Received: 24 May 2021 | Revised: 26 September 2021 | Accepted: 1 October 2021

DOI: 10.1111/lnc3.12439

REVIEW ARTICLE

WILEY

Poisson regression for linguists: A tutorial introduction to modelling count data with brms

Bodo Winter[1] | **Paul-Christian Bürkner**[2]

[1]Department of English Language and Linguistics, University of Birmingham, Birmingham, UK

[2]Cluster of Excellence SimTech, University of Stuttgart, Stuttgart, Germany

Correspondence
Bodo Winter, Department of English Language and Linguistics, University of Birmingham, Frankland Bldg, Edgbaston, Birmingham B15 2TT, UK.
Email: bodo@bodowinter.com

Funding information
UKRI Future Leaders Fellowship, Grant/Award Number: MR/T040505/1; Deutsche Forschungsgemeinschaft, Grant/Award Number: EXC 2075 - 390740016

Abstract

Count data is prevalent in many different areas of linguistics, such as when counting words, syntactic constructions, discourse particles, case markers, or speech errors. The Poisson distribution is the canonical distribution for characterising count data with no or unknown upper bound. Given the prevalence of count data in linguistics, Poisson regression has wide utility no matter what subfield of linguistics is considered. However, in contrast to logistic regression, Poisson regression is surprisingly little known. Here, we make a case for why linguists need to consider Poisson regression, and give recommendations for when Poisson regression is more appropriate compared to logistic regression. This tutorial introduces readers to foundational concepts needed to understand the basics of Poisson regression, followed by a hands-on tutorial using the R package `brms`. We discuss a dataset where Catalan and Korean speakers change the frequency of their co-speech gestures as a function of politeness contexts. This dataset also involves exposure variables (the incorporation of time to deal with unequal intervals) and overdispersion (excess variance). Altogether, we hope that more linguists will consider Poisson regression for the analysis of count data.

1 | INTRODUCTION

Counting the frequency of discrete events is fundamental to much research in linguistics. Examples include counting the number of pronouns in different referential contexts, the number of speech errors as a function of language disorders, or the number of co-speech gestures in different social contexts. Count data is also of pivotal importance to corpus linguistics, where the analysis of word frequencies takes centre stage. Poisson regression, a specific instance of the class of generalised linear models (GLMs), is the canonical way of analysing count data with no or no known upper bound in many different fields outside of linguistics (McElreath, 2020; Zuur et al., 2009). Surprisingly, despite the ubiquity of count data in linguistics, Poisson regression is used only very little, and most statistics textbooks targeted at linguists do not even mention the approach (Brezina, 2018; Desagulier, 2017; Gries, 2017; Levshina, 2015; Wallis, 2021). Here, we make a case for incorporating Poisson regression more strongly into the statistical analysis toolkit of linguists.

The field of linguistics is rapidly moving away from traditional significance tests towards approaches that emphasise statistical modelling (rather than testing) and parameter estimation. In particular, arguments for linear mixed effects models, also discussed under the terms of 'hierarchical' or 'multilevel' models, have been made in psycholinguistics (Baayen et al., 2008), sociolinguistics (Johnson, 2009; Tagliamonte & Baayen, 2012), corpus linguistics (Gries, 2015), phonetics (Kirby & Sonderegger, 2018; Vasishth et al., 2018), and typology (Jaeger et al., 2011), among others (for an overview, see Winter & Grice, 2021). Poisson regression fits naturally within this trend because Poisson models can also be fitted as mixed models, thereby allowing researchers to model count data while also incorporating complex dependency structures into the analysis.

After a brief primer on GLMs and Poisson regression (Section 2), we guide the reader step-by-step through a mixed Poisson regression analysis implemented in R (R Core Team, 2019) and the package `brms` (Bürkner, 2017) (Section 3), which the reader can follow alongside a full data analysis script that is available in the accompanied Open Science Framework repository (https://osf.io/ugpfd/). We conclude by discussing additional issues that may come up in the analysis of count data, as well as discussing frequently asked questions (Section 4).

2 | CONCEPTUAL INTRODUCTION TO POISSON REGRESSION

We begin by considering the Poisson distribution. In contrast to the normal or Gaussian distribution, which has two parameters (a mean μ 'mu' and a standard deviation σ 'sigma'), the Poisson distribution has only one parameter, λ 'lambda'. This parameter specifies the mean and variation (see below) of a count process. Lambda can be thought of as being analogous to the mean of the normal distribution, as it describes the mean number of occurrences of discrete events. So, the mean of any variable for which a researcher is counting discrete units (e.g., words, sentences, grammatical markers, gestures, and so on) is principally amenable to being modelled in terms of the parameter lambda (see Section 4.2 for cases that are more appropriate for logistic regression).

Figure 1 visualises the Poisson distribution for two different lambda values, with the x-axis indicating the possible outcomes (discrete integer counts), and the y-axis indicating the expected probability of each discrete count.

The normal distribution has the standard deviation σ as a parameter, modelling the variance or 'spread' of this distribution. In contrast, the Poisson lacks a separate parameter for the variance. Instead, the variance is tied to the mean λ, with the assumption that they are equal. The

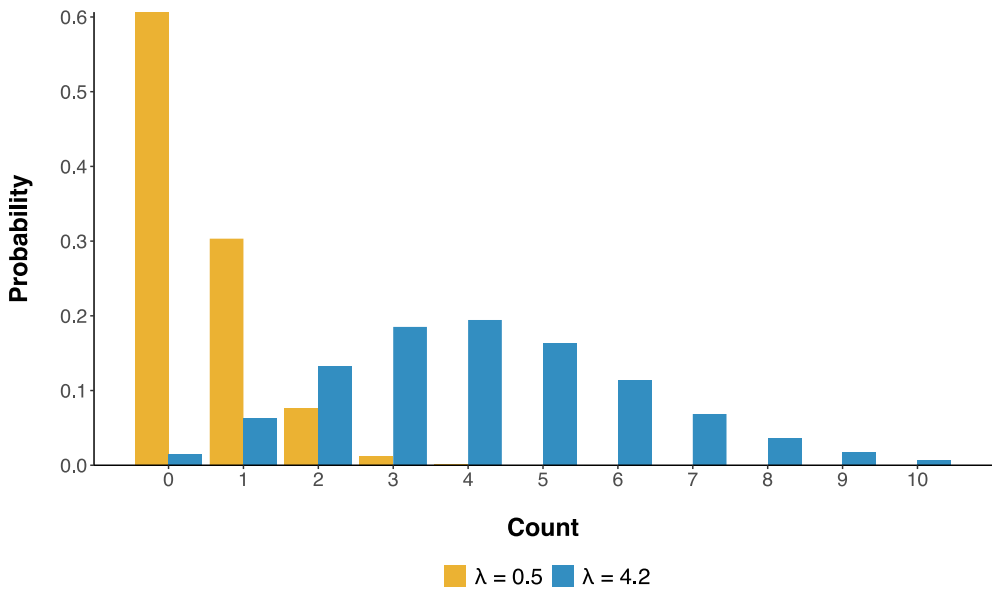

FIGURE 1 Counts with their associated probabilities expected under the Poisson distribution for two different values of lambda; yellow: $\lambda = 0.5$; blue: $\lambda = 4.2$

dependence of mean and variance is also apparent in Figure 1, where the blue distribution is more fanned out than the yellow one. Because counts are bound by zero, counts with very low rates scrunch up close to zero. Higher counts go along with more variance because there is 'space' for the probability to be spread across high counts as well as low counts.

While the Poisson distribution may visually appear quite similar to the normal distribution for high lambda values, this similarity is deceiving. The distributions fundamentally differ in that the normal distribution is a continuous one, whereas the Poisson distribution is discrete. That is, whereas the normal distribution allows values in between positive integers (e.g., values in between 1 and 2), this is not the case for the Poisson. If one were to use the normal distribution for discrete count data, the data-generating process would not be correctly identified because the normal distribution can generate values that could not possibly be counts. In the example above, we explicitly chose lambda values with decimals to underscore the point that in contrast to the discrete counts, lambda is a continuous parameter.

In this tutorial, we demonstrate the utility of Poisson regression with a study that seeks to analyse the co-speech gestures used by Catalan and Korean speakers. The data comes from a dyadic task performed by Brown et al. (in press) in which participants first watched a cartoon and subsequently told a partner about what they had seen. The research question was whether the social context modulates people's politeness strategies, including nonverbal politeness strategies, such as changing the frequency of one's co-speech gestures. The key experimental manipulation was whether the partner was a friend or a confederate, who was an elderly professor. We ignore language differences for the time being and focus on the condition manipulation.

Framed in terms of the Poisson distribution, the goal of the analysis of count data is to estimate the parameter λ from the data. In the case of the multimodal data from Brown et al. (in press), λ specifies the mean number of gestures, and the goal is to use regression to see to what extent this number differs as a function of two conditions (friend/professor). We first begin by considering Gaussian linear models before moving into Poisson models. Linear models are a

general approach to data analysis that can cover many of the cases commonly dealt with by run-of-the-mill significance tests such as *t*-tests, ANOVAs, ANCOVAs etc., but they have the clear advantage of emphasising parameter estimation over testing, and they are a more flexible approach to data analysis that allows building more complex, theory-guided models.

A linear model involves a response variable (y) conditioned on a predictor variable (x):

$$\begin{aligned} y_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha + \beta x_i \end{aligned} \tag{1}$$

The first statement specifies that y values are assumed to be generated by draws from a normal distribution with a specified mean and standard deviation. The mean μ_i bears the subindex i, which is an index variable for the ith data point, $i = 1$, $i = 2$ etc. This subindex expresses the fact that the parameter of interest changes as a function of what data point is considered, such as depending on the value of a predictor (e.g., condition A vs. condition B). The function by which the mean varies is $\mu_i = \alpha + \beta x_i$, corresponding to the equation of a line, involving a linear combination of an intercept α and a slope β that is multiplied with a predictor x. In the context of GLMs, this right-hand side is called the linear predictor. The slope of the linear predictor specifies how much the mean μ changes as a function of one or more predictors, x. In many practical applications, researchers are primarily interested in interpreting these slopes, which specify the relation between the parameter of interest (μ) and the predictor variables, such as how response times, ratings, or voice pitch measurements differ as a function of a condition predictor.

Besides Gaussian linear regression, logistic regression and Poisson regression are two common versions of the GLM, among many others. Figure 2 shows the parallel architecture of the three most canonical GLMs. In each case, there is a linear predictor ($\alpha + \beta x_i$) that predicts a parameter of interest: the mean μ of the normal distribution, the probability p of an event occurring of the Bernoulli distribution,[1] and finally, λ of the Poisson distribution, the mean of a count process.

A key difference between standard linear regression and other GLMs is that logistic regression and Poisson regression need an extra step to ensure that the linear predictor $\alpha + \beta x_i$ predicts only sensible parameter values. The issue is that the parameter p of the Bernoulli/binomial distribution is a probability, so it has to be bounded by [0, 1]. Likewise, the parameter λ of the Poisson distribution has to be positive as the mean of a count process cannot sensibly be negative. Because the output of $\alpha + \beta x_i$ can be any value ranging from negative to positive infinity, a 'link' function is introduced to ensure that only sensible parameter values are allowed. In the case of logistic regression, the logistic function ensures that values from negative to positive infinity are compressed into the range [0, 1]. In the case of Poisson regression, the exponential function is introduced to ensure that the output of $\alpha + \beta x_i$ is always positive (exponentiating any number always yields a positive number).

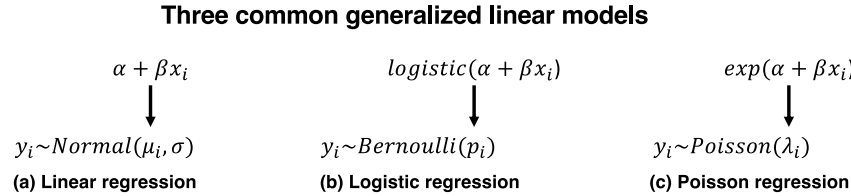

FIGURE 2 Overview of three common generalised linear models

We can then formulate a simple Poisson regression model with one predictor (x) as follows:

$$\begin{aligned} y_i &\sim \text{Poisson}(\lambda_i) \\ \lambda_i &= \exp(\alpha + \beta x_i) \end{aligned} \tag{2}$$

Notice that the y's are draws from the Poisson distribution, that is, they will always be positive integers. In contrast, the parameter lambda is continuous-valued and positive.

To interpret the intercepts and slopes of this model more directly, a more useful representation of the above model 'frees' the linear predictor from the exponential function. This can be done by logarithmically transforming both sides of the equation (as the logarithmic is the inverse of the exponential function), which results in the following representation of our Poisson regression model:

$$\begin{aligned} y_i &\sim \text{Poisson}(\lambda_i) \\ \log(\lambda_i) &= \alpha + \beta x_i \end{aligned} \tag{3}$$

In the case of the multimodal politeness example, the predictor x specifies the main condition variable—whether the participant is speaking to a friend or a professor. The slope β then describes the difference between these two conditions. As we discuss below, it is important to keep in mind that in the context of Poisson regression, the slope term β does not describe the raw difference in gesture counts between conditions, but the log difference. This means that special care needs to be taken when interpreting the output of Poisson regression models, which we discuss in more detail in the following hands-on tutorial.

3 | A HANDS-ON POISSON REGRESSION ANALYSIS EXAMPLE

3.1 | Software and Bayesian modelling

The following hands-on example uses the `brms` package (Bürkner, 2017), which stands for 'Bayesian regression models using Stan', to implement a mixed Poisson regression model of co-speech gesture data. 'Stan' is a probabilistic programming language specifically designed for Bayesian data analysis (Stan Development Team, 2021). `brms` acts as an R interface with Stan. Its syntax was inspired by the widely used `lme4` package (Bates et al., 2015). Users familiar with fitting mixed effects models with the `lme4` package can thus easily switch to fitting the corresponding Bayesian mixed effects models.

There are several reasons for us to use `brms` rather than `lme4` for this tutorial. One practical reason is that `brms` gives more flexibility in the types of distributions and model types that can be used (see Bürkner, 2017, tab. 1), as will become clear in the hands-on tutorial below. Another practical reason is that complex random effects structures that are often required for linguistic data analysis often do not, or not easily, converge with `lme4`. In contrast, complex random effects structures are more likely to converge when fitting Bayesian models (Eager & Roy, 2017; Kimball et al., 2019; Sorensen & Vasishth, 2015).

On top of these purely practical considerations, there are many conceptual reasons to prefer Bayesian models over the corresponding frequentist models, such as implemented in `lme4`. In particular, the results of analyses involving p-values are widely misinterpreted (Haller & Krauss, 2002; Schneider, 2015), including by experienced statisticians (Lecoutre et al., 2003; McShane & Gal, 2017). In fact, the notion of significance is so counterintuitive that the majority of introductory textbooks explain it incorrectly (Cassidy et al., 2019). As people naturally veer towards a Bayesian interpretation of frequentist statistics (e.g., Dienes, 2011; Lecoutre

et al., 2001; Morey et al., 2016), more and more researchers call for using those methods that actually correspond to people's natural intuitions. Bayesian modelling also allows the incorporation of prior knowledge via the specification of prior distributions, which has several advantages for more theory-guided statistical modelling. Among other things, it allows specifying models in such a way that inferences are more conservative than the corresponding frequentist models (cf. Lemoine, 2019; McElreath, 2020), as we also detail below.

While this tutorial uses Bayesian models, the reader does not need to be familiar with Bayesian modelling until the section that deals with prior specification. Luckily, as Bayesian methods are increasingly gaining traction in linguistics, there are by now numerous tutorials available. For an accessible conceptual introduction, see Nicenboim and Vasishth (2016). For an easy introduction to analysing linguistic data with mixed effects models and `brms`, see Franke and Roettger (2019). More advanced introductions with hands-on examples include Vasishth et al. (2018) and Nalborczyk et al. (2019). For excellent conceptual introductions to Bayesian statistics with a psychological orientation, see Dienes (2008) and Kruschke and Liddell (2018). For a full course in Bayesian statistics, we highly recommend McElreath's book-length treatment (2020).

3.2 | A simple Poisson regression model

The data for the following analysis is accessible here: https://osf.io/ugpfd/. To begin our analysis, we first load the tidyverse package for data processing (Wickham et al., 2019), as well as `brms`. We consistently use tidyverse-style code throughout this tutorial, for which Wickham and Grolemund (2016) give an excellent book-length introduction.

In the following, code in bold face is user input.

```
library(tidyverse)
library(brms)
```

Next, we load the data from a comma-separated spreadsheet and assign it to a tibble called `'dyads'`.

```
dyads <- read_csv('dyads.csv')
```

We can use `sample_n()` to print four random rows to the console to get a first impression of this dataset:

```
sample_n(dyads, 4)
# A tibble: 4 × 6
ID           context    dur       language    gender     gestures
<chr>        <chr>      <dbl>     <chr>       <chr>      <dbl>
1 Catalan_2    prof       107       Catalan     F          40
2 Korean_6     friend     87        Korean      F          26
3 Catalan_1    friend     137       Catalan     M          61
4 Korean_3     prof       138       Korean      F          49
```

There are two data points per participant, one from the friend condition, and one from the professor condition. The `'ID'` column lists participant identifiers for all 27 participants (14

Catalan speakers and 13 Korean speakers). The `'gestures'` column contains the primary response variable that we are trying to model, the number of gestures observed on each trial. The `'context'` predictor specifies the social context that was experimentally manipulated.

In the following, we start with a simple Poisson regression model to explain certain aspects of GLM before proceeding to mixed models. This progression is for pedagogical purposes only, and we do not wish to imply that researchers should explore a whole range of models. The reader should consider, as much as possible, what model may be appropriate prior to beginning the analysis. The process of expanding a model should be guided by posterior predictive simulations (Gelman & Shalizi, 2013; Kruschke, 2013), which we detail below.

For our first and most simple model, we can condition the count of gestures on the `'condition'` predictor with a Poisson regression model using the `brm()` function as follows:

```
mdl <- brm(gestures ~ 1 + condition, data = dyads,
           family = poisson)
```

The `'1'` in the above function call corresponds to the intercept (R uses the number 1 as a default placeholder for intercepts in all linear model formulas), and the `'condition'` term corresponds to the predictor variable. It is possible to omit the intercept term and use the model formula `'gestures ~ condition'`, for which the intercept is fitted automatically. However, the above notation is more explicit, making it clearer that there are two terms in this equation.

Typing `mdl` into the console returns a summary output, of which we show here an abbreviated coefficient table. This coefficient table is based on the posterior distribution, the principal outcome of fitting a Bayesian model. Each estimated parameter is associated with such a distribution. This posterior distribution is a probability distribution that indicates how probable particular parameter values are, given the model specification (including the priors), and given the evidence from the data. For example, the posterior distribution for the slope coefficient tells the analyst which coefficient values are more or less probable. The coefficient table below lists the posterior mean (point estimate) of each term and the associated standard error, which is the standard deviation of the posterior distribution. The output also shows the 95% credible interval of each coefficient. In Bayesian statistics, credible intervals indicate the range within which a parameter value falls with a particular probability. More narrow credible intervals indicate higher precision in one's estimate.

```
mdl
```

```
             Estimate  Est.Error  l-95% CI  u-95% CI
Intercept        3.99       0.03      3.93      4.04
contextprof     -0.18       0.04     -0.26     -0.10
```

We can combine the estimates for the intercept and slope together in the following equation of the mean (log) number of gestures in each condition:

$$\log(\lambda) = 3.99 - 0.18 * context \tag{4}$$

R uses treatment coding by default to 'dummy code' categorical predictors into 0's and 1's. For this, whatever comes first in the alphabet will be assigned 0, the reference level. For the categorical variable `'context'`, R will assign 0 to the `'friend'` level and 1 to the `'prof'` level. The negative slope `(-0.18)` therefore means that the number of gestures is estimated to be *less* in the professor condition.

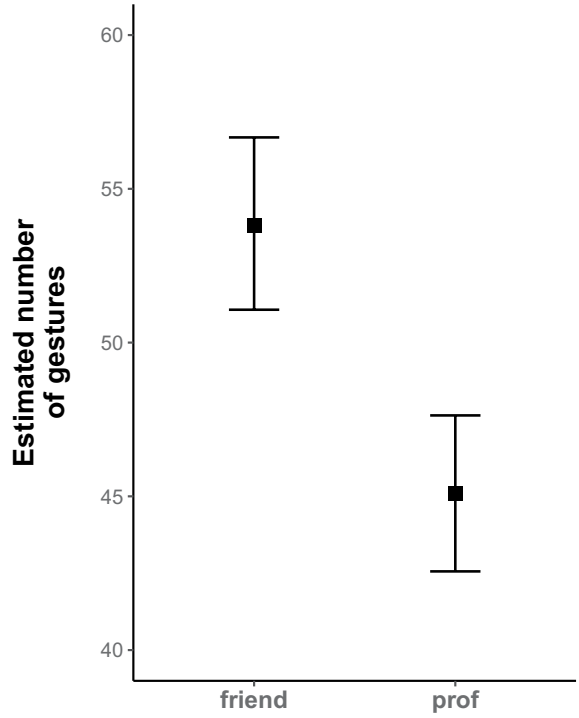

FIGURE 3 Conditional effects plot of the Poisson regression model; the error bars display 95% credible intervals; the dots represent posterior medians

The `brms` package has several functions that make it possible to interpret the model's prediction on the original count scale. One of these functions is `conditional_effects()`, which can simply be wrapped around the model object as follows, which yields Figure 3.

```
conditional_effects(mdl)
```

In addition, the `hypothesis()` function provides a generic framework for using the model to evaluate specific hypotheses. This function also makes it possible to compute values on the original count scale from the model. The function takes two arguments: the model object and a character vector of the hypothesis to be evaluated. The following two lines of code first define a character vector `h` specifying the hypothesis to be evaluated. Then this vector is given to the `hypothesis()` function together with the model to evaluate the specified hypothesis.

```
h <- 'exp(Intercept + contextprof * 1) = exp(Intercept + contextprof * 0)'
hypothesis(mdl, h)
Hypothesis Tests for class b:

                Hypothesis  Estimate  Est.Error  CI.Lower  CI.Upper
1 (exp(Intercept+co... = 0     -8.76       1.92    -12.64     -5.02
```

For this particular hypothesis, we wrapped the linear predictor '$\alpha + \beta$' in the `exp()` function to indicate that we want the hypothesis function to report values on the original count scale, rather than logged values. This effectively undoes the effects of the log link function. We plug

in the values 0 (= friend) and 1 (= professor) of the dummy-coded categorical predictor. Thus, the left-hand side `'exp(Intercept + contextprof * 1)'` corresponds to the prediction for the professor condition; the right-hand side `'exp(Intercept + contextprof * 1)'` corresponds to the prediction for the friend condition. The results (abbreviated output shown) indicate that the raw difference between the condition is 8.76 gestures, with a credible interval spanning from 5.02 to 12.64.

3.3 | Extending the Poisson regression model with random effects

The above model does not account for the fact that there are repeated observations for speakers. Different observations from the same individual cannot be treated as independent data points, which warrants the inclusion of a speaker random effect,[2] which is done in the following function call:

```
mdl <- brm(gestures ~ 1 + context + (1|ID),
           data = dyads, family = poisson)
```

The notations for `'(1|ID)'` follow the general `lme4` notation for random effects (Bates et al., 2015; Bürkner, 2018). This model adds one random intercept term to the model. In contrast to the previous model without this term, this model (which is now a *mixed* Poisson regression model) does not assume that all participants have the same intercepts, allowing for some participants to have higher/lower intercepts (= higher/lower gesture counts) than others. As a result of adding this random effect, the output now lists a standard deviation under section `'Group-Level Effects'`, where random effects are listed. This standard deviation is an estimate of by-participant variation in intercepts.

```
mdl
Group-Level Effects:
~ID (Number of levels: 27)

               Estimate Est.Error l-95% CI u-95% CI
sd(Intercept)      0.50      0.08     0.37     0.68
```

While intercepts vary by participants within this random intercept-only model, the degree to which participants respond to the condition manipulation is assumed to be fixed, that is, all participants change the rate of co-speech gestures the same way when speaking to the professor. This is clearly an unrealistic assumption to make as social contexts can be expected to have different effects on different people. It seems highly probable that some people would change gestures more than others when speaking with different partners, which warrants the inclusion of random slopes. The next model incorporates by-participant variation in the condition manipulation via the `'(1 + context|ID)'` term:

```
mdl <- brm(gestures ~ 1 + context + (1 + context|ID),
           data = dyads, family = poisson)
```

This random slope model now has additional random effects terms listed in the output:

```
~ID (Number of levels: 27)
                              Estimate Est.Error l-95% CI u-95% CI
sd(Intercept)                     0.46      0.08     0.34     0.63
sd(contextprof)                   0.29      0.07     0.16     0.45
cor(Intercept, contextprof)       0.26      0.25    -0.26     0.70
```

The first new term `'sd(contextprof)'` is an estimate of the standard deviation across condition slopes, thereby showing how much participants' mean number of gestures differ in response to social context. The third term is a random intercept/slope correlation term that can be interpreted like Pearson's *r*. In this case, this term estimates the degree to which participant variation in intercepts is associated with participant variation in slopes. A large intercept/slope correlation would indicate that people who produce overall more gestures (= higher intercepts) also show bigger differences between the friend and professor conditions (with the sign indicating the direction of the difference). However, in this case, the 95% credible interval of this parameter is quite wide [−0.26, 0.70], suggesting that there is no convincing evidence for a slope/intercept correlation with this data. Importantly, in what follows, we will stick to the random slope model, which not only is motivated on theoretical grounds (it makes sense to assume that people could differ in how social context affects gesture rate), but also because models without important random slopes terms can lead to severely anti-conservative estimates of fixed effects terms (Aarts et al., 2015; Barr et al., 2013; Schielzeth & Forstmeier, 2008).

3.4 | Incorporating exposure information with offset terms

The task used by Brown et al. (in press) allowed participants to talk with their partner for as long as they wished. This means that trials had differing durations. When analysing count data, it is crucial to account for unequal intervals, given that counting for longer intervals is generally expected to lead to higher counts. Time can be incorporated into a Poisson regression model as an 'exposure variable'. To incorporate time into the model, we can think of lambda as being the ratio of the average counts μ over time, for which we use the symbol τ 'tau' in the equation below.

$$\log\left(\lambda\right) = \log\left(\frac{\mu}{\tau}\right) \tag{5}$$

Replacing $\log\left(\lambda\right)$ in Equation (3) with $\log\left(\mu / \tau\right)$ and rearranging[3] yields the following model with exposure term (random effects omitted for ease of presentation).

$$\log\left(\mu\right) = \log\left(\tau\right) + \alpha + \beta x \tag{6}$$

Thus, the log time variable is a term on the right-hand side of the equation, but in contrast to the predictors, the exposure variable is associated with a regression coefficient that is fixed to 1. In `brms`, adding an exposure variable involves adding an `'offset()'` term as follows:

```
mdl <- brm(count ~ context + offset(log(dur)) +
           (1 + context|ID),
           data = dyads, family = poisson)
```

A look at the coefficients shows that the resultant numbers have drastically changed. For example, the intercept was previously `-3.99` and is now `-0.97`. This change in value has to do with the fact that the model with an exposure term models a different quantity, not the counts but the counts over units of exposure. That is, in this case, the model with the time offset variable models *gestures per second*, rather than just the average count of gestures.

```
mdl
```

```
             Estimate  Est.Error  l-95% CI  u-95% CI
Intercept       -0.97       0.07     -1.12     -0.82
contextprof     -0.12       0.06     -0.23     -0.01
```

To calculate the predicted rate of gestures (rather than the log rate), we can again use the `hypothesis()` function. To demonstrate the flexibility of this function, this time around we calculate the values separately for each condition, which requires specifying separate 'hypotheses' for the friend and professor conditions. Notice that we have to introduce some sort of equality (in this case = 0) for the function to work, as it expects input that can be interpreted as a hypothesis to be tested. The output (not shown here) reveals that the estimated rate of gestures in the friend condition is 0.38, in contrast to 0.34 for the professor condition. Thus, there was on average about 1 gesture every three seconds in the professor condition, and slightly more than that in the friend condition. These numbers are much lower than the values shown in Figure 3 because they now correspond to rates of gestures rather than counts of gestures.

```
friend_h <- 'exp(Intercept + contextprof * 0) = 0'
hypothesis(exposure_mdl, friend_h)

prof_h <- 'exp(Intercept + contextprof * 1) = 0'
hypothesis(exposure_mdl, prof_h)
```

3.5 | Dealing with overdispersion with negative binomial regression

As mentioned above, for the Poisson distribution, the mean is equal to the variance. In the context of the Poisson distribution, the variance is also called 'dispersion'. When the variance of counts exceeds what is expected under the mean, we speak of 'overdispersion'. To estimate whether there is excess variance, we can switch the above model to what is variously either called the 'negative binomial distribution' or the 'gamma-Poisson distribution'. The key difference between the negative binomial distribution and the Poisson distribution is the inclusion of an additional parameter, often symbolised as ϕ 'phi' or in `brms` called 'shape', which directly estimates dispersion from the data. Making the dispersion ϕ a parameter that is estimated from the data relaxes the 'mean = variance' assumption of standard Poisson regression.

To implement this in `brms`, all that is needed is to switch the `'family = poisson'` argument to `'family = negbinomial'`. For negative binomial regression, the shape parameter which characterises overdispersion (discussed below) can also scale with the rate/exposure. Because of this, the syntax for the exposure variable is different from the case of Poisson model above.

```
negbinom_mdl <- brm(gestures | rate(dur) ~ 1 + context +
(1 + context|ID),
data = dyads,
family = negbinomial)
```

The output now contains an additional line with an estimate of the shape parameter.

```
Family Specific Parameters:
        Estimate  Est.Error  l-95% CI  u-95% CI
shape      19.45      40.08      0.30    138.68
```

The less overdispersion there is in the data, the higher the shape parameter. Conversely, smaller shape parameters correspond to more overdispersion. To gain an intuition about this parameter, Figure 4 shows the negative binomial distribution for a mean of 4 with two different shape parameters; shape = 1 and shape = 100. The higher shape parameter creates a distribution that looks almost indistinguishable from the Poisson distribution shown in Figure 1 (blue). In the limit, as the shape parameter moves closer to infinity, the negative binomial distribution converges on the Poisson distribution.

As the standard link function for negative binomial models is the same as the one for Poisson regression (the log link function), the model coefficients can be interpreted the same way.

```
            Estimate  Est.Error  l-95% CI  u-95% CI
Intercept      -0.97       0.08     -1.12     -0.82
contextprof    -0.12       0.06     -0.23     -0.01
```

To assess whether there is enough overdispersion in the data to warrant moving from Poisson regression to negative binomial regression, we can perform model comparison using (approxi-

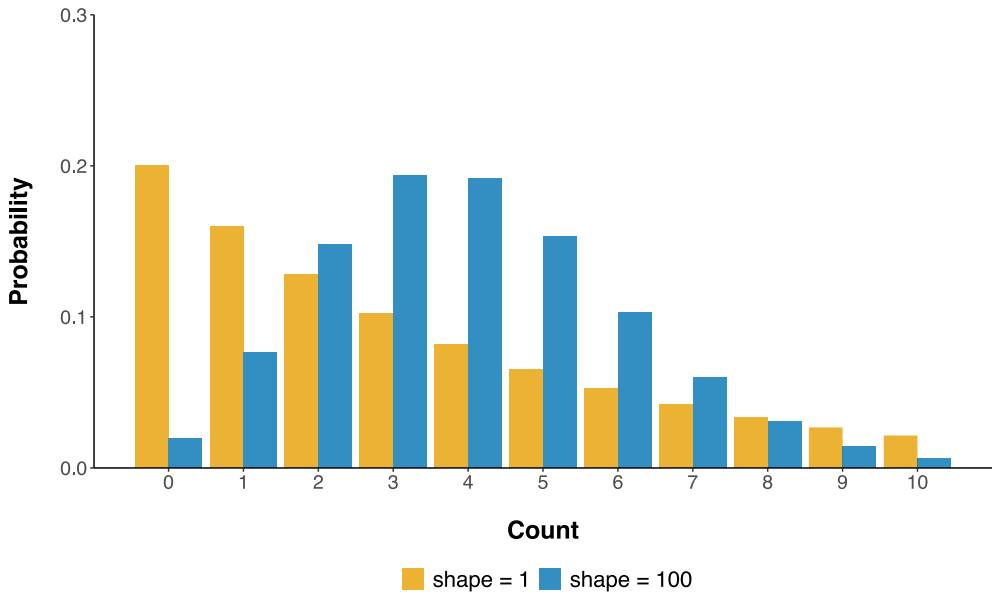

FIGURE 4 Counts with their associated probabilities expected under the negative binomial distribution for $\lambda = 4$ and two different shape parameters; yellow: shape = 1; blue: shape = 100

mate) leave-one-out cross-validation (LOO-CV) (Vehtari et al., 2017).[4] In using LOO-CV, we are evaluating models based on how well they predict unseen data. For more detailed descriptions with linguistic applications, see Vasishth et al. (2018) and Nalborczyk et al. (2019). We can perform LOO-CV separately for each model as follows:

```
pois_loo <- loo(mdl)
negbinom_loo <- loo(negbinom_mdl)
```

The comparison between the two models is performed with `loo_compare()`:

```
loos <- loo_compare(pois_loo, negbinom_loo)
loos
```

```
             elpd_diff  se_diff
negbinom_mdl  0.0         0.0
mdl          -0.1         0.7
```

ELPD stands for 'expected log-predictive density'; it is a measure of the expected predictive accuracy of the model, that is, it estimates how well the model is expected to predict unseen data. The model in the first row is always taken as the baseline and the model with the highest ELPD value in the comparison set. This means that in this particular case, the negative binomial model performs slightly better than the Poisson model. However, this difference is negligible. The `loo_compare()` function also provides estimates of uncertainty in the resultant LOO-CV criteria, which can be used to assess whether the differences between models are reliable. In this case, the difference in predictive power between the two models is negligible (elpd_diff = `-0.1`) and associated with a comparably large standard error (se_diff = `0.7`), suggesting that the Poisson model and the negative binomial model do not differ much from each other. This may lead one to consider using a Poisson model because it is the simpler model (one parameter less is being estimated) that performs about equally well in terms of predictive accuracy. However, there also would not be much harm by fitting a negative binomial model, which only includes one additional term, but importantly it has the flexibility of capturing overdispersion if it is present in the data. Linguistic data is very often overdispersed (for examples, see e.g., Bentz & Winter, 2014; Winter et al., 2018), and we recommend that if a researcher is in doubt about which model should be used, a negative binomial model is most often the more conservative choice and a sensible default to account for possible overdispersion.

3.6 | Incorporating weakly informative priors

So far, we have neglected a major advantage of Bayesian modelling because we have not specified any priors, a key component of Bayesian inference. Priors embody assumptions about plausible parameter values that are specified based on domain knowledge, such as drawn from the existing literature. Priors can vary in how specific or 'informative' they are, with more informative priors embodying stronger pre-existing information, which puts stronger constraints on what can be learned from the data.

Noninformative priors are priors where the range of uncertainty in a parameter's value is larger than any plausible parameter value. Using such noninformative priors effectively negates a key advantage of using Bayesian statistics (Lemoine, 2019) as it means that posterior estimates of pa-

rameter values are almost entirely driven by the data. This feature of noninformative priors means that the corresponding models have a tendency to overfit (Lemoine, 2019; McElreath, 2020), which is also a characteristic of the corresponding frequentist models (such as fitted with `lme4`), which assume that all values are equally likely a priori. In contrast, 'weakly informative priors' allow incorporating 'mild skepticism' (McElreath, 2020, p. 214) into the model while at the same time not imposing constraints that are too strong (e.g., wholly ruling out specific values). The use of weakly informative priors is generally recommended, but it is especially important in the case of small samples, as are often dealt with in linguistic data. This is because estimates from small samples are inherently more variable compared to the population they are drawn from, which means that extreme values should be penalised more strongly. Weakly informative priors are an excellent way to prevent overfitting, especially when sample size is small.

What counts as a 'weakly informative prior' depends on the parameter to be estimated and the given model (Gelman et al., 2017). Here, we discuss what is perhaps the most important prior from the analyst's perspective, which is the prior of the slope term—the difference between the friend and professor condition. By setting a normally distributed prior centred at zero on this slope term, the posterior estimates will be shrunken towards zero compared to a model that assumes that all slope values are equally likely. This embodies the view that unrealistically large effects, especially if they are based on small sample sizes, should be constrained (Lemoine, 2019). Thus, weakly informative priors allow making models more conservative than the corresponding frequentist models that uses noninformative priors, thereby improving out-of-sample predictions and reducing the probability of Type I errors.

Some possible prior choices for the slope term of the above model are visualised in Figure 5, which shows three normal distributions centred at zero differing only in their standard deviations. The very wide prior (dotted line) corresponds to $SD = 2$; the solid line corresponds to $SD = 1$; the dashed line to $SD = 0.5$. The narrower the standard deviation, the more strongly the model assumes that slopes are closer to zero prior to having seen the data.

In this case, we went with $SD = 0.5$. Using the 68%–95% heuristic of the normal distribution as a guide, this embodies the prior belief that 68% of all slope values on the log scale would fall in between [−0.5, +0.5] (one standard deviation away from a zero difference), and 95% would fall in between the interval [−1, +1] (two standard deviations away from the mean). It is important to real-

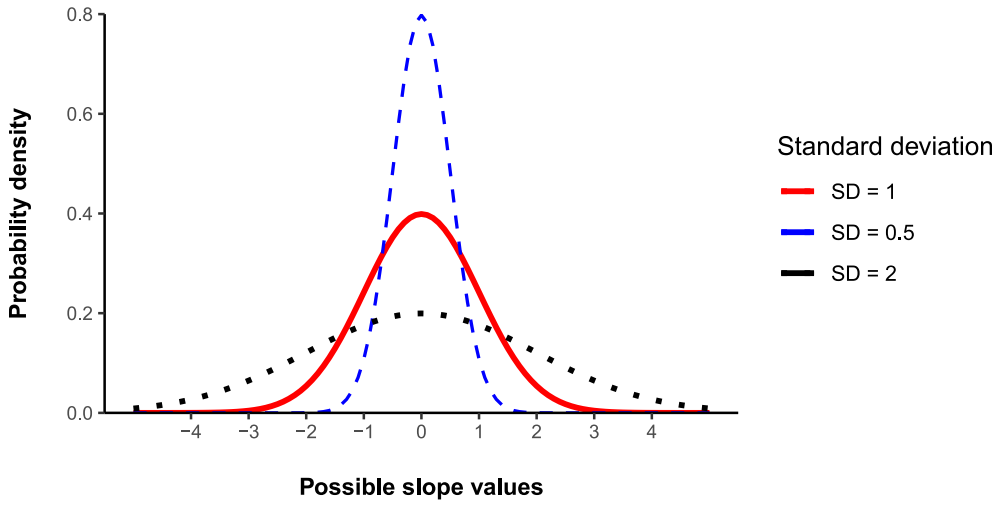

FIGURE 5 A normally distributed prior centred at zero for three different standard deviations

ise here that this is the prior of the slope, which in this case is the difference between the polite and non-polite conditions on the log scale. So, a normally distributed prior with $SD = 0.5$ means that the model assigns high probability (0.95) to differences ranging from −1 to +1 on a log scale. This prior specification correspondingly assigns low probabilities to very large differences exceeding this range.

An issue with choosing priors in the case of a GLM is that we have to keep the link function in mind (Lemoine, 2019). Because the log link is a nonlinear transformation, the same slope coefficient may result in different condition differences depending on the value of the intercept. For example, if we assume an intercept of −1 (similar to the model reported above), a prior of $SD = 0.5$ would mean that 68% of all differences would fall in between `exp(-1 + 0.5)` and `exp(-1 - 0.5)`, which is between ~0.22 (about 1 gesture every five seconds) and ~0.60 (about 3 gestures every five seconds). Thus, $SD = 0.5$ is already quite a wide prior, allowing for a whole range of differences between the conditions. Nonetheless, it is more conservative than a fully noninformative prior and estimates of the main condition effects will be slightly shrunken towards zero, which is an appropriate degree of scepticism when working with a relatively small dataset like the present one. For a more detailed tutorial discussion on prior choice, see Lemoine (2019), as well as Schad et al. (2021).

We specify the prior outside of the `brms()` function call as follows:

```
weak_priors <- prior(normal(0, 0.5), class = b)
```

Refitting the above (negative binomial) model with the `prior` argument set to `weak_priors` yields the following new coefficient estimates:

```
Population-Level Effects:

             Estimate  Est.Error  l-95% CI  u-95% CI
Intercept       -0.97       0.08     -1.12     -0.82
contextprof     -0.11       0.06     -0.23      0.00
```

Notice that in this particular case, the point estimate of the condition effect (`-0.11`) is nearly identical compared to the previous negative binomial model. It is important to stress that the impact of priors depends on the sample size: prior choices tend to matter progressively less the more data there is. For a 'consumer's guide to weakly informative priors' with detailed recommendations for making choices, see Lemoine (2019).

In a Bayesian framework, all model parameters (including the random effects terms, the shape parameter in a negative binomial regression etc.) are associated with priors. `brms` has weakly or non-informative default prior choices (Bürkner, 2017) that can be interrogated with the function `prior_summary()`, but researchers can (and most often should) specify priors for all model parameters by hand to exercise control over model assumptions. In the context of a mixed model, for example, researchers also need to specify priors for the random effect terms as well. A detailed discussion of these additional prior specifications is beyond the scope of this paper, but we refer the reader to Vasishth et al. (2018) and Nalborczyk et al. (2019), who provide introductions with linguistic examples.

3.7 | Posterior predictive checks

Posterior predictive checks are a basic way of assessing model adequacy in a Bayesian framework (Gabry et al., 2019; Gelman et al., 2020). The purpose of performing posterior predictive checks

is to look for any major discrepancies between the real data and the data simulated from the model. If there are major discrepancies, the model is highly unlikely to be close to the true (but unknown) data generating process, which is an argument for revising the original model. The following code executes posterior predictive checks for 100 posterior simulations based on the negative binomial model specified above.

```
pp_check(negbinom_mdl, ndraws = 100,
         type = 'ecdf_overlay')
```

The `pp_check()` function allows a number of different visualisation types. Here we specify the argument `type = 'ecdf_overlay'` to return an empirical cumulative distribution function (ECDF). By default, `pp_check()` returns a smoothed output which may be inappropriate for discrete data, such as count data. The ECDF plot shown in Figure 6 results from the function call above and can be interpreted as follows: The *y*-axis indicates the proportion of values falling below a value on the *x*-axis. For example, for a count of 50 on the *x*-axis, the proportion is about 0.5, indicating that about 50% of all observed counts fell beneath this value. The distribution function converges on 1.0 for values of 100, indicating that almost all observed counts were lower than that. The blue lines represent the simulated data, the black lines the actual data. What we are looking for is that the black line (the cumulative distribution function of the data) could reasonably fall within the simulated data. This appears to be the case for this data, suggesting that the model could reasonably have generated the data.

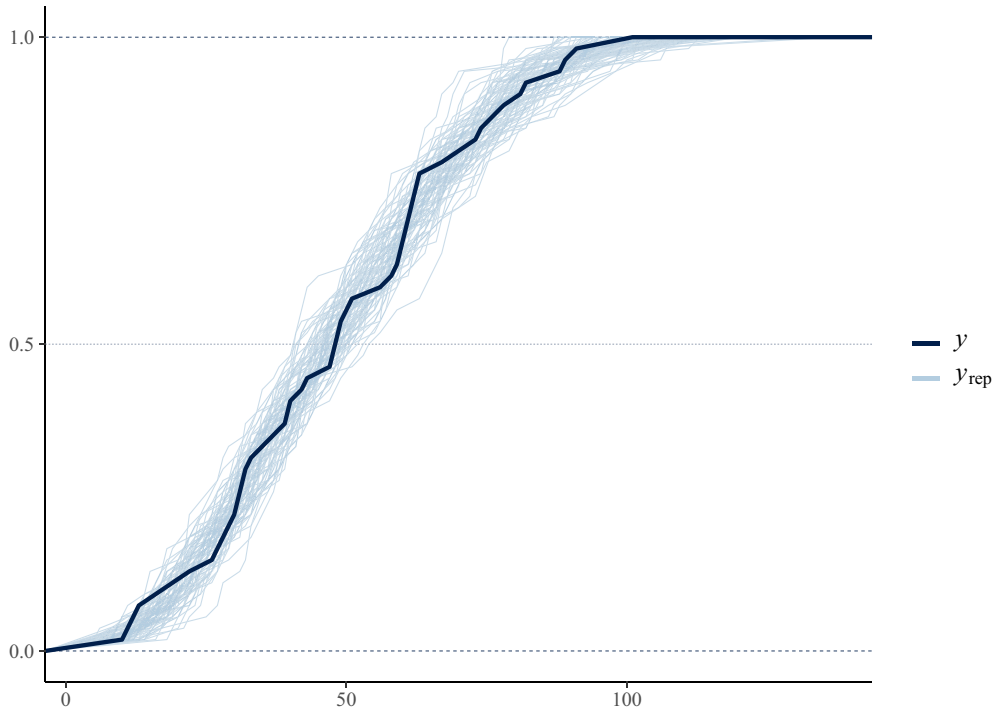

FIGURE 6 Posterior predictive checks (empirical cumulative distribution function)

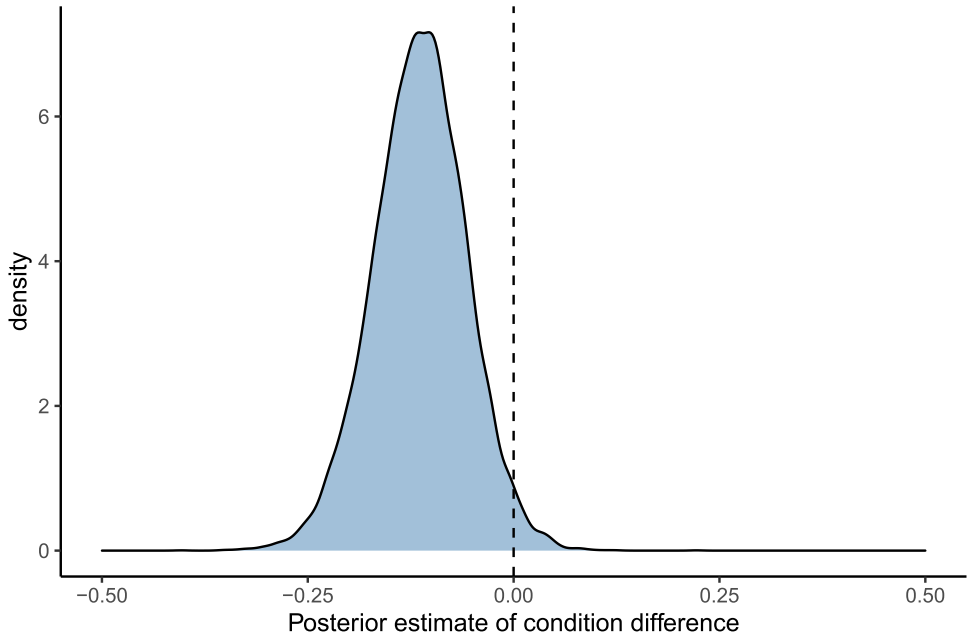

FIGURE 7 Posterior distribution of the condition slopes

3.8 | Inference

There are many ways of performing inference with Bayesian models, as well as many different ways of assessing model performance via different criteria (Navarro, 2019; Vehtari et al., 2017). Here, we merely focus on interpreting the posterior estimates, the most basic aspect of interpreting the implications of one's model. We discuss two ways to communicate uncertainty in the parameter estimates to an audience. First, by communicating information about the posterior distribution, using 95% Bayesian credible intervals and visualisations of the posterior distribution. Second, by computing posterior probabilities for specific comparisons of interest.

Figure 7 shows the posterior distribution for the condition difference. This distribution clearly shows that positive values (*more* gestures in the professor condition) are not entirely ruled out, but overall, it is much more plausible that participants gestured *less* in the professor condition.

We can also calculate the posterior probability of the effect being of below zero with the `hypothesis()` function as follows:

```
hypothesis(negbinom_mdl, 'contextprof < 0')
```

The resulting output shows that, given this model specification and data, the estimated posterior probability of the condition effect being of the same sign is $p(\beta < 0) = 0.98$. While posterior probabilities may be appealing to researchers as single-number summaries for specific hypothesis tests, these should always be reported together with at least 95% credible intervals (or some other percentage interval, see McElreath, 2020). Even better is to present a full visualisation of the posterior distribution.

This completes our introductory analysis. In the following, we discuss common questions that we anticipate the reader may have.

4 | ANTICIPATING COMMON QUESTIONS AND EXTENSIONS

4.1 | Can't I just log-transform counts?

When analysts new to Poisson regression hear that these models estimate the log rate of a count process, they often wonder whether one could apply standard linear regression to log-transformed counts. There are several reasons why this is not advisable (cf. O'hara & Kotze, 2010): First, conceptually, a simple linear regression of a log-transformed count variable models a different quantity—log counts—and not the primary quantity of interest (the actual counts). The Poisson model directly relates to the count process, which is different from applying linear regression to log counts. Second, log-transforming counts runs into issues when there are zeros in the data, since zeroes cannot be log-transformed ($\log(0) = -\text{infinity}$). Third, due to the natural bounds of count data discussed above, there will often be heteroskedasticity (unequal variance) across the range of the predictor, even after log-transforming. This means that log transformed count data will frequently violate the homoscedasticity assumption of standard linear regression.

It is often possible to convert a count problem into a rate problem, such as when dividing counts over a temporal interval. This again results in interpretational problems with zeros, as any number divided by zero is still zero, thereby effectively cancelling any exposure information out for these values. This is problematic because observing a zero count for 10 s is much less informative than observing a zero count for 10 h, even though both would have the same zero value when rates are calculated by hand. As detailed above, exposure variables allow directly modelling rates within a Poisson regression context.

4.2 | How to decide between logistic regression and Poisson regression

Readers familiar with logistic regression may wonder how to make decisions about when to use Poisson regression as opposed to logistic regression. The critical difference between the two types of GLMs has to do with the presence of an upper bound. The binomial distribution that lies at the heart of logistic regression has a clear upper bound (the parameter N), whereas the Poisson distribution extends towards positive infinity. Figure 1 above only showed the probabilities associated with the counts from zero to 10, but even very high counts have positive probabilities, although they may be exceedingly low probabilities for very high values. This is different from the binomial distribution, which assigns zero probability to values above N. That is, the Poisson distribution is appropriate when dealing with count data for which the upper limit N is not known, does not exist, or is difficult or impossible to derive. When counts have a fixed and known upper limit, logistic regression is the more natural choice, such as when modelling correct versus incorrect responses on a trial-by-trial basis ($N = 1$), or when modelling the proportion of students in a classroom passing a test (where N is the number of students in a classroom).

The absence of a clear upper N is also why Poisson regression was a natural choice for the gesture data considered here. In this data, the unit of analysis was an entire trial, which included multiple sentences. Even if we broke the data down to individual sentences, there would be no clear upper N: a given sentence can have no gestures or many gestures, but it is not clear how many gestures can fit into a sentence. Many other phenomena in linguistics are similar to this, such as certain discourse markers, fillers, or silent pauses. For example, Winter and Grawun-

der (2012) modelled the number of 'hisses' Korean speakers produced when speaking formally (polite speech) or informally (intimate speech). These hisses are similar to interjections, and due to their lack of syntactic integration, they can occur within or in between sentences, or they can have multiple occurrences within the same sentence. Again, there is no a priori theory to suggest that there may be a knowable upper limit for how many hisses fit into a sentence, which is why the Poisson distribution was a natural choice for the analysis presented in Winter and Grawunder (2012). Similarly, examples with no clear upper *N* abound in corpus linguistics, for which there is generally no fixed upper limit for how many words a corpus, or a section of a corpus, can contain. See Winter et al. (2018) for an example of an analysis of word frequency data with Poisson/negative binomial regression.

4.3 | Zero-inflation

This tutorial has dealt with overdispersion. Another situation analysts can run into is an excess number of zeroes. For example, when counting case markers in a language, a lot of languages have no case system to begin with, thus having zero counts (Bentz & Winter, 2014). As another example, consider the above-mentioned counts of 'hisses' (Winter & Grawunder, 2012): some people may never use these within a given experiment for any reason (e.g., personal preference). These and other situations will involve an excess number of zeroes (more than is expected under the Poisson distribution), which can be dealt with by using zero-inflated Poisson regression (or zero-inflated negative binomial regression). These models assume that additional zeroes are generated by a separate process, distinct from the count process. Zero-inflated models are also implemented in `brms` and described in more detail with a hands-on example in Bürkner (2018).

5 | OUTLOOK

Given the ubiquity of count data in linguistics, Poisson regression should arguably be a much more widely adopted approach. We have applied Poisson/negative binomial regression to data analysis problems in phonetics (Winter & Grawunder, 2012), cognitive linguistics (Winter et al., 2018), and typology (Bentz & Winter, 2014). A particular field that would seem to benefit tremendously from considering Poisson regression as an alternative analysis approach is corpus linguistics. There, textbooks still teach students how to apply such significance tests as Chi-square tests to corpus data (Desagulier, 2017; Stefanowitsch, 2020; Wallis, 2021), even though when such tests are applied to corpus data, they violate the independence assumption, a problem that has received a lot of attention in corpus linguistics (Baroni & Evert, 2009; Brezina & Meyerhoff, 2014; Gries, 2015; Kilgarriff, 2005; Koplenig, 2019; Lijffijt et al., 2016; Oakes & Farrow, 2006; Winter & Grice, 2021). Mixed models are a natural solution to dealing with dependency structures within corpora (Gries, 2015), however, without Poisson regression, many situations where aggregate counts are analysed (e.g., word frequencies across entire texts), the cases to which mixed models can be applied are more limited. Thus, alongside logistic regression, Poisson regression is a natural extension of the analytical toolkit to widen the type of linguistic datasets that can be analysed with mixed models.

ACKNOWLEDGEMENTS

Partially funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. Bodo Winter was supported by the UKRI Future Leaders Fellowship MR/T040505/1.

ORCID

Bodo Winter https://orcid.org/0000-0001-6036-6774
Paul-Christian Bürkner https://orcid.org/0000-0001-5765-8995

ENDNOTES

[1] The Bernoulli distribution is a specific instance of the binomial distribution when $N = 1$, as is common when logistic regression models are fitted to the individual trial level, or in the case of corpus data, to the level of individual choices (such as double object vs. prepositional dative construction).

[2] For an extensive discussion of mixed models in various subfields of linguistics, see Winter and Grice (2021).

[3] According to the quotient rule of logarithms, $\log(\mu/\tau) = \log(\mu) - \log(\tau)$. Replacing $\log(\lambda)$ in Equation (3) with this difference yields: $\log(\mu) - \log(\tau) = \alpha + \beta x$. Moving $\log(\tau)$ over by adding it to both sides of the equation yields Equation (6).

[4] Another way of assessing whether a negative binomial model is needed is by looking at posterior predictive checks (see Section 3.7), which could reveal overdispersion via a discrepancy between the simulated and the raw data.

REFERENCES

Aarts, E., Dolan, C. V., Verhage, M., & van der Sluis, S. (2015). Multilevel analysis quantifies variation in the experimental effect while optimizing power and preventing false positives. *BMC Neuroscience*, *16*(1), 94. https://doi.org/10.1186/s12868-015-0228-5

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

Baroni, M., & Evert, S. (2009). Statistical methods for corpus exploitation. In A. Lüdeling & M. Kytö (Eds.), Corpus linguistics: An international handbook (Vol. 2, pp. 777–803). Mouton de Gruyter.

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

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Bentz, C., & Winter, B. (2014). Languages with more second language learners tend to lose nominal case. In S. Wichmann & J. Good (Eds.), *Quantifying language dynamics* (pp. 96–124). Brill.

Brezina, V. (2018). *Statistics in corpus linguistics: A practical guide*. Cambridge University Press.

Brezina, V., & Meyerhoff, M. (2014). Significant or random? A critical review of sociolinguistic generalisations based on large corpora. *International Journal of Corpus Linguistics*, *19*(1), 1–28. https://doi.org/10.1075/ijcl.19.1.01bre

Brown, L., Kim, H., Hübscher, I., & Winter, B. (in press). Iconic gestures are modulated by social context: A study of multimodal politeness across two cultures. *Gesture*.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28.

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411.

Cassidy, S. A., Dimova, R., Giguère, B., Spence, J. R., & Stanley, D. J. (2019). Failing grade: 89% of introduction-to-psychology textbooks that define or explain statistical significance do so incorrectly. *Advances in Methods and Practices in Psychological Science*, *2*(3), 233–239. https://doi.org/10.1177/2515245919858072

Desagulier, G. (2017). *Corpus linguistics and statistics with R: Introduction to quantitative methods in linguistics*. Springer.

Dienes, Z. (2008). *Understanding psychology as a science: An introduction to scientific and statistical inference*. Macmillan.

Dienes, Z. (2011). Bayesian versus orthodox statistics: Which side are you on? *Perspectives on Psychological Science*, *6*(3), 274–290. https://doi.org/10.1177/1745691611406920

Eager, C., & Roy, J. (2017). Mixed effects models are sometimes terrible. *ArXiv Preprint ArXiv:1701.04858*.

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

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A*, *182*(2), 389–402. https://doi.org/10.1111/rssa.12378

Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press.

Gelman, A., & Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, *66*(1), 8–38. https://doi.org/10.1111/j.2044-8317.2011.02037.x

Gelman, A., Simpson, D., & Betancourt, M. (2017). The prior can often only be understood in the context of the likelihood. *Entropy*, *19*(10), 555. https://doi.org/10.3390/e19100555

Gries, S. (2015). The most under-used statistical method in corpus linguistics: Multi-level (and mixed-effects) models. *Corpora*, *10*(1), 95–125. https://doi.org/10.3366/cor.2015.0068

Gries, S. (2017). *Quantitative corpus linguistics with R: A practical introduction*. Taylor & Francis.

Haller, H., & Krauss, S. (2002). Misinterpretations of significance: A problem students share with their teachers. *Methods of Psychological Research*, *7*(1), 1–20.

Jaeger, T. F., Graff, P., Croft, W., & Pontillo, D. (2011). Mixed effect models for genetic and areal dependencies in linguistic typology. *Linguistic Typology*, *15*(2), 281–319. https://doi.org/10.1515/lity.2011.021

Johnson, D. E. (2009). Getting off the GoldVarb standard: Introducing Rbrul for mixed-effects variable rule analysis. *Language and Linguistics Compass*, *3*(1), 359–383. https://doi.org/10.1111/j.1749-818X.2008.00108.x

Kilgarriff, A. (2005). Language is never, ever, ever, random. *Corpus Linguistics and Linguistic Theory*, *1*(2), 263–276. https://doi.org/10.1515/cllt.2005.1.2.263

Kimball, A. E., Shantz, K., Eager, C., & Roy, J. (2019). Confronting quasi-separation in logistic mixed effects for linguistic data: A Bayesian approach. *Journal of Quantitative Linguistics*, *26*(3), 231–255. https://doi.org/10.1080/09296174.2018.1499457

Kirby, J., & Sonderegger, M. (2018). Mixed-effects design analysis for experimental phonetics. *Journal of Phonetics*, *70*, 70–85. https://doi.org/10.1016/j.wocn.2018.05.005

Koplenig, A. (2019). Against statistical significance testing in corpus linguistics. *Corpus Linguistics and Linguistic Theory*, *15*(2), 321–346. https://doi.org/10.1515/cllt-2016-0036

Kruschke, J. K. (2013). Posterior predictive checks can and should be Bayesian: Comment on Gelman and Shalizi, 'Philosophy and the practice of Bayesian statistics.' *British Journal of Mathematical and Statistical Psychology*, *66*(1), 45–56. https://doi.org/10.1111/j.2044-8317.2012.02063.x

Kruschke, J. K., & Liddell, T. M. (2018). Bayesian data analysis for newcomers. *Psychonomic Bulletin & Review*, *25*(1), 155–177. https://doi.org/10.3758/s13423-017-1272-1

Lecoutre, B., Lecoutre, M.-P., & Poitevineau, J. (2001). Uses, abuses and misuses of significance tests in the scientific community: Won't the Bayesian choice be unavoidable? *International Statistical Review*, *69*(3), 399–417. https://doi.org/10.1111/j.1751-5823.2001.tb00466.x

Lecoutre, M.-P., Poitevineau, J., & Lecoutre, B. (2003). Even statisticians are not immune to misinterpretations of Null Hypothesis Significance Tests. *International Journal of Psychology*, *38*(1), 37–45. https://doi.org/10.1080/00207590244000250

Lemoine, N. P. (2019). Moving beyond noninformative priors: Why and how to choose weakly informative priors in Bayesian analyses. *Oikos*, *128*(7), 912–928. https://doi.org/10.1111/oik.05985

Levshina, N. (2015). *How to do linguistics with R: Data exploration and statistical analysis*. John Benjamins.

Lijffijt, J., Nevalainen, T., Säily, T., Papapetrou, P., Puolamäki, K., & Mannila, H. (2016). Significance testing of word frequencies in corpora. *Literary and Linguistic Computing*, *31*(2), 374–397. https://doi.org/10.1093/llc/fqu064

McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan* (2nd ed.). CRC Press.

McShane, B. B., & Gal, D. (2017). Statistical significance and the dichotomization of evidence. *Journal of the American Statistical Association*, *112*(519), 885–895. https://doi.org/10.1080/01621459.2017.1289846

Morey, R. D., Hoekstra, R., Rouder, J. N., Lee, M. D., & Wagenmakers, E.-J. (2016). The fallacy of placing confidence in confidence intervals. *Psychonomic Bulletin & Review*, *23*(1), 103–123. https://doi.org/10.3758/s13423-015-0947-8

Nalborczyk, L., Batailler, C., Løevenbruck, H., Vilain, A., & Bürkner, P.-C. (2019). An introduction to Bayesian multilevel models using brms: A case study of gender effects on vowel variability in standard Indonesian. *Journal of Speech, Language, and Hearing Research*, *62*(5), 1225–1242. https://doi.org/10.1044/2018_JSLHR-S-18-0006

Navarro, D. J. (2019). Between the devil and the deep blue sea: Tensions between scientific judgement and statistical model selection. *Computational Brain & Behavior*, *2*(1), 28–34. https://doi.org/10.1007/s42113-018-0019-z

Nicenboim, B., & Vasishth, S. (2016). Statistical methods for linguistic research: Foundational Ideas—Part II. *Language and Linguistics Compass*, *10*(11), 591–613. https://doi.org/10.1111/lnc3.12207

Oakes, M. P., & Farrow, M. (2006). Use of the chi-squared test to examine vocabulary differences in English language corpora representing seven different countries. *Literary and Linguistic Computing*, *22*(1), 85–99. https://doi.org/10.1093/llc/fql044

O'hara, R. B., & Kotze, D. J. (2010). Do not log-transform count data. *Methods in Ecology and Evolution*, *1*(2), 118–122. https://doi.org/10.1111/j.2041-210X.2010.00021.x

R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Schad, D. J., Nicenboim, B., Bürkner, P.-C., Betancourt, M., & Vasishth, S. (2021). Workflow techniques for the robust use of Bayes Factors. *ArXiv Preprint ArXiv:2103.08744*.

Schielzeth, H., & Forstmeier, W. (2008). Conclusions beyond support: Overconfident estimates in mixed models. *Behavioral Ecology*, *20*(2), 416–420. https://doi.org/10.1093/beheco/arn145

Schneider, J. W. (2015). Null hypothesis significance tests. A mix-up of two different theories: The basis for widespread confusion and numerous misinterpretations. *Scientometrics*, *102*(1), 411–432. https://doi.org/10.1007/s11192-014-1251-5

Sorensen, T., & Vasishth, S. (2015). Bayesian linear mixed models using Stan: A tutorial for psychologists, linguists, and cognitive scientists. *ArXiv Preprint ArXiv:1506.06201*.

Stan Development Team. (2021). *Stan modeling language user's guide and reference manual*. http://mc-stan.org/

Stefanowitsch, A. (2020). *Corpus linguistics: A guide to the methodology*. Language Science Press.

Tagliamonte, S. A., & Baayen, H. (2012). Models, forests, and trees of York English: Was/were variation as a case study for statistical practice. *Language Variation and Change*, *24*(2), 135–178.

Vasishth, S., Nicenboim, B., Beckman, M. E., Li, F., & Kong, E. J. (2018). Bayesian data analysis in the phonetic sciences: A tutorial introduction. *Journal of Phonetics*, *71*, 147–161. https://doi.org/10.1016/j.wocn.2018.07.008

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5), 1413–1432. https://doi.org/10.1007/s11222-016-9696-4

Wallis, S. (2021). *Statistics in corpus linguistics research: A new approach*. Routledge.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., & Hester, J. (2019). Welcome to the Tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., & Grolemund, G. (2016). *R for data science*. O'Reilly.

Winter, B., & Grawunder, S. (2012). The phonetic profile of Korean formal and informal speech registers. *Journal of Phonetics*, *40*(6), 808–815. https://doi.org/10.1016/j.wocn.2012.08.006

Winter, B., & Grice, M. (2021). Independence and generalizability in linguistics. *Linguistics*, *59*(5), 1251–1277. https://doi.org/10.1515/ling-2019-0049

Winter, B., Perlman, M., & Majid, A. (2018). Vision dominates in perceptual language: English sensory vocabulary is optimized for usage. *Cognition*, *179*, 213–220.

Zuur, A., Ieno, E. N., Walker, N., Saveliev, A. A., & Smith, G. M. (2009). *Mixed effects models and extensions in ecology with R*. Springer.

AUTHOR BIOGRAPHIES

Bodo Winter is a Senior Lecturer at the Department of Linguistics at the University of Birmingham, Editor-in-Chief at the interdisciplinary journal *Language and Cognition*, and UKRI Future Leader Fellow. He received his PhD in Cognitive and Information Sciences from the University of California, Merced. He is the author of the statistics textbook *Statistics for linguists: An introduction using R* and co-founder of the Birmingham Statistics for Linguistics Summer School.

Paul-Christian Bürkner is a statistician currently working as an Independent Junior Research Group Leader at the Cluster of Excellence SimTech at the University of Stuttgart, Germany. He received his PhD from the University of Münster, Germany, where he studied Psychology and Mathematics. He is the author of the R package brms and member of the Stan Development Team.

How to cite this article: Winter, B., & Bürkner, P.-C. (2021). Poisson regression for linguists: A tutorial introduction to modelling count data with brms. *Language & Linguistics Compass*, e12439. https://doi.org/10.1111/lnc3.12439