Title: A Computational Analysis of the Dynamics of R Style Based on 108 Million Lines of Code from All CRAN Packages in the Past 21 Years

Dear Editor,

we are delighted to learn that our paper is tentatively accepted. We would like to thank Reviewer 2 for his comments. As instructed, we provide a revised paper and our point-by-point response as follows.

- **C2-1: I think the changes made to the repository do help the accessibility, but the repository is still confusing; there is a mixture of figures, code and data files at the root level. RDS files are not handled well by git - plain text files (even if larger) would be much better and in the long run more accessible; Some further comments on the repo, especially the README:**
    - **"How to lunch docker" > "How to launch Docker" ("Docker" is a name); I don't understand what is "remote server" here.**
    - **https://github.com/chainsawriot/rstyle#line and https://github.com/chainsawriot/rstyle#comm lack the explaning sentence (that "fun" and "syntax" have); I don't see a value in shortening "communities" to "comm" - computers can handle a bit longer file names for better understandability**

  We thank the reviewer for this suggestion. We have separated all RDS files and figures with the analysis scripts. All analysis scripts are now number-coded with a non-shortened name. Only the important ones remain at the root level.

  We also agree with Reviewer 2 that in comparison with plain text files, RDS files are not handled well by git. However, it is not ideal to convert all RDS files to plain text files because many RDS files in the repository are intermediate existing for speeding up the I/O. From a preservation perspective, we should provide the **key RDS files** in a plain text format. Having said so, we still found it difficult to provide the CSV version of *pkgs_functions_with_syntax_feature.RDS* (a key RDS file) on GitHub. For one, it is a tibble with list-columns of tibbles. For two, exporting the unnested version of the tibble produces a 1.2G CSV file (vs 25 MB as an RDS). We provide now the code (*5conversion_*.R*) to convert these two files.

  We have edited the Docker section in the README as per the suggestions by Reviewer 2.

- **C2-4: not "we have summarised", but actual results: we find in the data patterns due to introduction of new language features and adoption of style guides and make a suggestion for a conensus-based style.**

  We thank the reviewer for this suggestion. We agree with Reviewer 2 that those are actual results. On second thought, we are not in a good position to make suggestions in such a “religious issue” (at least not in an active voice, i.e. “we suggest...”). We therefore modified the text as follows:

  “If such an effort would be undertaken by someone else, the following consensus-based style

could be used as the basis."

Also, in the abstract, we reported what we observe in the data in a "detached observer" manner:

"We observe in the data that a consensus in programming style is forming such as using lower snake case for function names (e.g. softplus_func) and <- rather than = for assignment."

- **C2-19: I think this is an interesting aspect of your dataset that you could briefly mention, but understand your concerns about scope.**

  We thank the reviewer for this comment. We have added in our paper as follows:

  "Based on our within-package analysis, we noticed that it is rare for a package to use a consistent style in all of its functions, except those packages with only a few functions."

- **C2-22..24: I find this solution a little bit weak. Maybe you can find a value within the communities that makes this less arbitrary, e.g., the median number of packages in a community, or the top 10%? Or something with the change in in number of packages from one community to the next in the ranks? How many packages a in the "community" after the one you labeled "rJava" - maybe just 26? So why is "the next one" not included, when the difference is the same as the one between "IO" and "rJava"? In any case, please do add the inforamtion that the included communities cover 88% of packaegs - that makes your decision stronger.**

  We thank the reviewer for this comment. We have added the coverage information in the revised paper:

  "These 20 identified communities cover 88% of the total 14,491 packages, which shows that the coverage of our analysis is comprehensive. "

  From our knowledge, we do not think there is a "correct" way to select the number of detected communities to report. Any choice would be perceived as arbitrary. The choice should remove small communities but keep most of the nodes (i.e. packages). But once again, the definition of "small communities" and the constitution of "most of the nodes" are in the eye of the beholder. Reviewer 2 would agree with us that our choice would retain most of the packages, as indicated by the 88% coverage. Readers who don't agree with this choice — as we have written in the text — can explore other choices with our provided data.

  The bottom line, however, is that regardless of the choice (except 1), the analysis comes to the same conclusion: community-based variations exist. Therefore, the choice won't change our conclusion.

- **C2-25: Excellent, though I did not find it - which repository do you mean? I strongly suggest not hiding the efforts you put in and mention this (including the rough hardware**

specification, I have no idea what you deem "modest") in the paper.

We thank the reviewer for this comment. Actually, we added that in the README of the **GitHub** repository that running *2syntax01_extract_features.R* (previously "syntax01_extract_features.R") "takes a long time." We further updated the README to make the running time more explicit. Regarding the hardware specification, we put that information in the README. Footnote 3 has been added to refer to that.

- **C2-31: I must have misunderstood - of course Bioconductor packages are not analysed. However, you do use Bioconducter's coding style. So, when it comes to the "whole community of R users", it remains unclear whether Bioconductor style is more or less common than the other styles, because all the packages from Bioconductor have not been included (and presumably have to follow Bioconductor's style guidelines?). If you understand what I mean, maybe you can clarify.**

  We thank the reviewer for this comment. To the best of our understanding, his concern is that we do not have a say on whether the Bioconductor style guide is more or less common than other style guides because we did not include Bioconductor packages in our analysis. This would be a fair point if the purpose of this study were to analyze which style guide prevails. However, this is not our focus. We rather focus on the analysis of each individual **style element** instead of any individual style guide *en bloc*. Therefore, "whether Bioconductor style is more or less common than the other styles" (or the compliance to any style guide) is not a research question that this study was designed to answer.

  Regarding the relationship between excluding Bioconductor and the "whole community of R users": we reread our manuscript and our previous reply thoroughly, and we didn't (and also don't) write any sweeping statement that our analysis covers "the whole community of R users"; not even "R users". The only instance of "R community" in the text reads "the diversity of the R community", which we believe is a fair statement.

- **C2-32: Thanks for the figure updates. On second thought, I am not sure a line plot is a good idea here. Why not use a barplot (geom_bar - worked well in my view) instead? What does it mean to interpolate the value of 40.5 number of characters to the share of lines? (That is what the line plot implies - that you can "fill up" values in between in a reasonable way.)**

  We thank the reviewer for this comment. We can understand Reviewer 2's concern, but we maintain our decision to plot the distributions as line plots. Our reasoning is threefold.

  First, plotting discrete distribution as a line plot, also known as frequency polygon, is quite well established. If the x-axis is time, then that is a typical discrete-valued time series plot. By no means a frequency polygon (or a discrete-valued time series plot) implies that the discrete data in the x-axis is continuous (as Reviewer 2 implied in his question about the share of line with 40.5 characters). The same logic would apply for Figure 1 and Figure 2. ("What is the share of exported functions using a specific style at 2001-07-21 23:02:45.12345?") As in all frequency polygons and discrete-valued time series plots, one should not interpret the connection between two data points as interpolation.

Second, it would be quite difficult to read the difference between two distributions (comment vs non-comment) in a subplot of barplots, either putting two bars side-by-side horizontally or stacking them vertically. Stacking them vertically is technically incorrect because the two distributions are independent. Another way is to create yet another facet and make the chart 2 x 2 x 2. But the chart would be too convoluted to comprehend and make direct comparison difficult.

Last, the same visualization was used by Vanderplas in his analysis of Python packages. https://jakevdp.github.io/blog/2017/11/09/exploring-line-lengths-in-python-packages/

With that being said: if the editor has a strong preference over barplots, which implies that Figure 1 and 2 also need to be corrected, we could then change all figures (1,2,3, and the Shiny app) accordingly.

- **C2-35: I suggest sorting the communities in Figures 4 and 5 in the same order as Table 2, so the reader can get an understanding of which used style is actually more common.**

  We thank the reviewer for this suggestion. This might be a good suggestion for consistency. However, these different rankings are also the messages we would like to convey. With the current ranking in Figure 4, one can see how the most popular naming convention, lower_snake, is adopted among communities. Besides, it can be clearly seen in Figure 5 that the “GUI: Gtk” community seems to have an exceptionally higher rate of adopting the unconventional “x_opencurly” style-element than any other community. Arranging the communities by size, as in Table 2, makes these observations difficult to see. Therefore, we suggest not to sort the communities in the same order as Table 2.

- **C2-37: Suggest to rephrase "contains all submissions", because you only consider each submission only once per year**

  We thank the reviewer for this suggestion. We modified the mention of “contains all submissions” to “the latest snapshot of all packages from 1998 to 2019”.

- **C2-48: Fair enough - I hope you reconsider this though. Changes to the GitHub repository can be made any time and do not impact the actual paper, but the current structure makes it hard for others to build upon the complex analysis. I would applaud the authors if they continue to develope their analysis even after the paper has been published, and make the parts more reusable - possibly so that anyone can easily re-run their analysis next year.**

  We thank the reviewer for this comment. Please refer to our response to C2-1 above. We will try to make the parts more reusable for future studies.

- **New comments**
  - **"by study the time trend" > "by studying"**
  - **Don't see a reason why footnote 3 should not be part of the main body of text: "Maintaining a consistent style in source code can enable efficient reading by multiple readers and is though to be a sign of code quality (...)."**
  - **remove the "install.packages(..)" call in yen.R**

  We have addressed these comments in the text.

To keep the reply compact, we skip the following comments:

- **C2-2: Good!**
- **C2-5: Thanks! I strongly suggest to double check if**
  We skipped this comment because it is incomplete.

- **C2-6..14: Good!**
- **C2-15: Thanks for the clarification.**
- **C2-17..18: OK!**
- **C2-26..30: Thank you for the explanations and addressing the comments.**
- **C2-38..47: Good. And thanks!**
- **C2-33: Understood!**
- **C2-34: Thanks!**
- **C2-20..21: Good!**
- **C2-49..50: Good.**

Decision: I see the need for some minor revisions and fixing language, yet only see the decision of the number of communities (see C2-22) and the figures (see C2-32, C2-35) as issues that must be addressed before publication, but do not require another round of review.