Towards Automatic Grading of D3.js Visualizations

Matthew Hull*
Georgia Institute of Technology

Connor Guerin†
Georgia Institute of Technology

Justin Chen ‡
Georgia Institute of Technology

Susanta Routray §
Georgia Institute of Technology

Duen Horng (Polo) Chau ¶
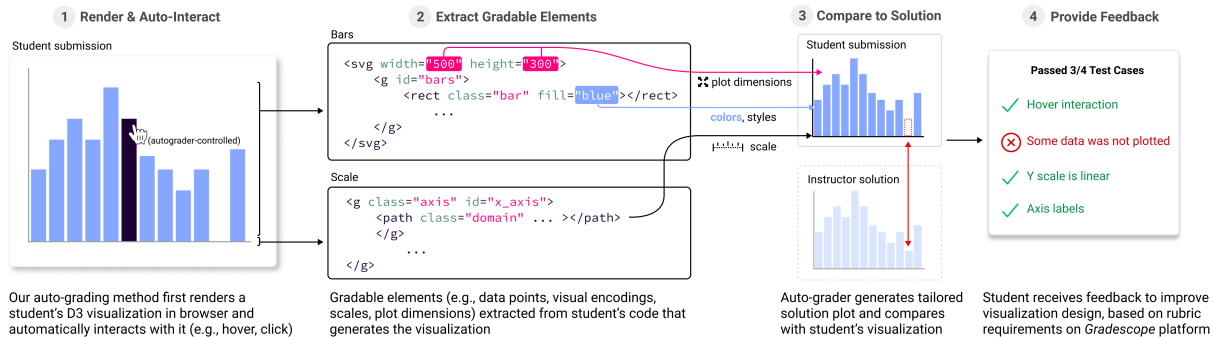Georgia Institute of Technology

Figure 1: Our auto-grading workflow for D3 visualizations (1) first renders student's plot in a web browser and automatically interacts with it (e.g., hover, click); (2) extracts gradable elements (e.g., data points, visual encodings, plot dimensions) from the plot's backing code; (3) generates a student-specific solution and compares it with student's plot to identify issues; and (4) provides feedback to help student improve their design.

ABSTRACT

Manually grading D3 data visualizations is a challenging endeavor, and is especially difficult for large classes with hundreds of students. Grading an interactive visualization requires a combination of interactive, quantitative, and qualitative evaluation that are conventionally done manually and are difficult to scale up as the visualization complexity, data size, and number of students increase. We present a first-of-its kind automatic grading method for D3 visualizations that scalably and precisely evaluates the data bindings, visual encodings, interactions, and design specifications used in a visualization. Our method has shown potential to enhance students' learning experience, enabling them to submit their code frequently and receive rapid feedback to better inform iteration and improvement to their code and visualization design. Our method promotes consistent grading and enables instructors to dedicate more focus to assist students in gaining visualization knowledge and experience. We have successfully deployed our method and auto-graded D3 submissions from more than 1000 undergraduate and graduate students in Georgia Tech's *CSE6242 Data and Visual Analytics* course, and received positive feedback and encouragement for expanding its adoption.

Index Terms: Human-centered computing—Visualization—Visualization design and evaluation methods

1 INTRODUCTION

Manually grading D3 [5] data visualizations is a challenging endeavor, and is especially difficult for large classes with hundreds

---

*e-mail: matthewhull@gatech.edu
†e-mail: cguerin6@gatech.edu
‡e-mail: jchen3001@gatech.edu
§e-mail: sroutray@gatech.edu
¶e-mail: polo@gatech.edu

of students. Even for a "simpler" interactive bar plot visualization, manually grading it usually requires a combination of interactive, quantitative, and qualitative evaluation, with tasks such as verifying the correctness of data bindings and visual encodings, testing interactive elements and actions work as intended (e.g., whether tooltip displays upon mouse-hover), and determining if a visualization's overall layout and spacing is appropriate. However, such "simple" evaluation tasks quickly become daunting as the visualization complexity and data size increase. For a large class with hundreds or a thousand students, such manual grading becomes extremely tedious. Furthermore, manual grading presents the logistic challenge of requiring congruence between student and grader environments. This necessitates the recreation of the student's environment or in some cases, modification of student code by the grader to run the visualization, resulting in a cumbersome grading process, especially for large classes with many students. To address the common challenges in evaluating D3 visualizations, our ongoing work makes the following contributions:

1. **We present a first-of-its-kind automatic grading approach for D3 visualizations** that scalably and precisely evaluates the data bindings, visual encodings, interactions, and design specifications used in a visualization. Our method avoids a rigid "one-size-fits-all" approach to grading, by offering a novel flexible way that provides students with tailored feedback while supporting their design freedom in developing visualizations. For students' visualizations to be scalably auto-graded using our approach, the only requirement is to assign identifiers to a small number of high-level structures of a visualization so that "gradable" elements may be extracted for grading. Our approach is inspired by Harper and Agrawala's D3 deconstruction approach [6]. For example, assigning the "bar" CSS class name to the group of bars in bar plot (see Fig. 1.2) enables all the bars and their visual properties to be precisely extracted. Our approach scalably and systematically accomplishes important grading tasks that are otherwise tedious, such as verifying

whether all data are plotted, or correct scale types are used. Our approach provides ample room for students to exercise their design freedom. For example, as illustrated in Fig. 1.3, when evaluating a student's bar plot, the auto-grader generates a *solution* plot specifically for that student (e.g., using plot dimensions, color schemes chosen by student), then compares the solution plot with the student's plot via instructor-supplied *test cases* that help students identify issues. Students receive textual feedback on test cases that they pass or fail, and visual feedback in the form of a downloadable image containing their rendered visualization produced during auto-grading (Fig. 1.4). Students can run the auto-grader as often as they want before the assignment due date, students' learning experience is enhanced through such frequent feedback while they iteratively improve their work while gaining clarity in how their work is graded.

2. **Grading Interactivity.** D3 visualizations may be designed to incorporate a myriad of interactive features, ranging from simple actions such as *hover* and *click*, to more complex actions such as filtering a dataset, drag-drop, or displaying a subplot. We address this challenge to grade interactivity by utilizing Selenium's [4] browser automation package to compose rich interaction sequences, known as *Action Chains*, that facilitate the evaluation of common user interaction. As a result, our auto-grader can efficiently and automatically interact with and grade interactions and "hidden" visualization elements that display after a complex sequence of interactions. Without our approach, such grading would need to be tediously performed for every student's submission.

3. **Large Scale D3 Auto-Grading: Early Usage.** To demonstrate the feasibility and scalability of our approach, we have successfully deployed our auto-grader on the Gradescope [1] platform, and auto-graded D3 visualization submissions from over 1000 undergraduate and graduate students in Georgia Tech's *CSE6242 Data and Visual Analytics*[1] course during the Spring 2021 semester. We have enjoyed positive feedback from both students and members of the instruction team. Excited with this early success, we proceeded and have completed the work necessary to release the first entirely auto-graded D3 class assignment for the Fall 2021 semester; the assignment consists of designing and implementing 4 complex interactive data visualizations.

2 AUTO-GRADING IMPACT AND INITIAL FEEDBACK

2.1 Student Impact

Improved Student-Instructor Interaction. Since deploying the auto-grader, we observed its positive effects on our *Data and Visual Analytics* course's 1000+ students at Georgia Tech via student activity on Piazza, the course's discussion forum. Specifically, by comparing the content of Piazza posts across semesters, we found that the auto-grader had greatly reduced the number of question posts related to grading (*"would I lose points for ___?"*), acceptable code functions (*"can I use ___?"*), and visualization styling. This reduction was likely due to our approach supporting flexibility in students' design choices and offering immediate feedback as they evolved their visualizations. We are excited by having increased student-instructor interaction on more qualitative topics about gaining knowledge and experience with visualization design and implementation, and less on logistical issues that could distract students.

Reducing Configuration Errors. Our approach has dramatically reduced configuration errors that students may experience, such as incorrectly referenced libraries or data files; before using auto-grader, such errors would prevent the visualization from being rendered or

graded, and graders would need to manually resolved them. The auto-grader re-creates the student's development environment and eliminates potential system configuration mismatch between student and instructor machines. Each semester, a student may request that an assignment be regraded if they feel there was an error in the grading process. We saw that our auto-graded D3 question received only 1 regrade request, compared to 22 in the previous semester for the same question that was manually graded, 19 of which were due to system configuration issues that could not have occurred in the unified auto-grader environment.

2.2 Instructor Impact

Our method has greatly reduced the time and effort required from instructors when evaluating students' work. In previous semesters, it would take each grader hours to assess 100–150 student submissions for an assignment question. With the auto-grading method in place, not only was the overall grading time and effort reduced, but so was the effort needed for processing regrade requests and responding to discussion posts on Piazza. Indeed, since students' work was immediately auto-graded as they ran the auto-graders, grading was effectively "done" as soon as the assignment was due. In other words, the instruction team only needed to review a small number of submissions that received anomalous scores (e.g., a student's final submission received lower scores than earlier submissions). We found that with this reduction in workload and subsequent increase in availability, we can dedicate more focus to instruction and assisting students.

3 CONCLUSION AND FUTURE WORK

We believe our auto-grading method is a major first step in scaling up instruction and evaluation of D3 visualizations. We are excited by our successful deployment and the positive feedback from students and members of the instruction team. We plan to extend our auto-grader's qualitative evaluation ability to consider aspects like color choices, to ensure that the visualization is readable and accessible. To further improve students' learning experience, we plan to provide more visual feedback, such as highlighting parts of a visualization that cause an issue, similar to the annotation method used by Hopkins *et al.* [7]. Our auto-grading method is not limited to pure D3 applications — it is readily extendable to other visualization platforms based on D3, which include popular visualization tools such as Vega-Lite [8], Observable Notebooks [2], and the recently-released Observable Plot [3].

REFERENCES

[1] Gradescope. https://www.gradescope.com. Accessed: 2020-05-01.
[2] Observable notebooks. https://observablehq.com. Accessed: 2021-02-01.
[3] Observable plot. https://observablehq.com/@observablehq/plot. Accessed: 2021-05-04.
[4] Selenium. https://www.selenium.dev. Accessed: 2021-02-01.
[5] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *TVCG (Proc. InfoVis)*, 2011.
[6] J. Harper and M. Agrawala. Deconstructing and restyling d3 visualizations. In *UIST*, pp. 253–262, 2014.
[7] A. K. Hopkins, M. Correll, and A. Satyanarayan. Visualint: Sketchy in situ annotations of chart construction errors. In *Computer Graphics Forum*, vol. 39, pp. 219–228. Wiley Online Library, 2020.
[8] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *TVCG (Proc. InfoVis)*, 2017.

---

[1] https://poloclub.github.io/#cse6242