|z|)
(Intercept)      -1.07644093  0.19034365 -5.6552 1.556e-08
age               0.00554896  0.00785464  0.7065    0.4799
I(age^2)         -0.00460180  0.00071638 -6.4237 1.330e-10
urbanY            0.87145946  0.22228750  3.9204 8.840e-05
childrenY         0.95142628  0.18072909  5.2644 1.407e-07
urbanY:childrenY -0.25574959  0.25002441 -1.0229    0.3064
```

The interaction is not significant.

We can check if there is significant variation by district in the effect of urban versus rural or in the effect of any children versus no children by fitting further models

```
> if (exists("nlminb", mode = "function")) {
+     (Cm9 <- lmer(use ~ age + I(age^2) + urban + children + (urban |
+         district), Contraception, binomial, method = "Laplace"))
+     anova(Cm7, Cm9)
+     (Cm10 <- lmer(use ~ age + I(age^2) + urban + children + (children |
+         district), Contraception, binomial, method = "Laplace"))
+     anova(Cm7, Cm10)
+ }
Data: Contraception
Models:
Cm7: use ~ age + I(age^2) + urban + (livch == 0) + (1 | district)
Cm10: use ~ age + I(age^2) + urban + children + (children | district)
     Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
Cm7   7  2386.9  2425.9 -1186.5
Cm10  9  2387.9  2438.0 -1184.9 3.0549      2     0.2171
```

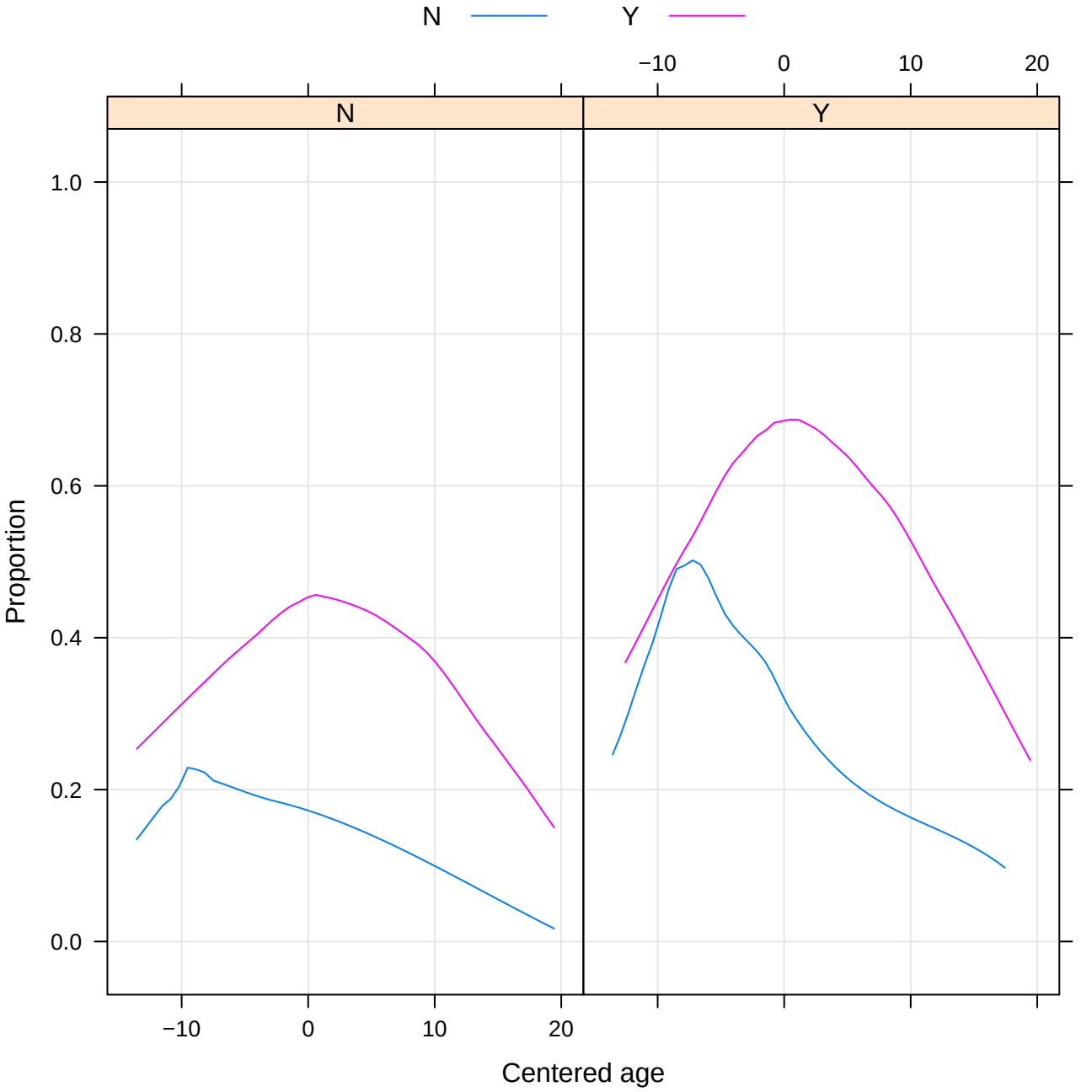
Variation due to district in the effect of any children versus no children does not appear to be significant but there is a significant variation in the effect of urban versus rural. It is interesting that AIC (Akaike's Information Criterion) and BIC (Schwartz's Bayesian Information Criterion) lead to different conclusions in the comparison of `Cm7` versus `Cm9`. For both these criteria the preferred model is the one with the lower value of the criterion. Hence, AIC prefers `Cm9` and BIC prefers `Cm7`. The BIC criterion puts a heavier penalty on having a greater number of parameters.

At present the code for creating a Markov Chain Monte Carlo sample from the distribution of the parameters in a generalized linear mixed model does not allow for random effects of dimension greater than one so we produce a sample from the posterior distribution of the parameters in model `Cm11`

```
> if (exists("nlminb", mode = "function")) {
+     (Cm11 <- lmer(use ~ age + I(age^2) + urban + children + (1 |
+         district), Contraception, binomial, method = "AGQ"))
+     Cs11 <- mcmcsamp(Cm11, 10000, trans = FALSE)
+ }
```

```
> library(lme4)
> library(lattice)
> library(mlmRev)
> data(Contraception)
> str(Contraception)
'data.frame':   1934 obs. of  6 variables:
 $ woman   : Factor w/ 1934 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ district: Factor w/ 60 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ use     : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
 $ livch   : Factor w/ 4 levels "0","1","2","3+": 4 1 3 4 1 1 4 4 2 4 ...
 $ age     : num  18.44 -5.56 1.44 8.44 -13.56 ...
 $ urban   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
> attach(Contraception)
> length(unique(woman))
[1] 1934
> fivenum(tapply(woman,district, length))
    3    36    39    40    14
  2.0  17.5  26.0  41.5 118.0
> fivenum(tapply(as.numeric(use),district, mean))
      11        1        8       50        3
1.000000 1.261538 1.381497 1.482744 2.000000
> head(Contraception)
  woman district use livch      age urban
1     1        1   N    3+  18.4400     Y
2     2        1   N     0  -5.5599     Y
3     3        1   N     2   1.4400     Y
4     4        1   N    3+   8.4400     Y
5     5        1   N     0 -13.5590     Y
6     6        1   N     0 -11.5600     Y

> table(urban, district)
     district
urban   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
    N  54  20   0  19  37  58  18  35  20  13  21  23  16  17  14  18  24  33  22  15  10  20  15  14  49  13  39  45  25  45  27  24   7  26
    Y  63   0   2  11   2   7   0   2   3   0   0   6   8 101   8   2   0  14   4   0   8   0   0   0  18   0   5   4   7  16   6   0   7   9
     district
urban  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  55  56  57  58  59  60  61
    N  28  14  13   7  24  12  23   6  28  27  34  74   9  26   4  15  20  42   0   0  24  23  20  10  22  31
    Y  20   3   0   7   2  29   3   5  17   0   5  12   6  16   0   4  17  19  19   6  21   4  13   0  10  11

> tapply(as.numeric(urban), district, mean)
       1        2        3        4        5        6        7        8        9       10       11       12       13       14       15       16
1.538462 1.000000 2.000000 1.366667 1.051282 1.107692 1.000000 1.054054 1.130435 1.000000 1.000000 1.206897 1.333333 1.855932 1.363636 1.100000
      17       18       19       20       21       22       23       24       25       26       27       28       29       30       31       32
1.000000 1.297872 1.153846 1.000000 1.444444 1.000000 1.000000 1.000000 1.268657 1.000000 1.113636 1.081633 1.218750 1.262295 1.181818 1.000000
      33       34       35       36       37       38       39       40       41       42       43       44       45       46       47       48
1.500000 1.257143 1.416667 1.176471 1.000000 1.500000 1.076923 1.707317 1.115385 1.454545 1.377778 1.000000 1.128205 1.139535 1.400000 1.380952
      49       50       51       52       53       55       56       57       58       59       60       61
1.000000 1.210526 1.459459 1.311475 2.000000 2.000000 1.466667 1.148148 1.393939 1.000000 1.312500 1.261905

> Contraception$useN = as.numeric(Contraception$use) - 1  # need 0,1 outcome for lmList
> cList = lmList(useN ~ 1|district, family = binomial, data = Contraception)
> cList2 = lmList(useN ~ age + I(age^2)|district, family = binomial, data = Contraception)

> Contraception$ch <- factor(Contraception$livch != 0, labels = c("N","Y")) # child or not
```

```
> print(xyplot(ifelse(use == "Y", 1, 0) ~ age|urban, Contraception,   # 2nd in series of plots
+              groups = ch, type = c("g", "smooth"),
+              auto.key = list(space = "top", points = FALSE,
+              lines = TRUE, columns = 2),
+              ylab = "Proportion", xlab = "Centered age"))
Waiting to confirm page change..
# plot in class

> # variation over districts in effect of urban vs rural
> m4 = glmer(use ~ age*ch + I(age^2) + urban + (urban|district), Contraception, binomial)
Warning message:
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model is nearly unidentifiable: very large eigenvalue
 - Rescale variables?

> summary(m4)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: use ~ age * ch + I(age^2) + urban + (urban | district)
   Data: Contraception

     AIC      BIC   logLik deviance df.resid
  2371.5   2421.6  -1176.8   2353.5     1925

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.9337 -0.7335 -0.4457  0.8973  3.0327

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 district (Intercept) 0.3782   0.6150
          urbanY      0.5260   0.7253   -0.79
Number of obs: 1934, groups:  district, 60

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.3441349  0.2236004  -6.011 1.84e-09 ***
age         -0.0461793  0.0219057  -2.108  0.03502 *
chY          1.2115316  0.2086851   5.806 6.42e-09 ***
I(age^2)    -0.0056507  0.0008496  -6.651 2.91e-11 ***
urbanY       0.7901326  0.1633220   4.838 1.31e-06 ***
age:chY      0.0664651  0.0255739   2.599  0.00935 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
         (Intr) age    chY    I(g^2) urbanY
age       0.692
chY      -0.854 -0.789
I(age^2) -0.093  0.298 -0.096
urbanY   -0.368 -0.060  0.087 -0.011
age:chY  -0.570 -0.928  0.672 -0.496  0.051
> exp(fixef(m4))  # odds metric
(Intercept)         age         chY    I(age^2)      urbanY     age:chY
  0.2607652   0.9548708   3.3586247   0.9943652   2.2036887   1.0687237
> exp(confint(m4))  # odds metric
```


N
Y
N
Y
1.0
0.8
0.6
0.4
0.2
0.0
Proportion
−10
0
10
20
Centered age

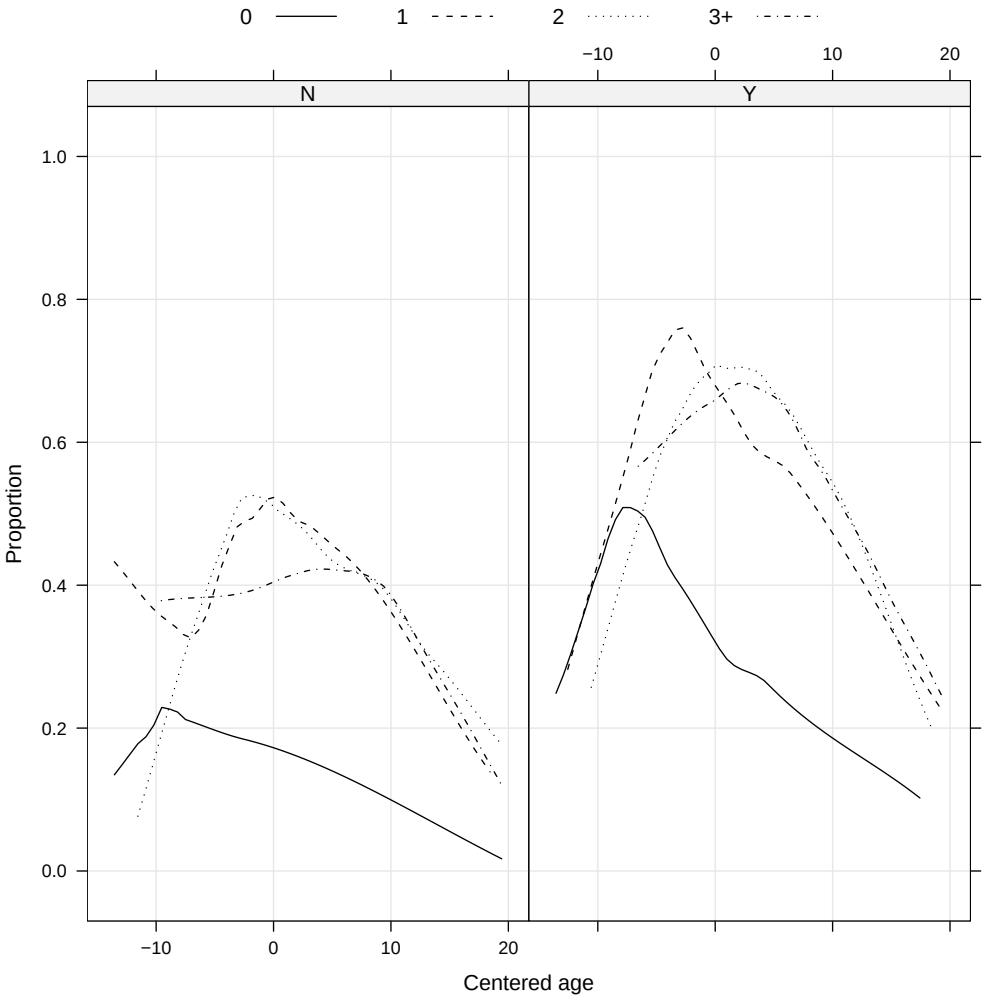
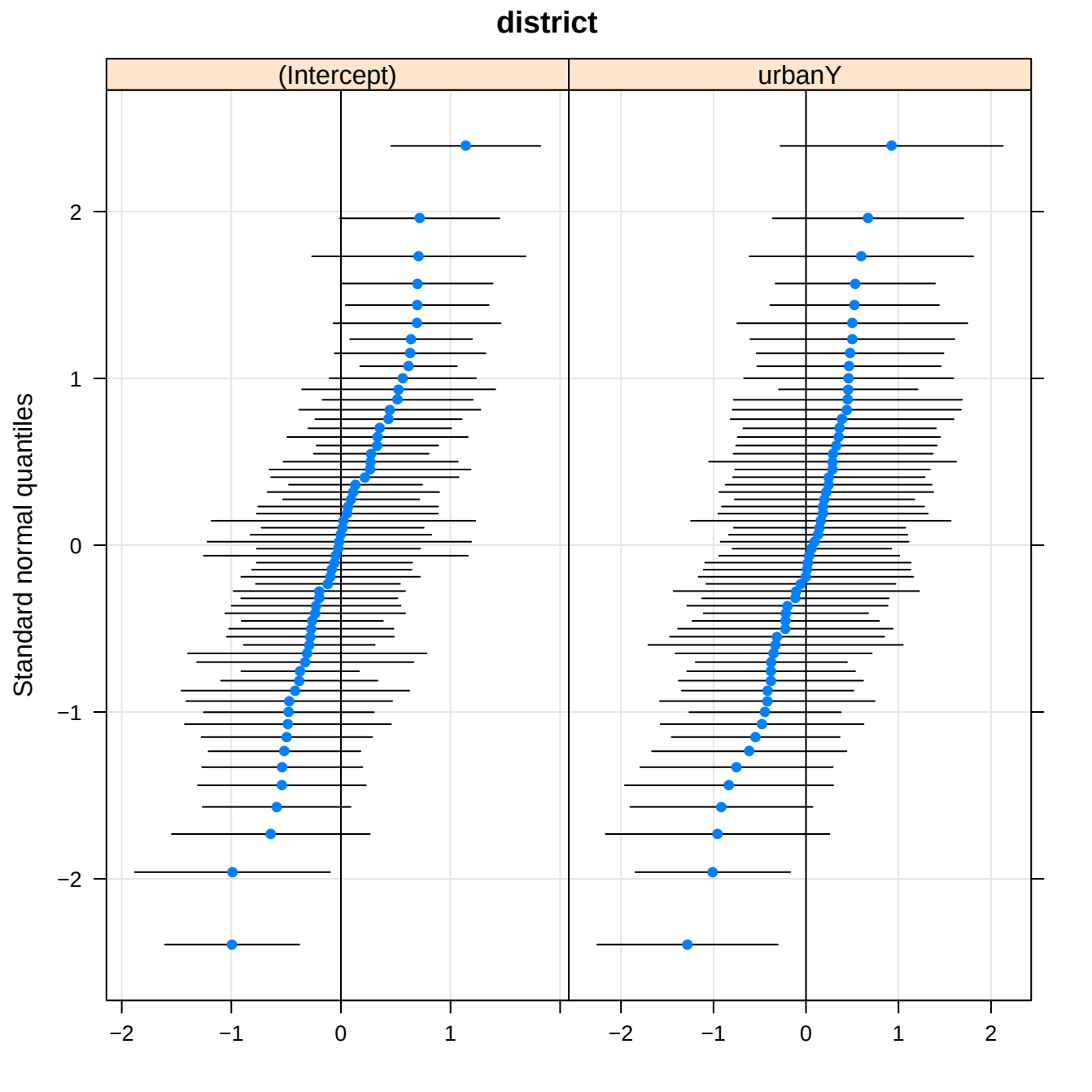


Figure 16: Scatterplot smoother curves of the use of artificial contraception versus age for women in the Bangladesh fertility survey. The panel on the left depicts the proportion for rural women and the panel on the right depicts the proportion for urban women.

```
##Prediction intervals for the bivariate random e ects
print(qqmath(ranef(m4, post = TRUE))$district)

#################### refs
http://lme4.r-forge.r-project.org/slides/2011-01-11-Madison/5GLMMH.pdf
http://lme4.r-forge.r-project.org/slides/2011-01-11-Madison/5GLMM.R
```


district
(Intercept)
urbanY
Standard normal quantiles
2
1
0
−1
−2
−2
−1
0
1
2

8

Multilevel Generalized Linear Models

In the previous chapter, we introduced generalized linear models (GLMs) that are useful when the outcome variable of interest is categorical in nature. We described a number of models in this broad family, including logistic regression for binary, ordinal, and multinomial data distributions along with Poisson regression models for count or frequency data. In the examples given, the data were collected at a single level. However, just as is true for normally distributed outcome variables, it is common for categorical variables to be gathered in a multilevel framework. The focus of this chapter is on models designed specifically for scenarios in which the outcome of interest is either categorical or counted and the data have been collected in a multilevel framework. Chapter organization will mirror that of Chapter 7. We will start with a description of fitting logistic regression for dichotomous data, followed by models for ordinal and nominal dependent variables. The chapter will conclude with models for frequency count data that fit the Poisson distribution and overdispersed counts.

Chapter 7 provided the relevant mathematical underpinnings for these various models in the single-level case. Chapter 2 introduced some of the theory underlying multilevel models. This chapter will focus almost exclusively on the application of the R software package to fit these models and on the interpretation of the resultant output.

8.1 Multilevel Generalized Linear Model for Dichotomous Outcome Variable

To introduce multilevel generalized linear models (MGLMs) for dichotomous outcomes, let us consider the following example. A researcher has collected testing data indicating whether 9,316 students passed a state mathematics assessment, along with several measures of mathematics aptitude that were obtained before administration of the achievement test. She is interested in whether a relationship exists between the score on number sense aptitude and the likelihood that a student will achieve a passing score on the mathematics achievement test, for which all examinees are categorized as either passing (1) or failing (0). Because the outcome variable is dichotomous, we could use the binary logistic regression method introduced in Chapter 7.

```
  Correlation:
          (Intr)  numsns   female
numsense  -0.839
female    -0.070   0.000
L_Free    -0.502   0.023   -0.003

Standardized Within-Group Residuals:
       Min          Q1         Med          Q3         Max
-5.0701013  -0.7049072   0.2744782   0.6513570   3.8891426

Number of Observations: 9316
Number of Groups: 34
```

A statistically significant negative relationship exists between the proportion of students in the school receiving free lunches and the likelihood that a student will pass the mathematics assessment based on a coefficient value of –0.008517 and a p value of 0.0235. In addition, the coefficient for free lunch is unrelated to that of number sense or gender, but is negatively associated with the intercept. Results for the other predictor variables are largely the same as in the prior analyses.

8.3 Fitting Multilevel Dichotomous Logistic Regression Using `lme4`

The previous examples in this chapter were based on the `nlme` library in R. However, as noted previously in this chapter and earlier in this book, researchers can use functions from the alternative `lme4` library to conduct essentially the same analyses demonstrated above. The specific function for fitting GLMs with `lme4` is `glmer`. From a mathematical perspective, the algorithms used to obtain parameter estimates in `nlme` and `lme4` differ in some fundamental ways. While `glmmPQL` is based on the penalized quasi-likelihood method, `glmer` relies on an adaptive Gauss–Hermite likelihood approximation (Liu & Pierce, 1994) to fit the model to the data. As with the partial quasi-likelihood, we will not devote time to the technical specifications of this method for fitting the model to the data. However, interested readers are referred to the Liu & Pierce work for a description of this method.

As an introduction to `glmer`, we will fit the simple random intercept model with number sense as the independent variable. The R command and resultant output appear below.

```
summary(model8.5<-glmer(score2~numsense+(1|school),family =
  binomial, nAGQ = 25))

Generalized linear mixed model fit by the Laplace approximation
Formula: score2 ~ numsense + (1 | school)
```

```
### Multilevel book Ch 8

> math = read.csv("D:\\drr15\\ed401D\\week2\\MathFinal.csv")
# need to go to book website, and download the csv data
> head(math)
  TestDurationInMinutesf10 TestRITScoref10 Goal1RitScoref10 Goal2RitScoref10
1                       NA              NA               NA               NA
2                       26             213              199              208
3                       31             186              191              179
4                       16             192              199              170
5                       15             161              168              160
6                       NA              NA               NA               NA
  Goal3RitScoref10 Goal4RitScoref10 caucasian L_Free L_Reduced Salary ISTEPM
1               NA               NA        NA  83.87      4.30  49404      3
2              212              223         0     NA        NA     NA     NA
3              184              188         0  81.27      8.36  36656      3
4              181              207         0  24.68      5.63  39938      3
5              158              155         0  52.20     13.83  34631      1
6               NA               NA        NA  70.43      6.46  40412      1
  specialed female score score2 school
1         0     NA    NA     NA      2
2         0     NA    NA     NA      3
3         0     NA     1      0      6
4         0     NA     1      0      7
5         0     NA     1      0     13
6         0     NA    NA     NA     15

> summary(math) # 1/3 scores missing
 TestDurationInMinutesf10 TestRITScoref10 Goal1RitScoref10 Goal2RitScoref10 Goal3RitScoref10 Goal4RitScoref10   caucasian          L_Free
 Min.   :   5.00          Min.   :110.0   Min.   :107.0    Min.   :100.0    Min.   :105.0    Min.   :100.0    Min.   :0.0000   Min.   : 0.00
 1st Qu.:  28.00          1st Qu.:175.0   1st Qu.:176.0    1st Qu.:173.0    1st Qu.:176.0    1st Qu.:177.0    1st Qu.:0.0000   1st Qu.:46.12
 Median :  37.00          Median :200.0   Median :201.0    Median :198.0    Median :199.0    Median :200.0    Median :0.0000   Median :67.73
 Mean   :  40.32          Mean   :195.7   Mean   :197.3    Mean   :195.1    Mean   :196.3    Mean   :197.2    Mean   :0.1972   Mean   :59.14
 3rd Qu.:  48.00          3rd Qu.:218.0   3rd Qu.:220.0    3rd Qu.:218.0    3rd Qu.:220.0    3rd Qu.:219.0    3rd Qu.:0.0000   3rd Qu.:78.01
 Max.   :1093.00          Max.   :289.0   Max.   :304.0    Max.   :299.0    Max.   :293.0    Max.   :280.0    Max.   :1.0000   Max.   :95.68
 NA's   :7                NA's   :7       NA's   :16       NA's   :16       NA's   :25       NA's   :86       NA's   :7        NA's   :3421
   L_Reduced          Salary          ISTEPM        specialed          female           score           score2          school
 Min.   : 0.000   Min.   :    0   Min.   :1.000   Min.   :0.00000   Min.   :0.0000   Min.   :1.000   Min.   :0.000   Min.   : 2.00
 1st Qu.: 5.900   1st Qu.:32939   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:14.00
 Median : 8.360   Median :38452   Median :1.000   Median :0.00000   Median :1.0000   Median :2.000   Median :1.000   Median :28.00
 Mean   : 9.772   Mean   :38099   Mean   :1.829   Mean   :0.02553   Mean   :0.5016   Mean   :1.651   Mean   :0.565   Mean   :26.82
 3rd Qu.:10.600   3rd Qu.:41247   3rd Qu.:3.000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:39.00
 Max.   :72.800   Max.   :62760   Max.   :4.000   Max.   :1.00000   Max.   :1.0000   Max.   :3.000   Max.   :1.000   Max.   :53.00
 NA's   :3421     NA's   :3694    NA's   :3640                      NA's   :18       NA's   :3453    NA's   :3453
> length(unique(math$school)) # only 40 have data
[1] 52
> attach(math)
> table(school, score2)
      score2
school   0   1
    2    0   0
    3   72 118
    4   81 122
    5  160 254
    6  173 213
    7   57 120
    8    0   0
    9  112 159
    10 129  86
    11 105 153
    12  25 138
    13  41  48
    14 118 178
    15   0   0
    16 112 170
    17  55  59
    18  54  74
    19 191 190
    20  20  60
    21  57 148
    22   0   0
    23  42 181
    24  24  36
    25   0   0
```

```
    26 282 133
    27   0   0
    28 187 124
    29 214 339
    30 143  82
    31  82 160
    32   0   0
    33 178 143
    34   5  35
    35   0   0
    36  73 285
    37  98  80
    38 124  94
    39 157 137
    40  31  57
    41   0   0
    42   0   0
    43   0   0
    44  74  82
    45  32  84
    46  46  88
    47  33  57
    48 268 423
    49   0   0
    50  83  33
    51  36  60
    52 191 175
    53  84  89
> table(score2, school)
      school
score2   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
     0   0  72  81 160 173  57   0 112 129 105  25  41 118   0 112  55  54 191  20  57   0  42  24   0 282   0 187 214 143  82   0 178   5   0  73
     1   0 118 122 254 213 120   0 159  86 153 138  48 178   0 170  59  74 190  60 148   0 181  36   0 133   0 124 339  82 160   0 143  35   0 285
      school
score2  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
     0  98 124 157  31   0   0   0  74  32  46  33 268   0  83  36 191  84
     1  80  94 137  57   0   0   0  82  84  88  57 423   0  33  60 175  89
> # kill the schools with 0 data
> tapply(score2,school, length)
   2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
   1  344  322  576  536  177   19  416  215  324  163   89  380    1  363  114  201  552   80  270  168  283   60    1  535    1  412  691  359
  31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53
 323   85  480   58    1  481  178  218  409   88   58    1   79  156  116  212  104 1122    1  116  124  489  217
> tapply(score2,school, mean) # very few schools have complete data on the outcome
        2         3         4         5         6         7         8         9        10        11        12        13        14        15
       NA        NA        NA        NA        NA 0.6779661        NA        NA 0.4000000        NA 0.8466258 0.5393258        NA        NA
       16        17        18        19        20        21        22        23        24        25        26        27        28        29
       NA 0.5175439        NA        NA 0.7500000        NA        NA        NA 0.6000000        NA        NA        NA        NA        NA
       30        31        32        33        34        35        36        37        38        39        40        41        42        43
       NA        NA        NA        NA        NA        NA        NA 0.4494382 0.4311927        NA 0.6477273        NA        NA        NA
       44        45        46        47        48        49        50        51        52        53
0.5256410 0.7241379        NA        NA        NA        NA 0.2844828        NA        NA        NA
> dim(math)
[1] 12769    16

> library(lme4)
> names(math)
 [1] "TestDurationInMinutesf10" "TestRITScoref10"          "Goal1RitScoref10"         "Goal2RitScoref10"         "Goal3RitScoref10"
 [6] "Goal4RitScoref10"         "caucasian"                "L_Free"                   "L_Reduced"                "Salary"
[11] "ISTEPM"                   "specialed"                "female"                   "score"                    "score2"
[16] "school"

> fivenum(Goal1RitScoref10)
[1] 107 176 201 220 304
> fivenum(Goal2RitScoref10)
[1] 100 173 198 218 299
> fivenum(Goal3RitScoref10) # this is my guess at numsense
[1] 105 176 199 220 293
> fivenum(Goal4RitScoref10)
[1] 100 177 200 219 280


> model8.5<-glmer(score2~Goal3RitScoref10+(1|school),family = binomial, nAGQ = 25, data = math)
Warning message:
```

```
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model is nearly unidentifiable: very large eigenvalue
 - Rescale variables?
> summary(model8.5)
Generalized linear mixed model fit by maximum likelihood (Adaptive Gauss-Hermite Quadrature, nAGQ = 25) ['glmerMod']
 Family: binomial  ( logit )
Formula: score2 ~ Goal3RitScoref10 + (1 | school)
   Data: math

     AIC      BIC   logLik deviance df.resid
 10045.3  10066.7  -5019.6  10039.3     9313

Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.0920 -0.7272  0.2954  0.6739  3.6525

Random effects:
 Groups Name        Variance Std.Dev.
 school (Intercept) 0.3033   0.5507
Number of obs: 9316, groups:  school, 40

Fixed effects:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -11.124358   0.300062  -37.07   <2e-16 ***
Goal3RitScoref10   0.056835   0.001424   39.91   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr)
Gl3RtScrf10 -0.952

> model8.6<-glmer(score2~Goal3RitScoref10+(Goal3RitScoref10|school),family = binomial,  data = math)
Warning messages:
1: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with max|grad| = 0.00189563 (tol = 0.001, component 1)
2: In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model is nearly unidentifiable: very large eigenvalue
 - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
 - Rescale variables?

> summary(model8.6)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: score2 ~ Goal3RitScoref10 + (Goal3RitScoref10 | school)
   Data: math

     AIC      BIC   logLik deviance df.resid
  9977.6  10013.3  -4983.8   9967.6     9311

Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.0100 -0.7213  0.2901  0.6679  3.7673

Random effects:
 Groups Name             Variance  Std.Dev. Corr
 school (Intercept)      1.943e+01 4.40783
        Goal3RitScoref10 3.799e-04 0.01949  -0.99
Number of obs: 9316, groups:  school, 40

Fixed effects:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -12.428997   0.794990  -15.63   <2e-16 ***
Goal3RitScoref10   0.062980   0.003607   17.46   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr)
Gl3RtScrf10 -0.992
>
>
> anova(model8.5, model8.6)
Data: math
```

```
Models:
model8.5: score2 ~ Goal3RitScoref10 + (1 | school)
model8.6: score2 ~ Goal3RitScoref10 + (Goal3RitScoref10 | school)
         Df     AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
model8.5  3 10045.3 10067 -5019.6  10039.3
model8.6  5  9977.6 10013 -4983.8   9967.6 71.738      2  2.644e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
    AIC    BIC  logLik  deviance
  10019  10040   -5006     10013
Random effects:
  Groups Name          Variance   Std.Dev.
  school (Intercept)    0.25708    0.50703
Number of obs: 9316, groups: school, 40

Fixed effects:
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept)  -11.099903     0.274561    -40.43     <2e-16 ***
numsense       0.056219     0.001352     41.59     <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
         (Intr)
numsense -0.994
```

The function call is similar to that for `glmmPQL`. The only difference is the manner in which the random effect is specified. In addition, the parameter estimates and standard errors for the fixed effects, the correlation between the fixed effects parameters, and the estimate of intercept variation across schools are also very similar to those for Model 8.1. The results from `glmer` include values for the AIC and BIC. As noted in previous chapters, they can be used to compare the relative fits of various models in an attempt to pick the optimal one. These values were not available with `glmmPQL`.

The fit of two nested models created by `glmer` can be compared with one another in the form of a likelihood ratio test with the `anova` function. The null hypothesis of this test is that the fit of two nested models is equivalent, so that a statistically significant result (i.e., $p \leq 0.05$) would indicate that the models provide different fits to the data, with the more complicated (fuller) model typically providing an improvement in fit beyond what would be expected with the additional parameters added. The `anova` function is not available for comparing model fits using `glmmPQL`.

Just as we fit the random intercept model, it is also possible to fit the random coefficients model using `glmer`, as below.

```
summary(model8.6<-glmer(score2~numsense+(numsense|school),
  family = binomial))
Generalized linear mixed model fit by the Laplace approximation
Formula: score2 ~ numsense + (numsense | school)
   AIC   BIC  logLik  deviance
  9769  9804   -4879      9759
Random effects:
       Groups Name     Variance   Std.Dev.     Corr
school (Intercept)  2.1701e+01   4.658438
          numsense  4.1048e-04   0.020260  -0.996
Number of obs: 9316, groups: school, 40
```

```
Fixed effects:
             Estimate   Std. Error  z value   Pr(>|z|)
(Intercept) -12.901317    0.819828   -15.74     <2e-16 ***
numsense      0.064902    0.003648    17.79     <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

  Correlation of Fixed Effects:
          (Intr)
numsense -0.995
```

These results are very closely aligned to those from `glmmPQL`. We have already discussed the results in some detail and will not do so again here. However, the inclusion of AIC and BIC in `GLMER` output allows a direct comparison of model fit, thus aiding in the selection of an optimal model for the data. As a brief reminder, AIC and BIC are both measures of unexplained variations in the data with a penalty for model complexity. Therefore, models with lower values provide relatively better fit. Comparison of AIC or BIC with Models 8.5 and 8.6 reveals that BIC provides better fit to the data. We do need to remember that AIC and BIC are not significance tests, but rather are measures of relative model fit. In contrast to the relative fit indices, we can compare the fit of the two models using the `anova` command:

```
anova(model8.5, model8.6)
Data:
Models:
model8.5: score2 ~ numsense + (1 | school)
model8.6: score2 ~ numsense + (numsense | school)
         Df     AIC     BIC  logLik    Chisq  Chi Df   Pr(>Chisq)
model8.5  3  10018.9 10040.4 -5006.5
model8.6  5   9768.7  9804.4 -4879.4   254.22       2    < 2.2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These results indicate a statistically significant difference in the relative fits of the two models. Furthermore, the AIC and BIC are both lower for Model 8.6, suggesting that it provides better fit to the data than Model 8.5. Thus, we can conclude that the coefficients for `numsense` are significantly different across schools. Thus allowing them to vary among the schools leads to a more optimal model than forcing them to be the same. It should be noted that this comparison of model fit carried out by the `anova` command relies on a two-degrees-of-freedom test in which a significant difference in fit may be due to the fixed effects, the random effects, or both. Another way to interpret this result is that there appear to be school differences in the relationship of number sense and the likelihood of students passing the mathematics achievement test.

The inclusion of additional independent variables at both Level 1 (student) and Level 2 (school) follows the previous model syntax structure. In the

Distribution of Counts:

Poisson distribution

<Ott 6.6; MT 6.4> <Agresti 4.2.1 pp4-6 <NWK p609>

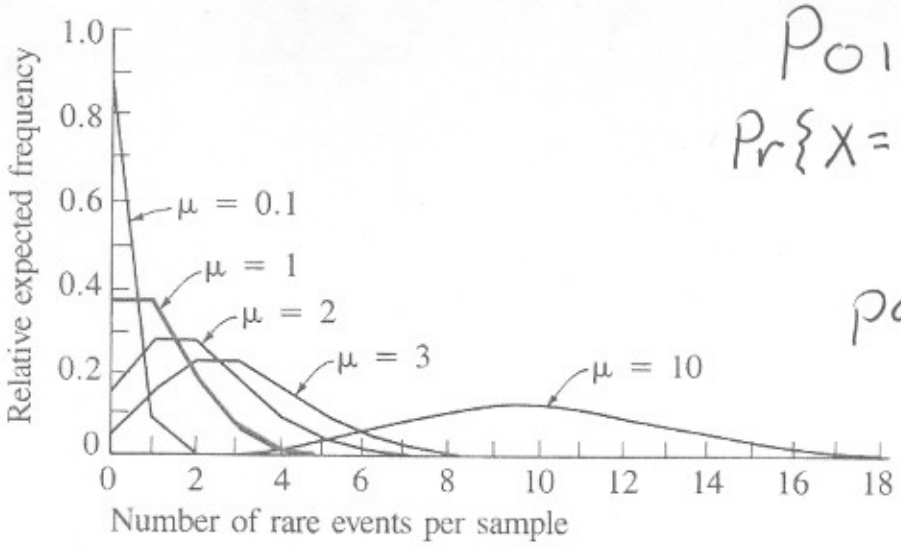
automobile accidents
side effects from medication.
important journal publications
class disruptions.

"rare" events

$$Pr\{Y=k\} = \frac{\mu^k e^{-\mu}}{k!}$$

$E(Y) = \mu$; $Var(Y) = \mu$.

lnk 2



Poisson

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

parameter λ (or μ)

$E(X) = \lambda$

$Var(X) = \lambda$

The Poisson Distribution

The Poisson distribution is a modified form of the binomial distribution that is useful for the analysis of phenomena characterized by discrete, rare events. For example, in a study of the distribution of a rare plant among a number of quadrats, a majority of plots may contain no specimens, a smaller number a single plant, and still smaller numbers two, three, or more plants. If a single plant per quadrat is expected, the mean **μ = 1** and the "**0**" and "**1**" classes occur at 37% each, the "**2**" class at 18%, the "**3**" class at 6%, and larger classes take up the remaining 2%. The characteristic test for a Poisson is that the **variance** equals the **mean**, which in the plant example means that the rare plant is randomly distributed. Note, that, In the distributions above, as the mean **μ** increases towards 10, the distribution approaches normality.

The Poisson may be used to evaluate whether events occur independently in time as well as space. In a classic example, Bortkiewicz (1898) studied the distribution of 122 men kicked to death by horses in ten Prussian army corps over 20 years. In most years in most corps, no one is kicked; in one corp in one year, four men were kicked to death. The question arises whether something was amiss in this particular corp. Analysis indicates that the observed frequencies conform quite closely to the expected Poisson frequencies, and that the mean and variance are almost identical, as expected. The corp was just unlucky.

Number of men kicked to death by horses in ten Prussian army corps

# men killed / year / corp	Observation (# deaths)	Poisson Expectation
0	109 (0)	108.7 (0.0)
1	65 (65)	66.3 (66.3)
2	22 (44)	20.2 (40.4)
3	3 (9)	4.1 (12.3)
4	1 (4)	0.6 (2.4)
5+	0 (0)	0.1 (0.5)
# corp-years	200	200.0
Total deaths	122	121.9
Mean	0.610	0.610
Variance	0.611	0.610

All text material ©2005 by Steven M. Carr; figures after Sokal & Rohlf (1993). Biometry, 3rd ed. Freeman.

Fitting Distributions to Data: Recreate Poisson Expectations Above

[redacted] "family name" for Poission is pois

```
> dpois(2, .610)
[1] 0.1010904
> dpois(2, .610)*200
[1] 20.21809
> k = 0:6
> list(k, dpois(k, .610)*200)
[1] 0 1 2 3 4 5 6
[1] 1.086702e+02 6.628881e+01 2.021809e+01 4.111011e+00 6.269291e-01
[6] 7.648536e-02 7.776011e-03
```

Poisson Distribution exs from Glover&Mitchell text
EXAMPLE 3.12 An ichthyologist studying the spoonhead sculpin, Cottus ricei, catches specimens in a large bag seine that she trolls through the lake. She knows from many years experience that on average she will catch 2 fish per trolling run. Find the probabilities of catching
(a) No fish on a particular run;
(b) Fewer than 6 fish on a particular run;
(c) Between 3 and 6 fish on a particular run.

Use Poisson Distribution (unit is an outing)
```
> rbind(k, signif(dpois(k, 2),3), signif(ppois(k, 2),3))

k 0.000 1.000 2.000 3.000 4.0000 5.0000 6.000
  0.135 0.271 0.271 0.180 0.0902 0.0361 0.012
  0.135 0.406 0.677 0.857 0.9470 0.9830 0.995
```
Y=#sculpin caught
(a) Pr{Y=0} = .135 (b) Pr{Y<=5} =.983
(c) Pr{Y<=6 and Y>= 3} = Pr{Y<= 6} - Pr{Y<= 2} = .995 - .677 = .318
wording issue: above "between" includes the endpoints 3 and 6; if not then the answer is Pr{Y=4 or 5} = .1263

Alternatively, if....
```
> # binomial approximation; discretize the outing into say 1000 seine troll segments
> # each segment having p = .002 of catch
> rbind(k, signif(dbinom(k, s=1000, p=.002),3), signif(pbinom(k, s=1000, p=.002),3))

k 0.000 1.000 2.000 3.000 4.0000 5.000 6.000
  0.135 0.271 0.271 0.181 0.0902 0.036 0.012
  0.135 0.406 0.677 0.857 0.9480 0.984 0.996
```

Discrete Distributions: Binomial-Poisson approx for np, p small

```
MTB > pdf;                              |MTB > pdf;
SUBC> poisson 1.                        |SUBC> binomial 1000 .001.
    POISSON WITH MEAN =   1.000         |    BINOMIAL WITH N =1000  P = 0.001000
       K          P( X = K)             |       K          P( X = K)
       0            0.3679              |       0            0.3677
       1            0.3679              |       1            0.3681
       2            0.1839              |       2            0.1840
       3            0.0613              |       3            0.0613
       4            0.0153              |       4            0.0153
       5            0.0031              |       5            0.0030
       6            0.0005              |       6            0.0005
       7            0.0001              |       7            0.0001
       8            0.0000              |       8            0.0000
```

Agresti Table 1.1 p.5

```
MTB > pdf;                          MTB > pdf ;
SUBC> poisson 2.                    SUBC> binomial 10 .2.
Poisson with mu = 2.00000           Binomial with n = 10 and p = 0.20

         x      P( X = x)              x      P( X = x)
         0        0.1353               0        0.1074
         1        0.2707               1        0.2684
         2        0.2707               2        0.3020
         3        0.1804               3        0.2013
         4        0.0902               4        0.0881
         5        0.0361               5        0.0264
         6        0.0120               6        0.0055
         7        0.0034               7        0.0008
         8        0.0009               8        0.0001
         9        0.0002               9        0.0000
        10        0.0000      (match less good.)
```

Spoonhead Sculpin Fact Sheet

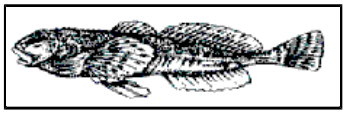
More information from this division:

Fish, Wildlife and Marine Resources
Endangered Species Program

Spoonhead Sculpin
Cottus ricei

New York Status: Endangered

Description



The spoonhead sculpin has a flat, triangle-shaped head, slender tubular-shaped body, and a well developed semicircular preopercular (cheekbone) spine. Its body has no scales but is covered with prickles (small, fine, curved spines). The gill covers are attached to the breast (or isthmus) and the pectoral (front sides) fins are long and round. The second dorsal (back) fin and the anal (bottom rear) fin are similar in length and shape. Spoonheads usually range from 1.5 to 2.4 inches in length, although one specimen was found to measure 5.3 inches. The tops of their bodies are olive-brown shading to light yellow on the sides and white on the belly. Their heads, bodies and fins are speckled.

Life History

From existing limited information, scientists have determined that spoonhead sculpins spawn in the fall and live in turbid rivers or deep areas of lakes. Scientists theorize that the fish have adapted to these waters because spoonheads have flat heads, large head pores, and small eyes. Although no detailed food studies have been carried out for this species, it is assumed that plankton and bottom-dwelling aquatic insects are important diet items.

Distribution and Habitat

Spoonheads are found mostly in Canada, from the McKenzie River in the northwest to the St. Lawrence River in the east. This fish also lives in the Great Lakes and in Lake Charlevoix, Michigan. In New York, spoonheads were historically taken in Lakes Ontario and Erie, but are now believed to be extirpated.

Population Status

Spoonheads are believed to be locally extinct in Lakes Ontario and Erie. The causes of the decline are unknown.

Management/Research Needs

The New York State Department of Environmental Conservation will continue to search for and report on spoonhead sculpins in New York waters.

To view other articles in the ***Endangered Species*** series, click on one of the following:

Shortnose Sturgeon
Bluebreast Darter
Silver Chub
Pugnose Shiner
Gilt Darter
Deepwater Sculpin
Round Whitefish

Back to top of page

Approximation for Binomial

when $n\pi$ small; e.g. $\pi = .001$
$n = 100$, Poisson is good
approx for # successes.

let $\mu = n\pi$

Computation MT: SUBC Poisson μ
for Random, PDF, CDF.

Off Ex 6.10, Ex 6.11

MT # accidents p 145-6

Transformation to normality

$Y \sim Po(\mu)$

$\sqrt{Y + 3/8} \sim N$

Discrete Distributions: Binomial-Poisson approx for np, p small

```
MTB > pdf;                                  |MTB > pdf;
SUBC> poisson 1.                            |SUBC> binomial 1000 .001.
    POISSON WITH MEAN =   1.000             |    BINOMIAL WITH N =1000  P = 0.001000
       K          P( X = K)                |       K          P( X = K)
       0            0.3679                 |       0            0.3677
       1            0.3679                 |       1            0.3681
       2            0.1839                 |       2            0.1840
       3            0.0613                 |       3            0.0613
       4            0.0153                 |       4            0.0153
       5            0.0031                 |       5            0.0030
       6            0.0005                 |       6            0.0005
       7            0.0001                 |       7            0.0001
       8            0.0000                 |       8            0.0000
```

rare event

Agresti Table 1.1 p.5

```
MTB > pdf;                                  MTB > pdf ;
SUBC> poisson 2.                            SUBC> binomial 10 .2.
Poisson with mu = 2.00000                   Binomial with n = 10 and p = 0.20

         x      P( X = x)                      x      P( X = x)
         0        0.1353                       0        0.1074
         1        0.2707                       1        0.2684
         2        0.2707                       2        0.3020
         3        0.1804                       3        0.2013
         4        0.0902                       4        0.0881
         5        0.0361                       5        0.0264
         6        0.0120                       6        0.0055
         7        0.0034                       7        0.0008
         8        0.0009                       8        0.0001
         9        0.0002                       9        0.0000
        10        0.0000
match less good.
```

not so rare

Individual Trajectories (learning curves, growth curves)

notation (varies) measurement $Y(t)$ or Y_t

$Y(t)$ contains measurement error

perfect measurement $E(Y(t)) = \eta(t)$

Straight-line Growth in $E(Y(t))$

constant rate of change

$E(Y(t)) = \eta(t) = \eta(0) + \theta t$

or $= \eta_0 + \eta_1 t$ [or center at a t^*]

$\frac{d\eta(t)}{dt}$ = constant rate of change (θ or η_1)

slope from lm($y \sim t$) estimates constant rate of change

Quadratic $\eta(t) = \eta_0 + \eta_1 t + \eta_2 t^2$

$d\eta(t)/dt = \eta_1 + 2\eta_2 t$ accelerating or decelerating

extended to polynomials

GLM versions (e.g. counts)

aids ex $E(Y(t)) = N_0 e^{\theta t}$

$\log(E(Y(t)) = \log N_0 + \theta t$ [or quadratic in t]

glm(, family = poisson)

NON-LINEAR Growth Curves

Logistic SSlogis w/ n/s

R, MEMSS parameterization $\eta(t) = \frac{\phi_1}{1+\exp[(\phi_2 - t)/\phi_3]}$

(ϕ_1: asymptote; limit as $t \to \infty$; ϕ_2: time $\eta = \phi_1/2$)

$\phi_3 > 0$ accelerating then decelerating growth

$\frac{d\eta(t)}{dt}$ quadratic in $\eta(t)$

Asymptotic Regression (linear state dependence)

SSasymp

$\frac{d\eta(t)}{dt} = (\lambda - \eta(t))\gamma$ (λ: asymptote; γ: curvature)

$\eta(t) = \lambda - (\lambda - \eta(0))e^{-\gamma t}$

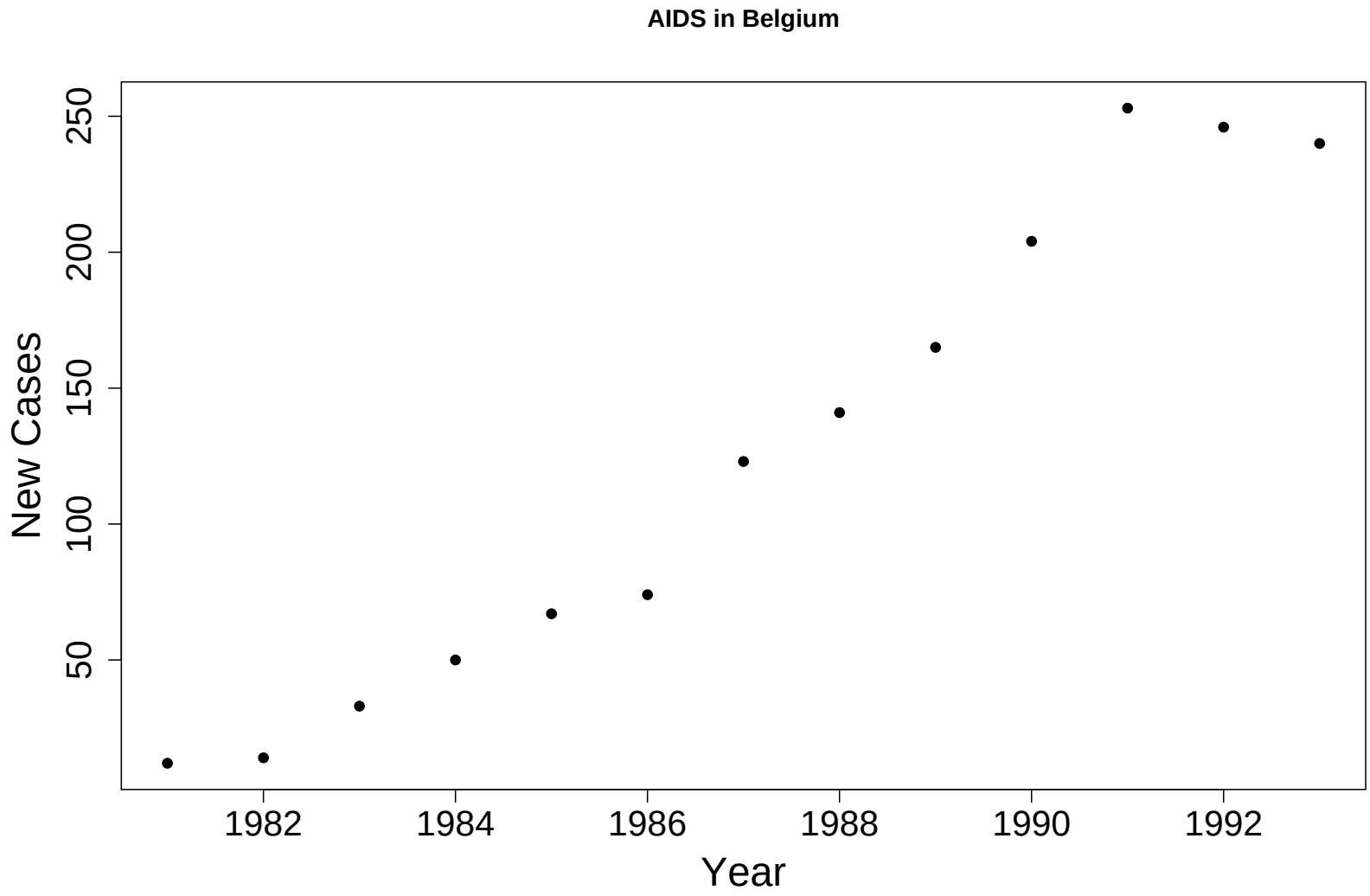
$\phi_1 = \lambda$ $\phi_2 = \eta(0)$

$\phi_3 = \gamma$ lrc

	cases	year
1	12	1
2	14	2
3	33	3
4	50	4
5	67	5
6	74	6
7	123	7
8	141	8
9	165	9
10	204	10
11	253	11
12	246	12
13	240	13

```
> Chick.1
   Time weight
1     0     42
2     2     51
3     4     59
4     6     64
5     8     76
6    10     93
7    12    106
8    14    125
9    16    149
10   18    171
11   20    199
12   21    205
```

Example: AIDS in Belgium



Example: AIDS rate model

- A simple model for these data might be

$$\mathbb{E}(y_i) = N_0 e^{\beta_1 t_i} \qquad y_i \sim \text{Poi}$$

  - y_i is new cases in year t_i; N_0 is number of new cases in 1980.
  - Model is for exponential increase of the expected rate.
  - Observed number of cases, follows a Poisson distribution.

- The model is non-linear, but taking logs yields

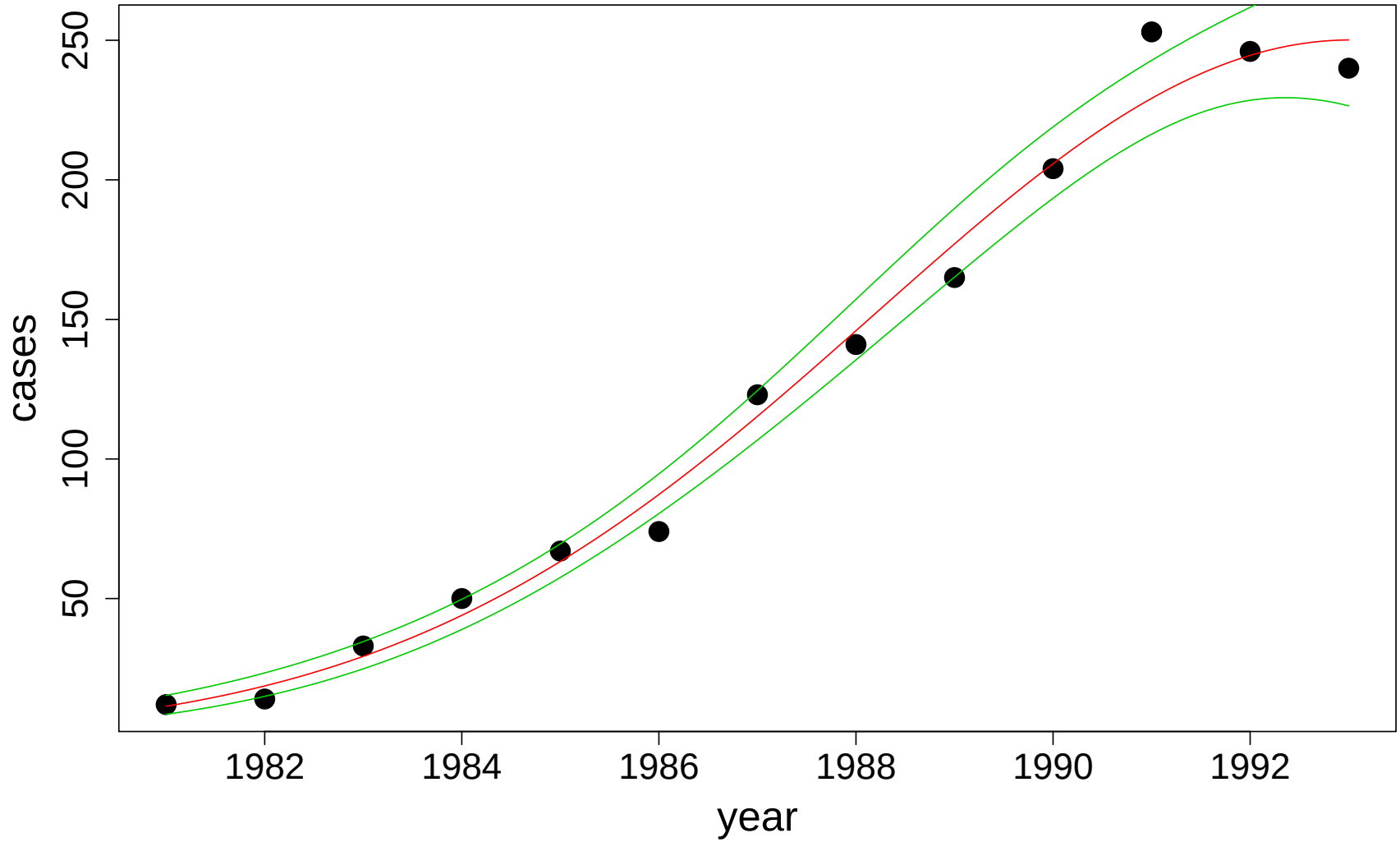
$$\begin{aligned} \log\left(\mathbb{E}(y_i)\right) &= \log(N_0) + \beta_1 t_i \\ &= \beta_0 + \beta_1 t_i, \qquad y_i \sim \text{Poi} \end{aligned}$$

  i.e. a GLM with a log link ($\beta_0 \equiv \log(N_0)$).

```
==========================================================================
#######Count Data, Generalized Linear Models
# slide for Poisson link function (log)

> am1 = glm(cases ~ year,data=belg.aids, family=poisson(link=log))
> summary(am1)
Call: glm(formula = cases ~ year, family = poisson(link = log), data = belg.aids)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6784  -1.5013  -0.2636   2.1760   2.7306
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.140590   0.078247   40.14   <2e-16 ***
year        0.202121   0.007771   26.01   <2e-16 ***
---

(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 872.206  on 12  degrees of freedom
Residual deviance:  80.686  on 11  degrees of freedom
AIC: 166.37
Number of Fisher Scoring iterations: 4

> belg.aids
   cases year
1     12    1
2     14    2
3     33    3
4     50    4
5     67    5
6     74    6
7    123    7
8    141    8
9    165    9
10   204   10
11   253   11
12   246   12
13   240   13

> plot(am1)  # gives you the set of diagnostic plots--resids vs fitted etc
# Quadratic in year
> am2 = glm(cases ~ year+I(year^2),data=belg.aids, family=poisson(link=log))
> summary(am2)
Call:
glm(formula = cases ~ year + I(year^2), family = poisson(link = log), data = belg.aids)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.45903  -0.64491   0.08927   0.67117   1.54596
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.901459   0.186877  10.175  < 2e-16 ***
year         0.556003   0.045780  12.145  < 2e-16 ***
I(year^2)   -0.021346   0.002659  -8.029 9.82e-16 ***
---
(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 872.2058  on 12  degrees of freedom
Residual deviance:   9.2402  on 10  degrees of freedom
AIC: 96.924
Number of Fisher Scoring iterations: 4

> anova(am1,am2) # compare nested models
Analysis of Deviance Table
Model 1: cases ~ year
Model 2: cases ~ year + I(year^2)
  Resid. Df Resid. Dev Df Deviance
1        11     80.686
2        10      9.240  1   71.446
```

```
> anova(am1,am2, test = "Chisq")
Analysis of Deviance Table
Model 1: cases ~ year
Model 2: cases ~ year + I(year^2)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        11     80.686
2        10      9.240  1   71.446 < 2.2e-16 ***

> AIC(am1,am2)
    df       AIC
am1  2 166.36982
am2  3  96.92358

> # cubic doesn't help, see link

> year = seq(1,13,length=100)
> fv = predict(am2,newdata=data.frame(year=year),se=TRUE)
> plot(belg.aids$year+1980,belg.aids$cases) # data
> lines(year+1980,exp(fv$fit),col=2) # fit
> lines(year+1980,exp(fv$fit+2*fv$se),col=3) # upper c.l.
> lines(year+1980,exp(fv$fit-2*fv$se),col=3) # lower c.l.
> # produces nice final plot, note the overlay of fit and CI bands (2*se)

==========================================================================
```

Fitted AIDS model



The canonical link for the Poisson distribution

- The logarithm of the probability mass is

$$\log(p(y|\mu)) = \log(y!) - \mu + y\log(\mu)$$

- Thus, the canonical link function for the Poisson is the *log* link

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

- The inverse link is

$$\mu = g^{-1}(\eta) = e^{\eta}$$

The canonical link related to the variance

- For the canonical link function, the derivative of its inverse is the variance of the response.
- For the Bernoulli, the canonical link is the logit and the inverse link is $\mu = g^{-1}(\eta) = 1/(1 + e^{-\eta})$. Then

$$\frac{d\mu}{d\eta} = \frac{e^{-\eta}}{(1 + e^{-\eta})^2} = \frac{1}{1 + e^{-\eta}} \frac{e^{-\eta}}{1 + e^{-\eta}} = \mu(1 - \mu) = \mathrm{Var}(\mathcal{Y})$$

- For the Poisson, the canonical link is the log and the inverse link is $\mu = g^{-1}(\eta) = e^{\eta}$. Then

$$\frac{d\mu}{d\eta} = e^{\eta} = \mu = \mathrm{Var}(\mathcal{Y})$$

8.5 MGLM for Count Data

In the previous chapter, we examined statistical models designed for use with outcome variables that represented the frequency of occurrence of some event. Typically these events were relatively rare, such as the addition of babies to a family. Perhaps the most common distribution associated with such counts is the Poisson in which the mean and variance are equal. However, as we saw in Chapter 7, this equality of the two moments does not always hold in all empirical contexts and the result is what is commonly called overdispersed data. In such cases, the Poisson regression model relating one or more independent variables to a count dependent variable is not appropriate, and we must make use of either the quasi-Poisson or negative binomial distribution, each of which is able to model the inequality of the mean and variance appropriately. Extending any of these models to the multilevel context is fairly straightforward, both conceptually and using R with the appropriate packages.

In the following sections, we will demonstrate analysis of multilevel count data outcomes in the context of Poisson regression, quasi-Poisson regression, and negative binomial regression in R. The example involves the number of cardiac warning incidents (e.g., chest pain, shortness of breath, dizzy spell) for 1000 patients associated with 110 cardiac rehabilitation facilities in a large state over a six-month period. Patients who recently suffered heart attacks and were entering rehabilitation agreed to be randomly assigned to a new exercise treatment program or to the standard treatment protocol. Of particular interest to the researcher leading this study is the relationship between treatment condition and the number of cardiac warning incidents. The new approach to rehabilitation is expected to result in fewer such incidents as compared to the traditional method. In addition, the researcher also collected data on patient sex and the number of hours that each rehabilitation facility is open during the week. This latter variable is of interest as it reflects the overall availability of the rehabilitation programs. The new method of conducting cardiac rehabilitation is coded in the data as 1 and the standard approach is coded 0. Males are also coded 1 and females are assigned 0 values.

8.5.1 Random Intercept Poisson Regression

The R commands and resultant output for fitting the Poisson regression model to the data appear below using the `glmmPQL` function in the `nlme` library used earlier to fit the dichotomous logistic regression models. This function will accommodate the same link functions available for the `glm` function, including the quasi-Poisson, as we will see shortly.

```
summary(model8.10<-glmmPQL(heart~trt+sex,random =
   ~1|rehab,family = poisson))
```

```
Linear mixed-effects model fit by maximum likelihood
```

```
> rehb = read.csv("D:\\drr15\\ed401D\\week3\\Rehab.csv")
> names(rehb)
[1] "heart" "trt"   "sex"   "hours" "rehab"
> summary(rehb)
     heart              trt                 sex                hours              rehab
 Min.   :  0.000   Min.   :-3.511615   Min.   :-2.980184   Min.   :-2.44794   Min.   :  1.00
 1st Qu.:  0.000   1st Qu.:-0.711387   1st Qu.:-0.706413   1st Qu.:-0.39242   1st Qu.: 45.75
 Median :  0.000   Median : 0.009853   Median :-0.001576   Median : 0.05511   Median : 70.50
 Mean   :  4.156   Mean   : 0.013700   Mean   : 0.011438   Mean   : 0.15018   Mean   : 67.00
 3rd Qu.:  4.000   3rd Qu.: 0.697434   3rd Qu.: 0.701227   3rd Qu.: 0.81811   3rd Qu.: 94.00
 Max.   :225.000   Max.   : 2.976833   Max.   : 3.315273   Max.   : 2.81099   Max.   :110.00
> attach(rehb)
> head(rehb)
  heart       trt       sex     hours rehab
1     0 0.461845  1.949033 0.340346     1
2    13 0.767064  0.354377 0.340346     1
3     0 1.455039 -0.646735 0.340346     1
4    10 0.470696 -1.571799 0.340346     1
5    12 0.944963  1.363045 0.340346     1
6     0 0.500870  0.591512 -0.617893     2
> str(rehb)
'data.frame':   1000 obs. of  5 variables:
 $ heart: int  0 13 0 10 12 0 4 4 1 4 ...
 $ trt  : num  0.462 -0.767 1.455 0.471 -0.945 ...
 $ sex  : num  1.949 0.354 -0.647 -1.572 1.363 ...
 $ hours: num  0.34 0.34 0.34 0.34 0.34 ...
 $ rehab: int  1 1 1 1 1 2 2 2 2 2 ...
##so can't do gender and/or treatment comparisons
## descriptives and base model below

> fivenum(heart)
[1]   0   0   0   4 225

> length(unique(rehab))
[1] 110
> tapply(heart, rehab, length)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5   5
 38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74
  5   5   5  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10
 75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110
 10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
> fivenum(tapply(heart, rehab, length))
  1  28  55  83 110
  5   5  10  10  15

> tapply(heart, rehab, max)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
 13   4   1  17   2   9   7   4   9   0   5  25  10  36   0  11  47   5  11   0 135   5   1   3   4   2   6   3   4  11   1   2  24   4   2  22  21
 38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74
  5   8  20  12   6   7  13  57   4   9   6   5   3  11   2   7   3  13  10  22   4  16  27  23   4  22   2   8   1  16  11  39  18  18   9  73  13
 75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110
 37  49   7  17  31  26  10   9  19   4  13  17 110  32   1  23   3   4  11  10  39  71  11  10   9 225 222  32  19  16  27  44  27  23  21   4

> fivenum(tapply(heart, rehab, mean))
      10       77       38      107       21
 0.000000 1.000000 2.500000 5.333333 30.200000
> fivenum(tapply(heart, rehab, max))
   10    84    98    57   100
  0.0   4.0  10.5  22.0 225.0

> tapply(heart, rehab, mean)
        1         2         3         4         5         6         7         8         9        10        11        12        13
7.0000000 2.6000000 0.4000000 8.0000000 0.6000000 4.6000000 2.4000000 0.8000000 2.0000000 0.0000000 1.4000000 9.2000000 3.4000000
       14        15        16        17        18        19        20        21        22        23        24        25        26
9.4000000 0.0000000 3.4000000 9.4000000 2.4000000 4.0000000 0.0000000 30.2000000 3.0000000 0.2000000 0.6000000 1.6000000 0.4000000
       27        28        29        30        31        32        33        34        35        36        37        38        39
1.6000000 0.8000000 0.8000000 4.2000000 0.2000000 0.4000000 7.4000000 3.0000000 0.4000000 9.6000000 6.6000000 2.4000000 3.2000000
       40        41        42        43        44        45        46        47        48        49        50        51        52
9.4000000 2.8000000 1.0000000 2.2000000 2.9000000 8.3000000 0.7000000 1.9000000 1.9000000 0.8000000 0.3000000 1.1000000 0.6000000
       53        54        55        56        57        58        59        60        61        62        63        64        65
2.6000000 0.3000000 1.7000000 2.2000000 4.7000000 0.7000000 5.9000000 4.2000000 5.9000000 1.2000000 5.3000000 0.3000000 1.8000000
       66        67        68        69        70        71        72        73        74        75        76        77        78
0.3000000 6.4000000 1.9000000 10.1000000 4.8000000 6.7000000 1.9000000 9.5000000 2.6000000 9.3000000 6.7000000 1.0000000 4.1000000
```

```
      79       80       81       82       83       84       85       86       87       88       89       90       91
8.6000000 4.1000000 1.1000000 3.0000000 2.2000000 0.9000000 3.0000000 1.8000000 28.4000000 4.0000000 0.1000000 4.8000000 0.3333333
      92       93       94       95       96       97       98       99      100      101      102      103      104
0.5333333 1.8666667 4.6000000 3.4666667 6.8000000 3.3333333 1.8666667 1.0666667 17.8666667 19.4000000 8.0666667 1.5333333 2.3333333
     105      106      107      108      109      110
5.0000000 6.8000000 5.3333333 5.9333333 1.8666667 1.0000000

> mean(heart)
[1] 4.156
> var(heart)
[1] 191.8855

> m1 = glmer(heart ~ 1 + (1|rehab), family = poisson, data = rehb)
> summary(m1)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: heart ~ 1 + (1 | rehab)
   Data: rehb
     AIC      BIC   logLik deviance df.resid
 11774.1  11783.9  -5885.0  11770.1      998
Scaled residuals:
   Min     1Q Median     3Q    Max
-5.458 -1.721 -0.937  0.536 49.179

Random effects:
 Groups Name        Variance Std.Dev.
 rehab  (Intercept) 1.247    1.117
Number of obs: 1000, groups:  rehab, 110

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.8408     0.1107   7.596 3.04e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> exp(fixef(m1))
(Intercept)
   2.318311

> exp(confint(m1))
Computing profile confidence intervals ...
                2.5 %   97.5 %
.sig01      2.620921 3.699753
(Intercept) 1.855602 2.876929
>
```

References

Cook, F. E. (1938) *Chocolate cake, I. Optimum baking temperature*. Master's Thesis, Iowa State College.

Cochran, W. G., and Cox, G. M. (1957) *Experimental designs*, 2nd Ed. New York, John Wiley \& Sons.

Lee, Y., Nelder, J. A., and Pawitan, Y. (2006) *Generalized linear models with random effects. Unified analysis via H-likelihood*. Boca Raton, Chapman and Hall/CRC.

Examples

```
str(cake)
## 'temp' is continuous, 'temperature' an ordered factor with 6 levels

(fm1 <- lmer(angle ~ recipe * temperature + (1|recipe:replicate), cake, REML= FALSE))
(fm2 <- lmer(angle ~ recipe + temperature + (1|recipe:replicate), cake, REML= FALSE))
(fm3 <- lmer(angle ~ recipe + temp        + (1|recipe:replicate), cake, REML= FALSE))

## and now "choose" :
anova(fm3, fm2, fm1)
```

---

cbpp *Contagious bovine pleuropneumonia*

Description

Contagious bovine pleuropneumonia (CBPP) is a major disease of cattle in Africa, caused by a mycoplasma. This dataset describes the serological incidence of CBPP in zebu cattle during a follow-up survey implemented in 15 commercial herds located in the Boji district of Ethiopia. The goal of the survey was to study the within-herd spread of CBPP in newly infected herds. Blood samples were quarterly collected from all animals of these herds to determine their CBPP status. These data were used to compute the serological incidence of CBPP (new cases occurring during a given time period). Some data are missing (lost to follow-up).

Format

A data frame with 56 observations on the following 4 variables.

`herd` A factor identifying the herd (1 to 15).

`incidence` The number of new serological cases for a given herd and time period.

`size` A numeric vector describing herd size at the beginning of a given time period.

`period` A factor with levels `1` to 4.

Details

Serological status was determined using a competitive enzyme-linked immuno-sorbent assay (cELISA).

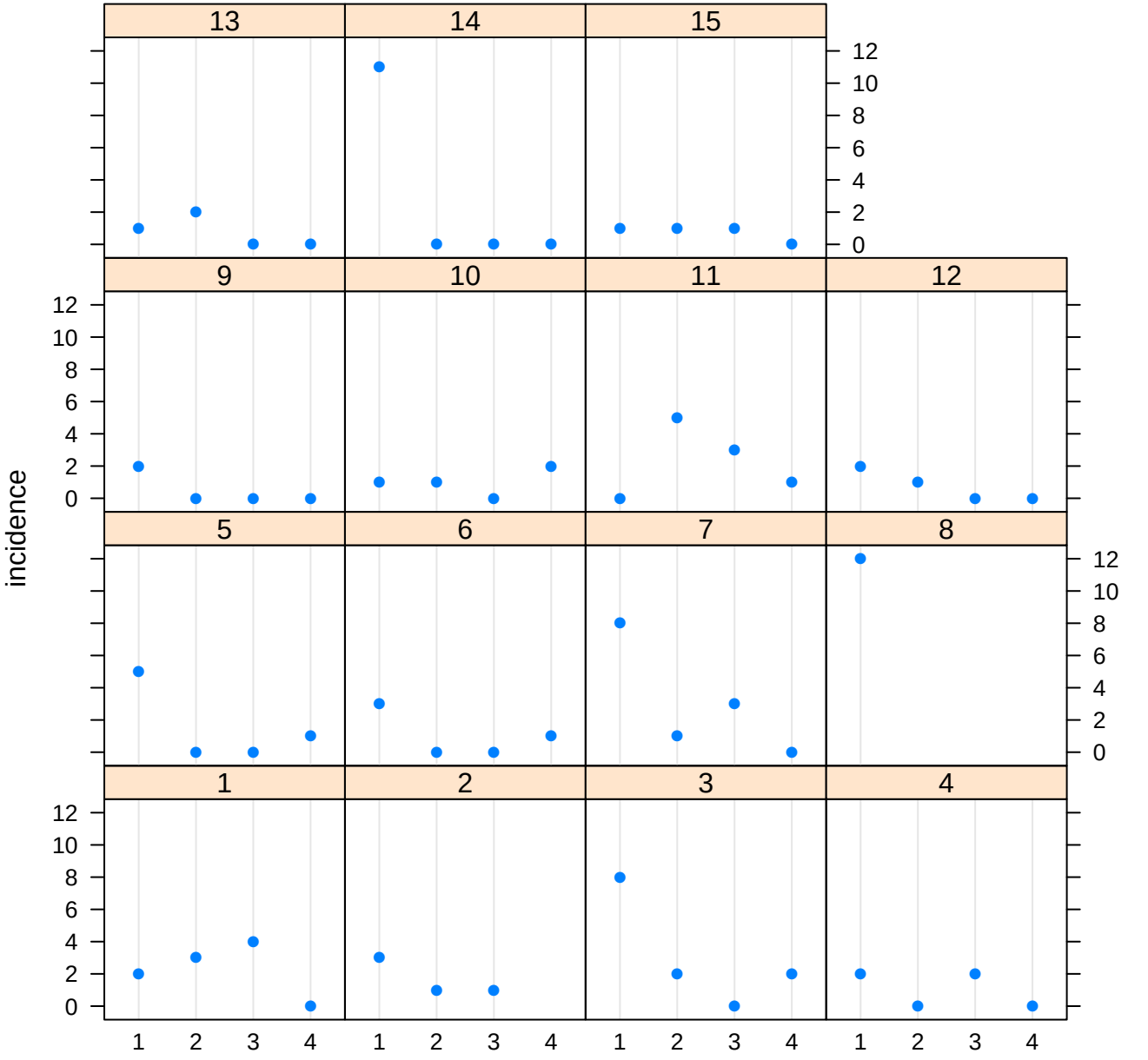
Source

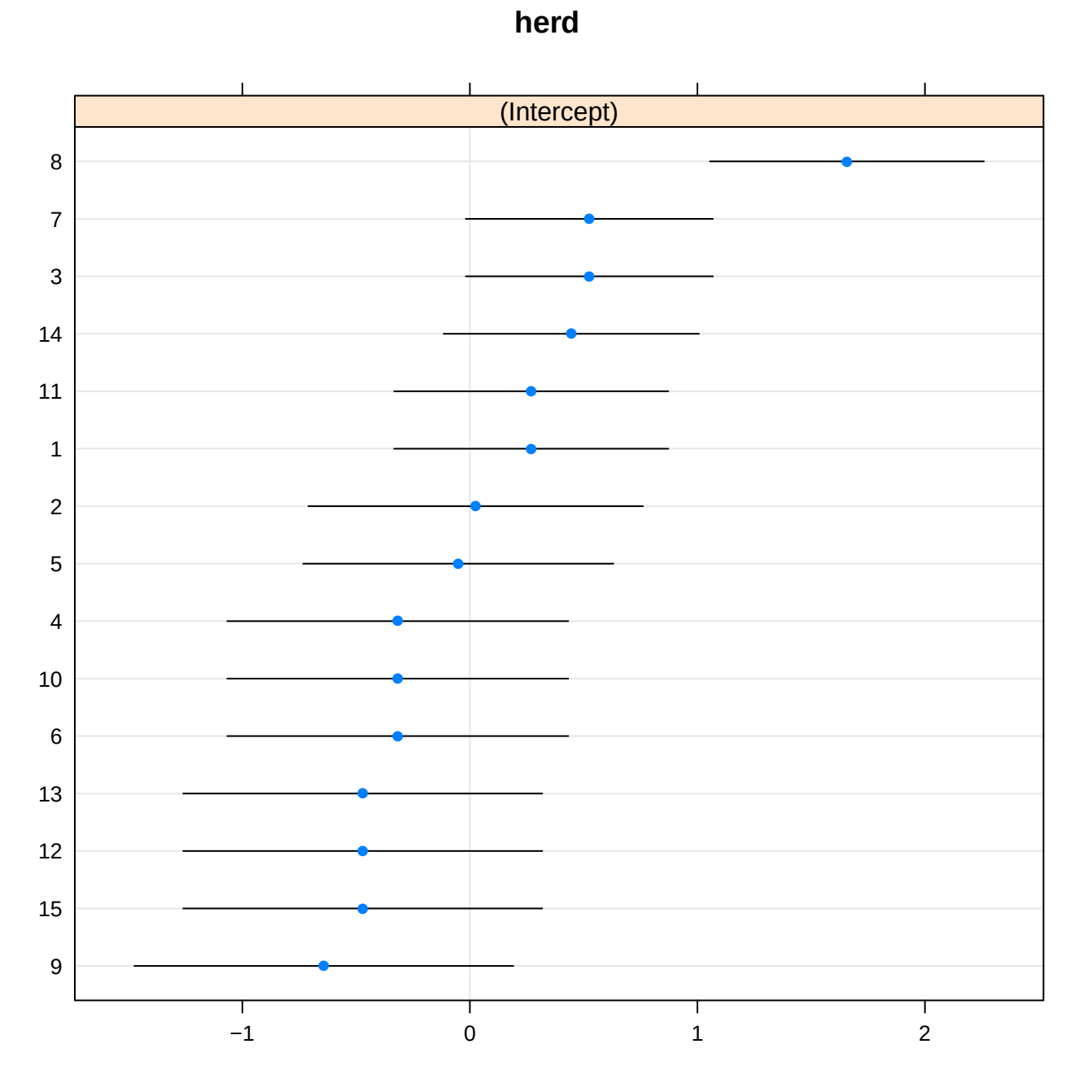
Lesnoff, M., Laval, G., Bonnet, P., Abdicho, S., Workalemahu, A., Kifle, D., Peyraud, A., Lancelot, R., Thiaucourt, F. (2004) Within-herd spread of contagious bovine pleuropneumonia in Ethiopian highlands. *Preventive Veterinary Medicine* **64**, 27–40.

Examples

```
## response as a matrix
(m1 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd),
             family = binomial, data = cbpp))
## response as a vector of probabilities and usage of argument "weights"
m1p <- glmer(incidence / size ~ period + (1 | herd), weights = size,
             family = binomial, data = cbpp)
## Confirm that these are equivalent:
stopifnot(all.equal(fixef(m1), fixef(m1p), tolerance = 1e-5),
          all.equal(ranef(m1), ranef(m1p), tolerance = 1e-5))


## GLMM with individual-level variability (accounting for overdispersion)
cbpp$obs <- 1:nrow(cbpp)
(m2 <- glmer(cbind(incidence, size - incidence) ~ period + (1 | herd) +  (1|obs),
              family = binomial, data = cbpp))
```

```
> library(lme4)
> library(lattice)

> data(cbpp)
> str(cbpp)
'data.frame':   56 obs. of  4 variables:
 $ herd     : Factor w/ 15 levels "1","2","3","4",..: 1 1 1 1 2 2 2 3 3 3 ...
 $ incidence: num  2 3 4 0 3 1 1 8 2 0 ...
 $ size     : num  14 12 9 5 22 18 21 22 16 16 ...
 $ period   : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 1 2 3 ...
> attach(cbpp)
> length(unique(herd))
[1] 15
> table(incidence,herd)
         herd
incidence 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       0  1 0 1 2 2 2 1 0 3  1  1  2  2  3  1
       1  0 2 0 0 1 1 1 0 0  2  1  1  1  0  3
       2  1 0 2 2 0 0 0 0 1  1  0  1  1  0  0
       3  1 1 0 0 0 1 1 0 0  0  1  0  0  0  0
       4  1 0 0 0 0 0 0 0 0  0  0  0  0  0  0
       5  0 0 0 0 1 0 0 0 0  0  1  0  0  0  0
       8  0 0 1 0 0 0 1 0 0  0  0  0  0  0  0
       11 0 0 0 0 0 0 0 0 0  0  0  0  0  1  0
       12 0 0 0 0 0 0 0 0 1  0  0  0  0  0  0

> dotplot(incidence ~ period|herd, data = cbpp) #attached plot
###############
> m1 = glmer(incidence ~ 1 + (1|herd), family = poisson, data = cbpp)
> summary(m1)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: incidence ~ 1 + (1 | herd)
   Data: cbpp
     AIC      BIC   logLik deviance df.resid
   244.0    248.1   -120.0    240.0       54
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.6484 -1.0518 -0.4310  0.7664  5.3589

Random effects:
 Groups Name        Variance Std.Dev.
 herd   (Intercept) 0.4642   0.6813
Number of obs: 56, groups:  herd, 15

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4746     0.2080   2.281   0.0225 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> exp(fixef(m1))
(Intercept)
   1.607382
> mean(incidence)
[1] 1.767857

> confint(m1, method = "boot", nsim = 1000, boot.type = "perc")
Computing bootstrap confidence intervals ...
                          2.5 %    97.5 %
sd_(Intercept)|herd 0.24700100 1.0115845
(Intercept)         0.06200498 0.9067111

> exp(confint(m1, method = "boot", nsim = 1000, boot.type = "perc"))
Computing bootstrap confidence intervals ...
                        2.5 %   97.5 %
sd_(Intercept)|herd 1.297958 2.779228
(Intercept)         1.025472 2.388858

> dotplot(ranef(m1, postVar = TRUE))
$herd
Warning message:
In ranef.merMod(m1, postVar = TRUE) :
  ‘postVar’ is deprecated: please use ‘condVar’ instead
```


13
14
15
9
10
11
12
5
6
7
8
1
2
3
4
incidence
0
2
4
6
8
10
12
1
2
3
4


herd
(Intercept)
8
7
3
14
11
1
2
5
4
10
6
13
12
15
9
−1
0
1
2

```
> dotplot(ranef(m1, condVar = TRUE))
$herd

> ranef(m1)
$herd
   (Intercept)
1   0.26961543
2   0.02527144
3   0.52501789
4  -0.31699533
5  -0.05099810
6  -0.31699533
7   0.52501789
8   1.65721021
9  -0.64209460
10 -0.31699533
11  0.26961543
12 -0.47097791
13 -0.47097791
14  0.44567263
15 -0.47097791

#overdispersion??
# due to period?
> m2 = glmer(incidence ~ period + (1|herd), family = poisson, data = cbpp)
> anova(m1,m2)
Data: cbpp
Models:
m1: incidence ~ 1 + (1 | herd)
m2: incidence ~ period + (1 | herd)
   Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
m1  2 244.04 248.09 -120.020   240.04
m2  5 203.38 213.51  -96.692   193.38 46.657      3  4.112e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(m2)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: incidence ~ period + (1 | herd)
   Data: cbpp

     AIC      BIC   logLik deviance df.resid
   203.4    213.5    -96.7    193.4       51

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.1468 -0.8733 -0.4164  0.8251  2.8642

Random effects:
 Groups Name        Variance Std.Dev.
 herd   (Intercept) 0.2502   0.5002
Number of obs: 56, groups:  herd, 15

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2768     0.1908   6.693 2.19e-11 ***
period2      -1.1249     0.2748  -4.094 4.24e-05 ***
period3      -1.3191     0.2963  -4.453 8.49e-06 ***
period4      -1.9450     0.3971  -4.898 9.66e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
        (Intr) period2 period3
period2 -0.344
period3 -0.319  0.225
period4 -0.236  0.167  0.155

###################################
> m3 = glmer(incidence ~ period + size + (1|herd), family = poisson, data = cbpp)
> summary(m3)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
```

```
Formula: incidence ~ period + size + (1 | herd)
   Data: cbpp

     AIC      BIC   logLik deviance df.resid
   195.0    207.2    -91.5    183.0       50

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2004 -0.8864 -0.2497  0.5065  2.9627

Random effects:
 Groups Name        Variance Std.Dev.
 herd   (Intercept) 0.1932   0.4395
Number of obs: 56, groups:  herd, 15

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.02360    0.47010   0.050 0.959956
period2     -0.91632    0.28643  -3.199 0.001379 **
period3     -1.02422    0.31547  -3.247 0.001168 **
period4     -1.55485    0.41731  -3.726 0.000195 ***
size         0.06463    0.02099   3.079 0.002080 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
        (Intr) period2 period3 period4
period2 -0.360
period3 -0.421  0.293
period4 -0.359  0.230  0.236
size    -0.923  0.237  0.318  0.284
> exp(fixef(m3))
(Intercept)     period2     period3     period4        size
  1.0238837   0.3999895   0.3590754   0.2112203   1.0667594 > anova(m2,m3)

anova(m2, m3)
Data: cbpp
Models:
m2: incidence ~ period + (1 | herd)
m3: incidence ~ period + size + (1 | herd)
   Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
m2  5 203.38 213.51 -96.692   193.38
m3  6 195.04 207.19 -91.520   183.04 10.345      1   0.001298 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


##############################
> cbpp$obs = 1:nrow(cbpp)
> head(cbpp)
  herd incidence size period obs
1    1         2   14      1   1
2    1         3   12      2   2
3    1         4    9      3   3
4    1         0    5      4   4
5    2         3   22      1   5
6    2         1   18      2   6

> m3o = glmer(incidence ~ period + size + (1|herd) + (1|obs), family = poisson, data = cbpp)
> anova(m3,m3o)
Data: cbpp
Models:
m3: incidence ~ period + size + (1 | herd)
m3o: incidence ~ period + size + (1 | herd) + (1 | obs)
    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
m3   6 195.04 207.19 -91.520   183.04
m3o  7 190.06 204.24 -88.031   176.06 6.9767      1   0.008258 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> exp(fixef(m3o))
(Intercept)     period2     period3     period4        size
  1.1572410   0.3503044   0.3178136   0.1790956   1.0542937
```

Mixed-effects models

Remko Duursma[1], *Jeff Powell*[1]
[1]Hawkesbury Institute for the Environment
Western Sydney University

WESTERN SYDNEY
UNIVERSITY

Hawkesbury Institute
for the Environment

September 14, 2016

1.10 Exercises

In these exercises, we use the following colour codes:

- ■ **Easy**: make sure you complete some of these before moving on. These exercises will follow examples in the text very closely.
- ⧫ **Intermediate**: a bit harder. You will often have to combine functions to solve the exercise in two steps.
- ▲ **Hard**: difficult exercises! These exercises will require multiple steps, and significant departure from examples in the text.

We suggest you complete these exercises in an **R** markdown file. This will allow you to combine code chunks, graphical output, and written answers in a single, easy-to-read file.

1.10.1 PREF Canopy data

1. ⧫ In the analysis of the `pref` data, use model selection (`AIC`, `anova`) to evaluate the importance of `species` and `dfromtop`.

1.10.2 Litter decomposition data

1. ⧫ The `litter` data contain a factor (`variety`) describing whether the litter is derived from a genetically modified (`gm`) or conventional (`nongm`) soy variety. Plot the data to observe the effect of `variety`. Use `lmer` to test the effect of `variety`, in addition to the other significant variables, on litter decomposition.

1.10.3 EucFACE ground cover data

The file `eucfaceGC.csv` contains estimates of plant and litter cover within the rings of the EucFACE experiment, evaluating forest ecosystem responses to elevated CO_2, on two dates; the data description can be found in Section A.20 (p. 271).

1. ⧫ Convert the variables indicating the nested sampling design to factors, then use `glmer` in `lme4` to test for an interaction between `Trt` and `Date` on `Grass` and `Litter` cover. Grass cover represents a frequency across a maximum of 16 points within a quadrat (use the `binomial` family), while litter cover represents counts (use the `poisson` family).
2. ▲ Following on from exercise 3, generate subsets to determine the sources of the interactions (i.e., does the treatment effect differ between the two dates or does the date effect differ between the two treatments?).

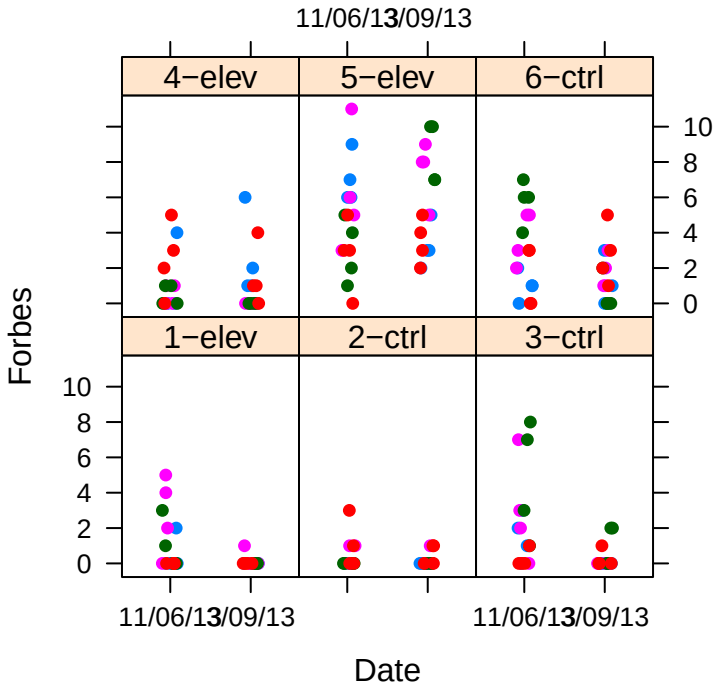


Figure 1.15: Counts of forbs within plant communities exposed to ambient or elevated carbon dioxide concentrations at two dates. Points represent estimates within subplots, subplots common to the same plot within each ring have a common colour.

the rings of the EucFACE experiment, evaluating forest ecosystem responses to elevated CO_2, on two dates. There are six rings (`Ring`), three for each treatment (`Trt`; ambient and elevated CO_2). Within each ring are four plots (`Plot`) and within each plot are four 1m by 1m subplots (`Sub`). Here we will test for an interaction between `Trt` and `Date` on ground cover measurements of `Forbes` (these are count data).

The following code produces Fig. 1.15, showing a plot of the raw groundcover data.

```
# read data and convert random effects to factors
eucface <- read.csv("eucfaceGC.csv")
eucface$Ring <- as.factor(paste(eucface$Ring, eucface$Trt, sep='-'))
eucface$Plot <- as.factor(eucface$Plot)
eucface$Sub <- as.factor(eucface$Sub)

# load packages
library(lme4)
library(lattice)

# A quick plot to visualize ground cover by Date and Ring.
# Colours represent 'Plot'.
xyplot(Forbes~Date|Ring,groups=Plot,data=eucface,pch=16,jitter.x=T)
```

Since the data are count data, it is usually appropriate to use the Poisson distribution. The following code fits a `glmer` with the `poisson` error family, and produces diagnostic plots in Fig. 1.16.

To usual way to decide on the appropriate family is to inspect the diagnostic plots, especially a plot of residuals versus the fitted values. In the following, we fit the model three times, first with normal

```
stat196, week3, Poisson outcome factorial design
HIE Syndney

> ?read.csv
starting httpd help server ... done
> ?read.table
> forest = read.table("https://www.westernsydney.edu.au/__data/assets/text_file/0011/917237/eucfaceGC.csv", header = TRUE, sep = ",")
> str(forest)
'data.frame':   192 obs. of  8 variables:
 $ Date  : Factor w/ 2 levels "11/06/13","3/09/13": 1 1 1 1 1 1 1 1 1 1 ...
 $ Ring  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Plot  : int  1 1 1 1 2 2 2 2 3 3 ...
 $ Sub   : int  1 2 3 4 1 2 3 4 1 2 ...
 $ Forbes: int  0 2 0 0 2 5 4 0 1 3 ...
 $ Grass : int  16 14 16 16 15 16 16 16 16 13 ...
 $ Litter: int  5 6 2 2 3 3 4 3 0 4 ...
 $ Trt   : Factor w/ 2 levels "ctrl","elev": 2 2 2 2 2 2 2 2 2 2 ...
> dim(forest)
[1] 192   8
> stem(forest$Forbes)

  The decimal point is at the |

   0 | 00000000000000000000000000000000000000000000000000000000000000000000+14
   1 | 00000000000000000000000000000
   2 | 0000000000000000
   3 | 000000000000000000
   4 | 000000
   5 | 00000000000
   6 | 000000
   7 | 000000
   8 | 000
   9 | 00
  10 | 00
  11 | 0

> table(forest$Forbes)

 0  1  2  3  4  5  6  7  8  9 10 11
94 27 16 18  6 11  6  6  3  2  2  1
> fivenum(forest$Forbes)
[1]  0  0  1  3 11

> forb.pois <- glmer(Forbes~Date*Trt+(1|as.factor(Ring)),family=poisson, data=forest)
Error: couldn't evaluate grouping factor as.factor(Ring) within model frame: try adding grouping factor to data frame explicitly if po
> forest$Ring <- as.factor(paste(forest$Ring, forest$Trt, sep='-'))

> forb.pois <- glmer(Forbes~Date*Trt+(1|Ring),family=poisson, data=forest)
> summary(forb.pois)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: Forbes ~ Date * Trt + (1 | Ring)
```

```
   Data: forest

     AIC      BIC   logLik deviance df.resid
   659.5    675.8   -324.7    649.5      187

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2892 -0.8289 -0.6934  0.7290  5.6446

Random effects:
 Groups Name        Variance Std.Dev.
 Ring   (Intercept) 0.7981   0.8934
Number of obs: 192, groups:  Ring, 6

Fixed effects:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.36114    0.53258   0.678    0.498
Date3/09/13         -1.05605    0.20412  -5.174  2.3e-07 ***
Trtelev              0.03945    0.75050   0.053    0.958
Date3/09/13:Trtelev  1.02866    0.24458   4.206  2.6e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) Dt3/09/13 Trtelv
Date3/09/13 -0.099
Trtelev     -0.709  0.070
Dt3/09/13:T  0.083 -0.835    -0.107
> exp(fixef(forb.pois))
        (Intercept)         Date3/09/13             Trtelev Date3/09/13:Trtelev
          1.4349572           0.3478259           1.0402348           2.7973069
> exp(confint(forb.pois))
Computing profile confidence intervals ...
                        2.5 %    97.5 %
.sig01              1.7010845 6.2396119
(Intercept)         0.4091106 4.8319798
Date3/09/13         0.2292726 0.5140375
Trtelev             0.1838199 5.9957720
Date3/09/13:Trtelev 1.7420442 4.5743513

# The HIE guide does a more complex (within Ring) replication.
> forest$Plot <- as.factor(forest$Plot)
> forest$Sub <- as.factor(forest$Sub)
> forb.pois.nest <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),family=poisson, data=forest)
> summary(forb.pois.nest)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: Forbes ~ Date * Trt + (1 | Ring/Plot/Sub)
   Data: forest

     AIC      BIC   logLik deviance df.resid
   614.7    637.5   -300.3    600.7      185
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.9409 -0.6627 -0.4394  0.4823  3.6061

Random effects:
 Groups          Name        Variance Std.Dev.
 Sub:(Plot:Ring) (Intercept) 0.1437   0.3790
 Plot:Ring       (Intercept) 0.5900   0.7681
 Ring            (Intercept) 0.8504   0.9222
Number of obs: 192, groups:  Sub:(Plot:Ring), 96; Plot:Ring, 24; Ring, 6

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.08894    0.59923   0.148    0.882
Date3/09/13        -1.05606    0.19836  -5.324 1.02e-07 ***
Trtelev             0.03243    0.84154   0.039    0.969
Date3/09/13:Trtelev  1.02866    0.23830   4.317 1.58e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) Dt3/09/13 Trtelv
Date3/09/13 -0.085
Trtelev     -0.705  0.061
Dt3/09/13:T  0.071 -0.832    -0.094
> exp(fixef(forb.pois.nest))
        (Intercept)         Date3/09/13             Trtelev Date3/09/13:Trtelev
          1.0930189           0.3478244           1.0329578           2.7973041
> exp(confint(forb.pois.nest))
Computing profile confidence intervals ...
                        2.5 %    97.5 %
.sig01              1.1187957 1.8936468
.sig02              1.5465222 3.6756420
.sig03              1.5221009 7.5639339
(Intercept)         0.2624446 4.2286415
Date3/09/13         0.2292838 0.5140144
Trtelev             0.1461481 7.4206146
Date3/09/13:Trtelev 1.7420895 4.5742068
> # about same result
```

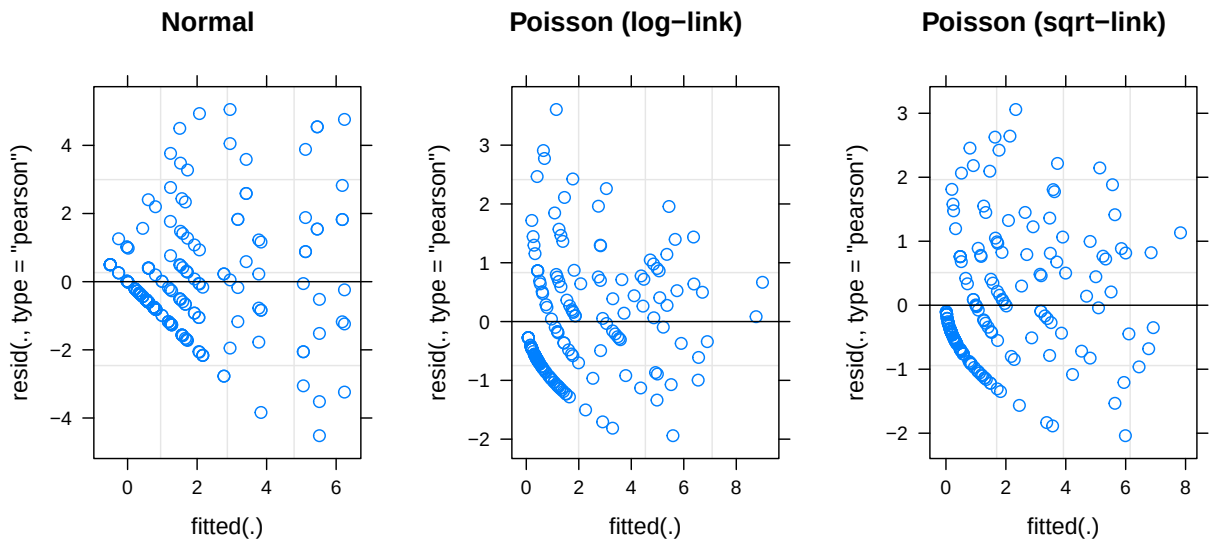


Figure 1.16: Diagnostic plots for three model fits of the EucFACE groundcover data.

errors (using `lmer`), then with Poisson errors (the distribution we expect to be appropriate), and a Poisson distribution with square-root link function.

The following code produces three diagnostic plots (Fig. 1.16).

```
forb.norm <- lmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                  data=eucface)

forb.pois <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                   family=poisson, data=eucface)

forb.pois.sqrt <- glmer(Forbes~Date*Trt+(1|Ring/Plot/Sub),
                        family=poisson('sqrt'), data=eucface)

# The standard plot() on an (g)lmer object makes a 'grid-based'
# plot. To arrange plots side-by-side we have to use gridExtra, like so.
library(gridExtra)
p1 <- plot(forb.norm, main="Normal")
p2 <- plot(forb.pois, main="Poisson (log-link)")
p3 <- plot(forb.pois.sqrt, main="Poisson (sqrt-link)")
grid.arrange(p1,p2,p3,ncol=3)
```

Note that the syntax of `glmer` is identical to that of `lmer`, with the exception of the `family` argument.

Inspecting Fig. 1.16, the residuals look better for the models fit with the Poisson family compared to the gaussian error (as assumed by `lmer`). The residuals improve a bit more when using the `sqrt` link function in the Poisson family.

Note that the fit of the model may be improved further by using one of the other plant cover variables as a covariate or by incorporating other data from the site, but for the purpose of this chapter the fit is good enough.

Something else to consider when it comes to poisson errors is that overdispersion can result in the underestimation of error terms for the model coefficients. Overdispersion occurs when you have a large number of zeros in the data and/or an important predictor is not accounted for. Testing hypotheses from models where overdispersion is evident is dangerous as the probability of Type I error

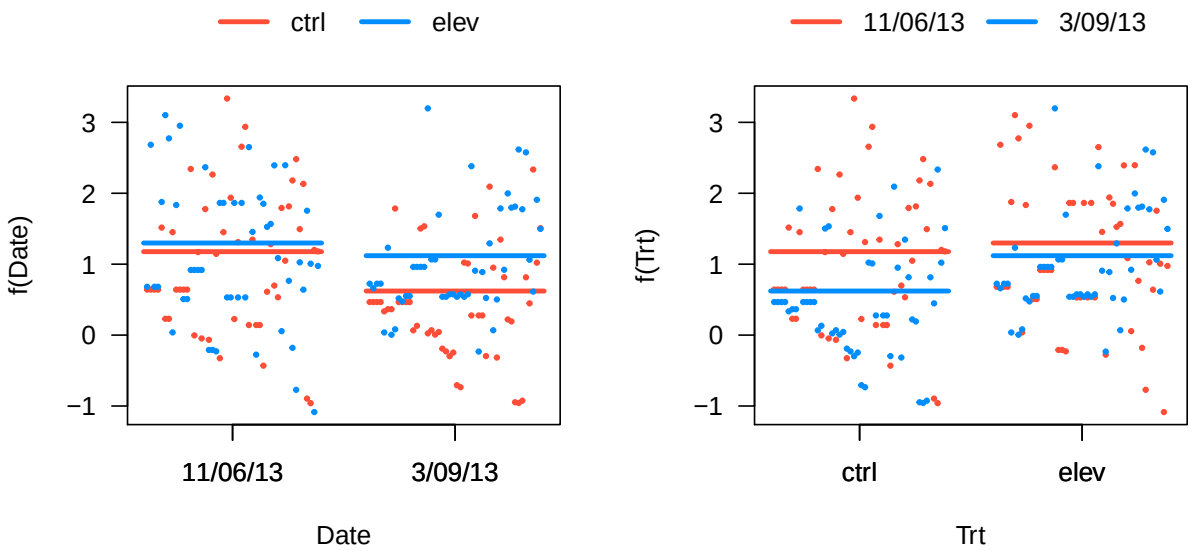


Figure 1.17: Model predictions, looking at treatment effects by date (left) and date effects by treatment (right).

significant we must dig deeper to understand the source of the interaction. Looking at Fig. 1.17, it appears that :

1. Forbs decreased in frequency between the two dates in the control but not in the elevated CO_2 treatment
2. Forb abundance was lower in the control than in the elevated CO_2 treatment on the second date, but not the first date.

```
# set up graphics window for two plots
par(mfrow=c(1, 2))

# plot model preditions and data
visreg(forb.pois.ind, 'Date', 'Trt', overlay=TRUE)
visreg(forb.pois.ind, 'Trt', 'Date', overlay=TRUE)
```

We can evaluate each of these hypotheses through model selection after combining treatment levels.

```
# create a three level factor that combines both dates in the 'elev' treatment
eucface$trtcomb.elev <- with(eucface, paste(Trt, Date, sep='-'))
eucface$trtcomb.elev[eucface$Trt == 'elev'] <- 'elev'
eucface$trtcomb.elev <- as.factor(eucface$trtcomb.elev)
levels(eucface$trtcomb.elev)
```

```
## [1] "ctrl-11/06/13" "ctrl-3/09/13"  "elev"
```

```
# create a three level factor that combines both treatments for the '11/06/13' sampling
eucface$trtcomb.date <- with(eucface, paste(Trt, Date, sep='-'))
eucface$trtcomb.date[eucface$Date == '11/06/13'] <- '11/06/13'
eucface$trtcomb.date <- as.factor(eucface$trtcomb.date)
levels(eucface$trtcomb.date)
```

```
## [1] "11/06/13"     "ctrl-3/09/13" "elev-3/09/13"
```

Mmmec *Malignant melanoma deaths in Europe*

Description

Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.

Usage

```
data(Mmmec)
```

Format

A data frame with 354 observations on the following 6 variables.

nation a factor with levels `Belgium`, `W.Germany`, `Denmark`, `France`, `UK`, `Italy`, `Ireland`, `Luxembourg`, and `Netherlands`

region Region ID - a factor.

county County ID - a factor.

deaths Number of male deaths due to MM during 1971–1980

expected Number of expected deaths.

uvb Centered measure of the UVB dose reaching the earth's surface in each county.

Source

http://multilevel.ioe.ac.uk/softrev/mmmec.html

References

Langford, I.H., Bentham, G. and McDonald, A. 1998: Multilevel modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41-58.

Examples

```
str(Mmmec)
summary(Mmmec)
(fm1 <- glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)))
```

<- Back to Examples from Multilevel Modelling Software Review (/cran/mlmRev/)

Malignant melanoma deaths in Europe

Description

Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.

Usage

```
data(Mmmec)
```

Format

A data frame with 354 observations on the following 6 variables.

nation
a factor with levels `Belgium`, `W.Germany`, `Denmark`, `France`, `UK`, `Italy`, `Ireland`, `Luxembourg`, and `Netherlands`

region
Region ID - a factor.

county
County ID - a factor.

deaths
Number of male deaths due to MM during 1971–1980

expected
Number of expected deaths.

uvb
Centered measure of the UVB dose reaching the earth's surface in each county.

Source

http://multilevel.ioe.ac.uk/softrev/mmmec.html (http://multilevel.ioe.ac.uk/softrev/mmmec.html)

References

```
##### Week 3    Malignant melanoma deaths in Europe
R version 3.2.2 (2015-08-14) -- "Fire Safety"
> library(lme4)
> library(mlmRev)
> data(Mmmec)
> attach(Mmmec)
> head(Mmmec)
   nation region county deaths expected     uvb
1 Belgium      1      1     79  51.2220 -2.9057
2 Belgium      2      2     80  79.9560 -3.2075
3 Belgium      2      3     51  46.5169 -2.8038
4 Belgium      2      4     43  55.0530 -3.0069
5 Belgium      2      5     89  67.7580 -3.0069
6 Belgium      2      6     19  35.9760 -3.4175
> dim(Mmmec)
[1] 354   6
> str(Mmmec)
'data.frame':   354 obs. of  6 variables:
 $ nation  : Factor w/ 9 levels "Belgium","W.Germany",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 78 levels "1","2","3","4",..: 1 2 2 2 2 2 3 3 3 3 ...
 $ county  : Factor w/ 354 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ deaths  : int  79 80 51 43 89 19 19 15 33 9 ...
 $ expected: num  51.2 80 46.5 55.1 67.8 ...
 $ uvb     : num  -2.91 -3.21 -2.8 -3.01 -3.01 ...

> write.table(Mmmec, file = "D:\\drr16\\stat196\\week3\\eudata", quote = FALSE) # so I can look at it
> # counties unique numb (1,...355), regions unique over countries

> # 78 regions (#26NA), 354 counties (individ obs)
>  setwd("D:\\drr16\\stat196\\week3\\")
> table(region)
region
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32 33
 1  5  5  4  7  1  1  1  2  4  5  3  1  1  3  3  8  2  5  4  3  4  4  6  4  4  2  8  5  3  4  8
34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
 2  5  3  4  6  8  3  5  5  4 13  7  5  4  8 12  4  4  2  3  5  8  4  5  4  9  4  2  6  5  4  9
66 67 68 69 70 71 72 73 74 75 76 77 78 79
 9  2  2  1  7  5 12  6  3  3  3  2  4  2
> fivenum(table(region))
region
 1 16 47 71 44
 1  3  4  5 13
> #so 7 regions have only 1 county
> sort(table(region))
region
 1  6  7  8 13 14 69  9 18 28 34 52 61 67 68 77 79 12 15 16 21 31 36 40 53 74 75 76  4 10 20 22
 1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  3  3  3  4  4  4  4
23 25 27 32 37 43 47 50 51 56 58 60 64 78  2  3 11 19 30 35 41 42 46 54 57 63 71 24 38 62 73  5
 4  4  4  4  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  5  5  5  5  5  5  5  6  6  6  6  7
```

```
45 70 17 29 33 39 48 55 59 65 66 49 72 44
 7  7  8  8  8  8  8  8  9  9  9 12 12 13

> # two level analysis: counties within region (ignoring country for now)
> fm1 = glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "expected"
> summary(fm1)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 | region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2256.4   2268.0  -1125.2   2250.4      351

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6593 -0.7340 -0.0528  0.6122  4.2276

Random effects:
 Groups Name        Variance Std.Dev.
 region (Intercept) 0.1697   0.4119
Number of obs: 354, groups:  region, 78

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.13859    0.04934  -2.809 0.004971 **
uvb         -0.03443    0.01002  -3.435 0.000593 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.047

> exp(fixef(fm1)) # so here mult effect of uvb is to reduce mel?
(Intercept)         uvb
  0.8705813   0.9661564

> exp(confint(fm1))
Computing profile confidence intervals ...
                2.5 %    97.5 %
.sig01      1.4146433 1.6428082
(Intercept) 0.7888942 0.9597464
uvb         0.9477699 0.9861602

# base (unconditional) model without uvb
> fm0 = glmer(deaths ~ 1 + (1|region), Mmmec, poisson, offset = log(expected))
> summary(fm0)
```

```
> fivenum(deaths - expected)
[1] -64.379  -7.300  -1.647   3.488 130.103
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ 1 + (1 | region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2264.5   2272.3  -1130.3   2260.5      352

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6257 -0.7233 -0.0542  0.6024  4.2657

Random effects:
 Groups Name        Variance Std.Dev.
 region (Intercept) 0.22     0.4691
Number of obs: 354, groups:  region, 78

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.13280    0.05552  -2.392   0.0168 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> anova(fm0, fm1) # compare models (test for uvb)
Data: Mmmec
Models:
fm0: deaths ~ 1 + (1 | region)
fm1: deaths ~ uvb + (1 | region)
    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
fm0  2 2264.5 2272.2 -1130.3   2260.5
fm1  3 2256.4 2268.0 -1125.2   2250.4 10.113      1   0.001473 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # so uvb helps improve fit

> library(lattice)
> # random effects far from normal
> qqmath(ranef(fm0), condVar = TRUE)
$region
> qqmath(ranef(fm1), condVar = TRUE)
$region
######################################################################
>

> fm1a = glmer(deaths ~ uvb + (1|region), Mmmec, poisson) # omit offset, now uvb has positive effect on mel
> summary(fm1a)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
```

```
Formula: deaths ~ uvb + (1 | region)
   Data: Mmmec

     AIC      BIC   logLik deviance df.resid
  5722.4   5734.0  -2858.2   5716.4      351

Scaled residuals:
   Min     1Q Median     3Q    Max
-7.077 -1.779 -0.448  0.764 32.198

Random effects:
 Groups Name        Variance Std.Dev.
 region (Intercept) 1.177    1.085
Number of obs: 354, groups:  region, 78

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.00212    0.12399  24.212   <2e-16 ***
uvb          0.03470    0.01859   1.866    0.062 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb 0.016

> exp(fixef(fm1a))
(Intercept)         uvb
  20.128232    1.035305
```

Mmmec *Malignant melanoma deaths in Europe*

Description

Malignant Melanoma Mortality in the European Community associated with the impact of UV radiation exposure.

Usage

```
data(Mmmec)
```

Format

A data frame with 354 observations on the following 6 variables.

nation a factor with levels `Belgium`, `W.Germany`, `Denmark`, `France`, `UK`, `Italy`, `Ireland`, `Luxembourg`, and `Netherlands`

region Region ID - a factor.

county County ID - a factor.

deaths Number of male deaths due to MM during 1971–1980

expected Number of expected deaths.

uvb Centered measure of the UVB dose reaching the earth's surface in each county.

Source

http://multilevel.ioe.ac.uk/softrev/mmmec.html

References

Langford, I.H., Bentham, G. and McDonald, A. 1998: Multilevel modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41-58.

Examples

```
str(Mmmec)
summary(Mmmec)
(fm1 <- glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)))
```

```
R version 3.3.3 (2017-03-06) -- "Another Canoe"
  ######  more two-level analyses, melanoma data
> library(lme4)
Loading required package: Matrix
>  library(mlmRev)

>  data(Mmmec)
>  head(Mmmec)
   nation region county deaths expected     uvb
1 Belgium      1      1     79  51.2220 -2.9057
2 Belgium      2      2     80  79.9560 -3.2075
3 Belgium      2      3     51  46.5169 -2.8038
4 Belgium      2      4     43  55.0530 -3.0069
5 Belgium      2      5     89  67.7580 -3.0069
6 Belgium      2      6     19  35.9760 -3.4175
>  str(Mmmec)
'data.frame':   354 obs. of  6 variables:
 $ nation  : Factor w/ 9 levels "Belgium","W.Germany",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ region  : Factor w/ 78 levels "1","2","3","4",..: 1 2 2 2 2 2 3 3 3 3 ...
 $ county  : Factor w/ 354 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ deaths  : int  79 80 51 43 89 19 19 15 33 9 ...
 $ expected: num  51.2 80 46.5 55.1 67.8 ...
 $ uvb     : num  -2.91 -3.21 -2.8 -3.01 -3.01 ...
>  # two level analysis: counties within region (ignoring country for now)
>  fm1 = glmer(deaths ~ uvb + (1|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "expect

## fm1 in week 3; try in fm1a allowing death uvb gradient to vary by region
>  fm1a = glmer(deaths ~ uvb + (1+ uvb|region), Mmmec, poisson, offset = log(expected)) # multiplicative model with base "
> summary(fm1a)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: poisson  ( log )
Formula: deaths ~ uvb + (1 + uvb | region)
   Data: Mmmec
 Offset: log(expected)

     AIC      BIC   logLik deviance df.resid
  2257.4   2276.7  -1123.7   2247.4      349

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.6536 -0.7579 -0.0382  0.6025  4.2340

Random effects:
 Groups Name        Variance  Std.Dev. Corr
```

```
 region (Intercept) 0.1558795 0.39482
        uvb         0.0002171 0.01473  -1.00
Number of obs: 354, groups:  region, 78

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.149304   0.048105  -3.104  0.00191 **
uvb         -0.037124   0.008658  -4.288 1.81e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
    (Intr)
uvb -0.253

> anova(fm1,fm1a)
Data: Mmmec
Models:
fm1: deaths ~ uvb + (1 | region)
fm1a: deaths ~ uvb + (1 + uvb | region)
     Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
fm1   3 2256.4 2268.0 -1125.2   2250.4
fm1a  5 2257.4 2276.7 -1123.7   2247.4 3.026      2     0.2202
```

glmer.nb {lme4} R Documentation

Fitting Negative Binomial GLMMs

Description

Fits a generalized linear mixed-effects model (GLMM) for the negative binomial family, building on `glmer`, and initializing via `theta.ml` from **MASS**.

Usage

```
glmer.nb(..., interval = log(th) + c(-3, 3),
         tol = 5e-5, verbose = FALSE, nb.control = NULL,
         initCtrl = list(limit = 20, eps = 2*tol, trace = verbose))
```

Arguments

`...`	arguments as for `glmer(.)` such as `formula`, `data`, `control`, etc, but *not* `family`!
`interval`	interval in which to start the optimization. The default is symmetric on log scale around the initially estimated theta.
`tol`	tolerance for the optimization via `optimize`.
`verbose`	`logical` indicating how much progress information should be printed during the optimization. Use `verbose = 2` (or larger) to enable `verbose=TRUE` in the `glmer()` calls.
`nb.control`	optional `list`, like `glmerControl()`, used in `refit(*, control = control.nb)` during the optimization.
`initCtrl`	(***experimental***, *do not rely on this*:) a `list` with named components as in the default, passed to `theta.ml` (package **MASS**) for the initial value of the negative binomial parameter `theta`.

Value

An object of class `glmerMod`, for which many methods are available (e.g. `methods(class="glmerMod")`), see `glmer`.

Note

For historical reasons, the shape parameter of the negative binomial and the random effects parameters in our (G)LMM models are both called `theta` (θ), but are unrelated here.

The negative binomial θ can be extracted from a fit `g <- glmer.nb()` by `getME(g, "glmer.nb.theta")`.

Parts of `glmer.nb()` are still experimental and methods are still missing or suboptimal. In particular, there is no inference available for the dispersion parameter θ, yet.

See Also

`glmer`; from package **MASS**, `negative.binomial` (which we re-export currently) and `theta.ml`, the latter for initialization of optimization.

The 'Details' of `pnbinom` for the definition of the negative binomial distribution.

Examples

```
set.seed(101)
dd <- expand.grid(f1 = factor(1:3),
                  f2 = LETTERS[1:2], g=1:9, rep=1:15,
          KEEP.OUT.ATTRS=FALSE)
summary(mu <- 5*(-4 + with(dd, as.integer(f1) + 4*as.numeric(f2))))
dd$y <- rnbinom(nrow(dd), mu = mu, size = 0.5)
str(dd)
require("MASS")## and use its glm.nb() - as indeed we have zero random effect:
## Not run:
m.glm <- glm.nb(y ~ f1*f2, data=dd, trace=TRUE)
summary(m.glm)
m.nb <- glmer.nb(y ~ f1*f2 + (1|g), data=dd, verbose=TRUE)
m.nb
## The neg.binomial theta parameter:
getME(m.nb, "glmer.nb.theta")
LL <- logLik(m.nb)
## mixed model has 1 additional parameter (RE variance)
stopifnot(attr(LL,"df")==attr(logLik(m.glm),"df")+1)
plot(m.nb, resid(.) ~ g)# works, as long as data 'dd' is found

## End(Not run)
```

hglm: A Package for Fitting Hierarchical Generalized Linear Models

by Lars Rönnegård, Xia Shen and Moudud Alam

Abstract We present the **hglm** package for fitting hierarchical generalized linear models. It can be used for linear mixed models and generalized linear mixed models with random effects for a variety of links and a variety of distributions for both the outcomes and the random effects. Fixed effects can also be fitted in the dispersion part of the model.

Introduction

The **hglm** package (Alam et al., 2010) implements the estimation algorithm for hierarchical generalized linear models (HGLM; Lee and Nelder, 1996). The package fits generalized linear models (GLM; McCullagh and Nelder, 1989) with random effects, where the random effect may come from a distribution conjugate to one of the exponential-family distributions (normal, gamma, beta or inverse-gamma). The user may explicitly specify the design matrices both for the fixed and random effects. In consequence, correlated random effects, as well as random regression models can be fitted. The dispersion parameter can also be modeled with fixed effects.

The main function is `hglm()` and the input is specified in a similar manner as for `glm()`. For instance,

```
R> hglm(fixed = y ~ week, random = ~ 1|ID,
        family = binomial(link = logit))
```

fits a logit model for `y` with `week` as fixed effect and `ID` representing the clusters for a normally distributed random intercept. Given an `hglm` object, the standard generic functions are `print()`, `summary()` and `plot()`.

Generalized linear mixed models (GLMM) have previously been implemented in several R functions, such as the `lmer()` function in the **lme4** package (Bates and Maechler, 2010) and the `glmmPQL()` function in the **MASS** package (Venables and Ripley, 2002). In GLMM, the random effects are assumed to be Gaussian whereas the `hglm()` function allows other distributions to be specified for the random effect. The `hglm()` function also extends the fitting algorithm of the **dglm** package (Dunn and Smyth, 2009) by including random effects in the linear predictor for the mean, i.e. it extends the algorithm so that it can cope with mixed models. Moreover, the model specification in `hglm()` can be given as a formula or alternatively in terms of `y`, `X`, `Z` and `X.disp`. Here `y` is the vector of observed responses, `X` and `Z` are the design matrices for the fixed and random effects, respectively, in the linear predictor for the means and `X.disp` is the design matrix for the fixed effects in the dispersion parameter. This enables a more flexible modeling of the random effects than specifying the model by an R formula. Consequently, this option is not as user friendly but gives the user the possibility to fit random regression models and random effects with known correlation structure.

The **hglm** package produces estimates of fixed effects, random effects and variance components as well as their standard errors. In the output it also produces diagnostics such as deviance components and leverages.

Three illustrating models

The **hglm** package makes it possible to

1. include fixed effects in a model for the residual variance,
2. fit models where the random effect distribution is not necessarily Gaussian,
3. estimate variance components when we have correlated random effects.

Below we describe three models that can be fitted using `hglm()`, which illustrate these three points. Later, in the Examples section, five examples are presented that include the R syntax and output for the `hglm()` function.

Linear mixed model with fixed effects in the residual variance

We start by considering a normal-normal model with heteroscedastic residual variance. In biology, for instance, this is important if we wish to model a random genetic effect (e.g., Rönnegård and Carlborg, 2007) for a trait y, where the residual variance differs between the sexes.

For the response y and observation number i we have:

$$y_i \mid \beta, u, \beta_d \sim N\left(X_i\beta + Z_i u, \exp\left(X_{d,i}\beta_d\right)\right)$$
$$u \sim MVN\left(0, \mathbf{I}\sigma_u^2\right)$$

where β are the fixed effects in the mean part of the model, the random effect u represents random variation among clusters of observations and β_d is the fixed effect in the residual variance part of the model. The variance of the random effect u is given by σ_u^2.

The subscript i for the matrices $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{X}_d$ indicates the i'th row. Here, a log link function is used for the residual variance and the model for the residual variance is therefore given by $\exp(X_{d,i}\beta_d)$. In the more general GLM notation, the residual variance here is described by the dispersion term ϕ, so we have $\log(\phi_i) = X_{d,i}\beta_d$.

This model cannot be fitted with the **dglm** package, for instance, because we have random effects in the mean part of the model. It is also beyond the scope of the `lmer()` function since we allow a model for the residual variance.

The implementation in `hglm()` for this model is demonstrated in Example 2 in the Examples section below.

A Poisson model with gamma distributed random effects

For dependent count data it is common to model a Poisson distributed response with a gamma distributed random effect (Lee et al., 2006). If we assume no overdispersion conditional on u and thereby have a fixed dispersion term, this model may be specified as:

$$E(y_i \mid \beta, u) = \exp(X_i\beta + Z_i v)$$

where a level j in the random effect v is given by $v_j = \log(u_j)$ and u_j are iid with gamma distribution having mean and variance: $E(u_j) = 1$, $var(u_j) = \lambda$.

This model can also be fitted with the **hglm** package, since it extends existing GLMM functions (e.g. `lmer()`) to allow a non-normal distribution for the random effect. Later on, in Example 3, we show the `hglm()` code used for fitting a gamma-Poisson model with fixed effects included in the dispersion parameter.

A linear mixed model with a correlated random effect

In animal breeding it is important to estimate variance components prior to ranking of animal performances (Lynch and Walsh, 1998). In such models the genetic effect of each animal is modeled as a level in a random effect and the correlation structure $\mathbf{A}$ is a matrix with known elements calculated from the pedigree information. The model is given by

$$y_i \mid \beta, u \sim N\left(X_i\beta + Z_i u, \sigma_e^2\right)$$
$$u \sim MVN\left(0, \mathbf{A}\sigma_u^2\right)$$

This may be reformulated as (see Lee et al., 2006; Rönnegård and Carlborg, 2007)

$$y_i \mid \beta, u \sim N\left(X_i\beta + Z_i^* u^*, \sigma_e^2\right)$$
$$u^* \sim MVN(0, \mathbf{I}\sigma_u^2)$$

where $\mathbf{Z}^* = \mathbf{ZL}$ and $\mathbf{L}$ is the Cholesky factorization of $\mathbf{A}$.

Thus the model can be fitted using the `hglm()` function with a user-specified input matrix `Z` (see R code in Example 4 below).

Overview of the fitting algorithm

The fitting algorithm is described in detail in Lee et al. (2006) and is summarized as follows. Let n be the number of observations and k be the number of levels in the random effect. The algorithm is then:

1. Initialize starting values.
2. Construct an augmented model with response $y_{aug} = \begin{pmatrix} y \\ E(u) \end{pmatrix}$.
3. Use a GLM to estimate β and v given the vector ϕ and the dispersion parameter for the random effect λ. Save the deviance components and leverages from the fitted model.
4. Use a gamma GLM to estimate β_d from the first n deviance components d and leverages h obtained from the previous model. The response variable and weights for this model are $d/(1-h)$ and $(1-h)/2$, respectively. Update the dispersion parameter by putting ϕ equal to the predicted response values for this model.
5. Use a similar GLM as in Step 4 to estimate λ from the last k deviance components and leverages obtained from the GLM in Step 3.
6. Iterate between steps 3-5 until convergence.

For a more detailed description of the algorithm in a particular context, see below.

H-likelihood theory

Let y be the response and u an unobserved random effect. The **hglm** package fits a hierarchical model $y \mid u \sim f_m(\mu, \phi)$ and $u \sim f_d(\psi, \lambda)$ where f_m and f_d are specified distributions for the mean and dispersion parts of the model.

We follow the notation of Lee and Nelder (1996), which is based on the GLM terminology by McCullagh and Nelder (1989). We also follow the likelihood approach where the model is described in terms of likelihoods. The conditional (log-)likelihood for y given u has the form of a GLM

$$\ell(\theta', \phi; y \mid u) = \frac{y\theta' - b(\theta')}{a(\phi)} + c(y, \phi) \qquad (1)$$

where θ' is the canonical parameter, ϕ is the dispersion term, μ' is the conditional mean of y given u

where $\eta' = g(\mu')$, i.e. $g()$ is a link function for the GLM. The linear predictor is given by $\eta' = \eta + v$ where $\eta = X\beta$ and $v = v(u)$ for some strict monotonic function of u. The link function $v(u)$ should be specified so that the random effects occur linearly in the linear predictor to ensure meaningful inference from the h-likelihood (Lee et al., 2007). The *h-likelihood* or hierarchical likelihood is defined by

$$h = \ell(\theta', \phi; y \mid u) + \ell(\alpha; v) \tag{2}$$

where $\ell(\alpha; v)$ is the log density for v with parameter α. The estimates of β and v are given by $\frac{\partial h}{\partial \beta} = 0$ and $\frac{\partial h}{\partial v} = 0$. The dispersion components are estimated by maximizing the *adjusted profile h-likelihood*

$$h_p = \left(h - \frac{1}{2}\log|-\frac{1}{2\pi}H| \right)_{\beta=\hat{\beta}, v=\hat{v}} \tag{3}$$

where H is the Hessian matrix of the h-likelihood. The dispersion term ϕ can be connected to a linear predictor $X_d\beta_d$ given a link function $g_d(\cdot)$ with $g_d(\phi) = X_d\beta_d$. The adjusted profile likelihoods of ℓ and h may be used for inference of β, v and the dispersion parameters ϕ and λ (pp. 186 in Lee et al., 2006). More detail and discussion of h-likelihood theory is presented in the **hglm** vignette.

Detailed description of the hglm fitting algorithm for a linear mixed model with heteroscedastic residual variance

In this section we describe the fitting algorithm in detail for a linear mixed model where fixed effects are included in the model for the residual variance. The extension to distributions other than Gaussian is described at the end of the section.

Lee and Nelder (1996) showed that linear mixed models can be fitted using a hierarchy of GLM by using an augmented linear model. The linear mixed model

$$y = \mathbf{X}b + \mathbf{Z}u + e$$
$$v = \mathbf{Z}\mathbf{Z}^T\sigma_u^2 + \mathbf{R}\sigma_e^2$$

where $\mathbf{R}$ is a diagonal matrix with elements given by the estimated dispersion model (i.e. ϕ defined below). In the first iteration of the HGLM algorithm, $\mathbf{R}$ is an identity matrix. The model may be written as an augmented weighted linear model:

$$y_a = \mathbf{T}_a\delta + e_a \tag{4}$$

where

$$y_a = \begin{pmatrix} y \\ 0_q \end{pmatrix} \qquad \mathbf{T}_a = \begin{pmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I}_q \end{pmatrix}$$
$$\delta = \begin{pmatrix} b \\ u \end{pmatrix} \qquad e_a = \begin{pmatrix} e \\ -u \end{pmatrix}$$

Here, q is the number of columns in $\mathbf{Z}$, 0_q is a vector of zeros of length q, and $\mathbf{I}_q$ is the identity matrix of size $q \times q$. The variance-covariance matrix of the augmented residual vector is given by

$$V(e_a) = \begin{pmatrix} \mathbf{R}\sigma_e^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_q\sigma_u^2 \end{pmatrix}$$

Given σ_e^2 and σ_u^2, this weighted linear model gives the same estimates of the fixed and random effects (b and u respectively) as Henderson's mixed model equations (Henderson, 1976).

The estimates from weighted least squares are given by:

$$\mathbf{T}_a^t\mathbf{W}^{-1}\mathbf{T}_a\hat{\delta} = \mathbf{T}_a^t\mathbf{W}^{-1}y_a$$

where $\mathbf{W} \equiv V(e_a)$.

The two variance components are estimated iteratively by applying a gamma GLM to the residuals e_i^2 and u_i^2 with intercept terms included in the linear predictors. The leverages h_i for these models are calculated from the diagonal elements of the hat matrix:

$$\mathbf{H}_a = \mathbf{T}_a(\mathbf{T}_a^t\mathbf{W}^{-1}\mathbf{T}_a)^{-1}\mathbf{T}_a^t\mathbf{W}^{-1} \tag{5}$$

A gamma GLM is used to fit the dispersion part of the model with response

$$y_{d,i} = e_i^2/(1-h_i) \tag{6}$$

where $E(y_d) = \mu_d$ and $\mu_d \equiv \phi$ (i.e. σ_e^2 for a Gaussian reponse). The GLM model for the dispersion parameter is then specified by the link function $g_d(.)$ and the linear predictor $X_d\beta_d$, with prior weights $(1-h_i)/2$, for

$$g_d(\mu_d) = X_d\beta_d \tag{7}$$

Similarly, a gamma GLM is fitted to the dispersion term α (i.e. σ_u^2 for a GLMM) for the random effect v, with

$$y_{\alpha,j} = u_j^2/(1-h_{n+j}), j = 1,2,...,q \tag{8}$$

and

$$g_\alpha(\mu_\alpha) = \lambda \tag{9}$$

where the prior weights are $(1-h_{n+j})/2$ and the estimated dispersion term for the random effect is given by $\hat{\alpha} = g_\alpha^{-1}(\hat{\lambda})$.

The algorithm iterates by updating both $\mathbf{R} = diag(\hat{\phi})$ and $\sigma_u^2 = \hat{\alpha}$, and subsequently going back to Eq. (4).

For a non-Gaussian response variable y, the estimates are obtained simply by fitting a GLM instead of Eq. (4) and by replacing e_i^2 and u_j^2 with the deviance components from the augmented model (see Lee et al., 2006).

Implementation details

Distributions and link functions

There are two important classes of models that can be fitted in **hglm**: GLMM and conjugate HGLM. GLMMs have Gaussian random effects. Conjugate HGLMs have been commonly used partly due to the fact that explicit formulas for the marginal likelihood exist. HGLMs may be used to fit models in survival analysis (frailty models), where for instance the complementary-log-log link function can be used on binary responses (see e.g., Carling et al., 2004). The gamma distribution plays an important role in modeling responses with a constant coefficient of variation (see Chapter 8 in McCullagh and Nelder, 1989). For such responses with a gamma distributed random effect we have a gamma-gamma model. A summary of the most important models is given in Tables 1 and 2. Note that the random-effect distribution can be an arbitrary conjugate exponential-family distribution. For the specific case where the random-effect distribution is a conjugate to the distribution of y, this is called a *conjugate HGLM*. Further implementation details can be found in the **hglm** vignette.

Possible future developments

In the current version of `hglm()` it is possible to include a single random effect in the mean part of the model. An important development would be to include several random effects in the mean part of the model and also to include random effects in the dispersion parts of the model. The latter class of models is called Double HGLM and has been shown to be a useful tool for modeling heavy tailed distributions (Lee and Nelder, 2006).

The algorithm of `hglm()` gives true marginal likelihood estimates for the fixed effects in conjugate HGLM (Lee and Nelder, 1996, pp. 629), whereas for other models the estimates are approximated. Lee and co-workers (see Lee et al., 2006, and references therein) have developed higher-order approximations, which are not implemented in the current version of the **hglm** package. For such extensions, we refer to the commercially available GenStat software (Payne et al., 2007), the recently available R package **HGLMMM** (Molas, 2010) and also to coming updates of **hglm**.

Examples

Example 1: A linear mixed model

Data description The output from the `hglm()` function for a linear mixed model is compared to the results from the `lme()` function in the **nlme** (Pinheiro et al., 2009) package using simulated data. In the simulated data there are five clusters with 20 observations in each cluster. For the mean part of the model, the simulated intercept value is $\mu = 0$, the variance for the random effect is $\sigma_u^2 = 0.2$, and the residual variance is $\sigma_e^2 = 1.0$.

Both functions produce the same estimate of the fixed intercept effect of 0.1473 (s.e. 0.16) and also the same variance component estimates. The `summary.hglm()` function gives the estimate of the variance component for the random intercept (0.082) as well as the residual variance (0.84). It also gives the logarithm of the variance component estimates together with standard errors below the lines `Model estimates for the dispersion term` and `Dispersion model for the random effects`. The `lme()` function gives the square root of the variance component estimates.

The model diagnostics produced by the `plot.hglm` function are shown in Figures 1 and 2. The data are completely balanced and therefore produce equal leverages (hatvalues) for all observations and also for all random effects (Figure 1). Moreover, the assumption of the deviance components being gamma distributed is acceptable (Figure 2).

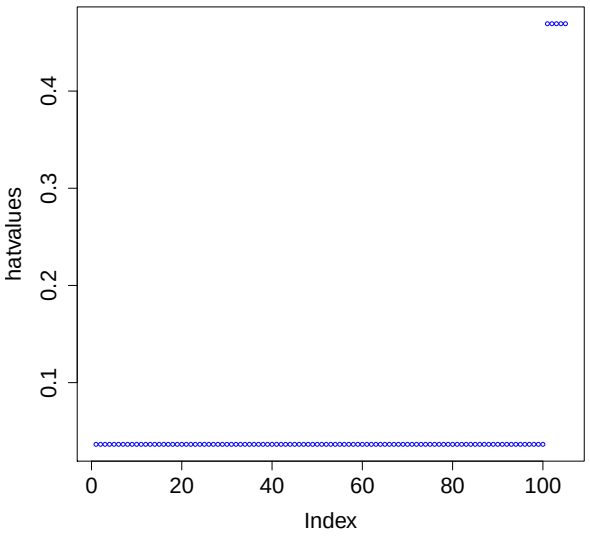


Figure 1: Hatvalues (i.e. diagonal elements of the augmented hat-matrix) for each observation 1 to 100, and for each level in the random effect (index 101-105).

Table 1: Commonly used distributions and link functions possible to fit with `hglm()`

Model name	$y \mid u$ distribution	Link $g(\mu)$	u distribution	Link $v(u)$
Linear mixed model	Gaussian	identity	Gaussian	identity
Binomial conjugate	Binomial	logit	Beta	logit
Binomial GLMM	Binomial	logit	Gaussian	identity
Binomial frailty	Binomial	comp-log-log	Gamma	log
Poisson GLMM	Poisson	log	Gaussian	identity
Poisson conjugate	Poisson	log	Gamma	log
Gamma GLMM	Gamma	log	Gaussian	identity
Gamma conjugate	Gamma	inverse	Inverse-Gamma	inverse
Gamma-Gamma	Gamma	log	Gamma	log

Table 2: `hglm` code for commonly used models

Model name	Setting for `family` argument	Setting for `rand.family` argument
Linear mixed model[a]	`gaussian(link = identity)`	`gaussian(link = identity)`
Beta-Binomial	`binomial(link = logit)`	`Beta(link = logit)`
Binomial GLMM	`binomial(link = logit)`	`gaussian(link = identity)`
Binomial frailty	`binomial(link = cloglog)`	`Gamma(link = log)`
Poisson GLMM	`poisson(link = log)`	`gaussian(link = identity)`
Poisson frailty	`poisson(link = log)`	`Gamma(link = log)`
Gamma GLMM	`Gamma(link = log)`	`gaussian(link = identity)`
Gamma conjugate	`Gamma(link = inverse)`	`inverse.gamma(link = inverse)`
Gamma-Gamma	`Gamma(link = log)`	`Gamma(link = log)`

[a]For example, the `hglm()` code for a linear mixed model is

```
hglm(family = gaussian(link = identity), rand.family = gaussian(link = identity), ...)
```

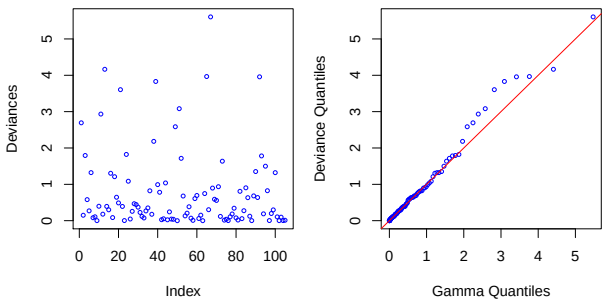


Figure 2: Deviance diagnostics for each observation and each level in the random effect.

The R code and output for this example is as follows:

```
R> set.seed(123)
R> n.clus <- 5 #No. of clusters
R> n.per.clus <- 20  #No. of obs. per cluster
R> sigma2_u <- 0.2 #Variance of random effect
R> sigma2_e <- 1 #Residual variance
R> n <- n.clus*n.per.clus
R> X <- matrix(1, n, 1)
R> Z <- diag(n.clus)%x%rep(1, n.per.clus)
R> a <- rnorm(n.clus, 0, sqrt(sigma2_u))
R> e <- rnorm(n, 0, sqrt(sigma2_e))
R> mu <- 0
R> y <- mu + Z%*%a + e
R> lmm <- hglm(y = y, X = X, Z = Z)
R> summary(lmm)
```

```
R> plot(lmm)

Call:
hglm.default(X = X, y = y, Z = Z)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:[1] 0.8400608

Model estimates for the dispersion term:
Link = log
Effects:
  Estimate Std. Error
   -0.1743     0.1441

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.08211

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
   -2.4997     0.8682

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).

MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1   0.1473     0.1580   0.933    0.353
Note: P-values are based on 96 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]  -0.3237     0.1971
[2,]  -0.0383     0.1971
```

```
[3,]   0.3108     0.1971
[4,]  -0.0572     0.1971
[5,]   0.1084     0.1971

EQL estimation converged in 5 iterations.
```

```
R> #Same analysis with the lme function
R> library(nlme)
R> clus <- rep(1:n.clus,
+              rep(n.per.clus, n.clus))
R> summary(lme(y ~ 0 + X,
+              random = ~ 1 | clus))

Linear mixed-effects model fit by REML
Data: NULL
      AIC      BIC    logLik
  278.635 286.4203 -136.3175

Random effects:
Formula: ~1 | clus
        (Intercept) Residual
StdDev:   0.2859608   0.9166

Fixed effects: y ~ 0 + X
      Value Std.Error DF   t-value p-value
X 0.1473009 0.1573412 95 0.9361873  0.3516

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.5834807 -0.6570612  0.0270673  0.6677986  2.1724148

Number of Observations: 100
Number of Groups: 5
```

Example 2: Analysis of simulated data for a linear mixed model with heteroscedastic residual variance

Data description Here, a heteroscedastic residual variance is added to the simulated data from the previous example. Given the explanatory variable x_d, the simulated residual variance is 1.0 for $x_d = 0$ and 2.72 for $x_d = 1$. The output shows that the variance of the random effect is 0.109, and that $\hat{\beta}_d = (-0.32, 1.47)$, i.e. the two residual variances are estimated as 0.72 and 3.16. (Code continued from Example 1)

```
R> beta.disp <- 1
R> X_d <- matrix(1, n, 2)
R> X_d[,2] <- rbinom(n, 1, .5)
R> colnames(X_d) <- c("Intercept", "x_d")
R> e <- rnorm(n, 0,
+        sqrt(sigma2_e*exp(beta.disp*X_d[,2])))
R> y <- mu + Z%*%a + e
R> summary(hglm(y = y, X = X, Z = Z,
+                X.disp = X_d))

Call:
hglm.default(X = X, y = y, Z = Z, X.disp = X_d)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:
Link = log
Effects:
          Estimate Std. Error
Intercept  -0.3225     0.2040
x_d         1.4744     0.2881

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.1093

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
   -2.2135     0.8747

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1  -0.0535     0.1836  -0.291    0.771
Note: P-values are based on 96 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]   0.0498     0.2341
[2,]  -0.2223     0.2276
[3,]   0.4404     0.2276
[4,]  -0.1786     0.2276
[5,]  -0.0893     0.2296

EQL estimation converged in 5 iterations.
```

Example 3: Fitting a Poisson model with gamma random effects, and fixed effects in the dispersion term

Data description We simulate a Poisson model with random effects and estimate the parameter in the dispersion term for an explanatory variable x_d. The estimated dispersion parameter for the random effects is 0.6556. (Code continued from Example 2)

```
R> u <- rgamma(n.clus,1)
R> eta <- exp(mu + Z%*%u)
R> y <- rpois(length(eta), eta)
R> gamma.pois <- hglm(y = y, X = X, Z = Z,
+                      X.disp = X_d,
+                      family = poisson(
+                      link = log),
+                      rand.family =
+                      Gamma(link = log))
R> summary(gamma.pois)

Call:
hglm.default(X = X, y = y, Z = Z,
    family = poisson(link = log),
    rand.family = Gamma(link = log), X.disp = X_d)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:
Link = log
Effects:
          Estimate Std. Error
Intercept  -0.0186     0.2042
x_d         0.4087     0.2902

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
```

```
[1] 1.926

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
    0.6556     0.7081

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1   2.3363     0.6213    3.76 0.000293
---

Note: P-values are based on 95 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]   1.1443     0.6209
[2,]  -1.6482     0.6425
[3,]  -2.5183     0.6713
[4,]  -1.0243     0.6319
[5,]   0.2052     0.6232

EQL estimation converged in 3 iterations.
```

Example 4: Incorporating correlated random effects in a linear mixed model - a genetics example

Data description The data consists of 2025 individuals from two generations where 1000 individuals have observed trait values y that are approximately normal (Figure 3). The data we analyze was simulated for the QTLMAS 2009 Workshop (Coster et al., 2010)[1]. A longitudinal growth trait was simulated. For simplicity we analyze only the values given on the third occasion at age 265 days.

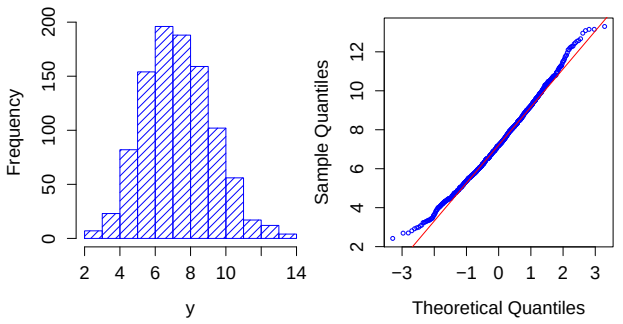


Figure 3: Histogram and qqplot for the analyzed trait.

We fitted a model with a fixed intercept and a random animal effect, a, where the correlation structure of a is given by the additive relationhip matrix $\mathbf{A}$ (which is obtained from the available pedigree information). An incidence matrix $\mathbf{Z}_0$ was constructed and relates observation number with id-number in the pedigree. For observation y_i coming from individual j in the ordered pedigree file $\mathbf{Z}_0[i,j] = 1$, and all other elements are 0. Let $\mathbf{L}$ be the Cholesky factorization of $\mathbf{A}$, and $\mathbf{Z} = \mathbf{Z}_0\mathbf{L}$. The design matrix for the fixed effects, $\mathbf{X}$, is a column of ones. The estimated variance components are $\hat{\sigma}^2_e = 2.21$ and $\hat{\sigma}^2_u = 1.50$.

The R code for this example is given below.

```
R> data(QTLMAS)
R> y <- QTLMAS[,1]
R> Z <- QTLMAS[,2:2026]
R> X <- matrix(1, 1000, 1)
R> animal.model <- hglm(y = y, X = X, Z = Z)
R> print(animal.model)

Call:
hglm.default(X = X, y = y, Z = Z)

Fixed effects:
     X.1
7.279766
Random effects:
   [1] -1.191733707  1.648604776  1.319427376 -0.928258503
   [5] -0.471083317 -1.058333534  1.011451565  1.879641994
   [9]  0.611705900 -0.259125073 -1.426788944 -0.005165978
...

Dispersion parameter for the mean model:[1] 2.211169
Dispersion parameter for the random effects:[1] 1.502516

EQL estimation converged in 2 iterations
```

Example 5: Binomial-beta model applied to seed germination data

Data description The seed germination data presented by Crowder (1978) has previously been analyzed using a binomial GLMM (Breslow and Clayton, 1993) and a binomial-beta HGLM (Lee and Nelder, 1996). The data consists of 831 observations from 21 germination plates. The effect of seed variety and type of root extract was studied in a 2×2 factorial lay-out. We fit the binomial-beta HGLM used by Lee and Nelder (1996) and setting `fix.disp = 1` in `hglm()` produces comparable estimates to the ones obtained by Lee and Nelder (with differences $< 2 \times 10^{-3}$). The beta distribution parameter α in Lee and Nelder (1996) was defined as $1/(2a)$ where a is the dispersion term obtained from `hglm()`. The output from the R code given below gives $\hat{a} = 0.0248$ and the corresponding estimate given in Lee and Nelder (1996) is $\hat{a} = 1/(2\hat{\alpha}) = 0.023$. We conclude that the **hglm** package produces similar results as the ones presented in Lee and Nelder (1996) and the dispersion parameters estimated using the EQL method in GenStat differ by less than 1%. Additional examples, together with comparisons to estimates produced by GenStat, are given in the **hglm** vignette included in the package on CRAN.

```
R> data(seeds)
R> germ <- hglm(
+    fixed = r/n ~ extract*I(seed=="073"),
```

[1] http://www.qtlmas2009.wur.nl/UK/Dataset

```
+     weights = n, data = seeds,
+     random = ~1|plate, family = binomial(),
+     rand.family = Beta(), fix.disp = 1)
R> summary(germ)

Call:
hglm.formula(family = binomial(), rand.family = Beta(),
    fixed = r/n ~ extract * I(seed == "O73"),
    random = ~1 | plate, data = seeds,
    weights = n, fix.disp = 1)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:[1] 1

Model estimates for the dispersion term:
Link = log
Effects:
[1] 1

Dispersion = 1 is used in Gamma model on deviances to
calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.02483

Dispersion model for the random effects:
Link = log

Effects:
  Estimate Std. Error
   -3.6956     0.5304

Dispersion = 1 is used in Gamma model on deviances to
calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
                              Estimate Std. Error t value
(Intercept)                    -0.5421    0.1928  -2.811
extractCucumber                 1.3386    0.2733   4.898
I(seed == "O73")TRUE            0.0751    0.3114   0.241
extractCucumber:I(seed=="O73")  -0.8257    0.4341  -1.902
                              Pr(>|t|)
(Intercept)                   0.018429
extractCucumber               0.000625
I(seed == "O73")TRUE          0.814264
extractCucumber:I(seed=="O73") 0.086343
---

Note: P-values are based on 10 degrees of freedom
Summary of the random effects estimate
      Estimate Std. Error
 [1,]  -0.2333     0.2510
 [2,]   0.0085     0.2328
 ...
[21,]  -0.0499     0.2953

EQL estimation converged in 7 iterations.
```

Summary

The hierarchical generalized linear model approach offers new possibilities to fit generalized linear models with random effects. The **hglm** package extends existing GLMM fitting algorithms to include fixed effects in a model for the residual variance, fits models where the random effect distribution is not necessarily Gaussian and estimates variance components for correlated random effects. For such models there are important applications in, for instance: genetics (Noh et al., 2006), survival analysis (Ha and Lee, 2005), credit risk modeling (Alam and Carling, 2008), count data (Lee et al., 2006) and dichotomous responses (Noh and Lee, 2007). We therefore expect that this new package will be of use for applied statisticians in several different fields.

Bibliography

M. Alam and K. Carling. Computationally feasible estimation of the covariance structure in generalized linear mixed models GLMM. *Journal of Statistical Computation and Simulation*, 78:1227–1237, 2008.

M. Alam, L. Ronnegard, and X. Shen. *hglm: Hierarchical Generalized Linear Models*, 2010. URL http://CRAN.R-project.org/package=hglm. R package version 1.1.1.

D. Bates and M. Maechler. *lme4: Linear mixed-effects models using S4 classes*, 2010. URL http://CRAN.R-project.org/package=lme4. R package version 0.999375-37.

N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25, 1993.

K. Carling, L. Rönnegård, and K. Roszbach. An analysis of portfolio credit risk when counterparties are interdependent within industries. *Sveriges Riksbank Working Paper*, 168, 2004.

A. Coster, J. Bastiaansen, M. Calus, C. Maliepaard, and M. Bink. QTLMAS 2010: Simulated dataset. *BMC Proceedings*, 4(Suppl 1):S3, 2010.

M. J. Crowder. Beta-binomial ANOVA for proportions. *Applied Statistics*, 27:34–37, 1978.

P. K. Dunn and G. K. Smyth. *dglm: Double generalized linear models*, 2009. URL http://CRAN.R-project.org/package=dglm. R package version 1.6.1.

I. D. Ha and Y. Lee. Comparison of hierarchical likelihood versus orthodox best linear unbiased predictor approaches for frailty models. *Biometrika*, 92: 717–723, 2005.

C. R. Henderson. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics*, 32(1):69–83, 1976.

Y. Lee and J. A. Nelder. Double hierarchical generalized linear models with discussion. *Applied Statistics*, 55:139–185, 2006.

Y. Lee and J. A. Nelder. Hierarchical generalized linear models with discussion. *J. R. Statist. Soc. B*, 58: 619–678, 1996.

Y. Lee, J. A. Nelder, and Y. Pawitan. *Generalized linear models with random effects*. Chapman & Hall/CRC, 2006.

Y. Lee, J. A. Nelder, and M. Noh. H-likelihood: problems and solutions. *Statistics and Computing*, 17: 49–55, 2007.

M. Lynch and B. Walsh. *Genetics and analysis of Quantitative Traits*. Sinauer Associates, Inc., 1998. ISBN 087893481.

P. McCullagh and J. A. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989.

M. Molas. *HGLMMM: Hierarchical Generalized Linear Models*, 2010. URL http://CRAN.R-project.org/package=HGLMMM. R package version 0.1.1.

M. Noh and Y. Lee. REML estimation for binary data in GLMMs. *Journal of Multivariate Analysis*, 98:896–915, 2007.

M. Noh, B. Yip, Y. Lee, and Y. Pawitan. Multicomponent variance estimation for binary traits in family-based studies. *Genetic Epidemiology*, 30:37–47, 2006.

R. W. Payne, D. A. Murray, S. A. Harding, D. B. Baird, and D. M. Soutar. GenStat for Windows (10th edition) introduction, 2007. URL http://www.vsni.co.uk/software/genstat.

J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, and the R Core team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2009. URL http://CRAN.R-project.org/package=nlme. R package version 3.1-96.

L. Rönnegård and Ö. Carlborg. Separation of base allele and sampling term effects gives new insights in variance component QTL analysis. *BMC Genetics*, 8(1), 2007.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

Lars Rönnegård
Statistics Unit
Dalarna University, Sweden
and
Department of Animal Breeding and Genetics
Swedish University of Agricultural Sciences, Sweden
lrn@du.se

Xia Shen
Department of Cell and Molecular Biology
Uppsala University, Sweden
and
Statistics Unit
Dalarna University, Sweden
xia.shen@lcb.uu.se

Moudud Alam
Statistics Unit
Dalarna University, Sweden
maa@du.se