A tutorial on fitting Bayesian linear mixed models using Stan

Tanner Sorensen
Department of Linguistics, University of Potsdam, Potsdam 14476, Germany

Shravan Vasishth
School of Mathematics and Statistics, University of Sheffield, UK
Department of Linguistics, University of Potsdam, Potsdam 14476, Germany

Abstract

With the arrival of the R packages nlme and lme4, linear mixed models (LMMs) have come to be widely used in psychology, cognitive science, and related areas. In this tutorial, we provide a practical introduction to fitting LMMs in a Bayesian framework using the probabilistic programming language Stan. Although the Bayesian framework has several important advantages, specifying a Bayesian model requires quite a lot of background knowledge compared to frequentist tools like lme4. This tutorial provides the necessary background through two detailed examples of self-paced reading studies with repeated measures. One is a two-condition design, and the other a $2 \times 2$ factorial design. These two examples can easily be extended to more complex factorial designs. The data and code associated with this tutorial are available as a supplement.

Keywords: Bayes’ theorem, linear mixed model, hierarchical model, inference, reading time
1. Introduction

Ever since the arrival of the nlme package (Pinheiro and Bates, 2000) and its successor, lme4 (Bates and Sarkar, 2007), the use of linear mixed models (and hierarchical models more generally) has increased dramatically in psychology and psycholinguistics. Until very recently, within the programming environment R, the function lmer in the package lme4 has been the tool of choice for fitting such models. The biggest advantage of lmer over other tools is simplicity: one can quickly fit (often in a single line of code) fairly complex models.

In recent years, probabilistic programming languages like WinBUGS (Lunn et al., 2000), JAGS (Plummer, 2012) and Stan (Stan Development Team, 2014) have made it possible to fit relatively complex Bayesian linear mixed models (LMMs) quite easily within R. One drawback of these programming languages is that a formidable amount of background knowledge is needed in order to fit Bayesian models.

In this tutorial, we provide researchers with a starter kit for fitting commonly used LMMs in a Bayesian setting. For simplicity, we develop models for two common designs: a two-condition repeated measures design, and a factorial $2 \times 2$ repeated measures design.

There are no prerequisites for this tutorial apart from having some exposure to fitting LMMs using lme4 and having the relevant software installed. For Bayesian modeling we will use rstan in R and any associated software; see the Stan website (mc-stan.org) website for details. The reader will benefit greatly from reading Gelman and Hill (2007), but this is optional. We provide more references at the end of the tutorial.

We begin by reviewing the inferential process in the frequentist and Bayesian setting. Then we enumerate several advantages of fitting Bayesian over frequentist LMMs. A review of the structure of the LMM follows; we use two published data-sets as running examples, fitting models using lme4, then fitting the corresponding models in Stan. We end by discussing how the model may be summarized, its quality of fit evaluated, and inferences made.

2. Brief review of frequentist versus Bayesian methodology

Imagine a dependent measure $Y$ for which we assume a probability model; $Y$ could be reading time in milliseconds. Let us suppose that $Y$ follows a normal distribution with some mean $\mu$ and variance $\sigma^2$; we write this as
\( Y \sim N(\mu, \sigma^2) \). Then, given parameter values \( \mu \) and \( \sigma^2 \), we can state the probability distribution of \( Y \) given the parameters as \( N(Y \mid \mu, \sigma^2) \). In reality, we know neither parameter \( \mu \) nor \( \sigma^2 \). The goal of fitting a model to a sample \( y = y_1, y_2, \ldots, y_n \) of \( n \) experimental observations of \( Y \) is to estimate \( \mu \) and \( \sigma^2 \) and to draw inferences about what the true values of \( \mu \) and \( \sigma^2 \) are. The frequentist method relies on the fact that, given a large enough sample size, the sampling distribution of the sample mean, \( \bar{Y} \), is distributed as \( N(\mu, \sigma^2/n) \). The standard method is to use the sample mean \( \bar{y} \) as the maximum likelihood estimate (MLE) of \( \mu \) and the sample standard deviation \( s \) as the MLE of \( \sigma \); see the MLE tutorial by Myung (2003). If \( n \) is large enough, we can then compute an approximate 95% confidence interval \( \bar{y} \pm 2 \times (s^2/n) \). The 95% confidence interval has a slightly complicated interpretation: if we were to repeatedly carry out the experiment and compute a confidence interval each time using the above procedure, 95% of those confidence intervals would contain the true parameter value \( \mu \) (as long as model assumptions are satisfied). The particular confidence interval we calculate using the sample \( y \) does not give us a range such that we are 95% certain that the true \( \mu \) lies within it, although this is how many users of statistics seem to (mis)interpret the confidence interval (Hoekstra et al., 2014).

By contrast, the Bayesian approach starts with a probability distribution that defines our prior belief about the possible values of the parameters \( \mu \) and \( \sigma^2 \). This prior probability distribution expresses what we currently know or believe about these two parameters. We may not know much, but in practical situations in psychology and cognitive science, it is not the case that we know nothing about their possible values. Given this prior distribution, the probability model \( N(Y \mid \mu, \sigma^2) \), and the data \( y \), we can compute the probability distribution of the parameters given the data, \( N(\mu, \sigma^2 \mid y) \). This probability distribution, called the posterior distribution, is what we use for inference. We can use the posterior to define a 95% credible interval which — unlike the 95% confidence interval — represents the range within which we are 95% certain that the true value of the parameter lies, given the data \( y \). We define this as the interval between the 2.5 and the 97.5 percentiles, but this credible interval is not associated with any hard cut-off \( \alpha \)-value that is conventional in the use of the frequentist confidence interval. We could just as well consider 90% credible intervals. Recall that in the frequentist setting, the parameters are point values: \( \mu \) is assumed to have a particular value in nature. In the Bayesian setting, \( \mu \) is a random variable with a probability distribution; it has a mean, but there is also some uncertainty associated with
its true value. The uncertainty is made explicit in the posterior probability distribution. Bayes’ theorem makes it possible to derive the posterior distribution of $\theta$ given the prior $p(\theta)$ and the data $y$. The conditional probability rule in probability theory (Kerns, 2010) is that the joint distribution $p(\theta, y)$ of $\theta, y$ is equal to $p(\theta \mid y)p(y)$. From this fact, we can derive the following:

$$p(\theta, y) = p(\theta \mid y)p(y) = p(y, \theta) \quad \text{(because } p(\theta, y) = p(y, \theta))$$

$$= p(y \mid \theta)p(\theta). \quad \text{(1)}$$

The first and third lines in the equalities above imply that

$$p(\theta \mid y)p(y) = p(y \mid \theta)p(\theta). \quad \text{(2)}$$

Dividing both sides by $p(y)$ we get:

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)} \quad \text{(3)}$$

The term $p(y \mid \theta)$ is the probability of the data given $\theta$. If we treat this as a function of $\theta$, we have the likelihood function. Since $p(\theta \mid y)$ is the posterior distribution of $\theta$ given $y$, and $p(y \mid \theta)$ the likelihood, and $p(\theta)$ the prior, the following relationship is established:

$$\text{Posterior } \propto \text{Likelihood } \times \text{Prior} \quad \text{(4)}$$

We ignore the denominator $p(y)$ here because it only serves as a normalizing constant that renders the left-hand side (the posterior) a probability distribution (it makes the total probability density $p(y)$ sum to 1). The relationship shown in 4 is Bayes’ theorem, and allows us to determine the posterior distribution given a prior and the likelihood. This is the basis for Bayesian probabilistic inference. In the example above of a normal distribution probability model of the dependent measure $Y$, $\theta$ would be $\langle \mu, \sigma \rangle$, and the posterior would be the joint probability distribution of $\mu$ and $\sigma$. Since the result of Bayesian modeling, the posterior, is a probability distribution, a Bayesian model thus allows inferences like that “the parameter $\mu$ is greater than zero with 95% certainty,” which is not possible in the frequentist setting. This seems like a much more informative basis for drawing inferences from
data than rejecting a null hypothesis.

3. Some advantages of fitting Bayesian linear mixed models

Apart from the obvious advantage of Bayesian modeling for purposes of inference, we see two major advantages of fitting Bayesian LMMs for psychology and related areas such as psycholinguistics. First, if we want our model to reflect the experimental design faithfully, we should fit full variance-covariance structures for the variance components (Barr et al., 2013; Gelman and Hill, 2007; Gelman et al., 2014). However, for the relatively small sample sizes used in psychology, fitting such “maximal” models is often not feasible. Modeling often produces estimates of parameters of correlation between random effects which are ±1 or NaN. In this case, either the correlated random effects are collinear (when the estimate of their correlation is ±1) or the correlation parameter is not identifiable (when the estimate is NaN). In either case, the data analysis would proceed by dropping parameters from the model. Bayesian LMMs provide a way to retain the maximal random effects variance-covariance structure. Appendix A shows that a Bayesian LMM makes fewer ±1 and NaN estimates than lmer does.¹ This facilitates fitting models with a full variance-covariance structure for random effects (i.e. “maximal” models) even when there are not enough data to make inference on the parameters of random effects correlation.

Second, reading time data in psychology is often analyzed using normal distribution theory; in a Bayesian LMM it is straightforward to define a more realistic distribution like the log-normal as the probability model for the data. As discussed in Appendix B, a log-normal model of rt provides greater sensitivity than a normal model of logrt in discovering effects if they really are present in nature.

We turn next to the application of Bayes’ theorem to fitting LMMs.

4. A linear mixed model with a full variance-covariance matrix by subject and by items

We will now motivate the LMM with a representative example of an experimental paradigm commonly used in psychology and related areas. Sub-

---

¹It is worth stressing that this is not a failure of the lmer function; the user is simply asking for too much from it.
ject and object relative clauses have been widely used in reading studies to investigate sentence comprehension processes. A typical finding for English is that subject relatives are (for reasons that don’t concern us here) easier to process than object relatives (Just and Carpenter, 1992). A subject relative is a sentence like *The senator who interrogated the journalist resigned* where a noun (here, *senator*) is modified by a relative clause (here, the clause *who...journalist*), and the modified noun is the grammatical subject of the relative clause. In an object relative, the noun modified by the relative clause is the grammatical object of the relative clause (example: *The senator who the journalist interrogated resigned*). In both cases, the noun that is modified (senator) is called the head noun.

All natural languages have relative clauses, and this so-called subject relative advantage has until recently been considered to be true cross-linguistically. However, Chinese relative clauses apparently represent an interesting counter-example to this generalization; recent work (Hsiao and Gibson, 2003) has suggested that in Chinese, *object* relatives are easier to process than subject relatives at a particular point in the sentence (the head noun of the relative clause). We now present an analysis of a published data-set (Gibson and Wu, 2013) that evaluates this claim. This experiment had two conditions (subject relative and object relative), with 37 subjects and 15 items, presented in a standard Latin square design. The original item sample size was 16, but one item was removed; in addition, most of the data from one subject is missing. These missing values result in a total of 547 data points. The first few lines from the data frame are shown below; “or” refers to object relative and “sr” to subject relative.

<table>
<thead>
<tr>
<th>subj</th>
<th>item</th>
<th>type</th>
<th>rt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>or</td>
<td>1561</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>sr</td>
<td>959</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>or</td>
<td>582</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>or</td>
<td>294</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>sr</td>
<td>438</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>sr</td>
<td>286</td>
</tr>
</tbody>
</table>

In order to carry out a statistical analysis, we must define a probability model that expresses our assumption about how the data were generated. The rest of this section motivates an LMM which defines such a model for reading time data. We make the working assumption that rt follows a log-
normal distribution (see Appendix B for a justification).

The logarithm of the reading times, \( \log rt \), has some unknown grand mean \( \beta_0 \). This mean can be expressed conditional on whether \( rt \) was measured in the subject or object relative condition. So the mean of the normal distribution which we draw \( \log rt \) from is the sum of \( \beta_0 \) and an adjustment \( \beta_1 \) whose magnitude depends on whether \( \log rt \) is from the subject or object relative condition.

\[
\log rt = \beta_0 + \beta_1 so + \varepsilon
\]

(5)

The categorical predictor \( so \) has a numerical value associated with each relative clause type. We code this categorical predictor using sum contrasts: we can give the value \(-1\) to subject relatives and \(+1\) to object relatives. With this coding, \( \beta_0 \) represents the grand mean of \( \log rt \) which does not condition on the predictor \( so \). The parameter \( \beta_1 \) is the adjustment which makes the mean of \( \log rt \) conditional on whether \( so \) is \(-1\) or \(+1\), i.e., conditional on whether a data point \( \log rt \) comes from the subject or object relative condition. So \( \log rt \) is drawn from a normal distribution with mean \( \beta_0 - 1\beta_1 \) when it comes from the subject relative condition and with mean \( \beta_0 + 1\beta_1 \) when it comes from the object relative condition. Together \( \beta_0 \) and \( \beta_1 \) make up the so-called fixed part of the model, which characterizes the effect of the experimental manipulation \( so \) on \( rt \).

For the above data, the estimates of the fixed effects \( \beta_0 \), and \( \beta_1 \), and their standard errors can in principle be computed using the \texttt{lm} function in R. If we fit such a model, our estimates for the Gibson and Wu data would be as shown below. The “hat” over each parameter indicates that this is an estimate.

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Est. Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\beta}_0 )</td>
<td>6.06</td>
<td>0.03</td>
</tr>
<tr>
<td>( \hat{\beta}_1 )</td>
<td>-0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 1: Estimated fixed effects and standard errors on the log ms scale.

The third term \( \varepsilon \) of model shown in equation 5 is the residual error: from model 5, it is clear that this is the amount by which the predicted values \( \beta_0 + \beta_1 so \) from the model differ from \( \log rt \). We assume here that \( \varepsilon \) is independently and identically distributed as a normal distribution with mean \( 0 \) and unknown variance \( \sigma^2 \); this is written as \( \varepsilon \sim N(0, \sigma^2) \). This variance is estimated from the data. It is a strong assumption that the error is normally
distributed. A consequence is that the distribution of $\varepsilon$ cannot have long or fat tails. Furthermore, there must be no correlation between observed groups of $\varepsilon$. In short, it is Gaussian white noise.

However, the fixed effects model 5 does not reflect that fact that $\log rt$ will vary systematically depending on the subject and on the item; some subjects will read fast, some slow, and similarly, some items will be read fast, some slow. Subjects and items therefore add variability in the response. As we did just above, we could in principle ignore these sources of variability (or variance components) and assume only one source of variance, $\sigma^2$. However, if we just ignore this structure among the subgroupings of $\log rt$ which are associated with a given subject or a given item, the distribution of $\varepsilon$ will no longer be Gaussian white noise, but rather “colored” by the different variance-covariance structures among the subgroupings of $\log rt$. The solution is to take all these variance components into account. We can do this by adding adjustment terms $u_{0j}$ and $w_{0k}$ which adjust $\beta_0$ for subject $j$ and item $k$. This partially decomposes $\varepsilon_i$ into a sum of the terms $u_{0j}$ and $w_{0k}$, which are an adjustment to the intercept $\beta_0$ for subject $j$ and item $k$. If subject $j$ is slower than the average of all the subjects, $u_{0j}$ would be some positive number, and if item $k$ is read faster than the average reading time of all the items, then $w_{0k}$ would be some negative number. These adjustments $u_{0j}$ and $w_{0k}$ are called random or varying intercepts, and by adjusting the $\beta_0$ by these we account for idiosyncratic patterns of speakers and items.

The random adjustments $u_{0j}$ and $w_{0k}$ are subtracted from $\varepsilon_i$ for those indices $i$ which are associated with subject $j$ and item $k$, respectively. The difference of two normally distributed random variables with mean 0 also has mean 0, so we accomplish this by making these adjustments normally distributed around 0 with unknown variances: $u_{0j} \sim N(0, \sigma_u^2)$ and $w_{0k} \sim N(0, \sigma_w^2)$. Note that now we have three variance components in this model: the variance of the errors $\sigma^2$, the variance of the by-subject random intercepts $\sigma_u^2$, and the variance of the by-item random intercepts $\sigma_w^2$. Note also that all three variance components are assumed to be independent of each other. We can now express the logarithm of reading time $i$ — which is associated with subject $j$ reading item $k$ — as the following sum.

$$\log rt_{ijk} = \beta_0 + u_{0j} + w_{0k} + \beta_1 s_{0i} + \varepsilon_i \quad (6)$$

This is an LMM, and more specifically a varying intercepts model. It can be fit in R using the \texttt{lmer} function available in the package \texttt{lme4}, using
the command shown below; so has the sum contrast coding described above, where subject relative is coded as \(-1\) and object relative is coded as \(+1\). Estimates of the variance components, the fixed effects coefficients, and the fixed effects standard errors computed by \texttt{lmer} are shown in Table 2.

\begin{verbatim}
library(lme4)
m1<-lmer(log(rt)~so+(1|subj)+(1|item),data)
\end{verbatim}

<table>
<thead>
<tr>
<th>Random effects</th>
<th>Variance component</th>
<th>Estimated Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\hat{\sigma}_u)</td>
<td>0.24</td>
<td></td>
</tr>
<tr>
<td>(\hat{\sigma}_w)</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>(\hat{\sigma}_e)</td>
<td>0.52</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>Estimate</th>
<th>Estimated Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\hat{\beta}_0)</td>
<td>6.06</td>
<td>0.07</td>
</tr>
<tr>
<td>(\hat{\beta}_1)</td>
<td>-0.04</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 2: The estimated parameters in the varying intercepts model for the Gibson and Wu (2013) data.

Suppose now that subjects who are fast readers exhibit greater slowdowns when they read the object relative compared to the subject relative. This pattern is shown in the data from Gibson and Wu (2013) (see the top panel of figure 1). Similarly, it is in principle possible that items that are read faster may show a greater slowdown in object relatives than subject relatives. This relationship between the different slopes and intercepts for each subject and item can be illustrated graphically by fitting a separate linear model for each subject (item), and plotting the relationship between each subject’s (item’s) slope and intercept. Figure 1 shows the result. These linear models use the same sum contrast coding as before, where subject relative is coded as \(-1\) and object relative is coded as \(+1\). The more negative the slope, the slower the subject reads the subject relatives. So figure 1 suggests that the slower a subject’s reading time is on average, the slower he reads the subject relatives. The by-item intercepts and slopes seem to be uncorrelated. It is important to note that we don’t have a lot of data here to conclude much about this intercept-slope relationship, and that these estimates of the
By subject intercepts and slopes

By item intercepts and slopes

Figure 1: The relationship between the different slopes and intercepts for each subject and item in the data-set of Gibson and Wu (2013). Each subject and item’s estimates are computed separately.
correlation between by-subject intercepts and slopes is based on unshrunken estimates of the intercepts and slopes (Gelman and Hill, 2007); but we ignore this issue for now.

In the LMM, the adjustment of the slope by subject and by item is expressed by adjusting \( \beta_1 \) by some quantities \( u_{1j} \) and \( w_{1k} \). These are varying slopes, and by adding them we account for the how the effect of relative clause type varies by subject \( j \) and by item \( k \). We now express \( \log rt_{ijk} \) as the following sum.

\[
\log rt_{ijk} = \beta_0 + u_{0j} + w_{0k} + (\beta_1 + u_{1j} + w_{1k})s_{0i} + \epsilon_i
\]  

(7)

The varying intercepts \( u_{0j} \) and \( w_{0k} \) are adjustments to the fixed intercept \( \beta_0 \), and the varying slopes \( u_{1j} \) and \( w_{1k} \) are adjustments to the fixed slope \( \beta_1 \). The LMM assumes that these adjustments are normally distributed with mean 0 and unknown variance. So we must estimate the variances of \( u_0 \), \( u_1 \), \( w_0 \), and \( w_1 \), the correlation between \( u_0 \) and \( u_1 \), and the correlation between \( w_0 \) and \( w_1 \). The standard deviations \( \sigma_{u0} \) and \( \sigma_{u1} \) of \( u_0 \) and \( u_1 \), respectively, and the parameter \( \rho_u \) of correlation between them are in the variance-covariance matrix \( \Sigma_u \).

\[
\Sigma_u = \begin{bmatrix}
\sigma_{u0}^2 & \rho_u \sigma_{u0}\sigma_{u1} \\
\rho_u \sigma_{u0}\sigma_{u1} & \sigma_{u1}^2
\end{bmatrix}
\]  

(8)

Likewise, the standard deviations \( \sigma_{w0} \) and \( \sigma_{w1} \) of \( w_0 \) and \( w_1 \), respectively, and the parameter \( \rho_w \) of correlation between them are in the variance-covariance matrix \( \Sigma_w \).

\[
\Sigma_w = \begin{bmatrix}
\sigma_{w0}^2 & \rho_w \sigma_{w0}\sigma_{w1} \\
\rho_w \sigma_{w0}\sigma_{w1} & \sigma_{w1}^2
\end{bmatrix}
\]  

(9)

As in the varying intercepts model 6, all the random adjustments by subject and item reduce the magnitude of the residual term \( \epsilon_i \) in model 7. Recall that the expected value of \( \epsilon \) was 0 in model 5 and that was still the case after subtracting the mean 0, normally distributed random intercepts \( u_{0j} \) and \( w_{1k} \). In model 7, \( \epsilon_i \) still has expected value of 0, so this constrains the expected value of the random slopes to likewise be 0. If the expected value of either of the random intercepts \( u_{0j}, w_{0k} \) or of the error term \( \epsilon_i \) were nonzero, we would add that quantity to \( \beta_0 \). Likewise, if the expected value of either \( u_{1j} \) or \( w_{1k} \) were nonzero, we would add that quantity to \( \beta_1 \) (Searle et al., 2009, p 11, 130).

To state all of this more formally, the LMM for this particular example can
be specified by assuming that the varying intercepts and slopes by subjects and by items have the following bivariate distribution:

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_u \right) \quad \begin{pmatrix} w_{0k} \\ w_{1k} \end{pmatrix} \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_w \right)$$

(10)

If we assume no correlation between $u_0$ and $u_1$, this would mean that we fix $\rho_u$ at 0. The joint distribution of $\langle u_0, u_1 \rangle$ is then an ellipse in the plane of all possible values for $\langle u_{0j}, u_{1j} \rangle$ which is symmetric about the $u_0$- and $u_1$-axis. If $\sigma_{u0}$ and $\sigma_{u1}$ happened to be .25 and .35, respectively, we could plot a large joint sample distribution of $\langle u_0, u_1 \rangle$ as in the middle row of figure 2. We see that knowing the value of the random intercept $u_{0j}$ for subject $j$ does not change the expected value of that subject’s random slope $u_{1j}$, although it does affect $\sigma_{u1}$. Figure 1 suggests, however, that $u_0$ covaries with $u_1$ in the sense that knowledge of the random intercept $u_{0j}$ for subject $j$ should change the expected value of that subject’s random slope $u_{1j}$ and vice versa. In particular, if $u_{0j}$ is large, then we would expect $u_{1j}$ to be large and negative, but if $u_{0j}$ is small, we would expect $u_{1j}$ to be more positive. The LMM cannot reflect these relationships when $\rho_u$ is fixed at 0. This consideration leads us to allow $\rho_u$ to vary over the interval $[-1, 1]$. Large simulated joint sample distributions for $\langle u_0, u_1 \rangle$ are plotted in the top and bottom rows of figure 2. We see that the joint distribution of $\langle u_0, u_1 \rangle$ is still an ellipse in the plane of all possible values for $\langle u_{0j}, u_{1j} \rangle$ when $\rho_u$ is nonzero, but it no longer symmetric about the $u_0$- and $u_1$-axis. This expresses the fact that knowing the value of $u_{0j}$ for some subject $j$ changes the expected value of $u_{1j}$. Likewise, knowing the value of $u_{1j}$ changes the expected value of $u_{0j}$. Figure 1 suggests that this is a more adequate assumption, so we let $\rho_u$ and $\rho_w$ vary in model 7. This means that $u_0$ and $u_1$ do not vary independently. On the other hand, we do assume that the matrices $u, w$ of by-subject and by-item random intercepts and slopes are independent. And $u$ and $w$ are both independent of $\varepsilon$ because $\varepsilon$ is still just Gaussian white noise.

Model 7, which is called a varying intercepts, varying slopes model, is useful in psycholinguistic research because it faithfully reflects the key sources of variance in the experimental design. The experimental design suggests a natural partitioning of logrt into groups associated with a given subject or a given item. These groups are specified by considering $\log rt_{ijk}$ with either the subject index $j$ or the item index $k$ held constant. Groups defined along these lines display systematically different patterns of variance. For example,
Figure 2: Simulated bivariate normal distributions for \( \langle u_0, u_1 \rangle \). The standard deviation \( \sigma_{u_0} \) of \( u_0 \) is taken to be .25 and the standard deviation \( \sigma_{u_1} \) of \( u_1 \) is taken to be .35. The parameter \( \rho_u \) of correlation takes on the values -.6 (top row), 0 (middle row), and .6 (bottom row). Notice that the marginal distribution of \( u_0 \) (center column) and \( u_1 \) (right column) does not change as \( \rho_u \) varies, but the joint (and conditional) distribution does vary (left column).
by-subject variance in language comprehension tasks has been attributed to factors such as individual differences in working memory capacity (Just and Carpenter, 1992) and in processing speed (Kliegl et al., 2010). Cognitive models of performance in such tasks make predictions about the relationship between a subject’s random intercept and slope. For example, in a two-condition design which manipulates relative clause type to be either subject or object relatives, a prediction of a working memory account could be that a slower subject will show a greater effect of relative clause type. This prediction can be evaluated experimentally by estimating the correlation between $u_0$ and $u_1$.

The variance components associated with subjects and items are often treated as nuisance parameters in psychology, i.e., as parameters which the psychologist does not desire to do any inference on. They are generally included in the model only to discount the possibility that the fixed effects estimates depend on the particular subjects and items used in the experiment. However, as mentioned above, there are situations where these variance components can be of theoretical interest. In these situations, it is important that the model provide a method for drawing inferences about these variance components.

LMMs have other important properties, such as “shrinkage” or partial pooling, which we do not discuss here; see Gelman et al. (2012) for details. We turn next to the practical details of model fitting in the probabilistic programming language Stan (Stan Development Team, 2014).

5. Example 1: A varying intercepts, varying slopes Bayesian LMM (two conditions)

In the previous section, we illustrated the LMM and motivated model 7, which has a fixed part to take into account planned experimental effects and a random part that defines by-subject and by-item variance. In this section, we implement this varying intercepts and varying slopes model in the Bayesian setting using the probabilistic programming language Stan. We illustrate this for the simple case of the two-condition design discussed above using the data of Gibson and Wu (2013). We begin by presenting the details of Stan syntax before applying our model to a simulated data-set, where we show how to estimate the random effects correlation.
5.1. Stan syntax

In the Bayesian setting, the end-result of the data analysis is the posterior distribution of the parameters of interest. In the above example, these parameters are $\beta_0$ and $\beta_1$ along with the variance estimates and correlations. These are the parameters which we would like to do inference with. We can refer collectively to these parameters as $\theta$. When we say $\theta$, we mean a list of parameters:

$$\theta = \langle \beta_0, \beta_1, \sigma_e, \sigma_u0, \sigma_w0, \sigma_w1, \rho_u, \rho_w \rangle$$ (11)

As discussed above, all the parameters are random variables with a probability distribution, not point values as they are in the frequentist setting.

Below we will use the probabilistic programming language Stan to make inferences about $\theta$. The key step in writing out an LMM specification in Stan is specifying the likelihood and the priors for the various parameters in $\theta$. The likelihood specification embodies our assumption about the underlying process that generated the observed data, and the specification of the prior distribution incorporates into the model what is known or believed about $\theta$.

The data that will be passed to the Stan model must have a particular format. The Gibson and Wu data-frame that was used to fit the lmer models above assumed that the data are an R object of type data-frame. For a Stan analysis, each of the columns of this data-frame is stored as a vector in an object of type list.

The Stan model itself is divided into several blocks of code: data, parameters, transformed parameters, model, and generated parameters. We discuss each of these blocks in sequence, building up the varying intercepts and varying slopes model for the Gibson and Wu data. Below we give the relevant Stan code chunks. Complete model specifications are provided in the online supplement.

5.2. The data block

In the data block of the Stan model, we assign types and lower and upper bounds to the variables that are present in the data that will be passed to Stan. Any text following two front-slashes (//) is a comment, similar to the hash mark (#) in R.

```stan
data{
  int<lower=1> N;  // no. rows
```
The strings real, int, and vector specify the data type for each variable. The variables so, and rt are vectors whose variable names are identical to the field names of the R object of type list which contains the data. Each of their $N$ elements is declared to be of type real. The variables $N$ (number of data points), $I$ (number of subjects), and $K$ (number of items) are integers passed from R, so they are assigned the int data type. The vector subj contains $N$ integer elements which indicate the subject id associated with the $n^{th}$ reading time. Thus, if the $n^{th}$ element in the vector subj is 3, then the $n^{th}$ reading time is associated with subject 3. The vector item has a similar interpretation. We also declare a vector zero containing two zeros; these will be used to define the priors on the variance-covariance matrices for the subject and item random effects. Variables can be constrained to take on only a subset of the values allowed by their data type, e.g. int or real. We do this in the variable declaration by specifying in brackets the lower and upper bounds on a variable’s support, e.g. $<$lower=1,upper=K$>$.

5.3. The parameters block

Next we have the parameters block. Here we declare model parameters, along with their types. We do not yet define priors for them. Note that we can also constrain the upper and lower bounds for the support of each parameter as we do for the variables in the data block. For example, the correlation parameters are constrained to lie between -1 and +1. Here is the parameters block for the Gibson and Wu data.

```
parameters{
  vector[2] beta; // fixed effects
  vector[2] u[I]; // subj adjustments
  vector[2] w[K]; // item adjustments
  real<lower=0> sigma; // SD of error term
```
5.4. The transformed parameters block

In the transformed parameters block, we define how the variance-covariance matrices are constituted, using the parameters declared in the parameters block above. We first declare a type for the $2 \times 2$ variance-covariance matrices $\Sigma_u$ and $\Sigma_w$, and then fill in each cell using the statement for the matrices that we saw above (see equations 8 and 9). There are more elegant ways to define the variance-covariance matrix, as discussed later.

```stan
transformed parameters{

cov_matrix[2] Sigma_u; // subj varcov matrix
cov_matrix[2] Sigma_w; // item varcov matrix
Sigma_u[1,1] <- sigma_u[1] * sigma_u[1];
Sigma_u[2,2] <- sigma_u[2] * sigma_u[2];
Sigma_w[1,1] <- sigma_w[1] * sigma_w[1];
Sigma_w[1,2] <- sigma_w[1] * sigma_w[2] * rho_w;
Sigma_w[2,2] <- sigma_w[2] * sigma_w[2];
}
```

5.5. The model block

In the model block, our main goal is to define how each dependent variable value $rt$ is generated. The standard approach in psychology is to assume the log-normal model: given some mean $\mu$ and some standard deviation $\sigma$, each data point $rt_i$, where $i = 1, \ldots, 547$, is assumed to be generated from a log-normal distribution with mean $\mu_i$ and fixed standard deviation $\sigma$. This means that $\log rt$ is normal. Stan represents the normal distribution using the mean and standard deviation and not the variance, unlike the standard practice in statistics. For consistency, we follow the Stan convention in the text of this paper as well.
\[ rt_i \sim \log N(\mu_i, \sigma) \]  

The parameter \( \sigma \) is familiar from before; it is the standard deviation of the error term, and it is constant for \( rt_i \) over all indices \( i \), i.e. it does not depend on what condition \( rt_i \) is from or which subject produced it by saying which item. But what is \( \mu_{ijk} \), and why does it have subscripted indices for observation, subject, and item? This is the sum of the intercept and slope parameters, and it has subscripted indices because its value is conditional on \( so_i \) and the identity of the associated subject \( j \) and item \( k \):

\[
\mu_{ijk} = \beta_0 + u_{0j} + w_{0k} + (\beta_1 + u_{1j} + w_{1k})so_i
\]  

In the Stan model block, we define a for-loop that specifies the value of each \( \mu_i \), and this is turn defines how the reading time is assumed to be generated.

```stan
model{
  real mu[N];
  // Define mu_i and how rt is generated:
  for(i in 1:N){
    mu[i] <- beta[1] + u[subj[i],1] + w[item[i],1]
    + (beta[2] + u[subj[i],2] + w[item[i],2]) * so[i];
  }
  rt ~ lognormal(mu,sigma);
  // Define how intercepts and slopes are generated:
  // By subject:
  for(i in 1:I)
    u[i] ~ multi_normal(zero,Sigma_u);
  // By item:
  for(i in 1:K)
    w[i] ~ multi_normal(zero,Sigma_w);
  // Priors:
  beta[1] ~ normal(0,10);
  beta[2] ~ normal(0,10);
  sigma ~ uniform(0,10);
  sigma_u ~ uniform(0,10);
  sigma_w ~ uniform(0,10);
  rho_u ~ uniform(-1,1);
}```
5.6. The generated quantities block

This block allows us to generate predicted values of the dependent variable (reading time) from the model. These represent predicted reading times which we haven’t yet observed but which we might observe if we repeated the experiment. We will use these predicted values to visualize how well the model fits the data, following a method advocated by Gelman et al. (2014). The predicted values, rt_pred, are generated by sampling them from a distribution with mean $\mu_{ijk}$ and standard deviation $\sigma$; the posterior distributions of these quantities have been computed above in the model block. The way we code the generated quantities block, the index of the for-loop runs from 1 to $N$, so we are not just generating an arbitrary set of predicted values, but rather predicting entire data-sets which share the same design as the data-set which defines the likelihood from the model block. So these are data-sets which the model predicts we might observe when replicating the experiment. As we discuss below, the posterior predictive distributions rt_pred are a useful tool for model evaluation and inference.

generated quantities{
  real rt_pred[N];
  real mu[N];
  for(i in 1:N){
    mu[i] <- beta[1] + u[subj[i],1] + w[item[i],1]
    + (beta[2] + u[subj[i],2] + w[item[i],2]) * so[i];
    rt_pred[i] <- lognormal_rng(mu[i],sigma);
  }
}

5.7. Running the model, evaluating fit, and inference on $\beta_1$

We assume that the Stan model is saved in an R object as a character string called normal rt GW, and the data are stored in an object called raw_dat. The command for running the model is shown below:

GW2 <- stan(model_code = lognorm rt GW,
    data = raw_dat,
    iter = 1000, warmup=500, chains = 4,
    pars=c("beta","sigma","sigma_u",
        "sigma_w","rho_u","rho_w",
        "rt_pred"))
The code above takes as parameters the model (which can be defined within R; see the R code accompanying this article), the data, the number of iterations over which the sampling from the posterior distribution should be done, the number of initial iterations that should be ignored (the warmup period), the number of independent Monte Carlo Markov Chains to be run for the sampling, and the list of parameters whose posterior distributions we want to examine.

Once we fit the model, we want to assess (a) whether the model converged, (b) what the relevant posterior distributions look like, and (c) whether the posterior predictive distribution of $rt$ is realistic given the data.

Convergence refers to the situation when samples are being drawn from the posterior distribution. Convergence is relatively easy to determine in rstan because the fitted model displays the Rubin-Gelman $\hat{R}$ convergence diagnostic (Gelman and Rubin, 1992). There is one $\hat{R}$ diagnostic for each parameter sampled from the posterior, and each of these should be approximately 1. This indicates good convergence and mixing of the MCMC chains. All models presented here had $\hat{R}$ of approximately 1. An alternative way to check for convergence is to use the traceplot function from rstan to examine whether the chains are mixing; Lunn et al. (2012) colorfully characterize well-mixed chains as “fat hairy caterpillars.”

Examining the posterior predictive distribution allows us to check whether the model makes sense, i.e., that it produces data similar to the data used to obtain the posterior distribution. These posterior predictive checks are shown in Figure 3. Figure 3 shows that the model does a reasonable job of predicting the minimum and median values in the data for each condition. However, the posterior predictive distribution does not extend to the maximum values observed in the data (subject relative: 6217 ms; object relative: 2308 ms). Although this model can be improved upon (for example, by assuming a mixture of distributions generating the reading times, see Gelman et al. (2014)), we keep the working assumption that $rt$ is approximately log-normal and turn next to making inferences from the posterior distribution of the parameter $\beta_1$.

The theoretical question is whether the slope $\beta_1$ is less than zero or greater than zero. If $\beta_1$ is negative, this means that object relatives are read more quickly than subject relatives, and if $\beta_1$ is positive, then subject relatives would be read more quickly than object relatives. This question can be answered straightforwardly by examining the posterior distribution of $\beta_1$. Figure 4 is a histogram of the posterior distribution. Stan returns the pos-
Figure 3: Posterior predictive checks on the log-normal model for the Gibson and Wu (2013) data-set. The figure shows a pseudo-randomly chosen posterior predictive distribution. The left panel shows the posterior predictive distribution for the subject relative condition and the right panel shows the posterior predictive distribution for the object relative condition. The three vertical lines in each plot show the minimum, median, and maximum reading times from each condition which were observed in the data-set.
Figure 4: The posterior distribution of the slope parameter in the Gibson and Wu data.
terior distribution of each model parameter. Once we have the posterior, we can then calculate whatever descriptive statistics we like on these probability distributions. For instance, we can take the 2.5 and 97.5 percentiles to obtain a 95% credible interval. We can also find the cumulative posterior probability distribution. On the basis of the cumulative probability distribution we report that the posterior probability of $\beta_1$ being less than zero is 0.92. This is quite strong evidence in favor of the claim that $\beta_1$ is less than zero.

5.8. Using new data to evaluate appropriateness of the model

The posterior predictive distribution of $rt$ represents the sample distribution of $rt$ which the model predicts if the experiment were run again. Gelman et al. (2014) suggest that one way to evaluate model fit is to compare such future data with the posterior predictive distribution of the earlier model. In previous work (Vasishth et al., 2013), we had repeated the Gibson and Wu experiment using their items (40 participants and 15 items), and so we are in a position to use the minimum, median, and maximum values of this new reading time distribution to evaluate the appropriateness of the log-normal model fit to the Gibson and Wu data.

We compare the posterior predictive distribution of the log-normal model fit to the original Gibson and Wu (2013) data with the sample distribution of $rt$ from the new Vasishth et al. (2013) data-set; see figure 5. We see that the minimum and median values of the new sample distribution fall within the predicted range, but the maximum values (subject relative: 10594; object relative: 3826) fall far outside the predicted range. This suggests that $rt$ is not log-normal. Either these extreme values should be modeled (using mixture modeling), or it is necessary to submit $rt$ to an appropriate transformation. See Vasishth et al. (2013) for more details about this data.

We turn next to an example illustrating how a $2 \times 2$ repeated measures design is analyzed.

6. Example 2: A varying intercepts, varying slopes Bayesian LMM with a $2 \times 2$ factorial design

In experiment 1 of Husain et al. (2014) we have a $2 \times 2$ repeated measures factorial design involving Hindi, which is an Indo-Aryan language spoken primarily in India. One factor is relative clause type (subject relative or object relative), and the other is the distance between the head noun and the relative clause verb. The theoretical interest is in determining whether
Figure 5: A comparison of the posterior predictive distribution based on the log-normal model for the Gibson and Wu (2013) data with the new data that was gathered using the Gibson and Wu items (Vasishth et al., 2013). The figure shows the distributions of four pseudo-randomly chosen posterior predictive samples for each condition. The three vertical lines in each plot show the minimum, median, and maximum values that occur in the new data.
each factor plays a role in determining reading time, and whether there is an
interaction between relative clause type (subject relative and object relative)
and distance.

The Stan code is very similar to the code for example 1, with a few ad-
ditional details. In the present case, since we have three predictors instead
of one (two main effects and one interaction), for a varying intercepts and
varying slopes model with a full variance-covariance matrix, we have to es-
timate the variance components in a 4 × 4 matrix for subjects and another
4 × 4 matrix for items. The method for specifying the variance-covariance
matrix prior that we used in example 1, although transparent, becomes quite
complicated. In order to understand the alternative method we use here,
recall that the covariance between two variables $u_0$ and $u_1$ is defined in terms
of the standard deviations of $u_0$ and $u_1$ ($\sigma_{u0}$, $\sigma_{u1}$) and their correlation ($\rho_u$):

$$\text{Covar}(u_0, u_1) = \rho_u \sigma_{u0} \sigma_{u1}$$  \hspace{1cm} (14)

A variance-covariance matrix has the variances for each variable along the
diagonal, and the covariances between the relevant pairs of variables in the
off-diagonal cells. It follows that for any variance-covariance matrix, we can
define a corresponding correlation matrix which has 1’s along the diagonal
and the correlations between the relevant pairs of variables along the off-
diagonals. To give a concrete example, the 2 × 2 variance-covariance matrix
$\Sigma_u$ in example 1 was:

$$\Sigma_u = \begin{bmatrix} \sigma_{u0}^2 & \rho_u \sigma_{u0} \sigma_{u1} \\ \rho_u \sigma_{u0} \sigma_{u1} & \sigma_{u1}^2 \end{bmatrix}$$  \hspace{1cm} (15)

It has a corresponding correlation matrix:

$$\begin{bmatrix} 1 & \rho_u \\ \rho_u & 1 \end{bmatrix}$$  \hspace{1cm} (16)

The diagonal elements are 1 because of equation 14. The variance $\sigma_{u0}^2$ can
be written as $\sigma_{u0} \sigma_{u0} 1$; i.e., the correlation between a variable $u0$ and itself
is 1.

In Stan, it is possible to define a prior for the correlation matrix corre-
sponding to a variance-covariance matrix, and then to assemble the matrix
using this specification. The Stan language has a built-in implementation
of a prior on correlation matrices. The function `lkj_corr` in Stan takes a
positive scalar value $\eta$ as its parameter. The correlation matrix can have a
prior specified in the following manner:

```
Omega_u ~ lkj_corr(4.0);
Omega_w ~ lkj_corr(4.0);
```

An lkj prior with $\eta$ equal to one corresponds to a uniform prior over the
space of correlation matrices of a given order ($2 \times 2$, $3 \times 3$, etc.). When
$\eta$ is between zero and one, there is a trough along the diagonal, meaning
that for each correlation parameter the prior probability density is shifted
away from zero toward the upper and lower bounds of the interval $[-1, 1]$.
When $\eta$ is greater than one, greater prior probability density lies near zero.
A parameter value of 4 is an appropriate value for $\eta$ because it is weakly
informative about the value of the correlation parameters. The variance-
covariance matrix is then assembled by first assigning the variances to the
diagonal elements and then assigning covariances to the off-diagonal elements
by equation 14; further details are in the Stan manual (Stan Development
Team, 2014).

6.1. Posterior predictive checks

As in example 1, we assume that the log-normal is the distribution which
underlyingly generates the observed reading times. As in the model of the
two condition design of example 1, we check the predictions of our model by
sampling $rt$ from its posterior predictive distribution $\text{logN}(\mu, \sigma)$. This yields
a set of posterior predictive distributions which represent future sample dis-
tributions which we might observe if we run another experiment with the
same $2 \times 2$ factorial design. The minimum, median and maximum reading
times of these predicted sample distributions are shown in Figure 6 for each
of the four conditions in the experiment. As in example 1 above, the ob-
served maximum values in each condition are outside the predicted range.
Together, examples 1 and 2 suggest that the log-normal model may need
some refinement. We leave this for future work, but we note that a further
advantage of the Bayesian framework is that it allows the researcher to add
such refinements.

6.2. Inference

Inference about the model parameters of the $2 \times 2$ design can be carried
out by examining the posterior distributions of the $\beta$ parameters that relate
Figure 6: Posterior predictive checks on the log-normal model for the Husain et al. (2014) data-set. This is one pseudo-randomly chosen posterior predictive distribution. Each panel shows the posterior predictive distribution for a different experimental condition. The three vertical lines in each plot show the minimum, median, and maximum reading times from each condition which were observed in the real data-set.
Figure 7: The posterior distributions of the three parameters representing the main effects and interaction in the Husain et al data.
to the main effects and interaction. As Figure 7 shows, the distance and relative clause type manipulations both have an effect in that both parameter estimates are negative, and an interaction is also present. The details about the interpretation are discussed in Husain et al. (2014).

Here, as always, inference about the magnitude or sign of model parameters like $\beta$ is contingent on the assumptions of the statistical model. A statistical model is a claim about the distribution which underlyingly generates the sample distribution of $rt$, so inference is contingent on our assumptions about how the data were generated. If we observe sample distributions in the experimental setting which are improbable under model assumptions, or if the model assumes the wrong sources of variability, then we are not justified in making inferences as if $rt$ were generated as that model assumes. For instance, if we assume that $rt$ is a random variable which follows a log-normal distribution, but we record sample distributions of $rt$ in the lab which indicate that $rt$ has too great a dispersion to be log-normal, then an inference made on the basis of the log-normal model would be difficult to interpret. We would be using the sample distribution of $rt$ to make inferences about the parameter of a model which does not generate $rt$. Similarly, if we assume that all systematic variability in $rt$ arises from by-subject and by-item random effects, but in fact there are sources of variability unaccounted for by the model such as which team of experimenters collected the data, then the inferences which we make under that model would be likewise of limited theoretical value. A strength of Bayesian modeling is that it offers a flexible way to define statistical models which conform to what we know about the underlying process which generates the sample distributions observed experimentally. This facilitates model criticism, which in turn leads to sounder inferences from improved statistical models.

7. Running the code associated with this tutorial

The source code for the Sweave file that generated this paper, as well as all the code and data used in this tutorial, can be downloaded from our website:

http://www.ling.uni-potsdam.de/~vasishth/statistics/BayesLMMs.html

8. Further reading

Most of the literature on Bayesian modeling uses the BUGS language (Lunn et al., 2000), which has a slightly different syntax than Stan. The
book by Gelman and Hill (2007) is an accessible introduction to fitting LMMs using BUGS syntax, which is what the probabilistic programming language JAGS uses (Plummer, 2012). The book by Lunn et al. (2012) is another important text that provides many instructive examples, also using BUGS syntax.

If the reader would like to learn more about Bayesian methods, we recommend Lynch (2007), which strikes a nice balance between formal rigor and accessibility; this book does assume some familiarity with calculus and probability theory. A less technical book, written for psychologists and cognitive scientists, is by Kruschke (2010). Many interesting applications of Bayesian models in cognitive modeling are discussed in Lee and Wagenmakers (2013). A more advanced treatment is provided by the classic textbook by Gelman et al. (2014), but this book assumes some background knowledge in calculus, probability theory, and statistics; the relevant background can be acquired from textbooks by Gilbert and Jordan (2002) (calculus and linear algebra) and Kerns (2010) (probability theory and statistics).

Acknowledgements

We are grateful to the developers of Stan (in particular, Andrew Gelman, Bob Carpenter) and members of the Stan mailing list for their advice regarding model specification. Douglas Bates and Reinhold Kliegl have helped considerably over the years in improving our understanding of LMMs from a frequentist perspective. We also thank Edward Gibson for releasing his published data.

References


Appendix A. Fitting full variance-covariance matrices using Bayesian LMMs vs \texttt{lme4}

In this section we report the results of a simulation study which compares a Bayesian LMM and an \texttt{lmer} model of the R package \texttt{lme4} with respect to how often the parameter of correlation between the random intercepts and slopes is at the boundary $\pm 1$ of its support or is NaN. We simulate 100 data-sets from a two-condition design which displays varying intercepts and varying slopes by subject and by item. The simulated subject and item sample sizes are 25 and 16, respectively. Each subject sees each item in the two conditions, so there are $25 \times 16 \times 2 = 800$ units of observation. This is a small sample size. The correlation between random intercepts and slopes is 0.6 for both subjects and items. We fit a Bayesian LMM to each data-set, as well as an \texttt{lmer} LMM using the restricted maximum likelihood algorithm of the R package \texttt{lme4}. We assess whether, under repeated sampling, the estimates of the parameter of correlation between random intercepts and slopes by subject $\rho_u$ and by item $\rho_w$ are at the boundary $\pm 1$ of their support or are NaN. We operationalize the notion of boundary estimate through the following definition. An estimate $\hat{\rho}$ of a random effects correlation parameter is a boundary estimate if $1 - |\hat{\rho}| < 1 \times 10^{-4}$. For the restricted maximum likelihood fits using \texttt{lmer}, we find that 58\% of the estimates of $\rho_u$ and 63\% of the estimates of $\rho_w$ are boundary estimates. In contrast, 0 of the 100 Bayesian LMMs has an upper or lower bound on the 95\% credible interval for either correlation estimate which lies within $1 \times 10^{-4}$ of $\pm 1$.

Appendix B. The log-normal versus normal distribution as the generative distribution

We typically want to make inference on a set of parameters which is of theoretical interest when analyzing experimental results in psychology. Suppose we are analyzing an experiment whose dependent measure is the reading time $rt$ on a target item. Each subject sees each item in each of two conditions, and we are interested in the difference $\delta$ in mean $rt$ between the two conditions. The size $\delta$ of the effect for the experimental manipulation is then the parameter of theoretical interest. We want to draw an inference on the value of $\delta$. This section justifies an operational assumption that $rt$ is approximately log-normal and shows that inference on $\delta$ is more reliable if we assume that $rt$ follows a log-normal distribution. Specifically, in the case
where $\delta$ really is nonzero, a log-normal model of $rt$ excludes zero from the interval from the 2.5% to the 97.5% quantile (hereafter just credible interval) of the posterior distribution of $\delta$ more often than a normal model fit to $\log rt$. This motivates the operational assumption of this paper that $rt$ follows an approximate log-normal distribution.

In this paper we make the operational assumption that $rt$ is distributed log-normally. This means that the logarithm of $rt$ is normal. We write this as

$$rt \sim \log N(\mu, \sigma), \quad (B.1)$$

where $\mu$ is the mean of $\log rt$ and $\sigma$ is the standard deviation of $\log rt$. We assess the adequacy of this distributional assumption with the published data-set of Gibson and Wu (2013). In their experiment, the dependent measure $rt$ was measured in two conditions. Each of 37 subjects saw each of 15 items in one of the two conditions.

We use this data-set as a characteristic example of a sample distribution of reading times in order to motivate the assumption that $rt$ follows an approximate log-normal distribution. Figure B.8 shows the distribution of $rt$ conditional on the experimental condition. We use maximum likelihood to estimate the parameter values of the log-normal distribution and of the normal distribution for each of the two conditions separately and superimpose the probability distribution function on the reading time data. The graphical presentation of the results suggests that the log-normal distribution is at least a more adequate distributional assumption for $rt$ than the normal distribution. The normal fit has substantial density below zero, but $rt$ only has positive support under a log-normal model. The normal model is also too symmetric compared to the data. In contrast, the long right tail of the log-normal fit characterizes the distribution of $rt$. These two comparisons justify that $rt$ is log-normal, at least as a working assumption.

Assuming that $rt$ follows a log-normal distribution leads furthermore to a practical advantage in the inferential process. When we plan an experiment with the goal of making inference on the effect size $\delta$, the theoretical question guiding our investigation determines in advance how precisely we need to know the value of $\delta$. This precision is reflected in the dispersion of the posterior distribution of $\delta$. For example, we quantify this dispersion as standard deviation when $\delta$ is normally distributed. Several properties of the experimental design and the data analysis determine the dispersion of the posterior distribution of $\delta$ given the sample distribution $rt$. The dispersion of
Figure B.8: The distributions of the reading time data in the Gibson and Wu (2013) dataset, with a superimposed log-normal distribution (top row), and a normal distribution (bottom row). The parameters of the log-normal and normal distributions were estimated using maximum likelihood estimation.
the posterior distribution depends, among other things, on the sample size \( n \) of \( rt \) and on the distributional assumptions which we make about the dependent variable \( rt \). Often a psychological experiment is designed to determine not the exact magnitude of \( \delta \), but only whether it falls above or below zero. In this case, the strength of the evidence about the sign of \( \delta \) depends not only on \( n \) and distributional assumptions, but on the dispersion of the posterior probability density of \( \delta \) and the absolute value of \( \delta \). When the posterior of \( \delta \) has little dispersion and the absolute value of \( \delta \) is great, much probability density will fall over the interval \((-\infty, 0)\) or \((0, \infty)\) in which \( \delta \) actually lies. In the design of a planned experiment whose aim is to determine the magnitude of \( \delta \), it is useful to calculate the magnitude of \( n \) needed to obtain the posterior distribution of \( \delta \) precisely enough, given a guess at \( \delta \) and a statistical model. Once we have a guess and a model, the precision with which we want to know \( \delta \) then determines \( n \). Say we want to derive a posterior distribution of \( \delta \) whose density is concentrated enough to have the whole credible interval of \( \delta \) in \((-\infty, 0)\) or \((0, \infty)\), depending on which interval \( \delta \) lies in. The probability of deriving such a posterior depends on data which we have not yet collected at this stage of experimental design, but hypothetical data-sets can be simulated which reflect what we know about the experiment. We can then use these to estimate how big \( n \) must be in order to obtain the posterior distribution of \( \delta \) sufficiently precisely. For instance, suppose we want to conduct an experiment with a two-condition repeated measures design, and expect (based on previous experiments) that the sample distribution of \( rt \) will have a mean somewhere around 600 milliseconds, a long right tail, and a minimum greater than zero. Then we can simulate data-sets which have these properties. We let the true \( \delta \) of the simulated data-sets range in four steps from 15 milliseconds, a small effect, to 100 milliseconds, a very large effect. We also let the subject sample size vary in four steps from 35 to 80. In our simulated experiment, each subject sees each of 16 items in each of the two experimental conditions, so increasing the subject sample size increases \( n \). We conceptualize each combination of the four simulated values of \( \delta \) with the four simulated subject sample sizes as forming a four by four grid as in the panels of figure B.9. We simulate 100 data-sets in each of these cells and fit two Bayesian LMMs to each of these data-sets. One model assumes that \( rt \) follows a normal distribution and the other model assumes that \( rt \) follows a log-normal distribution. A color scale indicates the percent fitted models which exclude zero from the credible interval of \( \delta \). The consequence of increasing the subject sample size (increasing \( n \)), is to increase this percentage,
Figure B.9: Figure B.9a shows the results of the simulation study for a normal model fitted to logrt. Each cell reports the results for a given effect size and subject sample size. A color scale indicates the percent fitted models for which the credible interval of $\delta$ excludes zero. Figure B.9b summarizes the same results as figure B.9a, but the model is log-normal and fitted to rt. Figure B.9c shows cellwise difference between figures B.9a and B.9a. In each cell, the log-normal model’s percentage is subtracted from that of the normal model, so a darker cell indicates the log-normal model performed better in that cell than the normal model.

and the same happens with an increasing magnitude of $\delta$. Qualitatively, we make this same observation for both the normal (figure B.9a) and the log-normal model (figure B.9b), but the percentage is greater overall for the log-normal model, especially in the intermediate range of subject sample size and effect size (cf. figure B.9c). These intermediate effect sizes and subject sample sizes are representative of actual experimental work done in psychology. If researchers collect sample distributions of rt in the laboratory setting which follow an approximate log-normal distribution, then the credible interval of $\delta$ more often excludes zero when $\delta$ is nonzero under a log-normal model than under a normal model. That is, the log-normal model is better at leading to the correct inference regarding $\delta$ when $\delta$ really is nonzero in nature.

A representation like figure B.9 offers practical guidelines for the design of experiments. Even when we do not know the magnitude of $\delta$ with much precision in advance, if the statistical model and the simulated data-sets reflect the process which generates rt underlingly, then we know from figure B.9 how many subjects to run in order to derive a posterior distribution of $\delta$ which is precise enough to support our hypothesis that $\delta$ is greater than zero with 97.5% certainty.

We further conduct a simulation study where $\delta$ does not vary as before
but is instead fixed at zero. Subject sample size varies over the same range as before. We fit the same two normal and log-normal models to 100 simulated data-sets for each of four subject sample sizes and report the percent fitted models for which the credible interval of $\delta$ excludes zero in table B.3. The effect size $\delta$ is truly zero in these simulated data-sets, and so the more adequate model excludes zero from this interval less often. We see that the log-normal model performs marginally better.

<table>
<thead>
<tr>
<th>no. subjects</th>
<th>normal</th>
<th>log-normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>62</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table B.3: Results from a simulation study on the percent fitted models (normal and log-normal) which uncover an effect in $\delta$ when the true value of $\delta$ is zero.