

Appendix A

Additional Information: Linear Mixed-Effects Models for ERP Analyses in Payne, Lee, & Federmeier

This appendix provides additional information about the statistical approach utilized in the current paper. Specifically, this appendix covers the following topics.

1. An overview on linear mixed-effects models (LMMs), parameter estimation, and model fitting using the *lme4* package in *R*, including example code and data reported in Payne, Lee, & Federmeier.
2. Estimation of conditional modes of the random effects for group-specific inference.
3. Comparative and absolute model fit indices for linear mixed-effects models.

Portions of this appendix were adapted from a mini-workshop on using mixed-effects models for event-related potential data analyses by Brennan Payne & Joost Rommers in 2014 at the EEG Reading Group at the Beckman Institute for Advanced Science and Technology.

A1 Fitting Linear Mixed-Effects Models using the *lmer* package in *R*.

This supplement is not meant to be an exhaustive tutorial on using LMMs, but rather to provide an introduction for readers unfamiliar with these methods so that they may understand their implementation in the current study. There are several excellent tutorials, reviews, and instructional texts available, and readers who are interested in learning more about LMMs are encouraged to examine the resources listed at the end of this Appendix.

The prototypical linear mixed-effects model is defined as:

$$(1) \quad \mathbf{y} = \mathbf{X}\boldsymbol{\Gamma} + \mathbf{Z}\mathbf{U} + \boldsymbol{\epsilon}$$

Where :

\mathbf{y} = an n-by-1 response vector.

\mathbf{X} = an n-by-p fixed-effects design matrix.

$\boldsymbol{\Gamma}$ = a p-by-1 fixed-effects vector.

\mathbf{Z} = an n-by-q random-effects design matrix.

\mathbf{U} = a q-by-1 random-effects vector.

$\boldsymbol{\epsilon}$ = an n-by-1 residual error vector.

The random-effects vector is assumed $\mathbf{U} \sim N(0, \mathbf{G})$, where \mathbf{G} is the variance-covariance matrix of the random-effects (i.e., the estimated variance and covariance estimates of the variables defined in matrix \mathbf{Z}). Further, $\boldsymbol{\epsilon} \sim N(0, \sigma^2)$, where $\sigma^2 = \text{var}(\boldsymbol{\epsilon})$. Parameters of the mixed-model are the fixed effects $\boldsymbol{\Gamma}$ and the variance-covariance parameters in \mathbf{G} and σ^2 .

The linear mixed-effects model is a special case of models that are commonly used in psychophysiology, including repeated measures (mixed-effects) ANOVA and ordinary least-squares

regression. LMMs are principally useful in modeling data with complex sampling or clustering, such that observations in the response vector are non-independent. Statistical dependencies of this sort exist, for example, when observations are drawn in a non-random and hierarchically structured manner (e.g., repeated measurements across trials within the same subjects). The model can be conceptualized as being made up of two major components: The fixed-effects ($X\Gamma$) and the random-effects (ZU). The fixed-effects portion of the model is familiar as the classic linear regression model. Parameters from this portion of the model describe the (population-average) linear effect of covariates, defined in the design matrix X , on the response vector. Random-effects index variation in individual experimental units (e.g., grouping, blocking, control, or design variables). LMMs model the variance-covariance structure of the data by representing group-level dependencies (e.g., across subjects, across trials, across time, across populations) as random deviations from the population-average fixed effects.

In any linear statistical model, the goal is to find values of the parameter estimates that maximize the degree of fit between the model predicted values and the observed data. A “loss” or “cost” function quantifies the degree of miss-fit between a model (under certain parameter values) and data. Parameters are estimated that seek to minimize this function (or to maximize the inverse of a loss function, a utility function). The cost function of a simple linear regression model has a well-understood closed-form analytical solution through minimizing the sum of the squared residuals (difference between the observed and model predicted values). No such closed form solution exists for linear mixed-effects models with an unbalanced design or complex random-effects structures. Instead, numerical solutions are used to find model parameters, based on maximum likelihood methods.

The likelihood (\mathcal{L}) of a set of data is the probability of obtaining that particular set of data given the chosen probability distribution model, including unknown parameter values (2).

$$\begin{aligned}
 (2) \quad \mathcal{L} &= \mathcal{L}_1 \times \mathcal{L}_2 \times \mathcal{L}_3 \times \dots \times \mathcal{L}_N \\
 &= \prod_{i=1}^N \mathcal{L}_i \\
 &= p(y_1|\hat{\theta}) \times p(y_2|\hat{\theta}) \times p(y_3|\hat{\theta}) \times \dots \times p(y_N|\hat{\theta}) \\
 &= \prod_{i=1}^N p(y_i|\hat{\theta})
 \end{aligned}$$

Where y_N is the observed data and θ is a given model parameter. The values of the parameters that maximize the likelihood (\mathcal{L}) are known as the Maximum Likelihood (ML) Estimates.

ML algorithms find parameter estimates that maximize the likelihood in an iterative “brute-force” manner. In an initial step, ML algorithms use reasonable “starting” values for all model parameters, then gradually refine these estimates by searching the parameter space for parameters that maximize (-2 times the natural log of) the likelihood function, until the algorithm converges to a solution. Linear mixed-models can be fit in nearly any commercial software package, including SAS, Stata, SPSS, HLM, & MPlus. We will illustrate some examples for fitting LMMs using the *lmer* function in the package *lme4* (Bates, Maechler, Bolker, & Walker, 2014) in the R language for statistical computing (R Core Team, 2014). In addition, a number of useful R packages can be used to supplement LMM analyses in R. For example, packages such as *LMERConvenienceFunctions* (Tremblay & Ransijn, 2013), *effects* (Fox, 2003), *lmerTest* (Kuznetsova et al., 2013), and *ggplot2* (Wickham, 2009) contain useful functions for probing, testing, and visualization of effects in mixed-models;

`pbkrtest` can be used to calculate approximate (Kenward-Rogers) and parametric bootstrap estimates for the likelihood ratio test (Halekoh & Højsgaard, 2014); `languageR` (Baayen, 2008) contains useful tools specific to the analysis of linguistic data, and `blme` (Chung, Rabe-Hesketh, Dorie, Gelman, & Liu, 2013) contains a Bayesian wrapper for the `lmer` function, allowing for specification of prior distributions, for example.

To illustrate basic `lmer()` syntax, we show a simple example of a random-intercepts model (Snijders & Bosker, 2011), which contains one or more fixed-effects and one variance component that models variability in the response across levels of a discrete grouping variable. The following pedagogical syntax is constructed for an example scenario of a simple repeated-measures lexical decision RT experiment with a single factor, `isWord`, that is dummy coded 0 for non-word and 1 for word. A random-effect, *subject*, accounts for individual differences in mean RT. This model is equivalent to the common RM-ANOVA model. Data are organized in *long-format*, such that a single data vector contains all observations for the response variable (and similarly, separate data vectors for individual covariates and grouping/clustering variables).

```
RTmodel.1 <- lmer(RT ~ isWord          #fixed effect
                  + (1 | SubID)        #random effect
                  data = rtData)       #data.frame

summary(RTmodel.1)                    #model summary
```

The `lmer()` function is used to fit a linear mixed-effects model. The response vector (\mathbf{y}), `RT`, is defined as a function (`~`) of the fixed effect (`isWord`) plus a random intercept across subjects (`1 | SubID`). Following, our `data.frame` (an R object for storing data tables) is defined (`rtData`). Note that the model automatically fits with an intercept term and a residual error term. The “#” syntax is used for in-line commenting and is ignored by R when the code is executed. After the model has converged, the `summary()` command prints model information (e.g., estimated parameter values) for the defined object (`RTmodel.1`). This simple code provides a basis for generating more complex models that incorporate multiple fixed-effects (and their interactions), as well as multiple random intercepts and random slope parameters. Multiple fixed-effects can be added by including more variables to the right of the tilde (`~`). Multiple random intercepts can be added to the model (for example, to control for variability across items in a completely crossed design) by adding `+(1 | ranEff)`, where `ranEff` is a data vector describing levels of a discrete random variable (e.g., an item identifier). Lastly, some covariates can be allowed to vary randomly over subjects or items (e.g., the effect of frequency may vary from one subject to the next). For example, the code below describes a slightly more complex model, building off the model defined above.

```
RTmodel.2 <- lmer(RT ~ isWord + workingMemory + timeOfDay #more fixed terms
                  + isWord:workingMemory                #cross-level interaction
                  + (1 + isWord | subID)                 #random slope: isWord
                  + (1 | itemID)                         #random intercept: item
                  data = rtData)                         #data.frame

summary(RTmodel.2)                                     #model summary
```

The `lmer()` function, by default, uses a restricted maximum likelihood algorithm (REML). With smaller sample sizes, REML provides a less-biased estimate of variance components than ML

estimates but, in most cases, the two approaches converge to very similar solutions (see Bates, 2000). One important point to note is that model comparison with REML is limited because the fixed-effects portion of models must remain identical in order to compare their likelihoods. Thus, ML estimation should be used for model comparison. The *anova()* command in R automatically re-estimates models under ML estimation (given an lmer object initially fit with REML estimation) in order to calculate valid comparative fit indices (see section A3 below).

Given this simple tutorial, readers can understand the code used to fit the LMMs in the current paper. The code and output from the two major analyses presented in the paper (Model 1 and Model 2) are presented below. Note that the code and the corresponding data file for fitting these models are available for download from GitHub here: <https://github.com/payne12/singleWordEEG>. In order to run the R code in this document, one needs to install R and the *lme4* package. Model 1, which was fit to test sentence context as a moderator of word position in open-class words, is presented below.

```
lmeModel1 <-lmer(meanAmpEEG ~ zWordOrder + C1 + C2
                + zWordOrder:C1 + zWordOrder:C2
                + (1 | SubID)
                + (1 | chan)
                + (1 | words)
                + (0 + zWordOrder:C1 | SubID)
                + (0 + zWordOrder:C2 | SubID),
                data = centroParietal_OpenClass)
summary(lmeModel1)
```

In this model, mean EEG amplitude within the N400 latency band is defined as a function of a series of fixed-effects: word order, sentence context (defined by two contrasts; see the text for details), and interactions between sentence context and word position. Similar to the above code snippets, random effects (random intercepts and slopes) are defined after the fixed portion of the model is written. Note that the “0 +” syntax fits random slope variance estimates across a defined factor without including covariances (see Barr et al., 2013; see also the *flexLambda* development branch of *lme4* for alternative variance-covariance structures).

The *summary()* function provides several pieces of information. First, information on the estimation method, model formula, and data used are presented, as well as basic information on model convergence. Following, a table of random-effect estimates is presented:

Random effects:			
Groups	Name	Variance	Std.Dev.
words	(Intercept)	3.70050	1.9237
SubID	zWordOrder:C2	0.11948	0.3457
SubID.1	zWordOrder:C1	0.19703	0.4439
SubID.2	(Intercept)	0.57796	0.7602
chan	(Intercept)	0.01818	0.1348
Residual		53.51642	7.3155

The first column (Groups) specifies the grouping variable indicated by the random-effect specification of the model (Z). Name specifies if the variable is a random intercept or a random slope estimate. For slope estimates, the name of the variable is indicated. The last two columns represent the

estimated variance parameters in G . The last row is the residual variance estimate. Following this, the fixed-effects table is printed (along with an optional correlation matrix of the fixed-effect vectors).

Fixed effects:			
	Estimate	Std. Error	t value
(Intercept)	0.62046	0.16730	3.709
zWordOrder	0.55298	0.03156	17.519
C1	-0.63189	0.03950	-15.998
C2	-0.61269	0.03924	-15.614
zWordOrder:C1	-0.45994	0.09390	-4.898
zWordOrder:C2	-0.50095	0.07751	-6.463

The fixed-effect estimates Γ are presented, along with their standard errors. The reported t-value is a simple Wald statistic (*Estimate* divided by *Standard Error*). While these t-values are valid and may be used for statistical inference, note that *p*-values are not printed along with the t-statistics. This is because exact *p*-values and degrees of freedom calculations do not exist for these models (see Baayen et al., 2008 for a discussion), though a number of approximate *p*-value methods do exist. Statistical inference may also be conducted based on likelihood ratio tests comparing nested models (via the *anova* or *drop1* functions). One may additionally request confidence intervals to aid in statistical inference. A number of different confidence intervals (e.g., traditional Wald-based, bootstrapped, or profile-likelihood) can be computed for the parameters of a lmer model via the *confint()* function. This allows for the quantification of variability around effect sizes (e.g., Cummings, 2013) and traditional null-hypothesis statistical inference may be used by evaluating whether values of $b = 0$ are contained within the confidence interval.

The code for Model 2, which was fit to examine the effects of accumulating sentence context on the effects of frequency and orthographic neighborhood, is included below:

```
lmeModel2 <- lmer(meanAmpEEG ~ zWordOrder + zLog_Freq_HAL + zOLD20 #lexical
+ C1 + C2 #sentence context factors
+ zWordOrder:(zLog_Freq_HAL + zOLD20) #two-way interactions
+ C1:(zWordOrder + zLog_Freq_HAL + zOLD20)
+ C2:(zWordOrder + zLog_Freq_HAL + zOLD20)
+ zLog_Freq_HAL:zOLD20
+ zWordOrder:C1:zLog_Freq_HAL #three-way interactions
+ zWordOrder:C2:zLog_Freq_HAL
+ zWordOrder:C1:zOLD20
+ zWordOrder:C2:zOLD20
+ zLength + zConc + zSentLength #covariates
+ (1 | SubID) #random effect structure
+ (1 | chan)
+ (1 | words)
+ (0 + zWordOrder:C1:zLog_Freq_HAL | SubID)
+ (0 + zWordOrder:C2:zLog_Freq_HAL | SubID)
+ (0 + zWordOrder:C1:zOLD20 | SubID)
+ (0 + zWordOrder:C2:zOLD20 | SubID),
data = centroParietal_OpenClass)
summary(lmeModel2)
```

The appended output for this model is included below. First, the random-effects:

Random effects:			
Groups	Name	Variance	Std.Dev.
words	(Intercept)	3.44775	1.8568
SubID	zWordOrder:C2:zOLD20	0.21434	0.4630
SubID.1	zWordOrder:C1:zOLD20	0.16504	0.4063
SubID.2	zWordOrder:C2:zLog_Freq_HAL	0.24270	0.4926
SubID.3	zWordOrder:C1:zLog_Freq_HAL	0.36901	0.6075
SubID.4	(Intercept)	0.57371	0.7574
chan	(Intercept)	0.01817	0.1348
Residual		53.24778	7.2971

Of key interest are the fixed-effect factors coding for three-way interactions between word frequency/orthographic neighborhood, word position, and sentence context (C1/C2). The word frequency interactions were reliable, whereas the neighborhood interactions were not, indicating the key dissociation between the effects of contextual constraint on lexical processing that were the focus of the current paper.

Fixed effects:			
Estimate	Std. Error	t value	
(Intercept)	0.738147	0.170153	4.338
zWordOrder	0.703975	0.033846	20.799
zLog_Freq_HAL	-0.207005	0.095903	-2.158
zOLD20	0.189629	0.085600	2.215
C1	-0.702608	0.039973	-17.577
C2	-0.666154	0.039595	-16.824
zLength	0.040550	0.023838	1.701
zConc	-0.323842	0.076033	-4.259
zSentLength	-0.109941	0.021611	-5.087
zWordOrder:zLog_Freq_HAL	-0.548812	0.035994	-15.247
zWordOrder:zOLD20	-0.057298	0.035775	-1.602
zWordOrder:C1	-0.535151	0.042480	-12.598
zLog_Freq_HAL:C1	0.537880	0.045303	11.873
zOLD20:C1	0.178799	0.044660	4.004
zWordOrder:C2	-0.563952	0.042169	-13.374
zLog_Freq_HAL:C2	0.555302	0.044800	12.395
zOLD20:C2	0.102871	0.044928	2.290
zLog_Freq_HAL:zOLD20	0.007082	0.063506	0.112
zWordOrder:zLog_Freq_HAL:C1	0.396009	0.123997	3.194
zWordOrder:zLog_Freq_HAL:C2	0.552062	0.103956	5.311
zWordOrder:zOLD20:C1	0.013106	0.090885	0.144
zWordOrder:zOLD20:C2	0.026112	0.100018	0.261

A2 Best Linear Unbiased Predictors of the Random Effects

Once a mixed model has been fit to the data, one may wish to examine specific levels of random effects—for example, if one’s goal is to estimate the effect of word position on N400 amplitude for a specific subject, channel, or word. In LMMs, the *best linear unbiased predictors* (also known as conditional modes, posterior means, or empirical Bayes estimates) of the random effects can be computed. BLUPs are not directly estimated as part of the model-fitting procedure, but rather are approximate estimates of unobserved latent variables. BLUP calculation in *lme4*, given a fitted mixed model, is described in Bates, 2013 (p. 22-25). Conceptually, BLUPs can be understood as estimates of group-specific deviations of the mean combined with the overall population mean effect.

BLUPs are so-called *shrinkage estimators* in that group-specific estimates are “shrunk” towards the grand mean differentially more for extreme and less reliable levels, producing more reliable estimators (Gelman, Hill, & Yajima, 2012 provide an accessible review and explanation of the computational benefits of shrinkage in mixed models). Briefly, group-specific deviations (e.g., across particular subjects, items, channels) from the grand-mean are modeled as samples from a (typically) normal population distribution with some estimated variance. By assuming a parametric distribution, large deviations from the population mean result in a larger penalty, whereby such deviations are differentially “pulled” toward the grand average. Importantly, the degree of this penalty varies as a function of the sample size at the level of the grouping variable, which is an important factor in considering missing data and unbalanced designs within LMMs. For example, a subject with 50 valid trials will provide much more precise estimates (and will contribute more toward the population average effect), whereas a subject with only 8 valid trials will be less stable and will be *shrunk* towards the population mean (providing less information). This weighting feature accommodates unbalanced designs nicely, whereas traditional analyses typically lose this information (e.g., where the mean amplitude an ERP component made up of 8 trials is treated similarly to one made up of 50 trials, although see Luck, 2014 for a weighted average method to overcome this issue that is conceptually similar).

Although BLUPs are biased toward the mean, they possess the statistical property that, for a randomly drawn group, BLUPs have smaller mean-squared error. Thus, the error over all groups will be smaller for BLUPs compared to estimates that do not take into account the overall population effects (e.g., computing separate regression equations for each level of a grouping variable; cf. Dambacher et al., 2006). This tool can be very powerful for exploring the random-effects structure of one’s data. In the current study, for example, we calculated by-channel BLUPs of the word position effect in a whole-head analysis and used the resulting BLUPs to create scalp topography plots, which would otherwise be difficult to estimate and visualize. BLUPs can be calculated using the `ranef()` function in *lme4* (Bates, 2013).

A3 Comparative and Absolute Goodness-of-Fit Indices

In Table 2, we report conditional and absolute fit statistics for the major models fit to the data (Model 1, Model2), and a null model (no fixed-effects) used as a baseline comparison. This section describes the comparative and absolute fit indices used in the current study.

Deviance Test. The deviance test (also known as the likelihood ratio test; see Agresti, 2002; Snijders & Bosker, 2011) is used to compare the difference in model fit between successive (nested) models. This test statistic is calculated as the difference between -2 times the natural logarithm of the likelihood for a full model and a nested model (with fewer fixed effect parameters) and follows an approximate χ^2 distribution with degrees of freedom (*df*) equal to the difference in parameters between

the full and nested models. Thus, test-statistics can be formulated for relative improvement in model fit across successive models.

Information Criteria. Information criteria (IC) based assessments of relative model fit are utilized in model comparison for LMMs (see Scaale & Welsh, 2013). Two common measures, the Akaike Information Criteria (AIC; Akaike, 1973) and the Bayesian Information Criteria (BIC; Schwarz, 1978) are commonly reported. Both measures are derived from the likelihood plus some parameter d_m , reflecting a penalty for model complexity (i.e., number of additional parameters). The (marginal) AIC utilized in the `lme4` package is as follows:

$$(3) \quad \text{AIC: } -2\ln(L) + 2d_m$$

Where L is the likelihood, d_m (the penalization parameter) = $p + q$, where p is the number of fixed-effect parameters (length of Γ) and q is the number of unknown parameters in the variance-covariance matrix G . While the value of AIC cannot be directly interpreted, a model with comparably smaller values of AIC indicates a better model fit. BIC uses a slightly different penalty parameter, imposing a stronger penalty with increasing model complexity. BIC is commonly obtained by taking the marginal AIC and replacing the constant (2) in the penalty by $\log(n)$ to obtain:

$$(4) \quad \text{BIC: } -2\ln(L) + \log(n)d_m$$

Similar to the AIC, BIC values that are smaller indicate a better fitting model. The benefit to IC-based measures is that they can be used to compare non-nested models, unlike the deviance test. Comparisons of the relative selection properties of these two IC statistics can be found in the literature (Yang, 2005).

Absolute Fit (approximate R^2 measures). The R^2 coefficient is an oft-used and favorite summary statistic to quantify the absolute goodness-of fit of fixed-effect models, such as OLS regression, and ANOVA. However, a comparable R^2 measure for (generalized) LMMs does not exist, due to the added complexity of the random-effects structure in the mixed-model (see Snijders & Bosker, 1994; 2011, Nakagawa & Schielzeth, 2013). In a simple linear regression model, only the residual variance is considered. However, in GLMMs, there may be many variance components, and we may be interested in understanding changes in one, some, or a combination of the variance parameters. In addition, it is not uncommon for some variables to reduce residual variance while simultaneously increasing the variance of an estimated random-effect, which can result in negative R^2 values (Snijders & Bosker, 1994), clearly calling into question to veracity of R^2 measures in LMM contexts. In strictly hierarchical LMMs, some researchers have proposed separate R^2 values for each level. This interpretation cannot be extended to more complex models with (partially) crossed random effects (cf. Baayen et al., 2008), however. Thus, exact R^2 measures do not currently exist for LMMs. However, we report two approximate R^2 statistics (that are not without their limitations; see Johnson, 2014; Nakagawa & Schielzeth, 2013; and Nagelkerke, 1991), defined below:

Pseudo- R^2 : This statistic is simply the squared correlation between the model predicted values and the observed values in the data:

$$(4) \quad pR^2 = r(\hat{y}, y)^2$$

This definition has intuitive appeal, cannot be negative, and is easily calculated (see Singer & Willet, 2003), but is limited in that it ignores any structure imposed on the data by the random effects.

Conditional R². More recently, Nakagawa and Schielzeth (2013) and Johnson (2014) have developed *R²* type measures, derived from estimated variance components, which can be applied to arbitrarily complex mixed-effects models. Their conditional *R²* (*cR²*) describes the proportion of variance explained by the combination of the fixed and random factors:

$$(5) \quad cR^2 = \frac{\sigma_f^2 + \sum_{l=1}^u \sigma_l^2}{\sigma_f^2 + \sum_{l=1}^u \sigma_l^2 + \sigma_e^2}$$

Where :

σ_f^2 is the variance of the fixed effect components of the LMM

$\sum_{l=1}^u \sigma_l^2$ is the sum of variance components

σ_e^2 is the residual variance

Johnson (2014) provides an extension of this statistic to accommodate random slope estimates. This function is implemented in the *R.squaredGLMM* function in the MuMIn package in R.

A4 Further Reading

Anderson, C.J., Verkuilen, J & Johnson, T.R. (in press). *Applied generalized linear mixed models: Continuous and discrete data*.

Baayen, R.H. (2008). *Analyzing linguistic data: A practical introduction to statistics using R*. Cambridge University Press.

Bates et al. (under review). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*. <http://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf>

Bates et al. (2010). *lme4* Book draft. <http://lme4.r-forge.r-project.org/book/>

De Leeuw, J., & Meijer, E. (2008). *Handbook of Multilevel Analysis*. NY: Springer.

Gelman & Hill (2007). *Data analysis using regression and multi-level/hierarchical models*. Cambridge: Cambridge University Press.

Journal of Memory & Language (2008). Special issue on “Emerging Data Analysis”

Molenburgs, G. & Verbeke, G. (2007). *Models for Discrete Longitudinal Data*. NY: Springer.

Raudenbush, S.W. & Bryk, A.S. (2002). *Hierarchical linear models: Applications and data analysis methods*, 2nd Edition. Thousand Oaks, CA: Sage.

Snijders, T. & Bosker, R. (2011). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. Thousand Oaks, CA: Sage.

Wood, S. (2006). *Generalized additive models: An introduction with R*. CRC Press.