Bayesian regression modeling (for factorial designs): A tutorial

Michael Franke & Timo Roettger

Generalized linear mixed models are handy tools for statistical inference, and Bayesian approaches to applying these become increasingly popular. This tutorial provides an accessible, non-technical introduction to the use and feel of Bayesian mixed effects regression models. The focus is on data from a factorial-design experiment.

This tutorial should take you about 1.5 hours.

We would like to thank Oliver Bott, Joseph Cassilas, Artur Czeszumski, Fabian Dablander, Judith Degen, Elisa Kreiss, and Bodo Winter for their invaluable comments and suggestions on an earlier draft.

Motivation & intended audience

This tutorial provides a very basic introduction to Bayesian regression modeling using R (R Core Team, 2017). We wrote this tutorial with a particular reader in mind. If you have used R before and if you have a basic understanding of linear regression, and now you want to find out what a Bayesian approach has to offer, this tutorial is for you. In comparison to other introductions (e.g. Sorensen, Hohensteinb, and Vasishth, 2016), this tutorial remains very conceptual. We don't want to "sell Bayes" to you, and we do not want to scare you away with mathematical details. We just want to give you an impression of how a Bayesian regression analysis looks and feels. The tutorial covers the essential concepts and explains how to run and interpret the output of a Bayesian regression analysis using the wonderful R package `brms` written by Paul Buerkner (2016).

This tutorial contains gray text boxes which contain additional background information. The information is sometimes a bit technical but never absolutely necessary for understanding the main ideas. So feel free to read or skip any of the text boxes to suit your needs.

If you don't have any experience with regression modeling, you will probably still be able to follow, but you might also want to consider doing a crash course. To bring you up to speed, we recommend the excellent two-part tutorial by Bodo Winter (2013) on mixed effects regression in a non-Bayesian —a.k.a. frequentist— paradigm. In a sense, this tutorial could be considered part three of the series started by Winter. We will for example use the same data set.

To actively follow this tutorial, you should have R installed on your computer (`https://www.r-project.org`). Unless you already have a favorite editor for tinkering with R scripts, we recommend to try out RStudio (`https://www.rstudio.com`). You will also need some packages, which you can import with the following code:

All code and data for this tutorial is also available for download here: `https://tinyurl.com/bmr-tutorial`

```
# package for convenience functions (e.g. plotting)
library(tidyverse)

# package for Bayesian regression modeling
library(brms)

# option for Bayesian regression models:
# use all available cores for parallel computing
options(mc.cores = parallel::detectCores())

# package for credible interval computation
library(HDInterval)

# set the random seed in order to make sure
# you can reproduce the same results
set.seed(1702)
```

Data, research questions & hypotheses

This tutorial looks at a data set relevant for investigating whether voice pitch differs across female and male speakers, and whether it differs across social contexts (say: informal and polite contexts). To load the data into your R environment, run the following code:

```
# load the data into variable "politedata"
politedata = read_csv("https://tinyurl.com/polite-data")
```

The data is originally from research presented by Winter and Grawunder (2012). It is used in the tutorials by Winter (2013) as well, but here we 'massaged' the data a bit, i.e., we renamed variables and removed a line with missing data.

Type `head(politedata)` and you should see the first lines of the data:

```
> head(politedata)
   subject gender sentence context pitch
   <chr>   <chr>  <chr>    <chr>   <dbl>
 1 F1      F      S1       pol      213.
 2 F1      F      S1       inf      204.
 3 F1      F      S2       pol      285.
 4 F1      F      S2       inf      260.
 5 F1      F      S3       pol      204.
 6 F1      F      S3       inf      287.
```

This data set contains information about different subjects, with an anonymous identifier stored in variable `subject`. Because voice pitch is highly dependent on gender (i.e. there are anatomical differences between women and men that affect voice pitch), the variable `gender` stores whether a subject is F(emale) or M(ale). Subjects produced different sentences (stored in the variable `sentence`), and the experiment manipulated whether the sentence was produced in a polite or an informal context, indicated by the variable `context`. Crucially, each row contains a measurement of pitch in Hz stored in the variable `pitch`.

Often, we are interested in an **outcome variable** (also called response or dependent variable, here `pitch`). We want to know how this outcome variable behaves across different conditions or groups, which we call **predictors** (also called independent variable, here `gender` and `context`). Before

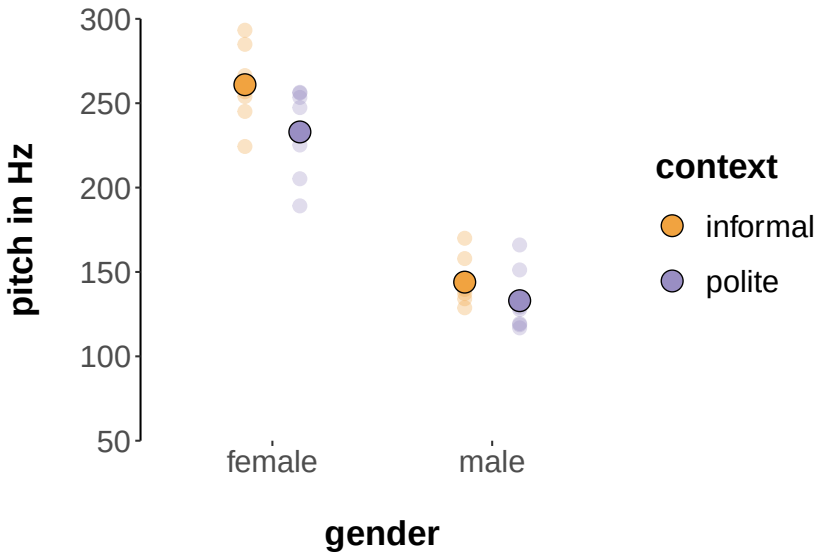

Figure 1: Basic plot of the data displaying overall averages (thick points) and averages for individual sentences (smaller semitransparent points).

our data collection, we might have formulated concrete predictions about the relationship between the outcome and predictors. Let's assume previous research suggests that pitch is an indicator of politeness. Informal speech is accompanied by higher pitch in both men and women. Because the physiological difference between men and women is very large, we also predict that even in informal contexts, men still have lower pitch than women in polite contexts. We formulate the following hypotheses:

H1: Female speakers have a lower average pitch in polite than in informal contexts.

H2: Male speakers have a lower average pitch in polite than in informal contexts.

H3: Male speakers have a lower average pitch in informal than female speakers have in polite contexts.

Exploring the data visually

Figure 1 displays the mean pitch values for each sentence (semi-transparent points) across gender and context. The solid points indicate the average pitch values across all sentences and speakers. Looking at the plot, we can see that pitch values from female speakers are generally higher than those from male speakers (points in the left column are higher than in the right column). We also see that pitch values in the informal context are slightly higher than those in the polite context (orange points are slightly higher than purple points).

Extensive plotting is always recommended to start data analysis. You need to know your data inside out. Pictures often reveal relationships much better than numbers can.

Looking at the plot, we might want to shout: "The data confirm all of our hypotheses!" But, of course, we need to be more careful. As Bayesians, we

would like to translate the data into an expression of **evidence**: does the data provide evidence for our research hypotheses? – Also, notice that there is quite a lot of variability between different sentences (the semi-transparent points in Figure 1). For example, some values from the informal condition for female speakers (orange points in left column) are lower than their corresponding polite counterparts. Similarly, there could be differences between individual speakers. What we want are precise estimates of potential differences between conditions. We also want a measure of how confident we can be in these estimates.

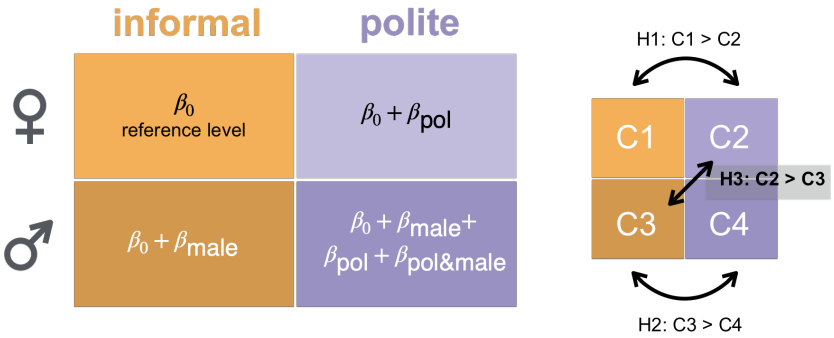

Figure 2: Coefficients of a dummy-coded regression model for the factorial 2×2 design, together with research hypotheses as statements about ordinal relations between cell means.

A regression model for our data

Another way of looking at the data in connection with our research hypotheses is displayed in Figure 2. Each cell in this **design matrix** represents one unique combination of the gender and the context factor. Our hypotheses can be related to the comparison between the means of these cells. H1 makes a statement about the comparison between C(ells) 1 and 2 (the context effect for female speakers); H2 makes a statement about C3 and C4 (the context effect for male speakers); and H3 makes a statement about C2 and C3 (the difference between informal male speakers and polite female speakers).

Before going into data analysis, let's look at the **regression model** we want to use. Our regression model assumes that pitch values observed in each cell are sampled from a population that is normally distributed around a mean, where each cell c_i has its own mean μ_i. We are interested in the probability of one cell mean being larger than another cell mean, i.e., the probability that $\mu_i > \mu_j$. Put differently, we are interested in the probability that the difference between μ_i and μ_j is larger than zero: $\mu_i - \mu_j > 0$. Figure 2 illustrates the encoding scheme of our cell means in terms of a regression analysis. It assumes that there is a **reference level** for each factor. Here it is the level `female` for the factor `gender` and the level `informal` for the factor `context`.

The setup of this (non-hierarchical) regression model is not specific to a Bayesian approach. You would use the exact same for a frequentist analysis.

All cell means can then be expressed in terms of differences between the

This is so-called **dummy coding** (also referred to as **treatment coding**) of the regression coefficients. Other coding schemes exist, but are not discussed here.

intercept aka our reference level β_0, deviations from this reference level for each individual factor (β_{male}, and β_{pol}), and a so-called **interaction term** $\beta_{\text{pol\&male}}$. In other words, our regression estimates the mean of the reference level and estimates how much we need to adjust this mean when we change either the context level (C2), the gender level (C3), or both (C4). The β terms are also called **coefficients**. They are **free parameters** of the model.

A Bayesian analysis of a (fixed effects) regression model

Another approach to testing hypotheses in a Bayesian setting is to use model comparison. This tutorial focuses on parameter estimation only for parallelism with the standard frequentist practice. We will briefly touch on model comparison at the end.

Having spelled out a model like the above, one way of testing our hypotheses in a Bayesian setting uses so-called **parameter inference**. Bayesian parameter inference asks: What should we believe about the values of the coefficients $\beta_0, \beta_{\text{pol}}, \beta_{\text{male}}$ and $\beta_{\text{pol\&male}}$ given the data, the model and whatever we believed before having seen the data?

Formally, if θ is a vector of parameter values of the model, we are interested in the **posterior distribution** $P(\theta \mid D)$, which assigns a non-negative number to each tuple of parameters in proportion to how likely that tuple is. From a Bayesian point of view, the model consists of a **likelihood function** $P(D \mid \theta)$, which specifies how likely an observation of data D is for each value of parameters θ, and a **prior distribution** $P(\theta)$, which specifies how likely we (the rational reasoner) believe each tuple of parameter values is in the first place. With these ingredients, we can compute the posterior distribution by Bayes rule, as follows:

Supplying a prior, really just *any* prior, is important to get a Bayesian analysis off the ground; a circumstance which is discussed controversially. For many practical purposes, however, the precise choice of priors is not decisive and tools like the `brms` package (which we will use here) chooses generically reasonable default priors for the model (more on this below).

$$P(\theta \mid D) = \frac{P(\theta)\ P(D \mid \theta)}{\int P(\theta')\ P(D \mid \theta') \mathrm{d}\theta'}$$

The R package `brms` (Buerkner, 2016) makes it easy to run Bayesian regression models. It uses a very similar formula syntax as related packages for regression analysis. In our case, we want to regress the dependent variable `pitch` against the independent variables `gender` and `context` and their two-way interaction. This model is expressed by the formula:

```
# formula for (fixed effects) regression model
formula_FE = pitch ~ gender * context
```

The Bayesian model can then be fitted with the function `brm` from the `brms` package. We only need to specify the formula and supply the data. Here, we also set a seed to make sure we all produce exactly the same results.

```
# run regression model in brms
model_FE = brm(
  formula = formula_FE,
  data = politedata,
  seed = 1702
)
```

The `brms` package uses the probabilistic programming language `Stan` in the background. `brms` basically translates our formula into Stan code that

implements a regression model and then executes it. The Stan code is then translated to C++ (hence the message about "compiling C++" when you run this code). Both the compilation in C++ and the sampling in Stan can sometimes take a while. It might also eat your battery in no time, so always make sure your laptop is plugged in. Conceptually, Stan obtains samples from the posterior distribution, based on an algorithm called *Hamiltonian Monte Carlo*. This is an instance of a more general class of algorithms, called *Markov Chain Monte Carlo (MCMC)* methods. The purpose of these methods is to return representative samples from the posterior distribution. If you are interested in finding out more about this 'sampling stuff', check out Info Box 2.

You can enter `model_FE` in order to get a summary of the model fit. It should look like the following output.

```
1  > model_FE
2   Family: gaussian
3    Links: mu = identity; sigma = identity
4  Formula: pitch ~ gender * context
5     Data: politedata (Number of observations: 83)
6  Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
7           total post-warmup samples = 4000
8
9  Population-Level Effects:
10                    Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
11 Intercept            260.64      7.76   245.18   275.77       2148 1.00
12 genderM             -116.17     11.16  -137.71   -94.36       2115 1.00
13 contextpol           -27.43     11.11   -48.75    -6.36       1959 1.00
14 genderM:contextpol    15.83     15.87   -15.16    46.89       1975 1.00
15
16 Family Specific Parameters:
17       Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
18 sigma    36.14      2.81    31.09    42.08       4416 1.00
19
20 Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
21 is a crude measure of effective sample size, and Rhat is the potential
22 scale reduction factor on split chains (at convergence, Rhat = 1).
```

Lines 2–5 give us information about the model and the data used. Lines 6 and 7 tell us about the sampling procedure (see Info Boxes 2 & 3 for more information). Lines 9–14 contain information about our parameters of interest. We will discuss them in detail below. Lines 16–18 contain information that look similar to those in lines 9–14. This is the estimation of the standard deviation `sigma`, describing the variance of the assumed normal distribution (which describe the distribution of measures in each design cell). Finally, lines 20–22 contain general information about the model fit and the information presented in this summary.

If the model failed to converge or other problems occurred, you would see an informative message in the last part of this summary. See Info Box 2 for more on "convergence".

Let us look at lines 9–14 in detail now. What these lines give us is a table with four rows, each of which corresponds to a parameter in the model, namely the coefficients shown in Figure 2. The variable `Intercept` refers to the coefficient β_0, which represents the mean of the reference level in cell 1 (female speakers in polite contexts). The variable `genderM` corresponds to

Markov Chain Monte Carlo (MCMC) sampling

Bayesian ideas are old, but they have recently seen a revival, mainly due to advances in computer science (notably: clever algorithms, not just faster computers). To understand this, consider Bayes rule for data analysis. We have a prior $P(\theta)$ over parameter θ and a likelihood function $P(D \mid \theta)$. We want to compute the posterior distribution:

$$P(\theta \mid D) = \frac{P(\theta)\ P(D \mid \theta)}{\int P(\theta')\ P(D \mid \theta')\mathrm{d}\theta'}$$

If θ is a high-dimensional vector of parameters (e.g., in a hierarchical regression model), it might be quite impossible to compute the integral in the denominator. Fortunately, clever algorithms like MCMC allow us to draw **representative samples** from the posterior distribution without having to calculate the integral-of-doom.

For common applications, it is not required to understand MCMC algorithms in full detail. It suffices to know that samples are collected by starting at (usually random) initial parameter values, and then "jump around the parameter space" in such a way that, if we were to jump infinitely often, we will have visited any particular tuple of parameter values with a relative frequency that corresponds exactly to its posterior probability. So these algorithms are guaranteed to give us representative samples, given unlimited time to jump around.

To ensure the trustworthiness of a finite set of samples, we routinely run several **chains**, i.e., we start the "jumping around" procedure at different random initial starting points and check whether the different chains reached a similar outcome. We do this by plotting (so-called *trace plots*) and the $\hat{R}$**-value** which is included in the `brm` model summary. An $\hat{R}$-statistic compares the variance of the samples within each chain to that from all samples across chains. If the $\hat{R}$-value is below 1.1, we commonly assume that the chains have converged sufficiently.

In practice, the `brm` model summary and the messages during the fitting process will inform you about potential problems, and will normally offer good advice on how to solve the issue. For example, sometimes the different chains do not converge on sufficiently similar results. One quick and simple solution to these "convergence issues" is increasing the number of samples (specifying the option `iter` in the `brm` function call).

Info Box 1: Background on sampling methods & diagnostics.

Information displayed in the summary of a `brm` *model fit*

Lines 6–7 of the summary of `model_FE` tell us that we collected a total of 4000 samples from the posterior distribution. These came from four chains, each running for 2000 iterations, but discarding the first 1000 samples as **warm up**. We discard the first 100 samples because the initial starting point might be quite "unrepresentative" (see also Info Box 2).
From the table in lines 9–14, the column *Estimate* gives the mean of the obtained samples, thereby approximating the mean of the (marginal) posterior distribution for each parameter. For example, the parameter `Intercept` is estimated to have a mean of about 261, which (here) coincides with the mean of the data points in cell 1, as shown in Figure 2. *Est.Error* is the estimation error, an indication of the certainty we should have about the whole inference procedure. The columns `l-95% CI` and `u-95% CI` give the lower and upper bound of the 95% credible interval for each parameter, as discussed in the main text. The column `Eff.Sample`, for efficient samples, gives us a rough measure of how many of all the samples we took (4000 in our case) are contributing non-redundant information to our estimation. The higher this number, the better. Finally, we get the `Rhat` column with the $\hat{R}$-values for each parameter (see Info Box 2).

Info Box 2: Information in summaries of `brm` model fits.

Intuitively, the 95% credible interval is the range of values that we can often practically consider credible enough to care about.

the coefficient β_{male}, `contextpol` corresponds to the coefficient β_{pol}, and `genderM:contextpol` is the interaction coefficient $\beta_{\text{pol\&male}}$. For each of these parameters, the table contains very useful summary statistics based on the samples returned from the model fit. More about the information given in the columns can be found in Info Box 3.

For our current purposes, the information in columns `l-95% CI` and `u-95% CI` is most important. These numbers give the lower and upper bounds of a **95% credible interval**. Take the parameter `contextpol`, corresponding to the coefficient β_{pol}. This parameter corresponds to the estimated change of the mean of the reference level when we change the context level to polite. The 95% CI is roughly [-49;-6]. We could take values outside of this interval to be sufficiently unlikely to be ignorable for most purposes. Consequently, this analysis suggests that zero is a very unlikely value for the coefficient β_{pol}. Given the data and the model, we should believe that β_{pol} is most likely negative. This directly addresses the first research hypothesis. Based on the regression model, the data suggests that H1 is likely true.

But how likely? How likely is it that β_{pol} is smaller than zero? Instead of simply making a binary thumbs-up / thumbs-down decision, it would be even more elegant if we could put a number to it. As Bayesians we fortunately can. To see how this works, let us have a more intimate look at the samples

that the `brm` function returns. We can access the samples of a model fitted with `brm` with the function `posterior_samples`:

```
# extract posterior samples
post_samples_FE = posterior_samples(model_FE)
head(post_samples_FE %>% round(1))
```

The output of this could look like this:

```
  b_Intercept b_genderM b_contextpol b_genderM:contextpol sigma   lp__
1       262.0    -117.6        -25.1                 13.5  38.8 -420.1
2       257.8    -118.9        -32.0                 12.6  37.6 -421.5
3       255.4    -114.4        -28.0                 11.4  37.2 -420.5
4       249.4    -101.1         -8.9                 -5.6  33.8 -421.0
5       283.4    -137.5        -46.1                 43.4  42.2 -425.4
6       284.8    -127.8        -45.8                 39.7  40.6 -427.6
```

What you see here is the top 6 rows of a data frame with columns for each parameter and 4000 rows, corresponding to each sample of that parameter (so the sampling method has generated 4000 values for each parameter, where each row is a sample from the posterior distribution $P(\theta \mid D)$). We can use these samples to produce density plots reflecting our posterior distribution. The plot in Figure 3 shows, for each of the four main model parameters, an estimate of the posterior density. Each curve shows how much credence we should put on particular parameter values. For example, we see that our beliefs concerning plausible values for the mean of cell 1 (female speakers in informal contexts, the reference level) should hover around 260, roughly spreading from about 240 to 280. We also see that all values that receive substantial probability density for `context:pol` (the coefficient β_{pol}) are negative (as captured in the 95% CI discussed above). In other words, the voice pitch of women in polite contexts is likely to be lower than in informal contexts. Zero is estimated to be a comparatively unlikely value for this parameter.

The column `lp__` contains the log-probability of the data for the parameterization in each row. This is useful for model comparison and model criticism but not important for our current adventures.

Now, here comes a nice gadget. Based on the samples obtained for `contextpol` (β_{pol}), it is very easy to estimate our belief that β_{pol} is indeed negative. We simply have to calculate the proportion of samples that were negative. That's all. For instance, with the code below, which reveals that the posterior probability of the proposition that $\beta_{\text{pol}} < 0$ is about 0.99375. This is very close to 1!

```
# proportion of negative samples for parameter b_contextpol
# this number approximates P(b_contextpol < 0 | model, data)
mean(post_samples_FE$b_contextpol < 0)
>0.99375
```

As an interim summary, we have seen how to run a Bayesian regression analysis with the `brms` package and how to deal with its output. We have also seen that the output can be interpreted in very intuitive ways (e.g., "The probability of H1, given our model, priors, and data, is more than .99").

Unfortunately, what we have not seen yet is what the model and data say

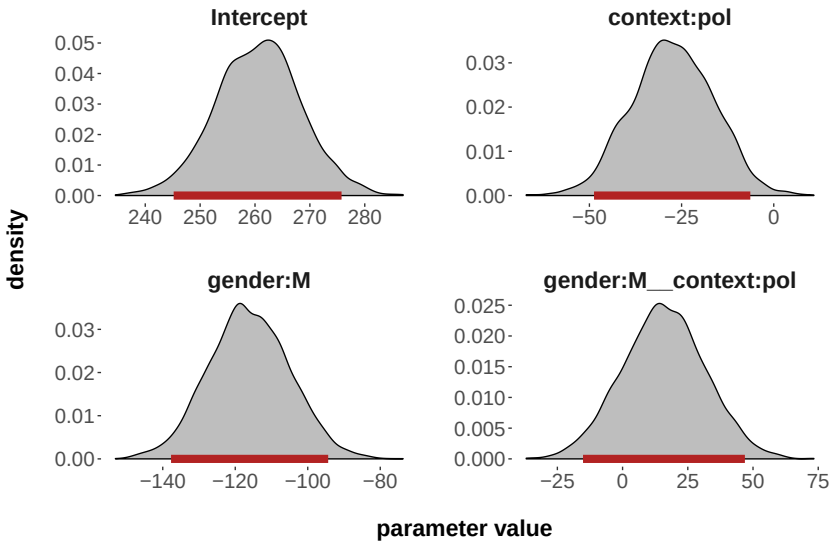

Figure 3: Posterior density of parameter values in the fixed-effects regression model. The thick red lines indicate the 95% credible intervals, i.e., the range of parameter values that it is reasonable to believe in.

about hypotheses 2 or 3. This is because there is no single parameter in the (dummy-coded) regression model that corresponds to the differences between cells 3 and 4 (for hypothesis 2) and cells 2 and 3 (for hypothesis 3). Notice that this problem is not specific to Bayesian analyses, but inherent in the way the regression coefficients were set up. Fortunately, we can recover information about any derived measure from the obtained samples. Here's how:

Take hypothesis 3 which requires us to compare cells 2 and 3. The hypothesis states that $\beta_0 + \beta_{\text{pol}} > \beta_0 + \beta_{\text{male}}$, which reduces to $\beta_{\text{pol}} > \beta_{\text{male}}$. We can approximate the posterior probability that this is true based on the samples that we obtained for the model in the same general way as before, namely:

```
# proportion of samples for which the mean of cell 2 was larger
# than that of cell 3
# this number approximates the quantity:
#    P(b_contextpol > b_genderM | model, data)
mean(post_samples_FE$b_contextpol > post_samples_FE$b_genderM)
>1
```

Based on the posterior samples, the estimated probability is 1. That's a strong result. If the model was true, (given the data) our certainty that hypothesis 3 is true should be pretty much almost at ceiling.

To sum up, the Bayesian approach to regression modeling allows us to retrieve all direct comparisons between cells in a factorial design. It also allows us to retrieve quantitative information about our hypotheses in a way which is easy to communicate and understand. We can calculate the (estimated) posterior probability that a particular hypothesis holds.

A potential way of testing different hypotheses of the kind we have set out here, is to run different regression analyses, each with a different reference cell. This is a rather unhandy work flow. It wouldn't help with hypothesis 3 either, which compares the cell means in the design matrix "diagonally": there is no way of changing the reference level of either factor such that dummy coding gives us a single coefficient as the difference between cells 2 and 3.

The `faintr` package

To make the comparison of pairs of cells even easier and applicable for even bigger factorial designs, this tutorial comes with a little R package, the `faintr` package. You can install the package from GitHub with the `devtools` package, as follows:

```
# load 'devtools' package to allow installation from GitHub
library(devtools)
# install 'faintr' package from GitHub
install_github(
  repo = "michael-franke/bayes_mixed_regression_tutorial",
  subdir = "faintr")
# load the 'faintr' package
library(faintr)
```

The `faintr` package provides two main helper functions. The function `post_cells()` takes as input the fitted regression model (technically: the `brmsfit` object returned by function `brm`) and outputs samples for all design cell means, a comparison of all design cells against each other, and a summary of each cell's inferred mean value. For example, although the model fitted β-coefficients, we can reconstruct posterior estimates of each cell's means by typing:

More on the use of the `faintr` package can be found here: `https://tinyurl.com/faintr-vignette`.

```
post_cells(model_FE)$predictor_values
```

Using these reconstructed samples, we can plot approximate posterior estimates for each cell, as in Figure 4.

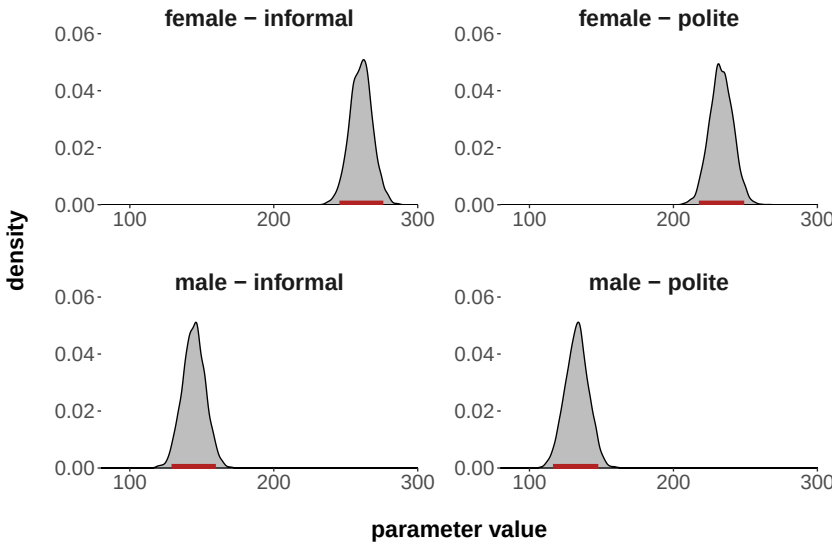

Figure 4: Posterior density of cell means in the fixed-effects regression model. The thick red lines indicate the 95% credible intervals, i.e., the range of parameter values that are reasonable to believe in (given the data and the model.

Another helpful function is `compare_groups()`, which takes the fitted regression model as input together with a specification of which two (subsets of) cells to compare against each other. For example, we can compare (diag-

onally) the cells for female speakers in polite contexts with male speakers in informal contexts with this call:

```
compare_groups(
  model = model_FE,
  lower = list(gender = "M", context = "inf"),
  higher = list(gender = "F", context = "pol")
)
```

The output looks like this:

```
Outcome of comparing groups:
 * higher:  gender:F context:pol
 * lower:   gender:M context:inf
Mean 'higher - lower':  88.74
95% CI: [ 68.09 ; 111 ]
P('higher - lower' > 0):  1
```

For the purposes of this tutorial, we gather the output of the `compare_groups()` for all of the three hypotheses of current interest in a convenience function (which the interested reader will find in the source code, but which is of no further theoretical interest here):

```
> get_posterior_beliefs_about_hypotheses(model_FE)
# A tibble: 3 x 2
  hypothesis                       probability
  <chr>                                  <dbl>
1 Female-polite < Female-informal        0.994
2 Male-polite < Male-informal            0.849
3 Male-informal < Female-polite          1
```

Based on the currently assumed model and data, we would conclude that hypotheses 1 and 3 are very likely true, but we can also see that there is quite a bit of uncertainty associated with hypothesis 2. The probability of H2 (given the data and the model) is only 0.849. So we should be sufficiently suspicious about H2 until more evidence is available.

It is generically reasonable to consider posteriors above 0.95 or 0.975 as large enough to warrant speaking of "evidence in favor of a hypothesis".

Priors

One important difference between frequentist and Bayesian inference are priors. Priors are pieces of information about our data that we assume before actually looking at them. Specifying priors has several advantages, both technical and conceptual. First, we can use **regularizing priors** to implement soft constraints on what counts as a plausible parameter setting for the model, thereby reducing the computational resources needed to estimate the model parameters. As a conceptual advantage, priors can also express relevant subjective prior beliefs about the situation or problem at hand. For example, pitch values (and many other things we measure in nature) cannot be smaller than zero. Human pitch values are also limited to a certain range defined by physiological and bio-mechanical constraints on our laryngeal system. In adults, values larger than, let's say, 1000 Hz are very unlikely.

Defining regularizing priors is essential for more complex models which have to estimate many parameters. Regularizing priors can help the model to converge more quickly.

But wait a minute. Subjective beliefs? This is science. We are supposed

to be objective, right? We should have a heart of stone and be skeptical about possible relationships in the first place. Practically, this means the following for us: We should not hesitate to make use of the possibility to bring all the potentially relevant background knowledge to bear when analyzing our data. The formalization of background knowledge should, of course, be made transparent and be explicitly justified. So, any pieces of reasonably uncontroversial background knowledge might well be included in the model. But, of course, when it comes to the specific hypotheses we would like to test, we should *not* engineer in the conclusions we would like to eventually draw, no matter what our subjective beliefs (possibly inspired by hope) are! If our research hypothesis is that female speakers lower their voices in polite contexts, we obviously don't want to specify a prior that assumes the hypothesized relationship before having seen the data. Instead, we can feed the model with a skeptical prior about the truth of our research hypothesis. Since there is probably little doubt about an effect of gender on voice pitch, the following explores what happens when we entertain a skeptical prior about the effect of context (on female speakers).

How does that look in practice? — First of all, you do not have to specify priors by hand; a call to `brm` will invoke generically defensible priors for the model. For example, to see the priors for the model fit we obtained above, which is stored in `model_FE`, we could type and inspect the automatically assumed priors like this:

```
> prior_summary(model_FE)
                  prior     class               coef
1 student_t(3, 204, 83) Intercept
2                               b
3                               b         contextpol
4                               b            genderM
5                               b genderM:contextpol
6   student_t(3, 0, 83)     sigma
```

This table tells us about `brm` default priors for the model. The intercept was sampled from a Student's *t* distribution with a mean at 204, a standard deviation of 83 and 3 degrees of freedom. These parameters have been determined for you behind the scenes by looking at the distribution of observed pitch values in the data set. Similarly, the prior for the standard deviation σ is also a Student's *t* distribution with suitable parameters determined from the data. But the column with information about the priors over coefficients ("class *b*" in the table above) is empty! That does not mean that these priors are a secret. It means that they have not been set. When we do not set a prior in a Bayesian model, software like *Stan*, which we use here in the background, assumes that any logically possible parameter value is equally likely. This is variably called an *unbiased prior*, a *flat prior* or a *maximum-entropy prior* because it encodes no bias in either direction at all.

Think of a Student's *t* distribution as a normal distribution with thicker tails, where the thickness is determined by the degrees of freedom.

Using unbiased priors for coefficients is a reasonable generic choice for `brm` because it cannot know which hypotheses you want to test. But, since

you (should) know which hypotheses you care about, you can be more explicit, and use the priors for conservative inference. So, let's define some priors for the model with fixed effects. Our goal is not to provide the "best" choice of prior here (which is highly debatable anyways), but to show one example of realizing a skeptical stance towards a given hypothesis. Here, we therefore specify the priors for the coefficient β_{pol} as normal distributions centered around zero with a rather small standard deviation of 20. This corresponds to the prior belief that there is likely no difference between (female speakers') voice pitch in informal and polite contexts.

You can specify priors for each class of parameters or every single parameter of the model individually. To see all parameters, classes and default priors *before* having to run a model, you can use the `get_prior()` function of `brms`, which takes a model formula as input. (For comparison, the function `prior_summary()` takes the fitted model as `brmsfit` object as input.)

```
priorFE <- c(
  # define skeptical prior for context effect on female speakers
  prior(normal(0, 10), coef = contextpol)
)
```

Notice that this only affects an effect of context of female speaker's voice pitch, as long as we allow for the other coefficients to "roam around freely", since effects of context on male speaker's pitch can be "accommodated" by the interaction term.

We add this prior to the model as an argument, like so:

```
model_FE_prior = brm(formula = pitch ~ gender * context,
  prior = priorFE,  # add prior
  data = politedata,
  control = list(adapt_delta = 0.99),
  seed = 1702
)
```

The argument 'control' here allows us to tweak the sampling procedure in order to ensure convergence. `brms` will encourage you to do so whenever you run into such convergence issues.

We then calculate the probability for our hypotheses again.

```
> get_posterior_beliefs_about_hypotheses(model_FE_prior)
# A tibble: 3 x 2
  hypothesis                      probability
  <chr>                                 <dbl>
1 Female-polite < Female-informal       0.951
2 Male-polite < Male-informal           0.834
3 Male-informal < Female-polite         1
```

We see that a very skeptical prior about H1 has decreased the posterior probability of the hypothesis being true. But nonetheless, the data still suggest that H1's probability is rather high. In this sense we can conclude that even if we were skeptical at the outset, the data should nonetheless convince us that H1 is true.

Model criticism

So we know now how to extract relevant comparisons to quantify the evidence surrounding our hypotheses. We have also learned how to specify prior information. The next important step is being able to check if the model really reflects the observed data. A common and easy way to answer this question is to use so-called **posterior predictive checks**. These checks generate new hypothetical data, sampled from the so-called **posterior predictive distribution**. This distribution quantifies how likely we would expect to see a particular outcome if the experiment were repeated in the same way (given our posterior beliefs about parameters). We can compare samples from the

posterior predictive distribution to the observed data. `brms` offers a neat function called `pp_check()` for this. If the simulated data diverges systematically from the observed data, we should be concerned. It would suggest that the model does not capture some of important properties of the data.

To illustrate this point, let's run the model from above without the gender predictor.

```
model_FE_noGender= brm(formula = pitch ~ context,
  prior = priorFE,
  data = politedata,
  control = list(adapt_delta = 0.99),
  seed = 1702
  )
```

We know that a big chunk of variation is accounted for by gender, so this model should be a poor fit. Figure 5 shows the output of the following function call:

```
pp_check(model_FE_noGender, nsample = 100)
```

If we run `pp_check()` on the model with and without gender (we also specify how many samples we want to compare the observations to), we can see that a model without `gender` (left panel) fails to capture an important property of the data: having pitch values from men and women leads to a bimodal distribution of pitch values (i.e. two bumps).

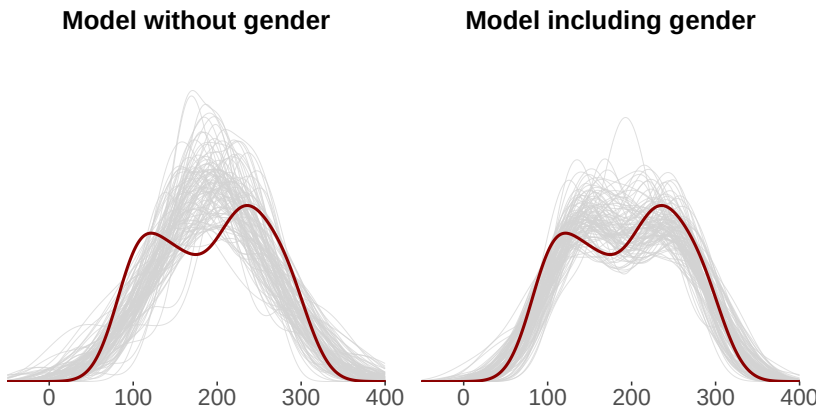

Figure 5: Output of the posterior predictive check for a model **without** the predictor `gender` in the left panel and a model **including** the predictor `gender` in the right panel. Red lines represent distribution of data, grey lines represent 100 posterior samples.

The model without gender overestimates the probability of very low and high pitch values, underestimates the probability of values that surround the two bumps and heavily overestimates the probability of values around 200 Hz. The earlier model which takes `gender` into account looks quite a bit better (right panel). While still showing much uncertainty surrounding the two bumps in the observed distribution, it clearly captures the evidence better.

The `pp_check()` can take additional arguments that allow you to display posterior predictive distributions for individual parameters

Adding random effects

In our experiment, we measured pitch multiple times for each subject (since they produced multiple sentences). We also have multiple measures for each sentence (as each sentence was produced by multiple speakers). The observant reader might have already noticed this, but it is probably worth reiterating that linear regression models make the crucial assumption that all samples (data points) are independent of each other. But if two data points are produced by the same participant, then observations will not necessarily be independent anymore. We need to inform the model about these dependencies between observations. The way we're going to handle this is to add random effects to the model, just as we do in the frequentist framework. Random effects are additional parameters that the Bayesian model estimates and that account for dependencies between data points. The choice of the random effect structure is controversial. This tutorial follows the approach advocated by Barr et al. (2013) to include the maximal random effect structure justified by the design (for a complementary view on random effect specifications, see Matuschek et al., 2017). For our case here, we estimate how individual sentences (`(1 | sentence)`) and individual subjects (`(1 | subject)`) differ in their overall pitch values (random intercepts). We also estimate how much pitch values across sentences vary according to the context they appear in and according to the gender of the speaker (as well as their interaction, i.e. `(0 + gender * context | sentence)`). This so-called by-sentence random slope allows the interaction term to vary between sentences. In other words, the impact of the predictors gender and context (and their interaction) might be different for different sentences. Finally, we estimate how much pitch values vary across subjects as a function of context (`(0 + context | subject)`). In other words, our model takes into account that the context dependent effect on pitch might differ between individual speakers.

Running hierarchical random effect models with brms is very similar to the look and feel of non-Bayesian approaches. Here is the function call to the model. The outcome of this model fit is shown below:

```
# model including random intercepts and random slopes
model_MaxRE = brm(formula = pitch ~ gender * context +
  (1 | sentence) + (1 | subject) +
  (0 + gender * context | sentence) +
  (0 + context | subject),
  data = politedata, control = list(adapt_delta = 0.99) )
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: pitch ~ gender * context + (1 + gender * context | sentence) + (1 + context | subject)
   Data: politedata (Number of observations: 83)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Group-Level Effects:
~sentence (Number of levels: 7)
                                  Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
sd(Intercept)                        21.60     10.11     6.94    47.11       1827 1.00
sd(genderM)                          10.65      8.59     0.41    30.40       2060 1.00
sd(contextpol)                       14.97     11.58     0.72    42.70       1526 1.00
sd(genderM:contextpol)               17.20     13.20     0.80    49.12       1785 1.00
cor(Intercept,genderM)               -0.22      0.44    -0.89     0.68       5035 1.00
cor(Intercept,contextpol)             0.02      0.42    -0.75     0.78       4760 1.00
cor(genderM,contextpol)              -0.06      0.45    -0.86     0.78       3019 1.00
cor(Intercept,genderM:contextpol)    -0.10      0.42    -0.81     0.73       4905 1.00
cor(genderM,genderM:contextpol)      -0.04      0.44    -0.83     0.78       3158 1.00
cor(contextpol,genderM:contextpol)   -0.15      0.44    -0.88     0.74       2897 1.00

~subject (Number of levels: 6)
                          Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
sd(Intercept)                35.63     17.94    14.55    80.85       1643 1.00
sd(contextpol)                9.38      8.99     0.33    33.61       2508 1.00
cor(Intercept,contextpol)     0.05      0.58    -0.94     0.97       4544 1.00

Population-Level Effects:
                   Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
Intercept            260.76     24.85   210.02   311.66       1557 1.00
genderM             -115.40     33.23  -181.79   -43.70       1438 1.00
contextpol           -27.53     12.96   -53.34    -2.40       2470 1.00
genderM:contextpol    16.21     16.90   -17.83    50.05       2441 1.00

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
sigma    25.01      2.37    20.88    30.09       3342 1.00

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
is a crude measure of effective sample size, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Lines 28–33 give the estimates of the fixed-effects coefficients. The mean estimates look very similar to the ones that we obtained in the fixed-effect only model. However, not surprisingly, our uncertainty surrounding these estimates is larger. We now also get information about the parameters implied by the specified random effect structure. Lines 8–20 cover the by-sentence random effects, lines 22–26 cover the by-subject random effects.

To check the probability of our hypotheses of interest, we can use the `faintr` package again, in conjunction with the convenience function for the specific hypotheses relevant for this tutorial:

```
> get_posterior_beliefs_about_hypotheses(model_MaxRE)
# A tibble: 3 x 2
  hypothesis                       probability
  <chr>                                  <dbl>
1 Female-polite < Female-informal        0.982
2 Male-polite < Male-informal            0.808
3 Male-informal < Female-polite          0.985
```

When we compare these values to the simpler model above, we can see that the evidence for our hypotheses seems a bit weaker for the maximal random effect model. This is not surprising because the simpler model assumed independence where there was none and therefore underestimated the variance. Speech is a messy aspect of human behavior. Speakers differ quite drastically in their pronunciation and even the same speaker varies a lot in how they speak. If we do not inform our model about these sources of variation, we might end up being overly confident in our results.

Reporting the results

How do we report our analysis and our results in a Bayesian framework? There is no gold standard. But the following is one example for how to do it.

Description of analysis. "We fitted Bayesian hierarchical linear models to pitch values as a function of dummy-coded factors GENDER (reference level "female"), and CONTEXT (reference level "informal") and their two-way interaction, using the Stan modeling language (Carpenter et al., 2016) and the package *brms* (Buerkner, 2016). The models included maximal random-effect structures justified by the design, allowing the predictors of interest and their interactions to vary by participants (CONTEXT) and sentences (GENDER, CONTEXT). We used the default priors of the `brms` package, namely a Student's t-distribution ($\nu = 3, \mu = 204, \sigma = 83$) for the mean of the reference cell (female speakers in an informal context), a Student's t-distribution ($\nu = 3, \mu = 0, \sigma = 83$) for standard deviation for the likelihood function, and unbiased priors for regression coefficients. We used the *brms* package's default priors for standard deviations of random effects (a Student's t-distribution with $\nu = 3, \mu = 0$ and $\sigma = 20$), as well as for correlation coefficients in interaction models (LKJ $\eta = 1$).

Four sampling chains ran for 2000 iterations with a warm-up period of 1000 iterations for each model, thereby yielding 4000 samples for each parameter tuple. For all relevant cell means and differences between them, we report the expected values under the posterior distribution and their 95% credible intervals (CIs). For differences between cells, we also report the posterior probability that a difference δ is bigger than zero. If a hypothesis states that $\delta > 0$, we judge there to be *compelling evidence* for this hypothesis if zero is (by a reasonably clear margin) not included in the 95% CI of δ and the posterior $P(\delta > 0)$ is close to one.

Description of results. Female speakers produced higher voice pitch in informal contexts ($\mathbb{E}(\mu_{\text{fem, inf}}) = 261$, CI = [210, 312]) than in polite contexts ($\mathbb{E}(\mu_{\text{fem, pol}}) = 233$, CI = [179, 288]). There is compelling evidence for this difference ($\mathbb{E}(\mu_{\text{fem, inf}} - \mu_{\text{fem, pol}}) = 28$, CI = [2, 53], $P(\delta > 0) = 0.982$). We conclude that the data and the model support H1.

Male speakers also produced higher voice pitch in informal contexts ($\mathbb{E}(\mu_{\text{male, inf}}) = 145$, CI = [94, 191]) than in polite contexts ($\mathbb{E}(\mu_{\text{male, pol}}) = 134$, CI = [77, 184]). However, there is no sufficient evidence that this difference is larger than zero ($\mathbb{E}(\mu_{\text{male, inf}} - \mu_{\text{male, pol}}) = 11$, CI = [-16, 40], $P(\delta > 0) = 0.809$). We conclude that the data and the model do not support H2.

Female speakers in polite contexts produced higher voice pitch ($\mathbb{E}(\mu_{\text{fem, pol}}) = 233$, CI = $[179, 288]$) than male speakers in informal contexts ($\mathbb{E}(\mu_{\text{male, inf}}) = 145$, CI = $[94, 191]$). There is compelling evidence that the difference between these cells is larger than zero ($\mathbb{E}(\mu_{\text{fem, pol}} - \mu_{\text{male, inf}}) = 86$, CI = [18, 162], $P(\delta > 0) = 0.985$). We conclude that the data and the model support H3."

Notation $\mathbb{E}(\cdot)$ is shorthand for the expectation (mean) of the posterior distribution of some (possibly derived) quantity of interest.

Other forms of Bayesian inference

Alright, you made it! We have learned the basic reasoning behind Bayesian parameter estimation, we learned about priors and how to specify them, we learned how to criticize our models and how to write up our results. Phewww, if you feel a little overwhelmed by now, don't worry, it will come to you. Before we send you off into the Bayesian data analysis world, we would like to mention a couple of things that we were not able to cover here.

There are three major things we can do with models and data in Bayesian data analysis, two of which we have already seen:

1. **parameter inference**: we consider one model, (tentatively) assume that this model is true, and ask what we should believe about its parameters (given the data);
2. **model criticism**: we consider one model and ask whether this model is sufficient to deal with the data at hand; we can do that before showing the model any of the data, or after feeding it with some or all of the data;
3. **model comparison** we consider at least two models and we ask which of these models (plural!) is better at explaining the data.

So far, we have dealt with parameter inference and model criticism. In fact, we used parameter inference to test our research hypotheses, and we used model criticism as a sanity check to make sure that the one model we used was adequate.

This tutorial focused on parameter estimation because it's simple and very familiar to those who have used null hypothesis significance testing in

a frequentist framework before. But if you dive deeper into Bayesian data analysis, you will quickly discover that there are good arguments (e.g. Vandekerckhove, Matzke, and Wagenmakers, 2015) for an approach to hypothesis testing that is based on Bayesian model comparison in terms of so-called **Bayes factors** (Jeffreys, 1961; Kass and Raftery, 1995) or cross-validation like **Leave-One-Out (LOO)** (Vehtari, Gelman, and Gabry, 2016). These techniques lead us too deep into the rabbit hole of Bayesian inference for today, but we can offer some (hopefully) helpful departure points for further reading.

Further reading

The textbook by Gelman et al. (2014) is a standard reference for Bayesian data analysis. Kruschke (2015) provides a less technical introduction to Bayesian methods. McElreath (2016) is both an excellent introduction to core concepts of statistical analysis and and excellent point of departure into Bayesian analyses. . There are many other useful tutorials to Bayesian analysis using *brms*, some of which are technically a little bit more involved (e.g. Sorensen, Hohensteinb, and Vasishth, 2016). If you want to explore Bayesian approaches to data analysis with a graphical user interface, check out JASP: `https://jasp-stats.org`.

A. Solomon Kurz has an online book based on McElreath (2016), which uses the same tools as we did here, namely the *tidyverse* and the *brms* packages: `https://bookdown.org/ajkurz/Statistical_Rethinking_recoded/`

References

Barr, Dale J et al. (2013). "Random effects structure for confirmatory hypothesis testing: Keep it maximal". In: *Journal of memory and language* 68.3, pp. 255–278.

Buerkner, Paul-Christian (2016). "brms: An R package for Bayesian multilevel models using Stan". In: *Journal of Statistical Software* 80.1, pp. 1–28.

Carpenter, Bob et al. (2016). "Stan: A probabilistic programming language". In: *Journal of Statistical Software* 20, pp. 1–37.

Gelman, Andrew et al. (2014). *Bayesian Data Analysis*. 3rd edition. Boca Raton: Chapman and Hall.

Jeffreys, Harold (1961). *Theory of Probability*. 3rd. Oxford: Oxford University Press.

Kass, Robert E. and Adrian E. Raftery (1995). "Bayes Factors". In: *Journal of the American Statistical Association* 90.430, pp. 773–795.

Kruschke, John E. (2015). *Doing Bayesian Data Analysis*. 2nd edition. Burlington, MA: Academic Press.

Matuschek, Hannes et al. (2017). "Balancing Type I error and power in linear mixed models". In: *Journal of Memory and Language* 94, pp. 305–315.

McElreath, Richard (2016). *Statistical Rethinking*. Boca Raton: Chapman and Hall.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL: https://www.R-project.org/.

Sorensen, Tanner, Sven Hohensteinb, and Shravan Vasishth (2016). "Bayesian linear mixed models using Stan: A tutorial for psychologists, linguists, and cognitive scientists". In: *The Quantitative Methods for Psychology*.

Vandekerckhove, Joachim, Dora Matzke, and Eric-Jan Wagenmakers (2015). "Model Comparison and the Principle of Parsimony". In: *Oxford Handbook of Computational and Mathematical Psychology*. Ed. by J. Busemeyer et al. Oxford: Oxford University Press, pp. 300–319.

Vehtari, Aki, Andrew Gelman, and Jonah Gabry (2016). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC". Manuscript.

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

Winter, Bodo and S. Grawunder (2012). "The Phonetic Profile of Korean Formality". In: *Journal of Phonetics* 40, pp. 808–815.