Investigating the Impact of Methodological Choices on Source Code Maintenance Analyses

by

Syed Ishtiaque Ahmad

BSc., North South University, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

September 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Investigating the Impact of Methodological Choices on Source Code Maintenance Analyses

submitted by **Syed Ishtiaque Ahmad** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Reid Holmes, Associate Professor, Computer Science, UBC
Supervisor

Elisa Baniassad, Associate Professor, Computer Science, UBC
Supervisory Committee Member

Abstract

Many prediction models rely on past data about how a system evolves to learn and anticipate the number of changes and bugs it will have in the future. As a software engineer or data scientist creates these models, they need to make several methodological choices such as deciding on size measurements, whether size should be controlled, from what time range metrics should be obtained, etc. In this work, we demonstrate how different methodological decisions can cause practitioners to reach conclusions that are significantly and meaningfully different. For example, when measuring SLOC from evolving source code of a method, one could decide to use the initial, median, average, final, or a per-change measure of method size. These decisions matter; for instance, one prior study observed better performance of code metrics for defect prediction in general, while another study found negative results when performance was evaluated through a time-based approach. Our result identifies the reason behind this contradiction is due to the age of the methods not being explicitly controlled, where the first six months of a method's evolution could have provided a better understanding of maintenance. Understanding the impact of these different methodological decisions is especially important given the increasing significance of approaches that use these large datasets for software analysis tasks. This work can impact both practitioners and researchers by helping them understand which of the methodological choices underpinning their analyses are important, and which are not; this can lead to more consistency among research studies and improved decision making for deployed analyses.

Lay Summary

Software practitioners use a large amount of data about how their systems evolve to build analyses that predict the number of changes or defects they will have in future. Given that a method can undergo many changes in its lifetime, each of which can alter its size, practitioners need to decide how to represent these multiple size values (e.g., average size, median size, initial size etc.), and select the time frame from which data should be obtained amongst several other methodological choices. These decisions are important; for instance, while one study found better performance for a set of software measures for defect prediction models overall, another observed a negative result using a time based approach. Our work provides evidence explaining some of these contradictions and helps practitioners to understand the impact of their methodological decisions on the research outcome.

Preface

This work presented henceforth was conducted at the Software Practices Laboratory of the University of British Columbia, Point Grey campus. I was the lead investigator, responsible for research idea formation, implementation, data collection, and analysis of the results. The initial draft of this work was written by me which was later revised.

Dr. Reid Holmes was the supervisory author on this project and throughout the project, was involved in concept formation and manuscript composition.

Postdoctoral Fellow Shaiful Chowdhury helped with code implementation in Chapter 4 and wrote parts of Chapter 5.

Table of Contents

List of Tables

List of Figures

Acknowledgments

First, I must thank my supervisor Dr. Reid Holmes for giving me this opportunity to collaborate with him. He is an extremely helpful, kind, and fun-loving person who could lighten up any depressing atmosphere, may it be paper rejection or negative results. His never-give-up attitude is contagious which keeps me motivated and compels me to do my best.

I would also like to thank Dr. Shaiful Chowdhury for guiding me throughout and unofficially acting as a co-supervisor. I am grateful to Dr. Elisa Baniassad for happily accepting to be the second reader for this thesis and taking the time to read and provide feedback right before her holidays.

My appreciation extends to every individual of Software Practice Lab who made depressing COVID times pleasant through online movie nights, games, and MEMES. Thank you for making Software Practice Lab feel like a home away from home.

Finally, I would like to thank my family for their unconditional support and words of encouragement during difficult COVID times. I am forever indebted to them and my best friend, Shareen, who always believed in me!

Chapter 1

Introduction

Software maintenance is a challenging [7] and costly task [75] and researchers are investigating ways to reduce this cost by building predictive models [6, 20, 26, 36, 54, 92, 97] for identifying change [77, 89, 100] and defect proneness [46, 84, 85]. Researchers have investigated static code metrics (e.g., McCabe [52], C&K [16]) process metrics (e.g., code churn [59], change burstiness [60], number of developers [93]), and combinations of metrics [50, 81] as these are often associated with defect and change proneness. Unfortunately, over the last decade there have been disagreements among research communities related to the true effectiveness of these metrics [28, 29, 79].

While some studies argue that code metrics are better for predicting pre-release defects [25, 99], others claim metrics perform well for post-release [78] as well. Different studies have also had disagreements at different granularities (e.g., module-level [63, 66, 74, 99] vs. method-level [64]), or using different aggregation schemes (e.g., mean vs. entropy [96]). What is the cause of these disagreements? Is it impacted by different ways a metric can be measured (e.g., SLOC measurement), or does it depend on other factors such as the time frame from which metrics were obtained or the way data was aggregated across projects? The availability of a large amount of historical data poses new challenges, for example deciding what data to use for training these models [61].

In this work, we explore several of these key methodological decisions made by researchers and practitioners as they design their data analysis pipelines for

software evolution analyses. Specifically, we examine the impact of five methodological choices related to size measures, size normalization, timeframe selection, change analysis, and result aggregation schemes on investigating or deploying change and defect proneness prediction models. Our analysis reveals that these choices can lead to substantially different results and explains some of the previous contradictory findings from prior software maintenance prediction efforts.

We performed analysis of these methodological decisions at the method level granularity. This granularity was chosen because the method-level is commonly desired by the research community [27, 64] and industrial developers [33]. To do this, we extracted the complete method level history of 1,641,353 Java methods from 53 open-source projects. We then examined the impact on change-proneness and defect-proneness from the five different methodological decisions, using the most commonly-used variations of these decisions from prior studies.

The thesis of this work is that differing methodological choices can impact analysis outcomes that would be derived for common software evolution analyses; these differences are important because they explain some of the contradictions reported from prior work. To do this, we first give an overview of the problem space related to change- and defect prediction models (Chapter 2) and define our research questions. We then discuss the contradictions found in prior work (Chapter 3) and describe our approach for selecting projects and collecting method-level historical data (Chapter 4). We also describe our code metrics and the statistical tests used to investigate the impact of methodological choices (Chapter 4). In Chapter 5 and Chapter 6, we report our results and provide guidelines for future studies and discuss the threats to validity of our work. Finally, in Chapter 7 we conclude the thesis.

The main contribution of this research is to show and explain many contradictions in prior work so that future deployments of prediction models can be better designed and interpreted. Our results are particularly important for researchers and practitioners leveraging the vast trove of historical data available from past system evolution, such as ML-based approaches, so they can build their models using high-quality data to make accurate and meaningful predictions.

Chapter 2

Predicting Software Evolution

Given the expense associated with software maintenance, engineers and researchers have long sought to predict future software evolution trends in a bid to reduce the number of future defects and decrease future maintenance costs. These predictions typically use historical data, either about the system under analysis or other prior systems, to identify code units or changes that could be prone to future changes. Future changes are typically used as a proxy for both the expensive tasks of fixing defects and evolving existing source code. Two of the most common kinds of software evolution predictions are change-proneness and defect-proneness; these models share the same common methodological steps, as would any analysis that takes past historical project data as input.

Given a desired analysis unit granularity (usually a subsystem, file, or method) from one or more projects, practitioners wishing to perform change- or defect-proneness predictions generally need to make the following five methodological choices, each of which we investigate with a research question. In this work, we fix our granularity unit to the method level, for a total of 1,641,353 methods from the historical evolution of 53 projects, and use SLOC (source lines of code) as the selected code metric as previous work has shown that SLOC is highly correlated with other metrics [22, 29, 83]. The hypothesis is that, the observations we make with SLOC would be similar with other code metrics.

1. **Size measurement.** When working with change and defect prediction models, some practitioners train models using single value of SLOC such as a

method's introduction or last updated SLOC, whereas others opt to use mean SLOC to capture the overall SLOC change of methods. While all these are rational decisions one could take, would considering a single value and an average value lead to an unstable conclusion where models and maintenance studies relying on one of these measures might contradict each other? If not, are these measurements in general truly appropriate to use for understanding code maintenance (RQ1)?

2. **Size normalization.** While it is broadly understood that larger methods are more change prone, it is not clear if they undergo disproportionately more evolution than their size alone would suggest. Should analyses normalize size to avoid simply recommending that large methods are evolution-prone and small methods are not (RQ2)?

3. **Timeframe selection.** From prior work, we know that different chunks of data within one project could lead to different prediction models [55]. Nowadays, just-in-time models have gained traction where defect prediction training is done by using recent data. Although these models' performances vary depending on the time frame for which metrics are obtained, they do not differentiate between the impact of older and younger methods that can introduce biases. Therefore, should we normalize the age of methods and consider the entire history or are there other time ranges that could be used for training models or understanding maintenance where SLOC is strongly associated with the maintenance indicators (RQ3)?

4. **Change analysis.** It is intuitive that some changes will have greater evolutionary impact than others. For example, changing a code comment is less likely to directly introduce a future defect than modifying a key algorithm. Should the nature of the changes made to a code unit be considered when they are built into an evolutionary model (RQ4)?

5. **Result aggregation.** Given the breadth of past projects and their historical depth, it can be tempting to make future predictions in a given project by examining results from past successful (or unsuccessful) projects. This can also help with the *cold start* problem when a new project has little or no

historical data. But how well do predictions made form other projects generalize, or are projects and their histories unique enough that multi-project aggregation is detrimental (RQ5)?

While the options for each of these methodological choices are relatively straightforward and the selection of a given choice can be intuitively argued, these decisions can have significant impact on the predictions argued from a software evolutionary model. In this work we examine the impact these decisions can have in an effort to explain the contradictions from prior work while helping future researchers and practitioners design their analysis pipelines.

Chapter 3

Related Work

A variety of inconsistencies has arisen in prior work, driven at least partially by the five methodological decisions examined in this thesis.

3.1 Metrics for Evolution

Previous work examined several code metrics to predict defect and change proneness. However, the researchers could not establish a common ground on the usefulness of code metrics [28, 29, 79]. Some studies found favourable outcomes for code metrics [4, 10, 41, 48, 85], while others discovered negative results [64, 73, 79]. Some argue that process metrics [14, 58] perform better than static code metrics [2, 78] for fault prediction in highly iterative post-release software systems. In contrast, others found that the combination of process and static code metrics perform well in terms of accuracy [17]. Despite the continuous debate about code metrics, most of the studies agree that size is the effective code metric to estimate software maintenance [22, 29, 83].

3.2 Measurement Selection

When performing coarse-grained analyses (e.g., file, module, or system), researchers and practitioners are forced to select an aggregation scheme that combines the multiple method-level metrics to coarse-grained level. One of the most popular choices among the research community is to use the average value of a met-

ric [1, 64, 97, 98]. Zhang et al. [97] built a universal cross-project defect prediction model and aggregated method-level data to file level using max, average, and summation scheme and clustered projects based on the similar distributions of 6 metrics (e.g., lines of code, number of files, number of commits, number of developers, etc.). Similarly, in another work, Zhang and Hassan [98] experimented with different aggregation techniques of software metrics and observed their impact on defect prediction models. They also aggregated method-level data to file level and examined 11 aggregation schemes such as sum, mean, median, standard deviation, Gini [31], Hoover [37], Atkinson [8], Shannon entropy [76], etc., and compared the correlation of these aggregated metrics with SLOC. While these aggregation schemes are some rational choices a researcher could make, it can interfere with the correlation among metrics [40, 47, 88] and produce inconsistent results. Vasilescu et al. [88] found that values aggregated using mean from method level data to package level leads to inconsistent correlation results of SLOC with defects for some projects. As these aggregation schemes can also be applied to method [64] having different SLOC values at each revision, contradictory results could also be based on which schemes were used. In our work, we investigate how different SLOC measurements lead to inaccuracy in correlation values for change and defect-proneness studies and provide a guideline for mapping SLOC to the number of changes and bugs.

3.3 SLOC Control

Most practitioners agree that a file with more lines of code or bug fixes is more bug prone [90]. Size as a metric has been studied extensively in the literature at various granularity (e.g., class [1], system [44], packages [3, 25, 99]) and in different contexts. For example, prior work studied the relationship of size such as a large number of test methods per class with the density of test smells [32] and code coverage [38]. Although Greiler et al. [32] found that the number of test methods per class is related to the higher density of test smell, having higher number of test methods is not indicative of effective test suite [38].

Similarly, several works [22, 29, 101] have analyzed the impact of class size on defect-proneness prediction models. Kore et al. [45] applied a tree-based clas-

sification technique to detect change prone-class and observed that large classes tend to change more than smaller classes. Although the result obtained from these models are insightful, these results are not transferable to other count-based defect prediction models that rely on the number of defects [99]. A possible cause of this inconsistency could be that the number of bugs in such a system is not being normalized by SLOC appropriately.

Lefever et al. [49] found that after normalizing size, there are substantial inconsistencies in the software quality reporting tools (e.g., SonarQube, DV8, Structure 101) when identifying the most problematic files. Although, these disagreements are perceived at a higher level of granularity (e.g., system level [44], classes [1], files [91] or packages [3, 25, 99]), they also exist at method-level. Pascarella et al. [64] investigated bug prediction at method-level and found lower performance (up to 20 points percentage less) in terms of AUC-ROC for bug prediction.

In our work, we used change and bug density as defined by prior work [32, 59, 63, 72] to normalized the change- and bug-proneness indicator which is the dependent variable. In contrary, prior work have mostly used SLOC to normalized other metrics that are used as independent variable to understand software maintenance and quality [9, 29, 59]. After neutralizing the effect of SLOC by calculating the density we compared the distribution of maintenance indicator density in small, medium and large methods to gain more insight on change- and bug-prone methods.

3.4 Age Control

Many studies have used time-based sampling techniques to understand software evolution or to build defect prediction models at coarse granularity (e.g., system level [44], classes [1], file, packages [3, 25, 99]). For example, at the system level with a time interval of 3 months, Osman et al. [62] compared the evolution of different types of exception handing in code (e.g., standard exceptions, custom exceptions) between the application and third-party libraries. Additionally, for defect prediction, time based sampling such as 3 months or 6 months [53] and all data (e.g., [44, 91], [1]) have been explored.

In addition, Zhang et al. [99] examined the pre-and post-release ranking abil-

ity of SLOC at the package level and found that 20% of the largest modules were responsible for the majority of the defects (51-63%). Fenton and Ohlsson [25] observed a similar connection between size metric (SLOC) and the number of faults but in pre-release data. Although these coarse-grained analyses are a popular choice among the research community, the inconsistent results of these studies have been reported as well. Andersson and Runeson [3] replicated the study of [25, 99] and did not observe any ranking ability of SLOC for some projects. Moreover, a negative result was observed in the method-level study [64] at more realistic settings (such as release-by-release validation models) that contradict the findings of previous studies [27].

Menzies et al. [56] studied defect prediction results from 28 recent studies and reported that findings from these studies contradict each other about what influences software defects. They found that for different chunks of data, a completely different model [55] is learned even from the same project. One possible cause for inconsistent conclusion reported in previous studies could be accounted for the age of methods not being explicitly controlled. It is inaccurate to compare younger with older methods as older methods had more chances to change. Therefore, in this work, we provide insights on large evolutionary datasets at the method level to practitioners through age normalization.

3.5 Change Classification

Several studies have used the number of revisions as a maintenance indicator [4, 5, 82] by accounting for the type of transformation applied to a given file or method. For example, Koru et al. [45] evaluated change count by looking at the revision history to identify most change-prone classes. Additionally, Farago et al. [24] observed that files which are committed at a higher frequency (i.e., with longer revisions history) have a negative impact on maintainability than the files with lower commit frequency. As a result, software quality reporting tools depend on the number of revisions or commit frequency as a measure of change hotspot [11, 15, 49]. Shrikanth et al. [61] observed that the majority of the defects are localized in the early 50 commits of a project. However, commit patterns are dependent on developers' coding styles, expertise [34, 51], and experience. Kawrykow and Ro-

billard [43] analyzed the type of transformations applied to methods and found that a large fraction (15.5%) of method changes can be classified as non-essential. They reported that most changes involve removing or adding *this* keyword or are induced by rename refactoring. Additionally, Ray et al. [68] observed that most changes introduced by developers in multiple revisions are non-unique (i.e., follows a specific repetitive pattern). Therefore, accounting for the kind of changes applied to a method can produce inconsistent findings for code metric correlation. In this work, we adopt the differentiation technique of essential and non-essential changes by Kawrykow and Robillard [43] to demonstrate the impact of different types of changes at the method level.

3.6 Project Aggregation

Practitioners working with code metrics to understand software maintenance need to decide whether to analyze projects separately [42, 60, 68, 82] or combine them to obtain a single value for the metrics under analysis [29, 64, 85]. Pascarella et al. [64] aggregated 13 projects bug data together to build a single defect prediction model although the number of bugs in each system under analysis varied substantially. Similarly, Gil and Lalouche [29] combined 26 projects to evaluate a single value of SLOC and other metrics (e.g., cyclomatic complexity) and investigated the confounding effect of SLOC on them. Spadini et al. [85] combined 10 open source projects to study the change- and defect proneness of test code. While aggregated metrics of several projects to obtain a single value is useful, it obscures the effect of outlier projects having a highly skewed distribution of metrics [95]. These outlier projects are the result of software evolution that are impacted by several external factors such as developer expertise, commit pattern [34, 51], time, and budget. Therefore, one might opt for individual analysis for these reasons. Nagappan et al. [60] studied the ability of change burst to predict the number of defects and was able to achieve high accuracy of 90% for windows and 71% for eclipse projects by studying them separately. Shin et al. [82] analyzed the Mozilla Firefox web browser and Red Hat Enterprise Linux Kernel separately and investigated the ability of code complexity, churn and developer activities to discriminate between neutral and vulnerable files and predict vulnerabilities. Although indi-

vidual project analysis overcomes the problem of aggregated project analysis, it suffers from selection or publication bias [29]. As there is a lack of randomization in selecting these projects, this often results in missing out on projects with unique characteristics.

In this work, we show the impact of project aggregation and individual analysis by studying the relationship of SLOC with number of changes and bugs in methods of large number diverse projects.

Chapter 4

Methodology

To accurately examine the impact of design alternatives for the five methodological decision points, we collected a corpora of method-level historical data to facilitate a deeper analysis of how those methods evolved and what change-proneness and bug-proneness approaches would have predicted for them.

4.1 Project Selection

Over the last decade, GitHub has gained popularity as a rich source of open-source projects. In our work, we used the SEART GitHub Search Tool [19] to identify candidate systems for analysis. Our search criteria included actively maintained Java projects (excluding forks), that have $\geq$ 1,000 commits, $\geq$ 200 stars, and $\geq$ 30 contributors. We only selected Java projects as the number of defects and metrics values varies according to the programming language used [12, 67].

We sorted the resulting system by repository file size and manually filtered out systems less than 2,000 kilobytes to try to remove toy projects from the 1,204 Java projects. Finally, we applied purposive sampling [23] to select a wide range of projects based on stars (as the number of stars is considered a proxy for popularity [13]), age, and the number of commits. The final set of 53 projects had a median age of 11 years and a median of 35, 207 changes. We believe this final set of projects have undergone sufficient historical changes to observe temporal effects. Furthermore, the selected projects had a large number of contributors (me-

dian 157) to capture different developers' coding styles. Since developers can have individual coding styles [39], it may be possible that some developers introduce more non-unique changes than others.

The resulting 53 projects is a larger sample size than related method-level studies, which have used 13 [64], 21 [27] and 20 [33] projects. Additionally, subsets of our selected projects have also been used in previous studies [65, 85, 94]. The selected projects serve a diverse set of domains, including code analysis (checkstyle), mocking libraries (mockito), HTTP clients (okhttp), and JSON parsing (fastjson, gson). Table 4.1 shows the summary of 12 projects (top 6 and bottom 6) sorted by the number of methods in the project.[1] The final 53 projects have 1,641,353 methods (with an average of 30,162 methods per project) and a total of 653 years of evolutionary information.

[1]The full project list and statistics are available in Appendix A.

Table 4.1: Project details for top 6 and bottom 6 (out of 53 open-source projects) Java projects sorted by number of methods (*#Methods*). *TtlSLOC* is the total source line of Java code at the reference commit. The reference commit (*Ref. Com*) is the latest commit of each project from which history was backtracked. *TtlRevisions* is the total number of changes applied to methods in each project.

Repo	Ref. Com	Stars	Contr.	#Methods	#Files	TtlSLOC	TtlRevisions	Years
sonarqube	39abd3de	5,937	137	130,333	7,293	517,951	249,504	11
flink	6a8b0301	16,764	908	118,815	11,282	1,259,298	227,872	10
wildfly	a9e06167	2,593	343	112,098	10,081	616,780	153,071	11
hibernate-orm	cfc7b972	4,678	420	95,135	10,260	779,669	179,706	14
spring-framework	42061d27	43,641	544	90,138	7,512	696,598	243,270	12
docx4j	26aaa13f	1,620	37	68,568	3,897	321,657	96,542	13
truth	60dcd093	2,296	83	4,609	185	34,461	13,154	10
commons-io	29b70e15	787	76	4,207	360	39,836	16,667	19
rabbitmq-java-client	cc7c8677	996	52	4,161	367	32,052	10,092	13
hector	a302e68c	648	71	3,682	459	31,365	6,123	11
gson	ceae88bd	19,809	109	3,633	207	25,280	5,781	12
spark	d5e7bd0c	30,409	1,694	2,089	986	79,958	3,224	11
Total		518,433	14,793	1,598,592	122,535	10,846,053	3,171,244	653
Mean		9,782	279	30,162	2,312	204,643	59,835	12
Median		3,482	157	16,796	1,238	126,645	35,207	11
Min		303	34	2,089	158	15,259	3,224	4
Max		56,378	2,821	130,333	11,282	1,259,298	249,504	24

4.2 Data Collection

We used *CodeShovel* [33] to extract the complete set of commits where the method was modified. CodeShovel does this by walking backward through a project's commit history starting from the reference commit listed in Table 4.1. It can detect cross-file changes (file move and file rename), method signature changes (method rename, return type change, exception changes, parameter changes, parameter type changes), metadata changes (documentation and annotation changes), and method body changes. CodeShovel has been shown to accurately uncover 97% of all method changes. CodeShovel also collects meta-information about each commit such as commit message, authors name, `commit SHA`, and commit date. For each project, we extracted history of all methods (including methods that were later removed) by iterating backwards from the reference commit on the repository's main branch.

After using *CodeShovel* to extract the complete history for the 1,641,353 methods in our 53 project corpus, we computed key metrics for each method at each revision using a tool that we have implemented for this purpose. The tool uses the `javaparser`[2] library and parses the source code of each method at each changed commit to compute required metrics. Two of the authors have independently verified the correctness of the tool by randomly selecting and validating 200 Java methods. We represented the change in metric value at each commit for a given method as an array of metric values.

4.3 Maintenance Indicator Selection

This evaluation uses change-proneness and defect-proneness as exemplar analyses against which different methodological decisions can be evaluated.

4.3.1 Change-proneness

The number of revisions to a code unit has been used widely by the research community as an indication of maintenance effort [4, 5, 82]. A common software design idiom is that "code should be open to extension, but closed to modification".

[2] https://github.com/javaparser/javaparser

In terms of methods, this means that well-designed methods should not need many revisions, as a given method should mainly need to be changed to fix defects, not add new features. To measure the change-proneness of the method we count the ***#Revisions***, the total number of times a method was changed regardless of the type of modification applied.

4.3.2 Bug-proneness

We counted bug-proneness (***#Bugs***) by counting the number of bug-fix commits in the project history. Previous studies have used different sets of bug-related keywords to identify bug fixes. Zhang and Hassan used (*'bug'* ,*'fix'*, *'error'*, *'issue'*, *'crash'*, *'problem'*, *'fail'*, *'defect'*, and *'patch'*) [98] while Ray et al. used (*'error'*, *'bug'*, *'fix'*, *'issue'*, *'mistake'*, *'incorrect'*, *'fault'*, *'defect'*, *'flaw'*, and *'type'*) [69] to detect bug fixing commits. We considered the subset from union of all keywords used in earlier work (*error, bug, fixes, fixing, fixed, mistake, incorrect, fault, defect,* and *flaw*) [57, 69, 96]. We have excluded the *'issue'* keyword, because our manual inspection suggested that it produces too many false positives. We searched for exact matches of these keywords from the commit message, partial matches were not considered.

4.4 Statistical Tests

Throughout this analysis, we use a consistent set of statistical tests.

4.4.1 Correlation Analysis

Correlation analysis is usually the first step for building a defect prediction model [59, 98, 102, 103]. These analyses measure the strength of association and the direction of the relationship between two variables. If the two variables move in the same direction, those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation. The strength of correlation ranges from -1 to $+1$, where -1 indicates strong negative and $+1$ indicates a strong positive correlation. A value of 0 indicates the two variables are completely independent [98, 102].

Correlation analysis is important as it helps to select predictors to build models [29, 40, 98]. A strong association between the outcome and the independent variable implies the variable is a good candidate for defect prediction models [98]. Furthermore, a strong correlation between multiple predictors can help researchers remove unnecessary predictors to reduce model complexity.

There are several types of correlation coefficient statistics, including *Pearson r*, *Kendall* τ, and *Spearman* ρ [80]. The selection of a particular analysis depends on the distribution of data and the presence of outliers. *Pearson r* is a parametric test and assumes the data are normally distributed, but it is not robust to outliers [80]. *Spearman* ρ and *Kendall* τ are non-parametric tests that work with rank-ordered variables [80]. Although *Spearman* ρ is widely adopted by the research community [40, 48, 98, 99] due to its robustness to outliers and compatibility to highly skewed data compared to *Pearson's r*, it is more sensitive to errors and discrepancies in data than *Kendall* τ [30, 38]. For this reason we used *Kendall* τ.

4.4.2 Hypothesis testing and effect size

We used two-sided Mann-Whitney U [80] with a confidence interval of 95 percent (i.e., α= 0.05) to identify if there is any significant difference between correlation values obtained using various methodologies. This is a non-parametric test and does not assume the data is normally distributed. Our analyses use the p-value to either accept (p-value ≥ 0.05) or reject the null hypothesis (p-value ≤ 0.05).

To quantify the size of the difference in the correlation values, we use Cliff's δ [71]. For effect sizes, Cliff's δ (a non-parametric test) represents the degree of overlap between two samples. This is more accurate and robust than Cohen's d [18]. The δ value ranges between -1 and $+1$ where a negative value implies that second sample values are greater than first sample and positive value indicates the opposite. We used prior work mapping of Cliff's δ value to label the degree of effect size [35, 98]:
negligible: $(0 \leq |\delta| < 0.147)$ *small:* $(0.147 \leq |\delta| < 0.330)$
medium: $(0.330 \leq |\delta| < 0.474)$ *large:* $(0.474 \leq |\delta| \leq 1)$

Chapter 5

Methodological Decision Impact

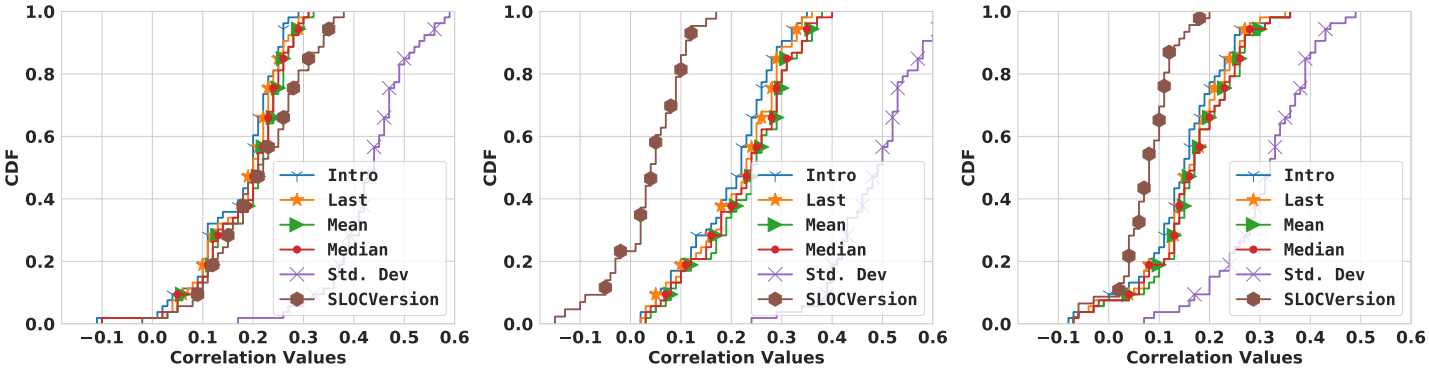
In this chapter we detail the resulting impact of reasonable and commonly used alternatives for the five methodological decision points researchers and practitioners must reason about when deploying analyses based on historical software evolution data.

5.1 RQ1: Does the SLOC selection methodology impact evolutionary analyses?

A method can have different SLOC values while it evolves. Consider a method that was revised 5 times with SLOC values 10 (r0), 20 (r1), 10 (r2), 20 (r3), and 50 (r4). Here the value of the dependent variable is 4 (i.e., #Revisions after the method's introduction r0), but what would be the value of the independent variable? Should the SLOC be the introduction value (10), the most recent (50) or the mean (22)? Although using the mean is a popular choice among the research community [1, 64, 97, 98], it is sensitive to skewness [21]. Zhang and Hassan also investigated dispersion, such as standard deviation, as an aggregation scheme for method-level to file-level data to determine its impact on defect prediction models [98].

While all these can be valid decisions for training change or bug prediction models, these SLOC measurements do not represent the actual number of changes or bugs triggered by a particular SLOC value. Therefore, we also consider a versioning technique where the above method will have three versions that map each

SLOC responsible for inducing a change or bug-fix commit. With the versioning technique, the method in the above example has the following SLOC values: SLOC 10, SLOC 20, and SLOC 50. SLOC 10 maps to r1 and r3, SLOC 20 maps to r2 and r4; SLOC 50 will not mapped until it is involved in a subsequent change.



(a) SLOC with #Revisions (all) **(b)** SLOC with #Revisions (modified once) **(c)** SLOC with #Bug (modified once)

Figure 5.1: Distribution of correlation coefficients between different SLOC measurements with number of revisions and bugs for all 53 projects. For graph readability, we marked after every 5 points only.

Figure 5.1a shows the cumulative distribution function (CDF) of the correlation coefficients between SLOC and #Revisions for all 53 projects[1]. Except for standard deviation and versioned SLOC, all other measurement techniques exhibit similar performance (Cliff's δ effect size is negligible in most cases)[2]. This is not surprising because a large number of the methods (41%) were never modified after their introduction, producing identical values for the introduction, last, mean, and median. Our finding about the large number of unmodified methods complements earlier studies [70, 86]. However, when considering versioned SLOC we observe that for $\sim$50% of the projects the correlation values are significantly higher compared to the other measurement techniques (except standard deviation). As the versioning technique maps SLOC accurately, the values are being underestimated by the other measurement techniques. Examining SLOC versioning, we observe that the strength of correlation between SLOC and maintenance is not as strong as has been previously claimed (e.g., [29, 42, 47, 79]). While standard deviation shows a much better performance, this is unsurprising since a method with few revisions has little opportunity for a high standard deviation.[3]

However, when we reevaluated the correlation values by considering methods that were revised at least once (Figure 5.1b), the correlation values of different SLOC measures with #Revisions are significantly higher than the values obtained using versioned SLOC[4]. Similarly, considering methods that change at least once, the correlation of SLOC with #Bugs (Figure 5.1c) also shows that other SLOC measures exhibit higher correlation than versioned SLOC for most projects. These lower correlations for versioned SLOC are because for methods that have undergone multiple changes, the versioned SLOC approach treats each SLOC entry independently and the total number of revisions that method underwent is lost. We will use versioned SLOC for the remainder of this thesis due to its greater correlation with #Revisions across all methods, not just those that changed at least once.

[1]The complete per-project correlation values of different SLOC measurements with #Revisions are available in Appendix B.1.

[2]Cliff's δ results are available in Appendix B.5.

[3]Such a prediction ('methods that have changed a lot are likely to change a lot') would not be especially useful or surprising.

[4]The complete per-project correlation values of modified methods are available in Appendix B.3 and B.4

RQ1 Summary

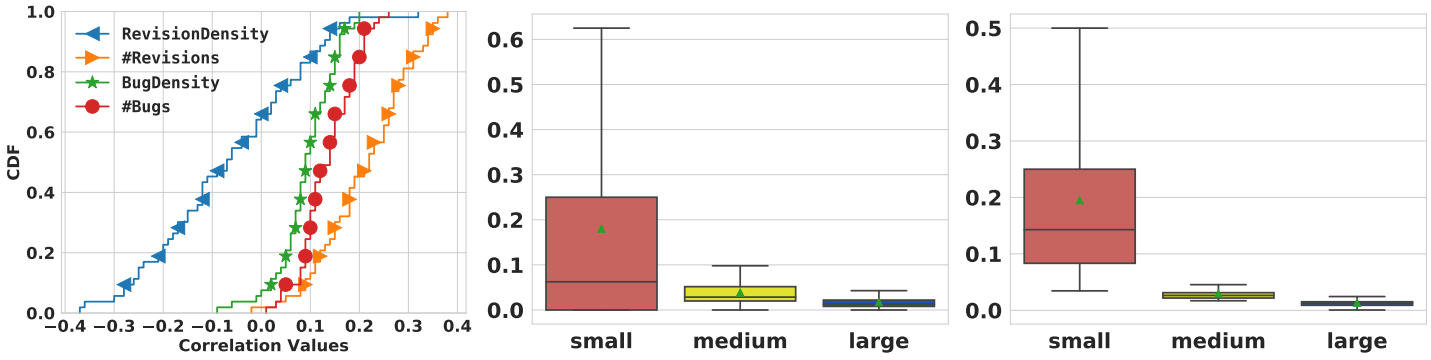
Traditional SLOC measurement techniques overestimate the strength of the correlation between SLOC and maintenance. Across all method changes, versioned SLOC outperforms more traditional measures including the the introduction, final, and mean SLOC.

5.2 RQ2: Should method size be controlled when correlating to change- or bug-proneness?

Prior studies reported that long methods [85, 87] and large packages [25, 99] are more prone to changes and bugs. This is because a positive correlation was always observed between SLOC and change- and bug-proneness. Our data suggests that this approach of analysis can be inaccurate. Consider a method with SLOC 100 that was associated with 3 revisions, and a second method with SLOC 20 and 2 revisions. A direct correlation analysis between SLOC and #Revisions will suggest that the former method is more change-prone than the later. We argue that change-proneness and bug-proneness should be analyzed using SLOC density (e.g., #Revisions/SLOC) and therefore evaluate the correlation between these.

Figure 5.2a shows that for all 53 projects, the correlations between SLOC and #Bugs/#Revisions are positive, complementing previous studies [25, 85, 87, 99]. However, when density is considered, the outcome becomes less clear. For the revision density, the correlations are negative for $\sim$70% of the projects. This clearly suggests that small methods are more change-prone if density is considered, contrary to prior studies (e.g., [25, 85, 87, 99]).[5] Bug density, does not show the same trend. For $\sim$90% of the projects the correlations are still positive, suggesting bug-proneness increases with SLOC.

[5]The complete per-project RQ2 correlations are available in Appendix C.



(a) SLOC with #Revisions/#Bugs and their densities

(b) Revisions per SLOC distribution

(c) Bugs per SLOC distribution

Figure 5.2: (a) compares the correlation distribution of #Revisions and #Bugs with their respective densities. (b) and (c) show their density distributions by grouping methods into small, medium, and large. In (c), we considered methods that have at least one bug.

We further explore the distribution of these density values by grouping methods in different SLOC categories based on Spadini et al.'s [85] work: small ($SLOC < 30$), medium ($60 > SLOC > 30$), and large ($SLOC > 60$). Figure 5.2b and 5.2c shows that methods in the small group have a higher revision and bug density than the methods in medium and larger groups. Most methods do not have a defect associated with them (83%), Figure 5.2c only displays the results for methods that associated with at least one defect.

To evaluate whether these observations are skewed by `getters` and `setters` (which are often excluded from these analyses), we reproduced the results after filtering them and observed statistically similar correlations to the populations with or without the `getter` and `setter` methods included[6].

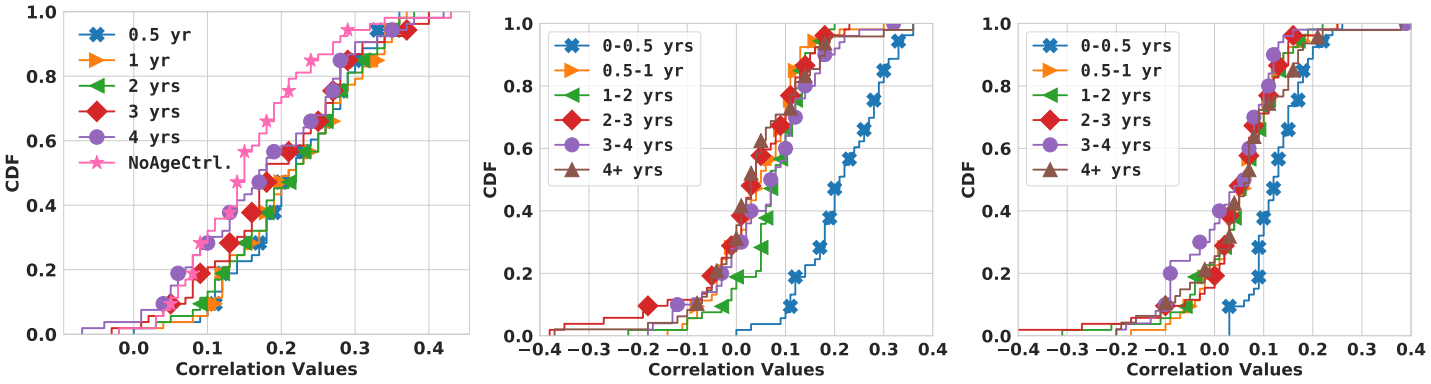
RQ2 Summary

Contrary to prior findings, small methods are more change-prone and bug-prone when method SLOC is controlled.

5.3 RQ3: Do different ages of methods and temporal choices induce inconsistent results for change/bug proneness?

Many of previous code maintenance studies either did not control method age or were not explicit about this control [3, 25, 27, 99]. We found a positive correlation between age of a method with #Bugs **(Kendall τ: 0.36)** and #Revisions **(Kendall τ: 0.73)**. These significant correlation coefficients indicate that age is an important factor that should be experimentally controlled—we should not compare a newly introduced method with a five year-old method. For age normalization, we follow two steps. In step 1, we remove all the methods that are less than x years of age. In the filtered samples, all the methods will be at least `x` years of age. Unfortunately, we may still compare an `x` year old method with an `x+i` year old method. In step 2, we remove all the revision and bug related data that occurred after `x` years of a method's life to address this.

[6]The Cliff's δ results of the difference in correlation values with or without *get* and *set* methods are available in Appendix C.3.



(a) SLOC with #Revisions **(b)** SLOC with #Revisions **(c)** SLOC with #Bugs

Figure 5.3: (a) compares different correlation coefficients of SLOC with #Revisions with no age normalization for 53 projects. (b) and (c) is the distribution of correlation coefficients at different intervals between SLOC with #Revisions and #Bugs for 53 projects. For graph readability, we marked after every 5 points only.

Figure 5.3a compares the performance of different correlation values of SLOC with #Revisions for different choices of x and no age normalization. We observe that without age control, the correlation of SLOC with #Revisions is underestimated for most projects when compared with age normalized methods. This is particularly true when the value of x is 0.5 years (Figure 5.3a). As a result, without age normalization, the interpretation of comparing correlation values is inaccurate. The performance difference between 0.5 year with no age control is statistically significant with medium effect size[7].

In contrast, correlating SLOC values with #Bugs, we did not observe any significant results when comparing the different time ranges with the no age control group[8]. For example, the performance difference between 0.5 year and no age control group is not statistically significant ($p > 0.05$) and the effect size is small.

We also evaluated if the choice of intervals, for capturing change and bug information, impacts code metric performance. For an interval between year x and year y, we removed all the methods with less than y years of age. Next, from the filtered samples, we consider changes and bugs that occur only between year x and year y. Consistent with our earlier observation, the performance of SLOC is much better (higher correlation coefficients with #Revisions) for 0-0.5 year interval than the other intervals (Figure 5.3b). Although to a lesser extent, this observation is true for bugs as well (Figure 5.3c).

As age normalization is important for accurate observations, for all the other RQs we have used two years of age normalization with which we were able to retain 88% of the methods and 65% of the change history that those method underwent. Had we considered less than two years, we would have significantly decreased our method histories potentially biasing our results.

[7]Cliff's δ results for SLOC with #Revisions at different times are available in Appendix D.1.

[8]Cliff's δ results and the figure for correlation of SLOC with #Bugs at different times are available in Appendix D.2.

RQ3 Summary

The amount of history analyzed for a method influences the predictive power of the model. The most effective models should use the first six months of method history, as this have the highest correlation of SLOC with #Revisions and #Bugs.

5.4 RQ4: Should change kind be considered when making change predictions?

Many code metric studies look at the total number of revisions without classifying the type of changes and use it as a maintenance indicator [4, 5, 82]. However, Kawrykow and Robillard found that out of all changes applied to a method, 15.5% were trivial [43].

As a result, the total number of changes performed on a method might exaggerate the number of important changes performed on it. Therefore, we further decompose the number of revisions according to the nature of the change and examine the correlation of different kinds of changes with SLOC to understand the change-proneness of a method. CodeShovel provides these raw change categories as it traverses the method history.

#BodyChanges: The number of times a method's body was modified, excluding those changes that only update formatting or whitespace.

#EssentialChanges: All changes that modify the content of a method's signature or its body. Method signature changes include modifier changes (e.g., from `public` to `private`), parameter changes (either in type or name), exception changes, parameter meta-information change (e.g., `final` keyword added), method rename, and return type changes.

#NonEssentialChanges: All changes related to formatting, annotation changes (e.g., `@test`, `@suppress`), and documentation changes. Also includes cross-file changes such as renaming the file or moving methods from one file to another. Formatting changes include both white space and indentation changes.

#Revisions: Any revision of a method, including non essential changes.

Figure 5.4 shows significantly different outcomes in correlation values can be obtained by accounting for the type of changes[9]. We found that the correlation of SLOC with #BodyChanges and #EssentialChanges are much higher compared with #NonEssentialChanges. Clearly, the performance with the #Revisions is significantly impacted because of #NonEssentialChanges.

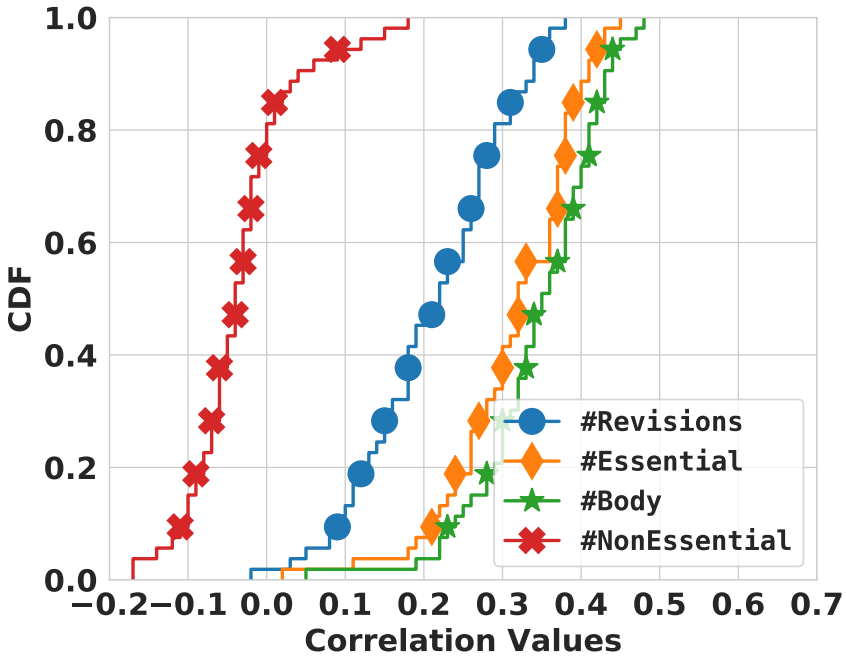


Figure 5.4: Distribution of correlation coefficients of SLOC with different types of changes for 53 projects. For graph readability, we marked after every 5 points only.

This could also impact decisions made on correlation thresholds as 50% of our projects would be excluded for low SLOC/#Revisions correlations if a correlation threshold of 0.2 was used.

[9]The complete per-project correlation values for RQ4 are available in Appendix E.

RQ4 Summary

Correlation strength for both defects and revisions changes by change kind. Non-essential changes, such as file rename, impact all types of methods similarly, and degrades predictive power.

5.5 RQ5: Do aggregate project analyses meaningfully reflect the correlation for individual project analyses?

Prior work is broadly divided into two categories for analyzing software projects related to code metrics and maintenance studies. While some studies aggregate all the code metric data from their selected projects to produce a single statistical value to understand maintenance [29, 64, 85], others opt for individual project analysis [42, 60, 68, 82]. Gil and Lalouche combined 26 projects to obtain a single value for SLOC and other metrics and compared the correlation values between them to understand the confounding effect of SLOC [29]. However, aggregating metrics by combining projects may be problematic as projects evolve differently based on several external factors (e.g., developers' commit patterns and expertise [34, 51]). Therefore, some studies preferred individual project analysis although this can be susceptible to selection or publication bias [29]. As a result, a sample of projects may not accurately represent all projects in a large population.

We explore the differences between aggregated and individual project analyses further with our dataset. Combining all projects, we found that the overall correlation coefficient for SLOC with #Revisions is 0.22 and is 0.15 for #Bugs (Figure 5.5). We observe that almost half of the projects are within $\pm$ 0.05 of these aggregated values for SLOC with #Revisions ($\sim$40%) and #Bugs ($\sim$52%). We selected 0.05 as this is $1/20^{th}$ of the max correlation value (1.0) and is small enough to capture all projects close to aggregated value. Therefore, aggregated project analysis seems to represent correlation accurately only for $\sim$50% of the projects.

Given the spread across the correlation values, the chances of an aggregate value overestimating or underestimating the #Revisions and #Bugs is high. As a result, individual project analysis is a better option for change- and bug-proneness studies.

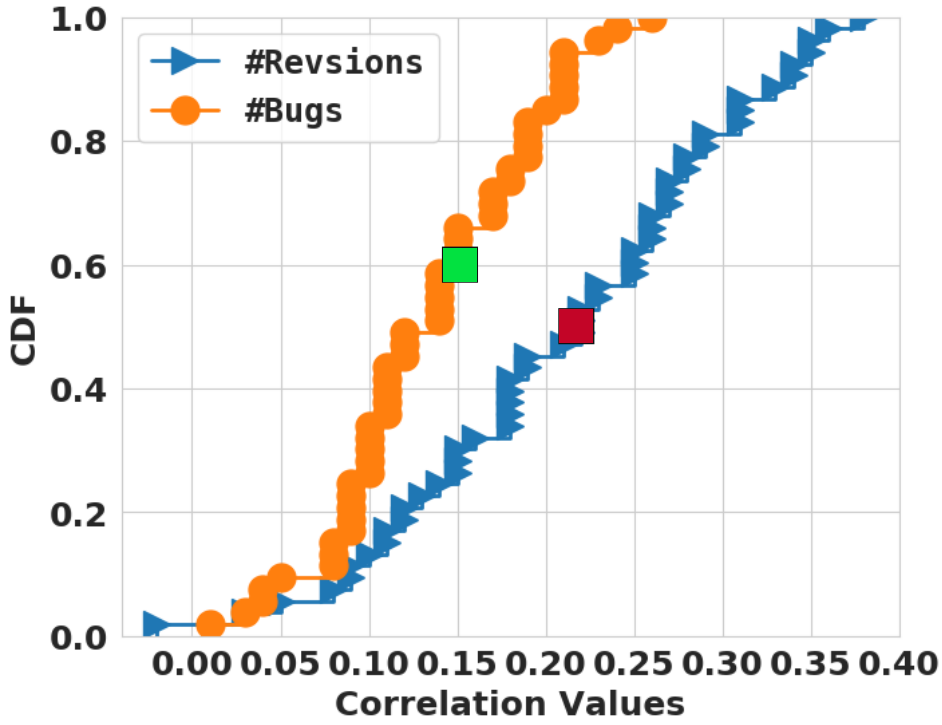


Figure 5.5: Distribution of correlation coefficients of SLOC with #Revisions and #Bugs for 53 projects. Red and green square denotes the aggregated correlation coefficient for SLOC.

RQ5 Summary

Per-project correlations differ meaningfully from aggregate analyses for maintenance indicator prediction. Correlations should be computed on individual projects; if generalizability is desire a project should be compared to many other individual projects.

Chapter 6

Discussion

In this chapter we discuss the implications of our findings along with the threats to validity for our analysis.

6.1 Guidelines for Future Studies

This work demonstrates the impact of different decisions on large historical datasets that can lead to inconsistent conclusions depending on methodological choices. Table 6.1 summarize the answer of the research questions.

Table 6.1: Summary of Research Questions

RQs	Answer
Does the SLOC selection methodology impact evolutionary analyses?	Yes
Should method size be controlled when correlating to change- or bug-proneness?	Yes
Do different ages of methods and temporal choices induce inconsistent results for change/bug proneness?	Yes
Should change kind be considered when making change predictions?	Yes
Do aggregate project analyses meaningfully reflect the correlation for individual project analyses?	No

Since the SLOC values of most methods do not change [70, 86], the correlation

values for a method's introduction, its last commit, and median provide consistent results with #Revisions and #Bugs. Therefore, out of these available options, practitioners can use them interchangeably to train models on past historical data and predict future changes and bugs without much contradiction in the outcome. However, these measurements overestimate the association of SLOC with maintenance indicators and are not truly effective as one might presume when compared with a more accurate representation of SLOC to their respective changes and bugs (e.g., SLOC versioning). Therefore, future studies could adopt the SLOC versioning technique for more consistent and generalizable outcomes.

Once a representation for SLOC is chosen, researchers and practitioners should neutralize the impact of SLOC on the prediction variables of interest (i.e., # changes or # bugs). To do this, they need to evaluate the density of these dependent variables. The next decision is to decide on a timeframe from which to train models. As prior work has reported unstable conclusions provided by models trained on different subsets of data [55], one could opt to use the entire data that is available to train models. However, mining a large amount of data can be resource exhaustive [61]. Our result indicates that the first six months of method history is sufficient for training the change and defect prediction models and understanding software maintenance when age of the methods are normalized.

When examining the first six months data, a method can experience a large number of changes. While each of these changes has a purpose, only a half of all changes (51%) are related to modifying the content of a given method. Therefore, change classification should be performed so that models do not overestimate future changes.

Finally, preliminary analysis on the relationship of code metrics with maintenance indicators should examine each project on its own. Although cross-project predictions are becoming quite popular [97], an alternative form of clustering could look into the association of different metrics for more appropriate project groupings. Alternatively, a finding from one project could be compared to a large set of predictions from many other individual projects.

We recommend future studies be more explicit about their methodological choices and consider these decisions in order to provide more consistent and reproducible outcomes.

6.2 Threats to Validity

As with any complex empirical analysis, this work has several threats to validity:

Internal validity: The findings in this thesis rely on the method histories being accurate and complete. While *CodeShovel* has been shown to have high accuracy, when finding the complete history for 90% of methods, including 97% of historical changes, any automated history-tracking approach is bound to miss some changes. While SLOC is the most commonly used metric used for predicting change- and defect-proneness, other metrics may also be appropriate. This thesis investigates a limited number of aggregation techniques at method-level. Other aggregation measure like Gini [31], Atkinson [8], and Shannon entropy [76] might produce different results.

Construct validity: Our tool for computing metrics for each method at each revision can induce measurement error since the process is fully automated. To reduce this threat, two authors verified the correctness of our tool by randomly selecting and validating 200 Java methods for all collected metrics.

External validity: The most significant threat to the generalizability of our results stems from our project selection. All projects were Open Source systems written in Java. Closed-source systems may exhibit different correlation between SLOC with number of revisions and bugs than open-source systems, as might non-Java systems. Finally, while the age range of the systems was acceptable (with 20 having histories of over 12 years), almost half of the systems had relatively small total size SLOC (15–87 KLOC).

Chapter 7

Conclusion

There have been many disagreement among research efforts related to inconsistent outcomes in software maintenance studies and prediction models. This makes it difficult to replicate the studies or to compare research projects against each other. In this work we use source lines of code (SLOC) as an example metric to explain some of these contradictions that could arise from five different methodological decisions. We analyzed the complete evolutionary data for 1,641,353 methods from 53 projects at the method level.

We found the way a code metric, such as SLOC, is measured and controlled for has a direct impact on its correlation with the number of changes and bugs in a method. For example, using mean SLOC to represent overall SLOC change at every revision of a given method overestimates the correlation values of SLOC with number of revisions and bugs. We also noted that small methods are more change-prone when the size of the method is controlled for.

Additionally, we demonstrated that not controlling the age of methods i.e., treating older and younger methods equally for maintenance studies and training models can impact predictive models. We found that the first six months of a method's history, after the age of methods have been normalized, exhibits a higher correlation of SLOC with number of revisions and bugs for a given method. The type of transformations applied to a given method, such as essential (e.g., body modification and method signature related changes) and non-essential (e.g., formatting fix, annotation update) also can significantly influences the outcome

of analyses. Finally, we showed that projects should be considered individually, rather than aggregated into a single dataset. We hope that these results can help researchers and practitioners curate higher quality evolutionary data leading to improved predictive power for software evolutionary analyses.

Bibliography

[1] A. Agrawal and T. Menzies. Is" better data" better than" better data miners"? In *International Conference on Software Engineering (ICSE)*, pages 1050–1061, 2018. → pages 7, 8, 18

[2] M. Alshayeb and W. Li. An empirical validation of object-oriented metrics in two different iterative software processes. *Transactions on Software Engineering (TSE)*, 29(11):1043–1049, 2003. → page 6

[3] C. Andersson and P. Runeson. A replicated quantitative analysis of fault distributions in complex software systems. *Transactions on Software Engineering (TSE)*, 33(5):273–286, 2007. → pages 7, 8, 9, 24

[4] V. Antinyan, M. Staron, W. Meding, P. Österström, E. Wikstrom, J. Wranker, A. Henriksson, and J. Hansson. Identifying risky areas of software code in agile/lean software development: An industrial experience report. In *Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, pages 154–163, 2014. → pages 6, 9, 15, 27

[5] V. Antinyan, M. Staron, J. Derehag, M. Runsten, E. Wikström, W. Meding, A. Henriksson, and J. Hansson. Identifying complex functions: By investigating various aspects of code complexity. In *Science and Information Conference (SAI)*, pages 879–888, 2015. → pages 9, 15, 27

[6] E. Arisholm, L. C. Briand, and A. Foyen. Dynamic coupling measurement for object-oriented software. *Transactions on Software Engineering (TSE)*, 30(8):491–506, 2004. → page 1

[7] L. J. Arthur. *Software evolution: the software maintenance challenge*. Wiley-Interscience, 1988. → page 1

[8] A. B. Atkinson et al. On the measurement of inequality. *Journal of economic theory*, 2(3):244–263, 1970. → pages 7, 33

[9] R. Baggen, J. P. Correia, K. Schill, and J. Visser. Standardized code quality benchmarking for improving software maintainability. *Software Quality Journal*, 20(2):287–307, 2012. → page 8

[10] R. K. Bandi, V. K. Vaishnavi, and D. E. Turk. Predicting maintenance performance using object-oriented design complexity metrics. *Transactions on Software Engineering (TSE)*, 29(1):77–87, 2003. → page 6

[11] S. Banitaan, K. Daimi, Y. Wang, and M. Akour. Test case selection using software complexity and volume metrics. In *Proceedings of the International Conference on Software Engineering and Data Engineering (SEDE).*, pages 12–14, 2015. → page 9

[12] E. D. Berger, C. Hollenbeck, P. Maj, O. Vitek, and J. Vitek. On the impact of programming languages on code quality: a reproduction study. *Transactions on Programming Languages and Systems (TOPLAS)*, 41(4): 1–24, 2019. → page 12

[13] H. Borges, A. Hora, and M. T. Valente. Understanding the factors that impact the popularity of GitHub repositories. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 334–344, 2016. → page 12

[14] B. Caglayan, A. Bener, and S. Koch. Merits of using repository metrics in defect prediction for open source projects. In *Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, pages 31–36, 2009. → page 6

[15] H. Cervantes and R. Kazman. Software archinaut: a tool to understand architecture, identify technical debt hotspots and manage evolution. In *Proceedings of the 3rd International Conference on Technical Debt*, pages 115–119, 2020. → page 9

[16] S. R. Chidamber and C. F. Kemerer. A metrics suite for object oriented design. *Transactions on Software Engineering (TSE)*, 20(6):476–493, 1994. → page 1

[17] G. R. Choudhary, S. Kumar, K. Kumar, A. Mishra, and C. Catal. Empirical analysis of change metrics for software fault prediction. *Computers & Electrical Engineering*, 67:15–24, 2018. → page 6

[18] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013. → page 17

[19] O. Dabic, E. Aghajani, and G. Bavota. Sampling projects in GitHub for MSR studies. In *International Conference on Mining Software Repositories (MSR)*, page To appear, 2021. URL https://arxiv.org/abs/2103.04682. → page 12

[20] D. Di Nucci, F. Palomba, G. De Rosa, G. Bavota, R. Oliveto, and A. De Lucia. A developer centered bug prediction model. *Transactions on Software Engineering (TSE)*, 44(1):5–24, 2017. → page 1

[21] D. P. Doane and L. E. Seward. Measuring skewness: a forgotten statistic? *Journal of statistics education*, 19(2), 2011. → page 18

[22] K. El Emam, S. Benlarbi, N. Goel, and S. N. Rai. The confounding effect of class size on the validity of object-oriented metrics. *Transactions on Software Engineering (TSE)*, 27(7):630–650, 2001. → pages 3, 6, 7

[23] I. Etikan, S. A. Musa, and R. S. Alkassim. Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, 5(1):1–4, 2016. → page 12

[24] C. Faragó, P. Hegedűs, and R. Ferenc. Cumulative code churn: Impact on maintainability. In *International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 141–150, 2015. → page 9

[25] N. E. Fenton and N. Ohlsson. Quantitative analysis of faults and failures in a complex software system. *Transactions on Software Engineering (TSE)*, 26(8):797–814, 2000. → pages 1, 7, 8, 9, 22, 24

[26] R. Ferenc, D. Bán, T. Grósz, and T. Gyimóthy. Deep learning in static, metric-based bug prediction. *Array*, 6, 2020. → page 1

[27] E. Giger, M. D'Ambros, M. Pinzger, and H. C. Gall. Method-level bug prediction. In *International Symposium on Empirical Software Engineering and Measurement*, pages 171–180, 2012. → pages 2, 9, 13, 24

[28] J. Y. Gil and G. Lalouche. When do software complexity metrics mean nothing?-when examined out of context. *J. Object Technol.*, 15(1):2–1, 2016. → pages 1, 6

[29] Y. Gil and G. Lalouche. On the correlation between size and metric validity. *Empirical Software Engineering*, 22(5):2585–2611, 2017. → pages 1, 3, 6, 7, 8, 10, 11, 17, 21, 29

[30] A. R. Gilpin. Table for conversion of kendall's tau to spearman's rho within the context of measures of magnitude of effect for meta-analysis. *Educational and psychological measurement*, 53(1):87–92, 1993. → page 17

[31] C. Gini. Measurement of inequality of incomes. *The Economic Journal*, 31 (121):124–126, 1921. → pages 7, 33

[32] M. Greiler, A. Zaidman, A. Van Deursen, and M.-A. Storey. Strategies for avoiding text fixture smells during software evolution. In *Working Conference on Mining Software Repositories (MSR)*, pages 387–396, 2013. → pages 7, 8

[33] F. Grund, S. A. Chowdhury, N. C. Bradley, B. Hall, and R. Holmes. Codeshovel: Constructing method-level source code histories. In *International Conference on Software Engineering (ICSE)*, pages 1510–1522, 2021. → pages 2, 13, 15

[34] K. Herzig and A. Zeller. The impact of tangled code changes. In *Working Conference on Mining Software Repositories (MSR)*, pages 121–130, 2013. → pages 9, 10, 29

[35] M. Hess and J. Kromrey. Robust confidence intervals for effect sizes: A comparative study of cohen's d and cliff's delta under non-normality and heterogeneous variances. 2004. → page 17

[36] T. Hoang, H. Khanh Dam, Y. Kamei, D. Lo, and N. Ubayashi. Deepjit: An end-to-end deep learning framework for just-in-time defect prediction. In *International Conference on Mining Software Repositories (MSR)*, pages 34–45, 2019. → page 1

[37] E. M. Hoover. The measurement of industrial localization. *The Review of Economic Statistics*, pages 162–171, 1936. → page 7

[38] L. Inozemtseva and R. Holmes. Coverage is not strongly correlated with test suite effectiveness. In *International conference on software engineering*, pages 435–445, 2014. → pages 7, 17

[39] T. Jiang, L. Tan, and S. Kim. Personalized defect prediction. In *International Conference on Automated Software Engineering (ASE)*, pages 279–289, 2013. → page 13

[40] J. Jiarpakdee, C. Tantithamthavorn, and A. E. Hassan. The impact of correlated metrics on defect models. *arXiv preprint arXiv:1801.10271*, 2018. → pages 7, 17

[41] J. Johnson, S. Lubo, N. Yedla, J. Aponte, and B. Sharif. An empirical study assessing source code readability in comprehension. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 513–523, 2019. → page 6

[42] D. Kafura and G. R. Reddy. The use of software complexity metrics in software maintenance. *Transactions on Software Engineering (TSE)*, (3): 335–343, 1987. → pages 10, 21, 29

[43] D. Kawrykow and M. P. Robillard. Non-essential changes in version histories. In *International Conference on Software Engineering (ICSE)*, pages 351–360, 2011. → pages 10, 27

[44] M. Kondo, D. M. German, O. Mizuno, and E.-H. Choi. The impact of context metrics on just-in-time defect prediction. *Empirical Software Engineering*, 25(1):890–939, 2020. → pages 7, 8

[45] A. G. Koru and H. Liu. Identifying and characterizing change-prone classes in two large-scale open-source products. *Journal of Systems and Software*, 80(1):63–73, 2007. → pages 7, 9

[46] A. G. Koru, D. Zhang, and H. Liu. Modeling the effect of size on defect proneness for open-source software. In *International Workshop on Predictor Models in Software Engineering*, pages 10–10, 2007. → page 1

[47] D. Landman, A. Serebrenik, and J. Vinju. Empirical analysis of the relationship between cc and sloc in a large corpus of java methods. In *International Conference on Software Maintenance and Evolution*, pages 221–230, 2014. → pages 7, 21

[48] D. Landman, A. Serebrenik, E. Bouwers, and J. J. Vinju. Empirical analysis of the relationship between cc and sloc in a large corpus of java methods and c functions. *Journal of Software: Evolution and Process*, 28 (7):589–618, 2016. → pages 6, 17

[49] J. Lefever, Y. Cai, H. Cervantes, R. Kazman, and H. Fang. On the lack of consensus among technical debt detection tools. In *International Conference on Software Engineering: Software Engineering in Practice*, pages 121–130, 2021. → pages 8, 9

[50] L. Madeyski and M. Jureczko. Which process metrics can significantly improve defect prediction models? an empirical study. *Software Quality Journal*, 23(3):393–422, 2015. → page 1

[51] D. Matter, A. Kuhn, and O. Nierstrasz. Assigning bug reports using a vocabulary-based expertise model of developers. In *International working conference on mining software repositories*, pages 131–140, 2009. → pages 9, 10, 29

[52] T. J. McCabe. A complexity measure. *Transactions on Software Engineering (TSE)*, (4):308–320, 1976. → page 1

[53] S. McIntosh and Y. Kamei. Are fix-inducing changes a moving target? a longitudinal case study of just-in-time defect prediction. *Transactions on Software Engineering (TSE)*, 44(5):412–428, 2017. → page 8

[54] T. Menzies, J. Greenwald, and A. Frank. Data mining static code attributes to learn defect predictors. *Transactions on Software Engineering (TSE)*, 33 (1):2–13, 2006. → page 1

[55] T. Menzies, A. Butcher, A. Marcus, T. Zimmermann, and D. Cok. Local vs. global models for effort estimation and defect prediction. In *International Conference on Automated Software Engineering (ASE)*, pages 343–351, 2011. → pages 4, 9, 32

[56] T. Menzies, A. Butcher, D. Cok, A. Marcus, L. Layman, F. Shull, B. Turhan, and T. Zimmermann. Local versus global lessons for defect prediction and effort estimation. *Transactions on Software Engineering (TSE)*, 39(6):822–834, 2012. → page 9

[57] A. Mockus and L. G. Votta. Identifying reasons for software changes using historic databases. In *icsm*, pages 120–130, 2000. → page 16

[58] R. Moser, W. Pedrycz, and G. Succi. A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. In *International conference on Software engineering*, pages 181–190, 2008. → page 6

[59] N. Nagappan and T. Ball. Use of relative code churn measures to predict system defect density. In *International conference on Software engineering*, pages 284–292, 2005. → pages 1, 8, 16

[60] N. Nagappan, A. Zeller, T. Zimmermann, K. Herzig, and B. Murphy. Change bursts as defect predictors. In *International symposium on software reliability engineering*, pages 309–318, 2010. → pages 1, 10, 29

[61] S. N.C., S. Majumder, and T. Menzies. Early life cycle software defect prediction. why? how? In *International Conference on Software Engineering (ICSE)*, pages 448–459, 2021. → pages 1, 9, 32

[62] H. Osman, A. Chis, C. Corrodi, M. Ghafari, and O. Nierstrasz. Exception evolution in long-lived Java systems. In *Proceedings of the International Conference on Mining Software Repositories*, pages 302–311, 2017. → page 8

[63] T. J. Ostrand, E. J. Weyuker, and R. M. Bell. Predicting the location and number of faults in large software systems. *Transactions on Software Engineering (TSE)*, 31(4):340–355, 2005. → pages 1, 8

[64] L. Pascarella, F. Palomba, and A. Bacchelli. Re-evaluating method-level bug prediction. In *International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 592–601, 2018. → pages 1, 2, 6, 7, 8, 9, 10, 13, 18, 29

[65] L. S. Pinto, S. Sinha, and A. Orso. Understanding myths and realities of test-suite evolution. In *International Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 1–11, 2012. → page 13

[66] S. S. Rathore and S. Kumar. Predicting number of faults in software system using genetic programming. *Procedia Computer Science*, 62:303–311, 2015. → page 1

[67] B. Ray, D. Posnett, V. Filkov, and P. Devanbu. A large scale study of programming languages and code quality in github. In *International Symposium on Foundations of Software Engineering*, pages 155–165, 2014. → page 12

[68] B. Ray, M. Nagappan, C. Bird, N. Nagappan, and T. Zimmermann. The uniqueness of changes: Characteristics and applications. In *Working Conference on Mining Software Repositories*, pages 34–44, 2015. → pages 10, 29

[69] B. Ray, V. Hellendoorn, S. Godhane, Z. Tu, A. Bacchelli, and P. Devanbu. On the" naturalness" of buggy code. In *International Conference on Software Engineering (ICSE)*, pages 428–439, 2016. → page 16

[70] G. Robles, I. Herraiz, D. M. Germán, and D. Izquierdo-Cortázar. Modification and developer metrics at the function level: Metrics for the study of the evolution of a software project. In *International Workshop on Emerging Trends in Software Metrics (WETSoM)*, pages 49–55, 2012. → pages 21, 31

[71] J. Romano, J. D. Kromrey, J. Coraggio, and J. Skowronek. Appropriate statistics for ordinal level data: Should we really be using t-test and cohen'sd for evaluating group differences on the nsse and other surveys. In *Annual meeting of the Florida Association of Institutional Research*, volume 13, 2006. → page 17

[72] J. Rosenberg. Some misconceptions about lines of code. In *Proceedings international software metrics symposium*, pages 137–142, 1997. → page 8

[73] S. Scalabrino, G. Bavota, C. Vendome, M. Linares-Vásquez, D. Poshyvanyk, and R. Oliveto. Automatically assessing code understandability: How far are we? In *International Conference on Automated Software Engineering (ASE)*, pages 417–427, 2017. → page 6

[74] A. Schröter, T. Zimmermann, and A. Zeller. Predicting component failures at design time. In *International symposium on Empirical software engineering*, pages 18–27, 2006. → page 1

[75] R. C. Seacord, D. Plakosh, and G. A. Lewis. *Modernizing legacy systems: software technologies, engineering processes, and business practices*. 2003. → page 1

[76] C. E. Shannon. A mathematical theory of communication. *Mobile computing and communications review*, 5(1):3–55, 2001. → pages 7, 33

[77] A. R. Sharafat and L. Tahvildari. Change prediction in object-oriented software systems: A probabilistic approach. *J. Softw.*, 3(5):26–39, 2008. → page 1

[78] R. Shatnawi and W. Li. The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process. *Journal of systems and software*, 81(11):1868–1882, 2008. → pages 1, 6

[79] M. Shepperd. A critique of cyclomatic complexity as a software metric. *Software Engineering Journal*, 3(2):30–36, 1988. → pages 1, 6, 21

[80] D. J. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC, 2003. → page 17

[81] Y. Shin and L. Williams. Can traditional fault prediction models be used for vulnerability prediction? *Empirical Software Engineering*, 18(1): 25–59, 2013. → page 1

[82] Y. Shin, A. Meneely, L. Williams, and J. A. Osborne. Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities. *Transactions on Software Engineering (TSE)*, 37 (6):772–787, 2010. → pages 9, 10, 15, 27, 29

[83] D. I. Sjøberg, A. Yamashita, B. C. Anda, A. Mockus, and T. Dybå. Quantifying the effect of code smells on maintenance effort. *Transactions on Software Engineering (TSE)*, 39(8):1144–1156, 2012. → pages 3, 6

[84] Q. Song, Z. Jia, M. Shepperd, S. Ying, and J. Liu. A general software defect-proneness prediction framework. *Transactions on Software Engineering (TSE)*, 37(3):356–370, 2010. → page 1

[85] D. Spadini, F. Palomba, A. Zaidman, M. Bruntink, and A. Bacchelli. On the relation of test smells to software code quality. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 1–12, 2018. → pages 1, 6, 10, 13, 22, 24, 29

[86] D. Steidl and F. Deissenboeck. How do java methods grow? In *International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 151–160, 2015. → pages 21, 31

[87] M. Tufano, F. Palomba, G. Bavota, M. Di Penta, R. Oliveto, A. De Lucia, and D. Poshyvanyk. An empirical investigation into the nature of test smells. In *International Conference on Automated Software Engineering*, pages 4–15, 2016. → page 22

[88] B. Vasilescu, A. Serebrenik, and M. Van den Brand. By no means: A study on aggregating software metrics. In *Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics*, pages 23–26, 2011. → page 7

[89] M. Viggiato, J. Oliveira, E. Figueiredo, P. Jamshidi, and C. Kästner. How do code changes evolve in different platforms? a mining-based investigation. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 218–222, 2019. → page 1

[90] Z. Wan, X. Xia, A. E. Hassan, D. Lo, J. Yin, and X. Yang. Perceptions, expectations, and challenges in defect prediction. *Transactions on Software Engineering (TSE)*, 46(11):1241–1266, 2018. → page 7

[91] S. Wang, T. Liu, J. Nam, and L. Tan. Deep semantic feature learning for software defect prediction. *Transactions on Software Engineering (TSE)*, 46(12):1267–1293, 2018. → page 8

[92] S. Wang, T. Liu, J. Nam, and L. Tan. Deep semantic feature learning for software defect prediction. *Transactions on Software Engineering (TSE)*, 46(12):1267–1293, 2020. → page 1

[93] E. J. Weyuker, T. J. Ostrand, and R. M. Bell. Do too many cooks spoil the broth? using the number of developers to enhance defect prediction models. *Empirical Software Engineering*, 13(5):539–559, 2008. → page 1

[94] A. Zaidman, B. Van Rompaey, S. Demeyer, and A. Van Deursen. Mining software repositories to study co-evolution of production & test code. In *International Conference on Software Testing, Verification, and Validation (ICST)*, pages 220–229, 2008. → page 13

[95] F. Zhang, A. Mockus, Y. Zou, F. Khomh, and A. E. Hassan. How does context affect the distribution of software maintainability metrics? In *International Conference on Software Maintenance*, pages 350–359, 2013. → page 10

[96] F. Zhang, A. E. Hassan, S. McIntosh, and Y. Zou. The use of summation to aggregate software metrics hinders the performance of defect prediction models. *Transactions on Software Engineering (TSE)*, 43(5):476–491, 2016. → pages 1, 16

[97] F. Zhang, A. Mockus, I. Keivanloo, and Y. Zou. Towards building a universal defect prediction model with rank transformed predictors. *Empirical Software Engineering*, 21(5):2107–2145, 2016. → pages 1, 7, 18, 32

[98] F. Zhang, A. E. Hassan, S. McIntosh, and Y. Zou. The use of summation to aggregate software metrics hinders the performance of defect prediction models. *Transactions on Software Engineering (TSE)*, 43(5):476–491, 2017. → pages 7, 16, 17, 18

[99] H. Zhang. An investigation of the relationships between lines of code and defects. In *International Conference on Software Maintenance*, pages 274–283, 2009. → pages 1, 7, 8, 9, 17, 22, 24

[100] Y. Zhou, H. Leung, and B. Xu. Examining the potentially confounding effect of class size on the associations between object-oriented metrics and

change-proneness. *Transactions on Software Engineering (TSE)*, 35(5): 607–623, 2009. → page 1

[101] Y. Zhou, B. Xu, H. Leung, and L. Chen. An in-depth study of the potentially confounding effect of class size in fault prediction. *Transactions on Software Engineering and Methodology*, 23(1):1–51, 2014. → page 7

[102] T. Zimmermann and N. Nagappan. Predicting defects using network analysis on dependency graphs. In *International conference on Software engineering*, pages 531–540, 2008. → page 16

[103] T. Zimmermann, R. Premraj, and A. Zeller. Predicting defects for eclipse. In *Third International Workshop on Predictor Models in Software Engineering*, pages 9–9, 2007. → page 16

Appendix A

Projects Information

This appendix includes information about 53 Java open-source projects discussed in Chapter 4.

Repo	Github url	Stars	Contributors	Ref. Commit	Years	#Methods	#Files	TtlSLOC	TtlRevisions
wildfly	wildfly/wildfly	2593	343	a9e061675ea9b3	10.85205479	112098	10081	616780	153071
wicket	apache/wicket	570	83	e8cdd56db3	16.47123288	46598	3256	216748	147121
vraptor4	caelum/vraptor4	343	48	593ce9ad60f9d3	5.64109589	6988	551	26725	9049
voldemort	voldemort/voldemort	2475	65	a7dbdea5803202	6.375342466	12157	997	175868	19722
Twitter4J	Twitter4J/Twitter4J	2563	132	8376fade8d5578	11.16986301	5546	411	31577	20667
truth	google/truth	2296	83	60dcd0931f52ac	10.00273973	4609	185	34461	13154
titan	thinkaurelius/titan	5135	34	ee226e524	3.769863014	16796	904	74510	27617
swagger-core	swagger-api/swagger-core	6796	202	567bb88c9a789c	9.819178082	9653	761	50583	13836
spring-framework	spring-projects/spring-framework	43641	544	42061d27bdac07	12.3369863	90138	7512	696598	243270
spring-boot	spring-projects/spring-boot	56378	828	7a3bd6d44f	8.295890411	49025	5674	330089	117460
spark	apache/spark	30409	1694	54079b0f95f007	10.64931507	2089	986	79958	3224
soot	soot-oss/soot	1860	109	beb5a98be5402b	24.39452055	66256	3684	391457	103727
sonarqube	SonarSource/sonarqube	5937	137	39abd3de73733b	10.81917808	130333	7293	517951	249504
RxJava	ReactiveX/RxJava	44922	278	82f489e1d3ca71	9.063013699	68428	1870	312499	96018
rabbitmq-java-client	rabbitmq/rabbitmq-java-client	996	52	cc7c86773e439d	13.20273973	4161	367	32052	10092
pmd	pmd/pmd	3482	229	848ec2e7bb	18.63835616	31604	2832	142831	108589
openmrs-core	openmrs/openmrs-core	1017	371	ffcc19911bea2f7	15.18082192	24382	1149	126645	63931
okhttp	square/okhttp	40428	236	edf477cb4	9.693150685	7387	158	37132	39574
netty	netty/netty	27185	526	6724786dcc9fa3	12.65479452	38016	2700	300383	68104
mybatis-3	mybatis/mybatis-3	15965	182	b791f4476f6fba6	11.0630137	8850	1238	60825	15166
mongo-java-driver	mongodb/mongo-java-driver	2422	151	38117f16f6df7b2	11.85479452	27991	1648	133717	74750
mockito	mockito/mockito	12043	221	43facffb6	13.23287671	8991	944	55569	27011
lombok	projectlombok/lombok	10391	114	5120abe4741c78	11.70684932	7789	1503	84609	15509
logging-log4j2	apache/logging-log4j2	1230	122	e24e2e8901ccfff	11.06849315	19997	2262	173812	46656
junit5	junit-team/junit5	4691	167	da2dee2bfb	5.838356164	9928	1190	80768	35207
junit4	junit-team/junit4	8160	151	02aaa01b	20.8739726	4762	471	31242	13060
jna	java-native-access/jna	6655	143	a0641dd92	23.2109589	4787	564	87353	13726
jgit	eclipse/jgit	983	140	4560bdf7e	10.82739726	19010	1587	232145	30665
jetty.project	eclipse/jetty.project	3160	159	4c67b886a4f3ef6	11.89589041	51534	2948	429856	109289
javaparser	javaparser/javaparser	3771	157	1ce7d85c2	9.273972603	20309	1651	175026	72545
Hystrix	Netflix/Hystrix	21751	109	3cb21589895e9f	6.676712329	5890	411	50510	7231
hibernate-search	hibernate/hibernate-search	390	58	994cb9e827	13.47671233	42596	4044	263523	136040
hibernate-orm	hibernate/hibernate-orm	4678	420	cfc7b972501bf26	13.59726027	95135	10260	779669	179706
hector	hector-client/hector	648	71	a302e68ca8d91b	10.83287671	3682	459	31365	6123
hawtio	hawtio/hawtio	1233	102	bbd6819052152	8.463013699	4642	204	15259	4140
guava	google/guava	41775	264	eec5e6d762357e	11.69863014	66468	1979	377142	68277
gson	google/gson	19809	109	ceae88bd	12.35068493	3633	207	25280	5781
flyway	flyway/flyway	5987	2821	03bec1cc4e78f8	11.14794521	7697	445	24948	12298
flink	apache/flink	16764	908	6a8b0301170f9d	10.14794521	118815	11282	1259298	227872
fastjson	alibaba/fastjson	23594	173	9b8b18677	9.652054795	20768	2982	179371	9460
Essentials	EssentialsX/Essentials	1086	215	2c68d1b8663322	10.09041096	6381	442	41610	19943
DSpace	DSpace/DSpace	573	146	3c1b90d60395f8	19.20273973	26394	2551	261003	61574
drill	apache/drill	1562	168	771c81194ab49a	9.246575342	48539	4289	576427	56317
docx4j	plutext/docx4j	1620	37	26aaa13fe	13.19726027	68568	3897	321657	96542
cucumber-jvm	cucumber/cucumber-jvm	2291	226	bb8f6bb1e	12.43835616	10678	777	43999	25250
commons-lang	apache/commons-lang	2138	160	a1495290b	18.55890411	13944	397	84443	39937
commons-io	apache/commons-io	787	76	29b70e15	19.03561644	4207	360	39836	16667
checkstyle	checkstyle/checkstyle	6115	287	3d8b43445	19.61369863	16350	2702	239197	51991
atmosphere	Atmosphere/atmosphere	3524	111	75f568553bfb38f	10.69863014	7466	417	41605	13463
astyanax	Netflix/astyanax	1021	55	dadde734e590a6	9.778082192	6391	618	55399	9732
assertj-core	assertj/assertj-core	1981	259	f8730b34cb54c6	11	48611	4466	192774	61390
antlr4	antlr/antlr4	10306	259	dc66788476a7e9	11.31232877	27909	652	56952	22107
ant	apache/ant	303	58	5f8c6370a	21.05753425	33616	1317	145017	78089
Total	**-**	**518433**	**14793**	**-**	**653**	**1598592**	**122535**	**10846053**	**3171244**
Mean	**-**	**9782**	**279**	**-**	**12**	**30162**	**2312**	**204643**	**59835**
Median	**-**	**3482**	**157**	**-**	**11**	**16796**	**1238**	**126645**	**35207**
Min	**-**	**303**	**34**	**-**	**4**	**2089**	**158**	**15259**	**3224**
Max	**-**	**56378**	**2821**	**-**	**24**	**130333**	**11282**	**1259298**	**249504**

Appendix B

RQ1-SLOC Measurements

This appendix includes all correlation values per project of different SLOC measurements with #Revisions and #Bugs along with Cliff's δ and Mann-Whitney U significance test results discussed in Chapter 5.1. The different SLOC measurements are:

- `introSLOC`: The initial SLOC of a method when it was introduced into the repository.
- `lastSLOC`: The final SLOC value of a method when it stopped evolving.
- `avgSLOC`: The mean SLOC value to represent all SLOC changes of an evolving method.
- `medianSLOC`: The median SLOC to represent all SLOC changes of an evolving method.
- `stdSLOC`: The standard deviation of SLOC values of an evolving method.
- `versionedSLOC`: The SLOC value evaluated using the versioning technique described in Chapter 5.1.

B.1 SLOC measurements with #Revisions (all methods)

Kendall τ correlation coefficients of all 53 projects for each SLOC measurements with #Revisions is given below. All the methods in each project were used to compute the correlation values.

corr	p_value	significant	correlation of	repo	Methods considered
0.21	0	yes	avgSLOC_with_revisions	ant	all
-0.1	3.25E-77	yes	avgSLOC_with_revisions	antlr4	all
0.17	0	yes	avgSLOC_with_revisions	assertj-core	all
0.1	4.65E-22	yes	avgSLOC_with_revisions	astyanax	all
0.3	1.68E-167	yes	avgSLOC_with_revisions	atmosphere	all
0.26	0	yes	avgSLOC_with_revisions	checkstyle	all
0.12	5.72E-20	yes	avgSLOC_with_revisions	commons-io	all
0.11	1.24E-61	yes	avgSLOC_with_revisions	commons-lang	all
0.2	2.47E-100	yes	avgSLOC_with_revisions	cucumber-jvm	all
0.21	0	yes	avgSLOC_with_revisions	docx4j	all
0.24	0	yes	avgSLOC_with_revisions	drill	all
0.29	0	yes	avgSLOC_with_revisions	DSpace	all
0.23	1.15E-91	yes	avgSLOC_with_revisions	Essentials	all
0.28	0	yes	avgSLOC_with_revisions	fastjson	all
0.19	0	yes	avgSLOC_with_revisions	flink	all
0.14	4.55E-31	yes	avgSLOC_with_revisions	flyway	all
0.14	2.70E-23	yes	avgSLOC_with_revisions	gson	all
0.2	0	yes	avgSLOC_with_revisions	guava	all
0.05	0.000232814950	yes	avgSLOC_with_revisions	hawtio	all
0.26	3.49E-70	yes	avgSLOC_with_revisions	hector	all
0.1	9.81E-263	yes	avgSLOC_with_revisions	hibernate-orm	all
0.22	0	yes	avgSLOC_with_revisions	hibernate-search	all
0.28	1.74E-129	yes	avgSLOC_with_revisions	Hystrix	all
0.21	4.81E-274	yes	avgSLOC_with_revisions	javaparser	all
0.32	0	yes	avgSLOC_with_revisions	jetty.project	all
0.2	7.81E-206	yes	avgSLOC_with_revisions	jgit	all
0.12	1.01E-26	yes	avgSLOC_with_revisions	jna	all
0.03	0.006224828137	yes	avgSLOC_with_revisions	junit4	all
0.21	1.36E-123	yes	avgSLOC_with_revisions	junit5	all
0.24	6.08E-298	yes	avgSLOC_with_revisions	logging-log4j2	all
0.23	1.74E-105	yes	avgSLOC_with_revisions	lombok	all
0.1	1.35E-30	yes	avgSLOC_with_revisions	mockito	all
0.17	1.10E-172	yes	avgSLOC_with_revisions	mongo-java-driver	all
0.25	1.33E-177	yes	avgSLOC_with_revisions	mybatis-3	all
0.27	0	yes	avgSLOC_with_revisions	netty	all
0.26	1.18E-141	yes	avgSLOC_with_revisions	okhttp	all
0.12	2.66E-100	yes	avgSLOC_with_revisions	openmrs-core	all
0.22	0	yes	avgSLOC_with_revisions	pmd	all
0.24	2.47E-76	yes	avgSLOC_with_revisions	rabbitmq-java-client	all
0.26	0	yes	avgSLOC_with_revisions	RxJava	all
0.23	0	yes	avgSLOC_with_revisions	sonarqube	all
0.12	2.00E-267	yes	avgSLOC_with_revisions	soot	all
0.05	0.04570099412	yes	avgSLOC_with_revisions	spark	all

0.24	0	yes	avgSLOC_with_revisions	spring-boot	all
0.24	0	yes	avgSLOC_with_revisions	spring-framework	all
0.13	1.13E-37	yes	avgSLOC_with_revisions	swagger-core	all
0.26	1.52E-130	yes	avgSLOC_with_revisions	titan	all
0.27	9.33E-113	yes	avgSLOC_with_revisions	truth	all
0.22	3.30E-72	yes	avgSLOC_with_revisions	Twitter4J	all
0.3	0	yes	avgSLOC_with_revisions	voldemort	all
0.13	5.79E-26	yes	avgSLOC_with_revisions	vraptor4	all
0.06	2.07E-44	yes	avgSLOC_with_revisions	wicket	all
0.23	0	yes	avgSLOC_with_revisions	wildfly	all
0.19	4.38E-278	yes	introSLOC_with_revisions	ant	all
-0.11	1.33E-100	yes	introSLOC_with_revisions	antlr4	all
0.17	0	yes	introSLOC_with_revisions	assertj-core	all
0.06	1.13E-08	yes	introSLOC_with_revisions	astyanax	all
0.24	2.09E-111	yes	introSLOC_with_revisions	atmosphere	all
0.26	0	yes	introSLOC_with_revisions	checkstyle	all
0.1	2.09E-14	yes	introSLOC_with_revisions	commons-io	all
0.1	1.49E-45	yes	introSLOC_with_revisions	commons-lang	all
0.17	3.16E-73	yes	introSLOC_with_revisions	cucumber-jvm	all
0.2	0	yes	introSLOC_with_revisions	docx4j	all
0.21	0	yes	introSLOC_with_revisions	drill	all
0.26	0	yes	introSLOC_with_revisions	DSpace	all
0.22	6.06E-78	yes	introSLOC_with_revisions	Essentials	all
0.26	0	yes	introSLOC_with_revisions	fastjson	all
0.18	0	yes	introSLOC_with_revisions	flink	all
0.11	1.31E-19	yes	introSLOC_with_revisions	flyway	all
0.11	1.60E-13	yes	introSLOC_with_revisions	gson	all
0.2	0	yes	introSLOC_with_revisions	guava	all
0.03	0.02561251304	yes	introSLOC_with_revisions	hawtio	all
0.2	1.17E-43	yes	introSLOC_with_revisions	hector	all
0.09	1.71E-194	yes	introSLOC_with_revisions	hibernate-orm	all
0.2	5.26E-283	yes	introSLOC_with_revisions	hibernate-search	all
0.26	2.51E-110	yes	introSLOC_with_revisions	Hystrix	all
0.14	1.82E-125	yes	introSLOC_with_revisions	javaparser	all
0.29	0	yes	introSLOC_with_revisions	jetty.project	all
0.18	1.90E-166	yes	introSLOC_with_revisions	jgit	all
0.11	1.28E-20	yes	introSLOC_with_revisions	jna	all
0.01	0.4252577261	no	introSLOC_with_revisions	junit4	all
0.19	5.01E-99	yes	introSLOC_with_revisions	junit5	all
0.22	3.98E-238	yes	introSLOC_with_revisions	logging-log4j2	all
0.21	5.82E-85	yes	introSLOC_with_revisions	lombok	all
0.1	2.36E-28	yes	introSLOC_with_revisions	mockito	all
0.13	2.71E-109	yes	introSLOC_with_revisions	mongo-java-driver	all
0.25	6.35E-181	yes	introSLOC_with_revisions	mybatis-3	all

0.25	0	yes	introSLOC_with_revisions	netty	all
0.23	4.37E-109	yes	introSLOC_with_revisions	okhttp	all
0.11	2.18E-97	yes	introSLOC_with_revisions	openmrs-core	all
0.2	0	yes	introSLOC_with_revisions	pmd	all
0.23	9.98E-67	yes	introSLOC_with_revisions	rabbitmq-java-client	all
0.25	0	yes	introSLOC_with_revisions	RxJava	all
0.21	0	yes	introSLOC_with_revisions	sonarqube	all
0.09	6.09E-156	yes	introSLOC_with_revisions	soot	all
0.04	0.0778211366	no	introSLOC_with_revisions	spark	all
0.23	0	yes	introSLOC_with_revisions	spring-boot	all
0.22	0	yes	introSLOC_with_revisions	spring-framework	all
0.11	1.48E-25	yes	introSLOC_with_revisions	swagger-core	all
0.22	9.47E-97	yes	introSLOC_with_revisions	titan	all
0.25	6.18E-92	yes	introSLOC_with_revisions	truth	all
0.2	4.44E-56	yes	introSLOC_with_revisions	Twitter4J	all
0.27	7.95E-252	yes	introSLOC_with_revisions	voldemort	all
0.11	3.75E-19	yes	introSLOC_with_revisions	vraptor4	all
0.02	1.54E-08	yes	introSLOC_with_revisions	wicket	all
0.2	0	yes	introSLOC_with_revisions	wildfly	all
0.2	1.67E-299	yes	lastSLOC_with_revisions	ant	all
-0.1	6.31E-80	yes	lastSLOC_with_revisions	antlr4	all
0.17	0	yes	lastSLOC_with_revisions	assertj-core	all
0.07	1.17E-11	yes	lastSLOC_with_revisions	astyanax	all
0.29	1.57E-154	yes	lastSLOC_with_revisions	atmosphere	all
0.25	0	yes	lastSLOC_with_revisions	checkstyle	all
0.11	1.09E-17	yes	lastSLOC_with_revisions	commons-io	all
0.1	1.33E-51	yes	lastSLOC_with_revisions	commons-lang	all
0.19	5.77E-89	yes	lastSLOC_with_revisions	cucumber-jvm	all
0.21	0	yes	lastSLOC_with_revisions	docx4j	all
0.23	0	yes	lastSLOC_with_revisions	drill	all
0.29	0	yes	lastSLOC_with_revisions	DSpace	all
0.22	8.90E-79	yes	lastSLOC_with_revisions	Essentials	all
0.27	0	yes	lastSLOC_with_revisions	fastjson	all
0.18	0	yes	lastSLOC_with_revisions	flink	all
0.12	4.90E-23	yes	lastSLOC_with_revisions	flyway	all
0.13	3.73E-18	yes	lastSLOC_with_revisions	gson	all
0.19	0	yes	lastSLOC_with_revisions	guava	all
0.04	0.000939792537	yes	lastSLOC_with_revisions	hawtio	all
0.22	1.00E-51	yes	lastSLOC_with_revisions	hector	all
0.1	5.15E-251	yes	lastSLOC_with_revisions	hibernate-orm	all
0.21	0.00E-02	yes	lastSLOC_with_revisions	hibernate-search	all
0.24	1.93E-95	yes	lastSLOC_with_revisions	Hystrix	all
0.2	2.05E-243	yes	lastSLOC_with_revisions	javaparser	all
0.31	0	yes	lastSLOC_with_revisions	jetty.project	all

0.19	1.69E-192	yes	lastSLOC_with_revisions	jgit	all
0.1	2.83E-19	yes	lastSLOC_with_revisions	jna	all
0.02	0.04571063803	yes	lastSLOC_with_revisions	junit4	all
0.18	7.19E-97	yes	lastSLOC_with_revisions	junit5	all
0.23	1.01E-270	yes	lastSLOC_with_revisions	logging-log4j2	all
0.23	3.34E-101	yes	lastSLOC_with_revisions	lombok	all
0.08	3.30E-20	yes	lastSLOC_with_revisions	mockito	all
0.15	1.05E-140	yes	lastSLOC_with_revisions	mongo-java-driver	all
0.25	1.14E-167	yes	lastSLOC_with_revisions	mybatis-3	all
0.26	0	yes	lastSLOC_with_revisions	netty	all
0.25	1.54E-129	yes	lastSLOC_with_revisions	okhttp	all
0.1	1.54E-69	yes	lastSLOC_with_revisions	openmrs-core	all
0.19	0	yes	lastSLOC_with_revisions	pmd	all
0.23	4.01E-69	yes	lastSLOC_with_revisions	rabbitmq-java-client	all
0.24	0	yes	lastSLOC_with_revisions	RxJava	all
0.22	0	yes	lastSLOC_with_revisions	sonarqube	all
0.11	5.77E-258	yes	lastSLOC_with_revisions	soot	all
0.04	0.1313814847	no	lastSLOC_with_revisions	spark	all
0.23	0	yes	lastSLOC_with_revisions	spring-boot	all
0.23	0	yes	lastSLOC_with_revisions	spring-framework	all
0.12	5.85E-34	yes	lastSLOC_with_revisions	swagger-core	all
0.22	8.02E-96	yes	lastSLOC_with_revisions	titan	all
0.26	3.12E-99	yes	lastSLOC_with_revisions	truth	all
0.2	9.54E-57	yes	lastSLOC_with_revisions	Twitter4J	all
0.29	2.41E-307	yes	lastSLOC_with_revisions	voldemort	all
0.11	2.53E-20	yes	lastSLOC_with_revisions	vraptor4	all
0.06	3.34E-46	yes	lastSLOC_with_revisions	wicket	all
0.22	0	yes	lastSLOC_with_revisions	wildfly	all
0.21	0	yes	medianSLOC_with_revisions	ant	all
-0.1	3.34E-77	yes	medianSLOC_with_revisions	antlr4	all
0.17	0	yes	medianSLOC_with_revisions	assertj-core	all
0.08	6.10E-15	yes	medianSLOC_with_revisions	astyanax	all
0.29	3.32E-156	yes	medianSLOC_with_revisions	atmosphere	all
0.26	0	yes	medianSLOC_with_revisions	checkstyle	all
0.12	2.08E-19	yes	medianSLOC_with_revisions	commons-io	all
0.11	4.79E-56	yes	medianSLOC_with_revisions	commons-lang	all
0.2	1.35E-94	yes	medianSLOC_with_revisions	cucumber-jvm	all
0.21	0	yes	medianSLOC_with_revisions	docx4j	all
0.24	0	yes	medianSLOC_with_revisions	drill	all
0.29	0	yes	medianSLOC_with_revisions	DSpace	all
0.24	2.65E-97	yes	medianSLOC_with_revisions	Essentials	all
0.28	0	yes	medianSLOC_with_revisions	fastjson	all
0.19	0	yes	medianSLOC_with_revisions	flink	all
0.13	4.15E-30	yes	medianSLOC_with_revisions	flyway	all

0.14	2.74E-22	yes	medianSLOC_with_revisions	gson	all
0.2	0	yes	medianSLOC_with_revisions	guava	all
0.05	0.000384003691	yes	medianSLOC_with_revisions	hawtio	all
0.25	6.19E-68	yes	medianSLOC_with_revisions	hector	all
0.1	6.62E-253	yes	medianSLOC_with_revisions	hibernate-orm	all
0.22	0	yes	medianSLOC_with_revisions	hibernate-search	all
0.28	8.41E-128	yes	medianSLOC_with_revisions	Hystrix	all
0.2	1.51E-241	yes	medianSLOC_with_revisions	javaparser	all
0.31	0	yes	medianSLOC_with_revisions	jetty.project	all
0.2	1.89E-201	yes	medianSLOC_with_revisions	jgit	all
0.12	3.42E-24	yes	medianSLOC_with_revisions	jna	all
0.02	0.03438039749	yes	medianSLOC_with_revisions	junit4	all
0.19	2.67E-102	yes	medianSLOC_with_revisions	junit5	all
0.23	1.19E-279	yes	medianSLOC_with_revisions	logging-log4j2	all
0.23	9.66E-102	yes	medianSLOC_with_revisions	lombok	all
0.09	5.54E-25	yes	medianSLOC_with_revisions	mockito	all
0.16	2.72E-148	yes	medianSLOC_with_revisions	mongo-java-driver	all
0.25	2.61E-173	yes	medianSLOC_with_revisions	mybatis-3	all
0.27	0	yes	medianSLOC_with_revisions	netty	all
0.26	2.69E-142	yes	medianSLOC_with_revisions	okhttp	all
0.11	5.53E-90	yes	medianSLOC_with_revisions	openmrs-core	all
0.2	0	yes	medianSLOC_with_revisions	pmd	all
0.23	8.57E-72	yes	medianSLOC_with_revisions	rabbitmq-java-client	all
0.25	0	yes	medianSLOC_with_revisions	RxJava	all
0.22	0	yes	medianSLOC_with_revisions	sonarqube	all
0.12	1.41E-264	yes	medianSLOC_with_revisions	soot	all
0.05	0.05926339546	no	medianSLOC_with_revisions	spark	all
0.24	0	yes	medianSLOC_with_revisions	spring-boot	all
0.23	0	yes	medianSLOC_with_revisions	spring-framework	all
0.13	7.84E-36	yes	medianSLOC_with_revisions	swagger-core	all
0.24	5.70E-116	yes	medianSLOC_with_revisions	titan	all
0.26	8.76E-107	yes	medianSLOC_with_revisions	truth	all
0.21	6.53E-64	yes	medianSLOC_with_revisions	Twitter4J	all
0.3	0	yes	medianSLOC_with_revisions	voldemort	all
0.12	1.46E-22	yes	medianSLOC_with_revisions	vraptor4	all
0.05	8.07E-36	yes	medianSLOC_with_revisions	wicket	all
0.23	0	yes	medianSLOC_with_revisions	wildfly	all
0.5	0	yes	stdSLOC_with_revisions	antlr4	all
0.27	0	yes	stdSLOC_with_revisions	assertj-core	all
0.47	0	yes	stdSLOC_with_revisions	astyanax	all
0.58	0	yes	stdSLOC_with_revisions	atmosphere	all
0.45	0	yes	stdSLOC_with_revisions	drill	all
0.46	0	yes	stdSLOC_with_revisions	DSpace	all
0.56	0	yes	stdSLOC_with_revisions	Essentials	all

0.44	0	yes	stdSLOC_with_revisions	flink	all
0.47	7.25E-281	yes	stdSLOC_with_revisions	flyway	all
0.3	0	yes	stdSLOC_with_revisions	guava	all
0.38	9.11E-139	yes	stdSLOC_with_revisions	hawtio	all
0.51	4.40E-230	yes	stdSLOC_with_revisions	hector	all
0.26	0	yes	stdSLOC_with_revisions	hibernate-orm	all
0.54	0	yes	stdSLOC_with_revisions	Hystrix	all
0.49	0	yes	stdSLOC_with_revisions	jetty.project	all
0.41	0	yes	stdSLOC_with_revisions	logging-log4j2	all
0.41	2.00E-265	yes	stdSLOC_with_revisions	lombok	all
0.46	0	yes	stdSLOC_with_revisions	mongo-java-driver	all
0.32	2.16E-231	yes	stdSLOC_with_revisions	mybatis-3	all
0.47	0	yes	stdSLOC_with_revisions	netty	all
0.41	0	yes	stdSLOC_with_revisions	openmrs-core	all
0.47	4.87E-224	yes	stdSLOC_with_revisions	rabbitmq-java-client	all
0.42	0	yes	stdSLOC_with_revisions	RxJava	all
0.47	0	yes	stdSLOC_with_revisions	sonarqube	all
0.34	0	yes	stdSLOC_with_revisions	soot	all
0.39	1.02E-45	yes	stdSLOC_with_revisions	spark	all
0.41	0	yes	stdSLOC_with_revisions	spring-framework	all
0.44	0	yes	stdSLOC_with_revisions	swagger-core	all
0.49	2.07E-299	yes	stdSLOC_with_revisions	truth	all
0.49	1.13E-284	yes	stdSLOC_with_revisions	Twitter4J	all
0.56	0	yes	stdSLOC_with_revisions	voldemort	all
0.52	0	yes	stdSLOC_with_revisions	vraptor4	all
0.46	0	yes	stdSLOC_with_revisions	wildfly	all
0.38	0	yes	stdSLOC_with_revisions	ant	all
0.37	0	yes	stdSLOC_with_revisions	checkstyle	all
0.39	8.36E-147	yes	stdSLOC_with_revisions	commons-io	all
0.36	0	yes	stdSLOC_with_revisions	commons-lang	all
0.45	0	yes	stdSLOC_with_revisions	cucumber-jvm	all
0.17	0	yes	stdSLOC_with_revisions	docx4j	all
0.37	0	yes	stdSLOC_with_revisions	fastjson	all
0.42	6.55E-142	yes	stdSLOC_with_revisions	gson	all
0.44	0	yes	stdSLOC_with_revisions	hibernate-search	all
0.42	0	yes	stdSLOC_with_revisions	javaparser	all
0.43	0	yes	stdSLOC_with_revisions	jgit	all
0.39	2.30E-200	yes	stdSLOC_with_revisions	jna	all
0.36	5.75E-166	yes	stdSLOC_with_revisions	junit4	all
0.43	0	yes	stdSLOC_with_revisions	junit5	all
0.44	0	yes	stdSLOC_with_revisions	mockito	all
0.53	0	yes	stdSLOC_with_revisions	okhttp	all
0.48	0	yes	stdSLOC_with_revisions	pmd	all
0.42	0	yes	stdSLOC_with_revisions	spring-boot	all

0.59	0	yes	stdSLOC_with_revisions	titan	all
0.42	0	yes	stdSLOC_with_revisions	wicket	all
-0.02	0.002308031025	yes	versionedSLOC_with_revisions	antlr4	all
0.18	0	yes	versionedSLOC_with_revisions	assertj-core	all
0.05	6.32E-08	yes	versionedSLOC_with_revisions	astyanax	all
0.36	0	yes	versionedSLOC_with_revisions	atmosphere	all
0.29	0	yes	versionedSLOC_with_revisions	drill	all
0.26	0	yes	versionedSLOC_with_revisions	DSpace	all
0.31	1.76E-219	yes	versionedSLOC_with_revisions	Essentials	all
0.25	0	yes	versionedSLOC_with_revisions	flink	all
0.27	7.88E-131	yes	versionedSLOC_with_revisions	flyway	all
0.22	0	yes	versionedSLOC_with_revisions	guava	all
0.23	3.94E-62	yes	versionedSLOC_with_revisions	hawtio	all
0.27	5.37E-100	yes	versionedSLOC_with_revisions	hector	all
0.12	0	yes	versionedSLOC_with_revisions	hibernate-orm	all
0.35	1.13E-259	yes	versionedSLOC_with_revisions	Hystrix	all
0.34	0	yes	versionedSLOC_with_revisions	jetty.project	all
0.19	3.81E-224	yes	versionedSLOC_with_revisions	logging-log4j2	all
0.25	3.85E-136	yes	versionedSLOC_with_revisions	lombok	all
0.16	4.52E-186	yes	versionedSLOC_with_revisions	mongo-java-driver	all
0.31	4.39E-193	yes	versionedSLOC_with_revisions	mybatis-3	all
0.31	0	yes	versionedSLOC_with_revisions	netty	all
0.14	5.67E-144	yes	versionedSLOC_with_revisions	openmrs-core	all
0.25	5.71E-83	yes	versionedSLOC_with_revisions	rabbitmq-java-client	all
0.26	0	yes	versionedSLOC_with_revisions	RxJava	all
0.28	0	yes	versionedSLOC_with_revisions	sonarqube	all
0.27	0	yes	versionedSLOC_with_revisions	soot	all
0.22	9.30E-24	yes	versionedSLOC_with_revisions	spark	all
0.19	0	yes	versionedSLOC_with_revisions	spring-framework	all
0.18	7.07E-92	yes	versionedSLOC_with_revisions	swagger-core	all
0.18	5.00E-62	yes	versionedSLOC_with_revisions	truth	all
0.15	7.39E-39	yes	versionedSLOC_with_revisions	Twitter4J	all
0.35	0	yes	versionedSLOC_with_revisions	voldemort	all
0.18	5.50E-59	yes	versionedSLOC_with_revisions	vraptor4	all
0.33	0	yes	versionedSLOC_with_revisions	wildfly	all
0.28	0	yes	versionedSLOC_with_revisions	ant	all
0.23	1.79E-251	yes	versionedSLOC_with_revisions	checkstyle	all
0.09	5.60E-11	yes	versionedSLOC_with_revisions	commons-io	all
0.13	5.13E-69	yes	versionedSLOC_with_revisions	commons-lang	all
0.22	3.46E-109	yes	versionedSLOC_with_revisions	cucumber-jvm	all
0.29	0	yes	versionedSLOC_with_revisions	docx4j	all
0.38	0	yes	versionedSLOC_with_revisions	fastjson	all
0.18	3.14E-38	yes	versionedSLOC_with_revisions	gson	all
0.11	9.77E-102	yes	versionedSLOC_with_revisions	hibernate-search	all

0.09	1.36E-64	yes	versionedSLOC_with_revisions	javaparser	all
0.26	0.00E-02	yes	versionedSLOC_with_revisions	jgit	all
0.11	3.09E-21	yes	versionedSLOC_with_revisions	jna	all
0.03	0.007717160702	yes	versionedSLOC_with_revisions	junit4	all
0.12	7.35E-56	yes	versionedSLOC_with_revisions	junit5	all
0.1	1.09E-29	yes	versionedSLOC_with_revisions	mockito	all
0.15	1.12E-66	yes	versionedSLOC_with_revisions	okhttp	all
0.21	0	yes	versionedSLOC_with_revisions	pmd	all
0.15	4.39E-286	yes	versionedSLOC_with_revisions	spring-boot	all
0.34	0	yes	versionedSLOC_with_revisions	titan	all
0.08	1.95E-103	yes	versionedSLOC_with_revisions	wicket	all

B.2 SLOC measurements with #Bugs (all methods)

Kendall τ correlation coefficient of all 53 projects for each SLOC measurements with #Bugs is given below. All the methods in each project were used to compute the correlation values.

corr	p_value	significant	correlation of	repo	Methods considered
0.16	1.26E-175	yes	avgSLOC_with_bugs	ant	all
0.01	0.2221380945	no	avgSLOC_with_bugs	antlr4	all
0.08	2.36E-65	yes	avgSLOC_with_bugs	assertj-core	all
0.08	1.62E-11	yes	avgSLOC_with_bugs	astyanax	all
0.29	8.81E-153	yes	avgSLOC_with_bugs	atmosphere	all
0.2	9.08E-172	yes	avgSLOC_with_bugs	checkstyle	all
0.08	6.76E-09	yes	avgSLOC_with_bugs	commons-io	all
0.12	5.28E-59	yes	avgSLOC_with_bugs	commons-lang	all
0.18	1.55E-66	yes	avgSLOC_with_bugs	cucumber-jvm	all
0.12	4.60E-212	yes	avgSLOC_with_bugs	docx4j	all
0.22	0	yes	avgSLOC_with_bugs	drill	all
0.26	0	yes	avgSLOC_with_bugs	DSpace	all
0.3	1.51E-127	yes	avgSLOC_with_bugs	Essentials	all
0.19	1.31E-177	yes	avgSLOC_with_bugs	fastjson	all
0.2	0	yes	avgSLOC_with_bugs	flink	all
0.11	2.27E-21	yes	avgSLOC_with_bugs	flyway	all
0.13	1.52E-17	yes	avgSLOC_with_bugs	gson	all
0.22	0	yes	avgSLOC_with_bugs	guava	all
0.1	4.21E-13	yes	avgSLOC_with_bugs	hawtio	all
0.25	5.77E-57	yes	avgSLOC_with_bugs	hector	all
0.14	0	yes	avgSLOC_with_bugs	hibernate-orm	all
0.15	6.89E-118	yes	avgSLOC_with_bugs	hibernate-search	all
0.24	1.97E-84	yes	avgSLOC_with_bugs	Hystrix	all
0.11	2.77E-64	yes	avgSLOC_with_bugs	javaparser	all
0.3	0	yes	avgSLOC_with_bugs	jetty.project	all
0	0.5066555528	no	avgSLOC_with_bugs	jgit	all
0.18	3.43E-49	yes	avgSLOC_with_bugs	jna	all
0.11	1.11E-18	yes	avgSLOC_with_bugs	junit4	all
0.16	1.88E-58	yes	avgSLOC_with_bugs	junit5	all
0.15	2.56E-89	yes	avgSLOC_with_bugs	logging-log4j2	all
0.16	2.37E-47	yes	avgSLOC_with_bugs	lombok	all
0.1	4.22E-27	yes	avgSLOC_with_bugs	mockito	all
0.14	6.49E-102	yes	avgSLOC_with_bugs	mongo-java-driver	all
0.25	1.21E-148	yes	avgSLOC_with_bugs	mybatis-3	all
0.28	0	yes	avgSLOC_with_bugs	netty	all
0.28	1.80E-128	yes	avgSLOC_with_bugs	okhttp	all
0.14	2.67E-115	yes	avgSLOC_with_bugs	openmrs-core	all
0.21	0	yes	avgSLOC_with_bugs	pmd	all
0.09	2.25E-11	yes	avgSLOC_with_bugs	rabbitmq-java-client	all
0.18	4.23E-237	yes	avgSLOC_with_bugs	RxJava	all
0.16	0	yes	avgSLOC_with_bugs	sonarqube	all
0.05	2.01E-38	yes	avgSLOC_with_bugs	soot	all
0.09	0.000370811793	yes	avgSLOC_with_bugs	spark	all

0.15	5.99E-216	yes	avgSLOC_with_bugs	spring-boot	all
0.14	0	yes	avgSLOC_with_bugs	spring-framework	all
0.15	2.18E-45	yes	avgSLOC_with_bugs	swagger-core	all
0.25	6.34E-122	yes	avgSLOC_with_bugs	titan	all
0.19	1.39E-48	yes	avgSLOC_with_bugs	truth	all
0.32	2.00E-117	yes	avgSLOC_with_bugs	Twitter4J	all
0.22	8.37E-162	yes	avgSLOC_with_bugs	voldemort	all
0.18	1.60E-43	yes	avgSLOC_with_bugs	vraptor4	all
0.06	7.38E-39	yes	avgSLOC_with_bugs	wicket	all
0.19	0	yes	avgSLOC_with_bugs	wildfly	all
0.15	2.47E-146	yes	introSLOC_with_bugs	ant	all
-0.01	0.01884359591	yes	introSLOC_with_bugs	antlr4	all
0.07	5.27E-55	yes	introSLOC_with_bugs	assertj-core	all
0.04	0.000390335961	yes	introSLOC_with_bugs	astyanax	all
0.25	1.40E-108	yes	introSLOC_with_bugs	atmosphere	all
0.2	3.73E-157	yes	introSLOC_with_bugs	checkstyle	all
0.07	3.64E-07	yes	introSLOC_with_bugs	commons-io	all
0.1	1.33E-41	yes	introSLOC_with_bugs	commons-lang	all
0.15	5.61E-44	yes	introSLOC_with_bugs	cucumber-jvm	all
0.12	7.53E-200	yes	introSLOC_with_bugs	docx4j	all
0.19	0	yes	introSLOC_with_bugs	drill	all
0.23	0	yes	introSLOC_with_bugs	DSpace	all
0.3	1.28E-124	yes	introSLOC_with_bugs	Essentials	all
0.17	1.73E-149	yes	introSLOC_with_bugs	fastjson	all
0.18	0	yes	introSLOC_with_bugs	flink	all
0.08	1.11E-11	yes	introSLOC_with_bugs	flyway	all
0.1	4.62E-11	yes	introSLOC_with_bugs	gson	all
0.22	0	yes	introSLOC_with_bugs	guava	all
0.07	1.05E-07	yes	introSLOC_with_bugs	hawtio	all
0.2	4.04E-37	yes	introSLOC_with_bugs	hector	all
0.14	0	yes	introSLOC_with_bugs	hibernate-orm	all
0.13	6.44E-90	yes	introSLOC_with_bugs	hibernate-search	all
0.22	2.22E-70	yes	introSLOC_with_bugs	Hystrix	all
0.04	2.58E-08	yes	introSLOC_with_bugs	javaparser	all
0.28	0	yes	introSLOC_with_bugs	jetty.project	all
-0.01	0.05480137078	no	introSLOC_with_bugs	jgit	all
0.16	2.64E-37	yes	introSLOC_with_bugs	jna	all
0.11	9.45E-17	yes	introSLOC_with_bugs	junit4	all
0.14	2.78E-45	yes	introSLOC_with_bugs	junit5	all
0.13	4.21E-69	yes	introSLOC_with_bugs	logging-log4j2	all
0.14	6.98E-35	yes	introSLOC_with_bugs	lombok	all
0.1	2.50E-24	yes	introSLOC_with_bugs	mockito	all
0.11	7.68E-60	yes	introSLOC_with_bugs	mongo-java-driver	all
0.23	1.42E-134	yes	introSLOC_with_bugs	mybatis-3	all

0.25	0	yes	introSLOC_with_bugs	netty	all
0.26	4.40E-110	yes	introSLOC_with_bugs	okhttp	all
0.12	8.67E-91	yes	introSLOC_with_bugs	openmrs-core	all
0.19	9.64E-266	yes	introSLOC_with_bugs	pmd	all
0.09	3.33E-10	yes	introSLOC_with_bugs	rabbitmq-java-client	all
0.17	2.02E-214	yes	introSLOC_with_bugs	RxJava	all
0.15	0	yes	introSLOC_with_bugs	sonarqube	all
0.01	0.000682324966	yes	introSLOC_with_bugs	soot	all
0.08	0.001405124328	yes	introSLOC_with_bugs	spark	all
0.13	3.14E-159	yes	introSLOC_with_bugs	spring-boot	all
0.13	0	yes	introSLOC_with_bugs	spring-framework	all
0.14	2.92E-38	yes	introSLOC_with_bugs	swagger-core	all
0.23	3.44E-96	yes	introSLOC_with_bugs	titan	all
0.16	3.01E-33	yes	introSLOC_with_bugs	truth	all
0.31	7.93E-114	yes	introSLOC_with_bugs	Twitter4J	all
0.2	1.26E-125	yes	introSLOC_with_bugs	voldemort	all
0.17	2.98E-38	yes	introSLOC_with_bugs	vraptor4	all
0.03	2.93E-08	yes	introSLOC_with_bugs	wicket	all
0.16	0	yes	introSLOC_with_bugs	wildfly	all
0.16	2.84E-166	yes	lastSLOC_with_bugs	ant	all
0.01	0.253126823	no	lastSLOC_with_bugs	antlr4	all
0.07	1.32E-58	yes	lastSLOC_with_bugs	assertj-core	all
0.05	1.51E-05	yes	lastSLOC_with_bugs	astyanax	all
0.28	5.51E-139	yes	lastSLOC_with_bugs	atmosphere	all
0.2	5.76E-164	yes	lastSLOC_with_bugs	checkstyle	all
0.07	2.96E-07	yes	lastSLOC_with_bugs	commons-io	all
0.11	7.78E-48	yes	lastSLOC_with_bugs	commons-lang	all
0.17	5.73E-60	yes	lastSLOC_with_bugs	cucumber-jvm	all
0.12	3.53E-209	yes	lastSLOC_with_bugs	docx4j	all
0.21	0	yes	lastSLOC_with_bugs	drill	all
0.26	0	yes	lastSLOC_with_bugs	DSpace	all
0.3	2.88E-120	yes	lastSLOC_with_bugs	Essentials	all
0.18	1.64E-163	yes	lastSLOC_with_bugs	fastjson	all
0.19	0	yes	lastSLOC_with_bugs	flink	all
0.09	5.50E-15	yes	lastSLOC_with_bugs	flyway	all
0.12	1.48E-15	yes	lastSLOC_with_bugs	gson	all
0.22	0	yes	lastSLOC_with_bugs	guava	all
0.1	9.81E-12	yes	lastSLOC_with_bugs	hawtio	all
0.24	9.25E-53	yes	lastSLOC_with_bugs	hector	all
0.14	0	yes	lastSLOC_with_bugs	hibernate-orm	all
0.14	3.66E-106	yes	lastSLOC_with_bugs	hibernate-search	all
0.2	1.01E-56	yes	lastSLOC_with_bugs	Hystrix	all
0.09	1.75E-41	yes	lastSLOC_with_bugs	javaparser	all
0.3	0	yes	lastSLOC_with_bugs	jetty.project	all

-0.01	0.1494876777	no	lastSLOC_with_bugs	jgit	all
0.17	1.22E-44	yes	lastSLOC_with_bugs	jna	all
0.1	8.01E-15	yes	lastSLOC_with_bugs	junit4	all
0.14	5.31E-45	yes	lastSLOC_with_bugs	junit5	all
0.14	3.02E-79	yes	lastSLOC_with_bugs	logging-log4j2	all
0.16	6.31E-45	yes	lastSLOC_with_bugs	lombok	all
0.08	1.69E-18	yes	lastSLOC_with_bugs	mockito	all
0.13	5.00E-86	yes	lastSLOC_with_bugs	mongo-java-driver	all
0.24	3.39E-136	yes	lastSLOC_with_bugs	mybatis-3	all
0.26	0	yes	lastSLOC_with_bugs	netty	all
0.27	1.30E-117	yes	lastSLOC_with_bugs	okhttp	all
0.13	1.68E-100	yes	lastSLOC_with_bugs	openmrs-core	all
0.19	1.38E-249	yes	lastSLOC_with_bugs	pmd	all
0.09	9.39E-10	yes	lastSLOC_with_bugs	rabbitmq-java-client	all
0.16	2.67E-200	yes	lastSLOC_with_bugs	RxJava	all
0.16	0	yes	lastSLOC_with_bugs	sonarqube	all
0.05	7.08E-38	yes	lastSLOC_with_bugs	soot	all
0.08	0.002432677509	yes	lastSLOC_with_bugs	spark	all
0.14	4.26E-195	yes	lastSLOC_with_bugs	spring-boot	all
0.13	0	yes	lastSLOC_with_bugs	spring-framework	all
0.14	8.78E-38	yes	lastSLOC_with_bugs	swagger-core	all
0.22	3.84E-93	yes	lastSLOC_with_bugs	titan	all
0.19	2.60E-44	yes	lastSLOC_with_bugs	truth	all
0.28	1.21E-89	yes	lastSLOC_with_bugs	Twitter4J	all
0.22	4.23E-154	yes	lastSLOC_with_bugs	voldemort	all
0.16	2.88E-34	yes	lastSLOC_with_bugs	vraptor4	all
0.06	5.45E-36	yes	lastSLOC_with_bugs	wicket	all
0.19	0	yes	lastSLOC_with_bugs	wildfly	all
0.16	5.29E-171	yes	medianSLOC_with_bugs	ant	all
0.01	0.1562042841	no	medianSLOC_with_bugs	antlr4	all
0.08	2.37E-61	yes	medianSLOC_with_bugs	assertj-core	all
0.06	2.72E-07	yes	medianSLOC_with_bugs	astyanax	all
0.28	1.04E-143	yes	medianSLOC_with_bugs	atmosphere	all
0.2	1.19E-168	yes	medianSLOC_with_bugs	checkstyle	all
0.08	2.07E-08	yes	medianSLOC_with_bugs	commons-io	all
0.12	2.90E-54	yes	medianSLOC_with_bugs	commons-lang	all
0.17	4.57E-61	yes	medianSLOC_with_bugs	cucumber-jvm	all
0.12	7.77E-212	yes	medianSLOC_with_bugs	docx4j	all
0.22	0	yes	medianSLOC_with_bugs	drill	all
0.26	0	yes	medianSLOC_with_bugs	DSpace	all
0.31	1.01E-131	yes	medianSLOC_with_bugs	Essentials	all
0.18	2.16E-171	yes	medianSLOC_with_bugs	fastjson	all
0.19	0	yes	medianSLOC_with_bugs	flink	all
0.11	1.64E-20	yes	medianSLOC_with_bugs	flyway	all

0.13	6.39E-17	yes	medianSLOC_with_bugs	gson	all
0.22	0	yes	medianSLOC_with_bugs	guava	all
0.1	1.84E-12	yes	medianSLOC_with_bugs	hawtio	all
0.24	1.79E-53	yes	medianSLOC_with_bugs	hector	all
0.14	0	yes	medianSLOC_with_bugs	hibernate-orm	all
0.14	3.02E-111	yes	medianSLOC_with_bugs	hibernate-search	all
0.24	3.41E-83	yes	medianSLOC_with_bugs	Hystrix	all
0.08	2.36E-33	yes	medianSLOC_with_bugs	javaparser	all
0.3	0	yes	medianSLOC_with_bugs	jetty.project	all
-0.01	0.4268824595	no	medianSLOC_with_bugs	jgit	all
0.18	3.05E-47	yes	medianSLOC_with_bugs	jna	all
0.11	1.25E-16	yes	medianSLOC_with_bugs	junit4	all
0.15	5.62E-49	yes	medianSLOC_with_bugs	junit5	all
0.14	2.78E-82	yes	medianSLOC_with_bugs	logging-log4j2	all
0.16	1.59E-45	yes	medianSLOC_with_bugs	lombok	all
0.09	2.72E-20	yes	medianSLOC_with_bugs	mockito	all
0.14	1.02E-91	yes	medianSLOC_with_bugs	mongo-java-driver	all
0.24	5.38E-145	yes	medianSLOC_with_bugs	mybatis-3	all
0.27	0	yes	medianSLOC_with_bugs	netty	all
0.28	9.01E-126	yes	medianSLOC_with_bugs	okhttp	all
0.13	8.12E-108	yes	medianSLOC_with_bugs	openmrs-core	all
0.2	8.98E-299	yes	medianSLOC_with_bugs	pmd	all
0.09	8.27E-11	yes	medianSLOC_with_bugs	rabbitmq-java-client	all
0.17	4.68E-213	yes	medianSLOC_with_bugs	RxJava	all
0.16	0	yes	medianSLOC_with_bugs	sonarqube	all
0.04	1.85E-37	yes	medianSLOC_with_bugs	soot	all
0.09	0.000606625307	yes	medianSLOC_with_bugs	spark	all
0.15	4.15E-199	yes	medianSLOC_with_bugs	spring-boot	all
0.14	0	yes	medianSLOC_with_bugs	spring-framework	all
0.15	2.88E-44	yes	medianSLOC_with_bugs	swagger-core	all
0.24	2.36E-108	yes	medianSLOC_with_bugs	titan	all
0.19	1.06E-46	yes	medianSLOC_with_bugs	truth	all
0.31	1.85E-111	yes	medianSLOC_with_bugs	Twitter4J	all
0.22	3.72E-158	yes	medianSLOC_with_bugs	voldemort	all
0.17	2.48E-39	yes	medianSLOC_with_bugs	vraptor4	all
0.05	4.19E-32	yes	medianSLOC_with_bugs	wicket	all
0.19	0	yes	medianSLOC_with_bugs	wildfly	all
0.26	0	yes	stdSLOC_with_bugs	ant	all
0.55	0	yes	stdSLOC_with_bugs	antlr4	all
0.25	0	yes	stdSLOC_with_bugs	assertj-core	all
0.44	2.13E-275	yes	stdSLOC_with_bugs	astyanax	all
0.58	0	yes	stdSLOC_with_bugs	atmosphere	all
0.32	0	yes	stdSLOC_with_bugs	checkstyle	all
0.22	1.34E-39	yes	stdSLOC_with_bugs	commons-io	all

0.37	0	yes	stdSLOC_with_bugs	commons-lang	all
0.39	2.41E-237	yes	stdSLOC_with_bugs	cucumber-jvm	all
0.32	0	yes	stdSLOC_with_bugs	docx4j	all
0.45	0	yes	stdSLOC_with_bugs	drill	all
0.36	0	yes	stdSLOC_with_bugs	DSpace	all
0.5	7.93E-286	yes	stdSLOC_with_bugs	Essentials	all
0.41	0	yes	stdSLOC_with_bugs	fastjson	all
0.46	0	yes	stdSLOC_with_bugs	flink	all
0.48	4.23E-280	yes	stdSLOC_with_bugs	flyway	all
0.27	6.69E-53	yes	stdSLOC_with_bugs	gson	all
0.19	0	yes	stdSLOC_with_bugs	guava	all
0.48	2.00E-205	yes	stdSLOC_with_bugs	hawtio	all
0.49	3.03E-179	yes	stdSLOC_with_bugs	hector	all
0.22	0	yes	stdSLOC_with_bugs	hibernate-orm	all
0.26	4.82E-302	yes	stdSLOC_with_bugs	hibernate-search	all
0.49	1.47E-272	yes	stdSLOC_with_bugs	Hystrix	all
0.35	0	yes	stdSLOC_with_bugs	javaparser	all
0.37	0	yes	stdSLOC_with_bugs	jetty.project	all
0.16	8.15E-93	yes	stdSLOC_with_bugs	jgit	all
0.35	7.47E-136	yes	stdSLOC_with_bugs	jna	all
0.36	8.35E-138	yes	stdSLOC_with_bugs	junit4	all
0.3	6.25E-162	yes	stdSLOC_with_bugs	junit5	all
0.33	0	yes	stdSLOC_with_bugs	logging-log4j2	all
0.39	4.27E-212	yes	stdSLOC_with_bugs	lombok	all
0.42	0	yes	stdSLOC_with_bugs	mockito	all
0.36	0	yes	stdSLOC_with_bugs	mongo-java-driver	all
0.4	0	yes	stdSLOC_with_bugs	mybatis-3	all
0.47	0	yes	stdSLOC_with_bugs	netty	all
0.46	2.56E-274	yes	stdSLOC_with_bugs	okhttp	all
0.34	0	yes	stdSLOC_with_bugs	openmrs-core	all
0.48	0	yes	stdSLOC_with_bugs	pmd	all
0.26	5.36E-61	yes	stdSLOC_with_bugs	rabbitmq-java-client	all
0.43	0	yes	stdSLOC_with_bugs	RxJava	all
0.33	0	yes	stdSLOC_with_bugs	sonarqube	all
0.28	0	yes	stdSLOC_with_bugs	soot	all
0.41	3.78E-47	yes	stdSLOC_with_bugs	spark	all
0.33	0	yes	stdSLOC_with_bugs	spring-boot	all
0.28	0	yes	stdSLOC_with_bugs	spring-framework	all
0.43	4.65E-296	yes	stdSLOC_with_bugs	swagger-core	all
0.54	0	yes	stdSLOC_with_bugs	titan	all
0.35	2.03E-127	yes	stdSLOC_with_bugs	truth	all
0.45	7.72E-198	yes	stdSLOC_with_bugs	Twitter4J	all
0.47	0	yes	stdSLOC_with_bugs	voldemort	all
0.46	1.06E-213	yes	stdSLOC_with_bugs	vraptor4	all

0.4	0	yes	stdSLOC_with_bugs	wicket	all
0.45	0	yes	stdSLOC_with_bugs	wildfly	all
0.03	1.65E-06	yes	versionedSLOC_with_bugs	antlr4	all
0.14	4.46E-188	yes	versionedSLOC_with_bugs	assertj-core	all
0.01	0.46728181	no	versionedSLOC_with_bugs	astyanax	all
0.26	2.58E-167	yes	versionedSLOC_with_bugs	atmosphere	all
0.19	0	yes	versionedSLOC_with_bugs	drill	all
0.12	1.23E-93	yes	versionedSLOC_with_bugs	DSpace	all
0.21	9.77E-90	yes	versionedSLOC_with_bugs	Essentials	all
0.14	0	yes	versionedSLOC_with_bugs	flink	all
0.21	4.56E-72	yes	versionedSLOC_with_bugs	flyway	all
0.19	0	yes	versionedSLOC_with_bugs	guava	all
0.11	1.97E-16	yes	versionedSLOC_with_bugs	hawtio	all
0.19	7.31E-43	yes	versionedSLOC_with_bugs	hector	all
0.11	1.09E-268	yes	versionedSLOC_with_bugs	hibernate-orm	all
0.18	1.80E-62	yes	versionedSLOC_with_bugs	Hystrix	all
0.21	0	yes	versionedSLOC_with_bugs	jetty.project	all
0.09	9.45E-47	yes	versionedSLOC_with_bugs	logging-log4j2	all
0.15	2.29E-44	yes	versionedSLOC_with_bugs	lombok	all
0.08	5.81E-37	yes	versionedSLOC_with_bugs	mongo-java-driver	all
0.24	2.04E-106	yes	versionedSLOC_with_bugs	mybatis-3	all
0.23	0	yes	versionedSLOC_with_bugs	netty	all
0.1	1.99E-74	yes	versionedSLOC_with_bugs	openmrs-core	all
0.05	0.000863834142	yes	versionedSLOC_with_bugs	rabbitmq-java-client	all
0.12	4.40E-106	yes	versionedSLOC_with_bugs	RxJava	all
0.12	0	yes	versionedSLOC_with_bugs	sonarqube	all
0.21	0	yes	versionedSLOC_with_bugs	soot	all
0.08	0.000481537290	yes	versionedSLOC_with_bugs	spark	all
0.1	7.36E-189	yes	versionedSLOC_with_bugs	spring-framework	all
0.15	3.33E-58	yes	versionedSLOC_with_bugs	swagger-core	all
0.1	1.96E-16	yes	versionedSLOC_with_bugs	truth	all
0.17	2.85E-41	yes	versionedSLOC_with_bugs	Twitter4J	all
0.15	2.90E-93	yes	versionedSLOC_with_bugs	voldemort	all
0.14	3.51E-31	yes	versionedSLOC_with_bugs	vraptor4	all
0.17	0	yes	versionedSLOC_with_bugs	wildfly	all
0.18	7.32E-211	yes	versionedSLOC_with_bugs	ant	all
0.19	1.48E-144	yes	versionedSLOC_with_bugs	checkstyle	all
0.04	0.006134954696	yes	versionedSLOC_with_bugs	commons-io	all
0.11	3.81E-49	yes	versionedSLOC_with_bugs	commons-lang	all
0.1	1.44E-21	yes	versionedSLOC_with_bugs	cucumber-jvm	all
0.14	5.30E-210	yes	versionedSLOC_with_bugs	docx4j	all
0.2	1.18E-202	yes	versionedSLOC_with_bugs	fastjson	all
0.11	9.46E-13	yes	versionedSLOC_with_bugs	gson	all
0.1	2.19E-74	yes	versionedSLOC_with_bugs	hibernate-search	all

0.09	5.42E-60	yes	versionedSLOC_with_bugs	javaparser	all
0.11	3.06E-50	yes	versionedSLOC_with_bugs	jgit	all
0.15	1.53E-34	yes	versionedSLOC_with_bugs	jna	all
0.09	6.54E-12	yes	versionedSLOC_with_bugs	junit4	all
0.09	9.82E-27	yes	versionedSLOC_with_bugs	junit5	all
0.08	7.57E-17	yes	versionedSLOC_with_bugs	mockito	all
0.14	5.42E-47	yes	versionedSLOC_with_bugs	okhttp	all
0.17	3.19E-232	yes	versionedSLOC_with_bugs	pmd	all
0.09	1.66E-85	yes	versionedSLOC_with_bugs	spring-boot	all
0.21	1.69E-103	yes	versionedSLOC_with_bugs	titan	all
0.04	4.90E-21	yes	versionedSLOC_with_bugs	wicket	all

B.3 SLOC measurements with #Revisions (modified once)

Kendall τ correlation coefficient of all 53 projects for each SLOC measurements with #Revisions is given below. Methods that have been modified at least once in each project were used to compute the correlation values.

corr	p_value	significant	correlation of	repo	Methods considered
0.3	6.63E-107	yes	avgSLOC_with_revisions	antlr4	modified at least once
0.08	1.13E-35	yes	avgSLOC_with_revisions	assertj-core	modified at least once
0.17	7.19E-39	yes	avgSLOC_with_revisions	astyanax	modified at least once
0.36	1.21E-98	yes	avgSLOC_with_revisions	atmosphere	modified at least once
0.24	0	yes	avgSLOC_with_revisions	drill	modified at least once
0.29	0	yes	avgSLOC_with_revisions	DSpace	modified at least once
0.38	1.54E-157	yes	avgSLOC_with_revisions	Essentials	modified at least once
0.3	0	yes	avgSLOC_with_revisions	flink	modified at least once
0.29	2.21E-52	yes	avgSLOC_with_revisions	flyway	modified at least once
0.04	2.53E-12	yes	avgSLOC_with_revisions	guava	modified at least once
0.29	1.87E-34	yes	avgSLOC_with_revisions	hawtio	modified at least once
0.31	3.91E-54	yes	avgSLOC_with_revisions	hector	modified at least once
0.1	2.59E-141	yes	avgSLOC_with_revisions	hibernate-orm	modified at least once
0.37	6.69E-93	yes	avgSLOC_with_revisions	Hystrix	modified at least once
0.34	0	yes	avgSLOC_with_revisions	jetty.project	modified at least once
0.25	8.22E-224	yes	avgSLOC_with_revisions	logging-log4j2	modified at least once
0.23	3.53E-63	yes	avgSLOC_with_revisions	lombok	modified at least once
0.19	1.31E-141	yes	avgSLOC_with_revisions	mongo-java-driver	modified at least once
0.31	3.55E-184	yes	avgSLOC_with_revisions	mybatis-3	modified at least once
0.26	0	yes	avgSLOC_with_revisions	netty	modified at least once
0.29	0	yes	avgSLOC_with_revisions	openmrs-core	modified at least once
0.24	2.03E-50	yes	avgSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.08	2.40E-19	yes	avgSLOC_with_revisions	RxJava	modified at least once
0.34	0	yes	avgSLOC_with_revisions	sonarqube	modified at least once
0.3	0	yes	avgSLOC_with_revisions	soot	modified at least once
0.17	0.000193680753	yes	avgSLOC_with_revisions	spark	modified at least once
0.24	0	yes	avgSLOC_with_revisions	spring-framework	modified at least once
0.17	1.56E-23	yes	avgSLOC_with_revisions	swagger-core	modified at least once
0.16	1.14E-28	yes	avgSLOC_with_revisions	truth	modified at least once
0.21	1.71E-47	yes	avgSLOC_with_revisions	Twitter4J	modified at least once
0.3	4.22E-157	yes	avgSLOC_with_revisions	voldemort	modified at least once
0.36	2.13E-87	yes	avgSLOC_with_revisions	vraptor4	modified at least once
0.29	0	yes	avgSLOC_with_revisions	wildfly	modified at least once
0.27	2.16E-293	yes	avgSLOC_with_revisions	ant	modified at least once
0.26	8.26E-251	yes	avgSLOC_with_revisions	checkstyle	modified at least once
0.12	1.62E-15	yes	avgSLOC_with_revisions	commons-io	modified at least once
0.12	1.96E-36	yes	avgSLOC_with_revisions	commons-lang	modified at least once
0.24	8.79E-103	yes	avgSLOC_with_revisions	cucumber-jvm	modified at least once
0.03	3.29E-07	yes	avgSLOC_with_revisions	docx4j	modified at least once
0.33	1.88E-114	yes	avgSLOC_with_revisions	fastjson	modified at least once
0.1	7.43E-07	yes	avgSLOC_with_revisions	gson	modified at least once
0.25	0	yes	avgSLOC_with_revisions	hibernate-search	modified at least once
0.19	7.55E-181	yes	avgSLOC_with_revisions	javaparser	modified at least once

0.29	3.09E-270	yes	avgSLOC_with_revisions	jgit	modified at least once
0.07	1.87E-07	yes	avgSLOC_with_revisions	jna	modified at least once
0.19	8.97E-40	yes	avgSLOC_with_revisions	junit4	modified at least once
0.2	1.51E-82	yes	avgSLOC_with_revisions	junit5	modified at least once
0.05	3.42E-07	yes	avgSLOC_with_revisions	mockito	modified at least once
0.21	2.70E-75	yes	avgSLOC_with_revisions	okhttp	modified at least once
0.24	0	yes	avgSLOC_with_revisions	pmd	modified at least once
0.26	0	yes	avgSLOC_with_revisions	spring-boot	modified at least once
0.37	2.55E-87	yes	avgSLOC_with_revisions	titan	modified at least once
0.09	5.84E-58	yes	avgSLOC_with_revisions	wicket	modified at least once
0.26	2.29E-78	yes	introSLOC_with_revisions	antlr4	modified at least once
0.08	6.90E-31	yes	introSLOC_with_revisions	assertj-core	modified at least once
0.13	8.07E-23	yes	introSLOC_with_revisions	astyanax	modified at least once
0.29	2.43E-61	yes	introSLOC_with_revisions	atmosphere	modified at least once
0.19	8.27E-242	yes	introSLOC_with_revisions	drill	modified at least once
0.25	4.79E-291	yes	introSLOC_with_revisions	DSpace	modified at least once
0.35	4.54E-125	yes	introSLOC_with_revisions	Essentials	modified at least once
0.27	0	yes	introSLOC_with_revisions	flink	modified at least once
0.24	2.47E-36	yes	introSLOC_with_revisions	flyway	modified at least once
0.02	1.36E-06	yes	introSLOC_with_revisions	guava	modified at least once
0.26	3.13E-27	yes	introSLOC_with_revisions	hawtio	modified at least once
0.23	1.04E-29	yes	introSLOC_with_revisions	hector	modified at least once
0.08	3.13E-100	yes	introSLOC_with_revisions	hibernate-orm	modified at least once
0.34	4.78E-76	yes	introSLOC_with_revisions	Hystrix	modified at least once
0.29	0	yes	introSLOC_with_revisions	jetty.project	modified at least once
0.22	2.05E-174	yes	introSLOC_with_revisions	logging-log4j2	modified at least once
0.19	2.09E-45	yes	introSLOC_with_revisions	lombok	modified at least once
0.15	1.59E-79	yes	introSLOC_with_revisions	mongo-java-driver	modified at least once
0.32	3.59E-192	yes	introSLOC_with_revisions	mybatis-3	modified at least once
0.22	1.44E-259	yes	introSLOC_with_revisions	netty	modified at least once
0.28	1.03E-307	yes	introSLOC_with_revisions	openmrs-core	modified at least once
0.22	1.46E-41	yes	introSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.08	7.43E-20	yes	introSLOC_with_revisions	RxJava	modified at least once
0.32	0	yes	introSLOC_with_revisions	sonarqube	modified at least once
0.25	0	yes	introSLOC_with_revisions	soot	modified at least once
0.16	0.000695161628	yes	introSLOC_with_revisions	spark	modified at least once
0.21	0	yes	introSLOC_with_revisions	spring-framework	modified at least once
0.12	5.51E-12	yes	introSLOC_with_revisions	swagger-core	modified at least once
0.13	1.52E-18	yes	introSLOC_with_revisions	truth	modified at least once
0.18	1.59E-35	yes	introSLOC_with_revisions	Twitter4J	modified at least once
0.25	2.02E-111	yes	introSLOC_with_revisions	voldemort	modified at least once
0.33	2.98E-71	yes	introSLOC_with_revisions	vraptor4	modified at least once
0.24	0	yes	introSLOC_with_revisions	wildfly	modified at least once
0.23	9.61E-225	yes	introSLOC_with_revisions	ant	modified at least once

0.26	8.51E-240	yes	introSLOC_with_revisions	checkstyle	modified at least once
0.1	3.17E-11	yes	introSLOC_with_revisions	commons-io	modified at least once
0.1	1.63E-26	yes	introSLOC_with_revisions	commons-lang	modified at least once
0.21	9.07E-73	yes	introSLOC_with_revisions	cucumber-jvm	modified at least once
0.02	0.001297099	yes	introSLOC_with_revisions	docx4j	modified at least once
0.28	3.38E-82	yes	introSLOC_with_revisions	fastjson	modified at least once
0.06	0.003771805014	yes	introSLOC_with_revisions	gson	modified at least once
0.22	1.29E-282	yes	introSLOC_with_revisions	hibernate-search	modified at least once
0.12	1.85E-64	yes	introSLOC_with_revisions	javaparser	modified at least once
0.27	7.39E-222	yes	introSLOC_with_revisions	jgit	modified at least once
0.05	5.24E-05	yes	introSLOC_with_revisions	jna	modified at least once
0.17	6.59E-30	yes	introSLOC_with_revisions	junit4	modified at least once
0.18	8.75E-66	yes	introSLOC_with_revisions	junit5	modified at least once
0.05	4.89E-06	yes	introSLOC_with_revisions	mockito	modified at least once
0.18	2.83E-54	yes	introSLOC_with_revisions	okhttp	modified at least once
0.22	1.19E-262	yes	introSLOC_with_revisions	pmd	modified at least once
0.24	0	yes	introSLOC_with_revisions	spring-boot	modified at least once
0.3	4.92E-57	yes	introSLOC_with_revisions	titan	modified at least once
0.05	1.15E-19	yes	introSLOC_with_revisions	wicket	modified at least once
0.29	1.39E-97	yes	lastSLOC_with_revisions	antlr4	modified at least once
0.07	4.30E-29	yes	lastSLOC_with_revisions	assertj-core	modified at least once
0.13	2.99E-23	yes	lastSLOC_with_revisions	astyanax	modified at least once
0.34	9.30E-87	yes	lastSLOC_with_revisions	atmosphere	modified at least once
0.23	0	yes	lastSLOC_with_revisions	drill	modified at least once
0.28	0	yes	lastSLOC_with_revisions	DSpace	modified at least once
0.36	1.85E-138	yes	lastSLOC_with_revisions	Essentials	modified at least once
0.28	0	yes	lastSLOC_with_revisions	flink	modified at least once
0.26	3.08E-40	yes	lastSLOC_with_revisions	flyway	modified at least once
0.03	1.36E-07	yes	lastSLOC_with_revisions	guava	modified at least once
0.28	1.58E-31	yes	lastSLOC_with_revisions	hawtio	modified at least once
0.25	2.27E-35	yes	lastSLOC_with_revisions	hector	modified at least once
0.1	2.09E-132	yes	lastSLOC_with_revisions	hibernate-orm	modified at least once
0.29	1.83E-57	yes	lastSLOC_with_revisions	Hystrix	modified at least once
0.33	0	yes	lastSLOC_with_revisions	jetty.project	modified at least once
0.24	1.93E-202	yes	lastSLOC_with_revisions	logging-log4j2	modified at least once
0.22	1.19E-59	yes	lastSLOC_with_revisions	lombok	modified at least once
0.18	3.94E-116	yes	lastSLOC_with_revisions	mongo-java-driver	modified at least once
0.3	2.17E-171	yes	lastSLOC_with_revisions	mybatis-3	modified at least once
0.24	3.52E-291	yes	lastSLOC_with_revisions	netty	modified at least once
0.26	8.78E-272	yes	lastSLOC_with_revisions	openmrs-core	modified at least once
0.22	4.33E-44	yes	lastSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.05	1.82E-09	yes	lastSLOC_with_revisions	RxJava	modified at least once
0.33	0	yes	lastSLOC_with_revisions	sonarqube	modified at least once
0.29	0	yes	lastSLOC_with_revisions	soot	modified at least once

0.14	0.001679939373	yes	lastSLOC_with_revisions	spark	modified at least once
0.23	0	yes	lastSLOC_with_revisions	spring-framework	modified at least once
0.16	1.43E-20	yes	lastSLOC_with_revisions	swagger-core	modified at least once
0.15	1.33E-24	yes	lastSLOC_with_revisions	truth	modified at least once
0.18	3.05E-35	yes	lastSLOC_with_revisions	Twitter4J	modified at least once
0.29	1.59E-150	yes	lastSLOC_with_revisions	voldemort	modified at least once
0.34	8.87E-75	yes	lastSLOC_with_revisions	vraptor4	modified at least once
0.28	0	yes	lastSLOC_with_revisions	wildfly	modified at least once
0.25	2.16E-251	yes	lastSLOC_with_revisions	ant	modified at least once
0.25	1.97E-232	yes	lastSLOC_with_revisions	checkstyle	modified at least once
0.11	3.12E-13	yes	lastSLOC_with_revisions	commons-io	modified at least once
0.1	1.65E-27	yes	lastSLOC_with_revisions	commons-lang	modified at least once
0.23	1.41E-90	yes	lastSLOC_with_revisions	cucumber-jvm	modified at least once
0.02	5.19E-06	yes	lastSLOC_with_revisions	docx4j	modified at least once
0.3	2.80E-95	yes	lastSLOC_with_revisions	fastjson	modified at least once
0.07	0.000469624222	yes	lastSLOC_with_revisions	gson	modified at least once
0.23	0.00E-02	yes	lastSLOC_with_revisions	hibernate-search	modified at least once
0.18	1.56E-153	yes	lastSLOC_with_revisions	javaparser	modified at least once
0.28	2.51E-252	yes	lastSLOC_with_revisions	jgit	modified at least once
0.05	0.000806116086	yes	lastSLOC_with_revisions	jna	modified at least once
0.18	2.78E-35	yes	lastSLOC_with_revisions	junit4	modified at least once
0.17	2.23E-62	yes	lastSLOC_with_revisions	junit5	modified at least once
0.03	0.00929227103	yes	lastSLOC_with_revisions	mockito	modified at least once
0.2	3.81E-65	yes	lastSLOC_with_revisions	okhttp	modified at least once
0.2	2.72E-223	yes	lastSLOC_with_revisions	pmd	modified at least once
0.25	0	yes	lastSLOC_with_revisions	spring-boot	modified at least once
0.32	1.16E-64	yes	lastSLOC_with_revisions	titan	modified at least once
0.09	1.11E-54	yes	lastSLOC_with_revisions	wicket	modified at least once
0.3	2.96E-104	yes	medianSLOC_with_revisions	antlr4	modified at least once
0.08	8.33E-32	yes	medianSLOC_with_revisions	assertj-core	modified at least once
0.15	4.69E-28	yes	medianSLOC_with_revisions	astyanax	modified at least once
0.35	8.64E-91	yes	medianSLOC_with_revisions	atmosphere	modified at least once
0.23	0	yes	medianSLOC_with_revisions	drill	modified at least once
0.28	0	yes	medianSLOC_with_revisions	DSpace	modified at least once
0.4	4.45E-164	yes	medianSLOC_with_revisions	Essentials	modified at least once
0.29	0	yes	medianSLOC_with_revisions	flink	modified at least once
0.29	8.36E-51	yes	medianSLOC_with_revisions	flyway	modified at least once
0.03	5.77E-10	yes	medianSLOC_with_revisions	guava	modified at least once
0.29	1.99E-33	yes	medianSLOC_with_revisions	hawtio	modified at least once
0.31	5.24E-52	yes	medianSLOC_with_revisions	hector	modified at least once
0.1	4.93E-133	yes	medianSLOC_with_revisions	hibernate-orm	modified at least once
0.37	4.60E-92	yes	medianSLOC_with_revisions	Hystrix	modified at least once
0.34	0	yes	medianSLOC_with_revisions	jetty.project	modified at least once
0.24	2.46E-208	yes	medianSLOC_with_revisions	logging-log4j2	modified at least once

0.22	9.54E-60	yes	medianSLOC_with_revisions	lombok	modified at least once
0.18	1.35E-118	yes	medianSLOC_with_revisions	mongo-java-driver	modified at least once
0.31	1.35E-178	yes	medianSLOC_with_revisions	mybatis-3	modified at least once
0.25	0	yes	medianSLOC_with_revisions	netty	modified at least once
0.28	0	yes	medianSLOC_with_revisions	openmrs-core	modified at least once
0.23	1.80E-46	yes	medianSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.07	1.01E-14	yes	medianSLOC_with_revisions	RxJava	modified at least once
0.34	0	yes	medianSLOC_with_revisions	sonarqube	modified at least once
0.3	0	yes	medianSLOC_with_revisions	soot	modified at least once
0.16	0.000351253954	yes	medianSLOC_with_revisions	spark	modified at least once
0.23	0	yes	medianSLOC_with_revisions	spring-framework	modified at least once
0.16	2.45E-22	yes	medianSLOC_with_revisions	swagger-core	modified at least once
0.15	3.44E-26	yes	medianSLOC_with_revisions	truth	modified at least once
0.2	2.70E-41	yes	medianSLOC_with_revisions	Twitter4J	modified at least once
0.29	1.87E-151	yes	medianSLOC_with_revisions	voldemort	modified at least once
0.35	1.40E-80	yes	medianSLOC_with_revisions	vraptor4	modified at least once
0.28	0	yes	medianSLOC_with_revisions	wildfly	modified at least once
0.26	2.71E-275	yes	medianSLOC_with_revisions	ant	modified at least once
0.26	2.15E-244	yes	medianSLOC_with_revisions	checkstyle	modified at least once
0.12	5.58E-15	yes	medianSLOC_with_revisions	commons-io	modified at least once
0.11	1.43E-31	yes	medianSLOC_with_revisions	commons-lang	modified at least once
0.24	1.18E-96	yes	medianSLOC_with_revisions	cucumber-jvm	modified at least once
0.03	1.71E-06	yes	medianSLOC_with_revisions	docx4j	modified at least once
0.32	7.21E-110	yes	medianSLOC_with_revisions	fastjson	modified at least once
0.1	2.87E-06	yes	medianSLOC_with_revisions	gson	modified at least once
0.24	0	yes	medianSLOC_with_revisions	hibernate-search	modified at least once
0.18	5.88E-155	yes	medianSLOC_with_revisions	javaparser	modified at least once
0.29	4.27E-264	yes	medianSLOC_with_revisions	jgit	modified at least once
0.06	2.97E-06	yes	medianSLOC_with_revisions	jna	modified at least once
0.18	9.82E-36	yes	medianSLOC_with_revisions	junit4	modified at least once
0.18	1.19E-65	yes	medianSLOC_with_revisions	junit5	modified at least once
0.04	0.000163045256	yes	medianSLOC_with_revisions	mockito	modified at least once
0.21	2.00E-75	yes	medianSLOC_with_revisions	okhttp	modified at least once
0.21	1.45E-262	yes	medianSLOC_with_revisions	pmd	modified at least once
0.26	0	yes	medianSLOC_with_revisions	spring-boot	modified at least once
0.35	5.01E-77	yes	medianSLOC_with_revisions	titan	modified at least once
0.08	1.65E-47	yes	medianSLOC_with_revisions	wicket	modified at least once
0.53	7.14E-253	yes	stdSLOC_with_revisions	antlr4	modified at least once
0.34	0	yes	stdSLOC_with_revisions	assertj-core	modified at least once
0.52	5.41E-280	yes	stdSLOC_with_revisions	astyanax	modified at least once
0.6	6.15E-225	yes	stdSLOC_with_revisions	atmosphere	modified at least once
0.55	0	yes	stdSLOC_with_revisions	drill	modified at least once
0.53	0	yes	stdSLOC_with_revisions	DSpace	modified at least once
0.63	0	yes	stdSLOC_with_revisions	Essentials	modified at least once

0.52	0	yes	stdSLOC_with_revisions	flink	modified at least once
0.58	1.07E-167	yes	stdSLOC_with_revisions	flyway	modified at least once
0.39	0	yes	stdSLOC_with_revisions	guava	modified at least once
0.53	1.12E-88	yes	stdSLOC_with_revisions	hawtio	modified at least once
0.58	1.79E-151	yes	stdSLOC_with_revisions	hector	modified at least once
0.29	0	yes	stdSLOC_with_revisions	hibernate-orm	modified at least once
0.61	3.02E-211	yes	stdSLOC_with_revisions	Hystrix	modified at least once
0.57	0	yes	stdSLOC_with_revisions	jetty.project	modified at least once
0.43	0	yes	stdSLOC_with_revisions	logging-log4j2	modified at least once
0.47	1.39E-211	yes	stdSLOC_with_revisions	lombok	modified at least once
0.48	0	yes	stdSLOC_with_revisions	mongo-java-driver	modified at least once
0.42	2.49E-259	yes	stdSLOC_with_revisions	mybatis-3	modified at least once
0.53	0	yes	stdSLOC_with_revisions	netty	modified at least once
0.46	0	yes	stdSLOC_with_revisions	openmrs-core	modified at least once
0.52	2.64E-187	yes	stdSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.38	0	yes	stdSLOC_with_revisions	RxJava	modified at least once
0.52	0	yes	stdSLOC_with_revisions	sonarqube	modified at least once
0.5	0	yes	stdSLOC_with_revisions	soot	modified at least once
0.41	1.10E-16	yes	stdSLOC_with_revisions	spark	modified at least once
0.46	0	yes	stdSLOC_with_revisions	spring-framework	modified at least once
0.45	7.56E-137	yes	stdSLOC_with_revisions	swagger-core	modified at least once
0.42	3.46E-154	yes	stdSLOC_with_revisions	truth	modified at least once
0.49	1.57E-206	yes	stdSLOC_with_revisions	Twitter4J	modified at least once
0.62	0	yes	stdSLOC_with_revisions	voldemort	modified at least once
0.58	3.83E-185	yes	stdSLOC_with_revisions	vraptor4	modified at least once
0.57	0	yes	stdSLOC_with_revisions	wildfly	modified at least once
0.48	0	yes	stdSLOC_with_revisions	ant	modified at least once
0.4	0	yes	stdSLOC_with_revisions	checkstyle	modified at least once
0.42	2.56E-125	yes	stdSLOC_with_revisions	commons-io	modified at least once
0.39	4.73E-298	yes	stdSLOC_with_revisions	commons-lang	modified at least once
0.5	0	yes	stdSLOC_with_revisions	cucumber-jvm	modified at least once
0.24	0	yes	stdSLOC_with_revisions	docx4j	modified at least once
0.61	2.80E-305	yes	stdSLOC_with_revisions	fastjson	modified at least once
0.49	4.76E-95	yes	stdSLOC_with_revisions	gson	modified at least once
0.46	0	yes	stdSLOC_with_revisions	hibernate-search	modified at least once
0.43	0	yes	stdSLOC_with_revisions	javaparser	modified at least once
0.54	0	yes	stdSLOC_with_revisions	jgit	modified at least once
0.43	1.40E-171	yes	stdSLOC_with_revisions	jna	modified at least once
0.4	8.80E-131	yes	stdSLOC_with_revisions	junit4	modified at least once
0.4	4.84E-280	yes	stdSLOC_with_revisions	junit5	modified at least once
0.5	0	yes	stdSLOC_with_revisions	mockito	modified at least once
0.52	0	yes	stdSLOC_with_revisions	okhttp	modified at least once
0.51	0	yes	stdSLOC_with_revisions	pmd	modified at least once
0.43	0	yes	stdSLOC_with_revisions	spring-boot	modified at least once

0.53	5.54E-155	yes	stdSLOC_with_revisions	titan	modified at least once
0.39	0	yes	stdSLOC_with_revisions	wicket	modified at least once
0.04	0.000370853906	yes	versionedSLOC_with_revisions	antlr4	modified at least once
0.02	0.002272386591	yes	versionedSLOC_with_revisions	assertj-core	modified at least once
-0.13	3.18E-28	yes	versionedSLOC_with_revisions	astyanax	modified at least once
0.04	0.001660802317	yes	versionedSLOC_with_revisions	atmosphere	modified at least once
0.03	9.07E-10	yes	versionedSLOC_with_revisions	drill	modified at least once
0.02	0.001189692679	yes	versionedSLOC_with_revisions	DSpace	modified at least once
0.15	6.81E-38	yes	versionedSLOC_with_revisions	Essentials	modified at least once
0.07	2.82E-54	yes	versionedSLOC_with_revisions	flink	modified at least once
0	0.9730048965	no	versionedSLOC_with_revisions	flyway	modified at least once
-0.03	2.62E-09	yes	versionedSLOC_with_revisions	guava	modified at least once
0.09	0.000119379895	yes	versionedSLOC_with_revisions	hawtio	modified at least once
0.1	2.27E-09	yes	versionedSLOC_with_revisions	hector	modified at least once
0.05	1.43E-43	yes	versionedSLOC_with_revisions	hibernate-orm	modified at least once
0.07	1.75E-07	yes	versionedSLOC_with_revisions	Hystrix	modified at least once
0.09	6.03E-57	yes	versionedSLOC_with_revisions	jetty.project	modified at least once
0.09	1.40E-39	yes	versionedSLOC_with_revisions	logging-log4j2	modified at least once
0	0.7356570302	no	versionedSLOC_with_revisions	lombok	modified at least once
-0.04	2.32E-11	yes	versionedSLOC_with_revisions	mongo-java-driver	modified at least once
0.1	1.57E-13	yes	versionedSLOC_with_revisions	mybatis-3	modified at least once
0.06	1.04E-27	yes	versionedSLOC_with_revisions	netty	modified at least once
0.01	0.209322657	no	versionedSLOC_with_revisions	openmrs-core	modified at least once
0.02	0.2080410994	no	versionedSLOC_with_revisions	rabbitmq-java-client	modified at least once
0.03	4.78E-05	yes	versionedSLOC_with_revisions	RxJava	modified at least once
0.09	2.32E-133	yes	versionedSLOC_with_revisions	sonarqube	modified at least once
0.11	2.09E-55	yes	versionedSLOC_with_revisions	soot	modified at least once
0.05	0.1670061293	no	versionedSLOC_with_revisions	spark	modified at least once
0.01	3.00E-05	yes	versionedSLOC_with_revisions	spring-framework	modified at least once
-0.15	1.60E-30	yes	versionedSLOC_with_revisions	swagger-core	modified at least once
-0.1	5.20E-14	yes	versionedSLOC_with_revisions	truth	modified at least once
-0.03	0.01683852192	yes	versionedSLOC_with_revisions	Twitter4J	modified at least once
-0.02	0.007963438321	yes	versionedSLOC_with_revisions	voldemort	modified at least once
0.17	5.20E-28	yes	versionedSLOC_with_revisions	vraptor4	modified at least once
0.05	1.12E-31	yes	versionedSLOC_with_revisions	wildfly	modified at least once
0.1	1.59E-37	yes	versionedSLOC_with_revisions	ant	modified at least once
0.11	3.41E-46	yes	versionedSLOC_with_revisions	checkstyle	modified at least once
0.02	0.2751215176	no	versionedSLOC_with_revisions	commons-io	modified at least once
0.01	0.1493320928	no	versionedSLOC_with_revisions	commons-lang	modified at least once
0.05	9.40E-06	yes	versionedSLOC_with_revisions	cucumber-jvm	modified at least once
0.08	1.08E-27	yes	versionedSLOC_with_revisions	docx4j	modified at least once
0.12	5.78E-17	yes	versionedSLOC_with_revisions	fastjson	modified at least once
0.03	0.158790865	no	versionedSLOC_with_revisions	gson	modified at least once
0.04	2.16E-13	yes	versionedSLOC_with_revisions	hibernate-search	modified at least once

-0.01	0.09668866834	no	versionedSLOC_with_revisions	javaparser	modified at least once
0.12	5.12E-33	yes	versionedSLOC_with_revisions	jgit	modified at least once
-0.09	1.85E-10	yes	versionedSLOC_with_revisions	jna	modified at least once
0.03	0.02167995263	yes	versionedSLOC_with_revisions	junit4	modified at least once
0	0.9936589096	no	versionedSLOC_with_revisions	junit5	modified at least once
-0.05	5.47E-05	yes	versionedSLOC_with_revisions	mockito	modified at least once
0.02	0.01384838815	yes	versionedSLOC_with_revisions	okhttp	modified at least once
0.02	0.002643898349	yes	versionedSLOC_with_revisions	pmd	modified at least once
0.02	1.58E-06	yes	versionedSLOC_with_revisions	spring-boot	modified at least once
0.05	0.000103666922	yes	versionedSLOC_with_revisions	titan	modified at least once
-0.05	1.17E-22	yes	versionedSLOC_with_revisions	wicket	modified at least once

B.4 SLOC measurements with #Bugs (modified once)

Kendall τ correlation coefficient of all 53 projects for each SLOC measurements with #Bugs is given below. Methods that have been modified at least once in each project were used to compute the correlation values.

corr	p_value	significant	correlation of	repo	Methods considered
0.24	1.91E-56	yes	avgSLOC_with_bugs	antlr4	modified at least once
-0.06	4.45E-18	yes	avgSLOC_with_bugs	assertj-core	modified at least once
0.1	1.80E-11	yes	avgSLOC_with_bugs	astyanax	modified at least once
0.32	5.90E-78	yes	avgSLOC_with_bugs	atmosphere	modified at least once
0.2	1.89E-246	yes	avgSLOC_with_bugs	drill	modified at least once
0.23	7.72E-234	yes	avgSLOC_with_bugs	DSpace	modified at least once
0.36	3.14E-116	yes	avgSLOC_with_bugs	Essentials	modified at least once
0.23	0	yes	avgSLOC_with_bugs	flink	modified at least once
0.24	2.95E-34	yes	avgSLOC_with_bugs	flyway	modified at least once
0.18	5.37E-251	yes	avgSLOC_with_bugs	guava	modified at least once
0.2	8.28E-16	yes	avgSLOC_with_bugs	hawtio	modified at least once
0.27	5.03E-34	yes	avgSLOC_with_bugs	hector	modified at least once
0.17	0	yes	avgSLOC_with_bugs	hibernate-orm	modified at least once
0.26	8.38E-40	yes	avgSLOC_with_bugs	Hystrix	modified at least once
0.27	0	yes	avgSLOC_with_bugs	jetty.project	modified at least once
0.13	9.24E-50	yes	avgSLOC_with_bugs	logging-log4j2	modified at least once
0.13	2.85E-19	yes	avgSLOC_with_bugs	lombok	modified at least once
0.15	1.87E-64	yes	avgSLOC_with_bugs	mongo-java-driver	modified at least once
0.26	3.06E-121	yes	avgSLOC_with_bugs	mybatis-3	modified at least once
0.26	0	yes	avgSLOC_with_bugs	netty	modified at least once
0.18	7.38E-112	yes	avgSLOC_with_bugs	openmrs-core	modified at least once
-0.03	0.08470482722	no	avgSLOC_with_bugs	rabbitmq-java-client	modified at least once
-0.01	0.2436893125	no	avgSLOC_with_bugs	RxJava	modified at least once
0.17	0	yes	avgSLOC_with_bugs	sonarqube	modified at least once
0.04	1.81E-15	yes	avgSLOC_with_bugs	soot	modified at least once
0.15	0.002273687906	yes	avgSLOC_with_bugs	spark	modified at least once
0.13	2.67E-286	yes	avgSLOC_with_bugs	spring-framework	modified at least once
0.21	5.60E-30	yes	avgSLOC_with_bugs	swagger-core	modified at least once
0.13	8.49E-16	yes	avgSLOC_with_bugs	truth	modified at least once
0.31	2.98E-84	yes	avgSLOC_with_bugs	Twitter4J	modified at least once
0.16	7.37E-40	yes	avgSLOC_with_bugs	voldemort	modified at least once
0.26	2.37E-39	yes	avgSLOC_with_bugs	vraptor4	modified at least once
0.22	0	yes	avgSLOC_with_bugs	wildfly	modified at least once
0.15	9.73E-82	yes	avgSLOC_with_bugs	ant	modified at least once
0.18	4.47E-111	yes	avgSLOC_with_bugs	checkstyle	modified at least once
0.07	3.75E-05	yes	avgSLOC_with_bugs	commons-io	modified at least once
0.13	5.93E-41	yes	avgSLOC_with_bugs	commons-lang	modified at least once
0.18	1.14E-49	yes	avgSLOC_with_bugs	cucumber-jvm	modified at least once
0.14	1.59E-138	yes	avgSLOC_with_bugs	docx4j	modified at least once
0.17	1.62E-31	yes	avgSLOC_with_bugs	fastjson	modified at least once
0.11	2.26E-07	yes	avgSLOC_with_bugs	gson	modified at least once
0.15	1.13E-101	yes	avgSLOC_with_bugs	hibernate-search	modified at least once
0.1	2.73E-39	yes	avgSLOC_with_bugs	javaparser	modified at least once

-0.07	2.08E-15	yes	avgSLOC_with_bugs	jgit	modified at least once
0.17	1.45E-33	yes	avgSLOC_with_bugs	jna	modified at least once
0.16	4.54E-24	yes	avgSLOC_with_bugs	junit4	modified at least once
0.15	1.07E-39	yes	avgSLOC_with_bugs	junit5	modified at least once
0.09	4.02E-15	yes	avgSLOC_with_bugs	mockito	modified at least once
0.27	5.78E-96	yes	avgSLOC_with_bugs	okhttp	modified at least once
0.21	1.98E-214	yes	avgSLOC_with_bugs	pmd	modified at least once
0.13	3.54E-125	yes	avgSLOC_with_bugs	spring-boot	modified at least once
0.3	2.17E-49	yes	avgSLOC_with_bugs	titan	modified at least once
0.08	2.45E-33	yes	avgSLOC_with_bugs	wicket	modified at least once
0.19	3.98E-36	yes	introSLOC_with_bugs	antlr4	modified at least once
-0.07	8.46E-27	yes	introSLOC_with_bugs	assertj-core	modified at least once
0.07	5.75E-06	yes	introSLOC_with_bugs	astyanax	modified at least once
0.28	1.02E-56	yes	introSLOC_with_bugs	atmosphere	modified at least once
0.16	8.55E-163	yes	introSLOC_with_bugs	drill	modified at least once
0.2	1.01E-180	yes	introSLOC_with_bugs	DSpace	modified at least once
0.36	6.41E-114	yes	introSLOC_with_bugs	Essentials	modified at least once
0.21	0	yes	introSLOC_with_bugs	flink	modified at least once
0.19	1.40E-22	yes	introSLOC_with_bugs	flyway	modified at least once
0.18	2.27E-249	yes	introSLOC_with_bugs	guava	modified at least once
0.16	1.68E-10	yes	introSLOC_with_bugs	hawtio	modified at least once
0.22	2.51E-22	yes	introSLOC_with_bugs	hector	modified at least once
0.16	0	yes	introSLOC_with_bugs	hibernate-orm	modified at least once
0.23	3.05E-32	yes	introSLOC_with_bugs	Hystrix	modified at least once
0.24	0	yes	introSLOC_with_bugs	jetty.project	modified at least once
0.11	7.05E-38	yes	introSLOC_with_bugs	logging-log4j2	modified at least once
0.1	3.13E-12	yes	introSLOC_with_bugs	lombok	modified at least once
0.11	5.28E-36	yes	introSLOC_with_bugs	mongo-java-driver	modified at least once
0.25	6.35E-106	yes	introSLOC_with_bugs	mybatis-3	modified at least once
0.23	4.02E-257	yes	introSLOC_with_bugs	netty	modified at least once
0.16	6.60E-85	yes	introSLOC_with_bugs	openmrs-core	modified at least once
-0.03	0.07447537051	no	introSLOC_with_bugs	rabbitmq-java-client	modified at least once
-0.01	0.2021607384	no	introSLOC_with_bugs	RxJava	modified at least once
0.15	1.18E-253	yes	introSLOC_with_bugs	sonarqube	modified at least once
0	0.586524016	no	introSLOC_with_bugs	soot	modified at least once
0.13	0.006470420676	yes	introSLOC_with_bugs	spark	modified at least once
0.12	1.05E-238	yes	introSLOC_with_bugs	spring-framework	modified at least once
0.2	1.09E-26	yes	introSLOC_with_bugs	swagger-core	modified at least once
0.09	4.70E-08	yes	introSLOC_with_bugs	truth	modified at least once
0.32	5.45E-84	yes	introSLOC_with_bugs	Twitter4J	modified at least once
0.13	4.67E-28	yes	introSLOC_with_bugs	voldemort	modified at least once
0.25	1.00E-35	yes	introSLOC_with_bugs	vraptor4	modified at least once
0.18	2.39E-240	yes	introSLOC_with_bugs	wildfly	modified at least once
0.13	3.04E-66	yes	introSLOC_with_bugs	ant	modified at least once

0.17	7.50E-99	yes	introSLOC_with_bugs	checkstyle	modified at least once
0.06	0.000168559869	yes	introSLOC_with_bugs	commons-io	modified at least once
0.11	2.77E-28	yes	introSLOC_with_bugs	commons-lang	modified at least once
0.15	5.55E-32	yes	introSLOC_with_bugs	cucumber-jvm	modified at least once
0.13	7.50E-128	yes	introSLOC_with_bugs	docx4j	modified at least once
0.15	1.82E-22	yes	introSLOC_with_bugs	fastjson	modified at least once
0.08	0.000131591551	yes	introSLOC_with_bugs	gson	modified at least once
0.13	4.58E-77	yes	introSLOC_with_bugs	hibernate-search	modified at least once
0.02	0.005388290806	yes	introSLOC_with_bugs	javaparser	modified at least once
-0.08	8.42E-18	yes	introSLOC_with_bugs	jgit	modified at least once
0.15	1.62E-24	yes	introSLOC_with_bugs	jna	modified at least once
0.16	1.30E-22	yes	introSLOC_with_bugs	junit4	modified at least once
0.14	4.21E-30	yes	introSLOC_with_bugs	junit5	modified at least once
0.08	4.97E-13	yes	introSLOC_with_bugs	mockito	modified at least once
0.25	6.35E-82	yes	introSLOC_with_bugs	okhttp	modified at least once
0.18	3.88E-165	yes	introSLOC_with_bugs	pmd	modified at least once
0.11	3.58E-81	yes	introSLOC_with_bugs	spring-boot	modified at least once
0.26	7.14E-39	yes	introSLOC_with_bugs	titan	modified at least once
0.04	5.36E-11	yes	introSLOC_with_bugs	wicket	modified at least once
0.24	1.53E-56	yes	lastSLOC_with_bugs	antlr4	modified at least once
-0.06	3.89E-20	yes	lastSLOC_with_bugs	assertj-core	modified at least once
0.07	1.57E-06	yes	lastSLOC_with_bugs	astyanax	modified at least once
0.3	2.68E-67	yes	lastSLOC_with_bugs	atmosphere	modified at least once
0.2	8.50E-231	yes	lastSLOC_with_bugs	drill	modified at least once
0.23	2.90E-232	yes	lastSLOC_with_bugs	DSpace	modified at least once
0.35	8.54E-112	yes	lastSLOC_with_bugs	Essentials	modified at least once
0.22	0	yes	lastSLOC_with_bugs	flink	modified at least once
0.21	2.28E-26	yes	lastSLOC_with_bugs	flyway	modified at least once
0.18	5.22E-242	yes	lastSLOC_with_bugs	guava	modified at least once
0.19	2.62E-14	yes	lastSLOC_with_bugs	hawtio	modified at least once
0.26	9.01E-33	yes	lastSLOC_with_bugs	hector	modified at least once
0.17	0	yes	lastSLOC_with_bugs	hibernate-orm	modified at least once
0.2	4.15E-24	yes	lastSLOC_with_bugs	Hystrix	modified at least once
0.27	0	yes	lastSLOC_with_bugs	jetty.project	modified at least once
0.12	7.82E-44	yes	lastSLOC_with_bugs	logging-log4j2	modified at least once
0.13	7.30E-18	yes	lastSLOC_with_bugs	lombok	modified at least once
0.14	6.88E-55	yes	lastSLOC_with_bugs	mongo-java-driver	modified at least once
0.25	2.39E-110	yes	lastSLOC_with_bugs	mybatis-3	modified at least once
0.24	4.77E-283	yes	lastSLOC_with_bugs	netty	modified at least once
0.18	1.55E-102	yes	lastSLOC_with_bugs	openmrs-core	modified at least once
-0.04	0.03018473701	yes	lastSLOC_with_bugs	rabbitmq-java-client	modified at least once
-0.03	0.001270367457	yes	lastSLOC_with_bugs	RxJava	modified at least once
0.16	1.18E-279	yes	lastSLOC_with_bugs	sonarqube	modified at least once
0.04	2.93E-15	yes	lastSLOC_with_bugs	soot	modified at least once

0.13	0.005664765732	yes	lastSLOC_with_bugs	spark	modified at least once
0.12	2.99E-253	yes	lastSLOC_with_bugs	spring-framework	modified at least once
0.19	4.29E-24	yes	lastSLOC_with_bugs	swagger-core	modified at least once
0.13	1.20E-14	yes	lastSLOC_with_bugs	truth	modified at least once
0.28	9.13E-64	yes	lastSLOC_with_bugs	Twitter4J	modified at least once
0.15	3.68E-38	yes	lastSLOC_with_bugs	voldemort	modified at least once
0.23	1.75E-31	yes	lastSLOC_with_bugs	vraptor4	modified at least once
0.21	0	yes	lastSLOC_with_bugs	wildfly	modified at least once
0.15	5.72E-80	yes	lastSLOC_with_bugs	ant	modified at least once
0.18	1.55E-105	yes	lastSLOC_with_bugs	checkstyle	modified at least once
0.05	0.000933003427	yes	lastSLOC_with_bugs	commons-io	modified at least once
0.12	3.06E-32	yes	lastSLOC_with_bugs	commons-lang	modified at least once
0.18	5.23E-45	yes	lastSLOC_with_bugs	cucumber-jvm	modified at least once
0.13	2.98E-136	yes	lastSLOC_with_bugs	docx4j	modified at least once
0.16	8.06E-28	yes	lastSLOC_with_bugs	fastjson	modified at least once
0.11	1.16E-06	yes	lastSLOC_with_bugs	gson	modified at least once
0.14	7.11E-92	yes	lastSLOC_with_bugs	hibernate-search	modified at least once
0.08	1.78E-23	yes	lastSLOC_with_bugs	javaparser	modified at least once
-0.07	9.83E-18	yes	lastSLOC_with_bugs	jgit	modified at least once
0.17	6.27E-31	yes	lastSLOC_with_bugs	jna	modified at least once
0.15	7.52E-20	yes	lastSLOC_with_bugs	junit4	modified at least once
0.14	3.58E-30	yes	lastSLOC_with_bugs	junit5	modified at least once
0.07	4.85E-10	yes	lastSLOC_with_bugs	mockito	modified at least once
0.26	6.26E-87	yes	lastSLOC_with_bugs	okhttp	modified at least once
0.18	5.77E-157	yes	lastSLOC_with_bugs	pmd	modified at least once
0.13	7.18E-113	yes	lastSLOC_with_bugs	spring-boot	modified at least once
0.27	6.47E-40	yes	lastSLOC_with_bugs	titan	modified at least once
0.07	1.73E-29	yes	lastSLOC_with_bugs	wicket	modified at least once
0.24	6.09E-57	yes	medianSLOC_with_bugs	antlr4	modified at least once
-0.06	3.14E-20	yes	medianSLOC_with_bugs	assertj-core	modified at least once
0.08	5.63E-08	yes	medianSLOC_with_bugs	astyanax	modified at least once
0.32	6.58E-75	yes	medianSLOC_with_bugs	atmosphere	modified at least once
0.2	3.68E-238	yes	medianSLOC_with_bugs	drill	modified at least once
0.23	4.50E-232	yes	medianSLOC_with_bugs	DSpace	modified at least once
0.36	1.82E-119	yes	medianSLOC_with_bugs	Essentials	modified at least once
0.23	0	yes	medianSLOC_with_bugs	flink	modified at least once
0.23	6.61E-33	yes	medianSLOC_with_bugs	flyway	modified at least once
0.18	2.23E-249	yes	medianSLOC_with_bugs	guava	modified at least once
0.2	3.75E-15	yes	medianSLOC_with_bugs	hawtio	modified at least once
0.26	1.24E-31	yes	medianSLOC_with_bugs	hector	modified at least once
0.17	0	yes	medianSLOC_with_bugs	hibernate-orm	modified at least once
0.26	3.33E-39	yes	medianSLOC_with_bugs	Hystrix	modified at least once
0.27	0	yes	medianSLOC_with_bugs	jetty.project	modified at least once
0.12	1.18E-45	yes	medianSLOC_with_bugs	logging-log4j2	modified at least once

0.13	2.60E-18	yes	medianSLOC_with_bugs	lombok	modified at least once
0.14	1.14E-58	yes	medianSLOC_with_bugs	mongo-java-driver	modified at least once
0.26	3.45E-118	yes	medianSLOC_with_bugs	mybatis-3	modified at least once
0.25	3.91E-306	yes	medianSLOC_with_bugs	netty	modified at least once
0.18	6.04E-106	yes	medianSLOC_with_bugs	openmrs-core	modified at least once
-0.03	0.09917664645	no	medianSLOC_with_bugs	rabbitmq-java-client	modified at least once
-0.02	0.01425017817	yes	medianSLOC_with_bugs	RxJava	modified at least once
0.16	3.08E-305	yes	medianSLOC_with_bugs	sonarqube	modified at least once
0.04	6.73E-15	yes	medianSLOC_with_bugs	soot	modified at least once
0.14	0.002684856084	yes	medianSLOC_with_bugs	spark	modified at least once
0.13	2.15E-271	yes	medianSLOC_with_bugs	spring-framework	modified at least once
0.21	1.19E-29	yes	medianSLOC_with_bugs	swagger-core	modified at least once
0.13	2.70E-15	yes	medianSLOC_with_bugs	truth	modified at least once
0.31	2.68E-80	yes	medianSLOC_with_bugs	Twitter4J	modified at least once
0.15	4.32E-39	yes	medianSLOC_with_bugs	voldemort	modified at least once
0.25	2.11E-36	yes	medianSLOC_with_bugs	vraptor4	modified at least once
0.22	0	yes	medianSLOC_with_bugs	wildfly	modified at least once
0.15	1.91E-80	yes	medianSLOC_with_bugs	ant	modified at least once
0.18	5.60E-109	yes	medianSLOC_with_bugs	checkstyle	modified at least once
0.06	9.51E-05	yes	medianSLOC_with_bugs	commons-io	modified at least once
0.13	8.82E-38	yes	medianSLOC_with_bugs	commons-lang	modified at least once
0.18	2.06E-45	yes	medianSLOC_with_bugs	cucumber-jvm	modified at least once
0.14	3.35E-138	yes	medianSLOC_with_bugs	docx4j	modified at least once
0.17	5.82E-30	yes	medianSLOC_with_bugs	fastjson	modified at least once
0.11	5.19E-07	yes	medianSLOC_with_bugs	gson	modified at least once
0.14	3.74E-96	yes	medianSLOC_with_bugs	hibernate-search	modified at least once
0.06	9.31E-18	yes	medianSLOC_with_bugs	javaparser	modified at least once
-0.07	1.29E-15	yes	medianSLOC_with_bugs	jgit	modified at least once
0.17	2.01E-32	yes	medianSLOC_with_bugs	jna	modified at least once
0.16	7.92E-22	yes	medianSLOC_with_bugs	junit4	modified at least once
0.14	6.10E-33	yes	medianSLOC_with_bugs	junit5	modified at least once
0.08	9.22E-11	yes	medianSLOC_with_bugs	mockito	modified at least once
0.27	1.12E-93	yes	medianSLOC_with_bugs	okhttp	modified at least once
0.2	6.56E-193	yes	medianSLOC_with_bugs	pmd	modified at least once
0.13	2.04E-113	yes	medianSLOC_with_bugs	spring-boot	modified at least once
0.28	3.24E-44	yes	medianSLOC_with_bugs	titan	modified at least once
0.07	1.42E-28	yes	medianSLOC_with_bugs	wicket	modified at least once
0.39	8.09E-118	yes	stdSLOC_with_bugs	antlr4	modified at least once
0.2	6.99E-156	yes	stdSLOC_with_bugs	assertj-core	modified at least once
0.37	9.06E-128	yes	stdSLOC_with_bugs	astyanax	modified at least once
0.47	2.93E-134	yes	stdSLOC_with_bugs	atmosphere	modified at least once
0.38	0	yes	stdSLOC_with_bugs	drill	modified at least once
0.31	0	yes	stdSLOC_with_bugs	DSpace	modified at least once
0.44	1.61E-147	yes	stdSLOC_with_bugs	Essentials	modified at least once

0.41	0	yes	stdSLOC_with_bugs	flink	modified at least once
0.49	7.96E-120	yes	stdSLOC_with_bugs	flyway	modified at least once
0.09	2.09E-48	yes	stdSLOC_with_bugs	guava	modified at least once
0.43	2.78E-54	yes	stdSLOC_with_bugs	hawtio	modified at least once
0.4	2.51E-63	yes	stdSLOC_with_bugs	hector	modified at least once
0.2	0	yes	stdSLOC_with_bugs	hibernate-orm	modified at least once
0.39	4.64E-73	yes	stdSLOC_with_bugs	Hystrix	modified at least once
0.27	0	yes	stdSLOC_with_bugs	jetty.project	modified at least once
0.29	3.15E-197	yes	stdSLOC_with_bugs	logging-log4j2	modified at least once
0.36	2.34E-108	yes	stdSLOC_with_bugs	lombok	modified at least once
0.3	3.10E-223	yes	stdSLOC_with_bugs	mongo-java-driver	modified at least once
0.39	1.61E-209	yes	stdSLOC_with_bugs	mybatis-3	modified at least once
0.39	0	yes	stdSLOC_with_bugs	netty	modified at least once
0.28	1.79E-221	yes	stdSLOC_with_bugs	openmrs-core	modified at least once
0.14	2.10E-13	yes	stdSLOC_with_bugs	rabbitmq-java-client	modified at least once
0.3	5.08E-180	yes	stdSLOC_with_bugs	RxJava	modified at least once
0.24	0	yes	stdSLOC_with_bugs	sonarqube	modified at least once
0.22	0	yes	stdSLOC_with_bugs	soot	modified at least once
0.28	6.70E-08	yes	stdSLOC_with_bugs	spark	modified at least once
0.26	0	yes	stdSLOC_with_bugs	spring-framework	modified at least once
0.34	3.78E-64	yes	stdSLOC_with_bugs	swagger-core	modified at least once
0.24	2.68E-44	yes	stdSLOC_with_bugs	truth	modified at least once
0.39	6.00E-106	yes	stdSLOC_with_bugs	Twitter4J	modified at least once
0.33	9.75E-144	yes	stdSLOC_with_bugs	voldemort	modified at least once
0.35	3.16E-59	yes	stdSLOC_with_bugs	vraptor4	modified at least once
0.36	0	yes	stdSLOC_with_bugs	wildfly	modified at least once
0.2	2.59E-114	yes	stdSLOC_with_bugs	ant	modified at least once
0.29	2.30E-213	yes	stdSLOC_with_bugs	checkstyle	modified at least once
0.16	3.27E-18	yes	stdSLOC_with_bugs	commons-io	modified at least once
0.32	2.56E-169	yes	stdSLOC_with_bugs	commons-lang	modified at least once
0.34	2.23E-135	yes	stdSLOC_with_bugs	cucumber-jvm	modified at least once
0.32	0	yes	stdSLOC_with_bugs	docx4j	modified at least once
0.33	1.59E-83	yes	stdSLOC_with_bugs	fastjson	modified at least once
0.17	9.33E-12	yes	stdSLOC_with_bugs	gson	modified at least once
0.24	1.80E-219	yes	stdSLOC_with_bugs	hibernate-search	modified at least once
0.33	0	yes	stdSLOC_with_bugs	javaparser	modified at least once
0.07	9.77E-14	yes	stdSLOC_with_bugs	jgit	modified at least once
0.31	1.68E-82	yes	stdSLOC_with_bugs	jna	modified at least once
0.33	2.54E-73	yes	stdSLOC_with_bugs	junit4	modified at least once
0.25	3.38E-85	yes	stdSLOC_with_bugs	junit5	modified at least once
0.37	2.24E-180	yes	stdSLOC_with_bugs	mockito	modified at least once
0.41	1.20E-177	yes	stdSLOC_with_bugs	okhttp	modified at least once
0.43	0	yes	stdSLOC_with_bugs	pmd	modified at least once
0.28	0	yes	stdSLOC_with_bugs	spring-boot	modified at least once

0.32	7.42E-52	yes	stdSLOC_with_bugs	titan	modified at least once
0.31	0	yes	stdSLOC_with_bugs	wicket	modified at least once
0.08	3.54E-11	yes	versionedSLOC_with_bugs	antlr4	modified at least once
0.05	1.35E-08	yes	versionedSLOC_with_bugs	assertj-core	modified at least once
-0.06	6.96E-07	yes	versionedSLOC_with_bugs	astyanax	modified at least once
0.04	0.003361900251	yes	versionedSLOC_with_bugs	atmosphere	modified at least once
0.06	4.24E-28	yes	versionedSLOC_with_bugs	drill	modified at least once
0.02	0.001941984196	yes	versionedSLOC_with_bugs	DSpace	modified at least once
0.16	2.69E-38	yes	versionedSLOC_with_bugs	Essentials	modified at least once
0.08	4.99E-67	yes	versionedSLOC_with_bugs	flink	modified at least once
-0.02	0.3230433711	no	versionedSLOC_with_bugs	flyway	modified at least once
0.11	2.30E-80	yes	versionedSLOC_with_bugs	guava	modified at least once
0.01	0.8136356375	no	versionedSLOC_with_bugs	hawtio	modified at least once
0.14	1.82E-14	yes	versionedSLOC_with_bugs	hector	modified at least once
0.12	5.52E-182	yes	versionedSLOC_with_bugs	hibernate-orm	modified at least once
0.11	4.98E-13	yes	versionedSLOC_with_bugs	Hystrix	modified at least once
0.09	3.26E-60	yes	versionedSLOC_with_bugs	jetty.project	modified at least once
0.06	8.44E-17	yes	versionedSLOC_with_bugs	logging-log4j2	modified at least once
0.04	0.000864489803	yes	versionedSLOC_with_bugs	lombok	modified at least once
0.02	0.001163250156	yes	versionedSLOC_with_bugs	mongo-java-driver	modified at least once
0.18	4.20E-37	yes	versionedSLOC_with_bugs	mybatis-3	modified at least once
0.1	1.19E-56	yes	versionedSLOC_with_bugs	netty	modified at least once
0.08	4.61E-29	yes	versionedSLOC_with_bugs	openmrs-core	modified at least once
-0.07	4.31E-05	yes	versionedSLOC_with_bugs	rabbitmq-java-client	modified at least once
-0.06	3.33E-11	yes	versionedSLOC_with_bugs	RxJava	modified at least once
0.06	2.21E-54	yes	versionedSLOC_with_bugs	sonarqube	modified at least once
0.2	5.13E-168	yes	versionedSLOC_with_bugs	soot	modified at least once
-0.01	0.8188510354	no	versionedSLOC_with_bugs	spark	modified at least once
0.07	7.66E-71	yes	versionedSLOC_with_bugs	spring-framework	modified at least once
0.1	1.40E-13	yes	versionedSLOC_with_bugs	swagger-core	modified at least once
0.02	0.1943580845	no	versionedSLOC_with_bugs	truth	modified at least once
0.12	1.22E-18	yes	versionedSLOC_with_bugs	Twitter4J	modified at least once
-0.03	0.001110191766	yes	versionedSLOC_with_bugs	voldemort	modified at least once
0.12	5.83E-14	yes	versionedSLOC_with_bugs	vraptor4	modified at least once
0.08	4.01E-74	yes	versionedSLOC_with_bugs	wildfly	modified at least once
0.11	3.78E-47	yes	versionedSLOC_with_bugs	ant	modified at least once
0.13	4.58E-56	yes	versionedSLOC_with_bugs	checkstyle	modified at least once
0	0.9144872895	no	versionedSLOC_with_bugs	commons-io	modified at least once
0.1	6.54E-19	yes	versionedSLOC_with_bugs	commons-lang	modified at least once
0.04	0.001945972113	yes	versionedSLOC_with_bugs	cucumber-jvm	modified at least once
0.15	9.88E-93	yes	versionedSLOC_with_bugs	docx4j	modified at least once
0.06	2.20E-05	yes	versionedSLOC_with_bugs	fastjson	modified at least once
0.04	0.06728464084	no	versionedSLOC_with_bugs	gson	modified at least once
0.09	5.03E-54	yes	versionedSLOC_with_bugs	hibernate-search	modified at least once

0.08	1.70E-35	yes	versionedSLOC_with_bugs	javaparser	modified at least once
0.03	0.00330332825	yes	versionedSLOC_with_bugs	jgit	modified at least once
0.1	1.03E-11	yes	versionedSLOC_with_bugs	jna	modified at least once
0.1	8.03E-11	yes	versionedSLOC_with_bugs	junit4	modified at least once
0.07	5.41E-14	yes	versionedSLOC_with_bugs	junit5	modified at least once
0.04	0.002321798954	yes	versionedSLOC_with_bugs	mockito	modified at least once
0.11	4.75E-25	yes	versionedSLOC_with_bugs	okhttp	modified at least once
0.11	2.52E-70	yes	versionedSLOC_with_bugs	pmd	modified at least once
0.04	2.21E-13	yes	versionedSLOC_with_bugs	spring-boot	modified at least once
0.04	0.003935807171	yes	versionedSLOC_with_bugs	titan	modified at least once
0	0.8453559085	no	versionedSLOC_with_bugs	wicket	modified at least once

B.5 Cliff's Delta and Mann-Whitney U Test Between Correlation Values of Different SLOC Measures with #Revisions (All Methods)

The table below shows the effect size and significance test result between correlation values of different SLOC measures with #Revisions for 53 projects. All the methods in each project were considered to compute the correlation values.

d	size	w	p_value	significant	grp1 correlation values	grp2 correlation values
-0.196	small	1129	0.082	no	introSLOC_with_revisions	avgSLOC_with_revisions
-0.15	small	1193.5	0.183	no	introSLOC_with_revisions	medianSLOC_with_revisions
-0.204	small	1117.5	0.07	no	lastSLOC_with_revisions	versioned_sloc_with_revisions
-0.151	small	1193	0.182	no	medianSLOC_with_revisions	versioned_sloc_with_revisions
-0.262	small	1037	0.02	yes	introSLOC_with_revisions	versioned_sloc_with_revisions
-0.069	negligible	1308	0.543	no	introSLOC_with_revisions	lastSLOC_with_revisions
-0.124	negligible	1230.5	0.272	no	lastSLOC_with_revisions	avgSLOC_with_revisions
-0.084	negligible	1286	0.455	no	lastSLOC_with_revisions	medianSLOC_with_revisions
0.045	negligible	1467.5	0.692	no	avgSLOC_with_revisions	medianSLOC_with_revisions
-0.124	negligible	1230.5	0.273	no	avgSLOC_with_revisions	versioned_sloc_with_revisions
-0.973	large	38.5	0	yes	introSLOC_with_revisions	stdSLOC_with_revisions
-0.967	large	46	0	yes	lastSLOC_with_revisions	stdSLOC_with_revisions
-0.961	large	55	0	yes	avgSLOC_with_revisions	stdSLOC_with_revisions
-0.964	large	51	0	yes	medianSLOC_with_revisions	stdSLOC_with_revisions
0.93	large	2711	0	yes	stdSLOC_with_revisions	versioned_sloc_with_revisions

B.6 Cliff's Delta and Mann-Whitney U Test Between Correlation Values of Different SLOC Measures with #Bugs (All Methods)

The table below shows the effect size and significance test result between correlation values of different SLOC measures with #Bugs for 53 projects. Here all the methods in each project were considered to compute the correlation values.

d	size	w	p_value	significant	grp1 correlation values	grp2 correlation values
0.233	small	1699	0.04	yes	avgSLOC_with_bugs	versioned_sloc_with_bugs
0.148	small	1581.5	0.193	no	lastSLOC_with_bugs	versioned_sloc_with_bugs
0.194	small	1645.5	0.087	no	medianSLOC_with_bugs	versioned_sloc_with_bugs
0.035	negligible	1453.5	0.759	no	avgSLOC_with_bugs	medianSLOC_with_bugs
-0.066	negligible	1312	0.561	no	introSLOC_with_bugs	lastSLOC_with_bugs
-0.141	negligible	1207	0.213	no	introSLOC_with_bugs	avgSLOC_with_bugs
-0.11	negligible	1250	0.33	no	introSLOC_with_bugs	medianSLOC_with_bugs
0.079	negligible	1487.5	0.484	no	introSLOC_with_bugs	versioned_sloc_with_bugs
-0.076	negligible	1298	0.502	no	lastSLOC_with_bugs	avgSLOC_with_bugs
-0.044	negligible	1342.5	0.697	no	lastSLOC_with_bugs	medianSLOC_with_bugs
-0.917	large	116.5	0	yes	avgSLOC_with_bugs	stdSLOC_with_bugs
-0.942	large	81.5	0	yes	introSLOC_with_bugs	stdSLOC_with_bugs
-0.93	large	99	0	yes	lastSLOC_with_bugs	stdSLOC_with_bugs
-0.921	large	110.5	0	yes	medianSLOC_with_bugs	stdSLOC_with_bugs
0.973	large	2718.5	0	yes	stdSLOC_with_bugs	versioned_sloc_with_bugs

B.7 Cliff's Delta and Mann-Whitney U Test Between Correlation Values of Different SLOC Measures with #Revisions (Modified Once)

The table below shows the effect size and significance test result between correlation values of different SLOC measures with #Revisions for 53 projects. Here methods that have been modified at least once in each project were considered to compute the correlation values.

d	size	w	p_value	significant	grp1 correlation values	grp2 correlation values
-0.21	small	1109	0.062	no	introSLOC_with_revisions	avgSLOC_with_revisions
-0.163	small	1176	0.149	no	introSLOC_with_revisions	medianSLOC_with_revisions
-0.132	negligible	1219.5	0.243	no	lastSLOC_with_revisions	avgSLOC_with_revisions
-0.08	negligible	1291.5	0.477	no	lastSLOC_with_revisions	medianSLOC_with_revisions
-0.098	negligible	1266.5	0.384	no	introSLOC_with_revisions	lastSLOC_with_revisions
0.052	negligible	1477	0.649	no	avgSLOC_with_revisions	medianSLOC_with_revisions
1	large	2279	0	yes	stdSLOC_with_revisions	versioned_sloc_with_revisions
-0.963	large	51.5	0	yes	medianSLOC_with_revisions	stdSLOC_with_revisions
0.86	large	2119	0	yes	medianSLOC_with_revisions	versioned_sloc_with_revisions
-0.974	large	36	0	yes	lastSLOC_with_revisions	stdSLOC_with_revisions
0.836	large	2092.5	0	yes	lastSLOC_with_revisions	versioned_sloc_with_revisions
-0.98	large	28	0	yes	introSLOC_with_revisions	stdSLOC_with_revisions
0.824	large	2078	0	yes	introSLOC_with_revisions	versioned_sloc_with_revisions
-0.963	large	52	0	yes	avgSLOC_with_revisions	stdSLOC_with_revisions
0.874	large	2135.5	0	yes	avgSLOC_with_revisions	versioned_sloc_with_revisions

B.8 Cliff's Delta and Mann-Whitney U Test Between Correlation Values of Different SLOC Measures with #Bugs (Modified Once)

The table below shows the effect size and significance test result between correlation values of different SLOC measures with #Bugs for 53 projects. Methods that have been modified at least once in each project were considered to compute the correlation values.

d	size	w	p_value	significant	grp1 correlation values	grp2 correlation values
-0.166	small	1171	0.14	no	introSLOC_with_bugs	avgSLOC_with_bugs
-0.098	negligible	1267.5	0.388	no	introSLOC_with_bugs	lastSLOC_with_bugs
-0.137	negligible	1211.5	0.223	no	introSLOC_with_bugs	medianSLOC_with_bugs
-0.073	negligible	1302.5	0.521	no	lastSLOC_with_bugs	avgSLOC_with_bugs
-0.045	negligible	1341.5	0.693	no	lastSLOC_with_bugs	medianSLOC_with_bugs
0.031	negligible	1447.5	0.788	no	avgSLOC_with_bugs	medianSLOC_with_bugs
-0.804	large	275.5	0	yes	introSLOC_with_bugs	stdSLOC_with_bugs
-0.783	large	305	0	yes	lastSLOC_with_bugs	stdSLOC_with_bugs
-0.74	large	364.5	0	yes	avgSLOC_with_bugs	stdSLOC_with_bugs
-0.754	large	346	0	yes	medianSLOC_with_bugs	stdSLOC_with_bugs
0.543	large	1881	0	yes	introSLOC_with_bugs	versioned_sloc_with_bugs
0.628	large	1985	0	yes	lastSLOC_with_bugs	versioned_sloc_with_bugs
0.64	large	1999.5	0	yes	medianSLOC_with_bugs	versioned_sloc_with_bugs
0.672	large	2038	0	yes	avgSLOC_with_bugs	versioned_sloc_with_bugs
0.953	large	2380.5	0	yes	stdSLOC_with_bugs	versioned_sloc_with_bugs

Appendix C

RQ2-SLOC Control

This appendix includes per project correlation values of SLOC with revision and bug density described in Chapter 5.2.

C.1 Correlation Values of SLOC with Revision Density

The table below shows the correlation value of SLOC with revision density (i.e., revisions per SLOC) for the 53 projects discussed in Chapter 5.2.

corr	p_value	significant	correlation between	repo
-0.06	1.30E-31	yes	versionedSLOC_with_revision_density	antlr4
0.08	2.35E-70	yes	versionedSLOC_with_revision_density	assertj-core
-0.28	4.63E-226	yes	versionedSLOC_with_revision_density	astyanax
0.1	4.57E-32	yes	versionedSLOC_with_revision_density	atmosphere
0	0.7117940594	no	versionedSLOC_with_revision_density	drill
-0.19	8.35E-306	yes	versionedSLOC_with_revision_density	DSpace
-0.2	1.59E-105	yes	versionedSLOC_with_revision_density	Essentials
-0.07	8.69E-125	yes	versionedSLOC_with_revision_density	flink
0.12	1.19E-27	yes	versionedSLOC_with_revision_density	flyway
0.08	1.76E-115	yes	versionedSLOC_with_revision_density	guava
0.1	6.07E-15	yes	versionedSLOC_with_revision_density	hawtio
-0.01	0.2658017702	no	versionedSLOC_with_revision_density	hector
-0.17	0	yes	versionedSLOC_with_revision_density	hibernate-orm
0.08	3.96E-17	yes	versionedSLOC_with_revision_density	Hystrix
0.03	1.20E-11	yes	versionedSLOC_with_revision_density	jetty.project
-0.26	0	yes	versionedSLOC_with_revision_density	logging-log4j2
-0.09	7.54E-20	yes	versionedSLOC_with_revision_density	lombok
-0.12	1.29E-113	yes	versionedSLOC_with_revision_density	mongo-java-driver
-0.01	0.1727094407	no	versionedSLOC_with_revision_density	mybatis-3
0.02	6.20E-06	yes	versionedSLOC_with_revision_density	netty
-0.12	1.23E-127	yes	versionedSLOC_with_revision_density	openmrs-core
-0.13	4.37E-27	yes	versionedSLOC_with_revision_density	rabbitmq-java-client
0.16	5.16E-216	yes	versionedSLOC_with_revision_density	RxJava
0.04	2.96E-50	yes	versionedSLOC_with_revision_density	sonarqube
0.14	3.01E-299	yes	versionedSLOC_with_revision_density	soot
0.06	0.006347201474	yes	versionedSLOC_with_revision_density	spark
-0.25	0	yes	versionedSLOC_with_revision_density	spring-framework
0.03	9.26E-05	yes	versionedSLOC_with_revision_density	swagger-core
-0.17	4.00E-64	yes	versionedSLOC_with_revision_density	truth
-0.25	8.01E-119	yes	versionedSLOC_with_revision_density	Twitter4J
0.02	0.0009357631649	yes	versionedSLOC_with_revision_density	voldemort
-0.04	0.0002822230013	yes	versionedSLOC_with_revision_density	vraptor4
0.14	0	yes	versionedSLOC_with_revision_density	wildfly
-0.01	0.2506995356	no	versionedSLOC_with_revision_density	ant
-0.18	3.05E-181	yes	versionedSLOC_with_revision_density	checkstyle
-0.3	1.49E-120	yes	versionedSLOC_with_revision_density	commons-io
-0.07	6.62E-22	yes	versionedSLOC_with_revision_density	commons-lang
-0.12	9.53E-36	yes	versionedSLOC_with_revision_density	cucumber-jvm
0.13	9.32E-187	yes	versionedSLOC_with_revision_density	docx4j
0.32	0	yes	versionedSLOC_with_revision_density	fastjson
-0.04	0.001996072112	yes	versionedSLOC_with_revision_density	gson
-0.37	0	yes	versionedSLOC_with_revision_density	hibernate-search

-0.36	0	yes	versionedSLOC_with_revision_density	javaparser
-0.06	1.93E-20	yes	versionedSLOC_with_revision_density	jgit
-0.28	2.99E-155	yes	versionedSLOC_with_revision_density	jna
-0.2	9.93E-70	yes	versionedSLOC_with_revision_density	junit4
-0.15	1.58E-94	yes	versionedSLOC_with_revision_density	junit5
-0.11	5.91E-37	yes	versionedSLOC_with_revision_density	mockito
-0.24	1.39E-189	yes	versionedSLOC_with_revision_density	okhttp
-0.15	1.16E-250	yes	versionedSLOC_with_revision_density	pmd
-0.21	0	yes	versionedSLOC_with_revision_density	spring-boot
0.18	3.09E-94	yes	versionedSLOC_with_revision_density	titan
-0.12	2.56E-208	yes	versionedSLOC_with_revision_density	wicket

C.2 Correlation Values of SLOC with Bug Density

The table below shows the correlation value of versioned SLOC with bug density (i.e., bugs per SLOC) for the 53 projects discussed in Chapter 5.2.

corr	p_value	significant	correlation between	repo
0.02	0.0001848814995	yes	versionedSLOC_with_bug_density	antlr4
0.11	1.52E-113	yes	versionedSLOC_with_bug_density	assertj-core
-0.09	2.10E-21	yes	versionedSLOC_with_bug_density	astyanax
0.15	1.38E-59	yes	versionedSLOC_with_bug_density	atmosphere
0.12	2.38E-186	yes	versionedSLOC_with_bug_density	drill
0.04	1.10E-12	yes	versionedSLOC_with_bug_density	DSpace
0.16	5.38E-57	yes	versionedSLOC_with_bug_density	Essentials
0.11	2.18E-215	yes	versionedSLOC_with_bug_density	flink
0.1	3.84E-19	yes	versionedSLOC_with_bug_density	flyway
0.14	0	yes	versionedSLOC_with_bug_density	guava
0.09	3.74E-11	yes	versionedSLOC_with_bug_density	hawtio
0.16	4.37E-31	yes	versionedSLOC_with_bug_density	hector
0.11	1.17E-250	yes	versionedSLOC_with_bug_density	hibernate-orm
0.16	8.07E-53	yes	versionedSLOC_with_bug_density	Hystrix
0.16	4.46E-298	yes	versionedSLOC_with_bug_density	jetty.project
0.06	2.86E-22	yes	versionedSLOC_with_bug_density	logging-log4j2
0.07	1.84E-11	yes	versionedSLOC_with_bug_density	lombok
0.05	7.18E-19	yes	versionedSLOC_with_bug_density	mongo-java-driver
0.2	3.11E-77	yes	versionedSLOC_with_bug_density	mybatis-3
0.14	2.38E-227	yes	versionedSLOC_with_bug_density	netty
0.09	6.09E-52	yes	versionedSLOC_with_bug_density	openmrs-core
-0.01	0.365683509	no	versionedSLOC_with_bug_density	rabbitmq-java-client
0.09	6.34E-71	yes	versionedSLOC_with_bug_density	RxJava
0.1	2.71E-305	yes	versionedSLOC_with_bug_density	sonarqube
0.2	0	yes	versionedSLOC_with_bug_density	soot
0.06	0.005387447232	yes	versionedSLOC_with_bug_density	spark
0.08	1.04E-145	yes	versionedSLOC_with_bug_density	spring-framework
0.13	9.04E-43	yes	versionedSLOC_with_bug_density	swagger-core
0.05	6.11E-06	yes	versionedSLOC_with_bug_density	truth
0.13	8.91E-28	yes	versionedSLOC_with_bug_density	Twitter4J
0.1	3.67E-43	yes	versionedSLOC_with_bug_density	voldemort
0.12	2.83E-23	yes	versionedSLOC_with_bug_density	vraptor4
0.15	0	yes	versionedSLOC_with_bug_density	wildfly
0.15	5.79E-149	yes	versionedSLOC_with_bug_density	ant
0.09	3.82E-35	yes	versionedSLOC_with_bug_density	checkstyle
-0.06	1.49E-05	yes	versionedSLOC_with_bug_density	commons-io
0.08	1.19E-25	yes	versionedSLOC_with_bug_density	commons-lang
0.06	5.41E-10	yes	versionedSLOC_with_bug_density	cucumber-jvm
0.14	1.21E-208	yes	versionedSLOC_with_bug_density	docx4j
0.19	4.70E-187	yes	versionedSLOC_with_bug_density	fastjson
0.07	9.76E-07	yes	versionedSLOC_with_bug_density	gson
0.08	9.25E-52	yes	versionedSLOC_with_bug_density	hibernate-search

0.08	6.58E-40	yes	versionedSLOC_with_bug_density	javaparser
0.07	1.82E-22	yes	versionedSLOC_with_bug_density	jgit
0.05	2.48E-05	yes	versionedSLOC_with_bug_density	jna
0.07	1.66E-08	yes	versionedSLOC_with_bug_density	junit4
0.08	1.02E-18	yes	versionedSLOC_with_bug_density	junit5
0.03	0.0008926772741	yes	versionedSLOC_with_bug_density	mockito
0.09	8.49E-23	yes	versionedSLOC_with_bug_density	okhttp
0.11	7.56E-111	yes	versionedSLOC_with_bug_density	pmd
0.02	0.0002051462903	yes	versionedSLOC_with_bug_density	spring-boot
0.17	4.41E-71	yes	versionedSLOC_with_bug_density	titan
0	0.7364092526	no	versionedSLOC_with_bug_density	wicket

C.3 Getter and Setter Analysis

The table below shows the Cliff's δ and Mann-Whitney U test result between correlation values of SLOC with revision and bug density with and without `get` and `set` methods.

d	size	w	p_value	significant	grp1	grp2	density type
0.125	negligible	1581	0.267	no	get-set-methods-present	get-set-methods-filtered	revision density
0.138	negligible	1598	0.222	no	get-set-methods-present	get-set-methods-filtered	bug density

The table below shows the Cliff's δ and Mann-Whitney U test result between density values of small, medium and large methods with and without `get` and `set` methods.

d	size	w	p_value	significant	grp1 (no get and set)	grp2 (get and set present)	density type
0.012	negligible	521384468694	0	yes	small	small	revision density
0	negligible	1329870007	0.993	no	medium	medium	revision density
0	negligible	241700481	0.972	no	large	large	revision density
0.012	negligible	521643495268	0	yes	small	small	bug density
0	negligible	1329944035	0.993	no	medium	medium	bug density
0	negligible	241671869.5	0.941	no	large	large	bug density

Appendix D

RQ3-Age Control

This appendix includes Cliff's δ and Mann-Whitney U test results of SLOC with #Revisions and #Bugs for different time intervals.

D.1 Correlation of SLOC with #Revisions at Different Times

The table below shows the Cliff's δ and Mann-Whitney U test results of comparing correlation coefficient of SLOC wih #Revisions at different times.

d	size	w	p_value	significant	grp1	grp2	correlation group
-0.408	medium	831.5	0	yes	No Age Normalization	0.5 yrs	SLOC_with_Revisions
-0.391	medium	856	0.001	yes	No Age Normalization	1 yr	SLOC_with_Revisions
-0.362	medium	896	0.001	yes	No Age Normalization	2 yrs	SLOC_with_Revisions
-0.245	small	1061	0.03	yes	No Age Normalization	3 yrs	SLOC_with_Revisions
-0.126	negligible	1227	0.263	no	No Age Normalization	4 yrs	SLOC_with_Revisions
-0.009	negligible	1392.5	0.942	no	0.5 yrs	1 yr	SLOC_with_Revisions
0.025	negligible	1439	0.83	no	0.5 yrs	2 yrs	SLOC_with_Revisions
0.139	negligible	1599.5	0.219	no	0.5 yrs	3 yrs	SLOC_with_Revisions
0.219	small	1712	0.052	no	0.5 yrs	4 yrs	SLOC_with_Revisions
0.033	negligible	1451.5	0.769	no	1 yr	2 yrs	SLOC_with_Revisions
0.132	negligible	1590.5	0.241	no	1 yr	3 yrs	SLOC_with_Revisions
0.214	small	1704.5	0.058	no	1 yr	4 yrs	SLOC_with_Revisions
0.105	negligible	1552.5	0.351	no	2 yrs	3 yrs	SLOC_with_Revisions
0.186	small	1666	0.099	no	2 yrs	4 yrs	SLOC_with_Revisions
0.091	negligible	1532	0.422	no	3 yrs	4 yrs	SLOC_with_Revisions

D.2 Correlation of SLOC with #Bugs at Different Times

The table below shows the Cliff's δ and Mann-Whitney U test results of comparing correlation coefficient of SLOC wih #Bugs at different times.

d	size	w	p_value	significant	grp1	grp2	correlation group
-0.175	small	1158.5	0.12	no	No Age Normalization	0.5 yrs	SLOC_with_Bugs
-0.242	small	1064	0.031	yes	No Age Normalization	1 yr	SLOC_with_Bugs
-0.257	small	1043	0.022	yes	No Age Normalization	2 yrs	SLOC_with_Bugs
-0.235	small	1075	0.037	yes	No Age Normalization	3 yrs	SLOC_with_Bugs
-0.151	small	1193	0.182	no	No Age Normalization	4 yrs	SLOC_with_Bugs
-0.075	negligible	1298.5	0.504	no	0.5 yrs	1 yr	SLOC_with_Bugs
-0.094	negligible	1272.5	0.405	no	0.5 yrs	2 yrs	SLOC_with_Bugs
-0.067	negligible	1310	0.552	no	0.5 yrs	3 yrs	SLOC_with_Bugs
0.016	negligible	1427.5	0.887	no	0.5 yrs	4 yrs	SLOC_with_Bugs
-0.028	negligible	1365	0.805	no	1 yr	2 yrs	SLOC_with_Bugs
-0.006	negligible	1396.5	0.962	no	1 yr	3 yrs	SLOC_with_Bugs
0.09	negligible	1530.5	0.427	no	1 yr	4 yrs	SLOC_with_Bugs
0.021	negligible	1434.5	0.852	no	2 yrs	3 yrs	SLOC_with_Bugs
0.109	negligible	1557	0.336	no	2 yrs	4 yrs	SLOC_with_Bugs
0.078	negligible	1514.5	0.488	no	3 yrs	4 yrs	SLOC_with_Bugs

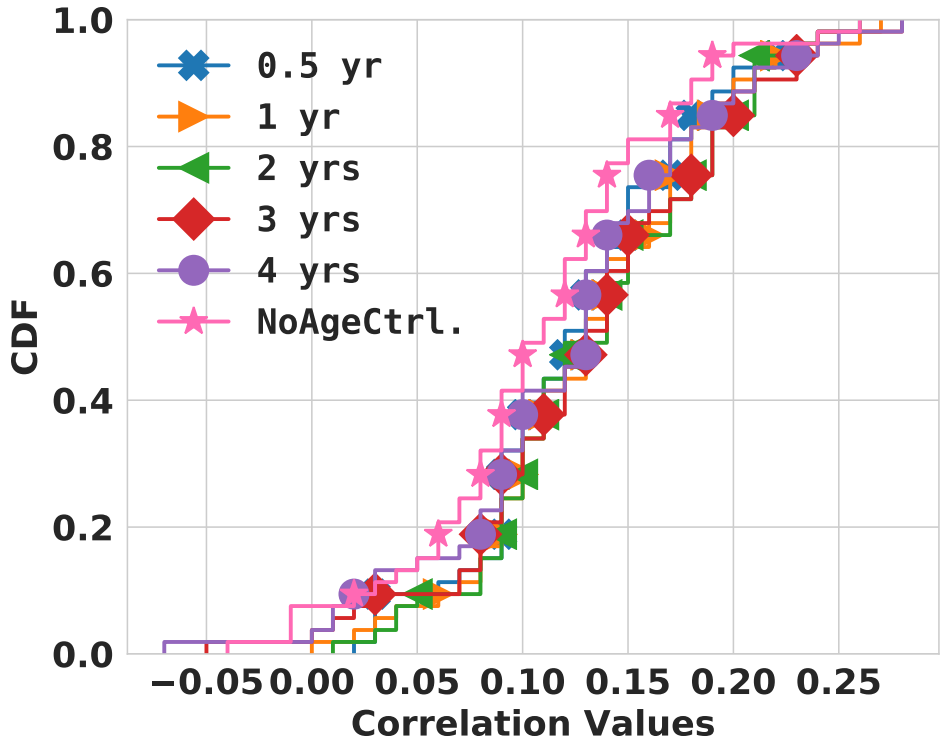


Figure D.1: Correlation coefficients of SLOC with #Bugs at different times compared with no age control group.

D.3 Comparing Correlation Values of SLOC with #Revisions at Different Choice of Time Intervals

The table below shows the Cliff's δ and Mann-Whitney U test results of comparing correlation coefficient of SLOC with #Revisions at different time intervals.

d	size	w	p_value	significant	grp1	grp2
0.843	large	2588	0	yes	sloc-revisions-0-0.5	sloc-revisions-0.5-1
0.813	large	2547	0	yes	sloc-revisions-0-0.5	sloc-revisions-1-2
0.839	large	2534.5	0	yes	sloc-revisions-0-0.5	sloc-revisions-2-3
0.257	small	2319	0	yes	sloc-revisions-0-0.5	sloc-revisions-3-4
-0.296	small	2279	0	yes	sloc-revisions-0-0.5	sloc-revisions-4+
-0.413	medium	1415	0.445	no	sloc-revisions-3-4	sloc-revisions-4+
-0.208	small	1112	0.065	no	sloc-revisions-0.5-1	sloc-revisions-1-2
0.035	negligible	1426.5	0.758	no	sloc-revisions-0.5-1	sloc-revisions-2-3
-0.199	small	1166	0.229	no	sloc-revisions-0.5-1	sloc-revisions-3-4
-0.416	medium	1324	0.861	no	sloc-revisions-0.5-1	sloc-revisions-4+
0.235	small	1701.5	0.038	yes	sloc-revisions-1-2	sloc-revisions-2-3
-0.073	negligible	1363.5	0.94	no	sloc-revisions-1-2	sloc-revisions-3-4
-0.36	medium	1590.5	0.121	no	sloc-revisions-1-2	sloc-revisions-4+
-0.242	small	1103.5	0.143	no	sloc-revisions-2-3	sloc-revisions-3-4
-0.434	medium	1246.5	0.602	no	sloc-revisions-2-3	sloc-revisions-4+

D.4 Comparing Correlation Values of SLOC with #Bugs at Different Choice of Time Intervals

The table below shows the Cliff's δ and Mann-Whitney U test results of comparing correlation coefficient of SLOC wih #Bugs at different time intervals.

d	size	w	p_value	significant	grp1	grp2
0.538	large	2159.5	0	yes	sloc-bugs-0-0.5	sloc-bugs-0.5-1
0.53	large	2148.5	0	yes	sloc-bugs-0-0.5	sloc-bugs-1-2
0.564	large	2155.5	0	yes	sloc-bugs-0-0.5	sloc-bugs-2-3
0.615	large	2182	0	yes	sloc-bugs-0-0.5	sloc-bugs-3-4
-0.228	small	1845	0.001	yes	sloc-bugs-0-0.5	sloc-bugs-4+
-0.444	medium	1070.5	0.124	no	sloc-bugs-3-4	sloc-bugs-4+
0.009	negligible	1417.5	0.937	no	sloc-bugs-0.5-1	sloc-bugs-1-2
0.013	negligible	1396.5	0.908	no	sloc-bugs-0.5-1	sloc-bugs-2-3
0.12	negligible	1514	0.292	no	sloc-bugs-0.5-1	sloc-bugs-3-4
-0.374	medium	1238.5	0.464	no	sloc-bugs-0.5-1	sloc-bugs-4+
0.02	negligible	1405.5	0.862	no	sloc-bugs-1-2	sloc-bugs-2-3
0.119	negligible	1512.5	0.296	no	sloc-bugs-1-2	sloc-bugs-3-4
-0.385	medium	1239	0.466	no	sloc-bugs-1-2	sloc-bugs-4+
0.089	negligible	1444	0.438	no	sloc-bugs-2-3	sloc-bugs-3-4
-0.387	medium	1177	0.327	no	sloc-bugs-2-3	sloc-bugs-4+

Appendix E

RQ4-Change Analysis

This appendix includes the correlation coefficient values of SLOC with different kinds of changes for 53 projects.

corr	p_value	significant	group	repo	change kind
-0.02	0.002308031025	yes	versionedSLOC_with_revisions	antlr4	revisions
0.18	0	yes	versionedSLOC_with_revisions	assertj-core	revisions
0.05	6.32E-08	yes	versionedSLOC_with_revisions	astyanax	revisions
0.36	0	yes	versionedSLOC_with_revisions	atmosphere	revisions
0.29	0	yes	versionedSLOC_with_revisions	drill	revisions
0.26	0	yes	versionedSLOC_with_revisions	DSpace	revisions
0.31	1.76E-219	yes	versionedSLOC_with_revisions	Essentials	revisions
0.25	0	yes	versionedSLOC_with_revisions	flink	revisions
0.27	7.88E-131	yes	versionedSLOC_with_revisions	flyway	revisions
0.22	0	yes	versionedSLOC_with_revisions	guava	revisions
0.23	3.94E-62	yes	versionedSLOC_with_revisions	hawtio	revisions
0.27	5.37E-100	yes	versionedSLOC_with_revisions	hector	revisions
0.12	0	yes	versionedSLOC_with_revisions	hibernate-orm	revisions
0.35	1.13E-259	yes	versionedSLOC_with_revisions	Hystrix	revisions
0.34	0	yes	versionedSLOC_with_revisions	jetty.project	revisions
0.19	3.81E-224	yes	versionedSLOC_with_revisions	logging-log4j2	revisions
0.25	3.85E-136	yes	versionedSLOC_with_revisions	lombok	revisions
0.16	4.52E-186	yes	versionedSLOC_with_revisions	mongo-java-driver	revisions
0.31	4.39E-193	yes	versionedSLOC_with_revisions	mybatis-3	revisions
0.31	0	yes	versionedSLOC_with_revisions	netty	revisions
0.14	5.67E-144	yes	versionedSLOC_with_revisions	openmrs-core	revisions
0.25	5.71E-83	yes	versionedSLOC_with_revisions	rabbitmq-java-client	revisions
0.26	0	yes	versionedSLOC_with_revisions	RxJava	revisions
0.28	0	yes	versionedSLOC_with_revisions	sonarqube	revisions
0.27	0	yes	versionedSLOC_with_revisions	soot	revisions
0.22	9.30E-24	yes	versionedSLOC_with_revisions	spark	revisions
0.19	0	yes	versionedSLOC_with_revisions	spring-framework	revisions
0.18	7.07E-92	yes	versionedSLOC_with_revisions	swagger-core	revisions
0.18	5.00E-62	yes	versionedSLOC_with_revisions	truth	revisions
0.15	7.39E-39	yes	versionedSLOC_with_revisions	Twitter4J	revisions
0.35	0	yes	versionedSLOC_with_revisions	voldemort	revisions
0.18	5.50E-59	yes	versionedSLOC_with_revisions	vraptor4	revisions
0.33	0	yes	versionedSLOC_with_revisions	wildfly	revisions
0.28	0	yes	versionedSLOC_with_revisions	ant	revisions
0.23	1.79E-251	yes	versionedSLOC_with_revisions	checkstyle	revisions
0.09	5.60E-11	yes	versionedSLOC_with_revisions	commons-io	revisions
0.13	5.13E-69	yes	versionedSLOC_with_revisions	commons-lang	revisions
0.22	3.46E-109	yes	versionedSLOC_with_revisions	cucumber-jvm	revisions
0.29	0	yes	versionedSLOC_with_revisions	docx4j	revisions
0.38	0	yes	versionedSLOC_with_revisions	fastjson	revisions
0.18	3.14E-38	yes	versionedSLOC_with_revisions	gson	revisions
0.11	9.77E-102	yes	versionedSLOC_with_revisions	hibernate-search	revisions
0.09	1.36E-64	yes	versionedSLOC_with_revisions	javaparser	revisions

0.26	0.00E-02	yes	versionedSLOC_with_revisions	jgit	revisions
0.11	3.09E-21	yes	versionedSLOC_with_revisions	jna	revisions
0.03	0.007717160702	yes	versionedSLOC_with_revisions	junit4	revisions
0.12	7.35E-56	yes	versionedSLOC_with_revisions	junit5	revisions
0.1	1.09E-29	yes	versionedSLOC_with_revisions	mockito	revisions
0.15	1.12E-66	yes	versionedSLOC_with_revisions	okhttp	revisions
0.21	0	yes	versionedSLOC_with_revisions	pmd	revisions
0.15	4.39E-286	yes	versionedSLOC_with_revisions	spring-boot	revisions
0.34	0	yes	versionedSLOC_with_revisions	titan	revisions
0.08	1.95E-103	yes	versionedSLOC_with_revisions	wicket	revisions
0.02	9.59E-06	yes	versionedSLOC_with_essential_changes	antlr4	essential
0.26	0	yes	versionedSLOC_with_essential_changes	assertj-core	essential
0.22	7.07E-109	yes	versionedSLOC_with_essential_changes	astyanax	essential
0.41	0	yes	versionedSLOC_with_essential_changes	atmosphere	essential
0.38	0	yes	versionedSLOC_with_essential_changes	drill	essential
0.4	0	yes	versionedSLOC_with_essential_changes	DSpace	essential
0.43	0	yes	versionedSLOC_with_essential_changes	Essentials	essential
0.38	0	yes	versionedSLOC_with_essential_changes	flink	essential
0.36	1.85E-221	yes	versionedSLOC_with_essential_changes	flyway	essential
0.21	0	yes	versionedSLOC_with_essential_changes	guava	essential
0.32	9.63E-121	yes	versionedSLOC_with_essential_changes	hawtio	essential
0.37	6.72E-173	yes	versionedSLOC_with_essential_changes	hector	essential
0.32	0	yes	versionedSLOC_with_essential_changes	hibernate-orm	essential
0.45	0	yes	versionedSLOC_with_essential_changes	Hystrix	essential
0.42	0	yes	versionedSLOC_with_essential_changes	jetty.project	essential
0.3	0	yes	versionedSLOC_with_essential_changes	logging-log4j2	essential
0.37	1.11E-282	yes	versionedSLOC_with_essential_changes	lombok	essential
0.3	0	yes	versionedSLOC_with_essential_changes	mongo-java-driver	essential
0.38	4.61E-269	yes	versionedSLOC_with_essential_changes	mybatis-3	essential
0.37	0	yes	versionedSLOC_with_essential_changes	netty	essential
0.33	0	yes	versionedSLOC_with_essential_changes	openmrs-core	essential
0.38	7.02E-185	yes	versionedSLOC_with_essential_changes	rabbitmq-java-client	essential
0.27	0	yes	versionedSLOC_with_essential_changes	RxJava	essential
0.36	0	yes	versionedSLOC_with_essential_changes	sonarqube	essential
0.31	0	yes	versionedSLOC_with_essential_changes	soot	essential
0.28	1.08E-35	yes	versionedSLOC_with_essential_changes	spark	essential
0.32	0	yes	versionedSLOC_with_essential_changes	spring-framework	essential
0.39	0	yes	versionedSLOC_with_essential_changes	swagger-core	essential
0.26	1.89E-122	yes	versionedSLOC_with_essential_changes	truth	essential
0.38	3.00E-226	yes	versionedSLOC_with_essential_changes	Twitter4J	essential
0.43	0	yes	versionedSLOC_with_essential_changes	voldemort	essential
0.33	2.13E-181	yes	versionedSLOC_with_essential_changes	vraptor4	essential
0.41	0	yes	versionedSLOC_with_essential_changes	wildfly	essential
0.37	0	yes	versionedSLOC_with_essential_changes	ant	essential

0.36	0	yes	versionedSLOC_with_essential_changes	checkstyle	essential
0.23	2.63E-57	yes	versionedSLOC_with_essential_changes	commons-io	essential
0.21	7.63E-164	yes	versionedSLOC_with_essential_changes	commons-lang	essential
0.3	1.83E-198	yes	versionedSLOC_with_essential_changes	cucumber-jvm	essential
0.28	0	yes	versionedSLOC_with_essential_changes	docx4j	essential
0.4	0	yes	versionedSLOC_with_essential_changes	fastjson	essential
0.24	4.65E-63	yes	versionedSLOC_with_essential_changes	gson	essential
0.32	0	yes	versionedSLOC_with_essential_changes	hibernate-search	essential
0.18	2.45E-235	yes	versionedSLOC_with_essential_changes	javaparser	essential
0.36	0	yes	versionedSLOC_with_essential_changes	jgit	essential
0.29	6.93E-127	yes	versionedSLOC_with_essential_changes	jna	essential
0.11	1.96E-17	yes	versionedSLOC_with_essential_changes	junit4	essential
0.24	5.79E-207	yes	versionedSLOC_with_essential_changes	junit5	essential
0.19	1.08E-91	yes	versionedSLOC_with_essential_changes	mockito	essential
0.32	1.20E-272	yes	versionedSLOC_with_essential_changes	okhttp	essential
0.3	0	yes	versionedSLOC_with_essential_changes	pmd	essential
0.26	0	yes	versionedSLOC_with_essential_changes	spring-boot	essential
0.37	0	yes	versionedSLOC_with_essential_changes	titan	essential
0.26	0	yes	versionedSLOC_with_essential_changes	wicket	essential
0.05	2.34E-18	yes	versionedSLOC_with_bodychangess	antlr4	body
0.28	0	yes	versionedSLOC_with_bodychangess	assertj-core	body
0.24	3.76E-129	yes	versionedSLOC_with_bodychangess	astyanax	body
0.44	0	yes	versionedSLOC_with_bodychangess	atmosphere	body
0.41	0	yes	versionedSLOC_with_bodychangess	drill	body
0.42	0	yes	versionedSLOC_with_bodychangess	DSpace	body
0.47	0	yes	versionedSLOC_with_bodychangess	Essentials	body
0.4	0	yes	versionedSLOC_with_bodychangess	flink	body
0.39	6.92E-254	yes	versionedSLOC_with_bodychangess	flyway	body
0.23	0	yes	versionedSLOC_with_bodychangess	guava	body
0.33	1.17E-124	yes	versionedSLOC_with_bodychangess	hawtio	body
0.39	6.14E-189	yes	versionedSLOC_with_bodychangess	hector	body
0.33	0	yes	versionedSLOC_with_bodychangess	hibernate-orm	body
0.48	0	yes	versionedSLOC_with_bodychangess	Hystrix	body
0.43	0	yes	versionedSLOC_with_bodychangess	jetty.project	body
0.38	0	yes	versionedSLOC_with_bodychangess	logging-log4j2	body
0.43	0	yes	versionedSLOC_with_bodychangess	lombok	body
0.35	0	yes	versionedSLOC_with_bodychangess	mongo-java-driver	body
0.38	4.51E-272	yes	versionedSLOC_with_bodychangess	mybatis-3	body
0.41	0	yes	versionedSLOC_with_bodychangess	netty	body
0.34	0	yes	versionedSLOC_with_bodychangess	openmrs-core	body
0.41	6.52E-207	yes	versionedSLOC_with_bodychangess	rabbitmq-java-client	body
0.28	0	yes	versionedSLOC_with_bodychangess	RxJava	body
0.38	0	yes	versionedSLOC_with_bodychangess	sonarqube	body
0.32	0	yes	versionedSLOC_with_bodychangess	soot	body

0.32	5.94E-46	yes	versionedSLOC_with_bodychangess	spark	body
0.35	0	yes	versionedSLOC_with_bodychangess	spring-framework	body
0.42	0	yes	versionedSLOC_with_bodychangess	swagger-core	body
0.3	1.99E-162	yes	versionedSLOC_with_bodychangess	truth	body
0.44	1.28E-301	yes	versionedSLOC_with_bodychangess	Twitter4J	body
0.45	0	yes	versionedSLOC_with_bodychangess	voldemort	body
0.34	7.54E-194	yes	versionedSLOC_with_bodychangess	vraptor4	body
0.43	0	yes	versionedSLOC_with_bodychangess	wildfly	body
0.39	0	yes	versionedSLOC_with_bodychangess	ant	body
0.37	0	yes	versionedSLOC_with_bodychangess	checkstyle	body
0.31	5.07E-96	yes	versionedSLOC_with_bodychangess	commons-io	body
0.25	2.12E-241	yes	versionedSLOC_with_bodychangess	commons-lang	body
0.34	2.67E-255	yes	versionedSLOC_with_bodychangess	cucumber-jvm	body
0.3	0	yes	versionedSLOC_with_bodychangess	docx4j	body
0.4	0	yes	versionedSLOC_with_bodychangess	fastjson	body
0.26	1.27E-72	yes	versionedSLOC_with_bodychangess	gson	body
0.36	0	yes	versionedSLOC_with_bodychangess	hibernate-search	body
0.22	0	yes	versionedSLOC_with_bodychangess	javaparser	body
0.38	0	yes	versionedSLOC_with_bodychangess	jgit	body
0.33	3.31E-162	yes	versionedSLOC_with_bodychangess	jna	body
0.19	3.46E-54	yes	versionedSLOC_with_bodychangess	junit4	body
0.29	1.10E-278	yes	versionedSLOC_with_bodychangess	junit5	body
0.22	4.67E-114	yes	versionedSLOC_with_bodychangess	mockito	body
0.36	0	yes	versionedSLOC_with_bodychangess	okhttp	body
0.32	0	yes	versionedSLOC_with_bodychangess	pmd	body
0.3	0	yes	versionedSLOC_with_bodychangess	spring-boot	body
0.41	0	yes	versionedSLOC_with_bodychangess	titan	body
0.3	0	yes	versionedSLOC_with_bodychangess	wicket	body
-0.1	4.31E-74	yes	versionedSLOC_with_nonessential_changes	antlr4	non-essential
-0.01	0.002880093411	yes	versionedSLOC_with_nonessential_changes	assertj-core	non-essential
-0.1	7.26E-24	yes	versionedSLOC_with_nonessential_changes	astyanax	non-essential
0.01	0.4306766575	no	versionedSLOC_with_nonessential_changes	atmosphere	non-essential
-0.05	4.45E-31	yes	versionedSLOC_with_nonessential_changes	drill	non-essential
-0.05	1.46E-16	yes	versionedSLOC_with_nonessential_changes	DSpace	non-essential
-0.17	1.62E-56	yes	versionedSLOC_with_nonessential_changes	Essentials	non-essential
-0.06	3.02E-76	yes	versionedSLOC_with_nonessential_changes	flink	non-essential
-0.04	0.0004656830356	yes	versionedSLOC_with_nonessential_changes	flyway	non-essential
0.09	3.21E-149	yes	versionedSLOC_with_nonessential_changes	guava	non-essential
-0.02	0.2023971245	no	versionedSLOC_with_nonessential_changes	hawtio	non-essential
0	0.7892317663	no	versionedSLOC_with_nonessential_changes	hector	non-essential
-0.04	1.23E-37	yes	versionedSLOC_with_nonessential_changes	hibernate-orm	non-essential
-0.03	0.002523621207	yes	versionedSLOC_with_nonessential_changes	Hystrix	non-essential
0.02	2.73E-07	yes	versionedSLOC_with_nonessential_changes	jetty.project	non-essential
-0.04	5.14E-09	yes	versionedSLOC_with_nonessential_changes	logging-log4j2	non-essential

-0.09	6.88E-18	yes	versionedSLOC_with_nonessential_changes	lombok	non-essential
-0.08	4.13E-48	yes	versionedSLOC_with_nonessential_changes	mongo-java-driver	non-essential
0	0.6957487754	no	versionedSLOC_with_nonessential_changes	mybatis-3	non-essential
-0.02	3.27E-05	yes	versionedSLOC_with_nonessential_changes	netty	non-essential
-0.04	1.30E-12	yes	versionedSLOC_with_nonessential_changes	openmrs-core	non-essential
-0.08	3.19E-10	yes	versionedSLOC_with_nonessential_changes	rabbitmq-java-client	non-essential
0.18	3.01E-272	yes	versionedSLOC_with_nonessential_changes	RxJava	non-essential
-0.03	1.60E-33	yes	versionedSLOC_with_nonessential_changes	sonarqube	non-essential
0.03	2.08E-10	yes	versionedSLOC_with_nonessential_changes	soot	non-essential
-0.03	0.1246315152	no	versionedSLOC_with_nonessential_changes	spark	non-essential
-0.06	2.85E-94	yes	versionedSLOC_with_nonessential_changes	spring-framework	non-essential
-0.02	0.006359889033	yes	versionedSLOC_with_nonessential_changes	swagger-core	non-essential
-0.07	9.60E-11	yes	versionedSLOC_with_nonessential_changes	truth	non-essential
-0.14	4.54E-33	yes	versionedSLOC_with_nonessential_changes	Twitter4J	non-essential
-0.02	0.003534956553	yes	versionedSLOC_with_nonessential_changes	voldemort	non-essential
-0.1	1.58E-17	yes	versionedSLOC_with_nonessential_changes	vraptor4	non-essential
-0.01	1.02E-05	yes	versionedSLOC_with_nonessential_changes	wildfly	non-essential
0.04	1.49E-14	yes	versionedSLOC_with_nonessential_changes	ant	non-essential
-0.11	1.16E-48	yes	versionedSLOC_with_nonessential_changes	checkstyle	non-essential
-0.09	6.32E-10	yes	versionedSLOC_with_nonessential_changes	commons-io	non-essential
0	0.5692707434	no	versionedSLOC_with_nonessential_changes	commons-lang	non-essential
-0.03	0.002779883456	yes	versionedSLOC_with_nonessential_changes	cucumber-jvm	non-essential
0.15	5.92E-237	yes	versionedSLOC_with_nonessential_changes	docx4j	non-essential
0.12	1.83E-69	yes	versionedSLOC_with_nonessential_changes	fastjson	non-essential
-0.04	0.0103243555	yes	versionedSLOC_with_nonessential_changes	gson	non-essential
-0.17	9.75E-247	yes	versionedSLOC_with_nonessential_changes	hibernate-search	non-essential
-0.02	8.33E-05	yes	versionedSLOC_with_nonessential_changes	javaparser	non-essential
-0.06	3.26E-18	yes	versionedSLOC_with_nonessential_changes	jgit	non-essential
-0.12	1.55E-22	yes	versionedSLOC_with_nonessential_changes	jna	non-essential
-0.05	1.21E-05	yes	versionedSLOC_with_nonessential_changes	junit4	non-essential
-0.07	2.66E-18	yes	versionedSLOC_with_nonessential_changes	junit5	non-essential
-0.06	4.68E-10	yes	versionedSLOC_with_nonessential_changes	mockito	non-essential
-0.06	4.12E-10	yes	versionedSLOC_with_nonessential_changes	okhttp	non-essential
0.01	0.06654157488	no	versionedSLOC_with_nonessential_changes	pmd	non-essential
-0.03	1.85E-16	yes	versionedSLOC_with_nonessential_changes	spring-boot	non-essential
0.06	1.37E-10	yes	versionedSLOC_with_nonessential_changes	titan	non-essential
-0.07	3.70E-70	yes	versionedSLOC_with_nonessential_changes	wicket	non-essential