Report on the Assessment for Midterm Project

CS 510-01 Computing for Scientists

10 December, 2020

Summary

This report is for: **Niko Todorov**. Peer reviewers were: Lucas Jeay-Bizot, Ryan Kassab, Tristan Tran.

The version reviewed was: https://github.com/CS-510-Fall-2020/AALinR at commit e010756.

A bit about the report: this is a report of aggregate submissions from the Midterm Assessment Tool form. This will help you understand how the reviewers were able to understand the code in your midterm project. The report will help determine what the Peer score (out of 30 points) and Instructor score (out of 30 points) is. These are then added to the other sections of the final project score.

Reproducibility

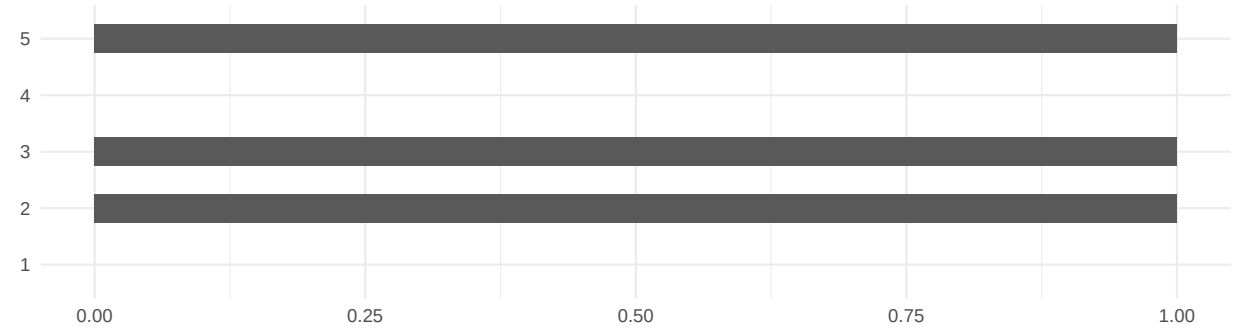
This section describes the reproducibility of your code in the reviewers' environments.

Were you able to reproduce any associate figures, tables, or other analyses that are described?

Reviewer scores: I was not able to reproduce the results, I was not able to reproduce the results, Yes, I reproduced the results with one or two changes

Rate the ease of reproducing the results

1 - *Not easy/Unable to reproduce*, 5 - *Very easy.*



The instructor rated the ease of reproduction as 4. Higher scores indicate that the code has fewer problems reproducing on different systems.

In the case that code could not be reproduced, specific comments:

I was not able to reproduce the same figure as in Rplot.png. Line 25 generates a plot with a max y-axis value of 8. Line 24 gives an error because the names of the columns do not exist (the line runs if I replace with column names that exists: e.g. LA6‘AAL‘ $< -C.10.20 * (LA61$+LA6‘2‘) $+ C.20.50 * (LA62$+LA6‘3‘) +

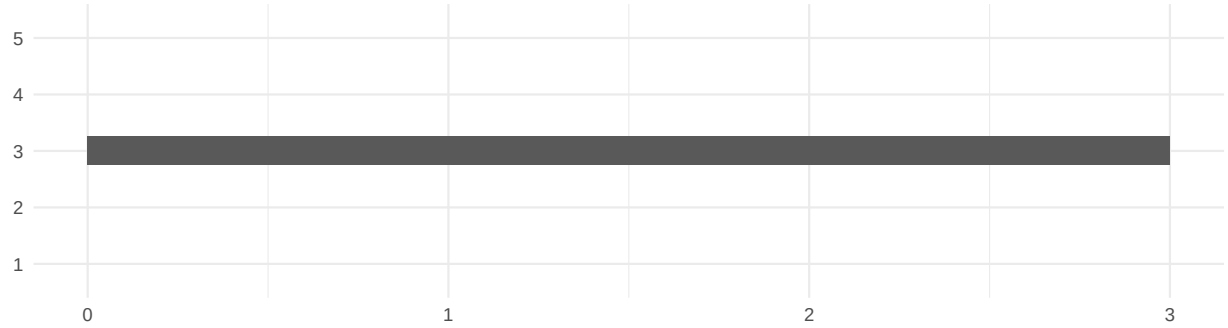
$C.50.100 * (LA63+LA6`4`) + C.100.200 * (LA64+LA6`5`) + C.200.500 * (LA65+LA6\$6))$, I am unsure about what results that was supposed to be produced as there was nothing described and no plots were produced, I had to installl the required packages. I would recommending lines like the following below,

Documentation

This section evaluates the quality of the documentation of your code and how well your peers are able to understand what the code is accomplishing.

Rate the quality of the documentation of the code

1 - *Poor*, 5 - *Excellent*.



The instructor rated the documentation as 5.

In terms of documentation and structure, 33% of your peer reviewers agreed that they could easily understand the code as written and documented.

Comments regarding documentation:

I can understand most steps. However I do not understand some lines e.g. line 46. More generally I do not understand how the code achieves an AAL using Riemann's method., There a very few comments that are embedded in the code and the comments aren't super helpful, but it seems like the code is just reading in data from tables so its not hard to follow., It was easy I would recommend adding a /doc folder with information on the methods, background and also how to understand the outputs (the figures), as well as more in line commenting (explaining each chunk of code). , I would say to add more comments within the code and make the comments more useful and explanatory on what the line(s) of code is/are doing. , It was easy to run the code and repeat the analysis; however, I was a little confused at the end. It would be nice to have an output of some type. Either a csv or a print statement letting me know the desired result has been achieved. This can also be mentioned in the Readme or documentation. Documentation is pretty good. For the final, I'd add some context to the graph of the results to help the user interpret what they are seeing.

Coding Practice

Best Practices

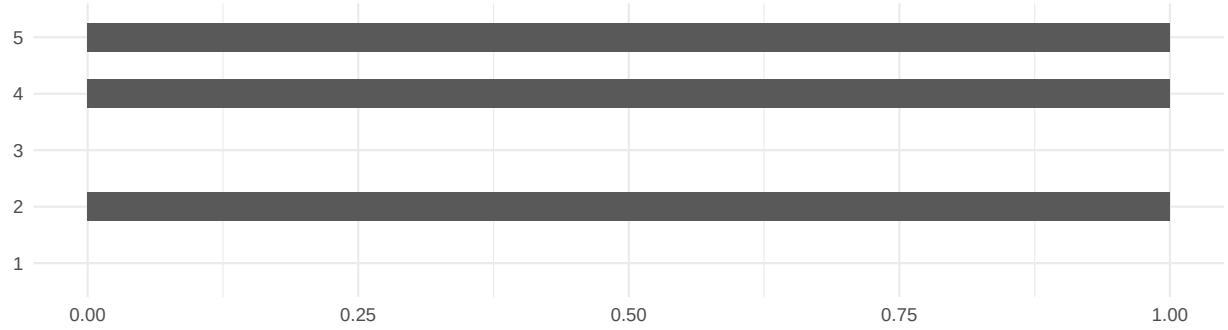
Overall, the instructor notes that your project did follow standard organization set forth in Wilson et al. 2014. The code did not follow best practices as set out in the *Advanced R* book. Consider refactoring to adhere to these standards before submitting your final project.

Testing

The instructor notes that your project did not contain working tests.

Rate the level of "code smells"

1 - *Many issues*, 5 - *Clean code following best practices.*



High scores indicate code was easy to read and free from problematic features that commonly plague code. Your average score amongst all reviewers is 4.

Improvement

The instructor noted that there was not notable improvement between the first-draft version and this version of the project. Peer responses indicated that 67% saw notable improvement. Please address comments before you submit a final project.

Here are additional, general comments from the reviewers:

For the final version of this project, I would recommend using more folders: one for documentation with more explanation on what the code is doing and one for exemples or results with exemples of results. I would also recommend the use of more commenting in the code, this would make it easier to look back to. As is stated previously suggested, using a check in the script for dependencies will make it more reproducible (e.g. some algorithm that installs missing packages). I think that the NOLA variable is not plotted either. I understand from the commenting that some sections are still under construction - my best advice when filling in the missing code is to add commenting and documentation to go along with it. This approach to study natural hazards looks very interesting, looking forward to the final result., I would specify what the code is doing in either the README or the code itself. From there I would try and specify any tests that could be run using the code., in line 24 of your code, you do a lengthy calculation. I think you can break it up into sections or modularize it a bit. I would recommend using ggsave() after you plot the figure, because it does not show when you run as a script. Throughout, replace = with <- to conform to style guide. Otherwise, it would be good if you could add a test or vignette to go along with the code to give the user some context.