

# GLMM FAQ

Code ▼

Ben Bolker and others

01 Mar 2022

- Introduction
  - Other sources of help
- References
  - linear mixed models
    - web/open
    - books (dead-tree/closed)
- Model definition
  - Model specification
  - Should I treat factor xxx as fixed or random?
  - Nested or crossed?
  - (When) can I include a predictor as both fixed and random?
- Model extensions
  - Overdispersion
    - Testing for overdispersion/computing overdispersion factor
    - Fitting models with overdispersion?
    - Underdispersion
  - Gamma GLMMs
  - Beta GLMMs
  - Zero-inflation
    - Count data
    - Continuous data
    - Tests for zero-inflation
  - Spatial and temporal correlation models, heteroscedasticity (“R-side” models)
  - Penalization/handling complete separation
  - Non-Gaussian random effects
- Estimation
  - What methods are available to fit (estimate) GLMMs?
  - Troubleshooting
    - Convergence warnings
    - Singular models: random effect variances estimated as zero, or correlations estimated as +/- 1
    - Setting residual variances to a fixed value (zero or other)
    - Other problems/ `lme4` error messages
  - REML for GLMMs
- Model diagnostics
- Inference and confidence intervals
  - Testing hypotheses
    - What are the p-values listed by `summary(glmerfit)` etc.? Are they reliable?
    - Methods for testing single parameters
    - Tests of effects (i.e. testing that several parameters are simultaneously zero)
    - Is the likelihood ratio test reliable for mixed models?
    - Why doesn't `lme4` display denominator degrees of freedom/p values? What other options do I have?
    - Testing significance of random effects
    - Standard errors of variance estimates
    - P-values: MCMC and parametric bootstrap
  - Predictions and/or confidence (or prediction) intervals on predictions
    - `lme`
    - `lme4`

- glmmTMB
- Confidence intervals on conditional means/BLUPs/random effects
  - lme4
- Power analysis
- Model selection and averaging
  - Can I use AIC for mixed models? How do I count the number of degrees of freedom for a random effect?
- Model summaries (goodness-of-fit, decomposition of variance, etc.)
  - How do I compute a coefficient of determination ( $R^2$ ), or an analogue, for (G)LMMs?
    - Problem
    - Simple/crude solutions
    - Sophisticated solutions
  - Variable importance
  - Do I have to specify the levels of fixed effects in lmer?
- Miscellaneous/procedural
  - Pronunciation of lmer / glmer /etc.
  - Storing information
- Mixed modeling packages
  - Which R packages (functions) fit GLMMs?
  - Should I use aov(), nlme, or lme4, or some other package?
    - linear and nonlinear mixed models
    - GLMMs
    - Additive and generalized-additive mixed models
    - Hierarchical GLMs
    - diagnostic and modeling frameworks
    - data and examples
    - extensions
    - Interfaces to other systems
    - modeling based on LMMs
  - Off-CRAN mixed modeling packages:
    - R-forge and Github:
    - Other open source:
    - Commercial:
  - Package versions used
  - To do
- Bibliography

## Introduction

This is an informal FAQ list for the `r-sig-mixed-models` mailing list.

The most commonly used functions for mixed modeling in R are

- *linear mixed models*: `aov()`, `nlme::lme`<sup>1</sup>, `lme4::lmer`; `brms::brm`
- *generalized linear mixed models* (GLMMs)
  - frequentist: `MASS::glmmPQL`, `lme4::glmer`; `glmmTMB`
  - Bayesian: `MCMCglmm::MCMCglmm`; `brms::brm`
- *nonlinear mixed models*: `nlme::nlme`, `lme4::nlmer`; `brms::brm`
- *GNLMMs*: `brms::brm`

Another quick-and-dirty way to search for mixed-model related packages on CRAN:

```
grep("l.?m[me][^t]",rownames(available.packages()),value=TRUE)
```

```
## [1] "blmeco"          "buildmer"        "cellVolumeDist"  "climextRemes"
## [5] "curtailment"     "elementR"        "glmertree"       "glmmEP"
## [9] "glmmfields"      "glmmLasso"       "glmmML"          "glmmSeq"
## [13] "glmsr"           "glmmTMB"         "lamme"           "lme4"
## [17] "lmec"            "lmeInfo"         "lmem.qtlr"       "lmeNB"
## [21] "lmeNBBayes"      "lmeresampler"    "lmerTest"        "lmeSplines"
## [25] "lmmot"           "lmmpar"          "lrmest"          "lsmeans"
## [29] "mailmerge"       "mlmm.gwas"       "mlmm"            "mvglmmRank"
## [33] "nlmeODE"         "nlmeU"           "palmerpenguins"  "tlmec"
## [37] "vagalumeR"
```

There are some false positives here (e.g. `palmerpenguins`); see here (<https://xkcd.com/1313/>) if you're interested in "regex golf".

## Other sources of help

- the mailing list is `r-sig-mixed-models@r-project.org`
  - sign up here (<https://stat.ethz.ch/mailman/listinfo/r-sig-mixed-models>)
  - archives here (<https://stat.ethz.ch/pipermail/r-sig-mixed-models/>)
  - or Google search with the tag `site:https://stat.ethz.ch/pipermail/r-sig-mixed-models/`
- The source code of this document is available on GitHub (<https://github.com/bbolker/mixedmodels-misc/blob/master/glmmFAQ.rmd>); the rendered (HTML) version lives on GitHub pages (<http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>).
- Searching on StackOverflow with the `[r]` `[mixed-models]` tags (<http://stackoverflow.com/questions/tagged/r%20mixed-models?mode=all>), or on CrossValidated with the `[mixed-model]` tag (<http://stats.stackexchange.com/questions/tagged/mixed-model>) may be helpful (these sites also have an `[lme4]` tag).

### DISCLAIMERS:

- (G)LMMs are hard - harder than you may think based on what you may have learned in your second statistics class, which probably focused on picking the appropriate sums of squares terms and degrees of freedom for the numerator and denominator of an  $F$  test. 'Modern' mixed model approaches, although more powerful (they can handle more complex designs, lack of balance, crossed random factors, some kinds of non-Normally distributed responses, etc.), also require a new set of conceptual tools. In order to use these tools you should have at least a general acquaintance with classical mixed-model experimental designs but you should also, probably, read something about modern mixed model approaches. Littell et al. (2006) and Pinheiro and Bates (2000) are two places to start, although Pinheiro and Bates is probably more useful if you want to use R. Other useful references include Gelman and Hill (2006) (focused on Bayesian methods) and Zuur et al. (2009b). If you are going to use generalized linear mixed models, you should understand generalized linear models (Dobson and Barnett (2008), Faraway (2006), and McCullagh and Nelder (1989) are standard references; the last is the canonical reference, but also the most challenging).
- All of the issues that arise with regular linear or generalized-linear modeling (e.g.: inadequacy of p-values alone for thorough statistical analysis; need to understand how models are parameterized; need to understand the principle of marginality and how interactions can be treated; dangers of overfitting, which are not mitigated by stepwise procedures; the non-existence of free lunches) also apply, and can apply more severely, to mixed models.
- When SAS (or Stata, or Genstat/AS-REML or ...) and R differ in their answers, R may not be wrong. Both SAS and R may be 'right' but proceeding in a different way/answering different questions/using a different philosophical approach (or both may be wrong ...)
- The advice in this FAQ comes with **absolutely no warranty of any sort**.

## References

### linear mixed models

#### web/open

- UCLA IDRE statistical consulting (<https://stats.idre.ucla.edu/other/mult-pkg/introduction-to-linear-mixed-models/>)

- Barr (2020) Chapters 5-8

## books (dead-tree/closed)

- pinheiro\_mixed-effects\_2000: LMM only.
- Zuur et al. (2009b): Focused on ecology.
- Gelman and Hill (2006): LMM and GLMM; Bayesian; examples from social science. Intermediate mathematics.
- (Rethinking)

# Model definition

## Model specification

The following formula extensions for specifying random-effects structures in R are used by

- lme4
- nlme (nested effects only, although crossed effects can be specified with more work)
- glmmADMB and glmmTMB

MCMCglmm uses a different specification, inherited from AS-REML.

(Modified from Robin Jeffries, UCLA:)

formula	meaning
$(1 \text{group})$	random group intercept
$(x \text{group}) = (1+x \text{group})$	random slope of x within group with correlated intercept
$(0+x \text{group}) = (-1+x \text{group})$	random slope of x within group: no variation in intercept
$(1 \text{group}) + (0+x \text{group})$	uncorrelated random intercept and random slope within group
$(1 \text{site/block}) = (1 \text{site})+(1 \text{site:block})$	intercept varying among sites and among blocks within sites (nested random effects)
$\text{site}+(1 \text{site:block})$	<i>fixed</i> effect of sites plus random variation in intercept among blocks within sites
$(x \text{site/block}) = (x \text{site})+(x \text{site:block}) = (1 + x \text{site})+(1+x \text{site:block})$	slope and intercept varying among sites and among blocks within sites
$(x1 \text{site})+(x2 \text{block})$	two different effects, varying at different levels
$x*\text{site}+(x \text{site:block})$	fixed effect variation of slope and intercept varying among sites and random variation of slope and intercept among blocks within sites
$(1 \text{group1})+(1 \text{group2})$	intercept varying among crossed random effects (e.g. site, year)

Or in a little more detail:

equation	formula
$\beta_0 + \beta_1 X_i + e_{si}$	n/a (Not a mixed-effects model)
$(\beta_0 + b_{S,0s}) + \beta_1 X_i + e_{si}$	$\sim X + (1 \text{Subject})$
$(\beta_0 + b_{S,0s}) + (\beta_1 + b_{S,1s})X_i + e_{si}$	$\sim X + (1 + X \text{Subject})$

$(\beta_0 + b_{S,0s} + b_{I,0i}) + (\beta_1 + b_{S,1s})X_i + e_{si}$	$\sim X + (1 + X   \text{Subject}) + (1   \text{Item})$
As above, but $S_{0s}, S_{1s}$ independent	$\sim X + (1   \text{Subject}) + (0 + X   \text{Subject}) + (1   \text{Item})$
$(\beta_0 + b_{S,0s} + b_{I,0i}) + \beta_1 X_i + e_{si}$	$\sim X + (1   \text{Subject}) + (1   \text{Item})$
$(\beta_0 + b_{I,0i}) + (\beta_1 + b_{S,1s})X_i + e_{si}$	$\sim X + (0 + X   \text{Subject}) + (1   \text{Item})$

Modified from: <http://stats.stackexchange.com/questions/13166/rs-lmer-cheat-sheet?lq=1>

(<http://stats.stackexchange.com/questions/13166/rs-lmer-cheat-sheet?lq=1>) (Livius)

The **magic** development version of the equatiomatic package (<https://github.com/datalorax/equatiomatic>) can handle mixed models ( `remotes::install_github("datalorax/equatiomatic")` ), e.g.

```
library(lme4)
library(equatiomatic)
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)
equatiomatic::extract_eq(fm1)
```

$$\text{Reaction}_i \sim N(\alpha_{j[i]} + \beta_{1j[i]}(\text{Days}), \sigma^2)$$

$$\begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_j}^2 & \rho_{\alpha_j \beta_{1j}} \\ \rho_{\beta_{1j} \alpha_j} & \sigma_{\beta_{1j}}^2 \end{pmatrix}\right), \text{ for Subject } j = 1, \dots, J$$

It doesn't handle GLMMs (yet), but you could fit two fake models — one LMM like your GLMM but with a Gaussian response, and one GLM with the same family/link function as your GLMM but without the random effects — and put the pieces together.

More possibly useful links:

- Rense Nieuwenhuis's blogpost/lesson on lme4 model specification (<http://www.rensenieuwenhuis.nl/r-sessions-16-multilevel-model-specification-lme4/>)
- CrossValidated's lmer cheat sheet (<https://stats.stackexchange.com/questions/13166/rs-lmer-cheat-sheet>)
- Kristoffer Magnusson's Using R and lme/lmer to fit different two- and three-level longitudinal models (<https://rpsychologist.com/r-guide-longitudinal-lme-lmer>)

## Should I treat factor xxx as fixed or random?

This is in general a far more difficult question than it seems on the surface. There are many competing philosophies and definitions. For example, from Gelman (2005):

Before discussing the technical issues, we briefly review what is meant by fixed and random effects. It turns out that different—in fact, incompatible—definitions are used in different contexts. [See also Kreft and de Leeuw (1998), Section 1.3.3, for a discussion of the multiplicity of definitions of fixed and random effects and coefficients, and Robinson (1998) for a historical overview.] Here we outline five definitions that we have seen: 1. Fixed effects are constant across individuals, and random effects vary. For example, in a growth study, a model with random intercepts  $\alpha_i$  and fixed slope  $\beta$  corresponds to parallel lines for different individuals  $i$ , or the model  $y_{it} = \alpha_i + \beta t$ . Kreft and de Leeuw [(1998), page 12] thus distinguish between fixed and random coefficients. 2. Effects are fixed if they are interesting in themselves or random if there is interest in the underlying population. Searle, Casella and McCulloch [(1992), Section 1.4] explore this distinction in depth. 3. “When a sample exhausts the population, the corresponding variable is fixed; when the sample is a small (i.e., negligible) part of the population the corresponding variable is random” [Green and Tukey (1960)]. 4. “If an effect is assumed to be a realized value of a random variable, it is called a random effect” [LaMotte (1983)]. 5. Fixed effects are estimated using least squares (or, more generally, maximum likelihood) and random effects are estimated with shrinkage [“linear unbiased prediction” in the terminology of Robinson (1991)]. This definition is standard in the multilevel modeling literature [see, e.g., Snijders and Bosker (1999), Section 4.2] and in econometrics.

Another useful comment (via Kevin Wright) reinforcing the idea that “random vs. fixed” is not a simple, cut-and-dried decision: from Schabenberger and Pierce (2001), p. 627:

Before proceeding further with random field linear models we need to remind the reader of the adage that one modeler’s random effect is another modeler’s fixed effect.

Clark and Linzer (2015) address this question from a mostly econometric perspective, focusing mostly on practical variance/bias/RMSE criteria.

One point of particular relevance to ‘modern’ mixed model estimation (rather than ‘classical’ method-of-moments estimation) is that, for practical purposes, there must be a reasonable number of random-effects levels (e.g. blocks) – more than 5 or 6 at a minimum. This is not surprising if you consider that random effects estimation is trying to estimate an among-block variance. For example, from Crawley (2002) p. 670:

Are there enough levels of the factor in the data on which to base an estimate of the variance of the population of effects? No, means [you should probably treat the variable as] fixed effects.

Some researchers (who treat fixed vs random as a philosophical rather than a pragmatic decision) object to this approach.

Also see a very thoughtful chapter in Hodges (2016).

Treating factors with small numbers of levels as random will in the best case lead to very small and/or imprecise estimates of random effects; in the worst case it will lead to various numerical difficulties such as lack of convergence, zero variance estimates, etc.. (A small simulation exercise (<https://rpubs.com/bbolker/4187>) shows that at least the estimates of the standard deviation are downwardly biased in this case; it’s not clear whether/how this bias would affect the point estimates of fixed effects or their estimated confidence intervals.) In the classical method-of-moments approach these problems may not arise (because the sums of squares are always well defined as long as there are at least two units), but the underlying problems of lack of power are there nevertheless.

Thierry Onkelinx has a blog post (<https://www.muscardinus.be/2018/09/number-random-effect-levels/>) with some simulations on the impact of the number of levels and concludes with a few recommendations for the number of levels of the grouping variable  $n_s$ : > - get  $n_s > 1000$  levels when an accurate estimate of the random effect variance is crucial. E.g. when a single number will be use for power calculations. > - get  $n_s > 100$  levels when a reasonable estimate of the random effect variance is sufficient. E.g. power calculations with sensitivity analysis of the random effect variance. > - get  $n_s > 20$  levels for an experimental study > - in case  $10 < n_s < 20$  you should validate the model very cautious before using the output > - in case  $n_s < 10$  it is safer to use the variable as a fixed effect.

Oberpriller, Leite, and Pichler (2021) also performed a simulation study and found that while the estimates are similar for treating a variable with a small number of levels as fixed or random are similar, there was an impact on Type 1 and Type 2 error rates. They also found that the precise random effects structure (e.g., inclusion of random slopes) had a large impact on these properties.

Also see this thread (<https://stat.ethz.ch/pipermail/r-sig-mixed-models/2010q2/003709.html>) on the r-sig-mixed-models mailing list and this question (<https://stats.stackexchange.com/questions/37647/what-is-the-minimum-recommended-number-of-groups-for-a-random-effects-factor>) on CrossValidated.

## Nested or crossed?

- Relatively few mixed effect modeling packages can handle crossed random effects, i.e. those where one level of a random effect can appear in conjunction with more than one level of another effect. (This definition is confusing, and I would happily accept a better one.) A classic example is crossed temporal and spatial effects. If there is random variation among temporal blocks (e.g. years) “and” random variation among spatial blocks (e.g. sites), “and” if there is a consistent year effect across sites and “vice versa”, then the random effects should be treated as crossed.
- `lme4` does handled crossed effects, efficiently
- if you need to deal with crossed REs in conjunction with some of the features that `nlme` offers (e.g. heteroscedasticity of residuals via `weights / varStruct`, correlation of residuals via `correlation / corStruct`, or if you want to used crossed REs with the `gamlss` package, see p. 163ff of Pinheiro and Bates (2000) (section 4.2.2: Google books link (<http://tinyurl.com/crossedRE>)). I give a worked example here (<https://gist.github.com/bbolker/83c1ff036d4850d7dc4ca832e354f6a8>). As far as I can tell, a couple of hacks are necessary to get this to work: (1) the data must be expressed as a `groupedData` object (at least, I haven’t managed to get it to work in any other way); (2) the crossed effects must be *nested within another grouping factor* - in the example here I define a dummy group, which is awkward (it makes the variance component for this group and the residual variance jointly unidentifiable), but otherwise seems to work OK.
- I rarely find it useful to think of fixed effects as “nested” (although others disagree); if for example treatments A and B are only measured in block 1, and treatments C and D are only measured in block 2, one still assumes (because they are fixed effects) that each treatment would have the same effect if applied in the other block. (One might like to estimate treatment-by-block interactions, but in this case the experimental design doesn’t allow it; one would have to have multiple treatments measured within each block, although not necessarily all treatments in every block.) One would code this analysis as `response~treatment+(1|block)` in `lme4`. Also, in the case of fixed effects, crossed and nested specifications change the parameterization of the model, but not anything else (e.g. the number of parameters estimated, log-likelihood, model predictions are all identical). That is, in R’s `model.matrix` function (which implements a version of Wilkinson-Rogers notation) `a*b` and `a/b` (which expand to `1+a+b+a:b` and `1+a+a:b` respectively) give model matrices with the same number of columns.
- Whether you explicitly specify a random effect as nested or not depends (in part) on the way the levels of the random effects are coded. If the ‘lower-level’ random effect is coded with unique levels, then the two syntaxes `(1|a/b)` (or `(1|a)+(1|a:b)`) and `(1|a)+(1|b)` are equivalent. If the lower-level random effect has the same labels within each larger group (e.g. blocks 1, 2, 3, 4 within sites A, B, and C) then the explicit nesting `(1|a/b)` is required. It seems to be considered best practice to code the nested level uniquely (e.g. A1, A2, ..., B1, B2, ...) so that confusion between nested and crossed effects is less likely.

## (When) can I include a predictor as both fixed and random?

See blog post by Thierry Onkelinx (<https://www.muscardinus.be/2017/08/fixed-and-random/>)

## Model extensions

# Overdispersion

## Testing for overdispersion/computing overdispersion factor

- with the usual caveats, plus a few extras – counting degrees of freedom, etc. – the usual procedure of calculating the sum of squared Pearson residuals and comparing it to the residual degrees of freedom should give at least a crude idea of overdispersion. The following attempt counts each variance or covariance parameter as one model degree of freedom and presents the sum of squared Pearson residuals, the ratio of (SSQ residuals/rdf), the residual df, and the  $p$ -value based on the (approximately!!) appropriate  $\chi^2$  distribution. **Do PLEASE note the usual, and extra, caveats noted here: this is an APPROXIMATE estimate of an overdispersion parameter.** Even in the GLM case, the expected deviance per point equaling 1 is only true as the distribution of individual deviates approaches normality, i.e. the usual  $\lambda > 5$  rules of thumb for Poisson values and  $\min(Np, N(1-p)) > 5$  for binomial values (e.g. see Venables and Ripley (2002), p. 208-209 (<http://books.google.com/books?id=974c4vKurNkC&pg=PA209>)). (And that's without the extra complexities due to GLMM, i.e. the “effective” residual df should be large enough to make the sums of squares converge on a  $\chi^2$  distribution ...)
- overdispersion is not estimable (and hence practically irrelevant) for Bernoulli models (= binary data = binomial with  $N = 1$ ).
- overdispersion does occur, but is much less central, for models that estimate a scale parameter (i.e. almost anything but Poisson or binomial: Gamma, negative binomial, ...). Whereas overdispersion for one-parameter distributions like the Poisson or binomial suggests the need to move to an analogous overdispersed distribution (e.g. negative binomial or beta-binomial respectively), overdispersion for more complex models is a more general indication of model misspecification.
- The recipes below may need adjustment for some of the more complex model types allowed by `glmmTMB` (e.g. zero-inflation/variable dispersion), where it's less clear what to measure to estimate overdispersion.

The following function should work for a variety of model types (at least `glmmADMB`, `glmmTMB`, `lme4`, ...).

```
overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model,type="pearson")
  Pearson.chisq <- sum(rp^2)
  prat <- Pearson.chisq/rdf
  pval <- pchisq(Pearson.chisq, df=rdf, lower.tail=FALSE)
  c(chisq=Pearson.chisq,ratio=prat,rdf=rdf,p=pval)
}
```

Example:

```
library(lme4)
library(glmmTMB)
```

```
set.seed(101)
d <- data.frame(x=runif(1000),
                f=factor(sample(1:10,size=1000,replace=TRUE)))
suppressMessages(d$y <- simulate(~x+(1|f), family=poisson,
                                newdata=d,
                                newparams=list(theta=1,beta=c(0,2)))[[1]])
m1 <- glmer(y~x+(1|f),data=d,family=poisson)
overdisp_fun(m1)
```

```
##          chisq          ratio          rdf          p
## 1035.9966326    1.0391140  997.0000000    0.1902294
```

```
m2 <- glmmTMB(y~x+(1|f),data=d,family="poisson")
overdisp_fun(m2)
```



```
##          chisq          ratio          rdf          p
## 1035.9961394    1.0391135  997.0000000    0.1902323
```

The `gof` function in the `aods3` provides similar functionality (it reports both deviance- and  $\chi^2$ -based estimates of overdispersion and tests).

## Fitting models with overdispersion?

- quasilielihood estimation: `MASS::glmmPQL` (<http://finzi.psych.upenn.edu/R/library/MASS/html/glmmPQL.html>). Quasi- was deemed unreliable in `lme4`, and is no longer available. (Part of the problem was questionable numerical results in some cases; the other problem was that DB felt that he did not have a sufficiently good understanding of the theoretical framework that would explain what the algorithm was actually estimating in this case.) `geepack::geeglm` (<http://finzi.psych.upenn.edu/R/library/geepack/html/geeglm.html>) may be workable (haven't tried it)

If you really want quasi-likelihood analysis for `glmer` fits, you can do it yourself by adjusting the coefficient table - i.e., by multiplying the standard error by the square root of the dispersion factor<sup>2</sup> and recomputing the  $Z$ - and  $p$ -values accordingly, as follows:

```
## extract summary table; you may also be able to do this via
## broom::tidy or broom.mixed::tidy
quasi_table <- function(model,ctab=coef(summary(model)),
                        phi=overdisp_fun(model)["ratio"]) {
  qctab <- within(as.data.frame(ctab),
  {
    `Std. Error` <- `Std. Error`*sqrt(phi)
    `z value` <- Estimate/`Std. Error`
    `Pr(>|z|)` <- 2*pnorm(abs(`z value`), lower.tail=FALSE)
  })
  return(qctab)
}
printCoefmat(quasi_table(m1),digits=3)
```

```
##          Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.2277    0.2700    0.84    0.4
## x           2.0640    0.0528   39.11 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## to use this with glmmTMB, we need to separate out the
## conditional component of the summary
printCoefmat(quasi_table(m2,
                        ctab=coef(summary(m2))["cond"]),
            digits=3)
```

```
##          Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.2277    0.2700    0.84    0.4
## x           2.0640    0.0528   39.09 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Another version, this one tidyverse-centric:

```
library(broom.mixed)
library(dplyr)
tidy_quasi <- function(model, phi=overdisp_fun(model)["ratio"],
                      conf.level=0.95) {
  tt <- (tidy(model, effects="fixed")
    %>% mutate(std.error=std.error*sqrt(phi),
              statistic=estimate/std.error,
              p.value=2*pnorm(abs(statistic), lower.tail=FALSE))
  )
  return(tt)
}
tidy_quasi(m1)
```

```
## # A tibble: 2 × 6
##   effect term      estimate std.error statistic p.value
##   <chr> <chr>      <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed (Intercept)  0.228    0.270     0.843    0.399
## 2 fixed x          2.06     0.0528   39.1     0
```

```
tidy_quasi(m2)
```

```
## # A tibble: 2 × 7
##   effect component term      estimate std.error statistic p.value
##   <chr> <chr>    <chr>      <dbl>    <dbl>    <dbl>    <dbl>
## 1 fixed cond      (Intercept)  0.228    0.270     0.843    0.399
## 2 fixed cond        x          2.06     0.0528   39.1     0
```

These functions make some simplifying assumptions: (1) this overdispersion computation is approximate

In this case using quasi-likelihood doesn't make much difference, since the data we simulated in the first place were Poisson.) Keep in mind that once you switch to quasi-likelihood you will either have to eschew inferential methods such as the likelihood ratio test, profile confidence intervals, AIC, etc., or make more heroic assumptions to compute "quasi-" analogs of all of the above (such as QAIC).

- observation-level random effects (OLRE: this approach should work in most packages). If you want to a citation for this approach, try Elston et al. (2001), who cite Lawson et al. (1999); apparently there is also an example in section 10.5 of Maindonald and Braun (2010), and (according to an R-sig-mixed-models post) this is also discussed by Rabe-Hesketh and Skrondal (2008). Also see Browne et al. (2005) for an example in the binomial context (i.e. logit-normal-binomial rather than lognormal-Poisson). Agresti's excellent (2002) book Agresti (2002) also discusses this (section 13.5), referring back to Breslow (1984) and Hinde (1982). [Notes: (a) I haven't checked all these references myself, (b) I can't find the reference any more, but I have seen it stated that observation-level random effect estimation is probably dodgy for PQL approaches as used in Elston et al 2001]
- alternative distributions
  - Poisson-lognormal model for counts or binomial-logit-Normal model for proportions (see above, "observation-level random effects")
  - negative binomial for counts or beta-binomial for proportions
    - `lme4::glmer.nb()` should fit a negative binomial, although it is somewhat slow and fragile compared to some of the other methods suggested here. `lme4` cannot fit beta-binomial models (these cannot be formulated as a part of the exponential family of distributions)
    - `glmmTMB` (<https://github.com/glmmTMB/glmmTMB/>) will fit two parameterizations of the negative binomial: `family="nbinom2"` gives the classic parameterization with  $\sigma^2 = \mu(1 + \mu/k)$  ("NB2" in Hardin and Hilbe's terminology) while `family="nbinom1"` gives a parameterization with  $\sigma^2 = \phi\mu$ ,  $\phi > 1$  ("NB1" to Hardin and Hilbe). The latter might also be called a "quasi-Poisson" parameterization because it matches the mean-variance relationship assumed by quasi-Poisson models, i.e. the variance is strictly proportional to the mean (although the proportionality constant must be  $>1$ , a limitation that does not apply to quasi-likelihood approaches). `glmmADMB` (<http://github.com/bbolker/glmmADMB/>) will also fit these models, with `family="nbinom"` for NB2, but is deprecated

- in favour of `glmmTMB`.)
  - `glmmTMB` allows beta-binomial models ((Harrison 2015) suggests comparing beta-binomial with OLRE models to assess reliability)
  - the `brms` package has a `negbinomial` family (no beta-binomial, but it does have a wide range of other families)
- other packages/approaches (less widely used, or requiring a bit more effort)
  - `gamlss.mx:gamlssNP`
  - WinBUGS/JAGS (via `R2WinBUGS/Rjags`)
  - AD Model Builder (possibly via `R2admb` package) or `TMB`
  - `gnlmm` in the `repeated` package (off-CRAN (<http://www.commanster.eu/rcode.html>))
  - ASREML (<http://www.asreml.com/software/genstat/htmlhelp/server/GLMM.htm>)

Negative binomial models in `glmmTMB` and lognormal-Poisson models in `glmer` (or `MCMCglmm`) are probably the best quick alternatives for overdispersed count data. If you need to explore alternatives (different variance-mean relationships, different distributions), then `ADMB`, `TMB`, `WinBUGS`, `Stan`, `NIMBLE` are the most flexible alternatives.

## Underdispersion

Underdispersion (much *less* variability than expected) is a less common problem than overdispersion.

- mild underdispersion is sometimes ignored, since it tends in general to lead to conservative rather than anti-conservative results
- quasi-likelihood (and the quasi-hack listed above) can handle under- as well as overdispersion
- some other solutions exist, but are less widely implemented
  - for distributions with a small range (e.g. litter sizes of large mammals), one can treat responses as ordinal (e.g. using the `ordinal` package, or `MCMCglmm` or `brms` for Bayesian solutions)
  - the COM-Poisson distribution and generalized Poisson distributions, implemented in `glmmTMB`, can handle underdispersion (J. Hilbe recommends the latter in this CrossValidated answer (<https://stats.stackexchange.com/questions/67385/what-is-the-appropriate-model-for-underdispersed-count-data>)). (`VGAM` has a generalized Poisson distribution, but doesn't handle random effects.)

## Gamma GLMMs

While one (well, OK I) would naively think that GLMMs with Gamma distributions would be just as easy (or hard) as any other sort of GLMMs, it seems that they are in fact harder to implement. Basic simulated examples of Gamma GLMMs can fail in `lme4` despite analogous problems with Poisson, binomial, etc. distributions. Solutions: - the default inverse link seems particularly problematic; try other links (especially `family=Gamma(link="log")`) if that is possible/makes sense - consider whether a lognormal model (i.e. a regular LMM on logged data) would work/makes sense. - Lo and Andrews (2015) argue that the Gamma family with an *identity* link is superior to lognormal models for reaction-time data. I (BMB) don't find their argument particularly convincing, but lots of people want to do this. Unfortunately this is technically challenging (see here (<https://github.com/lme4/lme4/issues/573>)), because it is likely that some "illegal" values (predicted responses  $\leq 0$ ) will occur while fitting the model, even if the final fitted model makes no impossible predictions. Thus something has to be done to make the model-fitting machinery tolerant of such values (i.e. returning `NA` for these model evaluations, or clamping illegal values to the constrained space with an appropriate smooth penalty function).

Gamma models can be fitted by a wide variety of platforms (`lme4::glmer`, `MASS::glmmPQL`, `glmmADMB`, `glmmTMB`, `MixedModels.jl`, `MCMCglmm`, `brms` ... not sure about others).

## Beta GLMMs

Proportion data where the denominator (e.g. maximum possible number of successes for a given observation) is not known can be modeled using a Beta distribution. Smithson and Verkuilen (2006) is a good introduction for non-statisticians (*not* in the mixed-model case), and the `betareg` package (Cribari-Neto and Zeileis 2009) handles *non*-mixed Beta regressions. The `glmmTMB` and `brms` packages handle Beta mixed models (`brms` also handles zero-inflated and zero-one inflated models).

## Zero-inflation

See e.g. Martin et al. (2005) or Warton (2005) (“many zeros does not mean zero inflation”) or Zuur et al. (2009a) for general information on zero-inflation.

## Count data

- `MCMCglmm` handles zero-truncated, zero-inflated, and zero-altered models, although specifying the models is a little bit tricky: see Sections 5.3 to 5.5 of the CourseNotes vignette (<https://cran.r-project.org/web/packages/MCMCglmm/vignettes/CourseNotes.pdf>)
- `glmmADMB` handles
  - zero-inflated models (with a single zero-inflation parameter – i.e., the level of zero-inflation is assumed constant across the whole data set)
  - truncated Poisson and negative binomial distributions (which allows two-stage fitting of hurdle models)
- `glmmTMB` handles a variety of Z-I and Z-T models (allows covariates, and random effects, in the zero-alteration model)
- `brms` does too
- so does `GLMMadaptive`
- Gavin Simpson has a detailed writeup (<http://www.fromthebottomoftheheap.net/2017/05/04/compare-mgcv-with-glmmTMB/>) showing that `mgcv::gam()` can do simple mixed models (Poisson, not NB) with zero-inflation, and comparing `mgcv` with `glmmTMB` results
- `gamlssNP` in the `gamlss.mx` package should handle zero-inflation, and the `gamlss.tr` package should handle truncated (i.e. hurdle) models – but I haven’t tried them
- roll-your-own: `ADMB/R2admb`, `WinBUGS/R2WinBUGS`, `TMB`, `Stan`, ...

## Continuous data

Continuous data are a special case where the mixture model for zero-inflated data is less relevant, because observations that are exactly zero occur with *probability* (but not probability density) zero. There are two cases of interest:

### Probability density of $x$ zero or infinite

In this case zero is a problematic observation for the distribution; it’s either impossible or infinitely (locally) likely. Some examples:

- Gamma distribution: probability density at zero is infinite (if  $\text{shape} < 1$ ) or zero (if  $\text{shape} > 1$ ); it’s finite only for an exponential distribution ( $\text{shape} = 1$ )
- Lognormal distribution: the probability density at zero is zero.
- Beta distribution: the probability densities at 0 and 1 are zero (if the corresponding shape parameter is  $> 1$ ) or infinite (if  $\text{shape} < 1$ )

The best solution depends very much on the data-generating mechanism.

- If the bad (0/1) values are generated by rounding (e.g. proportions that are too close to the boundaries are reported as being on the boundaries), the simplest solution is to “squeeze” these in slightly, e.g.  $y \rightarrow (y + a)/2a$  for some sensible value of  $a$  (Smithson and Verkuilen 2006)
- If you think that zero values are generated by a separate process, the simplest solution is to fit a Bernoulli model to the zero/non-zero data, then a *conditional* continuous model for the non-zero values; this is effectively a *hurdle model*.
- you might have *censored* data where all values below a certain limit (e.g. a detection limit) are recorded as zero. The `lme4` package (<https://CRAN.R-project.org/package=lme4>) handles *linear* mixed models; `brms` and `GLMMadaptive` both provide support for censored data in mixed models.
- The `cplm` and `glmmTMB` packages handles ‘Tweedie compound Poisson linear models’, which in a particular range of parameters allows for skewed continuous responses with a spike at zero

### Probability density of $x$ positive and finite

In this case (e.g. a spike of zeros in the center of an otherwise continuous distribution), the hurdle model probably makes the most sense.

## Tests for zero-inflation

- you can use a likelihood ratio test between the regular and zero-inflated version of the model, but be aware of boundary issues (search “boundary” elsewhere on this page ...) – the null value (no zero inflation) is on the boundary of the feasible space
- you can use AIC or variations, with the same caveats

- you can use Vuong’s test, which is often recommended for testing zero-inflation in GLMs, because under some circumstances the various model flavors under consideration (hurdle vs zero-inflated vs “vanilla”) are not nested. Vuong’s test is implemented (and referenced) in the `pscl` package, but not for (G)LMMs. However, the `nonnest` package provides an example (in conjunction with the `merDeriv` package) for using its `vuongtest` function with `merMod` objects. (May also work with `glmmTMB`, haven’t tried it ...)
- two untested but reasonable approaches:
  - use a `simulate()` method if it exists to construct a simulated distribution of the proportion of zeros expected overall from your model, and compare it to the observed proportion of zeros in the data set
  - compute the probability of a zero for each observation. On the basis of (conditionally) independent Bernoulli trials, compute the expected number of zeros and the confidence intervals – compare it with the observed number.

## Spatial and temporal correlation models, heteroscedasticity (“R-side” models)

In `nlme` these so-called **R-side** (R for “residual”) structures are accessible via the `weights / VarStruct` (heteroscedasticity) and `correlation / corStruct` (spatial or temporal correlation) arguments and data structures. This extension is a bit harder than it might seem. In LMMs it is a natural extension to allow the residual error terms to be components of a single multivariate normal draw; if that MVN distribution is uncorrelated and homoscedastic (i.e. proportional to an identity matrix) we get the classic model, but we can in principle allow it to be correlated and/or heteroscedastic.

It is not too hard to define marginal correlation structures that don’t make sense. One class of reasonably sensible models is to always assume an observation-level random effect (as `MCMCglmm` does for computational reasons) and to allow that random effect to be MVN on the link scale (so that the full model is lognormal-Poisson, logit-normal binomial, etc., depending on the link function and family).

For example, a relatively simple Poisson model with spatially correlated errors might look like this:

$$\begin{aligned}\eta &\sim \text{MVN}(a + bx, \Sigma) \\ \Sigma_{ij} &= \sigma^2 \exp(-d_{ij}/s) \\ y_i &\sim \text{Poisson}(\lambda = \exp(\eta_i))\end{aligned}$$

That is, the marginal distributions of the response values are Poisson-lognormal, but on the link (log) scale the latent Normal variables underlying the response are *multivariate* normal, with a variance-covariance matrix described by an exponential spatial correlation function with scale parameter  $s$ .

How can one achieve this?

- These types of models are not implemented in `lme4`, for either LMMs or GLMMs; they are fairly low priority, and it is hard to see how they could be implemented for GLMMs (the equivalent for LMMs is tedious but should be straightforward to implement).
- For LMMs, you can use the spatial/temporal correlation structures that are built into `(n)lme`
- You can use the spatial/temporal correlation structures available for `(n)lme`, which include basic geostatistical (space) and ARMA-type (time) models.

```
library(sos)
findFn("corStruct")
```

finds additional possibilities in the `ramps` (extended geostatistical) and `ape` (phylogenetic) packages.

- You can use these structures in GLMMs via `MASS::glmmPQL` (see Dormann et al.)
- `geepack::geeglm`
- `geoR`, `geoRglm` (power tools); these are mostly designed for fitting spatial random field GLMMs via MCMC – not sure that they do random effects other than the spatial random effect
- `R-INLA` (<http://r-inla.org>) (super-power tool)
- it is possible to use AD Model Builder to fit spatial GLMMs, as shown in these AD Model Builder examples (<http://admb-project.org/examples/spatial-models>); this capability is not in the `glmmADMB` package (and may not be for a while!), but it would be possible to run AD Model Builder via the `R2admb` package (requires installing – and learning! ADMB)
- `geoBUGS` (<http://mathstat.helsinki.fi/openbugs/Manuals/GeoBUGS/Manual.html>), the geostatistical/spatial correlation module for

WinBUGS, is another alternative (but again requires going outside of R)

## Penalization/handling complete separation

*Complete separation* occurs in a binary-response model when there is some linear combination of the parameters that perfectly separates failures from successes - for example, when all of the observations are zero for some particular combination of categories. The symptoms of this problem are unrealistically large parameter estimates; ridiculously large Wald standard errors (the *Hauck-Donner effect*); and various warnings.

In particular, binomial `glmer()` models with complete separation can lead to “Downdated VtV is not positive definite” (e.g. see here (<https://github.com/lme4/lme4/issues/483>)) or “PIRLS step-halvings failed to reduce deviance in pwrssUpdate” errors (e.g. see here (<https://github.com/lme4/lme4/issues/179#issuecomment-424445187>)). Roughly speaking, the complete separation is likely to appear even if one considers only the fixed effects part of the model (counterarguments or counterexamples welcome!), suggesting two quick-and-dirty diagnostic methods. If `fixed_form` is the formula including only the fixed effects:

- `summary(g1 <- glm(fixed_form, family=binomial, data=...))` will show one or more of the following symptoms:
  - warnings that `glm.fit: fitted probabilities numerically 0 or 1 occurred`
  - parameter estimates of large magnitude (e.g. `any(abs(g1)>8)`), assuming that predictors are either categorical or scaled to have standard deviations of  $\approx 1$ )
  - extremely large Wald standard errors, and large p-values (*Hauck-Donner effect*)
  - the `brglm2` package has a method for detecting complete separation:
 

```
library("brglm2"); glm(fixed_form, data = ..., family = binomial, method="detect_separation")
```

 This should say whether complete separation occurs, and in which (combinations of) variables, e.g.

```
Separation: TRUE
Existence of maximum likelihood estimates
(Intercept)      height
      Inf         Inf
0: finite value, Inf: infinity, -Inf: -infinity
```

If complete separation is occurring between categories of a single categorical fixed-effect predictor with a large number of levels, one option would be to treat this fixed effect as a random effect, which will allow some degree of shrinkage to the mean. (It might be reasonable to specify the variance of this term *a priori* to a large value [minimal shrinkage], rather than trying to estimate it from the data.)

(**TODO:** worked example)

The general approach to handling complete separation in logistic regression is called *penalized regression*; it's available in the `brglm`, `brglm2`, `logistf`, and `rms` packages. However, these packages don't handle mixed models, so the best available *general* approach is to use a Bayesian method that allows you to set a prior on the fixed effects, e.g. a Gaussian with standard deviation of 3; this can be done in any of the Bayesian GLMM packages (e.g. `blme`, `MCMCglmm`, `brms`, ...) (See supplementary material for Fox et al. 2016 ([http://bbolker.github.io/mixedmodels-misc/ecostats\\_chap.html#digression-complete-separation](http://bbolker.github.io/mixedmodels-misc/ecostats_chap.html#digression-complete-separation)) for a worked example.)

## Non-Gaussian random effects

I'm not aware of easy ways to fit mixed models with non-Gaussian random effects distributions in R (i.e., convenient, flexible, well-tested implementations). McCulloch and Neuhaus (2011) discusses when this misspecification may be important. This presentation (<https://niasra.uow.edu.au/content/groups/public/@web/@inf/@math/documents/mm/uow236296.pdf>) discusses various approaches to solving the problem (e.g. using a Gamma rather than a Normal distribution of REs in log-link models). The `spamm` package implements H-likelihood models (Lee, Nelder, and Pawitan 2017), and claims to allow a range of random-effects distributions (perhaps not well tested though ...)

In principle you can implement any random-effects distribution you want in a fully capable Bayesian modeling language (e.g. JAGS/Stan/PyMC/etc.); see e.g. this StackOverflow answer (<https://stackoverflow.com/questions/45656714/user-defined-random-intercept-distribution-for-glmer>), which uses the `rethinking` package's interface to Stan.

# Estimation

## What methods are available to fit (estimate) GLMMs?

(adapted from Bolker et al TREE 2009)

Method	Advantages	Disadvantages	Packages
Penalized quasi-likelihood	Flexible, widely implemented	Likelihood inference may be inappropriate; biased for large variance or small means	PROC GLIMMIX (SAS), GLMM (GenStat), glmmPQL (R:MASS), ASREML-R
Laplace approximation	More accurate than PQL	Slower and less flexible than PQL	glmer (R:lme4, lme4a), glmm.admb (R:glmmADMB), INLA, glmmTMB, AD Model Builder, HLM
Gauss-Hermite quadrature	More accurate than Laplace	Slower than Laplace; limited to 2-3 random effects	PROC NLMIXED (SAS), glmer (R:lme4, lme4a), glmmML (R:glmmML), xtlogit (Stata)
Markov chain Monte Carlo	Highly flexible, arbitrary number of random effects; accurate	Slow, technically challenging, Bayesian framework	MCMCglmm (R:MCMCglmm), rstanarm (R), brms (R), MCMCpack (R), WinBUGS/OpenBUGS (R interface: BRugs/R2WinBUGS), JAGS (R interface: rjags/R2jags), AD Model Builder (R interface: R2admb), glmm.admb (post hoc MCMC after Laplace fit) (R:glmmADMB)

## Troubleshooting

- double-check the model specification and the data for mistakes
- center and scale continuous predictor variables (e.g. with `scale()`)
- try all available optimizers (e.g. several different implementations of BOBYQA and Nelder-Mead, L-BFGS-B from `optim`, `nlopt`, `nlminb()`, ...). While this will of course be slow for large fits, we consider it the gold standard; if all optimizers converge to values that are practically equivalent (it's up to the user to decide what "practically equivalent means for their case"), then we would consider the model fit to be good enough. For example:

```
modelfit.all <- lme4::allFit(model)
ss <- summary(modelfit.all)
```

## Convergence warnings

Most of the current advice about troubleshooting `lme4` convergence problems can be found in the help page `?convergence`. That page explains that the convergence tests in the current version of `lme4` (1.1-11, February 2016) generate lots of false positives. We are considering raising the gradient warning threshold to 0.01 in future releases of `lme4`. In addition to the general troubleshooting tips above:

- double-check the Hessian calculation with the more expensive Richardson extrapolation method (see examples)
- restart the fit from the apparent optimum, or from a point perturbed slightly away from the optimum (`getME(model, c("theta", "beta"))` should retrieve the parameters in a form suitable to be used as the `start` parameter)
- a common error is to specify an offset to a log-link model as a raw searching-effort value, i.e. `offset(effort)` rather than `offset(log(effort))`. While the intention is to fit a model where counts  $\propto$  effort, specifying `offset(effort)` leads to a model

where counts  $\propto \exp(\text{effort})$  instead;  $\exp(\text{effort})$  is often a huge (and model-destabilizing) number.

## Singular models: random effect variances estimated as zero, or correlations estimated as +/- 1

It is very common for overfitted mixed models to result in singular fits. Technically, singularity means that some of the  $\theta$  (variance-covariance Cholesky decomposition) parameters corresponding to diagonal elements of the Cholesky factor are exactly zero, which is the edge of the feasible space, or equivalently that the variance-covariance matrix has some zero eigenvalues (i.e. is positive semidefinite rather than positive definite), or (*almost* equivalently) that some of the variances are estimated as zero or some of the correlations are estimated as +/-1. This commonly occurs in two scenarios:

- small numbers of random-effect levels (e.g. <5), as illustrated in these simulations (<http://rpubs.com/bbolker/4187>) and discussed (in a somewhat different, Bayesian context) by Gelman (2006).
- complex random-effects models, e.g. models of the form  $(f|g)$  where  $f$  is a categorical variable with a relatively large number of levels, or models with several different random-slopes terms.
- When using `lme4`, singularity is most obviously detectable in the output of `summary.merMod()` or `VarCorr.merMod()` when a variance is estimated as 0 (or very small, i.e. orders of magnitude smaller than other variance components) or when a correlation is estimated as exactly  $\pm 1$ . However, as pointed out by D. Bates, Kliegl, et al. (2015), singularities in larger variance-covariance matrices can be hard to detect: checking for small values among the diagonal elements of the Cholesky factor is a good start.

```
theta <- getME(model,"theta")
## diagonal elements are identifiable because they are fitted
## with a lower bound of zero ...
diag.element <- getME(model,"lower")==0
any(theta[diag.element]<1e-5)
```

As of `lme4` version 1.1-19, this functionality is available as `isSingular(model)`. - In `MCMCglmm`, singular or near-singular models will provoke an error and a requirement to specify a stronger prior.

At present there are a variety of strong opinions about how to resolve such problems. Briefly:

- Barr et al. (2013) suggest always starting with the maximal model (i.e. the most random-effects component of the model that is *theoretically* identifiable given the experimental design) and then dropping terms when singularity or non-convergence occurs (please see the paper for detailed recommendations ...)
- Matuschek et al. (2017) and D. Bates, Kliegl, et al. (2015) strongly disagree, suggesting that models should be simplified *a priori* whenever possible; they also provide tools (<https://github.com/dmbates/RePsychLing>) for diagnosing and mitigating singularity.
- One alternative (suggested by Robert LaBudde) for the small-numbers-of-levels scenario is to “fit the model with the random factor as a fixed effect, get the level coefficients in the sum to zero form, and then compute the standard deviation of the coefficients.” This is appropriate for users who are (a) primarily interested in measuring variation (i.e. the random effects are not just nuisance parameters, and the variability [rather than the estimated values for each level] is of scientific interest), (b) unable or unwilling to use other approaches (e.g. MCMC with half-Cauchy priors in WinBUGS), (c) unable or unwilling to collect more data. For the simplest case (balanced, orthogonal, nested designs with normal errors) these estimates of standard deviations should equal the classical method-of-moments estimates.
- Bayesian approaches allow the user to specify a informative prior that avoids singularity.
  - The `blme` package (Chung et al. 2013) provides a wrapper for the `lme4` machinery that adds a particular form of weak prior to get an approximate a Bayesian maximum *a posteriori* estimate that avoids singularity.
  - The `MCMCglmm` package allows for priors on the variance-covariance matrix
  - The `rstanarm` and `brms` packages provide wrappers for the Stan Hamiltonian MCMC engine that fit GLMMs via `lme4` syntax, again allowing a variety of priors to be set.
- If a variance component is zero, dropping it from the model will have no effect on any of the estimated quantities (although it will affect the AIC, as the variance parameter is counted even though it has no effect). Pasch, Bolker, and Phelps (2013) gives one example where random effects were dropped because the variance components were consistently estimated as zero. Conversely, if one chooses for philosophical grounds to retain these parameters, it won't change any of the answers.



## Setting residual variances to a fixed value (zero or other)

For some problems it would be convenient to be able to set the residual variance term to zero, or a fixed value. This is difficult in `lme4`, because the model is parameterized internally in such a way that the residual variance is profiled out (i.e., calculated directly from a residual deviance term) and the random-effects variances are scaled by the residual variance.

Searching the `r-sig-mixed-models` list for “fix residual variance” (<https://www.google.ca/search?q=site%3A%2Fstat.ethz.ch%2Fpipermail%2Fr-sig-mixed-models%2Ffix+residual+variance>)

- This is done in the `metafor` package, for meta-analytic models
- You can use the `blme` package to fix the residual variance: from Vincent Dorie,

```
library(blme)
blmer(formula = y ~ 1 + (1 | group), weights = V,
       resid.prior = point(1.0), cov.prior = NULL)
```

This sets the residual variance to 1.0. You *cannot* use this to make it exactly zero, but you can make it very small (and experiment with setting it to different small values, e.g. 0.001 vs 0.0001, to see how sensitive the results are). - Similarly, you can fix the residual variance to a small positive value in `[n]lme` via the `control()` argument (Heisterkamp et al. 2017):

```
nlme::lme(Reaction~Days, random=~1|Subject,
          data=lme4::sleepstudy,
          control=list(sigma=1e-8))
```

- the `glmmTMB` package can set the residual variance to zero, by specifying `dispformula = ~0`
- There is an `rrBlupMethod6` package (<https://CRAN.R-project.org/package=rrBlupMethod6>) on CRAN (“Re-parametrization of mixed model formulation to allow for a fixed residual variance when using RR-BLUP for genom[e]wide estimation of marker effects”), but it seems fairly special-purpose.
- it might be possible *in principle* to adapt `lme4`’s internal `devfun2()` function (used in the likelihood profiling computation for LMMs), which uses a specified value of the residual standard deviation in computing likelihood, but as D. Bates, Mächler, et al. (2015) say:

The resulting function is not useful for general nonlinear optimization — one can easily wander into parameter regimes corresponding to infeasible (non-positive semidefinite) variance-covariance matrices — but it serves for likelihood profiling, where one focal parameter is varied at a time and the optimization over the other parameters is likely to start close to an optimum.

## Other problems/ `lme4` error messages

Most of the following error messages are relatively unusual, and happen mostly with complex/large/unstable models. There is often no simple fix; the standard suggestions for troubleshooting are (1) try rescaling and/or centering predictors; (2) see if a simpler model can be made to work; (3) look for severe lack of balance and/or complete separation in the data set.

- PIRLS step-halvings failed to reduce deviance in `pwrssUpdate`
  - this can also occur due to complete or quasi-complete separation (see Penalization/handling complete separation)
  - When using `lme4` to fit GLMMs with link functions that do not automatically constrain the response to the allowable range of the distributional family (e.g. binomial models with a log link, where the estimated probability can be  $>1$ , or inverse-Gamma models, where the estimated mean can be negative), it is not unusual to get this error. This occurs because `lme4` doesn’t do anything to constrain the predicted values, so `NaN` values pop up, which aren’t handled gracefully. If possible, switch to a link function to one that constrains the response (e.g. logit link for binomial or log link for Gamma).
  - otherwise this message often occurs when there is something else wrong with the model or data, e.g. - a model fitted to underdispersed data includes both a negative binomial response and observation-level random effects (<https://stackoverflow.com/questions/28036334/using-glmer-nb-the-error-messagemaxstephalfit-pirls-step-halvings-failed-t>) - negative response values for a link function that doesn’t allow them (<https://stackoverflow.com/questions/37533825/error->

maxstephalfit-pirls-step-halvings-failed-to-reduce-deviance-in-pwrssupd)

- Downdated VtV is not positive definite: no specific advice, see general suggestions above
- convergence code 3 from bobyqa: bobyqa -- a trust region step failed to reduce q: again no specific advice about fixing this, although there is a useful discussion of the meaning of the error message on CrossValidated (<https://stats.stackexchange.com/questions/89945/meaning-of-a-convergence-warning-in-glmer>)
- Warning message: In theta.ml(Y, mu, weights = object@resp\$weights, limit = limit, : iteration limit reached. This warning occurs within glmer.nb during the step that generates an initial estimate for the dispersion parameter. It typically means that the data (i.e. the conditional distribution of the response variable) is *equidispersed* (variance = mean) or *underdispersed* (variance < mean), so that the estimate of the dispersion parameter is becoming extremely large/infinite. This SO question (<https://stackoverflow.com/a/70130208/190277>) and the section of this document on underdispersion (<https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#underdispersion>) provide more information.

## REML for GLMMs

- While restricted maximum likelihood (REML) procedures (Wikipedia ([http://en.wikipedia.org/wiki/Restricted\\_maximum\\_likelihood](http://en.wikipedia.org/wiki/Restricted_maximum_likelihood))) are well established for linear mixed models, it is less clear how one should define and compute the equivalent criteria (integrating out the effects of fixed parameters) for GLMMs. Millar (2011) and Berger, Liseo, and Wolpert (1999) are possible starting points in the peer-reviewed literature, and there are mailing-list discussions of these issues here (<https://stat.ethz.ch/pipermail/r-sig-mixed-models/2009q1/002104.html>) and here (<http://lists.admb-project.org/pipermail/users/2011-June/001224.html>).
- Attempting to use REML=TRUE with glmer will produce the warning extra argument(s) 'REML' disregarded
- glmmTMB allows REML=TRUE for GLMMs (it uses the Laplace approximation to integrate over the fixed effect parameters), since version 0.2.2

## Model diagnostics

## Inference and confidence intervals

## Testing hypotheses

### What are the p-values listed by summary(glmerfit) etc.? Are they reliable?

By default, in keeping with the tradition in analysis of generalized linear models, lme4 and similar packages display the Wald Z-statistics for each parameter in the model summary. These have one big advantage: they're convenient to compute. However, they are asymptotic approximations, assuming both that (1) the sampling distributions of the parameters are multivariate normal (or equivalently that the log-likelihood surface is quadratic) and that (2) the sampling distribution of the log-likelihood is (proportional to)  $\chi^2$ . The second approximation is discussed further under "Degrees of freedom". The first assumption usually requires an even greater leap of faith, and is known to cause problems in some contexts (for binomial models failures of this assumption are called the *Hauck-Donner effect*), especially with extreme-valued parameters.

### Methods for testing single parameters

From worst to best:

- Wald Z-tests
- **For balanced, nested LMMs** where degrees of freedom can be computed according to classical rules: Wald t-tests
- Likelihood ratio test, either by setting up the model so that the parameter can be isolated/dropped (via anova or drop1, or via computing likelihood profiles)
- Markov chain Monte Carlo (MCMC) or parametric bootstrap confidence intervals

### Tests of effects (i.e. testing that several parameters are simultaneously zero)

From worst to best:

- Wald chi-square tests (e.g. `car::Anova`)
- Likelihood ratio test (via `anova` or `drop1`)
- **For balanced, nested LMMs** where df can be computed: conditional F-tests
- **For LMMs:** conditional F-tests with df correction (e.g. Kenward-Roger in `pbkrtest` package: see notes on K-R etc below).
- MCMC or parametric, or nonparametric, bootstrap comparisons (nonparametric bootstrapping must be implemented carefully to account for grouping factors: see e.g. the REB or 'case bootstrap' methods in the `lmeresampler` package)

## Is the likelihood ratio test reliable for mixed models?

- It depends.
- Not for fixed effects in finite-size cases (see Pinheiro and Bates (2000)): may depend on 'denominator degrees of freedom' (number of groups) and/or total number of samples - total number of parameters
- Conditional F-tests are preferred for LMMs, **if** denominator degrees of freedom are known

## Why doesn't lme4 display denominator degrees of freedom/p values? What other options do I have?

There is an R FAQ entry ([http://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-are-p\\_002dvalues-not-displayed-when-using-lmer\\_0028\\_0029\\_003f](http://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-are-p_002dvalues-not-displayed-when-using-lmer_0028_0029_003f)) on this topic, which links to a mailing list post (<https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html>) by Doug Bates (there is also a voluminous mailing list thread (<http://rwiki.sciviews.org/doku.php?id=guides:lmer-tests>) reproduced on the R wiki). The bottom line is

- For special cases that correspond to classical experimental designs (i.e. balanced designs that are nested, split-plot, randomized block, etc.) ... we can show that the null distributions of particular ratios of sums of squares follow an  $F$  distribution with known numerator and denominator degrees of freedom (and hence the sampling distributions of particular contrasts are t-distributed with known df). In more complicated situations (unbalanced, GLMMs, crossed random effects, models with temporal or spatial correlation, etc.) it is not in general clear that the null distribution of the computed ratio of sums of squares is really an F distribution, for any choice of denominator degrees of freedom.
- For each simple degrees-of-freedom recipe that has been suggested (trace of the hat matrix, etc.) there seems to be at least one fairly simple counterexample where the recipe fails badly (e.g. see this r-help thread from September 2006 (<https://stat.ethz.ch/pipermail/r-help/2006-September/112495.html>)).
- When the responses are normally distributed and the design is balanced, nested etc. (i.e. the classical LMM situation), the scaled deviances and differences in deviances are exactly  $F$ -distributed and looking at the experimental design (i.e., which treatments vary/are replicated at which levels) tells us what the relevant degrees of freedom are (see "df alternatives" below)
- Two approaches to approximating df (Satterthwaite and Kenward-Roger) have been implemented in R, Satterthwaite in `lmerTest` and Kenward-Roger in `pbkrtest` (as `KRmodcomp`) (various packages such as `lmerTest`, `emmeans`, `car`, etc., import `pbkrtest::get_Lb_ddf`).
  - K-R is probably the most reliable option (Schaalje, McBride, and Fellingham 2002), although it may be prohibitively computationally expensive for large data sets.
  - K-R was derived for LMMs (and for REML?) in particular, it isn't clear how it would apply to GLMMs. Stroup (2014) states (referencing Stroup (2013)) that K-R actually works reasonably well for GLMMs (K-R is not implemented in R for GLMMs; Stroup suggests that a pseudo-likelihood (Wolfinger and O'Connell 1993) approach is necessary in order to implement K-R for GLMMs):

Notice the non-integer values of the denominator df. They, and the  $F$  and  $p$  values, reflect the procedure developed by Kenward and Roger (2009) to account for the effect of the covariance structure on degrees of freedom and standard errors. Although the Kenward–Roger adjustment was derived for the LMM with normally distributed data and is an ad hoc procedure for GLMMs with non-normal data, informal simulation studies consistently have suggested that the adjustment is accurate. The Kenward-Roger adjustment requires that the SAS GLIMMIX default computing algorithm, pseudo-likelihood, be used rather than the Laplace algorithm used to obtain AICC statistics. Stroup (2013b) found that for binomial and Poisson GLMMs, pseudo-likelihood with the Kenward–Roger adjustment yields better Type I error control than Laplace while preserving the GLMM’s advantage with respect to power and accuracy in estimating treatment means.

- There are several different issues at play in finite-size (small-sample) adjustments, which apply slightly differently to LMMs and GLMMs.
  - When the data don’t fit into the classical framework (crossed, unbalanced, R-side effects), we might still guess that the deviances etc. are approximately F-distributed but that we don’t know the real degrees of freedom – this is what the Satterthwaite, Kenward-Roger, Fai-Cornelius, etc. approximations are supposed to do.
  - When the responses are not normally distributed (as in GLMs and GLMMs), and when the scale parameter is not estimated (as in standard Poisson- and binomial-response models), then the deviance differences are only asymptotically F- or chi-square-distributed (i.e. not for our real, finite-size samples). In standard GLM practice, we usually ignore this problem; there is some literature on finite-size corrections for GLMs under the rubrics of “Bartlett corrections” and “higher order asymptotics” (see McCullagh and Nelder (1989), Cordeiro, Paula, and Botter (1994), Cordeiro and Ferrari (1998) and the `cond` package (on CRAN (<https://cran.r-project.org/package=cond>)) [which works with GLMs, not GLMMs]), but it’s rarely used. (The bias correction/Firth approach implemented in the `brglm` package attempts to address the problem of finite-size bias, not finite-size non-chi-squaredness of the deviance differences.)
  - When the scale parameter in a GLM is estimated rather than fixed (as in Gamma or quasi-likelihood models), it is sometimes recommended to use an  $F$  test to account for the uncertainty of the scale parameter (e.g. Venables and Ripley (2002) recommend `anova(..., test="F")` for quasi-likelihood models)
  - Combining these issues, one has to look pretty hard for information on small-sample or finite-size corrections for GLMMs: Feng, Braun, and McCulloch (2004) and Bell and Grunwald (2010) look like good starting points, but it’s not at all trivial.

### Df alternatives:

- use MASS::glmmPQL (uses old `nlme` rules approximately equivalent to SAS ‘inner-outer’/‘within-between’ rules) for GLMMs, or `(n)lme` for LMMs
- Guess the denominator df from standard rules (for standard designs, e.g. see Gotelli and Ellison (2004)) and apply them to  $t$  or  $F$  tests
- Run the model in `lme` (if possible) and use the denominator df reported there (which follow a simple ‘inner-outer’ rule which should correspond to the canonical answer for simple/orthogonal designs), applied to  $t$  or  $F$  tests. For the explicit specification of the rules that `lme` uses, see page 91 of Pinheiro and Bates (*this page was previously available on Google Books (<http://tinyurl.com/ntygq3>), but the link is no longer useful, so here are the relevant paragraphs*):

These conditional tests for fixed-effects terms require denominator degrees of freedom. In the case of the conditional  $F$ -tests, the numerator degrees of freedom are also required, being determined by the term itself. The denominator degrees of freedom are determined by the grouping level at which the term is estimated. A term is called inner relative to a factor if its value can change within a given level of the grouping factor. A term is outer to a grouping factor if its value does not change within levels of the grouping factor. A term is said to be estimated at level  $i$ , if it is inner to the  $i - 1$ st grouping factor and outer to the  $i$ th grouping factor. For example, the term `Machine` in the `fm2Machine` model is outer to `Machine %in% Worker` and inner to `Worker`, so it is estimated at level 2 (`Machine %in% Worker`). If a term is inner to all  $Q$  grouping factors in a model, it is estimated at the level of the within-group errors, which we denote as the  $Q + 1$ st level.

The intercept, which is the parameter corresponding to the column of all 1's in the model matrices  $X_i$ , is treated differently from all the other parameters, when it is present. As a parameter it is regarded as being estimated at level 0 because it is outer to all the grouping factors. However, its denominator degrees of freedom are calculated as if it were estimated at level  $Q + 1$ . This is because the intercept is the one parameter that pools information from all the observations at a level even when the corresponding column in  $X_i$  doesn't change with the level.

Letting  $m_i$  denote the total number of groups in level  $i$  (with the convention that  $m_0 = 1$  when the fixed effects model includes an intercept and 0 otherwise, and  $m_{Q+1} = N$ ) and  $p_i$  denote the sum of the degrees of freedom corresponding to the terms estimated at level  $i$ , the  $i$ th level denominator degrees of freedom is defined as

$$\text{denDF}_i = m_i - (m_{i-1} + p_i), i = 1, \dots, Q$$

This definition coincides with the classical decomposition of degrees of freedom in balanced, multilevel ANOVA designs and gives a reasonable approximation for more general mixed-effects models.

Note that the implementation used in `lme` **gets the wrong answer for random-slopes models**:

```
library(nlme)
lmeDF <- function(formula=distance~age,random=~1|Subject) {
  mod <- lme(formula,random,data=Orthodont)
  aa <- anova(mod)
  return(setNames(aa[, "denDF"],rownames(aa)))
}
lmeDF()
```

```
## (Intercept)      age
##           80      80
```

```
lmeDF(random=~age|Subject) ## wrong!
```

```
## (Intercept)      age
##           80      80
```

I (BB) have re-implemented this algorithm in a way that does slightly better for random-slopes models (but may still get confused!), see here (R/calcDenDF.R).

```
source("R/calcDenDF.R")
calcDenDF(~age,"Subject",nlme::Orthodont)
```

```
## (Intercept)      age
##           80      80
```

```
calcDenDF(~age,data=nlme::Orthodont,random=~1|Subject)
```

```
## (Intercept)      age
##           80      80
```

```
calcDenDF(~age,data=nlme::Orthodont,random=~age|Subject) ## off by 1
```

```
## (Intercept)      age
##           81      25
```

- use SAS, Genstat (AS-REML), Stata?
- Assume infinite denominator df (i.e.  $Z/\chi^2$  test rather than  $t/F$ ) if number of groups is large (>45? Various rules of thumb for how large is “approximately infinite” have been posed, including (in Angrist and Pischke 2009), 42 (in homage to Douglas Adams)

## Testing significance of random effects

- the most common way to do this is to use a likelihood ratio test, i.e. fit the full and reduced models (the reduced model is the model with the focal variance(s) set to zero). For example:

```
library(lme4)
m2 <- lmer(Reaction~Days+(1|Subject)+(0+Days|Subject),sleepstudy,REML=FALSE)
m1 <- update(m2,~Days+(1|Subject))
m0 <- lm(Reaction~Days,sleepstudy)
anova(m2,m1,m0) ## two sequential tests
```

```
## Data: sleepstudy
## Models:
## m0: Reaction ~ Days
## m1: Reaction ~ Days + (1 | Subject)
## m2: Reaction ~ Days + (1 | Subject) + (0 + Days | Subject)
##      npar    AIC    BIC logLik deviance   Chisq Df Pr(>Chisq)
## m0      3 1906.3 1915.9 -950.15   1900.3
## m1      4 1802.1 1814.8 -897.04   1794.1 106.214  1 < 2.2e-16 ***
## m2      5 1762.0 1778.0 -876.00   1752.0  42.075  1 8.782e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With recent versions of `lme4`, goodness-of-fit (deviance) can be compared between `(g)lmer` and `(g)lm` models, although `anova()` must be called with the mixed `(g)lmer` model listed first. Keep in mind that LRT-based null hypothesis tests are conservative when the null value (such as  $\sigma^2 = 0$ ) is on the boundary of the feasible space (Self and Liang 1987; Stram and Lee 1994; Goldman and Whelan 2000); in the simplest case (single random effect variance), the p-value is approximately twice as large as it should be (Pinheiro and Bates 2000).

- Consider *not* testing the significance of random effects. If the random effect is part of the experimental design, this procedure may

be considered ‘sacrificial pseudoreplication’ (Hurlbert 1984). Using stepwise approaches to eliminate non-significant terms in order to squeeze more significance out of the remaining terms is dangerous in any case.

- consider using the `RLRsim` package, which has a fast implementation of simulation-based tests of null hypotheses about zero variances, for simple tests. (However, it only applies to `lmer` models, and is a bit tricky to use for more complex models.)

```
library(RLRsim)
## compare m0 and m1
exactLRT(m1,m0)
```

```
##
## simulated finite sample distribution of LRT. (p-value based on 10000
## simulated values)
##
## data:
## LRT = 106.21, p-value < 2.2e-16
```

```
## compare m1 and m2
mA <- update(m2,REML=TRUE)
m0B <- update(mA, . ~ . - (0 + Days|Subject))
m.slope <- update(mA, . ~ . - (1|Subject))
exactRLRT(m0=m0B,m=m.slope,mA=mA)
```

```
##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 42.796, p-value < 2.2e-16
```

- Parametric bootstrap: fit the reduced model, then repeatedly simulate from it and compute the differences between the deviance of the reduced and the full model for each simulated data set. Compare this null distribution to the observed deviance difference. This procedure is implemented in the `pbkrtest` package (messages and warnings suppressed).

```
(pb <- pbkrtest::PBmodcomp(m2,m1,seed=101))
```

```
## Bootstrap test; time: 7.43 sec; samples: 996; extremes: 0;
## Requested samples: 996 Used samples: 514 Extremes: 0
## large : Reaction ~ Days + (1 | Subject) + (0 + Days | Subject)
## Reaction ~ Days + (1 | Subject)
##          stat df    p.value
## LRT      42.075  1 8.782e-11 ***
## PBtest 42.075    0.001942 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Standard errors of variance estimates

- Paraphrasing Doug Bates: the sampling distribution of variance estimates is in general strongly asymmetric: the standard error may be a poor characterization of the uncertainty.
- `lme4` allows for computing likelihood profiles of variances and computing confidence intervals on their basis; these likelihood profile confidence intervals are subject to the usual caveats about the LRT with finite sample sizes.
- Using an MCMC-based approach (the simplest/most canned is probably to use the `MCMCglmm` package, although its model specifications are not identical to those of `lme4`) will provide posterior distributions of the variance parameters: quantiles or credible

intervals (`HPDinterval()` in the `coda` package) will characterize the uncertainty.

- (don't say we didn't warn you ...) `[n]lme` fits contain an element called `apVar` which contains the approximate variance-covariance matrix (derived from the Hessian, the matrix of (numerically approximated) second derivatives of the likelihood (REML?) at the maximum (restricted?) likelihood values): you can derive the standard errors from this list element via `sqrt(diag(lme.obj$apVar))`. For whatever it's worth, though, these estimates might not match (<http://www.biostat.wustl.edu/archives/html/s-news/2003-07/msg00127.html>) the estimates that SAS gives (<http://www.tau.ac.il/cc/pages/docs/sas8/stat/chap41/sect25.htm#mixedcpe>) which are supposedly derived in the same way.
- it's not a full solution, but there is some more information here (<https://rpubs.com/bbolker/waldvar>). I have some delta-method computations there that are off by a factor of 2 for the residual standard deviation, as well as some computations based on reparameterizing the deviance function.

## P-values: MCMC and parametric bootstrap

Abandoning the approximate  $F/t$ -statistic route, one ends up with the more general problem of estimating  $p$ -values. There is a wider range of options here, although many of them are computationally intensive ...

### Markov chain Monte Carlo sampling:

- pseudo-Bayesian: post-hoc sampling, typically (1) assuming flat priors and (2) starting from the MLE, possibly using the approximate variance-covariance estimate to choose a candidate distribution
  - via `mcmcscamp` (if available for your problem: i.e. LMMs with simple random effects – not GLMMs or complex random effects)
  - via `pvals.fnc` in the `languageR` package, a wrapper for `mcmcscamp`
  - in AD Model Builder, possibly via the `glmmADMB` package (use the `mcmc=TRUE` option) or the `R2admb` package (write your own model definition in AD Model Builder), or outside of R
  - via the `sim` function from the `arm` package (simulates the posterior only for the beta (fixed-effect) coefficients; not yet working with development `lme4`; would like a better formal description of the algorithm ...?)
- fully Bayesian approaches
  - via the `MCMCglmm` package
  - `glmmBUGS` (a WinBUGS wrapper/R interface)
  - JAGS/WinBUGS/OpenBUGS etc., via the `rjags` / `r2jags` / `R2WinBUGS` / `BRugs` packages

### Status of `mcmcscamp`

`mcmcscamp` is a function for `lme4` that is supposed to sample from the posterior distribution of the parameters, based on flat/improper priors for the parameters [ed: I believe, but am not sure, that these priors are flat **on the scale of the theta (Cholesky-factor) parameters**]. At present, in the CRAN version (`lme4` 0.999999-0) and the R-forge “stable” version (`lme4` 0.999999-1), this covers only linear mixed models with uncorrelated random effects.

As has been discussed in a variety of places (e.g. on *r-sig-mixed* models (<http://article.gmane.org/gmane.comp.lang.r.lme4.devel/1788/>), and on the *r-forge* bug tracker ([https://r-forge.r-project.org/tracker/?func=detail&aid=68&group\\_id=60&atid=298](https://r-forge.r-project.org/tracker/?func=detail&aid=68&group_id=60&atid=298)), it is challenging to come up with a sampler that accounts properly for the possibility that the posterior distributions for some of the variance components may be mixtures of point masses at zero and continuous distributions. Naive samplers are likely to get stuck at or near zero. Doug Bates has always been a bit unsure that `mcmcscamp` is really performing as intended, even in the limited cases it now handles.

Given this uncertainty about how even the basic version works, the `lme4` developers have been reluctant to make the effort to extend it to GLMMs or more complex LMMs, or to implement it for the development version of `lme4` ... so unless something miraculous happens, it will not be implemented for the new version of `lme4`. As always, users are encouraged to write and share their own code that implements these capabilities ...

### Parametric bootstrap

The idea here is that in order to do inference on the effect of (a) predictor(s), you (1) fit the reduced model (without the predictors) to the data; (2) many times, (2a) simulate data from the reduced model; (2b) fit both the reduced and the full model to the simulated (null) data; (2c) compute some statistic(s) [e.g.  $t$ -statistic of the focal parameter, or the log-likelihood or deviance difference between the models]; (3) compare the observed values of the statistic from fitting your full model to the data to the null distribution generated in step 2. -

`pbmodcomp` in the `pbkrtest` package - see the example in `help("simulate-mer")` in the `lme4` package to roll your own, using a combination of `simulate()` and `refit()` - `bootMer` in `lme4` version >1.0.0 - a presentation at UseR! 2009 (abstract



(<http://www.agrocampus-ouest.fr/math/useR-2009/abstracts/pdf/SanchezEspigares+Ocana.pdf>), slides (<http://www.agrocampus-ouest.fr/math/useR-2009/slides/SanchezEspigares+Ocana.pdf>) went into detail about a proposed `bootMer` package and suggested it could work for GLMMs too – but it does not seem to be active.

## Predictions and/or confidence (or prediction) intervals on predictions

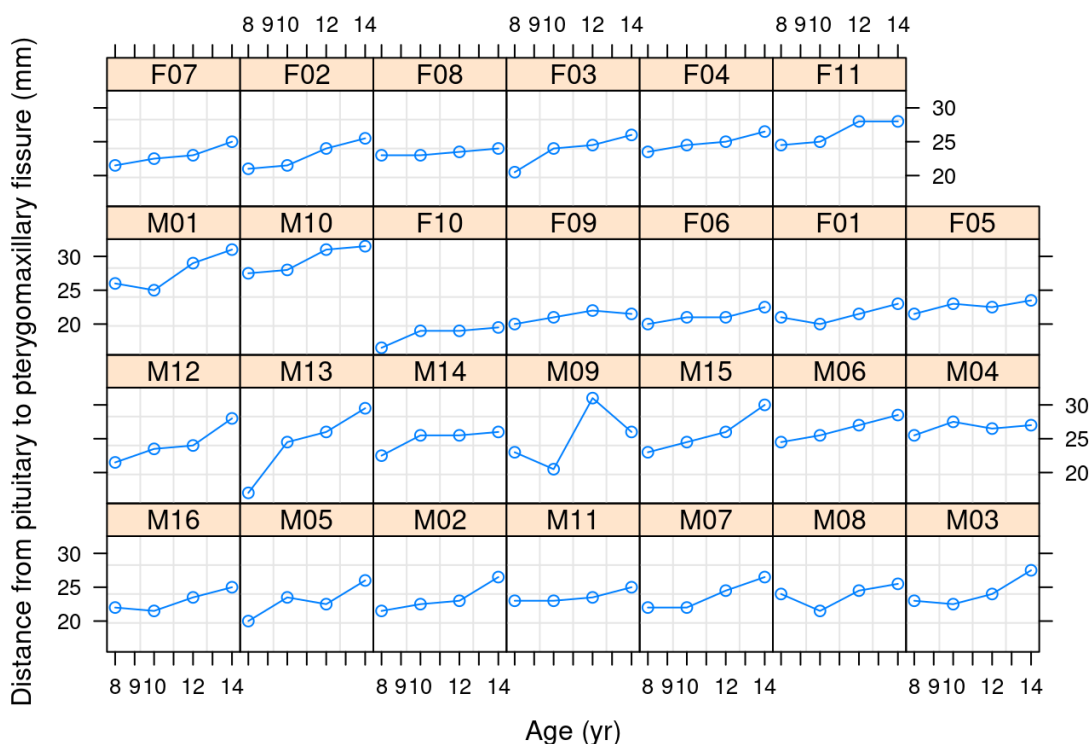
Note that none of the following approaches takes the uncertainty of the random effects parameters into account ... if you want to take RE parameter uncertainty into account, a Bayesian approach is probably the easiest way to do it.

The general recipe for computing predictions from a linear or generalized linear model is to

- figure out the model matrix  $X$  corresponding to the new data;
- matrix-multiply  $X$  by the parameter vector  $\beta$  to get the predictions (or linear predictor in the case of GLM(M)s);
- extract the variance-covariance matrix of the parameters  $V$
- compute  $XVX'$  to get the variance-covariance matrix of the predictions;
- extract the diagonal of this matrix to get variances of predictions;
- if computing prediction rather than confidence intervals, add the residual variance;
- take the square-root of the variances to get the standard deviations (errors) of the predictions;
- compute confidence intervals based on a Normal approximation;
- for GL(M)Ms, run the confidence interval boundaries (not the standard errors) through the inverse-link function.

## lme

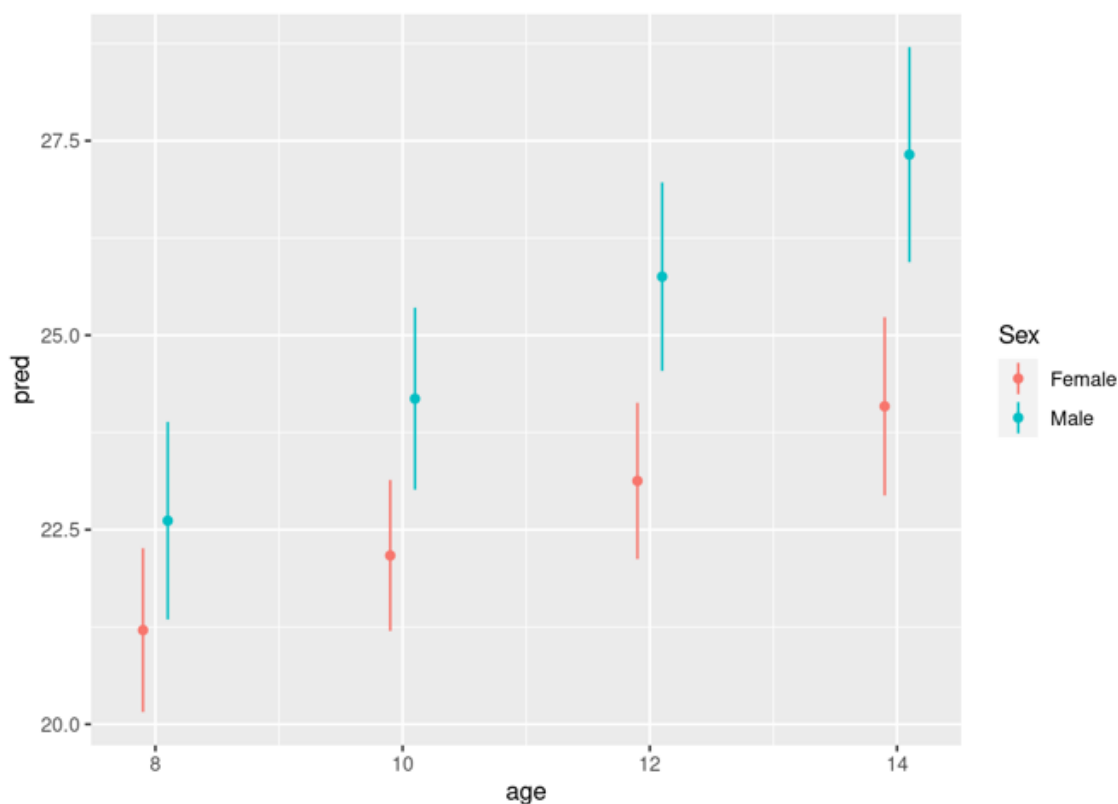
```
library(nlme)
fml <- lme(distance ~ age*Sex, random = ~ 1 + age | Subject,
          data = Orthodont)
plot(Orthodont, asp="fill") ## plot responses by individual
```



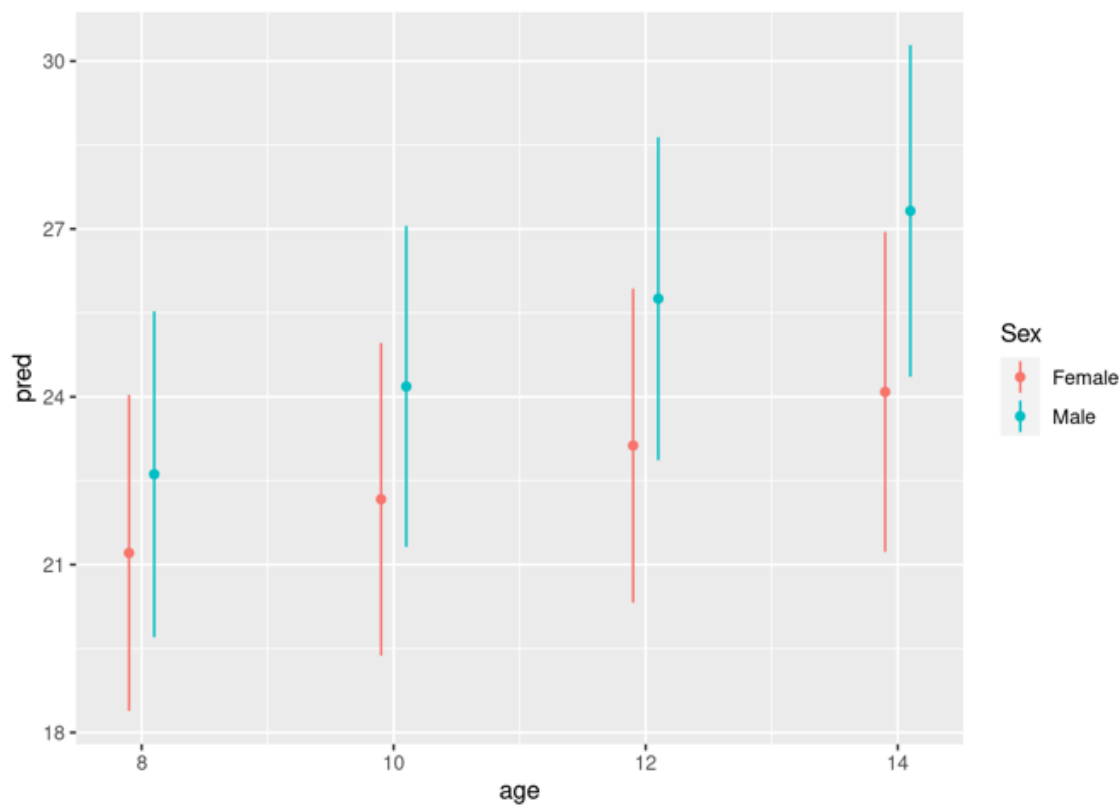
```
## note that expand.grid() orders factor levels by *order of
## appearance* -- must match levels(Orthodont$Sex)
newdat <- expand.grid(age=c(8,10,12,14), Sex=c("Female","Male"))
newdat$pred <- predict(fml, newdat, level = 0)

## [-2] drops response from formula
Designmat <- model.matrix(formula(fml)[-2], newdat)
predvar <- diag(Designmat %*% vcov(fml) %*% t(Designmat))
newdat$SE <- sqrt(predvar)
newdat$SE2 <- sqrt(predvar+fml$sigma^2)

library(ggplot2)
pd <- position_dodge(width=0.4)
g0 <- ggplot(newdat,aes(x=age,y=pred,colour=Sex))+
  geom_point(position=pd)
cmult <- 2 ## could use 1.96 instead
g0 + geom_linerange(aes(ymin=pred-cmult*SE,ymax=pred+cmult*SE), position=pd)
```



```
## prediction intervals
g0 + geom_linerange(aes(ymin=pred-cmult*SE2,ymax=pred+cmult*SE2), position=pd)
```



A similar answer is laid out in the responses to this StackOverflow question (<http://stackoverflow.com/questions/14358811/extract-prediction-band-from-lme-fit>).

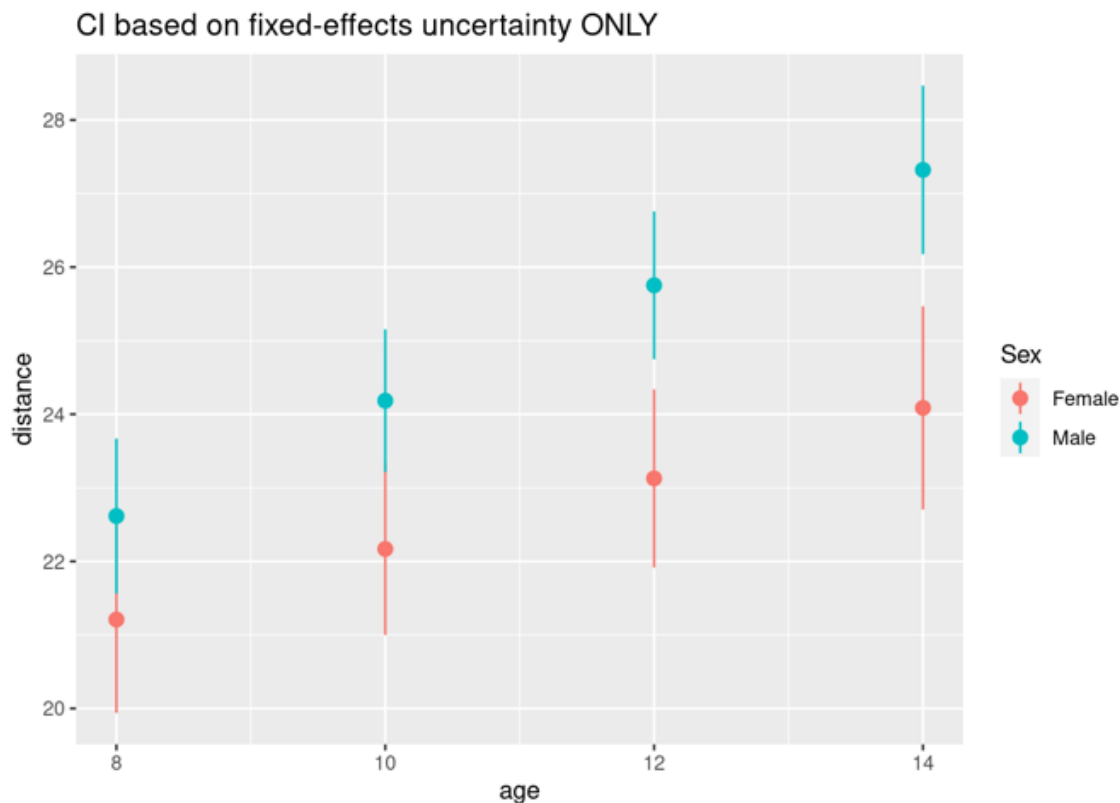
## lme4

Current versions of lme4 have a `predict` method.

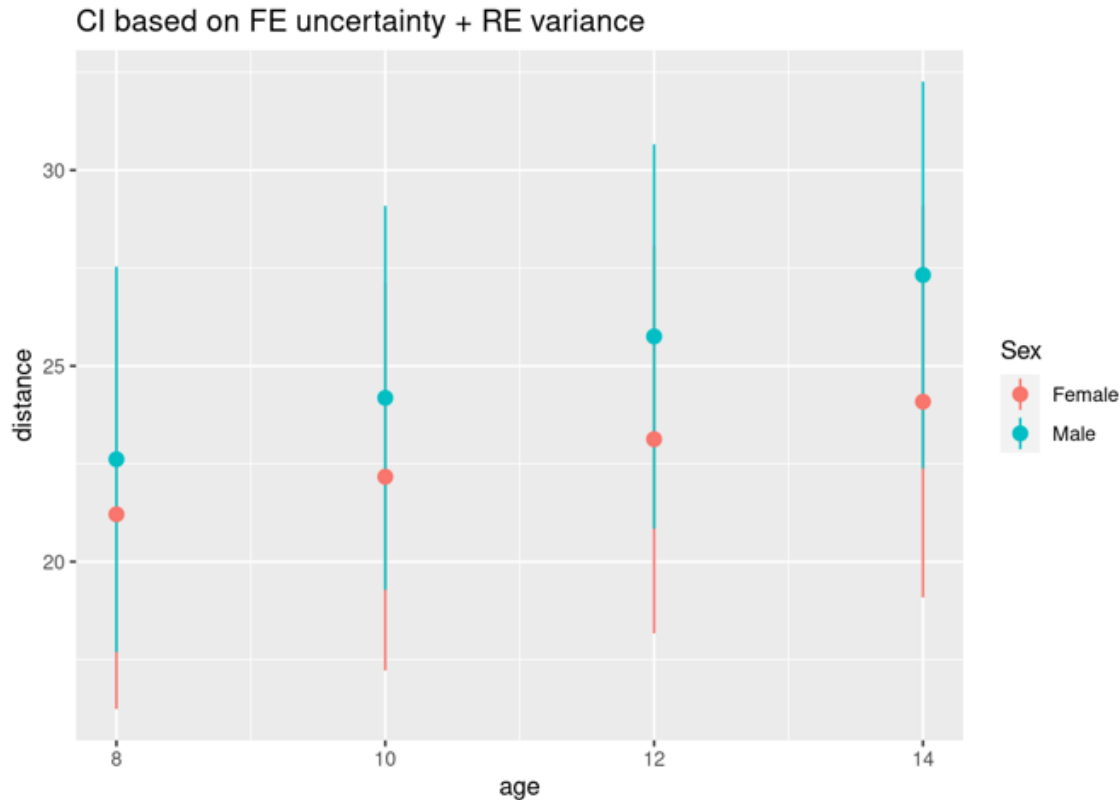
```

library(lme4)
library(ggplot2)
data("Orthodont", package="MEMSS")
fml <- lmer(
  formula = distance ~ age*Sex + (age|Subject)
  , data = Orthodont
)
newdat <- expand.grid(
  age=c(8,10,12,14)
  , Sex=c("Female","Male")
  , distance = 0
)
newdat$distance <- predict(fml,newdat,re.form=NA)
mm <- model.matrix(terms(fml),newdat)
## or newdat$distance <- mm %*% fixef(fml)
pvar1 <- diag(mm %*% tcrossprod(vcov(fml),mm))
tvar1 <- pvar1+VarCorr(fml)$Subject[1] ## must be adapted for more complex models
cmult <- 2 ## could use 1.96
newdat <- data.frame(
  newdat
  , plo = newdat$distance-cmult*sqrt(pvar1)
  , phi = newdat$distance+cmult*sqrt(pvar1)
  , tlo = newdat$distance-cmult*sqrt(tvar1)
  , thi = newdat$distance+cmult*sqrt(tvar1)
)
#plot confidence
g0 <- ggplot(newdat, aes(x=age, y=distance, colour=Sex))+geom_point()
g0 + geom_pointrange(aes(ymin = plo, ymax = phi))+
  labs(title="CI based on fixed-effects uncertainty ONLY")

```



```
#plot prediction
g0 + geom_pointrange(aes(ymin = tlo, ymax = thi))+
  labs(title="CI based on FE uncertainty + RE variance")
```



```
rm("Orthodont") ## clean up
```

## glmmTMB

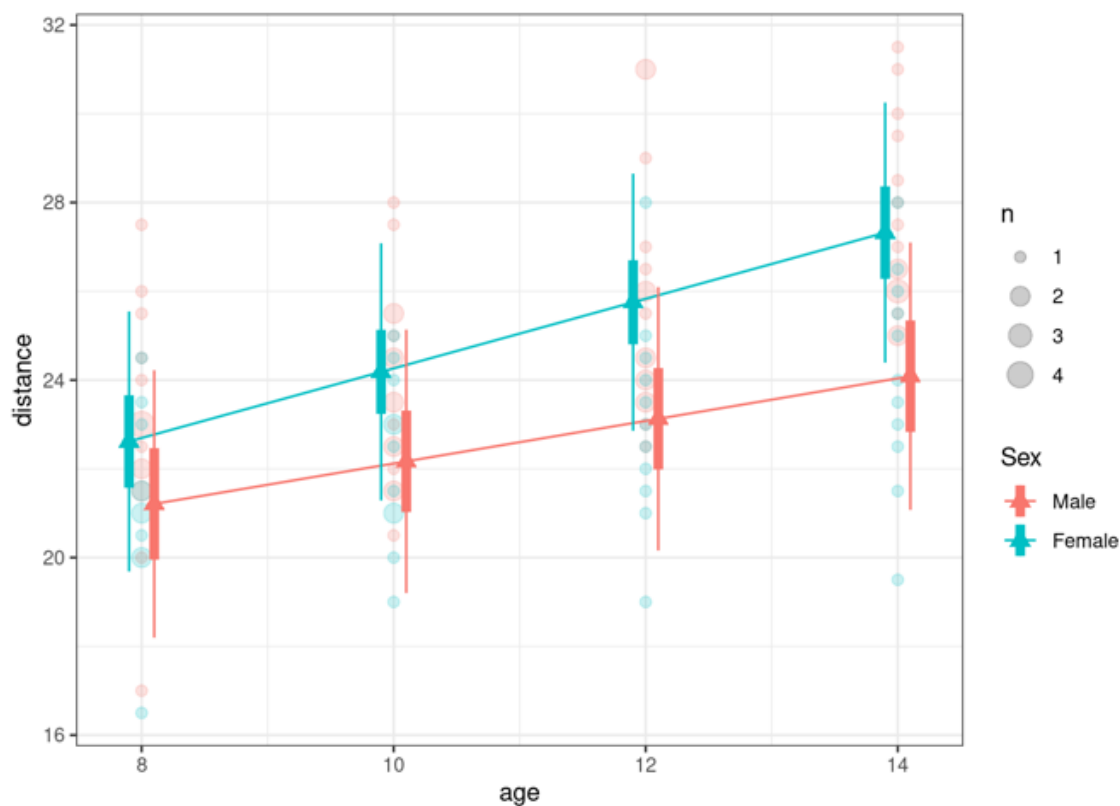
```
library(glmmTMB)
data(Orthodont, package="nlme")
fm2 <- glmmTMB(distance ~ age*Sex + (age | Subject),
  data = Orthodont,
  family="gaussian")
```

```
## Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-definite
## Hessian matrix. See vignette('troubleshooting')
```

```
## make prediction data frame
newdat <- expand.grid(age=c(8,10,12,14), Sex=c("Female","Male"))
## design matrix (fixed effects)
mm <- model.matrix(delete.response(terms(fm2)),newdat)
## linear predictor (for GLMMs, back-transform this with the
## inverse link function (e.g. plogis() for binomial, beta;
## exp() for Poisson, negative binomial)
newdat$distance <- drop(mm %*% fixef(fm2)[["cond"]])
predvar <- diag(mm %*% vcov(fm2)[["cond"]] %*% t(mm))
newdat$SE <- sqrt(predvar)
newdat$SE2 <- sqrt(predvar+sigma(fm2)^2)
```

(Probably overly complicated) ggplot code:

```
library(ggplot2); theme_set(theme_bw())
pd <- position_dodge(width=0.4)
g0 <- ggplot(Orthodont,aes(x=age,y=distance,colour=Sex))+
  stat_sum(alpha=0.2,aes(size=.n..))+
  scale_size_continuous(breaks=1:4,range=c(2,5))
g1 <- g0+geom_line(data=newdat,position=pd)+
  geom_point(data=newdat,shape=17,size=3,position=pd)
## confidence intervals
g2 <- g1 + geom_linerange(data=newdat,
  aes(ymin=distance-2*SE,ymax=distance+2*SE),
  lwd=2, position=pd)
## prediction intervals
g2 + geom_linerange(data=newdat,
  aes(ymin=distance-2*SE2,ymax=distance+2*SE2), position=pd)
```



The `effects`, `emmeans`, and `sjPlot` packages are also useful here.

# Confidence intervals on conditional means/BLUPs/random effects

## lme4

(From Harold Doran, updated by Assaf Oron Nov. 2013:)

If you want the standard errors of the conditional means, you can extract them as follows:

```
library(lme4)
fm1 <- lmer(Reaction ~ Days + (Days|Subject), sleepstudy)
cV <- ranef(fm1, condVar = TRUE)
```

`cV` is a list; each element is a data frame containing the conditional modes for a particular grouping factor. If you use scalar random effects (typically random intercepts), then specifying `ranef(..., drop=TRUE)` will return the conditional modes as a single named vector instead.

The conditional variances are returned as an attribute of the conditional modes. It may be easiest to use `as.data.frame(cV)`, or `broom.mixed::tidy(fm1, effects="ran_vals")`, to extract all of the conditional means and standard deviations.

Or, digging in to the data structure by hand: if we set

```
ranvar <- attr(cV[[1]], "postVar")
```

then `ranvar` is a 3-D array (the attribute is still called `postVar`, rather than `condVar`, for historical reasons/backward compatibility). Individual-level covariance matrices of the conditional modes will sit on the `[, , i]` faces. For example, `ranvar[, , 1]` is the variance-covariance matrix of the conditional distribution for the first group, so

```
sqrt(diag(ranvar[, , 1]))
```

```
## [1] 12.070857 2.304839
```

will provide the intercept and slope standard standard deviations for the first group's conditional modes. If you have a scalar random effect and used `drop=TRUE` in `ranef()`, then you will (mercifully) receive only a vector of individual variances here (one for each level of the grouping factor). The following incantation will give a matrix of conditional variances with one row for each group and one column for each parameters:

```
ng <- dim(ranvar)[3]
np <- dim(ranvar)[2]
mm <- matrix(ranvar[cbind(rep(seq(np), ng),
                        rep(seq(np), ng),
                        rep(ng, each=np))],
            byrow=TRUE,
            nrow=ng)
```

Getting the uncertainty of the coefficients (i.e., what's returned by `coef()`): the sum of the fixed-effect and random-effect predictions for a particular level) is not (alas) currently easy with `lme4`. If the fixed and random effects were independent then we could simply add the conditional variance and the variance of the fixed-effect predictions, but they aren't in general. There is a long [r-sig-mixed-models mailing list thread](https://stat.ethz.ch/pipermail/r-sig-mixed-models/2013q1/019795.html) (<https://stat.ethz.ch/pipermail/r-sig-mixed-models/2013q1/019795.html>) that discusses the issues, focusing on (1) how to extract the covariance between fixed-effect estimate and the random-effect prediction; (2) whether this value (the covariance between an *estimated* parameter and a *predicted* mode of a conditional distribution of a random variable) even makes sense in a frequentist framework. If you're willing to *assume* independence of the conditional variance and the fixed-effect sampling variance, then (e.g.) the variance of the intercepts for each group would be the sum of the fixed-effect intercept variance and the conditional variance of the intercept for each group:

```
vcov(fm1)[1,1]+mm[,1]
```

```
## [1] 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807
## [9] 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807 192.2807
## [17] 192.2807 192.2807
```

## Power analysis

Power analysis for (G)LMMs is mostly done by simulation, although there are some closed-form solutions and approximations, e.g. Snijders and Bosker (1993) (Snijders has links to programs and other resources on his web page (<https://www.stats.ox.ac.uk/~snijders/multilevel.htm>)). There are resources and bits of code examples spread all over the internet. e.g. here (<https://rpubs.com/bbolker/11703>).

Kain, Bolker, and McCoy (2015) and Johnson et al. (2015) are peer-reviewed papers that discuss power analysis via simulation in more detail.

```
library(sos); findFn("{power analysis} mixed simulation")
```

locates the `fullfact`, `pamm`, `simr`, and `simglm` packages. Depending on the goal, one of these packages may have sufficient flexibility to do what you want.

## Model selection and averaging

### Can I use AIC for mixed models? How do I count the number of degrees of freedom for a random effect?

- Yes, with caution.
- It depends on the “level of focus” (*sensu* Spiegelhalter et al. (2002)) whether (e.g.) a single random-effect variance should be counted as 1 degree of freedom (i.e., the variance parameter or as a value between 1 and N-1 (where N is the number of groups): see Vaida and Blanchard (2005) and Greven and Kneib (2010). If you are interested in population-level prediction/inference, then the former (called *marginal AIC* [mAIC]); if individual-level prediction/inference (i.e., using the BLUPs/conditional modes), then the latter (called *conditional AIC* [cAIC]). Greven and Kneib present results for linear models, giving a version of cAIC that is both computationally efficient and takes uncertainty in the estimation of the variances into account. (Bob O’Hara has a very nice, illustrative blog post (<http://deephoughtsandsilliness.blogspot.ca/2007/12/focus-on-dic.html>) on this topic in the context of DIC ...)
- in cases when testing a variance parameter, AIC may be subject to the same kinds of boundary effects as likelihood ratio test p-values (i.e., AICs may be conservative/overfit slightly when the nested parameter value is on the boundary of the feasible space). Greven and Kneib (2010) explore the problems with mAIC in this context, but do not suggest a solution (they point out that Hughes and King (2003) present a ‘one-sided’ AIC, but not one that deals with the non-independence of data points. I haven’t read Hughes and King, I should go do that).
- AIC also inherits the primary problem of likelihood ratio tests in the GLMM context – that is, that LRTs are asymptotic tests. A finite-size correction for AIC does exist (AICc) – but it was developed in the context of linear models. As far as I know its adequacy in the GLMM case has not been established. See e.g. Richards (2005) for caution about AICc in the GLM (not GLMM) case.
- lme4 and nlme count parameters for AIC(c) as follows:
  - the number of fixed-effect parameters is straightforward (the length of the fixed-effect parameter vector beta, i.e. `length(fixef(model))`)
  - each random term in the model with  $q$  components counts for  $q(q + 1)/2$  parameters – for example, a term of the form (slope|group) has 3 parameters (intercept variance, slope variance, correlation between intercept and slope).
  - models that use a scale parameter (e.g. the variance parameter of linear mixed models, or the scale parameter of a Gamma GLMM – standard GLMMs such as binomial and Poisson do not) get an extra parameter counted. Whether to add nuisance parameters or not, such as the residual variance parameter (estimated based on the residual variance, rather than an explicit parameter in the optimization) is as far as I know an open question. In the classic AIC context it doesn’t matter as long as one



is consistent. In the AICc context, I don't think anyone really knows the answer ... adding +1 for the residual variance parameter (as lme4 does) would make the model selection process slightly more conservative.

## Model summaries (goodness-of-fit, decomposition of variance, etc.)

### How do I compute a coefficient of determination ( $R^2$ ), or an analogue, for (G)LMMs?

#### Problem

*(This topic applies to both LMMs and GLMMs, perhaps more so to LMMs, because the issues are even harder for GLMMs.)*

Researchers often want to know if there is a simple (or at least implemented-in-R) way to get an analogue of  $R^2$  or another simple goodness-of-fit metric for LMMs or GLMMs. This is a challenging question in both the GLM and LMM worlds (and therefore doubly so for GLMMs), because it turns out that the wonderful simplicity of  $R^2$  breaks down in the extension to GLMs or LMMs. If you're trying to quantify "fraction of variance explained" in the GLM context, should you include or exclude sampling variation (e.g., Poisson variation around the expected mean)? [According to an `sos::findFn` search for "Nagelkerke", one of the common solutions to this problem, the `LogRegR2` function in the `descr` package computes several different "pseudo- $R^2$ " measures for logistic regression.] If you're trying to quantify it in the LMM context, should you include or exclude variation of different random-effects terms?

The same questions apply more generally to decomposition of variance (i.e. trying to assess the contribution of various model components to the overall fit, not just trying to assess the overall goodness-of-fit of the model); there is unlikely to be a single recipe that does everything you want.

This has been discussed at various times on the mailing lists. This thread (<http://thread.gmane.org/gmane.comp.lang.r.lme4.devel/3281>) and this thread (<http://thread.gmane.org/gmane.comp.lang.r.lme4.devel/684>) on the r-sig-mixed-models mailing list are good starting points, and this post (<http://thread.gmane.org/gmane.comp.lang.r.lme4.devel/2143>) is useful too.

In one of those threads, Doug Bates said:

Assuming that one wants to define an  $R^2$  measure, I think an argument could be made for treating the penalized residual sum of squares from a linear mixed model in the same way that we consider the residual sum of squares from a linear model. Or one could use just the residual sum of squares without the penalty or the minimum residual sum of squares obtainable from a given set of terms, which corresponds to an infinite precision matrix. I don't know, really. It depends on what you are trying to characterize.

#### Simple/crude solutions

In one of those threads, Jarrett Byrnes contributed the following code:

```
r2.corr.mer <- function(m) {
  lmfit <- lm(model.response(model.frame(m)) ~ fitted(m))
  summary(lmfit)$r.squared
}
```

$\Omega_0^2$  (Xu 2003), which is almost the same, is based on comparing the residual variance of the full model against the residual variance of a (fixed) intercept-only null model:

```
1-var(residuals(m))/var(model.response(model.frame(m)))
```

Another possibility is the squared correlation between the response variable and the predicted values:

```
cor(model.response(model.frame(m)), predict(m, type="response"))^2
```

## Sophisticated solutions

Gelman and Pardoe (2006) propose/discuss Bayesian measures of  $R^2$  (I don't know if anyone has created a canned implementation of these measures in R). Nakagawa and Schielzeth (2013) and Johnson (2014) have also proposed a general methodology for computing  $R^2$ ; J. Lefcheck gives examples here (<https://jonlecheck.net/2013/03/13/r2-for-linear-mixed-effects-models/>) and here (<https://github.com/jslefcche/piecewiseSEM/blob/master/README.md#get-r2-for-individual-models>), based on his implementation in the `piecewiseSEM` package (CRAN (<https://cran.r-project.org/web/packages/piecewiseSEM/index.html>), Github (<https://github.com/jslefcche/piecewiseSEM>)). See also Jaeger et al. (2017), Rights and Sterba (2018) ...

A related question is how to quantify “repeatability” (i.e., ratios of variance at different levels) in GLMMs, especially how to compute the “residual error” term for GLMMs: see Nakagawa and Schielzeth (2010) and the `rptR` package (<http://rptr.r-forge.r-project.org/>).

The bottom line is that there are some simple recipes (and some more complex recipes that may or may not have been coded up by someone), but that “you have to think carefully about what information you want to get out of the coefficient of determination”, because no recipe will have all of the properties of  $R^2$  in the simple linear model case.

**Packages/functions:** See `performance::r2()` (<https://easystats.github.io/performance/reference/r2.html>), `MuMIn::r.squaredGLMM()`, the `r2glmm` package (<https://CRAN.R-project.org/package=r2glmm>), the standalone `r2MLM` function (<https://my.vanderbilt.edu/jasonrights/software/r2MLM/>), stuff in the `piecewiseSEM` package, `psycho::R2_nakagawa` ([http://finzi.psych.upenn.edu/R/library/psycho/html/R2\\_nakagawa.html](http://finzi.psych.upenn.edu/R/library/psycho/html/R2_nakagawa.html)), `partR2` (<https://github.com/mastoffel/partR2>) package ... (try e.g. `sos::findFn("Nakagawa Schielzeth")` for an up-to-date list ...)

## Variable importance

- The simplest way to get (within-study) measures of variable importance is to standardize the predictor variables (scaling by 1 SD or 2SD: Gelman (2008), Schielzeth (2010))
- The `r2glmm` package computes partial  $R^2$  values for fixed effects (only for `lmer`, `lme`, and `glmmPQL` models)
- Henrik Singmann has a detailed answer here (<https://afex.singmann.science/forums/topic/compute-effect-sizes-for-mixed-objects>) on why standardized measures such as partial eta-squared are problematic:

The fact that calculating a global measure of model fit (such as  $R^2$ ) is already riddled with complications and that no simple single number can be found, should be a hint that doing so for a subset of the model parameters (i.e., main-effects or interactions) is even more difficult. Given this, I would not recommend to try finding a measure of standardized effect sizes for mixed models.

He even gives suggested wording for responding to reviewers who want standardized measures!

## Do I have to specify the levels of fixed effects in lmer?

No. See Doug Bates reply to this question here (<http://r.789695.n4.nabble.com/lme4-and-Variable-level-detection-td881680.html>)

## Miscellaneous/procedural

## Pronunciation of lmer / glmer / etc.

- `lmer` : I have heard “ell emm ee arr” (i.e. pronouncing each letter); “elmer” (probably most common); and “lemur”
- `glmer` : “gee ell emm ee arr”, “gee elmer”, “glimmer”, or “gleamer”
- for `lme` and `nlme` people just seem to spell out the names (rather than saying e.g. “lemmy” and “nelmy”)

## Storing information

Recent versions of `lme4` output contain an `@optinfo` slot that stores warnings.

Copied from <https://stat.ethz.ch/pipermail/r-help/2012-February/302767.html> (<https://stat.ethz.ch/pipermail/r-help/2012-February/302767.html>) :

There’s a somewhat hack-ish solution, which is to use `options(warn=2)` to ‘upgrade’ warnings to errors, and then use `try()` or `tryCatch()` to catch them.

More fancily, I used code that looked something like this to save warnings as I went along (sorry about the `<-` ) in a recent simulation study. You could also check `w$message` to do different things in the case of different warnings.

```
## n.b. have to set up a 3D warn array first ...
withCallingHandlers(tryCatch(fun(n=nvec[j],tau=tauvec[i],...),
  error = function(e) {
    warn[k,i,j] <- paste("ERROR:",e$message)
    NA_ans}),
  warning = function(w) {
    warn[k,i,j] <- w$message
    invokeRestart("muffleWarning")
  })
```

## Mixed modeling packages

- also see the package comparison (<http://glmm.wikidot.com/pkg-comparison>) on [glmm.wikidot.com](http://glmm.wikidot.com)

## Which R packages (functions) fit GLMMs?

- `MASS::glmmPQL` (penalized quasi-likelihood)
- `lme4::glmer` (Laplace approximation and adaptive Gauss-Hermite quadrature [AGHQ])
- `MCMCglmm` (Markov chain Monte Carlo)
- `glmmML` (AGHQ)
- `glmmAK` (AGHQ?)
- `glmmADMB` (Laplace)
- `glmm` (from Jim Lindsey’s `repeated` package: AGHQ)
- `gamlss.mx`
- `ASREML-R`
- `sabreR`

## Should I use `aov()`, `nlme`, or `lme4`, or some other package?

- `aov()` (in the `stats` package in base R: balanced, orthogonal designs only (R analogue of SAS PROC GLM))

- `nlme` (analogue of SAS `PROC MIXED / NLMIXED`)
  - allows more complex designs than `aov` (unbalanced, heteroscedasticity and/or correlation among residual errors)
  - more mature than `lme4`
  - well-documented (Pinheiro and Bates 2000)
  - implements R-side effects (heteroscedasticity and correlation)
  - estimates “denominator degrees of freedom” for  $F$  statistics, and hence  $p$  values, for LMMs (but see above)
- `lme4` (also SAS `PROC MIXED / NLMIXED`):
  - fastest
  - best for crossed designs (although they are possible in `nlme`)
  - GLMMs
  - cutting-edge (for better or worse!)
  - likelihood profiles
  - use `lme4` for GLMMs, or if you have big data (thousands to tens of thousands of records)

The following is modified from a contribution by Kingsford Jones, found 2010-03-16

## linear and nonlinear mixed models

- `lme` (<http://cran.r-project.org/web/packages/lme4/index.html>) – Linear mixed-effects models using S4 classes
- `lmm` – Linear mixed models
- `nlme` – Linear and Nonlinear Mixed Effects Models
- `sabreR`
- `regress` (<http://cran.r-project.org/web/packages/regress/index.html>) Linear mixed models

## GLMMs

- `glmmAK` – Generalized Linear Mixed Models
- `MASS` – Main Package of Venables and Ripley’s `MASS` (see function `glmmPQL`)
- `MCMCglmm` – MCMC Generalised Linear Mixed Models
- `lme4` (`glmer`)
- `glmmML`
- `gamlss.mx`
- `sabreR`

## Additive and generalized-additive mixed models

- `amer` – Additive mixed models with `lme4`
- `gamm4` – Generalized additive mixed models using `mgcv` and `lme4`
- `mgcv` (`gamm` function, via `glmmPQL` in `MASS` package)
- `gamlss.mx`

## Hierarchical GLMs

- `hglm` – `hglm` is used to fit hierarchical generalized linear models
- `HGLMMM` – Hierarchical Generalized Linear Models

## diagnostic and modeling frameworks

- `influence.ME` – Tools for detecting influential data in mixed effects models
- `arm` – Data Analysis Using Regression and Multilevel/Hierarchical Models
- `pamm` – Power analysis for random effects in mixed models
- `RLRsim` – Exact (Restricted) Likelihood Ratio tests for mixed and additive models
- `npde` – Normalised prediction distribution errors for nonlinear mixed-effect models
- `multilevel` – Multilevel Functions (psychology-oriented; within-group agreement, random group resampling, etc.)
- `languageR`

- `pbkrtest` – parametric bootstrap and Kenward-Roger tests

## data and examples

- `MEMSS` – Data sets from Mixed-effects Models in S
- `mlmRev` – Examples from Multilevel Modelling Software Review
- `SASmixed` – Data sets from “SAS System for Mixed Models”

## extensions

- `lmeSplines` – `lmeSplines`
- `lmeC` – Linear Mixed-Effects Models with Censored Responses
- `kinship` – mixed-effects Cox models, sparse matrices, and modeling data from large pedigrees
- `coxme` – Mixed Effects Cox Models
- `ordinal` – Regression Models for Ordinal Data
- `phmm` – Proportional Hazards Mixed-effects Model (PHMM)
- `pedigreemm` – Pedigree-based mixed-effects models
- (see also `MCMCglmm` for pedigree-based approaches)
- `heavy` – Estimation in the linear mixed model using heavy-tailed distributions
- `GLMMar` – Generalized Linear Multilevel Model with AR(p) Errors Package
- `glmmlasso` – penalized GLMM fitting
- `spatialCovariance` – spatial covariance matrix calculations

## Interfaces to other systems

- `glmmBUGS` – Generalised Linear Mixed Models and Spatial Models with BUGS
- Interfaces to WinBUGS/OpenBUGS/JAGS (roll your own model file):
- `R2WinBUGS`
- `r2jags`
- `rjags`
- `RBugs`

## modeling based on LMMs

- `nlmeODE` – Non-linear mixed-effects modelling in `nlme` using differential equations
- `longRPart` – Recursive partitioning of longitudinal data using mixed-effects models
- `PSM` – Non-Linear Mixed-Effects modelling using Stochastic Differential Equations

## Off-CRAN mixed modeling packages:

### R-forge and Github:

- `glmmADMB` (R-forge, interface to AD Model Builder)
- `spida`, `p3d` (Georges Monette)

### Other open source:

- `bernor` (<http://www.stat.umn.edu/geyer/bernor/%20bernor>) package (logit-normal fitting), by Yun Ju Sung and Charles J. Geyer
- `glmm` (in Jim Lindsey’s `repeated` package: at Lindsey’s web site (<http://www.commanster.eu/rcode.html>))

### Commercial:

- `OpenMx` – Advanced Structural Equation Modeling
- `ASReml-R` (commercial, but 30 days’ free use/free license for academic or developing-country use available). Very good at complex

LMMs (fast, flexible covariance structures, etc.), but only offers PQL for GLMMs, and the manual says: > we cannot recommend the use of this technique for general use. It is included in the current version of asreml() for advanced users. It is highly recommended that its use be accompanied by some form of cross-validators assessment for the specific dataset concerned." Resources:

- short R wiki tutorial (<http://rwiki.sciviews.org/doku.php?id=guides:tutorials:asreml>)
- reference manual (<http://www.vsnr.co.uk/downloads/asreml/release3/asreml-R.pdf%20reference%20manual>) (PDF)
- Luis Apolaza's asreml-r cookbook (<http://apolaza.net/asreml-r/>)

## Package versions used

```
sessionInfo()
```

```
## R Under development (unstable) (2022-02-25 r81818)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Pop!_OS 21.04
##
## Matrix products: default
## BLAS: /usr/local/lib/R/lib/libRblas.so
## LAPACK: /usr/local/lib/R/lib/libRlapack.so
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8 LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8 LC_NAME=C
## [9] LC_ADDRESS=C LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] ggplot2_3.3.5 RLRsim_3.1-6 nlme_3.1-155 dplyr_1.0.8
## [5] broom.mixed_0.2.8 glmmTMB_1.1.3 equatiomatic_0.3.1 lme4_1.1-28.9000
## [9] Matrix_1.4-0 Cairo_1.5-14 pandor_0.6.4 knitr_1.37
##
## loaded via a namespace (and not attached):
## [1] sass_0.4.0 tidyr_1.2.0 jsonlite_1.8.0
## [4] splines_4.2.0 bslib_0.3.1 shiny_1.7.1
## [7] assertthat_0.2.1 highr_0.9 yaml_2.3.5
## [10] numDeriv_2016.8-1.1 pillar_1.7.0 backports_1.4.1
## [13] lattice_0.20-45 glue_1.6.2 digest_0.6.29
## [16] promises_1.2.0.1 minqa_1.2.4 colorspace_2.0-3
## [19] sandwich_3.0-1 htmltools_0.5.2 httpuv_1.6.5
## [22] pkgconfig_2.0.3 broom_0.7.12 purrr_0.3.4
## [25] xtable_1.8-4 mvtnorm_1.1-3 scales_1.1.1
## [28] later_1.3.0 emmeans_1.7.2 tibble_3.1.6
## [31] mgcv_1.8-38 farver_2.1.0 generics_0.1.2
## [34] ellipsis_0.3.2 TH.data_1.1-0 withr_2.4.3
## [37] TMB_1.7.22 cli_3.2.0 survival_3.2-13
## [40] magrittr_2.0.2 crayon_1.5.0 mime_0.12
## [43] estimability_1.3 evaluate_0.15 fansi_1.0.2
## [46] MASS_7.3-55 tools_4.2.0 lifecycle_1.0.1
## [49] multcomp_1.4-18 stringr_1.4.0 munsell_0.5.0
## [52] compiler_4.2.0 jquerylib_0.1.4 rlang_1.0.1
## [55] grid_4.2.0 nloptr_2.0.0 labeling_0.4.2
## [58] rmarkdown_2.11.15 boot_1.3-28 gtable_0.3.0
## [61] codetools_0.2-18 DBI_1.1.2 R6_2.5.1
## [64] zoo_1.8-9 fastmap_1.1.0 utf8_1.2.2
## [67] stringi_1.7.6 Rcpp_1.0.8 vctrs_0.3.8
## [70] tidyselect_1.1.2 xfun_0.29 coda_0.19-4
```

## To do

- add links to `merDeriv` for standard devs of variances, robust estimates. More on Rizopoulos package
- update package descriptions; cross-link with Task View (<http://bbolker.github.io/mixedmodels-misc/MixedModels.html>) ?  
rethinking, brms, ...
- more on post-analysis (`broom(.mixed)`, `emmeans`, `multcomp`, ...)
- more on confidence intervals, simulating from conditional distributions, etc.)

# Bibliography

- Agresti, Alan. 2002. *Categorical Data Analysis*. 2d ed. Hoboken, NJ: Wiley.
- Angrist, Joshua D., and Jörn-Steffen Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. 1 edition. Princeton: Princeton University Press.
- Barr, Dale J. 2020. *Learning Statistical Models Through Simulation in R*. PsyTeachR Books. <https://psyteachr.github.io/ug3-stats/> (<https://psyteachr.github.io/ug3-stats/>).
- Barr, Dale J., Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. "Random Effects Structure for Confirmatory Hypothesis Testing: Keep It Maximal." *Journal of Memory and Language* 68 (3): 255–78. <https://doi.org/10.1016/j.jml.2012.11.001> (<https://doi.org/10.1016/j.jml.2012.11.001>).
- Bates, Douglas, Reinhold Kliegl, Shravan Vasishth, and Harald Baayen. 2015. "Parsimonious Mixed Models." *arXiv:1506.04967 [Stat]*, June. <http://arxiv.org/abs/1506.04967> (<http://arxiv.org/abs/1506.04967>).
- Bates, Douglas, Martin Mächler, Benjamin M. Bolker, and Steven C. Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. <https://doi.org/10.18637/jss.v067.i01> (<https://doi.org/10.18637/jss.v067.i01>).
- Bell, Melanie L., and Gary K. Grunwald. 2010. "Small Sample Estimation Properties of Longitudinal Count Models." *Journal of Statistical Computation and Simulation* 81 (9): 1067–79. <https://doi.org/10.1080/00949651003674144> (<https://doi.org/10.1080/00949651003674144>).
- Berger, J. O., B. Liseo, and R. L. Wolpert. 1999. "Integrated Likelihood Methods for Eliminating Nuisance Parameters." *Statistical Science* 14 (1): 1–22. [http://projecteuclid.org/download/pdf\\_1/euclid.ss/1009211804](http://projecteuclid.org/download/pdf_1/euclid.ss/1009211804) ([http://projecteuclid.org/download/pdf\\_1/euclid.ss/1009211804](http://projecteuclid.org/download/pdf_1/euclid.ss/1009211804)).
- Breslow, N. E. 1984. "Extra-Poisson Variation in Log-Linear Models." *Journal of the Royal Statistical Society C* 33: 38–44. <http://www.jstor.org/stable/234766> (<http://www.jstor.org/stable/234766>).
- Browne, W. J, S. V. Subramanian, K. Jones, and H. Goldstein. 2005. "Variance Partitioning in Multilevel Logistic Models That Exhibit Overdispersion." *Journal of the Royal Statistical Society A* 168 (3): 599–613. <https://doi.org/10.1111/j.1467-985X.2004.00365.x> (<https://doi.org/10.1111/j.1467-985X.2004.00365.x>).
- Chung, Yeojin, Sophia Rabe-Hesketh, Vincent Dorie, Andrew Gelman, and Jingchen Liu. 2013. "A Nondegenerate Penalized Likelihood Estimator for Variance Parameters in Multilevel Models." *Psychometrika*, 1–25. <https://doi.org/10.1007/s11336-013-9328-2> (<https://doi.org/10.1007/s11336-013-9328-2>).
- Clark, Tom S, and Drew A Linzer. 2015. "Should I Use Fixed or Random Effects?" *Political Science Research and Methods* 3 (02): 399–408.
- Cordeiro, Gauss M., and Silvia L. P. Ferrari. 1998. "A Note on Bartlett-Type Correction for the First Few Moments of Test Statistics." *Journal of Statistical Planning and Inference* 71 (1-2): 261–69. [https://doi.org/10.1016/S0378-3758\(98\)00005-6](https://doi.org/10.1016/S0378-3758(98)00005-6) ([https://doi.org/10.1016/S0378-3758\(98\)00005-6](https://doi.org/10.1016/S0378-3758(98)00005-6)).
- Cordeiro, Gauss M., Gilberto A. Paula, and Denise A. Botter. 1994. "Improved Likelihood Ratio Tests for Dispersion Models." *International Statistical Review / Revue Internationale de Statistique* 62 (2): 257–74. <https://doi.org/10.2307/1403512> (<https://doi.org/10.2307/1403512>).
- Crawley, Michael J. 2002. *Statistical Computing: An Introduction to Data Analysis Using S-PLUS*. John Wiley & Sons.
- Cribari-Neto, Francisco, and Achim Zeileis. 2009. "Beta Regression in R." 98. Vienna, Austria: WU Vienna University of Economics; Business. <https://cran.r-project.org/web/packages/betareg/index.html> (<https://cran.r-project.org/web/packages/betareg/index.html>).
- Dobson, Annette J., and Adrian Barnett. 2008. *An Introduction to Generalized Linear Models, Third Edition*. 3rd ed. Chapman; Hall/CRC.
- Elston, D. A., R. Moss, T. Boulinier, C. Arrowsmith, and X. Lambin. 2001. "Analysis of Aggregation, a Worked Example: Numbers of Ticks on Red Grouse Chicks." *Parasitology* 122 (5): 563–69.
- Faraway, Julian J. 2006. *Extending Linear Models with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman & Hall/CRC.



- Feng, Ziding, Thomas Braun, and Charles McCulloch. 2004. "Small Sample Inference for Clustered Data." In *Proceedings of the Second Seattle Symposium in Biostatistics*, edited by D. Y. Lin and P. J. Heagerty, 179:71–87. New York, NY: Springer.  
<http://www.springerlink.com/content/h2g33m7127790343/> (<http://www.springerlink.com/content/h2g33m7127790343/>).
- Gelman, Andrew. 2005. "Analysis of Variance: Why It Is More Important Than Ever." *Annals of Statistics* 33 (1): 1–53.  
<https://doi.org/doi:10.1214/009053604000001048> (<https://doi.org/doi:10.1214/009053604000001048>).
- . 2006. "Prior Distributions for Variance Parameters in Hierarchical Models." *Bayesian Analysis* 1 (3): 515–33.  
<http://ba.stat.cmu.edu/journal/2006/vol01/issue03/gelman.pdf> (<http://ba.stat.cmu.edu/journal/2006/vol01/issue03/gelman.pdf>).
- . 2008. "Scaling Regression Inputs by Dividing by Two Standard Deviations." *Statistics in Medicine* 27 (15): 2865–73.  
<https://doi.org/10.1002/sim.3107> (<https://doi.org/10.1002/sim.3107>).
- Gelman, Andrew, and Jennifer Hill. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge, England: Cambridge University Press. <http://www.stat.columbia.edu/~gelman/arm/> (<http://www.stat.columbia.edu/~gelman/arm/>).
- Gelman, Andrew, and Iain Pardoe. 2006. "Bayesian Measures of Explained Variance and Pooling in Multilevel (Hierarchical) Models." *Technometrics* 48 (2): 241–51. <http://amstat.tandfonline.com/doi/abs/10.1198/004017005000000517>  
<http://amstat.tandfonline.com/doi/abs/10.1198/004017005000000517>.
- Goldman, Nick, and Simon Whelan. 2000. "Statistical Tests of Gamma-Distributed Rate Heterogeneity in Models of Sequence Evolution in Phylogenetics." *Molecular Biology and Evolution* 17 (6): 975–78.
- Gotelli, Nicholas J., and Aaron M. Ellison. 2004. *A Primer of Ecological Statistics*. Sunderland, MA: Sinauer.
- Greven, Sonja, and Thomas Kneib. 2010. "On the Behaviour of Marginal and Conditional Akaike Information Criteria in Linear Mixed Models." *Biometrika* 97 (4): 773–89. <http://www.bepress.com/jhubiostat/paper202/> (<http://www.bepress.com/jhubiostat/paper202/>).
- Harrison, Xavier A. 2015. "A Comparison of Observation-Level Random Effect and Beta-Binomial Models for Modelling Overdispersion in Binomial Data in Ecology and Evolution." *PeerJ* 3 (July): e1114. <https://doi.org/10.7717/peerj.1114> (<https://doi.org/10.7717/peerj.1114>).
- Heisterkamp, Simon H., Engelbertus van Willigen, Paul-Matthias Diderichsen, and John Maringwa. 2017. "Update of the Nlme Package to Allow a Fixed Standard Deviation of the Residual Error." *The R Journal* 9 (1): 239–51.
- Hinde, John. 1982. "Compound Poisson Regression Models." In *GLIM82: Proc. Int. Conf. On Glms*, edited by R. Gilchrist, 109–21. Springer.
- Hodges, James S. 2016. *Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects*. Chapman; Hall/CRC.
- Hughes, A., and M. King. 2003. "Model Selection Using AIC in the Presence of One-Sided Information." *Journal of Statistical Planning and Inference* 115: 497–11.
- Hurlbert, S. 1984. "Pseudoreplication and the Design of Ecological Field Experiments." *Ecological Monographs* 54: 187–211.
- Jaeger, Byron C., Lloyd J. Edwards, Kalyan Das, and Pranab K. Sen. 2017. "An R2 Statistic for Fixed Effects in the Generalized Linear Mixed Model." *Journal of Applied Statistics* 44 (6): 1086–1105. <https://doi.org/10.1080/02664763.2016.1193725>  
<https://doi.org/10.1080/02664763.2016.1193725>.
- Johnson, Paul C. D. 2014. "Extension of Nakagawa & Schielzeth's R2GLMM to Random Slopes Models." *Methods in Ecology and Evolution* 5 (9): 944–46. <https://doi.org/10.1111/2041-210X.12225> (<https://doi.org/10.1111/2041-210X.12225>).
- Johnson, Paul C. D., Sarah J. E. Barry, Heather M. Ferguson, and Pie Müller. 2015. "Power Analysis for Generalized Linear Mixed Models in Ecology and Evolution." *Methods in Ecology and Evolution* 6 (2): 133–42. <https://doi.org/10.1111/2041-210X.12306>  
<https://doi.org/10.1111/2041-210X.12306>.
- Kain, Morgan P., Ben M. Bolker, and Michael W. McCoy. 2015. "A Practical Guide and Power Analysis for GLMMs: Detecting Among Treatment Variation in Random Effects." *PeerJ* 3 (September): e1226. <https://doi.org/10.7717/peerj.1226>  
<https://doi.org/10.7717/peerj.1226>.
- Lawson, A., A. Biggeri, D. Bohning, E. LeSaffre, J. F. Viel, and R. Bertollini, eds. 1999. *Disease Mapping and Risk Assessment for Public Health*. New York: Wiley.

- Lee, Youngjo, John A. Nelder, and Yudi Pawitan. 2017. *Generalized Linear Models with Random Effects: Unified Analysis via H-Likelihood, Second Edition*. 2 edition. Boca Raton, Florida: Chapman; Hall/CRC.
- Littell, Ramon C., George A. Milliken, Walter W. Stroup, Russell D. Wolfinger, and Oliver Schabenberger. 2006. *SAS for Mixed Models, Second Edition*. SAS Publishing.
- Lo, Steson, and Sally Andrews. 2015. "To Transform or Not to Transform: Using Generalized Linear Mixed Models to Analyse Reaction Time Data." *Frontiers in Psychology* 6. <https://doi.org/10.3389/fpsyg.2015.01171> (<https://doi.org/10.3389/fpsyg.2015.01171>).
- Maindonald, J., and J. Braun. 2010. *Data Analysis and Graphics Using R, an Example-Based Approach*. 3rd ed. Cambridge University Press.
- Martin, Tara G., Brendan A. Wintle, Jonathan R. Rhodes, Petra M. Kuhnert, Scott A. Field, Samantha J. Low-Choy, Andrew J. Tyre, and Hugh P. Possingham. 2005. "Zero Tolerance Ecology: Improving Ecological Inference by Modelling the Source of Zero Observations: Modelling Excess Zeros in Ecology." *Ecology Letters* 8 (11): 1235–46. <https://doi.org/10.1111/j.1461-0248.2005.00826.x> (<https://doi.org/10.1111/j.1461-0248.2005.00826.x>).
- Matuschek, Hannes, Reinhold Kliegl, Shravan Vasishth, Harald Baayen, and Douglas Bates. 2017. "Balancing Type I Error and Power in Linear Mixed Models." *Journal of Memory and Language* 94: 305–15. <https://doi.org/10.1016/j.jml.2017.01.001> (<https://doi.org/10.1016/j.jml.2017.01.001>).
- McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. London: Chapman; Hall.
- McCulloch, Charles E., and John M. Neuhaus. 2011. "Misspecifying the Shape of a Random Effects Distribution: Why Getting It Wrong May Not Matter." *Statistical Science* 26 (3): 388–402. <https://doi.org/10.1214/11-STS361> (<https://doi.org/10.1214/11-STS361>).
- Millar, Russell B. 2011. *Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB*. John Wiley & Sons.
- Nakagawa, Shinichi, and Holger Schielzeth. 2010. "Repeatability for Gaussian and Non-Gaussian Data: A Practical Guide for Biologists." *Biological Reviews* 85 (4): 935–56. <https://doi.org/10.1111/j.1469-185X.2010.00141.x> (<https://doi.org/10.1111/j.1469-185X.2010.00141.x>).
- . 2013. "A General and Simple Method for Obtaining R<sup>2</sup> from Generalized Linear Mixed-Effects Models." *Methods in Ecology and Evolution* 4 (2): 133–42. <https://doi.org/10.1111/j.2041-210x.2012.00261.x> (<https://doi.org/10.1111/j.2041-210x.2012.00261.x>).
- Oberpriller, Johannes, Melina de Souza Leite, and Maximilian Pichler. 2021. "Fixed or Random? On the Reliability of Mixed-Effects Models for a Small Number of Levels in Grouping Variables." *bioRxiv*, June, 2021.05.03.442487. <https://doi.org/10.1101/2021.05.03.442487> (<https://doi.org/10.1101/2021.05.03.442487>).
- Pasch, Bret, Benjamin M. Bolker, and Steven M. Phelps. 2013. "Interspecific Dominance via Vocal Interactions Mediates Altitudinal Zonation in Neotropical Singing Mice." *The American Naturalist* 182 (5): E161–E173. <https://doi.org/10.1086/673263> (<https://doi.org/10.1086/673263>).
- Pinheiro, José C., and Douglas M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.
- Rabe-Hesketh, Sophia, and Anders Skrondal. 2008. *Multilevel and Longitudinal Modeling Using Stata*. 2nd ed. Stata Press. <http://www.stata-press.com/books/mlmus2.html> (<http://www.stata-press.com/books/mlmus2.html>).
- Richards, Shane A. 2005. "Testing Ecological Theory Using the Information-Theoretic Approach: Examples and Cautionary Results." *Ecology* 86 (10): 2805–14. <https://doi.org/10.1890/05-0074> (<https://doi.org/10.1890/05-0074>).
- Rights, Jason D., and Sonya K. Sterba. 2018. "Quantifying Explained Variance in Multilevel Models: An Integrative Framework for Defining R-Squared Measures." *Psychological Methods*. <https://doi.org/10.1037%2Fmet0000184> (<https://doi.org/10.1037%2Fmet0000184>).
- Schaalje, G., J. McBride, and G. Fellingham. 2002. "Adequacy of Approximations to Distributions of Test Statistics in Complex Mixed Linear Models." *Journal of Agricultural, Biological & Environmental Statistics* 7 (14): 512–24. <http://www.ingentaconnect.com/content/asa/jabes/2002/00000007/00000004/art00004> (<http://www.ingentaconnect.com/content/asa/jabes/2002/00000007/00000004/art00004>).
- Schabenberger, Oliver, and Francis J. Pierce. 2001. *Contemporary Statistical Models for the Plant and Soil Sciences*. Boca Raton, FL: CRC Press.

- Schielzeth, Holger. 2010. "Simple Means to Improve the Interpretability of Regression Coefficients." *Methods in Ecology and Evolution* 1: 103–13. <https://doi.org/10.1111/j.2041-210X.2010.00012.x> (<https://doi.org/10.1111/j.2041-210X.2010.00012.x>).
- Self, Steven G., and Kung-Yee Liang. 1987. "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions." *Journal of the American Statistical Association* 82 (398): 605–10. <https://doi.org/10.1080/01621459.1987.10478472> (<https://doi.org/10.1080/01621459.1987.10478472>).
- Smithson, Michael, and Jay Verkuilen. 2006. "A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables." *Psychological Methods* 11 (1): 54–71. <https://doi.org/2006-03820-004> (<https://doi.org/2006-03820-004>).
- Snijders, Tom A. B., and Roel J. Bosker. 1993. "Standard Errors and Sample Sizes for Two-Level Research." *Journal of Educational Statistics* 18 (3): 237. <https://doi.org/10.2307/1165134> (<https://doi.org/10.2307/1165134>).
- Spiegelhalter, D. J., N. Best, B. P. Carlin, and A. Van der Linde. 2002. "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society B* 64: 583–640.
- Stram, Daniel O, and Jae Won Lee. 1994. "Variance Components Testing in the Longitudinal Fixed Effects Model." *Biometrics* 50 (4): 1171–7. <http://links.jstor.org/sici?sici=0006-341X%28199412%2950%3A4%3C1171%3AVCTITL%3E2.0.CO%3B2-H> (<http://links.jstor.org/sici?sici=0006-341X%28199412%2950%3A4%3C1171%3AVCTITL%3E2.0.CO%3B2-H>).
- Stroup, Walter W. 2014. "Rethinking the Analysis of Non-Normal Data in Plant and Soil Science." *Agronomy Journal* 106: 1–17. <https://doi.org/10.2134/agronj2013.0342> (<https://doi.org/10.2134/agronj2013.0342>).
- Stroup, W. W. 2013. "Non-Normal Data in Agricultural Experiments." In. Kansas State University. <http://newprairiepress.org/agstatconference/2013/proceedings/8> (<http://newprairiepress.org/agstatconference/2013/proceedings/8>).
- Vaida, Florin, and Suzette Blanchard. 2005. "Conditional Akaike Information for Mixed-Effects Models." *Biometrika* 92 (2): 351–70. <https://doi.org/10.1093/biomet/92.2.351> (<https://doi.org/10.1093/biomet/92.2.351>).
- Venables, W., and Brian D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. New York: Springer.
- Warton, David I. 2005. "Many Zeros Does Not Mean Zero Inflation: Comparing the Goodness-of-Fit of Parametric Models to Multivariate Abundance Data." *Environmetrics* 16 (3): 275–89. <https://doi.org/10.1002/env.702> (<https://doi.org/10.1002/env.702>).
- Wolfinger, Russ, and Michael O'Connell. 1993. "Generalized Linear Mixed Models a Pseudo-Likelihood Approach." *Journal of Statistical Computation and Simulation* 48 (3-4): 233–43. <https://doi.org/10.1080/00949659308811554> (<https://doi.org/10.1080/00949659308811554>).
- Xu, R. 2003. "Measuring Explained Variation in Linear Mixed Effects Models." *Statist. Med.* 22: 3527–41. <https://doi.org/10.1002/sim.1572> doi:10.1002/sim.1572 (<https://doi.org/10.1002/sim.1572> doi:10.1002/sim.1572).
- Zuur, Alain F., Elena N. Ieno, Neil J. Walker, Anatoly A. Saveliev, and Graham M. Smith. 2009a. "Zero-Truncated and Zero-Inflated Models for Count Data." In *Mixed Effects Models and Extensions in Ecology with R*, 261–93. New York, NY: Springer New York. <http://www.springerlink.com/content/m087275807178771/> (<http://www.springerlink.com/content/m087275807178771/>).
- — —. 2009b. *Mixed Effects Models and Extensions in Ecology with R*. Springer.

1. in R, `foo::bar` (or `foo::bar()`) denotes "function `bar` in package `foo`".↩

2. the dispersion factor is estimated on a variance, so we need to take the square root to apply it to the standard error↩