arXiv:1904.12765v2 [stat.ME] 2 May 2019

Toward a principled Bayesian workflow in cognitive science

Daniel J. Schad[1], Michael Betancourt[2], & Shravan Vasishth[1]

[1] University of Potsdam, Germany

[2] Symplectomorphic, New York, USA

Author Note

Correspondence concerning this article should be addressed to Daniel J. Schad, University of Potsdam, Germany. E-mail: danieljschad@gmail.com

Abstract

Experiments in research on memory, language, and in other areas of cognitive science are increasingly being analyzed using Bayesian methods. This has been facilitated by the development of probabilistic programming languages such as Stan, and easily accessible front-end packages such as brms. However, the utility of Bayesian methods ultimately depends on the relevance of the Bayesian model, in particular whether or not it accurately captures the structure of the data and the data analyst's domain expertise. Even with powerful software, the analyst is responsible for verifying the utility of their model. To accomplish this, we introduce a principled Bayesian workflow (Betancourt, 2018) to cognitive science. Using a concrete working example, we describe basic questions one should ask about the model: prior predictive checks, computational faithfulness, model sensitivity, and posterior predictive checks. The running example for demonstrating the workflow is data on reading times with a linguistic manipulation of object versus subject relative sentences. This principled Bayesian workflow also demonstrates how to use domain knowledge to inform prior distributions. It provides guidelines and checks for valid data analysis, avoiding overfitting complex models to noise, and capturing relevant data structure in a probabilistic model. Given the increasing use of Bayesian methods, we aim to discuss how these methods can be properly employed to obtain robust answers to scientific questions. All data and code accompanying this paper are available from https://osf.io/b2vx9/.

Keywords: Workflow, prior predictive checks, posterior predictive checks, model building, Bayesian data analysis

Toward a principled Bayesian workflow in cognitive science

Introduction

Recent years have seen a rise in the use of Bayesian statistics for data analysis in the cognitive sciences and other areas (for a recent overview see contributions presented at Stancon 2018: https://mc-stan.org/events/stancon2018/). This rise has been fueled by increasing recognition of the advantages of robust Bayesian analyses (Gelman et al., 2014). Moreover, much progress has been made in the development of probabilistic programming languages such as Stan (Carpenter et al., 2017), WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000), and JAGS (Plummer, 2012). Packages like *brms* (Bürkner, 2017) now provide easy access to fitting complex hierarchical (non-)linear mixed-effects models in the R System for Statistical Computing (R Core Team, 2016). Front-ends like *brms* have the advantage that standardly used models in cognitive science can be fit using a familiar syntax that is well-known from fitting frequentist linear mixed effects models (the *lme4* package, Baayen, Davidson, & Bates, 2008; Bates, Mächler, Bolker, & Walker, 2015).

Although complex and powerful, Bayesian analysis tools are thus now easily accessible to lay users, and although these tools greatly facilitate Bayesian computations, the model specification is still (as it should be) the responsibility of the user. However, the steps needed to arrive at a useful and robust analysis are usually not spelt out in introductory textbooks or tutorial articles. The present paper seeks to fill this gap in the literature.

Much research has been carried out in recent years to develop tools to ensure robust Bayesian data analyses (e.g., Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2017; Talts, Betancourt, Simpson, Vehtari, & Gelman, 2018). One of the most recent products of this research has been the formulation of a principled Bayesian workflow for conducting a probabilistic analysis (Betancourt, 2018). This workflow provides an initial coherent picture of steps to take for a robust analysis, leaving room for further improvements and

methodological developments. At an abstract level, parts of this workflow can be applied to any kind of data analysis, be it frequentist or Bayesian, be it based on sampling or on analytic procedures.

However, the papers cited above are written either for a general audience or for the professional statistics researcher. For newcomers to Bayesian methods, translating these ideas to their own domain is often very difficult to impossible. What is needed is an explicit, reproducible, fully worked-out example of the workflow for a common type of experimental design. The main challenge with field-specific translations of statistical methods is that they need to be accessible to a non-technical audience but at the same time uncompromising on the details.

In order to fill this gap, for the field of cognitive science (linguistics, psychology, and related areas), we introduce this principled Bayesian workflow by illustrating its use with experimental data from a reading-time experiment. We demonstrate how to implement analyses in R, and use the R package *brms* (Bürkner, 2017) for statistical analysis. Note that some parts of this principled Bayesian workflow can demand considerable computational resources. However, some of these checks could be implemented once for a given research program, as similar experimental designs and statistical analyses may not demand a fundamental re-analysis of all the steps of this workflow for every single follow-up experiment.

Before starting with describing the workflow, we briefly lay out some basic definitions and terminology related to Bayesian modeling and inference. For a detailed introductory treatment, see Lambert (2018), and also https://github.com/betanalpha/knitr_case_studies/tree/master/modeling_and_inference.

In the Bayesian framework, we aim to understand the processes that have generated some observed data y. For this, we use a statistical (or possibly a computational) model $\pi(y \mid \theta)$, which maps a set of parameters θ via the model to predict the observed data y. A

simple example of a model could be a linear regression. Here, we consider the more complex case of a hierarchical linear model, familiar to cognitive scientists as the linear mixed-effects model (Pinheiro & Bates, 2000). The model specifies a conditional probability distribution for the data given the parameters, that is, it defines a likelihood function. Bayes' rule allows us to use the likelihood to obtain the posterior probability distribution of the parameters given the data $\pi(\theta \mid y)$:

$$\pi(\theta \mid y) = \frac{\pi(y \mid \theta)\pi(\theta)}{\pi(y)} \tag{1}$$

This involves prior distributions over the parameters $\pi(\theta)$. These prior distributions can represent pre-existing knowledge or beliefs about the parameters that are available before the data are observed. Moreover, the probability for the data $\pi(y)$ is a normalizing constant, and is obtained as the integral over all parameter settings: $\pi(y) = \int \pi(y \mid \theta)\pi(\theta)\mathrm{d}\theta$. Because it is a constant, it can be ignored when computing the posterior distributions of the parameters.

Given the likelihood and the prior, a key challenge in Bayesian computations is to compute posterior expectations of the parameters accurately, for example their mean, or alternatively posterior quantiles in credible intervals. For most interesting applications, the posterior expectations cannot be computed analytically. Instead, probabilistic posterior sampling (as implemented in various probabilistic programming languages such as Stan) is a method of key choice for performing accurate posterior computations.

Here, we provide a detailed description of a number of questions to ask about a model and checks to perform to validate a probabilistic model. Before going into the details of this discussion, we first treat the process of model building, and how different traditions have yielded different approaches to this questions.

Model building

One strategy for model building is to start with a minimal model that captures just the phenomenon of interest but not much other structure in the data. For example, this could be a linear model with just the factor or covariate of main interest. For this model, we perform a number of checks described in detail in the following sections. If the model passes all checks and does not show signs of inadequacy, then it can be applied in practice and we can be confident that the model provides reasonably robust inferences on our scientific question. However, if the model shows signs of trouble on one or more of these checks, then the model may need to be improved. Alternatively, we may need to be more modest with respect to our scientific question. For example, in a repeated measures data-set, we may be interested in estimating the correlation parameter between two random effects (i.e., their random effects correlation) based on a sample of 30 subjects. If model analysis reveals that our sample size is not sufficiently large to estimate the random effects correlation reliably, then we may need to either increase our sample size, or give up our plan of analyzing the random effects correlation based on this data.

During the model building process, we make use of an aspirational model π_A: we mentally imagine a model with all the possible details that the phenomenon and measurement process contain; i.e. we imagine a model that one would fit if there were no limitations in resources, time, mathematical and computational tools, subjects, and so forth. It would contain all systematic effects that might influence the measurement process. For example, influences of time or heterogeneity across individuals. This should be taken to guide and inform model development; such a procedure prevents random walks in model space during model development. Note that the model has to consider both the latent phenomenon of interest as well as the environment and experiment used to probe it.

The initial model π_1, to the contrary, may only contain enough structure to incorporate

the phenomenon of core scientific interest, but none of the additional aspects/structures relevant for the modeling or measurement. The additional, initially left-out structures, which reflect the difference between the initial (π_1) and the aspirational model (π_A), can then be probed for using specifically designed summary statistics. These summary statistics can thus inform model expansion from the initial model π_1 into the direction of the aspirational model π_A. If the initial model proves inadequate, then the aspirational model and the associated summary statistics guide model developement. If the expanded model is still not adequate, then another cycle of model developement is conducted.

The range of prior and posterior predictive checks discussed in the following sections provide a principled Bayesian workflow of how this model expansion is done. The notion of *expansion* is critical here. If an expanded model does not prove more adequate, one can always fall back to the previous model version.

As an alternative analysis strategy, a rich tradition in the cognitive and other experimental sciences relies on *maximal models* for a given experimental design (Barr, Levy, Scheepers, & Tily, 2013). This maximal model contains all effects from experimental manipulations (main effects and interactions) as well as differences in these effects varying across subjects, items, or other random factors in the experimental design. That is, this model is maximal within the scope of a linear regression. Such a maximal model provides an alternative starting point for the principled Bayesian workflow. In this case, the focus does not lie so much on model expansion. Instead, core goals are to specify priors encoding domain expertise, and to ensure computational faithfulness, model sensitivity, and model adequacy. Some steps in the principled Bayesian workflow (e.g., the computationally more demanding steps of assessing computational faithfulness and model sensitivity) may even be performed only once for a given research program, where similar designs are repeatedly used. We will explain this in more detail below.

Note that in the maximal model, “maximal” refers to maximal within the scope of the

linear regression approximation, not maximal with respect to the actual data generating process. Models are still bound by the linear regression structure and hence cannot capture effects such as selection bias in the data, dynamical changes in processes across time, or measurement error. At the same time this restricted class of (linear regression) models is exactly what packages like *brms* target, so it might also be "maximal" within the possibilities of such tools. Importantly, however, these "maximal" models are not the aspirational model, which is an image of the true data generating process, but rather just another approximation. Indeed, aiming to formulate models closer to the aspiration model, which go beyond the linear model framework, may be one reason to consider investing in learning to express models directly within probabilistic programming languages such as Stan and JAGS instead of the limited range of models provided in packages like *brms*.

Finally, we note that sometimes the results from the Bayesian workflow will show that our experimental design or data is not sufficient to answer our scientific question at hand. In this case, ambition need to be reduced, or new data needs to be collected, possibly with a different experimental design more sensitive to the phenomenon of interest.

One important development in open science practices is pre-registration of experimental analyses before the data are collected. This can be done using online-platforms such as the Open Science Foundation or AsPredicted. What information can or should one document in preregistration of the Bayesian workflow? If one plans on using the maximal model for analysis, then this maximal model, including contrast coding (Schad, Hohenstein, Vasishth, & Kliegl, 2018), fixed effects, and random effects should be described. In the case of incremental model building, if a model isn't a good fit to the data, then any resulting inference will be limited if not useless, so a rigid preregistration is useless unless one knows exactly what the model is. Thus, the deeper issue with preregistration is that a model cannot be confirmed until the phenomenon *and* experiment are all extremely well understood. One practical possibility is to describe the initial and the aspirational model,

and the incremental strategy used to probe the initial model to move more towards the aspirational model. Note that this can also include delineation of summary statistics that one plans to use for probing the tested models. Even if it is difficult to spell out the aspirational model fully, it can be useful to preregister the initial model, summary statistics, and the principles one intends to apply in model selection. Note that while the maximal modeling approach clearly reflects confirmatory hypothesis testing, the incremental model building strategy towards the aspirational model may be seen as lying at the boundary between confirmatory and exploratory, and becomes more confirmatory the more clearly the aspirational model can be spelled out a priori.

Principled questions on a model

What characterizes a useful probabilistic model? First, a useful probabilistic model should be consistent with domain expertise. Second, it is key for it to allow accurate posterior approximation. Third, it must capture enough of the experimental design to give useful answers to our questions. Finally, crucially, a useful probabilistic model should be rich enough to capture the structure of the true data generating process needed to answer scientific questions.

So what can we do aiming to meet these properties of our probabilistic model? In the following, we will outline a number of analysis steps to take and questions to ask in order to improve these properties for our model (a more technical presentation is provided in Betancourt, 2018).

In a first step, we will use prior predictive checks to investigate whether our model is consistent with our domain expertise. Next, we will investigate computational faithfulness by studying whether posterior estimation is accurate. Third, we study model sensitivity and the question whether we can recover model parameters with the given design and model. As the

last step in model validation, posterior predictive checks assess model adequacy for the given data-set, that is, they investigate the question whether the model captures the relevant structure of the true data generating process.

Prior predictive checks: Checking consistency with domain expertise

The first key question for checking the model is whether the model and the distributions of prior parameters are consistent with domain expertise. Prior distributions can be selected based on prior research or plausibility. However, for complex models it is often difficult to know which prior distributions should be chosen, and what consequences distributions of prior model parameters have for expected data. A viable solution is to use prior distributions to simulate hypothetical data from the model and to check whether the simulated data are plausible and consistent with domain expertise, which is often much easier to judge compared to assessing prior distributions in complex models directly. This approach has been suggested in the *device of imaginary results* by Good (1950). In practice, this can be implemented by the following steps:

1. Take the prior $\pi(\theta)$ and randomly draw a parameter set $\tilde{\theta}$ from it: $\tilde{\theta} \sim \pi(\theta)$
2. Use this parameter set $\tilde{\theta}$ to simulate hypothetical data $\tilde{y}$ from the model: $\tilde{y} \sim \pi(y \mid \tilde{\theta})$

To assess whether prior model predictions are consistent with domain expertise, it is useful to compute summary statistics of the simulated data $t(\tilde{y})$. The distribution of these summary statistics can be visualized using, for example, histograms (see Fig. 1). This can quickly reveal whether the data falls in an expected range, or whether a substantial amount of extreme data points are expected a priori. For example, in a study using self-paced reading times, "extreme" values may be considered to be reading times smaller than 50 ms or larger than 2000 ms, which would not be impossible, but would be implausible and largely inconsistent with domain expertise. A small number of observations may actually take

extreme values. However, observing a large number of extreme data points in the hypothetical data would be inconsistent with domain expertise. In this case, the priors or the model should be adjusted to yield hypothetical data within a range of reasonable values.

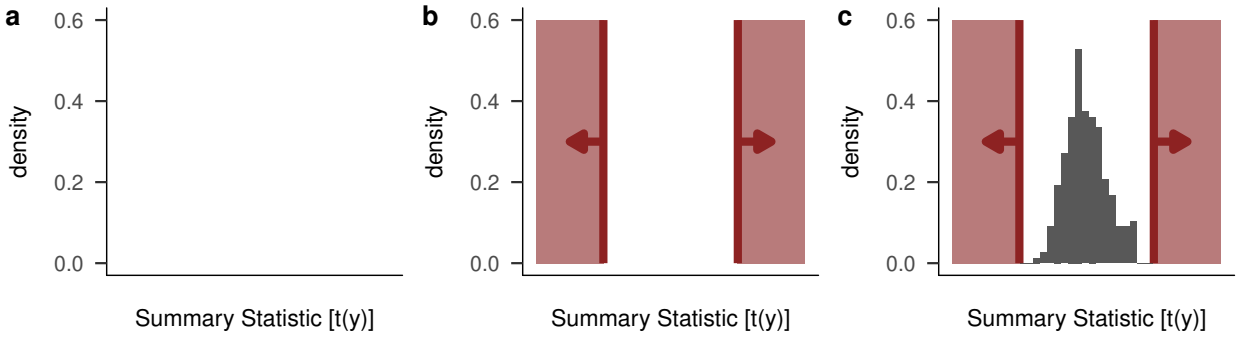

Figure 1. Prior predictive checks. a) In a first step, define a summary statistic that one wants to investigate. b) Second, define extremity thresholds (shaded areas), for which one does not expect a lot of prior data. c) Third, simulate prior model predictions for the data (histogram) and compare them with the extreme values (shaded areas).

Choosing good summary statistics is more an art than a science. The choice of summary statistics will be crucial, however, as they provide key markers of what we want the model to account for in the data. They should thus be carefully chosen and designed based on expectations we have about the true data generating process and about the kinds of structures and effects we expect the data may exhibit. Interestingly, summary statistics can also be used to encode criticisms: if someone wants to criticize an analysis, then they can formalize that criticism into a summary statistic they expect to show undesired behavior, which could e.g., provide a very constructive way to write reviews. Here, we will show some examples of useful summary statistics below when discussing data analysis for a concrete example data-set.

Note that choosing good priors will be particularly relevant in cases where the likelihood is not precise (see Fig. 2, in particular g-i). In linear mixed-effects models, for example, this often occurs in cases where a “maximal model” is fitted for a small data-set

that does not constrain estimation of all random effects variance and covariance parameters. In frequentist methods (such as implemented in the lme4 package in the lmer program), this problem is visible as problems with convergence of the optimizer, which indicates that the likelihood is too flat and that the parameter estimates are not constrained by the data.

In such situations, using a prior in a Bayesian analysis (or a more informative prior rather than a diffuse one) should incorporate just enough domain expertise to suppress extreme, although not impossible parameter values. This may allow the model to be fitted, as the posterior is now sufficiently constrained.

To the contrary, frequentist statistics of linear model regression is built on assumptions about asymptotic properties in the limit of very large data sets. If the likelihood is not sufficiently informative to constrain the parameter values (such as in Fig. 2e), these asymptotic assumptions are invalid and the results of a frequentist linear model regression no longer fully characterize inferences about the model. Therefore, introducing prior information in Bayesian computation allows fitting and interpreting models that can not be validly estimated using frequentist tools.

A welcome side-effect of incorporating more domain expertise (into what still constitutes weakly informative priors) is thus more concentrated prior distributions, which can facilitate Bayesian computation. This allows more complex models to be estimated, that is, using prior knowledge can make it possible to fit models that could otherwise not be estimated using the available tools. In other words, incorporating prior knowledge can allow us to get closer to the aspirational model in the iterative model building procedure. Moreover, more informative priors also lead to faster convergence of MCMC algorithms.

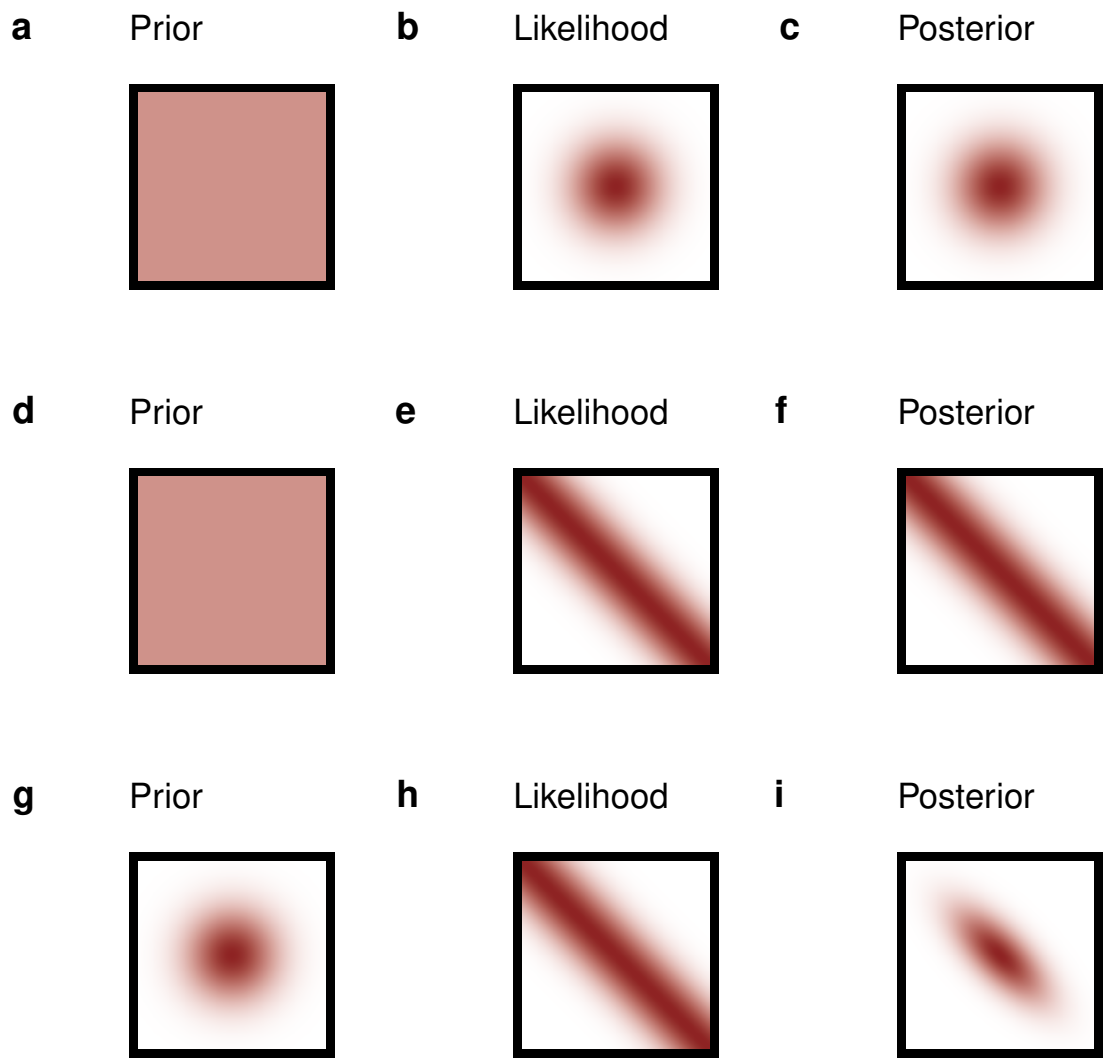

Figure 2. The role of priors for informative and uninformative data. a)-c) When the data provides good information via the likelihood (b), then a flat prior (a) is sufficient to obtain a concentrated posterior (c). d)-f) When the data does not sufficiently constrain the parameters through the likelihood (e), then using a flat prior (d) also leaves the posterior (f) diffuse. g)-i) When the data does not constrain the parameter through the likelihood (h), then including domain expertise into a (weakly) informative prior (g) can help to constrain the posterior (i) to reasonable values.

Incorporating more domain expertise into the prior also has crucial consequences for Bayesian modeling when computing Bayes factors. Bayes factors are highly sensitive to the prior, and in particular to the prior uncertainty. Priors are thus never uninformative when it comes to Bayes factors. Choosing very diffuse priors makes it very difficult to find posterior evidence in favor of an expanded model, and will often support the simpler model. For example, in nested model comparison of linear (mixed-effects) models, large prior uncertainty implies the assumption that the effect of interest could be very (implausibly) large. Using Bayes factors for such nested model comparison with high prior uncertainty thus tests whether there is evidence for a very big effect size for the predictor term in question, which is usually not supported by the data (because the diffuse prior covers implausibly large effect sizes). When using a weakly informative or even an informative prior, with much smaller uncertainty, to the contrary, the Bayes factor tests whether there is evidence for a small effect of the additional predictor term, which is much more likely to be the case. Thus, using prior knowledge and specifying priors with reasonable uncertainty (rather than diffuse priors with large uncertainty) are crucial in model comparison using Bayes factors.

The described process ends up simulating from the *prior predictive distribution*, which specifies how the prior interacts with the likelihood. Mathematically, it computes an average (integral) over different possible (prior) parameter values. The prior predictive distribution is:

$$\pi(y) = \int \pi(y, \theta)\ \mathrm{d}\theta = \int \pi(y \mid \theta)\pi(\theta)\ \mathrm{d}\theta = \int \text{likelihood}(y \mid \theta) \cdot \text{prior}(\theta)\ \mathrm{d}\theta \tag{2}$$

As a concrete example, suppose we assume that our likelihood is a Normal distribution with mean μ and standard deviation σ. Suppose that we now define the following priors on the parameters: $\mu \sim Normal(0, 1)$, and $\sigma \sim Uniform(1, 2)$. We can generate the prior predictive distribution using the following steps:

- Do the following 100,000 times:
  - Take one sample m from a Normal(0,1) distribution
  - Take one sample s from a Uniform(1,2) distribution

  - Generate and save a data point from Normal(m,s)
- The generated data is the prior predictive distribution.

More complex generative processes involving repeated measures data can also be defined.

Computational faithfulness: Testing for correct posterior approximations

A key aim in Bayesian data analysis is to compute posterior expectations, such as the posterior mean or posterior credible intervals (quantiles) of some parameter. For some simple models and prior distributions, these posterior expectations can be computed exactly by analytical derivation. However, this is not possible in most interesting and realistic more complex models, where analytical solutions cannot be computed. Instead, computational approximations are needed for estimation. One possible approximation is variational Bayes, where parameterized probability density functions (e.g., the Gaussian distribution) are used for approximate posterior inference: the function parameters are estimated such that the function approximates the posterior as closely as possible. Here, we use another option: while it is often not possible to compute the posterior exactly, it is possible to draw samples from it, and we accordingly use (MCMC) sampling to approximate posterior expectations.

Approximations of posterior expectations can be inaccurate. For example, the computer program built to sample from a posterior can be erroneous. This could involve an error in the specification of the likelihood, or insufficient sampling of the full density of the posterior. The sampler may be biased by sampling parameter values that are larger or smaller than the true posterior, or the variance of the posterior samples may be larger or smaller than the true posterior uncertainty.

Given that posterior approximations can be inaccurate, it is important to design a procedure to test whether the posterior approximation of choice is indeed accurate, e.g., that

the software used to implement the sampling works without errors for the specific problem at hand.

To test software, it is possible to use it in situations where there is a known "correct answer" and to compare the correct result with what the software generates. This approach is more difficult for testing Bayesian estimation software, which can be stochastic. Here, an alternative can be to randomly sample model parameters from the prior distribution, then sample fake data from the model (i.e., the likelihood function) given the sampled parameters, and perform Bayesian inference on the simulated fake data. If the Bayesian estimation software works correctly, then on average the obtained posterior will be correct. This means, for example, that over an ensemble of such simulations the true parameter values will be contained in *any* 95% posterior credible intervals in approximately 95% of the simulations. (A 95% credible interval indicates a Bayesian interval in which 95% of the posterior probability mass is contained.) Note that there are many possible 95% credible intervals — within these simulations the average coverage will be 0.95 for all of them.

A powerful method for testing whether the posterior approximation of a software is correct is provided by simulation-based calibration (SBC) (Talts et al., 2018). It tests not only the correctness of 95% or e.g., also of 75% posterior credible intervals, but systematically tests the correctness of the whole posterior approximation. This is possible as it can be shown (see below) that on average the posterior looks like the prior. That is, if we repeatedly sample from the prior and then from the data, and compute many different posteriors from the sampled fake data, then the resulting ensemble of posteriors can be compared to the prior, and the average over the ensemble of posteriors should be the same as the prior.

Mathematically, it can be shown that

$$\pi(\theta') = \int \int \pi(\theta' \mid y)\pi(y \mid \theta)\pi(\theta) \, \mathrm{d}y\mathrm{d}\theta \tag{3}$$

That is, we draw a sample θ from the prior, $\pi(\theta)$, simulate some data from the likelihood,

$\pi(y \mid \theta)$, and then estimate posterior parameters, $\pi(\theta' \mid y)$, from the simulated data. When we next take the average of the posterior over different simulated true parameters ($\int \mathrm{d}\theta$) and over different simulated data-sets ($\int \mathrm{d}y$), we will obtain a posterior that recovers the prior distribution ($\pi(\theta')$).

In practice, to conduct SBC, we use the following procedure:

1. Take the prior $\pi(\theta)$ and randomly draw a parameter set $\tilde{\theta}$ from it: $\tilde{\theta} \sim \pi(\theta)$
2. Use this parameter set $\tilde{\theta}$ to simulate hypothetical data $\tilde{y}$ from the model: $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
3. Fit the model to this hypothetical data and draw samples from the posterior distribution: $\tilde{\theta}' \sim \pi(\theta \mid \tilde{y})$

We repeat steps 1 to 3 many times. In each cycle, we can compare posterior samples to the parameter set (from step 2.) used to simulate the hypothetical data. We record for each run where in the posterior distribution the prior parameters lie. If the distributions of the posterior samples and the sampled prior parameters are the same, the prior parameters should equally frequently lie at every location (i.e., rank) within the distribution of the posterior. Collecting all these locations gives an (ensemble) posterior sample of $\tilde{\theta}'$ values.

Accordingly, in simulation-based calibration (Talts et al., 2018), we take each simulated prior parameter value and test where it is located within the estimated posterior distribution by computing its rank statistic within the posterior. Said differently, we count the number of posterior parameter samples $\tilde{\theta}'_r$ that are larger than the given prior parameter $\tilde{\theta}$: $\rho = \#\{\tilde{\theta} < \tilde{\theta}'_r\}$. We perform this calculation repeatedly, for every sampled prior parameter set. The resulting rank statistic of the prior parameters can be plotted as a histogram. As a central result of SBC (Cook, Gelman, & Rubin, 2006; Talts et al., 2018), if the posterior model fitting works accurately, then the rank statistics are exactly uniformly distributed. Different patterns of how the distribution deviates from unity can diagnose different specific problems of the posterior. We here illustrate some examples (see Fig. 3) (taken from Talts et

al., 2018).

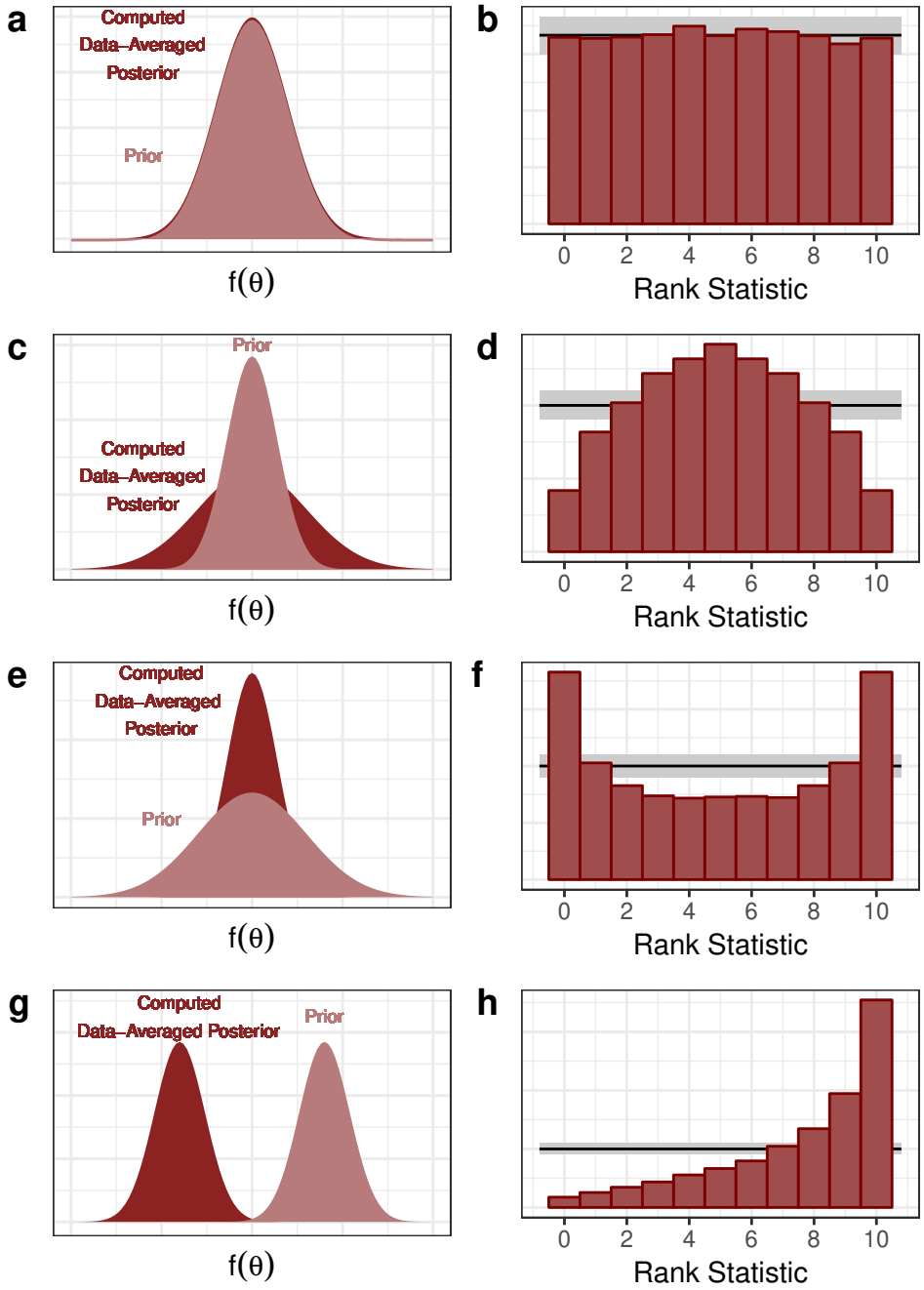

Figure 3. Exemplary results in simulation-based calibration (SBC). a)+b) If the data-averaged posterior exactly reflects the prior (a) then the SBC histogram is uniformly distributed, indicating correct posterior approximation. c)-d) When the SBC histogram shows an inverse U-shaped form (d), then this indicates that the data-averaged posterior is over-dispersed (c), that is, that it has higher variance than the prior. e)-f) An SBC histogram showing a symmetric U-shape (f) indicates that the data-averaged posterior is under-dispersed (e), that is, it has lower variance than the prior. g)-h) If the SBC histogram is asymmetric (h), then the posterior will be biased in the opposite direction (g).

Note that even highly powerful tools like Hamiltonian Monte Carlo (HMC) sampling do not always provide accurate posterior estimation, and that it's therefore important to check computational faithfulness for a given experimental design, model, and data-set. Only when we see that our posterior computations are accurate and faithful can we take the next step, namely looking at the sensitivity of the model analyses.

Model sensitivity

What can we realistically expect from the posterior of a model, and how can we check whether these expectations are justified for the current setup? First, we might expect that the posterior recovers the true parameters generating the data without bias. That is, when we simulate hypothetical data based on a true parameter value, we may expect that the posterior mean exhibits the same value. Although desirable, however, this expectation may or may not be justified for a given model, experimental design, and data-set. Indeed, parameter estimation for some, e.g., non-linear, models may be biased, such that the true value of the parameter can practically not be recovered from the data. At the same time, we might expect from the posterior that it is highly informative with respect to the parameters that generated the data. That is, we may hope for small posterior uncertainty relative to our prior knowledge, that is, e.g., a small posterior standard deviation. Just as with the mean, however, the certainty in a posterior may not always be high. Some experimental designs, models, or data-sets may yield highly uninformative estimates, where uncertainty is not reduced compared to our prior information. This, for example can be the case when we have very little data, or when the experimental design does not allow us to constrain certain model parameters.

To study model sensitivity, we here investigate two questions about the model:

1) How well does the estimated posterior mean match the true simulating parameter?

2) How much is uncertainty reduced from the prior to the posterior?

First, to determine the distance of the posterior mean from the true simulating parameter, it is possible to compute a *posterior z-score*:

$$z = \frac{\mu_{post} - \tilde{\theta}}{\sigma_{post}} \tag{4}$$

Here, the posterior mean μ_{post} is compared to the true simulating parameter $\tilde{\theta}$, and the difference between them is scaled by the posterior uncertainty (standard deviation, σ_{post}). This measure estimates how closely the posterior mean is to the truth. Small (absolute) values close to zero indicate that the posterior mean is close to the true parameter value, and large (absolute) values indicate that the posterior mean is far off the true generating model parameter. Note that large positive values versus large negative values indicate different positive versus negative biases in the posterior estimation.

Second, *posterior contraction* estimates how much prior uncertainty is reduced in the posterior estimation:

$$s = 1 - \frac{\sigma^2_{post}}{\sigma^2_{prior}} \tag{5}$$

Here, the variance of the posterior distribution, σ^2_{post}, is divided by the prior variance, σ^2_{prior}. In general, additional information from the likelihood will reduce uncertainty, such that the posterior variance will be smaller than the prior variance. If the data is highly informative, then the variance in the estimate is strongly reduced, and there will be strong posterior contraction s close to 1. However, when the data provide little information, then the posterior variance will be of similar size as the prior variance, and posterior contraction s will be close to 0.

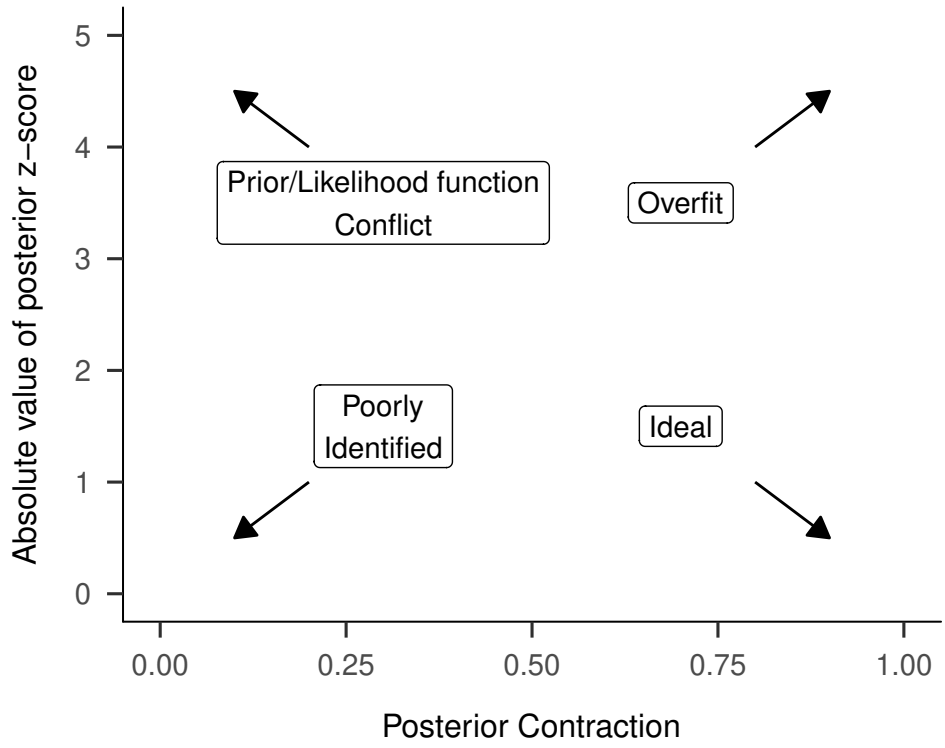

Figure 4. Posterior z-Scores as a function of Posterior Contraction. Arrows show four possible results and their interpretation. The combination of high posterior contraction with large posterior z-scores reflects situations of overfitting to noise in the data. Low posterior contraction with small z-scores reflect a poorly identified model. Low contraction with large z-scores indicate a substantial conflict between the prior and the observations. Last, high posterior contraction and low posterior z-scores reflect an ideal situation of good model fit.

To obtain these estimates, we take the following steps:

1. Take the prior $\pi(\theta)$ and randomly draw a parameter set $\tilde{\theta}$ from it: $\tilde{\theta} \sim \pi(\theta)$
2. Use this parameter set $\tilde{\theta}$ to simulate hypothetical data $\tilde{y}$ from the model: $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
3. Fit the model to this hypothetical data and draw samples from the posterior distribution: $\tilde{\theta}' \sim \pi(\theta \mid \tilde{y})$
4. Compute the *posterior z-score* and the *posterior contraction* for each sample of posterior parameters $\tilde{\theta}'$

The distribution of *posterior z-scores* and *posterior contractions* can be plotted in a scatterplot (see Fig. 4), which provides a useful model diagnostic: results in the upper right corner indicate overfitting of the posterior to a wrong parameter value; results in the lower left corner indicate that the model is poorly identified, i.e., that the data do not well constrain estimation of model parameters; the upper left corner reflects a situation of substantial conflict between the prior and the likelihood function; and the lower right corner indicates the ideal situation of correct estimates with low uncertainty.

Importantly, the scatter plot doesn't provide a test for rejecting the current model. Instead its intent is to provide key information about the current setup, and to help us make a decision about the efficacy of the model, that is, whether the model allows us to obtain good estimates in this experimental design at all. Note that this can vary considerably over specific collected (or drawn) data-sets. Even with the model and the experimental design held constant, the model may be highly sensitive for some data-sets, but exhibit problems for another. This is one reason why we assess sensitivity for a range of different simulating parameter values covered by the prior distributions, to obtain information about the range of possible outcomes.

Note that this way of visualizing accuracy and contraction is just a general means of evaluating the utility of a model. More specific inferential goals can motivate more specific evaluations. For example, if we want to make a binary decision, such as $\mid \theta \mid < 0.1$, then we might look at the distribution of false discovery rates and true discovery rates.

Posterior predictive checks: Does the model adequately capture the data?

"*All models are wrong, but some models are useful.*" (Box, 1976) We know that our model probably does not fully capture the true data generating process, which is noisily reflected in the observed data. Our question therefore is whether our model is *close enough*

to the true process that has generated the data, and whether the model is useful for informing our scientific question. To compare the model to the true data generating process (i.e., to the data), we can simulate data from the model and compare the simulated to the real data. This can be formulated via a *posterior predictive distribution*: the model is fit to the data, and the estimated posterior model parameters are used to simulate new data. The question then is how close the simulated data is to the observed data.

One way to assess this is using Bayesian cross validation or one of the many information criteria, such as BIC, DIC, or WAIC (e.g., Spiegelhalter, Best, Carlin, & Linde, 2014; Vehtari, Gelman, & Gabry, 2017). However, this approach only allows for relative comparison between different models, but not for an absolute measure of model fit. Moreover, the information criteria are only approximations and hence can be misleading when the approximation is inaccurate. The information criteria also consider the entire fit of the model, and hence can't differentiate between relevant aspects and irrelevant aspects.

An alternative approach is to use features of the data that we care about, and to test how well the model can capture these. Indeed, we had already defined summary statistics in the prior predictive checks. We can now compute these summary statistics for the data simulated from the posterior predictive distribution. This will yield a distribution for each summary statistic. We moreover compute the summary statistic for the observed data, and can now see whether the data falls within the distribution of the model predictions (cf. Fig. 5a), or whether the model predictions are far off the observed data (see Fig. 5b). If the data is in the distribution of the model, then this supports model adequacy. If, to the contrary, we observe a large discrepancy, then this indicates that our model likely misses some important structure of the true process that has generated the data, and that we have to consider our domain expertise to further improve the model. Alternatively, however, a large discrepancy can be due to the data being an extreme observation, which was nevertheless generated by the process captured in our model. Note that in general we can't

discriminate between these two possibilities. Consequently, we have to use our best judgement as to which possibility is more relevant, in particular changing the model only if the discrepancy is consistent with a known missing model feature.

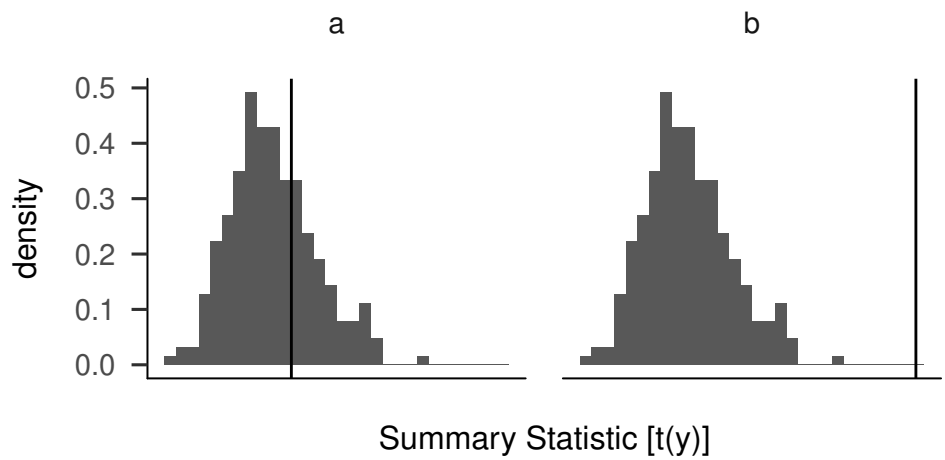

Figure 5. Posterior predictive checks. Compare posterior model predictions (histogram) with observed data (vertical line) for a specific summary statistic, t(y). a) This displays a case where the observed summary statistic (vertical line) lies within the posterior model predictions (histogram). b) This displays a case where the summary statistic of the observed data (vertical line) lies clearly outside of what the model predicts a posteriori (histogram).

Mathematically, the *posterior predictive distribution* is written:

$$\pi(y_{pred} \mid y) = \int \pi(y_{pred} \mid \theta)\pi(\theta \mid y)\, \mathrm{d}\theta \tag{6}$$

Here, the observed data y is used to infer the posterior distribution over model parameters ($\pi(\theta \mid y)$). This is combined with the model or likelihood function ($\pi(y_{pred} \mid \theta)$) to yield new, now simulated, data y_{pred}. The integral $\int \mathrm{d}\theta$ indicates averaging across different possible values for the posterior model parameters (θ).

We can't evaluate this integral exactly: θ can be a vector of many parameters, making this a very complicated integral with no analytical solution. However, we can approximate it using sampling. Specifically, we can obtain samples from the posterior distribution, e.g.,

using HMC or a different MCMC sampling scheme. We can now use each of the posterior samples as parameters to simulate new data from the model. This procedure then approximates the integral and yields an approximation to the posterior predictive distribution.

Exemplary data analysis

We perform an exemplary analysis of a data-set from Gibson and Wu (2013). The methodology they used is called self-paced reading; this method is commonly used in psycholinguistics as a cheaper and faster substitute to eyetracking during reading. The participant is seated in front of a computer screen and is initially shown a series of broken lines that mask words from a complete sentence. The participant then unmasks the first word (or phrase) by pressing the space bar. Upon pressing the space bar again, the second word/phrase is unmasked and the first word/phrase is masked again; the time in milliseconds that elapsed between these two space-bar presses counts as the reading time for the first word/phrase. In this way, the reading time for each successive word/phrase in the sentence is recorded. Usually, the participant is also asked a yes/no question at the end of the trial, about the sentence. This is intended to ensure that the participant is adequately attending to the meaning of the sentence.

Gibson and Wu collected self-paced reading data using Chinese relative clauses. Relative clauses are sentences like: *The student who praised the teacher was very happy.* Here, the head noun, *student*, is modified by a relative clause *who... teacher*, and the head noun is the subject of the relative clause as well: the student praised the teacher. Such relative clauses are called subject relatives. By contrast, one can also have object relative clauses, where the head noun is modified by a relative clause which takes the head noun as an object. An example is: *The student whom the teacher praised was very happy.* Here, the teacher praised the student. Gibson and Wu were interested in testing the hypothesis that

Chinese shows an object relative (OR) processing advantage relative to subject relatives (SR). The theoretical reasons for this are not interesting for the present purposes. Their experimental design had one factor with two levels: (i) object relative sentence, and (ii) subject relative sentence. We use sum coding (-1, +1) for this factor, which we call "so", an abbreviation for subject-object. Following Gibson and Wu (2013), we analyze reading time on a target word, which was the head noun of the relative clause; in Chinese, unlike English, the head noun appears after the relative clause. By the time the participant reads the head noun, they already know whether they are reading a subject or an object relative. The theory being tested here states that the meaning of the relative clause is resolved at the head noun and that in Chinese, this meaning resolution process is easier in object relatives vs. subject relatives.

The data-set contains reading time measurements in milliseconds from 37 subjects and from 15 items. The design is a classic repeated measures Latin square design; there were originally 16 items, but one item had a typo and had to be removed. Some participants were also removed. We analyze the data using the R function *brm* in the *brms* package (Bürkner, 2017), which provides an (*lme4*-style syntax) interface to fit hierarchical (non-)linear models in the probabilistic programming language Stan.

Prior predictive checks

The first step in Bayesian data analysis is to specify the statistical model and the priors for the model parameters. In *brms* the latter can be done using the function `set_prior()`. One possible standard setup for diffuse priors which is sometimes used in reading studies (e.g., Paape, Nicenboim, & Vasishth, 2017; Vasishth, Mertzen, Jäger, & Gelman, 2018) is as follows: For the intercept we use a normal distribution with mean 0 and standard deviation 10. Note that this is on the log scale as we assume a lognormal distribution of reading times. That is, this approach assumes a priori that the intercept for reading times varies between 0

seconds and (one standard deviation) $exp(10) = 22,026$ ms (i.e., 22 sec) or (two standard deviations) $exp(20) = 485,165,195$ ms (i.e., 135 hours). Note that going from seconds to hours within one standard deviation shows how diffuse this prior is. In *brms* this is specified as: `set_prior("normal(0, 10)", class = "Intercept")`. That is, we can specify a distribution (`normal(0, 10)`), and define the class of parameters for which this should apply; here, we specify the prior should apply to the intercept (`class = "Intercept"`). Moreover, for the effect of linguistic manipulations on reading times, one common standard prior is to assume a mean of 0 and a standard deviation of 1 (also on the log scale). Note that the prior on the effect size on log scale is a multiplicative factor, that is, the prediction for the effect size depends on the intercept. For an intercept of $exp(6) = 403$ ms, a variation to one standard deviation above multiples the base effect by 2.71, increasing the mean from 403 to $exp(6) \times exp(1) = 1097$. Likewise a variation to one standard deviation below mutliplies the base effect by 1/2.71, decreasing the mean from 403 to $exp(6) \times exp(-1) = 148$. This effect size is strongly changed when assuming a different intercept: for a slightly smaller value for the intercept of $exp(5) = 148$ ms, the expected condition difference is reduced to 37% (349 ms), and for a slightly larger value for the intercept of $exp(7) = 1097$ ms, the condition difference is enhanced to 272% (2578 ms). Here, we use this prior for the difference between object-relative and subject-relative sentences (i.e., the slope), and write this as `set_prior("normal(0, 1)", class = "b", coef="so")`, where `class = "b"` indicates that all fixed effects share this prior, and `coef="so"` restricts it to the effect of the slope (OR - SR). We use the same `normal(0, 1)` prior for the random effects standard deviations (`class = "sd"`) and for the residual variance (`class = "sigma"`). Finally, for the random effects correlation between the intercept and the slope, we use an `lkj` prior (Lewandowski, Kurowicka, & Joe, 2009) with diffuse prior parameter value of 2 (for visualization of the prior see Fig. 6). We store these priors in an R object called `priors`:

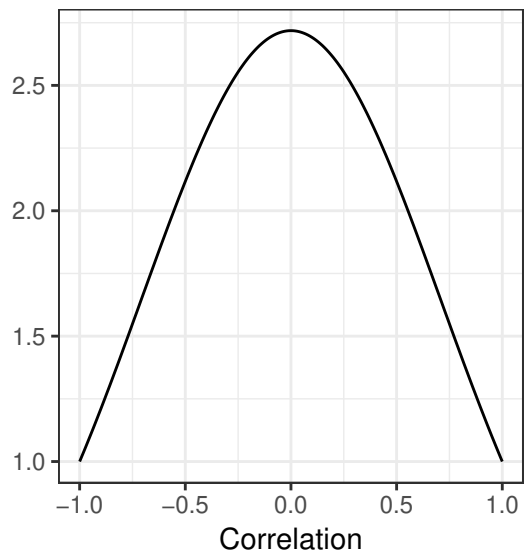

Figure 6. Shape of the LKJ prior. This is the prior density for the random effects correlation parameter, here used as a prior for the correlation between the effect size (so) and the intercept. The shape shows that correlation estimates close to zero are expected, and that very strong positive correlations (close to 1) or negative correlations (close to -1) are increasingly unlikely. Thus, correlation estimates are regularized towards values of zero.

```
priors <- c(set_prior("normal(0, 10)", class = "Intercept"),
            set_prior("normal(0, 1)", class = "b", coef="so"),
            set_prior("normal(0, 1)", class = "sd"),
            set_prior("normal(0, 1)", class = "sigma"),
            set_prior("lkj(2)", class = "cor"))
```

For the prior predictive checks, we use these priors to draw random parameter sets from the distributions, and to simulate hypothetical data using the statistical model. As a statistical model, we use the so-called maximal model (Barr et al., 2013) for the design. Such a model includes fixed effects for the intercept and the slope (coded using sum contrast coding: +1 for object relatives, and -1 for subject relatives), correlated random intercepts and slopes for participants (called subjects in the data frame), and correlated random

intercepts and slopes for items. In `brms` syntax: `rt ~ 1+so + (1+so|subj) + (1+so|item)` and `family=lognormal()`. We load the data to extract the experimental design. We next use our custom R functions `SimFromPrior()` and `genfake()` to simulate parameters from the priors and to simulate data from the model.

```
# prepare data
gw    <- read.table("data/gibsonwu2012data.txt", header=TRUE) # load data
gw$so <- ifelse(gw$type%in%c("subj-ext"),-1,1) # sum-coding for predictor
gw1   <- subset(gw,region=="headnoun") # subset critical region
expdesign <- gw1[,c("subj","item","so")] # extract experimental design
source("scripts/SimFromPrior.R") # load simulation functions
source("scripts/genfake.R")
# prepare simulation
nsim <- 1000
Nsj  <- max(unique(expdesign$subj)) # determine no. of subjects
Nit  <- max(unique(expdesign$item)) # determine no. of items
beta0 <- beta1 <- sigma_u0 <- sigma_u1 <- sigma_w0 <- sigma_w1 <-
  rho_u <- rho_w <- sigma <- NA
rtfakemat <- matrix(NA,nrow(expdesign),nsim)
set.seed(123)
# simulations
for (i in 1:nsim) {
  # simulate parameters
  beta0[i] <- -1; while (beta0[i]<0) # restrict to positive values
     beta0[i] <- SimFromPrior(priors,class="Intercept")
  beta1[i]    <- SimFromPrior(priors,class="b", coef="so")
  sigma_u0[i] <- SimFromPrior(priors,class="sd")
  sigma_u1[i] <- SimFromPrior(priors,class="sd")
```

```
  sigma_w0[i] <- SimFromPrior(priors,class="sd")
  sigma_w1[i] <- SimFromPrior(priors,class="sd")
  rho_u[i]    <- SimFromPrior(priors,class="cor")
  rho_w[i]    <- SimFromPrior(priors,class="cor")
  sigma[i]    <- SimFromPrior(priors,class="sigma")
  # simulate data
  rtfakemat[,i] <- genfake(expdesign, Nsj, Nit,
          beta0    = beta0[i],    beta1    = beta1[i],
          sigma_u0 = sigma_u0[i], sigma_u1 = sigma_u1[i],
          sigma_w0 = sigma_w0[i], sigma_w1 = sigma_w1[i],
          rho_u    = rho_u[i],    rho_w    = rho_w[i],
          sigma    = sigma[i])
}
truePars <- data.frame(beta0,beta1,sigma_u0,sigma_u1,sigma_w0,sigma_w1,
                       rho_u,rho_w,sigma)
```

Based on the simulated data we can now perform prior predictive checks: we compute summary statistics, and plot the distributions of the summary statistic across simulated data-sets. First, we visualize the distribution of the simulated data. For a single data-set, this could be visualized as a histogram. Here, we have a large number of simulated data-sets, and thus a large number of histograms. We represent this uncertainty: for each bin, we plot the median as well as quantiles showing where 10%-90%, 20%-80%, 30%-70%, and 40%-60% of the histograms lie.

```
# Colour codes to use
c_light <- "#DCBCBC"; c_light_highlight <- "#C79999"
c_mid   <- "#B97C7C"; c_mid_highlight   <- "#A25050"
c_dark  <- "#8F2727"; c_dark_highlight  <- "#7C0000"
```

```
# set very large data points to a value of 2000
rtfakematH <- rtfakemat; rtfakematH[rtfakematH>2000] <- 2000
# Compute one histogram per simulated data-set
binwidth <- 20
breaks <- seq(0,max(rtfakematH,na.rm=TRUE)+binwidth,binwidth)
histmat <- matrix(NA,ncol=nsim,nrow=length(breaks)-1)
for (i in 1:nsim)
  histmat[,i] <- hist(rtfakematH[,i],breaks=breaks,plot=FALSE)$counts
# For each bin, compute quantiles across histograms
probs <- seq(0.1,0.9,0.1)
quantmat<- as.data.frame(matrix(NA,nrow=dim(histmat)[1],ncol=length(probs)))
names(quantmat) <- paste0("p",probs)
for (i in 1:dim(histmat)[1])
  quantmat[i,] <- quantile(histmat[i,],p=probs)
quantmat$x <- breaks[2:length(breaks)] - binwidth/2 # add bin mean
# Plot
FigPri1a <- ggplot(data=quantmat, aes(x=x))+
  geom_ribbon(aes(ymax=p0.9, ymin=p0.1), fill=c_light) +
  geom_ribbon(aes(ymax=p0.8, ymin=p0.2), fill=c_light_highlight) +
  geom_ribbon(aes(ymax=p0.7, ymin=p0.3), fill=c_mid) +
  geom_ribbon(aes(ymax=p0.6, ymin=p0.4), fill=c_mid_highlight) +
  geom_line(aes(y=p0.5), colour=c_dark, size=1) +
  labs(title="Prior Predictive Distribution", y="", x="Reading Time [ms]")
```

For the current prior data simulations, this shows (see Figure 7a) that most of the hypothetical reading times are close to zero or larger than $2,000$ ms. It is immediately visible that the data predicted by this prior follows a very implausible distribution: it looks

exponential, while we would expect a lognormal (or normal) distribution for reading times. Most data points take on extreme values.

As an additional summary statistic, we take a look at the mean per simulated data-set (at the log scale) and also at the variance (ms scale).

```
tmpM  <- apply(log(rtfakemat),2,mean) # Mean
tmpSD <- apply(rtfakemat,2,sd);
tmpSD[tmpSD>2000] <- 2000 # SD
FigPri1b<-ggplot()+stat_bin(aes(x=tmpM), fill=c_dark)+
  labs(x="Mean [log RT]")
FigPri1d<-ggplot()+stat_bin(aes(x=tmpSD), fill=c_dark)+
  labs(x="Standard Deviation [RT]")
```

The results, displayed in Figure 7b+d, show that the mean varies across a wide range, with a substantial number of data-sets having a mean larger than 10 on log scale or of $exp(10) = 22,026$ on ms scale. Again, this reveals a highly unplausible assumption about the intercept parameter. The standard deviation, shown on ms scale for easier readability (Fig. 7d), exhibits a substantial number of values larger than $2,000$, which again is clearly larger than what we would expect for reading times.

Moreover, we also plot the size of the effect of object relative minus subject relative sentence as a measure of effect size.

```
effectSize <- NA
for (i in 1:nsim)
  effectSize[i] <- mean(rtfakemat[expdesign$so==+1,i])-
                   mean(rtfakemat[expdesign$so==-1,i])
effectSize[effectSize> +2000] <- +2000
effectSize[effectSize< -2000] <- -2000
```

```
FigPri1c <- ggplot()+stat_bin(aes(x=effectSize), fill=c_dark)+
         labs(x="Object - Subject [S-O RT]")
```

The results show that our priors commonly assume differences in reading times between conditions of more than $2,000$ ms (see Fig. 7c), which is larger than we would expect for a psycholinguistic manipulation of the kind investigated here. More specifically, given that we model reading times using a lognormal distribution, the expected effect size depends on the value for the intercept. For example, for an intercept of $exp(1) = 2.7$ ms and an effect size in log space of 1 (i.e., one standard deviation of the prior for the effect size), expected reading times for the two conditions are $exp(1-1) = 1$ ms and $exp(1+1) = 7$ ms. For an intercept of $exp(10) = 22,026$ ms, to the contrary, the corresponding reading times for the two conditions would be $exp(10-1) = 8,103$ ms and $exp(10+1) = 59,874$ ms.

Note that this implies highly varying expectations for the effect size, including the possibility for very large effect sizes. However, if we haven't seen an effect before running the experiment, then we would probably expect the effect to be rather small - unless the data were strongly selected for the effect. Thus, priors with smaller expected effect sizes may be reasonable.

We moreover look at individual differences in the effect of object versus subject relatives. We take a look at the participant with the largest (absolute) difference in reading times between object versus subject relatives.

```
# effect size per subject
EffectSizeMax <- EffectSizeSD <- NA
for (i in 1:nsim) { tmp <- NA
  expdesign$rtfake <- rtfakemat[,i]
  tmp <- expdesign%>%group_by(subj,so)%>%summarize(rtfake=mean(rtfake))%>%
    spread(key="so",value="rtfake") %>% mutate(dif = `1` - `-1`)
```

```
  EffectSizeSD[ i] <- sd(     tmp$dif ,na.rm=TRUE)
  EffectSizeMax[i] <- max(abs(tmp$dif),na.rm=TRUE)
}
EffectSizeMax[EffectSizeMax>2000] <- 2000
FigPri1e <- ggplot()+stat_bin(aes(x=EffectSizeMax), fill=c_dark)+
         labs(x="Max Effect Size [S-O RT]")
EffectSizeSD[EffectSizeSD>2000] <- 2000
FigPri1f <- ggplot()+stat_bin(aes(x=EffectSizeSD), fill=c_dark)+
         labs(x="SD Effect Size [S-O RT]")
```

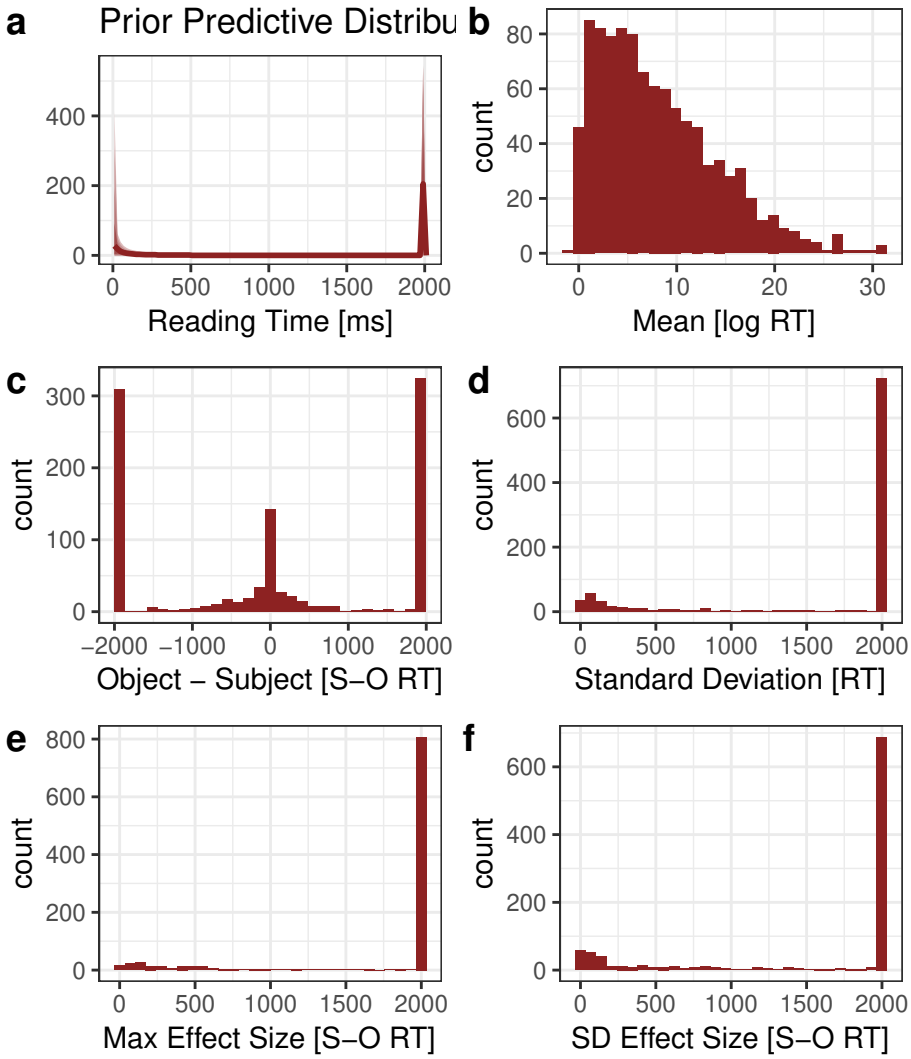

Figure 7. Prior predictive checks for a high-variance prior. Distributions are over simulated hypothetical data. a) Multivariate summary statistic: Distribution of histograms of reading times shows very short and also very long reading times are expected too frequently by the diffuse prior. b)-f) Scalar summary statistics. b) Distribution of average log reading times shows that extremely large reading times, e.g., of exp(10) = 22,026 ms are too frequently expected. c) Distribution of differences in reading times between object minus subject relatives shows that very large effect sizes are far too frequently expected. d) Distribution of standard deviations of reading times shows that very large variances are over-expected in the priors. e) Maximal effect size (object - subject relatives) across subjects again shows far too many extreme values. f) Standard deviation of effect size (object - subject relatives) across subjects; again far too many extreme values are expected. a)+c)-f) Values > 2000 or < -2000 are plotted at a value of 2000 or -2000 for visualization.

The prior simulations show common maximal effect sizes of larger than $2,000$ ms (Fig. 7e), which is more than we would expect for observed data. Similarly, the variance in hypothetical effect sizes is large, with many SDs larger than $2,000$ ms (Fig. 7f), and thus again takes many values that are inconsistent with our domain expertise about reading experiments.

Adjusting priors. Based on these analyses of prior predictive data, we can next use our domain expertise to refine our priors and adjust them to values for which we expect more plausible prior predictive hypothetical data as captured in the summary statistics.

First, we adapt the intercept. Upon re-consideration, we choose a normal distribution in log-space with a mean of 6. This corresponds to an expected grand average reading time of $exp(6) = 403$ ms. For the standard deviation, we use a value of SD $= 0.6$. For these prior values, we expect a strongly reduced mean reading time and a strongly reduced residual standard deviation in the simulated hypothetical data. Moreover, we expect that implausibly small or large values for reading times will no longer be expected. For a visualization of the prior distribution of the intercept parameter in log-space and in ms-space see Figure 8a+b. Note that similar values, e.g., for the standard deviation (e.g., SD $= 0.5$ or 0.7), may yield similar results. Our goal is not to specify a precise value, but rather to use prior parameter values that are qualitatively in line with our domain expertise about expected observed reading time data, and that do not produce highly implausible hypothetical data.

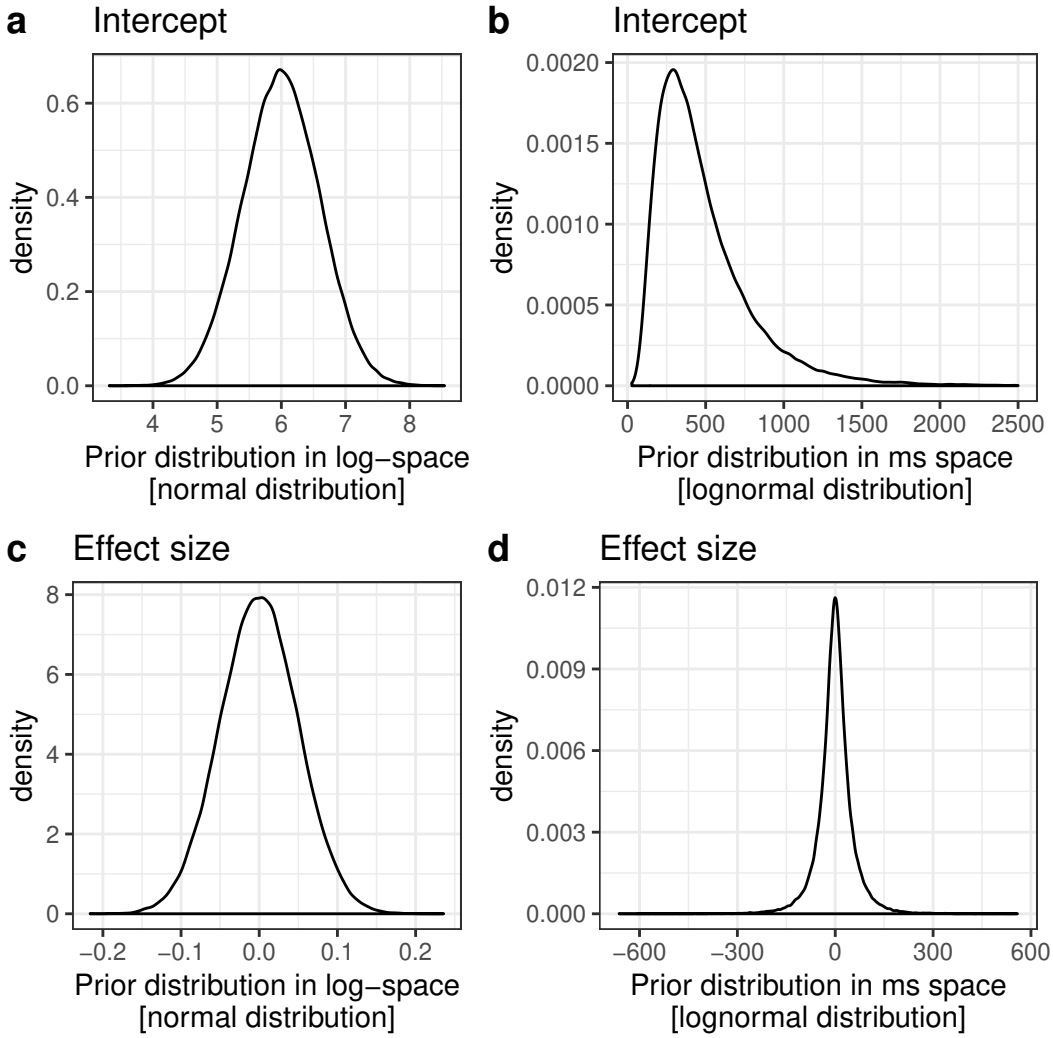

Figure 8. Prior distribution in log-space and in ms-space for a toy example of a linear regression. a) Displays the prior distribution of the intercept in log-space. b) Displays the prior distribution of the intercept in ms-space. c) Displays the prior distribution of the effect size in log-space. d) Displays the prior distribution of the effect size in ms-space.

Next, for the effect of object minus subject relative sentences, we define a normally distributed prior with mean 0 and a much smaller standard deviation of 0.05. Again, we do not have precise information on the specific value for the standard deviation. We expect a generally smaller effect size (see the meta-analysis on Chinese relatives presented in Vasishth, Chen, Li, & Guo, 2013), and we can check through prior predictive checks (data simulation and investigation of summary statistics) whether this yields a plausible pattern of expected results. Figures 8c+d show expected effects in log-scale and in ms-scale for a simple linear regression example.

In addition, we assume much smaller values for the standard deviations in how the intercept and the slope vary across subjects and across items of 0.1, and a smaller residual standard deviation of 0.5. Our expectation for the correlation between random effects is unchanged. In summary:

```
priors2 <- c(set_prior("normal(6, 0.6)", class = "Intercept"),
             set_prior("normal(0, 0.05)", class = "b", coef = "so"),
             set_prior("normal(0, 0.1)", class = "sd"),
             set_prior("normal(0, 0.5)", class = "sigma"),
             set_prior("lkj(2)", class = "cor"))
```

Prior predictive checks for weakly informative priors. Based on this new set of now weakly informative priors, we can again perform prior predictive checks. We again randomly draw samples of parameters from the priors, use these to simulate data from the statistical model, and compute summary statistics for the simulated data. We do not show the R code again for these analyses.

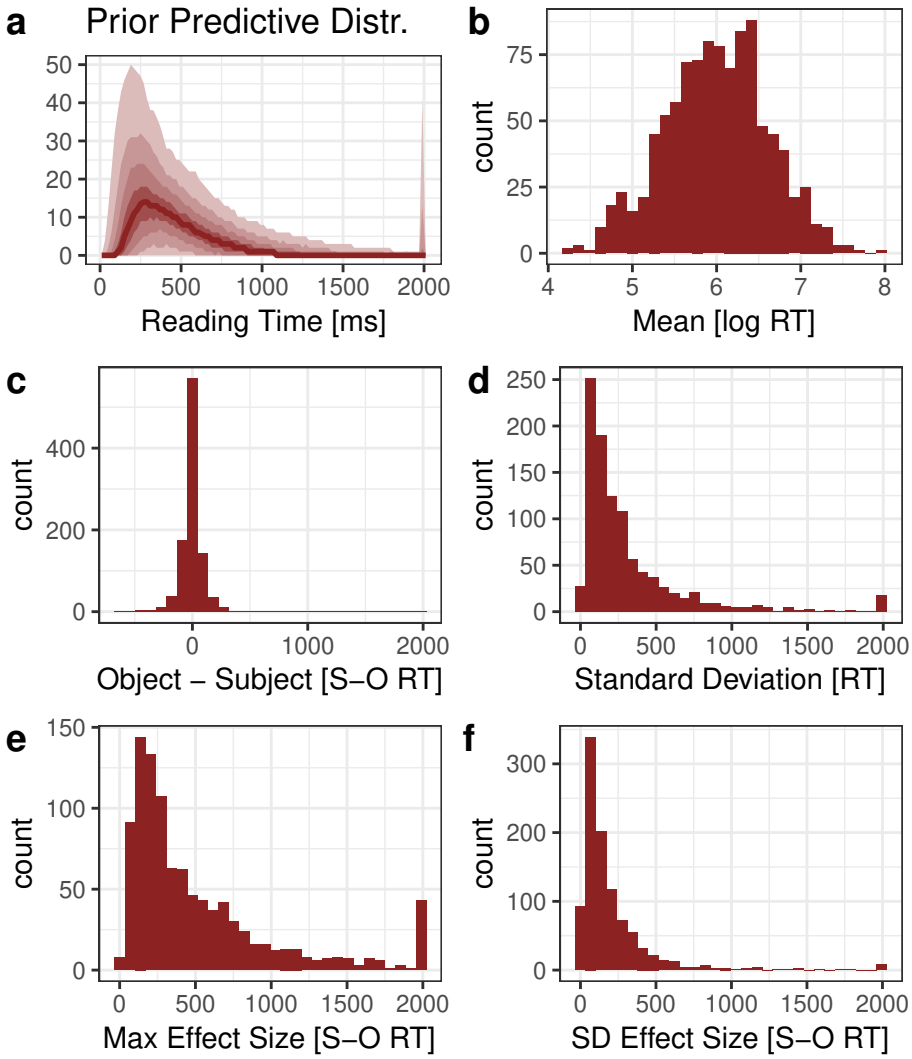

Figure 9. Prior predictive checks for weakly informative prior. Distributions are over simulated hypothetical data. a) Histograms of reading times. Shaded areas correspond to 10-90 percent, 20-80 percent, 30-70 percent, and 40-60 percent quantiles across histograms; the solid line indicates the median across hypothetical data-sets. This now provides a much more reasonable range of expectations. b) Average log reading times now span a more reasonable range of values. c) Differences in reading times between object minus subject relatives; the values are now much more constrained without lots of extreme values. d) Standard deviations of reading times; contrary to the diffuse priors, values are in a reasonable range. e) Maximal effect size (object - subject relatives) across subjects; again, prior expectations are now much more reasonable compared to the diffuse prior. f) Standard deviation of effect size (object - subject relatives) across subjects; this no longer shows a dominance of extreme values any more. a)+c)-f) Values > 2000 or < -2000 are plotted at 2000 or -2000 for visualization.

Figure 9a shows that now the distribution over histograms of the data looks much more reasonable, i.e., more like what we would expect for a histogram of observed data. Very small values for reading times are now rare, and not heavily inflated any more. Moreover, extremely large values for reading times larger than $2,000$ ms are rather unlikely.

We also take a look at the hypothetical average reading times (in log space; Fig. 9b), and find that our expectations are now much more reasonable. We expect average reading times of around $log(6) = 403$ ms. Most of the expected average reading times lie between $log(5) = 148$ ms and $log(7) = 1097$ ms, and few extreme values beyond these numbers are observed. The standard deviations of reading times are also in a much more reasonable range (see Fig. 9d), with only very few values larger than the extreme value of $2,000$ ms.

As a next step, we look at the expected effect size (OR minus SR) in the hypothetical data (Fig. 9c). Extreme values of larger or smaller than $2,000$ ms are now very rare, and most of the absolute values of expected effect sizes are smaller than 200 ms. More specifically, we also check the maximal effect size among all subjects (Fig. 9e). Most of the distribution centers below a value of 1000 ms, reflecting a more plausible range of expected values. Likewise, the standard deviation of the psycholinguistically interesting effect size now rarely takes values larger than 500 ms (Fig. 9f), reflecting more modest assumptions than our first diffuse prior.

Computational faithfulness

We next investigate the computational faithfulness of our computational methods for estimating posterior model parameters for the current experimental design and priors. In Stan and *brm*, lack of divergences and Rhats close to 1 indicate no problem for each individual posterior fit. SBC, to the contrary, aggregates those results together to see if the ensemble shows any indication of inaccurate computation. To investigate accuracy of

computations, we use the simulated data drawn from the prior distributions and fit the statistical model to this simulated data, estimating (approximate) posterior distributions. We use the function *brm* from the `brms` package for model fitting. Note that this process can take a considerable amount of time and computational resources. Parallelizing model fits to multiple cores on a computing cluster can help bring computing time down. While the following steps can ideally be performed for every single model and every single analysis, an alternative path can be to investigate computational faithfulness (and model sensitivity) once for a given research program, where many aspects of the experimental design, the statistical model, and the priors may be repeated across analyses. For example, here we perform the analysis for the Gibson and Wu (2013) design, and could also use these analyses when analyzing a replication of their experiment. First, we estimate the model for all $1,000$ simulated data-sets:

```
expdesign$fakert <- rtfakemat2[,1]
brm1 <- brm(fakert ~ so + (1+so|subj) + (1+so|item), expdesign,
                    family=lognormal(), prior=priors2,
                    control=list(adapt_delta=0.99, max_treedepth=15))
mods_gw <- list(brm1)
for (i in 2:nsim) {
  expdesign$fakert <- rtfakemat2[,i]
  mods_gw[[i]] <- update(brm1, newdata=expdesign, recompile=FALSE)
}
```

Next, we extract the number of divergent transitions of the HMC sampler to diagnose potential problems in model fitting.

```
logPost <- div_trans <- rhat <- neffRatio <- NA
for (i in 1:nsim) { # Extract convergence issues
  logPost[i]   <- log_posterior(mods_gw[[i]])
```

```
  np           <- nuts_params(mods_gw[[i]])
  div_trans[i] <- sum(subset(np, Parameter == "divergent__")$Value)
  rhat[i]      <- rhat(mods_gw[[i]])
  neffRatio[i] <- neff_ratio(mods_gw[[i]])
}
```

```
table(div_trans)
```

```
## div_trans
##    0
## 1000
```

We see that none of the models exhibited a difficulty with divergent samples. Divergent transitions indicate problems in the model fitting, and should be diagnosed and removed. In the present case, setting a control parameter known as "adapt_delta" to a value higher than its default of 0.80 in `brms` removed the divergent transitions. Another convergence diagnostic, $\hat{R}$, is very close to 1 in all of the models (see Fig. 10), indicating no problems in model convergence.

```
hist(rhat,100)
```

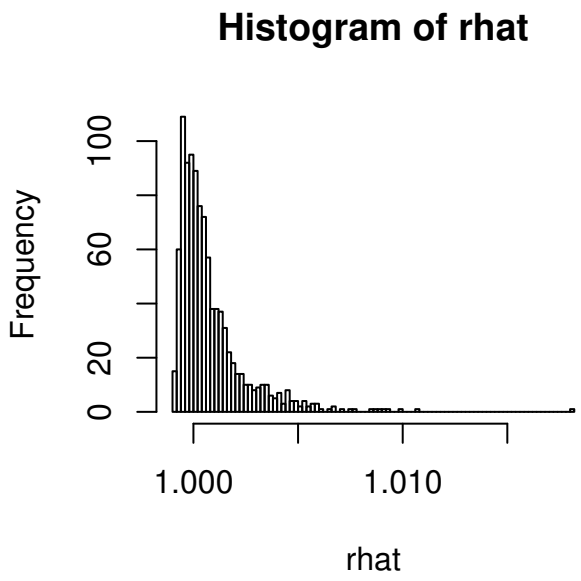

Figure 10. All values of rhat are close to 1, which indicates good model convergence for all of the fitted models.

The above steps suggest that the HMC implementation in Stan has not encountered any signs of an inaccurate fit

Next, we perform simulation-based calibration (SBC) (Talts et al., 2018). This method aims to determine whether estimated posterior model parameters follow the same distribution as the prior model parameters used to generate the data. It does so by comparing posterior estimates with the prior parameters used for simulating data. We performed simulation-based calibration on the current data-set by computing for each simulated data-set the rank of the prior parameter sample within the posterior samples. More specifically, we compute the number of posterior samples that are larger than the prior (simulating) parameter. If the posterior distributions accurately estimate the distribution of the parameter, then their distribution should be the same as the distribution of the actual parameters used to generate the data, and the ranks should be uniformly distributed.

Note that SBC presumes that posterior samples are independent and not correlated. Our HMC samples, however, do exhibit autocorrelation. To remove the autocorrelation from the posterior samples, we thin our samples by taking only every eighth sample. To test how many samples should be removed, it's possible to investigate the autocorrelation in the samples.

```
sbc_rank <- NA
for (i in 1:nsim) { # Compute SBC rank
  postgw      <- posterior_samples(mods_gw[[i]],"^b")
  idx_so      <- seq(1,nrow(postgw),8) # thin to remove autocorrelation
  sbc_rank[i] <- sum( truePars2$beta1[i] < postgw$b_so[idx_so] )
}
est_pars <- data.frame(sbc_rank)
```

Next, we plot a histogram of SBC ranks.

```
hbins <- seq(0,500,25)-0.5
binom <- qbinom(c(0.005,0.5,0.995), length(sbc_rank), 1/(length(hbins)-1))
pdat  <- data.frame(min=binom[1],med=binom[2],max=binom[3],x=c(-20,520))
ggplot()+
  geom_ribbon(data=pdat, aes(x=x, ymin=min, ymax=max), fill="grey80")+
  geom_line(  data=pdat, aes(x=x,y=med)) +
  stat_bin(data=est_pars, aes(x=sbc_rank), breaks=hbins,
           fill=c_dark, colour=c_dark_highlight) +
  scale_x_continuous(breaks=seq(0,500,100))+
  labs(x="Prior Rank",y="", title="Weakly informative priors")
```

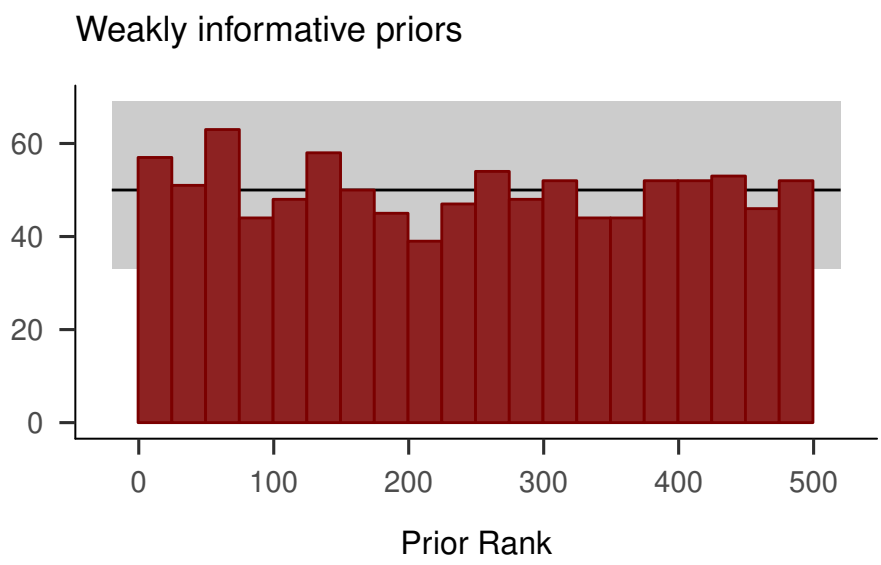

Figure 11. Simulation-based calibration. Histogram of simulation-based ranks of true parameters used for simulating data (randomly drawn from the prior distributions) within posterior samples fit to the simulated data. Shown for weakly informative priors. The results show that the red bars are all contained within the grey horizontal bar in the background, indicating the SBC ranks are uniformly distributed. This shows that the posterior recovers the prior distribution well and that the posterior is thus estimated accurately.

In SBC, good recovery of the true simulating parameters in a posterior analysis is evident when the SBC ranks are uniformly distributed. Figure 11 shows the histogram of SBC ranks. The grey bar in the background reflects the range of values to be expected based on a uniform distribution. The results do not show any evidence of divergence from the uniform distribution. This result suggests that we can trust the posterior estimates from the *brm* function for the current experimental design, statistical model, and prior distributions, that posterior samples do not exhibit bias, but instead that the samples from the posterior follow the same distribution as the prior.

Note, however, that good recovery of model parameters is also possible when using more diffuse priors. As discussed above, diffuse priors have been used by many previous

reading studies, choosing e.g., as the prior for the size of an experimental effect (here SR versus OR) a normal distribution with mean zero and standard deviation one. Figure 12 shows that such diffuse priors support similarly accurate posterior computations.

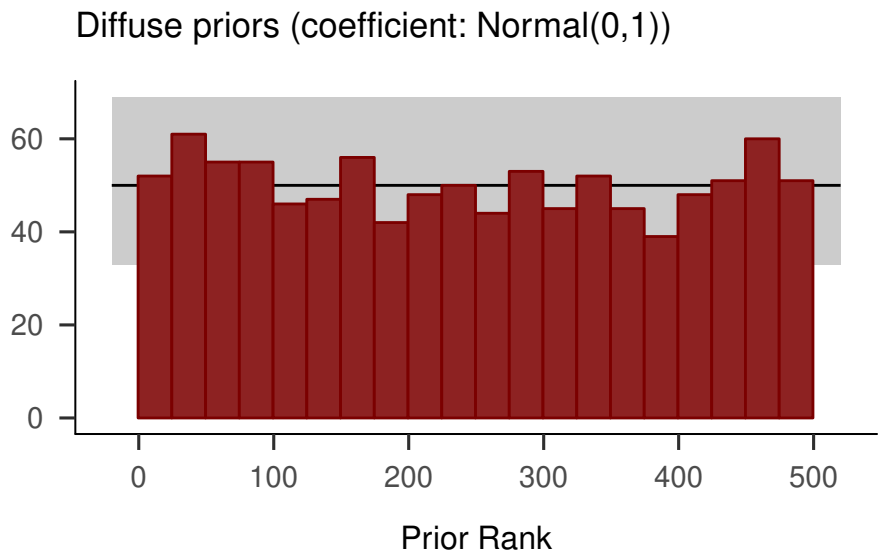

Figure 12. Simulation-based calibration for diffuse priors (intercept: Normal(0,10); coefficients: Normal(0,1)). This plot shows that, as for weakly informative priors, the SBC samples are uniformly distributed, demonstrating that the computational methods work accurately also for the more diffuse priors.

Note that here we only investigated the slope parameter that estimates the difference in reading times between object relative and subject relative sentences (in log space). This is the parameter of biggest theoretical interest to us. Similar analyses, however, are possible for all other model parameters to investigate whether the given computational methods provide accurate posterior estimates. Examples for other parameters of interest could be the standard deviation of the experimental effect across subjects, or the correlations of the effect with the intercept across subjects or items. We point out that if the researcher actually cares only about one phenomenon, then the accuracy of that one effect is all one needs to check. That is, as long as the model provides good faithfulness for the effect of interest, it does not matter or hurt too much if other and irrelevant effects in the model are estimated more

poorly.

Last, we note that computational faithfulness seems to be an issue in frequentist approaches to standard linear mixed-effects models (LMMs), where maximal LMMs frequently encounter difficulties with model fitting. Alternatively, the HMC methods implemented in Stan and accessed using the brm function may be well able to cope with such models (even when using rather vague priors), such that computational faithfulness may be less of an issue with standard LMMs here. In brm the formulation of the likelihood is moreover highly standardized, preventing errors in its formulation. Testing computational faithfulness, moreover, will become very important when we define more customized models - which is exactly one of the advantages of this framework, that this is possible (Nicenboim & Vasishth, 2018).

Model sensitivity

The next step is to investigate how sensitive estimated posterior model parameters are to the true simulating parameters for the current experimental design, statistical model, and set of prior parameters. We compute posterior z-scores to assess deviation of estimated means from true means, and compute posterior contraction to investigate how much information is gained in the posterior relative to the prior, that is, how much the uncertainty about a parameter of interest is reduced. We study this for the theoretically important effect size of object versus subject relative sentences.

```
z_score <- contraction <- NA
for (i in 1:nsim) {
  prior_sd_beta1  <- 0.05 # prior standard deviation
  post_mean_beta1 <- fixef(mods_gw[[i]])[2,1] # posterior mean
  post_sd_beta1   <- fixef(mods_gw[[i]])[2,2] # posterior sd
```

```
  z_score[i]      <- (post_mean_beta1 - truePars2$beta1[i]) / post_sd_beta1
  contraction[i]    <- 1 - (post_sd_beta1^2 / prior_sd_beta1^2)
}
```

```
estPars <- data.frame(contraction, z_score)
ggplot(data=estPars, aes(x=contraction, y=z_score))+
  geom_hline(yintercept=0)+
  geom_point()+ xlim(c(0,1))+
  labs(x="Posterior Contraction",y="Posterior z-Score")
```

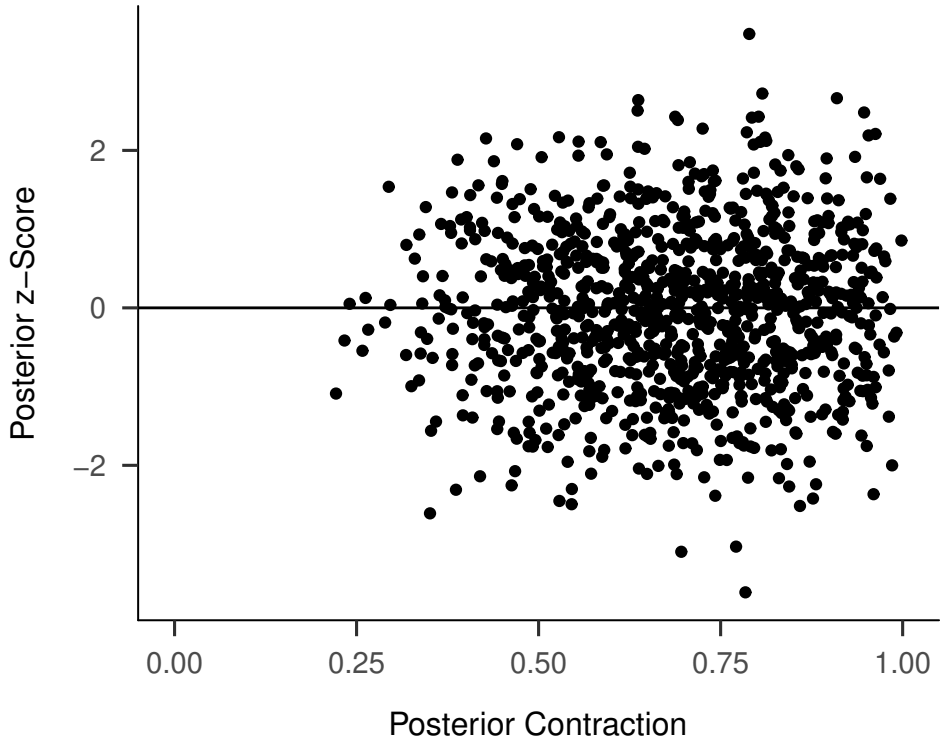

Figure 13. Analysis of model sensitivity. For each of N = 1,000 data-sets simulated from the prior parameters and the model, plot the posterior z-score as a function of the posterior contraction. The results show good posterior z-scores, as most of them are very close to zero, indicating little overfitting to wrong values or prior/likelihood conflict. The results for posterior contraction, however, are mixed. While some data-sets show very good (high) contraction close to one, others have only weak contraction of about 0.5, reflecting the relatively low number of participants and items in the experimental design.

Figure 13 shows posterior z-scores as a function of posterior contraction for all simulated data-sets. The results show that posterior z-scores are overall relatively close to zero, and mostly below absolute values of 2 (average: 0.82), reflecting good recovery of the ground truth. At the same time, posterior contraction ranges from only weak contraction (i.e., less than 0.5), where not a lot of information is gained about the parameter in the posterior relative to the prior, to very strong contraction (i.e., approaching 1), where posterior uncertainty is much reduced. The average contraction is 0.69.

The sometimes low posterior contraction indicates a tendency that for some simulated data-sets the slope parameter is not very well identified. This shortcoming may reflect the relatively small number of observed data points in the present study, reflecting low statistical power. Nevertheless, in a larger range of simulations model sensitivity looks reasonable.

This observation shows that posterior behavior will vary with the observed data, and that a model can be pathological for some data but not others. Since a priori we don't know what data we will ultimately observe, we want to check as many reasonable data-sets as possible before running the study, which is conveniently done in the prior predictive analyses.

How does model sensitivity look like in a situation with higher power? An exact replication study (Vasishth et al., 2013) is available for the present data-set by Gibson and Wu (2013). Here, we combine the data from both studies to see how the associated increase in statistical power affects model sensitivity. The results show that posterior z-scores are relatively unchanged compared to the previous anaylsis, and mainly are between absolute values of 2 (average: 0.83). However, for the posterior contraction, the samples now cluster slightly more to the right of Figure 14a and b: mean contraction is now 0.75 compared to the previous 0.69. This indicates somewhat stronger posterior contraction as a result of the higher number of subjects. On average, the data thus now provide more information on the parameter, and lead to a stronger reduction of uncertainty about the parameter of interest. At the same time, however, the amount of contraction strongly varies across different

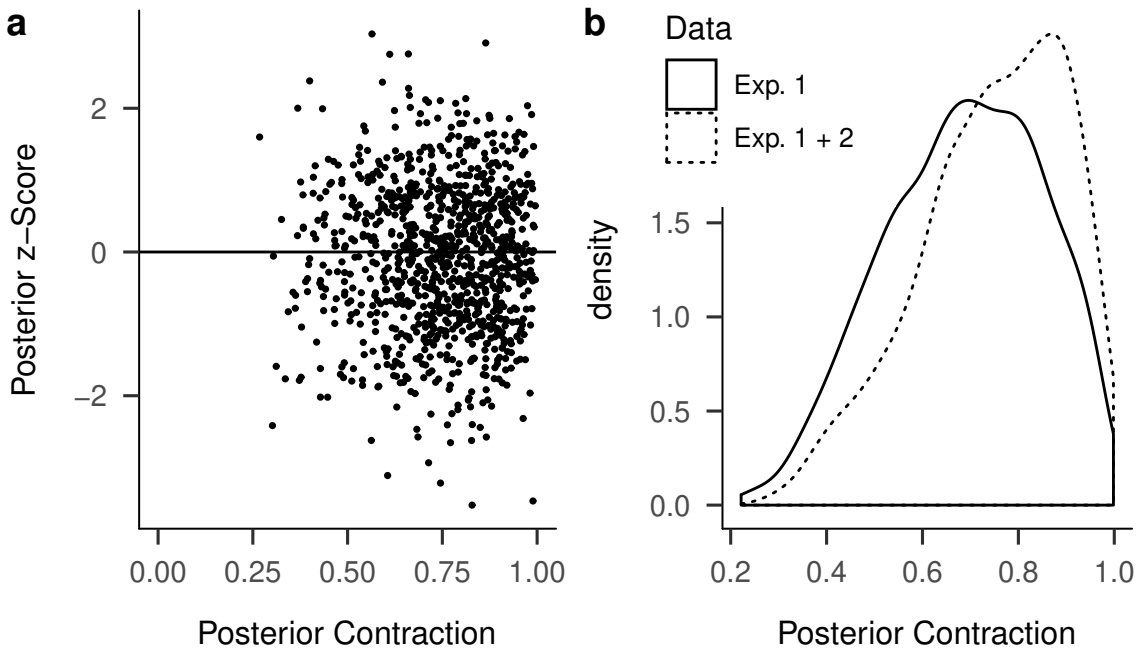

Figure 14. Analysis of model sensitivity for more observations, achieved by pooling across two data-sets. a) Results for pooled data-set (Exp 1: Gibson and Wu, 2012; Exp 2: Vasishth et al., 2013). b) Results only for posterior contraction for Experiment 1 (Gibson and Wu, 2012) and for the pooled data-set (Exp. 1 + 2). a)+b) The results show that - as the data now provides more information via the likelihood - this increased posterior contraction leads to more concentrated and hence more informative posteriors. a) At the same time, posterior z-scores are still rather close to zero, indicating good sensitivity.

simulated data-sets. Even with the larger number of subjects, contraction can be quite low (i.e., values around 0.5) for some simulated data-sets, whereas it is high for others. This means that even with double the number of subjects, we have no guarantee of getting informative results from our experiment.

Posterior predictive checks: Model adequacy

Having examined the prior predictive data in detail, we can now take the real, observed data and perform posterior inference on it. We start by fitting a maximal *brm* model to the

observed data.

```
m_gw <- brm(rt ~ so + (1+so|subj) + (1+so|item), gw1,
            family=lognormal(), prior=priors2, cores=4)
```

```
round(fixef(m_gw),3)
```

```
##           Estimate Est.Error   Q2.5 Q97.5
## Intercept    6.056     0.062  5.938 6.179
## so          -0.029     0.023 -0.074 0.017
```

Figure 15 shows the posterior distribution for the slope parameter, which estimates the difference in reading times between object minus subject relative sentences.

```
postgw <- posterior_samples(m_gw)
ggplot(data=postgw)+stat_bin(aes(x=b_so,y=..density..))+
  labs(x="Object - subject relatives", title="Posterior distribution")
mean(postgw$b_so<0)
```

```
## [1] 0.89575
```

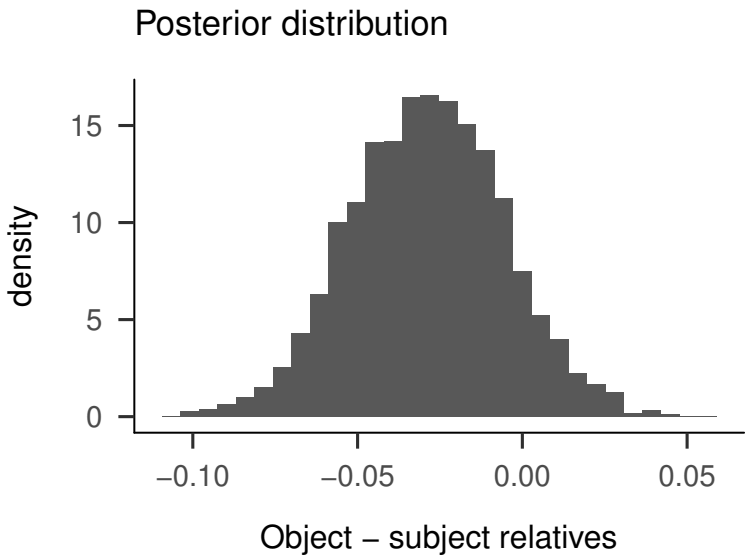

Figure 15. Posterior distribution for the slope parameter, estimating the difference in reading times between object relative minus subject relative sentences.

Figure 15 shows that the reading times in object relative sentences tends to be slightly faster than in subject relative sentences (p(b<0) = 0.90); this is as predicted by Gibson and Wu (2013). However, the 95% confidence intervals overlap with zero; it is difficult to rule out the possibility that there is effectively no difference in reading time between the two conditions.

To assess model adequacy, we perform posterior predictive checks. We simulate data based on posterior samples of parameters. This then allows us to investigate the simulated data by computing the summary statistics that we used in the prior predictive checks, and by comparing model predictions with the observed data.

```
nsamp     <- nrow(postgw)
rtpostmat <- matrix(NA,nrow(expdesign),nsamp)
for (i in 1:nsamp)
  rtpostmat[,i] <- genfake(expdesign, Nsj, Nit,
          beta0    = postgw$b_Intercept[i],
```

```
          beta1    = postgw$b_so[i],
          sigma_u0 = postgw$sd_subj__Intercept[i],
          sigma_u1 = postgw$sd_subj__so[i],
          sigma_w0 = postgw$sd_item__Intercept[i],
          sigma_w1 = postgw$sd_item__so[i],
          rho_u    = postgw$cor_subj__Intercept__so[i],
          rho_w    = postgw$cor_item__Intercept__so[i],
          sigma    = postgw$sigma[i])
```

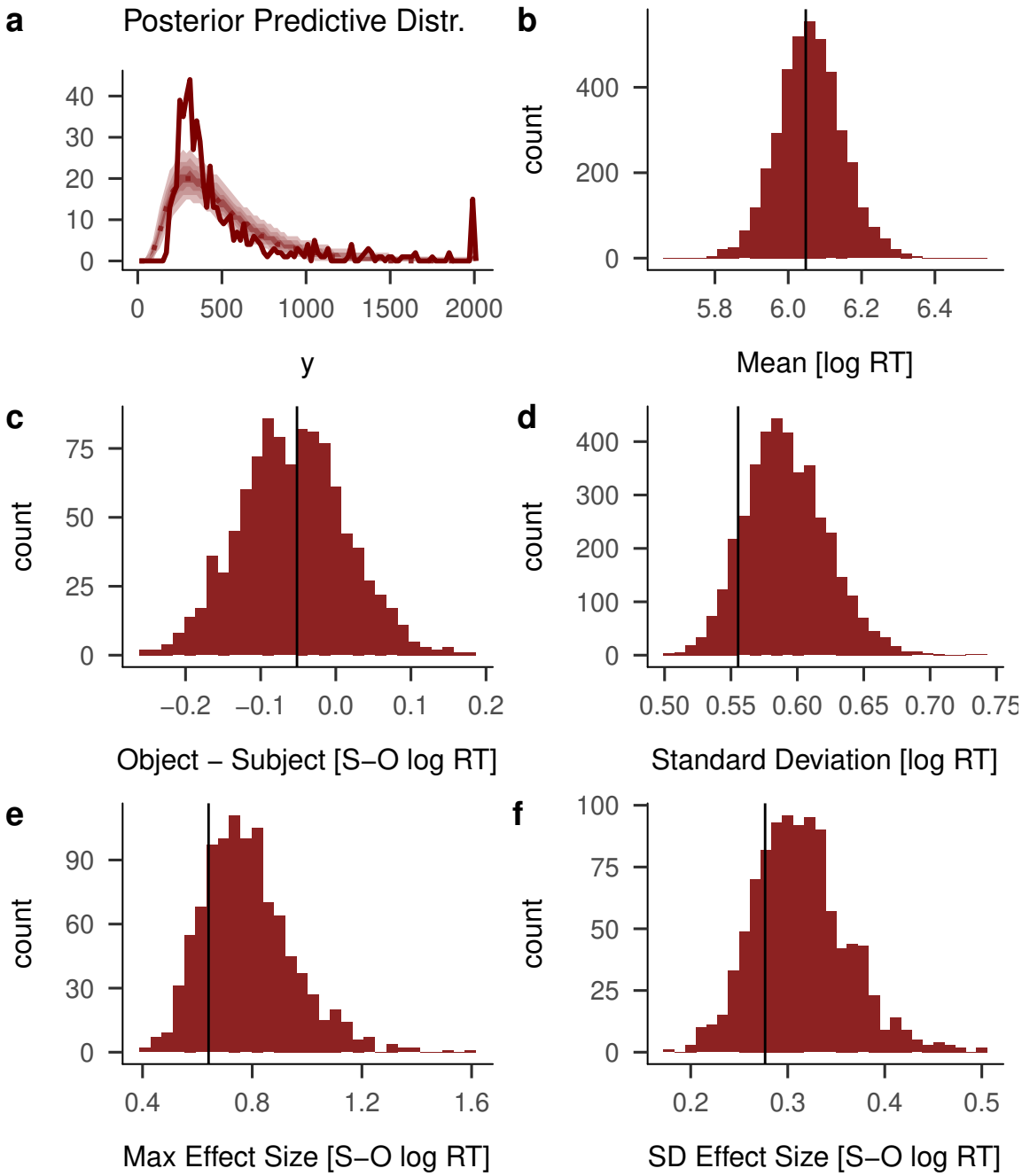

Figure 16. Posterior predictive checks for weakly informative priors. Distributions are over posterior predictive simulated data. a) Histograms of reading times. 10-90 percent, 20-80 percent, 30-70 percent, and 40-60 percent quantiles across histograms are shown as shaded areas; the median is shown as a dotted line and the observed data as a solid line. For illustration, values > 2000 are plotted as 2000; modeling was done on the original data. b) Average log reading times. c) Differences in log reading times between object minus subject relatives. d) Standard deviations of log reading times. e) Maximal effect size (object - subject relatives) across subjects. f) Standard deviation of effect size (object - subject relatives) across subjects.

The results from these analyses show that the lognormal distribution (see Fig. 16a) provides an approximation to the distribution of the data. However, although the fit looks reasonable, there is still systematic deviation from the data of the model's predictions. This deviation suggests that maybe a constant offset is needed in addition to the lognormal distribution. This can be implemented in *brm* by replacing the family specification `family=lognormal()` with the shifted version `family=shifted_lognormal()`, and motivates another round of model validation.

Next, for the other summary statistics, we transform all of these to log space, by taking the log of reading times before computing the summary statistics. We do this as the model parameters are operating in log space (we use a lognormal distribution). Therefore, computing summary statistics on the ms scale will necessarily lead to deviations between the model and the observed data, which are solely due to the scale, and are expected even when the model fits the data in log space perfectly.

First, we look at the distribution of means. The posterior predictive means well capture the mean reading time in the observed data (i.e., the vertical line in Fig. 16b). The same is true for the standard deviation - the model captures the standard deviation of the data well (Fig. 16d). Figure 16c shows the effect size of object minus subject relative sentences predicted by model (histogram) and observed in the data (vertical line). Again, posterior model predictions for the effect are well in line with the empirical data. The same is true for the biggest effect among all subjects (Fig. 16e) and for the standard deviation of the effect across subjects (Fig. 16f).

We had mentioned above that some subjects were removed due to invalid data. Note that in the frameworks of lme4 and brms these missing subjects cannot be modelled without adding new functionality to these packages. This, however, is possible when using Stan directly.

Bayes factor analysis

The posterior predictive checks suggest that the maximal model captures the summary statistics of the data well. This makes us confident that we can rely on the model for interpreting the estimate for the effect size of subject minus object relatives ("so"). However, simply testing the effect size does not provide evidence on whether the effect of "so" exists, i.e., whether it is different from zero. To answer this question, we can compute Bayes factors, where we compare the maximal model to a reduced model, where the fixed effect of "so" is missing (which essentially sets its parameter to zero). In the Bayes factor analysis, very importantly, we are using the more informative priors that we have derived from the prior predictive checks.

Note that to compute a Bayes factor in brm, a very large number of posterior samples are needed in order to obtain stable estimates of the Bayes factor. We therefore re-fit the model twice with larger number of samples (`iter=10000`), once with the fixed effect "so" included, and once without the fixed effect for "so".

```
m_gw1 <- brm(rt ~ so + (1+so|subj) + (1+so|item), gw1,
             family=lognormal(), prior=priors2, cores=4,
             save_all_pars=TRUE, iter=10000, warmup=2000)
t0 <- proc.time()
m_gw0 <- brm(rt ~ 1 + (1+so|subj) + (1+so|item), gw1,
             family=lognormal(), prior=priors2[-2,], cores=4,
             save_all_pars=TRUE, iter=10000, warmup=2000)
BF_informative <- bayes_factor(m_gw1, m_gw0)
time.brm <- proc.time() - t0
```

```
BF_informative
```

```
## Estimated Bayes factor in favor of bridge1 over bridge2: 0.62816
```

The results show a Bayes factor of approximately 1. This indicates that there is similar support for the maximal model and the reduced model, and that the data do not allow us to prefer any of the two models over the other. Thus, it is not clear from this data-set whether there is a difference in reading times between Chinese subject and object relative clauses.

Last, we note that the choice of informative priors is crucial for the valid analysis of Bayes factors. If the prior assumes that extremely large values for the "so" effect are possible, then the Bayes factor assesses whether there is evidence for such extremely large effect sizes. Of course, this is very unlikely to be the case empirically. To the contrary, when using weakly informative priors (e.g., informed by the prior predictive checks outlined above), then the prior assumes small to medium effect sizes for the "so" effect, and the Bayes factor accordingly tests whether there is evidence for such small or medium effect sizes.

Indeed, when we re-compute the Bayes factor between the maximal model and the reduced model, but using the vague or diffuse priors discussed above, the Bayes factor is strongly reduced:

```
BF_diffusePrior
```

```
## Estimated Bayes factor in favor of bridge1 over bridge2: 0.03949
```

The meaning of the Bayes factor is determined by the order in which the models are entered into the calculation. A Bayes factor of roughly 0.04[1] indicates that the first of the two models, which is here the reduced model, is much more likely than the second model,

[1] The `bayes_factor()` command should be run several times to make sure that the number is stable with respect to the randomness inherent to bridge sampling estimators.

which here is the maximal model. Now, there is suddenly strong evidence for the null hypothesis! Note that the only thing that was changed was the prior. With the diffuse/vague priors employed in the second Bayes factor analysis, there is clear evidence that extremely large differences between subject and object relatives are not supported by the data.

These results on how Bayes factors are highly sensitive to the prior highlight the importance of using reasonable priors when estimating Bayesian models. Simply using diffuse/vague priors can strongly bias Bayes factors towards the reduced model. It is therefore crucial to use domain knowledge to encode reasonable expectations about the size of the expected effect(s) of interest into (weakly) informative priors. A good strategy is to display a range of Bayes factors using increasingly informative priors. For an example, see Nicenboim, Vasishth, and Rösler (2019).

An important open question we do not discuss in the present paper is that the Bayes factor itself will be quite variable under repeated sampling; the stability of the Bayes factor will be discussed in a future paper.

An additional benefit of incorporating more domain knowledge into the prior is that this speeds up posterior sampling. For example, consider the maximal model for the "so" effect: for this, we recorded the time it took (in seconds) to fit one model and to perform the Bayes factor analysis (using the command `proc.time()`), and we did so for the diffuse priors and for the weakly informative priors:

```
c(diffuse=time.brm_diffusePrior[3], weakly.informative=time.brm[3])
```

```
##             diffuse.elapsed weakly.informative.elapsed
##                     136.859                     87.566
```

Incorporating domain knowledge into the priors lead to a speed-up in fitting the brm model and running the Bayes factor analysis from 137 seconds for the diffuse/vague priors to

88 seconds for the weakly informative priors. With more complex models, larger data sets, or when investigating computational faithfulness or model sensitivity, these time differences can become substantial.

Summary

We have introduced key questions to ask about a model and the inference process as discussed by Betancourt (2018), and have applied this to a data-set from an experiment involving a typical repeated measures experimental design used in cognitive psychology and psycholinguistics. Prior predictive checks using analyses of simulated prior data suggest that, compared to previous applications in reading experiments (e.g., Nicenboim & Vasishth, 2018), far more informative priors can and should be used. We found that including such additional domain knowledge into the priors leads to more plausible expected data. Moreover, incorporating more informative priors should also speed up the HMC sampling process. These more informative priors, however, may not alter posterior inferences much for the present design. We also investigated computational faithfulness using simulation-based calibration (SBC) and found that prior model parameters were well recovered using posterior estimation, supporting the used estimation procedure (`brm()` function as a wrapper to Stan) for the current setup. Analysis of model sensitivity showed that the critical theoretical effect of a psycholinguistic manipulation was estimated without bias as posterior z-scores were centered around zero. Posterior contraction varied between medium and strong contraction, indicating somewhat weak power in a rather small sample size. In line with this limited model sensitivity, posterior inference on the experimental effect based on the observed data did not provide strong evidence for the experimental effect of interest, leaving uncertain whether it differs from zero. Posterior predictive checks showed strong support for our statistical model, as the model successfully recovered most of the tested summary statistics. The Bayes factor analysis showed strong evidence for no effect (with diffuse priors) and an

inconclusive result (with informative priors). Our overall conclusion would be that we did not learn much from the experiment. As an aside, note that the published result in Gibson and Wu (2013) showed a statistically significant effect (using repeated measures ANOVA) and concluded that the effect was present.

In summary, this analysis provides a fully worked example and tutorial for using the principled Bayesian workflow (Betancourt, 2018) in cognitive science experiments. The workflow reveals useful information about which (weakly informative) priors to use, and performs checks of the used inference procedures and the statistical model. The workflow provides a robust foundation for using a statistical model to answer scientific questions, and will be useful for researchers developing analysis plans as part of pre-registrations, registered reports, or simply as preparatory design analyses prior to conducting an experiment.

Acknowledgements

Thanks to Bruno Nicenboim, Titus von der Malsburg, and participants of a lab meeting of the Vasishth lab for discussion. This work was partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 317633480 – SFB 1287, Project Q (Principal Investigators Shravan Vasishth and Ralf Engbert).

References

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

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

Betancourt, M. (2018). Towards a principled Bayesian workflow. Retrieved from https://github.com/betanalpha/jupyter_case_studies/blob/master/principled_bayesian_workflow/principled_bayesian_workflow.ipynb

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

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1).

Cook, S. R., Gelman, A., & Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, *15*(3), 675–692.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2017). Visualization in Bayesian workflow. *arXiv Preprint arXiv:1709.01449*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian data analysis* (Third.). Boca Raton, FL: Chapman; Hall/CRC.

Gibson, E., & Wu, H.-H. I. (2013). Processing Chinese relative clauses in context. *Language and Cognitive Processes*, *28*(1-2), 125–155.

Good, I. J. (1950). *Probability and the weighing of evidence*. New York: Hafners.

Lambert, B. (2018). *A student's guide to Bayesian statistics*. Sage.

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, *100*(9), 1989–2001.

Lunn, D., Thomas, A., Best, N., & Spiegelhalter, D. J. (2000). WinBUGS-a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*(4), 325–337.

Nicenboim, B., & Vasishth, S. (2018). Models of retrieval in sentence comprehension: A computational evaluation using bayesian hierarchical modeling. *Journal of Memory and Language*, *99*, 1–34.

Nicenboim, B., Vasishth, S., & Rösler, F. (2019). Are words pre-activated probabilistically during sentence comprehension? Evidence from new data and a bayesian random-effects meta-analysis using publicly available data. *PsyArXiv*.

Paape, D., Nicenboim, B., & Vasishth, S. (2017). Does antecedent complexity affect ellipsis processing? An empirical investigation. *Glossa: A Journal of General Linguistics*, *2*(1).

Pinheiro, J. C., & Bates, D. M. (2000). Linear mixed-effects models: Basic concepts and examples. *Mixed-Effects Models in S and S-Plus*, 3–56.

Plummer, M. (2012). JAGS version 3.3.0 manual. *International Agency for Research on Cancer. Lyon, France*.

R Core Team. (2016). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Schad, D. J., Hohenstein, S., Vasishth, S., & Kliegl, R. (2018). How to capitalize on a priori contrasts in linear (mixed) models: A tutorial. *arXiv Preprint arXiv:1807.10451*.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. (2014). The Deviance Information Criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(3), 485–493.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration. *arXiv Preprint arXiv:1804.06788*.

Vasishth, S., Chen, Z., Li, Q., & Guo, G. (2013). Processing Chinese relative clauses: Evidence for the subject-relative advantage. *PLoS ONE*, *8*(10), 1–14.

Vasishth, S., Mertzen, D., Jäger, L. A., & Gelman, A. (2018). The statistical significance filter leads to overoptimistic expectations of replicability. *Journal of Memory and Language*, *103*, 151–175.

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.