

Corrigendum: Empirical analysis of the relationship between CC and SLOC in a large corpus of Java methods and C functions published on 9 December 2015

Citation for published version (APA):

Landman, D., Serebrenik, A., Bouwers, E., & Vinju, J. J. (2017). Corrigendum: Empirical analysis of the relationship between CC and SLOC in a large corpus of Java methods and C functions published on 9 December 2015. *Journal of Software : Evolution and Process*, 29(10), [e1914]. DOI: 10.1002/smr.1914

DOI:

[10.1002/smr.1914](https://doi.org/10.1002/smr.1914)

Document status and date:

Published: 01/01/2017

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Corrigendum: Empirical analysis of the relationship between CC and SLOC in a large corpus of Java methods and C functions published on 9 December 2015

Davy Landman¹, Alexander Serebrenik², Eric Bouwers³ and Jurgen J. Vinju^{1,2,4*}

¹*Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*

²*Eindhoven University of Technology, Eindhoven, The Netherlands*

³*Software Improvement Group, Amsterdam, The Netherlands*

⁴*INRIA Lille Nord Europe, Lille, France*

Received ...

KEY WORDS: erratum

1. INTRODUCTION

During the preparation of the corresponding chapter in Davy Landman's PhD thesis, some minor graphical and statistical discrepancies were found in the paper "Empirical analysis of the relationship between CC and SLOC in a large corpus of Java methods and C functions" [1].

To support future reproduction and use of this work, we prepared the current erratum, containing several updated figures, a diagnosis of the cause of the errors, and an explanation of the effect on the original paper.

None of the issues reported in this erratum influence the conclusions of the original paper.

2. ISSUES DISCOVERED

- The hexagonal scatter plots in figure 8 lack a more prominent line at CC = 0. This was caused by a bug[†] in ggplot, that would filter out data around the limits.
- The R^2 values in the tables IV(b) and V(b) of the C corpus were off by a maximum of 0.01 from the actual result. The cause was that this table was not re-calculated after fixing a bug in the "remove out-of-scope code" phase. Note that the impact of this error is scattered throughout the paper, as the correlations of Table IV and V are often repeated for clarity in the remaining sections (for example, the R^2 of the linear model for all the C functions is 0.43 instead of 0.44).
- Our R code calculating the log-transformed linear fit contained an error. The dashed lines in Figures 8, 9, and 12 are impacted and the shape of the residual plot in 11. The biggest impact is in Figure 12, where the original fit seemed to miss the data almost entirely. We mis-interpreted this phenomenon in the last sentence of the second paragraph of Section 4.4.2; it is not caused by the skewness of the distributions of the two metrics, but rather by the current bug.

*Correspondence to: Centrum Wiskunde & Informatica, Amsterdam, The Netherlands. E-mail: jurgen.vinju@cwi.nl

[†]reported and confirmed: <https://github.com/tidyverse/ggplot2/issues/2061>

- The custom implementation of the log-scaled y-axis of the residual plots in Figure 11 contained two errors:
 - The labels on the y axis were off by a factor 10
 - For the negative side of the residual plot, we took the absolute, calculated the log10 value, and made it negative again. However, values between 0 and 1, (the values close to the linear fit) turn into a negative value (as $\log_{10}(1)$ equals 0). This caused strange outliers in the original plots that were not scrutinized. The fixed residual plots do not have this outliers and look much more like the data in Figure 9.
- We republished the data sets related to the current paper on Zenodo to increase their availability:
 - Landman, Davy. (2015). A Curated Corpus of Java Source Code based on Sourcerer (2015) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.208213>
 - Landman, Davy. (2015). A Large Corpus of C Source Code based on Gentoo packages [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.208215>
 - Davy Landman. (2015, February 26). cwi-swat/jsep-sloc-versus-cc. Zenodo. <http://doi.org/10.5281/zenodo.293795>

3. NEW TABLES AND FIGURES

The remaining part of this erratum contains updated tables and figure as replacements for the original paper.

Table IV. Correlations for part of the tail of the independent variable SLOC. All correlations have a high significance level ($p \leq 1 \times 10^{-16}$).

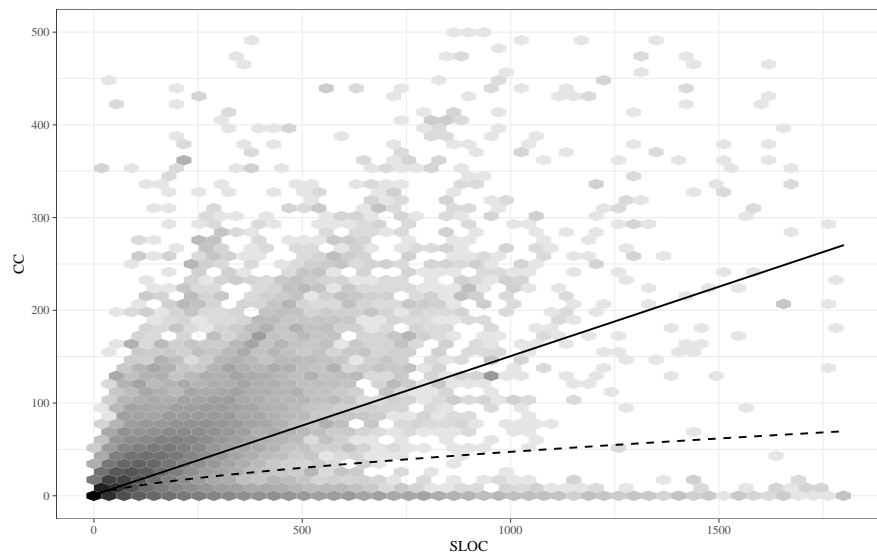
(b) C functions

Min. SLOC	Coverage	R^2	$\log R^2$	ρ	Functions
1	100%	0.43	0.70	0.83	5 810 834
12	50%	0.41	0.52	0.70	2 905 417
16	40%	0.40	0.47	0.68	2 324 334
27	25%	0.38	0.37	0.63	1 452 709
33	20%	0.38	0.33	0.61	1 162 167
56	10%	0.35	0.22	0.55	581 084
220	1%	0.28	0.05	0.38	58 109
714	0.100%	0.20	0.01	0.28	5811
2695	0.010%	0.13	0.00	0.04	582

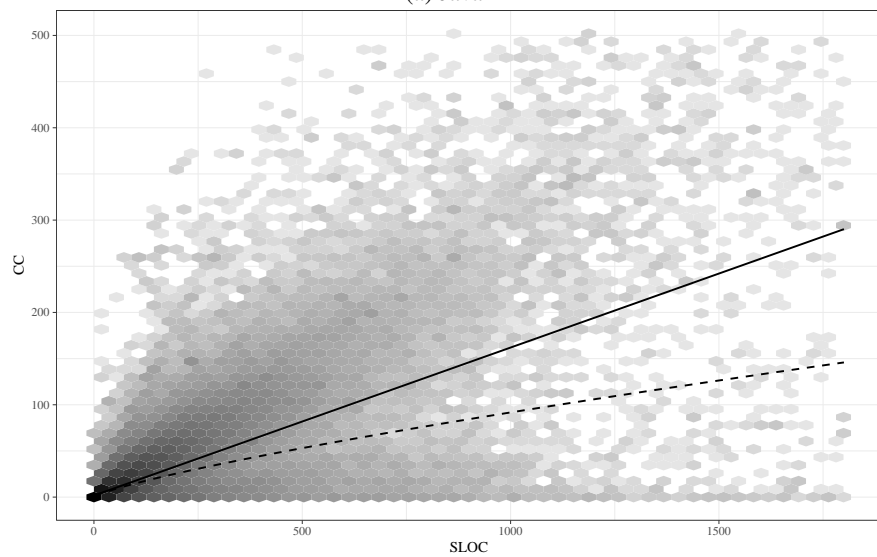
Table V. Correlations for part tail of the independent variable SLOC *removed*. All correlations have a high significance level ($p \leq 1 \times 10^{-16}$).

(b) C functions

Max. SLOC	Coverage	R^2	$\log R^2$	ρ	Functions
44 881	100%	0.43	0.70	0.83	5 810 834
3825	99.995%	0.62	0.70	0.83	5 810 543
2693	99.990%	0.62	0.70	0.83	5 810 252
714	99.900%	0.66	0.70	0.83	5 805 023
220	99%	0.66	0.69	0.83	5 752 725
56	90%	0.56	0.61	0.79	5 229 750
33	80%	0.47	0.53	0.75	4 648 667
27	75%	0.43	0.49	0.73	4 358 125
16	60%	0.33	0.37	0.65	3 486 500
12	50%	0.26	0.28	0.58	2 905 417



(a) Java



(b) C

Figure 8. Scatter plots of SLOC vs CC zoomed in on the bottom left quadrant. The solid and dashed lines are the linear regression before and after the log transform. The grayscale gradient of the hexagons is logarithmic.

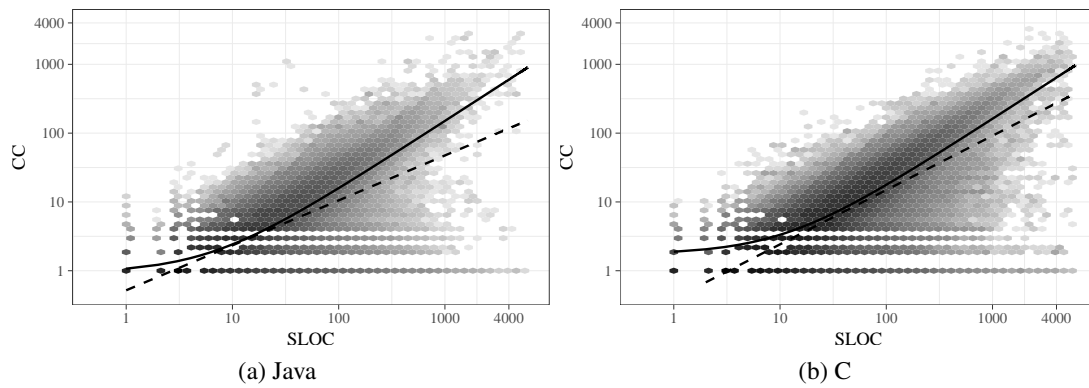


Figure 9. Scatter plots of SLOC vs CC on a log-log scale. The solid and dashed lines are the linear regression before and after the log transform. The grayscale gradient of the hexagons is logarithmic.

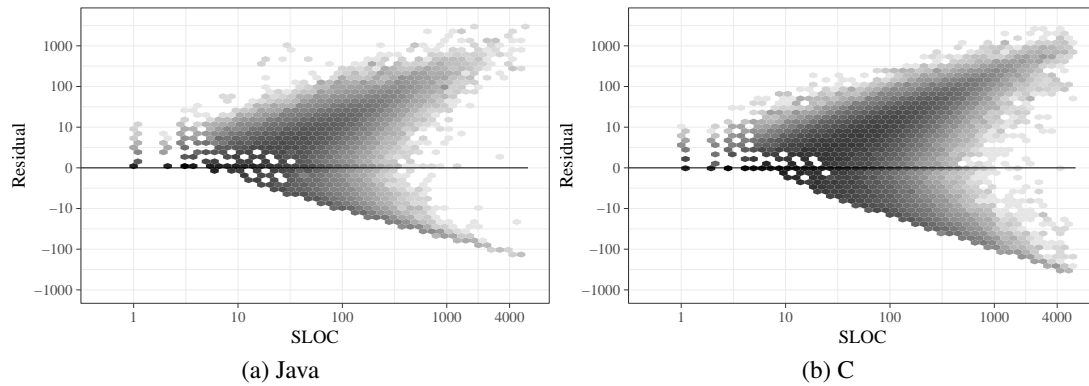


Figure 11. Residual plot of the linear regressions after the log transform, both axis are on a log scale. The grayscale gradient of the hexagons is logarithmic.

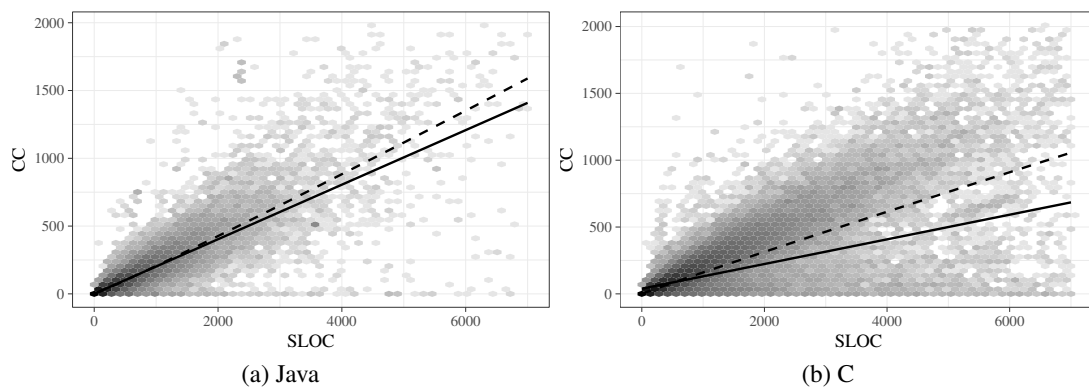


Figure 12. Scatter plots of SLOC vs CC for Java and C files. The solid and dashed lines are the linear regression before and after the log transform. The grayscale gradient of the hexagons is logarithmic.

REFERENCES

1. Landman D, Serebrenik A, Bouwers E, Vinju JJ. 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* 2016; **28**(7):589–618, doi:10.1002/smr.1760.