Fitting generalized linear mixed-effects models using lme4

Anna Ly
McMaster University

Rune Haubo Bojesen Christensen
Copenhagen Research Centre
for Biological and Precision Psychiatry

Douglas Bates
University of Wisconsin - Madison

Martin Mächler
ETH Zurich

Benjamin M. Bolker
McMaster University

Abstract

The **lme4** R package can be used to fit generalized linear mixed models (GLMMs), which extend the class of linear mixed models (LMMs). The two main extensions provided by GLMMs are (1) allowing for the conditional distribution of the response given the random effects to be non-Gaussian (e.g. binomial, Poisson) and (2) allowing the conditional mean to be a nonlinear function of a linear combination of the fixed and random effect coefficients, via an inverse link function. The conditional mode of the random effects given the observed data, the variance-covariance matrix of the random effects, and the fixed effect parameters are determined using penalized iteratively reweighted least squares (PIRLS). We compute an approximation of the integral over the distributions of the conditional modes to compute the MLE for a given set of parameters (by default we use the Laplace approximation or, alternatively, the more computationally expensive adaptive Gauss-Hermite quadrature). The package provides all the standard features available for GLMs in base R, including the the standard set of accessor functions as well as the possibility of user-specified distributions (within the exponential dispersion family) and link functions.

Keywords: sparse matrix methods, generalized linear mixed models, penalized least squares, Cholesky decomposition.

1. Introduction

FIXME: check digits of output?

The **lme4** package for R can be used to fit a broad range of mixed-effects models. One major advantage of **lme4** over its predecessor, **nlme**, is that it can be used to fit generalized linear mixed models (GLMMs), which combine the flexibility of linear mixed models (LMMs) and generalized linear models (GLMs). In a companion paper, we have described the facilities in **lme4** for fitting linear mixed models (LMMs). Here we describe the facilities for fitting

GLMMs.

2. Generalized Linear Mixed Models

Generalized linear mixed models extend the class of generalized linear models by allowing for both fixed and random effects. In a GL(M)M, the length-n vector-valued response variable, $\mathcal{Y}$, has a conditional distribution in the exponential dispersion family (e.g. Normal, binomial, Poisson).[1] The mean, $\boldsymbol{\mu}_{\mathcal{Y}}$, of $\mathcal{Y}$ depends on a linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}. \tag{1}$$

where $\boldsymbol{\beta}$ is a p-dimensional coefficient vector and $\boldsymbol{X}$ is an $n \times p$ model matrix. The mapping from $\boldsymbol{\mu}_{\mathcal{Y}}$ to $\boldsymbol{\eta}$, which is called the *link function* and written,

$$\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\eta} = \boldsymbol{g}\left(\boldsymbol{\mu}_{\mathcal{Y}}\right), \tag{2}$$

is a *diagonal mapping* in the sense that there is a scalar function, g, such that the ith component of $\boldsymbol{\eta}$ is g applied to the ith component of $\boldsymbol{\mu}_{\mathcal{Y}}$. (The name "diagonal" reflects the fact that the Jacobian matrix, $\frac{d\eta}{d\mu'}$, of such a mapping will be diagonal.) The scalar link function must be invertible and differentiable over its range. The vector-valued *inverse link* function, $\boldsymbol{g}^{-1}$, will be the scalar inverse link, g^{-1}, applied component-wise to $\boldsymbol{\eta}$.

In the GLMM case, the vector of means of the exponential dispersion family distribution of $\mathcal{Y}$ depends on an unobserved random vector, $\mathcal{B}$, of length q, called the random-effects coefficient vector. In particular, the conditional mean of $\mathcal{Y}$ given that $\mathcal{B} = \boldsymbol{b}$, written $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}$, depends on the linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}\boldsymbol{\beta}. \tag{3}$$

where $\boldsymbol{Z}$ is an $n \times q$ random-effects model matrix. Similar to the GLM case, the mapping from the conditional mean, $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}$, to the linear predictor, $\boldsymbol{\eta}$, is

$$\boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\eta} = \boldsymbol{g}\left(\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{B}=\boldsymbol{b}}\right), \tag{4}$$

The random vector $\mathcal{B}$ is assumed to be distributed multivariate normally,

$$\mathcal{B} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\theta) \tag{5}$$

where $\boldsymbol{\Sigma}_\theta$ is the covariance matrix of $\mathcal{B}$, which depends on a vector of covariance parameters, $\boldsymbol{\theta}$.

The optimization routines of **lme4** never actually compute $\boldsymbol{\Sigma}_\theta$ directly, and instead use the elements of the covariance factor, $\boldsymbol{\Lambda}_\theta$, which is a matrix square root of $\boldsymbol{\Sigma}_\theta$ (in practice a Cholesky factor),

$$\boldsymbol{\Sigma}_\theta = \boldsymbol{\Lambda}_\theta \boldsymbol{\Lambda}_\theta'. \tag{6}$$

[1] The `lme4` package supports fitting negative binomial GLMMs, which are an extension of GLMMs; the negative binomial family is within the exponential dispersion family if the dispersion parameter is fixed. The `glmer.nb` function uses a one-dimensional optimizer to estimate the dispersion parameter, fitting a negative binomial GLMM with a fixed dispersion parameter at each trial value. In practice this implementation is slower than those in other R packages (e.g., the `glmmTMB` package).

This characterization of the random-effects covariance structure allows us to write the linear predictor as

$$\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{\Lambda}_\theta \boldsymbol{u} + \boldsymbol{X}\boldsymbol{\beta}, \tag{7}$$

where the spherical random effects vector, $\boldsymbol{u}$ (see Bates, Mächler, Bolker, and Walker (2015)) is a realization of the random vector, $\mathcal{U}$,

$$\mathcal{U} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_q) \tag{8}$$

where $\boldsymbol{I}_q$ is the identity matrix.

Common forms of the conditional distribution are Bernoulli, for binary responses, binomial for binary responses that are recorded as the number of trials and the number of successes, and Poisson, for count data. The combination of a distributional form and a link function is called a *family*. For distributional forms in the exponential dispersion family there is a *canonical link*. For Bernoulli or binomial forms the canonical link is the *logit* link function

$$\eta_i = \log\left(\frac{\mu_i}{1-\mu_i}\right); \tag{9}$$

for the Poisson distribution the canonical link is the natural logarithm.

The form of the distribution determines the conditional variance, $\mathrm{Var}(\mathcal{Y}|\mathcal{U} = \boldsymbol{u})$, as a function of the conditional mean and, possibly, a separate scale factor. (In the most common cases the conditional variance is completely determined by the conditional mean.)

Assuming that the conditional distribution belongs to the exponential dispersion family, we can weight observations differently by scaling the dispersion parameter, ϕ. These **prior weights**, $\boldsymbol{w}$, a vector of size n, are known positive constants (equal to the number of observations per trial in the particular case of a binomial GLMM).

By scaling the dispersion parameters, we also modify the conditional variance. Consider the form, where $i = 1, 2, \ldots, n$:

$$\mathrm{Var}(\mathcal{Y}_i|\mathcal{U} = \boldsymbol{u}) = \frac{\phi}{w_i}\mathrm{Var}(\mu_i) \tag{10}$$

where $\mathrm{Var}(\mu_i)$ is the family-specific variance function. Higher weights indicate a smaller variance.

If prior weights are not specified, by default $\boldsymbol{w}$ is just a vector of 1's.

Another common modification when dealing with generalized linear mixed models is to allow for an **offset**, which is an efficient way to include known scaling factors (like population or area) without introducing additional parameters to the model.

In particular, the inclusion of an offset would modify the linear predictor as follows:

$$\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}\boldsymbol{\beta} + \text{offset}. \tag{11}$$

For GLMMs, the scale parameter is not typically estimated as part of the nonlinear minimization of the negative likelihood over the β and θ parameters. As is common in generalized linear model contexts, it is set equal to the (residual) deviance divided by the residual degrees of freedom. (Except for the Binomial and Poisson families, where the scale parameter is fixed at 1.)

The deviance itself is estimated using the method of moments estimator, which is the Pearson residual sum of squares. This approach is preferred by McCullagh and Nelder (1989) as it is less sensitive to small errors or model misspecification than the maximum likelihood estimator.

We therefore estimate the deviance using the penalized weighted Pearson residual sum of squares, as described in Section 3.2 of Bates *et al.* (2015).

The likelihood of the parameters, given the observed data, is now

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) = \int_{\mathbb{R}^q} f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})\, d\boldsymbol{u} \tag{12}$$

where, as in the case of linear mixed models, $f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})$ is the unscaled conditional density of $\mathcal{U}$ given $\mathcal{Y} = \boldsymbol{y}_{\text{obs}}$. The notation here is a bit blurred because, although the joint distribution of $\mathcal{Y}$ and $\mathcal{U}$ is always continuous with respect to $\mathcal{U}$, it can be (and often is) discrete with respect to $\mathcal{Y}$. However, when we condition on the observed value $\mathcal{Y} = \boldsymbol{y}_{\text{obs}}$, the resulting function is continuous with respect to $\boldsymbol{u}$ so the unscaled conditional density is indeed well-defined as a density, up to a scale factor.

To evaluate the integrand in (12) we need the unscaled conditional density $f_{\mathcal{Y},\mathcal{U}}$, which depends on the likelihood of $\boldsymbol{y}_{\text{obs}}$ conditional on $\boldsymbol{u}$: $f(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u}) = f(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u})f(\boldsymbol{u})$. As usual in likelihood estimation we will eventually be working, for convenience, with negative log likelihoods rather than likelihoods. However, base R's machinery for defining GLM models, or "families", does not provide a convenient expression for the negative log-likelihood: instead, objects of class `family` include a `dev.resids` function that returns a vector of observation-specific deviances, i.e. $-2(\log L(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u}) - \log L_{\text{sat}})$ where L_{sat} is the likelihood corresponding to a saturated model with as many parameters as observations. [2] For example, the log-likelihood for a Poisson observation y with mean λ is (omitting the normalization constant) $\log L(y, \lambda) = y \log \lambda - \lambda$; the deviance is

$$-2(\log L(y, \lambda) - \log L(y, y)) = -2(y(\log \lambda - \log y) - (\lambda - y))$$

(with appropriate adjustments if $y = 0$). Because deviances differ from (negative) log likelihoods only by a parameter-independent constant (the log-likelihood of the saturated model) and a change of scale (multiplication by 2), this change does not affect our optimization procedure.

One advantage of using the pre-existing GLM family structure is that the software can thus fit models with user-specified families and link functions (although common families and link functions are hard-coded in C++ for computational speed), in contrast to early versions of `lme4` and some other R packages for GLMM fitting.

If `dev.resids` returns $\boldsymbol{d}(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u})$, with elements $d_i(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u}), i = 1, \ldots, n$, the full deviance of a generalized linear model is

$$f(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u}) = \sum_{i=1}^{n} d_i(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u}) + \|\boldsymbol{u}\|^2.$$

[2] There is some confusion in R (and in its predecessor, S) about the exact definition of the deviance residuals of a family. As indicated above, we will use this name for the value of the `dev.resids` function for the family. The signed square root of this vector, using the signs of $\boldsymbol{y}_{\text{obs}} - \mu$, is returned when the `residuals` method is applied to a fitted model of class `"glm"` when `type="deviance"`, the default, is specified. Both are called "deviance residuals" at different points in the documentation.

The likelihood can now be expressed as

$$L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) = \int_{\mathbb{R}^q} \exp\left(-\frac{\sum_{i=1}^n d_i(\boldsymbol{y}_{\text{obs}}|\boldsymbol{u}) + \|\boldsymbol{u}\|^2}{2}\right)(2\pi)^{-q/2}\,d\boldsymbol{u} \tag{13}$$

As for linear mixed models, we simplify evaluation of the integral (12) by determining the value, $\tilde{\boldsymbol{u}}_{\beta,\theta}$, that maximizes the integrand. When the conditional density, $\mathcal{U}|\mathcal{Y} = \boldsymbol{y}_{\text{obs}}$, is multivariate Gaussian, this conditional mode will also be the conditional mean. However, for most families used in GLMMs, the mode and the mean need not coincide so we use the more general term and call $\tilde{\boldsymbol{u}}_{\beta,\theta}$ the *conditional mode*. We first describe the numerical methods for determining the conditional mode using the Penalized Iteratively Reweighted Least Squares (PIRLS) algorithm then return to the question of evaluating the integral (12).

2.1. Determining the conditional mode

The iteratively reweighted least squares (IRLS) algorithm is an efficient method of determining the maximum likelihood estimates of the coefficients in a generalized linear model. We extend it to a *penalized iteratively reweighted least squares* (PIRLS) algorithm for determining the conditional mode, $\tilde{\boldsymbol{u}}_{\beta,\theta}$. This algorithm has the form

1. Given parameter values, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, and starting estimates, $\boldsymbol{u}_0$, evaluate the linear predictor, $\boldsymbol{\eta}$, the corresponding conditional mean, $\boldsymbol{\mu}_{\mathcal{Y}|\mathcal{U}=\boldsymbol{u}}$, and the conditional variance. Establish the weights as the inverse of the variance. We write these weights in the form of a diagonal weight matrix, $\boldsymbol{W}$, although they are stored and manipulated as a vector.

2. Let
$$Q(\boldsymbol{u}) = \left\|\boldsymbol{W}^{1/2}\left(\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_{\mathcal{Y}|\mathcal{U}=\boldsymbol{u}}\right)\right\|^2 + \|\boldsymbol{u}\|^2. \tag{14}$$
Solve the penalized, weighted, nonlinear least squares problem
$$\arg\min_{\boldsymbol{u}}\left(Q(\boldsymbol{u})\right). \tag{15}$$

3. Update the weights, $\boldsymbol{W}$, and check for convergence. If not converged, go to step 2.

We use a Gauss-Newton algorithm with an orthogonality convergence criterion (Bates and Watts 1988, §2.2.3) to solve the penalized, weighted, nonlinear least squares problem in step 2. At the ith iteration we determine an increment, $\boldsymbol{\delta}_i$, as the solution to the penalized, weighted, linear least squares problem

$$\boldsymbol{\delta}_i = \arg\min_{\boldsymbol{\delta}}\left\|\begin{bmatrix}\boldsymbol{W}^{1/2}\left(\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_i\right)\\ \boldsymbol{u}_i\end{bmatrix} - \begin{bmatrix}\boldsymbol{W}^{1/2}\boldsymbol{M}_i\boldsymbol{Z}\boldsymbol{\Lambda}_\theta\\ \boldsymbol{I}_q\end{bmatrix}\boldsymbol{u}\right\|^2 \tag{16}$$

where $\boldsymbol{u}_i$ is current value of $\boldsymbol{u}$, $\boldsymbol{\mu}_i$ is the corresponding conditional mean of $\mathcal{Y}|\mathcal{U} = \boldsymbol{u}_i$ and $\boldsymbol{M}_i$ is the Jacobian matrix of the vector-valued inverse link, evaluated at $\boldsymbol{\mu}_i$. That is

$$\boldsymbol{M}_i = \left.\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}'}\right|_{\boldsymbol{\eta}_i}, \tag{17}$$

which will be a diagonal matrix so, as for the weights, we store and manipulate the Jacobian as a vector.

When solving for the minimum of $Q(\boldsymbol{u})$, the proposed update for $\boldsymbol{u}$ may increase rather than decrease the value of the objective function $Q(\boldsymbol{u})$, if the step size estimated by the Gauss-Newton step is too large. Let $Q(\boldsymbol{u}_i)$ represent the value of the objective function $Q(\boldsymbol{u})$ at the current iterate $\boldsymbol{u}_i$; similarly, let $Q(\boldsymbol{u}_{i-1})$ represent the the value of the objective function of the previous iteration.

If $Q(\boldsymbol{u}_i) > Q(\boldsymbol{u}_{i-1})$, we proceed with a *step-halving* procedure. That is, we successively consider scaled updates of the form

$$\boldsymbol{u}_i^{(j)} = \boldsymbol{u}_{i-1} + \frac{\boldsymbol{\delta}_i}{2^j}, \quad j = 1, 2, \ldots, \tag{18}$$

computing $Q(\boldsymbol{u}_i^{(j)})$ at each step. The step-halving procedure continues until a value of j is found such that $Q(\boldsymbol{u}_i^{(j)}) < Q(\boldsymbol{u}_{i-1})$, which the iterate is then updated by setting $\boldsymbol{u}_i = \boldsymbol{u}_i^{(j)}$. If no such j is found after a predetermined number of halvings, then the step-halving procedure has failed and the algorithm terminates.

The minimizer, $\boldsymbol{\delta}_i$, of (16) satisfies

$$\boldsymbol{P}\left(\boldsymbol{\Lambda}_\theta'\boldsymbol{Z}'\boldsymbol{M}_i\boldsymbol{W}\boldsymbol{M}_i\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q\right)\boldsymbol{P}'\boldsymbol{\delta}_i = \boldsymbol{\Lambda}_\theta'\boldsymbol{Z}'\boldsymbol{M}_i\boldsymbol{W}(\boldsymbol{y}_{\text{obs}} - \boldsymbol{\mu}_i) - \boldsymbol{u}_i \tag{19}$$

which we solve using the sparse Cholesky factor. At convergence, the factor, $\boldsymbol{L}_{\beta,\theta}$, satisfies

$$\boldsymbol{L}_{\beta,\theta}\boldsymbol{L}_{\beta,\theta}' = \boldsymbol{P}\left(\boldsymbol{\Lambda}_\theta'\boldsymbol{Z}'\boldsymbol{M}\boldsymbol{W}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q\right)\boldsymbol{P}' \tag{20}$$

As we show in the next section, the matrix $(\boldsymbol{L}_{\beta,\theta}\boldsymbol{L}_{\beta,\theta}')^{-1}$ is a Laplace approximation of the covariance matrix for the spherical random effects, conditional on the observed data. This fact is useful for constructing a nonlinear objective function for finding the approximate maximum likelihood estimates of θ and β.

2.2. Evaluating the likelihood for GLMMs using the Laplace approximation

Evaluating the likelihood for generalized linear mixed models requires approximating an intractable integral over the random effects distribution. The `glmer` function offers several approximations, controlled by the `nAGQ` argument. The default value of `nAGQ=1` specifies the *Laplace approximation* (Madsen and Thyregod 2011).

A second-order Taylor series approximation to $-2\log[f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})]$ based at $\tilde{\boldsymbol{u}}$ provides an approximation of the unscaled conditional density as a multiple of the density for the multivariate Gaussian $\mathcal{N}(\tilde{\boldsymbol{u}}, \boldsymbol{L}\boldsymbol{L}')$. The change of variable

$$\boldsymbol{u} = \tilde{\boldsymbol{u}} + \boldsymbol{L}\boldsymbol{z} \tag{21}$$

provides

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) &= \int_{\mathbb{R}^q} f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}}, \boldsymbol{u})\, d\boldsymbol{u} \\ &\approx \tilde{f}\,|\boldsymbol{L}| \int_{\mathbb{R}^q} e^{-\|\boldsymbol{z}\|^2/2}\,(2\pi)^{-q/2}\, d\boldsymbol{z} \\ &= \tilde{f}\,|\boldsymbol{L}| \end{aligned} \tag{22}$$

or, on the deviance scale,

$$-2\ell(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) \approx \sum_{i=1}^{n} d_i(\boldsymbol{y}_{\text{obs}},\tilde{\boldsymbol{u}}) + \|\tilde{\boldsymbol{u}}\|^2 + \log(|\boldsymbol{L}|^2) + \frac{q}{2}\log(2\pi) \tag{23}$$

The Laplace approximation normally conditions on both the fixed effects β and the variance-covariance parameters θ. A further approximation, which is denoted in `glmer` by `nAGQ=0`, profiles out the fixed effects by minimizing β and the conditional modes $\boldsymbol{u}$ simultaneously in eq. 15. This approximation is exact when (1) $\partial(\log L)/\partial\beta$ is a linear function of the conditional modes $\boldsymbol{u}$ and (2) when the conditional mode is equal to the conditional mean (typically, although not necessarily, implying a symmetric conditional distribution). Both assumptions hold for linear mixed models (although a Laplace approximation is not necessary there), consistent with Bates *et al.* (2015) showing that the fixed effects can be profiled out of the log-likelihood for LMMs. The Julia `MixedModels.jl` package offers the same approximation as the `fast` argument to the `pirls!` function (`https://juliastats.org/MixedModels.jl/stable/optimization/`); Template Model Builder (Kristensen, Nielsen, Berg, Skaug, and Bell 2016), and downstream packages such as `glmmTMB`, provide this function via a `profile` argument.

By default, `glmer` uses a two-stage optimization procedure (described below) with `nAGQ=0` in the first stage; users can also specify `nAGQ=0` for faster, approximate model fits.

Decomposing the deviance for simple models

A common special case of mixed models is those where scalar (typically intercept) random effects are associated with levels of a single grouping factor, $\boldsymbol{h}$. In this case the dimension, q, of the random effects is the number of levels of $\boldsymbol{h}$ — i.e. there is exactly one random effect associated with each level of $\boldsymbol{h}$. We will write the vector of variance-covariance parameters, which is one-dimensional, as a scalar, θ. The matrix $\boldsymbol{\Lambda_\theta}$ is a multiple of the identity, $\theta\boldsymbol{I}_q$, and $\boldsymbol{Z}$ is the $n \times q$ matrix of indicators of the levels of $\boldsymbol{f}$. The permutation matrix, $\boldsymbol{P}$, can be set to the identity and $\boldsymbol{L}$ is diagonal, although not necessarily homogeneous (i.e., a scalar multiple of the identity matrix).

Because each element of $\boldsymbol{\mu}$ depends on only one element of $\boldsymbol{u}$ and the elements of $\mathcal{Y}$ are conditionally independent, given $\mathcal{U} = \boldsymbol{u}$, the conditional densities of the $u_j, j = 1,\dots,q$ given $\mathcal{Y} = \boldsymbol{y}_{\text{obs}}$ are independent. We partition the indices $1,\dots,n$ as $\mathbb{I}_j, j = 1,\dots,q$ according to the levels of $\boldsymbol{h}$. That is, the index i is in $\mathbb{I}_j$ if $h_i = j$. This partitioning also applies to the deviance residuals in that the ith deviance residual depends only on u_j when $i \in \mathbb{I}_j$.

Writing the univariate conditional densities as

$$f_j(\boldsymbol{y}_{\text{obs}},u_j) = \exp\left(-\frac{\sum_{i\in\mathbb{I}_j} d_i(\boldsymbol{y}_{\text{obs}},u_j) + u_j^2}{2}\right)(2\pi)^{-1/2} \tag{24}$$

we have

$$f_{\mathcal{Y},\mathcal{U}}(\boldsymbol{y}_{\text{obs}},\boldsymbol{u}) = \prod_{j=1}^{q} f_j(\boldsymbol{y}_{\text{obs}},u_j) \tag{25}$$

and

$$L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) = \prod_{j=1}^{q}\int_{\mathbb{R}} f_j(\boldsymbol{y}_{\text{obs}},u)\,du \tag{26}$$

We consider this special case both because it occurs frequently and because, for some software, it is the only type of GLMM that can be fit. Also, in this particular case we can graphically assess the quality of the Laplace approximation by comparing the actual integrand to its approximation.

Consider the `cbpp` data on contagious bovine pleuropneumonia (CBPP) incidence according to season and herd, available in the **lme4** package (see 5.1 for more details), and the model

```
> print(m1 <- glmer(cbind(incidence, size-incidence) ~ period + (1|herd),
+       cbpp, binomial), corr=FALSE)

Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(incidence, size - incidence) ~ period + (1 | herd)
   Data: cbpp
      AIC       BIC    logLik -2*log(L)  df.resid
 194.0531  204.1799  -92.0266  184.0531        51
Random effects:
 Groups Name        Std.Dev.
 herd   (Intercept) 0.6421
Number of obs: 56, groups:  herd, 15
Fixed Effects:
(Intercept)      period2      period3      period4
    -1.3983      -0.9919      -1.1282      -1.5797
```

This model has been fit by minimizing the Laplace approximation to the deviance. We can assess the quality of this approximation by evaluating the unscaled conditional density at $u_j(z) = \tilde{u_j} + z/\boldsymbol{L}_{j,j}$ and comparing the ratio, $f_j(\boldsymbol{y}_{\text{obs}}, u)/(\tilde{f}_j\sqrt{2\pi})$, to the standard normal density, $\phi(z) = e^{-z^2/2}/\sqrt{2\pi}$, as shown in Figure 1.

As Figure 1 shows, the univariate integrands are very close to the standard normal density, indicating that the Laplace approximation to the deviance is a good approximation in this case.

3. Adaptive Gauss-Hermite quadrature for GLMMs

When the integral (12) can be expressed as a product of low-dimensional integrals, we can use Gauss-Hermite quadrature to provide a closer approximation to the integral. Univariate Gauss-Hermite quadrature evaluates the integral of a function that is multiplied by a "kernel" where the kernel is a multiple of e^{-z^2} or $e^{-z^2/2}$. For statisticians the natural candidate is the standard normal density, $\phi(z) = e^{-z^2/2}/\sqrt{(2\pi)}$. A kth-order Gauss-Hermite formula provides knots, $z_i, i = 1, ..., k$, and weights, $w_i, i = 1, \ldots, k$, such that

$$\int_{\mathbb{R}} t(z)\phi(z)\,dz \approx \sum_{i=1}^{k} w_i t(z_i)$$

The function `GHrule` in **lme4** (based on code in the **SparseGrid** package) provides knots and weights relative to the standard normal kernel for orders k from 1 to 100. For example,

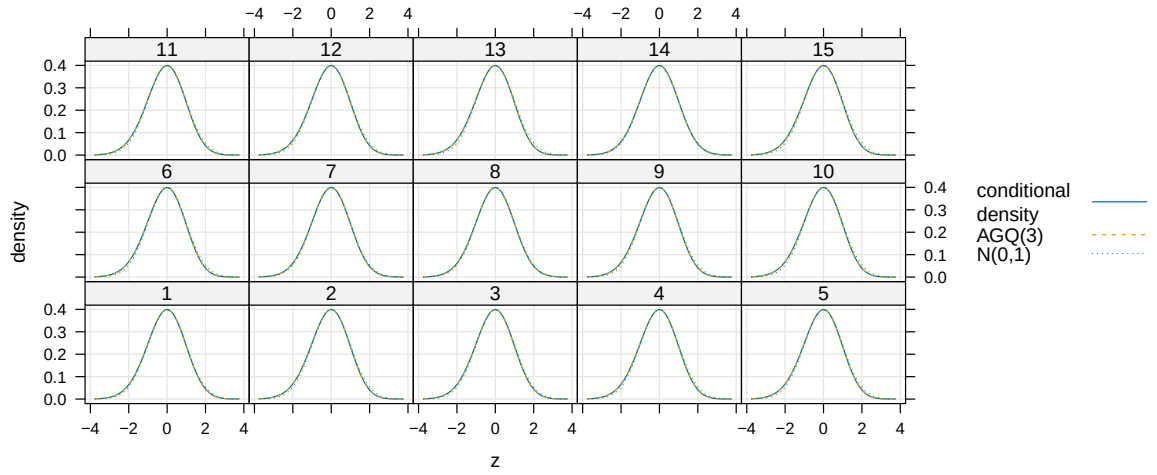

Figure 1: Comparison of univariate integrands (solid, blue line); 3-point Gauss-Hermite quadrature (dashed, yellow); and standard normal density function (green, dotted) for the CBPP model. For this model, the three approximations are nearly indistinguishable.

```
> GHrule(5)

             z          w     ldnorm
[1,] -2.856970 0.01125741 -5.0000774
[2,] -1.355626 0.22207592 -1.8377997
[3,]  0.000000 0.53333333 -0.9189385
[4,]  1.355626 0.22207592 -1.8377997
[5,]  2.856970 0.01125741 -5.0000774
```

where `z` is the vector of knots, `w` is the vector of weights, and `dlnorm` is the log-density of the standard normal distribution at z.

The choice of the value of k depends on the behavior of the function $t(z)$. If $t(z)$ is a polynomial of degree $2k-1$ then the Gauss-Hermite formula for orders k or greater provides an exact answer. The fact that we want $t(z)$ to behave like a low-order polynomial is often neglected in the formulation of a Gauss-Hermite approximation to a quadrature. The quadrature knots on the u scale are chosen as

$$u_{i,j}(z) = \tilde{u_j} + z_i/\boldsymbol{L}_{j,j}, \quad i = 1, \ldots, k;\ j = 1, \ldots, q \tag{27}$$

exactly so that the function $t(z)$ should behave like a low-order polynomial over the region of interest, which is to say the region where quadrature knots with large weights are located. The term "adaptive Gauss-Hermite quadrature" reflects the fact that the approximating Gaussian density is scaled and shifted to provide a second order approximation to the logarithm of the unscaled conditional density.

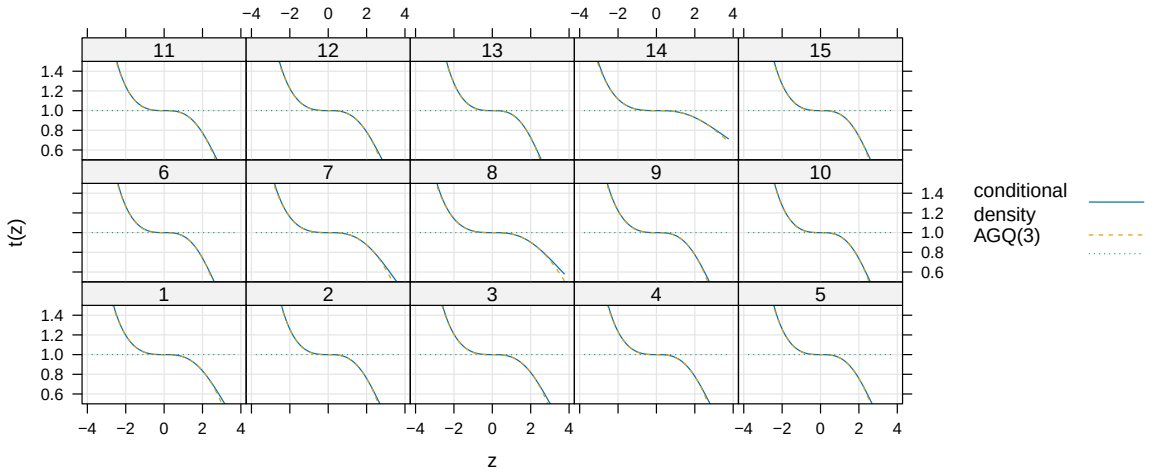

Figure 2: The function $t(z)$, which is the ratio of the normalized unscaled conditional density to the standard normal density, for each of the univariate integrals in the evaluation of the deviance for model `m1` (blue, solid line). As in Figure 1, the dashed yellow line shows the approximation for 3-point adaptive Gauss-Hermite quadrature. The horizontal green reference line ($z = 1$) gives the reference value that would apply for the Laplace approximation, which assumes the conditional density is exactly equal to the standard normal. The y-axis limits are truncated to (0.5, 1.5).

Figure 2 shows $t(z)$ for each of the unidimensional integrals in the likelihood for the model `m1` at the parameter estimates. While this view shows the deficiency of the Laplace approximation (deviation from a horizontal reference line at $z = 1$ clearly, The AGQ(3) approximation fits extremely well over the range shown. The tails of the polynomials implied by the AGQ approximation can fluctuate widely, but these fluctuations are suppressed by multiplying by the thin tails of the Gaussian distribution.

The CBPP data set is a relatively well-behaved data set, where Laplace approximation works well. In contrast, a widely used data set on toenail onychomycosis (De Backer, De Vroey, Lesaffre, Scheys, and De Keyser 1998), which has a very low effective sample size per cluster — an average of about 6.5 binary observations ("moderate or severe" vs "none or mild" disease) per patient — represents an example where Gauss-Hermite quadrature is necessary for reliable results. In this case the conditional densities depart much more clearly from the standard normal (Figure 3; note the scale of the density ratios goes from 0 to 10, in contrast the maximum of 4 in Figure 2). Stringer, Bilodeau, and Tang (2022) and Stringer (2024) further explore the limitations of Laplace approximation.

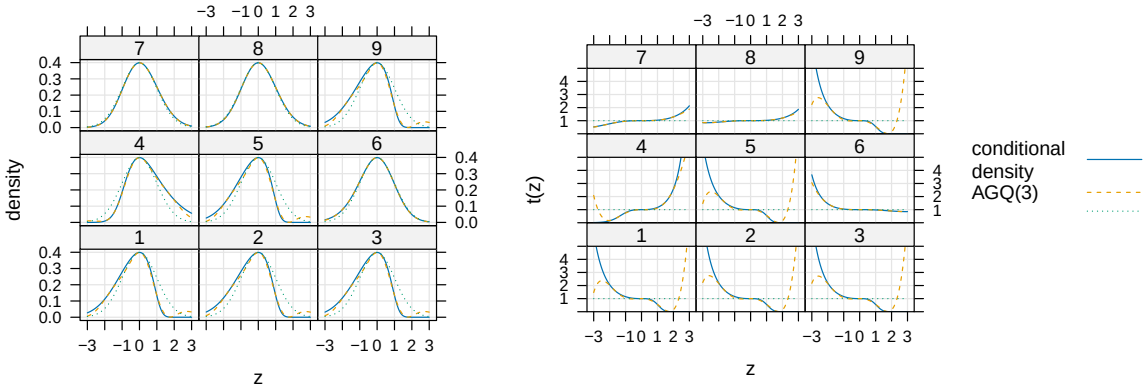

Figure 3: Normalized unscaled conditional density (left) and ratio of density to the standard normal density (right) for a random sample of 9 patients from the toenail onychomycosis data set. The y-axis limits in the right panel ($t(z)$) are truncated to $(0, 5)$.

To use adaptive Gauss-Hermite quadrature for model fitting in `glmer` models, set the argument `nAGQ`, the number of quadrature points, to a value greater than 1. Increasing the number of nodes generally improves the accuracy of the likelihood approximation at the expense of computation time — although rounding errors may accumulate when using large numbers of quadrature points. At present, AGQ is only available for models with a single scalar random effect. (The `GLMMadaptive` package implements AGQ for vector-valued random effect models, although it is still restricted to models with a single random effect.)

4. Model fitting

Once we can calculate the deviance by PIRLS for specified values of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ (or only $\boldsymbol{\theta}$ if profiling out the fixed effects via `nAGQ=0`), we then estimate the parameters by nonlinear optimization. This procedure largely follows the description in Bates *et al.* (2015), using derivative-free optimizers with box constraints to prevent non-positive-(semi)definite covariance matrices. Specifically, the elements of $\boldsymbol{\theta}$ corresponding to the diagonal of $\boldsymbol{\Lambda}_\theta$ are currently constrained to be non-negative.

The only difference is that by default `glmer` uses a two-step fitting procedure, using `nAGQ=0` at the first stage to get preliminary estimates which are then used as starting points for a second optimization with Laplace approximation or Gauss-Hermite quadrature as specified by the user. Different nonlinear optimizers can be used at each stage: the current default, as specified in `glmerControl`, is to use Powell's BOBYQA followed by a box-constrained variant of the Nelder-Mead simplex algorithm. For faster, approximate fitting, the second stage can be omitted; in the rare cases where the initial `nAGQ=0` fit gives poor results, the first stage can be skipped via `glmerControl(nAGQ0initStep = FALSE)`.

However, because the profiled log-likelihood is an even function of the diagonal elements of $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}_\theta = \boldsymbol{\Lambda}_\theta \boldsymbol{\Lambda}_\theta'$ depends on them only through their squares, positive and negative values yield identical likelihoods. This symmetry means that constrained optimization is not strictly necessary — an unconstrained optimizer will converge to a correct solution, approaching zero from either side in the boundary case of a singular random-effects covariance matrix. (Once a solution is found, we can map it to a unique solution where the diagonal elements are all non-negative.) A similar approach to removing constraints could work for structured covariance matrices where correlation parameters are constrained to $(-1, 1)$, e.g. by parameterizing the model in terms of a phase parameter p where $\rho = \sin(p)$ (and then mapping p to $(0, 2\pi)$). Removing these constraint would permit the use of a broader class of unconstrained optimizers, though this feature has not yet been incorporated into `lme4`.

5. Examples

5.1. CBPP

The `?cbpp` help page describes the CBPP data set (Lesnoff *et al.* 2004) as follows:

> 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).

Lesnoff *et al.* (2004) estimated the effects of different treatments using (1) ordinary logistic regression incorporating a variance-inflation factor, also known as a quasi-binomial model ("logistic regression" is sometimes used specifically to describe analyses of Bernoulli responses, but in this case there are multiple trials per observation [cows that could become seropositive], and so a dispersion or scale parameter can be estimated); (2) a GLMM implemented in `lme4`; and a (3) Markov chain Monte Carlo algorithm Zeger and Karim (1991), which as they state allows for a non-parametric rather than a Normal model for the random effects. The authors did not find any significant effects of treatment, ascribing the null results to "a lack of power in the statistical analyses or to a quality problem for the medications used (and more generally, for health-care delivery in the Boji district)."

(Note that Table 1 of Lesnoff *et al.* (2004) contains a known typographical error for herd 6. Consequently, results obtained using the `cbpp` data set may not exactly reproduce some of the findings reported in that paper.)

The `lme4` package includes two variants of the `cbpp` data set. The second variant, `cbpp2`, contains corrected values corresponding to Table 1 of Lesnoff *et al.* (2004). Although the precise provenance of these data sets is unclear, the `cbpp` data set matches the version held by the corresponding author of Lesnoff *et al.* (2004).

We model proportional incidence as a binomial response depending on the additive fixed effects of period, treatment and average herd size; to account for repeated measures we fit a model with a random effect of herd. In practice we might also be interested in a period by treatment interaction, but we neglect that term here.

As in `glm`, we can specify a binomial response as a proportion and use the `weights` argument to specify the sample size, instead of the more typical two-column `cbind(successes,failures)` format:

```
> gm1 <- glmer(incidence/size ~ period + treatment + avg_size + (1 | herd),
+              family = binomial,
+              data = cbpp2, weights = size)
```

It is also worth considering adding an observation-level random effect to the model, which we can do by creating a new factor based on observation number and using `update()` on the previous model:

```
> cbpp2 <- transform(cbpp2,obs=factor(seq(nrow(cbpp2))))
> gm2 <- update(gm1,.~.+(1|obs))  ## herd and observation-level REs
> gm3 <- update(gm1,.~.-(1|herd)+(1|obs))  ## observation-level REs only
```

Model summary

The first part of the summary reiterates the family and link function used, the model formula,

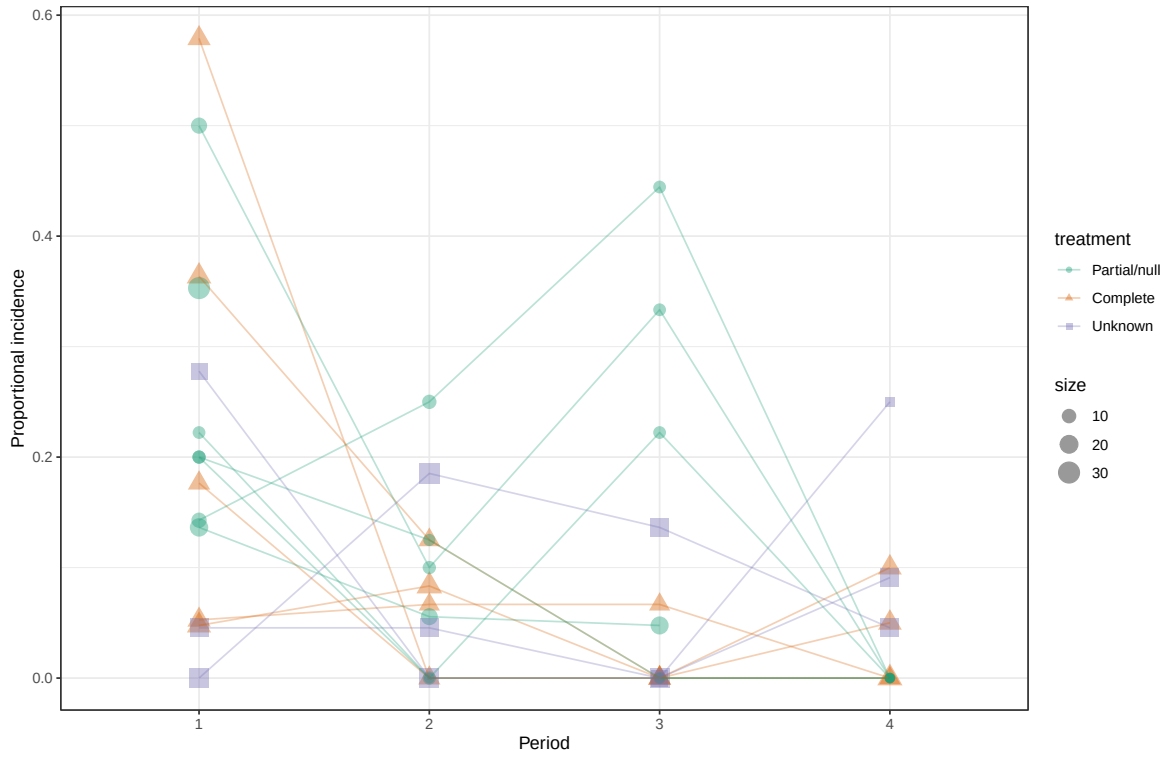

Figure 4: Incidence (proportion of cows becoming seropositive per observation period) vs. period. Colours show treatment category for each herd; point sizes reflect the number of seronegative cows at the start of each period. Lines connect the sets of observations from each herd.

and gives various summary statistics (log-likelihood etc.), as well as quantiles of the scaled (Pearson) residuals:

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: incidence/size ~ period + treatment + avg_size + (1 | herd)
   Data: cbpp2
Weights: size

      AIC       BIC    logLik -2*log(L)  df.resid
    197.8     214.0     -90.9     181.8        48

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.2311 -0.7967 -0.3732  0.4684  2.7557
```

These quantities are also accessible via standard accessors (`AIC()`, `BIC()`, `logLik()`).

The next chunk of `summary()` describes the random effects and the number of levels associated with each grouping factor (the latter is useful for checking that random-effects formulae have been specified correctly):

```
Random effects:
 Groups Name        Variance Std.Dev.
 herd   (Intercept) 0.3116   0.5582
Number of obs: 56, groups:  herd, 15
```

This information is also accessible via `VarCorr()`, which returns a list of variance-covariance matrices (the `print` method for `VarCorr` objects allows control of whether the variance, or standard deviation, or both, are printed).

Next come the estimates of the fixed effects, along with Wald estimates of the standard error, Z statistic, and p-value:

```
Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.005623   0.708418  -1.420 0.155743
period2           -0.986283   0.303381  -3.251 0.001150
period3           -1.125147   0.323142  -3.482 0.000498
period4           -1.561098   0.422631  -3.694 0.000221
treatmentComplete -0.376225   0.503000  -0.748 0.454483
treatmentUnknown  -0.683246   0.645179  -1.059 0.289599
avg_size          -0.006135   0.045608  -0.135 0.893002
```

One can use `coef(summary())` to retrieve this information, and optionally format it with `printCoefmat()`.

The last component of `summary()` gives the estimated correlations among the fixed-effect parameters, which can be useful for assessing multicollinearity (it can also be overwhelming:

it is suppressed by default for models with more than 20 fixed-effect parameters, and can also be suppressed by using `print(summary(.),correlation=FALSE)`).

```
Correlation of Fixed Effects:
            (Intr) perid2 perid3 perid4 trtmnC trtmnU
period2     -0.135
period3     -0.130  0.278
period4     -0.086  0.210  0.195
trtmntCmplt  0.289 -0.017 -0.021 -0.059
trtmntUnknw  0.431 -0.053 -0.045 -0.043  0.588
avg_size    -0.910  0.026  0.028  0.020 -0.547 -0.649
```

Diagnostics

A range of graphical diagnostic tools is available for `merMod` objects. The plot methods in the `lme4` package are inspired by those in the `nlme` package, using `lattice` plots to provide a reasonable blend of convenience and flexibility.

`merMod` objects are also compatible with the `performance` package and the `DHARMa` package, both commonly used for model checking.

The following code produces a standard range of diagnostic plots (Figure 5), similar to the ones in base R's `plot.lm` method. These diagnostics will generally be useful for models where the conditional density is approximately normal (but heteroscedastic) — e.g., Poisson responses with large mean or binomial responses with large numbers of successes — and less so otherwise, e.g. for binary responses.

```
> ## basic residual plot
> plot(gm1)
> ## scale-location plot
> plot(gm1,sqrt(abs(resid(.)))~fitted(.),type=c("p","smooth"))
> ## boxplot of residuals grouped by a categorical predictor
> plot(gm1,period~resid(.))
> ## Q-Q plot
> qqmath(gm1)
```

The `ranef()` accessor extracts the conditional modes; the argument `condVar=TRUE` additionally extracts the variances of the conditional modes, which are stored as an attribute labelled `"postVar"` — a three-dimensional array that gives the variance-covariance matrix of the conditional modes for each level of the grouping variable. The plotting methods `dotplot()` and `qqmath()` return lists of graphical objects showing *caterpillar plots* (ordered values of the random effects with confidence bars); in the case of the Q-Q plot (`qqmath`) the y-axis shows corresponding values of the standard normal quantiles (Figure 6).

`performance::check_model()` checks a variety of classical assumptions of GLMs. For mixed-effect models, it also assesses the normality of the distribution of conditional modes (Figure 7).

The `DHARMa` package is also compatible with `merMod` objects. `DHARMa` generates simulation-based residuals for generalized linear (mixed) models and uses them for graphical and statistical tests of model assumptions (Figure 8).

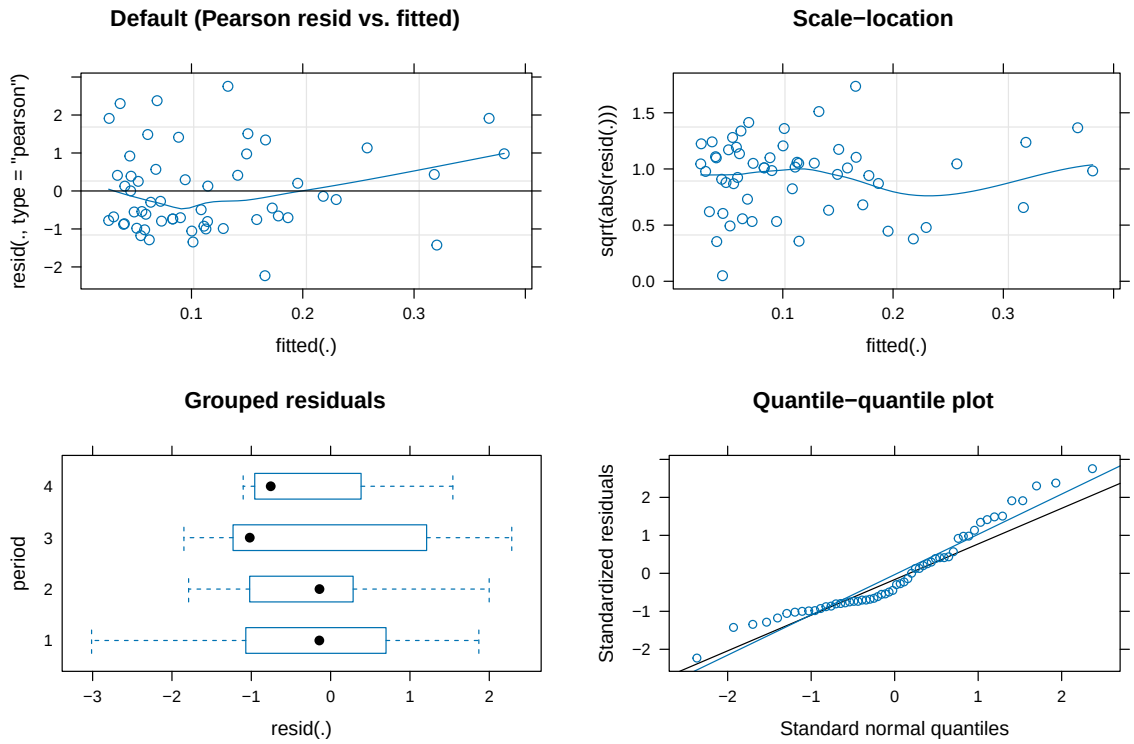

Figure 5: Graphical diagnostics (using package built-in functions)

Having checked the diagnostics, we would now like to compare the three models we have fitted. Inspecting the `VarCorr` components, we see that when we fit both herd- and observation-level random effects, the among-herd variance is estimated as zero. The appropriate procedure at this point (e.g. whether one drops non-significant terms, or those with small scaled magnitudes, or those that worsen the AIC or BIC of the model) depends on the goals of the analysis and one's philosophy of model-building (Barr, Levy, Scheepers, and Tily 2013; Matuschek, Kliegl, Vasishth, Baayen, and Bates 2017; Scandola and Tidoni 2024).

One might either stick with the full model, or continue with the reduced model with observation-level random effects only (as it has exactly the same likelihood as the full model but uses an additional parameter, it would be chosen according to either an information-theoretic or a hypothesis-testing model selection framework).

Here we will start by computing likelihood profiles and confidence intervals for the model incorporating both random effects; although it has the same point estimates and maximum likelihood as the reduced model, confidence intervals that incorporate non-local information (i.e. profile- or parametric bootstrap-based) will give different, more conservative results for the full model.

The `profile` method computes profile likelihoods. The computation can be slow, since complete profiling for a model with p random- and fixed-effect parameters requires fitting p profiles, each of which requires many $p-1$-dimensional optimizations. The machinery for generating likelihood profiles for GLMMs is similar to that for LMMs (see Bates *et al.* (2015), § 5.1).

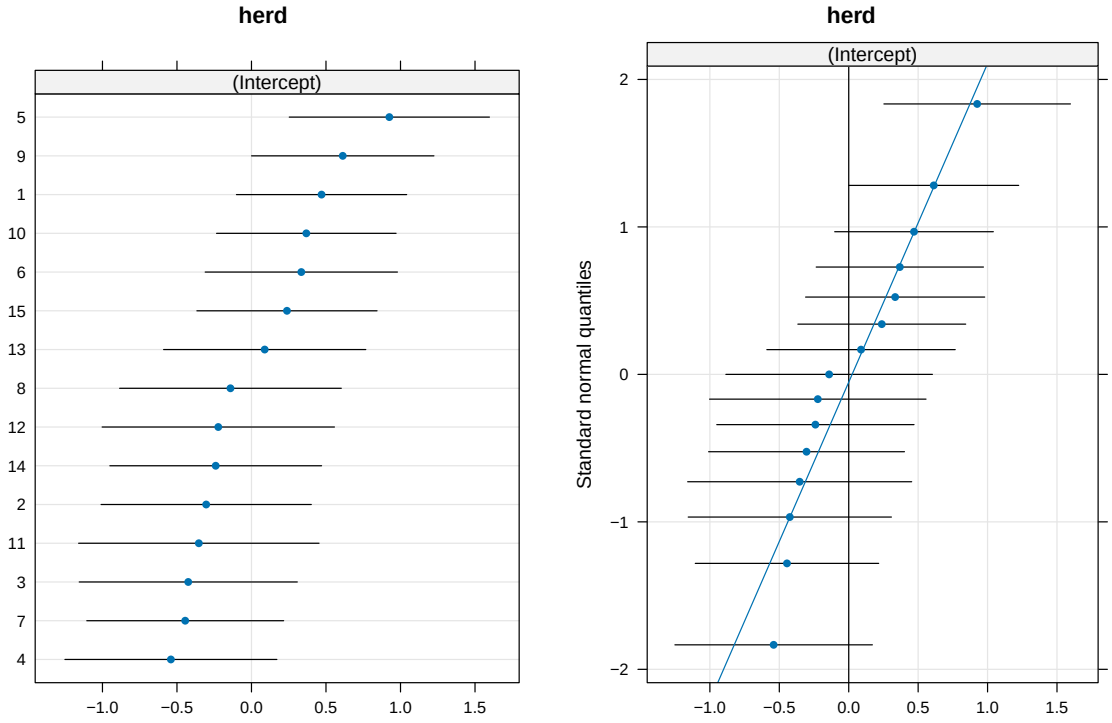

Figure 6: Graphical display of random effects. *Left*: conditional modes $\pm 1.96 \times$ conditional standard deviation, ordered by magnitude. *Right*: quantile-quantile plot, with linear regression line overlaid.

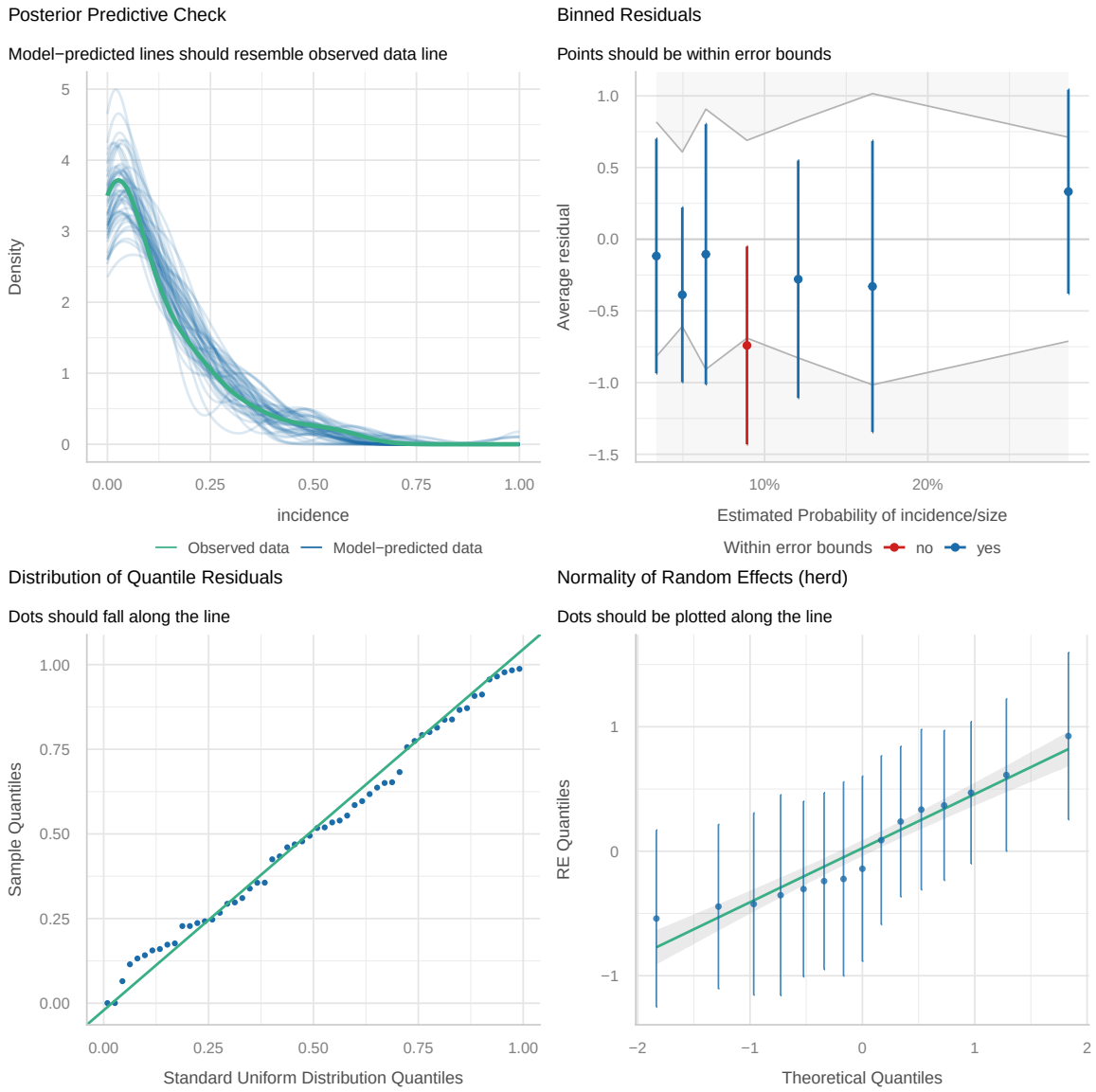

Figure 7: Model diagnostics using `performance::check_model`. The plot showcasing the normality of random effects (lower right panel) is the transpose of the right panel in Figure 6.

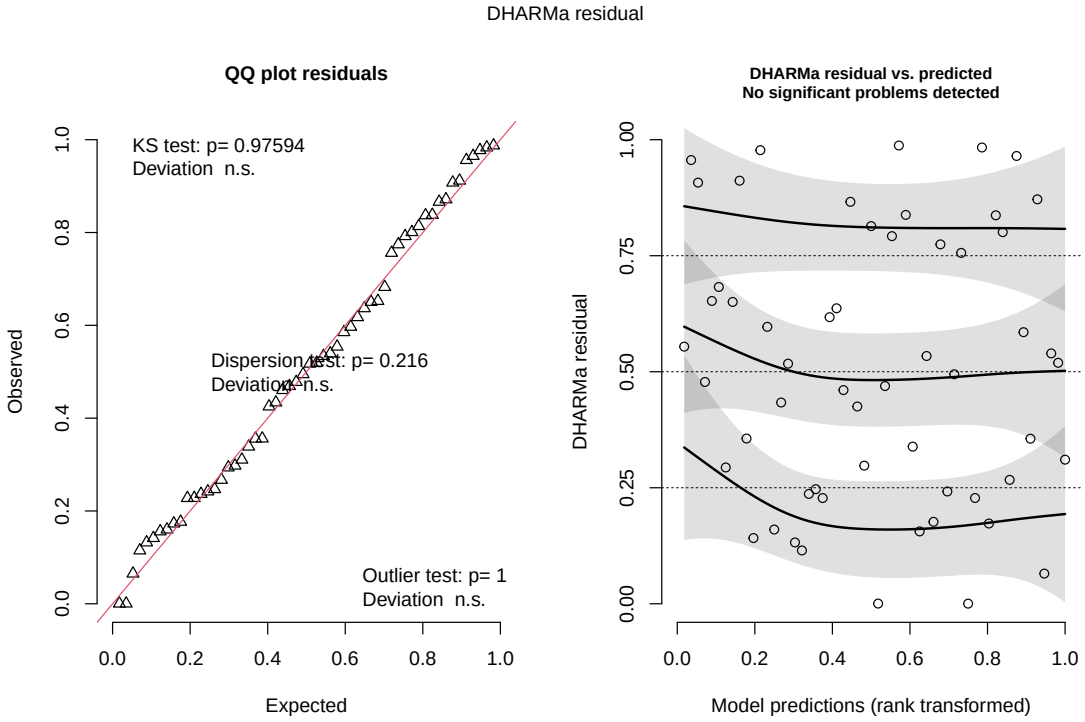

Figure 8: Model assumption checks using the DHARMa package. `n.s.` denotes "not significant".

The `profile` method returns an object of class `thpr` — a data frame containing the profiles, augmented with attributes containing interpolation splines for each parameter profile and their inverses (using `splines::interpSpline` and `splines::backSpline`); the latter are used for plotting profiles and computing confidence intervals. An `as.data.frame` method adds `.focal` and `.par` variables to the data frame, useful for customized plots.

Profiles can be used for univariate (`xyplot`) and bivariate (`splom`) profile plots, and to compute profile confidence intervals (`confint`). (`confint` applied to a `glmer` fit will first fit the profile, then use it to compute profile confidence intervals. Given the computational cost of profiling, it makes sense to compute and save the profile as an intermediate step if one plans to do anything other than computing confidence intervals.)

Two other common methods for computing confidence intervals are parametric bootstrapping (`method = "boot"`) and the classical Wald approximation (`method = "Wald"`). Parametric bootstrapping is much slower, but more accurate (and, via the `FUN` argument, can generate confidence intervals for any quantity that can be derived from a fitted model). The Wald approximation is faster and less accurate than profile confidence intervals. By default `glmer` only returns estimates for the fixed-effects parameters, as the assumptions of the Wald approximation are often violated badly for random-effects (co)variances and correlations. In the examples below we use the finite-difference Hessian (second derivative matrix of the estimated parameters) and the delta method to compute Wald confidence intervals for random-effects standard deviations and correlations, when possible.

Figure 9 compares all three of these confidence intervals across all three of the models fitted.

If the default optimizers (Nelder-Mead followed by BOBYQA) do not perform well, one could attempt to re-fit a [g]lmer model with a variety of different optimizers using `allFit()`. To see which optimizers `lme4` currently supports, one can use the command `allFit(show.meth.tab=TRUE)` to show all of the different methods.

To use `allFit()`, supply the initial fitted model as input. Useful results are obtained from `summary(allFit())`:

- `$which.OK` returns which optimizers worked. Below only applies to optimizers that were successful.
- `$llik` returns log-likelihoods
- `$fixef` fixed-effect estimates
- `$sdcor` random-effect standard deviations and correlations
- `$theta` random-effect parameters on the Cholesky scale

We will show the summary results for `$sdcor`; the other components are similar.

```
> gm_all <- allFit(gm1)
```

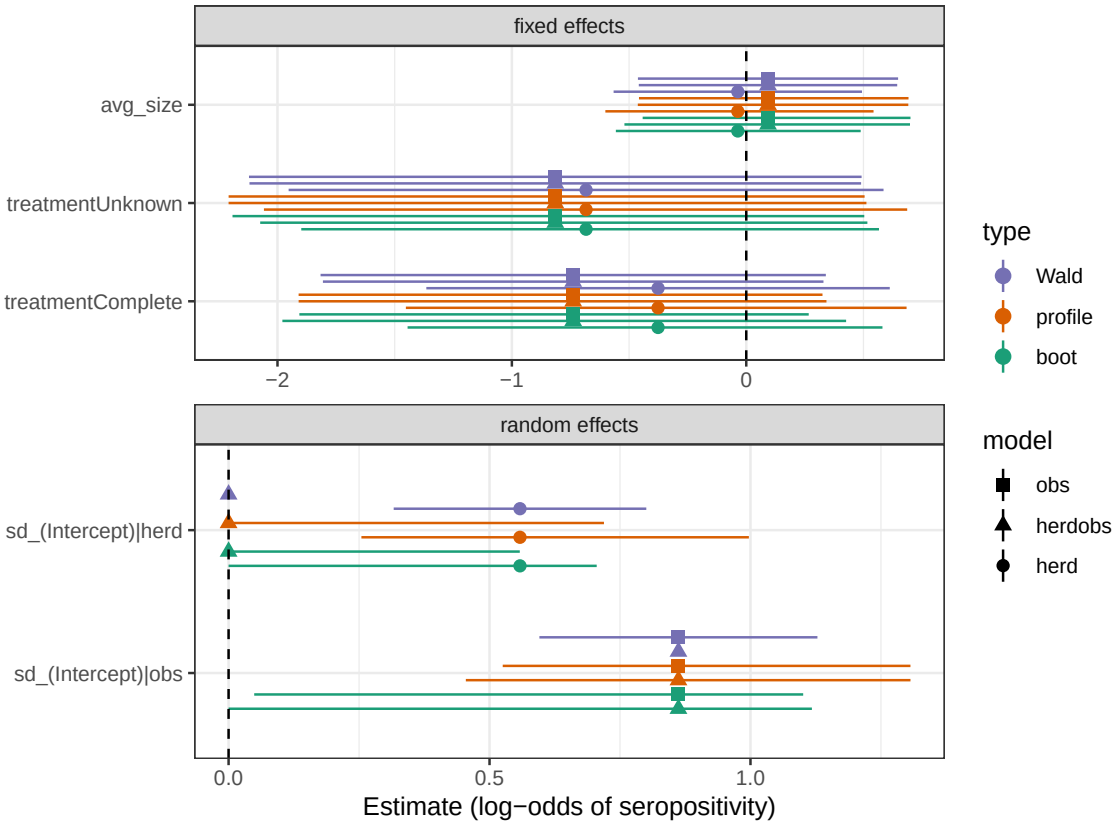

Figure 9: CBPP example: comparison of point and confidence interval estimation for different methods. Wald CIs are missing for random-effects parameters when the fitted model is singular.

```
bobyqa : [OK]
Nelder_Mead : [OK]
nlminbwrap : [OK]
nmkbw :
[OK]
optimx.L-BFGS-B : [OK]
nloptwrap.NLOPT_LN_NELDERMEAD : [OK]
nloptwrap.NLOPT_LN_BOBYQA : [OK]
```

```
> ss <- summary(gm_all)
> ss$sdcor

                               herd.(Intercept)
bobyqa                                0.5581743
Nelder_Mead                           0.5581777
nlminbwrap                            0.5581814
nmkbw                                 0.5580124
optimx.L-BFGS-B                       0.5581753
nloptwrap.NLOPT_LN_NELDERMEAD         0.5581391
nloptwrap.NLOPT_LN_BOBYQA             0.5581637
```

As of version 2.0, `lme4` can also specify structured variance-covariance matrices for (generalized) linear mixed models. `lme4` now supports unstructured (general positive definite), diagonal, compound symmetry, and first-order autoregressive (AR1) structures. By default, AR1 models assume a homogeneous-variance model (the variance is the same for all time steps), while the other models assume heterogeneous-variance models (variances differ for every level of the varying term); users can adjust this with the `hom` argument (e.g. `ar1(..., hom = FALSE)`).

The unstructured covariance structure is the default for mixed models. Here we illustrate fitting an AR1 model; the next example will show diagonal and compound symmetric models.

```
> gm.ar1 <- glmer(incidence/size ~ ar1(1 + herd | period),
+                 family = binomial,
+                 data = cbpp, weights = size)
> print(VarCorr(gm.ar1))

 Groups Name        Std.Dev. Corr
 period (Intercept) 0.78959  -0.041 (ar1)
```

For this particular model, a heterogeneous AR1 model (`ar1(..., hom = FALSE)`) results in a singular fit.

See the lme4 covariance structures vignette for more detail.

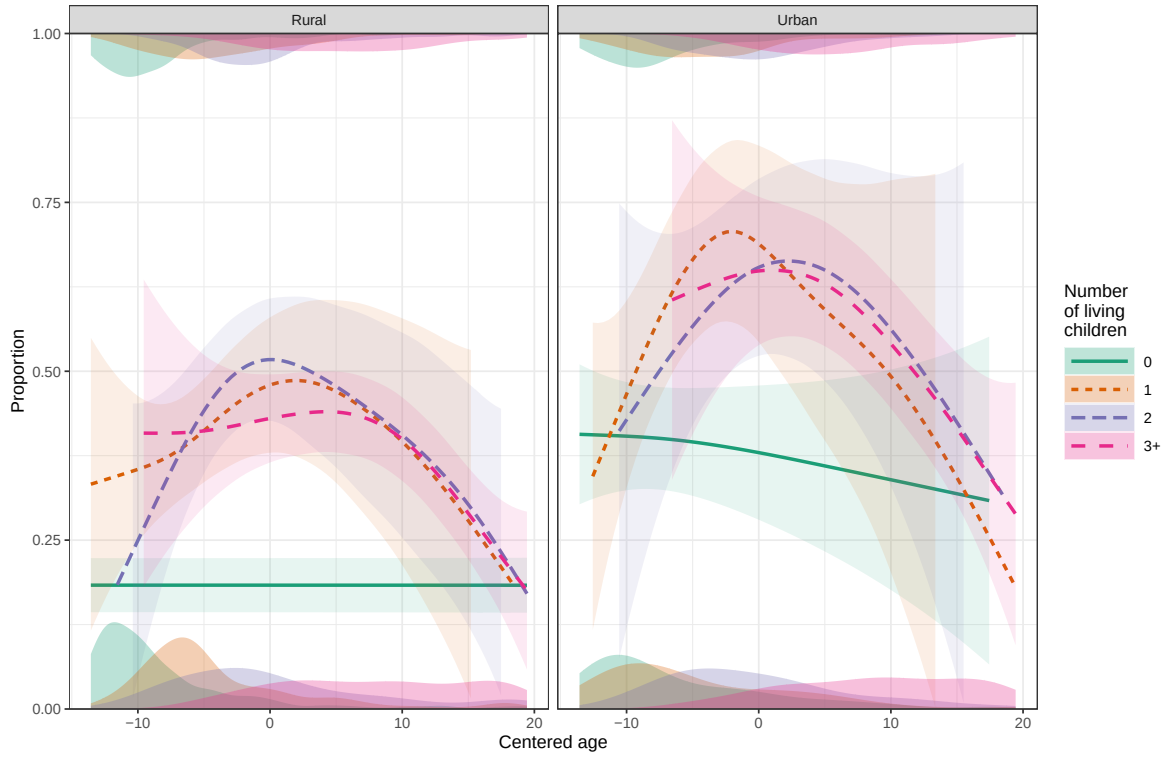

Figure 10: Contraception example: proportion of contraceptive use by centered age, number of living children (line type), and urban/rural residence (facet). Curves fitted using generalized additive models with cubic spline smoothing. Density plots along bottom and top margins show the age distributions of women not using contraception (bottom) and using contraception (top).

5.2. Contraception

Huq and Cleland (1990) use multilevel models to analyze data from a fertility survey of women in Bangladesh. These data are available as the `Contraception` object in the **mlmRev** package. The response variable is binary and indicates whether or not each woman was using contraception at the time of the survey. Covariates included the woman's age, the number of live children she had, whether she lived in an urban or rural setting, and the district in which she lived.

Figure 10 shows exploratory smooth curves of contraceptive use as a function of centered age, stratified by number of living children and urban/rural residence. In rural areas, women with two living children show the highest rates of contraceptive use, peaking near 50% at the average age, while women in urban areas with one living child show the highest rates near the average age but with a more pronounced decline at older ages. In both settings, women with no living children consistently show the lowest rates of contraceptive use. The nonmonotonic effect of age on contraceptive use and the apparent dependence of this age trend on child status motivate the model specifications explored below. (These nonmonotonic trends were not noticed in the original analysis of the data, which concluded that there was no significant effect of age.)

	df	Δnegloglik	ΔAIC
binary_child × age + (1 | district:urban)	7	0.472	0.00
binary_child × age + (1 | district/urban)	8	0.467	1.99
binary_child × age + (1 + urban | district)	9	0.000	3.06
binary_child × age + (1 | district)	7	5.825	10.71
binary_child + age + (1 | district)	6	9.828	16.71
int_child + age + (1 | district)	8	9.599	20.25

Table 1: Model comparison for Contraception fits. Note that for likelihood ratio tests, or for AIC comparisons restricted to nested models (Ripley 2004), the nesting sequence for the random-effects models is `(1+urban|district)` > `(1|district/urban)` > {`(1|district)`, `(1|district:urban)`}.

We construct six `glmer` models with varying choices of explanatory variables and random effects structure, comparing them via `anova`. The models considered different variables.

For instance, there are two different ways to encode the number of living children: `livch`, a four-level factor distinguishing `0`, `1`, `2`, or `3+` children, while other models use `ch` instead (later we label as `binary_child`), which is a binary indicator for whether the woman has any living children.

Second, we vary whether child status interacts with the woman's age: two models include `ch` (or `livch`) and `age` as additive terms, while the other four include a `ch` and `age` interaction. A quadratic effect of age (`I(age^2)`) was added to account for the nonlinear effect of age.

Third, we explore different random effects structures at the district level. Three of them use a single random intercept per district `(1 | district)`. One of them extends this with a random slope for urban status `(urban | district)`, allowing the urban/rural difference to vary by district. The next uses nested random intercepts for district and site within district (`1 | district/urban`) separating district-level from urban-within-district variation. (Scandola and Tidoni (2024) refer to this formulation as a "complex random intercepts" model). The remaining model uses only the `urban:district` grouping. To reduce convergence warnings and facilitate interpretability, age (already approximately centered in the data set) was standardized by scaling by twice the standard deviation (Gelman 2008).

```
> cm1 <- glmer(use ~ age_s + I(age_s^2) + urban + livch + (1|district),
+                            Contraception, binomial)
> ## switch from livch (ordinal) to ch (binary)
> cm2 <- update(cm1, . ~ . - livch + ch)
> ## add age by children interaction
> cm3 <- update(cm2, . ~ . + age_s:ch)
> ## allow urban effect to vary across districts (correlated)
> cm4 <- update(cm3, . ~ . - (1|district) + (1+urban|district))
> ## compound symmetric/nested formulation
> cm5 <- update(cm3, . ~ . - (1|district) + (1 | district/urban))
> ## as above but drop district effect
> cm6 <- update(cm3, . ~ . - (1|district) + (1 | district:urban))
```

Table 1 shows that the top three models, all of which include a child-by-age interaction

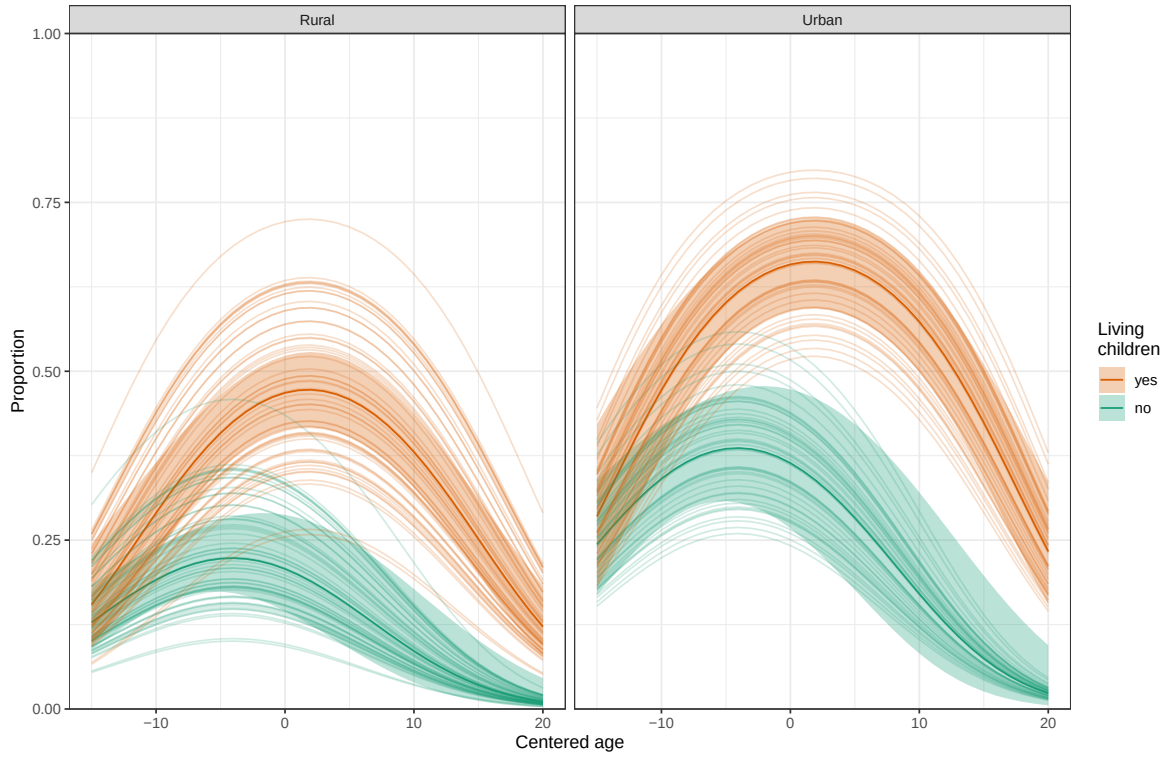

Figure 11: Predictions for contraception fit (`(1|district/urban)` model). Heavy lines and ribbons show population-level predictions; light lines show district-level predictions.

and some effect of urbanization, fit approximately equally well (Δ negative log-likelihood < 0.5). The `(1|district/urban)` model barely improves on the fit of `(1|district:urban)` (0.005 log-likelihood units), at the cost of an extra variance parameter, so it is almost 2 AIC units worse. `(1 + urban | district)` is a bit better (≈ 0.5 log-likelihood units), but includes a covariance parameter. Models that drop the interaction between child status and age or replace the binary child indicator with an integer count perform substantially worse (ΔAIC > 10).

Overall, the results suggest that accounting for urban/rural variation at the district level and including the age and child interaction are both important, but the precise random effects structure for urban/rural variation is relatively unimportant.

As with the CBPP data set, we can also compute profile, Wald, and parametric bootstrap confidence intervals for all of the models to understand the effects of each variable and visualize the among-model variation.

The point estimates and confidence intervals of the explanatory variables are similar across the six models, and across different methods for confidence interval construction, with a few exceptions.

Urban residence, presence of living children, and age were all strong predictors of contraceptive use among women in Bangladesh (because we are fitting a quadratic model for the effect of age, the non-significant effect of `age_s` simply means that the marginal effect of age *at the mean age* is not clearly negative or positive).

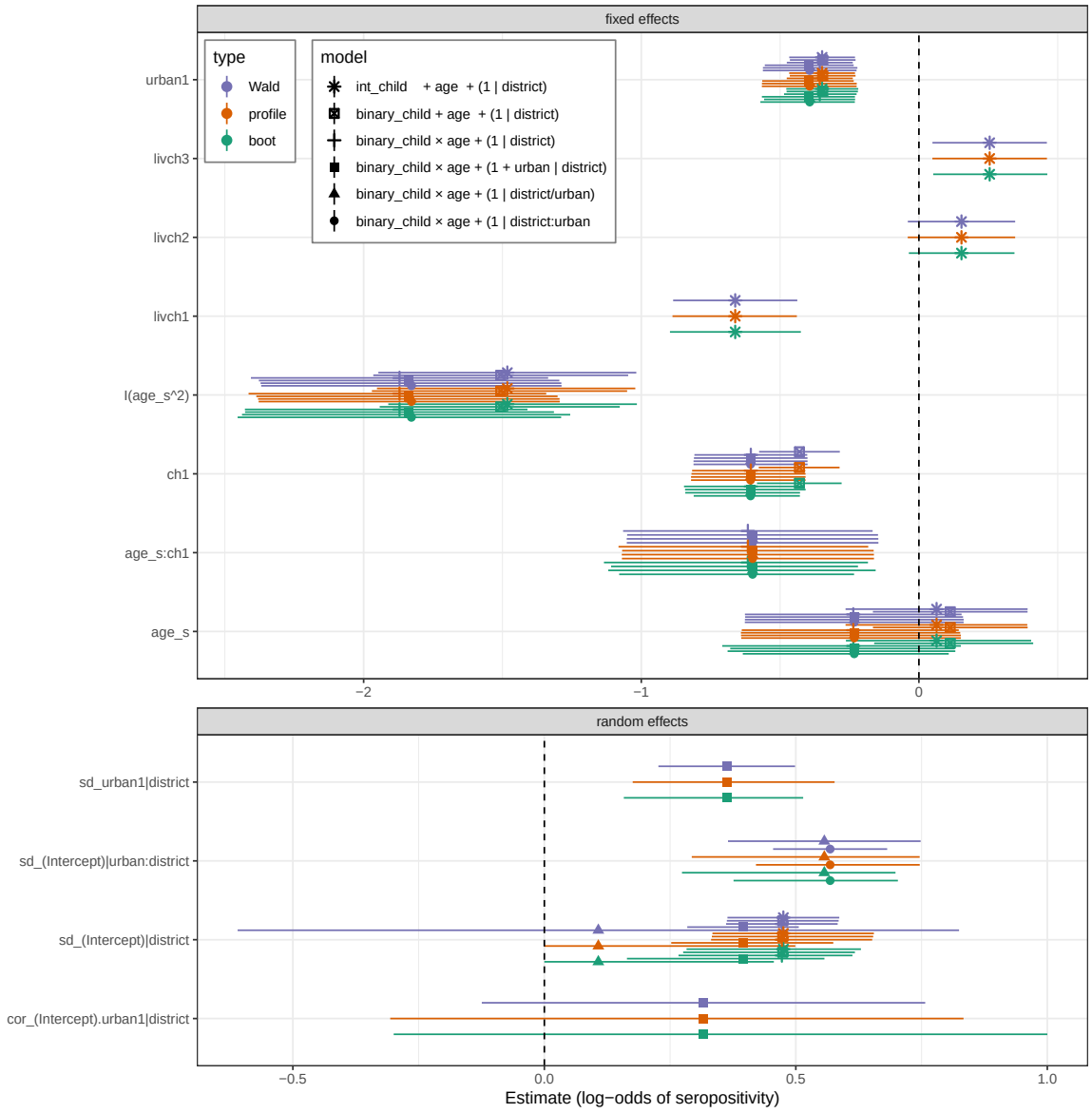

Figure 12: Contraception example: comparison of point and confidence interval estimation for different methods. (Note that the Wald CIs for variation in the intercept across districts, in the `(1|district/urban)` model, include negative values.)

With `lme4` versions greater than 2.0, we can also set up models with structured random effects covariance matrices. In the models below `diag` specifies a diagonal variance-covariance structure, while `cs` specifies a compound symmetric model. The `hom` argument specifies whether the model should allow different variances for each varying effect (e.g. for intercepts and slopes in the models below, which allow the effects of the continuous covariate of age to vary across districts).

Thus instead of (e.g.) `(1+urban|district)`, we can specify the random effect with diagonal, heterogeneous variances (`diag(1+urban|district)`); diagonal, homogeneous variances (`diag(1+urban|district, hom = TRUE)`); compound symmetric, heterogeneous variances (`cs(1+urban|district)`); or compound symmetric, homogeneous variances (`cs(1+urban|district, hom = TRUE)`).

We can use `VarCorr` and other accessor methods as we would for models with default, unstructured covariance matrices, e.g.:

```
> VarCorr(cm.cs)

 Groups   Name        Std.Dev. Corr
 district (Intercept) 0.615    -0.79 (cs)
          urbanY      0.725
```

References

Barr DJ, Levy R, Scheepers C, Tily HJ (2013). "Random Effects Structure for Confirmatory Hypothesis Testing: Keep It Maximal." *Journal of Memory and Language*, **68**(3), 255–278. doi:10.1016/j.jml.2012.11.001.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting linear mixed-effects models using lme4." *Journal of Statistical Software*, **67**, 1–48.

Bates DM, Watts DG (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley, Hoboken, NJ. ISBN 0-471-81643-4.

De Backer M, De Vroey C, Lesaffre E, Scheys I, De Keyser P (1998). "Twelve weeks of continuous oral therapy for toenail onychomycosis caused by dermatophytes: a double-blind comparative trial of terbinafine 250 mg/day versus itraconazole 200 mg/day." *Journal of the American Academy of Dermatology*, **38**(5, Supplement 2), S57–S63. doi:10.1016/S0190-9622(98)70486-4.

Gelman A (2008). "Scaling regression inputs by dividing by two standard deviations." *Statistics in medicine*, **27**(15), 2865–2873.

Huq N, Cleland J (1990). *Bangladesh fertility survey 1989*. National Institute of Population Research and Training, Dhaka.

Kristensen K, Nielsen A, Berg CW, Skaug H, Bell BM (2016). "TMB : Automatic Differentiation and Laplace Approximation." *Journal of Statistical Software*, **70**(5). doi:10.18637/jss.v070.i05.

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**(1), 27–40. doi:10.1016/j.prevetmed.2004.03.005.

Madsen H, Thyregod P (2011). *Introduction to General and Generalized Linear Models*. CRC Press. ISBN 978-1-4200-9155-7.

Matuschek H, Kliegl R, Vasishth S, Baayen H, Bates D (2017). "Balancing Type I Error and Power in Linear Mixed Models." *Journal of Memory and Language*, **94**, 305–315. doi:10.1016/j.jml.2017.01.001.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman and Hall, London.

Ripley BD (2004). "Selecting amongst Large Classes of Models." In NM Adams, M Crowder, D Hand, D Stephens (eds.), *Methods and models in statistics: In honor of Professor John Nelder, FRS*, pp. 155–170. Imperial College Press.

Scandola M, Tidoni E (2024). "Reliability and Feasibility of Linear Mixed Models in Fully Crossed Experimental Designs." *Advances in Methods and Practices in Psychological Science*, **7**(1), 25152459231214454. doi:10.1177/25152459231214454.

Stringer A (2024). "Exact Gradient Evaluation for Adaptive Quadrature Approximate Marginal Likelihood in Mixed Models for Grouped Data." *Statistics and Computing*, **35**(1), 4. ISSN 1573-1375. doi:10.1007/s11222-024-10536-z.

Stringer A, Bilodeau B, Tang Y (2022). "Asymptotics of Numerical Integration for Two-Level Mixed Models." doi:10.48550/arXiv.2202.07864.

Zeger SL, Karim MR (1991). "Generalized linear models with random effects: a Gibbs sampling approach." *Journal of the American Statistical Association*, **86**(413), 79–86.

6. Package versions used

Compiled with R version 4.6.0 (2026-04-24) and package versions lme4: 2.0.2, performance: 0.17.0, DHARMa: 0.5.0, see: 0.14.0.

7. Appendix: derivation of PIRLS

We seek to maximize the unscaled conditional log density for a GLMM over the conditional modes, $\boldsymbol{u}$. This problem is very similar to maximizing the log-likelihood for a GLM, which is a very thoroughly studied problem (e.g. McCullagh and Nelder 1989). The standard algorithm for dealing with this kind of problem is iteratively reweighted least squares (IRLS). Here we modify IRLS by incorporating a penalty term that accounts for variation in the random effects; we call the resulting algorithm penalized iteratively reweighted least squares (PIRLS).

The unscaled conditional log-density takes the form,

$$f(\boldsymbol{u}) = \log p(\boldsymbol{y}, \boldsymbol{u}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \boldsymbol{\psi}^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{a}^\top \boldsymbol{\phi} + \boldsymbol{c} - \frac{1}{2}\boldsymbol{u}^\top \boldsymbol{u} - \frac{q}{2}\log 2\pi \tag{28}$$

where $\boldsymbol{\psi}$ is the n-by-1 canonical parameter of an exponential family, $\boldsymbol{\phi}$ is the n-by-1 vector of cumulant functions, $\boldsymbol{c}$ an n-by-1 vector of normalizing constants, and $\boldsymbol{A}$ is an n-by-n diagonal matrix of prior weights, $\boldsymbol{a}$. Both $\boldsymbol{a}$ and $\boldsymbol{c}$ could depend on a dispersion parameter, although we ignore this possibility for now.

The canonical parameter, $\boldsymbol{\psi}$, and vector of cumulant functions, $\boldsymbol{\phi}$, depend on a linear predictor,

$$\boldsymbol{\eta} = \boldsymbol{o} + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\Lambda}_\theta \boldsymbol{u} \tag{29}$$

where $\boldsymbol{o}$ is an n-by-1 vector of *a priori* offsets. The specific form of this dependency is specified by the choice of the exponential family. The mean of this distribution, $\boldsymbol{\mu}$, is the *inverse link function* g^{-1} applied to $\boldsymbol{\eta}$.

Our goal is to find the values of $\boldsymbol{u}$ that maximize the unscaled conditional density, for given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ vectors. These maximizers are the conditional modes, which we require for the Laplace approximation and adaptive Gauss-Hermite quadrature. To do this maximization we use a variant of the Fisher scoring method, which is the basis of the iteratively reweighted least squares algorithm for generalized linear models. Fisher scoring is itself based on Newton's method, which we apply first.

7.1. Newton's method

To apply Newton's method, we need the gradient and the Hessian of the unscaled conditional log-likelihood. Following standard GLM theory (McCullagh and Nelder 1989), we use the chain rule,

$$\frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}} = \frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{\psi}} \frac{d\boldsymbol{\psi}}{d\boldsymbol{\mu}} \frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}} \frac{d\boldsymbol{\eta}}{d\boldsymbol{u}}$$

The first derivative in this chain follow from basic results in GLM theory,

$$\frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{\psi}} = (\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{A}$$

Again from standard GLM theory, the next two derivatives define the inverse diagonal variance matrix,

$$\frac{d\boldsymbol{\psi}}{d\boldsymbol{\mu}} = \boldsymbol{V}^{-1}$$

and the diagonal Jacobian matrix,

$$\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}} = \boldsymbol{M} \quad .$$

Finally, because $\boldsymbol{\beta}$ affects $\boldsymbol{\eta}$ only linearly,

$$\frac{d\boldsymbol{\eta}}{d\boldsymbol{u}} = \boldsymbol{Z}\boldsymbol{\Lambda}_\theta$$

Therefore we have,

$$\frac{dL(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}} = (\boldsymbol{y}-\boldsymbol{\mu})^\top \boldsymbol{A}\boldsymbol{V}^{-1}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{u}^\top \quad . \tag{30}$$

This is very similar to the gradient for GLMs with respect to fixed effects coefficients, $\boldsymbol{\beta}$. The only difference induced by differentiating with respect to the random effects, $\boldsymbol{u}$, is the addition of the $\boldsymbol{u}^\top$ term.

Again we apply the chain rule to take the Hessian,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{u}} = \frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{\mu}}\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}}\frac{d\boldsymbol{\eta}}{d\boldsymbol{u}} + \boldsymbol{I}_q \tag{31}$$

which leads to,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{u}} = \frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{\mu}}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \tag{32}$$

The first derivative in this chain can be expressed as,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{\mu}} = -\boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\boldsymbol{M}\boldsymbol{V}^{-1}\boldsymbol{A} + \boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\left[\frac{d\boldsymbol{M}\boldsymbol{V}^{-1}}{d\boldsymbol{\mu}}\right]\boldsymbol{A}\boldsymbol{R} \tag{33}$$

where $\boldsymbol{R}$ is a diagonal residuals matrix with $\boldsymbol{y}-\boldsymbol{\mu}$ on the diagonal. The two terms arise from a type of product rule, where we first differentiate the residuals, $\boldsymbol{y}-\boldsymbol{\mu}$, and then the diagonal matrix, $\boldsymbol{M}\boldsymbol{V}^{-1}$, with respect to $\boldsymbol{\mu}$.

The Hessian can therefore be expressed as,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}d\boldsymbol{u}} = -\boldsymbol{\Lambda}_\theta^\top\boldsymbol{Z}^\top\boldsymbol{M}\boldsymbol{A}^{1/2}\boldsymbol{V}^{-1/2}\left(\boldsymbol{I}_n - \boldsymbol{V}\boldsymbol{M}^{-1}\left[\frac{d\boldsymbol{M}\boldsymbol{V}^{-1}}{d\boldsymbol{\mu}}\right]\boldsymbol{R}\right)\boldsymbol{V}^{-1/2}\boldsymbol{A}^{1/2}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta + \boldsymbol{I}_q \tag{34}$$

This result can be simplified by expressing it in terms of a weighted random-effects design matrix, $\boldsymbol{U} = \boldsymbol{A}^{1/2}\boldsymbol{V}^{-1/2}\boldsymbol{M}\boldsymbol{Z}\boldsymbol{\Lambda}_\theta$,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{u}} = -\boldsymbol{U}^\top\left(\boldsymbol{I}_n - \boldsymbol{V}\boldsymbol{M}^{-1}\left[\frac{d\boldsymbol{V}^{-1}\boldsymbol{M}}{d\boldsymbol{\mu}}\right]\boldsymbol{R}\right)\boldsymbol{U} + \boldsymbol{I}_q \quad . \tag{35}$$

7.2. Fisher-like scoring

There are two ways to further simplify this expression for $\boldsymbol{U}^\top\boldsymbol{U}$. The first is to use the canonical link function for the family being used. Canonical links have the property that $\boldsymbol{V} = \boldsymbol{M}$, which means that for canonical links,

$$\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{u}} = -\boldsymbol{U}^\top\left(\boldsymbol{I}_n - \boldsymbol{I}_n\left[\frac{d\boldsymbol{I}_n}{d\boldsymbol{\mu}}\right]\boldsymbol{R}\right)\boldsymbol{U} + \boldsymbol{I}_q = \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{I}_q \tag{36}$$

The second way to simplify the Hessian is to take its expectation with respect to the distribution of the response, conditional on the current values of the spherical random effects coefficients, $\boldsymbol{u}$. The diagonal residual matrix, $\boldsymbol{R}$, has expectation 0. Therefore, because the response only enters into the expression for the Hessian via $\boldsymbol{R}$, we have that,

$$E\left(\frac{d^2L(\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{u})}{d\boldsymbol{u}\,d\boldsymbol{u}}|\boldsymbol{u}\right) = -\boldsymbol{U}^\top\left(\boldsymbol{I}_n - \boldsymbol{U}\boldsymbol{M}^{-1}\left[\frac{d\boldsymbol{V}^{-1}\boldsymbol{M}}{d\mu}\right]E(\boldsymbol{R})\right)\boldsymbol{U} + \boldsymbol{I}_q = \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{I}_q \tag{37}$$

Affiliation:

Anna Ly
Department of Mathematics & Statistics
McMaster University
1280 Main Street W
Hamilton, ON L8S 4K1, Canada

Rune Haubo Bojesen Christensen
Copenhagen Research Centre for Biological and Precision Psychiatry
Mental Health Centre Copenhagen, Copenhagen University Hospital - Bispebjerg and Frederiksberg
Copenhagen, Denmark
Email: Rune.Haubo@pm.me

Douglas Bates
Department of Statistics, University of Wisconsin - Madison
1205 University Ave.
Madison, WI 53706, U.S.A.
E-mail: bates@stat.wisc.edu

Martin Mächler
Seminar für Statistik, HG G 16
ETH Zurich
8092 Zurich, Switzerland
E-mail: maechler@stat.math.ethz.ch

Benjamin M. Bolker
Departments of Mathematics & Statistics and Biology
McMaster University
1280 Main Street W
Hamilton, ON L8S 4K1, Canada
E-mail: bolker@mcmaster.ca