Two Differing Approaches to Survival Analysis of Open Source Python Projects

DEREK ROBINSON, KEANELEK ENNS, NEHA KOULECAR, and MANISH SIHAG,
University of Victoria

CCS Concepts: • **Information systems → Data mining**;

Additional Key Words and Phrases: data science, survival analysis, open source, python, Kaplan Meier, Cox proportional hazards model, Bayesian analysis

ACM Reference format:

Derek Robinson, Keanelek Enns, Neha Koulecar, and Manish Sihag. 2021. Two Differing Approaches to Survival Analysis of Open Source Python Projects. 1, 1, Article 1 (November 2021), 3 pages.

https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The developers of Open Source Software (OSS) projects are often part of decentralized and geographically distributed teams of volunteers. As these developers volunteer their free time to build such OSS projects, they likely want to be confident that the projects they work on will not become inactive. If OSS developers are aware of key attributes that are associated with long-lasting projects, they can make informed assessments of a given project before devoting their time to it or they can strive to make their own projects exhibit those attributes. Understanding which attributes of an OSS project lead to its longevity is what motivated Ali *et al.* to apply survival analysis techniques commonly found in biostatistics to study the probability of survival for popular OSS Python projects [1]. Ali *et al.* specifically studied the effect of the following attributes on the survival of OSS Python projects: publishing major releases, the use of multiple repositories, the type of version control system (VCS), and the size of the volunteer team.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
Manuscript submitted to ACM

Survival analysis is a set of methods used to determine how long an entity will live (or the time to a given event) and is most often used in the medical field. **[Keanu: do we need to find a citation for this claim/definition?]** For example, survival analysis techniques can determine the probability that a patient will survive when given a certain treatment. Ali *et al.* use a frequentist approach to survival analysis with methods including the Kaplan-Meier (K-M) survival estimator and the Cox Proportional-Hazards model. Though there are advantages to using such approaches **[Keanu: list some advantages and cite]**, another approach to survival analysis, Bayesian analysis, has its own set of advantages, namely **[Keanu: find the advantages, list them, cite]**.

The authors of this paper resonate with Ali *et al.'s* motivation. This paper serves as a replication of [1] and seeks to determine if there are gaps or shortcomings in their analysis. This replication also provides artifacts so that others may see how the study was conducted and reproduce it with ease. In addition to the replication, this paper analyzes the same data set using a Bayesian approach to survival analysis as outlined in [2] and seeks to compare the results of frequentist and bayesian approaches in the same domain. Thus, the research questions we plan on answering are as follows:

RQ1. How do major releases, the use of multiple repositories, the type of VCS, and the size of the volunteer team affect the survival of an OSS Python project?

RQ2. How do the findings of frequentist survival analysis differ from Bayesian survival analysis?

2 METHODS

2.1 Replication

2.2 Bayesian Survival Analysis

3 RESULTS

4 LIMITATIONS

4.1 Limitations of the Methods

4.2 Limitations of the Data

5 CONCLUSION

REFERENCES

[1] Rao Hamza Ali, Chelsea Parlett-Pelleriti, and Erik Linstead. 2020. Cheating Death: A Statistical Survival Analysis of Publicly Available Python Projects. In *Proceedings of the 17th International Conference on Mining Software Repositories*. 6–10.

[2] Riko Kelter. 2020. Bayesian survival analysis in STAN for improved measuring of uncertainty in parameter estimates. *Measurement: Interdisciplinary Research and Perspectives* 18, 2 (2020), 101–109.

A ARTIFACTS

Project Repository: https://github.com/DerekRobin/CSC578B-Project