*Tutorial*

Advances in Methods and Practices in Psychological Science
January-March 2021, Vol. 4, No. 1, pp. 1–19

DOI: 10.1177/2515245920960351
www.psychologicalscience.org/AMPPS

# An Introduction to Linear Mixed-Effects Modeling in R

**Violet A. Brown**
Department of Psychological & Brain Sciences, Washington University in St. Louis

**Abstract**

This Tutorial serves as both an approachable theoretical introduction to mixed-effects modeling and a practical introduction to how to implement mixed-effects models in R. The intended audience is researchers who have some basic statistical knowledge, but little or no experience implementing mixed-effects models in R using their own data. In an attempt to increase the accessibility of this Tutorial, I deliberately avoid using mathematical terminology beyond what a student would learn in a standard graduate-level statistics course, but I reference articles and textbooks that provide more detail for interested readers. This Tutorial includes snippets of R code throughout; the data and R script used to build the models described in the text are available via OSF at https://osf.io/v6qag/, so readers can follow along if they wish. The goal of this practical introduction is to provide researchers with the tools they need to begin implementing mixed-effects models in their own research.

**Keywords**

mixed-effects modeling, R, language, speech perception, open data

Received 4/11/20; Revision accepted 8/20/20

In many areas of experimental psychology, researchers collect data from participants responding to multiple trials. This type of data has traditionally been analyzed using repeated measures analyses of variance (ANOVAs)—statistical analyses that assess whether conditions differ significantly in their means, accounting for the fact that observations within individuals are correlated. Repeated measures ANOVAs have been favored for analyzing this type of data because using other statistical techniques, such as multiple regression, would violate a crucial assumption of many statistical tests: the *independence assumption*. This assumption states that the observations in a data set must be independent; that is, they cannot be correlated with one another. But take, for example, a reaction time study in which participants respond to the same 100 trials, each of which corresponds to a different item (e.g., a particular word in a psycholinguistics study). Reaction times within a given participant and within an item will certainly be correlated; some participants are faster than others, and some items are responded to more quickly than others. Given that observations are not independent, data in which participants respond to multiple trials must be analyzed with a statistical test that takes the dependencies in the data into account.

For this reason, when analyzing data in which observations are nested within participants, repeated measures ANOVAs are preferable to standard ANOVAs and multiple regression, which both ignore the hierarchical structure of the data. However, repeated measures ANOVAs are far from perfect. Although they can model either participant- or item-level variability (often referred to as $F_1$ and $F_2$ analyses in the ANOVA literature), they cannot simultaneously take both sources of variability into account, so observations within a condition must be collapsed across either items or participants. When the data are aggregated in this way, however, important information about variability within participants or items is lost, which reduces statistical power (see Barr, 2008), that is, the likelihood of detecting an effect if one exists.

**Corresponding Author:**
Violet A. Brown, Department of Psychological & Brain Sciences, Washington University in St. Louis
E-mail: violet.brown@wustl.edu

Another limitation of ANOVAs is that they deal with missing observations via *listwise deletion*; this means that if a single observation is missing, the entire case is deleted, and none of the observations from that individual (or item) will be used in the analysis. Depending on the number of complete cases in the data set, this can substantially reduce sample size, which leads to inflated standard error estimates and reduced statistical power (though the estimates will be unbiased if the data are missing completely at random; see Enders, 2010). ANOVAs also assume that the dependent variable is continuous and the independent variables are categorical; experiments in which the outcome is categorical (e.g., accuracy at identifying particular items in a recognition memory task) must be aggregated or analyzed using a different technique, and continuous predictors (e.g., time in a longitudinal study) must be treated categorically (i.e., binned), which reduces statistical power and makes it difficult to model nonlinear relationships between predictors and outcomes (e.g., Liben-Nowell et al., 2019; Royston et al., 2005). A final drawback of ANOVAs is that although they indicate whether an effect is significant, they do not provide information about the magnitude or direction of the effect; that is, they do not provide individual coefficient estimates for each predictor that indicate growth or trajectory.

## Mixed-Effects Models Take the Stage

These shortcomings of ANOVAs and multiple regression can be avoided by using linear mixed-effects modeling (also referred to as multilevel modeling or mixed modeling). Mixed-effects modeling allows a researcher to examine the condition of interest while also taking into account variability within and across participants and items simultaneously. It also handles missing data and unbalanced designs quite well; although observations are removed when a value is missing, each observation represents just one of many responses within an individual, so removal of a single observation has a much smaller effect in the mixed-modeling framework than in the ANOVA framework, in which all responses within a participant are considered to be part of the same observation. Participants or items with more missing cases also have weaker influences on parameter estimates (i.e., the parameter estimates are precision weighted), and extreme values are "shrunk" toward the mean (for more details on shrinkage, see Raudenbush & Bryk, 2002; Snijders & Bosker, 2012). Further, continuous predictors do not pose a problem for mixed-effects models (see Baayen, 2010), and the fitted model provides coefficient estimates that indicate the magnitude and direction of the effects of interest. Finally, the mixed-effects regression framework can easily be extended to handle a variety of response variables (e.g., categorical outcomes) via *generalized* linear mixed-effects models, and operating in this framework makes the transition to Bayesian modeling easier, as reliance on ANOVAs tends to create a fixed mind-set in which statistical testing and categorical "significant versus nonsignificant" thinking are paramount. Mixed-effects modeling is therefore appropriate in many cases in which standard ANOVAs, repeated measures ANOVAs, and multiple regression are not. Thus, it is a more flexible analytic tool.

## Disclosures

The data and R script used to generate the models described in this article are available via OSF, at https://osf.io/v6qag/.

## Introducing the Data

In this Tutorial, I use examples from my own research area, human speech perception, but the concepts apply to a wide variety of areas within and beyond psychology. For example, participants in a social-psychology experiment might view videos and be asked to evaluate the affect associated with each of them, or participants in a clinical experiment might read a series of narratives and be asked to describe the extent to which each of them generates anxiety.[1] The goal of this Tutorial is to provide a practical introduction to linear mixed-effects modeling and introduce the tools that will enable you to conduct such analyses on your own. This overview is not intended to address every issue you may encounter in your own analyses, but is meant to provide enough information that you have a sense of what to ask if you get stuck. To help you along the way, I provide snippets of R code using dummy data that serve as a running example.

The example data I provide (see https://osf.io/v6qag/), which we will work with later in this Tutorial, come from a within-subjects speech-perception study in which each of 53 participants was presented with 553 isolated words, some in the auditory modality alone (audio-only condition) and some with an accompanying video of the talker (audiovisual condition). Participants listened to and repeated these isolated words aloud while simultaneously performing an unrelated response time task in the tactile modality (classifying the length of pulses that coincided with the presentation of each word as short, medium, or long). The response time data are based on data from a previous experiment of mine (Brown & Strand, 2019; complete data set available at https://osf.io/86zdp/), but the response times themselves have been modified for pedagogical purposes (i.e., to illustrate particular issues that you may encounter when analyzing data with mixed-effects models). The accuracy data have not been modified, but variables have been removed for simplicity.

Previous research has shown that being able to see as well as hear a talker in a noisy environment substantially improves listeners' ability to identify speech relative to

**Table 1.** First Six Rows of the Example Data Set in Unaggregated and Aggregated Formats

| Unaggregated data set | | | | Aggregated data set | | |
|---|---|---|---|---|---|---|
| PID | modality | stim | RT | PID | modality | RT |
| 301 | Audio-only | gown | 1024 | 301 | Audio-only | 1027 |
| 301 | Audio-only | might | 838 | 301 | Audiovisual | 1002 |
| 301 | Audio-only | fern | 1060 | 302 | Audio-only | 1047 |
| 301 | Audio-only | vane | 882 | 302 | Audiovisual | 1043 |
| 301 | Audio-only | pup | 971 | 303 | Audio-only | 883 |
| 301 | Audio-only | rise | 1064 | 303 | Audiovisual | 938 |

Note: PID = participant identification number; stim = stimulus; RT = response time.

hearing the talker alone (e.g., Erber, 1972). The goals of this dual-task experiment were to determine whether seeing the talker would also affect response times in the secondary task (slower response times were taken as an indication of increased cognitive costs associated with the listening task—"listening effort") and to replicate the well-known intelligibility benefit from seeing the talker. In what follows, we will use mixed-effects modeling to assess the effect of modality (audio-only vs. audiovisual) on response times and word intelligibility while simultaneously modeling variability both within and across participants and items. We will assume that modality was manipulated within subjects and within items, which means that each participant completed the task in both modalities, and each word was presented in both modalities (but each word occurred in only one modality for each participant).

For all the analyses described below, we will use a dummy-coding (also referred to as treatment-coding) scheme such that the audio-only condition serves as the reference level and is therefore coded as 0, and the audiovisual condition is coded as 1. Thus, in the mixed-effects models, the regression coefficient associated with the intercept represents the estimated mean response time in the audio-only condition (when modality = 0), and the coefficient associated with the effect of modality indicates how the mean response time changes in the audiovisual condition (when modality = 1). We could instead use the audiovisual condition as the reference level, in which case the intercept would represent the estimated mean response time in the audiovisual condition (when modality = 0), and the modality effect would indicate how this estimate changes in the audio-only condition (when modality = 1). Altering the coding scheme, either by changing the reference level or by switching to a different coding scheme altogether (e.g., sum or deviation coding, which involves coding the groups as −0.5 and 0.5 or −1 and 1 so the intercept corresponds to the grand mean) will not change the fit of the model; it will simply change the interpretation of the regression coefficients (but often leads to misinterpretation of interactions, as I discuss below; see Wendorf, 2004, for a description of various coding schemes).

The left side of Table 1 shows the first six lines of the data in the desired format: *unaggregated long format*. For a helpful tutorial on how to wrangle the data into this format, I recommend Wickham and Grolemund's (2017) open-access textbook *R For Data Science* (Chapter 12.3.1) and the *tidyverse* collection of R packages (Wickham et al., 2019). If you are following along with your own data, ensure that your data are in long format such that each row represents an individual observation (i.e., do not aggregate across either participants or items). Notice that in this half of the table, each of the first six rows corresponds to a different word (stim) presented to the same participant (PID). In contrast, for an ANOVA, the data frame would contain two rows per participant, one for each modality, and the value in the response time (RT) column for a given row would reflect the mean response time for all words presented to that individual in the indicated condition (right side of Table 1).

## Fixed and Random Effects

Mixed-effects models are called "mixed" because they simultaneously model *fixed* and *random* effects. Fixed effects represent population-level (i.e., average) effects that should persist across experiments. Condition effects are typically fixed effects because they are expected to operate in predictable ways across various samples of participants and items. Indeed, in our example, modality will be modeled as a fixed effect because we expect that there is an average relationship between modality and response times that will turn up again if we conduct the same experiment with a different sample of participants and items.

Whereas fixed effects model average trends, random effects model the extent to which these trends vary across levels of some grouping factor (e.g., participants or items). Random effects are clusters of dependent data points in which the component observations come from

the same higher-level group (e.g., an individual participant or item) and are included in mixed-effects models to account for the fact that the behavior of particular participants or items may differ from the average trend. Given that random effects are discrete units sampled from some population, they are inherently categorical (Winter, 2019). Thus, if you are wondering if an effect should be modeled as fixed or random and it is continuous in nature, be aware that it cannot be modeled as a random effect and therefore must be considered a fixed effect. In our hypothetical experiment, participants and words are modeled as random effects because they are randomly sampled from their respective populations, and we want to account for variability within those populations.

Including random effects for participants and items resolves the nonindependence problem that often plagues multiple regression by accounting for the fact that some participants respond more quickly than others, and some items are responded to more quickly than others. These random deviations from the mean response time are called *random intercepts*. For example, the model may estimate that the mean response time for some condition is 1,000 ms, but specifying by-participant random intercepts allows the model to estimate each participant's deviation from this fixed estimate of the mean response time. So if one participant tended to respond particularly quickly, that person's individual intercept might be shifted down 150 ms (i.e., the estimated intercept would be 850 ms). Similarly, including by-item random intercepts enables the model to estimate each item's deviation from the fixed intercept, reflecting the fact that some words tend to be responded to more quickly than others. In multiple regression, in contrast, the same regression line (both intercept and slope) is applied to all participants and items, so predictions tend to be less accurate than in mixed-effects regression, and residual error tends to be larger. Thus, in mixed modeling, the fixed-intercept estimate represents the average intercept, and random intercepts allow each participant and item to deviate from this average.[2] These deviations are assumed to follow a normal distribution with a mean of zero and a variance that is estimated by the model.

An additional source of variability that mixed-effects models can account for comes from the fact that a variable that is modeled as a fixed effect may actually have different influences on different participants (or items). In our example, some participants may show very small differences in response times between the audio-only and audiovisual conditions, and others may show large differences. Similarly, some words may be more affected by modality than others. To model this type of variability, we will include *random slopes* in the model specification. In our hypothetical study, the model may estimate that the effect of modality is 83 ms—meaning that participants were, on average, 83 ms slower in the audiovisual condition than the audio-only condition—but one participant may have been very strongly affected by modality (e.g., a response time difference between modalities of 200 ms), and another may have been only weakly affected by modality (e.g., a response time difference between modalities of 10 ms). These individual deviations from the average modality effect are modeled via random slopes (note that a simple mean difference like the one described here is represented in a regression equation as a slope).

It may be confusing that modality comes up in the context of fixed and random effects, but recall that an effect is considered fixed if it is assumed that the effect would persist in a different sample of participants. In our case, modality is modeled as a fixed effect because we are modeling the common influence of modality on response times across participants and items. However, given that participants represent a random sample from the population of interest, the effect of modality within participants represents a subset of possible ways modality and participants can interact. In other words, modality itself is not a random effect, but the way it interacts with participants is random, and including random slopes for modality allows the model to estimate each participant's deviation from the overall (fixed) trend. (For more on the distinction between fixed and random effects and a description of when a researcher may actually want to model participants as a fixed effect, see Mirman, 2014).

One question people often have at this point is, if mixed-effects models derive an intercept and slope estimate for each participant, why are these seemingly systematic effects called *random* effects? The answer is that although an effect might be consistent within a particular individual (e.g., one participant may systematically respond more quickly and be less affected by modality than average), the source of this variability is unknown and is therefore considered random. If you find yourself stumbling over the use of the word *random*, it may be helpful to instead consider the synonymous terminology *by-participant* (or *-item*) *varying intercepts* (or *slopes*). However, given that these effects are most commonly referred to as random intercepts and slopes, I use that terminology here.

## Visualizing Random Intercepts and Slopes

In this section, I provide plots to help you visualize what happens when one builds on ordinary regression by introducing random intercepts and random slopes. These plots are derived from fake data from four hypothetical participants who each responded to four items (note that random effects really should have at least five or six levels, and having more levels is preferable; e.g.,

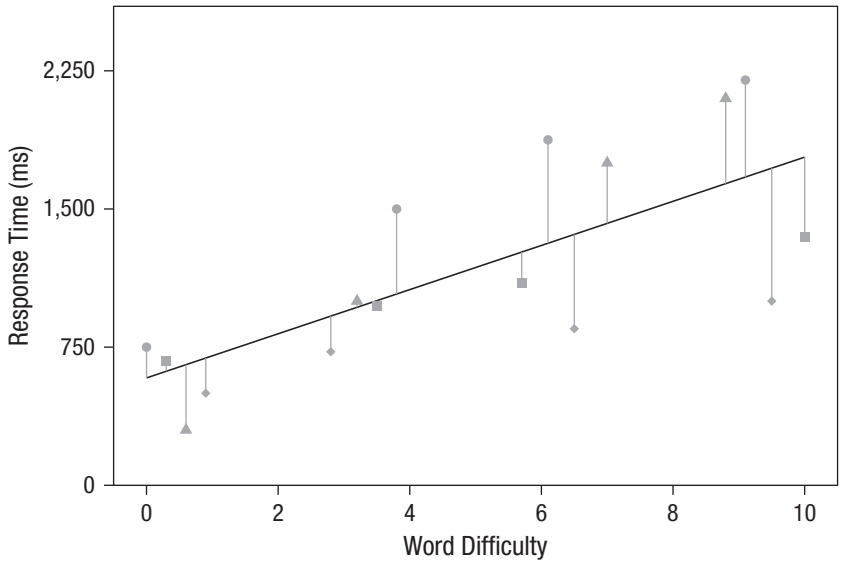


**Fig. 1.** Fixed-effects-only regression line depicting the relationship between word difficulty and response time. The plotted points represent individual response times for each word for each participant, and the vertical lines represent the deviation of each point from the line of best fit (i.e., residual error). Note that although you can discern the nested nature of the data from this plot because each participant's data are represented by a different shape, the model does not take such dependencies in the data into account. For visualization purposes, data from the four participants for each word have been jittered horizontally to avoid overlap.

Bolker, 2020). The effect of interest is the influence of word difficulty on response times (where 0 represents "very easy" by some collection of criteria, such as the frequency with which the word occurs in the language and the number of similar-sounding words, and 10 represents "very difficult"). First, consider a model with no random effects (i.e., fixed-effects-only regression; Fig. 1). More difficult words tend to elicit slower response times, but because there are no random effects, the model estimates are the same for every participant; that is, although you can tell which points in the plot correspond to each participant because I have represented data from each participant with a different shape, the model does not have access to this information. Further, given that this model predicts just one regression line that applies to all observations, the residual error (represented by vertical lines connecting every point to the regression line) is relatively large.

Next, consider a model that includes random intercepts for participants. In Figure 2, each dashed gray line depicts model predictions for a single participant, and the solid black line depicts the estimates for the average (fixed) effects. This model takes into account the fact that some participants tend to have slower response times than others. Here, the overall effect of word difficulty on response times is still apparent, but this model does a better job predicting response times for a given participant because it allows for each participant to have a different intercept (representing the predicted response time for a word with a 0 on the difficulty scale). In this example, the relationship between word difficulty and response time is equally strong for all participants (i.e., the slope is fixed); random intercepts simply shift each participant's regression line up or down depending on that individual's deviation from the mean (see Winter, 2019, p. 237, for another way of visualizing random intercepts, via a histogram of each individual's deviation from the population intercept). Notice that the residual error, indicated by the vertical lines, is substantially smaller in the random-intercepts model relative to the fixed-effects-only model. Indeed, the residual standard deviation in the original model is 410 ms, but it is reduced to 275 ms in the random-intercepts model (these values are obtained via the `summary()` command in R). This is because we have considered the fact that each participant's intercept can vary from the average intercept, so residual error represents deviation from a specific participant's regression line rather than the overall regression line.

Figure 3 shows how the model changes when by-participant random slopes are included; this model allows for the relationship between word difficulty and response time to vary across participants. Here, participants differ not only in how quickly they respond when word difficulty is 0 (random intercepts), but also in the extent to which they are affected by changes in word difficulty (random slopes). Although the general trend that difficult words are responded to more slowly is still

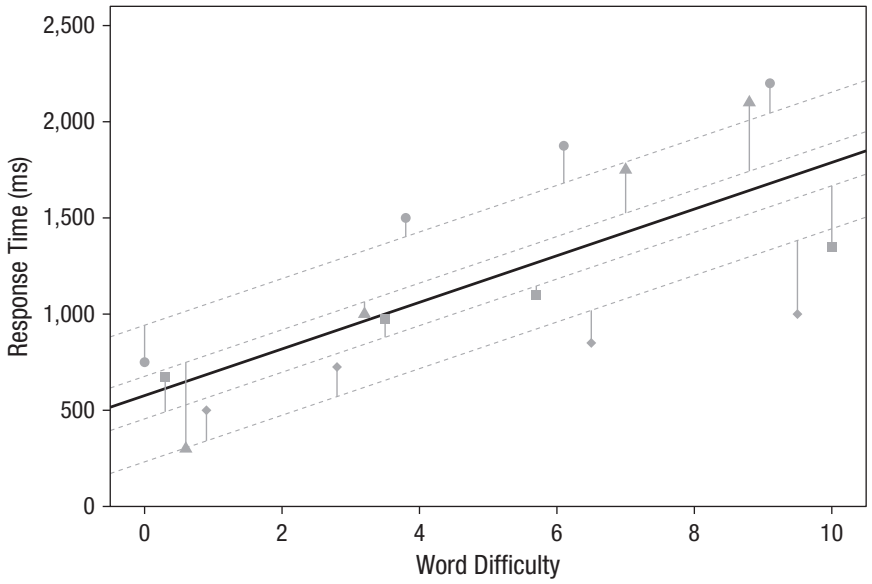


**Fig. 2.** Mixed-effects regression lines depicting the relationship between word difficulty and response time, generated from a model including by-participant random intercepts but no random slopes. Each dashed gray line represents model predictions for a single participant, and the solid black line represents the fixed-effects estimates for the intercept and slope. The plotted points represent individual response times for each word for each participant, and the vertical lines represent the deviation of each point from the participant's individual regression line. Notice that including random intercepts reduces residual error relative to the error in the fixed-effects-only model (Fig. 1). For visualization purposes, data from the four participants for each word have been jittered horizontally to avoid overlap.

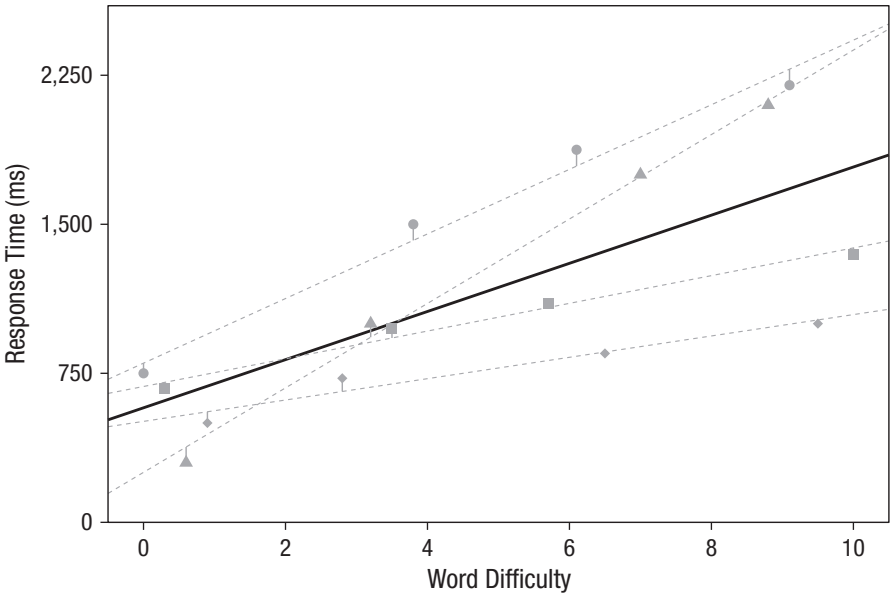


**Fig. 3.** Mixed-effects regression lines depicting the relationship between word difficulty and response time, generated from a model including by-participant random intercepts as well as by-participant random slopes for word difficulty. Each dashed gray line represents model predictions for a single participant, and the solid black line represents the fixed-effects estimates for the intercept and slope. The plotted points represent individual response times for each word for each participant, and the vertical lines represent the deviation of each point from the participant's individual regression line. Notice that including random slopes reduces residual error relative to the error in both the random-intercepts model and the fixed-effects-only model (Figs. 1 and 2). For visualization purposes, data from the four participants for each word have been jittered horizontally to avoid overlap.

apparent, the strength of this relationship varies across participants. The result is that the residual error is even smaller because each regression line is tailored to the individual; indeed, the residual standard deviation has decreased from 275 ms in the random-intercepts model to 75 ms in the random-slopes model. Note that for simplicity, these plots do not take item-level variability into account. (See Barr et al., 2013, for a helpful visualization depicting the simultaneous influences of participant and item random effects.)

## Correlations Among Random Effects

The discussion of mixed-effects models thus far has focused on fixed effects, random intercepts, and random slopes, but the models estimate additional parameters that are often overlooked: correlations among random effects. For example, when you specify that a model should include by-participant random intercepts and slopes for modality, the model will also estimate the correlations among those random intercepts and slopes. Although experimental psychology typically focuses on fixed effects, correlations among random effects can provide useful information about individual differences in condition effects.

Suppose, for example, that in our hypothetical dual-task experiment, the correlation between the by-participant random intercepts and slopes was negative (e.g., $r = -.17$). This would suggest that individuals who have higher intercepts (i.e., slower response times in the audio-only condition) tend to have lower slopes. Interpreting what "lower" means in the context of our experiment also requires knowledge of the direction of the modality effect. If the modality effect is positive, then "lower slopes" means slopes that are less positive (i.e., closer to zero), and the correlation therefore suggests that individuals with slower response times are less affected by the modality manipulation. If, however, the modality effect is negative, then "lower slopes" means slopes that are more negative, which suggests that individuals with slower response times tend to be more affected by the condition manipulation.

If the modality effect is 83 ms, a negative correlation between by-participant random intercepts and slopes would indicate that individuals who had slower response times in the audio-only condition tended to show a less pronounced slowing in the audiovisual condition. One interpretation of this correlation is that people who respond more slowly are completing the task more carefully, and this slow, deliberate responding washes out the condition effect in those individuals. Although this correlation is not of particular interest in this experiment, there are situations in which correlations among random effects are key to the research question. For example, a researcher conducting a longitudinal study might be interested in whether students' baseline mathematical abilities are related to the trajectory of their improvement over the course of a training program, so the correlation between by-participant random intercepts and slopes for the training effect would be of particular interest.

Another reason for examining correlations among random effects is that they can be informative about possible ceiling or floor effects. Consider the intelligibility effect I described when I introduced the data: Seeing the talker improves speech intelligibility. A negative correlation between by-participant random intercepts and slopes for modality in this case would indicate that individuals with higher intercepts (i.e., better speech identification in the audio-only condition) had shallower slopes (i.e., benefited less from seeing the talker)—a correlation that would also emerge if the speech-identification task was too easy. That is, if participants could attain a high level of performance without seeing the talker, then seeing the talker would have little effect on performance; this would result in a negative correlation between by-participant random intercepts and slopes, but only because ceiling effects prevented the modality effect from emerging in some individuals (see Winter, 2019, p. 239, for another example of random-effects correlations that may indicate the presence of ceiling effects). Thus, even if your research question primarily concerns fixed effects, examining random effects and their correlations will help you understand your data more deeply.

## Which Random Effects Can You Include?

Before we move on to implementation in R, it is important to note one other issue regarding random-effects structures in mixed-effects modeling: deciding which random slopes are justified by your design. Consider again the example in which we modeled response times to words as a function of their difficulty. Word difficulty was manipulated within subjects, but because the words differed on an intrinsic property—namely, their difficulty—word difficulty was a between-items variable. Given that by-item random slopes account for variability across items in the extent to which they are affected by the predictor of interest, we cannot model the effect of word difficulty on a particular item because each word has only one level of difficulty; that is, we cannot include by-item random slopes.[3] In contrast, if the predictor of interest was a within-items variable (as in our running example in which all words appeared in both the audio-only and the audiovisual conditions), we could include by-item random slopes for that predictor in our model, which would account for the fact that different words may be differently affected by the predictor. Put simply, by-participant and by-item slopes are justified only for within-subjects and within-items designs, respectively. Thus, our random-effects structure in the word-difficulty

example can include random intercepts for both participants and items, as well as by-participant random slopes for word difficulty, but cannot include by-item random slopes for word difficulty.

Because including by-item random slopes for word difficulty would not be justified in this example, the random-effects structure including random intercepts for both participants and items, as well as by-participant random slopes for word difficulty, would represent the maximal random-effects structure justified by the design (see Barr et al., 2013; Matuschek et al., 2017). In cases in which by-participant and by-item random slopes are justified, mixed-effects models can incorporate the simultaneous influences of both participant and item random slopes (but note that just because you can include a random effect does not necessarily mean that it would be advisable to do so. I discuss this further in the Model Building and Convergence Issues section).

## Examples and Implementation in R

Now that you have a conceptual understanding of what mixed-effects models are and why they are useful, let us consider how to implement them in R. First, you will need to install R (R Core Team, 2020) and then RStudio (RStudio Team, 2020), a programming environment that allows you to write R code, run it, and view graphs and data frames all in one place. I suggest working in RStudio rather than R (though this is not a rule, and some people code in the R console without RStudio). Base R (the set of tools that is built into R) has a host of functions, but to create mixed-effects models you will need to install a specific package called *lme4* (Bates et al., 2020). Packages, also referred to as libraries, are sets of functions that work together and are not already built into Base R. To install *lme4*, run the following line of code (you should run this line of code only if you have not already installed the package):

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

Once the package is installed, it is always on your computer, and you will not need to run that line of code again. Whenever you want to create mixed-effects models, you will need to load the installed package, which will give you access to all the functions you need (you need to rerun this line of code every time you start a new R session). The following line of code will load the *lme4* package:

```
> library(lme4)
```

In this section, I assume rudimentary knowledge of R. If you are new to R, I recommend installing and loading the *swirl* package (Kross et al., 2020), which serves as an introduction to R that can be completed in R itself.

### *Analyzing data with a continuous outcome (response time)*

Now we can start building some models. For these examples, which I conducted in R Version 4.0.3 with *lme4* Version 1.1-26, I used the hypothetical data set introduced earlier to assess whether seeing the talker affects response times to a secondary task and word intelligibility. I used a dummy-coding scheme with the audio-only condition as the reference level. To follow along, go to https://osf.io/v6qag/ and navigate to the R Markdown[4] file called "intro_to_lmer.Rmd."

***Model building and convergence issues.*** The basic syntax for mixed-effects modeling for an experiment with one independent variable and random intercepts but no random slopes for (crossed)[5] participants and items is

```
> lmer(outcome ~ 1 + predictor +
  (1|participant) + (1|item), data = data)
```

The portions in the interior sets of parentheses are the random effects, and the portions not in these parentheses are the fixed effects. The vertical lines within the random-effects portions of the code are called pipes, and they indicate that within each set of parentheses, the effects to the left of the pipe vary by the grouping factor to the right of the pipe. Thus, in this example, the intercept (indicated by the `1`) varies by the two grouping factors in this experiment: participants and items. Note that the `1` is optional in the fixed-effects portion of the model specification because the fixed intercept is included by default, but it is not optional in the random-effects portions because there must be some indication about which effects are allowed to vary by each grouping factor (i.e., the region to the left of the pipe cannot be left blank). I recommend always labeling intercepts with a `1` in both the fixed- and the random-effects portions of the model specification to avoid any confusion about when the `1` must be included. Finally, the `data` argument indicates the name of the R object containing the data, and the `lmer` part is the function that builds a mixed-effects model (which you can access because you installed the *lme4* package).

The model thus far includes random intercepts but no random slopes. However, my experience in speech-perception research leads me to expect that both participants and words might differ in the extent to which they are affected by the modality manipulation. We will therefore fit a model that includes both by-participant and by-item random slopes for modality. Failing to include random slopes would amount to assuming that all participants and words respond to the modality effect in exactly the same way, which is an unreasonable assumption to make. Although the model including by-participant and by-item random intercepts and slopes reflects the maximal random-effects structure justified

by the design, the decision to include by-participant and by-item random slopes is also theoretically justified. Theoretical motivation should always be considered, as blind maximization can lead to nonconverging models and a loss of statistical power (Matuschek et al., 2017). Notice how the basic syntax for the model changes when we include by-participant and by-item varying slopes in the random-effects structure:

```
> lmer(outcome ~ 1 + predictor +
  (1 + predictor|participant) + (1 +
  predictor|item), data = data)
```

Here, the portions in parentheses indicate that both the intercept (indicated by the `1`, which in this case is optional because it is implied by the presence of random slopes but is included for clarity) and the predictor (indicated by `+ predictor`) vary by participants and items. In plain language, this syntax means "predict the outcome from the predictor and the random intercepts and slopes for participants and items, using the data I provide."[6]

The model above includes only one predictor, but if a model includes multiple predictors the researcher may decide which of the predictors can vary by participant or item; in other words, any fixed effect to the left of the interior parentheses can be included to the left of the pipe (inside the interior parentheses), provided that including it is justified given the design of the experiment. For example, if we wanted to include a second predictor that varied within both participants and items, but there was no theoretical motivation for including by-item random slopes for the second predictor—or, alternatively, if the second predictor varied between items, so including the by-item random slope would not be justified given the experimental design—the syntax would look like this:

```
> lmer(outcome ~ 1 + predictor1 +
  predictor2 + (1 + predictor1 +
  predictor2|participant) + (1 +
  predictor1|item), data = data)
```

In the example we will be working with, the full model (i.e., the model including the fixed effects of interest and all theoretically motivated random effects) is specified as follows:

```
> rt_full.mod <- lmer(RT ~ 1 +
  modality + (1 + modality|PID) +
  (1 + modality|stim), data = rt_data)
```

Here, we are predicting response times (`RT`) on the basis of the fixed effects for the intercept (`1`) and modality (audio-only vs. audiovisual condition), we are including random intercepts and slopes for both participants (`PID` = participant identification number) and words (`stim` = stimulus), and we are telling R to use the data frame called `rt_data`.[7] Also note that this line of code includes the `<-` operator. This is used to assign a name to an object (a data structure with specific attributes that is stored in R's memory) and save it for later. Thus, with this line of code we have created a model and given it an intuitive name so that we know what that object represents later on.

If you run this line of code in the R script, you may notice that you get a warning message saying that the model failed to converge. Linear mixed-effects models can be computationally complex, especially when they have rich random-effects structures, and failure to converge basically means that a good fit for the data could not be found within a reasonable number of iterations of attempting to estimate model parameters. It is important never to report the results of a nonconverging model, as the convergence warnings are an indication that the model has not been reliably estimated and therefore cannot be trusted.

When a model fails to converge, you as the researcher have several options, and this is a situation potentially introducing *researcher degrees of freedom*—the numerous seemingly innocuous choices made during the research process that enable researchers to find "'statistically significant' evidence consistent with *any* hypothesis" (Simmons et al., 2011, p. 1359). As a general rule, you should consider which random effects are theoretically important to include in your model beforehand, using knowledge of your particular domain and previous research (e.g., ask yourself the question, "Does it make sense for modality to vary by participants or by items?"), and remove random effects only if all other ways of addressing convergence issues have been unsuccessful. If you must remove a random effect, this decision should be documented and reported in your published manuscripts and/or accompanying code.

The first step you should take to address convergence issues is to consider your data set and how your model relates to it, and to ensure that your model has not been misspecified (e.g., have you included by-item random slopes for a predictor that does not actually vary within items?). It is also possible that the convergence warnings stem from imbalanced data: If you have some participants or items with only a few observations, the model may encounter difficulty estimating random slopes, and those participants or items may need to be removed to enable model convergence. Although attempting to resolve convergence issues can feel like a hassle, keep in mind that these warnings serve as a friendly reminder to think deeply about your data and not model with your eyes closed. Assuming you have done this, the next step is to add *control parameters* to your model, so that you can tinker with the nuts and bolts of estimation. There

are many control parameters, and depending on the source of the convergence issues, some may be more appropriate or useful than others. The one I recommend starting with is adjusting the optimizer (i.e., the method by which the model finds an optimal solution). The model specification below is identical to the one above, with the exception that it includes a control parameter that explicitly specifies the optimizer:

```
> rt_full.mod <- lmer(RT ~ 1 + modality +
  (1 + modality|PID) + (1 +
  modality|stim), data = rt_data,
  control = lmerControl(optimizer =
  "bobyqa"))
```

This model converges, but how did I know which optimizer to choose? And what if the model had not successfully converged with that optimizer? When it comes to selecting an optimizer, I highly recommend the `all_fit()` function from the *afex* package (Singmann et al., 2020). This function takes a model as input, refits the model with a variety of optimizers, and lets you know which ones produce warning messages. This package integrates nicely with *lme4*, so the model syntax need not be changed before running the function. Here is the relevant code and abbreviated output:

```
> all_fit(rt_full.mod)
bobyqa. : [OK]
Nelder_Mead. : [OK]
optimx.nlminb : [OK]
optimx.L-BFGS-B : [OK]
nloptwrap.NLOPT_LN_NELDERMEAD : [OK]
nloptwrap.NLOPT_LN_BOBYQA : [OK]
nmkbw. : [OK]
```

This output indicates that none of the optimizers tested led to convergence warnings or singular fits, both of which are indicative of problems with estimation. Thus, any of these optimizers should produce reliable parameter estimates.

In this example, our model converged when we changed the optimizer, but this will not always be the case, and you may sometimes need to address convergence issues in another way.[8] One option is to force the correlations among random effects to be zero. Recall that in addition to estimating fixed and random effects, mixed-effects models estimate correlations among random effects. If you are willing to accept that a correlation may be zero,[9] this will reduce the computational complexity of the model and may allow the model to converge on parameter estimates. Note, however, that it is advisable to conduct likelihood-ratio tests (described in detail in the next subsection) on nested models differing in the presence of the correlation parameter—or examine the confidence interval around the correlation—to determine whether elimination is warranted. To remove a correlation between two random effects in R, simply put a `0` where the `1` was in the random-effects specification. When you do this, however, the `lmer()` function no longer estimates the random intercept, so you need to be sure to put it back into the model specification. Here is what the code would look like if you wanted to remove the correlation between the random intercept for participants and the by-participant random slope for modality:

```
> rt_full.mod <- lmer(RT ~ 1 + modality +
  (0 + modality|PID) + (1|PID) + (1 +
  modality|stim), data = rt_data)
```

Other ways to resolve convergence warnings include increasing the number of iterations before the model "gives up" on finding a solution (e.g., `control = lmerControl(optCtr = list(maxfun = 1e9))`), centering or scaling continuous predictors (or sum-coding categorical predictors), or removing some of the derivative calculations that occur after the model has reached a solution using the following control parameter: `control = lmerControl(calc.derivs = FALSE)`. I also suggest typing `?convergence` into the R console, which will open a help file offering other recommendations for resolving convergence warnings.

Finally, it may be that a model fails to converge simply because the random-effects structure is too complex (Bates, Kliegl, et al., 2015). In this case, one can selectively remove random effects based on model selection techniques (Matuschek et al., 2017). It is important to reiterate, however, that simplification of the random-effects structure should only be done as a last resort, and these decisions should be documented—the random-effects structure should be theoretically motivated, so it is best to try to maintain that structure unless all other methods of addressing convergence issues are unsuccessful.

***Likelihood-ratio tests.*** Now that we have a model to work with, how do we determine if modality actually affected response times? This is typically done by comparing a model including the effect of interest (e.g., modality) with a model lacking that effect (i.e., a *nested* model) using a *likelihood-ratio test*.[10] This test is used to compare two nested models by calculating likelihoods for the two models using a technique called maximum likelihood estimation and then statistically comparing those likelihoods. If you obtain a small *p* value from the likelihood-ratio test, this indicates that the full model provides a better fit for the data.

When we run a likelihood-ratio test for our example, we are basically asking, does a model that includes information about the modality in which words are presented fit the data better than a model that does not include that information? Here is how you do this in R,

first by building the reduced model that lacks the fixed effect for modality but is otherwise identical to the full model (including any control parameters used), and then by conducting the test via the `anova()`[11] function (which does not actually compute an analysis of variance, but is a convenient function for conducting a likelihood-ratio test):

```
> rt_reduced.mod <- lmer(RT ~ 1 + (1 +
  modality|PID) + (1 + modality|stim),
  data = rt_data, control =
  lmerControl(optimizer = "bobyqa"))
> anova(rt_reduced.mod, rt_full.mod)
Data: rt_data
Models:
rt_reduced.mod: RT ~ 1 + (1 + modality|stim)
  + (1 + modality|PID)
rt_full.mod: RT ~ 1 + modality + (1 +
  modality|stim) + (1 + modality|PID)
        Df  AIC    BIC    logLik deviance Chisq Chi Df Pr(>Chisq)
rt_reduced
  .mod    8 302449 302513 -151217 302433
rt_full
  .mod    9 302419 302491 -151200 302401   32.385  1  1.264e-08
```

The small *p* value in the `Pr(>Chisq)` column indicates that the model including the modality effect provides a better fit for the data than the model without it; thus, the modality effect is significant. I have added boldface for the *p* value, the $\chi^2$ value in the `Chisq` column (`32.385`), and the degrees of freedom for the test (`1`, found in the `Chisq Df` column) because these three values should be reported in your results section (I return to this point below).

Given that the full model includes only one condition effect (modality), conducting the test is relatively straightforward. However, performing likelihood-ratio tests can quickly become a tedious task for complex models with many fixed effects. This is because these tests must be conducted on nested models, so in order to test a particular effect, a reduced model (sometimes referred to as a *null model*) lacking that effect needs to be built for comparison. Another issue with this approach is that although the reduced models are built solely for the purpose of comparison with the full model, it can be quite tempting to examine those intermediate models and consider them plausible candidates for the "best model" (i.e., perform stepwise regression without knowing it). For example, suppose you build a full model with two fixed effects—modality and background-noise level—and a reduced model to test whether the effect of modality is significant. In doing so, you may notice that noise level is significant in the reduced model but not in the full model and convince yourself that a model without modality is actually more appropriate, even though you had not considered this possibility before examining the models. This is a questionable research practice (John et al., 2012) known as hypothesizing after the results are known (HARKing) and should be avoided because, as Kerr (1998) put it, HARKing transforms Type I error (false positives) into theory.

Luckily, the *afex* package has another handy function that allows you to avoid this practice altogether. The `mixed()` function takes a model specification as input and conducts likelihood-ratio tests on all fixed (but not random) effects in the model when the argument `method = 'LRT'` is included. Crucially, you do not see the reduced models that were built to obtain the relevant *p* values, so the temptation to inadvertently *p*-hack is reduced. This function is more useful when your model has multiple fixed effects, but here is how to implement the function in our example and what the output looks like (notice that the $\chi^2$ value is the same as when we used the `anova()` function, because both functions conduct likelihood-ratio tests):

```
> mixed(RT ~ 1 + modality + (1 +
  modality|PID) + (1 + modality|stim),
  data = rt_data, control =
  lmerControl(optimizer = "bobyqa"),
  method = 'LRT')

Model: RT ~ 1 + modality + (1 +
  modality|stim) + (1 + modality|PID)
Data: rt_data
Df full model: 9
   Effect   df      Chisq    p.value
1 modality  1    32.39***     <.001
```

***Interpreting fixed and random effects.*** The likelihood-ratio test comparing our full and reduced models indicated that the modality effect was significant, but it did not tell us about the direction or magnitude of the effect. So how do we assess whether the audiovisual condition resulted in slower or faster response times? And how do we gain insight into the variability across participants and items that we asked the model to estimate? To answer these questions, we need to examine the model output via the `summary()` command. The output contains two main sections: The top part contains information about random effects, and the bottom part contains information about fixed effects. The following code chunk implements the `summary()` command and shows the abbreviated output relevant to interpreting fixed effects:

```
> summary(rt_full.mod)
Fixed effects:
            Estimate Std. Error df   t value
(Intercept) 1044.14   23.36    52.14  44.704
modality      83.18   12.58    52.10   6.615
```

Recall that we used a dummy-coding scheme with the audio-only condition as the reference level; the intercept therefore represents the estimated mean response time

in the audio-only condition, and the modality effect represents the adjustment to the intercept in the audiovisual condition. Thus, response times in the audio-only condition averaged an estimated 1,044 ms, and response times were an estimated 83 ms slower in the audiovisual condition.

Now let us focus on the random-effects portion of the output:

```
Random effects:
Groups   Name        Variance  Std.Dev.  Corr
stim   (Intercept)    303.9     17.43
       modality       216.6     14.72    0.16
PID    (Intercept)  28552.7    168.98
       modality      7709.8     87.81   -0.17
Residual            65258.8    255.46
```

The `Groups` column lists the grouping factors that appeared to the right of the pipes in the model specification (along with the residuals), and the `Name` column lists the effects that were grouped by each factor (i.e., the intercepts and modality slopes that appeared to the left of the pipes in the model specification). Each of these random intercepts and slopes has an associated variance (and standard deviation) estimate, which tells you the extent to which response times for particular stimuli and participants varied around the fixed intercept and slope. For example, the standard deviation for by-item random intercepts (in boldface in the output above) indicates that response times for particular items varied around the average intercept of 1,044 ms by about 17 ms. Similarly, the standard deviation for by-participant random slopes (in boldface in the output above) indicates that participants' estimated slopes varied around the average slope of 83 ms by about 88 ms. Thus, an individual whose slope was 1 *SD* below the mean would have an estimated slope near 0 (indicating that this person's response times were not affected by the modality in which the words were presented), whereas an individual whose slope was 1 *SD* above the mean would have a very steep slope (indicating a difference between modalities of about 171 ms). The `coef()` function in *lme4* provides individual intercept and slope estimates for every participant and item, which not only helps make the concept of random-intercept and -slope estimates more concrete, but can also help you identify outliers. Here is the code and abbreviated output indicating estimates for the first four items and participants:

```
> coef(rt_full.mod)
$stim
      (Intercept)    modality
babe     1038.921    82.11521
back     1050.914    86.52633
bad      1041.122    81.12267
bag      1042.896    86.40601

$PID
      (Intercept)     modality
301     1024.0668   -16.936415
302     1044.1377     1.842626
303      882.8306    57.789321
304     1232.7544   -27.919775
```

This output indicates that the estimated intercept for the word "bag" is 1,043 ms, and the estimated slope is 86 ms; these values are very similar to the estimates for the fixed intercept (1,044 ms) and slope (83 ms). The participant part of the output indicates that Participant 303 had an estimated intercept of 883 ms and an estimated slope of 58 ms, indicating that this person responded much more quickly than average and was less affected by modality than average. Notice that even though we are looking at estimates for only four items and participants, it is clear that there is more intercept and slope variability across participants than across items. The standard deviations are consistent with this observation. Specifically, the standard deviations for the by-participant random intercepts (169 ms) and slopes (88 ms) are much larger than those for the by-item random intercepts (17 ms) and slopes (15 ms). This is not surprising—in my experience, participants tend to vary more than items—but it is useful to know that participants vary considerably in their response times because this could have important consequences for power calculations and could uncover avenues for individual differences research (e.g., why do people vary so much in the way the modality manipulation affects their response times?) and follow-up studies (e.g., do the results hold when one controls for individual differences in simple reaction time?).

Although the focus of our hypothetical study is on fixed effects, random-effects estimates can be interesting and informative in their own right, and in some cases provide insight into the key research question. For example, Idemaru and colleagues (2020) recently concluded that loudness is a more informative cue than pitch in predicting whether an utterance is perceived as respectful or not respectful. This claim was supported both by greater variation in pitch than loudness slopes across participants (i.e., participants responded more consistently to loudness cues) and by the fact that the direction of the loudness effect was negative for every single participant (this is an example of the `coef()` function in action), but the direction of the pitch effect varied considerably across participants. Thus, random effects rather than fixed effects were at the crux of the authors' argument that listeners use loudness as an indicator of respect more consistently than they use pitch.

The last piece of information in the random-effects output concerns correlations among random effects. The `Corr` column indicates that the correlation between

random intercepts for items and by-item random slopes is .16, and the correlation between random intercepts for participants and by-participant random slopes is −.17 (these values have been put in boldface in the output above). This means that items that were responded to more slowly in the audio-only condition tended to have larger (more positive, steeper) slopes, and participants who responded more slowly in the audio-only condition had shallower slopes. One possible explanation for the positive correlation for items is that the items that were responded to more slowly tended to be the more difficult ones and may have been particularly affected by any distraction coming from the visual signal. The negative correlation between by-participant random intercepts and slopes is consistent with the one described earlier in this article and may suggest that slow, deliberate responding washes out the modality effect.

Finally, it is important to note that it is possible for a model to encounter estimation issues (i.e., produce unreliable parameter estimates) without any warning messages appearing in aggressive red text in your R console, and the random-effects portion of the output contains some clues that may help you identify when this happens. One clue comes from the random-effects correlations, which are set to −1.00 or 1.00 by default when they cannot be estimated, and another comes from the variance estimates, which are set to 0 when they cannot be estimated (i.e., the variance and correlation parameters are set to their boundary values when they cannot be estimated; Bates, Mächler, et al., 2015). Although random-effects correlations of −1.00 or 1.00 are often accompanied by "singular fit" warning messages, this is not always the case, so it is crucial to examine the random-effects portion of the model output to ensure that estimation went smoothly.

***Reporting results in a manuscript.*** There are no explicit rules for reporting findings from model comparisons and the associated parameter estimates from the preferred model (Meteyard & Davies, 2020). How results are reported depends on the number and nature of model comparisons, the journal submission guidelines, and author and reviewer preferences. That said, I typically report the $\chi^2$ value from the likelihood-ratio test, the degrees of freedom of the test, and the associated *p* value, as well as the coefficient estimates, *t* values, and standard errors associated with the parameters of interest from the selected model. To report the findings described in the example above, you could write,

> A likelihood-ratio test indicated that the model including modality provided a better fit for the data than a model without it, $\chi^2(1) = 32.39$, $p < .001$. Examination of the summary output for the full model indicated that response times were on average an estimated 83 ms slower in the audiovisual relative to the audio-only condition($\hat{\beta} = 83.18$, $SE = 12.58$, $t = 6.62$).

As long as you report your results transparently and include details of the model specification and any simplifications you made to the random-effects structure in your manuscript or accompanying code, the particular convention you follow is up to you (and, of course, making your data and code publicly available reduces the impact of the reporting convention you adopt). Finally, you should be sure to cite R as well as the specific packages you used to conduct your analyses, including the versions you used, both to facilitate reproducibility of your results (indeed, it is not uncommon for a model that once converged to no longer converge with an *lme4* update) and to give credit to the package developers who have put a lot of work into making your analyses possible.

***Interpreting interactions.*** The data set we have been working with throughout this Tutorial contains just one condition effect. Although this simplicity is convenient for learning about mixed-effects models, many experiments test multiple conditions and the interactions among them. Interpreting interactions is tricky, and doing so accurately depends critically on knowledge of the coding scheme used for categorical predictors. R's default (and usually my own) is to use dummy coding, which leads to misinterpretation of interactions and lower-order effects if sum coding is assumed to be the default. Therefore, for this section, I continue to use dummy-coded predictors. The example I provide uses the same data set we have been working with, but contains one additional categorical predictor representing the difficulty of the background-noise level. Participants identified speech in audio-only (coded 0) and audiovisual (coded 1) conditions in both an easy (coded 0) and a hard (coded 1) level of background noise. The goal of this analysis is to assess whether the effect of modality on response time depends on (i.e., interacts with) the level of the background noise (i.e., the signal-to-noise ratio, or SNR). On the basis of previous research, we expect that response times will be slower in the audiovisual condition (as in the analyses above), but that this slowing will be more pronounced in easy listening conditions because the cognitive costs associated with simultaneously processing auditory and visual information are amplified in conditions in which seeing the talker is unnecessary to attain a high level of performance (see Brown & Strand, 2019).

There are a few ways to specify an interaction in R that produce identical results. One way is to use an asterisk (`modality*SNR`), which automatically includes all lower-order terms even if you do not type them in (the following syntax is abbreviated for readability, but random effects and control parameters are also included; see the accompanying code at https://osf.io/v6qag/):

```
> rt_int.mod <- lmer(RT ~ 1 + modality*
  SNR + . . ., data = rt_data_interaction)
```

Another way to specify an interaction is to use a colon rather than an asterisk (`modality:SNR`), but in this case, you need to explicitly specify the lower-order terms in the model specification (I use this method for clarity):

```
> rt_int.mod <- lmer(RT ~ 1 + modality
  + SNR + modality:SNR + . . .,
data = rt_data_interaction)
```

Here is the abbreviated model output:

```
Fixed effects:
             Estimate  Std. Error     df   t value
(Intercept)  998.824     22.214      52.729 44.964
modality      98.510     13.199      59.065  7.464
SNR           92.339     14.790      58.004  6.243
modality:SNR -29.532      6.755   21298.850 -4.372
```

Recall that the intercept represents the estimated response time when all other predictors are set to 0. The intercept of 999 ms therefore represents the estimated mean response time in the audio-only modality (modality = 0) in the easy listening condition (SNR = 0). If you are having difficulty understanding why this is the case, it may be helpful to plug in 0 for both modality and SNR in the following regression equation. Notice that the intercept is the only term that does not drop out:

$$\begin{aligned} RT &= 999 + 99 * modality + 92 * SNR - 30 * modality * SNR \\ &= 999 + 99 * 0 + 92 * 0 - 30 * 0 * 0 \\ &= 999 \end{aligned}$$

To interpret the remaining three coefficients, it is important to note that when an interaction is included in a model, it no longer makes sense to interpret the predictors that make up the interaction in isolation. This means that the coefficient for the modality term should not be interpreted as the average modality effect if SNR is held constant (this would be the interpretation if we had not included an interaction in the model), because the presence of the interaction tells us that the modality effect changes depending on the SNR. Instead, the coefficient for the modality term should be interpreted as the estimated change in response time from the audio-only to the audiovisual condition when all other predictors are set to 0. Thus, the modality effect indicates that response times are on average 99 ms slower in the audiovisual relative to the audio-only condition in the easy listening condition (SNR = 0). Think of it this way: When the SNR dummy code is set to 0 (easy), the SNR and interaction terms drop out of the model, and we are left with a 99-ms adjustment to the intercept when we move from the audio-only to the audiovisual condition. However, when the SNR dummy code is set to 1 (hard), those terms do not drop out of the model, and it is no longer accurate to say that the modality effect is 99 ms (again, plugging 0s and 1s into the regression equation above may help you here).

Similarly, the SNR effect indicates that response times are on average 92 ms slower in the hard relative to the easy listening condition, but this applies only when the modality dummy code is set to 0 (representing the audio-only condition). The modality and SNR effects I have just described are called *simple effects*, but are often misinterpreted as *main effects*. Simple effects represent the effect of a predictor on an outcome at a particular level of another predictor, whereas main effects represent the average effect of a predictor on an outcome across levels of another predictor. Thus, when an interaction is present and you have used a coding scheme centered on 0 (e.g., sum coding), lower-order effects are considered main effects, but if you have used a dummy-coding scheme, they are simple effects. Keep this common misinterpretation in mind any time you use dummy coding.

Just as the modality and SNR effects can be thought of as adjustments to the intercept in particular conditions (e.g., estimates are shifted up 99 ms in the audiovisual relative to the audio-only condition, but only in the easy listening condition), the interaction term can be thought of as an adjustment to the modality or SNR slope when both predictors are set to 1 (note that interactions adjust coefficient estimates only for a single cell of the design because the interaction term drops out when one or both of the predictors are set to 0). In this example, the coefficient for the modality term indicates that the modality effect is 99 ms when the SNR is easy, but the presence of an interaction tells us that the effect of modality differs depending on the level of the background noise; that is, the modality slope needs to be adjusted when the SNR is hard. Specifically, the negative interaction term indicates that the modality slope is 30 ms lower (less steep) when the SNR is hard, which is consistent with the hypothesis I described above: Seeing the talker slows response times, but it does so to a greater extent when the listening conditions are easy, presumably because the visual signal is distracting and unnecessary when the auditory signal is highly intelligible. Note that interactions are symmetric in that if the modality slope varies by SNR, then the SNR slope varies by modality. You can therefore also interpret the interaction term as an adjustment to the SNR slope: The 92-ms SNR effect is 30 ms weaker in the audiovisual condition. If you are struggling to interpret interactions with dummy-coded predictors, I recommend making a table containing the coefficient estimate for each of the cells in the design by plugging all combinations of 0s and 1s into the regression equation (Table 2); this can help you visualize the role of each individual coefficient estimate in generating cell-wise predictions (see also Winter, 2019).

**Table 2.** Estimates for All Cells in the 2 × 2 Design When the Model Includes an Interaction Term

| | Modality | |
|---|---|---|
| Signal-to-noise ratio | Audio-only condition (0) | Audiovisual condition (1) |
| Easy (0) | 999 | 999 + 99 |
| Hard (1) | 999 + 92 | 999 + 99 + 92 − 30 |

## *Analyzing data with a binary outcome (identification accuracy)*

Now you should have a general understanding about how to build and interpret models in which the outcome is continuous (e.g., response time), but what if you wanted to test for an effect of modality on accuracy at identifying words, when accuracy for each trial is scored as 0 or 1? These values are discrete and bounded by 0 and 1, so you need to use *generalized* linear mixed-effects models; if we instead modeled this discrete outcome assuming a continuous outcome, the model would generate impossible predictions (e.g., a predicted probability of −0.2 or 1.3). The R code for building these kinds of models is almost exactly the same as that described above, except rather than using the `lmer()` function you use the `glmer()` (*g*eneralized *l*inear *m*ixed-*e*ffects *r*egression) function, and you need to include at least one additional argument within the `glmer()` function indicating the assumed distribution of the dependent variable. The `glmer()` function also contains an argument for specifying a link function, which transforms the outcome into a continuous and unbounded scale, but each family of distributions has a default link function that typically does not need to be changed. In this case, the discrete outcomes of 0 and 1 follow a binomial distribution, which should be modeled with logistic regression, typically using a *logit link function* (the default). The logit link function transforms probabilities, which are bounded by 0 and 1, into a continuous, unbounded scale (log odds). Using the logit link function allows us to model the linear relationship between the predictors and the log odds of the outcome (which can be transformed back into odds and probabilities for ease of interpretation) without generating nonsensical predictions.

Put simply, the logit link function first transforms probabilities, which are bounded by 0 and 1, into odds, which are bounded by 0 and infinity (a probability of 0 corresponds to odds of 0, and a probability of 1 corresponds to odds of infinity). However, this scale still has a lower bound of 0, so the link function takes the natural logarithm of the odds (the logarithm of 0 is negative infinity, so the lower bound of the scale is extended from 0 to negative infinity), which results in the continuous and unbounded log-odds scale. Using this function means that any predictions generated from the model will also be on a log-odds scale, which is not particularly informative, but luckily, these predictions can be exponentiated to put them back on an odds scale, and the odds can then be converted into probabilities (see Jaeger, 2008, for a tutorial on using logit mixed models).

Here is the code to build the full model:

```
> acc_full.mod <- glmer(acc ~ 1 +
  modality + (1 + modality|PID) +
  (1 + modality|stim), data = acc_data,
  family = binomial)
```

This code is very similar to that for the response time analysis, but it contains a few key differences. First, the dependent variable is `acc` (0 for incorrect and 1 for correct word identification) rather than `RT`. Because this outcome is binomially distributed, we indicate that we are using generalized linear mixed-effects modeling by using the `glmer()` function, and we indicate that our dependent variable follows a binomial distribution with the additional parameter `family = binomial`.

This model converged, but remember that you should always examine the random-effects portion of the output to ensure that estimation went smoothly:

```
> summary(acc_full.mod)
Random effects:
Groups   Name        Variance Std.Dev. Corr
stim   (Intercept) 0.72085  0.8490
        modality    0.46663  0.6831  -0.06
PID    (Intercept) 0.04346  0.2085
        modality    0.04903  0.2214  -0.15
```

Not only did we not encounter any convergence or singularity warnings, but the variance estimates and estimated correlations among random effects seem reasonable (i.e., the variance estimates are not exactly zero, and the correlations are not −1.00 or 1.00). It is slightly unusual that in this data set there is more variability across items than across participants in both intercepts and slopes, but this may simply reflect the fact that the speech-identification task was relatively easy for most participants, which resulted in little variability.[12]

Next, we will build a reduced model lacking modality as a fixed effect so we can conduct a likelihood-ratio test:

```
> acc_reduced.mod <- glmer(acc ~ 1 +
  (1 + modality|PID) + (1 +
  modality|stim), data = acc_data,
  family = binomial)
```

It is important to note that although both the full and reduced models converged with this random-effects structure and no control parameters, it is certainly possible (and indeed not uncommon) for the full model to converge but the reduced model to encounter convergence issues. In this case, you should find a random-effects structure and combination of control parameters

**Table 3.** Helpful Links and Additional Resources

| Helpful links | Additional resources |
|---|---|
| R (R Core Team, 2020) for Macs: https://cran.r-project.org/bin/macosx/ | An excellent introduction to linear models and mixed-effects modeling for individuals with limited statistical experience: Winter (2013) |
| R (R Core Team, 2020) for Windows: https://cran.r-project.org/bin/windows/base/ | An introduction to analyzing eye-tracking data with mixed-effects modeling: Barr (2008) |
| RStudio (RStudio Team, 2020): https://www.rstudio.com/products/rstudio/download/ | An argument in favor of utilizing the maximal random-effects structure justified by the design (within reason): Barr et al. (2013) |
| *swirl* package for learning R in R (Kross et al., 2020): https://swirlstats.com/ | An argument in favor of using parsimonious mixed models: Bates et al. (2015) |
| Wickham and Grolemund's (2017) *R for Data Science* book: https://r4ds.had.co.nz/index.html | A model-selection approach to selecting random-effects structures: Matuschek et al. (2017) |
| Franke and Roettger's (2019) *brms* tutorial: https://psyarxiv.com/cdxv3 | Descriptions of mixed models with crossed random effects for participants and items: Baayen et al. (2008), Quené and van den Bergh (2008) |
| | Overviews of design types and statistical power for analyzing data with mixed-effects models: Judd et al. (2017), Westfall et al. (2014) |
| | Description of logit mixed models: Jaeger (2008) |
| | Descriptions of how to extend mixed-effects modeling to growth-curve analysis: Mirman et al. (2008), Mirman (2014) |
| | Introduction to modeling other nonlinear effects (e.g., linguistic change) and implementing general additive modeling: Winter and Wieling (2016) |

that enable both models to converge (e.g., via the `all_fit()` function in the *afex* package), because the models being compared via a likelihood-ratio test should be nested and built with the same control parameters. That is, the models should be identical except for the presence of the fixed effect of interest. Here is the code and output for the likelihood-ratio test:

```
> anova(acc_reduced.mod, acc_full.mod)
Data: acc_data
Models:
acc_reduced.mod: acc ~ 1 + (1 +
  modality | PID) + (1 + modality | stim)
acc_full.mod: acc ~ 1 + modality +
  (1 + modality | PID) + (1 + modality | stim)
             npar   AIC   BIC   logLik deviance Chisq  Df Pr(>Chisq)
acc_reduced
.mod            7   28147 28205 -14067   28133
acc_full
.mod            8   27989 28055 -13986   27973   160.78 1  < 2.2e-16
```

The small *p* value indicates that the full model provides a better fit for the data than the reduced model, and thus that modality has a significant effect on spoken-word identification accuracy.

## Conclusions

Mixed-effects modeling is becoming an increasingly popular method of analyzing data from experiments in which each participant responds to multiple items—and for good reason. The beauty of mixed-effects models is that they can simultaneously model participant and item variability while being far more flexible and powerful than other commonly used statistical techniques: They handle missing observations well, they can seamlessly include continuous predictors, they provide estimates for average (as well as by-participant and by-item) effects of predictors on the outcome, and they can be easily extended to model categorical outcomes.

However, as Uncle Ben once said to Spider-Man, with great power comes great responsibility (Lee & Ditko, 1962). These models can be easily implemented in R without cost, but it is important that researchers ensure that this powerful tool is used correctly. Indeed, although more and more researchers are implementing mixed-effects models, there is a concerning lack of standards guiding implementation and reporting of these models (Meteyard & Davies, 2020). Many analytic decisions must be made when using this statistical technique. Consider, for example, the number of options available to the researcher if a model fails to converge. This results in a massive number of "forking paths" (Gelman & Loken, 2014) that the researcher may embark upon to obtain statistically significant results. Given the considerable number of choices a researcher may make during data analysis (i.e., researcher degrees of freedom; Simmons et al., 2011), it is important that these models be used carefully and reported transparently (see Meteyard & Davies, 2020, for an example of how models and results should be reported).

The goal of this article is to serve as an accessible, broad overview of mixed-effects modeling for researchers with minimal experience with this type of modeling. I have focused on what mixed-effects models are, what they offer over other analytic techniques, and how to implement them in R. Table 3 lists helpful links, as well as additional resources for readers interested in more in-depth descriptions of particular topics.

## Transparency

*Action Editor:* Mijke Rhemtulla
*Editor:* Daniel J. Simons
*Author Contributions*

V. A. Brown is the sole author of this article and is responsible for its content. She devised the idea for the article, wrote the article in its entirety, wrote the accompanying R script, and generated the dummy data on which the models are based.

*Declaration of Conflicting Interests*

The author(s) declared that there were no conflicts of interest with respect to the authorship or the publication of this article.

*Funding*

This work was supported by the National Science Foundation through a Graduate Research Fellowship awarded to V. A. Brown (DGE-1745038).

*Open Practices*

Open Data: https://osf.io/v6qag/
Open Materials: not applicable
Preregistration: not applicable

All data have been made publicly available via OSF and can be accessed at https://osf.io/v6qag/. This article has received the badge for Open Data. More information about the Open Practices badges can be found at http://www.psychologicalscience.org/publications/badges.

## ORCID iD

Violet A. Brown https://orcid.org/0000-0001-5310-6499

## Acknowledgments

I am grateful to Michael Strube for providing detailed feedback on an earlier draft of the manuscript and to Julia Strand and Kristin Van Engen for providing helpful comments and suggestions throughout the writing process.

## Notes

1. It is important to note that the examples in this Tutorial concern *crossed* rather than *nested* random effects (mixed-effects models with nested random-effects structures are typically referred to as *hierarchical linear models*). Random effects (defined later) of participants and items are considered crossed when every participant responds to every item and nested when every participant responds to a different set of items. The classic example of nested random effects comes from education research in which students are nested within classes, which in turn are nested within schools (see Raudenbush, 1988). The motivation for using mixed modeling applies to both design types, but the examples and R code I provide assume a crossed design (see Baayen et al., 2008; Judd et al., 2017; Quené & van den Bergh, 2008; Westfall et al., 2014, for more on the distinction between crossed and nested designs).
2. Note that this is not literally how parameters in mixed-effects models are estimated. Those details are beyond the scope of this Tutorial, and this simplified description is provided to help you conceptualize what mixed models are doing behind the scenes. See Snijders and Bosker (2012) for more detail.
3. Note that including by-item random slopes might be unjustified even when the conditions are not defined by stimulus-intrinsic properties. For example, if you are interested in the effect of background noise on response times to words, but different words are assigned to different conditions (each word appears in only one level of noise), it would not be justified to include by-item random slopes. It is therefore crucial to consider your experimental design before building mixed-effects models.
4. R Markdown is a file format that is easily accessed via RStudio and incorporates plain text, code, and R output.
5. The corresponding code for a nested random-effects structure in which classes are nested within schools is

```
> lmer(outcome ~ 1 + predictor + (1|school/
  class), data = data)
```

6. When you are creating a mixed-effects model like this one, R uses maximum likelihood estimation to compute the values of the parameters that maximize the likelihood of the data given the structure that you specify for the model (see Etz, 2018, for an approachable introduction to the concept of likelihood).
7. Response time data should really be analyzed with *generalized* linear mixed-effects models (discussed in the section on analyzing binomial data) assuming, for example, an inverse Gaussian distribution and an identity link function because response times tend to be positively skewed (Lo & Andrews, 2015). For simplicity, however, we will use *general* linear mixed models via the `lmer()` function; the parameter estimates change a bit with generalized mixed modeling, but the conclusions do not change. Mixed modeling is quite robust to violations of the normality assumption, so it is acceptable to use general mixed models here.
8. Note that convergence issues are far less common in Bayesian mixed models than in frequentist mixed models (*lme4* falls into the frequentist category), so if you find yourself struggling with convergence issues, you might consider switching to a Bayesian framework. For individuals who are comfortable using *lme4*, this switch is made easy by the *brms* package (Bürkner, 2017) because this package uses *lme4* formula syntax but Bayesian statistics behind the scenes. This Tutorial uses *lme4* only, but interested readers may want to refer to Franke and Roettger's (2019) helpful tutorial on how to use *brms* (https://psyarxiv.com/cdxv3).
9. A situation in which you may not be willing to assume the correlation is zero is when that correlation is a crucial part of your research question. For example, if your research question addresses whether people with slower overall response times tend to be less affected by modality, then it would be critical to allow the model to estimate the correlation between the random effects. However, a typical study in experimental psychology is more interested in the fixed-effects parameter estimates, so assuming the correlation is zero is often acceptable. The code to examine the confidence interval around standard deviation and correlation estimates is `confint(rt_full.mod, parm = "theta_", oldNames = F)`. The `parm` parameter indicates which parameters in the model will be given confidence intervals, and setting the `oldNames` parameter to `FALSE` (`F`) simply gives the output more interpretable names.
10. If you examine the summary output for a mixed-effects model, you may notice that the `lmer()` function does not include *p* values. This is because the null distribution is unknown (the error structure in multilevel models is complex, and the degrees of freedom cannot be calculated). Bates, one of the creators of the *lme4* package and the person who wrote the `lmer()` function, has posted a helpful description of why he did not include

*p* values in that function (see Bates, 2006). You can obtain *p* values by loading the *lmerTest* package (Kuznetsova et al., 2017), but I recommend using likelihood-ratio tests instead.

11. Note that if you are testing only random effects, you should include the argument `refit = FALSE` in the `anova()` command. This is because *lme4* automatically refits the models using maximum likelihood (ML) estimation when you conduct a likelihood-ratio test via the `anova()` command, but this is necessary only when testing fixed effects. This default is in place to make it really difficult to test fixed effects when the models have been built using the default estimation procedure in *lme4* (restricted maximum likelihood estimation, or REML), as this method is not appropriate for comparing models differing in fixed effects. However, you should override the default by including `refit = FALSE` if you are testing random effects.

12. The experiment on which these data are based also included an SNR manipulation whereby each word in the data set occurred in both audio-only and audiovisual conditions at both a very easy and a moderate SNR. I have ignored the SNR variable for simplicity, but the relatively easy SNRs in which the words were presented may explain why accuracy was high for all participants.

## References


Baayen, R. H. (2010). A real experiment is a factorial experiment. *The Mental Lexicon*, *5*(1), 149–157.

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

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

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, *68*(3), 255–278. https://doi.org/10.1016/j.jml.2012.11.001

Bates, D. (2006). *[R] lmer, p-values, and all that*. R-help. https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html

Bates, D., Kliegl, R., Vasishth, S., & Baayen, H. (2015). *Parsimonious mixed models*. arXiv. http://arxiv.org/abs/1506.04967

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1). https://doi.org/10.18637/jss.v067.i01

Bates, D., Maechler, M., Bolker, B., Walker, S., Christensen, R., Singmann, H., Dai, B., Scheipl, F., Grothendieck, G., & Green, P. (2020). *Package 'lme4'* (Version 1.1-26) [Computer software]. Comprehensive R Archive Network. https://cran.r-project.org/web/packages/lme4/lme4.pdf

Bolker, B. (2020). *GLMM FAQ*. GitHub. http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#should-i-treat-factor-xxx-as-fixed-or-random

Brown, V. A., & Strand, J. F. (2019). About face: Seeing the talker improves spoken word recognition but increases listening effort. *Journal of Cognition*, *2*(1). Article 44. https://doi.org/10.5334/joc.89

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1). https://doi.org/10.18637/jss.v080.i01

Enders, C. K. (2010). *Applied missing data analysis*. Guilford Press.

Erber, N. P. (1972). Auditory, visual, and auditory-visual recognition of consonants by children with normal and impaired hearing. *Journal of Speech and Hearing Research*, *15*(2), 413–422.

Etz, A. (2018). Introduction to the concept of likelihood and its applications. *Advances in Methods and Practices in Psychological Science*, *1*(1), 60–69.

Franke, M., & Roettger, T. (2019). *Bayesian regression modeling (for factorial designs): A tutorial*. PsyArXiv. https://doi.org/10.31234/osf.io/cdxv3

Gelman, A., & Loken, E. (2014). The statistical crisis in science: Data-dependent analysis—a "garden of forking paths"—explains why many statistically significant comparisons don't hold up. *American Scientist*, *102*(6), 460–466.

Idemaru, K., Winter, B., Brown, L., & Oh, G. E. (2020). Loudness trumps pitch in politeness judgments: Evidence from Korean deferential speech. *Language and Speech*, *63*(1), 123–148.

Jaeger, T. F. (2008). Categorical data analysis: Away from ANOVAs (transformation or not) and towards logit mixed models. *Journal of Memory and Language*, *59*(4), 434–446.

John, L. K., Loewenstein, G., & Prelec, D. (2012). Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, *23*(5), 524–532.

Judd, C. M., Westfall, J., & Kenny, D. A. (2017). Experiments with more than one random factor: Designs, analytic models, and statistical power. *Annual Review of Psychology*, *68*, 601–625.

Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and Social Psychology Review*, *2*(3), 196–217.

Kross, S., Carchedi, N., Bauer, B., Grdina, G., Schouwenaars, F., & Wu, W. (2020). *Package 'swirl'* (Version 2.4.5) [Computer software]. Comprehensive R Archive Network. https://cran.r-project.org/web/packages/swirl/swirl.pdf

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13). https://doi.org/10.18637/jss.v082.i13

Lee, S., & Ditko, S. (1962). *Amazing Fantasy*, *1*(15) [Comic book]. Marvel.

Liben-Nowell, D., Strand, J., Sharp, A., Wexler, T., & Woods, K. (2019). The danger of testing by selecting controlled subsets, with applications to spoken-word recognition. *Journal of Cognition*, *2*(1). Article 2. https://doi.org/10.5334/joc.51

Lo, S., & Andrews, S. (2015). To transform or not to transform: Using generalized linear mixed models to analyse reaction time data. *Frontiers in Psychology*, *6*, Article 1171. https://doi.org/10.3389/fpsyg.2015.01171

Matuschek, H., Kliegl, R., Vasishth, S., Baayen, H., & Bates, D. (2017). Balancing Type I error and power in linear mixed models. *Journal of Memory and Language*, *94*, 305–315.

Meteyard, L., & Davies, R. A. I. (2020). Best practice guidance for linear mixed-effects models in psychological science. *Journal of Memory and Language*, *112*, Article 104092. https://doi.org/10.1016/j.jml.2020.104092

Mirman, D. (2014). *Growth curve analysis and visualization using R*. Chapman and Hall.

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

Quené, H., & van den Bergh, H. (2008). Examples of mixed-effects modeling with crossed random effects and with binomial data. *Journal of Memory and Language*, *59*(4), 413–425.

Raudenbush, S. W. (1988). Educational applications of hierarchical linear models: A review. *Journal of Educational and Behavioral Statistics*, *13*(2), 85–116.

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

R Core Team. (2020). *R: A language and environment for statistical computing* [Computer software]. R Foundation for Statistical Computing. http://www.R-project.org/

Royston, P., Altman, D. G., & Sauerbrei, W. (2005). Dichotomizing continuous predictors in multiple regression: A bad idea. *Statistics in Medicine*, *25*(1), 127–141.

RStudio Team. (2020). *RStudio: Integrated development environment for R* [Computer software]. RStudio, PBC. http://www.rstudio.com/

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366.

Singmann, H., Bolker, B., Westfall, J., & Aust, F. (2020). *afex: Analysis of factorial experiments* (Version 0.27.2) [Computer software]. GitHub. https://github.com/singmann/afex

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2nd ed.). SAGE Publications.

Wendorf, C. A. (2004). Primer on multiple regression coding: Common forms and the additional case of repeated contrasts. *Understanding Statistics*, *3*(1), 47–57.

Westfall, J., Kenny, D. A., & Judd, C. M. (2014). Statistical power and optimal design in experiments in which samples of participants respond to samples of stimuli. *Journal of Experimental Psychology. General*, *143*(5), 2020–2045.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., . . . Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), Article 1686. https://doi.org/10.21105/joss.01686

Wickham, H., & Grolemund, G. (2017). *R for data science: Import, tidy, transform, visualize, and model data* (1st ed.). O'Reilly Media.

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

Winter, B. (2019). *Statistics for linguists: An introduction using R*. Taylor & Francis.

Winter, B., & Wieling, M. (2016). How to analyze linguistic change using mixed models, growth curve analysis and generalized additive modeling. *Journal of Language Evolution*, *1*(1), 7–18.