Performing an RM ANOVA the Mixed-Effects Way

There are two basic research designs in the field of second language research that might call for a repeated-measures ANOVA: either 1) data is collected from the same people at different time periods (longitudinal data as in the Lyster experiment or data obtained after participants undergo different conditions of a treatment) or 2) data is collected from the same people at one time but divided up into categories where each person has more than one score (such as the regular versus irregular verbs in Murphy's experiment). Sometimes this second type consists of just different but related parts (such as different phonemic categories), and sometimes it consists of hierarchically related structures (this is also traditionally called a "split plot" design). Although traditionally the main division has been labeled as temporal versus spatial replication, the term "spatial replication" does not make much sense for data in our field. I will label these as cases of "data category replication." Both of these situations will be illustrated in this section, and I will provide strong arguments as to why mixed-effects modeling can be seen as an improvement on RM ANOVA and really a better way to treat repeated measures data.

However, this section will only contain information on how to do mixed-effects modeling using R. This is because mixed-effects models are less "cookie-cutter" than other models and have a number of conceptual issues that must be decided on and tested, and this is more easily done in a syntax environment rather than a windows environment. In fact, almost all of the exposition I have seen about how to do a mixed-effects model in SPSS actually gives SPSS syntax instead of showing how the windows work. Since SPSS users can use R for free, I don't see any reason to

try to give two different types of syntax here, and so will instead concentrate on R. SPSS users are referred to Section 1.2 in the book for directions on how to download R and R Commander, and can easily use the menu interface in R Commander to get data into R (see Section 1.3.2 in the book). After that, the code in this section will walk users through the process of the mixed-effect modeling in R.

Many statistical textbooks in general recommend analyzing repeated measures that have both temporal and data category replication by using a mixed-effects linear model (Crawley, 2007; Everitt & Hothorn, 2006; Faraway, 2006; Venables & Ripley, 2002), and a recent paper by Cunnings (2012) provides arguments as to why these kinds of models are useful for second language researchers and some details for their usage. Cunnings (2015) should provide even more help, especially with modeling change in language over time. Such models may not be familiar to readers as it is only recently that computing power has increased to the point where such models are feasible for practical implementation (Galwey, 2006). However, these models make more precise predictions than traditional ANOVA models possible because they use more realistic assumptions about the residuals (Baayen, 2008, Bontempo & Kemper, 2013) as well as provide broader validity than traditional regression or ANOVA models (Galwey, 2006). They can combine both categorical and continuous (such as age) variables in one analysis and can model change over time (Cunnings, 2012). Another benefit is that they can use all of the data available in a model even if there are incomplete cases, which can be especially helpful for longitudinal data where participants may miss some testing periods (Cunnings, 2012, Bontempo & Kemper, 2013). They can also deal well with hierarchical data such as the Gass & Varonis (1994) model, and Quene & van den Bergh (2004) state that Monte Carlo simulations show

multi-level modeling (another name for mixed-effects modeling) has higher power than ANOVA for repeated-measures analysis.

Linear models we have previously seen in this book postulate that any variance that is not explained by the included variables is part of the error. It is clear that part of this error is variations among individuals—in other words, different people are different, whether they all receive the same experimental treatment or not. Since these individuals differences are not built into the variables of a linear model, what we may call the "subject effect" (the individual variation that is inevitable) is always part of the error in a linear model.

We noted previously in the chapter that RM ANOVA does address this issue. Repeated measures models, along with mixed-effects models, recognize within-subject measures as a source of variation separate from the unexplained error. If a "Subject" term is included in the model, this will take the variance due to individuals and separate it from the residual error. The way that RM ANOVA and mixed-effects models differ, however, is in how they go about calculations. Solving RM ANOVA calculations is a straightforward process which can even be done easily by hand if the design is balanced, as can be seen in Howell's (2002) Chapter 14 on RM ANOVA. This calculation is based on least-squares (LS) methods where mean scores are subtracted from actual scores. Galwey (2006) says that solving the equations involved in a mixed-effects model is much more complicated and involves fitting a model using a method called residual or restricted maximum likelihood (REML). Unlike LS methods, REML methods are iterative, meaning that the fitting process requires recursive numerical models (Galwey, 2006). This process is much smarter than LS; for example, it can recognize that if subjects have extreme

scores at an initial testing time, on the second test with the same subjects they will have acclimated to the process and their scores are not bound to be so extreme and it adjusts to anticipate this, essentially "considering the behavior of any given subject in the light of what it knows about the behavior of all the other subjects" (Baayen, 2008, p. 302).

There is then ample reason to consider the use of mixed-effects models above a traditional RM ANOVA analysis for repeated measures data. This section, however, can only scratch the surface of this broad topic and I do recommend that readers take a look at other treatments of mixed-effects models that use R and include less mathematics for the social scientist, including Bliese (2013), Crawley (2007) and Galwey (2006), and works by psycholinguists such as Baayen (2008), which focuses on examples with reaction times and decision latencies. I also benefitted from reading Cunnings (2012), which contains a worked example of rating sentences.

How Mixed-effects Models Differ from other Regression Models (Hint: Random Effects)

When we run a regular regression (or ANOVA, which as you will have seen if you use the R syntax, is a regression model), what is it we want to know? We want to know whether factors we included in the model help explain the within-group variance in the response variable. Remember that in a regression we reported R^2, which is the percentage of variance explained by a model. In an ANOVA, we reported which interactions and main effects significantly contributed to explaining the variance of the model, and then we talked about effect sizes for those parts of the regression (ANOVA) model.

A mixed-effects model is called mixed because it can contain both fixed and random effects. The fixed part is the part we have been using up until now, and the random part is the new part we will be adding on, which nests individuals within groups. This part will help us know what the regular regression (or ANOVA) let us know as well as adding on two additional pieces of information, according to Bliese (2013). One is "a term that reflects the degree to which groups differ in their mean values (intercepts) on the dependent variable" (p. 51). We will look at the variance term for intercepts and see how large it is. The second additional piece of information is how individuals that are in the same group vary individually, and this is the "degree to which slopes between independent and dependent variables vary across groups" (ibid.). In other words, we will be looking at information about individuals or groups by allowing the slope and/or intercept of their regression lines to vary.

In summary, a mixed-effects model will keep the part that you already understand from a regression analysis but add in information about random effects that will be understood by looking at the size of the variances of intercepts and slopes.

I think this concept is really best understood visually. Figure 1 shows a series of idealized pictures of data from an imagined study on how NS and three groups of learners of Swahili pronounce different phonemic contrasts. First there is a means plot in box A, which is the kind of information we are used to seeing about how each group performed. The second box shows a regression line that might be drawn over the data from each group, where there is only one line that tries to account for all of the data from all of the groups. This is the information we get from a normal regression, which returns an intercept value and then coefficients for the (fixed) terms

that we have entered into the equation. For example, the equation for the regression line in box B might be y = 3.43 – 7.1 (Group affiliation) + 356.2 (the error). By using a mixed-effects model, however, we can begin to model how groups differ on the dependent variable by allowing their intercepts to vary. Box C shows different intercepts for the different groups. This analysis holds the slopes of the groups the same but allows the intercept to be different. Lastly, box D shows what we might find if we allowed both the intercept and the slope of that line to vary according to each group. Gelman and Hill (2007) have a similar picture on p. 238 that I felt really helped me to better understand what it means to let the slopes and intercepts vary for some factor.

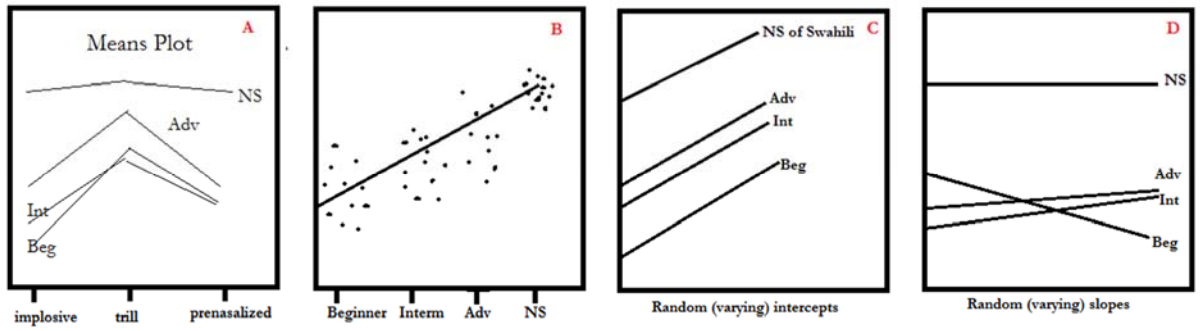

Figure 1 Data in different forms: a) means plots; b) as a linear regression model over all the data; c) as a linear regression model with varying intercepts; d) as a linear regression model with varying intercepts and slopes.

Now understanding what is a fixed effect and what is a random effect is an effort that takes some work and experience, but I will try to provide some guidance here. Note that previous to the RM ANOVA we have only considered fixed effects in our models.

Fixed effects are those whose parameters are fixed and are the only ones we want to consider. Crawley (2007) says that fixed variables have "informative factor levels" (p. 479) and other authors talk about fixed effect levels being the only levels we are interested in. Random effects are those effects where we want to generalize beyond the parameters that comprise the variable. Crawley (ibid.) says that random variables are those that have uninformative factor levels. Gelman and Hill (2007) note that there are actually a number of conflicting definitions of fixed and random variables, and prefer the terms "modeled" (or grouped) and "unmodeled" (not grouped). I understand Gelman and Hill's objections but do not find their terms any clearer so I will continue to use "fixed" and "random." Perhaps some examples from our field will help clarify the idea.

A "subject" term is clearly a random effect because we want to generalize the results of our study beyond those particular individuals who took the test. If "subject" were a fixed effect, that would mean we were truly only interested in the behavior of those particular people in our study, and no one else. Note that the difference between fixed and random factors is *not* the same as between-subject and within-subject factors. A between-subject factor is one where participants belong to only one level, and that grouping will often be a fixed effect, but the participants themselves have individual variation and will still be a random effect. A within-subject factor is a variable that has repeated measurements, and we may want to look at whether there are different slopes and intercepts at those repeated measurements, but that factor may also help explain the fixed part of the equation too. In that case we want to know specifically whether the factor itself, say verb type for the Murphy (2004) dataset, was important for explaining variance. A different repeated factor, such as the classroom that subjects come from, may simply be a

nuisance variable that we want to factor out and generalize beyond, so in this case this would be a random factor that was not used in the fixed part of the equation.

Table 1 (which is also found in Section 2.1.6 in the book) contains a list of attributes of fixed and random effects, and gives possible examples of each for language acquisition studies (although classification always depends upon the intent of the researcher). This table draws upon information from Crawley (2007), Galwey (2006) and Pinheiro and Bates (2000).

Fixed effects	Examples of fixed effects:
have informative labels for factor levels	• treatment type • male or female • native speaker or not • child versus adult • first language (L1) • target language
if one of the levels of a variable were replaced by another level, the study would be radically altered	
have factor levels that exhaust the possibilities	
we are only interested in the levels that are in our study, and we don't want to generalize further	
associated with an entire population or certain repeatable levels of experimental factors	

Random effects	Examples of random effects:
have uninformative factor levels	• subjects • words or sentences used • classroom • school
if one of the levels of a variable were replaced by another level, the study would be essentially unchanged	
have factor levels that do not exhaust the possibilities	
we want to generalize beyond the levels that we currently have	
associated with individual experimental units drawn at random from a population	

Table 1 Characteristics of Fixed and Random Effects.

Crawley (2007, p.628) explains that random effects are drawn from a pool where there is potentially infinite variation, but we "do not know exactly how or why they [populations] differ." For this reason, when looking at random effects, we focus only on how they influence the

variance of the response variable, whereas for fixed effects we can focus on how they influence the *mean* of the response variable.

The Syntax of Mixed-effect Models

Probably the hardest part of creating a mixed-effect model is thinking about the nature of your data and choosing which parts will go into the fixed part and which will go into the random part, which I'm calling the syntax of the model. This is where I feel I may be getting in over my head, but I'm calmed by the advice of Gelman and Hill (2007), who say that you might as well try a mixed-effects model—it really can't hurt! They say that there is little risk in applying it, although if the number of groups is small (they say about 5) there is usually not enough information to be gained beyond classical RM ANOVA by estimating group-level variation.

Another calming idea is that we can try out different models and test them to see whether they are different enough from each other that we should adopt a more complex model. Several authors that I looked at took this approach (Bliese, 2013, Galwey, 2006, Crawley, 2007). However, other authors, such as Gelman and Hill (2007) do not use this type of testing, and Cunnings (2012) specifically advocates letting your theory drive your model specification instead of letting the characteristics of the data drive it. He says that mixed-effect models should employ "maximal" random effects by putting all of the variables that the researcher thinks are theoretically plausible into the random part of the equation, and then report the results. According to Cunnings (2012), model selection is not necessary except perhaps in cases where other control variables might be added and the researcher wants to assess whether they should be included in the model at all (although Cunnings does actually use model selection in his article). Because even after doing a lot of reading I don't feel very confident that I am picking out a

sensible model, I am going to show the method that uses model comparison and testing here. But I wanted readers to be aware that not all authors advocate such an approach and that it is not necessary.

There are two main packages that we could use to model mixed-effects in R, namely the nlme package or the lme4 package. The nlme package seems to me to be better documented; Crawley, 2007, Pinheiro & Bates, 2000 and Galwey, 2006 all provide fairly comprehensive explanations of how to use nlme from the syntax to checking assumptions, whereas I had a hard time finding ways to check the model assumptions with the lme4 package. The nlme package is quite stable, whereas the lme4 package appears to be in a state of some flux and has changed several things over the last few years, such as how to obtain *p*-values and check model assumptions. This may be a good thing for the field as advances are made, but for describing in a textbook such as this one stability is crucial! I wanted to follow Cunnings (2012) and use the lme4 package, but for you, my reader, and for me, I feel there is less frustration at this point if I use nlme, so that is what I will do. The good news is that switching between the syntax used in the models of both packages is relatively easy if you like, and I will show you how to do this later. I also recently noticed a command for ezMixed( ) from the ezStats package. Chapter 11 of the book uses the ezANOVA( ) command from that package that makes setting up the syntax for the command quite simple, and it's possible that the ezMixed( ) command is similarly simple, but I didn't have time to describe it in this document.

So first download and open the nlme package by typing the following commands into the R Console:

```
install.packages("nlme")
library(nlme)
```

Our data also has to be in the correct format. In general, we want the "long" form of the data, but we want the type of "long" form that we got from changing a "wide" form dataset into the "long" form. That is, we want data repeated for participants over multiple rows. We'll need to have an index of data that then specifies which participant the data came from. If you used R and changed the "wide" forms of the Murphy (2004) and Lyster (2004) data into "long" forms, your data is ready to go. If you have your own data, you'll want to change it to make sure you have more than one row of data per participant (if you don't, you are not working with repeated measures data!). If there are any other variables that are repeated, like verb type or similarity for the Murphy data, or testing time for the Lyster data, there should also be indexes that repeat for that type of data (see Figures 11.8 and 11.9 in the book for this data). In Cunnings' (2012) example, participants judged 20 sentences, half of which were grammatical, so each participant's number was repeated over 20 different rows. Thus, even though there were only 24 participants, there were 480 rows of data ($24 \times 20 = 480$).

We'll look first at the Lyster (2004) data (if you previously converted the Lyster.Written.sav file into the "long" form, you can use that, or import the R file lyster.long), which involves replication of measurement over time (what I called temporal replication). Just to remind you, here is the structure of the data:

```
> str(lyster.long)
'data.frame':   540 obs. of  4 variables:
 $ participant: num  1 2 3 4 5 6 7 8 9 10 ...
 $ cond       : Factor w/ 4 levels "FFIrecast","FFIprompt",.
 $ time       : Factor w/ 3 levels "pretaskcompl",..: 1 1 1
 $ compscore  : num  34 25 25 27 31 26 29 26 31 33 ...
```

Bliese (2013) advocates that your first step be to make a model that includes a participant (or subject) variable in the random effects part of the model and one response variable in the fixed part of the model; this model "estimates how much variability there is in mean Y values (i.e., how much variability there is in the intercept) relative to the total variability" (p. 51). He calls this a "null model."

```
modelL1=lme(fixed=compscore~1, random=~1|participant, data=lyster.long)
```

Notice that the random model, here just (1|participant), has two parts—the part that comes before the vertical bar, and the part that comes after. The part that comes before the vertical bar will allow the *slopes* of the response variable to vary, while the part that comes after the vertical bar will allow the *intercepts* of the response variable to vary. Since participant is only found after the bar, this model is allowing the participants' intercepts to vary (reference box C of Figure 1) but not their slopes.

Tip: Just because mixed-effects models can deal with incomplete datasets doesn't mean they can deal with missing data in the file! I took the lyster.long dataset and purposely put an NA in the file, then tried the same model as above:

```
> modelL1=lme(fixed=compscore~1, random=~1|participant, data=lysterNA)
Error in na.fail.default(list(compscore = c(34, 25, 25, 27, NA, 26, 29,
  missing values in object
```

You can see I got a warning and could not continue with the analysis. You can add the argument na.action=na.exclude to tell the command the exclude data with NAs like this, and the model runs:

```
modelL1=lme(fixed=compscore~1, random=~1|participant, data=lysterNA, na.action=na.exclude)
```

The problem is that in this dataset, there is only numeric data in the response (dependent) variable, so excluding this NA is in effect losing a case of data. Gelman and Hill (2007) note that when data is in the response variable, the best way of dealing with it is multiple imputation, so if this is your case I recommend looking at Section 1.5 for more information on how to impute values.

If a random variable is found only *after* the vertical bar as seen in the model for the Lyster data, this is called a random intercept type of mixed-effects model. The syntax (1|participant) means that the response variable of scores on the cloze task is allowed to vary in intercept depending upon the participant. Put another way, random intercepts means that "the repeated measurements for an individual vary about that individual's own regression line which can differ in intercept but not in slope from the regression lines of other individuals" (Faraway, 2006, p. 164).

According to Galwey (2006), one should first approach a mixed-effects model by treating as many variables as possible as fixed-effect terms. Those factors that should be treated as random-effects are what Galwey (2006, p. 173) terms "nuisance effects (block effects). All block terms should normally be regarded as random." The Lyster (2004) data does not have any of these nuisance effects, so we will move ahead by adding fixed-effect terms to the model.

Model L1 states that the only predictor of compscore (the score on gender assignment) is the intercept (the grand mean), and says that the intercept can vary as a function of individual variation. This "random intercept model" (Bliese, 2013) may be the only one we need to adequately account for the data. Now that we have the model, we can examine some information about it by using the summary( ) command:

```
summary(modelL1)
```

```
Linear mixed-effects model fit by REML
 Data: lyster.long
       AIC      BIC    logLik
  3440.137 3453.006 -1717.068
```

. . . (rest of output omitted)

The first point to notice is that by default this model has been fit by a technique called REML, which means relativized maximum likelihood (REML). I discussed briefly how this works in Section 11.5, and how this type of fitting is basically too complex for hand calculation, so it must be done by computer. Because we are able to get a printout, the REML fitting converged, but sometimes models do not converge. If that happens, you can try alternative optimizers; Bliese (2013) mentions the argument control=list(opt="optim"), which uses a general purpose optimizing routine. Type help(lmeControl) to look at more options. Other options include using alternative packages to cross-check results or centering and scaling continuous predictors (I give an example of how to do this in the section called "Understanding the output from a mixed-effect model").

Pinheiro & Bates (2000, p. 76) note that if you intend to do comparisons of models with the anova( ) command, you should use the maximum likelihood technique instead of REML. Therefore, I will add the argument method="ML" to the command. I can easily do this without retyping the whole model by using the update( ) function.

```
modelL2<-update(modelL1, method="ML")
summary(modelL2)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: lyster.long
       AIC      BIC    logLik
  3440.076 3452.951 -1717.038
```

I'll just show the top part of the output from the summary, which now shows that the fit is being done by maximum likelihood. This part also contains measures of model fit, such as the AIC, which is the Akaike Information Criterion (AIC), the Bayesian Information Criterion (**BIC**), which penalizes additional parameters more heavily than the AIC, and the log likelihood. These will be useful later in comparing model fits. For all of these numbers, the lower the number the better the fit, but because they are calculated differently not all of these values will always agree! We'll look at additional information in the summary in the following section ("Understanding the output from a mixed-effect model").

Continuing on with the analysis, Lyster (2004) had 4 different teaching conditions for French gender, and we think that condition may help explain why some participants scored higher than others, so let's add that to the model:

```
modelL3=lme(fixed=compscore~cond, random=~1|participant, data=lyster.long,
method="ML")
```

For now, while we're trying to find the best model, I'll just print out the model fit numbers from the summary( ) command:

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: compscore ~ cond + (1 | participant)
   Data: lyster.long

     AIC      BIC   logLik deviance df.resid
  3414.7   3440.5  -1701.4   3402.7      534
```

The AIC value from this model (3414.7) is lower than the value from the model that did not include Condition (model1a: 3440.1), which means that more of the variance is accounted for in model 2. We're happy about that, but we have several more terms we want to enter into the fixed part of the model. We want testing Time, plus the interaction between Time and Condition. However, in order to test whether any models are better than another, we need to only enter one changed part in at a time (Bliese, 2013). So for the next couple of models enter main effect of Time, and then the interaction between Time and Condition in the fixed effect portion:

```
modelL4=lme(fixed=compscore~cond + time, random=~1|participant, data=lyster.long,
method="ML")
modelL5=lme(fixed=compscore~cond + time + cond:time, random=~1|participant,
data=lyster.long, method="ML")
```

The AIC value from model L4 is 3292.2, and from model L5 is 3193.7, so we see the AIC going down as we fit more variables in the equation. Notice that another way of calling for all three arguments for the fixed portion of the model is cond*time and I'll write it this way from now on.

This is the end of everything we want to put in the fixed effects, so now we need to consider the random effects portion. We have let individuals' intercepts vary, but not their slopes. This is unrealistic. Basically, if we are going to let a variable's intercept vary, we want to let its slope vary as well (Gelman & Hill, 2007). We want to think about letting the slope and intercept vary with the variables on the fixed effect side of the equation. Do we think that individuals' slopes varied at the different testing times? Yes, probably. We have already assumed within-group variation on the fixed side of the equation, that is, that different groups' slopes varied at the different testing times, and we modeled that by putting Time in the fixed side of the equation. Now we need to think about whether individuals *between* (or among all) those groups also varied in ways that the Time division doesn't take into account. I'm not sure, but it seems possible, so I'll add Time to the random part of the equation by creating the argument (1+time|participant), which is the participant random slope of time, meaning that the model will let participants have different slopes depending on the time of testing:

```
modelL6=lme(fixed=compscore~cond*time, random=~1+time|participant,
data=lyster.long, method="ML")
```

The random part of this model means that the response variable of scores on the cloze task (compscore) is allowed to vary in slope depending on the time it was tested, and in intercept depending upon the particular subject being measured. Faraway (2006) says this type of model where both slopes and intercepts can vary is a more realistic model, especially for longitudinal data where observations may be highly correlated. The AIC from this model is 3178.5, slightly lower than model M5.

Notice that we only have 3 time periods here, which basically makes it a categorical variable. Time is more appropriately put into the random effects part of the model when it is a continuous variable that is a measurement of growth. Cunnings (2015) explores modeling of longitudinal data in more detail.

So how about condition? Actually, condition is a between-subjects effect and as such, it is not repeated measures and should not be put into the random part of the model (I. Cunnings, personal communication, October 16, 2014).

So this random syntax is not an option. What other choices might we make for the random part of the equation? The variable of Time didn't vary randomly, so we wouldn't want to add a term like (1|time). I think I've considered all of the options I have for the random effects part of the model, so I will stop at Model L6. To test whether Model L6 is my best option, I'll use the anova( ) command to do a test as to whether one model is better than another. This formally tests the fit with the likelihood ratio, evaluated with a chi-square test for significance. We have to make sure to compare the models in order of increasingly complexity though, with just one

different argument per model. This is true of all of my models after L1, so I will put all of those terms in the test:

```
anova(modelL2, modelL3, modelL4, modelL5, modelL6)
```

```
        Model df      AIC      BIC    logLik   Test   L.Ratio p-value
modelL2     1  3 3440.076 3452.951 -1717.038
modelL3     2  6 3414.702 3440.451 -1701.351 1 vs 2  31.37412  <.0001
modelL4     3  8 3292.157 3326.490 -1638.079 2 vs 3 126.54495  <.0001
modelL5     4 14 3193.691 3253.773 -1582.845 3 vs 4 110.46606  <.0001
modelL6     5 19 3178.455 3259.995 -1570.227 4 vs 5  25.23610   1e-04
```

The output shows that all of the models were statistically different from one another because their p-value is less than .05. The model with the lowest AIC and log likelihood is L6, but the one with the lowest BIC is L5. Actually, I have my doubts about model L6 because when I was trying to model these data in the lme4 package, I got this warning:

```
Error: number of observations (=540) <= number of random effects (=540) for
term (time | participant); the random-effects parameters and the residual
variance (or scale parameter) are probably unidentifiable
```

Therefore, even though it might make sense in some cases to choose the model with the lowest AIC whatever the BIC value (since BIC is more conservative than AIC), I'll choose model L5 as my final model and the one that explains the most variance. In the next section we'll look at the summary of this model.

For the Murphy data (use the murphy.long dataset created in this chapter, or import the R file murphy.long if you don't want to create it yourself), here's the structure of the data:

```
> str(murphy.long)
'data.frame':   360 obs. of  6 variables:
 $ group      : Factor w/ 3 levels "NS children",..: 1 1 1
 $ participant: Factor w/ 60 levels "1","2","3","4",..: 1
 $ variable   : Factor w/ 6 levels "reg_proto","reg_int",.
 $ value      : num  5 5 5 5 5 5 5 5 5 5 ...
 $ type       : chr  "reg" "reg" "reg" "reg" ...
 $ similarity : chr  "proto" "proto" "proto" "proto" ...
```

Now let's start with the "null model," with the scores for suffixing verbs modeled by the intercept and random participant intercepts:

```
modelm1=lme(fixed=value~1, random=~1|participant, data=murphy.long, method="ML")
summary(modelm1)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: murphy.long
       AIC      BIC    logLik
  1150.925 1162.584 -572.4626
```

I've just showed the fit criteria in the output so far. If we think about the fixed effects next, we'll want to put in Group and verb Type and Similarity, so I'll do those one at a time so we can check model fit later:

```
modelm2=lme(fixed=value~group, random=~1|participant, data=murphy.long,
method="ML")
modelm3=lme(fixed=value~group + type, random=~1|participant, data=murphy.long,
method="ML")
```

```
modelm4=lme(fixed=value~group + type + similarity, random=~1|participant,
data=murphy.long, method="ML")
modelm5=lme(fixed=value~group + type + similarity + group:type,
random=~1|participant, data=murphy.long, method="ML")
modelm6=lme(fixed=value~group + type + similarity + group:type + group:similarity,
random=~1|participant, data=murphy.long, method="ML")
modelm7=lme(fixed=value~group + type + similarity + group:type + group:similarity +
type:similarity, random=~1|participant, data=murphy.long, method="ML")
modelm8=lme(fixed=value~group*type*similarity, random=~1|participant,
data=murphy.long, method="ML")
```

Let's go ahead and check to see if our full effects model (with all of the components of the three-way interaction) in Model M8 is the best fit for the data:

```
anova(modelm1, modelm2, modelm3, modelm4, modelm5, modelm6, modelm7,
modelm8)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
modelm1	1	3	1150.9252	1162.5835	-572.4626			
modelm2	2	5	1144.0473	1163.4778	-567.0236	1 vs 2	10.87790	0.0043
modelm3	3	6	1013.9493	1037.2659	-500.9747	2 vs 3	132.09799	<.0001
modelm4	4	8	976.2928	1007.3816	-480.1464	3 vs 4	41.65654	<.0001
modelm5	5	10	976.6743	1015.5353	-478.3371	4 vs 5	3.61850	0.1638
modelm6	6	14	975.1961	1029.6016	-473.5981	5 vs 6	9.47817	0.0502
modelm7	7	16	871.9331	934.1108	-419.9666	6 vs 7	107.26296	<.0001
modelm8	8	20	866.1268	943.8489	-413.0634	7 vs 8	13.80631	0.0079

Although some models along the way were not statistically different, it looks like Model M8 has the lowest AIC and log likelihood (although not smallest BIC), and is different from a model

without a three-way interaction, so we will keep Model M8 and move forward with random effects.

In Murphy's data we need a model where the random effects are nested within each other and this allows the intercept to vary for different participants at the level of verb similarity. The nested model is a logical model because the three terms for similarity (Distant, Intermediate and Prototypical) are repeated for both verb types (Regular and Irregular), which means that the verb similarity category is actually nested within the verb type category, as shown in Figure 2. You should only put nested effects in the error term if they are replicated. Pinheiro and Bates (2000, p. 27) say "[N]ested interaction terms can only be fit when there are replications available in the data."

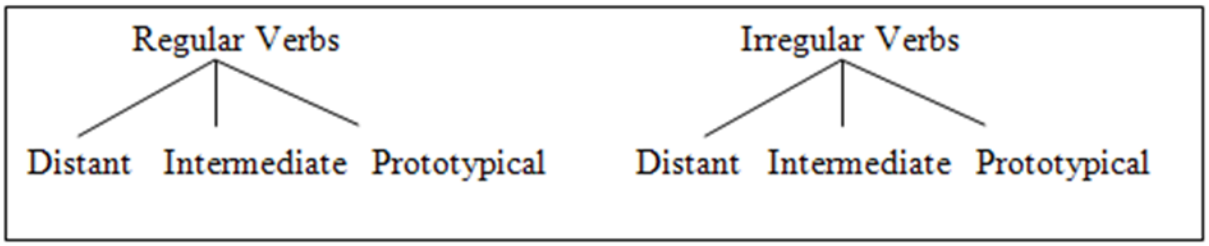

Figure 2 The nested nature of the Murphy terms Type (verb type) and Similarity.

If you have effects that are nested within each other (this design is also called a split-plot design), you will need to put those nested effects in order from largest to smallest (spatially or conceptually) to the right of the vertical bar. We will note, however, that Murphy's model is not actually a split-plot design since all participants were asked to suffix both regular and irregular verbs. With true split-plot designs (such as the Gass and Varonis (1994) hierarchical design described below) the participants are divided by the dominating variable and then the variable

that is dominated is repeated. In other words, if Murphy's design were a split-plot design, participants would have either suffixed regular or irregular verbs only (the dominating variable), but then suffixed verbs from all of the similarity types (the dominated variable).

An example of data category replication which uses hierarchical structures is the study by Gass and Varonis (1994), where 8 of the 16 NS-NNS dyads performed a task using a script, while half had no script. These two groups were further split so that 4 out of the 8 dyads could ask and answer questions about the placement of figures on the board (interaction) while the other 4 dyads could not. Figure 3 represents this hierarchical structure where the presence of interaction is nested within the providing of modified input. In the split plot design, groups are divided on the basis of one variable (here, whether they have the script or not), but in the nested variable their categories are repeated (here, the interaction or no interaction condition is repeated under both the script and no script condition).

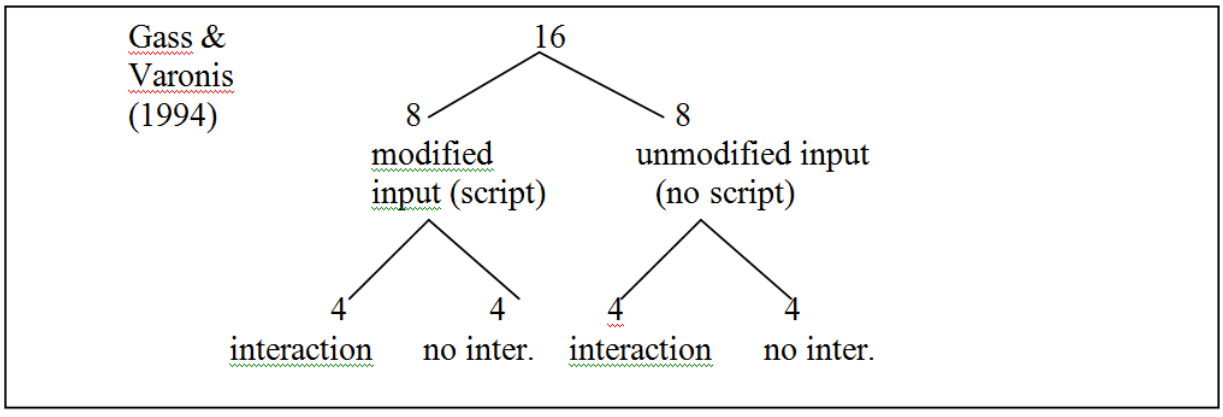

Figure 3 The nested nature of the Gass & Varonis (1994) study.

Here is the syntax for a mixed-effects model for a true split-plot design, such as the Gass & Varonis design (note that this is conceptual as I do not have their actual data):

```
ModelSplitPlot<-lmer(score~ScriptGroup*InteractionGroup,
random=~1|ScriptGroup/InteractionGroup/participant, data=Gass&Varonis)
```

So going back to the Murphy data, I might try nesting within the intercept portion of the random effects, which allows the intercept to vary for different participants at the level of verb similarity:

```
modelm9=lme(fixed=value~group*type*similarity, random=~1|similarity/participant,
data=murphy.long, method="ML")
```

This model has an AIC of 915.0, worse than that of model M8, so we may assume that this is not a good way to model the data. I could also try out other possibilities that the participants' random slope of type or similarity might vary. Notice that I do not want to add group in here because it is not a repeated measure.

```
modelm10=lme(fixed=value~group*type*similarity, random=~1+similarity|participant,
data=murphy.long, method="ML")
```

```
Error in lme.formula(fixed = value ~ group * type * similarity, random = ~1 +  :
  nlminb problem, convergence error code = 1
  message = iteration limit reached without convergence (10)
```

```
modelm11=lme(fixed=value~group*type*similarity, random=~1+type|participant,
data=murphy.long, method="ML")
modelm12=lme(fixed=value~group*type*similarity, random=~type*similarity|participant,
data=murphy.long, method="ML")
```

I got an error message (printed above under model M10) for both M10 and M11 about non-convergence, and for M12 the model just kept working until I pressed the ESC button to stop it, so I assume it also does not converge. I will try using a different iteration method.

```
modelm10=lme(fixed=value~group*type*similarity, random=~1+similarity|participant, data=murphy.long, control=list(opt="optim"))
```

Although it takes a little longer to run, using this method M10 converges without problems, so I add the new iteration method to model M11 and M12 as well and run them (not shown). M11 converges, but M12 does not (again, I have to stop it with the ESC key).

When I try, I find I cannot do a check of all of these models against Model M8 unless I use the same estimation method, so I go back and change the estimation method in models M8 through M10, and then run the anova( ).

```
> anova(modelm8, modelm9, modelm10, modelm11)
         Model df      AIC       BIC    logLik   Test  L.Ratio p-value
modelm8      1 20 896.2860  972.9822 -428.1430
modelm9      2 21 942.8186 1023.3497 -450.4093 1 vs 2 44.53266  <.0001
modelm10     3 25 903.8683  999.7386 -426.9341 2 vs 3 46.95034  <.0001
modelm11     4 22 869.8591  954.2249 -412.9296 3 vs 4 28.00917  <.0001
```

There is a statistical difference between every model, but the one with the smallest AIC, BIC and log likelihood is model M11. This model, with a participant random slope of verb type, explains more of the variance than any of the other models I used.

The example Cunnings (2012) examined had two random terms in their effect on grammaticality ratings for sentences—one was for subjects and the other for items. Unlike the Murphy data, Cunnings' example did not use nested random terms. He had two random terms, and they were crossed (marginally independent), and since he was using the lme4 package, he could just write them separately, like this:

```
FinalModel<-lmer(rating~condition + length + (1+condition|subject) + (1 +
condition|item), data=ratings)
```

It appears that this type of crossed random terms is easy to write in lme4 but not so easy in nlme.

If Murphy had recorded the scores on every item instead of collapsing the data for a possible score of 5, we could have included item as a random factor as well, although we would have had to use a logistic regression mixed effects model since the response variable would have been simple a 1 or a 0 (prefixed or not). This is not difficult to do—one simply specifies a logit model with the term family=binomial(link="logit") and performs the analysis in the same way as we have been doing, but this type of analysis is beyond my scope here. I refer the interested reader to Gelman and Hill (2007), who have an entire chapter on multilevel logistic regression.

This idea of a mixed effects model will surely seem new and confusing. But if you are not sure you have constructed the best model, there is no need to worry because you can try out different models and compare the fit of each model later. Checking the 95% confidence intervals of parameter estimates, both fixed and random, will also help ascertain if there are problems with

the model (I'll show how to do this in the following section). If parameters are extremely wide, this indicates an inappropriate model. Pinheiro and Bates say, "Having abnormally wide intervals usually indicates problems with the model definition" (2000, p. 27). For more reading about syntax that is used in mixed-effect models, see Baayen (2008), Crawley (2007), Faraway (2006), Galwey (2006), and Pinheiro and Bates (2000). If you plan to use mixed-effect models regularly I do recommend further reading as this chapter is only a short introduction.

Understanding the Output from a Mixed-effect Model

You should have some idea now of how the syntax of a mixed-effects model works. Let's examine the output we'll receive when we summarize our created models. To contrast the output, first look at the output generated from a least-squares analysis of the Lyster data.

```
lyster.linear=lm(compscore~cond*time,data=lyster.long)
summary(lyster.linear)
```

. . . output cut . . .

```
Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                      24.5789     0.9008  27.284  < 2e-16 ***
cond[T.FFIprompt]                                 0.4006     1.2004   0.334  0.73869
cond[T.FFIonly]                                   2.3020     1.2433   1.852  0.06465 .
cond[T.Comparison]                                0.8720     1.1900   0.733  0.46402
time[T.post1taskcompl]                            5.1053     1.2740   4.007 7.02e-05 ***
time[T.post2taskcompl]                            4.2105     1.2740   3.305  0.00101 **
cond[T.FFIprompt]:time[T.post1taskcompl]          5.2009     1.6976   3.064  0.00230 **
cond[T.FFIonly]:time[T.post1taskcompl]           -2.2005     1.7583  -1.252  0.21130
cond[T.Comparison]:time[T.post1taskcompl]        -4.3013     1.6830  -2.556  0.01087 *
cond[T.FFIprompt]:time[T.post2taskcompl]          4.5650     1.6976   2.689  0.00739 **
cond[T.FFIonly]:time[T.post2taskcompl]           -2.0201     1.7583  -1.149  0.25112
cond[T.Comparison]:time[T.post2taskcompl]        -3.7988     1.6830  -2.257  0.02440 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.553 on 528 degrees of freedom
Multiple R-squared:  0.2734,    Adjusted R-squared:  0.2583
F-statistic: 18.06 on 11 and 528 DF,  p-value: < 2.2e-16
```

Remember that a linear model assumes that all observations are independent. We have 540 observations, but we know that this experiment only had 180 participants. Therefore, we can see from the error degrees of freedom (528) that this model has pseudoreplication. Crawley (2007) defines pseudoreplication as analysis of the data with more degrees of freedom than you really have. Even if you don't really understand exactly what degrees of freedom are, you need to take notice of them to make sure that your model does not show pseudoreplication. As we will see, models that correctly take the repeated measurements into account will have much lower degrees of freedom.

In the previous section we found that the best mixed-effects model of the Lyster (2004) data was:

```
modelL5=lme(fixed=compscore~cond + time + cond:time, random=~1|participant,
data=lyster.long, method="ML")
summary(modelL5)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: lyster.long
       AIC      BIC    logLik
  3193.691 3253.773 -1582.845

Random effects:
 Formula: ~1 | participant
        (Intercept) Residual
StdDev:    4.336864 3.368128
```

The first line of the output describes what kind of model we have used, which is a model fit by maximum likelihood. The next part of the output gives three measures of goodness of fit, including the AIC, BIC and log likelihood. We used these to help determine the best model.

Next comes information about random effects. Notice that just the standard deviation is given. There are no parameter estimates and no *p*-values. This is because random effects are, given the definition of the model, presumed to have a mean of zero, and it is their variance (or standard deviation, which is the square root of the variance) that we are estimating (Baayen, 2008). In a linear model, we run the model intending to estimate parameters (coefficients) of the fixed models, while in the mixed model, we estimate variances of the random factors as well as parameters of the fixed factors. We have two standard deviations here, one for the effect of subject (defined relative to the intercept, so its value is found under the "Intercept" column), and the other for the residuals, which is the variance the model has not been able to explain. Baayen points out that this residual variance "is a random variable with mean zero and unknown variance, and is therefore a random effect" (2008, p. 268), just as our subject variable is random.

The number for Intercept is the between-group standard deviation, and here it is 4.34, while the number under Residual is the within-group variance, and here it is 3.37. Next to each of these numbers is the standard deviation of this variance. Crawley (2007, p. 628) says "[v]ariance components analysis is all about estimating the size of this variance, and working out its percentage contribution to the overall variation." So here we see two sources of variance—the subjects and the residual (leftover, unexplained) error. Now we'll square these to make them variances, and express each as a percentage of the total, as Crawley (2007, p.640) explains:

```
sd=c(4.34, 3.37)
var=sd^2
100*var/sum(var)
[1] 62.38503 37.61497
```

So the participants' variance explains 62% of the model, while there is 37% left over that is unexplained error. We interpret this similarly to the R^2 effects of a regression model. Starkweather (2010) says that if all the percentages for the random effects are small, then random effects are not present and we do not need to perform a mixed-effects analysis and can revert to normal least-squares analysis. The conclusion here is that a model that allows for random variation among individuals is better than one that does not.

I continue with the output, which shows the fixed effects of the model in regression format.

```
Fixed effects: compscore ~ cond + time + cond:time
                                            Value Std.Error  DF   t-value p-value
(Intercept)                                24.578947  0.900847 352 27.284263  0.0000
cond[T.FFIprompt]                           0.400644  1.200363 176  0.333769  0.7390
cond[T.FFIonly]                             2.302005  1.243287 176  1.851548  0.0658
cond[T.Comparison]                          0.872033  1.190038 176  0.732777  0.4647
time[T.post1taskcompl]                      5.105263  0.781433 352  6.533206  0.0000
time[T.post2taskcompl]                      4.210526  0.781433 352  5.388211  0.0000
cond[T.FFIprompt]:time[T.post1taskcompl]    5.200859  1.041246 352  4.994842  0.0000
cond[T.FFIonly]:time[T.post1taskcompl]     -2.200501  1.078480 352 -2.040373  0.0421
cond[T.Comparison]:time[T.post1taskcompl] -4.301342  1.032290 352 -4.166796  0.0000
cond[T.FFIprompt]:time[T.post2taskcompl]    4.564984  1.041246 352  4.384155  0.0000
cond[T.FFIonly]:time[T.post2taskcompl]     -2.020050  1.078480 352 -1.873053  0.0619
cond[T.Comparison]:time[T.post2taskcompl] -3.798762  1.032290 352 -3.679937  0.0003
```

This is in the format of regression output, so to see the ANOVA results, call for the anova( ) analysis of the model:

```
anova(modelL5)
```

```
            numDF denDF  F-value p-value
(Intercept)     1   352 6270.901  <.0001
cond            3   176   11.171  <.0001
time            2   352  100.759  <.0001
cond:time       6   352   21.069  <.0001
```

We can also use confidence intervals to look at the data:

```
intervals(modelL5)
```

```
Approximate 95% confidence intervals

 Fixed effects:
                                                    lower       est.       upper
(Intercept)                                     22.8270242 24.5789474 26.33087050
cond[T.FFIprompt]                               -1.9418438  0.4006445  2.74313272
cond[T.FFIonly]                                 -0.1242485  2.3020050  4.72825856
cond[T.Comparison]                              -1.4503067  0.8720330  3.19437275
time[T.post1taskcompl]                           3.5855706  5.1052632  6.62495576
time[T.post2taskcompl]                           2.6908337  4.2105263  5.73021892
cond[T.FFIprompt]:time[T.post1taskcompl]         3.1758952  5.2008593  7.22582342
cond[T.FFIonly]:time[T.post1taskcompl]          -4.2978763 -2.2005013 -0.10312620
cond[T.Comparison]:time[T.post1taskcompl] -6.3088883 -4.3013416 -2.29379485
cond[T.FFIprompt]:time[T.post2taskcompl]         2.5400198  4.5649839  6.58994802
cond[T.FFIonly]:time[T.post2taskcompl]          -4.1174252 -2.0200501  0.07732492
cond[T.Comparison]:time[T.post2taskcompl] -5.8063083 -3.7987616 -1.79121487
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: participant
                   lower     est.    upper
sd((Intercept)) 3.827522 4.336864 4.913986

 Within-group standard error:
   lower     est.    upper
3.130882 3.368128 3.623353
```

You will notice that not all of the terms in the regression are statistical. Gelman and Hill (2007, p.271) say that "it is *not* appropriate to use statistical significance as a criterion for including particular group indicators in a multilevel model." Instead, one should start simple and then build up the model (as we have been doing) and try to understand the models being fit. There may be some imprecision in the estimators but that is all right.

In interpreting the estimates of the fixed effects, it is done the same way as for normal regression terms. The estimates give the constant (the intercept) and slope of each predictor. The estimate of the Intercept is simply the constant term for the linear equation. Next, all of the predictors are

compared to a reference, which in this case is the first level of the variable. So for example, cond[T.FFIprompt] is compared to the first level of Condition which is FFIrecast. The estimate of cond[T.FFIprompt] = 0.40, meaning that the mean FFIprompt score is 0.40 higher than the mean FFIrecast score, but the 95% CI [-1.94, 2.74] shows that this difference is not statistical. In fact, none of the comparisons between FFIrecast and the other levels are statistical predictors because the CIs all go through zero. For Time, the first level is pretask, so time[T.post1taskcompl] with an estimate of 5.11 is 5.11 higher than the pretask Time, and this CI does not go through zero [3.59, 6.62]. So what is important to look at is whether the coefficient is negative or positive, which means it is higher or lower than the reference level, and the closer the number is to zero the less effect it has. Besides looking for significance, we want to examine these intervals to see whether they are of a reasonable size. If they are abnormally large, this can indicate that there is something wrong with the model. Here everything looks reasonable.

At the bottom of this output we see confidence intervals for the standard deviations we got from the random part of the data as well. I'll report those, so I'm glad to have them.

I promised that I would tell you how to use lme4 syntax, and this is the time I will stop to do it. At this point using lme4, I was frustrated that there were no *p*-values, but there are workarounds for this on the web, and you know how I feel about *p*-values anyway. But where I really got frustrated and decided to go back and revise to showing using nlme was when I was trying to check model assumptions. I couldn't find any way to check some of them, so I returned to nlme. But many scientists are using lme4, and if you want to use it too, it is not difficult to change your syntax. You will recognize it right away as almost identical to what we have done here. The parts

in red are the parts that get changed (you can actually see more easily what gets taken *away* from nlme in the change to lme4).

```
modelL5<-lme(fixed=compscore~cond*time, random=~1|participant, data=lyster.long,
method="ML")
```

```
modelL5.lme4<- lmer(compscore~cond*time + (1|participant), data=lyster.long,
REML=F)
```

To use lme4, install and open the package:

```
install.packages("lme4")
library(lme4)
```

Whether you use the lme4 or nlme package, the last part of the results are correlations between the regression components:

```
 Correlation:
                                          (Intr) cnd[T.FFIp] cnd[T.FFIn] cn[T.C] t[T.1] t[T.2] cnd[T.FFIp]:[T.1] cnd[T.FFIn]:[T.1] c[T.C]:[T.1 cnd[T.FFIp]:[T.2] cnd[T.FFIn]:[T.2]
cond[T.FFIprompt]                         -0.750
cond[T.FFIonly]                           -0.725  0.544
cond[T.Comparison]                        -0.757  0.568      0.548
time[T.post1taskcompl]                    -0.434  0.325      0.314     0.328
time[T.post2taskcompl]                    -0.434  0.325      0.314     0.328   0.500
cond[T.FFIprompt]:time[T.post1taskcompl]   0.325 -0.434     -0.236    -0.246  -0.750 -0.375
cond[T.FFIonly]:time[T.post1taskcompl]     0.314 -0.236     -0.434    -0.238  -0.725 -0.362  0.544
cond[T.Comparison]:time[T.post1taskcompl]  0.328 -0.246     -0.238    -0.434  -0.757 -0.378  0.568             0.548
cond[T.FFIprompt]:time[T.post2taskcompl]   0.325 -0.434     -0.236    -0.246  -0.375 -0.750  0.500             0.272           0.284
cond[T.FFIonly]:time[T.post2taskcompl]     0.314 -0.236     -0.434    -0.238  -0.362 -0.725  0.272             0.500           0.274           0.544
cond[T.Comparison]:time[T.post2taskcompl]  0.328 -0.246     -0.238    -0.434  -0.378 -0.757  0.284             0.274           0.500           0.568           0.548
```

This part of the output is a full correlation matrix of all of the parameters. Baayen (2008) asserts that these correlations are not the same as what you would get when using the cor() command on

pairs of vectors, but that these numbers "can be used to construct confidence ellipses for pairs of fixed-effects parameters" (p. 268). Since we will not be doing this, Baayen suggests suppressing this part of the output by typing the following line, which then subsequently affects the summary( ) function for Model L5:

```
print(modelL5, corr=FALSE)
```

Fox (2002) concurs that the correlation output is not usually anything we would use, but does note that if there are large correlations this would be indicative of problems within the model. I have not talked about centering, but sometimes it is necessary to center your data in order to avoid problems with multicollinearity of the terms, which can result in non-convergence of models. Cunnings (2012) says this is often what we want to do with continuous predictors, which in his example are age and length of sentences. It is relatively trivial to center a predictor—all that you do is subtract the mean of that variable from itself. For example, if I had an age variable in the Lyster dataset, I could center this in this manner:

```
lyster.long$cage = lyster.long$age – mean(lyster.long$age)
```

Then I would simple add the variable cage instead of age to the mixed effects model.

One more thing to notice at the bottom of the summary( ) is the number of observations and participants.

```
Number of Observations: 540
Number of Groups: 180
```

Use this information to make sure you do not have pseudoreplication in your model. We see here that everything is in order—we did have 540 observations, but only 180 participants.

Since we found a statistical interaction between condition and groups in the fixed effects part of the model, we would want to continue on to look more closely at that interaction, which could be accomplished as explained in Section 11.6 of the book.

I will leave an analysis of the Murphy model for the reader to work as an exercise, as it does not differ substantially from the Lyster model in its interpretation.

Creating a Mixed-effect Model

1. Use the nlme package (or lme4, if you want to find out more about it) library and the lme( ) command (or lmer( ) if using lme4).
2. Your model has a fixed effect part (which you should be used to seeing by now if you have been working through the book and looking at regression or Anova in R) and a random effects part (which is new). The random effects part has a line separating variables that will affect the slopes (before the line) and variables that will affect the intercepts (after the line). For our field, in many cases the only term you may wish to put in the random effects part is a "Participants" term in order to separate out the individual variation in subjects. Here is an example of one such model:
   Model1=lme(fixed=Score~Condition, random=~1|Subject), data=lyster, method="ML")
3. Evaluate the fixed-effects part of your model by traditional means with regression output or use the anova( ) command.
4. Evaluate the random-effects part of the model by looking at the standard deviation of the random effects and calculating the percentage contribution to the variation in the model. Square the standard deviations to make them variances, and express each as a percentage of the total, like this (parts in red are parts you will replace with your own data):

   ```
   sd=c(4.34, 3.37)
   var=sd^2
   100*var/sum(var)
   ```
   Result:
   ```
   [1] 62.38503 37.61497
   ```

   The calculation shows that the participant effect explained 62.4% of the variance in the model.

Testing the Assumptions of the Model

Just as with any type of regression modeling, we need to examine the assumptions behind our model after we have settled on the best one. The assumptions of a mixed effect model are somewhat different than those we are used to seeing for parametric statistics.

Crawley (2007, p. 628) says there are 5 assumptions:

1. "Within-group errors are independent with mean zero and variance σ^2"
2. "Within-group errors are independent of the random effects"

3 “The random effects are normally distributed with mean zero and covariance matrix Ψ”

4 “The random effects are independent in different groups”

5 “The covariance matrix does not depend on the group”

The plot( ) command is the primary way of examining the first two assumptions that concern the error component. The first plot is a plot of the standardized residuals versus fitted values for both groups. Figure 4 shows this plot.

```
plot(modelL5, main="Lyster (2004) data")
```

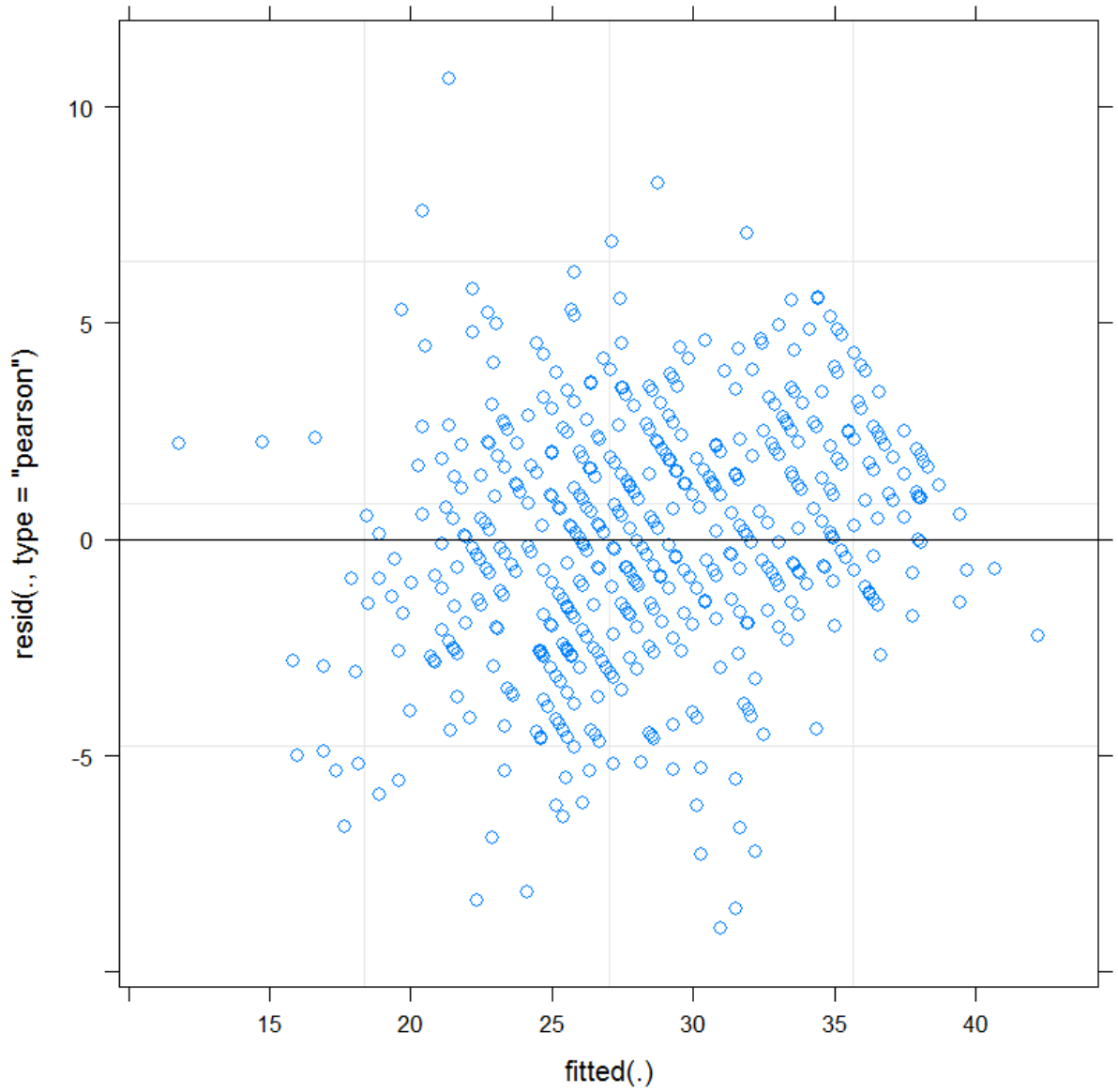

Figure 4 Fitted values vs. standardized residual plots to examine homoscedasticity assumptions for Lyster (2004).

The plot in Figure 4 is used to assess the assumption of the constant variance of the residuals. Remember that these plots should show a random scattering of data with no tendency toward a pie-shape. This residual plot indicates some problems with heteroscedasticity in the residuals because there is a tapering of the data on the right-hand side of the graph, but as it does not

appear extreme we will not worry about it. It is possible to fit variance functions that will model heteroscedasticity (and thus correct for it), but this topic is beyond the scope of this book.

The following command looks for linearity in the response variable by plotting it against the fitted values (the with( ) command makes me specify the dataset for the command):

```
with(lyster.long, plot(modelL5,compscore~fitted(.)))
```

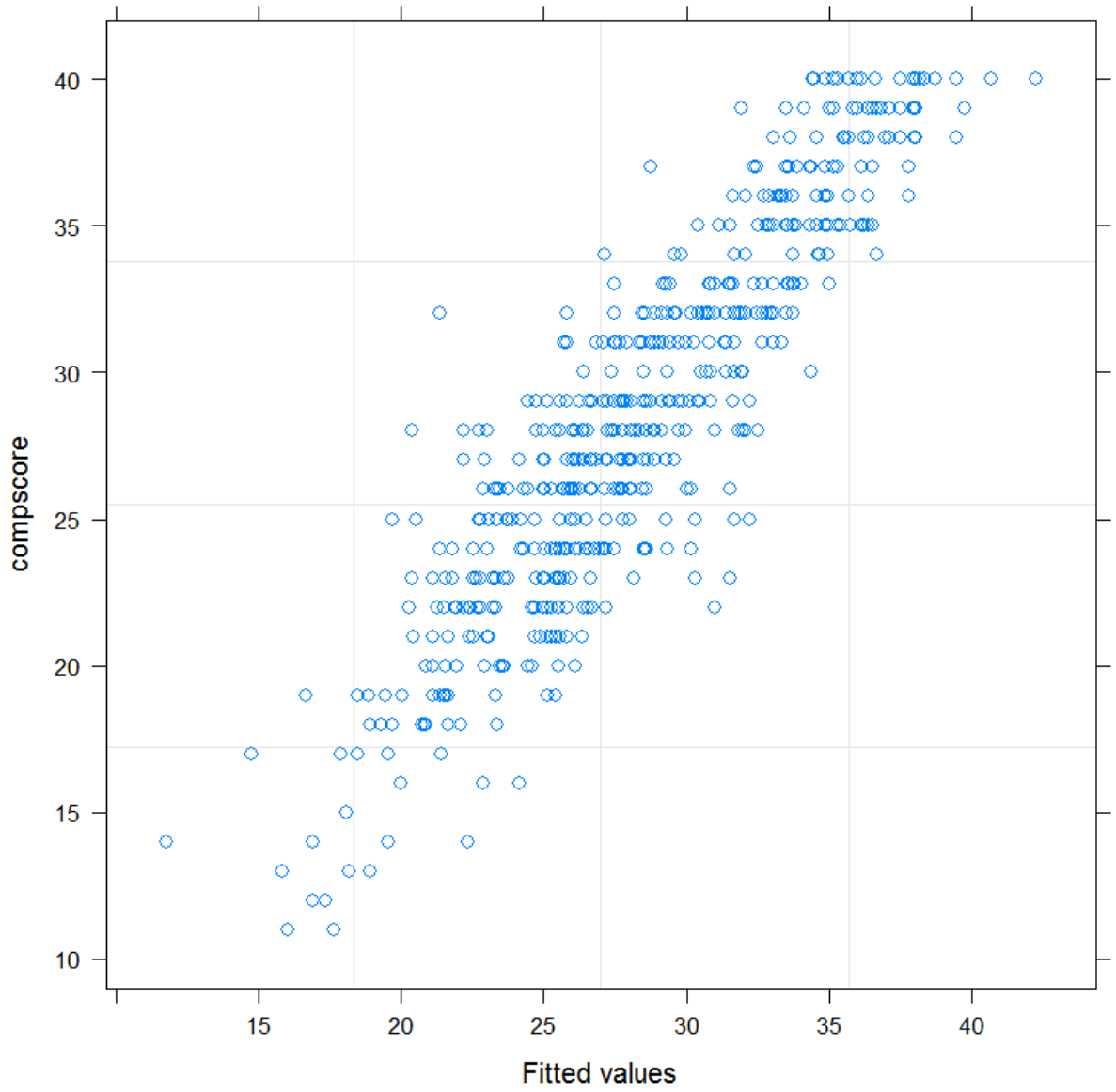

Figure 5 Response variable plotted against fitted values to examine linearity assumption for Lyster (2004) data.

The plot in Figure 5 returned by this command looks reasonably linear. Next we can examine the assumption of normality for the within-group errors (assumption #3) by looking at a Q-Q plot.

```
qqnorm(modelL5,~resid(.)|time)
```

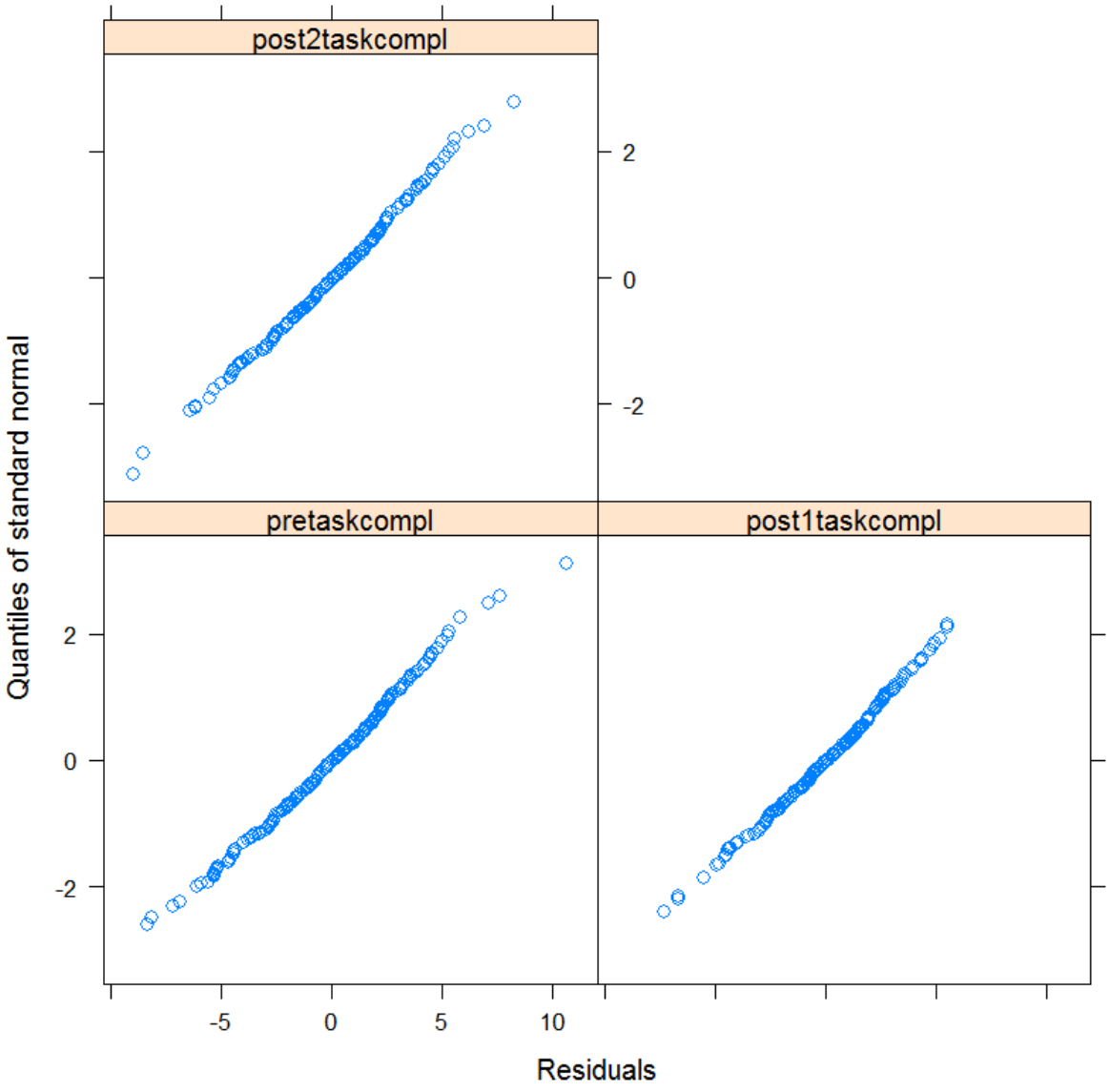

Figure 6 Q-Q plots to examine normality assumption for Lyster (2004) data.

The residuals in Figure 6 look reasonably linear and do not seem to show any departures from normality.

For assumptions about random effects (assumption #4) we will look at a Q-Q plot of the random effects. The estimated best linear estimated predictors (BLUPs) of the random effects can be extracted with the ranef( ) command:

```
qqnorm(modelL5,~ranef(.))
```

This Q-Q plot shows that the assumption of normality is reasonable (it is not shown here). For help in checking the fifth assumption listed here, that of the homogeneity of the random effects covariance matrix, see Pinheiro and Bates (2000).

Reporting the Results of a Mixed-effect Model

When reporting the results of a mixed-effect model, Winter (2013) wisely notes that you should describe what you did so that someone else can reproduce your analysis. Thus, you'll need to say what package (and what version of it) you used in R, and describe all your fixed effects and random effects as well as say whether you have random intercepts and slopes. Actually, as Cunnings (2012) says, "[a]s a relatively new tool in language research, the conventions for best practice in the use of mixed-effects models are still under debate" (p. 378). So it's hard to give firm advice about what to report, but basically, I would want to at least understand (see) the syntax of the final model that you used, and it could be good in many cases to see the process you went through in finding the best model. Winter (2013) also points out that it's important to give the people who gave their free time into making the packages and R some credit too. You can find out your version and how to cite it by typing:

citation("nlme") #or just type citation( ) with nothing inside for your R version

What will you report about your final model? For the fixed effects report the fixed-effect parameter estimates (regression coefficients) and CIs for those estimates (or *p*-values). If you feel your audience is looking for an ANOVA summary, give that information for the fixed effects as well as or instead of the regression estimates. For the random effects, variances (the standard deviations squared) should be reported and the percentage variance that the random effects explain can be noted. And of course you should note whether your model satisfied model assumptions.

Here is a sample report about a mixed effect model of the Lyster data with the Cloze task:

> Using the nlme package (Pinheiro, Bates et al., 2014) from R (R Core Team, 2014), I found the best model for the Cloze task in Lyster (2004) was one that had Condition and Time and their interaction as explanatory variables for performance on the cloze task in the fixed effects part of the model. The final model looked like this:
>
> fixed=ClozeTask~Cond*Time, random=~1|Subject
>
> The relationship between the cloze task and the independent variables of Condition and Time showed considerable variance in intercepts across participants, sd = 4.34, 95% CI [3.83, 4.91], accounting for 62% of the variance in the model. Analysis of the residuals did

not reveal any problems important problems with normal distribution or independence of random effects.

For the fixed effects part of the model you have the choice as to whether to report ANOVA results or regression results. For this model, regression results would probably best be summarized in a table that listed the B coefficient (not forgetting the intercept) and the confidence interval for the coefficient. Here is an example for a selected few results:

	β	95%CI
Intercept	24.58	[22.83, 26.33]
Condition (FFI prompt) compared to baseline FFI recast	0.40	[-1.94, 2.74]
Time (Immediate posttask) compared to baseline pretask	5.11	[3.18, 7.23]
Interaction of Condition (FFI prompt) and Time (Immediate posttask) compared to baseline Condition:Time	5.20	[3.18, 7.23]

For the ANOVA results, you could just give the results in a table, or in prose as here: In the fixed effects part of the model, the main effect of Condition was statistical ($F_{3,176} = 11.2$, $p<.0001$), the main effect of Time was statistical ($F_{2,352} = 100.8$, $p<.0001$) and the interaction of Condition and Time was statistical ($F_{6,352} = 21.1$, $p<.0001$).

Note that you would want to do further analysis of these results but you wouldn't use a mixed-effects model to do it, so I won't report on them here.

Application Activity for Mixed-effect Models

N.B. Answers for this activity are given below, not in a separate document as in most cases for this book, and only selected answers are given.

1 Recreate the mixed-effect model proposed for the Murphy dataset in the section of this paper titled "The syntax of mixed-effect models" and verify that the final model you end up with is the same as mine (look at AIC, BIC and log likelihood numbers to choose the best model). Then run a summary on the data and report on what you find for the random effects and for an ANOVA analysis of the fixed effects. Don't forget to look at model assumptions.

2 Write a maximal mixed-effects model (the model with the most information) for the following research designs which have been detailed in other places in this chapter. Use the nlme syntax. There is no right answer for the random effects part, but in most cases you will want to let intercepts vary by participant.

a Chrabaszcz & Gor (2014): 2(Group) × 3 (Contrast) × 2 (Vowel) RM ANOVA, repeated measures on Contrast and Vowel, response variable was score for listening discrimination. Assume a participant variable.

b Erdener and Burnham (2005): 4(condition of orthographic and audio-visual material) × 2(L1) × 2(L2) RM ANOVA, repeated measures on Condition and L2, response variable was score. Assume a participant variable.

c Toth (2008): 3 (Testing Times) × 3 (Groups) RM ANOVA, repeated measures on Testing Times, response variable was accuracy of productions of Spanish clitic *se*.

d Cole (1927): 3 (Language Measure) × 2 (Testing Time) × 2 (Teaching Method) RM ANOVA, repeated measures on Language Measure and Testing Time, response variable was score on test.

3 Lyster (2004) binary-choice test. In Section 11.2.4 in the book you created a long-form file for Lyster's binary-choice test. Use this to run a mixed-effects model and use the results of anova( ) to report on statistical terms for the fixed effects. Report on the variances for the random effects, and calculate percentages for variances. Describe your search for the best model. Perform checks on model assumptions and report on how well the data fit the assumptions.

Selected Answers for Application Activity for Mixed-effect Models

- 2a. For example: Model=lme(fixed=group*contrast*vowel, random=~contrast* vowel|participant)
- 2b. For example: Model=lme(fixed=condition*L1*L2, random=~condition*L1*L2|participant)
- 2c. For example: Model=lme(fixed=group*Time, random=~Time|participant)
- 2d. For example: Model=lme(fixed=Measure*Time*Method, random=~time*Measure|participant)

Bibliography

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

Bliese, P. (2013). Multilevel modeling in R (2.5): A brief introduction to R, the multilevel package and the nmle package [Software]. Available from http://cran.r-project.org/doc/contrib/Bliese_Multilevel.pdf

Bontempo, D. & Kemper, S. (2013). Multilevel analyses for repeated measures. *BNCD Newsletter: Analytical Techniques & Technology Core*, 1–2.

Chrabaszcz, A. & Gor, K. (2014). Context effects in the processing of phonolexical ambiguity in L2. *Language Learning*, 64(3), 415–455.

Cole, R.D. (1927). Free Composition vs. Translation into the foreign language in development of ability to write a foreign language. *Modern Language Journal*, 11(4), 200–206.

Crawley, M. J. (2007). *The R book*. New York: Wiley.

Cunnings, I. (2012). An overview of mixed-effects statistical models for second language researchers. *Second Language Research*, 28(3), 369–382.

Cunnings, I. (2015). Mixed effects modeling and longitudinal data analysis. In L. Plonsky (Ed.), *Advancing quantitative methods in second language research*. New York: Routledge.

Erdener, V. D., & Burnham, D. K. (2005). The role of audiovisual speech and orthographic information in nonnative speech production. *Language Learning*, *55*(2), 191–228.

Everitt, B., & Hothorn, T. (2006). *A handbook of statistical analyses using R*: CRC.

Faraway, J. J. (2006). *Extending the linear model with R: Generalized linear, mixed effects, and nonparametric regression models*. New York: CRC.

Fox, J. (2002). Linear mixed models. On *An R and S-PLUS companion to applied regression*. R home page.

Galwey, N. W. (2006). *Introduction to mixed modelling: Beyond regression and analysis of variance*. Chichester, West Sussex: Wiley.

Gass, S., & Varonis, E. (1994). Input, interaction, and second language production. *Studies in Second Language Acquisition, 16*, 283–302.

Gelman, A. & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. New York: Cambridge University Press.

Howell, D. C. (2002). *Statistical methods for psychology* (5th ed.). Pacific Grove, CA:

Duxbury/Thomson Learning.

Lyster, R. (2004). Differential effects of prompts and recasts in form-focused instruction. *Studies in Second Language Acquisition*, *26*(4), 399–432.

Murphy, V. A. (2004). Dissociable systems in second language inflectional morphology. *Studies in Second Language Acquisition*, *26*(3), 433–459.

Pinheiro, J. C., & Bates, D. (2000). *Mixed-effects models in S and S-PLUS*. New York: Springer.

Quene, H. & van den Bergh, H. (2004). On multi-level modeling of data from repeated measures designs: A tutorial. *Speech Communication,* 43, 103–21.

Starkweather, J. (2010). Linear mixed effects modeling using R Retrieved from: http://www.unt.edu/rss/class/Jon/Benchmarks/LinearMixedModels_JDS_Dec2010.pdf

Toth, P. (2008). Teacher- and learner-led discourse in task-based grammar instruction: Providing procedural assistance for L2 morphosyntactic development. *Language Learning*, 58(2), 237–283.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S*. New York: Springer.

Winter, B. (2013). Linear models and linear mixed effects models in R with linguistic applications. arXiv:1308.5499. [http://arxiv.org/pdf/1308.5499.pdf]