Towards Base Rates in Software Analytics

Magiel Bruntink
SNE Research Group, University of Amsterdam
m.bruntink@uva.nl

Abstract

Nowadays a vast and growing body of open source software (OSS) project data is publicly available on the internet. Despite this public body of project data, the field of software analytics has not yet settled on a solid quantitative base for basic properties such as code size, growth, team size, activity, and project failure. What is missing is a quantification of the base rates of such properties, where other fields (such as medicine) commonly rely on base rates for decision making and the evaluation of experimental results. The lack of base rates impairs both research activities in the field of software analytics and decision making on software projects in general. This poster contributes initial results of our research towards obtaining base rates using the data available at Ohloh (now BlackDuck OpenHub), a large-scale index and analytics platform for OSS projects.

Further reading

[1] Black Duck Software, Inc. Ohloh, 2013.

[2] Magiel Bruntink. OhlohAnalytics data set and analysis tools, 2013.

[3] Magiel Bruntink. Towards base rates in software analytics: Early results and challenges from studying Ohloh. *Science of Computer Programming*, November 2013.

[4] Magiel Bruntink. An Initial Quality Analysis of the Ohloh Software Evolution Data. *Electronic Communications of the EASST*, 65, February 2014.

What is a base rate?

A base rate[a] is a statistical concept commonly used, for instance, in the field of medicine to indicate the prevalence of a phenomenon (e.g., a disease) in the population-at-large. The base rate, also referred to as *prior probability*, is the measure used when no categorical evidence (e.g., medical history, smoking or non-smoking, sex) is available.
Bases rates are needed to judge experimental results. An example within software engineering is a new methodology that indicates (with 100% recall and 70% precision) whether a project will fail. Without knowing the base rate of project failure in the population one would likely suffer a base rate fallacy by thinking the failure chance is 70% given just a positive test result. Given a project failure base rate of 20%, the chance of a project failing given a positive test would be only 45%.

[a] Illustrated further on Wikipedia by Keith Devlin: `http://en.wikipedia.org/wiki/Base_rate`

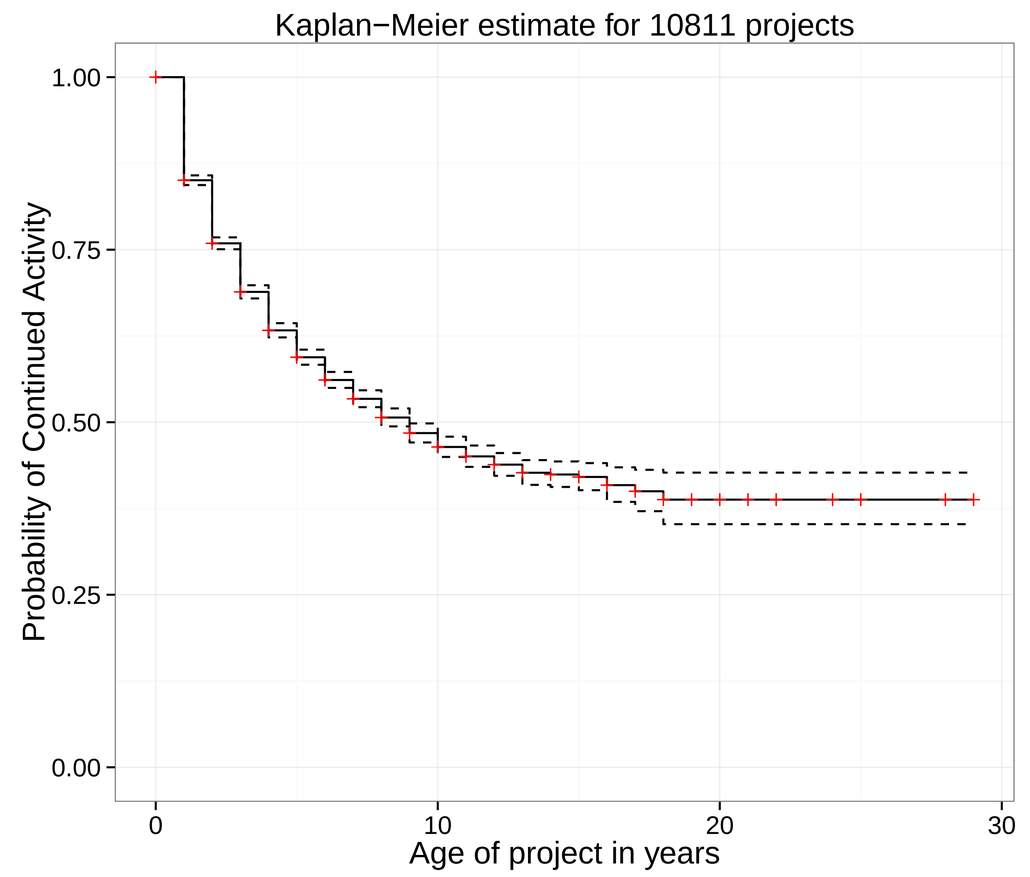
Research design: Goal-Question-Metric

Goal	**Purpose**	Extend the quantitative knowledge base
	Issue	on commonly used metrics of project evolution
	Object	of OSS projects
	Viewpoint	from the viewpoint of a researcher or decision maker, in either case external to the projects under study.
Questions	**Q1**	What is the rate of OSS projects becoming inactive?
	Q2	What is the yearly code growth rate of OSS projects?
Metrics	**PCA**	**Probability of Continued Activity**: Measured by a Kaplan-Meier estimate based on (right-censored) inactivity events. A project is considered to suffer from inactivity if it has 0 commits in a year.
	CS	**Code Size**: Measured by lines of code, i.e., lines of source text excluding comments and white space, including all of a project's source text in all programming languages recognised by Ohloh.
	CGi	**Indexed Code Growth**: An index of the code size at the end of a year compared to the beginning of the year. For example, a value of 1.05 represents 5% code growth since the beginning of the year.

Data set and data quality

The data set was collected in July 2013 from Ohloh (now BlackDuck OpenHub), an open-source software index and analytics platform [1]. Generally these data consist of a monthly aggregate resulting from analysis (done by Ohloh) of the source code and the version control system(s) of an OSS project. In total, data were collected for 12,360 OSS projects, resulting in a grand total of 828,312 project months [3]. In other work we analysed the data set for quality problems and found around 15% problematic cases [4]. The results on this poster are based on a cleaned data set consisting of 10,811 projects. The data and software used in our research are available for replication at [2].

Inactivity of Open Source projects



Code growth of Open Source projects

The 10 most used main programming languages in the data set

Some challenges

- **Data quality:** The results presented on this poster are preliminary and may be subject to future changes or corrections. The first challenge of this research is increasing trust in the quality of the data. To meet this challenge we have been applying basic internal sanity checks, we are researching ways to apply statistical tools such as Gaussian processes, and finally we plan to triangulate data with other sources.
- **Beyond open source:** A big, but potentially rewarding, challenge consists of coupling our data set on OSS projects to data on industrial projects. Having base rates for both OSS and industrial software domains enables quantification of their differences.
- **Categorisation:** There is an obvious next step of applying categorisations to the data set to decrease dispersion. Essentially the question is if we can identify (plausible) subsets of projects for which the base rates are more precise.