

# NemoGB: An R Package for Calculating Course Grades

Nikita Jayaprakash<sup>1</sup>, Monika Voutov<sup>1, 2</sup>, Zach Turner<sup>1</sup>, and Andrew Bray<sup>1</sup>

<sup>1</sup> UC Berkeley, Department of Statistics <sup>2</sup> UC Berkeley, College of Engineering

DOI:

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted:

Published:

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

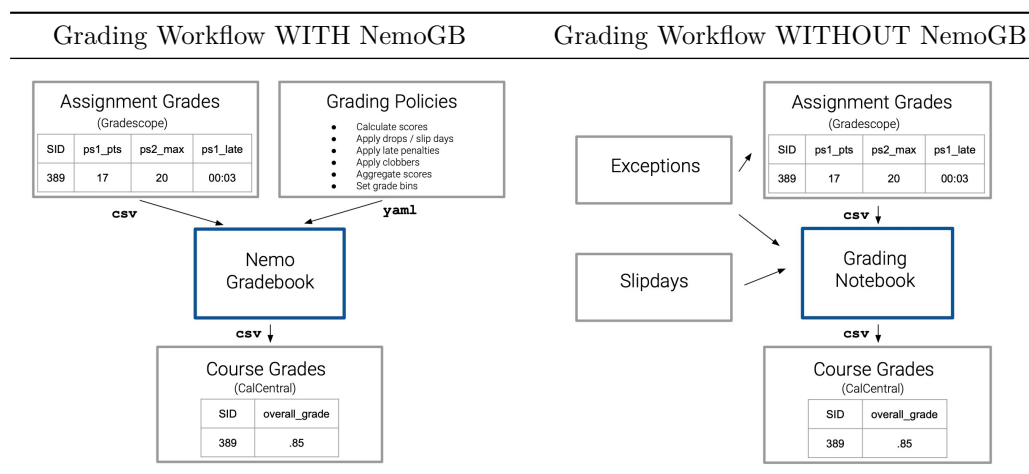
## Summary

NemoGB (Jayaprakash\* et al. 2024) allows for accurate and systematic computations of the final course letter grades. These computations require two inputs: a specifically structured YAML file representing the grading policy from the class syllabus and the assignment grades in CSV (comma-separated value) format from Gradescope (Singh et al. 2017) or other similar learning management systems, like Canvas (Canvas Community 2024). The package uses these two inputs to break down any complex syllabi into methodical computations that can be documented and tested.

## Statement of Need

While the final grade at the end of a course is an elementary part of most college courses, the computations for these grades quickly become deceptively intricate, especially with larger STEM classes that use various complexities to accommodate a diverse student body. Even though most classes use slight variations of the same policies, many LMS cannot sustain these complex computations. In response, courses will turn to hard-coded scripts. These scripts quickly accumulate hundreds of lines of code, and there is no method to assess accuracy of the final computation.

NemoGB is an R package that maintains the structure and complexity of a course grade while guaranteeing accuracy through comprehensive unit-testing. The challenges of consistency and precision in grading systems are addressed by applying the practices of data analysis and the principles of software development. The rigorous unit-testing in the package minimizes computational error and reduces the manual inputs, significantly lowers the risks of typographic and logical errors in scripts. Because of this, course grades can be computed accurately and quickly: the accuracy allows course instructors to have reliable grade computations, and the speed allows them to compute grades throughout the semester in order to monitor student progress. The structure of the package – and the open-source nature of it – allows for courses to contribute functionality that is unique to their course. This R package also functions as the backend of the NemoGB Shiny app (Voutov\*, Jayaprakash\*, and Bray 2024), which lets the user create their grading policy file in a straightforward way.



## Story of the Project

As students, we understand the importance of having an informed instructor who understands how we stand in their course. This began as an independent study centered around using R Shiny in order to create a web-based grading tool that allowed for the computation of course grades for diverse syllabi. We wanted to develop a tool that allowed for flexibility in their syllabus and precision in their computations. This led to the development of this package, which operates as the backend of the NemoGB app but can also be used independently to compute course grades.

## Underlying Principle

NemoGB breaks down the calculation of a course grade into a series of nested aggregations. It accommodates the generic policies included in most syllabi: applying lateness penalties, dropping the  $n$  lowest scores in a category, and using averages or weighted averages to aggregate assignment scores into overarching category scores. As previously mentioned, the structure of this package also allows for outside contribution of unique policies in order for any course structure to be computed with this package.

The details of the course grading structure – usually detailed in the syllabus or on the class website – can be articulated in YAML format using a series of accepted keys (e.g. **score**, **aggregation**, **lateness**, **drop\_n\_lowest**, etc.). More direction about creating a policy file is provided in the **Building a Policy File** vignette. The nested structure of this policy file reflects the nested structure of the course grade. The assignment scores come directly from Gradescope or from Canvas in a .csv file. These two files (the YAML policy file and the Gradescope or Canvas data) function as the two inputs for NemoGB's primary and overarching function: `get_grades()`. After reading in the assignment data from Gradescope or Canvas using `read_gs()` and reading in their YAML policy file (that reflects their course syllabus) using `read_policy()`, this singular function computes the entirety of the final course grade computation.

While `get_grades()` encapsulates the entire computational functionality of the R package, it is comprised of four sequential functions:

- `process_gs()` ensures the correct format of the Gradescope or Canvas csv.
- `process_policy()` similarly ensures the correct format of the policy file.

- `reconcile_policy_with_gs()` checks the compatibility of the policy file and the Gradescope or Canvas data.
- `calculate_grades()` computes the course grades and returns the final grade (and the scores for every intermediate category) appended to the original data.

## Comparison to Other Packages

Most other commonly-used packages – particularly for R packages – are used for grading on an assignment-level basis. For example, the `gradeR` package “helps grade your students’s assignment submissions that are R Scripts” (Brown 2021) whereas `NemoGB` is used for the computations of the final, overall course grade. The software that has the most similar computational purpose as `NemoGB` are popular learning-management systems (LMS) used in higher education. This includes Canvas, Moodle (Moodle Docs 2024), Blackboard Learn (Blackboard Help 2024), and D2L Brightspace (D2L Brightspace Community 2024), all of which provide an integrated gradebook that allow the instructor to specify the manner in which assessment scores are used to calculate a final course grade. What makes Gradebook unique is its flexibility of functionality and its capacity for instructor collaboration and contribution: the flexible YAML structure allows for the former and the open-source nature of the project allows for the latter.

## Acknowledgements

The authors would like to thank Iain Carmichael and Calvin Carter for helpful ideas and discussions throughout the development of this project.

As of summer 2024, this project was funded by the an Instructional Technology and Innovation Micro Grant Program through the University of California, Berkeley.

## References

- Blackboard Help. 2024. “Calculate Grades.” [https://help.blackboard.com/Learn/Instructor/Ultra/Grade/Grading\\_Tasks/Calculate\\_Grades](https://help.blackboard.com/Learn/Instructor/Ultra/Grade/Grading_Tasks/Calculate_Grades).
- Brown, Taylor. 2021. “gradeR: Helps Grade Assignment Submissions That Are r Scripts.” *GitHub Repository*. <https://github.com/tbrown122387/gradeR.git>; GitHub.
- Canvas Community. 2024. “How Do i Use the Gradebook?” <https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-use-the-Gradebook/ta-p/701>.
- D2L Brightspace Community. 2024. “About Grades.” <https://community.d2l.com/brightspace/kb/articles/3305-about-grades>.
- Jayaprakash\*, Nikita, Monika Voutov\*, Zach Turner, and Andrew Bray. 2024. “NemoGB Package.” *GitHub Repository*. <https://github.com/nemogb-dev/nemogb-r.git>; GitHub.
- Moodle Docs. 2024. “Grade Calculations.” [https://docs.moodle.org/404/en/Grade\\_calculations](https://docs.moodle.org/404/en/Grade_calculations).
- Singh, Arjun, Sergey Karayev, Kevin Gutowski, and Pieter Abbeel. 2017. “Gradescope: A Fast, Flexible, and Fair System for Scalable Assessment of Handwritten Work.” In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, 81–88. L@S ’17. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3051457.3051466>.
- Voutov\*, Monika, Nikita Jayaprakash\*, and Andrew Bray. 2024. “NemoGB Web App.” *GitHub Repository*. <https://github.com/nemogb-dev/nemogb-app.git>; GitHub.