

# Survival Analysis - A Promising Technique for Empirical Software Evolution Studies

Mathieu Goeminne

COMPLEXYS Research Institute, University of Mons

mathieu.goeminne@umons.ac.be

**Abstract**—In empirical studies on software evolution, many research questions relate to the time-dependent nature of the systems being analysed. However, the specificities of such a nature are too often neglected. The statistical technique of survival analysis can provide a solution to this problem. Survival analysis aims to analyse the distribution of the occurrences of a particular event over time, and takes into account absence of data points. We illustrate the opportunities offered by survival analysis by presenting some empirical studies on software evolution in which the technique has been used.

## I. INTRODUCTION

The analysis of evolving software systems often aims to predict how the system will evolve in the future, in order to have better control over this evolution process. In particular, the prediction of future software failures or the occurrence of specific events may be useful for understanding the impact of external factors on the considered systems. It also may help to make informed decisions on how to reduce down time and risk of failure, to increase the time before failure, etc.

Several metrics have been proposed to measure the reliability of a system, i.e., its probability that a particular event (such as a failure) does not occur over time. The time to first failure (TFF) and mean time to first failure (MTFF) are simple yet convenient measures for assessing a system's reliability, because they reduce a potentially complex system behaviour to a single scalar value. They are mainly used in hardware engineering [15], but have also found their way in software engineering [8], [20], healthcare [12], [19], [13] and criminology [9], [1], among others.

Unfortunately, these and similar approaches suffer from several weaknesses that limit their value when studying complex systems:

- They strongly **aggregate** the observations, and hide the temporal distribution.
- They don't take into account **censored data**.

Two types of censoring might occur:

- With *right censoring*, if a subject leaves the study before the event of interest could occur, the occurrence of the event after that point remains unknown for that particular subject. For example, in studies for a particular medical treatment, patients may drop-out before the end of the study. Right censoring also deals with the fact that the observation period may end before the event of interest could occur on some of the studied subjects.

- With *left censoring*, the event of interest may already have occurred before the subject was enrolled in the study, in which case the occurrence time cannot be determined.

While generally neglected in empirical studies of evolving software systems, right censoring is a frequently recurring property. For example, when studying software projects hosted in some open source forge, some projects may leave the forge before the end of the study.

In the rest of this paper, we review the notion of survival analysis that aims to circumvent these limitations. We also present some empirical studies on software engineering based on a survival analysis.

## II. SURVIVAL ANALYSIS

### A. Survival Function

The *survival function* of a population is the probability that one does not observe the occurrence of a given event for a member of this population before a given time. Depending on the context of use, the event may be referred to by a more specific term such as *failure* or *death*. To conform with common usage, and despite the fact that the occurrence of any type of event can be considered in a survival study, we will use the term *death* and its associated terminology here to refer to the considered event.

The survival function is monotonously decreasing, with a value between 0 and 1. It offers an easily interpretable visualisation of the reliability of a studied population, and can be used to determine the median survival time of the population, or, conversely, the probability that a particular subject survives longer than a given time.

If there are no right-censored observations, its value at time 0 is 1 and the entire survival function is equal to the complement of the cumulative distribution function:

$$S(t) = \frac{n_t}{N} \quad (1)$$

Where  $n_t$  is the number of subjects still alive at time  $t$ , and  $N$  is the population size.

### B. Kaplan-Meier Estimator

A more complex model is required for taking into account and quantifying the uncertainty due to censored observations. The Kaplan-Meier estimator [14] is often used for this purpose

in various research domains, including hardware engineering [17] and healthcare [4]. Because this estimator is non-parametric, it can be used even if the shape of the survival distribution over time is unknown.

This estimator determines the maximum likelihood of a survival function, i.e., the maximum probability that a subject survives longer than a given time  $t$ . More specifically, for a given population of  $N$  subjects, if  $t_1 \leq \dots \leq t_N$  are the times at which each of the subjects have been last observed (either because they died, or because their histories are censored after these times), the estimator  $\hat{S}(t)$  of the survival function of the population is based on the number of subjects that *risk* to be seen dead after  $t$ :

$$\hat{S}(t) = \prod_{t_i < t} \frac{n_i - d_i}{n_i} \quad (2)$$

Where  $n_i$  is the number of living subjects that are not censored just prior  $t_i$  and  $d_i$  is the number of subjects that died at time  $t_i$ .

The survival curve is sometimes plotted with an estimated confidence interval around the maximum likelihood estimate. Greenwood's formula [11] is a frequently used approach for estimating the standard error of the Kaplan-Meier estimator.

Figure 1 is a typical representation of the survival function of an hypothetical study in which each subject is a HIV positive patient<sup>1</sup>. The function is represented by a piecewise-constant curve that only decreases at the times corresponding to the (uncensored) death of one or many subject(s). Ticks are added to the curve to mark censored observations, while a confidence interval of 0.95% surrounds the survival function.

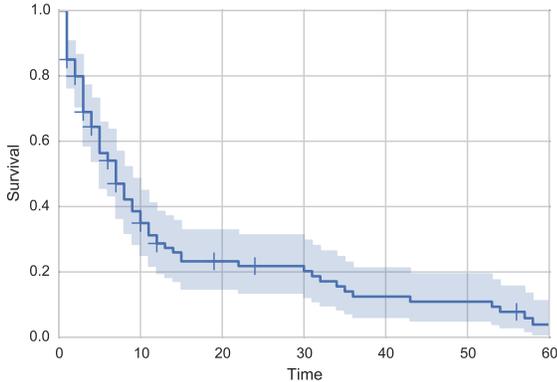


Fig. 1. Survival function of HIV positive patients.

### C. Comparing Multiple Survival Functions

The comparison of multiple survival functions is a common step in survival analysis. For instance, the efficacy of a treatment can be determined by comparing the survival function of the group of patients who benefit from the tested drug to the survival function of a control group that takes

a placebo [7]. The population can also be split according to some intrinsic property, such as the gender [24], in order to highlight significantly different survival behaviours between subpopulations.

Due to the presence of censored observations, the Wilcoxon–Mann–Whitney test cannot be used to determine if two survival functions are significantly different. The log-rank test, also known as the Mantel-Cox test [16], [3] is generally used instead. However, its ability to determine the accuracy of predictive survival models is controversial. This has led to alternatives such as the  $F^*$  test proposed by Berthy et al. [2]. Whichever test is chosen for comparing survival functions, the interpretation of a significant difference remains the responsibility of the analyst: in some cases, a significantly different survival function can not be considered to be ‘better’ or ‘worse’ than another one.

Figure 2 refines the results obtained in Figure 1 by distinguishing IV drug users from the other HIV positive patients. The Mantel-Cox test reveals a significant ( $p < 0.01$ ) difference between the two survival functions.

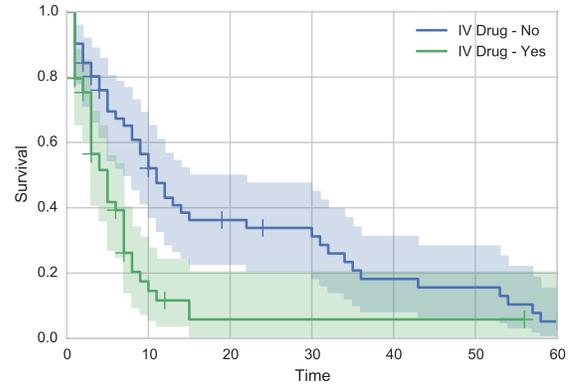


Fig. 2. Survival functions distinguishing IV drug users from other HIV positive patients.

As usual, when multiple statistics are considered simultaneously, the multiple comparison problem must be taken into account. The risk to incorrectly reject one of the tested null hypotheses, and therefore the risk to incorrectly consider that two survival functions are significantly different, increases with the number of survival functions being pairwise compared. The Bonferroni and the Šidák corrections are examples of approaches for countering this problem [5], [22], at the cost of increasing the probability of incorrectly considering that the difference between two survival functions is insignificant.

## III. USING SURVIVAL FUNCTIONS FOR EMPIRICALLY STUDYING SOFTWARE EVOLUTION

In empirical studies on software evolution, many research questions are related to the time-dependent nature of the systems being analysed. Many kinds of entities (for instance: files, classes, methods, but also developers, bug report tickets, and mailing list threads) can all be seen as potential subjects

<sup>1</sup>Data source: [http://www.ats.ucla.edu/stat/r/examples/asa/asa\\_ch2\\_r.htm](http://www.ats.ucla.edu/stat/r/examples/asa/asa_ch2_r.htm)

of a survival analysis for which the entire history cannot be known at the time the study is carried out. Survival functions should therefore be used to accurately represent time-dependent properties and to correctly take into account the censored observations.

In order to illustrate the opportunities offered by survival analysis, this section briefly presents empirical studies on software engineering in which such a survival analysis has been carried out to answer some time related research questions.

Samoladas et al. [21] built a predictive model to determine whether the likelihood of projects being discontinued in the future. The model was based on the survival function of project durations and other project properties, including metrics describing the involvement of stakeholders in the projects. The results of their empirical study support the relevance of this approach for predicting the evolution of software systems.

Scanniello [18] used the Kaplan-Meier estimator for analysing dead code in five open source Java software systems. In this study, the considered event was the appearance of dead code blocks in methods, while removed methods were considered as right censoring. For two of the studied projects, the survival functions remain very high during the entire observation period. For the other projects, a quick decrease of the survival functions is observed. While no general conclusion can be drawn from this small scale study, the authors are convinced of the relevance of the Kaplan-Meier estimator as a tool for analysing the progressive introduction of dead code in software projects.

Survival analysis models have been used by Wedel et al. for studying the occurrence of faults over time [23]. Survival function estimators allow to take into account the fact that some bug reports are not closed before the end of the observation period. The authors also discuss the automatic pre-processing that must be applied to the used data sources in order to carry out a large scale statistical analysis of fault occurrence in software systems.

Claes et al. [6] have carried out a survival analysis of packages in the Debian open source Linux distribution. As potential factors influencing a package's longevity they considered the presence and absence of strong conflicts in these packages, as well as the time at which these conflicts appear. They also studied the time needed for the conflicts to disappear. Among other results, it turns out that strong conflicts almost never get removed. It also appears that the longer a package has survived without strong conflicts, the less likely it becomes that strong conflicts will appear.

In [10], we analysed the Kaplan-Meier estimates of technologies on which open source Java software systems rely to manage access to a relational database. Contrary to our initial intuition, a comparative study of the survival function of a technology after another one has been introduced in the same project did not reveal any significant difference in survival. We found no evidence that the introduction of a second technology in a project influences the survival of a first technology.

## IV. CONCLUSION

Survival analysis is extensively used in medical research for studying time-dependent properties. Different approaches exist for taking into account censored data and for studying an unknown distribution of event occurrences over time. These approaches have been successfully applied to empirical studies on software evolution for estimating the quality (in the broadest sense of the word) of a software system or for determining the factors influencing this quality.

Because of the relative simplicity of the presented survival tools and their advantages over more traditional metrics such as the mean time to first failure, it is likely that these tools will become part of the standard toolkit for software analysts in the near future.

## REFERENCES

- [1] Ola W. Barnett, Cindy L. Miller-Perrin, and Robin D. Perrin. *Family violence across the lifespan: an introduction*. SAGE Publications, 2011.
- [2] Holly P. Berty, Haiwen Shi, and James Lyons-Weiler. Determining the statistical significance of survivorship prediction models. *Journal of Evaluation in Clinical Practice*, 16(1):155–165, 2010.
- [3] J Martin Bland and Douglas G Altman. The logrank test. *328(7447):1073*, 2004.
- [4] Elfriede Bollschweiler. Benefits and limitations of kaplan-meier calculations of survival chance in cancer surgery. *Langenbeck's Archives of Surgery*, 388(4):239–244, 2003.
- [5] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.
- [6] Maelick Claes, Tom Mens, Roberto Di Cosmo, and Jérôme Vouillon. A historical analysis of debian package incompatibilities. In *Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working Conference on*, pages 212–223. IEEE, 2015.
- [7] S. A. Cooper and M. Voelker. Evaluation of onset of pain relief from micronized aspirin in a dental pain model. *Inflammopharmacology*, 20(4):233–242, 2012.
- [8] Irene Eusgeld, Falk Fraikin, Matthias Rohr, Felix Salfner, and Ute Wappler. Software reliability. In Irene Eusgeld, FelixC. Freiling, and Ralf Reussner, editors, *Dependability Metrics*, volume 4909 of *Lecture Notes in Computer Science*, pages 104–125. Springer Berlin Heidelberg, 2008.
- [9] Joel Garner, Jeffrey Fagan, and Christopher Maxwell. Published findings from the spouse assault replication program: A critical review. *Journal of Quantitative Criminology*, 11(1):3–28, 1995.
- [10] Mathieu Goeminne and Tom Mens. Towards a survival analysis of database framework usage in java projects. In *Proc. ICSME*, pages 551–555. IEEE, 2015.
- [11] M. Greenwood. The natural duration of cancer. *Reports on Public Health and Medical Subjects*, 33(1-26):1–26, 1926.
- [12] Joseph P. Iannotti, Allen Deutsch, Andrew Green, Sally Rudicel, Jared Christensen, Shannon Marraffino, and Scott Rodeo. Time to failure after rotator cuff repair. *The Journal of Bone & Joint Surgery*, 95(11):965–971, 2013.
- [13] Faruk Incecik, Sakir Altunbasak, and Ozlem M. Herguner. First-drug treatment failures in children with typical absence epilepsy. *Brain and Development*, 37(3):311 – 314, 2015.
- [14] E. L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):pp. 457–481, 1958.
- [15] Xiaolin Liang and Zunguo Hu. Reliability analysis of repairable system with arbitrary structure. In *Multimedia Technology (ICMT), 2011 International Conference on*, pages 5977–5980, July 2011.
- [16] N. Mantel. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer chemotherapy reports. Part 1*, 50(3):163–170, March 1966.
- [17] V. N. A Naikan. *Reliability Engineering and Life Testing*. PHI Learning Pvt. Ltd., 2008.

- [18] G. Scanniello. Source code survival with the kaplan meier estimator. In *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*, pages 524–527, Sept 2011.
- [19] Marcy L. Schwartz, Kimberlee Gauvreau, and Tal Geva. Predictors of outcome of biventricular repair in infants with multiple left heart obstructive lesions. *Circulation*, 104(6):682–687, 2001.
- [20] Nozer D. Singpurwalla and Simon P. Wilson. *Statistical Methods in Software Engineering: Reliability and Risk*. Springer Series in Statistics. Springer, December 1999.
- [21] Diomidis Spinellis, Georgios Gousios, Vassilios Karakoidas, Panagiotis Louridas, Paul J. Adams, Ioannis Samoladas, and Ioannis Stamelos. Evaluating the quality of open source software. *Electron. Notes Theor. Comput. Sci.*, 233:5–28, 2009.
- [22] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):pp. 626–633, 1967.
- [23] Michael Wedel, Uwe Jensen, and Peter Göhner. Mining software code repositories and bug databases using survival analysis models. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '08*, pages 282–284, New York, NY, USA, 2008. ACM.
- [24] Yue Zhang and MartinL. Puterman. Analytical long-term care capacity planning. In Gregory S. Zaric, editor, *Operations Research and Health Care Policy*, volume 190 of *International Series in Operations Research & Management Science*, pages 39–70. Springer New York, 2013.