Applying Bayesian Analysis Guidelines to Empirical Software Engineering Data: The Case of Programming Languages and Code Quality

CARLO A. FURIA, Software Institute, USI Università della Svizzera italiana, Switzerland
RICHARD TORKAR, University of Gothenburg, Sweden and Stellenbosch Institute for Advanced Study (STIAS), South Africa
ROBERT FELDT, Chalmers and University of Gothenburg, Sweden

Statistical analysis is the tool of choice to turn data into information and then information into empirical knowledge. However, the process that goes from data to knowledge is long, uncertain, and riddled with pitfalls. To be valid, it should be supported by detailed, rigorous guidelines that help ferret out issues with the data or model and lead to qualified results that strike a reasonable balance between generality and practical relevance. Such guidelines are being developed by statisticians to support the latest techniques for *Bayesian* data analysis. In this article, we frame these guidelines in a way that is apt to empirical research in software engineering.

To demonstrate the guidelines in practice, we apply them to reanalyze a GitHub dataset about code quality in different programming languages. The dataset's original analysis [Ray et al. 2014] and a critical reanalysis [Berger et al. 2019] have attracted considerable attention—in no small part because they target a topic (the impact of different programming languages) on which strong opinions abound. The goals of our reanalysis are largely orthogonal to this previous work, as we are concerned with demonstrating, on data in an interesting domain, how to build a principled Bayesian data analysis and to showcase its benefits. In the process, we will also shed light on some critical aspects of the analyzed data and of the relationship between programming languages and code quality—such as the impact of project-specific characteristics other than the used programming language.

The high-level conclusions of our exercise will be that Bayesian statistical techniques can be applied to analyze software engineering data in a way that is principled, flexible, and leads to convincing results that inform the state-of-the-art while highlighting the boundaries of its validity. The guidelines can support building solid statistical analyses and connecting their results. Thus, they can help buttress continued progress in empirical software engineering research.

CCS Concepts: • **Mathematics of computing → Bayesian computation**; • **Software and its engineering → Empirical software validation;**

Additional Key Words and Phrases: Bayesian data analysis, statistical analysis, guidelines, empirical software engineering, programming languages

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018–05973.

Authors' addresses: Carlo A. Furia (furiac@usi.ch): Software Institute, USI Universitá della Svizzera italiana, Via Giuseppe Buffi 13, CH-6904 Lugano, Switzerland; Richard Torkar (Richard.Torkar@cse.gu.se) and Robert Feldt (robert.feldt @chalmers.se): Chalmers University of Technology/University of Gothenburg, Department of Computer Science and Engineering, SE-412 96 Göteborg, Sweden.

1049-331X/2022/03-ART40 $15.00
https://doi.org/10.1145/3490953

ACM Reference format:
Carlo A. Furia, Richard Torkar, and Robert Feldt. 2022. Applying Bayesian Analysis Guidelines to Empirical Software Engineering Data: The Case of Programming Languages and Code Quality. *ACM Trans. Softw. Eng. Methodol.* 31, 3, Article 40 (March 2022), 38 pages.
https://doi.org/10.1145/3490953

1 INTRODUCTION

Empirical disciplines, including a substantial part of software engineering research, mine data for information and then use the information as evidence to build, extend, and refine empirical knowledge. Statistical analysis is key to implementing this process. However, statistical techniques are merely tools, which need detailed *guidelines* to be applied properly and consistently. It is only through the combination of powerful statistical techniques and rigorous guidelines to apply them that we can distill empirical knowledge following a process that is consistent, rests on solid principles, and ultimately is more likely to lead to valid results with a higher degree of confidence.

Whereas frequentist statistical techniques have been commonplace in science for over a century—since the influential work of the likes of Pearson [68] and Fisher [20]—the state-of-the-art in applied statistics is moving towards using *Bayesian* analysis techniques. As we discussed in previous work [21], recent developments in Bayesian analysis techniques (such as using Hamiltonian Monte Carlo fitting algorithms [11]), coupled with an increasing availability of the computing power needed to run them on large datasets, have convincingly demonstrated the advantages of using Bayesian statistics and the flexibility and rigor of the analysis that they support. More recently, applied statisticians have also been working out practical *guidelines* that can boost usability and impact of Bayesian statistical data analysis [1, 23, 31, 57]. In this article, we present some of these guidelines and frame them in a way that is suitable for empirical research in the software engineering domain with the goal of demonstrating how they can support a principled way of building statistical analyses of software engineering data.

To demonstrate the guidelines in practice, we follow them to analyze a large dataset about the code quality of projects written in disparate programming languages and hosted on GitHub [55]. The empirical study that curated this dataset and performed the original analysis [55] was followed by a critical reanalysis by a different group of researchers [6]. As we recall in Section 1.1, the topic has received much attention and stirred some controversy. This visibility makes the dataset an attractive target for our own purposes.

In the article, we go through various aspects of the data analysis performed in the previous studies [6, 55], illustrating the versatile features of Bayesian statistical models in practice. We demonstrate how the guidelines support an incremental and iterative analysis process, in which several key features of a statistical model can be validated. This, in turn, encourages trying out different models and comparing them in a rigorous way as opposed to blindly relying on one-size-fits-all rules of thumb. Following this process, we demonstrate that some issues of the original analysis [55] or criticized by the follow-up reanalysis [6] could have been identified more easily. Furthermore, the limitations and actual impact of previous studies could have been framed more straightforwardly and more transparently. The conclusion of our exercise will be that flexible statistical techniques coupled with principled and structured guidelines can help address empirical research questions directly and transparently. This can lead to explanations that are nuanced and detailed. Hence, ultimately, they can become convincing foundations for building shared knowledge.