You Can't Control the Unfamiliar: A Study on the Relations Between Aggregation Techniques for Software Metrics

Bogdan Vasilescu, Alexander Serebrenik, Mark van den Brand
Technische Universiteit Eindhoven,
Den Dolech 2, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
{b.n.vasilescu@student., a.serebrenik@, m.g.j.v.d.brand@}tue.nl

***Abstract*—A popular approach to assessing software maintainability and predicting its evolution involves collecting and analyzing software metrics. However, metrics are usually defined on a micro-level (method, class, package), and should therefore be aggregated in order to provide insights in the evolution at the macro-level (system). In addition to traditional aggregation techniques such as the *mean*, *median*, or *sum*, recently econometric aggregation techniques, such as the Gini, Theil, Kolm, Atkinson, and Hoover inequality indices have been proposed and applied to software metrics.**

In this paper we present the results of an extensive correlation study of the most widely-used traditional and econometric aggregation techniques, applied to lifting SLOC values from class to package level in the 106 systems comprising the Qualitas Corpus. Moreover, we investigate the nature of this relation, and study its evolution on a subset of 12 systems from the Qualitas Corpus.

Our results indicate high and statistically significant correlation between the Gini, Theil, Atkinson, and Hoover indices, i.e., aggregation values obtained using these techniques convey the same information. However, we discuss some of the rationale behind choosing between one index or another.

I. Introduction

Software maintenance is an area of software engineering with deep financial implications. Indeed, it was reported that between 60% and 90% of the software budgets represent maintenance and evolution costs [1]–[3]. Furthermore, maintenance and evolution costs were forecasted to account for more than half of North American and European software budgets in 2010 [4]. Similar or even higher figures were reported for countries such as Norway [5] and Chile [6].

Controlling software maintenance costs requires predicting how the system will evolve in the future, which in turn requires a better understanding of software evolution [7]–[9]. A popular approach to assessing software maintainability and predicting its evolution involves performing measurements on code artifacts. It starts off by identifying a number of specific properties of the system under investigation, and then collecting the corresponding software metrics and analyzing their evolution. Although it is debatable whether one cannot control what one cannot measure, it is without a doubt that collecting and analyzing metrics helps increase one's familiarity and understanding of the analyzed systems.

However, metrics are usually defined at micro level (method, class, package), while the analysis of maintainability and evolution requires insights at macro (system) level. Moreover, due to privacy reasons, it might be undesirable to disclose metrics pertaining to a single developer as opposed to those pertaining to the entire project [10]. Metrics should therefore be aggregated [11].

Popular aggregation techniques include such standard summary statistical measures as *mean*, *median*, or *sum* [12], [13]. Their main advantage is universality (metrics-independence): whatever metrics are considered, the measures should be calculated in the same way. However, as the distribution of many interesting software metrics is skewed [14], the interpretation of such measures becomes unreliable [15].

Alternatively, *distribution fitting* [14], [16], [17] consists of selecting a known family of distributions (e.g., log-normal or exponential) and fitting its parameters to approximate the metric values observed. The fitted parameters can be then seen as aggregating these values. However, the fitting process should be repeated whenever a new metric is being considered. Moreover, it is still a matter of controversy whether, e.g., software size is distributed log-normally [16] or double Pareto [18]. We do not consider distribution fitting.

Recently, there is an emerging trend in using more advanced aggregation techniques borrowed from econometrics, where they are used to study inequality of income or welfare distributions [19]–[21]. The motivation for applying such techniques to software metrics is twofold. First, as numerous countries have few rich and many poor, numerous software systems have few very big or complex components, and many small or simple ones [15], [22], [23]. Consequently, it is common both for software metrics, as well as for econometric variables to have strongly-skewed distributions (Figure 1).

Second, the shape of these distributions, which appear visually to follow a power law, renders the use of traditional aggregation techniques such as the sample mean and variance questionable at best. Indeed, it was reported that many important relationships between software artifacts follow a power-law distribution [16], [25], and it is known that a power-law distribution may not have a finite mean and variance [22].

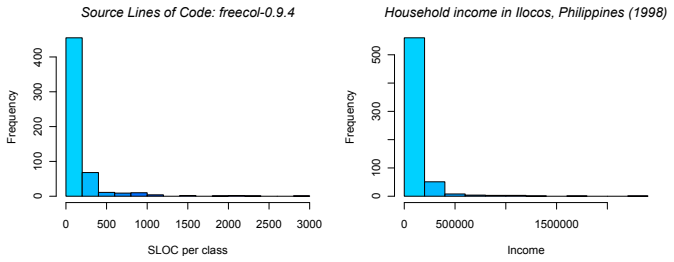

Fig. 1. Software metrics (SLOC) and econometric variables (household income in the Ilocos region, the Philippines [24]) have distributions with similar shapes.

These realizations led to the application of econometric techniques to aggregation of software metrics [15], [23], [26]–[28], and to our current interest in these aggregation techniques. In this paper we consider the *Gini*, *Theil*, *Atkinson*, *Hoover*, and *Kolm* indices, commonly used in econometrics to study inequality of income or welfare [19]–[21], and reliable for highly-skewed distributions such as the ones of source lines of code (SLOC) considered in this paper.

In two previous studies [26], [27], we have set the grounds for a theoretical and an empirical comparison of different aggregation techniques for software metrics. In this paper we build on [26], [27] and extend this work in two ways. First, we present the results of an extensive correlation study of the most widely-used traditional (*mean*, *median*, *sum*, *standard deviation*, *variance*, *skewness*, and *kurtosis*) and econometric (*Gini*, *Theil*, *Atkinson*, *Hoover*, and *Kolm*) aggregation techniques, applied to lifting SLOC values from class to package level in the 106 systems comprising the Qualitas Corpus [29]. Second, apart from measuring the strength of the correlation between the various aggregation techniques, we also investigate the nature of this relation, and study its evolution. Specifically, we address the following questions:

1) Which and to what extent do the inequality indices *agree*? Which and to what extent do the aggregation techniques rank distributions of SLOC values similarly?
2) What is the *nature of the relation* between the various aggregation techniques, i.e., does the scatter plot of the relation exhibit a clear shape?
3) How does the relation between the various aggregation techniques *change in time*, i.e., how does the correlation coefficient change as the systems evolve?

The remainder of this paper is organized as follows. Section II introduces the aggregation techniques considered, while Section III discusses related work. Section IV focuses on the methodology to collect and analyse the data. Section V studies agreement between different aggregation techniques and the nature of relation between them, i.e., Questions 1 and 2, while Section VI considers the change of this relation in time, i.e., Question 3. Finally, we present the conclusions and sketch directions for future work in Section VII.

II. Aggregating Software Metrics

In this section we introduce the aggregation techniques considered, and discuss their appropriateness for software metrics.

We consider two categories of aggregation techniques. The first category includes standard summary statistics such as additive measures (*sum*), central tendency measures (*mean*, *median*), statistical dispersion measures (*standard deviation*, *variance*), or distribution shape measures (*skewness*, *kurtosis*).

Let $X = \{x_1, \ldots, x_n\}$ be the collection of (metrics data) values to be aggregated. For *mean*, *sum*, and *median* we use standard definitions [26]. For standard deviation σ and variance *var* we compute the measure for a sample rather than the entire population [30]: $var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$, and $\sigma(X) = \sqrt{var(X)}$. Both the standard deviation and the variance are based on the mean, hence they also become unreliable for highly skewed distributions, where they do not convey information about the asymmetry. Moreover, albeit easily computable, the *sum* is unbounded, making relative comparisons difficult. On the other hand, the *mean* can be misleading for highly skewed distributions due to influence of outliers. The *median* is less sensitive to outliers, but can yield different results if a small change occurs in the data set, e.g., one value is removed.

The skewness and the kurtosis offer two more alternatives to aggregation techniques. *Skewness*, denoted as γ_1, measures the asymmetry of a distribution and is defined as $\gamma_1(X) = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i-\bar{x})^3}{\sigma^3(X)}$. In contrast, *kurtosis*, denoted as γ_2, measures the peakedness of a distribution, i.e., high kurtosis corresponds to a distribution with sharp peaks and long fat tails, while low kurtosis corresponds to a distribution with rounded peaks and short thin tails. Kurtosis is defined as $\gamma_2(X) = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i-\bar{x})^4}{\sigma^4(X)}$. Similar to the variance and standard deviation, the skewness and kurtosis are unbounded, hence cause difficulties when comparing systems with different population sizes [15].

The second category of aggregation techniques considered consists of inequality indices, commonly used to study inequality of income or welfare distributions [19]–[21]. Specifically, we consider the *Gini* [31], *Theil* [32], *Atkinson* [33], *Hoover* [34] (also known as the Ricci-Schutz coefficient, or the Robin Hood index), and *Kolm* [35] income inequality indices:

$$
\begin{aligned}
I_{\text{Gini}}(X) &= \frac{1}{2n^2\bar{x}}\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j| \\
I_{\text{Theil}}(X) &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i}{\bar{x}}\log\frac{x_i}{\bar{x}}\right) \\
I_{\text{Atkinson}}(X) &= 1 - \frac{1}{\bar{x}}\left(\frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i}\right)^2 \\
I_{\text{Hoover}}(X) &= \frac{1}{2}\sum_{i=1}^{n}\left|\frac{x_i}{\sum_{j=1}^{n}x_j} - \frac{1}{n}\right| \\
I_{\text{Kolm}}(X) &= \log\left[\frac{1}{n}\sum_{i=1}^{n}e^{\bar{x}-x_i}\right]
\end{aligned}
$$

where $|x|$ is the absolute value of x, I_{Theil} is the so-called first Theil index[1], and I_{Kolm} and I_{Atkinson} are standard instantiations of the Kolm and Atkinson families of indices, for parameter

[1]In addition to I_{Theil} above, Theil [32] has also introduced the second Theil index, known as the *mean logarithmic deviation* I_{MLD}, and defined as $I_{\text{MLD}}(X) = \frac{1}{n}\sum_{i=1}^{n}\left(\log\frac{\bar{x}}{x_i}\right)$. In this paper we do not consider I_{MLD} and whenever "the Theil index" is mentioned, I_{Theil} is meant.

values of 1 and 0.5, respectively. Mathematical properties of the inequality indices and implications of these properties on aggregation of software metrics have been discussed in [26].

In contrast with the traditional techniques, the econometric inequality indices can be used successfully to aggregate software metrics such as SLOC, since they provide a synthesis of the skewness, kurtosis, mean, and variance statistics of the data, while effectively capturing the nature of the software metric under skewed distributions [15].

In econometrics, such indices are used to measure the inequality of income or wealth distributions. For example, inequality indices applied to 138 countries show reductions in global inequality of GDP per capita during the 1980s and 1990s [36]. Some inequality indices such as I_{Theil} are *decomposable* and hence can also be used to explain the inequality [19]. For instance, inequality of expenditure in Indonesian households can be better explained by the education level of the head of the household rather than by the province of residence, or the gender of the household head [37]. Formally, we say that the inequality index I is decomposable if, for any given a partition of the population into mutually exclusive and completely exhaustive groups, the aggregation result computed at a population level is expressed as the sum of a non-negative *within-group* term and a non-negative *between-groups* term: the *within-group* contribution is itself a weighted sum of applying the same inequality index at the group level [38]. In econometrics, one commonly further requires that the sum of the weighting coefficients be 1.

Similarly, in software engineering, inequality indices applied to software metrics can be used to measure the degree of concentration of functionality (e.g., to identify packages with unevenly-sized classes), to reveal significant architectural shifts, or to indicate the presence of god classes or machine-generated code [15], [28]. Furthermore, using the Theil index it was observed that the partitioning of Debian Linux into packages provides the best explanation for the inequality in SLOC values, rather than the implementation language or the maintainer [28]. This means that if one would like to reduce this inequality, i.e., to distribute functionality across the units in a more egalitarian way, one should focus on establishing cross-package size guidelines first.

The Theil index, however, does not support negative values since $\log x$ is undefined if $x < 0$. Nevertheless, SLOC has only non-negative values, hence I_{Theil} as well as all other techniques considered here are appropriate under these circumstances. Special care is necessary with the Theil index in the presence of zero values, since $\log x$ is then also undefined. In [28] we have distinguished between the particular meaning of a value being zero. The usual approach in econometrics is to consider that a person with no income does not contribute to the income distribution, hence $I_{\text{Theil}}(x_1, \ldots, x_{n-1}, 0)$ should be defined as $I_{\text{Theil}}(x_1, \ldots, x_{n-1})$. Alternatively, for the sake of simplicity, one can replace 0 by a very small $\varepsilon > 0$, such that $I_{\text{Theil}}(x_1, \ldots, x_{n-1}, 0) \stackrel{\text{def}}{=} I_{\text{Theil}}(x_1, \ldots, x_{n-1}, \varepsilon)$. This observation can be generalized for an arbitrary number of zeros as long as at least one non-zero value is present. While both approaches are valid for software metrics where zero denotes emptiness (e.g., SLOC), in this paper we choose the latter and we take ε to be equal to 10^{-50}.

III. Related work

The term "aggregation of software metrics" can be understood in two ways: either as aggregation of values obtained by applying the different metrics to the same software artifacts, or as aggregation of values obtained by applying the same metrics to different software artifacts. Maintainability index (MI) [39] and modularization quality (MQ) [40] are examples of aggregating results of different metrics applied to the same software artefact: MI aggregates the Halstead's effort, McCabe's cyclomatic complexity, lines of code count and number of comment lines, while MQ aggregates intra-connectivity (cohesion) and inter-connectivity (coupling). Our work as well as the main body of papers discussed in this section pertain to aggregation of values obtained by applying the same metrics to different software artifacts.

Application of econometric inequality indices to software metrics has been first advocated in [15]. The authors proposed the *Gini* index as the aggregation technique as it is both universal and reliable, can be given an intuitive interpretation using the Lorenz curve, and ranges between 0 and 1. This approach has been successfully applied to study the getter and setter methods usage profiles in Java software [41]. While the *Gini* index has a number of advantages, it is not decomposable, and, hence, as realized in [28], the *Gini* index can be used to measure inequality but not to *explain* it. Therefore, *Theil* index, being universal, reliable and decomposable, has been proposed [28]. In addition to the *Gini* index and the *Theil* index, in their study of the Pareto principle evidence in open source software activity, Goeminne and Mens [23] have also applied the *Hoover* index. Considering multiple indices instead of one [23] follows an existing practice in economics. For instance, [42] employs six different indices, including the *Gini*, *Theil*, and *Atkinson* indices. Still, a natural question arises: when should one use each one of the aggregation techniques?

Champernowne [43] has applied six aggregation techniques to synthetic data. The author observed that different indices exhibit different sensitivity to different "dimensions of inequality": while $1 - n^{I_{\text{Theil}}}$ was most sensitive to inequality associated with the exceptionally rich, I_{Gini} is second-most sensitive to inequality reflecting a wide spread of the less extreme incomes, without much tendency for the majority of them to be bunched within quite a narrow range. A similar study of synthetic data mimicking small countries with relatively small population and a limited number of regions has been recently conducted in [44]. These works, however, do not consider real world data sets and, therefore, generality of the observations and conclusions derived require additional verification.

In a previous study [27], we have set the grounds for a theoretical and an empirical comparison of different aggregation techniques for software metrics (*mean*, as well as the *Gini*, *Theil*, *Atkinson*, and *Kolm* inequality indices). We have

observed on a single snapshot of a case study (ArgoUML) that the choice of aggregation technique matters, i.e., it influence the correlation between the aggregated values and a validation metric (in that case number of defects per package), and that the aggregation techniques fall into two groups (*mean* and *Kolm* on the one hand, and *Gini*, *Theil*, and *Atkinson* on the other hand), for which we observed high and statistically significant correlation among the methods in each group.

Later, in another study [26] we have investigated a single version from three case studies (ArgoUML, Adempiere, and Mogwai Java Tools) by means of the aforementioned aggregation techniques augmented by the *sum*, *median*, and *Hoover* inequality index. We have observed that indeed the choice of aggregation technique influences the correlation with defects and that, e.g., *mean* leads to very inconsistent correlation results. However, the separation of the techniques into two groups with high and statistically significant correlation among the elements in each was not as clear as before, and was not consistent across systems. Nevertheless, the *Gini*, *Theil*, *Atkinson*, and *Hoover* inequality indices showed high and statistically significant correlation among themselves, i.e., the aggregated values obtained using these techniques convey the same information.

Apart from the corresponding nature of economic and software societies, studies of software maintenance and econometrics show other important similarities. Indeed, similar differences between the economic behavior of the individual producer and consumer (microeconomics) and nations total economic behavior (macroeconomics) have been observed when comparing the evolution of individual applications with the evolution of compilations or distributions of applications [45]. Moreover, the common interest in evolutionary processes is witnessed by studies of software evolution [8], as well as of evolutionary economics [46] and econometrics [47].

Beyond the studies of software metrics aggregation by means of econometric inequality indices, new techniques for metrics aggregation have been proposed in Squale [11]. While Squale covers *both* aggregation of different metrics of the same artifact and aggregation of the same metric of different artifacts, in the coming discussion we focus only on the second form of aggregation. This form of aggregation is considered as a two-phase process in Squale. First, values of individual metrics are translated to *individual marks* such that clearly desirable values get the highest mark (3), and clearly undesirable values get the lowest mark (0). The translation function is chosen such that when a certain threshold is exceeded the individual mark decreases following an exponential curve: the individual mark tends quickly towards zero, stressing the presence of undesirable metric values. Second, individual marks are aggregated to the *global mark*, corresponding to the entire project: $-\log_\lambda(\frac{1}{n}\sum_{i=1}^{n}\lambda^{-x_i})$, where x_i $(1 \leq i \leq n)$ are individual marks, and λ reflects tolerance for bad individual marks (the authors consider $\lambda = 3$, $\lambda = 9$ and $\lambda = 30$ corresponding to high, intermediate and low tolerance, respectively). We consider the work of [11] as complementary to ours. As opposed to the econometric approaches, the aggregation technique of [11] is asymmetric, i.e., it labels metric values as being desirable or not, while inequality indices do not make this distinction. This also implies that for the Squale approach to be applicable threshold values for different metrics should be known. The Squale project has published such thresholds [48] but a more extensive threshold validation is desirable. Additional advantage of such inequality indices as I_{Theil} and I_{Kolm} consists in their decomposability and invariance. A more extensive comparison of the Squale approach with econometric indices is an ongoing collaboration effort between both teams.

IV. Methodology

To perform empirical evaluation of different aggregation techniques we conducted two series of experiments. In the first set of experiments (Section V) we investigated relation between pairs of aggregation techniques, i.e., we addressed Questions 1 and 2. As case studies we chose the 106 open-source Java systems comprising the Qualitas Corpus version 20101126r (Section IV-A). For each case study we determined the metrics data (SLOC) and aggregated it from class level to package level using all pairs of aggregation techniques. We stress that statistical correlations are not transitive [49], i.e., we have to consider *all pairs* of aggregation techniques[2].

An obvious threat to validity for such a study is the representativeness of the versions considered. In order to reduce this threat and to address Question 3, we performed a second set of experiments (Section VI), in which we investigated the evolution of the correlation between similar pairs of SLOC data collections, again aggregated from class to package level using all combinations of aggregation techniques. As case studies, we chose 12 open-source Java systems with more than 10 versions, which are part of the Qualitas Corpus version 20101126e (Section IV-A).

A. Qualitas Corpus Dataset

The Qualitas Corpus [29] is a curated collection of open-source Java software systems, intended to be used for empirical studies of code artifacts.

In this paper we consider the Corpus version 20101126[3], which comes in two main distributions. For our first study (Section V) we consider the "r" (recent) variant, which contains the most recent versions available at the time of release, from 106 systems ranging from *FitJava v1.1* (2 packages, 2240 SLOC) to *NetBeans v6.9.1* (3373 packages 1890536 SLOC).

Our second study uses the "e" (evolution) variant of the Qualitas Corpus 20101126, which contains all versions from 13 systems (out of the 106 systems) with 10 or more versions available, totaling 414 versions. We have excluded *Eclipse SDK* (represented by 35 versions) from the consideration, because for 34 out of the 35 versions there is only bytecode available, while we focus on SLOC. All other systems had the

[2]While [49] considered Pearson's correlation coefficient, their counterexample also shows lack of transitivity for Spearman's ρ and Kendall's τ.

[3]Qualitas Research Group, Qualitas Corpus Version 20101126, http://qualitascorpus.com. The University of Auckland, February 2009.

source code available for all versions. From here on we refer to the remaining twelve systems as the Evolution Corpus.

The most covered systems of the Evolution Corpus in terms of number of versions available are *Hibernate* (86 versions), *Azureus/Vuze* (51 versions), and *Weka* (49 versions), while the least covered three systems are *ArgoUML* (10 versions), *ANTLR* (18 versions), and *JMeter* (18 versions). In terms of size (in terms of number of packages), the Evolution Corpus ranges between 634 packages in *Hibernate v3.6.0-beta4* and 6 packages in *Ant v1.1*.

B. Data collection

For both the single snapshot study, as well as the evolutionary study, the source code for each version of each system was automatically processed, and first the list of packages, and then the list of classes contained in each package were built. Alternatively, one could have used the Qualitas Corpus metadata in order to extract such information. However, we have preferred to extract the metadata using our own tooling as it was reported in the release notes of the 20101126 version of the Corpus [50] that some of the previous metadata files contained incorrectly-computed values.

An important note is the distinction between source code of the actual system, and source code of third-party libraries. It is possible that some systems distribute original source code of third-party libraries (in a previous release of the Corpus [29] it was reported that, e.g., *Compiere* v250 distributes a copy of the source code of the Apache Element Construction Set), while others provide their own implementations of such libraries, i.e., they distribute modified versions of third-party libraries together with their own source code.

In this paper we focus on source code of the actual systems and we exclude libraries. The decision regarding what is identified as actual source code of a version, and what is considered third-party is documented and provided as metadata alongside the Corpus, i.e., a space-separated list of prefixes of packages of Java types which are considered as developed for the system. For example, for *ArgoUML* all packages with names prefixed by `org.argouml` are considered as source packages (e.g., `org.argouml.model.uml`), while all others are considered as externals (e.g., `org.apache.xerces`).

For all pairs of aggregation techniques compared, we considered packages containing at least 2 classes. This is motivated by the fact that, when applied to packages containing only one class, most of the traditional aggregation techniques (standard deviation, variance, skewness, and kurtosis) are undefined, and all inequality indices are equal to 0. For Qualitas Corpus 20101126r, the most affected systems by this filtering were some of the small ones, namely *IvataGroupware* v0.11.3 (lost 34 out of 81 packages), *Sandmark* v3.4 (lost 45 out of 123 packages), and Quilt v0.6-a-5 (lost 5 out of 14 packages). Over the entire Corpus 20101126r, 86.7% of all systems lost less than 20% of their packages.

For each package in each version of each system (in both studies), we aggregate the SLOC values of all the classes *directly* contained in that package, in turn, using each of the aggregation techniques considered. We say that a class C is directly contained in a package P if there exists no subpackage P' of P different from P such that C is contained in P'.

C. Data analysis

To measure correlation between values aggregated using different techniques we have a choice between linear or rank correlation coefficients.

Linear coefficients (e.g., Pearson [51]) are sensitive only to a linear relation between two variables. On the other hand, rank coefficients (e.g., Kendall [52] or Spearman [53]) are more robust to nonlinear relations since they only measure the extent to which an increase in one variable (not necessarily linear) corresponds to an increase in the other variable. Consequently, since the nature of the relation between different aggregation techniques is one of the objects of our study rather than an assumption, we use a rank coefficient. We opt for Kendall's rank correlation coefficient τ since Spearman's ρ is known to be difficult to interpret [54].

All computations were performed using R [55]. Next we describe the two series of experiments performed.

V. Studying the correlation between aggregation techniques

In the first series of experiments we study correlation between the aggregation techniques on a single snapshot of each system in the Corpus, and we answer Questions 1 and 2 from the introduction.

A. Which and to what extent do aggregation techniques agree?

We start by measuring the Kendall correlation between values aggregated using the inequality indices (Figure 2) across the entire Corpus. Note that the percentage of the systems in the Corpus for which the correlation value is statistically significant using the common threshold of 0.05 is displayed between parentheses below each boxplot, and the name of each aggregation technique is abbreviated to its first three letters. Moreover, we display horizontal dotted lines to indicate the median for each of the boxplots. To simplify comparison of the plots in Figure 2 with similar plots on Figures 3 to 7 we opt for the same scale for all figures.

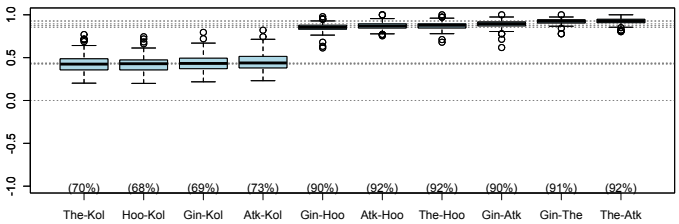

Fig. 2. Significant Kendall correlation between inequality indices.

We observe high and statistically significant correlation between *Theil*, *Gini*, *Atkinson*, and *Hoover* in more than 90% of the Corpus, i.e., aggregation values obtained using these techniques convey the same information. Correlation between *Kolm* and the other indices is average at best (0.4–0.5), and

significant for only approximately 70% of the Corpus. This confirms our observation in [27] and answers the first part of Question 1 from the introduction: I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} agree, i.e., they rank distributions of SLOC values similarly (there is high and statistically significant correlation between them).

Next we study how the other aggregation techniques correlate to each other and to the inequality indices. Since I_{Kolm} did not show high correlation with any of the other inequality indices, we are also interested in studying whether and to which traditional techniques I_{Kolm} correlates more.

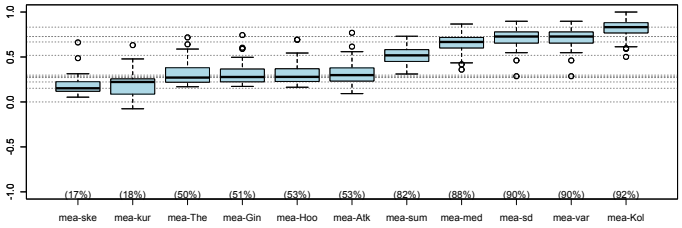

Fig. 3. Significant Kendall correlation between *mean* and the other aggregation techniques.

In Figure 3, *mean* shows very low significant correlation (0.3) with all the inequality indices except I_{Kolm} (in approximately 50% of the Corpus). The correlation between *mean* and I_{Kolm} is the highest (0.8) among all other techniques, and also statistically significant for 92% of the systems, i.e., aggregates obtained using these techniques convey the same information. Moreover, *mean* shows high (0.7–0.8) and statistically significant correlation for 90% of the Corpus with *median*, *standard deviation*, and *variance*.

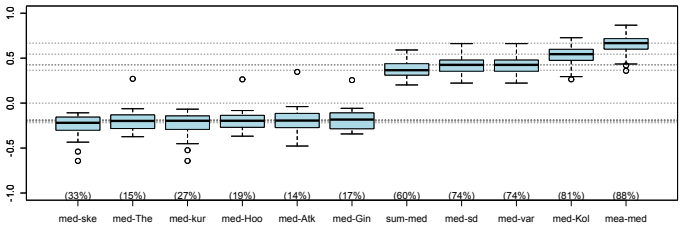

Fig. 4. Significant Kendall correlation between *median* and the other aggregation techniques.

In the case of *median* (Figure 4), the highest measured correlation is the one with *mean* (0.7 for 88% of the Corpus), and there is very low correlation (-0.2) with *skewness*, *kurtosis*, or either of I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson}.

Closer inspection at I_{Kolm} (Figure 5) reveals high (0.8) and statistically significant correlation in 90% of the Corpus with *mean*, *standard deviation*, and *variance*. It is interesting to observe that while *mean* shows high correlation with both *median* (0.7) and I_{Kolm} (0.8), the correlation between *median* and I_{Kolm} is lower (0.5–0.6).

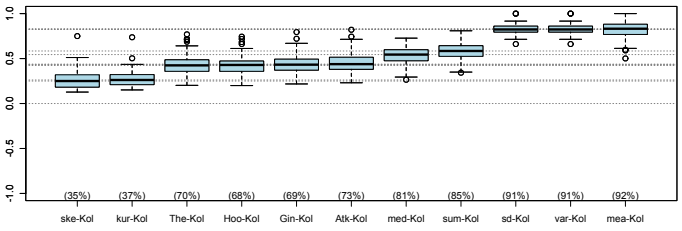

Fig. 5. Significant Kendall correlation between I_{Kolm} and the other aggregation techniques.

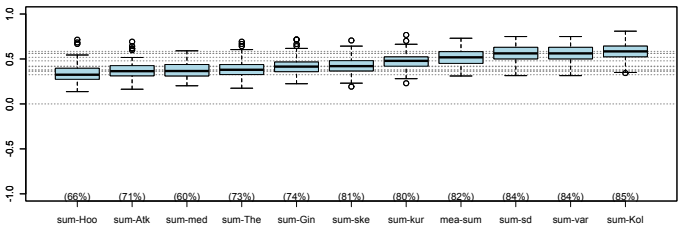

Fig. 6. Significant Kendall correlation between *sum* and the other aggregation techniques.

Sum shows at most average correlation (0.3–0.6) with *mean*, *kurtosis*, *standard deviation*, *variance*, and I_{Kolm} for approximately 80% of the Corpus (Figure 6).

The close mathematical relations between *standard deviation* and *variance*, as well as between *skewness* and *kurtosis* are reflected in the perfect (1) and high (0.8) correlation values from Figure 7, respectively.

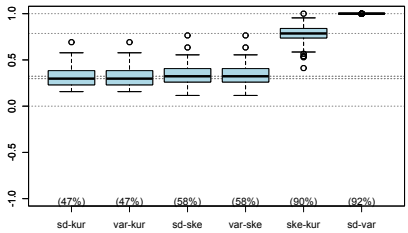

Fig. 7. Significant Kendall correlation between *standard deviation*, *variance*, *skewness*, and *kurtosis*.

Hence, to summarize our answer to Question 1 from the introduction, we note that:

- I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} show consistently high and statistically significant correlation between them, i.e., the aggregates obtained using these techniques convey the same information.
- The correlation between *mean* and I_{Kolm} is statistically significant, and the highest among all other techniques considered. Moreover, *mean* shows high and statistically significant correlation with *median*, *standard deviation*, and *variance*.
- *Median* shows high and statistically significant correlation with *mean*, while *sum* does not correlate with any of the other techniques. Aggregated values obtained using

standard deviation or *variance* on the one hand, or *skewness* and *kurtosis* on the other hand convey the same information.

B. What is the nature of the relation between aggregation techniques?

In order to study the nature of these relations, we draw scatter plots for each pair of aggregation techniques and each system in the Corpus and we analyze if the scatter plots exhibit a clear shape. In particular, we are interested in observing linear, superlinear, or chaotic patterns, although the Kendall *rank* correlation values previously computed are not sensitive to the linearity of these relations.

To illustrate the relation between values aggregated using the inequality indices, *Compiere* is a representative example[4]. In Figure 8 we distinguish a clear linear relation between I_{Theil} and I_{Atkinson}, which also confirms the highest measured correlation between these two indices among all indices considered.

The also high measured correlation between I_{Theil} and I_{Gini}, however, corresponds to a clear relation which visually exhibits superlinear rather than linear growth. This observation agrees with the econometric-based distinction between different *dimensions of inequality* [43], and the sensitivity of the different inequality indices to different such dimensions. For example, in econometrics one can distinguish between inequality due to extreme relative wealth, among the less extreme incomes, or due to extreme poverty [43]. In case of SLOC, extreme relative wealth, i.e., inequality associated with the exceptionally rich, corresponds to a non-egalitarian distribution of functionality caused by systems having few very big or complex components and many small or simple ones. Analogously, inequality due to extreme poverty is caused by systems having few very small components rather than few very big ones. Inequality among the less extreme incomes corresponds to a more uniform distribution of differences in functionality among the components of the system.

It is known, for example, that the *Theil* index is highly sensitive to inequality associated with the exceptionally rich, while the *Gini* coefficient is highly sensitive to inequality reflecting a wide spread of the less extreme incomes, without much tendency for the majority of them to be bunched within quite a narrow range [43], [56]. This results in a sharper increase in I_{Theil} as I_{Gini} increases, i.e., as the inequality between the "rich" and the "poor" increases, which is visible in Figure 8.

The high correlation values between I_{Theil} and I_{Hoover} are also supported by a relation similar to the one between I_{Theil} and I_{Gini}, which appears visually to be superlinear, although we observe more dispersion, i.e. disagreement, towards the "rich". On the other hand, the chaotic pattern is observed between I_{Theil} and I_{Kolm} (Figure 2).

Next we study the relation between *mean* and some of the other aggregation techniques with which it showed high correlation, using *JRE* as illustration (Figure 9). We observe a linear relation between *mean* and I_{Kolm}. For *median*, *standard deviation* and *variance* the relation is much more chaotic.

[4]The other systems in the Corpus exhibit similar patterns. For complete results see www.student.tue.nl/X/b.n.vasilescu/scatterPlots/scatter_SLOC.html

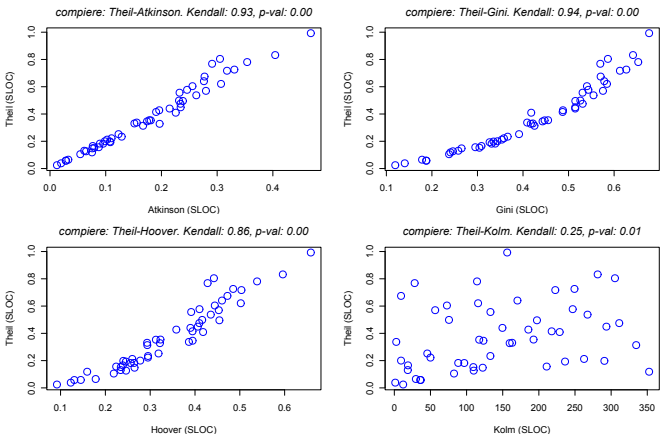

Fig. 8. Shape of the relation between I_{Theil} and each of I_{Atkinson}, I_{Gini}, I_{Hoover}, and I_{Kolm}.

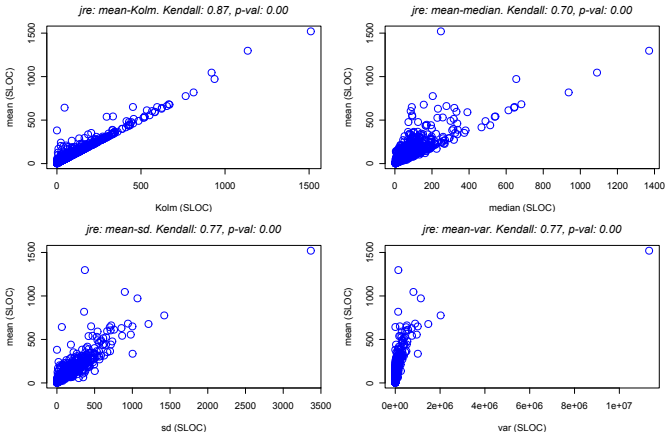

Fig. 9. Shape of the relation between *mean* and I_{Kolm} (top left), *median* (top right), *standard deviation* (bottom left), and *variance* (bottom right).

Finally, the close mathematical connections between *standard deviation* and *variance*, as well as between *skewness* and *kurtosis* are witnessed by clear superlinear patterns in Figure 10, using *Tomcat* as representative illustration.

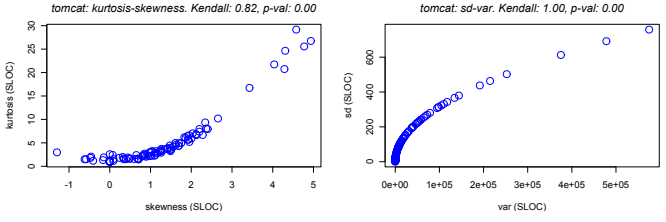

Fig. 10. Shape of the relation between *skewness* and *kurtosis* (left), and *standard deviation* and *variance* (right).

Hence to summarize our answer to Question 2 from the introduction, we note that linear, superlinear, as well as chaotic

patterns can be observed in the scatter plots. However high correlation values do not always correspond to clear-shaped relations (e.g., between *mean* and *standard deviation*). We observe that linear or superlinear relations always correspond to high correlation values, while chaotic patterns correspond to both high and average/low correlation values.

C. Which index to choose?

Answering Questions 1 and 2 from the introduction, allows us to provide a guideline when different aggregation techniques should be used. The only inequality index showing strong correlation with the mean is I_{Kolm}. Since the interpretation of the mean is known to become unreliable for skewed distributions [15], I_{Kolm} can be seen as more easily interpretable alternative.

It might seem that since I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} convey the same information, all these indices are equally appropriate. This is, however, not true as different indices have different application domains, emphasize different dimensions of inequality and possess different decomposability properties. Neither I_{Theil} nor I_{Atkinson} are applicable to negative values, while I_{Gini} and I_{Hoover} can be applied to any values as long as the mean of the values being aggregated differs from 0. We stress, however, that the range of I_{Gini} and I_{Hoover} becomes $\mathbb{R}$ in presence of negative values, challenging interpretation of the aggregated values. If the quality assessor believes presence of a few very large ("rich") modules to be undesirable, she should use I_{Theil} and I_{Atkinson} as these indices are most sensitive to the "rich". Alternatively, if she chooses to focus on deviations from a more uniform distribution of size among the components of the system, I_{Gini} and I_{Hoover} are more appropriate as they are more sensitive to mid-range inequality. Finally, if the inequality index is intended to be used to explain the inequality observed, the inequality index should be decomposable. I_{Theil} is the only decomposable index of the four and hence the only one that can be used to also explain inequality [28].

VI. Studying the Evolution of the Correlation between Aggregation Techniques

In the second series of experiments we study the evolution of the Kendall correlation between the aggregation techniques on the Evolution Corpus, and we answer Question 3 from the introduction. In order to better understand the evolution of the correlation, we employ two thresholds for statistical significance: we draw the correlation coefficients supported by two-sided p-values at most equal to 0.01 as *filled blue squares*, those supported by two-sided p-values between 0.01 and 0.05 as *empty blue squares*, and finally those supported by two-sided p-values above 0.05 as *empty blue triangles*.

In the previous section we have observed high and statistically significant correlation between I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson}. For the Evolution Corpus[5], *Weka* is a representative illustration (Figure 11), which shows that this observation holds across all versions, i.e., the correlation coefficient between

[5]For complete results see www.student.tue.nl/X/b.n.vasilescu/evolution/SLOC.html

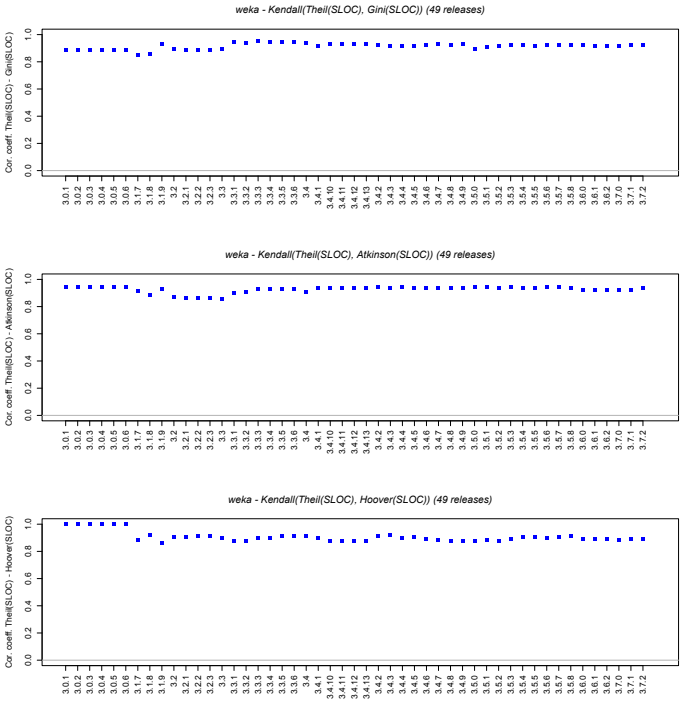

Fig. 11. Evolution of Kendall correlation between I_{Theil} and I_{Gini}, I_{Atkinson}, and I_{Hoover}, respectively.

I_{Theil} and each of I_{Gini}, I_{Atkinson}, and I_{Hoover} does not drop below 0.8 and is always statistically very significant.

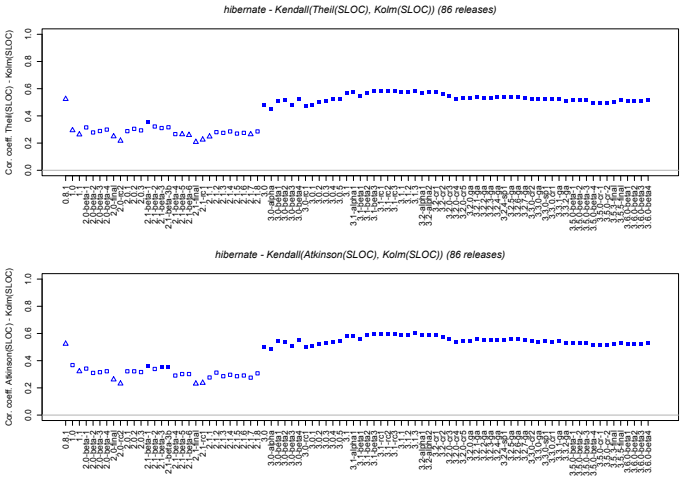

Fig. 12. Evolution of Kendall correlation between I_{Kolm} and I_{Theil}, I_{Atkinson}, respectively.

However, the evolution is different between I_{Theil} and I_{Kolm}, as well as between I_{Atkinson} and I_{Kolm}. For example, for *Hibernate* (Figure 12) we observe that the correlation became statistically significant and, although still average (0.5), the correlation coefficient significantly increased (from 0.2–0.3) starting with v3.0 in both cases. Closer inspection revealed

that *Hibernate* underwent a significant increase in size when moving from v2.1.8 (29 packages) to v3.0 (109 packages).

An interesting case is the correlation between *mean* and I_{Kolm}, which we previously observed to be high (0.8) and statistically significant in 90% of the Corpus. On the other hand, in the Evolution Corpus, there are significant variations in the correlation coefficient between different versions of the system. We illustrate this in Figure 13 on *Hibernate* and *Weka*. While correlation between *mean* and I_{Kolm} seems to fluctuate without clear relation to system size, "jumps" of the correlation values in both graphs of Figure 13 show major releases such as 3.5.0 in *Hibernate* and 3.6.0 in *Weka*.

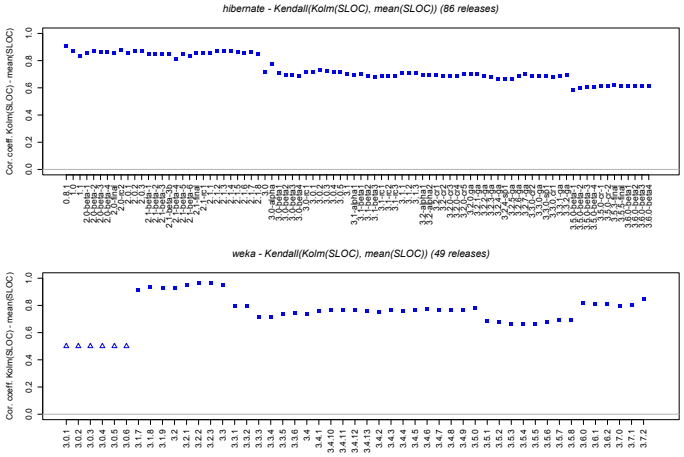

Fig. 13. Evolution of Kendall correlation between *mean* and I_{Kolm} in *Hibernate* and *Weka*.

To summarize our answer to Question 3, we note that:

- Consistently high (> 0.8) and statistically significant correlation can be observed between I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} across the Evolution Corpus.
- Correlation between I_{Theil} and I_{Kolm}, as well as between I_{Atkinson} and I_{Kolm} increases as the system size increases, while correlation between *mean* and I_{Kolm} fluctuates without clear relation to system size.

VII. Conclusions

A popular approach to assessing software maintainability and predicting its evolution involves collecting and analyzing software metrics. As metrics are usually defined on a micro-level, and should provide insights in the evolution at the macro-level, the metrics values should be aggregated. Two main groups of aggregation techniques can be found in the literature on software metrics: traditional aggregation techniques such as the *mean*, *median*, or *sum*, and more recent econometric aggregation techniques, such as the *Gini*, *Theil*, *Kolm*, *Atkinson*, and *Hoover* inequality indices. Profound comparison of different aggregation techniques was, however, missing so far.

In this paper we present the results of an extensive comparative study of both traditional and econometric aggregation techniques, applied to lifting SLOC values from class to package level in the 106 systems comprising the Qualitas Corpus. Question 1 concerned agreement between different aggregation techniques. We have observed that I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} show consistently high and statistically significant correlation between them, and similarly, the correlation between *mean* and I_{Kolm} is statistically significant, and the highest among all other techniques considered.

Furthermore, we investigate the nature of the relation between various aggregation techniques (Question 2). We note that linear (e.g., between I_{Theil} and I_{Atkinson}), superlinear (e.g., between I_{Theil} and I_{Gini}), as well as chaotic (e.g., between I_{Theil} and I_{Kolm}) patterns can be observed in the scatter plots. This led to the observation that some indices may be more appropriate than others depending on which dimension of inequality one is interested in emphasizing, the choice of metric, or the intended application.

Finally, we study evolution of the correlation between different aggregation techniques on a subset of 12 systems from the Qualitas Corpus, comprising the Evolution Corpus (Question 3). We note that consistently high (> 0.8) and statistically significant correlation can be observed between I_{Gini}, I_{Theil}, I_{Hoover} and I_{Atkinson} across the Evolution Corpus. Moreover, correlation between I_{Theil} and I_{Kolm}, as well as between I_{Atkinson} and I_{Kolm} increases as the system size increases, while correlation between *mean* and I_{Kolm} fluctuates without clear relation to system size.

We consider a number of directions of *future work*. First, by means of empirical studies we intend to compare the inequality indices with the aggregation technique of [11]. Application of inequality metrics to some of the Squale metrics and MQ will allow us to cover both notions of "aggregation of software metrics" (cf. Section III). From the econometric perspective application of inequality indices to data involving multiple economic indicators (metrics) is known as multidimensional inequality indices [57]. Therefore, studying such multidimensional inequality indices should also be considered as future work. Finally, the comparative study presented in Section V focused on aggregating the SLOC values of different Java files. We intend to replicate the study by considering other software metrics. Specifically, we intend to investigate *limited-range metrics* such as the normalized distance from the main sequence [17], [58], *low variance metrics* such as NOC and DIT [59], [60] and *metrics with negative values* such as MI [39].

References

[1] L. Erlikh, "Leveraging legacy system dollars for e-business," *IT Professional*, vol. 2, no. 3, pp. 17–23, 2000.

[2] B. W. Boehm, *Software engineering economics*. Prentice Hall, 1981.

[3] C. Verhoef, "Quantitative IT portfolio management," *Science of Computer Programming*, vol. 45, no. 1, pp. 1–96, 2002.

[4] H. Kisker, S. Ried, and H. Shey, "The state of enterprise software and emerging trends: 2010," Forrester Research, Tech. Rep., 2010.

[5] M. K. Davidsen and J. Krogstie, "A longitudinal study of development and maintenance," *Inf. Softw. Technol.*, vol. 52, pp. 707–719, Jul. 2010.

[6] Software y Servicios Chile A.G., "Sexto diagnstico de la industria nacional de software y servicios," GECHS, Tech. Rep., 2008.

[7] Y. Qian, S. Zhang, and Z. Qi, "Mining change patterns in AspectJ software evolution," in *Int. Conf. Computer Science and Software Engineering*. IEEE, 2008, pp. 108–111.

[8] T. Mens and S. Demeyer, *Software Evolution*. Springer, 2008.

[9] T. Mens, M. Wermelinger, S. Ducasse, S. Demeyer, R. Hirschfeld, and M. Jazayeri, "Challenges in software evolution," in *8th Int. Workshop on Principles of Software Evolution*. IEEE, 2005, pp. 13–22.

[10] A. Sillitti, A. Janes, G. Succi, and T. Vernazza, "Collecting, integrating and analyzing software metrics and personal software process data," in *29th Euromicro Conference*, Sep. 2003, pp. 336–342.

[11] K. Mordal-Manet, J. Laval, S. Ducasse, N. Anquetil, F. Balmas, F. Bellingard, L. Bouhier, P. Vaillergues, and T. McCabe, "An empirical model for continuous and weighted metric aggregation," in *15th Eur. Conf. Soft. Maintenance and Reeng.* IEEE, 2011, pp. 141–150.

[12] M. Lanza and R. Marinescu, *Object-oriented metrics in practice: using software metrics to characterize, evaluate, and improve the design of object-oriented systems*. Springer, 2006.

[13] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari, "Coupling metrics for predicting maintainability in service-oriented designs," in *18th Australian Softw. Engg Conf.*, Apr. 2007, pp. 329–340.

[14] I. Turnu, G. Concas, M. Marchesi, S. Pinna, and R. Tonelli, "A modified Yule process to model the evolution of some object-oriented system properties," *Inf. Sci.*, vol. 181, pp. 883–902, Feb. 2011.

[15] R. Vasa, M. Lumpe, P. Branch, and O. M. Nierstrasz, "Comparative analysis of evolving software systems using the Gini coefficient," in *Int. Conf. on Software Maintenance*. IEEE, 2009, pp. 179–188.

[16] G. Concas, M. Marchesi, S. Pinna, and N. Serra, "Power-laws in a large object-oriented software system," *IEEE Trans. Software Eng.*, vol. 33, no. 10, pp. 687–708, 2007.

[17] A. Serebrenik, S. Roubtsov, and M. van den Brand, "D_n-based architecture assessment of Java open source software systems," in *17th Int. Conf. on Program Comprehension*. IEEE, 2009, pp. 198–207.

[18] I. Herraiz, "A statistical examination of the evolution and properties of libre software," in *ICSM*. IEEE Computer Society, 2009, pp. 439–442.

[19] F. A. Cowell and S. P. Jenkins, "How much inequality can we explain? a methodology and an application to the United States," *Economic Journal*, vol. 105, no. 429, pp. 421–30, March 1995.

[20] F. A. Cowell and K. Kuga, "Inequality measurement: An axiomatic approach," *Eur. Econ. Review*, vol. 15, no. 3, pp. 287–305, Mar. 1981.

[21] F. A. Cowell, "Measurement of inequality," ser. Handbook of Income Distribution. Elsevier, 2000, vol. 1, pp. 87 – 166.

[22] G. Baxter, M. Frean, J. Noble, M. Rickerby, H. Smith, M. Visser, H. Melton, and E. Tempero, "Understanding the shape of Java software," *ACM SIGPLAN Notices*, vol. 41, no. 10, pp. 397–412, 2006.

[23] M. Goeminne and T. Mens, "Evidence for the Pareto principle in Open Source Software Activity," in *Proc. Int'l Workshop SQM 2011*. CEUR-WS workshop proceedings, 2011.

[24] A. Zeileis, "Package 'ineq' for R," CRAN, Tech. Rep., 2009.

[25] R. Wheeldon and S. Counsell, "Power law distributions in class relationships," in *Source Code Analysis and Manipulation, 2003. Proceedings. Third IEEE International Workshop on*, Sep. 2003, pp. 45–54.

[26] B. Vasilescu, A. Serebrenik, and M. G. J. van den Brand, "By no means: A study on aggregating software metrics," in *2nd International Workshop on Emerging Trends in Software Metrics*, G. Concas, M. Di Penta, E. Tempero, and H. Zhang, Eds., Honolulu, Hawaii, USA, 2011.

[27] ——, "Comparative study of software metrics' aggregation techniques," in *9th Belgian-Netherlands Softw. Evolution Seminar*, S. Ducasse, L. Duchien, and L. Seinturier, Eds., Lille, 2010, pp. 1–5.

[28] A. Serebrenik and M. van den Brand, "Theil index for aggregation of software metrics values," in *Int. Conf. Softw. Maint.* IEEE, 2010, pp. 1–9.

[29] E. Tempero, C. Anslow, J. Dietrich, T. Han, J. Li, M. Lumpe, H. Melton, and J. Noble, "Qualitas corpus: A curated collection of java code for empirical studies," in *Asia Pacific Softw. Engg Conf.*, 2010.

[30] W. Navidi, *Statistics for engineers and scientists*. McGraw-Hill, 2008.

[31] C. Gini, "Measurement of inequality of incomes," *The Economic Journal*, vol. 31, pp. 124–126, 1921.

[32] H. Theil, *Economics and Information Theory*. North-Holland, 1967.

[33] A. B. Atkinson, "On the measurement of inequality," *Journal of Economic Theory*, vol. 2, no. 3, pp. 244–263, 1970.

[34] E. Hoover Jr, "The measurement of industrial localization," *The Review of Economic Statistics*, vol. 18, no. 4, pp. 162–171, 1936.

[35] S.-C. Kolm, "Unequal inequalities I," *Journal of Economic Theory*, vol. 12, no. 3, pp. 416–442, 1976.

[36] X. Sala-i-Martin, "The world distribution of income: Falling poverty and convergence, period," *The Quarterly Journal of Economics*, vol. 121, no. 2, pp. 351–397, 2006.

[37] T. Akita, R. A. Lukman, and Y. Yamada, "Inequality in the distribution of household expenditures in Indonesia: A Theil decomposition analysis," *Developing Economies*, vol. XXXVII, no. 2, pp. 197–221, June 1999.

[38] A. F. Shorrocks, "The class of additively decomposable inequality measures," *Econometrica*, vol. 48, no. 3, pp. 613–625, 1980. [Online]. Available: http://www.jstor.org/stable/1913126

[39] P. Oman and J. Hagemeister, "Construction and testing of polynomials predicting software maintainability," *Journal of Systems and Software*, vol. 24, no. 3, pp. 251–266, 1994.

[40] S. Mancoridis, B. Mitchell, Y. Chen, and E. Gansner, "Bunch: a clustering tool for the recovery and maintenance of software system structures," in *Int. Conf. on Softw. Maintenance*, 1999, pp. 50–59.

[41] M. Lumpe, S. Mahmud, and R. Vasa, "On the use of properties in java applications," in *21st Australian Software Engineering Conference*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 235–244.

[42] C. Papatheodorou and M. Petmesidou, "Poverty profiles and trends: How do southern European countries compare to each other?" in *Poverty and social deprivation in the Mediterranean*, ser. CROP international studies in poverty research. Zed Books, 2006, pp. 47–94.

[43] D. G. Champernowne, "A comparison of measures of inequality of income distribution," *The Econ. J*, vol. 84, no. 336, pp. 787–816, 1974.

[44] B. A. Portnov and D. Felsenstein, "Measures of regional inequality for small countries," in *Regional Disparities in Small Countries*, B. A. Portnov and D. Felsenstein, Eds. College Station, Texas: Springer Verlag, 2005, ch. 4, pp. 47–62.

[45] G. Robles, J. M. Gonzalez-Barahona, M. Michlmayr, and J. J. Amor, "Mining large software compilations over time: another perspective of software evolution," in *International Workshop on Mining Software Repositories*. ACM, 2006, pp. 3–9.

[46] K. Boulding, "What is evolutionary economics?" *Journal of Evolutionary Economics*, vol. 1, no. 1, pp. 9–17, 1991.

[47] J. Foster and W. Hölzl, *Applied evolutionary economics and complex systems*. Edward Elgar Pub, 2004.

[48] F. Balmas, F. Bellingard, S. Denier, S. Ducasse, B. Franchet, J. Laval, K. Mordal-Manet, and P. Vaillergues, *The Squale Quality Model. Modèle enrichi d'agrégation des pratiques pour Java et C++*, INRIA, 2010. [Online]. Available: http://www.squale.org/quality-models-site/research-deliverables/WP1.3_Practices-in-the-Squale-Quality-Model_v2.pdf

[49] E. Langford, N. Schwertman, and M. Owens, "Is the property of being positively correlated transitive?" *The American Statistician*, vol. 55, no. 4, pp. 322–325, 2001.

[50] E. Tempero, "Qualitas Corpus 20101126 release notes," http://qualitascorpus.com/docs/history/20101126.html, 2010.

[51] K. Pearson, "Note on Regression and Inheritance in the Case of Two Parents," *Royal Society Proceedings*, vol. 58, pp. 240–242, 1895.

[52] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[53] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, pp. 441–471, 1987.

[54] G. E. Noether, "Why Kendall tau?" *Teaching Statistics*, vol. 3, no. 2, pp. 41–43, 1981.

[55] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2010. [Online]. Available: http://www.R-project.org

[56] P. N. Conceicao and P. M. Ferreira, "The Young Person's Guide to the Theil Index: Suggesting Intuitive Interpretations and Exploring Analytical Applications," *SSRN eLibrary*, 2000.

[57] K.-Y. Tsui, "Multidimensional Generalizations of the Relative and Absolute Inequality Indices: The Atkinson-Kolm-Sen Approach," *Journal of Economic Theory*, vol. 67, no. 1, pp. 251–265, 1995.

[58] R. Martin, "OO design quality metrics: An analysis of dependencies," 1994. [Online]. Available: http://condor.depaul.edu/~dmumaugh/OOT/Design-Principles/oodmetrc.pdf

[59] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," *IEEE Trans. Softw. Eng.*, vol. 20, no. 6, pp. 476–493, 1994.

[60] G. Succi, W. Pedrycz, S. Djokic, P. Zuliani, and B. Russo, "An empirical exploration of the distributions of the Chidamber and Kemerer object-oriented metrics suite," *Empirical Softw. Eng.*, vol. 10, pp. 81–104, Jan. 2005.