

## Choosing between `lme` and `lmer` for Growth Curve Analysis

Allison Britt

Dan Mirman

### Abstract

Growth curve analysis in R can be carried out using either the `nlme` package or the `lme4` package and, in general, the two packages produce nearly identical results. This document summarizes the syntax differences between the two packages and discusses technical reasons why researchers may wish to choose one or the other method. Specifically, the technical factors covered are: convergence reliability, model output, and running times.

## Choosing between `lme` and `lmer` for Growth Curve Analysis

There are currently several packages in R designed to fit linear mixed models to data. The package Linear and Nonlinear Mixed-Effects Models (`nlme`) was originally used to carry out growth curve analysis (GCA) on eyetracking data from the visual world paradigm (Mirman, Dixon, & Magnuson, 2008). This package is well suited for this kind of analysis, but Linear Mixed-Effects Models Using S4 Classes (`lme4`) is another package that is commonly used for linear mixed effects regression (e.g., Baayen, Davidson, & Bates, 2008; Barr, 2008; Kliegl et al., 2011). In the following report, we describe differences in syntax for running GCA using `nlme` vs. `lme4` along with key differences related to convergence reliability, model output, and running time.

The analyses were carried out in R version 2.10.1 using `nlme` version 3.1-96 and `lme4` version 0.999375-34. We were unable to install `lme4` in Mac R version 2.11, though this problem seems to have been resolved in the newer R 2.12.1 version.

### Syntax

The most basic difference between `nlme` and `lme4` is the syntax used to fit a model. In `nlme`, the critical function is `lme`. Fixed and random effects are included separately. The method defaults to REML, which maximizes the restricted log-likelihood. It can also be set to ML, which maximizes the log-likelihood. The following is an example of the syntax used to run growth curve analysis on eyetracking data with `lme`:

```
full.lme.model <- lme(fixationProportion ~ ot1+ot2+ot3+ot4
  +subj+subj:ot1+subj:ot2+subj:ot3+subj:ot4
  +cond+cond:ot1+cond:ot2+cond:ot3+cond:ot4,
  random = ~ ot1 | subcond, data = data, method="ML")
```

In `lme4`, the critical function is `lmer`. Fixed and random effects are included together in one formula. In `lmer`, REML is a logical argument, which defaults to true. If it is set to false, log-likelihood, rather than restricted log-likelihood, will be maximized. The syntax for the model above using `lmer` instead of `lme` would be:

```
full.lmer.model <- lmer(fixationProportion ~ ot1+ot2+ot3+ot4
  +subj+subj:ot1+subj:ot2+subj:ot3+subj:ot4
  +cond+cond:ot1+cond:ot2+cond:ot3+cond:ot4
  + (ot1 | subcond), data=data, REML=FALSE)
```

In every test we have run so far, the parameter estimates produced by `lme` and `lmer` were identical and the t-values were nearly identical (approximately 1% difference for empirical data, about 6% difference for simulated data, with `lmer` consistently producing greater values). In summary, the main differences between `lme` and `lmer` are how random effects are included and how the method for fitting the model is specified.

### Convergence

Occasionally, using `lme` will result in a convergence problem with the following error message:

```
nlminb problem, convergence error code = 1
message = iteration limit reached without convergence (9)
```

We have not yet discovered the cause for this error or a reliable solution, but `lmer` does not appear to run into this problem – neither for simulated data nor for empirical data. It seems that the convergence algorithm in `lmer` is more robust than the one in `lme`.

## Output

The `summary` function provides parameter estimates, standard errors, degrees of freedom, and *t*-values for the fixed effects in models fit by either `lme` or `lmer` (though for `lmer`, using `print(m, corr=F)` produces somewhat easier-to-read output for a model `m`). Although the contents are very similar, the models produced by the two functions are different kinds of objects: `lme` models are list objects, `lmer` models are S4 class objects. This has implications for accessing model object components such as coefficients. For example, here are the different function calls for getting the fixed effect coefficients from the two models above:

```
full.lme.model$coefficients$fixed
full.lmer.model@fixef
```

## p-values

Users will immediately notice that, unlike `lme` model summaries, `lmer` model summaries do not include *p*-values for the model's fixed effects. Douglas Bates, one of the developers of `lme4`, has written an informative blog post in which he explains why *p*-values are not included (<https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html>). Since experimental psychologists need to report *p*-values, we describe three methods for obtaining them for `lmer` models.

### 1. Model comparisons

For GCA, we recommend gradually building up to the full model (Mirman et al., 2008). On this approach, the effects of a given parameter on model fit can be evaluated using the `anova` function. For example (tutorial walkthrough data from Mirman & Magnuson, 2009):

```
anova(m.base, m.0, m.1, m.2, m.3, m.full)
```

	Df	AIC	BIC	logLik	Chisq	ChiDf	Pr(>Chisq)	
m.base	194	-2681.2	-1652.8	1534.6				
m.0	196	-2742.8	-1703.8	1567.4	65.5892	2	5.721e-15	***
m.1	198	-2755.8	-1706.2	1575.9	16.9983	2	0.0002036	***
m.2	200	-2817.9	-1757.6	1608.9	66.0849	2	4.465e-15	***
m.3	202	-2819.9	-1749.1	1612.0	6.0809	2	0.0478145	*
m.full	204	-2832.5	-1751.1	1620.3	16.5891	2	0.0002499	***

However, as this example makes clear, this method only evaluates the effect of adding factor effects on individual terms without testing specific pair-wise comparisons between

levels within a factor. Running additional analyses on all pair-wise subsets of the data would be one way to test those effects, though this would be somewhat tedious.

## 2. *Markov Chain Monte Carlo sampling*

Bates recommends using Markov Chain Monte Carlo sampling to estimate the p-values, which can be done using `pvals.fnc` from the `languageR` package (see also Baayen et al., 2008):

```
pvals.fnc(full.lmer.model, nsim=10000)
```

However, MCMC sampling is not yet implemented in `lme4` for models with random correlation parameters. It is possible to specify a GCA model with uncorrelated random effects by changing the random effects code from

```
(ot1 | subcond)
to
(1 | subcond) + (0 + ot1 | subcond)
```

in which case `pvals.fnc` will provide p-values for the fixed effects. We do not recommend using this approach because, in general, the random effects in a GCA model of VWP data are correlated and this would produce inaccurate results.

## 3. *Using the normal distribution*

When there are many observations relative to the number of fixed effect parameters, the t-value approximates a z-value and the normal distribution can be used to obtain p-values. These p-values will be slightly anticonservative, but considering that eye-tracking data sets tend to be very large relative to the number of fixed effects, the likelihood of false positives is fairly low. Nevertheless, if using this method, we suggest using a more conservative significance level (e.g., 0.01 instead of 0.05). The following commands will extract the t-values from an `lmer` model object and use a normal distribution to compute p-values for them:

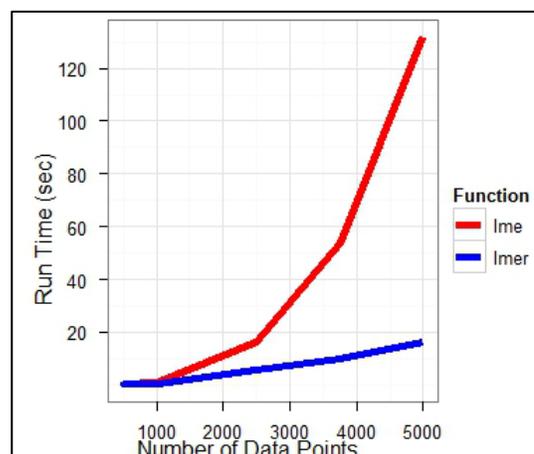
```
coefs <- data.frame(summary(full.lmer.model)$coefs)
coefs$p <- 2*(1-pnorm(abs(coefs$t.value)))
```

## Running times

With a relatively small number of data points ( $N_{\text{points}} = N_{\text{subjects}} * N_{\text{timepoints}} * N_{\text{conditions}}$ ), the two functions both run fairly quickly. However, as the number of data points increases, `lmer` becomes much faster than `lme` (see Figure on right).

## Additional Differences

There are a few broader differences between the `nlme` and `lme4` packages, including the kinds of random effects



they can handle and the kinds of distributions they can fit. The following threads address such differences in more detail:

<http://www.mail-archive.com/r-help@r-project.org/msg109346.html>

<http://markmail.org/message/6003jkjbatt2r6xy>

<https://stat.ethz.ch/pipermail/r-help/2006-October/115572.html>

<http://stats.stackexchange.com/questions/5344/how-to-choose-nlme-or-lme4-r-library-for-mixed-effects-models>

## References

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4), 390-412.

Barr, D. J. (2008). Analyzing 'visual world' eyetracking data using multilevel logistic regression. *Journal of Memory and Language*, 59(4), 457-474.

Kliegl, R., Wei, P., Dambacher, M., Yan, M., & Zhou, X. (2011). Experimental effects and individual differences in linear mixed models: Estimating the relation of spatial, object, and attraction effects in visual attention. *Frontiers in Psychology*, 1:238.

Mirman, D., Dixon, J. A., & Magnuson, J. S. (2008). Statistical and computational models of the visual world paradigm: Growth curves and individual differences. *Journal of Memory and Language*, 59(4), 475-494.

Mirman, D., & Magnuson, J. S. (2009). Dynamics of activation of semantically similar concepts during spoken word recognition. *Memory & Cognition*, 37(7), 1026-1039.