

D. A. S. Fraser, N. Reid, and J. Wu. A simple general formula for tail probabilities for frequentist and Bayesian inference. *Biometrika*, 86:249–264, 1999. [26](#)

D. A. Pierce and D. Peters. Practical use of higher-order asymptotics for multiparameter exponential families. *Journal of the Royal Statistical Society Series B*, 54:701–737, 1992. [21](#)

T. A. Severini. *Likelihood Methods in Statistics*. Oxford University Press, Oxford, 2000. [20](#), [21](#), [26](#)

I. M. Skovgaard. An explicit large-deviation approximation to one-parameter tests. *Bernoulli*, 2:145–165, 1996. [26](#)

I. M. Skovgaard. Likelihood asymptotics. *Scandinavian Journal of Statistics*, 28:3–32, 2001. [21](#)

Alessandra R. Brazzale

Institute of Biomedical Engineering, Italian National Research Council

[alessandra.brazzale@isib.cnr.it](mailto:alessandra.brazzale@isib.cnr.it)

## Fitting linear mixed models in R

### Using the `lme4` package

by Douglas Bates

The `lme` function, which fits linear mixed models of the form described in [Pinheiro and Bates \(2000\)](#), has been available in the required R package `nlme` for several years. Recently my colleagues and I have been developing another R package, called `lme4`, and its `lmer` function which provides more flexible fitting of linear mixed models and also provides extensions to generalized linear mixed models.

The good news for `lme` users is that the `lmer` function fits a greater range of models, is more reliable, and is faster than the `lme` function. The bad news is that the model specification has been changed slightly. The purpose of this article is to introduce `lmer`, to describe how it can be used to fit linear mixed models and to highlight some of the differences between `lmer` and `lme`.

### Linear mixed models

Before describing how to fit linear mixed models I will describe what they are. In building a statistical model for experimental or observational data we often want to characterize the dependence of a response, such as a patient’s heart rate, on one or more covariates, such as the patient identifier, whether the patient is in the treatment group or the control group, and the time under treatment. We also want to characterize the “unexplained” variation in the response. Empirical models (i.e., models that are derived from the data itself, not from external assumptions on the mechanism generating the data) are generally chosen to be linear in the parameters as these are much simpler to use than are nonlinear models.

Some of the available covariates may be repeatable in the sense that (conceptually, at least) we can obtain new observations with the same values of the covariate as in the current study. For example, we

can recruit a new patient to the study and assign this patient to the treatment group or to the control group. We can then observe this patient’s heart rate at one minute, five minutes, etc. after treatment. Hence we would regard both the treatment factor and the time after treatment covariate as repeatable.

A factor is repeatable if the set of possible levels of the factor is fixed and each of these levels is itself repeatable. In most studies we would not regard the patient identifier factor (or, more generally, the “subject” factor or any other factor representing an experimental unit) as being repeatable. Instead we regard the subjects in the study as a random sample from the population of interest.

Our goals in modeling repeatable covariates and non-repeatable covariates are different. With a repeatable covariate we want to characterize the change in the response between different levels and for this we use *fixed-effects* terms that represent, say, the typical rate of change of the response with respect to time under treatment or the difference between a typical response in the treatment and the control groups. For a non-repeatable covariate we want to characterize the variation induced in the response by the different levels of the covariate and for this we use *random-effects* terms.

A statistical model that incorporates both fixed-effects terms and random-effects terms is called a *mixed-effects model* or, more simply, a *mixed model*.

### Single grouping factor

As indicated above, a random effect is associated with a *grouping factor*, which would be the patient identifier in our example, and possibly with other covariates. We specify a random-effects term in `lmer` by a linear model term and a grouping factor separated by ‘|’, which we would read as “given” or “conditional on”. That is, a random effect is a linear model term conditional on the level of the grouping factor.

Because the precedence of ‘|’ as an operator

is lower than most other operators used in linear model formulae, the entire random-effects expression should be enclosed in parentheses.

Many models for longitudinal data (repeated measurements over time on each of several subjects) incorporate random effects associated with a single grouping factor. Consider the HR (heart rate) data from the SASmixed package

```
> data("HR", package = "SASmixed")
> names(HR)
```

```
[1] "Patient" "Drug"      "baseHR"
[4] "HR"      "Time"
```

Initially we fit a linear mixed model with fixed-effects terms for the base heart rate, the time since administration of the medication, the type of drug and the time/drug interaction. The random effect associated with the patient is a simple additive shift.

```
> (fm1 <- lmer(HR ~ baseHR + Time *
+ Drug + (1 | Patient), HR))
```

```
Linear mixed-effects model fit by REML
Formula: HR ~ baseHR+Time*Drug+(1|Patient)
Data: HR
AIC    BIC logLik MLdeviance REMLdeviance
790.7 815.8 -386.3    791.9        772.7
Random effects:
Groups   Name          Variance Std.Dev.
Patient (Intercept) 44.5      6.67
Residual                    29.8      5.46
# of obs: 120, groups: Patient, 24
```

```
Fixed effects:
              Estimate Std. Error DF
(Intercept)  33.962    9.931 113
baseHR        0.588    0.118 113
Time         -10.698    2.421 113
Drugb         3.380    3.784 113
Drugp        -3.778    3.802 113
Time:Drugb    3.512    3.424 113
Time:Drugp    7.501    3.424 113
              t value Pr(>|t|)
(Intercept)   3.42  0.00087
baseHR        4.97  2.5e-06
Time         -4.42  2.3e-05
Drugb         0.89  0.37358
Drugp        -0.99  0.32244
Time:Drugb    1.03  0.30717
Time:Drugp    2.19  0.03050
```

The first few lines of the output state that the model has been fit using the REML (restricted or residual maximum likelihood) criterion and indicate the formula and the data set used. The next line provides several measures of the quality of the fit including Akaike's Information Criterion, Schwartz's Bayesian Information Criterion, the log-likelihood (actually the log-restricted likelihood because REML

is being used) and the ML and REML versions of the deviance.

The estimates of the variance components are given next. Note that the column headed *Std.Dev.* is **not** the standard error of the estimate of the variance component. It is simply the square root of the estimated variance and is included because many people (and I am one) find it easier to interpret an estimated standard deviation instead of an estimated variance.

Finally the estimates for the fixed-effects coefficients are summarized. If we wish to consider the significance of terms rather than individual coefficients we can use a single-argument call to the *anova* generic which provides the sequential sums of squares and the corresponding F tests.

```
> anova(fm1)
```

```
Analysis of Variance Table
              Df Sum Sq Mean Sq Denom
baseHR        1    746    746    113
Time          1    753    753    113
Drug          2     87     43    113
Time:Drug     2    143     72    113
              F value Pr(>F)
baseHR        25.05 2.1e-06
Time          25.28 1.9e-06
Drug          1.46  0.237
Time:Drug     2.40  0.095
```

At present the denominator degrees of freedom shown in the coefficient table and in the analysis of variance table are upper bounds. In a sense there is no "correct" denominator degrees of freedom because the F test is always an approximation for these models. But, even taking this into account, it is still the case that the denominator degrees of freedom for the Drug term should be lower than shown. The reason that these degrees of freedom are not more accurately approximated at present is because it is difficult to decide exactly how this should be done for the general models described below.

Before changing the fixed effects terms in the model we may wish to examine models with more general specifications of the random effects, such as both a random intercept and a random slope (with respect to time) for each patient.

```
> VarCorr(fm2 <- lmer(HR ~ baseHR +
+ Time * Drug + (Time | Patient),
+ HR))
```

```
Groups   Name          Variance Std.Dev.
Patient (Intercept) 60.6      7.79
         Time          37.8      6.15
Residual                    24.4      4.94
Corr
-0.563
```

```
> head(model.matrix(~Time, HR), n = 3)
```

```
(Intercept)    Time
1             1 0.01667
2             1 0.08333
3             1 0.25000
```

```
> head(ranef(fm2)$Patient, n = 3)
```

```
(Intercept)    Time
201          0.871 4.04733
202         -9.341 6.79574
203          5.005 -0.07822
```

```
> anova(fm1, fm2)
```

```
Data: HR
```

```
Models:
```

```
fm1:HR ~ baseHR+Time*Drug+(1|Patient)
fm2:HR ~ baseHR+Time*Drug+(Time|Patient)
      Df  AIC  BIC logLik Chisq Chi Df
fm1   9  810  835   -396
fm2  11  810  841   -394  3.77    2
      Pr(>Chisq)
fm1
fm2          0.15
```

To save space I summarized this fitted model using `VarCorr` which describes only the estimates of the variance components. Notice that the expression `Time` generates a model matrix (`model.matrix`) with two columns, `(Intercept)` and `Time`, so there are two random effects associated with each patient. The distribution of the random effects is a bivariate normal distribution with mean zero and a positive-definite  $2 \times 2$  variance-covariance matrix. The estimates of the two variances and the estimated correlation are given in the output.

The `ranef` generic function returns the BLUPs (Best Linear Unbiased Predictors) of the random effects and `anova` provides a likelihood ratio test when given multiple model fits to the same data set. As described in [Pinheiro and Bates \(2000\)](#), the p-value calculated for this test will be conservative (i.e. it is an upper bound on the true p-value) because the parameter space is bounded and in the null hypothesis one of the parameters is at the boundary.

## Multiple random-effects expressions

As shown above, the estimated variance-covariance matrix from a random-effects expression, such as `(Time|Patient)`, for which the model matrix has multiple columns is a general, positive-definite, symmetric matrix. (“General” in the sense that there are no constraints imposed on this matrix other than its being positive-definite.)

Occasionally we wish to impose further constraints such as independence of random effects associated with the same grouping factor. For example we can fit a model with an intercept and slope for each patient but assuming independence of these random effects with

```
> VarCorr(fm3 <- lmer(HR ~ baseHR +
+   Time * Drug + (1 | Patient) +
+   (Time - 1 | Patient), HR))
```

```
Groups   Name          Variance Std.Dev.
Patient (Intercept)  47.9    6.92
Patient Time         25.0    5.00
Residual                    25.6    5.06
```

```
> anova(fm1, fm3, fm2)
```

```
Data: HR
```

```
Models:
```

```
fm1: HR~baseHR+Time*Drug+(1|Patient)
fm3: HR~baseHR+Time*Drug+(1|Patient)
      +(Time-1|Patient)
fm2: HR~baseHR+Time*Drug+(Time|Patient)
      Df  AIC  BIC logLik Chisq Chi Df
fm1   9  810  835   -396
fm3  10  811  839   -396  0.84    1
fm2  11  810  841   -394  2.93    1
      Pr(>Chisq)
fm1
fm3          0.358
fm2          0.087
```

The fitted model `fm3` has a random intercept and a random slope for each patient, as does `fm2`, but in `fm3` these random effects are assumed to be independent within patient. Hence `fm3` uses one fewer degree of freedom than does `fm2`.

In general the random effects from each expression are modeled as independent. If the model matrix from the  $i$ th random-effects expression has  $q_i$  columns and the grouping factor has  $k_i$  levels (this number is well-defined because unused levels are dropped during construction of the model frame) the corresponding random effects vector has length  $k_i q_i$  and could be arranged as a  $k_i \times q_i$  matrix as shown in the `ranef` output above. The rows share a common  $k_i \times k_i$  variance-covariance matrix. Different rows are independent.

## Nested and non-nested grouping factors

We have seen an example of multiple random-effects expressions for the same grouping factor. It is more common to create random-effects expressions associated with different grouping factors. For example in a multicenter trial we may have observations over time on each of several patients at each of several institutions and it could be appropriate to use random effects for both patient and institution.

If each patient is observed at only one institution we say that the levels of patient are *nested* within the

levels of institution; otherwise the factors are non-nested. Notice that it only takes one patient migrating between institutions to cause the grouping factors to lose the nesting property. This may not be a primary concern in the analysis of multicenter trials but it is an important issue in the analysis of student test scores over time where it is quite common to have some portion of the students observed at multiple schools. In these cases analysis of the data as if students were nested within schools is at best an approximation and at worst quite inappropriate.

A major distinction between `lme` and `lmer` is that `lme` is optimized for nested grouping factors whereas `lmer` handles nested and non-nested grouping factors equally easily. In `lmer` one simply provides multiple random-effects expressions and the determination of nested or non-nested is done in the code.

Consider the `star` data (Student Teacher Achievement Ratio) from the `mlmRev` package. These data are from a large study - 26,796 observations on a total of 11,598 students in 80 schools. We inferred teacher ids from characteristics of the teacher and classroom, resulting in 1387 distinct teacher ids. (This teacher count is not entirely accurate but it is a reasonable approximation.)

We fit an initial model to the math test scores.

```
> VarCorr(fm4 <- lmer(math ~ gr +
+   sx + eth + cltype + (yrs |
+   id) + (yrs | sch), star))
```

Groups	Name	Variance	Std.Dev.	Corr
id	(Intercept)	1079.2	32.85	
	yrs	22.9	4.78	-0.315
sch	(Intercept)	292.8	17.11	
	yrs	56.8	7.53	-0.777
Residual		584.1	24.17	

```
> anova(fm4)
```

Analysis of Variance Table

	Df	Sum Sq	Mean Sq	Denom
gr	3	1622511	540837	24566
sx	1	9641	9641	24566
eth	5	192909	38582	24566
cltype	2	63848	31924	24566

	F value	Pr(>F)
gr	925.9	< 2e-16
sx	16.5	4.9e-05
eth	66.0	< 2e-16
cltype	54.6	< 2e-16

It happens in this case that the grouping factors `id` and `sch` are not nested but if they were nested there would be no change in the model specification. It is likely that the `lmer` function would converge more rapidly if the grouping factors were nested but even with the nonnested grouping factors in this large study convergence is reasonably fast.

## Specifying levels

Because nested and non-nested grouping factors are expressed in exactly the same way, it is not possible to use implicit nesting when specifying the levels of the grouping factors. Implicit nesting occurs when the levels of the “inner” grouping factor are reused for different levels of the “outer” grouping factor. For example, if the patients are numbered starting at patient 1 for each institution then patient 1 at institution 1 is distinguished from patient 1 at institution 2 only if it is assumed that patient is nested within institution.

For `lmer` each distinct experimental unit must correspond to a distinct level in the corresponding grouping factor. It is easy to create a new grouping factor with this property from implicitly nested factors using the interaction operator ‘:’. The `Pixel` data set in the `MEMSS` package has one grouping factor `Dog` and another factor `Side`. If we wish to fit a model with random effects for “side within dog” we must first create a `dog/side` factor as

```
> Pixel$DS <- with(Pixel, Dog:Side) [drop=TRUE]
```

In this case the subset expression `[drop=TRUE]` is not needed but it is a good practice to use it whenever creating such a factor. The expression results in any combinations of levels that does not occur in the data being dropped from the list of levels in the newly created factor.

## Summary

The `lmer` function in the `lme4` package fits linear mixed models. There are vast changes in the internals of `lmer` relative to the earlier `lme` function and we hope they will ensure that `lmer` is faster, more reliable, and easier to use than was `lme`. The biggest impact on the user is the change in model specification, which was made so as to clarify the model being fit. The `lmer` function uses a single model formula including random-effects expressions that specify both a linear model term and a grouping factor and can be used to fit models based on multiple nested or non-nested grouping factors. It can also be used to fit generalized linear mixed models but that is a topic for another day.

## Bibliography

J. C. Pinheiro and D. M. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000. 27, 29

Douglas Bates  
University of Wisconsin - Madison, U.S.A.  
[bates@wisc.edu](mailto:bates@wisc.edu)