arXiv:1904.00661v3 [cs.SE] 1 Jan 2020

Bayesian data analysis in empirical software engineering—The case of missing data

Richard Torkar, Robert Feldt, and Carlo A. Furia

Abstract Bayesian data analysis (BDA) is today used by a multitude of research disciplines. These disciplines use BDA as a way to embrace uncertainty by using multilevel models and making use of all available information at hand. In this chapter, we first introduce the reader to BDA and then provide an example from empirical software engineering, where we also deal with a common issue in our field, i.e., missing data. The example we make use of presents the steps done when conducting state of the art statistical analysis. First, we need to understand the problem we want to solve. Second, we conduct causal analysis. Third, we analyze non-identifiability. Fourth, we conduct missing data analysis. Finally, we do a sensitivity analysis of priors. All this before we design our statistical model. Once we have a model, we present several diagnostics one can use to conduct sanity checks. We hope that through these examples, the reader will see the advantages of using BDA. This way, we hope Bayesian statistics will become more prevalent in our field, thus partly avoiding the reproducibility crisis we have seen in other disciplines.

1 Introduction

Statistics, we argue, is one of the principal tools researchers in empirical software engineering have at their disposal to build an argument that guides them towards the ultimate objective, i.e., practical significance and (subsequent) impact of their

Richard Torkar and Robert Feldt
Chalmers and University of Gothenburg, SE-412 96 Gothenburg, Sweden e-mail: torkarr@chalmers.se,robert.feldt@chalmers.se

Carlo A. Furia
Università della Svizzera italiana, Via Giuseppe Buffi 13, I-6904 Lugano, Switzerland e-mail: furiac@usi.ch

findings[1]. Practical significance is, as we have seen (Torkar et al., 2017), not very often explicitly discussed in software engineering research today and we argue that this is mainly out of two reasons.

The first one being that statistical maturity of empirical software engineering research is not high enough (Torkar et al., 2017), leading to difficulties with connecting statistical findings to practical significance. The second reason is a combination of issues hampering our research field, e.g., small sample sizes, failure to analyze disparate types of data in a unified framework or lack of data availability (only 13% of publications provide a replication package and carefully describe each step to make reproduction feasible (Rodríguez-Pérez et al., 2018)).

Both of the above issues are worrisome since it could make it hard to strengthen arguments concerning practical significance, e.g., connecting effort and, conclusively, ROI[2] to the findings of a research study, if one would want so. For academic research to be more relevant and have more impact on practitioners, its practical significance and its implications need to be precise.

Furthermore, issues such as the above is also likely to lead empirical software engineering towards a replication crisis as we have seen in other disciplines, e.g., medicine (Ioannidis, 2005b,a, 2016; Glick, 1992), psychology (Aarts et al., 2015; John et al., 2012; Shanks et al., 2013), economics (Ioannidis et al., 2017; Camerer et al., 2016), and marketing (Hunter, 2001).

In order to solve some of the above challenges researchers have proposed that we need to focus on, e.g., (i) openness, i.e., that data and manuscripts are accessible for all stakeholders, (ii) preregistration, i.e., a planned study is peer-reviewed in the usual manner and accepted by a journal *before* the experiment is run, so that there is no incentive to look for significance after-the-fact (Dutilh et al., 2017), (iii) increasing the sample size, (iv) lowering the significance threshold from $p < 0.05$ to $p < 0.005$ (Benjamin et al., 2018), and (v) removing null hypothesis significance testing (NHST) altogether, which the journal *Basic and Applied Social Psychology* advocates (Trafimow and Marks, 2015), as do McShane et al. (2017).

However, some researchers, most notably Gelman (2018), claim that even the above is not enough and argue that a unified approach for these matters should mainly evolve from three components: Procedural solutions, solutions based on design and data collection, and improved statistical analysis.

Concerning procedural solutions, Gelman (2018) like others, suggests publishing papers on, e.g., Arxiv, to encourage post-publication review, and to use preregistration as a tool for lowering the 'file drawer' bias. For design and data collection, Gelman provides convincing arguments that we should focus on reducing measurement error (the example being that reducing the measurement error by a factor of two is like multiplying the sample size by a factor of four), and move to within-subject

[1] In this chapter, we focus on empirical software engineering research where quantitative data is a major component; for studies that are mainly qualitative a different set of concerns need to be taken into account, see for example (Lenberg et al., 2017)

[2] In literature, Return-On-Investment refers to, in various ways, the calculation one does to see the benefit (*return*) an investment (*cost*) has.

from between-subject study designs when possible.[3] Finally, concerning improved statistical analysis, Gelman advocates the use of Bayesian inference and multilevel models (MLMs),[4] as a way to discuss ". . . the range of applicability of a study", i.e., practical significance.

Overall, we side with these arguments and believe they are critical also for software engineering to better connect empirical research with the practice it ultimately aims to improve. We will thus introduce and exemplify the use of Bayesian statistical methods in empirical software engineering research. They are a good starting point since individual researchers can learn them and apply them in isolation without waiting for the community as a whole to take further steps needed to avoid a replication crisis and to become more practically relevant. We argue that using Bayesian methods allows us to better connect our findings to practical significance through the use of more balanced out-of-sample predictions, i.e., one of the outputs from Bayesian data analysis (this will be further elaborated on in Sect. 2). Additionally, we have yet to face data from empirical software engineering where Bayesian data analysis can not be employed, and when having a small sample size, due to the priors employed, Bayesian data analysis, we would argue, shows its strengths.

In this chapter, we rely on three key concepts: Bayes' theorem, multilevel models, and Markov chain Monte Carlo sampling.

Bayes' theorem states that,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

where A and B are events, and $P(B) \neq 0$. In the theorem, we have two conditional probabilities, $P(A|B)$ and $P(B|A)$, the likelihood of event A occurring given that B is true, and vice versa. The marginal probability is then observing A and B independently of each other, i.e., $P(A)$ and $P(B)$. Often the above is rewritten as, $P(A|B) \propto P(B|A) \times P(A)$, i.e., the posterior is proportional to the likelihood times the prior or, in other words, given a likelihood and a prior we will be able to approximate the posterior probability distribution; this is, of course, also applicable to MLMs. We will come back to these concepts in the next section.

Multilevel models are not a particularly new thing. However, in the last decades, they have become accessible to researchers due to the rise in computational power, and they go nicely in hand with Bayesian analysis. Bayesian MLMs have several advantages (McElreath, 2015): (i) When using repeated sampling they do not underfit or overfit the data to the extent single-level models do (i.e., maximally), (ii) the uncertainty across uneven sample sizes is handled automatically, (iii) they model variation explicitly (between and within clusters of data), and (iv) they preserve uncertainty and makes much data transformation unnecessary. In our particular case, Bayes' theorem is the foundation for conducting inference when using MLMs, and Markov chain Monte Carlo (MCMC) is the engine that drives it.

[3] In a within-subject design the same group of subjects are used in more than one treatment.

[4] Multilevel models can also be called hierarchical linear models, nested data models, mixed models, random coefficient, random-effects models, random parameter models, or split-plot designs.

The reason for using MCMC for sampling is simply that before MCMC was introduced, it was virtually impossible to sample large Bayesian multilevel models (Banerjee et al., 2014). Today, if one wants to sample from a complex, multidimensional, unknown, posterior probability distribution, MCMC is a widespread technique to use since we have the computational power available. (For more background on sampling algorithms please see (McElreath, 2015, Ch. 8).)

Next, we first introduce the main elements of Bayesian Data Analysis (BDA) with a non-Software Engineering example. Our main contribution is then a detailed worked case study of applying BDA to an estimation problem in empirical software engineering. In particular, the example highlights that with this BDA analysis, we do not need to delete data points for which some data is missing. We conclude the chapter by discussing the methodological implications.

Bayesian data analysis (BDA) is growing and, as such, is being used in many disparate scientific disciplines. The approach of BDA that we use in this chapter relies on designing a generative model which we then can use to do out of sample predictions. It will be a more involved analysis, but in the end we hope that it will also provide us with a richer understanding of the phenomena under study.

2 A Short Introduction to Bayesian Data Analysis

Lately, many tools and probabilistic programming languages have been developed to tackle some of the challenges we face when designing more powerful statistical models. In our view, several things have improved. First, probabilistic programming languages, e.g., using `Julia` with `Turing.jl`, or `Stan`, tailored for statistical programming, in combination with resampling techniques, have matured.[5] Second, resampling techniques based on MCMC have improved (Brooks et al., 2011). Third, procedures for using these techniques now exist (Talts et al., 2018; Betancourt, 2018; Gabry et al., 2017; Gelman et al., 2017; Betancourt, 2017) and are being improved iteratively (Vehtari et al., 2019). Together, these development make more powerful analysis methods available to a wider audience.

In this section, we will provide a short introduction to model design, its tool support, and some terminology that we will use in this chapter. To keep it simple and general, we will take data and an example from everyday life, rather than an empirical software engineering example. We do not expect the reader to be an expert after this, but rather be able to follow what we present in this chapter, be better prepared for the empirical software engineering case study that then follows, and then perhaps read further into the literature we present in Sect. 4. Let us start with terminology.

[5] See `https://julialang.org`, `http://turing.ml`, and `https://mc-stan.org`.

In this chapter, we will design statistical models. We will use mathematical notation for precision as well as brevity. To generalize, a model will consist of a likelihood, a linear equation, and priors. The purpose of the model is ultimately to make predictions/inferences concerning the outcome by using a posterior predictive distribution. Let us introduce a simple example inspired by McElreath (2015).

We want to predict the height of human beings given their weight. A model could then look like this,

$$\begin{aligned}
\text{height}_i &\sim \mathcal{N}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_w \times \text{weight}_i \\
\alpha &\sim \mathcal{N}(181, 20) \\
\beta_w &\sim \mathcal{N}(0, 10) \\
\sigma &\sim \text{Half-Cauchy}(0, 10)
\end{aligned}$$

Let us now go through this line by line. First, we claim that height has a Normal distribution with mean μ and standard deviation σ, i.e., our likelihood. The subset i in height, weight, and μ is an indication that this holds for each height we have in the data set, i.e., for every human being in the dataset. But why a Normal distribution? Well, there are ontological and epistemological reasons for this (McElreath, 2015), but in short: if we add together random values from the same distribution it converges to a normal distribution. Since there are many different factors that, jointly, determines the height of a person, e.g., their genetics, nutrients of the mother during pregnancy, food intake as a small child, etc., and their effects 'add up', it is often a sensible assumption to assume the result will be normally distributed.

The next line encodes our main assumption about the heights, i.e., they have a linear connection to the weight (our linear equation). We have an intercept labeled α, expressing the average height of a human that has average weight, together with a slope β_w, which captures how much longer (shorter) a human can be expected to be for each added (subtracted) unit of weight they have. We want to estimate these two *parameters* using the data: height and weight. In this example, height is the *outcome* and weight is the *predictor*. We can have more than one outcome, this is known as a multivariate model (compared to univariate models as in the example above), and we can have more than one predictor, as we will see later in this chapter.

Next, in a Bayesian model, we need to express our prior belief, our so-called 'priors'. The α parameter is the intercept, and hence captures the mean height we expect (see Fig. 1 for a graphical presentation). What we are saying is that we have prior knowledge, i.e., we believe that the mean height will be 181 cm. Why 181? Well, this is the average height of the three authors of this chapter, and when writing the chapter, we had direct and reliable access to this data. Also, our prior expresses that we can expect the mean to vary with a standard deviation of 20, i.e., finding humans with a height in the range 161 to 201 cm would not be too uncommon even if values outside that range can also sometimes happen. For β_w, our prior indicates that the slope has a mean of 0 and a standard deviation of 10 (Fig. 1). We could also

set a more specific prior here, e.g., we have a feeling that an increase in weight also leads to an increase in height, but let's use a very wide and 'allowing' prior.

Finally, we have a prior on σ, which is the expected variation in actual, measured data from what our linear, 'core' model predicts. We have chosen a Half-Cauchy distribution here since we know that the variation cannot be negative (Fig. 1). The Half-Cauchy is a common prior for σ and is roughly a Normal distribution cut in half, i.e., we do not allow negative values on σ.[6] It also has a higher probability for larger values than a Normal distribution, since we have less information on how much variation to expect. In the end, if we have enough evidence (data), it will dominate, also known as 'swamp', the priors. This means that the priors are not as critical in situations when we collect lots of data. Before we go on and use our statistical model on actual, measured data, we should study how our model behaves based on only the priors. We can do this by sampling from our priors and 'executing' the model to see which heights it predicts.

In the lower half of Fig. 1, we see the joint prior probability distribution, which is a combination of the figures on the first row. The mixtures of the priors for α, β, and σ, and the linear regression they imply could be seen as representative for our height given a weight.

So what does our prior probability distribution (Fig. 1) tell us? Well, 2.93% of the population is assumed to be more than 272 cm tall.[7] Additionally, 13.8% are less than 147 cm.[8] This seems a bit strange in our view, and even more bizarre is that 0.015% of the population is shorter than 20 cm, when the shortest human recorded was approximately 53 cm.

But there is no need to worry. The main idea when selecting priors is to delimit the volume that the sampling needs to cover. We want to get rid of obviously absurd values while ensuring that we do not rule out values that could happen. Who knows, maybe someone who is <53 cm or >272 cm will be found this year. We have just conducted a *prior* predictive analysis, which is, we claim, a compulsory part of doing Bayesian data analysis.

For actually making inferences, i.e., determining the likely ranges of the parameters given our model, we will need data. We will make use of a data set found in the `rethinking` R package and a R Markdown script of our analysis can be downloaded.[9] After sampling, we will have a *posterior* distribution, which is proportional to the likelihood and the prior distribution. In Fig. 2, we have plotted the empirical data set (circles) and the linear prediction (straight line). The narrow shaded interval is the 95% distribution of μ (i.e., the exact values for many μ from the posterior), and the wider and lighter interval is the 95% plausible region (i.e., 95% of our μ should be found within that region).

---

[6] Other priors for σ can of course also be used. Please see `https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations`.

[7] The tallest man, for whom there is irrefutable evidence, was 272 cm.

[8] People of short stature are <147 cm

[9] `https://github.com/torkar/BDA_in_ESE`

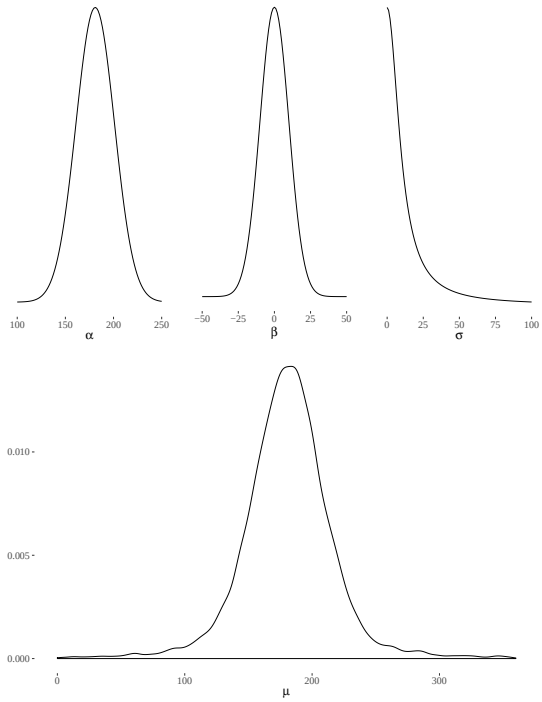


Fig. 1 Our selected prior probabilities (priors) for parameters α, β, and σ, respectively (top row). These are then combined into our prior predictive distribution for the height μ (bottom row)

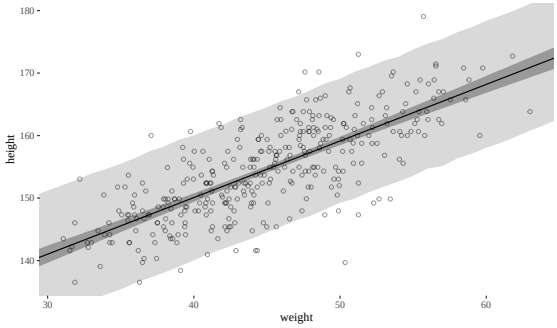


Fig. 2 Height as a function of weight. The line represents the μ, the dark shade the 95% distribution of μ, and the lighter shade the 95% plausible region

Having a posterior predictive distribution we can now start to conduct various inferences, but we will stop here for now and instead, in Sect. 3, present multiple ways we can make use of a posterior.

To summarize, the three main steps of Bayesian model design and analysis are:

1. Understand the data and the problem.
2. Design a probability model (conduct model checking and iterate if the model needs to be revised) and sample from the posterior to conduct diagnosis.

3. Conduct inference. That is, learn something about the population by using the posterior probability distribution, e.g., by conducting statistical tests or deriving estimates.

The above is an iterative process, and in the last step, we also can change the parameters to see how they affect the outcome variable, i.e., to analyze the practical implications of different scenarios and thus assessing the practical significance of the results. The above steps will next be covered in a detailed case study within empirical software engineering.

3 Case Study

Most would argue that to conduct estimations in software projects, one should not rely exclusively on expert opinion, but also on quantitative data collected in a more unbiased way. To this end, researchers have published studies making use of, among others, the International Software Benchmarking Standards Group's data repository (ISBSG).[10] (For an overview and introduction to the data sets please see (Hill et al., 2001).) Their large collection of data sets include cost, size, and defect data from projects and sprints, which can be used for research, estimation, and prediction for and in future projects.

While the ISBSG data sets are typically cleaned and anonymized version of data sets collected from industrial projects they, like similar data sets in other collections, still exhibit many of the same characteristics that we can expect from actual projects in industry. For example, they have missing data, disparate quality in data collection procedures, and a large variety of data types. These are data quality issues we see also in empirical software engineering research in general.

As we will later see in Sect. 4, the dominant strategy to handle missing data in empirical software engineering research is to merely remove cases that have missing data (listwise deletion). We believe that this strategy is sub-optimal and, generally speaking, not good for our research discipline. Even in cases when data can be classified according to the quality of the data collection procedure, as is the case with the ISBSG data sets, one sees that our community often chooses only to use a subset of data, classified to be of the highest quality (see, e.g., (Keung, 2008; Liebchen and Shepperd, 2008) for recommendations, and (Mittas et al., 2015) for an example where the authors use the recommendations). In short, we believe that data of low quality should be seen as better than no data at all, and the general rule of thumb should be never to throw away data. This is the context for our showcase and, in the following, we will both apply techniques for data imputation and conduct Bayesian data analysis on effort estimation data on the ISBSG data set collection.

[10] `http://isbsg.org`

[11] As we will see, missing data can be naturally handled in Bayesian analysis and, thus, showcases one of the unique and pertinent strengths in an empirical software engineering context.

3.1 The Data and the Problem

We will use the ISBSG Release 10 data set and set the dependent variable to `Effort`, i.e., the total number of person-hours to conduct a certain development task. According to, e.g., (Keung, 2008; Liebchen and Shepperd, 2008) and the International Software Benchmarking Standards Group (ISBSG), the following preprocessing steps are appropriate:

1. Only projects classified with data quality rating 'A' are kept, and 'B–D' are excluded.
2. Only projects using IFPUG (unadjusted functional size measurement) should be kept. However, the data description clearly states that versions ≥ 4.0 should not be compared to < 4.0. Hence, we only use versions ≥ 4.0.
3. According to Keung (2008), some additional variables should be kept for compatibility with previous studies.
4. Cases with missing values should be excluded.

The above leads to variables of interests as listed in Table 1, according to Mittas et al. (2015). If we use the variables in Table 1, and follow the advice above, we will later see that we need to remove 3,895 projects out of the total 4,106 (close to 95% of the projects). This seems wasteful but is the practice in our discipline, given the current standards and relevant recommendations.

Imagine instead that we aim to keep as much data as possible, i.e., a data-greedy approach. Well, first of all, we should consider including all projects no matter the quality rating. After all, we can easily classify them differently in a statistical model and even investigate the difference between projects depending on the data quality rating. Hence, we decide to include all projects and mark them according to their data quality rating, i.e., `DQR`, no matter if they have missingness in them.

The next, crucial step before stating our model, is to analyze causality among our variables (in Table 1). While this was not needed in the simple, height-of-humans example above, it is essential in more complex situations when multiple variables measure entities that might be causally related. Since this is almost always the case in real-world empirical software engineering research, it should be an essential step in building our statistical model. Otherwise, we risk that dependencies, e.g., correlations and collinearity, among variables might influence, and could potentially

[11] A reproducibility package, making use of brms (Bürkner, 2017) (with Stan (Carpenter et al., 2017)) written in R (R Core Team, 2018), can be downloaded: `https://github.com/torkar/BDA_in_ESE`. The raw data can, however, not be downloaded due to copyright reasons. Please see `README.txt` in the repository for more information and what you need to do to access the raw ISBSG data.

Table 1 Variables of interest according to previous studies. The variable names that are underlined have been removed in this study as explained in Sects. 3.1.1 and 3.1.2. The variable name in bold was added as explained in Sect. 3.1

Name	Description
AFP	Adjusted function points
Input	Number of inputs
Output	Number of outputs
Enquiry	Number of enquiries
File	Number of files
Interface	Number of interfaces
Added	Number of added features
Changed	Number of changed features
Deleted	Number of deleted features
Effort	Actual effort (person-hours)
DQR	Data quality rating

weaken our analysis. Below, we will then also directly analyze correlations between predictors, through an analysis of non-identifiability before we can then decide on our model we will also discuss how to handle the missing data and the sensitivity of our priors.

3.1.1 Causal Analysis

All predictors except DQR, seem to be raw and unadjusted measurements that are later used to calculate the adjusted function point (AFP).

Drawing our causal model (Fig. 3) as a directed acyclic graph (DAG) shows something generally considered to be a pipe confounder (one of four types of relations in causal DAGs) (Pearl, 2009).

In short, AFP mediates association between the other predictors and our outcome Effort, i.e., Effort $\perp\!\!\!\perp$ Input|AFP, or to put it differently, Effort is independent of Input, when conditioning on AFP.

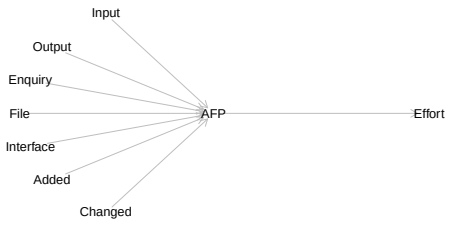


Fig. 3 A directed acyclic graph of our scientific model

We often worry about not having a predictor that we need for making good predictions (omitted variable bias). However, we do not often consider mistaken inferences because we rely on variables that are derived from other variables, i.e., post-treatment bias (McElreath, 2015). (Rosenbaum (1984) calls this the concomitant variable bias.)

In experimental studies, one would declare `AFP` to be a post-treatment variable, but the same nomenclature can be used in observational studies. To summarize, the above indicates that `AFP` should not be a predictor since it is derived from other, more basic, variables, so we will leave it out for now.

Note that causal analysis is not absolute in the sense that there is only one possible causal model that could be posited. Much scientific debate might be needed to argue for one or the other specific model and, thus, could lead to the exclusion of different sets of variables by different authors. As a consequence, this might lead different researchers to include or exclude different sets of variables and, thus, obtain different statistical results from separate analyses of the same data. However, in theory, this uncertainty and apparent subjectivity are still present with a traditional approach to statistical analysis, albeit hidden under the simplicity and familiarity of just applying a known statistical test.

3.1.2 Identifying Non-Identifiability

Strong correlations between variables are generally speaking a challenge when building statistical models. The model will make good predictions, but it will be harder to analyze and understand since the impression will be that variables that have a strong correlation, do not seem to have predictive power, while in fact, they have strong associations with the outcome (McElreath, 2015, Sect. 5). As a concrete example, if you want to build a model of a person's length, using the length of both her legs as separate predictors will not help the matter, i.e., adding another leg will not add predictive power to the model since it will correlate very strongly with the length of the first included leg. This type of multi-collinearity we generally want to avoid in statistical models.

Traditionally, there are two ways one can investigate this: Examining a pairs plot where all combinations of parameters and their correlations are visualized, or check if the matrix of predictor values is a full rank matrix, and thus identify non-identifiability that way.[12]

The latter, matrix-based approach consists of declaring a model $y = \beta_1 x_1 + \ldots + \beta_n x_n$, using the data, i.e., the values of the predictor variables, $x_i, \ldots, x_n$ as a matrix $\mathbb{A}$, and decompose it into a product $\mathbb{A} = \mathbb{Q}\mathbb{R}$ of an orthogonal matrix $\mathbb{Q}$, and an upper triangular matrix $\mathbb{R}$. By analyzing the diagonal of the matrix $\mathbb{R}$ a threshold value of $|d_{ij}| < 0.1$ along the diagonal is then often used to declare a variable as unsuited for inclusion in a model. Often something like $1e^{-12}$—quasi-zero for a computer—could be used, but generally speaking anything below 1.0 has traditionally been excluded. The argument is that if it is below 1.0, then the variable would provide very little additional value to the model and should thus not be included.

Identifying non-identifiability for our data set from Table 1 clearly indicates that `Deleted` should be a candidate for removal ($|d_{\text{Deleted}}| = 9e^{-12}$). It might feel strange

[12] In short, different values of the parameters must generate different probability distributions of the observable variables. Otherwise we face various degrees of non-identifiability, i.e., essentially that (too) many parameter combinations could lead to the same observations.

to remove predictors when our ultimate goal is to use as much data as possible. However, to build a statistical model that is sane, has good out of sample prediction, and is understandable, a trade-off is needed. Here we argue that the bulk of work should be done before we design our statistical model, to make use of the missing data techniques available to us.

3.1.3 Missing Data Analysis

When data is missing from cases of our data sets the most common solutions is to either delete such cases or impute, i.e., 'guess', based on the values we do have. Rubin (1987) has shown that very often 3–5 imputations are enough (the complete dataset is imputed fully 3–5 times) and that the relative efficiency of an estimate based on m imputations is approximately:

$$\text{Relative Efficiency} \approx (1 + \frac{\gamma}{m})^{-1}$$

where γ is the fraction of missing information. The relative efficiency in this case refers to using the finite m imputation estimator instead of the infinite number for the fully efficient imputation.

As an example, consider that we have 20% missing information in a variable ($\gamma = 0.2$), given $m = 5$, we have reached a relative efficiency of approximately 96%. Setting $m = 10$ we reach 98%. By doubling the computational effort, we have only a slight gain in relative efficiency. On the other hand, we have lots of computing power at our hands nowadays. However, more recently, we have seen that other recommendations for handling missing data have been presented.

Bodner (2008) and White et al. (2011) showed through simulations and by analytically deriving Monte Carlo errors, respectively, that the general rule of thumb should be $m = \gamma \times 100$, i.e., if a variable has 40% missing data ($\gamma = 0.4$) we should set $m = 40$ (using Rubin's efficiency estimate this would mean 92.6%→99.0%).

As will be evident, we will take the more conservative approach (i.e., $m = \gamma \times 100$), when we present the model implementation in Sect. 3.2.1.

Before we continue with the next section, it might be worthwhile to note that missingness in a Bayesian framework can be done in different ways. Either we conduct multiple imputations to derive uncertainty for all parameters, including our missing data. This is the path we have chosen here. The other approach would be to design a model of all data, including the missingness mechanism.

To model the missingness mechanism can be more involved and requires us to be very explicit about how our missingness occurred. At the best of times, this is a challenging task. We could instead argue that using the first approach, as we do here, shows the strengths of the Bayesian approach, since it easily can make use of various techniques, to handle missing data, in a coherent and principled way.

3.1.4 Sensitivity Analysis of Priors

Our analysis so far, to summarize, indicates that we should use seven predictors and one group variable (quality level of the project, in ISBSG terms), to predict one outcome variable (Effort). We can already now assume that we will most likely use a likelihood (i.e., our assumptions regarding the data generative process) that is based on counts (`Effort` is after all a count, the number of hours, going from zero to infinity).

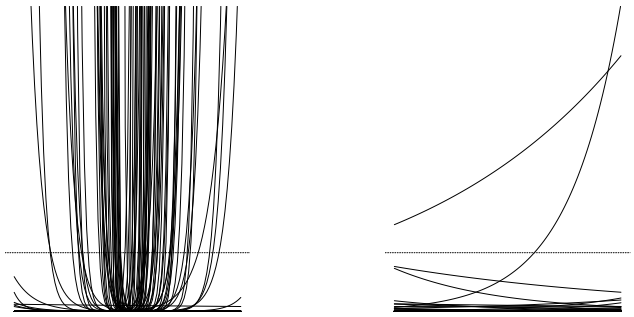
Thus, we plan to use a generalized linear model, with a link function that translates between the linear predictor value and the mean of the distribution function. In the case of count distributions, such as Poisson, it is customary to use a log link function, i.e., a parameter's value is the exponentiation of the linear model. However, when setting priors for parameters and using a link function, unexpected things can happen, and the priors might not have the effect one would expect. To this end, we should always do prior predictive simulations, i.e., a sensitivity analysis of how different settings of the priors affect the predicted variable.

The description of the data set indicates that approximately 20% of the projects have more than 20 people in the team. If we assume, roughly 1,700 h/year for an individual, having 60 people in a team sums up to approximately 100,000 person-hours per year. Let us now assume that this is the maximum value for our outcome variable `Effort`. Random sampling from $e^{\mathcal{N}(5,4)}$ provides us with $\bar{x} \approx 208,000$ (we use the exponential since we assume a log link function) indicating that this could be an acceptable prior for the intercept α.

We arrived at the values 5 and 4 above by starting from typical default values such as assuming α to be $\mathcal{N}(0, 10)$ (not uncommon as a default choice in, e.g., Poisson regressions), this would lead to $\bar{x} = 8 \times 10^{11}$ hours of work effort for an average project. This would correspond to close to half a billion people working on the project for one year. Thus we should use non-default priors to adapt the priors better so that they do not (often) give absurd values.

To assess the impact of very broad priors like $\mathcal{N}(0, 10)$ for our seven parameters, we thus, iteratively, compared their usage to that of other, narrower priors like $\mathcal{N}(0, 0.25)$. Furthermore, assuming a log link function, the additive effects of the seven priors for our β parameters would, on a normalized scale, correspond to $\mathcal{N}(0, (10 \times 7)^2)$ and $\mathcal{N}(0, (0.25 \times 7)^2)$, respectively. As is evident from Fig. 4 (a), we have a massive emphasis on extremely high y-values (we would require the world's total population to work in a project for this to happen). Now compare (a) with (b). We still allow extremely large y-values (up to 640×10^6!), but the emphasis is now a bit more realistic.

To summarize, prior predictive simulations indicate that setting $\mathcal{N}(5, 4)$ and $\mathcal{N}(0, 0.25)$ on α and β, respectively, allows us to delimit the multidimensional Gaussian space of possible parameter values, while still not remove the probability of extreme values altogether. If we would still be uncertain, one could have conducted even more prior predictive simulations (Simpson et al., 2014). The prior knowledge we took with us when doing this analysis, was that it was not likely that very many projects had billions of people involved. One could have taken a more conservative



(a) β priors $\mathcal{N}(0, 10)$ (b) β priors $\mathcal{N}(0, 0.25)$

Fig. 4 Prior predictive simulation of broad and informative priors, respectively. The x-axes are z-scores, while the y-axes represents our outcome variable `Effort`. The dashed horizontal line corresponds to our assumed maximum value for our outcome variable, and is the only value of interest in this case. We have plotted 100 simulations with the intercept $\mathcal{N}(5, 4)$ and our seven priors for the respective β parameters

approach and claim that it is not likely that we have millions of people in our projects; however, as we will see, Hamiltonian Monte Carlo will handle these priors well given the available data.

One should always conduct prior sensitivity analysis (prior predictive simulation) before making use of the available data. There is always *some* prior knowledge one can use!

3.2 Design of Model and Diagnosis

Based on our initial analysis, we have a much clearer picture of which variables to include and the overall sensitivity of our priors. For our model one could imagine using a Poisson likelihood, that is, we have a count ($\texttt{Effort}_{0\to\infty}$), which we can model binomial events for when the trials N are very large and the probability p small. However, that would be a mistake.

In any analysis, it is important first to get to know the actual data. Thus, let us look at some descriptive statistics of the data (taking into account the pre-processing steps previously introduced).[13] Some issues catch the eye in Table 2. First, `Effort` has a max value of 645,694 (three times larger than the mean for our priors). Second, the medians are consistently lower than the means (in one case the median is zero) indicating positive skewness. Third, not visible in the table, `Effort`, compared to

[13] For all data sets we use single quotes to emphasize the names, e.g., 'A_clean', while we print out variable names in `verbatim`.

the predictors, contain no zeros (indicating that we do not need to consider zero-inflated or hurdle models (Hu et al., 2011)). Finally, the mean and the variance for our outcome variable are very different (the variance is approximately 70,000 times larger than the mean).

Table 2 Descriptive statistics of our predictors and the outcome variable `Effort` using all data available to us (i.e., 4,106 projects). after conducting the pre-processing steps in Sect. 3.1. From left to right: Name, mean, median, max, min, and sample variance (with removed NAs). All numbers rounded to the nearest integer

Variable	$\bar{x}$	$\tilde{x}$	$\max(x)$	$\min(x)$	s^2
`Input`	143	56	9404	0	144 167
`Output`	125	47	3653	0	70 522
`Enquiry`	74	27	2886	0	22 318
`File`	118	43	10 821	0	137 172
`Interface`	39	10	1572	0	11 082
`Added`	357	142	15 121	0	591 300
`Changed`	128	0	18 357	0	344 677
`Effort`	5384	1828	645 694	4	391 631 309

Concerning the latter issue, a Poisson likelihood assumes the mean and the variance be approximately equal. This allows us to use the negative binomial, known as the Gamma–Poisson (mixture) distribution, as our likelihood, i.e., a continuous mixture model where we assume each Poisson count observation has its own rate. However, since we are still using a Poisson model, in essence, one could claim that we do not redo the sensitivity analysis.

To summarize our findings so far we can now formulate our model:

$$
\begin{aligned}
\text{Effort}_i &\sim \text{Gamma-Poisson}(\lambda_i, \phi_i) \\
\log(\lambda_i) &\sim \alpha + \beta_{\text{Input}} \times \text{Input} + \beta_{\text{Output}} \times \text{Output} + \beta_{\text{Enquiry}} \times \text{Enquiry} \\
&\quad + \beta_{\text{File}} \times \text{File} + \beta_{\text{Interface}} \times \text{Interface} + \beta_{\text{Added}} \times \text{Added} \\
&\quad + \beta_{\text{Changed}} \times \text{Changed} + \alpha_{\text{DQR}[i]} \\
\alpha &\sim \mathcal{N}(5, 4) \\
\beta_{1,\ldots,7} &\sim \mathcal{N}(0, 0.25) \\
\alpha_{\text{DQR}} &\sim \mathcal{N}(0, \sigma) \\
\sigma &\sim \text{HalfCauchy}(0, 1) \\
\log(\phi_i) &\sim \text{Gamma}(0.5, 0.5)
\end{aligned}
$$

We model each observation from a negative-binomial (Gamma-Poisson) distribution, with a failure rate λ and shape ϕ. We then use a log link for our linear model λ where we include an intercept α and parameters (β) for all predictors.

We also add varying intercepts in the form of our `DQR` variable. The idea is that each data quality rating should be treated uniquely by allowing us to estimate α for each rating, i.e., each `DQR` will have its own intercept. This will enable us to see

if there is an overall difference between projects judged to have different quality ratings.

Finally, we set the aforementioned priors on our β parameters (but we use $\mathcal{N}(0, \sigma)$ for our unique intercepts, to separately estimate σ for each level of `DQR`). We also set HalfCauchy$(0, 1)$ and Gamma$(0.5, 0.5)$ for σ and ϕ, respectively. Both of these priors are regularizing priors common for these types of parameters.[14]

3.2.1 Using the Model

In the previous sections, we presented our statistical model with assumptions. In this section, we will make use of it in two ways: sampling with complete data and imputed data. However, before we begin, Table 3 describes the data sets we will use.

Table 3 Data sets used. The 'A*' and 'AD*' categories are of different dimensions due to our index variable, `DQR`, added to the 'AD*' sets. From left to right. Name of data set, number of projects (rows), number of NAs, percentage of NAs, and number of zeros

Name	# projects	# NAs	% NAs	# zeros
'A'	501	2109	23.8	316
'A_clean'	214	0	0	316
'AD'	1689	8507	19.8	736
'AD_clean'	494	0	0	727

The data sets are divided into two categories. First, we have data sets that only take into account projects classified as having the highest quality rating ('A') and data sets where we use all four quality ratings ('AD'); taking into account the pre-processing steps in Sect. 3.1.

First, we have subsets with NAs ('A' and 'AD') and, second, subsets where all original NAs are removed ('*_clean'). The logic to use these data sets is that we want to use as much data as possible (but we pay the price of missing data), and removing all NAs is, as we have discussed, not uncommon.

One could also imagine having subsets where all zeros are assumed to be NAs, but that would be a bit too conservative assumption in our opinion, and we leave it to the reader to try out such a scenario.

If one would like to compare our data sets with what is commonly seen in literature then, taking into account that we have a more restrictive view on which IFPUG versions are included, 'A_clean' would be the most similar data set (e.g., Mittas et al. (2015) report using 501 projects, while we end up with 214, using our more restrictive subset). However, we are more interested in the cases where we have larger data sets, together with missing data, and comparing these with, e.g., 'A_clean'.

[14] Please see here for prior choice recommendations: `https://goo.gl/fx2F7V`.

Missing Data Imputation

Summarizing missing data (Table 4) shows that the missingness is multivariate (there is missing data in more than one variable), connected (the second row with data indicates that we have 214 rows that are complete, i.e., no data is missing for these rows), and non-monotone (we have zeros spread out within all the ones, i.e., there is no monotonicity). Generally speaking, this indicates that data imputation is possible (in particular, connectivity is an essential part of missing data imputation).

Table 4 Missingness of missing data. Top row lists each variable. Bottom row the number of missing entries per variable. First column, the frequency of each pattern. Last column, number of missing entries per pattern

	Effort	Added	Input	Output	Enquiry	File	Interface	Changed	
Freq.									**# missing entries**
214	1	1	1	1	1	1	1	1	0
14	1	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	0	1	1
2	1	1	1	1	1	1	0	0	2
1	1	1	1	1	1	0	0	1	2
2	1	1	1	1	0	1	1	0	2
1	1	1	1	1	0	0	1	0	3
3	1	1	1	0	1	1	1	0	2
1	1	1	1	0	1	1	0	0	3
1	1	1	1	0	1	0	0	1	3
2	1	1	0	1	0	0	1	0	4
5	1	1	0	0	0	0	0	1	5
4	1	1	0	0	0	0	0	0	6
1	1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	0	0	1	3
84	1	0	0	0	0	0	0	0	7
157	0	0	0	0	0	0	0	0	8
Missingness per variable	157	243	252	255	255	256	264	270	$\sum$ **1952**

van Buuren (2007) recommends that one calculates each variable's influx and outflux and plots them. Influx (I) is defined as the number of variable pairs with Y_j missing and Y_k observed, divided by the total number of *observed* data cells while, in the case of outflux (O), we instead divide by the total number of *incomplete* data cells. In short, a completely observed variable gives $I_j = 0$, while the opposite holds for O_j. If one has two variables with the same degree of missing data, then the variable with the highest O_j is potentially more useful for imputation purposes. Examining Fig. 5, `Effort` and `Change` have the highest O_j and I_j. To summarize, `Effort` will be the most influential variable for imputation, while `Change` will be the easiest variable to impute. This is worth keeping in mind later when we analyze the results.

To conclude, we will create multiple imputations (replacement values) for multivariate missing data, based on Fully Conditional Specification (van Buuren, 2007).

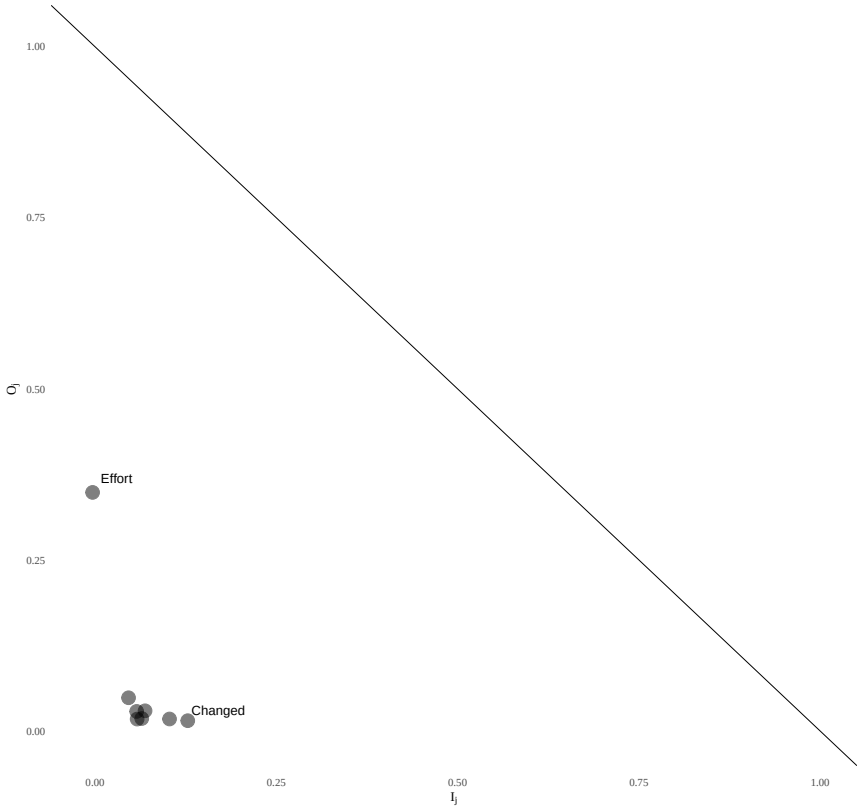


Fig. 5 Outflux versus influx of data set 'A' as described in Sect. 3. (A small degree of random noise, 'jitter', was added to the plot to make it more readable.)

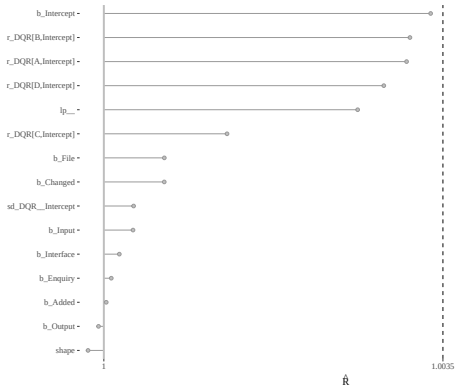
Each incomplete variable is imputed by a separate model using predictive mean matching (numeric data), or proportional odds model/ordered logit model (factor data with >2 ordered levels) (Rubin, 1986; van Buuren, 2007). Concerning predictive mean matching, the assumption is that the missingness follows approximately the same distribution as the data, but the variability between the imputations over repeated draws reflects the uncertainty of the actual value.

We will approach this conservatively and follow the latest guidelines, as already discussed in Sect. 3.1.3, and hence set the number of imputations $m = 25$ (see Table 3) since we have approximately 25% missingness in certain variables.

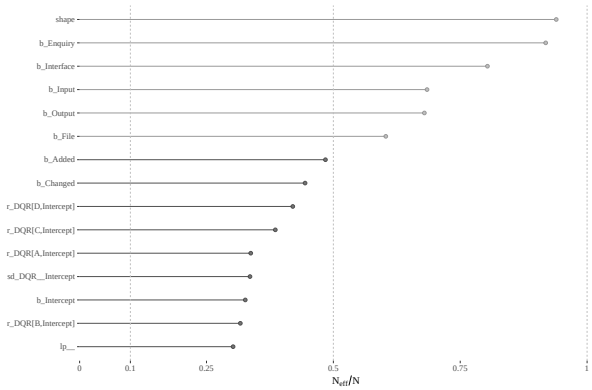
3.2.2 Diagnostics

In this section, we will first present some diagnostics from the Hamiltonian Monte Carlo sampling we conducted.

First, the ratio of the average variance of draws within each chain to the variance of the pooled draws across chains is an estimate we can use to see how well our chains have diverged towards a common posterior. This is measured by $\widehat{R}$ and generally speaking $\widehat{R}$ should go towards 1.00, and anything above 1.01 should be a clear warning sign of bias. In our case, for all sampling conducted, $\widehat{R}$ was consistently low (see Fig. 6a for one example where we used the 'AD_clean' data set).



(a)



(b)

Fig. 6 $\widehat{R}$ values for each parameter (a) and effective sample sizes (b). For $\widehat{R}$ (a), generally speaking, any values above 1.01 are not appropriate and indicates that one or more chains are biased. Concerning effective sample sizes (b), anything below 0.2 is generally speaking a warning sign of a misspecified model. Here we used the 'AD_clean' data set

Second, the effective sampling was consistently high, i.e., $n_{eff} \gg 0.2$. As an example, if we have 2,000 samples from each chain, the default is then to throw away the first 1,000 as warmup samples. But then we use four chains, ending up with

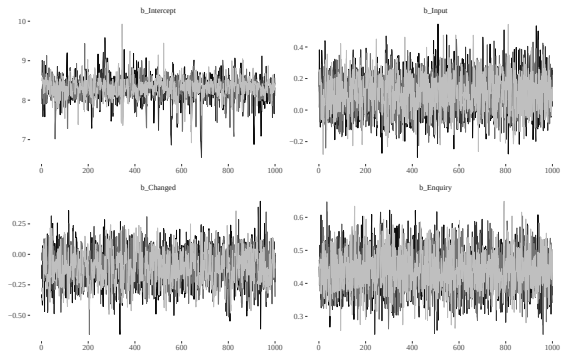


Fig. 7 Trace plots of four parameters using the 'AD_clean' data set

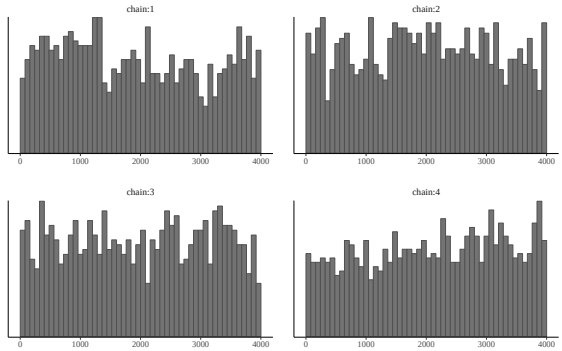


Fig. 8 Rank plots of the four chains from the $\beta_{\text{Intercept}}$ sampling. The chains should be close to uniform. Here we see a slight dip at the start of Chain 1 and the end of Chain 3. These dips can be much more exaggerated (e.g., no samples collected at all) and then there would be reasons to worry

$1,000 \times 4$ samples. This means that we should not, in our example, have less than 400 samples for a parameter (see Fig. 6b).

Third, the visual inspection revealed that the chains seemed to mix well (hairy caterpillar ocular test). It should look messy, tight, and mixed (Fig. 7). If we are a bit hesitant concerning the mixture, and in particular the sampling conducted at the tails, once could also use rank plots (Vehtari et al., 2019). If we investigate Fig. 7 we see a clear difference between $\beta_{\text{Intercept}}$ and the other parameters. Using rank plots for the chains for $\beta_{\text{Intercept}}$, Fig. 8, provides a better view. We can see that there is a dip at the start of Chain 1 and the end of Chain 3, but it still indicates that the chain managed to sample quite well at the tails.

Finally, the Bayesian fraction of missing information, another diagnostic, shows a significant overlap between the energy transition density, π_E, and the marginal energy distribution, $\pi_{\Delta E}$ (Fig. 9). When the two distributions are well-matched, the random walk will explore the marginal energy distribution efficiently (Betancourt, 2017).

We sampled four chains, each with 2,000 iterations, and the first half of the iterations were discarded as warmup iterations. For our imputed data sets this means that we have $1,000 \times 4 \times 25 = 100,000$ posterior samples. Figure 10 provides a comparison of our empirical outcome y (using the data set 'A_clean'), with draws

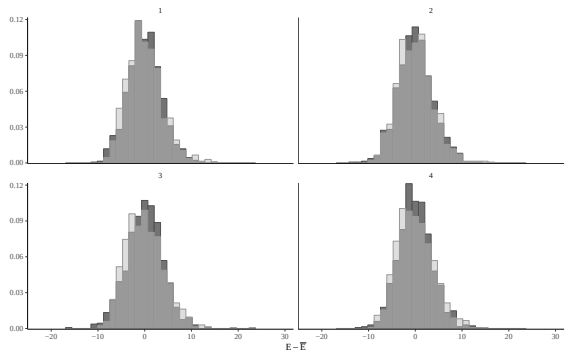


Fig. 9 Comparisons of the energy transition density, π_E, and the marginal energy distribution, $\pi_{\Delta E}$ (light and dark gray, respectively). A significant overlap is visible

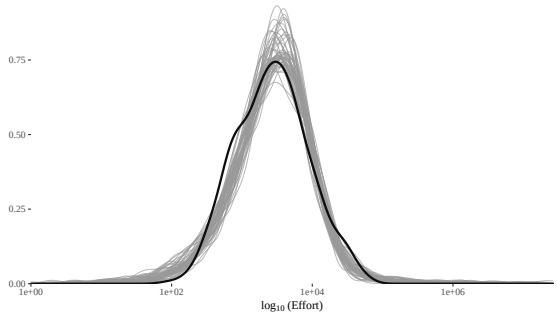


Fig. 10 Comparison of the empirical distribution of the data ('AD_clean'), to the distributions of simulated data from the posterior predictive distribution (50 samples). Note that the x-axis has been transformed

from our posterior distribution; we see evidence of a fairly good match; a perfect match is not what we want since then we could just use our data as-is, i.e., the variability of each y_{rep} vis-à-vis y is what interests us.

Another check one should do is to investigate how well the sampling (y_{rep}) matches our empirical data y for each project (row) in our data set. In Fig. 11, we have drawn 500 samples from the posterior. As we can see our empirical data, y, does not always match y_{rep}, but that is all fine actually, what we want is a model that *on average* makes better predictions. After all, if we would want perfect predictions for our data set, why not use the data set as-is?

3.3 Conduct Inference

If we turn our attention to the estimated intercepts for our group-level variable `DQR` (i.e., a project's rating according to the quality of data collected), we see something interesting in Fig. 12. There is a clear pattern, in both data sets, where quality rating 'A' and 'D' are to the left, while 'B' and 'C' are to the right. It is an indication that these two groups are perceived as similar to each other, which is a bit ironic since 'A' and 'D' are conceptually the opposite of each other in terms of data quality rating.

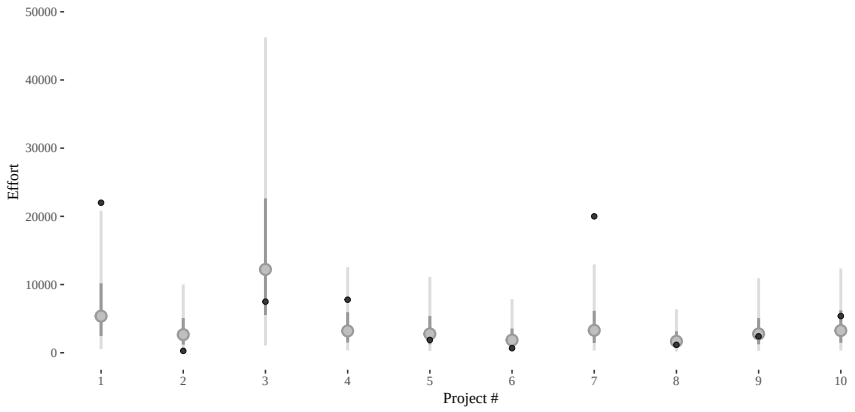


Fig. 11 Posterior predictive checks of the first 10 projects in our data set. Vertical bars with points indicate the simulated medians and darker points indicate our empirical values. Thicker and thinner lines indicate 50% and 90% central intervals, respectively. We drew 500 samples from the posterior

This indicates that one should question the data quality ratings in the data set and, given enough data, DQR seems to become less critical.

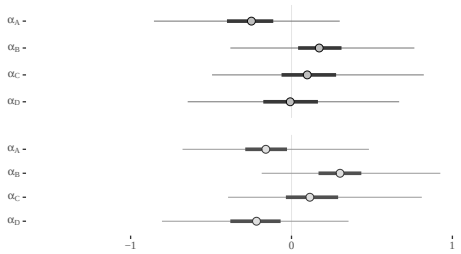


Fig. 12 Interval plots of α estimates with 50% and 95% uncertainty intervals. Upper plot is the imputed data set, below plot the cleaned data set. The uncertainty is slightly different between each category even though this is not very obvious in this plot (the cleaned data set, below, has higher uncertainty)

Examining the estimated parameters (Fig. 13) we see that parameters perceived as 'significant' differ between the data sets.[15] There are three comparisons we should make here. First, comparing imputed with cleaned data sets (within each column). Second, comparing 'A' and 'AD' models (between columns). Third, compare the upper right and lower left plots (data-greed approach vs. state of practice).

First, if we look at the left column, we see that nothing significant has changed. On the other hand, examining the right column, we see that β_{Added} is no longer significant in the imputed data set. This might make you sad. However, using all data can lead to weaker inferences and, honestly, should we not make use of all data available no matter our wishful thinking concerning inferences?

[15] Our notion of 'significant' is here that the 95% highest density posterior interval does not cross zero. However, other notions do exist (Kruschke, 2018).

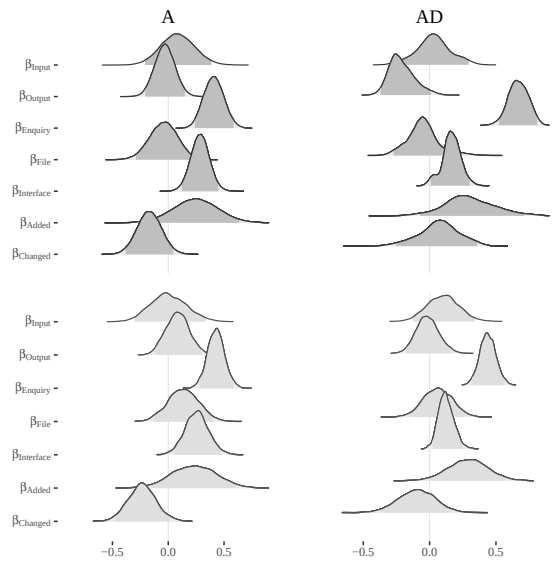


Fig. 13 Density plots drawn with overlapping ridgelines of β estimates and 95% uncertainty intervals. Left column presents data sets 'A', while right column presents data sets 'AD'. Dark gray plots indicate imputed data sets (first row), while light gray represents cleaned sets. In particular the top right and lower left plots are of interest to us since they represent the data-greed approach and the state of practice, respectively

Second, if we compare our simple models with our multi-level models (between column comparisons), examining the two lower plots we see that β_{Added} has become 'significant' in the model where we make use of all quality ratings (right plot). In this particular case, one would lean towards the multi-level model (right plot) since it, after all, makes use of more data and employs partial pooling to avoid over-fitting.

Finally, we should compare the upper right and lower left plots (our data-greed approach with state of practice); they are the reason for conducting this study. Two things are worth noticing here: (i) β_{Enquiry} is shifted noticeably more to the right in the imputed data set and is significant, as is $\beta_{\text{Interface}}$. (ii) β_{Changed} is clearly not significant in the imputed data set, while in the cleaned data set, it is nearly so. Once again, making use of more data can lead to weaker inferences, which is a good thing.

Let us now examine what the posterior predictive distribution provides us with concerning point estimates regarding our outcome `Effort`. The posterior distribution allows us to set predictors at different values and generate predicted outcomes with uncertainty. In our case, we would like to examine the difference between posterior predictive distributions of 'A_clean' and 'AD', since 'A_clean' is based on the assumptions commonly used in literature and 'AD' makes use of as much data as possible, i.e., our data-greed approach.

Plotting our two posterior distributions indicates the differences (see Fig. 14). As is clear, the median is higher when taking into account more data ('AD'), but the uncertainty is also slightly larger. Ultimately, comparing these types of point estimates should always go hand in hand with the purpose of the analysis, i.e., the utility function.

The posterior distribution allows us to make probabilistic statements that provide us with a deeper understanding of the phenomena we study. We could make

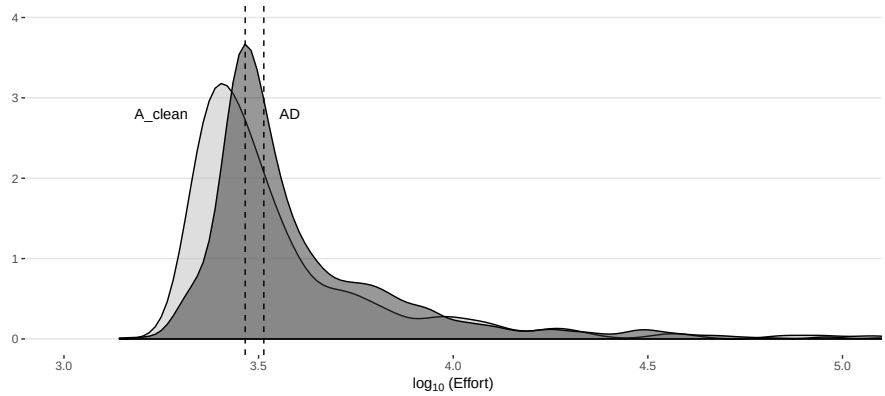


Fig. 14 Comparison of the posterior distributions of the 'A_clean' and 'AD' data sets (n = 4000). Notice the transformation of the x-axis. The median values on natural scale, with 95% highest posterior density intervals for 'A_clean' and 'AD', are $\tilde{\mu}_{\text{A_clean}} = 2937$, 95% HPDI[2069, 15441] and $\tilde{\mu}_{\text{AD}} = 3280$, 95% HPDI[1920, 17783], respectively (medians indicated by vertical lines). Highest posterior density interval (HPDI) is the tightest interval containing the specified probability mass, i.e., 95% in our case

statements for each model separately, i.e., investigating 'AD' we can say that the probability that the effect is greater than 10% (a decline of >10%) for the `D` category is 2.6%, while for the `A` category it is 7.2%. Alternatively, in the case of 'A_clean', that the probability that the effect of `Enquiry` is greater than 5% (an increase of >5% in this case) is 58.6%. Of course, one could also look at the probability that the estimates of parameter β_{Input} is larger than 0 using the 'AD_clean' and 'AD' data sets, i.e., 79.8% and 60.2%, respectively.

These types of probability statements are a positive aspect of Bayesian analysis and the posterior probability distributions that accompanies it.

3.4 Threats to Validity

In this section, we will cover threats to validity from a quantitative and statistical perspective. The below threats are not the type of threats that are normally discussed in empirical software engineering (Wohlin et al., 2012) (such as threats to internal and external validity). The latter refers to a rigid experimental design (often based on statistical hypothesis testing) and are mainly qualitative; in contrast, the threats we discuss are grounded in the quantitative analysis we performed, and as such they address the very design of the analysis (which allows for much more flexibility using the tools of Bayesian data analysis). In Sect. 3.4.1 we will compare our study's design to recent guidelines concerning the design and reporting of software analytics studies.

Directed Acyclic Graphs (DAGs). Making ones scientific model explicit is dangerous since it becomes open to attack. We believe, however, that it should be compulsory in any scientific publication. We employed DAGs to this end, a concept refined by Pearl and others (Pearl, 2009). Using DAGs make things explicit. If things are explicit, they can be criticized. One threat to validity is of course that our scientific model is wrong and `AFP` is not a mediator (Sect. 3.1.1). This is for the reader to comment on. Of course, instead of using the graphical approach of DAGs, and applying *do*-calculus to determine *d*-separation, one could walk down the road of numerical approaches according to Peters et al. (2017).

Non-Identifiability. Through our non-identifiability analysis we concluded that the variable `Deleted` should be removed. Removing this variable is a trade-off. The non-identifiability analysis showed that it should be removed, thus allowing for better understandability and better out of sample prediction. However, we could have taken a more prudent approach and investigated `Deleted`'s role in predictions, but in this particular case, we believe the initial analysis provided us with a convincing argument to remove it.

Priors. The sensitivity analysis of priors provided us with confidence regarding the choice of priors. We conducted prior predictive analysis and, together with recommendations regarding default priors, concluded that our selection of priors was balanced. However, our conclusions could be wrong, and further studies could indicate that our priors are too broad. The latter is, however, what one can expect when doing science.

Bayesian Data Analysis. In a Bayesian context, model comparison is often divided into three categories $\mathcal{M}$-open, $\mathcal{M}$-complete, and $\mathcal{M}$-closed (Yao et al., 2018; Navarro, 2019). In the $\mathcal{M}$-open world the relationship between the data generating process and our list of models $\mathcal{M} = M_1, \ldots, M_K$ is unknown. However, what we do know is that M_t, our 'true' model, is not in $\mathcal{M}$, and we cannot specify the explicit form $p(\tilde{y}|y)$ due to computational or conceptual reasons. In the $\mathcal{M}$-complete world, M_t is also not in $\mathcal{M}$, but we use any model in $\mathcal{M}$ that is closest in Kullback-Leibler divergence (Yao et al., 2018; Betancourt, 2015). Finally, in the case of the $\mathcal{M}$-closed world, $M_t \in \mathcal{M}$.

The bulk of statistical methodology is concerned with the latter category ($\mathcal{M}$-closed) and Clarke et al. (2013) claims that,

> this class of problems is comparatively simple and well studied.

Many problems we face are in the $\mathcal{M}$-complete and not the $\mathcal{M}$-closed world (this chapter is such an example). Selecting the 'best' model is often done through relative comparisons of $M_1, \ldots, M_K$ using the Watanabe-Akaike information criterion (WAIC) or leave one out cross-validation with Pareto-smoothed importance sampling (Vehtari et al., 2017). However, to use WAIC or PSIS-LOO for out of sample prediction, one should use the same data set for each model (e.g., you can change the likelihood and priors, but the data set is fixed).

In this chapter, we have not done model comparison (e.g., using PSIS-LOO (Vehtari et al., 2017)), but the reason is apparent—we use different data sets—which is the purpose of this chapter. To this end we defend our choice of likelihood, i.e.,

the Gamma-Poisson (a.k.a. negative binomial), epistemologically: If we have counts from zero to infinity, where the variance is significantly different from the mean then, from a maximum entropy point of view, Gamma-Poisson is a rational choice. By conducting posterior predictive checks, we ultimately received yet another validation to strengthen us in our opinion that the model has predictive capabilities (Sect. 3.3).

Residuals. One threat to validity is also the residuals of the model (fitting deviation). If the residuals are too large, it is an indication that the model does not fit data well. This is a trade-off since a perfect fit could imply overfitting. By conducting posterior predictive checks, we concluded that the models, as such, had a convincing fit (see, e.g., Fig. 10). However, investigating the residuals for each estimated parameter provides us with a better view.

In Fig. 15, we see estimates of the parameters with the most unobservable statistical error. The residuals are linear, which indicates predictability in our imputation. But, remember, `Changed` was judged to be the easiest to impute, but still `Enquiry` and `Interface` provide less uncertainty considering residuals (Sect. 3.1.3)—that is due to the imputation most likely. Nevertheless, in the end, we have employed multi-level models when possible and, thus, taken a conservative approach by using partial pooling. The latter should encourage us to put some trust in the model's ability to avoid overfitting, i.e., learn too much from our data.

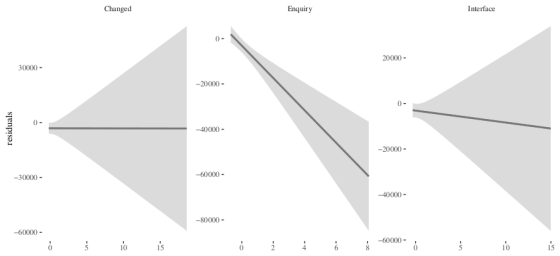


Fig. 15 Residuals of three parameters of interest

3.4.1 Common Threats in Software Analytics Papers

Finally, we believe that using the traditional threats to validity nomenclature seen in empirical software engineering research most likely does not fit the type of studies we present here. Instead we will propose something different.

Menzies and Shepperd (2019) presents 12 'bad smells' in software analytics papers. Below we will now cover each 'smell' and contrast it with what we did in this chapter.

1. **Not interesting.** (Research that has negligible software engineering impact.) We argue that the problem we have analyzed in this section is not only relevant, common, and interesting. To this we mainly point to Sect. 4.

2. **Not using related work.** (Unawareness of related work concerning RQs and SOA.) We point to further reading (Sect. 4) as a basis for this study, i.e., studies which throw away data, and show how other state of the art analytical approaches might be more suitable.
3. **Using deprecated and suspect data.** (Using data out of convenience.) The data is definitely suspect as we have shown by our analysis of data quality ratings, but it would be hard to argue that the data is deprecated. However, we used a particular version of the data set, but that was due to our intention to align with previous work.
4. **Inadequate reporting.** (Partial reporting, e.g., only means.) In this section, we have presented not only point estimates but also contrasted different distributions and derived probabilistic statements. We would have also liked to provide model comparisons but, alas, the design of this study did not allow this. To this end, we rely on posterior predictive checks.
5. **Under-powered experiments.** (Small effect sizes and little theory.) Using more data provides us with more statistical power, and we base our priors on state of the art recommendations and logical conclusions, e.g., estimating that the world's population is part of a project is not appropriate.
6. **$p < 0.05$ and all that.** (Abuse of null hypothesis testing.) We mention p-values only when making a point not to use them.
7. **Assumptions of normality and equal variances.** (impact of outliers and heteroscedasticity.) We use a Bayesian generalized linear model, which we model using a Gamma-Poisson likelihood. Additionally, we employ multi-level models when possible, and hence make use of partial pooling (which takes into account the presence of outliers).
8. **Not exploring stability.** We conducted a sensitivity analysis of priors, and we report on the differences between imputed and cleaned data sets.
9. **No data visualization.** We leave it up to the reader to decide if appropriate levels of visualization were used. We have followed guidelines on data visualization (Gabry et al., 2017).
10. **Not tuning.** We avoid bias in comparisons mainly by clearly stating our assumptions, conducting a sensitivity analysis, making use of multi-level models, and, generally speaking, following guidelines on how to conduct Bayesian data analysis.
11. **Not exploring simplicity.** Using state of the art missing data analysis is needed and wanted to decrease our bias. Additionally, using a complex mixture model was unavoidable because of epistemological reasons, as presented earlier in this section. We used simulated data to assess the appropriateness of our likelihood and priors independently.
12. **Not justifying choice of learner.** This concerns, ultimately, the risk of over-estimation (or over-fitting). We would argue that any usage of frequentist statistics would potentially introduce this 'smell', i.e., using uniform priors, as is the case in a traditional frequentist setting, ensures maximum over-fitting.

3.5 Discussion

We argue that one should have solid reasons to throw away data since we now have techniques available that can provide us with the opportunity to use as much data as possible. The example we provided showed that by using missing data techniques, in combination with Bayesian multi-level models, we could better make use of the data and, thus, gain higher confidence concerning our findings. The inferences can become weaker, but ask yourself if that is not how you think your fellow researchers should conduct their analysis. We could show two things in our analysis: (*i*) Parameters' 'significance' changed depending on if we used imputation or not, and (*ii*) there was really not much of a difference between the various data quality ratings (once again indicating that we should use as much data as possible).

However, we pay a price for this more sophisticated analysis. It is a more involved analysis compared to a frequentist analysis where only the likelihood of the outcome is required to be specified, and maximum likelihood estimates are not conditioned on the observed outcome; the uncertainty is instead connected to the sampling distribution of the estimator. The same applies to confidence intervals in the frequentist world, i.e., one can set up a distribution of predictions, but it entails repeating the process of random sampling on which we apply the estimator every time to then generate point predictions. Contrast this with conditioning on the posterior predictive distribution, which is based on the observed outcome.

Additionally, making probabilistic statements is very much more natural when having a posterior at hand, while the p-values we have made use of traditionally rely on observing a z-statistic that is so large (in magnitude) if the null hypothesis is true, i.e., not if the scientific hypothesis is true. To make the point, the term 'p-value' was used in this section for the first time here in this section and in our case, where we used different data sets, one could have expected us to lean towards traditional hypothesis testing, since it was not possible to compare models explicitly, regarding out of sample predictions.

We will not further contrast our approach with how analyses are done in empirical software engineering today. Suffice to say, issues such as the arbitrary $\alpha = .05$ cut-off, the usage of null hypothesis significance testing and the reliance on confidence intervals have been criticized (Ioannidis, 2005b; Morey et al., 2016; Nuzzo, 2014; Woolston, 2015), and when analyzing the arguments, we have concluded that many of the issues plaguing other scientific fields are equally relevant to come to terms with in empirical software engineering.

We believe that evidence-based interpretation is more straightforward with Bayesian data analysis, and empirical software engineering should embrace it as soon as possible. In our view, it is a natural choice to make in this particular case; to base one's inferences on more data is wise, and doing so in a Bayesian context is natural.

4 Recommended Further Reading

There are few early publications in software engineering where we see evidence of using MLMs. In (Ehrlich and Cataldo, 2012) the authors used multilevel models for assessing communication in global software development, while in (Hassan et al., 2017), the authors applied MLMs for studying reviews in an app store. However, both studies used a frequentist approach (maximum likelihood), i.e., not a Bayesian approach.

As far as we can tell, there are only a few examples of studies in software engineering that have applied BDA with MLMs to this date (Furia, 2016; Ernst, 2018). Furia (2016) presents several cases of how BDA could be used in computer science and software engineering research. In particular, the aspects of including prior belief/knowledge in MLMs are emphasized, which is further elaborated on in (Furia et al., 2019). Ernst (2018), on the other hand, presents a conceptual replication of an existing study where he shows that MLMs support cross-project comparisons while preserving local context, mainly through the concept of partial pooling, as used in Bayesian MLMs.[16]

Finally, much literature on BDA exist, but not all have the clarity that is needed to explain, sometimes, relatively complex concepts. If one would like to read up on the basics of probability and Bayesian statistics we recommend (Jaynes, 2003), for a slightly more in-depth view of Bayesian statistics we would recommend (Lambert, 2018). For a hands-on approach to BDA, we recommend (McElreath, 2015); McElreath's book *Statistical Rethinking: A Bayesian Course with Examples in R and Stan* is an example of how seemingly complex issues can be explained beautifully, while at the same time help the reader improve their skills in BDA. To conclude, there is one book that every researcher should have on their shelf, *Bayesian Data Analysis* by Gelman et al. (2013), which is considered the leading text on Bayesian methods.

Missing data can be handled in two main ways. Either we delete data using one of three main approaches (listwise or pairwise deletion, and column deletion) or we impute new data. Concerning missing data, we conclude the matter is not new to the empirical software engineering community. Liebchen and Shepperd (2008) have pointed out that the community needs more research into ways of identifying and repairing noisy cases, and Mockus (2008) claims that the standard approach to handle missing data, i.e., remove cases of missing values (e.g., listwise deletion), is prevalent in empirical software engineering.

Two additional studies on missing data are, however, worthwhile pointing out. Myrtveit et al. (2001) investigated listwise deletion, mean imputation, similar response pattern imputation, and full information maximum likelihood (FIML), and conclude that FIML is the only technique appropriate when data are not missing completely at random. Finally, Cartwright et al. (2003) conclude that k-nearest neighbor and sample mean imputation are significantly better in improving model fit when dealing with imputation. However, much has happened concerning research in imputation techniques lately.

[16] Partial pooling takes into account variance between units.

In this chapter, we focused on multivariate imputation by chained equations (MICE), sometimes called fully conditional specification or sequential regression multiple imputation, a technique that has emerged as a principled method of dealing with missing data during the last decade (van Buuren, 2007). MICE specifies a multivariate imputation model on a variable-by-variable basis by a set of conditional densities (one for each incomplete variable) and draws imputations by reiterating the conditional densities. The original idea behind MICE is old, see, e.g., stochastic relaxation (Geman and Geman, 1984), but the recent refinements and implementations have made the technique easily accessible.

Finally, related work concerning the ISBSG data set is worthwhile pointing out. Fernández-Diego and de Guevara (2014) present pros and cons of using the ISBSG data set. A systematic mapping review was used as the research method, which was applied to over 120 papers. The dependent variable of interest was usually `Effort` (more than 70% of the studies), and the most frequently used methods were regression (~60%) and machine learning (~35%), the latter a term where many techniques can hide. Worth noting is also that Release 10 was used most frequently, which provided us with a reason to also use that data set. Additionally, we also used `Effort` as the dependent variable of interest, since a majority of the studies seem to find that variable interesting to study. (The importance of the ISBSG data set, when considering replication studies in empirical software engineering, has already been pointed out by Shepperd et al. (2018).) By and large, our chapter took another approach entirely, we imputed missing data in a Bayesian context, and we see this more as complementary to some of the work mentioned above.

5 Conclusion

In this chapter, we introduced the reader to Bayesian data analysis. Before even designing the model, we took several steps, each providing us with a better understanding of the data. We did a causal analysis, analyzed non-identifiability, performed sensitivity analysis of priors, and an analysis of missing data. Additionally, we presented the reader with several diagnostics one should use for sanity checking a statistical model. Except for the missing data analysis (if no missingness is present), we would argue that this is something one should always do when conducting Bayesian data analysis.

Missing data was an additional complexity that our case presented. We recommend that one should always be conservative with throwing away data. Many state of the art techniques exist today, which provides the researcher with ample of possibilities to conduct rigorous, systematic, and transparent missing data analysis. We followed a traditional imputation approach, but other approaches, i.e., purely Bayesian, do exist. In our example, we showed that inferences can become weaker, which is not a bad thing, and that the qualitative assessment of quality ratings can be biased. This further strengthens the argument never to throw data away.

By using Bayesian data analysis, we believe that researchers will be able to get a more nuanced view of the challenges they are investigating. In short, we do not need p-values for this.

References

Aarts AA, et al. (2015) Estimating the reproducibility of psychological science. Science 349(6251), DOI 10.1126/science.aac4716

Banerjee S, Carlin B, Gelfand A (2014) Hierarchical modeling and analysis for spatial data, 2nd edn. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis

Benjamin DJ, et al. (2018) Redefine statistical significance. Nature Human Behaviour 2:6–10, DOI 10.1038/s41562-017-0189-z

Betancourt M (2015) A unified treatment of predictive model comparison. arXiv e-prints 1506.02273

Betancourt M (2017) A conceptual introduction to Hamiltonian Monte Carlo. arXiv e-prints 1701.02434

Betancourt M (2018) Calibrating model-based inferences and decisions. arXiv e-prints arXiv:1803.08393, 1803.08393

Bodner TE (2008) What improves with increased missing data imputations? Structural Equation Modeling: A Multidisciplinary Journal 15(4):651–675, DOI 10.1080/10705510802339072

Brooks S, Gelman A, Jones G, Meng XL (2011) Handbook of Markov Chain Monte Carlo. CRC press

Bürkner PC (2017) brms: An R package for Bayesian multilevel models using Stan. Journal of Statistical Software 80(1):1–28, DOI 10.18637/jss.v080.i01

van Buuren S (2007) Multiple imputation of discrete and continuous data by fully conditional specification. Statistical Methods in Medical Research 16(3):219–242, DOI 10.1177/0962280206074463

Camerer CF, et al. (2016) Evaluating replicability of laboratory experiments in economics. Science 351(6280):1433–1436, DOI 10.1126/science.aaf0918

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: A probabilistic programming language. Journal of Statistical Software 76(1):1–32, DOI 10.18637/jss.v076.i01

Cartwright MH, Shepperd MJ, Song Q (2003) Dealing with missing software project data. In: Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717), pp 154–165, DOI 10.1109/METRIC.2003.1232464

Clarke JL, Clarke B, Yu CW (2013) Prediction in $\mathcal{M}$-complete problems with limited sample size. Bayesian Analysis 8(3):647–690, DOI 10.1214/13-BA826

Dutilh G, Vandekerckhove J, Ly A, Matzke D, Pedroni A, Frey R, Rieskamp J, Wagenmakers EJ (2017) A test of the diffusion model explanation for the worst performance rule using preregistration and blinding. Attention, Perception, & Psychophysics 79(3):713–725, DOI 10.3758/s13414-017-1304-y

Ehrlich K, Cataldo M (2012) All-for-one and one-for-all?: A multi-level analysis of communication patterns and individual performance in geographically distributed software development. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, ACM, New York, NY, USA, CSCW '12, pp 945–954, DOI 10.1145/2145204.2145345

Ernst NA (2018) Bayesian hierarchical modelling for tailoring metric thresholds. In: Proceedings of the 15th International Conference on Mining Software Repositories, IEEE Press, Piscataway, NJ, USA, MSR '18, p pp, DOI 10.1145/3196398.3196443

Fernández-Diego M, de Guevara FGL (2014) Potential and limitations of the ISBSG dataset in enhancing software engineering research: A mapping review. Information and Software Technology 56(6):527–544, DOI 10.1016/j.infsof.2014.01.003

Furia CA (2016) Bayesian statistics in software engineering: Practical guide and case studies. ArXiv e-prints 1608.06865

Furia CA, Feldt R, Torkar R (2019) Bayesian Data Analysis in empirical software engineering research. IEEE Transactions on Software Engineering DOI 10.1109/TSE.2019.2935974

Gabry J, Simpson D, Vehtari A, Betancourt M, Gelman A (2017) Visualization in Bayesian workflow. ArXiv e-prints 1709.01449

Gelman A (2018) The failure of null hypothesis significance testing when studying incremental changes, and what to do about it. Personality and Social Psychology Bulletin 44(1):16–23, DOI 10.1177/0146167217729162

Gelman A, Carlin J, Stern H, Dunson D, Vehtari A, Rubin D (2013) Bayesian data analysis, 3rd edn. Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis

Gelman A, Simpson D, Betancourt M (2017) The prior can often only be understood in the context of the likelihood. Entropy 19(10), DOI 10.3390/e19100555

Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6(6):721–741, DOI 10.1109/TPAMI.1984.4767596

Glick JL (1992) Scientific data audit—A key management tool. Accountability in Research 2(3):153–168, DOI 10.1080/08989629208573811

Hassan S, Tantithamthavorn C, Bezemer CP, Hassan AE (2017) Studying the dialogue between users and developers of free apps in the Google Play Store. Empirical Software Engineering DOI 10.1007/s10664-017-9538-9

Hill PR, Stringer M, Lokan C, Wright T (2001) Organizational benchmarking using the ISBSG data repository. IEEE Software 18:26–32, DOI 10.1109/52.951491

Hu MC, Pavlicova M, Nunes EV (2011) Zero-inflated and hurdle models of count data with extra zeros: Examples from an HIV-risk reduction intervention trial. The American Journal of Drug and Alcohol Abuse 37(5):367–375, DOI 10.3109/00952990.2011.597280

Hunter JE (2001) The desperate need for replications. Journal of Consumer Research 28(1):149–158, DOI 10.1086/321953

Ioannidis JPA (2005a) Contradicted and initially stronger effects in highly cited clinical research. JAMA 294(2):218–228, DOI 10.1001/jama.294.2.218

Ioannidis JPA (2005b) Why most published research findings are false. PLOS Medicine 2(8), DOI 10.1371/journal.pmed.0020124

Ioannidis JPA (2016) Why most clinical research is not useful. PLOS Medicine 13(6):1–10, DOI 10.1371/journal.pmed.1002049

Ioannidis JPA, Stanley TD, Doucouliagos H (2017) The power of bias in economics research. The Economic Journal 127(605):F236–F265, DOI 10.1111/ecoj.12461

Jaynes ET (2003) Probability theory: The logic of science. Cambridge University Press

John LK, Loewenstein G, Prelec D (2012) Measuring the prevalence of questionable research practices with incentives for truth telling. Psychological Science 23(5):524–532, DOI 10.1177/0956797611430953

Keung J (2008) Empirical evaluation of Analogy-X for software cost estimation. In: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM, New York, NY, USA, ESEM '08, pp 294–296, DOI 10.1145/1414004.1414057

Kruschke JK (2018) Rejecting or accepting parameter values in Bayesian estimation. Advances in Methods and Practices in Psychological Science DOI 10.1177/2515245918771304

Lambert B (2018) A student's guide to Bayesian statistics. SAGE Publications

Lenberg P, Feldt R, Wallgren Tengberg LG, Tidefors I, Graziotin D (2017) Behavioral software engineering - guidelines for qualitative studies. arXiv e-prints arXiv:1712.08341, 1712.08341

Liebchen GA, Shepperd M (2008) Data sets and data quality in software engineering. In: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering, ACM, New York, NY, USA, PROMISE '08, pp 39–44, DOI 10.1145/1370788.1370799

McElreath R (2015) Statistical rethinking: A Bayesian course with examples in R and Stan. CRC Press

McShane BB, Gal D, Gelman A, Robert C, Tackett JL (2017) Abandon statistical significance. ArXiv e-prints 1709.07588

Menzies T, Shepperd M (2019) "bad smells" in software analytics papers. Information and Software Technology 112:35–47, DOI 10.1016/j.infsof.2019.04.005

Mittas N, Papatheocharous E, Angelis L, Andreou AS (2015) Integrating non-parametric models with linear components for producing software cost estimations. Journal of Systems and Software 99:120–134, DOI 10.1016/j.jss.2014.09.025

Mockus A (2008) Missing data in software engineering. In: Shull F, Singer J, Sjøberg DIK (eds) Guide to Advanced Empirical Software Engineering, Springer London, London, pp 185–200, DOI 10.1007/978-1-84800-044-5_7

Morey RD, Hoekstra R, Rouder JN, Lee MD, Wagenmakers EJ (2016) The fallacy of placing confidence in confidence intervals. Psychonomic Bulletin & Review 23(1):103–123, DOI 10.3758/s13423-015-0947-8

Myrtveit I, Stensrud E, Olsson UH (2001) Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods. IEEE Transactions on Software Engineering 27(11):999–1013, DOI 10.1109/32.965340

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

Nuzzo R (2014) Scientific method: Statistical errors. Nature 506(7487):150–152, DOI 10.1038/506150a

Pearl J (2009) Causality: Models, reasoning and inference, 2nd edn. Cambridge University Press, New York, NY, USA

Peters J, Janzing D, Schölkopf B (2017) Elements of causal inference: Foundations and learning algorithms. Adaptive Computation and Machine Learning, MIT Press

R Core Team (2018) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

Rodríguez-Pérez G, Robles G, González-Barahona JM (2018) Reproducibility and credibility in empirical software engineering: A case study based on a systematic literature review of the use of the SZZ algorithm. Information and Software Technology 99:164–176, DOI 10.1016/j.infsof.2018.03.009

Rosenbaum PR (1984) The consequences of adjustment for a concomitant variable that has been affected by the treatment. Journal of the Royal Statistical Society Series A (General) 147(5):656–666

Rubin DB (1986) Statistical matching using file concatenation with adjusted weights and multiple imputations. Journal of Business & Economic Statistics 4:87–94, DOI 10.1080/07350015.1986.10509497

Rubin DB (1987) Multiple imputation for nonresponse in surveys. Wiley

Shanks DR, et al. (2013) Priming intelligent behavior: An elusive phenomenon. PLOS ONE 8(4):1–10, DOI 10.1371/journal.pone.0056515

Shepperd M, Ajienka N, Counsell S (2018) The role and value of replication in empirical software engineering results. Information and Software Technology 99:120–132, DOI 10.1016/j.infsof.2018.01.006

Simpson DP, Rue H, Martins TG, Riebler A, Sørbye SH (2014) Penalising model component complexity: A principled, practical approach to constructing priors. ArXiv e-prints 1403.4630

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

Torkar R, Feldt R, de Oliveira Neto FG, Gren L (2017) Statistical and practical significance of empirical software engineering research: A maturity model. CoRR abs/1706.00933, 1706.00933

Trafimow D, Marks M (2015) Editorial. Basic and Applied Social Psychology 37(1):1–2, DOI 10.1080/01973533.2015.1012991

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

Vehtari A, Gelman A, Simpson D, Carpenter B, Bürkner PC (2019) Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of mcmc. arXiv e-prints arXiv:1903.08008, 1903.08008

White IR, Royston P, Wood AM (2011) Multiple imputation using chained equations: Issues and guidance for practice. Statistics in Medicine 30(4):377–399, DOI 10.1002/sim.4067

Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in Software Engineering. Springer Publishing Company

Woolston C (2015) Psychology journal bans P values. Nature 519(7541):9, DOI 10.1038/519009f

Yao Y, Vehtari A, Simpson D, Gelman A (2018) Using stacking to average Bayesian predictive distributions (with discussion). Bayesian Analysis 13(3):917–1007, DOI 10.1214/17-BA1091