The Journal of Open Source Software

Metrics As Scores: A Tool- and Analysis Suite and Interactive Application for Exploring Context-Dependent Distributions

Sebastian Hönel [1¶], **Morgan Ericsson** [1], **Welf Löwe** [1], **and Anna Wingkvist** [1]

1 Department of Computer Science and Media Technology, Linnaeus University, Sweden ¶ Corresponding author

DOI: 10.21105/joss.04913

Software

- Review
- Repository
- Archive

Editor: Mikkel Meyer Andersen

Reviewers:

- @mdhaber
- @kostiantyn-kucher

Submitted: 03 October 2022
Published: 25 August 2023

Summary

Metrics As Scores can be thought of as an interactive, *multiple* analysis of variance (abbr. "ANOVA," Chambers et al., 2017). An ANOVA might be used to estimate the *goodness-of-fit* of a statistical model. Beyond ANOVA, which is used to analyze the differences among hypothesized group means for a single quantity (feature), Metrics As Scores seeks to answer the question of whether a sample of a certain feature is more or less common across groups. This approach to data visualization and -exploration has been used previously (e.g., Jiang et al., 2022). Beyond this, Metrics As Scores can determine what might constitute a good/bad, acceptable/alarming, or common/extreme value, and how distant the sample is from that value, for each group. This is expressed in terms of a percentile (a standardized scale of $[0, 1]$), which we call ***score***. Considering all available features among the existing groups furthermore allows the user to assess how different the groups are from each other, or whether they are indistinguishable from one another.

The name *Metrics As Scores* was derived from its initial application: examining differences of software metrics across application domains (Hönel et al., 2022). A software metric is an aggregation of one or more raw features according to some well-defined standard, method, or calculation. In software processes, such aggregations are often counts of events or certain properties (Florac & Carleton, 1999). However, without the aggregation that is done in a quality model, raw data (samples) and software metrics are rarely of great value to analysts and decision-makers. This is because quality models are conceived to establish a connection between software metrics and certain quality goals (Kaner & Bond, 2004). It is, therefore, difficult to answer the question "is my metric value good?".

With Metrics As Scores we present an approach that, given some *ideal* value, can transform any sample into a score, given a sample of sufficiently many relevant values. While such ideal values for software metrics were previously attempted to be derived from, e.g., experience or surveys (Benlarbi et al., 2000), benchmarks (Alves et al., 2010), or by setting practical values (Grady, 1992), with Metrics As Scores we suggest deriving ideal values additionally in non-parametric, statistical ways. To do so, data first needs to be captured in a *relevant* context (group). A feature value might be good in one context, while it is less so in another. Therefore, we suggest generalizing and contextualizing the approach taken by Ulan et al. (2021), in which a score is defined to always have a range of $[0, 1]$ and linear behavior. This means that scores can now also be compared and that a fixed increment in any score is equally valuable among scores. This is not the case for raw features, otherwise.

Metrics As Scores consists of a tool- and analysis suite and an interactive application that allows researchers to explore and understand differences in scores across groups. The operationalization

of features as scores lies in gathering values that are context-specific (group-typical), determining an ideal value non-parametrically or by user preference, and then transforming the observed values into distances. Metrics As Scores enables this procedure by unifying the way of obtaining probability densities/masses and conducting appropriate statistical tests. More than 120 different parametric distributions (approx. 20 of which are discrete) are fitted through a common interface. Those distributions are part of the `scipy` package for the Python programming language, which Metrics As Scores makes extensive use of (Virtanen et al., 2020). While fitting continuous distributions is straightforward using maximum likelihood estimation, many discrete distributions have integral parameters. For these, Metrics As Scores solves a mixed-variable global optimization problem using a genetic algorithm in `pymoo` (Blank & Deb, 2020). Additionally to that, empirical distributions (continuous and discrete) and smooth approximate kernel density estimates are available. Applicable statistical tests for assessing the goodness-of-fit are automatically performed. These tests are used to select some best-fitting random variable in the interactive web application. As an application written in Python, Metrics As Scores is made available as a package that is installable using the Python Package Index (PyPI): `pip install metrics-as-scores`. As such, the application can be used in a stand-alone manner and does not require additional packages, such as a web server or third-party libraries.

Statement Of Need

Metrics As Scores is a supplement to existing analyses that enables the exploration of differences a mong groups in a novel, mostly interactive way. Raw features are seldomly useful as, e.g., indicators of quality. Only the transformation to scores enables an apples-to-apples comparison of different quantities (features) across contexts (groups). This is particularly true for software metrics, which often cannot be compared directly, because due to their different scales and distributions, there does not exist a mathematically sound way to do so (Ulan et al., 2018). While some have attempted to associate blank software metrics with quality (e.g., Basili et al., 1996), most often applications have to resort to using software metrics as, e.g., fault indicators (Aziz et al., 2019; Caulo, 2019), or as indicators of reliability and complexity (Chidamber & Kemerer, 1994). Furthermore, none of the existing approaches that attempted to associate software metrics with quality paid great attention to the fact that software metrics have different distributions and, therefore, different statistical properties across application domains. Therefore, the operationalization of software metrics as scores ought to be conditional on the application domain.

MAS – The Tool- and Analysis Suite

The main purpose of the Metrics As Scores tool- and analysis suite for Python is to approximate or estimate, enable the exploration of, and sample from context-dependent distributions. Three principal types of distributions are supported: empirical and parametric (both continuous and discrete), as well as kernel density estimates. These are all unified using the class `Density`, which provides access to the probability density/mass function (PDF/PMF), the cumulative distribution function (CDF) and its complement (CCDF) for scores, and the percent point function (PPF). As a unified representation for all these we choose line plots, as these are most commonly used for continuous densities. Instead of, e.g., histograms for discrete data, the plotting will fall back to using step-wise linear functions. Metrics As Scores carries out a number of statistical tests for fitted distributions. The results for each test are stored in a separate spreadsheet after the fitting process and may be used to further investigate how well certain distributions fit and what the alternatives are. The carried out tests are: Cramér–von Mises (Cramér, 1928) and Kolmogorov–Smirnov one-sample (Stephens, 1974) tests, Cramér–von Mises (Anderson, 1962), Kolmogorov–Smirnov, and Epps–Singleton (Epps & Singleton, 1986) two-sample tests. The second sample required for the two-sample test is

obtained by uniformly sampling from the fitted distribution's PPF. The best-fitting distribution is selected for pre-generating densities that are used by the web application, such that only the single best fit is used for visualization. The Epps–Singleton two-sample test is compatible with discrete data and is used for discrete distributions. For continuous data, the one-sample Kolmogorov–Smirnov test is used.

Metrics As Scores supports the transformation of samples into distances using ideal values that are computed non-parametrically. Given a sample X from an arbitrary population and an ideal value i_X, the corresponding distance, D, is obtained as $D = |X - i_X|$. In order to obtain a discrete ideal value (e.g., when transforming a discrete sample in order to fit a discrete probability distribution), the expectation (mean), median, infimum, and supremum can be obtained in a straightforward way and then rounded. A discrete value for the mode (most common value) is determined using `scipy`. When a continuous ideal value is required, we first estimate a kernel density $f_{\mathcal{X}}$ using a Gaussian kernel. Then, the expectation is obtained as $\mathbb{E}[\mathcal{X}] = \int_{-\infty}^{\infty} t\, f_{\mathcal{X}}(t)\, dt$. The mode of a sample X is obtained by solving $\hat{x} = \operatorname{argmax}_{x \in \mathcal{X}} f_{\mathcal{X}}(x)$. In order to approximate the median, we obtain a large sample from the kernel density and compute its median using `numpy` (Harris et al., 2020).

In order to understand whether or not the available groups in the data matter before obtaining any of these distributions, Metrics As Scores supports additional tools for generating and outputting results for three other statistical tests. The ANOVA test is used to analyze differences among sample means (which, e.g., stem from the same feature in different groups). Tukey's Honest Significance Test (abbr. "TukeyHSD," Tukey, 1949) is used to gain insights into the results of an ANOVA test. While the former only allows obtaining the amount of corroboration for the null hypothesis, TukeyHSD performs all pairwise comparisons (for all possible combinations of any two groups). Lastly, Welch's two-sample t-test (which does not assume equal population variances) compares the means of two samples to give an indication of whether or not they appear to come from the same distribution (Welch, 1947).

Metrics As Scores includes a scientific template for generating a report for a dataset that exploits the results of these analyses (for example, see Hönel, 2023b). Users are encouraged to import their own datasets and have Metrics As Scores conduct all necessary analyses, generate a report, and bundle a publishable dataset. The application comes with a rich text-based user interface, which offers wizards that afford completely code-free interactions. These interactions include, for example, showing installed datasets, downloading of known datasets from a curated list, creating own datasets, automatically attempting to fit more than 120 random variables, report creation and bundling of own datasets, pre-generating densities for the interactive web application, and running the web application with a locally available dataset.

MAS – The Interactive Application

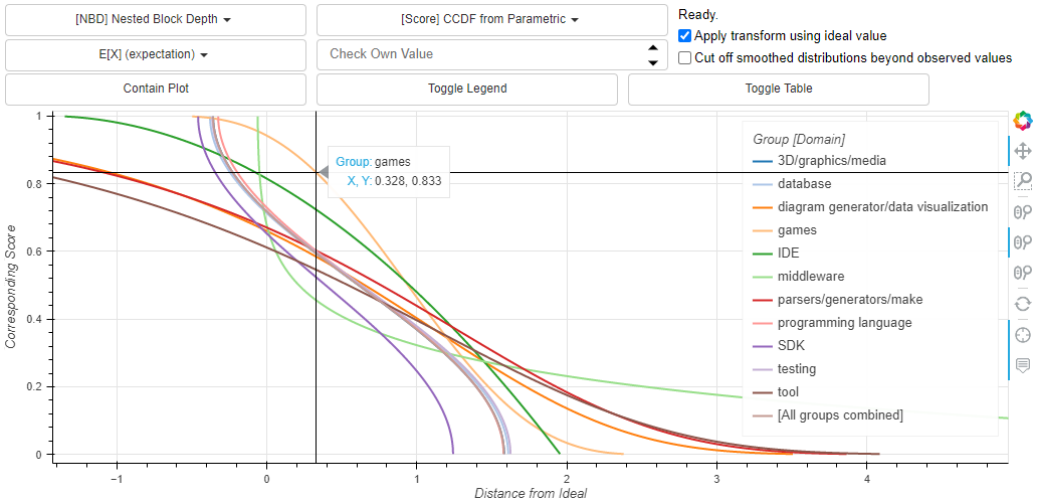

Figure 1: Main plot area of the application "Metrics As Scores". Using the Qualitas.class corpus, software metrics values of own applications can be scored against the corpus' groups (application domains). Shown are the CCDFs (scores) of the fitted parametric distributions for the metric NBD transformed using the domain-specific expectation as ideal value.

The interactive application is partially shown in Figure 1. Not shown are the header, UI controls, a tabular with numerical data for the current selection, and the footer which contains help. The application supports all transforms, continuous and discrete distributions, obtaining scores for own features/sampling from inverse CDFs (PPFs), and grouping of features into discrete/continuous. The main tool, the plot, allows the user to zoom, pan, select, enable/disable contexts, and manually hover the graphs to obtain precise x/y-values. The web application can be launched with any of the available datasets (manually created or downloaded). The interactive application is built using Bokeh and facilitates customization using a few steps described in the software's manual (Bokeh Development Team, 2018).

Applications

The Metrics As Scores tool- and analysis suite and interactive application have initially been used to study the "Qualitas.class corpus" of software metrics (Terra et al., 2013). The results of studying the software metrics of the corpus show that, for example, the context (application domain) software metrics were captured in is always of importance and must not be neglected. In addition, some of the software metrics in the corpus are *never* similar across application domains and must be applied with great care when used in quality models (Hönel et al., 2022). Evidently, the approach offered by Metrics As Scores enables not only to examine and compare samples but also the contexts these are embedded in as a whole.

Metrics As Scores has since been extended to work with almost arbitrary datasets. Three well-known datasets have been added: the Iris flower dataset (Hönel, 2023d), the Diamonds dataset (Hönel, 2023c), and the Elisa Spectrophotometer Positive Samples dataset (Hönel, 2023a). While these datasets are well understood, Metrics As Scores can reveal additional insights. For example, visual inspection of the Iris flower dataset shows that the probability densities for the features of flower petals do only overlap somewhat or not at all across the three species *setosa*, *versicolor*, and *virginica*. This is corroborated by the generated report which confirms that these features are not statistically significantly similar across species.

Related Work

Metrics As Scores finds itself among other visualization tools related to statistical analysis and learning. Some existing tools support a visual and interactive approach to exploring the results of an ANOVA. In Sturm-Beiss (2005), the goal is to enable a what-if analysis by allowing the user to assume arbitrary groups in the data. Fox et al. (2009) provide a package for R to visually test hypotheses of linear models (as is ANOVA). A number of packages for creating non-interactive ANOVA visualizations exists (e.g., Pruzek & Helmreich, 2023). To the best of our knowledge, however, Metrics As Scores is the first application to enable the interactive exploration of differences among groups. It appears that it is also the first tool to enable the transformation of samples into scores and to produce and aggregate group-related results derived from these. CorpusViz by Slater et al. (2019) is a tool that exclusively targets the Qualitas corpus (Tempero et al., 2010), *not* the Qualitas.class corpus. CorpusViz attempts to satisfy the three primary requirements of composite viewing of multiple visualizations, the ability to change between software systems and versions, as well as allowing the user to configure the visualizations.

Acknowledgments

The authors would like to sincerely express their gratitude towards the reviewers of the Journal of Open Source Software for their invaluable comments.

This work is supported by the Linnaeus University Centre for Data Intensive Sciences and Applications (DISA) High-Performance Computing Center.

References

Alves, T. L., Ypma, C., & Visser, J. (2010). Deriving Metric Thresholds from Benchmark Data. In R. Marinescu, M. Lanza, & A. Marcus (Eds.), *26th IEEE International Conference on Software Maintenance, ICSM 2010, Timisoara, Romania, September 12–18, 2010* (pp. 1–10). IEEE Computer Society. https://doi.org/10.1109/ICSM.2010.5609747

Anderson, T. W. (1962). On the Distribution of the Two-Sample Cramer-von Mises Criterion. *The Annals of Mathematical Statistics*, *33*(3), 1148–1159. https://doi.org/10.1214/aoms/1177704477

Aziz, S. R., Khan, T. A., & Nadeem, A. (2019). Experimental Validation of Inheritance Metrics' Impact on Software Fault Prediction. *IEEE Access*, *7*, 85262–85275. https://doi.org/10.1109/ACCESS.2019.2924040

Basili, V. R., Briand, L. C., & Melo, W. L. (1996). A Validation of Object-Oriented Design Metrics as Quality Indicators. *IEEE Transactions on Software Engineering*, *22*(10), 751–761. https://doi.org/10.1109/32.544352

Benlarbi, S., Emam, K. E., Goel, N., & Rai, S. N. (2000). Thresholds for Object-Oriented Measures. *11th International Symposium on Software Reliability Engineering, ISSRE 2000, San Jose, CA, USA, October 8–11, 2000*, 24–39. https://doi.org/10.1109/ISSRE.2000.885858

Blank, J., & Deb, K. (2020). pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Bokeh Development Team. (2018). *Bokeh: Python library for interactive visualization*. https://bokeh.pydata.org/en/latest/

Caulo, M. (2019). A Taxonomy of Metrics for Software Fault Prediction. In M. Dumas, D. Pfahl, S. Apel, & A. Russo (Eds.), *27th ACM Joint European Software Engineering*

Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26–30, 2019 (pp. 1144–1147). ACM. https://doi.org/10.1145/3338906.3341462

Chambers, J. M., Freeny, A. E., & Heiberger, R. M. (2017). Analysis of Variance; Designed Experiments. In J. M. Chambers & T. J. Hastie (Eds.), *Statistical Models in S* (1st ed.). Routledge. https://doi.org/10.1201/9780203738535

Chidamber, S. R., & Kemerer, C. F. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*, *20*(6), 476–493. https://doi.org/10.1109/32.295895

Cramér, H. (1928). On the composition of elementary errors. *Scandinavian Actuarial Journal*, *1928*(1), 13–74. https://doi.org/10.1080/03461238.1928.10416862

Epps, T. W., & Singleton, K. J. (1986). An Omnibus Test for the Two-Sample Problem Using the Empirical Characteristic Function. *Journal of Statistical Computation and Simulation*, *26*(3-4), 177–203. https://doi.org/10.1080/00949658608810963

Florac, W. A., & Carleton, A. D. (1999). *Measuring the Software Process: Statistical Process Control for Software Process Improvement* (1st ed.). Addison-Wesley Professional. ISBN: 9780201604443

Fox, J., Friendly, M., & Monette, G. (2009). Visualizing hypothesis tests in multivariate linear models: the *heplots* package for R. *Computational Statistics*, *24*(2), 233–246. https://doi.org/10.1007/s00180-008-0120-1

Grady, R. B. (1992). *Practical Software Metrics For Project Management And Process Improvement* (1st ed.). Prentice Hall, Inc. ISBN: 9780137203840

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hönel, S. (2023a). *Metrics As Scores Dataset: Elisa Spectrophotometer Positive Samples* (Version v1.2) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7633989

Hönel, S. (2023b). *Metrics As Scores Dataset: Metrics and Domains From the Qualitas.class Corpus* (Version v1.2) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7633949

Hönel, S. (2023c). *Metrics As Scores Dataset: Price, Weight, and Other Properties of Over 1,200 Ideal-Cut and Best-Clarity Diamonds* (Version v1.2) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7647596

Hönel, S. (2023d). *Metrics As Scores Dataset: The Iris Flower Data Set* (Version v1.1) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7669664

Hönel, S., Ericsson, M., Löwe, W., & Wingkvist, A. (2022). Contextual Operationalization of Metrics As Scores: Is My Metric Value Good? *22nd IEEE International Conference on Software Quality, Reliability and Security, QRS 2022, Guangzhou, China, December 5–9, 2022*, 333–343. https://doi.org/10.1109/QRS57517.2022.00042

Jiang, W., Chen, H., Yang, L., & Pan, X. (2022). moreThanANOVA: A user-friendly Shiny/R application for exploring and comparing data with interactive visualization. *PLOS ONE*, *17*(7), e0271185. https://doi.org/10.1371/journal.pone.0271185

Kaner, C., & Bond, W. P. (2004). Software Engineering Metrics: What Do They Measure and How Do We Know? *10th IEEE International Software Metrics Symposium, METRICS 2004, Chicago, IL, USA, September 11–17, 2004*, 1–12. https://web.archive.org/web/20221002003552/https://kaner.com/pdfs/metrics2004.pdf

Pruzek, R. M., & Helmreich, J. E. (2023). *granova: Graphical Analysis of Variance*. https://CRAN.R-project.org/package=granova

Slater, J., Anslow, C., Dietrich, J., & Merino, L. (2019). CorpusVis - visualizing software metrics at scale. *7th Working Conference on Software Visualization, VISSOFT 2019, Cleveland, OH, USA, September 30 - October 1, 2019*, 99–109. https://doi.org/10.1109/VISSOFT.2019.00020

Stephens, M. A. (1974). EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association*, *69*(347), 730–737. https://doi.org/10.1080/01621459.1974.10480196

Sturm-Beiss, R. (2005). A Visualization Tool for One- and Two-Way Analysis of Variance. *Journal of Statistics Education*, *13*(1), 1–7. https://doi.org/10.1080/10691898.2005.11910641

Tempero, E. D., Anslow, C., Dietrich, J., Han, T., Li, J., Lumpe, M., Melton, H., & Noble, J. (2010). The Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies. In J. Han & T. D. Thu (Eds.), *17th Asia Pacific Software Engineering Conference, APSEC 2010, Sydney, Australia, November 30 - December 3, 2010* (pp. 336–345). IEEE Computer Society. https://doi.org/10.1109/APSEC.2010.46

Terra, R., Miranda, L. F., Valente, M. T., & Silva Bigonha, R. da. (2013). Qualitas.class corpus: a compiled version of the qualitas corpus. *ACM SIGSOFT Software Engineering Notes*, *38*(5), 1–4. https://doi.org/10.1145/2507288.2507314

Tukey, J. W. (1949). Comparing Individual Means in the Analysis of Variance. *Biometrics*, *5*(2), 99–114. https://doi.org/10.2307/3001913

Ulan, M., Löwe, W., Ericsson, M., & Wingkvist, A. (2018). Poster: Introducing Quality Models Based On Joint Probabilities. In M. Chaudron, I. Crnkovic, M. Chechik, & M. Harman (Eds.), *40th International Conference on Software Engineering: Companion Proceedings, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018* (pp. 216–217). ACM. https://doi.org/10.1145/3183440.3195103

Ulan, M., Löwe, W., Ericsson, M., & Wingkvist, A. (2021). Copula-based software metrics aggregation. *Software Quality Journal*, *29*(4), 863–899. https://doi.org/10.1007/s11219-021-09568-9

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Welch, B. L. (1947). The Generalization of "Student's" Problem When Several Different Population Variances Are Involved. *Biometrika*, *34*(1-2), 28–35. https://doi.org/10.1093/biomet/34.1-2.28